Imperial College London

# Detection of Civil Unmanned Aerial Vehicles by Sound Processing

## EE2-PRJ E2 Project Final Report


| | | |
|---|---|---|
| Group Number: | 11 | |
| Team Members: | Zhi Chua | (00950919) |
| | Guy Haroush | (00818520) |
| | Chi Leung | (00978321) |
| | Aaditya Malhotra | (00975663) |
| | Pavol Olexa | (00985071) |
| | Alexander Wilson | (00962745) |
| | Yangshuowen Zhao | (00921627) |
| Project Supervisor: | Mr Mike Brookes | |
| Submission Date: | 13 March 2016 | |

# Table of Contents

# 1 ABSTRACT

The purpose of this project was to develop an economical, precise, reliable and legal detection system to locate and alert an individual of the position of a nearby drone. The need for drone detection systems has risen steadily, with growing cases of drone misuse despite regulation to stop invasive and malicious drone use in restricted areas. Following from the interim report, the team decided to pursue concept two out of the three proposed for further development. These three all involved using sound processing of the drone's noise as the principle method for detection. This led to the following five separate modules being developed: a sound detection, signal conditioning, data collection and transmission, processing and output display module. Each of these modules were developed and improved before a complete integration to form a complete, working prototype. Testing demonstrated that the prototype performed accurately up to a distance of 10 m, while only having a small error in both the azimuth and elevation angle of 2.22° and 0.7238° respectively. The prototype was also cost efficient, with the manufacturing valued at £12.80 while the nearest competitor's drone detection system stands at approximately £50. Therefore, the prototype ultimately met the Design Criteria, as it proved to be a reliable, economical and a high performance device.

# 2 INTRODUCTION & BACKGROUND

Unmanned Aerial Vehicles (UAVs), commonly referred to as drones, are quickly becoming a staple of modern society. Spurred by a long list of clients seeking out the technology, several forecasters have suggested that the industry will exceed a market value of £8 billion by 2020 (Camhi, 2016). The diverse range of uses for the technology is the principal reason for the mounting popularity, with applications ranging from commercial, military and recreational use. Many novel applications for drones have arisen as well, an example being purpose built drones for shark spotting in Australia (The Sydney Morning Herald, 2016). However, although the drone has proved itself as an invaluable tool for both increased efficiency and safety, it has also demonstrated its ability to become a significant and global security issue.

The issue has exhibited itself in many forms, so much so that leading think tank *Open Briefing* reported that they believe UAVs, and their advancement into a sophisticated and versatile technology, present the greatest threat to society due to their widespread availability and capabilities (Open Briefing, 2016). This assertion is beginning to be justified worldwide, with misuse of the technology rising. In England alone, drone related calls to the police have grown by over 2000% over the past three years (Sky News, 2015).

Prominent examples of drone abuse stem from attempts by criminals to circumvent existing defense systems in secure environments. Numerous police reports have come to light illustrating drones used to try and sneak contraband into prisons, such as a recent case in a high security prison in Manchester, England (BBC, 2015). Drones have also been increasingly used to enter restricted airspace, such as airports, with the US Federal Aviation Administration (FAA) reporting that drone sightings by pilots have trebled since 2014 (BBC, 2015). Even more recently, the National Football League (NFL) security and organizers released a statement ahead of the most recent Super Bowl that they have been operating at a higher level of security than during the September 11$^{th}$ terrorist attacks, due to potential drone threats (IB Times, 2016).

Many state that current bids to mitigate these issues via new legislation have so far fallen short (Open Briefing, 2016). With drone misuse only continuing to grow, it is argued that lawmakers are unable to keep up with the approximately 200,000 drones that are being bought every month worldwide for commercial use (Open Briefing, 2016). Therefore, a call has been made to develop more active countermeasures, to defend civilians against the current imposing threat. This call in turn inspired this project, with the goal of developing a system to deter future incidents by locating and alerting users of a drone mid-flight, eradicating the threat before it is in a position to cause harm.

This report will provide a brief explanation of the concept selection process for the proposed device (brief because it was covered in detail in the interim report) as well as a comprehensive analysis of the technical design of the finalised prototype.

# 3 MARKET RESEARCH

Market research was conducted to provide the group with pertinent information about existing drone detection methods already in circulation. This was deemed necessary as this information allowed informed design choices to be made as the group pursued the development of an innovative and commercially attractive solution to drone detection.

Extensive research revealed that current industry leaders in drone detection usually have devices which use audio sensors as the principal method for detection, with four competitors employing the technology (Anon., 2015). However, those companies using audio sensors only alert the user that a drone is nearby, rather than providing the exact location. There are other, albeit fewer, cited solutions that are able to give the direction of an incoming drone; these include thermal, video, radar, and radio frequency detection (Anon., 2015). Furthermore, it was found that the range of products varied in their cost, depending on the technology implemented, with one of the most affordable solutions standing at a price of approximately £50 (U.S. News, 2013).

Having discovered that sound processing is the most popular detection method for commercially available devices, an investigation was carried out to determine why this is the case. It was found that audio sensors are usually used as in most scenarios it provides a cost efficient and reliable solution to the problem. This is in contrast to other possible solutions which commonly exhibit inaccurate detection or high operation costs. For example, systems using radar are often extremely unaffordable, even though they produce consistently accurate results, rendering it an unattractive product for consumers. Furthermore, thermal processing is generally inadequate due to its inability to detect small, electric powered plastic drones, as they do not generate a sufficient amount of heat (Help Net Security, 2015). This is a similar issue to that found in image processing techniques, such as video surveillance, which are known to frequently yield false alarms as they implement algorithms which are unable to distinguish drones from birds (Help Net Security, 2015).

# 4 DESIGN CRITERIA

An exhaustive Design Criteria, based on the Product Design Specification (PDS), was designed to ensure that any device developed would be optimal given the social, economic and environmental context. Due to the fact that the Design Criteria was explained in full in the previous interim report, it will only be referenced here. For the complete Design Criteria and PDS refer to Appendix 1 and Appendix 2 respectively.

# 5 CONCEPT DESIGNS CONSIDERED

Before any concepts were conceived, different possible methods of detection were presented and weighed against their merits and drawbacks. Electromagnetic systems involving both active and passive detection mechanisms were proposed, such as radar and image processing. However, they were removed from consideration due to infringements to the Design Criteria in aspects such as cost and reliability (refer to Market Research of the report for more details). Ultimately, it was decided that a system involving sound processing would be optimal for the project's goal. This is due to the favorable amalgamation of low cost and high performance (ability to differentiate a drone from other sound sources) such a system would exhibit.

Therefore, it was agreed that any system designed would function by first receiving the sound emitted by a quadcopter, followed by a module that processes that sound in such a manner that would provide information regarding the instantaneous location of the device. To achieve this, the device was divided into five separate modules:

1. Sound Detection
2. Signal Conditioning
3. Data Collection and Transmission
4. Processing
5. Output to User

For the project, three designs were considered for further development. These three concepts fell within the same high level design proposed above, with differences only within the implementation of the sound detection and processing modules.

## 5.1 Concept 1: Spinning Microphone

The first concept recommends that a single unidirectional (narrow beam) spinning microphone be used to detect sound from an incoming drone. The microphone will spin in such a manner that it will periodically scan every position in the sky, constantly searching for a drone's sound signature.

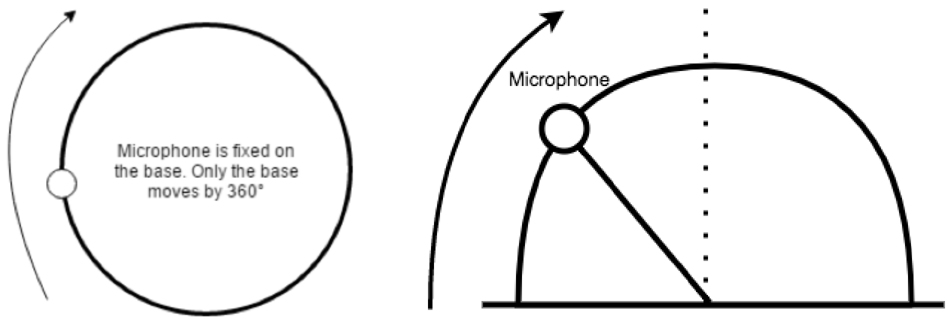



Figure 1 *Top view (left) and side view (right) of proposed layout*

The presence of a drone is detected when the microphone receives the frequency signature of a drone noise for a power level greater than a preset threshold. The position of the microphone at that instance of time returns the azimuth and elevation angles for which the drone presence was detected. The microphone then continues to spin for every azimuth – elevation angle pair to update its results on the position of the drone, or for new drone presences.

## 5.2 Concept 2: Planar Microphone Array

For the second concept, the difference in the sound detection module compared to the first option is that an array of eight static microphones will be used in contrast to one spinning microphone. Three is the minimum required to determine a sound's direction, but with more microphones, sensitivity and noise rejection improves. Therefore, taking into consideration the tradeoff between performance, cost and ease of design, as well as limitations imposed by other modules (i.e. limited ADC sampling rate and processing speed), eight microphones in an octagonal configuration was chosen. This layout provides a sufficient number of microphones for high performance, while meeting any external constraints.

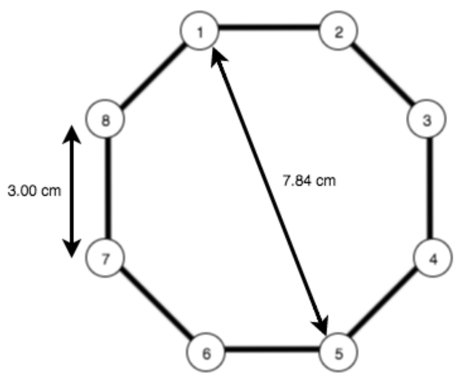



Figure 2 *Top view of planar microphone array layout*

For processing the sound, an algorithm called delay and sum beamforming is used for detection. This method exploits the fact that each microphone will record the sound signals from a drone but with a delay that corresponds to the position of the drone. The procedure delays and sums the outputs from the array of microphones in a manner such that sound arriving from a particular direction is amplified while sound from other directions are rejected (Mucci, 1984). By doing so for different delays, the device can scan the sky, finding the sound direction(s) where drone noise is the most prevalent and consequently the direction of the incoming drone(s).

### 5.3 Concept 3: 3D Microphone Array

The third concept has a similar sound detection module to the second concept but is distinct in that its microphone array is not in a fully planar configuration. One extra microphone (nine total) is placed on a separate plane above the other eight microphones to produce a 3D microphone array instead.

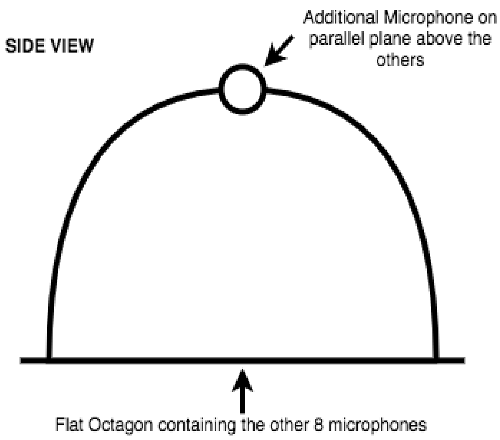



Figure 3 *Side view of 3D microphone array layout*

The algorithm used in the processing module is based on determining which two microphones have the greatest time delay between them in recording the incoming sound signal from a drone. Once the two microphones are determined, the direction of the drone can be deduced as the straight line pointing outwards in the direction of the microphone that received the sound signal first. This is first done between the eight microphones on the base of the array, determining the azimuth angle, and then between each microphone on the base and the extra microphone above, determining the elevation angle.

## 6 Discussion and Concept Selection

In order to choose between the three concepts, the advantages and disadvantages of each concept were discussed against the Design Criteria. In addition, a Concept Selection Matrix was compiled, where the marks were assigned according to each concept's capacity to adhere to each of the five categories in the Design Criteria. As a result, Concept 2 was chosen for further development. The full discussion is only referenced here as it was covered in the interim report. For a full discussion of each concepts advantages and disadvantages, refer to Appendix 3.

Table 1 *Concept Selection Matrix – Marks given on a scale from 1 (worst) to 5 (best)*

| Criteria | Option 1 | Option 2 | Option 3 |
|---|---|---|---|
| **Economics** | 3 | 4 | 2 |
| **Performance - Speed** | 1 | 5 | 3 |
| **Performance - Resolution** | 3 | 5 | 3 |
| **Ethics and Safety** | 5 | 5 | 5 |
| **Reliability and Repeatability** | 1 | 5 | 3 |
| **Maintenance and Construction** | 1 | 5 | 3 |
| **Simple to Design, Analyse, and Debug** | 5 | 1 | 3 |
| **Total Marks** | 19 | 25 | 22 |

# 7 Design Changes from Interim Report

One major design change that took place after the interim report was that cross-correlation would be used instead of sum and delay beamforming in the processing module to determine the delay between signals. This decision stemmed from inadequacies in the beamforming method led by the fact that for consistent accurate results to be produced, the number of microphones would have to be greatly increased. However, doing so would have infringed on the Design Criteria as the cost of the resulting array would be large. Furthermore, barring any significant optimizations on the algorithm implemented in the processing module, beamforming with more microphones would have made the computation time too large to generate timely results. Therefore, an algorithm implementing cross-correlation was adopted instead, in order to keep computation time and cost within the restrictions outlined in the Design Criteria.

# 8 Technical Analysis and Understanding

Having finalized the high-level design of the proposed device, the group proceeded to design the low-level aspects of each module and build a working prototype. This section depicts the design of each of the five modules that compose the prototype, illustrating the functionality of the device. The block diagram below shows the modular breakdown.

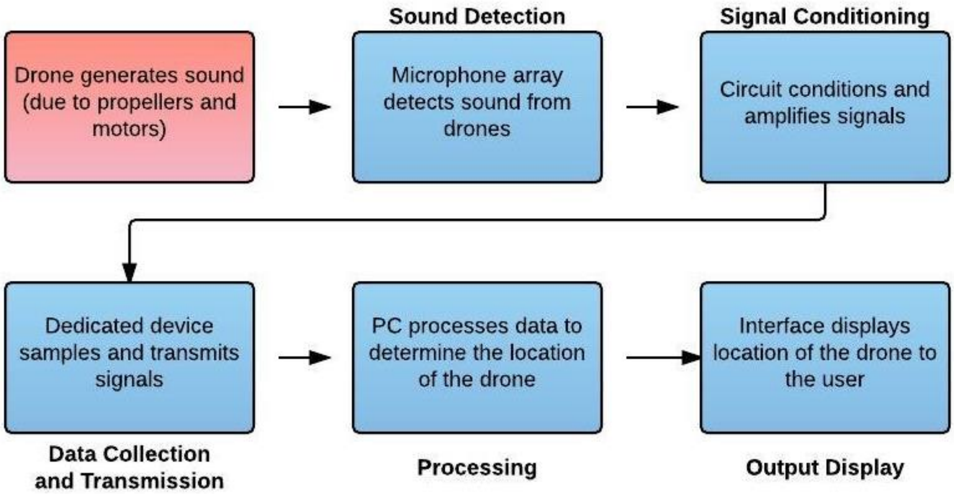



Figure 4 *High level block diagram*

By integrating these modules, a prototype was built which successfully detects and displays to the user the direction of an incoming drone. The next section of the report fully details the operation and implementation of each module. However, before elucidating the functionality of each module, the assumptions and constraints imposed must be made clear.

## 8.1 PRE-DESIGN TESTING AND ASSUMPTIONS

Four important assumptions and design constrains were made before building the prototype. The first assumption was that an incoming drone will always be at a higher altitude than the device. Another assumption was that the drone will always be sufficiently far such that the sound waves it generates can be approximated to be planar at the device's location (far-field approximation) (G.Kino, 2000). In addition, it was assumed that only one drone would be in the proximity of the device at any given time. This was done in an attempt to downscale the design challenge to a level that was feasible due to time constraints.

The final assumption that was made was concerning the frequencies the device should operate at. To reduce interfering background noise, it was proposed that the system filter noise outside the expected frequency range. In order to determine this range, frequency spectrum analysis of drone sound samples was carried out on *MATLAB* (refer to Appendix 4.3 for the MATLAB code used).

Five sound recordings of a variety of drones were used to analyze the frequency spectrum, consistently resulting in clusters of peaks being found at approximately 1000 Hz. Therefore, the frequency range of operation was decided to be from 500 Hz – 1500 Hz, this was done to accommodate the Design Criteria which recommend the device to be operational over a 1000 Hz range, allowing different drones, of a variety of sizes, to be detected.

## 8.2 SOUND DETECTION MODULE

The sound detection module, consisting of the microphone array, is the first module in the device. Its principle operation is to receive and record sound from an incoming drone. Processing requires that constraints are imposed on the microphone layout, furthermore, an optimal layout will enhance the performance of the device.

### 8.2.1 Microphone Array Layout

Initially, an octagonal layout with eight microphones lying parallel to the horizon was proposed (as depicted in the Concept 2 description). This was later down-scaled to four microphones in an attempt to increase feasibility and manageability of the project goal due to time constraints. The group proceeded with four planar microphones, but following testing, it became evident that this layout led to errors when the drone is near 90º in elevation from the device. As the calculations would be based on far-field approximations, the delays between microphones would be very small at high elevation angles (75º to 90º), rendering it difficult to produce accurate results. Hence, a tetrahedral layout was adopted instead, allowing for one of the microphones to be placed above the plane formed by the other three microphones. This was decided following the discovery of research suggesting that a tetrahedron is the optimal 3D layout for four microphones (B Yang, 2005). This layout significantly improved the accuracy of the elevation angle calculated, as the time delays between the top microphone and the remaining three were sufficiently large.

### 8.2.2 Microphone Array Dimensions

A dimension restriction exists such that the maximum distance between any two microphones needs to be less than half the wavelength of the maximum frequency used in processing, in the case of this device 1500 Hz (chosen during pre-design testing).

Such a restriction is required when the signal is periodic. The phase difference between a signal received by microphone 1, and another signal received by microphone 2 at $t_2 + kT$ will be the same for any $k \in \mathbb{Z}$, where $T$ is the period of the signal. Meaning each phase difference does not have a unique solution.

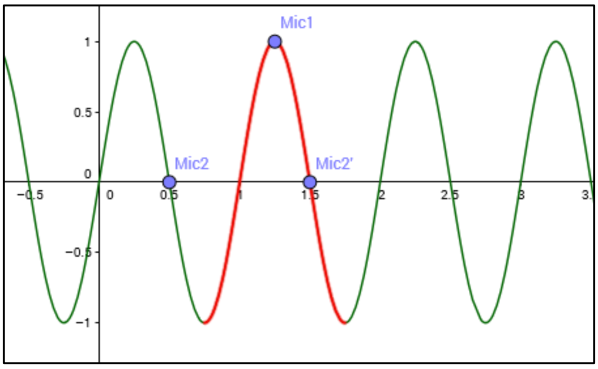



Figure 5 *Diagram representing two possible time positions given a specific phase difference*

As demonstrated in the above example, when a sinusoidal signal of frequency 1 Hz is received, and two samples are taken by microphone 2, at $t = 0.5$ s and $t = 1.5$ s, microphone 1 will not be able to distinguish between the two samples because they have the same relative phase difference. This illustrates how a sample taken by microphone 2 can be mistakenly interpreted as leading the sample of microphone 1, when it is actually lagging it. Hence, by imposing a physical restriction, microphone 2's signal can only lie in the red region ($t_1 \pm T/2$), removing any ambiguity (each phase difference will have a unique solution).

Therefore, by considering the maximum concerned frequency, the maximum distance should be:

$$\frac{0.5 \times \text{Speed of Sound}}{\text{Maximum Frequency}} = \frac{0.5 \times 340 \text{ ms}^{-1}}{1500 \text{ Hz}} = 0.113 \text{ m} = 11.3 \text{ cm}$$

Below are two diagrams illustrating the microphone layout and dimensions with microphone 1 chosen to be the reference microphone (placed at the origin).

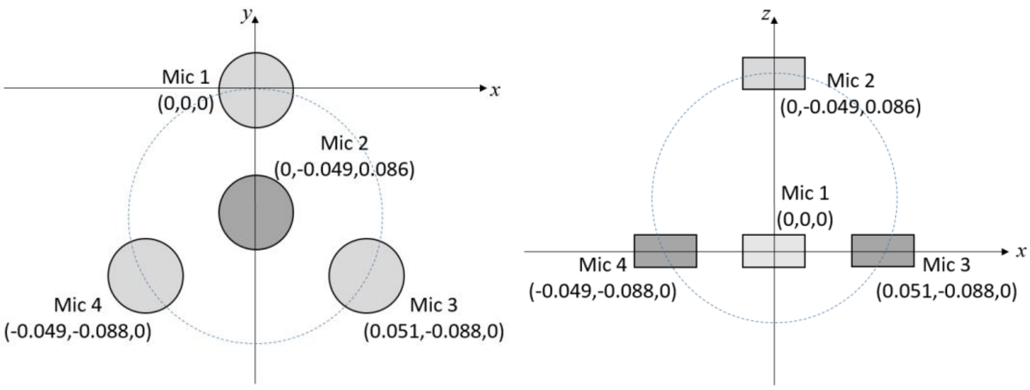



Figure 6 *Top view (left) and side view (top)*

## 8.3 SIGNAL CONDITIONING MODULE

Following the sound detection, which receives sound from an incoming drone, is the signal conditioning module. For each microphone, there are four principal requirements:

1. Bias the microphone appropriately ensuring its correct operation.
2. Convert the microphone output current to a suitably ranged line level voltage signal for use in the rest of the device.
3. Provide further amplification, mapping the maximum expected sound level to the maximum ADC input value.
4. Reject noise not deemed to be from a drone (frequencies above 1500 Hz and below 500Hz).

To fulfill these requirements, the circuit was split into two separate parts, the pre-amp stage followed by the additional amplification stage.

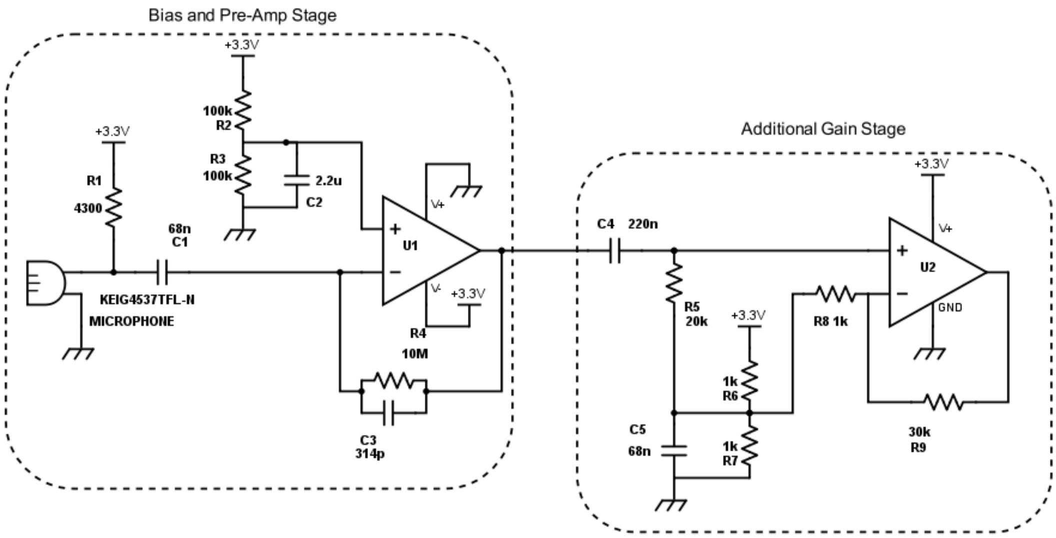



Figure 7 *Signal conditioning circuit for each microphone*

### 8.3.1 Microphone Selection

The microphone chosen (KEIG4537TFL-N) is of unidirectional electret type. Electret microphones have high sensitivity and magnitude response at the desired frequency range. Electret microphones are also small (easy to integrate into the physical design) and are low in price, aiding the attempt to meet the Design Criteria of an affordable device. Unidirectionality is desirable since by rejecting noise from below the device, the signal to noise ratio of the drone (above the device) will be increased.

### 8.3.2 Biasing and Pre-Amplification Stage

This stage of the module is used to ensure the correct operation and conditioning of the microphones and their signals (Texas Instruments, 2015). The first step in this stage is to bias the microphone as specified in the datasheet (Farnell, n.d.). The datasheet states that the maximum current consumption of the microphone is 0.4 mA and the standard operating voltage is 1.5 V. Therefore, R1 is used to bias the microphone with the appropriate amount of current.

$$R1 = \frac{(3.3 - 1.5)\ \mathrm{V}}{0.4\ \mathrm{mA}} = 4.5\ \mathrm{k\Omega}$$

In the actual circuit, a 4.3 kΩ resistor is used, as it is the nearest component value that was available. In addition, a smaller value allows for correct operation in case of a reduction in supply voltage due to loading.

The second stage of the module is to convert the microphone output current to a voltage which can be used by the proceeding modules. To do so, a transimpedance amplifier is used where the exact design is predicated upon the approximate output current of the microphones. The output current was determined by referring to the microphones sensitivity, which is -37 dBV, which in mV $Pa^{-1}$ is:

$$10^{\left(-\frac{37}{20}\right)} = 14.12\ \mathrm{mV\ Pa^{-1}}$$

Given that the output resistance of the microphone is 1.8 kΩ, as shown in datasheet, then the microphone output current is 7.844 μA $Pa^{-1}$. Comparing drone sound levels to the intensities of common everyday noises (CHSL, 2014), it was estimated that the maximum expected noise from a commercial quadcopter is about 70 dB, which in Pa is 0.0631 Pa. Therefore, the maximum expected output current from the microphone is:

$$7.844\ \mathrm{\mu A\ Pa^{-1}} \times 0.0631\ \mathrm{Pa} = 0.495\ \mathrm{\mu A}$$

Furthermore, due to the single supply used, the signal was biased at half the 3.3V supply voltage. Two resistors of the same value are used for R2 and R3 (100 kΩ is large enough so this network does not draw too much current from the supply) to create this potential divider to halve the supply. Consequently, now that both the voltage desired is known, 1.65V, and the current is known to be approximately 0.495 μA, the feedback resistor value can be calculated to be at least:

$$\frac{1.65\ \mathrm{V}}{0.495\mu A} = 3.3\ \mathrm{M\Omega}$$

The op-amp chip chosen for implementation was regarded as the optimal component available for this circuit. It has 4 op-amps in each chip, allowing all the additional gain stages to be designed on one single chip. The op-amp also has a gain bandwidth product of 10 MHz, suitable for the devices expected operational frequency range.

The rest of the components were chosen for a variety of reasons. These ranged from providing necessary amplification in the trans-impedance amplifier to band limiting the signals as described set out in the pre-design testing by filtering. Refer to Appendix 4.1 for a full description of the decision process for all the components.

### 8.3.3 Additional Amplification Stage

Additional amplification was required to ensure that the device operates at larger distances. At large distances, when the drone is farther away, the produced signal is insufficiently small for accurate use by the following processing module. Amplification is a method of rectifying this issue and therefore a gain is imposed on the signal output from the bias and pre-amp stage. The gain is implemented using a simple non-inverting amplifier where the resistors R8 and R9 were chosen carefully to obtain the correct amplification. However, unlike typical amplifier configurations, R8 is not connected to ground, but to the midpoint of the power supply to set this line level voltage at the output of the op-amp. Furthermore, for the same reason as the pre-amp stage, R6 and R7 are used again to provide a midway DC bias. Another reason for providing a DC bias at halfway the power supply is to map the signal to the range allowable by the ADC input (0-3.3V).

The value of the gain was chosen to be 31 due to results from testing showing that this would be a good amount of amplification to reach distances of 10m, before the direct signal of the source becomes indistinguishable from general noise. It was also a good choice because it does not saturate the signals when the drone is close to the device. Although larger operating distances are desired in the future, this downscaling ensured that the goal for the project was manageable given the time constraints. Furthermore, as the gain can be easily changed, demonstrating the devices functionality with a gain of 31 illustrates the device's potential for operation at even larger distances.

The rest of the components in the amplification stage were chosen to provide filtering and decoupling for the circuit. A full description and explanation for the chosen components is available in Appendix 4.1. Furthermore, for a circuit diagram illustrating the complete circuit, with all the IC op-amp chips in place, refer to Appendix 4.2.

### 8.3.4 Filtering

As explained above, filtering is implemented in both the pre-amp and the amplification stage of the signal conditioning module. First order filters are used to remove noise from the power supply as well as to restrict the frequency content of the signals to between 500 Hz and 1500 Hz. However, given that low-pass filters do not provide enough attenuation beyond the corner frequencies, the frequency content was only band limited sufficiently to 10 kHz. Further filtering by means of DSP will be implemented in the data collection and transmission module to adhere to pre-design testing constraints.

## 8.4 Data Collection and Transmission Module

This module acts as the interface between the analogue signals from the previous stage to the digital data required by the PC for processing. In order to work effectively, the system requirements are as follows:

1. Sample the signals from each microphone such that the following module is able to process the data to a high degree of accuracy.
2. Collect as many samples as is necessary in order to make the system resilient to random noise processes (acoustic and electrical noise in the system).
3. Collect as few samples as is necessary to ensure a satisfactory tracking refresh rate.
4. Send the data as quickly as possible over communication link to the processing computer, to avoid delay between receiving data and analysing it.
5. Sample at a rate that satisfies the Nyquist Criteria in order to prevent aliasing of the received signal. A higher sample rate will reduce the requirements on analogue filtering.

This design of this module is separated into three stages (as shown in the block diagram in blue):

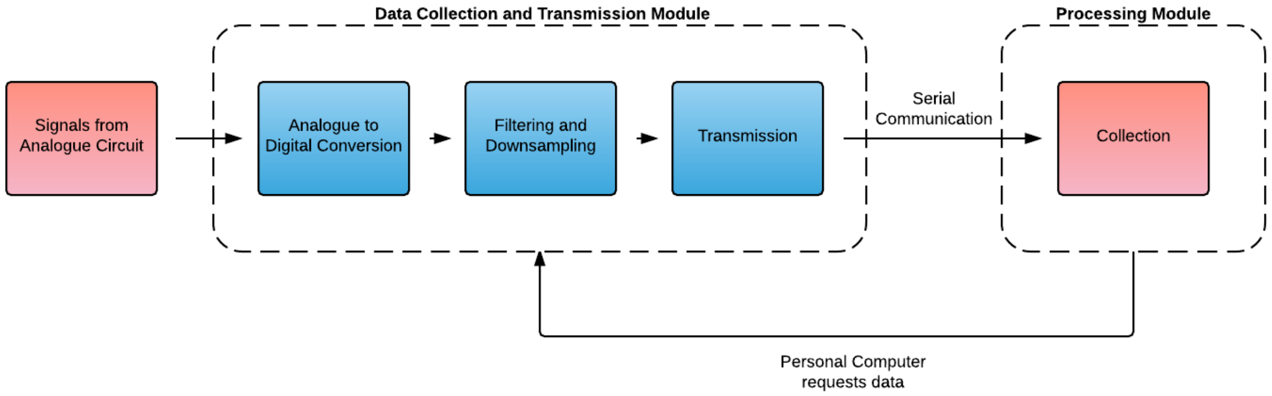



Figure 8 *Data collection and transmission module block diagram*

To perform the three stages, it was decided to use a microcontroller whose sole purpose was to fulfill the requirements of the data collection and transmission module. For the microcontroller the Nucleo F303RE was chosen because it has four 5 Msps 12-Bit ADCs, sufficient for the device's sampling requirements. It also supports the mBed development suite allowing for rapid prototyping without prior familiarity with the ARM Cortex M4 architecture. A further benefit of the chosen microcontroller is the built-in micro USB port and Virtual Serial Port (allowing for easy communication with the PC). This also allows the entire prototype to be powered by USB, making it desirable for a fully autonomous device.

The entire process of this module takes approximately 90 milliseconds, from data collection to transmission. This results in a refresh rate of 11 updates per second (meaning the position of the drone is refreshed 11 times a second). Refer to Appendix 4.4 for the mBed code that implements this module.

### 8.4.1 Analogue to Digital Conversion

One signal for each microphone is input to the ADC. The ADC then proceeds to sample sequentially, scaling the results to 16-bits at a rate of 80 ksps. Sampling each microphone sequentially refers to the process where the first sample taken is from the first microphone, the second sample from the second microphone, and so on until the fifth sample is again from the first microphone (after each microphone has been sampled once) and the loop continues. Therefore, the effective sampling for each microphone is 20 ksps (80 ksps divided by four due to there being four microphones). The signals are sampled at 20 ksps in order to adhere to the Nyquist criteria, given that the signals are band limited to 10 kHz from the signal condition module.

Each microphone is sampled over 40 milliseconds, meaning that each microphone signal is converted to a digital signal with 800 samples. A constant time period between samples is ensured by implementing software interrupts in the code to request and record a sample at a rate of 80 ksps**.** It is important to know the time between each sample of data and to ensure that this time is constant as otherwise this will distort the analysis of the data.

An issue arises when doing the conversion due to an inherent delay when sampling sequentially. To ensure accurate results in the processing, it is imperative that the first sample of each microphone data set occurs at the time $t = 0$. However, in reality, only the first sample of the first microphone occurs at $t = 0$ while the second, third and fourth microphone's first sample is taken at $t = t_{samp}$, $t = 2*t_{samp}$ and $t = 3*t_{samp}$ respectively, where $t_{samp}$ is the time between each sample. A solution for this was found by compensating this inherent delay in the processing module. This will be described in more detail in the processor module part of the report (8.5.2).

### 8.4.2 Filtering and Downsampling

The primary purpose of this stage is to increase the throughput of data to the processing module. Since the ADC input signal is only bandlimited to 10 kHz, it is necessary to sample at 20 kHz to prevent aliasing. However, since processing is only performed on signals below 1.5 kHz (due to pre-defined constraints), it is only necessary to send frequency content within the 1.5 kHz bandwidth to the processing module.

By implementing a Low Pass FIR filter to bandlimit the signal to 2 kHz following the data collection, each data set can be downsampled to 4 ksps whilst adhering to the Nyquist Rate. This decreases the number of samples in each data set from 800 samples for each microphone to 150 samples. Since data transmission takes more time than filtering, this improved the refresh rate from 6.25 updates per second to 11 updates a second, an increase by a factor of just below 2.

### 8.4.3 Transmission

Following the previous two stages of the module, the data is transmitted over serial to the PC. Rapid data transmission to the processing module is crucial to allow for a high tracking refresh rate. A high tracking refresh rate ensures that the location of the drone is updated regularly. A refresh rate of 11 updates per second was possible due to three innovative optimizations done to the transmission process:

1. Data Protocol Optimization

When using built in mBed functions to send data over serial, the basic process that it implements for each sample is as follows:

a. Convert 16 bit binary integer sample to a string
b. Split string into characters
c. Send ASCII binary representation of characters

The maximum 16 bit value of 65,535 is 5 ASCII characters long. Since each ASCII character is 8 bits, the worst case is that for each sample 40 bits are required to be sent, instead of the original 16 bits. An optimization was implemented by changing the data protocol such that the data sent over serial was the original 16 bit binary representation instead of its ASCII encoded equivalent. This gave a throughput increase by a factor of 2.5.

2. Data transmission bit rate

The standard bit rate of USB communication on the F303RE is 9.6 kbps. This means that it would take 1 second to transmit the required data.

$$150 \text{ samples } \times \text{ } 4 \text{ microphones } \times \text{ } 16 \text{ bits} = 9.6 \text{ kb}$$

$$\text{Total time of transmission} = \frac{\text{number of bits}}{\text{bits transmitted per second}}$$

$$\text{Total time of transmission} = \frac{9.6 \times 10^3 \text{ bits}}{9.6 \times 10^3 \text{ bits per second}} = 1 \text{ second}$$

By increasing the serial transmission rate to the maximum it would support without corruption (through trial and error) to 400 kbps, it instead takes 0.024 seconds, a throughput increase of 42 times.

3. Pipelining

The initial design was to implement a strictly sequential process as shown below:

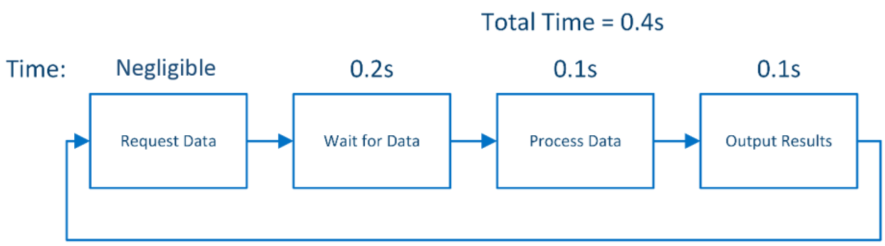



Figure 9 *Strictly linear data collection and transmission*

This process was optimized by exploiting the fact that the system does not have to operate in a strictly sequential process. Whilst data is being processed, the module can also collect and transmit data from the microphones instead of waiting for the processing to complete as was done before (shown below):

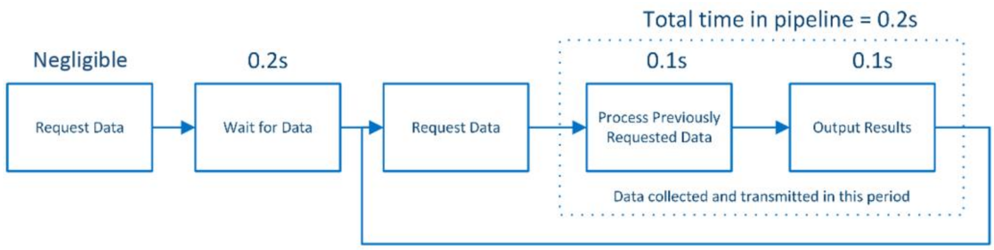



Figure 10 *Pipelined data collection and transmission*

This optimization doubled the refresh rate of the device.

## 8.5 Processing Module

The processing module follows the data collection and transmission module and its purpose is to analyze the data received from the four microphones (after digital conversion by the previous module). The algorithm used on the prototype is implemented using MATLAB on a PC. Even though the initial goal was to implement the algorithm on an Single Board Computer (SBC) using a fast language such as C++, this was changed to downscale the size of the assignment in attempt to make the project more feasible due to time constraints.

The following two sections will outline not only the algorithmic design of the module but how the algorithm was actually implemented in MATLAB.

### 8.5.1 Algorithmic Design

For each angle of arrival of sound on the tetrahedral microphone structure, there exists a unique set of delays by which the sound signal reaches the various microphones. By finding these delays between the signals, one is able to determine the angle of arrival of the sound.

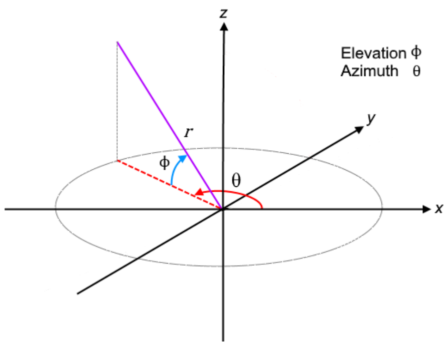



Figure 11 *Definitions of elevation angle and azimuth*

The first step is to declare one of the microphones as the default reference microphone. As the sound wave is received by the array, the other microphones will receive the signal with either a negative or positive time delay compared to the reference microphone. Since the array is designed with the maximum distance between microphones as less than half the wavelength of the highest frequency, for every direction in the sky, there will be a unique delay for each microphone and the reference microphone (as discussed in Microphone Array Dimensions).

In a 2D plane:

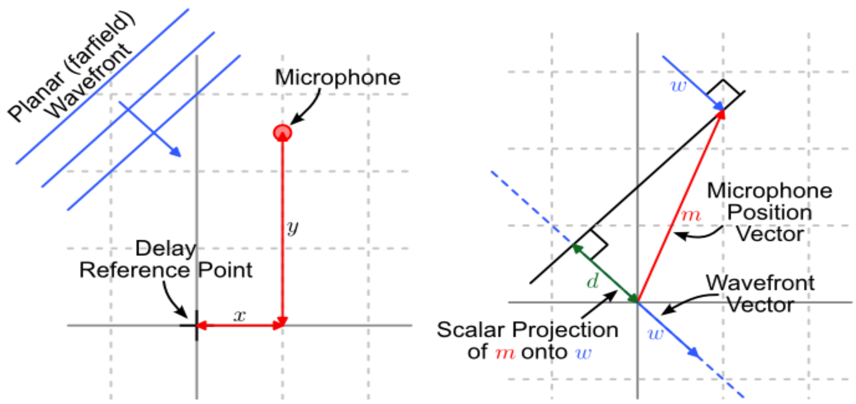



Figure 12 *Diagrams depicting 2D microphone position (left) and dot product delay calculation (right)* (The Lab Book Pages, 2010)

The following relationship, derived from the figure above, is then used to determine the position of the sound source (the drone):

$$d = \text{speed of sound} \times \text{delay} = \mathbf{m} \cdot \mathbf{w}$$

Where $\mathbf{w}$ is the unit vector pointing to the direction of propagation of the sound signal wavefront, and $\mathbf{m}$ is the position vector of the microphone.

This result can be extended to 3D space, giving:

$$\text{delay} = \frac{\mathbf{m} \cdot \mathbf{w}}{\text{speed of sound}}$$

$$\text{delay} = \frac{m_x w_x + m_y w_y + m_z w_z}{\text{speed of sound}}$$

A delay exists between each microphone and the reference microphone, so this result can be imposed on each microphone pairing which includes the reference microphone. As a result, three equations with three unknowns $w_x$, $w_y$ and $w_z$ are produced, due to there being three microphones other than the one chosen for reference. This system of equations can then be solved using Gaussian elimination or other mathematical methods to determine the direction vector components of the drone sound signal wavefront, and by extension the direction of the drone.

The final step is to use the trigonometric relationships of the direction vector components to calculate the azimuth and elevation angle pair of the drone with respect to the device (or more precisely the reference microphone) providing the location of the drone in terms of elevation angle and azimuth.

$$w_x = -\cos\theta\cos\phi$$

$$w_y = -\cos\theta\sin\phi$$

$$w_z = -\sin\phi$$

Where $\theta$ is the azimuth and $\phi$ is the elevation angle from origin to the sound source. The negative sign is due to the fact that position vector of the drone has an opposite direction to the wavefront vector.

### 8.5.2 Algorithm Implementation

To implement the algorithm described above, the algorithm was split into steps where each step in the program has a uniquely defined function corresponding to it. This ensures that not only is the program readable but that it has the necessary functional decomposability desired in high quality programs. Each of the steps and its corresponding function will be outlined below, in order to give a comprehensive summation of the procedure.

1. <u>Resample</u>

A built in MATLAB function called *Resample* allows a signal to be resampled at a different frequency. The function was used to resample the microphone signals at twenty times the original sampling frequency, giving better resolution. Better resolution allows for higher degree of accuracy in the final result of the analysis and was necessary to meet the accuracy requirements set out in the Design Criteria.

2. <u>Calculate Delay</u>

After obtaining the resampled raw data, the next task is to find the delay of the microphones with respect to the first one (previously defined). This is accomplished by performing cross-correlation between the first microphone and the other microphones.

Cross-correlation will provide a measure of the similarity of the two signals being correlated for different time delays. The time delay that corresponds to the largest value (the highest similarity) produced in the cross-correlation is therefore the time delay between the two signals (since with this time delay the two signals correlated are essentially the same). However, because the signals are sampled at discrete time intervals, the maximum correlation may lie between two consecutive points. Hence, we use quadratic interpolation to find the time delay at which maximum correlation occurs. This is done by fitting three consecutive points on the cross-correlation graph into a quadratic curve and taking the time where the curve peaks as the time delay which yields the maximum correlation.

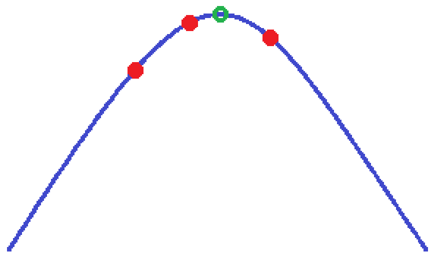



Figure 13 *Quadratic Interpolation Diagram*

A function called *calculateDelay* was written to perform this operation.

3. Adjust Delay

A function called *adjustDelay* is used to compensate the intrinsic delay introduced by the sampling method in the data collection and transmission module (refer to Analogue to Digital Conversion).

4. Calculate Azimuth and Angle of Elevation

Having determined the delay between microphone one and the other microphones previously in the program, these delays can now be used to determine the position of the sound source (the drone), as explained above. A function called *calculateAzimuthElevation* was written to implement this. Its procedure begins with the following relationship:

$$\text{distance difference } \Delta d = \text{speed of sound} \times \text{delay} = \mathbf{m} \cdot \boldsymbol{w}$$

$$\text{delay} = \frac{m_x w_x + m_y w_y + m_z w_z}{\text{speed of sound}}$$

Each microphone gives an equation of 3 variables, and as 4 microphones are used in the prototype, a system of 3 equations is established.

$$\begin{pmatrix} m_{x2} & m_{y2} & m_{z2} \\ m_{x3} & m_{y3} & m_{z3} \\ m_{x4} & m_{y4} & m_{z4} \end{pmatrix} \begin{pmatrix} w_x \\ w_y \\ w_z \end{pmatrix} = \begin{pmatrix} \Delta d_{1\to2} \\ \Delta d_{1\to3} \\ \Delta d_{1\to4} \end{pmatrix}$$

Where $\Delta d_{i\to j}$ refers to the distance difference from microphone $i$ to microphone $j$.

This system of equations can easily be solved by using the inverse matrix method. However, an over determined system would be formed when the number of microphones is increased for future prototypes (to increase accuracy and reduce error in the final result). To enhance the compatibility of the program, the method of least square is used instead of the inverse matrix method. It states that for the system $A\mathbf{x} = b$, the least squares formula (Howard Anton, 2005) is obtained from the problem:

$$\min||A\mathbf{x} - b||$$

The solution of this problem can be written with the normal equations:

$$\mathbf{x} = (A^T A)^{-1} A^T b$$

Provided that $(A^T A)^{-1}$ exists (*A* has full column rank).

However, once implementing this method, errors were found in the angle pair calculation. For the azimuth angle, a systematic error of about 20 degrees was discovered, while complex numbers were consistently generated for the angle of elevation. Investigation revealed that the source of the error was due to the vector $\mathbf{w}$ not being a unit vector as was anticipated. Hence, the constraint of $|w| = 1$ had to be added. A MATLAB script called *spherelsq.m* was used, which was developed based on the method of least square with a constraint on the magnitude published by Bruno Luong (Mathworks, 2010) with reference to a paper by Walter Gander (Walter Gander, 1989). By integrating this script into the algorithm, the azimuth and elevation angle calculation error was significantly decreased.

By combining all these functions together, the raw data sent over from the previous module is successfully analyzed and produces an azimuth and elevation angle, corresponding to the location of the drone, to be used in the following module for the user display. Refer to Appendix 4.5 for the main processing MATLAB code.

## 8.6 OUTPUT DISPLAY MODULE

After calculating the azimuth and elevation angle of the drone with respect to the device's position, the results need to be clearly shown and visualized to the user to be able to divert the threat in a timely manner. To do so, a display implementing a 3D model to illustrate the drone's position in space was developed using MATLAB.

The model consists of a cube with a hemisphere drawn inside on the x-y plane. As the drone enters the range of the device, its location is determined relative to the device. The location information, in the form of azimuth and angle of elevation, is passed to the output display module. The module uses the elevation angle and azimuth to plot a straight line (unit vector) on the 3D model pointing towards the source of sound from the origin, where the origin demonstrates the location of the device. As the drone continues its flight, the straight line from the origin will continuously change its direction in accordance with the most recent value of the azimuth and elevation, essentially tracking the drone. The exact plot shown to the user is displayed below.

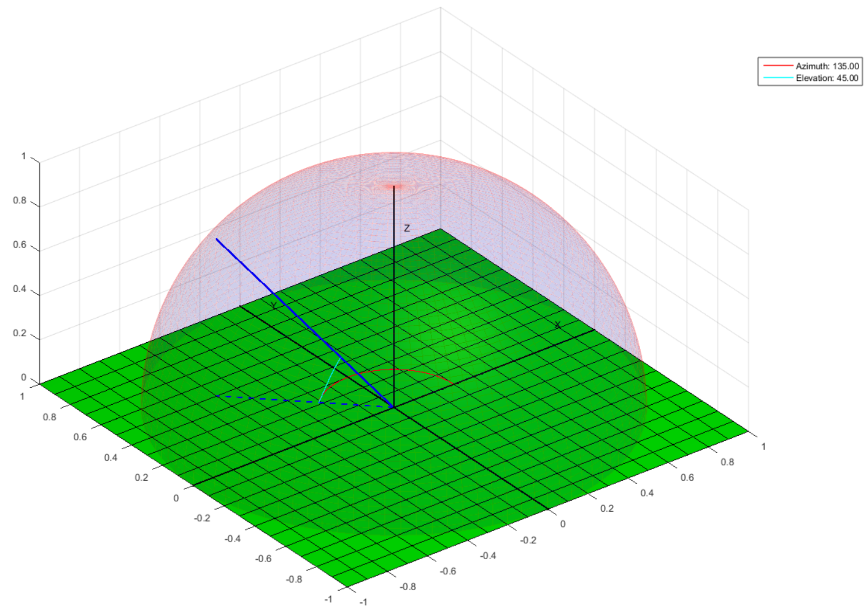



Figure 14 *Output display*

The origin of this system is placed on the reference point, the device (or more specifically microphone 1 as explained in the processing module section). The solid blue line, of constant unit length, points to the sound source from the perspective of the device serving to tell the user the location of the incoming drone.

More information is available on the display such as the red and cyan lines representing the azimuth and elevation respectively, with the exact value of each displayed in the top right corner of the output for reference. The dotted line is a projection of the blue unit vector onto the XY plane and it is displayed in order to decrease the chance of ambiguity in the representation. Refer to Appendix 4.6 for the full output display MATLAB code.

# 9 PROTOTYPE

Having successfully designed and built each module separately, they were then integrated into a working prototype for demonstration. The performance, cost and innovative aspects of this prototype will now be described.

## 9.1 PERFORMANCE

Following the successful integration of each module into a working prototype, testing was conducted to ascertain the level of performance of the drone in parameters such as the operating distance and accuracy.

One test, to determine the maximum operating distance, involved placing the device outside in an open environment with low noise levels, while flying a small drone around the device. It was found that the device was able to consistently detect the drone from distances of up to 10 m. It is expected that for a standard sized drone, the maximum operating distance will extend to up to 500 m (refer to Appendix 4.7 for the full derivation of this result).

A second set of tests to establish the degree of error in the results due to software (the processing module) were also carried out. The testing procedure involved inputting data to the processing module with known elevation and azimuth angles and comparing the recorded output result against the input (the difference was computed). This method was then iterated for each elevation and angle pair with intervals of 1° in each angle. The final step was to calculate the mean error from the data set formed after each iteration was processed. The result was that the average error in azimuth and elevation angle were approximately 2.22° and 0.7238° respectively, exhibiting a high degree of accuracy. However, it was found that the azimuth error increases when the elevation angle approaches 90°. This is because the delay between microphones is very small when the elevation is close to 90°. An extreme case exists when elevation is at exactly 90°, where the azimuth can take any value, yielding a significant error. However, this value was taken as an outlier and was therefore not included when calculating the mean error. This also by extension illustrates that the average errors quoted are only accurate for elevation angles of up to 89°. Possible methods of mitigating this include increasing interpolation and precision of sampling, as well as using a higher number of microphones. Refer to Appendix 4.8 for the MATLAB code used to carry out the error analysis.

## 9.2 NON-TECHNICAL ASPECTS

Non-technical aspects which were addressed during development were regarding the manufacturing process and industrial design of the device. When manufacturing the device, care was taken to ensure that the device remained small and compact. This was done so that the final product would not require a large amount of space to operate, allowing flexibility in the placement of the device. Furthermore, with regards to the industrial design, it was important to keep the device cheap and easily portable. Achieving this ensures that the device is appealing and affordable to a myriad of markets (as stressed in the PDS). Measures taken to achieve this included designing the device to function with a Single Board Computer (SBC), keeping the final product cheap and small. Even though for the prototype a PC was used, this was only done to downscale the project to a feasible level due to time constraints.

## 9.3 COST

The overall cost of the device stands at £12.80, significantly cheaper from other detection devices in circulation (refer to Market Research).

Table 2 *Cost breakdown*

| Components Used | Quantity Used | Price per Unit/£ [1] | Total Component Cost/£ |
|---|---|---|---|
| **Stripboard 95 mm X 127 mm** | 1 | 1.48 | 1.48 |
| **ADC Nucleo Board** | 1 | 7.46 | 7.46 |
| **Electret Condenser Microphone** | 4 | 1.30 | 5.20 |
| **MCP-6021 Operational Amplifier IC** | 8 | 0.80 | 6.40 |
| **Socket IC Holder, Tag, 8-Pin** | 4 | 0.11 | 0.44 |
| **Steel Standoffs (80 mm)** | 2 | 1.65 | 3.30 |
| **Total Product Cost** | | | £12.80 |

## 9.4 INNOVATION

This section depicts how the prototype built in the project differs from already existing drone detecting solutions. One major difference between the device developed and the devices of competitors, is that the prototype is able to notify the user of the exact location of an incoming drone. Currently available audio detection systems only alert the user that there is a drone in the vicinity (refer to Market Research for more details), therefore supplying less information to the user. By providing the exact location of an invasive drone, users will be capable of taking more informed precautionary measures, leading to more successful drone deterring attempts.

Although other drone detection systems exist which are able to discern the exact location of a drone, albeit implementing different detection methods such as radar or image processing, they are invariably more expensive than the prototype developed. Furthermore, even the cheapest audio detection solution available, standing at £50 (U.S. News, 2013), is more costly than the prototype. Two elements outlined in the Design Criteria address the fact that any device built in this project needs to be affordable while also being able to determine the location of an incoming drone (refer to Appendix 1). By developing an innovative and cheap device, which is able to discern the location of a drone, the Design Criteria is sufficiently met.

# 10 PROJECT MANAGEMENT

The group decided at the start of the project to take measures ensuring the project progressed in a timely manner and that the workload was divided equally amongst team members. One measure taken was to form roles for each member and clearly outline the responsibilities every role entailed. This provided a platform for a balanced division of the work as well as guaranteeing that each member's work would be important and relevant for the duration of the project. The roles were as follows:

**Group Leader – Alex Wilson:**

Alex's role was to lead the technical design and manufacturing of the product. This included devising the high and low level design, designing the interfaces between modules, and building the prototype. Also, Alex organised and chaired meetings where the focus was technical.

[1] All prices were taken from EEE stores when available, otherwise were sourced from the Farnell website.

**Operations Manager – Guy Haroush:**

Guy's main responsibility was to ensure the group was meeting its deadlines and weren't lagging behind the pre-determined timeline. Guy also organized and chaired meetings where the focus was administrative, as well as assisting Victor in his development of the website.

**Software Engineer – Pavol Olexa:**

Pavol was a member of the software team, who were given the responsibility of developing the algorithm to be used for the processing component of the prototype. He was also later given the role of developing a means of displaying the results of the device detection to the user.

**Analogue Engineer – Aaditya Malhotra:**

Aaditya was assigned the task of designing and building the circuitry for the signal conditioning module, and ensuring its successful operation. Aaditya was also the main contributor to the interim report.

**Webmaster – Yangshuowen Zhao (Victor):**

Victor was given autonomy to develop the website for the group. This included designing the webpages and deciding what content would appear on the website.

**Software Engineer & Secretary – Chi Leung (Vincent):**

Vincent's principal responsibility, as a member of the software team, was to help design and implement the algorithm used in the processing module. Vincent has also fulfilled his role as secretary by recording minutes during the team meetings.

**Software Engineer – Zhi Chua (Ben):**

Ben, also a member of the software team, ensured the processing module was meeting its requirements by helping to develop the algorithm used. Ben was also tasked with carrying out an error analysis of the processing module.

Apart from delegating roles and responsibilities to each team member, the group also devised a detailed timeline outlining the sub-tasks to be carried out throughout the duration of the project. This allowed the group to track its progress, making sure the team does not lag behind its intended schedule.

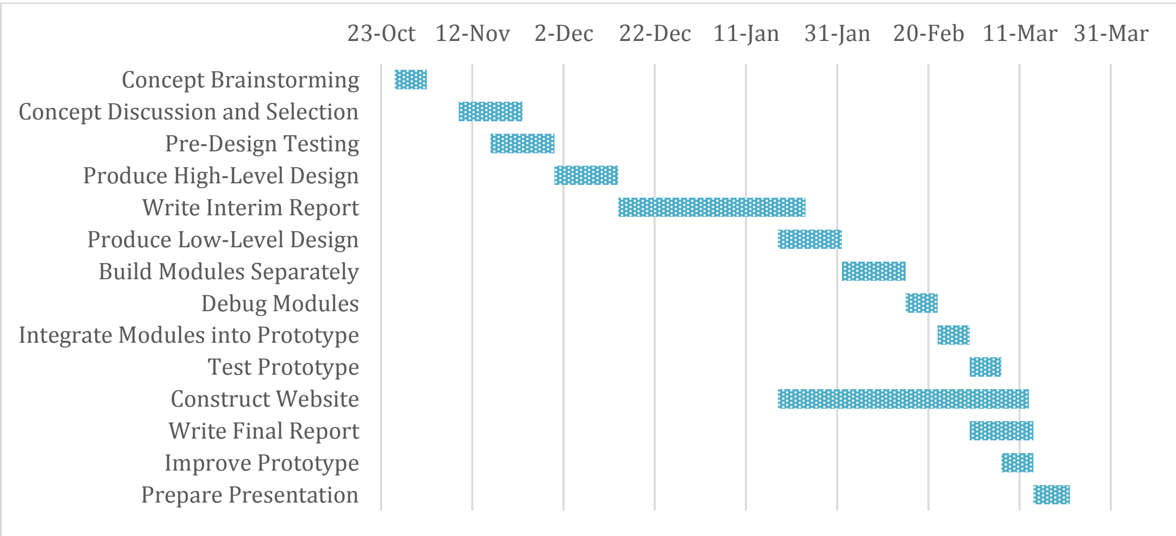



Figure 15 *Gantt chart*

# 11 Conclusion and Future Work

In conclusion, following the development and integration of the five separate modules, a successful drone detection prototype was built. The prototype performed at a high level, able to both detect and alert an individual of a drone while operating at distances of up to 10m. Therefore, the prototype successfully met the Design Criteria, fulfilling the requirements outlined in the performance section (refer to Appendix 1). In addition, the prototype had a manufacturing price of £12.80, thus, meeting the Design Criteria of an affordable device. This also provides a platform for such drone detection products to appeal to large consumer markets. Furthermore, when comparing to other existing drone detection devices, the prototype is innovative in that it is the only device that simultaneously provides the exact location of the drone while alerting the user of its presence.

Although quite promising, there are still aspects of the prototype that can be addressed and improved to achieve an even higher level of performance. Currently, the design of the product is predicated upon the assumption that there would only ever be one drone within the vicinity of the device. However, given the growing commercial drone industry, it is very likely that in the future, multiple drones will be found in any given location where the device may be deployed. Therefore, an improvement would be to change the algorithm implemented in the processing module such that the device is able to detect and output to the user the locations of several present drones.

Further improvements can be found in increasing the critical distance at which the device can detect a drone. Currently, one factor inhibiting this is the presence of crosstalk between microphones. Sources of this include the close proximity of microphone signal paths, and the fact that they share connections to the same nodes in the circuit. However, debugging of the circuit with an oscilloscope showed that the majority of the cross-talk occurs between microphones that share the same op-amp IC chip and the same mid-supply bias networks (refer to Appendix 4.2 for complete circuit diagram). Therefore, efforts into isolating each microphone to a single IC op-amp chip and bias point are key for a second version prototype. This can be implemented by assigning each microphone a MCP6021 op-amp IC that includes one op-amp (instead of four) and a 'Vref' pin that self-biases the op-amp chip at midway the supply.

Another possible improvement to the prototype is to create a mobile application that communicates directly with the device (possibly via Bluetooth). The application would be used to alert and display the location of nearby drones straight to the user via his or her phone, as opposed to doing so via a stationary personal computer next to the device, as is currently done. This would allow members relying on the device, such as security guards at various venues (i.e. airports, prisons, etc.), to not be restricted to one spot when attempting to monitor invasive drone activity.

Furthermore, future work will involve integrating an innovative disabling system that complements the current detection device. Although the prototype can already be coupled with existing disabling methods, by developing a system specially catered to the disabling method in place, the resultant device will offer a high quality autonomous and comprehensive solution to any future drone issues.

# 12 REFERENCES


Anon., 2015. *Drone Detector.* [Online]
Available at: http://www.dronedetector.com/compare-drone-detector/
[Accessed 20 October 2015].

B Yang, J. S., 2005. *Cramer-Rao Bound and Optimum Sensor Array for Source Localization From Time Differences of Arrival.* s.l., IEEE.

BBC, 2015. *Can technology keep our skies safe from nuisance drones?.* [Online]
Available at: http://www.bbc.co.uk/news/business-33989289
[Accessed 22 December 2015].

Camhi, J., 2016. *THE DRONES REPORT: Market forecasts, key players and use cases, and regulatory barriers to the proliferation of drones,* s.l.: Business Insider Intelligence.

CHSL, 2014. *How Loud Is It?.* [Online]
Available at: http://www.chsl.org/soundchart.php

Farnell, n.d. s.l.: s.n.

G.Kino, 2000. *Acoustic waves: devices, imaging, and analog signal processing.* s.l.:s.n.

Help Net Security, 2015. *Drone detection: What works and what doesn't.* [Online]
Available at: https://www.helpnetsecurity.com/2015/05/28/drone-detection-what-works-and-what-doesnt/
[Accessed 3 Janurary 2016].

Hobby King, 2016. *Turnigy Aerodrive SK3 - 6364-190kv Brushless Outrunner Motor.* [Online]
Available at:
http://www.hobbyking.com/hobbyking/store/__18181__Turnigy_Aerodrive_SK3_6364_190kv_Brushless_Outrunner_Motor.html

Howard Anton, C. R., 2005. *Elementary Linear Algebra.* 9th ed. s.l.:John Wiley and Sons.

IB Times, 2016. *Super Bowl 2016: FBI warns of possible 'hi-tech attack' ahead of NFL game.* [Online]
Available at: http://www.ibtimes.co.uk/super-bowl-2016-fbi-warns-possible-hi-tech-attack-ahead-nfl-game-1539316
[Accessed 20 February 2016].

Mathworks, 2010. *Least-square with 2-norm constraint.* [Online]
Available at: http://www.mathworks.com/matlabcentral/fileexchange/27596-least-square-with-2-norm-constraint/content/SphereLsq/spherelsq.m
[Accessed 1 March 2016].

MIT, 2009. *Flight Thrust, Power, and Energy Relations.* [Online]
Available at: http://web.mit.edu/16.unified/www/SPRING/systems/Lab_Notes/airpower.pdf

Mucci, R. A., 1984. A Comparison of Efficient Beamforming Algorithms. *IEEE Transactions on Acoustics, Speech, and Signal Processing,* ASSP-3(3), pp. 548-558.

Open Briefing, 2016. *Hostile Drones: The Hostile use of Droens by Non-State Actors Against British Targets,* London: Remote Control Project.

Ready Go, 2016. *CL720E1 Motor 4 Pcs/set for Hubsan X4 H107C H107-a23.* [Online]
Available at: http://readygorc.com/index.php?route=product/product&product_id=443

Sky News, 2015. *Calls To Police Over Drones Up By 2,000%.* [Online]
Available at: http://news.sky.com/story/1607283/calls-to-police-over-drones-up-by-2000-percent
[Accessed 22 December 2015].

Stanford, n.d. [Online]
Available at: http://adg.stanford.edu/aa241/noise/noise.html

Texas Instruments, 2015. *Single-Supply, Electret Microphone Pre-Amplifier,* Dallas: s.n.

The Lab Book Pages, 2010. *Delay Calculation.* [Online]
Available at: http://www.labbookpages.co.uk/audio/beamforming/delayCalc.html

The Sydney Morning Herald, 2016. *Shark spotting drone the 'future of rescue' in NSW.* [Online]
Available at: http://www.smh.com.au/nsw/shark-spotting-drone-the-future-of-rescue-in-nsw-20160228-gn5leq.html
[Accessed 10 February 2016].

U.S. News, 2013. *Tiny Device Will Detect Domestic Drones.* [Online]
Available at: http://www.usnews.com/news/articles/2013/05/01/tiny-device-will-detect-domestic-drones
[Accessed 20 October 2015].

Walter Gander, G. G. G. U. v. M., 1989. A Constrained Eigenvalue Problem. *Linear Algebra and its Applications ,* pp. 815-839.

# 13 APPENDICES

## 13.1 APPENDIX 1 – DESIGN CRITERIA

Below is the design criteria developed for the project from the previous report. The five categories were chosen as the most important from the product design specification (refer to Appendix 2).

Economics

The design needs to be economical as the product is targeted for widespread commercial use. This necessitates an affordable design which by extension results in a trade-off between performance and cost for the choice of components such as computers and microphones. High-cost techniques such as radar are therefore ruled out as possible solutions. Cost minimization can also be done by avoiding superfluous computer and user interfaces to display the results, or by keeping the physical design of the system modest.

Performance

The basic requirement of the system is to determine the direction (azimuth and the elevation angle) of the drone relative to the detector. The level of precision in the results is expected to be in the order of 0.1° with respect to the device. Refresh rate is also a very important characteristic. As the drone may not be stationary, the system must be able to track it and update the results in real-time. A result of this is to minimize the computations involved in processing the sound signals by devising intelligent algorithms.

The system is expected to detect a drone above its own location, and suppress other noise that may be below it. This might require the use of unidirectional microphones that are only sensitive to the sky above the system.

Ethics and Safety

The system must not conduct, or have the capabilities to conduct, any illegal activity. This includes the jamming of the drone's operation, shooting at it, or any other activity that is deemed illegal by the most current standards of legislation. The latter part of the project may involve developing a drone disabling device to complement the detector, therefore rendering the adherence to this particular aspect of the design criteria crucial in the scope of project.

Reliability and Repeatability

The system must be reliable and able to repeat high quality results consistently despite external conditions or variation in drones. This renders image processing and video detection solutions challenging in this regard due to their struggle to detect the difference between small UAV's and birds[7]. Thermal energy also falters here due to the inability to differentiate many small, electric powered drones that don't produce much heat relative to background radiation[7]. Sound analysis is therefore the most suitable design choice for the device due to the fact that all drones emit a similar type of propeller noise.

Ergonomics and Consumer Interface

The entire system should be enclosed in a casing that will be aesthetically pleasing to the consumer, and can stand alone without support. A screen/monitor is also necessary to display the results of the drone detection to the consumer.

## 13.2 Appendix 2 – Product Design Specification

## Product Design Specification

**Project :** Detection of Civil Unmanned Aerial Vehicles    **Date:** 11/03/16

**Author:** Group 11    **Version:** 2

1. Performance

The product must satisfy a very high speed scan. This must mean that the product will have to minimize computations with intelligent algorithms and component design choices. A minimum criterion is for the system to be able to update its results 30 times a second. This is the minimum rate to provide a fluid flow of the drone's position on the display screen.

Resolution is another important factor of performance. The product must be able to deliver precise results to an order of at least 0.1°.

2. Environment

Product must not directly release any substances that are harmful to the environment (greenhouse gases for example). The system must also conduct its detection without sending out any signals (for jamming purposes) into the environment as jamming this way is illegal.

3. Life in Service

System will work for about 95% of the average quadcopter drones. But as new drone technologies are developed such as glider drones that do not emit any noise, the product will not be of any service and will have to be modified to adapt.

4. Maintenance

Product must be designed in a manner, in which the user cannot accidently cause damage that requires maintenance. This pertains to fitting a wooden casing that encloses the electronic components on the strip board, the single board computer, and other required components. This means that any maintenance should not be done by the consumer, and this also avoids the risk of them damaging any openly placed components.

5. Target Product Cost

The team estimates the product to cost about £50. This includes the purchasing of material as plastics to construct the casing, strip board to mount microphone, mechanical components if needed, microcontroller, SBC, and output display screen. This still makes the product considerably cheaper than its competition.

## 6. Competition

The competition of sound processing to detect a drone usually lies in the realms of radar, sonar, thermal, video imaging, and even detecting via radio frequency analysis.

The concept needs to be cheap enough so that even domestic consumers can purchase it, rather than just well-funded government entities. This in turn rules out radar and sonar, as they are high performance, but quite expensive technologies. Thermal processing involves looking at the specific heat energy generated by the average drone, and then analysing the heat waves being received into the system. This is sometimes quite unreliable with small, electric powered plastic drones that do not generate the amount of heat expected by the system. Also, video or image processing detection methods employ certain algorithms that look at the fight paths of drones in the sky so as to not mistake them for other things, but they are frequently known to yield false alarms by detecting birds.

## 7. Shipping

The product will be manufactured in one location, but it may have a global reach, requiring shipping of the entire assembled product.

## 8. Packing

The product will come ready for use, assembled and constructed by the team. The packing will be a small, sealed cardboard box that is packed with Styrofoam on the inside to protect the system from damage.

## 9. Quantity

One system is delivered for every purchase. This system is a combination of the all the electronic components, the single board computer, microcontroller, and the display screen.

## 10. Manufacturing Facility

Manufacture of the product is rudimentary, and will be done at only one site by team members for the prototype. Manufacturing will only include wiring up the third-party electronic components onto a strip board, 3-D printing out the plastic casing, and connecting the display screen onto the outside of the casing.

## 11. Customer

The target customer for this project is an average consumer who may not be backed by large funds as government entities. Therefore, an emphasis has been made to keep the cost of the product low. The customers can be general individuals, prisons, football stadiums, government agencies to protect important figures or to track terrorist activities.

12. <u>Size</u>

The size of the system is not strictly restricted to any bound size, although keeping a minimal size will allow us to reduce the cost on materials. Having the entire system placed in a cube casing of sides of length of about 1 foot keeps the system small enough to be placed in any suitable location for detection, but also is large enough to encapsulate the electronic components, wiring, and single board computer.

13. <u>Weight</u>

No restriction on weight of the system. Nevertheless, the team is not expecting a heavy product (as electronic components are light and the casing will be constructed from light plastic). Consumers will be able to hold the system in their hands with ease.

14. <u>Materials</u>

Materials used will depend on the specifications of how the system will perform the detection. However, there are some general materials that will be needed regardless of the method of detection. These include a single board computer for processing the received sound signals, microcontroller to sample the analogue signals, a strip board to mount microphones and to produce bias circuitry for the microphones, electronic components (resistors, capacitors, op amps, wiring, etc.), a screen to output results to the user, and plastic to construct the casing for the system.

15. <u>Product Life Span</u>

Will run for as long as the third-party electronic components will function.

16. <u>Aesthetics, Appearance and Finish</u>

As mentioned before, a wooden casing will be used to enclose all the electronic wiring and components. All the user will be able to see will be a screen used to display the results, and the microphones used to detect the sound.

17. <u>Ergonomics</u>

The system should be enclosed in a cube-like casing that will protect it from damage that may be accidentally inflicted by the user. This will hide the complex electronic wiring and components from the user's view.

A cube will also provide a flat bottom surface, allowing the system to be placed on surfaces without external support.

18. <u>Standards and Specifications</u>

Standards set out are that the system will deliver results to angles of 0.1 degrees, and that results are updated 30 times a second.

Other standards are that the system must oblige with health and safety standards for every country that the product is sold to.

19. Quality and Reliability

System must be able to function over a decent range of drone sound frequencies. It is impossible to cover the range of each and every drone, but a bandwidth of approximately 1KHZ can be used as leverage. Sound analysis of a variety of drones can be conducted prior to determine this bandwidth. Reliable functioning is needed, despite external conditions or variation in drones. This includes minor bad weather conditions; if the average quadcopter can function in the given external weather condition, so should the system.

20. Shelf Life (storage)

Not applicable

21. Testing

Two stages of testing are conducted. Pre-design testing is carried out to find the range of frequencies of the average quadcopter drone's sound wave. This is required, as the system will only want to detect sound signals in this given frequency range. The testing comprises downloading sound clips of quadcopter drones and running them through a Matlab function that analyses the sound's frequency spectrum and highlights the peaks corresponding to the frequency of sound made by the drone. At least five such samples are taken and a range is created.

Testing is then done again once the prototype has been constructed. This will comprise of testing all the functions outlined in the concept design section of the report, and making sure the concept adheres to any performance design criteria laid out above.

22. Processes

The process will begin with manufacture of the system as described above. As a customer purchases the system from wherever in the world, the team will then transport the product via air freight, and then by truck to deliver at the consumer's doorsteps.

23. Time Scale

Once a working prototype has been developed, further systems sold to consumers will be able to be manufactured within a week.

24. Safety

The product must pose no physical harm to the user. This involves using materials and components that are not hazardous, toxic, and by ensuring safe connections to the mains power supply.

25. Company Constraints

There are no constraints laid out yet. However, if the team chooses to pursue a disabling feature in the future, the main constraint is to develop a system that is perfectly legal. Lots of disabling techniques can be developed by shooting down the drone or by radiating noise at it, however none of these are legal.

## 26. Market Constraints

The main constraint to the market is the cost of the product. Usual drone detection systems are very expensive and are unable to cater to the general personal consumer market. This constraint is the key for the product to surpass competition and cater to a new market of consumers who are not backed by large funds (as government agencies).

## 27. Patents, Literature and Product Data

Not really applicable. Detection of drones via sound processing has been attempted before and so no new inventions are used in the product. The innovation in the product lies in providing a low cost solution and in using already existing technology in ways that have not been done before.

## 28. Legal

The system must not conduct, or have the capabilities to conduct, any illegal activity. This includes the jamming of the drone's operation, shooting at it, directing noise at it, or any other activity that is deemed illegal by the most current standards of legislation.

## 29. Political and Social Implications

The system will have lots of positive implications. The system will allow users to monitor their privacy very well by being aware of any presence of drones spying on them. It will also allow consumers to remain alert of drones that may carry a physical threat or that may be trespassing (or conducting any other illegal activities). Also this will allow government agencies to use the system to monitor prisons, government owned sport stadiums, residences of important figures for any sign of drones. The system can even be used by rescue services to prevent drones from interfering in rescue efforts.

The system itself will only display the location of the drone, and does not **yet** contain any disabling features. Therefore, there are no such negative implications of the device. It only can give user knowledge over the drone's location, but no power over taking any action (that can be done independently of this system or the team may develop a legal disabling feature in the future).

## 30. Installation

No installation will be needed on the part of the user. The product will come ready, and all the user has to do is to connect the system to the mains and switch it on.

## 31. Documentation

Documentation given with the system to the consumer will contain brief information on how to use the system, how to interpret its results, and who to contact if the system malfunctions and needs maintenance.

32. <u>Disposal</u>

After end of life, approximately 75% of the weight of the system is able to be recycled, this includes components such as the casing and the microphone mounts.  All electronic components will have to be properly disposed of.

## 13.3 Appendix 3 – Discussion and Concept Selection

The team discussed the relative advantages and disadvantages for each concept in the context of the Design Criteria. This led to the group assigning marks for each concept's ability to adhere to each of the five categories in the criteria. These marks were compiled on a concept selection matrix which was used to choose a concept to ultimately pursue for the remainder of the project.

*Option 1: Singular Spinning Microphone*

The advantage of this concept is that it is very simple to design, debug and has a simpler algorithm for detection than the other two concepts. However, the mechanical rotations of the microphone makes it more prone to failure and damage, as well as necessitating more expensive long-term maintenance. The mechanical feature in the concept also means that this method of scanning the sky is slower than the other two concepts. The scanning speed is limited by the speed of the mechanical rotations, while option 2 is able to scan the entire sky just by changing the set of delays used for summation by the computer. In this concept, there also exists a trade-off between speed and resolution. As a scan is done faster, the microphone has less time to process the information received from a particular direction in the sky. Despite choosing a very directional microphone, the concept will inherently have a lower resolution than an array of multiple microphones. Reliability can also be hindered by the fact that the motor and linear actuator controlling the microphone sweep will produce noise to the microphone. This concept is also less cost efficient than the other options because even though it utilizes only one microphone (compared to 8 or 9 in the other options), its cost is significantly augmented due to the stepper motors required to allow rotation. Furthermore, extra material such as plastic and mechanical components will be needed for the construction only further driving up the cost.

*Option 2: Flat Microphone Array with Delay and Sum Beamforming*

The flat microphone array requires less construction material and mechanical components than the other options, making it more affordable. To further reduce the cost, efforts will be made to purchase components without unnecessarily high performance, thus keeping costs low without sacrificing the quality of the product. The flat microphone array also benefits from a very efficient algorithm that is able to scan the sky faster than option 1 as the scan speed is no longer dependent on the mechanical rotation speed. Furthermore, the resolution of the array can be designed to be significantly higher than the others options and is easily adjustable. This extra adjustability means that the concept design is flexible, catering to a myriad of operating environments. Additionally, the adjustability infers that time delays can be designed such that the direction of the drone is known within 0.1° or 0.00001° of an angle from the device, providing a high degree of precision for the location of the drone when in flight. Even though higher degrees of precision implicate longer operating periods for scanning the entire sky, the extra time taken is negligible due to the extremely high initial functioning speeds of the microcontroller (component used to collect data). This concept incorporates the most complicated detection algorithm, but is the most precise and cost efficient.

*Option 3: 3-D Microphone Array*

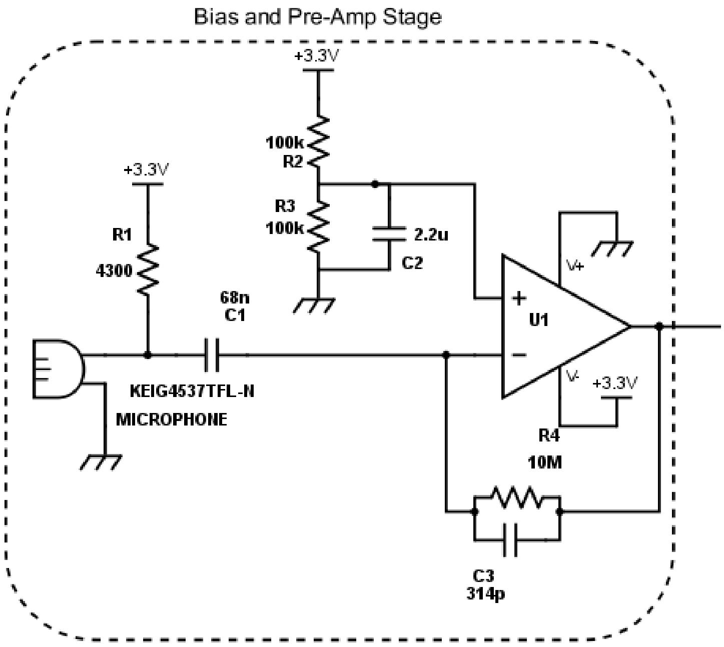
The design of the array is very similar to option 2, but it has an additional microphone and dome structure to support it, which cost more and is more effort to construct. The main disadvantage with this concept's detection method is the lack of high resolution and speed. The method of determining the azimuth angle in this option only gives a rough approximation of the source's angle and isn't as accurate as the previous option. Also, determining the elevation angle requires calculating the angle from recorded time delay measurements for every scan separately. This lowers the speed of the sweep.

## 13.4 APPENDIX 4 – TECHNICAL ANALYSIS AND TESTING

### 13.4.1 Appendix 4.1 – Circuit Design Choices

Design Choices were done using a TI guide (Texas Instruments, 2015).

Bias and Pre-Amp Stage

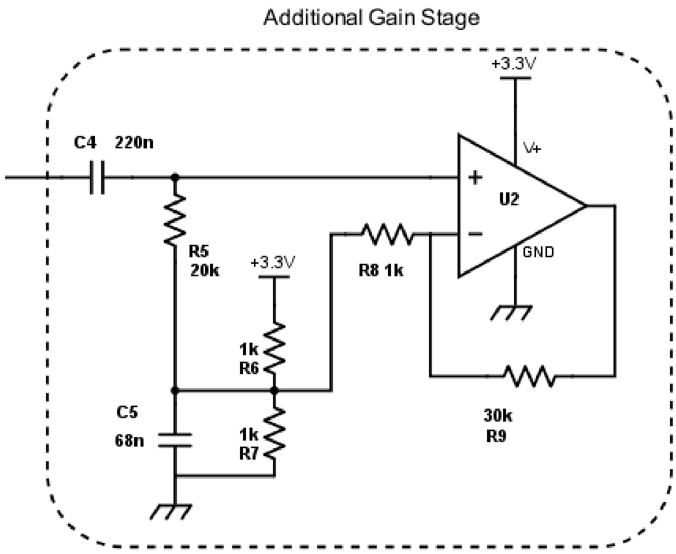



The capacitor, C2, is used to filter any thermal noise from the power supply or noise created by this bias network. C2 also becomes a low pass filter with the 100k parallel resistors and so its value is chosen to filter noise from the supply. The value ultimately chosen for the capacitor was 2.2μF so that corner frequency of the low pass filter stood at 1.44 Hz, which is low enough to block power supply noise.

In order to provide sufficient gain in the trans-impedance amplifier, R4 is shown to have to be at least 3.3MΩ for appropriate gain above, so 10MΩ is chosen as a large safe value to maximize as much gain as possible given the op-amp used.

C1 is used to create a high pass filter with R1 and its value is chosen to place a corner frequency appropriately for our range. C1 = 68nF so corner frequency for high pass is 544Hz.

Additional Amplification Stage




C4 is used to provide AC coupling between the two stages so the output voltage bias is removed from the first stage.

R5 forms a high pass filter with C4 to block DC. The chosen value for RC now provides a corner frequency of 36Hz, which is sufficient to block DC.

C5 is used to create a low pass filter with R6 and R7, to produce a corner frequency at 4681Hz, sufficiently above our interested frequency range.

R8 and R9 are the standard resistor arrangements for a non-inverting amplifier. R8 and R9 values give gain of about 30 (31 to be exact).

Note, all power supplies and ground points in the circuit are common and are sourced from the pins of the ADC Nucleo Board.

The same op-amp is used for the additional gain stage.

### 13.4.2 Appendix 4.2 – Complete Circuit Diagram

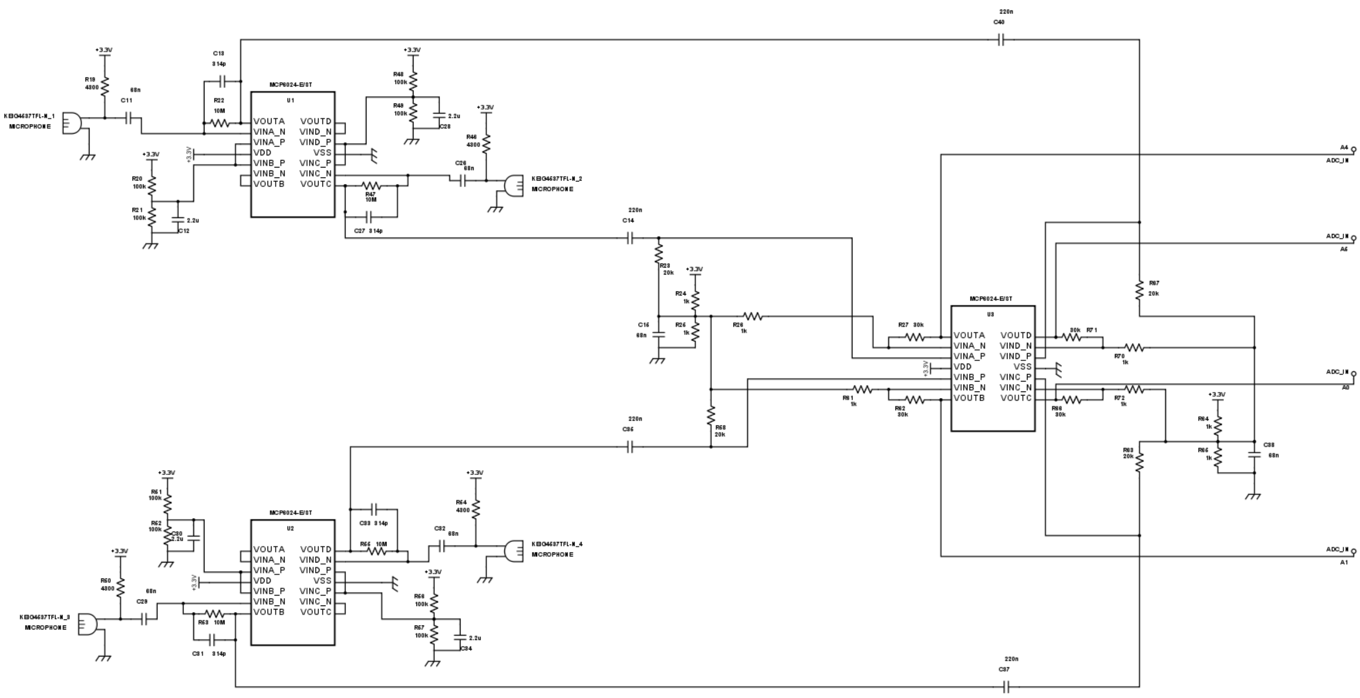


Figure 16 *Complete Circuit Diagram*

This has no difference in circuitry to the simplified diagram shown above in Figure 7 *Signal conditioning circuit for each microphone*. This figure demonstrates how the IC op-amp chips are connected to the rest of the circuit, and also that the unused pins on the op-amps on the chips are connected as normal voltage followers with output shorted to negative input and positive input connected to the midway bias point of the supply.

The design utilizes two 4-op amp chips instead of one for the pre-amp stage. This is to stop any output from op-amp paths crossing with any input (microphone signal output) to op-amp signals from a different microphone. This is because the input signal to the op-amp is much smaller than the output, and if different microphones have their input signals crossing with output signals from other mics then crosstalk is sure to occur. That is why two IC's are used and microphones only use op-amps diagonal to each other in the IC.

### 13.4.3 Appendix 4.3 – MATLAB Code to Determine Drone Frequency Range

```
//reads audio .mp3 file
//takes the fourier transform of the audio signal to analyse its frequency spectrum
//creates a plot of magnitude of the signal vs frequency, illustrating peaks at certain frequencies
//sound(y,Fs) sends the audio signal to the speaker at sample rate Fs to hear the input audio signal

[y,Fs] = audioread('quad (mp3cut.net).mp3');
fourier = fft(y);
amp = sqrt(fourier.*conj(fourier));
plot(amp);
sound(y,Fs)
```

#### 13.4.4 Appendix 4.4 – Data Collection and Transmission mBed Code

```
#include "mbed.h"
#include "Servo.h"
#include "LowPassFilter.c"

#define numMics 4
#define interruptInterval 12.5 //microsecond delay for ADC reading interrupt
#define numSamples 800
#define downSampFactor 5

Ticker readTimer;

Servo elevationServo(D3);
Servo azimuthServo(D5);

Timer timer1;

DigitalOut myled(LED1);

Serial pc(SERIAL_TX, SERIAL_RX); // declare a serial object to communicate with PC

AnalogIn analog_value[] = {A0, A1, A4, A5}; //array of AnalogIn objects defined to the
available analogue in pins, A0=MIC1, D0=MIC8

//initialize global variables (cannot pass variables to ticker function)
uint16_t micCount, colCount;
uint16_t micValues[numMics][numSamples];    //array of mic values
uint16_t interruptIntervalnS = interruptInterval*1000;
unsigned short azimuth;
char  azimuthLSB, azimuthMSB, elevation;
int numberOfSamplesSent;

void readMic(); //function prototype for reading the mic
void setServos(unsigned short azimuth, char elevation); //function prototype for
setting Servos
//function prototype for setting Servos

int main()
{
    LowPassFilter micFilter[numMics]; //array of 4 LowPassFilters

    pc.baud(400000); //increase baud rate from 9600
    elevationServo.calibrate(0.0009, 90.0);
    azimuthServo.calibrate(0.0009, 90.0);

/*      while(1)
        {
             setServos(0,0);
             wait(1);
             setServos(90,45);
             wait(1);
             setServos(180,90);
             wait(1);
    }*/


    while(1) {
        //timer1.start(); //start timer

        //initialise filters for low pass signal
```

```
        for(int i = 0; i < numMics; i++) {
            LowPassFilter_init(&micFilter[i]);
        }

        azimuthLSB = pc.getc(); //get azimuth low byte
        azimuthMSB = pc.getc(); //get azimuth high byte
        elevation = pc.getc(); //get elevation

        azimuth = azimuthLSB + (azimuthMSB << 8); //get azimuth low byte

        setServos(azimuth,elevation);

        micCount = 0;
        colCount = 0;

        //timer1.reset(); //reset timer

        //initialise ticker interrupt
        readTimer.attach_us(&readMic, interruptInterval);   //set 5 microsecond rate
for ticker interrupt giving 50KSPS

        while (colCount != numSamples) {
            myled = ~myled;
        }


        //transfer interruptIntervalnS to serial port
        pc.putc(interruptIntervalnS*downSampFactor & 0xFF); //low byte
        pc.putc(interruptIntervalnS*downSampFactor >> 8); //high byte

        //transfer numberOfMics to serial port
        pc.putc(numMics & 0xFF); //low byte
        pc.putc(numMics >> 8); //high byte

        //transfer numberOfSamplesSent to serial port
        numberOfSamplesSent = (numSamples-LOWPASSFILTER_DELAY)/downSampFactor;
        pc.putc(numberOfSamplesSent & 0xFF); //low byte
        pc.putc(numberOfSamplesSent >> 8); //high byte

        //Filtering
        for(int mic = 0; mic < numMics; mic++) {
            for(int col = 0; col<numSamples; col++) {
                // puts data into the filter
                LowPassFilter_put(&micFilter[mic], micValues[mic][col]);
                // gets corresponding data from filter
                micValues[mic][col] = LowPassFilter_get(&micFilter[mic]);
            }
        }

        //print arrays to serial
        for(int mic = 0; mic < numMics; mic++) {
            for(int col = LOWPASSFILTER_DELAY; col<numSamples; col=col+downSampFactor)
{
                pc.putc(micValues[mic][col] & 0xFF); //low byte
                pc.putc(micValues[mic][col] >> 8); //high byte
            }
        }
        //timer1.stop(); //stop timer
        //pc.printf("process took %f seconds \n", timer1.read());
    }
}
```

```
void readMic()
{

    micValues[micCount][colCount] = analog_value[micCount].read_u16(); //micCount-1
since array starts at 0 but micCount starts at 1

    micCount++;

    if(micCount == numMics) {    //once reach end of mic restart colCount
        micCount=0;
        colCount++;
    }

    if (colCount == numSamples) {            //if reached end of mics, stop interrupts
and return
        readTimer.detach();
        return;
    }

}

void setServos(unsigned short azimuth, char elevation)
{
    if(azimuth > 180 && azimuth <= 360) {
        azimuthServo.write((azimuth-180)/180.0);
        elevationServo.write((180-elevation)/180.0);
    }
    else {
        azimuthServo.write(azimuth/180.0);
        elevationServo.write(elevation/180.0);
    }
}
```

### 13.4.5 Appendix 4.5 – Main Processing Module MATLAB Code

```
% Initialisation code
azimuth = 0;
elevation = 0;
micPositions = [0 0 0; 0 -0.049 0.086; 0.051 -0.088 0; -0.049 -0.088 0];
interpolationResolution = 0.01; % Resample at 100 times the original sampling frequency
speedOfSound = 343.216; % Calculated for at 20 degrees centigrade

sphereLinePlotHandle = setPlot();

while(1)

    % Request parameters and data from serial
    [numSampPerMic, micSampFreq, numOfMics, data] = collectData(serialInstance1);
    requestData(azimuth, elevation, serialInstance1);

    % DATA PROCESSING

    % Filter using design tool fdatool
    % Perform zero-phase filtering on input data
    for i=1:4
       data(i,:)=filtfilt(SOS,G, data(i,:))+(2^16-1)/2;
    end

    % Calculated parameter(s)
    periodSamp = 1/micSampFreq;

    % Shift data values as close to 0 as possible
    for micIndex = 1 : numOfMics
        dataArray(micIndex,:) = data(micIndex,:) - min(data(micIndex,:)) + 1;
    end

    % Interpolate data for each mic

    for micIndex = 1 : numOfMics
        dataArrayInterpolated(micIndex,:) =
resample(dataArray(micIndex,:),1/interpolationResolution,1);
    end

    % Find mic signals delays wrt mic one signal
    % Set delay of mic one signal with itself to zero
    micDelays(1) = 0;
    for micIndex = 2 : numOfMics
        micDelays(micIndex) =
calculateDelay(dataArrayInterpolated(1,:),dataArrayInterpolated(micIndex,:))*periodSamp
*interpolationResolution;
    end

    % Adjust each mic delay due to sampling
    adjustedMicDelays = adjustDelay(micDelays,1/20000,numOfMics);

    % Convert adjusted mic delay to difference in distance
    distanceDifference = adjustedMicDelays .* speedOfSound;

    % Calculate azimuth and elevation of signal source
    [azimuth,elevation] = calculateAzimuthElevation(micPositions,-distanceDifference)
```

```
    % Output display
    sphereLinePlotHandle = displayAzimElev(azimuth,elevation,sphereLinePlotHandle);

    pause(0.0001) % Allows graph to update
end
```

#### 13.4.6 Appendix 4.6 - Output Display Module MATLAB Code

The output display code was composed of two MATLAB functions, *displayAzimElev* and *setPlot*.

```
function [handles] =  displayAzimElev(azi,elev,handles)

    %Set up and display the still objects
    %do this only once
    azi=degtorad(azi);
    elev=degtorad(elev);

    h = handles{1};
    h2 = handles{2};
    arc_azi = handles{3};
    arc_elev = handles{4};
    track_line = handles{5};
    k_arc = 0.3;




    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Drawing
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


    clearpoints(h);
    clearpoints(h2);
    clearpoints(arc_azi);
    clearpoints(arc_elev);
    x = cos(azi)*cos(elev);
    y = sin(azi)*cos(elev);
    z = sin(elev);

    vec_x = linspace(0,x,2);
    vec_y = linspace(0,y,2);
    vec_z = linspace(0,z,2);

    elev_const = elev/pi;
    vec_arc_z = linspace(0,pi*elev_const,10);
    vec_arc_x = linspace(1,cos(elev),10);
    vec_arc_y = linspace(1,cos(elev),10);

    arc_azi.DisplayName = sprintf('Azimuth: %.2f',azi*180/pi);
    arc_elev.DisplayName = sprintf('Elevation: %.2f',elev*180/pi);
    legend([arc_azi,arc_elev],'Location',[0.8,0.8,0.03,0.07]);

    addpoints(h,vec_x,vec_y,vec_z);
    addpoints(h2,vec_x,vec_y);
    addpoints(arc_azi,k_arc*cos(linspace(0,azi)),k_arc*sin(linspace(0,azi)));

addpoints(arc_elev,k_arc*vec_arc_x*cos(azi),k_arc*vec_arc_y*sin(azi),k_arc*sin(vec_arc_
z));
    %addpoints(track_line,x,y,z);
```

```
    drawnow limitrate

    handles = {h,h2,arc_azi,arc_elev,track_line};

end



function [handles] = setPlot()
    view(180,35);
    axis([-1 1 -1 1 -1 1]);
    grid on;


    %x asxis
    x_ax = animatedline;
    x_ax.Color = 'black';
    x_ax.LineWidth = 2;
    x_ax.Tag = 'text';
    addpoints(x_ax,linspace(-1,1),linspace(0,0),linspace(0,0));
    text(0.8,0,0.1,'X');

    %y asxis
    x_ax = animatedline;
    x_ax.Color = 'black';
    x_ax.LineWidth = 2;
    addpoints(x_ax,linspace(0,0),linspace(-1,1),linspace(0,0));
    text(0,0.8,0.1,'Y');

    %z asxis
    x_ax = animatedline;
    x_ax.Color = 'black';
    x_ax.LineWidth = 2;
    x_ax.Tag = 'text';
    addpoints(x_ax,linspace(0,0),linspace(0,0),linspace(0,1));
    text(0.05,0,0.8,'Z');

    %green plane
    x_s = [-1 -1 1 1];
    y_s = [-1 1 1 -1];
    origin_plane = patch(x_s,y_s,'green');

    %grid
    spacing = 0.1; %distances between the lines
    for i = -1 : spacing : 1
        patch(linspace(i,i),linspace(-1,1),linspace(0,0));
    end

    for i = -1 : spacing : 1
        patch(linspace(-1,1),linspace(i,i),linspace(0,0));
    end


    %moving line
    h = animatedline;
    h.Color = 'blue';
    h.LineWidth = 2;

    %xy-plane projection
```

```
    h2 = animatedline;
    h2.Color = 'blue';
    h2.LineWidth = 2;
    h2.LineStyle = '--';

    %azimuth projection
    arc_azi = animatedline;
    arc_azi.Color = 'red';
    arc_azi.LineWidth = 1;
    arc_azi.DisplayName = '';

    %elevation projetion
    arc_elev = animatedline;
    arc_elev.Color = 'cyan';
    arc_elev.LineWidth = 1;

    %track line
    track_line = animatedline;
    track_line.Color =  'black'; %[105/256,105/256,105/256];
    track_line.LineWidth = 2;
    track_line.LineStyle = ':';
    track_line.MaximumNumPoints = 100;

    spherePoints = 100;

    [x1,y1,z1] = sphere(spherePoints);
    x1 = x1((spherePoints/2+1):end,:);       %# Keep top 11 x points
    y1 = y1((spherePoints/2+1):end,:);       %# Keep top 11 y points
    z1 = z1((spherePoints/2+1):end,:);       %# Keep top 11 z points
    hold;

surf(x1,y1,z1,'FaceAlpha','0.1','FaceColor','blue','EdgeColor','red','EdgeAlpha','0.1')

    axis equal  vis3d; %off
    lighting phong;
    camlight('right');

    handles = {h,h2,arc_azi,arc_elev,track_line};

end
```

### 13.4.7 Appendix 4.7 – Derivation of Maximum Operational Distance

Current Drone (Husban X4) has motors that are 1.1w each (Ready Go, 2016).

Drones that are of interest have standard motor sizes of approximately 2.5kW (up to 5kW each) (Hobby King, 2016)

Thrust level is directly proportional to noise power:

‘The effect of thrust level on noise is obtained by simply scaling the sound intensity (I) by the ratio of thrust to reference thrust. ‘ (Stanford, n.d.)

Motor shaft power is directly proportional to thrust (MIT, 2009)

Assumptions made in derivation:

- Current miniature drone has motors that are 1W each
- Medium to large drones that are expected to be a problem have motors on average of 2.5kW each
- Electric motor power is directly proportional to motor shaft power (constant efficiency)
- Motor shaft power is directly proportional to thrust (constant propeller efficiency)
- Thrust level is directly proportional to noise power

Current drone motor power = $P_c$

Operational distance with current drone = $D_c$

Expected drone motor power is $M_e$ times the current drone power

Expected working distance = $D_e$

Result derived from application of inverse square law:

$$P_c \times \frac{1}{D_c^2} = M_e P_c \times \frac{1}{D_e^2}$$

$$\frac{1}{D_c^2} = M_e \times \frac{1}{D_e^2}$$

$$\sqrt{D_c^2 \times M_e} = D_e$$

$$\sqrt{10^2\ m \times 2500\ W} = D_e$$

$$500\ m = D_e$$

### 13.4.8 Appendix 4.8 – Error Analysis MATLAB Code

```
delay = -delayArrayGenerator();
fSigChosen = 1000;
errorArrayfSig = zeros(360, 90);
errorArrayAzimuth = zeros(360, 90);
errorArrayElevation = zeros(360, 90);

for azimuthChosen = 0:359
    for elevationChosen =0:89
        dataArray = dataGenerator(azimuthChosen, elevationChosen, delay);
        [azimuth, elevation] = dataProcessing(dataArray);
        errorArrayAzimuth(azimuthChosen+1, elevationChosen+1) = abs(azimuthChosen -
azimuth);
        if abs(azimuthChosen - azimuth) > 180
        errorArrayAzimuth(azimuthChosen+1, elevationChosen+1) = 360 - abs(azimuthChosen
- azimuth);
        end
        errorArrayElevation(azimuthChosen+1, elevationChosen+1) = abs(elevationChosen -
elevation);
    end
end

meanAzimuth = mean2(errorArrayAzimuth)
meanElevation = mean2(errorArrayElevation)
```

## 13.5 APPENDIX 5 – MINUTES FROM MEETINGS

Monday, 26 October 2015

**Present:** Ben Chua, Guy Haroush, Vincent Leung, Aaditya Malhotra, Alexander Wilson, Victor Zhao, Pavol Olexa
**Absent:** /
**Next meeting:** Scheduled on 02/11/2015
**Discussion:**
*Framework of the Project*

It was agreed that this project would be focusing on detection of position of drones due to time constraint. It was proposed that audio processing would be the method of detection, rather than radar and sonar technology, due to limited budget. Image processing could be considered as assistance to audio technology.

*High Level Design*

To detect the position of drone, it was proposed to use microphone arrays. A brief initial design of having a 2-D array was done, which involves 8 microphones, forming a circle on the same plane. Direction would be determined by choosing the least phase difference between neighbouring microphones as well as largest amplitude among the 2-D array. Distance could be determined by knowing the drone's power output. Through power-distance relationship and the amplitude of sound wave collected by microphone, approximate distance could be obtained. Ways to extend it to 3-D would be discussed next time.

**Action Points:**

*Algorithm*
It was decided that Pavol would be responsible for the algorithm due to relevant researches done by him previously, which would be presented in next meeting.

*Purchase of Drones*
A small drone costing around £10 would be purchased for initial product development. Larger drone by Alex could be used at later stages for testing the prototype of the system.

*Microphone Arrays Design - Beamforming*
Everyone would do research on beamforming on microphone arrays in order to keep track of the target's position (direction and distance). This would be discussed upon meeting with supervisor, Mr. Mike Brookes, on 28/10/2015.

Monday, 02 November 2015

Present: Ben Chua, Guy Haroush, Vincent Leung, Aaditya Malhotra, Alexander Wilson, Victor Zhao

Absent: Pavol Olexa

Next meeting: Scheduled on 09/11/2015

Issues from previous meeting:

*Algorithm*

As Pavol was absent from meeting due to personal reasons, it will be discussed in next meeting.

*Beamforming*

Everyone came up with their own design of microphone arrays.
Discussion:

*Microphone Array Design Resolution*

There were doubts on whether microphones are mostly unidirectional (i.e. the sound coming perpendicularly with the microphone would be of less resolution) or omnidirectional. It was agreed that initially, microphones can be assumed to be omnidirectional, which we will do testing later and prove the assumption.

*Shape and Number of Microphones*

It was agreed that 3 microphones would be required for detecting direction in a 2-D plane. However, for 3D plane, there were uncertainty about the shape due to the resolution problem aforementioned. It was proposed that the minimum number of microphones required is 4, as we have no need to sense sound below the device. Further discussion would be needed with our supervisor, Mr Mike Brookes.

*Design Evaluation*

Despite the fact that we have come up with ideas, there were no concrete benchmarks for evaluation of the design. Therefore, it was agreed that we have to come up with a standard to evaluate the effectiveness and feasibility of a design.

Action Points:

*Criteria of the Design Evaluation*
The criteria of design evaluation have to be concrete enough in order to proceed with a feasible design. This would be discussed upon meeting with supervisor, Mr. Mike Brookes, on 04/11/2015.

Monday, 16 November 2015

Present: Ben Chua, Guy Haroush, Vincent Leung, Aaditya Malhotra, Alexander Wilson, Victor Zhao, Pavol Olexa

Next meeting: Scheduled on 23/11/2015

1. Discussion:

a) Array Design

Upon meeting with Mr. Mike Brookes, it was confirmed the maximum distance between two microphones would be half a wavelength of the highest frequency wave. Hence, the maximum frequency emitted from the drones has to be found in order to design the distance between microphones.

b) Work Allocation

As the interim report approaches, it was decided that every team member would take up a part of it, and compile upon completion. It was agreed that the scope of the report would be focused on detection of drones. The allocation is shown below:

<u>Topic Member(s) Responsible:</u>

Background Victor

Concept Design:

a) Radar Pavol
b) Beamforming (Line, Sphere) Guy
c) Spinning Microphone Alex

Product Design Specification (PDS) Aaditya

Technical Development Ben, Vincent

2. Action Points:

a) Submission of Work

The deadline for every member to submit their part was decided to be two weeks after this meeting.

Wednesday, 25 November 2015

Present: Ben Chua, Guy Haroush, Vincent Leung, Aaditya Malhotra, Alexander Wilson, Victor Zhao, Pavol Olexa

Next meeting: Scheduled on 30/11/2015 (Both General and Technical)

1. Issues from Previous Meeting

As Guy is away on 23rd November, the date of the original scheduled meeting, it is rescheduled to 25th November.

3. Discussion:

a) Principle of Finding Direction

In the previous technical meeting, it was decided that the detection would be based on time delays. By sweeping azimuth and elevation at a certain interval we would like, sets of delays on microphones would be stored in database.

Two methods have been raised to obtain the time delay. First method is through correlation integral, where the time delay is given when the phase difference is largest. Another method is to try sets of delay pre-calculated in database and sum the waveforms up. The one with highest gain/ amplitude would be the correct set of delay. Backtracking gives us the direction.

It was proposed by Alex and agreed by others that in the scope of our project, in which prototype would be made, the second method would be more suitable, due to the fact that the implementation would be much simpler than the first method.

It was also proposed that the interval for sweeping would be 1°. Moreover, successive approximation, e.g. binary search, would make our implementation faster and more efficient.

b) Specifications of Microphone Arrays

It was agreed that ten microphones would be used for accuracy in direction measurement. Microphones used can be a Low-pass Filter as the maximum desired frequency is at about 1kHz. A Bandpass filter around 1k-2kHz would also be added to mostly eliminate the effect of high frequency noise.
Single-Board Computer, e.g. Raspberry Pi, would be used as a main core of our design. Data processing would all be done within the single-board computer.

4. Action Points:

a) Work Submission

It was agreed that Alex would be responsible for the High Level Block Diagram, while Ben and Pavol would be responsible for Algorithms and Simulations, e.g. pseudocode. Deadline would be next meeting.

b) Array Design

It was agreed that we would come up with array design with 10 microphones in next meeting.

Wednesday, 02 December 2015

Present: Ben Chua, Guy Haroush, Vincent Leung, Aaditya Malhotra, Alexander Wilson, Victor Zhao, Pavol Olexa

Next meeting: Scheduled on 07/12/2015 (Tentative)

1. Issues from Previous Meeting

a) Block Diagram by Alex

Alex presented his block diagram to all of the group members. The actual sampling frequency should be higher than the Nyquist Rate $2 * fsig$ It was agreed that as the samples are taken sequentially rather than parallel, the frequency of the single-board computer would be at least $2 * fsig * n$, where n is the number of microphones.

b) Design Algorithm by Ben and Pavol

The Design algorithm is presented through MATLAB. To make the calculation more convenient, a cube is used for the microphone arrays, due to the fact that a cube is symmetric and the position vectors of each microphone could be easily derived. However, problems arise for finding the maximum peak of the sound source.

5. Discussion:

a) Shape/Layout of Microphone Arrays

As aforementioned, it was agreed that the shape of the microphone is a cube. Hence, the number of microphone used is 8 (vertices of a cube).

b) Principle of Finding Maximum Amplitude

To prevent the dominance by sudden noise other than the sound from the drone, it was suggested that using the mean might be a better choice.

c) Negative Time Delay

It was questioned that if negative delay matters, as it is equivalent to predicting future sound waves. It was explained by Ben that it doesn't matter, as long as we fix the time frame where we sum the samples (which is not immediately from the microphone input, but rather the input from the past). To know the data we have to store in our system, we have to know the worst case scenario. The total number of samples we have to store is $(Tdmax - Tdmin) * fsamp + n$, where n is the number of samples (frame) that we would like to sum them up.

d) Sampling Frequency

It was agreed that the higher the sampling frequency, the higher the accuracy of retrieving the original waveform. A compulsory requirement is that it must be higher than or equal to the Nyquist Rate. However, it is limited by the clock frequency of the single-board computer we will have.

6. Action Points:

a) Circuit Diagram

It was suggested that a detailed circuit diagram for the overall circuit would be done. The Block Diagram done by Alex would be used as a reference, where the variable gain amplifier would be a possible optional add-on to the circuit.

b) Work Allocation

The group is divided into various sub-groups. Each sub-group would have to come up a detailed design (e.g. circuit diagram, component, drawings). The work is allocated as following:

Topics Member(s) Responsible
Microphones, Multiplexer Alex
Filter Aaditya
Single-Board Computer and Display Ben, Vincent, Pavol
Physical Structure/Building the Microphone Array Victor, Guy

Note 1: The tentative time for meeting of group responsible for Single-Board Computer and Display is at 2pm, Sunday, 6th December.
Note 2: For the group on Physical Structure, please refer to the 1st technical meeting on setting the length between microphones.
c) Drone Sound Frequency

As measured before, the maximum drone sound frequency is around 1kHz. However, it was only based on the only recording that was available online. To ensure the accuracy of the frequency, it is suggested that more and longer soundtrack samples could be obtained, from which the maximum source frequency could be found. The related MATLAB file is already on SharePoint, which can be used to analyse the recordings. The frequency obtained would be a determining factor to the length of the cube microphone array.
It was agreed that Victor would be responsible for collecting more samples. The minimum number of samples is 5.

Friday, 11 December 2015

**Present:** Ben Chua, Guy Haroush, Vincent Leung, Aaditya Malhotra, Alexander Wilson, Victor Zhao, Pavol Olexa

1. Discussion:

a) Physical Design of the Microphone Arrays

After discussion at the Technical Meeting, it was decided that octagon would be preferred rather than cube, so the device would be more compact, as well as easier to implement.

b) Design of Bandpass Filter(BPF)

After collection of more sound samples, it was decided that we would mostly consider the sound around 1kHz. Hence, we would need a BPF with corner frequency at 710Hz and 1.4kHz. (3dB point)

c) Modifications of Block Diagram

Rather than having a single-board computer alone, it was agreed that another microcontroller, replacing the multiplexer due to transient problems, would be added for functions like selecting microphones, while the single-board computer would only be served for data manipulation (data processing center), in which delay would be found. Some choices of components were suggested by Alex, and would be uploaded to SharePoint for reference. It is expected that this method would enhance the sampling rate, which would be ideal for our purpose.

2. Action Points:

a) Work Allocation

As the end of first stage arrives, the work has been allocated for the implementation of the prototype. Ben, Vincent and Pavol would be responsible for the software for data processing on the PC, which would mainly be processed through Python. The part for the microcontroller (ADC, data transfer) would be done by Alex, while the report-writing and physical design would be done by Guy, Aaditya and Victor. Victor would also be responsible for the website. Everyone would need to do research on the components required for their parts, so that they can be purchased next term.

Thursday, 21 January 2016

**Present:** Ben Chua, Guy Haroush, Vincent Leung, Aaditya Malhotra, Alexander Wilson, Victor Zhao, Pavol Olexa

**Next meeting:** Scheduled on 28/01/2016

**Issues from Previous Meeting**

Following work allocation in previous meeting, this week's meeting would be focusing on the updates on every team on their work. The components ordered have all arrived. Testing on the components has begun.

**Discussion:**

*Work Progress*

*Data Collection*
For Data Collection, the code has been done using MATLAB. It has been tested that using the components ordered, the microphones collect data correctly, hence showing the phase shift according to different azimuth and elevation of the incoming sound. Only a slight error in detection of frequency was observed when 1 kHz sine wave is played, due to limited sampling frequency (20 kHz for each microphone).

*Software Development*
Code has been done on Python. However, contrary to our original decision, correlation is used because an existing algorithm would be easier to implement, rather than developing our own algorithm. Correlation can be done through Fast Fourier Transform (FFT), where the time when correlation function attains maximum is the time delay of the sample concerned with respect to the template. Time delays has been input to the python code, and the correct azimuth and elevation was shown. However, there were problems when we tried with inputs generated from MATLAB. It is hoped to be solved before next meeting.

*Physical Design*
Instead of starting with 8 microphones, it was agreed that 4 microphones would be a better start-off point, as it is occupies less space, and would involve a simpler geometry (square). Further increment of microphones might be done if the accuracy and precision is not as good as the specification we set for the product.

**Action Points:**
*Work for Individual Teams*
For software team, they would be working on their codes and debug it. For Data Collection Team, it is expected that code-refining and Serial Interface would be the main focus to work on. For Physical Design team, they would be responsible for designing the layout of the microphones, meanwhile Guy and Victor would also be working on the website.

Friday, 05 February 2016

**Present:** Ben Chua, Guy Haroush, Vincent Leung, Aaditya Malhotra, Alexander Wilson, Victor Zhao, Pavol Olexa

**Next meeting:** Scheduled on 24/02/2016

1. **Issues from Previous Meeting**

   This meeting was organised to present every team's progress on their work following last meeting.

2. **Discussion:**

   a) *Work Progress*

      i) *Prototype*

         As all of the team has finished the first draft of their work, all are combined to a first prototype. It was demonstrated that the first prototype responded to sound source, where we used our mobile phone to play drone sound, well, displaying the correct azimuth. However, there are still some bugs in the prototype, like the elevation is an imaginary number. Regarding to the error, in the meeting, changes of product specification have been proposed and agreed.

   b) *Product Specifications*

      i) *Physical Design*

         According to the software team, the reason behind imaginary elevation is due to the lack of 3$^{rd}$ dimension. Also, the current design is planar, hence it is vulnerable to sound source right above the mics. Therefore, it was proposed and agreed that tetrahedral should be used, to remove the bug. It would be implemented by placing a bolt up from the stripboard to fix the mic. An alternate design would be putting four bolts up to hold stripboard for the mic. A power bank might be used to power up our device.

      ii) *Data Collection*

         Due to the limitation of the USB interface on speed for data transfer, as well as the necessity to put the box remotely from the computer, the data transfer method would be changing from wired to wireless through Bluetooth. A Single-Board Computer is no longer required, where laptop is used for data processing.

      iii) *Distance of Detection & Disabling*

         It was agreed that the target range of our product should be up to 20m. For disabling, at the moment it would be hard to implement. Instead of actual implementation, it would be replaced by placing a servo that show the direction of drone. However, the indicator was not confirmed due to safety concern, where laser pointer had been considered.

   c) *Deliverables*

      i) *Website*

         The first draft of the website was discussed with basic layout done. It was agreed that the layout is good, yet the background has to be changed. Other content, e.g. team photo, would be added as soon as possible.

      ii) *Presentation*

         For presentation, it was agreed that flying big drones in the lecture hall or presentation room is dangerous, therefore, small drones or even mobile phone can be considered as replacement. Video would be taken as a supplement to our presentation, which would demonstrate the work of the prototype with real drones.

3. **Action Points:**

   a) *Work for Individual Teams*

      i) *Physical Design*

It is expected that the tetrahedral building would be completed by next meeting.

*ii)* *Data Collection*
Alex would be responsible for the wireless data transmission and servo, among which Bluetooth data transmission is prioritised. It is expected that it would be completed by next meeting.

*iii)* *Software Development*
On the problem of solving imaginary elevation, meeting with Mr. Mike Brookes would be arranged, discussing the problem on solving system of equations, and also noise rejection in cross-correlation.

Wednesday, 02 March 2016

**Present:** Ben Chua, Guy Haroush, Vincent Leung, Aaditya Malhotra, Alexander Wilson, Victor Zhao, Pavol Olexa

**Next meeting:** Scheduled on 09/03/2016

4. **Issues from Previous Meeting**

   This meeting was organised to discuss the upcoming work as deliverable deadline approaches.

5. **Discussion:**

   d) *Change of Role*

   The role change is as below:
   Aaditya - Analogue Engineer
   Ben, Vincent - Software Engineer (Mathematical Analysis)
   Pavol - Software Engineer (Implementation)
   Victor - Webmaster
   Guy - Head of Administration
   Alex - Team Leader

   e) *Website Design*

   It was agreed that the layout of the website is good. GIF of the working principle was considered to be added to the website. Team photos and video would be added as soon as they are processed.

   f) *Software Algorithm*

   After meeting with Mr Brookes, it was suggested the interpolation and the findFrequency could be improved. He suggested that our prototype will have a range of around 2m, as that is the critical point where echo power will be similar signal power. It was tested in 408 that there might be flips when the distance becomes 2m.

   g) *Output Demonstration*

   After discussion, it was agreed that the layout would include a hemisphere, azimuth and elevation, vector and its projection and flight path. Pavol and Alex will be working on that. Smoothing function will be added for the flight path. (The range is 4m in anechoic chamber.)

   h) *Datasheet*

   It was agreed that datasheet would be made for specifications of the prototype. It will include the frequency range, distance range for expected final product and prototype, error analysis, SNR. The breakdown cost of the prototype will also be done by Aaditya.

6. **Action Points:**

   b) *Final Report*

   It was agreed that everyone would write up their part and submit it to Guy by Friday. Draft report would be completed by next Wednesday.

   c) *New Prototype*

   A new prototype will be built by Aaditya to optimize the circuit and reduces noise and crosstalk in chips, while the current prototype would be treated as a backup for presentation. Alex will work on the servo and Pavol would be working on the smoothing function.