Intelligent low-altitude air traffic management system

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

Revised: 30 Nov 2017

Table of contents:

List of figures:

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 an x number of drones fulfilling an ever growing demand over time. Similar to a flight tracker for commercial planes, our software will display and update each specific drone movement at any point of time. One of the main key features is to make sure that the drones will not collide into each other at any point of time. Our project, being software based requires little or no 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 had our package arrive just hours after placing an order? The idea of expanding all possibilities and putting customers 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. However, this seemingly lucrative solution has 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. However, 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 two 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 the drone will 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 it runs, checks for the possibility of intersecting flight paths, and delays the delivery till its path is clear. A huge

assumption on our part is that the drone will not fail in any case 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 two 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 any 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 they 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 client's interest)
3) Drones will always depart from their warehouse to a demand location and get back to the warehouse to restock for the next delivery.
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 obstacles 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 be 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, IA (think of this as an imaginary square surrounding 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
2- Updating drone's latest position on the map by displaying it either at the front end display or some temporary output
3- Erasing completed demands off the map
4- Generation of demands on map
5- Developing an algorithm that will help to satisfy demands in an efficient manner
6-Displaying 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 can achieve.

Our final product sits in a system called Eclipse, and we are using Java language. We will be solely using the software for the creation of all back-end and front-end capabilities in this semester. All tests/product demo will be done using this particular software. In the event where we fail to achieve the display of certain portions in the given software, we will be displaying the

results in a temporary “stand in” software: Matlab. We will be delivering all of the above 6 points in a visual demo in the event where the output GUI is not done.

Section 2: Proposed Approach and Statement of Work

2.1 Functional requirements

When the GUI is fully operational, the simulation will be able to continue running until the stop button is pressed. There will be no collision between any drones 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 three 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 a drone and a dot to represent customer’s demands. Drones on the map will only “fly in a straight line” between two points and to ensure that there will be no air to air collision at any time, our system will only permit the flight between two points after checking for intersecting flight paths. Which simply means that if there is intersecting flight paths between any two points, our algorithm will ensure that the flight paths will be granted the permission to fly 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 the 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 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 learned 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 lived in such a technology based century, the perks involve 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 two parts that seemed like a good approach in achieving our final goal. The two parts are as follows: backend and front-end. We will be working on the completion of all back-end codes before progressing over to our front-end.

Generally, our backend code involves a mixture of algorithms, each responsible for specific tasks. When combined together, we would achieve the specific requirements indicated earlier. Then making use of 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, After that we will test them and do the debugging process if the outcome isn't intended. In the event, when we are at a headlock, 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 one specific task. Following which, when each functions works the way we wanted, we will compile them together and retest it as a whole again.

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

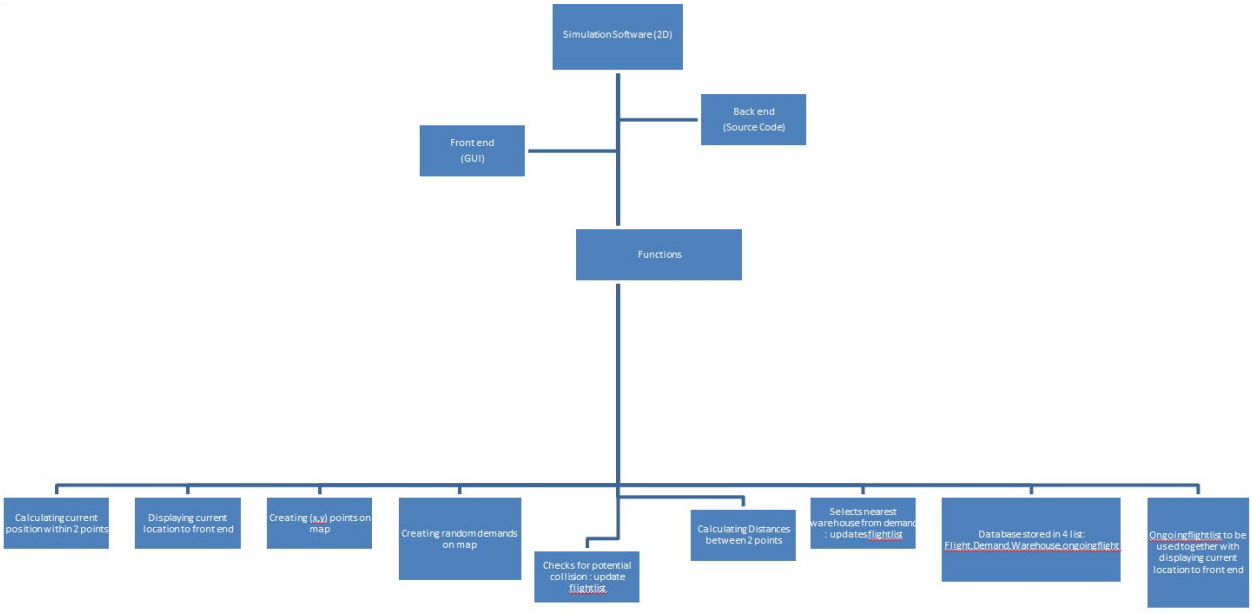

Fig 1. FlowPlan

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 customers 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 misunderstanding on the concept, we tend to unintentionally 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	Create a .exe file for a comprehensive testing

	and debugging process of all backend codes
Dec-17	Wrap up documentation & 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

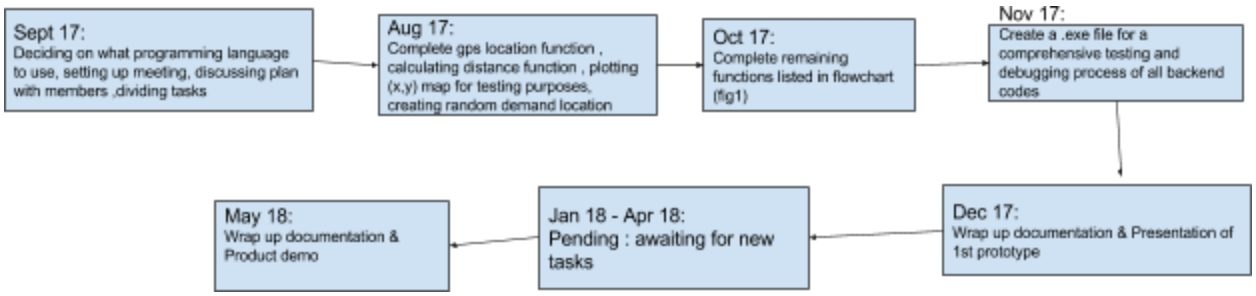

Fig.2 Milestones

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 through numbers,logic and comparison of data with given results provided by google map. This is further discussed in section 2.10.

2.8 Project tracking procedures

The weekly report that we will be uploading to the web page 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 always try to make sure that any added code works 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 travelling from one point to the other and avoid collision in the process of doing so. This simulation acts as an attempt to replicate realistic situations that drones might encounter during delivery.

Due to the fact that we had not get our front end display to be fully operational, results of some of the working tests were shown below.

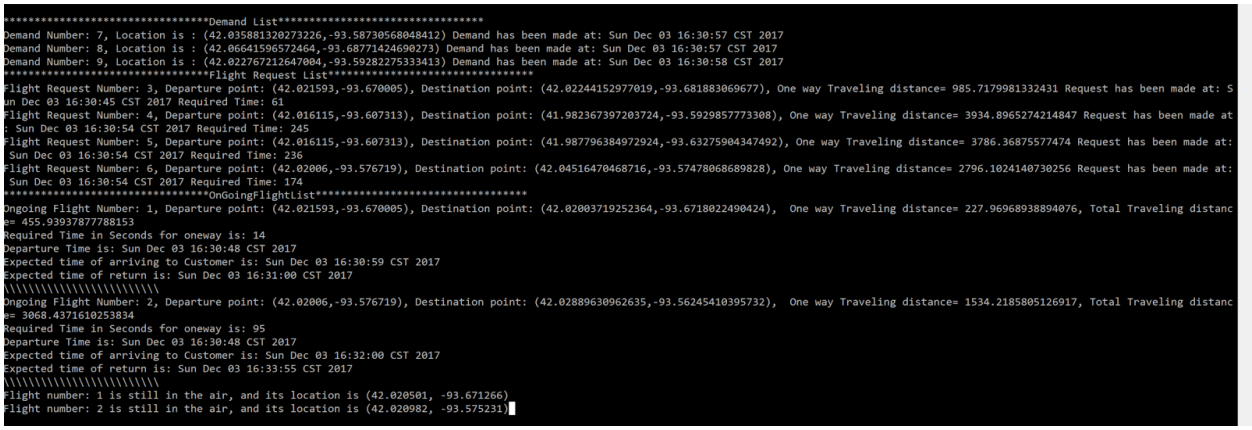

Fig.3. Some Results

2.10 Proposed solution

There are a couple of ways as to “how can we create a software prototype for the management of low altitude aircrafts”. After much brainstorming, we came into a conclusion that this is the method that we will be using to provide the solution to the problem. The design our team proposes consists of many functions each with specific tasks. We will be creating four arraylist to act as our database which keeps track of customer demands, initiating drone clearance between the pathway from the warehouse to a demand, storing of warehouse information and data for ongoing flight. Then 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 using the data from the ongoing flight arraylist. Tasks of other functions that will be responsible in achieving our deliverables are shown below.

The problem is broken down into two main portions that will provide our potential solution:

1) Front end display
2) Backend (coding)

Front display ensures the correct output: we will visually check when our software is ready.

We were tasked to create a software that fulfills these conditions.

1- Ensuring no collisions between drones
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 that will help to satisfy demands in an efficient manner
6-Displaying of simulated results on some form of User Interface/ Visual output

Front-end: We will be creating a java based GUI
Back-end:
We had decided to create multiple simple functions that will be assigned in doing a specific task.Following which we will be making use of these functions for the creation of more complex functions.

The functions are:

1) Distance function:

Calculates distance between two points of the map using latitude and longitude parameters.These information can be obtained from google maps.

2) Calculating current drone position function:

Calculate current drone position using basic math techniques of sine,cosine, tangent properties, projection of axis. For e.g, in a x-y plot, when x and y are both positive, angle is positive.Likewise, when x-y is negative, angle is negative. So by incorporating all four possibilities of " different angles" we can get our current drone position.

3) Random demand generator:

Using a random algorithm to generate different locations for the representation of "random customer demands".

4) Creating arbitrary map function:

This function is needed till we can find out a way to import all data from google map for the purpose of getting latitude and longitude information of demands. This serves as an arbitrary map to get the latitude and longitude information by creating a boundary with 10000 points vs 10000 points.

5) Collision checking function:

This will be one of the more complex function that uses previous functions to ensure that whenever there is an intersecting path/paths, permission will not be granted for both flight paths. In the event where one path is granted permission(ongoing), the other intersecting path/paths will not take off, it will be assigned to fulfil other non-intersecting paths demands.

6) Few arraylist for storing different types of data (acts as a database of the system)

There will be 4 arraylist used as a database of our system. In future should there require a modification of our current system, arraylist can easily be added for that purpose. The 4 four list includes: warehouse list,flight-request list, demands- list and ongoing flight list. The primary data found in these lists would include the latitude, longitude information and possibly time information at which the demand has been made.

The amount of data parameters will be constantly evaluated during the process of developing the code.

7) Transferring demands data to flight request list:

This function will literally do the task as indicated by its name

8) Selecting nearest warehouse function:

This function will allocate demands to the nearest warehouse associated to it.Although in the limitations section, it is indicated that the only way for our drone to fail is through an air to air collision, “battery” life would not be posing as a problem in our simulation. But to simulate the realisticness of drones having a short flight time, the best way to show this feature is to follow this algorithm.

9) Plotting of current drone position function through the use of stored database:

This function would be the main function used to output our code to a front end GUI.

10) time taken function:

This function acts as an upgrade for the realisticness of our product. For now, this is secondary but will eventually be completed in the next semester. This function will calculate, the time taken from A to B and work out the arrival and departure times for drones.

The strengths of our proposed design includes:

1) 100% collision free algorithm between drones
2) Algorithm that fulfills demands the most efficient way
3) Allowing users to see a possible “what if” scenario when drones are to be allowed for the purpose of delivery
4) Providing users with the basic idea of “what will happen if...”
5) Allowing users to brainstorm for more possible scenarios
6) Allowing users to save money on real time drone testing scenarios for data interpolation
7) User friendly usage of software

The weakness of our proposed design includes:

1) limitations of the map size
2) Not much options as to what the users can do when running the simulation (minimal user to software interaction)
3) Massive amount of assumptions
4) Drone's flight paths are only in a "straight line"
5) Depicting only a scenario

Observations of the proposed solution:

1) Depicting only a scenario
2) Massive amount of assumptions to get things started and keeping things simple
3) 1 drone per warehouse which is not that realistic

2.11 Expected Results and Validation

The desired outcome is having a working software that will be able to monitor trajectories of drones. Its main feature will be: the ability to avoid collision with other drones.

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.These are further validated by comparing it with the coordinates shown on google maps.

To further illustrate what the above paragraph means referring to fig.1 flow plan. In the function that creates (x,y)positions on the arbitrary map, we validate the "correctness" by "printing" every single point and checking it that it should be in the boundary that we had set. Naturally, checking the logic of the code works as well but having a "backup plan" like outputting results ensures 100% correctness.

Similarly for the function that creates random demands on this arbitrary map, we will validate this function by copying the output data to excel and plot it as an xy plot. From there we can easily tell if this functions works the way we want by looking out for points that exceeds our pre-set map limit. Excel works well in this aspect since our limit of the map is created by deciding on four corner points within the map of ames, if there's a demand that exceeds the limit, it would be easy to tell using our naked eye.

For the distance calculation function, we used google maps as our source of validation. For e.g first we will select 2 points(with coordinates transferable to google map) pass it through our

written function and eventually returned a result(distance). This calculated distance is then compared to the distance provided in google maps. Usually for the validation process, we will pick 2 points that will provides us with a “linear line”. Since our assumption of drone pathways are a straight line, everything we do will be based on that.

Similarly for the calculation of current position function, our method of validation works the same as our distance calculation. We will select 2 points on the map, compare it with our code output and eventually locate the points on the map. This current position will have to be on the “imaginary linear line” in between”the 2 selected points. Achieving this would allow us to update drone’s latest position on the front end map in the later parts of the project.

Ensuring no collisions between drones and Satisfying demands in an efficient manner: For now, until we have successfully created our java GUI, we will test it by exporting our output code in terms of matlab compatible style coding for the purpose of testing. This algorithm works by assigning the nearest warehouse to a particular demand. Due to the implementation of this algorithm, it so happens that collision between drones will never happen. The output for example shown during our validation process through matlab looks like fig 4.

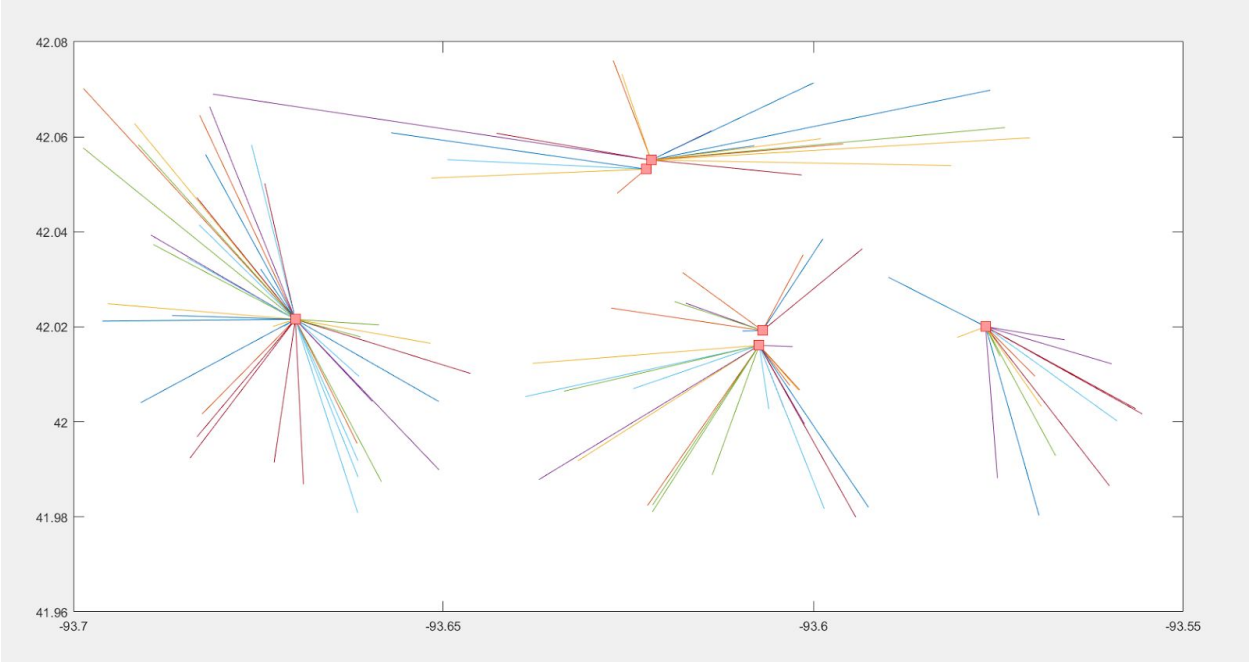

Fig. 4 Demands to warehouse

Displaying of final results to some UI: This is pretty much the last step of our software. We will integrate our backend code to work together with our front end display (GUI). There will be 3 basic things that we can see on the UI: 1) warehouse (represented as square) ,2) demand(as dot , 3) drones(as X). So as the simulation time proceed, there will be an increasing number of dots appearing on the map and "X" will decide on which demands to fulfil first base on the criterion on "avoiding collision at all cost". And so, after "X" reaches its destination, it will begin to make its way back to where it came from, then the demand that "X" had just fulfilled will be removed.When this works, we could say that our prototype is ready.

So ultimately that is basically how we would go about doing validation of our written codes.

2.12 IEEE standards

There are few IEEE standards that could possibly be related to our project but are not limited to the ones discussed below. The ones to be discuss in relation to our project are "Standard for consumer drones: Privacy and security" , "IEEE recommended practice on software reliability" and "Standard for Measures of the software aspects of dependability". A short description of the follow 3 standards are written below in its order.

1)Standard for consumer drones

This standard that focuses on consumer/commercial drones will be the authority in creating requirements, rules, systems, methods, testing and verification for consumer drones to maintain the privacy and security of people and properties within range of the drones. This standard basically ensures that privacy and security of people are not compromised.

2)IEEE recommended practice on software reliability

This standard provides a "checklist" in assessing and rating the reliability of a software based on the indicated software reliability engineering. The "checklist" involves the creation of a foundation that ensures consistency in checking for reliability of the software.

3)Standard for Measures of the software aspects of dependability

This standard provides a "standard" way to measure software dependability that will be applicable to a system's operational context. It will usually be used in conjunction in the design

phase such as: software construction and integration. The overall objective is to enable people to check if the needed level of dependability has been specified, built in and working accordingly.

The standards listed will not be unethical in any form to our context. Since of the 3 listed, 2 were basically "standard" practice in coding that institutions emphasized during their teaching process. Some of the chosen standards apply to a certain extent. For example, in the 1st standard drone' privacy, since our project involves the use of drones for delivery(although it's just a simulation) it will be inevitable(when implemented) for drones to fly over household/buildings/properties for the purpose of delivering goods. These drones might require real time feeding of coordinates to reach the end point but as a safety net, the drones will also rely on real time imagery for accuracy. Due to this privacy will inevitably be unintentionally violated.

Although our project is about stimulating the process of goods delivery from A to B, the potential problems listed earlier might/might not happen when companies decided to go for drone deliveries. Hence for the 1st standard, it applies to a small extent in our context. The remaining 2 other standards basically "works" a little closer to our project. The software that we used had their specification/technical information, functions specified, which is pretty much "completed" during their building phase. On our part, when working on our project we could reference the "way of the standard" in keeping our format of code in a systematic way. We could shadow and make use of the "rules of the standard" when integrating our codes together.

Section 3:

3.1 Other Resource Requirements

No physical parts will be involved, except for the program called Eclipse. The travelling speed of the drone will be based on the specifications given by DJI phantom 3. Occasionally, we will use Matlab for the validation of our front end display output as a temporary measure to provide us the basis of our expected output when we try to do it in Eclipse.

3.2 Financial Requirements

Not applicable to our project. Eclipse is a free software that doesn't involve any fees while matlab is provided under the school's license. At the same time, there isn't any hardware component that we will be using hence, our monetary expenditure will be 0.

3.3 Project Timeline

Week 5	Started 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 its 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 its 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 1st prototype
Week 17	finals

Sem 2	Pending tasks

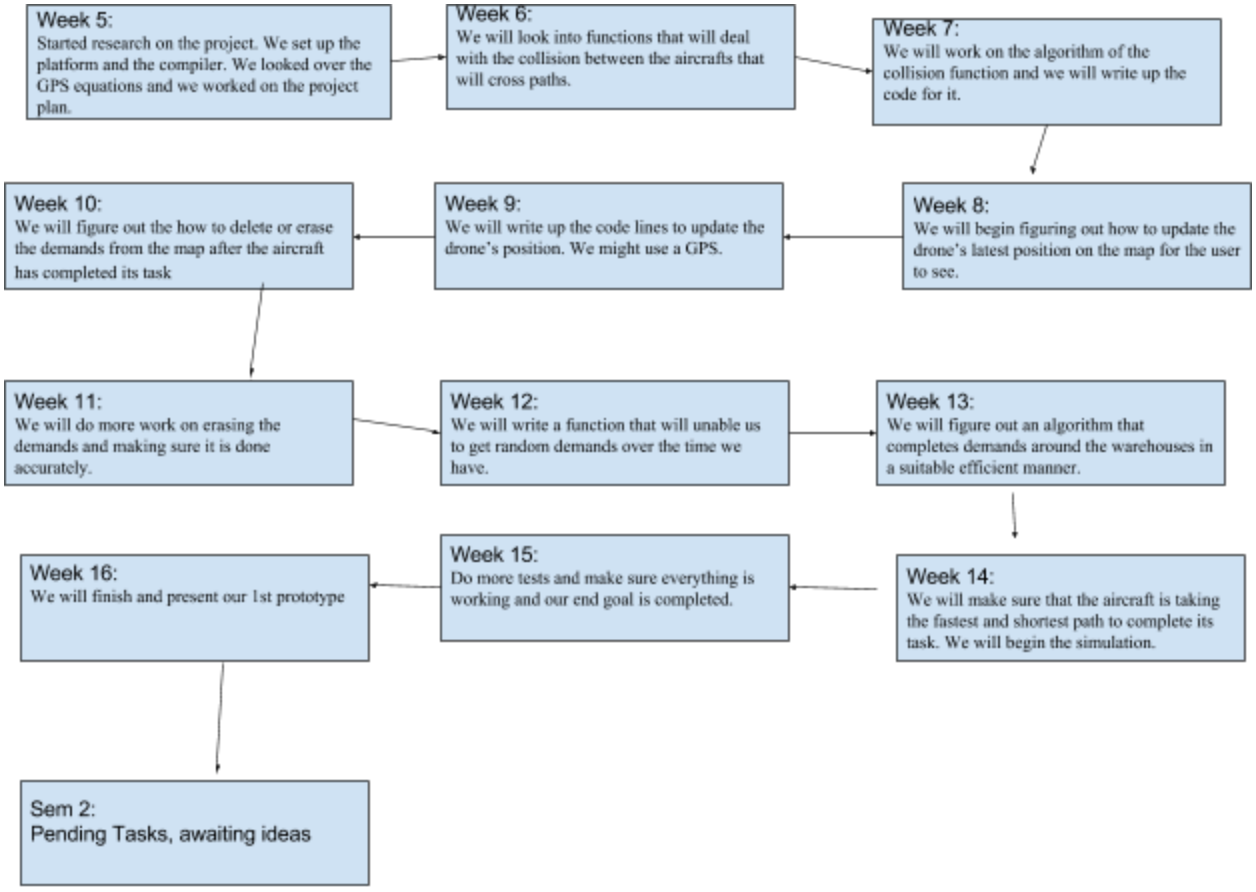

Fig. 5 Project Timeline

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 capability 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. By having the main feature in providing a collision free system simulation, companies/users can manage,plan and maximise potential profits to the things that are beneficial for them. At the same time this software could also be used as potential debate to fight for legality issues especially in the context of “drone privacy”. Hence this software will potentially serve as a stepping stone for future autonomous drone activities.

4.2 References

[1]"Stack Overflow - Where Developers Learn, Share, & Build Careers", *Stackoverflow.com*, 2017. [Online]. Available: https://stackoverflow.com/. [Accessed: 30- Oct- 2017].

[2]w. Chris Veness, "Calculate distance and bearing between two Latitude/Longitude points using haversine formula in JavaScript", *Movable-type.co.uk*, 2017. [Online]. Available: http://www.movable-type.co.uk/scripts/latlong.html. [Accessed: 30- Oct- 2017].

[3]"Build software better, together", *GitHub*, 2017. [Online]. Available: http://github.com. [Accessed: 30- Oct- 2017].

[4]P. geotools, "Plot the longitude and latitudes on map using geotools", *Gis.stackexchange.com*, 2017. [Online]. Available: https://gis.stackexchange.com/questions/178890/plot-the-longitude-and-latitudes-on-map-using-geotools. [Accessed: 30- Oct- 2017].