Range-Based Multi-Robot Integrity Monitoring For Cyberattacks and Faults: An Anchor-Free Approach

Vishnu Vijay, *Graduate Student Member, IEEE*, Kartik A. Pant, *Graduate Student Member, IEEE*, Minhyun Cho, Yifan Guo, James M. Goppert, *Member, IEEE*, and Inseok Hwang, *Member, IEEE*

***Abstract*—Coordination of multi-robot systems (MRSs) relies on efficient sensing and reliable communication among the robots. However, the sensors and communication channels of these robots are often vulnerable to cyberattacks and faults, which can disrupt their individual behavior and the overall objective of the MRS. In this work, we present a multi-robot integrity monitoring framework that utilizes inter-robot range measurements to (i) detect the presence of cyberattacks or faults affecting the MRS, (ii) identify the affected robot(s), and (iii) reconstruct the resulting localization error of these robot(s). The proposed iterative algorithm leverages sequential convex programming and alternating direction of multipliers method to enable real-time and distributed implementation. Our approach is validated using numerical simulations and demonstrated using PX4-SiTL in Gazebo on an MRS, where certain agents deviate from their desired position due to a GNSS spoofing attack. Furthermore, we demonstrate the scalability and interoperability of our algorithm through mixed-reality experiments by forming a heterogeneous MRS comprising real Crazyflie UAVs and virtual PX4-SiTL UAVs working in tandem.**

***Index Terms*—Multi-robot systems (MRSs), failure detection and recovery, networked robots.**

I. Introduction

RECENT advances in sensing, networking, planning, and control, and the development of high-performance computational hardware, have enabled the deployment of robotic systems (RSs) for real-world applications. Within robotics, a multi-robot system (MRS) refers to the coordination and teaming of more than one robot to accomplish complex tasks autonomously. Such collaboration allows the system to solve complicated problems that could not have been possible by a single robot; it has been widely used in warehouse logistics [1], vehicle platooning [2], connected vehicle-to-vehicle operations [3], surveillance [4], and disaster relief [5].

However, the reliance on autonomous MRS often raises concerns about safety, security, and reliability in real-world settings. Furthermore, these systems present security challenges as malicious actors (i.e., hackers) can alter sensor measurements and/or jam the communication channels to disrupt their operations. The presence of an attacked or faulty robot in the network could potentially manipulate the entire MRS, leading to catastrophic failures and harming critical infrastructure, such as UAVs crashing into buildings, etc. To maintain the integrity of the system, it is essential to ensure (i) safety (i.e., avoiding collision or reaching an undesired state) and (ii) liveliness (i.e., the ability to complete the task without any disruption). Thus, detecting and identifying rogue or faulty robots and isolating them becomes essential for safe and efficient multi-robot operations.

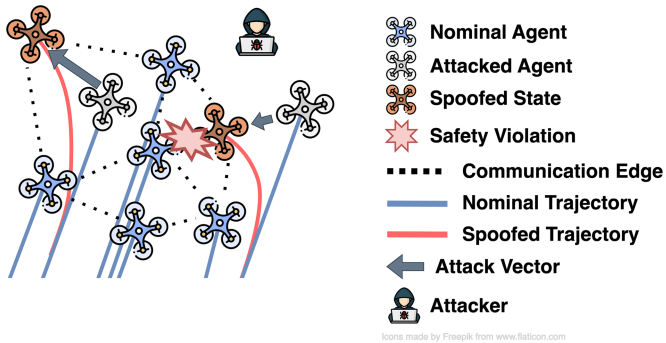


Fig. 1. Illustration of spoofing attacks in a multi-robot system.

Unlike a single-robot system, in an MRS, there exists an inherent redundancy of information in the form of locally sensed or communicated information from neighbors. This external information can be leveraged to enhance the robustness of the MRS against cyberattacks and system faults. One such type of information is inter-robot measurements between robots of the MRS. However, combining these external inputs from other robots (healthy or otherwise) in a real-time distributed fashion is challenging. One main challenge is how we trust which robots are relaying correct information. Existing works on robust multi-robot anomaly detection have been focused on an anchor-based approach [6], where some of the robots are considered anchors, whose positions are assumed correct. Anchor-based approaches are powerful in theory as there is always at least one robot against which to check for anomalies. In practice, however, malicious agents can adversarially design their attacks to affect an anchor, which could affect the anomaly detection process for the whole MRS. Thus, we require a method to investigate every robot of an MRS, necessitating an anchor-free approach. In this work, we consider an anchor-free approach that utilizes information from each robot to identify and localize errors in the system. We propose a distributed runtime integrity monitor for an MRS that leverages inter-robot range measurements to

Received 3 August 2024; accepted 6 January 2025. Date of publication 27 January 2025; date of current version 5 February 2025. This letter was recommended for publication by Associate Editor G. Notomista and Editor M. A. Hsieh upon evaluation of the reviewers' comments. This work was supported by the Secure Systems Research Center (SSRC) at the Technology Innovation Institute (TII), UAE. *(Corresponding author: Vishnu Vijay.)*

The authors are with the School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN 47906 USA (e-mail: vvijay@purdue.edu; kpant@purdue.edu; mhcho@purdue.edu; guo781@purdue.edu; jgoppert@purdue.edu; ihwang@purdue.edu).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2025.3534068, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3534068

indiscriminately detect and identify rogue or faulty robots. We design an iterative algorithm that solves the nonlinear integrity monitoring optimization problem using sequential convex programming (SCP). This SCP problem can be solved efficiently in a distributed manner using the alternating direction of multiplier method (ADMM) [7].

We introduce two different metrics to assess the integrity of an MRS: first, *overall system integrity*, which evaluates the integrity of the overall MRS to inform system error detection, and second, *robot integrity*, which assesses the integrity of each robot to inform robot error identification. Through extensive numerical simulations, we demonstrate the robustness of our proposed algorithm, which is subject to errors in the state estimation and noise in the inter-robot range measurements. We also devise a threshold-based mechanism to bypass the warm start method of ADMM and reset internal parameters of the optimization problem, which specifically addresses the time-varying network topology of an MRS. This enables our algorithm to more effectively adapt to changes in its network topology. For example, in a multi-robot inspection and surveillance mission, if a certain number of robots are identified as attacked or faulty and the system's network topology is reconfigured, our proposed approach will adjust the fault monitor by resetting the algorithm's internal parameters and bypassing warm start.

In this letter, our main contributions are as follows,

- We propose a novel anchor-free multi-robot integrity monitoring framework using inter-robot range measurements to detect cyberattacks or system faults that could go unnoticed by traditional single-robot fault detectors.
- We validate the performance and robustness of the proposed algorithm through extensive numerical simulations with noise, as well as a time-varying network topology resulting from a cyberattack or fault.
- We create a Mixed Reality inter-robot range sensor emulation by leveraging the physics engine of Gazebo.
- Finally, we demonstrate the effectiveness of the proposed algorithm through Mixed Reality experiments on a heterogeneous MRS comprised of real Crazyflie UAVs and virtual PX4-SiTL.

The rest of the paper is organized as follows. Section II summarizes related works in multi-agent fault detection and isolation. Section III reviews the background concepts from graph theory, control theory, and optimization utilized to set up the multi-robot integrity monitoring problem. In Section IV, we present the details of the algorithm, including a sensitivity analysis of the robustness of the proposed approach subject to noise and a time-varying network topology. Section V presents the numerical simulation results and elucidates the effect of the algorithm's parameters on the convergence of the algorithm. Section VI describes the experimental validation of the proposed algorithm using Mixed-Reality-in-the-loop (MRiTL) experiments comprised of real Crazyflie UAVs and virtual PX4-SiTL. Finally, Section VII concludes the paper and presents future directions.

II. Related Works

Multi-agent fault detection and identification is a topic that has fostered exciting research in several fields. In control theory, significant efforts have been made to theoretically analyze multi-agent fault detection and isolation algorithms using approaches such as H_∞ performance indices [8], [9], residual testing [10], l_1 norm minimization [11], and interval observers [12]. These works extend the scope of fault detection from single-agent to multi-agent systems; however, they are limited to linear measurement models and do not accommodate nonlinear measurement models. On the other hand, cooperative localization using nonlinear measurements (e.g., range, bearing, etc.) has been proposed in [13], [14]; yet, the proposed algorithms mainly consider cyberattacks in a shared communication network, which is a specific case of our work. In our earlier work [15], [16], we proposed a rigidity theory approach that can be applied to various nonlinear measurement models. We focused on providing a theoretical foundation for anchor-free multi-agent fault detection and isolation while considering an idealistic noise-free setting and static network topology. We extend the work in [15] to enable its use for practical applications, with a focus on robustness to noise and time-varying network topologies.

Researchers in the machine learning community address single- and multi-agent anomaly detection solely through signal patterns. Major approaches in this domain include prediction [17], [18], reconstruction [19], and classification [20], [21], [22]. Machine learning-based methods, especially detecting anomalies using deep neural networks, often demonstrate superior performance and flexibility compared to control theory-based methods on benchmark datasets. However, these methods typically rely on extensive experimental validation to establish reliability. A statistical classifier [23] can provide more conservative, and thus reliable, detection; however, the approach still requires an extensive data set with anomalies, which is difficult to obtain in safety-critical applications. Therefore, the lack of conservative control-theoretical guarantees may limit their suitability for safety-critical applications. Additionally, they often overlook practical constraints such as communication limitations in real-world systems [6], which can impact anomaly detection performance. Our proposed method enables real-time anomaly detection in a distributed setting, while backed by control-theoretical convergence properties.

In the robotics community, researchers combine ideas from both control and machine learning areas to focus on real-world MRSs, where the system dynamics, communication ability, power conditions, etc., are further specified [24], [25], [26]. Such specifications can facilitate the validation of their algorithms on real-world RSs, but may also limit the scope of their work to a specific MRS. For example, Suarez et al. [25] proposed a voting-based fault detection algorithm for a multi-UAV system where each UAV raises a flag on its direct neighbors when the discrepancy between independent state estimators surpasses a threshold. However, such an approach requires at least two perfectly healthy UAVs that can play as anchors (attack-free) equipped with independent estimators, which might not be practical as the multiple robots in a swarm can be attacked simultaneously. Lee et al. [27] proposed a data-driven approach to select the most effective fault detection metric. However, their method can only select from a pre-defined metric set, which may differ substantially for each system and limits its applicability to a general MRS. On the contrary, our proposed approach applies to a wide range of inter-robot measurements accessible for most MRSs. Other representative works in robotics for multi-robot fault detection and isolation include [12], [24], [26], [28], [29], which study fault detection in RSs but don't consider cyberattacks, which are carefully designed to evade statistical fault detection methods.

III. Background

A. Notation

The set of real numbers and integers are denoted as $\mathbb{R}$, $\mathbb{Z}$, and the superscript $+$ stands for the non-negativeness of the set. Then, the vector $\mathbf{v}[i] \in \mathbb{R}^{\ell}$ refers to the i^{th} block of a block vector $\mathbf{v} \in \mathbb{R}^{k\ell}$. Then, the l_q norm of $\mathbf{v}$ is denoted as:

$$\|\mathbf{v}\|_{2,q} = \begin{cases} \sum_{i=1}^{k} \mathbb{I}(\|\mathbf{v}[i]\|_2 > 0) & \text{when} \quad q = 0 \\ (\sum_{i=1}^{k} \|\mathbf{v}[i]\|_2^q)^{1/q} & \text{when} \quad 0 < q < \infty \\ \max_{1 \le i \le k} (\|\mathbf{v}[i]\|_2) & \text{when} \quad q = \infty, \end{cases}$$

where $\mathbb{I}(\cdot)$ is an indicator function that yields 1 when the entity is positive and 0 otherwise. Therefore, the vector's l_0 norm represents the block vector's non-zero blocks, i.e., block sparsity. For a given index set $\mathcal{S}$, we denote the cardinality of the set as $|\mathcal{S}|$; in other words, $\mathcal{S} = \{1, 2, \ldots, |\mathcal{S}|\}$.

B. Sensing and Communication Network

We consider an MRS described by a graph $\mathcal{G}$, composed of a set of vertices $\mathcal{V}$ and a set of edges $\mathcal{E}$. With the graph vertices representing the agents of the multi-robot system, the edges represent the measurements that can be computed between the agents. Two robots are considered neighbors if an edge connects them, and the neighbors of robot $i \in \mathcal{V}$ are included in set $\mathcal{N}_i$. Similarly, the set of edges connecting robot $i \in \mathcal{V}$ is denoted by $\mathcal{E}_i$. This structure can be generalized to a hypergraph with hyperedges to allow for the use of inter-robot measurement models that connect three or more robot, such as time difference-of-arrival. An important assumption made in the MRS communication network is that neighboring robots can communicate synchronously with each other.

C. Dynamics, Consensus Protocol and State Estimation

Defining the state of the i^{th} robot of the MRS to be $\mathbf{p}[i] \in \mathbb{R}^{n_i}$, the collective state of the system is represented by the block vector $\mathbf{p} \in \mathbb{R}^{n}$

$$\mathbf{p} = \begin{bmatrix} \mathbf{p}[1]^\top & \mathbf{p}[2]^\top & \ldots & \mathbf{p}[|\mathcal{V}|]^\top \end{bmatrix}^\top \tag{1}$$

Each robot in the MRS, corresponding to the vertices $\mathcal{V}$, is governed by dynamics and a consensus control law with a state estimator. The i^{th} robot dynamics is given as the following linear discrete-time system:

$$\begin{cases} \mathbf{p}[i](k+1) = A_i \mathbf{p}[i](k) + B_i \mathbf{u}[i](k) + E_i \mathbf{w}[i](k) \\ \mathbf{q}[i](k) = C_i \mathbf{p}[i](k) + F_i \mathbf{v}[i](k) + \Gamma_i \mathbf{f}[i](k), \end{cases} \tag{2}$$

where $\mathbf{p}[i](k) \in \mathbb{R}^{n_i}$ is the state of robot i, $\mathbf{u}[i](k) \in \mathbb{R}^{u_i}$ is the control input, $\mathbf{q}[i](k) \in \mathbb{R}^{q_i}$ is the measurement output, $\mathbf{w}[i](k) \in \mathbb{R}^{w_i}$ and $\mathbf{v}[i](k) \in \mathbb{R}^{v_i}$ are the process and measurement noises satisfying the norm bounded condition:

$$\begin{aligned} \|\mathbf{w}[i](k)\|_2^2 &= \mathbf{w}[i](k)^\top \mathbf{w}[i](k) \le W_i \\ \|\mathbf{v}[i](k)\|_2^2 &= \mathbf{v}[i](k)^\top \mathbf{v}[i](k) \le V_i \end{aligned} \tag{3}$$

with $W_i, V_i \in \mathbb{R}^+$ and $k \in \mathbb{Z}^+$. The matrices $A_i \in \mathbb{R}^{n_i \times n_i}$, $B_i \in \mathbb{R}^{n_i \times u_i}$ and $C_i \in \mathbb{R}^{q_i \times n_i}$ denote the system matrix, the input matrix, and the output matrix, respectively. The matrices $E_i \in \mathbb{R}^{n_i \times w_i}$ and $F_i \in \mathbb{R}^{q_i \times v_i}$ are the perturbation matrices of the process and measurement noises, respectively. The attack or fault input is denoted as $\mathbf{f}[i](k) \in \mathbb{R}^{f_i}$ with the associated attack matrix $\Gamma_i \in \mathbb{R}^{q_i \times f_i}$. The system pairs (A_i, B_i) and (A_i, C_i) are assumed to be stabilizable and detectable. Each robot estimates their state using their own measurements $\mathbf{q}[i](k)$ and the following state estimator:

$$\begin{cases} \hat{\mathbf{p}}[i](k+1) = A\mathbf{p}[i](k) + B\mathbf{u}[i](k) \\ \qquad\qquad + L_{o,i}\left(\mathbf{q}[i](k) - \hat{\mathbf{q}}[i](k)\right) \\ \hat{\mathbf{q}}[i](k) = C\hat{\mathbf{p}}[i](k), \end{cases} \tag{4}$$

where $L_{o,i} \in \mathbb{R}^{n_i \times q_i}$ is the estimator gain, $\hat{\mathbf{p}}[i](k) \in \mathbb{R}^{n_i}$ is the estimated state of robot, and $\hat{\mathbf{q}}[i](k) \in \mathbb{R}^{q_i}$ is its estimated output. Then, the consensus control protocol that governs the behavior of the MRS is given as:

$$\mathbf{u}[i](k) = K_{c,i}\Big(\sum_{j \in \mathcal{N}_i} a_{ij}(\hat{\mathbf{p}}[i](k) - \hat{\mathbf{p}}[j](k))\Big) + \mathbf{u}_r(k), \tag{5}$$

where $K_{c,i} \in \mathbb{R}^{n_i \times u_i}$ is the consensus gain, $\mathbf{u}_r(k)$ is the reference trajectory tracking input and a_{ij} is the element of the adjacency matrix $\mathcal{A}$. Since we assume an undirected communication network in the problem, $a_{ij} = 1$ if the robot j is a neighbor of i, i.e., $j \in \mathcal{N}_i$, $a_{ij} = 0$ otherwise. Time index k is dropped in subsequent sections for simplicity. The gains $K_{c,i}$ and $L_{o,i}$ can be designed to satisfy H_∞ performance criterion, which results in the optimal estimation with a bounded error norm, i.e., $\|\mathbf{p} - \hat{\mathbf{p}}\|_2 = \|\boldsymbol{\nu}\|_2 \le \nu_{\max}$ when the system is without attack or fault, i.e., $\mathbf{f}[i] = 0$. The consensus gain and the estimator gain can be implemented by solving LMI [30], [31]. We omit the detailed implementation since it is not in the scope of the paper.

D. Attack or Fault Model

We define the error vector $\mathbf{x} \in \mathbb{R}^n$ as the discrepancy between the true and estimated states. This gives $\mathbf{x} = \mathbf{p} + \boldsymbol{\nu} - \hat{\mathbf{p}}$ for a system with localization error. Defining $\mathcal{D}$ to be the set of affected (i.e., attacked or faulty) robots, we see $|\mathcal{D}| = \|\mathbf{x}\|_{2,0}$. We further assume the error vector $\mathbf{x}$ is sparse. Specifically, we assume $|\mathcal{D}| < \frac{1}{2}|\mathcal{V}|$ as this will provide guarantees to (6) under certain network topologies [16]. This is a reasonable assumption as, in practice, attackers will have a limited budget and recoverable localization faults do not often occur frequently. Finally, we denote the desired estimated error $\hat{\mathbf{x}} = \mathbf{p} - \hat{\mathbf{p}}$.

E. Inter-Robot Measurements

The measurement model $\boldsymbol{\Phi} \in \mathbb{R}^m$ is a block vector with each subvector corresponding to an edge. The subvectors are denoted by $\boldsymbol{\Phi}^{(l)} \in \mathbb{R}^{m_l}$ for $l \in \{0, 1, \ldots, |\mathcal{E}|\}$. The true inter-robot measurement $\mathbf{y} \in \mathbb{R}^m$ can be found according to $\mathbf{y} = \boldsymbol{\Phi}(\mathbf{p})$. There are various types of inter-robot measurements for MRS implementation; for the purposes of this work, we use range measurements due to the ease and cost of implementation. For example, Ultra-Wide Band (UWB) sensors require low processing for range computations. Other types of inter-robot measurements may require more complex sensors that would necessitate more processing power.

Each robot measures the range to its neighbors using noisy sensors. Letting the block vector $\boldsymbol{\omega} \in \mathbb{R}^m$ represent the range sensor noise, we define a noisy inter-robot range measurement $\hat{\mathbf{y}} = \mathbf{y} + \boldsymbol{\omega}$. We assume the noise is norm bounded such that $\|\boldsymbol{\omega}\|_2 \le \omega_{\max}$ for this analysis. When $\mathbf{p} \neq \hat{\mathbf{p}}$ in a noisy, attacked, and/or faulty system, it can be seen that $\mathbf{y} \neq \boldsymbol{\Phi}(\hat{\mathbf{p}})$. We recognize

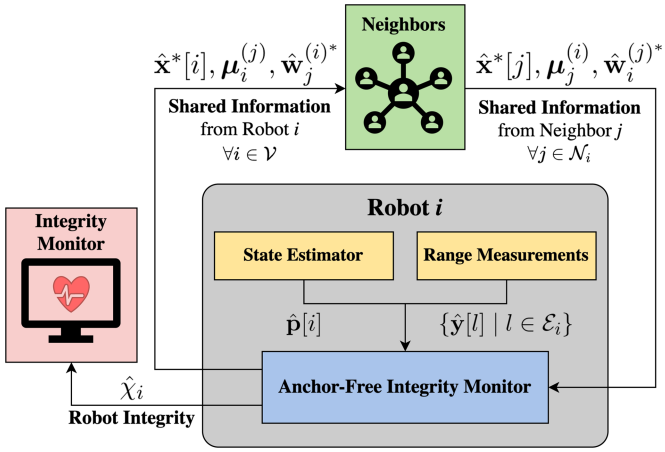


Fig. 2. Overall system architecture.

that including error and noise will result in: $\hat{\mathbf{y}} = \boldsymbol{\Phi}(\hat{\mathbf{p}} + \hat{\mathbf{x}}) + \boldsymbol{\omega}$. Thus, the model $\boldsymbol{\Phi}$ can be leveraged to solve for $\hat{\mathbf{x}}$. We assume no knowledge of $\hat{\mathbf{x}}$ and solve for the error term in a process known as *error reconstruction*.

IV. Anchor-Free Error Detection and Reconstruction

In this section, we describe the steps required to estimate the true error vector $\mathbf{x}$ using robot state estimates and noisy inter-robot range measurements. To do so, we first present the optimization problem, which is relaxed using SCP and ADMM, and then discuss parameter tuning required for real-time and distributed implementation. We define a threshold so only significant estimated error vectors $\hat{\mathbf{x}}$ are considered, while others are attributed to noise or transient behavior. We then address challenges associated with implementing such an algorithm on a real-world system, including noise and time-varying network topology. The overall architecture of our proposed MRS integrity monitor is described in Fig. 2.

A. Error Reconstruction

We adopt the same problem formulation as presented in [15] and seek to find the sparsest error vector (i.e., block error vector with least nonzero subvectors) that explains the inconsistency in our measurement model $\boldsymbol{\Phi}(\hat{\mathbf{p}}) \neq \hat{\mathbf{y}}$. Under noise-free setting [15], this problem can be posed as

$$\begin{aligned} &\min_{\hat{\mathbf{x}} \in \mathbb{R}^n} \|\hat{\mathbf{x}}\|_{2,0} \\ &\text{s. t.} \quad \boldsymbol{\Phi}(\hat{\mathbf{p}} + \hat{\mathbf{x}}) = \hat{\mathbf{y}} \end{aligned} \tag{6}$$

As the objective function and the constraints are nonlinear, the optimization problem (6) cannot be solved efficiently in polynomial time. Also, the coupling of information among robots in the constraints doesn't allow (6) to be solved in a distributed manner. Thus, a relaxed, linearized problem is formulated by applying the SCP and ADMM algorithms.

We first apply SCP, which solves successive local convex optimization problems, by re-linearizing the constraints at each iteration. Let $\bar{\mathbf{x}}$ be the minimized error vector from the convexified sub-problem. The $\bar{\mathbf{x}}$ from successive iterations are summed to reconstruct the error vector. Note that $\hat{\mathbf{x}}$ is the local error vector being minimized. By replacing nonconvex $\|\cdot\|_{2,0}$ with convex $\|\cdot\|_{2,1}$, we introduce the convex objective function $\|\hat{\mathbf{x}} + \bar{\mathbf{x}}\|_{2,1}$. Additionally, the constraint can be rewritten using a first-order Taylor approximation of the measurement model: we let $\mathbf{z} = \hat{\mathbf{y}} - \boldsymbol{\Phi}(\hat{\mathbf{p}} + \bar{\mathbf{x}})$ and $\mathbf{R} = \mathbf{J}_\Phi(\hat{\mathbf{p}} + \bar{\mathbf{x}})$ to form a linearized constraint $\mathbf{R}\hat{\mathbf{x}} = \mathbf{z}$.

This can be further transformed using ADMM, which decouples the optimization problem by breaking it down into smaller parts and is well suited for distributed convex optimization [7]. At each robot $i \in \mathcal{V}$, we introduce a new set of primal variables $\hat{\mathbf{w}}_i^{(j)}$ that act as copies of the state of robot i for each neighbor $j \in \mathcal{N}_i$. Consistency constraints are defined to ensure $\{\hat{\mathbf{x}}_i = \hat{\mathbf{w}}_i^{(j)} \mid j \in \mathcal{N}_i\}$. The objective function is augmented with quadratic constraint penalty terms to define a new optimization problem:

$$\begin{aligned} \min_{\hat{\mathbf{x}}, \hat{\mathbf{w}}} \quad & \|\hat{\mathbf{x}} + \bar{\mathbf{x}}\|_{2,1} + \frac{\rho}{2} \sum_{i \in \mathcal{V}} \sum_{l \in \mathcal{E}_i} \|\mathbf{c}_i^{(l)}(\hat{\mathbf{x}}, \hat{\mathbf{w}})\|_2^2 \\ & + \frac{\rho}{2} \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{N}_i} \|\mathbf{d}_i^{(j)}(\hat{\mathbf{x}}, \hat{\mathbf{w}})\|_2^2 \\ \text{s. t.} \quad & \{\mathbf{c}_i^{(l)}(\hat{\mathbf{x}}, \hat{\mathbf{w}}) = \mathbf{0} \mid l \in \mathcal{E}_i, i \in \mathcal{V}\} \\ & \{\mathbf{d}_i^{(j)}(\hat{\mathbf{x}}, \hat{\mathbf{w}}) = \mathbf{0} \mid j \in \mathcal{N}_i, i \in \mathcal{V}\} \end{aligned} \tag{7}$$

where $\rho \in \mathbb{R}^+$ is known as a penalty parameter [7], [15], $\mathbf{c}(\cdot)$ is the discrepancy between the expected and true range measurements, and $\mathbf{d}(\cdot)$ is the discrepancy between the local and neighbor reconstructed error vectors, defined as

$$\begin{aligned} \mathbf{c}_i^{(l)}(\hat{\mathbf{x}}, \hat{\mathbf{w}}) &= \mathbf{R}[l, i]\hat{\mathbf{x}}[i] - \left(\mathbf{z}[l] - \sum_{j \in \mathcal{N}_i} \mathbf{R}[l, j]\hat{\mathbf{w}}_j^{(i)}\right) \\ \mathbf{d}_i^{(j)}(\hat{\mathbf{x}}, \hat{\mathbf{w}}) &= \hat{\mathbf{x}}[i] - \hat{\mathbf{w}}_i^{(j)}. \end{aligned} \tag{8}$$

The choice of ρ is vital to ADMM's convergence and is discussed in Section IV-B. We define the dual variables $\boldsymbol{\lambda} = \{\boldsymbol{\lambda}_i^{(l)} \in \mathbb{R}^{m_l} \mid l \in \mathcal{E}_i, i \in \mathcal{V}\}$ and $\boldsymbol{\mu} = \{\boldsymbol{\mu}_i^{(j)} \in \mathbb{R}^{n_i} \mid j \in \mathcal{N}_i, i \in \mathcal{V}\}$ for the constraints of (7) to construct the Lagrangian of the constraint optimization problem in (9). For brevity, we omit the arguments of $\mathbf{c}(\cdot)$ and $\mathbf{d}(\cdot)$:

$$\begin{aligned} L(\hat{\mathbf{x}}, \hat{\mathbf{w}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) &= \sum_{i \in \mathcal{V}} L_i(\hat{\mathbf{x}}, \hat{\mathbf{w}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \\ L_i(\hat{\mathbf{x}}, \hat{\mathbf{w}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) &= \|\hat{\mathbf{x}}[i] + \bar{\mathbf{x}}[i]\|_2 \\ &\quad + \sum_{l \in \mathcal{E}_i} \left[\frac{\rho}{2}\|\mathbf{c}_i^{(l)}\|_2^2 + \left(\boldsymbol{\lambda}_i^{(l)}\right)^\top \mathbf{c}_i^{(l)}\right] \\ &\quad + \sum_{j \in \mathcal{N}_i} \left[\frac{\rho}{2}\|\mathbf{d}_i^{(j)}\|_2^2 + \left(\boldsymbol{\mu}_i^{(j)}\right)^\top \mathbf{d}_i^{(j)}\right]. \end{aligned} \tag{9}$$

Applying SCP and ADMM creates a nested loop structure that allows for iterative reconstruction of the error vector. Linearization of the measurement model constraint and updates to $\bar{\mathbf{x}}$ are done in the outer loop, shown in Algorithm 1. The inner loop is composed of the ADMM framework, which handles minimization over the primal variables $\hat{\mathbf{x}}$ and $\hat{\mathbf{w}}$, along with dual variable updates. This is shown in Algorithm 2. Note linearization occurs after N_{ADMM} iterations of the inner loop.

In many cases, it was found that retaining the values of dual variables $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ between iterations of the SCP loop could

Algorithm 1: Distributed Multi-Robot FDIR for Robot i.

Input: (MRS info.) ρ, N_{ADMM}, cold start thresholds
Input: (Robot info.) $\{\hat{\mathbf{y}}[l]\}_{l\in\mathcal{E}_i}$ and $\{\hat{\mathbf{p}}[j]\}_{j\in\mathcal{N}_i}$
1: Letting $\mathcal{N}_i' := \mathcal{N}_i \cup \{i\}$, initialize the following to $\mathbf{0}$: $\{\boldsymbol{\lambda}_i^{(l)} \in \mathbb{R}^{m_l} \mid l \in \mathcal{E}_i\}$, $\{\boldsymbol{\mu}_i^{(j)} \in \mathbb{R}^{n_i} \mid l \in \mathcal{N}_i\}$, $\{\mathbf{x}^*[j] \in \mathbb{R}^{n_j} \mid j \in \mathcal{N}_i'\}$
2: **while** fault monitor activated, **do**
3: Linearize the constraint by computing $\{\mathbf{z}[l] \mid l \in \mathcal{E}_i\}$ and $\{\mathbf{R}[l,j] \mid l \in \mathcal{E}_i, j \in \mathcal{N}_i\}$
4: Compute $\{\hat{\mathbf{x}}^*[j]\}_{\forall j\in\mathcal{N}_i'}$ using Algorithm 2.
5: Update the error vectors, $\forall j \in \mathcal{N}_i'$:

$$\bar{\mathbf{x}}[j]^+ \leftarrow \bar{\mathbf{x}}[j] + \hat{\mathbf{x}}^*[j]$$

6: Reset dual variables if cold start flag is set
7: **end while**
Output: Reconstructed error vector $(\hat{\mathbf{x}}[i] + \bar{\mathbf{x}}[i])$
Output: Robot integrity $\hat{\chi}_i$ from (10)

Algorithm 2: ADMM Subroutine for Robot i.

1: Initialize $\hat{\mathbf{w}}_i^{(j)^*} \in \mathbb{R}^{n_i}$ and $\hat{\mathbf{w}}_j^{(i)^*} \in \mathbb{R}^{n_j}$ as $\mathbf{0}$, $\forall j \in \mathcal{N}_i$.
2: **for** N_{ADMM} iterations, **do**
3: Solve first primal variable minimization:

$$\hat{\mathbf{x}}^*[i] = \arg\min_{\hat{\mathbf{x}}[i]} L_i(\hat{\mathbf{x}}, \hat{\mathbf{w}}^*, \boldsymbol{\lambda}, \boldsymbol{\mu})$$

4: Communicate $\hat{\mathbf{x}}^*[i]$ and $\boldsymbol{\mu}_i^{(j)}$ to neighbors $\mathcal{N}_i$.
5: Solve second primal variable minimization:

$$\{\hat{\mathbf{w}}_j^{(i)^*} \mid j \in \mathcal{N}_i\} = \arg\min_{\{\hat{\mathbf{w}}_j^{(i)} \mid j\in\mathcal{N}_i\}} L_i(\hat{\mathbf{x}}^*, \hat{\mathbf{w}}, \boldsymbol{\lambda}, \boldsymbol{\mu})$$

6: Communicate $\hat{\mathbf{w}}_j^{(i)^*}$ to neighbors $\mathcal{N}_i$.
7: Update the dual variables, $\forall j \in \mathcal{N}_i, l \in \mathcal{E}_i$:

$$\boldsymbol{\lambda}_i^{(l)^+} \leftarrow \boldsymbol{\lambda}_i^{(l)} + \rho \mathbf{c}_i^{(l)}(\hat{\mathbf{x}}^*, \hat{\mathbf{w}}^*)$$

$$\boldsymbol{\mu}_i^{(j)^+} \leftarrow \boldsymbol{\mu}_i^{(j)} + \rho \mathbf{d}_i^{(j)}(\hat{\mathbf{x}}^*, \hat{\mathbf{w}}^*)$$

8: Set cold start flag if any dual variable exceeds the predefined threshold
9: **end for**
Output: Solution to local linearized problem, $\{\hat{\mathbf{x}}^*[j]\}_{j\in\mathcal{N}_i'}$

promote convergence properties and is especially true for a static network topology. This technique, outlined in [7], [15], is known as warm start. However, when this network topology is time-varying, we find that warm start struggles to promote convergence. This is further discussed in Section IV-C.

Note that the global constants indicated in Algorithm 1 are only disseminated at the algorithm's start. Thus, after initialization, the error reconstruction process is decentralized since the minimization problems in Algorithm 2 are computed only over robot i's neighbors. It can also be claimed that the algorithm's computational and communication cost scale linearly with the size of the robot's neighborhood $|\mathcal{N}_i|$, not with the size of the MRS $|\mathcal{V}|$. This promotes the scalability of the algorithm to large robot swarms without affecting computational and communication efficiency, unless the swarm is densely packed.

B. Analysis on Noise Effect

The presence of noise was shown to be an issue with error reconstruction, occasionally resulting in divergence in initial numerical simulations. We use the penalty parameter ρ to mitigate this undesirable behavior. Noisy measurements directly affect the constraints $\mathbf{c}(\cdot)$ and $\mathbf{d}(\cdot)$ in (9). Thus, the penalty parameter ρ can control the effect of noise. That is, reducing ρ relaxes the constraints, thereby decreasing their effect on the objective function (7). This effectively acts as a filter for the noisy measurements. A similar approach to mitigating the effect of noise is theorized in [16]. We validate this approach with extensive numerical simulations and mixed reality experiments in Section V-A.

C. Analysis on Variations in Network Topology

Through numerical simulations, we found that the algorithm could exhibit unstable behavior when the network topology is time-varying. Attributed to the warm start method, this would result in highly inaccurate reconstructed errors for a time-varying network topology. It was found that time dependent true and/or estimated states with the warm start method could result in large variations in the problem constraints and dual variables between iterations. Even simple configuration changes could cause the algorithm's dual variables to grow unbounded, causing divergence or poor reconstruction of the error. This is an expected consequence of constant constraint violations with ADMM [7]. To solve this issue, we propose adding a check to bypass the warm start technique when the dual variables grow beyond a heuristically determined threshold. When this check, referred to as "cold start," is triggered, the dual variable values are reset to $\mathbf{0}$. This indicates the subsequent sub-problem is deemed sufficiently different enough to reset the problem parameters (i.e., the dual variables). The cold start method is validated in Section V-B. The dual variable resetting is handled on a per-robot basis, maintaining the distributed nature of the algorithm.

D. Integrity Monitor

With the reconstructed error computed, we define a measure for the integrity of each robot $i \in \mathcal{V}$ to be

$$\hat{\chi}_i = \|\hat{\mathbf{x}}[i] + \bar{\mathbf{x}}[i]\|_2. \tag{10}$$

Note that at each iteration of the inner loop, robot i communicates its integrity χ_i to a fault monitor. To account for noisy sensor measurements and transient behavior in the convergence of $\mathbf{x}$, we define a threshold value ε for robot integrity. When $\hat{\chi}_i > \varepsilon$, we declare that an attack or fault has been *identified* at robot i. The predefined threshold ε is determined heuristically from the values of $\nu_{\max}$ and $\omega_{\max}$ to balance the detection accuracy and false detection.

The integrity of robots $i \in \mathcal{V}$ are summed to measure the robot swarm's overall integrity: $\hat{\chi} = \|\hat{\mathbf{x}} + \bar{\mathbf{x}}\|_{2,1}$. With the same threshold ε, we declare the integrity monitor has *detected* an attack on or fault in the MRS when $\hat{\chi} > \varepsilon$.

While the integrity monitor is a centralized hub for aggregating the received robot integrity $\hat{\chi}_i$, it does not affect the fact that error reconstruction is a decentralized process. Additionally, since the integrity aggregation only uses simple operations, the computational overhead of the integrity monitor is low, even for large swarms. Additionally, note that accurate error

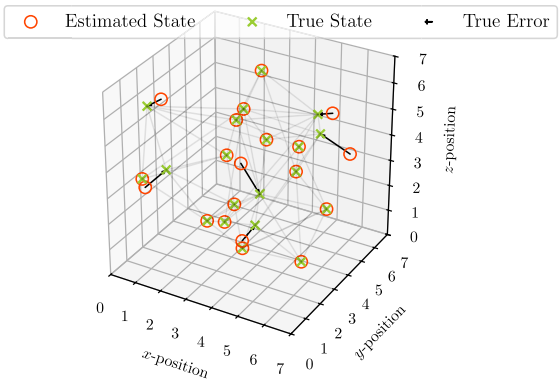


Fig. 3. Example swarm configuration showing true and estimated robot states with true error vector.

reconstruction may not be possible for *highly* dynamic MRS. However, this does not interfere with the error detection and identification goals of the integrity monitor as the robot integrity $\hat{\chi}_i$ will still grow and trigger the threshold.

V. Simulation Results

The proposed approach for multi-robot integrity monitoring is evaluated using numerical simulations and PX4-SiTL simulations on Gazebo. Through extensive Monte Carlo simulations, we validate the robustness of the proposed method to changes in the noise levels and time-varying network topologies. These issues are important as they occur during real-world deployment of an MRS but were not considered in the previous work [15]. For the numerical simulations, we consider a UAV swarm comprising 20 UAVs, out of which 14 are attack-free (healthy) and 6 were randomly selected to be attacked by an adversary using GNSS spoofing (see Fig. 3). To compare the results, we compute the difference between the reconstructed and true error vectors, denoted with $\mathbf{e}$, at each step in the simulation such that $\mathbf{e} = \mathbf{x} - (\hat{\mathbf{x}} + \bar{\mathbf{x}})$. A video of the overall description and performance of our algorithm in simulations is provided.[1]

A. Effect of Noise

In this section, we compare our algorithm's response to different constraint penalty parameters ρ and inter-robot measurement noise $\omega_{\max}$. To do so, we first conduct four simulations with the following parameter values and compare the error convergence properties: (i) $(\rho = 0.25, \omega_{\max} = 0.02)$, (ii) $(\rho = 0.25, \omega_{\max} = 0.05)$, (iii) $(\rho = 1.25, \omega_{\max} = 0.02)$, and (iv) $(\rho = 1.25, \omega_{\max} = 0.05)$. All tests are conducted with a maximum error in state estimation set at $\nu_{\max} = 0.02$. The root mean squared error (RMSE) $\|\mathbf{e}[i]\|_2$ is plotted in Fig. 4(a), with the darker lines indicating the average RMSE over the agents and the lighter shaded region representing 1 standard deviation bounds. We see that while parameter configurations (i) - (iii) exhibit similar convergence, simulation (iv), with high noise and ρ, failed to converge with the reconstructed error approaching infinity. This hints at a relationship between the noise level, parameter ρ, and convergence. For further investigation, we conducted Monte Carlo (MC) simulations with $\omega_{\max} = 0.02$ and $\omega_{\max} = 0.05$ along various values of ρ. Each simulation configuration was conducted for 100 trials. The RMSE was computed from each trial and averaged over the agents and trials. The results, plotted in Fig. 4(b), show convergence to the true error is diminished when higher values of ρ are selected with high noise levels. Thus, in a noisy system, we must seek to balance a high convergence rate (achieved with higher ρ values) with low divergence chances (achieved with lower ρ values). Similar results were achieved when replacing the uniform norm-bounded errors with unbounded Gaussian noise, as well as when testing with more than 6 attacked robots.

B. Effect of Variations in Network Topology

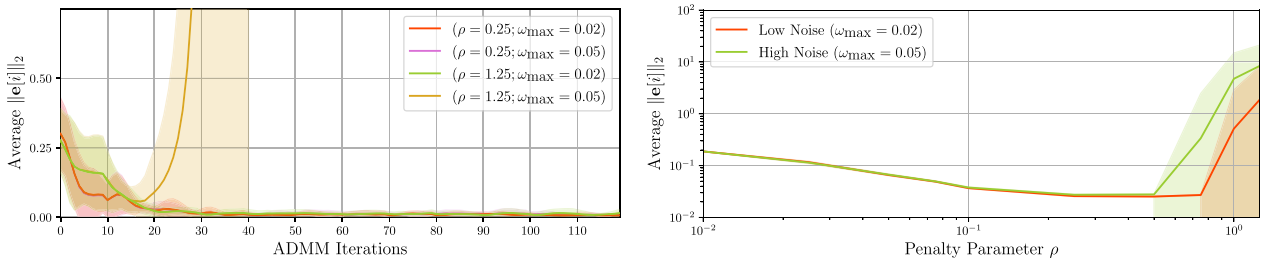
We consider a time-varying network topology induced by a 2-phase cyberattack, with only 3 robots being affected in each phase. Noise levels are set in all trials at $(\nu_{\max} = 0.02, \omega_{\max} = 0.02)$. The proposed algorithm's performance is compared under warm and cold start with $\rho = 0.25$ and $\rho = 0.75$. The RMSE $\|\mathbf{e}[i]\|_2$ is computed for each simulation and plotted in Fig. 5(a) in a similar fashion as Fig. 4(a). We see that 3 simulation configurations exhibit similar error reconstruction properties, while the simulation with $\rho = 0.75$ under warm start diverges after the second attack phase is engaged. The robustness of cold start under the changing network topology was further investigated with MC simulations at various values of ρ under warm and cold start, with 100 trials for each simulation configuration. The results are plotted in Fig. 5(b) and show that there isn't a significant disparity between warm and cold start for low ρ. This increases for large ρ, with cold start significantly outperforming warm start, demonstrating its increased robustness against time-varying network topology.

VI. Experimental Demonstration

A. Sensor Emulation for UWB Range Measurements

We leverage our Mixed Reality sensor emulation framework, i.e., Mixed-Sense [32], to generate inter-robot range measurements. The mixed-reality concept is, in general, vitally important for cybersecurity applications, where spoofing sensors and/or hacking communication channels could present logistical and legal challenges. In this framework, we create digital twins of real robots, tracked using motion capture cameras, inside Gazebo. Additionally, we spawn virtual simulated robots along with real robots in real-time. This allows us to create an augmented reality for the real robots, where these robots, although moving inside the controlled motion capture volume, get the sensory information from a simulated 3D environment. The sensor measurements from the simulator are transported to the real robots in real-time, allowing real robots to operate together with virtual robots inside the mixed reality environment according to their true dynamics. This feature enables us to validate the (i) scalability by spawning a large number of virtual robots alongside real robots, and (ii) interoperability by choosing various types of SiTL instances with the real robots. To generate the inter-robot range measurements, the pose of real robots tracked by the motion capture system is transported to Gazebo via ROS2. The pose of the virtual robots is retrieved from Gazebo's transport. A ROS node utilizes the poses of real and virtual robots to generate the inter-robot range measurement and publish it as a ROS topic.

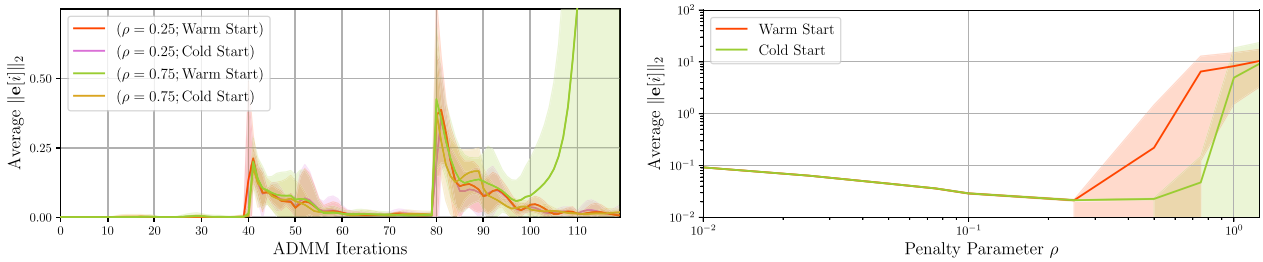
[1] Video link: https://youtu.be/CofOOgJyotQ



(a) Comparing trials with varying ρ and $\omega_{\max}$ values

(b) MC Simulations with varying ρ and $\omega_{\max}$ values

Fig. 4. Effect of noise on error reconstruction.



(a) Comparing trials at varying ρ with warm vs. cold start

(b) MC Simulations at varying ρ with warm vs. cold start

Fig. 5. Effect of cold start on error reconstruction.

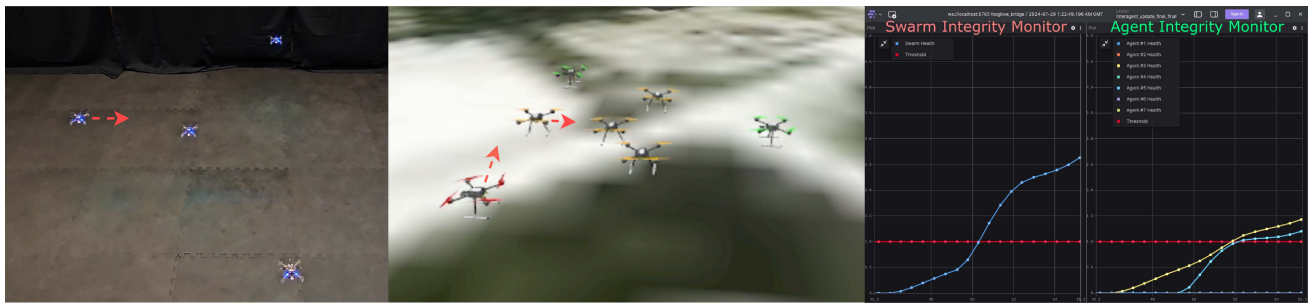


Fig. 6. Mixed reality experiments showcasing the effectiveness of proposed framework. **Left**: Real Crazyflie UAVs in motion capture environment **Center**: Real + Virtual UAVs in Gazebo **Right**: Integrity monitors for swarm and individual robots.

B. Mixed Reality Experiments

In our mixed reality setup, we use four real Crazyflies UAVs and spawn three PX4-SiTL instances. The real UAVs are tracked using the Qualisys Motion Capture system, which consists of 6 Oqus 7+ cameras. We consider the scenario where an adversary spoofs one real and one PX4-SiTL drone. The position of these drones drifts, affecting the safety of the overall swarm operation. The simulator hosting Gazebo is launched using a Dell Precision-3581 workstation laptop with a 13th-Gen Intel Core i7 processor and an Nvidia RTX A1000 GPU, where inter-robot range measurements are also emulated. We use Crazyflie 2.0 swarms as real UAVs for our demonstration. We use the open-source PX4 autopilot to instantiate virtual SiTL UAVs in Gazebo.

Under nominal operations, the swarm, comprising real and virtual UAVs, is commanded to first takeoff, hover, and transition into a static formation flight shown in Fig. 6. For the experimental demonstration, the integrity monitoring algorithm is implemented centrally, where all the information is aggregated at the ground station, i.e., the Dell Precision-3581 workstation laptop. The adversary then spoofs the GNSS sensor of two UAVs, one real and one virtual. We programmatically add a constant offset in the GNSS coordinates to alter the position measurements of both drones shown in Fig. 6. The integrity monitor for each vehicle detects an attack when the integrity measure crosses the threshold. The provided video also validates the algorithm using mixed reality experiments.[1]

VII. Conclusion

In this letter, we proposed a multi-robot integrity monitoring algorithm that utilizes locally sensed inter-robot range measurements to detect and identify the presence of rogue or faulty

robots in the system. We validated the robustness of the proposed algorithm to noise and time-varying network topologies through extensive numerical Monte Carlo simulations. Finally, we demonstrated the effectiveness of our algorithm through mixed-reality experiments on a heterogeneous robot swarm. In the future, we will demonstrate a distributed implementation of our algorithm in mixed-reality, allowing each agent to implement it independently using its locally observed information. We also plan to investigate the use of other inter-robot measurements, such as bearing and subtended angle, and integrate them with our algorithm to make it more robust against cyberattacks or faults.

ACKNOWLEDGMENT

The authors would like to thank the Dr. Shreekant (Ticky) Thakkar and his team members at the SSRC for their valuable comments and support.

REFERENCES

[1] D. Claes, F. Oliehoek, H. Baier, and K. Tuyls, "Decentralised online planning for multi-robot warehouse commissioning," in *Proc. 16th Int. Conf. Auton. Agents Multiagent Syst.*, 2017, pp. 492–500.

[2] J. Axelsson, "Safety in vehicle platooning: A systematic literature review," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1033–1045, May 2016.

[3] D. Jia, K. Lu, J. Wang, X. Zhang, and X. Shen, "A survey on platoon-based vehicular cyber-physical systems," *IEEE Commun. Surv. Tuts.*, vol. 18, no. 1, pp. 263–284, First Quarter 2015.

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

[5] P. Ghassemi and S. Chowdhury, "Multi-robot task allocation in disaster response: Addressing dynamic tasks with deadlines and robots with range and payload constraints," *Robot. Auton. Syst.*, vol. 147, 2022, Art. no. 103905.

[6] S. Zhao and D. Zelazo, "Bearing rigidity theory and its applications for control and estimation of network systems: Life beyond distance rigidity," *IEEE Control Syst. Mag.*, vol. 39, no. 2, pp. 66–83, Apr. 2019.

[7] S. Boyd et al., "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[8] M. Chadli, M. Davoodi, and N. Meskin, "Distributed state estimation, fault detection and isolation filter design for heterogeneous multi-agent linear parameter-varying systems," *IET Control Theory Appl.*, vol. 11, no. 2, pp. 254–262, 2017.

[9] Z. Gallehdari, N. Meskin, and K. Khorasani, "An H_∞ cooperative fault recovery control of multi-agent systems," *Automatica*, vol. 84, pp. 101–108, 2017.

[10] M. Guo, D. V. Dimarogonas, and K. H. Johansson, "Distributed real-time fault detection and isolation for cooperative multi-agent systems," in *Proc. Amer. Control Conf.*, 2012, pp. 5270–5275.

[11] K. Hashimoto, M. S. Chong, and D. V. Dimarogonas, "Distributed l_1-state-and-fault estimation for multiagent systems," *IEEE Trans. Control Netw. Syst.*, vol. 7, no. 2, pp. 699–710, Jun. 2020.

[12] Z.-H. Zhang and G.-H. Yang, "Distributed fault detection and isolation for multiagent systems: An interval observer approach," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 50, no. 6, pp. 2220–2230, Jun. 2020.

[13] T. K. Tasooji and H. J. Marquez, "Cooperative localization in mobile robots using event-triggered mechanism: Theory and experiments," *IEEE Trans. Automat. Sci. Eng.*, vol. 19, no. 4, pp. 3246–3258, Oct. 2022.

[14] S. Khodadadi, T. K. Tasooji, and H. J. Marquez, "Observer-based secure control for vehicular platooning under DoS attacks," *IEEE Access*, vol. 11, pp. 20542–20552, 2023.

[15] S. Khan and I. Hwang, "Distributed error-identification and correction using block-sparse optimization," 2024, *arXiv:2309.11784*.

[16] S. Khan and I. Hwang, "Recovery of localization errors in sensor networks using inter-agent measurements," 2023, *arXiv:2307.12078*.

[17] Y. Tan, C. Hu, K. Zhang, K. Zheng, E. A. Davis, and J. S. Park, "LSTM-based anomaly detection for non-linear dynamical system," *IEEE Access*, vol. 8, pp. 103301–103308, 2020.

[18] W. Hao, T. Yang, and Q. Yang, "Hybrid statistical-machine learning for real-time anomaly detection in industrial cyber–physical systems," *IEEE Trans. Automat. Sci. Eng.*, vol. 20, no. 1, pp. 32–46, Jan. 2023.

[19] R. Bhatia, S. Benno, J. Esteban, T. Lakshman, and J. Grogan, "Unsupervised machine learning for network-centric anomaly detection in IoT," in *Proc. 3rd ACM Conext Workshop Big Data, Mach. Learn. Artif. Intell. Data Commun. Netw.*, 2019, pp. 42–48.

[20] P. Lin, K. Ye, and C.-Z. Xu, "Dynamic network anomaly detection system by using deep learning techniques," in *Cloud Computing*, Berlin, Germany: Springer, 2019, pp. 161–176.

[21] J. Goh, S. Adepu, M. Tan, and Z. S. Lee, "Anomaly detection in cyber physical systems using recurrent neural networks," in *Proc. IEEE 18th Int. Symp. High Assurance Syst. Eng.*, 2017, pp. 140–145.

[22] L. Shen, Z. Li, and J. Kwok, "Timeseries anomaly detection using temporal hierarchical one-class network," in *Proc. 34th Int. Conf. . Neural Inf. Process. Syst.*, 2020, pp. 13016–13026.

[23] H. Lau, I. Bate, P. Cairns, and J. Timmis, "Adaptive data-driven error detection in swarm robotics with statistical classifiers," *Robot. Auton. Syst.*, vol. 59, no. 12, pp. 1021–1035, 2011.

[24] D. Tarapore, J. Timmis, and A. L. Christensen, "Fault detection in a swarm of physical robots based on behavioral outlier detection," *Trans. Robot.*, vol. 35, no. 6, pp. 1516–1522, 2019.

[25] A. Suarez, G. Heredia, and A. Ollero, "Cooperative sensor fault recovery in multi-UAV systems," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 1188–1193.

[26] V. Sindhwani, H. Sidahmed, K. Choromanski, and B. Jones, "Unsupervised anomaly detection for self-flying delivery drones," in *Proc. 2020 IEEE Int. Conf. Robot. Automat.*, 2020, pp. 186–192.

[27] S. Lee, E. Milner, and S. Hauert, "A data-driven method for metric extraction to detect faults in robot swarms," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 10746–10753, Oct. 2022.

[28] F. Arrichiello, A. Marino, and F. Pierri, "Distributed fault detection and recovery for networked robots," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2014, pp. 3734–3739.

[29] M. D. Kutzer, M. Armand, D. H. Scheid, E. Lin, and G. S. Chirikjian, "Toward cooperative team-diagnosis in multi-robot systems," *Int. J. Robot. Res.*, vol. 27, no. 9, pp. 1069–1090, 2008.

[30] S. Hwang, M. Cho, S. Kim, and I. Hwang, "An LMI-based risk assessment of leader–follower multi-agent system under stealthy cyberattacks," *IEEE Control Syst. Lett.*, vol. 7, pp. 2419–2424, 2023.

[31] M. Cho, S. Hwang, and I. Hwang, "Risk assessment of multi-agent system under denial-of-service cyberattacks using reachable set synthesis," in *Proc. IEEE 2024 Amer. Control Conf.*, 2024, pp. 1293–1298.

[32] K. A. Pant, L.-Y. Lin, J. Kim, W. Sribunma, J. M. Goppert, and I. Hwang, "Mixed-sense: A mixed reality sensor emulation framework for test and evaluation of UAVs against false data injection attacks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024, pp. 12414–12419.