Requirements Specifications and System Level Design

Matthew Paternoster, Matthew Wyatt, Tutku Gizem Guder, Brenden Martins, Tarek Kayali

Group 32

Florida Atlantic University
EGN 4950C Engineering Design 1

Drs. Waseem Asghar, KwangSoo Yang, Xiangnan Zhong
November 2, 2024

Table of Contents

A. Requirements Specifications

1. Introduction

This document outlines the requirements specifications for the Multiple Drones Coordination project. It includes, detailing functional, usability, and safety requirements necessary to achieve the project's objectives.

2. Overall Description

The Multiple Drones Coordination platform aims to facilitate seamless coexistence and coordination of multiple drones within a realistic 3D simulation environment. The platform supports target detection, real-time monitoring, and coordinated rescue operations in various disaster scenarios.

2.1 Key Terms + Definitions

- **MDCS (Multiple Drone Coordination System)**: The system designed to manage and coordinate the operation of multiple UAVs within a simulated environment, in this case, for disaster response scenarios.
- **UAV (Unmanned Aerial Vehicle)**: Aircraft with no human's onboard, commonly referred to as reconnaissance drones, used in the MDCS for recon, rescue, and communication tasks.
- **Simulation Environment**: Realistic 3D virtual space created within the confines of the MDCS to simulate real world conditions, including terrain, weather, and obstacles, allowing for controlled testing and training of UAV operations.
- **Target Detection**: Process which UAVs identify and locate specific objectives or persons on the ground within the simulation, using sensors and algorithms for accurate recognition.
- **Collision Avoidance**: Mechanism which prevents UAVs from crashing into each other or obstacles in the MDCS by utilizing algorithms to detect and avoid potential collisions.
- **Coordinate Rescue Operation**: Collaborative mission involving multiple UAVs working together to locate and assist targets in scenarios like natural disasters or power outages.
- **Emergency Protocols**: Predetermined procedures within the MDCS that are automatically activated during critical system failures, such as loss of communication or power, ensuring UAVs return to a safe state.
- **Subsystem**: Distinct functional module with the MDCS, such as Drone Control, Communication, or Target Detection, that performs specific tasks to support overall system objectives.

2.2 Scope and Purpose

2.2.1 Scope

- The Multiple Drones Coordination System (MDCS) is created with the ability to enable seamless and coordinated operation of multiple unmanned aerial vehicles (UAVs) in simulated environments.
- Support emergency response scenarios, including, but not limited to:
  - Natural disasters (hurricanes, earthquakes, etc.
  - Power outages
  - Severe weather events
- Providing functionality for:
  - Ground target detection and identification
  - Coordinated rescue operations
  - Real-time monitoring and control through a mobile interface
- Serve as a testbed for:
  - Multi-drone collaboration
  - Collision avoidance
  - Task management under various simulated environmental conditions
- Potential applications in disaster management and relief efforts.

2.2.2 Purpose

- Define the detailed requirements for the Multiple Drones Coordination System (MDCS).
- Serve as a guiding document to:
  - Developers
  - Testers
  - End-users and other stakeholders
- Outline functional, usability, and safety requirements to meet system goals.
- Enable alignment of efforts across stakeholders during development.
- Support project planning and enhance communication among team members.
- Provide a foundation for testing and validation to ensure the MDCS meets specified requirements.

3. Specific Requirements

3.1 Functional Requirements

3.1.1 Simulated Environment

- The system will provide a highly realistic 3D simulation environment that accurately represents real-world conditions for drone operations.
- The system will support the simultaneous operation of multiple UAVs within the simulation, allowing for dynamic interactions among them.

3.1.2 Drone Coordination and Control

- The system will enable real-time communication and coordination among multiple UAVs to coordinate actions effectively.
- The system will implement collision avoidance mechanisms to prevent drones from colliding in the simulated environment.
- The system will allow for dynamic assignment and reassignment of tasks among drones based on mission requirements and real-time data.

3.1.3 Target Detection

- The system will incorporate algorithms that enable drones to detect and identify target objectives on the ground with high accuracy.
- The system will process and display detection results in real-time, providing operators with immediate feedback on target status.

3.1.4 Rescue Operations

- The system will facilitate coordinated rescue missions involving multiple drones operating collaboratively in various disaster scenarios.
- The system will support mission planning and execution for emergencies, including hurricanes, natural disasters, and power outages.

3.1.5 Data Management

- The system will efficiently handle large volumes of real-time data generated by multiple drones during operations.
- The system will ensure quick processing and secure storage of operational data, enabling timely access for analysis and reporting.

3.1.6 User Interface

- The system will provide an intuitive, real-time interface accessible via mobile devices enabling operators to monitor and control UAV operations remotely.
- The interface will clearly, and concisely, display critical information, such as UAV status, target locations, and mission progress .

3.2 Usability Requirements

3.2.1 User-Friendly Interface

- The system will feature an intuitive user interface that is easy to navigate and understand for users with varying levels of technical expertise.
- The interface will display real-time data and status updates in a clear and organized manner.

3.2.2 Accessibility

- The system will be compatible with various mobile devices, including smartphones and tablets, to ensure broad usability.

- The system will support multiple user accounts with varying access levels to accommodate different operational roles.

3.2.3 Responsiveness

- The system will respond to user inputs and control commands within a predefined time frame to ensure smooth operation, with a target response time of less than 2 seconds.

3.3 Safety Requirements

3.3.1 Collision Avoidance

- The system will implement robust collision avoidance algorithms to prevent drones from crashing into each other or obstacles within the simulation environment.
- The system will include fail-safes that automatically transitions drones to a safe state in case of system malfunctions.

3.3.2 Data Security

- The system will ensure that all data transmitted between drones and the user interface is encrypted to prevent unauthorized access.
- The system will implement authentication mechanisms to verify user identities before granting access to control functions.

3.3.3 Emergency Protocols

- The system will have predefined emergency protocols that activate in the event of critical failures, such as loss of communication or power outages.
- The system will allow drones to autonomously return to a designated safe zone or land safely upon activation of emergency protocols.

B. System Level Design

1. System Overview

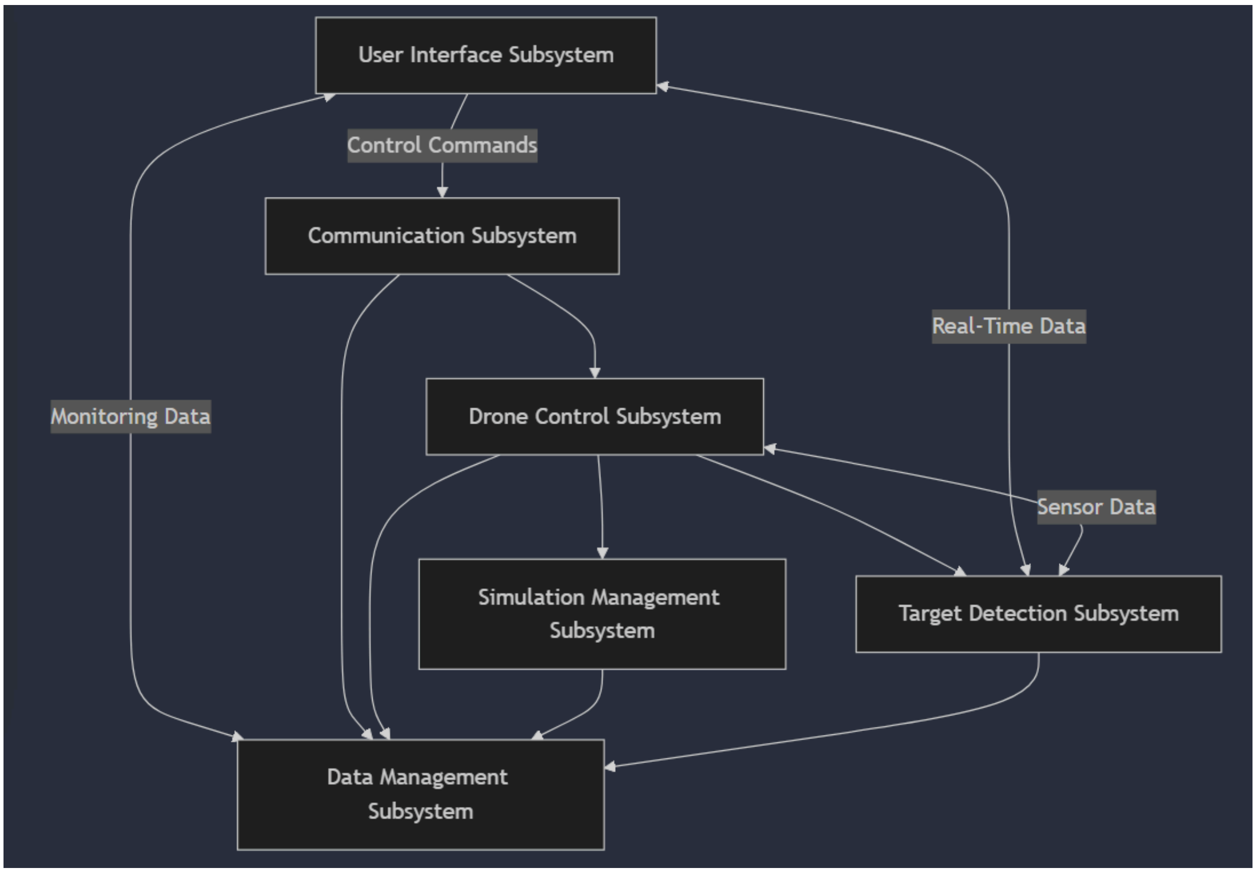
- The Multiple Drones Coordination system is designed to enable the seamless operation and coordination of multiple drones within a realistic 3D simulation environment. The system comprises several key subsystems that work collaboratively to achieve the project's objectives, including simulation management, drone control, target detection, user interface, data management, and communication. Each subsystem plays a unique role in ensuring that UAVs can safely and efficiently perform tasks related to detection, coordination, and rescue operations in disaster scenarios.

2. System Architecture

- Below is a description of the system architecture, outlining the main subsystems and their interactions.

2.1 Block Diagram

- The block diagram illustrates the MDCS's main subsystems and the flow of information and control signals between them. Each subsystem has a specific role, interacting with others through data and control flows to achieve system goals.



Information and Control Flows:

- **User Interface Subsystem (A)**: Sends control commands to the Communication Subsystem (B) and receives real-time monitoring data from the Data Management Subsystem (F).
- **Communication Subsystem (B)**: Facilitates communication between the User Interface Subsystem (A) and the Drone Control Subsystem (C), and interacts with the Data Management Subsystem (F).
- **Drone Control Subsystem (C)**: Receives control commands via the Communication Subsystem (B), communicates with the Simulation Management Subsystem (D) for environmental data, and interacts with the Target Detection Subsystem (E) for sensor data. It also interacts with the Data Management Subsystem (F).
- **Simulation Management Subsystem (D)**: Provides environmental data to the Drone Control Subsystem (C) and interacts with the Data Management Subsystem (F).
- **Target Detection Subsystem (E)**: Receives sensor data from the Drone Control Subsystem (C), processes it, and sends real-time data back to the User Interface Subsystem (A). It also interacts with the Data Management Subsystem (F).

- **Data Management Subsystem (F)**: Acts as a central hub for storing and managing data from all other subsystems.

Team Member Responsibilities:

Brenden Martins: Simulation Management Subsystem
Tarek Kayali: Drone Control Subsystem
Matthew Paternoster: Target Detection Subsystem
Tutku Gizem Guder: User Interface Subsystem
Matthew Wyatt: Data Management Subsystem
Matthew Wyatt: Communication Subsystem
*Subject to change

3. Subsystem Descriptions

Each subsystem is responsible for specific functionalities within the overall system. Below are the descriptions for each subsystem, authored by the respective team members.

3.1 Simulation Management Subsystem

- The Simulation Management Subsystem is responsible for creating and maintaining the 3D simulation environment where drones operate. It models realistic environmental conditions, including terrain, weather, and obstacles, to provide a comprehensive testing ground for drone coordination and target detection algorithms. This subsystem ensures that the simulation parameters can be adjusted to mimic various disaster scenarios, enabling thorough testing and validation of drone performance under different conditions.

3.1.1 Purpose

- Creates and maintains the 3D simulation environment in which UAVs operate.

3.1.2 Functionality

- Models realistic environmental conditions such as terrain, weather, and obstacles to provide a comprehensive testing ground.
- Supports testing of coordination and target detection under different disaster scenarios.

3.2 Drone Control Subsystem

- The Drone Control Subsystem manages the navigation, stability, and coordination of each drone within the simulation. It implements advanced control algorithms that facilitate smooth movement, collision avoidance, and task execution. This subsystem ensures that drones can autonomously perform assigned missions while maintaining real-time communication with other drones and the central coordination system. It also handles the dynamic allocation of tasks to optimize mission efficiency.

3.2.1 Purpose

- Manages UAV navigation, stability, and coordination within the simulation environment.

3.2.2 Functionality

- Implements control algorithms for collision avoidance, dynamic task assignment, and mission execution.
- Ensures UAVs can autonomously perform assigned tasks while communicating with other UAVs and subsystems.

3.3 Target Detection Subsystem

- The Target Detection Subsystem processes data from various sensors, such as cameras, LIDAR, and GPS, to identify and locate target objectives on the ground. Utilizing machine learning algorithms, this subsystem enhances the accuracy and reliability of target identification under varying environmental conditions. It provides real-time detection results to the User Interface Subsystem, enabling informed decision-making during rescue operations.

3.3.1 Purpose

- Identifies and locates specific ground objectives using onboard sensors and algorithms.

3.3.2 Functionality

- Processes data from sensors (e.g., cameras, LIDAR) to detect targets, providing real-time detection data to the User Interface Subsystem.
- Utilizes machine learning models to enhance detection accuracy across varied conditions.

3.4 User Interface Subsystem

The User Interface Subsystem offers an intuitive, real-time interface accessible via mobile devices. It allows users to monitor drone operations, view real-time data, and issue control commands remotely. The interface displays critical information, such as drone status, target locations, and mission progress, in a clear and organized manner. Additionally, it provides tools for configuring missions, managing drone fleets, and reviewing operational logs.

3.4.1 Purpose

- Provides an accessible interface for operators to monitor and control UAVs remotely.

3.4.2 Functionality

- Displays UAV status, mission progress, and target locations in real-time. Allows users to configure mission parameters, manage UAV fleets, and review operation logs.

3.5 Data Management Subsystem

The Data Management Subsystem is responsible for handling large volumes of real-time data generated by multiple drones. It ensures efficient data storage, processing, and retrieval through the use of robust database systems. This subsystem supports data analysis, reporting, and archival, enabling comprehensive monitoring and evaluation of drone performance. It also ensures data integrity and security throughout the data lifecycle.

3.5.1 Purpose

- Manages the storage, processing, and retrieval of real-time data generated by UAVs.

3.5.2 Functionality

- Ensures secure and efficient data handling, supporting analysis and reporting.
- Maintains data integrity and facilitates timely access to critical information.

3.6 Communication Subsystem

The Communication Subsystem facilitates reliable and efficient data exchange between drones and the User Interface Subsystem. It implements secure wireless communication protocols to ensure seamless coordination and information sharing. This subsystem manages network connectivity, data transmission, and protocol compliance, enabling real-time synchronization of drone actions and user commands. It also monitors communication quality and handles reconnections in case of signal disruptions.

3.6.1 Purpose

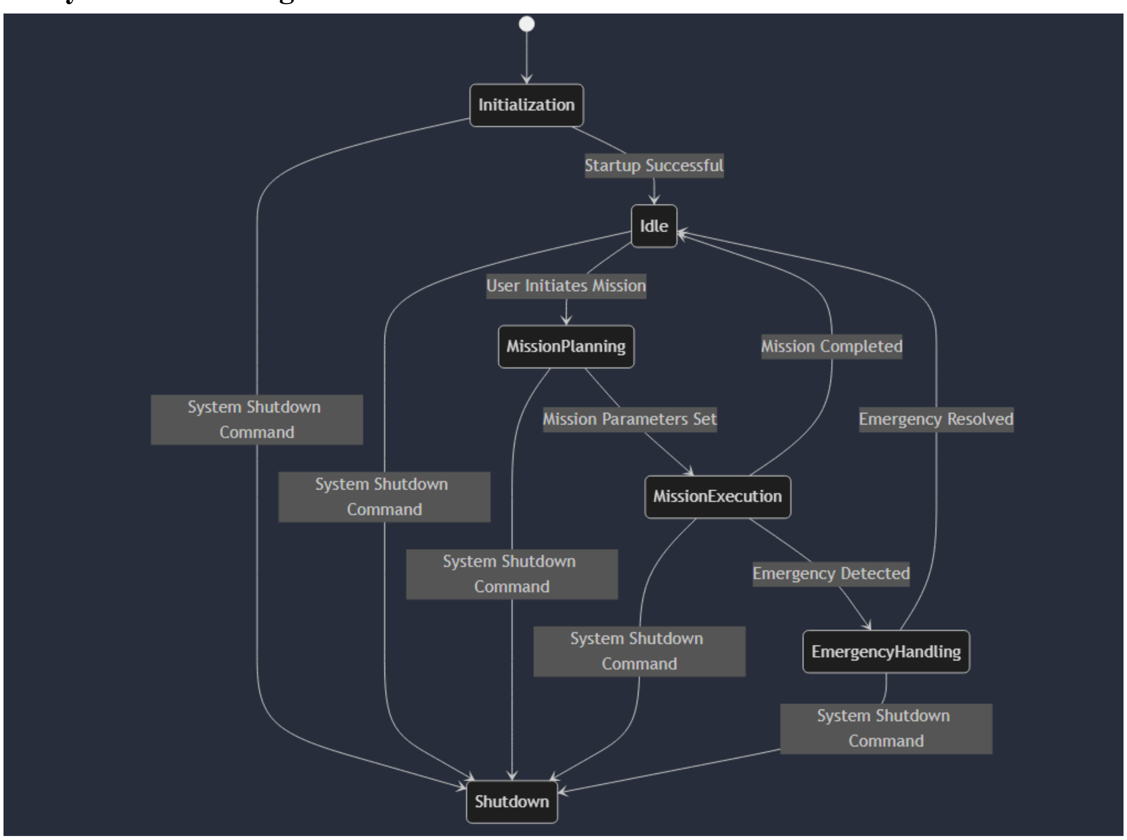
- Enables reliable and secure data exchange between UAVs and the User Interface.

3.6.2 Functionality

- Implements communication protocols for real-time synchronization of UAV actions and user commands.
- Monitors communication quality and ensures system connectivity in case of signal disruptions.

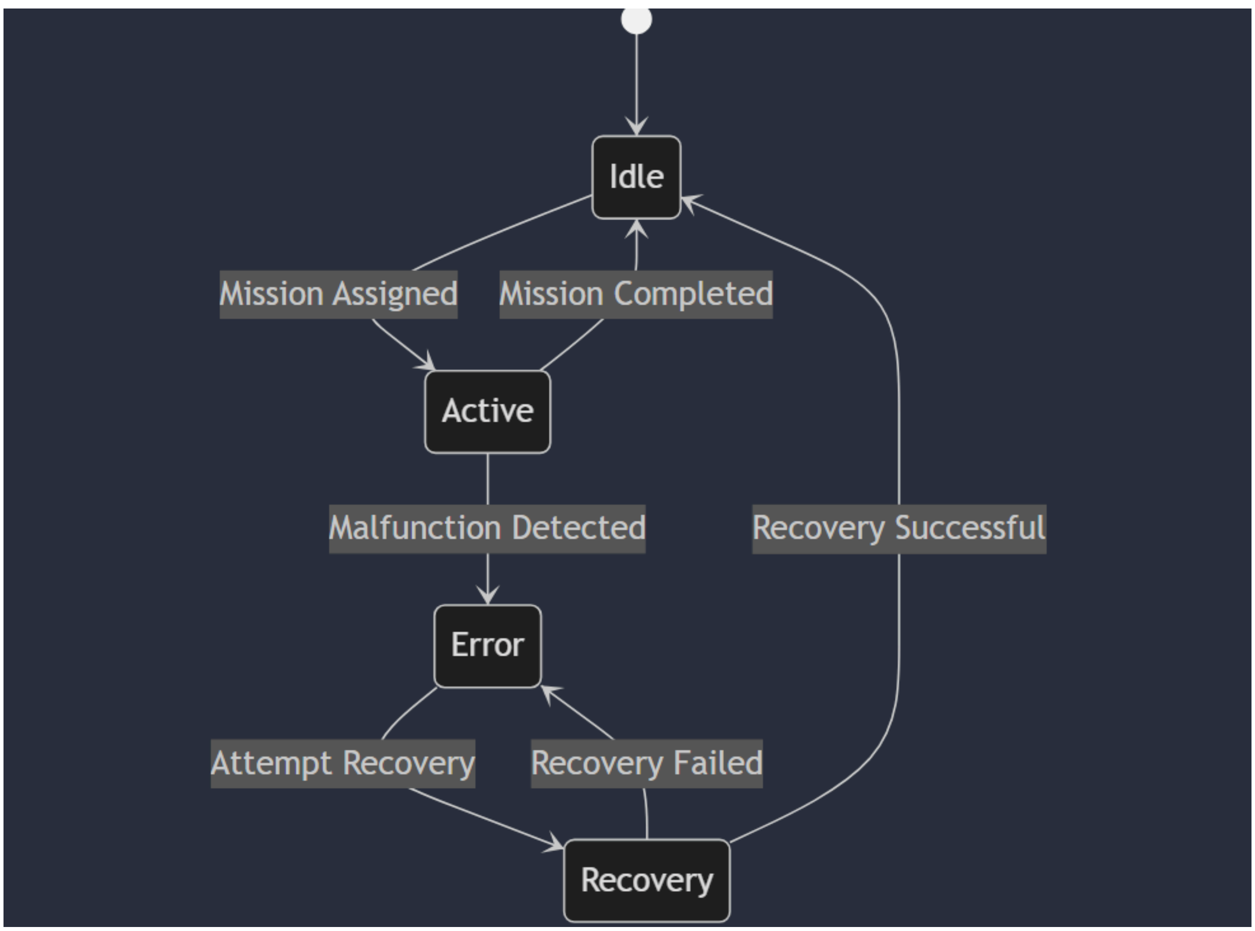
4. State and State Transitions

4.1 System State Diagram



- 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.

4.2 Subsystem State Diagram



- 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.

5. Use Cases

5.1 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.

5.2 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.