TEB1113
Algorithm and data structure
Research on Drone Swarm Report

No.	Name	ID No	Program
1	Mohammad Hafiz Ikram Bin Mohd Irwan	24000258	Bachelor Of Computer Science
2	Muhammad Thaifur Bin Ahmad Shahidan	24000641	Bachelor Of Computer Science
3	Muhammad Aiman Haikal bin Mohammad Akmal Surish	24000458	Bachelor Of Computer Science
4	Muhammad Amin Zufar Bin Ramlan	24000162	Bachelor Of Computer Science
5	Mohammad Amir Hazman bin Nawandi	24000387	Bachelor Of Computer Science

Introduction

Drone swarm technology has been implemented throughout various fields of work such as disaster relief, agriculture, in-game environments through unity as well as military uses.

Drone swarm technology has proved to be practical even in the real world such as disaster relief where the drones are used to assess how much damage was done in the area, identifying any potential survivors left by the disaster and even deliver supplies to those who cannot be evacuate to safer areas.

In agriculture, Drone swarms are useful mainly in farms that require large areas of land as the drones can alert the landowners about potential crop damages, drop seeds onto the farm as well as spraying chemicals such as fertilizer onto the farm which is much more efficient with drone swarms than manually executing each task.

The fundamentals behind a drone swarm are essentially multiple drones that are codependent on each other and rely on AI and various algorithms to replicate a ‘swarm’ by being able to navigate themselves without relying on human control, being able to maintain a position in a formation as well as avoiding obstacles to ensure the swarm formation is not disrupted. The development of drone swarm technology was to change from a centralized operation to an autonomous decentralized operation and rather than rely on human control and decisions, which are liable to human error, is instead left to algorithms and AI to adapt to certain environments and execute complicated tasks in a more efficient manner.

We aim to research the different ways we can replicate drone swarm technology as well as evaluate how well the implementations are able to perform. The two methods of algorithms we plan to experiment with are algorithms that are made from scratch and algorithms from existing libraries or packages.

Drone swarm unity simulation overview

To replicate the idea of a drone swarm in unity, we aim to recreate the characteristics of a drone swarm such as autonomous navigation which allows individual drones to act independently in the swarm while still maintaining the rules of the swarm. Collision detection will also need to be implemented so that the drones are able to communicate with other drones in the swarm to not collide into each other and retain their formation. The algorithms implemented must also ensure that the drones can be changed into various formations such as a grid or V-shape formation which may make executing certain tasks easier for the drones if certain formations are upheld.

Unity is a useful game engine to aid in creating a drone swarm because of features such as a built-in physics engine which helps in the simulation of collisions allowing us to model the behavior of the drones to a realistic simulation of how the drones would interact with the environment and certain situations.

Another reason to utilize Unity for our simulation are the different tools such as libraries and packages provided within unity which allows users and developers to implement algorithms for a drone swarm much easier as they can simply use already existing algorithms such as flocking, pathfinding as well as decentralized coordination.

Algorithm from scratch for drone swarm

One way of implementing a drone warm in unity is using algorithms which are made from scratch which refers to algorithms that are made completely by hand of the developer instead of relying on existing algorithms from libraries or packages. The developer needs to build the algorithm from the foundation and improve the algorithm step by step from the core logic of what the developer wants the algorithm to achieve.

There are multiple algorithmic ideas that a developer can use and build from scratch into their drone swarm, namely Boid's algorithm which is meant to imitate flocking behavior seen in a flock of birds or a school of fish which are behaviors we can copy to create similar behavior for a drone swarm. This algorithm follows three main concepts, separation, alignment and cohesion which each drone follows. Separation ensures there is a minimum distance for each drone in a swarm and avoids collisions in the swarm as each drone must maintain a safe distance between each other. For the drones to travel in the same direction and speed, alignment is used where a drone measures the direction and speed of surrounding drones and matches that velocity. This makes sure that each drone stays in a swarm and formation by travelling at the same pace. Cohesion makes sure the drones do not drift too far from the swarm by making them flock towards the center of the swarm whilst still maintaining a safe distance between other drones.

Another algorithm that can be built from scratch is the A* algorithm which is a pathfinding algorithm which identifies the best possible route to the destination which can be defined by shortest time taken or least distance travelled. This algorithm is practical for drone swarms because the drones are programmed to find the most optimal path without colliding into other obstacles. A real-life example would be during disaster relief where the drones may need to navigate through tight areas and with A* algorithm, the drones can locate safe areas to steer through. When this algorithm is implemented in a swarm it can be used to make sure that each individual drone path does not collide with another and thus reducing the chances of a collision happening.

An algorithm the developer may also want to consider when creating a drone swarm is formation control. The idea behind this algorithm is to control the drone swarm to be able to maintain certain positions such as grid or V-shape formation without colliding into each other. This algorithm would require the drones to adjust their velocity and position relative to the drones that surround it adapt to the environment so that the formation is not disrupted. Formation control can be a way for drones to carry out their tasks in a much more efficient manner as in cases of agriculture or disaster relief, the drones would be able to achieve larger coverage if they were in a set formation instead of individually covering their own areas as each area only needs to be maintained and observed by a single drone rather than multiple drones covering the same path.

Algorithm using libraries drone swarm

In drone swarm applications, reinforcement learning (RL) algorithms are frequently employed to address the challenges of swarm coordination, path planning, and real-time decision-making. These tasks can be highly complex due to the dynamic nature of the environments in which drone swarms operate. To simplify the implementation and management of these systems, developers often use libraries and frameworks that provide pre-built components for advanced RL algorithms. One commonly used library for implementing RL algorithms in drone swarm simulations is TensorFlow (or PyTorch), which offers tools for building and training neural networks in reinforcement learning models. In particular, these libraries support frameworks like Ray RLlib, which provide optimized implementations of algorithms such as Multi-Agent Deep Deterministic Policy Gradient (MADDPG) and Proximal Policy Optimization (PPO).

One widely used MARL algorithm is Multi-Agent Deep Deterministic Policy Gradient (MADDPG). MADDPG is a powerful reinforcement learning method used in multi-agent environments, and libraries like Ray RLlib or Unity's ML-Agents Toolkit can efficiently implement this algorithm. MADDPG extends the Deep Deterministic Policy Gradient (DDPG) algorithm, a model-free, off-policy reinforcement learning method designed for environments with continuous action spaces. In drone swarms, MADDPG is particularly valuable because it allows for continuous control over drone movements and decision-making in real-time environments. By leveraging libraries such as Ray RLlib, each drone can develop a distinct policy that optimizes its actions based on both its individual observations and shared information from other agents, leading to effective swarm coordination. For instance, a MADDPG implementation can control a fleet of drones to collectively track and follow moving targets, avoid collisions, and dynamically adjust flight patterns using the computational resources provided by libraries.

Similarly, Proximal Policy Optimization (PPO) is another RL algorithm used in drone swarm applications, and libraries like Stable Baselines3 provide a highly efficient implementation of PPO. PPO is a popular reinforcement learning algorithm that strikes a balance between performance and stability, making it ideal for tasks that involve drone swarm coordination. Using libraries such as ML-Agents Toolkit in Unity, PPO can be applied to allow drones to optimize their behavior incrementally, avoiding drastic policy changes that could destabilize the learning process. This makes PPO particularly useful in tasks like environmental monitoring or search and rescue, where drones need to divide a large area efficiently and maintain a balanced coverage strategy. These libraries help ensure smooth and stable policy refinement, enabling consistent swarm performance even as external conditions change.

Runtime performance:

The runtime performance of drone swarms is influenced by various factors, including coordination, communication delays, and the complexity of path planning algorithms. Drone swarms typically rely on algorithms and local sensors to coordinate multiple drones autonomously, minimizing human intervention. These swarms are often more efficient than single drones because they can perform tasks in parallel, which is particularly useful in applications such as firefighting, agriculture, and surveillance.

However, achieving optimal runtime performance can be challenging. Communication delays and inaccuracies in GPS positioning are significant concerns. For example, errors in communication between drones or delays in receiving positional data can affect the coordination and collision avoidance capabilities of the swarm.

Path planning and obstacle avoidance algorithms also impact performance. Algorithms such as particle swarm optimization, ant colony optimization, and differential evolution are used to compute efficient paths for drones while avoiding obstacles in real-time. The complexity of these algorithms can sometimes slow down real-time decision-making, especially in environments where drones must continuously adapt to new obstacles or changes in terrain.

In simulation environments, using Software-In-The-Loop (SITL) can accelerate development by allowing for the testing of swarm behavior without real-world risks. Simulations help assess operational efficiency and can highlight issues like communication delays and collision risks, which ultimately affect the runtime performance of real-world drone swarms.

Conclusion

Drone swarm technology has demonstrated its potential across various fields, including disaster relief, agriculture, and military applications. Through the autonomous coordination of multiple drones, swarms can efficiently perform complex tasks like damage assessment, crop monitoring, and search and rescue operations. The development of drone swarms is driven by the goal of decentralized control, where artificial intelligence (AI) and algorithms handle dynamic environments, reducing human error and increasing operational efficiency.

In simulations, particularly using platforms like Unity, drone swarms can replicate real-world behaviors, enabling the testing of formations and collision avoidance. Developers have the option to implement custom algorithms from scratch or leverage pre-built libraries for reinforcement learning, such as TensorFlow and PyTorch. These libraries simplify the integration of advanced algorithms like MADDPG and PPO, enhancing the performance and decision-making capabilities of drone swarms in real-time environments.

Despite the promising advancements, challenges such as communication delays, GPS inaccuracies, and algorithm complexity can affect the runtime performance of drone swarms. Simulations offer a valuable tool for addressing these issues before deployment in real-world scenarios. As drone swarm technology continues to evolve, it holds the potential to revolutionize a range of industries through increased automation, efficiency, and adaptability.