Model Predictive Selection:
A Receding Horizon Scheme for Actuator Selection

Vinicius L. Silva, Luiz F. O. Chamon, and Alejandro Ribeiro

***Abstract*— We propose a model predictive scheme for selecting actuators in dynamical systems. In control applications, selection problems arise due to the high cost associated to simultaneously using all sensors or actuators in large-scale systems. Since these problems are NP-hard in general, finding an optimal solutions is impractical and approximations based on greedy or convex relaxations are commonly used. In most approaches, however, the control policy and actuator subsets are obtained *a priori*. In this work, we address the online problem using a *model predictive selection* (MPS). This iterative procedure inspired by model predictive control methods determines a near-optimal actuator subset for a finite operation horizon starting at the current state, applies the first control action on this subset, and repeats the procedure starting from the new state. Despite using suboptimal solutions of the selection problem, we derive conditions that guarantee this procedure is stable. We illustrate these conditions for the LQR problem by leveraging the concept of approximate submodularity and conclude with numerical experiments that showcase the use of the proposed approach.**

I. INTRODUCTION

The study of large-scale systems is inspired by applications in power transmission, transportation systems, and biological networks [1]. Due to the large number of possible actuation points, communication/power constraints, and/or elevated operation costs, simultaneously acting on all the inputs of the system becomes impractical. Thus, the control design process usually includes a step that selects a subset of actuators used to operate the system. This selection is typically performed to optimize some control metric, such as controllability of the linear quadratic regulator (LQR) cost, so as to reduce the complexity or operation costs without hindering the control task. In most approaches, actuator selection is performed offline, before the system begins to operate [2]–[12]. Here, on the other hand, we are interested in solving this problem in an online similar to model-predictive control (MPC).

Model predictive or receding horizon control uses of an online optimization procedure to design the control policy. At each iteration, the agent solves an open-loop optimal control problem starting at the current system state. It then applies the first action of the resulting control sequence and starts again by solving a new optimal control problem starting from the updated state. MPC is an attractive solution to deal with nonlinear system dynamics, incorporate state and input constraints, and account for model uncertainty [13]–[15]. It has been especially successful in the chemical industry [16], though it has found applications in a myriad of domains including transportation systems [17] and power transmission networks [18].

In this work, we borrow the MPC structure to cast the actuator selection problem in a recursive fashion. Namely, at each step the controller seeks an actuators subset that meets a prescribed budget and a sequence of control actions on this subset so as to optimize a control performance metric. The controller then actuate on this subset using the first optimal control action and repeats the procedure in the next time step starting at the updated state. From the viewpoint of selection problems, the main distinction of this approach is the online, recursive nature of this solution that can help mitigate model mismatch issues. On the MPC side, the ability of selecing a subset of actuators may contribute to reducing the operation cost of a large-scale system.

The paper starts by introducing the *model predictive selection* (MPS) problem in Section II. Due to the NP-hardness of these selection problems, solving them exactly online is impractical even for moderately large systems. Nevertheless, near-optimal approximate solutions can be obtained using low-completely methods such as greedy search. To address this issue, we analyze the stability of near-optimal MPS controllers in Section III. We illustrate these results by obtaining stability conditions for actuator selection in the context of LQRs by leveraging the approximate supermodularity of their objectives (Section IV). We conclude by presenting numerical experiments to illustrate the performance of MPS (Section V).

In the following, $\mathbb{R}$ and $\mathbb{N}$ are the set of real and natural numbers. Vectors are indicated by lowercase and matrices by uppercase letters. Positive (semi-) definiteness of a matrix X is indicated by $X > (\geq)\ 0$. Upper-case, caligraphic letters stand for sets and the cardinality of set $\mathcal{X}$ is given by $|\mathcal{X}|$. As usual, $\emptyset$ is the empty set.

A. Related literature

Minimal selection problems for control of large-scale systems consist in finding a minimum number of inputs or outputs to operate the system while satisfying some constraints or performance criteria. Many works on the topic consider the notion of submodular functions, for which there exist near-optimality bounds for solutions obtained with greedy algorithms [19]. In [2], for example, the author studies the minimal placement of actuators so that a linear, time-invariant system is controllable and shows that this is a NP-hard problem which can be approximated via greedy algorithms. In [3] the authors use the smallest eigenvalue

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

of the controllability Gramian matrix to assess the worst-case control energy and derive tradeoffs between the control energy and the number of control inputs, whereas the paper [4] analyzes the structural controllability problem, with results later extended to include output selection [5]. Tzoumas et al discuss in [7] actuator selection problems where the objective is to minimize the control effort. Papers [8], [9] treat actuator selection problems for usual control metrics: the authors show the linear quadratic regulator in deterministic or stochastic settings is neither sub- nor supermodular, but also establish some special conditions under which these problems can be considered modular. In [10] the authors extend the discussion to non-submodular control objectives, including the (negative) trace of the inverse and the minimum eigenvalue of the controllability Gramian. Chamon et al discuss in [20] sensor selection for nonsubmodular objectives, relying on multiplicative and additive relaxations of supermodularity and corresponding approximate optimality guarantees. The paper [12] relies on convex relaxations and also uses the trace of the Gramian controllability matrix as optimization criterion. More recently, paper [11] tackles the actuator selection problem for systems with additive and multiplicative uncertainties and time delays. They show this is equivalent to a selection problem with bounded submodularity ratio, which yields near-optimality guarantees [11], [21]. Another recent work on the topic is [22], where the authors discuss sensor and actuator selection problems in uncertain cyber-physical systems.

Within the context of a model predictive structure for input selection, [23] is perhaps the work with motivation closest to ours. The paper presents a *"Model Predictive Sensor Scheduling"* framework for the online scheduling of sensors for discrete-time, time-varying linear Gaussian systems. The authors consider, among other problems, the use of a sensor at each time instant and evaluate a monotone class of objective functions which includes the trace and maximal eigenvalue of the error covariance matrix. Here, we exploit the combinatorial nature of the selection problem and discuss the stability of approximate solutions for an arbitrary number of inputs.

II. MODEL PREDICTIVE SELECTION

Consider a discrete, time-invariant linear system with state vector $x(t) \in \mathbb{R}^n$ and control inputs $u(t) \in \mathbb{R}^m$ indexed by $\mathcal{V} := \{1, \dots, m\}$. When actuated using the input subset $\mathcal{S} \subseteq \mathcal{V}$, the system evolves according to the difference equation

$$\begin{aligned} x(t+1) &= Ax(t) + B_{\mathcal{S}} u_{\mathcal{S}}(t), \\ y(t) &= Cx(t) + w(t) \end{aligned} \tag{1}$$

where $u_{\mathcal{S}} \in \mathbb{R}^{|\mathcal{S}|}$ collects the control signals for the active inputs, i.e., $u_{\mathcal{S}} = [u_i \, : \, i \in \mathcal{S}]$, and $B_{\mathcal{S}} = [b_i \, : \, i \in \mathcal{S}]$, where $b_i \in \mathbb{R}^n$ is the input vector of actuator $i \in \mathcal{V}$.

Our goal is to select an input subset $\mathcal{S}$ and a sequence of control actions $u(t)$ in an online fashion so as to optimize a control metric for the dynamical system (1) under an actuator budget constraint. To do so, we use an MPS policy in which $\mathcal{S}$ and $u(t)$ are chosen for a finite horizon N, but only the first control action is executed. Explicitly, if the system is at state $x(t)$ at time t, the controller solves the selection problem

$$\begin{aligned} J_N^*(x(t)) = \min_{\boldsymbol{u}_{\mathcal{S}},\, \mathcal{S} \subseteq \mathcal{V}} \quad & J_N(\boldsymbol{\xi}, \boldsymbol{u}_{\mathcal{S}}) \\ \text{subject to} \quad & |\mathcal{S}| \leq k \\ & \xi(\tau+1) = A\xi(\tau) + B_{\mathcal{S}} u_{\mathcal{S}}(\tau), \\ & \tau = 0, \dots, N-1,\, \xi(0) = x(t) \end{aligned} \tag{2}$$

where $k \in \mathbb{N}$ is the maximum number of actuators that can be used at each time and J_N is an N-step control performance metric that depends on $\boldsymbol{\xi} = \{\xi(0), \xi(1), \dots, \xi(N)\}$, the open-loop prediction of the state evolution starting at $\xi(0) = x_t$, and $\boldsymbol{u}_{\mathcal{S}} = \{u_{\mathcal{S}}(0), \dots, u_{\mathcal{S}}(N-1)\}$, the control action sequence. For instance, J_N can trade-off regulation and actuation energy using the LQR objective (as in Section IV). The solution of (2) yields an optimal input subset $\mathcal{S}^*$ and control sequence $\boldsymbol{u}^*_{\mathcal{S}^*}$. The first control action in $\boldsymbol{u}^*_{\mathcal{S}^*}$ is then applied to the system and the procedure is repeated for the resulting $x(t+1)$. An immediate generalization of (2) is to allow the controller to select different actuator subsets at each instant of the horizon, i.e., to replace the selection problem by a scheduling one [24]. We leave this extension for future work.

Notice that at each step the controller must solve a cardinality constrained set function optimization problem to obtain the actuator subset $\mathcal{S}$. Hence, even if J_N is convex in $\boldsymbol{\xi}$ and $\boldsymbol{u}$ (as is the case of the LQR in Section IV), (2) remains a discrete optimization problem. In fact, it is in general NP-hard [2]. Solving problem (2) exactly is therefore infeasible even for moderately large systems, specially in the online setting in which this problem must be solved in the (short) interval between two successive control actions.

Still, it is sometimes possible to efficiently obtain approximate solutions for selection problems. This is the case, for instance, if the control objective J_N is a supermodular function, a form of diminishing return property from which near-optimal guarantees can be obtained for greedy minimization [25]. Though set functions such as the rank or log determinant of the controllability Gramian are supermodular [7], [9], this is a stringent condition that is not met by many cost functions of interest, such as the LQR objective.

In the sequel, we analyze MPS schemes based on near-optimal solutions of (2) and derive conditions to guarantee their stability. We then provide a unified view of these conditions for a myriad of control objectives (e.g., LQR cost function) using the theory of α-supermodular functions.

III. STABILITY OF NEAR-OPTIMAL MPS

As we discussed in the previous section, selection problems such as (2) cannot be solved efficiently in general. Nevertheless, polynomial (sometimes even linear) time approximations that trade-off optimality and computational complexity are often available. In this section, we analyze under which conditions these approximate solutions do not degrade the stability of MPS schemes.

To begin, let us formalize the concept of approximate solution. To do so, consider the generic cardinality constrained minimization problem

$$\begin{aligned} &\text{minimize} && f(\mathcal{X}) \\ &\text{subject to} && |\mathcal{X}| \leq k \end{aligned} \tag{3}$$

where $f : \mathcal{V} \to \mathbb{R}_+$ is a non-negative function. Note that (2) is of the form (3). We are then interested in the following form of suboptimality:

Definition 1 (((ν, μ)-optimal solution)**.** A solution $\mathcal{X}^\nu$ for (3) is said to be (ν, μ)-optimal if

$$f\left(\mathcal{X}^\nu\right) \leq \nu f\left(\mathcal{X}^*\right) + \mu, \tag{4}$$

where $\mathcal{X}^*$ is an optimal solution of (3).

Observe that $\nu \geq 1$ in Definition 1 by optimality. For $\nu = 1$ and $\mu = 0$, $\mathcal{X}^\nu$ is an optimal solution of (3). For $\nu > 1$ or $\mu > 0$, we say $\mathcal{X}^\nu$ is near-optimal.

The following theorem gives conditions under which a near-optimal MPS scheme is stable.

Theorem 1. *Let J_N^* be a Lyapunov function and $u^\nu_{\mathcal{S}^\nu}(x)$ be the first control action determined by a ν-optimal solution with value $J^\nu(x)$ of (2) starting at state x. Then, the resulting near-optimal MPS scheme is stable if for all $x(t) \in \mathbb{R}^n$ and $(x(t), x^\nu(t+1))$ following the dynamics (1) with input $u^\nu_{\mathcal{S}^\nu}(x(t))$ it holds that*

$$J_N^\nu(x(t)) \geq \nu J_N^\nu(x^\nu(t+1)) + \mu. \tag{5}$$

Theorem 1 ties the stability of a near-optimal MPS to the stability of its optimal counterpart, which in turn depends on specific structural properties of the system and cost function. It does so by relating the suboptimality of the solution of problem (2) to the regulation rate of the dynamical system. In a sense, it quantifies the intuition that if the optimal MPS controller is able to quickly regulate the system, then the approximate MPS has a larger “suboptimality margin.” Or alternatively, that when using poor approximate solutions of (2), the suboptimal MPS scheme can only stabilize systems that are “easier” to stabilize. This is in contrast with results for approximate MPC that (essentially) only require feasibility [26]. In Section (IV), we relate this stability result to how many additional inputs a particular class of near-optimal MPS problems needs compared to the optimal one.

Before proceeding with the proof of Theorem 1, it is worth noting that the Lyapunov function assumption of Theorem 1 is fairly mild. Indeed, optimal control schemes often seek to minimize Lyapunov functions of the dynamical system. Moreover, for common choices of J_N, such as the LQR objective studied in Section IV, it can be shown that J_N^* is a Lyapunov function, cf. Theorem 11.2 on [27]. Finally, observe that Theorem 1 can readily be strengthened to show the near-optimal scheme is asymptotically stable by assuming that the corresponding optimal MPS is asymptotically stable and making the inequality in (4) strict.

Proof. Start by recalling that a Lyapunov function is any $V : \mathcal{D} \to \mathbb{R}$ satisfying $V(0) = 0$ and $V(x) > 0$ for all $x \in \mathcal{D} - \{0\}$. Moreover, if $V(x(t+1)) \leq V(x(t))$ for all trajectories $\{x(t)\} \in \mathcal{D}^\infty$ of a dynamical system, then this system is stable [28]. Since J_N^* is a Lyapunov function, proving that the ν-optimal MPS scheme is stable is the same as proving that

$$J_N^*(x(t)) \geq J_N^*(x^\nu(t+1)) \tag{6}$$

for all $x(t) \in \mathbb{R}^n$ and $(x(t), x^\nu(t+1))$ following the dynamics (1) with input $u^\nu_{\mathcal{S}^\nu}(x(t))$.

To do so, start by using the definition of ν-optimality in (4) to get that

$$J_N^*(x(t)) \geq \frac{J_N^\nu(x(t)) - \mu}{\nu},$$

for all $x(t)$, which from (5) readily implies that $J_N^*(x(t)) \geq J^\nu(x^\nu(t+1))$ for all $(x(t), x^\nu(t+1))$ following the dynamics (1) with near-optimal actuation. We can then leverage the optimality of J_N^* to obtain (6). ☐

In a sense, Theorem 1 is a self-fulfilling prophecy: near-optimality ties the value of the suboptimal problem to that of the optimal one, so that if the former decreases enough, the latter will too. A natural follow up question is how condition (5) translates to the structure of particular MPS problem, i.e., how it is related to their cost function or constraints. In the next section, we analyze the stability of MPS problems with α-supermodular objectives, writing the near-optimality condition in Theorem 1 in terms of the actuation budget of the approximate MPS scheme.

IV. α-SUPERMODULAR MPS

Theorem 1 provides stability guarantees in terms of properties of the solution of problem (2). The goal of this section is to translate these results into properties of the optimal control problem itself. To do so, we study MPS schemes whose objectives J_N are α-supermodular.

Supermodularity is a *diminishing returns* property of set functions, i.e., functions $f : 2^\mathcal{V} \to \mathbb{R}$ that map subsets $\mathcal{S} \subseteq \mathcal{V}$ of a ground set $\mathcal{V}$ to real values $f(\mathcal{S})$ [19], [29]. Formally,

Definition 2 (Supermodularity)**.** A function $f : 2^\mathcal{V} \to \mathbb{R}$ is *supermodular* if for every $\mathcal{A} \subset \mathcal{B} \subseteq \mathcal{V}$ and $e \in \mathcal{V} \setminus \mathcal{B}$ it holds that

$$\Delta_e f(\mathcal{A}) \geq \Delta_e f(\mathcal{B}), \tag{7}$$

where $\Delta_e f(\mathcal{S}) \coloneqq f(\mathcal{S}) - f(\mathcal{S} \cup e)$ is the *discrete derivative* of f with respect to e at $\mathcal{S}$.

Function f is said to be submodular if $-f$ is supermodular. Moreover, f is said to be monotone non-increasing if $f(\mathcal{A}) \geq f(\mathcal{B})$ for $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{V}$. Supermodular functions can be near-optimally minimized using greedy search. Explicitly, if f is monotone and submodular, then the value of the greedy solution of (3) is within $(1 - 1/e)$ of the optimal value [25]. Though some control metrics are supermodular, e.g., the log determinant of the controllability Gramian [6], [7], it is a strict condition that does not hold for many common optimal control objectives [2], [8].

To address this issue, different relaxations of (7) have been proposed, such as the submodularity ratio [21] and different notions of curvature [30], and used to obtain guarantees in a myriad of applications [10], [11], [20], [31]. Here, we consider the following definition:

Definition 3 (α-supermodularity)**.** A set function $f : 2^{\mathcal{V}} \to \mathbb{R}$ is *α-supermodular*, for $\alpha \in \mathbb{R}$, if for all sets $\mathcal{A} \subset \mathcal{B} \subseteq \mathcal{V}$ and all $e \in \mathcal{V} \setminus \mathcal{B}$ it holds that

$$\Delta_e f(\mathcal{A}) \geq \alpha \Delta_e f(\mathcal{B}). \tag{8}$$

For $\alpha \geq 1$, (8) reduces to the original definition in (7). For $\alpha < 1$, f is said to be *approximately supermodular*. We are interested in the largest α for which (8) holds, i.e.,

$$\alpha = \min_{\substack{\mathcal{A} \subset \mathcal{B} \subseteq \mathcal{V} \\ e \in \mathcal{V} \setminus \mathcal{B}}} \frac{\Delta_e f(\mathcal{A})}{\Delta_e f(\mathcal{B})}. \tag{9}$$

The concept of α-supermodularity first appeared in the context of auction design [32], though it has since been used in discrete optimization, estimation, and control [20], [30], [33], [34]. The idea is that if a function is close to supermodular ($\alpha \approx 1$), then its properties should not differ too much from a truly supermodular function. The following theorem confirms this intuition.

Theorem 2. *Let $\mathcal{X}^*$ be an optimal solution of problem (3) and $\mathcal{G}_q$ be the q-th iteration of its greedy solution, obtained by taking $\mathcal{G}_0 = \emptyset$ and repeating for $j = 1, \dots, q$,*

$$\begin{aligned} w &= \arg\min_{v \in \mathcal{V}} f(\mathcal{G}_{j-1} \cup \{v\}), \\ \mathcal{G}_j &= \mathcal{G}_{j-1} \cup \{w\} \text{ and } \mathcal{V} = \mathcal{V} \setminus \{w\}. \end{aligned} \tag{10}$$

If f is (i) monotone non-increasing and (ii) α-supermodular, then

$$\frac{f(\mathcal{G}_q) - f(\mathcal{X}^*)}{f(\emptyset) - f(\mathcal{X}^*)} \leq \left(1 - \frac{\alpha}{s}\right)^q \leq e^{-\alpha q/s}. \tag{11}$$

Proof. See, e.g., [20]. □

Observe from (8) and (11) that α measures both how much f violates the supermodular property and the corresponding loss in performance guarantee. Indeed, the e^{-1} approximation factor obtained for supermodular functions when the greedy solution has cardinality $q = s$ now decreases to $e^{-\alpha}$. Moreover, notice that the guarantee in Theorem 2 can be written in the form of Definition 1, so that we immediately obtain from Theorem 1 that

Corollary 1. *Suppose the optimal value J_N^* of problem (2) with cardinality constraint k is a Lyapunov function. If J_N induces an α-supermodular set function, then the MPS scheme obtained by solving (2) greedily is stable if the actuation budget at time t is increased to*

$$\bar{k}_t \geq \frac{k}{\alpha} \ln \left(\frac{J_N^{\emptyset}(x(t)) - J_N^g(x(t+1))}{J_N^g(x(t)) - J_N^g(x(t+1))} \right), \tag{12}$$

where $J_N^{\emptyset}(x)$ is the value of (2) starting at state x with $k = 0$, i.e., using no actuators, and $J_N^g(x)$ is the value of the greedy solution of (2) starting at state x with cardinality $\bar{k}_t$.

Proof. First notice that (11) can be rearranged to read

$$f(\mathcal{G}_q) \leq (1 - e^{-\alpha q/s}) f(\mathcal{X}^*) + e^{-\alpha q/s} f(\emptyset),$$

which has the form of (4) in Definition 1. Hence, if the cost function of (2) is α-supermodular, we can use Theorem 1 to show that stability of the greedy MPS scheme is guaranteed if

$$J_N^g(x(t)) \geq (1 - e^{-\alpha \bar{k}_t/k}) J_N^g(x(t+1)) + e^{-\alpha \bar{k}_t/k} J_N^{\emptyset}(x(t)), \tag{13}$$

where $J_N^g(x)$ and $J_N^{\emptyset}(x)$ are the values of the cardinality $\bar{k}_t$ greedy solution and the empty solution ($k = 1$) of (2) respectively. Inequality (13) readily yields (12). □

Corollary (1) establishes a relation between the α-supermodularity of the control performance metric and the actuation budget required to guarantee the stability of the greedy MPS scheme. Explicitly, (13) states that, up to a logarithmic factor, we need to increase the number of actuators by a factor of $1/\alpha$ to ensure the greedy solution of problem (2) does not affect the stability of MPS. In this sense, $1/\alpha$ quantifies the additional actuation cost to pay for not solving problem (2) exactly. Naturally, what we trade-off here is actuation cost for computational tractability. In the sequel, we further particularize this result for the common case in which MPS is performed using a linear quadratic regulator (LQR).

A. Linear quadratic MPS

When employing an LQR for MPS, the cost function of (2) becomes

$$\begin{aligned} J_N(\boldsymbol{\xi}, \boldsymbol{u}_{\mathcal{S}}) = \mathbb{E} \Bigg[& \xi(N)^\intercal Q_N \xi(N) \\ & + \sum_{i=0}^{N-1} \Big(\xi(i)^\intercal Q \xi(i) + u_{\mathcal{S}}(i)^\intercal R_{\mathcal{S}} u_{\mathcal{S}}(i) \Big) \Bigg] \end{aligned} \tag{14}$$

for $Q \succeq 0$ and $R \coloneqq \operatorname{diag}[r_1, r_2, \dots, r_m] \succ 0$. Recall that, in this case, the optimal value J_N^* of (2) is indeed a Lyapunov function [27, Thm. 11.2]. To show this function is α-supermodular and bound the value of α, we use the same technique used in [20] to show the mean square error for sensor selection in Kalman filtering problems is approximately supermodular.

Proposition 1. *The quadratic cost (14) is α-supermodular in $\mathcal{S}$ with*

$$\alpha \geq \frac{\lambda_{\min}\left[\tilde{P}_1^{-1}(\emptyset)\right]}{\lambda_{\max}\left[\tilde{P}_1^{-1}(\overline{\mathcal{V}}) + \tilde{B}_{\mathcal{S}}^\intercal R_{\mathcal{S}}^{-1} \tilde{B}_{\mathcal{S}}\right]}, \tag{15}$$

with $\tilde{P}_1(\mathcal{S}) = W^{1/2} P_1(\mathcal{S}) W^{1/2}$ and $\tilde{B}_{\mathcal{S}} = W^{1/2} B_{\mathcal{S}}$, where $W = A \Pi_0 A^\intercal$ and $P_j(\mathcal{S})$ is the $j-$th solution of the backward Riccati recursion

$$\begin{aligned} P_{j-1}(\mathcal{S}) = A^\intercal \big[& P_j(\mathcal{S}) - P_j(\mathcal{S}) B_{\mathcal{S}} \\ & (B_{\mathcal{S}}^\intercal P_j(\mathcal{S}) B_{\mathcal{S}} + R_{\mathcal{S}})^{-1} B_{\mathcal{S}}^\intercal P_j(\mathcal{S}) \big] A + Q, \end{aligned} \tag{16}$$

starting from $P_N = Q$.

Proof. See appendix. □

Fig. 1. Active control inputs.

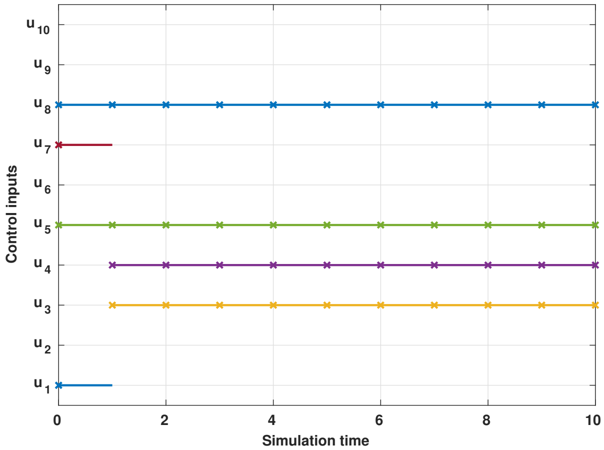
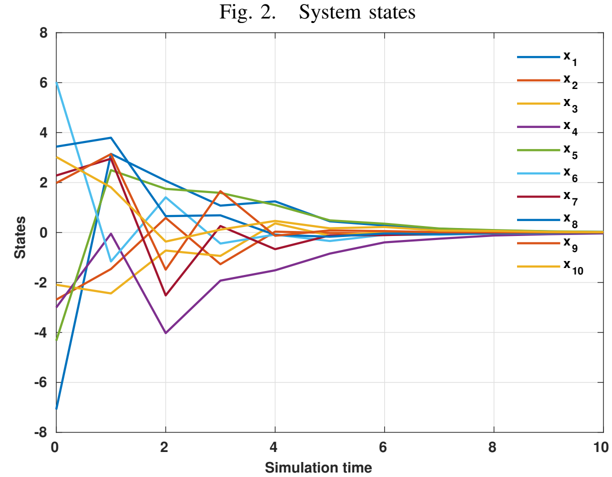


Fig. 2. System states



Algorithm 1: MPS with quadratic cost and greedy search

Required: System and cost matrices A, B, Q, R; optimization horizon N; actuator budget k; ground set $\mathcal{V} = [m]$.
Result: Actuator subset and control action at time t.

while *MPS actuating* **do**
  /* retrieve current state */
  $y(t) \leftarrow Cx(t) + w(t)$
  /* find actuator subset */
  Procedure `greedy search()`:
    for $j = 1, ..., q$ **do**
      $w \leftarrow \arg\min_{v \in \mathcal{V}} f(\mathcal{G}_{j-1} \cup v)$
      $\mathcal{G}_j \leftarrow \mathcal{G}_{j-1} \cup w$
      $\mathcal{V} \leftarrow \mathcal{V} \setminus w$
    end
  /* compute control action */
  RDE recursion (16)
  $K^t = (R_{\mathcal{S}^*} + B_{\mathcal{S}^*}^{\intercal} P_{t+1} B_{\mathcal{S}^*})^{-1} B_{\mathcal{S}^*}^{\intercal} P_{t+1} A$
  /* update system */
  $u_{\mathcal{S}^*}^0 \leftarrow -K^0 x(t)$
  $x(t+1) \leftarrow Ax(t) + B_{\mathcal{S}} u_{\mathcal{S}^*}^0(t)$
end

Using the bound in Proposition 1 together with Corollary 1, we can determine the number of actuators necessary to guarantee that greedy linear quadratic MPS (Algorithm 1) is stable. Naturally, other cost functions related to (14) also lead to stable MPS schemes. For instance, note that if we let $N \to \infty$ in (14), the linear quadratic cost is equivalent to the $\mathcal{H}_2$ norm of the system, which in turn corresponds to the trace of a weighted gramian, a submodular function [6]. In this case, $\alpha \geq 1$, so that the number of actuator needed to stabilize MPS is essentially the same as if (2) was solved exactly. This also occurs, for instance, when using the $\log\det$ of the controllability Gramian. Conic combination of α-supermodular metrics can also be used, since they are also approximately supermodular [20].

V. Numerical experiments

We next devise a few numerical experiments to discuss the use of the MPS framework. We start with an illustration of the use of the MPS scheme to select a subset of actuators to control a linear discrete-time system. The system matrices are randomly chosen and system dimensions satisfy $n = 10$, $m = 10$, with the maximum number of active inputs given by $k = 4$. We adopt a quadratic cost structure with $Q = \mathbb{I}_n$, $R = \mathbb{I}_m$ and optimization horizon $N = 30$. To update the MPS scheme (using a time step of $1s$), we use the covariance matrix of a noisy observation process as covariance of the random "initial" state x_0 at each instance of the control/selection problem — recall that since the matrices are time-invariant, we can use x_0 to represent the starting point $x(t)$ of the control problem at event (x, t). We show some results in Figures 1 — 2. The plots show the active inputs during a given time frame and the corresponding evolution of the states. It is interesting to note that the offline greedy selection algorithm and the MPS scheme start from the same actuator set, but that changes from the second time step, reflecting a difference in the cost (17) due to a different starting point for the following iteration of the MPS problem.

We next evaluate the relative suboptimality of the MPS scheme, since the relatively low dimension of the system allows us to find the optimal solutions and compare them to the greedy ones. We ran 1000 realizations of randomly generated discrete-time systems with $n = 10$ and $m = 10$. We adopt the same cost structure as in the last case. Results are shown in Figures 3 and 4. The first one of them shows a histogram of the relative suboptimality of the MPS greedy solution with relation to the optimal one at the first iteration, and the second one shows a similar histogram after 5 iterations. Even though the greedy algorithm provides good results already in the first iteration, we note that the performance of the MPS scheme in this case improves after a few iterations. Moreover, note that the relative suboptimality shown in the plots corresponds to the near-optimality gap ν from Definition 1 and Proposition 1. For the quadratic case, the greedy algorithm provides, at each instance of the selection problem, a solution very close to the optimal one, which provides stability for the overall scheme.

We also compare the use of the MPS scheme with an offline greedy selection algorithm with the same actuator budget $k = 4$. The same linear quadratic structure is used, but note that in the selection case the objective is to find a single actuator subset to operate the system in the next N time instants. The simulation follows the same structure as in the last case. We show the simulation results in Table I. The table brings the average quadratic cost (14) for 1000 realizations of a discrete linear system for different optimization horizons $N = 10, 20$ or 30. The MPS approach offers a lower cost overall, which is expected from the change in the subset of active inputs.

TABLE I
QUADRATIC COST: MPS AND OFFLINE SELECTION

	Average over 1000 realizations	
Optimization horizon	MPS	Offline selection
$N = 10$	269.0075	273.1953
$N = 20$	269.6135	272.3935
$N = 30$	269.6137	287.6110

VI. CONCLUSION

In this paper we present a novel framework for input selection in large-scale systems: instead of defining *a priori* a subset of control inputs to operate the system, we borrow from the model predictive control approach a recursive structure for the input selection problem. Since selection problems are in general NP-hard, it is impractical to use optimal selection in a recursive manner, and thus we next discussed the stability of near-optimal solutions, that is, approximate solutions which are within a bound of the optimal one. We showed how the sequence of near-optimal solutions remains stable if the approximation errors at each time step are not too large. For the commonly used linear quadratic structure, we showed that each instance of this sequence of selection problems is approximately supermodular, a property which allows us to find near-optimal solutions in a timely fashion. More extensive numerical experiments are needed in order to compare the proposed framework with the use of offline selection and scheduling algorithms, but the initial experiments show the MPS scheme can achieve good performance.

A first extension of the initial results presented here concerns the optimization metric: the approximate supermodularity of the linear quadratic cost structure yields near-optimal solutions in polynomial time, but we still need to investigate if similar properties hold for other metrics commonly used in control. On the model predictive control side, one of the main advantages of MPC is its ability to handle state and control constraints [14] , and incorporating these notions into the MPS framework might make it more attractive to MPC practicioners.

REFERENCES

[1] Mark Newman, *Networks: An Introduction*, Oxford University Press, New York, NY, USA, 2010.

[2] A. Olshevsky, “Minimal Controllability Problems,” *IEEE Transactions on Control of Network Systems*, vol. 1, no. 3, pp. 249–258, Sept. 2014.

[3] F. Pasqualetti, S. Zampieri, and F. Bullo, “Controllability Metrics, Limitations and Algorithms for Complex Networks,” *IEEE Transactions on Control of Network Systems*, vol. 1, no. 1, pp. 40–52, Mar. 2014.

[4] Sérgio Pequito, Soummya Kar, and A. Pedro Aguiar, “On the complexity of the constrained input selection problem for structural linear systems,” *Automatica*, vol. 62, pp. 193 – 199, 2015.

[5] S. Pequito, S. Kar, and A. P. Aguiar, “A Framework for Structural Input/Output and Control Configuration Selection in Large-Scale Systems,” *IEEE Transactions on Automatic Control*, vol. 61, no. 2, pp. 303–318, Feb. 2016.

[6] T. H. Summers, F. L. Cortesi, and J. Lygeros, “On Submodularity and Controllability in Complex Dynamical Networks,” *IEEE Transactions on Control of Network Systems*, vol. 3, no. 1, pp. 91–101, Mar. 2016.

[7] V. Tzoumas, M. A. Rahimian, G. J. Pappas, and A. Jadbabaie, “Minimal actuator placement with bounds on control effort,” *IEEE Transactions on Control of Network Systems*, vol. 3, no. 1, pp. 67–78, March 2016.

[8] T. Summers, “Actuator placement in networks using optimal control performance metrics,” in *2016 IEEE 55th Conference on Decision and Control (CDC)*, Dec 2016, pp. 2703–2708.

[9] T. Summers and J. Ruths, “Performance bounds for optimal feedback control in networks,” *ArXiv e-prints*, July 2017.

[10] T. Summers and Maryam Kamgarpour, “Performance guarantees for greedy maximization of non-submodular controllability metrics,” *ArXiv e-prints*, Dec. 2017.

[11] Z. Liu, Y. Long, A. Clark, L. Bushnell, D. Kirschen, and R. Poovendran, “Minimal input and output selection for stability of systems with uncertainties,” *IEEE Transactions on Automatic Control*, pp. 1–1, 2018.

[12] L. Romao, K. Margellos, and A. Papachristodoulou, “Distributed actuator selection: Achieving optimality via a primal-dual algorithm,” *IEEE Control Systems Letters*, vol. 2, no. 4, pp. 779–784, Oct 2018.

[13] Manfred Morari and Jay H. Lee, “Model predictive control: past, present and future,” *Computers & Chemical Engineering*, vol. 23, no. 4, pp. 667 – 682, 1999.

[14] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 36, no. 6, pp. 789–814, June 2000.

[15] David Q. Mayne, “Model predictive control: Recent developments and future promise,” *Automatica*, vol. 50, no. 12, pp. 2967 – 2986, 2014.

[16] S. Joe Qin and Thomas A. Badgwell, “A survey of industrial model predictive control technology,” *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, July 2003.

[17] R.R. Negenborn, B. De Schutter, and J. Hellendoorn, “Multi-agent model predictive control for transportation networks: Serial versus parallel schemes,” *Engineering Applications of Artificial Intelligence*, vol. 21, no. 3, pp. 353 – 366, 2008.

[18] Ralph M. Hermans, Andrej Jokić, Mircea Lazar, Alessandro Alessio, Paul P.J. van den Bosch, Ian A. Hiskens, and Alberto Bemporad, “Assessment of non-centralised model predictive control techniques for electrical power networks,” *International Journal of Control*, vol. 85, no. 8, pp. 1162–1177, 2012.

[19] Andreas Krause and D. Golovin, “Submodular function maximization,” *Tractability: Practical Approaches to Hard Problems*, vol. 3, pp. Tractability: Practical Approaches to Hard Problems, 2012, Vol.3, 2012.

[20] L. F. O. Chamon, G. J. Pappas, and A. Ribeiro, “The mean square error in Kalman filtering sensor selection is approximately supermodular,” in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Dec 2017, pp. 343–350.

[21] Abhimanyu Das and David Kempe, “Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection,” in *Proceedings of the 28th International Conference on International Conference on Machine Learning*, USA, 2011, ICML’11, pp. 1057–1064, Omnipress.

[22] A. Taha, N. Gatsis, T. H. Summers, and S. A. Nugroho, “Time-varying sensor and actuator selection for uncertain cyber-physical systems,” *IEEE Transactions on Control of Network Systems*, pp. 1–1, 2018, Early access.

[23] Eriko Iwasa and Kenko Uchida, “Model predictive sensor scheduling,” *SICE Journal of Control, Measurement, and System Integration*, vol. 2, no. 3, pp. 139–145, 2009.

Fig. 3. Relative suboptimality at $t_{\text{sim}} = 1$.

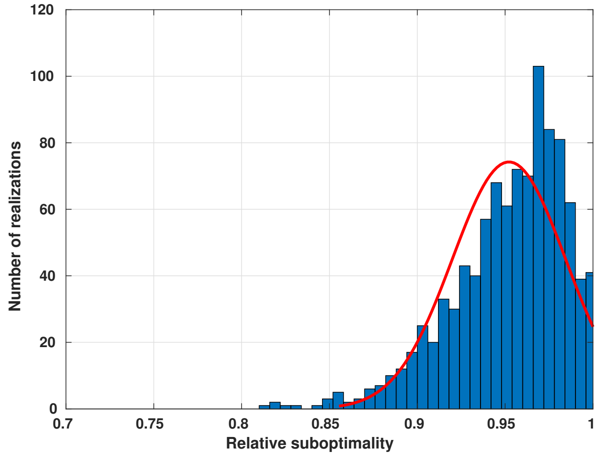
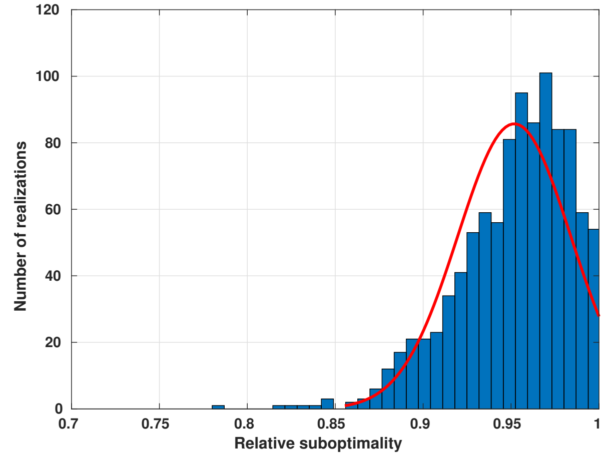


Fig. 4. Relative suboptimality at $t_{\text{sim}} = 5$



[24] Syed Talha Jawaid and Stephen L. Smith, "Submodularity and greedy algorithms in sensor scheduling for linear dynamical systems," *Automatica*, vol. 61, pp. 282 – 288, 2015.

[25] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions—i," *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, Dec 1978.

[26] P. O. M. Scokaert, D. Q. Mayne, and J. B. Rawlings, "Suboptimal model predictive control (feasibility implies stability)," *IEEE Transactions on Automatic Control*, vol. 44, no. 3, pp. 648–654, March 1999.

[27] Francesco Borrelli, Alberto Bemporad, and Manfred Morari, *Predictive Control for Linear and Hybrid Systems*, Cambridge University Press, 2017.

[28] Nicoletta Bof, Ruggero Carli, and Luca Schenato, "Lyapunov Theory for Discrete Time Systems," *arXiv e-prints*, Sep 2018.

[29] Satoru Fujishige, *Submodular Functions and Optimization*, vol. 58 of *Annals of Discrete Mathematics*, Elsevier, 2005.

[30] Maxim Sviridenko, Jan Vondrák, and Justin Ward, "Optimal approximation for submodular and supermodular optimization with bounded curvature," *Mathematics of Operations Research*, vol. 42, no. 4, pp. 1197–1218, 2017.

[31] Zengfu Wang, Bill Moran, Xuezhi Wang, and Quan Pan, "Approximation for maximizing monotone non-decreasing set functions with a greedy method," *Journal of Combinatorial Optimization*, vol. 31, no. 1, pp. 29–43, Jan 2016.

[32] Benny Lehmann, Daniel Lehmann, and Noam Nisan, "Combinatorial auctions with decreasing marginal utilities," in *Proceedings of the 3rd ACM Conference on Electronic Commerce*, New York, NY, USA, 2001, EC '01, pp. 18–28, ACM.

[33] L. F. O. Chamon and A. Ribeiro, "Near-optimality of greedy set selection in the sampling of graph signals," in *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Dec 2016, pp. 1265–1269.

[34] Andrew An Bian, Joachim M. Buhmann, Andreas Krause, and Sebastian Tschiatschek, "Guarantees for greedy maximization of non-submodular functions with applications," in *Proceedings of the 34th International Conference on Machine Learning*, Doina Precup and Yee Whye Teh, Eds., International Convention Centre, Sydney, Australia, 06–11 Aug 2017, vol. 70 of *Proceedings of Machine Learning Research*, pp. 498–507, PMLR.

Appendix

A. α-supermodularity of the LQ MPS problem

Proof. The value function of a linear quadratic optimal control problem is given by $J(\cdot) = \operatorname{tr}(x_0^\intercal P_0 x_0)$, where P_0 is the matrix obtained from a sequence of Riccati difference equations (RDE). For system (1) controlled by an actuator set $\mathcal{S}$ the value function reduces to

$$J(\mathcal{S}) = \operatorname{tr}\left(x_0^\intercal P_0(\mathcal{S}) x_0\right), \tag{17}$$

with $P(\mathcal{S})$ given by the backward recursion of RDEs

$$\begin{aligned} P_{j-1}(\mathcal{S}) = A^\intercal \big[& P_j(\mathcal{S}) - P_j(\mathcal{S}) B_{\mathcal{S}} \\ & (B_{\mathcal{S}}^\intercal P_j(\mathcal{S}) B_{\mathcal{S}} + R_{\mathcal{S}})^{-1} B_{\mathcal{S}}^\intercal P_j(\mathcal{S}) \big] A + Q \end{aligned} \tag{18}$$

starting from $P_N = Q$. Here $R_{\mathcal{S}}$ is the $|\mathcal{S}| \times |\mathcal{S}|$ diagonal matrix made up by elements r_i of R (14) corresponding to the active inputs $i \in \mathcal{S}$. We apply the matrix inversion lemma to equation (18) to obtain

$$P_{j-1}(\mathcal{S}) = Q + A^\intercal \left[P_j^{-1}(S) + B_{\mathcal{S}} (R_{\mathcal{S}})^{-1} B_{\mathcal{S}}^\intercal \right]^{-1} A. \tag{19}$$

Considering the variance of the initial state, $\mathbb{E}\left[x_0^\intercal x_0\right] = \Pi_0$, the optimal cost in (17) can be rewritten as

$$\begin{aligned} J(\mathcal{S}) &= \operatorname{tr}(\Pi_0 Q) \\ &\quad + \operatorname{tr}\left(A \Pi_0 A^\intercal (P_1^{-1}(\mathcal{S})) + B_{\mathcal{S}} (R_{\mathcal{S}})^{-1} B_{\mathcal{S}}^\intercal)^{-1} \right). \end{aligned} \tag{20}$$

From Proposition 2 in [20], the sum of an α-supermodular function and a constant is still α-supermodular, and thus it suffices to show approximate supermodularity of the second term in the right hand side of (20). To do that, let $W = A\Pi_0 A^\intercal \geq 0$ and take $W = W^{1/2} W^{1/2}$ to write (20) as

$$J(\mathcal{S}) = \operatorname{tr}(\cdot) + \operatorname{tr}\left(\left(\tilde{P}_1^{-1}(\mathcal{S}) + \tilde{B}_{\mathcal{S}} R_{\mathcal{S}}^{-1} \tilde{B}_{\mathcal{S}}^\intercal) \right)^{-1} \right), \tag{21}$$

with $\tilde{P}(\mathcal{S}) = W^{1/2} P(\cdot) W^{1/2}$ and $\tilde{B}_{\mathcal{S}} = W^{-1/2} B_{\mathcal{S}}$. Note that due to the diagonal form of R, $\tilde{B}_{\mathcal{S}} R_{\mathcal{S}}^{-1} \tilde{B}_{\mathcal{S}}^\intercal$ is equivalent to $\sum_{i \in \mathcal{S}} r_i^{-1} \tilde{b}_i \tilde{b}_i^\intercal$. Then we can use Theorem 2 in [20] to conclude that (21) is α-supermodular with

$$\alpha \geq \min_{\mathcal{S} \subseteq \overline{V}} \frac{\lambda_{\min}\left[\tilde{P}_1^{-1}(\mathcal{S}) \right]}{\lambda_{\max}\left[\tilde{P}_1^{-1}(\mathcal{S}) + \tilde{B}_{\mathcal{S}} R_{\mathcal{S}}^{-1} \tilde{B}_{\mathcal{S}}^\intercal \right]}. \tag{22}$$

Finally, $\tilde{P}_1(\mathcal{X})$ is monotonically decreasing in the sense that $\mathcal{X} \subseteq \mathcal{Y}$ implies $\tilde{P}_1(\mathcal{Y}) \leq \tilde{P}_1(\mathcal{X})$. Together with the antitone property of the matrix inversion operator this yields the desired result. □