Automatica 157 (2023) 111264

Contents lists available at ScienceDirect

Automatica

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

Survey paper

Graph-theoretic approaches for analyzing the resilience of distributed control systems: A tutorial and survey☆

Mohammad Pirani [a,*], Aritra Mitra [b], Shreyas Sundaram [c]

[a] Department of Mechanical Engineering, University of Ottawa, Ottawa, ON, Canada
[b] Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, USA
[c] Elmore Family School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA

ARTICLE INFO

Article history:
Received 25 May 2022
Received in revised form 15 April 2023
Accepted 20 July 2023
Available online 5 September 2023

Keywords:
Distributed control systems
Graph theory
Resilient distributed algorithms

ABSTRACT

As the scale of distributed control systems over networks increases and interactions between different subsystems become more sophisticated, questions of the resilience of such networks to attacks and faults increase in importance. The need to adapt (and redefine) classical system and control-theoretic notions to answer such questions using the language of graphs has recently started to gain attention as a fertile and important area of research. This paper presents an overview of graph-theoretic methods for analyzing the resilience of distributed control systems. We survey distributed algorithms that have been proposed to solve a variety of problems in networked systems, and summarize their resilience against adversarial actions in the context of the structure of the underlying networks. We also discuss graph-theoretic methods to quantify the attack impact, and reinterpret some system-theoretic notions of robustness from a graph-theoretic standpoint to mitigate the impact of the attacks. We conclude by introducing some avenues for further research in this field.

1. Introduction

Distributed control systems (DCS) are systems consisting of multiple autonomous units (also referred to as agents, nodes, or components) that are spatially distributed and communicate with each other to perform tasks such as monitoring, estimation, learning, and control with no central operator. Depending on the specific application and deployed technologies, the nodes can represent sensors, controllers, and actuators (or combinations of the above elements into larger platforms), and the communication between nodes can be through wired or wireless communication channels. An example of a DCS is shown in Fig. 1. DCSs are often also highly dynamic, with subsystems, actuators, sensors and communication channels which turn on and off over time, or with the network topology changing due to mobility of the nodes (as would be the case where the nodes represent platforms such as robots or vehicles).

As opposed to monolithic systems where a single decision-maker (human or machine) possesses all available knowledge and information related to the system, each node in a DCS typically has access to information that is not available to other nodes. Coordination among the nodes in such systems can be achieved through the use of distributed algorithms. These algorithms are executed concurrently by each of the nodes, incorporating both their local information and any information received from other nodes in the network (via the communication channels). However, as the scale and complexity of interconnections in DCSs

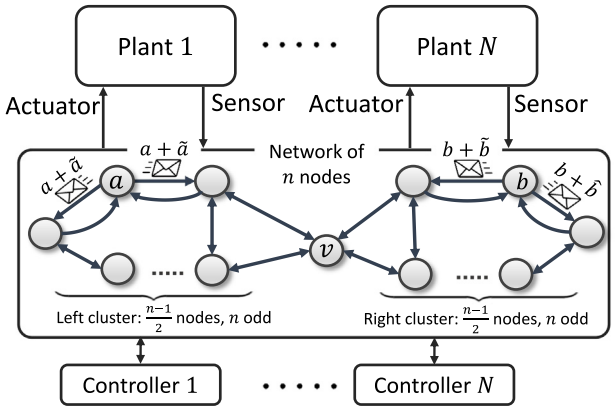


Fig. 1. A schematic figure of a distributed control system.

☆ This material is based in part upon work supported by NSF CAREER award, USA 1653648. This paper was recommended for publication in revised form by Associate Editor Sonia Martinez.

* Corresponding author.
E-mail addresses: mpirani@uwaterloo.ca (M. Pirani), amitra20@seas.upenn.edu (A. Mitra), sundara2@purdue.edu (S. Sundaram).

https://doi.org/10.1016/j.automatica.2023.111264
0005-1098/

increase, these distributed algorithms become more prone to failures, degradation, and attacks. In this paper, our focus is specifically on attacks carried out by adversarial agents in DCSs. A crucial distinction between a fault and an attack is that in the latter, the attacker exploits their knowledge of the system model to target the most vulnerable parts of a DCS. The aim may be to maximize the impact of the attack, minimize its detectability, or minimize the effort required to carry out the attack. As the attacker intelligently optimizes its actions, distributed algorithms have to be carefully designed to withstand adversarial actions, rather than more generic classes of faults considered by classical fault-tolerant control methods. We will use the term "resilient" to describe distributed algorithms that are able to mitigate such sophisticated and targeted misbehavior by certain nodes in the network, in keeping with the terminology from the classical computer science literature where such algorithms were first studied (Lynch, 1997). Since our focus is on attacks, we will also use the term "secure" interchangeably with "resilient" in this survey. Among various approaches to the resilience of distributed algorithms, the goal of this survey paper is to focus specifically on tools and analysis at the intersection of systems and control theory, graph theory, and communication and computation techniques. Notably, addressing the resilience of distributed control systems is a complex and multifaceted challenge that requires a synergistic integration of several disciplines and tool sets, some of which are discussed in this paper.

1.1. Resilience of distributed systems

As mentioned above, the main difference between adversarial actions and faults stems from the ability of the attacker to carefully target vulnerable parts of the system, potentially by learning about the system (and the deployed algorithms) before the attack. Attacks are thus classified into different categories based on their knowledge level and their ability to disrupt resources. In addition to system-theoretic properties, the structure of the underlying network plays a key role in determining the outcomes of specific attacks. The following simple example shows the role of the graph structure in distinguishing attacks from faults.

Example 1. Consider the DCS shown in Fig. 1. If the network experiences a single fault distributed uniformly at random over set of n nodes in the network, then any given node fails with probability $p = \frac{1}{n}$. In particular, when n is large, the probability that the node labeled v fails under this model is small. On the other hand, an attacker that wishes to disconnect the graph can do so simply by targeting agent v. This prevents nodes in the left cluster from receiving information from nodes in the right cluster. This example shows how targeted attacks can lead to significantly larger impacts than random faults in networked systems.

The dichotomy between random failures and targeted node removal has been studied in the complex networks literature, particularly in the case of scale-free networks (Albert, Jeong, & Barabási, 2000). Scale-free networks are those in which the degree distribution follows a power law, meaning that a few "hub" nodes have a very high number of connections while the majority of nodes have relatively few connections. This results in a network that is resilient to random failures but vulnerable to targeted attacks on the hubs. Such networks can be found in many real-world systems, such as the internet, social networks, and biological systems, and substantial research has been dedicated to studying them across disciplines including computer science, biology, and sociology (Albert et al., 2000; Barabasi & Albert, 1999; Newman, 2001).

Other than simply removing agents from the network as discussed in Example 1, the attacker can perform more complex actions. One such action is to manipulate the dynamics of a subset of agents in the network by injecting carefully crafted attack signals (or incorrect data) into their dynamics. These attacks create a discrepancy between the information that the targeted agents send to their neighbors and the true information they are supposed to send. Here, based on the communication medium, another network-theoretic feature of attacks arises which distinguishes the severity of the attacker: if the communication is point-to-point (as opposed to wireless broadcast), the attacker has the ability to send incorrect *and* inconsistent information to different neighbors; however, for broadcast models, attackers can transmit arbitrary information, but such information is consistent across neighbors. We formally distinguish between these two kinds of attacks (termed "Byzantine" and "malicious" attacks, respectively) in Definition 1 of Section 3 later in the paper. An example is shown in Fig. 1 where the agent whose true value is a sends a wrong but consistent value of $a + \tilde{a}$ to its neighbors, whereas the agent with value b shares wrong and inconsistent information to neighbors, i.e., it sends $b+\tilde{b}$ to one agent and $b+\hat{b}$ to the other.

1.2. Applications

Distributed control systems have found numerous applications in today's engineering systems, such as intelligent transportation systems, smart buildings, and power systems (Bemporad, Heemels, & Johansson, 2010). Some of these applications are briefly discussed below.

Automotive and intelligent transportation systems

The concept of connected vehicles, denoted by V2X, effectively transforms transportation systems into a network of processors. From this perspective, V2X refers to (i) each vehicle's wireless communications with its surroundings, including other vehicles, road infrastructure, and the cloud, and (ii) wired communication within each vehicle between several electronic control units (ECU) in a controller area network (CAN). At the higher level, the agents represent vehicles, road infrastructure, the cloud, and any other component that is able to send information. The wireless communication between those agents is modeled by edges, e.g., the dedicated short-range communication system (DSRC) for vehicle-to-vehicle (V2V) communication. For the wired intra-vehicle network, the agents are ECUs and the edges are buses transmitting data. Wireless communications between vehicles and their surroundings are prone to intrusions. Several works have reported different types of attacks on inter-vehicular networks, along with defense mechanisms (Biron, Dey, & Pisu, 2018; Mokdad, Ben-Othman, & Nguyen, 2015). An example of an attack on vehicular networks is the "Sybil attack", which involves an attacker creating multiple fake identities in a network to gain control or influence over it. In the context of vehicular networks, a Sybil attack could involve an attacker creating virtual vehicles in the network, making it appear as though there are more vehicles on the road than there actually are. Attacks on the wired intra-vehicle network or on safety critical ECUs (e.g., engine control unit, active steering, or brake system) can have life-threatening consequences. Moreover, stealthy attacks on the CAN bus system, including an attack that embeds malicious code in a car's telematics unit and completely erases any evidence of its presence after a crash, have been reported (Koscher et al., 2010). Several other attacks using both wired and wireless communications have been studied in the literature. An example is an attack that can enter the vehicle via a Bluetooth connection through the radio ECU and be disseminated to other safety-critical ECUs (Checkoway et al., 2011). This shows that intra-vehicle network structure through the CAN bus can determine the security of critical ECUs.

Smart buildings and internet of things

Smart buildings are the integration of a vast number of sensors, smart devices, and appliances to control heating, ventilation and air conditioning, lighting, and home security systems through a *building automation system* (BAS). In a building automation networked system, the home appliances are the agents and the wireless communications between them and between each appliance and the center are the edges. When home devices are connected to the internet, they form a key part of the *Internet of Things* (IoT). The objectives of building automation are to improve occupant comfort, ensure efficient operation of building systems, and reduce energy consumption and operating costs. However, the high level of connectivity, automation, and remote accessibility of devices also makes it critical to protect smart buildings against failures and attacks (Stankovic, 2014).

Power systems

The traditional practice in power grids is to institute safeguards against physical faults using protective devices (Hooshyar & Iravani, 2017). However, the emergence of new technologies including smart meters, smart appliances, and renewable energy resources, together with available communication technologies introduces further vulnerabilities to potential cyber-attacks (Sridhar, Hahn, & Govindarasu, 2011). Cyber-attacks in power systems can happen at three different levels: (i) *Generation and transmission level*: Generation is the process of converting mechanical or chemical energy into electrical energy and transmission is the process of transporting electrical energy over long distances from the point of generation to the point of consumption. Examples of security issues in these levels include possible cyberattacks in automatic generation control (AGC) loops and ways to detect those attacks (Huang, Satchidanandan, Kumar, & Xie, 2018; Teixeira, Amin, Sandberg, Johansson, & Sastry, 2010). (ii) *Distribution level:* Power distribution is the final step in the delivery of electrical energy consisting of delivering electricity from high-voltage transmission lines to homes, businesses, and other end users at low voltages. At the distribution level, security of islanded micro-grids has been well studied in the literature (Li, Shahidehpour, & Aminifar, 2017). Islanded micro-grids are self-contained power systems that can operate independently of the larger power grid and operate as a separate power distribution system. (iii) *Market level:* This is the stage of the electricity supply chain where energy is traded and prices are determined. An example of security issues in this level is false data injection in electricity markets (Xie, Mo, & Sinopoli, 2011). Further discussion on the cyber-security of power systems, including some surveys on the topic, can be found in Ericsson (2010), Liang, Zhao, Luo, Weller, and Yang Dong (2017), Musleh, Chen, and Yang Dong (2020) and Ten, Liu, and Manimaran (2008).

Blockchain

A blockchain is a growing list of records, called blocks, that are linked together using cryptography. Each block contains a cryptographic hash of the previous block in a tree structure, called a Merkle tree. As each block contains information about the block previous to it, they form a chain, with each additional block reinforcing the ones before it. Blockchains are considered secure by design, because they utilize several key cryptographic and decentralized mechanisms to ensure the integrity and security of the data stored on the chain. Hence, they exemplify a distributed computing system with high attack tolerance (Lin & Liao, 2017). One of the most recognized applications of blockchains is in cryptocurrency, e.g., bitcoins.

Other applications

There are several other applications for which the security of large-scale distributed systems plays a crucial role. Examples include (1) cloud computing (as a popular application of distributed systems) which enables users to access computing resources such as servers, storage, and applications over the internet; (2) online social networks such as Facebook, Twitter, and LinkedIn which allow users to connect with each other and share information across the network; (3) swarm robotics (with applications ranging from search and rescue missions to mining and agricultural systems); and (4) water and waste-water networks. Further discussion of these applications can be found in Abdelrazek, Grundy, and Müller (2010), Dibaji et al. (2019), Falliere, Murchu, and Chien (2013), Hamann (2018), Jackson (2010), Kushleyev, Mellinger, Powers, and Kumar (2013), Marsden and Friedkin (1993), Slay and Miller (2007) and Zhang, Sun, Zhu, and Fang (2010). Yet another instance of a distributed system where security plays a key role is the emerging paradigm of federated learning (FL). The general principle of FL is to train local models (e.g., the weights of a deep neural network) on raw local data samples. These local models are then intermittently uploaded to a central aggregator for the generation of a global model (Jiang, Balu, Hegde, & Sarkar, 2017; Konečný, McMahan, Ramage, & Richtárik, 2016). Importantly, in FL, only model/parameter vectors are exchanged; to guarantee privacy, these models are often encrypted before being uploaded. For the cases where these parameters may still leak information about the underlying data samples, e.g., by making multiple specific queries on specific datasets, secure aggregation techniques have been developed (Ghosh, Hong, Yin, & Ramchandran, 2019; Muñoz-González, Co, & Lupu, 2019; Pillutla, Kakade, & Harchaoui, 2022; Yang, Liu, Chen, & Tong, 2019).

1.3. Early works on the resilience of distributed control systems

We provide a brief literature review on the security of control systems; starting from centralized approaches and then followed up by distributed methods.

1.3.1. Centralized resilient control techniques

Centralized fault-tolerant techniques have a long history in the systems and control community (Darouach, Zasadzinski, & Xu, 1994; De Persis & Isidori, 2001; Guan & Saif, 1991; Massoumnia, Verghese, & Willsky, 1989). These early works focused on detecting and mitigating faults in control systems. Later efforts focused on the goal of providing defense mechanisms for control systems against specific types of attacks via three layers of *attack prevention, attack detection*, and *attack resilience*. We provide a brief overview of these defense layers below. Detailed discussions can be found in Dibaji et al. (2019) and Teixeira, Sou, Sandberg, and Johansson (2015).

The first layer of defense is to prevent the attack from happening. Cryptography, network coding, model randomization, differential privacy, and moving target defense are among well-known attack prevention mechanisms used for control systems (Dwork, 2008; Farokhi, Shames, & Batterham, 2017; Koetter & Médard, 2003; Motwani & Raghavan, 1996; Ny & Pappas, 2014; Weerakkody & Sinopoli, 2015). In many cases, however, it is not possible to prevent all attacks (e.g., if attackers exploit subtle "zero-day" vulnerabilities Bilge & Dumitraş, 2012, or rely on insider threats Salem, Hershkop, & Stolfo, 2008). In those cases, the second layer comes into play which aims to detect and isolate the attack. Observer-based techniques have been proposed to detect the attacks by leveraging appropriate notions of redundancy between sensors; these approaches compare the state estimates under the healthy and the attacked cases when models of possible attacks are considered (Hwang, Kim, Kim, & Seah, 2009;

Pasqualetti, Dorfler, & Bullo, 2013). When the control system does not satisfy the required observability conditions, coding-theory, e.g., parity check methods, can be used to detect the attacks (Blahut, 2003). In some cases, an adversary delivers compromised sensor measurements to a system operator to conceal its effect on the plant. Certain types of such attacks, referred to as "replay attacks", have been addressed by introducing physical watermarking (e.g., by adding a Gaussian signal to the control input) to bait the attacker to reveal itself (Mo, Weerakkody, & Sinopoli, 2015). The sub-optimality of the resulting control action is the cost paid to detect the attacks in those cases. In addition to the above model-based techniques, anomaly detection methods have been proposed based on machine learning techniques. For example, Neural Networks (NNs) and Bayesian learning have been studied for anomaly detection in the context of security (Garcia-Teodoro, Diaz-Verdejo, Macia-Fernandez, & Vazquez, 2009; He, Mendis, & Wei, 2016; Tsai, Hsu, Lin, & Lin, 2009). When attack detection is not possible, the system must be at least resilient enough to withstand the attacks or mitigate the impact of the attack. Probabilistic methods for attack resilience in DCSs for both estimation and control were studied in Fawzi, Tabuada, and Diggavi (2014), Hadjicostis and Touri (2002), Hespanha, Naghshtabrizi, and Xu (2007) and Sinopoli et al. (2004). Redundancy-based approaches are used to bypass the attacks by using the healthy redundant parts (Abbas, Laszka, & Koutsoukos, 2017; Baras & Liu, 2019; Chow & Willsky, 1984; Momani & Challa, 2010; Sklaroff, 1976). Such redundancy in large-scale systems can be in the form of adding capabilities to the network, e.g., extra sensors, or infrastructures enabling the connection between the parts, i.e., network connectivity, typically at additional cost. On the other hand, several control-theoretic methods have also been proposed to mitigate the attack impact, including event-triggered control for tackling denial of service attacks (De Persis & Tesi, 2014; Heemels, Johansson, & Tabuada, 2012). Robust control techniques have also been shown to be useful tools to mitigate the attack impact (Zhu & Basar, 2015).

1.3.2. Resilient distributed techniques

The theory of distributed algorithms has a long history in computer science with a variety of applications in telecommunications, scientific computing, distributed information processing, and real-time process control (Bertsekas & Tsitsiklis, 1989; Tsitsiklis, 1984). The control systems community has studied distributed control algorithms for several decades (Borkar & Varaiya, 1982; Sandell, Varaiya, Athans, & Safonov, 1978; Teneketzis & Varaiya, 1984; Tsitsiklis, 1984; Witsenhausen, 1968), with an explosion of interest in recent decades due to their applications in distributed coordination of multi-agent systems, formation control of mobile robots, state estimation of power-grids, smart cities, intelligent transportation systems, and distributed energy systems (Jadbabaie, Lin, & Morse, 2003; Olfati-Saber, Fax, & Murray, 2007; Ren & Beard, 2005; Tomlin, Pappas, & Sastry, 1998).

The earliest works on the security of distributed algorithms can be found in the computer science literature (Fischer, Lynch, & Paterson, 1985; Lynch, 1997; Pease, Shostak, & Lamport, 1980), typically with the focus on simple network topologies (such as complete graphs). One of the main approaches to address the resilience and security of distributed systems is to leverage the redundancy that the network topology provides (in the form of multiple paths through the network between given pairs of nodes). Hence, resilient distributed estimation and control algorithms usually use (different types of) network connectivity measures to quantify the resilience against certain adversarial actions (LeBlanc, Zhang, Koutsoukos, & Sundaram, 2013; Pasqualetti, Bicchi, & Bullo, 2012; Sundaram & Hadjicostis, 2011). To do this, system-theoretic notions, such as controllability (or observability) and detectability are reinterpreted in terms of graph-theoretic quantities with the help of tools such as algebraic graph theory or structured systems theory (Dion, Commault, & van der Woude, 2003; Pasqualetti et al., 2013).

When a DCS becomes larger in scale, the notions of attack prevention, detection, and resilience discussed above depend more on the interconnections between components (sensors, actuators, or controllers) in the network. With this in mind, the focus of this survey paper is to present the theoretical works in the literature on graph-theoretic interpretations of the security in DCSs.

Related Survey Papers. There are some recently published survey papers on related topics, including Giraldo, Sarkar, Cardenas, Maniatakos, and Kantarcioglu (2017), which provide an overview of security in a variety of cyber–physical systems (e.g., smart-grids, manufacturing systems, healthcare units, industrial control systems, etc.); Ishii, Wang, and Feng (2022), which focus on resilient consensus problems; Yang, Gang, and Bajwa (2020), which focus on distributed statistical inference and machine learning under attacks; and Prorok et al. (2021), which discuss the applications of resilient distributed algorithms to multi-robot systems. The tutorial paper (Chong, Sandberg, & Teixeira, 2019) considers security and privacy methods in control systems and Lun, D'Innocenzo, Smarra, Malavolta, and Di Benedetto (2019) analyze existing research on cyber–physical security from an automatic control perspective. Compared to Giraldo et al. (2017) where the exposition is essentially of a qualitative nature, our survey provides a mathematical treatment of security in distributed systems, covering the necessary technical background in linear algebra, graph theory, dynamical systems, and structured systems theory. Our paper differs from Ishii et al. (2022), Prorok et al. (2021), and Yang et al. (2020) in that it has a much broader scope: we provide a detailed discussion of graph-theoretic measures for the resilience of a variety of distributed algorithms, special cases of which include consensus and distributed statistical inference. Moreover, we provide a comprehensive view of the role of various connectivity measures on the resilience levels for distributed algorithms. We also discuss ways to maintain a desired level of resilience when the network loses connectivity. Our paper differs from Chong et al. (2019) and Lun et al. (2019) in that the scope of those papers was on general methods in secure control systems without focusing on the network aspects of control systems and the role that the network topology plays in the overall system's resilience. More recently, Ramos, Aguiar, and Pequito (2022) presented an overview of structured systems theory, one of the main tools that is used in analyzing the resilience of DCSs. However, rather that the system's resilience and security, the focus of the work was more on deepening the understanding of how some theoretical graph tools can be leveraged in analyzing control-theoretic notions of large-scale systems.

2. Mathematical preliminaries

2.1. Graph theory

A weighted graph is a pair $(\mathcal{G}, w)$ where $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ is a directed graph in which $\mathcal{V}$ is the set of vertices (or agents),[1] $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges and $w : \mathcal{E} \to \mathbb{R}$ is a weight function. In particular, $(j, i) \in \mathcal{E}$ if and only if there exists an edge from j to i with some weight $w_{ji} \neq 0$. Graph $\mathcal{G}$ is undirected if $(j, i) \in \mathcal{E}$ implies $(i, j) \in \mathcal{E}$ and $w_{ij} = w_{ji}$. The in-neighbors of vertex $i \in \mathcal{V}$ are denoted $\mathcal{N}_i^{\text{in}} \triangleq \{j \in \mathcal{V} \mid (j, i) \in \mathcal{E}, j \neq i\}$. Similarly, the out-neighbors of i are $\mathcal{N}_i^{\text{out}} \triangleq \{j \in \mathcal{V} \mid (i, j) \in \mathcal{E}, j \neq i\}$. The in-degree

[1] As mentioned in the introduction, we use the terms node, vertex, and agent interchangeably.

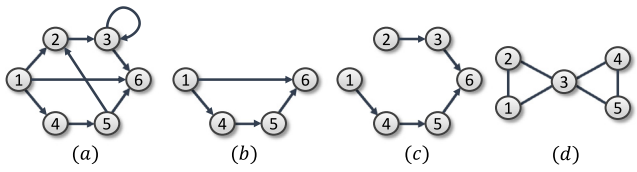


Fig. 2. (a) An example of a digraph $\mathcal{G}$, (b) an induced subgraph of $\mathcal{G}$, (c) a spanning subgraph of $\mathcal{G}$, (d) an example of a 1-vertex and 2-edge connected graph with $d_{\min} = 2$.

(or simply degree) of agent i is $d_i = \sum_j w_{ji}$. The minimum degree of a graph $\mathcal{G}$ is denoted by $d_{\min}(\mathcal{G}) = \min_{i\in\mathcal{V}} d_i$. If $(i, i) \in \mathcal{E}$, then i is said to have a self-loop (but it is not counted in the degree of i). A subgraph of $\mathcal{G}$ is a graph $\bar{\mathcal{G}} = \{\bar{\mathcal{V}}, \bar{\mathcal{E}}\}$ with $\bar{\mathcal{V}} \subseteq \mathcal{V}$ and $\bar{\mathcal{E}} \subseteq \mathcal{E}$. A subgraph is induced if it is obtained from $\mathcal{G}$ by deleting a set of vertices (and all edges coming into and out of those vertices), but leaving all other edges intact. The subgraph $\mathcal{H}$ is called spanning if it contains all vertices of $\mathcal{G}$, i.e., $\bar{\mathcal{V}} = \mathcal{V}$. An example of a digraph $\mathcal{G}$ together with an induced and a spanning subgraph of $\mathcal{G}$ is shown in Fig. 2(a–c).

2.1.1. Paths and cycles in graphs

For subsets $\mathcal{F}, \mathcal{B} \subset \mathcal{V}$, a path from $\mathcal{F}$ to $\mathcal{B}$ is a sequence of vertices $v_1, v_2, \ldots, v_t$ where $v_1 \in \mathcal{F}$, $v_t \in \mathcal{B}$, and $(v_j, v_{j+1}) \in \mathcal{E}$ for $1 \leq j \leq t-1$. A cycle is a path where $v_t = v_1$. A simple path contains no repeated vertices. A *directed acyclic graph* is a digraph with no cycles. For a subset $\mathcal{X} \subset \mathcal{V}$, an $\mathcal{X}$-rooted path (respectively $\mathcal{X}$-topped path) is a path which starts from a vertex $v \in \mathcal{X}$ (respectively ends at some $v \in \mathcal{X}$). Two paths are disjoint if they have no common vertices and two paths are internally disjoint if they have no common vertices except for possibly the starting and ending vertices. A set of paths $P_1, P_2, \ldots, P_r$ are (internally) vertex disjoint if the paths are pairwise (internally) vertex disjoint. For example, the two paths $P_1 : 1, 2, 3, 6$ and $P_2 : 1, 4, 5, 6$ in Fig. 2(a) are internally vertex disjoint paths between agents 1 and 6. Given two subsets $\mathcal{X}_1, \mathcal{X}_2 \subset \mathcal{V}$, a set of r vertex disjoint paths, each with start vertex in $\mathcal{X}_1$ and end vertex in $\mathcal{X}_2$, is called an *r-linking* from $\mathcal{X}_1$ to $\mathcal{X}_2$.[2] The length of a path is the summation of the edge weights in the path. The distance between a pair of agents i and j is the length of the shortest path between i and j. The *effective resistance*, $\mathfrak{R}_{ij}$, between two vertices i and j in a graph is the equivalent resistance between these two vertices when we treat the resistance of each edge e as $\frac{1}{w_e}$, where w_e is the edge weight.[3]

2.1.2. Graph redundancy measures

A graph $\mathcal{G}$ is called *strongly connected* if there is a path between each pair of vertices $i, j \in \mathcal{V}$. Throughout this paper, we use the term 'connected' to refer to a strongly connected graph, unless stated otherwise. A graph is said to be disconnected if there exists at least one pair of vertices $i, j \in \mathcal{V}$ such that there is no path between the two agents. Other than the above binary measures of connectivity, there are several other graph connectivity measures, some of which are mentioned below.

- **Vertex and Edge Connectivity:** A *vertex-cut* in a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ is a subset $\mathcal{S} \subset \mathcal{V}$ of vertices such that removing the vertices in $\mathcal{S}$ (and any resulting associated edges) from the graph causes the remaining graph to be disconnected. A (j, i)-*cut* in a graph is a subset $\mathcal{S}_{ij} \subset \mathcal{V}$ such that if the vertices $\mathcal{S}_{ij}$ are removed, the resulting graph contains no path from vertex j to vertex i. Let κ_{ij} denote the size of the smallest (j, i)-cut between any two vertices j and i. The graph $\mathcal{G}$ is said to have *vertex connectivity* $\kappa(\mathcal{G})$ (or to be *κ-vertex connected*) if $\kappa_{ij} \geq \kappa$ for all $i, j \in \mathcal{V}$. Similarly, the *edge connectivity* $e(\mathcal{G})$ of a graph $\mathcal{G}$ is the minimum number of edges whose deletion disconnects the graph. The vertex connectivity, edge connectivity, and minimum degree satisfy

$$\kappa(\mathcal{G}) \leq e(\mathcal{G}) \leq d_{\min}(\mathcal{G}). \tag{1}$$

  For instance, in the graph shown in Fig. 2(d), we have $\kappa(\mathcal{G}) = 1$, $e(\mathcal{G}) = 2$, and $d_{\min} = 2$
- **Graph Robustness** (LeBlanc et al., 2013; Zhang & Sundaram, 2012): For a given $r \in \mathbb{N}$, a subset S of vertices in the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is said to be *r-reachable* if there exists a vertex $i \in S$ such that $|\mathcal{N}_i^{\text{in}} \setminus S| \geq r$. Graph $\mathcal{G}$ is said to be *r-robust* if for every pair of nonempty, disjoint subsets $\mathcal{X}_1, \mathcal{X}_2 \subseteq \mathcal{V}$, either $\mathcal{X}_1$ or $\mathcal{X}_2$ is r-reachable. If $\mathcal{G}$ is r-robust, then it is at least r-vertex connected. An example of a 3-reachable together with a 4-reachable set is shown in Fig. 3(a). Note that by examining all pairs of disjoint subsets of vertices in this graph, we can find that the graph is 3- robust.
  For given $r, s \in \mathbb{N}$, a graph is said to be (r, s)-robust if for all pairs of disjoint nonempty subsets $\mathcal{X}_1, \mathcal{X}_2 \subset \mathcal{V}$, at least one of the following conditions holds:

  (i) All vertices in $\mathcal{X}_1$ have at least r neighbors outside $\mathcal{X}_1$.
  (ii) All vertices in $\mathcal{X}_2$ have at least r neighbors outside $\mathcal{X}_2$.
  (iii) There are at least s vertices in $\mathcal{X}_1 \cup \mathcal{X}_2$ that each have at least r neighbors outside their respective sets.

  Based on the above definitions, $(r, 1)$-robustness is equivalent to r-robustness.

It is instructive to note the key difference between the properties of vertex connectivity and graph robustness defined above. Specifically, if a graph is r-vertex connected, then for every pair of nonempty disjoint subsets $\mathcal{X}_1, \mathcal{X}_2 \subset \mathcal{V}$, at least one of the subsets must satisfy the property that all vertices in that subset *collectively* have at least r different neighbors outside that subset. Otherwise, one could simply remove all those neighbors from outside that subset, thereby disconnecting the graph and contradicting the fact that the vertex connectivity is r. In contrast, the graph robustness property is stronger: an r-robust graph implies that for every pair of nonempty, disjoint subsets $\mathcal{X}_1, \mathcal{X}_2 \subset \mathcal{V}$, at least one of the subsets has at least one vertex that *by itself* has at least r neighbors outside that set. As we will see later, this captures the ability of information to penetrate into sets (via the edges) through individual vertices, as opposed to penetrating into the set as a whole (as would be considered by the notion of vertex connectivity).

Based on the definitions and the above discussion, one can see that r-robustness is a stronger property than r-vertex connectivity. The Venn diagram in Fig. 3(c) shows the relationship between various graph connectivity measures. In fact, the gap between the robustness and vertex connectivity (and minimum degree) parameters can be arbitrarily large, as illustrated by the graph $\mathcal{G}$ in Fig. 3(b). While the minimum degree and vertex connectivity of the graph $\mathcal{G}$ is $n/2$, it is only 1-robust (consider subsets $\mathcal{V}_1$ and $\mathcal{V}_2$).

2.2. Matrix terminology

For a real-valued matrix $\boldsymbol{M} \in \mathbb{R}^{m\times n}$ with $n \leq m$, the singular values are ordered as $\sigma_1(\boldsymbol{M}) \leq \sigma_2(\boldsymbol{M}) \leq \cdots \leq \sigma_n(\boldsymbol{M})$. When M

[2] There are various algorithms to find linkings, such as the Ford-Fulkerson algorithm, which has run-time polynomial in the number of vertices (West, 2001a).

[3] Some variations of effective resistance are introduced for directed graphs (Young, Scardovi, & Leonard, 2016).

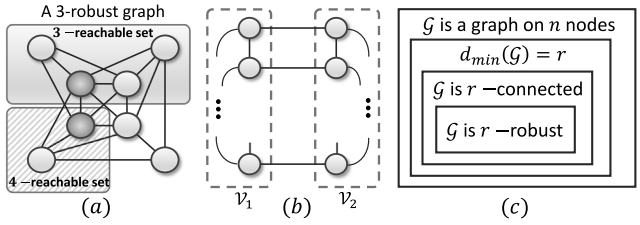


Fig. 3. (a) A 3-robust network and two of its vertex subsets, a 3-reachable set (top) and a 4-reachable set (bottom). (b) Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2$ and $|\mathcal{V}_1| = |\mathcal{V}_2| = \frac{n}{2}$. $\mathcal{V}_1$ and $\mathcal{V}_2$ are complete graphs. There is a one to one connection between vertices in $\mathcal{V}_1$ and agents in $\mathcal{V}_2$. (b) Relationships between different graph connectivity measures.

is a square matrix, the real parts of the eigenvalues are ordered as $\Re(\lambda_1(\boldsymbol{M})) \leq \Re(\lambda_2(\boldsymbol{M})) \leq \cdots \leq \Re(\lambda_n(\boldsymbol{M}))$. Matrix $\boldsymbol{M}$ is called nonnegative if its elements are nonnegative, and it is a *Metzler* matrix if its off-diagonal elements are nonnegative. We use $\mathbf{e}_i$ to indicate the ith vector of the canonical basis and A' to indicate the transpose of matrix A.

2.3. Spectral graph theory and linear systems

The adjacency matrix of a graph of n agents is denoted by $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, where $\boldsymbol{A}_{ij} = w_{ij}$ if $(j,i) \in \mathcal{E}$ with the edge weight w_{ij} and $\boldsymbol{A}_{ij} = 0$ otherwise. The Laplacian matrix of the graph is $\boldsymbol{L} \triangleq \boldsymbol{D} - \boldsymbol{A}$, where $\boldsymbol{D} = \text{diag}(d_1, d_2, \ldots, d_n)$. The real parts of the Laplacian eigenvalues are nonnegative and are denoted by $0 = \Re(\lambda_1(\boldsymbol{L})) \leq \Re(\lambda_2(\boldsymbol{L})) \leq \ldots \leq \Re(\lambda_n(\boldsymbol{L}))$.[4] The second smallest eigenvalue of the Laplacian matrix, $\lambda_2(\boldsymbol{L})$, is called the *algebraic connectivity* of the graph and is greater than zero if and only if $\mathcal{G}$ is a connected graph. Moreover, we always have (Godsil & Royle, 2001)

$$\lambda_2(\boldsymbol{L}) \leq \kappa(\mathcal{G}). \tag{2}$$

Given a connected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, an orientation of the graph $\mathcal{G}$ is defined by assigning a direction (arbitrarily) to each edge in $\mathcal{E}$. For graph $\mathcal{G}$ with m edges, labeled as $e_1, e_2, \ldots, e_m$, its agent-edge incidence matrix $\mathcal{B}(\mathcal{G}) \in \mathbb{R}^{n\times m}$ is defined as

$$[\mathcal{B}(\mathcal{G})]_{kl} = \begin{cases} 1 & \text{if agent } k \text{ is the head of edge } l, \\ -1 & \text{if agent } k \text{ is the tail of edge } l, \\ 0 & \text{otherwise.} \end{cases}$$

The graph Laplacian satisfies $\boldsymbol{L} = \mathcal{B}(\mathcal{G})\mathcal{B}(\mathcal{G})'$.

A discrete-time linear time-invariant system is represented in the state–space form as follows:

$$\begin{aligned} \boldsymbol{x}[t+1] &= \boldsymbol{W}\boldsymbol{x}[t] + \boldsymbol{B}\boldsymbol{u}[t], \\ \boldsymbol{y}[t] &= \boldsymbol{C}\boldsymbol{x}[t] + \boldsymbol{D}\boldsymbol{u}[t], \end{aligned}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the state vector, $\boldsymbol{u} \in \mathbb{R}^m$ is the vector of m inputs, $\boldsymbol{y} \in \mathbb{R}^q$ is the vector of q outputs, and $\boldsymbol{W} \in \mathbb{R}^{n\times n}$, $\boldsymbol{B} \in \mathbb{R}^{n\times m}$, $\boldsymbol{C} \in \mathbb{R}^{q\times n}$, and $\boldsymbol{D} \in \mathbb{R}^{q\times m}$, are called state, input, output, and feed-forward matrices, respectively. Similarly, the state–space model of a continuous-time linear system is given by

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{W}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}, \\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u}. \end{aligned} \tag{3}$$

A state space form of a linear system is compactly represented as $(\boldsymbol{W}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$ or $(\boldsymbol{W}, \boldsymbol{B}, \boldsymbol{C})$ for cases where there is no feed-forward term. A linear system is called (internally) positive if its state and output are non-negative for every non-negative input and every non-negative initial state. A continuous-time linear system $(\boldsymbol{W}, \boldsymbol{B}, \boldsymbol{C})$ is positive if and only if $\boldsymbol{W}$ is a Metzler matrix and $\boldsymbol{B}$ and $\boldsymbol{C}$ are non-negative element-wise (Farina & Rinaldi, 2000). Moreover, for such a positive system with transfer function $G(s) = \boldsymbol{C}(s\boldsymbol{I}_n - \boldsymbol{W})^{-1}\boldsymbol{B}$, the system $\mathcal{H}_\infty$ norm is obtained from the DC gain of the system, i.e., $\|G\|_\infty = \sigma_n(G(0))$, where σ_n is the maximum singular value of matrix $G(0)$ (Farina & Rinaldi, 2000).

2.4. Structured systems theory

In this subsection, we provide an introduction to structured systems theory, which provides a bridge between system-theoretic properties and graph-theoretic concepts. For comprehensive surveys of this topic, see Dion et al. (2003) and Ramos et al. (2022) and the references therein. Consider the linear time-invariant system (3). With this system, associate the matrices $\boldsymbol{W}_\lambda \in \{0,\lambda\}^{n\times n}$, $\boldsymbol{B}_\lambda \in \{0,\lambda\}^{n\times m}$, $\boldsymbol{C}_\lambda \in \{0,\lambda\}^{q\times n}$, and $\boldsymbol{D}_\lambda \in \{0,\lambda\}^{q\times m}$. Specifically, an entry in these matrices is zero if the corresponding entry in the system matrices is equal to zero, and the matrix entry is a free parameter (denoted by λ) otherwise. This type of representation of (3) shows the structure of the linear system regardless of the specific values of the elements in the matrices. Thus, it is called a *structured system* and can be equivalently represented by a directed graph $\mathcal{G} = \{\mathcal{X}, \mathcal{U}, \mathcal{Y}, \mathcal{E}_{\mathcal{X}\mathcal{X}}, \mathcal{E}_{\mathcal{X}\mathcal{Y}}, \mathcal{E}_{\mathcal{U}\mathcal{X}}, \mathcal{E}_{\mathcal{U}\mathcal{Y}}\}$, where

- $\mathcal{X} \triangleq \{x_1, x_2, \ldots, x_n\}$ is the set of states;
- $\mathcal{Y} \triangleq \{y_1, y_2, \ldots, y_q\}$ is the set of measurements;
- $\mathcal{U} \triangleq \{u_1, u_2, \ldots, u_m\}$ is the set of inputs;
- $\mathcal{E}_{\mathcal{X}\mathcal{X}} = \{(x_j, x_i) | \boldsymbol{W}_{ij} \neq 0\}$ is the set of edges corresponding to interconnections between the state vertices;
- $\mathcal{E}_{\mathcal{U}\mathcal{X}} = \{(u_j, x_i) | \boldsymbol{B}_{ij} \neq 0\}$ is the set of edges corresponding to connections between the input vertices and the state vertices;
- $\mathcal{E}_{\mathcal{X}\mathcal{Y}} = \{(x_j, y_i) | \boldsymbol{C}_{ij} \neq 0\}$ is the set of edges corresponding to connections between the state vertices and the output vertices;
- $\mathcal{E}_{\mathcal{U}\mathcal{Y}} = \{(u_j, y_i) | \boldsymbol{D}_{ij} \neq 0\}$ is the set of edges corresponding to connections between the input vertices and the output vertices.

A structured system is said to have a certain property, e.g., controllability or invertibility, if that property holds for at least one numerical choice of free parameters λ in the system. The following theorem introduces graphical conditions for structural controllability and observability of linear systems.

Theorem 1 (*Dion et al., 2003*)**.** *The pair $(\boldsymbol{W}, \boldsymbol{B})$ (resp. $(\boldsymbol{W}, \boldsymbol{C})$) is structurally controllable (resp. observable) if and only if the graph $\mathcal{G} = \{\mathcal{X}, \mathcal{U}, \mathcal{E}_{\mathcal{X},\mathcal{X}}, \mathcal{E}_{\mathcal{U},\mathcal{X}}\}$ satisfies both of the following properties:*

(i) Every state vertex $x_i \in \mathcal{X}$ can be reached by a path from (resp. has a path to) some input vertex (resp. some output vertex).
(ii) $\mathcal{G}$ contains a subgraph that is a disjoint union of cycles and $\mathcal{U}$-rooted paths (resp. $\mathcal{Y}$-topped paths), which covers all of the state vertices.

In Section 8, we will revisit structured systems by discussing structural conditions for the system to be invertible.

Example 2. The graph shown in Fig. 4(a) is structurally controllable as it satisfies both conditions in Theorem 1. However, it is not structurally observable: condition (ii) does not hold since the cycle and $\mathcal{Y}$-topped paths are not disjoint. The graph in Fig. 4(b) is structurally controllable and observable. The set of disjoint $\mathcal{U}$-rooted paths (respectively $\mathcal{Y}$-topped paths) and cycles is $\mathcal{P} = \big\{\{u, x_1, x_4, x_5\}, \{x_2\}, \{x_3\}\big\}$ (respectively $\mathcal{C} = \big\{\{x_1, x_4, x_5, y\}, \{x_2\}, \{x_3\}\big\}$).

[4] From now, we consider Laplacian matrices for *undirected* graphs with positive edge weights. Graph spectra for negative edge weights have been studied in Chen et al. (2016) and Hou, Li, and Pan (2003).

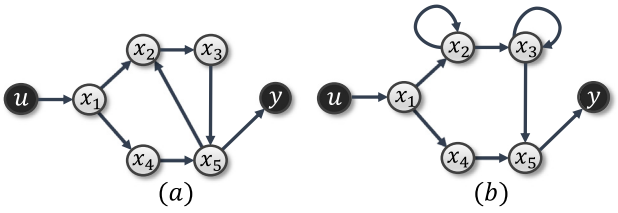


Fig. 4. (a) A structurally controllable but not observable graph, (b) A structurally controllable and observable graph.

3. Notions of resilience

In this section, we discuss notions of resilience in DCSs which are sought in various distributed algorithms. To classify the resilience against each type of adversary, the first step is to distinguish a regular agent from an adversarial one in DCSs.

In distributed algorithms, each agent is given a communication rule (indicating what information to transmit to neighbors) and an updating rule (a computation task) which is a function of its own states (and local information) and the states and information obtained from its neighbors. However, the *knowledge* available to each agent about the rest of the network may vary between algorithms. For example, as we will see later in Sections 4–7, some algorithms rely on each agent knowing the entire network topology, while others rely only on each agent knowing its own neighbors in the network. *Regular nodes* (agents) in a DCS are those who obey the prescribed updating and communication rules. The objective of the regular agents can be either to calculate an exact desired value (e.g., a function of the state of other agents), an approximation of that value, or simply to arrive at an agreement on some value. The required precision of calculation depends on many factors including the cost of computing the exact value. Deviations from normal behavior can be considered as either a fault or an adversarial action (attack). Faults are those which happen unintentionally and (often) randomly with a given distribution. On the other hand, as discussed in Section 1.1, attacks can be viewed as targeted actions that are deliberately chosen by some *adversarial* nodes to prevent the other nodes from achieving their desired objectives. From the network's perspective, the attacker (or an adversarial agent) is one who intentionally disregards the prescribed updating rule: the attacker updates its state and sends it to its neighbors in an arbitrary (and potentially in a worst case) manner. Since this type of deliberate adversarial behavior is the focus of the current paper, we further classify them in the following definition.

Definition 1 (*Malicious vs. Byzantine*)**.** An adversarial agent is called malicious if it updates its state in an arbitrary manner. Thus, it sends incorrect but consistent values to all of its out-neighbors at each time-step. An adversarial agent is Byzantine if it can update its state arbitrarily and is capable of sending inconsistent values to different neighbors at each time-step.

Both malicious and Byzantine agents are allowed to know the entire network topology, the local information of all agents, and the algorithms executed by all agents. Furthermore, both malicious and Byzantine agents are allowed to collude amongst themselves to select their actions. An example of malicious and Byzantine agents was discussed in Section 1.1 (agents with values a and b in Fig. 1). Based on Definition 1, note that Byzantine agents can choose to behave like malicious agents by simply transmitting the same value to all neighbors (even though Byzantine agents have the power to transmit inconsistent values to different neighbors). Thus, malicious behavior is a special case of Byzantine behavior. Malicious attacks may happen in wireless broadcast models of communication or when the state of an agent is directly sensed by its neighbors (e.g., via cameras), whereas Byzantine attacks follow the wired (point-to-point) model of communication.

In return for providing so much power to the adversarial agents, it is typical to assume a bound on the number of such agents. The following definitions quantify the maximum number of tolerable attacks in a given network.

Definition 2 (*f-total and f-local Sets*)**.** For $f \in \mathbb{N}$, a set $\mathcal{C} \subset \mathcal{V}$ is said to be f-total if it contains at most f agents in the network, i.e., $|\mathcal{C}| \leq f$. A set $\mathcal{C} \subset \mathcal{V}$ is f-local if it contains at most f agents in the neighborhood of each agent outside that set, i.e., $|\mathcal{N}_i^{\text{in}} \cap \mathcal{C}| \leq f$ for all $i \in \mathcal{V} \setminus \mathcal{C}$.

Definition 3 (*f-local Adversarial Model*)**.** For $f \in \mathbb{N}$, a set $\mathcal{F}$ of adversarial agents is f-locally bounded if $\mathcal{F}$ is an f-local set.

The set of adversarial agents $\{a, b\}$ in Fig. 1 is a 2-total and 1-local set. Thus, it is 1-locally bounded. Note that every f-total set is also an f-local set but not vice versa. The f-total adversarial model is predominant in the literature on resilient distributed algorithms (Bouzid, Potop-Butucaru, & Tixeuil, 2010; Lamport, Shostak, & Pease, 1982; Lynch, 1997). However, in order to allow the number of adversarial agents to potentially scale with the network, several of the algorithms discussed in this survey allow the adversarial set to be f-local.

Based on the above discussions, the notion of resilience can be stated as follows.

> **Resilient Distributed Algorithm:** Under a given adversarial model (e.g., f-locally bounded or f-total, malicious or Byzantine), a distributed algorithm operating on network $\mathcal{G}$ is called *resilient* if each regular agent in $\mathcal{G}$ can compute its desired value (within some specified tolerance) despite the actions of the adversarial agents in $\mathcal{G}$.

Thus, various specific notions of resilience can be considered based on the above definition, depending on the type and number of adversaries and the desired value computed by the regular agents.

4. Connectivity: The earliest measure of resilience

In this section, we first discuss the role of network connectivity in reliable information dissemination over networks. Then, with the help of structured systems theory (Dion et al., 2003), we tie together the traditional graph property of connectivity with system-theoretic notions to find conditions for reliable calculation of agent values in a network.

4.1. Connectivity as a measure of resilience

In the following example, we see how the existence of redundant paths between a pair of agents can facilitate reliable transmission of information between those agents.

Example 3. For the graph shown in Fig. 5(a), suppose that each agent v_i has an initial value denoted by $\psi_i[0]$. Agent v_1 tries to obtain the true value of v_4, i.e., $\psi_4[0] = 2$. This value can be transmitted to v_1 through v_2 and v_3.[5] Suppose that v_2 is malicious

[5] This can be done using a flooding algorithm, i.e., each agent reads and stores their neighbors' values and broadcasts them to their out-neighbors in the next time step.

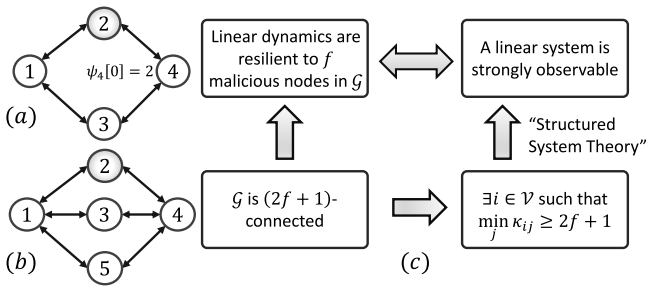


Fig. 5. (a) A graph with (a) two disjoint paths between agents v_1 and v_4, and (b) three disjoint paths between agents v_1 and v_5. (c) Schematics of interconnections between system and graph properties.

and pretends that $\psi_4[0]$ has a value other than its true value. In this case, as v_1 receives inconsistent information from v_2 and v_3, it cannot conclude which value is the true one. In this case, a redundant path can serve as a tie breaker and help v_1 to obtain the true value, Fig. 5(b).

One can generalize the observation given in Example 3 by saying that if there are f adversarial agents in the network, there should be $2f+1$ disjoint paths between any given pair of agents in order to make sure that information can be transmitted reliably between those agents. The number of disjoint paths between agent pairs is related to the vertex connectivity via Menger's theorem (West, 2001b).

Theorem 2 (*Menger's Theorem*)**.** *Graph $\mathcal{G}$ has vertex connectivity r if and only if there are r internally vertex disjoint paths between each pair of agents in $\mathcal{G}$.*

This fundamental observation that $2f+1$ connectivity is required to overcome f adversarial agents is classical in the computer science literature (Lynch, 1997). In the following subsection, we describe how the same result (namely that a connectivity of $2f+1$ is required to reliably exchange information in networks despite malicious agents) arises in the context of linear iterative dynamics for information dissemination in networks. In the process, this will introduce the use of zero-dynamics (and strong observability) together with structured systems theory as a means to analyze the resilience of linear dynamics on graphs.

4.2. A system-theoretic perspective on resilient exchange of information

In this subsection, we discuss reliable calculation of agent values in a network in the presence of adversaries. In this setting, each agent $i \in \mathcal{V}$ tries to gather the values (measurements, positions, votes, or other data) of all other agents, despite the actions of adversarial agents in the network. These values can be later used to calculate any arbitrary function of the agents' values. Here, we consider a broadcast model of communication where each agent transmits the same value to all its neighbors. Hence, the adversarial agents are malicious, but do not exhibit full Byzantine behavior. Our goal is to show that the topology of the network (specifically, its connectivity) completely characterizes the resilience of linear iterative strategies to malicious behavior. To this end, we first formally introduce the model of a distributed system under attack.

Distributed System Model Under Attack: Consider a network of n agents (or processors) whose communication is represented by a time-invariant graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$. Suppose that each agent $i \in \mathcal{V}$ begins with some initial value[6] $\psi_i[0] \in \mathbb{R}$ and updates its value over time according to a prescribed rule, i.e.,

$$\psi_i[t+1] = g_i(\{\psi_j^i[t]\}_{j\in\mathcal{N}_i}, t), \quad t \in \mathbb{Z}_{\geq 0}, \tag{4}$$

where $\psi_j^i[t]$ is the state of agent j sent to agent i at time step t and $\psi_i^i[t] = \psi_i[t]$. The update rule, g_i, which is designed *a priori*, can be an arbitrary function and may be different for each agent. For example, for the standard linear consensus protocol (Jadbabaie et al., 2003), this function is simply some linear combination of the values of agent i's neighbors:

$$\psi_i[t+1] = \sum_{j\in\mathcal{N}_i} w_{ij}\psi_j^i[t], \tag{5}$$

where w_{ij} is the weight assigned to agent j's value by agent i.[7] In the absence of any attack, each agent j sends the same consistent information to every neighbor, and hence, $\psi_j^i[t] = \psi_j[t]$, and we can represent the above dynamics in the following compact form:

$$\boldsymbol{\psi}[t+1] = \boldsymbol{W}\boldsymbol{\psi}[t], \tag{6}$$

where the matrix $\boldsymbol{W}$ captures the communication pattern between agents. Moreover, the observation model for each agent i can be described as follows:

$$y_i[t] = \boldsymbol{C}_i\boldsymbol{\psi}[t], \tag{7}$$

where $\boldsymbol{C}_i$ is a $(d_i+1)\times n$ matrix with a single 1 in each row that denotes the states available to agent i (these positions correspond to the neighbors of agent i, including agent i).

Recall that agent i is a *regular* agent if it does not deviate from its prescribed update rule $g_i(\cdot)$. The set of regular agents is denoted by $\mathcal{R}$. A deviation can stem from a failure, e.g., disturbance or noise with a known model, a time delay or signal dropout, or an adversarial action (attack) in the form of arbitrary state updates. Some fundamental differences between faults and attacks were discussed in Section 1.1. Consider a set $\mathcal{F} = \{i_1, i_2, \ldots, i_f\} \subset \mathcal{V}$ of malicious agents. One way to represent an adversarial action at time step t is to use an additive attack signal $\zeta_i[t]$ in the updating rule (5) (Pasqualetti et al., 2012; Sundaram & Hadjicostis, 2011). In particular, instead of applying the update Eq. (5), each agent $i \in \mathcal{F}$ updates its state as

$$\psi_i[t+1] = \sum_{j\in\mathcal{N}_i[t]} w_{ij}[t]\psi_j^i[t] + \zeta_i[t]. \tag{8}$$

Here, an agent is malicious in T time steps if $\zeta_i[t] \neq 0$ for at least one time step $0 \leq t \leq T-1$. Noting that the additive input $\zeta_i[t]$ can be completely arbitrary, this model allows the malicious agent to update its value (via (8)) however it wishes. Writing (8) in vector form yields

$$\begin{aligned}\boldsymbol{\psi}[t+1] &= \boldsymbol{W}\boldsymbol{\psi}[t] + \underbrace{[\mathbf{e}_{i_1} \quad \mathbf{e}_{i_2} \quad \ldots \quad \mathbf{e}_{i_f}]}_{\boldsymbol{B}_{\mathcal{F}}}\boldsymbol{\zeta}_{\mathcal{F}}[t],\\ \boldsymbol{y}[t] &= \boldsymbol{C}\boldsymbol{\psi}[t],\end{aligned} \tag{9}$$

where $\boldsymbol{\zeta}_{\mathcal{F}}[t] = [\zeta_{i_1}, \zeta_{i_2}, \ldots, \zeta_{i_f}]$ models an unknown additive error (attack) vector, $\boldsymbol{C} = [\boldsymbol{C}_1' \ \boldsymbol{C}_2' \ \ldots \ \boldsymbol{C}_q']'$, and $\boldsymbol{C}_i$ is as in (7). Since the set of malicious agents $\mathcal{F}$ is unknown to the regular agents, the specific value of matrix $\boldsymbol{B}_{\mathcal{F}}$ is also not known to those agents (although those agents do know that $\boldsymbol{B}_{\mathcal{F}}$ will contain at

[6] This value represents the initial information available to agent i. For example, it could represent a sensor measurement (if the agent is a sensor), a vote (if the agents are trying to determine a course of action), or an opinion (if the agents represent individuals).

[7] Throughout this survey, we will primarily discuss linear updating rules as they cover a broad class of distributed protocols studied in the literature.

most f columns under the f-total adversary model, with a single 1 in each column). It is easy to see that in the absence of attacks, the model in (9) boils down to the one in (6) and (7).

Observability and Connectivity: Recall that the goal of each agent in this section is to gather the initial values of the other agents (since those represent the pieces of information inherently held by each agent), and use them for calculating some function of the agents' values. Moreover, throughout this section, we will assume that each agent knows the structure of the network, i.e., $\boldsymbol{W}$ is known to every agent. As such, in the absence of attacks, based on (6), it is easy to see recovering the initial value vector $\boldsymbol{\psi}[0] = [\psi_1[0]\ \psi_2[0], \ldots, \psi_n[0]]'$ simply boils down to solving an initial-state observability problem for the system (6)–(7). In particular, from basic linear systems theory, it then follows that $\boldsymbol{\psi}[0]$ can be recovered by agent i if the pair $(\boldsymbol{W}, \boldsymbol{C}_i)$ is observable. To connect the underlying network (which is captured by the sparsity pattern in $\boldsymbol{W}$) to the observability of the pair $(\boldsymbol{W}, \boldsymbol{C}_i)$, Sundaram and Hadjicostis (2008) leveraged structured system theory (Section 2.4) to show that if the underlying network is connected, then the weights in the update Eq. (5) for each agent (which form the nonzero elements in $\boldsymbol{W}$) can be designed in a way that ensures observability of $(\boldsymbol{W}, \boldsymbol{C}_i)$.

The above discussion reveals how concepts from systems theory such as observability can be combined with basic graph-theoretic notions such as connectivity to study the process of information diffusion over networks. It is natural to thus wonder whether a marriage of ideas between systems theory and graph theory will continue to be fruitful while analyzing the adversarial setting. The results from Sundaram and Hadjicostis (2011), summarized below, establish that this is indeed the case.

Suppose that a subset $\mathcal{F}$ of agents is malicious and deviates from the update rule (5). Thus, the new goal is to recover $\boldsymbol{\psi}[0]$ for the model in (9). As in the non-adversarial case, we start by examining the observation model at agent i. We note that the set of all values seen by agent i during the first $L+1$ time-steps of the linear iteration (for any non-negative integer L) is given by

$$y_i[0:L] = \mathcal{O}_{i,L}\boldsymbol{\psi}[0] + \mathcal{M}^{\mathcal{F}}_{i,L}\boldsymbol{\zeta}_{\mathcal{F}}[0:L-1], \tag{10}$$

where $y_i[0:L] = \left[y_i'[0]\ \ y_i'[1]\ \ \cdots\ \ y_i'[L]\right]'$ and $\boldsymbol{\zeta}_{\mathcal{F}}[0:L-1] = \left[\zeta_{\mathcal{F}}'[0]\ \ \zeta_{\mathcal{F}}'[1]\ \ \cdots\ \ \zeta_{\mathcal{F}}'[L-1]\right]'$. Matrices $\mathcal{O}_{i,L}$ and $\mathcal{M}^{\mathcal{F}}_{i,L}$ are the observability and invertibility matrices, respectively (from the perspective of agent i), and can be expressed recursively as

$$\mathcal{O}_{i,L} = \begin{bmatrix} \boldsymbol{C}_i \\ \mathcal{O}_{i,L-1}\boldsymbol{W} \end{bmatrix}, \quad \mathcal{M}^{\mathcal{F}}_{i,L} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \mathcal{O}_{i,L-1}\boldsymbol{B}_{\mathcal{F}} & \mathcal{M}^{\mathcal{F}}_{i,L-1} \end{bmatrix}, \tag{11}$$

where $\mathcal{O}_{i,0} = \boldsymbol{C}_i$ and $\mathcal{M}^{\mathcal{F}}_{i,0}$ is the empty matrix (with zero columns). The question of interest is the following: *Under what conditions can agent i recover $\boldsymbol{\psi}[0]$ based on a sufficiently large sequence of observations, despite the presence of the unknown inputs $\boldsymbol{\zeta}_{\mathcal{F}}$?*

As it turns out, the answer to the above question is intimately tied to the system-theoretic concept of *strong observability*. In particular, the linear system (9) is said to be strongly observable w.r.t. agent i if $y_i[t] = 0$ for all t implies $\boldsymbol{\psi}[0] = 0$ (regardless of the values of the unknown inputs $\boldsymbol{\zeta}_{\mathcal{F}}[t]$). Moreover, if such a strong observability condition holds, then this is equivalent to saying that agent i will be able to uniquely determine the initial condition $\boldsymbol{\psi}[0]$ based on the knowledge of its output sequence, regardless of the unknown inputs.

Strong Observability and Connectivity: For the non-adversarial setting, having a connected graph is enough to ensure observability of the pair $(\boldsymbol{W}, \boldsymbol{C}_i), \forall i \in \mathcal{V}$. In a similar vein, we need to now discern how the structure of the underlying network impacts strong observability. To this end, we present a simple argument to demonstrate that if the network is not adequately connected, then system (9) will not be strongly observable w.r.t. certain agents in the graph. For simplicity, let $\mathcal{G}$ be undirected. Now suppose the connectivity κ of $\mathcal{G}$ is such that $\kappa \leq |\mathcal{F}|$. This implies the existence of a vertex cut $\mathcal{S}_2$ of size at most $|\mathcal{F}|$ that separates the graph into two disjoint parts. Let the vertex sets for these disjoint parts be denoted by $\mathcal{S}_1$ and $\mathcal{S}_3$. After reordering the agents such that the agents in $\mathcal{S}_1$ come first, followed by those in $\mathcal{S}_2$ and then $\mathcal{S}_3$, the weight matrix takes the following form:

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{W}_{11} & \boldsymbol{W}_{12} & \boldsymbol{0} \\ \boldsymbol{W}_{21} & \boldsymbol{W}_{22} & \boldsymbol{W}_{23} \\ \boldsymbol{0} & \boldsymbol{W}_{32} & \boldsymbol{W}_{33} \end{bmatrix}.$$

The structure of the above matrix follows immediately from the fact that the agents in $\mathcal{S}_1$ can interact with those in $\mathcal{S}_3$ only via the agents in $\mathcal{S}_2$. Now suppose all the agents in $\mathcal{S}_2$ are adversarial; this is indeed feasible since $|\mathcal{S}_2| \leq |\mathcal{F}|$. Moreover, let the initial condition $\boldsymbol{\psi}[0]$ be of the form

$$\boldsymbol{\psi}[0] = \begin{bmatrix} \boldsymbol{\psi}_{\mathcal{S}_1}[0] & \boldsymbol{\psi}_{\mathcal{S}_2}[0] & \boldsymbol{\psi}_{\mathcal{S}_3}[0] \end{bmatrix}' = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{v} \end{bmatrix}',$$

where $\boldsymbol{v}$ is a non-zero vector in $\mathbb{R}^{|\mathcal{S}_3|}$. If the adversarial inputs are of the form $\boldsymbol{\zeta}_{\mathcal{F}}[t] = -\boldsymbol{W}_{23}\boldsymbol{\psi}_{\mathcal{S}_3}[t]$, then it is easy to see that the states of the agents in $\mathcal{S}_1$ and $\mathcal{S}_2$ remain at zero, i.e., according to (9) we have

$$\boldsymbol{\psi}[1] = \boldsymbol{W}\boldsymbol{\psi}[0] + \boldsymbol{\zeta}_{\mathcal{F}}[0] = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{W}_{23}\boldsymbol{v} \\ \boldsymbol{W}_{33}\boldsymbol{v} \end{bmatrix} - \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{W}_{23}\boldsymbol{v} \\ \boldsymbol{0} \end{bmatrix}.$$

Thus, an agent i in $\mathcal{S}_1$ observes a sequence of zeros. It follows that there is no way for agent i to distinguish the zero initial condition from the non-zero initial condition we considered in this example. Thus, system (9) is not strongly observable w.r.t. agent i.

The above argument serves to once again highlight the interplay between control- and graph-theory in the context of information diffusion over networks. Moreover, it suggests that in order for every agent to uniquely determine the initial condition, the *connectivity of the network has to somehow scale with the number of adversaries.* Using a more refined argument than the one we presented above, it is possible to show that a connectivity of $2f+1$ is necessary for the problem under consideration (Pasqualetti et al., 2012; Sundaram & Hadjicostis, 2011), where f is the maximum number of malicious agents in the network. In Sundaram and Hadjicostis (2011), it is established that a connectivity of $2f+1$ is also sufficient for the linear iterative strategy to reliably disseminate information between regular agents in the network despite the actions of up to f malicious adversaries.

Theorem 3 (*Sundaram & Hadjicostis, 2011*)**.** *Given a fixed network with n agents described by a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, let f denote the maximum number of malicious agents that are to be tolerated in the network, and let κ_{ij} denote the size of the smallest (i,j)-cut between any two vertices j and i. Then, regardless of the actions of the malicious agents, agent i can uniquely determine all of the initial values in the network via a linear iterative strategy if and only if $\min_j \kappa_{ij} \geq 2f+1$. Furthermore, if this condition is satisfied, i will be able to recover the initial values after the agents run the linear iterative strategy with almost any choice of weights for at most n time-steps.*

A key ingredient in the proof of Theorem 3 is establishing that if $\mathcal{G}$ is $(2f+1)$-connected, then the tuple $(\boldsymbol{W}, \boldsymbol{B}_{\mathcal{F}}, \boldsymbol{C}_i)$ is strongly observable (i.e., does not possess any zero dynamics) $\forall i \in \mathcal{V}$, for almost all choices of the weight matrix $\boldsymbol{W}$, and under any f-total

adversarial set $\mathcal{F}$.[8] This interdependence between system and graph-theoretic properties is schematically illustrated in Fig. 5(c).

One important point to note is that the approaches developed in Pasqualetti et al. (2012) and Sundaram and Hadjicostis (2011) to combat adversaries require each regular agent to possess complete knowledge of the network structure, and to perform a large amount of computation to identify the malicious sets. For large-scale networks, this may be infeasible. One may thus ask whether it is possible to resiliently diffuse information across a network when each regular agent only has *local* knowledge of its own neighborhood, and can run only simple computations. In the subsequent sections, we provide an overview of work showing that this is indeed possible. However, as we shall see, the lack of global information will dictate the need for stronger requirements on the network topology (relative to $(2f+1)$-connectivity).

Example 4. Consider the graph shown in Fig. 6(a). The objective is for agent v_1 to calculate the function $\sum_{i=1}^{6}\psi_i^2[0]$ despite the presence of a malicious agent in the network, i.e. $f=1$. In this graph, agents v_2, v_3, and v_4 are neighbors of v_1, and v_5 and v_6 have three internally vertex-disjoint paths to v_1. Thus, $\kappa_{j1}\geq 3$ for all j and, based on Theorem 3, v_1 is able to calculate the desired function after running the linear iteration (with almost any choice of weights) for at most $n=6$ time-steps. We take each of the edge and self loop weights to be i.i.d. random variables from the set $\{-5,-4,-3,-2,-1,1,2,3,4,5\}$ with equal probabilities. These weights produce the matrix shown in Fig. 6.

Since agent 1 has access to its own state and the states of its neighbors, we have $\boldsymbol{C}_1=[\mathbf{I}_4 \quad \mathbf{0}]$. Based on these values, matrices $\mathcal{O}_{1,L}$ and $\mathcal{M}_{1,L}^{\mathcal{F}}$ are obtained and v_1 can calculate the initial state vector $\boldsymbol{\psi}[0]$ using (10).

Suppose that the initial values of the agents are $\boldsymbol{\psi}[0]=[3\ \ -1\ \ 4\ \ -4\ \ 7\ \ 11]'$ and v_4 is a malicious agent. At time steps 1 and 2, v_4 adds an additive error of $\zeta_4[1]=-8$ and $\zeta_4[1]=-12$ to its updating rule. The values of all agents over the first three time-steps of the linear iteration are given by $\boldsymbol{\psi}[0]=[3\ \ -1\ \ 4\ \ -4\ \ 7\ \ 11]'$, $\boldsymbol{\psi}[1]=[-26\ \ 0\ \ 26\ \ 49\ \ 46\ \ -80]'$, and $\boldsymbol{\psi}[2]=[199\ \ 43\ \ -134\ \ -222\ \ -446\ \ 309]'$. The values seen by v_1 at time-step t are given by $y_1[t]=\boldsymbol{C}_1\boldsymbol{\psi}[t]$; agent v_1 can now use $y_1[t]$ to calculate the vector of initial values, despite the efforts of the malicious agent. Agent v_1 has to find a set $\mathcal{F}_i$ for which $y_1[0:2]$ falls into the column space of $\mathcal{O}_{1,2}$ and $\mathcal{M}_{1,2}^{\mathcal{F}}$. In this example, v_1 can figure out that this holds for $j=4$. Then, it finds vectors $\bar{\boldsymbol{\psi}}$ and $\boldsymbol{\zeta}_{\mathcal{F}_4}[0:1]$ such that $y_i[0:2]=\mathcal{O}_{1,2}\bar{\boldsymbol{\psi}}+\mathcal{M}_{1,2}^{\mathcal{F}_4}\boldsymbol{\zeta}_{\mathcal{F}_4}[0:1]$ as $\boldsymbol{\zeta}=[3\ \ -1\ \ 4\ \ -4\ \ 7\ \ 11]'$ and $\boldsymbol{\zeta}_{\mathcal{F}_4}[0:1]=[-8\ \ -12]'$. Agent v_1 now has access to $\boldsymbol{\psi}[0]=\bar{\boldsymbol{\psi}}$ and can calculate $\sum_{i=1}^{6}\psi_i^2[0]=212$.

It is worth noting that for the network in Fig. 6(a), we have $\kappa_{26}=2$, since the set $\mathcal{F}=\{v_4,v_5\}$ forms a $(2,6)$- cut (i.e., removing agents v_4 and v_5 removes all paths from v_2 to v_6). Thus, agent v_6 is not guaranteed to be able to calculate any function of agent v_2's value when there is a faulty agent in the system. In particular, one can verify that in the example above, where agent v_4 is malicious and updates its values with the errors $\zeta_4[1]=-8$ and $\zeta_4[1]=-12$, the values seen by v_6 during the first three time steps of the linear iteration are the same as the values seen by agent v_6 when $\psi_1[0]=4$ or $\psi_1[0]=-3$ and agent 5 is malicious with $\psi_5[0]=4$ or $\psi_5[0]=6$. In other words, agent v_6 cannot distinguish the case when agent v_4 is faulty from the case where agent v_5 is faulty (with different initial values in the network).

Extra Conditions for Byzantine Attacks: For point-to-point communications, which are prone to Byzantine attacks, in addition to the network connectivity, which has to be at least $2f+1$, the total number of agents must satisfy $n\geq 3f+1$. This is because of the fact that if i receives j's value reliably, it still does not know what j told other agents in the network. Thus, there must be a sufficient number of non-Byzantine agents in the network in order for i to ascertain what j told 'most' of the agents (Dolev, 1982; Dolev, Dwork, Waarts, & Yung, 1993). This is schematically shown in Fig. 6(b). Suppose that a decision must be made between two values 0 and 1. Since there are f Byzantine agents, agent i must receive at least $f+1$ consistent messages so that it makes sure that the majority of the votes it receives are from non-Byzantine agents. Clearly, at least half of the messages sent from regular agents to agent i are consistent. These $\frac{n-f}{2}$ messages must outnumber the f messages sent from the Byzantine set to agent i. Thus, we must have $\frac{n-f}{2}>f$ or $n>3f$.

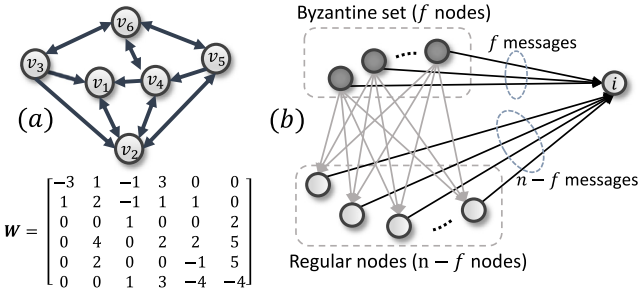


Fig. 6. (a) Graph discussed in Example 4 and its weight matrix. (b) Schematics showing the necessity of having sufficiently large number of non-Byzantine agents in a network.

5. Resilient distributed consensus

Distributed consensus is a well studied application of information diffusion in networks. In distributed consensus, every agent in the network has some information to share with the others, and the entire network must come to an agreement on an appropriate function of that information (Jadbabaie et al., 2003; Olfati-Saber et al., 2007; Olfati-Saber & Murray, 2004; Sundaram & Hadjicostis, 2008). In the resilient version of distributed consensus, the algorithm has to be modified in such a way that it maintains the consensus value in a desired region despite the actions of adversarial agents who attempt to steer the states outside that region (or disrupt agreement entirely). The desired steady state value can vary according to the application of interest (Ghaderi & Srikant, 2014; Pirani, Shahrivar, & Sundaram, 2015).

Remark 1 (*A Fundamental Limitation*)**.** In the standard linear consensus dynamics, a single malicious agent (shown in black in Fig. 7) can drive the consensus value towards its own state simply by keeping its value constant, as shown in Fig. 7, left. More generally, since the initial values of the agents are assumed to be known only to the agents themselves, an adversarial agent can simply change its own initial value and participate in the rest of the algorithm as normal. This would allow the adversary to affect the final consensus value (through its modified local value), but never be detected. Thus, perfect calculation of any function of initial values is generally impossible under adversarial behavior. This is a fundamental limitation of all distributed algorithms for any problem where each agent holds data that is required by others to compute their functions (e.g., as in consensus, function calculation, or distributed optimization), as will be discussed later.

[8] Note that when $\mathcal{F}=\emptyset$ (i.e., in the absence of adversaries), we immediately recover that connectivity of $\mathcal{G}$ implies observability of $(\boldsymbol{W},\boldsymbol{C}_i)$, $\forall i\in\mathcal{V}$, for almost all choices of the weight matrix $\boldsymbol{W}$.

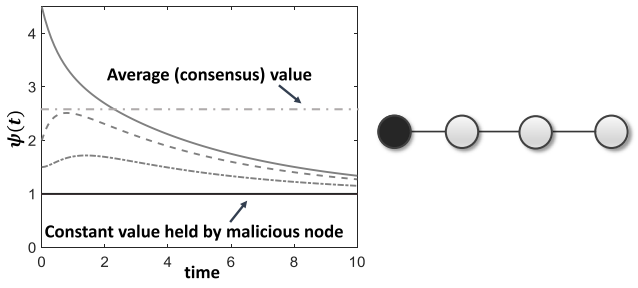


Fig. 7. State trajectories of agents in the presence of a stubborn agent.

Stubborn Agents. In the context of opinion dynamics that capture the spread of ideas and behaviors over a social network, agents that never update their opinions (as mentioned in Remark 1) are typically referred to as *stubborn agents* or *zealots* (Ghaderi & Srikant, 2014; Pirani & Sundaram, 2014; Waagen, Verma, Chan, Swami, & D'Souza, 2015; Yildiz, Acemoglu, Ozdaglar, Saberi, & Scaglione, 2011). Such stubborn agents can be viewed as adversarial agents who act in a specific structured manner (Gentz, Wu, Wai, Scaglione, & Leshem, 2016). The authors in Ghaderi and Srikant (2014) examine how the location of stubborn agents, and their extent of stubbornness, affect consensus dynamics in a variety of networks. In particular, they show that for a certain class of linear update protocols, the values of the other agents converge asymptotically to a convex combination of the values held by the stubborn agents. For continuous-time opinion dynamics, it turns out that the rate of such convergence is characterized by spectral properties of a matrix known as the grounded Laplacian matrix (Barooah & Hespanha, 2006).[9] In Pirani and Sundaram (2014), the authors derive new bounds on the smallest eigenvalue of such grounded Laplacian matrices; these bounds have immediate implications for the rate of convergence of consensus dynamics with stubborn agents.

5.1. Classical approaches

Classical results on distributed consensus in the presence of Byzantine agents date back to the computer science literature (Dolev, Lynch, Pinter, Stark, & Weihl, 1986) showing that the regular agents can always reach a consensus if and only if (1) the number of Byzantine agents is less than $\frac{1}{2}$ of the network connectivity, and (2) less than $\frac{1}{3}$ of the total number of agents. This aligns with the results obtained via control- and graph-theoretic analysis of the linear iterative dynamics with malicious agents as discussed in the previous section (Pasqualetti et al., 2012; Sundaram & Hadjicostis, 2011). However, these works require each regular agent to have full knowledge of the network topology, and for each regular agent to perform a substantial amount of computations. The following subsection describes an alternative scalable and "purely local" method for resilient consensus.

5.2. Purely local approaches

By imposing stronger conditions on the network topology (beyond being just $(2f+1)$-connected), one can formulate algorithms that can handle worst case f-local Byzantine attacks with much less computational cost. In this class of algorithms, which were first named *approximate agreement* (Dolev et al., 1986), each regular agent disregards the largest and smallest f values received from its neighbors at each iteration and updates its state to be the average of a carefully chosen subset of the remaining values (such quantities are known as *trimmed means* in the robust statistics literature Huber, 1972). These methods were extended to a class of algorithms named *Mean-Subsequence-Reduced* (MSR) algorithms (Kieckhafer & Azadmanesh, 1994). In LeBlanc and Koutsoukos (2011) a continuous-time variation of the MSR algorithms, named the *Adversarial Robust Consensus Protocol* (ARC-P) was proposed.

In what follows, we discuss an extension of MSR algorithms, called *Weighted-Mean-Subsequence Reduced* (W-MSR) in LeBlanc et al. (2013), which can handle f-local adversarial agents. The algorithm is as follows.

(1) Let $\psi_i[t] \in \mathbb{R}$ be the value maintained at each time step by each regular agent i. At each time-step t, each regular agent i receives neighbor values $\psi_j[t], j \in \mathcal{N}_i$, and ranks them from largest to smallest.
(2) If there are f or more values larger than $\psi_i[t]$, regular agent i removes the f largest values. If there are fewer than f values larger than $\psi_i[t]$, regular agent i removes all of these larger values. This same logic is applied to the smallest values in regular agent i's neighborhood. Let $\mathcal{R}_i[t]$ denote the set of agents whose values were removed by i at time step t.
(3) Each regular agent i updates its value as

$$\psi_i[t+1] = w_{ii}[t]\psi_i[t] + \sum_{j \in \mathcal{N}_i \setminus \mathcal{R}_i[t]} w_{ij}[t]\psi_j[t], \tag{12}$$

where $w_{ii}[t]$ and $w_{ij}[t]$ satisfy the following conditions:

- $\sum_{j=1}^{n} w_{ij}[t] = 1, \quad \forall t \in \mathbb{Z}_{\geq 0}$;
- $w_{ij}[t] = 0$ whenever $j \notin \mathcal{N}_i \setminus \mathcal{R}_i[t], t \in \mathbb{Z}_{\geq 0}$;
- there exists $\alpha > 0$ such that $w_{ij}[t] \geq \alpha$ for all $j \in \mathcal{N}_i \setminus \mathcal{R}_i[t], t \in \mathbb{Z}_{\geq 0}$.

We call the largest number of values that each agent could throw away the *parameter* of the algorithm (it is equal to $2f$ in the above algorithm). Note that the above steps are only followed by the regular agents; the adversarial agents can update their states however they wish. Before presenting conditions for resilient consensus, in the following example, we show that even for networks that have a high connectivity (i.e., networks with several disjoint paths between every pair of agents), the W-MSR algorithm can fail to guarantee consensus in the presence of adversaries.

Example 5. In the graph shown in Fig. 3(b), suppose that the initial value of agents in set $\mathcal{V}_1$ and set $\mathcal{V}_2$ is zero and 1, respectively. For $f = 1$, if each agent disregards the largest and smallest values in its neighborhood, then the value of agents in sets $\mathcal{V}_1$ and $\mathcal{V}_2$ remains the same as their initial value for all $t \geq 0$. As a result, consensus will not be achieved even though there are no malicious agents. This lack of consensus is despite the fact that the connectivity of the graph is $\frac{n}{2}$, and arises due to the fact that the local state-dependent filtering in W-MSR causes the sets $\mathcal{V}_1$ and $\mathcal{V}_2$ to be disconnected at each iteration.

Although the network connectivity is no longer an appropriate metric for analyzing the resilience of W-MSR dynamics, the notion of graph robustness from LeBlanc et al. (2013) and Zhang and Sundaram (2012) (see Section 2.1.2) turns out to be the key concept. We first start with the following concept. Denote the maximum and minimum values of the normal agents at time-step t as $M[t]$ and $m[t]$.

[9] Given a graph Laplacian matrix $\boldsymbol{L}$, the grounded Laplacian is a sub-matrix of $\boldsymbol{L}$ obtained by removing the rows and columns corresponding to the stubborn agents from $\boldsymbol{L}$.

Definition 4 (*f-local Safe*)**.** Under the f-local adversarial model, the W-MSR algorithm is said to be f-local safe if both of the following conditions are satisfied: (i) all regular agents reach consensus for any choice of initial values, and (ii) the regular agents' values (including the final consensus value) are always in the range $\left[m[0], M[0]\right]$.

The following result provides conditions under which the W-MSR algorithm guarantees (or fails) to be f-local safe.

Theorem 4 (*LeBlanc et al., 2013*)**.** *Under the f-local Byzantine adversary model, the W-MSR algorithm with parameter $2f$ is f-local safe if the network $\mathcal{G}$ is $(2f+1)$-robust. Furthermore, for any $f > 0$, there exists a $2f$-robust network which fails to reach consensus based on the W-MSR algorithm with parameter $2f$.*

As discussed in Section 2.1.2, the robustness condition used in Theorem 4 is much stronger than the network connectivity condition which was required in classical distributed consensus algorithms. However, this stronger condition can be considered as the price to be paid for a computationally tractable resilient consensus algorithm which is able to tolerate worst-case Byzantine attacks. Furthermore, under this condition, one gains the ability to tolerate f-local adversaries (rather than f-total adversaries).

Several other variations of the above approach, including extensions to second order consensus with asynchronous time delay and applications to formation control of mobile robots, are discussed in Dibaji and Ishii (2017), Dibaji, Ishii, and Tempo (2018) and Saulnier, Saldana, Prorok, Pappas, and Kumar (2017). Applications of resilient consensus in multi-robot systems using Wi-Fi communication are studied in Yemini, Nedić, Goldsmith, and Gil (2021). Resilient flocking in multi-robot systems requires the extension of the above techniques to time varying networks, which is studied in Saldana, Prorok, Sundaram, Campos, and Kumar (2017). There, it is shown that if the required network robustness condition is not satisfied at all times, the network can still reach resilient consensus if the union of communication graphs over a bounded period of time satisfies $(2f+1)$-robustness. Moreover, a control policy to attain such resilient behavior in the context of perimeter surveillance with a team of robots was proposed.

5.3. Resilient vector consensus

The W-MSR algorithm described in the previous section considered the case where agents maintain and exchange scalar quantities and remove "extreme" values at each iteration. However, the extension to multi-dimensional vectors requires further considerations since there may not be a total ordering among vectors. One option is to simply run W-MSR on each component of the vector separately. If the graph is $(2f+1)$-robust, this would guarantee that all regular agents reach a consensus in a hypercube formed by the initial vectors of the regular agents (despite the actions of any f local set of Byzantine agents). Keeping the agents' states within the *convex hull* of the initial vectors (as opposed to simply a hypercube), however, requires further considerations, as we will discuss in this section. In particular, the convex hull of a set of vectors is a subset of the region (the hypercube) formed by the convex hull of their components separately. This is shown in Fig. 8(a) in which the triangle is the convex hull of the three points in $\mathbb{R}^2$ and the gray rectangle is the box formed by calculating the convex hull of each component of the three vectors separately. Thus, the component-wise convex hull gives an overestimate of the actual convex hull of the vectors.

First attempts to address Byzantine resilient vector consensus to the convex hull of the initial values of the regular agents were provided in Vaidya (2014) and Xiang and Vaidya (2016), and were further developed in the context of rendezvous multi robot systems by Park and Hutchinson (2017). To describe these approaches, the following notion of safe points in a d-dimensional space is useful.

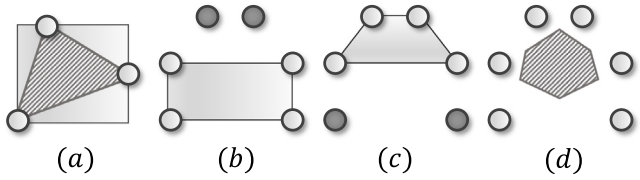


Fig. 8. (a) The box over-estimating the convex-hull of agents in $\mathbb{R}^2$. (b),(c) show non-intersecting convex hull of the regular agents. (d) the dark region is the centerpoint region of all six agents. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Definition 5 (*Park & Hutchinson, 2017*)**.** Given a set of n agents in $\mathbb{R}^d$ of which at most f are adversarial, a point p that is guaranteed to lie in the interior of the convex hull of $(n-f)$ regular points (i.e., non-adversarial) is called an f-safe point.

Based on the above definition, the resilient vector consensus algorithm relies on the computation of f-safe points by each agent as follows:

(1) Let $\boldsymbol{\psi}_i[t] \in \mathbb{R}^d$ be the value maintained at each time step by each regular agent i. At iteration t, each regular agent i gathers the state values of its neighbors $\mathcal{N}_i$.
(2) Each regular agent i computes an f-safe point, denoted by $s_i[t]$, of points corresponding to its neighbors' states.
(3) Each regular agent i then updates its state by moving toward the safe point $s_i[t]$, i.e.,

$$\boldsymbol{\psi}_i[t+1] = \alpha_i[t] s_i[t] + (1-\alpha_i[t])\,\boldsymbol{\psi}_i[t], \tag{13}$$

where $\alpha_i[t] \in (0,1)$ is a dynamically chosen parameter whose value depends on the application.

It was shown in Park and Hutchinson (2017) that if all regular agents follow the above routine, they are guaranteed to converge to some point in the convex hull of their initial states. The following proposition shows conditions for the existence of f-safe points.

Proposition 1 (*Park & Hutchinson, 2017*)**.** *Given n points in $\mathbb{R}^d$, where $d \in \{1, 2, \ldots, 8\}$, and at most f points belong to adversaries, then there exists an f-safe point if $n \geq (f+1)(d+1)$. The claim also holds for $d > 8$ if Reay's conjecture is true (Reay, 1968).*

In Abbas, Shabbir, Li et al. (2020), it is shown that $n \geq (f+1)(d+1)$ is also a necessary condition for the existence of an f-safe point. An example is shown in Fig. 8(b) and (c). The malicious agents are shown with darker colors. Here, for $f = 2$ malicious agents, there is no 2-safe point (an interior point in the intersection of convex hull of four regular agents).

The main question is how to find these f-safe points from a given set of points in $\mathbb{R}^d$. In Park and Hutchinson (2017) and Vaidya (2014), this is done via the *Tverberg partitioning algorithm*, which partitions points into subsets such that the convex hull of the partitions has a non-empty intersection, provided that the number of agents is sufficiently large; see Bárány and Soberón (1966) for more details. A similar approach is recently adopted in Yan et al. (2020). However, finding Tverberg partitions is computationally hard in practice (although it is not proved that the problem is NP hard). To achieve fast algorithms, one has to pay the price of reducing the number of parts in the partition,

Table 1
Summary of references on **resilient consensus**. We classify references based on (i) the attack model; (ii) the dimension of the consensus variable $\psi_i[\cdot]$ at each agent i; (iii) the graph-structure, i.e., whether the underlying graph is fully-connected (complete) or a general network; and (iv) the information content available to each agent: we use the term 'local' for update rules that require each regular node to only their own neighbors (and not the global network topology), and 'non-local' otherwise.

Attack model	Dimension of variable		Graph topology		Information content	
	Scalar	Vector	Complete	General	Local	Non-Local
f-total malicious	LeBlanc and Koutsoukos (2011), Pasqualetti et al. (2012) and Sundaram and Hadjicostis (2011)	–	LeBlanc and Koutsoukos (2011)	Pasqualetti et al. (2012) and Sundaram and Hadjicostis (2011)	–	LeBlanc and Koutsoukos (2011), Pasqualetti et al. (2012) and Sundaram and Hadjicostis (2011)
f-total Byzantine	Dolev et al. (1986), LeBlanc et al. (2013) and Vaidya, Tseng, and Liang (2012)	Park and Hutchinson (2017), Vaidya (2014) and Vaidya and Garg (2013)	Dolev et al. (1986) and Vaidya and Garg (2013)	LeBlanc et al. (2013), Park and Hutchinson (2017), Vaidya (2014) and Vaidya et al. (2012)	LeBlanc et al. (2013), Park and Hutchinson (2017), Vaidya (2014) and Vaidya et al. (2012)	Dolev et al. (1986) and Vaidya and Garg (2013)
f-local Byzantine	LeBlanc et al. (2013)	Abbas, Shabbir, Li and Koutsoukos (2020), Shabbir, Li, Abbas, and Koutsoukos (2020a) and Yan, Mo, Li, and Wen (2020)	–	Abbas, Shabbir, Li et al. (2020), LeBlanc et al. (2013), Shabbir et al. (2020a) and Yan et al. (2020)	Abbas, Shabbir, Li et al. (2020), LeBlanc et al. (2013), Shabbir et al. (2020a) and Yan et al. (2020)	–
Stubborn Agents	Ghaderi and Srikant (2014), Pirani and Sundaram (2014) and Yildiz et al. (2011)	–	–	Ghaderi and Srikant (2014), Pirani and Sundaram (2014) and Yildiz et al. (2011)	Ghaderi and Srikant (2014), Pirani and Sundaram (2014) and Yildiz et al. (2011)	–

i.e., the number of malicious agents. Linear time approximation algorithms to find Tverberg points, i.e., f-safe agents, have been proposed in Mulzer and Werner (2013), provided that $f \leq \lceil \frac{n}{2^d} \rceil - 1$. On the other hand, Tverberg partitioning provides strong conditions for f-safe points, i.e., the outcome of the Tverberg partitioning algorithm are f-safe points, but the reverse is not true. A less conservative approach to find f-safe points via the notion of a centerpoint was developed in Shabbir, Li, Abbas, and Koutsoukos (2020b), as explained below.

Definition 6. Given a set X of n points in $\mathbb{R}^d$, a centerpoint p is a point, not necessarily from X, such that any closed half-space[10] of $\mathbb{R}^d$ containing p also contains at least $\lceil \frac{n}{d+1} \rceil$ points from X.

By the centerpoint theorem, every finite set of points in $\mathbb{R}^d$ has a centerpoint (Matousek, 2002). It is shown that for a set of n points in $\mathbb{R}^d$ and $f \leq \lceil \frac{n}{d+1} \rceil - 1$, the region of f-safe points is *equivalent to* the centerpoint region (Shabbir et al., 2020b). The centerpoint of six points is shown in Fig. 8(d). The following result provides conditions for finding f-safe points.

Proposition 2 (*Shabbir et al., 2020b*)**.** *Given n points in $\mathbb{R}^d$ for which at most f points belong to adversaries, then an f-safe point can be computed (using centerpoint) if*

$$f \leq \lceil \frac{n}{d+1} \rceil - 1 \quad d = 2, 3,$$
$$f \leq \lceil \frac{n}{d^2} \rceil - 1 \qquad d > 3. \tag{14}$$

For a summary of the references surveyed in this section, please see Table 1.

[10] A closed half-space in $\mathbb{R}^d$ is a set of the form $\{x \in \mathbb{R}^d : a'x \geq b\}$ for some $a \in \mathbb{R}^d \setminus \{0\}$.

6. Resilient distributed optimization

While the consensus problem discussed in the previous section considered the scenario where each agent has a static initial value, a more general setting is that of distributed optimization. In this setting, each agent $i \in \mathcal{V}$ has a convex function $g_i : \mathbb{R} \to \mathbb{R}$ (with bounded subgradients) which is only available to agent i.[11] The objective is for the agents to solve the following global optimization problem in a distributed manner:

$$\min_{\psi} \quad g(\psi) = \frac{1}{n} \sum_{i=1}^{n} g_i(\psi). \tag{15}$$

A common approach to solve this problem is to use a synchronous iterative consensus-based protocol in which agents use a combination of consensus dynamics and gradient flow to find a minimizer of $g(\psi)$ (Nedic & Ozdaglar, 2009). More specifically, at every time-step $t \in \mathbb{N}$, each agent i maintains an estimate $\psi_i(t) \in \mathbb{R}$ of the solution to (15), and updates it based on the information received from its neighbors, as follows

$$\psi_i[t+1] = w_{ii}[t]\psi_i[t] + \sum_{j \in \mathcal{N}_i[t]} w_{ij}[t]\psi_j[t] - \alpha_t d_i[t]. \tag{16}$$

In the above update rule, $d_i[t]$ is a subgradient of g_i evaluated at $w_{ii}[t]\psi_i[t] + \sum_{j \in \mathcal{N}_i[t]} w_{ij}[t]\psi_j[t]$, and α_t is the step size sequence corresponding to the influence of the subgradient on the update rule at each time-step. As before, weights $w_{ii}[t], w_{ij}[t], j \in \mathcal{N}_i[t]$ specify a convex combination at each time-step t, and are bounded away from zero (whenever they are positive). Dynamics (16) can be represented in the following vector form as

$$\boldsymbol{\psi}[t+1] = \boldsymbol{W}[t]\boldsymbol{\psi}[t] - \alpha_t \boldsymbol{d}[t], \tag{17}$$

where $\boldsymbol{W}[t]$ is a doubly-stochastic matrix. The following result shows that the update rule (16) allows the agents in the network to distributively solve the global optimization problem (15).

[11] We discuss the case of multi-dimensional functions later in this section.

Proposition 3 (*Nedic & Ozdaglar, 2009*)**.** *Suppose that $\mathcal{G}$ is strongly connected at each time step and that the subgradients of each of the local functions g_i are bounded. For the update rule (16) with step sizes satisfying $\sum_{t\in\mathbb{N}} \alpha_t = \infty$ and $\sum_{t\in\mathbb{N}} \alpha_t^2 < \infty$, we have*

$$\lim_{t\to\infty} \|\psi_i[t] - \psi^*\| = 0, \tag{18}$$

for all $i \in \mathcal{V}$, where ψ^* is the global minimizer of $g(\psi)$.

Our objective here is to summarize the vulnerabilities of such protocols to adversarial agents, and to provide an overview of secure distributed optimization algorithms that have provable safety guarantees despite the presence of such agents. Recent works have focused on some gradient-based metrics to detect and identify malicious agents in a distributed optimization algorithm (Ravi, Scaglione, & Nedic, 2019). However, similar to previous sections, our focus is on graph-theoretic methods. As before, we assume that adversarial agents can update their states in a completely arbitrary manner.

Remark 2 (*Fundamental Limitation Revisited*)**.** Following the fundamental limitation discussed in Remark 1, one can easily argue that it is generally impossible to compute ψ^* when there are adversarial agents in the network, since one can never infer their local functions accurately. As an example, suppose that agent n is adversarial and wishes the states to converge to $\bar{\psi} \in \mathbb{R}$. It simply chooses a function $\bar{g}(\psi)$ such that the minimizer of $\sum_{i=1}^{n-1} g_i(\psi) + \bar{g}_n(\psi)$ is $\bar{\psi}$. For a vanishing step size, i.e., $\lim_{t\to\infty} \alpha_t = 0$, all regular agents will asymptotically converge to $\bar{\psi}$ when following the distributed optimization dynamics (16). Since the functions g_i are arbitrary and known only to the agents themselves, such deceptions cannot be detected.

The above fact is formally stated as follows.

Theorem 5 (*Su & Vaidya, 2021; Sundaram & Gharesifard, 2018*)**.** *Suppose Γ is a distributed algorithm that guarantees that all agents calculate the global optimizer of (15). Then a single adversary can cause all agents to converge to any arbitrary value when they run algorithm Γ, and furthermore, will remain undetected.*

Theorem 5 indicates that it is generally impossible to develop an algorithm that always finds optimal solutions and is also resilient to carefully crafted attacks. Thus, the price that is paid for resilient distributed optimization is a *loss in optimality*. It should be noted, however, that the discussion above pertains to scenarios where no further assumption is made on the availability of side information, or on additional structure of the nature of the problem. In turns out that under additional information/problem-structure, one can compute the exact optimal solution in the presence of adversaries. For more on this topic, we refer the reader to the discussion under "Exact Fault Tolerance" in Section 6.2.

In what follows, we describe resilient consensus-based distributed optimization protocols.

Suppose the adversarial agents form an f-local set. At each time step, every regular agent gathers and sorts the states of all of its neighbors and, similar to the W-MSR algorithm in Section 5, each agent disregards the highest f and lowest f states from the gathered states, denoted by set $\mathcal{R}_i[t]$, and updates its state as

$$\psi_i[t+1] = w_{ii}[t]\psi_i[t] + \sum_{j\in\mathcal{N}_i[t]\setminus\mathcal{R}_i[t]} w_{ij}[t]\psi_j[t] - \alpha_t d_i[t], \tag{19}$$

where $d_i[t], \alpha_t$ are the same as (16).

In Sundaram and Gharesifard (2018), graph-theoretic conditions for agents to reach consensus in the presence of f-total malicious agents under the linear filtering rule in (19) are discussed. The arguments rely on the fact that in $(2f+1)$-robust networks, the weight matrix $\boldsymbol{W}[t]$ corresponding to the regular agents is rooted at each time-step, i.e., there is a agent with a directed path to every other agent in the graph induced by the regular agents. Other sufficient conditions (similarly assuming the existence of rooted agents in the set of regular agents $\mathcal{R}$) are presented in Su and Vaidya (2021).

The following result shows that (19) provides a safety guarantee for distributed optimization, i.e., convergence to the interval containing the local minimizers of the regular agents, under certain conditions on the network and the step size α_t.

Theorem 6 (*Sundaram & Gharesifard, 2018*)**.** *Suppose that one of the following conditions holds:*

(i) The adversarial agents are f-total malicious and the network is $(f+1, f+1)$-robust; or
(ii) The adversarial agents are f-local Byzantine and the network is $(2f+1)$-robust.

Furthermore, for each agent $i \in \mathcal{R}$, let the local function $g_i(\cdot)$ have minimizer m_i. Define $\bar{M} = \max\{m_i | i \in \mathcal{R}\}$ and $\underline{M} = \min\{m_i | i \in \mathcal{R}\}$. If stepsizes α_t satisfy $\sum \alpha_t = \infty$ and $\lim_{t\to\infty} \alpha_t = 0$, then $\limsup_{t\to\infty} \psi_i[t] \leq \bar{M}$ and $\liminf_{t\to\infty} \psi_i[t] \geq \underline{M}$ for all $i \in \mathcal{R}$, regardless of the actions of the adversarial agents and the initial values.

Remark 3 (*Lack of Convergence*)**.** Despite the fact that the resilient distributed optimization technique in Theorem 6 guarantees that the states maintained by all regular agents eventually converge to the interval specified by the local minimizers of the regular agents, it does not guarantee convergence to a *constant* value within that interval under certain type of adversarial actions and specific classes of step sizes; see examples in Sundaram and Gharesifard (2018).

6.1. Factors that affect the performance of resilient distributed optimization algorithms

The following example from Sundaram and Gharesifard (2018) shows that under the dynamics (19), the nature of the individual optimization functions together with the network topology determine how far away the convergence point is from the minimizer of the average of the regular agents' functions.

Example 6. Consider network $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ which is $2f+1$ robust and let $\mathcal{T} \subset \mathcal{V}$ be an f-local set. Suppose all agents are regular. Pick an $a \in \mathbb{R}$ and let agents in $\mathcal{T}$ have local functions $g_a(\psi) = (\psi - a)^2$ and the agents in $\mathcal{V}\setminus\mathcal{T}$ have local functions $g_b(\psi) = \psi^2$ (both functions can be modified to have their gradients capped at sufficiently large values, so as to not affect the minimizer of any convex combination of the functions). Let $g(\psi)$ be the average of all of the functions, with minimizer $\psi^* = \frac{|\mathcal{T}|}{n}a$. Then, under the local filtering dynamics (19) with parameter f, all agents converge to the value $\bar{\psi} = 0$ and thus $\bar{\psi} - \psi^* = \frac{|\mathcal{T}|}{n}a$ and $g(\bar{\psi}) - g(\psi^*) = \frac{|\mathcal{T}|^2}{n^2}a^2$.

The above result shows that if the network contains a large f-local set (in relation to the total number of agents) or the local functions have minimizers that are very different (corresponding to a large $|a|$ in the above result), then the value computed by (19) will have a greater divergence from the globally optimal solution. Note that an f-local set in a graph will have size at least equal to f (since any set of size f is f-local).

Table 2
Summary of references on **resilient distributed optimization**. We classify references based on (i) the dimension of the variable ψ in (15), i.e., based on whether ψ is a scalar or a vector; (ii) the graph-structure, i.e., whether the underlying graph is fully-connected (complete) or a general network; and (iii) the nature of the guarantee: here, by exact (resp., approximate), we imply convergence (in a suitable sense) to the desired optimal point (resp., to a neighborhood of the desired optimal point). Exact convergence usually comes at a price: Gupta and Vaidya (2020) and Gupta, Doan, and Vaidya (2021) make certain redundancy assumptions on the functions, Yang and Bajwa (2019) impose an i.i.d. statistical assumption on the agents' data distributions, and Yemini, Nedić, Gil, and Goldsmith (2022) assume the existence of certain indicators of trust.

Attack model	Dimension of variable		Graph topology		Type of guarantee	
	Scalar	Vector	Complete	General	Exact	Approximate
f-total malicious	Sundaram and Gharesifard (2018)	–	–	Sundaram and Gharesifard (2018)	–	Sundaram and Gharesifard (2018)
f-total Byzantine	Su and Vaidya (2021)	Gupta et al. (2021), Gupta and Vaidya (2020), Yang and Bajwa (2019) and Yemini et al. (2022)	Gupta and Vaidya (2020) and Gupta et al. (2021)	Su and Vaidya (2021), Yang and Bajwa (2019) and Yemini et al. (2022)	Gupta et al. (2021), Gupta and Vaidya (2020), Yang and Bajwa (2019) and Yemini et al. (2022)	Su and Vaidya (2021)
f-local Byzantine	Sundaram and Gharesifard (2018)	Kuwaranancharoen, Xin, and Sundaram (2020)	–	Kuwaranancharoen et al. (2020) and Sundaram and Gharesifard (2018)	–	Kuwaranancharoen et al. (2020) and Sundaram and Gharesifard (2018)

6.2. Extension to multi-dimensional functions

In this subsection, we consider the case where the value for each agent is a d-dimensional vector. Thus, the local cost function for agent i becomes $g_i : \mathbb{R}^d \to \mathbb{R}$, and the objective is to collaboratively solve (15) over $\psi \in \mathbb{R}^d$.

The extension to general multi-dimensional functions is a challenging problem as even the region containing the true minimizer of the functions is not easy to characterize. In particular, unlike the scalar case for which the minimizer of $g(\psi)$ lies within the convex hull of the minimizers of the individual functions, for the multi-dimensional case, the true minimizer vector may lie outside the convex hull of the individual minimizer vectors (Kuwaranancharoen & Sundaram, 2018). However, there have been some recent attempts to address the resilient distributed multi-dimensional optimization problem, as we now briefly summarize. Assuming the malicious agents behave in a prescribed manner, Ravi et al. (2019) proposed an attack detection and isolation technique before the execution of the distributed optimization algorithm. In Yang and Bajwa (2019), the authors consider a resilient decentralized machine learning problem, and show that by utilizing a block coordinate descent method, the states of the regular agents will converge to the statistical minimizer with high probability. However, the analysis in Yang and Bajwa (2019) is restricted to i.i.d. training data across the network.

In contrast to Yang and Bajwa (2019), the authors in Kuwaranancharoen et al. (2020) develop a two-step filtering technique and provide convergence guarantees that do not make any statistical assumptions on the agents' objective functions. For an f-local Byzantine attack model, under the assumption that $\mathcal{G}$ is $((2d+1)f+1)$-robust, the approach in Kuwaranancharoen et al. (2020) guarantees asymptotic consensus of the states of all regular agents within a bounded region containing the global minimizer.[12] Simply applying the W-MSR algorithm to each coordinate of the parameter vector does not immediately lead to the above result. Instead, the approach in Kuwaranancharoen et al. (2020) relies on a carefully designed second filtering step.

Exact Fault Tolerance: Instead of settling for convergence to a *proximity* of the global minimizer (as in Kuwaranancharoen et al., 2020; Sundaram & Gharesifard, 2018), one may ask whether it is possible to converge *exactly* to the minimizer of the sum of the objective functions of the regular agents, despite Byzantine attacks. As argued in Sundaram and Gharesifard (2018), this is impossible unless additional assumptions are made on the agents' functions. In this context, the authors in Gupta et al. (2021) show that under a $2f$-redundancy assumption on the agents' objective functions, one can indeed achieve *exact* convergence even in the multi-dimensional case based on a norm filter. In fact, such an assumption turns out to be necessary for guaranteeing exact convergence, as established in Gupta and Vaidya (2020). Yet another avenue for achieving exact fault tolerance is to assume the existence of some source of *trusted* side information; this is the approach pursued in Yemini et al. (2022). The main message here is that in the presence of adversarial agents, one can still retain exact optimality guarantees; however, this comes at the expense of additional side-information or assumptions on the problem-structure.

The analysis in Gupta et al. (2021) is carried out for a complete peer-to-peer network under an f-total Byzantine attack model. Extending the results in Gupta et al. (2021) to general networks remains an open direction of research. Moreover, investigating whether the graph-theoretic conditions in Kuwaranancharoen et al. (2020) can be relaxed is also an interesting open problem.

For a summary of the references surveyed in this section, please see Table 2.

7. Resilient distributed estimation and inference

Another canonical distributed problem involves estimating/tracking an unknown state of interest based on measurements that are collected by a network of sensors. Within this broad setting, there can be several variations: the unknown state may be static or may evolve based on a dynamical model; the measurements may be noise-free or may be corrupted by stochastic noise; and the goal could be to estimate the state asymptotically, or to derive finite-time guarantees.

All of the above variations share a common unifying feature: *new information flows into the network at every time-step.* This distinguishes the distributed estimation setup from the consensus and optimization problems that we discussed earlier, where each node started with a single piece of information (i.e., their initial values or functions), and no additional information subsequently entered into the network from outside. Another important difference stems from the fact that agents typically have *heterogeneous measurement/observation models* in an estimation problem. As

[12] Note that the requirement on the network topology scales with the dimension d of the parameter, as was the case for the resilient vector consensus schemes described in Section 5.3.

a consequence, some agents may be more "informative" than others. This disparity in information content across the network is another key feature that is absent in the standard consensus or distributed optimization formulations. As such, the algorithmic techniques and graph-theoretic conditions that we will cover in this section will differ significantly from those in Sections 5 and 6. Furthermore, the computational complexity involved in assessing the required properties will be different, as we will discuss later in Remark 4.

Before formally discussing resilient estimation algorithms, we outline two important considerations: the nature of the unknown quantity to be estimated, and the nature of the threat model.

Static Parameter Estimation vs. Dynamic State Estimation: As the name suggests, in static parameter estimation, the goal is to estimate a static parameter θ^* based on noisy sensor observations acquired by the agents. In contrast, the task in (dynamic) state estimation is to track a state $\mathbf{x}[t]$ that evolves based on a dynamical system model such as a linear time-invariant (LTI) model. Even in the absence of adversaries, tracking the state of an *unstable* system based on dispersed measurements is a significantly challenging task. Thus, we will discuss the relatively simpler resilient distributed parameter estimation problem first, and then move on to the dynamic state estimation setting.

Sensor Attacks vs. Byzantine Attacks: The works that we will review can also be broadly classified in terms of the threat model. In particular, there are two predominant attack models that are studied in the resilient distributed estimation literature: sensor attacks and Byzantine attacks. In the former case, measurement streams of certain agents are corrupted by an additive attack signal; these signals may or may not be bounded. However, *all* agents behave normally, i.e., they follow the prescribed protocol at all times.[13] In contrast, recall that a Byzantine agent can act *arbitrarily* (i.e., it may not process the sensor information as it supposed to). As we shall see, the nature of the threat model has significant implications for the graph-theoretic properties needed to combat attacks.

7.1. Parameter estimation

In the distributed parameter estimation problem, each agent $i \in \mathcal{V}$ receives measurements as follows:

$$y_i[t] = \mathbf{H}_i\theta^* + n_i[t]. \tag{20}$$

Here, $\theta^* \in \mathbb{R}^d$ is the true unknown parameter, $y_i[t] \in \mathbb{R}^{r_i}$ is the measurement vector for agent i, $\mathbf{H}_i \in \mathbb{R}^{r_i \times d}$ is the local observation matrix for agent i, and $n_i[t]$ is the measurement noise that is typically assumed to independent and identically (i.i.d.) distributed over time, with zero mean and finite variance. Moreover, the noise sequences across different agents are assumed to be independent.

Objective: In the non-adversarial setting, the goal is to design a *consistent* distributed estimator, i.e., an estimator that ensures that the estimates of all agents converge to θ^* asymptotically almost surely. This is typically achieved by designing "consensus + innovations"-type estimators (Kar & Moura, 2011, 2013; Kar, Moura, & Ramanan, 2012; Xie, Choi, Kar, & Poor, 2012) that work under two standard assumptions: (i) the joint observation model is *globally observable*, i.e., $\sum_{i\in\mathcal{V}} \mathbf{H}_i'\mathbf{H}_i$ is invertible, and (ii) the graph $\mathcal{G}$ is *connected*.

In the resilient version of the above problem, a certain subset $\mathcal{A} \subseteq \mathcal{V}$ of the agents is corrupted either due to sensor attacks or due to Byzantine attacks. We now discuss the key algorithmic approaches to tackle such attacks. For each approach, we will focus on highlighting (i) the threat model; (ii) the main technique; (iii) the guarantees provided by the approach; and (iv) the assumptions on the observation model and the underlying graph needed to provide such guarantees.

(1) Methods based on adversary detection: In Chen, Kar, and Moura (2018b), the authors consider a Byzantine attack model, and propose the *Flag Raising Distributed Estimation* (FRDE) algorithm where agents simultaneously perform parameter estimation and adversary detection. Specifically, for parameter estimation, the regular agents employ a consensus+innovations update rule, similar to those in Kar and Moura (2011, 2013), Kar et al. (2012) and Xie et al. (2012). The consensus part of the update rule is based on a weighted average of neighbors' parameter estimates, while the innovation part processes the agent's own local measurements. For adversary detection, an agent computes the Euclidean distance between its own estimate and the estimates of its neighbors. If this distance exceeds a time-varying threshold, then an attack flag is raised. The design of this adaptive threshold constitutes the key part of the FRDE algorithm.

The FRDE algorithm is analyzed under two main assumptions: (i) the joint observation model of the *regular* agents is globally observable, i.e., $\sum_{i\in\mathcal{R}} \mathbf{H}_i'\mathbf{H}_i$ is invertible, where $\mathcal{R} = \mathcal{V} \setminus \mathcal{A}$; and (ii) the *induced* sub-graph $\mathcal{G}_\mathcal{R}$ of the regular agents is connected. Under these assumptions, it is shown in Chen et al. (2018b) that either all regular agents detect the presence of adversaries, or their local estimates converge to θ^* asymptotically almost surely. In other words, "strong" attacks get detected while "weak" attacks fail to disrupt the process of estimation.

Discussion: The assumption that the parameter is globally observable w.r.t. the joint measurements of the regular agents is quite intuitive, and in fact necessary (under the Byzantine attack model). The necessity of the graph condition in Chen et al. (2018b) is, however, an open question. Based on the FRDE algorithm, if the presence of adversaries is detected, the system needs to go through an external "repair" phase; multiple such repair phases could potentially be quite expensive. An alternative is to thus design algorithms that *always* allow the regular agents to estimate the true parameter, despite the presence of adversaries. We now discuss such methods.

(2) Saturating adaptive gain methods: We will discuss this technique in some detail since it has recently been used in the context of resilient distributed state estimation as well (He, Ren, Sandberg and Johansson, 2021). To convey the core idea, we will review the simplest version of this method introduced in Chen, Kar, and Moura (2018a) to tackle sensor attacks. For an agent $i \in \mathcal{A}$ under attack, its measurement model is as follows:

$$y_i[t] = \theta^* + a_i[t], \tag{21}$$

where $a_i[t]$ is the attack signal injected in the measurements of agent i. For agents whose measurements have not been corrupted, the attack signal is identically zero at all times. It is important to note that the measurement model here is *homogeneous* and *noise-free*, and that all uncompromised agents can directly measure θ^*. For this model, the authors in Chen et al. (2018a) propose the *Saturated Innovation Update* (SIU) algorithm where all agents employ a consensus+innovations estimator with a time-varying

[13] It should be noted that sensor attacks are neither malicious nor Byzantine. Whereas malicious and Byzantine agents (in the sense of Definition 1) can be defined for *any* distributed problem over a network, the notion of sensor attacks only makes sense for state estimation/inference problems due to the injection of new information into the network via those sensors. Moreover, unlike malicious/Byzantine agents, when an agent is under a sensor attack, it still continues to transmit whatever it is required to transmit (under a given protocol) accurately.

gain applied to the local innovation term. Specifically, agent i's estimate $x_i(t)$ of θ^* is iteratively updated as

$$x_i[t+1] = x_i[t] - \beta_t \sum_{j\in\mathcal{N}_i} \big(x_i[t] - x_j[t]\big) + \alpha_t K_i[t]\,(y_i[t] - x_i[t])\,, \tag{22}$$

where α_t, β_t, and $K_i[t]$ are strictly positive, scalar-valued design parameters. The time-varying gain $K_i[t]$ is defined as

$$K_i[t] = \begin{cases} 1, & \|y_i[t] - x_i[t]\|_2 \le \gamma_t \\ \frac{\gamma_t}{\|y_i[t]-x_i[t]\|_2} & \text{otherwise,} \end{cases} \tag{23}$$

where γ_t is an adaptive threshold. For a detailed description of how the parameter sequences $\{\alpha_t\}, \{\beta_t\}$, and $\{\gamma_t\}$ are designed, we refer the reader to Chen et al. (2018a). In what follows, we briefly explain why the design of γ_t is a delicate matter. Indeed, if γ_t is chosen to be too small, then the innovation gain $K_i[t]$ will limit the impact of adversaries; however, a very small innovation again may also prevent correct identification of θ^*. On the other hand, if γ_t is too large, then it may provide the adversaries with enough flexibility to direct the agents' estimates away from θ^*. Thus, striking the right balance in the design of γ_t is critical. The following theorem from Chen et al. (2018a) characterizes the performance of the SIU algorithm.

Theorem 7 (*Chen et al., 2018a*). *Suppose the following conditions hold. (i) The graph $\mathcal{G}$ is connected. (ii) The true parameter θ^* is bounded, i.e., $\|\theta^*\|_2 \le \eta$, for some finite η that is known a priori to all agents. (iii) Less than half of the agents are under sensor attack, i.e., $|\mathcal{A}|/|\mathcal{V}| < 1/2$. Then, the parameters α_t, β_t, and γ_t can be designed such that the update rule in (22) ensures*

$$\lim_{t\to\infty} (t+1)^{\tau_0} \|x_i[t] - \theta^*\|_2 = 0, \forall i \in \mathcal{V}, \tag{24}$$

for all τ_0 such that $0 \le \tau_0 < \tau_1 - \tau_2$, where τ_1, τ_2 are design parameters satisfying $0 < \tau_2 < \tau_1 < 1$.[14]

The above theorem tells us that SIU is a consistent estimator, and that the rate of convergence is of the order of $1/t^{\tau_0}$, for any τ_0 satisfying $0 \le \tau_0 < \tau_1 - \tau_2$. Building on the main idea of using an adaptive threshold to design the innovation gain, the authors in Chen et al. (2018a) later generalized their results to account for heterogeneous measurement models corrupted by noise; see Chen, Kar, and Moura (2018c, 2019). In Chen, Kar, and Moura (2020), it was shown that the saturating adaptive gain idea is also effective in the context of resilient distributed field estimation under measurement attacks.

Discussion: We now highlight two subtle implications of the choice of threat model. First, note that the guarantee in Theorem 7 holds for *all* agents, as opposed to just the regular agents. This is the typical guarantee one provides for measurement/sensor attack models. It is instructive to compare such a result with those for the Byzantine setting (see, for instance, Theorem 8) where the goal is to enable only the regular agents to estimate the unknown quantity of interest.

The second key observation pertains to the graph condition in Theorem 7. All that is needed is connectivity of the underlying network — the exact same condition even in the absence of adversaries. Thus, the main takeaway here is that *the graph-theoretic conditions for solving the distributed parameter estimation problem are the same with and without sensor attacks.* The main reason for this can be attributed to the fact that even if an agent's measurements are compromised, it does not try to actively disrupt the flow of information between regular agents; however, a Byzantine agent might. This necessitates much stronger graph-theoretic conditions to tackle Byzantine attacks, as we shall see in Section 7.2.

(3) Methods based on online optimization: Yet another way to approach the distributed parameter estimation problem is to view it from the lens of online optimization. This is precisely the method adopted in Su and Shahrampour (2020), where the authors consider a Byzantine attack model. To explain this method, for each agent $i \in \mathcal{V}$, define its local *asymptotic* loss function $g_i : \mathbb{R}^d \to \mathbb{R}$ as

$$g_i(x) = \frac{1}{2}\mathbb{E}\left[\|\mathbf{H}_i x - y_i\|_2^2\right], \tag{25}$$

where y_i is as in (20), and the expectation is taken w.r.t. the measurement noise $w_i[t]$. Since the distribution of the noise sequence is unknown to agent i, it cannot access the above loss function. Nonetheless, agent i can use all the measurements it has acquired up to each time-step t to compute an *empirical approximation* of $g_i(x)$:

$$g_{i,t}(x) = \frac{1}{2t}\sum_{s=1}^{t} \|\mathbf{H}_i x - y_i[s]\|_2^2. \tag{26}$$

The algorithm in Su and Shahrampour (2020) essentially combines local gradient descent on the above empirical loss functions, followed by coordinate-wise trimming to aggregate neighboring information; trimmed means are used to account for the presence of adversaries. This algorithm enables each regular agent to estimate the true parameter asymptotically almost surely. Moreover, as a departure from existing results on this problem, the authors provide *finite-time concentration bounds* that hold with high probability. To arrive at the above results, the conditions imposed on the graph topology are the same as those for Byzantine-resilient scalar consensus (Vaidya et al., 2012). In Su and Shahrampour (2020), certain additional graph-theoretic assumptions are made on the observation model that may not be necessary.

Additional Results: Before moving on to the dynamic state estimation setting, we briefly comment on a couple of related works. For a somewhat different observation model than in (20), the authors in LeBlanc and Hassan (2014) provide guarantees against Byzantine attacks by drawing on the techniques and graph-conditions in LeBlanc et al. (2013). To the best of our knowledge, this is the earliest work on resilient distributed parameter estimation.

Recently, the authors in An and Yang (2021) proposed a *min-switching* technique to account for the presence of Byzantine agents in the context of least-squares static estimation. The main idea behind the approach in An and Yang (2021) is to first construct an appropriate local Lyapunov function at each regular agent. The filtering technique then comprises of using only those neighboring estimates that lead to maximum decrease of the Lyapunov function. It is shown that this method can help relax the graph-theoretic conditions in both (Mitra & Sundaram, 2019; Su & Shahrampour, 2020).

7.2. Dynamic state estimation and inference

In a typical distributed state estimation problem, the goal is to track the state of a linear time-invariant system of the following form:

$$\boldsymbol{x}[t+1] = \boldsymbol{A}\boldsymbol{x}[t], \tag{27}$$

where $\boldsymbol{x}[t] \in \mathbb{R}^N$ is the state vector and $\boldsymbol{A}$ is the state transition matrix. The system is monitored by a network $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$

[14] In Chen et al. (2018a), the adversarial set $\mathcal{A}$ is allowed to change over time.

consisting of n agents. The measurement model of the ith agent is given by

$$y_i[t] = \boldsymbol{C}_i \boldsymbol{x}[t], \tag{28}$$

where $y_i[t] \in \mathbb{R}^{r_i}$ and $\boldsymbol{C}_i \in \mathbb{R}^{r_i \times N}$. Note that r_i is typically assumed to be smaller than N, and thus each agent only receives a partial measurement of the entire state. We use $\boldsymbol{C} = [\boldsymbol{C}_1', \boldsymbol{C}_2', \ldots, \boldsymbol{C}_N']'$ to collect all the individual agent observation matrices and $\boldsymbol{y}[t] = [y_1'[t], y_2'[t], \ldots, y_N'[t]]'$ to aggregate all the individual measurement vectors; accordingly, $\boldsymbol{y}[t] = \boldsymbol{C}\boldsymbol{x}[t]$. Each agent i maintains an estimate $\hat{\boldsymbol{x}}_i[t]$ of the state $\boldsymbol{x}[t]$, and the goal is to ensure that these estimates converge to $\boldsymbol{x}[t]$ asymptotically.

Even in the absence of adversaries, the distributed state estimation problem is quite challenging, and only recently were necessary and sufficient conditions discovered for this problem (del Nozal, Millán, Orihuela, Seuret, & Zaccarian, 2019; Han, Trentelman, Wang, & Shen, 2019; Kim, Lee, & Shim, 2019; Mitra & Sundaram, 2018; Park & Martins, 2017; Rego, Aguiar, Pascoal, & Jones, 2017; Rego, Pascoal, Aguiar, & Jones, 2019; Wang & Morse, 2018). The key technical challenge arises from the fact that $(\boldsymbol{A}, \boldsymbol{C}_i)$ may not be detectable w.r.t. the measurements of any individual agent i. This is precisely what necessitates communication between agents in the graph. The difficulty of tracking an unstable dynamical process based on dispersed measurements only gets exacerbated in the presence of adversaries. Following the same style of exposition as in Section 7.1, we now discuss the main techniques for resilient distributed state estimation.

(1) Methods Based on Observable Decompositions and Local Filtering: We start by reviewing the approach developed in Mitra and Sundaram (2019) for solving the resilient distributed state estimation problem subject to an f-local Byzantine adversary model. In order to focus on the core ideas behind this approach, we assume that the system matrix $\boldsymbol{A}$ has real and simple eigenvalues; extensions to general spectra can be found in Mitra and Sundaram (2019). As a first step, we diagonalize $\boldsymbol{A}$ using a coordinate transformation matrix $\boldsymbol{V} = [\boldsymbol{v}^1, \boldsymbol{v}^2, \ldots, \boldsymbol{v}^N]$ formed by N linearly independent eigenvectors of $\boldsymbol{A}$. In the new coordinate system where $\boldsymbol{z}[t] = \boldsymbol{V}^{-1}\boldsymbol{x}[t]$ is the state, the dynamics (27) and (28) take the following form:

$$\begin{aligned} \boldsymbol{z}[t+1] &= \boldsymbol{M}\boldsymbol{z}[t], \\ y_i[t] &= \bar{\boldsymbol{C}}_i \boldsymbol{z}[t]. \end{aligned} \tag{29}$$

Here, $\boldsymbol{M} = \boldsymbol{V}^{-1}\boldsymbol{A}\boldsymbol{V}$ is a diagonal matrix with diagonal entries comprising of the distinct eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_N$ of $\boldsymbol{A}$, and $\bar{\boldsymbol{C}}_i = \boldsymbol{C}_i\boldsymbol{V}$. We note that the above transformation is completely independent of the network topology and can be done individually by each agent; it is purely a coordinate transformation on the external system and the local measurement matrix. The only requirement is that all agents use the same coordinate transformation matrix $\boldsymbol{V}$, which can be ensured if they are all programmed with the same algorithm to compute that matrix based on the given matrix $\boldsymbol{A}$.

Based on the above decomposition, each agent i can immediately identify (locally) the set of eigenvalues $\mathcal{O}_i$ that are detectable w.r.t. its own measurements. The key observation made in Mitra and Sundaram (2019) is the following. Each agent i can estimate the components of the state vector $\boldsymbol{z}[t]$ corresponding to its detectable eigenvalues $\mathcal{O}_i$ without interacting with any neighbor. It needs to employ consensus only for estimating those components that correspond to its undetectable eigenvalues $\mathcal{UO}_i$. Specifically, every regular agent i employs the following scheme.

(i) For each $\lambda_j \in \mathcal{O}_i$, agent i uses a standard Luenberger observer to estimate $z^{(j)}[t]$ — the component of $\boldsymbol{z}[t]$ corresponding to λ_j.

(ii) For each $\lambda_j \in \mathcal{UO}_i$, agent i uses a "local filtering" technique to estimate $z^{(j)}[t]$.

We now elaborate on item (ii). Let $\Lambda_U(\boldsymbol{M})$ represent the set of all unstable and marginally stable eigenvalues of $\boldsymbol{M}$. Moreover, for each $\lambda_j \in \Lambda_U(\boldsymbol{M})$, let $\mathcal{S}_j$ be the set of agents that can detect λ_j; we will refer to $\mathcal{S}_j$ as the set of source agents for mode λ_j. To enable the agents in $\mathcal{V} \setminus \mathcal{S}_j$ to estimate $z^{(j)}[t]$, the following two requirements (stated loosely here, and more formally later) turn out to be critical.

- **Information redundancy:** The set $\mathcal{S}_j$ needs to be sufficiently large. Otherwise, if the sources of information for the mode λ_j are corrupted by too many adversaries, then the other agents cannot hope to recover the correct information about that mode.
- **Network-structure redundancy:** There must exist sufficient disjoint paths in $\mathcal{G}$ that link the agents in $\mathcal{S}_j$ to those in $\mathcal{V} \setminus \mathcal{S}_j$. Otherwise, the adversaries can form a bottleneck and disrupt the flow of information from $\mathcal{S}_j$ to $\mathcal{V} \setminus \mathcal{S}_j$.

In Mitra and Sundaram (2019), the authors introduce a graph-theoretic construct called the Mode Estimation Directed Acyclic Graph (MEDAG) to capture the above requirements. Essentially, a MEDAG $\mathcal{G}_j$ for mode λ_j is a subgraph of $\mathcal{G}$ that provides a secure *uni-directional* medium of information-flow from $\mathcal{S}_j$ to $\mathcal{V} \setminus \mathcal{S}_j$.[15] The uni-directional aspect is important to ensure stability of the estimation error dynamics; it has nothing to do with adversaries. Once a MEDAG $\mathcal{G}_j$ has been constructed for each $\lambda_j \in \Lambda_U(\boldsymbol{M})$, an agent $i \in \mathcal{V} \setminus \mathcal{S}_j$ uses the estimates of only its neighbors $\mathcal{N}_i^{(j)}$ in the MEDAG $\mathcal{G}_j$ to update its estimate $\hat{z}_i^{(j)}[t]$ of $z^{(j)}[t]$ as follows:

$$\hat{z}_i^{(j)}[t+1] = \lambda_j \sum_{\ell \in \mathcal{M}_i^{(j)}[t]} w_{i\ell}^{(j)}[t] \hat{z}_\ell^{(j)}[t]. \tag{30}$$

In the above update rule, $\mathcal{M}_i^{(j)}[t] \subset \mathcal{N}_i^{(j)} \subseteq \mathcal{N}_i$ is the set of those neighbors from whom agent i accepts estimates of $z^{(j)}[t]$ at time-step t, after removing the f largest and f smallest estimates of $z^{(j)}[t]$ from $\mathcal{N}_i^{(j)}$; the properties of a MEDAG $\mathcal{G}_j$ ensure that $\mathcal{M}_i^{(j)}[t]$ is always non-empty. The weights in (30) are non-negative and chosen to satisfy $\sum_{\ell \in \mathcal{M}_i^{(j)}[t]} w_{i\ell}^{(j)}[t] = 1$. The overall approach we described above is called the Local-Filtering based Resilient Estimation (LFRE) algorithm in Mitra and Sundaram (2019).

To analyze the performance of the LFRE algorithm, we need to first understand when a given network $\mathcal{G}$ contains a MEDAG. The following graph-theoretic property is what we need in this context.

Definition 7 (*Strongly r-robust Graph w.r.t. $\mathcal{S}$*)**.** Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a positive integer $r \in \mathbb{N}_+$, and a non-empty set $\mathcal{S} \subset \mathcal{V}$, $\mathcal{G}$ is *strongly r-robust w.r.t. the set of agents $\mathcal{S}$* if for any non-empty subset $\mathcal{C} \subseteq \mathcal{V} \setminus \mathcal{S}$, $\mathcal{C}$ is r-reachable.

For an illustration of the above definition, consider the setup in Fig. 9 where a scalar unstable plant is monitored by a network of agents. Agents 1, 2, and 3 are the source agents for this system, i.e., $\mathcal{S} = \{1, 2, 3\}$. The graph on the left in Fig. 9 is an example of a network that is strongly 3-robust w.r.t. the set of source agents $\mathcal{S}$. Specifically, all subsets of $\{4, 5, 6, 7\}$ are 3-reachable (i.e., each such subset has an agent that has at least 3 neighbors outside that subset). The graph on the right is an example of a MEDAG. The next result exemplifies the role played by *strong-robustness* in resilient distributed state estimation.

[15] For a precise description of the properties of a MEDAG, see Mitra and Sundaram (2019), where a distributed algorithm is provided to construct such subgraphs.

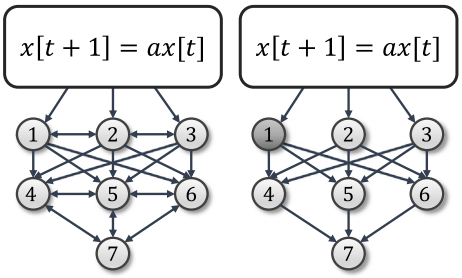


Fig. 9. (**Left**) A scalar unstable plant is monitored by a network of 7 agents. The network is strongly 3-robust w.r.t. the set of source agents $\mathcal{S} = \{1, 2, 3\}$. (**Right**) A subgraph of the original graph satisfying the properties of a MEDAG.

Theorem 8 (*Mitra & Sundaram, 2019*)**.** *Suppose $\mathcal{G}$ is strongly $(2f+1)$-robust w.r.t. $\mathcal{S}_j, \forall \lambda_j \in \Lambda_U(\boldsymbol{A})$. Then, the LFRE algorithm guarantees $\lim_{t\to\infty} \|\hat{\boldsymbol{x}}_i[t] - \boldsymbol{x}[t]\| = 0$ for every regular agent i, despite the presence of any f-locally bounded set of Byzantine adversaries.*

Discussion: While the above result provides a sufficient condition for tolerating Byzantine adversaries in the context of distributed state estimation, separate necessary conditions are also identified in Mitra and Sundaram (2019). These necessary conditions are based on two key ingredients: (i) critical information sets: sets of agents whose measurements *have to be utilized* for observing the state, i.e., if such agents were to be removed from the network, the system would lose observability; and (ii) cuts in the network that separate such critical informative agents from the rest of the network. For a precise definition of the above concepts, we refer the reader to Mitra and Sundaram (2019, Section 3). Thus, the necessary conditions in Mitra and Sundaram (2019) turn out to be a blend of requirements on both the observation model and the network structure, and generalize the conditions for centralized state estimation subject to attacks (Chong, Wakaiki, & Hespanha, 2015; Fawzi et al., 2014).

The techniques in Mitra and Sundaram (2019) were later generalized to account for time-varying networks in Mitra, Richards, Bagchi and Sundaram (2019, 2021). Moreover, in Mitra, Ghawash, Sundaram and Abbas (2021), the authors formally showed how one can incorporate the ideas of trust (i.e., making a small subset of the agents immune to attacks) and diversity (in hardware and software implementations) to relax some of the stringent redundancy requirements and make sparse DCSs as resilient to adversarial attacks as dense networks (Mitra & Sundaram, 2019).

One important takeaway is that the ideas of information redundancy and network-structure redundancy are quite general, and as such, applicable beyond the specific estimation problem we considered here. Indeed, we will later briefly comment on the fact that these ideas also turn out to be crucial in the context of resilient distributed hypothesis testing/statistical inference.

Remark 4. It is worth noting the difference between the strong r-robustness property in Definition 7 and the r-robustness property in Section 2.1.2 that was used for resilient consensus and optimization. In the former, one has to account for the fact that some nodes in the network are more informative than the others, vis-à-vis the quantity that all agents are trying to estimate. Thus, one has to ensure that information can flow reliably from that subset of nodes to the others in the network (capturing the information and structural redundancy requirements discussed earlier). This is precisely the reason why the strong r-robustness property is defined with respect to a source set $\mathcal{S}$. On the other hand, for problems like distributed consensus and optimization where *every* node has information that is pertinent to the final quantity, there is no designated set of source nodes. Thus, the r-robustness condition is defined with respect to the graph as a whole.

It turns out that this subtle difference has a significant implication for the computational complexity of checking whether a given graph has those properties. In particular, it was shown in Zhang, Fata, and Sundaram (2015) that checking whether a given graph is r-robust is coNP-complete. However, the strong r-robustness property in Definition 7 can be checked in polynomial-time (Mitra & Sundaram, 2019).

(2) Methods Based on Robust Control Theory: In some papers, distributed state estimation of a continuous-time LTI system has been considered where both the state and measurement dynamics are subject to L_2-integrable disturbances (Deghat, Ugrinovskii, Shames, & Langbort, 2019). Given the nature of the disturbances, the authors build on the theory of distributed H_∞ filters developed in Ugrinovskii (2011) and Ugrinovskii and Langbort (2011). The threat model is that of a *biasing attack* where an attacker injects an additive attack signal directly to the state estimator/observer dynamics of certain agents.

To tackle such biasing attacks, the authors in Deghat et al. (2019) develop certain attack detection filters. Specifically, the attack detection filter at each agent i takes as input two different innovation signals. The first innovation signal is the gap between the actual measurement $y_i[t]$ and the predicted sensor measurement $C_i\hat{x}_i[t]$. Intuitively, if agent i's observer dynamics is compromised, then one should expect $C_i\hat{x}_i[t]$ to be biased, leading to a significant deviation from $y_i[t]$.[16] With a similar motivation, a second innovation signal is computed based on the deviations of agent i's state estimate from those of its neighbors. It is shown that the problem of designing the parameters of the above detector can be recast as the problem of stabilizing a distributed dynamical system via output injection. The latter problem is addressed by drawing on ideas from vector dissipativity theory (Ugrinovskii, 2011; Ugrinovskii & Langbort, 2011). Since the overall design procedure is quite intricate, we refer the reader to Deghat et al. (2019) for details.

As their main result, the authors in Deghat et al. (2019) show that if certain LMI's are feasible, then their approach ensures attack-detection and guarantees a desired level of H_∞ disturbance attenuation. The requirements on the network structure are implicitly captured by the LMI's.

Discussion: It is instructive to compare the results in Mitra and Sundaram (2019) with those in Deghat et al. (2019). In Mitra and Sundaram (2019), accounting for a worst-case Byzantine attack model necessitates the requirement of sufficient redundancy in the underlying network, as captured by the "strong robustness" condition in Definition 7. Moreover, for the problem to be meaningful in Mitra and Sundaram (2019), only a subset of the agents can be adversarial. In contrast, since a specific class of biasing attacks is considered in Deghat et al. (2019), the network requirements are relatively less stringent, and it is plausible for the entire network to be under attack.

Following up on Deghat et al. (2019), more general biasing attacks are considered in Ugrinovskii (2019) where an attacker is allowed to bias both the state observer and the attack detector at an agent. Moreover, unlike Deghat et al. (2019) where the design of the filter gains involves solving LMI's that are coupled across

[16] As in Deghat et al. (2019), the innovation signal $(y_i[t] - H_i\hat{x}_i[t])$ plays a key role in the saturating adaptive gain method of Chen et al. (2018a). However, while $y_i[t]$ is accurate and $C_i\hat{x}_i[t]$ is biased in Deghat et al. (2019), the situation is exactly the opposite in Chen et al. (2018a), where the measurements are biased but the agents' estimators are not.

agents, the design procedure is carried out locally at every agent in Ugrinovskii (2019).

(3) Methods Based on Saturating Gains and Attack Detection: Recently, in He, Ren et al. (2021), the authors developed secure distributed filters for tackling measurement attacks. Their main approach relies on a saturating adaptive gain technique, similar to Chen et al. (2018a). There are, however, considerable differences with Chen et al. (2018a) that stem from the fact that He, Ren et al. (2021) consider dynamical state estimation while Chen et al. (2018a) study static parameter estimation. One such difference is that He, Ren et al. (2021) employ a two-time-scale estimation technique: between two consecutive time-steps of the dynamics, the agents are allowed to perform multiple consensus steps to bridge the gap between their estimates. Under reasonable assumptions on observability, and connectivity of the graph, the estimation error is shown to be uniformly bounded.

Discussion: Notably, the analysis in He, Ren et al. (2021) applies to *time-varying adversarial sets*. When the adversarial set is fixed, an attack detection algorithm is further developed in He, Ren et al. (2021) that leads to tighter error bounds. It should be noted, however, that the two-time-scale approach is crucial to the stability of the distributed filter in He, Ren et al. (2021).

Remark 5. In some very recent work, the resilient distributed state estimation problem from a dynamic average consensus perspective has been studied (Mao & Tabuada, 2021). In Mao, Diggavi, Fragouli, and Tabuada (2020), the setting where some of the communication links can also be corrupted (in addition to agent attacks) has been explored.

A practical problem in Resilient Distributed Estimation: One important open question pertaining to graph theoretic methods in distributed estimation is how to jointly control the graph and the dynamics in order to maintain the desired resilience properties. For instance, in a multi robot system, the communication graph may be determined by proximity of the robots. This necessitates that the agents stay sufficiently close together (over time) in order to ensure that the resulting communication graph has the required robustness property. However, the estimation task may also require the agents to visit different locations in the environment in order to obtain sensor measurements of the dynamical process that they are monitoring. These two requirements (maintaining network properties while gathering enough information from the environment) may be at odds with one another, and thus finding principled approaches to solving such problems is a rich avenue for future research.

7.3. Hypothesis testing and statistical inference

In this subsection, we will briefly discuss an approach for tackling Byzantine attacks in the context of distributed hypothesis testing (Jadbabaie, Molavi, Sandroni, & Tahbaz-Salehi, 2012; Jadbabaie, Molavi, & Tahbaz-Salehi, 2013; Lalitha, Javidi, & Sarwate, 2018; Liu, Fang, Wang, & Wang, 2014; Mitra, Richards and Sundaram, 2019; Mitra, Richards, & Sundaram, 2020; Nedić, Olshevsky, & Uribe, 2017; Shahrampour, Rakhlin, & Jadbabaie, 2016; Su & Vaidya, 2016; Uribe, Hare, Kaplan, & Jadbabaie, 2019) — a problem similar in flavor to the static parameter estimation setting we considered in Section 7.1. In this problem, each agent in a network receives a sequence of stochastic measurements generated by a common underlying distribution that is parameterized by an *unknown*, *static* state θ^*. Each agent is equipped with a local likelihood model, and is aware that θ^* belongs to a finite set $\Theta = \{\theta_1, \ldots, \theta_m\}$ of m candidate hypotheses. The goal is for the agents to collaboratively identify θ^* from Θ. The challenge arises from the fact that θ^* may not be locally identifiable w.r.t. the likelihood model of any specific agent. In other words, no one agent can, in general, eliminate every false hypothesis on its own. Instead, we assume *global identifiability* of the joint observation model, i.e., θ^* can be uniquely identified based on the collective observations of the agents.[17]

In a typical approach to solving the above problem, each agent maintains a belief vector $\boldsymbol{\mu}_{i,t}$ which is a distribution over Θ. Formally, the objective is to design belief-update and propagation rules that ensure $\mu_{i,t}(\theta^*) \to 1, \forall i \in \mathcal{V}$, almost surely. The predominant approach is to employ some form of "belief-averaging" protocol to update the belief vectors (Jadbabaie et al., 2012, 2013; Lalitha et al., 2018; Liu et al., 2014; Nedić et al., 2017; Shahrampour et al., 2016; Su & Vaidya, 2016; Uribe et al., 2019). In a departure from these algorithms, a min-rule was recently developed in Mitra et al. (2020); the asymptotic learning rate of this rule is strictly better than those based on averaging. However, all of the above approaches are vulnerable in the face of attacks: a single malicious agent can essentially cause all good agents to eliminate the true hypothesis.

The Min-Rule: A desirable feature of the min-rule in Mitra et al. (2020) is that it admits a simple, computationally-efficient extension that is robust to worst-case attacks. We first describe the basic min-rule, and then its adversarial extension. Each agent $i \in \mathcal{V}$ maintains an auxiliary local belief vector $\boldsymbol{\pi}_{i,t}$ that is updated in a Bayesian manner based on just the observations of agent i. For every false hypothesis $\theta \in \Theta \setminus \{\theta^*\}$ that agent i can eliminate on its own, we will have $\pi_{i,t}(\theta) \to 0$ almost surely. Thus, agent i only needs to interact with neighbors for eliminating those false hypotheses that it cannot rule out on its own. Let $\mathcal{S}(\theta^*, \theta)$ be those agents that can distinguish between θ^* and θ, i.e., these agents can eliminate θ individually. The main idea is to transmit low beliefs on θ from agents in $\mathcal{S}(\theta^*, \theta)$ to the rest of the network. This is achieved via the following rule at each agent i:

$$\mu_{i,t+1}(\theta) \propto \min\{\{\mu_{j,t}(\theta)\}_{j\in\mathcal{N}_i\cup\{i\}}, \pi_{i,t+1}(\theta)\}. \tag{31}$$

The above beliefs are normalized to ensure that $\boldsymbol{\mu}_{i,t}$ is a valid distribution at every time-step. For the adversarial setting, the approach is very similar, except that the min-rule is applied to a set of *moderate beliefs*. Specifically, each regular agent i updates its belief on a state θ by first rejecting the highest f and lowest f beliefs on θ received from $\mathcal{N}_i$, and then employing:

$$\mu_{i,t+1}(\theta) \propto \min\{\{\mu_{j,t}(\theta)\}_{j\in\mathcal{M}^{\theta}_{i,t}}, \pi_{i,t+1}(\theta)\}, \tag{32}$$

where $\mathcal{M}^{\theta}_{i,t}$ are those agents that do not get rejected in the above filtering step. This is known as the Local-filtering based Resilient Hypothesis Elimination (LFRHE) algorithm. The correctness of the LFRHE algorithm once again rests on the two key ingredients we identified in Section 7.1, namely, information-redundancy and network-structure redundancy. In particular, for every pair θ_p, θ_q, we need $\mathcal{S}(\theta_p, \theta_q)$ to be large enough, and we also need a sufficient number of disjoint paths from $\mathcal{S}(\theta_p, \theta_q)$ to $\mathcal{V}\setminus\mathcal{S}(\theta_p, \theta_q)$. These requirements are succinctly captured in the following theorem.

Theorem 9 (*Mitra et al., 2020*)**.** *Suppose that for every pair of hypotheses $\theta_p, \theta_q \in \Theta$, the graph $\mathcal{G}$ is strongly $(2f+1)$-robust w.r.t. the source set $\mathcal{S}(\theta_p, \theta_q)$. Moreover, suppose each regular agent i has a non-zero prior belief on every hypothesis, i.e., $\pi_{i,0}(\theta) > 0$ and $\mu_{i,0}(\theta) > 0, \forall \theta \in \Theta$. Then, the LFRHE algorithm guarantees that $\mu_{i,t}(\theta^*) \to 1$ almost surely for every regular agent i, despite the actions of any f-local set of Byzantine adversaries.*

[17] Global identifiability for distributed hypothesis testing is the exact analogue of global observability for distributed state estimation.

Table 3
Summary of references on **resilient distributed estimation and inference**. We have classified these references based on three aspects. First, we consider the specific nature of the problem being studied: parameter estimation, state estimation, and hypothesis testing/statistical inference. Second, we consider the nature of the attack model: sensor attacks (resp., biasing attacks) are ones where the measurements (resp., estimator/observer update equations) are corrupted by additive attack inputs, and Byzantine attacks are as in Definition 1. Third, the references can be broadly classified in terms of the main algorithmic approaches: saturating adaptive gain and attack-detection techniques (in **bold**), local filtering-based approaches (with underline), and robust control techniques (references Deghat et al., 2019; Ugrinovskii, 2019 in gray). A finer description of these approaches is provided in Section 7.

Distributed problem	Attack model		
	Sensor attacks	Biasing attacks	Byzantine attacks
Static parameter estimation	**Chen et al. (2018a, 2018c, 2019)**	–	An and Yang (2021), **Chen et al. (2018b)**, LeBlanc and Hassan (2014), Su and Shahrampour (2020)
Dynamic state estimation	**He, Ren et al. (2021)**	Deghat et al. (2019), Ugrinovskii (2019)	Mitra and Sundaram (2019)
Hypothesis testing/Statistical inference	–	–	Mitra et al. (2020), Su and Vaidya (2016) , Wu, Carr, Bharadwaj, Xu, and Topcu (2020)

Discussion: One of the main takeaways from the above result is that just like the resilient distributed estimation problem, the strong-robustness property in Definition 7 ends up playing a crucial role when it comes to tolerating Byzantine attacks for distributed hypothesis-testing as well. We conjecture that this graph-theoretic property will prove to be useful for other distributed learning problems where information is diffused across the network. A recent work that studies resilient distributed best-arm identification for stochastic multi-armed bandits supports this conjecture (Mitra, Hassani and Pappas, 2021).

Additional Results: The algorithm in Mitra et al. (2020) was later extended in Wu et al. (2020) to account for time-varying networks. In Su and Vaidya (2016), the authors proposed an alternate approach to tackling adversaries by building on the log-linear belief-update rule in Lalitha et al. (2018), Nedić et al. (2017) and Shahrampour et al. (2016). Their approach requires the agents to compute Tverberg partitions (see Section 5.3); however, as discussed in that section, there is no known algorithm that can compute an exact Tverberg partition in polynomial time for a general d-dimensional finite point set (Mulzer & Werner, 2013).

For a summary of the references we discussed in this section, we refer the readers to Table 3.

8. Attack detection and identification over networks

In the preceding sections, we covered several techniques for solving a variety of distributed information-processing problems subject to attacks (e.g., consensus, optimization, and estimation). As discussed in those sections, depending on the attack model and nature of the information available to the agents, detection and identification of (worst-case) adversarial behavior may be impossible in general. In particular, the "local-filtering" algorithms discussed in those sections did not explicitly rely on detection/identification of adversarial behavior. However, in other settings, detection and identification of adversarial behavior may indeed be possible. The purpose of this section is to briefly summarize algorithms for such settings.

Since the precise nature of the attack detection algorithm is usually dictated by the specific distributed task at hand, we will not be able to cover all such detection mechanisms here. Instead, we will primarily restrict our attention to the attack model in (9) of Section 4.2 that we studied in the context of distributed function calculation. Using this model, we will discuss graph-theoretic requirements for detecting and identifying attacks in a network. In particular, we will demonstrate how *structured systems theory* plays a key role in this context. Before delving into the technical details, we remind the reader that the f-total attack model under consideration involves a set of malicious agents $\mathcal{F}$, where $|\mathcal{F}| \leq f$ for a known $f \geq 0$.

8.1. Attack detection

Centralized and distributed detection techniques can be used to detect attacks, see Pasqualetti et al. (2013). Here, we characterize graph-theoretic conditions for detecting attacks. We start by considering that the initial states, $\boldsymbol{\psi}[0]$, are known. An attack vector $\boldsymbol{\zeta}$ is called *undetectable* or *perfect* if $\boldsymbol{y}(\boldsymbol{\psi}[0], \boldsymbol{\zeta}, t) = \boldsymbol{y}(\boldsymbol{\psi}[0], 0, t)$ for all $t \geq 0$, i.e., the measurement is the same as the case of no attack. The notion of a perfect attack has an equivalent algebraic condition, which is based on the following definition.

Definition 8. The generic normal rank (gnr) of the matrix pencil of dynamics (9)

$$\boldsymbol{P}(z) = \begin{bmatrix} \boldsymbol{W} - z\mathbf{I}_n & \boldsymbol{B}_{\mathcal{F}} \\ \boldsymbol{C} & \mathbf{0} \end{bmatrix},$$

is the maximum rank of the matrix over all choices of free parameters in $(\boldsymbol{W}, \boldsymbol{B}_{\mathcal{F}}, \boldsymbol{C})$ and $z \in \mathbb{C}$.[18]

It is shown in Dion et al. (2003) that having a perfect attack and the generic normal rank of $\boldsymbol{P}(z)$ being less than $n + |\mathcal{F}|$ are equivalent. Recalling the input set $\mathcal{U}$ and measurement set $\mathcal{Y}$ for a structured system from Section 2.4, the following result interprets the generic normal rank of $\boldsymbol{P}(z)$ in terms of the disjoint paths in the graph of structured system $\mathcal{G}$.

Lemma 1 (*Dion, Commault, & van der Woude, 1999*)**.** *The generic normal rank of the matrix pencil $\boldsymbol{P}(z)$ is equal to $n + r$, where r is the size of the largest linking in $\mathcal{G}$ from the input vertices, $\mathcal{U}$, to the output vertices, $\mathcal{Y}$.*

Note that the generic normal rank of the matrix pencil is at least n, since the matrix $\boldsymbol{W} - z\mathbf{I}_n$ will have generic rank n for any choice of parameters in $\boldsymbol{W}$ and any z that is not an eigenvalue of $\boldsymbol{W}$. From Dion et al. (2003) and Lemma 1 it follows that to prevent perfect attacks, parameter r has to be equal to the number of attacks, i.e., $r = |\mathcal{F}|$. This implicitly indicates that the number of sensors must be at least $|\mathcal{F}|$. Lemma 1 along with Menger's theorem and the Expansion lemma, cf. West (2001b), yields the following graph-theoretic result on attack detectability.

Theorem 10. *Suppose that dynamics (9) with measurement (7) is subject to a set of f attacked agents and that the initial states, $\boldsymbol{\psi}[0]$, are known. To prevent a perfect attack, it is sufficient for graph $\mathcal{G}$ to be $(f + 1)$-connected.*

[18] By a generic property of a structured system, we mean a property that holds for almost all values of the free (nonzero) parameters of the system, i.e., all values except those that belong to a proper algebraic variety of Lebesgue measure zero in the parameter space.

Remark 6. As the number of sensors is often limited and the underlying network may be sparse, detecting all attacked agents may not be always possible. An alternative approach is to place the available sensors on key agents in the network in order to maximize r in Lemma 1, i.e., detecting maximum number of attacks. The sensor placement problem for optimal attack detection is discussed in Section 10.2.

For the cases where the initial condition of the system is unknown, an undetectable attack $\boldsymbol{\zeta}$ is characterized by the existence of a pair of initial states $\boldsymbol{\psi}_1[0]$ and $\boldsymbol{\psi}_2[0]$ such that $\boldsymbol{y}(\boldsymbol{\psi}_1[0], 0, t) = \boldsymbol{y}(\boldsymbol{\psi}_2[0], \boldsymbol{\zeta}, t)$ for all $t \geq 0$. In such cases, one needs to first recover the initial conditions of the system in order to detect the attack. This demands the system to be strongly observable. Recall from Section 4 and Theorem 3 that system (9) is strongly observable if the graph is $(2f+1)$-connected.

Theorem 11. *Suppose that dynamics (9) with measurement (7) is subject to a set of f attacked agents and unknown initial conditions. To prevent a perfect attack, it is sufficient for graph $\mathcal{G}$ to be $(2f+1)$-connected.*

The extra level of graph connectivity stated in Theorem 11, compared to Theorem 10, is a price paid for the lack of the knowledge of the initial states.

8.2. Attack identification procedure

In order to identify the attacked agents, first, agent i must find the true initial value of all other agents (e.g., via the procedure described in Section 4.2 and under the conditions in Theorem 3). After obtaining the vector of initial states, $\boldsymbol{\psi}[0]$, and assuming that the interaction matrix, $\boldsymbol{W}$, is known to i, it can apply dynamics (9) to obtain $\boldsymbol{\psi}[1] - \boldsymbol{W}\boldsymbol{\psi}[0] = \boldsymbol{B}_{\mathcal{F}}\boldsymbol{\zeta}[0]$. Every nonzero component in the vector on the left hand side of this equation indicates an additive error injected by the corresponding agent. Thus, every agent that is malicious during time-step 0 can be identified by this method. The same process can be repeated to find all agents that were malicious during the first L time steps from the transmitted values $y_i[0:L]$ in (10). Note that using iteration policy (9) to identify the attacks requires the system to be free of noise or external disturbances. Further details on centralized and distributed attack detection and identification techniques can be found in Pasqualetti et al. (2013).

8.3. Other attack detection and identification approaches

Our discussion in this section has thus far focused on the attack model in (9) for distributed function calculation. Before closing this section, we briefly summarize certain other attack detection mechanisms that are relevant in other contexts. For resilient distributed parameter estimation, attack detection constitutes a key component of the FRDE algorithm in Chen et al. (2018b) that we discussed in Section 7.1. For state estimation, agents under attack are identified based on robust control techniques in Deghat et al. (2019), and a saturating adaptive gain method in He, Ren et al. (2021); see Section 7.2. At a high-level, the approaches in Chen et al. (2018b), Deghat et al. (2019), and He, Ren et al. (2021) share a common principle for detecting whether an agent i is under attack: they involve computing an appropriate innovation signal that captures the extent to which the estimates/measurements of agent i differ from those of its neighbors. For distributed optimization, Ravi et al. (2019) proposed a heuristic gradient based method to detect the misbehaving agents. The insight is that the attackers' biasing actions can result in a large gradient value. Hence, a regular agent can attempt to detect such attacks by approximating the gradient of each neighbor and tracking it over time relative to the mean of the gradients of the remaining neighbors. Finally, we note that in the context of multi-robot coordination, the authors in Gil, Kumar, Mazumder, Katabi, and Rus (2017) and Renganathan and Summers (2017) propose methods to tackle the so called "Sybil attack", where an attacker spoofs or impersonates the identities of existing agents to gain a disproportionate advantage in the network. The key idea in these works is to detect such spoofing attacks by exploiting the physics of wireless signals. For specific details, we refer the reader to the respective papers.

9. Graph-theoretic interpretations of the attack impact

In the previous sections, we primarily focused on the topic of attack mitigation for different classes of problems, and noted that the connectivity of the graph plays a key role in this regard. In this section, we will instead turn our attention to the attacker, and quantify the attacker's impact on distributed control systems in terms of the topology of the underlying network.

9.1. Controllability of networks under attack

One way to define the attacker's impact is via the largest subset of agents which can be controlled (or reached) by the set of the attacked agents. There is a vast literature that studies the controllability of networks with a limited number of actuators (Dion et al., 2003; Olshevsky, 2014; Pequito, Kar, & Aguiar, 2016; Rahmani, Ji, Mesbahi, & Egerstedt, 2009; Sundaram & Hadjicostis, 2013). In this section, our focus is on structural controllability as a qualitative measure of controllability and Gramian-based methods as quantitative measures.

9.1.1. Structural and strong structural controllability

As mentioned in Section 2.4, a structured system is said to be controllable if this property holds for at least one numerical choice of free parameters in the system. Theorem 1 provided graph-theoretic conditions for structural controllability. Condition (i) in Theorem 1 is called the reachability condition. The reachability, by itself, can be a measure of the attack impact. In particular, when the attacker's goal is to disseminate a signal throughout the network and infect as many agents as possible, maximizing reachability is beneficial to the attacker. On the other hand, when the attacker's objective is to steer the states towards its desired direction, controllability is the appropriate measure. We present the following definition.

Definition 9. The reachable set, $\mathcal{S}_f^r$ (respectively controllable set, $\mathcal{S}_f^c$), is the largest subset of agents in a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ which can be reached (controlled) by any configuration of f attacked agents in the network.

Based on Theorem 1, it is clear that $|\mathcal{S}_f^c| \leq |\mathcal{S}_f^r|$. Fig. 10(a) shows a graph whose largest reachable set, for $f = 1$, is larger than the largest controllable set. Based on the definition of structural controllability, if we add a self-loop to agent 3 or 4 in this graph, then all agents become controllable. In graph (b) the largest reachable and controllable sets are identical. Note that $\mathcal{S}_1^c$ is not necessarily a subset of $\mathcal{S}_1^r$, as shown in graphs (c) and (d).

The source agents are shown with a darker color. Due to resource constraints, the attacker naturally tries to solve either of the following problems:

(i) Controlling (or reaching) the *largest* possible subset of agents in the network with a given number of attacked agents, or

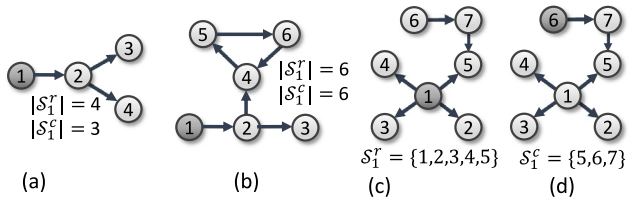


Fig. 10. Graphs and their largest reachable subsets, $\mathcal{S}^r$, and controllable subsets, $\mathcal{S}^c$. Dark colors represent attack agents.

(ii) Controlling (or reaching) the whole network with the *minimum* number of attacked agents.

The latter has been investigated under the context of minimal structural controllability problems (Pequito, Kar, & Aguiar, 2015). It is shown that the problem of finding the minimum number of control input (attacked) agents is in general NP-hard, but in specific cases (such as dedicated inputs) can be solved in polynomial time. When reachability is the objective, the problem is related to estimating the reachable set of agents which can be solved in polynomial time (Aji, 2014).

A system is called strong structurally controllable if rank $(\boldsymbol{W}, \boldsymbol{B}) = n$ for *all* (nonzero) choice of free parameters in $\boldsymbol{W}$ and $\boldsymbol{B}$. The dimension of the strong structurally controllable subspace is the minimum rank of the controllability matrix, $\mathcal{C}(\boldsymbol{W}, \boldsymbol{B})$. There are graph-theoretic bounds on this quantity for consensus dynamics as stated below.

Theorem 12 (*Yazıcıoglu, Abbas, & Egerstedt, 2016*)**.** *For set $\mathcal{F}$ chosen by the attacker, the dimension of the strong structurally controllable subspace is lower bounded by*

$$\operatorname{rank} \mathcal{C}(\boldsymbol{W}, \boldsymbol{B}) \geq \max_{i \in \mathcal{V}, j \in \mathcal{F}} \operatorname{dist}(i, j) + 1, \tag{33}$$

where $\operatorname{dist}(i, j)$ is the shortest distance between agents i and j. Tighter lower bounds for the dimension of the strong structurally controllable subspace can be found in Mousavi, Haeri, and Mesbahi (2018) and Yazıcıoglu et al. (2016). According to Theorem 12, from the attacker's perspective, the optimal decision is to select an agent with maximum distance from the rest of the agents in the graph. Another interpretation from the above result, which had been discussed before in Rahmani et al. (2009), is the reverse effect of network connectivity on the controllability. Specifically, sparse networks may contain pairs of agents that are far apart, and consequently may have large controllability subspaces, while well-connected networks may have smaller controllability subspaces. More recently, graph-theoretic conditions for strong structural controllability have been proposed in Jia, van Waarde, Trentelman, and Camlibel (2021). In particular, necessary and sufficient conditions for strong structural controllability in terms of full rank tests of certain pattern matrices have been presented.

9.1.2. Gramian-based controllability

Unlike the (discrete) structural or rank-based controllability measures of dynamical systems (Kalman, Ho, & Narendra, 1963), the controllability Gramian provides a qualitative measure for this property, in the form of the energy required to drive the dynamical system towards specific directions in the state space. The T-step controllability Gramian is defined as

$$\mathcal{W}_{\mathcal{F},T} \triangleq \sum_{\tau=0}^{T-1} \boldsymbol{W}^{\tau} \boldsymbol{B}_{\mathcal{F}} \boldsymbol{B}_{\mathcal{F}}' (\boldsymbol{W}')^{\tau}, \tag{34}$$

where $\boldsymbol{B}_{\mathcal{F}}$ corresponds to the set of attacked agents and defined in (9). The controllability Gramian $\mathcal{W}_{\mathcal{F},T}$ is positive definite if and only if the system is controllable in T steps (Kailath, 1980). However, even if a system is controllable, certain directions of the state space may be hard to reach (Sun & Motter, 2013). The smallest eigenvalue of the Gramian, $\lambda_1(\mathcal{W}_{\mathcal{F},T})$, is inversely related to the amount of energy required to move the system in the direction that is the most difficult to control, i.e., the eigenvector corresponding to $\lambda_1(\mathcal{W}_{\mathcal{F},T})$. Other controllability metrics, such as $\operatorname{trace}(\mathcal{W}_{\mathcal{F},T})$ and $\operatorname{trace}(\mathcal{W}_{\mathcal{F},T}^{-1})$, quantify the energy needed on average to move the system around on the state space. From the attacker's perspective, the system should be easily controllable. Thus, it targets agents for which one of the above mentioned spectra is optimized. In particular, the attacker attempts to minimize its effort to steer the system by maximizing $\lambda_1(\mathcal{W}_{\mathcal{F},T})$ and $\operatorname{trace}(\mathcal{W}_{\mathcal{F},T})$ or minimizing $\operatorname{trace}(\mathcal{W}_{\mathcal{F},T}^{-1})$.

The selection of control agents to optimize the spectrum of the Gramian does not generally admit a closed-form solution. Hence, finding graph-theoretic interpretation for most of these metrics is challenging and requires further research, see Baggio and Zampieri (2018) and Pasqualetti, Favaretto, Zhao, and Zampieri (2018) for further reading. The exception is $\operatorname{trace}(\mathcal{W}_{\mathcal{F},T})$ for which (Pasqualetti, Zampieri, & Bullo, 2014)

$$\operatorname{trace}(\mathcal{W}_{\mathcal{F},T}) = \sum_{i \in \mathcal{F}} \left(\sum_{\tau=0}^{T-1} \boldsymbol{W}^{2\tau} \right)_{ii}.$$

When $\boldsymbol{W}$ is Schur stable, then $\sum_{\tau=0}^{\infty} \boldsymbol{W}^{2\tau} = (I - \boldsymbol{W}^2)^{-1}$. For continuous time systems, if $\boldsymbol{W}$ is Hurwitz, this closed form solution becomes $\operatorname{trace}(\mathcal{W}_{\mathcal{F}}) = \sum_{i \in \mathcal{F}} \left(\boldsymbol{W}^{-1}\right)_{ii}$. For other metrics, selecting optimal agents, from the attacker's perspective, is a combinatorial problem and (in general) hard to solve. However, recent studies on submodularity and monotonicity of some of those metrics indicate that greedy algorithms for selecting the control agents result in a sub-optimal solution with a guaranteed performance bound (Clark, Alomair, Bushnell, & Poovendran, 2016; Olshevsky, 2017; Summers, Cortesi and Lygeros, 2015; Summers, Shames, Lygeros and Dörfler, 2015).

In the following subsection, we outline a graph-theoretic interpretation of the attacker's strategy to optimize $\operatorname{trace}(\mathcal{W}_{\mathcal{F},T})$ in consensus dynamics.

9.1.3. Case study: Consensus dynamics

We consider Gramian-based controllability on two types of consensus dynamics on undirected graphs, namely average consensus and leader-following consensus.

1. Edge Attack in Average Consensus: Assume that the attack happens in the form of a flow which enters one agent and exists from another agent. In particular, we say that $ik \in \mathcal{F}$ if the pair i, k is chosen by the attacker and their dynamics are

$$\begin{aligned} \dot{\psi}_i &= \sum_{j \in \mathcal{N}_i^{\text{in}}} (\psi_j - \psi_i) + \zeta_{ik}, \\ \dot{\psi}_k &= \sum_{j \in \mathcal{N}_k^{\text{in}}} (\psi_j - \psi_k) - \zeta_{ik}, \end{aligned} \tag{35}$$

where ζ_{ik} is the attack flow. This type of input signal, as schematically shown in Fig. 11(a), happens in power systems (DC input links) and distribution networks (Pirani & Taylor, 2020; Wei & van der Schaft, 2013). If we write the dynamics in vector form, it becomes

$$\dot{\boldsymbol{\psi}} = -L\boldsymbol{\psi} + \mathcal{B}\boldsymbol{\zeta}, \tag{36}$$

where $\mathcal{B}$ is the incidence matrix of the graph induced by the attacked edges. The controllability Gramian is

$$\mathcal{W}_c = \int_0^{\infty} \mathbf{e}^{-L\tau} \mathcal{B}\mathcal{B}' \mathbf{e}^{-L'\tau} d\tau. \tag{37}$$

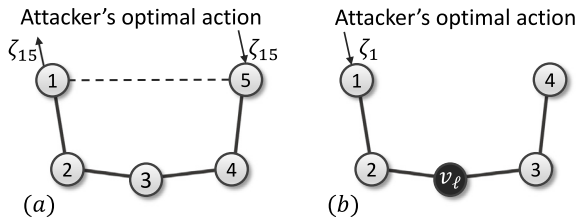


Fig. 11. (a) Attacker's optimal decisions based on (39), (b) attacker's optimal decision based on (41).

Since L is marginally stable, the infinite integral does not exist. However, the eigenvector of the marginally stable eigenvalue belongs to the subspace corresponding to the consensus value, which is of little interest to the attacker (as otherwise no attack would be needed). We remove this subspace by grounding one agent (removing a row and column corresponding to that agent) which makes the Laplacian non-singular. The *grounded Laplacian* matrix induced by the grounded agent v is denoted by L_{fv} (or simply L_f). Then, we have

$$\mathrm{tr}(\mathcal{W}_c) = \mathrm{tr}\left(\mathcal{B}' \int_0^\infty \mathbf{e}^{-2L_f\tau} d\tau \mathcal{B}\right) = \frac{1}{2}\mathrm{tr}\left(\mathcal{B}' L_f^{-1} \mathcal{B}\right). \tag{38}$$

It was shown in Ghosh, Boyd, and Saberi (2008) that the above value is independent of the choice of the grounded agent, i.e., we have

$$\mathrm{tr}\left(\mathcal{B}' L_f^{-1} \mathcal{B}\right) = \mathrm{tr}\left(\mathcal{B}' L^\dagger \mathcal{B}\right) = \sum_{ij\in\mathcal{F}} \mathfrak{R}_{ij}, \tag{39}$$

where $L^\dagger$ is the Moore–Penrose inverse of L and $\mathfrak{R}_{ij}$ is the effective resistance between agents i and j. Eq. (39) indicates that the trace of the controllability Gramian is the summation of effective resistances between agent pairs chosen by the attacker. Thus, if the attacker seeks to maximize (39), i.e., minimize the attack energy, by choosing m agent pairs, it should choose m pairs with *the largest effective resistance* in the graph. Fig. 11(b) is an example of an optimal attack.

2. Agent Attack in Leader–Follower Consensus: We consider a leader–follower dynamical system on undirected graphs which is widely studied in formation control problems (Hao & Barooah, 2013; Pirani, Shahrivar, Fidan, & Sundaram, 2018; Rahmani et al., 2009). Based on this model, there is a leader, which has access to the control input or determines the set-point, and a set of followers, which follow the state of the leader. The dynamics of leader and followers are given by

$$\begin{bmatrix}\dot{\boldsymbol{\psi}}_f(t)\\ \dot{\boldsymbol{\psi}}_\ell(t)\end{bmatrix} = -\underbrace{\begin{bmatrix}L_f & L_{\ell f}\\ L_{f\ell} & L_\ell\end{bmatrix}}_{L}\begin{bmatrix}\boldsymbol{\psi}_f(t)\\ \boldsymbol{\psi}_\ell(t)\end{bmatrix} + \begin{bmatrix}\boldsymbol{B}_{\mathcal{F}}\\ 0\end{bmatrix}\boldsymbol{\zeta}_{\mathcal{F}}(t), \tag{40}$$

where $\boldsymbol{\psi}_f(t)$, $\boldsymbol{\psi}_\ell(t)$ and $\boldsymbol{\zeta}_{\mathcal{F}}(t)$ are the state of followers, the state of the leader, and the attack vector, respectively. The leader is not affected by communication attacks. It also keeps its state constant and does not incorporate the states of the followers, i.e., $\dot{\boldsymbol{\psi}}_\ell(t) = 0$. Thus, we have $L_{f\ell} = L_\ell = 0$. Matrix $\boldsymbol{B}_{\mathcal{F}}$, formed from indicator vectors as in (9), determines the agents that are under attack and $L_{\ell f}$ determines the connection of the leader to the followers. Matrix L_f is the grounded Laplacian matrix discussed earlier. One of the key properties of this matrix, which is used in this analysis, is that $[L_f^{-1}]_{ii} = \mathfrak{R}_{i\ell}$, where $\mathfrak{R}_{i\ell}$ is the effective resistance between agent i and the leader. Similar to the case of edge attack, the objective of the attacker is to maximize the trace of the controllability Gramian. Calculating the Gramian integral yields

$$\mathrm{trace}(\mathcal{W}_{\mathcal{F}}) = \sum_{i\in\mathcal{F}} \left(L_f^{-1}\right)_{ii} = \sum_{i\in\mathcal{F}} \mathfrak{R}_{i\ell}. \tag{41}$$

Based on (41), if the attacker wants to minimize the average energy by attacking m agents, it must select m agents with the largest effective resistance from ℓ. Fig. 11(b) is an example of an optimal attack.

9.2. System norm approaches

Another way to quantify the attacker's impact is through the use of system norms from the attack signal to the output of interest, e.g., the state of the agents. Two widely used system norms in robust control are $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms. Since these methods were initially developed on continuous time systems, we focus on such systems in this section.

Suppose that the evolution of a network of agents is described by (3) where $\boldsymbol{D} = 0$ and $\boldsymbol{B} = \boldsymbol{B}_{\mathcal{F}}$ which corresponds to the set of attacked agents defined in (9). The transfer function of the error dynamics from attack input $\boldsymbol{\zeta}_{\mathcal{F}}(t)$ to output $\boldsymbol{y}(t)$ in the Laplace domain is $G(s) \triangleq \boldsymbol{C}(sI - \boldsymbol{W})^{-1}\boldsymbol{B}_{\mathcal{F}}$. The system $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms are defined as

$$\begin{aligned}\|G\|_2 &\triangleq \left(\frac{1}{2\pi}\mathrm{trace}\int_0^\infty G^*(j\omega)G(j\omega)d\omega\right)^{\frac{1}{2}},\\ \|G\|_\infty &\triangleq \sup_{\omega\in\mathbb{R}} \sigma_{\max}(G(j\omega)),\end{aligned} \tag{42}$$

where $\sigma_{\max}(.)$ is the maximum singular value of a matrix. The system $\mathcal{H}_2$ norm can also be calculated based on the observability Gramian $\mathcal{W}_o$, which is the solution of the following Lyapunov equation

$$\begin{aligned}&\boldsymbol{W}'\mathcal{W}_o + \mathcal{W}_o\boldsymbol{W} = -\boldsymbol{C}'\boldsymbol{C},\\ &\|G\|_2 = \mathrm{trace}(\boldsymbol{B}_{\mathcal{F}}'\mathcal{W}_o\boldsymbol{B}_{\mathcal{F}}).\end{aligned} \tag{43}$$

Unlike the approach of the structured system theory to resilient DCSs, which was based on nonzero patterns of system matrices and not the magnitude of the elements, the system norm approach depends on the exact value of the matrix elements. Thus, having knowledge about the nonzero patterns of dynamic matrices is not sufficient and one has to specify the type of matrices which describe the interactions between agents. We revisit the leader–follower consensus dynamics discussed in Section 9.2.1 to further explain this fact.

Remark 7 ($\mathcal{H}_2$ *vs.* $\mathcal{H}_\infty$ *Norm*)**.** From a security perspective, either the system $\mathcal{H}_2$ or $\mathcal{H}_\infty$ norm can be used to quantify the attack impact. If the frequency content of the attack signal is unknown, using a $\mathcal{H}_2$ norm is a more reasonable choice as it is calculated over all frequencies. However, if the objective is to find the worst-case attack impact over all frequencies, the system $\mathcal{H}_\infty$ norm is an appropriate choice.

9.2.1. Case study: Consensus dynamics revisited

1. Edge Attack in Average Consensus: Consider the consensus dynamics under an attack flow as in (36). Suppose that the attacker measures the difference between agent values under attack, i.e., $\boldsymbol{y} = \mathcal{B}'\boldsymbol{\psi}$. Dynamics (36) together with this measurement form a symmetric system, i.e., L is symmetric and $\boldsymbol{B} = \boldsymbol{C}'$. Thus, the $\mathcal{H}_\infty$ norm is equal to the DC gain of the system (Tan & Grigoriadis, 2001), i.e., for the transfer function from $\boldsymbol{\zeta}$ to $\boldsymbol{y}$ we have

$$\|G\|_\infty = \sigma_{\max}\left(\mathcal{B}' L^\dagger \mathcal{B}\right). \tag{44}$$

Unlike the trace of $\mathcal{B}'L^{\dagger}\mathcal{B}$, in (39), interpreting its largest singular value is hard. We consider the simple case where only one agent pair, i and j, is under attack, i.e., $\mathcal{B} = \mathbf{e}_{ij}$. In this case, (44) becomes scalar and we have $\|G\|_{\infty} = \sigma_{\max}\left(\mathbf{e}_{ij}'L^{\dagger}\mathbf{e}_{ij}\right) = \mathfrak{R}_{ij}$. Thus, in order to have a large impact, the attacker must choose agent pairs with the largest effective resistance in the network.

2. Agent Attack in Leader–Follower Consensus: Here, a single attacker i targets a set of agents and $\boldsymbol{B}_{\mathcal{F}} = C'$. As before, the $\mathcal{H}_{\infty}$ norm is equal to the DC gain of the system (Farina & Rinaldi, 2000) and we have $\|G\|_{\infty} = \sigma_{\max}\left(\boldsymbol{B}_{\mathcal{F}}'L_f^{-1}\boldsymbol{B}_{\mathcal{F}}\right)$. For the case of a single agent under attack, we have $\boldsymbol{B}_{\mathcal{F}} = \mathbf{e}_i$ and $\|G\|_{\infty} = \mathbf{e}_i'L_f^{-1}\mathbf{e}_i = \mathfrak{R}_{i\ell}$. Hence, to have a large impact on a target agent, the attacker must choose an agent in the network with the largest effective resistance from the leader.

10. Related problems in resilient distributed control systems

In this section, we briefly discuss other problems on the resilience of distributed control systems which use graph theory as a tool in the analysis.

10.1. Resilience to actuator/sensor and link removals

So far, we discussed the case where the agents or the communications between the agents are under attack. In that context, we assume that the control inputs are not affected by the attacker. In some situations, however, the attacker may choose to remove certain sensors, actuators, or communication links entirely; for instance, removal of communication links can be captured within the setup of denial of service attacks (Biron et al., 2018; De Persis & Tesi, 2014) or jamming attacks (Mokdad et al., 2015). In those cases, the main concern is to retain the controllability of the system. This can be written in terms of a *robust structural controllability* problem. In particular, the objective is to maintain a system's controllability despite the removal of a subset of actuators.

Consider the linear time invariant system (3) where a subset of control inputs are removed (potentially due to adversarial actions). In this case, the minimum number of actuators which retain the controllability of the system is determined by the following problem

$$\begin{aligned} \underset{\boldsymbol{B}\in\mathbb{R}^{n\times m}}{\arg\min} \quad & \|\boldsymbol{B}\|_0 \\ \text{such that} \quad & (\boldsymbol{W}, \boldsymbol{B}_{\mathcal{U}\setminus\mathcal{F}}) \text{ is structurally controllable} \\ & \forall \mathcal{F} \subset \mathcal{U} \ with |\mathcal{F}| \leq m, \end{aligned} \tag{45}$$

where $\boldsymbol{B}_{\mathcal{U}\setminus\mathcal{F}}$ corresponds to the structure of the input matrix $\boldsymbol{B}$ whose columns corresponding to set $\mathcal{F}$ are removed (i.e., actuators are failed). The number of actuator faults are upper bounded by m. It is shown that the above problems are NP-hard and polynomial time algorithms to approximate the solution of those problems have been proposed (Liu et al., 2013; Pequito, Ramos, Kar, Aguiar, & Ramos, 2017).

In other set of problems, the attacker targets a set of links in the network to remove. Several performance measures may be affected by such an action. When robust controllability (or observability) is of interest, the problem can be written as follows (Liu et al., 2013)

$$\begin{aligned} \underset{\boldsymbol{B}\in\mathbb{R}^{n\times m}}{\arg\min} \quad & \|\boldsymbol{B}\|_0 \\ \text{such that} \quad & (\boldsymbol{W}_{\mathcal{E}_{\mathcal{X},\mathcal{X}}\setminus\mathcal{E}_{\mathcal{F}}}, \boldsymbol{B}) \text{ is structurally controllable} \\ & \forall \mathcal{E}_{\mathcal{F}} \subset \mathcal{E}_{\mathcal{X},\mathcal{X}} \text{ with } |\mathcal{E}_{\mathcal{F}}| \leq \bar{m}, \end{aligned} \tag{46}$$

where $\mathcal{E}_{\mathcal{F}}$ is the set of edges affected by the attacker, upper bounded by $\bar{m}$. Similar to (45), the above problem is NP-hard in general. The main approach adopted to the above problems is through showing the relation between robust structural observability (or controllability) and the set cover problem. Upper and lower bounds on the feasible dedicated sensor or actuator configurations have been proposed in terms of the spanning trees of the system graph. More recently, the problem of *resilient strong structural controllability* has been studied considering uncertain zero and non-zero parameters for cases where there are sensor, actuator, or communication failures (Mousavi, Haeri, & Mesbahi, 2017; Popli, Pequito, Kar, Pedro, & Ilić, 2019).

10.2. Strategic sensor and actuator placement on graphs

Our focus in the previous sections was to find graph conditions which ensure resiliency to attacks for certain distributed algorithms. However, in many situations, the underlying network topology is sparse and cannot be changed. Furthermore, the number of sensors/actuators is limited. In these cases, an alternative approach is to place those limited number of sensors (or actuators) on specific agents in the network in order to optimally detect the attack or mitigate its impact.

There is a vast literature on sensor (or actuator) placement to enhance the observability (or controllability) in terms of the rank of the observability (or controllability) matrix, Gramian-based metrics (as discussed previously), or the error variance of the Kalman filters (Ye, Woodford, Roy and Sundaram, 2020; Zhang, Ayoub, & Sundaram, 2017). In all these problems, there is a single decision maker which deploys sensors (or place actuators) on a set of agents. However, in security problems, the adversary plays the role of a different decision maker which tries to optimize its own cost function, e.g., maximize impact or minimize visibility. This introduces a *strategic* sensor (or actuator) placement problem, taking the attacker's actions into account. In this direction, game theory can be used as a powerful tool to address this set of problems; see Alpcan and Buchegger (2011), Han, Niyato, Saad, Başar, and Hjorungnes (2012), Manshaei, Zhu, Alpcan, Basar, and Hubaux (2013), Ye, Roy and Sundaram (2020) and Zhu and Basar (2015) and references therein.

Strategic sensor placement in the network to detect cyber-attacks has been recently studied (Dahan, Sela, & Amin, 2022; Milošević, Dahan, Amin, & Sandberg, 2019; Pirani, Nekouie, Sandberg, & Johansson, 2021a). In this setting, the attacker seeks to apply attack inputs while being stealthy and the detector tries to detect the attack. Several approaches have been adopted to characterize the equilibria of the security games. Nash equilibrium is used to model simultaneous decision making and Stackelberg game model is used for the case where the defender must act before the attacker. In design problems, the Stackelberg game is a popular approach to defend against cyber-attacks. In particular, the detector acts as the game leader and places sensors on agents considering the worst case attack strategies. The applicability of each method, based on the nature of the attack and the structure of the cyber–physical system, is discussed in Manshaei et al. (2013).

In addition to the strategic attack detection, a defense mechanism can help mitigate the impact of the attack via certain control actions (Gueye & Marbukh, 2012; He et al., 2021; Pirani, Nekouie, Sandberg, & Johansson, 2021b; Zhu & Basar, 2015). The attacker's impact can be quantified by either of the methods discussed in Section 9. A comprehensive defense strategy must include both strategic detection and mitigation mechanisms. From the defender's perspective, the game equilibrium determines the optimal placement of sensors and actuators to detect the attack and mitigate its impact. Recently, it is shown that for certain classes of game, optimal defender's decisions are explainable via certain network centrality measures (Pirani et al., 2021b).

10.3. Network coherence as a measure of resiliency

In the DCSs literature, the notion of network coherence is used to quantify the ability of a network to reject communication disturbances while performing a formation control or a consensus algorithm in large scale systems (Bamieh, Jovanovic, Mitra, & Patterson, 2012; Patterson & Bamieh, 2010; Tegling, Mitra, Sandberg, & Bamieh, 2019). It is usually described in terms of system $\mathcal{H}_2$ or $\mathcal{H}_\infty$ norms from the disturbance signal to the output of interest, e.g., position (phase) or velocity (frequency) (Pirani, Simpson-Porco, & Fidan, 2017; Poolla, Bolognani, & Dörfler, 2015; Tegling, Bamieh, & Gayme, 2015). Interpreting the disturbances as attack inputs, some of the approaches and the results in this line of research, e.g., scalability of algorithms and leader selection schemes, can be readily used in resilient distributed algorithms, as we will briefly discuss in the following paragraphs.

10.3.1. Leader selection

The objective is to choose the optimal agents in the network as leaders, i.e., agents which receive the control signal, such that the network coherence is maximized, i.e., system $\mathcal{H}_2$ or $\mathcal{H}_\infty$ norms from the disturbance input to the output of interest are minimized. It is shown in Fitch and Leonard (2015) that the optimal leader to minimize the $\mathcal{H}_2$ norm in consensus dynamics is the *information central* agent in the network: a agent in which the summation of effective resistances to the rest of the agents in the network is minimized. It is also shown that the leader which optimizes the $\mathcal{H}_\infty$ norm is not necessarily the graph's information center (Pirani et al., 2018). Moreover, a graph-theoretic condition for the leader to co-optimize both metrics is discussed in Pirani et al. (2018).

10.3.2. Scalability

Another problem of interest is the scalability of the network coherence in graphs with various structures. In Bamieh et al. (2012), it is discussed that how the network coherence scales with the network size for regular lattices in 1, 2 and higher dimensions. The scalability of these metrics is also discussed for random graphs. In particular, for Erdős–Rényi random graphs and random regular graphs, tight characterizations of the network coherence are discussed in Pirani and Sundaram (2016) and Pirani et al. (2018).

11. Resilience in certain classes of graphs

In this section, we summarize some pertinent topological properties of certain classes of graphs. We focus on the properties of graph connectivity and graph robustness provided in Section 2.1.2, since they are particularly relevant to the resilience of distributed algorithms against adversarial actions. Our focus is on undirected networks, unless otherwise indicated. We start from simple graph structures.

11.1. Paths, cycles, trees, and complete graphs

An undirected path of length n is the simplest connected graph with connectivity $\kappa = 1$ and robustness $r = 1$. A cycle is a path of length n whose start and end agents are connected. For a cycle graph, the connectivity is $\kappa = 2$ and the robustness is $r = 1$. Trees are connected acyclic graphs with connectivity $\kappa = 1$ and robustness $r = 1$. A complete graph is $(n-1)$-connected and $\lceil \frac{n}{2} \rceil$-robust.

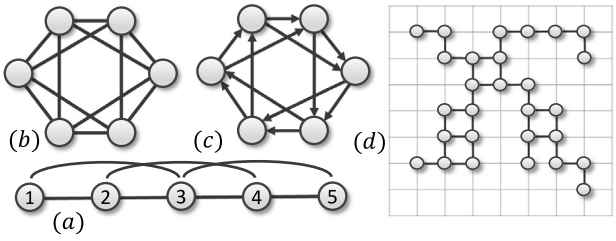


Fig. 12. (a) A 2-nearest neighbor path (b,c) an undirected and a directed circulant graph, (d) A connected formation on a 2-dimensional lattice.

11.2. Circulant networks, k-nearest neighbor paths, and 1-D random geometric graphs

Definition 10 (*k-Nearest Neighbor Paths*)**.** A k-nearest neighbor path, $\mathcal{P}(n,k)$, is a network comprised of n agents in a path, where the agents are labeled as $1, 2, \ldots, n$ from one end of the path to the other, and each agent i can communicate with its k nearest neighbors behind it and k nearest neighbors ahead of it (whenever possible), i.e., $i-k, i-k+1, \ldots, i-1, i+1, i+2, \ldots, i+k$, for some $k \in \mathbb{N}$. An example of $\mathcal{P}(n,k)$ is shown in Fig. 12(a).

Such k-nearest neighbor paths are relevant for modeling vehicle platoons, due to the limited sensing and communication range for each vehicle. It is shown in Pirani et al. (2019) that a k-nearest neighbor path, $\mathcal{P}(n,k)$, is a k-connected graph. We will discuss the robustness of $\mathcal{P}(n,k)$ later in this section. A similar structure to $\mathcal{P}(n,k)$ is a 1-dimensional geometric random graph which captures edges between agents that are in close (spatial) proximity to each other.

Definition 11 (*Geometric Random Graphs*)**.** A geometric random graph $\mathcal{G}^d_{n,\rho,l} = \{\mathcal{V}, \mathcal{E}\}$ is an undirected graph generated by first placing n agents (according to some mechanism) in a d-dimensional region $\Omega_d = [0,l]^d$, where $d \in \mathbb{Z}_{\geq 1}$. We denote the position of agent $i \in \mathcal{V}$ by $x_i \in \Omega_d$. Agents $i, j \in \mathcal{V}$ are connected by an edge if and only if $\|x_i - x_j\| \leq \rho$ for some threshold ρ, where $\|.\|$ is some appropriate norm (often taken to be the standard Euclidean norm). When the agent positions are generated randomly (e.g., uniformly and independently) in the region, one obtains a geometric random graph.

In the more general models of $\mathcal{G}^d_{n,\rho,l}$, the length l is also allowed to increase and the density $\frac{n}{l^d}$ can converge to some constant, making it suitable for capturing both dense and sparse random networks. The following result holds for 1-dimensional geometric random graphs.

Proposition 4 (*Zhang et al., 2015*)**.** *In $\Omega_1 = [0,l]$ with fixed l, if $\mathcal{G}^1_{n,\rho,l}$ is k-connected, then it is at least $\lfloor \frac{k}{2} \rfloor$-robust.*

Based on Definition 11, the k-nearest neighbor path can be seen as a geometric graph $\mathcal{G}^d_{n,\rho,l}$ with $\rho = \frac{lk}{n-1}$ and placing the agents as follows: the first agent, 1, is placed on one end of the line and the ith agent is placed at distance $\frac{(i-1)l}{n-1}$ from 1. Thus, based on Proposition 4 and the fact that the k-nearest neighbor path $\mathcal{P}(n,k)$ is k-connected, we conclude that it is at least $\lfloor \frac{k}{2} \rfloor$-robust.

Definition 12 (*Circulant Graphs*)**.** An undirected graph of n agents is called circulant if the n vertices of the graph can be numbered from 0 to $n-1$ in such a way that if some two vertices numbered x and $(x \pm d)$ mod n are adjacent, then every two vertices numbered z and $(z \pm d)$ mod n are adjacent. A directed graph is circulant

(with the above labeling) if some two vertices numbered x and $(x+d)$ mod n are adjacent, then every two vertices numbered z and $(z+d)$ mod n are adjacent.

Informally speaking, an undirected k-circulant graph is a k-nearest neighbor cycle graph. Thus, with the same reasoning, an undirected k-circulant graph is $2k$-connected and at least $\lfloor \frac{k}{2} \rfloor$-robust. The result is extended to directed circulant graphs where it is shown that it is at least $\lfloor \frac{k+2}{4} \rfloor$-robust (Usevitch & Panagou, 2017). An example of an undirected and a directed circulant graph is shown in Fig. 12(b,c).

11.3. Formation graphs on 2-Dimensional lattice

A specific type of geometric graph is the two dimensional lattice which has been widely used in formation control of autonomous robots (Guerrero-Bonilla, Saldana, & Kumar, 2018; Saulnier et al., 2017). A lattice is a set of linear combinations with integer coefficients of the elements of a basis of $\mathbb{R}^2$. The elements of the set are lattice points. Let $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ be bases of a 2-dimensional lattice with $\|\boldsymbol{v}_1\| = \|\boldsymbol{v}_2\| = \ell$ where ℓ is the lattice length. Every point x on the lattice can be described by $x = a_i\boldsymbol{v}_1 + b_j\boldsymbol{v}_2$ where $a_i, b_j \in \mathbb{Z}$. In a lattice, two agents i and j are connected if $\|x_i - x_j\| \leq \ell$ where $\|.\|$ is the Euclidean norm. Given a set of agents $\mathcal{V}$ and a distance ℓ, the graph $\mathcal{G}_\ell = \{\mathcal{V}, \mathcal{E}_\ell\}$ with edge set $\mathcal{E}_\ell \triangleq \{(i,j) | \|x_i - x_j\| \leq \ell\}$ is called the proximity graph of the set $\mathcal{V}$. We describe the communication range by a function $R : \mathbb{Z}_{\geq 1} \to \mathbb{R}$ that maps the number of robots m to a distance where m robots are ensured to be reached. A formation of n robots is said to be connected if its associated proximity graph $\mathcal{G}_\ell$ is connected. An example of a lattice and a connected formation is shown in Fig. 12(d). It was shown in Guerrero-Bonilla et al. (2018) that in a connected formation of n robots, every robot has at least $1 \leq m \leq n-1$ robots within a distance $m\ell$. Based on this, one can compute a minimum communication range for the robots in a formation to guarantee resilience in the communication network.

Proposition 5 (*Guerrero-Bonilla et al., 2018*). *Given a set $\mathcal{V}$ of $4f+1$ agents in a connected formation, if the communication range of every agent satisfies $R \geq 3f\ell$, then the associated graph of the formation is $(2f+1)$-robust.*

11.4. Random graphs

A common approach to modeling complex networks is via the framework of *random graphs*, i.e., by drawing a graph from a certain probability distribution over the set of all possible graphs on a given set of agents. Such random graph models have diverse applications (Bollobás, 2001; Newman, Strogatz, & Watts, 2001), including in modeling cascading failures in large scale systems (Crucitti, Latora, & Marchiori, 2004; Yagan, Qian, Zhang, & Cochran, 1998). Here, we summarize the connectivity and robustness properties of certain commonly studied random graph models.

11.4.1. Erdős–Rényi random graphs

An Erdős–Rényi (ER) random graph $\mathcal{G}(n,p)$ is a graph on n agents, where each edge between two distinct agents is presented independently with probability p (which could be a function of n). We say that a graph property holds *asymptotically almost surely* if the probability of drawing a graph with that property goes to 1 as $n \to \infty$. The following theorem shows the probability threshold for which a graph $\mathcal{G} \in \mathcal{G}(n,p)$ is r-connected and r-robust.

Theorem 13 (*Zhang et al., 2015*). *For any constant $r \in \mathbb{Z}_{\geq 1}$,*

$$t(n) = \frac{\ln n + (r-1)\ln\ln n}{n}$$

is a threshold function for the ER random graph $\mathcal{G}$ to have minimum degree r, to be r-connected, and to be r-robust.

According to the example graph in Fig. 3(b), graph robustness is a much stronger property than the graph connectivity and the minimum degree. However, Theorem 13 indicates that the above threshold function for r-connectivity (and minimum degree r) is also a threshold function for the stronger property of r-robustness in ER random graphs.

11.4.2. Random regular graphs

Let $\Omega_{n,d}$ be the set of all undirected graphs on n agents where every agent has degree d (note that this assumes that nd is even). A *random d-regular graph* (d-RRG), denoted $\mathcal{G}_{n,d}$ is a graph drawn uniformly at random from $\Omega_{n,d}$. For $d \geq 3$, it is shown that $\mathcal{G}_{n,d}$ is asymptotically almost surely d-connected (Bollobás, 2001). Based on Friedman (2003), for any $\epsilon > 0$, the algebraic connectivity of a random d-regular graph satisfies

$$\lambda_2(L) \geq d - 2\sqrt{d-1} - \epsilon, \tag{47}$$

asymptotically almost surely. As discussed in Saulnier et al. (2017), if the algebraic connectivity of a graph is bigger than $r-1$, then the network is at least $\lfloor \frac{r}{2} \rfloor$-robust. Hence, according to (47), an d-RRG is at least $\lfloor \frac{d-2\sqrt{d-1}}{2} \rfloor$-robust asymptotically almost surely.

11.4.3. Random interdependent networks

An interdependent network $\mathcal{G}$ is denoted by a tuple $\mathcal{G} = (\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_k, \mathcal{G}_p)$ where $\mathcal{G}_l = (\mathcal{V}_l, \mathcal{E}_l)$ for $l = 1, 2, \ldots, k$ are called the *subnetworks* of the network $\mathcal{G}$, and $\mathcal{G}_p = (\mathcal{V}_1 \cup \mathcal{V}_2 \cup \cdots \cup \mathcal{V}_k, \mathcal{E}_p)$ is a k-partite network with $\mathcal{E}_p \subseteq \cup_{l \neq t} \mathcal{V}_l \times \mathcal{V}_t$ specifying the interconnection (or inter-network) topology. Applications of interdependent networks in modeling communication networks and power grid are discussed in Parandehgheibi and Modiano (2013). Define the sample space Ω_n to consist of all possible interdependent networks $(\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_k, \mathcal{G}_p)$ and the index $n \in \mathbb{N}$ denotes the number of agents in each subnetwork. A random interdependent network is a network $\mathcal{G} = (\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_k, \mathcal{G}_p)$ drawn from Ω_n according to a given probability distribution.

We assume that $|\mathcal{V}_1| = |\mathcal{V}_2| = \cdots = |\mathcal{V}_k| = n$ and that the number of subnetworks k is at least 2. Similar to Theorem 13 for ER random graphs, there exists a sharp threshold for connectivity and robustness of random interdependent networks.

Theorem 14 (*Shahrivar, Pirani, & Sundaram, 2017*). *Consider a random interdependent network $\mathcal{G} = (\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_k, \mathcal{G}_p)$. Then, for any positive integers r and $k \geq 2$,*

$$t(n) = \frac{\ln n + (r-1)\ln\ln n}{(k-1)n}$$

is a threshold for r-connectivity and r-robustness of $\mathcal{G}$.

11.4.4. Random intersection networks

Random intersection graphs belong to class of random graphs for which every agent is assigned a set of objects selected by some random mechanism. They have applications in wireless sensor networks, frequency hopping spread spectrum, spread of epidemics, and social networks (Singer-Cohen, 1995).

Given a agent set $\mathcal{V} = \{1, 2, \ldots, n\}$, each agent i is assigned an object set S_i from an object pool $\mathcal{P}$ consisting of P_n distinct objects, where P_n is a function of n. Each object S_i is constructed using the following two-step procedure: (i) The size of S_i, $|S_i|$,

is determined according to some probability distribution $\mathcal{D} : \{1, 2, \ldots, P_n\} \rightarrow [0, 1]$ in which $\sum_{x=1}^{P_n} \mathbb{P}(|S_i| = x) = 1$. (ii) Conditioning on $|S_i| = s_i$, set S_i is chosen uniformly among all s_i-size subsets of $\mathcal{P}$. Finally, an undirected edge is assigned between two agents if and only if their corresponding object sets have at least one object in common. There are variations of the general random intersection graph such as binomial random intersection graphs and uniform random intersection graphs, each of which focuses on a certain probability distribution $\mathcal{D}$. The following theorem discusses the connectivity and robustness of random intersection graphs.

Theorem 15 (*Zhao, Yağan, & Gligor, 2014*)**.** *Consider a general random intersection graph $\mathcal{G}(n, P_n, \mathcal{D})$. Let X be a random variable following probability distribution $\mathcal{D}$. With a sequence α_n for all n defined through $\frac{\{\mathbb{E}[X]\}^2}{P_n} = \frac{\ln n + (r-1)\ln\ln n + \alpha_n}{n}$, if $\mathbb{E}[X] = \Omega(\sqrt{\ln n})$, $Var[X] = o(\frac{\{\mathbb{E}[X]\}^2}{n(\ln n)^2})$ and $\alpha_n = o(\ln n)$, and $\lim_{n\to\infty} \alpha_n = \infty$, then the graph is asymptotically almost surely r-connected and r-robust.*

12. Future directions

This paper provided an overview of the existing graph-theoretic tools which can be used to analyze the resilience of distributed control systems. Compared to system-theoretic approaches to the robustness and fault tolerance of control systems, graph-theoretic approaches are relatively new and demand more development, primarily in the following three directions: (i) Developing graph-theoretic methods to facilitate analyzing a wider range of distributed algorithms and more complex adversarial actions; (ii) Reinterpreting the known system-theoretic notions of resilience and robustness of dynamical systems from a graph-theoretic perspective; (iii) Investigating the resilience of a wider range of distributed algorithms using available graph-theoretic tools.

Here, we propose a few research avenues which are worth investigating in the future.

- **Spectral Approach to Network Structures:** One of the necessary steps towards reconciling system-theoretic approaches and graph theory is to find algebraic interpretations of certain network structures. Algebraic graph theory is an active topic of research in mathematics (Godsil & Royle, 2001). However, specific structural properties of networks that are widely used in analyzing the resilience of distributed control systems, e.g., network robustness, are quite new notions defined within the field of systems and control. Hence, their algebraic interpretations are still under study.
  An example is the relation of the algebraic connectivity and the network robustness. Defining the *edge-boundary* of a set of agents $S \subset \mathcal{V}$ is given by $\partial S = \{(i, j) \in \mathcal{E} \mid i \in S, j \in \mathcal{V} \setminus S\}$. The *isoperimetric constant* of $\mathcal{G}$ is defined as (Chung, 1997)
  $$i(\mathcal{G}) \triangleq \min_{S\subset\mathcal{V}, |S|\leq \frac{n}{2}} \frac{|\partial S|}{|S|}. \tag{48}$$
  Based on the above definition and the definition of the network robustness, we conclude that if $i(\mathcal{G}) > r - 1$, then the graph is at least r-robust. Moreover, we have $\lambda_2(L) \leq 2i(\mathcal{G})$ (Chung, 1997). Based on this, if $\lambda_2(L) > r - 1$, then the network is at least $\lfloor \frac{r}{2} \rfloor$-robust (Saulnier et al., 2017; Shahrivar et al., 2017). However, $\lfloor \frac{r}{2} \rfloor$ provides a loose lower bound for the network robustness. An example is a star graph which is 1-robust with $\lambda_2(L) = 1$. Further research is needed to be done to find tighter relations between algebraic connectivity and the network robustness.
- **Resilience with Minimum Communication:** Due to the fact that communications between agents can be costly in many applications, the problem of reaching a certain level of resilience with minimum communication, i.e., edges between agents, is worth investigating. In specific dynamical systems, e.g., consensus dynamics, adding edges may also degrade the controllability of the system. Few recent works have focused on minimizing the number of edges while reaching a certain level of security (Weerakkody, Liu, & Sinopoli, 2017) or maximizing the connectivity without violating the controllability of the system (Abbas, Shabbir, Jaleel and Koutsoukos, 2020). While reaching a certain level of connectivity or robustness with the minimum number of edges being studied, many questions remain, particularly in scenarios where edges can be dynamically added or changed over time (as in networks with mobile nodes).
- **Graph-Theoretic Approach to Attack Energy and the Safe Operating Set:** In Section 9 some graph-theoretic interpretations of the attack's impact were discussed. Among those, the attack energy has not been well studied. The objective of the attacker, other than impact and detectability, can be to access the system with minimum energy. One way to quantify the attack energy is via using the spectra of the controllability Gramian $\mathcal{W}_{\mathcal{F}}$. An interesting research avenue is to design the network to maximize the attack energy using an appropriate spectrum of the controllability Gramian. One of applications of these quantitative (and continuous) metrics of controllability is to provide a measure for the distance of the system to boundaries of the safe operating region. The level of the controllability that the attacker has over the network can be interpreted as its ability to push the system to the boundaries of the safe operating region. From this view, graph-theoretic interpretations of control techniques to mitigate the attack impact or reduce its controllability, such as event-triggered control methods or control barrier function methods (Ames, Xu, Grizzle, & Tabuada, 2017), is an interesting avenue for further research.
- **Resilience of other Classes of Random Graphs:** The study of the connectivity and network robustness of random graphs with the various probability distribution of edge formation is another important future research line. As discussed in Section 11.4, structural properties of several classes of random networks (vis-à-vis their relevance to resilience) have already been studied. However, extensions to a broader class of networks (including geometric random graphs in higher dimensions, as well as sparse random networks with different degree distributions) would be of significant interest.
- **Resilience of Networks with Nonlinear Interactions:** Throughout this survey paper, the focus was on DCSs in which the interactions between agents are linear. In some cases of DCSs, the local interactions are nonlinear, e.g., synchronization of Kuramoto oscillators with applications to power systems (Dörfler & Bullo, 2014) and attraction –repulsion functions in swarm robotics (Gazi & Passino, 2003). Extension of the methods discussed in this survey to those classes of nonlinear systems requires further investigation.
- **Resilience in Distributed Closed-Loop Settings, including Multi-Agent Reinforcement Learning:** Much of the work that we described in this survey pertains to open-loop settings, where the agents are attempting to compute a certain quantity (e.g., consensus, optimization, estimation, inference), but are not closing the loop by applying control inputs based on their computations. Closing the loop in networks with adversaries poses several significant challenges that remain to be addressed. A specific emerging

version of this problem pertains to enabling a team of agents to cooperatively learn optimal policies for interacting with their environment, particularly when the dynamics of the environment are not initially known to the agents. Such *multi-agent reinforcement learning* settings share some common features with the distributed consensus, optimization, and estimation problems we described earlier in the paper, in that adversarial agents can provide incorrect information about their observations and costs to the other agents. However, these problems also introduce additional lines of complexity for resilience, in that adversarial agents can also affect the underlying system with their inputs. There have been recent initial explorations of resilient algorithms in such settings (Figura, Kosaraju, & Gupta, 2021; Lin, Gade, Sandhu, & Liu, 2020; Wu, Shen, Chen, & Ling, 2021; Xie, Mou, & Sundaram, 2021), but much work remains to be done to understand how to mitigate adversaries that can not only send incorrect information but can also take destructive actions on the shared system that all agents are observing.

- **Resilience of Graph Neural Networks:** A *Graph Neural Network* (GNN) is a type of neural network designed to process and analyze graph-structured data. GNNs operate on graphs by performing computations on the agents and edges, using information from the graph structure to guide the computation. The main idea behind GNNs is to learn agent embeddings, which are low-dimensional vectors that represent the agents in the graph. These embeddings capture the structural information of the graph and can be used for a variety of tasks, such as agent classification, link prediction, and graph classification. The security of GNNs is a relatively new and active research area. There is ongoing research on different types of attacks on GNNs, as well as the development of defenses against these attacks. One important area of research is the vulnerability of GNNs to adversarial attacks, where an attacker can modify the input graph or the GNN model itself to manipulate the output. Some recent papers in this area include (Sun et al., 2022; Zugner & Gunnemann, 2019).
- **Graceful Degradation of DCS:** Much of the existing work pertaining to resilience of DCS that we have surveyed in this paper has a *binary* flavor: if the network satisfies certain properties and the number of adversaries is less than a certain value, then the guarantees hold in full, but if those conditions are not satisfied, then there are no guarantees. However, in practice it would be highly desirable for guarantees to scale with the number (or behavior) of adversaries in the network, as opposed to exhibiting the kind of binary behavior described above. In particular, graceful degradation of performance as the number or sophistication of adversaries increases is an area of research that is significantly underdeveloped at this point of time, and deserves additional research. Indeed, the notion of resilience in many research communities has the connotation of adapting and "bouncing back" (or "bouncing forward") after disruptions (Manyena, O'Brien, O'Keefe, & Rose, 2011; Smith et al., 2008). New techniques to enable this kind of adaptation in networks to recover and adapt to adversarial actions would be of significant theoretical and practical interest.

References

Abbas, W., Laszka, A., & Koutsoukos, X. (2017). Improving network connectivity and robustness using trusted nodes with application to resilient consensus. *IEEE Transactions on Control of Network Systems*, *5*, 2036–2048.

Abbas, Waseem, Shabbir, Mudassir, Jaleel, Hassan, & Koutsoukos, Xenofon (2020). Improving network robustness through edge augmentation while preserving strong structural controllability. In *2020 American control conference* ACC, (pp. 2544–2549). IEEE.

Abbas, Waseem, Shabbir, Mudassir, Li, Jiani, & Koutsoukos, Xenofon (2020). Interplay between resilience and accuracy in resilient vector consensus in multi-agent networks. In *Proc. of the 59th IEEE conference on decision and control* CDC, (pp. 3127–3132).

Abdelrazek, M., Grundy, J., & Müller, I. (2010). An analysis of the cloud computing security problem. In *2010 Asia Pacific cloud workshop, Vol. 50* (pp. 37–66).

Aji, S. (2014). *Estimating reachability set sizes in dynamic graphs*. Virginia Polytechnic Institute and State University.

Albert, R., Jeong, H., & Barabási, A. L. (2000). Error and attack tolerance of complex networks. *Nature*, *406*, 378–482.

Alpcan, A., & Buchegger, S. (2011). Security games for vehicular networks. *SIAM Journal on Control and Optimization*, *10*, 280–290.

Ames, A. D., Xu, X., Grizzle, J. W., & Tabuada, P. (2017). Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, *62*, 3861–3876.

An, Liwei, & Yang, Guang-Hong (2021). Byzantine-resilient distributed state estimation: A min-switching approach. *Automatica*, *129*, Article 109664.

Baggio, G., & Zampieri, S. (2018). On the relation between non-normality and diameter in linear dynamical networks. In *European control conference* (pp. 1839–1844).

Bamieh, B., Jovanovic, M. R., Mitra, P., & Patterson, S. (2012). Coherence in large-scale networks: Dimension-dependent limitations of local feedback. *IEEE Transactions on Automatic Control*, *57*, 2235–2249.

Barabasi, A. L., & Albert, R. (1999). Emergence of scaling in random networks. *Science*, *286*(5439), 509–512.

Bárány, I., & Soberón, P. (1966). A generalization of Radon's theorem. *Journal of the London Mathematical Society*, *41*, 123–128.

Baras, John S., & Liu, Xiangyang (2019). Trust is the cure to distributed consensus with adversaries. In *2019 27th mediterranean conference on control and automation* MED, (pp. 195–202). IEEE.

Barooah, Prabir, & Hespanha, Joao P. (2006). Graph effective resistance and distributed control: Spectral properties and applications. In *Proceedings of the 45th IEEE conference on decision and control* (pp. 3479–3485). IEEE.

Bemporad, A., Heemels, M., & Johansson, M. (2010). *Networked control systems*. Springer-Verlag New York Inc.

Bertsekas, D. P., & Tsitsiklis, J. N. (1989). *Parallel and distributed computation: numerical methods*. Prentice hall.

Bilge, Leyla, & Dumitraş, Tudor (2012). Before we knew it: an empirical study of zero-day attacks in the real world. In *Proceedings of the 2012 ACM conference on computer and communications security* (pp. 833–844).

Biron, Z., Dey, S., & Pisu, P. (2018). Real-time detection and estimation of denial of service attack in connected vehicle systems. *IEEE Transaction on Intelligent Transportation Systems*, *19*, 3893–3902.

Blahut, R. E. (2003). *Algebraic codes for data transmission*. Cambridge, UK: Cambridge University Press.

Bollobás, B. (2001). *Random graphs* (2nd ed.). Cambridge University Press.

Borkar, V., & Varaiya, P. (1982). Asymptotic agreement in distributed estimation. *IEEE Transactions on Automatic Control*, *27*, 650–655.

Bouzid, Z., Potop-Butucaru, M. G., & Tixeuil, S. (2010). Optimal Byzantine resilient convergence in uni-dimensional robot networks. *Theoretical Computer Science*, *411*(34), 3154–3168.

Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., Savage, S., et al. (2011). Comprehensive experimental analyses of automotive attack surfaces. In *USENIX security symposium* (pp. 447–462).

Chen, Yuan, Kar, Soummya, & Moura, José M. F. (2018a). Resilient distributed estimation: Sensor attacks. *IEEE Transactions on Automatic Control*, *64*(9), 3772–3779.

Chen, Yuan, Kar, Soummya, & Moura, Jose M. F. (2018b). Resilient distributed estimation through adversary detection. *IEEE Transactions on Signal Processing*, *66*(9), 2455–2469.

Chen, Yuan, Kar, Soummya, & Moura, J. M. (2018c). Topology free resilient distributed estimation. arXiv preprint arXiv:1812.08902.

Chen, Yuan, Kar, Soummya, & Moura, José M. F. (2019). Resilient distributed parameter estimation with heterogeneous data. *IEEE Transactions on Signal Processing*, *67*(19), 4918–4933.

Chen, Yuan, Kar, Soummya, & Moura, José M. F. (2020). Resilient distributed field estimation. *SIAM Journal on Control and Optimization*, *58*(3), 1429–1456.

Chen, W., Liu, J., Chen, Y., Zhen Khong, S., Wang, D., Başar, T., et al. (2016). Characterizing the positive semidefiniteness of signed Laplacians via effective resistances. In *IEEE 55th conference on decision and control* (pp. 985–990).

Chong, Michelle S., Sandberg, Henrik, & Teixeira, André M. H. (2019). A tutorial introduction to security and privacy for cyber-physical systems. In *2019 18th European control conference* ECC, (pp. 968–978). IEEE.

Chong, Michelle S., Wakaiki, Masashi, & Hespanha, Joao P. (2015). Observability of linear systems under adversarial attacks. In *Proceedings of the 2015 American control conference* (pp. 2439–2444). IEEE.

Chow, E. Y., & Willsky, A. S. (1984). Analytical redundancy and the design of robust failure detection systems. *IEEE Transactions on Automatic Control*, *29*, 603–614.

Chung, F. (1997). *Spectral graph theory*. American Mathematical Society.

Clark, A., Alomair, B., Bushnell, L., & Poovendran, R. (2016). *Submodularity in dynamics and control of networked systems*. Springer.

Crucitti, P., Latora, V., & Marchiori, M. (2004). Model for cascading failures in complex networks. *Physical Review E*, *69*.

Dahan, Mathieu, Sela, Lina, & Amin, Saurabh (2022). Network inspection for detecting strategic attacks. *Operations Research*, *70*(2), 1008–1024.

Darouach, M., Zasadzinski, M., & Xu, S. J. (1994). Full-order observers for linear systems with unknown inputs. *IEEE Transactions on Automatic Control*, *39*, 606–609.

De Persis, C., & Isidori, A. (2001). A geometric approach to nonlinear fault detection and isolation. *IEEE Transactions on Automatic Control*, *46*, 853–865.

De Persis, C., & Tesi, P. (2014). On resilient control of nonlinear systems under denial-of-service. In *Proceedings of IEEE conference on decision and control* (pp. 5254–5259).

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

del Nozal, Álvaro Rodríguez, Millán, Pablo, Orihuela, Luis, Seuret, Alexandre, & Zaccarian, Luca (2019). Distributed estimation based on multi-hop subspace decomposition. *Automatica*, *99*, 213–220.

Dibaji, S. M., & Ishii, H. (2017). Resilient consensus of second-order agent networks: Asynchronous update rules with delays. *Automatica*, *81*, 123–132.

Dibaji, S. M., Ishii, H., & Tempo, R. (2018). Resilient randomized quantized consensus. *IEEE Transactions on Automatic Control*, *63*, 2508–2522.

Dibaji, S. M., Pirani, M., Flamholz, D., Annaswamy, A. M., Johansson, K. H., & Chakrabortty, A. (2019). A systems and control perspective of CPS security. *Annual Reviews in Control*.

Dion, J. M., Commault, C., & van der Woude, J. (1999). The generic number of invariant zeros of a structured linear system. *SIAM Journal on Control and Optimization*, *38*(1), 1–21.

Dion, J. M., Commault, C., & van der Woude, J. (2003). Generic properties and control of linear structured systems: a survey. *Automatica*, *39*(7), 1125–1144.

Dolev, Danny (1982). The Byzantine generals strike again. *Journal of Algorithms*, *3*(1), 14–30.

Dolev, D., Dwork, C., Waarts, O., & Yung, M. (1993). Perfectly secure message transmission. *Journal of the Association for Computing Machinery*, *40*(1), 17–47.

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

Dörfler, F., & Bullo, F. (2014). Synchronization in complex networks of phase oscillators: A survey. *Automatica*, *50*(6), 1539–1564.

Dwork, C. (2008). Differential privacy: A survey of results. In *International conference on theory and applications of models of computation* (pp. 1–19).

Ericsson, G. N. (2010). Cyber security and power system communication essential parts of a smart grid infrastructure. *IEEE Transactions on Power Delivery*, *25*, 1501–1507.

Falliere, N., Murchu, L., & Chien, E. (2013). W32. Stuxnet dossier: Symantec security response. https://www.symantec.com.

Farina, L., & Rinaldi, S. (2000). *Positive linear systems: theory and applications*. John Wiley & Sons.

Farokhi, F., Shames, I., & Batterham, N. (2017). Secure and private control using semi-homomorphic encryption. *Control Engineering Practice*, *67*, 13–20.

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

Figura, Martin, Kosaraju, Krishna Chaitanya, & Gupta, Vijay (2021). Adversarial attacks in consensus-based multi-agent reinforcement learning. In *Proceedings of the 2021 American control conference* (pp. 3050–3055). IEEE.

Fischer, M. J., Lynch, N. A., & Paterson, M. S. (1985). Impossibility of distributed consensus with one faulty process. *Journal of the ACM*, *32*, 374–382.

Fitch, K. E., & Leonard, N. E. (2015). Joint centrality distinguishes optimal leaders in noisy networks. *IEEE Transactions on Control of Network Systems*, *3*, 366–378.

Friedman, J. (2003). A proof of Alon's second eigenvalue conjecture. In *Proceedings of the thirty fifth annual ACM symposium on theory of computing* (pp. 720–724).

Garcia-Teodoro, P., Diaz-Verdejo, J., Macia-Fernandez, G., & Vazquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers and Security*, *28*, 18–28.

Gazi, V., & Passino, K. M. (2003). Stability analysis of swarms. *IEEE Transactions on Automatic Control*, *48*(4), 692–697.

Gentz, R., Wu, S. X., Wai, H., Scaglione, A., & Leshem, A. (2016). Data injection attacks in randomized gossiping. *IEEE Transactions on Signal and Information Processing over Networks*, *2*, 523–538.

Ghaderi, J., & Srikant, R. (2014). Opinion dynamics in social networks: A local interaction game with stubborn agents. *Automatica*, *50*, 3209–3215.

Ghosh, A., Boyd, S., & Saberi, A. (2008). Minimizing effective resistance of a graph. *SIAM Review*, *50*, 37–66.

Ghosh, Avishek, Hong, Justin, Yin, Dong, & Ramchandran, Kannan (2019). Robust federated learning in a heterogeneous environment. arXiv preprint arXiv: 1906.06629.

Gil, Stephanie, Kumar, Swarun, Mazumder, Mark, Katabi, Dina, & Rus, Daniela (2017). Guaranteeing spoof-resilient multi-robot networks. *Autonomous Robots*, *41*(6), 1383–1400.

Giraldo, Jairo, Sarkar, Esha, Cardenas, Alvaro A., Maniatakos, Michail, & Kantarcioglu, Murat (2017). Security and privacy in cyber-physical systems: A survey of surveys. *IEEE Design & Test*, *34*(4), 7–17.

Godsil, C., & Royle, G. (2001). *Algebraic graph theory*. Springer.

Guan, Y., & Saif, M. (1991). A novel approach to the design of unknown input observers. *IEEE Transactions on Automatic Control*, *36*, 632–635.

Guerrero-Bonilla, L., Saldana, D., & Kumar, V. (2018). Design guarantees for resilient robot formations on lattices. *IEEE Robotics and Automation Letters*, *4*, 89–96.

Gueye, A., & Marbukh, V. (2012). A game-theoretic framework for network security vulnerability assessment and mitigation. In *International conference on decision and game theory for security*.

Gupta, Nirupam, Doan, Thinh T., & Vaidya, Nitin H. (2021). Byzantine fault-tolerance in decentralized optimization under 2f-redundancy. In *2021 American control conference* (pp. 3632–3637). IEEE.

Gupta, Nirupam, & Vaidya, Nitin H. (2020). Fault-tolerance in distributed optimization: The case of redundancy. In *Proceedings of the 39th symposium on principles of distributed computing* (pp. 365–374).

Hadjicostis, C. N., & Touri, R. (2002). Feedback control utilizing packet dropping network links. In *Proceedings of the 41st IEEE conference on decision and control* (pp. 1205–1210).

Hamann, H. (2018). *Swarm robotics: A formal approach*. New York: Springer International Publishing.

Han, Z., Niyato, D., Saad, W., Başar, T., & Hjorungnes, A. (2012). *Game theory in wireless and communication networks: theory, models, and applications*. Cambridge university press.

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

Hao, H., & Barooah, P. (2013). Stability and robustness of large platoons of vehicles with double-integrator models and nearest neighbor interaction. *International Journal of Robust and Nonlinear Control*, *23*, 2097–2122.

He, Y., Mendis, G. J., & Wei, J. (2016). Real-time detection of false data injection attacks in smart grids: A deep learning-based intelligent mechanism. *IEEE Transactions on Smart Grid*, *8*, 1–12.

He, X., Ren, X., Sandberg, H., & Johansson, K. H. (2021). How to secure distributed filters under sensor attacks. *IEEE Transactions on Automatic Control*, *67*(6), 2843–2856.

He, Q., Wang, C., Cui, G., Li, B., Zhou, R., Zhou, Q., et al. (2021). A game-theoretical approach for mitigating edge DDoS attack. *IEEE Transactions on Dependable and Secure Computing*.

Heemels, W., Johansson, K. H., & Tabuada, P. (2012). An introduction to event-triggered and self-triggered control. In *Proceedings of IEEE conference on decision and control* (pp. 3270–3285).

Hespanha, J. P., Naghshtabrizi, P., & Xu, Y. (2007). A survey of recent results in networked control systems. *Proceedings of the IEEE*, *95*(1), 138–162.

Hooshyar, A., & Iravani, R. (2017). Microgrid protection. *Proceedings of the IEEE*, *105*, 1332–1353.

Hou, Y., Li, J., & Pan (2003). On the Laplacian eigenvalues of signed graphs. *Linear and Multilinear Algebra*, *51*, 21–30.

Huang, T., Satchidanandan, B., Kumar, P. R., & Xie, L. (2018). An online detection framework for cyber attacks on automatic generation control. *IEEE Transactions on Power Systems*, *33*, 6816–6827.

Huber, Peter J. (1972). The 1972 wald lecture robust statistics: A review. *The Annals of Mathematical Statistics*, *43*(4), 1041–1067.

Hwang, Inseok, Kim, Sungwan, Kim, Youdan, & Seah, Chze Eng (2009). A survey of fault detection, isolation, and reconfiguration methods. *IEEE Transactions on Control Systems Technology*, *18*(3), 636–653.

Ishii, H., Wang, Y., & Feng, S. (2022). An overview on multi-agent consensus under adversarial attacks. *Annual Reviews in Control*.

Jackson, M. (2010). *Social and economic networks*. Princeton University Press.

Jadbabaie, A., Lin, J., & Morse, A. S. (2003). Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, *48*, 988–1001.

Jadbabaie, Ali, Molavi, Pooya, Sandroni, Alvaro, & Tahbaz-Salehi, Alireza (2012). Non-Bayesian social learning. *Games and Economic Behavior*, *76*(1), 210–225.

Jadbabaie, Ali, Molavi, Pooya, & Tahbaz-Salehi, Alireza (2013). *Information heterogeneity and the speed of learning in social networks*: *Columbia Business School Research Paper*, (pp. 13–28).

Jia, J., van Waarde, H. J., Trentelman, H. L., & Camlibel, M. K. (2021). A unifying framework for strong structural controllability. *IEEE Transactions on Automatic Control*, *66*, 391–398.

Jiang, Z., Balu, A., Hegde, C., & Sarkar, S. (2017). Collaborative deep learning in fixed topology networks. *Advances in Neural Information Processing Systems*, 5904–5914.

Kailath, T. (1980). *Linear systems*. Englewood Cliffs, NJ, USA: Prentice-Hall.

Kalman, R. E., Ho, Y. C., & Narendra, S. K. (1963). Controllability of linear dynamical systems. *Contributions to Differential Equations*, *1*, 189–213.

Kar, Soummya, & Moura, José M. F. (2011). Convergence rate analysis of distributed gossip (linear parameter) estimation: Fundamental limits and tradeoffs. *IEEE Journal of Selected Topics in Signal Processing*, *5*(4), 674–690.

Kar, Soummya, & Moura, Jose M. F. (2013). Consensus+ innovations distributed inference over networks: cooperation and sensing in networked systems. *IEEE Signal Processing Magazine*, *30*(3), 99–109.

Kar, Soummya, Moura, José M. F., & Ramanan, Kavita (2012). Distributed parameter estimation in sensor networks: Nonlinear observation models and imperfect communication. *IEEE Transactions on Information Theory*, *58*(6), 3575–3605.

Kieckhafer, R. M., & Azadmanesh, M. H. (1994). Reaching approximate agreement with mixed mode faults. *IEEE Transactions on Parallel and Distributed Systems*, *5*, 53–63.

Kim, Taekyoo, Lee, Chanhwa, & Shim, Hyungbo (2019). Completely decentralized design of distributed observer for linear systems. *IEEE Transactions on Automatic Control*.

Koetter, R., & Médard, M. (2003). An algebraic approach to network coding. *IEEE/ACM Transactions on Networking*, *11*, 782–795.

Konečný, J., McMahan, H. B., Ramage, D., & Richtárik, P. (2016). Federated optimization: Distributed machine learning for on-device intelligence. arXiv preprint arXiv:1610.02527.

Koscher, K., Czeskis, A., Roesner, F., Patel, S., Kohno, Tadayoshi, Checkoway, S., et al. (2010). Experimental security analysis of a modern automobile. In *IEEE symposium on security and privacy* (pp. 447–462). IEEE.

Kushleyev, A., Mellinger, D., Powers, C., & Kumar, V. (2013). Towards a swarm of agile micro quadrotors. *Autonomous Robots*, *35*, 287–300.

Kuwaranancharoen, Kananart, & Sundaram, Shreyas (2018). On the location of the minimizer of the sum of two strongly convex functions. In *IEEE conference on decision and control* (pp. 1769–1774).

Kuwaranancharoen, Kananart, Xin, Lei, & Sundaram, Shreyas (2020). Byzantine-resilient distributed optimization of multi-dimensional functions. In *2020 American control conference* (pp. 4399–4404). IEEE.

Lalitha, Anusha, Javidi, Tara, & Sarwate, Anand (2018). Social learning and distributed hypothesis testing. *IEEE Transactions on Information Theory*, *64*(9).

Lamport, L., Shostak, R., & Pease, M. (1982). The Byzantine generals problem. *ACM Transactions on Programming Language Systems*, *4*(2), 382–401.

LeBlanc, Heath J., & Hassan, Firas (2014). Resilient distributed parameter estimation in heterogeneous time-varying networks. In *Proceedings of the 3rd international conference on high confidence networked systems* (pp. 19–28).

LeBlanc, H., & Koutsoukos, X. (2011). Consensus in networked multi-agent systems with adversaries. In *HSCC*, *Proceedings of the 14th international conference on Hybrid systems: computation and control* (pp. 281–290).

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

Li, Z., Shahidehpour, M., & Aminifar, F. (2017). Cybersecurity in distributed power systems. *Proceedings of the IEEE*, *105*, 1367–1388.

Liang, G., Zhao, J., Luo, F., Weller, S. R., & Yang Dong, Z. (2017). A review of false data injection attacks against modern power systems. *IEEE Transactions on Smart Grids*, *8*, 1630–1638.

Lin, Yixuan, Gade, Shripad, Sandhu, Romeil, & Liu, Ji (2020). Toward resilient multi-agent actor-critic algorithms for distributed reinforcement learning. In *2020 American control conference* (pp. 3953–3958). IEEE.

Lin, I. C., & Liao, T. C. (2017). A survey of blockchain security issues and challenges. *International Journal of Network Security*, *19*, 653–659.

Liu, Qipeng, Fang, Aili, Wang, Lin, & Wang, Xiaofan (2014). Social learning with time-varying weights. *Journal of Systems Science and Complexity*, *27*(3), 581–593.

Liu, X., Pequito, S., Kar, S., Mo, Y., Sinopoli, B., & Aguiar, A. Pedro (2013). Minimum robust sensor placement for large scale linear time-invariant systems: A structured systems approach. In *4th IFAC workshop on distributed estimation and control in networked systems* (pp. 417–424).

Lun, Yuriy Zacchia, D'Innocenzo, Alessandro, Smarra, Francesco, Malavolta, Ivano, & Di Benedetto, Maria Domenica (2019). State of the art of cyber-physical systems security: An automatic control perspective. *Journal of Systems and Software*, *149*, 174–216.

Lynch, N. A. (1997). *Distributed algorithms*. San Francisco, California: Morgan Kaufmann Publishers Inc.

Manshaei, M., Zhu, Q., Alpcan, T., Basar, T., & Hubaux, J. P. (2013). Game theory meets network security and privacy. *ACM Computing Surveys*, *45*, 53–73.

Manyena, Bernard, O'Brien, Geoff, O'Keefe, Phil, & Rose, Joanne (2011). Disaster resilience: a bounce back or bounce forward ability? *Local Environment: The International Journal of Justice and Sustainability*, *16*(5), 417–424.

Mao, Yanwen, Diggavi, Suhas, Fragouli, Christina, & Tabuada, Paulo (2020). Secure state-reconstruction over networks subject to attacks. *IEEE Control Systems Letters*, *5*(1), 157–162.

Mao, Yanwen, & Tabuada, Paulo (2021). Decentralized resilient state-tracking. In *Proceedings of the 2021 60th IEEE conference on decision and control* (pp. 3480–3485). IEEE.

Marsden, P. V., & Friedkin, N. E. (1993). *Network studies of social influence*. Sociological Methods & Research.

Massoumnia, M. A., Verghese, G. C., & Willsky, A. S. (1989). Failure detection and identification. *IEEE Transactions on Automatic Control*, *34*, 316–321.

Matousek, J. (2002). *Lectures on discrete geometry*. Springer.

Milošević, Jezdimir, Dahan, Mathieu, Amin, Saurabh, & Sandberg, Henrik (2019). A network monitoring game with heterogeneous component criticality levels. In *2019 IEEE 58th conference on decision and control* CDC, (pp. 4379–4384). IEEE.

Mitra, Aritra, Ghawash, Faiq, Sundaram, Shreyas, & Abbas, Waseem (2021). On the impacts of redundancy, diversity, and trust in resilient distributed state estimation. *IEEE Transactions on Control of Network Systems*.

Mitra, Aritra, Hassani, Hamed, & Pappas, George (2021). Exploiting heterogeneity in robust federated best-arm identification. arXiv preprint arXiv:2109.05700.

Mitra, Aritra, Richards, John A., Bagchi, Saurabh, & Sundaram, Shreyas (2019). Resilient distributed state estimation with mobile agents: overcoming Byzantine adversaries, communication losses, and intermittent measurements. *Autonomous Robots*, *43*(3), 743–768.

Mitra, Aritra, Richards, John A., Bagchi, Saurabh, & Sundaram, Shreyas (2021). Distributed state estimation over time-varying graphs: Exploiting the age-of-information. *IEEE Transactions on Automatic Control*, *67*(12), 6349–6365.

Mitra, Aritra, Richards, John A., & Sundaram, Shreyas (2019). A new approach for distributed hypothesis testing with extensions to Byzantine-resilience. In *Proc. of the American control conference* (pp. 261–266).

Mitra, Aritra, Richards, John A., & Sundaram, Shreyas (2020). A new approach to distributed hypothesis testing and non-Bayesian learning: Improved learning rate and Byzantine-resilience. *IEEE Transactions on Automatic Control*.

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

Mitra, A., & Sundaram, S. (2019). Byzantine-resilient distributed observers for LTI systems. *Automatica*, *108*.

Mo, Y., Weerakkody, S., & Sinopoli, B. (2015). Physical authentication of control systems: Designing watermarked control inputs to detect counterfeit sensor outputs. *IEEE Control Systems Magazine*, *35*, 93–109.

Mokdad, L., Ben-Othman, J., & Nguyen, A. T. (2015). DJAVAN: Detecting jamming attacks in Vehicle Ad hoc Networks. *Performance Evaluation*, *87*, 47–59.

Momani, M., & Challa, S. (2010). Survey of trust models in different network domains. arXiv preprint arXiv:1010.0168.

Motwani, R., & Raghavan, P. (1996). Randomized algorithms. *ACM Computing Surveys*, *28*.

Mousavi, S. S., Haeri, M., & Mesbahi, M. (2017). Robust strong structural controllability of networks with respect to edge additions and deletions. In *American control conference* (pp. 5007–5012).

Mousavi, S. S., Haeri, M., & Mesbahi, M. (2018). On the structural and strong structural controllability of undirected networks. *IEEE Transactions on Automatic Control*, *63*, 2234–2241.

Mulzer, W., & Werner, D. (2013). Approximating tverberg points in linear time for any fixed dimension. *Discrete & Computational Geometry*, *50*, 520–535.

Muñoz-González, Luis, Co, Kenneth T., & Lupu, Emil C. (2019). Byzantine-robust federated machine learning through adaptive model averaging. arXiv preprint arXiv:1909.05125.

Musleh, A., Chen, G., & Yang Dong, Z. (2020). A survey on the detection algorithms for false data injection attacks in smart grids. *IEEE Transactions on Smart Grids*, *11*, 2218–2234.

Nedić, Angelia, Olshevsky, Alex, & Uribe, César A. (2017). Fast convergence rates for distributed Non-Bayesian learning. *IEEE Transactions on Automatic Control*, *62*(11), 5538–5553.

Nedic, A., & Ozdaglar, A. (2009). Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, *54*, 48–61.

Newman, M. J. (2001). Clustering and preferential attachment in growing networks. *Physical Review E*, *64*(2), Article 025102.

Newman, M. E. j., Strogatz, S. H., & Watts, D. J. (2001). Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, *64*.

Ny, J. L., & Pappas, G. J. (2014). Differentially private filtering. *IEEE Transactions on Automatic Control*, *59*, 341–354.

Olfati-Saber, R., Fax, J. A., & Murray, R. M. (2007). Consensus and cooperation in networked multi-agent systems. *IEEE Transactions on Automatic Control*, *95*, 215–233.

Olfati-Saber, R., & Murray, R. M. (2004). Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, *49*, 1520–1533.

Olshevsky, A. (2014). Minimal controllability problems. *IEEE Transactions on Control of Network Systems*, 249–258.

Olshevsky, A. (2017). On (non) supermodularity of average control energy. *IEEE Transactions on Control of Network Systems*, *5*, 1177–1181.

Parandehgheibi, M., & Modiano, E. (2013). Robustness of interdependent networks: The case of communication networks and the power grid. In *IEEE global communications conference* (pp. 2164–2169).

Park, H., & Hutchinson, S. A. (2017). Fault-tolerant rendezvous of multirobot systems. *IEEE Transactions on Robotics*, *33*, 565–582.

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

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

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

Pasqualetti, F., Favaretto, C., Zhao, S., & Zampieri, S. (2018). Fragility and controllability tradeoff in complex networks. In *American control conference* (pp. 216–221).

Pasqualetti, F., Zampieri, S., & Bullo, F. (2014). Controllability metrics, limitations and algorithms for complex networks. *IEEE Transactions on Control of Network Systems*, *1*, 40–52.

Patterson, S., & Bamieh, B. (2010). Leader selection for optimal network coherence. In *49th IEEE conference on decision and control* (pp. 2692–2697).

Pease, Marshall, Shostak, Robert, & Lamport, Leslie (1980). Reaching agreement in the presence of faults. *Journal of the ACM*, *27*(2), 228–234.

Pequito, S., Kar, S., & Aguiar, A. P. (2015). On the complexity of the constrained input selection problem for structural linear systems. *Automatica*, *62*, 193–199.

Pequito, S., Kar, S., & Aguiar, A. P. (2016). A framework for structural input/output and control configuration selection in large-scale systems. *IEEE Transactions on Automatic Control*, *61*, 303–318.

Pequito, S., Ramos, G., Kar, S., Aguiar, A. P., & Ramos, J. (2017). The robust minimal controllability problem. *Automatica*, *82*, 261–268.

Pillutla, Krishna, Kakade, Sham M., & Harchaoui, Zaid (2022). Robust aggregation for federated learning. *IEEE Transactions on Signal Processing*, *70*, 1142–1154.

Pirani, M., Hashemi, E., Khajepour, A., Fidan, B., Litkouhi, B., Chen, S. K., et al. (2019). Cooperative vehicle speed fault diagnosis and correction. *IEEE Transactions on Intelligent Transportation Systems*, *20*(2), 783–789.

Pirani, M., Nekouie, E., Sandberg, H., & Johansson, K. H. (2021a). A game-theoretic framework for security-aware sensor placement problem in networked control systems. *IEEE Transactions on Automatic Control*, *67*, 3699–3706.

Pirani, M., Nekouie, E., Sandberg, H., & Johansson, K. H. (2021b). A graph-theoretic equilibrium analysis of attacker-defender game on consensus dynamics under H_2 performance metric. *IEEE Transactions on Network Science and Engineering*, *8*, 1991–2000.

Pirani, M., Shahrivar, E. Moradi, Fidan, B., & Sundaram, S. (2018). Robustness of leader - follower networked dynamical systems. *IEEE Transaction on Control of Network Systems*, *5*(4), 1752–1763.

Pirani, M., Shahrivar, E. M., & Sundaram, S. (2015). Coherence and convergence rate in networked dynamical systems. In *Proceedings of CDC 2015, the 54th IEEE conference on decision and control* (pp. 968–973).

Pirani, M., Simpson-Porco, J. W., & Fidan, B. (2017). System-theoretic performance metrics for low-inertia stability of power networks. In *56th IEEE conference on decision and control* (pp. 5106–5111).

Pirani, M., & Sundaram, S. (2014). Spectral properties of the grounded Laplacian matrix with applications to consensus in the presence of stubborn agents. In *American control conference* (pp. 2160–2165).

Pirani, Mohammad, & Sundaram, Shreyas (2016). On the smallest eigenvalue of grounded Laplacian matrices. *IEEE Transactions on Automatic Control*, *61*(2), 509–514.

Pirani, M., & Taylor, J. (2020). Controllability of AC power networks with DC lines. *IEEE Transactions on Power Systems*.

Poolla, B. K., Bolognani, S., & Dörfler, F. (2015). Optimal placement of virtual inertia in power grids. *IEEE Transactions on Automatic Control*, *62*, 6209–6220.

Popli, N., Pequito, S., Kar, S., Pedro, A., & Ilić, M. (2019). Selective strong structural minimum-cost resilient co-design for regular descriptor linear systems. *Automatica*, *102*, 80–85.

Prorok, A., Malencia, M., Carlone, L., Sukhatme, G. S., Sadler, B. M., & Kumar, V. (2021). Beyond robustness: A taxonomy of approaches towards resilient multi-robot systems. arXiv preprint arXiv:2109.12343.

Rahmani, A., Ji, M., Mesbahi, M., & Egerstedt, M. (2009). Controllability of multi-agent systems from a graph-theoretic perspective. *SIAM Journal on Control and Optimization*, *48*, 162–186.

Ramos, G., Aguiar, A. P., & Pequito, S. (2022). An overview of structural systems theory. *Automatica*, *140*, Article 110229.

Ravi, N., Scaglione, A., & Nedic, A. (2019). A case of distributed optimization in adversarial environment. In *IEEE international conference on acoustics, speech and signal processing* (pp. 5252–5256).

Reay, J. R. (1968). An extension of Radon's theorem. *Illinois Journal of Mathematics*, *12*, 184–189.

Rego, Francisco F. C., Aguiar, A. Pedro, Pascoal, António M., & Jones, Colin N. (2017). A design method for distributed Luenberger observers. In *Proc. of the 56th IEEE conference on decision and control* (pp. 3374–3379).

Rego, F., Pascoal, A., Aguiar, P., & Jones, C. (2019). Distributed state estimation for discrete-time linear time invariant systems: A survey. *Annual Reviews in Control*, *48*, 36–56.

Ren, W., & Beard, R. W. (2005). Consensus seeking in multi agent systems under dynamically changing interaction topologies. *IEEE Transactions on Automatic Control*, *50*, 655–661.

Renganathan, Venkatraman, & Summers, Tyler (2017). Spoof resilient coordination for distributed multi-robot systems. In *2017 international symposium on multi-robot and multi-agent systems* (pp. 135–141). IEEE.

Saldana, D., Prorok, A., Sundaram, S., Campos, M. F. M., & Kumar, V. (2017). Resilient consensus for time-varying networks of dynamic agents. In *American control conference, Vol. 65* (pp. 252–258).

Salem, Malek Ben, Hershkop, Shlomo, & Stolfo, Salvatore J. (2008). A survey of insider attack detection research. In *Insider attack and cyber security: Beyond the hacker* (pp. 69–90). Springer.

Sandell, N., Varaiya, P., Athans, M., & Safonov, M. (1978). Survey of decentralized control methods for large scale systems. *IEEE Transactions on Automatic Control*, *23*, 108–128.

Saulnier, K., Saldana, D., Prorok, A., Pappas, G. J., & Kumar, V. (2017). Resilient flocking for mobile robot teams. *IEEE Robotics and Automation Letters*, *2*, 1039–1046.

Shabbir, M., Li, J., Abbas, W., & Koutsoukos, X. (2020a). Resilient vector consensus in multi-agent networks using centerpoints. In *American control conference* (pp. 4387–4392).

Shabbir, M., Li, J., Abbas, W., & Koutsoukos, X. (2020b). Resilient vector consensus in multi-agent networks using centerpoints. In *Proceedings of the 2020 American control conference, Vol. 33* (pp. 4387–4392).

Shahrampour, Shahin, Rakhlin, Alexander, & Jadbabaie, Ali (2016). Distributed detection: Finite-time analysis and impact of network topology. *IEEE Transactions on Automatic Control*, *61*(11), 3256–3268.

Shahrivar, E. M., Pirani, M., & Sundaram, S. (2017). Spectral and structural properties of random interdependent networks. *Automatica*, *83*, 234–242.

Singer-Cohen, K. (1995). *Random intersection graphs* (Ph.D. thesis), Department of Mathematical Sciences, The Johns Hopkins University.

Sinopoli, B., Schenato, L., Franceschetti, M., Poolla, K., Jordan, M. I., & Sastry, S. S. (2004). Kalman filtering with intermittent observations. *IEEE Transactions on Automatic Control*, *49*, 1453–1464.

Sklaroff, J. R. (1976). Redundancy management technique for space shuttle computers. *IBM Journal of Research and Development*, *20*, 20–28.

Slay, Jill, & Miller, Michael (2007). Lessons learned from the maroochy water breach. In *Proceedings of international conference on critical infrastructure protection* (pp. 73–82).

Smith, Bruce W., Dalen, Jeanne, Wiggins, Kathryn, Tooley, Erin, Christopher, Paulette, & Bernard, Jennifer (2008). The brief resilience scale: assessing the ability to bounce back. *International Journal of Behavioral Medicine*, *15*, 194–200.

Sridhar, S., Hahn, A., & Govindarasu, M. (2011). Cyber–physical system security for the electric power grid. *Proceedings of the IEEE*, *100*, 210–224.

Stankovic, J. A. (2014). Research directions for the Internet of things. *IEEE Internet of Things Journal*, 3–9.

Su, L., & Shahrampour, S. (2020). Finite-time guarantees for Byzantine-resilient distributed state estimation with noisy measurements. *IEEE Transactions on Automatic Control*, *65*, 3758–3771.

Su, Lili, & Vaidya, Nitin H. (2016). Defending Non-Bayesian learning against adversarial attacks. *Distributed Computing*, 1–13.

Su, L., & Vaidya, N. H. (2021). Byzantine-resilient multiagent optimization. *IEEE Transactions on Automatic Control*, *66*, 2227–2233.

Summers, T. H., Cortesi, F. L., & Lygeros, J. (2015). On submodularity and controllability in complex dynamical networks. *IEEE Transactions on Control of Network Systems*, *1*, 91–101.

Summers, Tyler, Shames, Iman, Lygeros, John, & Dörfler, Florian (2015). Topology design for optimal network coherence. In *Proc. of the IEEE European control conference* ECC, (pp. 575–580).

Sun, L., Dou, Y., Yang, C., Wang, J., Yu, P. S., He, L., et al. (2022). Adversarial attack and defense on graph data: A survey. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*.

Sun, J., & Motter, A. E. (2013). Controllability transition and nonlocality in network control. *Physical Review Letters*, *110*, Article 208701.

Sundaram, S., & Gharesifard, B. (2018). Distributed optimization under adversarial nodes. *IEEE Transactions on Automatic Control*, *64*, 1063–1076.

Sundaram, S., & Hadjicostis, C. N. (2008). Distributed function calculation and consensus using linear iterative strategies. *IEEE Journal on Selected Areas in Communications*, *26*, 650–660.

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

Sundaram, S., & Hadjicostis, C. (2013). Structural controllability and observability of linear systems over finite fields with applications to multi-agent systems. *IEEE Transactions on Automatic Control*, *58*, 60–73.

Tan, K., & Grigoriadis, K. M. (2001). Stabilization and H_∞ control of symmetric systems: an explicit solution. *Systems & Control Letters*, *44*, 57–72.

Tegling, E., Bamieh, B., & Gayme, D. F. (2015). The price of synchrony: Evaluating the resistive losses in synchronizing power networks. *IEEE Transactions on Control of Network Systems*, *2*, 254–266.

Tegling, Emma, Mitra, Partha, Sandberg, Henrik, & Bamieh, Bassam (2019). On fundamental limitations of dynamic feedback control in regular large-scale networks. *IEEE Transactions on Automatic Control*, *64*(12), 4936–4951.

Teixeira, A., Amin, S., Sandberg, H., Johansson, K. H., & Sastry, S. S. (2010). Cyber security analysis of state estimators in electric power systems. In *Proceedings of the 2010 IEEE conference on decision and control* (pp. 5991–5998).

Teixeira, Andre, Sou, Kin Cheong, Sandberg, Henrik, & Johansson, Karl Henrik (2015). Secure control systems: A quantitative risk management approach. *IEEE Control Systems Magazine*, *35*(1), 24–45.

Ten, C. W., Liu, C. C., & Manimaran, G. (2008). Vulnerability assessment of cybersecurity for SCADA systems. *IEEE Transactions on Power Delivery*, *23*, 1836–1846.

Teneketzis, D., & Varaiya, P. (1984). Consensus in distributed estimation with inconsistent beliefs. *Systems & Control Letters*, *4*, 217–221.

Tomlin, C., Pappas, G. J., & Sastry, S. (1998). Conflict resolution for air traffic management: A study in multiagent hybrid systems. *IEEE Transactions on Automatic Control*, *43*, 509–521.

Tsai, C.-F., Hsu, Y.-F., Lin, C.-Y., & Lin, W.-Y. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. In *Intrusion detection by machine learning: A review, Vol. 36* (pp. 11994–12000).

Tsitsiklis, J. (1984). *Problems in decentralized decision making and computation*. Massachusetts Inst. of Tech. Cambridge Lab for Information and Decision Systems.

Ugrinovskii, Valery (2011). Distributed robust filtering with H_∞ consensus of estimates. *Automatica*, *47*(1), 1–13.

Ugrinovskii, V. (2019). Distributed H_∞ estimation resilient to biasing attacks. *IEEE Transactions on Control of Network Systems*.

Ugrinovskii, Valery, & Langbort, Cédric (2011). Distributed H_∞ consensus-based estimation of uncertain systems via dissipativity theory. *IET Control Theory & Applications*, *5*(12), 1458–1469.

Uribe, César A., Hare, James Z., Kaplan, Lance, & Jadbabaie, Ali (2019). Non-Bayesian social learning with uncertain models over time-varying directed graphs. In *Proc. of the 58th IEEE conference on decision and control* (pp. 3635–3640).

Usevitch, J., & Panagou, D. (2017). R-robustness and (r, s)-robustness of circulant graphs. In *Annual conference on decision and control* (pp. 4416–4421).

Vaidya, N. H. (2014). Iterative Byzantine vector consensus in incomplete graphs. *Distributed Computing and Networking*, 14–28.

Vaidya, Nitin H., & Garg, Vijay K. (2013). Byzantine vector consensus in complete graphs. In *Proceedings of the 2013 ACM symposium on principles of distributed computing* (pp. 65–73).

Vaidya, Nitin H., Tseng, Lewis, & Liang, Guanfeng (2012). Iterative approximate Byzantine consensus in arbitrary directed graphs. In *Proceedings of the 2012 ACM symposium on principles of distributed computing* (pp. 365–374).

Waagen, A., Verma, G., Chan, K., Swami, A., & D'Souza, R. (2015). Effect of zealotry in high-dimensional opinion dynamics models. *Physical Review E*, *91*, Article 022811.

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

Weerakkody, S., Liu, X., & Sinopoli, B. (2017). Robust structural analysis and design of distributed control systems to prevent zero dynamics attacks. In *Proceedings of the 56th annual conference on decision and control* (pp. 1356–1361).

Weerakkody, S., & Sinopoli, B. (2015). Detecting integrity attacks on control systems using a moving target approach. In *IEEE conference on decision and control* (pp. 5820–5826).

Wei, J., & van der Schaft, A. J. (2013). Load balancing of dynamical distribution networks with flow constraints and unknown in/outflows. *Systems & Control Letters*, *62*, 1001–1008.

West, D. B. (2001a). *Introduction to graph theory*. Upper Saddle River, New Jersey: Prentice-Hall Inc..

West, D. B. (2001b). *Introduction to graph theory*. Upper Saddle River, NJ: Prentice-Hall.

Witsenhausen, H. (1968). A counterexample in stochastic optimum control. *SIAM Journal on Control*, *6*, 131–147.

Wu, Bo, Carr, Steven, Bharadwaj, Suda, Xu, Zhe, & Topcu, Ufuk (2020). Resilient distributed hypothesis testing with time-varying network topology. In *Proceedings of the 2020 American control conference* (pp. 1483–1488). IEEE.

Wu, Zhaoxian, Shen, Han, Chen, Tianyi, & Ling, Qing (2021). Byzantine-resilient decentralized TD learning with linear function approximation. In *Proceedings of the 2021 IEEE international conference on acoustics, speech and signal processing* (pp. 5040–5044).

Xiang, Z., & Vaidya, N. H. (2016). Brief announcement: Relaxed Byzantine vector consensus. In *Proceedings of the 28th ACM symposium on parallelism in algorithms and architectures* (pp. 401–403).

Xie, Le, Choi, Dae-Hyun, Kar, Soummya, & Poor, H. Vincent (2012). Fully distributed state estimation for wide-area monitoring systems. *IEEE Transactions on Smart Grid*, *3*(3), 1154–1169.

Xie, L., Mo, Y., & Sinopoli, B. (2011). Integrity data attacks in power market operations. *IEEE Transactions on Smart Grid*, *2*, 659–666.

Xie, Yijing, Mou, Shaoshuai, & Sundaram, Shreyas (2021). Towards resilience for multi-agent QD-learning. In *Proceedings of the 60th IEEE conference on decision and control* (pp. 1250–1255).

Yagan, O., Qian, D., Zhang, J., & Cochran, D. (1998). Optimal allocation of interconnecting links in cyber-physical systems: Interdependence, cascading failures, and robustness. *IEEE Transactions on Parallel and Distributed Systems*, *23*, 1708–1720.

Yan, J., Mo, Y., Li, X., & Wen, C. (2020). A "safe kernel" approach for resilient multi-dimensional consensus. In *21st IFAC world congress, Vol. 55* (pp. 2507–2512).

Yang, Z., & Bajwa, W. U. (2019). Byrdie: Byzantine-resilient distributed coordinate descent for decentralized learning. *IEEE Transactions on Signal and Information Processing over Networks*.

Yang, Zhixiong, Gang, Arpita, & Bajwa, Waheed U. (2020). Adversary-resilient distributed and decentralized statistical inference and machine learning: An overview of recent advances under the Byzantine threat model. *IEEE Signal Processing Magazine*, *37*(3), 146–159.

Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology*, *10*, 1–19.

Yazıcıoglu, A. Y., Abbas, W., & Egerstedt, M. (2016). Graph distances and controllability of networks. *IEEE Transactions on Automatic Control*, *61*, 4125–4130.

Ye, Lintao, Roy, Sandip, & Sundaram, Shreyas (2020). Resilient sensor placement for Kalman filtering in networked systems: Complexity and algorithms. *IEEE Transactions on Control of Network Systems*, *7*(4), 1870–1881.

Ye, Lintao, Woodford, Nathaniel, Roy, Sandip, & Sundaram, Shreyas (2020). On the complexity and approximability of optimal sensor selection and attack for Kalman filtering. *IEEE Transactions on Automatic Control*, *66*(5), 2146–2161.

Yemini, Michal, Nedić, Angelia, Gil, Stephanie, & Goldsmith, Andrea J. (2022). Resilience to malicious activity in distributed optimization for cyberphysical systems. In *2022 IEEE 61st conference on decision and control* CDC, (pp. 4185–4192). IEEE.

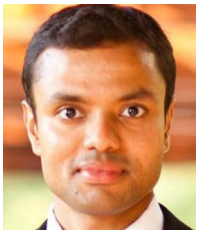
Yemini, Michal, Nedić, Angelia, Goldsmith, Andrea J., & Gil, Stephanie (2021). Characterizing trust and resilience in distributed consensus for cyberphysical systems. *IEEE Transactions on Robotics*, *38*(1), 71–91.

Yildiz, E., Acemoglu, D., Ozdaglar, A., Saberi, A., & Scaglione, A. (2011). Discrete opinion dynamics with stubborn agents. *Operations Research*.

Young, G. F., Scardovi, L., & Leonard, N. E. (2016). A new notion of effective resistance for directed graphs—Part I: Definition and properties. *IEEE Transactions on Automatic Control*, *61*, 1727–1736.
Zhang, H., Ayoub, R., & Sundaram, S. (2017). Sensor selection for Kalman filtering of linear dynamical systems: Complexity, limitations and greedy algorithms. *Automatica*, *78*, 202–210.
Zhang, H., Fata, E., & Sundaram, S. (2015). A notion of robustness in complex networks. *IEEE Transactions on Control of Network Systems*, *2*, 310–320.
Zhang, C., Sun, J., Zhu, X., & Fang, Y. (2010). Privacy and security for online social networks: challenges and opportunities. *IEEE Network*, *24*, 13–18.
Zhang, Haotian, & Sundaram, Shreyas (2012). Robustness of information diffusion algorithms to locally bounded adversaries. In *Proceedings of the 2012 American control conference* (pp. 5855–5861). IEEE.
Zhao, J., Yağan, O., & Gligor, V. (2014). On the strengths of connectivity and robustness in general random intersection graphs. In *IEEE conference on decision and control* (pp. 3661–3668).
Zhu, Q., & Basar, T. (2015). Game-theoretic methods for robustness, security, and resilience of cyberphysical control systems: Games-in-games principle for optimal cross-layer resilient control systems. *IEEE Control Systems*, *35*, 45–65.
Zugner, D., & Gunnemann, S. (2019). Certifiable robustness and robust training for graph convolutional networks. *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery and data mining*, 246–256.

Mohammad Pirani is an assistant professor with the Department of Mechanical Engineering, University of Ottawa, Canada. He was a research assistant professor in the Department of Mechanical and Mechatronics Engineering at the University of Waterloo (2022–2023). He held postdoctoral researcher positions at the University of Toronto (2019–2021) and KTH Royal Institute of Technology, Sweden (2018–2019). He received a MASc degree in electrical and computer engineering and a Ph.D. degree in Mechanical and Mechatronics Engineering, both from the University of Waterloo in 2014 and 2017, respectively. His research interests include resilient and fault-tolerant control, networked control systems, and multi-agent systems.

Aritra Mitra received the B.E. degree from Jadavpur University, Kolkata, India, in 2013, the M.Tech. degree from the Indian Institute of Technology Kanpur, Kanpur, India, in 2015, and the Ph.D. degree from Purdue University, West Lafayette, IN, USA, in 2020, all in Electrical engineering. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC, USA. He was a Postdoctoral Researcher at the University of Pennsylvania from 2020 to 2022. His current research interests include control theory, optimization, statistical signal processing, machine learning, and distributed algorithms. Dr. Mitra was a recipient of the University Gold Medal at Jadavpur University and the Academic Excellence Award at IIT Kanpur.



Shreyas Sundaram received the M.S. and Ph.D. degrees in electrical engineering from the University of Illinois at Urbana-Champaign, Champaign, IL, USA, in 2005 and 2009, respectively. He is currently an Associate Professor with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA. He was a Postdoctoral Researcher with the University of Pennsylvania from 2009 to 2010, and an Assistant Professor with the Department of Electrical and Computer Engineering, University of Waterloo from 2010 to 2014. 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. Dr. Sundaram is a recipient of the National Science Foundation 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.