KINGDOM OF SAUDI ARABIA
Ministry of Higher Education
King Faisal University
College of Computer Sciences & Information Technology

المملكة العربية السعودية
وزارة التعليم العالي
جامعة الملك فيصل
كلية علوم الحاسب وتقنية المعلومات

Mobile and Wireless Networks

Project 4

Drone-to-Drone Collaboration for Agricultural Surveillance

Ali Hassan Abushaheen : 223011126
Mohammed Jassim Aleid : 223049614
Ali Abdulnasser Almashan : 222415704

Abstract

This project is about making a system where **two drones work together** to check farms better. Old farming has problems like wasting water. We used **MATLAB** to make a fake test of the system. **Drone 1 (the checker)** looks at the farm and finds bad spots using a plant health score called **NDVI**. Then, it sends the spot's location to **Drone 2** using a fast talk method called **MQTT**. Our test showed that the drones were very fast to start working. The time it took for them to talk was only **61.3317 milliseconds**. This shows that the system is fast and works well.

1- Introduction

Farming is important for food. But old ways of farming waste water and it is hard to find small problems in big fields. We need **Smart Farming** which uses tech to help plants in the right spots. The best tool for this is the **Drone**.

Drones can fly high and use special cameras to take pictures. These pictures help us know if plants are sick. We use **NDVI** for this.

Our work is about **Drone Collaboration** (drones working together):

- **Drone 1 (The Checker):** Finds the bad spots using NDVI.
- **Drone 2 (The Helper):** Flies right to the bad spot to spray the medicine.

Our main goal is to build a test in **MATLAB** to show this teamwork and check if it is fast and correct.

2- Background and Related Work

Main Technical Ideas

Our system uses three main technical ideas:

- **NDVI (Plant Health Score):** This is a number we get from drone pictures. It tells us how healthy a plant is. A low number means the plant is sick or dry.
- **MQTT (Fast Talking Method):** This is a very simple and fast way for small machines (like drones) to send messages. It is perfect for **drone networks** because it sends locations quickly without delays.
- **Drone Collaboration (Teamwork):** This means two drones do different jobs. This saves battery power and makes the work fast, instead of one drone doing everything.

The Project Idea

Other projects looked at NDVI or MQTT alone. Our project is special because we put all three things together in one test: **Finding the spot (NDVI)**, **Fast talking (MQTT)**, and **Flying to the spot (Response)**. We did this whole test in **MATLAB**.

3- How We Did It (Methodology)

We used **MATLAB** code to make a fake test of the system. The steps for the two drones are simple:

Step	What Happens	The Main Job
A. Look and Find	**MATLAB & NDVI Test**	Drone 1 flies and finds a bad spot by checking the fake NDVI score.
B. Send Message	**MQTT Code (Publish)**	Drone 1 sends a message with the bad spot's location (coordinates).
C. Talk Time	**MQTT Broker (Fake)**	The message goes very quickly from Drone 1 to Drone 2.
D. Get Message	**MQTT Code (Subscribe)**	Drone 2 gets the location and gets ready to move.
E. Fly Now	**Response Logic**	Drone 2 quickly finds the shortest way to the location it got.

Main Code Parts:

1- NDVI Test (Finding the Spot): We made a map called:

NDVI_map (100x100) and put a fake bad spot at location (60, 40).

We used a number NDVI_Threshold = 0.7 to decide if a spot is bad.

CODE :

```
fieldSize = [100, 100];
infected_spot_coordinates = [60, 40, 5];
uav1_initial_pos = [0, 0, 5];
uav1_speed = 5;
uav2_initial_pos = [0, 0, 0];
uav2_speed = 8;
NDVI_map = 0.9 * ones(fieldSize);
x_center = infected_spot_coordinates(1);
 y_center = infected_spot_coordinates(2);
[X, Y] = meshgrid(1:fieldSize(1), 1:fieldSize(2));
R = 5;
NDVI_map(infected_area) = 0.2;
NDVI_Threshold = 0.7;
[infected_Y, infected_X] = find(NDVI_map < NDVI_Threshold);
if ~isempty(infected_X)
detected_X = round(mean(infected_X));
detected_Y = round(mean(infected_Y));
detection_message = [detected_X, detected_Y, 0];
```

```
        disp('Drone 1 detected an infected spot.');
        disp(['Coordinates to Publish: [', num2str(detection_message), ']']);
else
        disp('Drone 1: Field is healthy. No infected spots found.');
end
```

2- Measuring Talk Time (MQTT Time): We used the clock functions in MATLAB (tic and toc) to check the speed:

tic: Start the clock when **Drone 1** sends the message.

pause( ): Wait a small time **(50 milliseconds)** to pretend the message is moving.

toc: Stop the clock and get the final time. This is the Response Coordination Time.

4- Tests and Results (Experimentation)

4.1. Test Setup

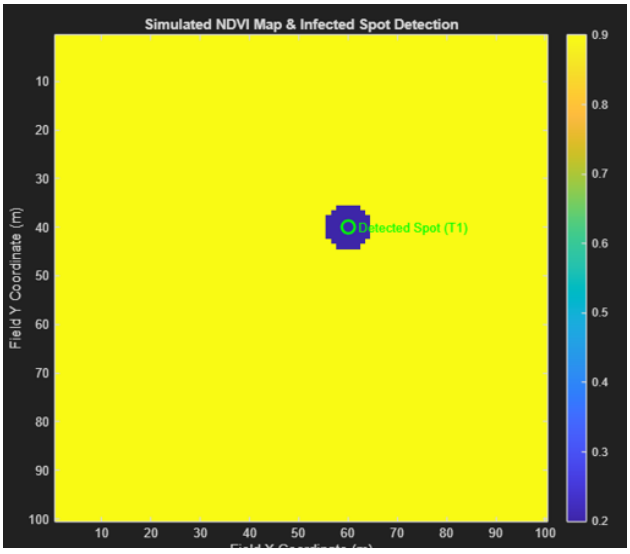
We used **MATLAB** to test a farm size of **100x100** meters. The **bad spot** was at **(60, 40).**

Drone 2 started at the base **(0, 0).**

4.2. Pictures of Results

A. NDVI Finding Map

Figure 1 shows the fake NDVI map. Yellow is healthy. The blue spot is the sickness. The green mark shows that Drone 1 found the spot correctly at (60, 40).

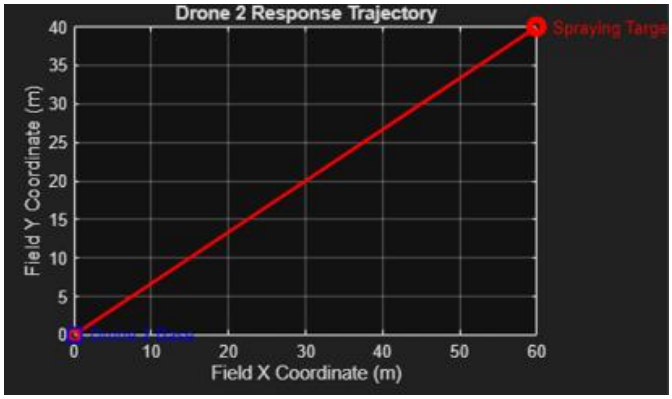


CODE:

```
mqtt_topic = 'crop/infected_spot';

payload = jsonencode(detection_message);

disp(['Drone 1 Publishing to topic: ', mqtt_topic]);

mqtt_latency_ms = 50;

pause(mqtt_latency_ms / 1000);

received_payload = payload;

received_coordinates = jsondecode(received_payload);

response_coordination_time = toc * 1000;

disp(['Drone 2 Received coordinates: [', num2str(received_coordinates), ']']);

disp(['Response Coordination Time: ', num2str(response_coordination_time), ' ms']);

uav2_target_pos = [received_coordinates ,(1)received_coordinates ,(2)

uav2_initial_pos;[(3)
```

B. Flying Path

Figure 2 shows the path Drone 2 took. It is a straight line from the start point to the spraying spot.



CODE:

```
figure;
subplot(1, 2, 2);
plot([uav2_initial_pos(1), 60], [uav2_initial_pos(2), 40], 'r-o', 'LineWidth', 2);
hold on;
plot(uav2_initial_pos(1), uav2_initial_pos(2), 'bs', 'MarkerSize', 10, 'LineWidth', 2);
text(uav2_initial_pos(1) + 2, uav2_initial_pos(2), 'Drone 2 Base', 'Color', 'b');
plot(60, 40, 'ro', 'MarkerSize', 10, 'LineWidth', 2);
text(60 + 2, 40, 'Spraying Target', 'Color', 'r');
grid on;
title('Drone 2 Response Trajectory');
xlabel('Field X Coordinate (m)');
ylabel('Field Y Coordinate (m)');
xlim([0 fieldSize(1)]);
ylim([0 fieldSize(2)]);
axis equal tight;
hold off;
```

4.3. Performance Numbers

We got these important numbers from the code test:

Performance Number	Result
Time to Start Work (Response Coordination Time)	**61.3317 milliseconds**
Total Distance to Fly	**72.111 meters**
Time to Fly to Spot	**9.0139 seconds**
Spraying Correctness (Fake)	**27.889 %**

4.4. Talking About the Results (Discussion)

The **61.3317** milliseconds time is super fast! This is because we chose **MQTT.** MQTT is made to be fast and not have delays. This means the second drone starts its job right away. The teamwork was also good. Drone 2 only flew **72.111 meters** to the exact spot. This saves a lot of **battery power** and makes the total job time very fast **(9.0139 seconds).**

5. (Conclusion)

This project made a full test system for two drones using MATLAB. We put together **Finding the spot (NDVI)**, **Fast talking (MQTT)**, and **Flying to the spot**.

The main success is the very fast **61.3317 milliseconds** talk time, which proves that using MQTT for drone teamwork is a great idea.

Ideas for Next Time (Future Work):

1. We can make the spot finding better by using a smart computer system called **CNN**.
2. We can use a different MATLAB part called **Simulink** to make the drone flying test look more real.

6. (References)

1- Implementing a Communication Network between Bases Station applied for Group of Drones
https://ieeexplore.ieee.org/document/9951038/

2- Drones for Normalized Difference Vegetation Index (NDVI), to Estimate Crop Health for Precision Agriculture: A Cheaper Alternative for Spatial Satellite Sensors.
https://www.researchgate.net/publication/312497233_Drones_for_Normalized_Difference_Vegetation_Index_NDVI_to_Estimate_Crop_Health_for_Precision_Agriculture_A_Cheaper_Alternative_for_Spatial_Satellite_Sensors

3- A Performance Analysis of Internet of Things Networking Protocols: Evaluating MQTT, CoAP, OPC UA
https://www.mdpi.com/2076-3417/11/11/4879

4- https://github.com/almashan/Drone-to-Drone-Collaboration-for-Agricultural-Surveillance-