# Matroid-Constrained Approximately Supermodular Optimization for Near-Optimal Actuator Scheduling

Luiz F. O. Chamon, Alexandre Amice, and Alejandro Ribeiro

*Abstract*— **This work considers the problem of scheduling actuators to minimize the Linear Quadratic Regulator (LQR) objective. In general, this problem is NP-hard and its solution can therefore only be approximated even for moderately large systems. Although convex relaxations have been used to obtain these approximations, they do not come with performance guarantees. Another common approach is to use greedy search. Still, classical guarantees do not hold for the scheduling problem because the LQR cost function is neither submodular nor supermodular. Though surrogate supermodular figures of merit, such as the $\log\det$ of the controllability Gramian, are often used as a workaround, the resulting problem is not equivalent to the original LQR one. This work shows that no change to the original problem is needed to obtain performance guarantees. Specifically, it proves that the LQR cost function is approximately supermodular and provides new near-optimality certificates for the greedy minimization of these functions over a generic matroid. These certificates are shown to approach the classical $1/2$ guarantee of supermodular functions in relevant application scenarios.**

## I. INTRODUCTION

Many problems in control theory and application are combinatorial in nature, such as selecting sensors for state estimation, actuators for regulation, or allocating tasks to autonomous agents [1]–[4]. In these problems, we seek to choose from a discrete set of possible elements (sensors, tasks, or actuators) so as to optimize some objective, e.g., the state estimation mean-square estimation (MSE), the number of agents required or the time to complete all tasks, or some control performance measure. What makes these problems difficult is the fact that only a limited number of elements can be chosen due to cost, power, or communication constraints. This leads to discrete optimization problems that are NP-hard in general and whose solutions can therefore only be approximated even for moderately large systems [5]–[11].

To make the discussion more concrete, consider the problem known as actuator scheduling in which we seek to select a given number of actuators per time instant so as to minimize some control cost. One approach to approximate the optimal solution of this problem is using a convex relaxation as in [12]–[15], typically including a sparsity promoting regularization. Though practical, these methods do not come with approximation guarantees. Another common avenue is to build the schedule using discrete optimization methods, such as tree pruning [16] or greedy search [17]–[19]. The latter solution is attractive in practice due to its low complexity and iterative nature: actuators are matched to time instants one-by-one by selecting the feasible match that most reduces the objective. What allows us to construct solutions in this way is the fact that the constraint on the number of actuators can be mapped into a combinatorial structure called a matroid, a family of sets that can be built element by element [20, Ch. 39]. Without a theoretical guarantee, however, it is unclear how well the actuators selected by this heuristic perform due to the intractability of the problem.

Still, near-optimal guarantees exist for greedy solutions in certain circumstances. For generic matroid-constrained minimization problems, such as actuator scheduling (see Section II), a common guarantee relies on the supermodularity of the objective, i.e., on a form of diminishing return displayed by set functions such as the log determinant or the rank of the controllability Gramian. Greedily minimizing these supermodular functions over an arbitrary matroid yields $1/2$-optimal solutions [21]. Other methods based on continuous extensions of these functions can be used to improve this factor to $1 - 1/e$ [22]. Nevertheless, many cost functions of interest are not supermodular. In particular, this is the case of the LQR objective for actuator scheduling [9], [11].

In this work, we address this issue using the concept of approximate supermodularity [4], [23]–[25]. We start by casting the actuator scheduling problem as a matroid-constrained optimization problem and showing how it can be solved greedily (Section II). Since the resulting constraints are more general than bounding set cardinality, we derive a novel near-optimality certificate for the greedy minimization of approximately supermodular set functions over a generic matroid (Section III). We then prove that the LQR cost function is approximately supermodular by bounding how much it violates the diminishing returns property, giving explicit guarantees on the greedy actuator scheduling solution (Section IV). We argue that these guarantees improve in scenarios of practical interest and illustrate their typical values in simulations (Section V).

**Notation**: Lowercase and uppercase boldface letters represent vectors ($\boldsymbol{x}$) and matrices ($\boldsymbol{X}$) respectively, while calligraphic letters denote sets ($\mathcal{A}$). We write $|\mathcal{A}|$ for the cardinality of $\mathcal{A}$ and $\boldsymbol{X} \succeq 0$ ($\boldsymbol{X} \succ 0$) to say $\boldsymbol{X}$ is positive semi-definite (definite) matrix. Hence, $\boldsymbol{X} \preceq \boldsymbol{Y} \Leftrightarrow \boldsymbol{b}^T \boldsymbol{X} \boldsymbol{b} \leq \boldsymbol{b}^T \boldsymbol{Y} \boldsymbol{b}$ for all $\boldsymbol{b} \in \mathbb{R}^n$. We write $\lambda_{\text{max}}$ and $\lambda_{\text{min}}$ for the maximum and minimum eigenvalue of a matrix, respectively, and use $\mathbb{R}_+$ to denote the non-negative real numbers.

## II. PROBLEM FORMULATION

Consider a discrete-time, time invariant dynamical system and denote its set of available inputs/actuators at time $k$

Electrical and Systems Engineering, University of Pennsylvania. {luizf, amice, aribeiro}@seas.upenn.edu. This work is supported by the ARL DCIST CRA W911NF-17-2-0181.

by $\mathcal{V}_k$. Suppose that this system is actuated only through a subset $\mathcal{S}_k \subseteq \mathcal{V}_k$ of its inputs, so that its states $\boldsymbol{x}_k \in \mathbb{R}^n$ evolve according to

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \sum_{i\in\mathcal{S}_k} \boldsymbol{b}_i u_{i,k}, \tag{1}$$

where $\boldsymbol{b}_i \in \mathbb{R}^n$ is the input vector and $u_{i,k}$ is the control action of the $i$-th input and $\boldsymbol{A}$ is the state transition matrix. Let $\overline{\mathcal{V}} = \mathcal{V}_0 \cup \mathcal{V}_1 \cup \cdots \cup \mathcal{V}_{N-1}$ be the set of all actuators available throughout the time window $k = 0, 1, \ldots, N-1$ and call $\overline{\mathcal{V}} \supseteq \mathcal{S} = \mathcal{S}_0 \cup \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_{N-1}$ a *schedule*, as it denotes the set of active inputs at each time instant. Assume without loss of generality that $\mathcal{V}_j \cap \mathcal{V}_k = \emptyset$ for all $j \neq k$. If an input $v$ is available at different instants, it can be represented by different elements, e.g., $v^j \in \mathcal{V}_j$ and $v^k \in \mathcal{V}_k$.

Actuator scheduling seeks to choose which actuators are used at each time step so as to minimize a control cost subject to a budget constraint. Formally, given a limit $s_k$ on the number of active inputs at iteration $k$, we seek a feasible schedule that minimizes the value of the LQR problem, i.e.,

$$\begin{aligned} \underset{\mathcal{S}\subseteq\overline{\mathcal{V}}}{\text{minimize}} \quad & J(\mathcal{S}) \triangleq V(\mathcal{S}) - V(\emptyset) \\ \text{subject to} \quad & |\mathcal{S} \cap \mathcal{V}_k| \leq s_k, \quad k = 0, \ldots, N-1 \end{aligned} \tag{PI}$$

with

$$V(\mathcal{S}) = \min_{\mathcal{U}(\mathcal{S})} \mathbb{E}\left[\boldsymbol{x}_N^T \boldsymbol{Q}_N \boldsymbol{x}_N + \sum_{k=0}^{N-1}\left(\boldsymbol{x}_k^T \boldsymbol{Q}_k \boldsymbol{x}_k + \sum_{i\in\mathcal{S}\cap\mathcal{V}_k} r_{i,k} u_{i,k}^2\right)\right], \tag{2}$$

where $\mathcal{U}(\mathcal{S}) = \{u_{i,k} \mid i \in \mathcal{S} \cap \mathcal{V}_k\}_{k=0}^{N-1}$ is the set of control actions, $\boldsymbol{Q}_k \succ 0$ are the state weights, and $r_{i,k} > 0$, for all $i$ and $k$, are the input weights. The expectation in (2) is taken with respect to the initial state $\boldsymbol{x}_0 \sim \mathcal{N}(\bar{\boldsymbol{x}}_0, \boldsymbol{\Pi}_0)$ assumed to be a Gaussian random variable with mean $\bar{\boldsymbol{x}}_0$ and covariance $\boldsymbol{\Pi}_0 \succ 0$. Observe that the constant $V(\emptyset)$ in the objective of (PI) does not affect the solution of the optimization problem. It is used so that $J(\emptyset) = 0$, which simplifies the presentation of our near-optimal certificates (see Section III).

Note from (2) that the objective of (PI) itself involves a minimization. However, it reduces to the classical LQR problem for any fixed schedule $\mathcal{S} \subseteq \overline{\mathcal{V}}$. Hence, it has a closed form expression that we collect in the following proposition.

**Proposition 1.** *The value of the LQR problem in* (2) *can be written as*

$$V(\mathcal{S}) = \operatorname{Tr}\left[\boldsymbol{\Pi}_0 \boldsymbol{P}_0(\mathcal{S})\right], \tag{3}$$

*where $\boldsymbol{P}_0(\mathcal{S})$ is obtained via the backward recursion*

$$\boldsymbol{P}_k(\mathcal{S}) = \boldsymbol{Q}_k + \boldsymbol{A}^T\left(\boldsymbol{P}_{k+1}^{-1}(\mathcal{S}) + \sum_{i\in\mathcal{S}\cap\mathcal{V}_k} r_{i,k}^{-1}\boldsymbol{b}_i\boldsymbol{b}_i^T\right)^{-1}\boldsymbol{A}, \tag{4}$$

*starting with $\boldsymbol{P}_N = \boldsymbol{Q}_N$.*

**Algorithm 1** Greedy search for minimization over a matroid

Given an objective function $f$ and a matroid $(\mathcal{E}, \mathcal{I})$, initialize $\mathcal{G}_0 \leftarrow \emptyset$, $\mathcal{Z} \leftarrow \mathcal{E}$, and $t \leftarrow 0$
**while** $\mathcal{Z} \neq \emptyset$
  $g \leftarrow \operatorname{argmin}_{u\in\mathcal{Z}} f(\mathcal{G}_t \cup \{u\})$
  $\mathcal{Z} \leftarrow \mathcal{Z} \setminus \{g\}$
  **if** $\mathcal{G}_t \cup \{g\} \in \mathcal{I}$ **then** $\mathcal{G}_{t+1} \leftarrow \mathcal{G}_t \cup \{g\}$ and $t \leftarrow t+1$
**end**

*Proof.* This result follows directly from the classical dynamic programming argument for the LQR by taking into account only the active actuators in $\mathcal{S}$ (e.g., see [26]). For ease of reference, we provide a derivation in the appendix. ■

Although (PI) is NP-hard in general [5]–[11], its feasible sets have a matroidal structure. Matroids are algebraic structures that generalize the notions of linear independence in a vector space. Formally,

**Definition 1.** *A matroid $M = (\mathcal{E}, \mathcal{I})$ consists of a finite set of elements $\mathcal{E}$ and a family $\mathcal{I} \subseteq 2^{\mathcal{E}}$ of subsets of $\mathcal{E}$, called* independent sets*, that satisfy:*

*1) $\emptyset \in \mathcal{I}$;*
*2) if $\mathcal{A} \subseteq \mathcal{B}$ and $\mathcal{B} \in \mathcal{I}$, then $\mathcal{A} \in \mathcal{I}$;*
*3) if $\mathcal{A}, \mathcal{B} \in \mathcal{I}$ and $|\mathcal{A}| < |\mathcal{B}|$, then there exists $e \in \mathcal{B} \setminus \mathcal{A}$ such that $\mathcal{A} \cup \{e\} \in \mathcal{I}$.*

The most common example of matroid is the *uniform matroid*, for which the independent sets are defined as $\mathcal{I}_u = \{\mathcal{A} \subseteq \mathcal{E} \mid |\mathcal{A}| \leq k\}$ for some $k \geq 1$. It is ready that $\mathcal{I}_u$ obeys the properties in Definition 1. These matroids are found in cardinality constrained problems, such as actuator or sensor selection [4], [7]–[10], [27]. The feasible sets of problem (PI), on the other hand, form a more general structure known as a *partition matroid*: given a partition $\{\mathcal{E}_k\}$ of the ground set $\mathcal{E}$, i.e., $\mathcal{E} = \cup_k \mathcal{E}_k$ and $\mathcal{E}_j \cap \mathcal{E}_k = \emptyset$ for $j \neq k$, and parameters $a_k$, the independent sets of a partition matroid are given by $\mathcal{I}_p = \{\mathcal{A} \subseteq \mathcal{E} \mid |\mathcal{A} \cap \mathcal{E}_k| \leq a_k\}$.

Notice that properties 1 and 2 from Definition 1 imply that the independent sets of matroids can be constructed iteratively, element-by-element, which suggests that a solution of (PI) could be obtained by using the greedy procedure in Algorithm 1 with $\mathcal{E} = \overline{\mathcal{V}}$, $f(\mathcal{S}) = J(\mathcal{S})$, and $\mathcal{I} = \{\mathcal{A} \subseteq \overline{\mathcal{V}} \mid |\mathcal{A} \cap \mathcal{V}_k| \leq s_k\}$. Although this greedy solution $\mathcal{G}$ is feasible by construction, its performance depends on the objective function: if $J$ were to be a monotone decreasing modular set function, then $\mathcal{G}$ would be an optimal solution [20, Ch. 40]; if $J$ were a monotone decreasing supermodular function, then $\mathcal{G}$ would be 1/2-optimal [21]. However, the objective $J$ of (PI) is neither modular nor supermodular [9], [11]. Hence, existing guarantees do not apply to its greedy solution. To address this issue, we leverage the theory of $\alpha$-supermodular set functions and derive new near-optimal certificates for their minimization subject to an arbitrary matroid constraint. Though it may seem superfluous, the derivation of these guarantees heavily depends on property 3 in Definition 1, known as the augmentation property. We then show that $J$ in (PI) is $\alpha$-supermodular, obtaining guarantees for the greedy solution of the actuator scheduling problem.

## III. MATROID-CONSTRAINED $\alpha$-SUPERMODULAR MINIMIZATION

The concept of *$\alpha$-supermodularity* allows for relaxations or strengthenings of the classical definition of supermodularity, a diminishing returns property often used to obtain near-optimality results. Motivated by the fact that supermodularity is a stringent condition to satisfy in general, $\alpha$-supermodular functions allow for controlled violations of the original diminishing returns property. The idea is that if a function is close to supermodular, then its properties should not differ too much from a truly supermodular function. Indeed, greedily minimizing an $\alpha$-supermodular function subject to a cardinality constraint is $(1-e^{-\alpha})$-optimal [4], [24], [25]. This concept was first proposed in the context of auction design [23], though it has since been used in discrete optimization, estimation, and control [4], [24], [25], [28].

Formally, a set function $f : 2^{\mathcal{E}} \to \mathbb{R}$ is supermodular if the diminishing return property

$$f(\mathcal{A}) - f(\mathcal{A} \cup \{e\}) \geq f(\mathcal{B}) - f(\mathcal{B} \cup \{e\}) \tag{5}$$

holds for all sets $\mathcal{A} \subset \mathcal{B} \subseteq \mathcal{E}$ and all $e \in \mathcal{E} \setminus \mathcal{B}$. We say $f$ is *submodular* if $-f$ is supermodular. The same $f$ is said to be *$\alpha$-supermodular* if $\alpha \in \mathbb{R}_+$ is the largest number for which

$$f(\mathcal{A}) - f(\mathcal{A} \cup \{e\}) \geq \alpha \left[f(\mathcal{B}) - f(\mathcal{B} \cup \{e\})\right] \tag{6}$$

holds for all $\mathcal{A} \subset \mathcal{B} \subseteq \mathcal{E}$ and $e \in \mathcal{E} \setminus \mathcal{B}$. Explicitly,

$$\alpha = \min_{\substack{\mathcal{A} \subset \mathcal{B} \subseteq \mathcal{E} \\ e \in \mathcal{E} \setminus \mathcal{B}}} \frac{f(\mathcal{A}) - f(\mathcal{A} \cup \{e\})}{f(\mathcal{B}) - f(\mathcal{B} \cup \{e\})}. \tag{7}$$

Notice that $\alpha \geq 1$ in (6) implies (5), in which case we refer to $f$ simply as *supermodular* [29], [30]. When $\alpha \in (0,1)$, we say $f$ is *approximately supermodular*. Notice that (6) always holds for $\alpha = 0$ if $f$ is monotone decreasing. Indeed, a set function $f$ is *monotone decreasing* if for all $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{E}$ it holds that $f(\mathcal{A}) \geq f(\mathcal{B})$. Finally, say a set function $f$ is normalized if $f(\emptyset) = 0$.

Before proceeding, we provide an equivalent definition of $\alpha$-supermodularity in terms of set-valued updates that is convenient for the remaining derivations:

**Proposition 2.** *The function $f : 2^{\mathcal{E}} \to \mathbb{R}$ is an $\alpha$-supermodular set function if and only if*

$$f(\mathcal{Y}) - f(\mathcal{Y} \cup \mathcal{C}) \geq \alpha \left[f(\mathcal{Z}) - f(\mathcal{Z} \cup \mathcal{C})\right] \tag{8}$$

*for all sets $\mathcal{Y} \subset \mathcal{Z} \subseteq \mathcal{E}$ and $\mathcal{C} \in \mathcal{E} \setminus \mathcal{Z}$.*

*Proof.* The necessity part is straightforward: taking $\mathcal{C} = \{e\}$ for $e \in \mathcal{E} \setminus \mathcal{Z}$ in (8) yields (6). Sufficiency follows by induction. Consider an arbitrary enumeration of $\mathcal{C}$ and let $\mathcal{C}_t = \{c_1, \ldots, c_t\}$ be the set containing its first $t$ elements. Then, the base case $\mathcal{C}_1 = \{c_1\}$ holds directly from the definition of $\alpha$-supermodularity in (6). Suppose now that (8) holds for $\mathcal{C}_{t-1}$, i.e.,

$$f(\mathcal{Y}) - f(\mathcal{Y} \cup \mathcal{C}_{t-1}) \geq \alpha \left[f(\mathcal{Z}) - f(\mathcal{Z} \cup \mathcal{C}_{t-1})\right]. \tag{9}$$

Take $\mathcal{A} = \mathcal{Y} \cup \mathcal{C}_{t-1}$, $\mathcal{B} = \mathcal{Z} \cup \mathcal{C}_{t-1}$, and $e = c_t$ in (6) yields

$$\begin{aligned} f(\mathcal{Y} \cup \mathcal{C}_{t-1}) - f(\mathcal{Y} \cup \mathcal{C}_{t-1} \cup \{c_t\}) \geq \\ \alpha \left[f(\mathcal{Z} \cup \mathcal{C}_{t-1}) - f(\mathcal{Z} \cup \mathcal{C}_{t-1} \cup \{c_t\})\right]. \end{aligned} \tag{10}$$

Summing (9) and (10) proves (8) holds for $\mathcal{C}_t = \mathcal{C}_{t-1} \cup \{c_t\}$. ■

It is worth noting that $\alpha$ in (7) is related to the *submodularity ratio*, which is based on the relaxation of a different, though equivalent, definition of submodularity. Explicitly, the submodularity ratio is defined as the largest $\gamma$ such that

$$\sum_{u \in \mathcal{W}} \left[f(\mathcal{L} \cup \{u\}) - f(\mathcal{L})\right] \geq \gamma \left[f(\mathcal{L} \cup \mathcal{W}) - f(\mathcal{L})\right], \tag{11}$$

for all disjoint sets $\mathcal{W}, \mathcal{L} \subseteq \mathcal{V}$. It was introduced in [5] in the context of variable selection, though the resulting guarantees depended on the sparse eigenvalues of a matrix which are NP-hard to compute. The first explicit (P-computable) bounds on $\alpha$ were obtained in [4], [25], though the same bounds were more recently derived for $\gamma$ as well [28], [31], [32]. This is not surprising in view of the the following proposition that formalizes the relation between $\alpha$ and $\gamma$.

**Proposition 3.** *Let $f$ be an $\alpha$-submodular and denote its submodularity ratio by $\gamma$. Then, $\alpha \leq \gamma$.*

*Proof.* We proceed by showing that (11) holds with $\gamma = \alpha$ for any $\alpha$-submodular function. To do so, consider an enumeration of $\mathcal{W} = \{w_1, \ldots, w_{|\mathcal{W}|}\}$ to obtain

$$\begin{aligned} f(\mathcal{L} \cup \mathcal{W}) - f(\mathcal{L}) = \sum_{k=1}^{|\mathcal{W}|} \Big[ f(\mathcal{L} \cup \{w_1, \ldots, w_k\}) - \\ f(\mathcal{L} \cup \{w_1, \ldots, w_{k-1}\}) \Big]. \end{aligned} \tag{12}$$

Since $f$ is $\alpha$-submodular, it holds that $-f$ is $\alpha$-supermodular. Hence, we can upper bound each of the increments in (12) using (6) to obtain

$$f(\mathcal{L} \cup \mathcal{W}) - f(\mathcal{L}) \leq \frac{1}{\alpha} \sum_{k=1}^{|\mathcal{W}|} \left[f(\mathcal{L} \cup \{w_k\}) - f(\mathcal{L})\right]. \tag{13}$$

Comparing (11) and (13) shows that $\alpha$-submodular functions have submodularity ratio at least $\alpha$. ■

Still, guarantees in [4], [5], [24], [25], [28], [31] based on $\alpha$-supermodularity and the submodularity ratio hold for cardinality constrained problems, whereas the constraint in the scheduling problem (PI) is a partition matroid [20, Ch. 39]. The main result of this section presented in Theorem 1 is a novel near-optimality certificate for the minimization of $\alpha$-supermodular functions over arbitrary matroids. Interestingly, the result for $\alpha \geq 1$ first appeared in [23] in the context of auction design. We present here the result for general $\alpha$-supermodular functions.

**Theorem 1.** *Let $f$ be a normalized, monotone decreasing, $\alpha$-supermodular set function (i.e., $f(\mathcal{A}) \leq 0$ for all $\mathcal{A} \subseteq \mathcal{V}$)*

*and consider the problem*

$$\begin{aligned} \underset{\mathcal{X} \subseteq \mathcal{E}}{\text{minimize}} \quad & f(\mathcal{X}) \\ \text{subject to} \quad & \mathcal{X} \in \mathcal{I} \end{aligned} \tag{PII}$$

*where $(\mathcal{E}, \mathcal{I})$ is a matroid. Then,*

$$f(\mathcal{G}) \leq \min\left(\frac{\alpha}{2}, \frac{\alpha}{1+\alpha}\right) f(\mathcal{X}^\star), \tag{14}$$

*where $\mathcal{X}^\star$ is a solution of* (PII) *and $\mathcal{G}$ is the result of applying the greedy procedure from Algorithm 1 to* (PII).

Theorem 1 provides a near-optimal certificate for the greedy solution of matroid constrained $\alpha$-supermodular minimization problems of the form (PII). Since the values of $f$ are non-positive due to the normalization assumption, the results of Theorem 1 may not be straightforward to interpret. Nevertheless, (14) can also be written in terms of improvements with respect to the empty solution for unnormalized functions as in

$$\frac{f(\mathcal{G}) - f(\mathcal{X}^\star)}{f(\emptyset) - f(\mathcal{X}^\star)} \leq 1 - \min\left(\frac{\alpha}{2}, \frac{\alpha}{1+\alpha}\right). \tag{15}$$

The guarantee in (14) bounds the suboptimality of greedy search in terms of the $\alpha$-supermodularity of the cost function. When $\alpha < 1$, (14) quantifies the loss in performance guarantee due to the objective $f$ being approximately supermodular and violating the diminishing returns property. Note that the guarantee decays linearly with $\alpha$. On the other hand, when $\alpha \geq 1$, (14) shows how the classical $1/2$-optimal certificate can be strengthened for cost functions that have stronger diminishing returns structures. It is worth noting that Theorem 1 is a *worst case* bound. As is the case with the classical result for greedy supermodular minimization [21], [33], better performance is often obtained in practice.

Observe that though we obtain two different bounds depending on the value of $\alpha$, they differ by less than $0.09$ on $\alpha \in [0, 1)$. In contrast, the difference between (14) and the existing guarantee for cardinality constrained problems can go up to $0.2$, despite the fact that these constraints are also matroids. Theorem 1, however, applies to arbitrary matroid constraints, such as that in (PI). Closing this gap by leveraging continuous greedy optimization techniques [34] is the topic of future work.

We now proceed with the proof of Theorem 1.

*Proof.* The proof of this theorem follows the technique in [35] instead of the classical linear programming argument from [21]. It is based on the following lemma:

**Lemma 1.** *Let $f$ be a normalized, monotone decreasing, $\alpha$-supermodular set function and define $\mathcal{G}_t = \{g_1, \dots, g_t\}$ to be the set obtained after the $t$-th iteration of greedy search on* (PII). *There exists an enumeration of the elements of $\mathcal{X}^\star$ defining the sets $\mathcal{X}_t^\star = \{x_1^\star, \dots, x_t^\star\}$ such that*

$$f(\mathcal{G}_t) \leq \alpha f(\mathcal{X}_t^\star \cup \mathcal{G}_{t-1}) - f(\mathcal{G}_{t-1}), \quad \textit{for } \alpha \in [0,1) \tag{16}$$

$$f(\mathcal{G}_t) \leq f(\mathcal{X}_t^\star \cup \mathcal{G}_{t-1}) - \frac{1}{\alpha} f(\mathcal{G}_{t-1}), \quad \textit{for } \alpha \geq 1 \tag{17}$$

*for all $1 \leq t \leq |\mathcal{X}^\star|$.*

Before establishing these inequalities, we show how they imply (14). Using the fact that $f$ is monotone decreasing, we obtain from (16) that

$$f(\mathcal{G}_t) \leq \alpha f(\mathcal{X}_t^\star) - f(\mathcal{G}_t),$$

which yields

$$f(\mathcal{G}_t) \leq \frac{\alpha}{2} f(\mathcal{X}_t^\star). \tag{18}$$

Similarly, (17) gives

$$f(\mathcal{G}_t) \leq \frac{\alpha}{1+\alpha} f(\mathcal{X}_t^\star). \tag{19}$$

Since $\frac{\alpha}{2} \leq \frac{\alpha}{1+\alpha} \Leftrightarrow \alpha \in [0, 1]$, both guarantees can be written simultaneously as in (14).

To conclude, we still need to show that Algorithm 1 only terminates with $t = |\mathcal{X}^\star|$, in which case (18) and (19) imply (14). This is where the augmentation property property 3 in Definition 1 comes in handy. Indeed, start by noticing that $\mathcal{X}^\star$ must be a maximal independent set since $f$ is monotonically decreasing. Otherwise, there would be a $\mathcal{X}' \in \mathcal{I}$ with $|\mathcal{X}'| > |\mathcal{X}^\star|$, which would imply $f(\mathcal{X}') < |\mathcal{X}^\star|$ and violate the optimality of $\mathcal{X}^\star$. Hence, Algorithm 1 terminates with $t \leq |\mathcal{X}^\star|$. Furthermore, property 3 in Definition 1 implies that Algorithm 1 cannot discard every element of $\mathcal{X}^\star$ while $t < |\mathcal{X}^\star|$. To see why this is the case, observe that $|\mathcal{G}_t| < |\mathcal{X}^\star|$ for $t < |\mathcal{X}^\star|$. Thus, there exists an element $x' \in \mathcal{X}^\star$ such that $\mathcal{G}_t \cup \{x'\} \in \mathcal{I}$ (property 3). Thus, there exists at least one feasible element to grow $\mathcal{G}_t$ until $|\mathcal{G}_t| = |\mathcal{X}^\star|$, i.e., $t = |\mathcal{X}^\star|$. ■

We now proceed with the proof of Lemma 1.

*Proof.* We obtain both inequalities in Lemma 1 by induction. But first, we establish an inequality for $\alpha$-supermodular functions that will be useful throughout our derivations. To do so, recall that $\mathcal{X}_t^\star$ contains the first $t$ elements of an enumeration of $\mathcal{X}^\star$ and that $\mathcal{G}_t$ is the greedy solution set after the $t$-th iteration. We can then take $\mathcal{Y} = \mathcal{G}_{t-1}$, $\mathcal{Z} = \mathcal{X}_t^\star \cup \mathcal{G}_{t-1}$, and $\mathcal{C} = \{x_{t+1}^\star, g_t\}$ in (8) to obtain

$$\begin{aligned} f(\mathcal{G}_{t-1}) - f\left(\mathcal{G}_{t-1} \cup \{x_{t+1}^\star, g_t\}\right) \geq \\ \alpha\left[f(\mathcal{X}_t^\star \cup \mathcal{G}_{t-1}) - f\left(\mathcal{X}_t^\star \cup \mathcal{G}_{t-1} \cup \{x_{t+1}^\star, g_t\}\right)\right] \end{aligned}$$

and rearrange it to read

$$\begin{aligned} \alpha f(\mathcal{X}_t^\star \cup \mathcal{G}_{t-1}) \leq \alpha f\left(\mathcal{X}_{t+1}^\star \cup \mathcal{G}_t\right) \\ + f(\mathcal{G}_{t-1}) - f\left(\mathcal{G}_t \cup \{x_{t+1}^\star\}\right). \end{aligned} \tag{20}$$

With (20) in hands, we start with the proof of (16).

The base case $t = 1$ holds from the greedy property and the fact that $\alpha < 1$. Indeed, from property 2 in Definition 1, every subset of $\mathcal{X}^\star$ is a feasible set, so that $f(\mathcal{G}_1) = f(g_1) \leq f(x_1^\star) < \alpha f(x_1^\star) = \alpha f(\mathcal{X}_1^\star)$.

For the induction step, assume (16) holds for $t < |\mathcal{X}^\star|$. As we argued in the proof of Theorem 1, the fact that $\mathcal{G}_t, \mathcal{X}^\star \in \mathcal{I}$ and $|\mathcal{G}_t| < |\mathcal{X}^\star|$ implies there exists an element $x_{t+1}^\star \in \mathcal{X}^\star$ such that $\mathcal{G}_t \cup \{x_{t+1}^\star\} \in \mathcal{I}$ (property 3 in Definition 1).

Thus, Algorithm 1 will obtain a feasible $\mathcal{G}_{t+1}$ and by adding $f(\mathcal{G}_{t+1})$ to both sides of (16), we obtain

$$f(\mathcal{G}_{t+1}) \leq \alpha f(\mathcal{X}_t^\star \cup \mathcal{G}_{t-1}) - f(\mathcal{G}_{t-1}) - f(\mathcal{G}_t) + f(\mathcal{G}_{t+1}).$$

Using the bound on $\alpha f(\mathcal{X}_t^\star \cup \mathcal{G}_{t-1})$ from (20) then yields

$$\begin{aligned} f(\mathcal{G}_{t+1}) \leq \alpha f\left(\mathcal{X}_{t+1}^\star \cup \mathcal{G}_t\right) - f(\mathcal{G}_t) \\ + f(\mathcal{G}_{t+1}) - f\left(\mathcal{G}_t \cup \{x_{t+1}^\star\}\right). \end{aligned} \tag{21}$$

Since $\mathcal{G}_t \cup \{x_{t+1}^\star\} \in \mathcal{I}$, $f(\mathcal{G}_{t+1}) \leq f\left(\mathcal{G}_t \cup \{x_{t+1}^\star\}\right)$ from the greedy property of Algorithm 1, and (21) simplifies to

$$f(\mathcal{G}_{t+1}) \leq \alpha f\left(\mathcal{X}_{t+1}^\star \cup \mathcal{G}_t\right) - f(\mathcal{G}_t),$$

which implies that (16) also holds for $t+1$.

In the case of (17), note that it also holds trivially for $t = 1$ due to the greedy property and the matroid structure: $f(\mathcal{G}_1) = f(g_1) \leq f(x_1^\star) = f(\mathcal{X}_1^\star)$. Assume now that (17) holds for $t < |\mathcal{X}^\star|$. Arguing the existence of a feasible $\mathcal{G}_{t+1}$ as before, add $f(\mathcal{G}_{t+1})$ to both sides of the inequality and rearrange the terms to get

$$f(\mathcal{G}_{t+1}) \leq f(\mathcal{X}_t^\star \cup \mathcal{G}_{t-1}) - \frac{1}{\alpha} f(\mathcal{G}_{t-1}) - f(\mathcal{G}_t) + f(\mathcal{G}_{t+1}). \tag{22}$$

To proceed, divide (20) by $\alpha$ to obtain the bound

$$\begin{aligned} f(\mathcal{X}_t^\star \cup \mathcal{G}_{t-1}) - \frac{1}{\alpha} f(\mathcal{G}_{t-1}) \leq \\ f\left(\mathcal{X}_{t+1}^\star \cup \mathcal{G}_t\right) - \frac{1}{\alpha} f\left(\mathcal{G}_t \cup \{x_{t+1}^\star\}\right). \end{aligned} \tag{23}$$

Putting (22) and (23) together yields

$$\begin{aligned} f(\mathcal{G}_{t+1}) \leq f\left(\mathcal{X}_{t+1}^\star \cup \mathcal{G}_t\right) - \frac{1}{\alpha} f\left(\mathcal{G}_t \cup \{x_{t+1}^\star\}\right) \\ + f(\mathcal{G}_{t+1}) - f(\mathcal{G}_t). \end{aligned}$$

As in the derivation of (16), we can again use the augmentation property of matroids (property 3 in Definition 1) to show that $\mathcal{G}_t \cup \{x_{t+1}^\star\}$ is feasible. Thus, $f\left(\mathcal{G}_t \cup \{x_{t+1}^\star\}\right) \geq f(\mathcal{G}_{t+1})$ from the greedy property, so that

$$f(\mathcal{G}_{t+1}) \leq f\left(\mathcal{X}_{t+1}^\star \cup \mathcal{G}_t\right) + \left(\frac{\alpha - 1}{\alpha}\right) f(\mathcal{G}_{t+1}) - f(\mathcal{G}_t). \tag{24}$$

Finally, adding and subtracting $\alpha^{-1} f(\mathcal{G}_t)$ from (24) gives

$$\begin{aligned} f(\mathcal{G}_{t+1}) \leq f\left(\mathcal{X}_{t+1}^\star \cup \mathcal{G}_t\right) - \frac{1}{\alpha} f(\mathcal{G}_t) \\ + \left(\frac{\alpha - 1}{\alpha}\right) \left[f(\mathcal{G}_{t+1}) - f(\mathcal{G}_t)\right], \end{aligned}$$

from which we establish that (17) holds for $t+1$ by using the fact that $f$ is decreasing, so that $f(\mathcal{G}_{t+1}) \leq f(\mathcal{G}_t)$, and that $\alpha > 1$ to obtain

$$f(\mathcal{G}_{t+1}) \leq f\left(\mathcal{X}_{t+1}^* \cup \mathcal{G}_t\right) - \frac{1}{\alpha} f(\mathcal{G}_t). \qquad \blacksquare$$

## IV. NEAR-OPTIMAL ACTUATOR SCHEDULING

The previous section established a near-optimal certificate for the minimization of $\alpha$-supermodular functions subject to a generic matroid constraint, which is the case of the actuator scheduling problem (PI). Nevertheless, to obtain a guarantee for this problem from Theorem 1, we must bound $\alpha$ for the cost function $J$ as in the following proposition:

**Proposition 4.** *Let $\boldsymbol{A}$ in* (1) *be full rank. The normalized actuator scheduling problem objective $J$ is (i) monotonically decreasing and (ii) $\alpha$-supermodular with*

$$\alpha \geq \frac{\lambda_{\min}\left[\tilde{\boldsymbol{P}}_1^{-1}(\emptyset)\right]}{\lambda_{\max}\left[\tilde{\boldsymbol{P}}_1^{-1}(\overline{\mathcal{V}}) + \sum_{i \in \mathcal{V}_0} r_{i,0}^{-1} \tilde{\boldsymbol{b}}_i \tilde{\boldsymbol{b}}_i^T\right]} \tag{25}$$

*where $\tilde{\boldsymbol{P}}_1(\mathcal{S}) = \boldsymbol{W}^{1/2} \boldsymbol{P}_1(\mathcal{S}) \boldsymbol{W}^{1/2}$ and $\tilde{\boldsymbol{b}}_i = \boldsymbol{W}^{-1/2} \boldsymbol{b}_i$ with $\boldsymbol{W} = \boldsymbol{A}\boldsymbol{\Pi}_0 \boldsymbol{A}^T$.*

*Proof.* See appendix. $\blacksquare$

**Remark 1.** *The full rank hypothesis on $\boldsymbol{A}$ can be lifted using a continuity argument. However, the bound in* (25) *is trivial ($\alpha \geq 0$) for rank deficient state transition matrices, so we focus only on the case of practical significance.*

Proposition 4 provides a lower bound on the $\alpha$-supermodularity of the objective of (PI) in terms of the parameters of the dynamical system and the weights $\boldsymbol{Q}_k$ and $r_{i,k}$ in (2). Notice that the bound is explicit: (25) can be evaluated directly given the problem parameters. In other words, (25) allows us to determine $\alpha$ *a priori* for the objective $J$ and with Theorem 1, give near-optimality guarantees on the greedy solution of (PI). We collect this result in the theorem below.

**Theorem 2.** *Consider the actuator scheduling problem* (PI)*, its optimal solution $\mathcal{S}^\star$, and its greedy solution $\mathcal{G}$ obtained from Algorithm 1. Then,*

$$J(\mathcal{G}) \leq \min\left(\frac{\alpha}{2}, \frac{\alpha}{1+\alpha}\right) J(\mathcal{S}^\star) \tag{26}$$

*with*

$$\alpha \geq \frac{\lambda_{\min}\left[\tilde{\boldsymbol{P}}_1^{-1}(\emptyset)\right]}{\lambda_{\max}\left[\tilde{\boldsymbol{P}}_1^{-1}(\overline{\mathcal{V}}) + \sum_{i \in \mathcal{V}_0} r_{i,0}^{-1} \tilde{\boldsymbol{b}}_i \tilde{\boldsymbol{b}}_i^T\right]}, \tag{27}$$

*where $\tilde{\boldsymbol{P}}_1(\mathcal{S}) = \boldsymbol{W}^{1/2} \boldsymbol{P}_1(\mathcal{S}) \boldsymbol{W}^{1/2}$ with $\boldsymbol{P}_1(\mathcal{S})$ defined recursively as in* (4)*, $\tilde{\boldsymbol{b}}_i = \boldsymbol{W}^{-1/2} \boldsymbol{b}_i$, and $\boldsymbol{W} = \boldsymbol{A}\boldsymbol{\Pi}_0 \boldsymbol{A}^T$.*

Theorem 2 provides a near-optimal certificate for the greedy solution of the actuator scheduling problem (PI) depending on its parameters. Note that the larger the $\alpha$, the better the guarantee, and that although $J$ is not supermodular in general, there are situations in which its violations of the diminishing returns property are small.

To see when this is the case, observe that (27) is larger when (i) $\sum_{i \in \mathcal{V}_0} r_{i,0}^{-1} \tilde{\boldsymbol{b}}_i \tilde{\boldsymbol{b}}_i^T$ is small and (ii) $\tilde{\boldsymbol{P}}_1^{-1}(\overline{\mathcal{V}}) \approx \tilde{\boldsymbol{P}}_1^{-1}(\emptyset)$. This occurs when the $\operatorname{diag}(r_{i,k}) \gg \boldsymbol{Q}_k$, i.e., when the LQR gives more weight to the input energy than the

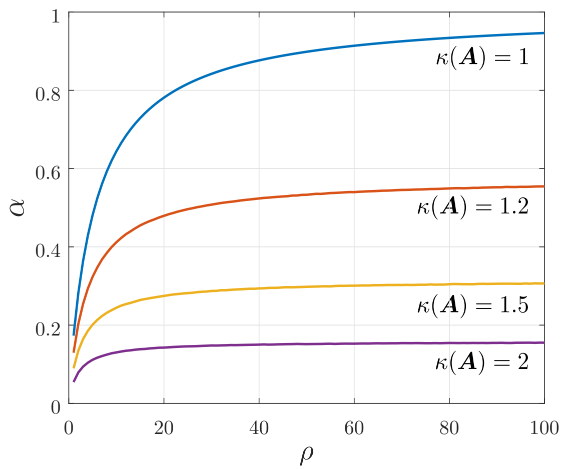


Fig. 1. Bound on $\alpha$ from (25) for a schedule of length $N = 3$ time steps.

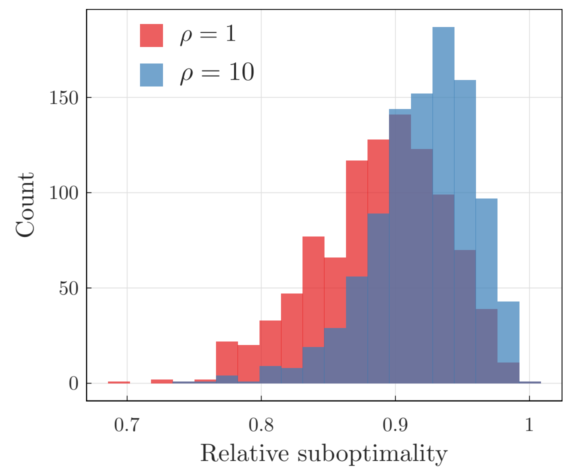


Fig. 2. Relative suboptimality of the greedy solution of (PI) for different $\rho$.

states. This condition is readily obtained from the definition of $\boldsymbol{P}_k$ in (4). It is worth noting that this is a scenario of practical value: in the limit where no restriction is placed on the input energy, any controllable set of actuators can drive the system to any state in a single step (*dead-beat controller*). Careful scheduling of actuators is thus paramount when the cost of driving the inputs is high. This is also the case in which Theorem 2 provides stronger guarantees.

Additionally, note that since all matrices are weighted by $\boldsymbol{W}$, the condition numbers of both $\boldsymbol{A}$ and $\boldsymbol{\Pi}_0$ play an important role on the guarantees. Indeed, if $\boldsymbol{W}$ is poorly conditioned, there may be a large difference between the minimum and maximum eigenvalues in (27). In the sequel, we illustrate typical values of these guarantees and show how different problem parameters affect them.

## V. SIMULATIONS

Consider a system with $p$ inputs available at all steps actuating directly on $p$ states, i.e., $\overline{\mathcal{V}} = \mathcal{V} \times \{0, \dots, N-1\}$ with $\mathcal{V} = \{v_1, \dots, v_p\}$ and $\boldsymbol{b}_{v_i} = \boldsymbol{e}_i$ for $v_i \in \mathcal{V}$, where $\boldsymbol{e}_i$ is the $i$-th vector of the canonical basis. The transition matrix $\boldsymbol{A}$ is a random matrix with prescribed condition number and spectral norm of $1.1$, $\boldsymbol{\Pi}_0 = 10^{-2}\boldsymbol{I}$, $\boldsymbol{Q}_k = \boldsymbol{I}$, and $r_{k,v} = \rho$ for all $k$ and $v \in \mathcal{V}$. Thus, $\rho$ represents the relative importance of the actuation cost in the LQR.

Fig. 1 shows the bound on $\alpha$ in (27) for a system with $p = 100$ inputs and states and a schedule with $N = 4$ steps. Note that the more weight is placed on the controller action cost, i.e., as $\rho$ increases, $\alpha$ grows yielding stronger guarantees. However, the conditioning of $\boldsymbol{A}$ also impacts its value. Indeed, observe that when the decay rate of the system modes are very different, the guarantee from Theorem 2 worsens. Still, recall that this is a worst case guarantee. Hence, as with the classical results for greedy supermodular minimization, better performance is typically obtained in practice, which we illustrate in Fig. 2. Here, we display the relative suboptimality $J(\mathcal{G})/J(\mathcal{S}^\star)$ of 1000 system realizations with random transition matrices $\boldsymbol{A}$ [$\kappa(\boldsymbol{A}) = 1.2$], $p = 5$, $N = 4$, and schedule at most $s_k = 2$ actuators per time step. We compare the results for $\rho = 1$, for which the lower bound on (27) ranged from $0.11$ to $0.16$, and $\rho = 10$, for which the theoretical $\alpha$ bounds was between $0.35$ and $0.53$. Observe that greedy scheduling performed better than predicted by the worst case guarantee in these instances. Nevertheless, there exist dynamical systems for which greedy performs close to the guarantee in Theorem 2. Indeed, consider $\boldsymbol{A} = \boldsymbol{I}$ and

$$\boldsymbol{B} = \begin{bmatrix} 2 & 1 & 0 \\ 2 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

If we schedule up to $s_k = 2$ actuators for all $k$ over a window of $N = 2$ steps with $\rho = 100$, the guarantee in (26) yields $J(\mathcal{G})/J(\mathcal{S}^\star) \geq 0.392$ (recall that $J$ is non-positive) whereas in practice we achieve $J(\mathcal{G})/J(\mathcal{S}^\star) = 0.423$.

## VI. CONCLUSION

This work considered the problem of scheduling actuators to minimize the LQR objective subject to a budget constraint. It showed that the greedy solution of this problem is near-optimal by deriving novel performance certificates for matroid-constrained $\alpha$-supermodular minimization and providing bounds on $\alpha$ for the LQR cost function. In particular, the guarantees obtained are stronger when more weight is given to the control action cost, which was argued to be a scenario of particular interest in practice. These guarantees provide a theoretical justification for the empirical performance of greedy scheduling algorithms, precluding the need for surrogates costs. Future extensions of this work include deriving near-optimal certificates for more intricate constraint structures (e.g., intersection of matroid).

## REFERENCES


[1] B.P. Gerkey and M.J. Matarić, "A formal analysis and taxonomy of task allocation in multi-robot systems," *The Int. J. of Robot. Research*, vol. 23[9], pp. 939–954, 2004.

[2] M. Shamaiah, S. Banerjee, and H. Vikalo, "Greedy sensor selection: Leveraging submodularity," in *Conf. on Decision and Contr.*, 2010, pp. 2572–2577.

[3] T.H. Summers, F.L. Cortesi, and J. Lygeros, "On submodularity and controllability in complex dynamical networks," *IEEE Trans. Contr. Netw. Syst.*, vol. 3[1], pp. 91–101, 2016.

[4] L.F.O. Chamon, G.J. Pappas, and A. Ribeiro, "The mean square error in Kalman filtering sensor selection is approximately supermodular," in *Conf. on Decision and Contr.*, 2017, pp. 343–350.
[5] A. Das and D. Kempe, "Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection," in *Int. Conf. on Mach. Learning*, 2011.
[6] G. Sagnol, "Approximation of a maximum-submodular-coverage problem involving spectral functions, with application to experimental designs," *Discrete Appl. Math.*, vol. 161[1-2], pp. 258–276, 2013.
[7] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies," *J. Mach. Learning Research*, vol. 9, pp. 235–284, 2008.
[8] J. Ranieri, A. Chebira, and M. Vetterli, "Near-optimal sensor placement for linear inverse problems," *IEEE Trans. Signal Process.*, vol. 62[5], pp. 1135–1146, 2014.
[9] V. Tzoumas, M.A. Rahimian, G.J. Pappas, and A. Jadbabaie, "Minimal actuator placement with bounds on control effort," *IEEE Trans. Contr. Netw. Syst.*, vol. 3[1], pp. 67–78, 2016.
[10] H. Zhang, R. Ayoub, and S. Sundaram, "Sensor selection for Kalman filtering of linear dynamical systems: Complexity, limitations and greedy algorithms," *Automatica*, vol. 78, pp. 202–210, 2017.
[11] A. Olshevsky, "On (non)supermodularity of average control energy," 2017, arXiv:1512.00037v4.
[12] J.E. Weimer, B. Sinopoli, and B.H. Krogh, "A relaxation approach to dynamic sensor selection in large-scale wireless networks," in *Int. Conf. on Distributed Computing Syst.*, 2008, pp. 501–506.
[13] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Trans. Signal Process.*, vol. 57[2], pp. 451–462, 2009.
[14] N.K. Dhingra, M.R. Jovanović, and Z.Q. Luo, "An ADMM algorithm for optimal sensor and actuator selection," in *Conf. on Decision and Contr.*, 2014, pp. 4039–4044.
[15] C. Rusu, J. Thompson, and N.M. Robertson, "Sensor scheduling with time, energy, and communication constraints," *IEEE Trans. Signal Process.*, vol. 66[2], pp. 528–539, 2018.
[16] M.P. Vitus, W. Zhang, A. Abate, J. Hu, and C.J. Tomlin, "On efficient sensor scheduling for linear dynamical systems," *Automatica*, vol. 48[10], pp. 2482–2493, 2012.
[17] S.T. Jawaid and S.L. Smith, "Submodularity and greedy algorithms in sensor scheduling for linear dynamical systems," *Automatica*, vol. 61, pp. 282–288, 2015.
[18] P. Singh, M. Chen, L. Carlone, S. Karaman, E. Frazzoli, and D. Hsu, "Supermodular mean squared error minimization for sensor scheduling in optimal Kalman filtering," in *American Contr. Conf.*, 2017, pp. 5787–5794.
[19] A. Jadbabaie, A. Olshevsky, and M. Siami, "Deterministic and randomized actuator scheduling with guaranteed performance bounds," 2018, arXiv:1805.00606v1.
[20] A. Schrijver, *Combinatorial Optimization*, Springer, 2003.
[21] M.L. Fisher, G.L. Nemhauser, and L.A. Wolsey, "An analysis of approximations for maximizing submodular set functions—II," *Mathematical Programming*, vol. 14[1], pp. 73–87, 1978.
[22] G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák, "Maximizing a monotone submodular function subject to a matroid constraint," *SIAM J. on Computing*, vol. 40[6], pp. 1740–1766, 2011.
[23] B. Lehmann, D. Lehmann, and N. Nisan, "Combinatorial auctions with decreasing marginal utilities," *Games and Economic Behavior*, vol. 55[2], pp. 270–296, 2006.
[24] M. Sviridenko, J. Vondrák, and J. Ward, "Optimal approximation for submodular and supermodular optimization with bounded curvature," in *SIAM Symposium on Discrete Algorithms*, 2014, pp. 1134–1148.
[25] L.F.O. Chamon and A. Ribeiro, "Near-optimality of greedy set selection in the sampling of graph signals," in *Global Conf. on Signal and Inform. Process.*, 2016.
[26] D.P. Bertsekas, *Dynamic Programming and Optimal Control - Vol. 1*, Athena Scientific, 2017.
[27] L. Ye, S. Roy, and S. Sundaram, "On the complexity and approximability of optimal sensor selection for Kalman filtering," 2017, arXiv:1711.01920v1.
[28] A. Bian, J.M. Buhmann, A. Krause, and S. Tschiatschek, "Guarantees for greedy maximization of non-submodular functions with applications," in *ICML*, 2017.
[29] A. Krause and D. Golovin, "Submodular function maximization," in *Tractability: Practical Approaches to Hard Problems*. Cambridge University Press, 2014.
[30] F. Bach, "Learning with submodular functions: A convex optimization perspective," *Foundations and Trends in Machine Learning*, vol. 6[2-3], pp. 145–373, 2013.
[31] T. Summers and M. Kamgarpour, "Performance guarantees for greedy maximization of non-submodular set functions in systems and control," 2017, arXiv:1712.04122v1.
[32] O. Karaca and M. Kamgarpour, "Exploiting weak supermodularity for coalition-proof mechanisms," in *Conf. on Decision and Contr.*, 2018, pp. 1118–1123.
[33] G.L. Nemhauser, L.A. Wolsey, and M.L. Fisher, "An analysis of approximations for maximizing submodular set functions—I," *Mathematical Programming*, vol. 14[1], pp. 265–294, 1978.
[34] J. Vondrák, "Optimal approximation for the submodular welfare problem in the value oracle model," in *ACM Symposium on Theory of Computing*, 2008, pp. 67–74.
[35] B. Gharesifard and S.L. Smith, "On distributed submodular maximization with limited information," in *American Contr. Conf.*, 2016, pp. 1048–1053.
[36] R.A. Horn and C.R. Johnson, *Matrix analysis*, Cambridge University Press, 2013.

## APPENDIX

## PROOF OF PROPOSITION 1

*Proof.* This result follows directly from the classical dynamic programming argument for the LQR by considering only the active actuators in $\mathcal{S}$ (see, e.g., [26]). We display the derivations here for ease of reference. Explicitly, we proceed by backward induction, first defining the cost-to-go function

$$V_j(\mathcal{S}) = \min_{\mathcal{U}_j(\mathcal{S})} \mathbb{E}\left[\boldsymbol{x}_N^T \boldsymbol{Q}_N \boldsymbol{x}_N + \sum_{k=j}^{N-1}\left(\boldsymbol{x}_k^T \boldsymbol{Q}_k \boldsymbol{x}_k + \sum_{i \in \mathcal{S}\cap\mathcal{V}_k} r_{i,k} u_{i,k}^2\right)\right], \quad (28)$$

where $\mathcal{U}_j(\mathcal{S}) = \{u_{i,k} | i \in \mathcal{S}\cap\mathcal{V}_k\}_{k=j}^{N-1}$ is the subsequence of control actions from time $j$ to the end of the schedule. Recall that the expected value is taken over the initial state $\boldsymbol{x}_0$. We then postulate that (28) has the form

$$V_j(\mathcal{S}) = \mathbb{E}\left[\boldsymbol{x}_j^T \boldsymbol{P}_j(\mathcal{S}) \boldsymbol{x}_j\right], \quad (29)$$

for some $\boldsymbol{P}_j(\mathcal{S}) \succ 0$ and show that this is indeed the case.

To do so, observe that the base case $j = N$ holds trivially by taking $P_{N-1}(\mathcal{S}) = \boldsymbol{Q}_N \succ 0$. Then, notice that the additivity of (28) allows us to write it as

$$V_j(\mathcal{S}) = \min_{\mathcal{U}_j(\mathcal{S})} \mathbb{E}\left[V_{j+1}(\mathcal{S}) + \boldsymbol{x}_j^T \boldsymbol{Q}_j \boldsymbol{x}_j + \sum_{i \in \mathcal{S}\cap\mathcal{V}_j} r_{i,j} u_{i,j}^2\right].$$

To proceed, assume that (29) holds for $j+1$. Using the system dynamics (1) in (29), we can expand $V_j(\mathcal{S})$ to read

$$V_j(\mathcal{S}) = \min_{\mathcal{U}_j(\mathcal{S})} \mathbb{E}\Bigg[\boldsymbol{x}_j^T\left(\boldsymbol{Q}_j + \boldsymbol{A}^T \boldsymbol{P}_{j+1} \boldsymbol{A}\right)\boldsymbol{x}_j + \left(\sum_{i \in \mathcal{S}\cap\mathcal{V}_j} \boldsymbol{b}_i u_{i,j}\right)^T \boldsymbol{P}_{j+1}(\mathcal{S})\left(\sum_{i \in \mathcal{S}\cap\mathcal{V}_j} \boldsymbol{b}_i u_{i,j}\right) + 2\boldsymbol{x}_j^T \boldsymbol{A}^T \boldsymbol{P}_{j+1}(\mathcal{S})\left(\sum_{i \in \mathcal{S}\cap\mathcal{V}_j} \boldsymbol{b}_i u_{i,j}\right) + \sum_{i \in \mathcal{S}\cap\mathcal{V}_j} r_{i,j} u_{i,j}^2\Bigg]. \quad (30)$$

Finally, notice that (30) is a quadratic optimization problem in the $\{u_{i,j}\}$. This is straightforward to see by rewriting (30) in matrix form:

$$V_j(\mathcal{S}) = \min_{\boldsymbol{u}_j} \mathbb{E}\left[\boldsymbol{x}_j^T\left(\boldsymbol{Q}_j + \boldsymbol{A}^T\boldsymbol{P}_{j+1}\boldsymbol{A}\right)\boldsymbol{x}_j + \boldsymbol{u}_j^T\left(\boldsymbol{R}_j + \boldsymbol{B}_j^T\boldsymbol{P}_{j+1}(\mathcal{S})\boldsymbol{B}_j\right)\boldsymbol{u}_j + 2\boldsymbol{x}_j^T\boldsymbol{A}^T\boldsymbol{P}_{j+1}(\mathcal{S})\boldsymbol{B}_j\boldsymbol{u}_j\right], \tag{31}$$

where $\boldsymbol{u}_j = [u_{i,j}]_{i\in\mathcal{S}\cap\mathcal{V}_j}$ is a $|\mathcal{S}\cap\mathcal{V}_j|\times 1$ vector that collects the control actions, $\boldsymbol{B}_j = [\boldsymbol{b}_i]_{i\in\mathcal{S}\cap\mathcal{V}_j}$ is an $n\times|\mathcal{S}\cap\mathcal{V}_j|$ matrix whose columns contain the input vectors corresponding to each control action, and $\boldsymbol{R}_j = \operatorname{diag}(r_{i,j})$. Since (31) has the form of the classical LQR problem, $V_j$ can be written in closed form as [26]

$$V_j(\mathcal{S}) = \mathbb{E}\left[\boldsymbol{x}_j^T\boldsymbol{P}_j(\mathcal{S})\boldsymbol{x}_j\right]$$

with

$$\boldsymbol{P}_j(\mathcal{S}) = \boldsymbol{Q}_j + \boldsymbol{A}^T\left(\boldsymbol{P}_{j+1}^{-1}(\mathcal{S}) + \sum_{i\in\mathcal{S}\cap\mathcal{V}_j} r_{i,j}^{-1}\boldsymbol{b}_i\boldsymbol{b}_i^T\right)^{-1}\boldsymbol{A},$$

which proves the induction hypothesis in (29) hold. Using the fact that $\mathbb{E}[\boldsymbol{x}_0\boldsymbol{x}_0^T] = \boldsymbol{\Pi}_0$ concludes the proof. ∎

## Proof of Proposition 4

*Proof.* Start by expanding $J$ using (3) to obtain

$$J(\mathcal{S}) = \operatorname{Tr}\left[\boldsymbol{\Pi}_0\boldsymbol{A}^T\left(\boldsymbol{P}_1^{-1}(\mathcal{S}) + \sum_{i\in\mathcal{S}\cap\mathcal{V}_0} r_{i,0}^{-1}\boldsymbol{b}_i\boldsymbol{b}_i^T\right)^{-1}\boldsymbol{A}\right] + \operatorname{Tr}(\boldsymbol{\Pi}_0\boldsymbol{Q}_0) - V(\emptyset). \tag{32}$$

Notice then that the definition of $\alpha$-supermodularity in (6) is invariant to constants [4, Prop. 2]. Hence, it is enough to study the $\alpha$-supermodularity of

$$\bar{J} = \operatorname{Tr}\left[\boldsymbol{W}\left(\boldsymbol{P}_1^{-1}(\mathcal{S}) + \sum_{i\in\mathcal{S}\cap\mathcal{V}_0} r_{i,0}^{-1}\boldsymbol{b}_i\boldsymbol{b}_i^T\right)^{-1}\right], \tag{33}$$

where we used the circular shift property of the trace and wrote $\boldsymbol{W} = \boldsymbol{A}\boldsymbol{\Pi}_0\boldsymbol{A}^T$.

Notice that $\boldsymbol{W}\succ 0$ since $\boldsymbol{\Pi}_0\succ 0$ and $\boldsymbol{A}$ is full rank. It therefore has a unique positive definite square root $\boldsymbol{W}^{1/2}\succ 0$ such that $\boldsymbol{W} = \boldsymbol{W}^{1/2}\boldsymbol{W}^{1/2}$ [36]. Using once again the circular shift property of the trace and the invertibility of $\boldsymbol{W}^{1/2}$, (33) can be written as

$$\bar{J}(\mathcal{S}) = \operatorname{Tr}\left[\left(\tilde{\boldsymbol{P}}_1^{-1}(\mathcal{S}) + \sum_{i\in\mathcal{S}\cap\mathcal{V}_0} r_{i,0}^{-1}\tilde{\boldsymbol{b}}_i\tilde{\boldsymbol{b}}_i^T\right)^{-1}\right], \tag{34}$$

with $\tilde{\boldsymbol{P}}_1(\mathcal{S}) = \boldsymbol{W}^{1/2}\boldsymbol{P}_1(\mathcal{S})\boldsymbol{W}^{1/2}$ and $\tilde{\boldsymbol{b}}_i = \boldsymbol{W}^{-1/2}\boldsymbol{b}_i$. The function $\bar{J}$ in (34) has a form that allows us to use the following result from [4, Thm. 2], which we reproduce here for ease of reference:

**Lemma 2.** *Let $h : 2^{\mathcal{E}}\to\mathbb{R}$ be the set trace function*

$$h(\mathcal{A}) = \operatorname{Tr}\left[\left(\boldsymbol{M}_\emptyset + \sum_{i\in\mathcal{A}}\boldsymbol{M}_i\right)^{-1}\right], \tag{35}$$

*where $\mathcal{A}\subseteq\mathcal{E}$, $\boldsymbol{M}_\emptyset\succ 0$, and $\boldsymbol{M}_i\succeq 0$ for all $i\in\mathcal{E}$. Then, $h$ is (i) monotonically decreasing and (ii) $\alpha$-supermodular with*

$$\alpha \geq \frac{\lambda_{\min}\left[\boldsymbol{M}_\emptyset\right]}{\lambda_{\max}\left[\boldsymbol{M}_\emptyset + \sum_{i\in\mathcal{V}}\boldsymbol{M}_i\right]} > 0. \tag{36}$$

Comparing (34) and (35), we can bound the $\alpha$ for $J$ as

$$\alpha \geq \min_{\mathcal{S}\subseteq\overline{\mathcal{V}}} \frac{\lambda_{\min}\left[\tilde{\boldsymbol{P}}_1^{-1}(\mathcal{S})\right]}{\lambda_{\max}\left[\tilde{\boldsymbol{P}}_1^{-1}(\mathcal{S}) + \sum_{i\in\mathcal{V}_0} r_{i,0}^{-1}\tilde{\boldsymbol{b}}_i\tilde{\boldsymbol{b}}_i^T\right]}. \tag{37}$$

Still, the bound in (37) depends on the choice of $\mathcal{S}$. To obtain a closed-form expression, we use the following proposition:

**Proposition 5.** *For any $\mathcal{S}\subseteq\overline{\mathcal{V}}$ it holds that*

$$\tilde{\boldsymbol{P}}_1(\overline{\mathcal{V}}) \preceq \tilde{\boldsymbol{P}}_1(\mathcal{S}) \preceq \tilde{\boldsymbol{P}}_1(\emptyset). \tag{38}$$

Using (38) in (37) and the fact that matrix inversion is operator antitone yields the bound in (25). ∎

All that remains is therefore to prove Proposition 5.

*Proof.* We prove both inequalities by recursion. For the upper bound in (38), note from (4) that $\boldsymbol{P}_k$ can be increased by using no actuators at instant $k$. Formally, for any choice of $\mathcal{S}\subseteq\overline{\mathcal{V}}$,

$$\begin{aligned}\boldsymbol{P}_k(\mathcal{S}) &= \boldsymbol{Q}_k + \boldsymbol{A}^T\left(\boldsymbol{P}_{k+1}^{-1}(\mathcal{S}) + \sum_{i\in\mathcal{S}\cap\mathcal{V}_k} r_{i,k}^{-1}\boldsymbol{b}_i\boldsymbol{b}_i^T\right)^{-1}\boldsymbol{A} \\ &\preceq \boldsymbol{Q}_k + \boldsymbol{A}^T\boldsymbol{P}_{k+1}(\mathcal{S})\boldsymbol{A} = \boldsymbol{P}_k(\mathcal{S}\setminus\mathcal{V}_k).\end{aligned}$$

Additionally, it holds for any $\bar{\boldsymbol{P}}_{k+1}\succeq\boldsymbol{P}_{k+1}(\mathcal{S})$ for all $\mathcal{S}\subseteq\overline{\mathcal{V}}$ that

$$\boldsymbol{P}_k(\mathcal{S}) \preceq \boldsymbol{Q}_k + \boldsymbol{A}^T\bar{\boldsymbol{P}}_{k+1}\boldsymbol{A} \triangleq \bar{\boldsymbol{P}}_k. \tag{39}$$

Starting from $\boldsymbol{P}_N\preceq\boldsymbol{Q}_N\triangleq\bar{\boldsymbol{P}}_N$, we obtain that $\boldsymbol{P}_1$ is upper bounded by taking $\mathcal{S}=\emptyset$, i.e., using no actuators.

The lower bound is obtained in a similar fashion by using all possible actuators. Explicitly,

$$\begin{aligned}\boldsymbol{P}_k(\mathcal{S}) &= \boldsymbol{Q}_k + \boldsymbol{A}^T\left(\boldsymbol{P}_{k+1}^{-1}(\mathcal{S}) + \sum_{i\in\mathcal{S}\cap\mathcal{V}_k} r_{i,k}^{-1}\boldsymbol{b}_i\boldsymbol{b}_i^T\right)^{-1}\boldsymbol{A} \\ &\succeq \boldsymbol{Q}_k + \boldsymbol{A}^T\left(\boldsymbol{P}_{k+1}^{-1}(\mathcal{S}) + \sum_{i\in\mathcal{V}_k} r_{i,k}^{-1}\boldsymbol{b}_i\boldsymbol{b}_i^T\right)^{-1}\boldsymbol{A} \\ &= \boldsymbol{P}_k(\mathcal{S}\cup\mathcal{V}_k).\end{aligned}$$

Moreover, for any $\underline{\boldsymbol{P}}_{k+1}\preceq\boldsymbol{P}_{k+1}(\mathcal{S})$ for all $\mathcal{S}\subseteq\overline{\mathcal{V}}$, we have

$$\boldsymbol{P}_k \succeq \boldsymbol{Q}_k + \boldsymbol{A}^T\left(\underline{\boldsymbol{P}}_{k+1}^{-1} + \sum_{i\in\mathcal{V}_k} r_{i,k}^{-1}\boldsymbol{b}_i\boldsymbol{b}_i^T\right)^{-1}\boldsymbol{A} \triangleq \underline{\boldsymbol{P}}_k. \tag{40}$$

Starting from $\boldsymbol{P}_N\succeq\boldsymbol{Q}_N\triangleq\underline{\boldsymbol{P}}_N$ yields the lower bound in (38). ∎