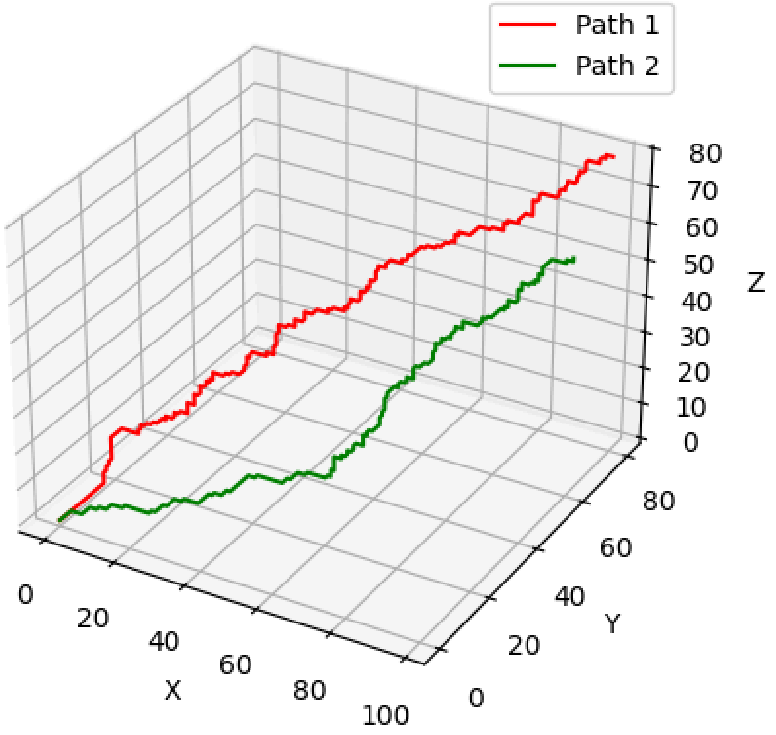
VECROS Assignment Solutions

1. Consider a 3D Grid of points from 0,0,0 to 100,100,100 with unit increment in all
axes. Assign higher weights to some of the points randomly, and zero weights for rest of
points. User inputs two or more sets of {start, end points}.
a. Starting from a given time, determine shortest path for each set, where velocity of
travel for one grid to another is v m/s such that not two paths should not have a
common point at any time.
b. Plot the paths in 3D

SOLUTION: My approach are in this steps:

1)First create a 3d grid using the numpy library of size 101x101x101.
2)Since I am using the `dijkstra` algorithm from scipy library, it requires an argument of 2d graph . So for that reason I have to convert the grid to a 2d graph.
3)I have created two functions: one to convert 3d coordinate to 1d value and again 1d value to 3d coordinate .
4)Using this function I have converted the grid to 2d graph for the utilisation of the library.
5)The graph stores the cost of moving from one coordinate to adjacent coordinate.
6)Now use the graph which is passed to the dijkstra function to get the shortest path in the grid given the start and goal coordinates.
7)The path are stored in paths list and later plotted in a 3d grid using matplotlib library.

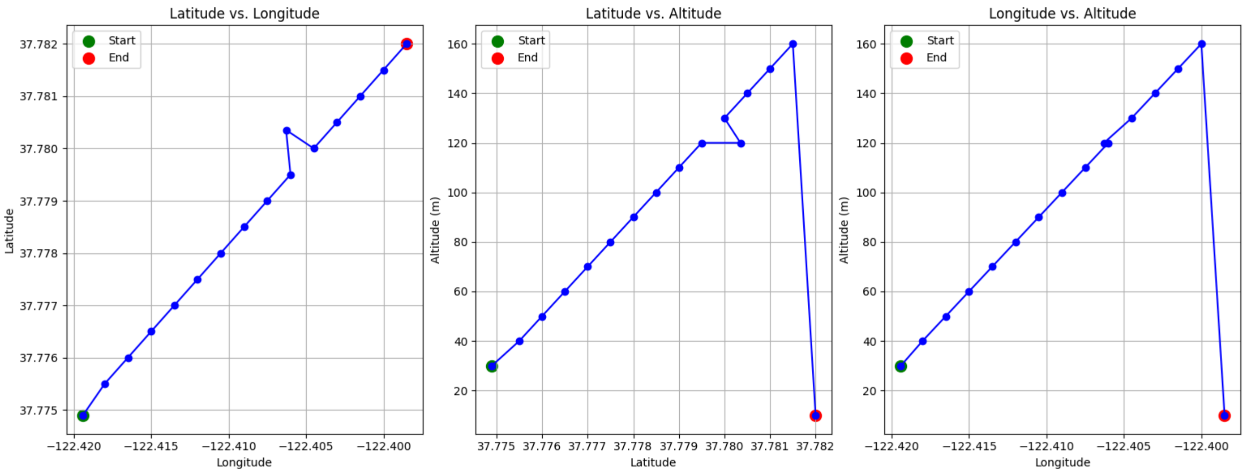


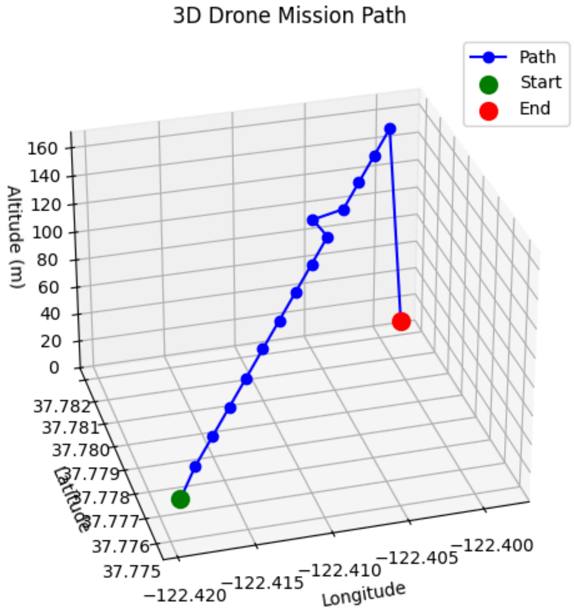
2. Make a dictionary of 15 waypoints with keys being ‘lat’, ’lon’, ‘alt’.
a. Using Dronekit or pymavlink, plan a mission in auto mode using these waypoints
(quadcopter). Drone should land at last waypoint
b. After 10 waypoints, include a new waypoint to the mission at 100m perpendicular to
the current direction of travel and continue the updated mission
c. Print at every instance the estimated time and distance to complete the mission.
d. Plot the path of travel in 2D

SOLUTION: My approach are in this steps:

I had some error in installing the ardupilot software due to which i just have to show visualization using matplotlib which indeed caused some trouble in using the dronekit library.

1)Initialised some random 15 waypoints dictionaries with keys as lat, lon and alt.
2)Now using the geopy library I have calculated the distance between two adjacent waypoints .
3) I have defined a constant speed since I didn't have any simulation or real drone connected in real time to give actual speed.
4)Using the distance and the speed I calculated the time.
5)The distance and time required to reach each next point is printed.
6)After the 10th waypoint i calculated the vector direction between 10th and 11th waypoints and used that to get the new waypoint that is located 100m perpendicular to the vector calculated.
7)Finally it has been plotted in a 2d graph usin7)Finally it has been plotted in a 2d graph using matplotlib.


Latitude vs. Longitude
Start
End
Latitude
37.782
37.781
37.780
37.779
37.778
37.777
37.776
37.775
−122.420
−122.415
−122.410
−122.405
−122.400
Longitude
Latitude vs. Altitude
Start
End
Altitude (m)
160
140
120
100
80
60
40
20
37.775
37.776
37.777
37.778
37.779
37.780
37.781
37.782
Latitude
Longitude vs. Altitude
Start
End
Altitude (m)
160
140
120
100
80
60
40
20
−122.420
−122.415
−122.410
−122.405
−122.400
Longitude


3D Drone Mission Path
Path
Start
End
Altitude (m)
160
140
120
100
80
60
40
20
0
37.782
37.781
37.780
37.779
37.778
37.777
37.776
37.775
Latitude
−122.420
−122.415
−122.410
−122.405
−122.400
Longitude