Automatica 108 (2019) 108487

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Byzantine-resilient distributed observers for LTI systems[☆]

Aritra Mitra [*], Shreyas Sundaram

School of Electrical and Computer Engineering at Purdue University, West Lafayette, IN 47907, USA

ARTICLE INFO

Article history:
Received 28 February 2018
Received in revised form 21 April 2019
Accepted 21 June 2019
Available online 12 July 2019

Keywords:
Resilient state estimation
Distributed estimation
Byzantine attacks

ABSTRACT

Consider a linear time-invariant (LTI) dynamical system monitored by a network of sensors, modeled as nodes of an underlying directed communication graph. We study the problem of collaboratively estimating the state of the system when certain nodes are compromised by adversaries. Specifically, we consider a Byzantine adversary model, where a compromised node possesses complete knowledge of the system dynamics and the network, and can deviate arbitrarily from the rules of any prescribed algorithm. We first characterize certain fundamental limitations of any distributed state estimation algorithm in terms of the measurement and communication structure of the nodes. We then develop an attack-resilient, provably correct state estimation algorithm that admits a fully distributed implementation. To characterize feasible network topologies that guarantee success of our proposed technique, we introduce a notion of 'strong-robustness' that captures both measurement and communication redundancy. Finally, by drawing connections to bootstrap percolation theory, we argue that given an LTI system and an associated sensor network, the 'strong-robustness' property can be checked in polynomial time.

1. Introduction

The control of large-scale complex networked systems such as power grids, transportation networks, and multi-agent robotic systems requires precise estimation of the state of the underlying dynamical process. Typically, in these applications, sensors (nodes) collecting information about the process are scattered over a geographical region. As the diameters of such networks increase, routing information from all the sensors to a central computational resource induces large delays and creates communication bottlenecks. To bypass these difficulties, it thus becomes important to consider distributed algorithms where individual sensors communicate only with sensors within a given distance. However, the potential merits (Estrin, Govindan, Heidemann, & Kumar, 1999) of such a distributed approach are matched by various challenges. In particular, a key challenge is to design networks and distributed algorithms that guarantee reliable operation of the system in the face of faults or sophisticated adversarial attacks on certain sensors. This leads to the motivation behind our present work.

In a classical distributed state estimation setup, each node receives partial measurements of the state of an LTI process, and seeks to asymptotically estimate the entire state by exchanging information with its neighbors in the network. For background, we direct the interested reader to recent work on single-time-scale distributed observers in del Nozal, Millán, Orihuela, Seuret, and Zaccarian (2019), Han, Trentelman, Wang, and Shen (2019), Kim, Shim, and Cho (2016), Mitra and Sundaram (2018b), Park and Martins (2017), Wang and Morse (2018) and Wang and Ren (2017), and distributed Kalman filtering in Battistelli and Chisci (2014), Battistelli, Chisci, Mugnai, Farina, and Graziano (2015), Kamal, Farrell, Roy-Chowdhury, et al. (2013), Khan and Moura (2008), Matei and Baras (2012) and Olfati-Saber (2005). However, none of these papers address the challenges associated with tolerating unreliable components in the network. Accordingly, we now provide a survey of the cyber-security literature that is most relevant to our present cause.

Related work: Over the last decade, a significant amount of research has focused on security in networked control systems. In particular, for noiseless dynamical systems, it has been established that zero-dynamics play a key role in characterizing the stealth of an attack (Pasqualetti, Dörfler, & Bullo, 2013; Sundaram & Hadjicostis, 2011). For networked control systems affected by noise, the authors in Bai, Pasqualetti, and Gupta (2017) recently introduced an information-theoretic metric that quantifies the detectability of an attack. A unifying feature of Bai et al. (2017), Pasqualetti et al. (2013), and the ones in Mishra, Shoukry, Karamchandani, Diggavi, and Tabuada (2017), Mo and Sinopoli (2015),

[☆] This research was supported in part by NSF CAREER award 1653648. The material in this paper was partially presented at the 55th IEEE Conference on Decision and Control, Dec 12–14, Las Vegas, NV, USA. This paper was recommended for publication in revised form by Associate Editor Julien M. Hendrickx under the direction of Editor Christos G. Cassandras.

[*] Corresponding author.
E-mail addresses: mitra14@purdue.edu (A. Mitra), sundara2@purdue.edu (S. Sundaram).

https://doi.org/10.1016/j.automatica.2019.06.039
0005-1098/

Pajic, Lee, and Pappas (2017) and Teixeira, Shames, Sandberg, and Johansson (2015), is that they involve systems where all the sensor measurements are available at a single location. In the sequel, we shall refer to such systems as centralized control systems. Our problem formulation and subsequent analysis differs from the above literature by constraining each sensor to exchange information with only its neighbors in the communication graph. Some recent related work on resilient distributed parameter estimation and resilient decentralized hypothesis testing are reported in Chen, Kar, and Moura (2018), Su and Shahrampour (2018) and Hashlamoun, Brahma, and Varshney (2018), respectively; each of these works considers worst-case sensor attack models. The authors in Forti, Battistelli, Chisci, Li, Wang, and Sinopoli (2018) consider the problem of joint attack detection and state estimation over a cluster-based sensor network configuration. Specifically, they consider signal, packet substitution and extra packet injection attacks, each of which falls within the purview of the worst-case attack model studied in this paper.

While the study of security in centralized control systems is now mature, a comprehensive theoretical understanding of analogous questions in a distributed setting is still lacking. Preliminary attempts to counter adversarial behavior in a distributed state estimation context are reported in Matei, Baras, and Srinivasan (2012) and Khan and Stankovic (2013). However, unlike our results, these papers consider attack models that are limited in scope, and neither provide any theoretical guarantees of success, nor allude to graph-theoretic conditions that are needed for their respective algorithms to work. Recently, in Deghat, Ugrinovskii, Shames, and Langbort (2019), the authors employ an H_∞ based approach for detecting biasing attacks in distributed estimation networks. Specifically, the authors consider a scenario where the attacker attacks the observer dynamics at certain nodes directly, i.e., the data processing algorithm at such compromised nodes is subjected to spurious attack signals. Our present work deviates from Deghat et al. (2019) in several aspects, namely: (i) while the analysis in Deghat et al. (2019) is limited to a certain class of attack inputs, our attack model allows compromised nodes to behave *arbitrarily*, i.e., no restrictions are placed on the inputs that can be injected by an adversary, (ii) unlike Deghat et al., 2019, we develop a filtering algorithm that allows each uncompromised node to asymptotically recover the state of the plant *without* explicitly detecting the nodes under attack, and (iii) the existence of the attack detection filter proposed in Deghat et al. (2019) relies on solving an LMI; however, the authors neither provide graph-theoretic insights regarding the solvability of such an LMI nor discuss whether the LMI can be solved in a distributed manner. In contrast, we detail graph-theoretic conditions that allow each step of our approach to have a resilient, distributed implementation. Summing up, this paper attempts to bridge the gap between centralized and distributed resilient state estimation via the following contributions.

Contributions: Our contributions are threefold. First, in Section 3, we characterize certain necessary conditions that need to be satisfied by the sensor measurements and the communication graph for the distributed state estimation problem to be solvable in the presence of arbitrary adversarial behavior. Our results hold for *any* algorithm and hence identify fundamental limitations that are of both theoretical and practical importance in the design of attack-resilient robust networks. We also argue that our impossibility results in the distributed setting generalize those existing for centralized control systems subject to sensor attacks (Chong, Wakaiki, & Hespanha, 2015; Fawzi, Tabuada, & Diggavi, 2014). Our second contribution is to develop a distributed filtering algorithm in Section 4 that enables each uncompromised node to recover the entire state, provided certain graph conditions are met. A thorough analysis of the proposed filtering scheme is then presented in Section 5. As our third contribution, in Section 7, we introduce a topological property called 'strong-robustness' to characterize feasible systems and networks that guarantee applicability of our approach. By drawing connections to bootstrap percolation theory, we show that the 'strong-robustness' property can be checked in polynomial time (in the size of the system and the network).

Comparison with prior work by the authors: We reported certain preliminary results in Mitra and Sundaram (2016). In this paper, we significantly expand upon our prior work in the following ways. (i) While the analysis in Mitra and Sundaram (2016) was limited to system matrices with real, distinct eigenvalues, our present framework allows the system matrix to have arbitrary spectrum. This generalization (accounting for complex, possibly repeated eigenvalues) requires various appropriate modifications to the algorithm developed in Mitra and Sundaram (2016), along with a more detailed technical analysis. (ii) Section 3 is a new addition entirely, and discusses fundamental limitations of any distributed state estimation algorithm in the face of arbitrary adversarial attacks. (iii) Section 7 contains additional details about properties of feasible network topologies, and establishes the key result that the topological condition (namely 'strong-robustness') needed for implementing our proposed algorithm can be checked in polynomial time. The latter result (missing in Mitra & Sundaram, 2016) is particularly important since it highlights the applicability of our overall approach.

Notation: A directed graph is denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \dots, N\}$ is the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represents the edges. An edge from node j to node i, denoted by (j, i), implies that node j can transmit information to node i. The neighborhood of the i-th node is defined as $\mathcal{N}_i \triangleq \{j \in \mathcal{V} \mid (j, i) \in \mathcal{E}\}$. A node j is said to be an out-neighbor of node i if $(i, j) \in \mathcal{E}$. By an *induced* subgraph of $\mathcal{G}$ obtained by removing certain nodes $\mathcal{C} \subset \mathcal{V}$, we refer to the subgraph that has $\mathcal{V} \setminus \mathcal{C}$ as its node set and contains only those edges of $\mathcal{E}$ with both end points in $\mathcal{V} \setminus \mathcal{C}$. The notation $|\mathcal{V}|$ is used to denote the cardinality of a set $\mathcal{V}$. The set of all eigenvalues (or modes) of a matrix $\mathbf{A}$ is denoted by $sp(\mathbf{A}) = \{\lambda \in \mathbb{C} \mid det(\mathbf{A} - \lambda \mathbf{I}) = 0\}$, and the set of all unstable eigenvalues by $\Lambda_U(\mathbf{A}) = \{\lambda \in sp(\mathbf{A}) \mid |\lambda| \geq 1\}$. We use $a_{\mathbf{A}}(\lambda)$ and $g_{\mathbf{A}}(\lambda)$ to denote the algebraic and geometric multiplicities, respectively, of an eigenvalue $\lambda \in sp(\mathbf{A})$. An eigenvalue λ is said to be simple if $a_{\mathbf{A}}(\lambda) = g_{\mathbf{A}}(\lambda) = 1$. Given a set of matrices $\{\mathbf{A}_1, \dots, \mathbf{A}_n\}$, we use $diag(\mathbf{A}_1, \dots, \mathbf{A}_n)$ to refer to a block diagonal matrix with $\mathbf{A}_i$ as its i-th block entry. For a set $\mathcal{J} = \{m_1, \dots, m_{|\mathcal{J}|}\} \subseteq \{1, \dots, N\}$, and a matrix $\mathbf{C} = \begin{bmatrix} \mathbf{C}_1^T & \cdots & \mathbf{C}_N^T \end{bmatrix}^T$, we define $\mathbf{C}_{\mathcal{J}} \triangleq \begin{bmatrix} \mathbf{C}_{m_1}^T & \cdots & \mathbf{C}_{m_{|\mathcal{J}|}}^T \end{bmatrix}^T$. The identity matrix of dimension r is denoted $\mathbf{I}_r$, and $\mathbb{N}_+$ is used to refer to the set of all positive integers. The terms 'communication graph' and 'network' are used interchangeably, and the term 'resilient' is used in the same context as that used traditionally in the computer science literature to deal with worst-case adversarial attack models (Dolev, Lynch, Pinter, Stark, & Weihl, 1986).

2. System and attack model

System model: Consider the LTI dynamical system

$$\mathbf{x}[k+1] = \mathbf{A}\mathbf{x}[k], \tag{1}$$

where $k \in \mathbb{N}$ is the discrete-time index, $\mathbf{x}[k] \in \mathbb{R}^n$ is the state vector and $\mathbf{A} \in \mathbb{R}^{n \times n}$ is the system matrix. The system is monitored by a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consisting of N nodes. The i-th node receives a measurement of the state, given by

$$\mathbf{y}_i[k] = \mathbf{C}_i \mathbf{x}[k], \tag{2}$$

where $\mathbf{y}_i[k] \in \mathbb{R}^{r_i}$ and $\mathbf{C}_i \in \mathbb{R}^{r_i \times n}$. We use $\mathbf{C} = \begin{bmatrix}\mathbf{C}_1^T & \cdots & \mathbf{C}_N^T\end{bmatrix}^T$ to represent the collection of the individual node observation matrices; accordingly, $\mathbf{y}[k] = \begin{bmatrix}\mathbf{y}_1^T[k] & \cdots & \mathbf{y}_N^T[k]\end{bmatrix}^T$ represents the collective measurement vector, i.e., $\mathbf{y}[k] = \mathbf{C}\mathbf{x}[k]$.[1]

Each node is tasked with estimating the entire system state $\mathbf{x}[k]$ based on information received from its neighbors and its local measurements (if any). As such, we assume that the pair $(\mathbf{A}, \mathbf{C})$ is detectable (this is a necessary condition for solving the distributed state estimation problem even in the absence of adversaries); however, we do not assume that the pair $(\mathbf{A}, \mathbf{C}_i)$ is detectable for any $i \in \mathcal{V}$. Two immediate challenges are as follows: (i) As the pair $(\mathbf{A}, \mathbf{C}_i)$ may not be detectable for some (or all) $i \in \{1, \ldots, N\}$, information exchange is necessary; and (ii) information exchange is restricted by the underlying communication graph $\mathcal{G}$. In addition to the above challenges, in this paper, we allow for the possibility that certain nodes in the network are compromised by an adversary, and *do not* follow their prescribed state estimate update rule. We will use the following adversary model in this paper.

Adversary model: We consider a subset $\mathcal{A} \subset \mathcal{V}$ of the nodes in the network to be adversarial. We assume that the adversarial nodes are completely aware of the network topology, the system dynamics and the algorithm employed by the non-adversarial nodes. Such an assumption of omniscient adversarial behavior is standard in the literature on resilient distributed algorithms (Koo, 2004; LeBlanc, Zhang, Koutsoukos, & Sundaram, 2013; Lynch, 1996; Pagourtzis, Panagiotakos, & Sakavalas, 2017; Pasqualetti, Bicchi, & Bullo, 2012; Pelc & Peleg, 2005; Sundaram & Hadjicostis, 2011; Vaidya, Tseng, & Liang, 2012), and allows us to provide guarantees against "worst-case" adversarial behavior. In terms of capabilities, an adversarial node can leverage the aforementioned information to arbitrarily deviate from the rules of any prescribed algorithm, while colluding with other adversaries in the process. Furthermore, following the Byzantine fault model (Dolev et al., 1986), adversaries are allowed to send differing state estimates to different neighbors at the same instant of time. To characterize the threat model in terms of the number of adversaries in the network, we will use the following definitions from Koo (2004) and Pelc and Peleg (2005).

Definition 1 (*f-total set*)**.** A set $\mathcal{C} \subset \mathcal{V}$ is *f-total* if it contains at most f nodes in the network, i.e., $|\mathcal{C}| \leq f$.

Definition 2 (*f-local set*)**.** A set $\mathcal{C} \subset \mathcal{V}$ is *f-local* if it contains at most f nodes in the neighborhood of the other nodes, i.e., $|\mathcal{N}_i \cap \mathcal{C}| \leq f, \forall i \in \mathcal{V} \setminus \mathcal{C}$.

Definition 3 (*f-local and f-total adversarial models*)**.** A set $\mathcal{A}$ of adversarial nodes is *f-locally bounded* (resp., f-totally bounded) if $\mathcal{A}$ is an f-local (resp., f-total) set.

In the literature dealing with distributed fault-tolerant algorithms, it is a common assumption to consider an f-total adversarial model. However, to allow for a large number of adversaries in large scale networks, we will allow the adversarial set to be f-local. Summarily, the adversary model considered throughout this paper will be referred to as an f-locally bounded Byzantine adversary model. The non-adversarial nodes will be referred to as regular nodes and be represented by the set $\mathcal{R} = \mathcal{V} \setminus \mathcal{A}$. Note that the actual number and identities of the adversarial nodes are not known to the regular nodes. As is standard, any reliable system is designed to provide a desired level of resilience against a maximum number of component failures or attacks. We share the same philosophy. Specifically, we assume that each node in the network is programmed to tolerate up to a maximum of f adversaries in the entire network (in an f-total model) or in its own neighborhood (in an f-local model). Such an assumption is typical in the design of distributed protocols (see for instance Koo, 2004; LeBlanc et al., 2013; Lynch, 1996; Pagourtzis et al., 2017; Pasqualetti et al., 2012; Pelc & Peleg, 2005; Sundaram & Hadjicostis, 2011; Vaidya et al., 2012) that are resilient to worst-case Byzantine attack models like the one considered in this paper.

Throughout this paper, we shall only consider causal (i.e., nodes act only on past and present information), synchronous (i.e., all nodes share a common clock w.r.t. their iterates), and deterministic (i.e., given the same input, such algorithms generate the same output) algorithms; note however that the notions of causal and deterministic behavior apply only to the regular nodes. We shall also assume that all quantities being updated iteratively by the regular nodes are initialized identically in each execution. With $\hat{\mathbf{x}}_i[k]$ representing the estimate of $\mathbf{x}[k]$ maintained by node i, the problem studied in this paper can be formally stated as follows.

Problem 1 (*Resilient Distributed State Estimation*)**.** Given an LTI system (1), a linear measurement model (2), and a time-invariant directed communication graph $\mathcal{G}$, design a set of state estimate update and information exchange rules such that $\lim_{k\to\infty} \|\hat{\mathbf{x}}_i[k] - \mathbf{x}[k]\| = 0, \forall i \in \mathcal{R}$, *regardless* of the actions of any f-locally bounded set of Byzantine adversaries.

The interplay between the measurement structure of the nodes and the underlying communication graph results in certain conditions being necessary for solving Problem 1, irrespective of the choice of algorithms. We provide such conditions in the following section.

3. Fundamental limitations of any distributed state estimation algorithm

Intuitively, the network must possess a certain degree of measurement redundancy as well as redundancy in its communication structure so as to counteract the effects of adversarial behavior. More specifically, the measurements of the regular nodes must ensure collective detectability of the state, and the network structure should prevent the malicious nodes from acting as bottlenecks between correctly functioning nodes. To identify necessary conditions for resilient distributed state estimation that capture the above notions of redundancy, we first introduce some terminology.

Definition 4 (*Critical Set*)**.** A set of nodes $\mathcal{F} \subset \mathcal{V}$ is said to be a critical set if the pair $(\mathbf{A}, \mathbf{C}_{\mathcal{V}\setminus\mathcal{F}})$ is not detectable.

Note that detectability of $(\mathbf{A}, \mathbf{C})$ implies that a critical set must necessarily be non-empty.

Definition 5 (*Minimal Critical Set*)**.** A set $\mathcal{F} \subset \mathcal{V}$ is said to be a minimal critical set if $\mathcal{F}$ is a critical set and no subset of $\mathcal{F}$ is a critical set.

Let $\mathcal{M} = \{\mathcal{F}_1, \ldots, \mathcal{F}_{|\mathcal{M}|}\}$ denote the set of all minimal critical sets. With each set $\mathcal{F}_i \in \mathcal{M}$, we associate a virtual node s_i as follows. Directed edges are added from s_i to each node in $\mathcal{F}_i$ and the resulting network is denoted by $\mathcal{G}'_i = (\mathcal{V} \cup s_i, \mathcal{E} \cup \mathcal{E}_i)$, where $\mathcal{E}_i$ represents the set of edges from s_i to $\mathcal{F}_i$.

[1] Like our present formulation, Chen et al. (2018), Deghat et al. (2019), Khan and Stankovic (2013), Matei et al. (2012) and Su and Shahrampour (2018) also consider linear system and measurement models.

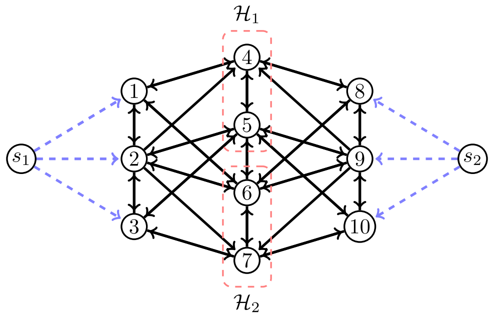


Fig. 1. A 2-dimensional LTI system with two distinct, real, unstable eigenvalues (modes) λ_1, λ_2 is monitored by a network $\mathcal{G}$ of 10 nodes as shown above. Mode λ_1 (resp., mode λ_2) is only detectable w.r.t. the measurements of nodes 1–3 (resp., nodes 8–10). Thus, the two minimal critical sets associated with the above system and network are $\mathcal{F}_1 = \{1, 2, 3\}$ and $\mathcal{F}_2 = \{8, 9, 10\}$. An example of a set that is critical, but not minimal, is $\{1, 2, 3, 8\}$. The virtual source nodes associated with $\mathcal{F}_1$ and $\mathcal{F}_2$ are s_1 and s_2, respectively. There are no 1-total pair cuts w.r.t. s_1 or s_2. The set $\mathcal{H} = \{4, 5, 6, 7\}$ is a 1-local pair cut w.r.t. both s_1 and s_2 since $\mathcal{H}$ can be partitioned into $\mathcal{H}_1 = \{4, 5\}$ and $\mathcal{H}_2 = \{6, 7\}$, each of which each of which is a 1-local set. Since $\mathcal{H}_1$ and $\mathcal{H}_2$ are each 2-total sets, $\mathcal{H}$ is also a 2-total pair cut w.r.t. both s_1 and s_2.

Definition 6 (*f-local pair and f-total pair cuts w.r.t. s_i*)**.** Consider a minimal critical set $\mathcal{F}_i \in \mathcal{M}$. A set $\mathcal{H} \subset \mathcal{V}$ is called a cut w.r.t. s_i if removal of $\mathcal{H}$ from $\mathcal{G}'_i$ results in an induced subgraph of $\mathcal{G}'_i$ whose node set can be partitioned into two non-empty sets $\mathcal{X}$ and $\mathcal{Y}$ with $s_i \in \mathcal{X}$, and no directed paths from $\mathcal{X}$ to $\mathcal{Y}$ in the induced subgraph. A cut $\mathcal{H}$ w.r.t. s_i is called an f-local pair cut (resp., f-total pair cut) w.r.t. s_i if it can be partitioned as $\mathcal{H} = \mathcal{H}_1 \cup \mathcal{H}_2$ such that both $\mathcal{H}_1$ and $\mathcal{H}_2$ are f-local (resp., f-total) in $\mathcal{G}$.

For an illustration of the above definitions, see Fig. 1. The following result identifies a fundamental limitation for f-local adversarial models.

Theorem 1. *Suppose there exists an f-local pair cut w.r.t. s_i in $\mathcal{G}'_i$ for some minimal critical set $\mathcal{F}_i \in \mathcal{M}$. Then, it is impossible for any causal, synchronous, and deterministic algorithm to solve Problem 1.*

Proof. Suppose there exists an f-local pair cut $\mathcal{H} = \mathcal{H}_1 \cup \mathcal{H}_2$ w.r.t. s_i for some minimal critical set $\mathcal{F}_i \in \mathcal{M}$. For the sake of contradiction, suppose there exists a causal, synchronous, and deterministic algorithm $\mathcal{T}$ that solves Problem 1 for the given network $\mathcal{G}$. From the definition of $\mathcal{H}$, we see that $\mathcal{Y}$ contains no elements of $\mathcal{F}_i$. Since $\mathcal{F}_i$ is a critical set, it then follows that the pair $(\mathbf{A}, \mathbf{C}_{\mathcal{Y}})$ is not detectable. Thus, there exists an initial condition $\mathbf{x}[0] = \boldsymbol{\eta}$ that causes the measurement set $\mathbf{y}_{\mathcal{Y}}[k]$ corresponding to $\mathcal{Y}$ to be identically zero for all time, while the state $\mathbf{x}[k]$ remains bounded away from zero. The idea of the proof will be to demonstrate that the nodes in $\mathcal{Y}$ cannot distinguish between the zero initial condition and the initial condition $\boldsymbol{\eta}$ under an appropriately constructed attack. To this end, noting that each of the sets $\mathcal{H}_1$ and $\mathcal{H}_2$ is f-local and can hence act as valid adversarial sets, we consider the following executions σ and σ' of $\mathcal{T}$.

Execution σ: The initial condition is $\mathbf{x}[0] = \mathbf{0}$. The nodes in $\mathcal{H}_1$ are regular while the nodes in $\mathcal{H}_2$ are adversarial. The nodes in $\mathcal{H}_2$ pretend that their state estimates are $\hat{\mathbf{x}}_{\mathcal{H}_2}[k]$ and that their measurements are $\mathbf{C}_{\mathcal{H}_2}\mathbf{A}^k\boldsymbol{\eta}$, where $\hat{\mathbf{x}}_{\mathcal{H}_2}[k]$ represents the collection of the state estimates maintained by the nodes in $\mathcal{H}_2$ during the execution σ' of $\mathcal{T}$. Additionally, at each time-step, the nodes in $\mathcal{H}_2$ perform the exact same actions that they perform during the execution σ'.

Execution σ': The initial condition is $\mathbf{x}[0] = \boldsymbol{\eta}$. The nodes in $\mathcal{H}_2$ are regular while the nodes in $\mathcal{H}_1$ are adversarial. The nodes in $\mathcal{H}_1$ pretend that their state estimates are $\hat{\mathbf{x}}_{\mathcal{H}_1}[k]$ and that their measurements are zero, where $\hat{\mathbf{x}}_{\mathcal{H}_1}[k]$ represents the collection of the state estimates maintained by the nodes in $\mathcal{H}_1$ during the execution σ of $\mathcal{T}$. Additionally, at each time-step, the nodes in $\mathcal{H}_1$ perform the exact same actions that they perform during the execution σ.

Since the actions of the adversaries in the two executions described above are coupled, it becomes important to establish that such actions are in fact well-defined. To do so, we argue as follows. Consider the actions of the adversarial set $\mathcal{H}_2$ at time $k = 0$ of execution σ. Due to their omniscient nature, these adversaries can anticipate the information that a regular set $\mathcal{H}_2$ is supposed to transmit at time $k = 0$ of execution σ' based on algorithm $\mathcal{T}$. Thus, their actions are well-defined at time $k = 0$. Note that the last two statements rely on the deterministic nature of $\mathcal{T}$. Specifically, under a deterministic algorithm $\mathcal{T}$, the actions of the regular nodes are also deterministic, and hence, can be predicted in advance by an omniscient adversary who is aware of the information set available to such regular nodes. An identical argument defines the actions of the adversarial set $\mathcal{H}_1$ at time $k = 0$ of execution σ'. Since the actions of both the sets $\mathcal{H}_1$ and $\mathcal{H}_2$ at time $k = 0$ are well-defined in each of the executions σ and σ', the response of the regular nodes to such actions (in the respective executions) at time $k = 1$ can be anticipated by any adversarial set. Specifically, to generate their actions at time $k = 1$ of execution σ (resp., execution σ'), the adversarial set $\mathcal{H}_2$ (resp., $\mathcal{H}_1$) simply simulates execution σ' (resp., execution σ) for time $k = 0$ to figure out how a regular set $\mathcal{H}_2$ (resp., $\mathcal{H}_1$) would act at time $k = 1$ of execution σ' (resp., execution σ). Repeating the above argument reveals that the actions of the respective adversarial sets in each of the executions σ and σ' are well-defined at every time step.

Based on the attack described above, it is clear that the nodes in $\mathcal{Y}$ receive the same state estimate and measurement information from the nodes in $\mathcal{H}$ in each of the two executions. Further, their own measurements are identically zero for all time in each of the two executions. Hence, based on such identical information, it is impossible for the nodes in $\mathcal{Y}$ to resolve the difference in the underlying initial conditions via algorithm $\mathcal{T}$. This leads to the desired contradiction and completes the proof. □

Remark 1. Interestingly, the necessary condition presented in the above theorem bears close resemblance to the necessary condition in Pagourtzis et al. (2017) and Pelc and Peleg (2005) for resilient broadcasting subject to the same f-local Byzantine adversary model that we consider here. This similarity can be attributed to the following analogy: viewing the virtual nodes as originators of messages in a broadcasting context, Problem 1 can be interpreted as a version of the resilient broadcasting problem where the regular nodes are required to agree (asymptotically) on a time-varying message that captures the state evolution of the system.

Our next result provides a necessary condition for an f-total (and hence, also an f-local) adversarial model.

Theorem 2. *Suppose there exists a causal, synchronous, and deterministic algorithm that solves the variant of Problem 1 corresponding to an f-total Byzantine adversary model. Then, the following equivalent statements are true.*

(i) Consider any minimal critical set $\mathcal{F}_i \in \mathcal{M}$. There exists no f-total pair cut w.r.t. s_i.
(ii) Consider any node $i \in \mathcal{V}$ such that $(\mathbf{A}, \mathbf{C}_i)$ is not detectable. Let $\mathcal{X}_i$ denote the set of all nodes in $\mathcal{G}$ that have directed paths to node i, and consider a set $\mathcal{D}_i \subseteq \mathcal{X}_i$ such that $|\mathcal{D}_i| \leq 2f$. Let $\mathcal{P}_i \subseteq \mathcal{X}_i$ represent the set of nodes that have directed paths to node i in the induced subgraph obtained by removing $\mathcal{D}_i$ from $\mathcal{G}$. Then, $(\mathbf{A}, \mathbf{C}_{i \cup \mathcal{P}_i})$ is detectable.

The proof of necessity mimics the proof of Theorem 1, while the equivalence between the two conditions stated in Theorem 2 is established in Appendix A.

Remark 2. In Chong et al. (2015) and Fawzi et al. (2014), the authors showed that for centralized systems subject to f sensor attacks, a necessary condition for estimating the state asymptotically is that the system should remain detectable after the removal of any $2f$ sensors. In our present distributed setting, the maximum information about the state that any given node i can hope to obtain is from the set $\{i \cup \mathcal{X}_i\}$, where $\mathcal{X}_i$ is as defined in Theorem 2. Thus, the second part of Theorem 2 generalizes the necessary conditions in Chong et al. (2015) and Fawzi et al. (2014). In Pasqualetti et al. (2012) and Sundaram and Hadjicostis (2011), the authors established that the connectivity of the graph plays a pivotal role in the analysis of fault-tolerant and resilient distributed consensus algorithms for settings where there are no underlying state dynamics that need to be estimated. The results stated in Theorems 1 and 2 differ from those in Pasqualetti et al. (2012) and Sundaram and Hadjicostis (2011) since the former blend both graph-theoretic and system-theoretic requirements. Finally, it can be easily shown that when there are no adversaries, i.e., when $f = 0$, the conditions identified in Theorem 2 reduce to the necessary and sufficient condition for distributed state estimation, namely every source component (strong components with no incoming edges) of the graph should be detectable (Mitra & Sundaram, 2018b; del Nozal et al., 2019; Park & Martins, 2017; Wang & Morse, 2018).

We now discuss certain implications of Theorem 2. Given an LTI system (1), a measurement model specified by (2), and a communication graph $\mathcal{G}$, it is of both theoretical and practical interest to know the maximum number of adversaries that can be tolerated when one seeks to solve Problem 1. Leveraging Theorem 2, we can provide an upper bound on this number, as follows.

Corollary 1. *Let k denote the smallest positive integer such that there exists a k-total pair cut w.r.t. s_i for some $\mathcal{F}_i \in \mathcal{M}$. Then, the total number of adversaries f must satisfy the inequality $f < k$ for Problem 1 to have a solution.*[2]

Corollary 2. *The condition $|\mathcal{F}_i| \geq (2f+1)\, \forall \mathcal{F}_i \in \mathcal{M}$ is necessary for resilient distributed state estimation subject to the f-local or f-total adversarial model.*

The proofs of the above results are straightforward and are hence omitted here. With the above corollaries in hand, one can gain insights regarding the distribution of certain specific critical sets in the network. To do so, given an eigenvalue $\lambda_j \in \Lambda_U(\mathbf{A})$, let $\{\boldsymbol{\rho}^{(j)}_1, \ldots, \boldsymbol{\rho}^{(j)}_{g_{\mathbf{A}}(\lambda_j)}\}$ represent a basis for the null space of $(\mathbf{A} - \lambda_j \mathbf{I}_n)$, and let $\phi^{(j)}_i = span\{\boldsymbol{\rho}^{(j)}_i\}, i \in \{1, \ldots, g_{\mathbf{A}}(\lambda_j)\}$. We say that node i can detect the subspace $\phi^{(j)}_i$ if $\mathbf{C}_i \boldsymbol{\rho}^{(j)}_i \neq \mathbf{0}$.[3] Let $\mathcal{W}^{(j)}_i \subseteq \mathcal{V}$ denote the set of all nodes that can detect $\phi^{(j)}_i$. The next result follows readily from Corollary 2 and the classical PBH test (Hespanha, 2009), and identifies a necessary redundancy requirement on the set $\mathcal{W}^{(j)}_i$, whenever this set is a strict subset of $\mathcal{V}$.

Proposition 1. *For each $\lambda_j \in \Lambda_U(\mathbf{A})$, if $\mathcal{W}^{(j)}_i \subset \mathcal{V}$, where $1 \leq i \leq g_{\mathbf{A}}(\lambda_j)$, then $|\mathcal{W}^{(j)}_i| \geq (2f + 1)$ is a necessary condition for resilient distributed state estimation subject to the f-local or f-total adversarial model.*

For systems with distinct eigenvalues, a direct consequence of the above result is the requirement of at least $(2f + 1)$ nodes that can detect each unstable eigenvalue of the system. The preceding analysis builds up to the distributed estimation strategy adopted in this paper. In particular, our approach involves identifying the locally detectable and undetectable eigenvalues associated with a given node, and subsequently devising separate estimation strategies for the subspaces associated with such eigenvalues. We formalize this idea in the next section.

Remark 3. Two important directions of future investigation are (i) finding an efficient algorithm (if one exists) for computing k in Corollary 1 either exactly or approximately, and (ii) determining whether the conditions stated in Theorem 1 (resp., Theorem 2) are sufficient for achieving resilient distributed state estimation subject to an f-local (resp., f-total) Byzantine adversary model. Note that the main source of computational complexity associated with the first point lies in finding all the minimal critical sets associated with the given system.

4. Resilient distributed state estimation

4.1. Preliminaries

For each eigenvalue $\lambda \in sp(\mathbf{A})$, let $\mathbf{V}(\lambda)$ represent a block diagonal matrix with the Jordan blocks corresponding to λ (in the standard Jordan canonical representation of $\mathbf{A}$) along the main block diagonal. We begin by recalling certain properties of the real Jordan canonical form of a square matrix that will be useful for our subsequent development (Horn & Johnson, 2012). We first note that if λ represents a non-real eigenvalue of $\mathbf{A}$ and $\bar{\lambda}$ represents its complex-conjugate, then (Horn & Johnson, 2012, Lemma 3.1.18) ensures that the Jordan structure of $\mathbf{A}$ corresponding to λ is the same as the Jordan structure of $\mathbf{A}$ corresponding to $\bar{\lambda}$. Next, let $\lambda = a + ib$ where $a, b \in \mathbb{R}$, and $i = \sqrt{-1}$. Let $\mathbf{D}(a, b)$ be defined as $\mathbf{D}(a, b) \triangleq \begin{bmatrix} a & b \\ -b & a \end{bmatrix}$. Then, the matrix $diag(\mathbf{V}(\lambda), \mathbf{V}(\bar{\lambda}))$ is similar to a real block upper triangular matrix $\mathbf{W}(\lambda) \in \mathbb{R}^{2a_{\mathbf{A}}(\lambda) \times 2a_{\mathbf{A}}(\lambda)}$ which has $a_{\mathbf{A}}(\lambda)$ 2-by-2 blocks $\mathbf{D}(a, b)$ on the main block diagonal and $(a_{\mathbf{A}}(\lambda) - 1)$ blocks $\mathbf{I}_2$ on the block superdiagonal. Henceforth, for a non-real eigenvalue $\lambda \in sp(\mathbf{A})$, $\mathbf{W}(\lambda)$ will have the meaning discussed above. Let $sp(\mathbf{A}) = \{\{\lambda_1, \bar{\lambda}_1\}, \ldots, \{\lambda_p, \bar{\lambda}_p\}, \lambda_{p+1}, \ldots, \lambda_\gamma\}$ with the first p pairs representing the non-real eigenvalues, and λ_{p+1} to λ_γ representing the real eigenvalues of $\mathbf{A}$. Then, the *real Jordan canonical form theorem* (Horn & Johnson, 2012, Theorem 3.4.1.5) can be stated as follows.

Theorem 3. *There exists a real similarity transformation matrix $\mathbf{T}$ that transforms the state transition matrix $\mathbf{A}$ in (1) to a real block diagonal matrix $\mathbf{M}$ given by $\mathbf{M} = diag(\mathbf{W}(\lambda_1), \ldots, \mathbf{W}(\lambda_p), \mathbf{V}(\lambda_{p+1}), \ldots, \mathbf{V}(\lambda_\gamma))$.*

With $\mathbf{T}$ as in the above theorem, and $\mathbf{z}[k] = \mathbf{T}^{-1}\mathbf{x}[k]$, the dynamics (1) are transformed into the form

$$\begin{aligned} \mathbf{z}[k+1] &= \mathbf{M}\mathbf{z}[k] \\ \mathbf{y}_i[k] &= \bar{\mathbf{C}}_i \mathbf{z}[k], \quad \forall i \in \{1, \ldots, N\} \end{aligned} \tag{3}$$

where $\mathbf{M} = \mathbf{T}^{-1}\mathbf{A}\mathbf{T}$ and $\bar{\mathbf{C}}_i = \mathbf{C}_i\mathbf{T}$.

For a non-real eigenvalue pair $\{\lambda_j, \bar{\lambda}_j\} \in sp(\mathbf{A})$, let $\mathbf{z}^{(j)}[k] \in \mathbb{R}^{2a_{\mathbf{A}}(\lambda_j)}$ represent the portion of the state $\mathbf{z}[k]$ associated with the matrix $\mathbf{W}(\lambda_j)$. Similarly, for a real eigenvalue $\lambda_j \in sp(\mathbf{A})$, $\mathbf{z}^{(j)}[k] \in \mathbb{R}^{a_{\mathbf{A}}(\lambda_j)}$ is the portion of the state $\mathbf{z}[k]$ associated with

[2] Similar bounds for static power system models subject to attacks were obtained in Kosut, Jia, Thomas, and Tong (2011).

[3] Throughout the paper, for the sake of conciseness, we use the terminology "node i can detect eigenvalue λ_j" to imply that $rank \begin{bmatrix} \mathbf{A}-\lambda_j \mathbf{I}_n \\ \mathbf{C}_i \end{bmatrix} = n$. Each stable eigenvalue is considered detectable w.r.t. the measurements of every node.

the matrix $\mathbf{V}(\lambda_j)$. For each node i, we denote the detectable and undetectable eigenvalues by the sets $\mathcal{O}_i$ and $\overline{\mathcal{O}}_i$, respectively. Next, we introduce the notion of *source nodes*.

Definition 7 (*Source Nodes*)**.** For each $\lambda_j \in \Lambda_U(\mathbf{A})$, the set of nodes that can detect λ_j is denoted by $\mathcal{S}_j$, and called the set of *source nodes* for λ_j.

We now proceed to develop an estimation scheme that enables each regular node to estimate $\mathbf{z}[k]$ (from which they can obtain $\mathbf{x}[k] = \mathbf{T}\mathbf{z}[k]$). Accordingly, let $\hat{\mathbf{z}}_i^{(j)}[k]$ denote the estimate of $\mathbf{z}^{(j)}[k]$ (the portion of $\mathbf{z}[k]$ corresponding to the eigenvalue λ_j[4]) maintained by node $i \in \mathcal{R}$. For each $\lambda_j \in \Lambda_U(\mathbf{A})$, our estimation scheme relies on separate strategies for nodes in $\mathcal{S}_j$ and $\mathcal{V} \setminus \mathcal{S}_j$. In particular, each node in $\mathcal{S}_j$ employs a Luenberger observer for estimating $\mathbf{z}^{(j)}[k]$. The nodes in $\mathcal{V} \setminus \mathcal{S}_j$, on the other hand, cannot detect the eigenvalue λ_j, and thus rely on a resilient consensus algorithm to estimate $\mathbf{z}^{(j)}[k]$. In what follows, we discuss these ideas in detail.

The first step in the estimation process involves the above common coordinate transformation given by $\mathbf{z}[k] = \mathbf{T}^{-1}\mathbf{x}[k]$, to be performed by each regular node of the graph. As this only requires knowledge of the system matrix $\mathbf{A}$ (which is assumed to be known by all the nodes), all of the nodes can do this in a distributed manner (e.g., by using an agreed-upon convention for ordering the eigenvalues and corresponding eigenvectors). Building on the general theme in Mitra and Sundaram (2018b), we first present a method that allows a regular node $i \in \mathcal{R}$ to estimate the locally detectable portion of the state $\mathbf{z}[k]$ *without* communicating with neighbors. To this end, consider the following result.

Lemma 1. *Let $\lambda_j \in \mathcal{O}_i$ be a non-real eigenvalue. Let $\bar{\mathbf{C}}_i^{(j)}$ denote the columns of $\bar{\mathbf{C}}_i$ corresponding to $\mathbf{W}(\lambda_j)$ in (3). Then, the pair $(\mathbf{W}(\lambda_j), \bar{\mathbf{C}}_i^{(j)})$ is detectable.*

The proof of the above lemma follows readily from the PBH test and is hence omitted here. Let $\mathcal{O}_i = \{\{\lambda_{n_1}, \bar{\lambda}_{n_1}\}, \dots, \{\lambda_{n_{p_i}}, \bar{\lambda}_{n_{p_i}}\}, \lambda_{n_{p_i+1}}, \dots, \lambda_{n_{\gamma_i}}\}$ be the eigenvalues of $\mathbf{A}$ that are detectable w.r.t. the measurements of node i, where the first p_i pairs are non-real eigenvalues, and $\lambda_{n_{p_i+1}}$ to $\lambda_{n_{\gamma_i}}$ are real eigenvalues. Let $\mathbf{M}_{\mathcal{O}_i} = diag(\mathbf{W}(\lambda_{n_1}), \dots, \mathbf{W}(\lambda_{n_{p_i}}), \mathbf{V}(\lambda_{n_{p_i+1}}), \dots, \mathbf{V}(\lambda_{n_{\gamma_i}}))$. Let $\mathbf{C}_{\mathcal{O}_i}$ represent the columns of $\bar{\mathbf{C}}_i$ corresponding to the matrix $\mathbf{M}_{\mathcal{O}_i}$ and $\mathbf{z}_{\mathcal{O}_i}[k]$ denote the portion of the state $\mathbf{z}[k]$ corresponding to the detectable eigenvalues of node i, i.e., corresponding to $\mathcal{O}_i$. Based on Lemma 1, it is easy to see that the pair $(\mathbf{M}_{\mathcal{O}_i}, \mathbf{C}_{\mathcal{O}_i})$ is detectable. Thus, a standard Luenberger observer can be locally constructed by node i for estimating $\mathbf{z}_{\mathcal{O}_i}[k]$. The details of such a construction are straightforward, and are similar to those in Section VI-A of Mitra and Sundaram (2018b). We thus skip minor details and state the following result which will be useful later on.

Lemma 2. *For each regular node $i \in \mathcal{R}$ and each $\lambda_j \in \mathcal{O}_i$, a Luenberger observer can be locally constructed by node i such that $\lim_{k\to\infty} \|\hat{\mathbf{z}}_i^{(j)}[k] - \mathbf{z}^{(j)}[k]\| = 0$.*

Based on the previous result, we see that a regular node i can estimate certain portions of the state space without having to exchange information with neighbors. The challenge, however, lies in estimating the locally undetectable portion of the state in the presence of adversaries. The following section presents a resilient consensus based strategy to address this issue.

[4] Throughout the rest of the paper, if $\lambda_j \in sp(\mathbf{A})$ is a non-real eigenvalue, then the terminology "$\mathbf{z}^{(j)}[k]$ corresponds to the eigenvalue λ_j" should be interpreted as "$\mathbf{z}^{(j)}[k]$ corresponds to the eigenvalue pair $\{\lambda_j, \bar{\lambda}_j\}$".

4.2. Local-filtering based resilient estimation

For any $\lambda_j \in sp(\mathbf{A})$, let $z^{(jm)}[k]$ denote the m-th component of the vector $\mathbf{z}^{(j)}[k]$, and let $\hat{z}_i^{(jm)}[k]$ denote the estimate of that component maintained by node $i \in \mathcal{V}$. Consider an unstable eigenvalue $\lambda_j \in \overline{\mathcal{O}}_i$. For such an eigenvalue, node i has to rely on the information received from its neighbors, some of which might be adversarial, in order to estimate $\mathbf{z}^{(j)}[k]$. To this end, we propose a resilient consensus algorithm that requires each regular node $i \in \mathcal{V} \setminus \mathcal{S}_j$ to update its estimate of $\mathbf{z}^{(j)}[k]$ using the following two stage filtering strategy:

(1) At each time-step k, each regular node i collects the state estimates of $\mathbf{z}^{(j)}[k]$ received from *only* those neighbors that belong to a certain subset $\mathcal{N}_i^{(j)} \subseteq \mathcal{N}_i$ (to be defined later). For every component m of $\mathbf{z}^{(j)}[k]$, the estimates of $z^{(jm)}[k]$ received from nodes in $\mathcal{N}_i^{(j)}$ are sorted from largest to smallest.

(2) For each component m of $\mathbf{z}^{(j)}[k]$, node i removes the largest and smallest f estimates (i.e., removes $2f$ estimates in all) of $z^{(jm)}[k]$ received from nodes in $\mathcal{N}_i^{(j)}$, and computes the quantity:

$$\bar{z}_i^{(jm)}[k] = \sum_{l \in \mathcal{M}_i^{(jm)}[k]} w_{il}^{(jm)}[k] \hat{z}_l^{(jm)}[k], \tag{4}$$

where $\mathcal{M}_i^{(jm)}[k] \subset \mathcal{N}_i^{(j)} (\subseteq \mathcal{N}_i)$ is the set of nodes from which node i chooses to accept estimates of $z^{(jm)}[k]$ at time-step k, after removing the f largest and f smallest estimates of $z^{(jm)}[k]$ received from $\mathcal{N}_i^{(j)}$. Node i assigns the weight $w_{il}^{(jm)}[k]$ to the l-th node at the k-th time-step for estimating the m-th component of $\mathbf{z}^{(j)}[k]$. The weights are nonnegative and chosen to satisfy $\sum_{l \in \mathcal{M}_i^{(jm)}[k]} w_{il}^{(jm)}[k] = 1, \forall \lambda_j \in \overline{\mathcal{O}}_i$, and for each component m of $\mathbf{z}^{(j)}[k]$. With the quantities $\bar{z}_i^{(jm)}[k]$ in hand, node i updates $\hat{\mathbf{z}}_i^{(j)}[k]$ as follows:

$$\hat{\mathbf{z}}_i^{(j)}[k+1] = \begin{cases} \mathbf{V}(\lambda_j)\bar{\mathbf{z}}_i^{(j)}[k], & \text{if } \lambda_j \in \overline{\mathcal{O}}_i \text{ is real} \\ \mathbf{W}(\lambda_j)\bar{\mathbf{z}}_i^{(j)}[k], & \text{if } \lambda_j \in \overline{\mathcal{O}}_i \text{ is not real,} \end{cases} \tag{5}$$

where $\bar{\mathbf{z}}_i^{(j)}[k] = \left[\bar{z}_i^{(j1)}[k], \dots, \bar{z}_i^{(j\sigma_j)}[k]\right]^T$, $\sigma_j = a_{\mathbf{A}}(\lambda_j)$ if $\lambda_j \in \overline{\mathcal{O}}_i$ is real, and $\sigma_j = 2a_{\mathbf{A}}(\lambda_j)$ if $\lambda_j \in \overline{\mathcal{O}}_i$ is not real.

We refer to the above algorithm as the Local-Filtering based Resilient Estimation (LFRE) algorithm. For implementing this algorithm, a regular node i needs to construct the set $\mathcal{N}_i^{(j)}$, $\forall \lambda_j \in \overline{\mathcal{O}}_i$, based on the relative positions of its neighbors (with respect to its own position) in $\mathcal{G}$. We will provide the exact definition of $\mathcal{N}_i^{(j)}$, and a distributed algorithm for constructing such a set in the following sections where we analyze the convergence of the LFRE algorithm. We conclude this section by commenting on certain features of the LFRE algorithm.

Remark 4. The rationale behind performing a real Jordan canonical decomposition at every node (as opposed to a standard Jordan transformation) is to ensure that the state estimates featuring in Eqs. (4) and (5) are real at every time-step, thereby making the sorting operation performed in Step 1 of the algorithm meaningful. At any time-step, if a regular node i either receives a non-real estimate of $z^{(jm)}[k]$ from some node $l \in \mathcal{N}_i^{(j)}$, or does not receive an estimate at all, it would immediately identify node l as an adversarial node, and simply assign a 0 value to node l's estimate of $z^{(jm)}[k]$. Note that every regular node in $\mathcal{N}_i^{(j)}$ will always transmit a real estimate to node i at every time-step.

Remark 5. The strategy of disregarding the most extreme values in one's neighborhood, and using a convex combination of

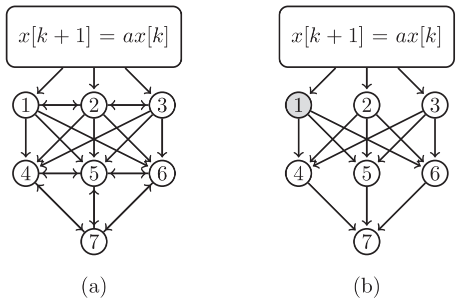


Fig. 2. A scalar unstable plant is monitored by a network of 7 nodes as depicted in the figure on the left. Nodes 1, 2 and 3 are the source nodes for this system. The figure on the right represents a subgraph of the original graph satisfying the properties of a MEDAG in Definition 8 for all 1-local sets (i.e., with $f = 1$). For example, when $\mathcal{A} = \{1\}$ (as shown in the right figure), every non-source node has at least $2f + 1 = 3$ neighbors. The levels that partition $\mathcal{R} = \mathcal{V} \setminus \mathcal{A}$ are level 0 with nodes 2 and 3, level 1 with nodes 4, 5 and 6, and level 2 with node 7. Each regular node has all its regular neighbors in levels that are numbered lower than its own.

the rest for performing linear scalar updates, has been used for designing resilient distributed algorithms for consensus (LeBlanc et al., 2013; Vaidya et al., 2012). In this paper, we show that such an idea is also applicable to resilient distributed state estimation, with certain substantial differences arising from the fact that the nodes are trying to track the state of an external dynamical system.

Remark 6. The consensus weights $w_{il}^{(jm)}$ appearing in Eq. (4) can be chosen arbitrarily to achieve an exponential rate of convergence, as long as the weights meet the rules specified by the LFRE algorithm. Since our primary focus is on resilience against worst-case adversarial behavior, the problem of optimizing such weights (or exploiting sensor memory) for achieving improved performance against noise is not considered in this paper. In a non-adversarial setting (i.e., when $f = 0$), the proposed LFRE algorithm will continue to guarantee exponential convergence in the absence of noise, and bounded mean square error in the presence of i.i.d. noise with bounded second moments (provided the topological conditions outlined in Section 7 are met). However, disregarding the estimates of certain neighbors in the absence of attacks may potentially degrade performance against noise; we do not delve deeper into this topic here.

It should be noted that the algorithmic development in this section can be considerably simplified if more structure is imposed on the system matrix $\mathbf{A}$ (for instance, the assumption made in Mitra & Sundaram, 2016 that $\mathbf{A}$ has only real, distinct eigenvalues).

5. Analysis of the resilient distributed estimation strategy

In this section, we provide our main result concerning the convergence of the LFRE algorithm. Let $\Omega_U(\mathbf{A}) \triangleq \{\lambda_j \in \Lambda_U(\mathbf{A}) | \mathcal{V} \setminus \mathcal{S}_j \text{ is non-empty}\}$. By this definition, all nodes are source nodes for each eigenvalue in $\Lambda_U(\mathbf{A}) \setminus \Omega_U(\mathbf{A})$, and are hence capable of recovering the corresponding portions of the state based on locally constructed Luenberger observers (as discussed in Section 4.1). Consequently, the LFRE algorithm specifically applies to only those eigenvalues that belong to $\Omega_U(\mathbf{A})$. Consider the following definition.

Definition 8 (*Mode Estimation Directed Acyclic Graph (MEDAG)*)**.** Consider an eigenvalue $\lambda_j \in \Omega_U(\mathbf{A})$. Suppose there exists a spanning subgraph $\mathcal{G}_j = (\mathcal{V}, \mathcal{E}_j)$ of $\mathcal{G}$ with the following properties for all f-local sets $\mathcal{A}$ and $\mathcal{R} = \mathcal{V} \setminus \mathcal{A}$.

(i) If $i \in \{\mathcal{V} \setminus \mathcal{S}_j\} \cap \mathcal{R}$, then $|\mathcal{N}_i^{(j)}| \geq 2f + 1$, where $\mathcal{N}_i^{(j)} = \{l \in \mathcal{V} | (l, i) \in \mathcal{E}_j\}$ represents the neighborhood of node i in $\mathcal{G}_j$.

(ii) There exists a partition of $\mathcal{R}$ into the sets $\{\mathcal{L}_0^{(j)}, \ldots, \mathcal{L}_{T_j}^{(j)}\}$, where $T_j \in \mathbb{N}_+$, $\mathcal{L}_0^{(j)} = \mathcal{S}_j \cap \mathcal{R}$, and if $i \in \mathcal{L}_q^{(j)}$ (where $1 \leq q \leq T_j$), then $\mathcal{N}_i^{(j)} \cap \mathcal{R} \subseteq \bigcup_{r=0}^{q-1} \mathcal{L}_r^{(j)}$. Furthermore, $\mathcal{N}_i^{(j)} = \emptyset, \forall i \in \mathcal{L}_0^{(j)}$.

Then, we call $\mathcal{G}_j$ a *Mode Estimation Directed Acyclic Graph (MEDAG)* for $\lambda_j \in \Omega_U(\mathbf{A})$.

An example of a MEDAG is shown in Fig. 2. The "for all $\mathcal{A}$" in the definition accounts for the fact that the set of adversarial nodes during the process of state estimation is unknown, and hence can be any f-local set of $\mathcal{V}$. Note that T_j and the levels $\mathcal{L}_0^{(j)}$ to $\mathcal{L}_{T_j}^{(j)}$ can vary across different f-local sets. For a given f-local set $\mathcal{A}$, we say a regular node $i \in \mathcal{L}_m^{(j)}$ "belongs to level m", where the levels are indicative of the distances of the regular nodes from the source set $\mathcal{S}_j$. The first property indicates that every regular node $i \in \mathcal{V} \setminus \mathcal{S}_j$ has at least $(2f + 1)$ neighbors in the subgraph $\mathcal{G}_j$, while the second property indicates that all its regular neighbors in such a subgraph belong to levels strictly preceding its own level. In essence, the edges of the MEDAG $\mathcal{G}_j$ represent a medium for transmitting information securely from the source nodes $\mathcal{S}_j$ to the non-source nodes, by preventing the adversaries from forming a bottleneck between such nodes. Intuitively, this requires redundant nodes and edges, and such a requirement is met by the first property of the MEDAG. In particular, as regards measurement redundancy, it follows from the definition that for each $\lambda_j \in \Omega_U(\mathbf{A})$, a MEDAG $\mathcal{G}_j$ contains at least $(2f + 1)$ source nodes that can detect λ_j.[5] The LFRE algorithm described in the previous section relies on a special *uni-directional* information flow pattern that requires a node i to listen to *only* its neighbors in $\mathcal{N}_i^{(j)}$ for estimating $\mathbf{z}^{(j)}[k]$. The second property of a MEDAG then indicates that nodes in level m only use the estimates of regular nodes in levels 0 to $m - 1$ for recovering $\mathbf{z}^{(j)}[k]$. The implications of the above properties will become apparent in the proof of the following result which provides a sufficient condition for solving Problem 1 based on our approach.

Theorem 4. *Suppose that $\mathcal{G}$ contains a MEDAG $\mathcal{G}_j$ for each $\lambda_j \in \Omega_U(\mathbf{A})$. Then, based on the LFRE dynamics described by Eqs.* (4) *and* (5)*, each regular node $i \in \mathcal{R}$ can asymptotically estimate the state of the plant, despite the actions of any f-locally bounded set of Byzantine adversaries.*

The proof of the above theorem is given in Appendix B. Notice that Theorem 4 hinges on the existence of a MEDAG $\mathcal{G}_j$, for each $\lambda_j \in \Omega_U(\mathbf{A})$; in the following section we describe an approach for checking whether a given graph $\mathcal{G}$ contains such MEDAGs.

6. Checking the existence of a MEDAG

From the foregoing discussion, it is apparent that the MEDAGs described in Definition 8 play a key role in solving Problem 1 based on our proposed technique. In particular, recall that for each $\lambda_j \in \Omega_U(\mathbf{A})$, the LFRE algorithm described in Section 4.2 requires a regular node $i \in \mathcal{V} \setminus \mathcal{S}_j$ to accept estimates from only its neighbor set $\mathcal{N}_i^{(j)}$ in the MEDAG $\mathcal{G}_j$ for estimating $\mathbf{z}^{(j)}[k]$. With these points in mind, our immediate goal in this section will be to develop a distributed algorithm, namely Algorithm 1, that constructs a MEDAG $\mathcal{G}_j$ for each $\lambda_j \in \Omega_U(\mathbf{A})$, and in the process enables each regular node i to determine the set $\mathcal{N}_i^{(j)}$ for

[5] Recall from the discussion immediately following Proposition 1 that such a condition is in fact necessary for systems with distinct eigenvalues.

Algorithm 1 MEDAG Construction Algorithm

For each eigenvalue $\lambda_j \in \Omega_U(\mathbf{A})$ **do**:

Initialization: Initialize $c_i(j) = 0$, $\mathcal{N}_i^{(j)} = \emptyset$, $\forall i \in \mathcal{V}$. Each node determines whether it belongs to $\mathcal{S}_j$.

Actions of the source nodes: Each node in $\mathcal{S}_j$ updates its counter value $c_i(j) = 1$, and transmits the message "1" to its out-neighbors. Following this step, it does not listen to any other node, i.e., $\mathcal{N}_i^{(j)} = \emptyset$ and $c_i(j) = 1, \forall i \in \mathcal{S}_j$ for the remainder of the algorithm.

Actions of the non-source nodes: Each node $i \in \mathcal{V} \setminus \mathcal{S}_j$ does the following:

- If $c_i(j) = 0$ *and* node i has received "1" from at least $(2f+1)$ distinct neighbors (not necessarily all in the same round), it updates $c_i(j)$ to 1, appends the labels of the neighbors from which it received "1" to $\mathcal{N}_i^{(j)}$, and transmits "1" to its out-neighbors.
- If $c_i(j) = 1$, it discards all messages received from its neighbors, i.e., it does not update $c_i(j)$ or $\mathcal{N}_i^{(j)}$.

Return : A set of sets $\{\mathcal{N}_i^{(j)}\}$, $\lambda_j \in \Omega_U(\mathbf{A})$, $i \in \mathcal{V}$.

each $\lambda_j \in \overline{\mathcal{O}}_i$. The construction of these MEDAGs constitutes the initialization phase of our design, which can then be followed up by the LFRE algorithm described earlier. We briefly describe the implementation of Algorithm 1 as follows.

Algorithm 1 requires each node i to maintain a counter $c_i(j)$ and a list of indices $\mathcal{N}_i^{(j)}$ for each $\lambda_j \in \Omega_U(\mathbf{A})$. The nodes in $\mathcal{N}_i^{(j)} \subseteq \mathcal{N}_i$ will be the parents of node i in the DAG constructed for the estimation of $\mathbf{z}^{(j)}[k]$. Algorithm 1 is initialized with $c_i(j) = 0$ and $\mathcal{N}_i^{(j)} = \emptyset$, for each $i \in \mathcal{V}$. Subsequently, the algorithm proceeds in rounds where in the first round each node in $\mathcal{S}_j$ broadcasts the message "1" to its out-neighbors, sets $c_i(j) = 1$, maintains $\mathcal{N}_i^{(j)} = \emptyset$ for all future rounds, and goes to sleep. Each node $i \in \mathcal{V} \setminus \mathcal{S}_j$ waits until it has received "1" from at least $(2f+1)$ distinct neighbors, at which point it sets $c_i(j) = 1$, appends the labels of each of the neighbors from which it received "1" to $\mathcal{N}_i^{(j)}$, broadcasts the message "1" to its out-neighbors, and goes to sleep. Let $\mathcal{R}' \subseteq \mathcal{V}$ denote the set of nodes that behave regularly during the execution of Algorithm 1. We say that the MEDAG construction algorithm "*terminates for* λ_j" if there exists $T_j \in \mathbb{N}_+$ such that $c_i(j) = 1\ \forall i \in \mathcal{R}'$, for all rounds following round T_j. The objective of the algorithm is to return a set of sets $\{\mathcal{N}_i^{(j)}\}$, where $\lambda_j \in \Omega_U(\mathbf{A})$, and $i \in \mathcal{V}$.

We emphasize that in addition to misbehavior during the state estimation phase (run-time), an adversarial node is allowed to misbehave during the implementation of Algorithm 1 (design-time) as well. For example, it can transmit the message out of turn, i.e., before receiving "1" from at least $(2f+1)$ neighbors. It can also choose not to transmit the message at all. Note however that we must have $\mathcal{V} \setminus \mathcal{R}' \subseteq \mathcal{A}$, i.e., the f-local set of adversaries during the estimation phase must contain the set of adversaries during the design phase. In the next section, we shall detail graph conditions that guarantee the termination of the MEDAG construction algorithm under arbitrary adversarial behavior. For the following discussion, we characterize the properties of the output of Algorithm 1 if it terminates. To this end, consider the spanning subgraph $\mathcal{G}_j = (\mathcal{V}, \mathcal{E}_j)$ induced by the sets $\{\mathcal{N}_i^{(j)}\}$ returned by Algorithm 1. Keeping in mind that $\mathcal{R}' \supseteq \mathcal{R}$ represents the set of nodes that behave regularly during the execution of Algorithm 1, we have the following result (the proof is omitted due to space constraints, and can be found in Mitra & Sundaram, 2016).

Theorem 5. *Suppose the MEDAG construction algorithm terminates for $\lambda_j \in \Omega_U(\mathbf{A})$. Then, there exists a subgraph $\mathcal{G}_j$ satisfying the properties of a MEDAG for all f-local sets $\mathcal{A}$ that contain $\mathcal{V} \setminus \mathcal{R}'$ as a subset.*

Remark 7. Based on the above theorem, we make the following observations. If Algorithm 1 terminates for each $\lambda_j \in \Omega_U(\mathbf{A})$, and $\mathcal{V} \setminus \mathcal{R}' = \emptyset$, then the $\mathcal{G}_j$ subgraphs satisfy all the properties of a MEDAG and we can directly invoke Theorem 4. If Algorithm 1 terminates for each $\lambda_j \in \Omega_U(\mathbf{A})$ and $\mathcal{V} \setminus \mathcal{R}' \neq \emptyset$, (i.e., there is some adversarial activity during the MEDAG construction phase), then we do not need to provide any guarantees of state estimation for the set of misbehaving nodes $\mathcal{V} \setminus \mathcal{R}'$, since $\mathcal{V} \setminus \mathcal{R}' \subseteq \mathcal{A}$. In this case too, the subgraphs returned by Algorithm 1 have enough redundancy to ensure that Problem 1 can be solved based on our proposed approach; this fact can be established using arguments identical to those used for proving Theorem 4. In what follows, we summarize our overall approach.

6.1. Summary of the resilient distributed state estimation scheme

(1) Each regular node $i \in \mathcal{R}$ performs the coordinate transformation $\mathbf{z}[k] = \mathbf{T}^{-1}\mathbf{x}[k]$ described in Section 4.1; accordingly, it identifies its detectable and undetectable eigenvalues ($\mathcal{O}_i$ and $\overline{\mathcal{O}}_i$ respectively).
(2) The MEDAG construction algorithm described by Algorithm 1 is implemented for each $\lambda_j \in \Omega_U(\mathbf{A})$; graph conditions for termination of this algorithm are provided in the next section. At the end of this algorithm, each regular node i knows the subset $\mathcal{N}_i^{(j)}$ of neighbors it should use in the LFRE algorithm.
(3) Each regular node i employs a locally constructed Luenberger observer (refer to Lemma 2 and the discussion preceding it) for estimating $\mathbf{z}_{\mathcal{O}_i}[k]$, namely the portion of the state $\mathbf{z}[k]$ corresponding to its detectable eigenvalues.
(4) Each regular node i employs the LFRE algorithm governed by Eqs. (4) and (5) for estimating $\mathbf{z}_{\overline{\mathcal{O}}_i}[k]$, namely the portion of the state $\mathbf{z}[k]$ corresponding to its undetectable eigenvalues.

Remark 8. Whereas steps 1 and 2 correspond to the initial design phase of our scheme, steps 3 and 4 constitute the estimation phase. A key benefit of the proposed method is that if certain graph-theoretic conditions (to be discussed in the following section) are met, then our overall scheme provably admits a **fully distributed implementation** even under worst-case adversarial behavior.

7. Feasible graph topologies

In this section, we characterize feasible graph topologies that guarantee the termination of the MEDAG construction algorithm (Algorithm 1) described in the previous section. In other words, based on Remark 7, feasible graph topologies guarantee that Problem 1 can be solved based on our proposed approach (summarized in Section 6.1). We first recall the following definition from LeBlanc et al. (2013).

Definition 9 (*r-reachable set*)**.** For a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a set $\mathcal{S} \subset \mathcal{V}$, and an integer $r \in \mathbb{N}_+$, $\mathcal{S}$ is an *r-reachable set* if there exists an $i \in \mathcal{S}$ such that $|\mathcal{N}_i \setminus \mathcal{S}| \geq r$.

Thus, if a set $\mathcal{S}$ is r-reachable, then it contains a node that has at least r neighbors outside $\mathcal{S}$. We will also use the notion of a strongly r-robust graph as follows.

Definition 10 (*Strongly r-robust graph w.r.t. $\mathcal{S}_j$*)**.** For $r \in \mathbb{N}_+$ and $\lambda_j \in \Omega_U(\mathbf{A})$, a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is *strongly r-robust w.r.t. to the set of source nodes $\mathcal{S}_j$* if for any non-empty subset $\mathcal{C} \subseteq \mathcal{V} \setminus \mathcal{S}_j$, $\mathcal{C}$ is r-reachable.

For an illustration of the above definitions, the reader is referred back to Fig. 2. Fig. 2(a) is an example of a network that is strongly 3-robust w.r.t. the set of source nodes, namely nodes $\{1, 2, 3\}$. Specifically, all subsets of $\{4, 5, 6, 7\}$ are 3-reachable (i.e., each such subset has a node that has at least 3 neighbors outside that subset).

Lemma 3. *The MEDAG construction algorithm terminates for $\lambda_j \in \Omega_U(\mathbf{A})$ if $\mathcal{G}$ is strongly $(3f + 1)$-robust w.r.t. $\mathcal{S}_j$.*

Proof. We prove by contradiction. Consider any $\lambda_j \in \Omega_U(\mathbf{A})$ and let $\mathcal{G}$ be strongly $(3f + 1)$-robust w.r.t. the set of source nodes $\mathcal{S}_j$. Suppose that the MEDAG construction algorithm for λ_j does not terminate. Since the possibility of the counter $c_i(j)$ oscillating between 0 and 1 (where $i \in \mathcal{R}$) is ruled out based on our MEDAG construction algorithm, there must then exist a non-empty set $\mathcal{C} \subseteq \mathcal{V} \setminus \mathcal{S}_j$ of regular nodes that never update their counter $c_i(j)$ from 0 to 1, where $i \in \mathcal{C}$. As $\mathcal{G}$ is strongly $(3f + 1)$-robust w.r.t. $\mathcal{S}_j$, it follows that $\mathcal{C}$ is $(3f + 1)$-reachable, i.e., there exists a node $i \in \mathcal{C}$ which has at least $(3f + 1)$ neighbors outside $\mathcal{C}$. Under the f-local adversarial model, at least $(2f + 1)$ of these neighbors are regular nodes with $c_i(j) = 1$. Thus, at least $(2f + 1)$ regular nodes must have transmitted the message "1" to node i. Thus, based on the rules of Algorithm 1, node i must have updated $c_i(j)$ from 0 to 1 at some point of time, leading to a contradiction. □

Whereas the $(2f + 1)$ term appears in various contexts when dealing with security problems on networks (see for instance Koo, 2004; LeBlanc et al., 2013; Pelc & Peleg, 2005; Vaidya et al., 2012), the $(3f + 1)$ term featuring in our analysis accounts for misbehavior that involves transmission of no messages by the adversarial nodes during execution of the MEDAG construction algorithm described in Section 6. We now present the main result of this paper which ties together the previous results presented in this paper and, in turn, provides a connection between feasible graph topologies and the solution to Problem 1 based on our proposed approach.

Theorem 6. *Consider an LTI system (1) and a measurement model (2). Let the communication graph $\mathcal{G}$ be strongly $(3f + 1)$-robust w.r.t. $\mathcal{S}_j, \forall \lambda_j \in \Omega_U(\mathbf{A})$. Then, the proposed algorithm summarized in Section 6.1 provides a solution to Problem 1.*

Proof. From Lemma 3, it follows that if $\mathcal{G}$ is strongly $(3f + 1)$-robust w.r.t. $\mathcal{S}_j$ for every $\lambda_j \in \Omega_U(\mathbf{A})$, then the MEDAG construction algorithm terminates for each such eigenvalue. Combining Theorem 5, Remark 7 and Theorem 4 then leads to the desired result. □

If the adversarial attacks are restricted to the estimation phase only, i.e., if there are no attacks during the initial MEDAG construction phase, then the following result provides a tight graph condition for our algorithm.

Theorem 7. *Consider an LTI system (1) and a measurement model (2). Suppose adversarial behavior is restricted to the estimation phase (steps 3 and 4) of the proposed algorithm summarized in Section 6.1. Then, this algorithm solves Problem 1 if and only if $\mathcal{G}$ is strongly $(2f + 1)$-robust w.r.t. $\mathcal{S}_j, \forall \lambda_j \in \Omega_U(\mathbf{A})$.*

For a proof of the above result (omitted here due to space constraints), see Mitra and Sundaram (2018a). Theorem 7 alludes to the fact that $\mathcal{G}$ contains a MEDAG $\mathcal{G}_j$ for each $\lambda_j \in \Omega_U(\mathbf{A})$ *if and only if* $\mathcal{G}$ is strongly $(2f + 1)$-robust w.r.t. $\mathcal{S}_j, \forall \lambda_j \in \Omega_U(\mathbf{A})$. Note that although Theorem 7 provides a graph condition that is necessary and sufficient for the algorithm developed in this paper, such a condition may not be necessary for solving Problem 1 in general.

Theorems 6 and 7 reveal that '*strong r-robustness w.r.t. $\mathcal{S}_j, \forall \lambda_j \in \Omega_U(\mathbf{A})$*' is the key topological property required for guaranteeing success of our proposed algorithm. Accordingly, given a system model (1) and measurement model (2), a network that is strongly r-robust w.r.t. $\mathcal{S}_j, \forall \lambda_j \in \Omega_U(\mathbf{A})$, will be called an '*$r$-feasible network*' for simplicity. Properties of an r-feasible network are summarized in Mitra and Sundaram (2018a, Proposition 4).

Applicability of the proposed approach: Building on the insights developed in this section, we make a case for the applicability of the approach developed in this paper by addressing the following question: How efficiently can one verify whether a given system and network is r-feasible? To answer the above question, we will exploit a connection between the 'strong r-robustness property w.r.t. a certain set of nodes' and the dynamic process of 'bootstrap percolation' on networks (Janson, Łuczak, Turova, & Vallier, 2012). Given a graph $\mathcal{G}$ and a threshold $r \geq 2$, bootstrap percolation can be viewed as a process of spread of *activation* where one starts off with a set $\mathcal{I} \subseteq \mathcal{V}$ of initially active nodes. Subsequently, the process evolves over the network based on the rule that an inactive node becomes active if and only if it has at least r active neighbors, with active nodes remaining active forever. The process terminates when no more nodes become active; an initial set $\mathcal{I}$ is said to *percolate* if upon termination the final active set equals the entire node set $\mathcal{V}$. Consider the following simple, yet key observation.

Lemma 4. *Given a graph $\mathcal{G}$ and a threshold $r \geq 2$, an initial set $\mathcal{I}$ percolates via the process of bootstrap percolation if and only if $\mathcal{G}$ is strongly r-robust w.r.t. $\mathcal{I}$.*

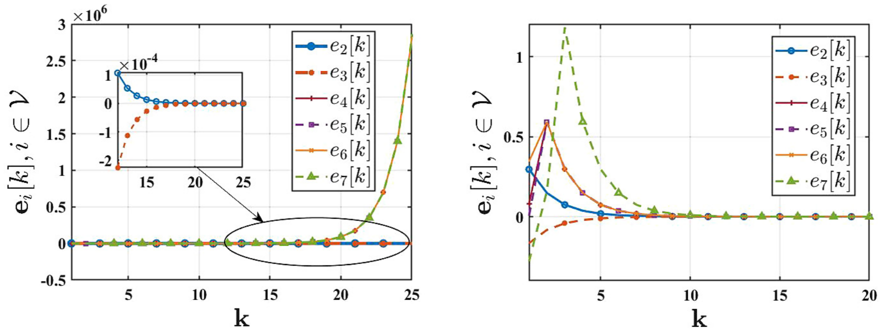
The proof of the above result follows similar arguments as Lemma 3, and is hence omitted. Leveraging Lemma 4, we obtain the following result.

Proposition 2. *Given a system matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ (1), a measurement model (2), a communication graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = N$, the source set $\mathcal{S}_j$ for each $\lambda_j \in sp(\mathbf{A})$, and an integer $r \geq 2$, one can verify whether the network is r-feasible in $O(nN|\mathcal{E}|)$ time.*

Proof. Notice that $|\Omega_U(\mathbf{A})| \leq n$, i.e., there are at most n source sets $\mathcal{S}_j$ for which we need to verify the strong r-robustness property in Definition 10. Based on Lemma 4, for each $\mathcal{S}_j$ corresponding to some $\lambda_j \in \Omega_U(\mathbf{A})$, verifying whether $\mathcal{G}$ is strongly r-robust w.r.t. $\mathcal{S}_j$ is equivalent to verifying whether $\mathcal{S}_j$ percolates via the process of bootstrap percolation with threshold r. Thus, we analyze the complexity of simulating a bootstrap percolation process on a given network.[6] First, notice that it takes at most N iterations/rounds for a bootstrap percolation process to terminate on a network of N nodes. In each round, every inactive node checks whether it has at least r active neighbors; the entire process of checking is thus completed in $O(\sum_{i=1}^{N} d_i) = O(|\mathcal{E}|)$ time, where d_i represents the in-degree of node i. Thus, for a given initial set, it takes $O(N|\mathcal{E}|)$ time to simulate the bootstrap percolation process. The result then follows readily. □

Remark 9. Based on the above result, one can check whether the approach developed in this paper is applicable for a given system

6 Algorithm 1 essentially simulates the evolution of a bootstrap percolation process with threshold $r = (2f + 1)$, provided there is no adversarial activity during the distributed implementation of such an algorithm.



(a) (b)

Fig. 3. Consider the system and network in Fig. 2, and let $e_i[k] = \hat{x}_i[k] - x[k]$ denote the estimation error of node i at time-step k. Fig. (a) depicts how a single adversary, namely node 1, can cause the estimation errors of all the non-source regular nodes (namely, nodes 4–7) to diverge when a non-resilient distributed observer is employed. Fig. (b) shows hows the proposed LFRE algorithm counteracts the effect of the adversary, and allows each regular node to track the state.

and network in polynomial time.[7] Interestingly, leveraging the equivalence described in Lemma 4, it is possible to show that the strong r-robustness property described in Definition 10 is exhibited by various large-scale complex network models such as the Barabási–Albert (BA) preferential attachment model, the Erdős–Rényi random graph model, and the 2-dimensional random geometric graph model. A detailed discussion on this topic can be found in Mitra and Sundaram (2018a).

8. Simulations

Consider the system and network given by Fig. 2. The state evolves as $x[k+1] = ax[k]$, with $a = 2$. Nodes 1, 2 and 3 are the source nodes and directly estimate the state, i.e., $y_i[k] = x[k], \forall i \in \{1, 2, 3\}$. The rest of the nodes have zero measurements. Node 1 is the only adversarial node in the network, and it simply transmits a constant signal of magnitude $\epsilon = 0.001$ to each of its neighbors at every time-step. Each regular source node updates its state estimate based on a standard Luenberger observer as follows:

$$\hat{x}_i[k+1] = a\hat{x}_i[k] + l_i(y_i[k] - \hat{x}_i[k]), \; i \in \{2, 3\}, \tag{6}$$

with the observer gain l_i set to 1.5 (this gain is simply chosen to ensure stability). We first consider a scenario where each non-source node updates its estimate as follows: $\hat{x}_i[k+1] = a\sum_{l\in\mathcal{N}_i^{(j)}} w_{il}^{(j)}\hat{x}_l[k]$, where the weights form a convex combination, and $\mathcal{N}_i^{(j)}$ represents the neighbors of node i in the MEDAG shown in Fig. 2(b).[8] If node 1 were to update its estimate as per (6), then it can be easily verified analytically that all nodes would be able to track the state asymptotically (the distributed observer described above is based on the general design procedure outlined in Mitra & Sundaram, 2018b). However, as seen from Fig. 3(a), a single adversarial node (node 1 in this case) transmitting a small constant signal can cause the estimates of all the non-source nodes to diverge. This example demonstrates that although the underlying network is strongly 3-robust (i.e., has enough built-in redundancy to deal with a single adversarial node[9]), the non-resilient distributed observer employed above proves to be inadequate in the face of attacks. However, as seen from Fig. 3(b), the LFRE algorithm complements the robust network structure and succeeds in counteracting the adversarial attack. For all simulations, $x[0] = 0.5$, and $\hat{x}_i[0], i \in \mathcal{V}$ is a random number between 0 and 1.

9. Conclusions

We studied the problem of collaboratively estimating the state of an LTI system subject to worst-case adversarial behavior. For the attack models under consideration, we identified certain necessary conditions that need to be satisfied by any system and network for the problem posed in this paper to have a feasible solution. We then developed a local-filtering algorithm to enable each non-compromised node to recover the entire state. Finally, using a topological property called strong r-robustness, we characterized networks that guarantee success of our proposed strategy. Two notable features of our approach are as follows: (i) each step of our approach admits an attack-resilient and completely distributed implementation provided certain graph-theoretic conditions are met; and (ii) these graph-theoretic conditions can be checked in polynomial time as discussed in the previous section.

There are various interesting directions for future research, some of which were pointed out in Remark 3. Finding an algorithm-independent necessary and sufficient condition for the problem posed in this paper will likely be a challenging proposition. Whereas the focus of this paper has been on obtaining sufficient graph-theoretic conditions that account for worst-case adversarial behavior, it might be of interest to see if such conditions can be relaxed when confronted with less sophisticated adversarial attacks. Extensions of our framework to account for network-induced issues such as packet-drops, delays and asynchronicity also merit attention; see Mitra and Sundaram (2018c) for preliminary results on this topic.

Appendix A. Proof of Theorem 2

Proof. "(i)⟹(ii)" We prove by contraposition. Suppose statement (ii) is violated for some node $i \in \mathcal{V}$, i.e., there exists a set $\mathcal{D}_i$ such that its removal from $\mathcal{G}$ causes the pair $(\mathbf{A}, \mathbf{C}_{i\cup\mathcal{P}_i})$ to become undetectable (where $\mathcal{D}_i$ and $\mathcal{P}_i$ have the same meaning as in the statement of Theorem 2). It then follows that $\mathcal{F} = \mathcal{V} \setminus \{i \cup \mathcal{P}_i\}$ is a critical set. Suppose it is also a minimal critical set. We construct $\mathcal{G}'$ by adding directed edges from a virtual node s to each node in $\mathcal{F}$.[10] Observe that $\mathcal{H} = \mathcal{D}_i$ satisfies all the properties

7 This result is in stark contrast with analogous results existing in the resilient distributed consensus literature (LeBlanc et al., 2013; Vaidya et al., 2012), since checking the 'robustness' condition needed for solving such problems is coNP-complete.

8 For this scalar system, there is only one mode, i.e., $j = 1$.

9 For this example, we assume that node 1 misbehaves only during the estimation phase. Hence, Theorem 7 is applicable.

10 Throughout this proof, we drop the subscript on $\mathcal{G}'$, $\mathcal{F}$ and s, unlike the notation in Section 3. This is done since the subscript i is used to denote sets defined w.r.t. a node i in this proof.

of an f-total pair cut w.r.t. s. In particular, $\mathcal{Y} = \{i \cup \mathcal{P}_i\}$ and $\mathcal{X} = \{\mathcal{V} \setminus \{\mathcal{D}_i \cup \mathcal{Y}\}\} \cup \{s\}$. Thus, statement (i) is violated. A similar argument holds when $\mathcal{F}$ contains a minimal critical set.

"(i)$\Longleftarrow$(ii)" We again prove by contraposition. Suppose statement (i) is violated, i.e., there exists an f-total pair cut $\mathcal{H}$ w.r.t. a virtual node s corresponding to some minimal critical set $\mathcal{F}$. Consider a node i in $\mathcal{Y}$ (recall that $\mathcal{Y}$ is non-empty based on Definition 6). First consider the case when node i is not reachable from any node in $\mathcal{H}$ in the graph $\mathcal{G}$. It then follows that in the graph $\mathcal{G}$, directed paths to node i can only exist from the set $\mathcal{Y}$. But since $i \in \mathcal{Y}$ and $(\mathbf{A}, \mathbf{C}_{\mathcal{Y}})$ is not detectable, it is trivially impossible for node i to estimate the state. We thus focus on the case where node i is reachable from a certain set of nodes, say $\mathcal{D}_i$, within the set $\mathcal{H}$. Since $|\mathcal{H}| \leq 2f$ and $\mathcal{D}_i \subseteq \mathcal{H}$, we have that $|\mathcal{D}_i| \leq 2f$. It can be easily argued that the removal of $\mathcal{D}_i$ from $\mathcal{G}$ results in an induced subgraph where node i can only be reached from the set $\mathcal{Y}$. In other words, the set $\mathcal{P}_i$, as defined in the statement of Theorem 2, is a subset of $\mathcal{Y}$. As $(\mathbf{A}, \mathbf{C}_{\mathcal{Y}})$ is not detectable, it follows that $(\mathbf{A}, \mathbf{C}_{i\cup\mathcal{P}_i})$ is not detectable either, and thus statement (ii) is violated. □

Appendix B. Proof of Theorem 4

Proof. Let $\mathcal{A}$ be the (unknown) set of f-local adversaries, and consider $\mathcal{R} = \mathcal{V} \setminus \mathcal{A}$. Given a node $i \in \mathcal{R}$, the state vector $\mathbf{z}[k]$ can be partitioned into the components $\mathbf{z}_{\mathcal{O}_i}[k]$ and $\mathbf{z}_{\overline{\mathcal{O}}_i}[k]$ that correspond to the detectable and undetectable eigenvalues, respectively, of node i. Based on Lemma 2, we know that node i can estimate $\mathbf{z}_{\mathcal{O}_i}[k]$ asymptotically via a locally constructed Luenberger observer. It remains to show that node i can recover $\mathbf{z}_{\overline{\mathcal{O}}_i}[k]$, or in other words, for each $\lambda_j \in \overline{\mathcal{O}}_i$, we need to prove that $\lim_{k\to\infty} \|\hat{\mathbf{z}}_i^{(j)}[k] - \mathbf{z}^{(j)}[k]\| = 0$. To this end, consider a non-real eigenvalue $\lambda_j \in \Omega_U(\mathbf{A})$. As $\mathcal{G}$ contains a MEDAG for each $\lambda_j \in \Omega_U(\mathbf{A})$, the sets $\{\mathcal{L}_0^{(j)}, \mathcal{L}_1^{(j)}, \ldots, \mathcal{L}_q^{(j)}, \ldots \mathcal{L}_{T_j}^{(j)}\}$ form a partition of the set $\mathcal{R}$. We prove that each node in $\mathcal{R}$ can asymptotically estimate $\mathbf{z}^{(j)}[k]$ by inducting on the level number q.

For $q = 0$, by definition of the set $\mathcal{L}_0^{(j)}$, all nodes in $\mathcal{L}_0^{(j)}$ are regular and belong to the set $\mathcal{S}_j$, i.e., $\lambda_j \in \mathcal{O}_i$ for each node i in $\mathcal{L}_0^{(j)}$. Thus, by Lemma 2, each node in level 0 can estimate $\mathbf{z}^{(j)}[k]$ asymptotically. Notice that for any node i belonging to a level q, where $1 \leq q \leq T_j$, we have $\lambda_j \in \overline{\mathcal{O}}_i$. Consider a node i in $\mathcal{L}_1^{(j)}$ and let its error in estimation of the component $z^{(jm)}[k]$ be denoted by $e_i^{(jm)}[k] \triangleq \hat{z}_i^{(jm)}[k] - z^{(jm)}[k]$. The estimation errors of the individual components are aggregated in the vector $\mathbf{e}_i^{(j)}[k] = \hat{\mathbf{z}}_i^{(j)}[k] - \mathbf{z}^{(j)}[k]$. Subtracting $\mathbf{z}^{(j)}[k+1]$ from both sides of Eq. (5), noting that $\mathbf{z}^{(j)}[k+1] = \mathbf{W}(\lambda_j)\mathbf{z}^{(j)}[k]$ (based on the dynamics given by (3)), and using (4), we obtain

$$\mathbf{e}_i^{(j)}[k+1] = \mathbf{W}(\lambda_j) \underbrace{\begin{bmatrix} \sum_{l\in\mathcal{M}_i^{(j1)}[k]} w_{il}^{(j1)}[k] e_l^{(j1)}[k] \\ \vdots \\ \sum_{l\in\mathcal{M}_i^{(j\sigma_j)}[k]} w_{il}^{(j\sigma_j)}[k] e_l^{(j\sigma_j)}[k] \end{bmatrix}}_{\bar{\mathbf{e}}_i^{(j)}[k]}, \tag{B.1}$$

where $\sigma_j = 2a_{\mathbf{A}}(\lambda_j)$ (since λ_j is non-real). For arriving at (B.1), we used the fact that $\sum_{l\in\mathcal{M}_i^{(jm)}[k]} w_{il}^{(jm)}[k] = 1$ for every component m of $\mathbf{z}^{(j)}[k]$. We now analyze the error dynamics (B.1). To this end, for each component m of the vector $\mathbf{z}^{(j)}[k]$, we partition the set $\mathcal{N}_i^{(j)}$ into the sets $\mathcal{U}_i^{(jm)}[k]$, $\mathcal{J}_i^{(jm)}[k]$, and $\mathcal{M}_i^{(jm)}[k]$, such that the sets $\mathcal{U}_i^{(jm)}[k]$ and $\mathcal{J}_i^{(jm)}[k]$ contain f nodes each, with the highest and lowest estimate values for $z^{(jm)}[k]$ respectively, transmitted to node i at time-step k, and $\mathcal{M}_i^{(jm)}[k]$ contains the rest of the nodes in $\mathcal{N}_i^{(j)}$. According to the update rule (4), node i only uses estimates from the set $\mathcal{M}_i^{(jm)}[k]$ (which is non-empty at all time-steps based on the properties of a MEDAG) to compute the quantity $\bar{z}_i^{(jm)}[k]$. Now, for any component m of $\mathbf{z}^{(j)}[k]$, consider the following two cases. (i) $\mathcal{M}_i^{(jm)}[k] \cap \mathcal{A} = \emptyset$, *i.e., there are no adversarial nodes in the set* $\mathcal{M}_i^{(jm)}[k]$: in this case, all the nodes in the set $\mathcal{M}_i^{(jm)}[k]$ are regular and belong to $\mathcal{L}_0^{(j)}$ (as $\mathcal{N}_i^{(j)} \cap \mathcal{R} \subseteq \mathcal{L}_0^{(j)}$). (ii) $\mathcal{M}_i^{(jm)}[k] \cap \mathcal{A}$ *is non-empty, i.e., there are some adversarial nodes in the set* $\mathcal{M}_i^{(jm)}[k]$: based on the f-local adversarial model, it is apparent that each of the sets $\mathcal{U}_i^{(jm)}[k]$ and $\mathcal{J}_i^{(jm)}[k]$ contains at least one regular node belonging to $\mathcal{L}_0^{(j)}$. Let u and v be two such regular nodes belonging to $\mathcal{U}_i^{(jm)}[k]$ and $\mathcal{J}_i^{(jm)}[k]$, respectively. Based on the definitions of the sets $\mathcal{U}_i^{(jm)}[k]$, $\mathcal{J}_i^{(jm)}[k]$, and $\mathcal{M}_i^{(jm)}[k]$, we have $\hat{z}_v^{(jm)}[k] \leq \hat{z}_l^{(jm)}[k] \leq \hat{z}_u^{(jm)}[k]$, and hence $e_v^{(jm)}[k] \leq e_l^{(jm)}[k] \leq e_u^{(jm)}[k]$, for every node $l \in \mathcal{M}_i^{(jm)}[k]$. In particular, since $u, v \in \mathcal{L}_0^{(j)}$, it follows that for any node $l \in \mathcal{M}_i^{(jm)}[k]$, $e_{min}^{(jm)}[k] \leq e_l^{(jm)}[k] \leq e_{max}^{(jm)}[k]$, where $e_{min}^{(jm)}[k] = \min_{u\in\mathcal{L}_0^{(j)}} e_u^{(jm)}[k]$ and $e_{max}^{(jm)}[k] = \max_{u\in\mathcal{L}_0^{(j)}} e_u^{(jm)}[k]$. This property holds for every component m of $\mathbf{z}^{(j)}[k]$. Analyzing each of the two cases, we infer that at every time-step k, each component of the vector $\bar{\mathbf{e}}_i^{(j)}[k]$ in (B.1) lies in the convex hull of the corresponding components of the error vectors $\mathbf{e}_u^{(j)}[k]$, $u \in \mathcal{L}_0^{(j)} = \mathcal{S}_j \cap \mathcal{R}$. Based on Lemma 2, we have that $\lim_{k\to\infty} \mathbf{e}_u^{(j)}[k] = \mathbf{0}$, $\forall u \in \mathcal{S}_j \cap \mathcal{R}$, and hence it follows that $\hat{\mathbf{z}}_i^{(j)}[k]$ converges asymptotically to $\mathbf{z}^{(j)}[k]$ for every regular node i in $\mathcal{L}_1^{(j)}$.

Suppose the result holds for all levels from 0 to q (where $1 \leq q \leq T_j - 1$). It is easy to see that the result holds for all regular nodes in $\mathcal{L}_{q+1}^{(j)}$ as well, by noting the following. (i) A regular node $i \in \mathcal{L}_{q+1}^{(j)}$ has $\mathcal{N}_i^{(j)} \cap \mathcal{R} \subseteq \bigcup_{r=0}^{q} \mathcal{L}_r^{(j)}$. (ii) For each $i \in \mathcal{L}_{q+1}^{(j)}$, a similar analysis reveals that at every time-step k, each component of the vector $\bar{\mathbf{e}}_i^{(j)}[k]$ lies in the convex hull of the corresponding components of the error vectors $\mathbf{e}_u^{(j)}[k]$, $u \in \bigcup_{r=0}^{q} \mathcal{L}_r^{(j)}$. The desired result then follows from the induction hypothesis. An identical argument can be sketched for a real eigenvalue $\lambda_j \in \Omega_U(\mathbf{A})$, and thus the result holds for any $\lambda_j \in \Omega_U(\mathbf{A})$. We arrive at the conclusion that every node $i \in \mathcal{R}$ can asymptotically estimate $\mathbf{z}^{(j)}[k]$ for every eigenvalue $\lambda_j \in \overline{\mathcal{O}}_i$. Thus, each node $i \in \mathcal{R}$ can asymptotically estimate $\mathbf{z}[k]$, and hence $\mathbf{x}[k] = \mathbf{T}\mathbf{z}[k]$. □

References

Bai, C.-Z., Pasqualetti, F., & Gupta, V. (2017). Data-injection attacks in stochastic control systems: Detectability and performance tradeoffs. *Automatica*, *82*, 251–260.

Battistelli, G., & Chisci, L. (2014). Kullback–Leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability. *Automatica*, *50*(3), 707–718.

Battistelli, G., Chisci, L., Mugnai, G., Farina, A., & Graziano, A. (2015). Consensus-based linear and nonlinear filtering. *IEEE Transactions on Automatic Control*, *60*(5), 1410–1415.

Chen, Y., Kar, S., & Moura, J. M. (2018). Resilient distributed estimation through adversary detection. *IEEE Transactions on Signal Processing*, *66*(9), 2455–2469.

Chong, M. S., Wakaiki, M., & Hespanha, J. P. (2015). Observability of linear systems under adversarial attacks. In *Proc. of the American control conf.* (pp. 2439–2444).

Deghat, M., Ugrinovskii, V., Shames, I., & Langbort, C. (2019). Detection and mitigation of biasing attacks on distributed estimation networks. *Automatica*, *99*, 369–381.

Dolev, D., Lynch, N. A., Pinter, S. S., Stark, E. W., & Weihl, W. E. (1986). Reaching approximate agreement in the presence of faults. *Journal of the ACM*, *33*(3), 499–516.

Estrin, D., Govindan, R., Heidemann, J., & Kumar, S. (1999). Next century challenges: Scalable coordination in sensor networks. In *Proc. of the annual ACM/IEEE international conf. on mobile comp. and networking* (pp. 263–270). ACM.

Fawzi, H., Tabuada, P., & Diggavi, S. (2014). Secure estimation and control for cyber-physical systems under adversarial attacks. *IEEE Transactions on Automatic Control*, *59*(6), 1454–1467.

Forti, N., Battistelli, G., Chisci, L., Li, S., Wang, B., & Sinopoli, B. (2018). Distributed joint attack detection and secure state estimation. *IEEE Transactions on Signal and Information Processing over Networks*, *4*(1), 96–110.

Han, W., Trentelman, H. L., Wang, Z., & Shen, Y. (2019). A simple approach to distributed observer design for linear systems. *IEEE Transactions on Automatic Control*, *64*(1), 329–336.

Hashlamoun, W., Brahma, S., & Varshney, P. K. (2018). Mitigation of Byzantine attacks on distributed detection systems using audit bits. *IEEE Transactions on Signal and Information Processing over Networks*, *4*(1), 18–32.

Hespanha, J. P. (2009). *Linear systems theory*. Princeton university press.

Horn, R. A., & Johnson, C. R. (2012). *Matrix analysis*. Cambridge university press.

Janson, S., Łuczak, T., Turova, T., & Vallier, T. (2012). Bootstrap percolation on the random graph $G_{N,P}$. *Annals of Applied Probability*, *22*(5), 1989–2047.

Kamal, A. T., Farrell, J. A., Roy-Chowdhury, A. K., et al. (2013). Information weighted consensus filters and their application in distributed Camera networks. *IEEE Transactions on Automatic Control*, *58*(12), 3112–3125.

Khan, U., & Moura, J. M. (2008). Distributing the Kalman filter for large-scale systems. *IEEE Transactions on Signal Processing*, *56*(10), 4919–4935.

Khan, U., & Stankovic, A. M. (2013). Secure distributed estimation in cyber-physical systems. In *Proc. of the IEEE International conf. on acoustics, speech and signal processing* (pp. 5209–5213).

Kim, T., Shim, H., & Cho, D. D. (2016). Distributed Luenberger observer design. In *Proc. of the IEEE conf. on decision and control* (pp. 6928–6933).

Koo, C. Y. (2004). Broadcast in radio networks tolerating Byzantine adversarial behavior. In *Proc. of the ACM symp. on principles of distributed comp.* (pp. 275–282). ACM.

Kosut, O., Jia, L., Thomas, R. J., & Tong, L. (2011). Malicious data attacks on the smart grid. *IEEE Transactions on Smart Grid*, *2*(4), 645–658.

LeBlanc, H. J., Zhang, H., Koutsoukos, X., & Sundaram, S. (2013). Resilient asymptotic consensus in robust networks. *IEEE Journal on Selected Areas in Communications*, *31*(4), 766–781.

Lynch, N. A. (1996). *Distributed algorithms*. Morgan Kaufmann.

Matei, I., & Baras, J. S. (2012). Consensus-based linear distributed filtering. *Automatica*, *48*(8), 1776–1782.

Matei, I., Baras, J. S., & Srinivasan, V. (2012). Trust-based multi-agent filtering for increased smart grid security. In *Proc. of the med. conf. on control & autom.* (pp. 716–721).

Mishra, S., Shoukry, Y., Karamchandani, N., Diggavi, S. N., & Tabuada, P. (2017). Secure state estimation against sensor attacks in the presence of noise. *IEEE Transactions on Control of Network Systems*, *4*(1), 49–59.

Mitra, A., & Sundaram, S. (2016). Secure distributed observers for a class of linear time invariant systems in the presence of Byzantine adversaries. In *Proc. of the IEEE conf. on decision and control* (pp. 2709–2714).

Mitra, A., & Sundaram, S. (2018). Byzantine-Resilient Distributed Observers for LTI Systems. arXiv:1802.09651.

Mitra, A., & Sundaram, S. (2018b). Distributed observers for LTI systems. *IEEE Transactions on Automatic Control*, *63*(11), 3689–3704.

Mitra, A., & Sundaram, S. (2018c). Secure distributed state estimation of an LTI system over time-varying networks and analog erasure channels. In *Proc. of the American control conf.* (pp. 6578–6583).

Mo, Y., & Sinopoli, B. (2015). Secure estimation in the presence of integrity attacks. *IEEE Transactions on Automatic Control*, *60*(4), 1145–1151.

del Nozal, A. R., Millán, P., Orihuela, L., Seuret, A., & Zaccarian, L. (2019). Distributed estimation based on multi-hop subspace decomposition. *Automatica*, *99*, 213–220.

Olfati-Saber, R. (2005). Distributed Kalman filter with embedded consensus filters. In *Proc. of the IEEE conf. on decision and control and european control conf.* (pp. 8179–8184).

Pagourtzis, A., Panagiotakos, G., & Sakavalas, D. (2017). Reliable broadcast with respect to topology knowledge. *Distributed Computing*, *30*(2), 87–102.

Pajic, M., Lee, I., & Pappas, G. J. (2017). Attack-resilient state estimation for noisy dynamical systems. *IEEE Transactions on Control of Network Systems*, *4*(1), 82–92.

Park, S., & Martins, N. C. (2017). Design of distributed LTI observers for state omniscience. *IEEE Transactions on Automatic Control*, *62*(2), 561–576.

Pasqualetti, F., Bicchi, A., & Bullo, F. (2012). Consensus computation in unreliable networks: A system theoretic approach. *IEEE Transactions on Automatic Control*, *57*(1), 90–104.

Pasqualetti, F., Dörfler, F., & Bullo, F. (2013). Attack detection and identification in cyber-physical systems. *IEEE Transactions on Automatic Control*, *58*(11), 2715–2729.

Pelc, A., & Peleg, D. (2005). Broadcasting with locally bounded Byzantine faults. *Information Proceeding Letters*, *93*(3), 109–115.

Su, L., & Shahrampour, S. (2018). Finite-time Guarantees for Byzantine-Resilient Distributed State Estimation with Noisy Measurements. arXiv preprint arXiv:1810.10086.

Sundaram, S., & Hadjicostis, C. N. (2011). Distributed function calculation via linear iterative strategies in the presence of malicious agents. *IEEE Transactions on Automatic Control*, *56*(7), 1495–1508.

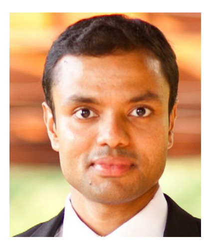
Teixeira, A., Shames, I., Sandberg, H., & Johansson, K. H. (2015). A secure control framework for resource-limited adversaries. *Automatica*, *51*, 135–148.

Vaidya, N. H., Tseng, L., & Liang, G. (2012). Iterative approximate Byzantine consensus in arbitrary directed graphs. In *Proc. of the ACM symp. on principles of distributed comp.* (pp. 365–374).

Wang, L., & Morse, A. (2018). A distributed observer for a time-invariant linear system. *IEEE Transactions on Automatic Control*, *63*(7), 2123–2130.

Wang, S., & Ren, W. (2017). On the convergence conditions of distributed dynamic state estimation using sensor networks: A unified framework. *IEEE Transactions on Control Systems Technology*, *26*(4), 1300–1316.

Aritra Mitra received the B.E. degree from Jadavpur University, Kolkata, India, and the M.Tech. degree from the Indian Institute of Technology Kanpur, India, in 2013 and 2015, respectively, both in electrical engineering. He is currently working toward the Ph.D. degree in electrical engineering at the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA. His current research interests include the design of distributed algorithms for estimation, inference and learning, networked control systems, and secure control. He was a recipient of the University Gold Medal at Jadavpur University and the Academic Excellence Award at IIT Kanpur.



Shreyas Sundaram is an Associate Professor in the School of Electrical and Computer Engineering at Purdue University. He received his M.S. and Ph.D. degrees in Electrical Engineering from the University of Illinois at Urbana–Champaign in 2005 and 2009, respectively. He was a Postdoctoral Researcher at the University of Pennsylvania from 2009 to 2010, and an Assistant Professor in the Department of Electrical and Computer Engineering at the University of Waterloo from 2010 to 2014. He is a recipient of the NSF CAREER award, and an Air Force Research Lab Summer Faculty Fellowship. At Purdue, he received the Hesselberth Award for Teaching Excellence and the Ruth and Joel Spira Outstanding Teacher Award. At Waterloo, he received the Department of Electrical and Computer Engineering Research Award and the Faculty of Engineering Distinguished Performance Award. He received the M. E. Van Valkenburg Graduate Research Award and the Robert T. Chien Memorial Award from the University of Illinois, and he was a finalist for the Best Student Paper Award at the 2007 and 2008 American Control Conferences. His research interests include network science, analysis of large-scale dynamical systems, fault-tolerant and secure control, linear system and estimation theory, game theory, and the application of algebraic graph theory to system analysis.