Intelligent low-altitude air traffic management system

Table of contents:

Section 1: Introductory Material **3**

1.1 Acknowledgement: 4

1.2 Problem Statement: 5

1.3 Intended users and uses: 5

1.4 Assumptions and Limitation: 6

1.5 Expected End Product and Other Deliverables: 6

Section 2: Proposed Approach and Statement of Work **7**

2.1 Functional requirements 7

2.2 Constraints considerations 7

2.3 Technology considerations & Task Approach 7

2.4 Safety considerations 7

2.5 Related work / market survey / literature review: 7

2.6 Possible Risks and risk management 8

2.7 Project Proposed Milestones and evaluation criteria 8

2.8 Project tracking procedures 8

2.9 Objective of the task 8

2.10 Expected Results and Validation 8

Section 3:Estimated Resources and Project Timeline **9**

3.1 Other resource requirements 9

3.2 Financial requirements 9

3.3 Project Timeline 10

Section 4:Closure Materials **11**

4.1 Conclusion 11

4.2 References 11

List of figures:

Fig. 1 Flow plan
Fig. 2 Some Results

Section 1:

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 an 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 are not given any monetary aid for the creation of our system.
In the second semester, our group will continue to improve the system through the addition of additional features that might either be requested by our client or proposed by us.

1.2 Problem 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 Intended Users and Uses:

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

1) Delivery companies such as amazon air prime, Google that are looking into delivering packages with the use of drones
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 at the time if in the event 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 will be flying autonomously.

1.4 Assumptions and Limitations:

The assumptions we made will be:

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" (might be subjected to change depending on cilent's interest)
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

1.5 Expected End Product and Other Deliverables:

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

The problems we are expected to solve are:

1- Ensuring no collisions between drones with intersected pathways
2- Updating drone's latest position on the map by displaying it at the front end display
3- Erasing completed demands off the map
4- Generation of random demands over time on map
5- Developing an algorithm(finding the shortest pathway between warehouses and demands) that will help to satisfy demands in an efficient manner
6-Displaying of simulated results on some form of User Interface (front end display)

As for the tasks for the second semester, we are only briefed to expect the addition of realistic problems into our basic prototype that will be completed in semester 1. The client's intention is to create a system that is capable of simulating realistic scenarios that drone deliveries might be facing.

Our final product sits in a system called: eclipse (java). We will be solely using the software for the creation of all back-end and front-end capabilities. All tests/product demo will be done using this particular software.

Section 2: Proposed Approach and Statement of Work

2.1 Functional requirements

The simulation will be able to continue running until the stop button is pressed. There will be no collision between any 2 drones on intersecting flight path at any point of time. Our front end map would be based on ames,IA that would have most landmarks/buildings found off google maps. If our software allows the importation of google maps to our user interface, we will then use the imported map as our front end display. In the event where it doesn't allow that, we will create a front end map that will have most of the landmarks/buildings found off google maps.

There will be 3 main symbols on the map that serves as a distinguishment between warehouses, customer demands and drones. For example, we could use a square to represent warehouses, a plane symbol to represent drone and a dot to represent customer's demands. Drones on the map will only "fly in a straight line" between 2 points and to ensure that there will be no air to air collision at any time, our system will only permit the flight between 2 points after checking for intersecting flight paths. Which simply means that if there is intersecting flight paths between any 2 points, our algorithm will ensure that the flight paths will be granted a "go" one at a time.

Demands will appear on the map as time goes by and warehouses will be required to fulfil them in the shortest possible time. This is achieved by considering the shortest distance between demands and the pre-selected locations of the warehouses. Our rule of thumb is that warehouses will fulfil request/demands by looking at the distances between them and the demands. At the same time, users will notice that drone will be travelling back and forth from their specifically assigned warehouses after the completion of a demand. Finally, this software would be able to at a glance observe the locations of drones, demands, warehouses at any point of time into the simulation.

2.2 Constraints considerations

Non technical requirements are not applicable to our project. From the few programming classes, we will follow the standard coding protocol the way we learnt it. There won't be any "ethicalness" issue in our project for we will cite whatever equations/notes that helped us in the making of our project.

2.3 Technology considerations & Task Approach

Having living in such a technology based century, the perks involves in the readily available information at a click distance away. Making use of technology especially the internet allows us to debug our code and seek for potential solutions the “right way” by discussing on forums that were meant for specific programming languages.

Task Approach:
Due to the fact that our project is 100% software based, we do not need to care anything about hardware designing.In terms of software design: we had broken down our problem/project into parts that seemed like a good approach in achieving our final goal. Generally, our code involves a mixture of algorithms, each responsible for specific tasks. When combined together, we would achieve the specific requirements indicated earlier. Using the lists of tasks in section 1.5, we simply plan to break down the problems into parts/functions that can achieve its desired outcome. Generally our approach for each function will be: converting ideas into code,test them and finally debug the code if it doesn’t go according to plan. In the event, when we are stuck at debugging , we will look for similar encountered problems that might have had a solution to it. In order to keep things trackable and easier for debugging, we would make sure each function completes a specific task. Only then after making sure that each functions works the way we wanted, we will them put them together and re-test it.

The functions you see below would be our goals in weeks to come. We will first complete most of our backend functions(code) before thinking on a way to do our front end.

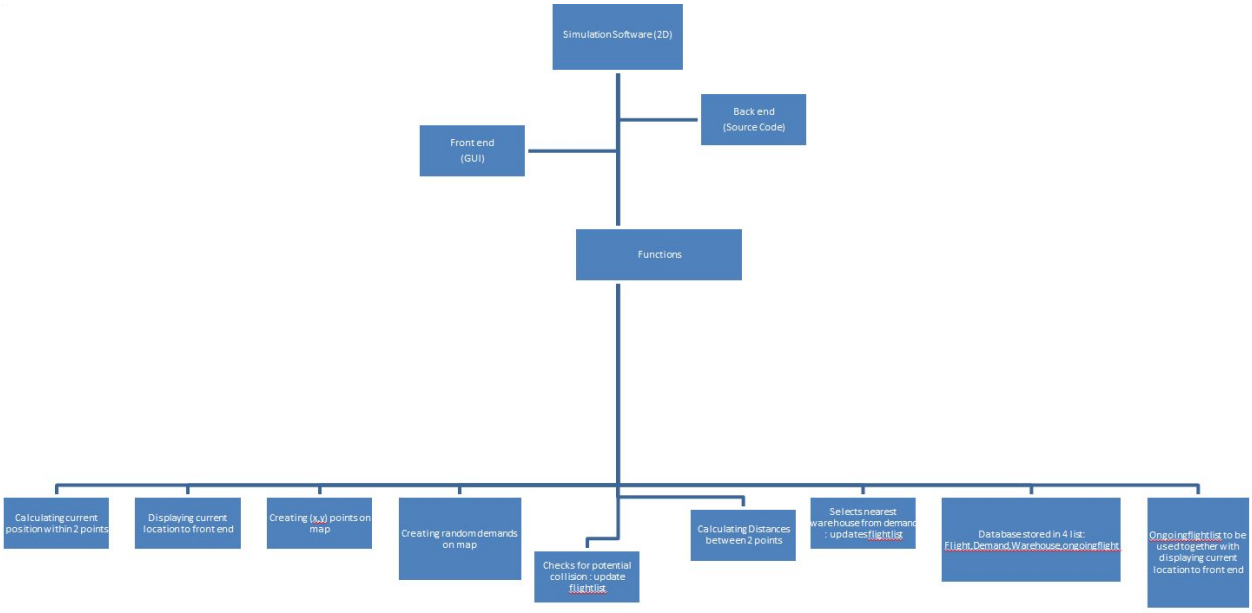

Fig 1. FlowPlan

2.4 Safety considerations

Not applicable for our project. Our final prototype/product will not involve the usage of any “real” drones as it will be a simulation system that works with eclipse. From the debugging process to our final demo, and eventually to whoever that might be using our product, things users would need: a computer.

2.5 Related work / market survey / literature review:

There has been many work done with aircrafts. For example, Amazon started working on Amazon Prime Air earlier this year which is basically a drone that delivers packages. It is based in kentucky and deals with other airports across the states. The product has its limits such as: packages has to be small enough to fit the drone and has to be delivered to customer within 30 minutes as long as they are in a radius of 16Km. Google have been testing the same concept with a project called “Project Wing”, unfortunately they haven’t been very successful in finding a way to drop their packages safely to their customers which will be a safety breach.

2.6 Possible Risks and risk management

During our research process, we had decided that java is the best option for us in creating a software that interfaces GUI with it. It will be substantially easier for us to convert java codes to java scripts. Yet, as our team consists of full electrical engineering majors, we were only introduced to C programming in our curriculum. Half of the team, fortunately had taken a technical elective class that teaches the basics of java. But still that might hinder our progress on how fast we can proceed. Members that have no prior java knowledge have to apply the “self taught knowledge” almost simultaneously during the coding/debugging process. Due to some simple misunderstands on the concept, we tend to waste a lot of time in the debugging process. Another potential problem that we will be facing, is the use of GUI. Our progress will be further delayed due to the fact that we have no clue at all as to do the convert from code to GUI. We will definitely be spending a whole chunk of time researching and putting things to work simultaneously.

2.7 Project Proposed Milestones and evaluation criteria

Milestones	
Sept-17	Deciding on what programming language to use, setting up meeting, discussing plan with members ,dividing tasks
Aug-17	Complete gps location function , calculating distance function , plotting (x,y) map for testing purposes, creating random demand location
Oct-17	Complete remaining functions listed in flowchart (fig1)
Nov-17	Get GUI up and running
Dec-17	Presentation of 1st prototype
Jan-18	Pending: awaiting for new tasks
Feb-18	Pending: awaiting for new tasks
Mar-18	Pending: awaiting for new tasks
Apr-18	Pending: awaiting for new tasks
May-18	Wrap up documentation & Product demo

Our testing procedure has been discussed in sections 2.3. But generally, the plot map function created will be our primary way to check for results without the need of creating a front end display. We will validate our results by looking at numbers and logic.

2.8 Project tracking procedures

The weekly report that we will be uploading to the web keeps track of our progress throughout both semesters. At the same, we will constantly be in touch with our advisor/client on matters regarding our progress and if he would like to add in more tasks to the project. At any point of time, we will make sure that as long as new code is written, it will work smoothly before the day of demo.

2.9 Objective of the task

The final goal is develop a software that will monitor the movement of simulated drones and avoid collision between them. This simulation acts as an attempt to replica realistic situations that might happen to drones during delivery.

Due to the fact that we had not started working on our front end display, results of some working tests in numbers is shown below.

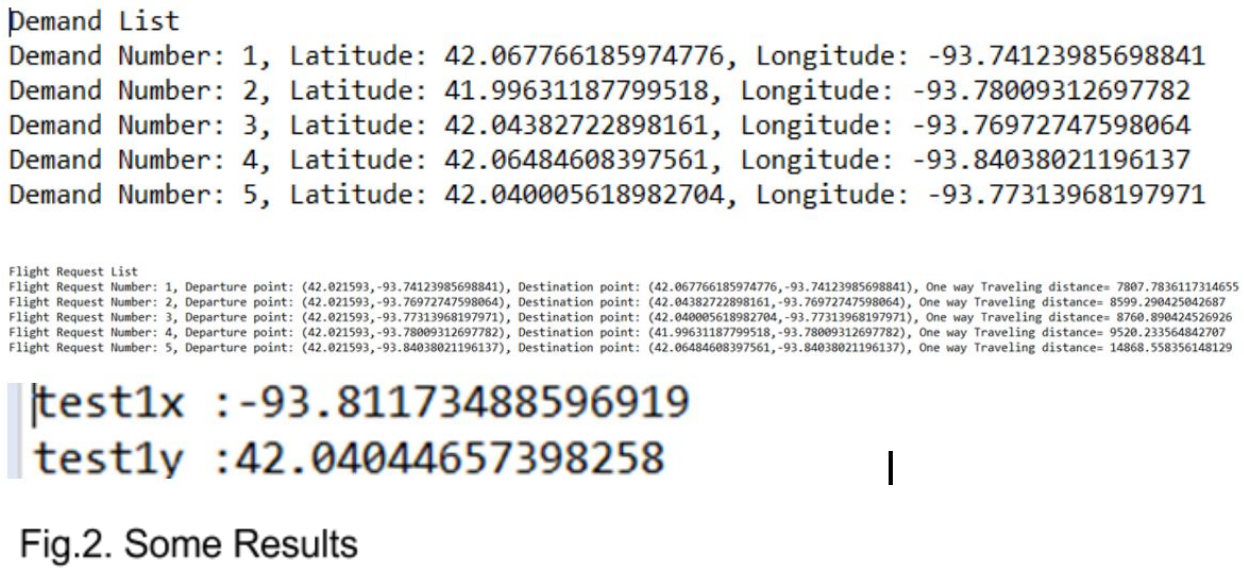

Fig.2. Some Results

2.10 Expected Results and Validation

The desired outcome is having a working software that will be able to monitor trajectories of drones. It will be able to avoid collision with other drones and will be suitable and easy for a user to use.

Up to now, we have been testing our functions with the use of our plot map function that provides us with the coordinates/ numbers that will eventually correspond to a location at the front end display. Since we have not started on our front end display, we ensure that we are getting the results we want by observing the outputs of our function in terms of numbers alongside logic.

Section 3:

3.1 Personnel effort requirements

3.1 other resource requirements

No physical parts will be used except a program called Eclipse.

3.2 Financial requirements

Not applicable to our project. Eclipse is a free software that doesn't involve any fees. Moreover, there isn't any hardware component that we will be using.

3.3 Project Timeline

Week 5	We did more research on the project. We set up the platform and the compiler. We looked over the GPS equations and we worked on the project plan.
Week 6 (Sep 25th to October 2nd)	We will look into functions that will deal with the collision between the aircrafts that will cross paths.
Week 7 (October 2nd to October 9th)	We will work on the algorithm of the collision function and we will write up the code for it.
Week 8 (October 9th to October 16th)	We will begin figuring out how to update the drone's latest position on the map for the user to see.
Week 9 (October 16th to October 23rd)	We will write up the code lines to update the drone's position. We might use a GPS.
Week 10 (October 23rd to October 30th)	We will figure out the how to delete or erase the demands from the map after the aircraft has completed it's task

Week 11 (October 30th to November 6th)	We will do more work on erasing the demands and making sure it is done accurately.
Week 12 (November 6th to November 13th)	We will write a function that will unable us to get random demands over the time we have.
Week 13 (November 13th to November 20th)	We will figure out an algorithm that completes demands around the warehouses in a suitable efficient manner.
Week 14 (November 20 to November 27th)	We will make sure that the aircraft is taking the fastest and shortest path to complete it's task. We will begin the simulation.
Week 15	Do more tests and make sure everything is working and our end goal is completed.
Week 16	We will finish and present our prototype 1by running the simulation.
Week 17	finals
Sem 2	Pending tasks

4 Closure Materials

4.1 Conclusion

Our plan is to develop a working software that will simulate realistic conditions for possible planning or analysis of autonomous drone delivering process. The primary function of this software will ensure that drones would not collide into each other at any point of time. The output of the software will function like radar systems found in air traffic control towers, where it provides real time monitoring of aircrafts in the vicinity. We plan to make our 2D software "realistic" by considering functions that were provided or used in an actual air traffic control radar system. By following the plan that we had set, we hope to deliver our completed product at the scheduled timelines. After all, the success of the team depends on our competency,teamwork and drive to get things done.

4.2 References

[1] https://stackoverflow.com/
[2] http://www.movable-type.co.uk/scripts/latlong.html