Multiple Drones Coordination

Project Members (Group 32)

Tutku Gizem Guder

tguder2021@fau.edu

Brenden Martins

bmartins2013@fau.edu

Matthew Wyatt

mwyatt2023@fau.edu

Matthew Paternoster

mpaternoster2022@fau.edu

Tarek Kayali

tkayali2023@fau.edu

Advisor (Sponsor)

Xiangnan Zhong

xzhong@fau.edu

Florida Atlantic University
Department of Electrical Engineering and Computer Science Spring 2025

Table of Contents

I. Abstract (written by Brenden Martins)

This project presents the development of a Multi-Drone Coordination System (MDCS) within a realistic 3D simulation environment using Unreal Engine and Microsoft AirSim. The goal is to explore how multiple drones can be coordinated for efficient and scalable disaster response operations, including target detection and autonomous navigation. Our team built a customizable simulated environment that integrates mobile drone control, real-time positioning, and object targeting using camera feeds and predefined search patterns. We focused on implementing a flexible simulation infrastructure capable of supporting diverse disaster scenarios, such as collapsed urban areas and obstructed terrains. By leveraging open source standards and modular design, this project aims to reduce the risk to human responders, improve situational awareness in critical scenarios, and serve as a testing framework for future real world UAV deployments. The results highlight both the promise and the ongoing challenges in building AI-driven, collaborative drone systems for high stakes environments.

II. Summary (written by Tutku Gizem Guder)

Disasters like hurricanes and power outages pose significant challenges for emergency response teams, often delaying the ability to locate and assist affected individuals. This project aims to develop a multi-drone coordination system within a realistic 3D simulation environment to enhance target detection and improve disaster response efficiency. By providing a platform for testing and validating algorithms, this system contributes to more effective rescue operations, advance scientific research, and support future real-world deployments.

III. INTRODUCTION

Background (Rewritten by Tutku Gizem Guder, original text from the proposal)

With the rapidly advancing pace of technology, unmanned aerial vehicles (UAVs) or drones are now a multi-purpose means of improving disaster response functions. However, most current systems have been developed to run individual drones without delivering the cooperation between groups of many drones to carry out complex operations like cooperative search, real-time monitoring, and target discovery. Drone operations to reduce casualties and aid in disaster response have been widely studied and proven to be effective. This is observed by

Duong Van Hung, United Nations Development Programme Disaster Risk Reduction Programme Analyst:

The use of drones to assess damage after disasters has been adopted by many countries. Drones have proven their outstanding strength when assessing damage to facilities, houses, fields, and the likes after storms, floods and other disasters. Drones also provide concrete and clear evidence immediately after the assessment. In Vietnam, UNDP has equipped drones and organized drone training courses for six central provinces from 2022. Each province has set up a group of provincial experts who are regularly updated and provided with technical support. These drones enable experts to assess damage to infrastructure, houses, fields, and more with unparalleled efficiency. The immediate and concrete evidence collected by drones facilitates accurate analysis and informed decision-making.

Statement of the Problem (Written by Tutku Gizem Guder)

Delays or insufficient equipment and training in disaster response can determine the outcome of situations of life and death. Occurrence and strength of disasters like wildfires, earthquakes, and floods, can increase due to factors such as climate change, urbanization, and geopolitical instability. When faced with an increased rate and number of disasters, the traditional methods of emergency response experience ever expanding challenges. These challenges need to be tackled with an improved and up-to-date technologically-enhanced disaster response approach.

IV. SCOPE OF WORK

Overview (Written by Tutku Gizem Guder)

The project seeks to design a Multi-Drone Coordination System (MDCS) for assisting disaster response missions through optimal UAV deployment, coordination, and real-time decision assistance. The project creates a system that leverages recent advancements in autonomous UAV communication, task allocation, and environmental adaptability to improve emergency response efficiency. The project is designed in a number of stages to ensure a well-organized approach from research to implementation.

1. **Research and Analysis:** The team conducted an extensive literature review of UAV application in disaster relief, focusing on multi-drone coordination, communication protocols, and AI-based target detection. Additionally, relevant patents and industry standards were analyzed to inform system requirements.
2. **System Design and Development:** Based on research deliverables, the group designed the MDCS architecture, comprising communication frameworks, task allocation algorithms, and safety mechanisms. Designing hardware blocks, software frameworks, and machine learning models for real-time object detection and environmental adaptability is also part of the design process.
3. **Prototype Construction and Simulation:** A prototype system iis developed and tested in a simulated environment to gauge performance for disaster-like scenarios. The team calibrates UAV behaviors, validates communication robustness, and streamline resource allocation methods.
4. **Field Testing and Calibration:** After successful simulations, the MDCS prototype was deployed in controlled field tests to assess real-world functionality. Calibration efforts ensures system reliability in diverse environments, addressing challenges such as poor visibility and dynamic obstacle avoidance.

5. **Final Evaluation and Deliverables:** The project concludes with a thorough analysis of the MDCS system performance, noting findings, and suggested future enhancements. Final project reports, system blueprints, functional prototypes, and sponsor and faculty demonstrations are considered deliverables.

The team works collaboratively with the faculty project advisor and sponsor liaison throughout all stages to ensure alignment with project goals and approval of design decisions. This structured approach aims to demonstrate the feasibility and impact of a coordinated multi-drone system in disaster response scenarios.

Literature Review (written by Tutku Gizem Guder with the support of the rest of the team, based on the proposal)

The utilization of unmanned aerial vehicles (UAVs) for responding to disasters has experienced considerable improvement, with researchers continuing to study their applications, limitations, and possibilities for future development. The current literature review traces important studies, models, and findings that have influenced the design of the Multi-Drone Coordination System (MDCS). Academic research continues to demonstrate the efficiency of UAVs in enhancing disaster response. A research on Flying Ad Hoc Networks (FANETs) by Erdelj et al. (2017) illustrates the capability of drones to operate in difficult terrain, carry out airborne surveillance, and provide instant communication over extended distances. Multi-drone cooperation is supported heavily by FANETs since they support skill in distributed communications and autonomous choices. Similarly, Van Hung (2022) describes how UAVs have been an essential part of post-disaster damage evaluation, as determined in UNDP's Disaster Risk Reduction initiatives. Such studies show the importance of UAVs in acquiring rapid and accurate data for infrastructure evaluation and resource allocation purposes with less use of ground crew in hazardous situations.

While solitary drones were effective in disaster response, increasing the operation to many UAVs creates challenges in communication, task delegation, and collision avoidance. Y. Ma et al. (2020) address these issues through the development of autonomous cooperation algorithms for drones. The project concludes with a meticulous analysis of the MDCS. Scaling UAV missions to multi-drone systems is complicated by several problems, such as communication, task allocation, and collision avoidance. Effective communication must be guaranteed, which requires low-latency and real-time exchange of data among UAVs within uncertain and changing environments. Effective task allocation, further, also requires addressing by complex algorithms in the distribution of tasks across drones in a manner that optimally uses resources, maximizes productivity, and limits redundancy. Lastly, further safety precautions are needed to implement adaptive collision-avoidance systems capable of responding to dynamic obstacles and congested airspace.

Recent innovations in computer vision and machine learning have greatly enhanced the ability of drones to recognize and detect objects. The YOLO object detection algorithm by Redmon et al. (2016) has been particularly instrumental in building fast and precise detection systems for UAVs. These technologies have been adapted to be used in disaster scenarios where drones can easily detect survivors, hazardous materials, and structural loss in real time. Additionally, research by Tang et al. (2019) emphasizes the need to optimize the detection models under various environmental conditions, such as poor visibility, heavy rain, or heavy smoke, to improve the performance of UAVs in adverse disaster environments.

These innovations are a requirement for the assurance of the reliability of UAVs in emergency response situations. The patent discussed here, US10777051B1, provides a comprehensive design for UAV systems specifically for emergency response. It provides specifications of autonomous navigation, obstacle detection, and task achievement methods and offers valuable information on the practical application of drone technology. The modular payload and autonomous swarm features are particularly relevant to the goals of the MDCS project, which further supports the feasibility and potential impact of the proposed design. In conclusion, today's body of research offers a solid foundation to utilize multi-drone systems for disaster response. While single UAVs work effectively, one needs to scale larger to teams of multiple drones in order to handle the increasing sophistication of disaster scenarios. By harnessing the application of lessons drawn from current research, patents, and technological progress, the MDCS project aims to drive UAV potential to new heights, proving their potential to transform disaster response.

Alternative Solutions (Written by Tutku Gizem Guder)

Approaching this problem, we came up with many different solutions; here will be only a small number of them presented.

1. AI-Powered Satellite and Aerial Imaging

Solution: Employ high-resolution satellite imagery together with AI-driven analysis to assess disaster-affected areas in real-time. This system is able to combine inputs from drones, satellites, and fixed-wing aircraft for an all-round situational overview.

Advantages:

- Area-wide coverage without on-ground deployment.

- Ability to work in poor weather where drones will struggle.
- AI models are able to rapidly determine damage, infrastructure weakness, and survivors.

Challenges:

- Satellite image refresh may be delayed.
- Cloud cover or environmental conditions can interfere with satellite imagery.

2. Ground-Based Robotic Systems

Solution: Employ autonomous ground robots with cameras, thermal imaging sensors, and LiDAR to navigate through rubble, collapsed buildings, or poisonous environments. These robots may be used together with drones for a combined response system.

Advantages:

- Functional in locations where drones are prohibited, i.e., underground facilities or obstructed areas.
- Capable of carrying essential supplies to stranded individuals.
- Operate without putting human responders at risk in dangerous environments..

Challenges:

- Low mobility in extreme terrain.
- Battery life and communication problems in confined spaces..

3. IoT Sensor Networks for Disaster Monitoring

Solution: Deploy an Internet of Things (IoT) sensor network before disasters (e.g., along floodplains, earthquake fault lines) to record early warning signals and provide real-time environmental feedback.

Advantages:

- Early warning detection of events such as gas leaks, structural vulnerability, or approaching floodwaters.
- Reduces reliance on drones for initial surveys.
- Enables predictive analytics to maximize resource allocation

Challenges:

- Requires pre-deployment and maintenance.
- Only feasible in fixed locations, not mobile UAVs.

Evaluation ((Written by Tarek Kayali)

The evaluation process for the Multi-Drone Coordination System (MDCS) involved assessing various design approaches to determine the most effective and efficient solution for disaster response. Several key criteria were used to evaluate alternative solutions, ensuring that the selected approach aligns with project objectives and constraints.

Evaluation Criteria

To systematically assess the proposed solutions, the following evaluation criteria were considered:

1. **Effectiveness in Disaster Response**

    - The ability to quickly locate and identify targets (survivors, hazards, damaged infrastructure).
    - Real-time situational awareness and coordination capabilities.
2. **Scalability and Adaptability**
    - The system's ability to operate in various disaster scenarios (earthquakes, floods, wildfires).
    - Compatibility with different UAV models and potential integration with **additional technologies (satellites, ground robots).**
3. **Autonomy and AI Integration**
    - The effectiveness of AI-based object detection (YOLOv5) in identifying survivors and hazards.
    - The system's ability to autonomously allocate tasks and avoid collisions in real-time.
4. **Communication Reliability**
    - Low-latency data exchange and coordination among multiple drones.
    - Resilience in environments with limited or disrupted network infrastructure.
5. **Cost and Resource Efficiency**
    - Hardware and software affordability in comparison to other solutions.
    - Energy consumption and battery life optimization for extended operation.
6. **Ease of Implementation and Deployment**
    - Complexity of integrating the system with existing emergency response operations.
    - Required training and user-friendliness of the mission planning interface.

Evaluation of Alternative Solutions

Three alternative solutions were initially considered:

1. **AI-Powered Satellite and Aerial Imaging**
   - Strengths: Large-area coverage, no need for UAV deployment in hazardous zones.
   - Weaknesses: Delayed image refresh rates, cloud cover limitations, and reliance on external satellite data.
2. **Ground-Based Robotic Systems**
   - Strengths: Can operate in enclosed spaces and hazardous areas where drones struggle.
   - Weaknesses: Mobility issues in rough terrains, slower than aerial drones, limited real-time overview.
3. **IoT Sensor Networks for Disaster Monitoring**
   - Strengths: Can provide early disaster detection and environmental data.
   - Weaknesses: Requires pre-deployment, limited real-time adaptability, high installation costs.

Final Evaluation and Selection

After comparing these solutions against the evaluation criteria, the Multi-Drone Coordination System (MDCS) was determined to be the most viable solution. It offers:

- High efficiency in disaster response with real-time data collection and AI-enhanced object detection.
- Scalability and adaptability to different disaster types.

- Strong autonomy and coordination through AI-driven task allocation and collision avoidance.
- Lower costs and faster deployment compared to satellite and ground-based alternatives.

Decision (*Brenden Martins*)

With AirSim as the foundation for our 3D Simulated Environment project with a Multiple Drone Coordinated System, we were tasked with figuring out how to best utilize these tools to create a meaningful and impactful solution. After careful consideration, our team decided on leveraging drones within a 3D simulated environment to improve disaster response efforts. After seeing the disasters that occurred with Hurricane Helene, the fires in Maui, and now the fires in the Palisades, it is evident now, more than ever, that this is an important direction to work towards for our project. We recognized that AirSim could serve as a realistic testing ground for developing and refining MDCS operations before real-world deployment. By simulating disaster ridden areas, we could explore how drones can coordinate autonomously, or manually, detect critical targets, and assist in emergency response.

Key Factors in Our Decision

1. **Simulating Real Disaster Scenarios**
    a. Instead of using AirSim for general drone testing, we chose to simulate realistic disaster environments where drones can be tested in extreme conditions.
    b. By simulating disaster ridden environments, we can get an idea of how drones would perform in actual emergency situations and scope out the area, before sending in people, to check the environment and map out the best course.

2. **Testing Multi-Drone Coordination in Complex Environments**
    a. Coordinating multiple drones in disaster scenarios introduces unique challenges, such as avoiding collisions, optimizing search patterns, and managing communication.
    b. We decided to use AirSim to test multi-drone collaboration, ensuring the unmanned drones can efficiently divide tasks, cover large areas, and relay information to first responders.
    c. This approach allows us to refine navigation algorithms and AI-based task allocation in a controlled setting before real-world application.
3. **Enhancing Target Detection and Search-and-Rescue Capabilities**
    a. One of the biggest challenges in disaster response is quickly identifying survivors, hazards, and structural damage.
    b. We are aiming to integrate AI-powered object detection within the AirSim simulation to test how drones can locate and classify critical targets.
    c. This approach ensures that unmanned drones can provide real-time insights to emergency teams, improving response efficiency.
4. **Utilizing AirSim for Performance Optimization Before Real-World Deployment**
    a. A major challenge in real-world drone deployment is ensuring reliability in unpredictable environments.
    b. Instead of testing in live disaster zones, which is costly and dangerous, we chose to use AirSim to refine the unmanned drone's movement, obstacle avoidance, and communication strategies in a controlled and safe space.
    c. This allows us to identify and resolve issues before field testing, reducing risk and operational failures.
5. **Possible Future Expansion: Integration with GIS & Real-World Mapping**

a. While our current approach relies on simulated environments, we've also looked into incorporating real-world mapping data in future iterations.

b. By integrating GIS mapping and USGS 3DEP elevation data, we can make our simulation more geospatially accurate and adaptable for real disaster response efforts.

c. This would further ensure that our MDCS system is scalable and applicable to a variety of real-world disaster scenarios.

V. IMPLEMENTATION DETAILS

A. System Specifications and Functionalities (*Matthew Paternoster,* Updated by Tutku Gizem Guder and Matthew Wyatt)

The Multiple Drones Coordination System (MDCS) is designed to simulate coordinated drone operations within a 3D environment. This section provides a detailed breakdown of the system specifications and core functionalities across its various subsystems.

1. System Specifications

Hardware Requirements (Simulation-Based) Since the MDCS operates exclusively within a 3D simulation environment, the hardware requirements are focused on the computational infrastructure needed to run the simulation and related subsystems effectively.

- **High-Performance Computing:** Multi-core processors and high-end GPU for real-time 3D rendering and physics simulations.
- **Simulation Environment:** Microsoft AirSim running on Unreal Engine to provide realistic terrain and weather conditions.
- **Network Infrastructure:** Stable Wi-Fi connections to simulate communication between drones and control interfaces.

Software Components

- **Programming Languages:** Python (drone scripting, AI models), C++ (performance-critical algorithms), JavaScript (UI).

- **APIs & Frameworks:** AirSim API for drone control, YOLOv5 for object detection.
- **Database:** MySQL for storing telemetry and mission data.

2. Functionalities by Subsystem

a. Drone Control Subsystem (C)

- **Coordinated Rescue Operations:** Manages drone deployment for synchronized search patterns.
- **Collision Avoidance:** Utilizes RRT and potential field algorithms for safe navigation.
- **Emergency Protocols:** Activates autonomous safe landing in critical situations.

b. User Interface Subsystem (A)

- **Mission Planning and Execution:** Provides an intuitive interface for defining, starting, and monitoring missions.
- **Real-Time Data Display:** Displays telemetry and mission status via interactive maps.
- **Accessibility & Compatibility:** Support with responsive design for desktop users.

c. Data Management Subsystem (F)

- **Data Storage and Retrieval:** Utilizes a MySQL database to store and manage sensor data, mission logs, and system telemetry. The database schema is optimized for fast queries, ensuring quick access to historical and real-time data.
- **API Integration**: Exposes RESTful APIs for querying and updating the database, allowing seamless integration with the user interface and external systems. These APIs

support operations such as retrieving mission history, storing telemetry data, and fetching drone status.

- **Data Backup and Recovery**: Implements automated backup routines to ensure data integrity and recovery in case of system failures, minimizing downtime and data loss.

d. Communication Subsystem (B)

- **Low-Latency Communication:** Ensures real-time data exchange between drones, the user interface, and the backend using ROS 2 (Robot Operating System 2) for efficient message passing.
- **API Integration**: Provides RESTful APIs for external systems to interact with the drones, enabling mission updates, telemetry retrieval, and command execution. These APIs are secured with token-based authentication to ensure data integrity.
- **Protocol Support**: Implements MAVLink emulation for drone telemetry and command communication, ensuring compatibility with industry-standard protocols. Additionally, custom Python scripts interface directly with AirSim APIs for fine-tuned control and simulation management.

e. Target Detection Subsystem (E)

- **Object Recognition:** Uses the drone cameras and distance detection to detect and classify objects from drone camera feeds.

f. Simulation Management Subsystem (D)

- **3D Environment Simulation:** Configures AirSim to mimic real-world conditions for realistic training.

- **Scenario Testing:** Allows simulation of different disaster scenarios to evaluate drone behavior and system performance.

Communication Protocols

- **MAVLink Emulation:** Simulates drone telemetry and command communication.
- **Custom Python Scripting:** Directly interfaces with AirSim APIs for fine-tuned control.

The MDCS implementation emphasizes modularity, scalability, and real-time responsiveness to support disaster response training and mission planning in a simulated environment.

B. Overall System Design with Block Diagram(s)

The design of the Multiple Drone Coordination System (MDCS) is structured to ensure efficient communication, coordination, and operation of multiple drones within a disaster response scenario. This section presents the overall system design through various diagrams, each providing insights into different aspects of the system's functionality and architecture.

1. System Block Diagram

The system block diagram provides a high-level overview of the MDCS architecture, illustrating the interactions between its key subsystems. It highlights how data flows between components, including the drone control, communication, user interface, simulation management, target detection, and data management subsystems.

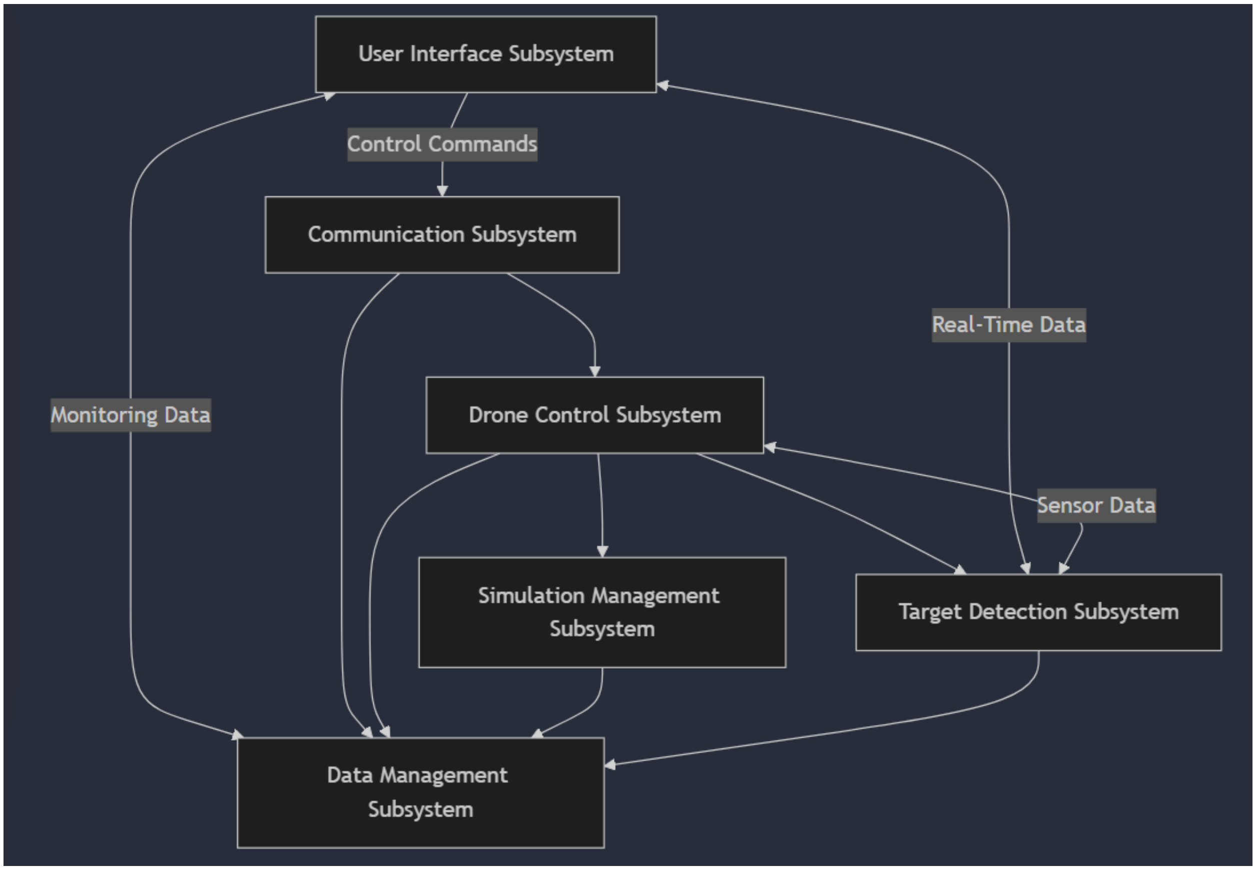

Figure 1 shows the system block diagram of the project

Description:

- **Drone Control Subsystem (C):** Handles drone navigation, collision avoidance, and mission execution.
- **Communication Subsystem (B):** Manages real-time communication between drones, the simulation environment, and the user interface.
- **User Interface Subsystem (A):** Provides operators with intuitive tools for mission planning, monitoring, and control.
- **Simulation Management Subsystem (D):** Generates a virtual environment for testing drone behavior.
- **Target Detection Subsystem (E):** Processes visual data to identify and classify targets.
- **Data Management Subsystem (F):** Stores and retrieves operational and mission data.

2. System State Diagram

The system state diagram illustrates the various states the MDCS transitions through during its lifecycle, from initialization to mission execution and termination. It helps in understanding the dynamic behavior of the system under different operational scenarios.

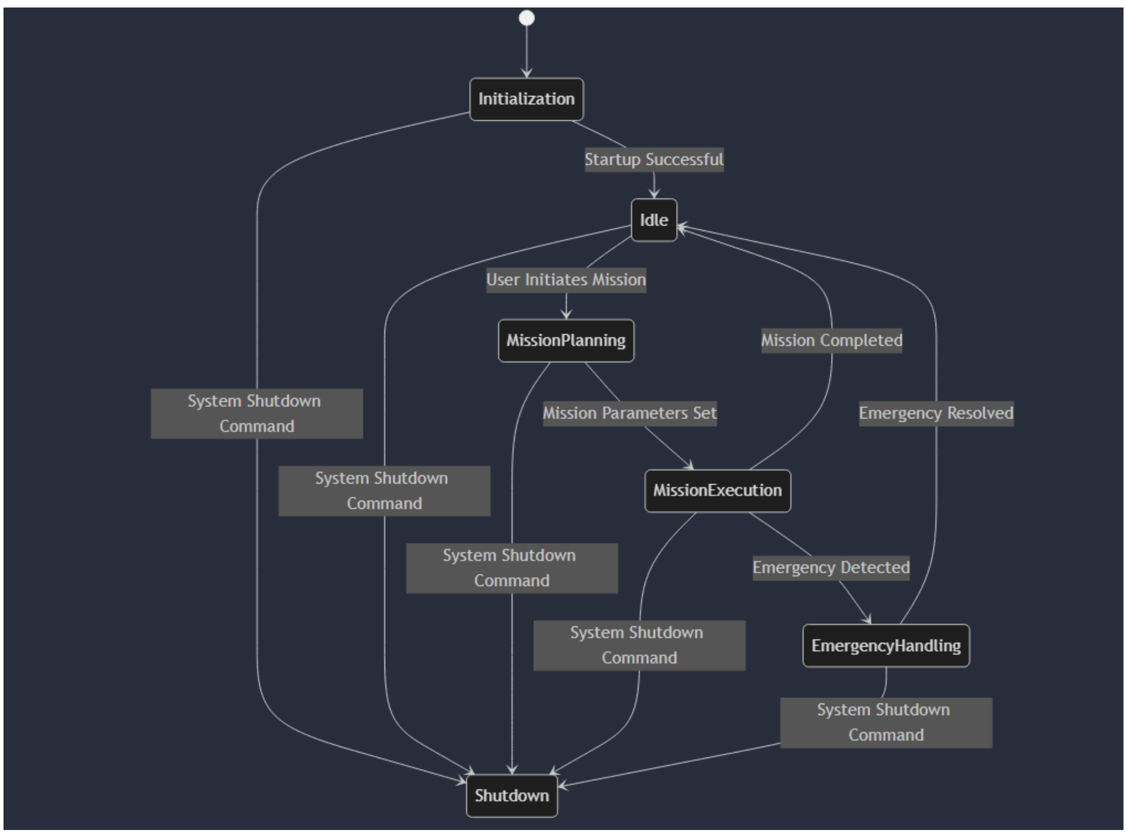

Figure 2 shows the system state diagram of the project

- States:
    - Initialization: System startup phase.
    - Idle: System is waiting for user input or missions.

- Mission Planning: Configuring mission parameters.
    - Mission Execution: Active phase where drones carry out the mission.
    - Emergency Handling: Managing unexpected emergencies during missions.
    - Shutdown: System is powering down.
- Transitions:
    - Initialization → Idle: Upon successful startup.
    - Idle → Mission Planning: When a user initiates a mission.
    - Mission Planning → Mission Execution: After mission parameters are set.
    - Mission Execution → Emergency Handling: If an emergency is detected during execution.
    - Mission Execution → Idle: Upon successful mission completion.
    - Emergency Handling → Idle: After resolving the emergency.
    - Any State → Shutdown: When a shutdown command is received.

3. Subsystem State Diagram

The subsystem state diagram provides a more granular view of the internal state transitions within individual subsystems, particularly the drone control and communication components. It helps in understanding how specific functionalities are managed during mission execution.

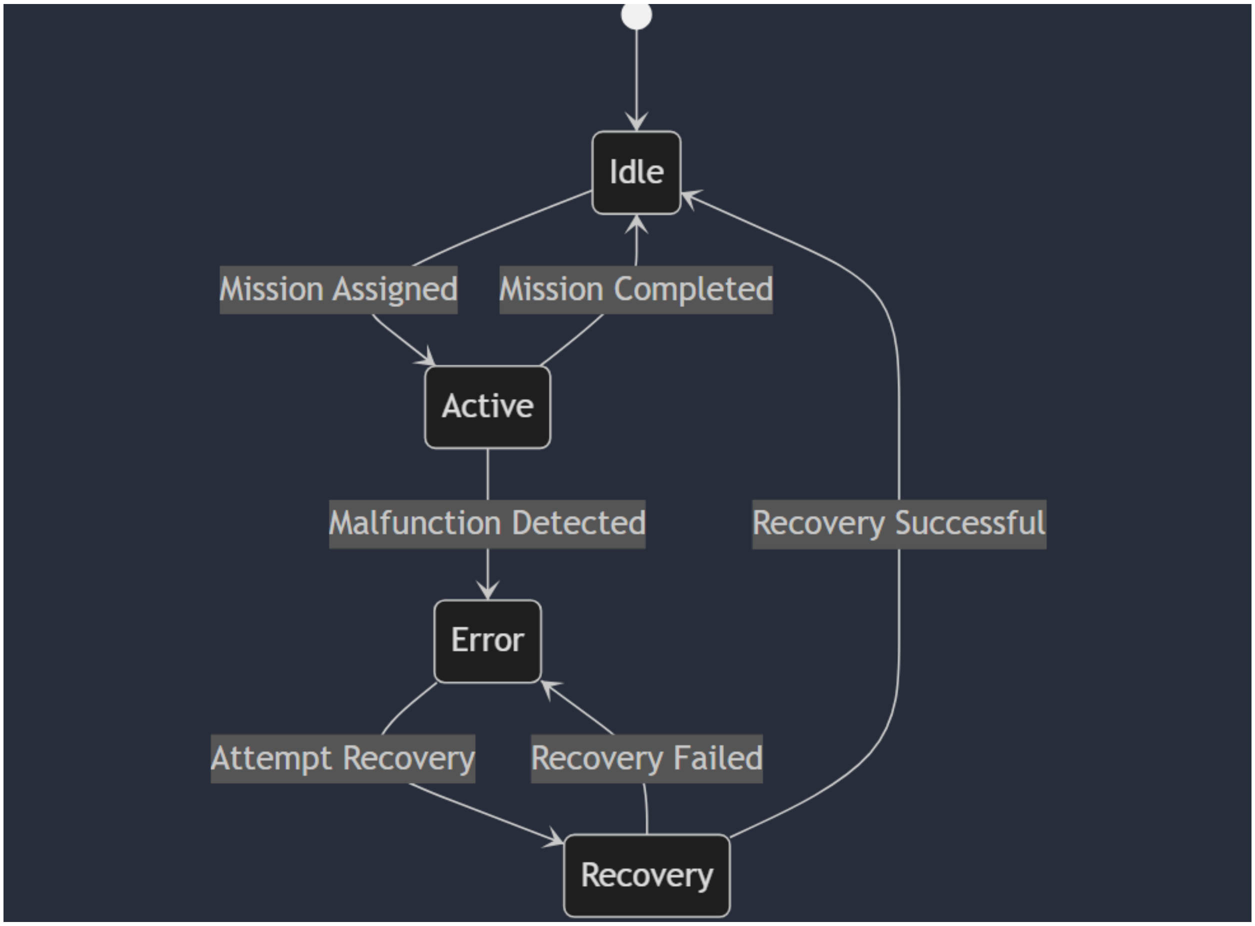

Figure 3 shows the subsystem state diagram of the project

- States:
  - Idle: Waiting for mission assignments.
  - Active: Currently executing a mission.
  - Error: A malfunction or anomaly has been detected.
  - Recovery: Attempting to resolve the error.

- Transitions:
  - Idle → Active: When a mission is assigned.
  - Active → Idle: Upon mission completion.

- Active → Error: If a malfunction is detected during mission execution.
- Error → Recovery: When attempting to fix the issue.
- Recovery → Idle: If recovery is successful.
- Recovery → Error: If recovery fails and the system remains in an error state.

4. Design Considerations

The diagrams collectively illustrate the MDCS's modular structure, emphasizing scalability, reliability, and ease of maintenance. The block diagram shows the static component relationships, while the state diagrams reveal dynamic operational flows. This combined perspective ensures that developers and operators alike can grasp both high-level architecture and detailed subsystem behavior effectively.

By utilizing these diagrams, the MDCS can be better understood, analyzed, and tested throughout its development lifecycle.

C. Use Cases

Use Case 1: Coordinated Rescue Mission

Description: In a hurricane scenario, multiple drones are deployed to perform a coordinated rescue mission. The user initiates the mission through the User Interface Subsystem, specifying the affected area and objectives.

State Transitions:

1. Initialization → Idle: System starts and awaits mission commands.

2. Idle → Mission Planning: User initiates the rescue mission.
3. Mission Planning → Mission Execution: Mission parameters are set, and drones are deployed.
4. Mission Execution → Idle: Rescue mission is successfully completed, and drones return to idle state.

Detailed Flow:

- User inputs mission details via the interface.
- System assigns tasks to each drone based on their capabilities and positions.
- Drones navigate to designated locations, avoiding obstacles and coordinating with each other.
- Upon successful completion, drones report back, and the system updates the mission status to completed.

Use Case 2: Emergency Collision Avoidance

Description: During a mission, two drones are on a collision course due to unexpected environmental changes. The system detects the potential collision and initiates collision avoidance protocols.

State Transitions:

1. Mission Execution → Emergency Handling: Potential collision detected.
2. Emergency Handling → Mission Execution: Collision avoidance maneuvers successfully executed.

3. Mission Execution → Emergency Handling: If collision avoidance fails, initiate emergency protocols.
4. Emergency Handling → Idle: Drones return to a safe state post-resolution.

Detailed Flow:

- The Drone Control Subsystem continuously monitors drone positions and trajectories.
- A potential collision is detected based on real-time data.
- The system commands the involved drones to alter their paths to prevent collision.
- If successful, drones resume their missions; otherwise, emergency protocols are activated to ensure safety.

D. Technology and Technical Standards Used (Written by Brenden Martins)

The development of the Multi-Drone Coordination System (MDCS) is supported by a wide array of hardware, software, and communication technologies, all of which adhere to recognized technical and industry standards. The goal is to ensure reliability, interoperability, and future scalability of the system for real-world disaster response.

Simulation Environment

- **Microsoft AirSim** serves as the core simulation platform for drone testing and environmental modeling. While not governed by a formal standards body, it follows internal Microsoft development practices and is widely accepted for drone simulation.

- **Unreal Engine 4.27** was used to create the 3D simulation environment. Though proprietary, it supports industry standards such as **OpenGL (ISO/IEC 14882)**, **Vulkan (ISO/IEC 23360)**, and **DirectX12** for rendering, and **PhysX** for physics simulation.

Objection Detection

- **OpenCV** supports real-time object detection for identifying targets like survivors or structural damage.
- Key standards include:
  - **ISO/IEC 1449602**: Coding of audio-visual objects (image/video compression and processing).
  - Standards bodies: **IEEE, ISO, and Khronos Group.**

Sensor Technologies (LiDAR)

- **ASTM E3125-17**: Standard for evaluating point-to-point distance measurements.

Communication and Data Management

- **Wi-Fi (IEEE 802.11-2020)** standard governs wireless communication simulations between drones and the ground system.
- **MySQL** database follows **ISO/IEC 9075**: Database languages - SQL.
- **Express.js API** and **JavaScript (ECMA-262)** support backend data transfer and web-based communication between subsystems.
- **DNS protocol** used in deployment architecture is based on **RFC 1035** by the **IETF**.

User Interface and Front-End Technologies

- **React.js** was used to build the UI, aligning with **W3C Web Components** standards for structure and accessibility.
- These technologies ensure component modularity, real-time feedback, and user scalability.

Drone Subsystem and Control Standards

- The drone control logic is being developed with reference to:
  - **IEEE 1073-2018**: Control and Data interfaces for drones.
  - **RTCA DO-178C**: Software considerations in airborne systems certification.
  - **ISO 21384-3:2019**: Operational procedures for drone systems.

By utilizing these technologies and adhering to established standards, the MDCS ensures its components are robust, interoperable, and capable of evolving into a deployable real-world solution for coordinated drone-based disaster response.

E. Testing and Performance Evaluation (Written by Brenden Martins, Updated by Tutku Gizem Guder)

Testing and performance evaluation are critical phases in the development of the MDCS, especially since the system is intended for use in high-stakes disaster scenarios. The testing was conducted primarily within the AirSim simulation environment using a custom-built 3D world designed in Unreal Engine 4.27.

Initial Simulation Tests

- Manual drone control using a keyboard-based input mapping was tested to ensure user fallback options were available when the standard control scheme encountered issues.

- Collision detection was validated through mapped building meshes and obstacles in the simulation to ensure accurate UAV path planning.

- Environmental dynamics (such as rain, fog, and lighting) were implemented via AirSim's console system and evaluated for drone responsiveness under adverse conditions.

Subsystem Integration Testing

- Early-stage integration between the simulation environment, drone control logic, and user interface was evaluated for consistency and stability.

- React.js interface was tested for live feedback accuracy, ensuring real-time status updates and command responsiveness.

Performance Benchmarks

- Frame rates, simulation lag, and control input latency were measured under varying levels of environmental detail and drone activity.

- Test results showed that the system maintains stable performance when managing multiple drones, though optimization is still needed for large-scale scenarios.

Testing

- AI-based object detection integration using Python and OpenCV was tested with static and dynamic targets.

- Autonomous multi-drone coordination scenarios were implemented to evaluate swarm behavior, path optimization, and task assignment logic.

Each test serves as a step toward refining the system to ensure it is both technically sound and operationally viable for real-world deployment. Testing results, logs, and benchmark data are being recorded to support future evaluations and scalability assessments.

VI. CONCLUSION

The deployment of the Multi-Drone Coordination System (MDCS) has demonstrated the capability to enhance disaster response efficiency through coordinated UAVs in a real-world 3D simulation environment. Through the use of state-of-the-art AI-based object detection, real-time communication, and autonomous task allocation, the project has demonstrated that multi-drone systems can efficiently detect and locate critical targets in disaster scenarios. AirSim's usage for simulation permitted testing under controlled conditions, with fewer risks and improved system performance prior to possible real-world application.

While the MDCS has a lot to offer, it also has challenges to overcome, such as improving communication reliability in networks that are disrupted and being more flexible in varied disaster settings. Later development could explore real-world mapping integration, further improved AI capabilities toward more advanced target identification, and scaling the system for more far-reaching disaster scenarios. Overall, the project has performed well in establishing the building blocks of an adaptive, scalable, and effective multi-drone disaster relief solution.

REFERENCES

[1][3] Duong, H. V. (2023, May 20). *Drones for assessment of disaster damage and impact - revolutionizing disaster response*. UNDP. https://www.undp.org/vietnam/blog/drones-assessment-disaster-damage-and-impact-revolutionizing-disaster-response

[2] Erdelj, M. et al. (2017, June 3). *Wireless sensor networks and Multi-UAV Systems for Natural Disaster Management*. Computer Networks. https://www.sciencedirect.com/science/article/abs/pii/S1389128617302220

[4]Y. Ma, Y. Zhao, S. Bai, J. Yang and Y. Zhang, "Collaborative task allocation of heterogeneous multi-UAV based on improved CBGA algorithm," *2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Shenzhen, China, 2020, pp. 795-800, doi: 10.1109/ICARCV50220.2020.9305380.

[5]Redmon, J., et al (n.d.). You only look once: Unified, real-time object detection. https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Redmon_You_Only_Look_CVPR_2016_paper.pdf

[6]Tang, J., Duan, H. & Lao, S. Swarm intelligence algorithms for multiple unmanned aerial vehicles collaboration: a comprehensive review. *Artif Intell Rev* 56, 4295–4327 (2023). https://doi.org/10.1007/s10462-022-10281-7