

Simultaneous Sensor and Actuator Selection/Placement through Output Feedback Control

Sebastian Nugroho*, Ahmad F. Taha*, Tyler Summers†, Nikolaos Gatsis*

Abstract—In most dynamic networks, it is impractical to measure all of the system states; instead, only a subset of the states are measured through sensors. Consequently, and unlike full state feedback controllers, output feedback control utilizes only the measured states to obtain a stable closed-loop performance. This paper explores the interplay between the selection of minimal number of sensors and actuators (SaA) that yield a stable closed-loop system performance. Through the formulation of the static output feedback control problem, we show that the simultaneous selection of minimal set of SaA is a combinatorial optimization problem with mixed-integer nonlinear matrix inequality constraints. To address the computational complexity, we develop two approaches: The first approach relies on integer/disjunctive programming principles, while the second approach is a simple algorithm that is akin to binary search routines. The optimality of the two approaches is also discussed. Numerical experiments are included showing the performance of the developed approaches.

Index Terms—Sensor and actuator selection and placement, static output feedback control, mixed-integer nonlinear matrix inequality, disjunctive programming, binary search algorithm.

I. INTRODUCTION

The interplay between the selection of minimal number of sensors and actuators (SaA) in dynamic systems is investigated in this paper. In general, the SaA selection or placement problem can be described as finding the optimal binary, on/off configuration of SaA that satisfy certain dynamic system metrics such as closed-loop system stability, output-feedback stability, linear quadratic regulator and robust $\mathcal{H}_2/\mathcal{H}_\infty$ control/estimation metrics. This problem has potential applications in areas such as: large scale power systems [1], [2], power systems integration with microgrids [3], municipal water networks [4], and transportation systems [5], [6].

Various studies investigate the problem of selecting sensors or actuators separately, while invoking the separation principle that decouples the problems of designing controllers and state estimators, while assuming classical state feedback controller. A more interesting problem is that of simultaneously selecting SaA in the context of output feedback control, where the control law is obtained explicitly from the output measurements, rather than the states of the network. Even

when the separation principle is conveniently exploited, the SaA selection problems are inherently coupled.

Three major approaches have been developed in the recent literature of SaA selection. The first approach is based on combinatorial algorithms, heuristics, and detailed algorithms that often exploit network structure and properties [7]–[15]. The second approach entails utilizing semidefinite programming (SDP) formulations of control/estimation methods while including sparsity promoting penalties on the gain matrix—thereby minimizing the total number of activated SaA [16]–[19]. The third approach uses a combination of mixed-integer convex programming, convex relaxations and approximations to obtain the minimal set of SaA [1], [20], [21]. In particular, the problem of simultaneously selecting/placing SaA with dynamic output feedback control is studied in [18], [22]. In this paper, we investigate the problem of simultaneously selecting SaA through static output feedback control framework, where the objective is to stabilize the closed-loop system through the least number of SaA given logistic constraints on the selection of SaA. Two different approaches to solve this problem are proposed.

The paper organization are as follows. First, we discuss the needed assumptions, definitions, and the formulation of the classical static output feedback problem through an SDP—all in Section II. The problem formulation is presented in Section III, where we show that the simultaneous SaA selection requires solving a nonconvex optimization problem with mixed-integer nonlinear matrix inequality (MI-NMI) constraints. Section IV presents the first approach, whereby the problem is transformed to MI-SDP by using disjunctive programming principles [23], [24]. Section V presents a departure from the mixed-integer formulations to an algorithm that is akin to binary search routines. The developed algorithm leverages the SaA problem structure and the sub-optimality or infeasibility of specific SaA combinations. We prove that both approaches yield optimal solutions to the formulated nonconvex problem. Numerical tests are provided in Section VI.

Some of the mathematical proofs are omitted in this version of the paper, but will be included in an extended version of this work.

II. STATIC OUTPUT FEEDBACK CONTROL REVIEW AND PROBLEM FORMULATION

In this section, we present some necessary background including the definition of static output feedback stabilizability

*Department of Electrical and Computer Engineering, The University of Texas at San Antonio, 1 UTSA Circle, San Antonio, TX 78249. †Department of Mechanical Engineering, The University of Texas at Dallas, 800 W Campbell Rd, Richardson, TX 75080. Emails: sebastian.nugroho@my.utsa.edu, {ahmad.taha, nikolaos.gatsis}@utsa.edu, tyler.summers@utdallas.edu. This material is based upon work supported by the National Science Foundation under Grants CMMI 1728629 and CMMI 1728605.

and the SDP that solves for the output feedback gain given a fixed SaA combination.

A. Notation

The set of $n \times n$ symmetric and positive definite matrices are denoted \mathbb{S}^n and \mathbb{S}_{++}^n . For a square matrix \mathbf{X} , the notation $\Lambda(\mathbf{X})$ denotes the set of all eigenvalues of \mathbf{X} . The function $\text{Re}(c)$ extracts the real part of a complex number c , whereas $\text{blkdiag}(\cdot)$ is used to construct a block diagonal matrix. For a matrix $\mathbf{X} \in \mathbb{R}^{p \times q}$, the operator $\text{Vec}(\mathbf{X})$ returns a stacked $pq \times 1$ column vector of entries of \mathbf{X} , while $\text{Diag}(\mathbf{Y})$ returns a $n \times 1$ column vector of diagonal entries of square matrix $\mathbf{Y} \in \mathbb{R}^{n \times n}$. The symbol \otimes denotes the Kronecker product. For any $x \in \mathbb{R}$, $|x|$ and $\lceil x \rceil$ denote the absolute value and ceiling function of x . The cardinality of a set \mathcal{S} is denoted by $|\mathcal{S}|$, whereas $(0)^n$ denotes a n -tuple with zero valued elements.

B. Systems Description

Consider a linear time invariant (LTI) dynamical system consisting of N nodes, with $\mathcal{N} = \{1, \dots, N\}$ defining the set of nodes, modeled in the following state-space equations

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (1a)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t), \quad (1b)$$

where the state, input, and output vectors on each node $i \in \mathcal{N}$ are represented by $\mathbf{x}_i(t) \in \mathbb{R}^{n_{x_i}}$, $\mathbf{u}_i(t) \in \mathbb{R}^{n_{u_i}}$, and $\mathbf{y}_i(t) \in \mathbb{R}^{n_{y_i}}$. The global state, input, and output vectors are written as $\mathbf{x}(t) \triangleq [\mathbf{x}_1^\top(t), \dots, \mathbf{x}_N^\top(t)]^\top$, $\mathbf{u}(t) \triangleq [\mathbf{u}_1^\top(t), \dots, \mathbf{u}_N^\top(t)]^\top$, and $\mathbf{y}(t) \triangleq [\mathbf{y}_1^\top(t), \dots, \mathbf{y}_N^\top(t)]^\top$ where $\mathbf{x}(t) \in \mathbb{R}^{n_x}$, $\mathbf{u}(t) \in \mathbb{R}^{n_u}$, and $\mathbf{y}(t) \in \mathbb{R}^{n_y}$. We assume that the SaA on each node i only correspond to that particular node. Therefore, \mathbf{B} and \mathbf{C} can be respectively constructed as $\mathbf{B} \triangleq \text{blkdiag}(\mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_N)$ and $\mathbf{C} \triangleq \text{blkdiag}(\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_N)$ where $\mathbf{B} \in \mathbb{R}^{n_x \times n_u}$ and $\mathbf{C} \in \mathbb{R}^{n_y \times n_x}$. This assumption enforces the coupling among nodes to be represented in the state evolution matrix $\mathbf{A} \in \mathbb{R}^{n_x \times n_x}$, which is realistic in various dynamic networks as control inputs and observations are often determined locally. In addition, we also assume that \mathbf{B} and \mathbf{C} are full column rank and full row rank, respectively.

To formalize the SaA selection problem, let $\gamma_i \in \{0, 1\}$ and $\pi_i \in \{0, 1\}$ be two binary variables that represent the selection of SaA at node i of the dynamic network. We consider that $\gamma_i = 1$ if the sensor of node i is selected (or activated) and $\gamma_i = 0$ otherwise. Similarly, $\pi_i = 1$ if the actuator of node i is selected and $\pi_i = 0$ otherwise. The augmented dynamics can be written as

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{\Pi}\mathbf{u}(t) \quad (2a)$$

$$\mathbf{y}(t) = \mathbf{\Gamma}\mathbf{C}\mathbf{x}(t), \quad (2b)$$

where $\mathbf{\Pi}$ and $\mathbf{\Gamma}$ are symmetric block matrices defined as

$$\mathbf{\Pi} \triangleq \text{blkdiag}(\pi_1 \mathbf{I}_{n_{u_1}}, \pi_2 \mathbf{I}_{n_{u_2}}, \dots, \pi_N \mathbf{I}_{n_{u_N}}) \quad (3a)$$

$$\mathbf{\Gamma} \triangleq \text{blkdiag}(\gamma_1 \mathbf{I}_{n_{y_1}}, \gamma_2 \mathbf{I}_{n_{y_2}}, \dots, \gamma_N \mathbf{I}_{n_{y_N}}). \quad (3b)$$

C. The Static Output Feedback Stabilizability Problem

We begin this section by providing the definition of static output feedback stabilizability.

Definition 1. *The dynamical system (1) is stabilizable via static output feedback if there exists $\mathbf{F} \in \mathbb{R}^{n_u \times n_y}$, with control law defined as $\mathbf{u}(t) = \mathbf{F}\mathbf{y}(t)$, such that $\text{Re}(\lambda) < 0$ for every $\lambda \in \Lambda(\mathbf{A} + \mathbf{B}\mathbf{F}\mathbf{C})$.*

By using the above definition, the static output feedback stabilizability problem can be defined as the problem of finding \mathbf{F} such that the closed loop system $\mathbf{A} + \mathbf{B}\mathbf{F}\mathbf{C}$ is asymptotically stable. Throughout this paper, we require that dynamical system (1) satisfies the following assumption.

Assumption 1. *The following conditions apply to (1):*

- 1) *The pair (\mathbf{A}, \mathbf{B}) is stabilizable,*
- 2) *The pair (\mathbf{A}, \mathbf{C}) is detectable.*

Note that above assumption is not enough to guarantee that (1) is stabilizable via static output feedback. To proceed, the following proposition provides a sufficient condition for static output feedback stabilizability.

Proposition 1. *The dynamic network (1) is static output feedback stabilizable if there exist an invertible matrix $\mathbf{M} \in \mathbb{R}^{n_u \times n_u}$, matrices $\mathbf{P} \in \mathbb{S}_{++}^{n_x}$, $\mathbf{N} \in \mathbb{R}^{n_u \times n_y}$, and $\mathbf{F} \in \mathbb{R}^{n_u \times n_y}$ such that the following linear matrix inequalities are feasible*

$$\mathbf{A}^\top \mathbf{P} + \mathbf{P}\mathbf{A} + \mathbf{C}^\top \mathbf{N}^\top \mathbf{B}^\top + \mathbf{B}\mathbf{N}\mathbf{C} \prec 0 \quad (4a)$$

$$\mathbf{B}\mathbf{M} = \mathbf{P}\mathbf{B}, \quad (4b)$$

with control law $\mathbf{u}(t) = \mathbf{F}\mathbf{y}(t)$ where $\mathbf{F} = \mathbf{M}^{-1}\mathbf{N}$.

The proof of the above proposition is available in [25]. The condition presented in Proposition 1 allows the static output feedback stabilization problem to be solved as an LMI. The problem formulation of output feedback stabilizability with simultaneous SaA selection is given next.

III. PROBLEM FORMULATION

The simultaneous SaA selection with static output feedback control is the problem of selecting a minimal set of SaA while still maintaining the stability of the system through static output feedback control. Thus, based on Proposition 1, the SaA selection problem for output feedback stabilization can be formulated as follow.

$$\text{minimize} \quad \sum_{k=1}^N \pi_k + \gamma_k \quad (5a)$$

$$\text{subject to} \quad \mathbf{A}^\top \mathbf{P} + \mathbf{P}\mathbf{A} + \mathbf{C}^\top \mathbf{\Gamma} \mathbf{N}^\top \mathbf{\Pi} \mathbf{B}^\top + \mathbf{B}\mathbf{\Pi} \mathbf{\Gamma} \mathbf{N} \mathbf{C} \prec 0 \quad (5b)$$

$$\mathbf{B}\mathbf{\Pi} \mathbf{M} = \mathbf{P}\mathbf{B}\mathbf{\Pi} \quad (5c)$$

$$\mathbf{\Phi} \begin{bmatrix} \boldsymbol{\pi} \\ \boldsymbol{\gamma} \end{bmatrix} \leq \boldsymbol{\phi} \quad (5d)$$

$$\mathbf{P} \succ 0, \boldsymbol{\pi} \in \{0, 1\}^N, \boldsymbol{\gamma} \in \{0, 1\}^N. \quad (5e)$$

In (5), the optimization variables are $\{\boldsymbol{\pi}, \boldsymbol{\gamma}, \mathbf{N}, \mathbf{M}, \mathbf{P}\}$ with $\mathbf{P} \in \mathbb{S}^{n_x}$, $\boldsymbol{\pi} = [\pi_1, \dots, \pi_N]^\top$, and $\boldsymbol{\gamma} = [\gamma_1, \dots, \gamma_N]^\top$.

The additional constraint (5d) can be regarded as a linear logistic constraint, which is useful to model preferred activation or deactivation of SaA on particular nodes and to define the desired minimum and maximum number of active SaA. This constraint is also useful in multi-period selection problems where certain actuators and sensors are deactivated due to logistic constraints.

Upon solving (5), the SaA selection is obtained and represented by $\{\pi^*, \gamma^*\}$ with static output feedback gain F to be computed as $M^{-1}N$, assuming that M is invertible. Note that, (5) is nonconvex due to the presence of MI-NMI in the form of $\Pi N \Gamma$ and mixed-integer bilinear matrix equality in (5c). Thus, problem (5) cannot be solved by any general-purpose mixed integer convex programming solver. To that end, two different approaches that solve or approximate (5) are developed. The first approach is based on disjunctive programming, while the other approach is based on a binary search algorithm. The next section presents the first approach.

IV. DISJUNCTIVE PROGRAMMING FOR SAA SELECTION

The first approach is developed based on disjunctive programming principles [23], [24]. The following theorem presents this result.

Theorem 1. *The optimization problem (5) is equivalent to*

$$\text{minimize } \sum_{k=1}^N \pi_k + \gamma_k \quad (6a)$$

subject to

$$A^\top P + PA + C^\top \Theta^\top B^\top + B\Theta C < 0 \quad (6b)$$

$$\Omega = (B^\top B)^{-1} B^\top P B \quad (6c)$$

$$\Xi = (I - B(B^\top B)^{-1} B^\top) P B \quad (6d)$$

$$\begin{bmatrix} \Psi_1(N, \Theta) \\ \Psi_2(M, \Omega) \\ \Psi_3(\Xi) \end{bmatrix} \leq \begin{bmatrix} L_1 \Delta_1(\Gamma, \Pi) \\ L_2 \Delta_2(\Pi) \\ L_3 \Delta_3(\Pi) \end{bmatrix} \quad (6e)$$

$$(5d), (5e), \quad (6f)$$

where

$$\Psi_1(N, \Theta) = \begin{bmatrix} \text{Vec}(\Theta) \\ -\text{Vec}(\Theta) \\ \text{Vec}(\Theta) \\ -\text{Vec}(\Theta) \\ \text{Vec}(\Theta - N) \\ -\text{Vec}(\Theta - N) \end{bmatrix}$$

$$\Delta_1(\Gamma, \Pi) = \begin{bmatrix} \text{Diag}(\mathbf{I}_{n_y} \otimes \Pi) \\ \text{Diag}(\mathbf{I}_{n_y} \otimes \Pi) \\ \text{Diag}(\Gamma \otimes \mathbf{I}_{n_u}) \\ \text{Diag}(\Gamma \otimes \mathbf{I}_{n_u}) \\ \text{Diag}(2\mathbf{I}_{n_u \times n_y} - \mathbf{I}_{n_y} \otimes \Pi - \Gamma \otimes \mathbf{I}_{n_u}) \\ \text{Diag}(2\mathbf{I}_{n_u \times n_y} - \mathbf{I}_{n_y} \otimes \Pi - \Gamma \otimes \mathbf{I}_{n_u}) \end{bmatrix}$$

$$\Psi_2(M, \Omega) = \begin{bmatrix} \text{Vec}(M) \\ -\text{Vec}(M) \\ \text{Vec}(\Omega) \\ -\text{Vec}(\Omega) \\ \text{Vec}(M - \Omega) \\ -\text{Vec}(M - \Omega) \end{bmatrix}$$

$$\Delta_2(\Pi) = \begin{bmatrix} \text{Diag}(\mathbf{I}_{n_u}^2 - \mathbf{I}_{n_u} \otimes \Pi + \Pi \otimes \mathbf{I}_{n_u}) \\ \text{Diag}(\mathbf{I}_{n_u}^2 - \mathbf{I}_{n_u} \otimes \Pi + \Pi \otimes \mathbf{I}_{n_u}) \\ \text{Diag}(\mathbf{I}_{n_u}^2 + \mathbf{I}_{n_u} \otimes \Pi - \Pi \otimes \mathbf{I}_{n_u}) \\ \text{Diag}(\mathbf{I}_{n_u}^2 + \mathbf{I}_{n_u} \otimes \Pi - \Pi \otimes \mathbf{I}_{n_u}) \\ \text{Diag}(2\mathbf{I}_{n_u}^2 - \mathbf{I}_{n_u} \otimes \Pi - \Pi \otimes \mathbf{I}_{n_u}) \\ \text{Diag}(2\mathbf{I}_{n_u}^2 - \mathbf{I}_{n_u} \otimes \Pi - \Pi \otimes \mathbf{I}_{n_u}) \end{bmatrix}$$

$$\Psi_3(\Xi) = \begin{bmatrix} \text{Vec}(\Xi) \\ -\text{Vec}(\Xi) \end{bmatrix}$$

$$\Delta_3(\Pi) = \begin{bmatrix} \text{Diag}(\mathbf{I}_{n_x \times n_u} - \Pi \otimes \mathbf{I}_{n_x}) \\ \text{Diag}(\mathbf{I}_{n_x \times n_u} - \Pi \otimes \mathbf{I}_{n_x}) \end{bmatrix},$$

with $\Theta \in \mathbb{R}^{n_u \times n_y}$, $\Omega \in \mathbb{R}^{n_u \times n_u}$ and $\Xi \in \mathbb{R}^{n_x \times n_u}$ are additional variables and $L_1, L_2, L_3 \in \mathbb{R}_{++}$ are sufficiently large constants.

The proof is omitted from this version of the work, and will be included in the extended version of the manuscript [26].

Although (6) is equivalent to (5), the quality of the solution that comes out of (6) depends on the choice of L_1 and L_2 . Theorem 1 allows the SaA selection for static output feedback stabilizability to be solved as a MI-SDP. The next section presents a departure from MI-SDP to an algorithm that solves (5).

V. BINARY SEARCH ALGORITHM FOR SAA SELECTION

A. Introduction

In this section, we present an algorithm that is similar in spirit to binary search routines. In what follows, we provide the definitions and examples that are important to understand the algorithm.

Definition 2. *Let \mathcal{S}_π and \mathcal{S}_γ be two N -tuples representing the selection of actuator and sensor. That is, $\mathcal{S}_\pi \triangleq (\pi_1, \dots, \pi_N)$ and $\mathcal{S}_\gamma \triangleq (\gamma_1, \dots, \gamma_N)$. Then, the selection of SaA can be defined as $\mathcal{S} \triangleq (\mathcal{S}_\pi, \mathcal{S}_\gamma)$ such that $\{\Pi, \Gamma\} = \mathcal{G}(\mathcal{S})$, $\Pi = \mathcal{G}_\pi(\mathcal{S})$, and $\Gamma = \mathcal{G}_\gamma(\mathcal{S})$ where $\mathcal{G}(\cdot) : \mathcal{S} \rightarrow \mathbb{R}^{n_u \times n_u} \times \mathbb{R}^{n_y \times n_y}$, $\mathcal{G}_\pi(\cdot) : \mathcal{S} \rightarrow \mathbb{R}^{n_u \times n_u}$, and $\mathcal{G}_\gamma(\cdot) : \mathcal{S} \rightarrow \mathbb{R}^{n_y \times n_y}$ are linear maps. The number of nodes with active SaA can be defined as $\mathcal{H}(\mathcal{S}) \triangleq \sum_{k=1}^N \pi_k + \gamma_k$ where $\mathcal{H}(\cdot) : \mathcal{S} \rightarrow \mathbb{Z}_+$.*

Definition 3. *Let $\mathcal{S} \triangleq \{\mathcal{S}_q\}_{q=1}^\sigma$ be the candidate set such that it contains all possible combinations of SaA where σ denotes the number of total combinations, i.e., $\sigma \triangleq |\mathcal{S}|$. Then, the following conditions hold:*

- 1) For all $\mathcal{S} \in \mathcal{S}$, $\{\Pi, \Gamma\} = \mathcal{G}(\mathcal{S})$ is feasible for (5d), and
- 2) \mathcal{S} is ordered such that $\mathcal{H}(\mathcal{S}_{q-1}) \leq \mathcal{H}(\mathcal{S}_q)$.

Example 1. *Suppose that the dynamical system consists of two nodes with one input and one output on each node. If the logistic constraint dictates that $1 \leq \mathcal{H}(\mathcal{S}) < 4$ for all $\mathcal{S} \in \mathcal{S}$, then the candidate set \mathcal{S} can be constructed as*

$$\mathcal{S} = \left\{ (1, 0, 0, 0), (0, 1, 0, 0), (0, 0, 1, 0), (0, 0, 0, 1), \right. \\ (1, 1, 0, 0), (1, 0, 1, 0), (1, 0, 0, 1), (0, 1, 1, 0), \\ (0, 1, 0, 1), (0, 0, 1, 1), (1, 1, 1, 0), (1, 1, 0, 1), \\ \left. (1, 0, 1, 1), (0, 1, 1, 1) \right\}.$$

B. Binary Search Algorithm to Solve (5)

The objective of this algorithm is to find an optimal solution $\mathcal{S}^* \in \mathcal{S}$ such that $\mathcal{H}(\mathcal{S}^*) \leq \mathcal{H}(\mathcal{S})$ for all $\mathcal{S} \in \mathcal{V}$ where $\mathcal{V} \triangleq \{\mathcal{S} \in \mathcal{S} \mid \{\Pi, \Gamma\} = \mathcal{G}(\mathcal{S}) \text{ is feasible for (4)}\}$. Realize that \mathcal{S}^* might be not unique* and finding one is adequate for our purpose.

The routine to solve SaA selection with static output feedback is now described as follows. Let p be the index of iteration and q be the index of position in the ordered set \mathcal{S} . Hence at iteration p , the candidate set that contains all possible combinations of SaA can be represented as \mathcal{S}_p , with $\sigma = |\mathcal{S}_p|$, and any element of \mathcal{S}_p at position q can be represented by \mathcal{S}_q . Also, let \mathcal{S}^* be the current solution, which is initialized as $\mathcal{S}^* = (0)^{2N}$.

Next, obtain \mathcal{S}_q where $\mathcal{S}_q \in \mathcal{S}_p$ and $q = \lceil \sigma/2 \rceil$. At this step, we need to determine whether system (2) is output feedback stabilizable with a certain combination of SaA $\{\Pi_q, \Gamma_q\} = \mathcal{G}(\mathcal{S}_q)$. To that end, we use the LMIs from Proposition 1. When solving (4) for given $\{\Pi_q, \Gamma_q\}$, let B and C in (4) be substituted with B_q and C_q so that both represent the nonzero components of $B\Pi_q$ and $\Gamma_q C$ that correspond to activated SaA. If B_q and C_q are feasible for (4), then \mathcal{S}^* is updated such that $\mathcal{S}^* = \mathcal{S}_q$. Since \mathcal{S}_q is feasible, then we can discard all combinations that have more or equal number of active SaA. Otherwise, if B_q and C_q are infeasible for (4), then we can discard \mathcal{S}_q and all combinations that (a) have less number of active SaA than \mathcal{S}_q and (b) the active SaA are included in \mathcal{S}_q .

Realize that the above method reduces the size of \mathcal{S}_p in every iteration because one or more elements of \mathcal{S}_p are discarded. Let \mathcal{S}_{p+1} be the new set of all possible combination of SaA after all unwanted combinations of SaA are discarded. Then, we can update the number of possible combinations of SaA as $\sigma = |\mathcal{S}_{p+1}|$. The algorithm now continues and terminates when $\mathcal{S}_p = \emptyset$. The detail of this algorithm is given in Algorithm 1. Example 2 gives an illustration how \mathcal{S}_p is constructed in every iteration.

Example 2. Consider again the dynamic system from Example 1. Let $(1, 0, 0, 1)$ be the starting combination and, for the sake of illustration, assume that (4) is infeasible for this combination. Then, by Algorithm 1, combinations $(1, 0, 0, 0)$ and $(0, 0, 0, 1)$ are discarded. The candidate set now comprises the following elements

$$\mathcal{S}_2 = \left\{ (0, 1, 0, 0), (0, 0, 1, 0), (1, 1, 0, 0), (1, 0, 1, 0), \right. \\ (0, 1, 1, 0), (0, 1, 0, 1), (0, 0, 1, 1), (1, 1, 1, 0), \\ \left. (1, 1, 0, 1), (1, 0, 1, 1), (0, 1, 1, 1) \right\}.$$

Let $(0, 1, 0, 1)$ be the new starting point and assume that this combination is feasible for (4). Then, all combinations that

*The solution might not be unique since there could be more than one combinations of SaA that yield minimum number of activated SaA, while still generating feasible solution to the LMIs for static output feedback stabilizability.

have greater or equal number of active SaA can be discarded. The remaining possible candidates on the candidate set are

$$\mathcal{S}_3 = \left\{ (0, 1, 0, 0), (0, 0, 1, 0) \right\}.$$

This algorithm continues in a fashion similar to the above routine. If none of these combinations in \mathcal{S}_3 is feasible, then Algorithm 1 returns $\mathcal{S}^* = (0, 1, 0, 1)$ as the solution.

Algorithm 1 Binary Search Algorithm

```

1: initialize:  $\mathcal{S}^* = (0)^{2N}$ ,  $p = 1$ 
2: input:  $\mathcal{S}_p$ 
3: while  $\mathcal{S}_p \neq \emptyset$  do
4:   compute:  $\sigma \leftarrow |\mathcal{S}_p|$ ,  $q \leftarrow \lceil \sigma/2 \rceil$ ,  $\mathcal{S}_q \in \mathcal{S}_p$ 
5:   if (4) is feasible then
6:      $\mathcal{S}^* \leftarrow \mathcal{S}_q$ ,  $\mathcal{S}_p \leftarrow \mathcal{S}_p \setminus \{\mathcal{S} \in \mathcal{S}_p \mid \mathcal{H}(\mathcal{S}) \geq \mathcal{H}(\mathcal{S}_q)\}$ 
7:   else
8:      $\mathcal{S}_p \leftarrow \mathcal{S}_p \setminus \{\mathcal{S} \in \mathcal{S}_p \mid \mathcal{S}_q \vee \mathcal{S} = \mathcal{S}_q\}$ 
9:   end if
10:   $p \leftarrow p + 1$ 
11: end while
12: output:  $\mathcal{S}^*$ 

```

Theorem 2. Algorithm 1 returns an optimal solution of (5).

The proof is omitted from this version of the work, and will be included in the extended version of the manuscript [26]. The reason why Algorithm 1 returns an optimal solution of (5) is due to the fact that two SaA configurations can return the same objective value of (5). However, one SaA configuration can yield a *more* stable closed loop system in terms of the distance from the $j\omega$ -axis. This is shown in the numerical tests (Section VI).

C. Modified Binary Search Algorithm

In Algorithm 1, the LMI (4) is solved in every iteration to determine whether a particular combination of SaA yields a feasible or infeasible solution to the static output feedback problem. In this section, we provide a modification to Algorithm 1 so that it no longer requires solving the LMI feasibility problem at each iteration—potentially resulting in a reduction in the computational time.

This simple modification is carried out by replacing Step 5 in Algorithm 1 with stabilizability and detectability tests of linear dynamic systems. To this end, the following propositions are useful.

Proposition 2. Let \mathcal{S} be an arbitrary combination of SaA. If system (2) is stabilizable for \mathcal{S} , then activating one or more actuators from \mathcal{S} will keep system (2) stabilizable. Similarly, if system (2) is unstabilizable for \mathcal{S} , then deactivating one or more actuators from \mathcal{S} will keep system (2) unstabilizable.

Proof. We first prove the first part of the proposition. Let $\mathcal{S} \in \mathcal{S}$ with $\Pi = \mathcal{G}_\pi(\mathcal{S})$. Let $B_1 \in \mathbb{R}^{n_x \times m_1}$ be a matrix that represents the nonzero components of $B\Pi$ that correspond to activated actuators. Since the pair (A, B_1) is stabilizable, then we have $v^\top B_1 = u$ where $u^\top \in \mathbb{R}^{m_1}$ and $u \neq 0$ [27, Theorem 1] for all $v \in \{v \in \mathbb{R}^{n_x} \mid v^\top (A - \lambda I) = 0, \forall \lambda \geq$

$0, \lambda \in \Lambda(\mathbf{A})\}$. Now, define $\mathbf{B} \in \mathbb{R}^{n_x \times m}$ and $\mathbf{B}_2 \in \mathbb{R}^{n_x \times m_2}$, with $m = m_1 + m_2$ and $m \leq n_u$, such that \mathbf{B}_2 represents the addition of activated actuators and $\mathbf{B} = [\mathbf{B}_1 \ \mathbf{B}_2]$. Then, $\mathbf{v}^\top \mathbf{B} = \mathbf{v}^\top [\mathbf{B}_1 \ \mathbf{B}_2] = [\mathbf{v}^\top \mathbf{B}_1 \ \mathbf{v}^\top \mathbf{B}_2] = [\mathbf{u} \ \mathbf{v}^\top \mathbf{B}_2]$.

Since $\mathbf{u} \neq \mathbf{0}$, the pair (\mathbf{A}, \mathbf{B}) is also stabilizable, proving the first part of the proposition. Since the second part of the proposition is the contraposition of the first part, then the proof is complete. ■

Proposition 3. *Let \mathcal{S} be an arbitrary combination of SaA. If system (2) is detectable for \mathcal{S} , then activating one or more sensors from \mathcal{S} will keep system (2) detectable. Similarly, if system (2) is undetectable for \mathcal{S} , then deactivating one or more sensors from \mathcal{S} will keep system (2) undetectable.*

Proposition 3 is the detectability equivalence of Propositions 2 and thus the proof is omitted for brevity. These two propositions allow discarding some combinations of SaA that are either unstabilizable and/or undetectable, or stabilizable and detectable but have more active SaA. Since stabilizability and detectability tests provide no guarantee of static output stabilizability for system (1), we save all combinations of SaA that pass the tests according to the routine in Algorithm 1. This allows the now-modified algorithm to consider the remaining combinations of SaA that contain more active SaA in the case when the best combination that passes the tests cannot give a stabilizing feedback gain. After all combinations of SaA that pass the tests have been stored, we solve (4) starting from the combination with least number of SaA. If a feasible solution exists given this least-cost combination, the modified algorithm terminates. Otherwise, a stored combination having more active SaA is tested until (4) is successfully solved.

The modified algorithm offers flexibility in assessing stabilizability/detectability of dynamic networks, although it no longer yields an optimal solution of (5) as the stabilizability/detectability tests are not enough to guarantee the existence of stabilizing, static output feedback control gain. Specifically, either the PBH or the eigenvector tests [27] can be used. If the pairs $(\mathbf{A}, \mathbf{B}\mathbf{II})$ and/or $(\mathbf{A}, \mathbf{\Gamma}\mathbf{C})$ have large condition number, then the eigenvector test is preferable to be used since MATLAB's rank function tends to return unreasonable results for pairs with large condition number [11].

VI. NUMERICAL EXPERIMENTS

We test the developed methods on a mass spring system [28], [29] that consists of $N = 10$ subsystems with $\mathbf{C} = \mathbf{I}$. All the simulations are performed using MATLAB R2017b running on a 64-bit Windows 10 with 2.5GHz Intel Core i7-6500U CPU and 8 GB of RAM, where each optimization problem is solved using YALMIP [30] with MOSEK version 8.1 [31]. Here, we impose a logistic constraint so that there are at least 2 activated sensors and 2 activated actuators. In this simulation, we consider three different scenarios that follow from the developed approaches in the previous sections:

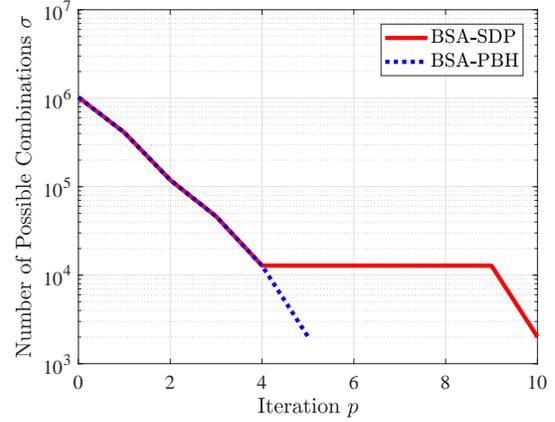


Fig. 1. The reduction of the number of possible combinations. The algorithm terminates when $\sigma = 0$.

- The first scenario (MI-SDP) is carried out by solving problem (6) via YALMIP's MI-SDP branch and bound [30]. We choose $L_1 = 10^5$, $L_2 = 10^5$, $L_3 = 10^5$, $\epsilon_1 = 10^{-9}$, and $\epsilon_2 = 10^{-6}$ such that the left-hand side of (6b) is upper bounded by $-\epsilon_1 \mathbf{I}$ and $\mathbf{P} \succeq \epsilon_2 \mathbf{I}$. Smaller values for L_1 and L_2 resulted in large computational time.
- The second scenario (BSA-SDP) directly follows Algorithm 1 and solves (4) in each iteration to check the feasibility of the given combination of SaA, while also computing the static output feedback gain matrix simultaneously from the solution of LMIs (4).
- The third scenario (BSA-PBH) uses the modified version of Algorithm 1, as explained in section V-C, along with the PBH tests. When the algorithm terminates, the obtained SaA solutions are tested to solve (4). The combination that is feasible for (4) and has the least number of active SaA is then reported as the solution.

The results of this numerical tests are presented in Table I. All scenarios successfully return optimal solutions: 2 sensors and 2 actuators. Among these scenarios, the MI-SDP takes the longest time to compute an optimal solution. The BSA-PBH outperforms the other two scenarios in terms of computational time, while also taking fewer iterations compared to the BSA-SDP. This occurs because in BSA-SDP, problem (4) is solved in each iteration, whereas BSA-PBH only checks the stabilizability and detectability of a given combination of SaA, a process that does not require much computations compared to solving SDPs. The reduction of the number of possible combinations of SaA between BSA-SDP and BSA-PBH is depicted in Figure 1. Note that the algorithm terminates when the candidate set is empty.

VII. SUMMARY AND FUTURE WORK

Two general approaches to minimize the number of selected SaA for static output feedback stabilization are proposed. The first approach is based on solving a MI-SDP, while the second one uses a simple algorithm based on the binary search algorithm. The numerical tests on a mass spring

TABLE I
NUMERICAL TEST RESULTS FOR THE THREE SCENARIOS/METHODS.

Scenario	$\text{Max}(\text{Re}(\Lambda(\mathbf{A} + \mathbf{B}\mathbf{\Pi}^*\mathbf{F}\mathbf{T}^*\mathbf{C})))$	$\sum_{k=1}^N \pi_k + \gamma_k$	$\Delta t(s)$	Iterations	γ^* and π^*
MI-SDP	-3.44×10^{-3}	4	14.13	—	$\gamma^* = \{0, 0, 1, 0, 1, 0, 0, 0, 0, 0\}$ $\pi^* = \{1, 0, 0, 0, 1, 0, 0, 0, 0, 0\}$
BSA-SDP	-2.92×10^{-3}	4	6.77	11	$\gamma^* = \{0, 0, 1, 0, 0, 0, 0, 0, 1, 0\}$ $\pi^* = \{0, 0, 0, 1, 0, 0, 0, 0, 1, 0\}$
BSA-PBH	-1.41×10^{-2}	4	2.68	6	$\gamma^* = \{0, 0, 1, 0, 0, 0, 0, 0, 1, 0\}$ $\pi^* = \{0, 0, 1, 0, 0, 0, 0, 0, 1, 0\}$

system show that both approaches are able to give optimal solutions for the SaA selection problem.

Our future work will focus on investigating the scaling of the proposed methods into larger dynamic networks. Solving the MI-SDP of problem (6) might consume large computational resources for larger systems. Also, a limitation of Algorithm 1 is that it requires traversing the database of all possible SaA combinations. To that end, we plan to develop heuristics so that SaA selection problem can be applied for larger dynamic networks.

REFERENCES

- [1] 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.
- [2] S. Pequito, N. Popli, S. Kar, M. D. Ilic, and A. P. Aguiar, "A framework for actuator placement in large scale power systems: Minimal strong structural controllability," in *In the Proceedings of the IEEE 5th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. IEEE, 2013, pp. 416–419.
- [3] S. Bansal and Y. P. Verma, "Dynamic scheduling of microgrid connected system with storage devices," in *2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, July 2016, pp. 1–6.
- [4] J. W. Berry, L. Fleischer, W. E. Hart, C. A. Phillips, and J.-P. Watson, "Sensor placement in municipal water networks," *Journal of Water Resources Planning and Management*, vol. 131, no. 3, pp. 237–243, 2005.
- [5] M. Tubaishat, P. Zhuang, Q. Qi, and Y. Shang, "Wireless sensor networks in intelligent transportation systems," *Wireless Communications and Mobile Computing*, vol. 9, no. 3, pp. 287–302, 2009. [Online]. Available: <http://dx.doi.org/10.1002/wcm.616>
- [6] S. Contreras, P. Kachroo, and S. Agarwal, "Observability and sensor placement problem on highway segments: A traffic dynamics-based approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 3, pp. 848–858, March 2016.
- [7] 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.
- [8] 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.
- [9] T. Nepusz and T. Vicsek, "Controlling edge dynamics in complex networks," *Nature Physics*, vol. 8, no. 7, pp. 568–573, 2012.
- [10] J. Ruths and D. Ruths, "Control profiles of complex networks," *Science*, vol. 343, no. 6177, pp. 1373–1376, 2014.
- [11] A. Olshevsky, "Minimal controllability problems," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 3, pp. 249–258, 2014.
- [12] 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.
- [13] T. H. Summers, F. L. Cortesi, and J. Lygeros, "On submodularity and controllability in complex dynamical networks," *IEEE Transactions on Control of Network Systems*, vol. 3, no. 1, pp. 91–101, March 2016.
- [14] T. Summers, "Actuator placement in networks using optimal control performance metrics," in *IEEE Conference on Decision and Control*. IEEE, 2016, pp. 2703–2708.
- [15] A. Haber, F. Molnar, and A. E. Motter, "State observation and sensor selection for nonlinear networks," *IEEE Transactions on Control of Network Systems*, 2017, in press.
- [16] B. Polyak, M. Khlebnikov, and P. Shcherbakov, "An lmi approach to structured sparse feedback design in linear control systems," in *European Control Conference*, July 2013, pp. 833–838.
- [17] N. K. Dhingra, M. R. Jovanović, and Z.-Q. Luo, "An admm algorithm for optimal sensor and actuator selection," in *IEEE Conference on Decision and Control*. IEEE, 2014, pp. 4039–4044.
- [18] A. Argha, S. W. Su, and A. Savkin, "Optimal actuator/sensor selection through dynamic output feedback," in *Decision and Control (CDC), 2016 IEEE 55th Conference on*. IEEE, 2016, pp. 3624–3629.
- [19] 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, Nov 2014.
- [20] P. V. Chanekar, N. Chopra, and S. Azarm, "Optimal actuator placement for linear systems with limited number of actuators," in *2017 American Control Conference (ACC)*, May 2017, pp. 334–339.
- [21] A. F. Taha, N. Gatsis, T. Summers, and S. Nugroho, "Time-varying sensor and actuator selection for uncertain cyber-physical systems," Tech. Rep., August 2017. [Online]. Available: <https://arxiv.org/pdf/1708.07912.pdf>
- [22] M. C. De Oliveira and J. Geromei, "Linear output feedback controller design with joint selection of sensors and actuators," *IEEE Transactions on Automatic Control*, vol. 45, no. 12, pp. 2412–2419, 2000.
- [23] G. L. Nemhauser and L. A. Wolsey, "Integer programming and combinatorial optimization," Wiley, Chichester. *GL Nemhauser, MWP Savelsbergh, GS Sigismondi (1992). Constraint Classification for Mixed Integer Programming Formulations. COAL Bulletin*, vol. 20, pp. 8–12, 1988.
- [24] I. E. Grossmann, "Review of nonlinear mixed-integer and disjunctive programming techniques," *Optimization and engineering*, vol. 3, no. 3, pp. 227–252, 2002.
- [25] C. A. Crusius and A. Trofino, "Sufficient lmi conditions for output feedback control problems," *IEEE Transactions on Automatic Control*, vol. 44, no. 5, pp. 1053–1057, 1999.
- [26] S. A. Nugroho, A. F. Taha, N. Gatsis, T. H. Summers, and R. Krishnan, "Algorithms for joint sensor and control nodes selection in dynamic networks," *under review*, 2018.
- [27] M. Hautus, "Stabilization controllability and observability of linear autonomous systems," *Indagationes Mathematicae (Proceedings)*, vol. 73, pp. 448 – 455, 1970. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S138572587080049X>
- [28] F. Lin, M. Fardad, and M. R. Jovanović, "Design of optimal sparse feedback gains via the alternating direction method of multipliers," *IEEE Trans. Automat. Control*, vol. 58, no. 9, pp. 2426–2431, 2013. [Online]. Available: www.umn.edu/~mihailo/software/lqrsp/
- [29] M. Jovanović, "Mass-spring system." [Online]. Available: http://people.ece.umn.edu/~mihailo/software/lqrsp/mass_spring.html
- [30] J. Löfberg, "Yalmip: A toolbox for modeling and optimization in matlab," in *Proc. IEEE Int. Symp. Computer Aided Control Systems Design*. IEEE, 2004, pp. 284–289.
- [31] *The MOSEK optimization toolbox for MATLAB manual. Version 8.1.*, 2018. [Online]. Available: <https://docs.mosek.com/8.1/toolbox/index.html>