Project Title: Privacy Preserving Drone Mobility

Introduction:

The provided code implements a drone mobility model simulation with two distinct models. Model 0 represents the basic movement of the drone along straight lines with a specified velocity and pausing time. Model 1 involves the drone moving between randomly generated rectangles, and for each rectangle, it identifies the minimum and maximum distances to randomly placed points within that rectangle. The simulation aims to visualize and analyse the drone's movement patterns and the interactions with the randomly generated rectangles.

Parameters:

mobility_model_parameter: An integer value (0 or 1) indicating the mobility model to be used.

velocity_parameter: The speed of the drone while moving in Model 0.

pausing_time_parameter: The time interval during which the drone pauses between movements in Model 0.

k_parameter: An integer representing the number of random points to be generated in Model 1.

number_of_lines_parameter: The number of lines to be simulated in both mobility models.

Simulation Execution:

1. Initialization:

- The code defines five input parameters, namely 'mobility_model_parameter', 'velocity_parameter', 'pausing_time_parameter', 'k_parameter', and 'number_of_lines_parameter', to control the simulation.

- 'network_size' is set to 1000 to define the boundaries of the drone's mobility area.

2. Line and Point Generation:

- The code generates 'num_lines' lines, each consisting of two points, where the second point is randomly placed within the network size.

- The starting point of the drone is initialized at coordinates (0, 0).

3. Mobility Model 0 (Straight Line Movement):

- If 'mobility_model_parameter' is set to 0, the code enters Mobility Model 0.

- Each line is plotted on the graph, representing the drone's trajectory between the two points.

- The drone's starting point is marked with a red asterisk on the graph.

- For each line, the distance between the two points is calculated.

- The time needed to travel between the points is computed based on the 'velocity_parameter'.

- The movement is divided into multiple steps, allowing the drone to be visually simulated moving between the points.

- A pause of duration 'pausing_time_parameter' occurs between each movement to create a time gap.

- The total distance covered by the drone and the total flying time for all lines are calculated and displayed in the MATLAB command window.

4. Mobility Model 1 (Random Rectangle Movement):

- If 'mobility_model_parameter' is set to 1, the code enters Mobility Model 1.

- Each line represents a rectangle, and its sides are plotted on the graph.

- Random points are generated within each rectangle and plotted in green.

- The distances between the starting point of the drone and each randomly generated point are calculated and stored.

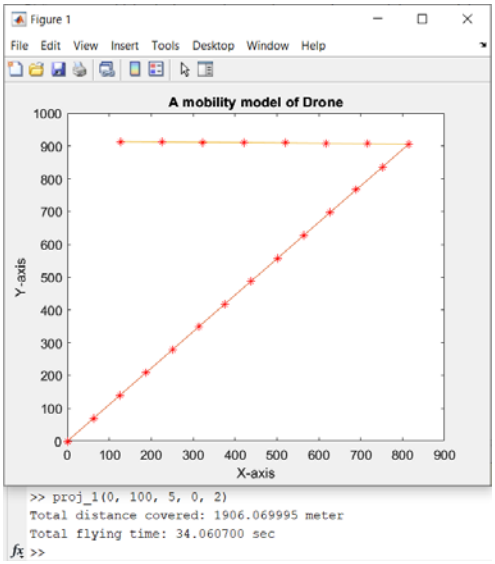
- The minimum and maximum distances are identified, and lines connecting the drone to these points are plotted in blue, representing the minimum and maximum distances.

- For each rectangle, the drone's movement towards the minimum and maximum distance points is simulated using similar steps as in Mobility Model 0.

- The total distance covered by the drone and the total flying time for all movements are calculated and displayed in the MATLAB command window.

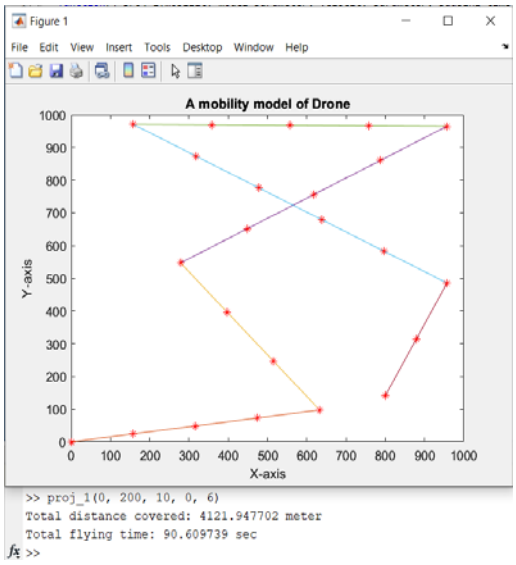
Snapshot of result graphs:

i) **Proj_1(0, 100, 5, 0, 2)** % RWP, 100 m/s, 5 secs pausing time, zero dummy location, 2 lines



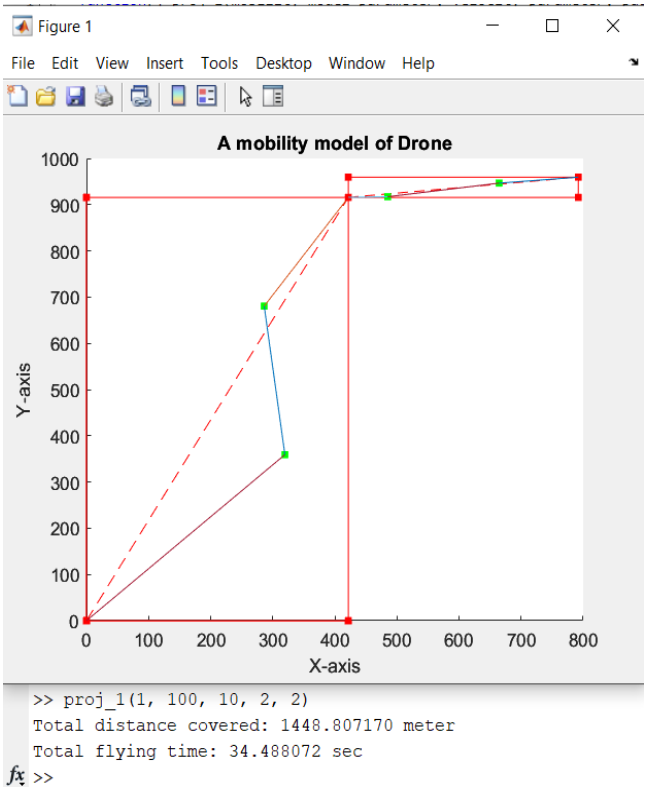
(image_1)

ii) **Proj_1(0, 200, 10, 0, 6)** % RWP, 200 m/s, 10 secs pausing time, zero dummy location, 6 lines



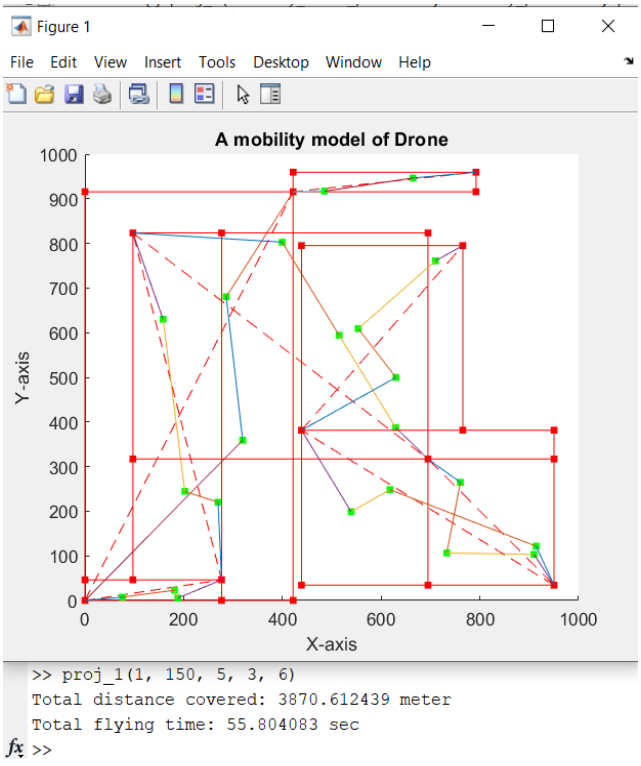
(image_2)

iii) **Proj_1(1, 100, 10, 2, 2)** % PPR, 100 m/s, 10 secs pausing time, two dummy locations, 2 lines



(image_3)

iv) **Proj_1(1, 150, 5, 3, 6)** % PPR, 150 m/s, 5 secs pausing time, three dummy locations, 6 lines



(image_4)

Note: For RWP, the drone pauses for the given pausing time upon reaching the random destination.

Conclusion:

The implemented drone mobility model simulation successfully visualizes the movement of the drone using two different models. Model 0 shows basic straight-line movement, while Model 1 demonstrates interactions with randomly generated rectangles and their points. By analysing the total distance covered and the total flying time, the simulation provides insights into the drone's behaviour in each mobility model. Users can experiment with different parameter values to gain further understanding of the drone's mobility patterns and make comparisons between the two models. The simulation contributes to the understanding of drone mobility and assists in exploring potential applications in various scenarios.