Intelligent low-altitude air traffic management system

DESIGN DOCUMENT

Group 30
Prof. Peng Wei
Humaid Alkaabi
Jun An Tan
Saad Alsudayri
Suhail Aldhaheri

Revised: 15/10/2017-Version 1

Table of Contents

List of figures/tables/symbols/definitions

1 Introduction

1.1 Acknowledgement

Our group is working with professor Peng Wei to create a system that manages air-traffic in low altitudes. The system will be a simulation interface, a software that simulates the delivery process using drones in 2D where you have x number of drones fulfilling ever growing demands over time. Similar to a flight tracker for commercial planes, our software will display and update each specific drones movement at any point of time.One of the main key features is to make sure that drones will not collide into each other especially when they have intersected trajectories. Our project, being software based requires little or close to none equipment utilities. Therefore we will not be given any monetary aid for the creation of our system. In our second semester, our group will modify and improve the quality of our simulation system to cater to more realistic conditions that might be faced by drones.

1.2 Problem and Project Statement

In today's world of technology, wouldn't it be great if we have our package arriving just in hours after placing an order? The idea of expanding all possibilities and putting customer first is every delivery company's dream. As such, using drones has become a possible, plausible alternative to help speed up the delivery process. After all, the company's profits work hand in hand with the number of "happy customers" they have. But this seemingly lucrative solution seemed to have a common problem i.e: Regulations from the FAA. Hence if we were to consider all of the rules listed by the FAA on drones for the purpose of delivery or not, this project will not be able to fully provide solutions to all of the rules. But, for a start this project will be a starting point for future attempts on providing full solutions for abiding all of the list rules regarding drone activities.

Our project aims to mainly solve the issue of air to air collision between autonomous drones with preset flight paths. In our system there will be 2 main parameters: warehouses and demands. Each warehouses will be allocated a unique drone where they will be responsible for the completion of demands appearing randomly on the map as time goes. The pathway of drone always follows this rule: they will fly from their warehouse to the place of interest and eventually get back to their warehouse.

Hence our system will eventually be a simulation software that when run, checks for the possibility of intersecting flight paths, and delays the delivery till its preset paths are clear. A huge assumption on our part is that the drone will not fail in any cases except for colliding with other drones during simulation. The fanciness of our simulation software to make it as realistic as possible would only be done after solving the main issue of air to air collision.

1.3 operational Environment

The final product that we will have, will be a simulation software created using java eclipse. This simply means that there won't be any potential life threatening hazards to any of the users or hazards that will in any form damage/destroy our product. To prevent our code from being accidentally deleted, we will circulate among members of the team through the use of email on the latest copy of the code. Using the fact that we have a small team, we decided to not complicate our progress by using gitlab.The software that we will be using is called Eclipse, it's free and members of the team will each be downloading it to work on the code for our project.

1.4 Intended Users and uses

There will be 2 groups of users that we are targeting.

1) Delivery companies such as amazon air prime and Google are prying into the prospects of using drones from delivery.
2) Transport companies such as uber elevate that plans to develop autonomous flying cars.

In both cases, our system would be able to simulate air traffic conditions during a time when the airspace is getting populated with different flying objects. The concept of delivering packages from point A to B is the same as delivering humans from point A to B. Flying objects must be able to reach their destination safely especially if there were flying autonomously.

1.5 Assumptions and Limitations

The assumptions, we made, are:

1) Each warehouse has unlimited number of goods for the purpose or delivery.
2) Each warehouse will be assigned a drone for the purpose of fulfilling customers demands (Number of drones might be subjected to change depending on cilent's interest).
3) Drones will always depart from their warehouse to a demand location and back to the warehouse to "refill" and wait for the next order in line.
4) Drones will all be flying at the same altitude at the same speed given by the specifications of the Dji 3 phantom aircraft.
5) Other possible obstructions such as tall trees, building, power-plants… etc will not be present.
6) Preset trajectory paths would always be assumed as a "straight line" in between two points.
7) Population distribution in ames will be assumed "uniformly distributed".
8) No actual drones will be incorporated/use as part of the simulation/testing process
9) Windy,snowy, rainy conditions will not considered.
10) Drones will not fail under any circumstances except for an air to air collision with other drones.

Limitations:

1) Map size will be limited to the city of Ames (think of this as an imaginary square surround ames)
2) FAA drone regulations to be followed.

1.6 Expected End Product and Deliverables

By the end of this semester, we will have a prototype software with basic functionality that will primarily solve our listed problems for this semester.

The problems we are expected to solve this semester are:

1- collisions between drones with intersected pathways.

2- updating drone's latest position on the map.

3- Erasing completed demands off the map.

4- Generation of random demands over time.

5- Developing an algorithm to complete orders in a safe and efficient manner

6-Displaying of simulated results on some form of User Interface
As for our tasks for the second semester, we would foresee that considerations of new problems will be added to allow our simulation system simulate conditions that are more realistic and commonly encountered.

2. Specifications and Analysis

In order to solve the given problems, we came up with an idea that basically uses four different array list and many other functions that work hand in hand to eventually simulate the movements of the drones fulfilling demands of customers. Each functions has specific tasks, in which will be discussed shortly. The lists are customer demand list, request flight list,warehouse location list and ongoing flight list. These four list served as our main databases where we would constantly use them to ensure the conditions are met.

The first function will handle the information needed by the user (customer); the function will be able to read or receive data from the customer and be able to add it, print, manipulate the data and give feedback on this data. (assuming if the user added longitude and altitude out of range). this could be handled easily by defining the boundaries for the vector x and y.

The second function will be doing calculation to find the exact distance between two points, and we have many options to go with in this function such as the distance formula, and the haversine formula.

The third function will be working on finding if there a collision will occur or not. If no collision will happen the flights will keep going and nothing will stop. On the other hand, if a collision might happen the system will prevent one of the flight from taking off until the other flight finish it flight and go back.

the fourth function is to calculate the new location of ongoing flight with respect to the time or distance traveled. this is a complicated function, and we only found one way to implement this function.

finally, the simulation function can be done in many different ways: the best of them are that the function could use an external website or program such as google map to show how flight are moving on a 2D paths. Another way is that we could create a map that has the most important places and stores in Ames, IA.

2.1 Proposed Design

The design our team proposes consists of many functions each with specific tasks, and we will be having four arraylist to act as our database which keeps track of customer demands, initiating drone clearance between the pathway from warehouse to a demand. We will have another function specifically the trajectory function, which will constantly be updating and displaying the current location of the drone on our map. Tasks of other functions might include calculating some form of distance, plotting of our map ...etc

See below:

1- several functions that make it possible to add, print and manipulates the main three lists.

a- Add: help to add information to the lists. this information is going to be added by the customer.

b- print: keep track of the data and make sure the function works.

c- manipulate functions: help to fix an error made by the user. out of the range (City Of Ames)

2- a function that calculate the distance between two coordinates points.

3- a function that check if there is a possible collision if we set a new trajectory in the air field.

4- a function that calculate the new location of ongoing flight with respect to the time or distance traveled.

5- An output type of user interface (front end) that shows the simulation of the system. It could be any type of a platform such as google maps or the software that we had been using.

2.2 Design Analysis

The software so far has multiple functions including checking if the aircraft will collide with another aircraft and checking the trajectories. We also have another function that deals with the demands which means that the software will check what the user wants and it will implement it. Another function will check the nearest warehouse to complete the tasks required. The warehouse will have the aircraft or drone and the stuff that needs to be delivered. The last function is the real time plotting which is yet to be completed. We are having difficulty updating the drone's position in real time and also showing a real map that shows the aircraft's position.

3 Testing and Implementation

The testing process throughout the 2 semesters will always be: 1) test it individually as a function first 2) test it again when the function is added to the main buck of the code. The brainstorming process would always be:1) Personal attempt in solving problems 2) researching on relevant methods 3) discuss as a group. Regardless of what types of problems that we will be adding on or be currently solving, these are the kinds of approach we will be going about as a team to achieve our final goal.

In the event where testing is not successful, we will have to debug and rework on the code.

Parts that were tested in our projects were:

1- mathematical test for the distance function:

> When we were done with the coding of this function, we tested it by plugging in some coordinates within the range of (Ames, Ia). We then calculated the distance using the function and compared it to the actual distance shown in google maps.

2- Debugging process to ensure function works for all possibilities:

> We plugged in random points using the coordinates from google map in the form of (x1,y1) and (x2 , y2) and observed its result. A specific function should gives us a close to gps location when compared to the two points on google map.

This is one method that we will be using to test part of/ all of our code and solve potential problems before accumulating them without knowing the exact cause of failure. We would always like to try it with the most primitive methods and observe the behavior of the function with the requirements that we had to meet.

3.1 Interface Specifications

We will be using java coding (Eclipse) as a way to test our simulation for this semester. We will do all our work on eclipse as we found it to be more convenient to use for the making our our front end code as compared to C language.

Hardware and software

We are using the Java software for this project. We are writing the java code in eclipse and that is where the whole work will be done in for the rest of this semester. We will later on use this software on an aircraft next semester hopefully.

3.2 Process

1- Mathematical test for the distance function:

The function we used to find the distance:

$$\operatorname{haversin}\left(\frac{d}{R}\right) = \operatorname{haversin}(\Delta\varphi) + \cos(\varphi_1)\cos(\varphi_2)\operatorname{haversin}(\Delta\lambda).$$

We coded the formula above, and in order to test it , we pick two different points compared it with google maps result, good enough it worked out well.

2- Testing of function:

In the trajectory function (simulating the movements of the aircraft), we tested it by using two points in order to try the algorithm and to see how it works.

we would like to show the following (in progress):

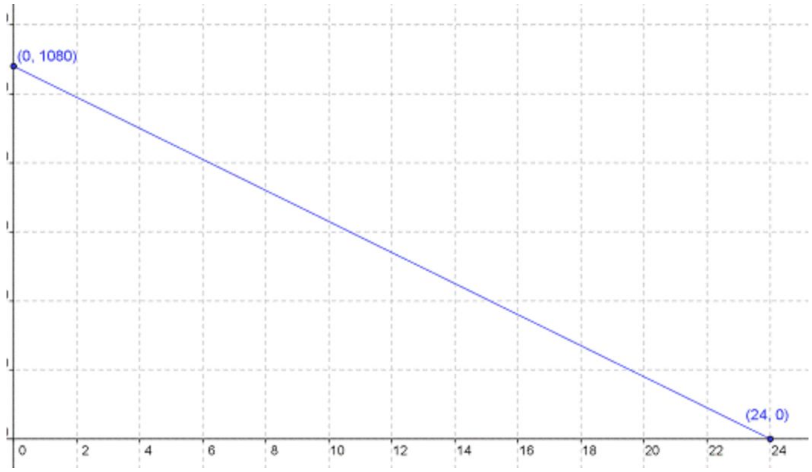

fig.1.

3.3 Results

There had been some minor failures during our coding process but fortunately they were solved within a couple of days. These minor failures that cause things to work a little weird was due to the lack of understanding in java. We kind of expected a little of such setbacks due to the composition of our team and that half of us had to "learn on the spot" a new programming language. Looking at how we have our plans planned out, our progress has been great, we managed to achieve the minor milestones by completing functions that are the solution to our designated problem list. As of now, one of our pending problem that we had been trying to solve is figuring out a way to plot and refresh the drone's position that will be displayed on the map. Until now things has been relatively smooth for us, and we hope to keep our progress on schedule. In the future, more functions will be added to cater to our cilent's request in turning our software into a realistic simulation program.

Some screenshots of parts of the code that works and in progress:

```
Demand List
Demand Number: 1, Latitude: -93.74372015798774, Longitude: 41.98132920699946
Demand Number: 2, Latitude: -93.77933326597802, Longitude: 42.054606128978534
Demand Number: 3, Latitude: -93.83253787296351, Longitude: 42.016821184989325
Demand Number: 4, Latitude: -93.77950530997798, Longitude: 42.00087821199388
Demand Number: 5, Latitude: -93.78429386797667, Longitude: 42.02184807198789
Demand Number: 6, Latitude: -93.76885291898088, Longitude: 41.990599060996814
Demand Number: 7, Latitude: -93.7728242679798, Longitude: 42.0025636399934
Demand Number: 8, Latitude: -93.75143346398563, Longitude: 41.981427196999434
0
 shortest distance is:  20360.349066779065

 at index : 0

Removing  Demand Number:1
```

Fig.2.

4 Closing Material

4.1 Conclusion

So far we have tested functions that calculates distance between 2 points, getting the gps location of the drone, plotting of map with limits in numbers, changing google map's degree coordinate into numbers, generating random demands numbers, arraylists for database storages and flight initiation, developed an algorithm that will satisfy the fulfilment of demands in an efficient manner (in progress),the trajectory function that will update drone's gps location along with time on our yet to be done output map(front end) and one of the more important function that prevent collisions of drone with intersecting flight paths(in progress). These functions have already been both tested individually and when combined into the main code works just as fine. However, what we are seeing here is just "numbers".We plan to stick to our goals by first finishing up the codes that will solve our designated problems before working on the front end of our system(output screen). In terms of work progress wise, we would say that we are on our way in successfully making a "basic enough" prototype that will be upgraded in the next semester.

4.2 References

https://stackoverflow.com/

http://www.movable-type.co.uk/scripts/latlong.html

4.3 Appendices

Any additional information that would be helpful to the evaluation of your design document.

If you have any large graphs, tables, or similar that does not directly pertain to the problem but helps support it, include that here. This would also be a good area to include hardware/software manuals used. May include CAD files, circuit schematics, layout etc. PCB testing issues etc. Software bugs etc.

We foresee that our simulation output would look something like in this fig.3.

fig 3.