

Towards Understanding Sensor and Control Nodes Selection in Nonlinear Dynamic Systems: Lyapunov Theory Meets Branch-and-Bound

Sebastian A. Nugroho^a, Ahmad F. Taha^{b,*}

^a*Department of Electrical Engineering and Computer Science, University of Michigan, 1301 Beal Ave., Ann Arbor, MI 48109*

^b*Department of Civil and Environmental Engineering, Vanderbilt University, 2201 West End Ave, Nashville, TN 37235*

Abstract

Sensor and actuator selection problems (SASPs) are some of the core problems in dynamic systems design and control. These problems correspond to determining the optimal selection of sensors (measurements) or actuators (control nodes) such that certain estimation/control objectives can be achieved. While the literature on SASPs are indeed inveterate, the vast majority of the work focuses on linear(ized) representation of the network dynamics, resulting in the placements of sensors or actuators (SAs) that are valid for confined operating regions. As an alternative, herein we propose a new general framework for addressing SASPs in nonlinear dynamic systems (NDSs), assuming that the inputs and outputs are linearly coupled with the nonlinear dynamics. This is investigated through *(i)* classifying and parameterizing the NDSs into various nonlinear function sets, *(ii)* utilizing rich Lyapunov theoretic formulations, and *(iii)* designing a new customized branch-and-bound (BnB) algorithm that exploits problem structure of the SASPs. The newly designed BnB routines are computationally more attractive than the standard one and also directly applicable to solve SASPs for linear systems. In contrast with contemporary approaches from the literature, our approach is suitable for finding the optimal SAs combination for stable/unstable NDSs that ensures stabilization of estimation error and closed-loop dynamics through a simple linear feedback control policy.

Keywords: Sensor selection, actuator selection, nonlinear dynamic systems, Lipschitz continuous, mixed-integer semidefinite programming, branch-and-bound algorithm

1. Introduction and Paper Contributions

Sensor and actuator selection problems (SASPs)—which fall into *(i)* finding optimal geographic placements or time-varying selection of sensors or actuators (SAs) (but not necessarily simultaneously), while *(ii)* optimizing state estimation or control metrics—have become prevalent research topics in numerous science and engineering fields. For instance, in power networks application, sensor selection corresponds to the placement of phasor measurement units for the purpose of power systems monitoring [1] while actuator selection corresponds to the placement of energy storage devices to ensure system controllability [2]. Both problems are crucial to achieve more economical and robust power systems operations while ensuring that certain minimal requirements, such as allowable tolerance on dynamic estimation error and voltage/frequency deviation, are met.

Extensive studies have been carried out in the literature to address SASPs for particularly *linear* dynamic systems—see [3–6] for some notable references. Unlike SASPs in linear(ized) systems which yield feasible or optimal SAs configurations for specific operating points, SASPs for *nonlinear*

dynamic systems (NDSs) offer SA selections that are applicable for much larger operating regions or even globally—this is demonstrated in our prior work through a simple example [7]. Indeed, SASPs for NDSs have not received comparable attention in the literature as the SASPs for linear systems, until recently. In [1], the concept of *empirical* Gramian to quantify the observability of NDSs is considered jointly with the sensor selection problem (SSP) to determine the combination of sensors that maximizes the logarithmic determinant of the empirical Gramian matrix. The joint problem of sensor selection and state observation is investigated in [8]. In particular, a new method for reconstructing the initial states of NDSs while simultaneously selecting sensors for a given observation window is presented. A different approach is pursued in [9] where the authors introduce a randomized algorithm for dealing with SSP and develop theoretical bounds for eigenvalue and condition number of observability Gramian. Recently, a method for selecting the control nodes and designing control actions for NDSs, which is based on an open-loop predictive control framework, is proposed in [10]. This approach is later adopted in [11] for jointly selecting sensors and performing state estimation. In spite of these efforts, their potential applicability to address SASPs in unstable NDSs along with their ability to incorporate SAs selection with some estimation/control-theoretic metrics, unfortunately, remains unclear.

*Corresponding author. This material is based upon work supported by the National Science Foundation under Grants 1728629, 1917164, and 2044430.

Email addresses: snugroho@umich.edu (Sebastian A. Nugroho), ahmad.taha@vanderbilt.edu (Ahmad F. Taha)

The objective of this paper is to find the minimal SAs combination for stable/unstable NDSs that ensures stabilization for estimation error (via minimal sensor selection) and closed-loop system through feedback control (via minimal actuator selection). The proposed algorithms developed herein are based on the following observations. First, numerous observer and controller designs for NDSs have been developed in the past two decades. By using Lyapunov stability theory and given that the nonlinearities in NDSs satisfy some function sets such as *bounded Jacobian* [12], *Lipschitz continuous* [13], *one-sided Lipschitz* [14], *quadratically inner-bounded* [14], and *quadratically bounded* [15], the procedure to compute stabilizing observer/controller gain matrix can be posed as convex semidefinite programming (SDPs). Secondly, physical states in many practical NDS models are almost always bounded. Hence, it can be shown or safely assumed that the corresponding nonlinearities satisfy the properties of the aforementioned function sets in confined state-space regions—regions that are much larger than operating regions of linearization points.

Without loss of generality, this paper focuses on SASPs for Lipschitz NDSs in which the vector-valued nonlinearity in the system dynamics satisfies the Lipschitz continuity assumption. The proposed method can be easily extended for other types of non-Lipschitz nonlinearities. It is assumed throughout the paper that the inputs and outputs of the NDSs form linear relations with the nonlinear dynamics. It is also worth noting that we do *not* address the simultaneous selection of SAs. That is, we only consider the minimal selection of sensors with observer design or actuators with controller design. A preliminary version of this work appeared in [16], where the SSP for Lipschitz NDSs is studied and solved via a general purpose solver [17]. The key contributions of this paper are summarized as follows:

- We present a novel generalized framework to address SASPs for NDSs by: (i) parameterizing NDSs' nonlinearities based on the corresponding function sets, (ii) formulating the SASPs using a plethora of observer and/or controller designs, and (iii) reformulating the resulting SASPs into convex mixed-integer SDPs (MISDPs).
- Particularly for Lipschitz NDSs with Lipschitz constant γ_l , we formalize NDSs observability and controllability (coined as γ_l -*observability* and γ_l -*controllability*, respectively) as SDP feasibility problems and show that it is crucial to obtain the smallest Lipschitz constant in order to obtain less number of SAs in SASPs. In particular, we theoretically investigate the relationship between the Lipschitz constant of the system and the number of activated SAs.
- A new customized branch-and-bound (BnB) algorithm for MISDPs to solve SASPs is proposed. This BnB algorithm, referred to as *structure-exploiting* BnB (SE-BnB), utilizes heuristics and exploits problem structure to efficiently find optimal and suboptimal solutions for SASPs, owing to the reduced number of constraints and/or variables of the corresponding SDP problems when finding upper and lower bounds on each node of the BnB tree.

- We showcase the computational advantages of the proposed BnB routines for solving SSP in comparison with our implementation of the standard BnB algorithm and the ℓ_1 -norm relaxation technique [18].

The distinction of our approach to address SASPs for NDSs in comparison with the other contemporary methods includes (a) its ability to find optimal placement of SAs with stabilizing estimation/control gain matrix, and (b) its flexibility to incorporate estimation/control metrics for continuous-time/discrete-time NDSs having different classes of nonlinearity. The paper's organization and notations are as follows.

Organization. Section 2 formalizes the NDSs model. Section 3 investigates the relation between Lipschitz constant and γ_l -observability while Section 4 constructs the SSP for Lipschitz NDSs. Next, Section 5 focuses on transforming SSP into a convex MISDP form and provides a detailed description for the SE-BnB algorithm. In Section 6, some discussions related to actuator selection problem, solving SASPs beyond Lipschitz NDSs, and robust SSP are provided. Finally, Section 7 presents some numerical results and Section 8 concludes the paper.

Notation. The notation $\mathbf{1}$ represents a matrix of appropriate dimension with elements of 1. The notations \mathbb{R}^n and $\mathbb{R}^{p \times q}$ denote the sets of row vectors with n elements and matrices with size p -by- q with elements in \mathbb{R} . Given \mathbf{A} as a matrix, its i -th and j -th element is denoted by A_{ij} . The operators $\text{Blkdiag}(\cdot)$ constructs a block diagonal matrix and $\text{vec}(\cdot)$ constructs a vector by stacking each column of a matrix. The symbol $*$ is used to represent symmetric entries in symmetric matrices. The set $\mathbb{I}(n)$ is defined as $\mathbb{I}(n) := \{i \in \mathbb{N} \mid 1 \leq i \leq n\}$, which is usually used to represent the set of indices.

2. System Description and Preliminaries

Consider a continuous-time networked NDS comprised of N subsystems (also referred to as *nodes*) modeled as

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{G}\mathbf{f}(\mathbf{x}) + \mathbf{B}\mathbf{u}(t), \quad \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t). \quad (1)$$

Numerous NDSs such as electric power grids, highway traffic systems, combustion networks, and epidemic outbreaks (see Table 1 in [19]) can be represented in the form of (1). In this model, the global state $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^{n_x}$ consists of N numbers of nodal state $\mathbf{x}_i \in \mathcal{X}_i \subset \mathbb{R}^{n_{x_i}}$. Likewise, each subsystem i is also comprised of n_{u_i} inputs and n_{y_i} measurements such that $\mathbf{u}_i \in \mathcal{U}_i \subset \mathbb{R}^{n_{u_i}}$ and $\mathbf{y}_i \in \mathbb{R}^{n_{y_i}}$. The notations \mathbf{u}_i and \mathbf{y}_i denote the nodal input and measurements vectors whereas \mathcal{X} and \mathcal{U} represent the operating region and admissible inputs for NDS (1). The global input and output vectors are $\mathbf{u} \in \mathcal{U} \subset \mathbb{R}^{n_u}$ and $\mathbf{y} \in \mathbb{R}^{n_y}$. Matrices \mathbf{A} , \mathbf{B} , \mathbf{C} , and \mathbf{G} are all assumed to have appropriate dimensions and in particular, \mathbf{B} and \mathbf{C} are representative of input-to-state and state-to-output mappings and supposed to possess the following structure: $\mathbf{B} = \text{Blkdiag}(\mathbf{B}_1, \dots, \mathbf{B}_N)$, $\mathbf{C} = \text{Blkdiag}(\mathbf{C}_1, \dots, \mathbf{C}_N)$. Matrix \mathbf{A} expresses the *linear* dynamics of the system as well as interaction between subsystems while matrix \mathbf{G} depicts the distribution of nonlinear

mapping $\mathbf{f} : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_y}$ that may capture any linear and nonlinear phenomena in the system.

The ultimate objective of this work is to search for both optimal and suboptimal SAs configurations or locations for NDSs (or equivalently, the number of nonzero columns of \mathbf{B} and nonzero rows of \mathbf{C}) while satisfying some user-defined constraints, which generally may include closed-loop system stability, convergence of estimation error dynamics, and actuator/sensor constraints. It is worth mentioning that this paper emphasizes solving SASPs for continuous-time NDSs with Lipschitz nonlinearities since (i) Lipschitz is one of widely used function sets both in observer and controller designs—for example, see [13, 20]—and (ii) the proposed methodology to solve SASPs can be directly extended for other classes of nonlinearities mentioned previously as well as for discrete-time NDSs with additional estimation/control objectives. To proceed, the subsequent assumption is considered in the paper.

Assumption 1. *The mapping $\mathbf{f} : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_y}$ is locally Lipschitz continuous in \mathcal{X} such that for any $\mathbf{x}, \hat{\mathbf{x}} \in \mathcal{X}$*

$$\|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\hat{\mathbf{x}})\|_2 \leq \gamma_l \|\mathbf{x} - \hat{\mathbf{x}}\|_2, \quad (2)$$

where $\gamma_l \geq 0$ is the corresponding Lipschitz constant.

Readers are referred to [21] for scalable numerical methods to compute Lipschitz constants—including the ones corresponding to the other function sets. In the next section, we formalize the notion of observability for Lipschitz NDSs which is crucial in the context of SSP.

3. Observability for Lipschitz NDS

In contrast to linear systems, quantifying observability for nonlinear systems is, without doubt, much more difficult and less straightforward. To that end, in this section we focus on the concept of observability for NDS (1) in regard to solvability of the corresponding SDPs and Lipschitz constant. To proceed, consider a NDS in the form of (1). A standard Luenberger-like observer for NDS (1) can then be constructed as follows

$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{A}\hat{\mathbf{x}}(t) + \mathbf{G}\mathbf{f}(\hat{\mathbf{x}}) + \mathbf{B}\mathbf{u}(t) + \mathbf{L}(\mathbf{y}(t) - \hat{\mathbf{y}}(t)) \quad (3a)$$

$$\hat{\mathbf{y}}(t) = \mathbf{C}\hat{\mathbf{x}}(t), \quad (3b)$$

where in (3), $\mathbf{L} \in \mathbb{R}^{n_x \times n_y}$ is the observer gain matrix, $\hat{\mathbf{x}}$ is the estimated states, and $\hat{\mathbf{y}}$ is the estimated outputs. The objective here is to compute \mathbf{L} that makes the estimation error dynamics, where estimation error is constructed as $\mathbf{e}(t) := \mathbf{x}(t) - \hat{\mathbf{x}}(t)$, to converge asymptotically towards zero. The next definition quantifies observability for NDS (1), posed as a SDP feasibility problem, for a given γ_l .

Definition 1. *NDS (1) is said to be γ_l -observable if and only if there exist $\mathbf{P} \in \mathbb{S}_{++}^{n_x}$, $\mathbf{Y} \in \mathbb{R}^{n_x \times n_y}$, and $\epsilon \in \mathbb{R}_{++}$ such that the following linear matrix inequality (LMI) is feasible*

$$\begin{bmatrix} \mathbf{A}^\top \mathbf{P} + \mathbf{P}\mathbf{A} - \mathbf{C}^\top \mathbf{Y}^\top - \mathbf{Y}\mathbf{C} + \epsilon \gamma_l^2 \mathbf{I} & * \\ \mathbf{G}^\top \mathbf{P} & -\epsilon \mathbf{I} \end{bmatrix} \prec 0. \quad (4)$$

LMI (4) originates from [13] with special case $\mathbf{G} = \mathbf{I}$ and it can be proven that (4) is indeed sufficient and necessary for the existence of stabilizing matrix \mathbf{L} for the estimation error dynamics, provided that $V(\mathbf{e}) = \mathbf{e}(t)^\top \mathbf{P}\mathbf{e}(t)$ is the Lyapunov function candidate and $\epsilon \in \mathbb{R}_{++}$. Once the LMI is solved, \mathbf{L} can be recovered as $\mathbf{P}^{-1}\mathbf{Y}$. The advantages of utilizing Definition 1 to quantify observability are twofold. First, if there exists any solution for (4) then one can immediately obtain stabilizing gain matrix \mathbf{L} and second, the observability prevails for various operating points in \mathcal{X} . These are in contrast to other types of observability for NDS such as empirical observability Gramian, as it only quantifies observability based on local behaviors of the systems around certain operating points and, upon determining the observability, one is still required to design a state estimator algorithm to estimate the actual states.

At this point, we are now compelled to consider the following question regarding the feasibility of SDP that corresponds to γ_l -observability for NDS (1):

Q1: *What role does constant γ_l have on γ_l -observability?*

To answer **Q1**, we need to further study the LMI given in (4). For the sake of simplicity, let us assume that $\epsilon > 0$ is constant. Then, realize that $\mathbf{Y}\mathbf{C}$ can be expressed as

$$\mathbf{Y}\mathbf{C} = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} Y_{ij} \check{\mathbf{C}}_{ij},$$

where $\check{\mathbf{C}}_{ij} \in \mathbb{R}^{n_y \times n_x}$ is constructed as

$$\text{row}(\check{\mathbf{C}}_{ij})_k = \begin{cases} \{[C_{kl}]\}_{l=1}^{n_x} \in \mathbb{R}^{1 \times n_x}, & \text{if } j = k \\ \mathbf{0}_{1 \times n_x}, & \text{otherwise.} \end{cases}$$

Now let $\check{\mathbf{C}}_{S,ij} := \check{\mathbf{C}}_{ij} + \check{\mathbf{C}}_{ij}^\top$ for each i, j . By using the above expression, the term $\mathbf{Y}\mathbf{C} + \mathbf{C}^\top \mathbf{Y}^\top$ can be expressed as

$$\begin{aligned} \mathbf{Y}\mathbf{C} + \mathbf{C}^\top \mathbf{Y}^\top &= \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} Y_{ij} \check{\mathbf{C}}_{ij} + \left(\sum_{i=1}^{n_x} \sum_{j=1}^{n_y} Y_{ij} \check{\mathbf{C}}_{ij} \right)^\top \\ &= \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} Y_{ij} \check{\mathbf{C}}_{S,ij} = \sum_{k=1}^{n_x n_y} y_{v,k} \bar{\mathbf{C}}_k, \end{aligned} \quad (5)$$

in which $\mathbf{y}_v := [y_{v,1} \ y_{v,2} \ \cdots \ y_{v,n_x n_y}]^\top = \text{vec}(\mathbf{Y})$ and $\bar{\mathbf{C}}_k \in \mathbb{S}^{n_x}$ for each $k \in \mathbb{I}(n_x n_y)$ is associated with the corresponding $\check{\mathbf{C}}_{S,ij}$. Using (5), LMI (4) along with $\mathbf{P} \succ 0$ can be written in the following compact form

$$\mathcal{A}(\mathbf{P}, \mathbf{y}_v) + \mathcal{A}_0 \succ 0, \quad (6a)$$

in which $\mathcal{A} : \mathbb{S}^{n_x} \times \mathbb{R}^{n_x n_y} \rightarrow \mathbb{S}^{2n_x} \times \mathbb{S}^{n_x}$ is constructed as $\mathcal{A}(\mathbf{P}, \mathbf{y}_v) := \mathcal{A}_P(\mathbf{P}) + \mathcal{A}_Y(\mathbf{y}_v)$ and

$$\mathcal{A}_P(\mathbf{P}) := \text{Blkdiag} \left(- \begin{bmatrix} \mathbf{A}^\top \mathbf{P} + \mathbf{P}\mathbf{A} & * \\ \mathbf{G}^\top \mathbf{P} & \mathbf{O} \end{bmatrix}, \mathbf{P} \right) \quad (6b)$$

$$\mathcal{A}_Y(\mathbf{y}_v) := \text{Blkdiag} \left(\begin{bmatrix} \sum_{i=1}^{n_x n_y} y_{v,i} \bar{\mathbf{C}}_i & * \\ \mathbf{O} & \mathbf{O} \end{bmatrix}, \mathbf{O} \right) \quad (6c)$$

$$\mathcal{A}_0 := \text{Blkdiag} \left(\begin{bmatrix} -\epsilon \gamma_l^2 \mathbf{I} & * \\ \mathbf{O} & \epsilon \mathbf{I} \end{bmatrix}, \mathbf{O} \right). \quad (6d)$$

Using the expression provided in (6), we now present the first theoretical result of the paper.

Theorem 1 (Larger Lipschitz Constant Reduces Observability). *There exist $\mathbf{P} \in \mathbb{S}^{n_x}$ and $\mathbf{y}_v \in \mathbb{R}^{n_x n_y}$ such that $\mathcal{A}(\mathbf{P}, \mathbf{y}_v) + \mathcal{A}_0 \succ 0$ if and only if there is no $\mathbf{Z} \in \mathbb{S}^{3n_x}$ in the form of*

$$\mathbf{Z} = \text{Blkdiag} \left(\begin{bmatrix} \mathbf{Z}_1 & * \\ \mathbf{Z}_2^\top & \mathbf{Z}_3 \end{bmatrix}, \mathbf{Z}_4 \right), \quad (7a)$$

such that $\mathbf{Z} \succeq 0$ and $\mathbf{Z} \neq 0$ satisfying

$$\mathbf{Z}_4 = \mathbf{Z}_1 \mathbf{A}^\top + \mathbf{A} \mathbf{Z}_1 + \mathbf{G} \mathbf{Z}_2^\top + \mathbf{Z}_2 \mathbf{G}^\top \quad (7b)$$

$$\sum_{j=1}^{n_x} \sum_{k=1}^{n_y} \bar{C}_{i,jk} \mathbf{Z}_{1,jk} = 0, \quad \forall i \in \mathbb{I}(n_x n_y) \quad (7c)$$

$$\text{tr}(\mathbf{Z}_3) \leq \gamma_l^2 \text{tr}(\mathbf{Z}_1). \quad (7d)$$

The proof of the above theorem is presented in Appendix B of [19]. Problem (7) is referred to throughout the section as the *alternative problem* for (6a). To answer **Q1**, we need to look into inequality (7d). Realize that the term $\text{tr}(\mathbf{Z}_3)$ is lower bounded by zero since $\mathbf{Z} \succeq 0$. Consequently, we get

$$0 \leq \text{tr}(\mathbf{Z}_3) \leq \gamma_l^2 \text{tr}(\mathbf{Z}_1). \quad (8)$$

It is seen from (8) that the feasible set of (7) expands as the value of γ_l increases. In a particular case when $\gamma_l = 0$, the term $\text{tr}(\mathbf{Z}_3)$ is enforced to be equal to zero. Since eigenvalues of \mathbf{Z}_3 cannot be negative, then they must be equal to zero. On the other hand, if γ_l is considerably large, then there is not much restriction on the upper bound of $\text{tr}(\mathbf{Z}_3)$. Hence, it is important to get smaller Lipschitz constant γ_l since this will make the problem described in (7) to have narrower feasible space. This finding is in accordance with empirical observations from the literature: SDP (4) has no solution if γ_l is selected to be sufficiently large—such large Lipschitz constant is labeled as *conservative*. This provides an answer for **Q1**. The next section introduces the SSP for Lipschitz NDSs and presents some properties useful for the development of a more efficient BnB algorithm.

4. Sensor Selection and NDS Observability

In this section, we wish to answer the question:

Q2: *What is the impact of the number of placed sensors on γ_l -observability?*

The above question may seem trivial since it is expected that dynamic systems have *higher degree of observability* as more sensors are placed or utilized. The objective of this section is to demonstrate the above conjecture from γ_l -observability stand point. To begin with, we formally construct the SSP for NDS (1) as follows. Let $\gamma_i \in \{0, 1\}$ be a binary variable that ascertains the activation or deactivation of sensor on each subsystem i . That is, $\gamma_i = 1$ if the sensor measuring subsystem i is activated and $\gamma_i = 0$ otherwise. These variables can be combined into $\boldsymbol{\gamma}$, giving $\boldsymbol{\gamma} := [\gamma_1 \ \gamma_2 \ \cdots \ \gamma_N]^\top$. The following conventions are adopted in this section.

Definition 2. Let \mathcal{S}_γ be an N -tuple representing the selection of sensors, i.e., $\mathcal{S}_\gamma := (\gamma_1, \gamma_2, \dots, \gamma_N)$. The set of active sensor is constructed as $\mathcal{S}_\gamma^{(a)} := \{\gamma \in \mathcal{S}_\gamma \mid \gamma = 1\}$. Let $\bar{\mathcal{S}}_\gamma := (\gamma_1 \mathbf{1}_{n_{y_1}}, \gamma_2 \mathbf{1}_{n_{y_2}}, \dots, \gamma_N \mathbf{1}_{n_{y_N}})$ where $\text{card}(\bar{\mathcal{S}}_\gamma) =$

n_y . The selection of sensors in matrix form can be written as $\boldsymbol{\Gamma} := \text{Blkdiag}(\gamma_1 \mathbf{I}_{n_{y_1}}, \gamma_2 \mathbf{I}_{n_{y_2}}, \dots, \gamma_N \mathbf{I}_{n_{y_N}})$.

By using the above definition, NDS (1) together with sensor selection can be conveniently expressed as

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{G}\mathbf{f}(\mathbf{x}) + \mathbf{B}\mathbf{u}(t) \quad (9a)$$

$$\mathbf{y}(t) = \boldsymbol{\Gamma}\mathbf{C}\mathbf{x}(t). \quad (9b)$$

Additionally, it is also beneficial to consider a set $\mathcal{G}_\gamma \subseteq \{0, 1\}^N$ representing logistic constraints and availability of sensors such that $\boldsymbol{\gamma} \in \mathcal{G}_\gamma$ might be imposed. This in turn allows a simplified high level formulation of a general SSP

$$(\mathbf{P1}) \quad \text{minimize} \quad \mathbf{c}^\top \boldsymbol{\gamma} + \text{EstimationObjective}$$

$$\text{subject to} \quad (9), \boldsymbol{\gamma} \in \mathcal{G}_\gamma, \text{ EstimationConstraints.}$$

The objectives of **P1** are threefold: (i) performing state estimation for NDS (9) while (ii) utilizing the least number of sensors (or satisfying a given constraint over the collections of library of sensors) and (iii) optimizing a specific estimation metric. Vector $\mathbf{c} \in \mathbb{R}_+^N$ in the objective function of **P1** assigns weights for each sensor γ_i . Given **P1**, SSP for Lipschitz NDSs assuming no estimation objective (this is considered in Section 6), can be constructed by incorporating (9) into LMI (4). The resulting problem is given below.

$$(\mathbf{P2}) \quad \text{minimize} \quad \mathbf{c}^\top \boldsymbol{\gamma} \quad (10a)$$

$$\text{subject to} \quad \begin{bmatrix} \mathbf{A}^\top \mathbf{P} + \mathbf{P} \mathbf{A} + \epsilon \gamma_l^2 \mathbf{I} & * \\ -\mathbf{Y} \boldsymbol{\Gamma} \mathbf{C} - \mathbf{C}^\top \boldsymbol{\Gamma} \mathbf{Y}^\top & * \\ \mathbf{G}^\top \mathbf{P} & -\epsilon \mathbf{I} \end{bmatrix} \preceq 0 \quad (10b)$$

$$\mathbf{P} \succ 0, \epsilon > 0, \boldsymbol{\gamma} \in \mathcal{G}_\gamma, \boldsymbol{\gamma} \in \{0, 1\}^N. \quad (10c)$$

In **P2** the objective is to minimize the number of the activated sensors while (a) finding a stabilizing observer gain matrix \mathbf{L} and (b) satisfying sensor's logistic constraints. As $\boldsymbol{\Gamma}$ is a binary matrix variable, **P2** is a nonconvex optimization problem with *mixed-integer bilinear matrix inequalities* (MIBMIs) because of the $\mathbf{Y}\boldsymbol{\Gamma}$ term. Our approach to address this problem is given in the next section. Following (5), the term $\mathbf{Y}\boldsymbol{\Gamma}\mathbf{C} + \mathbf{C}^\top \boldsymbol{\Gamma} \mathbf{Y}^\top$ is equal to

$$\sum_{j=1}^{n_y} \gamma_j \left(\sum_{i=1}^{n_x} Y_{ij} \bar{\mathbf{C}}_{S,ij} \right) = \sum_{k=1}^{n_x n_y} \bar{\gamma}_k y_{v,k} \bar{\mathbf{C}}_k, \quad (11)$$

where each $\bar{\gamma}_k$ corresponds to $\gamma_j \in \bar{\mathcal{S}}_\gamma$. As such we may write $\bar{\gamma}_k \triangleleft \gamma_j$. For the sake of analysis, let us assume that $\boldsymbol{\gamma}$ is fixed. From (11), define $\mathcal{A}_{Y,\boldsymbol{\Gamma}}(\mathbf{y}_v)$ as follows

$$\mathcal{A}_{Y,\boldsymbol{\Gamma}}(\mathbf{y}_v) := \text{Blkdiag} \left(\begin{bmatrix} \sum_{k=1}^{n_x n_y} \bar{\gamma}_k y_{v,k} \bar{\mathbf{C}}_k & * \\ \mathbf{O} & \mathbf{O} \end{bmatrix}, \mathbf{O} \right), \quad (12a)$$

such that, by having $\mathcal{A}_\Gamma(\mathbf{P}, \mathbf{y}_v) := \mathcal{A}_P(\mathbf{P}) + \mathcal{A}_{Y,\boldsymbol{\Gamma}}(\mathbf{y}_v)$, matrix inequality (6a) now becomes

$$\mathcal{A}_\Gamma(\mathbf{P}, \mathbf{y}_v) + \mathcal{A}_0 = \mathcal{A}_P(\mathbf{P}) + \mathcal{A}_{Y,\boldsymbol{\Gamma}}(\mathbf{y}_v) + \mathcal{A}_0 \succ 0. \quad (12b)$$

The next result is established due to Theorem 1.

Theorem 2. Consider a certain sensor configuration \mathcal{S}_γ . There exist $\mathbf{P} \in \mathbb{S}^{n_x}$ and $\mathbf{y}_v \in \mathbb{R}^{n_x n_y}$ satisfying $\mathcal{A}_\Gamma(\mathbf{P}, \mathbf{y}_v) + \mathcal{A}_0 \succ 0$ if and only if there is no $\mathbf{Z} \in \mathbb{S}^{3n_x}$ provided in the form of (7a) such that $\mathbf{Z} \succeq 0$ and $\mathbf{Z} \neq 0$

satisfying (7b), (7d), and $\mathbf{Z}_1 \in \mathcal{Z}(\mathcal{S}_\gamma^{(a)})$ where

$$\mathcal{Z}(\mathcal{S}_\gamma^{(a)}) := \left\{ \mathbf{Z}_1 \in \mathbb{S}^{n_x} \left| \sum_{j=1}^{n_x} \sum_{k=1}^{n_x} \bar{\gamma}_i \bar{C}_{i,jk} Z_{1,jk} = 0, \right. \right. \\ \left. \left. \forall i \in \mathbb{I}(n_x n_y) \wedge \bar{\gamma}_i \triangleleft \gamma_m, \gamma_m \in \mathcal{S}_\gamma^{(a)} \right\}. \quad (13)$$

It follows from Theorem 2 (see Appendix C in [19] for the proof) that including more sensors will reduce the feasible set of the alternative problem and thus, in general, utilizing more sensors reduces the feasible set of the alternative problem. By reducing this set, one can expect that the constraint will be inconsistent thus making the alternative problem infeasible and consequently, giving a feasible solution for (12b). This provides an answer for **Q2**. From Theorem 2, the following additional result is obtained.

Corollary 1. Suppose $\mathcal{S}_{\gamma,1}$ and $\mathcal{S}_{\gamma,2}$ are two distinct sensor's configurations so that $\mathcal{S}_{\gamma,1}^{(a)} \subset \mathcal{S}_{\gamma,2}^{(a)}$. If (12b) has no solution for $\mathcal{S}_{\gamma,2}$, then it also has no solution for $\mathcal{S}_{\gamma,1}$.

The above corollary essentially states that if the system is not γ_l -observable for a particular sensor configuration $\mathcal{S}_{\gamma,2}$, then the system is also not γ_l -observable for any sensor configuration, represented by $\mathcal{S}_{\gamma,1}$, having fewer number of sensors if $\mathcal{S}_{\gamma,1}^{(a)} \subset \mathcal{S}_{\gamma,2}^{(a)}$.

Now, a natural question to ask is whether there is a relation between the value of Lipschitz constant γ_l and the number of activated sensors to achieve γ_l -observability for SSP. Consider a SSP where the objective is to achieve γ_l -observability with minimum number of activated sensors. From previous results, we know that minimizing γ_l while incorporating more sensors reduces the feasible set of the alternative problem. However, as we want to incorporate the least number of sensors, then we can only rely on using the smallest Lipschitz constant γ_l . The rationale here is, if the alternative problem can be made infeasible from (8) by reducing the Lipschitz constant, then we can utilize less number of activated sensors to achieve γ_l -observability. In the next section, we discuss our approach to solve SSP formulated in **P2**.

5. Solving the SSP Through Customized BnB

5.1. From MIBMI to Convex MISDP

Our approach involves transforming the SSP to a convex MISDP. Specifically, our prior work [16] reformulates the SSP **P2** into a convex MISDP using McCormick's relaxation [22]. This is not the only method to convert MIBMI into MISDP: the big-M method, which is popular in disjunctive programming, can also be employed—see [6]. The McCormick's reformulation is performed by defining a new matrix variable $\mathbf{M} := \mathbf{Y}\mathbf{\Gamma}$ where $\mathbf{M} \in \mathbb{R}^{n_x \times n_y}$ given that \mathbf{Y} is bounded such that $\underline{\mathbf{Y}} \leq \mathbf{Y} \leq \bar{\mathbf{Y}}$, while the big-M assumes that $-\mathbf{L1} \leq \mathbf{Y} \leq \mathbf{L1}$ for a sufficiently large constant $L > 0$. For convenience, we consider $\mathbf{Y} \in B(\mathbf{Y})$ where $B(\mathbf{Y})$ is defined as

$$B(\mathbf{Y}) := \begin{cases} \underline{\mathbf{Y}} \leq \mathbf{Y} \leq \bar{\mathbf{Y}}, & \text{for McCormick,} \\ -\mathbf{L1} \leq \mathbf{Y} \leq \mathbf{L1}, & \text{for Big-M.} \end{cases}$$

It is apparent here that the bounds on \mathbf{Y} are determined by a single constant L for big-M and matrices $\underline{\mathbf{Y}}, \bar{\mathbf{Y}}$ for McCormick, making big-M to be a special case of McCormick's relaxation. The resulting problem can now be written as

$$(\mathbf{P3}) \quad \underset{\mathbf{P}, \mathbf{Y}, \mathbf{M}, \epsilon, \gamma}{\text{minimize}} \quad \mathbf{c}^\top \gamma \quad (14a)$$

$$\text{subject to} \quad \begin{bmatrix} \mathbf{A}^\top \mathbf{P} + \mathbf{P}\mathbf{A} + \epsilon \gamma_l^2 \mathbf{I} \\ -\mathbf{M}\mathbf{C} - \mathbf{C}^\top \mathbf{M}^\top & * \\ \mathbf{G}^\top \mathbf{P} & -\epsilon \mathbf{I} \end{bmatrix} \preceq 0 \quad (14b)$$

$$\mathbf{M} = \mathbf{Y}\mathbf{\Gamma}, \mathbf{Y} \in B(\mathbf{Y}), (10c). \quad (14c)$$

Without loss of generality, it is assumed throughout the section that the number of variables in γ are equal to the number of rows in \mathbf{C} . By applying big-M or McCormick's relaxation, the nonconvex MISDP **P3** can be converted to an equivalent convex MISDP given in the next theorem.

Theorem 3. Problem **P3** is equivalent to

$$(\mathbf{P4}) \quad \underset{\mathbf{v}}{\text{minimize}} \quad \mathbf{c}_v^\top \mathbf{v} \quad (15a)$$

$$\text{subject to} \quad \mathcal{L}_v(\mathbf{v}) + \mathcal{C}_v \succeq 0, \quad (15b)$$

$$v_i \in \{0, 1\}, \quad \forall i \in \mathbb{I}_v(\gamma), \quad (15c)$$

where $\mathcal{L}_v(\cdot)$ denotes LMI terms, \mathcal{C}_v denotes the constant terms, and vector $\mathbf{v} \in \mathbb{R}^{n_v}$ populates all decision variables in the following fashion

$$\mathbf{v} := [\text{vec}(\mathbf{P})^\top \quad \epsilon \quad \text{vec}(\mathbf{M})^\top \quad \text{vec}(\mathbf{Y})^\top \quad \gamma^\top]^\top, \quad (15d)$$

with $n_v = \bar{n}_x + 1 + 2n_x n_y + n_y$, $\bar{n}_x := \frac{1}{2}n_x(n_x + 1)$.

The proof of Theorem 3 is provided in Appendix D of [19]. The notation $\mathbb{I}_v(\gamma)$ in (15c) is useful for indicating the index i of \mathbf{v} such that $v_i = \gamma_j$ for each corresponding $j \in \mathbb{I}(\gamma)$; similar notations are also used to label any variable in \mathbf{v} that corresponds to \mathbf{P} , \mathbf{Y} , \mathbf{M} , and ϵ . Specifically, the inequality $\Xi \mathbf{v} \leq \varphi$ which is embedded in LMI (15b) essentially encompasses the McCormick's envelope on the bilinear equality $M_{ij} = Y_{ij}\gamma_j$ expressed as

$$\begin{bmatrix} 1 & -1 & -\underline{Y}_{ij} \\ -1 & 1 & \bar{Y}_{ij} \\ 1 & 0 & -\underline{Y}_{ij} \\ -1 & 0 & \bar{Y}_{ij} \end{bmatrix} \begin{bmatrix} M_{ij} \\ Y_{ij} \\ \gamma_j \end{bmatrix} \leq \begin{bmatrix} -\underline{Y}_{ij} \\ \bar{Y}_{ij} \\ 0 \\ 0 \end{bmatrix}, \quad (16)$$

which is equivalent to the previous bilinear constraint (similarly for big-M). Now that we have a convex MISDP, problem **P4** can be solved using one of the aforementioned techniques. In the next section, we develop a customized BnB algorithm that can be used to compute both optimal and suboptimal solutions for **P4**.

5.2. Structure-Exploiting BnB Algorithm

A natural way to solve such MISDP is through BnB algorithm. In a general purpose BnB algorithm, an SDP is solved in every node, in which the SDP is obtained from relaxing the integer variables to their corresponding convex relaxations. Thus, the complexity of BnB algorithm stems from the difficulties in solving the SDPs and the number of SDPs that need to be solved. In a practical situation, finding a relatively good solution in a much shorter time is more desirable than finding an optimal one. Despite that some

Algorithm 1: SE-BnB Algorithm

```

1 input:  $\mathcal{F}$ ,  $\epsilon_f$ , maxBranch
2 initialize:  $\mathcal{C} = \emptyset$ ,  $\mathcal{W} = \emptyset$ , LB, UB =  $\infty$ ,  $k = 0$ ,  $\mathbf{v}^* = \mathbf{0}$ 
3 process root node:  $\mathcal{T} \leftarrow \text{RelaxSol}(\mathcal{F})$ 
4 update: LB  $\leftarrow \mathbf{c}_v^\top \mathcal{T}(2)$ ,  $\mathcal{C} \leftarrow \mathcal{C} \cup \{\mathcal{T}\}$ 
5 while UB - LB  $> \epsilon_f$  and  $k < \text{maxBranch}$  do
6    $\tilde{\mathcal{T}} = \arg \min_{\mathcal{T}_k \in \mathcal{C}} \mathbf{c}_v^\top \mathcal{T}_k(2)$ ,  $\mathcal{C} \leftarrow \mathcal{C} \setminus \{\tilde{\mathcal{T}}\}$ ,
   LB  $\leftarrow \mathbf{c}_v^\top \tilde{\mathcal{T}}(2)$ 
7   if  $\tilde{\mathcal{T}}(2)_i \in \{0, 1\}$ ,  $\forall i \in \mathbb{I}_v(\gamma)$  then
8      $\tilde{\mathcal{T}}(3) \leftarrow \tilde{\mathcal{T}}(2)$ 
9   else
10     $\{\tilde{\mathcal{T}}(3), \mathcal{W}\} \leftarrow \text{FeasSol}(\tilde{\mathcal{T}}, \mathcal{W})$ 
11    find:  $i \in \mathbb{I}_v(\gamma)$  such that  $\tilde{\mathcal{T}}(2)_i \notin \{0, 1\}$  and
       $\tilde{\mathcal{T}}(2)_i = \min_{j \in \mathbb{I}_v(\gamma)} \left| \tilde{\mathcal{T}}(2)_j - \frac{1}{2} \right|$ 
12     $\mathcal{F}_L := \{\mathbf{v} \in \tilde{\mathcal{T}}(1) \mid v_i \leq 0\}$ 
13    if  $\forall \gamma \in \{0, 1\}^{n_y}$ ,  $\gamma \in \mathcal{I}(\mathcal{F}_L)$ , then  $\gamma \notin \mathcal{W}$  then
14       $\mathcal{T}_L \leftarrow \text{RelaxSol}(\mathcal{F}_L)$ ,  $\mathcal{C} \leftarrow \mathcal{C} \cup \{\mathcal{T}_L\}$ 
15     $\mathcal{F}_R := \{\mathbf{v} \in \tilde{\mathcal{T}}(1) \mid v_i \geq 1\}$ 
16    if  $\forall \gamma \in \{0, 1\}^{n_y}$ ,  $\gamma \in \mathcal{I}(\mathcal{F}_R)$ , then  $\gamma \notin \mathcal{W}$  then
17       $\mathcal{T}_R \leftarrow \text{RelaxSol}(\mathcal{F}_R)$ ,  $\mathcal{C} \leftarrow \mathcal{C} \cup \{\mathcal{T}_R\}$ 
18     $k \leftarrow k + 1$ 
19  if  $\mathbf{c}_v^\top \tilde{\mathcal{T}}(3) < \text{UB}$  then
20    UB  $\leftarrow \mathbf{c}_v^\top \tilde{\mathcal{T}}(3)$ ,  $\mathbf{v}^* \leftarrow \tilde{\mathcal{T}}(3)$ 
21  foreach  $\mathcal{T}_j \in \mathcal{C}$  do
22    if  $\mathbf{c}_v^\top \mathcal{T}_j(2) > \text{UB}$  then
23       $\mathcal{C} \leftarrow \mathcal{C} \setminus \{\mathcal{T}_j\}$ 
24 output: LB, UB,  $\mathbf{v}^*$ 

```

general purpose BnB algorithms for MISDPs are available, these conventional solvers cannot be configured to make them amenable for exploiting the structure of the problem at hand in order to reduce the complexity of solving SSP.

To that end, we present a customized BnB algorithm that can be utilized to compute optimal and suboptimal solutions for **P4** in a more efficient manner, owing it to (a) its ability to exploit problem structure and (b) heuristics to quickly find good suboptimal solutions. To proceed, define

$$\mathcal{F} := \{\mathbf{v} \mid \mathcal{L}_v(\mathbf{v}) + \mathcal{C}_v \geq 0, v_i \in \{0, 1\}, \forall i \in \mathbb{I}_v(\gamma)\}$$

$$\mathcal{R} := \{\mathbf{v} \mid \mathcal{L}_v(\mathbf{v}) + \mathcal{C}_v \geq 0, v_i \in [0, 1], \forall i \in \mathbb{I}_v(\gamma)\},$$

i.e., \mathcal{F} denotes the feasible set for **P4** whereas \mathcal{R} is its convex relaxation. Note that we have $\mathcal{F} \subseteq \mathcal{R}$ because in \mathcal{R} , the integer constraints are relaxed. The proposed BnB algorithm solves two convex problems on each BnB node in order for it to be able to find a good suboptimal solution immediately: the convex relaxation of **P4** to get a *local* lower bound, and **P4** with fixed SAs combination to get a *global* upper bound. Within this section, it is supposed that each node i is representable by a 3-tuple $\mathcal{T}_i := (\mathcal{F}_i, \underline{\mathbf{v}}, \bar{\mathbf{v}})$, where

$$\underline{\mathbf{v}} = \arg \min_{\mathbf{v} \in \mathcal{F}_i} \mathbf{c}_v^\top \mathbf{v}, \quad \bar{\mathbf{v}} \in \{\mathbf{v} \mid \mathbf{v} \in \mathcal{F}_i\},$$

such that $\mathcal{F}_i = \mathcal{T}_i(1)$, $\underline{\mathbf{v}} = \mathcal{T}_i(2)$, and $\bar{\mathbf{v}} = \mathcal{T}_i(3)$. We also use

Algorithm 2: FeasSol

```

1 input:  $\tilde{\mathcal{T}}, \mathcal{W}$ 
2 candidate generation: generate a random sensor
  combination  $\gamma_c \in \{0, 1\}^{n_y}$  such that  $\gamma_c \notin \mathcal{W}$  and
   $\gamma_c \in \mathcal{I}(\tilde{\mathcal{T}}(1))$ 
3 solve:  $\bar{\mathbf{f}} = \min_{\mathbf{v} \in \tilde{\mathcal{T}}(1), v_i = \gamma_{c,i}} \mathbf{c}_v^\top \mathbf{v}$  (problem P6) with
  minimizer  $\bar{\mathbf{v}}$ 
4 if  $\bar{\mathbf{f}} = \infty$  then
5    $\mathcal{W} \leftarrow \mathcal{W} \cup \{\gamma_c\}$ 
6 output:  $\bar{\mathbf{v}}, \mathcal{W}$ 

```

an abstract set $\mathcal{I}(\cdot) \subseteq \mathbb{R}^{n_y}$ where $\mathcal{I}(\mathcal{T}_i) \subset \mathcal{T}_i(1)$ to represent the restrictions (or *integer cut*) on integer variable v_i , where $i \in \mathbb{I}_v(\gamma)$, which is added in the branching process of BnB algorithm. To stop the algorithm when a suboptimal solution is sought, two stopping criteria characterize by $\epsilon_f \in \mathbb{R}_+$ and maxBranch $\in \mathbb{N}_+$ are employed. Let LB and UB be the known best lower and upper bounds at the end of the algorithm. When the optimality gap UB - LB is zero, the solution is optimal. However, if UB - LB $\leq \epsilon_f$ the solution is regarded to be ϵ_f -optimal. Alternatively, the solution is only suboptimal. The notation maxBranch, sometimes abbreviated as mBr, determines the number of maximum allowable branching. These BnB routines, referred to as *structure-exploiting BnB* (SE-BnB), are detailed in Algorithm 1. The algorithmic function RelaxSol(\cdot) solves the relaxed problem $\bar{\mathbf{f}} = \min_{\mathbf{v} \in \mathcal{R}} \mathbf{c}_v^\top \mathbf{v}$ for a given \mathcal{F} where $\bar{\mathbf{v}}$ is the minimizer and wrap the results into \mathcal{T} . In addition to this, Algorithm 1 uses a heuristic procedure for efficiently finding a feasible solution for **P4** on each node, if such solution exists. This heuristic generates a random combination of sensors γ_c . To do so, we define the set \mathcal{W} as

$$\mathcal{W} := \left\{ \gamma \in \{0, 1\}^{n_y} \mid [\mathbf{w}^\top \gamma^\top]^\top \notin \mathcal{F}, \forall \mathbf{w} \right\},$$

i.e., the set of combinations of sensors that are infeasible for **P4**. By exploiting the property proposed in Corollary 1, we are able to effectively generate a candidate of sensor combination γ_c that is not positively infeasible for **P4**. This heuristic, embedded in the function FeasSol(\cdot), is presented in Algorithm 2. This idea is implemented in Steps 13 and 16 of Algorithm 1. Now, some modifications for making the SE-BnB algorithm to be more efficient are discussed.

- In RelaxSol(\cdot), the relaxed **P4** is solved. Realize that an integer cut is appended in every node (apart from the root node) that corresponds to whether a particular relaxed integer variable, say v_i for $i \in \mathbb{I}_v(\gamma)$, is constrained to be $v_i \leq 0$ or $v_i \geq 1$. As $v_i \in [0, 1]$, then the only feasible solution is $v_i = 0$ or $v_i = 1$. Either case, the resulting McCormick's envelope (or big-M) yields the *best* relaxation which makes (16) redundant. At this instance, constraint (16) can be discarded and replaced with $M_{ij} = 0$ or $M_{ij} = Y_{ij}$, depending on the value of v_i .
- Recall that the SE-BnB algorithm attempts to find a feasible solution on each node by fixing γ and solve **P4**—performed in FeasSol(\cdot). Since **P4** is equivalent to **P3** and **P3** becomes a SDP when γ is fixed, then one can

solve **P3** to find a feasible solution. Furthermore, given a fixed γ , the equality constraint $\mathbf{M} = \mathbf{Y}\mathbf{\Gamma}$ becomes redundant and hence, this constraint can be removed as well as the variable \mathbf{M} from **P3**. Moreover, when a certain variable $\gamma_i = 0$, due to redundancy, then the corresponding row of \mathbf{C} and column of \mathbf{Y} can also be removed to obtain smaller number of variables (essentially, only consider the nonzero row of \mathbf{C} and column of \mathbf{Y}).

The aforementioned procedures enable the SE-BnB to exploit the structure of SSP.

6. Some Important Extensions

6.1. Controllability and Actuator Selection

In this section, NDS controllability and actuator selection problem (ASP) are discussed, in which we consider a similar concept to quantify controllability for a given γ_l .

Definition 3. *NDS (1) is said to be γ_l -controllable if and only if there exist $\mathbf{Q} \in \mathbb{S}_{++}^{n_x}$, $\mathbf{X} \in \mathbb{R}^{n_u \times n_x}$, and $\sigma \in \mathbb{R}_{++}$ such that the LMI below is feasible*

$$\begin{bmatrix} \mathbf{Q}\mathbf{A}^\top + \mathbf{A}\mathbf{Q} - \mathbf{X}^\top \mathbf{B}^\top - \mathbf{B}\mathbf{X} + \sigma \mathbf{G}\mathbf{G}^\top & * \\ \mathbf{Q} & -\frac{\sigma}{\gamma_l^2} \mathbf{I} \end{bmatrix} \prec 0. \quad (17)$$

If the NDS is γ_l -controllable, then from the above LMI, the stabilizing controller gain matrix is given as $\mathbf{K} = \mathbf{X}\mathbf{Q}^{-1}$ with control action $\mathbf{u}(t) = -\mathbf{K}\mathbf{x}(t)$ [23]. It is worth noting that, unlike (4), LMI (17) is a sufficient condition. A similar question to **Q1** now emerges: *what role does γ_l have on γ_l -controllability?* By performing a similar analysis as in Section 3 (not shown here due to space constraint), it is revealed that if γ_l is made to be sufficiently small, then there exists at least a solution for LMI (17) (if σ is fixed).

Next, the relation between actuators and NDS controllability is discussed. Let $\pi_i \in \{0, 1\}$ be a binary variable representing the activation or deactivation of actuator/control node on each subsystem i such that $\pi_i = 1$ if control node at i is activated and $\pi_i = 0$ otherwise. For compactness, let us define $\boldsymbol{\pi} := [\pi_1 \ \pi_2 \ \cdots \ \pi_N]^\top$ and $\boldsymbol{\Pi}$ as

$$\boldsymbol{\Pi} := \text{Blkdiag}(\pi_1 \mathbf{I}_{n_{u_1}}, \pi_2 \mathbf{I}_{n_{u_2}}, \dots, \pi_N \mathbf{I}_{n_{u_N}}).$$

As such, ASP for NDS (1) can be expressed as

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{G}\mathbf{f}(\mathbf{x}) + \mathbf{B}\boldsymbol{\Pi}\mathbf{u}(t), \quad \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t), \quad (18)$$

from which a simplified high level formulation for ASP can be formulated as follows

$$\begin{aligned} (\mathbf{P5}) \quad & \text{minimize} \quad \mathbf{c}^\top \boldsymbol{\pi} + \text{ControlObjective} \\ & \text{subject to} \quad (18), \boldsymbol{\pi} \in \mathcal{G}_\pi, \text{ControlConstraints.} \end{aligned}$$

The goals in the above are threefold: (i) performing feedback stabilizing control on NDS (18) while (ii) utilizing smallest number of actuators (or satisfying a given constraint over the collections of library of actuators) and (iii) optimizing a specific control metric. In **P5**, vector $\mathbf{c} \in \mathbb{R}_+^N$ assigns weights for each actuator π_i whereas $\mathcal{G}_\pi \subseteq \{0, 1\}^N$ represents actuator's logistic constraints. The ASP for Lipschitz NDS can be formulated as

$$(\mathbf{P6}) \quad \text{minimize}_{\mathbf{Q}, \mathbf{X}, \epsilon, \pi} \quad \mathbf{c}^\top \boldsymbol{\pi} \quad (19a)$$

$$\text{subject to} \quad \begin{bmatrix} \mathbf{Q}\mathbf{A}^\top + \mathbf{A}\mathbf{Q} + \sigma \mathbf{G}\mathbf{G}^\top & * \\ -\mathbf{X}^\top \boldsymbol{\Pi} \mathbf{B}^\top - \mathbf{B}\boldsymbol{\Pi} \mathbf{X} & * \\ \mathbf{Q} & -\frac{\sigma}{\gamma_l^2} \mathbf{I} \end{bmatrix} \preceq 0 \quad (19b)$$

$$\mathbf{Q} \succ 0, \sigma > 0, \boldsymbol{\pi} \in \mathcal{G}_\pi, \boldsymbol{\pi} \in \{0, 1\}^N. \quad (19c)$$

Notice that **P6** is also a nonconvex optimization problem with a MIBMI constraint. Further analysis shows that, in general, in order to employ less number of actuators, one should obtain the least possible Lipschitz constant.

6.2. NDS Parameterization for Non-Lipschitz Systems

Note that many observer/controller designs for function sets other than Lipschitz continuous such as one-sided Lipschitz, quadratically inner-bounded, bounded Jacobian, and quadratically bounded—see [12, 14, 15] for notable examples—can also be considered and utilized to construct SASPs. For example, the nonlinearities in a NDS may satisfy one-sided Lipschitz (OSL) and quadratically inner-bounded (QIB) conditions—see Definition 4.

Definition 4 (OSL & QIB). *The mapping $\mathbf{f} : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_g}$ in (1) is locally OSL in \mathcal{X} if for any $\mathbf{x}, \hat{\mathbf{x}} \in \mathcal{X}$ then*

$$\langle \mathbf{G}(\mathbf{f}(\mathbf{x}) - \mathbf{f}(\hat{\mathbf{x}})), \mathbf{x} - \hat{\mathbf{x}} \rangle \leq \gamma_s \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2,$$

for $\gamma_s \in \mathbb{R}$ and QIB in \mathcal{X} if for any $\mathbf{x}, \hat{\mathbf{x}} \in \mathcal{X}$ it holds that

$$\langle \mathbf{G}(\mathbf{f}(\mathbf{x}) - \mathbf{f}(\hat{\mathbf{x}})), \mathbf{G}(\mathbf{f}(\mathbf{x}) - \mathbf{f}(\hat{\mathbf{x}})) \rangle \leq$$

$$\gamma_{q1} \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 + \gamma_{q2} \langle \mathbf{G}(\mathbf{f}(\mathbf{x}) - \mathbf{f}(\hat{\mathbf{x}})), \mathbf{x} - \hat{\mathbf{x}} \rangle,$$

for $\gamma_{q1}, \gamma_{q2} \in \mathbb{R}$.

In Definition 4, the notation $\langle \cdot, \cdot \rangle$ denotes the standard inner product. By considering OSL and QIB, the parameterization of NDSs entails to finding the constants γ_s, γ_{q1} , and γ_{q2} . These constants can be computed either analytically or numerically via stochastic point-based method or deterministic interval-based method. If numerical approach is pursued, then the computation of these constants reduces to solving global optimization problems. Our particular work in [21] investigates NDSs parameterization using interval-based global maximization.

6.3. From OSL and QIB to Robust Selection

Now we demonstrate how the proposed methodology can be easily extend to solve SASP for other class of nonlinearities. Now suppose that $\mathbf{f}(\cdot)$ in NDS (1) is both OSL and QIB with constants γ_s, γ_{q1} , and γ_{q2} . Using observer design presented in [24], the corresponding SSP can be formed as

$$\text{minimize}_{\mathbf{P}, \mathbf{Y}, \epsilon_1, \epsilon_2, \gamma} \quad \mathbf{c}^\top \boldsymbol{\gamma} \quad (21a)$$

$$\text{subject to} \quad \begin{bmatrix} \mathbf{A}^\top \mathbf{P} + \mathbf{P}\mathbf{A} + \epsilon_1 \gamma_s \mathbf{I} & * \\ +\epsilon_2 \gamma_{q1} \mathbf{I} - \mathbf{Y}\mathbf{\Gamma}\mathbf{C} - \mathbf{C}^\top \mathbf{\Gamma}\mathbf{Y}^\top & * \\ \mathbf{G}^\top \mathbf{P} + \frac{\gamma_{q2} \epsilon_2 - \epsilon_1}{2} \mathbf{I} & -\epsilon_2 \mathbf{I} \end{bmatrix} \preceq 0 \quad (21b)$$

$$\mathbf{P} \succ 0, \epsilon_1, \epsilon_2 > 0, \boldsymbol{\gamma} \in \mathcal{G}_\gamma, \boldsymbol{\gamma} \in \{0, 1\}^N, \quad (21c)$$

In addition to the above, the proposed sensor selection framework can also facilitate to include robustness in the objective. For instance, by considering the Lipschitz condition, robust SSP with \mathcal{L}_∞ metric can be constructed as [20]

$$\text{minimize}_{\mathbf{P}, \mathbf{Y}, \epsilon, \alpha, \mu_0, 1, 2} \quad \mu_0 \mu_1 + \mu_2 \quad (22a)$$

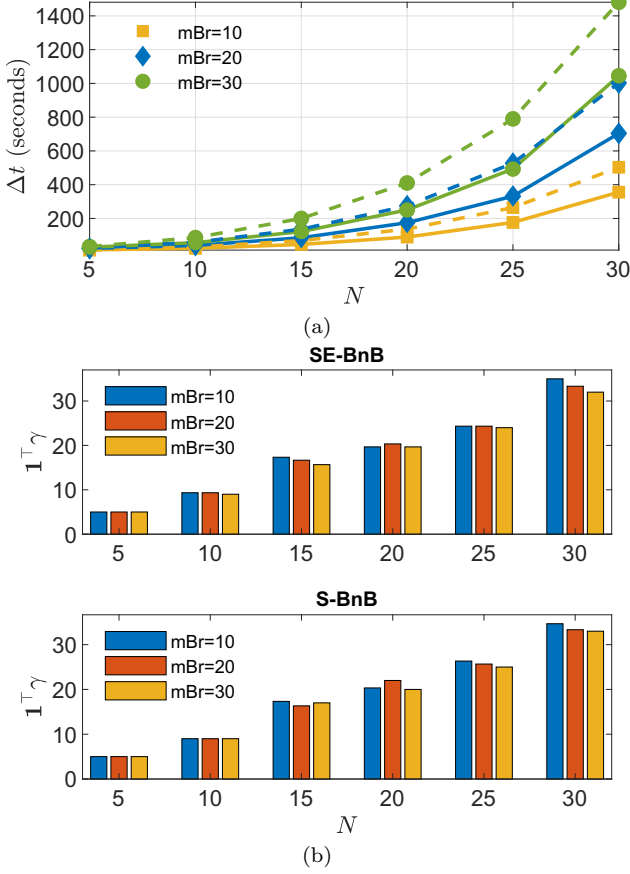


Figure 1: Numerical experiment results for network of nonlinear unstable nodes to find suboptimal solutions via SE-BnB and S-BnB: (a) computational time for different number of maximum allowable branching mBr and (b) the resulting sensor costs. The solid lines correspond to SE-BnB whereas dashed lines correspond to S-BnB. To compensate the heuristics on both SE-BnB and S-BnB, the numerical test is performed three times for each value of N and the results shown correspond to the average values.

subject to

$$\begin{bmatrix} A^T P + P A - C^T \Gamma Y^T \\ -Y \Gamma C + \alpha P + \epsilon \gamma_l^2 I & * & * \\ P & -\epsilon I & * \\ B_w^T P - D_w^T Y^T & O & -\alpha \mu_0 I \end{bmatrix} \preceq 0 \quad (22b)$$

$$\begin{bmatrix} -P & * & * \\ O & -\mu_2 I & * \\ Z & O & -\mu_1 I \end{bmatrix} \preceq 0, \quad (22c)$$

$$P \succ 0, \epsilon, \mu_0, \mu_1, \mu_2 \geq 0, \alpha > 0, \gamma \in \mathcal{G}_\gamma, \gamma \in \{0, 1\}^N, \quad (22d)$$

where matrices B_w and D_w represent how the disturbances are distributed, Z is the performance matrix with $z(t) = Z e(t)$, and $e(t)$ is the estimation error. The objective herein is to select the best sensor combination, given a fixed number of activated sensors, that minimizes the worst case disturbance attenuation $\mu = \sqrt{\mu_0 \mu_1 + \mu_2}$. If the problem described in (22) is solved, then it is ensured that $\lim_{t \rightarrow \infty} \sup \|z(t)\|_2 \leq \mu \|w(t)\|_{\mathcal{L}_\infty}$ where $w(t)$ denotes the disturbance vector [20]. By fixing some constants, (21) and (22) share similar nonconvexity with **P2** and thus the proposed methodology can be used to solve the SSP.

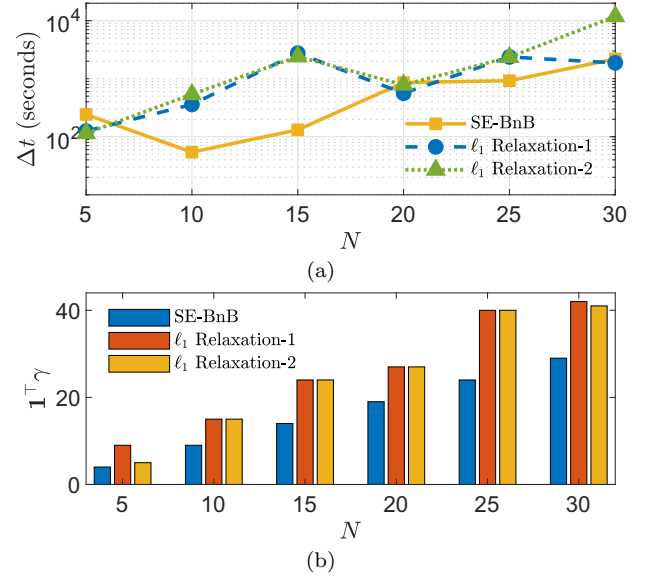


Figure 2: Comparison between SE-BnB and ℓ_1 -norm relaxation to search for suboptimal solutions: (a) computational time and (b) the resulting sensor costs. To compensate the heuristics on SE-BnB, the numerical test is performed three times and the results shown correspond to the average values.

7. Numerical Assessment

In this section, we demonstrate the proposed methodology to solve SSP for a NDS on network of nonlinear unstable nodes. The simulations are performed using MATLAB R2019a, where YALMIP's [17] optimization package with MOSEK [25] solver are used to solve the corresponding SDPs problem. The network of nonlinear unstable nodes considered here is adapted from [26], which is initially a network of linear systems. In order to obtain nonlinear systems, a sinusoidal function is introduced on each node such that each node i has the following dynamics

$$\dot{x}_i = \begin{bmatrix} \zeta_{1i} & 1 \\ 1 & \zeta_{2i} \end{bmatrix} x_i + \begin{bmatrix} 0 \\ \beta_i \end{bmatrix} \sin(x_{2i}) + \sum_{j \neq i} e^{\alpha(i,j)} x_j + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_i, \quad (23)$$

where the constants ζ_{1i} , ζ_{2i} , and β_i are randomly generated within $[-2, 2]$, $[-2, 2]$, and $[-1, 1]$ respectively for each i . The coupling between nodes i and j is determined by the function of Euclidean distance $\alpha(i, j)$. This distance is randomly generated inside a box of size 5×5 . It can be verified that the Lipschitz constant for this system is $\gamma_l = \sqrt{N}$ where N is the number of nodes. For the sake of simplicity, the weighting vector is set to be $c = \mathbf{1}$ where Y is bounded such that $-10^3 \leq Y_{i,j} \leq 10^3$ for each i, j . From (23), each node i can have at most two measurements—hence for N subsystems, there are at most $2N$ measurements.

7.1. SE-BnB and Standard BnB

In the first case of this numerical assessment, we evaluate the performance of the SE-BnB algorithm to find suboptimal solutions for SSP. The tolerance defining optimality gap is set to be 10^{-4} with integer tolerance 10^{-6} while restricting the maximum number of allowable branching. We also invoke a logistic constraint such that at least 20% of the

available measurements are equipped with sensors. We test the new SE-BnB algorithm with various number of nodes and compare the results with different values of maxBranch (or mBr) against our own implementation of the standard BnB algorithm (S-BnB)—that is, only takes MISDP **P4**—while still utilizes the heuristic discussed in Section 5.2. The results of this numerical test are illustrated in Fig. 1. It can be seen that, in overall, the SE-BnB requires less computational time than the S-BnB. This is due to the fact that, in the SE-BnB, the corresponding SDPs used to obtain local lower bound and global upper bound on each BnB node have reduced complexities as opposed to the S-BnB. It is also showcased in Fig. 1b that this reduction in computational time does not contribute to the resulting sensor costs since sensor costs are merely determined by the number of explored nodes as well as the heuristics used to find global upper bounds. Both SE-BnB and S-BnB show similar sensor costs. These costs are reduced, despite of some irregularities, as more branches are explored—these irregularities are attributed to the heuristics used in finding global upper bounds.

7.2. SE-BnB and ℓ_1 -Norm Relaxation

Next, we assess the ability of our SE-BnB algorithm in finding good suboptimal solutions relative to ℓ_1 -norm relaxation, which is developed in [18], to solve a class of combinatorial optimization problems. This technique finds its popularity in SASPs for linear systems—for example, see [27]. This particular approach works as follows. First, note that each γ_i determines the values of $\text{col}(\mathbf{Y})_i$, i.e., $\text{col}(\mathbf{Y})_i$ can be any value if and only if $\gamma_i = 1$ and zero otherwise. As such, the integer variable γ can be removed from optimization variables and consequently, the SSP boils down into minimizing the number of nonzero columns of \mathbf{Y} , i.e., $\sum_{i=1}^{n_y} \|\text{col}(\mathbf{Y})_i\|_{\ell_0}$. Since ℓ_0 -norm is nonconvex, it is replaced by ℓ_1 -norm, yielding the following relaxed problem

$$(\mathbf{P7}) \quad \underset{\mathbf{P}, \mathbf{Y}, \epsilon, \gamma}{\text{minimize}} \quad \sum_{i=1}^{n_y} c_i w_i^{(k)} \|\text{col}(\mathbf{Y})_i\|_{\ell_1} \quad (24a)$$

$$\text{subject to } (4), \mathbf{P} \succ 0, \epsilon > 0, \quad (24b)$$

where $w_i^{(k)}$ is the corresponding weighting factor for column i at iteration k . When $\mathbf{w}^{(k)}$ is fixed, **P7** becomes a convex SDP. This weighting factor is updated at each iteration k using the solution from previous iteration through the following update rule [18]

$$w_i^{(k+1)} := \frac{1}{\|\text{col}(\mathbf{Y})_i^{(k)}\|_{\ell_1} + \kappa}, \quad \forall i = 1, 2, \dots, n_y, \quad (25)$$

where $\kappa > 0$ is a relatively small constant. Algorithm 3, adapted from [18], provides the detailed steps.

In this numerical experiment, both algorithms are configured such that the objective is to find at least $N - 1$ activated sensors. The ℓ_1 -norm relaxation is used with $\epsilon_w = 0.1$ and 500 maximum iterations. Two different values for κ are considered which correspond to two scenarios for this approach: ℓ_1 -norm relaxation-1 uses $\kappa = 10^{-4}$ while

Algorithm 3: Iterative ℓ_1 -Norm Relaxation for SSP

```

1 input:  $\epsilon_w, \kappa, \text{maxIter}$ 
2 initialize:  $k = 0, \mathbf{w}^{(1)} = \mathbf{1}$ 
3 do
4    $k \leftarrow k + 1$ 
5   solve P7 in (24), obtain  $\mathbf{Y}^{(k)}$ 
6   foreach  $i \in \mathbb{I}(n_y)$  do
7      $\lfloor$  update  $w_i^{(k+1)}$  from  $\mathbf{Y}^{(k)}$  using (25)
8 while  $|\|\mathbf{w}^{(k+1)}\|_2 - \|\mathbf{w}^{(k)}\|_2| > \epsilon_w$  and  $k < \text{maxIter}$ 
9 find: a feasible solution of LMI (24b) where  $\mathbf{Y}$ 
    corresponds to the nonzero column of  $\mathbf{Y}^{(k)}$  from the
    last iteration
10 output:  $\mathbf{P}, \mathbf{Y}$ 

```

ℓ_1 -norm relaxation-2 uses $\kappa = 10^{-5}$. The corresponding results are illustrated in Fig. 2. In particular, Fig. 2b indicates that our SE-BnB algorithm is able to find suboptimal sensor combinations with $N - 1$ activated sensors while the ℓ_1 -norm relaxation fails to find sensor combinations under such prescribed limitation. On the matter of computational time, as shown in Fig. 2a, the SE-BnB algorithm requires less computational time compared to the ℓ_1 -norm relaxation. This demonstrates an advantage of the proposed SE-BnB algorithm in finding suboptimal solutions for SSP over the ℓ_1 -norm relaxation technique, at least for the network of nonlinear unstable nodes described in (23).

7.3. Corroborating the Theory

Finally, we employ SE-BnB algorithm to find optimal solutions for SSP while using different values of Lipschitz constant. This numerical test is intended to find empirical relation between the value of Lipschitz constant and the number of activated sensors. For this purpose, we consider the so called Lipschitz multiplier $\eta > 0$ such that the Lipschitz constant for N number of subsystems is formulated as $\gamma_l = \eta\sqrt{N}$. Fig. 3 illustrates the corresponding results. It can be observed that increasing Lipschitz constant (that is, by equivalently increasing η) raises the number of activated sensors—this is indicated by the nondecreasing fashion in the number of sensor costs as the value of η increases. This result corroborates the theoretical foundation given in Section 4 and Theorem 2. That is, in the context of SSP, increasing Lipschitz constant shrinks the feasible space of the SSP problem, hence requiring more sensors to stabilize the estimation error dynamics.

8. Paper Summary and Limitations

A novel general framework for dealing with SASPs for NDSs is presented. Our approach is built upon SDP formulations for observer/controller designs for various classes of NDSs developed in the literature. Specifically, this paper focuses on addressing SASPs for Lipschitz NDSs and our investigations show that smaller Lipschitz constant allows less number SAs are required to achieve stabilization

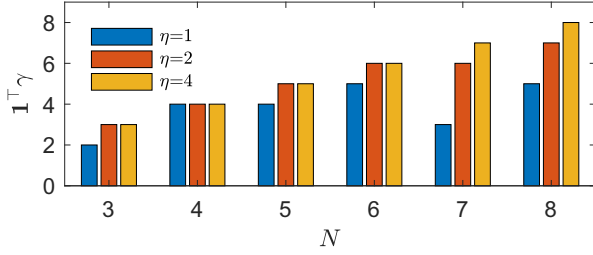


Figure 3: Optimal sensor costs computed through the SE-BnB for different Lipschitz multiplier η where the Lipschitz constant is given by $\gamma_l = \eta\sqrt{N}$.

in control/estimation purpose. A customized BnB algorithm, referred to as SE-BnB, which exploits problem structure and utilizing new heuristics to efficiently obtain optimal and suboptimal solutions for SASPs, is proposed. The main advantage of our framework as opposed to other approaches from the literature is its capability to obtain the optimal SAs combination by means of a BnB algorithm for stable/unstable NDS that ensures stability for either estimation and/or stabilization/tracking purposes.

The methods presented in this paper come with their limitations. First, this paper is not concerned with scaling the problem for extremely large-scale systems. This is a result of using SDPs and LMIs that still have serious scalability issues—in addition to the worst case complexity of BnB routines. Second, this paper also does *not* address the simultaneous selection of SAs and also assumes that all SAs are operating in ideal conditions (e.g. no saturation effect). To that end, our future work will focus on (i) investigating different approaches to solve SASPs other than using BnB algorithm such as the generalized Benders decomposition [28] and branch-and-cut algorithm [29], (ii) extending the proposed approach for addressing the robust SASPs as well as the simultaneous selection of SAs through output feedback control and observer-based control policies, and (iii) considering the effect of SAs saturation in SASPs.

Acknowledgments

This material is based upon work supported by the National Science Foundation (NSF) under Grants 1728629, 1917164, and 2044430. We gratefully acknowledge NSF's support and the thorough comments and suggestions made by the editor and the reviewers.

- [1] J. Qi, K. Sun, W. Kang, Optimal pmu placement for power system dynamic state estimation by using empirical observability gramian, *IEEE Transactions on power Systems* 30 (4) (2015) 2041–2054.
- [2] S. Pequito, N. Popli, S. Kar, M. D. Ilić, A. P. Aguiar, A framework for actuator placement in large scale power systems: Minimal strong structural controllability, in: 2013 5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2013, pp. 416–419.
- [3] T. H. Summers, J. Lygeros, Optimal sensor and actuator placement in complex dynamical networks, *IFAC Proceedings Volumes* 47 (3) (2014) 3784 – 3789, 19th IFAC World Congress. doi:https://doi.org/10.3182/20140824-6-ZA-1003.00226.
- [4] N. K. Dhingra, M. R. Jovanović, Z.-Q. Luo, An admm algorithm for optimal sensor and actuator selection, in: IEEE Conference on Decision and Control, IEEE, 2014, pp. 4039–4044.

- [5] A. F. Taha, N. Gatsis, T. Summers, S. A. Nugroho, Time-varying sensor and actuator selection for uncertain cyber-physical systems, *IEEE Transactions on Control of Network Systems* 6 (2) (2019) 750–762. doi:10.1109/TCNS.2018.2873229.
- [6] S. A. Nugroho, A. F. Taha, N. Gatsis, T. H. Summers, R. Krishnan, Algorithms for joint sensor and control nodes selection in dynamic networks, *Automatica* 106 (2019) 124 – 133. doi:https://doi.org/10.1016/j.automatica.2019.04.047.
- [7] S. A. Nugroho, A. F. Taha, On the need for sensor and actuator placement algorithms in nonlinear systems: Wip abstract, in: Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems, ICCPS '19, ACM, New York, NY, USA, 2019, pp. 304–305. doi:10.1145/3302509.3313313.
- [8] A. Haber, F. Molnar, A. E. Motter, State observation and sensor selection for nonlinear networks, *IEEE Transactions on Control of Network Systems* 5 (2) (2018) 694–708. doi:10.1109/TCNS.2017.2728201.
- [9] S. D. Bopardikar, O. Ennasr, X. Tan, Randomized sensor selection for nonlinear systems with application to target localization, *IEEE Robotics and Automation Letters* 4 (4) (2019) 3553–3560. doi:10.1109/LRA.2019.2928208.
- [10] A. Haber, S. A. Nugroho, P. Torres, A. F. Taha, Control node selection algorithm for nonlinear dynamic networks, *IEEE Control Systems Letters* 5 (4) (2021) 1195–1200.
- [11] A. Haber, Joint sensor node selection and state estimation for nonlinear networks and systems, *IEEE Transactions on Network Science and Engineering* (2021) 1–1doi:10.1109/TNSE.2021.3069890.
- [12] G. Phanomchoeng, R. Rajamani, The bounded jacobian approach to nonlinear observer design, in: Proceedings of the 2010 American Control Conference, 2010, pp. 6083–6088. doi:10.1109/ACC.2010.5531543.
- [13] G. Phanomchoeng, R. Rajamani, Observer design for lipschitz nonlinear systems using riccati equations, in: Proceedings of the 2010 American Control Conference, 2010, pp. 6060–6065. doi:10.1109/ACC.2010.5531294.
- [14] M. Abbaszadeh, H. J. Marquez, Nonlinear observer design for one-sided lipschitz systems, in: Proceedings of the 2010 American Control Conference, 2010, pp. 5284–5289. doi:10.1109/ACC.2010.5530715.
- [15] Y. Guo, Y. Chen, C. Zhang, Decentralized state-observer-based traffic density estimation of large-scale urban freeway network by dynamic model, *Information* 8 (3) (2017) 95.
- [16] S. A. Nugroho, A. F. Taha, Sensor placement strategies for some classes of nonlinear dynamic systems via lyapunov theory, in: 2019 Conference on Decision and Control (CDC), 2019.
- [17] J. Löfberg, YALMIP: A toolbox for modeling and optimization in matlab, in: Proc. IEEE Int. Symp. Computer Aided Control Syst. Design, 2004, pp. 284–289.
- [18] E. J. Candès, M. B. Wakin, S. P. Boyd, Enhancing sparsity by reweighted $\ell - 1$ minimization, *Journal of Fourier Analysis and Applications* 14 (5) (2008) 877–905. doi:10.1007/s00041-008-9045-x.
- [19] S. A. Nugroho, A. F. Taha, Towards understanding sensor and control nodes selection in nonlinear dynamic systems: Lyapunov theory meets branch-and-boundarXiv:2004.12061.
- [20] S. A. Nugroho, A. F. Taha, C. G. Claudel, A control-theoretic approach for scalable and robust traffic density estimation using convex optimization, *IEEE Transactions on Intelligent Transportation Systems* 22 (1) (2021) 64–78. doi:10.1109/TITS.2019.2953023.
- [21] S. A. Nugroho, A. F. Taha, V. Hoang, Nonlinear dynamic systems parameterization using interval-based global optimization: Computing lipschitz constants and beyond (2020). arXiv:2004.12061.
- [22] G. P. McCormick, Computability of global solutions to factorable nonconvex programs: Part i—convex underestimating problems, *Mathematical programming* 10 (1) (1976) 147–175.
- [23] M. Yadegar, A. Afshar, M. Davoodi, Observer-based tracking controller design for a class of lipschitz nonlinear systems, *Journal of Vibration and Control* 24 (11) (2018) 2112–2119.

- doi:10.1177/1077546317721597.
- [24] W. Zhang, H. Su, H. Wang, Z. Han, Full-order and reduced-order observers for one-sided lipschitz nonlinear systems using riccati equations, *Communications in Nonlinear Science and Numerical Simulation* 17 (12) (2012) 4968–4977.
 - [25] E. D. Andersen, K. D. Andersen, *The Mosek Interior Point Optimizer for Linear Programming: An Implementation of the Homogeneous Algorithm*, Springer US, Boston, MA, 2000, pp. 197–232.
 - [26] N. Motee, A. Jadbabaie, Optimal control of spatially distributed systems, *IEEE Transactions on Automatic Control* 53 (7) (2008) 1616–1629. doi:10.1109/TAC.2008.929366.
 - [27] A. Argha, S. W. Su, A. Savkin, B. Celler, A framework for optimal actuator/sensor selection in a control system, *International Journal of Control* 92 (2) (2019) 242–260. doi:10.1080/00207179.2017.1350755.
 - [28] J. Zhang, X. Wang, D. J. Chmielewski, Covariance-based hardware selection part iv: Solution using the generalized benders decomposition, *AIChE Journal* 62 (10) (2016) 3628–3638. doi:10.1002/aic.15285.
 - [29] K. Kobayashi, Y. Takano, A branch-and-cut algorithm for solving mixed-integer semidefinite optimization problems, *Computational Optimization and Applications* 75 (2) (2020) 493–513.