Summer 2023

Resilience Model for Teams of Autonomous Unmanned Aerial Vehicles (UAV) Executing Surveillance Missions

Robert Koeneke
Embry-Riddle Aeronautical University, koeneker@my.erau.edu

Resilience Model for Teams of Autonomous Unmanned Aerial Vehicles (UAV) Executing Surveillance Missions

by
Robert D. Koeneke

A dissertation submitted to the Faculty of
Embry-Riddle Aeronautical University in partial fulfillment
of the requirements for the degree of Doctor of Philosophy in
Electrical Engineering and Computer Science

Embry-Riddle Aeronautical University
Daytona Beach, Florida
August 2023

Resilience Model for Teams of Autonomous Unmanned Aerial Vehicles (UAV) Executing Surveillance Missions

by
Robert D. Koeneke

This dissertation was prepared under the direction of the candidate's Dissertation Committee Chair, Dr. Radu F. Babiceanu, and has been approved by the members of the dissertation committee. It was submitted to the College of Engineering and accepted in partial fulfillment of the requirements for the Degree of Doctor of Philosophy in Electrical Engineering and Computer Science.

Radu F. Babiceanu, Ph.D.
Committee Chair

Eduardo A. Rojas-Nastrucci, Ph.D.
Committee Member

Omar Ochoa, Ph.D.
Committee Member

Bryan Watson, Ph.D.
Committee Member

Nabeel Yousef, Ph.D.
Committee Member

Radu F. Babiceanu, Ph.D. — Date
Interim Chair, Electrical Engineering and Computer Science

James W. Gregory, Ph.D. — Date
Dean, College of Engineering

Christopher D. Grant, Ph.D. — Date
Assocaite Provost of Academic Support

ABSTRACT

Teams of low-cost Unmanned Aerial Vehicles (UAVs) have gained acceptance as an alternative for cooperatively searching and surveilling terrains. These UAVs are assembled with low-reliability components, so unit failures are possible. Losing UAVs to failures decreases the team's coverage efficiency and impacts communication, given that UAVs are also communication nodes. Such is the case of a Flying Ad Hoc Network (FANET), where the failure of a communication node may isolate segments of the network covering several nodes.

The main goal of this study is to develop a resilience model that would allow us to analyze the effects of individual UAV failures on the team's performance to improve the team's resilience.

The proposed solution models and simulates the UAV team using Agent-Based Modeling and Simulation. UAVs are modeled as autonomous agents, and the searched terrain as a two-dimensional MxN grid. Communication between agents permits having the exact data on the transit and occupation of all cells in real time. Such communication allows the UAV agents to estimate the best alternatives to move within the grid and know the exact number of all agents' visits to the cells.

Each UAV is simulated as a hobbyist, fixed-wing airplane equipped with a generic set of actuators and a generic controller. Individual UAV failures are simulated following reliability Fault Trees. Each affected UAV is disabled and eliminated from the pool of active units. After each unit failure, the system generates a new topology. It produces a set of minimum-distance trees for each node (UAV) in the grid. The new trees will thus depict the rearrangement links as required after a node failure or if changes occur in the

topology due to node movement. The model should generate parameters such as the number and location of compromised nodes, performance before and after the failure, and the estimated time of restitution needed to model the team's resilience.

The study addresses three research goals: identifying appropriate tools for modeling UAV scenarios, developing a model for assessing UAVs team resilience that overcomes previous studies' limitations, and testing the model through multiple simulations. The study fills a gap in the literature as previous studies focus on system communication disruptions (*i.e.*, node failures) without considering UAV unit reliability. This consideration becomes critical as using small, low-cost units prone to failure becomes widespread.

ACKNOWLEDGMENTS

Foremost, I want to recognize Dr. Radu Babiceanu for his guidance, sustained support, and patience throughout my doctoral studies and the development of my dissertation. I sincerely appreciate the thoughtful comments and feedback from my Doctoral Committee members, Dr. Bryan Watson, Dr. Omar Ochoa, Dr. Eduardo Rojas, and Dr. Nabeel Yousef. The collective advice of these bright colleagues helped me overcome obstacles and make progress in my dissertation project. I also am sincerely grateful to Dr. Eduardo Divo, for persuading me to pursue this goal. I couldn't have asked for a brighter and more dedicated group of colleagues.

I want to celebrate others who contributed to meeting my scholarly goals: my son Oscar A. Koeneke, who lent me his coding talents and insightful commentaries; my editor, Dr. Maruja Torres, for her assistance in structuring, formatting, and overall editing work. And above all, I want to thank my wife, Virginia, who lovingly and patiently nurtured me every day of this process, along with my other children, Roberto and Clara, whose support helped me stay on course and cheered me all the way. I could not have achieved my goals without their love and support.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

2-D	Two-dimensional
3-D	Three-dimensional
ABMS	Agent-based Modeling and Simulation
ABM	Agent-based Modeling
ABS	Agent-based Simulation
ACO	Ant Colony Optimization
API	Application Program Interface
BA	Barabási-Albert
C2	Command Control
CAGR	Compound Annual Growth Rate
DES	Discrete Event Simulation
FANET	Flying Ad-hoc Network
FIT	Failure in Time
GCS	Ground Control System
GPS	Global Positioning System
ICAO	International Civil Aviation Organization
IDE	Integrated Development Environment
ISR	Intelligence, Surveillance, and Reconnaissance

LTL	Linear Temporal Logic
MANET	Mobile Ad-hoc Network
MAV	Micro Aerial Vehicle
MSER	Marginal Standard Error Rule
MTBF	Mean Time Between Failures
MTTF	Mean Time To Fail
PA	Preferential Adaptation
POMDP	Partial Observable Markov Decision Process
RD	Recalculated Degree
RD	Random Detachment
RDA	Random Detachment Adaptation
SAR	Search and Rescue
SME	Subject Matter Expert
SMT	Satisfiability Modulo Theory
SNR	Signal-to-Noise
SoS	System-of-Systems
TBRS	Tracking Bigraphical Reactive System
UAV	Unmanned Aerial Vehicle
VANET	Vehicular Ad-hoc Network

LIST OF TABLES

LIST OF FIGURES

SECTION 1

GENERAL INFORMATION

1.1. Motivation and Purpose of the Study

The use of Unmanned Aerial Vehicles (UAV) systems or teams has grown recently, particularly for military missions, given their ability to reach a common goal through collaborative team effort and carry out tasks in places inaccessible or dangerous to humans (Li *et al.*, 2018). Skoroboagatov *et al.* (2020) acknowledge several significant benefits of using multiple over individual UAVs: time efficiency, simultaneous actions, complementarity, fault tolerance, and flexibility.

Considering the threat of the proliferation of accurate ballistic and cruise missiles, Hamilton & Ochmanek (2020) suggest employing large numbers of relatively low-cost, reusable, and expendable UAVs into the battle space. The aim is to saturate or exhaust the defense. Using teams of low-cost UAVs has been proposed in the military as a preventive measure on the assumption that the fault tolerance in UAV teams is higher than that of individual units when executing equivalent missions (Hamilton & Ochmanek, 2020). However, questions on the reliability of low-cost UAVs abound, given the UAV failure rate of $1/10^3$ flight hours against that of commercial aviation at $1/10^5$ flight hours, a two-order higher magnitude (Petritoli, Leccese, & Ciani, 2018).

Existing literature associates small and low-cost UAVs, based on their reduced dimensions, limited range, and lesser cost (about $10,000 as estimated in 2012). An example is the US Army RQ-7 Shadow (Abdullah, n/d), used by the Brigade Combat

Team. It has a 20 ft. wingspan, an approximate 60 lb. payload capacity, and an endurance of 9 hours from take-off to landing.

Murtha (2009) suggests that the high failure rate of low-cost UAVs stems from their short, imperfect design cycles and a tendency to sacrifice redundant systems for cost savings. Smaller, usually lower-cost, UAVs carry limited equipment to reduce power consumption and overall aircraft weight (Phadke & Medrano, 2022). For this reason, they typically do not feature hardware redundancy because of size, weight, power, and budget constraints (Freeman, 2014). Nonetheless, studies testing low-cost UAVs' performance against larger, more complex UAVs in diverse surveyance applications (*i.e.*, agriculture, topography, hydrology) have found them comparable (Grenzdörffer & Teichert, 2008; Koutalakis, Tzoraki, & Zaimes, 2019). Cook (2017) further questions whether increased UAV sophistication translates into improved results quality.

Alongside growth, UAV applications have become more complex, and the missions have gradually evolved into multiple-UAV (team) missions (Li *et al.*, 2018). However, in teams, each unit loss progressively degrades the team's effectiveness. As individual units become disabled, the number of operative units continues to diminish, thus eroding their ability to cooperate within the group until the team can no longer achieve its mission objectives. Disruptions to the UAV team's activities require actions to minimize the effect of the failure of individual UAVs on team performance and availability and to reorganize and redistribute resources that fix the disruption and allow successful mission outcomes.

Tran (2015) discusses two approaches for successful mission completion: designing for robustness and for resilience. He defines robustness in the context of a System of Systems (SoS) as "the reduced sensitivity of SoS performance to variations in individual system performances that could potentially generate cascading effects across an SoS network" (p.11). The standard means to achieve robustness is overdesigning the system to reduce its probability of failure. This entails using high-reliability components or materials offering high performance and reduced system uncertainty, which incur additional costs. However, overdesign and redundancy are no longer affordable for organizations facing increasingly strict budgets and pressure for cost efficiency. Instead of focusing on designing an SoS insensitive to failure, a better approach may be to assume that system failure will occur at some point. A widespread acknowledgment is that eventual losses within an SoS are unavoidable due to their complexities and emergent behaviors, regardless of any cautionary measures taken. Alluding to the team's resilience, Tran (2015, p.12) concludes, "Therefore, designers should instead focus on how an SoS will adapt to failures while using remaining operational systems."

To this end, knowing the resiliency of the UAV team is highly desirable in the planning stages of an operation. Operation planners would have information (mission type, topography of the terrain, number of units available for the task, and unit reliability) to estimate the potential impacts of a UAV team's disruption and, thus, its prospects for mission success. However, no framework currently exists to assess the team's resilience based on the reliability of the individual units. This framework would pave the way for implementing necessary adjustments to reach the desired outcomes. The purpose of this study is to fill the void of this lack of framework.

1.2. Context

Increasing advances in drone technology have driven the growth of the unmanned aerial vehicle (UAV) global market from $10.72 billion in 2019. The UAV market is projected to reach $25.13 billion by 2027, with a Compound Annual Growth Rate (CAGR) of 12.23%. Notably, the small UAV market segment is forecasted to be the fastest-growing, driven by demand for small military drones for Intelligence, Surveillance, and Reconnaissance (ISR) applications. Similarly, the highest CAGR is projected for fully autonomous UAVs, whose software systems refinements enhance the drones' capability for detecting real-time airspeed, altitude, and position for warfare missions, logistics & transportation, and disaster relief operations. (Fortune Business Insights, 2023). Other forecasts expect the global UAV payload market to reach $3 billion by 2027 (Shakhatreh, 2019), with the payload being all equipment carried by UAVs, such as sensors, cameras, lidars, etc.

Unmanned civilian aircraft such as UAVs are subject to Article 8 of the Convention on International Civil Aviation (Doc 7300), signed by the International Civil Aviation Organization (ICAO) and amended by its Assembly. It states that, with no persons on board the aircraft, the airworthiness (*i.e.*, suitability for safe flight) objective primarily focuses on protecting people and property on the ground (Bestaoui, 2018). The Advisory Circular (AC) (2020) developed for ICAO Member States mandates that when operating UAVs above populated areas, applicants should address, among others, the UAV and control system's reliability and have mitigation measures in case of a system failure.

A unit may experience hardware or software malfunctions, system bugs, or communication failures that can compromise its performance, disable it, or even cause it to crash. A preventative approach to such failures entails considering UAV unit reliability. Blanchard & Fabrycky (2011, p.262) define reliability as "the ability of a system to perform its intended mission when operating for a designated period or through a planned mission scenario (or series of scenarios) in a realistic operational environment." Reliability is often expressed as the probability of success measured in terms of Mean Time Between Failure (*MTBF*), Mean Time To Fail (*MTTF*), Failure Rate (λ), or a combination of these.

At the teams' level, the failure of a UAV unit would cause a communications disruption, a significant factor in mission success given the interdependency of current multiagent systems (Phadke & Medrano, 2022). Therefore, beyond individual unit reliability, the success of a UAV mission often depends on its *resilience* or the ability of the team to quickly and effectively respond to unexpected events, such as equipment malfunctions, adverse weather conditions, unexpected obstacles, or communication disruptions.

Resilience is a broad concept applied throughout multiple disciplines; it defines how well a system handles disruptions during normal functioning (Phadke & Medrano, 2022). More precisely, resilience is the ability of an information system to continue to: (i) operate under adverse conditions or stress, even if in a degraded or debilitated state, while maintaining essential operational capabilities; and (ii) recover to an effective operational posture in a time frame consistent with mission needs (Information Technology Laboratory, n/d).

A resilient UAV team can improve mission success by adapting to changing environments and maintaining its operational effectiveness in the face of unexpected events. Reliability and resiliency concepts are critical to a UAV team's mission outcomes and, as such, are the focus of this study.

1.3. Research Question and Goals

This study proposes a framework incorporating individual UAV reliability in a team model under a Flying Ad Hoc Network (FANET) structure, with unit failures modeled using representative scenarios of UAV environments. We can use this model to evaluate the impact of unit losses on their team performance, assess the team's resilience, and propose mitigation measures to restore the team's communication and complete the mission. Since we use representative scenarios based on realistic elements, the resulting model is more realistic and behaves more accurately than known models for UAV swarm resilience evaluation, such as those from Tran, Balchanos, *et al.*, 2016; Tran *et al.*, 2016; Tran *et al.*, 2015; Bai *et al.*, 2020; Petritoli *et al.*, 2018; and Jakaria & Rahman, 2018 in which the team's failures are modeled by randomly detaching network nodes.

Therefore, the overall research question underlying this study is, how can we design a framework that assesses the resilience of a UAV team accounting for unit reliability?

The goal of developing a model for mitigating system failure by considering the reliability of individual units and their impact on mission success drives this study. The study met three overarching research goals to answer the posed question:

Research Goal 1. Identify appropriate tools for modeling UAV scenarios. To build the model, apply the Reliability and Resilience Theory principles, tools for Reliability Calculation, Fault Trees, UAV technology, and Agent-Based Modeling and Simulation.

Research Goal 2. Develop a model for assessing UAV team resilience that overcomes the limitations of previous studies. Previous studies approach the model's resilience by using random detachment of nodes. A more realistic model is possible as data is available for modeling the removal of the nodes from the team by the incidence of failures in individual UAVs.

Research Goal 3. Validate the model. Agent-based simulations are difficult to validate (Klügl, 2008), where validation determines whether a simulation model accurately represents the original. Klügl presents a validation framework for agent-based simulation that includes several strategies: Face Validation, Sensitivity Analysis, Calibration, Plausibility Check, and Statistical Validation.

1.4. Contributions

The proposed model for UAV teams subject to failures simulates the impact of individual UAV failures on the team communication performance and allows identifying measures to mitigate the disruption and continue with the mission. It addresses the current lack of a resilience model that assesses the resilience of UAV teams subject to individual UAV failures.

The study fills a gap in the literature as previous studies focus on system communication disruptions (*i.e.*, node failures) without considering UAV unit reliability. This consideration becomes critical as using small, low-cost units prone to failure becomes widespread. As noted earlier, the UAV market trends toward increased use of autonomous, small/low-cost UAVs, especially their use in swarms for military and disaster relief missions. This trend implicitly acknowledges the unavoidability of UAV unit failures. It also suggests the importance of considering their reliability and creates the opportunity and desirability for this study.

Providing a tool for assessing UAV system resilience based on the reliability of its constituent units yields the following potential contributions to the community of UAV technologists:

a. The model would simulate a UAV team's performance on a search or surveillance mission and calculate its resilience. When done in the early stages of the operation design, it allows implementing any necessary changes to the system and its components to increase team resilience and thus approach the mission with greater confidence.

b. The model offers a framework for modeling and simulating UAV teams' behavior under multiple conditions, such as the number of units, size of the terrain (grid), probability of failure of each unit, and maximum distance between units to maintain communication links, among others. Such a framework would allow UAV team designers to discover the underlying collective behavior of the system and therefore design proactively.

c. The proposed allows for assessing the financial implications of a UAV team configuration. By simulating a proposed team's performance and ability for fault recovery (resilience), decision-makers can explore cost-effective options to complete the intended mission within a budget.

More specifically, the study contributes to the state-of-the-art knowledge of UAV system resilience. The collective behavior of the UAV team is challenging to describe and model using analytical tools. It is generally a complex system with multiple parts interacting and influencing each other. In some cases, it is tough, if not impossible, to produce a model that analytically describes all interactions of their components. On the other hand, agent-based modeling and simulations such as those proposed here take a bottom-up approach. They allow describing the behavior of a complex system, such as a team of autonomous UAVs, by modeling each agent engaged in the collective behavior.

Although this study does not consider learning and adaptation in agent behavior, using an agent-based modeling framework for UAV teams offers a means to estimate the effect of eventual changes on team performance. Such a framework would apply when the behavior of individual units is subject to changes due to hardware and software modifications, potential improvements, or failure of any component. It would allow evaluation of the effect of changes and alterations to the network topology and hence, its requirements for changes in routing algorithm parameters used in the mobile network. Additionally, it would allow assessing the consequences of losing units on the team's performance.

This study fills a gap in the body of knowledge on UAV team resilience, as previous studies (*e.g.*, Tran, 2015; Tran *et al.* , 2017; Bai *et al.*, 2020) attribute randomness to unit failures, therefore not accounting for their reliability. The main contribution of this study is that it presents a means to measure the resilience of a team of small, low-cost UAVs in a way that captures the reality of the failures associated with these vehicles and thus yields a more accurate prediction than that offered by previous studies.

SECTION 2

BACKGROUND AND LITERATURE REVIEW

2.1. Unmanned Aerial Vehicles (UAV)

UAVs have advantages over ground robots, such as their capacity to fly while avoiding obstacles. (Bestaoui, 2020). Uses for UAVs are diverse in the military and civil environments: supporting public safety, search and rescue missions and disaster management, remote sensing, construction and infrastructure inspection, precision agriculture, delivery of goods, monitoring of road traffic, surveillance, and wireless coverage, and others.

Unmanned vehicles are classified by size as very small (30-50 cm), small (50 cm-2 m), medium (5-10 m), and large (larger than 10 m). They may also be classified depending on their military or civil use and their operation technology for fixed, rotary, and flappy wings. Finally, unmanned aerial vehicles can be either remote-guided or autonomous.

UAV systems, such as teams of remote-controlled and autonomous aerial vehicles, have gained significant attention, given their capabilities to carry out tasks in places inaccessible or dangerous to humans (Li *et al.*, 2018). Another noteworthy option is that of teams of UAVs working to reach a common goal through collaborative team effort. In fact, using multiple UAV teams instead of a single UAV has been growing recently, particularly for military missions.

Using multiple UAVs has advantages over the use of individual UAVs. Skoroboagatov *et al.* (2020) mention several significant benefits of using more than a few UAVs over an individual UAV: time efficiency, lower cost, simultaneous actions, complementarity, fault tolerance, and flexibility. They consider as a threat the proliferation of accurate ballistic and cruise missiles. Low-cost UAVs have been proposed in the military as a preventive measure: "The general answer to this class of threats is to put enough small UAVs into the battle space to saturate or exhaust the defense" (Hamilton & Ochmanek, 2020). Teams of low-cost UAVs are used for this purpose.

Although the fault tolerance in teams of UAVs is higher than the fault tolerance of individual units when executing equivalent missions. In teams, each unit loss progressively degrades the group's effectiveness. As individual units become disabled, the number of operative units continues to diminish, thus eroding their ability to cooperate within the group until the team can no longer achieve its mission objectives.

2.2. Agent-based Simulations (ABS)

Bonneau (2002) offers a practical view of Agent-Based Modeling (ABM): ABM is not a technology but a mindset where a system is described from the perspective of its constituent units. He stresses that though ABMs are easily implemented, people may wrongly assume that the concepts are easy to master when the idea behind ABM is profound. He discusses areas of application such as flows (traffic, evacuation from disasters, etc.). Flow management may simulate the traffic of individual vehicles on a regional transportation network and estimate air pollution emissions generated by vehicle movements. An example is the simulation of stock markets, as in the case of a project for NASDAQ in which their model allowed the regulator to test and predict the effects of different financial strategies. In the case of a theme park, Bonneau defines ABM as "the most natural and easiest way of describing the system."

Macal and North (2010) comprehensively overview agent-based modeling and simulation (ABMS). They explain the main ABS concepts, discuss some applications in multiple fields and disciplines, and identify methods and toolkits for developing agent models. They rely on agent-based modeling to prototype complex and adaptive systems dynamics. They explain that systems are modeled from the bottom-up: agent-by-agent and interaction-by-interaction. Self-organization can often be observed in such models. Patterns, structures, and behaviors not explicitly programmed into the models arise through agent interactions. Macal and North propose a methodology for modeling through a checklist of guidance questions for adopting an ABMS solution. He suggests relevant questions to know the specific problem to be solved with ABMS: the role of the agents in the model, entities, behaviors, agent environment, and model validation.

Law (2015) presents a description of Agent-Based Simulation (ABS) from an author's perspective with a traditional view of modeling and simulation, and points out the power of ABS for simulating complex systems. He thus considers that ABS is a variation of Discrete Event Simulation (DES) through examples of systems modeled under traditional DES and as ABS. Whereas Law discusses emergent behavior as a result of agents' interaction over time, he points out that a system does not need to show emergent behavior as a condition to model under ABS.

Abar *et al.* (2017) review literature and tools for Agent-Based Modeling and Simulation. They compare a total of eighty-five agent-based modeling and simulation tools for license requirements and availability; source code; type of agent based on their interaction behavior (reactive agents, mobile, belief-desired intention, deliberative, evolutionary, etc.); the programming language for Application Program Interface (API) for model development; Integrated Development Environment (IDE); Operating System platform and Implementation platform; and model development effort (Complex, Hard, Moderate, Simply, Easy). Abar *et al.* clarify that some of the tools mentioned in their survey never were implemented or stopped receiving support, which is a shortcoming of their study.

2.3. UAV Team Modeling

Cybulski *et al.* (2021) present "a method of modeling a UAV swarm with the addition of generating a behavior policy for swarm elements based on a constructed model." This method of modeling UAV swarms grounds on "bigraphs with tracking" following work by Milner *et al.* (2009). Their work takes a different approach in considering a tighter connection between the mission requirements and model elements and the limitation of using identical behavior for all swarm elements. The model generation is a bottom-up method that can be automated, starting with the mission requirements as bi-graphical diagrams and robot capabilities with non-adaptive behavior.

Cybulski *et al.* first define a UAV swarm mission as a Tracking Bigraphical Reactive System (TBRS) to create the model. Subsequently, the TBRS is transformed into state space represented as a directed multigraph in which the edges correspond to actions performed by swarm elements and vertices representing the states. A walk is a finite-length alternative sequence of vertices and edges from the initial state to the vertex representing the final state. Cybulski *et al.* provide some recommendations for future users of this method, particularly the dimensional growth of the systems.

Soon-Jo *et al.* (2018) produced a thorough overview of 239 influential sources on the modeling, control, planning, sensing, design, and implementation of aerial swarms. They address the challenges of transitioning from 2-D to 3-D and integrating autonomous aerial swarm systems with other types of robots. They consider different kinds of robots using a hierarchical approach, as these are prevalent in the machine learning and control fields. Their paper introduces the topic by providing a general consideration of swarming robots, swarm autonomy, the use of hierarchical architecture, and the relation of timescales on systems dynamics and control systems' properties.

Later, Cybulski *et al.* cover the stability and controllability of swarms, types of multiagent systems, models for dynamic swarm systems, physics-based models for robot agents, synchronization with leader following, leader selection and sensor placement, synchronization, and stability for swarms. They also discuss swarm trajectory generation and motion planning, simultaneous planning with distributed assignment, collision avoidance and collision-free motions, aerial manipulation, and external control of aerial swarms. Finally, they focus on target search and tracking, surveillance and monitoring, and cooperative aerial mapping. They also discuss platforms, vehicle power management, pose and state estimation, and communication infrastructure.

Wang *et al.* (2018) present an alternative system of autonomous UAV swarms used for Search and Rescue (SAR) operations, in which each UAV can switch behavior. The UAVs make the proper decision on which behavior model to adopt. The decision to select the behavior derives from rule-based architecture and depends on the environment. For the implementation, a central mission controller generates decision policies. This controller mission is usually given to the ground station, disseminating the rules each agent must observe. All UAVs share an initial working memory, including the world state and decision policies. Rules are specified precisely using Linear Temporal Logic (LTL) formulas. They create models for the behavior using the Partial Observable Markov Decision Process (POMDP). A version of the system is simulated with two agents.

Brust and Strimbu (2015) propose a solution for using swarm formation with a leader to attain high-quality forest mapping. It applies a leader election algorithm to a set of autonomous micro-UAVs. The leader unit gathers information from the swarm's collective and leads it to the destination. Additionally, the leader UAV controls communication with the base station. The UAVs in the swarm sense and register the events of the environment. The leader collects and processes information from them, reacting to avoid obstacles and acting for route planning and maneuvers. The leader unit is selected based on its characteristics. Each unit keeps track of its neighbor's position, allowing it to maintain its formation.

Brust and Strimbu's algorithm assigns "weights" to each one of the units. Nevertheless, the relation between the given weight and their use and relevance within the formation is unclear. Their proposed scheme was simulated with swarms of 4, 8, and 12 UAVs. For each simulation, the time to travel between two points ({25,25,25} and {100,100,100}) was observed and charted. Results show no time difference between 4 and 8 UAVs; nevertheless, the travel time was substantially higher for the case of 12 UAVs. They offer no report on their analyses of the results.

Skoroboagatov *et al.* (2020) offer a survey based on reviewing 87 papers discussing multiple UAV systems with an emphasis on real-life applications. The paper covers the advantages of multiple-UAV systems, such as time efficiency, lowered costs, the possibility of accomplishing simultaneous actions, complementarity, fault tolerance, and flexibility. At the same time, they discuss disadvantages, such as legal restrictions, piloting complexity, and safety issues. Skoroboagatov *et al.* also explain some UAV applications in real-life scenarios like video surveillance, photogrammetry, networks, traffic monitoring, load carrying, and search-and-rescue. Finally, they discuss a taxonomy for multiple UAVs, the data analysis of the reviewed literature used to carry out the publication, communication technologies used in UAV teams, and future trends for multiple UAV systems.

2.4. Autonomous Agents

An agent perceives its environment through sensors and acts upon that environment through actuators (Russel & Norvig, 2021). The agent function specifies actions in response to any percept sequence; its program implements the agent function of mapping from percepts to actions. So, typically, the agent runs in a computer device that interacts with sensors through input ports and issues the command signals to the corresponding actuators through output ports.

An agent is rational if, for each possible percept sequence, it selects an action to maximize its performance measures based on the evidence provided by the percept sequence and whatever built-in knowledge the agent has. The agent is autonomous if it gains knowledge from its percepts and not merely from the designer (Russel & Norvig, 2021).

UAVs are often used in robotics and modeled as agents. Their identifiable building blocks are sensors, percept sequences, agent programs, and actuators. Cameras, communication antennae, GPS, and accelerometers are part of the set of sensors in the payload of any ordinary UAV. An inboard computer (controller) reads the signals from sensors, and the controller processes them according to a specific module. It then outputs the signals to the corresponding actuators (servos, amplifiers, motors, etc.).

2.5. Multiagent Systems

In a team of autonomous agents, the agents perceive their environment not only through their sensors but also from information about the perceived environment of other agents that share or not their sequence of percepts. Agent-agent communication is required to enable and synchronize their interaction. Creating a multiagent environment instead of a single-agent environment results in substantial complexity—specifically, decision-making. What to do when more than one agent inhabits the environment will depend on the relationship among these agents (Russell & Norvig, 2021).

2.6. Multiagent Communications

Interaction among autonomous agents is required to achieve the goals, such as sharing goals and knowledge (sequence of percepts). Efficient and reliable communication is a dominant consideration if the system's response requires fluidity. Autonomous agents may be mobile; therefore, it is necessary to handle mobility without restricting the system's autonomy. Each agent may move at high speed and with unpredicted direction, making communication challenging. Agents themselves may actively participate in the communication infrastructure.

Communications requirements in environments of mobile multiagents have been resolved primarily using ad hoc networks. These networks emerge as a solution in cases where no infrastructure is available to fulfill the needs of communications between agents and between base stations and agents. Adding to Bekmeczi *et al.*, Guillen-Perez and Cano (2018) differentiate the three major approaches for implementing ad hoc networks: MANET, for mobile networks, which directly connect mobile devices such as cell phones, laptops, sensors, etc.; VANET, for vehicle networks serving automobiles, buses, ambulances, etc.; and FANET, networks of flying vehicles. One of the features of FANETs is that the work allocated to them determines the mobility of the nodes (Wheeb *et al.*, 2022).

2.7. Reliability and Resilience of UAV Teams

Teams of low-cost, autonomous, small, or micro UAVs have become an attractive alternative for surveillance, searching, area mapping, land photography, and other applications. Assembling these vehicles using low-cost components with uncertain reliability requires all necessary provisions to minimize the effect of the failure of individual UAVs on team performance and availability. When one UAV fails and is deactivated, it may create undesirable outcomes. For example, if an uncontrolled UAV were to crash into an inhabited territory and cause damage to persons or property. At the same time, the missing unit creates a void by losing data from the coverage of the corresponding area surveyed, which may create inconsistency in the team's state.

In some UAV teams, the units are also communication nodes. Missing units may disrupt the message flow, risking the completion of the mission for lack of coordination. For instance, if the dismissed UAV were in the middle of an information update being shared with other UAVs or if the unit was functioning as a communication node in an ad-hoc network used by the team. In both cases, necessary actions would be required to reorganize and redistribute resources to fix the disruption.

A resilience model would allow designing a system that maintains or recovers the team's capability after a disruption. In our scenarios, the capability corresponds to the total number of messages received in the network.

2.7.1. Elements of Reliability

Ross (2006) provides an example for simulating the reliability function, where he considers a system of n components in which each one is either functioning or failing. The equations that follow describe the calculation of the reliability function of a parallel system with n independent components; Figure 1 illustrates the difference between series and parallel structures.

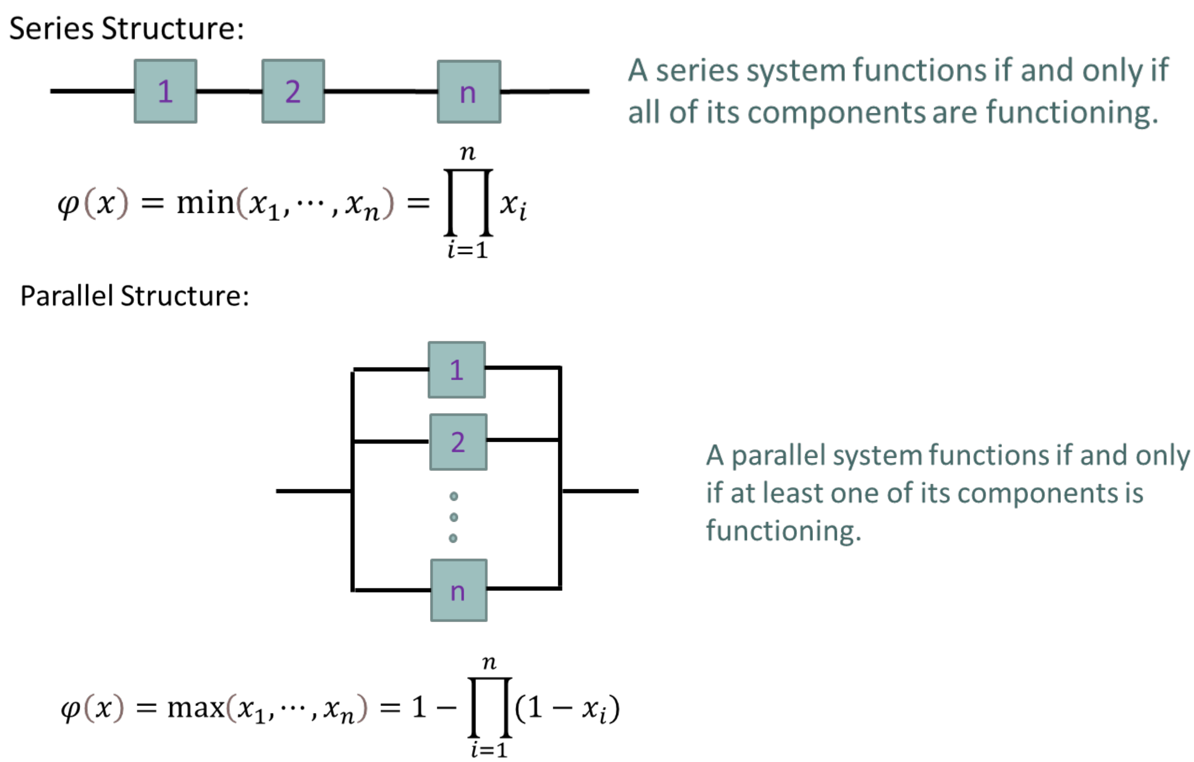

Figure 1. Reliability of series and parallel models. Adapted from Ross (2019).

For each component, Ross defines a Boolean function s_i that takes value = 1 if the component i works and zero if otherwise. Ross describes the state vector $s = (s_1, \ldots, s_n)$ and a nondecreasing function: structure $\varphi\ (s_1, \ldots, s_n)$ that similarly takes value = 1 if the system works under the state vector and 0 if otherwise. He refers to the most common structure functions, series structures, in which the system works if and only if all its components function, and a parallel structure, in which the system works if at least one of its components works.

Ross defines the k-out-of-n structure function. The reliability function $r\ (p_1, \ldots, p_n)$ represents the probability that the system works when the components are independent, with component i functioning with probability p_i where $i = 1, \ldots, n$. Finally, Ross defines the reliability function for series and parallel systems in terms of p_i. He shows a way to simulate the S_i by generating uniform random numbers $U_1, \ldots, U_n$ and comparing with p_i giving $S_i = 1$, if $U_i < p_i$ or 0, if $U_i > p_i$. System reliability refers to the probability that the system safely operates during the operational lifetime.

State vector $x=(x_1, \ldots, x_n)$

$$x_i = \begin{cases} 1, if\ the\ ith\ component\ is\ functioning \\ 0, if\ the\ ith\ component\ has\ failed \end{cases}$$

Structure function:

$$\varphi(x) = \begin{cases} 1, if\ the\ the\ system\ is\ functioning\ when\ the\ state\ vector\ is\ x \\ 0, if\ the\ system\ has\ failed\ when\ the\ state\ vector\ is\ x \end{cases}$$

Reliability of Systems of Independent Components:

$$P\{X_i = 1\} = p_i = 1 - P\{X_i = 0\}$$

The value p_i, which equals the probability that the *i*th component is functioning, is the reliability of the *i*th component (*r*).

$$r = P\{\varphi(X) = 1\},$$

r is the reliability of the system $where\ X = (X_1, \cdots, X_n)$

When the components, that is, the random variables Xi,i = 1,...,n, are independent, we may express *r* as a function of the reliabilities of the components.
That is

$$r = r(p), where\ p = (p_1, \cdots, p_n)$$

The function r(p) is the *reliability function.*

The reliability function of the series system of n independent components is given by:

$$r(p) = P\{\varphi(X) = 1\} = P\{X_i = 1\ for\ all\ i = 1, \dots, N\} =$$

$$= \prod_{i-1}^{n} p_i$$

The reliability function of the parallel system of n independent components is given by:

$$r(p) = P\{\varphi(X) = 1\} =$$

$$= P\{X_i = 1\ for\ some\ i = 1, \dots, N\} =$$

$$= 1 - P\{X_i = 0\ for\ all\ i = 1, \dots, n\}$$

$$= 1 - \prod_{i-1}^{n} (1 - p_i)$$

The complement of the reliability is scaled with the lifetime to obtain the average failure probability per hour. Besides reliability, mean time to failure (*MTTF*) is a standard measure of interest. Degraded states consider systems in which some faults have already occurred. If the system is degraded, it still safely operates but provides reduced functionality. Figure 1 above shows the mathematical relationships among the elements of reliability as a sequence of operations describing a parallel structure and ending with the reliability function calculation in a parallel system with independent components.

2.7.2. Reliability Model Precedent

Koeneke, Babiceanu, & Seker's (2019) "Target Area Surveillance Optimization with Swarms of Autonomous Vehicles" study is a precedent for this study. It considered a team (swarm) of UAVs modeled as a multiagent system surveilling a hostile territory modeled as a discrete search area, mapped as a rectangular grid. A modified ACO (ant colony optimization) algorithm selected the trajectory of the UAV agents during their surveillance mission, with the searching criteria prioritizing the least visited path. Given their communication capabilities, UAVs broadcasted and received each other's positions in the grid. A separate communication channel updated the results of the surveilled locations to a central station.

Swarm agent processors ran cooperation algorithms specific for distributed multiagent systems, which allowed updating the total visit count for all cells in the grid at any time. Having real-time data on the transit and occupation of all cells enables UAV agents to estimate the best alternatives to move within the grid. UAV agents prevent a collision by broadcasting their positions to the rest of the swarm before moving to the next cell. A conflict resolution protocol is triggered if multiple agents attempt to move to the same cell. Upon completing the mission, the UAV swarm would fly to a gathering point where the data is collected and assembled, and the battery re-charging process for the next task is started.

The algorithm resulting from this study assumed that the cost of UAVs would be kept at a minimum and did not consider any UAV redundancies due to component costs. However, Koeneke, Babiceanu, & Seker's algorithm relies on the idea that agent reliability is critical to maintaining the team's availability and operativity and, thus, is optimally suited for modeling UAV behavior.

2.7.3. Network Threats and Adaptation

Tran *et al.*'s 2015 ABS approach for evaluating the agility of adaptive C2 (Command Control) networks, "Evaluating the agility of adaptive command and control networks from a cyber complex adaptive systems perspective," is applied to a hypothetical military scenario where UAVs are assigned to maintain surveillance over enemy and neutral agents in a defined battlefield. The model uses random and targeted node removals to model threats in C2 networks since this method enables consideration of different network threats. A node removal can represent a targeted cyber-attack on a crucial node, random failure, or physical damage to a node. It is assumed that node attacks and failures result in total loss of functionality, causing the node and its links to be removed from the network when attacked. Targeting by recalculated degree (RD) removes nodes with the highest degree at each attack, updating the degree of all nodes once the network structure is changed.

Network adaptation is modeled by allowing nodes to rewire links randomly following a node removal event. Only links disconnected by the most recent node removal can be rewired. A time delay between when a node is removed and when the network adapts is implemented to represent the time it may take to decide how to adjust and rewire existing links. Rewired nodes choose new neighbors randomly; if a node is already connected to all other nodes, that node does not rewire its link. Network researchers have considered similar defensive or adaptive mechanisms to improve network resilience; however, most of these studies either pre-emptively rewire links or randomly restitute disconnected links anywhere in the network.

Tran *et al.*'s study considers a military scenario in which targeted nodes are simulated by random removal. The model proposed in this study entails a non-adversarial scenario in which the individual units' failures' effects compromise UAVs. These failures are modeled using a Fault Tree Reliability Model, in which the manufacturer provides the probability of failure of each component. The rewiring of nodes follows a selection of a minimum distance tree given by Dijkstra's shortest path. As a result, the expectation is to have a model with higher realism and effectiveness than the model in which nodes are removed and rewired randomly.

2.7.4. Resilient vs. Robust System-of-Systems

In "A Network-based Cost Comparison of Resilient and Robust System-of-Systems" (Tran *et al.*, 2016) 's scenario, a team of unmanned aerial vehicles (UAVs) is tasked with maintaining surveillance over a set of adversaries on a specified battlefield. The UAVs are networked together to enable communications, primarily sharing known locations of adversaries. In this SoS network, nodes represent UAVs, and links represent data links. An agent-based model is used to simulate and compare the performance of the network designs. As in their previous study, the model contains three types of agents: UAVs, adversaries, and neutrals. Simulations include 20 agents of each type, all moving within a square battlefield split into 36 search grids. UAVs attempt to maintain awareness of the location (*i.e.*, current search grid) of other agents on the battlefield by sensing nearby agents and sharing information with teammates. Messages sent between agents are received or dropped; corrupted or false messages are not considered. Network attacks occur every 200-time steps in a simulation; network adaptation occurs 100-time steps after each episode.

C2 performance is measured with an awareness metric, $A(t)$, calculated using Shannon's information entropy. Awareness is normalized to be within [0, 1]. The metric is formulated such that a UAV with complete uncertainty of the locations of all other agents of interest at time t has an awareness $A(t) = 0$. In this context, total uncertainty means the UAV gives all other agents a 1/36 probability of being in each search grid. A UAV with an awareness $A(t) = 1$ would have complete certainty of the current grid of every agent of interest. The mean awareness of all UAVs is used as the performance data in resilience metric calculations. The scenario shows Rtotal increasing initial density due to the increase in node density.

This study compares two design approaches: for the robustness and resilience of UAV teams. Both methods of simulating failures and node rewiring are based on random disablement. At the same time, the proposed study focuses on resilience. It considers that the effect of individual unit failures disables UAVs. As indicated, failures are modeled using a Fault Tree Reliability Model, in which the manufacturer gives the probability of failure of each component. The rewiring of nodes follows a selection of a minimum distance tree given by Dijkstra's shortest path.

2.7.5. Performance-based System Resilience

Tran *et al*. (2017) use an information exchange network model to demonstrate the framework's applicability in "A framework for the quantitative assessment of performance-based system resilience." They use stochastic simulation under an information exchange network model to demonstrate the framework's applicability toward system design. The network model is based on Dodds, Watts, and Sabel's model for organizational networks to simulate information exchange in a network. The exchange is modeled by messages passed between source and target nodes along existing paths in the network. Each node in the network creates a new message with probability μ at every time step in the simulation. Once generated, messages are forwarded along the shortest path in the network to their target node.

Messages are passed from one node to a neighboring node in a single time step. Each node is assumed to have complete knowledge of the current network topology, allowing nodes to determine the shortest path from themselves to another node in the network. Their study uses the Barabási-Albert (BA) preferential attachment model to generate the initial topology. Potential network disruptions are modeled as node removal events, where nodes are removed uniformly at or in a targeted manner (for example, intentional network attacks). Targeted node removal is based on node degree, where the most connected nodes are removed every time. Actions considered in this study anchor on network adaptation, where nodes affected by a disruption rewire any disconnected links. Two adaptation strategies are considered: random rewiring and preferential attachment. With random rewiring, nodes randomly decide whom to rewire disconnected links. System performance, $y(t)$, is the total number of messages received in a network at each time step in a simulation.

As in other contributions from Tran *et al.*, the model used to simulate disruptions by node removal is random, and the recovery actions are done by random rewiring or preferential attachment-based adaptation. Although the authors recognize a weakness regarding threat probabilities, they do not propose any options to simulate them, a solution that this study offers.

2.7.6. Individual UAV Reliability

"Reliability and maintenance analysis of unmanned aerial vehicles," a study by Petritoli, E., Leccese, F., & Ciani, L. (2018), discusses a new logistic approach to UAV reliability and its relationship with maintenance. It emphasizes the advantages of completing the reliability study a priori as it can produce design recommendations. It presents actual reliability parameters that may be used as starting point to calculate the initial reliability of UAVs: Mean Time Between Failures (*MTBF*), Failure in Time (*FIT*), and Intrinsic Reliability. The study establishes the Reliability Assessment Hierarchy for UAVs for 10^3 failures. It identifies six critical systems in common drones: Ground Control System (GCS), Mainframe, Power Plant, Navigation System, Electronic System, and Payload. The fraction of failures from 10^3 total system failures is given for each of the main systems; each is further divided into subsystems, giving their respective portion of failures out of the total 10^3 failures.

Additionally, the study classifies the types of UAV failures (catastrophic, severe, moderate, and soft). It compares different maintenance philosophies for UAVs. It discusses the concepts of preventive and corrective maintenance that consider the system subjected to partial performance degradation (soft failure) until a hard failure occurs. Finally, by evaluating UAV uncertainty through the confidence interval, Petritoli *et al.* (2018) determine new soft failure limits, considering the general knowledge of the systems and subsystems to guarantee the proper preventive maintenance interval.

Petritoli *et al.*'s study provides critical insight into the reliability of individual drones. Although the figures of reliability given in the study are for commercial drones, their numbers may be helpful to obtain accurate MTBF in our model, which also does not consider maintenance other than battery replacement, given the limitation on costs (as the principle is the use of low-cost units). Soft failures are not considered in the proposed model as these are attributes of more expensive unit types.

2.7.7. *k*-resiliency for Collaborative UAVs

In "Formal analysis of *k*-resiliency for collaborative UAVs," Jakaria and Rahman (2018) propose a verification framework that automatically determines the resiliency of a UAV network and can find out unsafe or vulnerable UAVs in terms of control or connectivity requirements when several UAVs are unavailable (k) because of failures, accidents, or cyberattacks. The framework inputs necessary UAV parameters, such as current position, communication range, velocity, direction, fuel levels, encryption capabilities, etc. It formally models these parameters and the requirements and constraints to maintain secure communication. They solve the model using a Satisfiability Modulo Theory (SMT)-based formal verification engine. The results can determine whether the UAV network is resilient under the unavailability of several UAVs allowing a navigator UAV to navigate others safely.

The authors treat resilience differently from other related work and our proposal: Jakaria and Rahman define resilience as a condition where k units are unavailable. The remaining UAV units continue with the mission and create a safe network. They implement a model encoding the network configuration and the constraints into Satisfiability Modulo Theory (SMT) logic with a Z3 SMT solver (theorem prover). The solver checks the verification constraints and provides a satisfactory (SAT) result if all constraints are satisfied or (UNSAT) if the result is not satisfied. Their proposed verification framework is evaluated by running experiments with 10 – 50 UAV topologies. The authors do not consider recovery (adaptation). In contrast, the proposed study is interested in knowing the minimum number $(n - k)$ of UAVs required to complete a mission.

2.7.8. UAV Swarm Communication Limits and Resilience

Bai, G., Li, Y., Fang, Y., Zhang, Y. A., & Tao, J.'s (2020) "Network approach for resilience evaluation of a UAV swarm by considering communication limits" features a similar scenario as in Tran, Domerçant, and Mavris (2015) described above. The case study is based on a multiagent simulation using Anylogic, where a UAV swarm executes a surveillance mission over a controlled area. The battlefield area is modeled as a rectangular grid of $S = 500 \times 600$. The UAV swarm is fully self-organized and self-adaptive; therefore, its awareness and actions are based on local interactions among UAVs. Initially, N units are released from an airplane in a safety zone. The units regroup following an improved model for the initial network generation that is unlikely to generate isolated clusters or nodes.

Each UAV moves in a snakelike search pattern within the battlefield. They receive and send information about the locations of detected opponents and other UAVs via multi-hop wireless communication. Their model includes the maximum range of communications. Since the network topology is highly dynamic, and the relative positions of each pair of UAVs change frequently, the model checks when two nodes exceed the maximum communication range. Suppose the distance between the units exceeds the maximum range. In that case, the node is removed and then rewired by the host node using the preferential attachment probability function. If there are no nodes within the range (the probability of linking with another node is 0), the node stays in active status until other nodes move within its range of communications. The authors apply a modified resilience metric with the communication range as a parameter.

SECTION 3

A RESILIENCE MODEL FOR UAV TEAMS EXECUTING SURVEILLANCE MISSIONS

This study relies on empirical strategies to develop and test a quantitative model for UAV team communication recovery from individual UAV failures to improve team resilience. It uses computational tools and fault tree methods for reliability calculation. It entails modeling and simulation-based research that applies the principles of reliability and resilience theories and UAV technology. The study comprises three phases, resilience model development, testing, and validation. The testing phase entails iterations to allow for the refinement of the tool. The model validity phase intends to verify whether the proposed model has met its purpose.

3.1. Target Area Surveillance Algorithm

Unlike Bai *et al.*'s study (2020), which considers the communication range between UAVs, the one proposed assumes that the maximum distance in the grid is lower than the maximum range of communication. In their study, the nodes are removed and then rewired by the host node using the preferential attachment probability function. We use the same approach as Koeneke, Babiceanu, & Seker (2019) to move the agents within the grid. The units affected by faults are disabled, and the flow of messages is affected due to the loss of a UAV, a communication node. The absence of this communication node disrupts the flow of messages in the network.

Individual MAV agents broadcast their positions to the rest of the swarm agents before moving to the next cell, which minimizes the probability of collision. Receiving position coordinates from other agents allows for evaluating the risk of advancing to a potentially overlapping and dangerous collision course. Consequently, each agent can update the total visit count for all cells in the grid. After selecting the next neighbor cell to move, the agent broadcasts the intended move.

Each UAV agent in the swarm has, at any point in time, except when located on a border cell, eight degrees of freedom for the next visit. There are eight neighboring cells to any cells in the grid except for the border ones.

At time T, the agent moves from $c(i_{t=T\text{-}1}, j_{t=T\text{-}1})$ to $c(i_{t=T}, j_{t=T})$, or given the representation in Figure 2, the agent can move from $c(i, j)$ to either one of the following cells (Moore neighborhood): $c(i\text{-}1, j\text{-}1)$, $c(i\text{-}1, j)$, $c(i\text{-}1, j+1)$, $c(i+1, j+1)$, $c(i+1, j)$, $c(i+1, j\text{-}1)$, $c(i, j\text{-}1)$, or $c(i\text{-}1, j\text{-}1)$.

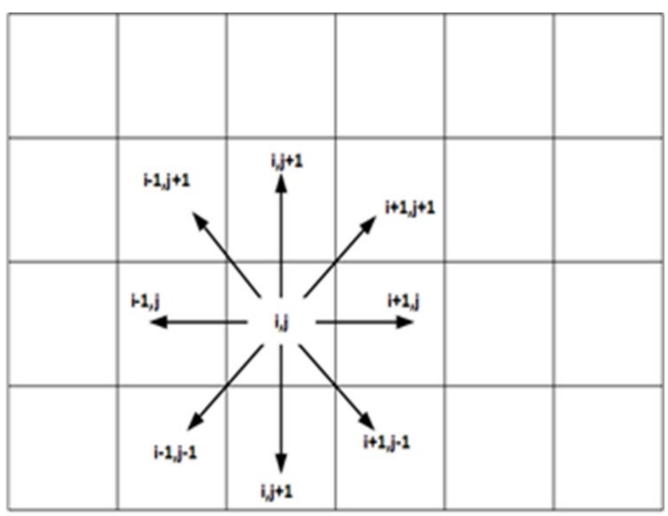

Figure 2. Agent movement at time T. Adapted from Koeneke, Babiceanu & Seker (2019).

Initially, each UAV is randomly assigned to a starting position cell in the grid. To choose the next cell to be visited,

1. Determine whether $c(i, j)$ is a border cell by calculating the row and column indices and checking the condition of exceeding or not the row (N) and column (M) sizes.

2. Sort-rank the table of grid cells such that their visit indices are updated in decreasing order.

3. Select the cell to be visited as having a lower visit index or the minimum visited cell index out of the candidate neighboring cells.

4. Choose randomly if more than one neighboring cell has the same visit index.

5. Broadcast updates and repeat the procedure if the selected next cell was already visited,

6. If contention exists among more than one UAV for the same cell, resolve the conflict by moving to another cell or remaining in the same one.

UAVs interchange messages with the rest of the team to support the surveillance application. Each node randomly decides on every tick to send a destination message to one or more UAVs. The probability of sending messages per node varies depending on the number of active nodes available, so traffic within the network remains constant as nodes fail.

The model keeps track of each message sent regarding the route followed (UAVs in transit), the time taken to reach the destination, and whether the message was delivered or failed. The course followed by each message corresponds to the least-cost route assigned using the Dijkstra algorithm. A matrix with the set of least-cost routes between a source node and its final destination node is kept in every node.

Similarly, a fault routine is run at each tick for each UAV. In the case of resulting unit failure, that unit is disabled. Once the rest of the UAVs realize the unit is unavailable, the failed unit is dismissed as an alternative. The model proceeds to reconfigure the connections with the rest of the UAVs. Considering the described model to quantify the number of messages the UAVs received as a performance measure, we can apply resilience metrics to evaluate the effect of failures on the number of messages received in the network and the network response to compensate for the disruption.

3.2. Reliability Model

One of the most critical aspects of the evaluation is to know whether the mission was successful. We should remember that the mission's main objective is to search or surveil terrain. A good metric for assessing whether the mission was successful is knowing the number of times a specific cell was searched. The reliability model keeps the count of the cells visited and the number of units disabled during each simulation time (tick). The model enables design trade-offs that may help in the design of missions by simulating scenarios with different numbers of UAVs, reliability, and allocation of time (number of ticks), as well as estimating the k-out-of-n figure to complete the missions.

This model considers each UAV an autonomous agent and the searched terrain as a two-dimensional $M \times N$ grid. Individual MAV agent broadcasting of positions to the remaining swarm agents before moving to the next cell minimizes the probability of collision. An agent's location in the surveilled area is known at any time by its computational unit, given its $c(i, j)$ designation. Each agent maintains and continuously updates an internal database with the number of visits to each cell in the grid. All agents broadcast their location to the rest of the team every time they move from one visited cell to the next. The rest of the team population records each agent's broadcast. Consequently, each agent can update the total visit count for all cells in the grid. Having the exact data on the transit and occupation of all cells in real-time allows the UAV agents to estimate the best alternatives to move within the grid.

The model uses Python-based Mesa as the agent-based simulation tool. One of the major advantages of using Agent-Based-Simulation is its bottom-up conceptualization of the scenarios. In a previous study, we used MATLAB to model an identical scenario. Using MESA for agent modeling in this current study enabled producing and simulating models significantly faster and easier than in our preceding study, which relied on MATLAB (Koeneke, Babiceanu, & Seker, 2019).

3.2.1. Individual Units Reliability Model

Each UAV is simulated as a hobbyist (low-cost), fixed-wing airplane like the Ultra Stick 120 used by Venkataraman *et al.* (2017). It is equipped with a generic controller and set of actuators, susceptible to failure. The assumption is that software faults are not present and only hardware, surfaces, and surface servo actuator failures would cause the dismissal of the unit. Figure 3 shows a block diagram of the hardware of a typical small autonomous UAV.

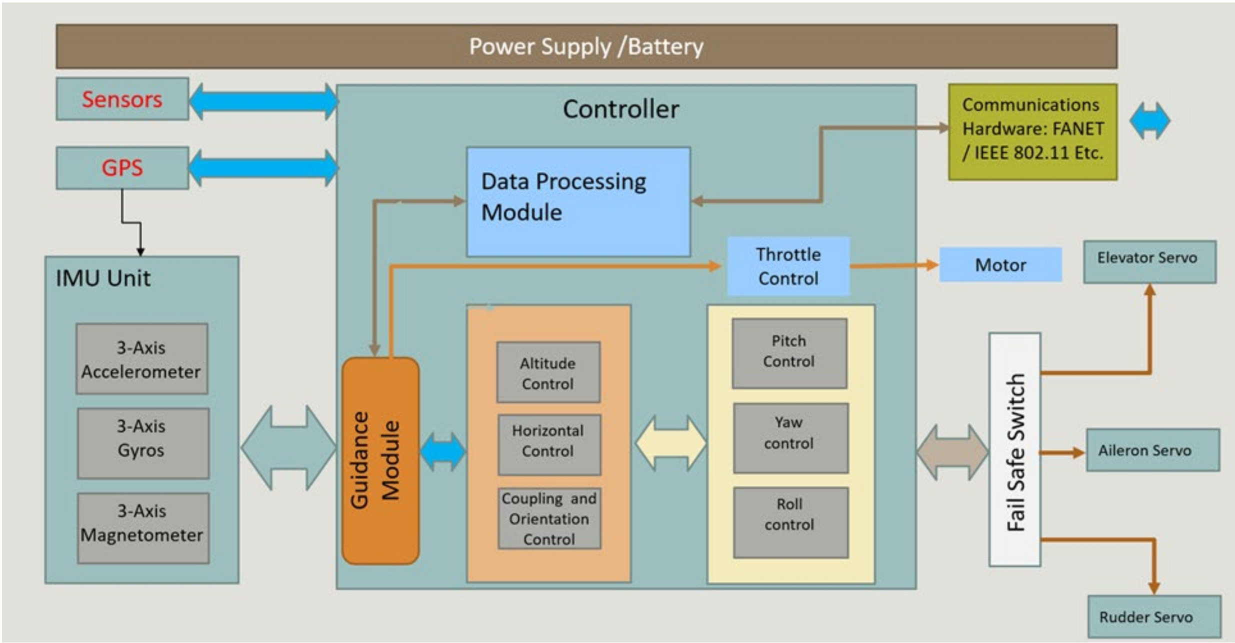

Figure 3. Block diagram of the hardware of a typical small UAV.

A total of 15 points of failures (components) are considered. Table 1 lists the components prone to failure and the approximate probability of failure for each one. Several providers of aero modeler components and suppliers of parts for airplane models were contacted asking for information on the reliability indexes, specifically *MTBF*, of the hobbyist airplanes like those used in this study. Given the scant information gathered from these sources, the probabilities of failure given in the table are estimations, as actual parameters of *MTBF* were not available at the time. Compensating for this, the values chosen for the simulation are extremely high compared to the usual values. The purpose of this is to pronounce the rate of team failure and, thus, to arrive more quickly at the simulation results.

Table 1: Components susceptible to failure and their probability.

No.	Component	Probability
1	Aileron Servo	0.04000
2	Aileron Coupling	0.03000
3	Aileron Surface	0.00500
4	Elevon Servo	0.02000
5	Elevon Coupling	0.01000
6	Elevon Surface	0.07000
7	Rudder Servo	0.09000
8	Rudder Coupling	0.04000
9	Rudder Surface	0.00800
10	IMU Failure	0.04000
11	Sensor Failure	0.02100
12	Motor Failure	0.01900
13	Communication Failure	0.14500
14	Power Supply	0.00400
15	Controller	0.00900

The criteria for composing the fault tree is that any single or multiple components faults are catastrophic. In other words, all components are critical. This is illustrated in Figure 4. Using the given probabilities, a program designed in MATLAB simulates the failures of the components. The program randomly generates a set of variates. It uses the probability of failure as the threshold for failure/no failure.

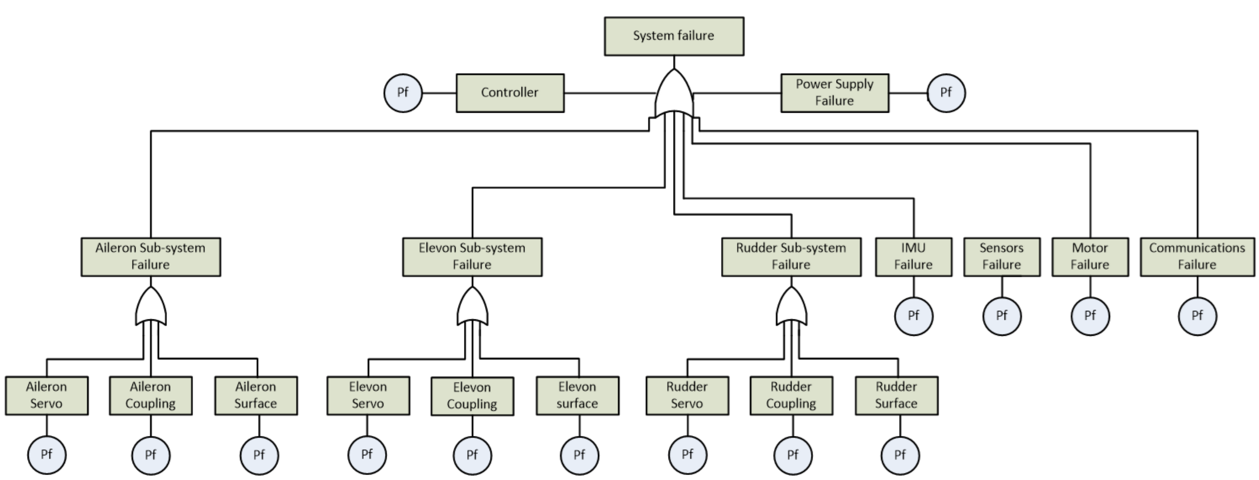

Figure 4. UAV fault tree.

Table 2. UAV failure simulation samples

`	Component	Probability	Faults in runs 0:fault ,1: no fault																	
1	Aileron Servo	0.04000	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	Aileron Coupling	0.03000	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	0
3	Aileron Surface	0.00500	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	Elevon Servo	0.02000	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	Elevon Coupling	0.01000	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	Elevon Surface	0.07000	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
7	Rudder Servo	0.09000	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
8	Rudder Coupling	0.04000	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
9	Rudder Surface	0.00800	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	IMU Failure	0.04000	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
11	Sensor Failure	0.02100	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
12	Motor Failure	0.01900	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13	Communication Failure	0.14500	1	1	0	1	1	1	1	0	1	1	0	1	1	1	0	1	1	1
14	Power Supply	0.00400	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
15	Controller	0.00900	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 2 shows a sample of 18 failure simulations for each UAV component. Each failure is depicted with "0" and not failure with "1." Figure 5 shows the total number of faults per component for 50,000 simulations.

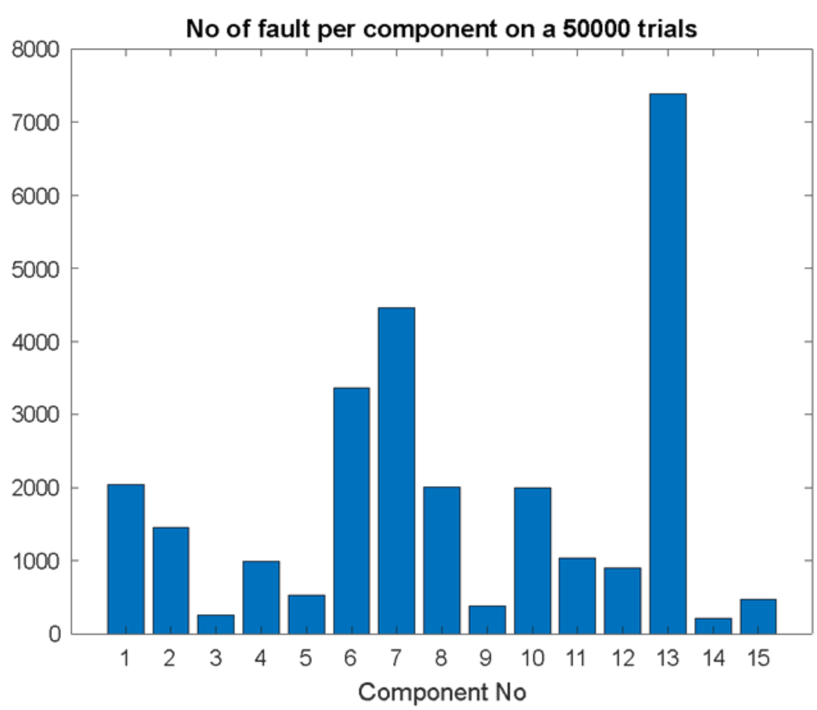

Figure 5. Number of faults per component after 50,000 simulations.

3.2.2. Modeling and Simulating UAV Team Failures

One of the most important aspects of the evaluation is to know whether the mission was successful in attaining the mission's primary objective: to search or surveil terrain. A good metric for assessing whether the mission was successful is knowing the number of times a specific cell was searched. The reliability model keeps the count of the cells visited and the number of units disabled during each simulation time (tick). The model enables design trade-offs that may help in the design of missions by simulating scenarios with different numbers of UAVs, reliability, and allocation of time (number of ticks), as well as estimating the *k*-out-of-*n* figure to complete the missions.

Recalling that the upper or team level is modeled and simulated as an agent-based system (ABS), the UAV teams in this work do not follow any formation or structure. Instead, each agent proceeds with its programmed behavior. There is available simulation software (Kopfstedt *et al.*, 2008; Luo & Sycara, 2019; Macal & North, 2010; Aggarwal, Kumar, & Tanwar, 2021), as well as proprietary software to approach an agent-based simulation (Ordoukhanian & Madni, 2019). Several alternatives have been evaluated to select a suitable agent-based simulation (ABM) that could quickly produce satisfactory results while addressing costs, flexibility, power, and simplicity. Some of the ABM tools compared were: SIMIO, MESA, Repast, Repast Symphony, and Repast HPC. The option MESA is based on Python and has enough flexibility to program the initial scenarios of this work. It was used to simulate the Team of UAVs and has provided satisfactory results in a short time.

The simulation scenario resembles the case of a surveillance mission covered in Klügl (2008), nonetheless in a non-adversarial territory. The UAV team will work collaboratively, flying over friendly territory modeled as an $N \times M$ grid. As the UAV moves to the next position, a fault is simulated for each of the 15 unit components. If a fault is present, the unit is disabled, and the incident parameters (Unit ID, time, grid position, and fault code) are reported.

The algorithm for the simulation is given in the pseudo-code below.

Algorithm: Agent-Based Simulation UAV Team

```
for k = 1 to MAX_NUM UAV                                /-initialize position agent-/
   next (i, j) = (i_k, j_k)
   perform other initialization steps
end
for t = t_1 to t_sim_max                                /-Simulation ticks-/
     for k = 1 to MAX_NUM UAV
        if k-th UAV is not disabled
           calculate the best next cell for moving      /- least visited -/
               if the next cell calculated is available
                   move to the next cell (i_next, j_next)
                 broadcast new cell location
                 apply fault induction routine      /-random fault generation-/
                 if not fault;
                     else disable k UAV
                 broadcast report for t_i
                 send messages to selected nodes
            else repeat with a new cell
```

else select the next UAV

end

end

On a per-tick base, the simulator first generates a failure routine for each one of the UAVs. Each UAV affected by a failure is disabled and eliminated from the pool of active units. Next, for the remaining active UAVs, the simulator estimates the grid movements and proceeds to move the UAVs to the new positions, where possible.

After a complete movement cycle, the system generates the new topology. It produces a set of minimum distance trees using Dijkstra's algorithm for each node (UAV) in the grid. The new shortest-path trees will thus depict the rearrangement links as required after a node failure or if changes occur in the topology due to node movement. The model generates parameters such as the number and location of compromised nodes, efficiency before and after the failure, and the estimated time of restitution needed to model the team's resilience.

The model takes a two-level bottom-up approach. A lower level, consisting of individual UAVs, is modeled as autonomous agents, *i.e.*, where the agents' software has programmed behaviors that give them the ability to decide or act within the simulation context, depending on the situations in which the agents find themselves. Individual hardware failure is represented as a Boolean parameter $H_{i,j}$. The upper level is modeled as a team of individual UAVs using Agent Base Modeling (ABM). Fault trees expose the failure probabilities of the critical components of the chosen model's hardware. Figure 6 shows a diagram of the two-level UAV Team Reliability Model.

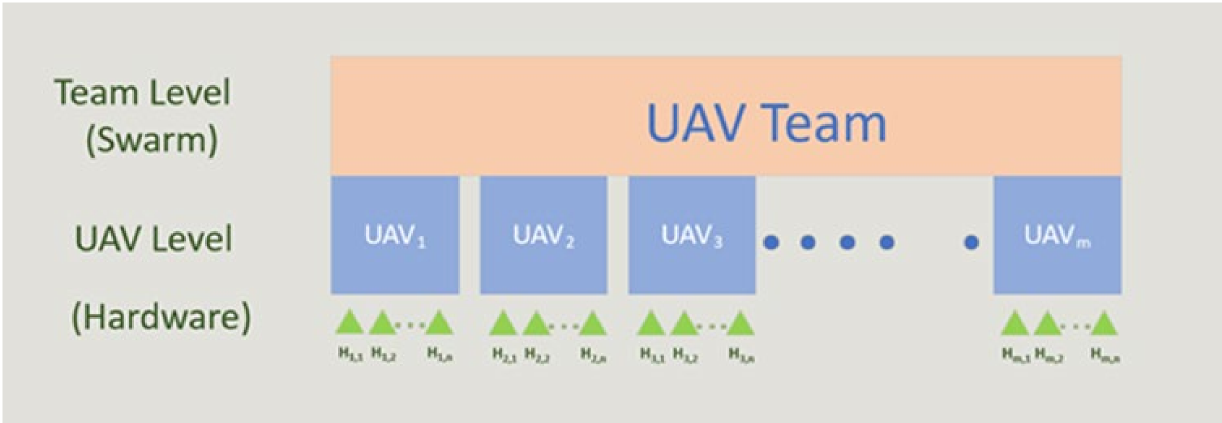

Figure 6. UAV Team reliability model.

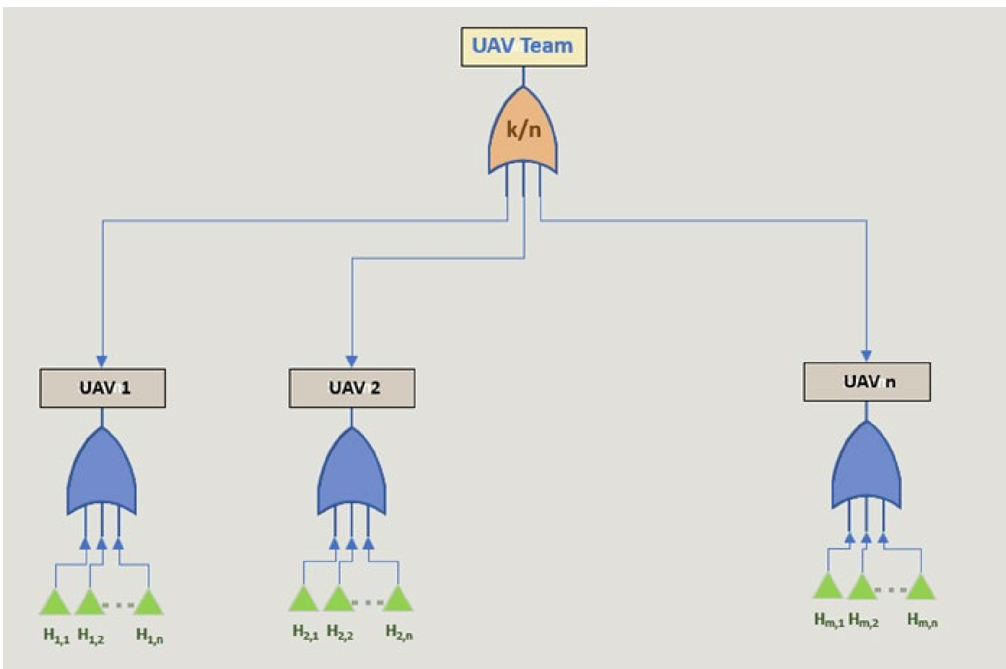

Figure 7. UAV Team *k*-out-of-*n* top node.

Figure 7 shows a fault tree approach to depict the team's reliability. The upper node of the tree is a k/n gate, which models a k-out-of-n structure that refers to a system if at least k out of the n UAVs are working, as Ross (2021) explained. The individual UAV reliability is modeled as all-or-nothing (series) in which all hardware modules of the UAV must work. Modeling faults in individual units follow this procedure:

Generate random reliability vectors $\boldsymbol{\psi}(X)$.

- A series model (from the fault tree) with fifteen components is used in this case.
- The method follows that suggested by Sheldon Ross (2006).
    - For each component, x_i of $\boldsymbol{\psi}(X)$ generates a random Uniform Distributed number U_i in the interval (0,1)
    - If $U_i > p_i$ (probability of x_i faulty) then $x_i = 1$ (no fault) otherwise $x_i = 0$ (fault)
    - p_i extracted from factor data
    - Vectors $\boldsymbol{\psi}(X)$ of 15 components
    - Failures of each component are independent of other components.

3.2.3. Reliability Trials

We ran the model in batches of 100 simulations to estimate reliability. We collected information about failures for different configurations (components). Each UAV had 15 components configured in a series structure. A series structure works only if all components function. For example, we simulated nine scenarios with the following variants: team sizes of 25, 35, and 40 UAVs, probability failure per UAV of 0.0249, 0.044, and 0.055. (The probability of failure per UAV is calculated as the sum of the individual component, as required in a series structure).

The following pages illustrate that each simulation corresponds to a batch of 100 models (simulations) for 150 ticks. The result of each simulation, depicted in Figures 8-16 and Tables 3-11, reveals the following parameters:

- Total number and percentage of UAVs disabled (failed)
- Total number and percentage of cells searched (visited)
- Averages of visits per cell
- Accumulated failure average of each component

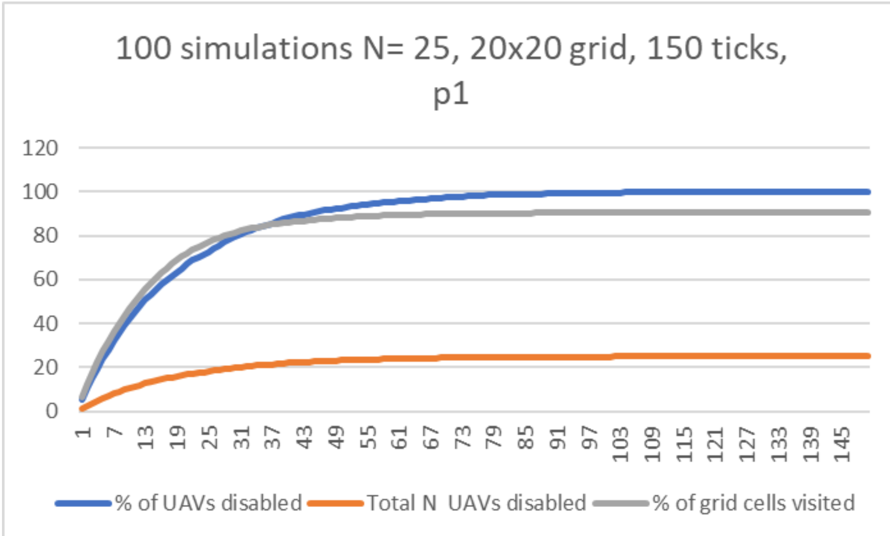

Figure 8. Trial 1: 100 simulations, N=25, 20x20 grid, 150 ticks, p1.

Table 3. Trial 1 data.

N_AGENTS PROBABILITY	25 0.055	TICK N º	% UAVS DISABLED	N º UAVS DISABLED	% CELLS VISITED	N º OF CELLS VISITED
grid_width	25	127	99.96	24.99	90.5425	362.17
grid_height	20	128	99.96	24.99	90.5425	362.17
n_ticks	150	129	100	25	90.5425	362.17
n_models	100	130	100	25	90.5425	362.17
part1_tolerance_level	100					
part2_tolerance_level	0.001					
part3_tolerance_level	0.0012					
part4_tolerance_level	0.0011					
part5_tolerance_level	0.0098					
part6_tolerance_level	0.006					
part7_tolerance_level	0.0067					
part8_tolerance_level	0.0013					
part9_tolerance_level	0.002					
part10_tolerance_level	0.0097					
part11_tolerance_level	0.0014					
part12_tolerance_level	0.0015					
part13_tolerance_level	0.001					
part14_tolerance_level	0.0012					
part15_tolerance_level	0.0098					
model_statistic_agg_method	0.0013					
time_in_seconds_per_model	average					
time_in_seconds_for_all_runs	0.162896					

Trial 1:

25 UAVs, grid 20x20 150 ticks

Probability of failure per single unit=0,055. The average number of visits per cell is 1.207875, measured in tick 149.

This model corresponds to the worse scenario: lowest number of UAVs (25) and highest probability of failure per unit.

In this model for 100 simulations, 100% of the 25 UAVs are disabled by the tick 129. As we can see the probability of completing the mission is very low. Observe that out of 100 simulations, none was 100% successful, leaving more than 9% of cells unsearched.

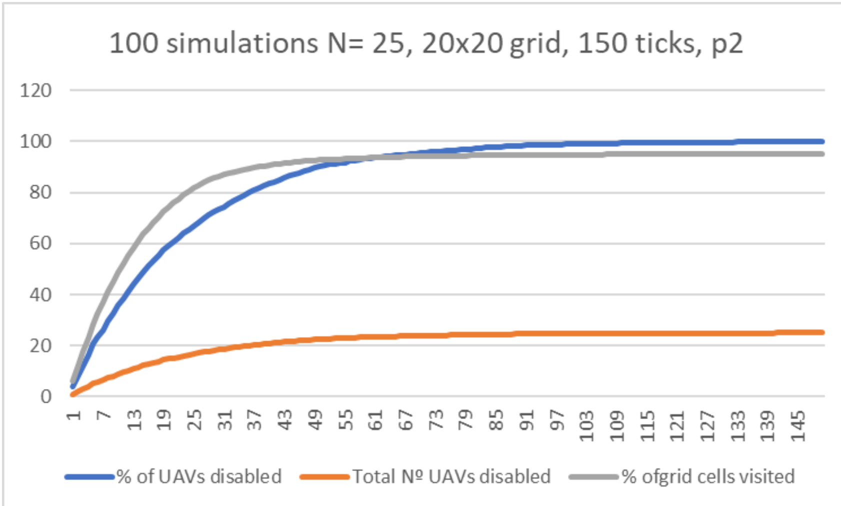

Figure 9. Trial 2: 100 simulations, N=25, 20x20 grid, 150 ticks, p2.

Table 4. Trial 2 data.

n_agents **probability**	**25** **0.044**	**Tick N º**	**% UAVs disabled**	**N º UAVs disabled**	**% cells visited**	**N º of cells visited**
grid_width	20	146	99.92	24.98	95.0275	380.11
grid_height	20	147	99.92	24.98	95.03	380.12
n_ticks	150	148	99.92	24.98	95.0325	380.13
n_models	100	149	99.92	24.98	95.035	380.14
part1_tolerance_level	0.001					
part2_tolerance_level	0.001					
part3_tolerance_level	0.0011					
part4_tolerance_level	0.001					
part5_tolerance_level	0.006					
part6_tolerance_level	0.005					
part7_tolerance_level	0.001					
part8_tolerance_level	0.002					
part9_tolerance_level	0.0097					
part10_tolerance_level	0.0014					
part11_tolerance_level	0.0015					
part12_tolerance_level	0.001					
part13_tolerance_level	0.0012					
part14_tolerance_level	0.0098					
part15_tolerance_level	0.0013					
model_statistic_agg_method	average					
time_in_seconds_per_model	0.164757					
time_in_seconds_for_all_runs	16.4757					

Trial 2:

25 UAVs, 20x20 grid cells

Probability of failure per single unit = 0,044. The average number of visits per cell is 1.424225, measured in tick 149.

Observe that in this case, 99.92 % of UAVs on disabled by the end of the 146-tick count. Although the percentage of completed mission is very low, there is an improvement with respect of trial 1 as the percentage of cells left unsearched is around 5 %, while in trial 1 is around 9.4%. In this case we are using the same number of UAVs, with an improved probability of failure per single unit of 0.044 (20% lower than in trial 1).

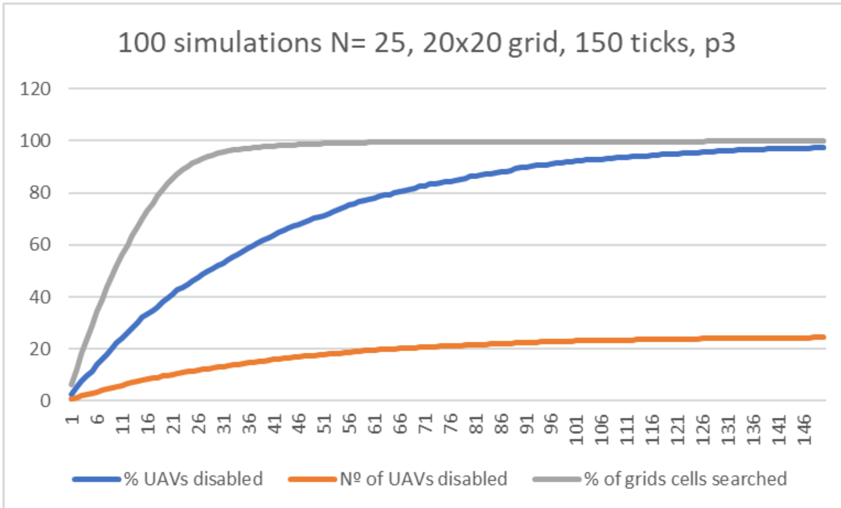

Figure 10. Trial 3: 100 simulations, N=25, 20x20 grid, 150 ticks, p3.

Table 5. Trial 3 data.

n_agents probability	25 0.0249	Tick N º	% UAVs disabled	N º UAVs disabled	% cells visited	N º of cells visited
grid_width	25	146	97.2	24.3	99.7925	399.17
grid_height	20	147	97.32	24.33	99.7925	399.17
n_ticks	150	148	97.4	24.35	99.7925	399.17
n_models	100	149	97.4	24.35	99.795	399.18
part1_tolerance_level	0.001					
part2_tolerance_level	0.0016					
part3_tolerance_level	0.0018					
part4_tolerance_level	0.003					
part5_tolerance_level	0.0013					
part6_tolerance_level	0.0023					
part7_tolerance_level	0.0017					
part8_tolerance_level	0.0013					
part9_tolerance_level	0.0025					
part10_tolerance_level	0.0022					
part11_tolerance_level	0.0012					
part12_tolerance_level	0.0012					
part13_tolerance_level	0.0012					
part14_tolerance_level	0.0012					
part15_tolerance_level	0.0011					
model_statistic_agg_method	0.0013					
time_in_seconds_per_model	average					
time_in_seconds_for_all_runs	0.210709					

Trial 3:

25 UAVs, 20x20 grid cells

Probability of failure per single unit=0,0249. The average number of visits per cell is 2.486225, measured in tick 149.

This model corresponds to the one with higher reliability for a team of 25 UAVs with a 99.7929 % to complete the mission successfully (100% of cells visited) thanks to an improved probability of failure per single unit of 0.0249 (20% lower than in trial 2).

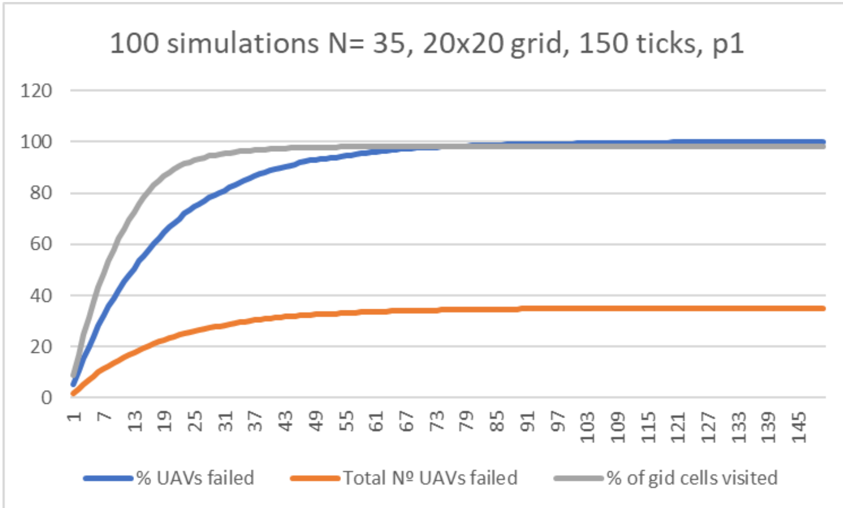

Figure 11. Trial 4: 100 simulations, N=35, 20x20 grid, 150 ticks, p1.

Table 6. Trial 4 data.

		Tick N º	%UAVs disabled	N º UAVs disabled	% cells visited	N º of cells visited
n_agents **probability**	**35** **0.055**					
grid_width	20	136	99.94286	34.98	98.33	393.32
grid_height	20	137	99.97143	34.99	98.33	393.32
n_ticks	150	138	99.97143	34.99	98.33	393.32
n_models	100	139	99.97143	34.99	98.33	393.32

part1_tolerance_level	0.0016
part2_tolerance_level	0.0018
part3_tolerance_level	0.003
part4_tolerance_level	0.0013
part5_tolerance_level	0.0023
part6_tolerance_level	0.0017
part7_tolerance_level	0.0013
part8_tolerance_level	0.0025
part9_tolerance_level	0.0022
part10_tolerance_level	0.0012
part11_tolerance_level	0.0012
part12_tolerance_level	0.0012
part13_tolerance_level	0.0012
part14_tolerance_level	0.0011
part15_tolerance_level	0.0013
model_statistic_agg_method	average
time_in_seconds_per_model	0.579457
time_in_seconds_for_all_runs	57.94572

Trial 4:

35 UAVs, grid 20x20, 150 ticks

Probability of failure per single unit = 0,055. The average number of visits per cell is 1.6608, measured in tick 149.

Trial 4: Comparing the outcomes of this trial with trial N º 1, in which we used 40 % less of UAVs with its same probability of failure, we may observe that the mission is 98.33% completed with 99.97% of UAVs disabled by tick 137, while in trial 1 only 90.54 % of the cells (362.17).

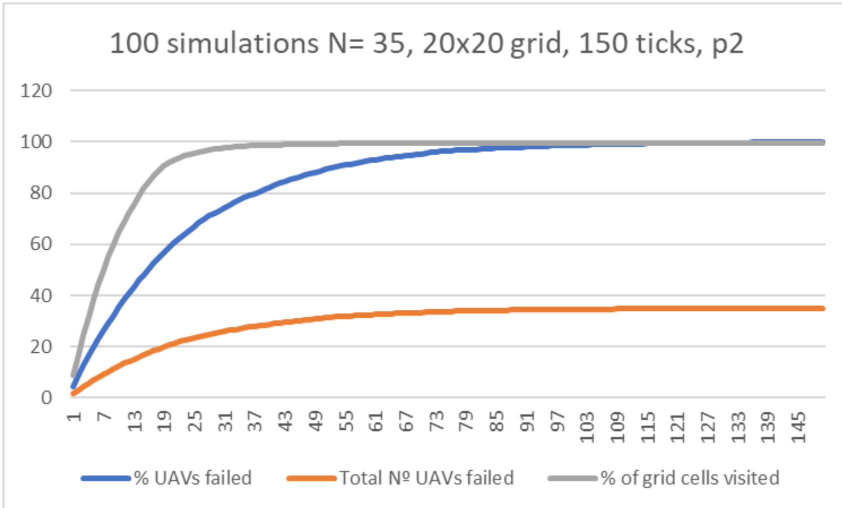

Figure 12. Trial 5: 100 simulations, N=35, 20x20 grid, 150 ticks, p2.

Table 7. Trial 5 data.

n_agents	**35**
probability	**0.044**
grid_width	20
grid_height	20
n_ticks	150
n_models	100
part1_tolerance_level	0.001
part2_tolerance_level	0.001
part3_tolerance_level	0.0011
part4_tolerance_level	0.001
part5_tolerance_level	0.006
part6_tolerance_level	0.005
part7_tolerance_level	0.001
part8_tolerance_level	0.002
part9_tolerance_level	0.0097
part10_tolerance_level	0.0014
part11_tolerance_level	0.0015
part12_tolerance_level	0.001
part13_tolerance_level	0.0012
part14_tolerance_level	0.0098
part15_tolerance_level	0.0013
model_statistic_agg_method	average
time_in_seconds_per_model	0.27036
time_in_seconds_for_all_runs	27.03597

Tick N º	**%UAVs disabled**	**N º UAVs disabled**	**% cells visited**	**N º of cells visited**
139	99.8	34.93	99.54	398.16
140	99.82857	34.94	99.54	398.16
141	99.82857	34.94	99.54	398.16
142	99.82857	34.94	99.54	398.16

Trial 5:

35 UAVs, grid 20x20, 150 ticks

Probability of failure per single unit = 0,044. The average number of visits per cell is 2.03845, measured in tick 149.

Observe that the mission is 99.54% (398.16 cells) successful with 40% more of UAVs of the same type of 35 UAVs used in trail 2, I which case the mission was completed with 95.035% of success.

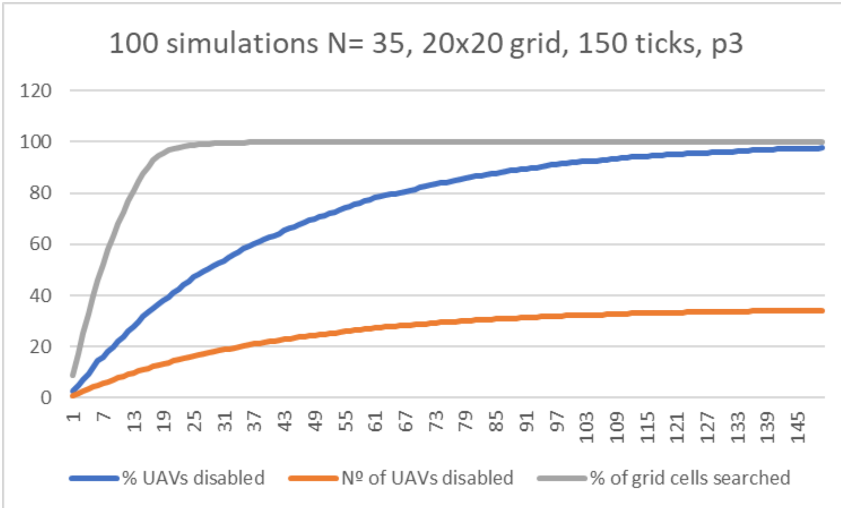

Figure 13. Trial 6: 100 simulations, N=35, 20x20 grid, 150 ticks, p3.

Table 8. Trial 6 data.

n_agents	**35**
probability	**0.0249**
grid_width	20
grid_height	20
n_ticks	150
n_models	100
part1_tolerance_level	0.0016
part2_tolerance_level	0.0018
part3_tolerance_level	0.003
part4_tolerance_level	0.0013
part5_tolerance_level	0.0023
part6_tolerance_level	0.0017
part7_tolerance_level	0.0013
part8_tolerance_level	0.0025
part9_tolerance_level	0.0022
part10_tolerance_level	0.0012
part11_tolerance_level	0.0012
part12_tolerance_level	0.0012
part13_tolerance_level	0.0012
part14_tolerance_level	0.0011
part15_tolerance_level	0.0013
model_statistic_agg_method	average
time_in_seconds_per_model	0.27036
time_in_seconds_for_all_runs	27.03597

Tick N º	%UAVs disabled	N º UAVs disabled	% cells visited	N º of cells visited
82	87.17143	30.51	99.995	399.98
83	87.42857	30.6	99.995	399.98
84	87.8	30.73	99.9975	399.99
149	97.6	34.16	99.9975	399.99

Trial 6:

35 UAVs, grid 20x20, 150 ticks

Probability of failure per single unit = 0,0249. The average number of visits per cell is 3.47765, measured in tick 149.

This result corresponds to the lowest probability of failure (0.0249) for a team of 35 UAVs. Observe that the mission is 99.9975% completed with 399.99 cell visited in average.

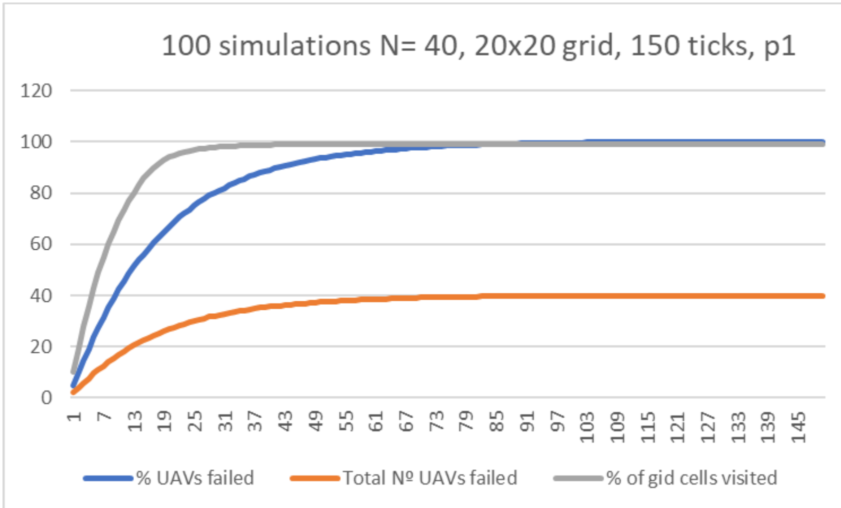

Figure 14. Trial 7: 100 simulations, N=40, 20x20 grid, 150 ticks, p1.

Table 9. Trial 7 data.

n_agents **probability**	**40** **0.055**	**Tick Nº**	**%UAVs disabled**	**Nº UAVs disabled**	**% cells visited**	**Nº of cells visited**
grid_width	20	146	99.975	39.99	99.1925	396.77
grid_height	20	147	99.975	39.99	99.1925	396.77
n_ticks	150	148	99.975	39.99	99.1925	396.77
n_models	100	149	99.975	39.99	99.1925	396.77
part1_tolerance_level	0.001					
part2_tolerance_level	0.0012					
part3_tolerance_level	0.0011					
part4_tolerance_level	0.0098					
part5_tolerance_level	0.006					
part6_tolerance_level	0.0067					
part7_tolerance_level	0.0013					
part8_tolerance_level	0.002					
part9_tolerance_level	0.0097					
part10_tolerance_level	0.0014					
part11_tolerance_level	0.0015					
part12_tolerance_level	0.001					
part13_tolerance_level	0.0012					
part14_tolerance_level	0.0098					
part15_tolerance_level	0.0013					
model_statistic_agg_method	average					
time_in_seconds_per_model	0.364482					
time_in_seconds_for_all_runs	36.44822					

Trial 7:

40 UAVs, grid 20x20, 150 ticks

Probability of failure per single unit = 0,055. The average number of visits per cell is 1.8658, measured in tick 149.

Observing that the mission is 99.8675% (with 399.47 cells searched) completed with 38.84 disabled UAVs.

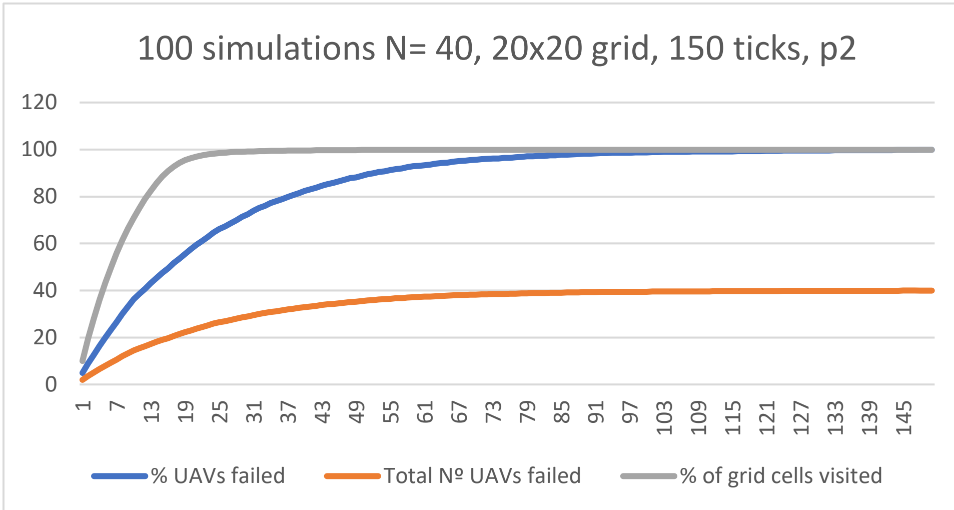

Figure 15. Trial 8: 100 simulations, N=40, 20x20 grid, 150 ticks, p2.

Table 10. Trial 8 data.

n_agents probability	40 0.044	Tick Nº	%UAVs disabled	Nº UAVs disabled	% cells visited	Nº of cells visited
grid_width	20	146	98.8	39.92	98.82	399.28
grid_height	20	147	98.8	39.92	98.82	399.28
n_ticks	150	148	98.8	39.92	98.82	399.28
n_models	100	149	98.85	39.94	98.82	399.28
part1_tolerance_level	0.001					
part2_tolerance_level	0.001					
part3_tolerance_level	0.0011					
part4_tolerance_level	0.001					
part5_tolerance_level	0.006					
part6_tolerance_level	0.005					
part7_tolerance_level	0.001					
part8_tolerance_level	0.002					
part9_tolerance_level	0.0097					
part10_tolerance_level	0.0014					
part11_tolerance_level	0.0015					
part12_tolerance_level	0.001					
part13_tolerance_level	0.0012					
part14_tolerance_level	0.0098					
part15_tolerance_level	0.0013					
model_statistic_agg_method	average					
time_in_seconds_per_model	0.391473					
time_in_seconds_for_all_runs	39.14733					

Trial 8:

40 UAVs, grid 20x20, 150 ticks

Probability of failure per single unit = 0,044. The average number of visits per cell is 2.3328, measured in tick 149.

In this trial the mission is 98.85% completed with 399.28 cells visited. Ends the with 1.15% of UAVs active.

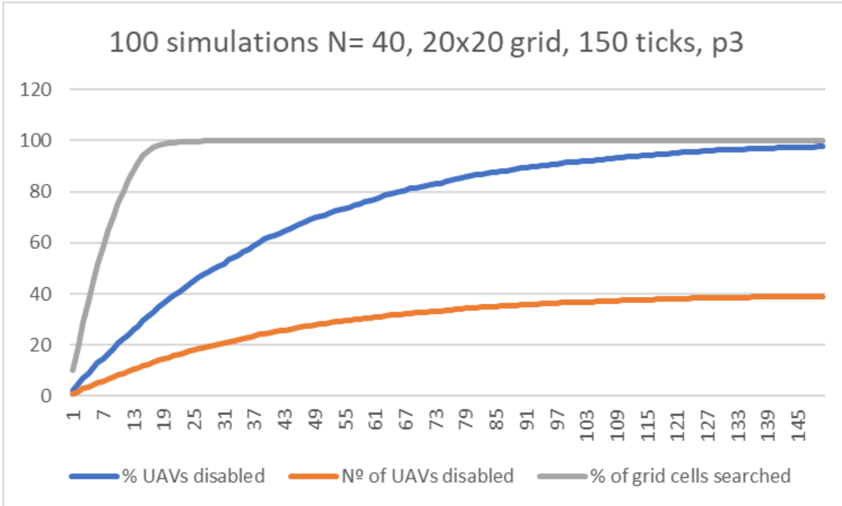

Figure 16. Trial 9: 100 simulations, N=40, 20x20 grid, 150 ticks, p3.

Table 11. Trial 9 data.

n_agents	**40**
probability	**0.0249**
grid_width	20
grid_height	20
n_ticks	150
n_models	100
part1_tolerance_level	0.0016
part2_tolerance_level	0.0018
part3_tolerance_level	0.003
part4_tolerance_level	0.0013
part5_tolerance_level	0.0023
part6_tolerance_level	0.0017
part7_tolerance_level	0.0013
part8_tolerance_level	0.0025
part9_tolerance_level	0.0022
part10_tolerance_level	0.0012
part11_tolerance_level	0.0012
part12_tolerance_level	0.0012
part13_tolerance_level	0.0012
part14_tolerance_level	0.0011
part15_tolerance_level	0.0013
model_statistic_agg_method	average
time_in_seconds_per_model	0.579457
time_in_seconds_for_all_runs	57.94572

Tick N º	**%UAVs disabled**	**N º UAVs disabled**	**% cells visited**	**N º of cells visited**
68	81.55	32.62	99.9975	399.99
69	81.85	32.74	100	400
70	82.2	32.88	100	400
71	82.6	33.04	100	400

Trial 9:

40 UAVs, grid 20x20, 150 ticks

Probability of failure=0,0249. The average number of visits per cell is 3.908125, measured in tick 149.

Observe that the mission is 100% completed with 7.26 UAVs still active, or it required to have at least 33 active UAVs to complete the mission.

3.2.4. Estimation of *k*-out-of-*n* Figure

As depicted in the runs above, the mission success can be measured upon completing 100% of cells searched (visited). The last run, which provides the most advantageous scenarios (more UAVs with the lowest probability of unit failure), is the only result that shows 100% of cells searched. This success is evident starting from tick 69, in which 32.74 UAVs have been disabled. According to the definition of a *k*-out-of-*n* system, we may expect to complete a mission with 33 UAVs, or in terms of *k*-out-of-*n*, we have 33 agents out of 40.

3.2.5. Veracity of Trial Results

The reliability trials presented in this study are based on the premise that UAV failures happen following a constant probability independent of time. In reality, the occurrence of failures in physical components is associated with time.

We are interested in predicting the probability of an entity surviving a certain time interval (tick time, mission time, life of the component, etc.) without failure. Time in this context is usually expressed as the mean time between failures (*MTBF*) or mean time to fail (*MTTF*) (Blanchard & Fabrycky, 2011). Ideally, component manufacturers would provide the *MTBF* or *MTTF* to the users. Nevertheless, this is not always the case, especially for low-cost components such as those used in this study's UAVs. The low cost of these products does not incentivize manufacturers to seek certifications from testing labs. This was apparent in our case, as we reached out to manufacturers and retailers for this information. The scant and limited information provided by those consulted did not prove useful for our study.

Knowing any of these figures would allow us to calculate other reliability figures, such as the failure rate. We also can estimate the probability of failure for the time of the tick simulation and, therefore, use it to test for faults in the tick per component. Thus, we propose for future work to continue the search for factual *MTBF*s and *MTTF*s, which will allow reaching more accurate results.

3.3. Resilience Model

Having a model for the reliability of the UAV team, we use Tran's (2015) Capability-based Resilience Assessment Framework to estimate the resilience of the UAV team. Tran (2015) developed a "Capability-based Resilience Assessment Framework" framework that presents a structured process for assessing resilience. He applies his model in a similar scenario to our UAVs team. In his work, the UAVs cover missions in adversarial, hostile environments, and the units can be disabled by enemy attack. Although, in our case, the environment is non-adversarial, random failures have the same effect on trans scenarios, which is the disability of units. We find that using Tran's framework is suitable for evaluating the resilience of UAV teams comprising surveyance and surveillance of terrains.

In the scenario described above, the resilience framework uses the individual units' reliability to model the failures. We modeled the scenario using agent-based modeling and simulation for reliability and resilience. Each UAV generates random messages to other randomly selected active units to model resilience. On every tick, each unit is subject to a failure injection test, in which each one of the UAVs' components is evaluated by comparing its probability of failure with a uniformly distributed random number between 0 and 1, as described in 2.7.1. Ross (2006).

3.3.1. UAV Team Resilience Model

For our team of UAVs, we are interested in knowing whether the team can respond to unexpected equipment malfunction. A resilient UAV team can improve mission success by adapting to changing environments and maintaining operational effectiveness in the face of unforeseen events.

The success of any surveillance and surveying mission of our UAVs team relies on the capability of communication between the units. Each UAV in our scenario is a Flying Ad-hoc Network (FANET) communication node. Therefore, losing a UAV disrupts the communication structure. This work is interested in assessing the resilience of the UAV team and proposing mitigation measures to restore the team's communication and complete the mission. Our resilience model includes functionality to determine the system's resilience subject to failures simulated with the reliability model described above. The proposed model rests on the Capability-based Resilience Assessment Framework developed by Tran (2015). In our model, the capability studied is a performance measure, $y(t)$, the total number of messages received in the network. Hence, a disruption in the network is envisioned as a diminishing amount of the total received in the network.

We model and simulate the UAV team resilience model using MESA Agent-Based Simulation tool. We programmed the resilience model on top of the reliability model so that the failures induced in individual UAVs will turn off a communication node, disrupting the network. In this model, each UAV exchanges information with the remaining nodes in messages while executing searching and surveillance tasks. Each UAV performs communication tasks in a FANET type of network. All nodes begin the simulation as active nodes. Every node has a probability μ of sending a message at every tick of the simulation. Messages generated are identified by $M_{i,j}$ where i identifies the source node and j the destination. The parameter μ is called the message generation rate. The message generation rate is adjusted in the simulation so that the flow of messages created in each tick is constant as nodes are disabled. The message generation rate at tick t is calculated with the expression,

$$\mu_t = \frac{\mu_0 N_0}{N_t}$$

where μ_0 is the initial message flow rate, and N_0 is the initial number of active nodes at tick t. Destination nodes are selected uniformly at random from the set of active nodes such that the probability of selecting a node j as destination $P(j)$ is given by:

$$P(j) = \frac{1}{N_t - 1}$$

All messages sent from the source node to the destination node are sent following the shortest distance. In each simulation tick, messages are forwarded from the current node to the next neighboring node following the shortest path from the current node position to the destination. It is assumed that each node knows the current status of the network topology and the different nodes so each node can determine the shortest path to all nodes at a given time. The message is lost if no path is available between the origin and destination nodes.

The performance of a network, $y(t)$, is given by the total number of messages received by the network at time t. The metric $y(t)$ is given by:

$$y(t) = \sum_{i=1}^{N_t} B(t)$$

where $B^i(t)$ corresponds to the number of messages received at time t by node i.

Reconfiguration needs in the network topology. The position of failed nodes (UAVs) concerning the rest of their neighborhood is required to establish restitution of the network. Several options were considered before settling on a model based on NetworkX, a Python-based graph tool, which was configured, tested, and adopted to generate new topologies of the UAV network.

Possible restitution schemes. A new topology can be established using NetworkX. Initially, a restitution scheme was applied immediately after the failure; however, the effects of immediate restitution opaqued the expected loss of performance experienced by the failure of a communication node. Consequently, it was not possible to expose the disruption.

As an alternative, the simulator was modified to include extra tick delays before acknowledging the failed node's disability. This alternative is associated with realistic scenarios since the restitution of the network in real situations would take time to detect the failure, set up the remediation, and rewire the connections.

UAV teams' communication performance. In our model, communication between UAVs depends on nodes' availability and mutual reachability while working. The proposed model includes a parameter for the maximum distance between nodes that allow communication. The network's topology relies on the Dijkstra theorem by taking distances among UAVs and minimizing the paths. Multiple simulations were performed to verify the correct behavior of the model.

Following Tran's work, we model the performance $y(t)$ using the number of messages received in the network. The network's capability in terms of messages received will be affected by the disability of nodes. We randomly generate messages sent from each node to the rest of the available nodes in each tick. At the end of the simulation, we obtain a series of numbers with the total of messages received per tick. The resulting measured data is noisy, given the stochastic nature of the simulation process. To improve, we use a smoothing method, Savitzkey-Golay (S-G) filter, which uses least-squares polynomial with tunning parameters polynomial degree (n) and window half width (M). The resilience metric can be calculated directly from the graph.

Total performance factors $\boldsymbol{\delta, \zeta, \rho, \tau}$. As mentioned above, all performance parameters can be determined directly from model. The basic calculations represents the measurement for the total number of messages received per tick in raw and filtered data for a simulation run with 10 UAVs and a probability of sending a message for each selected node of 0.5. The resilience is calculated as follows:

Resilience R:
$0 \leq R \leq \infty$

$$R = \begin{cases} \sigma\rho\left[\delta + \zeta + 1 - \tau^{(\rho-\delta)}\right] \, if \; \rho - \delta \geq 0 \\ \sigma\rho(\delta + \zeta) \;\; otherwise, \end{cases}$$

The value of R has a reference value of two for "normal operating scenarios." A normal operating scenario is a system with no loss over time; *i.e.*, the performance $y(t)$ remains constant for all τ.

Recovery time factor τ:
time to reach steady state
after disruption.

$$\tau = \text{recovery time factor} = \frac{t_{SS} - t_0}{t_{final} - t_0}$$

The total performance factor σ relates to the entire performance maintained by the system during the period of interest. It has a value of one in normal operating scenarios. The absorption factor δ also has a value of one in normal operating conditions.

Total Performance factor σ:

$$\sigma = \frac{\sum_{t=t_0}^{t_{final}} y(t)}{y_D(t_{final} - t_0)} \ (discrete)$$

$$\sigma = \frac{\int_{t=t_0}^{t_{final}} y(t)}{y_D(t_{final} - t_0)} (continuous)$$

Absorption factor δ:

$$\delta = \frac{y_{min}}{y_D}$$

Recovery factor ρ:

$$\rho = \frac{y_R}{y_D}$$

To calculate the volatility factor ζ, we use the series of raw data and its difference with the filtered data as noise to calculate the signal-to-noise ratio (*SNR*). Plugging all referred numbers into the formulas, we obtained a value of *R*. As a reference, a value of *R* = 2 corresponds to a "normal operation scenario." This scenario corresponds to the case in which the values of y_D and y_R are equal.

Volatility factor ζ:

$$\zeta = \frac{1}{1 + exp[-0.25(SNR_{dB} - 15]}$$

As we may observe, it is possible to use visual examination to gather the data. Using visual inspection may introduce errors, specifically when determining y_R. In this case, the recovery capability should be estimated on steady-state conditions. Tran recommends using the Marginal Standard Error Rule method (MSER) to assess the steady-state measure of the recovery time. MSER is a heuristic method for estimating the warm-up period and steady state condition (Wang & Glynn, 2016).

The capability-based resilience assessment framework proposed by Tran (2015) also considers resilience assessment in scenarios with multiple disruptions. In this case, the various scenarios are split into N epochs (*Nepochs*), each one containing a single disruption, and for each epoch, the resilience R_i is calculated using the same equations for R. The total resilience R_{total} is then calculated as:

$$R_{total} = \sum_{i=1}^{Nepochs} w_i R_i$$

where $0 \leq R \leq \infty$ and W_i are defined as the normalized coefficient of an exponential

$$w_i = \frac{\alpha(1-\alpha)^{N_{epochs}-i}}{\sum_{j=1}^{Nepochs} \alpha w_j} = \frac{(1-\alpha)^{N_{epochs}-i}}{\sum_{j=1}^{Nepochs} w_j}$$

weighted moving average filter.

The smoothing factor is α where $0 \leq \alpha \leq 1$ and $w_i = (1-\alpha)^{N_T - i}$. Figure 17 below illustrates the effect of multiple disruptions.

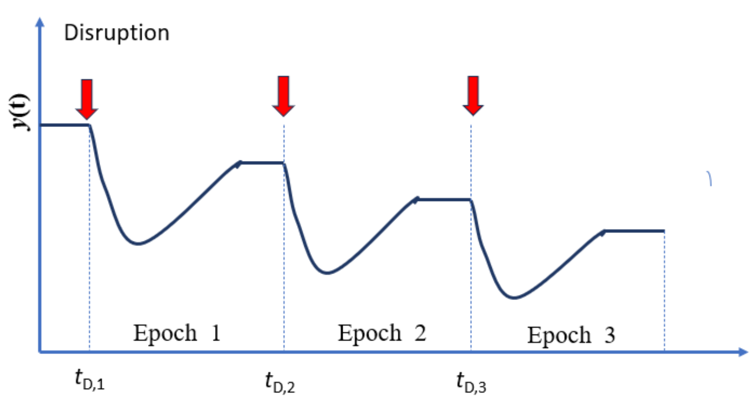

Figure 17. Multiple disruptions. Adapted from Tran (2015).

3.3.2. Resilience Simulations

The study presents a series of fourteen UAV resilience simulation trials, each with a different scenario, to yield R and R_{total} figures for each case. Scenarios vary in terms of total simulation time, number of agents (UAVs), and maximum distance between nodes to sustain a communications link. All scenarios results follow below.

Scenario 1. Figure 18 below shows the result of a 100-tick simulation (one model) with 20 agents, a 20 × 20 grid, and a maximum coverage distance between nodes of 15 units (each grid is 1 × 1 unit). The probability of unit failure is 0.00158.

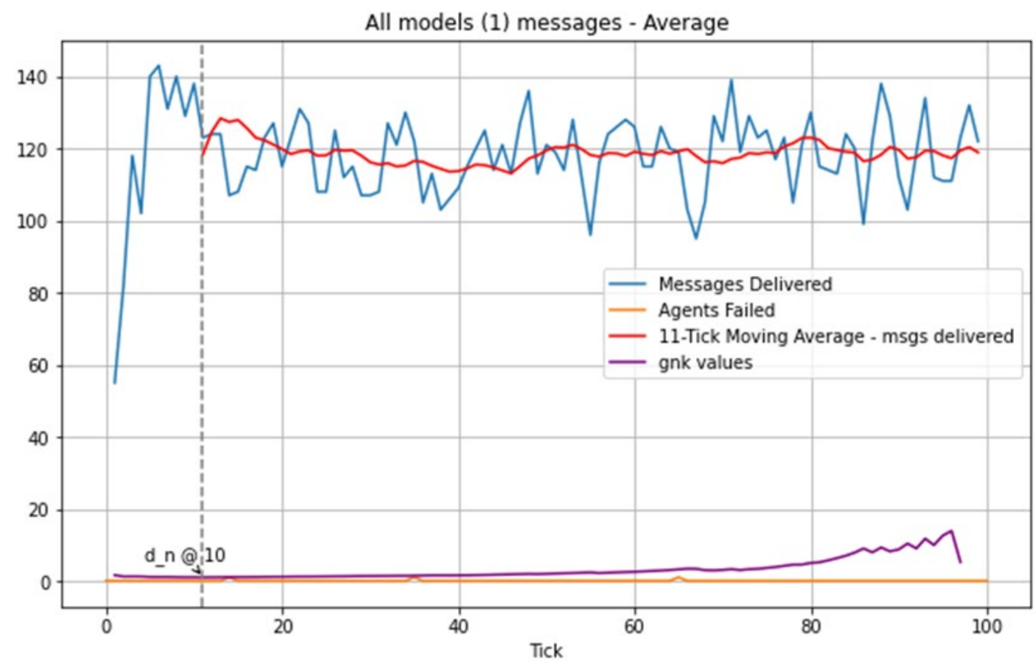

Figure 18. Simulation results graph for Scenario 1.

The results above show the number of messages delivered raw in blue and filtered (with Savitzky-Golay (S-G) filter) in red and MSER(d), showing the value of d that minimizes the MSER function in violet and the failure occurrence in orange. In this case, we can observe that the value of d that minimize the MSER is 10. Therefore, we can truncate all simulation values with $d \leq 10$. Note that three failures occurred in ticks 14, 35, and 65. We calculate the resilience of the UAV team in this example (Scenario 1) using Tran's framework, as shown in Table 12.

Table 12. Resilience outcomes for Scenario 1.

R_1	2.197318	R_2	1.902936	R_3	2.476528
N +10	28	N	45	N	75
d*	17	d*	35	d*	68
y_R	119.1619	y_R	114.0429	y_R	123.8448
y_D	114.366	y_D	118.9138	y_D	110.7692
τ	0.607143	τ	0.777778	τ	0.906667
ζ	0.917832	ζ	0.955356	ζ	0.873798
Pn	707.6842	Pn	271.8443	Pn	833.1279
Ps	206603.1	Ps	144431.4	Ps	156565.9
SNR	24.65297	SNR	27.25341	SNR	22.73985
δ	0.963048	δ	0.904673	δ	0.986469
ρ	1.041935	ρ	0.959038	ρ	1.118044
σ	1.098675	σ	1.059038	σ	1.182576
R=	**2.197318**	**R=**	**1.902936**	**R=**	**2.476528**
α	0.06				
NT	3				
w1	0.8836	w2	0.94	w3	1
w1+w2+w3	2.8236				
Rtotal	**2.1982**				

Scenario 2. Shown in Table 13 is another example (Scenario 2). Figure 19 below, for Scenario 2, shows two failures: in ticks 59 and 73. As in the previous scenario, the figure below shows the result of a 100-tick simulation (one model) with 20 agents, a 20 × 20 grid, and a maximum coverage distance between nodes of 15 units (each grid is 1 × 1 units). The resulting probability of unit failure is 0.00158.

Table 13. Resilience outcomes for Scenario 2.

R_1	2.162095		**R_2**	2.141294
Y_R	112.8177		Y_R	117.1249
σ	1.097872		σ	1.09094
δ	0.904643		δ	0.963515
ρ	0.997872		ρ	0.99094
τ	0.1		τ	0.1
ζ	0.875716		ζ	0.956032
S(n)	140627.8		S(n)	151221.6
N(n)	736.344		N(n)	280.4653
SNR	22.8099		SNR	27.31735
w1=	0.94		**Rtotal =**	2.151373
w2=	1			

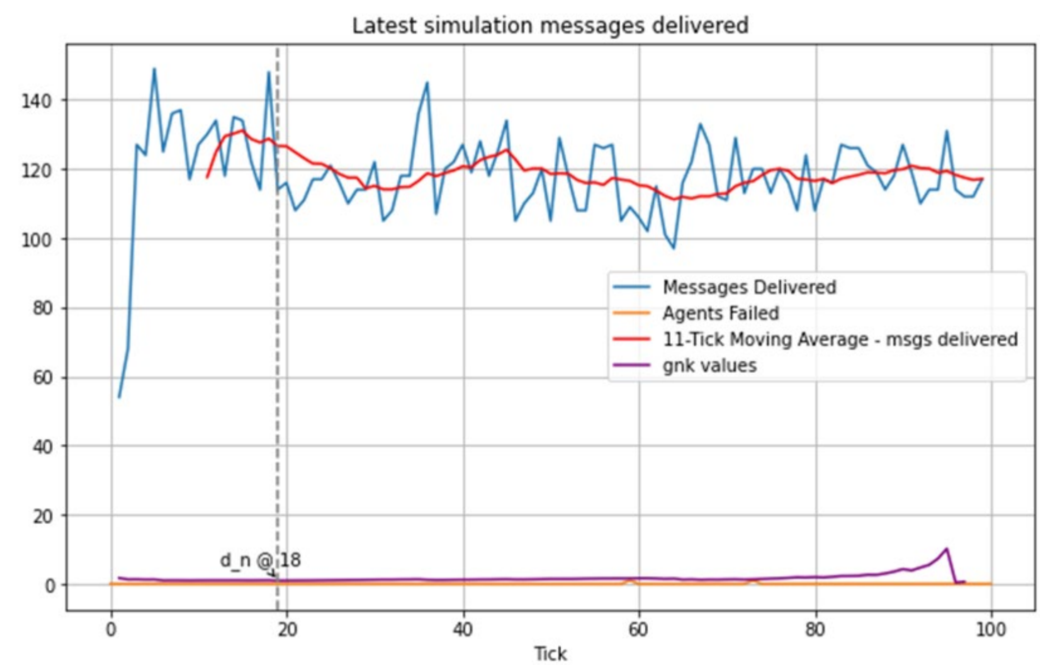

Figure 19. Simulation results graph for Scenario 2.

In both scenarios, we use the Marginal Standard Error Rule method (MSER) where $g_n(k)$ corresponds to the MSER and d corresponds to the minimum valuation of MSER that allows us to obtain k. A module to evaluate MSER has been programmed as part of the simulator. The minimum $d(n)$, shown as a vertical dashed line, is used as the truncation point in the graphs. In both scenarios, the disruptions analyzed occurred after the truncation index, which makes the results more reliable.

$$g_n(k) \triangleq \frac{1}{(n-k)^2} \sum_{j=k}^{n-1} \left(Y_j - \hat{Y}_{n,k}\right)^2 \qquad \hat{d}(n) = \underset{0 \le k \le n-2}{\arg\min}\, g_n(k)$$

Figure 20 shows the result of one hundred simulations using a grid size 20x20, 100 ticks, and a max distance of 15. Unit failure probability 0.00157. The average number of messages sent is close to 120 (for raw and filtered data). The average truncation index $d(n)$ is 15. The average failure rate is very low and distributed uniformly.

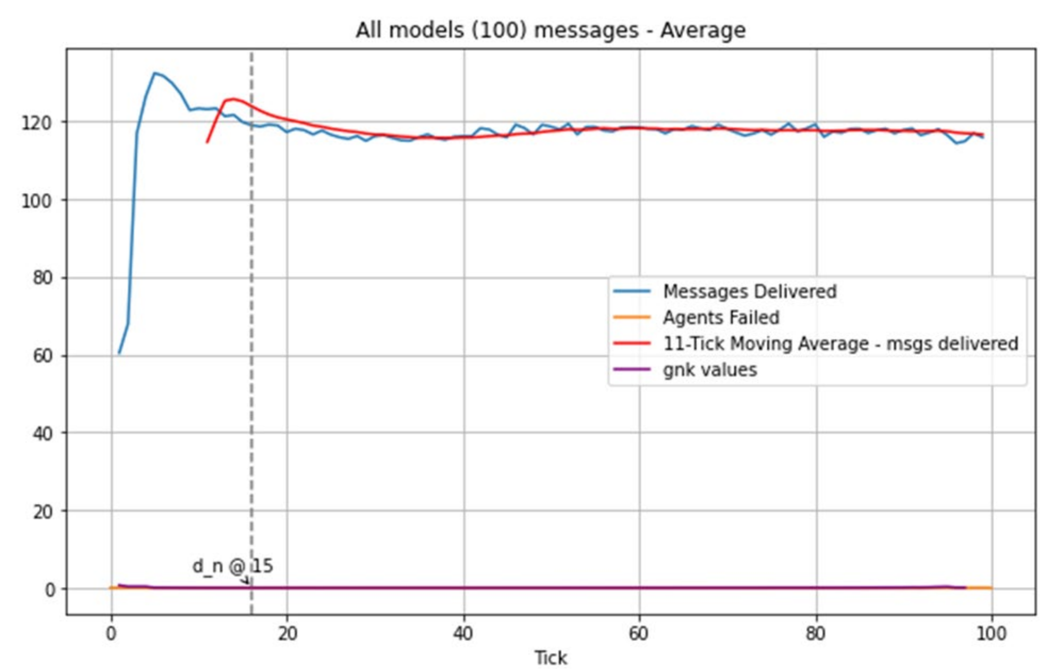

Figure 20. Simulation results graph: Average for 100 messages.

Scenario 3. Simulation results for Scenario 3 are seen in Table 14 and Figure 21. The parameters considered in this scenario are shown in Table 15, and the resulting probabilities of component and unit failure can be read from Table 16.

Table 14. Simulation results for Scenario 3.

failure#	drone_id		tick	failure_type	failure_position	R	Rtotal
1	19		53	part7_failure	(9, 5)	1.635198	
2	7		71	part3_failure	(8, 14)	1.505867	
3	4		88	part8_failure	(17, 19)	1.730154	
							1.625772

Table 15. Results for Scenario 3: Parameter values.

Parameter	Value
number_of_agents	20
grid_width	20
grid_height	20
number_of_ticks	100
number_of_models	10
probability_drone_sends_message	1
adjust_probability_mu_sub_t	TRUE
node_edge_limit	15
max_messages_per_drone_per_tick	8
offline_nodes_message_ack_delay	5
msg_graph_moving_average_ticks	11
alpha_constant	0.06
savgol_filter_order	3
savgol_filter_framelen	11

Note that at least 17 ticks separate the failures; therefore, the formula for determining R_{total} can be applied.

Table 16. Results for Scenario 3: Probability of component and unit failure.

Probability of component failure	
part1	0.0001
part2	0.0001
part3	0.00011
part4	0.00012
part5	0.00016
part6	0.00017
part7	0.00013
part8	0.00012
part9	0.000097
part10	0.00014
part11	0.00015
part12	0.00001
part13	0.000012
part14	0.00018
part15	0.00013
Probability of unit failure	0.001729

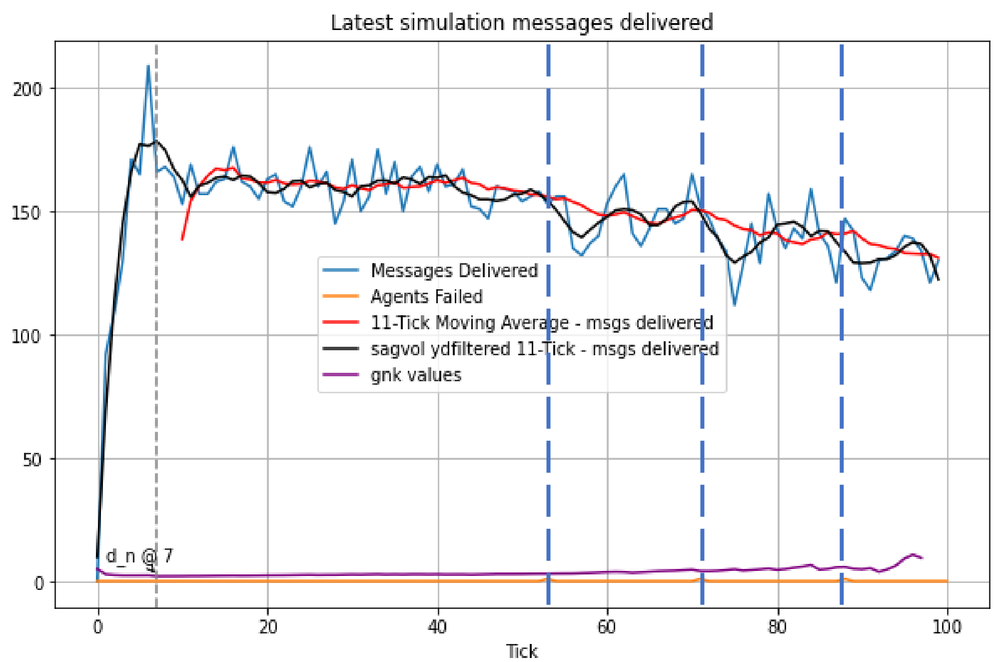

Figure 21. Simulation results graph for Scenario 3.

Scenario 4. Another scenario uses the same parameters as Scenario 3, with the following results, expressed in Table 17 and Figure 22.

Table 17. Simulation results for Scenario 4.

drone_id	tick	failure_type	failure_po	R	Rtotal	10 AVE Rt
8	14	part5_failure	(4, 5)	2.881766		2.289687805
17	29	part5_failure	(11, 5)	1.690433		1.735979861
10	52	part10_failure	(12, 7)	1.596578		1.583529963
12	71	part10_failure	(6, 16)	1.680679		1.676253373
7	75	part1_failure	(11, 13)	2.942936		1.961427417
14	98	part4_failure	(15, 2)	NAN		1.959010808
					2.161812	1.643431476
						1.769308647
						2.161812158

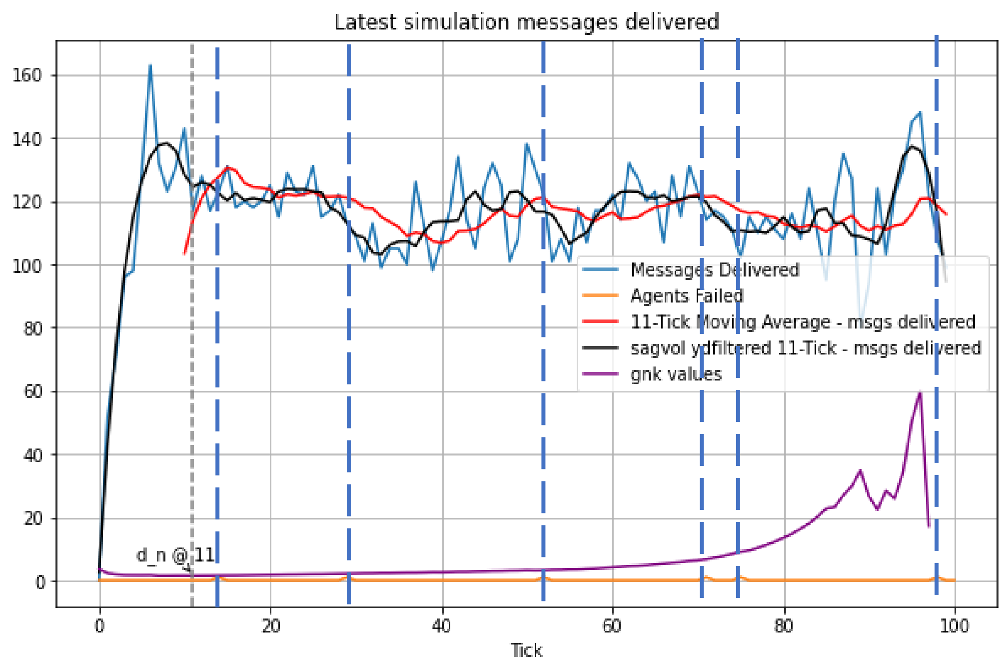

Figure 22. Simulation results graph for Scenario 4.

Observe that in this scenario, failures 4 and 5 (ticks 72 and 75, respectively) occur within a period of restitution expected of 5 ticks (given by the following metric:

ofline_node_message_ak_delay), resulting in an overlapping two-time series of interest. Therefore, results for R for failures 4 and 5 and the R_{total} value should be disregarded.

Scenario 5. In this scenario, we extend the simulation time up to 1500 ticks. We use a 40 UAVs team, grid of 20 × 20, probability of unit failure of 0.001729, node edge limit (maximum distance between nodes) of 15, a maximum number of messages per UAV per tick of 20, initial probability of sending messages of 0.01 and induced restitution time of 5 ticks per failure. See the results in Figure 23 and Table 18.

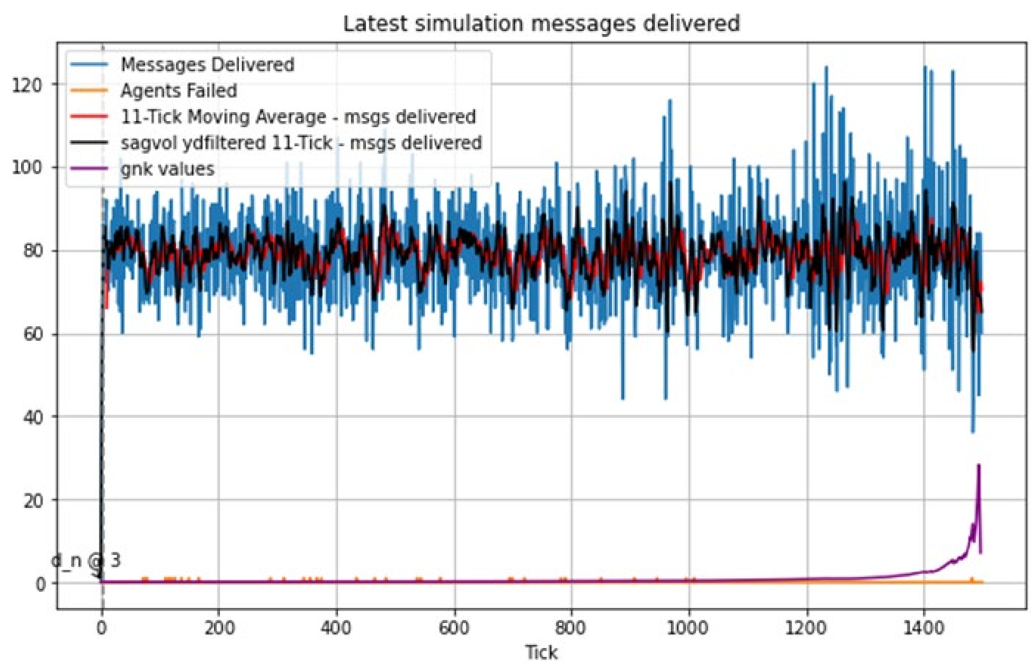

Figure 23. Simulation results graph for Scenario 5.

Table 18. Simulation results for Scenario 5.

failure	tick	r1
1	72	1.617451
2	75	1.785918
3	78	3.041077
4	110	2.862912
5	116	1.745233
6	119	1.995944
7	125	1.507678
8	126	1.352896
9	137	2.65369
10	150	1.867548
11	166	1.43531
12	289	1.758833
13	311	2.226914
14	345	2.662247
15	346	2.731954
16	356	1.466811
17	367	1.468729
18	375	2.02915
19	435	1.522384
20	466	2.161215
21	485	1.528175
22	538	1.356525
23	544	2.017813
24	578	2.615215
25	695	1.588499
26	700	2.038401
27	721	2.806847
28	783	1.69288
29	790	1.472754
30	851	2.698574
31	908	1.299552
32	946	1.415168
33	996	1.619458
34	1010	1.46209
35	1483	1.236709
prob. of unit fail.		0.001729

Scenario 6. In this scenario, we simulate 20 UAVs in a 20 × 20 grid with an adaptation (node rewiring) time of five ticks apart after a disruption. The probability of sending per tick equals 1 for 1000 ticks; that of unit failure is 0.001729. Observe that the minimum time between two consecutive fails is seven; therefore, the R and R_{total} can be applied without a problem. Also, observe that the initial probability of sending messages starts at 1. This corresponds to the highest possible transmission rate; therefore, there is no possibility of adjustment in the rate after each failure. Figure 24 and Table 19 reflect the results for this scenario.

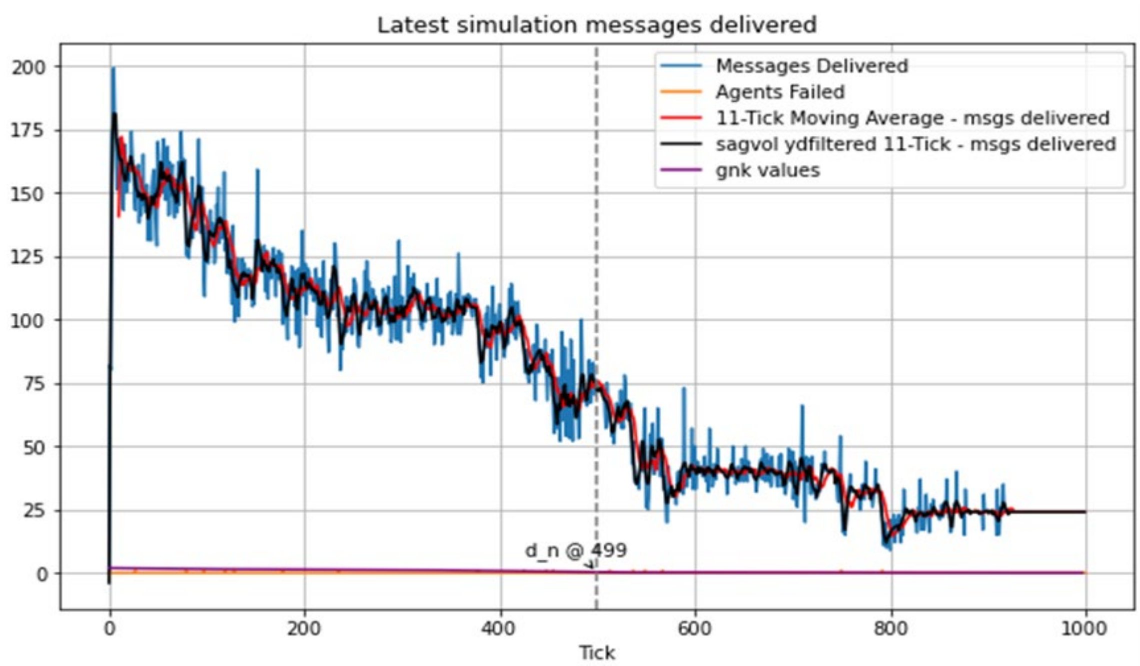

Figure 24. Simulation results graph for Scenario 6.

Table 19. Simulation results for Scenario 6.

tick	fail	R	R total
27	1	1.786574	
79	2	1.79485	
97	3	1.684937	
119	4	2.543884	
128	5	1.67925	
178	6	1.490723	
235	7	1.360695	
378	8	1.430567	
425	9	1.55454	
448	10	2.83663	
455	11	1.391226	
513	12	1.486304	
537	13	1.063004	
549	14	0.766672	
567	15	0.735971	
750	16	0.519702	
792	17	0.391541	
			1.305743

Scenario 7. In this scenario, we use an initial probability of sending of 0.25. Notice that the generation of messages remains steady (in average) from the first tick up to beyond around tick 550, in which the factored probability reaches the maximum of 1. This fact shows the approach to keep the generation of messages constant up to this point. Observe that there is a total of 17 failures.

There is a minimum separation between disruptions of 8 ticks except for tick 201, in which two losses overlap. Though the failures occur in the same interval, the overlapping cannot be treated as a unique failure, as two different nodes are involved, and two differing restitution schemes are applied. The calculated R_{total} is not a reliable figure. The results corresponding to this scenario are shown in Figure 25 and Table 20.

Table 20. Simulation results for Scenario 7.

Failure	Tick	R	R_{total}
1	13	1.307574	
2	57	1.500467	
3	65	2.604974	
4	165	1.773058	
5	175	2.230793	
6	201	1.015697	
7	201	1.015697	
8	227	1.059469	
9	258	1.815059	
10	267	1.018415	
11	308	1.275126	
12	393	1.327816	
13	416	1.079146	
14	480	1.015985	
15	526	1.076629	
16	570	0.95429	
17	721	0.500414	
			1.231299

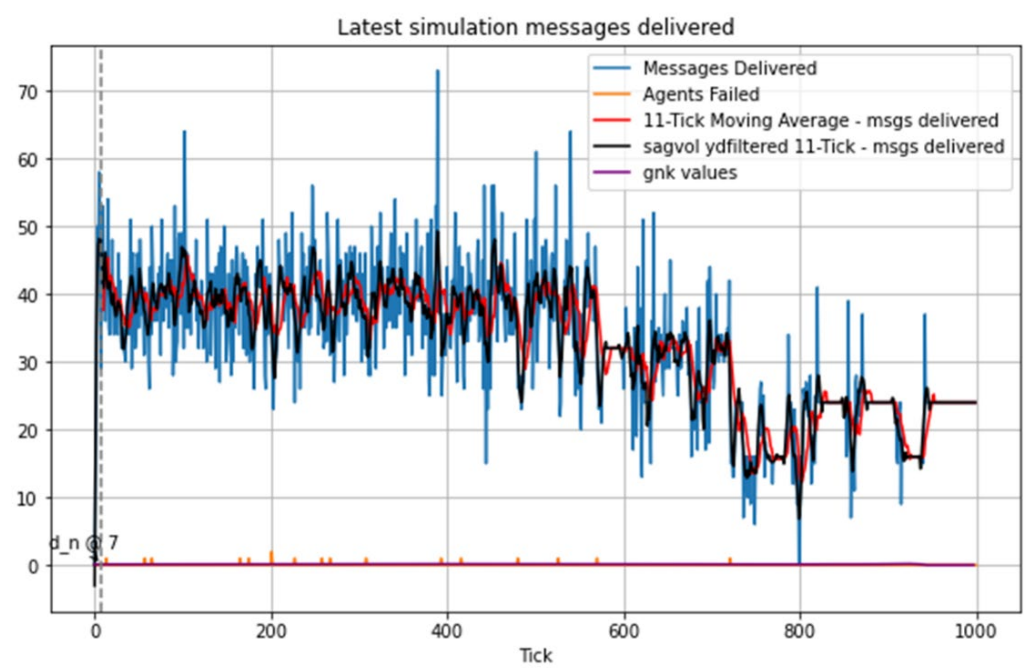

Figure 25. Simulation results graph for Scenario 7.

Scenario 8. This scenario corresponds to multiple simulations (10) with the same parameters as in Scenario 7.

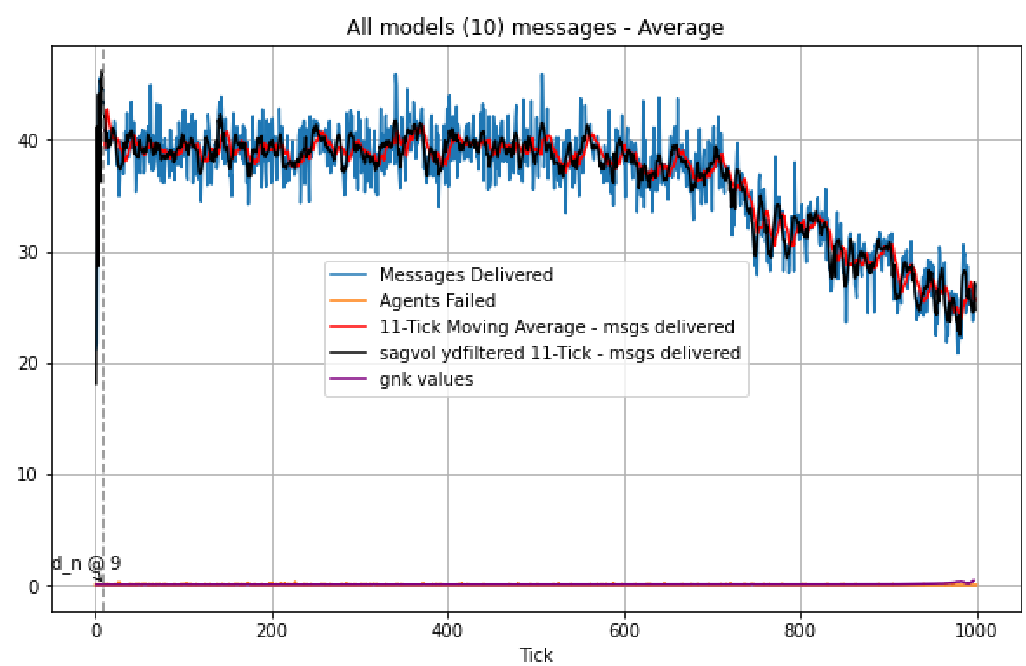

Figure 26. Simulation results graph for Scenario 8.

In Figure 26, we can observe with more clarity that the performance average is up to tick 600.

Scenario 9. This scenario presents a simulation with 40 UAVs in a 20 × 20 grid, 1500 ticks, node edge limit 15, and 100 delay restoration. The corresponding simulation results are shown in Table 21 and Figure 27.

Table 21. Simulation results for Scenario 9.

Failure #	Tick #	R
1	12	0.8255814
2	14	0.86737
3	38	0.9579294
4	49	0.8127329
5	64	1.6171863
6	66	1.2301486
7	76	2.2498632
8	92	2.2253236
9	100	2.5622395
10	142	2.4293234
11	142	2.4293234
12	158	3.3238113
13	162	2.7757092
14	169	2.5043986
15	203	2.1709132
16	229	2.578763
17	272	2.0548188
18	399	0.7230512
19	405	0.8414951
20	421	0.8511402
21	454	1.0572923
22	460	1.0891675
23	472	1.1122684
24	481	2.7516548
25	509	1.3116086
26	543	1.7240824
27	585	2.2385635
28	591	1.8064623
29	599	1.6255761
30	689	2.5616572
31	712	0.8958591
32	807	1.0796873
33	829	0.7405059
34	860	1.2922013
35	991	0.4462816
36	1048	0.9932836
37	1231	0.550706
38	1383	0.0847977
R	**Ave.**	1.3156275

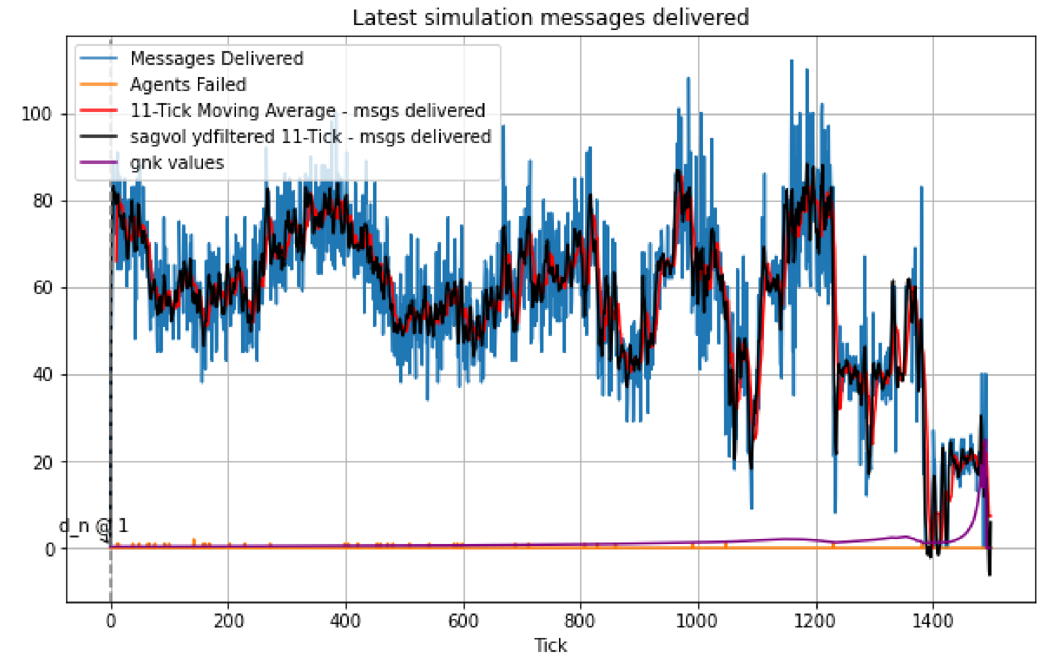

Figure 27. Simulation results graph for Scenario 9.

Scenario 10. The following graph (Figure 28) shows a scenario without adaptation with a 1000-tick simulation time.

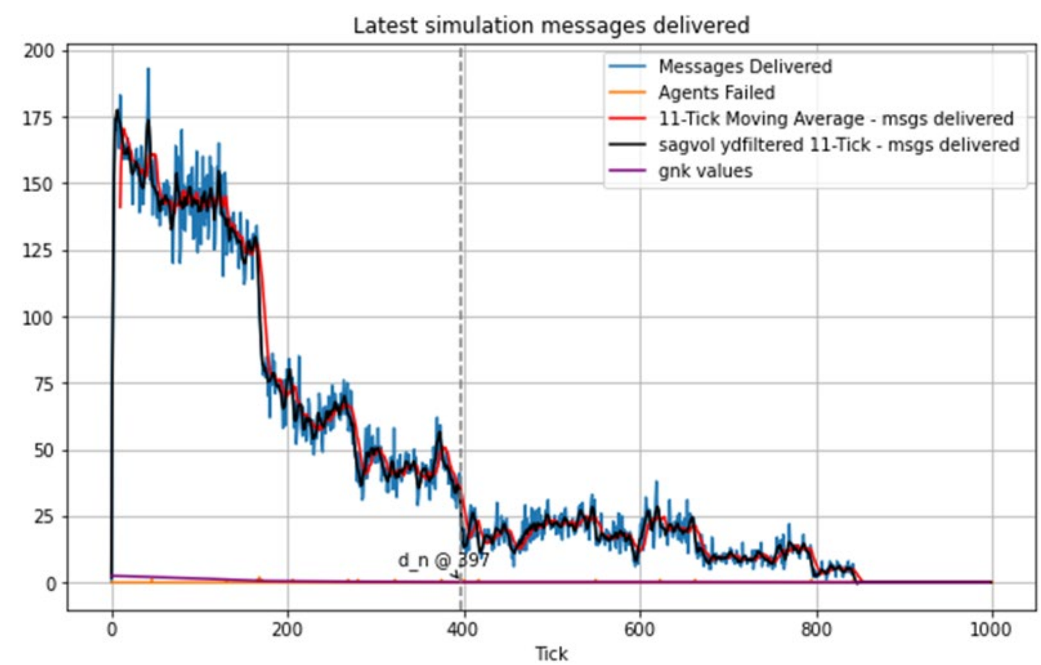

Figure 28. Simulation results graph for Scenario 10.

The simulation is performed on a 20 × 20 grid using 20 UAVs, with the initial probability of sending messages equal to one. The inserted restitution period is set to 999 ticksso that the absence of adaptation is accounted in the simulation. The results of this simulation are seen in Table 22.

Table 22. Simulation results for Scenario 10.

tick	failure	yd
18	1	151.06993
125	2	129.01865
127	3	120.97669
134	4	97.265734
175	5	85.920746
205	6	84.83683
233	7	61.475524
269	8	56.179487
360	9	43.710956
360	10	43.710956
378	11	32.517483
462	12	24.121212
566	13	14.967366
988	14	9.1748252

Figure 29 below is a chart showing the values of $y(t)$ through each one of the failure points, up to Scenario 10. The blue line represents performance of $y(t)$ without noise produced by raw data in the simulation.

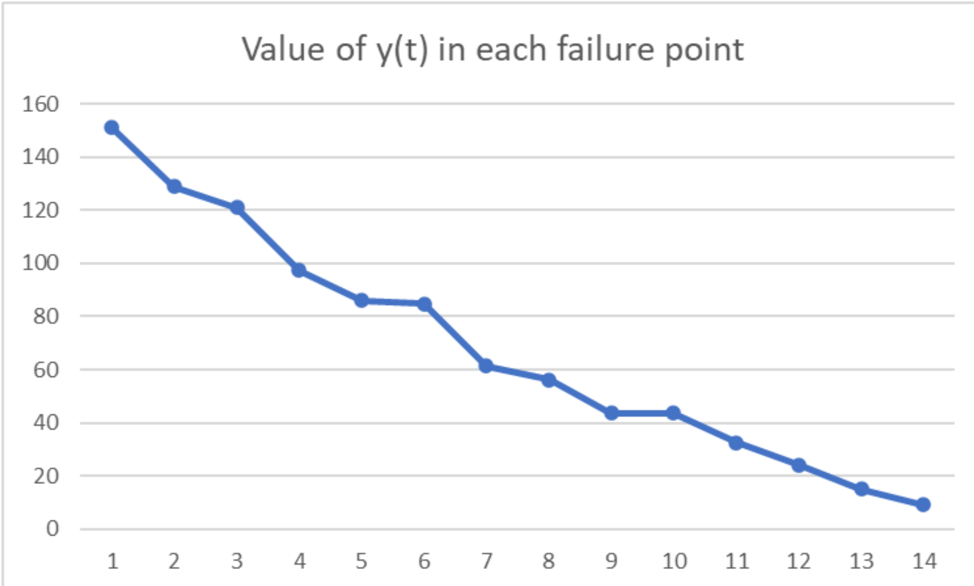

Figure 29. Value of $y(t)$ at each failure point up to Scenario 10.

Scenario 11. As shown in Figure 30, in this scenario, we present 40 agents on a 20 × 20 grid with no adaptation and an initial probability of sending messages of 0.1. The probability of unit failure of 0.001729. The total simulation time is 1000 ticks.

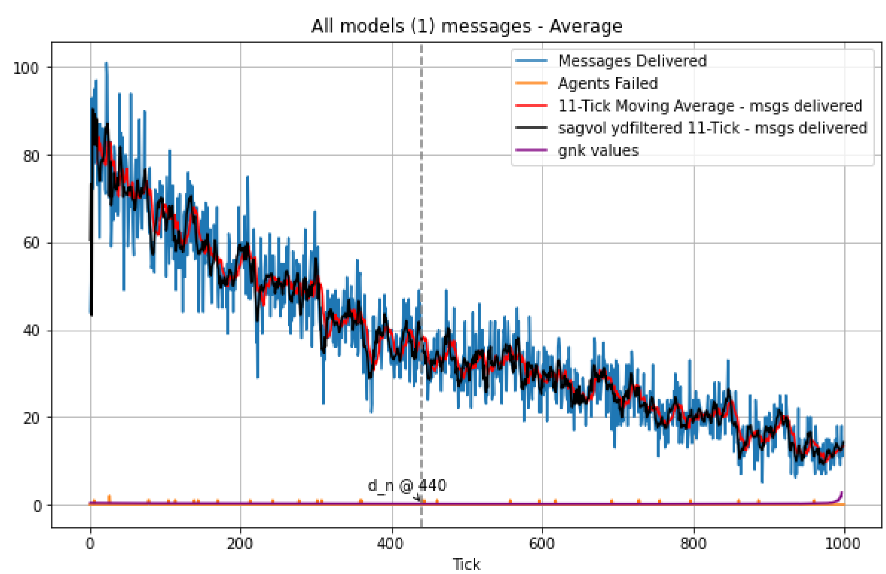

Figure 30. Simulation results graph for Scenario 11.

Scenario 12. In this scenario, we use the same parameters as in scenario 11, extended to 2000 ticks. The total dismissal of the team is reached with 40 failed units in tick 1361, which can be seen in Figure 31; the $y(t)$ behavior is also illustrated in Figure 31, again with the blue line representing $y(t)$ without raw data noise.

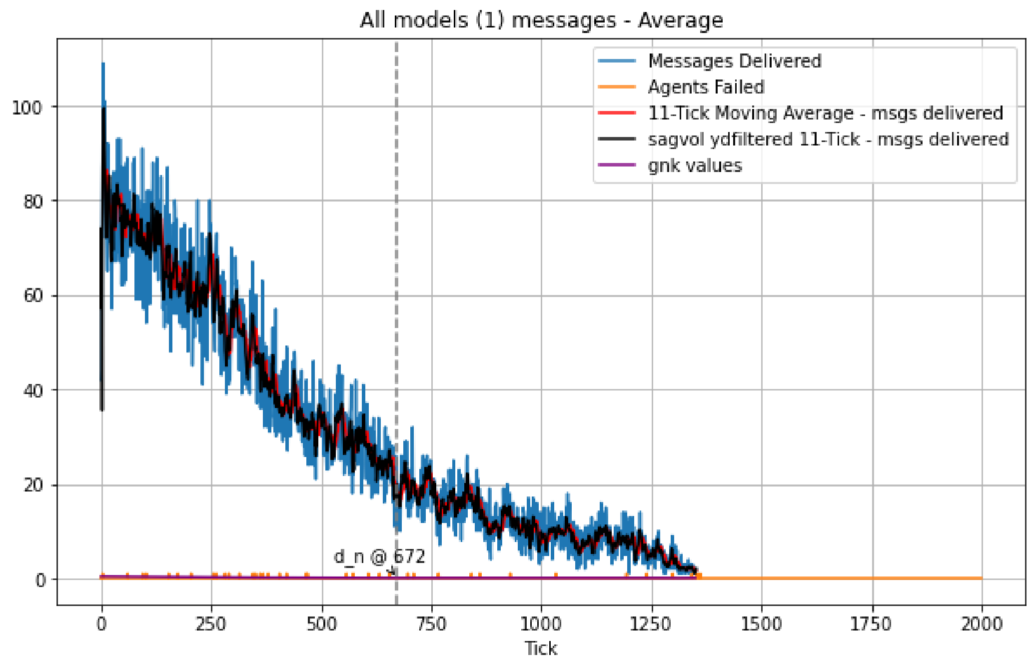

Figure 31. Simulation results graph for Scenario 12.

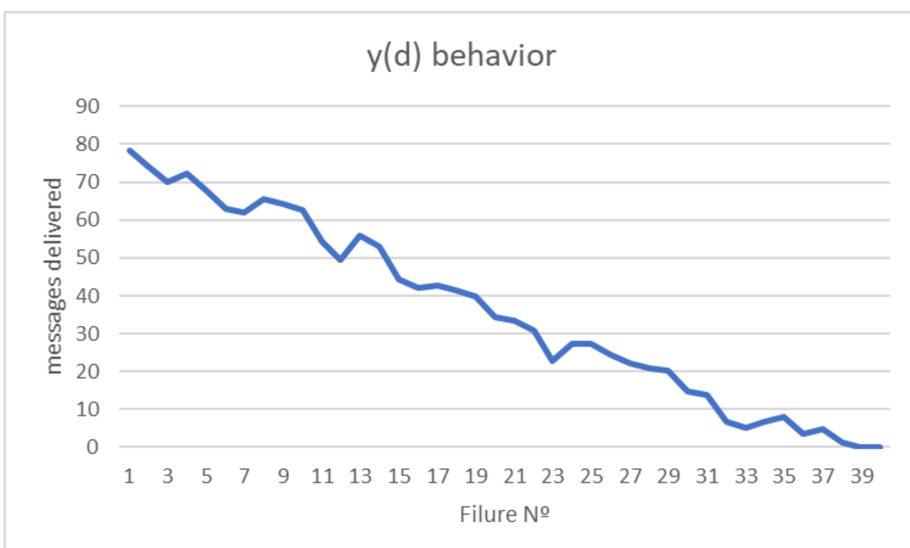

Figure 32. The $y(d)$ behavior for Scenario 12.

Scenario 13. In this scenario, we present 40 agents on a 20x20 grid, with no adaptation, with an initial probability of sending messages of 0.1 and a probability of unit failure of 0.001729. The total simulation time is 1000 ticks. The table corresponds to the results of the last simulation of the group. Observe that the occurrence of failures is within five ticks and hence is valid to calculate R_{total}. These results are illustrated in Figure 33 and Table 23.

Table 23. Simulation results for Scenario 13.

failure	tick	R
1	22	2.007378
2	83	0.707544
3	88	1.840578
4	142	1.100243
5	164	0.513178
6	216	1.815182
7	292	1.220888
8	344	1.39791
9	503	0.704828
10	535	1.081279
11	741	0.565056
12	757	0.171957
13	808	0.926044
14	822	0.270616
15	849	2.899545
16	906	0.437155
Rtotal		1.064587

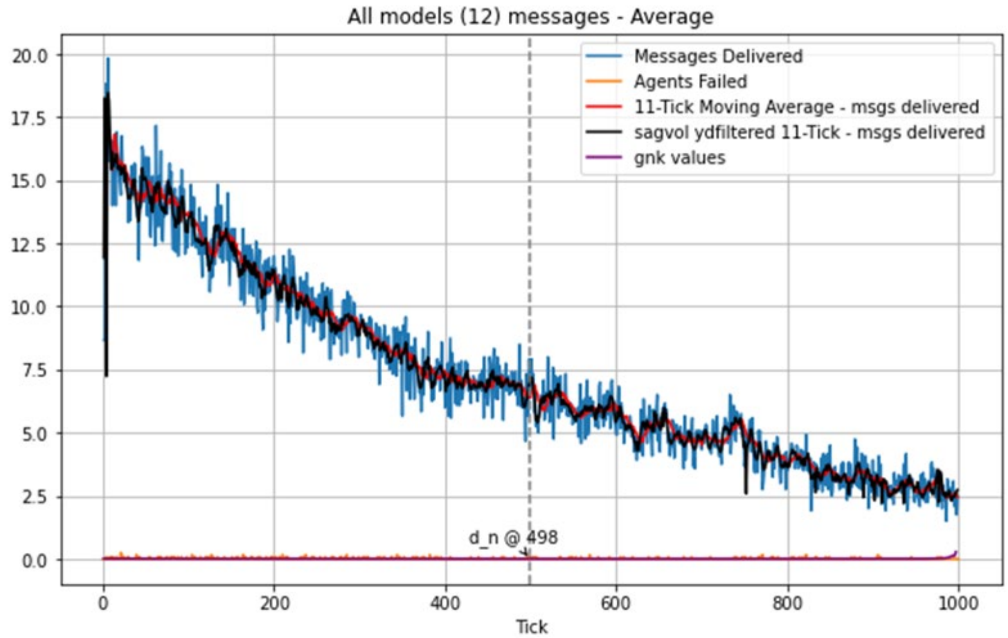

Figure 33. Simulation results graph for Scenario 13.

Scenario 14. This scenario shows a 1000-tick run with the same characteristics as Scenario 13. No overlapping (assuming 5 ticks delay) is observed, therefore results for *R* and R_{total} are valid. See Figure 34 and Table 24 for results.

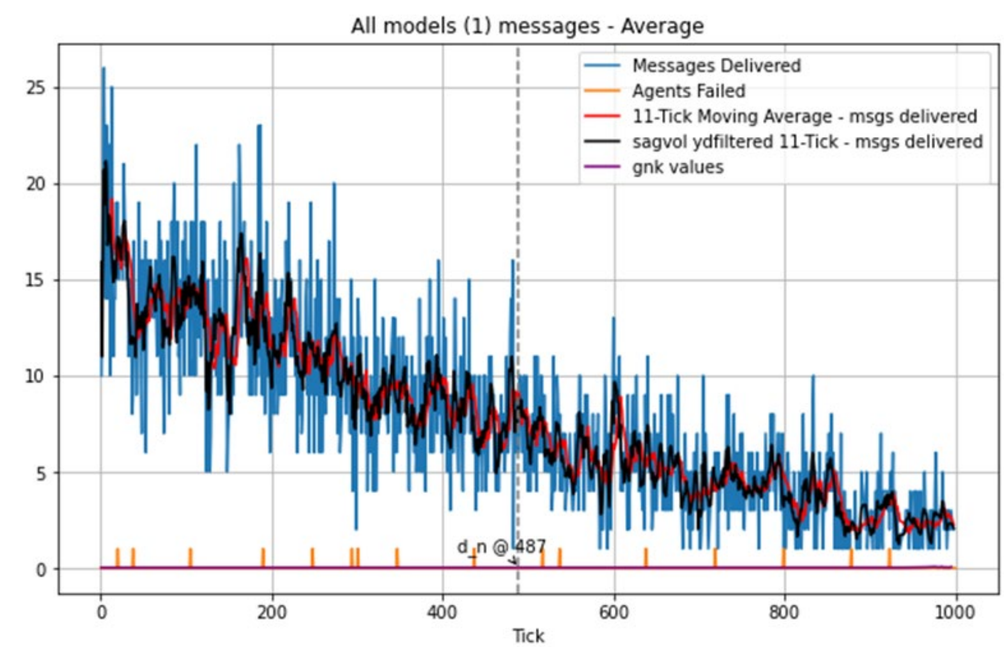

Figure 34. Simulation results graph for Scenario 14.

Table 24. Simulation results for Scenario 14.

failure	tick	R
1	20	1.807693
2	38	1.414756
3	105	1.231202
4	190	0.920793
5	248	1.067532
6	294	0.987102
7	301	1.169683
8	347	1.908125
9	437	0.923228
10	517	0.732973
11	537	0.542694
12	638	0.584244
13	719	1.905055
14	799	1.424675
15	878	0.700175
16	923	0.25076
Rtotal		1.041927

3.3.3. Impact of Failures on Team Communication

The Resilience Model includes a networking module based on NetworkX, which is the foundation for modeling and simulating communications within the module. This module keeps track of the optimum connectivity of all network nodes using the Dijkstra theorem. It determines the best path available from each node to the rest of the active nodes for a given maximum distance or scope of the links. The simulator keeps track of all node positions within the grid. The location of the operational units is updated every tick according to the Target Area Surveillance Algorithm. Similarly, the topology and connections of the network are updated in every tick of the simulation.

However, the fact that the topology updates every tick of the simulation creates an undesirable state in the model. The reason is that the rewiring is performed in every tick, so the disruption does not show up. As a consequence, we cannot use the framework under these conditions. Every time a disruption occurs, a latency variable is introduced to complete the rewiring in a selected number of ticks, thus overcoming this inconsistency. In real cases, node loss requires some time to reestablish the connections.

Given that the rewiring will require reestablishing connections at various levels of the network, each one with a particular protocol demanding time, the model follows a realistic approach. The simulator can select the latency for several ticks to delay the rewiring. During this latency, the nodes connected with the failed node will keep sending messages to the failed node. The other nodes will update the topology that excludes the failed node.

Tran (2015) runs into a similar situation, he uses a delay t_{adapt}, essentially a rewiring time, such that the restitution time is the sum of the disruption time plus rewiring time. The figures that follow, Figures 35 through 40, show the distribution of UAVs under varying conditions.

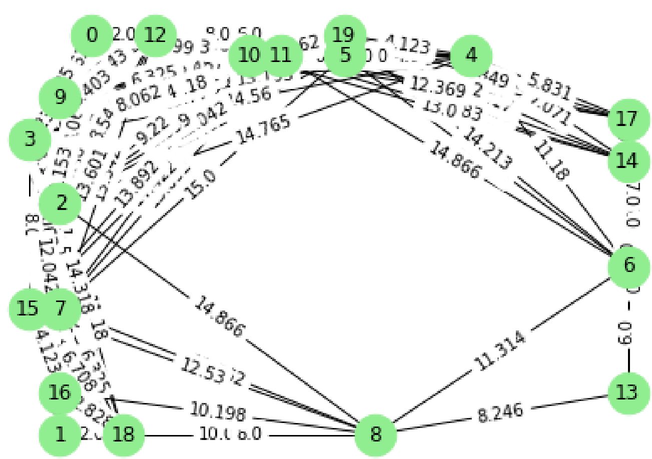

Figure 35. Initial distribution of the UAVs (tick 0) with a maximum distance between nodes in 15 (Euclidian distance).

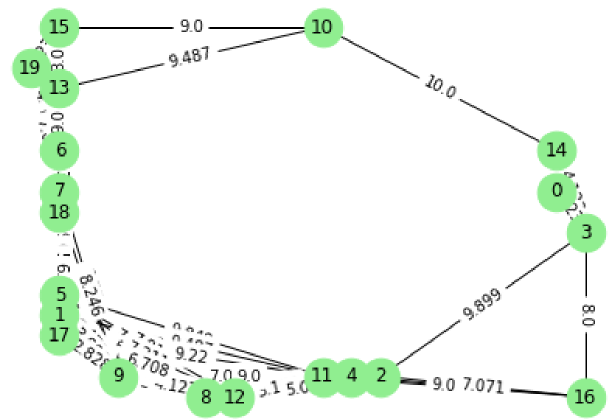

Figure 36. Topology with a maximum distance of 10.

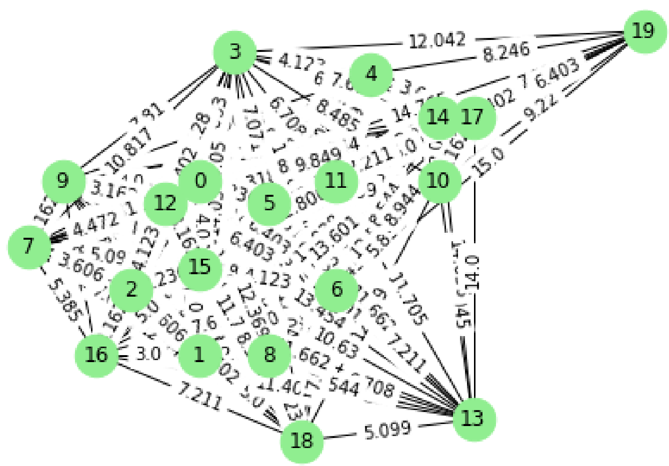

Figure 37. Network status on tick 14.

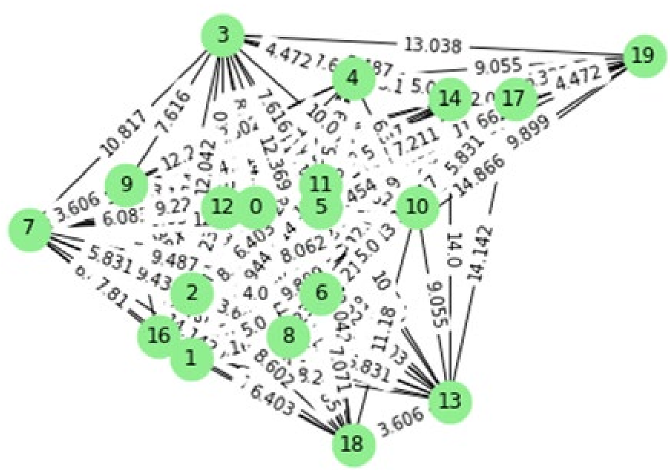

Figure 38. Tick 15 after the failure of unit 15.

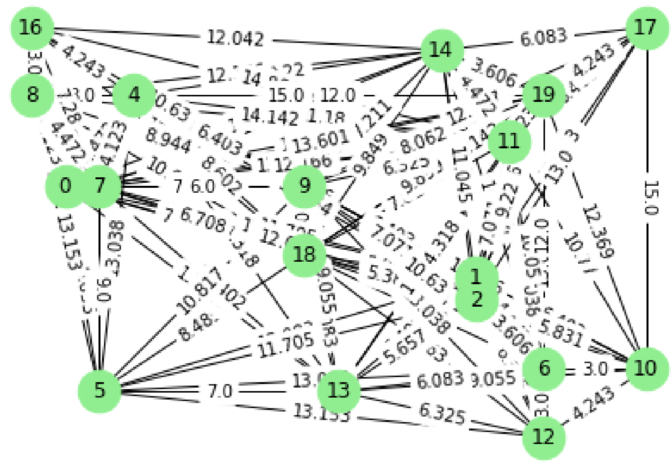

Figure 39. Network status on tick 64.

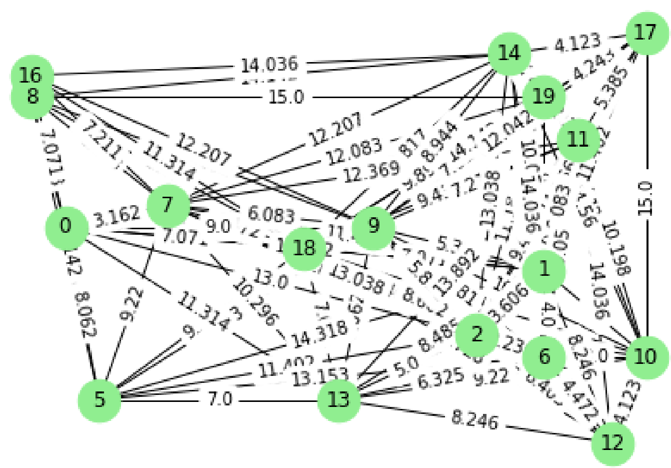

Figure 40. Tick 65 after the failure of unit 4.

These results show that the topology changes at every tick of the simulation. At any point in time, each UAV knows the complete topology of the network in terms of the location and distances of each other node. This allows for generating the minimum distance concerning the destination of messages. Figures also show the rewiring results after disruptions.

SECTION 4

PERFORMANCE EVALUATION OF THE PROPOSED UAV TEAM RESILIENCE MODEL

4.1. Validation Strategy

Validation is the process of determining whether a simulation model accurately represents the original. There is agreement in the discipline that agent-based simulations are difficult to validate; however, it is also acknowledged that validation is a basic prerequisite for ABS models and their reasonable use, particularly for multi-agent systems (Klügl, 2008). When considering such paradigms, she suggests several validation options, ideally performed in combination.

A primary validation strategy is the expert consensus strategy, which aims to demonstrate the validity of the proposed model through qualitative evaluation by subject matter experts (SMEs). This is a type of Delphi method defined by Skulmoski *et al.* (2007, p.2) as "an iterative process used to collect and distill the judgments of experts using a series of questionnaires interspersed with feedback… The process stops when the research question is answered… [for example] when sufficient information has been exchanged." Other strategies include additional types of face validation as well as sensitivity analyses, statistical model validation, and transitivity demonstration.

The research plan for this study listed face validation by SMEs as the validation strategy for the proposed UAV team resilience model, a sensitivity analysis, and a comparison of results with other published work (reviewed in Section 2). Twenty-one national and international researchers working on Agent-based Modeling were contacted and invited to participate in a survey as the first step in carrying out the validation. Researchers were asked to state their agreement or disagreement with 12 questions and encouraged to add comments to their responses. The list of survey questions is provided in *Appendix A*; they cover vital issues on the performance of different parts of the proposed model. Out of the 21 researchers contacted, only one agreed to participate, a response rate that does not meet the method's criteria. Further, despite the agreement, the researcher did not provide a response.

Limited time and resources prevented the complete validation of our model by engaging Subject Matter Experts (SMEs) in face validation, as initially intended and declared in Research Goal 3. To mitigate this issue, this study attempted transitive validation for the complete UAV team resilience model (Section 4.3) described by Klügl (2008).

4.2. Reliability Trials Analysis

A preliminary test of the model was inspired by Klügl's (2008) notion of sensitivity analysis. The strategy entails the realization of an experimental plan in which parameter values are changed according to a systematic plan to identify the variables to which the model is sensitive. For every one of these parameter value combinations, one or more simulation runs are executed and evaluated. Klügl recommends more simulations if the model contains stochastic elements. Our study followed a modified method to test the reliability model.

We used nine trials or simulations changing the number of UAVs, *i.e.*, 25, 35, and 40 units, and the probability of failure (0.055, 0.044, and 0.0249) for a total of 9 simulations, each running for 150 ticks. Mission success is regarded as reaching 100% of all cells searched; thus, the user of this model may play with different numbers of UAVs and reliability of units to design the mission for success and its costs.

The trials presented in Section 3 intend to reveal a relationship between two model variables, the number of UAVs and probability of failures, and the total number of cells searched at the end of the simulation. A clear trend was exposed when incrementing the number of UAVs (25, 35, and 40), maintaining the same probability, the number of visited cells increased. Similarly, keeping the number of UAVs fixed and selecting different failure probabilities, the number of visited cells changed accordingly. Although the results align with the expected behavior (*e.g.*, using more UAVs leads to a greater area coverage), they did not meet the thoroughness required for a Sensitivity Analysis.

These assessments do not validate the model presented in this study. Instead, they indicate the need to complete its validation following stricter protocols.

4.3. Transitive Validation

Klügl (2008) mentions transitive validation as another method for approving a simulation model, as it has been accepted as a verification technique in social science simulation. She argues that it can be considered as a tool for validation on the idea that validity is a transitive relation: "If a model *A* is validly reproducing a reference system *O* and a model *B* replicates the results of *A* with sufficient detail, then *B* is also valid for reproducing *O*." This is the case with the UAV Team model offered in this study and, thus, the argument for its validity.

Tran (2015) applies his capability-based resilience assessment framework to quantify the ability of IE (Information Exchange) networks to maintain and recover lost capabilities. He defines different scenarios based on three extents: Initial network topology, adaptation method (rewiring the remaining nodes after the disruption), and type of threats. Initial network topologies include scale-free and random network topologies (ER). Adaptation methods include recalculated degree adaptation, preferential adaptation, and aleatory wiring. He considers two types of threats: targeted node removal (recalculated node degree-RD) and random removal. The scenario is adversarial; the nodes are disabled by threats, attacks, and unplanned node failures (random removal). The threats occur on a regular time basis.

In this study, we are interested in comparing the results of our research with the most suitable scenarios. Since the initial topology in our research is generated in an aleatory manner, we should compare it with the results of random networks (ER model). In regard to the threats, random failures would make the scenario similar to random node removal (R). Since upon a failure, rewiring is required with the remaining nodes, in our model, nodes are rewired using the most convenient connection according to the topology. The best choice is to compare it with random rewiring.

Figure 41 shows Tran's results of the mean of R_{total} for two scenarios of a simulation with no adaptation, both using random network initial configuration (ER): One of them using random detachment removal (RD) and the other random removal (R). The median is 0.25 for ER-RD and 1 for ER-R. In our simulations of scenarios 14 and 15, R_{total} values are 1.064587 and 1.041927, respectively.

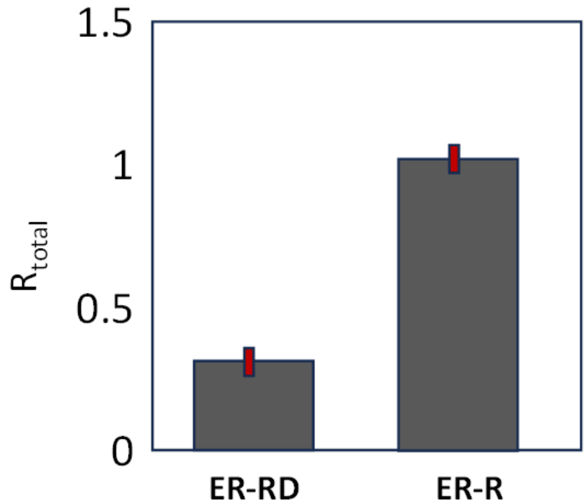

Figure 41. Mean R_{total} for two simulation scenarios with no adaptation. Adapted from Tran (2015).

Random topology (ER) with arbitrary removal and adaptation with time sensitivity (Tran, 2015) compares with our results:

Scenario 1: R_{total} = 2.151373

Scenario 2: R_{total} = 2.181982

Scenario 3: R_{total} = 1.625772

Scenario 6: R_{total} = 1.305734

R_{total} (average) = 1.81672

These results evidence the alignment of this study's results with those of Tran (2015) and suggest the validity of this model for the sample scenarios.

Figure 42 shows the mean R_{total} for four different scenarios using random initial topology with varying adaptation methods. These are specified as none, recalculated degree adaptation (RDA), preferential adaptation (PA), and random rewiring adaptation (Rand.).

The new topology is recalculated or updated following any change to the network topology, in our case, arbitrary removal similar to our scenario in which the failure of UAVs is random. We observed that the absence of adaptation contributes to a decline in resilience R. Regarding the R_{total} on the random removal option, the mean is around one.

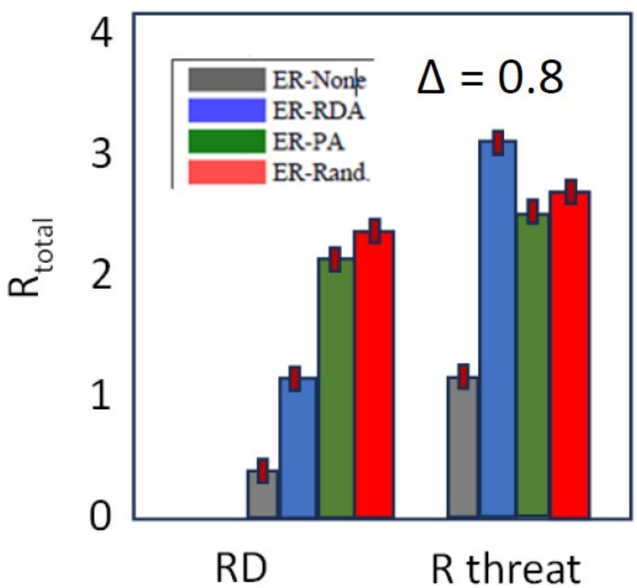

Figure 42. Mean R_{total} for four different scenarios using random initial topology with different threats. Adapted from Tran (2015).

Figure 43 shows the R_{total} for scenarios considering time sensitivity with $\Delta = 1$ (no time sensitivity), random topology, and random threats (failures). The random initial topology (ER) with random removal and adaptation with time sensitivity helps compare with our sample scenarios' results. The median corresponding to random removal and random rewiring is shown in red, as are our values of interest, which are those associated with random threats, similar to our random unit failures (the yellow bar).

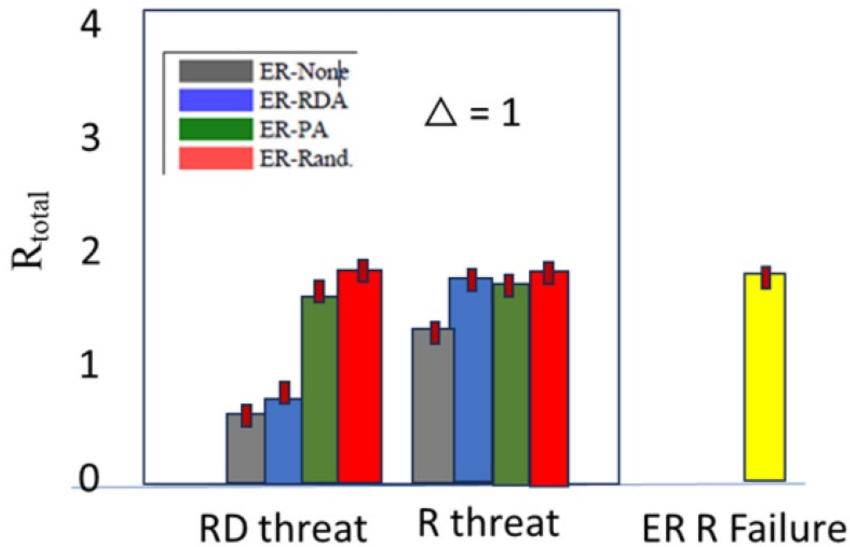

Figure 43. Comparison of Tran's (2015) results (in red) to this study's results (the yellow bar).

These study results point to alignment with those of Tran (2015) and suggest the validity of the UAV reliability component of the model for the sample scenarios.

SECTION 5

DISCUSSION AND CONCLUSIONS

5.1. Summary

In this study, we present a framework for assessing the resilience of a UAV team accounting for unit reliability. This framework incorporates individual UAV reliability in a team model under a FANET (Flying Ad-hoc) structure, with unit failures modeled using a reliability model based on realistic figures. We use this model to evaluate the impact of unit losses on their team performance, assess the team's resilience, and allow identifying mitigation measures to restore the team's communication and complete the mission. Such a model is more realistic and behaves more accurately than known models for UAV swarm resilience evaluation, as the probabilities of failures are obtained from component specifications.

We implemented a model of UAV teams subject to failures to simulate the impact of the individual UAV failures on the team communication performance, identify measures to mitigate the disruption and continue with the mission. This framework addresses the current lack of a model that assesses the resilience of UAV teams subject to individual UAV failures. The implemented model:

a) Constitutes a valuable tool for designers of UAV missions. It provides a means for assessing UAV system resilience based on the reliability of its constituent units. Additionally, it yields the following contributions to the community of UAV technologists:

b) Can simulate a UAV team's performance on a search or surveillance mission and calculate its resilience. When done in the early stages of the operation design, it allows implementing of any necessary changes to the system and its components to increase team resilience and thus approach the mission with greater confidence.

c) Offers a framework for modeling and simulating UAV teams' behavior under multiple conditions, including faults. Such a framework allows UAV team designers to discover the underlying collective behavior of the system and therefore design proactively.

d) Allows for assessing the financial implications of a UAV team configuration. By simulating a proposed team's performance and ability for fault recovery (resilience), decision-makers can explore cost-effective options to complete the intended mission within a budget.

More specifically, the study contributes to the state-of-the-art knowledge of UAV system resilience. The collective behavior of the UAV team is challenging to describe and model using analytical tools. It is generally a complex system with multiple parts interacting and influencing each other. In some cases, it is tough, if not impossible, to produce a model that analytically describes all interactions of their components. On the other hand, agent-based modeling and simulations such as those used in this study take a bottom-up approach. They allow describing the behavior of a complex system, such as a team of autonomous UAVs, by modeling each agent engaged in the collective behavior.

Additionally, the proposed agent-based modeling framework for UAV teams provides a means to estimate the effect of eventual changes on team performance. Such a framework would apply when the behavior of individual units is subject to changes due to hardware and software modifications, potential improvements, or failure of any component. It allows evaluation of the effect of changes and alterations to the network topology and hence, its requirements for changes in routing algorithm parameters used in the mobile network. Therefore, it allows for assessing the consequences of losing units on the team's performance.

The study set out to develop a model for mitigating system failure by considering the reliability of individual units and their impact on mission success. For this, it identified three main research goals:

Research Goal 1. To identify appropriate tools for modeling UAV scenarios. Reliability and resilience theory principles underlie this model, having applied existing tools for reliability calculation, fault trees, UAV technology, and agent-based modeling and simulation. Analysis of several options led to choosing MESA ABS for model implementation.

Research Goal 2. To develop a model for assessing UAVs team resilience that overcomes the limitations of previous studies. Previous studies approach the resilience of the model using random detachment of nodes. Our model presents a more realistic model as data is available for modeling the removal of the nodes from the team by the incidence of failures in individual UAVs. Our MESA-based simulation model allows modeling the reliability and resilience of UAV teams in surveyance missions.

Research Goal 3. Validate the model through expert consensus. The model was tested by sensitivity analysis and transitive validation, which suggested its validity. We recommend a thorough face validation following an iterative Delphi process as the next step for this inquiry.

5.2. Further Research

The study presents a framework to model the reliability and resilience of a team of unmanned autonomous vehicles, UAVs. The value of such a model to the eventual designer of UAV-based surveillance missions is evident as a planning tool. The simulation tool is conceived flexibly to adapt to improvements and modifications. We followed the same approach as the Capability-based Resilience Assessment Framework proposed by Tran (2015). We modeled and simulated multiple scenarios using such a framework, obtaining comparable results to Tran's work. He models the scenarios using random node removal and random rewiring. In contrast, in our model, the removal of the nodes is given by the probability of failure.

However, Tran's model has limitations, specifically in calculating R_{total}. Their model assumes that multiple disruptions occur in discriminated time intervals so that each disruption can be analyzed and calculated as R. We argue that disruptions may overlap each other in their time interval, in which case it is not possible to identify the correct values of parameters to calculate R and R_{total}.

For a more realistic model, we suggest using actual values of failure probabilities (*MTBF*) for the particular components. The model and the simulator can be improved to include more functionality, such as application to heterogeneous UAVs. We recommend continuing the quest to make this model a helpful planning tool for the users of UAV teams searching and surveying terrains.

Regarding the model validation, we acknowledge that the assessments reported here do not fully validate the model presented in this study. Instead, they indicate the need to complete its validation following strict protocols. Klügl (2008) proposes a validation framework consisting of four steps: Face Validation, Sensitivity Analysis, Calibration, and Statistical Validation. This process, applied holistically, is suggested as the next step in future research.

Additional strategies for model validation may also include empirical research, such as using real-life UAVs and running experiments in real-life environments. Such testing would provide clarity as to the validity of the model as well as its actual performance strengths and limitations. This approach would require the financial and logistical support that this study lacked.

REFERENCES

Abar, S., Theodoropoulos, G. K., Lemarinier, P., & O'Hare, G. M. P. (2017) Agent Based Modelling and Simulation tools: A review of the state-of-art software. *Computer Science Review*, 24, 13–33. DOI: 10.1016/j.cosrev.2017.03.001.

Abdullah, Q A (n/d). 1.4. Classification of the Unmanned Aerial Systems. *GEOG 892 Geospatial Applications of Unmanned Aerial Systems*. E-book. Penn State University e-Education Institute. https://www.e-education.psu.edu/geog892

Aggarwal, S., Kumar, N., & Tanwar, S. (2021). Blockchain-Envisioned UAV Communication Using 6G Networks: Open Issues, Use Cases, and Future Directions. IEEE Internet of Things Journal, 8(7), 5416–5441. DOI: 10.1109/JIOT.2020.3020819.

Bai, G., Li, Y., Fang, Y., Zhang, Y. A., & Tao, J. (2020). Network approach for resilience evaluation of a UAV swarm by considering communication limits. *Reliability Engineering & System Safety*, *193*, 106602.

Beiderbeck, D, Frevel, N, vonderGracht, HA, Schmidt, sl & Schweitzer, VM (2021). Preparing, conducting, and analyzing Delphi surveys: Cross-disciplinary practices, new directions, and advancements. *MethodsX* 8(2021), 101401. DOI: 10.1016/j.mex.2021.101401

Bekmezci, I., Sahingoz, O. K., & Temel, S. (2013). Flying Ad-Hoc Networks (FANETs): A survey. *Ad Hoc Networks*, 11(3), 1254–1270, doi10.1016/j.adhoc.2012.12.004.

Bestaoui, S. (2020a). *Intelligent Autonomy of UAVs: Advanced Missions and Future Use*. Chapman & Hall/CRC Press. ISBN 9780367571962

Blanchard B. S., Fabrycky W. (2011). *Systems Engineering and Analysis*, 5th Edition. Pearson. ISBN: 978-0132217354

Brust, M. R., Strimbu, B. M. (2015). A networked swarm model for UAV deployment in the assessment of forest environments. *2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, 1–6. IEEE. DOI: 10.1109/ISSNIP.2015.7106967.

Cook, C (2017). An evaluation of the effectiveness of low-cost UAVs and structure from motion for geomorphic change detection. *Geomorphology* 278(1 February 2017), 195–208. https://doi.org/10.1016/j.geomorph.2016.11.009

Crawford, M., Wright, G (2016). Delphi Method. *Wiley StatsRef: Statistics Reference Online*. DOI: 10.1002/9781118445112.stat07879.

Cybulski, P., Zielinski, Z. (2021). UAV Swarms Behavior Modeling Using Tracking Bigraphical Reactive Systems. *Sensors*, 21(2), 622–. DOI: 10.3390/s21020622.

De Queiroz, M., Cai X., & Feemster M. (2019). *Formation Control of Multiagent Systems*. Wiley. ISBN: 978-1-118-88744-8

Fortune Business Insights (2023). *Unmanned Aerial Vehicle Market Size, Share, Analysis and Regional Forecast, 2020-2027*. Report ID: FBI101603. https://www.fortunebusinessinsights.com/industry-reports/unmanned-aerial-vehicle-uav-market-101603

Freeman, P. M. (2014). *Reliability Assessment for Low-cost Unmanned Aerial Vehicles*. (Doctoral dissertation, University of Minnesota Digital Conservancy). https://hdl.handle.net/11299/170136

Grenzdorffer, G. Teichert, B, (2008). The photogrammetric potential of low-cost UAVs in forestry and agriculture. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVII. ISPRS.

Guillen-Perez, A., Cano, M.-D. (2018). Flying Ad Hoc Networks: A New Domain for Network Communications. *Sensors*, 18(10), 3571–. DOI: 10.3390/s18103571.

Hamilton, T. Ochmanek, D (2020). Operating Low-Cost, Reusable Unmanned Aerial Vehicles in Contested Environments: Preliminary Evaluation of Operational Concepts. Santa Monica, CA: RAND Corporation. https://www.rand.org/pubs/research_reports/RR4407.html.

ICAO (2020). *ICAO Advisory Circular (AC) 102-1*. International Civil Aviation Organization. https://www.icao.int/safety/UA/UAID/Documents/AC102-1.pdf

ICAO Convention on International Civil Aviation, (2006). *Document 7300*. International Civil Aviation Organization (ICAO) https://www.icao.int/publications/Documents/7300_9ed.pdf

Information Technology Laboratory (n/d). Computer Security Resource Center (CSRC), National Institute of Standards and Technology (NIST), US Department of Commerce. https://csrc.nist.gov/glossary/term/information_system_resilience

Jakaria, A. H. M., Rahman, M. A. (2018, July). Formal analysis of k-resiliency for collaborative UAVs. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)* (Vol. 1, pp. 583-592). IEEE.

Klugl, F. (2008). A validation methodology for agent-based simulations. *Proceedings of the 2008 ACM Symposium on Applied Computing*, pp. 39–43. ACM. DOI: 10.1145/1363686.1363696

Koeneke R., Babiceanu R. F., & Seker R. (2019). Target area surveillance optimization with swarms of autonomous micro aerial vehicles, *2019 IEEE International Systems Conference (SysCon)*, 1-6. DOI: 10.1109/SYSCON.2019.8836761.

Kopfstedt, T., Mukai, M., Fujita, M., & Ament, C. (2008). Control of Formations of UAVs for Surveillance and Reconnaissance Missions. *IFAC Proceedings Volumes*, 41(2), 5161–5166. DOI: 10.3182/20080706-5-KR-1001.00867.

Koutalakis, P., Tzoraki, O., & Zaimes, G (2019) UAVs for Hydrologic Scopes: Application of a Low-Cost UAV to Estimate Surface Water Velocity by Using Three Different Image-Based Methods. *Drones* 3(1), 14 https://doi.org/10.3390/drones3010014

Law, A. M. (2015). *Simulation modeling and analysis*, Fifth edition, McGraw-Hill. IBSN 978-0-07-340132-4.

Li, B., Bai, G., Guo, Y., Zhang, Y.-A., & Tao, J. (2018). Mission-Oriented Reliability Prediction for Unmanned Aerial Vehicle Swarm using Consecutive k-out-of-n Structure. *IET Conference Proceedings*. 2018 Stevenage: The Institution of Engineering & Technology. DOI: 10.1049/cp.2018.0317.

Luo, W., Sycara, K. (2019). Voronoi-based Coverage Control with Connectivity Maintenance for Robotic Sensor Networks. *2019 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, 148–154. IEEE. DOI: 10.1109/MRS.2019.8901078.

Macal, C. M., North, M. J. (2010). Tutorial on agent-based modelling and simulation. *Journal of Simulation, 4*(3), 151-162. DOI: 10.1057/jos.2010.3

Murtha, J. (2009). *An Evidence Theoretic Approach to Design of Reliable Low-cost UAVS*. Master's thesis. Virginia Polytechnic Institute and State University. https://vtechworks.lib.vt.edu/bitstream/handle/10919/33762/JustinMurthaThesisFinal.pdf

Niederberger M., Spranger J. (2020). Delphi Technique in Health Sciences: A Map. *Frontiers in Public Health* 8:457. DOI: 10.3389/fpubh.2020.00457.

Ordoukhanian, E., Madni, A. (2019) Model-Based Approach to Engineering Resilience in Multi-UAV Systems. *Systems* 7(1), 11–. DOI: 10.3390/systems7010011.

Petritoli E., Leccese F., & Ciani L. Reliability and Maintenance Analysis of Unmanned Aerial Vehicles. *Sensors* 2018, 18(9):3171. https://doi.org/10.3390/s18093171

Phadke, A. Medrano, F. A. (2022). Towards Resilient UAV Swarms—A Breakdown of Resiliency Requirements in UAV Swarms. *Drones* 6(11), 340. https://doi.org/10.3390/drones6110340

Polaris Consulting (2016). Mimicking ants to develop the capability of autonomous vehicles. CDE Marketplace September 2016.The Centre for Defence Enterprise. https://www.gov.uk/government/news/mimicking-ants-to-develop-capability-of-autonomous-vehicles

Qin L., X. He X. & Zhou D.H. (2014). A survey of fault diagnosis for swarm systems. *Systems Science & Control Engineering*, 2:1, 13-23. DOI: 10.1080/21642583.2013.873745

Ross, S. M. (2006). *Simulation*, 4th Edition. Elsevier Science. ISBN: 9780080517223

Russell S. J., Norvig P. (2021). *Artificial Intelligence: A Modern Approach*, 4th Edition. Pearson Series in Artificial Intelligence. ISBN-13: 9780137505135

Shakhatreh, H, *et al*. (2018). Unmanned Aerial Vehicles: A Survey on Civil Applications and Key Research Challenges. *IEEE Access*, vol. 7, pp. 48572-48634, 2019. DOI: 10.1109/ACCESS.2019.2909530.

Skorobogatov, G., Barrado, C., & Salami, E. (2020). Multiple UAV Systems: A Survey. *Unmanned Systems* 8(2), 149–169. DOI: 10.1142/S2301385020500090.

Skulmoski, G. J., Hartman, F.T., & Krahn, J. (2007). The Delphi Method for Graduate Research. Journal of Information Technology Education 6(1) 1-21. DOI: 10.28945/199.

Soon-Jo Chung, Paranjape, A. A., Dames, P., Shaojie Shen, & Kumar, V. (2018). A Survey on Aerial Swarm Robotics. *IEEE Transactions on Robotics*, 34(4), 837–855. DOI: 10.1109/TRO.2018.2857475.

Tran, H. T. (2015). *A complex networks approach to designing resilient system-of-systems* (Doctoral dissertation, Georgia Institute of Technology). http://hdl.handle.net/1853/54384

Tran, H. T., Domerçant, J. C., & Mavris, D. N. (2015). Evaluating the agility of adaptive command and control networks from a cyber complex adaptive systems perspective. *The Journal of Defense Modeling and Simulation*, *12*(4), 405-422.

Tran, H.T., Balchanos, M., Domercant, J.C., & Mavris, D.N. (2017). A framework for the quantitative assessment of performance-based system resilience. *Reliability Engineering & System Safety*, *158*, 73-84.

Tran, H.T., Domercant, J.C., & Mavris, D.N. (2016). A network-based cost comparison of resilient and robust system-of-systems. *Procedia Computer Science*, *95*, 126-133.

Uday, P. Marais, K. (2015). Designing Resilient Systems-of-Systems: A Survey of Metrics, Methods, and Challenges. *Systems Engineering* 18 (5): 491–510. https://doi.org/10.1002/sys.21325.

Venkataraman, R., Seiler, P., Lukátsi, M., & Vanek, B. (2017). Reliability Assessment of Actuator Architectures for Unmanned Aircraft. *Journal of Aircraft*, 54(3), 955–966. DOI: 10.2514/1.C033832.

Wang, R., and Glynn, P. (2016). On the Marginal Standard Error Rule and the Testing of Initial Transient Deletion Methods. *ACM Transactions on Modeling and Computer Simulation*, vol. 27, no. 1, 2016, pp. 1–30, https://doi.org/10.1145/2961052.

Wheeb, A. H., Nordin, R., Asma, A. S., Alsharif, M. H., & Muhammad, A. K. (2022). Topology-based routing protocols and mobility models for flying ad hoc networks: A contemporary review and future research directions. *Drones*, 6(1), 9. DOI: 10.3390/drones6010009.

Xiang, X., Kennedy, R., & Madey, G. (2005). Verification of Agent-based Scientific Simulation Models. *Proceedings of the Agent-Directed Simulation Symposium*. San Diego, California. ISBN: 1-56555-291. https://www3.nd.edu/~nom/Papers/ADS019_Xiang.pdf

APPENDIX A

VALIDATION SURVEY QUESTIONS

1. The target area surveillance (search) algorithm employed in this study provides a suitable scenario to simulate potential threats or disruptions on a team of UAVs executing surveillance missions.

    Please comment on your answer.

2. The tests conducted with multiple pilot simulations indicate that the proposed reliability model of individual UAV help obtain the number of faults per component.

    Please comment on your answer.

3. The proposed UAV unit failure model helps find suitable values of the reliability of UAV components.

    Please comment on your answer.

4. The proposed UAV team failure model uses an appropriate ABS tool to run UAV team simulations.

    Please comment on your answer.

5. The proposed resilience model helps find out k-out-of-n of disabled UAVs that guarantees completion of mission.

    Please comment on your answer.

6. The proposed reliability model helps the user to plan surveillance missions using teams UAVs.

   Please comment on your answer.

7. The proposed UAV team failure model is easy to use, relies on adequate programming language, and has adequate graphic capabilities.

   Please comment on your answer.

8. The proposed UAV team failure model helps identify reconfiguration needs in the network's communications structure.

   Please comment on your answer.

9. The proposed UAV team failure model's use of NetworkX graph tool is useful in establishing possible communication restitution.

   Please comment on your answer.

10. The proposed UAV team failure model's communication restitution scheme reveals unit (UAV) positions per tick and the minimum distance (Dijkstra) between UAVs appropriate to carry out the rewiring of nodes after random disruptions.

    Please comment on your answer.

11. The proposed UAV team failure model helps obtain the total performance factor, σ; absorption factor, δ; recovery factor, ρ; volatility factor, ζ; and recovery time factor, τ, to determine the resilience R and R_{total} of the UAV team.

Please comment on your answer.

12. The proposed resilience model helps Identify the threshold of the parameters for failure of UAV missions.

Please comment on your answer.