Drone Air Traffic Control System

Software to Coordinate Automated Drones, Safely

Research Manual

INSTITUTE *of* TECHNOLOGY CARLOW

Institiúid Teicneolaíochta Cheatharlach

Ronan Donohue - c00208501

Abstract:

With the recent proliferation of Unmanned Aerial Vehicles (UAVs) or drones in numerous sectors as diverse as national defence (Flir.eu. 2018), emergency response (SA, F. 2018), healthcare (Inc, Z. 2018), weather forecasting (Amtosoar.io 2018), waste management (Technology, R. 2018) and construction (Skycatch 2018), the skies above us are becoming crowded. It is imperative for the future of automated aviation that these UAVs, regardless of their role (governmental, commercial, or hobbyist) inhabit the same airspace peacefully. With Alphabet's Project Wing examining a form of air traffic control for controlled commercial delivery drones (Condliffe, J. 2018) and Deloitte researching the integration of 'passenger drones' to existing airspace technology (Deloitte. 2018), there seems to be a bright future for commercial drone air traffic control (DATC). But where do the hobbyist's drones find room to maneuver in these crowded skies? Drone Air Traffic Control System aims to bring control and coordination to a hobbyists fleet of drones by drawing inspiration from the Traffic Alert and Collision Avoidance System (Munoz, C., Narkawicz, A. and Chamberlain, J. 2018) found in most commercial aircraft and real air traffic control systems (Erzberger, H 2014).

Table Of Contents

Section 1 - Introduction:

There are two main aims in this research;

- To determine if it is viable to create a software application that can both schedule and route automated drones along an established drone corridor.
- To ensure a successful resolution in the event of an imminent mid-air collision between two automated drones.

In compliance with Irish government legislation, drone information must be stored so it can be produced later on request if needs be to any relevant parties. Furthermore, drone flight paths will be expected to conform to additional rules and regulations, such as flying near government buildings and national heritage sites (Irish Aviation Authority 2018).

In order to ensure maximum drone flight time and coverage, a drone corridor will need to be established. A drone corridor is very similar to a drone route, except that a corridor can facilitate multiple drones with slightly differing routes at once. A corridor should ideally be a ‘safe fly-zone’ for drones; while in the corridor, the drone will be considered minimally disruptive to everyday life that unfolds around it. In layman's terms, a drone corridor is no different than a shipping corridor; it is a safe and secure channel for the transportation of drones and their payloads, if any.

Drone corridors have been utilized in the commercial sector in the nation of Iceland to boost product delivery (Nichols, G. 2018) while in Malawi a drone corridor has been opened (UNICEF 2017) that is “*centred on Kasungu Aerodrome, ..., with a 40km radius (80km diameter) and is designed to provide a controlled platform for the private sector, universities and other partners to explore how UAVs can be used to help deliver services that will benefit communities.*” (Fabian, C. (2017).

To have a successful drone corridor in place, drones have to be able to fly through it in a coordinated fashion. This is easily done by permitting automated drones to fly through the corridor with pre-determined flight plans, as there are line-of-sight requirements for piloted drones in place in Ireland (Irish Aviation Authority 2018). A careful route-scheduling algorithm will be needed, in order to keep drone traffic controlled prior to take-off. Drones will require additional monitoring once airborne as plans are always subject to change. Weather or a faulty drone, or even a flock of birds or another automated drone can cause any number of issues with a drones flight path. While most of those incidents listed are outside the scope of an individual's control, some can be mitigated through careful design on the software end. Weather information can be obtained prior to take off and be factored into a drones flight plan. Drone information can be obtained by a system check on all moving and electrical components prior to take off to ensure the drone is fit to fly. Not much can be done to avoid a flock of birds, however a collision-avoidance algorithm that monitors each drone in the corridor and corrects them when a collision seems imminent can go some ways to ensuring that at least a drone in the corridor is safe from other drones.

Section 2 - Domain Research:

Section 2.1 - Air Traffic Control Systems:

To keep the skies safe for everyone flying with commercial airlines, there are many rules aircraft need to obey both in the air and prior to take-off. First, each aircraft pilot will need to file a flight plan. The flight plan consists of a chosen route, departure and arrival time, estimated flight time, weather conditions and aircraft info along with any proposed alterations that can be factored into the route (i.e emergency situations) (Houston, S. 2018) (Freudenrich, C. 2018). That flight plan will be fed into a scheduler, using a scheduling algorithm, most commonly First-Come-First-Served (FCFS) (Erzberger, H. 1995) in Air Traffic Control (ATC). An alternative scheduling algorithm in use is Earliest Deadline First (EDF), with studies showing a favourable number of arrivals and departures managed over FCFS (Collotta, M., Pau, G., Ticali, D. and Tirrito, S. 2015). The basic objective of the scheduler in ATC automation is to match traffic demand & airport capacity while minimizing delays (Erzberger, H. 1995). Busier airports naturally have far more arrivals and departures to coordinate, and a lot of the operations for monitoring incoming/outgoing flights, waiting aircraft and general traffic that would fall to RADAR (Radio Detection And Ranging).

Section 2.2 - TCAS:

The Traffic Alert and Collision Avoidance System (TCAS) is an automated airborne system designed to help in reducing mid-air collisions between commercial and non-commercial aircraft (Munoz, C., Narkawicz, A. and Chamberlain, J. 2018). TCAS is equipped on almost all aircraft, and is to be obeyed by pilots more so than air traffic control suggestions (Munoz, C., Narkawicz, A. and Chamberlain, J. 2018).

TCAS works by sending interrogations to other aircrafts transponders. The transponder will respond to the interrogation in a similar way to how it responds to radar pings. From the time difference between the interrogation and the reply, the distance to the other aircraft is calculated. Subsequent distance interrogations will take place to determine a trend in the oncoming aircrafts behaviour. Approximate calculations are performed to determine the time of closest point of approach (Munoz, C., Narkawicz, A. and Chamberlain, J. 2018). If the result of these calculations is below a certain threshold, a Traffic Alert (TA) is raised. Should the result of those calculations fall even lower, a Resolution Advisory (RA) is given.

The difference between a TA and an RA is that a TA is an indication of possible imminent danger; an RA is a direct instruction that must be obeyed to avoid a possible collision. In issuing an RA, the altitude and magnitude of the aircraft are taken into consideration, along with the crew and passengers aboard to ensure minimal disruption to the original flight plan. When an

RA is issued, it is issued to the other aircraft also. In the rare case that both aircraft issue an RA at the same time, the aircraft with the higher 'Mode S' address (in other words, priority) will concede its RA should it conflict with the other aircrafts. Fig 1 shows RA and TA protected space in front of an example aircraft.

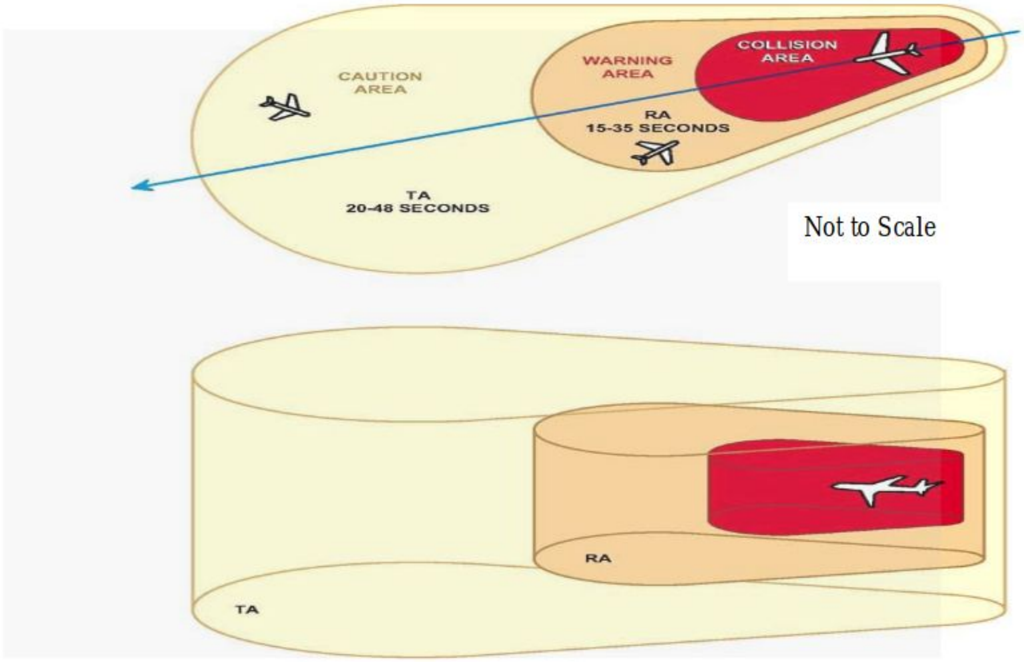

Fig 1 - A visualisation showing the difference between TAs and RAs in terms of volume around an aircraft. (Introduction to TCAS II. 2011)

Section 2.3 - Irish Drone Regulations:

All drone operators in Ireland must comply with the Irish Aviation Authorities regulations (Irish Aviation Authority 2018) alongside new GDPR requirements becoming enforceable in late May 2018 (Data Protection Commission 2018). This raises some issues regarding drone fly zones and data retention, data processing and data security for data captured by drones namely that "*...the dates and times of the flights, the flight path and the types of personal data (e.g. imagery, radio, geometry, location etc) that may be collected should accurately be described, along with the contact details of the operator and the data controller"* (DPC IAA 2018). Also, under Section 4 of the Data Protection Act, "*any person whose image is recorded on a drone system has a right to seek and be supplied with a copy of their own personal data from the footage"* (DPC IAA 2018). The main takeaway from this legislation is that <u>all</u> data pertaining from an automated drone flight should be stored and carefully safeguarded. Once stored away safely, it can be presented to any member of the public at their request, if relevant to that person.

Section 3 - Proposed Solution:

For this software to be successful, there are two main problems that require a solution going forward; the question of routing automated drones and preventing them from colliding mid-air. But before that again, there are certain 'soft' roles the software promises to fulfill to the end user. These are that drone flight paths are automated and that drone flight will be visualised. This information should inform any solution undertaken. WIth this in mind, there are two outlined solutions below:

Section 3.1 - Routing Algorithm:

While most hobbyist drones aren't in possession of radar technology, an efficient scheduling algorithm that generates a specialized and adaptable drone flight plan can ensure safe fly time for drones. One approach that may yield some positive results is to ensure that each drone flight plan mirrors an aircraft's flight plan;

- All information about the drone is taken into consideration.
- Each route must be designed to keep the drone at *X* distance from other drones at all times.
- No drone can be considered for flight unless all safety checks and hardware checks are passed.

A Scheduler is a proposed entity within the software application that will handle the creation of routes for drones. The Scheduler will be fed drone information along with the proposed flight plan. From this data, the Scheduler should;

1. Take into account the drone routes that will be active when planning a route for a certain time. It can access this information from a list of accepted flight plans.
2. Ensure that each drone route does not interlace with a planned route for the same time.
3. Ensure that each drone is fit for flight.
4. Possibly update the original flight plan as new flight plans are entered into the scheduler.

A drone route should be repeatable. Either at a certain time, or as a job that is given to a different drone in case of malfunction in the chosen drone.

Once airborne, drones may not be capable of storing all the necessary route information that they need. A Controller entity will be needed to update each drone with correct route data, as drones maybe only able to receive commands, not store and execute a series of commands (Parrot 2016).

Section 3.2 - Collision Avoidance:

One approach going forward is to have the software perform the role of an air traffic controller and a TCAS algorithm at the same time. As an air traffic controller examines aircraft coming and going from their airport, the software will monitor the drones that are airborne. It will do this by using the telemetry data that airborne drones will be transmitting back to the application. Using this data, active drones and their routes can easily be mapped to a 2-D screen, which will show both the drone, it's route and a cone facing outwards with an orange shell and a red shell, as depicted in Fig 2. Each drones shell will be calculated from the drone telemetry data and serve as a wrapper around the drone.

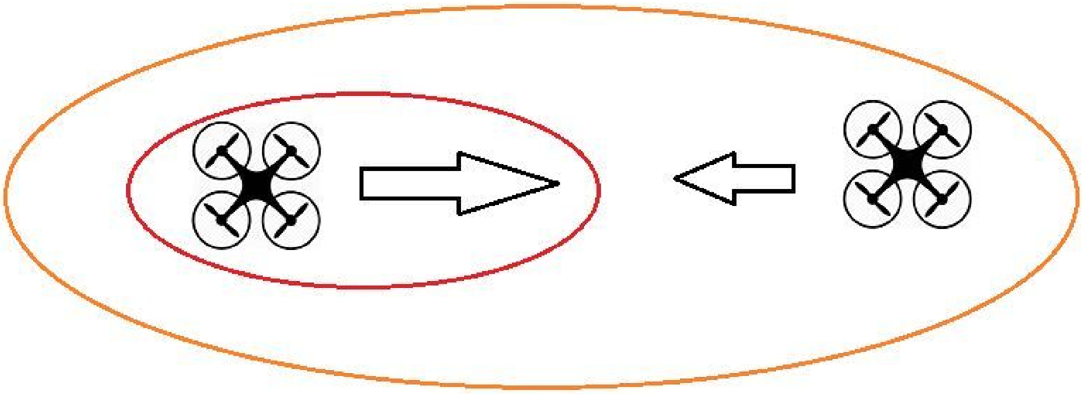

Fig 2 - A proposed display option for two oncoming drones. Not to scale.

In Fig 2, Drone A is moving towards the right, while Drone B is in Drone A's orange shell. Let us assume that Drone B has just moved into Drone A's orange shell. As both drones and their shells are being rendered by the software application, once a drone has entered another drones orange shell, a Collision Avoidance (CA) event will occur. For this to be successful, each drone will need to pause their automated flight. The details of each drones original flight will need to be stored temporarily, possibly via a push to a stack for easy recall. The Controller entity will relinquish control to a CA event handler. The CA event handler will take over instruction of both drones, determine the nature of the imminent collision i.e. in each others orange zones or red zones, and act accordingly.

If both drones are in each others orange zone, then the CA handler can, time permitting, alter a single drones route. It will do this by either dismissing the original route and causing the drone to hover until the other drone has passed or by quickly re-uploading a new route that is based off of the original route that moves the drone out of harm's way and lets it continue flying. The drone with the most battery life will be the drone chosen to alter its route. This information is obtained via drone telemetry data as outlined in Section 4.3 of this document.

If both drones are in the red shell zone, then both drones will be instructed to halt with their routes cancelled temporarily. A new route will be plotted for them that takes them safely around one another, with the easier flight path going to the drone with shorter battery life. Once completed, each drones original route will be re-uploaded to them and they can resume their route.

With both a working routing algorithm and a collision avoidance algorithm in place, the rest of the software can be built around representing this functionality to the user.

Section 4 - Technology Research:

Section 4.1 - Platform Options:

There will likely be a need for a large amount of processing power in this application, coupled with a more comprehensive UI and UX, which may prove ideal for laptop or desktop development. As it stands as of Oct 2018, the current market share for desktop operating systems in Ireland is as follows;

- Windows 73.92%
- OSX 16.78%
- Linux 2.46%
- Chrome OS at 1.03%
- All others 5.8% (StatsCounter 2018)

It may be prudent to develop for a Windows PC first and foremost, if going the desktop route, to ensure a wide swathe of the market is being targeted. It is worth noting that the use of Java as a development language may make the choice of desktop environment redundant. Java's 'Write Once, Run Anywhere' aspect (Langley N. 2002),enforced by its use of interpretation via the Java Virtual Machine (JVM) rather than compilation, promises cross-platform applications with very little need to 'tweak' the source code for each OS.

Another reason for Java comes from its status as the main language used to develop apps for Android (Ipl, S., 2018). In 2017, Android had over 64% of the mobile market share as compared to the 32% sharing of iOS. Despite having over half the market share, Apple's App Store has reported earnings of up to $5.4 billion compared to $3.3 billion for Google Play (Ipl. S., 2018). The main language for iOS development would be Objective-C, which is featured in Section 4.3 of this document along with Java. The main benefit of developing for a mobile app would be development speed, along with capabilities like 4G for such an outdoor pursuit as drone piloting. A potential drawback is the lack of screen real estate, which can limit the development of a rich user interface.

C++ is also a likely candidate for a desktop development language, although extra care will be needed during development to ensure a smooth port between Windows and Unix-based OS's. The main advantages of C++ in this case would be the additional flexibility and computational power the developer would have in areas like memory management, graphics programming, and mathematical calculations due to the fact that C++ is a compiled language while Java needs to convert instructions to bytecode to be run by the JVM. Given the nature of the software solution being researched, the extra speed available in C++ may prove beneficial down the line. That being said, the trade off for computational speed vs. lost development time solving bugs that would not be present in Java (i.e memory allocation for objects) is worth considering.

A suitable cross platform GUI library for Java would be JavaFX, and for C++, Qt (pronounced ‘cute’ (Qt 2018)).

Section 4.2 - Data Storage Options:

As Irish drone legislation mandates that all data pertaining from an automated drone flight should be stored and carefully safeguarded, it seems prudent that a database should be utilized in storing drone flight information. The Data Protection Act states that drone data “shall not be kept for longer than is necessary for” the purposes for which they were obtained (DPC IAA 2018). With over 8,000 drones registered in Ireland in 2017 (Fegan, J. 2017), it may prove useful to provide an efficient means of storing data.
A relational database is a database that utilizes a relational model of data, where data is grouped by its relationship to other data (E. Codd 1981). Some examples include mySQL or Oracle, both of which are more than capable of handling the potential volume of data from this project.

An alternative to a relational database is a NoSQL database, an example being MongoDB. ‘NoSQL’ stands for ‘Not Only SQL’ and is a relatively new type of database option that are designed to cope with the scale and agility challenges that face modern applications (MongoDB 2018). They are built to take advantage of the commodity storage and processing power available today (MongoDB 2018).

Drones are capable of taking photographs and capturing video, things a relational database cannot store very well. Similarly, NoSQL databases like MongoDB have a 16mb cap on data submitted for storage (MongoDB). But using technology like GridFS, a convention within MongoDB, the cap can be by-passed and larger data can be stored.

Failing that, local storage on a desktop, laptop or portable hard drive may be the only solution for drone image data storage. A proposed solution would be that for each drone flight that captures images and video, the full path to the data is stored as a string alongside the flight info in the database i.e “C:/users/John Everyman/Drone Data” for Windows.

Section 4.3 Parrot SDK:

The Parrot SDK is the official software development kit for most Parrot drones (Parrot 2016). The SDK comes in 3 languages; Java, C and Objective-C (Parrot 2016). The SDK was investigated primarily to examine the kind of events and commands that can be expected to be sent and received by the drone. The drone is able to send information such as its latitude, longitude, altitude and its X,Y,Z dimension speeds. Also, the drone can send information about its flying state such as if it is currently landed, landing, taking off, hovering, flying, in an emergency state or if its battery is low, critical or OK.

Section 5 - Summary:

Going forward, it is imperative that whatever software solution is implemented be able to handle the role of both an Air Traffic Controller and a TCAS transponder. As outlined above, a software based approximation of either an existing Air Traffic Control scheduling system or a TCAS-based early warning system is certainly feasible and in unison should prove more than capable of shepherding drones in real time. There are other responsibilities the software must provide; It must ensure that drones can be automated. It must ensure that data is stored reliably and securely to be in keeping with current legislation. Most import of all, it must read drone telemetry data, process the data quickly and make the correct decisions needed to preserve both the drone and the integrity of its original flight plan. With those features successfully implemented, there is potential here for a drone hobbyist to coordinate a fleet to perform mundane tasks or complex maneuvers with ease.

Section 6 - Glossary:

Air Traffic Control - The people and equipment that control and monitor air traffic within a particular area. Usually at airports and military bases.

Collision Avoidance - A term that denotes the prevention of a possible collision.

Drone - A Unmanned Aerial Vehicle or UAV. Not piloted, but controlled.

Drone Corridor - Similar to a shipping corridor, but applied to drones. An established area where drones fly safely.

Earliest Deadline First - The item with the nearest approaching deadline will be selected.

First-Come-First-Served - The first item ready for selection will be selected.

Flight Plan - A comprehensive overview of the aircraft, its route information, passenger information, results of safety checks etc. Fed into Air Traffic Control schedulers prior to departure.

Route - A planned journey, in this case undertaken by drone.

Scheduler - Air Traffic Controls equipment to ensure flight plans are coordinated. Utilizes a variety of scheduling algorithms depending on implementation.

TCAS - Traffic Alert and Collision Avoidance System. An automated airborne system designed to help in reducing mid-air collisions between commercial and non-commercial aircraft.

Section 7 - Bibliography:

Atmosoar.io. (2018). *Atmosoar - Atmospheric Soaring Technologies*. [online] Available at: http://www.atmosoar.io/ [Accessed 31 Oct. 2018].

Condliffe, J. (2018). *Alphabet's new air traffic control system steers drones away from peril*. [online] MIT Technology Review. Available at: https://www.technologyreview.com/s/608050/alphabets-new-air-traffic-control-system-steers-drones-away-from-peril/ [Accessed 31 Oct. 2018].

Flir.eu. (2018). Drone Thermal Imaging Solutions | FLIR Systems. [online] Available at: https://www.flir.eu/suas/ [Accessed 31 Oct. 2018].

Irish Aviation Authority. (2018). *Drones*. [online] Available at: https://www.iaa.ie/general-aviation/drones [Accessed 31 Oct. 2018].

DPC IAA (2018). *Guidance on the use of Drone Aircraft - Data Protection Commission - Ireland*. [online] Available at: https://www.dataprotection.ie/docs/Guidance-on-the-use-of-Drone-Aircraft/1510.htm [Accessed 1 Nov. 2018].

Erzberger, H. (1995). *Design Principles and Algorithms for Automated Air Traffic Management*. [online] Pdfs.semanticscholar.org. Available at: https://pdfs.semanticscholar.org/dca9/a1164666ee2b6d7c28cc55f50de6093e99b2.pdf [Accessed 1 Nov. 2018].

Munoz, C., Narkawicz, A. and Chamberlain, J. (2018). *A TCAS-II Resolution Advisory Detection Algorithm*. [online] Shemesh.larc.nasa.gov. Available at: https://shemesh.larc.nasa.gov/people/cam/publications/gnc2013-draft.pdf [Accessed 31 Oct. 2018].

SA, F. (2018). *Flyability — Safe Drones for Inaccessible Places*. [online] Flyability.com. Available at: https://www.flyability.com/ [Accessed 31 Oct. 2018].

Inc, Z. (2018). *Zipline — Lifesaving Deliveries by Drone*. [online] Flyzipline.com. Available at: http://www.flyzipline.com/ [Accessed 31 Oct. 2018].

Deloitte. (2018). *Managing The Evolving Skies*. [online] Available at: https://www2.deloitte.com/content/dam/Deloitte/global/Images/infographics/gx-eri-managing-the-evolving-skies.pdf [Accessed 31 Oct. 2018].

Skycatch. (2018). *Skycatch: Drone Image Processing Platform*. [online] Available at: https://www.skycatch.com/ [Accessed 31 Oct. 2018].

Technology, R. (2018). *Plastic, waste, survey, data | Aquadrones | RanMarine Technology*. [online] Plastic, waste, survey, data | Aquadrones | RanMarine Technology. Available at: https://www.ranmarine.io/ [Accessed 31 Oct. 2018].

Parrot. (2016). *Parrot for Developers*. [online] Available at: https://developer.parrot.com/ [Accessed 31 Oct. 2018].

Data Protection Commission. (2018). *GDPR - Data Protection Commission - Ireland*. [online] Available at: https://www.dataprotection.ie/docs/GDPR/1623.htm [Accessed 31 Oct. 2018].

Freudenrich, C. (2018). *How Air Traffic Control Works*. [online] HowStuffWorks. Available at: https://science.howstuffworks.com/transport/flight/modern/air-traffic-control.htm [Accessed 1 Nov. 2018].

Houston, S. (2018). *9 Steps to Help You Plan a Cross Country Flight*. [online] The Balance Careers. Available at: https://www.thebalancecareers.com/how-to-plan-a-vfr-cross-country-flight-282900 [Accessed 1 Nov. 2018].

Nichols, G. (2018). *Iceland's answer to Amazon adds drone routes to urban deliveries | ZDNet*. [online] ZDNet. Available at: https://www.zdnet.com/article/icelands-answer-to-amazon-adds-drone-routes-to-urban-deliveries/ [Accessed 1 Nov. 2018].

StatCounter. (2018). *Desktop Operating System Market Share Ireland | StatCounter Global Stats*. [online] Available at: http://gs.statcounter.com/os-market-share/desktop/ireland/2016 [Accessed 1 Nov. 2018].

Qt. (2018). *Qt - Company | About us*. [online] Available at: https://web.archive.org/web/20170222172844/https://www.qt.io/about-us/ [Accessed 1 Nov. 2018].

Fegan, J. (2017). *'Drone users need to know the law', warns Irish Aviation Authority*. [online] Irishexaminer.com. Available at: https://www.irishexaminer.com/ireland/drone-users-need-to-know-the-law-warns-irish-aviation-authority-463063.html [Accessed 1 Nov. 2018].

Fabian, C. (2017). *Malawi's Unique Drone Corridor – Stories of UNICEF Innovation*. [online] Unicefstories.org. Available at: http://unicefstories.org/2017/07/03/malawis-unique-drone-corridor/ [Accessed 1 Nov. 2018].

UNICEF. (2017). *Africa's first humanitarian drone testing corridor launched in Malawi by Government and UNICEF – Stories of UNICEF Innovation*. [online] Available at: http://unicefstories.org/2017/06/29/africas-first-humanitarian-drone-testing-corridor-launched-in-malawi-by-government-and-unicef/ [Accessed 1 Nov. 2018].

Collotta, M., Pau, G., Ticali, D. and Tirrito, S. (2015). *A comparison among scheduling algorithms in order to improve the air traffic control*. Ph.D. Kore University of Enna, Cittadella Universitaria.

Introduction to TCAS II. (2011). 7th ed. [ebook] U.S Department Of Transportation, pp.1 - 50. Available at: https://www.faa.gov/documentLibrary/media/Advisory_Circular/TCAS II V7.1 Intro booklet.pdf [Accessed 8 Nov. 2018].

Langley, N. (2002). *Write once, run anywhere?*. [online] ComputerWeekly.com. Available at: https://www.computerweekly.com/feature/Write-once-run-anywhere [Accessed 8 Nov. 2018].

Codd, E. (1981). *Relational Database: A Practical Foundation for Productivity*. Fellow. IBM San Jose Research Laboratory.

MongoDB. (2018). *NoSQL Databases Explained*. [online] Available at: https://www.mongodb.com/nosql-explained [Accessed 8 Nov. 2018].

ipl, S., 2018. *Android Vs. iOS Development – Which Platform is better and Why?*. [Online] Available at: https://blog.sagipl.com/android-vs-ios-development/ [Accessed 30 October 2018].