CS5331: Aerial Computing
Summer II 2023

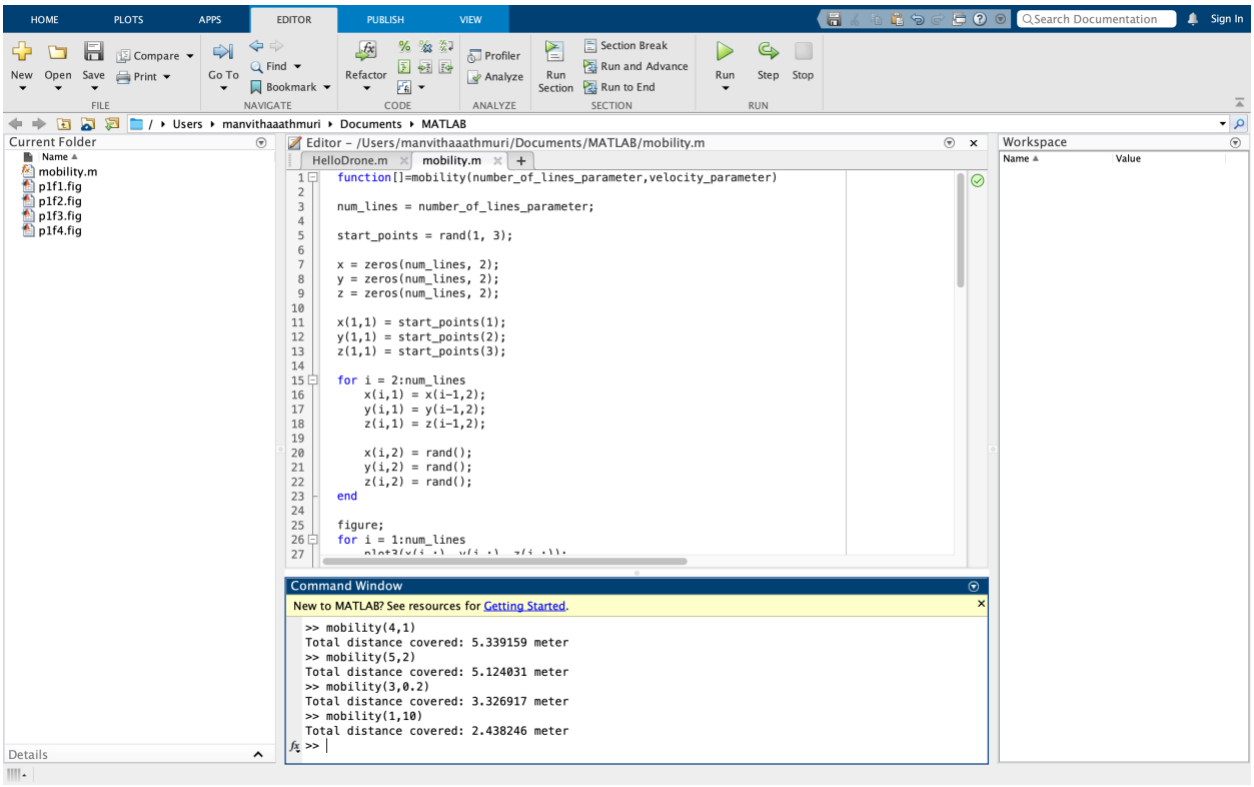
- Project number: Project #1 – HelloDrone
- Your name: Manvitha Aathmuri (R11847781)

The project HelloDrone is about the mobility of drone in a 3-dimensional space. The MATLAB program uses various functions to stimulate the mobility of drone. Let's go through the step-by-step analysis of what this program does.

i. A function **mobility()** is declared which takes two parameters 'number_of_lines_parameter' and 'velocity_parameter'.
ii. Now the variables are initialized like 'num_lines' is assigned to 'number_of_lines_parameter' and 'start_points' is assigned to random values. Using these random values, the initial positions of drones are set.
iii. 'x', 'y', 'z' are zero matrices which are used for taking the reference of co-ordinates in 3-D axes.
iv. Now with the help of loop iterations, the drone positions are generated. Each drone's current position is set to the previous drone's final position.
v. Using the function **plot()**, the trajectories of drone's movement is visually captured/plotted. In addition to this a black line is plotted between the initial and final positions of the drones.
vi. With the help of function **norm()** the distance is being calculated.
vii. As far as the loop iterates, the drones move.
viii. Using '**fprintf()**' function the total distance covered by the drone is printed.

Results and Graphs:

When we run the code giving the parameters in **mobility()** function, such as 'number_of_lines_parameter' and 'velocity_parameter', below are the results.



I have used four different input parameter pairs. Below are the graphs for each of the input pair.

mobility(4,1):
total distance covered: 5.339 mts

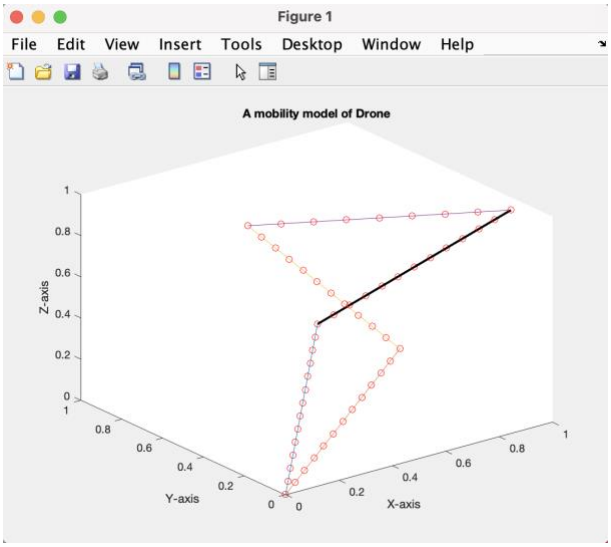


Fig (1)

mobility(5,2):
total distance covered: 5.124 mts

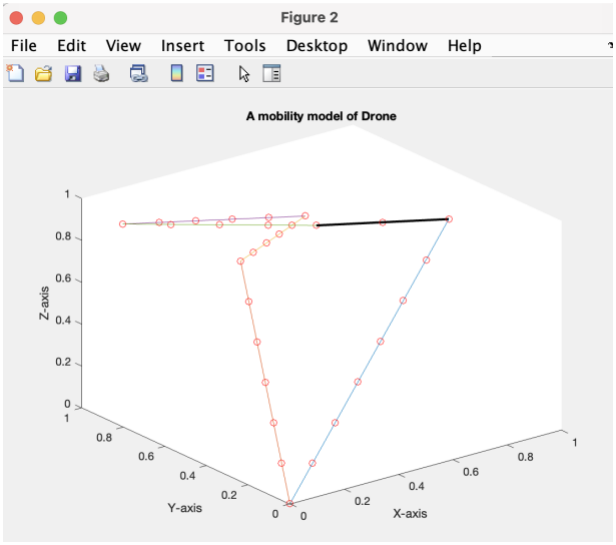


Fig (2)

mobility(3,0.2):
total distance covered: 3.326 mts

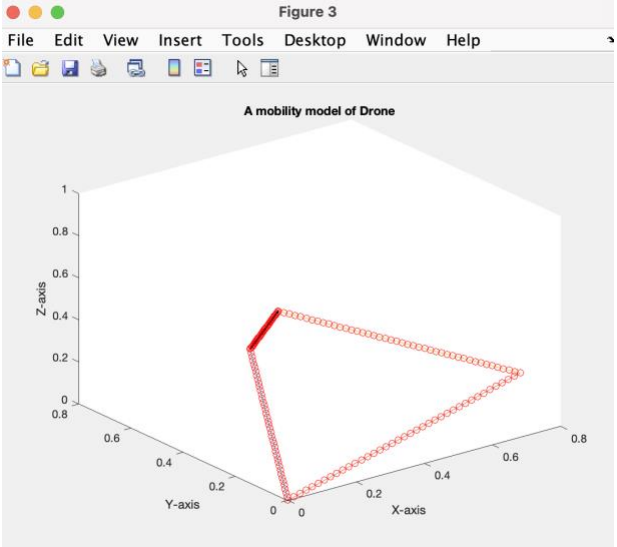


Fig (3)

mobility(1,10):
total distance covered: 2.438 mts

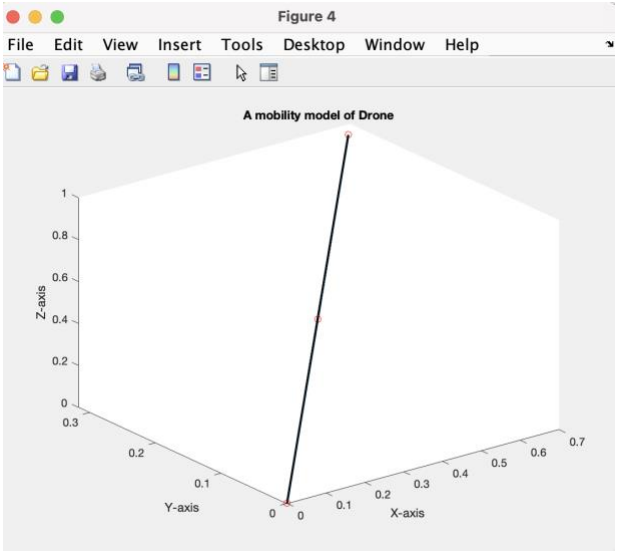


Fig (4)

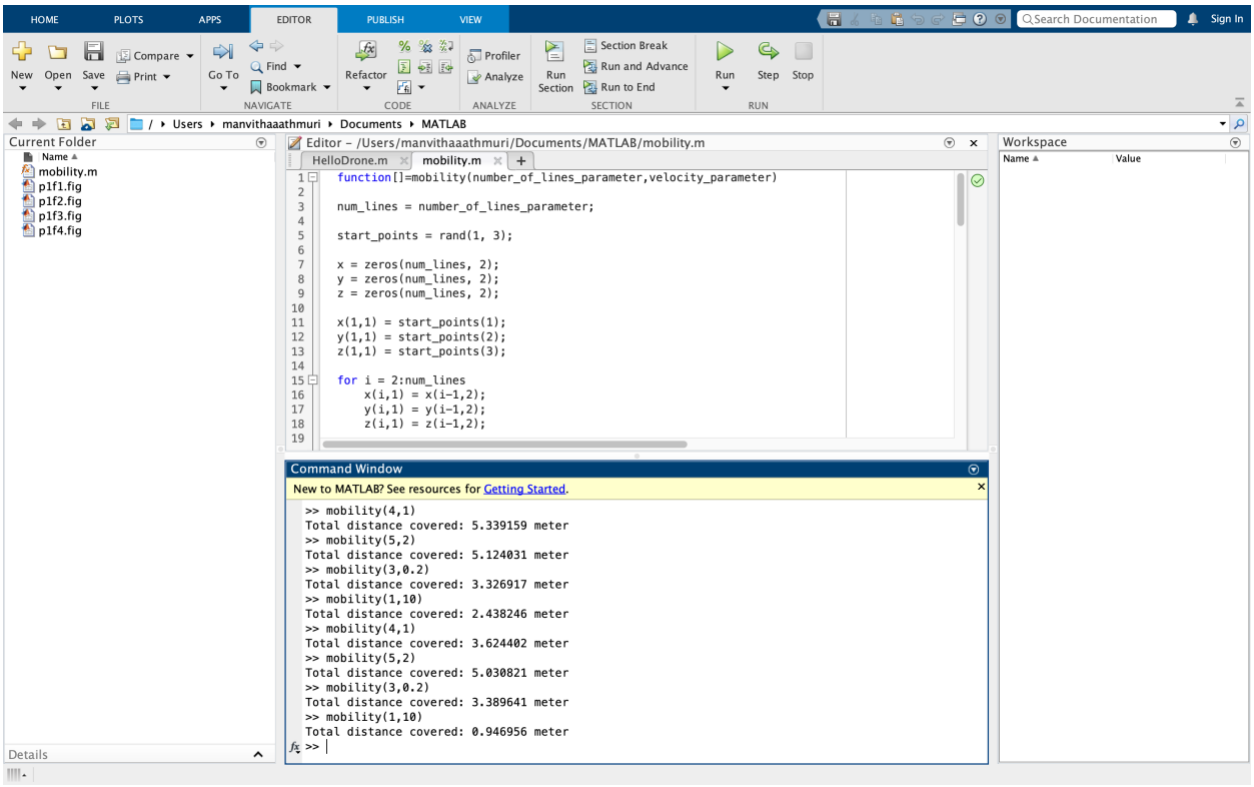
Observations:

As discussed above, the graph depicts the movement of drone in 3-dimensional space. The rings represent the position of the drones at different time steps during their movement. Thus, by visualizing these rings, we can observe how the drones move in 3D space and visualize their paths. And the number of lines that we see on a graph are the number_of_lines_parameter + 1(black line which is just an indication of initial drone to final drone).

There is an observation on the increase and decrease of velocity. Fig (3) & (4) are the examples of this observation.

- Fig (3) shows us the drone movement if the velocity_parameter is decreased, which means the drone is moving slowly. The rings represent that the drone positions which are near to each other. As a result, movement of drone becomes slower & trajectories will be long, and it is clearly seen in the total distance covered i.e., 3.326 mts.
- Fig (4) shows us the drone movement if the velocity_parameter is increased. In this case the rings are very far from each other, meaning the drone is moving faster. As a result, trajectories will be short. The total distance covered is 2.438 mts.

There is one more observation that I personally found it confusing at the initial stage. Even when I run the code with the same parameters, the total distance covered & the trajectory path turned out to be different from the initial results as shown below. But then I have realized that the variable start_point was initialized with random values. Thus, due to randomization this difference in results and graphs are found.



mobility(4,1):
total distance covered: 3.624 mts

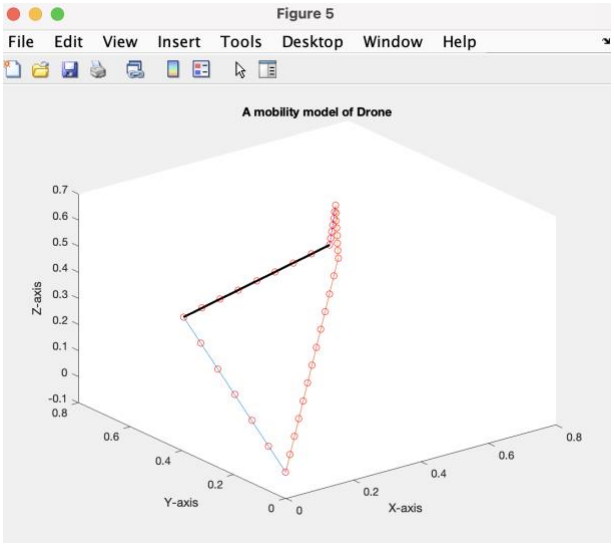


Fig (5)

mobility(5,2):
total distance covered: 5.030 mts

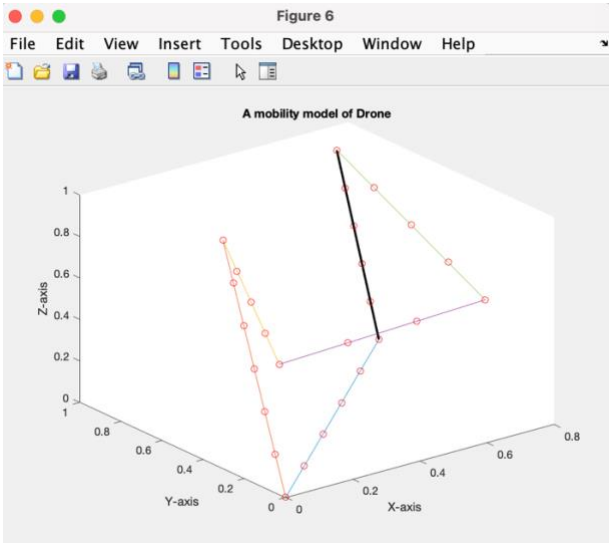


Fig (6)

mobility(3,0.2):
total distance covered: 3.389 mts

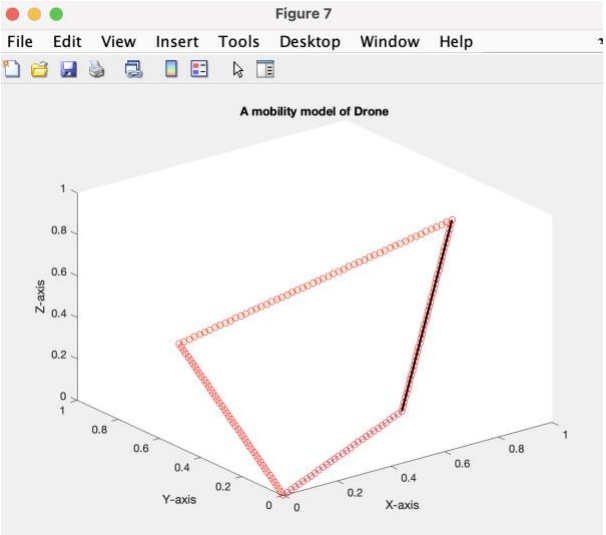


Fig (7)

mobility(1,10):
total distance covered: 0.946 mts

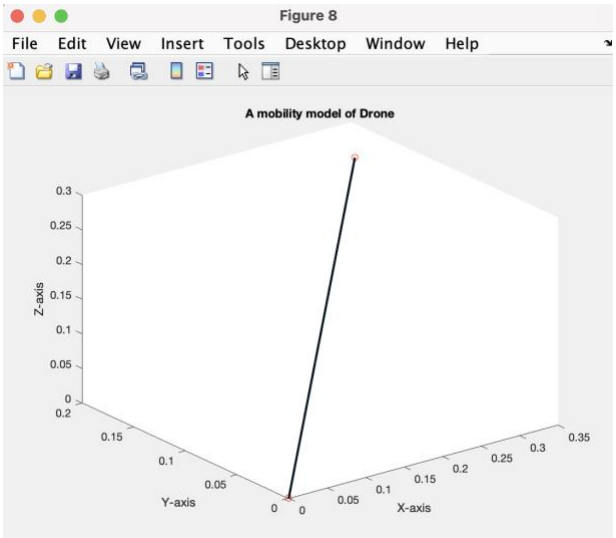


Fig (8)

Above are the observations and results of this project.