



Increasing Operational Resiliency of UAV Swarms: An Agent-Focused Search and Rescue Framework

Abhishek Phadke^{1,2*} and F. Antonio Medrano^{1,2}

¹Conrad Blucher Institute for Surveying and Science, Texas A&M University-Corpus Christi, Corpus Christi, TX, United States,

²Department of Computer Science, Texas A&M University-Corpus Christi, Corpus Christi, TX, United States

Resilient UAV (Unmanned Aerial Vehicle) swarm operations are a complex research topic where the dynamic environments in which they work significantly increase the chance of systemic failure due to disruptions. Most existing SAR (Search and Rescue) frameworks for UAV swarms are application-specific, focusing on rescuing external non-swarm agents, but if an agent in the swarm is lost, there is inadequate research to account for the resiliency of the UAV swarm itself. This study describes the design and deployment of a Swarm Specific SAR (SS-SAR) framework focused on UAV swarm agents. This framework functions as a resilient mechanism by locating and attempting to reconnect communications with lost UAV swarm agents. The developed framework was assessed over a series of performance tests and environments, both real-world hardware and simulation experiments. Experimental results showed successful recovery rates in the range of 40%–60% of all total flights conducted, indicating that UAV swarms can be made more resilient by including methods to recover distressed agents. Decision-based modular frameworks such as the one proposed here lay the groundwork for future development in attempts to consider the swarm agents in the search and rescue process.

Keywords: resilience, UAV, swarm, search and rescue, UAS

INTRODUCTION

The use of UAV swarms is becoming more widespread due to the reduced costs of UAVs and their ability to accomplish tasks more quickly and effectively as a group rather than individually. Advancements in aircraft design and control, communication topologies, and battery systems have made coordinated UAV swarms possible. The use of UAV swarms in applications of military [1, 2], ecology [3], remote sensing [4, 5], disaster management [6], crowd control, emergency communication [7], agriculture [8, 9], and victim search [10, 11] are just some of the use cases. As individual and multi-robotic development and their interaction with real-world entities advances, these applications are only limited because of a lack of improvement, a discrepancy that exponentially decreases as time passes. With the increased diversity of swarm usage

OPEN ACCESS

*Correspondence

Abhishek Phadke,

✉ aphadke@islander.tamucc.edu

Received: 15 November 2023

Accepted: 13 December 2023

Published: 04 January 2024

Citation:

Phadke A and Medrano FA (2024)
Increasing Operational Resiliency of
UAV Swarms: An Agent-Focused
Search and Rescue Framework.
Aerosp. Res. Commun. 1:12420.
doi: 10.3389/arc.2023.12420

Abbreviations: AS, Application specific; FANET, Flying Ad Hoc Networks; GNSS, Global Navigation Satellite System; GCP, Ground control points; HBS, Heartbeat signal; IDS, Intrusion Detection System; MANET, Mobile Ad Hoc Networks; MPC, Model Predictive Control; MRS, Multi-Robot System; OLSR, Optimized Link State Routing; PSO, Particle Swarm Optimization; POV, Point of View; ROI, Region Of Interest; S.S., Swarm Specific; SSI, Signal Strength Indicator; SAR, Swarm Agent Specific; SAR, Search and Rescue; SMF, Swarm Methodology Focused; UAV, Unmanned Aerial Vehicle; UGV, Unmanned Ground Vehicle; UWSV, Unmanned Water Surface Vehicle; VPS, Visual positioning system; VRRZ, Variable Radius Risk Zones.

applications, research in UAV resiliency has also grown [12, 13]. Due to the close-knit topology of these swarms, the failure of agents above a certain threshold can often lead to cascaded systemic collapse and a pause on mission progress. The cause of this failure may be structural, such as in a leader-follower topology, wherein the followers may get disconnected if the leader fails. Additional uncertainty in this failure also exists, such as the possibility of a failing swarm agent crashing into other agents during its collapse. Resiliency is defined as the ability of a system to withstand disruptions. Broader definitions include the ability of a system to bounce back after a disruption [14]. D.D. Woods summarizes system resilience perfectly in its four core concepts [15]. They are resilience, such as rebound, robustness, extensibility, and adaptability. Previous work by the authors addresses systemic resilience in UAV swarms on a broader range by classifying UAV swarm operations into components and modules [16]. The resilience of UAV swarms is a complex topic that integrates multiple components of navigation [17], mapping [4], control [18], defensive and intrusion detection policies [19], agent welfare, and physical characteristics of the swarm agent [20] into an intricate system designed to create balance in a dynamic environment.

An MRS (Multi-Robot System) and a swarm are both concepts in robotics that involve the coordination and interaction of multiple robots. However, they have distinct differences in organization, control, and behavior. An MRS has a structured and explicit interaction scheme with a centralized control. Swarm agents, on the other hand, are more decentralized and self-organized. There might not be explicit communication between robots, and collective behavior emerges from simple interactions between individual robots following local rules.

Additionally, MRS may involve centralized controllers and planners to assign tasks to robots. Swarms rely on local interactions and distributed control. Each robot can typically make decisions based on its immediate surroundings or information gathered from nearby agents. This also involves a degree of autonomy in decision-making from completely autonomous to semi-autonomous. However, while swarms are expected to be inherently scalable, there is an ongoing debate on the minimum number of agents that must be present and acting collectively to label it as a swarm. Adding more robots to a swarm does not necessarily increase performance. Selecting the number of agents in a swarm has long been contested. However, approaches with agents as few as five and as many as 1,000 have been implemented and studied. Article [21] discusses how aspects such as system scalability, technical capabilities of individual agents, and financial or logical constraints influence the selection of the number of agents in the swarm.

These factors were crucial in selecting the number of agents for swarm response experiments performed in this study. The number of agents available for experimentation was limited. Additionally, some agents were designated as reserve and spare agents to ensure experiments continued in case of equipment failure. Space constraints allowed only a certain number of agents to fly in the designated airspace without the

risk of agent collision and crashes due to induced airflow interactions. While all communication between agents was performed decentralized, primary communication and network protocols required for communication with GCS limited the number of agents connected to them.

Search and Rescue is a vast domain; focusing only on swarm agent welfare significantly narrows it. However, to concentrate results further, this SAR framework will be primarily described for exploratory swarm applications. Scenarios where a swarm of agents may be deployed over an area and, in the process, may lose contact with the swarm is the priority. This narrows down the framework focus as well as experiment design and validation. Two major types of SAR capabilities in UAV swarms are defined and categorized here. They are application-specific SAR [22] and swarm-specific SAR (SS-SAR). Although our study takes a different direction than a regular application-focused SAR use scenario, it remains an exploratory problem. The tracking, location, and Rescue of disabled swarm agents require other agents of the swarm to actively search the target space for the agent using techniques such as triangulated localization, computer vision, sensors on the ground, and the analysis of system-generated mission logs. A literature review reveals that most swarms lack the self-awareness needed to actively take care of their agents. More robust mechanisms for the welfare of UAV swarms [23] are needed as an additional means to increase systemic resilience.

To build robust applications and routine case scenarios that use UAV swarms, the swarm itself must be resilient to disruptions. Towards this goal concerning SAR swarm operations, the significant contributions of this paper are:

- A literature analysis that reveals a research gap in UAV swarm development related to the search and rescue of their agents.
- To address this gap, a novel UAV swarm framework, SS-SAR (Swarm Specific-SAR), is introduced to provide the ability to track, locate, and possibly rescue their agents. The framework uses a decentralized approach and local communication between neighboring agents and surrounding data to make semi-autonomous deployments of rescue craft that initiate direct communication with distressed agents.
- Experimental results show the SS-SAR framework's ability to reduce agent loss in swarm operations.
- Future framework upgrades and experiment designs are proposed to increase operational swarm resilience.

Using a decentralized approach for communication and agent decisions, this study aims to demonstrate scalable and robust responses of swarms to disruptive scenarios along with further scope for possible emergent behavior to avoid them altogether based on broad programmed constraints. The paper is arranged in the following way. Section *Introduction* gives a brief introduction of the area with research contributions of this study. Section *Summary of Recent Literature on Multi-Robot SAR* presents a categorization of current trends. Section *Swarm-Specific SAR Framework* presents the SS-SAR scenario

TABLE 1 | Summary of recent work on SAR using multi-agent UAV swarms categorized by approach.

Reference	Category	Description
[24]	AS	Using a modified fruit fly algorithm to improve the search efficiency of a multi-robot swarm
[25]	AS	Cooperative strategy for distributed UAV agents in a swarm performing unique functions for victim search and rescue operations
[26]	AS	Smart search for survivors using a genetic localization method to detect victim distress signals using autonomous maximum area searching UAV agents
[27]	AS	Collaboration between swarm agents for detecting victim presence
[28]	AS	Layered SAR based on disaster epicenter for improved victim detection using multiple agents
[29]	AS	Heterogeneous agent swarm based on ant colony optimization and agent decision process for victim searching at sea
[30]	AS	An open-source platform for managing drones for assistance in SAR operations
[31]	SMF	Using deep reinforcement learning to generate control commands for UAVs to search in an environment with an unknown number of targets
[32]	SMF	A dynamically varying number of swarm agents search for the target using MPC for generating cooperative search trajectories and maximizing performance
[33]	SMF	Creating target probability maps to guide swarm search actions based on flocking, velocity, and area coverage
[34]	SMF	Collaborative search function based on pigeon-inspired bio-inspired algorithm
[35]	SMF	Hexagonal grid decomposition of the search area for maximum efficiency during target search in a maritime rescue scenario
[36]	SMF	Planning using a Markov decision process and control using environmental exploration by deep learning for target detection
[37]	SMF	A bio-inspired algorithm based on fish schooling and foraging behaviors for improving target search functionality
[38]	SMF	A reinforcement learning-based concept to make a territory awareness map for generating cooperative search paths for multi-UAV swarms
[39]	SMF	A profit-driven adaptive search algorithm for moving targets using a UAV swarm capable of information exchange
[40]	SMF	PSO-MPC approach to solving and improving the efficiency of the SAR technique using multiple agents rather than a single agent
[41]	SS-SAR	A swarm-specific methodology for automatic replacement of any lost UAV during mission progress

description and framework workflow. Section *Performance Tests* describes both hardware and simulation performance tests and environmental parameters. Section *Results* presents the experimental results of the new SS-SAR framework. Section *Discussion and Future Research Directions* provides future directions to approach the problem with suggested framework upgrades, and Section *Conclusion* provides concluding statements.

SUMMARY OF RECENT LITERATURE ON MULTI-ROBOT SAR

The multi-UAV SAR problem is a broad problem domain. This section categorizes current research into three distinct approaches. **Table 1** identifies research on the topic and categorizes the study as application or swarm-specific problems. Application-specific SAR (AS-SAR) and SS-SAR are the two categorizations previously discussed. Depending on how the SAR problem is approached, a third category is also included: Search Methodology Focused (SMF). Research in this category does not have a specific search target type; instead, the focus is on the general SAR methodology, where any internal or external search target can be assigned.

As one can see from **Table 1**, swarm-specific implementations are less explored in the literature. In addition to the above literature review, generalized methodologies exist that propose novel approaches that would improve facets of the SAR process. These include using bio-inspired algorithms for area coverage [42] formation tracking [43] and environment exploration [44], updated and merged observation maps or information exchange pathways [45], and efficient task planning [46, 47]. Frameworks

such as [41] that propose automatic replacement of lost UAV agents are scarce. This example fits perfectly in this paper's proposed swarm-specific research category. To keep the literature analysis attainable, any approaches that do not directly describe the use of aerial vehicle swarms in the field for SAR have been eliminated. This includes broader research topics such as using machine learning methods to improve object recognition in aerial images taken by UAVs [48].

SWARM-SPECIFIC SAR FRAMEWORK

Workflow Description

Notations used in framework description and development are summarized in **Table 2**. This section briefly describes the broad workflow for the framework design process.

The SS-SAR workflow [49] is summarized in **Figure 1**. It is divided into four sections, with the first section defining the agent tracking phase, the second section containing the initial decision, section three having the primary decision process, and section four with the secondary decision process. The modular framework design assists in the testing and modification of one or more sections. This was especially useful in scenarios where the hardware and software test platforms could not simultaneously handle all the framework tests computationally or physically. For example, low-cost agents such as the DJI Telos [50] used in the lab scenarios required testing individual sections piecewise rather than the entire framework simultaneously due to inefficient hardware and lack of sensors. The experiment section describes the modular experiments designed to test the workflow to the extent that the agents could handle it.

TABLE 2 | Notations.

n	Total number of agents in the swarm
i	Index of UAV agents (from 1 to n)
t	Equally spaced time interval between HBS signals
k	Index of HBS time sample
$HBS_{i,k}$	HBS signal from agent i at time k
Ind	Binary variable to denote the presence or absence of HBS signal
$Ind_{i,k}$	Binary value for HBS signal from agent i at time k
Ind_{all}	Binary variable based on an AND logical operation of all binary indicator variables
$HBS_{i,k} \rightarrow loc$	Location of agent i at time k (included in the corresponding HBS signal)
$HBS_{i,k} \rightarrow b_{level}$	Battery level of agent i at time k (included in the corresponding HBS signal)
$SSI_{(GC, i)}$	Signal strength indication of ground control to agent i
OG	On-ground indicator that is set to 1 if an agent is actively connected but is on the ground
p_0	Real-time pose check using the distressed agent camera
p_1	Real-time pose check using the rescue agent camera
p_2	Real-time pose check using IMU
tk_{id}	Task ID
tk_{cost}	Cost of completing a task
$R_{loc(i,k-1)}$	Denotes the rescue agent moving to the location of distressed agent i at $k-1$ time
rcm	Reconnection message

allows it. The indicator tk_{cost} is used to determine the cost of completing the task that is estimated using the number of time intervals required t , and the expected change in battery level to complete the task, Δb_{level} .

Scenario Description

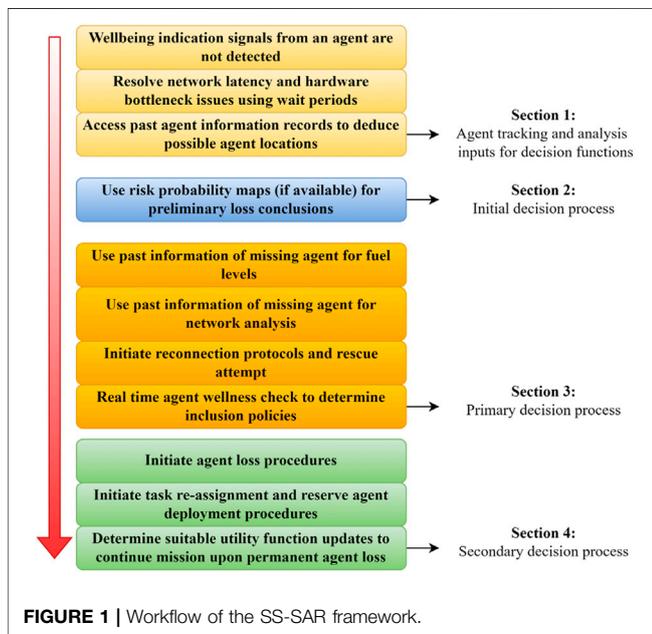
The generalized model in **Figure 2** was expanded into a specific scenario where a swarm of agents is performing a task, and one of the agents is in distress. This SS-SAR process is depicted in **Figure 2**.

The OLSR (Optimized Link State Routing) protocol has been extensively studied as an ideal routing protocol in SAR environments [51]. It routinely uses “Hello” and “Topology Control” messages to identify links and agent states. The heartbeat signal is often referred to as a modified hello message based on the base OLSR protocol. The heartbeat signal transmission is a small, quick transmission objective signal that each swarm agent can send at regular intervals. Various alternate implementations exist [26, 41, 52]; however, they follow a general structure that includes information denoting network I.D., transmitting agent ID, destination I.D., message type, security I.D., data segments, and error check. The HBS comprises location information, the battery level of the agent, signal strength indication, and the current task I.D.

$$HBS_{i,k} = [Loc_{i,k} + b_{level,i,k} + SSI_{(GC,i)} + tk_{id}]$$

A fixed number of agents, n , form the swarm. The HBS from every agent is expected after each time interval t , at a sample iteration denoted by k . An HBS is expected to be transmitted by every agent in the swarm after the time interval value of t . The signal is denoted by $HBS_{i,k}$. This denotes the signal transmission from the i^{th} agent at the k^{th} time interval, in the range of $i = 1$ to n , and $k > 0$. The Ind binary variable indicates the presence or absence of the HBS signal for every $Ind_{i,k}$. The value of 1 is recorded for every signal received and 0 if a signal is missing. The missing HBS agent id value determines which agent did not send the signal. The Ind_{all} is a logical operator set to 1 if all agents send a signal and 0 if HBS from an agent is missing.

Time intervals t are regular spaced and defined for transmitting the HBS. Careful consideration of this assigned value is required. A higher value of t can cause fewer HBS to be transmitted during mission time, i.e., a greater amount of time can elapse between a missing HBS and the system realization of an agent in distress. However, a lesser value of t can cause network bottlenecks if the system cannot receive and process HBS from all agents of the swarm. **Figure 3** shows HBS signal and sensor data transmission over a regular and disrupted time series. A longer period of HBS transmission intervals may result in delays between agent loss and system realization, $2t$. The disrupted time series shows the information delay for sensor data access of an agent by the operator. Since sensor data is sent at less frequent intervals than the HBS, the operator has access to information that may not give an exact interpretation of agent distress if the disruption occurs after a significant time interval after the last sensor data transmission.

**FIGURE 1** | Workflow of the SS-SAR framework.

The second advantage of the modular nature is that framework components can be upgraded, optimized, or changed. For example, while preliminary experiments for *Section Performance Tests* use an essential task re-assignment policy where only idle agents are given the tasks previously assigned to the lost agent, future iterations of the framework can use an optimized cost consideration, where characteristics of the task-receiving agent, such as its remaining fuel, are considered before re-assignment. An agent completing its task is only assigned the task of the fallen agent if its battery capacity

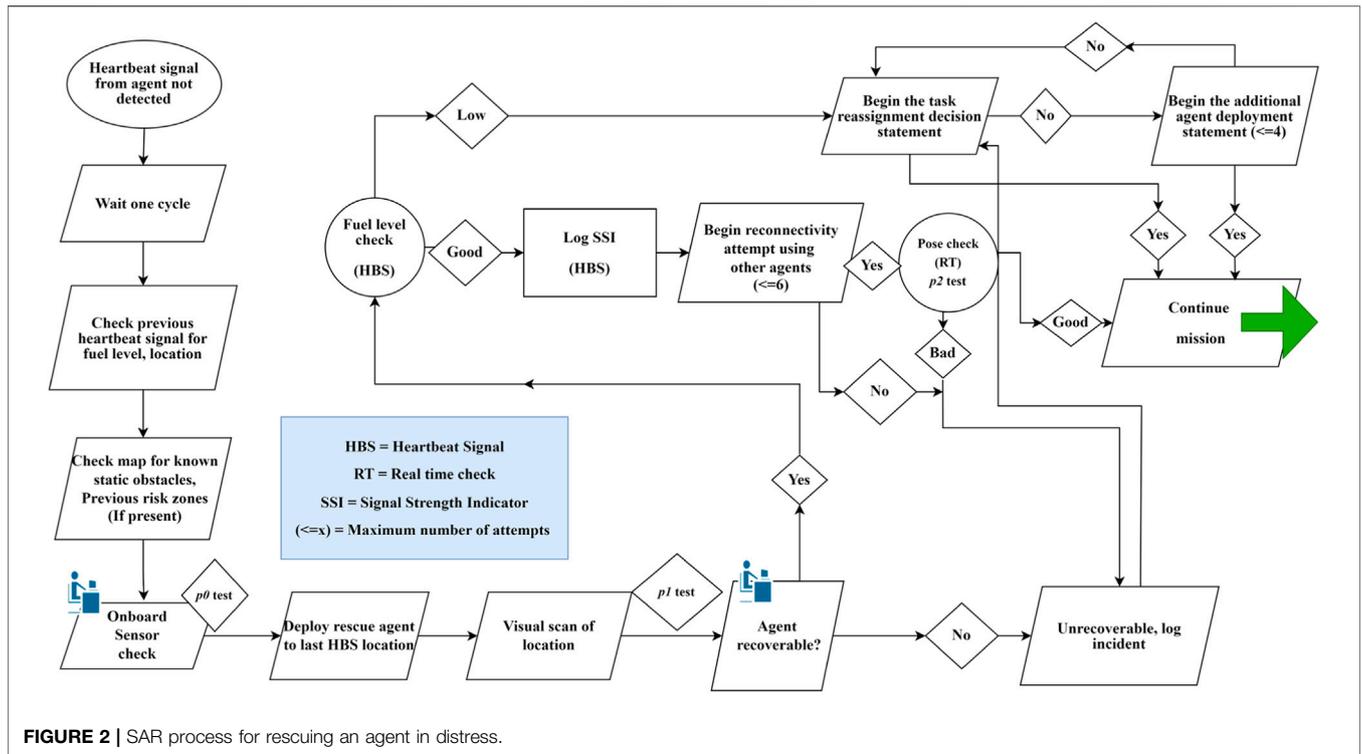


FIGURE 2 | SAR process for rescuing an agent in distress.

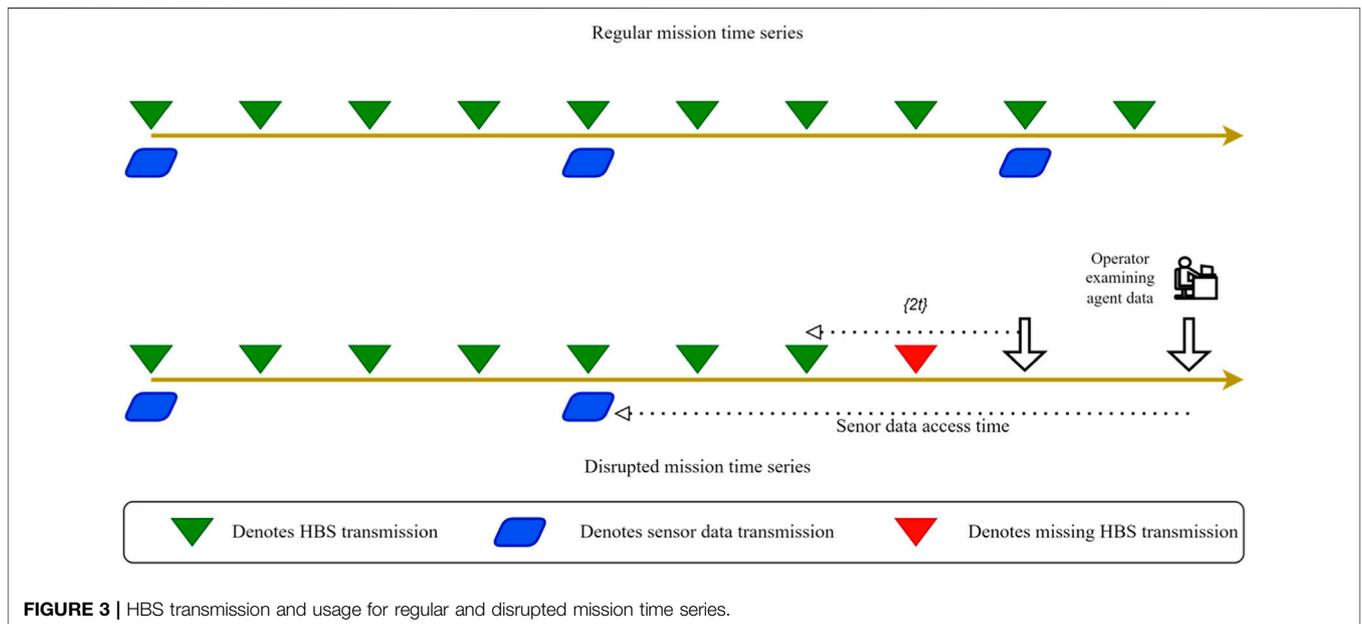
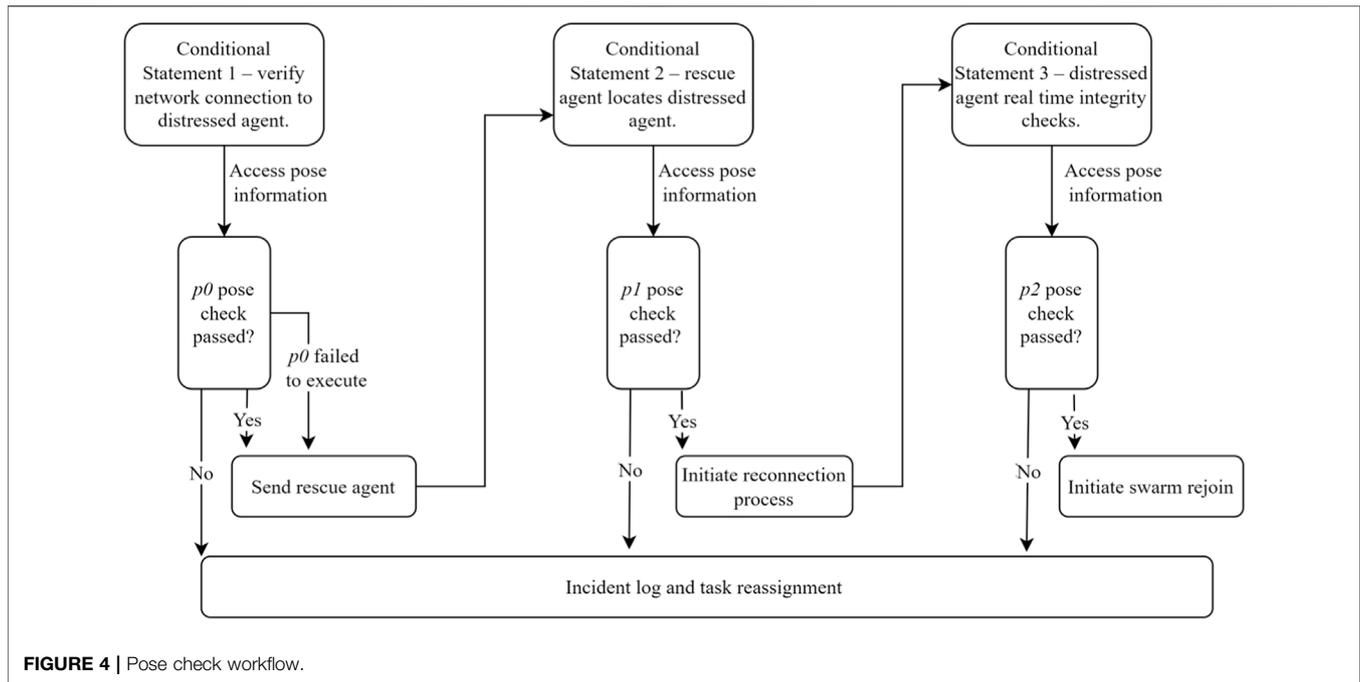


FIGURE 3 | HBS transmission and usage for regular and disrupted mission time series.

If an agent is missing, its past HBS record is retrieved and examined for its location during that transmission time interval. This location information may be outdated by a minimum value of the system realization time, that is, $2t$. A map overlay for known static obstacles is then used to determine if the agent was near obstacles during loss. A UAV agent can be distressed due to reasons such as collision with a static or dynamic obstacle, falling

out of range from other agents in a mesh-based topology or with ground control in a directed topology, or issues with hardware components and fuel. Multiple pose checks are designed in the framework and conducted at each step to systematically eliminate the cause of disruptions. It is assumed that the agent, even when on the ground, has an open broadcast connection request to accept incoming connection requests from other agents or



manual remote-control overrides. This feature is always present, even in basic agent builds. An agent that is connected to a network but has landed due to a collision, flock fragmentation, or getting stuck in an obstacle can still transmit an HBS with an on-ground indicator value of $O.G. = 1$. This indicates that an agent has landed but is still actively connected. However, this agent cannot be directly accepted back into the swarm. It is necessary to ensure that it can safely rejoin the swarm by gaining the necessary minimum altitude required by the agent to take a safe flight.

At specific decision points in the framework, agent status checks called pose checks are performed to gain additional information about the agent. The framework can perform three different checks: p_0 , p_1 , and p_2 . Pose check flow conducted at different times during framework operation is shown in **Figure 4**.

Once an agent is realized to be in distress, an initial attempt is made to see if it is still possible to access its onboard vision sensor to conduct a preliminary pose check p_0 . This checks if the agent has landed in such a position that it may be able to take off safely. Examples of passed and failed p_0 tests are shown in *Section Performance Tests*. The advantage of this method is that if the preliminary pose check fails, the framework can skip sending the rescue agent and directly move on to the unrecoverable agent process. However, this step is flexible: a rescue agent can still be deployed if the p_0 check cannot be conducted.

If p_0 passes, rescue agent R moves to the location of the missing agent ($R_{loc(i, k-1)}$) and performs a visual scan of the location. The operator conducts real-time viewing of the rescue agent's camera data to conduct p_1 . After an agent is located, the p_1 check using a rescue agent vision sensor is done to assess if the agent is in an environment from which it can take off safely. A fuel check using

b_{level} and a network connection check using $SSI_{(G.C., i)}$ are then performed. The SSI value contains agent connection data with ground control and neighboring agents. Depending on the network topology selected, an $SSI_{(G.C., i)}$ value of 0 can be acceptable if the distressed agent connects to another agent rather than to ground control.

$$p_2 = (b_{level} > 40\%) \text{ AND } (SSI_{(G.C., i)} > 80\%)$$

This data is taken from the agent's previous HBS to create record logs of why the agent failed. This information is used to create risk zones as an information overlay in mission maps, a framework feature designed to reduce the failure of future agent movements on the same map.

If p_1 passes, the rcm messages are sent to reconnect with the agent. Once an agent is actively connected, a real-time pose check p_2 is conducted, which checks the current agent fuel level and network connection. In higher-level agents, this check can also take feedback from individual components onboard the vehicle, such as autopilot and motor sensors, to check for hardware integrity and orientation. If this pose check is passed, the distressed agent is deemed capable of rejoining the swarm. If the pose check fails, a log is created, and the agent's location is marked with an overlay that denotes the perceived reason for failure. A task re-assignment policy is then initiated to reassign the task of the lost agent to other swarm agents.

To date, probability maps have been a prevalent approach in SAR problems. Global or local maps are proposed that decompose ROI in grids [40], and a probability rate of the target being in each of the cells is calculated. Agents are encouraged to explore cells with a higher probability rating of the target being present in them. Similar approaches have been examined in [31] where agents not only create and maintain

TABLE 3 | Test observations and map used for the four performance tests.

Performance test (P.T.)	Test objectives	Map used	Number of flights
PT1	Observe time to distress, time to rescue, collision occurrence, and battery level check	M1	15
PT2	Observe p_1 and p_2 , recovery, and operator log creation	M2	10
PT3	Observe p_0 , p_1 , p_2 , recovery, and operator log creation	M3	10
PT4	Observe p_0 , p_1 , and p_2 , time to distress, time to rescue, collision occurrence, and operator log creation	M4	10

observation map history, but maps from neighboring agents can also be combined. A similar logic is used in this case, where ground control creates and maintains a global risk map where each cell has an associated risk value. This is based on location data of previous agent loss, where an incident log is created every time an agent is lost in a particular area in the same map. This is especially useful in same-area routine flight scenarios where UAVs must visit the same area multiple times. Labeled hotspots can then be used as additional input constraints to path-generating algorithms by assigning proportional weights to high-risk zones, which the planning algorithms can then avoid or have issue mitigation resources ready if those areas are unavoidable.

Pose and orientation calculation can be upgraded with optimization loops coming from additional input sources. For example, the vision sensor data of the distressed UAV can be accessed, and an automatic pose orientation of the UAV can eliminate the need to dispatch a rescue UAV if the fallen UAV sends an unrecoverable camera pose. This was demonstrated during various experiments in which a human in the loop could access the sensor information of the distressed UAV to deduce its orientation. If determined to be unrecoverable, the agent's location is marked for post-mission recovery trials, and the swarm moves on directly to the task re-allocation phase of the fallen agent.

PERFORMANCE TESTS

Hardware and software tests were designed to test the proposed workflow under different conditions. Experiment range and series were selected considering the range and variability required to effectively demonstrate performance [53]. **Table 3** summarizes the primary objectives of each test, the map used, and the number of experimental flights performed. Overall, these tests represent a modular approach to developing and testing an SS-SAR framework for increasing the operational resiliency of a UAV swarm system.

Each performance test was associated with a map, as summarized in **Table 3**. The hardware tests were performed in maps M1 and M2, and the simulation tests were performed in maps M3 and M4. **Table 4** outlines characteristics of the map environments used in the performance tests.

The proposed framework is quite flexible regarding the agents that can be used. However, at minimum, lateral and downward vision sensors are required, along with either a GNSS module or capability for passive beacon georeferencing. Considering hardware and fly space limitations, an indoor location was

used for hardware tests. The DJI EDU [50] UAV platform was chosen to perform hardware performance tests. These low-cost drones provide a basic environment for drone testing and flights. In the past, there have been multiple approaches to using Tello drones as platforms for singular and swarm development. The authors of [54] use Tello agents to demonstrate an automated swarm flight in a restricted flight space. A matrix formation control that uses Tello to display patterns was adopted in [55]. In [56], the DJI frame was used to build visionless sensing drones for obstacle avoidance and maze solving.

Related research such as this assisted in realizing the various limitations of the Tello platform during experiment design. The Tello agents are low-cost entry-level hardware and are intended for proof-of-concept experiments. Without a dedicated GNSS receiver, the agents rely on a VPS using the downward-facing camera module to localize using ground planes and additional GCPs. All recorded video and image data is streamed in real-time to ground control without storage and post-processing ability. While these constraints prevent executing a full-scale framework representation on these agents, our experiments modify the complete framework based on its modular structure. This modular and stepwise process permits testing smaller decision statements using simple Tello agents. **Table 5** contains manufacturer-provided specifications for the DJI drones. These specifications have been referenced from online user manuals [50].

Hardware Performance Tests

A modular and stepwise process was developed to test individual decision statements of the proposed SAR framework using simple Tello EDU agents. The objective of PT1 was to evaluate the time to distress, time to rescue, log any collision occurrence, and perform a battery level check. In PT1, two DJI agents were used as a part of the same swarm, with a rescue agent on active standby. A 2D visualization of M1 with initial agent positions and other mission information is shown in **Figure 5**.

One of the agents moving along the mission pads was forced to switch off its VPS to emulate a disruption condition. Meanwhile, the rescue agent was on active standby in the center of the mission area (**Figure 6**, left) and could take off once the distressed agent did not send an expected HBS (**Figure 6**, right). Using mission pad information transmitted by the distressed agent before it faced disruption, the rescue agent located the fallen agent (**Figure 7**, left), conducted pose checks, and sent *rcm* messages to the distressed agent. If the distressed agent received the messages, it switched on its VPS, allowing it to rejoin the swarm. The rescue agent then moved back to its deployment point to await the next distress event (**Figure 7**, right).

TABLE 4 | Map designations and properties.

Map designation	Map environment	Map design	Major map properties
M1	Indoor	Real-world	GCP referenced, residential, obstacle-free
M2	Indoor	Real-world	GCP referenced, office space, obstacles present
M3	Indoor	Simulation	GCP referenced office space and obstacles present. (M3 is the M2 space, recreated for simulation tests)
M4	Outdoor	Simulation	No GCP, outdoor terrain, obstacles present

TABLE 5 | DJI TELLO EDU spec sheet.

General parameters	Value
Weight (including propeller guards)	87 g
Maximum speed	17.8 mph (28.8 kph)
Maximum flight time	13 min
Camera	Value
Maximum image size	2,592 × 1,936
Video recording modes	H.D.: 1,280 × 720
Battery	Value
Capacity	1,100 mAh
Voltage	3.8 V
Energy	4.18 Wh

This performance test measured the time to distress, time to rescue and observed the number of collisions, rescue decisions, and the battery threshold value. The only pose check conducted to evaluate the b_{level} of the distressed agent.

PT2 again used two regular agents and one rescue agent on map M2. The objective of PT2 was to observe p_1 and p_2 , and to

attempt a recovery. A 2D top-down representation of M2 is in **Figure 8**. The number of GCPs could be increased or decreased with a maximum number of up to 20 GCPs placed in the fly space.

Figure 9 shows the M2 space where the three agents were released. The distress condition was simulated for one agent, where it landed behind the table. The rescue agent moved on location to conduct pose checks and begin recovery attempts. The distressed agent was not visible in the global view. However, various situations were observed using the rescue agent and p_1 . **Figure 10** (top-right) shows the rescue agent’s POV, where the distressed agent fell at an oblique angle.

Additionally, as the agent had strayed under the table, the height of the table prevented the agent from gaining the minimum altitude required to conduct a safe rejoin operation. This exemplifies how p_1 helps understand the distressed agent’s situation. **Figure 10** (bottom-right) shows a different situation where the agent has landed in a pose that could allow it to take off. However, its minimum altitude rejoin value was still greater than the obstacle dimensions. In both situations, the operator recommended that further recovery operations be terminated.

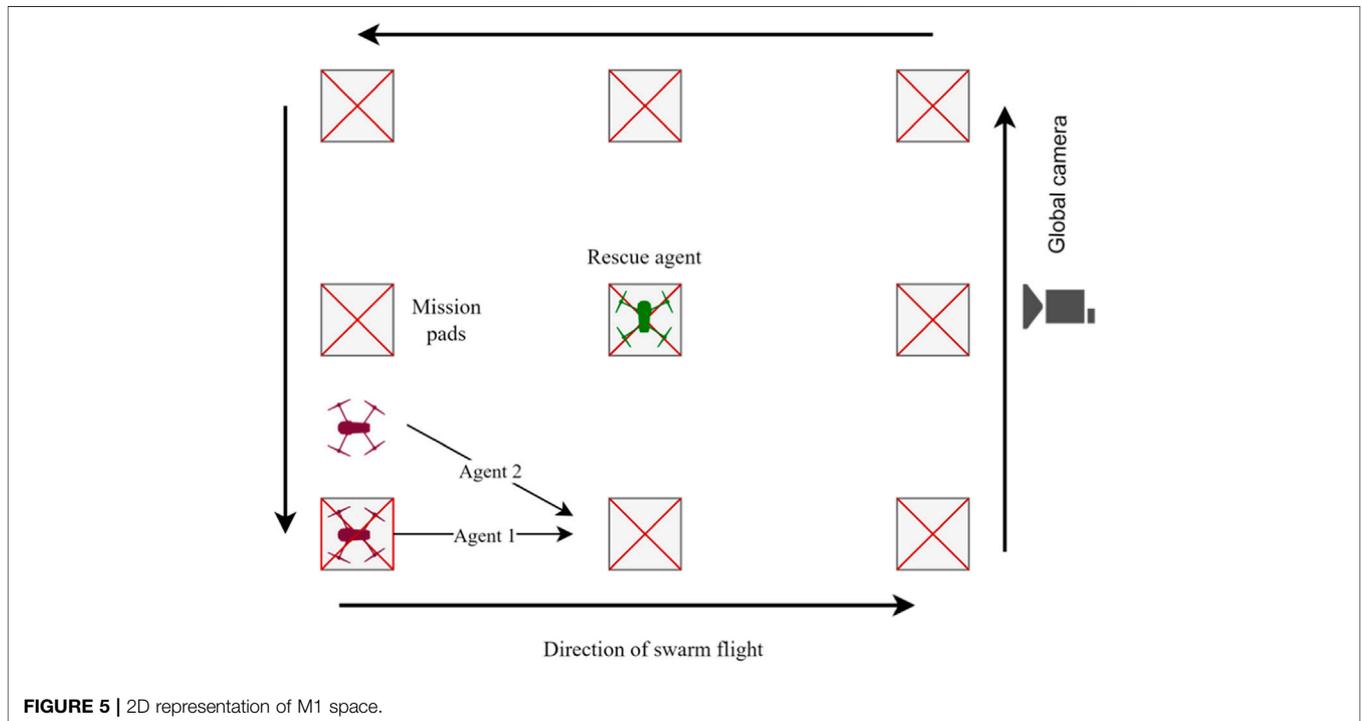




FIGURE 6 | Preliminary framework test PT1 in the M1 space.

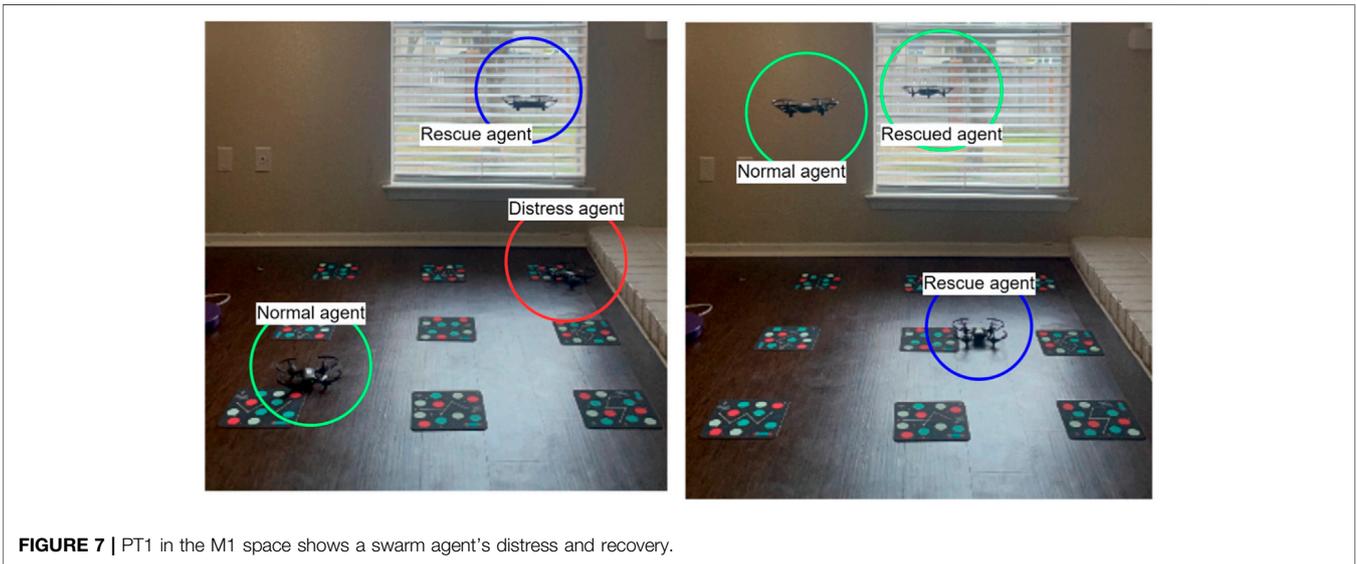


FIGURE 7 | PT1 in the M1 space shows a swarm agent's distress and recovery.

TABLE 6 | Simulation parameters for PT3.

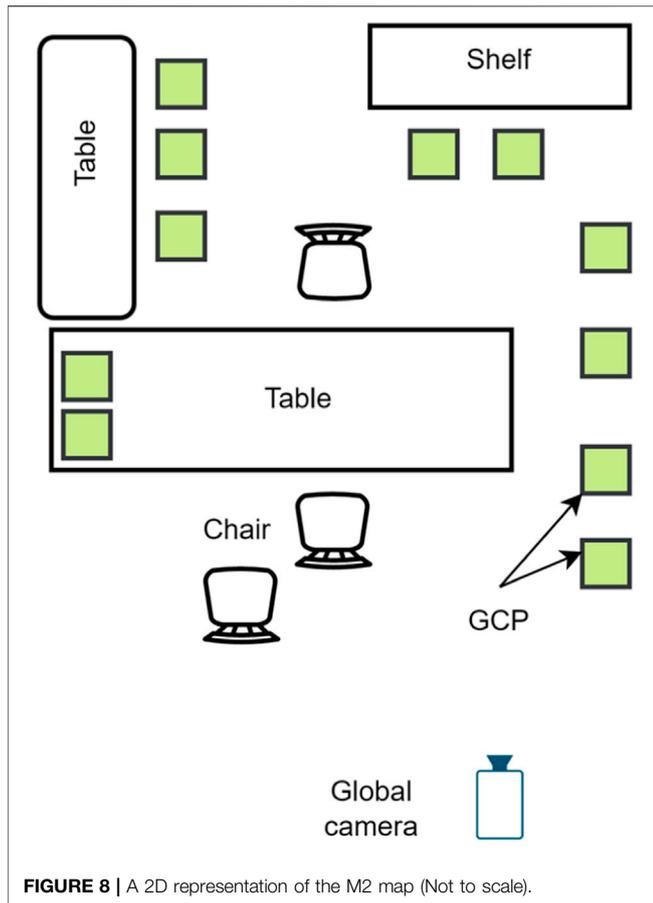
Simulation parameters	Description
Target space	Close re-creation of physical space M2 for PT3
Simulation time	<500 s (Variable)
Size	20 m × 20 m

Figure 11 shows a third situation where the agent landed in a position from which it could take off, plus the right side figure shows the rescue agent's POV from which the operator determined that obstacle dimensions did not impede the distressed agent's safe rejoin procedure upon reconnection. In this situation, the operator recommended the framework to carry further rescue statements on the distressed agent.

Simulation Performance Tests

Indoor and outdoor scenarios to test the proposed framework were modeled in CoppeliaSim, formerly VREP [57]. Table 6 outlines the basic simulation parameters for PT3 and PT4. PT3 was a simulation experiment carried out on M3 (Figure 12A), which is a close recreation of the M2 space used in the hardware experiments. The primary purpose of PT3 was to evaluate p_0 , p_1 , and p_2 and attempt a recovery.

A simple table and chair environment are used to show a failed p_1 and p_2 scenario (Figure 12B). Pose check p_1 used the agent vision sensor information to realize that the distressed agent failed in an inverted position. A normal decision cycle prevents the rescue agent from deploying on a failed p_1 ; however, a forced p_2 cycle using a rescue agent shows that the table dimensions would hinder a safe rejoin maneuver of



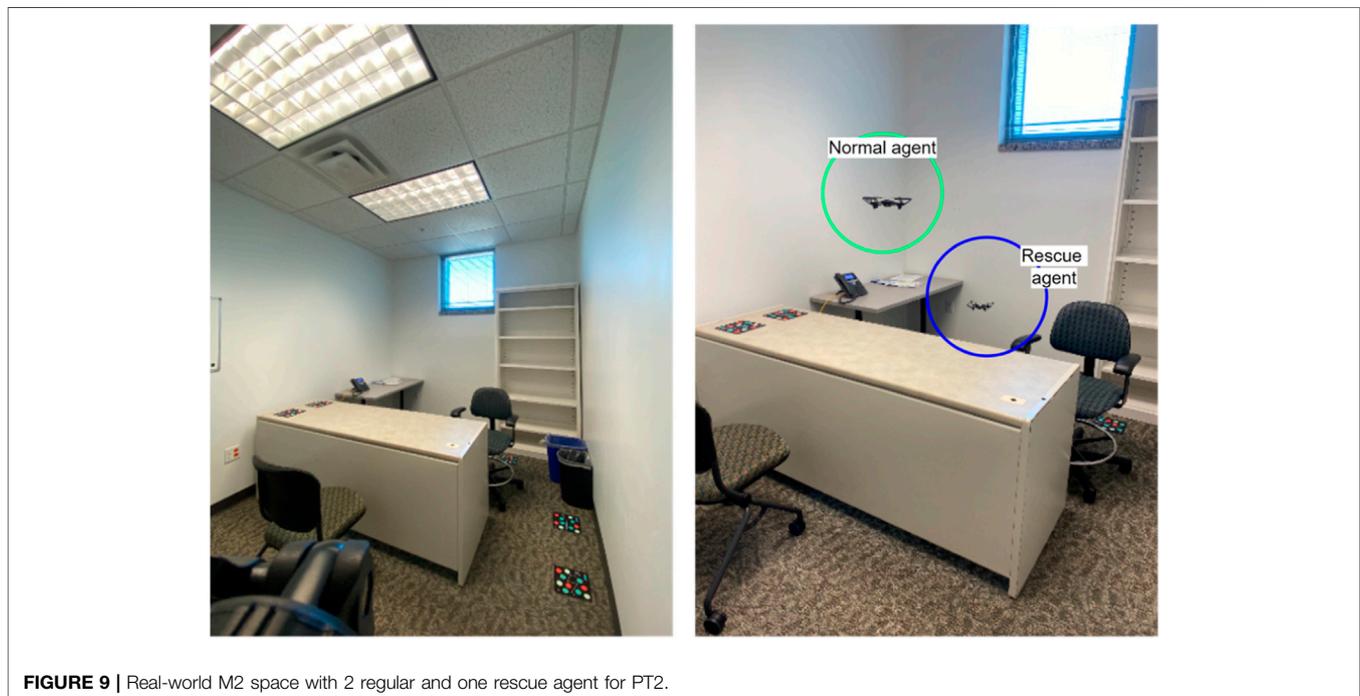
the distressed agent even if the agent were not in an inverted position.

The primary purpose of PT4 was to evaluate p_0 , p_1 , and p_2 , observe time to distress, time to rescue, and log any collision occurrence. **Table 7** outlines the basic simulation parameters for PT4. **Figure 13** shows PT4 on M4, where pose checks p_0 , p_1 , and p_2 were tested along with successful swarm rejoin scenarios. An abstract cube was placed in the field of view of the distressed agent to indicate its orientation for checking p_0 . The rescue agent was then used to determine p_1 by observing the status of the distressed agent. Finally, a p_2 test evaluated if the distressed agent sensed it could rejoin the swarm.

Figure 14 shows an updated map where trees were present as obstacles that hindered agent progress and rejoin maneuvers. Here, multiple agents in the distress scenario were tested, where one agent landed in an inverted position and the other in a normal position. The rescue agent conducted a p_2 check on both agents to determine which agent could be safely recovered. Floating view windows in the figure show p_1 checks by both agents and a p_2 check by the rescue agent on one of the distressed agents.

RESULTS

The following section highlights observations recorded during each performance test and their analysis. **Figure 15** shows PT1 time to distress logs, the time when the agent first experienced an issue, and the time to rescue, which is the amount of time the rescue agent took to move to the position



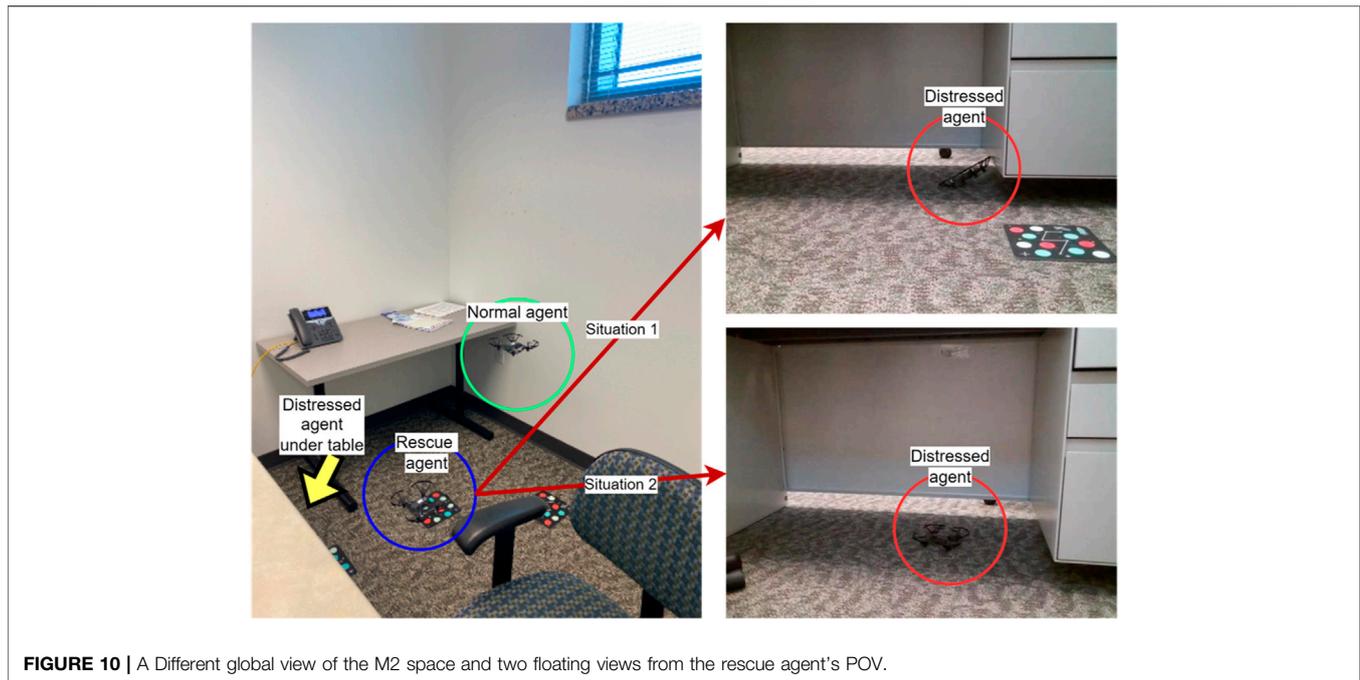


FIGURE 10 | A Different global view of the M2 space and two floating views from the rescue agent's POV.

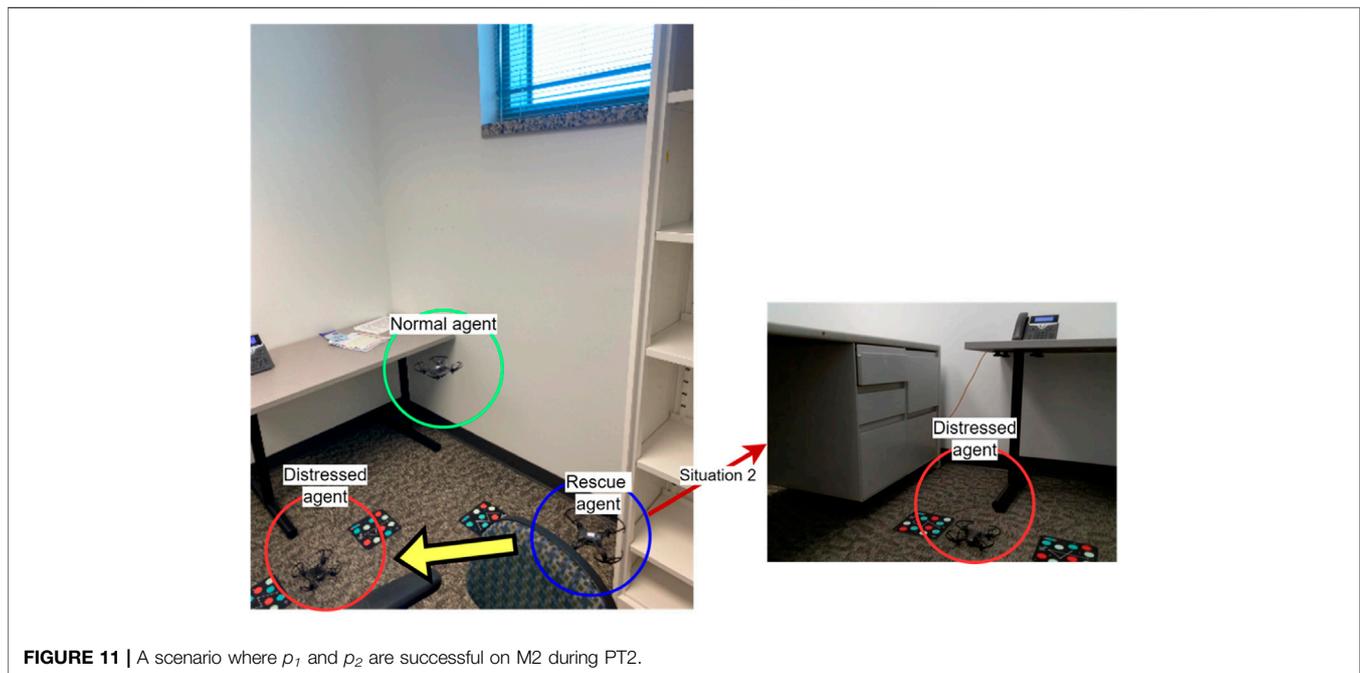


FIGURE 11 | A scenario where p_1 and p_2 are successful on M2 during PT2.

of the fallen agent and rescue it. Out of 15 flights, the rescue agent successfully rescues the distressed Tello agent nine times, as denoted by a green dot in **Figure 15**. A preliminary p_2 check was performed using the available battery percentage when rcm was successful. The battery values are in **Figure 16**. If the battery value after successful reconnection was established between the Rescue and the distressed agent was below the given threshold (50%), the distressed agent was deemed incapable of rejoining the

swarm. This was observed during flights 5, 7, 12, and 14. Each flight was independent, and the battery was charged to maximum capacity before each flight. Collision occurrence was counted when the rescue agent experienced collision at any time during the rescue process. As such, those flights were recorded as an unsuccessful recovery. During flights 4 and 10, the rescue agent experienced a collision and could not recover the fallen agent successfully; these flights were logged as failures. It was observed

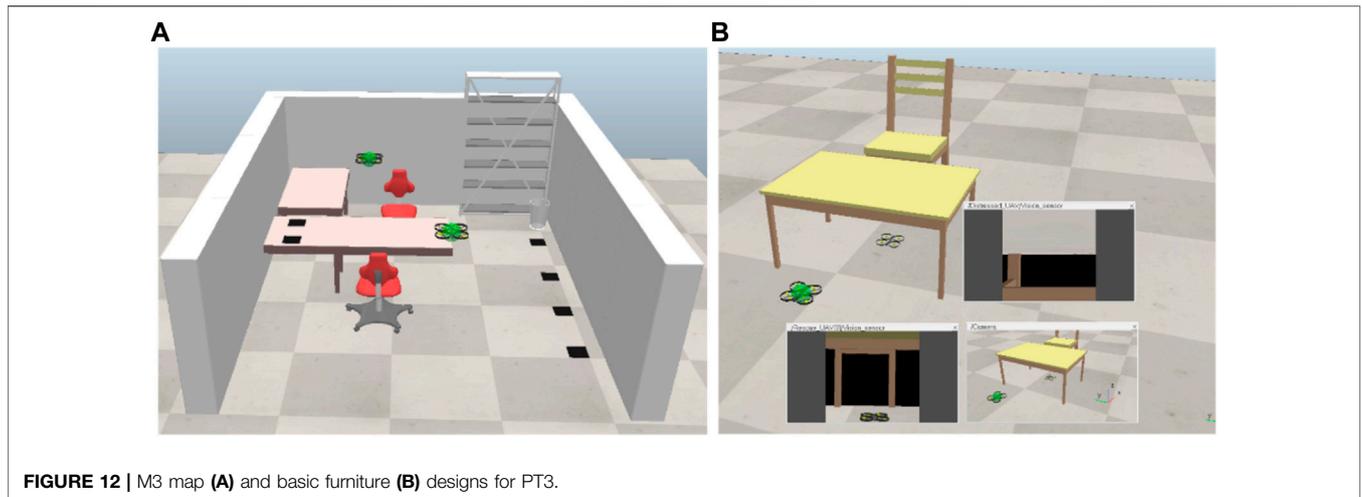


FIGURE 12 | M3 map (A) and basic furniture (B) designs for PT3.

TABLE 7 | Simulation parameters for PT4.

Simulation parameters	Description
Target space	Outdoor environment
Simulation time	<500 s (Variable)
Size	100 m × 100 m

that close interaction with swarm agents in the constrained airspace caused unpredictable drifts in agent movement due to induced airflow, resulting in collisions and crashes. Overall, a recovery rate of 60% was thus calculated for PT1.

Figure 17 shows PT2 performed on M2. For ten flights, p_1 and p_2 rescue decisions were recorded. This test aimed to observe these pose checks and how they affect agent recovery. Flights 4, 7, 8, and 9 showed a successful recovery. Flights 1 to 3 and 5 failed p_1 , where it was determined by the operator using

the rescue agent that the fallen agent was not in a position from which it could safely take off. In M2, this scenario was due to indoor obstacles, such as furniture, that might prevent the agents' safe take-off ability. In Flight 6, the agent passed p_1 , which denoted it was in an orientation and position that could enable safe take-off; however, it failed real-time pose check p_2 . For flight 10, both p_1 and p_2 were successful. However, the agent could not take off due to an internal malfunction. A success rate of 40% for PT2 was observed.

In addition to previous pose checks, PT3 on M3 also performed preliminary pose check p_0 on the agents using the distressed agent's onboard vision sensor, as shown in Figure 12. Figure 18 presents the ten flights performed. Flights 3, 5, 6, and 8 successfully recovered the distressed agent. For flights 1, 7, and 10, the agent failed p_0 , indicating that the agent was not in a position to take off safely. As a result of the preliminary pose check failing, the rescue agent was not deployed to conduct

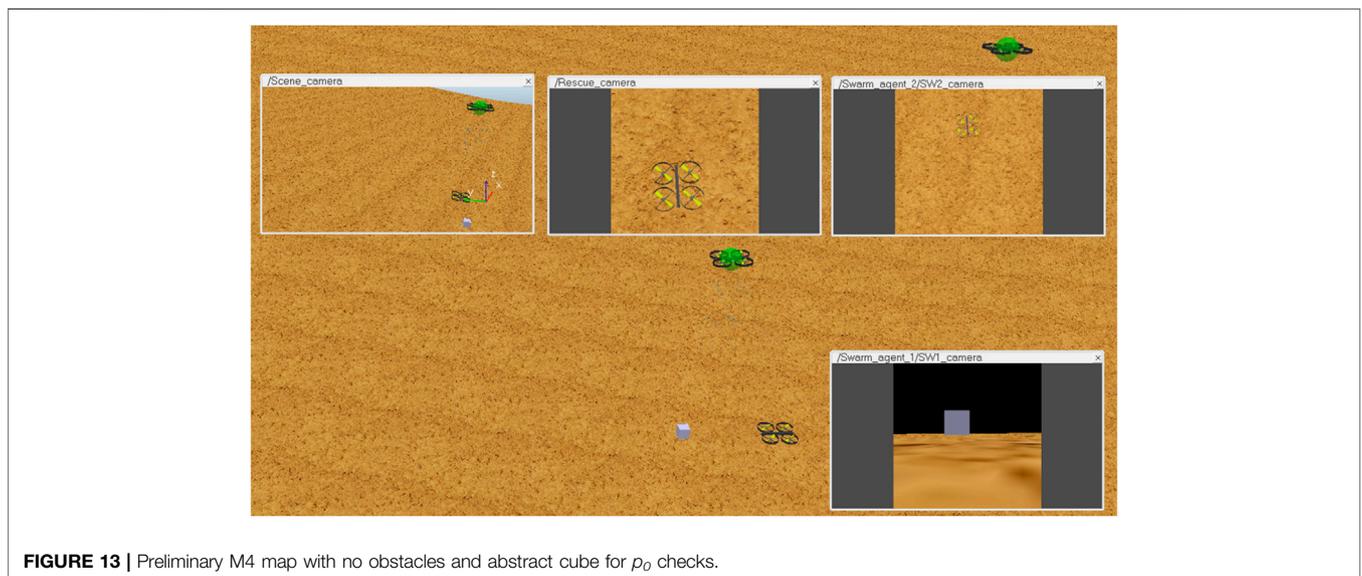


FIGURE 13 | Preliminary M4 map with no obstacles and abstract cube for p_0 checks.



FIGURE 14 | M4 map with outdoor terrain and tree obstacles.

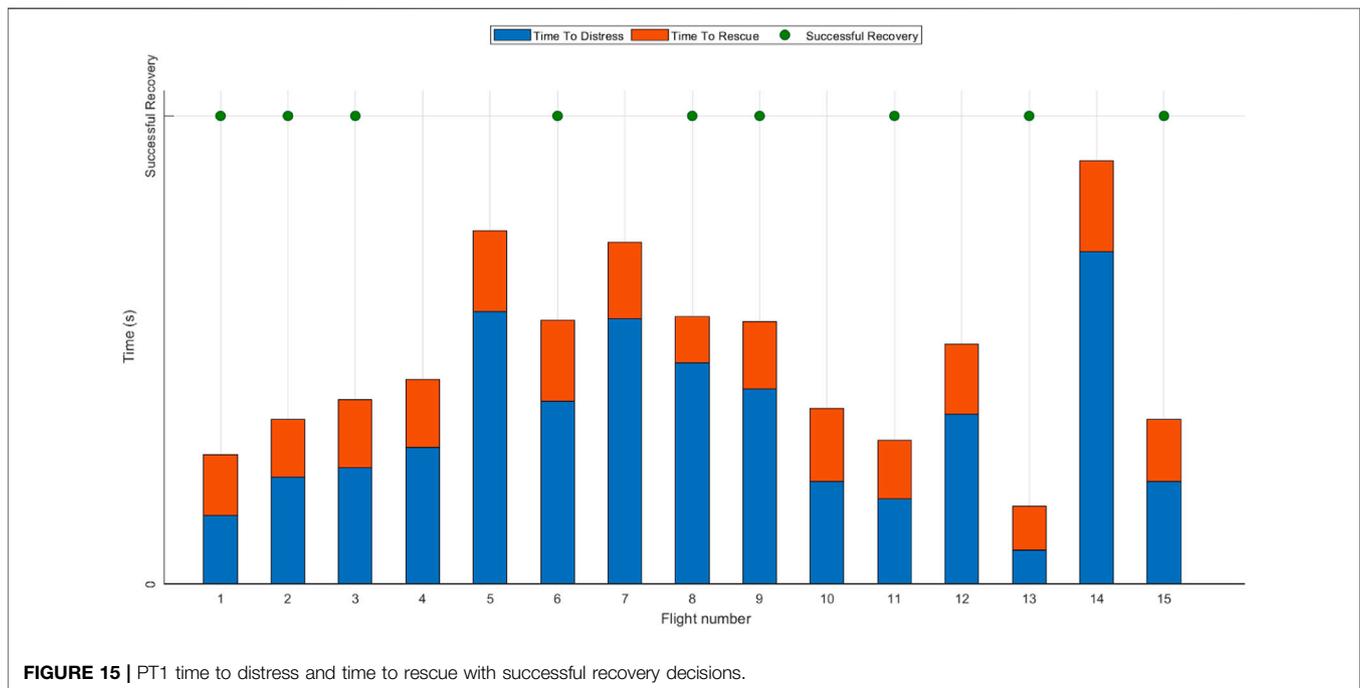


FIGURE 15 | PT1 time to distress and time to rescue with successful recovery decisions.

further observations. In Flight 2, p_0 passed. However, p_1 failed, leading to a failed rescue attempt. In Flights 4 and 9, a successful p_0 and p_1 were observed. However, the agent failed real-time check p_2 and was thus labeled unrecoverable. A success rate of 40% was observed for this test.

PT4 on M4 further examined an additional ten flights, and the results are shown in Figure 19, where time to distress logs the time a swarm agent experiences an issue, and time to rescue logs the time the rescue agent takes during rescue attempts. Flights 4, 5, 6, 7, and 9 showed the rescue agent's successful recovery of the distressed agent.

Further examination of operational parameters, as shown in Figure 20, gives additional failure information. In flights 1, 8, and 10, the distressed agent passed p_0 , which denoted its orientation passed requirements for safe rejoin. However, p_1 failed. Since PT4 was performed on an outdoor terrain map that included tree obstacles, the primary reason for p_1 to fail was the trees obstructing safe rejoin maneuvers. In-flight 2, the distressed agent failed to p_0 itself, as denoted by the onboard sensor that gave information regarding its orientation and crash position. The floating window views in Figure 14 for the distressed agent one vision sensor FOV show an example of an agent that has

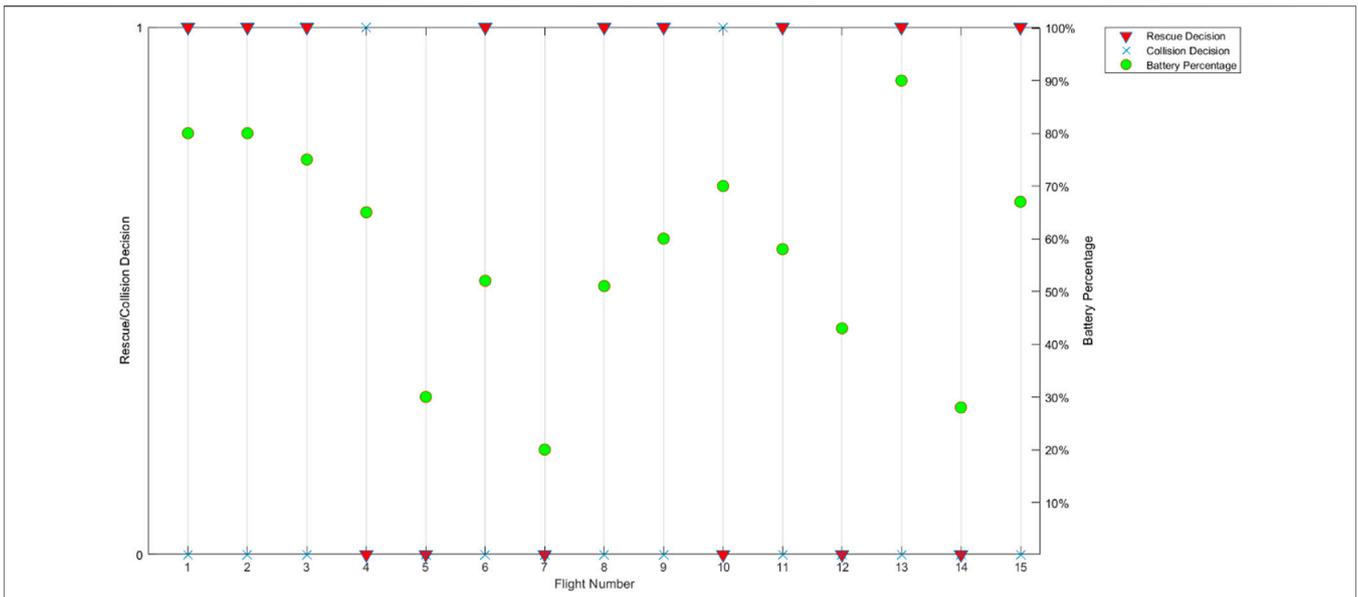


FIGURE 16 | PT1 rescue decision and collision occurrence plotted with battery percentage values.

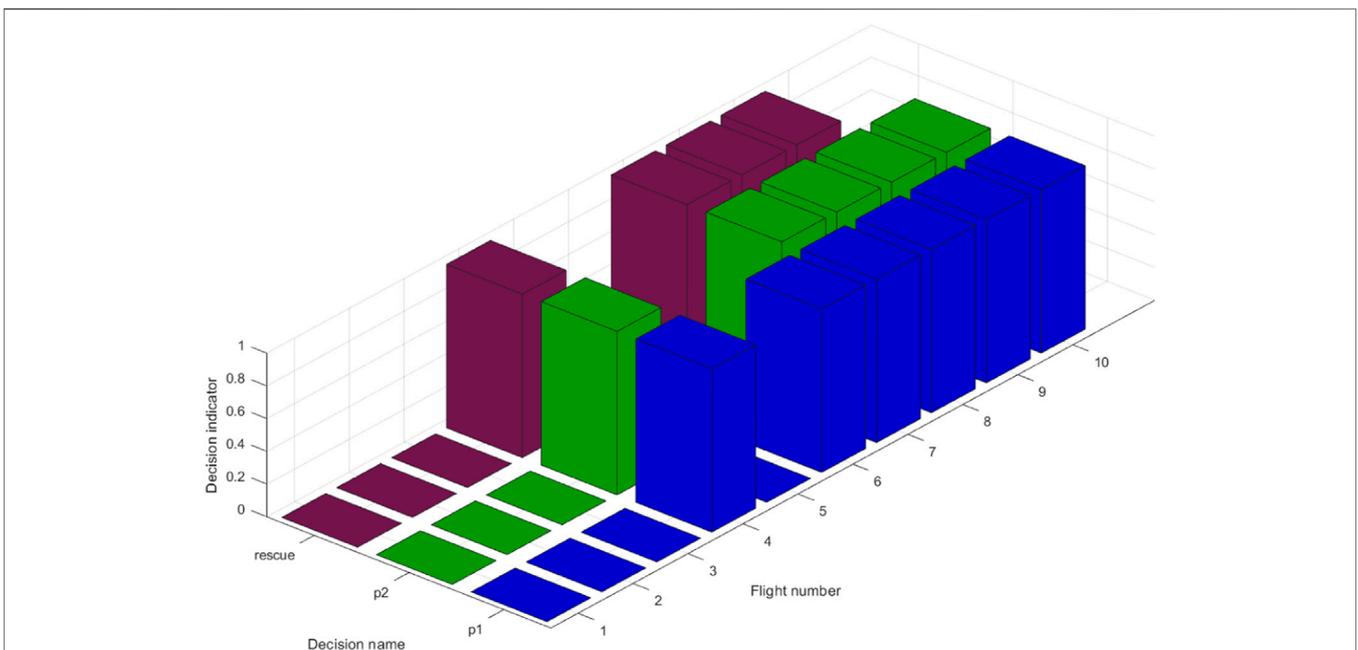


FIGURE 17 | PT2 pose checks and rescue decisions.

landed upside down. The inverted image shows the ground above and tree foliage below. In flight 3, the distressed agent passed all required pose checks; however, a collision with the rescue agent resulted in a failed attempt. Overall, a success rate of 50% was observed. A summary of success rates for all performance tests is shown in **Figure 21**.

A test on a sparse block map depicted in **Figure 22** was used to log agent loss and recovery incidents. Every distress signal

was mapped as a triangle or square in that process. The triangles indicate agent loss due to network issues, and the squares indicate agent loss due to collision. Red shapes indicate an unsuccessful attempt at recovery, and blue shapes indicate successful agent recovery. Each shape is the result of a separate flight, and consecutive flights did not have prior risk zone information. However, creating such risk zones can then be used as future input parameters to create safe flight paths.

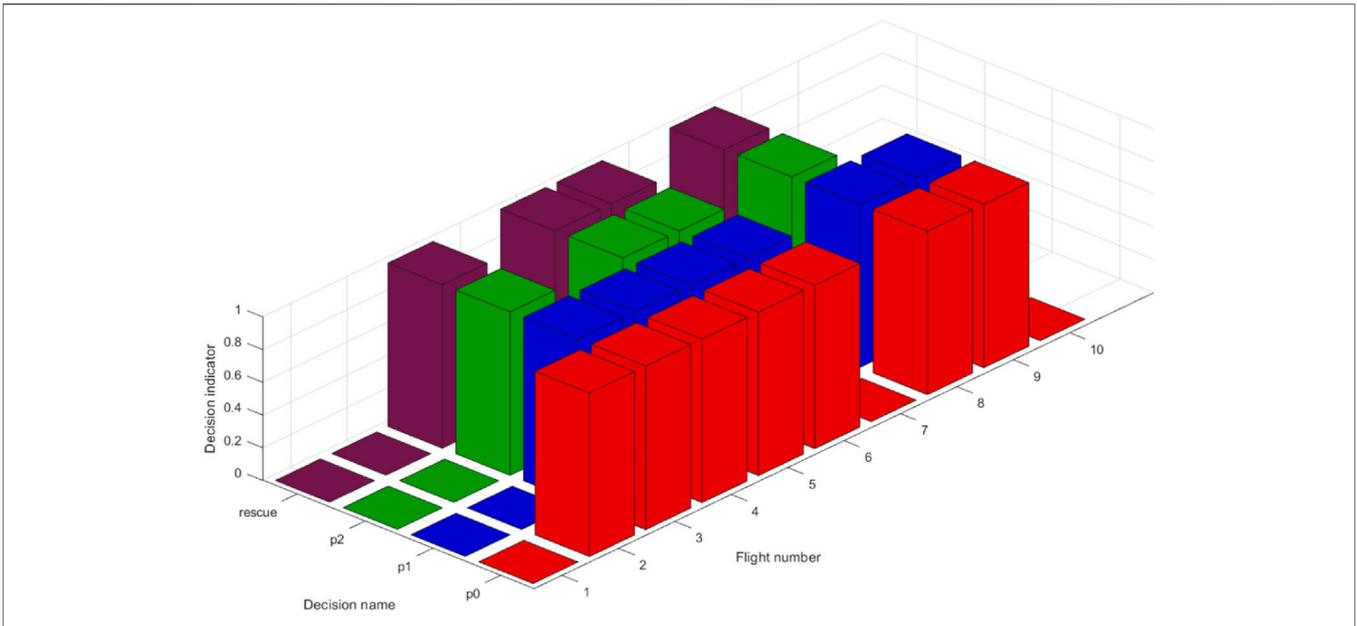


FIGURE 18 | PT3 pose checks and Rescue decisions.

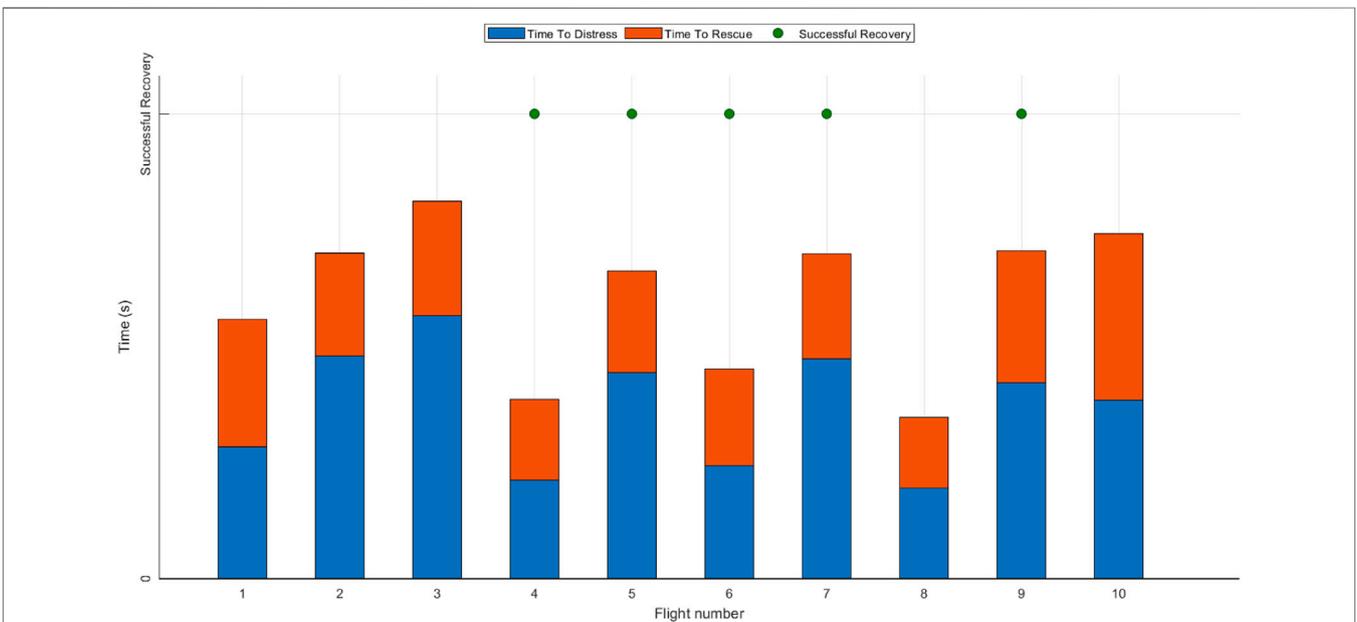


FIGURE 19 | PT4 time to distress and time to rescue with successful recovery decisions.

For example, if an area sees increased collisions due to dense obstacle geometry, a threat area can be modeled where agents entering that area do not venture below a preset altitude to avoid collisions. If agents moving to a particular area lose connection with ground control, the next iteration framework run will adjust the upper bound distance between the agents, which defines the maximum distance between two agents

based on SSI_n . Adjusting the upper bounds will result in agents flying in close formation and using data hop pathways to connect to ground control and prevent agent loss due to network range limitations. Future work using this approach can demonstrate adaptability, robustness, and emergent behavior in the swarm based on simple governance rules.

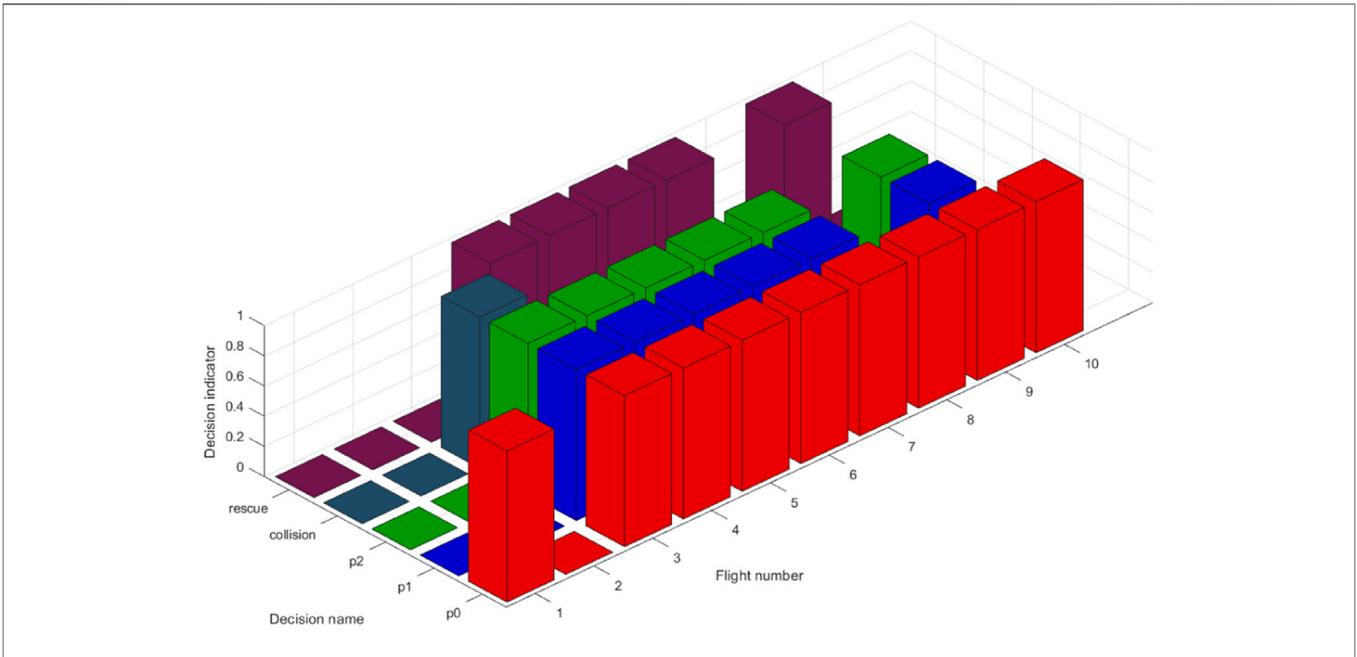


FIGURE 20 | PT4 pose checks and rescue decisions.

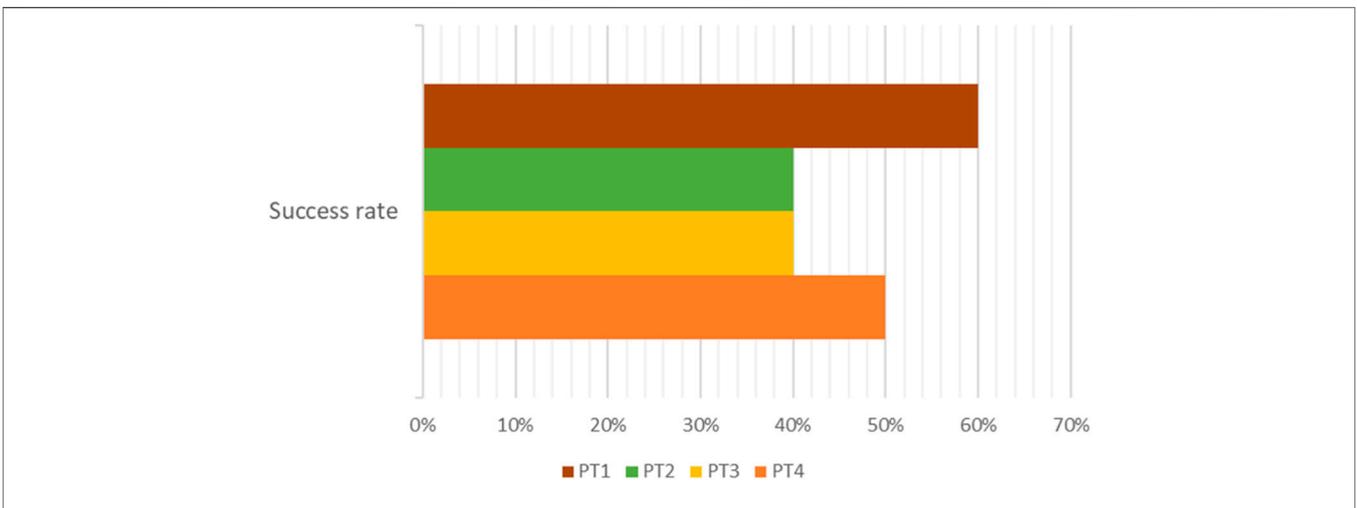


FIGURE 21 | Summary of PT results.

DISCUSSION AND FUTURE RESEARCH DIRECTIONS

The proposed framework is a preliminary step in developing robust methodologies for evaluating swarm awareness toward the wellbeing of its constituent agents. This includes testing capabilities such as keeping track of each agent’s progress toward its task, realizing the occurrence of agents in distress, locating the distressed agents, and initiating rescue operations to enable them to rejoin the swarm. Several modifications could be implemented via the modular nature of the designed framework, as initiated by research directions summarized below.

It is crucial to consider the impact of emerging regulations on UAV operations, particularly the recent implementation of the FAA’s Remote Identification (RID) rule [58]. This regulation mandates the use of Remote Identification modules on certain UAVs, allowing for the open broadcast and identification of these agents during flight. This rule ensures safer airspace and promotes regulated use of UAVs, UAV swarms, and their applications [59]. When integrated into our rescue framework, the potential for such information can significantly enhance tracking and rescue performance. By leveraging the real-time identification capabilities provided by RID, it is foreseen that such frameworks can precisely locate and rescue other agents within

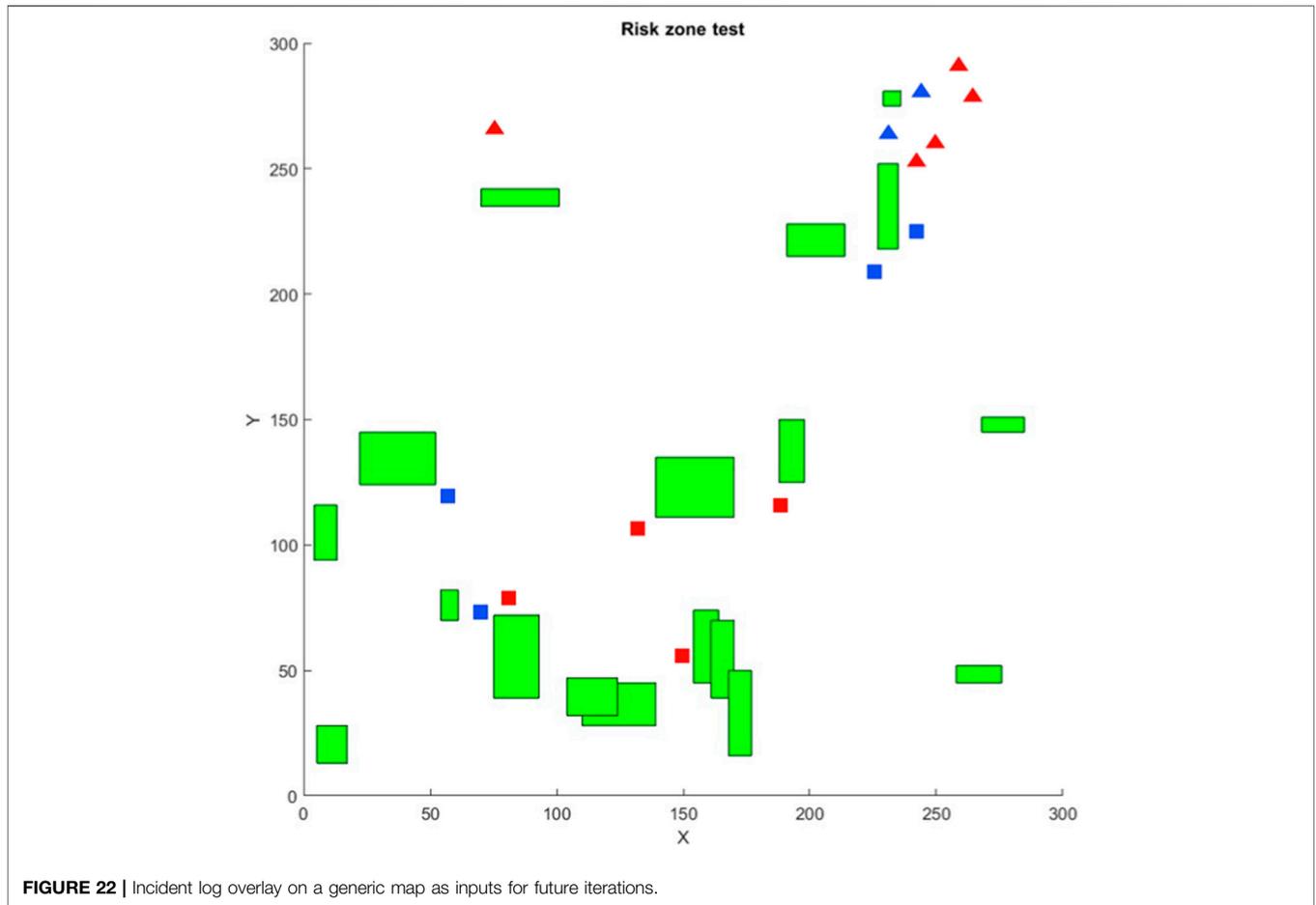


FIGURE 22 | Incident log overlay on a generic map as inputs for future iterations.

TABLE 8 | Summary of recent work on optimal abort policies, task rescheduling, and dynamic risk assessment.

Reference	Implementation	Description
[72]	General systemic deployment	Optimally aborting subtasks in heterogeneous swarms to increase overall unit survivability rate
[73]	General systemic deployment	Design the best abort strategy for multi-unit swarms based on the probability of external shocks damaging units
[74]	Single UAV focused	Design of replacement policies and maintenance cost for UAV reconnaissance system
[75]	Single UAV focused	Dynamic allocation of a fixed number of components to increase the mission completion rate by UAV in a reconnaissance scenario
[76]	UAV swarm focused	Considering the cost of damaged agents and unfinished tasks to compute abort policies
[77]	UAV swarm focused	Evaluate system mission reliability and suggest swarm maintenance strategies
[78]	UAV swarm focused	Incorporating abort policies in multi-UAV routing as a response to external shocks to ensure agent wellbeing
[79]	UAV swarm focused	Consider degradation level, mission time, and equipment health to create dynamic mission abort policies

the swarm more effectively, thus bolstering the overall efficiency and reliability of the proposed UAV rescue mechanism. Furthermore, exploring the compatibility and interoperability of our rescue system with other upcoming regulatory frameworks will be essential in ensuring the seamless integration and widespread adoption of our research in real-world UAV swarm applications.

Currently, some sections of the framework involve human decision-making. Most notable is the analysis of the distressed agent pose data transmitted by the rescue agent. The human

operator observes the images to create a preliminary decision on the fallen agent’s possibility of rejoining the swarm. The human in-loop component can be reduced by adding autonomous UAV detection capability that uses vision sensor data, deep learning, and image processing techniques. This is possible using approaches such as [60] that use agent vision sensors for target analysis. An additional upgrade involves multiple agents to capture disruption and distressed agent information from different angles to gather a richer dataset.

A modified task re-allocation algorithm would enable additional agents to join the swarm and take up the

interrupted task of the fallen agent or a re-allocation scheme for existing swarm agents to assume responsibility for the incomplete task. Resource allocation implementations such as in [61–63] exist that could be implemented. Further experiments could be explored for loss rates in the same airspace with LiDAR (for obstacle detection) and preset waypoints in a map (using GNSS). This would expand the feature of the existing framework to create probability maps. While current risk zones were labeled using agent failure location data obtained from transmitted HBS, future experiments may include the presence of VRRZ. This is the system's ability to create variable radius risk zones. Each risk zone can have a variable radius, thus allowing the mapping of larger disruptive structures to be represented more accurately.

The above study uses agents with similar capabilities acting in the same operational space. Including a diverse range of agents identified by their differences in nature, hardware, or operational space introduces heterogeneity in the swarm. The impacts of such inclusion on the performance of SAR agents can also be explored. Some existing research investigates the possibility of using a swarm composed of heterogeneous agents for victim detection after a disaster [64]. Although their main goal was exploring how swarm heterogeneity can affect performance, they modeled a target search and rescue problem to study it. Their proposed technique differentiated between different agents and labeled them as heterogeneous using behavior trees. A positive correlation was produced between the swarm's heterogeneous capability and the time to search and rescue the target. A similar approach can be explored in the future, where differently abled robots are introduced in the swarm and are tasked with looking for swarm agents whose operations have been disrupted during mission progress. In the above experiments, all distressed agents were located on the ground. Thus, adding a UGV to track and locate the fallen agents to create an in-depth pose check analysis would be a logical step for further exploration. Several implementations of heterogeneous swarms exist, such as UAV-UGV collaboration [65], UAV-UWSV [29], and UAV-UGV-UWSV [66], demonstrating promise for more effective results than a single operational space swarm.

Intrusion detection systems can be implemented on the UAV network as a backend process. While IDSs are most prevalent on traditional networks to deter unwanted network access and activity, current lightweight versions have been shown to run reasonably well on MANET and FANET deployments with acceptable performance [19, 67–70]. Various types of IDS are available depending on their makeup and method of detecting malicious entities [71]. IDS could detect external agents attempting to maliciously disturb swarm operations. Similar approaches could also address ground-based attempts to take over swarm networks. The possible advantage would be the existing periodically transmitted HBS signal that can be used as input to any IDS. Adding network transmission data from each agent in the HBS could be used to design either a rule-based or anomaly-based lightweight IDS, at the very least. In this way, the SAR framework could provide additional security features to the swarm using inherently built structures.

A different approach taken to designing robust behavior was observed as a way of defining reliability in systems. The methods

used in this category implement preemptive strategies for maintenance, abort policies, or recovery actions. This alternative form of resilience integration calls for an independent study. However, the results of the brief survey conducted on it are summarized below. These methods can be viewed as possible implementations and upgrades to this proposed SS-SAR framework. **Table 8** summarizes the examined work based on their development focus being broader systemic implementations or UAV swarm-focused.

CONCLUSION

This research addresses gaps in current swarm resiliency research by addressing swarm-specific SAR rather than application-specific SAR. The approach was not to replace current SAR methodologies but to create an add-on that enables them to keep track of swarm agents while performing other functions. Modular experiments conducted on real-world hardware and simulations validated the need for, the possibility of, and the success rate of swarm-specific SAR approaches. While low-cost Tello drones were limited in their ability to handle a complete SS-SAR framework, they were crucial in testing the constituent process of the framework, such as reconnection protocols and pose check handlers. Simulation results provided a greater insight into how such frameworks can handle swarm agent loss. Experimental results prove that focusing on this approach to resiliency integration in multi-agent systems can produce the anticipated benefits. Recovery rates of distressed agents during and after the mission process increased drastically, especially in systems with no contingency rules. UAV swarms are complex and highly dynamic, making integrating resilience factors much more arduous. A system must exhibit awareness and diagnosis capability regarding its health before and after a disruption to efficiently produce solutions to mitigate said disruptions. This swarm-specific SAR framework is a crucial design step in that direction.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

Conceptualization, Methodology, Validation, Writing- Original draft- AP. Methodology, Writing- review and editing, Visualization, Supervision- FM. All authors contributed to the article and approved the submitted version.

CONFLICT OF INTEREST

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

REFERENCES

- Siemiakowska B, Stecz W. A Framework for Planning and Execution of Drone Swarm Missions in a Hostile Environment. *Sensors (Basel)* (2021) 21:4150. doi:10.3390/s21124150
- Ko Y, Kim J, Duguma DG, Astillo PV, You I, Pau G. Drone Secure Communication Protocol for Future Sensitive Applications in Military Zone. *Sensors (Basel)* (2021) 21:2057. doi:10.3390/s21062057
- Allan BM, Nimmo DG, Ierodiaconou D, VanDerWal J, Koh LP, Ritchie EG. Futurecasting Ecological Research: The Rise of Technoecology. *Ecosphere* (2018) 9. doi:10.1002/ecs2.2163
- Nex F, Duarte D, Steenbeek A, Kerle N. Towards Real-Time Building Damage Mapping With Low-Cost UAV Solutions. *Remote Sensing* (2019) 11:287. doi:10.3390/rs11030287
- Noor NM, Abdullah A, Hashim M. Remote Sensing UAV/Drones and Its Applications for Urban Areas: A Review. In: *IGRSM International Conference and Exhibition on Geospatial & Remote Sensing*. Malaysia: Kuala Lumpur (2018).
- Hildmann H, Kovacs E. Review: Using Unmanned Aerial Vehicles (UAVs) as Mobile Sensing Platforms (MSPs) for Disaster Response, Civil Security and Public Safety. *Drones* (2019) 3:59. doi:10.3390/drones3030059
- Klaine PV, Nadas JPB, Souza RD, Imran MA. Distributed Drone Base Station Positioning for Emergency Cellular Networks Using Reinforcement Learning. *Cognit Comput* (2018) 10:790–804. doi:10.1007/s12559-018-9559-8
- Tsouros DC, Bibi S, Sarigiannidis PG. A Review on UAV-Based Applications for Precision Agriculture. *Information* (2019) 10:349. doi:10.3390/info10110349
- Panagiotis Radoglou-Grammatikis PS, Lagkas T, Moscholios I. A Compilation of UAV Applications for Precision Agriculture. *Computer Networks* (2020) 172:107148. doi:10.1016/j.comnet.2020.107148
- Lygouras E, Santavas N, Taitzoglou A, Tarchanidis K, Mitropoulos A, Gasteratos A. Unsupervised Human Detection With an Embedded Vision System on a Fully Autonomous UAV for Search and Rescue Operations. *Sensors (Basel)* (2019) 19:3542. doi:10.3390/s19163542
- Dong J, Ota K, Dong M. UAV-Based Real-Time Survivor Detection System in Post-Disaster Search and Rescue Operations. *IEEE J Miniaturization Air Space Syst* (2021) 2:209–19. doi:10.1109/jmass.2021.3083659
- Phadke A, Antonio Medrano F, Chu T. Engineering Resiliency in UAV Swarms—A Bibliographic Analysis. *J Phys Conf Ser* (2022) 2330:012007. doi:10.1088/1742-6596/2330/1/012007
- Ordoukhanian EM, Azad M. Engineering Resilience Into Multi-UAV Systems. *Proced Comp Sci* (2019) 153:9–16. doi:10.1016/j.procs.2019.05.050
- Hosseini S, Barker K, Ramirez-Marquez JE. A Review of Definitions and Measures of System Resilience. *Reliability Eng Syst Saf* (2016) 145:47–61. doi:10.1016/j.res.2015.08.006
- Woods DD. Four Concepts for Resilience and the Implications for the Future of Resilience Engineering. *Reliability Eng Syst Saf* (2015) 141:5–9. doi:10.1016/j.res.2015.03.018
- Phadke A, Medrano FA. Towards Resilient UAV Swarms—A Breakdown of Resiliency Requirements in UAV Swarms. *Drones* (2022) 6:340. doi:10.3390/drones6110340
- Zhang J, Zhou W, Wang X. UAV Swarm Navigation Using Dynamic Adaptive Kalman Filter and Network Navigation. *Sensors (Basel)* (2021) 21:5374. doi:10.3390/s21165374
- Bassolillo SR, D'Amato E, Notaro I, Blasi L, Mattei M. Decentralized Mesh-Based Model Predictive Control for Swarms of UAVs. *Sensors (Basel)* (2020) 20:4324. doi:10.3390/s20154324
- Tan X, Su S, Zuo Z, Guo X, Sun X. Intrusion Detection of UAVs Based on the Deep Belief Network Optimized by PSO. *Sensors (Basel)* (2019) 19:5529. doi:10.3390/s19245529
- Luo R, Zheng H, Guo J. Solving the Multi-Functional Heterogeneous UAV Cooperative Mission Planning Problem Using Multi-Swarm Fruit Fly Optimization Algorithm. *Sensors (Basel)* (2020) 20:5026. doi:10.3390/s20185026
- Kwa HL, Philippot J, Bouffanais R. Effect of Swarm Density on Collective Tracking Performance. *Swarm Intelligence* (2023) 17:253–81. doi:10.1007/s11721-023-00225-4
- Abhishek P, Medrano FA. Examining Application-Specific Resiliency Implementations in UAV Swarm Scenarios. *Intelligence & Robotics* (2023) 3:453–78. doi:10.20517/ir.2023.27
- Phadke A, Medrano A. *A Resilient Multi-UAV System of Systems (SoS)*. Report no. 2771-9359 (2021).
- Garg V, Tiwari R, Shukla A. Comparative Analysis of Fruit Fly-Inspired Multi-Robot Cooperative Algorithm for Target Search and Rescue. In: 2022 IEEE World Conference on Applied Intelligence and Computing (AIC); 17–19 June 2022; Sonbhadra, India (2022). p. 444–50.
- Cao Y, Qi F, Jing Y, Zhu M, Lei T, Li Z, et al. Mission Chain Driven Unmanned Aerial Vehicle Swarms Cooperation for the Search and Rescue of Outdoor Injured Human Targets. *Drones* (2022) 6:138. doi:10.3390/drones6060138
- Oh D, Han J. Smart Search System of Autonomous Flight UAVs for Disaster Rescue. *Sensors (Basel)* (2021) 21:6810. doi:10.3390/s21206810
- Cardona GA, Calderon JM. Robot Swarm Navigation and Victim Detection Using Rendezvous Consensus in Search and Rescue Operations. *Appl Sci* (2019) 9:1702. doi:10.3390/app9081702
- Alotaibi ET, Alqafari SS, Koubaa A. LSAR: Multi-UAV Collaboration for Search and Rescue Missions. *IEEE Access* (2019) 7:55817–32. doi:10.1109/access.2019.2912306
- Yang T, Jiang Z, Dong J, Feng H, Yang C. Multi Agents to Search and Rescue Based on Group Intelligent Algorithm and Edge Computing. In: 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData); 30 July 2018 - 03 August 2018; Halifax, NS, Canada (2018). p. 389–94.
- Ho Y-H, Tsai Y-J. Open Collaborative Platform for Multi-Drones to Support Search and Rescue Operations. *Drones* (2022) 6:132. doi:10.3390/drones6050132
- Yan P, Jia T, Bai C. Searching and Tracking an Unknown Number of Targets: A Learning-Based Method Enhanced With Maps Merging. *Sensors (Basel)* (2021) 21:1076. doi:10.3390/s21041076
- Papaioannou S, Kolios P, Theocharides T, Panayiotou CG, Polycarpou MM. Distributed Search Planning in 3-D Environments With a Dynamically Varying Number of Agents. *IEEE Trans Syst Man, Cybernetics: Syst* (2023) 53:4117–30. doi:10.1109/tsmc.2023.3240023
- Liu S, Yao W, Zhu X, Zuo Y, Zhou B. Emergent Search of UAV Swarm Guided by the Target Probability Map. *Appl Sci* (2022) 12:5086. doi:10.3390/app12105086
- Li L, Xu S, Nie H, Mao Y, Yu S. Collaborative Target Search Algorithm for UAV Based on Chaotic Disturbance Pigeon-Inspired Optimization. *Appl Sci* (2021) 11:7358. doi:10.3390/app11167358
- Cho S-W, Park J-H, Park H-J, Kim S. Multi-UAV Coverage Path Planning Based on Hexagonal Grid Decomposition in Maritime Search and Rescue. *Mathematics* (2021) 10:83. doi:10.3390/math10010083
- Walker O, Vanegas F, Gonzalez F. A Framework for Multi-Agent UAV Exploration and Target-Finding in GPS-Denied and Partially Observable Environments. *Sensors (Basel)* (2020) 20:4739. doi:10.3390/s20174739
- Alhaqbani A, Kurdi H, Youcef-Toumi K. Fish-Inspired Task Allocation Algorithm for Multiple Unmanned Aerial Vehicles in Search and Rescue Missions. *Remote Sensing* (2020) 13:27. doi:10.3390/rs13010027
- Yue W, Guan X, Wang L. A Novel Searching Method Using Reinforcement Learning Scheme for Multi-UAVs in Unknown Environments. *Appl Sci* (2019) 9:4964. doi:10.3390/app9224964
- Li X, Chen J, Deng F, Li H. Profit-Driven Adaptive Moving Targets Search With UAV Swarms. *Sensors (Basel)* (2019) 19:1545. doi:10.3390/s19071545
- de Alcantara Andrade FA, Reinier Hovenburg A, Netto de Lima L, Rodin CD, Johansen TA, Storvold R, et al. Autonomous Unmanned Aerial Vehicles in Search and Rescue Missions Using Real-Time Cooperative Model Predictive Control. *Sensors (Basel)* (2019) 19:4067. doi:10.3390/s19194067
- Huang S, Teo RSH, Kwan JLP, Liu W, Dymkou SM. Distributed UAV Loss Detection and Auto-Replacement Protocol With Guaranteed Properties. *J Intell Robot Syst* (2019) 93:303–16. doi:10.1007/s10846-018-0818-4
- Godio S, Primatesta S, Guglieri G, Dovis F. A Bioinspired Neural Network-Based Approach for Cooperative Coverage Planning of UAVs. *Information* (2021) 12:51. doi:10.3390/info12020051

43. Xie Y, Han L, Dong X, Li Q, Ren Z. Bio-Inspired Adaptive Formation Tracking Control for Swarm Systems With Application to UAV Swarm Systems. *Neurocomputing* (2021) 453:272–85. doi:10.1016/j.neucom.2021.05.015
44. Wang C, Wang D, Gu M, Huang H, Wang Z, Yuan Y, et al. Bioinspired Environment Exploration Algorithm in Swarm Based on Lévy Flight and Improved Artificial Potential Field. *Drones* (2022) 6:122. doi:10.3390/drones6050122
45. Xu B, Liu T, Bai G, Tao J, Zhang Y, Fang Y. A Multistate Network Approach for Reliability Evaluation of Unmanned Swarms by Considering Information Exchange Capacity. *Reliability Eng Syst Saf* (2022) 219:108221. doi:10.1016/j.res.2021.108221
46. Wang K, Zhang X, Qiao X, Li X, Cheng W, Cong Y, et al. Adjustable Fully Adaptive Cross-Entropy Algorithms for Task Assignment of Multi-UAVs. *Drones* (2023) 7:204. doi:10.3390/drones7030204
47. Zeng Y, Ren K, Sun Q, Zhang Y. *A Resilient Task Allocation Method for UAV Swarm Based on PSO-ILP Bilevel Nested Optimization* (2023). Available from: <https://doi.org/10.21203/rs.3.rs-2765091/v1>.
48. Bejiga M, Zeggada A, Nouffidj A, Melgani F. A Convolutional Neural Network Approach for Assisting Avalanche Search and Rescue Operations With UAV Imagery. *Remote Sensing* (2017) 9:100. doi:10.3390/rs9020100
49. Phadke A, Medrano A. Drone2Drone: A Search and rescue Framework for Finding Lost UAV Swarm Agents. Poster. In: *Symposium for Student Innovation, Research, and Creative Activities 2023*. United States: Texas A&M University-Corpus Christi (2023).
50. Tello DJI. *EDU User Manual* (2018).
51. Wheeb AH, Nordin R, Samah AA, Kanellopoulos D. Performance Evaluation of Standard and Modified OLSR Protocols for Uncoordinated UAV Ad-Hoc Networks in Search and Rescue Environments. *Electronics* (2023) 12:1334. doi:10.3390/electronics12061334
52. Li B, Jiang Y, Sun J, Cai L, Wen CY. Development and Testing of a Two-UAV Communication Relay System. *Sensors (Basel)* (2016) 16:1696. doi:10.3390/s16101696
53. Phadke A, Medrano FA, Sekharan CN, Chu T. Designing UAV Swarm Experiments: A Simulator Selection and Experiment Design Process. *Sensors* (2023) 23:7359. doi:10.3390/s23177359
54. Pohudina O, Kovalevskiy M, Pyvovar M. Group Flight Automation Using Tello EDU Unmanned Aerial Vehicle. In: 2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT); 22–25 September 2021; LVIV, Ukraine (2021). p. 151–4.
55. Virbora N, Sokoeun U, Saran M, Sovicheyratana S, Channareth S, Saravuth S. Implementation of Matrix Drone Show Using Automatic Path Generator With DJI Tello Drones. In: 2022 International Conference on Engineering and Emerging Technologies (ICEET); 27–28 October 2022; Kuala Lumpur, Malaysia (2022). p. 1–5.
56. Pikalov S, Azaria E, Sonnenberg S, Ben-Moshe B, Azaria A. Vision-Less Sensing for Autonomous Micro-Drones. *Sensors (Basel)* (2021) 21:5293. doi:10.3390/s21165293
57. Rohmer E, Singh SPN, Freese M. CoppeliaSim (Formerly V-REP): A Versatile and Scalable Robot Simulation Framework. In: IEEE/RSJ International Conference on Intelligent Robots and Systems; 03–07 November 2013; Tokyo, Japan (2013). p. 1321–6.
58. FAA. Remote Identification of Unmanned Aircraft-Final Rule. In: FAA D, editor (2023). Available at: https://www.faa.gov/sites/faa.gov/files/2021-08/RemoteID_Final_Rule.pdf.
59. Phadke A, Boyd J, Medrano FA, Starek M. Navigating the Skies: Examining the FAA's Remote Identification Rule for Unmanned Aircraft Systems. *Drone Syst Appl* (2023) 11:1–4. doi:10.1139/dsa-2023-0029
60. Opromolla R, Inchingolo G, Fasano G. Airborne Visual Detection and Tracking of Cooperative UAVs Exploiting Deep Learning. *Sensors (Basel)* (2019) 19:4332. doi:10.3390/s19194332
61. Razaq S, Xydeas C, Mahmood A, Ahmed S, Ratyal NI, Iqbal J. Efficient Optimization Techniques for Resource Allocation in UAVs Mission Framework. *PLoS One* (2023) 18:e0283923. doi:10.1371/journal.pone.0283923
62. Nguyen M-N, Nguyen LD, Duong TQ, Tuan HD. Real-Time Optimal Resource Allocation for Embedded UAV Communication Systems. *IEEE Wireless Commun Lett* (2019) 8:225–8. doi:10.1109/lwc.2018.2867775
63. Yu Z, Gong Y, Gong S, Guo Y. Joint Task Offloading and Resource Allocation in UAV-Enabled Mobile Edge Computing. *IEEE Internet Things J* (2020) 7: 3147–59. doi:10.1109/jiot.2020.2965898
64. Sanjay Sarma VO, Parasuraman R, Pidaparti R. Impact of Heterogeneity in Multi-Robot Systems on Collective Behaviors Studied Using a Search and Rescue Problem. In: IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR). Abu Dhabi, United Arab Emirates; 04–06 November 2020; Abu Dhabi, United Arab Emirates (2020).
65. Chatziparaschis D, Lagoudakis MG, Partinevelos P. Aerial and Ground Robot Collaboration for Autonomous Mapping in Search and Rescue Missions. *Drones* (2020) 4:79. doi:10.3390/drones4040079
66. López DS, Moreno G, Cordero J, Sanchez J, Govindaraj S, Marques MM, et al. Interoperability in a Heterogeneous Team of Search and Rescue Robots. In: *From Theory to Practice: Search and Rescue Robotics* (2017).
67. Condomines J-P, Zhang R, Larrieu N. Network Intrusion Detection System for UAV Ad-Hoc Communication: From Methodology Design to Real Test Validation. *Ad Hoc Networks* (2019) 90:101759. doi:10.1016/j.adhoc.2018.09.004
68. Choudhary G, Sharma V, You I, Yim K, Chen I-R, Cho J-H, et al. Intrusion Detection Systems for Networked Unmanned Aerial Vehicles: A Survey; 25–29 June 2018; Limassol, Cyprus (2018).
69. Basan E, Lapina M, Mudruk N, Abramov E. Intelligent Intrusion Detection System for a Group of UAVs. In: *Advances in Swarm Intelligence*. Cham: Springer (2021). p. 230–40.
70. Subbarayalu V, Venuslaus MA. An Intrusion Detection System for Drone Swarming Utilizing Timed Probabilistic Automata. *Drones* (2023) 7:248. doi:10.3390/drones7040248
71. Phadke A, Ustyenko S. Updating the Taxonomy of Intrusion Detection Systems. In: 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC); 12–16 July 2021; Madrid, Spain (2021). p. 1085–91.
72. Levitin G, Finkelstein M, Xiang Y. Optimal Abort Rules and Subtask Distribution in Missions Performed by Multiple Independent Heterogeneous Units. *Reliability Eng Syst Saf* (2020) 199:106920. doi:10.1016/j.res.2020.106920
73. Levitin G, Finkelstein M, Xiang Y. Optimal Abort Rules for Additive Multi-Attempt Missions. *Reliability Eng Syst Saf* (2021) 205:107245. doi:10.1016/j.res.2020.107245
74. Zhang Q, Fang Z, Cai J. Preventive Replacement Policies With Multiple Missions and Maintenance Triggering Approaches. *Reliability Eng Syst Saf* (2021) 213:107691. doi:10.1016/j.res.2021.107691
75. Zhao X, Dai Y, Qiu Q, Wu Y. Joint Optimization of Mission Aborts and Allocation of Standby Components Considering Mission Loss. *Reliability Eng Syst Saf* (2022) 225:108612. doi:10.1016/j.res.2022.108612
76. Zhu X, Zhu X, Yan R, Peng R. Optimal Routing, Aborting and Hitting Strategies of UAVs Executing Hitting the Targets Considering the Defense Range of Targets. *Reliability Eng Syst Saf* (2021) 215:107811. doi:10.1016/j.res.2021.107811
77. Guo J, Wang L, Wang X. A Group Maintenance Method of Drone Swarm Considering System Mission Reliability. *Drones* (2022) 6:269. doi:10.3390/drones6100269
78. Peng R. Joint Routing and Aborting Optimization of Cooperative Unmanned Aerial Vehicles. *Reliability Eng Syst Saf* (2018) 177:131–7. doi:10.1016/j.res.2018.05.004
79. Liu L, Yang J. A Dynamic Mission Abort Policy for the Swarm Executing Missions and Its Solution Method by Tailored Deep Reinforcement Learning. *Reliability Eng Syst Saf* (2023) 234:109149. doi:10.1016/j.res.2023.109149

Copyright © 2024 Phadke and Medrano. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.