

Distributed State Estimation over Time-Varying Graphs: Exploiting the Age-of-Information

Aritra Mitra, John A. Richards, Saurabh Bagchi and Shreyas Sundaram

Abstract—We study the problem of designing a distributed observer for an LTI system over a time-varying communication graph. The limited existing work on this topic imposes various restrictions either on the observation model or on the sequence of communication graphs. In contrast, we propose a single-time-scale distributed observer that works under mild assumptions. Specifically, our communication model only requires strong-connectivity to be preserved over non-overlapping, contiguous intervals that are even allowed to grow unbounded over time. We show that under suitable conditions that bound the growth of such intervals, joint observability is sufficient to track the state of any discrete-time LTI system exponentially fast, at any desired rate. We also develop a variant of our algorithm that is provably robust to worst-case adversarial attacks, provided the sequence of graphs is sufficiently connected over time. The key to our approach is the notion of a “freshness-index” that keeps track of the age-of-information being diffused across the network. Such indices enable nodes to reject stale estimates of the state, and, in turn, contribute to stability of the error dynamics.

I. INTRODUCTION

Given a discrete-time LTI system $\mathbf{x}[k+1] = \mathbf{A}\mathbf{x}[k]$, and a linear measurement model $\mathbf{y}[k] = \mathbf{C}\mathbf{x}[k]$, a classical result in control theory states that one can design an observer that generates an asymptotically correct estimate $\hat{\mathbf{x}}[k]$ of the state $\mathbf{x}[k]$, if and only if the pair $(\mathbf{A}, \mathbf{C})$ is detectable [1]. Additionally, if the pair $(\mathbf{A}, \mathbf{C})$ is observable, then one can achieve exponential convergence at any desired convergence rate. Over the last couple of decades, significant effort has been directed towards studying the distributed counterpart of the above problem, wherein observations of the process are distributed among a set of sensors modeled as nodes of a communication graph [2]–[13]. A fundamental question that arises in this context is as follows: What are the minimal requirements on the measurement structure of the nodes and the underlying communication graph that guarantee the existence of a distributed observer? Here, by a distributed observer, we imply a set of state estimate update and information exchange rules that enable each node to track the entire state asymptotically.

A. Mitra, S. Bagchi and S. Sundaram are with the School of Electrical and Computer Engineering at Purdue University. J. A. Richards is with Sandia National Laboratories. Email: {mitral4, sbagchi, sundara2}@purdue.edu, jaricha@sandia.gov. This work was supported in part by NSF CAREER award 1653648, and by the Laboratory Directed Research and Development program at Sandia National Laboratories. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy’s National Nuclear Security Administration under contract DE-NA0003525. The views expressed in the article do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

The question posed above was answered only recently in [5]–[11] for static graphs. However, when the underlying network changes with time, very little is known about the distributed state estimation problem - a gap that we intend to bridge in this paper. Our study is motivated by applications in environmental monitoring tasks using mobile robot teams [14], [15], where time-varying communication graphs arise naturally either as a consequence of robot-mobility, or due to packet drops and link failures typical in wireless sensor networks.

Related Work: The existing literature on the design of distributed observers can be broadly classified in terms of the assumptions made on the system model, the observation model, and the network structure. Recent works on this topic [5]–[11] that make minimal system- and graph-theoretic assumptions can be further classified based on a finer set of attributes, as follows. (i) Does the approach require multiple consensus iterations between two consecutive time-steps of the dynamics?¹ (ii) What is the dimension of the estimator maintained by each node? (iii) Is the convergence asymptotic, or in finite-time? (iv) In case the convergence is asymptotic, can the convergence rate be controlled? The techniques proposed in [5]–[11] operate on a single-time-scale, and those in [6], [8]–[11] require observers of dimension no more than that of the state of the system. The aspect of reduced order distributed observer design is studied in [16]. The notion of convergence in each of the papers [5]–[11] is asymptotic; the ones in [7], [8], [10], [11] can achieve exponential convergence at *any* desired rate. For dynamic networks, there is much less work. Two recent papers [12] and [17] provide theoretical guarantees for certain specific classes of time-varying graphs; while the former proposes a two-time-scale approach, the latter develops a single-time-scale algorithm. However, the extent to which such results can be generalized has remained open.

Contributions: The main contribution of this paper is the development of a single-time-scale distributed state estimation algorithm in Section IV that requires each node to maintain an estimator of dimension equal to that of the state (along with some simple counters), and works under assumptions that are remarkably mild in comparison with those in the existing literature. Specifically, in terms of the observation model, we only require joint observability, i.e., the state is observable w.r.t. the collective observations of the nodes. This assumption is quite standard, and can be relaxed to joint detectability, with appropriate implications for the convergence rate.

However, the key departure from existing literature comes

¹Such approaches, referred to as two-time-scale approaches, incur high communication cost, and might hence be prohibitive for real-time applications.

from the generality of our communication model, introduced formally in Section II. Based on this model, we require strong-connectivity to be preserved over non-overlapping, contiguous intervals that can even grow linearly with time at a certain rate. In other words, we allow the inter-communication intervals between the nodes to potentially grow unbounded. Even under the regime of such sparse communications, we establish that our proposed approach can track the state of any discrete-time LTI system exponentially fast, at *any* desired convergence rate. While all the works on distributed state estimation that we are aware of provide an asymptotic analysis, we show that under a suitable selection of the observer gains, one can achieve finite-time convergence based on our approach. To put our results into context, it is instructive to compare them with the work closest to ours, namely [12]. In [12], the authors study a continuous-time analog of the problem under consideration, and develop a solution that leverages an elegant connection to the problem of distributed linear-equation solving [18]. In contrast to our technique, the one in [12] is inherently a two-time-scale approach, requires each node to maintain and update auxiliary state estimates, and works under the assumption that the communication graph is strongly-connected at every instant.

Our work is also related to the vast literature on consensus [19] and distributed optimization [20] over time-varying graphs. It is important to recognize that, in contrast to these settings, the problem at hand requires tracking potentially unstable external dynamics, making the stability analysis considerably more challenging. In particular, one can no longer directly leverage convergence properties of products of stochastic matrices. From a system-theoretic point of view, the error dynamics under our communication model evolves based on a switched linear system, where certain modes are potentially unstable. Thus, one way to analyze such dynamics is by drawing on techniques from the switched system stability literature [21]. However, such an analysis can potentially obscure the role played by the network structure. Instead, we directly exploit the interplay between the structure of the changing graph patterns on the one hand, and the evolution of the estimation error dynamics on the other, to provide a simple, self-contained stability analysis of our algorithm.

Finally, in Section VI, we extend our study of distributed state estimation over dynamic graphs to settings where certain nodes are under attack, and can act arbitrarily. For the problem under consideration, accounting for adversarial behavior is highly non-trivial even for static graphs, with only a few recent results that provide any theoretical guarantees [22]–[25]. Accounting for worst-case adversarial behavior over time-varying networks is considerably harder, since the compromised nodes can not only transmit incorrect, inconsistent state estimates, but also lie about the time-stamps of their data. Nonetheless, we develop a novel algorithm to handle such scenarios, and establish its correctness when the graph-sequence is sufficiently connected over time.

The key idea behind our approach is the use of a suitably defined “freshness-index” that keeps track of the age-of-information being diffused across the network. Loosely speaking, the “freshness-index” of a node quantifies the extent

to which its estimate of the state is outdated. Exchanging such indices enables a node to assess, in real-time, the quality of the estimate of a neighbor. Accordingly, it can reject estimates that are relatively stale – a fact that contributes to the stability of the error dynamics. We point out that while this is perhaps the first use of the notion of age-of-information (AoI) in a networked control/estimation setting, such a concept has been widely employed in the study of various queuing-theoretic problems arising in wireless networks [26]–[28].²

To sum up, we propose the first single-time-scale distributed state estimation algorithm that provides both finite-time and exponentially fast convergence guarantees, under significantly milder assumptions on the time-varying graph sequences than those in existing works. Moreover, we show how our approach can be extended to allow for worst-case adversarial attacks.

A preliminary version of this paper appeared as [29]. Here, we significantly expand upon the content in [29] in two main directions. First, we generalize the results in [29] to allow for unbounded inter-communication intervals. Second, the extension to adversarial scenarios in Section VI is a new addition entirely. We also provide full proofs of all our results.

II. PROBLEM FORMULATION AND BACKGROUND

System and Measurement Model: We are interested in collaborative state estimation of a discrete-time LTI system of the form:

$$\mathbf{x}[k+1] = \mathbf{A}\mathbf{x}[k], \quad (1)$$

where $k \in \mathbb{N}$ is the discrete-time index, $\mathbf{A} \in \mathbb{R}^{n \times n}$ is the system matrix, and $\mathbf{x}[k] \in \mathbb{R}^n$ is the state of the system.³ A network of sensors, modeled as nodes of a communication graph, obtain partial measurements of the state of the above process as follows:

$$\mathbf{y}_i[k] = \mathbf{C}_i \mathbf{x}[k], \quad (2)$$

where $\mathbf{y}_i[k] \in \mathbb{R}^{r_i}$ represents the measurement vector of the i -th node at time-step k , and $\mathbf{C}_i \in \mathbb{R}^{r_i \times n}$ represents the corresponding observation matrix. Let $\mathbf{y}[k] = [\mathbf{y}_1^T[k] \ \cdots \ \mathbf{y}_N^T[k]]^T$ and $\mathbf{C} = [\mathbf{C}_1^T \ \cdots \ \mathbf{C}_N^T]^T$ represent the collective measurement vector at time-step k , and the collective observation matrix, respectively. The goal of each node i in the network is to generate an asymptotically correct estimate $\hat{\mathbf{x}}_i[k]$ of the true dynamics $\mathbf{x}[k]$. It may not be possible for any node i in the network to accomplish this task in isolation, since the pair $(\mathbf{A}, \mathbf{C}_i)$ may not be detectable in general. Throughout the paper, we will only assume that the pair $(\mathbf{A}, \mathbf{C})$ is observable; the subsequent developments can be readily generalized to the case when $(\mathbf{A}, \mathbf{C})$ is detectable.

Communication Network Model: As is evident from the above discussion, information exchange among nodes is necessary for all nodes to estimate the full state. At each

²The notion of age-of-information (AoI) was first introduced in [26] as a performance metric to keep track of real-time status updates in a communication system. In a wireless network, it measures the time elapsed since the generation of the packet most recently delivered to the destination. In Section IV, we will see how such a concept applies to the present setting.

³We use $\mathbb{N}$ and $\mathbb{N}_+$ to denote the set of non-negative integers and the set of positive integers, respectively.

time-step $k \in \mathbb{N}$, the available communication channels are modeled by a directed communication graph $\mathcal{G}[k] = (\mathcal{V}, \mathcal{E}[k])$, where $\mathcal{V} = \{1, \dots, N\}$ represents the set of nodes, and $\mathcal{E}[k]$ represents the edge set of $\mathcal{G}[k]$ at time-step k . Specifically, if $(i, j) \in \mathcal{E}[k]$, then node i can send information directly to node j at time-step k ; in such a case, node i will be called a neighbor of node j at time-step k . We will use $\mathcal{N}_i[k]$ to represent the set of all neighbors (excluding node i) of node i at time-step k . When $\mathcal{G}[k] = \mathcal{G} \ \forall k \in \mathbb{N}$, where $\mathcal{G}$ is a static, directed communication graph, the necessary and sufficient condition (on the system and network) to solve the distributed state estimation problem is the joint detectability of each source component of $\mathcal{G}$ [5].⁴ Our goal in this paper is to extend the above result to scenarios where the underlying communication graph is allowed to change over time. To this end, let the union graph over an interval $[k_1, k_2]$, $0 \leq k_1 \leq k_2$, denoted $\bigcup_{\tau=k_1}^{k_2} \mathcal{G}[\tau]$, indicate a graph with vertex set equal to $\mathcal{V}$, and edge set equal to the union of the edge sets of the individual graphs appearing over the interval $[k_1, k_2]$. Based on this convention, we now formally describe the communication patterns (induced by the sequence $\{G[k]\}_{k=0}^{\infty}$) that are considered in this paper. We assume that there exists a sequence $\mathbb{I} = \{t_0, t_1, \dots\}$ of increasing time-steps with $t_0 = 0$ and each $t_i \in \mathbb{N}_+$, satisfying the following conditions.

- (C1) Define the mapping $h : \mathbb{I} \rightarrow \mathbb{N}_+$ as $h(t_q) = t_{q+1} - t_q, \forall t_q \in \mathbb{I}$. We assume that $h(t_q)$ is a non-decreasing function of its argument.
- (C2) For each $k \in \mathbb{N}$, let $m(k) \triangleq \max\{t_q \in \mathbb{I} : t_q \leq k\}$, and $M(k) \triangleq \min\{t_q \in \mathbb{I} : t_q > k\}$. Define $g : \mathbb{N} \rightarrow \mathbb{N}_+$ as $g(k) = M(k) - m(k) = h(m(k))$. Then, we assume that the following holds:

$$\limsup_{k \rightarrow \infty} \frac{2(N-1)g(k)}{k} = \delta < 1. \quad (3)$$

- (C3) For each $t_q \in \mathbb{I}$, we assume that $\bigcup_{\tau=t_q}^{t_{q+1}-1} \mathcal{G}[\tau]$ is strongly-connected.

To understand the conditions above, and how they relate to existing conditions, let us start by noting that based on (C3), the union graph over each interval of the form $[t_q, t_{q+1} - 1]$ is strongly-connected, where $t_q \in \mathbb{I}$. The next natural question is: how large can the gap $t_{q+1} - t_q$ be? Clearly, larger gaps allow for more general graph sequences. To formally answer the question above, we introduce the mapping $h(\cdot)$ in (C1). The condition described in (3) essentially allows the intervals $t_{q+1} - t_q$ to grow linearly over time at a certain rate. In other words, based on our model, the intervals over which strong-connectivity is preserved can potentially grow unbounded over time. In contrast, the typical assumption made while studying consensus [19] or distributed optimization [20] over time-varying graphs is that there exists some $T \in \mathbb{N}_+$, such that $t_{q+1} - t_q \leq T, \forall t_q \in \mathbb{I}$, i.e., the inter-communication intervals are uniformly bounded. For the specific problem under study, the result closest to ours [12] further assumes that T as defined

above is 1, i.e., $\mathcal{G}[k]$ is strongly-connected $\forall k \in \mathbb{N}$. Thus, our communication model subsumes the cases discussed above.

While (C1) requires $h(\cdot)$ to be non-decreasing (to allow for sparse communication schemes), our results can be readily generalized to the case where $h(\cdot)$ is non-monotonic by modifying (C2). Finally, note that since we place no assumptions at all on the spectrum of the $\mathbf{A}$ matrix, stability of the estimation error process imposes natural restrictions on how fast the inter-communication intervals can be allowed to grow. Condition (C2) formalizes this intuition by constraining such intervals to grow at most linearly at a certain rate.

Consider the following two examples. (i) The mapping h satisfies $h(t_q) = T, \forall t_q \in \mathbb{I}$, where T is some positive integer. (ii) The mapping h satisfies $h(t_q) = \lfloor \sqrt{t_q + 1} \rfloor, \forall t_q \in \mathbb{I}$. It is easy to verify that (C2) is satisfied in each case. While in example (i), the inter-communication intervals are uniformly bounded, they are allowed to grow over time in example (ii).

Background: For communication graphs satisfying conditions (C1)-(C3) as described above, our **objective** will be to design a distributed algorithm that ensures $\lim_{k \rightarrow \infty} \|\hat{\mathbf{x}}_i[k] - \mathbf{x}[k]\| = 0, \forall i \in \mathcal{V}$, with $\hat{\mathbf{x}}_i[k]$ representing the estimate of the state $\mathbf{x}[k]$ maintained by node $i \in \mathcal{V}$. To this end, we recall the following result from [6].

Lemma 1. *Given a system matrix $\mathbf{A}$, and a set of N sensor observation matrices $\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_N$, define $\mathbf{C} \triangleq [\mathbf{C}_1^T \ \dots \ \mathbf{C}_N^T]^T$. Suppose $(\mathbf{A}, \mathbf{C})$ is observable. Then, there exists a similarity transformation matrix $\mathbf{T}$ that transforms the pair $(\mathbf{A}, \mathbf{C})$ to $(\bar{\mathbf{A}}, \bar{\mathbf{C}})$, such that*

$$\bar{\mathbf{A}} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{0} \\ \mathbf{A}_{21} & \mathbf{A}_{22} & \mathbf{0} \\ \vdots & \vdots & \vdots \\ \mathbf{A}_{N1} & \mathbf{A}_{N2} & \dots & \mathbf{A}_{N(N-1)} & \mathbf{A}_{NN} \end{bmatrix}, \quad (4)$$

$$\bar{\mathbf{C}} = \begin{bmatrix} \bar{\mathbf{C}}_1 \\ \bar{\mathbf{C}}_2 \\ \vdots \\ \bar{\mathbf{C}}_N \end{bmatrix} = \begin{bmatrix} \mathbf{C}_{11} & \mathbf{0} \\ \mathbf{C}_{21} & \mathbf{C}_{22} & \mathbf{0} \\ \vdots & \vdots & \vdots \\ \mathbf{C}_{N1} & \mathbf{C}_{N2} & \dots & \mathbf{C}_{N(N-1)} & \mathbf{C}_{NN} \end{bmatrix},$$

and the pair $(\mathbf{A}_{ii}, \mathbf{C}_{ii})$ is observable $\forall i \in \{1, 2, \dots, N\}$. $\square$

We use the matrix $\mathbf{T}$ given by Lemma 1 to perform the coordinate transformation $\mathbf{z}[k] = \mathbf{T}^{-1}\mathbf{x}[k]$, yielding:

$$\begin{aligned} \mathbf{z}[k+1] &= \bar{\mathbf{A}}\mathbf{z}[k], \\ \mathbf{y}_i[k] &= \bar{\mathbf{C}}_i\mathbf{z}[k], \quad \forall i \in \{1, \dots, N\}, \end{aligned} \quad (5)$$

where $\bar{\mathbf{A}} = \mathbf{T}^{-1}\mathbf{A}\mathbf{T}$ and $\bar{\mathbf{C}}_i = \mathbf{C}_i\mathbf{T}$ are given by (4). Commensurate with the structure of $\bar{\mathbf{A}}$, the vector $\mathbf{z}[k]$ is of the following form:

$$\mathbf{z}[k] = \begin{bmatrix} \mathbf{z}^{(1)}[k]^T & \dots & \mathbf{z}^{(N)}[k]^T \end{bmatrix}^T, \quad (6)$$

where $\mathbf{z}^{(j)}[k]$ will be referred to as the j -th sub-state. By construction, since the pair $(\mathbf{A}_{jj}, \mathbf{C}_{jj})$ is locally observable w.r.t. the measurements of node j , node j will be viewed as the unique source node for sub-state j . In this sense, the role of node j will be to ensure that each non-source node $i \in \mathcal{V} \setminus \{j\}$ maintains an asymptotically correct estimate of sub-state j . For a time-invariant strongly-connected graph, this is achieved in [6] by first constructing a spanning tree rooted at node j , and

⁴A source component of a static, directed graph is a strongly connected component with no incoming edges.

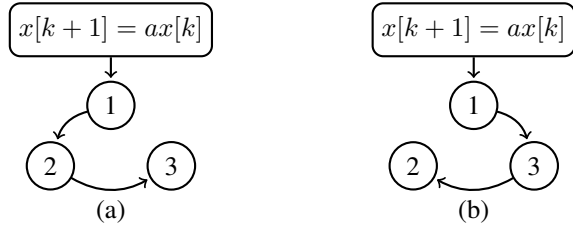


Fig. 1. An LTI system is monitored by a network of 3 nodes, where the communication graph $\mathcal{G}[k]$ switches between the two graphs shown above.

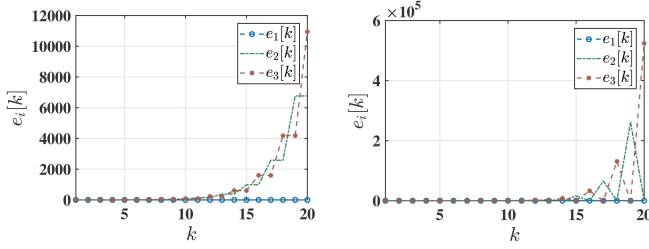


Fig. 2. Estimation error plots of the nodes for the model in Figure 1. Simulations are performed for a model where $a = 2$. The figure on the left corresponds to the case where consensus weights are distributed uniformly among neighbors, while the one on the right is the case where weights are placed along a tree rooted at node 1.

then requiring nodes to only listen to their parents in such a tree for estimating sub-state j . The resulting unidirectional flow of information from the source j to the rest of the network guarantees stability of the error process for sub-state j [6].

However, the above strategy is no longer applicable when the underlying communication graph is time-varying, for the following reasons. (i) For a given sub-state j , there may not exist a common spanning tree rooted at node j in each graph $\mathcal{G}[k]$, $k \in \mathbb{N}$. (ii) Assuming that a specific spanning tree rooted at node j is guaranteed to repeat at various points in time (not necessarily periodically), is restrictive, and qualifies as only a special case of conditions (C1)-(C3). (iii) Suppose for simplicity that $\mathcal{G}[k]$ is strongly-connected at each time-step (as in [12]), and hence, there exists a spanning tree $\mathcal{T}_j[k]$ rooted at node j in each such graph. For estimating sub-state j , suppose consensus at time-step k is performed along the spanning tree $\mathcal{T}_j[k]$. As we demonstrate in the next section, switching between such spanning trees can lead to unstable error processes over time. Thus, if one makes no further assumptions on the system model beyond joint observability, or on the sequence of communication graphs beyond conditions (C1)-(C3), ensuring stability of the estimation error dynamics becomes a challenging proposition. Nonetheless, we develop a simple algorithm in Section IV that bypasses the above problems. In the next section, we provide an example that helps to build the intuition behind this algorithm.

III. AN ILLUSTRATIVE EXAMPLE

Consider a network of 3 nodes monitoring a scalar unstable process $x[k+1] = ax[k]$, as shown in Figure 1. The communication graph $\mathcal{G}[k]$ switches between the two topologies shown in Figure 1. Specifically, $\mathcal{G}[k]$ is the graph in Figure

1(a) at all even time-steps, and the one in 1(b) at all odd time-steps. Node 1 is the only node with non-zero measurements, and thus acts as the source node for this network. Suppose for simplicity that it perfectly measures the state at all time-steps, i.e., its state estimate is $\hat{x}_1[k] = x[k]$, $\forall k \in \mathbb{N}$. Given this setup, a standard consensus based state estimate update rule would take the form (see for example [5], [6], [12]):

$$\hat{x}_i[k+1] = a \left(\sum_{j \in \mathcal{N}_i[k] \cup \{i\}} w_{ij}[k] \hat{x}_j[k] \right), i \in \{2, 3\}, \quad (7)$$

where the weights $w_{ij}[k]$ are non-negative, and satisfy $\sum_{j \in \mathcal{N}_i[k] \cup \{i\}} w_{ij}[k] = 1$, $\forall k \in \mathbb{N}$. The key question is: how should the consensus weights be chosen to guarantee stability of the estimation errors of nodes 2 and 3? Even for this simple example, if such weights are chosen naively, then the errors may grow unbounded over time. To see this, consider the following two choices: (1) consensus weights are distributed evenly over the set $\mathcal{N}_i[k] \cup \{i\}$, and (2) consensus weights are placed along the tree rooted at node 1. In each case, the error dynamics are unstable, as depicted in Figure 2. To overcome this problem, suppose nodes 2 and 3 are aware of the fact that node 1 has perfect information of the state. Since nodes 2 and 3 have no measurements of their own, intuitively, it makes sense that they should place their consensus weights entirely on node 1 whenever possible. The trickier question for node 2 (resp., node 3) is to decide *when* it should listen to node 3 (resp., node 2). Let us consider the situation from the perspective of node 2. At time-step 0, it adopts the information of node 1, and hence, the error of node 2 is zero at time-step 1. However, the error of node 3 is not necessarily zero at time-step 1. Consequently, if node 2 places a non-zero consensus weight on the estimate of node 3 at time-step 1, its error at time-step 2 might assume a non-zero value. Clearly, at time-step 1, node 2 is better off rejecting the information from node 3, and simply running open-loop. *The main take-away point here is that adoption or rejection of information from a neighbor should be based on the quality of information that such a neighbor has to offer.* In particular, a node that has come in contact with node 1 more recently is expected to have better information about the state than the other. Thus, to dynamically evaluate the quality of an estimate, the above reasoning suggests the need to introduce a metric that keeps track of how recent that estimate is with respect to (w.r.t.) the estimate of the source node 1. In the following section, we formalize this idea by introducing such a metric.

IV. ALGORITHM

Building on the intuition developed in the previous section, we introduce a new approach to designing distributed observers for a general class of time-varying networks. The main idea is the use of a “freshness-index” that keeps track of how delayed the estimates of a node are w.r.t. the estimates of a source node. Specifically, for updating its estimate of $\mathbf{z}^{(j)}[k]$, each node $i \in \mathcal{V}$ maintains and updates at every time-step a freshness-index $\tau_i^{(j)}[k]$. At each time-step $k \in \mathbb{N}$, the index $\tau_i^{(j)}[k]$ plays the following role: it determines whether

Algorithm 1

- 1: **Initialization:** $\tau_j^{(j)}[0] = 0, \tau_i^{(j)}[0] = \omega, \forall i \in \mathcal{V} \setminus \{j\}$.
- 2: **Update Rules for the Source Node:** Node j maintains $\tau_j^{(j)}[k] = 0, \forall k \in \mathbb{N}$. It updates $\hat{\mathbf{z}}_j^{(j)}[k]$ as:

$$\hat{\mathbf{z}}_j^{(j)}[k+1] = \mathbf{F}_{jj}\hat{\mathbf{z}}_j^{(j)}[k] + \sum_{q=1}^{(j-1)} \mathbf{G}_{jq}\hat{\mathbf{z}}_j^{(q)}[k] + \mathbf{L}_j\mathbf{y}_j[k], \quad (8)$$

where $\mathbf{F}_{jj} = (\mathbf{A}_{jj} - \mathbf{L}_j\mathbf{C}_{jj})$, $\mathbf{G}_{jq} = (\mathbf{A}_{jq} - \mathbf{L}_j\mathbf{C}_{jq})$, and $\mathbf{L}_j$ is an observer gain to be designed later.

- 3: **Update Rules for the Non-Source Nodes:** Each non-source node $i \in \mathcal{V} \setminus \{j\}$ operates as follows.
- 4: **Case 1:** $\tau_i^{(j)}[k] = \omega$. Define $\mathcal{M}_i^{(j)}[k] \triangleq \{l \in \mathcal{N}_i[k] : \tau_l^{(j)}[k] \neq \omega\}$. If $\mathcal{M}_i^{(j)}[k] \neq \emptyset$, let $u \in \arg\min_{l \in \mathcal{M}_i^{(j)}[k]} \tau_l^{(j)}[k]$. Node i updates $\tau_i^{(j)}[k]$ and $\hat{\mathbf{z}}_i^{(j)}[k]$ as:

$$\tau_i^{(j)}[k+1] = \tau_u^{(j)}[k] + 1, \quad (9)$$

$$\hat{\mathbf{z}}_i^{(j)}[k+1] = \mathbf{A}_{jj}\hat{\mathbf{z}}_u^{(j)}[k] + \sum_{q=1}^{(j-1)} \mathbf{A}_{jq}\hat{\mathbf{z}}_i^{(q)}[k]. \quad (10)$$

If $\mathcal{M}_i^{(j)}[k] = \emptyset$, then

$$\tau_i^{(j)}[k+1] = \omega, \quad (11)$$

$$\hat{\mathbf{z}}_i^{(j)}[k+1] = \mathbf{A}_{jj}\hat{\mathbf{z}}_i^{(j)}[k] + \sum_{q=1}^{(j-1)} \mathbf{A}_{jq}\hat{\mathbf{z}}_i^{(q)}[k]. \quad (12)$$

- 5: **Case 2:** $\tau_i^{(j)}[k] \neq \omega$. Define $\mathcal{F}_i^{(j)}[k] \triangleq \{l \in \mathcal{M}_i^{(j)}[k] : \tau_l^{(j)}[k] < \tau_i^{(j)}[k]\}$, where $\mathcal{M}_i^{(j)}[k]$ is as defined in line 4. If $\mathcal{F}_i^{(j)}[k] \neq \emptyset$, let $u \in \arg\min_{l \in \mathcal{F}_i^{(j)}[k]} \tau_l^{(j)}[k]$. Node i then updates $\tau_i^{(j)}[k]$ as per (9), and $\hat{\mathbf{z}}_i^{(j)}[k]$ as per (10). If $\mathcal{F}_i^{(j)}[k] = \emptyset$, then $\tau_i^{(j)}[k]$ is updated as

$$\tau_i^{(j)}[k+1] = \tau_i^{(j)}[k] + 1, \quad (13)$$

and $\hat{\mathbf{z}}_i^{(j)}[k]$ is updated as per (12).

node i should adopt the information received from one of its neighbors in $\mathcal{N}_i[k]$, or run open-loop, for updating $\hat{\mathbf{z}}_i^{(j)}[k]$, where $\hat{\mathbf{z}}_i^{(j)}[k]$ represents the estimate of $\mathbf{z}^{(j)}[k]$ maintained by node i . In case it is the former, it also indicates which specific neighbor in $\mathcal{N}_i[k]$ node i should listen to at time k ; this piece of information is particularly important for the problem under consideration, and ensures stability of the error process. The rules that govern the updates of the freshness indices $\tau_i^{(j)}[k]$, and the estimates $\hat{\mathbf{z}}_i^{(j)}[k]$, are formally stated in Algorithm 1. In what follows, we describe each of these rules.

Discussion of Algorithm 1: Consider any sub-state $j \in \{1, \dots, N\}$. Each node $i \in \mathcal{V}$ maintains an index $\tau_i^{(j)}[k] \in \{\omega\} \cup \mathbb{N}$, where ω is a dummy value. Specifically, $\tau_i^{(j)}[k] = \omega$ represents an “infinite-delay” w.r.t. the estimate of the source node for sub-state j , namely node j , i.e., it represents that node i has not received any information (either directly or indirectly) from node j regarding sub-state j up to time-step

k	$\tau_1[k]$	$\tau_2[k]$	$\tau_3[k]$	$\hat{x}_2[k]$	$\hat{x}_3[k]$
0	0	ω	ω	$\hat{x}_2[0]$	$\hat{x}_3[0]$
1	0	1	ω	$a\hat{x}_1[0]$	$a\hat{x}_3[0]$
2	0	2	1	$a^2\hat{x}_1[0]$	$a\hat{x}_1[1]$
3	0	1	2	$a\hat{x}_1[2]$	$a^2\hat{x}_1[1]$
4	0	2	1	$a^2\hat{x}_1[2]$	$a\hat{x}_1[3]$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Fig. 3. Illustration of Algorithm 1 for the example in Section III.

k . For estimation of sub-state j , since delays are measured w.r.t. the source node j , node j maintains its freshness-index $\tau_j^{(j)}[k]$ at zero for all time, to indicate a zero delay w.r.t. itself. For updating its estimate of $\mathbf{z}^{(j)}[k]$, it uses only its own information, as is evident from (8).

Every other node starts out with an “infinite-delay” ω w.r.t. the source (line 1 of Algo. 1). The freshness-index of a node $i \in \mathcal{V} \setminus \{j\}$ changes from ω to a finite value when it comes in contact with a neighbor with a finite delay, i.e., with a freshness-index that is not ω (line 4 of Algo. 1). At this point, we say that $\tau_i^{(j)}[k]$ has been “triggered”. Once triggered, at each time-step k , a non-source node i will adopt the information of a neighbor $l \in \mathcal{N}_i[k]$ only if node l ’s estimate of $\mathbf{z}^{(j)}[k]$ is “more fresh” relative to its own, i.e., only if $\tau_l^{(j)}[k] < \tau_i^{(j)}[k]$.⁵ Among the set of neighbors in $\mathcal{M}_i^{(j)}[k]$ (if $\tau_i^{(j)}[k]$ has not yet been triggered), or in $\mathcal{F}_i^{(j)}[k]$ (if $\tau_i^{(j)}[k]$ has been triggered), node i only adopts the information (based on (10)) of the neighbor u with the least delay. At this point, the delay of node i matches that of node u , and this fact is captured by the update rule (9). In case node i has no neighbor that has fresher information than itself w.r.t. sub-state j (where informativeness is quantified by $\tau_i^{(j)}[k]$), it increments its own freshness-index by 1 (as per (13)) to capture the effect of its own information getting older, and runs open-loop based on (12). Based on the above rules, at any given time-step k , $\tau_i^{(j)}[k]$ measures the age-of-information of $\hat{\mathbf{z}}_i^{(j)}[k]$, relative to the source node j . This fact is established later in Lemma 3. Finally, note that Algorithm 1 describes an approach for estimating $\mathbf{z}[k]$, and hence $\mathbf{x}[k]$, since $\mathbf{x}[k] = \mathbf{T}\mathbf{z}[k]$.

Example: Fig. 3 illustrates how the freshness-indices and state estimates of the nodes evolve for the example in Sec. III. Node 1 is the only source node in this example, and maintains $\tau_1[k] = 0, \forall k \in \mathbb{N}$. Observe that for $i \in \{2, 3\}$, (i) $\tau_i[k] \neq \omega, \forall k \geq 2$, (ii) $\tau_i[k] = m \implies \hat{x}_i[k] = a^m \hat{x}_i[k-m]$, where $m \in \mathbb{N}_+$, and (iii) $\tau_i[k] \leq 2, \forall k \geq 2$. In words, eventually, the estimates of the non-source nodes are at most 2-delayed versions of the estimate of the source node 1. Thus, since node 1 can estimate the state $x[k]$ on its own, the estimates of nodes 2 and 3 converge to $x[k]$ as well. This simple scalar example demonstrates the mechanism of Algo. 1, and provides intuition regarding why it works. As we shall see shortly, the above ideas translate to general dynamics in a natural way.

⁵Under Case 1 or Case 2 in Algo. 1, when a node $i \in \mathcal{V} \setminus \{j\}$ updates $\tau_i^{(j)}[k]$ via (9), and $\hat{\mathbf{z}}_i^{(j)}[k]$ via (10), we say that “ i adopts the information of u at time k for sub-state j ”; else, if it runs open-loop, we say it adopts its own information.

V. PERFORMANCE GUARANTEES FOR ALGORITHM 1

A. Statement of the Results

In this section, we state the theoretical guarantees afforded by Algorithm 1; proofs of these statements are deferred to Appendix A and B. We begin with one of our main results.

Theorem 1. *Given an LTI system (1), and a measurement model (2), suppose (A, C) is observable. Let the sequence of communication graphs $\{\mathcal{G}[k]\}_{k=0}^{\infty}$ satisfy conditions (C1)-(C3) in Section II. Then, given any desired convergence rate $\rho \in (0, 1)$, the observer gains $L_1, \dots, L_N$ can be designed in a manner such that the estimation error of each node converges to zero exponentially fast at rate ρ , based on Algorithm 1. $\square$*

The next result states that under the conditions in Theorem 1, one can in fact achieve finite-time convergence.

Proposition 1. (Finite-Time Convergence) *Suppose the conditions stated in Theorem 1 hold. Then, the observer gains $L_1, \dots, L_N$ can be designed in a manner such that the estimation error of each node converges to zero in finite-time.*

The next result follows directly from Prop. 1 and indicates that, when the sequence of communication graphs exhibits certain structure, one can derive a closed form expression for the maximum number of time-steps required for convergence.

Corollary 1. *Suppose the conditions stated in Theorem 1 hold. Additionally, suppose $h(t_q) \leq T, \forall t_q \in \mathbb{I}$, where $T \in \mathbb{N}_+$. Then, the observer gains $L_1, \dots, L_N$ can be designed in a manner such that the estimation error of each node converges to zero in at most $n + 2N(N - 1)T$ time-steps. $\square$*

A few remarks are now in order.

Remark 1. *The fact that a network of partially informed nodes can track the state of a dynamical system with arbitrarily large eigenvalues, over inter-communication intervals that can potentially grow unbounded, is non-obvious a priori. Our results in Thm. 1 and Prop. 1 indicate that not only can this be done exponentially fast at any desired rate, it can also be done in finite-time. In contrast, the result closest to ours [12] assumes strong-connectivity at each time-step - an assumption that is significantly stronger than what we make. $\square$*

Remark 2. *Notice that given a desired convergence rate ρ , the general design approach described in the proof of Thm. 1 (in Appendix A) offers a considerable degree of freedom in choosing the parameters $\lambda_1, \dots, \lambda_N$; the design flexibility so obtained in choosing the observer gains can be exploited to optimize transient performance, or performance against noise. In contrast, the proof of Prop. 1 highlights a specific approach to obtain finite-time convergence. However, such an approach may lead to undesirable transient spikes in the estimation errors, owing to large observer gains. $\square$*

Remark 3. *When strong-connectivity is preserved over uniformly bounded intervals, i.e., when $h(t_q) \leq T, \forall t_q \in \mathbb{I}$, for some $T \in \mathbb{N}_+$, our approach guarantees bounded estimation errors under bounded disturbances. However, as we will see in Sec. V-B, this may no longer be true if the inter-communication intervals grow unbounded, no matter how slowly. $\square$*

B. Implications of growing inter-communication intervals under bounded disturbances

While Theorem 1 shows that estimation is possible under growing inter-communication intervals, the goal of this section is to demonstrate via a simple example that this may no longer be true in the presence of disturbances. To this end, consider a scalar, unstable LTI system $x[k + 1] = ax[k] + d$, where $a > 1$, and $d > 0$ is a disturbance input to the system. The network comprises of just 2 nodes: node 1 with measurement model $y_1[k] = c_1x[k], c_1 \neq 0$, and node 2 with no measurements. Now consider an increasing sequence of time-steps $\mathbb{I} = \{t_0, t_1, \dots\}$ with $t_0 = 0$, and let $h(t_q) = t_{q+1} - t_q, \forall t_q \in \mathbb{I}$ be a non-decreasing function of its argument, as in Section II. Suppose the communication pattern comprises of an edge from node 1 to node 2 precisely at the time-steps given by $\mathbb{I}$. Node 1 maintains a standard Luenberger observer given by $\hat{x}_1[k + 1] = a\hat{x}_1[k] + l_1(y_1[k] - c_1\hat{x}_1[k])$, where l_1 is the observer gain. Node 2 applies Algorithm 1, which, in this case, translates to node 2 adopting the estimate of node 1 at each time-step t_q , and running open-loop at all other time-steps. Accordingly, we have $\hat{x}_2[t_{q+1}] = a^{h(t_q)}\hat{x}_1[t_q], \forall t_q \in \mathbb{I}$. With $e_i[k] = x[k] - \hat{x}_i[k], i \in \{1, 2\}$, one can then easily verify:

$$\begin{aligned} e_1[k + 1] &= \gamma e_1[k] + d, \forall k \in \mathbb{N}, \\ e_2[t_{q+1}] &= a^{h(t_q)} e_1[t_q] + d \frac{(a^{h(t_q)} - 1)}{(a - 1)}, \forall t_q \in \mathbb{I}, \end{aligned} \quad (14)$$

where $\gamma = (a - l_1 c_1)$. Now consider a scenario where the inter-communication intervals grow unbounded, i.e., $h(t_q) \rightarrow \infty$ as $t_q \rightarrow \infty$. Since $a > 1$ and $d > 0$, it is clear from (14) that the error subsequence $e_2[t_q], t_q \in \mathbb{I}$ will grow unbounded even if node 1 chooses l_1 such that $\gamma = 0$. For the specific example under consideration, although the above arguments were constructed w.r.t. our algorithm, it seems unlikely that the final conclusion would change if one were to resort to some other approach.⁶ The discussions in Section V can be thus summarized as follows.

- For a noiseless, disturbance free LTI system of the form (1), one can achieve exponential convergence at any desired rate, and even finite-time convergence based on Algorithm 1, under remarkably mild assumptions: joint observability, and joint strong-connectivity over intervals that can potentially grow unbounded.
- For an unstable system, any non-zero persistent disturbance, however small, can lead to unbounded estimation errors when the inter-communication intervals grow unbounded, no matter how slowly. Note however from Eq. (14) that our approach leads to bounded estimation errors under bounded disturbances if the sequence $\{h(t_q)\}_{t_q \in \mathbb{I}}$ is uniformly bounded above (see Remark 3).

In light of the above points, the reasons for stating our results in full generality, i.e., for unbounded communication intervals, are as follows. First, we do so for theoretical interest, since we believe our work is the first to establish that the distributed

⁶Note that we are only considering single-time-scale algorithms where nodes are not allowed to exchange their measurements. Also, we assume here that the nodes have no knowledge about the nature of the disturbance d , thereby precluding the use of any disturbance-rejection technique.

state estimation problem can be solved with growing inter-communication intervals. Second, we essentially get this result for “free”, i.e., accounting for such general scenarios incurs no additional steps in terms of the design of Algorithm 1. Finally, we emphasize that, while no existing approach can even handle the case where strong-connectivity is preserved over uniformly bounded time intervals (i.e., $\exists T \in \mathbb{N}_+$ such that $h(t_q) \leq T, \forall t_q \in \mathbb{I}$), the analysis for this scenario is simply a special case of that in Appendix A.

Comments on Algorithm 1: Algo. 1 requires a centralized design phase where the nodes implement the multi-sensor decomposition in Section II, and design their observer gains to achieve the desired convergence rate as outlined in the proof of Thm. 1.⁷ Moreover, the nodes need to know an upper-bound on the parameter δ in Eq. (3) while designing their observer gains. If, however, we constrain the inter-communication intervals to grow sub-linearly at worst, i.e. if $\delta = 0$, then such gains can be designed with no knowledge on the nature of the graph-sequences. Thus, there exists a trade-off between the generality of the graph sequences that can be tolerated, and the information required to do so. Notably, with the exception of [11] that studies continuous-time dynamics, all existing approaches we are aware of require a centralized design phase.

Notice that for the estimation of a given sub-state j , all nodes are reliant on the estimate of the unique source node for that sub-state, namely node j . Thus, even though other nodes may be in a position to estimate sub-state j as well, their measurement information is not made full use of in Algo. 1. While exploiting information redundancy is not necessary for the basic distributed observer design problem, it becomes important for improving estimation quality when measurements are noisy, or when certain nodes can drop out of the network. Thus, when it comes to these latter issues, Algo. 1 does have limitations. In the next section, we show exactly how to leverage measurement redundancy (under a slightly stronger assumption) to handle node drop-outs or even arbitrary malicious behavior; we anticipate that the approach in Section VI holds promise for tackling noise as well.

VI. RESILIENT DISTRIBUTED STATE ESTIMATION OVER TIME-VARYING NETWORKS

A. Description of Algorithm 2

We now consider a scenario where a subset of agents in the network is adversarial, and can deviate from the prescribed algorithm. We will show how one can employ the notion of freshness-indices to account for such adversarial agents over a time-varying network. To avoid cumbersome notation and to present the key ideas in a clear way, we will consider a scalar LTI system of the form $x[k+1] = ax[k]$. Later, we will discuss how our approach can be naturally extended to more general system models. We consider a worst-case Byzantine adversary model, where an adversarial node is assumed to be omniscient,

⁷Performing the decomposition in Lemma 1 requires assigning labels to the nodes, and also global knowledge of the observation model. While each choice of label assignment leads to a different coordinate transformation, the assertions of Theorem 1 and Proposition 1 hold *irrespective* of the way in which such labels are assigned.

Algorithm 2

- 1: **Initialization:** For each $i \in \mathcal{V} \setminus \mathcal{S}$, $\tau_i[0] = \omega$; each entry of $\mathbf{v}_i[0]$, $\mathbf{d}_i[0]$, and $\phi_i[0]$ is empty; $\mathcal{M}_i[0] = \emptyset$; and $q_i[0] = 0$. For each $i \in \mathcal{S}$, $\tau_i[0] = 0$.
- 2: **Update Rules for Source Nodes:** Each $i \in \mathcal{S}$ maintains $\tau_i[k] = 0, \forall k \in \mathbb{N}$. It updates $\hat{x}_i[k]$ based on the following Luenberger observer, where l_i is the observer gain:

$$\hat{x}_i[k+1] = a\hat{x}_i[k] + l_i(y_i[k] - c_i\hat{x}_i[k]). \quad (15)$$

- 3: **Update Rules for Non-Source Nodes:** At each $k \in \mathbb{N}$, every non-source node $i \in \mathcal{V} \setminus \mathcal{S}$ operates as follows (lines 4-18).
- 4: **Case 1:** $\tau_i[k] = \omega$. Define $\mathcal{J}_i[k] \triangleq \{j \in \mathcal{N}_i[k] : \tau_j[k] \neq \omega, \tau_j[k] \in \mathbb{N}, \tau_j[k] \leq k\}$.
- 5: **if** $|\mathcal{J}_i[k] \setminus \mathcal{M}_i[k]| < (2f+1) - q_i[k]$ **then**
- 6: Append each $l \in \mathcal{J}_i[k] \setminus \mathcal{M}_i[k]$ to $\mathcal{M}_i[k]$ ▷ **This involves adding l to $\mathcal{M}_i[k]$, and setting $v_{i,l}[k] = \hat{x}_l[k]$, $d_{i,l}[k] = \tau_l[k]$, and $\phi_{i,l}[k] = k$.**
- 7: Perform the following updates.

$$q_i[k+1] = q_i[k] + |\mathcal{J}_i[k] \setminus \mathcal{M}_i[k]|. \quad (16)$$

$$d_{i,l}[k+1] = d_{i,l}[k] + 1, l \in \mathcal{M}_i[k]. \quad (17)$$

$$\tau_i[k+1] = \omega. \quad (18)$$

$$\hat{x}_i[k+1] = a\hat{x}_i[k]. \quad (19)$$

- 8: **else**
- 9: Sort the nodes in $\mathcal{J}_i[k] \setminus \mathcal{M}_i[k]$ in ascending order of their freshness-indices $\tau_l[k], l \in \mathcal{J}_i[k] \setminus \mathcal{M}_i[k]$. Append the first $(2f+1) - q_i[k]$ nodes in the resulting list to $\mathcal{M}_i[k]$.
- 10: Set $q_i[\tau] = 2f+1, \forall \tau \geq k+1$; update $d_{i,l}[k], l \in \mathcal{M}_i[k]$ as per (17); update $\tau_i[k]$ as follows:

$$\tau_i[k+1] = \max_{l \in \mathcal{M}_i[k]} d_{i,l}[k] + 1. \quad (20)$$

- 11: For each $l \in \mathcal{M}_i[k]$, form the following quantity:

$$\bar{x}_{i,l}[k] = a^{(k-\phi_{i,l}[k])} \hat{x}_l[\phi_{i,l}[k]]. \quad (21)$$

- 12: Sort $\bar{x}_{i,l}[k], l \in \mathcal{M}_i[k]$ from highest to lowest, and reject the highest f and the lowest f of such quantities. Let the quantity that remains after such trimming be denoted $\bar{x}_i[k]$. Update $\hat{x}_i[k]$ as follows:

$$\hat{x}_i[k+1] = a\bar{x}_i[k]. \quad (22)$$

- 13: **end if**
- 14: Set $\mathcal{M}_i[k+1] = \mathcal{M}_i[k]$, $\mathbf{v}_i[k+1] = \mathbf{v}_i[k]$, and $\phi_i[k+1] = \phi_i[k]$.
- 15: **Case 2:** $\tau_i[k] \neq \omega$. For each $l \in \mathcal{J}_i[k] \cap \mathcal{M}_i[k]$, if $\tau_l[k] < d_{i,l}[k]$, then set $v_{i,l}[k] = \hat{x}_l[k]$, $d_{i,l}[k] = \tau_l[k]$, and $\phi_{i,l}[k] = k$ (we will classify this as an append operation).
- 16: Sort the nodes in $\mathcal{M}_i[k] \cup \{\mathcal{J}_i[k] \setminus \mathcal{M}_i[k]\}$ in ascending order of their freshness-indices, using $d_{i,l}[k]$ as the index for $l \in \mathcal{M}_i[k]$, and $\tau_l[k]$ as the index for $l \in \mathcal{J}_i[k] \setminus \mathcal{M}_i[k]$. Append the first $2f+1$ nodes in the resulting list to $\mathcal{M}_i[k]$.
- 17: For each $l \in \mathcal{M}_i[k]$, update $d_{i,l}[k]$ as per (17). Update $\tau_i[k]$ as per (20), and $\hat{x}_i[k]$ via (22) based on the filtering operation described in lines 11-12.
- 18: Execute the operations in line 14.

and can essentially act arbitrarily: it can transmit incorrect, potentially inconsistent information to its instantaneous out-neighbors, or choose not to transmit anything at all, while colluding with other adversaries in the process [30]–[32]. We will focus on an f -total adversarial model where the total number of adversaries in the network is bounded above by f , with $f \in \mathbb{N}$. The adversarial set will be denoted by $\mathcal{A}$, and the set of regular nodes by $\mathcal{R} = \mathcal{V} \setminus \mathcal{A}$. Finally, for the scalar model under consideration, we will define the set of source nodes as follows: $\mathcal{S} = \{i \in \mathcal{V} : c_i \neq 0, \text{ where } y_i[k] = c_i x[k]\}$, i.e., $i \in \mathcal{S}$ if and only if it can measure the state on its own.⁸ We will allow $\mathcal{S} \cap \mathcal{A} \neq \emptyset$.

Following the same line of reasoning as in Section IV, we are once again interested in answering the following question: When should a regular non-source node $i \in \{\mathcal{V} \setminus \mathcal{S}\} \cap \mathcal{R}$ use the information of a neighbor to update its estimate of the state? Unlike before, however, the presence of adversaries introduces certain complications: not only can an adversary transmit an arbitrary estimate of the state, it can also lie about its freshness-index. In particular, an adversary can follow the strategy of always reporting a freshness-index of 0 so as to prompt its instantaneous out-neighbors to use its estimate value. To carefully account for such misbehavior, we devise a novel protocol, namely Algorithm 2. In what follows, we explain the main idea behind Algorithm 2 in a nutshell.

High-level idea: Our goal is to ensure that the state estimate $\hat{x}_i[k]$ of a regular non-source node i is “close” to those of the regular source set $\mathcal{S} \cap \mathcal{R}$. To achieve this, we would like $\hat{x}_i[k]$ to be updated based on information that is neither too outdated, nor corrupted by the adversarial set. To meet the two requirements above, our main idea is to have node i store and process information received from a dynamic list of neighbors. At each time-step, the list is first updated so as to retain only those nodes that have the most recent state estimates w.r.t. the source set. Subsequently, $\hat{x}_i[k]$ is updated by filtering out extreme estimates in the list in an appropriate manner. We now describe in detail how the above steps are implemented.

Detailed Description: Like Algo. 1, Algo. 2 requires each node i to maintain a freshness-index $\tau_i[k] \in \mathbb{N} \cup \{\omega\}$. Each source node $i \in \mathcal{S}$ maintains $\tau_i[k] = 0$ for all time, and updates $\hat{x}_i[k]$ based on a standard Luenberger observer (see Eq. (15)). Given the presence of adversarial nodes in the network, each regular non-source node i relies on information redundancy for updating its estimate of the state. To achieve this, it maintains four additional vectors $\mathbf{v}_i[k]$, $\mathbf{d}_i[k]$, $\phi_i[k]$, and $\mathcal{M}_i[k]$, each of dimension $2f + 1$. The vector $\mathbf{v}_i[k]$ consists of state estimates received over time from $2f + 1$ distinct nodes. The labels of such nodes are stored in a list $\mathcal{M}_i[k]$, their freshness-indices in $\mathbf{d}_i[k]$, and the time-stamps associated with their estimates (i.e., the time-step at which their estimate is received) are recorded in $\phi_i[k]$. Algo. 2 requires each non-source node i to execute two key steps: (1) Maintaining a dynamical list $\mathcal{M}_i[k]$ of those $2f + 1$ neighbors that have the lowest freshness-indices based on all the information node i has acquired up to time k ; and (2) Performing a filtering operation to update $\hat{x}_i[k]$ based on

the latest state estimates of the nodes in the current list $\mathcal{M}_i[k]$. The second step, however, requires the cardinality of the set $\mathcal{M}_i[k]$ to be $2f + 1$. Thus, Algo. 2 involves an initial pre-filtering phase (lines 4–7) where a non-source node simply gathers enough estimates to later act on.

• *Discussion of Case 1:* We first describe the rules associated with the pre-filtering phase. Initially, each entry of $\mathbf{v}_i[0]$, $\mathbf{d}_i[0]$, and $\phi_i[0]$ is empty, and $\mathcal{M}_i[k]$ is an empty list (line 1). As time progresses, node i adds distinct nodes to the list $\mathcal{M}_i[k]$ based on rules that we will discuss shortly. Until the time when $|\mathcal{M}_i[k]| = 2f + 1$, node i maintains $\tau_i[k] = \omega$, and uses a counter $q_i[k]$ to keep track of the number of entries in $\mathcal{M}_i[k]$. When $\tau_i[k] = \omega$, node i operates as follows. It first considers the subset of neighbors $\mathcal{J}_i[k]$ at time k that have freshness-indices other than ω , belonging to $\mathbb{N}$, and at most k (line 4).⁹ In line 5 of Algo. 2, node i checks whether there are enough new nodes (i.e., nodes different from those already in $\mathcal{M}_i[k]$) in $\mathcal{J}_i[k]$ so as to bring the number of distinct entries in $\mathcal{M}_i[k]$ up to $(2f + 1)$. If not, it “appends” each node $l \in \mathcal{J}_i[k] \setminus \mathcal{M}_i[k]$ to $\mathcal{M}_i[k]$. By this, we mean that it adds l to $\mathcal{M}_i[k]$, sets $v_{i,l}[k] = \hat{x}_l[k]$, $d_{i,l}[k] = \tau_l[k]$, and $\phi_{i,l}[k] = k$ (see line 6). Here, we use the double subscript i, l to indicate the i -th node’s record of the various quantities associated with a node $l \in \mathcal{M}_i[k]$. Node i keeps track of the number of new nodes appended via (16). The entry $d_{i,l}[k]$ is the i -th node’s internal copy of the freshness-index of node l , which it updates via (17) for future comparisons (such as those in lines 15 and 16). As an indicator of the fact that it has not yet acquired $2f + 1$ state estimates to perform a filtering operation, node i sets $\tau_i[k + 1] = \omega$ (see Eq. (18)), and runs open-loop via (19).

Let us now discuss the case when $\tau_i[k]$ transitions from ω to some value other than ω (lines 8–12). At the transition time-step, node i appends those nodes from $\mathcal{J}_i[k] \setminus \mathcal{M}_i[k]$ to $\mathcal{M}_i[k]$ that have the lowest freshness-indices (see line 9). It does so in a way such that $|\mathcal{M}_i[k]|$ is precisely $2f + 1$. Since node i has now acquired enough estimates to perform the filtering step, it maintains $q_i[\tau] = 2f + 1$ for all $\tau \geq k + 1$ (see line 10). As in the non-adversarial setting, the freshness-index of a node i is a measure of how delayed its estimate is w.r.t. that of the source set $\mathcal{S}$. Since node i ’s estimate in turn is updated based on the estimates of nodes in $\mathcal{M}_i[k]$, its freshness-index is essentially dictated by the largest entry in $\mathbf{d}_i[k]$, i.e., the entry corresponding to the most delayed estimate. This facet is captured by (20). Lines 11–12 of Algo. 2 constitute the filtering step that a node $i \in \mathcal{V} \setminus \mathcal{S}$ employs to update $\hat{x}_i[k]$. Since the latest estimates of nodes in $\mathcal{M}_i[k]$ might have different time-stamps, we need a way to make meaningful comparisons between them. To this end, for each $l \in \mathcal{M}_i[k]$, node i constructs an intermediate quantity $\bar{x}_{i,l}[k]$ by propagating forward the latest estimate it has obtained from node l , namely $\hat{x}_l[\phi_{i,l}[k]]$, from time $\phi_{i,l}[k]$ to time k (see Eq. (21)).¹⁰ Note here that $\phi_{i,l}[k]$ is the latest time-step when node

⁹Note that in the absence of adversaries, any node j following Algo. 1 would satisfy $\tau_j[k] \in \mathbb{N}$ and $\tau_j[k] \leq k$, whenever $\tau_j[k] \neq \omega$. We would like the same to hold for any regular node following Algo. 2. Thus, any neighbor reporting otherwise need not be considered for inclusion in $\mathcal{M}_i[k]$.

¹⁰One can interpret $\bar{x}_{i,l}[k]$ as node i ’s prediction of node l ’s state estimate at time k , based on its current information.

⁸Note that with a slight abuse of terminology, the definition of source nodes here is different from that of a source node in Section II.

i appended node l to $\mathcal{M}_i[k]$. Having constructed the quantities $\bar{x}_{i,l}[k], l \in \mathcal{M}_i[k]$, node i then rejects the highest f and the lowest f of them (i.e., it filters out extreme values), and uses the one that remains, denoted $\bar{x}_i[k]$, to update $\hat{x}_i[k]$ via (22).

- *Discussion of Case 2:* The rules in lines 15-18 are essentially the same as those just discussed above for lines 8-12. The key difference between them stems from the manner in which $\mathcal{M}_i[k]$ is updated. In particular, when $\tau_i[k] = \omega$, or when $\tau_i[k]$ transitions from ω to a value other than ω , node i is less selective in terms of which nodes to include in $\mathcal{M}_i[k]$; at this point, its main concern is in gathering enough estimates for implementing the filtering step. In contrast, when $\tau_i[k] \neq \omega$, node i carefully examines the freshness-indices of its instantaneous neighbors prior to a potential inclusion in $\mathcal{M}_i[k]$. This is done in two steps. First, if it comes in contact with a node l that already exists in $\mathcal{M}_i[k]$, then it checks whether node l has fresher information to offer than when it was last appended to $\mathcal{M}_i[k]$ (see line 15). If so, node i replaces the entries corresponding to node l with their more recent versions. Next, in line 16, node i considers the set $\mathcal{M}_i[k] \cup \{\mathcal{J}_i[k] \setminus \mathcal{M}_i[k]\}$, and appends/retains only those $2f+1$ nodes that have the lowest freshness-indices.¹¹ The rationale behind breaking up the “appending” operation into two steps (lines 15-16) is to avoid the possibility of having multiple appearances of a single node in $\mathcal{M}_i[k]$. Note that for both Case 1 and Case 2, the values of $\mathcal{M}_i, \mathbf{v}_i$ and ϕ_i at the beginning of time-step $k+1$ are initialized with their values at the end of time-step k (lines 14 and 18); their values at the end of time-step $k+1$ will naturally depend on the new information available to node i at time $k+1$. Finally, we emphasize that the rules of Algo. 2 apply only to the regular nodes; an adversary can deviate from them in arbitrary ways.

VII. PERFORMANCE GUARANTEES FOR ALGORITHM 2

In order to state our main result concerning the performance of Algorithm 2, we need to recall the following concepts.

Definition 1. (*r*-reachable set) [31] Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a set $\mathcal{C} \subseteq \mathcal{V}$, and an integer $r \in \mathbb{N}_+$, $\mathcal{C}$ is said to be an *r*-reachable set if $\exists i \in \mathcal{C}$ such that $|\mathcal{N}_i \setminus \mathcal{C}| \geq r$. $\square$

Definition 2. (Strongly *r*-robust graph w.r.t. $\mathcal{S}$) [25] Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a set $\mathcal{S} \subset \mathcal{V}$, and an integer $r \in \mathbb{N}_+$, $\mathcal{G}$ is said to be strongly *r*-robust w.r.t. $\mathcal{S}$ if any non-empty subset $\mathcal{C} \subseteq \mathcal{V} \setminus \mathcal{S}$ is *r*-reachable. $\square$

Based on the above concepts, we now introduce the key graph property that will be of primary importance to us.

Definition 3. (Joint strong *r*-robustness w.r.t. $\mathcal{S}$) Given an integer $r \in \mathbb{N}_+$, and a set $\mathcal{S} \subset \mathcal{V}$, a sequence of graphs $\{\mathcal{G}[k]\}_{k=0}^\infty$ is said to be jointly strongly *r*-robust w.r.t. $\mathcal{S}$ if there exists $T \in \mathbb{N}_+$ such that $\bigcup_{\tau=kT}^{(k+1)T-1} \mathcal{G}[\tau]$ is strongly *r*-robust w.r.t. $\mathcal{S}, \forall k \in \mathbb{N}$. $\square$

¹¹In implementing this step, suppose a node l already existing in $\mathcal{M}_i[k]$ gets retained in $\mathcal{M}_i[k]$. Suppose the information concerning node l was stored in the p -th components of $\mathbf{d}_i[k], \mathbf{v}_i[k]$ and $\phi_i[k]$. Then, node i continues to store node l 's information in the p -th components of the above vectors. This is done only to simplify some of the arguments (see proof of Lemma 5).

The main result of this section is as follows; its proof is deferred to Appendix C.

Theorem 2. Consider a scalar LTI system of the form (1), and a measurement model of the form (2). Let the sequence of communication graphs $\{\mathcal{G}[k]\}_{k=0}^\infty$ be jointly strongly $(3f+1)$ -robust w.r.t. the source set $\mathcal{S}$. Then, based on Algorithm 2, the estimation error of each node $i \in \mathcal{R}$ can be made to converge to zero exponentially fast at any desired rate ρ , despite the actions of any f -total Byzantine adversarial set. $\square$

Remark 4. While in Def. 3, we required the strong-robustness property to be preserved over intervals of constant length T , we can easily allow for such intervals to grow linearly as well, in the spirit of condition (C2) in Section II. Moreover, following the proof of Theorem 2 in Section C, it is easy to see that if each regular source node i sets its observer gain l_i in (15) such that $a - l_i c_i = 0$, then the errors of all regular nodes will converge to 0 in at most $2(N - |\mathcal{S}|)T + 1$ time-steps. $\square$

Remark 5. The reason we require joint strong $(3f+1)$ -robustness as opposed to joint strong $(2f+1)$ -robustness is as follows. Consider a scenario where there are precisely $2f+1$ source nodes, f of whom are adversarial, and there is exactly one non-source node i . Suppose the graph sequence is jointly strongly $(2f+1)$ -robust w.r.t. $\mathcal{S}$, i.e., each source node will be in a position to transmit information to node i over every interval of the form $[kT, (k+1)T), k \in \mathbb{N}$. Suppose the f adversaries do not transmit at all. Node i will never be able to attribute such a phenomenon to adversarial behavior (since the lack of information from the adversarial nodes could be due to absence of links in the time-varying graph). Thus, $\tau_i[k]$ will equal ω for all time, and node i will keep running open-loop forever, thereby causing Algorithm 2 to fail. $\square$

Extension to vector dynamics: Note that there is a conceptual difference in our treatments of the adversarial and non-adversarial cases. For the non-adversarial case, Algorithm 1 relies on Lemma 1, which does not exploit information redundancy as we explained earlier. However, utilizing information redundancy is critical when dealing with adversaries, and this motivates a different initial transformation than the one in Lemma 1. In particular, transforming $\mathbf{A}$ to its real Jordan canonical form is the first step in extending Algorithm 2 to vector dynamical systems. When the system matrix $\mathbf{A}$ contains real, distinct eigenvalues, this amounts to a simple diagonalization. In the new coordinate frame, each state component is a scalar that can be treated analogously as in Algorithm 2. When $\mathbf{A}$ has arbitrary spectrum, one can combine ideas from [25] and Section VI. In each of the above cases, if every unstable mode (eigenvalue) λ_j of $\mathbf{A}$ is observable w.r.t. a set of nodes $\mathcal{F}_{\lambda_j}$, and (ii) joint strong $(3f+1)$ -robustness holds w.r.t. each such set $\mathcal{F}_{\lambda_j}$, then our results will go through.

At a high-level, the set $\mathcal{F}_{\lambda_j}$ can be viewed as a “source set” for each mode λ_j , just as node j was considered to be a source node for sub-state j in Algo. 1. In a broader sense, source nodes should be thought of as informative nodes that can estimate certain components of the state on their own; the specific definition of such nodes relies on the nature of the initial coordinate transformation. Suppose instead of

leveraging Lemma 1, we use a Jordan transformation when $f = 0$, i.e., when there are no adversaries (see [6, Section VI] for more details on this). To proceed, we would need to assume that the set $\mathcal{F}_{\lambda_j}$ (as defined above) is non-empty for each unstable eigenvalue λ_j - an assumption that is stronger than joint observability. This is why we resort to Lemma 1 for the non-adversarial case.

VIII. CONCLUSION

We proposed a novel approach to the design of distributed observers for LTI systems over time-varying networks. We proved that our algorithm guarantees exponential convergence at any desired rate (including finite-time convergence) under graph-theoretic conditions that are far milder than those existing. In particular, we showed that these results hold even when the inter-communication intervals grow unbounded over time. We then extended our framework to account for the possibility of worst-case adversarial attacks. In terms of future research directions, it would be interesting to explore the performance of our algorithm when the underlying network changes stochastically, as in [33]. We believe that the notion of a “freshness-index”, as employed in this paper, should be applicable to other related classes of problems that involve some form of collaborative estimation or inference - investigations along this line also merit attention.

APPENDIX A PROOF OF THEOREM 1

The goal of this section is to prove Theorem 1. Before delving into the technical details, we first provide an informal discussion of the main ideas underlying the proof of Theorem 1. To this end, let us fix a sub-state $j \in \{1, \dots, N\}$. The starting point of our analysis is Lemma 3 which establishes that for any non-source node $i \in \mathcal{V} \setminus \{j\}$, its error in estimation of sub-state j at time-step k can be expressed as a delayed version of the corresponding error of the source node j , where the delay is precisely the freshness-index $\tau_i^{(j)}[k]$. Given this result, we focus on bounding the delay $\tau_i^{(j)}[k]$ by exploiting the graph connectivity condition (C3). This is achieved in Lemma 4 where we first establish that $\tau_i^{(j)}[k]$ gets triggered after a finite period of time, and then show that it can be bounded above by the function $\tilde{g}(k) = 2(N-1)g(k)$, where $g(k)$ is as defined in Section II. At this point, we appeal to condition (C2) (which caps the rate of growth of $\tilde{g}(k)$) in designing the observer gain $\mathbf{L}_j$ at node j . Specifically, in the proof of Theorem 1, we carefully design $\mathbf{L}_j$ such that despite a potentially growing delay, every non-source node $i \in \mathcal{V} \setminus \{j\}$ inherits the same exponential convergence to the true dynamics $\mathbf{z}^{(j)}[k]$ as that of the corresponding source node j . With these ideas in place, we begin with a simple result that will be useful later on; it states that a non-source node for a certain sub-state will always adopt the information of the corresponding source node, whenever it is in a position to do so.

Lemma 2. *Consider any sub-state j , and suppose that at some time-step k , we have $j \in \mathcal{N}_i[k]$, for some $i \in \mathcal{V} \setminus \{j\}$. Then, based on Algorithm 1, we have:*

- (i) *If $\tau_i^{(j)}[k] = \omega$, then $j = \operatorname{argmin}_{l \in \mathcal{M}_i^{(j)}[k]} \tau_l^{(j)}[k]$.*
- (ii) *If $\tau_i^{(j)}[k] \neq \omega$, then $j = \operatorname{argmin}_{l \in \mathcal{F}_i^{(j)}[k]} \tau_l^{(j)}[k]$.*

□

Lemma 3. *Suppose all nodes employ Algorithm 1. Consider any sub-state j , and suppose that at some time-step k , we have $\tau_i^{(j)}[k] = m$, where $i \in \mathcal{V} \setminus \{j\}$, and $m \in \mathbb{N}_+$. Then, there exist nodes $v(\tau) \in \mathcal{V} \setminus \{j\}$, $\tau \in \{k-m+1, \dots, k\}$, such that the following is true:¹²*

$$\hat{\mathbf{z}}_i^{(j)}[k] = (\mathbf{A}_{jj})^m \hat{\mathbf{z}}_j^{(j)}[k-m] + \sum_{q=1}^{(j-1)} \sum_{\tau=k-m}^{(k-1)} (\mathbf{A}_{jj})^{k-\tau-1} \mathbf{A}_{jq} \hat{\mathbf{z}}_{v(\tau+1)}^{(q)}[\tau]. \quad (23)$$

□

For the proofs of Lemma 2 and 3, please refer to the conference version [29].

Lemma 4. *Suppose the sequence $\{\mathcal{G}[k]\}_{k=0}^\infty$ satisfies condition (C3) in Section II. Then, for each sub-state j , Algorithm 1 guarantees the following.*

$$\tau_i^{(j)}[k] \neq \omega, \forall k \geq \sum_{q=0}^{N-2} h(t_q), \forall i \in \mathcal{V}, \text{ and} \quad (24)$$

$$\tau_i^{(j)}[t_{p(N-1)}] \leq \sum_{q=(p-1)(N-1)}^{p(N-1)-1} h(t_q), \forall p \in \mathbb{N}_+, \forall i \in \mathcal{V}. \quad (25)$$

□

Proof. Fix a sub-state j , and notice that both (24) and (25) hold for the corresponding source node j , since $\tau_j^{(j)}[k] = 0, \forall k \in \mathbb{N}$. To establish these claims for the remaining nodes, we begin by making the following simple observation that follows directly from (9) and (13), and applies to every node $i \in \mathcal{V} \setminus \{j\}$:

$$\tau_i^{(j)}[k+1] \leq \tau_i^{(j)}[k] + 1, \text{ whenever } \tau_i^{(j)}[k] \neq \omega. \quad (26)$$

Our immediate goal is to establish (25) when $p = 1$ and, in the process, establish (24). Let $\mathcal{C}_0^{(j)} = \{j\}$, and define:

$$\mathcal{C}_1^{(j)} \triangleq \{i \in \mathcal{V} \setminus \mathcal{C}_0^{(j)} : \{\bigcup_{\tau=t_0}^{t_1-1} \mathcal{N}_i[\tau]\} \cap \mathcal{C}_0^{(j)} \neq \emptyset\}. \quad (27)$$

In words, $\mathcal{C}_1^{(j)}$ represents the set of non-source nodes (for sub-state j) that have a direct edge from node j at least once over the interval $[t_0, t_1)$. Based on condition (C3), $\mathcal{C}_1^{(j)}$ is non-empty (barring the trivial case when $\mathcal{V} = \{j\}$). For each $i \in \mathcal{C}_1^{(j)}$, it must be that $j \in \mathcal{M}_i^{(j)}[k]$ for some $k \in [t_0, t_1)$. Thus, based on (9) and (13), it must be that $\tau_i^{(j)}[k] \neq \omega, \forall k \geq t_1 = h(t_0), \forall i \in \mathcal{C}_1^{(j)}$. In particular, we note based on (26) that $\tau_i^{(j)}[t_1] \leq t_1$, and hence $\tau_i^{(j)}[t_{N-1}] \leq t_{N-1} = \sum_{q=0}^{N-2} h(t_q), \forall i \in \mathcal{C}_1^{(j)}$. We can keep

¹²Throughout the Appendix, superscripts within parentheses should be interpreted as indices, while those without parentheses should be interpreted as exponents.

repeating the above argument by recursively defining the sets $\mathcal{C}_r^{(j)}$, $1 \leq r \leq (N-1)$, as follows:

$$\mathcal{C}_r^{(j)} \triangleq \{i \in \mathcal{V} \setminus \bigcup_{q=0}^{(r-1)} \mathcal{C}_q^{(j)} : \{\bigcup_{\tau=t_{r-1}}^{t_r-1} \mathcal{N}_i[\tau]\} \cap \{\bigcup_{q=0}^{(r-1)} \mathcal{C}_q^{(j)}\} \neq \emptyset\}. \quad (28)$$

We proceed via induction on r . Suppose the following is true for all $r \in \{1, \dots, m-1\}$, where $m \in \{2, \dots, N-1\}$: $\tau_i^{(j)}[t_r] \neq \omega$ and $\tau_i^{(j)}[t_r] \leq t_r, \forall i \in \bigcup_{q=0}^r \mathcal{C}_q^{(j)}$. Now suppose

$r = m$. If $\mathcal{V} \setminus \bigcup_{q=0}^{(m-1)} \mathcal{C}_q^{(j)}$ is empty, then we are done establishing (24), and (25) for the case when $p = 1$. Else, based on condition (C3), it must be that $\mathcal{C}_m^{(j)}$ is non-empty. Consider a node $i \in \mathcal{C}_m^{(j)}$. Based on the way $\mathcal{C}_m^{(j)}$ is defined, note that at some time-step $\bar{k} \in [t_{m-1}, t_m)$, node i has some neighbor v (say) from the set $\bigcup_{q=0}^{(m-1)} \mathcal{C}_q^{(j)}$. Based on the

induction hypothesis and (26), it must be that $\tau_v^{(j)}[\bar{k}] \neq \omega$ and $\tau_v^{(j)}[\bar{k}] \leq \bar{k}$. At this point, if $\tau_i^{(j)}[\bar{k}] = \omega$, then since $v \in \mathcal{M}_i^{(j)}[\bar{k}]$, node i would update $\tau_i^{(j)}[\bar{k}]$ based on (9). Else, if $\tau_i^{(j)}[\bar{k}] \neq \omega$, there are two possibilities: (i) $v \in \mathcal{F}_i^{(j)}[\bar{k}]$, implying $\mathcal{F}_i^{(j)}[\bar{k}] \neq \emptyset$; or (ii) $v \notin \mathcal{F}_i^{(j)}[\bar{k}]$, implying $\tau_i^{(j)}[\bar{k}] \leq \tau_v^{(j)}[\bar{k}] \leq \bar{k}$. The above discussion, coupled with the freshness-index update rules for Case 2 of the algorithm (line 5 of Algo. 1), and (26), imply $\tau_i^{(j)}[t_m] \neq \omega$ and $\tau_i^{(j)}[t_m] \leq t_m$. This completes the induction step. Appealing to (26) once again, and noting that $\bigcup_{q=0}^{N-1} \mathcal{C}_q^{(j)} = \mathcal{V}$ and $t_{N-1} = \sum_{q=0}^{N-2} h(t_q)$, establishes (24), and (25) when $p = 1$.

In order to establish (25) for any $p \in \mathbb{N}_+$, one can follow a similar line of argument as above to analyze the evolution of the freshness indices over the interval $[t_{(p-1)(N-1)}, t_{p(N-1)}]$. In particular, for any $p > 1$, we can set $\mathcal{D}_0^{(j)} = \{j\}$, and define the sets $\mathcal{D}_r^{(j)}, 1 \leq r \leq (N-1)$ recursively as follows:

$$\mathcal{D}_r^{(j)} \triangleq \{i \in \mathcal{V} \setminus \bigcup_{q=0}^{(r-1)} \mathcal{D}_q^{(j)} : \{\bigcup_{\tau=t_{d(p,N,r)-1}}^{t_{d(p,N,r)}-1} \mathcal{N}_i[\tau]\} \cap \{\bigcup_{q=0}^{(r-1)} \mathcal{D}_q^{(j)}\} \neq \emptyset\}, \quad (29)$$

where $d(p, N, r) = (p-1)(N-1) + r$. One can then establish that $\tau_i^{(j)}[t_{d(p,N,r)}] \leq \sum_{q=(p-1)(N-1)}^{d(p,N,r)-1} h(t_q), \forall i \in \mathcal{D}_r^{(j)}, \forall r \in \{1, \dots, N-1\}$, via induction. $\square$

We are now in position to prove Theorem 1.

Proof. (Theorem 1) The proof is divided into two parts. In the first part, we describe a procedure for designing the observer gains $\{\mathbf{L}_i\}_{i=1}^N$. In the second, we establish that our design indeed leads to the desired convergence rate ρ .

Design of the observer gains: We begin by noting that for each sub-state j , one can always find scalars $\beta_j, \gamma_j \geq 1$, such that $\|(\mathbf{A}_{jj})^k\| \leq \beta_j \gamma_j^k, \forall k \in \mathbb{N}$ [34].¹³ Define $\gamma \triangleq \max_{1 \leq j \leq N} \gamma_j$. Next, fix a $\bar{\delta} \in (\delta, 1)$, where δ is as in (3). Given a desired rate of convergence $\rho \in (0, 1)$, we now recursively define two sets of positive scalars, namely $\{\rho_j\}_{j=1}^N$ and $\{\lambda_j\}_{j=1}^N$, starting

with $j = N$. With $\lambda_N = \rho$, let $\rho_j, j = N$, be chosen to satisfy:

$$\gamma^{\bar{\delta}} \rho_j^{1-\bar{\delta}} \leq \lambda_j. \quad (30)$$

Having picked $\rho_j \in (0, 1)$ to meet the above condition, we set λ_{j-1} to be any number in $(0, \rho_j)$, pick ρ_{j-1} to satisfy (30), and then repeat this process till we reach $j = 1$. Observe that the sets $\{\rho_j\}_{j=1}^N$ and $\{\lambda_j\}_{j=1}^N$ as defined above always exist, and satisfy: $\rho_1 < \lambda_1 < \rho_2 < \lambda_2 < \dots < \lambda_{N-1} < \rho_N < \lambda_N = \rho$. For each sub-state $j \in \{1, \dots, N\}$, let the corresponding source node j design the observer gain $\mathbf{L}_j$ (featuring in equation (8)) in a manner such that the matrix $(\mathbf{A}_{jj} - \mathbf{L}_j \mathbf{C}_{jj})$ has distinct real eigenvalues with spectral radius equal to ρ_j . Such a choice of $\mathbf{L}_j$ exists as the pair $(\mathbf{A}_{jj}, \mathbf{C}_{jj})$ is observable by construction. This completes our design procedure.

Convergence analysis: We first note that there exists a set of positive scalars $\{\alpha_1, \dots, \alpha_N\}$, such that [34]:

$$\|(\mathbf{A}_{jj} - \mathbf{L}_j \mathbf{C}_{jj})^k\| \leq \alpha_j \rho_j^k, \forall k \in \mathbb{N}. \quad (31)$$

For a particular sub-state j , let $\mathbf{e}_i^{(j)}[k] = \hat{\mathbf{z}}_i^{(j)}[k] - \mathbf{z}^{(j)}[k]$. Consider the first sub-state $j = 1$, and observe that based on (4), (5), and (8), the following is true: $\mathbf{e}_1^{(1)}[k+1] = (\mathbf{A}_{11} - \mathbf{L}_1 \mathbf{C}_{11}) \mathbf{e}_1^{(1)}[k]$. Thus, we obtain

$$\mathbf{e}_1^{(1)}[k] = (\mathbf{A}_{11} - \mathbf{L}_1 \mathbf{C}_{11})^k \mathbf{e}_1^{(1)}[0]. \quad (32)$$

Based on (31) and (32), we then have:

$$\|\mathbf{e}_1^{(1)}[k]\| \leq c_1 \rho_1^k, \forall k \in \mathbb{N}, \quad (33)$$

where $c_1 \triangleq \alpha_1 \|\mathbf{e}_1^{(1)}[0]\|$. Given that node 1's error for sub-state 1 decays exponentially as per (33), we want to now relate the errors $\mathbf{e}_i^{(1)}[k], i \in \mathcal{V} \setminus \{1\}$ of the non-source nodes (for sub-state 1) to $\mathbf{e}_1^{(1)}[k]$. To this end, consider any $i \in \mathcal{V} \setminus \{1\}$, and note that for any $k \geq t_{N-1}$, Eq. (24) in Lemma 4 implies that $\tau_i^{(1)}[k] \neq \omega$, and hence $\tau_i^{(1)}[k] \in \mathbb{N}_+$. Invoking Lemma 3, and using the fact that $\mathbf{z}^{(1)}[k] = (\mathbf{A}_{11})^m \mathbf{z}^{(1)}[k-m], \forall m \in \mathbb{N}$, we then obtain the following $\forall i \in \mathcal{V} \setminus \{1\}$:

$$\mathbf{e}_i^{(1)}[k] = (\mathbf{A}_{11})^{\tau_i^{(1)}[k]} \mathbf{e}_1^{(1)}[k - \tau_i^{(1)}[k]], \forall k \geq t_{N-1}. \quad (34)$$

Our next goal is to bound the delay term $\tau_i^{(1)}[k]$ in the above relation. For this purpose, consider any time-step $k \geq t_{N-1}$, and let $p(k)$ be the largest integer such that $t_{p(k)(N-1)} \leq k$. Then, for any sub-state j , and any $i \in \mathcal{V} \setminus \{j\}$, we observe:

$$\begin{aligned} \tau_i^{(j)}[k] &\stackrel{(a)}{\leq} \tau_i^{(j)}[t_{p(k)(N-1)}] + (k - t_{p(k)(N-1)}) \\ &\stackrel{(b)}{\leq} \sum_{q=(p(k)-1)(N-1)}^{p(k)(N-1)-1} h(t_q) + (k - t_{p(k)(N-1)}) \\ &\stackrel{(c)}{\leq} 2(N-1)h(m(k)) \stackrel{(d)}{=} 2(N-1)g(k). \end{aligned} \quad (35)$$

In the above inequalities, (a) follows from (24) in Lemma 4 and (26); (b) follows from (25) in Lemma 4; and (c) follows from the monotonicity of $h(\cdot)$ in condition (C1), and by recalling that $m(k) \triangleq \max\{t_q \in \mathbb{I} : t_q \leq k\}$. Finally, (d) follows by recalling that $g(k) = h(m(k))$. Recalling that $\|(\mathbf{A}_{11})^k\| \leq \beta_1 \gamma_1^k$, using the bounds in (33) and (35), the fact

¹³We use $\|\mathbf{A}\|$ to refer to the induced 2-norm of a matrix $\mathbf{A}$.

that $\gamma_1 \geq 1$ and $\rho_1 < 1$, and the sub-multiplicative property of the 2-norm, we obtain the following by taking norms on both sides of (34):

$$\left\| \mathbf{e}_i^{(1)}[k] \right\| \leq \bar{c}_1 \left(\frac{\gamma_1}{\rho_1} \right)^{\tilde{g}(k)} \rho_1^k, \forall k \geq t_{N-1}, \forall i \in \mathcal{V} \setminus \{1\}, \quad (36)$$

where $\tilde{g}(k) = 2(N-1)g(k)$ and $\bar{c}_1 \triangleq c_1\beta_1$. Based on condition (C3), and our choice of $\bar{\delta}$, observe that there exists $\bar{k}(\bar{\delta})$ such that $\tilde{g}(k) \leq \bar{\delta}k, \forall k \geq \bar{k}(\bar{\delta})$. With $k_1 \triangleq \max\{t_{N-1}, \bar{k}(\bar{\delta})\}$, we then obtain the following based on (30) and (36), for all $k \geq k_1$ and for all $i \in \mathcal{V} \setminus \{1\}$:

$$\left\| \mathbf{e}_i^{(1)}[k] \right\| \leq \bar{c}_1 \left(\gamma_1^{\bar{\delta}} \rho_1^{1-\bar{\delta}} \right)^k \leq \bar{c}_1 \left(\gamma_1^{\bar{\delta}} \rho_1^{1-\bar{\delta}} \right)^k \leq \bar{c}_1 \lambda_1^k. \quad (37)$$

Note that since $\bar{c}_1 \geq c_1$ and $\lambda_1 > \rho_1$, the above bound applies to node 1 as well (see equation (33)). We have thus established that exponential convergence at rate λ_1 for sub-state 1 holds for each node in the network.

Our aim is to now obtain a bound similar to that in (37) for each sub-state $j \in \{2, \dots, N\}$. To this end, with $g_{jq} = \|(\mathbf{A}_{jq} - \mathbf{L}_j \mathbf{C}_{jq})\|$ and $h_{jq} = \|\mathbf{A}_{jq}\|$, let us define the following quantities recursively for $j \in \{2, \dots, N\}$:

$$\begin{aligned} k_j &\triangleq \frac{k_{j-1}}{(1-\bar{\delta})}, \\ c_j &\triangleq \frac{\alpha_j}{\rho_j^{k_{j-1}}} \left(\left\| \mathbf{e}_j^{(j)}[k_{j-1}] \right\| + \sum_{q=1}^{(j-1)} \frac{g_{jq} \bar{c}_q}{(\rho_j - \lambda_q)} \lambda_q^{k_{j-1}} \right), \\ \bar{c}_j &\triangleq \beta_j \left(c_j + \sum_{q=1}^{(j-1)} \frac{h_{jq} \bar{c}_q}{(\gamma_j - \lambda_q)} \right), \end{aligned} \quad (38)$$

where $k_1 \triangleq \max\{t_{N-1}, \bar{k}(\bar{\delta})\}$, $c_1 \triangleq \alpha_1 \left\| \mathbf{e}_1^{(1)}[0] \right\|$, and $\bar{c}_1 = c_1\beta_1$. Based on the above definitions, we claim that for each sub-state $j \in \{1, \dots, N\}$, the following is true:

$$\left\| \mathbf{e}_i^{(j)}[k] \right\| \leq \bar{c}_j \lambda_j^k, \forall k \geq k_j, \forall i \in \mathcal{V}. \quad (39)$$

We will prove the above claim via induction on the sub-state number j . We have already established (39) for the base case when $j = 1$. For $j \geq 2$, our strategy will be to first analyze the evolution of $\mathbf{e}_j^{(j)}[k]$ at the source node j . From (4) and (5), we note that the dynamics of the j -th sub-state are coupled with those of the first $j-1$ sub-states. Thus, $\mathbf{e}_j^{(j)}[k]$ will exhibit exponential decay only when the errors for the first $j-1$ sub-states have already started decaying exponentially, with k_{j-1} (as defined in (38)) representing the instant when exponential decay for the $(j-1)$ -th sub-state kicks in. Let us now prove that as soon as this happens, the following holds:

$$\left\| \mathbf{e}_j^{(j)}[k] \right\| \leq c_j \rho_j^k, \forall k \geq k_{j-1}. \quad (40)$$

To do so, suppose (39) holds for all $j \in \{1, \dots, l-1\}$, where $l \in \{2, \dots, N\}$. Now let $j = l$ and observe that equations (4) and (5) yield:

$$\begin{aligned} \mathbf{z}^{(l)}[k+1] &= \mathbf{A}_{ll} \mathbf{z}^{(l)}[k] + \sum_{q=1}^{(l-1)} \mathbf{A}_{lq} \mathbf{z}^{(q)}[k] \\ &= (\mathbf{A}_{ll} - \mathbf{L}_l \mathbf{C}_{ll}) \mathbf{z}^{(l)}[k] + \sum_{q=1}^{(l-1)} (\mathbf{A}_{lq} - \mathbf{L}_l \mathbf{C}_{lq}) \mathbf{z}^{(q)}[k] + \mathbf{L}_l \mathbf{y}_l[k]. \end{aligned} \quad (41)$$

Based on the above equation and (8), we obtain:

$$\mathbf{e}_l^{(l)}[k+1] = (\mathbf{A}_{ll} - \mathbf{L}_l \mathbf{C}_{ll}) \mathbf{e}_l^{(l)}[k] + \sum_{q=1}^{(l-1)} (\mathbf{A}_{lq} - \mathbf{L}_l \mathbf{C}_{lq}) \mathbf{e}_l^{(q)}[k].$$

Rolling out the above equation starting from k_{l-1} yields:

$$\mathbf{e}_l^{(l)}[k] = (F_{ll})^{k-k_{l-1}} \mathbf{e}_l^{(l)}[k_{l-1}] + \sum_{q=1}^{(l-1)} \sum_{\tau=k_{l-1}}^{(k-1)} (F_{ll})^{k-\tau-1} G_{lq} \mathbf{e}_l^{(q)}[\tau], \quad (42)$$

where $F_{ll} = (\mathbf{A}_{ll} - \mathbf{L}_l \mathbf{C}_{ll})$, and $G_{lq} = (\mathbf{A}_{lq} - \mathbf{L}_l \mathbf{C}_{lq})$. Taking norms on both sides of the above equation, using the triangle inequality, and the sub-multiplicative property of the two-norm, we obtain:

$$\begin{aligned} \left\| \mathbf{e}_l^{(l)}[k] \right\| &\stackrel{(a)}{\leq} \alpha_l \rho_l^k \left(\frac{\left\| \mathbf{e}_l^{(l)}[k_{l-1}] \right\|}{\rho_l^{k_{l-1}}} + \frac{1}{\rho_l} \sum_{q=1}^{(l-1)} g_{lq} \sum_{\tau=k_{l-1}}^{(k-1)} \rho_l^{-\tau} \left\| \mathbf{e}_l^{(q)}[\tau] \right\| \right) \\ &\stackrel{(b)}{\leq} \alpha_l \rho_l^k \left(\frac{\left\| \mathbf{e}_l^{(l)}[k_{l-1}] \right\|}{\rho_l^{k_{l-1}}} + \frac{1}{\rho_l} \sum_{q=1}^{(l-1)} g_{lq} \bar{c}_q \sum_{\tau=k_{l-1}}^{\infty} \left(\frac{\lambda_q}{\rho_l} \right)^{\tau} \right) \\ &\stackrel{(c)}{\leq} c_l \rho_l^k, \forall k \geq k_{l-1}. \end{aligned} \quad (43)$$

In the above inequalities, (a) follows from (31) and by recalling that $g_{lq} = \|G_{lq}\|$; (b) follows by first applying the induction hypothesis noting that $q \leq (l-1)$ and $\tau \geq k_{l-1}$, and then changing the upper limit of the inner summation (over time); (c) follows by simplifying the preceding inequality using the fact that $\lambda_q < \rho_l, \forall q \in \{1, \dots, l-1\}$, and using the definition of c_l in (38). We have thus obtained a bound on the estimation error of sub-state l at node l . To obtain a similar bound for each $i \in \mathcal{V} \setminus \{l\}$, note that equation (41) can be rolled out over time to yield the following for each $m \in \mathbb{N}$:

$$\mathbf{z}^{(l)}[k] = (\mathbf{A}_{ll})^m \mathbf{z}^{(l)}[k-m] + \sum_{q=1}^{(l-1)} \sum_{\tau=(k-m)}^{(k-1)} (\mathbf{A}_{ll})^{k-\tau-1} \mathbf{A}_{lq} \mathbf{z}^{(q)}[\tau].$$

Leveraging Lemma 3, we can then obtain the following error dynamics for a node $i \in \mathcal{V} \setminus \{l\}, \forall k \geq t_{N-1}$.

$$\begin{aligned} \mathbf{e}_i^{(l)}[k] &= (\mathbf{A}_{ll})^{\tau_i^{(l)}[k]} \mathbf{e}_i^{(l)}[k - \tau_i^{(l)}[k]] \\ &\quad + \sum_{q=1}^{(l-1)} \sum_{\tau=(k-\tau_i^{(l)}[k])}^{(k-1)} (\mathbf{A}_{ll})^{k-\tau-1} \mathbf{A}_{lq} \mathbf{e}_{v(\tau+1)}^{(q)}[\tau]. \end{aligned} \quad (44)$$

Based on the above equation, we note that since $\mathbf{A}_{ll}$ can contain unstable eigenvalues, and since $\tau_i^{(l)}[k]$ may grow over time (owing to potentially growing inter-communication intervals), we need the decay in $\mathbf{e}_i^{(l)}[k - \tau_i^{(l)}[k]]$ to dominate the growth due to $(\mathbf{A}_{ll})^{\tau_i^{(l)}[k]}$ in order for $\mathbf{e}_i^{(l)}[k]$ to eventually remain bounded. To show that this is indeed the case, we begin by noting the following inequalities that hold when $k \geq k_l$:

$$\frac{k_{l-1}}{k} \stackrel{(a)}{\leq} 1 - \bar{\delta} \stackrel{(b)}{\leq} 1 - \frac{\tilde{g}(k)}{k} \stackrel{(c)}{\leq} 1 - \frac{\tau_i^{(l)}[k]}{k}, \quad (45)$$

where $\tilde{g}(k) = 2(N-1)g(k)$. In the above inequalities, (a) follows directly from (38); (b) follows by noting that $k \geq k_l \implies k \geq \bar{k}(\bar{\delta})$; and (c) follows from (35) and by noting that $k \geq k_l \implies k \geq t_{N-1}$. We conclude that if $k \geq k_l$, then $k - \tau_i^{(l)}[k] \geq k_{l-1}$. Thus, when $k \geq k_l$, at any time-step

$\tau \geq k - \tau_i^{(l)}[k]$, the errors of the first $l - 1$ sub-states would exhibit exponential decay based on the induction hypothesis. With this in mind, we fix $i \in \mathcal{V} \setminus \{l\}$, $k \geq k_l$, and bound $\mathbf{e}_i^{(l)}[k]$ by taking norms on both sides of (44), as follows:

$$\begin{aligned} \|\mathbf{e}_i^{(l)}[k]\| &\stackrel{(a)}{\leq} \beta_l \left(c_l \left(\frac{\gamma_l}{\rho_l} \right)^{\tilde{g}(k)} \rho_l^k + \gamma_l^{k-1} \sum_{q=1}^{(l-1)} h_{lq} \bar{c}_q \sum_{\tau=(k-\tau_i^{(l)}[k])}^{(k-1)} \left(\frac{\lambda_q}{\gamma_l} \right)^\tau \right) \\ &\stackrel{(b)}{\leq} \beta_l \left(c_l \left(\frac{\gamma_l}{\rho_l} \right)^{\tilde{g}(k)} \rho_l^k + \gamma_l^{k-1} \sum_{q=1}^{(l-1)} h_{lq} \bar{c}_q \sum_{\tau=(k-\tilde{g}(k))}^{\infty} \left(\frac{\lambda_q}{\gamma_l} \right)^\tau \right) \\ &\stackrel{(c)}{=} \beta_l \left(c_l \left(\frac{\gamma_l}{\rho_l} \right)^{\tilde{g}(k)} \rho_l^k + \sum_{q=1}^{(l-1)} \frac{h_{lq} \bar{c}_q}{(\gamma_l - \lambda_q)} \left(\frac{\gamma_l}{\lambda_q} \right)^{\tilde{g}(k)} \lambda_q^k \right) \\ &\stackrel{(d)}{\leq} \bar{c}_l \left(\gamma^\delta \rho_l^{1-\delta} \right)^k \stackrel{(e)}{\leq} \bar{c}_l \lambda_l^k. \end{aligned} \quad (46)$$

In the above steps, (a) follows by first recalling that $\|(\mathbf{A}_{ll})^k\| \leq \beta_l \gamma_l^k, \forall k \in \mathbb{N}$, $h_{lq} = \|\mathbf{A}_{lq}\|$, $\tilde{g}(k) = 2(N - 1)g(k)$, and then using the induction hypothesis, equations (35), (39), and (43), and the facts that $\rho_l < 1, \gamma_l \geq 1$; (b) follows by suitably changing the upper and lower limits of the inner summation (over time), a change that is warranted since each summand is non-negative; (c) follows by simplifying the preceding inequality; (d) follows by noting that $\lambda_q < \rho_l, \forall q \in \{1, \dots, l-1\}$, using the definition of $\bar{c}_l$ in (38), and the fact that $\tilde{g}(k) \leq \delta k, \forall k \geq k_l$; and finally (e) follows from (30). This completes the induction step. Let $\mathbf{e}_i[k] = \hat{\mathbf{z}}_i[k] - \mathbf{z}[k]$. Recalling that $\lambda_j \leq \rho, \forall j \in \{1, \dots, N\}$, we obtain as desired:

$$\|\mathbf{e}_i[k]\| = \sqrt{\sum_{j=1}^N \|\mathbf{e}_i^{(j)}[k]\|^2} \leq \left(\sqrt{\sum_{j=1}^N \bar{c}_j^2} \right) \rho^k, \forall k \geq k_N, \forall i \in \mathcal{V}. \quad \square$$

APPENDIX B PROOF OF PROPOSITION 1

Proof. (Proposition 1) For each sub-state $j \in \{1, \dots, N\}$, let the corresponding source node j design the observer gain $\mathbf{L}_j$ (featuring in equation (8)) in a manner such that the matrix $(\mathbf{A}_{jj} - \mathbf{L}_j \mathbf{C}_{jj})$ has all its eigenvalues at 0. Such a choice of $\mathbf{L}_j$ exists based on the fact that the pair $(\mathbf{A}_{jj}, \mathbf{C}_{jj})$ is observable by construction. Let $n_j = \dim(\mathbf{A}_{jj})$. Given the above choice of observer gains, we will prove the result by providing an upper bound on the number of time-steps it takes the error of each node to converge to $\mathbf{0}$. To this end, we first define a sequence $\{\bar{\tau}_j\}_{j=1}^N$ of time-steps as follows:

$$\bar{\tau}_1 = \inf\{\tau \in \mathbb{N}_+, \tau \geq t_{N-1} : k - \tilde{g}(k) \geq n_1, \forall k \geq \tau\}, \quad (47)$$

where $\tilde{g}(k) = 2(N - 1)g(k)$, and

$$\bar{\tau}_j = \inf\{\tau \in \mathbb{N}_+ : k - \tilde{g}(k) \geq \bar{\tau}_{j-1} + n_j, \forall k \geq \tau\}. \quad (48)$$

Based on condition (C3), namely $\limsup_{k \rightarrow \infty} g(k)/k = \delta < 1$, observe that $\bar{\tau}_j$ as defined above is finite $\forall j \in \{1, \dots, N\}$. Next, note that by construction, $(\mathbf{A}_{jj} - \mathbf{L}_j \mathbf{C}_{jj})$ is a nilpotent matrix of index at most n_j . Thus, it is easy to see that $\mathbf{e}_1^{(1)}[k] = \mathbf{0}, \forall k \geq n_1$, based on (32). Recall from (35) that for each sub-state j , $\tau_i^{(j)}[k] \leq \tilde{g}(k), \forall k \geq t_{N-1}, \forall i \in \mathcal{V}$. From the

definition of $\bar{\tau}_1$ in (47), and equation (34), we immediately obtain that $\mathbf{e}_i^{(1)}[k] = \mathbf{0}, \forall k \geq \bar{\tau}_1, \forall i \in \mathcal{V}$. One can easily generalize this argument to the remaining sub-states by using an inductive reasoning akin to that in the proof of Theorem 1. In particular, for any sub-state $j \in \{2, \dots, N\}$, one can roll out the error dynamics for node j as in (42), starting from time-step $\bar{\tau}_{j-1}$. By this time, the induction hypothesis would imply that the estimation errors of all nodes on all sub-states $q \in \{1, \dots, j-1\}$ have converged to zero. The nilpotency of $(\mathbf{A}_{jj} - \mathbf{L}_j \mathbf{C}_{jj})$ would then imply that $\mathbf{e}_j^{(j)}[k] = \mathbf{0}, \forall k \geq \bar{\tau}_{j-1} + n_j$. From the definition of $\bar{\tau}_j$ in (48), and (35), we note that $k \geq \bar{\tau}_j \implies k - \tau_i^{(j)}[k] \geq \bar{\tau}_{j-1} + n_j, \forall i \in \mathcal{V}$. Referring to (44), we conclude that $\mathbf{e}_i^{(j)}[k] = \mathbf{0}, \forall k \geq \bar{\tau}_j, \forall i \in \mathcal{V}$. Based on the above reasoning, the overall error for each node converges to $\mathbf{0}$ in at most $\bar{\tau}_N$ time-steps. $\square$

APPENDIX C PROOF OF THEOREM 2

In this section, we develop the proof of Theorem 2. We begin with the following lemma that characterizes the evolution of the vector of freshness-indices $\mathbf{d}_i[k]$.

Lemma 5. Suppose at any time-step $k \in \mathbb{N}_+$, $\tau_i[k] \neq \omega$ for some $i \in \{\mathcal{V} \setminus \mathcal{S}\} \cap \mathcal{R}$. Then, Algorithm 2 implies the following.

$$\mathbf{d}_i[k+1] \leq \mathbf{d}_i[k] + \mathbf{1}_{2f+1}, \quad (49)$$

$$\tau_i[k+1] \leq \tau_i[k] + 1, \quad (50)$$

where the first inequality holds component-wise. $\square$

Proof. Consider a node $i \in \{\mathcal{V} \setminus \mathcal{S}\} \cap \mathcal{R}$, and suppose $\tau_i[k] \neq \omega$. This implies that $|\mathcal{M}_i[k]| = 2f+1$, and hence, all entries of the vector $\mathbf{d}_i[k]$ are populated with non-negative integers. Now fix any component $p \in \{1, \dots, 2f+1\}$ of $\mathbf{d}_i[k]$, and suppose that it corresponds to node $l \in \mathcal{M}_i[k]$, i.e., focus on $d_{i,l}[k]$. This entry undergoes the following three operations (in order) prior to the end of time-step $k+1$: (i) It gets incremented by 1 as per (17); (ii) It gets potentially replaced by a value strictly smaller than $d_{i,l}[k] + 1$ if node i hears from node l at time $k+1$ (line 15 of Algo. 2); and (iii) It gets subjected to the operation in line 16 of Algo. 2. Clearly, after operations (i) and (ii), the entry corresponding to node l can increase by at most 1. At the end of operation (iii), node l either gets retained in $\mathcal{M}_i[k+1]$, or gets replaced. In case it is the former, we clearly have $d_{i,l}[k+1] \leq d_{i,l}[k] + 1$, and $d_{i,l}[k+1]$ gets stored in the p -th component of $\mathbf{d}_i[k+1]$ (see footnote 11). If node l gets removed from $\mathcal{M}_i[k+1]$, then given that the $2f+1$ nodes with the lowest freshness-indices populate $\mathcal{M}_i[k+1]$ (line 16 of Algo. 2), it must be that the node replacing l in $\mathcal{M}_i[k+1]$ has freshness-index at most $d_{i,l}[k] + 1$ at time $k+1$. Thus, regardless of whether node l gets retained or removed, we have argued that the p -th component of $\mathbf{d}_i[k]$ can increase by at most 1 by the end of time-step $k+1$. Noting that the above argument holds for any component p of $\mathbf{d}_i[k]$ establishes (49); Eq. (50) then follows readily from (20) and (49). $\square$

The next lemma is the analogue of Lemma 4 for the adversarial case. It tells us that the freshness-index of a regular node gets “triggered” (i.e., assumes a value other than ω) after

a finite period of time, and that it is eventually uniformly bounded above. Here, uniformity is w.r.t. all graph sequences that are jointly strongly $(3f + 1)$ -robust w.r.t. $\mathcal{S}$.

Lemma 6. *Suppose the conditions in the statement of Theorem 2 hold. Then, Algorithm 2 guarantees the following.*

$$\tau_i[k] \neq \omega, \forall k \geq (N - |\mathcal{S}|)T, \forall i \in \mathcal{R}, \text{ and} \quad (51)$$

$$\tau_i[k] \leq 2(N - |\mathcal{S}|)T, \forall k \geq (N - |\mathcal{S}|)T, \forall i \in \mathcal{R}, \quad (52)$$

where T has the same meaning as in Definition 3. $\square$

Proof. The proof of this result mirrors that of Lemma 4; hence, we only sketch the essential details. First, observe that the claims made in the statement of the lemma hold trivially for any $i \in \mathcal{S} \cap \mathcal{R}$, since each such node maintains $\tau_i[k] = 0$ for all time. Let us now focus on establishing (51) for all regular non-source nodes. To this end, let $\mathcal{C}_0 = \mathcal{S}$, and define:

$$\mathcal{C}_1 \triangleq \{i \in \mathcal{V} \setminus \mathcal{C}_0 : |\{\bigcup_{\tau=0}^{T-1} \mathcal{N}_i[\tau]\} \cap \mathcal{C}_0| \geq 3f + 1\}. \quad (53)$$

In words, $i \in \mathcal{C}_1$ if and only if node i has at least $3f + 1$ neighbors from the source set $\mathcal{C}_0 = \mathcal{S}$ over the interval $[0, T)$. Suppose $\mathcal{V} \setminus \mathcal{S} \neq \emptyset$ (else, there is nothing to prove). Then, $\mathcal{C}_1$ must also be non-empty based on the definition of joint strong $(3f + 1)$ -robustness w.r.t. $\mathcal{S}$. Now consider any $i \in \mathcal{C}_1 \cap \mathcal{R}$. Since at most f nodes are adversarial, node i must have heard from at least $2f + 1$ regular source nodes (not necessarily all at the same time-step) over the interval $[0, T)$, with each such node reporting a freshness-index of 0. Regardless of whether or not node i appends all such nodes to $\mathcal{M}_i$, the fact that it has the opportunity to do so, implies that $\tau_i[T] \neq \omega$. Moreover, given the fact that node i appends a node l to $\mathcal{M}_i[k]$ only if $\tau_l[k] \leq k$, and appealing to (50), we see that $\tau_i[T] \leq T$. Using (50) again, we have $\tau_i[(N - |\mathcal{S}|)T] \leq (N - |\mathcal{S}|)T$. Now define the sets $\mathcal{C}_r, 1 \leq r \leq (N - |\mathcal{S}|)$, recursively as follows:

$$\mathcal{C}_r \triangleq \{i \in \mathcal{V} \setminus \bigcup_{q=0}^{(r-1)} \mathcal{C}_q : |\{\bigcup_{\tau=(r-1)T}^{rT-1} \mathcal{N}_i[\tau]\} \cap \{\bigcup_{q=0}^{(r-1)} \mathcal{C}_q\}| \geq 3f + 1\}. \quad (54)$$

We can now employ an inductive argument similar to that in Lemma 4 to establish (51). In particular, proceeding as in the base case when $r = 1$, joint strong $(3f + 1)$ -robustness w.r.t. $\mathcal{S}$ implies that $\mathcal{C}_r \neq \emptyset$, whenever $\mathcal{V} \setminus \bigcup_{q=0}^{(r-1)} \mathcal{C}_q \neq \emptyset$. Since $|\mathcal{A}| \leq f$, any node $i \in \mathcal{C}_r \cap \mathcal{R}$ must then hear from at least $2f + 1$ regular nodes in $\bigcup_{q=0}^{(r-1)} \mathcal{C}_q$ over the interval $[(r - 1)T, rT)$. Moreover, based on an inductive reasoning, each such node would be a potential candidate for getting appended upon making contact with node i . It is easy to then see that for each $i \in \mathcal{C}_r \cap \mathcal{R}$, $\tau_i[rT] \neq \omega$ and, in particular, $\tau_i[rT] \leq rT$ based on (50). This establishes the claim in equation (51).

We now turn to establishing the correctness of (52). Let $\bar{T} = (N - |\mathcal{S}|)T$. We claim $\tau_i[p\bar{T}] \leq \bar{T}, \forall p \in \mathbb{N}_+, \forall i \in \mathcal{R}$. Note that the argument for the case when $p = 1$ follows from the above discussion, and by using (50). To prove the claim, it suffices to prove it for the case when $p = 2$, since an identical reasoning applies for all $p \geq 2$. For analyzing the case when

$p = 2$, let $\mathcal{D}_0 = \mathcal{S}$, and define the sets $\mathcal{D}_r, 1 \leq r \leq (N - |\mathcal{S}|)$, recursively as follows:

$$\mathcal{D}_r \triangleq \{i \in \mathcal{V} \setminus \bigcup_{q=0}^{(r-1)} \mathcal{D}_q : |\{\bigcup_{\tau=\bar{T}+(r-1)T}^{\bar{T}+rT-1} \mathcal{N}_i[\tau]\} \cap \{\bigcup_{q=0}^{(r-1)} \mathcal{D}_q\}| \geq 3f + 1\}. \quad (55)$$

Let us first consider the case when $r = 1$, and accordingly focus on the interval $[\bar{T}, \bar{T} + T)$. Fix $\tau \in [0, T - 1]$, and suppose a regular non-source node $i \in \mathcal{D}_1 \cap \mathcal{R}$ hears from m regular source nodes over the interval $[\bar{T}, \bar{T} + \tau]$. We then claim that at least m components of $\mathbf{d}_i[\bar{T} + \tau]$ are at most τ . This is immediate when $m = 1$ based on (49). In particular, whenever a regular source node transmits to node i , given that it always reports a freshness-index of 0, at least one of the entries of $\mathbf{d}_i$ will take on a value of 0 at that instant. Now suppose $m = 2$, and let node i hear from regular source nodes v_1 and v_2 at time-steps $\bar{T} + \tau_1$ and $\bar{T} + \tau_2$, respectively, where $0 \leq \tau_1 \leq \tau_2 \leq \tau$. Given that node i hears from v_1 at $\bar{T} + \tau_1$, based on (49), it must be that at least one component of $\mathbf{d}_i[\bar{T} + \tau_2]$ is at most $\tau_2 - \tau_1 \leq \tau_2$. Given this, the fact that node i hears from v_2 at $\bar{T} + \tau_2$, and noting that node i appends/retains those $2f + 1$ nodes in $\mathcal{M}_i[\bar{T} + \tau_2]$ that have the lowest freshness-indices (see line 16 of Algo. 2), observe that at the end of time-step $\bar{T} + \tau_2$, at least two components of $\mathbf{d}_i[\bar{T} + \tau_2]$ are at most τ_2 ; from (49), it follows that each of these components in $\mathbf{d}_i[\bar{T} + \tau]$ are at most τ . It is easy to see that the above argument can be generalized for $m \geq 2$. Now based on joint strong $(3f + 1)$ -robustness w.r.t. $\mathcal{S}$, $\mathcal{D}_1 \neq \emptyset$ whenever $\mathcal{V} \setminus \mathcal{D}_0 \neq \emptyset$. Since $|\mathcal{A}| \leq f$, any node $i \in \mathcal{D}_1 \cap \mathcal{R}$ is guaranteed to hear from $m = 2f + 1$ regular source nodes over the interval $[\bar{T}, \bar{T} + T - 1]$. The above discussion then implies that each component of $\mathbf{d}_i[\bar{T} + T - 1]$ is at most $T - 1$. It then follows from (20) that for each $i \in \mathcal{D}_1 \cap \mathcal{R}$, $\tau_i[\bar{T} + T] \leq T$, and hence $\tau_i[2\bar{T}] \leq \bar{T}$ based on (50). To establish that $\tau_i[2\bar{T}] \leq \bar{T}, \forall i \in \mathcal{R}$, one can proceed via induction to establish that $\tau_i[\bar{T} + rT] \leq rT, \forall i \in \mathcal{D}_r \cap \mathcal{R}$. This can be done using arguments similar to when $r = 1$, and hence we omit them. The rest of the proof can be completed as in Lemma 4. $\square$

The next lemma reveals the implication of $\tau_i[k]$ taking on a finite value, based on Algo. 2. To state the result, let us define $\Omega^{(r)}[k] \triangleq \{a^r \hat{x}_s[k - r] : s \in \mathcal{S} \cap \mathcal{R}\}$, where $r \in \mathbb{N}$. Let $\text{Conv}(\Omega)$ represent the convex hull of a finite set of points Ω .

Lemma 7. *Suppose that at any $k \in \mathbb{N}_+$, $\tau_i[k] = m$ for some $i \in \{\mathcal{V} \setminus \mathcal{S}\} \cap \mathcal{R}$, where $m \in \mathbb{N}_+$. Then, Algo. 2 implies:*

$$\hat{x}_i[k] \in \text{Conv}\left(\bigcup_{r=1}^m \Omega^{(r)}[k]\right). \quad (56)$$

$\square$

Proof. We will prove the result via induction on m . For the base case of induction with $m = 1$, suppose that $\tau_i[k] = 1$ for some node $i \in \{\mathcal{V} \setminus \mathcal{S}\} \cap \mathcal{R}$ at some time-step k . Based on (20), it must then be the case that $\max_{l \in \mathcal{M}_i[k-1]} d_{i,l}[k - 1] = 0$. Since each entry of $\mathbf{d}_i[k - 1]$ is non-negative by definition, we have $d_{i,l}[k - 1] = 0, \forall l \in \mathcal{M}_i[k - 1]$. Since $\mathcal{M}_i[k - 1]$ consists of $2f + 1$ distinct nodes, and $|\mathcal{A}| \leq f$, at least $f + 1$ entries in $\mathcal{M}_i[k - 1]$ correspond to regular nodes, each of which

are source nodes (all regular non-source nodes have freshness-indices strictly larger than 1). Thus, node i must have appended at least $f + 1$ regular source nodes at time $k - 1$. For each $l \in \mathcal{M}_i[k-1] \cap \{\mathcal{S} \cap \mathcal{R}\}$, observe that $\bar{x}_{i,l}[k-1] = \hat{x}_l[k-1]$. Given the fact that $|\mathcal{A}| \leq f$, it is easy to see that the following holds based on the filtering operation in line 12 of Algo. 2:

$$\bar{x}_i[k-1] \in \text{Conv}(\{\bar{x}_{i,l}[k-1] : l \in \mathcal{M}_i[k-1] \cap \mathcal{R}\}). \quad (57)$$

Based on the above discussion, we then have that $\bar{x}_i[k-1] \in \text{Conv}(\Omega^{(0)}[k-1])$, and hence $\hat{x}_i[k] \in \text{Conv}(\Omega^{(1)}[k])$, since $\hat{x}_i[k] = a\bar{x}_i[k-1]$. This completes the proof of the base case.

To generalize the above result, let us first observe the following identity which holds for any $l \in \mathcal{M}_i[k] \cap \mathcal{R}$, and follows directly from the rules of Algo. 2:¹⁴

$$\tau_l[\phi_{i,l}[k]] + (k - \phi_{i,l}[k]) = d_{i,l}[k]. \quad (58)$$

Now fix an integer $q \geq 2$, and suppose the inclusion in (56) holds for all $m \in \{1, \dots, q-1\}$. Suppose $\tau_i[k] = q$ for some $i \in \{\mathcal{V} \setminus \mathcal{S}\} \cap \mathcal{R}$ at some time-step k . Then, based on (20), we have $\max_{l \in \mathcal{M}_i[k-1]} d_{i,l}[k-1] = q-1$. Combining this with (58), we obtain the following for each $l \in \mathcal{M}_i[k-1] \cap \mathcal{R}$:

$$\tau_l[\phi_{i,l}[k-1]] + ((k-1) - \phi_{i,l}[k-1]) \leq q-1. \quad (59)$$

Fix $l \in \mathcal{M}_i[k-1] \cap \mathcal{R}$. The above relation can be now exploited to make certain inferences about the estimate of node l at the time-step $\phi_{i,l}[k-1]$ when it was last appended by node i . To see this, consider the case when l is a non-source node. Since $((k-1) - \phi_{i,l}[k-1])$ is non-negative, $\tau_l[\phi_{i,l}[k-1]] \leq q-1$ based on (59). The induction hypothesis thus applies to node l , and we have:

$$\hat{x}_l[\phi_{i,l}[k-1]] \in \text{Conv} \left(\bigcup_{r=1}^{\tau_l[\phi_{i,l}[k-1]]} \Omega^{(r)}[\phi_{i,l}[k-1]] \right). \quad (60)$$

On the other hand, if l is a source node, then clearly $\hat{x}_l[\phi_{i,l}[k-1]] \in \text{Conv}(\Omega^{(0)}[\phi_{i,l}[k-1]])$. Combining the two cases above, and noting that $\bar{x}_{i,l}[k-1] = a^{g_{i,l}[k-1]} \hat{x}_l[\phi_{i,l}[k-1]]$ based on (21), where $g_{i,l}[k-1] = ((k-1) - \phi_{i,l}[k-1])$, we have:

$$\begin{aligned} \bar{x}_{i,l}[k-1] &\in \text{Conv} \left(\bigcup_{r=g_{i,l}[k-1]}^{g_{i,l}[k-1] + \tau_l[\phi_{i,l}[k-1]]} \Omega^{(r)}[k-1] \right) \\ &\in \text{Conv} \left(\bigcup_{r=0}^{q-1} \Omega^{(r)}[k-1] \right), \end{aligned} \quad (61)$$

where the last inclusion follows by noting that $g_{i,l}[k-1] \geq 0$, and that $g_{i,l}[k-1] + \tau_l[\phi_{i,l}[k-1]] \leq q-1$ based on (59). Appealing to (57), and using (61), we conclude that

$$\bar{x}_i[k-1] \in \text{Conv} \left(\bigcup_{r=0}^{q-1} \Omega^{(r)}[k-1] \right). \quad (62)$$

¹⁴Essentially, (58) suggests that the difference between node i 's internal copy of node l 's freshness-index at time k , namely $d_{i,l}[k]$, and the actual freshness-index $\tau_l[\phi_{i,l}[k]]$ of node l when it was last appended by node i , is simply the time that has elapsed since node l was last appended by node i , namely $(k - \phi_{i,l}[k])$: this observation follows directly from (17).

Since $\hat{x}_i[k] = a\bar{x}_i[k-1]$, we then have

$$\hat{x}_i[k] \in \text{Conv} \left(\bigcup_{r=1}^q \Omega^{(r)}[k] \right), \quad (63)$$

which is precisely the desired conclusion. $\square$

We are now in position to prove Theorem 2.

Proof. (Theorem 2) Let us begin by noting that given any desired convergence rate $\rho \in (0, 1)$, each node $i \in \mathcal{S} \cap \mathcal{R}$ can choose its observer gain l_i to guarantee $|e_i[k]| \leq \alpha \rho^k$, $\forall k \in \mathbb{N}$, based on the observer (15). Here, $e_i[k] = \hat{x}_i[k] - x[k]$, and α is some suitable constant. Now consider any $i \in \{\mathcal{V} \setminus \mathcal{S}\} \cap \mathcal{R}$, and suppose $k \geq (N - |\mathcal{S}|)T$. From Lemma 6, we know that $\tau_i[k] \neq \omega$, and hence $\tau_i[k] \in \mathbb{N}_+$. Based on Lemma 7, we conclude that there exist non-negative weights $w_{is}^{(r)}[k]$, such that $\sum_{s \in \mathcal{S} \cap \mathcal{R}} \sum_{r=1}^{\tau_i[k]} w_{is}^{(r)}[k] = 1$, and

$$\hat{x}_i[k] = \sum_{s \in \mathcal{S} \cap \mathcal{R}} \sum_{r=1}^{\tau_i[k]} w_{is}^{(r)}[k] a^r \hat{x}_s[k-r]. \quad (64)$$

Based on the convexity of the weights $w_{is}^{(r)}[k]$, note that $x[k] = \sum_{s \in \mathcal{S} \cap \mathcal{R}} \sum_{r=1}^{\tau_i[k]} w_{is}^{(r)}[k] a^r x[k-r]$. Using this and (64) yields:

$$e_i[k] = \sum_{s \in \mathcal{S} \cap \mathcal{R}} \sum_{r=1}^{\tau_i[k]} w_{is}^{(r)}[k] a^r e_s[k-r]. \quad (65)$$

Taking norms on both sides of the above equation, we obtain:

$$|e_i[k]| \leq \alpha \sum_{s \in \mathcal{S} \cap \mathcal{R}} \sum_{r=1}^{\tau_i[k]} w_{is}^{(r)}[k] \left(\frac{|a|}{\rho} \right)^r \rho^k \quad (66)$$

$$\leq \alpha \left(\frac{|a|}{\rho} \right)^{\tau_i[k]} \rho^k \leq \alpha \left(\frac{|a|}{\rho} \right)^{2(N-|\mathcal{S}|)T} \rho^k. \quad (67)$$

In the above steps, we have assumed $|a| \geq 1$ (to avoid trivialities) and exploited the convexity of the weights $w_{is}^{(r)}[k]$ to arrive at the second inequality, and used (52) to arrive at the final inequality. This concludes the proof. $\square$

REFERENCES

- [1] C.-T. Chen, *Linear system theory and design*. Oxford University Press, Inc., 1998.
- [2] U. Khan, S. Kar, A. Jadbabaie, and J. M. Moura, "On connectivity, observability, and stability in distributed estimation," in *Proc. of the 49th IEEE Conference on Decision and Control*, 2010, pp. 6639–6644.
- [3] V. Ugrinovskii, "Conditions for detectability in distributed consensus-based observer networks," *IEEE Trans. on Autom. Control*, vol. 58, no. 10, pp. 2659–2664, 2013.
- [4] T. Kim, H. Shim, and D. D. Cho, "Distributed luenberger observer design," in *Proc. of the 55th IEEE Decision and Control Conference*, 2016, pp. 6928–6933.
- [5] S. Park and N. C. Martins, "Design of distributed LTI observers for state omniscience," *IEEE Trans. on Autom. Control*, vol. 62, no. 2, pp. 561–576, 2017.
- [6] A. Mitra and S. Sundaram, "Distributed observers for LTI systems," *IEEE Trans. on Autom. Control*, vol. 63, no. 11, pp. 3689–3704, 2018.
- [7] L. Wang and A. S. Morse, "A distributed observer for a time-invariant linear system," *IEEE Trans. on Autom. Control*, vol. 63, no. 7, 2018.
- [8] W. Han, H. L. Trentelman, Z. Wang, and Y. Shen, "A simple approach to distributed observer design for linear systems," *IEEE Trans. on Autom. Control*, vol. 64, no. 1, pp. 329–336, 2019.
- [9] F. F. Rego, A. P. Aguiar, A. M. Pascoal, and C. N. Jones, "A design method for distributed Luenberger observers," in *Proc. of the 56th IEEE Conference on Decision and Control*, 2017, pp. 3374 – 3379.

- [10] Á. R. del Nozal, P. Millán, L. Orihuela, A. Seuret, and L. Zaccarian, "Distributed estimation based on multi-hop subspace decomposition," *Automatica*, vol. 99, pp. 213–220, 2019.
- [11] T. Kim, C. Lee, and H. Shim, "Completely decentralized design of distributed observer for linear systems," *IEEE Trans. on Autom. Control*, 2019.
- [12] L. Wang, A. Morse, D. Fullmer, and J. Liu, "A hybrid observer for a distributed linear system with a changing neighbor graph," in *Proc. of the 56th IEEE Conf. on Decision and Control*, 2017, pp. 1024–1029.
- [13] S. Wang and W. Ren, "On the convergence conditions of distributed dynamic state estimation using sensor networks: A unified framework," *IEEE Trans. on Cont. Sys. Tech.*, vol. 26, no. 4, pp. 1300–1316, 2018.
- [14] K. M. Lynch, I. B. Schwartz, P. Yang, and R. A. Freeman, "Decentralized environmental modeling by mobile sensor networks," *IEEE transactions on robotics*, vol. 24, no. 3, pp. 710–724, 2008.
- [15] R. Graham and J. Cortés, "Adaptive information collection by robotic sensor networks for spatial estimation," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1404–1419, 2011.
- [16] W. Han, H. L. Trentelman, Z. Wang, and Y. Shen, "Towards a minimal order distributed observer for linear systems," *Systems & Control Letters*, vol. 114, pp. 59–65, 2018.
- [17] A. Mitra, J. A. Richards, S. Bagchi, and S. Sundaram, "Resilient distributed state estimation with mobile agents: overcoming Byzantine adversaries, communication losses, and intermittent measurements," *Autonomous Robots*, vol. 43, no. 3, pp. 743–768, 2019.
- [18] S. Mou, J. Liu, and A. S. Morse, "A distributed algorithm for solving a linear algebraic equation," *IEEE Trans. on Autom. Control*, vol. 60, no. 11, pp. 2863–2878, 2015.
- [19] A. Jadbabaie, J. Lin, and A. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. on Autom. Control*, vol. 48, no. 6, pp. 988–1001, 2003.
- [20] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. on Autom. Control*, vol. 54, no. 1, p. 48, 2009.
- [21] H. Lin and P. J. Antsaklis, "Stability and stabilizability of switched linear systems: a survey of recent results," *IEEE Trans. on Autom. control*, vol. 54, no. 2, pp. 308–322, 2009.
- [22] M. Deghat, V. Ugrinovskii, I. Shames, and C. Langbort, "Detection and mitigation of biasing attacks on distributed estimation networks," *Automatica*, vol. 99, pp. 369–381, 2019.
- [23] J. Kim, J. G. Lee, C. Lee, H. Shim, and J. H. Seo, "Local identification of sensor attack and distributed resilient state estimation for linear systems," in *Proc. of the 57th IEEE Conference on Decision and Control*, 2018, pp. 2056–2061.
- [24] X. He, X. Ren, H. Sandberg, and K. H. Johansson, "Secure distributed filtering for unstable dynamics under compromised observations," *arXiv:1903.07345*, 2019.
- [25] A. Mitra and S. Sundaram, "Byzantine-resilient distributed observers for LTI systems," *Automatica*, vol. 108, p. 108487, 2019.
- [26] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *IEEE INFOCOM*, 2012, pp. 2731–2735.
- [27] M. Costa, M. Codreanu, and A. Ephremides, "On the age of information in status update systems with packet management," *IEEE Transactions on Information Theory*, vol. 62, no. 4, pp. 1897–1910, 2016.
- [28] R. Talak, S. Karaman, and E. Modiano, "Minimizing age-of-information in multi-hop wireless networks," in *Proc. Annual Allerton Conf. on Comm., Control, and Computing*, 2017, pp. 486–493.
- [29] A. Mitra, J. A. Richards, S. Bagchi, and S. Sundaram, "Finite-time distributed state estimation over time-varying graphs: Exploiting the age-of-information," in *Proc. of the American Control Conference*. IEEE, 2019, pp. 4006–4011.
- [30] D. Dolev, N. A. Lynch, S. S. Pinter, E. W. Stark, and W. E. Weihl, "Reaching approximate agreement in the presence of faults," *Journal of the ACM (JACM)*, vol. 33, no. 3, pp. 499–516, 1986.
- [31] H. J. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram, "Resilient asymptotic consensus in robust networks," *IEEE Journal on Selected Areas in Comm.*, vol. 31, no. 4, pp. 766–781, 2013.
- [32] N. H. Vaidya, L. Tseng, and G. Liang, "Iterative approximate Byzantine consensus in arbitrary directed graphs," in *Proc. of the ACM Symp. on Principles of Distributed Comp.*, 2012, pp. 365–374.
- [33] A. Tahbaz-Salehi and A. Jadbabaie, "A necessary and sufficient condition for consensus over random networks," *IEEE Trans. on Autom. Control*, vol. 53, no. 3, pp. 791–795, 2008.
- [34] R. A. Horn, R. A. Horn, and C. R. Johnson, *Matrix analysis*. Cambridge university press, 1990.



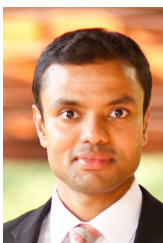
Aritra Mitra is a Postdoctoral Researcher in the Department of Electrical and Systems Engineering, University of Pennsylvania. He received the Ph.D. degree from Purdue University, USA, the M.Tech. degree from the Indian Institute of Technology Kanpur, India, and the B.E. degree from Jadavpur University, Kolkata, India, in 2020, 2015, and 2013, respectively, all in Electrical Engineering. His current research interests include distributed learning and optimization, statistical signal processing, 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.



John A. Richards is a Distinguished Member of Technical Staff in the Autonomy for Hypersonics Department of Sandia National Laboratories. Dr. Richards has led numerous projects involving sensor exploitation, autonomy, and signal and image processing applications. He is widely recognized as a leading figure in the field of automatic target recognition (ATR) for synthetic aperture radar (SAR) and high range-resolution radar (HRR). Dr. Richards is the author of numerous conference and journal papers, including the article on SAR in the Encyclopedia of Optical Engineering (Taylor & Francis Press, 2015). He is a member of the Executive Committee of the Automatic Target Recognition Working Group (ATRWG), a consortium of developers, sponsors, and users of sensor exploitation systems. Dr. Richards received his Ph.D. in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology (MIT) in 2001. He previously received his S.B. and M.Eng. in Electrical Engineering, also from MIT, in 1996.



Saurabh Bagchi is a Professor in the School of Electrical and Computer Engineering and the Department of Computer Science at Purdue University in West Lafayette, Indiana. He is the founding Director of a university-wide resiliency center at Purdue called CRISP (2017-present). He serves on the IEEE Computer Society Board of Governors for the 2017–19 term. Saurabh's research interest is in distributed systems and dependable computing. He is proudest of the 20 PhD students who have graduated from his research group and who are in various stages of building wonderful careers in industry or academia. In his group, he and his students have far too much fun building and breaking real systems. Saurabh received his MS and PhD degrees from the University of Illinois at Urbana-Champaign and his BS degree from the Indian Institute of Technology Kharagpur, all in Computer Science.



Shreyas Sundaram is an Associate Professor in the School of Electrical and Computer Engineering at Purdue University. He received his MS and PhD 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.