Detection of Nonrandom Sign-Based Behavior for Resilient Coordination of Robotic Swarms

Paul J Bonczek, Rahul Peddi, Shijie Gao, and Nicola Bezzo

***Abstract*—Cooperative multirobot systems coordinate their motion by exchanging information through consensus schemes to achieve a common goal. In the event of stealthy cyber attacks, compromised measurements and communication broadcasts can hijack a portion or the entire system toward undesired states. However, in order for these attacks to be effective, they have to exhibit nonrandom characteristics that contradict the expected multirobot system behavior. To deal with these hidden attacks, we propose a runtime monitoring framework that considers the signed *residual*, defined as the difference between the expected and the received information to identify and isolate unexpected nonrandom behavior within the multirobot system. Specifically, the technique that we propose—named *Cumulative Sign* detector—monitors and compares changes in signed values of residual with their expected occurrences to detect inconsistencies and trigger alarms when an attack is discovered. Our results are validated theoretically by providing detection bounds and are demonstrated with simulations and experiments on swarms of unmanned ground vehicles under different attacks in comparison with state-of-the-art residual-based detection schemes.**

***Index Terms*—Attack detection, distributed robot systems, multirobot systems, swarms.**

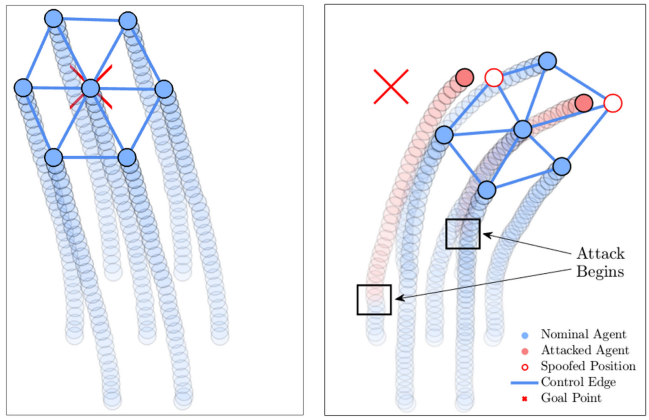


Fig. 1. Pictorial motivation of the problem investigated in this article: in nominal conditions, (a) and (b) i.e., with no attack, a multiagent system can reach the desired goal (red "X"), whereas in the presence of an attack [red disks in (b)], the system is hijacked away.

I. Introduction

MANY advancements in sensing, control, planning, mobility, and networking have enhanced mobile robotic systems allowing precise and robust autonomous operations that were unthinkable until only recently. Within robotics, multiagent system coordination and swarming have long been studied and are gaining back attention, thanks to the many technological advances, but this also brings upon security issues. Multiagent systems are typically used to perform coordinated tasks in a distributed fashion. This collaborative nature allows for numerous applications that would be more difficult or not possible to perform with just a single agent, such as factory and warehouse logistics [1], vehicle platooning [2], connected vehicle-to-vehicle operations [3], [4], surveillance [5], disaster relief [6], and exploration missions [7].

With such benefits in multirobot systems, however, comes the risk of cyber attacks. In fact, all the aforementioned applications are typically designed without considering cyber-security issues, assuming that all the actors (i.e., other robots) in the multirobot settings are cooperative. In the presence of a compromised robot in the network, liveness (i.e., the ability to perform and complete correctly a task) and safety (i.e., avoid collisions or reaching undesired states) properties can be violated. The presence of malicious actors in a network can potentially manipulate the entire multirobot system, hijacking a mission and potentially leading the system toward undesired states, as pictorially represented in Fig. 1.

Such situations can be caused by compromised communications, which results in incorrect sharing of information between robots, or by manipulated sensor measurements, leading compromised robots to react to altered on-board signals that are also broadcast to surrounding neighbors. In a successful hijacking attempt, an attacker is able to implement a *stealthy* attack sequence to degrade system performance, all while remaining hidden from detection. The term stealthy has been adopted in a wide range of attack scenarios on stochastic systems, such

Manuscript received July 28, 2021; accepted December 9, 2021. Date of publication January 19, 2022; date of current version February 8, 2022. This paper was recommended for publication by Associate Editor A. Prorok and Editor P. Robuffo Giordano upon evaluation of the reviewers' comments. This work was supported in part by the National Science Foundation under Grant 1816591 and in part by the Office of Naval Research under Grant N000141712012. *(Corresponding author: Paul J Bonczek.)*

Paul J Bonczek, Shijie Gao, and Nicola Bezzo are with the Charles L. Brown Department of Electrical and Computer Engineering and the Link Lab, University of Virginia, Charlottesville, VA 22904 USA (e-mail: pjb4xn@virginia.edu; sg9dn@virginia.edu; nbezzo@virginia.edu).

Rahul Peddi is with the Department of Engineering Systems and Environment and the Link Lab, University of Virginia, Charlottesville, VA 22904 USA (e-mail: rp3cy@virginia.edu).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TRO.2021.3139592.

Digital Object Identifier 10.1109/TRO.2021.3139592

1552-3098 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: University of Virginia Libraries. Downloaded on February 19,2022 at 19:19:55 UTC from IEEE Xplore. Restrictions apply.

as in zero-dynamics [8], replay [9], zero-alarm [10], and hidden [11] attack cases. In this article, the term *stealthy* indicates an attack sequence that mimics normal (attack-free) behavior of traditional detection schemes (i.e., a hidden attack [11]), where attackers leverage the noise characteristics within a multirobot system to evade detection during a hijacking attempt. To discover such attacks, the key principle that we leverage is that an attacker attempting to hijack one or more robots within a multirobot system via stealthy sensor and/or communication attacks will inherently exhibit nonrandom/inconsistent behaviors in order to be effective, contradicting an expected behavior of the system model. Specifically, in this article, we monitor the *residual*—which is defined as the difference between a measured/received value and the predicted/expected value—in order to discover inconsistent behavior due to these hijacking attempts. Our proposed monitoring scheme—which we name the Cumulative Sign (CUSIGN) detector—differs from other residual-based detectors [12]–[20] as its purpose is to monitor for inconsistencies in signed behavior (i.e., nonrandomness) of the residual in multirobot systems. Once an attack is discovered, we propose a framework to 1) isolate the compromised robots and 2) reconfigure the network to continue the desired task.

A. Related Work

The topic of resilience of multiagent systems has received extensive consideration in the engineering and computer science communities recently [21]. Much attention has gone into resilience of these systems based on network connectivity, determined by the underlying graph topology of the network [22]. A widely used method for multiagent resilience is through consensus protocols that leverage the mean subsequence reduced (MSR) algorithms [23]–[27], in which all vehicles in a network come to an agreement on a global variable of interest (e.g., velocity, position, and heading angle). Such consensus protocols are resilient to F number of compromised (e.g., noncooperative) agents, which rely on network topologies that satisfy the $(2F+1)$ robustness properties, in which every agent in the network follows the strategy of diminishing the effect of potentially deceptive information due to cyber attacks or faults by ignoring up to F agents with shared values that contrast the most from its own value of the global consensus variable. As noted by Wang and Ishii [28], the purpose of MSR algorithms is not to detect misbehaving (i.e., compromised) agents in a network, but rather to simply leave out values consisting of the greatest difference in magnitude.

An example of misbehaving agent detection in multiagent networks was presented by Chen *et al.* [29] that propose the Flag Raising Distributed Estimator such that each agent in the network estimates an unknown parameter by an iterative algorithm that leverages both its own sensor measurement and its neighbor's estimate of the parameter to detect the presence of adversarial agents. As a neighbor's parameter estimate differs from an agent's own parameter beyond a chosen threshold, the neighbor is deemed adversarial, thus raising a flag. Zhao *et al.* [30] utilize agents as mobile detectors that allow for isolation of any malicious agents that collude with each other in an attempt to take advantage of network connectivity constraints. Another example can be found in [31], where every uncompromised agent can detect and isolate misbehaving agents in leader–follower and leaderless consensus networked systems. Each agent employs a multiphase reputation-based protocol by relying on local observations and adaptive consensus weight updates on neighbors to allow for resilient convergence of uncompromised agents in the formation. Taking a different approach to detection, Khazraei *et al.* [32] propose a network-wide shared watermarking signal that is applied to control inputs of each agent in multirobot systems; then, a residual-based anomaly detection scheme is used to find any misbehaving agents. Lee and Min [33] leverage the residual-based *Cumulative Sum* (CUSUM) anomaly detector, first characterized in [34], to discover spoofs to on-board navigation systems of robots in multirobot systems, thus allowing the mobile robot team to arrive at its desired destination. Different from the aforementioned works, our proposed decentralized framework considers deceptive cyber attacks that intentionally hide within the uncertainties to avoid detection from traditional residual-based detection procedures in multirobot systems consisting of stochastic linear time-invariant (LTI) modeled agents.

This work builds on previous research considering deceptive cyber attacks to systems by injecting false data while trying to remain undetected within system noise [12]. Previous works have analyzed the effects of malicious sensor attacks on individual systems leveraging the Kalman filter for state estimation [35]. Similarly, Kwon *et al.* [16] characterize how undetected attacks compromise closed-loop systems that utilize the Kalman filter in terms of state and system dynamic degradation.

Several attack detection techniques exist in the literature that also analyze the residual and leverage an alarm-based procedure, one of which is Bad Data (BD) detection [12] that monitors each element within the residual vector and triggers an alarm any time the residual element extends beyond a chosen threshold value. Another popular method is compound scalar testing (CST) [16] that reduces the residual vector into a scalar test measure of chi-square distribution. An improvement of CST in [17] is made by including a coding matrix to sensor outputs that is unknown to attackers to improve detection capabilities of stealthy attacks. Furthermore, Murguia and Ruths [13], [14] formalize a model-based detector of the CUSUM algorithm that is commonly used as a monitor for change detection, by leveraging known characteristics of the system dynamical and noise models to provide a desired alarm rate during operation. While these traditional alarm-based detection methods offer compelling performance for discovering attacks, an intelligent attacker may be able to exploit system uncertainties (e.g., measurement noises) to evade detection by emulating an expected alarm-based behavior. In order for an attack to be effective (e.g., degrade system performance) while hiding within system uncertainties, it must inherently create inconsistent nonrandom signed behavior of the residual.

In our recent work, we have presented techniques to detect nonrandom residual behavior due to sensor spoofing attacks, like in [36]–[38]. In [36], the Wilcoxon Signed Rank and Serial Independence Runs statistical tests [39], [40] were leveraged to find inconsistencies within a windowed sequence of residual

Authorized licensed use limited to: University of Virginia Libraries. Downloaded on February 19,2022 at 19:19:55 UTC from IEEE Xplore. Restrictions apply.

data. Furthermore, we characterized the CUSIGN detector [37] on a single system, inspired by the CUSUM procedure in [34], with the purpose of finding nonrandom residual behavior by checking for changes in occurrence of the signed measurement residual values while leveraging a chi-square detection scheme. Additionally, in [37], we demonstrated the detection capabilities of CUSIGN when compared to the model-based CUSUM procedure (also utilizing the chi-square scheme) [13] in the presence of stealthy sensor attacks that intentionally hide within system noises. In this transaction, we expand on these works by further developing runtime techniques to monitor for nonrandom residual behavior and detect inconsistencies within multirobot systems due to cyber attacks.

B. Contribution

This article has the following contributions. We propose a novel residual-based attack detection scheme for multirobot systems to find nonrandom residual behaviors due to stealthy communication and sensor attacks that are undetectable by current state-of-the-art residual-based methods. We then present a decentralized framework, in which each robotic agent acts independently by leveraging local information received from nearby robots while employing the proposed detection scheme to enable resilient control of the multirobot system during stealthy attacks and reconfigure the network to maintain connectivity once one or more compromised robots have been isolated from the network. While we present the proposed framework in a general sense, as a case study, we consider cooperative autonomous multirobot applications that leverage virtual spring–damper mesh (VSDM) physics for decentralized formation control [41]–[45]. Our proposed framework, however, can be used in any proximity-based consensus formation control (e.g., nearest neighbors [46]). Finally, we validate the proposed scheme on ample MATLAB and Robot Operating System (ROS) simulations and experiments on swarms of unmanned ground vehicles (UGVs).

The remainder of this article is organized as follows. In Section II we begin by introducing the preliminaries and problem formulation, followed by Section III, where we characterize the residuals within a multirobot system and the CUSIGN attack detector for detection of inconsistent (i.e., nonrandom) residual behaviors. In Section IV, we describe the framework for resilient coordination of the multirobot system against stealthy sensor and communication attacks to maintain desired system performance. Numerical simulations and experiment results using UGVs are presented to verify our framework in Section V. Finally, Section VI concludes this article.

II. Preliminaries

Let us consider a multirobot system with N mobile robots that maintains a proximity-based formation during a mission. Such a system can be described using a *directed* graph, where each directed edge represents the control influence on a robot due to the proximity of a neighboring robot in the system. The directed graph describing the multirobot system is modeled as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the set of vertices $\mathcal{V} = \{1, 2, \ldots, N\}$ denote the mobile robots and the set of edges $\mathcal{E} = \{(i, j) \,|\, i, j \in \mathcal{V}\}$ are control links between robots. An edge $(i, j) \in \mathcal{E}$ means that the control input of robot i is affected by the state of robot j within the proximity-based formation.

Each of the robots is modeled as an LTI dynamical agent as follows:

$$\dot{\boldsymbol{x}}_i = \boldsymbol{A}\boldsymbol{x}_i + \boldsymbol{B}\boldsymbol{u}_i + \boldsymbol{\nu}_i, \qquad i = 1, 2, \ldots, N \tag{1}$$

where $\boldsymbol{x}_i \in \mathbb{R}^n$ is the state vector, $\boldsymbol{u}_i \in \mathbb{R}^m$ is the control input, $\boldsymbol{A}$ and $\boldsymbol{B}$ are state and input matrices with appropriate dimensions, respectively, and $\boldsymbol{\nu}_i \in \mathbb{R}^n$ is the zero-mean Gaussian process uncertainty. The robots successfully achieve tasks by performing a proximity-based consensus protocol $\psi(\cdot)$, in which all robots $i \in \mathcal{V}$ agree on a decentralized control input $\boldsymbol{u}_i \in \mathbb{R}^m$ that follows:

$$\boldsymbol{u}_i = \psi\left(\boldsymbol{x}_i, \boldsymbol{x}_j, \mathcal{O}_i\right), \qquad i = 1, 2, \ldots, N \tag{2}$$

where $\boldsymbol{x}_i$ is the state of robot i, $\boldsymbol{x}_j$ represents the states of the neighboring robots j, $j \neq i$, and the set $\mathcal{O}_i$ denotes any nearby obstacles of robot i that are utilized for obstacle avoidance.

To enable the robot network to satisfy the consensus-based control protocol in order to accomplish tasks, the robots exchange necessary information (e.g., state vector) with each other. The set $\mathcal{I} = \{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_N\}$ describes the information broadcast within the multirobot system that is available to any robots within communication range $\delta_c > 0$. When all robots are cooperative, the mobile team is able to complete the desired task at hand, where inputs are computed based on information received from nearby robots.

Definition 1 (Communication graph): Given the N robots in set $\mathcal{V}$ with a communication range δ_c, we define the graph $\mathcal{G}_\mathcal{C} = (\mathcal{V}, \mathcal{E}_\mathcal{C})$ with the edge set represented by

$$\mathcal{E}_\mathcal{C} = \left\{(i, j) \;\middle|\; \left\|\boldsymbol{p}_i - \boldsymbol{p}_j\right\| \leq \delta_c,\ i, j \in \mathcal{V}\right\} \tag{3}$$

as the *communication graph* of the robot set $\mathcal{V}$, where $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ are position coordinates (within the state vector) of robots $i, j \in \mathcal{V}$, $i \neq j$.

The set of all neighboring robots within communication range of a robot i, as defined by the communication graph, is represented by

$$\mathcal{C}_i = \left\{j \in \mathcal{V} \;\middle|\; (i, j) \in \mathcal{E}_\mathcal{C}\right\}. \tag{4}$$

Definition 2 (Control graph): Each robot $i \in \mathcal{V}$ leverages the received information to form a neighbor set $\mathcal{S}_i \subseteq \mathcal{C}_i$ for consensus control purposes to maintain a desired proximity from other robots. We define the graph $\mathcal{G}_\mathcal{U} = (\mathcal{V}, \mathcal{E}_\mathcal{U})$ with the edge set represented by

$$\mathcal{E}_\mathcal{U} = \left\{(i, j) \;\middle|\; j \in \mathcal{S}_i, \forall i \in \mathcal{V}\right\} \tag{5}$$

as the *control graph* of the robot set $\mathcal{V}$.

Given that the robot states are not directly available, each robot i is equipped with N_s on-board sensors that provide sampled state measurements every $t_s \in \mathbb{R}_+$ seconds as indicated by the output vector given by

$$\boldsymbol{y}_i^{(k)} = \boldsymbol{C}\boldsymbol{x}_i^{(k)} + \boldsymbol{\eta}_i^{(k)} \in \mathbb{R}^{N_s} \tag{6}$$

Authorized licensed use limited to: University of Virginia Libraries. Downloaded on February 19,2022 at 19:19:55 UTC from IEEE Xplore. Restrictions apply.

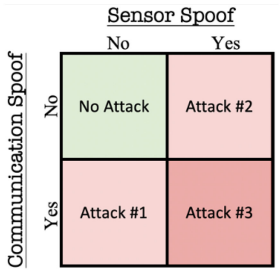


Fig. 2. Classes of attacks considered in this article.

with the output matrix $\boldsymbol{C}$ and the measurement uncertainty vector $\boldsymbol{\eta}_i^{(k)}$ at every time instant $k \in \mathbb{N}$. The process and measurement uncertainties of all robots are described in discrete time as multivariate zero-mean Gaussian distributed noise with covariance matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$, respectively. A Kalman filter, with gain matrix $\boldsymbol{K}_i^{(k)} \in \mathbb{R}^{n \times N_s}$, is implemented on-board each robot i to provide discrete-time state estimates $\hat{\boldsymbol{x}}_i^{(k|k)} \in \mathbb{R}^n$ using a discretized dynamical model of (1).

A. Attack Model

Summarized in Fig. 2 are the cyber attacks considered in this article, which are a combination of on-board sensor and/or communication spoofs that can maliciously affect any robot within the multirobot system. Next, we provide a brief description for each of the considered cyber attack scenarios.

(A1)—*Communication attack:* In this attack, an attacker intercepts and replaces broadcast data such that the receiver and sender data are different (e.g., a man-in-the-middle attack [47]). We assume that an attacker is able to intercept communication broadcasts replacing the message with modified, yet plausible information. As an example, the sender of a communication broadcast that is being attacked may not be aware of the attack in which a receiver is obtaining falsified data. For the case studies investigated in this article, the exchanged information $\mathcal{I}^{(k)}$ at each time instant k between robots is assumed to be state estimates, inputs, and measurements. We will indicate the spoofed broadcast information $\mathcal{I}_i^{(k)} \to \widetilde{\mathcal{I}}_i^{(k)}$ from a robot i as

$$\widetilde{\mathcal{I}}_i^{(k)} = \left\{ \hat{\boldsymbol{x}}_i^{(k|k)} + \boldsymbol{\xi}_{i,x}^{(k)},\, \boldsymbol{u}_i^{(k)} + \boldsymbol{\xi}_{i,u}^{(k)},\, \boldsymbol{y}_i^{(k)} + \boldsymbol{\xi}_{i,y}^{(k)} \right\} \tag{7}$$

where in the presence of an attack, at least one of the following conditions is true: $\boldsymbol{\xi}_{i,x}^{(k)} \neq \boldsymbol{0} \in \mathbb{R}^n$, $\boldsymbol{\xi}_{i,u}^{(k)} \neq \boldsymbol{0} \in \mathbb{R}^m$, and $\boldsymbol{\xi}_{i,y}^{(k)} \neq \boldsymbol{0} \in \mathbb{R}^{N_s}$, resulting in $\mathcal{I}_i^{(k)} \neq \widetilde{\mathcal{I}}_i^{(k)}$.

(A2)—*Sensor spoofing:* The second attack that we consider is sensor spoofing, in which an adversary manipulates on-board sensor measurements as follows:

$$\widetilde{\boldsymbol{y}}_i^{(k)} = \boldsymbol{y}_i^{(k)} + \boldsymbol{\xi}_{i,y}^{(k)} \tag{8}$$

where $\boldsymbol{\xi}_{i,y}^{(k)} \in \mathbb{R}^{N_s}$ is the attack vector that describes false data injections to sensor measurements. An attacker manipulating on-board sensor will be able to drive the state estimate of the robot away from its true state, leading to unreliable on-board control decisions and, consequently, diverting neighboring robots whose control actions are based on inaccurate position information received from the compromised robot.

(A3)—*Coordinated attack:* This is a combination of the previous two cases, in which attacks hide within the expected system behavior acting and hiding in a coordinated way on both sensing (A1) and communication (A2) constraints. The compromised robot in this case is able to perform a completely different operation while reporting plausible data to neighbor robots.

For each of the attack vectors ($\boldsymbol{\xi}_{i,x}^{(k)}, \boldsymbol{\xi}_{i,u}^{(k)}$, and $\boldsymbol{\xi}_{i,y}^{(k)}$), an attacker is assumed to be capable of leveraging both process and measurement uncertainties $\boldsymbol{Q}$ and $\boldsymbol{R}$, to construct attacks that emulate the expected behavior of measurements and communication broadcasts that can fool traditional residual-based attack detection techniques.

B. Problem Formulation

In this work, we consider a typical scenario, in which robots in a multirobot system coordinate their motion in a decentralized fashion to maintain a desired formation while navigating toward a given goal. The challenge is to provide a resilient approach for the multirobot system to continue these operations in the presence of cyber attacks that are intentionally hiding within system noises while attempting to hijack the multirobot system.

Problem 1 (Detection of inconsistencies in multirobot systems): Consider a set of N homogeneous robots $\mathcal{V}$ in a multirobot system. Design a decentralized policy for each robot $i \in \mathcal{V}$ to detect at runtime inconsistencies from any neighbor $j \neq i$ due to cyber attacks on: 1) sensor measurements, i.e., if the following holds:

$$\mathbb{E}[\boldsymbol{y}_j - \hat{\boldsymbol{y}}_{ij}] \neq 0 \tag{9}$$

or 2) the communication channel when the received state from j is different from the predicted state computed by i

$$\mathbb{E}[\boldsymbol{x}_j - \hat{\boldsymbol{x}}_{ij}] \neq 0 \tag{10}$$

where $\hat{\boldsymbol{y}}_{ij}$ and $\hat{\boldsymbol{x}}_{ij}$ are measurement and state predictions of j made by robot i, respectively.

To detect inconsistent behavior of neighboring robots, we employ an attack detection scheme that monitors inter-robot residuals (i.e., the comparison between received and predicted information) for unexpected behavior within the robot network. Upon detection, the system needs to isolate and reconfigure to continue its planned operation. Formally, we have the following.

Problem 2 (Multirobot system recovery): Find a decentralized policy for each robot $i \in \mathcal{V}$ to isolate and remove any maliciously attacked robot j from its neighbor set for control $\mathcal{S}_i$ that presents inconsistent behavior flagged by solving Problem 1, i.e., to obtain

$$\mathcal{S}_i' = \mathcal{S}_i \setminus \{j\}. \tag{11}$$

With the malicious robot j removed from any neighbor set $\mathcal{S}_i'$, the robot j is no longer able to influence the control of i.

III. Nonrandom Behavior Detection of Residuals

In this section, we first characterize both the on-board and inter-robot residuals that are monitored for nonrandom (i.e.,

Authorized licensed use limited to: University of Virginia Libraries. Downloaded on February 19,2022 at 19:19:55 UTC from IEEE Xplore. Restrictions apply.

inconsistent) behavior due to cyber attacks within multirobot systems. We then formalize the detection procedure that searches for nonrandom behavior in the residual sequences, before describing an attack sequence that an intelligent attacker must take to avoid detection.

A. Residual Characterizations

In our proposed detection framework within multirobot systems, each robot $i \in \mathcal{V}$ monitors its on-board measurement residual for discovery of sensor attacks as well as two types of inter-robot residuals to identify inconsistent behavior of communication broadcasts or sensor information that are received from neighboring robots j within the control graph, i.e., $(i,j) \in \mathcal{E}_{\mathcal{U}}$. Let us define the *on-board measurement residual* vector $\boldsymbol{r}_i^{(k)}$ on a robot i as

$$\boldsymbol{r}_i^{(k)} = \boldsymbol{y}_i^{(k)} - \boldsymbol{C}\hat{\boldsymbol{x}}_i^{(k|k-1)} \in \mathbb{R}^{N_s} \tag{12}$$

to monitor for on-board sensor attacks, which has an expected covariance matrix $\boldsymbol{\Sigma}_i^{(k)} = \mathbb{E}[\boldsymbol{r}_i^{(k)}(\boldsymbol{r}_i^{(k)})^{\mathsf{T}}] = \boldsymbol{C}\boldsymbol{P}_i^{(k|k-1)}\boldsymbol{C}^{\mathsf{T}} + \boldsymbol{R}$ during attack-free conditions, with $\boldsymbol{P}_i^{(k|k-1)}$ denoting the prediction error covariance. Each sth on-board measurement residual element is normally distributed as follows:

$$\mathbb{E}[r_{i,s}^{(k)}] = 0, \quad \mathrm{Var}[r_{i,s}^{(k)}] = \left(\sigma_{i,s}^{(k)}\right)^2 \tag{13}$$

where $(\sigma_{i,s}^{(k)})^2$ is the sth diagonal element of the on-board measurement residual covariance matrix $\boldsymbol{\Sigma}_i^{(k)}$.

In our proposed multirobot monitoring framework, each robot $i \in \mathcal{V}$ monitors its neighbors for consistent behavior by computing state predictions of each neighbor $j \in \mathcal{S}_i$ using their received state $\hat{\boldsymbol{x}}_j^{(k|k)}$ and input $\boldsymbol{u}_j^{(k)}$ information by

$$\hat{\boldsymbol{x}}_{ij}^{(k+1|k)} = \boldsymbol{A}_d\hat{\boldsymbol{x}}_j^{(k|k)} + \boldsymbol{B}_d\boldsymbol{u}_j^{(k)} \in \mathbb{R}^n \tag{14}$$

where $\boldsymbol{A}_d$ and $\boldsymbol{B}_d$ are discrete-time equivalents of the known robot dynamical model in (1). A robot i leverages these state predictions by comparing them to the received state and measurement information from neighboring robots. Let us define the *inter-robot state residual* by the following:

$$\breve{\boldsymbol{r}}_{ij}^{(k)} = \hat{\boldsymbol{x}}_j^{(k|k)} - \hat{\boldsymbol{x}}_{ij}^{(k|k-1)} \in \mathbb{R}^n \tag{15}$$

which enables a robot i to monitor for consistent state and input information from a robot j. Each qth element $q \in \{1,\ldots,n\}$ of the inter-robot state residual vector (15) is normally distributed as follows:

$$\mathbb{E}[\breve{r}_{ij,q}^{(k)}] = 0, \quad \mathrm{Var}[\breve{r}_{ij,q}^{(k)}] = \sum_{s=1}^{N_s}\left(K_{j,(q,s)}^{(k)}\sigma_{j,s}^{(k)}\right)^2 \tag{16}$$

with $K_{j,(q,s)}^{(k)}$ representing the element at the qth row and the sth column of the Kalman gain at time k on robot j. Additionally, robots compute the *inter-robot measurement residual*

$$\boldsymbol{r}_{ij}^{(k)} = \boldsymbol{y}_j^{(k)} - \boldsymbol{C}\hat{\boldsymbol{x}}_{ij}^{(k|k-1)} \in \mathbb{R}^{N_s} \tag{17}$$

to discover sensor attacks that may be occurring on the neighboring robot. The inter-robot measurement residual shares the expected zero-mean normally distributed characteristics of the on-board measurement residual in (13). Note that in order for a robot i to compute inter-robot residuals of a robot j at a time k in (15) and (17), a state prediction (14) must be made at the previous time $k-1$.

For ease of notation throughout the remaining of this section, we exclude subscripts i on any on-board measurement residual and ij for inter-robot residuals between robots i and j. Moreover, we further simplify notation by referring the on-board and inter-robot residual vector elements $s \in \{1,\ldots,N_s\}$ and $q \in \{1,\ldots,n\}$, respectfully, as the variable $r^{(k)}$, as all residuals are zero-mean normally distributed during nominal nonattacked conditions.

A robot that is operating in normal conditions will have an expected occurrence of signed residual characteristics over time. With these considerations in mind, we propose a detector to analyze the sign of incoming residuals within multirobot systems to determine whether the residual behavior follows the expected random behavior. This technique, which we name the CUSIGN detector, is unique to previous state-of-the-art residual-based detectors [12]–[20] in that instead of monitoring for magnitude changes, it relies on the sign of a residual variable within an expected distribution in order to discover stealthy cyber attacks that may remain hidden within noisy systems. Since the magnitude of a residual variable is overlooked, the CUSIGN detector is nonparametric in nature and can be used on any known distribution (see, e.g., [37]). Next, we briefly introduce the technique used for alarm rate estimation before characterizing our alarm-based attack detector.

The signed residual: In normal operating conditions, i.e., in the absence of attacks defined in (A1)–(A3), the signed value of both the measurement and state residuals have an expected probability of being higher or lower than their expected values $\mathbb{E}[r^{(k)}] = 0$. The signed residual probabilities $\Pr(\cdot)$ are computed based on the expected residual distributions characterized in Section III-A by the following:

$$\begin{aligned}\Pr\left(r^{(k)} < \mathbb{E}[r^{(k)}]\right) &= \Phi\left(\mathbb{E}[r^{(k)}]\right)\\ \Pr\left(r^{(k)} > \mathbb{E}[r^{(k)}]\right) &= 1 - \Phi\left(\mathbb{E}[r^{(k)}]\right)\end{aligned} \tag{18}$$

where $\Phi(\cdot)$ is the *cumulative distribution function* of the standard normal distribution [48]. The sign of $r^{(k)}$ with respect to the reference $\mathbb{E}[r^{(k)}]$ follows:

$$\mathrm{sgn}(r^{(k)}) = \begin{cases} 1, & \textbf{if } r^{(k)} > \mathbb{E}[r^{(k)}] \\ 0, & \textbf{if } r^{(k)} = \mathbb{E}[r^{(k)}] \\ -1, & \textbf{if } r^{(k)} < \mathbb{E}[r^{(k)}] \end{cases} \tag{19}$$

such that the probability of each scenario occurring is

$$\begin{aligned}\Pr\left(\mathrm{sgn}(r^{(k)}) = 1\right) &= p_+\\ \Pr\left(\mathrm{sgn}(r^{(k)}) = 0\right) &= 0\\ \Pr\left(\mathrm{sgn}(r^{(k)}) = -1\right) &= p_- = 1 - p_+\end{aligned} \tag{20}$$

Authorized licensed use limited to: University of Virginia Libraries. Downloaded on February 19,2022 at 19:19:55 UTC from IEEE Xplore. Restrictions apply.

where $p_+ = p_- = \frac{1}{2}$ for a zero-mean normally distributed residual from (13) and (16), as the mean and median are equal. The CUSIGN detector leverages the expected probabilities $\Pr(r^{(k)} > \mathbb{E}[r^{(k)}]) = p_+$ and $\Pr(r^{(k)} < \mathbb{E}[r^{(k)}]) = p_-$ in determining nonrandom behavior in the presence of attacks.

Alarm rate estimation: In the design of the nonrandomness detector, alarms are triggered during operation to aid in determining if a system is behaving normally. In our case of a multirobot network, the robots leverage this alarm-based method for self-detection and to monitor the residual sequence of their neighbors for inconsistent behaviors. Given a robot that is not under attack, the frequency at which these alarms are triggered should follow an expected alarm rate. We employ a windowless method, which we name *memoryless runtime estimator* (MRE), for computing the alarm rate estimate utilizing a "pseudo-window" length ℓ. The runtime update equation of the MRE for alarm rate estimation follows:

$$\hat{A}^{(k)} = \hat{A}^{(k-1)} + \frac{\left[\zeta^{(k)} - \hat{A}^{(k-1)}\right]}{\ell} \tag{21}$$

where $\zeta^{(k)} \in \{0, 1\}$ is the alarm, $\hat{A}^{(k)} \in [0, 1]$ is an estimated alarm rate at every time instant k, and $\hat{A}^{(0)} = \mathbb{E}[A]$ initially at $k = 0$, where $\mathbb{E}[A] \in [0, 1]$ is the expected alarm rate (to be characterized for CUSIGN in Section III-B). The resulting alarm rate estimate can be approximated to a normal distribution when $\ell \geq 10$, as demonstrated in [37], with a resulting variance that shares properties of the exponential moving average [49].

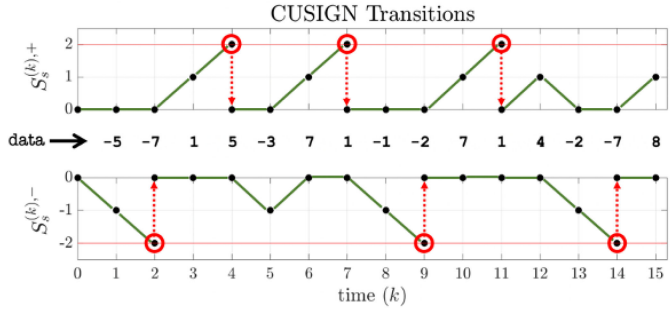


Fig. 3. Example of transitions for the CUSIGN test variable $S^{(k),\pm}$ with a threshold $\tau = 2$ given a sequence of data.

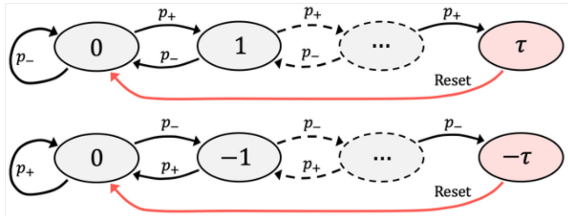


Fig. 4. Markov chain for both positive (top) and negative (bottom) cases of the CUSIGN test variable sequence with triggering threshold states in red.

B. CUSIGN Detector

To detect information inconsistencies (i.e., nonrandomness) in multirobot systems due to cyber attacks, we leverage the CUSIGN attack detector that analyzes residuals to determine whether nonrandom behavior is occurring. The CUSIGN detector monitors the residual over the sequence of time and outputs an alarm when a threshold is reached, which is then sent to the MRE to provide an updated alarm rate estimate. For any given user-defined threshold, an expected alarm rate can be found that is independent of the system model.

The CUSIGN procedure is an accumulation of signed residual values by two CUSIGN test variables $S^{(k),+}$ and $S^{(k),-}$, where each signifies a test variable at time instant k. Each test variable checks for changes in the probability for the signed residual value: one for *positive* and the other for *negative* changes. The following procedure summarizes the CUSIGN detector for both positive and negative cases:

$$\begin{aligned} &S^{(k),+} = \max\left(0, S^{(k-1),+} + \operatorname{sgn}(r^{(k)})\right), \\ &S^{(k),+} = 0 \text{ and Alarm } \zeta^{(k),+} = 1, \qquad \textbf{if } S^{(k),+} = \tau \\ &S^{(k),-} = \min\left(0, S^{(k-1),-} + \operatorname{sgn}(r^{(k)})\right) \\ &S^{(k),-} = 0 \text{ and Alarm } \zeta^{(k),-} = 1, \qquad \textbf{if } S^{(k),-} = -\tau \end{aligned} \tag{22}$$

The working principle of CUSIGN test variable sequences is to accumulate the signed residual value $\operatorname{sgn}(r^{(k)}) \in \{-1, 0, 1\}$ and trigger an alarm $\zeta^{(k),+}, \zeta^{(k),-} \in \{0, 1\}$ when the test variables reach their corresponding threshold values $\tau \in \mathbb{N}_+$. As either of the test variables reach their respective thresholds, then the test variable is reset to zero. An example of the CUSIGN detection procedure (22) is shown in Fig. 3, where an incoming data sequence of residuals transition the positive and negative CUSIGN test variables $S^{(k),+}$ and $S^{(k),-}$. When either test variable reaches the threshold, for this example $\tau = 2$, an alarm is triggered (indicated by the red circles) and a reset-to-zero condition occurs. The CUSIGN detector monitors the occurrence of triggered alarms as the CUSIGN test variables reach their respective thresholds, where irregular occurrences indicated an attack may be happening.

Similar to the implementation in [13], the transition of the CUSIGN test variable sequences can be constructed as a Markov chain with a transition matrix modeled from the probabilities p_+ and p_- computed in (20). Consisting of a user-defined threshold τ to trigger an alarm, we show the transitions of $S^{(k),\pm}$ with a Markov chain diagram in Fig. 4.

Given a chosen threshold value $\tau \in \mathbb{N}_+$ as a value that triggers an alarm when $|S^{(k),\pm}| = \tau$, we describe the Markov chain in Fig. 4 in the form of a Markov transition matrix $\mathcal{T}^{\pm} \in \mathbb{R}^{(\tau+1)\times(\tau+1)}$, denoted for both the positive and negative transition matrices, $\mathcal{T}^+$ and $\mathcal{T}^-$. The CUSIGN Markov chain, occurring in a discrete manner, contains $\tau + 1$ states denoted as $\mathcal{M} = \{M_0, M_1, \ldots, M_\tau\}$, where M_τ is an absorbing state that is equal to the threshold, causing the CUSIGN test sequence $S^{(k),\pm}$ to reset to M_0. The CUSIGN Markov transition matrix for the positive $\mathcal{T}^+$ with a probability distribution of $\operatorname{sgn}(r^{(k)})$

Authorized licensed use limited to: University of Virginia Libraries. Downloaded on February 19,2022 at 19:19:55 UTC from IEEE Xplore. Restrictions apply.

is written as

$$\mathcal{T}^+ = \begin{bmatrix} p_- & p_+ & 0 & 0 & \dots & 0 \\ p_- & 0 & p_+ & 0 & \dots & 0 \\ 0 & p_- & 0 & p_+ & & 0 \\ \vdots & & \ddots & & \ddots & \vdots \\ 0 & \dots & 0 & p_- & 0 & p_+ \\ 0 & \dots & 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathcal{Q}^+ & * \\ \mathbf{0}_{1\times\tau} & 1 \end{bmatrix}. \tag{23}$$

The transition matrix $\mathcal{T}^+$ structure remains the same on any system, where the matrix size depends only on the value of the threshold τ. Transition probabilities for transient states in $\mathcal{T}^+$ adhere to the following:

$$\begin{cases} \Pr(M_j \to M_{j+1}) = p_+, & \text{for } j = \{0, \dots, \tau-1\} \\ \Pr(M_j \to M_{j-1}) = p_-, & \text{for } j = \{1, \dots, \tau-1\} \\ \Pr(M_0 \to M_0) = p_- \end{cases} \tag{24}$$

and the final row represents an absorbing (i.e., triggering) state containing elements equal to 0, besides the last element equaling 1.

We define $\mathcal{Q}^+ \in \mathbb{R}^{\tau\times\tau}$ as the fundamental matrix obtained from $\mathcal{T}^+$ with its last row and column removed (i.e., the absorbing state at threshold τ is removed), representing the transition probabilities to and from the transient states. Elements of $\mathcal{Q}^+$ are all nonnegative, and row sums are equal to or less than 1, while the eigenvalues satisfy $\rho[\mathcal{Q}^+] < 1$ such that $(\mathcal{Q}^+)^k \to 0$ as $k \to \infty$ and $\sum_{k=0}^{\infty}(\mathcal{Q}^+)^k = (\boldsymbol{I}_\tau - \mathcal{Q}^+)^{-1}$, where $\rho[\cdot]$ is the spectral radius and $\boldsymbol{I}_\tau$ is the identity matrix of size τ. Leveraging the fundamental matrix $\mathcal{Q}^+$, we can compute an expected alarm rate as indicated in the following lemma.

Lemma 1: Given a system with a CUSIGN detector (22) with a user-defined threshold $\tau \in \mathbb{N}_+$ that is not affected by cyber attacks such that the residual sequence satisfies $r^{(k)} \sim \mathcal{N}(0, \sigma^{(k)})$, then the inverse of the first element of the following vector:

$$\boldsymbol{\mu}^+ = (\boldsymbol{I}_\tau - \mathcal{Q}^+)^{-1}\mathbf{1}_{\tau\times 1} = (\mu_1^+, \dots, \mu_\tau^+)^\mathsf{T} \tag{25}$$

is the expected alarm rate, i.e., $\mathbb{E}[A^+] = (\mu_1^+)^{-1}$.

Proof: Given the Markov chain containing $\tau + 1$ states denoted by $\mathcal{M} = \{M_0, M_1, \dots, M_\tau\}$, a fundamental matrix $\mathcal{Q}^+$ is taken from a designed Markov transition matrix (23) to satisfy the transition probabilities (24). Leveraging the theory of average run length (ARL) introduced in [50], the ARL is defined as the average length of time for the test sequence to reach the threshold τ to trigger an alarm, determined by the fundamental matrix $\mathcal{Q}^+$ containing the transient states within $\mathcal{T}^+$. By definition, the inverse of the ARL to observe an alarm results in the average frequency of obtaining an alarm, known as the alarm rate. The ARL can be found by computing (25); then, by inverting the first element of $\boldsymbol{\mu}^+$, i.e., $(\mu_1^+)^{-1}$, we finally obtain the expected alarm rate $\mathbb{E}[A^+] = (\mu_1^+)^{-1}$.

Remark 1: The design of transition matrix $\mathcal{T}^-$ with subsequent fundamental matrix $\mathcal{Q}^-$ and expected alarm rate $\mathbb{E}[A^-] = (\mu_1^-)^{-1}$ for the negative case is computed by (23)–(25) with transition probability (p_+ and p_-) signs inverted.

TABLE I
EMPIRICAL VALUES FOR THE SCALING VALUE θ GIVEN $\tau = 1, 2, 3, 4$

Threshold τ	$\tau = 1$	$\tau = 2$	$\tau = 3$	$\tau = 4$
θ	1	0.74	0.7	0.69

The expected variance of estimated alarm rates $A^{(k),\pm}$ using MRE for runtime estimation have been found through empirical results in [37]. A scaling factor $\theta \in \mathbb{R}_+$ is found to be dependent on the chosen threshold τ. The observed MRE scaling factor approximates of θ are presented in Table I for thresholds $\tau = 1, 2, 3, 4$ and $\ell \geq 10$ (see [37]).

Proposition 1: Assuming that a residual is not affected by a cyber attack while using (21) for alarm rate estimation, the alarm rate is normally distributed by the following:

$$\hat{A}^{(k),\pm} \sim \mathcal{N}\left(\mathbb{E}[A^\pm], \frac{\theta\mathbb{E}[A^\pm](1-\mathbb{E}[A^\pm])}{2\ell - 1}\right). \tag{26}$$

By leveraging the expected distribution of the estimated alarm rate in (26), bounds of the alarm rate can be made. The following corollary provides alarm rate detection bounds for the CUSIGN detector.

Corollary 1: Given a residual $r^{(k)}$ monitored by the CUSIGN detector (22) consisting of a threshold $\tau \in \mathbb{N}_+$, detection of cyber attacks occurring for a given level of significance $\alpha \in (0, 1)$ when $\Omega_- \leq \hat{A}^{(k),\pm} \leq \Omega_+$ is no longer satisfied.

Proof: With the CUSIGN detector consisting of a threshold τ, an expected alarm rate $\mathbb{E}[A^\pm]$ found in (25), and leveraging (21) with a pseudo-window of length ℓ, the distribution of the estimated alarm rate follows the normally distributed properties from (26). Detection bounds $\Omega_\pm = [\Omega_-, \Omega_+]$ of a user-defined level of significance $\alpha \in (0, 1)$ (i.e., the probability that a false detection occurs in nominal conditions) follow:

$$\begin{aligned} \mathbb{E}[A^\pm] - \left|\Phi^{-1}\left(\frac{\alpha}{2}\right)\right|\sqrt{\frac{\theta\mathbb{E}[A^\pm](1-\mathbb{E}[A^\pm])}{2\ell-1}} &\leq \hat{A}^{(k),\pm} \\ \leq \mathbb{E}[A^\pm] + \left|\Phi^{-1}\left(\frac{\alpha}{2}\right)\right|\sqrt{\frac{\theta\mathbb{E}[A^\pm](1-\mathbb{E}[A^\pm])}{2\ell-1}} \end{aligned} \tag{27}$$

where $\Phi^{-1}(\cdot)$ is the *inverse cumulative distribution function* of a standard normal distribution [48], thus satisfying Corollary 1 and concluding the proof.

In summary, with the CUSIGN detection procedure, we can monitor and detect nonrandom behavior in residual data. Under a worst-case scenario (i.e., assuming an attacker has full knowledge of the system model and detection procedure), an intelligent attacker could remain hidden by triggering alarms at rates that do not travel beyond detection bounds while maintaining an attack vector. However, the CUSIGN detector's attack deterring effects will be limited, and one could implement multiple detectors in parallel with different threshold values τ to further impair an attacker's ability to remain hidden. For a more detailed discussion about undetectable attacks, the reader can check Appendix A.

In the next section, we expand these thoughts and show how to deploy the proposed technique on the multirobot problem under the attacks presented in Section II-A.

Authorized licensed use limited to: University of Virginia Libraries. Downloaded on February 19,2022 at 19:19:55 UTC from IEEE Xplore. Restrictions apply.

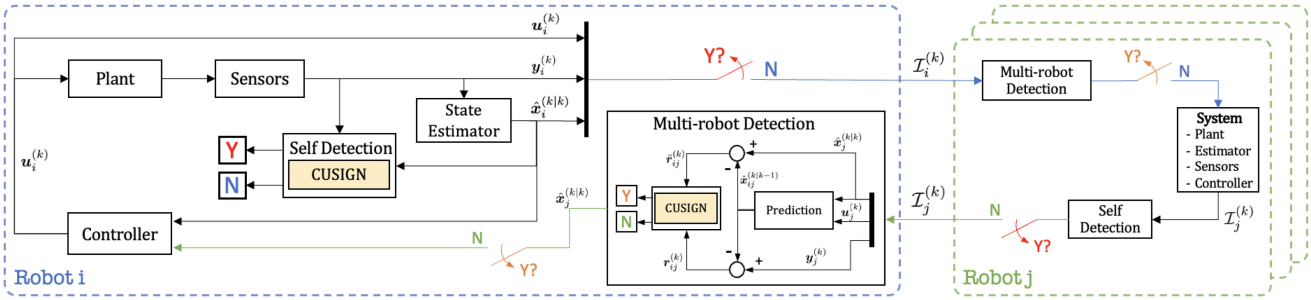


Fig. 5. Overall architecture describing our resilient robotic framework executed by each robot $i \in \mathcal{V}$ in the network. Both the multirobot detection on its neighbors $j \in \mathcal{S}_i$ and self-detection on itself are performed to find stealthy attacks that exhibit nonrandom residual behavior.

IV. Multirobot System Attack Detection and Recovery

In this section, we show how to deploy the proposed CUSIGN technique on a multirobot system to detect nonrandom (i.e., inconsistent) residual behavior due to cyber attacks and to recover/reconfigure the system. Our scheme is leveraged to monitor and detect if neighboring robots are compromised, as well as to perform self-monitoring for discovering inconsistencies to on-board sensor measurements due to cyber attacks. Fig. 5 summarizes the high-level procedure in a block diagram that is executed by each robot in the network to monitor for cyber attacks and locally recover the system when attacks have been detected. As a running case study for the remainder of this article, we employ a VSDM with the Gabriel graph (GG) rule for proximity-based formation control [51] to demonstrate our detection and recovery approach on multirobot system formations. However, our approach is valid for any cooperative proximity-based formation control for multirobot systems (e.g., nearest neighbor [46]).

A. Virtual Spring–Damper Mesh

In this article, we consider VSDMs alongside the use of the GG rule [52], [53] for proximity-based control of multirobot systems performing coordinated operations. Given that all agents are cooperative, this method allows for a decentralized algorithm, where agents are required to leverage local (proximity-based) interactions that result in a desired global behavior of the system. Furthermore, systems that leverage VSDMs incorporate favorable characteristics that include scalability, efficiency, and known stability properties such that all decentralized agents converge to a global consensus [41], [54]. Numerous works have leveraged VSDMs for various applications [42]–[45], [51], [55], [56], where all of these works did not consider security issues. A single compromised agent affected by cyber attacks can hijack the entire multirobot system to an undesirable state due to the control interconnections that are propagated throughout the robot network, as demonstrated in Fig. 1. In a decentralized manner, our work allows the agents to identify and remove nearby agents from the network (i.e., the control graph) that could potentially cause undesired behavior.

The objective of the multirobot system is to navigate to a goal location while maintaining a desired distance between neighboring robots. Furthermore, the multirobot system must be resilient to stealthy communication and sensor cyber attacks. In addition, we make the assumption that the robot network is navigating in an unknown cluttered environment: to this end, all robots are fitted with a range sensor providing 360° field of view (FOV) (e.g., a lidar) with limited range $\delta_r < \delta_c$ for obstacle/collision avoidance.

Each robot i is controlled to follow the VSDM network dynamics by leveraging a virtual spring–damper physics model

$$\boldsymbol{u}_i = \ddot{\boldsymbol{p}}_i = \Bigg[\sum_{j \in \mathcal{S}_i} \kappa_{ij}(l_{ij} - l_r^0)\vec{\boldsymbol{d}}_{ij} + \sum_{o \in \mathcal{O}_i} \kappa_{io}(l_{io} - l_o^0)\vec{\boldsymbol{d}}_{io} + \kappa_{ig} l_{ig} \vec{\boldsymbol{d}}_{ig} \Bigg] - \gamma_i \dot{\boldsymbol{p}}_i \in \mathbb{R}^m \tag{28}$$

where $\mathcal{S}_i$ is the neighbor set of robot $i \in \mathcal{V}$ in the control graph $\mathcal{G}_{\mathcal{U}}$ and $\mathcal{O}_i$ is the set of obstacles within the FOV of robot i. Given the use of a virtual spring model, l denotes the spring length between a robot i and neighboring robots j (l_{ij}), obstacles (l_{io}), and the goal g (l_{ig}), while l^0 are the desired virtual spring rest lengths, κ represents the spring constants, and $\vec{\boldsymbol{d}}$ is a unit vector. Given damping coefficients that satisfy $\gamma_i > 0$, the multirobot system leveraging the VDSM emulates a true spring–damper mesh, where dissipating forces act against the velocities, leading to an equilibrium state of zero velocity in the absence of other external forces.

Given the set of robots $\mathcal{V}$, each robot $i \in \mathcal{V}$ computes its neighbor set for control $\mathcal{S}_i \subset \mathcal{C}_i$ from the received information of nearby robots (4) in the communication graph $\mathcal{G}_{\mathcal{C}}$ by following the GG rule [52], [53]. A GG is constructed in the following way: a robot j belongs to the neighbor set $\mathcal{S}_i$ of a robot i (i.e., a directed control edge is formed between i and j) if and only if there are no other robots $h \in \mathcal{C}_i$ within the circle of diameter $\overline{ij}$ [57]:

$$\mathcal{S}_i = \left\{ j \in \mathcal{C}_i \setminus \mathcal{V}_i^C \mid \widehat{ihj} \leq \pi/2 \right\} \tag{29}$$

where $\widehat{ihj}$, $i \neq j \neq h$, is the interior angle of the three robot positions' configuration from the on-board position estimate $\hat{\boldsymbol{p}}_i^{(k)}$ of robot i and position estimates $\hat{\boldsymbol{p}}_j^{(k)}$ and $\hat{\boldsymbol{p}}_h^{(k)}$ from received information of nearby robots j and h. The set $\mathcal{V}_i^C \subset \mathcal{V}$ in (29) denotes robots that are deemed compromised by robot i, such that control edges are not constructed to remove compromised

Authorized licensed use limited to: University of Virginia Libraries. Downloaded on February 19,2022 at 19:19:55 UTC from IEEE Xplore. Restrictions apply.

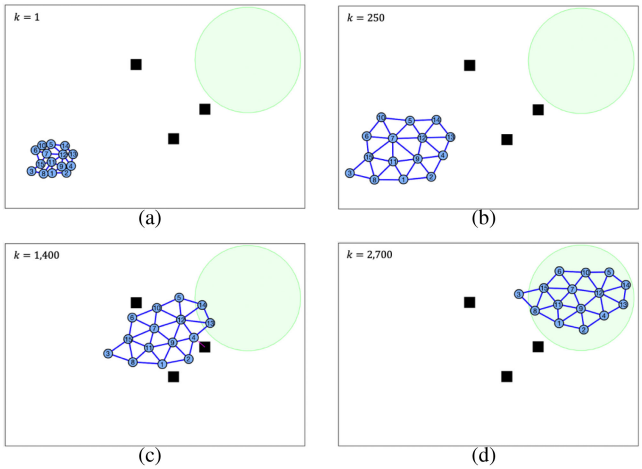


Fig. 6. (a)–(d) Sequence of snapshots with a robotic swarm consisting of $N = 15$ robots navigating toward a goal region (in green) using the VSDM network model (28) in the absence of cyber attacks.

robots from the network for resilient control. The determination of the compromised set $\mathcal{V}_i^C$ is discussed in Section IV-B. As a side note, the utilization of the GG rule allows for connected graphs with no crossing edges and, hence, an increased and uniform coverage as opposed to other graph techniques [58].

In Fig. 6, we show a sequence of snapshots for a simulation of a swarm of 15 robots deployed using the virtual spring model with GG in (28) and (29) to navigate toward a desired goal region while avoiding any obstacles in the environment.

B. Attack Detection and Reconfiguration

Robots within the multirobot system monitor for inconsistent behavior of their neighboring robots to avoid stealthy attacks from hijacking uncompromised robots and, potentially, the entire robot network. Each robot $i \in \mathcal{V}$ leverages received information $\mathcal{I}_j^{(k)}$ from any neighboring robot $j \in \mathcal{C}_i$ to perform attack detection by monitoring elements within the inter-robot measurement (17) and inter-robot state (15) residual vectors, as characterized in Section III-A.

To indicate that a robot $i \in \mathcal{V}$ is monitoring an sth inter-robot measurement residual element and the qth inter-robot state residual element on a robot $j \in \mathcal{V}$, we denote the alarm rates as $\hat{A}_{ij,s}^{(k),\pm} = \{\hat{A}_{ij,s}^{(k),+}, \hat{A}_{ij,s}^{(k),-}\}$ and $\hat{A}_{ij,q}^{(k),\pm} = \{\hat{A}_{ij,q}^{(k),+}, \hat{A}_{ij,q}^{(k),-}\}$, respectively. If an alarm rate no longer satisfies detection bounds in Corollary 1 (i.e., suggesting inconsistent behavior), a robot i deems the monitored robot j compromised. Once inconsistent behavior is detected, the robot i then isolates and removes the compromised robot j by placing it in its compromised set $\mathcal{V}_i^C \subset \mathcal{V}$. By placing robot j in its compromised set, robot i performs a local reconfiguration of the network topology using the GG rule on the communication graph presented in (29), hence forming a new control neighbor set $\mathcal{S}_i' = \mathcal{S}_i \setminus \{j\}$. A previously found compromised robot j is allowed re-entry into the robot network, and the control graph, in the event that the attack disappears and j behaves as expected again (i.e., the residuals follows the expected distribution). In this case, a local reconfiguration is again invoked using GG to compute $\mathcal{S}_i$.

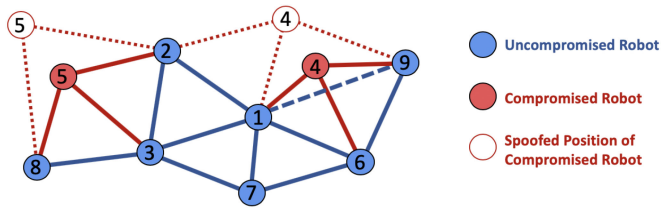


Fig. 7. Example of a network reconfiguration where uncompromised robots isolate and remove compromised robots that are sending spoofed position broadcasts during a communication attack.

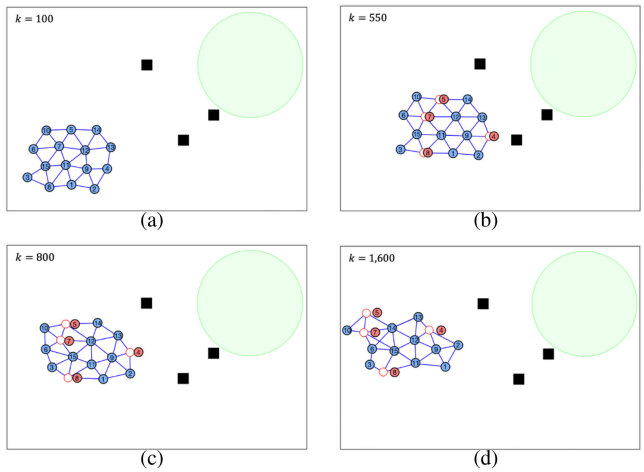


Fig. 8. (a)–(d) Swarm with four robots (red) that are experiencing malicious sensor attacks, causing their state estimates (white disks) to diverge from their true state. In turn, the network is dragged away from its intended goal (green).

Fig. 7 shows a pictorial example of the scheme in which compromised robots 4 and 5 are broadcasting spoofed position information [i.e., communication attack (A1)] to the robot network: the empty disks represent the spoofed broadcast position coordinates of the true positions of the compromised robots (red disks). The uncompromised robots (blue disks) detect nonrandom (i.e., inconsistent) behavior occurring from received information of robots 4 and 5, resulting in removal of any control edge connections that could affect the multirobot system performance, where $(i, j) \notin \mathcal{E}_{\mathcal{U}}$, $i \in \mathcal{V}$, and $j = \{4, 5\}$. After removing the malicious nodes, the remaining seven nodes reconfigure using the formation rules presented in Section IV-A.

Fig. 8 shows, as an example, the effect of a stealthy on-board sensor attack (A2 on an unprotected swarm with the same task in Fig. 6. The attack begins at time step $k = 400$ on four robots, dragging the entire multirobot system away from the desired goal. In this example, the empty red disks represent the unreliable on-board state estimates of the compromised robots that are used for on-board control and are also broadcast to nearby agents in the robot network, whereas the red disks denote the true positions of the compromised robots. The unreliable states that are broadcast to nearby robot are then leveraged by the uncompromised robots (denoted by blue disks), which

Authorized licensed use limited to: University of Virginia Libraries. Downloaded on February 19,2022 at 19:19:55 UTC from IEEE Xplore. Restrictions apply.

propagates the attack effects throughout the entire robot network affecting the overall mission.

C. Self-Detection

Similarly, each robot $i \in \mathcal{V}$ performs self-monitoring by leveraging the on-board measurement residual to search for stealthy on-board attacks on its sensors. As shown in Fig. 5, the CUSIGN detector is placed in the feedback of the traditional control loop to monitor the on-board measurement residual for potential attacks. As a sensor's measurements no longer satisfy an expected random behavior (i.e., alarm rates travel outside detection bounds), a robot i places itself into its compromised set $i \in \mathcal{V}_i^C \subset \mathcal{V}$.

In this article, the self-detected compromised robot isolates itself from the rest of the network by cutting any communication broadcasts to the network (i.e., $(i, j) \notin \mathcal{E}_{\mathcal{U}},\ \forall j \in \mathcal{V}$) and also stops moving toward the goal; formally, it will remove the first and third terms from (28), leaving any control effort only toward obstacle and other robot avoidance. While we decided to implement such a law for ease, different behaviors can be considered as we will discuss in more detail in Section VI.

D. System Stability

The multirobot system that leverages a VSDM with the GG rule for formation control, together with the attack detection scheme presented in Section III, creates a switching hybrid system, in which edges construct and deconstruct as the robots move through the environment. Past works have proved the static (i.e., fixed topology) and dynamic (i.e., switching topology) stability of this time varying switching system by using Lyapunov theory [41], [43].

Here, we extend some of these results and provide a stability proof also considering the cyber-security detection and isolation procedures described in the previous sections. As compromised robots are subject to cyber attacks that present detectable nonrandom behavior, certain directed edges (i.e., virtual springs used for control) from compromised robots are eliminated, while others between the remaining uncompromised robots may appear for network reconfiguration. Assuming that cyber attacks are detected using the proposed CUSIGN method, the multirobot system is guaranteed to reconverge to a new equilibrium after network reconfiguration occurs due to compromised robots being removed from the system, as formally described in the following theorem.

Theorem 1: The hybrid system in (28) with switching dynamics imposed by the GG rule (29), the CUSIGN (22) detector, and the network reconfiguration scheme, as discussed in Sections IV-B and IV-C, is stable (i.e., an equilibrium rest state can be reached).

Proof: See Appendix B.

V. Results

The proposed framework was validated with extensive MATLAB simulations as well as various ROS experiments on a swarm of TurtleBot UGVs and also Clearpath Jackal UGVs in Gazebo to cover different attack scenarios illustrated in Section II-A. Next, we present a few representative cases. While in this article, we showcase a few representative examples, extensive simulations and experiments under more attacks can be found in the provided supplemental material.[1]

Our proposed CUSIGN detector is compared with a BD detector [12] and a model-based CUSUM detector [13], whose detection procedures leverage alarm triggering based on thresholding magnitude values of the residual (see [13] for details on how to determine model-based BD and CUSUM thresholds). In comparison to BD and CUSUM detectors, our nonrandomness detector is nonparametric and only considers the signed value of the residual, while the residual magnitude is ignored. As we will see, certain attacks are undetectable by the BD and CUSUM detectors; however, a more resilient approach would be to deploy these magnitude-based detectors alongside the CUSIGN detector. Throughout the simulations and experiments in this section, all detectors use a level of significance $\alpha = 0.0004$ (i.e., $\sim 3.3\sigma$) for detection bounds, which are represented as dashed red lines in the figures displaying results for detector alarm rates. Additionally, BD and CUSUM detectors are tuned for a user-defined expected alarm rate A^{des} and their alarm rate detection bounds are chosen by assuming a normal approximation of the binomial alarm rate value (i.e., $\{0, 1\}$) using MRE (21) for alarm rate estimation.

A. MATLAB Simulations

For the MATLAB simulations, we considered double-integrator point mass dynamics for $N = 15$ robots in the swarm represented with the virtual spring model in (28) with each robot $i \in \mathcal{V}$ having a state vector $\boldsymbol{x}_i = [p_i^x, p_i^y, v_i^x, v_i^y]^\intercal \in \mathbb{R}^n$ consisting of positions and velocities in the xy plane. Throughout all simulations, the set of point mass robots $\mathcal{V}$ shares a maximum communication range $\delta_c = 15$ m, maximum range sensing distance $\delta_r = 3$ m, virtual spring rest lengths $l_r^0 = 4$ m and $l_o^0 = 3$ m, damping constant $\gamma_i = 3$, and spring constants $\kappa_{ij} = 15$, $\kappa_{io} = 40$, and $\kappa_{ig} = 5$. A pseudo-window of length $\ell = 50$ for MRE alarm rate estimation (21) is used by all detectors. Additionally, the CUSIGN test variable threshold is chosen to be $\tau = 2$ such that the expected CUSIGN alarm rates are $\mathbb{E}[A^{\pm}] = \frac{1}{6}$. Each robot $i \in \mathcal{V}$ measures the x and y positions with a sampling time $t_s = 0.05$ s, with measurement and process noise covariances $\boldsymbol{R} = \text{diag}(0.05, 0.05)$ and $\boldsymbol{Q} = \text{diag}(1\text{e}{-3}, 1\text{e}{-3}, 1\text{e}{-5}, 1\text{e}{-5})$. During all simulations, the CUSIGN detector is compared to the BD and CUSUM detectors, which monitor both the measurement and state residuals of the position for all robots $i \in \mathcal{V}$, which have thresholds tuned for an alarm rate of $A^{\text{des}} = 0.15$. Additionally, the CUSUM detector uses a bias of $b_s = 1.1\bar{b}_s$ (see [13] for further details on tuning of the BD and CUSUM detectors).

Two case studies are presented next: 1) a man-in-the-middle communication attack and 2) a sensor spoofing attack. In both cases, we consider the multirobot operation presented previously

[1] [Online]. Available: https://www.bezzorobotics.com/tro21

Authorized licensed use limited to: University of Virginia Libraries. Downloaded on February 19,2022 at 19:19:55 UTC from IEEE Xplore. Restrictions apply.

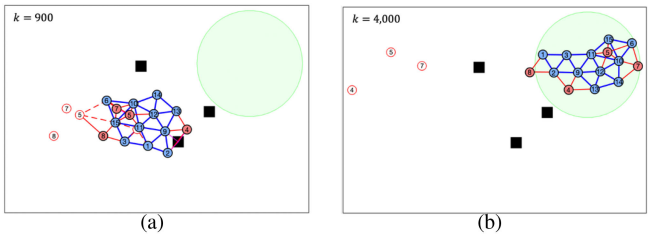


Fig. 9. (a) and (b) Robotic swarm navigating toward a goal point (in green) while protected from stealthy ramp attacks on communication broadcasts.

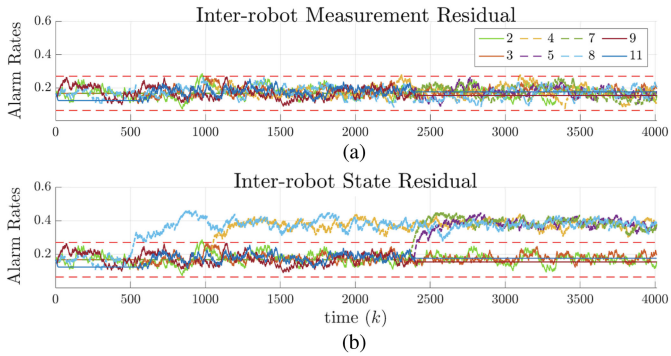


Fig. 10. Resulting alarm rates of robot 2 performing multirobot detection on any neighboring robots $j \in \mathcal{S}_2$ from the case depicted in Fig. 9 for inter-robot (a) measurement and (b) state residuals. The state residual is able to detect stealthy communication attacks that are not detected by the measurement residual from robots $j = \{4, 13\}$.

in Fig. 6 with robots $i = \{4, 5, 7, 8\}$ under attack from time instant $k = 500$.

Communication attack: Our first case study involves a stealthy man-in-the-middle communication attack (A1), as discussed in Section II-A, in which position measurement data from compromised robots are intercepted and replaced with incorrect data before broadcasting to the rest of the swarm while slowly ramping the position in the $(-x)$-direction. Fig. 9 shows the behavior of the swarm once we deploy our framework, in which the compromised robots are detected in this case through the CUSIGN (22) detector and isolated by their neighbors.

Fig. 10 shows the evolution of the alarm rate from the perspective of robot 2 monitoring robots $j = \{2, 3, 4, 5, 7, 8, 9, 11\}$ that belong to its neighbor set $\mathcal{S}_2$ at some point in time $k > 0$ during the stealthy communication attack case study presented in Fig. 9. For multirobot detection, robot 2 monitors both the inter-robot measurement and state residuals of its neighboring robots $j \in \mathcal{S}_2$. As shown in Fig. 10(a), the CUSIGN detector of robot 2 that monitors the measurement residual $\boldsymbol{r}_{2j}^{(k)}$ of its neighboring robots $j \in \mathcal{S}_2$ does not detect the attack, while in Fig. 10(b), the detectors that are monitoring the inter-robot state residual $\breve{\boldsymbol{r}}_{2j}^{(k)}$ find the inconsistent behavior as the attacker is pushing the state estimate slowly to one side.

Sensor attack: Our second case study involves stealthy on-board sensor measurement attacks (A2) described in (8), attempting to hijack compromised robots to an undesired state. Similar to our simulation case of a communication attack, an

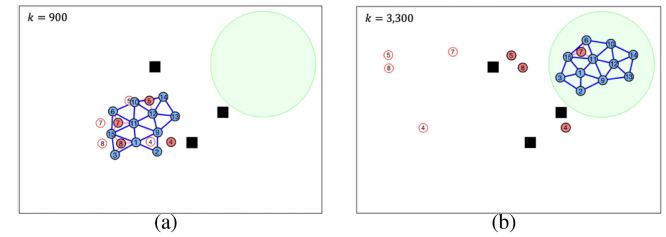


Fig. 11. (a) and (b) Robotic swarm navigating toward a goal point (in green) while protected from stealthy attacks to on-board sensor measurements.

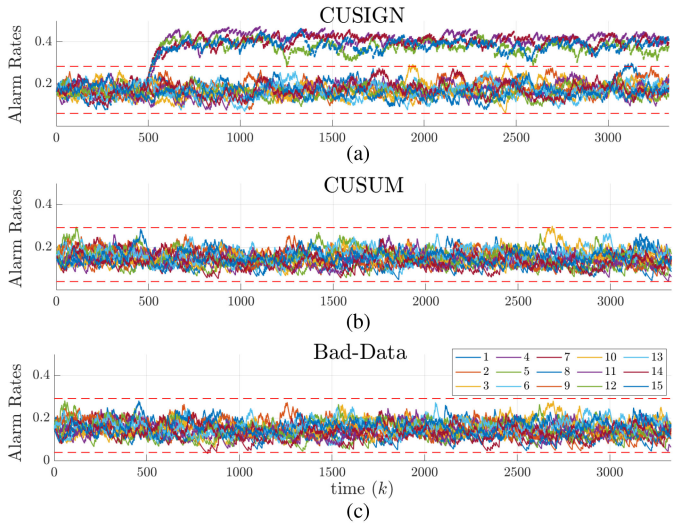


Fig. 12. Alarm rates comparison between (a) CUSIGN, (b) CUSUM, and (c) BD for the case study shown in Fig. 11 for self-detection while monitoring the on-board measurement residual for position in the x-direction as stealthy cyber attacks affect sensors on-board robots $i = \{4, 5, 7, 8\}$.

attack is slowly ramping the position measurement in the $(-x)$-direction, while remaining hidden from previously state-of-the-art detection schemes. Fig. 11 displays the detection results against stealthy sensor attacks, where uncompromised robots isolate and remove malicious robots from the network while maintaining the desired task of navigating to the goal point. The sensor spoof considered deliberately hides within the noise to evade detection from the CUSUM and BD detectors, as shown in Fig. 12(b) and (c), but the attacker leaves trails of nonrandom residual behavior, which is detected by the CUSIGN detector [see Fig. 12(a)].

B. TurtleBot Experiments

Experimental validations are performed on $N = 5$ TurtleBot2 differential-drive robots performing a go-to-goal operation within a laboratory environment. The hardware used is a Lenovo P51 Workstation equipped with an Intel Core i7-6820HQ processor at 2.7-GHz running Linux with ROS enabled. The controller for each robot and the attacks are implemented in MATLAB interfaced with ROS through the Robotic Systems Toolbox, and the operation is executed at 100 Hz. In this experiment case study, the network of UGVs is tasked to navigate

Authorized licensed use limited to: University of Virginia Libraries. Downloaded on February 19,2022 at 19:19:55 UTC from IEEE Xplore. Restrictions apply.

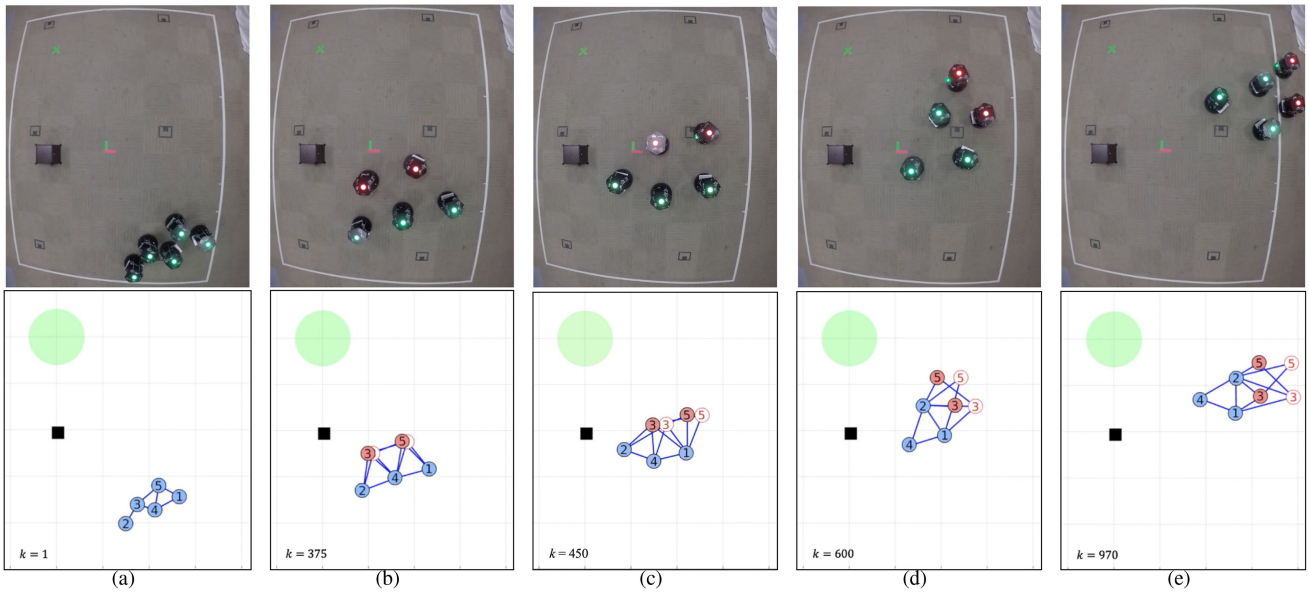


Fig. 13. (a)–(e) Robotic swarm attempting to navigate toward a goal region (in green) while unprotected from stealthy attacks on communication broadcasts.

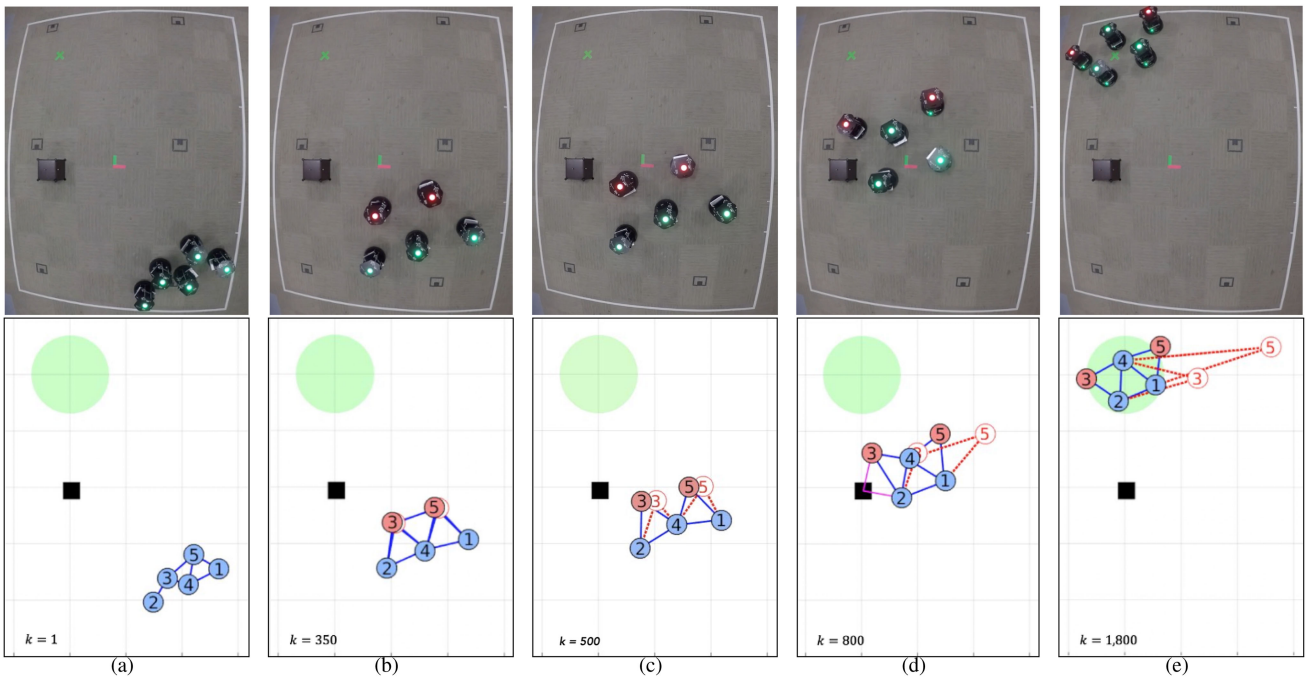


Fig. 14. (a)–(e) Robotic swarm attempting to navigate toward a goal region (in green) while protected from stealthy attacks on communication broadcasts. False broadcast information of the robot positions are discovered, and the swarm is able to isolate and remove any robots with spoofed communication broadcasts.

to a goal region (in green) while resiliently maintaining a desired network topology that satisfy edges by the GG rule (29).

Two different cases are implemented: 1) communication attack without detection and 2) communication attack with detection, with robots $i = \{3, 5\} \in \mathcal{V}$ subject to attacks. For both cases, we use the following system parameters: $\delta_c = 3$ m, $\delta_r = 0.6$ m, $l_r^0 = 0.7$ m, $l_o^0 = 0.5$ m, and $\gamma_i = 0.5$. Measurement noise covariance follows $\boldsymbol{R} = \text{diag}(0.01, 0.01, 0.002, 0.0004)$ on positions, velocity, and heading angle states, while a pseudo-window length $\ell = 40$ for MRE alarm rate estimation (21) is used for all detectors. We begin with the case where no detection occurs in Fig. 13, showing how a stealthy communication attack is able to drive the network of UGVs to an undesirable state, away from the intended goal region. Fig. 14 shows the case where we have the CUSIGN detector monitoring the inter-robot state residual from information received from neighboring robots. The communication attacks on robots $\{3, 5\}$ are discovered by the remaining uncompromised robots,

Authorized licensed use limited to: University of Virginia Libraries. Downloaded on February 19,2022 at 19:19:55 UTC from IEEE Xplore. Restrictions apply.

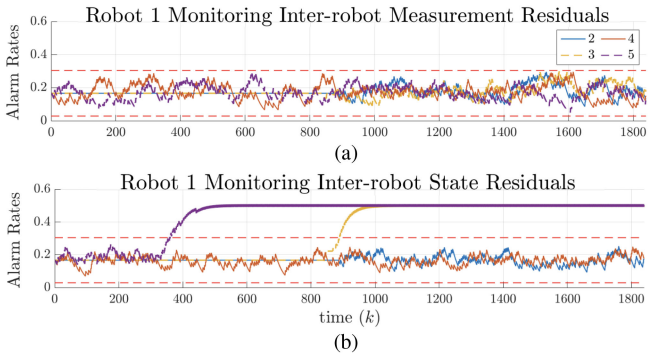


Fig. 15. Resulting alarm rates from the perspective of robot 1 for the experiment in Fig. 14 while monitoring the inter-robot residuals. (a) Robot 1 monitoring inter-robot measurement residuals. (b) Robot 1 monitoring inter-robot state residuals.

resulting in a network reconfiguration to remove the attacked robots. Fig. 15 displays the detector results from the perspective of robot $i = 1$ from Fig. 14, where in Fig. 15(a), the stealthy communication attack is not detectable on the inter-robot measurement residual, but in Fig. 15(b), it leaves traces of nonrandom behavior in the state residual $\breve{r}^{(k)}_{1j,q}$ for the position in the x-direction $\forall j \in \mathcal{V} \setminus \{1\}$.

C. Gazebo Experiments

To further reinforce these results, a case study on sensor spoofing was demonstrated with an experiment in Gazebo with $N = 10$ Clearpath Jackal Robots performing a go-to-goal operation in a larger environment with more obstacles, as demonstrated in Figs. 16 and 17. We leverage Gazebo because it allows us to run longer experiments with more robots, larger spaces, and considering even stealthier attacks than experiments in our laboratory space. In addition, in this case study, we decided to use the Jackal robots to show the flexibility of our framework to deal with different dynamical models.

In the case of sensor attacks, the objective of an attacker is to slowly push a sensor measurement (e.g., positions) to one side, resulting in hijacking of the true state of the robot that diverges from the on-board state estimate. With this in mind, a larger environment is needed to perform a truly and effective stealthy attack. The robots share a maximum communication and sensing range of $\delta_c = 15$ m and $\delta_r = 3$ m, with virtual spring rest lengths $l^0_r = 4$ m and $l^0_o = 3$ m. Sensor measurement noise covariance follows $\boldsymbol{R} = \text{diag}(0.05, 0.05, 0.002, 0.0004)$ on the N_s sensors receiving measurements of the robot position, velocity, and heading angle. A pseudo-window of length $\ell = 40$ for MRE alarm rate estimation (21) is used for all detectors. Additionally, the BD detector is tuned for an expected alarm rate $A^{\text{des}} = 0.15$, while CUSUM is tuned for $A^{\text{des}} = 0.1$ (with bias $b_s = 1.05\bar{b}_s$).

In Fig. 16, we show the sequence of snapshots for the robot network while experiencing stealthy on-board sensor attacks on robots $\{7, 8, 10\}$ beginning at $k = 200$ and robots $\{4, 6\}$ beginning at $k = 400$. During the attack, compromised robots have their position measurements slowly ramped away in the $(-x)$-direction with the intention of driving the swarm away from the desired goal. Avoidance actions are required from nearby robots that leverage their on-board range sensors to prevent collisions. A comparison between detectors—CUSIGN, BD, and CUSUM—during the stealthy sensor spoof from Fig. 16 is shown in Fig. 18, with their on-board alarm rates displayed over the entire length of the case study. The CUSUM and BD detectors on-board the robots fail to detect the stealthy sensor attacks, while the CUSIGN detector is able to identify that the compromised robots are presenting inconsistent information, which allows the compromised robots to safely remove themselves from the formation.

VI. Conclusion

This article presented a resilient approach to detect and defend against stealthy sensor and communication attacks that cause nonrandom behavior within homogeneous multirobot systems. The CUSIGN detector was introduced to counteract these stealthy attacks by monitoring alarm rates triggered by residual changes over time. Upon detection, the multirobot system reconfigures to isolate the malicious robots in a decentralized fashion. The proposed scheme is scalable since each robot only relies on the local information received from its neighbors to assess security issues. Finally, in our extensive simulations and experiments, we showed how our framework can outperform well-known residual-based detection schemes such as BD and CUSUM detectors. Assembling together these magnitude-based detection schemes with our proposed approach would increase the overall resilience of the system.

In the simulation and experiment demonstrations, we considered double integrator, differential drive, and skid-steering dynamics to show the generality and flexibility of our framework. The main assumption for our framework is to have *a priori* knowledge about the vehicle dynamics and the noise models. Currently, we have assumed that communication within the network is ideal, such that synchronization errors, time delays, and communication failures are negligible. Future efforts expanding on this work could include and leverage more accurate communication models with uncertainties as introduced in [59] and [60] to further increase resilience, for example, by using the dependencies between communication quality and distance between two communicating agents (i.e., as a side-channel detection scheme). Expanding the proposed work to heterogeneous robotic systems with different classes of vehicles and sensing capabilities is also another aspect that could be investigated in the future.

From a recovery/reconfiguration perspective, we believe that an important direction forward would be on how to deal with the robots that are found compromised. In this article, compromised robots were isolated and removed from the network, to avoid their malicious effect on the coordination of the rest of the uncompromised robots. However, more complicated approaches can be considered to stop the malicious robots, such as surrounding or dragging them toward a safe state. In order to enable such behaviors, it is necessary to predict the state of the compromised vehicles. One possibility here is to research checkpointing and

Authorized licensed use limited to: University of Virginia Libraries. Downloaded on February 19,2022 at 19:19:55 UTC from IEEE Xplore. Restrictions apply.

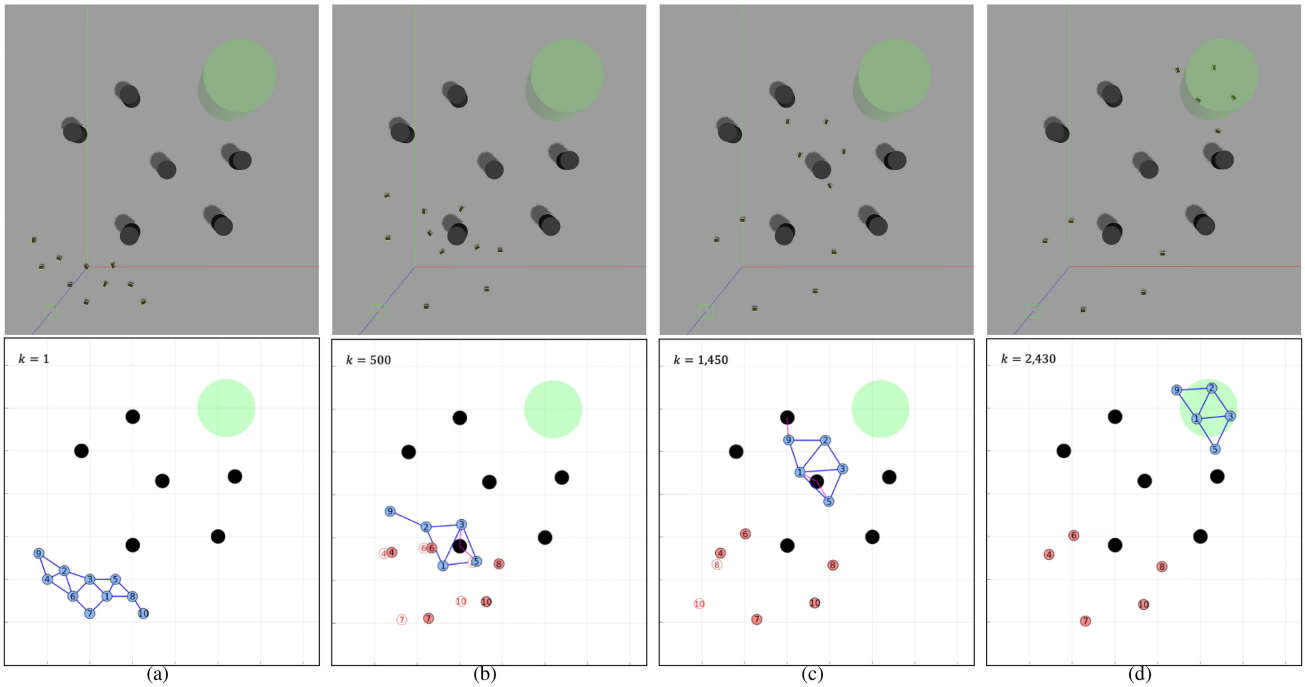


Fig. 16. (a)–(d) Robotic swarm attempting to navigate toward a goal region (in green) while protected from stealthy attacks on sensor measurements. False data injections to the robot position measurements are discovered, and the swarm is able to resiliently isolate and remove any robots under attack to reach the goal.

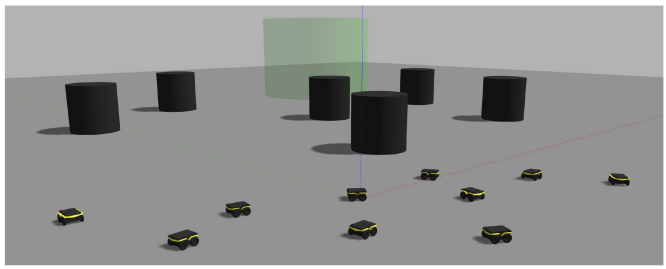


Fig. 17. Initial positions of the $N = 10$ Clearpath Jackal UGVs within a cluttered environment for the experiments using Gazebo.

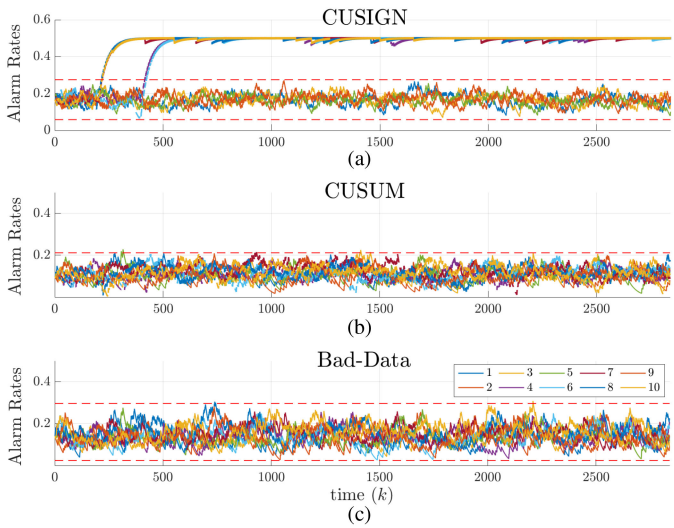


Fig. 18. Alarm rate results from the experiment in Fig. 16 for robots $i \in \mathcal{V}$ performing self-detection while monitoring the on-board measurement residuals as stealthy cyber attacks affect sensors on-board robots $i = \{4, 6, 7, 8, 10\}$. The CUSIGN detector detects nonrandom behavior of the sth measurement residual (affecting the x position) as alarm rates travel outside of detection bounds. The CUSUM and BD detectors do not recognize the stealthy attacks. (a) CUSIGN. (b) CUSUM. (c) BD.

recovery methods, inspired by traditional software engineering, by rolling back to reliable states of the compromised system to predict forward its possible states after it was compromised.

Predicting the intention of an attacker is also in our agenda since this will further increase resilience to better recover a system. The inclusion of learning-enabled components such as regression and classification techniques could further improve the on-board computation for detection. Furthermore, we plan to investigate the effects of worst-case attack sequences that an attacker can perform while evading detection from the CUSIGN detector to characterize the maximum damage in terms of the resulting true state divergence from the on-board state estimate.

Appendix A
Characterization of Undetectable Attacks

In this appendix, we discuss attack sequences that an attacker can take to remain hidden from detection from the CUSIGN detection scheme for both sensor and communication attacks. In order to evade detection from CUSIGN, an attacker must be mindful of both the positive and negative test variables $S^{(k),+}$ and $S^{(k),-}$ with their respective alarm rates $\hat{A}^{(k),+}$ and $\hat{A}^{(k),-}$. To maximize damage in a hijacking attack, a smart attacker would want to manipulate a variable of choice (e.g., sensor

Authorized licensed use limited to: University of Virginia Libraries. Downloaded on February 19,2022 at 19:19:55 UTC from IEEE Xplore. Restrictions apply.

measurement) to push the system in a specific direction with maximum effect, without passing alarm rate detection bounds $\Omega_{\pm}$ defined in (27). As a result of maximizing the effects of an attack, one alarm rate is driven toward the maximum alarm rate threshold Ω_{+}, and the other alarm rate is pushed toward the minimum threshold Ω_{-}.

Assumption 1: Under a worst-case scenario, an attacker has knowledge of the robot dynamical model (1), the network model (e.g., proximity-based consensus protocol), and the state estimation procedure (e.g., Kalman filter). Furthermore, a malicious attacker has the ability to manipulate any sth on-board sensor measurement $y_{i,s}^{(k)}$ and/or any information within $\mathcal{I}_i^{(k)}$ on a robot $i \in \mathcal{V}$ through communication broadcasts.

On-board sensor attack: The first case considered is in the event that an attacker can inject false data to sensor measurements on-board a robot i, where $\boldsymbol{\xi}_{i,y}^{(k)} \neq 0$. Utilizing the spoofed output vector in (8) combined with the on-board measurement residual defined in (12), we can rewrite the on-board measurement residual vector on an ith robot as

$$\begin{aligned}\boldsymbol{r}_i^{(k)} &= \widetilde{\boldsymbol{y}}_i^{(k)} - \boldsymbol{C}\hat{\boldsymbol{x}}_i^{(k|k-1)} \\ &= \boldsymbol{C}\boldsymbol{x}_i^{(k)} + \boldsymbol{\eta}_i^{(k)} + \boldsymbol{\xi}_{i,y}^{(k)} - \boldsymbol{C}\hat{\boldsymbol{x}}_i^{(k|k-1)} \\ &= \boldsymbol{C}\boldsymbol{e}_i^{(k|k-1)} + \boldsymbol{\eta}_i^{(k)} + \boldsymbol{\xi}_{i,y}^{(k)} \end{aligned} \tag{30}$$

where $\boldsymbol{e}_i^{(k|k-1)} = \boldsymbol{x}_i^{(k)} - \hat{\boldsymbol{x}}_i^{(k|k-1)} \in \mathbb{R}^n$ is the state prediction error. Each sth on-board measurement residual element, $s \in \{1, \ldots, N_s\}$, is defined as

$$r_{i,s}^{(k)} = \boldsymbol{C}_s \boldsymbol{e}_i^{(k|k-1)} + \eta_{i,s}^{(k)} + \xi_{i,y,(s)}^{(k)} \in \mathbb{R} \tag{31}$$

where $\boldsymbol{C}_s$ is the sth row of the output matrix $\boldsymbol{C}$ and $\xi_{i,y,(s)}^{(k)} \in \mathbb{R}$ is the sth element of the sensor measurement attack vector. An intelligent attacker can manipulate the measurement residual sign by constructing a suitable attack signal to create an attack sequence that avoids the CUSIGN detection bounds. An attacker can manipulate the residual sign by choosing an attack vector element s of the sensor measurement to satisfy

$$\operatorname{sgn}\left(r_{i,s}^{(k)}\right) = \begin{cases} 1, \text{ if } \xi_{i,y,(s)}^{(k)} > -\boldsymbol{C}_s \boldsymbol{e}_i^{(k|k-1)} - \eta_{i,s}^{(k)} \\ -1, \text{ if } \xi_{i,y,(s)}^{(k)} < -\boldsymbol{C}_s \boldsymbol{e}_i^{(k|k-1)} - \eta_{i,s}^{(k)} \end{cases}. \tag{32}$$

We first examine the scenario when either of the estimated alarm rates monitoring the positive $\hat{A}_{i,s}^{(k),+}$ or negative $\hat{A}_{i,s}^{(k),-}$ residual sign occurrences approach the maximum detection boundary threshold Ω_{+} on a robot i. The objective of the attacker is to drive the alarm rate of the desired sign as close to the maximum threshold without crossing it. The following equation is a restriction on the attack signal $\xi_{i,y,(s)}^{(k)}$ for a sensor s on-board a robot i, for both alarm rates denoted as $\hat{A}_{i,s}^{(k),\pm}$, such that neither cross the maximum threshold:

$$\xi_{i,y,(s)}^{(k)} = \left\{ \xi_{i,y,(s)}^{(k)} \underset{-}{\overset{+}{\lessgtr}} -\boldsymbol{C}_s \boldsymbol{e}_i^{(k)} - \eta_{i,s}^{(k)} \;\middle|\; \left(\Omega_{+} - \hat{A}_{i,s}^{(k-1),\pm} - \frac{1 - \hat{A}_{i,s}^{(k-1),\pm}}{\ell} \right) < 0 \right\}. \tag{33}$$

The constraint in (33) determines if the detection threshold will be broken if an alarm is triggered at the time instant k. This forces an attack signal $\xi_{i,y,(s)}^{(k)}$ to result in a desired residual element sign, such that an alarm is not triggered.

A similar restriction for both alarm rates is necessary as either one (i.e., alarm rate for the opposite sign that approaches the maximum bound) nears the minimum threshold bound, Ω_{-}. An attacker must ensure that an alarm is triggered before the given alarm rate for an sth residual element falls below the minimum detection bound, such that the sth attack signal element satisfies

$$\xi_{i,y,(s)}^{(k)} = \left\{ \xi_{i,y,(s)}^{(k)} \underset{-}{\overset{+}{\lessgtr}} -\boldsymbol{C}_s \boldsymbol{e}_i^{(k)} - \eta_{i,s}^{(k)} \;\middle|\; \left(\Omega_{-} - \hat{A}_{i,s}^{(k-1+\left|\pm\tau - S_{i,s}^{(k-1),\pm}\right|),\pm} \right) > 0 \right\} \tag{34}$$

such that

$$\hat{A}_{i,s}^{(k'),\pm} = \hat{A}_{i,s}^{(k'-1),\pm} - \frac{\hat{A}_{i,s}^{(k'-1),\pm}}{\ell} \tag{35}$$

where $\forall k' = k, \ldots, k + (\left| \pm\tau - S_{i,s}^{(k'-1),\pm} \right| - 1)$ denotes the number of time instants needed for the CUSIGN test variable $S_{i,s}^{(k)}$ to reach the CUSIGN threshold $\pm\tau$ in order to trigger an alarm.

Communication attacks: In this article, we assume that an attacker can manipulate any information $\mathcal{I}_i^{(k)}$ sent from communication broadcasts from a robot $i \in \mathcal{V}$, which contains the robot's state estimate, input, and measurements. For the case of a communication attack, we provide a worst-case scenario when the broadcast state estimate information from a robot i is altered by a malicious attacker (i.e., $\boldsymbol{\xi}_{i,x}^{(k)} \neq \boldsymbol{0}$). The neighboring robots $j \in \mathcal{C}_i$ monitor for inconsistent information received from robot i, as it would be unaware of an attacker maliciously altering its information via communication broadcasts. The objective for an attacker is to avoid detection from the neighbors that are monitoring robot i.

The state prediction on-board a neighboring robot j monitoring a robot i is a function of the information $\mathcal{I}_i^{(k-1)}$ sent at the previous time instant $k-1$:

$$\begin{aligned}\hat{\boldsymbol{x}}_{ji}^{(k|k-1)} &= f\left(\hat{\boldsymbol{x}}_i^{(k-1|k-1)}, \boldsymbol{u}_i^{(k-1|k-1)}, \boldsymbol{\xi}_{i,x}^{(k-1)}, \boldsymbol{\xi}_{i,u}^{(k-1)}\right) \\ &= \boldsymbol{A}_d\left(\hat{\boldsymbol{x}}_i^{(k-1|k-1)} + \boldsymbol{\xi}_{i,x}^{(k-1)}\right) \\ &\quad + \boldsymbol{B}_d\left(\boldsymbol{u}_i^{(k-1|k-1)} + \boldsymbol{\xi}_{i,u}^{(k-1)}\right) \end{aligned} \tag{36}$$

where $\boldsymbol{A}_d$ and $\boldsymbol{B}_d$ are discrete-time equivalents of $\boldsymbol{A}$ and $\boldsymbol{B}$ in (1) such that the inter-robot residual on robot j to monitor robot i follows:

$$\begin{aligned}\breve{\boldsymbol{r}}_{ji}^{(k)} &= \hat{\boldsymbol{x}}_i^{(k|k)} + \boldsymbol{\xi}_{i,x}^{(k)} - \boldsymbol{A}_d\left(\hat{\boldsymbol{x}}_i^{(k-1|k-1)} + \boldsymbol{\xi}_{i,x}^{(k-1)}\right) \\ &\quad - \boldsymbol{B}_d\left(\boldsymbol{u}_i^{(k-1|k-1)} + \boldsymbol{\xi}_{i,u}^{(k-1)}\right) \\ &= \left(\hat{\boldsymbol{x}}_i^{(k|k)} + \boldsymbol{\xi}_{i,x}^{(k-1)}\right) - \hat{\boldsymbol{x}}_{ji}^{(k|k-1)} \in \mathbb{R}. \end{aligned} \tag{37}$$

Authorized licensed use limited to: University of Virginia Libraries. Downloaded on February 19,2022 at 19:19:55 UTC from IEEE Xplore. Restrictions apply.

An attacker can manipulate the qth inter-robot state residual element sign by choosing an attack vector element of the broadcast state estimate to satisfy

$$\operatorname{sgn}\left(\breve{r}_{ji,q}^{(k)}\right)=\begin{cases}1, \text{ if } \xi_{i,x,(q)}^{(k)}>\hat{x}_{ji,q}^{(k|k-1)}-\hat{x}_{i,q}^{(k|k)}\\ -1, \text{ if } \xi_{i,x,(q)}^{(k)}<\hat{x}_{ji,q}^{(k|k-1)}-\hat{x}_{i,q}^{(k|k)}\end{cases}. \quad (38)$$

Similar to (33) and (34), an attack can manipulate the qth element of the sent state estimate signal $\xi_{i,x,(q)}^{(k)}$ in order to maximize the alarm rates for state residual by

$$\xi_{i,x,(q)}^{(k)}=\Bigg\{\xi_{i,x,(q)}^{(k)} \underset{-}{\overset{+}{\lessgtr}} \hat{x}_{i,q}^{(k|k)}-\hat{x}_{ji,q}^{(k|k)} \Bigg| \left(\Omega_{+}-\hat{A}_{ji,q}^{(k-1),\pm}-\frac{1-\hat{A}_{ji,q}^{(k-1),\pm}}{\ell}\right)<0\Bigg\} \quad (39)$$

and, similarly, to ensure the alarm rate never reaches the lower bound, the attack signal needs to satisfy

$$\xi_{i,x,(q)}^{(k)}=\Bigg\{\xi_{i,x,(q)}^{(k)} \underset{-}{\overset{+}{\lessgtr}} \hat{x}_{i,q}^{(k|k)}-\hat{x}_{ji,q}^{(k|k)} \Bigg| \left(\Omega_{-}-\hat{A}_{ji,q}^{(k-1+\left|\pm\tau-S_{ji,q}^{(k-1),\pm}\right|),\pm}\right)>0\Bigg\} \quad (40)$$

where

$$\hat{A}_{ji,q}^{(k'),\pm}=\hat{A}_{ji,q}^{(k'-1),\pm}-\frac{\hat{A}_{ji,q}^{(k'-1),\pm}}{\ell} \quad (41)$$

and $\forall k'=k,\ldots,k+(\left|\pm\tau-S_{ji,q}^{(k'-1),\pm}\right|-1)$.

We note that, since the CUSIGN attack detector monitors only the signed values of the residual elements (i.e., magnitude is overlooked), it is not possible to quantify the worst-case effects of the cyber attack in terms of true system state deviation with CUSIGN operating as the lone on-board detector. However, when augmented in parallel with a traditional magnitude-based detector [12]–[20], the impact on state deviation due to a cyber attack may be quantified.

APPENDIX B
PROOF OF THEOREM 1

Proof: To prove Theorem 1, we use a similar argument as in [43] and [54]. We first derive the potential energy of the system considering the removal of nodes due to detection of cyber attacks and then show that the energy of the system after detection converges to a rest state. The stored potential energy of each robot i in the network $\mathcal{V}$ is described as

$$U_i^{-}=\sum_{j\in\mathcal{V}}\left[\sum_{h\in\Delta h}\kappa_{jh}(l_{jh}-l_r^0)^2\right], \quad i\neq j\neq h. \quad (42)$$

where $\Delta h\subset\mathcal{S}_j\setminus\{i\}$ represents the change in neighboring robots for a robot j contained in the set $\mathcal{S}_j$ due to the removal of robot i. The assumption in (42) is that, because of the GG rule, by removing a robot i, new connections may appear between the remaining uncompromised robots. The total system potential energy $\boldsymbol{U}$ is then the sum of the stored potential energy of each robot $i\in\mathcal{V}$:

$$\boldsymbol{U}=\sum_{i\in\mathcal{V}}U_i^{-}. \quad (43)$$

Similarly, if a robot i is included into the system, the stored potential energy is described as

$$U_i^{+}=\sum_{j\in\mathcal{V}}\left[\sum_{h\in\Delta h}\kappa_{jh}(l_{jh}-l_r^0)^2\right], \quad i\neq j\neq h. \quad (44)$$

Let $\mathcal{V}^{A}\subset\mathcal{V}$ be the set of detected compromised robots. Given an instant of time when a robot i is removed due to an attack or introduced into the network, any uncompromised robots $j\in\mathcal{V}\setminus\mathcal{V}^{\mathcal{A}}$ reconverge to a new network equilibrium due to the changed number of uncompromised robots in the network, denoted by $|\mathcal{V}\setminus\mathcal{V}^{\mathcal{A}}|$. In the case of the removal of robot i, the remaining uncompromised robots $j\in\mathcal{V}\setminus\{i\}$ converge to the new network equilibrium by first removing edges to robot i, i.e., $(j,i)\notin\mathcal{E}_{\mathcal{U}}$. Thereafter, the robots $j\in\mathcal{V}\setminus\{i\}$ construct edges to the new neighbors $h\in\Delta h\subset\mathcal{S}_j$, such that $(j,h)\in\mathcal{E}_{\mathcal{U}}$, to dissipate any stored energy U_i^{-} that belonged to robot i after it is removed. Conversely, when i is introduced to the network, the robots $j\in\mathcal{V}$ update their virtual spring edges by (29) considering that robot i is now joining the system, i.e., $\mathcal{V}\cup\{i\}$, to converge to a new equilibrium.

As edge switching occurs due to network reconfiguration, the uncompromised robots $j\in\mathcal{V}\setminus\mathcal{V}^{\mathcal{A}}$ dissipate the stored potential energy U_i^{-} (or U_i^{+}) of robot $i\in\mathcal{V}^{\mathcal{A}}$ in order to converge to an equilibrium (i.e., rest state). Next, we prove stability of the system assuming that a reconfiguration of the system has happened (i.e., by removing or adding a node to the network).

Static stability: Let us consider the scenario, in which the network topology is not switching after the removal of a compromised robot in $\mathcal{V}^{\mathcal{A}}$ or introducing a robot, as described in (42) and (44). We let the total energy function of the system, including any remaining available potential energy from the removal or introduction of a robot, be described as

$$V=\sum_{i\in\mathcal{V}\setminus\mathcal{V}^{A}}\frac{1}{2}\Bigg[\sum_{j\in\mathcal{S}_i}\kappa_{ij}(l_{ij}-l_r^0)^2+\sum_{o\in\mathcal{O}_i}\kappa_{io}(l_{io}-l_o^0)^2+\kappa_{ig}l_{ig}^2+(\dot{\boldsymbol{p}}_i)^{\mathsf{T}}\dot{\boldsymbol{p}}_i\Bigg]. \quad (45)$$

By taking the first-order derivative of the total energy in (45), the time derivative becomes

$$\frac{dV}{dt}=-\sum_{i\in\mathcal{V}\setminus\mathcal{V}^{A}}\left(\gamma_i\left(\dot{\boldsymbol{p}}_i\right)^{\mathsf{T}}\dot{\boldsymbol{p}}_i\right) \quad (46)$$

in which because $\gamma_i>0$, $\forall i\in\mathcal{V}$, we obtain that the total energy dissipation is negative semidefinite. Taking the second derivative of the total system energy, we obtain $\frac{d^2\tilde{}V}{dt^2}=-2\sum_{i\in\mathcal{V}\setminus\mathcal{V}^{A}}(\gamma_i(\dot{\boldsymbol{p}}_i)^{\mathsf{T}}\ddot{\boldsymbol{p}}_i)$, which is bounded and finite if robot velocities and the differences $(l_{ij}-l_r^0)$ and $(l_{io}-l_o^0)$, $\forall i,j\in\mathcal{V}$ are finite.

Dynamic stability: For the purpose of proving dynamic stability, we follow similar techniques to those in [41] and [43] that

Authorized licensed use limited to: University of Virginia Libraries. Downloaded on February 19,2022 at 19:19:55 UTC from IEEE Xplore. Restrictions apply.

introduce an *energy reserve* variable ΔE that cancels switching effects of the network topology. Included in the switching topology of this proof are effects from robots removing or introducing other robots to the network, as described in (42) and (44). Given an interval of time Δt such that a switch occurs to create a different topology for uncompromised robots without network reconfiguration, the energy functions rate of variation is

$$\frac{\Delta V}{\Delta t} = \sum_{i \in \mathcal{V} \setminus \mathcal{V}^{\mathcal{A}}} \left[\frac{1}{2} \sum_{h \in \Delta h} L_U + \frac{1}{2} \sum_{j \in \Delta \mathcal{S}_i} L_r + \frac{1}{2} \sum_{o \in \Delta \mathcal{O}_i} L_o + \frac{1}{2} \kappa_{ig} l_{ig}^2 - \gamma_i (\Delta_t^{\boldsymbol{p}_i})^{\mathsf{T}} \Delta_t^{\boldsymbol{p}_i} \right] \quad (47)$$

where $L_U = \kappa_{ih}(l_{ih} - l_r^0)^2$, $L_r = \kappa_{ij}(l_{ij} - l_r^0)^2$, $L_o = \kappa_{io}(l_{io} - l_o^0)^2$, $\Delta \mathcal{S}_i$ and $\Delta \mathcal{O}_i$ are switches in the network topology (i.e., construction or deconstruction of edge connections), and $\Delta_t^{\boldsymbol{p}_i} = \frac{\Delta \boldsymbol{p}_i}{\Delta t}$.

Next, we build a modified potential function $V' = V + E$, where E is the *global energy reserve* by the following:

$$\frac{\Delta E}{\Delta t} = \frac{1}{2} \sum_{i \in \mathcal{V} \setminus \mathcal{V}^{\mathcal{A}}} \left[- \sum_{h \in \Delta h} L_U - \sum_{j \in \Delta \mathcal{S}_i} L_r - \sum_{o \in \Delta \mathcal{O}_i} L_o - \kappa_{ig} l_{ig}^2 + \gamma_i (\Delta_t^{\boldsymbol{p}_i})^{\mathsf{T}} \Delta_t^{\boldsymbol{p}_i} \right] \quad (48)$$

that is dependent on changes to $\mathcal{S}_i$ and $\mathcal{O}_i \, \forall i \in \mathcal{V}$. Including the expressions (47) and (48) with the first derivative of the modified potential function $\dot{V}' = \dot{V} + \dot{E}$, we obtain the following negative-semidefinite expression:

$$\frac{dV'}{dt} = \frac{dV}{dt} + \frac{dE}{dt} = -\frac{1}{2} \sum_{i \in \mathcal{V} \setminus \mathcal{V}^{\mathcal{A}}} \left(\gamma_i \, (\dot{\boldsymbol{p}}_i)^{\mathsf{T}} \dot{\boldsymbol{p}}_i \right). \quad (49)$$

Again, by taking the second derivative of V', we obtain $\frac{d^2 \tilde{} V'}{dt^2} = -\sum_{i \in \mathcal{V} \setminus \mathcal{V}^{\mathcal{A}}} (\gamma_i (\dot{\boldsymbol{p}}_i)^{\mathsf{T}} \ddot{\boldsymbol{p}}_i)$, which is bounded and finite.

The modified energy function contains the following properties: V' is positive definite, $\dot{V}'$ is negative semidefinite, and $\ddot{V}'$ is bounded and finite. Therefore, by Barbalat's lemma, we can conclude that the virtual spring network considering the modified energy function has stable dynamics.

REFERENCES

[1] D. Claes *et al.*, "Decentralised online planning for multi-robot warehouse commissioning," in *Proc. 16th Int. Conf. Auton. Agents Multiagent Syst.*, 2017, pp. 492–500.

[2] C. J. R. McCook and J. M. Esposito, "Flocking for heterogeneous robot swarms: A military convoy scenario," in *Proc. 39th Southeastern Symp. Syst. Theory*, 2007, pp. 26–31.

[3] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Syst. Mag.*, vol. 27, no. 2, pp. 71–82, Apr. 2007.

[4] G. Jing and L. Wang, "Multiagent flocking with angle-based formation shape control," *IEEE Trans. Autom. Control*, vol. 65, no. 2, pp. 817–823, Feb. 2020.

[5] D. van der Walle, B. Fidan, A. Sutton, C. Yu, and B. D. O. Anderson, "Non-hierarchical UAV formation control for surveillance tasks," in *Proc. Amer. Control Conf.*, 2008, pp. 777–782.

[6] M. Fachri, S. Juniastuti, S. M. S. Nugroho, and M. Hariadi, "Crowd evacuation using multi-agent system with leader-following behaviour," in *Proc. 4th Int. Conf. New Media Stud.*, 2017, pp. 92–97.

[7] R. Maeda, T. Endo, and F. Matsuno, "Decentralized navigation for heterogeneous swarm robots with limited field of view," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 904–911, Apr. 2017.

[8] F. Pasqualetti, F. Dörfler, and F. Bullo, "Attack detection and identification in cyber-physical systems," *IEEE Trans. Autom. Control*, vol. 58, no. 11, pp. 2715–2729, Nov. 2013.

[9] A. Zaman, B. Safarinejadian, and W. Birk, "Security analysis and fault detection against stealthy replay attacks," *Int. J. Control*, pp. 1–14, 2020, doi: 10.1080/00207179.2020.1862917.

[10] A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry, "Attacks against process control systems: Risk assessment, detection, and response," in *Proc. 6th ACM Symp. Inf., Comput. Commun. Secur.*, 2011, pp. 355–366.

[11] Y. Mo and B. Sinopoli, "On the performance degradation of cyber-physical systems under stealthy integrity attacks," *IEEE Trans. Autom. Control*, vol. 61, no. 9, pp. 2618–2624, Sep. 2016.

[12] Y. Mo, E. Garone, A. Casavola, and B. Sinopoli, "False data injection attacks against state estimation in wireless sensor networks," in *Proc. IEEE 49th Conf. Decis. Control*, 2010, pp. 5967–5972.

[13] C. Murguia and J. Ruths, "Characterization of a CUSUM model-based sensor attack detector," in *Proc. IEEE 55th Conf. Decis. Control*, Dec. 2016, pp. 1303–1309.

[14] C. Murguia and J. Ruths, "On model-based detectors for linear time-invariant stochastic systems under sensor attacks," *IET Control Theory Appl.*, vol. 13, no. 8, pp. 1051–1061, 2019.

[15] C. Kwon, S. Yantek, and I. Hwang, "Real-time safety assessment of unmanned aircraft systems against stealthy cyber attacks," *J. Aerosp. Inf. Syst.*, vol. 13, no. 1, pp. 27–45, 2016.

[16] C. Kwon, W. Liu, and I. Hwang, "Security analysis for cyber-physical systems against stealthy deception attacks," in *Proc. Amer. Control Conf.*, Jun. 2013, pp. 3344–3349.

[17] F. Miao, Q. Zhu, M. Pajic, and G. J. Pappas, "Coding sensor outputs for injection attacks detection," in *Proc. 53rd IEEE Conf. Decis. Control*, Dec. 2014, pp. 5776–5781.

[18] D. Dionne, H. Michalska, Y. Oshman, and J. Shinar, "Novel adaptive generalized likelihood ratio detector with application to maneuvering target tracking," *J. Guid., Control, Dyn.*, vol. 29, no. 2, pp. 465–474, 2006.

[19] N. Hashemi, C. Murguia, and J. Ruths, "A comparison of stealthy sensor attacks on control systems," in *Proc. Annu. Amer. Control Conf.*, Jun. 2018, pp. 973–979.

[20] T. Rangaswamy, C. Murguia, and J. Ruths, "Tuning windowed chi-squared detectors for sensor attacks," in *Proc. Annu. Amer. Control Conf.*, 2018, pp. 1752–1757.

[21] D. Zhang, G. Feng, Y. Shi, and D. Srinivasan, "Physical safety and cyber security analysis of multi-agent systems: A survey of recent advances," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 2, pp. 319–333, Feb. 2021.

[22] K. Saulnier, D. Saldaña, A. Prorok, G. J. Pappas, and V. Kumar, "Resilient flocking for mobile robot teams," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 1039–1046, Apr. 2017.

[23] Y. Zheng and L. Wang, "Consensus of switched multiagent systems," *IEEE Trans. Circuits Syst. II: Exp. Briefs*, vol. 63, no. 3, pp. 314–318, Mar. 2016.

[24] Y. Shang, "Scaled consensus of switched multi-agent systems," *IMA J. Math. Control Inf.*, vol. 36, no. 2, pp. 639–657, 2019.

[25] Y. Zhu, Y. Zheng, and L. Wang, "Containment control of switched multi-agent systems," *Int. J. Control*, vol. 88, no. 12, pp. 2570–2577, 2015.

[26] L. Guerrero-Bonilla, A. Prorok, and V. Kumar, "Formations for resilient robot teams," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 841–848, Apr. 2017.

[27] J. Chen and Q. Zhu, "Resilient and decentralized control of multi-level cooperative mobile networks to maintain connectivity under adversarial environment," in *Proc. IEEE 55th Conf. Decis. Control*, 2016, pp. 5183–5188.

[28] Y. Wang and H. Ishii, "Resilient consensus through event-based communication," *IEEE Control Netw. Syst.*, vol. 7, no. 1, pp. 471–482, Mar. 2020.

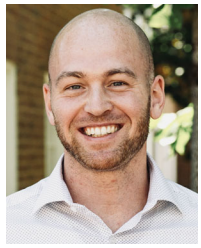
[29] Y. Chen, S. Kar, and J. M. F. Moura, "Resilient distributed estimation through adversary detection," *IEEE Trans. Signal Process.*, vol. 66, no. 9, pp. 2455–2469, May 2018.

[30] C. Zhao, J. He, and J. Chen, "Resilient consensus with mobile detectors against malicious attacks," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 4, no. 1, pp. 60–69, Mar. 2018.

[31] W. Zeng and M. Chow, "Resilient distributed control in the presence of misbehaving agents in networked control systems," *IEEE Trans. Cybern.*, vol. 44, no. 11, pp. 2038–2049, Nov. 2014.

Authorized licensed use limited to: University of Virginia Libraries. Downloaded on February 19,2022 at 19:19:55 UTC from IEEE Xplore. Restrictions apply.

[32] A. Khazraei, H. Kebriaei, and F. R. Salmasi, "Replay attack detection in a multi agent system using stability analysis and loss effective watermarking," in *Proc. Amer. Control Conf.*, 2017, pp. 4778–4783.
[33] S. Lee and B. Min, "Distributed direction of arrival estimation-aided cyberattack detection in networked multi-robot systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 1–9.
[34] E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, no. 1/2, pp. 100–115, 1954.
[35] C. Bai and V. Gupta, "On Kalman filtering in the presence of a compromised sensor: Fundamental performance bounds," in *Proc. Amer. Control Conf.*, Jun. 2014, pp. 3029–3034.
[36] P. J. Bonczek, S. Gao, and N. Bezzo, "Model-based randomness monitor for stealthy sensor attacks," in *Proc. Amer. Control Conf.*, 2020, pp. 2036–2042.
[37] P. J. Bonczek and N. Bezzo, "Memoryless cumulative sign detector for stealthy CPS sensor attacks," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 838–844, 2020.
[38] P. J. Bonczek and N. Bezzo, "Detection of hidden attacks on cyber-physical systems from serial magnitude and sign randomness inconsistencies," in *Proc. Amer. Control Conf.*, 2021, pp. 3281–3287.
[39] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bull.*, vol. 1, no. 6, pp. 80–83, 1945.
[40] C. Cammarota, "The difference-sign runs length distribution in testing for serial independence," *J. Appl. Statist.*, vol. 38, no. 5, pp. 1033–1043, 2011.
[41] B. Shucker, T. D. Murphey, and J. K. Bennett, "Convergence-preserving switching for topology-dependent decentralized systems," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1405–1415, Dec. 2008.
[42] N. Bezzo, P. J. Cruz, F. Sorrentino, and R. Fierro, "Decentralized identification and control of networks of coupled mobile platforms through adaptive synchronization of chaos," *Physica D: Nonlinear Phenomena*, vol. 267, pp. 94–103, 2014.
[43] N. Bezzo, Y. Yan, R. Fierro, and Y. Mostofi, "A decentralized connectivity strategy for mobile router swarms," *IFAC Proc. Vol.*, vol. 44, no. 1, pp. 4501–4506, 2011.
[44] Q. Chen, Y. Meng, and J. Xing, "Shape control of spacecraft formation using a virtual spring-damper mesh," *Chin. J. Aeronaut.*, vol. 29, no. 6, pp. 1730–1739, 2016.
[45] W. Ren, "Formation keeping and attitude alignment for multiple spacecraft through local interactions," *J. Guid., Control, Dyn.*, vol. 30, no. 2, pp. 633–638, 2007.
[46] F. Lin, M. Fardad, and M. R. Jovanovic, "Optimal control of vehicular formations with nearest neighbor interactions," *IEEE Trans. Autom. Control*, vol. 57, no. 9, pp. 2203–2218, Sep. 2012.
[47] M. Conti, N. Dragoni, and V. Lesyk, "A survey of man in the middle attacks," *IEEE Commun. Surv. Tut.*, vol. 18, no. 3, pp. 2027–2051, Jul.–Sep. 2016.
[48] S. M. Ross, *Introduction to Probability Models*, 9th ed. Orlando, FL, USA: Academic, 2006.
[49] S. I. Gass and C. M. Harris, "Encyclopedia of operations research and management science," *J. Oper. Res. Soc.*, vol. 48, no. 7, pp. 759–760, 1997.
[50] D. Brook and D. A. Evans, "An approach to the probability distribution of CUSUM run length," *Biometrika*, vol. 59, no. 3, pp. 539–549, 1972.
[51] N. Bezzo, B. Griffin, P. Cruz, J. Donahue, R. Fierro, and J. Wood, "A cooperative heterogeneous mobile wireless mechatronic system," *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 1, pp. 20–31, Feb. 2014.
[52] K. R. Gabriel and R. R. Sokal, "A new statistical approach to geographic variation analysis," *Syst. Biol.*, vol. 18, no. 3, pp. 259–278, 1969.
[53] E. Bertin, J.-M. Billiot, and R. Drouilhet, "Continuum percolation in the Gabriel graph," *Adv. Appl. Probab.*, vol. 34, no. 4, pp. 689–701, 2002.
[54] B. Shucker, "Control of distributed robotic macrosensors," Ph.D. dissertation, Dept. Comput. Sci., Univ. Colorado, Boulder, CO, USA, 2006.
[55] K. Derr and M. Manic, "Extended virtual spring mesh (EVSM): The distributed self-organizing mobile ad hoc network for area exploration," *IEEE Trans. Ind. Electron.*, vol. 58, no. 12, pp. 5424–5437, Dec. 2011.
[56] T. X. Lin, E. Yel, and N. Bezzo, "Energy-aware persistent control of heterogeneous robotic systems," in *Proc. Annu. Amer. Control Conf.*, 2018, pp. 2782–2787.
[57] F. Bullo, J. Cortés, and S. Martínez, *Distributed Control of Robotic Networks*. Princeton, NJ, USA: Princeton Univ. Press, 2009.
[58] J. A. Bondy and U. S. R. Murty, *Graph Theory With Applications*. New York, NY, USA: Elsevier, 1976.
[59] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.
[60] M. Pajic *et al.*, "Robustness of attack-resilient state estimators," in *Proc. ACM/IEEE Int. Conf. Cyber-Phys. Syst.*, 2014, pp. 163–174.

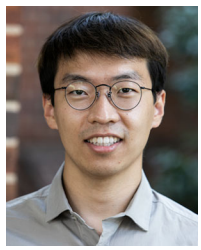


Paul J Bonczek received the B.S. degree dual majoring in electrical and computer engineering and applied mathematics from the State University of New York Polytechnic Institute, Utica, NY, USA, in 2016, and the M.E. degree in electrical engineering in 2021 from the University of Virginia, Charlottesville, VA, USA, where he is currently working toward the Ph.D. degree in electrical engineering with a focus on resiliency in cyber-physical systems.

His research interests include runtime attack detection, resilient multiagent swarms, cyber-physical system security, resilient path planning and control, and adaptive systems.

Rahul Peddi received the B.S. degree in mechanical and nuclear engineering from Virginia Commonwealth University, Richmond, VA, USA, in 2017. He is currently working toward the Ph.D. degree in systems engineering with the University of Virginia, Charlottesville, VA.

His research interests include motion planning, human–robot interaction, multirobot systems, and machine learning.



Shijie Gao received the B.S. degree in automation from the Beijing Institute of Technology, Beijing, China, in 2017. He is currently working toward the Ph.D. degree in computer engineering with the University of Virginia, Charlottesville, VA, USA.

His research interests include transfer learning, motion planning, and robotic system failure detection and recovery.

Nicola Bezzo received the B.S. and M.S. degrees (Hons.) (*summa cum laude*) in electrical engineering from the Politecnico di Milano, Milan, Italy, in 2006 and 2008, respectively, and the Ph.D. degree in electrical and computer engineering from the University of New Mexico, Albuquerque, NM, USA, in 2012.

He is currently an Assistant Professor with the Department of Engineering Systems and Environment and the Department of Electrical and Computer Engineering, University of Virginia (UVA), Charlottesville, VA, USA, where he also holds a courtesy appointment with the Department of Computer Science. He is one of the co-founders of the Link Lab and he directs the Autonomous Mobile Robots Lab, UVA. His research interests include resilient motion planning and control of autonomous mobile robots, assured autonomy, cyber physical system cyber-security, and heterogeneous robotic systems.

Dr. Bezzo is the recipient of the 2010 Gold Medal from the Politecnico School of Engineering, the 2016 *Robotics and Automation Magazine* Best Paper Award, and the Best Paper Award at the 2014 CPSWeek International Conference on Cyber-Physical Systems.

Authorized licensed use limited to: University of Virginia Libraries. Downloaded on February 19,2022 at 19:19:55 UTC from IEEE Xplore. Restrictions apply.