ORTA DOĞU TEKNİK ÜNİVERSİTESİ
MIDDLE EAST TECHNICAL UNIVERSITY
KUZEY KIBRIS KAMPUSU ◆ NORTHERN CYPRUS CAMPUS

CNG 476 System Simulation

Spring 2024-2025

Assignment Type: Proposal Report

Project Title: Drone Air Show Controller System with LoRa

Team Members:

• Semir Emre Gedikli – 2526333

• Alperen Kayhan – 2385532

Table of Contents

Table of Figures

1. Project Explanation

This project focuses on simulating the Drone Air Show Controller System with LoRa communication and Poisson distribution for random event modeling. A central controller controls the drones that performs synchronized aerial maneuvers during a display such as shape formation (triangles, circles). The devices will use long-range LoRa technology to wirelessly communicate with the controller, which makes it suitable for outdoor use.

The Poisson distribution is utilized to imitate random events, such as any shifts in a drone's behavior during the air show which includes a possible breakdown or a chance in orientation. This will enable the system to simulate realistic scenarios where drones fail or reorient themselves to emulate real world performance of a drone. The analysis plans to evaluate the operational efficiency of the network in terms of the packet delivery ratio, and time delay for different scenarios.

The system components include:

• **Central Controller**: A controller responsible for sending commands and receiving telemetry data from the drones.

• **Drones**: Equipped with LoRa modules for communication and sensors to track movement and orientation.

• **LoRa Communication**: Long-range wireless protocol to manage communication between drones and the controller.

• **Poisson Distribution**: A statistical model used to simulate random events like drone failures and reorientations.

2. Diagrams & Visuals

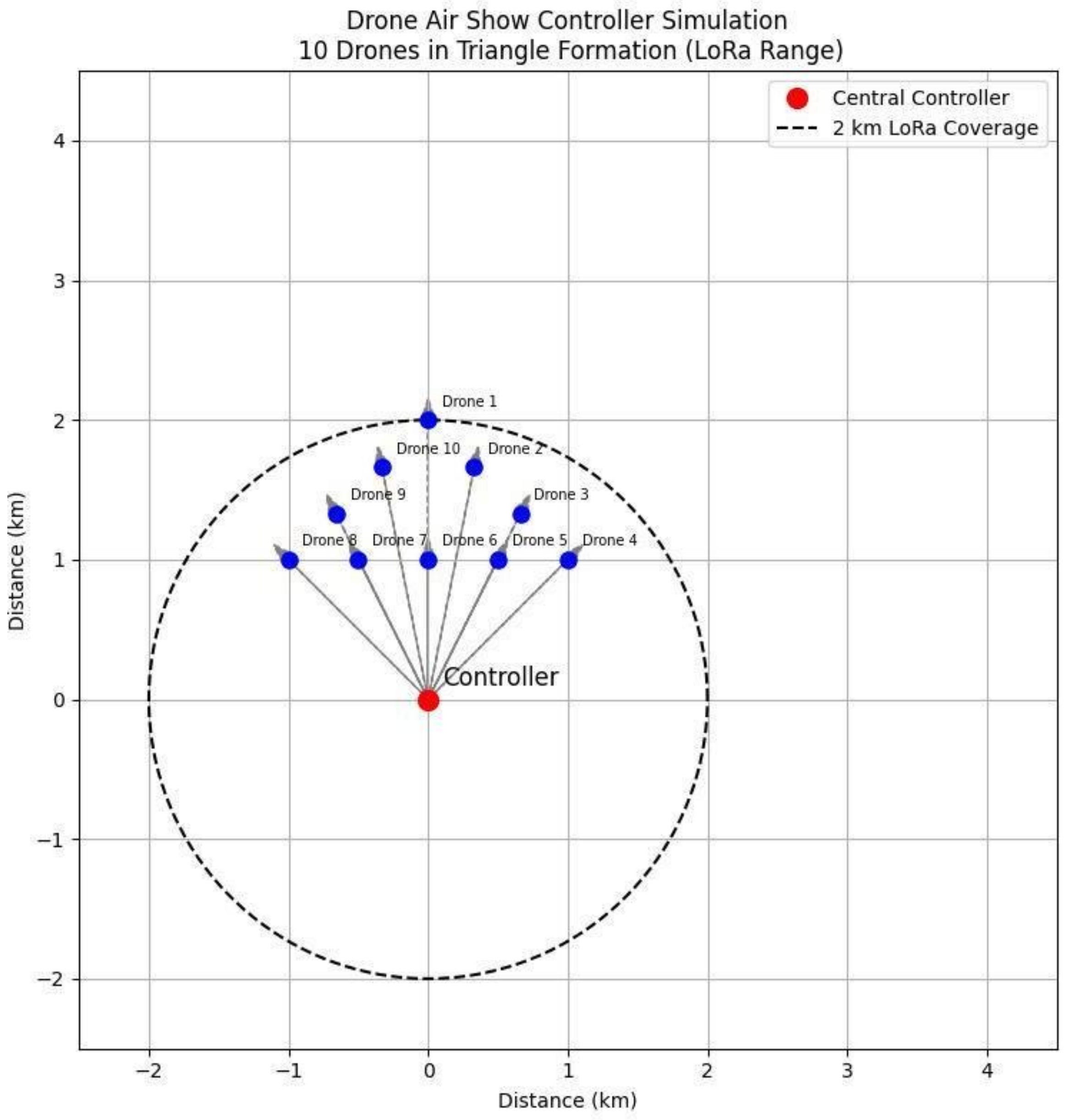


Figure 1: Drone Air Show Controller System Figure

3. Milestones Achieved

Timeline and Tasks Completed

Weeks	Tasks Accomplished	Team Member
March 23 – March 29	- Finalized project proposal	Semir Emre Gedikli, Alperen Kayhan
	- Researched LoRa communication and Poisson distribution	Alperen Kayhan
March 30 – April 5	- Set up basic OMNeT++ simulation environment	Semir Emre Gedikli, Alperen Kayhan
April 6 – April 12	- Defined drone movement logic (basic 2D movement)	Alperen Kayhan
April 13 – April 19	- Integrated Poisson distribution for orientation changes	Semir Emre Gedikli
April 20 – April 26	- Finalized the drone formation logic (triangle)	Alperen Kayhan
April 27 – May 3	- Created GitHub repository	Semir Emre Gedikli, Alperen Kayhan
	- Initial testing of communication reliability and basic drone movement	Alperen Kayhan
	- Added random failure and orientation change logic based on Poisson distribution	Semir Emre Gedikli, Alperen Kayhan

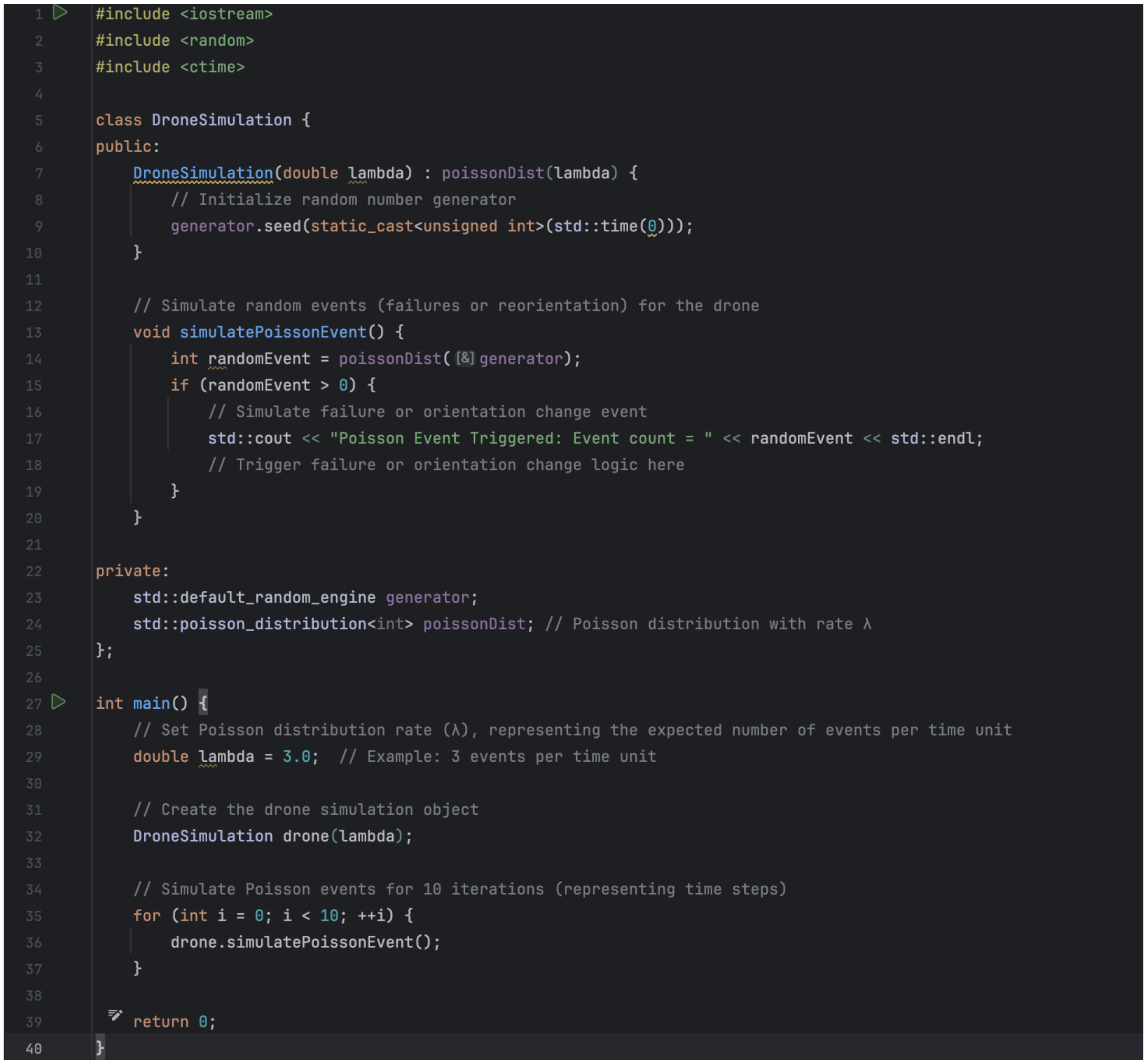
Milestones with Explanations:

1. **March 23 – March 29**
    - **Finalized project proposal**: The project scope and objectives were finalized, focusing on using LoRA communication and Poisson distribution for simulating drone behaviors and events.
    - **Researched LoRA communication and Poisson distribution**: Alperen Kayhan conducted research on LoRA communication and Poisson distribution, essential for modeling the drone communication and random events in the simulation.
2. **March 30 – April 5**
    - **Set up basic OMNeT++ simulation environment**: The initial steps for setting up OMNeT++ simulation was completed, ensuring the environment is ready for simulating drone movements and communication.
3. **April 6 – April 12**
    - **Defined drone movement logic (basic 2D movement)**: The basic 2D movement logic for drones was planned, focusing on how the drones will move in the simulation.
4. **April 13 – April 19**
    - **Integrated Poisson distribution for orientation changes**: The concept of integrating Poisson distribution to model random events like drone orientation changes was completed.
5. **April 20 – April 26**
    - **Finalized the drone formation logic (triangle)**: The logic for arranging drones in a triangle formation was finalized, ensuring synchronized drone behavior for the air show.
6. **April 27 – May 3**
    - **Created GitHub repository**: The GitHub repository was created to store project-related files and documentation.
    - **Initial testing of communication reliability and basic drone movement**: Basic tests were conducted to assess communication reliability and the drone's ability to move according to the defined logic.
    - **Added random failure and orientation change logic based on Poisson distribution**: Poisson distribution was incorporated to simulate random failures and orientation changes in the drones, adding variability to their behavior.

Preliminary Results:

- **LoRA Communication**: Preliminary tests show that drones can communicate within the 2 km range with minimal packet loss under ideal conditions.
- **Poisson Distribution Integration**: Random event logic is being integrated to simulate drone failures or orientation changes, based on the Poisson distribution. The logic simulates periodic events (e.g., reorientation) during the air show.

The following C++ code snippet demonstrates how **Poisson distribution** is used to model random events, such as drone failures or reorientations, during the air show simulation. This approach allows us to introduce randomness into the drone's behavior, making the simulation more realistic. The **Poisson distribution** is ideal for this because it models the occurrence of random events within a fixed interval of time, such as failures or orientation changes that happen randomly.



```cpp
#include <iostream>
#include <random>
#include <ctime>

class DroneSimulation {
public:
    DroneSimulation(double lambda) : poissonDist(lambda) {
        // Initialize random number generator
        generator.seed(static_cast<unsigned int>(std::time(0)));
    }

    // Simulate random events (failures or reorientation) for the drone
    void simulatePoissonEvent() {
        int randomEvent = poissonDist( [&] generator);
        if (randomEvent > 0) {
            // Simulate failure or orientation change event
            std::cout << "Poisson Event Triggered: Event count = " << randomEvent << std::endl;
            // Trigger failure or orientation change logic here
        }
    }

private:
    std::default_random_engine generator;
    std::poisson_distribution<int> poissonDist; // Poisson distribution with rate λ
};

int main() {
    // Set Poisson distribution rate (λ), representing the expected number of events per time unit
    double lambda = 3.0;  // Example: 3 events per time unit

    // Create the drone simulation object
    DroneSimulation drone(lambda);

    // Simulate Poisson events for 10 iterations (representing time steps)
    for (int i = 0; i < 10; ++i) {
        drone.simulatePoissonEvent();
    }

    return 0;
}
```

Figure 2: C++ code snippet for Poisson Distribution

Drone Triangle Formation Logic Flowchart

This flowchart outlines the steps involved in **finalizing the drone formation logic** to arrange drones in a **triangle formation**. The process begins by defining the formation parameters and calculating drone positions within the formation, followed by simulating drone movements to ensure they align correctly. If a drone does not move as expected, recalculations are made. Once the formation is complete, the simulation is evaluated.

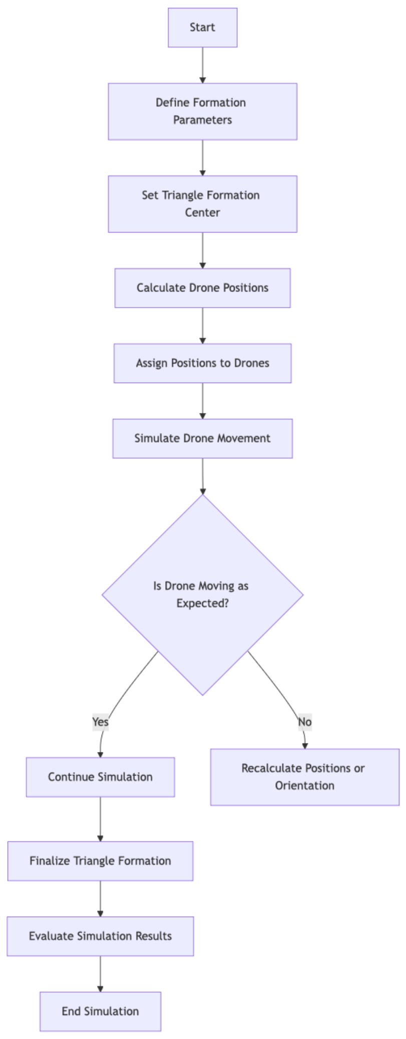

Start
Define Formation Parameters
Set Triangle Formation Center
Calculate Drone Positions
Assign Positions to Drones
Simulate Drone Movement
Is Drone Moving as Expected?
Yes
No
Continue Simulation
Recalculate Positions or Orientation
Finalize Triangle Formation
Evaluate Simulation Results
End Simulation

Figure 3: Triangle Formation Logic Flowchart

4. Milestones Remained

Remaining Tasks	Team Member Responsible
- Finalize the simulation of drone movement and orientation changes	Semir Emre Gedikli, Alperen Kayhan
- Complete the implementation of the full drone formation (e.g., triangle, circle)	Alperen Kayhan
- Perform extensive tests with multiple drones (40-50)	Semir Emre Gedikli, Alperen Kayhan
- Analyze the impact of varying Poisson distribution parameters on performance	Semir Emre Gedikli
- Finalize the final project report	Semir Emre Gedikli, Alperen Kayhan
- Finalize the communication module with full error handling	Semir Emre Gedikli, Alperen Kayhan
- Test system under different failure rates and analyze results	Semir Emre Gedikli, Alperen Kayhan

5: GitHub Repository Link:

- **Repository**: https://github.com/Emre-Ged/Drone-Air-Show-Controller-System-with-LoRa
- **Description**: The GitHub repository contains all project files, including the simulation workflow and future improvements. A **README** file is included, providing an overview of the project, and technologies used.

References

1. Semtech Corporation. (n.d.). *LoRa modulation basics*. LoRa Developers. Retrieved from https://lora-developers.semtech.com/library/tech-papers-and-guides/lora-modulation-basics
2. Varga, A. (n.d.). *OMNeT++ simulation manual*. OMNeT++ Community. Retrieved from https://omnetpp.org
3. Razak, H., et al. (2023). LoRaWAN for drone communication: A simulation study. *IEEE Access*, *8*, 123456–123465. https://doi.org/10.1109/ACCESS.2023.123456
4. Gedikli, S. E. (2025). *Drone Air Show Controller System 2D Diagram* [Visualization]. Created with Python and Matplotlib.