Actuator Scheduling for Linear Systems: A Convex Relaxation Approach

Junjie Jiao, *Member, IEEE*, Dipankar Maity, *Senior Member, IEEE*, John S. Baras, *Life Fellow, IEEE*, and Sandra Hirche, *Fellow, IEEE*

***Abstract*— In this letter, we investigate the problem of actuator scheduling for networked control systems. Given a stochastic linear system with a number of actuators, we consider the case that one actuator is activated at each time. This problem is combinatorial in nature and NP hard to solve. We propose a convex relaxation to the actuator scheduling problem, and use its solution as a *reference* to design an algorithm for solving the original scheduling problem. Using dynamic programming arguments, we provide a suboptimality bound of our proposed algorithm. Furthermore, we show that our framework can be extended to incorporate multiple actuator scheduling at each time and actuation costs. A simulation example is provided, which shows that our proposed method outperforms a random selection approach and a greedy selection approach.**

***Index Terms*— Actuator scheduling, LQG control.**

I. INTRODUCTION

In recent years, networked control systems (NCSs) have gained much interest in the controls community due to the advancements in communication architecture, computer technology, and network infrastructure that enable efficient distributed sensing, estimation, and control [1]–[3]. Due to potential constraints on the communication and computation resources of NCSs, sensor scheduling and actuator scheduling are two important and challenging problems, and efficient algorithms are sought for solving them.

The majority of the existing work focuses on sensor scheduling problems and their variants. Several approaches (e.g., stochastic selection [4], search tree pruning [5], greedy selection [6], semidefinite programming based trajectory tracking [7]) have been proposed to solve such problems. In contrast, the actuator scheduling problem has received much lesser attention. While sensor scheduling problems focus on minimizing (a function of) the estimation error, the actuator scheduling directly affects the controllability and stability of the system as well as the control performance. Therefore, a significant portion of the work on actuator scheduling focuses on studying the effects of actuator scheduling on the controllability and stability of the systems, e.g., [8]–[12] and others. It is shown in [8]–[10] that several classes of energy related metrics associated with the controllability Gramian have a structural property (*modularity*) that allows for an approximation guarantee by using a simple greedy heuristic. These problems are further investigated in [11], where a framework of sparse actuator schedule design was developed that guarantees performance bounds for a class of controllability metrics. Except [12], these works assume a time invariant scheduling problem, which is likely to be suboptimal and may impose restrictions on controllability for large systems. [12] uses a round robin scheme for selecting the actuators and show that local stability is attained if the switching between the actuators is fast enough. The efficacy of time-varying scheduling over time-invariant ones for interconnected systems is also demonstrated in [13]. However, how to find the optimal time-varying schedules remains unanswered.

The efficacy of the abovementioned controllers on a system with different performance criteria (e.g., quadratic cost) is unknown and likely to be suboptimal since these works solely focus on the controllability/stability aspect. In contrast to those works, a few existing works [14]–[16] consider a linear-quadratic optimal control problem for actuator scheduling. However, the focus on these works is to decide at each time whether to activate the *only* actuator available or not.

Motivated by the above, in this letter we study the actuator scheduling problem for a finite horizon linear-quadratic control system with a number of actuators. We consider the case that a nonempty subset of the actuators is active at each time. The performance of the actuator schedule is measured by a finite horizon quadratic cost function of the system state and control plus the cost of using each actuator (representing e.g., energy consumption). This problem is combinatorial in nature and is NP-hard in general. Due to space limitations, we first restrict ourselves to the case that only one actuator is activated at each time and that all actuators have equal actuation costs. We then provide discussions and simulation results on the general cases that multiple actuators are activated at each time and that the actuators have different actuation costs.

The work of J. Jiao was supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement no. 899987. The work of D. Maity was supported by The University of North Carolina at Charlotte. The work of J. S. Baras was partially supported by a DFG Mercator Fellowship within the DFG Priority Program SPP 1914 - Cyber-Physical Networking, by ONR grant N00014-17-1-2622, and by NSF grant ECCS2127605. The work of S. Hirche was supported by the DFG Priority Program SPP 1914 - Cyber-Physical Networking.

J. Jiao and S. Hirche are with the Chair of Information-oriented Control, TUM School of Computation, Information and Technology, Technical University of Munich, Munich, Germany. Email: junjie.jiao@tum.de; hirche@tum.de

D. Maity is with the Department of Electrical and Computer Engineering, The University of North Carolina at Charlotte, NC, 28223, USA. Email: dmaity@uncc.edu

J. S. Baras is with the Department of Electrical and Computer Engineering, University of Maryland at College Park, MD, 20742, USA. Email: baras@umd.edu

The main contributions of this letter are the following: (i) We propose a convex relaxation to the actuator scheduling problem, and use its solution as a 'reference' to design an algorithm for solving the original NP-hard scheduling problem. (ii) We provide a suboptimality bound for the proposed algorithm. (iii) We further show that our results can be extended to the cases with multiple actuator scheduling and actuation costs.

The outline of this letter is as follows: In Section II, we formulate the actuator scheduling problem, which is solved in Section III. In Section IV, we provide discussions on multiple actuator scheduling and actuation costs. Simulation results are provided in Section V. Section VI concludes this letter.

Notation: We denote the set of real numbers and positive real numbers by $\mathbb{R}$ and $\mathbb{R}_+$, respectively. The set of n dimensional vectors over $\mathbb{R}$ is denoted by $\mathbb{R}^n$ and the set of real $n\times m$ matrices by $\mathbb{R}^{n\times m}$. The identity matrix is denoted by I. For a given matrix A, its transpose and inverse (if exists) are denoted by $A^\top$ and A^{-1}, respectively. For a symmetric matrix P, we denote $P \succ 0$ ($P \succeq 0$) if it is positive definite (positive semidefinite). The trace of a square matrix A is denoted by $\mathrm{tr}(A)$ and the Frobenious norm by $\|A\|_F$. We use $\mathbb{E}[x]$ to denote the expectation of a random variable x.

II. Problem Formulation

We consider a system with N actuators of the form

$$x_{t+1} = A_t x_t + \textstyle\sum_{j\in\sigma_t} B_t(j)u_t(j) + w_t, \tag{1}$$

where $A_t \in \mathbb{R}^{n\times n}$, $B_t(j) \in \mathbb{R}^{n\times m_j}$, $\sigma_t \subseteq \mathsf{N} := \{1, 2, \ldots, N\}$ the set of selected actuators, $x_t \in \mathbb{R}^n$ the state, $u_t(j) \in \mathbb{R}^{m_j}$ the input from the j-th actuator, and $w_t \in \mathbb{R}^n$ an independent sequence of Gaussian random variables with $w_t \sim \mathcal{N}(0, W_t)$. The initial state is $x_0 \sim \mathcal{N}(0, W_{-1})$ and it is independent of w_t for all t. The matrix $B_t(j)$ describes how the control input, $u_t(j)$, of the j-th actuator enters the system at the time t. The mapping $\sigma : [0, T-1] \to 2^{\mathsf{N}}$ is called the scheduling function that determines which actuators are active at any time.

We consider the actuator scheduling problem that at each time only $N_t (1 \le N_t \le N)$ out of the N actuators are used to control the system (1) at time t. Consider a standard finite horizon quadratic control cost function

$$J_c = \mathbb{E}\Big[\textstyle\sum_{t=0}^{T-1}\Big(x_t^\top Q_t x_t + \sum_{j\in\sigma_t} u_t(j)^\top R_t(j)u_t(j)\Big) + x_T^\top Q_T x_T\Big], \tag{2}$$

where $Q_T, Q_t, R_t(j) \succ 0$ for all t. In addition, consider also an actuation cost function $J_a = \sum_{t=0}^{T-1}\sum_{j\in\sigma_t} c_t(j)$, where $c_t(j) \in \mathbb{R}_+$ is the cost of using actuator j at time t. Note that $c_t(i)$ and $c_t(j)$ are in general different for $i \neq j$ and $i, j \in \mathsf{N}$, which can be due to the fact that different actuators may have different energy consumption or resource usage. The objective of the actuator scheduling problem is then to find an actuator schedule that minimize the joint cost $J = J_c + J_a$.

Due to space limitations, in the sequel we will restrict ourselves to the case that $N_t = 1$ and $c_t(j) = c_t$ for all j and t. In other words, we consider the case that exactly one out of the N actuators is used at each time and that each actuator has the same actuation cost. The assumption $c_t(j) = c_t$ leads to J_a being independent of the actuator schedule and therefore, minimizing J is equivalent to minimizing J_c. The discussions on the cases with multiple actuators and actuation costs will be provided afterwards in Section IV.

We assume that perfect state measurement is available to the controllers. The information available at the controller at time t is denoted by $\mathcal{I}_t$, with $\mathcal{I}_t = \mathcal{I}_{t-1}\bigcup\{x_t\}$ for all $t \ge 1$ and $\mathcal{I}_0 = \{x_0\}$. For any given schedule σ, the controller for the j-th actuator at time t is $u_t(j) = -L_t(j, \sigma)x_t$, and the cost associated to this schedule σ is

$$J_c(\sigma) = \textstyle\sum_{t=0}^{T} \mathrm{tr}(K_t(\sigma)W_{t-1}),$$

where $L_t(j, \sigma)$ and $K_t(\sigma)$ satisfy the following equations

$$L_t(j, \sigma) = S_t(j, \sigma)^{-1}B_t(j)^\top K_{t+1}(\sigma)A_t, \tag{3}$$

$$S_t(j, \sigma) := B_t(j)^\top K_{t+1}(\sigma)B_t(j) + R_t(j) \tag{4}$$

$$\begin{aligned} K_t(\sigma) &= Q_t + A_t^\top K_{t+1}(\sigma)A_t - \\ &\quad A_t^\top K_{t+1}(\sigma)B_t(\sigma_t)S_t(\sigma_t)^{-1}B_t(\sigma_t)^\top K_{t+1}(\sigma)A_t \\ K_T(\sigma) &:= K_T = Q_T. \end{aligned} \tag{5}$$

Notice that, for any $t \le T-1$, the matrix $K_t(\sigma)$ depends on the actuator schedule for the interval $[t, T-1]$. Thus, the gains $L_t(j, \sigma)$ associated with the j-th actuator depends on the *future* schedule for the time interval $[t+1, T-1]$ and actuator j through $B_t(j)$ and $R_t(j)$.

Before proceeding, to maintain brevity in the subsequent analysis, we define two matrix valued functions:

$$\begin{aligned} g_t(j, M) &:= M \\ &\quad - MB_t(j)\big(B_t(j)^\top MB_t(j) + R_t(j)\big)^{-1}B_t(j)^\top M, \end{aligned} \tag{6a}$$

$$h_t(M) := A_t^\top M A_t + Q_t, \tag{6b}$$

for all $j \in \mathsf{N}$. By defining a new variable $K_{t|t+1}$ and substituting (6) into (5), we obtain that, for any given σ,

$$K_{t|t+1}(\sigma) := g_t(\sigma_t, K_{t+1}(\sigma)). \tag{7a}$$

$$K_t(\sigma) = h_t(K_{t|t+1}(\sigma)), \quad K_T = Q_T. \tag{7b}$$

In what follows, we will suppress the time subscript in σ_t to maintain notation brevity. The optimal actuator scheduling problem that we consider is then formulated as follows.

Problem 1 (Actuator Scheduling Problem): Given system (1) and N actuators, find a schedule $\sigma : [0, T-1] \to \mathsf{N}$ that solves the following optimization problem:

$$\begin{aligned} &\min \textstyle\sum_{t=0}^{T} \mathrm{tr}(K_t(\sigma)W_{t-1}) \\ &\text{subject to } K_{t|t+1}(\sigma) = g_t(\sigma, K_{t+1}(\sigma)), \\ &\qquad\qquad K_t(\sigma) = h_t(K_{t|t+1}(\sigma)),\ K_T = Q_T \end{aligned}$$

with the variables $\sigma, K_t, K_{t|t+1}$.

Problem 1 is combinatorial in nature and NP-hard in general [17]. We now propose an efficient solution to Problem 1 using a convex relaxation.

III. Actuator Scheduling with Suboptimality Guarantees

In this section, we solve Problem 1 and provide a suboptimal solution that is computationally inexpensive. We will propose a convex relaxation to the problem (see Problem 3) and will use the solution of the relaxed problem as a 'reference' to find a solution to Problem 1. In Section III-A we will propose a *tracking algorithm* that finds a solution which is 'close' to the reference solution found from solving the relaxed convex optimization problem. The suboptimality bound of the proposed algorithm is discussed using dynamic programming type arguments in Section III-B.

Before proceeding, we first reformulate Problem 1 into a form that is easier for the analysis afterwards. According to (6b) and (7b), we have $K_t(\sigma) = A_t^\top K_{t|t+1}(\sigma) A_t + Q_t$, $t = 0, 1, \ldots, T-1$ and $K_T = Q_T$. Subsequently, we obtain

$$\textstyle\sum_{t=0}^{T} \operatorname{tr}(K_t(\sigma) W_{t-1}) = \sum_{t=0}^{T-1} \operatorname{tr}(K_{t|t+1}(\sigma) \bar{W}_{t-1}) + r, \quad (8)$$

where $r = \sum_{t=0}^{T} \operatorname{tr}(Q_t W_{t-1})$ and $\bar{W}_{t-1} = A_t W_{t-1} A_t^\top$. Note that r is independent of $\sigma, K_t(\sigma)$ and $K_{t|t+1}(\sigma)$.

Next, we define two matrices P_t and $P_{t|t+1}$ as follows

$$P_t(\sigma) := K_t^{-1}(\sigma), \quad P_{t|t+1}(\sigma) := K_{t|t+1}^{-1}(\sigma).$$

According to Woodbury matrix equality, we have

$$P_{t|t+1}(\sigma) = P_{t+1}(\sigma) + B_t(\sigma) R_t^{-1}(\sigma) B_t^\top(\sigma). \quad (9)$$

Using (8) and the new variables $P_{t|t+1}(\sigma)$, $P_t(\sigma)$, Problem 1 can be rewritten as Problem 2.

Problem 2: Given system (1) with N actuators, find a schedule $\sigma : [0, T-1] \to \mathsf{N}$ that solves the following:

$$\begin{aligned} &\min \textstyle\sum_{t=0}^{T-1} \operatorname{tr}(K_{t|t+1}(\sigma) \bar{W}_{t-1}) \\ &\text{subject to } K_{t|t+1}(\sigma) = P_{t|t+1}^{-1}(\sigma), \\ &\qquad P_{t|t+1}(\sigma) = P_{t+1}(\sigma) + B_t(\sigma) R_t^{-1}(\sigma) B_t^\top(\sigma), \\ &\qquad P_t^{-1}(\sigma) = h_t(K_{t|t+1}(\sigma)),\ P_T = Q_T^{-1} \end{aligned}$$

with variables $\sigma, K_t, K_{t|t+1}, P_t, P_{t|t+1}$.

Note that, although the constraints in Problem 1 and Problem 2 appear differently, one can in fact verify that these two problems are equivalent.

Let us denote $V_t(\sigma) := B_t(\sigma) R_t^{-1}(\sigma) B_t^\top(\sigma)$ and the set $\mathsf{V}_t := \{B_t(j) R_t(j)^{-1} B_t(j)^\top \ : \ j \in \mathsf{N}\}$. Therefore, we may rewrite the constraints in Problem 2 to be $K_{t|t+1} = P_{t|t+1}^{-1}$, $P_{t|t+1} = P_{t+1} + V_t$, $V_t \in \mathsf{V}_t$, $P_t = \big(h_t(K_{t|t+1})\big)^{-1}$, and $P_T = Q_T^{-1}$. We have suppressed the arguments σ in the variables to maintain notational brevity. We can further relax the constraints in Problem 2 to their equivalent matrix inequality $K_{t|t+1} \succeq P_{t|t+1}^{-1}$, and $P_t \preceq (h_t(K_{t|t+1}))^{-1}$. Using Schur complement, one may write $K_{t|t+1} \succeq P_{t|t+1}^{-1}$ as the Linear matrix inequality $\begin{bmatrix} K_{t|t+1} & I \\ I & P_{t|t+1} \end{bmatrix} \succeq 0$. Similarly, using the definition of $h_t(\cdot)$ from (6b), the Woodbury matrix inverse identity, and Schur complement, we obtain the following problem from Problem 2.

Problem 3: Given $\mathsf{V}_t := \{B_t(j) R_t(j)^{-1} B_t(j)^\top \ : \ j \in \mathsf{N}\}$, solve the following optimization problem

$$\begin{aligned} &\min \textstyle\sum_{t=0}^{T-1} \operatorname{tr}(K_{t|t+1} \bar{W}_{t-1}) \\ &\text{subject to } P_{t|t+1} = P_{t+1} + V_t,\ V_t \in \mathsf{V}_t,\ P_T = Q_T^{-1}, \\ &\qquad \begin{bmatrix} K_{t|t+1} & I \\ I & P_{t|t+1} \end{bmatrix} \succeq 0, \\ &\qquad \begin{bmatrix} Q_t^{-1} - P_t & Q_t^{-1} A_t^\top \\ A_t Q_t^{-1} & P_{t|t+1} + A_t Q_t^{-1} A_t^\top \end{bmatrix} \succeq 0 \end{aligned}$$

with variables $K_{t|t+1}, P_{t|t+1}, P_t$, and V_t.

Notice that the constraint $V_t \in \mathsf{V}_t$ is sufficient to enforce the scheduling constraint $\sigma : [0, T-1] \to \mathsf{N}$.

While Problem 3 is a relaxation of Problem 2, we now show a key result that an optimal solution to Problem 3 is also an optimal solution to Problem 2.

Theorem 1: An optimal solution of the relaxed problem (Problem 3) is also an optimal solution of the original problem (Problem 2), and vice-versa.

Proof: The proof of this theorem is along the lines of [7, Theorem 1]. First, note that, due to the relaxations, any feasible solution of Problem 2 is a feasible solution for Problem 3, and hence the optimal solution of Problem 2 is a feasible solution for Problem 3. The theorem is proved once we show that for every feasible solution of Problem 3 there exists a feasible solution for Problem 2 that produces the same, if not a smaller, objective value.

In order to show that, let the tuple $\{K_{t|t+1}, P_{t|t+1}, P_t\}$ denote a feasible solution of Problem 3. Let us construct a new tuple $\{\bar{K}_{t|t+1}, \bar{P}_{t|t+1}, \bar{P}_t\}$ as follows

$$\begin{aligned} &\bar{P}_{t|t+1} = \bar{P}_{t+1} + \bar{V}_t, \quad \bar{P}_t = \big(h_t(\bar{K}_{t|t+1})\big)^{-1} \\ &\bar{V}_t = P_{t|t+1} - P_{t+1}, \quad \bar{K}_{t|t+1} = \bar{P}_{t|t+1}^{-1}, \quad \bar{P}_T = P_T. \end{aligned} \quad (10)$$

It then follows from (10) that $\bar{P}_{t|t+1} \succeq P_{t|t+1}$, $\bar{K}_{t|t+1} \preceq K_{t|t+1}$ and $\bar{P}_{t+1} \succeq P_{t+1}$ for all t. Note also that the matrix $\bar{V}_t$ in (10) satisfies $\bar{V}_t \in \mathsf{V}_t$. Since the tuple $\{\bar{K}_{t|t+1}, \bar{P}_{t|t+1}, \bar{P}_t\}$ satisfies all the constraints of Problem 2, this implies that it is a feasible solution of Problem 2. Next, note that $\bar{K}_{t|t+1} \preceq K_{t|t+1}$ for all t, this then implies that $\sum_{t=0}^{T-1} \operatorname{tr}(\bar{K}_{t|t+1} \bar{W}_{t-1}) \preceq \sum_{t=0}^{T-1} \operatorname{tr}(K_{t|t+1} \bar{W}_{t-1})$. Therefore, for any feasible solution of Problem 3 we can construct a feasible solution for Problem 2 that produces the same, if not less, cost. This completes the proof. ■

Remark 2: Theorem 1 shows that the LMI-based relaxations introduced in Problem 3 do not affect the optimality, since an optimal solution to the relaxed problem is also optimal for the original problem. This is a key advantage of this approach, as the LMI-based relaxations retain the optimality. Moreover, since Problem 3 is a mixed integer semidefinite program, one may attempt to directly solve it using available numerical techniques [18].

Next, note that Problem 3 is convex if V_t is a convex set for all t. When V_t is not convex, one could take the convex hull of the set V_t to make Problem 3 convex. In our case, since V_t is a collection of N matrices $\{V_t(1), \ldots, V_t(N)\}$ where $V_t(j) = B_t(j) R_t(j)^{-1} B_t(j)^\top$ for all t, we replace the constraint $V_t \in \mathsf{V}_t$ with the constraints $V_t = \sum_{i=1}^{N} \theta_t^i V_t(i)$,

$\theta_t^i \in [0,1]$ and $\sum_{i=1}^N \theta_t^i = 1$. In this case, Problem 3 can be further simplified to Problem 4.

Problem 4:

$$\begin{aligned} &\min \textstyle\sum_{t=0}^{T-1} \mathrm{tr}(K_{t|t+1}\bar{W}_{t-1}) \\ &\text{subject to } P_{t|t+1} = P_{t+1} + \textstyle\sum_{i=1}^N \theta_t^i V_t(i),\ P_T = Q_T^{-1}, \\ &\qquad \textstyle\sum_{i=1}^N \theta_t^i = 1, 0 \le \theta_t^i \le 1, \begin{bmatrix} K_{t|t+1} & I \\ I & P_{t|t+1} \end{bmatrix} \succeq 0, \\ &\qquad \begin{bmatrix} Q_t^{-1} - P_t & Q_t^{-1}A_t^\top \\ A_tQ_t^{-1} & P_{t|t+1} + A_tQ_t^{-1}A_t^\top \end{bmatrix} \succeq 0, \end{aligned}$$

with variables $\theta_t^i, K_{t|t+1}, P_{t|t+1}, P_t$.

At this point we have a convex optimization problem (semidefinite program) in Problem 4 which is much easier to solve compared to the mixed integer semidefinite program in Problem 3. If the optimal θ_t^j is binary-valued then the optimal schedule to Problem 1 is found by setting $\sigma_t = j$ such that $\theta_t^j = 1$. However, in general the optimal θ_t^j are not binary-valued and we need to design an algorithm to find a schedule σ from the solution to Problem 4.

Remark 3: At first glance, it may seem that selecting the actuator with the maximum value of θ_t^i at each time will lead to the smallest value of $\mathrm{tr}(K_{t|t+1}\bar{W}_{t-1})$. However, it is not necessarily the case (see simulation in Section V). In the next section we propose a more efficient algorithm and discuss its suboptimality bound.

A. Actuator Scheduling Algorithm

By solving the convex relaxation in Problem 4, we obtain $\{\{\theta_t^{i^o}\}_{i\in\mathsf{N}}\}_{t=0}^{T-1}$, or equivalently $V_t^o = \sum_{i=1}^N \theta_t^{i^o} V_t^i$ and the associated $K_{t|t+1}^o$, $P_{t|t+1}^o$ and P_t^o. In this section, we propose an algorithm that uses this solution of Problem 4 as a reference to obtain a suboptimal solution for Problem 1. The corresponding algorithm is presented in Algorithm 1. Note that this algorithm depends linearly on the number of the actuators.

Algorithm 1 takes the solution $\{K_{t|t+1}^o\}_{t=0}^{T-1}$ obtained from solving Problem 4 as an initial guess, and initializes the terminal condition K_T at Q_T. The algorithm produces a trajectory $\{K_{t|t+1}\}_{t=0}^{T-1}$ that is *close* to the reference trajectory $\{K_{t|t+1}^o\}_{t=0}^{T-1}$ in Frobenius norm. The reasoning behind the construction of Algorithm 1 is to keep the matrices $K_{t|t+1}(\sigma)$ close to $K_{t|t+1}^o$, and subsequently, to keep $\sum_{t=0}^{T-1}\mathrm{tr}(K_{t|t+1}(\sigma)\bar{W}_{t-1})$ close to $\sum_{t=0}^{T-1}\mathrm{tr}(K_{t|t+1}^o\bar{W}_{t-1})$, since $\sum_{t=0}^{T-1}\mathrm{tr}(K_{t|t+1}^o\bar{W}_{t-1})$ is the lowest one that could possibly be achieved given the set of actuators. The algorithm can be regarded as a *trajectory-tracking* problem in the space of positive definite matrices where $\{K_{t|t+1}^o\}_{t\ge 0}$ serves as the reference trajectory.

Although Algorithm 1 is heuristic in nature, we may use dynamic programming type arguments to analyze its performance. To this end, we denote the value function associated to Problem 2 as

$$U_t(K) = \min_{\{\sigma(k)\}_{k=0}^t} \textstyle\sum_{k=0}^t \mathrm{tr}(K_{k|k+1}(\sigma)\bar{W}_{k-1}), \quad (11)$$

given $K_{t+1} = K$ for some $K \succ 0$. Likewise, we denote the value function associated with Problem 4, which is the SDP

Algorithm 1 Reference Tracking Algorithm

Input $\{K_{t|t+1}^o\}_{t=0}^{T-1}$, $K_T = Q_T$
for $t = T-1:0$ **do**
  $M_t(i) \leftarrow g_t(i, K_{t+1}), \quad i \in \mathsf{N}$
  $\sigma_t \leftarrow \mathrm{argmin}_i \|K_{t|t+1}^o - M_t(i)\|_F$
  $K_{t|t+1} \leftarrow g_t(\sigma_t, K_{t+1})$
  $K_t \leftarrow h_t(K_{t|t+1})$
end for
Output σ

relaxation of Problem 1, to be

$$U_t^o(K) = \min_{\{\{\theta_k^i\}_{i\in\mathsf{N}}\}_{k=0}^t} \textstyle\sum_{k=0}^t \mathrm{tr}(K_{k|k+1}(\theta)\bar{W}_{k-1}). \quad (12)$$

It in fact can be shown that

$$U_t(K) \le \alpha_t + U_t^o(K_{t+1}^o) + c_1 \min_{\sigma_t} \|K_{t|t+1}(\sigma) - K_{t|t+1}^o\|_F,$$

where $K_t = A_t^\top K_{t|t+1}A_t + Q_t$, $c_1 := \|\bar{W}_{t-1}\|_F + c\|A_t\|_F^2$ and $\alpha_t > 0$ depends on t but not K or σ. Thus, optimizing $\min_\sigma \|K_{t|t+1}(\sigma) - K_{t|t+1}^o\|_F$ in Algorithm 1 in fact minimizes an upper bound of the value function U_t, or equivalently, an upper bound of $\sum_{t=0}^{T-1}\mathrm{tr}(K_{t|t+1}\bar{W}_{t-1})$. Therefore, in essence, Algorithm 1 performs an approximate dynamic programming type optimization by minimizing an upper bound of U_t.

The reader is referred to [19] for detailed derivations.

B. Suboptimality Guarantees

The following theorem provides a suboptimality bound of Algorithm 1.

Theorem 4: Let σ, σ^* and θ^* denote the schedule obtained from Algorithm 1, the true optimal schedule of Problem 1, and the solution to Problem 4, respectively. Then, we have

$$\textstyle\sum_{t=0}^T \mathrm{tr}(K_t(\sigma)W_{t-1}) \le \sum_{t=0}^T \mathrm{tr}(K_t(\sigma^*)W_{t-1}) + \epsilon, \quad (13)$$

where

$$\begin{aligned} \epsilon \triangleq \|\bar{W}_{t-1}\|_F \Big(&\textstyle\sum_{t=0}^{T-1} \frac{\lambda^{t+1}-1}{\lambda-1}\beta_t \\ &+ \textstyle\sum_{t=0}^{T-1} \|K_{t|t+1}(\theta^*) - K_{t|t+1}(\sigma^*)\|_F \Big) \end{aligned} \quad (14)$$

with $\lambda_t \triangleq \|A_{t+1}\|^2\|H_t(\sigma^*, K_{t+1}(\theta^*))\|^2$, $\beta_t \triangleq \|g_t(\sigma_t^*, K_{t+1}(\theta^*)) - K_{t|t+1}(\theta^*)\|_F$ and

$$\begin{aligned} H_t(\sigma^*, K_{t+1}(\theta^*)) &\triangleq I - K_{t+1}(\theta^*)B_t(\sigma^*) \\ &\times \big(B_t(\sigma^*)^\top K_{t+1}(\theta^*)B_t(\sigma^*) + R_t(\sigma^*)\big)^{-1} B_t(\sigma^*)^\top. \end{aligned}$$

Proof: First, let us recall (8) and we then have

$$\begin{aligned} &\textstyle\sum_{t=0}^T \mathrm{tr}(K_t(\sigma)W_{t-1}) - \sum_{t=0}^T \mathrm{tr}(K_t(\sigma^*)W_{t-1}) \\ &= \textstyle\sum_{t=0}^{T-1} \mathrm{tr}\Big(\big(K_{t|t+1}(\sigma) - K_{t|t+1}(\sigma^*)\big)\bar{W}_{t-1}\Big) \\ &\le \textstyle\sum_{t=0}^{T-1} \|K_{t|t+1}(\sigma) - K_{t|t+1}(\sigma^*)\|_F \|\bar{W}_{t-1}\|_F. \end{aligned}$$

Next, for all t, it holds that

$$\begin{aligned} \|K_{t|t+1}(\sigma) - K_{t|t+1}(\sigma^*)\|_F &\le \|K_{t|t+1}(\sigma) - K_{t|t+1}(\theta^*)\|_F \\ &+ \|K_{t|t+1}(\theta^*) - K_{t|t+1}(\sigma^*)\|_F, \end{aligned} \quad (15)$$

where $K_{t|t+1}(\sigma)$ is the obtained matrix when schedule σ is used from time $T-1$ backwards to t. Similarly, we define $K_t(\sigma^*)$ and $K_t(\theta^*)$. Furthermore, according to the definition of θ^*, we have $K_{t|t+1}(\theta^*) = K^o_{t|t+1}$. Next, note that, due to the design of our algorithm (line 4 in Algorithm 1), it holds

$$\begin{aligned}
&\|K_{t|t+1}(\sigma) - K_{t|t+1}(\theta^*)\|_F \\
=&\min_i \|g_t(i, K_{t+1}(\sigma)) - K_{t|t+1}(\theta^*)\|_F \\
\le&\|g_t(\sigma_t^*, K_{t+1}(\sigma)) - K_{t|t+1}(\theta^*)\|_F \\
\le&\|g_t(\sigma_t^*, K_{t+1}(\sigma)) - g_t(\sigma_t^*, K_{t+1}(\theta^*))\|_F \\
&\quad + \|g_t(\sigma_t^*, K_{t+1}(\theta^*)) - K_{t|t+1}(\theta^*)\|_F.
\end{aligned}$$

It then follows from [19, Lemma 4] and the concavity of $g_t(i, \cdot)$ that

$$\begin{aligned}
&\|g_t(\sigma_t^*, K_{t+1}(\sigma)) - g_t(\sigma_t^*, K_{t+1}(\theta^*))\|_F \\
&\qquad \le \|H_t(\sigma_t^*, K_{t+1}(\theta^*))\|^2 \|A_{t+1}\|^2 \\
&\qquad\qquad \times \|K_{t+1|t+2}(\sigma)) - K_{t+1|t+2}(\theta^*)\|_F.
\end{aligned}$$

By defining $\eta_t \triangleq \|K_{t|t+1}(\sigma) - K_{t|t+1}(\theta^*)\|_F$, $\lambda_t \triangleq \|A_{t+1}\|^2 \|H_t(\sigma_t^*, K_{t+1}(\theta^*))\|^2$, $\beta_t \triangleq \|g_t(\sigma_t^*, K_{t+1}(\theta^*)) - K_{t|t+1}(\theta^*)\|_F$, we obtain

$$\eta_t \le \lambda \eta_{t+1} + \beta_t, \ t = 0, 1, \ldots, T-2, \ \eta_{T-1} \le \beta_{T-1}, \quad (16)$$

where $\lambda = \max_t \lambda_t$. This further gives us that $\eta_t \le \sum_{i=1}^{T-t} \lambda^{T-t-i} \beta_{T-i}$, for $t = 0, 1, \ldots, T-1$. It then follows from (15), (16) and the definition of η_t that

$$\begin{aligned}
&\textstyle\sum_{t=0}^{T-1} \|K_{t|t+1}(\sigma) - K_{t|t+1}(\sigma^*)\|_F \\
\le& \textstyle\sum_{t=0}^{T-1} \eta_t + \sum_{t=0}^{T-1} \|K_{t|t+1}(\theta^*) - K_{t|t+1}(\sigma^*)\|_F \\
\le& \textstyle\sum_{t=0}^{T-1} \frac{\lambda^{t+1}-1}{\lambda-1} \beta_t + \sum_{t=0}^{T-1} \|K_{t|t+1}(\theta^*) - K_{t|t+1}(\sigma^*)\|_F.
\end{aligned}$$

This implies that

$$\textstyle\sum_{t=0}^{T-1} \mathrm{tr}(K_{t|t+1}(\sigma)\bar{W}_{t-1}) - \sum_{t=0}^{T-1} \mathrm{tr}(K_{t|t+1}(\sigma^*)\bar{W}_{t-1}) \le \epsilon,$$

where ϵ is given in (14). This completes the proof. ■

Remark 5: Note that equation (13) in Theorem 4 provides a suboptimality bound on Algorithm 1. According to the definition of β_t, it can be seen that the value of ϵ depends on the mismatch between the schedules θ^* and σ^*. Clearly, if the solution to Problem 4 is already integer in nature (i.e., $\theta_t^* \in \{0, 1\}$) for all t, then $\beta_t = 0$ for all t, and consequently we obtain $\epsilon = 0$.

IV. Discussion on Multiple Actuator Scheduling and Actuation Costs

In this section, we will provide brief discussions on the cases of multiple actuator scheduling and actuation costs.

1) Multiple Actuator Scheduling: In Section III, we considered the problem for the case where exactly one actuator is used at each time. In practice, one may encounter a situation that multiple actuators (e.g., N_t out of N) are scheduled at the same time. Such a problem can be solved in several ways using our method. Here we discuss two of them.

As a first approach, one may construct $\binom{N}{N_t}$ *virtual actuators*, each of these is a group of N_t actuators. Thus, selecting N_t out of N actuators is equivalent to selecting one out of these $\binom{N}{N_t}$ virtual actuators. However, complexity of such an approach grows factorially. A less computationally expensive approach is to use $\sum_{i=1}^N \theta_t^i = N_t$ in Problem 4, along with a modification in Algorithm 1, in which case the N_t actuators that give the smallest values of $\|K^o_{t|t+1} - M_t(i)\|_F$ are the actuators selected at time t. This modification in Problem 4 does not introduce any extra computational complexity. Computational requirements for Algorithm 1 slightly increases. However, given the simplicity of Algorithm 1, this is practically inconsequential.

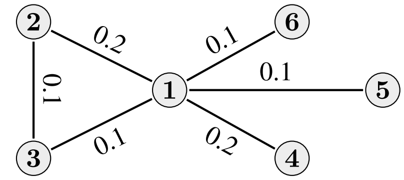

Fig. 1: Network model for simulation example.

2) Actuation Costs: The results in Section III are derived by considering all actuators to have equal actuation costs (i.e., $c_t(i) = c_t$ for all i). One possible way to incorporate the actuation costs is to include the term $\sum_{t=0}^{T-1} \sum_{i=1}^N c_t(i)\theta_t^i$ in the objective function of (4). Notice that the term $\sum_{t=0}^{T-1} \sum_{i=1}^N c_t(i)\theta_t^i$ is linear in the optimization variable θ_t, and hence the convexity of the problem is retained. In the simulation we adopt this approach to include actuation costs.

V. Simulation

We consider a networked system with 6 nodes as shown in Fig. 1. The i-th node follows the dynamics

$$x_{t+1}(i) = \textstyle\sum_{j=1}^6 a_{ij} x_t(j) + u_t(i) + w_t(i),$$

where $a_{ij} \ge 0$ denotes the weight on the link between nodes i and j and $a_{ii} = 1 - \sum_{j=1, j\neq i}^6 a_{ij}$. If there is no link present between node i and j, then $a_{ij} = 0$. Each node has an actuator associated with it through which one can directly control the state of that node. The overall system state $x_t = [x_t(1), \ldots, x_t(6)]^\top$ follows the dynamics

$$x_{t+1} = A x_t + \textstyle\sum_{i=1}^6 B(i) u_t(i) + w_t,$$

where $B(i) \in \mathbb{R}^6$ is i-th canonical basis vector in $\mathbb{R}^6$ and $w_t = [w_t(1), \ldots, w_t(6)]^\top$. We consider a cost function of the form (2) with $Q_t = \frac{1}{2}I, R_t(i) = I$ for all $t \le T-1$ and $Q_T = I$. Furthermore, we assume $x_0 \sim \mathcal{N}(0, \frac{1}{2}I)$ and $w_t \sim \mathcal{N}(0, \frac{1}{4}I)$. The actuation costs are $c_t(i) = 1$ for $i = 1, \ldots, 4$, $c_t(5) = 1.5$, and $c_t(6) = 2$. The costs for $c_t(5)$ and $c_t(6)$ are chosen to be higher because the system is fully controllable only with $B(5)$ and $B(6)$. For a horizon of $T = 30$, the schedule obtained from our algorithm is shown in Fig. 2, and the corresponding optimal cost is 101.0006.

Interestingly, from the solution to Problem 4 shown in Fig. 3, we notice that the actuator of the 1-st node is hardly used since the values of θ_t^1's are in orders of magnitude smaller than that for the rest of the nodes for all t. This is in contrast with the schedule we found in Fig. 2 where actuator 1 is scheduled for several time instances ($\sim$ 17% of the time) by Algorithm 1. Actuator 2 is used the least by Algorithm 1

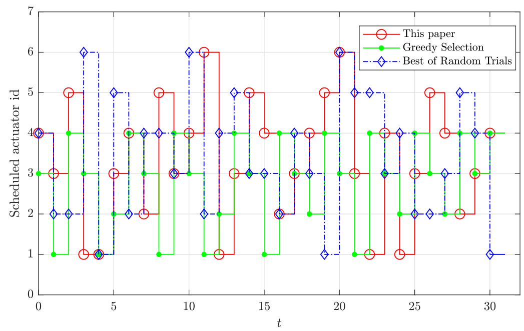

Fig. 2: The actuator schedules using different methods.

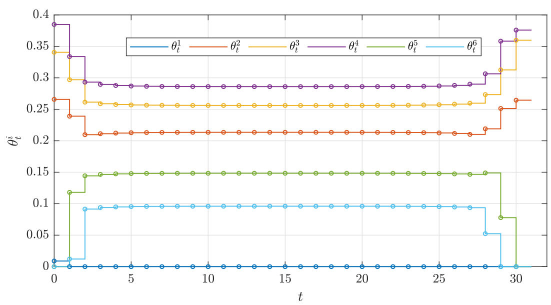

Fig. 3: Optimal θ_t^i from Problem 4. For all t, we obtained $\theta_t^5 = \theta_t^6$, and hence they are not distinguishable in the figure.

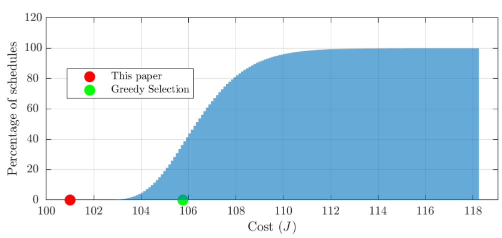

Fig. 4: x-axis: Cost (J), y-axis: Percentage of the randomly generated trials which produced a cost less than or equal to the value on the x-axis.

in Fig. 2, however, in Fig. 3 we notice that θ_t^2 is not the least among all θ_t^i's. While one might be tempted to only use actuators 3 and 4 since the corresponding θ_t^i values are the highest ones in Fig. 3, however, such restriction leads to a cost of 108.5531, which is higher than what our method found. This indeed validates our statements in Remark 3.

Next, we compare the performance of our approach with randomly generated schedules and a greedy selection approach[1]. We randomly selected 50,000 schedules and computed the cost corresponding to these schedules. The resulting cost distribution from the schedules are plotted in Fig. 4 and the minimum cost out of these 50,000 trials is 102.0693.

Evaluation of the 50,000 random trials took 34.65 seconds whereas our approach (convex optimization plus trajectory tracking) took 2.5 seconds, which is an order of magnitude less time. For the greedy approach, at each time instance we greedily selected the actuator that provides the minimum cost for that time stage. This approach is fast (< 1 sec) but the performance is the worst (see. Fig. 4).

[1]Scheduling problems generally has a supermodularity structure which ensures a level of optimality guarantee for the greedy approach.

VI. Conclusions

In this letter, we have studied the problem of actuator scheduling for stochastic linear NCSs. In particular, we have considered the case that only one actuator is active at each time. We have proposed a convex relaxation and used its solution as a reference for obtaining a suboptimal tracking algorithm for solving the actuator scheduling problem. Suboptimality guarantees for the proposed algorithm have been provided using dynamic programming arguments. We have also discussed the extensions on the cases with multiple actuator scheduling and actuation costs.

References

[1] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 138–162, 2007.

[2] G. C. Walsh and H. Ye, "Scheduling of networked control systems," *IEEE control systems magazine*, vol. 21, no. 1, pp. 57–65, 2001.

[3] J. S. Baras and A. Bensoussan, "Sensor scheduling problems," in *Proc. of the 27th IEEE Conference on Decision and Control (CDC)*, 1988.

[4] V. Gupta, T. H. Chung, B. Hassibi, and R. M. Murray, "On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage," *Automatica*, vol. 42, no. 2, pp. 251–260, 2006.

[5] M. P. Vitus, W. Zhang, A. Abate, J. Hu, and C. J. Tomlin, "On efficient sensor scheduling for linear dynamical systems," *Automatica*, vol. 48, no. 10, pp. 2482–2493, 2012.

[6] V. Tzoumas, L. Carlone, G. J. Pappas, and A. Jadbabaie, "LQG control and sensing co-design," *IEEE Transactions on Automatic Control*, vol. 66, no. 4, pp. 1468–1483, 2021.

[7] D. Maity, D. Hartman, and J. S. Baras, "Sensor scheduling for linear systems: a covariance tracking approach," *Automatica*, vol. 136, p. 110078, 2022.

[8] 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, 2014.

[9] T. H. Summers and J. Lygeros, "Optimal sensor and actuator placement in complex dynamical networks," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 3784–3789, 2014.

[10] F. L. Cortesi, T. H. Summers, and J. Lygeros, "Submodularity of energy related controllability metrics," in *53rd IEEE conference on decision and control*. IEEE, 2014, pp. 2883–2888.

[11] M. Siami, A. Olshevsky, and A. Jadbabaie, "Deterministic and randomized actuator scheduling with guaranteed performance bounds," *IEEE Transactions on Automatic Control*, vol. 66, no. 4, pp. 1686–1701, 2021.

[12] C. Maheshwari, S. Srikant, and D. Chatterjee, "Stabilization under round-robin scheduling of control inputs in nonlinear systems," *Automatica*, vol. 134, p. 109912, 2021.

[13] E. Nozari, F. Pasqualetti, and J. Cortés, "Time-invariant versus time-varying actuator scheduling in complex networks," in *2017 American Control Conference (ACC)*. IEEE, 2017, pp. 4995–5000.

[14] L. Shi, Y. Yuan, and J. Chen, "Finite horizon LQR control with limited controller-system communication," *IEEE Transactions on Automatic Control*, vol. 58, no. 7, pp. 1835–1841, 2013.

[15] L. Mo, P. You, X. Cao, Y. Song, and A. Kritikakou, "Event-driven joint mobile actuators scheduling and control in cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 11, pp. 5877–5891, 2019.

[16] P. Bommannavar and T. Basar, "Optimal control with limited control actions and lossy transmissions," in *2008 47th IEEE conference on decision and control*. IEEE, 2008, pp. 2032–2037.

[17] V. Tzoumas, A. Jadbabaie, and G. J. Pappas, "Near-optimal sensor scheduling for batch state estimation: Complexity, algorithms, and limits," in *55th IEEE Conference on Decision and Control*, Las Vegas, USA, 2016, pp. 2695–2702.

[18] T. Gally, M. E. Pfetsch, and S. Ulbrich, "A framework for solving mixed-integer semidefinite programs," *Optimization Methods and Software*, vol. 33, no. 3, pp. 594–632, 2018.

[19] J. Jiao, D. Maity, J. S. Baras, and S. Hirche, "Actuator scheduling for linear systems: a convex relaxation approach," *arXiv preprint*, 2022, [Online]. Available: https://arxiv.org/abs/2203.02321.