Time-Varying Sensor and Actuator Selection for Uncertain Cyber-Physical Systems*

Ahmad F. Taha, *Member, IEEE,* Nikolaos Gatsis *Member, IEEE,* Tyler Summers *Member, IEEE*,
Sebastian A. Nugroho, *Student Member, IEEE.*

***Abstract*—We propose methods to solve time-varying, sensor and actuator (SaA) selection problems for uncertain cyber-physical systems. We show that many SaA selection problems for optimizing a variety of control and estimation metrics can be posed as semidefinite optimization problems with mixed-integer bilinear matrix inequalities (MIBMIs). Although this class of optimization problems are computationally challenging, we present tractable approaches that directly tackle MIBMIs, providing both upper and lower bounds, and that lead to effective heuristics for SaA selection. The upper and lower bounds are obtained via successive convex approximations and semidefinite programming relaxations, respectively, and selections are obtained with a slicing algorithm from the solutions of the bounding problems. Custom branch-and-bound and combinatorial greedy approaches are also developed for a broad class of systems for comparison. Finally, comprehensive numerical simulations are performed to compare the different methods and illustrate their effectiveness.**

***Index Terms*—Sensor and actuator selection, cyber-physical systems, linear matrix inequalities, controller/estimator design.**

I. Introduction & Brief Literature Review

MANY emerging complex dynamical networks, from critical infrastructures to industrial cyber-physical systems (CPSs) and various biological networks, are increasingly able to be instrumented with new sensing and actuation capabilities. These networks comprise growing webs of interconnected feedback loops and must operate efficiently and resiliently in dynamic and uncertain environments. The prospect of incorporating large numbers of additional sensors and actuators (SaAs) raises fundamental and important problems of jointly and dynamically selecting the most effective SaAs, in addition to simultaneously designing corresponding estimation and control laws associated with the selected SaAs.

There are many different quantitative notions of network controllability and observability that can be used as a basis for selecting effective SaAs in uncertain and dynamic cyber-physical networks. Notions based on classical Kalman rank conditions for linear systems focus on binary structural properties [2]–[5]. More elaborate quantitative notions based on Gramians [6]–[12] and classical optimal and robust control and estimation problems [13]–[20] for linear systems have also been studied. For selecting SaAs based on these metrics, several optimization methods are proposed in this literature, including combinatorial greedy algorithms [8], [9], [17], [19], [21], convex relaxations using sparsity-inducing ℓ_1 penalty functions [13]–[16] and reformulations to mixed-integer semidefinite programming via the Big-M method or McCormick's relaxation [12], [20], [22]. As a departure from control-theoretic frameworks, the authors in [23] explore routines for reconstructing the initial states of nonlinear systems while optimally selecting a fixed number of sensors.

Despite the recent surge of interest in quantifying network controllability and observability and in associated SaA selection problems, a much wider set of metrics are relevant for uncertain cyber-physical systems. The existing literature tends to focus mainly on classical metrics (e.g., involving Kalman rank [2], Gramians [9], [11], [17], Linear Quadratic Regulators [12], [17], [18], and Kalman Filters [19], [20]) and deterministic linear time-invariant systems. Methods for time-varying systems with various uncertainties, constraints, and nonlinearities are also important to broaden applicability. It is well known that a broad variety of systems and control problems can be cast in the form of semidefinite programs (SDP) and linear matrix inequalities (LMIs) [24], but many of these more recent formulations have not been considered in the context of SaA selection. In general, the selection of sensors or actuators and design of associated estimation and control laws for many metrics can be posed as semidefinite optimization problems with mixed-integer bilinear matrix inequalities (MIBMIs) as we have recently shown in [25]. A general MIBMI formulation for the selection problem is also discussed in the ensuing sections.

Here we propose methods to solve time-varying sensor and actuator (SaA) selection problems for uncertain CPSs. Our methods can be applied to any of the broad range of problems formulated as MIBMIs. Although this class of optimization problems is computationally challenging, we present tractable approaches that provide bounds and lead to effective heuristics for SaA selection. The bounds are obtained via successive convex approximations and SDP relaxations, respectively, and selections are obtained with a slicing algorithm from the solutions of the bounding problems.

A preliminary version of this work appeared in [25] where we developed customized algorithms for actuator selection. Here we significantly extended the methodology with the successive convex approximation and convex relaxation approaches and provide comprehensive numerical simulations. **The extended version of this paper can be found in [1]**. It includes significant additions to the paper: (1) extensions of this work to include a variety of other control and estimation metrics with sensor/actuator selection; (2) an alternate formu-

*The extended version of this paper can be found on arXiv [1]. Ahmad F. Taha, Nikolaos Gatsis, and Sebastian A. Nugroho are affiliated with the Department of Electrical and Computer Engineering at the University of Texas San Antonio. Tyler Summers is with the Department of Mechanical Engineering, University of Texas at Dallas. Emails: {ahmad.taha, nikolaos.gatsis}@utsa.edu, tyler.summers@utdallas.edu, sebastian.nugroho@my.utsa.edu. This material is based upon work supported in part by the National Science Foundation under Grants No. ECCS-1462404, CMMI 1728629, and CMMI 1728605. The work of T. Summers was partially sponsored by the Army Research Office and was partially accomplished under Grant Number: W911NF-17-1-0058.

lation for the SaA selection problem via the Big-M method that amounts to solving mixed-integer SDPs, in comparison with the convex relaxations and approximations we develop in this paper; and (3) a thorough discussion on greedy algorithms, and extended numerical tests for another dynamical system with different network size and properties.

II. CPS Model and Paper Contributions

We consider time-varying CPSs with N nodes modeled as

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}^j\boldsymbol{x}(t) + \boldsymbol{B}_u^j\boldsymbol{\Pi}^j\boldsymbol{u}(t) + \boldsymbol{B}_w^j\boldsymbol{w}(t) + \boldsymbol{B}_f^j\boldsymbol{\phi}^j(\boldsymbol{x}), \tag{1a}$$

$$\boldsymbol{y}(t) = \boldsymbol{\Gamma}^j\boldsymbol{C}^j\boldsymbol{x}(t) + \boldsymbol{D}_u^j\boldsymbol{u}(t) + \boldsymbol{D}_v^j\boldsymbol{v}(t), \quad \boldsymbol{x}^j(t_0) = \boldsymbol{x}_0^j \tag{1b}$$

The network state $\boldsymbol{x}(t) \in \mathbb{R}^{n_x}$ consists of each of N nodal agent states $\boldsymbol{x}_i \in \mathbb{R}^{n_{x_i}}$, $i = 1, \ldots, N$. Each nodal agent has a set of available inputs $\boldsymbol{u}_i \in \mathbb{R}^{n_{u_i}}$ and measurements $\boldsymbol{y}_i(t) \in \mathbb{R}^{n_{y_i}}$. The mapping from the input to state vector can thus be written in the form $\boldsymbol{B}_u = \text{blkdiag}(\boldsymbol{B}_{u_1}, \ldots, \boldsymbol{B}_{u_N})$. The system nonlinearity can be expressed as $\boldsymbol{\phi}(\boldsymbol{x}) \in \mathbb{R}^{n_x}$ and $\boldsymbol{B}_f$ represents the distribution of the nonlinearities. The vectors $\boldsymbol{w}(t) \in \mathbb{R}^{n_w}$ and $\boldsymbol{v}(t) \in \mathbb{R}^{n_v}$ model unknown inputs and data perturbations. In summary, the system has n_x states, n_u control inputs, n_y output measurements, n_w unknown inputs, and n_v data perturbations, which are common in CPSs. Superscript j denotes the time-period and transitions in state-space matrices are assumed to be known.

The model (1) includes binary variables π_i, $i = 1, \ldots, N$, where $\pi_i = 1$ if the actuator of the i-th nodal agent is selected, and 0 otherwise. Similarly, we define binary variables γ_i, $i = 1, \ldots, N$, where $\gamma_i = 1$ if the sensor of the i-th nodal agent is selected, and 0 otherwise. Variables π_i and γ_i are organized in vectors $\boldsymbol{\pi}$ and $\boldsymbol{\gamma}$, i.e., $\boldsymbol{\Pi} = \text{blkdiag}(\pi_1\boldsymbol{I}_{n_{u_1}}, \ldots, \pi_N\boldsymbol{I}_{n_{u_N}})$ and $\boldsymbol{\Gamma} = \text{blkdiag}(\gamma_1\boldsymbol{I}_{n_{y_1}}, \ldots, \gamma_N\boldsymbol{I}_{n_{y_N}})$.

The formulations in this paper are building on SDP approaches for robust control and estimation routines; see [24], [26]. To set the stage, control and estimation formulations as SDPs are succinctly summarized in [1, Appendix E], where the system dynamics, controller/observer form, optimization variables, and the optimization problem are stated. The listed formulations are instrumental in formalizing the SaA selection problem since the LMIs share a similar structure. Many other control and estimation laws can fit directly into the proposed methodologies. The main contributions of this paper are detailed next.

- First, we show that a large array of optimal control and estimation problems with SaA selection share a similar level of computational complexity of solving optimization problems with MIBMIs (Section III).
- Second, we develop one-shot convex relaxation that produces a lower bound to the original problem with MIBMIs. Two successive convex approximations that yield upper bounds are also developed. Theoretical guarantees on the convergence of the convex relaxations and approximations are provided. The convex approximations draw from previous general methods [27], [28], but this paper develops specialized algorithms for the MIBMI problem structures that stem specifically from sensor and actuator selection. We also develop simple algorithms to recover the binary selection of SaAs, in addition to the state-feedback gains and performance indices (Sections IV–VI).
- Third, we include a general formulation that utilizes the Big-M method, thereby transforming the optimization problem that includes MIBMIs to a mixed-integer semidefinite program (MISDP)—this approach is detailed in the extended version of this manuscript [1]. Finally, comprehensive numerical examples are provided in Section VII. The numerical results corroborate the theoretical results, and the necessary assumptions needed to obtain convergence are satisfied. The next section presents the developed framework of time-varying SaA selection for uncertain dynamic systems.

III. Time-Varying SaA Selection with Various Metrics: A Unifying MIBMI Framework

In this section, we show that a plethora of control or estimation problems with time-varying SaA can be written as nonconvex optimization problems with MIBMIs. This observation considers different formulations pertaining to various observability and controllability metrics. In particular, replacing $\boldsymbol{B}_u$ with $\boldsymbol{B}_u\boldsymbol{\Pi}$ and $\boldsymbol{C}$ with $\boldsymbol{\Gamma}\boldsymbol{C}$ in the SDPs in [1, Appendix E] significantly increases the complexity of the optimization problem. This transforms the SDPs into nonconvex problems with MIBMIs, thereby necessitating the development of advanced optimization algorithms—the major contribution of this paper.

For concreteness, we only consider the actuator selection problem for robust $\mathcal{L}_\infty$ control of uncertain linear systems (see the second row of [1, Table III] or [29]), and leave the other SDP formulations with different control/estimation metrics as simple extensions. Focusing on the robust control with actuator selection, we can write the system dynamics as:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}^j\boldsymbol{x}(t) + \boldsymbol{B}_u^j\boldsymbol{\Pi}^j\boldsymbol{u}(t) + \boldsymbol{B}_w^j\boldsymbol{w}(t) \tag{2a}$$

$$\boldsymbol{z}(t) = \boldsymbol{C}_z^j\boldsymbol{x}(t) + \boldsymbol{D}_{wz}^j\boldsymbol{w}(t), \tag{2b}$$

where $\boldsymbol{\Pi}^j$ is binary matrix variable (cf. Section II) and $\boldsymbol{z}(t)$ is the control performance index. The time-varying sequence of selected actuators and stabilizing controllers is obtained as the solution of the following multi-period optimization problem:

$$\underset{\{\boldsymbol{S},\boldsymbol{Z},\zeta,\boldsymbol{\pi}\}^j}{\text{minimize}} \quad \sum_{j=1}^{T_f}(\eta+1)\zeta^j + \alpha_\pi^\top\boldsymbol{\pi}^j \tag{3a}$$

$$\text{subject to} \begin{bmatrix} \begin{matrix} \boldsymbol{A}^j\boldsymbol{S}^j + \boldsymbol{S}^j\boldsymbol{A}^{j\top} + \alpha\boldsymbol{S}^j \\ -\boldsymbol{B}_u^j\boldsymbol{\Pi}^j\boldsymbol{Z}^j - \boldsymbol{Z}^{j\top}\boldsymbol{\Pi}^j\boldsymbol{B}_u^{j\top} \end{matrix} & \boldsymbol{B}_w^j \\ \boldsymbol{B}_w^{j\top} & -\alpha\eta\boldsymbol{I} \end{bmatrix} \preceq \boldsymbol{O} \tag{3b}$$

$$\begin{bmatrix} -\boldsymbol{S}^j & \boldsymbol{O} & \boldsymbol{S}^j\boldsymbol{C}_z^{j\top} \\ \boldsymbol{O} & -\boldsymbol{I} & \boldsymbol{D}_{wz}^\top \\ \boldsymbol{C}_z^j\boldsymbol{S}^j & \boldsymbol{D}_{wz}^j & -\zeta^j\boldsymbol{I} \end{bmatrix} \preceq \boldsymbol{O} \tag{3c}$$

$$\boldsymbol{H}\boldsymbol{\pi} \leq \boldsymbol{h}, \ \boldsymbol{\pi} \in \{0,1\}^N. \tag{3d}$$

In (3), the optimization variables are matrices $(\boldsymbol{S}, \boldsymbol{Z}, \boldsymbol{Y})^j$, the actuator selection $\boldsymbol{\pi}^j$ (collected in vector $\boldsymbol{\pi}$ for all j), and the robust control index ζ^j for all $j \in \{1, \ldots, T_f\}$, where α and η are predefined positive constants [29]. Given the solution to (3), the stabilizing control law for the $\mathcal{L}_\infty$ problem can be written as $\boldsymbol{u}^*(t) = -\boldsymbol{Z}^{*j}(\boldsymbol{S}^{*j})^{-1}\boldsymbol{x}(t)$ for all $t \in [t_j, t_{j+1})$.

This guarantees that $\|\boldsymbol{z}(t)\|_2 \leq \sqrt{(\eta+1)\zeta^*}\|\boldsymbol{w}(t)\|_\infty$. Note that the $\mathcal{L}_\infty$ control LMIs represented here and in [1, Table III] are slightly different from the ones in [29], as some assumptions are made to simplify this robust control formulation.

The logistic constraint $\boldsymbol{H\pi} \leq \boldsymbol{h}$ couples the selected actuators across time periods, and is discussed in Appendix A. The optimization problem (3) includes MIBMIs due to the term $\boldsymbol{B}_u^j\boldsymbol{\Pi}^j\boldsymbol{Z}^j$. The bilinearity together with the integrality constraints bring about the need for specialized optimization methods. It should be emphasized that (3) is *not* a mixed-integer convex program. Therefore, general-purpose mixed-integer convex programming solvers are not applicable.

Interestingly, the design of the remaining controllers and observers in [1, Appendix E] largely share the optimization complexity of (3). It can be observed that *all* design problems in [1, Appendix E] feature MIBMIs with the form $\boldsymbol{B}_u\boldsymbol{\Pi Z}+\boldsymbol{Z}^\top\boldsymbol{\Pi B}_u^\top$ or a similar one. This simple idea signifies the impact of finding a solution to optimization problems with MIBMIs. In fact, many LMI formulations for control problems in [24] become MIBMIs when SaA selection is included. Using (3) as an exemplification for other problems with similar non-convexities, custom optimization algorithms to deal with such MIBMIs are proposed in the ensuing sections.

IV. From MIBMIs to BMIs

This section along with Sections V and VI develops a series of methods to deal with MIBMIs that all have the same starting point: Relaxing the integer constraints to continuous intervals. The resulting problem is still hard to solve, as it includes bilinear matrix inequalities (BMIs). For clarity, we consider a single-period version of the $\mathcal{L}_\infty$ problem with actuator selection, i.e., problem (3) with $T_f = 1$. This section presents some preparatory material that will be useful in the next sections. We start by considering the actuator selection problem with optimal value denoted by f^*.

$$f^* = \underset{\boldsymbol{S},\boldsymbol{Z},\zeta,\boldsymbol{\pi}}{\text{minimize}} \quad (\eta+1)\zeta + \boldsymbol{\alpha}_\pi^\top\boldsymbol{\pi} \tag{4a}$$

$$\text{subject to} \begin{bmatrix} \begin{array}{c}\boldsymbol{AS}+\boldsymbol{SA}^\top+\alpha\boldsymbol{S}\\ -\boldsymbol{B}_u\boldsymbol{\Pi Z}-\boldsymbol{Z}^\top\boldsymbol{\Pi B}_u^\top\end{array} & \boldsymbol{B}_w \\ \boldsymbol{B}_w^\top & -\alpha\eta\boldsymbol{I}\end{bmatrix} \preceq \boldsymbol{O} \tag{4b}$$

$$\begin{bmatrix} -\boldsymbol{S} & \boldsymbol{O} & \boldsymbol{SC}_z^\top \\ \boldsymbol{O} & -\boldsymbol{I} & \boldsymbol{D}_{wz}^\top \\ \boldsymbol{C}_z\boldsymbol{S} & \boldsymbol{D}_{wz} & -\zeta\boldsymbol{I}\end{bmatrix} \preceq \boldsymbol{O} \tag{4c}$$

$$\boldsymbol{H\pi} \leq \boldsymbol{h} \tag{4d}$$

$$\boldsymbol{\pi} \in \{0,1\}^N. \tag{4e}$$

The following standing assumption regarding the feasibility of (4) is made throughout the paper.

Assumption 1. *Problem (4) is feasible for $\pi_i = 1$, $i = 1,\ldots,N$ with constraints (4b), (4c), and (4d) holding as strict inequalities.*

The previous assumption essentially postulates that when all actuators are selected, then $\boldsymbol{S},\boldsymbol{Z},\zeta$ can be found so that matrix inequalities (4b) and (4c) hold with $\boldsymbol{O}$ on the left-hand side replaced by $-\epsilon\boldsymbol{I}$, and (4d) with $\boldsymbol{h}$ replaced by $\boldsymbol{h}-\epsilon'\boldsymbol{1}$, for sufficiently small $\epsilon > 0$ and $\epsilon' > 0$. Such a point does *not* need to be the optimal solution of (4); Assumption 1 only requires the existence of such a point in the feasible set. It follows from the previous discussion that finding such a point is a convex optimization problem.

The methods developed in Sections V and VI rely on substituting the integer constraint (4e) with the box constraint

$$\boldsymbol{0} \leq \boldsymbol{\pi} \leq \boldsymbol{1}. \tag{5}$$

Problem (4) with (4e) substituted by (5) can be written as

$$L = \underset{\boldsymbol{p}}{\text{minimize}} \quad f(\boldsymbol{p}) \tag{6a}$$

$$\text{subject to} \quad \mathcal{G}(\boldsymbol{p}) \preceq \boldsymbol{O} \tag{6b}$$

where the shorthand notation $\boldsymbol{p} = [\text{vec}(\boldsymbol{S})^\top \; \zeta \; \text{vec}(\boldsymbol{Z})^\top \; \boldsymbol{\pi})^\top]^\top$ denotes the optimization variables. The objective is $f(\boldsymbol{p}) = (\eta+1)\zeta + \boldsymbol{\alpha}_\pi^\top\boldsymbol{\pi}$, and $\mathcal{G}(\boldsymbol{p})$ is a matrix-valued function that includes the left-hand sides of (4b), (4c), (4d), and the two sides of (5), in a block diagonal form. Problem (6) has the general form of a nonlinear SDP [30]. The dimensions of $\boldsymbol{p}$ and $\mathcal{G}(\boldsymbol{p})$ are respectively given by $\boldsymbol{p} \in \mathbb{R}^d$ and $\mathcal{G}(\boldsymbol{p}) \in \mathbb{S}^\kappa$, where d and κ can be inferred from (4). The notation $D\mathcal{G}(\boldsymbol{p})$ is used for the differential of $\mathcal{G}(\boldsymbol{p})$ at $\boldsymbol{p}$, i.e., $D\mathcal{G}(\boldsymbol{p})$ maps a vector $\boldsymbol{q} \in \mathbb{R}^d$ to $\mathbb{S}^\kappa$ as follows

$$[D\mathcal{G}(\boldsymbol{p})]\boldsymbol{q} = \sum_{i=1}^d q_i \frac{\partial\mathcal{G}(\boldsymbol{p})}{\partial p_i}. \tag{7}$$

The optimal value serves as an index to formally compare the various formulations to be developed in the sequel. But comparison with respect to control metrics is also important, therefore, the resulting controllers are also evaluated in terms of the system closed-loop eigenvalues in the numerical tests of Section VII. The relationship between the optimal value of (4) and (6) is formalized in the following proposition.

Proposition 1. *With L denoting the optimal value of problem (6), it holds that $L \leq f^*$.*

Proof of Proposition 1: The proposition holds because (5) represents a relaxation of (4e). ■

Problem (6) is still hard to solve, because it contains the BMI (4b). Since the problem is nonconvex, several algorithms seek to find a stationary point of (6), instead of a globally optimal one. Before formally stating the definition of stationary point, the Lagrangian function of (6) is given next:

$$\mathcal{L}(\boldsymbol{p},\boldsymbol{\Lambda}) = f(\boldsymbol{p}) + \textbf{trace}[\boldsymbol{\Lambda}\mathcal{G}(\boldsymbol{p})], \tag{8}$$

where $\boldsymbol{\Lambda}$ is a Lagrange multiplier matrix. Stationary points of (6) abide by the following definition.

Definition 1. *A pair $(\boldsymbol{p}^*,\boldsymbol{\Lambda}^*)$ is a KKT point of (6), and $\boldsymbol{p}^*$ is a stationary point of (6), if the following hold: 1) Lagrangian optimality: $\nabla_{\boldsymbol{p}}\mathcal{L}(\boldsymbol{p}^*,\boldsymbol{\Lambda}) = \boldsymbol{0}$; 2) primal feasibility: $\mathcal{G}(\boldsymbol{p}^*) \preceq \boldsymbol{O}$; 3) dual feasibility: $\boldsymbol{\Lambda}^* \succeq \boldsymbol{O}$; and 4) complementary slackness: $\boldsymbol{\Lambda}^*\mathcal{G}(\boldsymbol{p}^*) = \boldsymbol{O}$.*

Conditions 1)–4) above are the KKT conditions for (6). These become necessary conditions that locally optimal solutions of (6) must satisfy, when appropriate constraint qualifications hold. Constraint qualifications are properties of the feasible set of an optimization problem. To make this concept concrete, two typical constraint qualifications are presented next [30].

Definition 2. *Problem* (6) *satisfies Slater's constraint qualification if there is a point* $\boldsymbol{p}^0 \in \mathbb{R}^d$ *satisfying* $\mathcal{G}(\boldsymbol{p}^0) \prec \boldsymbol{O}$*.*

Slater's constraint qualification guarantees zero duality gap for problems of the form (6) when $f(\boldsymbol{p})$ and $\mathcal{G}(\boldsymbol{p})$ are convex. Though $\mathcal{G}(\boldsymbol{p})$ is not convex for the problem at hand, we will use Slater's constraint qualification for convex approximations of (6) in the sequel. A constraint qualification useful for nonconvex nonlinear SDPs is given next.

Definition 3. *The Mangasarian-Fromovitz constraint qualification (MFCQ) holds at feasible point* $\boldsymbol{p}^0$ *if there exists a vector* $\boldsymbol{q} \in \mathbb{R}^d$ *such that*

$$\mathcal{G}(\boldsymbol{p}^0) + [D\mathcal{G}(\boldsymbol{p}^0)]\boldsymbol{q} \prec \boldsymbol{O}. \tag{9}$$

Under MFCQ, the KKT conditions become necessary for local optima of (6).

Lemma 1. *Let* $\boldsymbol{p}^*$ *be a locally optimal solution of* (6)*. Then under MFCQ, there exists a Lagrange multiplier matrix* $\boldsymbol{\Lambda}^*$ *that together with* $\boldsymbol{p}^*$ *satisfies the KKT conditions of Definition 1.*

Proof of Lemma 1: This result is typical in the literature of nonlinear SDPs; see [31, Sec. 4.1.3]. ■

The significance of Lemma 1 is that it characterizes the points which are local minima of (6). For future use, we mention next two refinements of the KKT conditions of Definition 1. Specifically, the complementary slackness condition implies that $\mathbf{rank}[\mathcal{G}(\boldsymbol{p}^*)] + \mathbf{rank}(\boldsymbol{\Lambda}^*) \leq \kappa$ [30, p. 307]. A stricter condition is defined as follows.

Definition 4. *A KKT point of* (6) *satisfies the strict complementarity if* $\mathbf{rank}[\mathcal{G}(\boldsymbol{p}^*)] + \mathbf{rank}(\boldsymbol{\Lambda}^*) = \kappa$*.*

To state the second condition, the definition of a feasible direction for problem (6) is provided next.

Definition 5. *Let* $\boldsymbol{p}^0$ *be a feasible point of* (6)*. A vector* $\boldsymbol{q} \in \mathbb{R}^d$ *is called a feasible direction for problem* (6) *at* $\boldsymbol{p}^0$ *if* $\boldsymbol{p}^0 + \varepsilon\boldsymbol{q}$ *is feasible for* (6) *for all sufficiently small* $\varepsilon > 0$*.*

The KKT conditions are of first order, i.e., they involve the gradient of the Lagrangian. The following definition states a second-order condition.

Definition 6. *Let* $(\boldsymbol{p}^*, \boldsymbol{\Lambda}^*)$ *be a KKT point of* (6)*. The second-order sufficiency condition holds for* $\boldsymbol{p}^*$ *if for all feasible directions* $\boldsymbol{q}$ *at* $\boldsymbol{p}^*$ *satisfying* $\nabla_{\boldsymbol{p}} f(\boldsymbol{p}^*)^\top \boldsymbol{q} = 0$*, it holds that* $\boldsymbol{q}^\top \nabla^2_{\boldsymbol{p}} \mathcal{L}(\boldsymbol{p}^*, \boldsymbol{\Lambda}^*)\boldsymbol{q} \geq \mu \|\boldsymbol{q}\|^2$*, for some* $\mu > 0$*.*

The second-order sufficiency condition will be useful for the convergence of one of the algorithms to solve BMIs in the sequel. Sections V and VI develop algorithms for solving problems of the form (6) that include BMIs. These algorithms typically return vectors $\boldsymbol{\pi}$ with non-integer, real entries. Based on the solutions produced by these algorithms, Appendix D details the procedure of actuator selection.

V. SDP Relaxations (SDP-R): A Lower Bound on (6)

This section develops a solver for BMI problems based on SDP relaxation of the BMI constraint. To this end, we introduce an additional optimization variable $\boldsymbol{G} = \boldsymbol{\Pi}\boldsymbol{Z}$. With this change of variables, $\boldsymbol{\Pi}\boldsymbol{Z}$ is replaced by $\boldsymbol{G}$ and $\boldsymbol{G}^\top$ replaces $\boldsymbol{Z}^\top\boldsymbol{\Pi}$ in (4b), while the constraint $\boldsymbol{G} = \boldsymbol{\Pi}\boldsymbol{Z}$ is added to the problem. Effectively, we have pushed the bilinearity into a new constraint $\boldsymbol{G} = \boldsymbol{\Pi}\boldsymbol{Z}$, which can actually be manipulated to much simpler constraints due to the diagonal structure of $\boldsymbol{\Pi}$.

Specifically, $\boldsymbol{Z}$ and $\boldsymbol{G}$ are stacks of N matrices

$$\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{Z}_1 \\ \vdots \\ \boldsymbol{Z}_N \end{bmatrix}, \quad \boldsymbol{G} = \begin{bmatrix} \boldsymbol{G}_1 \\ \vdots \\ \boldsymbol{G}_N \end{bmatrix} \tag{10}$$

where $\boldsymbol{Z}_i$ and $\boldsymbol{G}_i$ $(i = 1, \ldots, N)$ are both in $\mathbb{R}^{n_{u_i} \times n_x}$. Due to the diagonal structure of $\boldsymbol{\Pi}$, the constraint $\boldsymbol{G} = \boldsymbol{\Pi}\boldsymbol{Z}$ is equivalent to

$$\boldsymbol{G}_i = \pi_i \boldsymbol{Z}_i, \quad i = 1, \ldots, N. \tag{11}$$

Denote the (l, m) entries of matrices $\boldsymbol{Z}_i$ and $\boldsymbol{G}_i$ by $Z_{i,(l,m)}$ and $G_{i,(l,m)}$, respectively, where $l = 1, \ldots, n_{u_i}$ and $m = 1, \ldots, n_x$. Then, (11) is equivalent to the constraint

$$G_{i,(l,m)} = \pi_i Z_{i,(l,m)}, \quad i = 1, \ldots, N, \ l = 1, \ldots, n_{u_i}, \\ m = 1, \ldots, n_x. \tag{12}$$

It follows that problem (6) is equivalent to

$$L = \underset{\boldsymbol{S},\boldsymbol{Z},\zeta,\boldsymbol{\pi},\boldsymbol{G}}{\text{minimize}} \quad \zeta + \boldsymbol{\alpha}_\pi^\top \boldsymbol{\pi} \tag{13a}$$

$$\text{subject to} \begin{bmatrix} \boldsymbol{AS} + \boldsymbol{SA}^\top + \alpha\boldsymbol{S} & \\ -\boldsymbol{B}_u\boldsymbol{G} - \boldsymbol{G}^\top\boldsymbol{B}_u^\top & \boldsymbol{B}_w \\ \boldsymbol{B}_w^\top & -\alpha\eta\boldsymbol{I} \end{bmatrix} \preceq \boldsymbol{O} \tag{13b}$$

$$\text{(4c), (4d), (5), (12).} \tag{13c}$$

The next step is to relax (12) into an SDP constraint. To this end, define

$$\boldsymbol{E} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad \boldsymbol{e} = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}. \tag{14}$$

The SDP relaxation of (13) is provided in the next proposition.

Proposition 2. *The following SDP is a relaxation of* (13) *and yields a lower bound on the optimal value of* (6)*:*

$$\tilde{L} = \underset{\boldsymbol{S},\boldsymbol{Z},\zeta,\boldsymbol{\pi},\boldsymbol{G},\boldsymbol{V}}{\text{minimize}} \ (\eta + 1)\zeta + \boldsymbol{\alpha}_\pi^\top \boldsymbol{\pi} \tag{15a}$$

subject to

$$\begin{bmatrix} \boldsymbol{AS} + \boldsymbol{SA}^\top + \alpha\boldsymbol{S} & \\ -\boldsymbol{B}_u\boldsymbol{G} - \boldsymbol{G}^\top\boldsymbol{B}_u^\top & \boldsymbol{B}_w \\ \boldsymbol{B}_w^\top & -\alpha\eta\boldsymbol{I} \end{bmatrix} \preceq \boldsymbol{O} \tag{15b}$$

$$\mathbf{trace}\left(\boldsymbol{E}\boldsymbol{V}_{i,(l,m)}\right) - \boldsymbol{e}^\top \begin{bmatrix} G_{i,(l,m)} \\ Z_{i,(l,m)} \\ \pi_i \end{bmatrix} = 0 \tag{15c}$$

$$\left[\begin{array}{ccc|c} & \boldsymbol{V}_{i,(l,m)} & & \begin{matrix} G_{i,(l,m)} \\ Z_{i,(l,m)} \\ \pi_i \end{matrix} \\ \hline G_{i,(l,m)} & Z_{i,(l,m)} & \pi_i & 1 \end{array}\right] \succeq \boldsymbol{O} \tag{15d}$$

$$\forall\, i = 1,\dots,N,\ l = 1,\dots,n_{u_i},\ m = 1,\dots,n_x$$
$$\text{(4c), (4d), (5)} \tag{15e}$$

where $\boldsymbol{V}_{i,(l,m)} \in \mathbb{R}^{3\times 3}$ are auxiliary optimization variables collected in $\boldsymbol{V}$ for all i, l, and m. The optimal value of (15) has the property that $\tilde{L} \le L$. If, in addition, $\mathbf{rank}\left[\boldsymbol{V}_{i,(l,m)}\right] = 1$ *holds for all i, l, and m for the solution of (15), then $\tilde{L} = L$.*

Proof of Proposition 2: Introduce an auxiliary optimization variable $\boldsymbol{v}_{i,(l,m)}^\top = \begin{bmatrix} G_{i,(l,m)} & Z_{i,(l,m)} & \pi_i \end{bmatrix} \in \mathbb{R}^3$. One can verify that

$$\pi_i Z_{i,(l,m)} - G_{i,(l,m)} = \boldsymbol{v}_{i,(l,m)}^\top \boldsymbol{E} \boldsymbol{v}_{i,(l,m)} - \boldsymbol{e}^\top \boldsymbol{v}_{i,(l,m)}. \tag{16}$$

A relaxation trick can be used at this point. In particular, introduce an additional optimization variable $\boldsymbol{V}_{i,(l,m)} \in \mathbb{R}^{3\times 3}$ and the constraint $\boldsymbol{V}_{i,(l,m)} = \boldsymbol{v}_{i,(l,m)} \boldsymbol{v}_{i,(l,m)}^\top$. We have that

$$\begin{aligned} \boldsymbol{v}_{i,(l,m)}^\top \boldsymbol{E} \boldsymbol{v}_{i,(l,m)} &= \mathbf{trace}\left(\boldsymbol{v}_{i,(l,m)}^\top \boldsymbol{E} \boldsymbol{v}_{i,(l,m)}\right) \\ &= \mathbf{trace}\left(\boldsymbol{E} \boldsymbol{v}_{i,(l,m)} \boldsymbol{v}_{i,(l,m)}^\top\right) \\ &= \mathbf{trace}\left(\boldsymbol{E} \boldsymbol{V}_{i,(l,m)}\right). \end{aligned} \tag{17}$$

The previous development reveals that constraint (12) is equivalent to the constraint $\mathbf{trace}\left(\boldsymbol{E}\boldsymbol{V}_{i,(l,m)}\right) - \boldsymbol{e}^\top \boldsymbol{v}_{i,(l,m)} = 0$, which is linear in $\boldsymbol{V}_{i,(l,m)}$ and $\boldsymbol{v}_{i,(l,m)}$, as long as the constraint $\boldsymbol{V}_{i,(l,m)} = \boldsymbol{v}_{i,(l,m)} \boldsymbol{v}_{i,(l,m)}^\top$ is imposed, which is nonconvex. The constraint $\boldsymbol{V}_{i,(l,m)} = \boldsymbol{v}_{i,(l,m)} \boldsymbol{v}_{i,(l,m)}^\top$ is equivalent to

$$\begin{bmatrix} \boldsymbol{V}_{i,(l,m)} & \boldsymbol{v}_{i,(l,m)} \\ \boldsymbol{v}_{i,(l,m)}^\top & 1 \end{bmatrix} \succeq \boldsymbol{O},\ \mathbf{rank}(\boldsymbol{V}_{i,(l,m)}) = 1. \tag{18}$$

The rank constraint above is nonconvex, and by dropping it, we obtain the convex relaxation (15) of (13). As a relaxation of (13), its optimal value has the property that $\tilde{L} \le L$. ■

Proposition 2 asserts that $\tilde{L} = L$ if $\mathbf{rank}\left[\boldsymbol{V}_{i,(l,m)}\right] = 1$. Since the rank constraint is nonconvex, it is reasonable to consider surrogates for the rank in an effort to make the relaxation (15) tighter; one such convex surrogate is the nuclear norm of a matrix [32]. Thus, the constraint $\|\boldsymbol{V}_{i,(l,m)}\|_* \le 1$ can be added to promote smaller rank for $\boldsymbol{V}_{i,(l,m)}$; the optimal value of (15) is impacted as follows.

Corollary 1. *Let $\breve{L}$ be the optimal value of (15) with the added constraint $\|\boldsymbol{V}_{i,(l,m)}\|_* \le 1$. It holds that $\breve{L} \ge \tilde{L}$.*

Proof of Corollary 1: Adding the constraint restricts the feasible set of (15), yielding the stated relationship between the optimal values. ■

VI. Convex Approximations: An Upper Bound on (6)

The common thread between the previous and the present section is to replace the nonconvex feasible set given by constraints (4b), (4c), (4d), and (5) with convex sets. While the previous section relies on convex relaxations of the nonconvex feasible set, this section develops convex restrictions, i.e., replaces the nonconvex feasible set with a convex subset. The premise is to solve a series of optimization problems, in which the convex subset is improved. Thus, the algorithms in this section fall under the class of *successive convex approximations* (SCAs). Two SCA algorithms are developed in this section. Due to the convex restriction, the algorithms solve optimization problems that yield upper bounds for the optimal value L of problem (6).

Because the SCA algorithms rely on forming convex subsets of the feasible nonconvex set, they must be initialized at interior points of the nonconvex feasible set. The next proposition asserts that such points indeed exist under Assumption 1.

Proposition 3. *Under Assumption 1, problem (6) is strictly feasible, i.e., it satisfies Slater's constraint qualification.*

Proof of Proposition 3: Consider a point $\boldsymbol{p}^0$ that satisfies Assumption 1 (in particular, $\boldsymbol{\pi}^0 = \mathbf{1}$ holds). Constraints (4b), (4c), (4d), and (5) can be written in the form of a block diagonal matrix inequality (6b). The implication is that $\mathcal{G}(\boldsymbol{p}^0)$ is negative definite, i.e., all its eigenvalues are negative. By continuity of the eigenvalues as functions of the matrix elements [33, Appendix D], there is a ball of sufficiently small radius around $\boldsymbol{p}_0$ such that for all $\boldsymbol{p}$ is this ball, the eigenvalues of $\mathcal{G}(\boldsymbol{p})$ remain negative. Any point within the ball satisfying $\boldsymbol{\pi} < \mathbf{1}$ together with the associated $\boldsymbol{S}, \zeta, \boldsymbol{Z}$ yields a strictly feasible point for constraints (4b), (4c), (4d), and (5). ■

A. SCA using difference of convex functions (SCA-1)

The main idea is to replace (4b) with a surrogate convex inequality constraint. To this end, the left-hand side of (4b) is replaced by a convex function in the variables $\boldsymbol{Z}$, $\boldsymbol{\Pi}$, which is denoted by $\mathcal{C}(\boldsymbol{\Pi}, \boldsymbol{Z}; \boldsymbol{\Pi}_0, \boldsymbol{Z}_0)$, where $\boldsymbol{\Pi}_0, \boldsymbol{Z}_0$ are given matrices to be specified later. This approach has been investigated in the context of BMIs for control problems with bilinearities arising in output feedback control problems; see [27]. We first define the following linear function of $\boldsymbol{\Pi}, \boldsymbol{Z}$ with parameters $\boldsymbol{\Pi}_0, \boldsymbol{Z}_0$

$$\begin{aligned} &\mathcal{H}_{\text{lin}}(\boldsymbol{\Pi}, \boldsymbol{Z}; \boldsymbol{\Pi}_0, \boldsymbol{Z}_0) = \\ &\quad \boldsymbol{B}_u \boldsymbol{\Pi}_0 \boldsymbol{\Pi}_0^\top \boldsymbol{B}_u^\top - \boldsymbol{B}_u \boldsymbol{\Pi} \boldsymbol{\Pi}_0^\top \boldsymbol{B}_u^\top - \boldsymbol{B}_u \boldsymbol{\Pi}_0 \boldsymbol{\Pi}^\top \boldsymbol{B}_u^\top \\ &+ \boldsymbol{B}_u \boldsymbol{\Pi}_0 \boldsymbol{Z}_0^j - \boldsymbol{B}_u \boldsymbol{\Pi} \boldsymbol{Z}_0^j - \boldsymbol{B}_u \boldsymbol{\Pi}_0 \boldsymbol{Z}^j \\ &+ \boldsymbol{Z}_0^\top \boldsymbol{\Pi}_0 \boldsymbol{B}_u^\top - \boldsymbol{Z}_0^\top \boldsymbol{\Pi} \boldsymbol{B}_u^\top - \boldsymbol{Z}^\top \boldsymbol{\Pi}_0 \boldsymbol{B}_u^\top \\ &+ \boldsymbol{Z}_0^\top \boldsymbol{Z}_0 - \boldsymbol{Z}_0^\top \boldsymbol{Z} - \boldsymbol{Z}^\top \boldsymbol{Z}_0. \end{aligned} \tag{19}$$

The following proposition introduces a convex function that upper bounds the left-hand side of (4b).

Proposition 4. *It holds for all $\boldsymbol{\Pi}, \boldsymbol{Z}$ and $\boldsymbol{\Pi}_0, \boldsymbol{Z}_0$ that*

$$\begin{bmatrix} \boldsymbol{A}\boldsymbol{S} + \boldsymbol{S}\boldsymbol{A}^\top + \alpha \boldsymbol{S} \\ -\boldsymbol{B}_u \boldsymbol{\Pi} \boldsymbol{Z} - \boldsymbol{Z}^\top \boldsymbol{\Pi} \boldsymbol{B}_u^\top & \boldsymbol{B}_w \\ \boldsymbol{B}_w^\top & -\alpha\eta \boldsymbol{I} \end{bmatrix} \preceq \mathcal{C}(\boldsymbol{\Pi}, \boldsymbol{Z}; \boldsymbol{\Pi}_0, \boldsymbol{Z}_0) \tag{20}$$

where function $\mathcal{C}(\boldsymbol{\Pi}, \boldsymbol{Z}; \boldsymbol{\Pi}_0, \boldsymbol{Z}_0)$ is defined as follows and is convex in $\boldsymbol{\Pi}, \boldsymbol{Z}$:

$$\mathcal{C}(\cdot) = \begin{bmatrix} \boldsymbol{A}\boldsymbol{S} + \boldsymbol{S}\boldsymbol{A}^\top + \alpha \boldsymbol{S} \\ +\frac{1}{2}\left(\boldsymbol{B}_u \boldsymbol{\Pi} - \boldsymbol{Z}^\top\right)\left(\boldsymbol{B}_u \boldsymbol{\Pi} - \boldsymbol{Z}^\top\right)^\top \\ +\frac{1}{2}\mathcal{H}_{\text{lin}}(\boldsymbol{\Pi}, \boldsymbol{Z}; \boldsymbol{\Pi}_0, \boldsymbol{Z}_0) & \boldsymbol{B}_w \\ \boldsymbol{B}_w^\top & -\alpha\eta \boldsymbol{I} \end{bmatrix}. \tag{21}$$

The proof of Proposition 4 is included in Appendix B. Given this result, convex approximation of the BMI is obtained by replacing constraint (4b) with the convex constraint

$\mathcal{C}(\boldsymbol{\Pi}, \boldsymbol{Z}; \boldsymbol{\Pi}_0, \boldsymbol{Z}_0) \preceq \boldsymbol{O}$. The resulting problem has a restricted feasible set due to (20). Although $\mathcal{C}(\boldsymbol{\Pi}, \boldsymbol{Z}; \boldsymbol{\Pi}_0, \boldsymbol{Z}_0)$ is a convex function in $\boldsymbol{\Pi}$ and $\boldsymbol{Z}$, it is not linear in $\boldsymbol{\Pi}$ and $\boldsymbol{Z}$. Therefore, when we replace (4b) by the constraint $\mathcal{C}(\boldsymbol{\Pi}, \boldsymbol{Z}; \boldsymbol{\Pi}_0, \boldsymbol{Z}_0) \preceq \boldsymbol{O}$, a convex constraint is obtained, but not an LMI. Fortunately, the constraint $\mathcal{C}(\boldsymbol{\Pi}, \boldsymbol{Z}; \boldsymbol{\Pi}_0, \boldsymbol{Z}_0) \preceq \boldsymbol{O}$ can be equivalently written as an LMI as follows.

Lemma 2. *It holds that*

$$\mathcal{C}(\boldsymbol{\Pi}, \boldsymbol{Z}; \boldsymbol{\Pi}_0, \boldsymbol{Z}_0) \preceq \boldsymbol{O} \iff \mathcal{C}_s(\boldsymbol{\Pi}, \boldsymbol{Z}; \boldsymbol{\Pi}_0, \boldsymbol{Z}_0) = \begin{bmatrix} \begin{matrix} \boldsymbol{A}\boldsymbol{S} + \boldsymbol{S}\boldsymbol{A}^\top + \alpha \boldsymbol{S} \\ +\frac{1}{2}\mathcal{H}_{\text{lin}}(\boldsymbol{\Pi}, \boldsymbol{Z}; \boldsymbol{\Pi}_0, \boldsymbol{Z}_0) \end{matrix} & \frac{1}{\sqrt{2}}\left(\boldsymbol{B}_u\boldsymbol{\Pi} - \boldsymbol{Z}^\top\right) & \boldsymbol{B}_w \\ \frac{1}{\sqrt{2}}\left(\boldsymbol{B}_u\boldsymbol{\Pi} - \boldsymbol{Z}^\top\right)^\top & -\boldsymbol{I} & \boldsymbol{O} \\ \boldsymbol{B}_w^\top & \boldsymbol{O} & -\alpha\eta\boldsymbol{I} \end{bmatrix} \preceq \boldsymbol{O}. \tag{22}$$

Proof of Lemma 2: Applying the Schur complement to $\mathcal{C}(\cdot) \preceq \boldsymbol{O}$ yields the LMI $\mathcal{C}_s(\cdot) \preceq \boldsymbol{O}$. ■

To summarize, the convex approximation to (4) at $\boldsymbol{\Pi}_0, \boldsymbol{Z}_0$ is formed by replacing the integer constraints by the box constraints (5), and the BMI (4b) by the LMI constraint in (22). This problem is stated as follows:

$$\hat{L} = \underset{\boldsymbol{S},\boldsymbol{Z},\zeta,\boldsymbol{\pi}}{\text{minimize}} \quad (\eta+1)\zeta + \boldsymbol{\alpha}_\pi^\top \boldsymbol{\pi} \tag{23a}$$

$$\text{subject to} \quad \text{(4c), (4d), (5), (22).} \tag{23b}$$

Problem (23) is an SDP with optimal value denoted by $\hat{L}$, whose relationship with L is as follows.

Corollary 2. *The optimal value of the convex approximation (23) for all $\boldsymbol{\Pi}_0, \boldsymbol{Z}_0$ is an upper bound on the optimal value of (6), that is, $L \le \hat{L}$.*

Proof of Corollary 2: Due to (20) and (22), problem (23) has a restricted feasible set with respect to problem (6). ■

The convex approximation (23) depends on the point $\boldsymbol{\Pi}_0, \boldsymbol{Z}_0$, and can be successively improved. The main idea is to solve a sequence of convex approximations given by (23), where the values of $\boldsymbol{\Pi}_0, \boldsymbol{Z}_0$ for the next approximating problem are given by the solution of the previous problem.

Let $k = 1, 2, \ldots$ denote the index of the convex approximation to be solved, and let $\boldsymbol{S}_k, \zeta_k, \boldsymbol{\Pi}_k, \boldsymbol{Z}_k$ denote its solution. The k-th problem is obtained by adding a strictly convex regularizer to the objective (23a), which ensures that the problem has a unique solution. The k-th problem is thus

$$\hat{L}_k^{(1)} = \underset{\{\boldsymbol{S},\boldsymbol{Z},\zeta,\boldsymbol{\pi}\}}{\text{minimize}} \quad (\eta+1)\zeta + \boldsymbol{\alpha}_\pi^\top \boldsymbol{\pi} + \rho J_k \tag{24a}$$

$$\text{subject to} \quad \mathcal{C}_s(\boldsymbol{\Pi}, \boldsymbol{Z}; \boldsymbol{\Pi}_{k-1}, \boldsymbol{Z}_{k-1}) \preceq \boldsymbol{O} \tag{24b}$$

$$\text{(4c)}, \boldsymbol{H}\boldsymbol{\pi} \le \boldsymbol{h}, \quad 0 \le \boldsymbol{\pi} \le 1, \tag{24c}$$

where $J_k = \|\zeta - \zeta_{k-1}\|_2^2 + \|\boldsymbol{S} - \boldsymbol{S}_{k-1}\|_F^2 + \|\boldsymbol{Z} - \boldsymbol{Z}_{k-1}\|_F^2 + \|\boldsymbol{\Pi} - \boldsymbol{\Pi}_{k-1}\|_F^2$; the linearization point is given by $\boldsymbol{\Pi}_0 = \boldsymbol{\Pi}_{k-1}$, $\boldsymbol{Z}_0 = \boldsymbol{Z}_{k-1}$; ρ is the weight of the quadratic regularizers. For $k = 1$, the point $\boldsymbol{\Pi}_0, \boldsymbol{Z}_0$ can be selected as any interior point of (6); such is guaranteed to exist due to Proposition 3. Note that the regularization term ρJ_k penalizes the difference between the new solution and the previous. Upon algorithm convergence, the two successive solutions should be close to each other, which means that at optimality, the entire term ρJ_k should be close to zero.

Notice that for every k, problem (24) has the form of (6), but the objective is a strictly convex quadratic, and the constraint function is convex. The convergence is established in the following proposition.

Proposition 5. *Let $\boldsymbol{p}_k, \boldsymbol{\Lambda}_k$ denote a KKT point of (24). Suppose that the feasible set of (6) is bounded, and that the following hold for problem (24) for $k = 1, 2, 3, \ldots$*

i) Slater's constraint qualification holds.

ii) The Lagrange multiplier $\boldsymbol{\Lambda}_k$ is locally unique.

iii) Strict complementarity holds for the KKT point.

iv) The second-order sufficiency condition holds for the KKT point.

Then, the following are concluded:

a) It holds that $f(\boldsymbol{p}_k) \ge L$ and $L_k^{(1)} \ge L$ for $k = 1, 2, 3, \ldots$

b) The sequence $\{f(\boldsymbol{p}_k)\}_{k=1}^\infty$ is monotone decreasing, and converges to a limit $\hat{f}^{(1)} \ge L$.

c) Every limit point of the sequence $\{\boldsymbol{p}_k, \boldsymbol{\Lambda}_k\}_{k=1}^\infty$ is a KKT point of (6). If the set of KKT points of (6) is finite, then the entire sequence $\{\boldsymbol{p}_k, \boldsymbol{\Lambda}_k\}_{k=1}^\infty$ converges to a KKT point of (6).

The proof of Proposition 5 is included in Appendix B. Albeit some of the conditions of the previous proposition may be hard to verify in practice, we encountered no case where the SCA algorithm did not converge. In particular, we tested the algorithm on a variety of dynamic systems with varying sizes and conditions in Section VII.

B. Parametric SCA (SCA-2)

In this section, we depart from the difference of two convex functions approach used in the previous SCA, and use another approach to obtain an upper bound on the bilinear terms. The developments in this section follow the spirit of the methods presented in [28], where the authors investigate a new approach to solve BMIs that are often encountered in output feedback control problems.

First, let $\mathcal{F}_1(\boldsymbol{p})$ denote the left-hand side of (4b). Given $\boldsymbol{\Pi}_k$ and $\boldsymbol{Z}_k$, define $\Delta\boldsymbol{\Pi} = \boldsymbol{\Pi} - \boldsymbol{\Pi}_k$ and $\Delta\boldsymbol{Z} = \boldsymbol{Z} - \boldsymbol{Z}_k$. For any $\boldsymbol{Q} \in \mathbb{S}_{++}^{n_x}$, define further the following function:

$$\begin{aligned} \mathcal{K}_1(\boldsymbol{p}; \boldsymbol{p}_k, \boldsymbol{Q}) = &\begin{bmatrix} -\boldsymbol{B}_u\boldsymbol{\Pi}_k\boldsymbol{Z}_k - \boldsymbol{Z}_k^\top\boldsymbol{\Pi}_k\boldsymbol{B}_u^\top & \boldsymbol{B}_w \\ \boldsymbol{B}_w^\top & -\alpha\eta\boldsymbol{I} \end{bmatrix} \\ &+ \begin{bmatrix} \begin{matrix} \boldsymbol{A}\boldsymbol{S} + \boldsymbol{S}\boldsymbol{A}^\top + \alpha\boldsymbol{S} - \boldsymbol{B}_u\boldsymbol{\Pi}_k\Delta\boldsymbol{Z} \\ -\Delta\boldsymbol{Z}^\top\boldsymbol{\Pi}_k\boldsymbol{B}_u^\top - \boldsymbol{B}_u\Delta\boldsymbol{\Pi}\boldsymbol{Z}_k - \boldsymbol{Z}_k^\top\Delta\boldsymbol{\Pi}\boldsymbol{B}_u^\top \end{matrix} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix} \\ &+ \begin{bmatrix} \boldsymbol{B}_u\Delta\boldsymbol{\Pi}\boldsymbol{Q}\Delta\boldsymbol{\Pi}\boldsymbol{B}_u^\top + \Delta\boldsymbol{Z}^\top\boldsymbol{Q}^{-1}\Delta\boldsymbol{Z} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix}. \end{aligned} \tag{25}$$

Similar to Proposition 4, an upper bound on $\mathcal{F}_1(\boldsymbol{p})$ is provided by the next proposition.

Proposition 6. *It holds for all $\boldsymbol{p}$, $\boldsymbol{\Pi}_k, \boldsymbol{Z}_k$ and $\boldsymbol{Q} \in \mathbb{S}_{++}^{n_u}$ that*

$$\mathcal{F}_1(\boldsymbol{p}) \le \mathcal{K}_1(\boldsymbol{p}; \boldsymbol{p}_k, \boldsymbol{Q}). \tag{26}$$

The proof of Proposition 6 is included in Appendix B. The previous proposition suggests that constraint (4b) can be replaced by $\mathcal{K}_1(\boldsymbol{p};\boldsymbol{p}_k,\boldsymbol{Q}) \preceq \boldsymbol{O}$. There are two challenges to be addressed though. First, although $\mathcal{K}_1(\boldsymbol{p};\boldsymbol{p}_k,\boldsymbol{Q})$ is a convex function of $\boldsymbol{p}$, it is not linear, and thus constraint $\mathcal{K}_1(\boldsymbol{p};\boldsymbol{p}_k,\boldsymbol{Q}) \preceq \boldsymbol{O}$ is not an LMI. Second, although $\boldsymbol{Q}$ can remain constant, the approximation can be tightened if $\boldsymbol{Q}$ is allowed to be an optimization variable. The former challenge is addressed by Lemma 3, which is analogous to Lemma 2.

Lemma 3. *Constraint $\mathcal{K}_1(\boldsymbol{p};\boldsymbol{p}_k,\boldsymbol{Q}) \preceq \boldsymbol{O}$ is equivalent to*

$$\mathcal{K}(\boldsymbol{p};\boldsymbol{p}_k,\boldsymbol{Q}) = \begin{bmatrix} \boldsymbol{\Omega}(\boldsymbol{p};\boldsymbol{p}_k) & \boldsymbol{B}_w & \boldsymbol{B}_u\Delta\boldsymbol{\Pi} & \Delta\boldsymbol{Z}^\top \\ \boldsymbol{B}_w^\top & -\alpha\eta\boldsymbol{I} & \boldsymbol{O} & \boldsymbol{O} \\ \Delta\boldsymbol{\Pi}\boldsymbol{B}_u^\top & \boldsymbol{O} & -\boldsymbol{Q}^{-1} & \boldsymbol{O} \\ \Delta\boldsymbol{Z} & \boldsymbol{O} & \boldsymbol{O} & -\boldsymbol{Q} \end{bmatrix} \preceq \boldsymbol{O}, \tag{27}$$

where

$$\begin{aligned}\boldsymbol{\Omega}(\boldsymbol{p};\boldsymbol{p}_k) &= -\boldsymbol{B}_u\boldsymbol{\Pi}_k\boldsymbol{Z}_k - \boldsymbol{Z}_k^\top\boldsymbol{\Pi}_k\boldsymbol{B}_u^\top + \boldsymbol{A}\boldsymbol{S} + \boldsymbol{S}\boldsymbol{A}^\top + \alpha\boldsymbol{S} \\ &- \boldsymbol{B}_u\boldsymbol{\Pi}_k\Delta\boldsymbol{Z} - \Delta\boldsymbol{Z}^\top\boldsymbol{\Pi}_k\boldsymbol{B}_u^\top - \boldsymbol{B}_u\Delta\boldsymbol{\Pi}\boldsymbol{Z}_k - \boldsymbol{Z}_k^\top\Delta\boldsymbol{\Pi}\boldsymbol{B}_u^\top.\end{aligned}$$

Proof of Lemma 3: Use the Schur complement. ■

When $\boldsymbol{Q}$ is an optimization variable, function $\mathcal{K}(\boldsymbol{p};\boldsymbol{p}_k,\boldsymbol{Q})$ is not convex in $\boldsymbol{p}$ and $\boldsymbol{Q}$. An upper bound of $\mathcal{K}(\boldsymbol{p};\boldsymbol{p}_k,\boldsymbol{Q})$ that is linear in $\boldsymbol{p}$ and $\boldsymbol{Q}$ is given in Lemma 5. The following lemma gives a particular matrix property that becomes the foundation for Lemma 5.

Lemma 4. *Let $\mathcal{Q}(\boldsymbol{x}) : \mathbb{R}^n \rightarrow \mathbb{S}^m_{++}$ be a mapping defined as $\mathcal{Q}(\boldsymbol{x}) = \sum_{i=1}^n x_i\mathcal{Q}_i$ where $\mathcal{Q}_i \in \mathbb{S}^m$. The following inequality holds, where the right-hand side is the linearization of $-\mathcal{Q}(\boldsymbol{x})^{-1}$ around $\boldsymbol{x}_k$:*

$$-\mathcal{Q}(\boldsymbol{x})^{-1} \preceq -2\mathcal{Q}(\boldsymbol{x}_k)^{-1} + \mathcal{Q}(\boldsymbol{x}_k)^{-1}\mathcal{Q}(\boldsymbol{x})\mathcal{Q}(\boldsymbol{x}_k)^{-1}. \tag{28}$$

Lemma 5. *It holds for all $\boldsymbol{p}$, $\boldsymbol{Q} \in \mathbb{S}^{n_u}_{++}$, $\boldsymbol{\Pi}_k, \boldsymbol{Z}_k$, and $\boldsymbol{Q}_k \in \mathbb{S}^{n_u}_{++}$ that*

$$\mathcal{K}(\boldsymbol{p};\boldsymbol{p}_k,\boldsymbol{Q}) \preceq \mathcal{K}_s(\boldsymbol{p},\boldsymbol{Q};\boldsymbol{p}_k,\boldsymbol{Q}_k) \tag{29}$$

where $\mathcal{K}_s(\boldsymbol{p},\boldsymbol{Q};\boldsymbol{p}_k,\boldsymbol{Q}_k) =$

$$\begin{bmatrix} \boldsymbol{\Omega}(\boldsymbol{p};\boldsymbol{p}_k) & \boldsymbol{B}_w & \boldsymbol{B}_u\Delta\boldsymbol{\Pi}\boldsymbol{Q}_k & \Delta\boldsymbol{Z}^\top \\ \boldsymbol{B}_w^\top & -\alpha\eta\boldsymbol{I} & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{Q}_k\Delta\boldsymbol{\Pi}\boldsymbol{B}_u^\top & \boldsymbol{O} & -2\boldsymbol{Q}_k + \boldsymbol{Q} & \boldsymbol{O} \\ \Delta\boldsymbol{Z} & \boldsymbol{O} & \boldsymbol{O} & -\boldsymbol{Q} \end{bmatrix}. \tag{30}$$

The proofs of Lemmas 4 and 5 are included in Appendix B. Given these results, the constraint $\mathcal{K}_s(\boldsymbol{p},\boldsymbol{Q};\boldsymbol{p}_k,\boldsymbol{Q}_k) \preceq \boldsymbol{O}$ yields a restricted feasible set relative to constraint (3b). Similarly to Section VI-A, $k = 1, 2, 3, ...$ is the index of the optimization problem to be solved, and $\boldsymbol{p}_k, \boldsymbol{Q}_k$ denotes its solution. The k-th problem is an SDP and is stated as follows

$$\hat{L}_k^{(2)} = \underset{\{\boldsymbol{S},\boldsymbol{Z},\zeta,\boldsymbol{\pi},\boldsymbol{Q}\}}{\text{minimize}} \quad (\eta+1)\zeta + \boldsymbol{\alpha}_\pi^\top\boldsymbol{\pi} + \rho J_k \tag{31a}$$

$$\text{subject to} \quad \mathcal{K}_s(\boldsymbol{p},\boldsymbol{Q};\boldsymbol{p}_{k-1},\boldsymbol{Q}_{k-1}) \preceq \boldsymbol{O} \tag{31b}$$

$$c_1\boldsymbol{I} \preceq \boldsymbol{Q} \preceq c_2\boldsymbol{I}, \ -2\boldsymbol{Q}_{k-1} + \boldsymbol{Q} \preceq -c_3\boldsymbol{I} \tag{31c}$$

$$(4c), \ \boldsymbol{H}\boldsymbol{\pi} \leq \boldsymbol{h}, \ \ \boldsymbol{0} \leq \boldsymbol{\pi} \leq \boldsymbol{1}, \tag{31d}$$

where ρ, c_1, c_2, and c_3 are positive constants, and J_k is the same regularizer as the one in (24). Constraint (31c) guarantees that $\boldsymbol{Q}$ is positive definite, sequence $\{\boldsymbol{Q}_k\}_{k=1}^\infty$ is bounded, and that $-2\boldsymbol{Q}_k + \boldsymbol{Q}$, which appears as a diagonal block in (30) is negative definite for all k. Similar to the first convex approximation, the above problem can be initialized by letting $\{\boldsymbol{S}_0, \boldsymbol{Z}_0, \zeta_0, \boldsymbol{\pi}_0\}$ be any interior point of (6) and $\boldsymbol{Q}_0 = \boldsymbol{I}$. The algorithm convergence is characterized by the following proposition.

Proposition 7. *Assume that the MFCQ holds for every feasible point of (6) and that the sequence $\{\boldsymbol{p}_k\}_{k=1}^\infty$ is bounded. Then, the following are concluded:*

a) It holds that $f(\boldsymbol{p}_k) \geq L$ and $L_k^{(2)} \geq L$ for $k = 1, 2, \ldots$
b) The sequence $\{f(\boldsymbol{p}_k)\}_{k=1}^\infty$ is monotone decreasing, and converges to a limit $\hat{f}^{(2)} \geq L$.
c) Every limit point of $\{\boldsymbol{p}_k\}_{k=1}^\infty$ is a stationary point of (6).

The proof of Proposition 7 is included in Appendix B. Algorithm 1 in Appendix C provides the option to implement one of the two developed convex approximations [cf. (24) and (31)] sequentially until a maximum number of iterations or a stopping criterion are met. The next section compares the two approximations in terms of computational effort and their convergence claims.

C. Comparing the SCAs and Recovering the Integer Solutions

The first convex approximation is simpler to implement and involves a smaller number of SDP constraints and variables; see the difference in dimensions between constraints $\mathcal{K}_s(\boldsymbol{p},\boldsymbol{Q};\boldsymbol{z}_k,\boldsymbol{Q}_k) \preceq \boldsymbol{O}$ and $\mathcal{C}_s(\boldsymbol{\Pi},\boldsymbol{Z};\boldsymbol{\Pi}_k,\boldsymbol{Z}_k) \preceq \boldsymbol{O}$. In addition, constraint (31c) is added, and an extra variable $\boldsymbol{Q}$ is needed in (31). Both methods rely on constructing a series of feasible sets that are subsets of the original nonconvex feasible set in (6). Each produces a sequence of decreasing objective values $\{f(\boldsymbol{p}_k)\}_{k=1}^\infty$, yielding upper bounds on the optimal value of (6).

It is also worth noting that the first method requires a constraint qualification and additional assumptions on the KKT point to hold for each convex approximation problem k. Slater's constraint qualification is also an assumption in one of the earliest SCA methods for nonlinear programming [34]. On the other hand, the second method requires only the MFCQ to hold for the original nonconvex problem (6). Both methods have a boundedness assumption; the first method requires the feasible set of (6) to be bounded, the second method only the resulting sequence to be bounded. The boundedness assumption respectively guarantees the existence of at least one limit point of $\{\boldsymbol{p}_k\}_{k=1}^\infty$. Both methods enjoy the property that every limit point of $\{\boldsymbol{p}_k\}_{k=1}^\infty$ is a stationary point of (6).

Remark 1 (Existence of Local Minima)**.** *The stationarity is a necessary condition for local optimality (cf. Lemma 1). It is thus not guaranteed that the stationary point is locally optimal. In view of the fact that the methods attempt to solve a nonconvex problem, such convergence result is to be expected.*

The solutions obtained from (15), (24), and (31) produce a non-integer solution for the actuator selection problem. Since the objective is to determine a binary selection for the actuators, we present in this section a simple *slicing routine* that returns a binary selection given the solutions to the optimization problems in Sections V and VI. The algorithm is included and discussed in Appendix D.

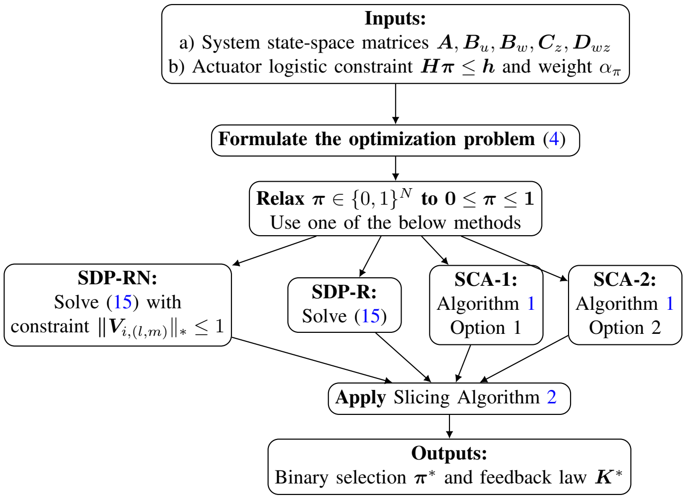
Inputs:
a) System state-space matrices $\boldsymbol{A}, \boldsymbol{B}_u, \boldsymbol{B}_w, \boldsymbol{C}_z, \boldsymbol{D}_{wz}$
b) Actuator logistic constraint $\boldsymbol{H\pi} \leq \boldsymbol{h}$ and weight α_π
Formulate the optimization problem (4)
Relax $\pi \in \{0,1\}^N$ to $0 \leq \pi \leq 1$
Use one of the below methods
SDP-RN:
Solve (15) with
constraint $\|\boldsymbol{V}_{i,(l,m)}\|_* \leq 1$
SDP-R:
Solve (15)
SCA-1:
Algorithm 1
Option 1
SCA-2:
Algorithm 1
Option 2
Apply Slicing Algorithm 2
Outputs:
Binary selection $\boldsymbol{\pi}^*$ and feedback law $\boldsymbol{K}^*$

Fig. 1. Flow chart showing the actuator selection and feedback control approach for the developed methods.

VII. Numerical Tests

In this work, we develop different computational methods to solve the actuator selection problem with a focus on the $\mathcal{L}_\infty$ control metric (4). Note that the extended manuscript [1] includes more thorough discussions and examples. The tested here methods are summarized as follows.

- *SDP-R*: An SDP relaxation providing a lower bound to the optimal solution of the problem with BMIs; see (15).
- *SDP-RN*: Same as SDP-R with the addition of the nuclear norm constraint to (15); see Corollary 1.
- *SCA-1* and *SCA-2*: Successive convex approximations producing upper bounds; see (23) and (31).
- These four methods (SDP-R, SDP-RN, SCA-1, SCA-2) are based on relaxing the integer constraints, and then followed by a slicing algorithm that returns an integer actuator selection and an upper bound on the optimization problem with MIBMIs (Algorithm 2). Fig. 1 shows a flowchart summarizing these four methods, where the feedback law $\boldsymbol{K}^*$ is obtained according to Algorithm 2.
- *Big-M*: The fifth method pertains to a formulation that transforms a problem with MIBMIs (4) into an MISDP via the Big-M method. This method is presented in the extended manuscript [1].

All the simulations are performed using MATLAB R2016b running on 64-bit Windows 10 with Intel Core i7-6700 CPU with base frequency of 3.4GHz and 16 GB of RAM. YALMIP [35] and its internal branch-and-bound solver are used as a modeling language and MOSEK [36] is used as the SDP solver for all methods.

A. Simulated Dynamic Systems, Parameters, and Setup

We use a randomly generated dynamic network from [37], [38] as a benchmark to test the presented methods. Additional numerical tests for another dynamic network are included in the extended manuscript [1]. The random dynamic network has the following structure

$$\dot{\boldsymbol{x}}_i = \underbrace{-\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}}_{\boldsymbol{A}_i} \boldsymbol{x}_i + \sum_{i \neq j} e^{-\alpha(i,j)} \boldsymbol{x}_j + \begin{bmatrix} 0 \\ 1 \end{bmatrix} (\boldsymbol{u}_i + \boldsymbol{w}_i),$$

where the coupling between nodes i and j is determined by the Euclidean distance $\alpha(i,j)$. These distances are unique for every N and randomly generated inside a box of size $N/5$.* The constraint $\boldsymbol{H\pi} \leq \boldsymbol{h}$ is represented as $\sum_{i=1}^{N} \boldsymbol{\pi}_i \geq \lfloor N/4 \rfloor$, where $\lfloor \cdot \rfloor$ denotes the floor function. We also set $\boldsymbol{\alpha}_\pi^\top = [1, \ldots, 1]$, that is all actuators have equal weight; $\alpha = 1$ and $\eta = 1$ (these constants appear in the LMIs). For SCA-1 and SCA-2, to obtain $\boldsymbol{S}_0$, ζ_0, and $\boldsymbol{Z}_0$, we initialize by assuming that $\boldsymbol{\Pi}_0 = 0.1\boldsymbol{I}_{n_u}$, and subsequently solving the $\mathcal{L}_\infty$ SDP with $\boldsymbol{S}_0 \succeq \epsilon_1 \boldsymbol{I}_{n_x}$ and $\zeta_0 \geq \epsilon_1$, where $\epsilon_1 = 10^{-8}$.

B. Results and Comparisons

Table I depicts the results after applying Algorithm 2 for SDP-R (15), SDP-RN, SCA-1 (23), and SCA-2 (31). Algorithm 2 is not applied to the Big-M solutions, as these solutions are binary. Table I presents the performance index $\sqrt{(\eta+1)\zeta}$, the total activated actuators $\sum_i^N \pi_i$, and the objective function value $f_{\text{final}} = (\eta+1)\zeta + \sum_{i=1}^N \pi_i$. The presented results for the Big-M method are for 300 iterations for the branch-and-bound solver of YALMIP. The maximum number of iterations is reached while the gap percentage is still between 1% for $N = 5$ all the way to 56% for $N = 50$ (the gap, provided in the caption of Table I, increases as N increases). Unfortunately, solving MISDPs would require weeks before the optimal solution (for larger values of N) is obtained and hence the choice of the default maximum iterations number of 300.

The boldfaced numbers in f_{final} column in Table I depict the method with the smallest objective function value. The Big-M/MISDP formulation has been proposed before for SaA selection in linear systems [12], [20]. While Big-M yields the smallest f_{final} in some cases, the other methods (SDP-R, SDP-RN, SCA-1, SCA-2) yield better objective values, while requiring significantly less computational time—often orders of magnitude smaller than Big-M. In particular, Table II shows the computational time (in seconds) for the five methods. Since SDP-R solves only a single SDP, it is expected to be computationally more efficient than the other methods—this can be observed from Table II. In addition, and since SCA-1 includes a smaller number of constraints and variables than SCA-2 (see Section VI-C), the former requires less computational time in several simulations. However, there are instances where the SCAs require less computational time the than the semidefinite relaxations (SDP-RN and SDP-R). The unifying theme here is that relaxing the integer constraints and using the convex approximations and relaxations is a good alternative to computationally costly MISDPs. In addition, we emphasize that although some methods can yield the same number of activated actuators, the specific activated actuators from each method can be significantly different. More discussions and numerical simulations are included in [1].

*Note that in these tests, we made the individual $\boldsymbol{A}_i$ matrix for each subsystem to be stable (in comparison with [37], [38] where $\boldsymbol{A}_i$ is unstable), so that the total number of unstable eigenvalues is smaller for the dynamic network ($\boldsymbol{A}$ still has few unstable eigenvalues). Keeping the same structure for the $\boldsymbol{A}$ matrix as in [37], [38] yields the trivial solution of activating all actuators which is needed to guarantee an $\mathcal{L}_\infty$-stable performance—and hence the modification in the state-space matrix $\boldsymbol{A}$.

TABLE I

FINAL RESULTS AFTER RUNNING ALGORITHM 2 TO RECOVER THE BINARY ACTUATOR SELECTION AND THE ACTUAL SYSTEM PERFORMANCE FOR THE SYSTEM WITH RANDOM NETWORK. THE BOLDFACED NUMBERS DESCRIBE THE METHOD THAT OUTPERFORMED OTHER METHODS (THE MISDP SOLVER ON YALMIP THAT IMPLEMENTS THE BIG-M APPROACH IS TERMINATED AFTER 300 BRANCH-AND-BOUND ITERATIONS). FOR THE BIG-M METHOD, THE GAP PERCENTAGES ARE $1.2, 10.19, 25.31, 44.90, 47.33, 51.48, 52.63, 52.21, 53.54, 55.91$ FOR $N = 5, 10, \dots, 50$.

N	Performance Index $\sqrt{(\eta+1)\zeta}$					Total Activated Actuators $\sum_{i=1}^{N} \pi_i$					$f_{\text{final}} = (\eta+1)\zeta + \sum_{i=1}^{N} \pi_i$				
	Big-M	SDP-RN	SDP-R	SCA-1	SCA-2	Big-M	SDP-RN	SDP-R	SCA-1	SCA-2	Big-M	SDP-RN	SDP-R	SCA-1	SCA-2
5	1.200	1.133	1.165	1.003	0.986	2	3	3	3	3	**3.441**	4.284	4.357	4.005	3.971
10	1.496	1.291	1.280	1.117	1.360	3	5	5	7	5	**5.239**	6.668	6.639	8.248	6.849
15	1.459	1.274	1.397	1.245	1.378	6	12	8	13	10	**8.129**	13.622	9.952	14.549	11.898
20	1.079	1.198	1.379	1.243	1.320	15	19	14	17	18	16.165	20.435	**15.903**	18.545	19.742
25	1.232	0.001	1.082	0.911	5.015	19	25	23	24	17	**20.517**	25.000	24.170	24.831	42.148
30	1.173	1.647	0.995	1.554	2.152	24	24	28	28	16	25.376	26.711	28.991	30.415	**20.631**
35	1.343	1.578	2.028	1.277	1.255	30	34	23	31	32	31.804	36.489	**27.112**	32.632	33.575
40	1.201	1.280	1.605	1.284	1.287	35	38	28	38	33	36.442	39.639	**30.576**	39.649	34.656
45	1.258	1.640	1.086	1.362	1.548	40	36	44	40	36	41.583	38.689	45.180	41.854	**38.396**
50	0.980	2.236	1.283	1.389	2.426	45	32	43	47	39	45.961	**37.001**	44.646	48.930	44.885

TABLE II

CPU TIME FOR THE DIFFERENT METHODS WITH VARIOUS VALUES FOR THE NUMBER OF NODES N FOR THE RANDOM DYNAMIC NETWORK.

N	Big-M	SDP-RN	SDP-R	SCA-1	SCA-2
5	3.92	1.84	**1.45**	2.10	1.84
10	87.47	3.73	**1.36**	2.97	2.58
15	369.49	14.18	**3.36**	10.49	8.86
20	1337.97	50.26	**19.35**	33.82	45.17
25	3774.93	142.73	90.88	120.51	**80.12**
30	9222.35	317.55	**281.83**	314.63	127.21
35	19760.87	853.73	**303.23**	615.68	674.92
40	41038.02	1901.40	**822.92**	1673.95	1258.57
45	76166.24	3103.24	3201.57	2695.33	**2192.87**
50	131035.62	4107.22	4441.03	5785.46	**4096.90**

C. Extensions to Sensor Selection for Nonlinear Systems

In this paper, we only use the $\mathcal{L}_\infty$ control problem with actuator selection to exemplify how the proposed methods can provide insights into the solution of MIBMIs. We emphasize that all other CPS dynamics and control/estimation formulations (see [1, Appendix E]) with SaA selection can be solved using the methods we develop here. For example, consider the sensor selection alongside the state estimator design problem for nonlinear systems $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}_u\boldsymbol{u} + \boldsymbol{\phi}(\boldsymbol{x}), \quad \boldsymbol{y} = \boldsymbol{\Gamma}\boldsymbol{C}\boldsymbol{x}$ where $\boldsymbol{\phi}(\boldsymbol{x})$ is the vector of nonlinearities with Lipschitz constant $\beta > 0$ and $\boldsymbol{\Gamma}$ is the binary sensor selection variable [cf. (1)]. By considering the last SDP in [1, Table IV], the sensor selection with observer design problem becomes:

$$\begin{aligned}
&\underset{\boldsymbol{\Gamma},\boldsymbol{P},\boldsymbol{Y},\kappa}{\text{minimize}} \;\; \alpha_\gamma^\top \boldsymbol{\gamma} \\
&\text{subject to} \;\; \begin{bmatrix} \begin{matrix}\boldsymbol{A}^\top\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} - \boldsymbol{Y}\boldsymbol{\Gamma}\boldsymbol{C} \\ -\boldsymbol{C}^\top\boldsymbol{\Gamma}\boldsymbol{Y}^\top + \alpha\boldsymbol{P} + \kappa\beta^2\boldsymbol{I}\end{matrix} & \boldsymbol{P} \\ \boldsymbol{P} & -\kappa\boldsymbol{I} \end{bmatrix} \preceq \boldsymbol{O} \\
&\qquad\qquad \boldsymbol{H}\boldsymbol{\gamma} \leq \boldsymbol{h}, \; \boldsymbol{\gamma} \in \{0,1\}^N,
\end{aligned}$$

which can be solved using the developed methods in the paper. This formulation yields observer gain $\boldsymbol{L}^* = (\boldsymbol{P}^*)^{-1}\boldsymbol{Y}^*$ that guarantees the asymptotic stability of the estimation error $\boldsymbol{e}(t) = \boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t)$ from $\dot{\hat{\boldsymbol{x}}} = \boldsymbol{A}\hat{\boldsymbol{x}} + \boldsymbol{B}_u\boldsymbol{u} + \boldsymbol{L}^*(\boldsymbol{y} - \hat{\boldsymbol{y}}) + \boldsymbol{\phi}(\hat{\boldsymbol{x}}), \;\; \hat{\boldsymbol{y}} = \boldsymbol{\Gamma}^*\boldsymbol{C}\hat{\boldsymbol{x}}$ with minimal number of sensors $\boldsymbol{\Gamma}^*$.

VIII. SUMMARY AND FUTURE WORK

This paper puts forth a framework to solve SaA selection problems for uncertain CPSs with various control and estimation metrics. Given the widely popular SDP formulations of various control and estimation problems (without SaA selection), we present various techniques that aim to recover, approximate, or bound the optimal solution to the combinatorial SaA selection problem via convex programming. While the majority of prior art focuses on specific metrics or dynamics, the objective of this paper is to present a unifying framework that streamlines the problem of time-varying SaA selection in uncertain and potentially nonlinear CPSs.

The developed methods in the paper have their limitation. First, the transition in the state-space matrices needs to be given before the time-varying actuator selection problem is solved. This narrows the scope of the actuator selection problem. In future work, we plan to study the actuator selection problem when the topological evolution is unknown, yet bounded. In particular, we plan to explore solutions to the SaA selection problem if the state-space matrix $\boldsymbol{A}$ includes bounded perturbations that mimic the evolution in the CPS topology.

In future work, we also plan to study the following related research problems: (1) applications to selection of distributed generation in electric power networks with frequency-performance guarantees; (2) customized branch-and-bound and cutting plane methods that can improve the performance of the Big-M method; and (3) theoretical analysis of the tightness of the lower and upper bounds resulting from the convex formulations in this paper.

REFERENCES

[1] A. F. Taha, N. Gatsis, T. Summers, and S. Nugroho, "Time-varying sensor and actuator selection for uncertain cyber-physical systems," 2018. [Online]. Available: https://arxiv.org/pdf/1708.07912.pdf

[2] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, "Controllability of complex networks," *Nature*, vol. 473, no. 7346, pp. 167–173, 2011.

[3] J. Ruths and D. Ruths, "Control profiles of complex networks," *Science*, vol. 343, no. 6177, pp. 1373–1376, 2014.

[4] A. Olshevsky, "Minimal controllability problems," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 3, pp. 249–258, 2014.

[5] S. Pequito, S. Kar, and A. 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, 2016.

[6] F. Pasqualetti, S. Zampieri, and F. Bullo, "Controllability metrics, limitations and algorithms for complex networks," *Control of Network Systems, IEEE Transactions on*, vol. 1, no. 1, pp. 40–52, 2014.

[7] 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.

[8] T. Summers, F. 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, 2016.

[9] V. Tzoumas, M. A. Rahimian, G. 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, 2016.

[10] Y. Zhao, F. Pasqualetti, and J. Cortés, "Scheduling of control nodes for improved network controllability," in *Proc. IEEE Conference on Decision and Control*, 2016, pp. 1859–1864.

[11] E. Nozari, F. Pasqualetti, and J. Cortes, "Time-varying actuator scheduling in complex networks," *arXiv preprint arXiv:1611.06485*, 2016.

[12] P. V. Chanekar, N. Chopra, and S. Azarm, "Optimal actuator placement for linear systems with limited number of actuators," in *Proc. American Control Conference*, May 2017, pp. 334–339.

[13] B. Polyak, M. Khlebnikov, and P. Shcherbakov, "An LMI approach to structured sparse feedback design in linear control systems," in *Proc. European Control Conference*, July 2013, pp. 833–838.

[14] N. K. Dhingra, M. R. Jovanović, and Z.-Q. Luo, "An ADMM algorithm for optimal sensor and actuator selection," in *Proc. IEEE Conference on Decision and Control*, 2014, pp. 4039–4044.

[15] U. Münz, M. Pfister, and P. Wolfrum, "Sensor and actuator placement for linear systems based on H_2 and H_∞ optimization," *IEEE Transactions on Automatic Control*, vol. 59, no. 11, pp. 2984–2989, 2014.

[16] A. Argha, S. W. Su, and A. Savkin, "Optimal actuator/sensor selection through dynamic output feedback," in *Proc. IEEE 55th Conference on Decision and Control*, Dec. 2016, pp. 3624–3629.

[17] T. Summers, "Actuator placement in networks using optimal control performance metrics," in *Proc. IEEE Conference on Decision and Control*, 2016, pp. 2703–2708.

[18] T. Summers and J. Ruths, "Performance bounds for optimal feedback control in networks," *arXiv preprint arXiv:1707.04528*, 2017.

[19] 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.

[20] J. A. Taylor, N. Luangsomboon, and D. Fooladivanda, "Allocating sensors and actuators via optimal estimation and control," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 3, pp. 1060–1067, May 2017.

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

[22] F. W. Kong, D. Kuhn, and B. Rustem, "A cutting-plane method for mixed-logical semidefinite programs with an application to multi-vehicle robust path planning," in *Proc. 49th IEEE Conference on Decision and Control*, Dec. 2010, pp. 1360–1365.

[23] A. Haber, F. Molnar, and A. E. Motter, "State observation and sensor selection for nonlinear networks," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 2, 2018.

[24] S. P. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. SIAM, 1994, vol. 15.

[25] A. F. Taha, N. Gatsis, T. Summers, and S. Nugroho, "Actuator selection for cyber-physical systems," in *Proc. American Control Conference*, May 2017, pp. 5300–5305.

[26] J. G. VanAntwerp and R. D. Braatz, "A tutorial on linear and bilinear matrix inequalities," *Journal of Process Control*, vol. 10, no. 4, pp. 363–385, 2000.

[27] Q. T. Dinh, S. Gumussoy, W. Michiels, and M. Diehl, "Combining convex–concave decompositions and linearization approaches for solving bmis, with application to static output feedback," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1377–1390, 2012.

[28] D. Lee and J. Hu, "A sequential parametric convex approximation method for solving bilinear matrix inequalities," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, Dec 2016, pp. 1965–1970.

[29] T. A. Pancake, "Analysis and control of uncertain/nonlinear systems in the presence of bounded disturbance inputs," Ph.D. dissertation, Purdue University, 2000.

[30] A. Shapiro, "First and second order analysis of nonlinear semidefinite programs," *Mathematical Programming*, vol. 77, no. 1, pp. 301–320, 1997.

[31] A. Shapiro and K. Scheinberg, "Duality and optimality conditions," in *Handbook of Semidefinite Programming*, H. Wolkowicz, R. Saigal, and L. Vandenberghe, Eds. Springer, 2000, ch. 4, pp. 67–110.

[32] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM review*, vol. 52, no. 3, pp. 471–501, 2010.

[33] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed. New York, NY: Cambridge University Press, 2013.

[34] B. R. Marks and G. P. Wright, "A general inner approximation algorithm for nonconvex mathematical programs," *Operations Research*, vol. 26, no. 4, pp. 681–683, 1978.

[35] J. Löfberg, "YALMIP: A toolbox for modeling and optimization in matlab," in *Proc. IEEE Int. Symp. Computer Aided Control Systems Design*, 2004, pp. 284–289.

[36] A. Mosek, "The mosek optimization toolbox for matlab manual," *Version 7.1 (Revision 28)*, p. 17, 2015.

[37] M. Jovanovic, "Network with 100 unstable nodes." [Online]. Available: http://people.ece.umn.edu/∼mihailo/software/lqrsp/dist_sys.html

[38] N. Motee and A. Jadbabaie, "Optimal control of spatially distributed systems," *IEEE Transactions on Automatic Control*, vol. 53, no. 7, pp. 1616–1629, Aug 2008.

[39] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[40] K. B. Petersen, M. S. Pedersen *et al.*, "The matrix cookbook," Technical University of Denmark, 2012.

Ahmad F. Taha is an assistant professor with the Department of Electrical and Computer Engineering at the University of Texas, San Antonio. He received the B.E. and Ph.D. degrees in Electrical and Computer Engineering from the American University of Beirut, Lebanon in 2011 and Purdue University, West Lafayette, Indiana in 2015. He Dr. Taha is interested in understanding how complex cyber-physical systems (CPS) operate, behave, and *misbehave*. His research focus includes optimization, control, and security of CPSs with applications to power, water, and transportation networks.

Nikolaos Gatsis received the Diploma degree in Electrical and Computer Engineering from the University of Patras, Greece, in 2005 with honors. He completed his graduate studies at the University of Minnesota, where he received the M.Sc. degree in Electrical Engineering in 2010, and the Ph.D. degree in Electrical Engineering with minor in Mathematics in 2012. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering at the University of Texas at San Antonio. His research focuses on optimal and secure operation of smart power grids and other critical infrastructures, including water distribution networks and the global positioning system. He has co-organized symposia in the area of smart grids in IEEE GlobalSIP 2015 and IEEE GlobalSIP 2016. He has also served as a co-guest editor for a special issue of the IEEE Journal on Selected Topics in Signal Processing on Critical Infrastructures.

Tyler Summers is an Assistant Professor of Mechanical Engineering at the University of Texas at Dallas. Prior to joining UT Dallas, he was an ETH Postdoctoral Fellow at the Automatic Control Laboratory at ETH Zurich from 2011 to 2015. He received a B.S. degree in Mechanical Engineering from Texas Christian University in 2004 and an M.S. and PhD degree in Aerospace Engineering with emphasis on feedback control theory at the University of Texas at Austin in 2007 and 2010, respectively. He was a Fulbright Postgraduate Scholar at the Australian National University in Canberra, Australia in 2007-2008. His research interests are in feedback control, optimization, and learning in complex dynamical networks, with applications to electric power networks and distributed robotics.

Sebastian A. Nugroho was born in Yogyakarta, Indonesia in 1990. He received the B.S. and M.S. degrees in Electrical Engineering from Institut Teknologi Bandung (ITB), Indonesia in 2012 and 2014. He is currently a graduate research assistant and a Ph.D. candidate with the ECE department at the University of Texas, San Antonio. His research interests are in control theory and engineering optimization with applications to cyber-physical systems.

Appendix A
Actuator Selection: The Logistic Constraints

The constraint $\boldsymbol{H}\boldsymbol{\pi} \leq \boldsymbol{h}$ couples the selected actuators across time periods, and is a linear logistic constraint that includes the following scenarios.

- Activation and deactivation of SaAs in a specific time-period j. For example, if actuator i cannot be selected at period j, we set $\pi_i^j \leq 0$.
- If actuator k is allowed to be selected only after actuator i is selected at period j, we set $\pi_k^{j+1} \leq \pi_i^j$, for $j = 1, \ldots, T_f$.
- If actuator k must be deselected after actuator i is selected at period j, we set $\pi_k^{j+1} \leq 1 - \pi_i^j$, for $j = 1, ...T_f$.
- Upper and lower bounds on the total number of active SaAs per period can be accounted for.
- Other constraints such as minimal number of required active actuators in a certain region of the CPS, and unit commitment constraints that are obtained from solutions day-ahead planning problems, can be included.

Appendix B
Proofs of Various Results

Proof of Proposition 4: To construct the upper bound (20), the bilinear term is written as

$$-\boldsymbol{B}_u\boldsymbol{\Pi}\boldsymbol{Z} - \boldsymbol{Z}^\top\boldsymbol{\Pi}\boldsymbol{B}_u^\top = \frac{1}{2}\Big[\left(\boldsymbol{B}_u\boldsymbol{\Pi} - \boldsymbol{Z}^\top\right)\left(\boldsymbol{B}_u\boldsymbol{\Pi} - \boldsymbol{Z}^\top\right)^\top - \left(\boldsymbol{B}_u\boldsymbol{\Pi} + \boldsymbol{Z}^\top\right)\left(\boldsymbol{B}_u\boldsymbol{\Pi} + \boldsymbol{Z}^\top\right)^\top\Big] \quad (32)$$

The term $\left(\boldsymbol{B}_u\boldsymbol{\Pi} - \boldsymbol{Z}^\top\right)\left(\boldsymbol{B}_u\boldsymbol{\Pi} - \boldsymbol{Z}^\top\right)^\top$ is convex in $\boldsymbol{Z}$ and $\boldsymbol{\Pi}$ since it comes from an affine transformation of the domain of a convex function [39, Example 3.48]. The term

$$\mathcal{H}(\boldsymbol{\Pi}, \boldsymbol{Z}) := -\left(\boldsymbol{B}_u\boldsymbol{\Pi} + \boldsymbol{Z}^\top\right)\left(\boldsymbol{B}_u\boldsymbol{\Pi} + \boldsymbol{Z}^\top\right)^\top$$

is concave in $\boldsymbol{Z}$ and $\boldsymbol{\Pi}$. We can therefore invoke the fact that the first-order Taylor approximation of a concave function (at any point) is a global over-estimator of the function. Let $\boldsymbol{\Pi}_0, \boldsymbol{Z}_0$ be the linearization point, and let $\mathcal{H}_{\text{lin}}(\boldsymbol{\Pi}, \boldsymbol{Z}; \boldsymbol{\Pi}_0, \boldsymbol{Z}_0)$ denote the linearization of $\mathcal{H}(\boldsymbol{\Pi}, \boldsymbol{Z})$ at the point $(\boldsymbol{\Pi}_0, \boldsymbol{Z}_0)$. It holds that

$$\mathcal{H}(\boldsymbol{\Pi}, \boldsymbol{Z}) \preceq \mathcal{H}_{\text{lin}}(\boldsymbol{\Pi}, \boldsymbol{Z}; \boldsymbol{\Pi}_0, \boldsymbol{Z}_0) \quad (33)$$

for all $\boldsymbol{\Pi}_0, \boldsymbol{Z}_0$ and $\boldsymbol{\Pi}, \boldsymbol{Z}$.

The linearization can be derived by substituting $\boldsymbol{\Pi} = \boldsymbol{\Pi}_0 + (\boldsymbol{\Pi} - \boldsymbol{\Pi}_0)$ and $\boldsymbol{Z} = \boldsymbol{Z}_0 + (\boldsymbol{Z} - \boldsymbol{Z}_0)$ into $\mathcal{H}(\boldsymbol{\Pi}, \boldsymbol{Z})$ and ignoring all second-order terms that involve $(\boldsymbol{\Pi} - \boldsymbol{\Pi}_0)$ and $(\boldsymbol{Z} - \boldsymbol{Z}_0)$. The result is (19). Combining (19) with (33) and (32), we conclude that the left-hand side of (4b) is upper bounded as

$$\begin{bmatrix} \begin{array}{c} \boldsymbol{A}\boldsymbol{S} + \boldsymbol{S}\boldsymbol{A}^\top + \alpha\boldsymbol{S} \\ -\boldsymbol{B}_u\boldsymbol{\Pi}\boldsymbol{Z} - \boldsymbol{Z}^\top\boldsymbol{\Pi}\boldsymbol{B}_u^\top \end{array} & \boldsymbol{B}_w \\ \boldsymbol{B}_w^\top & -\alpha\eta\boldsymbol{I} \end{bmatrix} \preceq \begin{bmatrix} \begin{array}{c} \boldsymbol{A}\boldsymbol{S} + \boldsymbol{S}\boldsymbol{A}^\top + \alpha\boldsymbol{S} \\ +\frac{1}{2}\left(\boldsymbol{B}_u\boldsymbol{\Pi} - \boldsymbol{Z}^\top\right)\left(\boldsymbol{B}_u\boldsymbol{\Pi} - \boldsymbol{Z}^\top\right)^\top \\ +\frac{1}{2}\mathcal{H}_{\text{lin}}(\boldsymbol{\Pi}, \boldsymbol{Z}; \boldsymbol{\Pi}_0, \boldsymbol{Z}_0) \end{array} & \boldsymbol{B}_w \\ \boldsymbol{B}_w^\top & -\alpha\eta\boldsymbol{I} \end{bmatrix}. \quad (34)$$

This can be obtained using the fact that

$$\begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{B} \\ \boldsymbol{B}^\top & \boldsymbol{C} \end{bmatrix} \succeq \boldsymbol{O}, \boldsymbol{A}_2 \succeq \boldsymbol{A}_1 \Longrightarrow \begin{bmatrix} \boldsymbol{A}_2 & \boldsymbol{B} \\ \boldsymbol{B}^\top & \boldsymbol{C} \end{bmatrix} \succeq \boldsymbol{O}$$

which can be proved using the definition of positive semidefiniteness. Inequality (34) holds for all $\boldsymbol{\Pi}_0, \boldsymbol{Z}_0$ and $\boldsymbol{\Pi}, \boldsymbol{Z}$, and its left-hand side is $\mathcal{C}(\boldsymbol{\Pi}, \boldsymbol{Z}; \boldsymbol{\Pi}_0, \boldsymbol{Z}_0)$. ■

Proof of Proposition 5: Notice that problem (24) has the same feasible set as (23) (with $\boldsymbol{\Pi}_0, \boldsymbol{Z}_0$ replaced by $\boldsymbol{\Pi}_{k-1}, \boldsymbol{Z}_{k-1}$). Corollary 2 establishes that its feasible set is a restriction of the one in (6). It follows that $f(\boldsymbol{p}_k) \geq L$, and $L_k^{(1)} \geq L$ holds because of the added regularizer in (24a). The monotonicity of $\{f(\boldsymbol{p}_k)\}_{k=1}^{\infty}$ follows from a corresponding result in [27, Lemma 4.2(c)]. The sequence is thus monotone decreasing and bounded [the latter follows from the assumption on the boundedness of the feasible set of (6)]. It is a standard result in analysis that a bounded and monotone decreasing sequence has a limit. Therefore, $\hat{f}^{(1)} \geq L$ holds for the limit due to $f(\boldsymbol{p}_k) \geq L$. The convergence result of part c) follows [27, Theorem 4.3]. It is emphasized that the existence of at least one limit point is guaranteed by the boundedness of the feasible set. ■

Proof of Proposition 6: Function $\mathcal{F}_1(\boldsymbol{p})$ is written as

$$\begin{aligned} \mathcal{F}_1(\boldsymbol{p}) &= \mathcal{C}_0 + \mathcal{A}(\boldsymbol{p}) + \mathcal{B}(\boldsymbol{p}) \\ &= \begin{bmatrix} \boldsymbol{O} & \boldsymbol{B}_w \\ \boldsymbol{B}_w^\top & -\alpha\eta\boldsymbol{I} \end{bmatrix} + \begin{bmatrix} \boldsymbol{A}\boldsymbol{S} + \boldsymbol{S}\boldsymbol{A}^\top + \alpha\boldsymbol{S} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix} \\ &\quad + \begin{bmatrix} -\boldsymbol{B}_u\boldsymbol{\Pi}\boldsymbol{Z} - \boldsymbol{Z}^\top\boldsymbol{\Pi}\boldsymbol{B}_u^\top & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix}. \end{aligned}$$

Substituting $\boldsymbol{\Pi} = \boldsymbol{\Pi}_k + \Delta\boldsymbol{\Pi} = \boldsymbol{\Pi}_k + \boldsymbol{\Pi} - \boldsymbol{\Pi}_k$ and $\boldsymbol{Z} = \boldsymbol{Z}_k + \Delta\boldsymbol{Z} = \boldsymbol{Z}_k + \boldsymbol{Z} - \boldsymbol{Z}_k$ into $\mathcal{B}(\boldsymbol{z})$ yields

$$\mathcal{B}(\boldsymbol{p}) = \begin{bmatrix} \begin{array}{c} -\boldsymbol{B}_u(\boldsymbol{\Pi}_k + \Delta\boldsymbol{\Pi})(\boldsymbol{Z}_k + \Delta\boldsymbol{Z}) \\ -(\boldsymbol{Z}_k + \Delta\boldsymbol{Z})^\top(\boldsymbol{\Pi}_k + \Delta\boldsymbol{\Pi})\boldsymbol{B}_u^\top \end{array} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix},$$

where $-\boldsymbol{B}_u(\boldsymbol{\Pi}_k + \Delta\boldsymbol{\Pi})(\boldsymbol{Z}_k + \Delta\boldsymbol{Z}) = -\boldsymbol{B}_u\boldsymbol{\Pi}_k\boldsymbol{Z}_k - \boldsymbol{B}_u\boldsymbol{\Pi}_k\Delta\boldsymbol{Z} - \boldsymbol{B}_u\Delta\boldsymbol{\Pi}\boldsymbol{Z}_k - \boldsymbol{B}_u\Delta\boldsymbol{\Pi}\Delta\boldsymbol{Z}$ and $-(\boldsymbol{Z}_k + \Delta\boldsymbol{Z})^\top(\boldsymbol{\Pi}_k + \Delta\boldsymbol{\Pi})\boldsymbol{B}_u^\top = -\boldsymbol{Z}_k^\top\boldsymbol{\Pi}_k\boldsymbol{B}_u^\top - \boldsymbol{Z}_k^\top\Delta\boldsymbol{\Pi}\boldsymbol{B}_u^\top - \Delta\boldsymbol{Z}^\top\boldsymbol{\Pi}_k\boldsymbol{B}_u^\top - \Delta\boldsymbol{Z}^\top\Delta\boldsymbol{\Pi}\boldsymbol{B}_u^\top$.

Given this, $\mathcal{B}(\boldsymbol{p})$ can be rearranged as

$$\begin{aligned} \mathcal{B}(\boldsymbol{p}) &= \begin{bmatrix} \begin{array}{c} -\boldsymbol{B}_u\boldsymbol{\Pi}_k\boldsymbol{Z}_k - \boldsymbol{Z}_k^\top\boldsymbol{\Pi}_k\boldsymbol{B}_u^\top - \boldsymbol{B}_u\boldsymbol{\Pi}_k\Delta\boldsymbol{Z} \\ -\Delta\boldsymbol{Z}^\top\boldsymbol{\Pi}_k\boldsymbol{B}_u^\top - \boldsymbol{B}_u\Delta\boldsymbol{\Pi}\boldsymbol{Z}_k - \boldsymbol{Z}_k^\top\Delta\boldsymbol{\Pi}\boldsymbol{B}_u^\top \end{array} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix} \\ &\quad + \begin{bmatrix} -\boldsymbol{B}_u\Delta\boldsymbol{\Pi}\Delta\boldsymbol{Z} - \Delta\boldsymbol{Z}^\top\Delta\boldsymbol{\Pi}\boldsymbol{B}_u^\top & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix}. \end{aligned}$$

By combining and grouping these results, we obtain

$$\begin{aligned} \mathcal{F}_1(\boldsymbol{p}) &= \begin{bmatrix} -\boldsymbol{B}_u\boldsymbol{\Pi}_k\boldsymbol{Z}_k - \boldsymbol{Z}_k^\top\boldsymbol{\Pi}_k\boldsymbol{B}_u^\top & \boldsymbol{B}_w \\ \boldsymbol{B}_w^\top & -\alpha\eta\boldsymbol{I} \end{bmatrix} \\ &\quad + \begin{bmatrix} \begin{array}{c} \boldsymbol{A}\boldsymbol{S} + \boldsymbol{S}\boldsymbol{A}^\top + \alpha\boldsymbol{S} - \boldsymbol{B}_u\boldsymbol{\Pi}_k\Delta\boldsymbol{Z} \\ -\Delta\boldsymbol{Z}^\top\boldsymbol{\Pi}_k\boldsymbol{B}_u^\top - \boldsymbol{B}_u\Delta\boldsymbol{\Pi}\boldsymbol{Z}_k - \boldsymbol{Z}_k^\top\Delta\boldsymbol{\Pi}\boldsymbol{B}_u^\top \end{array} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix} \\ &\quad + \begin{bmatrix} -\boldsymbol{B}_u\Delta\boldsymbol{\Pi}\Delta\boldsymbol{Z} - \Delta\boldsymbol{Z}^\top\Delta\boldsymbol{\Pi}\boldsymbol{B}_u^\top & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix}. \end{aligned}$$

An upper bound for the last bilinear term for any $\boldsymbol{Q} \in \mathbb{S}_{++}^{n_u}$ is given as [28, Lemma 1]

$$\begin{aligned} -\boldsymbol{B}_u\Delta\boldsymbol{\Pi}\Delta\boldsymbol{Z} - \Delta\boldsymbol{Z}^\top\Delta\boldsymbol{\Pi}\boldsymbol{B}_u^\top &\preceq \\ \boldsymbol{B}_u\Delta\boldsymbol{\Pi}\boldsymbol{Q}\Delta\boldsymbol{\Pi}\boldsymbol{B}_u^\top + \Delta\boldsymbol{Z}^\top\boldsymbol{Q}^{-1}\Delta\boldsymbol{Z}&. \end{aligned}$$

Combining the previous two results yields (26). ■

Proof of Lemma 4: Let $\mathcal{R}(\boldsymbol{x};\boldsymbol{x}_k)$ be the first-order Taylor approximation of $-\mathcal{Q}(\boldsymbol{x})^{-1}$ computed around $\boldsymbol{x}_k$. That is,

$$\mathcal{R}(\boldsymbol{x};\boldsymbol{x}_k) = -\mathcal{Q}(\boldsymbol{x}_k)^{-1} - [D(\mathcal{Q}(\boldsymbol{x}_k)^{-1})](\boldsymbol{x}-\boldsymbol{x}_k). \tag{35}$$

By setting $\Delta\boldsymbol{x} = \boldsymbol{x}-\boldsymbol{x}_k$, the differential $-[D(\mathcal{Q}(\boldsymbol{x}_k)^{-1})]\Delta\boldsymbol{x}$ is given by [40]

$$\begin{aligned}[D(\mathcal{Q}(\boldsymbol{x}_k)^{-1})]\Delta\boldsymbol{x} &= -\mathcal{Q}(\boldsymbol{x}_k)^{-1}[D\mathcal{Q}(\boldsymbol{x}_k)]\Delta\boldsymbol{x}\mathcal{Q}(\boldsymbol{x}_k)^{-1}\\ &= -\mathcal{Q}(\boldsymbol{x}_k)^{-1}\sum_{i=1}^{n}\frac{\partial\mathcal{Q}(\boldsymbol{x}_k)}{\partial x_i}\Delta x_i\mathcal{Q}(\boldsymbol{x}_k)^{-1}\\ &= -\mathcal{Q}(\boldsymbol{x}_k)^{-1}\mathcal{Q}(\boldsymbol{x})\mathcal{Q}(\boldsymbol{x}_k)^{-1} + \mathcal{Q}(\boldsymbol{x}_k)^{-1}.\end{aligned}$$

Substituting the latter into (35) yields

$$\mathcal{R}(\boldsymbol{x};\boldsymbol{x}_k) = -2\mathcal{Q}(\boldsymbol{x}_k)^{-1} + \mathcal{Q}(\boldsymbol{x}_k)^{-1}\mathcal{Q}(\boldsymbol{x})\mathcal{Q}(\boldsymbol{x}_k)^{-1}.$$

Since $\mathcal{Q}(\boldsymbol{x})$ is positive definite, then it follows that $-\mathcal{Q}(\boldsymbol{x})^{-1}$ is concave [39, Example 3.48]. Because the first-order approximation of a concave function is a global over-estimator, we obtain (28). ■

Proof of Lemma 5: By linearizing $-\boldsymbol{Q}^{-1}$ around a given $\boldsymbol{Q}_k \in \mathbb{S}^{n_u}_{++}$, an upper bound on $\mathcal{K}(\boldsymbol{p};\boldsymbol{p}_k,\boldsymbol{Q})$ can be derived as follows. Since $-\boldsymbol{Q}^{-1}$ is concave in $\boldsymbol{Q}$, then according to Lemma 4, the over approximation of $-\boldsymbol{Q}^{-1}$ around $\boldsymbol{Q}_k$ is $-2\boldsymbol{Q}_k^{-1} + \boldsymbol{Q}_k^{-1}\boldsymbol{Q}\boldsymbol{Q}_k^{-1}$. Substituting this over approximation of $-\boldsymbol{Q}^{-1}$ into $\mathcal{K}(\boldsymbol{p};\boldsymbol{p}_k,\boldsymbol{Q})$ and applying congruence transformation with $\mathrm{diag}(\boldsymbol{I},\boldsymbol{I},\boldsymbol{Q}_k,\boldsymbol{I})$ as the post and pre-multiplier yields (30). The relation in (29) is obtained due to the fact that $-\boldsymbol{Q}^{-1} \preceq -2\boldsymbol{Q}_k^{-1} + \boldsymbol{Q}_k^{-1}\boldsymbol{Q}\boldsymbol{Q}_k^{-1}$. ■

Proof of Proposition 7: The feasible set of problem (31) is a restriction of the one in (6) due to Proposition 6, Lemma 3, Lemma 5. It therefore holds that $f(\boldsymbol{p}_k) \geq L$, and $L_k^{(2)} \geq L$ follows from the addition of the regularizer in the objective. The monotonicity of $\{f(\boldsymbol{p}_k)\}_{k=1}^{\infty}$ follows from a related result in [28, Lemma 6]. The monotonicity and the boundedness imply the existence of the limit, similarly to Proposition 5. The convergence in part c) is analogous to [28, Proposition 5]. The existence of at least one limit point is ensured by the boundedness of the sequence $\{\boldsymbol{p}_k\}_{k=1}^{\infty}$. ■

Appendix C
Successive Convex Approximation Implementation

Algorithm 1 illustrates how the SCAs (24) and (31) can be solved sequentially until a maximum number of iterations (MaxIter) or a stopping criterion defined by a tolerance (tol) are met.

Appendix D
Recovering the Binary Selection

The solutions obtained from (15), (24), and (31) produce a noninteger solution for the actuator selection problem. Since the objective is to determine a binary selection for the actuators, we present in this section a simple *slicing routine* that returns a binary selection given the solutions to the optimization problems in Sections V and VI.

The slicing routine is presented in Algorithm 2. Since the objective of the $\mathcal{L}_\infty$ problem is to find a feedback gain $\boldsymbol{K} = \boldsymbol{Z}\boldsymbol{S}^{-1}$ that renders the closed-loop system stable, the slicing algorithm ensures that the spectrum $\Lambda(\boldsymbol{A}_{\mathrm{cl}})$ of the closed-loop system matrix $\boldsymbol{A}_{\mathrm{cl}} = \boldsymbol{A} - \boldsymbol{B}_u\boldsymbol{\Pi}\boldsymbol{K}$ lies on the left-half plane.

Algorithm 1 Solving the successive convex approximations.

```
input: $\mathrm{MaxIterNum}, \mathrm{tol}, k = 0, \boldsymbol{\Pi}_0 = \boldsymbol{I}_{n_u}$
while $k < \mathrm{MaxIterNum}$ do
  Option 1: Solve (24)
  Option 2: Solve (31)
  if $|\hat{L}_k^{(1)\text{ or }(2)} - \hat{L}_{k-1}^{(1)\text{ or }(2)}| < \mathrm{tol}$ then
    break
  else
    $k \leftarrow k+1$
  end if
end while
output: $\{\boldsymbol{S}^\star, \zeta^\star, \boldsymbol{Z}^\star, \boldsymbol{\Pi}^\star\} \leftarrow \{\boldsymbol{S}^k, \zeta^k, \boldsymbol{Z}^k, \boldsymbol{\Pi}^k\}$
```

The slicing routine takes as an input the real-valued solution to the actuator selection $\boldsymbol{\Pi}^*$ with $\boldsymbol{\pi}_i^* \in [0,1]$. First, the entries of $\boldsymbol{\pi}^*$ are sorted in decreasing order, and the minimum s-actuator selection is obtained such that the logistic constraints $\boldsymbol{H}\boldsymbol{\pi} \leq \boldsymbol{h}$ are satisfied, given that $\boldsymbol{\pi} \in \{0,1\}^N$. This ensures that we start the slicing algorithm from the minimum number of actuators, while still satisfying all of the actuator-related constraints in (4). The algorithm proceeds by activating the s-highest ranked actuators, followed by solving the $\mathcal{L}_\infty$ SDP (4a)–(4c) for $\boldsymbol{Z}$ and $\boldsymbol{S}$. Then, the maximum real part of the eigenvalues of $\boldsymbol{A}_{\mathrm{cl}}$, namely λ_m, is obtained. If $\lambda_m < 0$, the algorithm exits returning the actuator selection $\boldsymbol{\Pi}_s$ and the associated feedback gain.

The algorithm allows the addition of other user-defined requirements, such as a minimum performance index ζ or a maximum λ_m, which can guarantee a minimal distance to instability. It can also be generalized to other control or estimation problems. Notice that Algorithm 2 terminates when $\lambda_m < 0$ and the SDP (4a)–(4c) is solved. These conditions ensure by definition that the system is controllable for the resulting binary actuator combination. In short, the slicing algorithm guarantees the controllability of the system.

Algorithm 2 A Slicing Algorithm to Recover the Integer Selection from (15), (24), and (31)

```
input: $\boldsymbol{\Pi}^*$ from Algorithm 1, set $\lambda_m = \infty$
Sort $\boldsymbol{\pi}^*$ in a decreasing order
$s = \underset{\boldsymbol{\pi}\in\{0,1\}^N, \boldsymbol{H}\boldsymbol{\pi}\leq\boldsymbol{h}}{\mathrm{minimum}} \quad \boldsymbol{1}_N^\top\boldsymbol{\pi}$
while $\lambda_m \geq 0$ do
  Activate the $s$-highest ranked actuators in $\boldsymbol{\pi}$
  Obtain $\boldsymbol{\Pi}_s = \mathrm{blkdiag}(\pi_1\boldsymbol{I}_{n_{u_1}}, \ldots, \pi_N\boldsymbol{I}_{n_{u_N}})$
  Given $\boldsymbol{\Pi} = \boldsymbol{\Pi}_s$, solve the SDP (4a)–(4c) for $\boldsymbol{Z}$ and $\boldsymbol{S}$
  $\lambda_m = \max(\mathrm{real}(\lambda))$ where $\lambda \in \Lambda(\boldsymbol{A} - \boldsymbol{B}_{\boldsymbol{u}}\boldsymbol{\Pi}_s\boldsymbol{Z}\boldsymbol{S}^{-1})$
  $s \leftarrow s+1$
end while
output: $\boldsymbol{\Pi}_s^*, \boldsymbol{K}^* = \boldsymbol{Z}^*(\boldsymbol{S}^*)^{-1}$
```

The actuator selection and associated control law returned by Algorithm 2 yield an upper bound U to the optimal value of the actuator selection problem (3).