IITISOC 2025
PROJECT PROPOSAL
Coordinated Dual-Drone Framework for Coverage Based Irrigation Tasks

Team Details:

Team Leader:
Name: Burra Venkata Chakrapani
Roll Number: 240021005
GitHub: chakri2007
Discord: chakrapani2006
LinkedIn: `https://www.linkedin.com/in/chakrapani-burra-086878352`
Skills: ros2, px4, python

Team Member 2:
Name: Bhav Makhija
Roll Number: 240021004
GitHub: Bhavdex
Discord: Bhavj
LinkedIn: `https://www.linkedin.com/in/bhav-makhija`
Skills: Algorithm Development

Team Member 3:
Name: Sathvik
Roll Number: 240021001
GitHub: Sathvik557
Discord: Sathvik0586
LinkedIn: `https://www.linkedin.com/in/amadala-sathvik-333b7a323`
Skills: px4, CAD

Team Member 4:
Name: Srujana Patil

Roll Number: 240021018
GitHub: Sruj-08
Discord: Sruj08
LinkedIn: https://www.linkedin.com/in/srujana-patil-620677324
Skills: python

Team Member 5:
Name: Sana Tejasri Lakshmi
Roll Number: 240041033
GitHub: Tejasri676
Discord: Sana676
LinkedIn: https://www.linkedin.com/in/tejasri-lakshmi-sana-b6951b321
Skills: python

Domain: Intelligent Vehicles and Robotics
PS Name: Coordinated Dual-Drone Framework for Coverage Based Irrigation Tasks
PS Number: 04
Preference Number: 01

Project Solution:

In this proposal, we've explained how our team plans to approach the project step by step. We've included how we'll handle geotagging the area, how the drones will work together, communicate, manage things like battery levels and recharging. We've also described how the whole system will run as a coordinated swarm to get the job done as quickly and efficiently as possible. Overall, this proposal reflects our team's main goal to design a smart and reliable drone system that can handle both coverage and irrigation tasks smoothly.

To minimize energy consumption, the surveillance drone is initially deployed with an empty water tank, as its primary function is limited to coverage and monitoring, not irrigation. However, based on real-time irrigation requirements, the drone may subsequently adapt its role by filling its tank to perform irrigation tasks when necessary.

The drone speeds are selected based on optimized operational values as follows:

Drone Condition	Typical Speed (m/s)	Flight Time	Recharge Time
With Water	4	15 min	25 min
Without Water	8	30 min	25 min

Drones are deployed from the base station but begin operations from the farthest point of the area. This inward approach reduces energy usage during heavy irrigation tasks and improves overall efficiency.

Launch and Path Planning:

QGroundControl (QGC) is used to define the operational boundaries of the field through geofencing, ensuring the drones operate safely within the designated area. Within this geofence, waypoints are created to define the lawnmower path for surveillance and to ensure full coverage of the field. This waypoint data is exported as a KML file with each waypoint having a boolean variable `IsVisited`.

The takeoff point for both drones is selected to be along its edge, based on real-time convenience. This point is strategically chosen to be approximately equidistant from all areas of the field, ensuring balanced accessibility. This helps in cutting down on unnecessary travel time.

Initially, the Surveillance Drone starts the mission with an empty tank and generates the geotags using the predefined logic. As soon as the first geotag is generated, the irrigation drone is deployed.

Both the irrigation and surveillance drones are deployed from the base station and begin their operations from the farthest point of the designated area. As the drones progress inward towards the base station, the irrigation tasks are performed by both the drones when they come closer to the base, as the total surveillance will be completed. Since the drones are carrying water during irrigation, their energy consumption increases; however, as they are near the base station, the return distance is reduced, requiring less energy for the return journey. This results in a more efficient use of energy and minimizes the overall task execution time.

We chose the traditional lawn mower (back-and-forth) path because it's simple, easy for drones to follow, and doesn't involve too many turns. Even though some advanced methods like Hilbert curves might look efficient, they have a lot of turns which can slow the drone down and waste energy—especially when it's carrying water. Our method covers the area well while saving time and battery.

Logic for Geotagging and Severity Level:

The autonomous geotagging logic calculates latitude and longitude increments derived from the desired inter-point distance (25 meters) using the approximate conversion factors. Starting from a known boundary coordinate, a grid of GPS waypoints is generated by iteratively adding these increments, forming a coordinate matrix that the drone can follow autonomously for coverage tasks.

Each geotag is randomly associated with a severity score (1 to 5) that quantifies the intensity of irrigation required at that location. This is stored alongside the GPS coordinates in the geotag dataset. During mission planning, the drone prioritizes geotags with higher severity levels to optimize resource allocation and irrigation efficiency.

If a geotag has a severity level of 1, the drone will skip irrigation at that spot, as it's not needed. But if the severity level is between 2 and 5, the drone will perform irrigation, and

the amount of water irrigated depends on the severity level.

The generated geotags, along with their assigned severity scores, will be stored in the base computer as a list for task planning and execution.

Battery Recharging Logic:

In this project, we are developing a system where two autonomous drones work collaboratively to perform both field surveillance and targeted irrigation. The system is designed to be adaptive and energy-aware, allowing the drones to dynamically switch roles between surveillance and irrigation based on their real-time status.

Each drone independently monitors its battery level and task load, ensuring that at least one drone is always operational. This enables continuous mission progress even during extended tasks that involve multiple recharge cycles.

Battery Thresholds for Return-to-Base:

The drone continuously evaluates its remaining energy and initiates a return to base when it determines that the available energy is just sufficient to ensure a safe return to the base station. This evaluation is performed using the x' method, which assesses the energy threshold required for a successful return.

The X' Method For Battery Optimization:

- X = The maximum distance a drone can travel on a full battery.

- Y = The current distance from the drone's location to the charging station.

- $X' = X - Y$ = remaining safe operational range before the drone must initiate return.

The drone continuously recalculates X' as it moves. If the distance to the next waypoint or geotag exceeds X', the drone returns to base instead of proceeding. This real-time adjustment ensures the drone never runs out of battery during operation and can always return safely.

This method prevents drones from getting stranded mid-air, adjusts based on the drone's current location, avoids unnecessary early returns unlike fixed thresholds, and continuously adapts as the drone moves.

This method prevents drones from getting stranded mid-air, adjusts based on the drone's current location, avoids unnecessary early returns unlike fixed thresholds, and continuously adapts as the drone moves.

Swarm Coordination:

Swarm coordination helps both drones work together by sharing their status like location, battery, water level, and task progress. This way, they can decide who does what in real time, avoid overlapping, and handle tasks more smoothly. It saves time by avoiding repeated coverage, balancing the workload based on how close the drones are, how urgent the task

is, and their current status. If one drone runs low on battery or water, the other can take over, which also adds backup in case of any issue. Overall, it makes the whole system faster, smarter, and more flexible.

This planning makes Dual Drone Swarm Coordination (both drones perform both coverage and irrigation) a better one than assigning separate operations to each drone.

Once Drone A reaches its first recharge cycle, it also gets refilled with water, enabling both drones to alternate roles dynamically in the next cycle—ensuring load-balancing and continuous progress. Once the surveillance is completed and all the areas in need of irrigation are tagged properly, the Surveillance Drone (Drone A) starts irrigating from the opposite end of the field.

Each drone continuously checks and updates its:

- Current task - surveying, irrigating, returning to base

- Battery level

- Water level

- Position (GPS coordinates)

Visited waypoints and irrigated geotag positions are marked accordingly and updated in the list stored at the base station. This helps track which areas have been irrigated and which still require surveillance.

We will build a smart swarm system where drones can change roles when needed. For example:

- The surveillance drone will switch to irrigation after finishing its monitoring or when it's time to recharge and the irrigation drone needs help.

- The irrigation drone will switch to surveillance if its water is low but it still has enough battery to keep working.

Other times they can swap roles are when one drone's battery is low and the other has more power, or when one drone is closer to a task area, so they can work more efficiently together. This way, they help each other and get the job done faster.

Collision Avoidance:

We plan to operate the surveillance and irrigation drones at different altitudes to prevent collisions between them.

In the situation where both are at the same altitudes, collision avoidance helps drones fly safely by detecting and avoiding obstacles in real time, even in tricky environments. Our system uses a MAVLink-based module on a companion computer that works with PX4's built-in obstacle avoidance and a depth camera. It processes depth data to spot obstacles and adjusts the drone's path during AUTO missions. By sending updated commands to

the PX4 controller, the drone can smoothly reroute around obstacles without stopping its mission.

Time Efficiency:

Starting from the far end of the base station helps drones finish tasks closer to the base, saving time and battery especially when carrying water.

Precise mission planning, autonomous geotagging with severity scoring, allowing the drones to prioritize high-need areas, avoiding time wasted on over-irrigation or redundant scanning.

The described dual-drone system achieves high time efficiency by enabling both drones to perform coverage and irrigation tasks simultaneously, supported by a well-structured swarm coordination mechanism. Instead of waiting for one drone to complete surveying before another begins irrigation, the drones operate in parallel. Additionally, dynamic role switching ensures that when one drone goes for recharging or refilling, the other continues operating, maintaining uninterrupted mission flow.

The X' battery optimization method ensures drones only engage in tasks they can complete with enough energy to return safely, avoiding mid-task failures and unnecessary early returns.

Altogether, these features create a seamless, adaptive workflow that cuts down mission duration while increasing precision, making the entire framework significantly more time-effective and operationally streamlined.

Further Improvements:

To improve our project further, we plan to run simulations in MATLAB to test and find the most efficient way for our system to work. We're putting effort into planning and testing so we can make the system smarter, faster, and more reliable before actual implementation.

Initial Setup

Github repo link:

`https://github.com/chakri2007/IITI_SOC.git`

This contains the screenshots of the simulation for inital setup validation.

PROPOSAL SCHEDULE

Timeline:

Duration: June 3–August 2, 2025 (8 weeks).

Week 1: Mission Planning

- Use QGroundControl (QGC) to plan the mission and extract the KML file of the field layout.

- Generate waypoints using a lawn mower path pattern.

- Create Validated JSON datasets for both waypoints and geotags.

- Use a dataset update mechanism to track real-time mission progress.

Week 2: Surveillance Drone Implementation

- Configure the surveillance drone to follow pre-defined waypoints.

- Implement logic to transmit these waypoints and perform geotagging during surveillance.

- Ensure each geotag includes location data and severity scores.

Week 3: Irrigation Drone Development

- Program the irrigation drone to use the geotag data received from the surveillance drone to perform targeted irrigation.

- Include a mechanism to check off each geotag as "irrigated" or "pending," ensuring systematic coverage.

Week 4: Battery Monitoring & Recharge Logic

- Develop a real-time battery monitoring system for both drones.

- Implement autonomous Return-to-Base (RTB) functionality when battery levels drop below safe thresholds.

- Include state saving to resume tasks after recharging.

Week 5: Swarm Coordination & Work Sharing

- Enable coordination between the two drones.

- Implement workload division logic based on proximity to geotags or drone status (battery, irrigation tank).

- Ensure efficient parallel operation.

Week 6: Collision Avoidance and Improvements

- Set up obstacle avoidance using PX4.

- Conduct MATLAB simulations to test and change the algorithms to ensure the most time efficient strategy.

Week 7: Project Automation & GitHub Setup

- Create ROS launch files/automation scripts for deployment.

- Organize GitHub repository with structured folders and files.

- Write comprehensive README documentation.

Week 8: Report Finalization & Optimization

- Fine-tune code and optimize repository.

- Finalize project report with work summary, challenges, and future improvements in the report.