Project Report

Name: Sayed Erfan Arefin
Project: 1

Summary

The program simulates the mobility of a drone and calculates distance. I went through the program line by line and my understanding of the program is as follows:

1. It creates area points for 3-axis with uniform distribution
2. It draws the lines on the 3d plane
3. It creates a distance line and marks it with black
4. Based on the given velocity and time step 0.1, it calculates steps and measure the distance, except for the black line and plots it on the 3d figure.
5. Afterwards, it calculates the steps for the black line and plots it on the 3d figure.
6. It displays the total distance.
7. Labels the 3 axis of the figure and puts a title for the figure.
8. Set the 3d viewing for the figure.
9. Makes the 3d figure rotatable for analysis.

Trial run with different parameters

mobility(4, 1)

In this simulation, total number of lines used is 4 and the velocity is 1 meter / second(time step) This simulation has the total distance covered as 5.339159 meters. Figure 1 shows the drones mobility for this simulation.

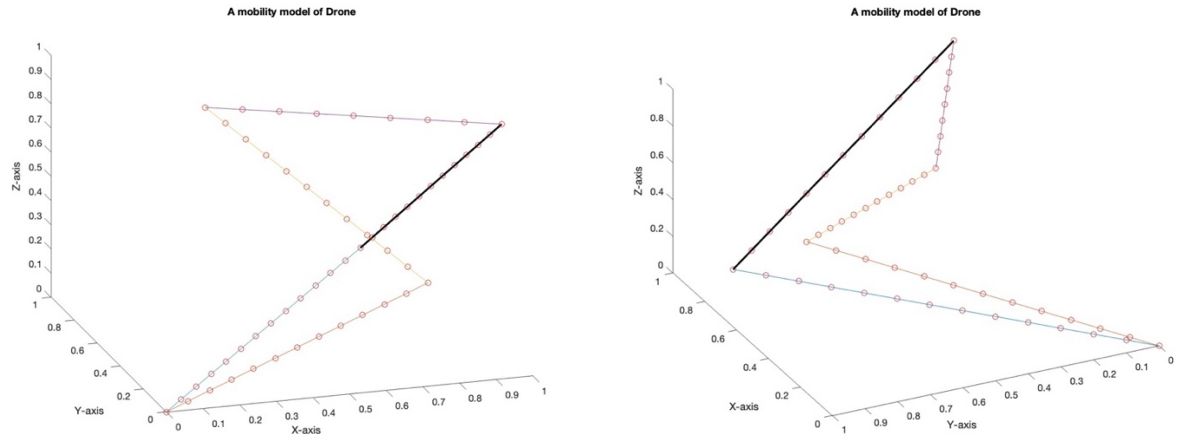

Figure 1: Two different perceptions of the simulated results of drones mobility for 4 number of lines and a velocity of 1 meter/second.

mobility(5, 2)

In this simulation, total number of lines used is 5 and the velocity is 2 meter / second(time step) This simulation has the total distance covered as 5.124031 meters. Figure 2 shows the drones mobility for this simulation.

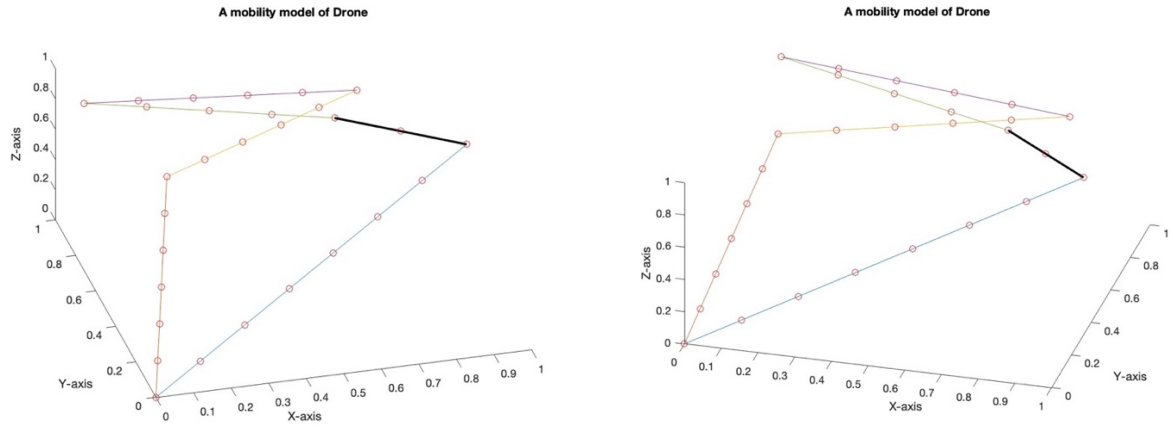

Figure 2: Two different perceptions of the simulated results of drone's mobility for 5 number of lines and a velocity of 2 meter/second.

Code breakdown

1. The following code of the program creates random points in 3d plane based on the given number of lines using a uniform distribution. The rand function creates a value between 0 and 1. The loop fills up the x, y, z values with uniformly distributed points. Each of those have a point (0, 0).

```
num_lines = number_of_lines_parameter;

start_points = rand(1, 3);

x = zeros(num_lines, 2);
y = zeros(num_lines, 2);
z = zeros(num_lines, 2);

x(1,1) = start_points(1);
y(1,1) = start_points(2);
z(1,1) = start_points(3);

for i = 2:num_lines
    x(i,1) = x(i-1,2);
    y(i,1) = y(i-1,2);
    z(i,1) = z(i-1,2);

    x(i,2) = rand();
    y(i,2) = rand();
    z(i,2) = rand();
end
```

The following code displays the lines in 3d plane.

```
figure;
for i = 1:num_lines
    plot3(x(i,:), y(i,:), z(i,:));
    hold on
end
```

2. The following code creates a black thick line for the drone mobility program.

```
plot3([x(1,1), x(num_lines,2)], [y(1,1), y(num_lines,2)], [z(1,1),
z(num_lines,2)], 'k-', 'LineWidth', 2);

point_pos = [x(1,1) y(1,1) z(1,1)];
plot3(point_pos(1), point_pos(2), point_pos(3), 'ro');
```

3. The following code calculates steps alongside the lines that are plotted. It then plots each step with a pause of 0.1 time step in order to create an animated simulation. It also calculates the total distance traveled by the drone.

```
velocity = velocity_parameter;
time_step = 0.1;

total_distance = 0;

for i = 1:num_lines
    dist = norm([x(i,2)-x(i,1) y(i,2)-y(i,1) z(i,2)-z(i,1)]);
    total_distance = total_distance + dist;

    time_needed = dist / velocity;

    num_steps = ceil(time_needed / time_step);

    step_vector = [x(i,2)-x(i,1) y(i,2)-y(i,1) z(i,2)-z(i,1)] / num_steps;

    for j = 1:num_steps
        point_pos = point_pos + step_vector;

        plot3(point_pos(1), point_pos(2), point_pos(3), 'ro');

        pause(0.1);
    end
end
```

4. The following code calculates the steps for the black thick line and similar to previous step, creates an animated display of the steps with a pause of 0.1.

```
dist_black = norm([x(1,1)-x(num_lines,2) y(1,1)-y(num_lines,2) z(1,1)-
z(num_lines,2)]);
total_distance = total_distance + dist_black; % add distance to total
```

```
num_steps_black = ceil(dist_black / velocity / time_step);

step_vector_black = ([x(num_lines,2)-x(1,1) y(num_lines,2)-y(1,1)
z(num_lines,2)-z(1,1)] / num_steps_black) * -1; % multiply by -1 to move in
opposite direction

for j = 1:num_steps_black
    point_pos = point_pos + step_vector_black;

    plot3(point_pos(1), point_pos(2), point_pos(3), 'ro');

    pause(0.1);
end
```

5. The following code displays the total distance traveled by the drone.

```
fprintf('Total distance covered: %f meter\n',
total_distance);
```

The following code displays the labels of the 3d figure and puts a title for the figure.

```
xlabel('X-axis');
ylabel('Y-axis');
zlabel('Z-axis');
title('A mobility model of Drone');
```

6. The following code sets the default 3 dimentional view and makes the figure rotateable in the 3d pane.

```
view(3);
rotate3d on;
```