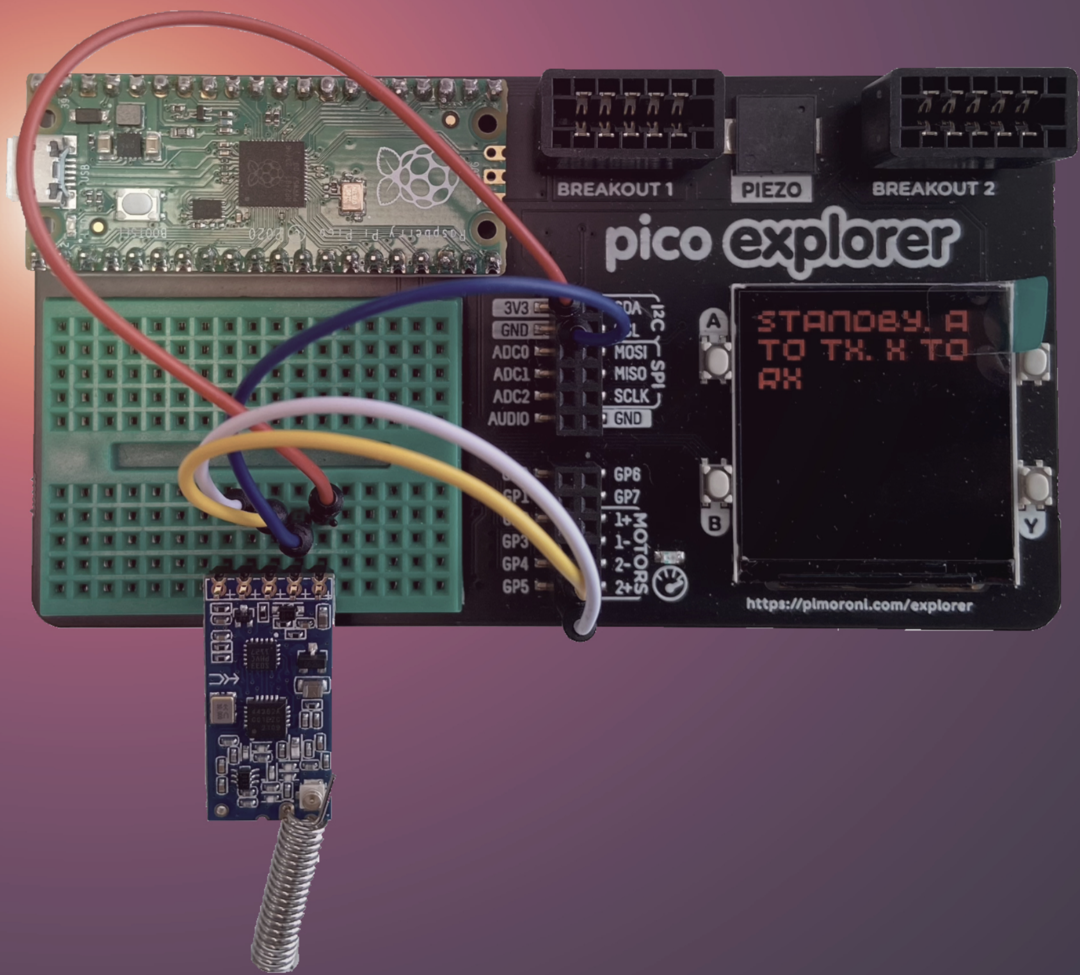
Automatic Dependent Surveillance for Drones: a Design and Capacity Study

Aleksandr 'Sasha' Vlaskin

Faculty of Aerospace Engineering



TU Delft Delft University of Technology

Challenge the future

Automatic Dependent Surveillance System for Drones: A Design and Capacity Study

by

Aleksandr 'Sasha' Vlaskin

to obtain the degree of

Master of Science

In Aerospace Engineering
at the Delft University of Technology
to be defended publically on Monday 11th of July 2022

Student Number:	4535308	
Thesis Committee:	Dr. J. Sun	TU Delft, Supervisor
	Prof. Dr. Ir. J.M. Hoekstra	TU Delft, Chair
	Dr. R.T. Rajan	TU Delft

TU Delft

Preface

This report contains an overview of the work that I have done throughout the past academic year on my Master's thesis. The principal focus is the outcome and underlying methodology, rather than the process. On a day-to-day basis, the tasks were always changing, as I alternated between theoretical elements, building the Python model, and working on the hardware demo. It was a topic that held my interest from start to finish, and while there is a significant amount of work to be done for such a system to be implemented on real drones, I hope that this thesis paper can help identify constraints and avenues of research helpful to future research. Overall, it has been a pleasure working on this topic, and in progressing through it, I have learned a lot.

I would like to thank Dr. Junzi Sun for proposing this research domain, as well as for his continued interest, tireless enthusiasm, invaluable insights, and help. The meetings were always positive and helped to steer the project in the right direction even when I was unsure of myself. Another person to whom I extend my gratitude is Prof. Jacco Hoekstra, whose input at the thesis milestones helped me ask the important questions about how to continue with my work.

Many thanks to my friends, with whom I shared a library study room for most of the working days. Thanks for all of the feedback on my presentations, reports, code and for listening to me rant when things were not working correctly. Lastly, a heartfelt thanks to my parents and family, who believed in me from the very start of my journey and have been there for me every step of the way. Without you to motivate me, I would never have gotten this far.

Contents

Part I

Scientific Paper

List of Figures

List of Tables

Automatic Dependent Surveillance System for Drones: A Design and Capacity Study

Aleksandr "Sasha" Vlaskin

Under supervision of Dr. J. Sun and Prof. Dr. Ir. J.M. Hoekstra

Faculty of Aerospace Engineering,
TU Delft,
The Netherlands

ABSTRACT

The consumer drone sector is expected to grow rapidly in the coming decades. In Europe alone, it is predicted that in excess of seven million such machines will be flying by 2050. This poses a risk of conflict in dense airspaces, with both aircraft and other drones. Such a growing market provides a need to make drones visible to ATC and other airspace users. While several passive surveillance methods exist, such as primary drone radar, a cooperative surveillance system would provide more data to airspace users and other drones to allow for features such as automatic separation. An Automatic Dependent Surveillance system concept is presented in this paper, allowing the drone to broadcast information about itself without external input. This is akin to ADS-B, from which the system inherits its format for the time being. The study's main contents are threefold. The first consists of recommendations made on the basis of literature. Then, a simulation approach to examine system capacity and related constraints through a sensitivity study is done. Finally, a hardware proof-of-concept, consisting of inexpensive and simple off-the-shelf components is built and tested. Overall, the paper demonstrates that such a system is indeed feasible. Through the literature, it was found that direct integration of the system with current ADS-B on the 1090 MHz frequency is possible, but may cause performance degradation for existing aircraft. Therefore, the carrier frequency and code allocation are changed. The simulation and capacity study shows that the system can function in dense scenarios (in excess of one drone per square kilometer), but will require additional work on hardware, format and modulation techniques to enable this. Finally, the hardware demonstrator shows that an inexpensive COTS implementation with a range of approximately 200 meters is possible, on hardware drawing less than five Watts of power.

Keywords: Drones, SUAS, surveillance, cooperative, ADS-B, Mode S

1 Introduction

The global consumer drone sector is entering a phase of rapid growth, with seven million consumer and 400,000 commercial drones expected to be flying by 2050 in European airspace alone [23]. With new applications such as parcel delivery [20] becoming more viable, drone densities in urban areas will grow, leading to a greater risk of collision between drones and aircraft [2], as these machines are not visible to each other or Air Traffic Control (ATC). While surveillance systems tailored to small unmanned vehicles exist (such as Robin's Elvira and Iris drone detection radar [10]), these are primary (passive), and only allow for tracking drone position and altitude.

There is an urgent need for a surveillance system that would allow cooperative surveillance, akin to that seen on aircraft. Compared to primary radar and passive surveillance, cooperative methods allow for the aircraft to share data from its onboard sensors, improving accuracy, reliability and reducing costs. It would therefore be beneficial for such a system to be adopted on drones. A candidate for this would be Automatic Dependent Surveillance-Broadcast (ADS-B), a well-established and routinely used Mode S Extended Squitter service that is used on modern aircraft. This service periodically broadcasts a Mode S message over the 1090 MHz frequency without the need for external input. The idea is to adopt this system directly, or adapt the technology for drone use. Such a technology would allow for drones to be monitored and guided with ease, enabling additional features such as automatic separation and conflict resolution in autonomous flight.

From traditional ADS-B, several early difficulties were identified and focused upon. Firstly, cooperative surveillance requires that the transmitter must have a unique identifier - else, information received by Air Traffic Control could be coming from a whole host of different aircraft. How does one provide such an ID for drones? Secondly, the current 1090 MHz scenario is threatened by heavy amounts of congestion and transponder over-interrogation [19]. While over-interrogation will not be an issue for the drone ADS-B-like system, system performance degradation due to message overlap is deemed to be an important parameter to consider.

As formulated at the start of the project, the main research question was: "Is it possible to design a (ADS-B compatible) ADS system for drones [27]?" Several other sub-questions were posed, the principal of which for this thesis is "How to design simulations to study the implementation and capability of such a system?"

2 Background and Previous Work

Before the simulation and hardware work could begin, all information related to the topic was gathered and analysed. There are three main literature facets identified for this topic: the Mode S and ADS-B literature, the literature pertaining to ADS-B implementations for drones, and finally examples of other drone surveillance systems.

2.1 ADS-B and Mode S literature

In order to evaluate the feasibility of implementing an ADS system on drones, it is important to understand the underlying principles of existing Mode S and ADS-B. The main sources used for this are *"The 1090 MHz Riddle"* [26], with the support of the ICAO documentation [1]. The most important contents are the ADS-B message specifics, elements of which are used as a reference for the drone ADS format.

From this, it is important to mention the format used for ADS-B messages. ADS-B messages rely on the Mode S message format, and therefore contains 112 bits of information [1] [26]. This structure is shown in Figure 1.

DF (5)	CA (3)	ICAO (24)	ME (56)	PI (24)

Figure 1: Mode S Message Structure [26]

The DF field is the Downlink Format, the CA field is the Transponder Capability, ICAO represents the 24-bit unique ID, the ME field is the message field and finally PI represents the Parity Index, which is used for error detection [26]. The 56-bit Message Field is of the utmost importance for the drone ADS system, as its contents define the sensor information (such as the position and altitude) received by ATC and other drones. The field is divided as follows:

TC (5)	CA (3)	C1 (6)	C2 (6)	C3 (6)	C4 (6)	C5 (6)	C6 (6)	C7 (6)	C8 (6)

Figure 2: ADS-B Message Structure [26]

The TC is the typecode, CA is the transponder capability, and the C fields are 6-bit characters used to store data. This means that with 8 6-bit fields, the sensor data transmitted, for example the position of the drone, must fit within the 48 available bits.

2.2 ADS-B implementation on Drones

An important facet to consider was the evaluation of the effects of direct ADS-B implementation on drones. There were several papers of note, namely "Impact of SUAS Equipped with ADS-B on 1090 MHz Environment"[18] and "ADS-B Surveillance System Performance With Small UAS at Low Altitudes" [17], which evaluate the effects of direct adoption of ADS-B on drones/small unmanned aerial systems.

While analysis differs between the two papers, the main conclusion is that the transmitter power needs to be balanced against the drone density for the impact of drone ADS-B transmissions not to degrade the situation for aircraft. The first paper even identifies that the aircraft density plays a role in the impact of the drone's ADS-B signals on the quality of aircraft ADS-B, as drones may have a stronger effect on low-density aircraft traffic [18].

2.3 Related Initiatives, Developments and Systems

In order to frame the research done, it is necessary to look at current initiatives and potential applications. The main project to integrate drone traffic and surveillance across Europe comes from SESAR Joint Undertaking [12]. Within SESAR, the principal of these is U-Space.

U-Space is defined to be a set of new services, which are to rely on data digitalisation and functional automation for unmanned vehicles. This set of services will focus on a safe and efficient integration of dense drone traffic into existing airspace [24]. The main features that were of interest to the project are foundation services that are intended to cover registration and identification of drones, and the advanced set of services which can enable automated detection and avoidance for conflict resolution in dense airspace. This conflict resolution will require drone positions to be known, and while passive sensors can achieve this, on-drone implementation of radar could be limited.

An ADS-B like system is likely to help enable this, through drone position and altitude broadcast, which makes it a relevant field of research.

The U-space initiative encapsulates projects such as CLASS (The Clear Air Situation for uaS) [21] [22], which aim to demonstrate and test potential technologies to enable the drone surveillance enhancement. For this set of trials, a passive holographic radar from Aveillant was used in conjunction with a proprietary GNSS-based DroneIT! board from Airbus [22] [21]. While the capabilities of this board may well solve the issue of implementing an ADS-B like system for drones and aid with conflict resolution, little to no concrete information could be gathered on it, requiring further research.

3 Methodology

The methodology used to obtain the results will be outlined in this section. They are obtained through two main simulation components: the power matrix overlap and the signal timing simulation.

3.1 Spatial Analysis through Power Matrix Overlap

The Spatial Analysis aims to evaluate the impact of messages transmitted simultaneously on the resulting signal-to-noise ratios of the signals. This is done through a simulation, which requires several main components.

Firstly, the initial conditions and variables are set up. The simulation space uses a 10-by-10 kilometer grid. The coordinate system is Cartesian, with the origin found at the bottom left of the matrix. Only positive coordinates are therefore used. For more efficient computation, each coordinate is taken as an (X,Y) tuple into a matrix comprised of nodes on the 10-10 km grid. The resolution can be varied, but the optimum between performance and number of nodes is found at 1000 nodes per axis, or a 10 m node-to-node distance.

It is possible to either define the number of drones and their parameters, or to generate these randomly. The drone position tuple is subtracted from every element of the position matrix, and the result is processed to obtain a matrix of distance to every point. This distance (or range) matrix will be referred to as $\mathbf{R}$. This is then processed through a function, which computes the SNR for each element, yielding an SNR matrix.

For the computation of SNR, the following power equation is often used to account for space loss [15]:

$$\frac{P_r}{P_t} = D_t D_r (\frac{\lambda}{4\pi d})^2 \qquad (1)$$

where P_r is the power at the receiver in Watts, P_t is the transmitter power, D_t is the transmitter directivity, D_r is the dimensionless receiver directivity, λ is the wavelength in meters, and finally d is the distance between the transmitter and receiver in meters.

This equation has several unknowns, namely the directivities, and therefore a model fitted from experimental data is used [25]. This is seen in Equation 2 below.

$$L_r = k - 20\log_{10}(d) \qquad (2)$$

where L_r is the signal power at the receiver in Decibels (dB), k is the fit constant determined through optimization with experimental data and d is the distance from transmitter to receiver. To elaborate, k consists of the following:

$$k = 10\log_{10}(P_t D_t D_r) + 20\log_{10}(\frac{\lambda}{4\pi}) \qquad (3)$$

This result in Equation 3 is especially important, as the SNR model is fitted using data from conventional ADS-B messages [25]. This k constant is therefore modified to adjust for the wavelength used (which differs from 1090 MHz) and a variety of transmitter power values (which are lower for drones). For the rest of the simulation, it is assumed that the directivities of the antennae D_t and D_r remain the same as those used by traditional Mode S transponders and receivers.

First, observe the SNR plot for the aircraft transponders in Figure 3. Here, a low SNR signal is still present at a 200 kilometer range, with a 20 dB value at around 40 kilometers.

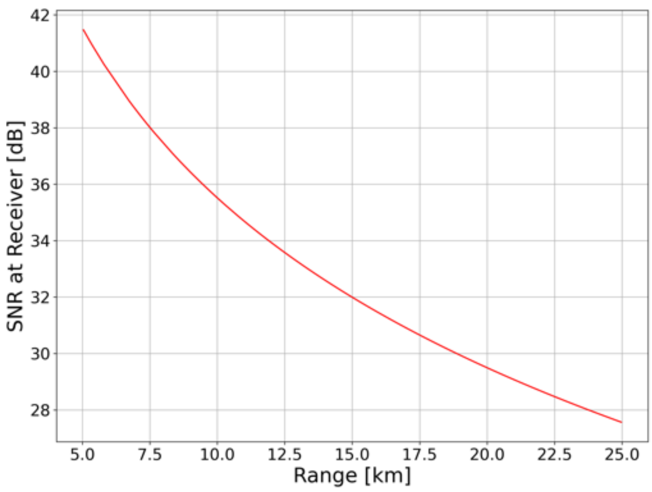


Figure 3: SNR VS Distance to Receiver: Aircraft Transponders

Adjusting the k constant to fit a drone application would require a lower transmitter power and wavelength. Assume for this case a wavelength of 433 MHz (as it is unlicensed and therefore can be used for this application) [6], and therefore a power of 10 mW. The k term is therefore adjusted such that

$$k_d = k - k_{adj} \qquad (4)$$

If the correction term is used, the SNR curve for drones for a 0.1 W transmitter is visualised in blue next alongside the red curve for aircraft transponders in Figure 4.

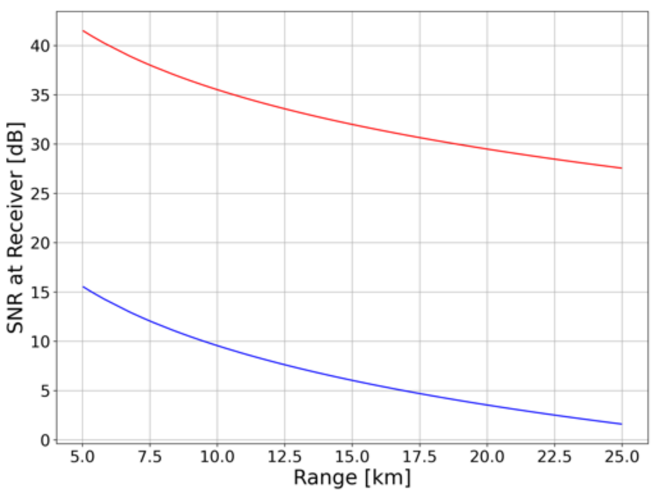


Figure 4: SNR VS Distance Rec.: Aircraft Red, Drone Blue

Note that the fit is not fully representative of the real-world environment: the frequency utilisation scenario is different for 433 MHz, and a different noise floor is therefore likely. For this reason, further analysis of SNR-vs-range for a single drone is flawed.

Now that the SNR-range model is modified, it is possible to use the distance matrix to compute the SNR matrix at every point on the grid for a given drone, as follows:

$$\mathbf{L_r} = k_d - 20\log_{10}(\mathbf{D}) \quad (5)$$

where k_d is the specific fit constant value for the drone in question and $\mathbf{D}$ is the drone distance-to-all-nodes matrix. Now that these power values have been computed, it is possible to evaluate the SNR of a given drone transponder's signal all across the grid. This power matrix can be seen presented as a heatmap in Figure 5.

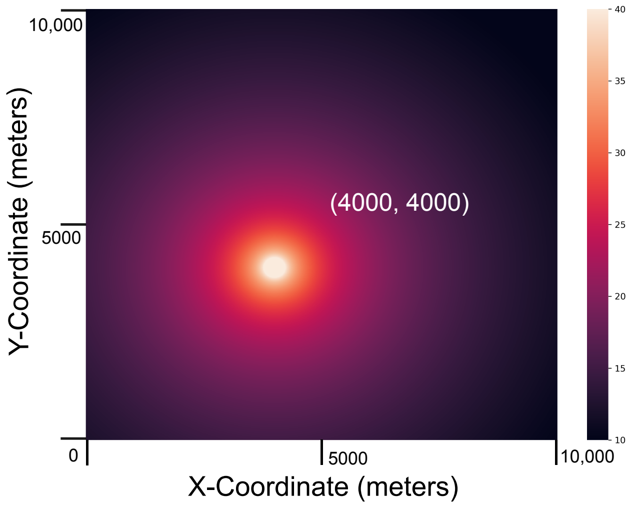


Figure 5: SNR Heatmap for Single Drone

When overlapped, the SNR matrices for two drones can effectively be subtracted from one another. This assumes the same base noise for both messages, since they are transmitted simultaneously and in the same area. This is expressed as:

$$\mathbf{L_\Delta} = \mathbf{L_1} - \mathbf{L_2}[\text{dB}] \quad (6)$$

where $\mathbf{L_\Delta}$ is the SNR matrix for drone 1 with drone 2 transmitting, $\mathbf{L_1}$ is the base SNR matrix for drone 1 and $\mathbf{L_2}$ is the base SNR matrix for drone 2. This originates from:

$$\begin{aligned}\mathbf{L_\Delta} &= 10\log_{10}(\frac{P_1/P_0}{P_2/P_0}) \\ &= 10\log_{10}(\frac{P_1}{P_2}) \\ &= 10\log_{10}(P_1) - 10\log_{10}(P_2) \\ &= \mathbf{L_1} - \mathbf{L_2}\end{aligned}$$

From this, the SNR matrices have been generated, and can now be analysed for a variety of cases.

3.2 *Message Overlap in Time*

Now that the first facet of the simulation is covered, the methodology of message overlap analysis is presented. First, it is necessary to define a typical data message sequence sent out by a transponder. This group of messages (and thus their transponder's output in time) can be modeled through several parameters:

1. Message Duration in Time (t_m [s])
2. Message Start Time (t_s [s])
3. Message Send Frequency (MSF [Hz])

A sequence of messages is seen on the time axis in Figure 6.

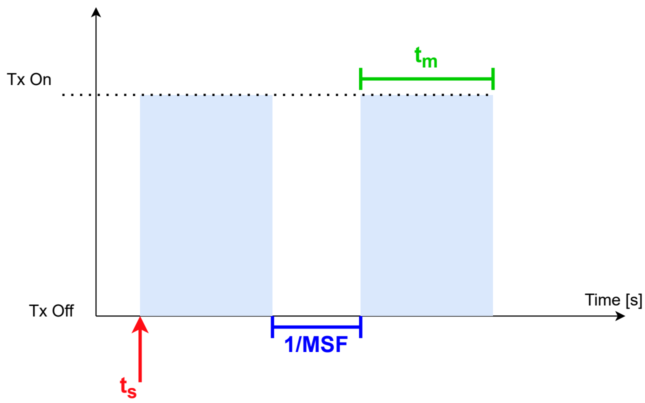


Figure 6: Message Timing: Ideal Case (not to scale)

The light blue peaks are the messages, and the nomenclature corresponds to the parameters shown above. Note that the message duration in time t_m depends on two other parameters, with the relation shown in Equation 7

$$t_m = \frac{l_m[\text{bits}]}{\text{DR}_t[\text{bits/s}]}[\text{s}] \quad (7)$$

where DR_t is the Transponder Data Rate [bits/second] and l_m is the Message Bit Content [bits]. Two sets of messages will overlap when their start times are within one message length of another, or transmitted simultaneously. Whether this will result in a total loss can be investigated using the Spatial Analysis part of the simulation - however, this is decoupled in this work, and all overlapped messages are considered lost. This is not entirely accurate, but is done to reduce complexity and to assume the worst-case scenario.

In a real system, computer clock signals will always experience a small degree of jitter, meaning that the clock edges will be offset from their desired locations [16]. In this case, it was deemed interesting to investigate whether artificially

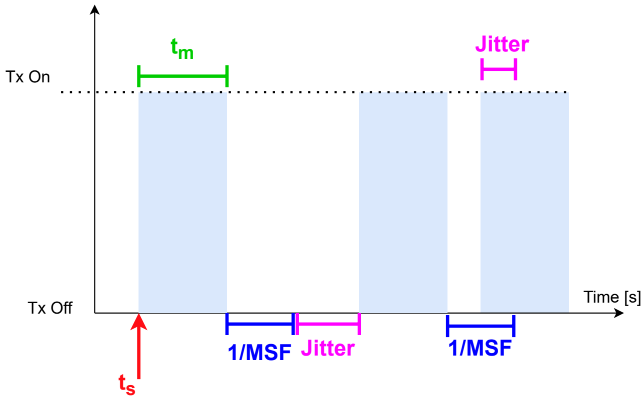


Figure 7: Message Timing: With Jitter

induced message time jitter would be useful to mitigate message overlap in time [28]. Consider the ideal situation in Figure 6 - adding jitter would result in the situation seen in Figure 7.

Note that with the jitter added, the messages are no longer spaced uniformly in time. Also note that the jitter can be negative.

The simulation is initialised by considering a user-defined drone density, message duration, maximum jitter and message repetition frequency. It then computes arrays of every start and end time, using a simple Python function. The jitter is then computed as an array, and added to both the start and end time arrays which are then zipped. With these, list comprehension and loops to check for overlap across all drones' start/end time arrays. The update rates are calculated across the 5-second simulation window, across several runs to ensure that the results are smoothed for the sensitivity study.

3.3 Demonstrator Hardware Selection and Setup

The selection and setup of the hardware used for the demonstration of a basic Drone ADS feasibility proof-of-concept is an important part of the thesis methodology.

The hardware demonstration unit needed to show that an ADS-B like system would be feasible on inexpensive, lower-power hardware. The priorities for the hardware were to be as inexpensive as possible, use the lowest power possible, be easy to set up and to comfortably fit inside any popular consumer drone. There were many options considered [28], but ultimately the choice was to use a microcontroller coupled with a transceiver module. The Pi Pico [9] was therefore selected, and was trialed with several different transceiver modules to find the best working combination. This decision was taken as microcontrollers are closer to lower-level hardware, utilising fewer resources to run an operating system and sensors which are not expressly required for the application. Despite this, it is easy to set up, as the software used (Micropython [5]) is based on Python. The transponder module already handles modulation and simply requires a UART [14] [13] data input to transmit a signal.

The complete package consists of two Pi Pico [9] microcontrollers coupled to Pimoroni Pico Explorer bases [8], two HC-12 [4] transceiver modules, two Micro USB to USB-A cables, 8 jumper cables and two USB-A power sources. The Pimoroni Pico Explorer base is not expressly needed, but provides the Pi Pico with a display and buttons, to make debugging easier in a research scenario. For the Pi Pico to work correctly, the Pimoroni Micropython libraries [7] are installed as shown in the supplied tutorials [8].

After this, a Micropython script was written, which allows for the module to transmit and receive an example signal. This script is found on GitHub [1]. The file which must be saved onto the Pi Pico's internal memory is the main.py script - this allows for the Pico to work when not connected to a computer. To set the unit up correctly, the HC-12 module must be connected to the Transmit (GP4), Receive (GP5), Power (VCC) and Ground (GND) ports of the Explorer base, which allows the transceiver to interface with the Pico. The ports are connected as shown in Table 1. Once the connection is done, the full board is assembled, as seen in Figure 8.

Pi Pico Port	**Designation on Pico**	**transceiver Port**
Tx	GP4	Rx
Rx	GP5	Tx
Power	VCC	VCC
Ground	GND	GND

Table 1: Port Connection Table

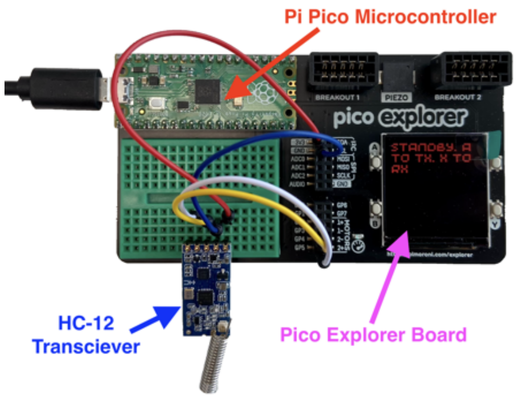


Figure 8: Single Tx/Rx Unit

It may appear counter-intuitive to connect the **Tx** port to the **Rx** port, but this is done in accordance with the UART protocol [14] [13]. Also note that the SET pin on the HC-12 [4] is not used, as the module works out of the box and the parameters are deemed sufficient for the application. Note that

[1] https://github.com/svlaskin/drone_ads_demo

this procedure, from software to wiring, is repeated twice for each of the two sets of components. This means that both the transmitter and receiver boards are interchangeable, because when run, each of the modules can work as a transmitter or receiver.

4 Results and Discussion

Running the simulations, both spatial and time overlap, with different conditions yielded ample data to draw conclusions from. This section will first present the consequences of signal overlap on the SNR matrices of the transmitters involved, followed by the time overlap (garbling) sensitivity analysis.

4.1 Spatial Analysis

The spatial analysis was performed as a means of identifying the resilience of a given transmitter to message overlap with other drones and noise on the same frequency. This is done by evaluating the SNR matrix of each drone using an adjusted model from [25], and simply subtracting one from the other to obtain the relative SNR. Several cases will be considered, to evaluate how the SNR matrices are affected by different scenarios. First, a case with two drones 11,300 meters apart, with 10 mW transmitters each (as per [6], located at (1000,1000) and (9000,9000) respectively, is investigated. The regions of the 10-10 km test grid where the signal exceeds the signal-to-noise ratios (SNR) of 10, 15, and 20 dB are plotted for both clean transmission and when both drones are transmitting simultaneously.

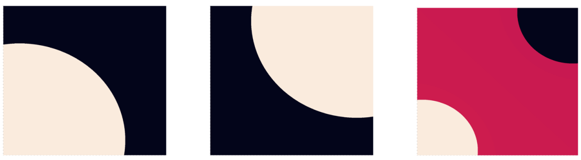


Figure 9: Case 1: 10 dB Threshold

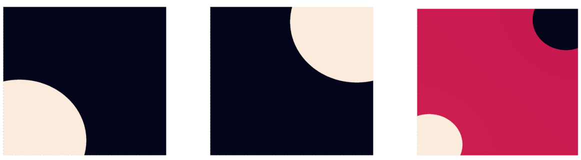


Figure 10: Case 1: 15 dB Threshold

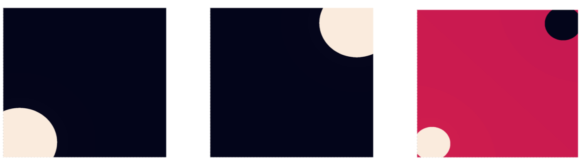


Figure 11: Case 1: 20 dB Threshold

	Threshold	Clean	Overlap
Drone 1 (10 mW)	10 dB	6495.17	2929.70
	15 dB	4098.13	1952.30
	20 dB	2304.54	1255.91
Drone 2 (10 mW)	10 dB	6495.17	2929.70
	15 dB	4098.13	1952.30
	20 dB	2304.54	1255.91

Table 2: Decodable Ranges [m] for Same Power Tx

The figures show the situations for different SNR threshold values. From left to right, the images depict the clean range of the first transmitter at a given threshold, the middle image is akin to the first but only for the second transmitter. The beige zones are the areas with SNR values past the threshold, whereas the black areas are below the threshold. For the rightmost figure, this depicts the ranges of Drone 1 and Drone 2 under overlap - the beige area is Drone 1's compliant zone, the black area is the same for Drone 2 and the red is the dead zone. This is presented as a table in Table 2.

Now, it is worth investigating the effects of a large gap in transmitter power. Assume a 10 mW drone located at (4000,4000) in overlap with a more powerful drone with a 1 W transmitter at (6000,6000). Visualising this can be done in Figure 12 for 10 dB, in Figure 13 for 15 dB and Figure 14 for 20 dB.

Figure 12: Case 2: 10 dB Threshold

Figure 13: Case 2: 15 dB Threshold

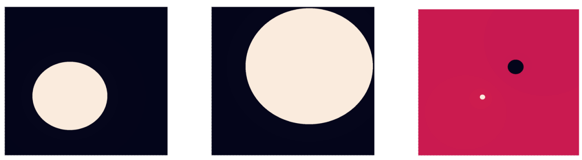


Figure 14: Case 2: 20 dB Threshold

The decodable ranges at different SNR thresholds are seen in Table 3.

	Threshold	Clean	Overlap
Drone 1 (10 mW)	10 dB	7278.56	646.15
	15 dB	4098.17	328.91
	20 dB	2304.57	175.37
Drone 2 (1W)	10 dB	8485.28	3282.29
	15 dB	6963.77	1222.33
	20 dB	3916.06	577.53

Table 3: Decodable Ranges [m] for Large Power Gap

Here, it is seen that for a 20 dB threshold, the range for Drone 1 is dropped from 2.3 kilometers to a mere 175 meters. There are applications where this will suffice but this shows that powerful (such as 1 W) sources on this frequency will cause significant performance degradation to the system if left unchecked. Several other cases were evaluated, but the general outcome is that the results of overlap are expected to be severe in terms of the drone's SNR matrix. The range at which the message can be correctly decoded decreases by up to 13 times for a 1 W noise source at a threshold of 20 dB. This means that overlaps need to be avoided if possible, as the results are severe, especially for drones close to each other, which is likely in a dense urban area.

4.2 Overlap in Time/Garbling Analysis

For the time overlap, the overall goal was to investigate what kind of parameters and drone densities would allow for the nominal update rate to be kept (that is, keep the message overlap in time to a minimum). Running the simulation for a variety of scenarios for a basic set of parameters yielded several absolute error plots in terms of the messages lost.

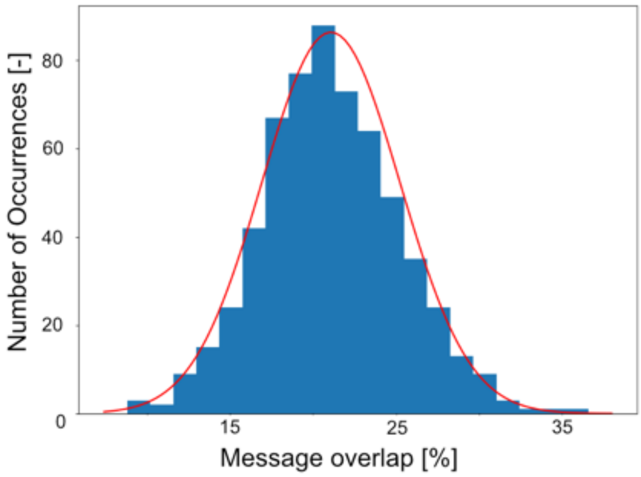


Figure 15: 1.2 ms message, 50 drones, 5x message length max jitter

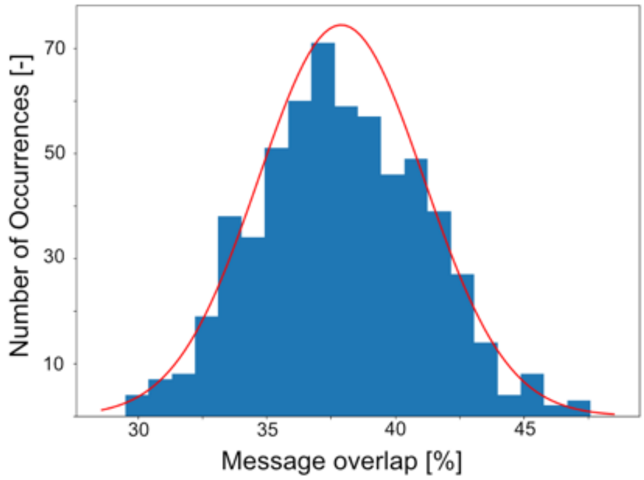


Figure 16: 1.2 ms message, 100 drones, 5x message length max jitter

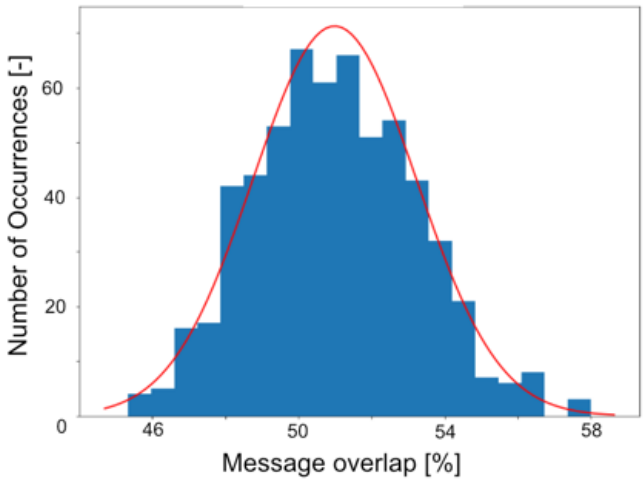


Figure 17: 1.2 ms message, 150 drones, 5x message length max jitter

Note that all of these plots utilise a 1.2 millisecond message time (based on the best-case COTS 100 kbps data rate quoted by SeedStudio Grove Serial RF Pro [3]) and the maximum jitter is kept at 5 times the message period. It is seen that increasing the number of drones at the originally assumed set of parameters yields higher error/messages lost - this is expected, as more drones will lead to a greater number of messages in a given time window. This larger number of messages will increase the probability of two or more of those messages overlapping. Instead of focusing on total error, it is best to focus on maximising a performance metric - the update rate.

The parameters, which are the number of drones in the test area, the message duration and the maximum jitter are now varied to investigate their effects on this performance metric, and whether combinations yielding maximum update rates would be feasible. Firstly, the number of drones is varied with all other parameters constant. This yields Figure 18.

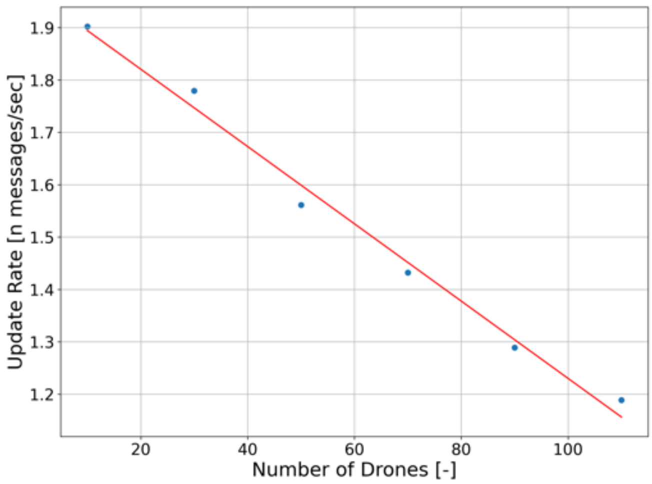


Figure 18: Varying Number of Drones [-] with Max Jitter [x ML] = 0.1 Message Length in Time [s] = 0.0012

A clear linear trend is seen, with the update rate dropping as the drone density is increased. This is the most expected result, with the update rate dropping to below 1.2 updates/second. Now, the message duration will be varied to evaluate its impact on the update rates. The output of this is with a Max Jitter of 0.1 and 110 drones seen in Figure 19. Here, the trend is clear: a longer message decreases the update rate, as the probability of overlap with another message grows. Again, this is a clear linearly decreasing trend.

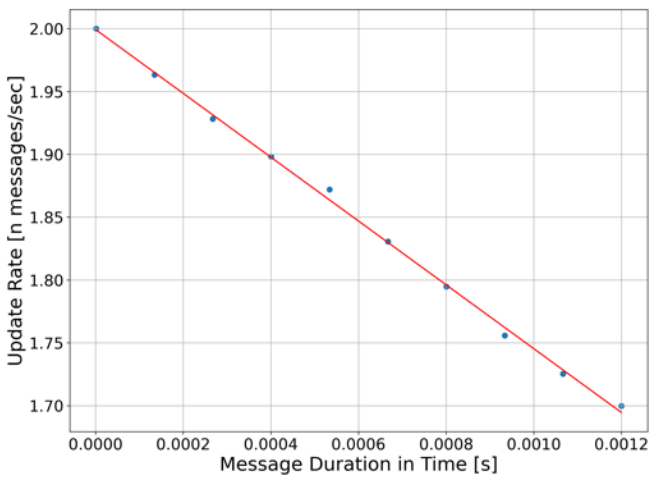


Figure 19: Varying Message Duration with Max Jitter [x ML] = 0.2 Number of Drones = 110

Finally, the jitter is varied. The result of its increase for 110 drones and a 1.2 millisecond message is seen in Figure 20.

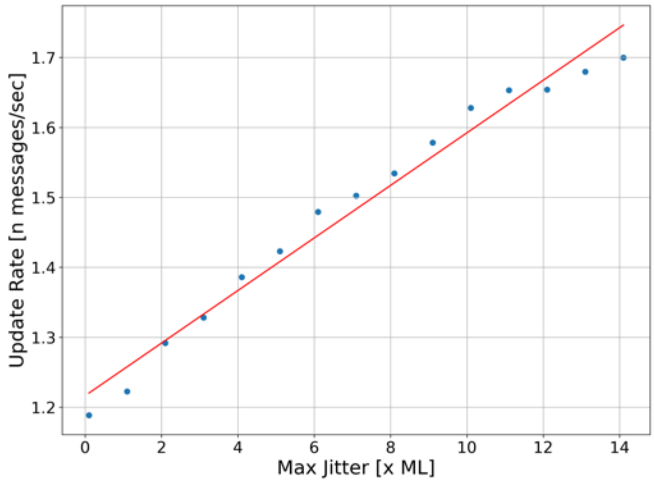


Figure 20: Varying Max Jitter [x ML] with Message Duration in Time [s] = 0.0012 Number of Drones [-] = 110

There is a linearly increasing trend, and shows that an increase in maximum jitter can increase the message update rate, especially for the high-density scenario considered. In this case, it increases up to 1.7 updates/second from a baseline value of 1.2 updates/second in case of low jitter. This mitigation technique is not necessary under low densities, or when the message length is short. However, it does not appear to degrade the update rate significantly for those cases.

4.3 Hardware Demonstrator - Use and Range Test

Now that the hardware is set up, the interest is to use it for basic testing. Here, the range of the system will be tested, by running one module in transmit mode and one in receive mode and monitoring the received messages. Firstly, the use basics will be outlined, followed by the test setup and results.

When a module has power, the screen will display red static text reading: "`STANDBY. A TO TX. X TO RX.`" To enter transmit mode, the user must press button A. The module is transmitting at a 2 Hz message repetition frequency when the screen reads: "`TX ON. B TO STDBY.`" To enter receive mode, the user must press button X. This has 3 cycling screens, namely the Message Receive Indicator (1), Correctly Received Message Rate (2) and Total Correctly Received Messages (3). Screen 1 is displayed by default when no messages are received for a long period of time, and simply reads "`NO MESSAGES RECEIVED`". Screen 2 is the display of the update rate achieved by the system. This is computed with a 5-second rolling window. Finally, screen 3 records the total number of correctly received messages in a session, and simply displays them as an integer.

The range testing took place on campus, with the receiver

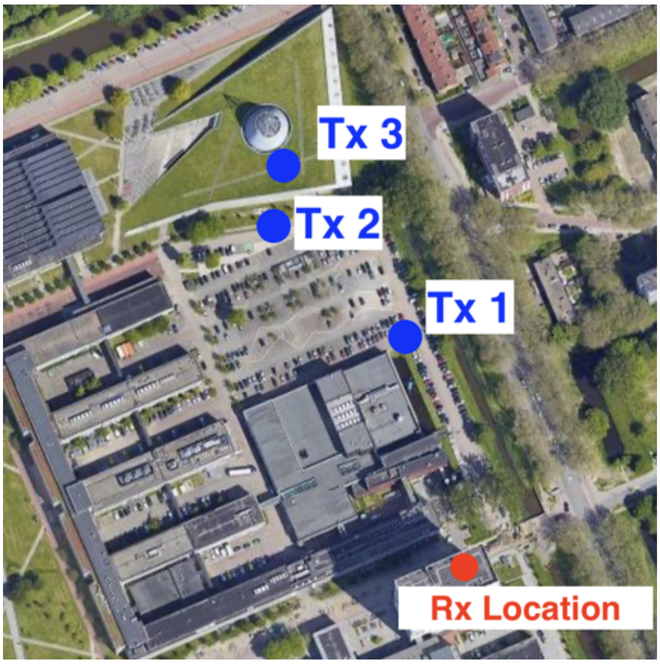


Figure 21: Map of Rx and Tx positions

set up on set up at a height of approximately 65 meters. This height is then factored into the range. The transmitter was moved between several positions, as can be seen in Figure 21.

The transmitter was also moved continuously, however the positions marked on the map (Tx 1, 2 and 3) are of special significance. For these, the results are outlined in Table 4.

	Distance to Rx [meters]	*Update Rate [messages/s]*	*Comments*
Tx 1	**155**	**2.0**	**Fully functional**
Tx 2	**220**	**0.8 to 1.3**	**Reduced capacity**
Tx 3	**240**	**0**	**Unusable**

Table 4: Outcome of Hardware Test

Note here that while several messages were in fact received at ranges of 240 meters, they were so few that the update rate was almost zero on average. The manufacturer claims that the module is capable of 1 kilometer [4], which is not the case. This is also possibly due to noise, but checking the spectrum on a Software-Defined Radio (SDR) [11] yielded no significant noise signals around the 433 MHz frequencies.

5 Recommendations

Now that the results have been presented, the recommendations based on these, as well as on what could not be done, will be outlined and presented. This will be done through the simulation results outcomes (in terms of parameter recommendations and identified constraints) and through recommendations for future research.

5.1 Format Recommendations, Parameter Selection, System Capacity and Technological Constraints

Several key recommendations on format, as well as the system design parameters, can now be made. This is concluded with the identified technological constraints.

For the purpose of this thesis, it is assumed that the 112-bit Mode-S format was retained. While this is likely to be changed for a future implementation, drones will still need to be identified by a unique address. For this identification, it is recommended to keep the 24-bit field, but allocate the address randomly at drone bootup. Given the significantly shorter overall range of the drone ADS system (due to the lower power transmitter), it is unlikely that a signal will be picked up more than 20 kilometers away from it. Considering several hundred concurrent drones in one such zone, the probability of ID intersection is very low with over $2^{24} = 16,777,216$ combinations.

Another recommendation is to adjust the position encoding algorithm slightly from the current Compact Position Reporting (CPR) used in ADS-B [26]. To correctly decode a CPR-formatted position, both an odd and even message are needed [26]. Alternatively, a reference position within the slice where the drone is positioned can be used for decoding. Since most drone transponders will be short-range, it is unlikely that the signal will be arriving at the receiver from a different lat/lon slice and for this reason, it could be safe to assume that the reference position will be the position of the receiver.

The time simulation shows that the main system constraint on performance is the message duration. In turn, this arises from the bit length and data rate. While it is possible that the message bit length could be reduced overall if the format is to be altered in the future, this paper assumes the worst case scenario, whereby all 112 bits of data (and 8 preamble) are kept, and the message is 120 bits long in total. Since this is kept constant, the main constraint becomes the Data Rate $\mathbf{DR_t}$ - it dictates the message duration in time as per Equation 7 and thereby limits performance as seen in the time simulation results. It is strongly recommended to use a transmitter superior to that considered in [28] or the selected HC-12 [4] to achieve a higher data rate - else, high-density scenarios see a performance degradation such that the desired 2 Hz update rate is not matched.

Transmitter power regulations for these drones stipulate a maximum of 10 mW [6]. It is recommended that the manufacturers select antennae of similar power - else, the message overlap impacts the lower-power drone with greater severity, as seen in Figure 14.

It is also recommended to do a detailed analysis of the RF spectrum around the useful 433 MHz bands, as this was out of the scope of the thesis. A 1W 433 MHz source has the potential to reduce the range of a 10 mW transmitter by 13 times for a 20 dB decoding threshold, as seen in Figure 14. This is a severe performance issue, and cannot be allowed to occur if the system is to be deployed. It is therefore recommended to review and alter the requirements for unlicensed 433 MHz bands (or alternative) usage if such a frequency is used for a drone ADS system.

Jitter with regards to the message start times is inevitable in a real electronic system. However, the sensitivity study shows that varying message start times by adding random bounded jitter can help reduce message garbling and increase the update rate in high-density scenarios. It is recommended that high maximum jitter values beyond 14 times the message duration be considered for such a system, as they will prevent total message loss in very high-density areas.

5.2 *Recommendations for Future Research*

Given the time constraints of this project, several avenues of research remain unexplored and can be improved upon. There are many of these, as the road towards the complete design and implementation of an ADS system for drones is just beginning.

The first of these is an improved SNR-distance model for small 433 MHz transmitters. While studies on 433 MHz channel occupancies exist, the resultant model is heavily dependent on the location of the measurement equipment and its properties, and it would be beneficial to have an idea of measured noise and channel occupancy for this frequency around the Delft/The Hague (or other) metropolitan area.

The second point of improvement would be an in-depth statistical analysis of the simulation output, along with an enhanced sensitivity study. With more simulation runs, less error-prone data can be generated, and from this, a probabilistic model for un-garbled signal reception can be constructed. The simulation time window could also be increased beyond 5 seconds in future research.

This work largely assumed that most of the ADS-B and Mode S message format characteristics would carry over to the drone ADS system, with some changes in the message content. However, there is a case to be made for this format to be developed from scratch. While keeping the Mode S format is simple, and may make the system easier to integrate with current ATM, it would not be tailored to the specific needs of drones. Therefore, a drone ADS format definition can be focused upon in future work and designed from scratch, taking into account the specific needs and applications of the system. The modulation method and carrier frequency can thereby also be selected specifically for this new format, for increased error resistance and efficiency.

Finally, the test hardware presented in this paper demonstrates that it is indeed possible for a simple COTS system to transmit and receive messages for a drone ADS system. However, decoding is left out, as is the design of a task-optimised transponder component. Adding a decoding component, as well as designing an optimised transponder module could be options for further development. The boards have the capability to be linked to sensors - it would also be worthwhile to investigate whether a pressure sensor or GNSS module could be used to fully simulate an ADS system by compiling and sending real location or altitude messages. Ultimately, such a system could be fitted on a real drone - for this, a lot of further development is required.

6 Conclusion

This paper demonstrates that the implementation of an Automatic Dependent Surveillance - Broadcast system is possible for the average consumer drone. This requires a reduction in transmitter power from what is seen on aircraft transponders, which means that the message detection range drops from 200 kilometers to values below 10 kilometers. Furthermore, the carrier frequency is changed from 1090 MHz - high-density drone usage is not realistically feasible without affecting current 1090 MHz applications, which are already seeing congestion due to overuse. A simulation was constructed, which tested the garbling severity with a spatial simulation and the extent of signal overlap and garbling given different system parameters.

For the spatial simulation, the penalty of message overlap is found to be large, as expected. While two equivalent transmitters at distances of 11 km only reduce the 10 dB ranges from 7 km to 3 for a 10 mW transmitter, the result is worse for a 20 dB threshold, with a reduction from 2.3 km to 1.2 km. It remains to be seen whether this is an acceptable consequence - if there is a large number of receivers placed in urban infrastructure, the sub-1000-meter range could prove to be acceptable.

The sensitivity and capacity analysis showed that the message duration is the parameter that constrains performance in high-density usage. To combat this, transceivers with improved data rates can be used in the future. It was also found that message start time jitter can be used to mitigate the amount of signal garbling and increase the average ADS update rates for every drone. On average, a value of 14 times the message duration or above is recommended, provided the message duration is short enough such that the jitter never exceeds the message time between two message transmissions to avoid overlap into the next transmission window. Finally, the hardware demo shows that building such a system can be done on a low budget with COTS components. The transceiver module used worked correctly up to a range of 200 meters, compared to the 1 kilometer stated in the specifications.

Research into this topic is only beginning, and this paper has aimed to provide some stepping stones for the next phases that follow. Further work is recommended on the format definition, transponder design, frequency selection and simulation refinement.

References

[1] Mode S downlink aircraft parameters implementation and operations guidance document edition 2.0. https://www.icao.int/APAC/Documents/edocs/APX.%20N%20-%20Mode%20S%20DAPsImplementation%20and%20Operation%20GuidanceDocument2.0-20200513.pdf.

[2] Uk airprox: Airprox involving uas drones. URL https://www.airproxboard.org.uk/Reports-and-analysis/Statistics/Airprox-involving-UAS-Drones/#collapse-airproxreportsandanalysis.

[3] SeedStudio Grove Serial RF Pro. https://wiki.seeedstudio.com/Grove-Serial_RF_Pro/. Accessed: 09-06-2022.

[4] HC-12 Datasheet, hc01.com. http://www.hc01.com/downloads/HC-12%20english%20datasheets.pdf. Accessed: 09-06-2022.

[5] Micropython.org Website. https://micropython.org. Accessed: 09-06-2022.

[6] Regeling gebruik van frequentieruimte zonder vergunning en zonder meldingsplicht 2015 overheid.nl. https://wetten.overheid.nl/BWBR0036378/2016-12-28. Accessed: 09-06-2022.

[7] Pimoroni Pico Libraries: GitHub. https://github.com/pimoroni/pimoroni-pico, . Accessed: 09-06-2022.

[8] Pimoroni Pico Explorer Base manufacturer website. https://shop.pimoroni.com/products/pico-explorer-base?variant=32369514315859, . Accessed: 09-06-2022.

[9] Raspberry Pi Pico manufacturer website. https://www.raspberrypi.com/products/raspberry-pi-pico/. Accessed: 09-06-2022.

[10] Drone Detection Radar, robin. https://www.robinradar.com/drone-detection-radar?hsa_acc=8561802120&hsa_cam=12617816363&hsa_grp=115147506610&hsa_ad=551234304231&hsa_src=g&hsa_tgt=kwd-578526307915&hsa_kw=radar+for+drone+detection&hsa_mt=p&hsa_net=adwords&hsa_ver=3. Accessed: 09-06-2022.

[11] RTL-SDR.COM. https://www.rtl-sdr.com. Accessed: 09-06-2022.

[12] SESAR Joint Undertaking website. https://www.sesarju.eu. Accessed: 09-06-2022.

[13] UART: A Hardware Communication Protocol. understanding universal asynchronous receiver/transmitter. https://www.analog.com/media/en/analog-dialogue/volume-54/number-4/uart-a-hardware-communication-protocol.pdf, . Accessed: 09-06-2022.

[14] class UART – duplex serial communication bus documentation from micropython.org. https://docs.micropython.org/en/latest/library/machine.UART.html, . Accessed: 09-06-2022.

[15] Daniele Cavallo. *Guided and Wireless EM Transfer (EE3120TU) Lectures*. TU Delft, 2018.

[16] Douglas E. Comer. *Computer Networks and Internets*. Pearson, 6th edition, 2014. ISBN 0133587932.

[17] Michael Guterres, Stanley Jones, Greg Orrell, and Robert Strain. *ADS-B Surveillance System Performance With Small UAS at Low Altitudes*. doi: 10.2514/6.2017-1154. URL https://arc.aiaa.org/doi/abs/10.2514/6.2017-1154.

[18] Petr Jonas, M. Jancik, S. Holoda, and J. Bodart. Impact of SUAS equipped with ADS-B on 1090 MHz environment. pages 63–67, 2020. doi: 10.23919/NTCA50409.2020.9291095.

[19] Kaduk Aguilar, Borin. Analysis and optimisation of radio spectrum pollution on 1030/1090 MHz bands associated with mode S transponders.

[20] J. Hoekstra M. Doole and M. Ellebroek. *Drone Delivery: Urban airspace traffic density estimation*, 2018.

[21] SESAR. *Ground Based Technologies For A Real-Time Unmanned Aerial System Traffic Management System (UTMS) - CLASS*. . URL https://www.sesarju.eu/projects/class.

[22] SESAR. *CLASS - Report Document Repository*. . URL https://cordis.europa.eu/project/id/763719/results.

[23] SESAR. *SESAR: European Drones Outlook Study*. . URL https://www.sesarju.eu/sites/default/files/documents/reports/European_Drones_Outlook_Study_2016.pdf.

[24] SESAR. *SESAR Smart ATM - U-Space*. . URL https://www.sesarju.eu/U-space.

[25] J. Sun and J. M. Hoekstra. *Analyzing Aircraft Surveillance Signal Quality at the 1090 Megahertz Radio Frequency*, 2020.

[26] Junzi Sun. *The 1090 Megahertz Riddle: A Guide to Decoding Mode S and ADS-B Signals*. TU Delft OPEN Publishing, 2 edition, 2021. ISBN 978-94-6366-402-8. doi: 10.34641/mg.11.

[27] Sasha Vlaskin. *Project Plan: Designing an Automatic Dependent Surveillance System for Drones*. 2021.

[28] Sasha Vlaskin. *Midterm: Designing an Automatic Dependent Surveillance System for Drones*. 2022.

Part II

Scientific Paper Appendices

Appendix A: Assumptions Used

This appendix lists the assumptions used that are relevant to the work presented in the scientific paper.

Geometric and Topological Assumptions

First, it is important to note that since the operational range of the system is low (transciever power limitation), only a 10-by-10 kilometer grid is assumed for the spatial simulation. For this reason, the Earth is assumed to have no curvature over this area (Flat Earth assumption). This assumption is not expected to have a large effect, due to negligible curvature over such a short distance.

Next, the study disregards the city topology, and effects that urban environments will have on the ADS signal. As such a signal is line-of-sight, it is expected that if drones are to be operated at street level, the effective ranges will decrease due to obstruction caused by buildings.

Required Simulation Time Assumptions

The simulation time window is a parameter that is assumed. The value is taken to be 5 seconds, leading to 10 messages per drone at an ideal update rate of 2 Hz.

This is deemed sufficient, but for significantly larger jitter values ranging beyond 15 times the message period (with are not investigated), this value may need to be increased to provide reliable data. The reason for it being kept at 5 seconds is due to a trade-off between computer memory constraints, computation time and a longer, more representative period. This is likely what is yielding the higher variance in the results for the average error due to overlap, espacially for lower drone densities as seen in Figure 15. Increasing this is expected to normalize the average error across runs.

Signal Parameter Assumptions

For simplification purposes, the simulation assumes that a message is lost in overlap if two drones transmit simultaneously, regardless of their relative positions. The spatial analysis showed that in reality, drones further apart may still preserve over a kilometer of decodable range. However, this study takes the pessimistic approach, assuming the message to be completely lost (as the ranges are impacted very significantly even in the best-case scenario, and the decoding threshold Signal-to-Noise ratio is not known).
This assumption will cause the Update Rate to be lower than in reality, especially when drones are far apart. The effects will be less pronounced in cases where drones are close together, which is more likely for high-density urban environments.

Another assumption made is the fact that multipath effects are negletcted - that is, any signal that is reflected off a surface is considered to be lost. This assumption is done to simplify the work and results from a lack of knowledge about the signal power characteristics. It is expected that a real scenario would see these multipath effects factor into the garbling and would cause some messages to be lost. This effect would be most prevalent in cities, especially below building height. Given the low power of the transmitters used, the overal impact is nevertheless predicted to be minimal.

Drone Position Assumptions

Initially, the drone positions were going to be factored into the signal overlap analysis in time, as mentioned in the previous assumption. However, this is not considered as the worst case (pessimistic) scenario is the best assumption for this stage and helps reduce simulation complexity. Therefore, the drone positions are assumed not to vary over the 5 second simulation period. Given that a drone such as the Mavic 3 (https://www.dji.com/nl/mavic-3/specs) will see a maximum speed of approximately 19 m/s, the maximum positional change will be of 100 meters. This is not significant when the assumption of garbling across the grid is made, because position does not change the outcome of overlap in time.

However, there are cases for which this may make a difference. This could be the case when two drones are in very close proximity to each other at the start of the window, and fly in opposite directions - 200 meters of separation could yield a usable signal at very short ranges when transmitting simultaneously. The inverse case would also hold as an exception. Nevertheless, this is not expected to have a major effect on the simulation results.

External Influence Assumption

It is also assumed that there are no other signal sources of note operating on the frequencies used. For the unlicensed frequencies in question, especially the 433 MHz bands, there are a multitude of users such as remote keys for cars, home automation systems, and wireless sensors. It is important to note that these operate at low powers, utilise a low duty cycle and are therefore not expected to pose a major problem. However, the emergence of technologies such as LoRa may cause problems in the future and should be considered once the impact of this technology is known.

The second assumption regarding external influence pertains to drones outside of the 10-10 km control area. It is assumed for the sake of the experiment that signals from drones outside the area are not allowed to permeate into it. This may appear to be a big flaw, but considering that the 10-10 km area represents a large metropolitan zone surrounded by less dense traffic, this assumption will hold. For areas of uniform density past the 100 squared kilometer area, different assumptions will need to be made.

Appendix B: Practical Tips and Tricks

There are several tips for further work with the hardware demo. These are divided into the setup and usage tips, the areas to develop upon, and future testing.

Setup and Usage

Using the two modules as a transmitter and receiver is simple - only a few button presses are needed. In order to get to that point, all components need to be connected.

First, the Pi Pico needs to be connected to the board as shown in the Pimoroni Pico Explorer manual (https://shop.pimoroni.com/products/pico-explorer-base?variant=32369514315859). Then, the Micropython libraries are installed as described in https://github.com/pimoroni/pimoroni-pico. Finally, the 'main.py' file from GitHub https://github.com/svlaskin/drone_ads_demo is flashed onto the onboard memory while holding the 'BOOTSEL' Button.

Once this is done, a full instruction video on how to connect everything correctly can be seen in https://youtu.be/4d_a8oD5VJM. In short, the VCC goes to VCC, the GND to GND (as expected), but the GP4 (Tx) pin is connected to the Rx pin on the transceiver and the GP5 (Rx) is connected to the Tx pin on the transceiver.

Once the connections are done, the module can be plugged in and the mode is selected using the buttons. The connection between the transceiver module and the pins can be fickle, as soldering has not been done for easy disassembly and transport. It is therefore recommended to either solder this, or apply pressure to the end of the transceiver such that the pins are touching the contacts.

At the time of writing, it is not possible to change the mode from transmitter to receiver and vice versa while the module is operational. Therefore, to restart, simply unplug the module from power.

Areas for Further Development

The large Explorer base, containing the screen and breadboard, has been used for quick prototyping. For a final proof of concept, the size of both the transmitter and receiver module can be drastically decreased by simply removing this component and keeping only the Pi Pico and transceiver module. To do this, however, the main file must be altered. It is recommended to re-install Micropython without the Explorer board-specific packages. Also, it will be necessary to rewrite the main.py file such that it no longer relies on button presses from the said board. The most important lines to keep will be those containing the UART read and listen sequences, and one of the modules will need to be connected to a computer to obtain data about incoming signals. The file can be found at https://github.com/svlaskin/drone_ads_demo.

Future Testing

Due to time constraints, only a basic range test was performed with the module. Other interesting areas to explore in further work are discussed here.

Firstly, an overlap situation could be forced with 2 transmitters and one receiver, to check the effects. This would require a third such module to be added to the setup. Then, overlap could be forced with two transmitters by continuously transmitting the signal. The model for range impact on decoding can be validated.

Another test could be to integrate a GNSS module or other sensors to the Pi Pico and attempt to transmit live sensor data in the message. This should be feasible via the simple UART input on the Pico. This could also be wired to a moving drone

A different type of antenna could be used on the HC-12 transceiver - it has a coaxial antenna port, which can be used for this purpose. This might be especially useful for the receiver end of the system.

While the system's transmitter was picked up using a Software-Defined Radio (SDR), decoding was a different issue and was left to the second module. It would be of interest to consider creating a tool that would allow for the Pico signals to be decoded by an SDR such that the Signal-to-Noise Ratio required to decode them correctly can be found experimentally.

Finally, the transceiver module used is the lower-performance pick of the two modules tested. With a 5V supply, it might also be possible to use the Grove Serial RF Pro module, which has a better theoretical range. However, the Raspberry Pi has a 3.3 V power output, so the power source would have to be external.

Appendix C: Hardware Demo Figures

Hardware Demo: RTL-SDR outputs

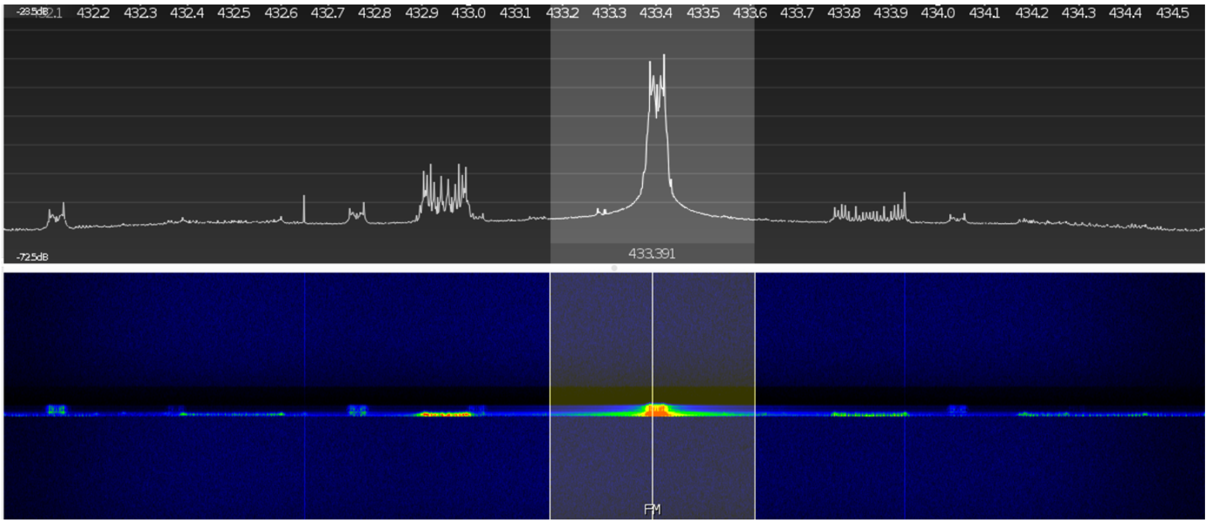


Figure 1: Frequency Domain Output of Demo Signal on RTL-SDR [Cubic SDR]. The peak visible at 433.4 MHz corresponds to the Drone ADS demo's signal, sent out by the HC-12 transciever module.

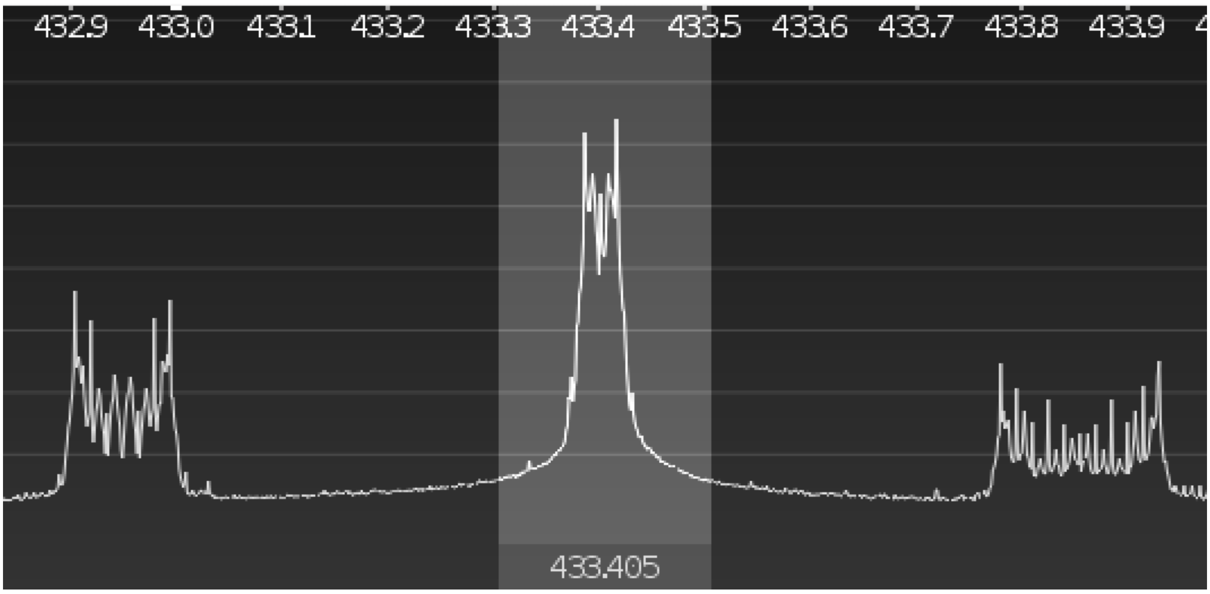


Figure 2: Zoomed-in picture of the frequency domain peak for the 433.4 MHz frequency. The SNR here is read to be -23 dB. Some sidelobes are present around the 433 and 434 frequencies. Disconnecting the supplied antenna yields a difference of 1-2 dB.

Hardware Demo: Screens in Operation

The hardware demo has several display states based on operational status. These will be outlined here.

Figure 3: Standby state for the module. The A button will trigger Transmit, X (behind green tab) will trigger receive.

Figure 4: Transmit state. Nothing further is displayed.

Transmit state: entered with 'A' button from standby

Figure 5: Receive state, no messages received. This screen appears when no message is received for more than 5 consecutive seconds.

Figure 6: Receive state - rolling window update rate. This is the number of messages corrrectly received per second averaged over a 5-second interval

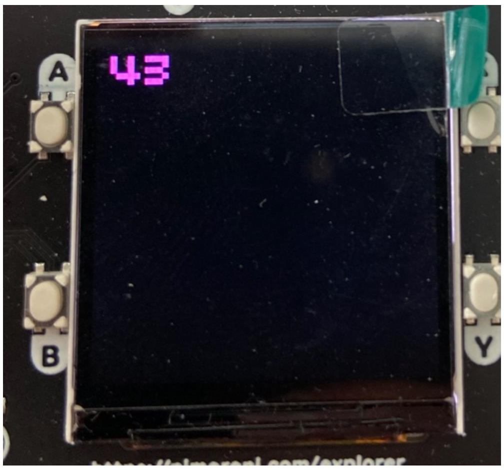


Figure 7: Receive state - total number of correctly received messages for current session.

Receive state: entered with 'X' button from standby

Part III
Preliminary Report

Preliminary Report

Sasha Vlaskin

June 28, 2022

1. Summary

In this chapter, the report is summarised for readers with little time. First, the underlying motivation for the thesis work is presented. Second, the updated project goals are given. Third, the preliminary results are presented. Lastly, the summary concludes with the next steps to be taken in the project.

1.1. Motivation for Thesis Work

It is projected that the consumer and commercial drone sectors will grow rapidly in the coming decades. Potentially, they will employ in excess of 100,000 people and may see 7 million leisure and 400,000 commercial drones by 2050 [12] in European airspace alone, with many flying beyond line of sight. Many different systems have been proposed to tackle this problem, both passive (such as primary radar or LIDAR) and active (co-operative systems for which the drone is interrogated). Currently, several drone radar systems are operational. However, they are entirely passive, and allow for drone detection, classification and tracking. For applications such as automatic separation, an ADS-B like system that allows for the drone to broadcast information from its own sensors would be highly beneficial. In fact, the objectives cited under SESAR's U-Space [13] are realistically only achievable if such a system is implemented.

1.2. Updated Project Goals

Now that a significant amount of time has been spent on this project, several developments have led to the re-calibration of the research question and goals. Originally, the aim of the project was to utilise a system design approach to come up with a Drone ADS system. However, the focus has now been shifted to providing design recommendations, relying on a simulation to evaluate the system capacity constraints and sensitivity analysis. The research objective has therefore been shifted to:**"Perform a capacity and feasibility study on a potential Automatic Dependent Surveillance - Broadcast system for drones."**

1.3. Preliminary Results

The preliminary results have been promising. So far, a simulation script has been built in Python, and has provided a 5 second simulation of a running ADS for drones system on a 10-10 kilometer grid with 100 m resolution. The aims are to:

- Evaluate the impact of messages overlapping on the signal-to-noise ratio (SNR) versus the distance from the transmitting drone
- Evaluate the extent of message overlap for different scenarios (number of drones in the defined grid)

This has been successful, and preliminary conclusions are made:

- In order to ensure that a drone transmitter's range is approximately 10 km, a power of 0.5 Watts must be used and the receivers must be able to decode messages of signal-to-noise ratios of at worst 10 dB
- In case of message overlap, the available signal-to-noise ratios decrease dramatically. However, messages may still be recoverable if the receiver is within 2.5 kilometers of the drone. Whether this is to have any impact on real update rate performance remains to be seen.

1.4. Next Steps

The next steps of the project will include:

- Expanding the range of simulation scenarios
- Checking statistical significance of data and running longer simulations
- Finalised recommendations and example system
- On-hardware verification and feasibility check for a low-power commercial-off-the-shelf implementation of the proposed system

2. Introduction and Research Objectives

The purpose of this report will be to combine the Literature Study, Project Plan and an overview of the work done and methods used.

2.1. Report Structure

Firstly, the literature study segment will cover all relevant branches of literature identified, citing work that could be of use to the thesis. Then, the envisioned planning will be presented. This is followed by the state of the art literature in the domain, as well as the programming language selection and setup description. The hardware setup possibilities for the verification will then be discussed, followed by the constraints defined through theory found in the literature study. The work done and preliminary recommendations are next, followed by the reformulated research goal. The report closes on the preliminary simulation results, further work and conclusion.

2.2. Introduction to Thesis Topic

The European Council predicts that the drone sector will grow significantly in the coming years. It is predicted that the sector will [2]:

1. Directly employ more than 100,000 people
2. Have an economic impact exceeding €10 billion per year, mainly in services
3. Consist of 400,000 commercial and 7 million leisure drones by 2050 [12]

Many applications, such as aerial videography, rescue coordination, parcel delivery and more could be handled by these drones. However, a major hurdle remains - drones cannot operate near controlled airspace due to their lack of visibility to ATM and pilots. Traditional aircraft rely on surveillance systems, such as Automatic Dependent Surveillance - Broadcast (ADS-B) to provide information about themselves to Air Traffic Control (ATC). Alternatively, they are interrogated by Secondary Surveillance Radar and thus can be guided and separated from other aircraft by ATC. This technology is enabled through the use of a Transponder (Transmitter-responder), a piece of hardware which receives a signal from ATC and responds automatically with the required information.

2.3. Research Objectives

To solve this issue, the thesis topic in question proposes to design a system which will broadcast relevant information about the drone to ATM and other aircraft. The goal of such a system will be to enhance drone surveillance and therefore safety. The research question is phrased as:

"Is it possible to design an ADS-B compatible ADS system for drones?"

This can be seen in more detail in the Project Plan for this report [18] - there, the sub-questions are also outlined and justified. From this, the research objective was proposed to be: "Design and test an ADS system for drones through system design, simulation and hardware verification".

This has been done to an extent, but relies heavily on assumptions, and has been proven to be more challenging than expected. Unlike a preliminary design of a standard aircraft or automobile, there exist no equations for "sizing" the ideal data format. For this reason, and given that regulating bodies such as ICAO will likely devise the format as a part of the U-Space initiative, a set of recommendations for the specification and format was deemed more reasonable. As expected, the focus of the project has shifted to a domain that is deemed more useful to investigate - the time/space simulation of such a system in action. The main outputs of this simulation are to be the spatial consequences of message overlap in time (namely, the Signal-to-noise ratio against range to the drone at the receiver) and the extent of the said message overlap in different drone density scenarios, which is evaluated as the proportion of messages lost.

For these reasons, it is best to reformulate the research question as: **"Perform a capacity and feasibility study on a potential Automatic Dependent Surveillance - Broadcast system for drones."** A secondary objective becomes: **"Evaluate the options for an ADS system for drones in terms of format, signal parameters and capacity"**. The thesis will therefore have two main deliverables - the capacity study and design constraints and recommendations. The capacity study is the focus of the project, and aims to inform the parties in charge of implementing such a system. The design constraints and recommendations will present an example system, and outline the limitations in terms of design parameters such as the carrier frequency and transmitter power. Once these main deliverables are complete, the final research objective becomes: "Test the feasibility of such a system on hardware". This is unchanged from the initial expectations.

3. Literature Review

3.1. Literature Categories

Before the literature review can be started, it is important to mention the domains of relevant literature that were identified. These categories include General Air Traffic Management, ADS-B and Electromagnetic Wave/Digital Communication Systems Theory, Drone-specific Automatic Dependent Surveillance - Broadcast (ADS-B) Implementation Literature, Literature About Other Types of Drone Surveillance Systems. Finally, ongoing projects related to the thesis topic, and those that do not have readily available detailed information will be listed and discussed.

3.1.1 General Air Traffic Management, ADS-B and Electromagnetic Wave/Digital Communication Systems Theory

Firstly, we have General Air Traffic Management and Automatic Dependent Surveillance - Broadcast (ADS-B) information. Background information about Air Traffic Management is found in the Air Traffic Management course slides [6]. This allows for a general overview of the concepts that define the current state of aircraft surveillance. For ADS-B and Mode S, the book entitled The 1090 MHz Riddle by Dr. Junzi Sun [17] was used, as it provides a comprehensive and detailed overview of the standard. As a fallback, the standard ICAO documentation can be used [1] [7].

For Electromagnetic Wave Theory and Digital Communication Systems Theory, the sources of interest are [4] and [8]. These provide a background for the development of the drone ADS standard as they offer theoretical background which will aid with the selection of the system design parameters such as the carrier frequency and the signal modulation method.

3.1.2 Drone-specific Automatic Dependent Surveillance - Broadcast (ADS-B) Implementation Literature

The specific literature type is useful, as it provides close insight into the research topic at hand. The relevant sources are found under [5] and [9]. From these studies, the main learning point is the impact of the direct adoption of current ADS-B technology for drone use.

In [9] for instance, a simulation is created to assess the impact of direct ADS-B adoption on Mode S performance for other airspace users. Multiple scenarios are designed, with different drone transmitter power values and drone densities per unit area.

3.1.3 Literature about other Types of Surveillance Systems for Drones

Finally, another category of importance is literature about other proposed surveillance systems for drones. These systems will be used as a benchmark for the developed ADS system, as its utility must be justified compared to them.

The research performed on behalf of the Single European Sky ATM Research (SESAR) is perhaps the most complete in terms of proposed surveillance network for drone use. An example is CLASS [10], which uses sensor fusion between holographic radar and a proprietary cooperative system to track drones.

3.2. Theoretical Background

In this section, the theoretical background obtained from the literature will be summarised and presented. First, the basic theory behind aircraft surveillance will be summarised, followed by a summary of Mode S theory and one of its services on which the report will focus: ADS-B.

3.2.1 Surveillance Basics

In today's aviation domain, aircraft large and small rely on instruments, both internal and external, to navigate safely through airspace. Surveillance is a term referring to the ability of Air traffic control (ATC) and airspace users to locate aircraft, track their movement and predict what they will do next.

This surveillance is acheived through the use of Radio Detection and Ranging (Radar). There are two main classes of radar - Primary Surveillance Radar (PSR) and Secondary Surveillance Radar (SSR). PSR uses electromagnetic pulses to detect aircraft - in short, it emits an electromagnetic pulse which is reflected by the airframe, and using the computed time delay between the emission and reception of the pulse, the range to the aircraft is computed. This type of surveillance is dubbed as passive 'passive', as the aircraft itself does not emit anything or contribute to the process.

While PSR has been widely used in the past and still features for some use cases, current aviation relies on SSR for surveillance tasks. This is thanks to several key advantages which it provides over PSR. SSR also relies on electromagnetic waves as its communication medium, however they carry significantly more information than PSR. This type of radar is capable of sending encoded data to aircraft, and receiving responses, which is why it is referred to as cooperative. The radar can interrogate aircraft for a variety of data, such as their airspeed, position, identification, status, vertical intent and more. The aircraft replies with the relevant data, which then allows air traffic controllers to have an accurate overview of the traffic situation. The basic overview of how an SSR works is provided in Figure 1.

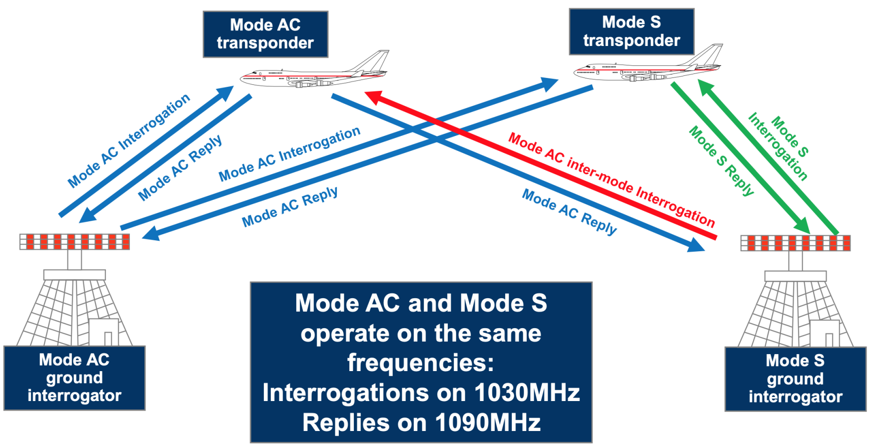


Figure 1: SSR Functional Principle [3]

As mentioned, this is the preferred system for modern air traffic control, as it proposes many advantages over PSR. These include a smaller antenna and lower transmission power - the signal need not account for the return journey in terms of space loss, as the aircraft emits the reply signal using its onboard transponder. Another key advantage is the greater abundance of information that this system provides, as well as the aircraft's ability to transmit data obtained from a variety of sources and sensors (such as more precise position data from satellite navigation systems). However, an additional hardware component is necessary: the transponder. The aircraft needs such a transponder to be able to receive, decode, process, encode and reply to interrogations. The block diagram in Figure 2 shows how an airplane can respond to an interrogation from an SSR. The (mod) and (demod) refer to modulation - that is the process of adding data to a carrier (radio) wave in order to transmit it and vice-versa.

As can be seen, the SSR is the block on the bottom of the figure. It codes, modulates and transmits the necessary message to the aircraft. The message is picked up via the on-

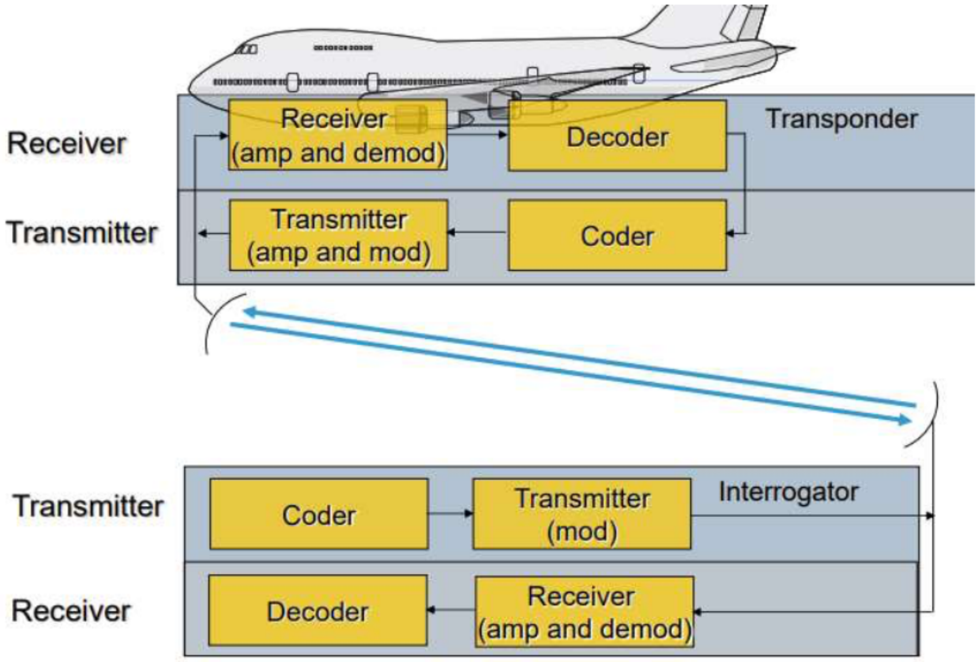


Figure 2: Block Diagram of full SSR system [3]

board antennae, amplified and demodulated via the receiver, decoded and then the relevant data needed for the response is coded, modulated, amplified and transmitted via the same antennae. The interrogator (SSR) receives this reply, amplifies and demodulates, then decodes the information, which is then accessible to ATC or other surveillance users.

Secondary surveillance radar relies on several protocols, with their own formats and capabilities. Legacy transponders use simper Mode A and C standards. These However, governments have mandated the more capable and complete Mode S standard, the principles of which are covered in the next section.

3.2.2 Mode S

Mode S, or Mode Select Beacon System, is characterised by its ability to interrogate aircraft selectively. Interrogation messages are sent over the 1030 MHz band, and use Differential Phase-Shift Keying (DPSK) as a modulation method. A short interrogation contains 56 bits of data, while the long variant uses 112 bits. Replies are broadcast over the 1090 MHz band, and use Pulse Position Modulation (PPM). These utilise the same message lengths as the interrogation. A Mode S message's first five bits define the contents of the message - the resulting number is called the Uplink Format (UF) for an interrogation and the Downlink Format (DF) for a reply. The possible UF/DF are seen in Figure 3.

The SSR works by acquiring aircraft first. This happens with All-Call messages, which address all aircraft within its coverage. The All-Call messages have the format seen in Figure 4.

When the aircraft is acquired, other message types are used to obtain information besides the aircraft's unique 24 bit ICAO address. The Selective aspect of Mode S originates from the fact that the interrogator can choose to interrogate a specific aircraft using its

UF/DF	Bits	Uplink type	Downlink type
0	56	Short air-air surveillance (ACAS)	Short air-air surveillance (ACAS)
4	56	Surveillance, altitude request	Surveillance, altitude reply
5	56	Surveillance, identity request	Surveillance, identity reply
11	56	Mode S All-Call	All-Call reply
16	112	Long air-air surveillance (ACAS)	Long air-air surveillance (ACAS)
17	112	-	Extended squitter
18	112	-	Extended squitter/non transponder
19	112	-	Military extended squitter
20	112	Comm-A, altitude request	Comm-B, altitude reply
21	112	Comm-A, identity request	Comm-B, identity reply
24	112	Comm-C (ELM)	Comm-D (ELM)

Figure 3: Mode S Message Formats [17]

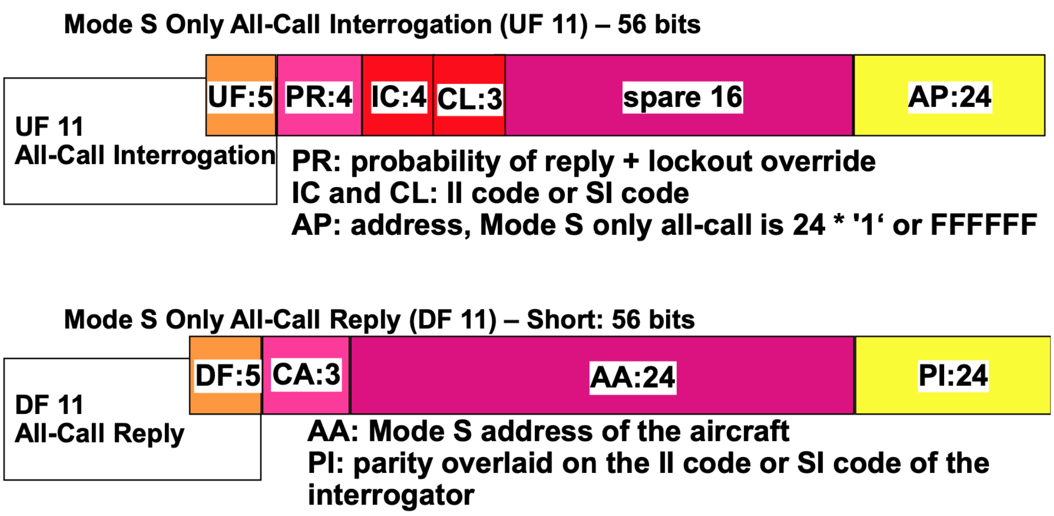


Figure 4: Mode S All-Call Format [3]

ICAO address, a unique 24-bit identifier that is allocated to it. This prevents over-interrogation, as only one airplane will reply to a query, as opposed to all airplanes within the radar's coverage, which would pollute the RF environment.

A raw Mode S message, when demodulated, is received in binary format. After this, it can be converted to hexadecimal and finally into the decimal system. Several of the message types have additional means of making information more compact (for instance the airborne position message for ADS-B) and thus require additional decoding (and encoding) procedures. A detailed guide on decoding Mode S messages can be found in Dr. Junzi Sun's book The 1090 MHz Riddle [17].

3.2.3 ADS-B Basics

Automatic Dependent Surveillance - Broadcast, or ADS-B, is a commonly used standard in modern aviation. As this thesis will dive deep into a similar standard, it is important to outline the theoretical basis for this technology.

Using ADS-B, an aircraft broadcasts information about itself to other airspace users. The "Automatic" part refers to the fact that this process happens without any external input/interrogation. "Dependent" refers to the fact that the system depends on other sensors for data, such as the satellite navigation system antenna in the airplane to obtain data about position and velocity. While "Surveillance" refers to the function that the system is enhancing, the "Broadcast" refers to the fact that the aircraft is transmitting information.

In terms of format, it is crucial to recognize that ADS-B is a Mode S service and thus relies on the same formatting as seen in subsubsection 3.2.2. For a message to be recognized as an ADS-B transmission, the Downlink Format seen in Figure 4 is either 17 or 18, and is refereed to as the Extended Squitter capability. The layout of the ADS-B message can be seen in Figure 5:

```
+----------+----------+-------------+-----------------------+-----------+
|  DF (5)  |  CA (3)  |  ICAO (24)  |        ME (56)        |  PI (24)  |
+----------+----------+-------------+-----------------------+-----------+
```

Figure 5: ADS-B Message Structure [17]

It is seen that the full length message is broken down into several sections. The first section is the Downlink Format (DF), which indicated what type of Mode S message has been sent, has a length of 5 bits and in the case of ADS-B messages takes the value of 17. The transponder capability (TC) is the next field and shows the level of the transponder - it has a length of 3 bits and thus takes values between 0 and 7. Next, the airframe's unique ICAO code is present, and is a 24-bit value. This is followed by a 56-bit message field and a parity index, which is a means of verifying the message's integrity upon decoding it.

The sections of such a message are summarised in Figure 6.

Bit	No. bits	Abbreviation	Information
1–5	5	DF	Downlink Format
6–8	3	CA	Transponder capability
9–32	24	ICAO	ICAO aircraft address
33–88 (33–37)	56 (5)	ME (TC)	Message, extended squitter (Type code)
89–112	24	PI	Parity/Interrogator ID

Figure 6: ADS-B Message Sections [17]

For this message type, several different types of information can be broadcast. As can be seen in Figure 5 and Figure 6, the message content is found in the "ME" field and constitutes 56 bits. From these bits, the first three are the typecode (TC) and determine what data the ME field is to contain. These typecodes range from 0 to 31, and can be summarised in Figure 7.

For instance, if the TC field is decoded to a value of 5, it is known that the surface position will be provided in the remainder of the ME field. Decoding ADS-B messages is

Messages	TC	Ground (still)	Ground (moving)	Airborne
Aircraft identification	1–4	0.1 Hz	0.2 Hz	0.2 Hz
Surface position	5–8	0.2 Hz	2 Hz	-
Airborne position	9–18, 20–22	-	-	2 Hz
Airborne velocity	19	-	-	2 Hz
Aircraft status	28	0.2 Hz (*no TCAS RA and Squawk Code change*)		
		1.25 Hz (*change in TCAS RA or Squawk Code*)		
Target states and status	29	-	-	0.8 Hz
Operational status	31	0.2 Hz	0.4 Hz (*no NIC/NAC/SIL change*)	
			1.25 Hz (*change in NIC/NAC/SIL*)	

Figure 7: ADS-B Message Typecodes [17]

straightforward if the procedures in [17] are followed. The decoding methods differ depending on the message content. The specifics are clearly explained in [17].

In Figure 7, the update rates for each message type are also seen. This is a relevant reference for the drone ADS system, as it defines the number of messages emitted per second.

3.2.4 Data Formatting

An important point to mention is the nature of data formatting present in ADS-B. When messages are received, they are present as a series of pulses.

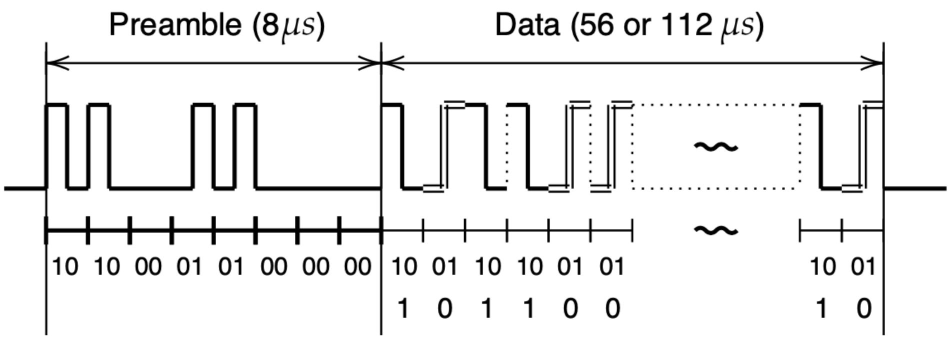


Figure 8: Mode S Message Example [17]

In Figure 8, a received set of pulses is shown. Here, the message is thus encoded in a binary format. Received messages are often represented in hexadecimal format. It is simple to convert between the two, but a manual conversion is tedious and a Python implementation is used. This is done as follows:

- For a binary to hexadecimal conversion of a binary message msg, use $common.bin2hex(msg)$
- For the inverse (hexadecimal message msg to binary), simply use $common.hex2bin(msg)$

This project will also require encoding into binary and hexadecimal. This is done in Python by using the following procedure:

1. Define a message component as a string **str_comp**

2. Use the Python syntax $bin\_comp = f'\{str\_comp : binN\}'$ where N is the number of bits

3. For hexadecimal, use same syntax only $hex\_comp = f'\{str\_comp : hexN\}'$

Finally, to convert a binary message to integer format (which is often found at the last step of the decoding process), the following syntax is used:

$$common.bin2int(msg)$$

3.2.5 Range-Power Model

For a simulation to be built, a range-power model is necessary. This model will model the signal power as a function of the emitter-to-receiver distance. For space loss as a function of distance and wavelength, the following relation is given [15]:

$$\frac{P_r}{P_t} = D_t D_r (\frac{\lambda}{4\pi d})^2 \tag{1}$$

where P_r is the power at the receiver in Watts, P_t is the transmitter power, D_t is the transmitter directivity, D_r is the dimensionless receiver directivity, λ is the wavelength in meters, and finally d is the distance between the transmitter and receiver in meters.

This equation [15] has many unknowns, and thus the model is built as follows:

$$L_r = k - 20log_{10}(d) \tag{2}$$

where L_r is the signal power at the receiver in Decibels (dB), k is the fit constant determined through optimization with experimental data and d is the distance from emitter to receiver. To elaborate, k consists of the following:

$$k = 10log(P_t D_t D_r) + 20log(\frac{\lambda}{4\pi})$$

where the constants have been seen previously. It is important that some of the values are known - for instance, the simulation will clearly define the wavelength and transmitter power. This means that the model can be scaled easily to fit the drone application. The values of the directivities are assumed to be constant between both implementations, due to the fact that the antennae are omnidirectional.

Experiments have been performed at TU Delft in order to measure the signal power at the receiver and thus compute the value of k for Mode-S signals [15]. These used measured power values at the receiver and reported ranges to create a range-power model, which is seen in Figure 9:

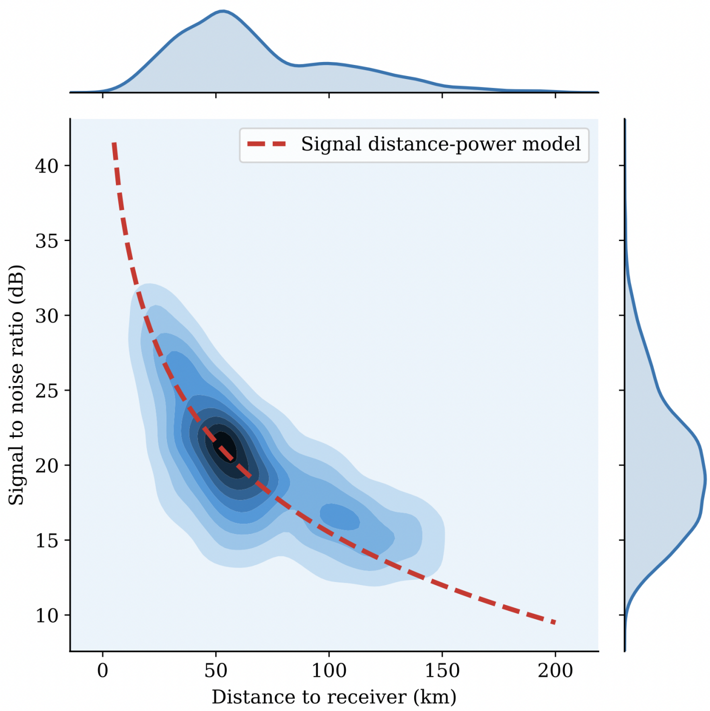


Figure 9: Power-Range Graph for ADS-B [15]

Here, the X axis is the range in kilometers, and the Y axis is the Signal to Noise ratio at the receiver in dB. With adjustments for power and wavelength, a similar plot can be made for the signal power at receiver in the case of drone use.

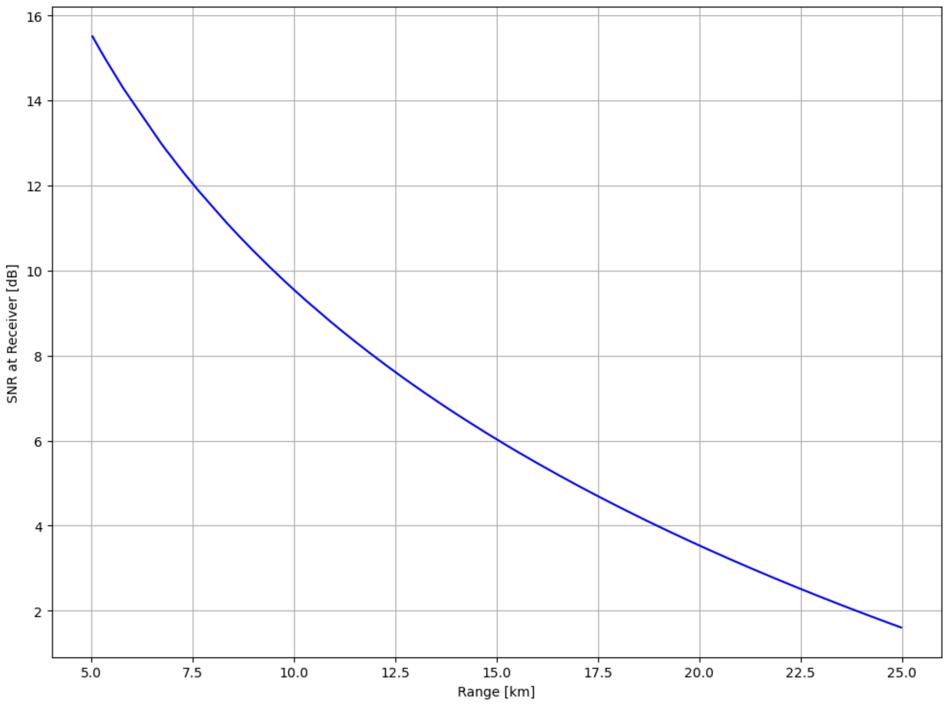


Figure 10: Power-Range Graph for Drone ADS [15]

In Figure 10 that simply adjusting the signal wavelength to fit the 433 MHz frequency and adjusting the transmitter power to the assumed 0.5 W (the best-case value - it is likely that this term will be lower in the final work) drops the signal power significantly. The two Signal-to-Noise Ratio plots can be superimposed to show the contrast (low-power drone ADS in blue, traditional Mode S in red), which is seen in Figure 11. Note the range when compared to Figure 9: there are signals coming in from ranges of 200 kilometers for traditional aircraft ADS-B, whereas the plots for drone ADS show a drop-off to 2 dB at a mere 25 kilometers.

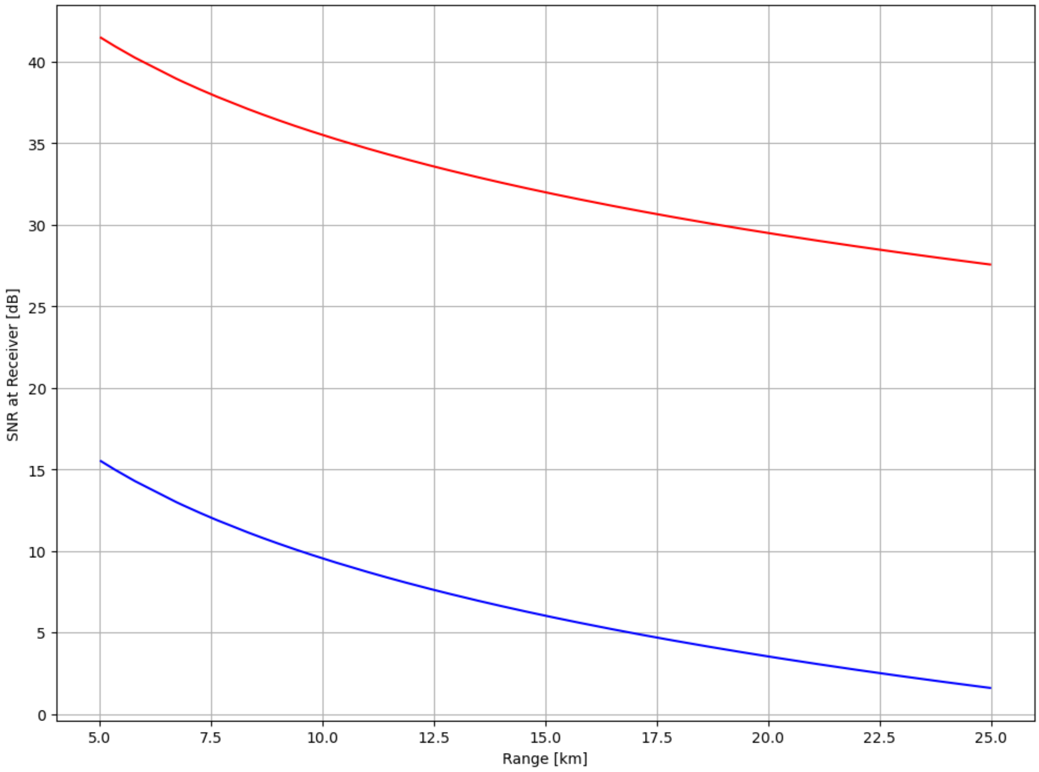


Figure 11: SNR Curve Comparison [15]. Drone curve is blue, Aircraft is red

This foreshadows that for current low-power drone implementations, the range cannot in any case be expected to exceed 20-25 kilometers.

3.3. State of the Art and Other Studies

When beginning research into a new field, it is important to provide an overview of existing research in topics closely related to the one that will be worked on. As mentioned in the previous secTions, the fields where related research has been done are drone-specific ADS-B implementation impact assessment and Other drone-specific Surveillance Systems. This section will cover several studies pertaining to the state of the art in the drone surveillance field .

3.3.1 Challenges in Drone ADS

Before the state of the art research is presented, it is important to state the challenges and limitations of implementing an ADS system for drones. These are:

- The power of a drone transponder should be lower due to the lower power available when compared to an aircraft
- ADS-B uses a 24-bit unique identifier system. The unique IDs are assigned to states in batches - as it stands, the number of unique identifiers will not be sufficient to assign a unique one to all consumer drones

As will be seen, these points have already been addressed by some of the research.

3.3.2 Impact of sUAS Equipped with ADS-B on 1090MHz Environment

The first of these sources is the research done by P. Jonaš et al. [9] on the RF environment impact of Small Unmanned Aerial Systems equipped with ADS-B. In more detail, this research considers high quantities of drones at low altitudes (below 500 feet above ground level), specifically evaluating their impact on the reception of aircraft ADS-B 1090 MHz extended squitter by ground sensors.

The methodology uses a computer simulation and EUROCONTROL's 1030/1090 MHz RF model [9].

Using multiple recorded air traffic scenarios [9] that vary drone density and transmitter power, it is found that the 98.5% 5 seconds probability of update range is decreased by up to 38% through the addition of ADS-B capable traffic.

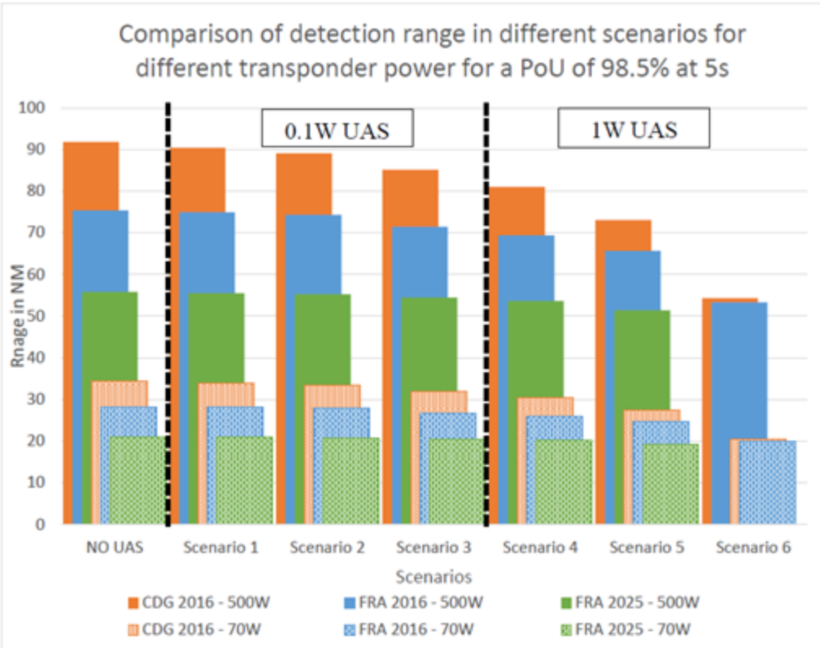


Figure 12: Impact of ADS-B capable drones on 98.5% detection range

In Figure 12, the results of the simulations run are presented more clearly. In the right of the figure, the lowest range for an acceptable probability of update is seen, and corresponds to a case with higher drone transponder power, as expected.

This shows that while feasible, a direct carryover of current ADS-B standards to drones has the potential of strongly impacting aircraft ADS-B performance, which is already stretched thing due to an overuse of the 1030/1090 frequencies. Taking this potential degradation into account, it is recommended that other carrier frequencies be considered to avoid further over-saturation of the 1030/1090 spectra.

3.3.3 ADS-B Surveillance System Performance with Small UAS at Low Altitudes

The research from the MITRE corporation [5] is similar in goal to that seen in [9], but utilises a different approach. This study, however, investigated both air-to-air and air-to-ground scenarios. Another difference is that instead of evaluating the maximum range at a certain probability of 5-second update, this research considers a probability of decoding of a message given a certain traffic situation.

3.3.4 SESAR: The Clear Air Situation for uaS (CLASS)

A current pinnacle of research in the field of drone surveillance is the work done by the SESAR joint undertaking, which focuses on testing different surveillance means in different scenarios while testing potential data fusion systems [10].

This project is a large industry partnership between Airbus Defence and Space, Aveillant, ENAC, The Norwegian University of Science and Technology and Unifly. For this project, a mixture of Airbus' DroneIt! and Aveillant's radar systems was used. The results showed that drones could be tracked reliably for a range of up to 5 km [11].

From this research, several observations can be made. Firstly, there is already a push into the integration of UAS surveillance with the Single European Sky project. This means that a flexible and robust drone ADS system can help greatly with this endeavor. the project utilses Aveillant holographic radar as a non-cooperative means of communication and Airbus' DroneIt! as the cooperative element. This is a custom chip, and unfortunately besides its integrated GPS module, no other data about it is available.

This is thus a point where the drone ADS system can be designed better - it must be open-source and therefore transparent to all potential users.

3.3.5 SESAR U-Space Research

The research performed for CLASS feeds into the concept of the U-Space. To quote SESAR, "U-space is a set of new services relying on a high level of digitalisation and automation of functions and specific procedures designed to support safe, efficient and secure access to airspace for large numbers of drones [13]". The U-Space services are shown as follows [13]:

- U1: U-space foundation services covering e-registration, e-identification and geofencing.
- U2: U-space initial services for drone operations management, including flight planning, flight approval, tracking, and interfacing with conventional air traffic control.
- U3: U-space advanced services supporting more complex operations in dense areas such as assistance for conflict detection and automated detect and avoid functionalities.
- U4: U-space full services, offering very high levels of automation, connectivity and digitalisation for both the drone and the U-space system.

A drone ADS system that is both flexible and simple to implement can help most of these points, and therefore the U-space projects provide another reason as to why such a system should be developed.

3.3.6 Existing Implementation

Currently, ADS-B implementation on drones, especially consumer models, is sparse. Recently, DJI have launched an ADS-B in capabilities on some of their consumer drones through AirSense [1]. This allows the drone pilot to see airliners as the drone receives and decodes their ADS-B transmissions. While this is helpful, it does not mitigate the issue posed by drones, as they remain invisible to aircraft and ATC - transmission is necessary.

As mentioned, an option for drones would be the direct implementation of ADS-B. Conventional aircraft transponders are not suitable, as their weight and size far exceeds what a drone could carry and power. There is one development the market that appears to be usable at first glance - the uAvioniX Ping20Si [2]. This piece of kit is a full transponder, weighing in at 20 grams and being 50x25x17mm in size. However, the key downside here is the fact that direct ADS-B is used and the issue of ICAO code is raised.

The final implementation of ADS on drones is the Airbus DroneIt!, as featured in the SESAR CLASS [11] [10] research seen in subsubsection 3.3.4. The DroneIt!, seen in Figure 13, is a small chipset that is GNSS-enabled, and can communicate with both ground installations and satellites.

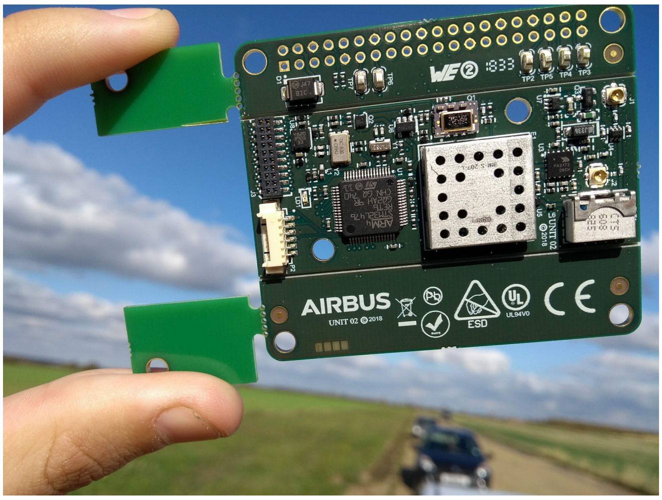


Figure 13: Airbus DroneIt! [10]

Unfortunately, no other information about the DroneIt! is provided, as it is a proprietary system. However, it does demonstrate the fact that an ADS system for drones is necessary, as it is part of the vision put forward by SESAR and Airbus.

[1]https://www.dji.com/nl/flysafe/airsense
[2]https://uavionix.com/products/ping20si/

3.3.7 Comparison of RADAR, passive optical with acoustic, and fused multi-modal active and passive sensing for UAS traffic management compliance and urban air mobility safety

This paper by S. Siewert et al. [14] focuses on the comparison of multiple drone surveillance systems, both active and passive. In the system proposed by the authors, the use of passive acoustic and optical sensors is coupled with GNSS and ADS-B.

The hypothesis in this study is that a passive optical system can be implemented at low cost when an active radar system would be unreliable or expensive to set up. A prototype All-Sky Camera system is therefore introduced - this is an omnidirectional camera array with a 120 Megapixel resolution. This is coupled with a Echodyne radar, that operate within a sensor fusion algorithm [14].

Overall, the report concludes by stating that more research is necessary into these systems. While the passive sensor arrays could potentially work on their own, it is recommended that they be integrated with active surveillance methods for additional redundancy.

3.3.8 Need for a Drone ADS System

As seen in the previous section, there is a solid research base into the use of surveillance systems, both passive and active, in the quest for drone deconfliction near or in high-density airspace.

While studies show that in theory, the RF interference from drones utilising the 1090 MHz band may be manageable, the same reports also state that there are many difficulties present with this, namely a transmitter power and traffic density balance that is needed to keep other 1090 MHz uses within spec. There is also a major difficulty regarding unique ID allocation to drones - how does one differentiate between drones and aircraft by ICAO 24-bit addresses alone?

Passive systems also show promise when drone detection is concerned. Systems such as Robin Radar Systems' [3] IRIS and Elvira radars are already operational at several major airports and are used to identify and track approaching drones. However, these lack the benefits of secondary radar - any data beyond the position is unknown, and a service such as ADS-B broadcasts not only position but ID, intent and other parameters.

Finally, it is important to consider that SESAR has several running projects (especially the U-space related work) which address the need for drone surveillance systems. While a tight set of parameters is proposed for each system type, and a comparable system exists in the form of Airbus' DroneIt! [11], it is unclear as to whether it can support future standards as no information about the specifications of the system is disclosed.

[3] https://www.robinradar.com

4. Planning and Research Design

This chapter will focus on the research planning and design. Firstly, the planning will be outlined. Then, the programming language selection, environment setup, hardware setup for validation, and design constraints from literature complete the chapter.

4.1. Envisioned Planning

Before the state of the art is discussed, a brief section is dedicated to the project planning. While a detailed overview (including Gantt charts) is given in the Project Plan [18], this section aims to frame the theory covered in this literature review report within the project's timeframe.

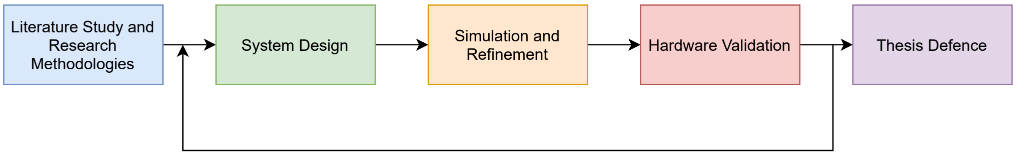


Figure 14: Project Plan Flowchart

In Figure 14, the project's high-level structure is seen. The first phase consists of the Literature Study and Research Methodologies. This culminates in the Project Plan report [18] and this Literature Review.

Next, the System Design phase begins. This phase focuses on designing an ADS standard for drones using a system design approach. The theory used here is taken heavily from the General ATM, EM wave and Digital Communications theory literature.

The Simulation and Refinement stage comes next, and relies on the results of the system design, along with the same literature categories. In addition, this uses Python, for which the setup is listed under subsubsection 4.2.2 .

Following from this, the on-hardware validation begins. This relies on the final branch of literature in addition to those already used: the literature about other possible ADS systems. It also utilises a hardware setup - while not defined at this stage of the project, potential setups and components for this are listed in subsection 4.5.

Finally, the thesis is to be wrapped up and presented. This is seen in the Thesis Defence Block.

Currently, the project is nearing the end of the model refinement stage, and sees a start to the on-hardware verification. This can be visualised in Figure 15, where the red text indicates the current placement in time and the greyed-out boxes show the stages that have been completed.

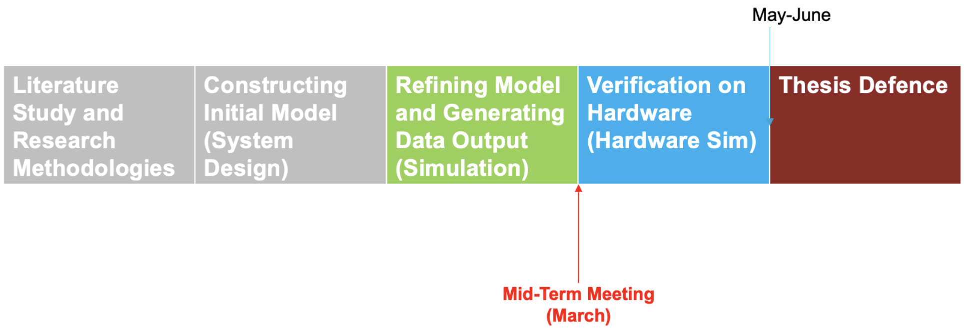


Figure 15: Rough Time Plan for Thesis Work

One last aspect to mention is that this process is not necessary linear - a feedback loop is present, and implies that modifications might need to be made to the standard based on the simulation or validation results.

4.2. Programming And Simulation Setup

Here, the programming setup will be described. This will cover the language choice, installation and packages, and work aspects such as IDE selection.

4.2.1 Programming Language choice

As mentioned, this thesis will rely heavily on Python. This is the principal scripting language of choice, as it simple to use and fast for prototyping work, which is expected to be of great importance for this project. It is possible that another language may be more useful when implementing the ADS standard on hardware. This will likely be a language such as C++, which being a lower-level allows for faster performance compared to Python. This is of paramount importance when message encoding and decoding occurs in real time. Whether a Python-only implementation is possible remains to be seen.

4.2.2 Python Installation and Packages

The Python installation used is obtained from Anaconda, and uses Python 3.8. There are several crucial packages that are employed here - these include:

- numpy
- scipy
- pandas
- pyModeS
- black

The first three are standard packages, and are simply installed by using the command pip install followed by the package name.

The package of principal importance here is pyModeS [16], which allows for the decoding of received Mode S messages. It will be expanded to feature encoding for more data types during this project.

The project will likely utilise other repositories and projects for the hardware testing phase, especially when it comes to signal modulation and transmission, potentially using a micro-controller. If a microcontroller is to be used, a project called MicroPython might prove useful, and can be found at `https://micropython.org/`. This package allows for interfacing with micro-controllers without using low-level languages such as C++, thus saving time by enabling the use of Python for fast and easy prototyping.

4.2.3 Python Setup and IDE

A final note to make in this section is the work setup used. The pyModeS GitHub repository is used to install the development version of the project, and to commit changes. The Integrated Development Environments (IDEs) used are Spyder (standard with Anaconda) and Visual Studio Code, which helps with consistent code formatting, variable name changes and auto-completion.

4.3. Hardware Setup Possibilities

This section will discuss potential hardware setups for testing the developed ADS standard. Firstly, some considerations that are of importance for the selection procedure are listed and justified. This is followed by a set of potential hardware setup types, with their own possibilities in terms of hardware.

4.4. Main Considerations for Hardware Setup

For a hardware setup to be useful, it has to fulfil the objective of: Test and validate the developed drone standard. More specifically, the goal of this hardware is to simulate both a transmitter and a receiver, and successfully demonstrate the capabilities of the standard that has been developed. However, it must also be cost-effective and time-effective to set up. Thus, this hardware has to be:

1. Effective for encoding/decoding of the signals
2. Cost-effective/affordable
3. Easy to use
4. Provide transparency and reproducibility

With these requirements in mind, it is time to consider the possible setups that will be used.

4.5. Possible Hardware Setups

There are several types of setup that can be configured for this testing. It is important to understand what the role of each component in this setup will be, and thus this will be an important aspect in each of these.

4.5.1 Computer-to-computer with simple SDRs and antennae

The simplest setup to trial would be a computer-to-computer system, both using Software Defined Radio. One computer (the 'drone') would simulate a drone's position and other parameters such as speed, code, modulate and transmit these via a connected antenna. The second computer would receiver the signal via an antenna, demodulate and decode, thus getting the data transmitted. This would be a basic demonstration, and perhaps the most simple to set up.

This has several advantages:

- Such as system is easy to set up, as both ends would use conventional Python
- There is no processing power shortage envisioned when using two full-scale computers
- The simulation algorithm can be coded on the same machine as it is simulated on - thus, the same Python installation is used, again facilitating the project

Nevertheless, it suffers two main drawbacks:

- It does not necessarily demonstrate that the hardware can be implemented on a small-scale, low power vehicle such as a drone
- Using two SDRs is not cost-effective, as one such piece of hardware costs in excess of 150 Euro (see HackRF)

Thus, this is kept as a last resort, if all other options are to fail.

4.5.2 Micro-controller with simple antenna to computer with antenna

Having considered a computer as a signal generator, it is now time to envision a setup more representative of a low-power, low-weight and compact system such as a commercial drone for the transmitter. This can be achieved through the use of a micro-controller on the transmitting end, that is equipped with a simple antenna that also handles modulation.

The advantages this presents are as follows:

- Best option for low-power applications
- Inexpensive hardware
- Closest to embedded systems found on drones and thus is the best option for meaningful validation

However, the downsides are:

- A micro-controller is difficult to set up. While MicroPython can help mitigate some of the difficulty by allowing the use of Python as opposed to low-level languages such as C++, the difficulty is increased regardless as it is impossible to simply run Python code on such a board without significant work
- More difficult to debug - most of the work will go through the terminal as opposed to a GUI, which is difficult for a novice user

4.5.3 Raspberry Pi with simple antenna to computer

Finally, a Raspberry Pi with simple antenna is considered as an output device. This is, in essence, a lower-power computer, and thus while being further from the embedded system approach achieved with a micro-controller is still more representative of a drone than a full-scale computer, as it uses significantly less power. Thus, its advantages are:

- Easy to implement compared to micro-controllers (essentially a low-power computer) as they directly support Python with all of its packages
- Easy to interface with and easy to connect peripherals
- Reasonably low cost compared to full-scale computers

The downsides are that:

- The validity of the results is a notch below the micro-controller implementation, as this is using a higher-level code and is thus further from the drone hardware
- Difficulty is mildly increased from the computer-to-computer setup, due to processing power limitations of the Pi. Realistically, the power of such a system should suffice for the validation scenario envisioned.

4.6. Potential Component Selection

This subsection summarizes the components that can be chosen for each setup type. These components are all readily available, but differ in price and performance.

4.6.1 Computer-to-computer case

Firstly, the Computer-to-Computer scenario is considered. For this, two generic computers can be used, and thus do not need to be purchased. The parts of importance are the antennae and software-defined radio.

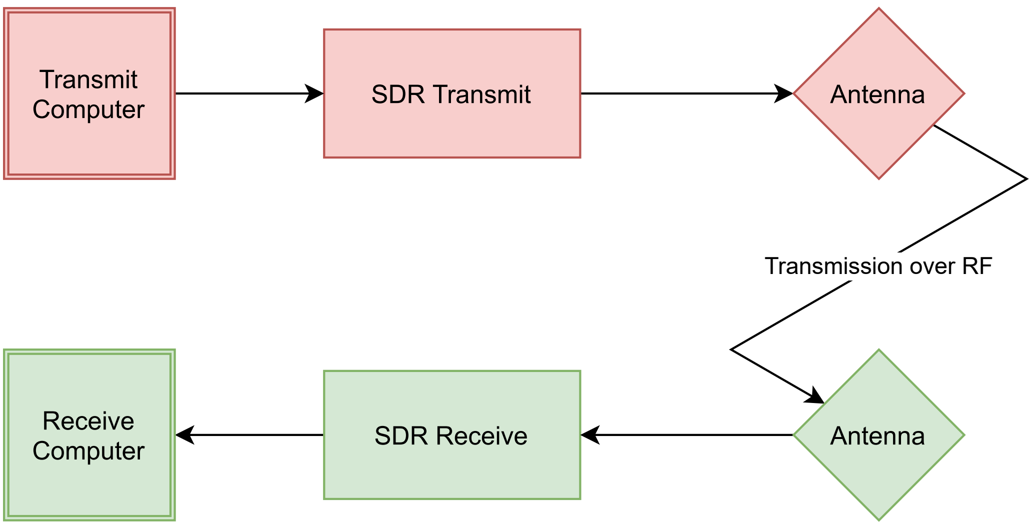


Figure 16: PC-to-PC setup

Figure 16 shows the expected setup flowchart for this configuration. Note that while the 'SDR Receive' (Receiver SDR) is mandatory, it would be possible to transmit without an SDR at the cost of performance loss and thus that component can be removed to save cost.

For the antennae, generic dongle antennae can be chosen.

Then, the SDR and cabling are selected. Some options include:

- HackRF (and clones), with prices ranging between 150 to 350 Euro
- SDR Play RSP
- Airspy at 219 Euro or Airspy Mini at 119 Euro
- LimeSDR at 350 Euro and up - while a positively received option, it is currently difficult to purchase in Europe
- RTL-SDR, a budget solution below 50 Euro (https://www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/)
- Ultrabudget option - NESDR `https://amzn.to/3cW2tas`

For the dongle-style receivers, no cable is needed as they use a simple USB-A port connection to the computer. For more complex SDRs, coaxial cables are needed for the connections.

The course of action will be to utilise a low-cost SDR at first, such as the proven RTL-SDR, and if performance is insufficient, a better performing SDR is to be considered.

It must also be noted that an SDR is necessary on the receiver end and not on the transmitter end, as inexpensive antennae can handle modulation.

Another option here is to purchase a complete receiver kit for ADS-B, such as the one seen in `https://www.passion-radio.com/adsb/start-adsb-834.html`.

4.6.2 Microcontroller to PC Case

Here, the microcontroller acts as the drone's on-board ADS processor and allows for the transmission of the ADS signal. The setup flowchart is seen in Figure 17.

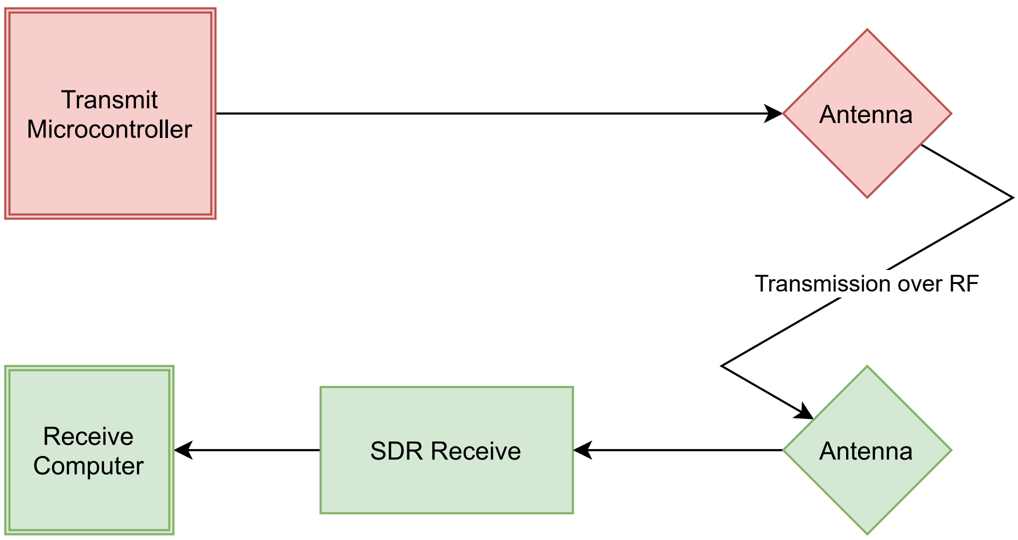


Figure 17: Microcontroller to PC setup

This setup's receiving end is the same as the PC-to-PC case, as expected. On the transmission end, the goal is to have a simple antenna connected to the microcontroller as the only hardware used.

There is a large number of potential choices here, and some have been summarized below:

- Raspberry Pi Pico (sub 5 Euro) [4]
- MicroPython pyboard v1.1 with headers (approx. 40 Euro) [5]
- Pyboard D-series with STM32F722 and WiFi/BT (approx. 50 Euro) [6]
- Arduino models (10 to 50 Euro) [7]

Here, the best starting point is the Raspberry Pi Pico, by virtue of its low price point compared to competitors. While the Pyboards do outperform the Pi in some regards, such as a greater number of IO pins, slightly higher processor spec and more RAM, it must be noted that the available processing power on all of these boards is strongly limited.

The range of Arduino microcontrollers has been cited but not explored in detail. This is because of the use of C++. which adds complexity to the validation process and will be more difficult to set up. Thus, boards that utilise languages other to Python will be researched if the Python-enabled solutions do not work.

4.6.3 Raspberry Pi to PC setup

Finally, the Raspberry Pi to PC setup is considered. Here, the flowchart is neglected, as it would look exactly like Figure 17, except the first microcontroller block is replaced with the Raspberry Pi. Many Raspberry Pi compute boards are available [8]:

- Raspberry Pi Zero 2 W (approx. 20 Euro)
- Raspberry Pi Zero
- Raspberry Pi Model 3
- Raspberry Pi Model 4

There are other options available here, but it appears that the smaller devices, such as the Pi Zero 2 (W) and Pi Zero could be the best choices due to their small footprint and lower power consumption.

4.7. Design Constraints from Literature

There are also some considerations to be taken regarding the potential constraints that the design will have. This section will be updated as the design phase is taken on.

[4] https://www.raspberrypi.com/products/raspberry-pi-pico/
[5] https://store.micropython.org/product/PYBv1.1H
[6] https://store.micropython.org/product/PYBD-SF2-W4F2
[7] https://www.arduino.cc/en/main/products
[8] https://www.raspberrypi.com/products/

4.7.1 Frequency Usage Constraints

It is expected that the implementation and testing are to take place in the Netherlands. This means that the system must be made in accordance with state law and recommendations on radio frequency usage.

The Dutch government has a dedicated website (https://wetten.overheid.nl/BWBR0036378/2016-12-28) listing the frequencies that are available for unregulated use.

This also means that testing cannot take place on the 1030/1090 MHz frequencies, as transmitters on these bands are not permitted without the required permission being secured.

From the Overheid website, some promising frequencies available for use are compiled below:

- The 433 and 446 MHz bands
- The circa 860 MHz bands

Beyond this, only options that are unlicensed will be considered - this simplifies the selection.

4.7.2 Duty Cycle and Power Constraints

As specified in (https://wetten.overheid.nl/BWBR0036378/2016-12-28), the frequencies to be used are unlicensed, but are subject to the constraints on the duty cycle of the transmitter.

The conditions for operation regarding duty cycles presented for each frequency are as follows:

- 433 MHz: <10% at max. 10 mW e.r.p.
- 869,400 – 869,650 MHz: <10% at max. 500 mW e.r.p.

Therefore, it is likely that the 433 MHz band will be dropped, as the maximum allowed ERP is a mere 10 mW.

4.7.3 Size Constraints

It is important to note that a consumer drone's internal systems (and especially the antenna) will be constrained by its size. For an ADS system to be implemented aboard a drone, it is expected that the onboard processor can handle the encoding task without issue. The constraint comes from antenna size, namely given the frequencies used.

A short compilation of 23 popular drone dimensions yielded several noteworthy results. For the 433 MHz and 869.4 MHz frequencies, the required quarter-wavelength antenna sizes can be computed to be:

$$\lambda_{433}/4 = \frac{c}{4 \cdot f} = 0.1731\text{m} = 17.31\text{cm}$$

$$\lambda_{869}/4 = \frac{c}{4 \cdot f} = 0.0862\text{m} = 8.62\text{cm}$$

Applying safety factors of 1/0.8 to account for unusable space inside the drone, the longest dimension for 23 popular consumer drones was considered [9].

For the 869 MHz, all popular drones could comfortably fit the antenna. On 433 MHz however, the antenna length of 17.31 cm becomes a prohibitive size for some smaller models such as the Mavic Mini and Mavic Air series, which have expected avaliable internal space of 15 centimeters or lower.

These may therefore have to utilise lower performance antennae - this is not investigated in detail at this stage but must be taken into account.

5. Work Done and Preliminary Results

This section will focus on the work done prior to the midterm, as well as the preliminary results and findings of the thesis work.

5.1. System Design Considerations

As mentioned in the Project Plan report [18], the goal of the project is a preliminary design and proof of concept of an ADS system developed uniquely for drones. It is for this reason that the System Design aspect came first and was of the utmost importance. Firstly, the requirements will be listed briefly.

5.1.1 Key Requirements

An ADS system tailored to drones must be:

- Simple to implement
- Concise
- Efficient
- Comprehensive
- Robust

The main points of focus were:

1. The data format
2. The carrier frequency
3. The update frequency
4. The recommended transmitter parameters

These will now be discussed separately in their respective sections.

[9] https://tud365-my.sharepoint.com/:x:/g/personal/svlaskin_tudelft_nl/EQR286UZ1blKs5b6QZXySJcB8Tz8HjTWbkukeoBynSRdAA?e=AbL7Zk

5.1.2 Data Format for Drone ADS

The Data Format is one of the most important aspects of the drone ADS system, as it defines the structure of the data that is transmitted. Several aspects were considered for this, stemming from the system requirements.

Firstly, the format must be simple. This means that the adjustment from the ADS format used for aircraft must be minimal.

Secondly, the format must be comprehensive - the most important aspect of this system will be to identify drones in controlled airspace and know their position (and other information such as their track).

From these considerations, some conclusions are made:

- The current ADS format includes fields that are not useful for drone applications
- The current ADS format is missing fields that would be beneficial to drone use
- For the sake of simplicity, the current ADS format's main characteristics can be carried over for drone use

While it may seem contradictory to say that the current format is sub-optimal while stating that its main characteristics should be kept at this stage, it is important to consider the context. It is certain that organisations such as ICAO will take this task on when necessary, so for the sake of this project, the current aircraft Mode-S data format will be kept but modified slightly to better fit drone use. These modifications are:

- The altitude encoding is scaled to lower altitudes
- The position encoding will use a reference receiver position for every area - that is such that even and odd messages [17] need not be sent
- The unique identifiers will be allocated randomly upon drone startup (as the overlap probabilities are low)
- Some message types are to be removed, and others added

5.1.3 Altitude Encoding Recommendations

The altitude encoding being changed to lower altitudes is logical - consumer drones are very limited in their operational ceiling, with theoretical values of below 3000-6000 meters. For military drones, some can reportedly go as high as 10,000 meters. For traditional ADS, the height is encoded as far as 50175 feet (15,293.34 meters) and beyond that as high as 254,000 feet for barometric altitude alone. This is clearly unnecessary for drone use, and drones operating at those altitudes would best be registered as aircraft with ICAO. Thus, the ceiling for the encoding can be changed to 15,000 meters, with steps of 20-100 meters (as this is the range of GPS altitude accuracy).

5.1.4 Position Encoding Recommendations

The position encoding in a traditional ADS system works through the use of Compact Position Reporting (CPR) [17]. This compact position reporting requires two messages to uniquely determine the position of a transmitter, that is an Even and an Odd message. Alternatively, if both the transmitter and receiver use a reference position within the latitude and longitude slice where the transmitter is found, a unique message would again be sufficient to decode the position. With the current Mode-S transponders in place, signals are picked up by receivers at ranges of up to 200 kilometers [15] - this means that the airplane can be found in a wide range of lat/lon "slices" and therefore it must transmit both even and odd position messages periodically such that its position may be decoded unambiguously.

For drones, the situation is different. The transmitter power is expected to be below 1 Watt compared to 250 Watts for an aircraft transponder. For this reason, the range at which the messages will be received and decoded correctly is significantly lower, as seen in Figure 9 of subsubsection 3.2.5. Here, it is clearly visible that the power-range curve for the drone ADS system will be shifted downwards, and therefore the decodabe SNRs will only be achievable at ranges below 15 kilometers. This will be elaborated upon in the results towards the end of the report. For this reason, a solution to the CPR 2-message issue is proposed. The majority of consumer drones are equipped with GNSS sensors, as well as applications with online connectivity. A database of reference positions can be generated, and a drone that finds itself in a given latitude/longitude region will use the database to "tune in" to that location (thus transmitting messages with that reference position taken into account). On the receiver end, the same reference position is known, as it is unlikely that a drone is to be transmitting from outside the same slice given the short message decoding range.

5.1.5 Additional Message Types

Some additional data formats are currently being considered. Since a large portion of drones to be used in the future are expected to be unmanned [12] and functions such as automatic separation will be key [13], the drone ADS format must work to enable these. Several early ideas are:

- A message type containing the drone's next waypoints
- A message type providing the drone's current use case

5.1.6 Final Comments on Drone ADS Format

Currently, the format recommendations are incomplete. It is expected that this section will be completed by the final report, as options for additional formats are being investigated and it is expected that the concrete outcome of the sensitivity study will dictate a significant part of the design choices.

5.2. Capacity Study and Simulation Design

An important pillar of this thesis is the capacity study regarding the use of this system. This subsection will outline the work done on this.

5.2.1 Motivation for Capacity Study

In order for the system to be usable in the real world, it must be made such that it supports an adequate number of drones. While FRUIT [17] is difficult to measure for this system, the capacity will be defined by message overlap in time (garbling). Naturally, the more drones are in a given airspace section, the more messages are expected to overlap (arrive at the receiver at the same time), therefore losing both messages if their power is similar. The real objective will be to test the limits of such a system, if it is to be used - the main metric will be the real update rates taking message overlap into consideration. The second metric will be the spatial consequence of a message overlap - there will be a study to see if some messages will be decodable in case of a message collision in time, and at what range.

Such a study will allow for an estimate of how many drones can be deployed in a single area, also proving that the system deployment will not cause excessive channel occupancy on the frequencies chosen.

This number will vary by scenario, but the limits of the system can be tested if a simulation is to be built. This is exactly what has been done for the project.

5.2.2 Simulation Parameters

The Python simulation is constrained in time and space. The environment consists of a 10-by-10 kilometer grid, representative of a typical urban area comprised of several cities. The grid used is divided into nodes, each 100 meters from each other. When the simulation is run, the signal to noise ratio of a message being emitted is computed for each of these nodes using the underlying principles shown in subsection 5.4. The time scale of the simulation is in given to be a fifth of the message length (currently a 1.2 millisecond length is used

For the drones themselves, they are defined by several key variables:

- The position in meters (x,y)
- The signal transmission starting time in seconds (t)
- The update rate and jitter

The position is self explanatory - it is simply given as an X-Y coordinate (a tuple of floats), ranging from 0 to 10,000 meters in both cases. These values are randomly generated for each drone in the simulation.

The transmission starting time is important, as this is a deciding factor in whether signals from two different drones are to overlap. This value ranges from 0 to 0.3 seconds.

From that starting time, the drone is to transmit signals periodically. The frequency for this transmission is another design parameter, and different values are investigated.

However, a real system cannot keep an exact transmission rate, thus jitter is added. This jitter is done in time, and ranges from 0 to 6 milliseconds (which is 5 times the message length).

The message length in time will also be varied in future tests. Since the same bit length is kept, the length depends directly on the drone's data rate, defined by its transmission characteristics. Considering a typical low-cost off-the-shelf component, these provide significantly lower data rates than what is seen for a Mode S transponder. For example, take the Grove Serial RF Pro [10], which does 115 kbps in the best-case scenario. This means that for 120 bits of data (including the preamble), the message becomes at least 120/115000 = 0,00104348 seconds = 1.044 milliseconds. As a safety, the message length is rounded up to 1.2 milliseconds.

5.2.3 Simulation Working Principles

As mentioned, the simulation's goal is twofold:

1. Evaluate the system capacity
2. Evaluate the range of the system (both clean and in case of signal overlap)

The working principles of the simulation are therefore defined by these two goals. To begin, an outline of the simulation is given:

1. Define spatial grid (100 by 100 km, 100m resolution)
2. Set/Randomize drone parameters and import scenario (number of drones, drone positions)
3. Compute distance matrix to every point in the grid for every drone
4. Compute signal power at every point in the grid for every drone
5. Generate signal transmission time bins
6. Add jitter of up to 5 times the pulse length to time bins
7. Compute overlap, output and analyse data

5.3. Comments on Preliminary Results

Now that the working principles of the simulation code and the reformulated research goals have been presented, the preliminary results will be outlined in this chapter. This will be done in three main sections: the impact of message overlap on decoding range, the message overlap and update rate considerations, finally concluding in general remarks about the findings.

After the current findings have been summarised, the next section will present the next steps of the research, first specifically focusing on their relation to current findings and then the next steps, such as validation.

5.4. Range Consequence of Message Overlap in Time

When designing an ADS-B like system, it is important to consider message overlap and update rates. In order to have a functional system, the message overlap must be kept to a minimum. Failure to do so leads to a potential loss of individual messages - if many messages overlap and are lost, the receiver is unable to update the information about a given drone with sufficient frequency.

In a stochastic system resembling ADS-B, overlap occurs frequently. For this reason, it is important to evaluate its consequences.

[10] https://wiki.seeedstudio.com/Grove-Serial_RF_Pro/

5.4.1 General Principles

As mentioned in subsubsection 5.2.3, the power matrices are generated for every drone depending on their position and transmitter power. These are 100-by-100 grids, and in order to evaluate the signal to noise ratio with all drones taken into account, they can simply be added as the powers are in decibels.

When this is done, it is possible to segment the matrix depending on the resulting signal to noise ratios at each node. From this, the consequence on decodable message range can be evaluated.

5.4.2 Range Impact - Two Similar Power Drones, Spaced ±10km

Firstly, the case of two drones with similar transmitter powers is considered. They are spaced on other sides of the grid at (247.45, 7707.27) and (9923.86, 9687.29), and have transmitter powers of 0.303 and 0.351 Watts.

Their signal powers can be visualized individually in Figure 18 and Figure 19 as heatmaps.

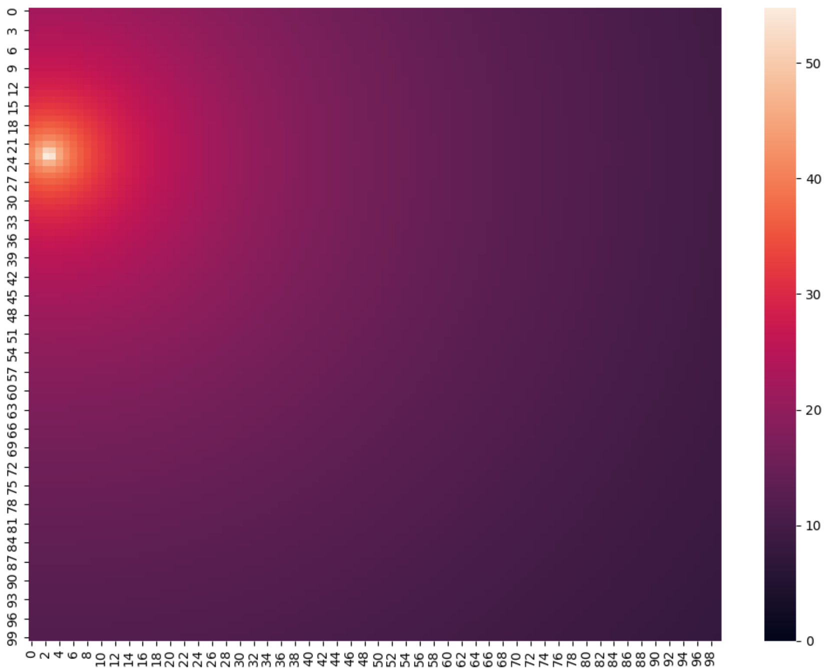


Figure 18: Drone 1 SNR heatmap

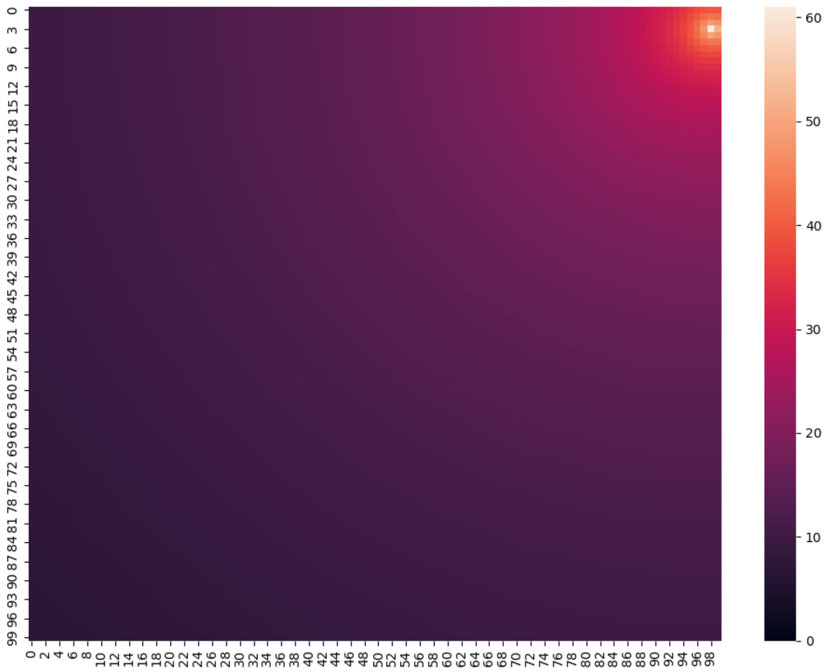


Figure 19: Drone 2 SNR heatmap

Now, consider several decoding thresholds for signal-to-noise ratio, such as 10, 15 and finally 20 decibels. It is possible to filter the data and visualize the zones where the the message can be decoded correctly for a given threshold.

For Drone 1, this is seen in Figure 20 for 10 dB, returning a range of 9.4 km, 15 dB in Figure 21 returning a range of 5.3 km and finally for 20 dB in Figure 22 km returning a range of 2.8 km. The zones where the message can be decoded are seen in beige, whereas the 'dead zones' are black.

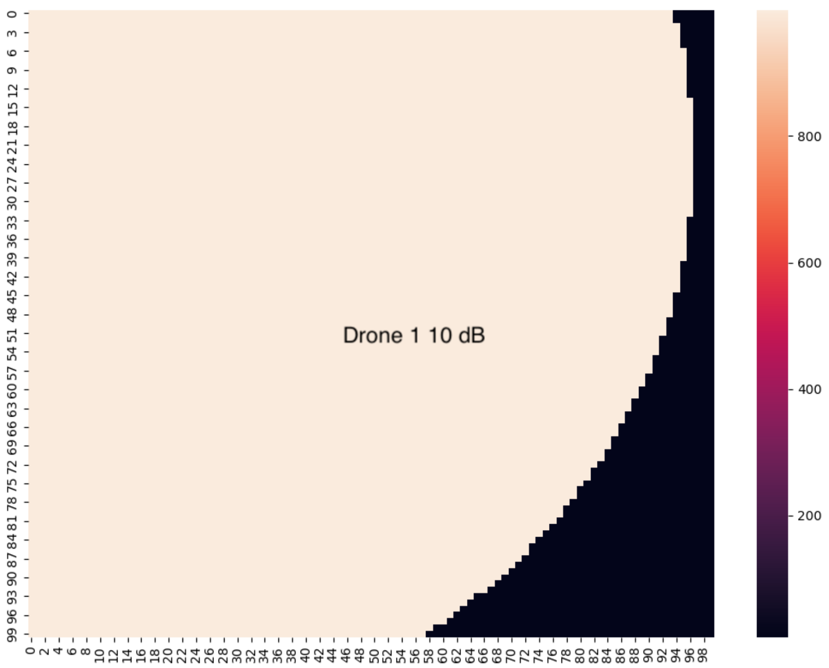


Figure 20: Drone 1 10dB Range

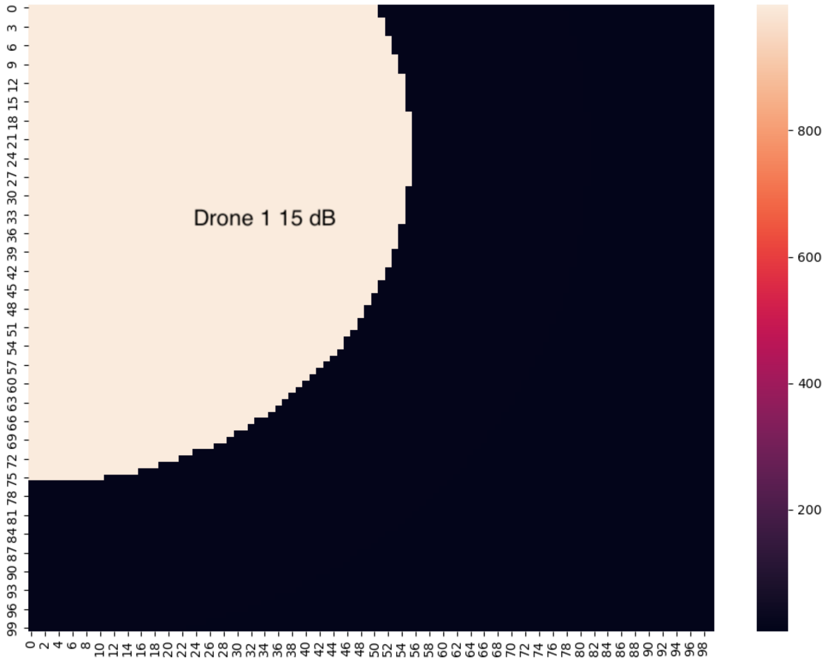


Figure 21: Drone 1 15dB Range

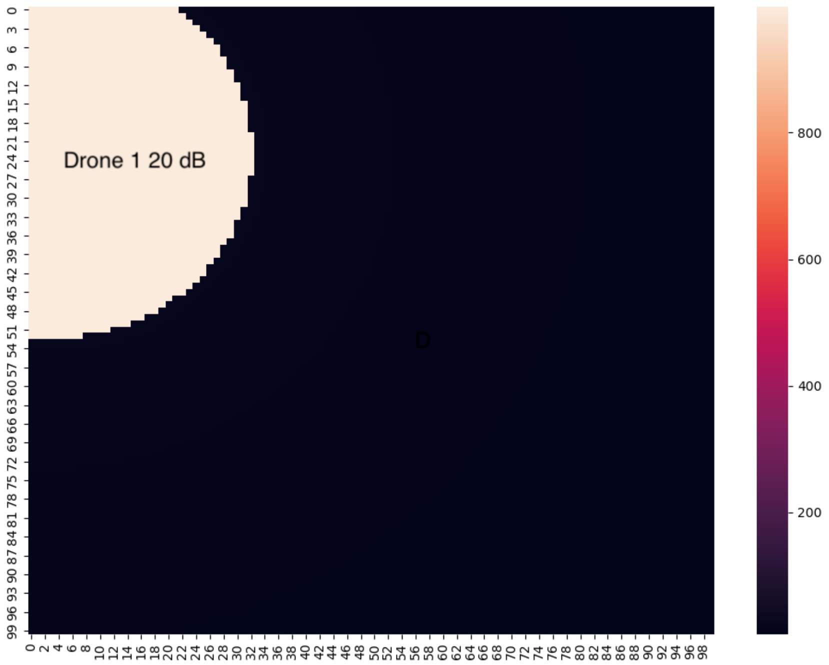


Figure 22: Drone 1 20dB Range

Likewise for Drone 2, this is seen in Figure 23 for 10 dB, 15 dB in Figure 24 and finally for 20 dB in Figure 25 The ranges for Drone 2 are measured to be the same, due to the 100m resolution.

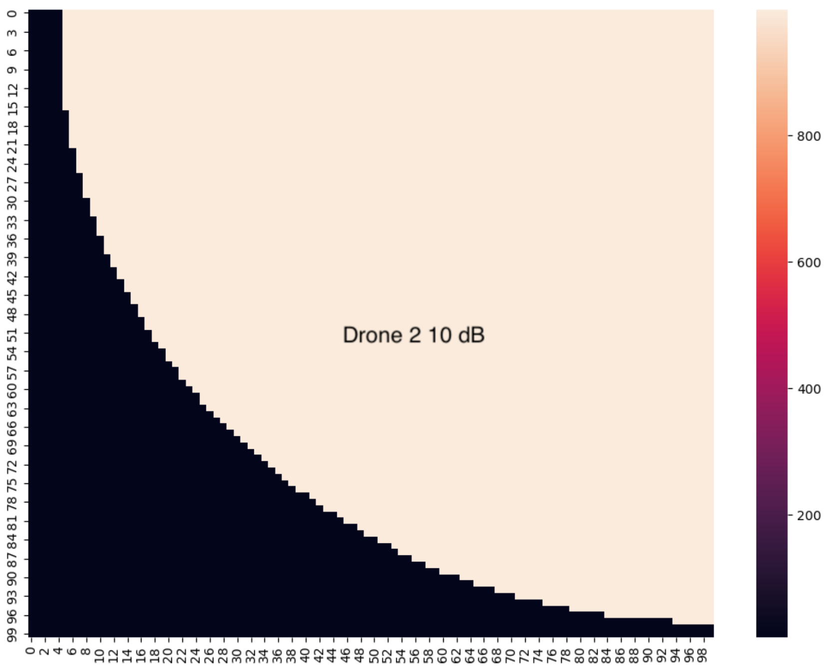


Figure 23: Drone 2 10dB Range

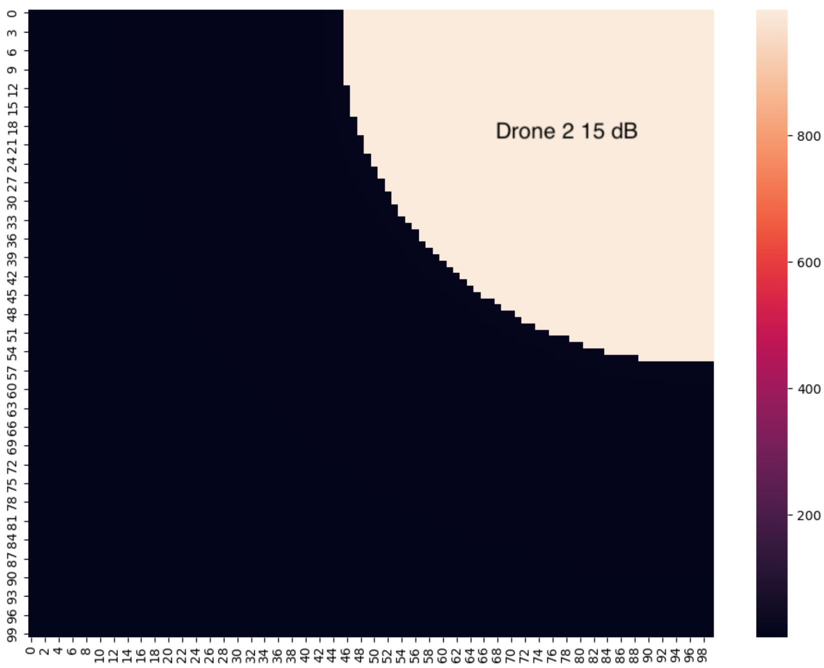


Figure 24: Drone 2 15dB Range

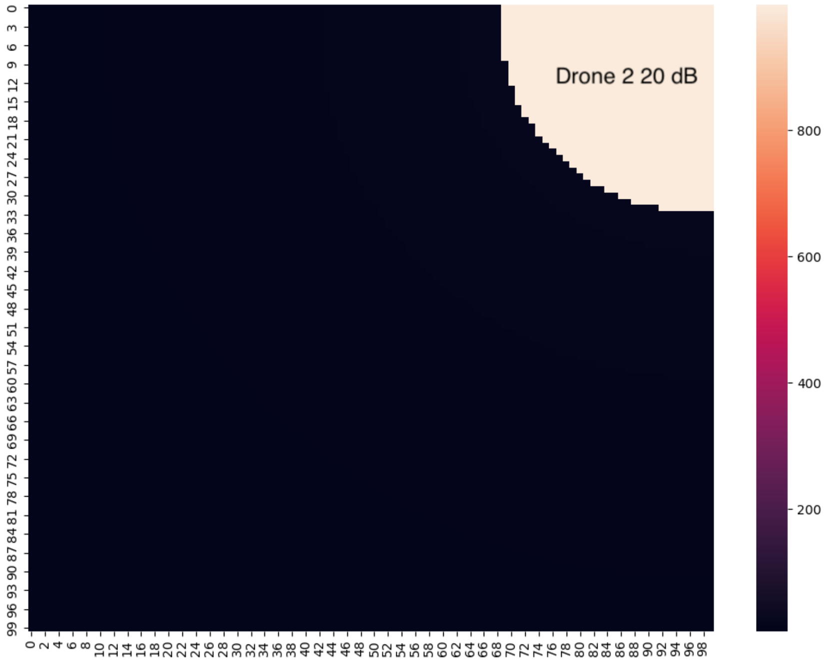


Figure 25: Drone 2 20dB Range

Now that this is done, it is possible to superimpose the signal power matrices. This yields the heatmap seen in Figure 26; Note that the power for drone 1 is given as positive and Drone 2 is negative.

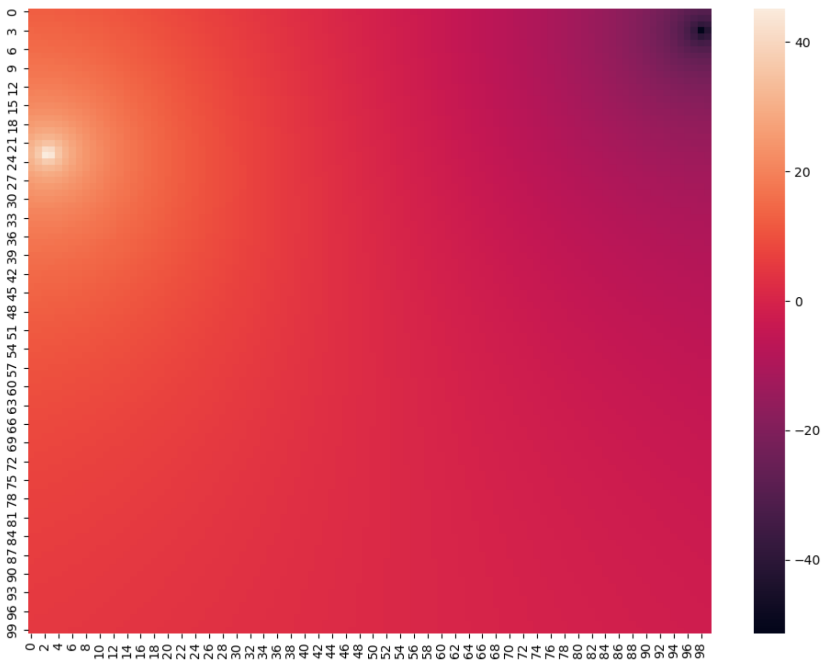


Figure 26: SNR VS Range for Signal Overlap

While this is beneficial at a glance, it does not give accurate insight to how the signal-to-noise ratios are affected as a function of range. The same filtering for 10, 15 and 20 dB thresholds is done, to yield the following results:

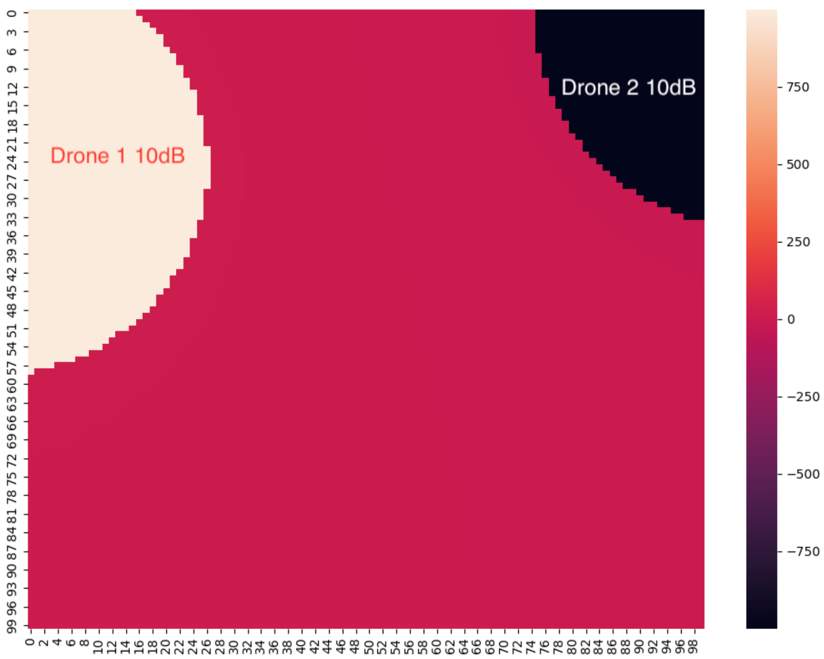


Figure 27: 10dB SNR Range for Overlap

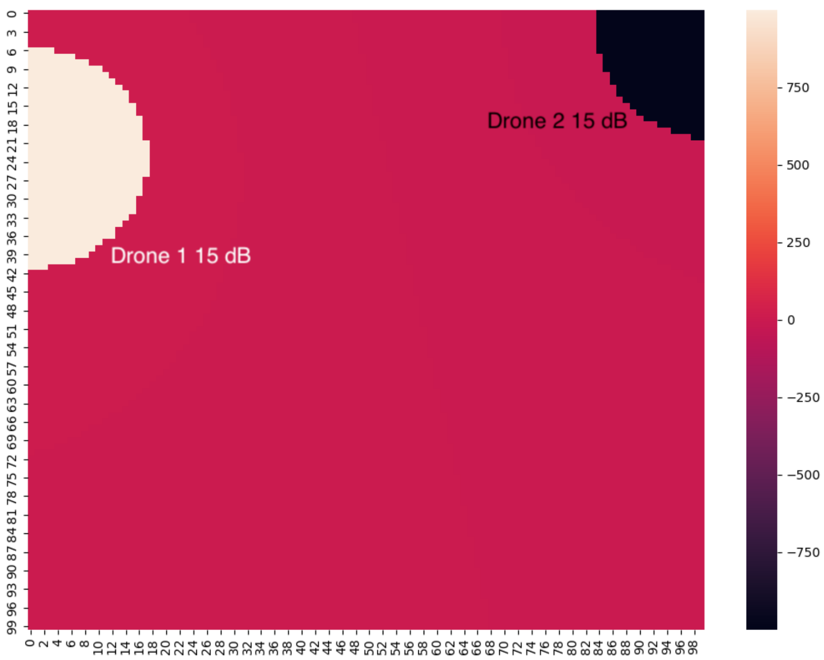


Figure 28: 15dB SNR Range for Overlap

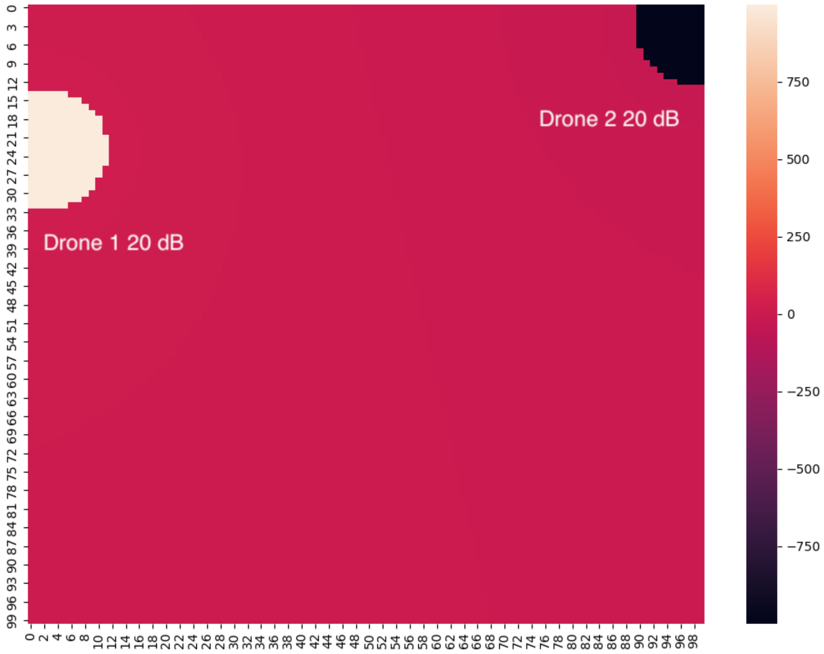


Figure 29: 20dB SNR Range for Overlap

It is seen that the ranges are reduced in every case:

- In Figure 27, the 10 dB range is reduced from 9.4 kilometers to below 2.5 kilometers for both drones.
- In Figure 28, the range drops from 5 to below 2 kilometers
- Finally, for 20 dB, the range goes from approximately 2.8 to well below 1 kilometer. This is truly unusable for the system, as will be mentioned in the overview

Clearly, the system is somewhat resilient to overlaps, especially at 10 dB. While the range drops significantly, from around 9.5 to 5 kilometers, this range is still serviceable for most scenarios. Detection at 5+ kilometers competes with certain passive radar systems for drones such as the Robin IRIS counter-drone radar [11], which cites an instrumented range of 5 kilometers.

For 15 dB, the drop-off is less severe when compared to the range without overlap, dropping from 2.8 to just below 2 kilometers. For 20 dB, the degradation is severe.

Here, several conclusions can be made. firstly, a 10 dB signal-to-noise ratio is achievable over the whole 10 kilometer grid. Therefore, a receiver setup should be optimised to decode messages at this SNR. For 15 dB, the decoding range is a manageable 5.3 kilometers. Finally, 20 dB is not achievable if a longer range is desired, unless consumer drones utilise more powerful antennae.

5.5. Message Overlap in Time

Following the study on the decoding range decrease due to message overlap in time, it is only logical that the extent of the message overlaps in time be considered.

This is done for multiple scenarios, from sparse to dense airspaces. The results can be summarised in this section.

[11] https://www.robinradar.com/iris-counter-drone-radar?hsLang=en

5.5.1 Sparse Airspace: 0.1 drones per km^2

A 600-run simulation is done for a case with 10 drones in the 10-10 kilometer area. The average by run overlap occurrence is shown in the histogram in Figure 30.

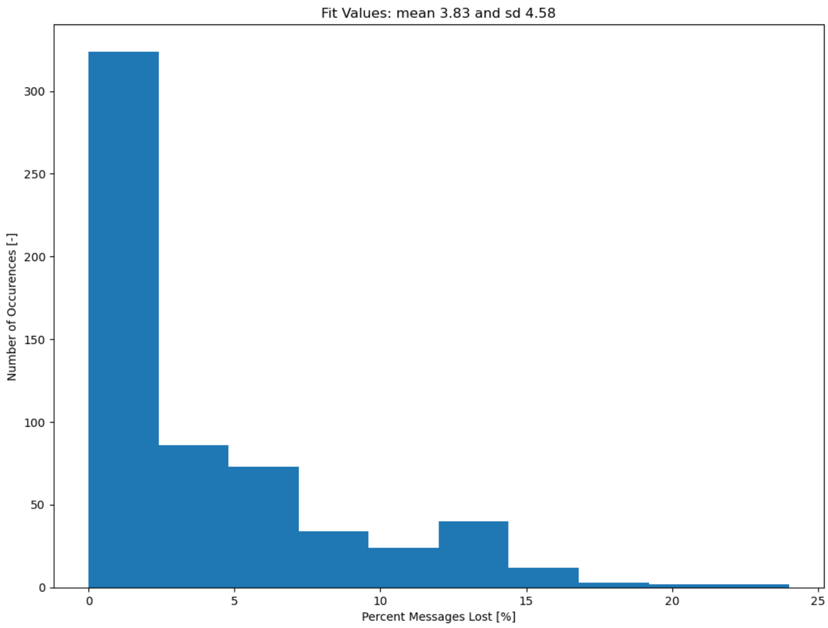


Figure 30: 600 run Overlap Histogram for 10 drones

There, the average messages overlapped is seen to be 3.83 % with a standard deviation of 4.58.

5.5.2 Rural Airspace: 0.25 drones per km^2

Now consider a rural airspace, with 25 drones in the simulation area. Here the result is worse, and is seen in Figure 31.

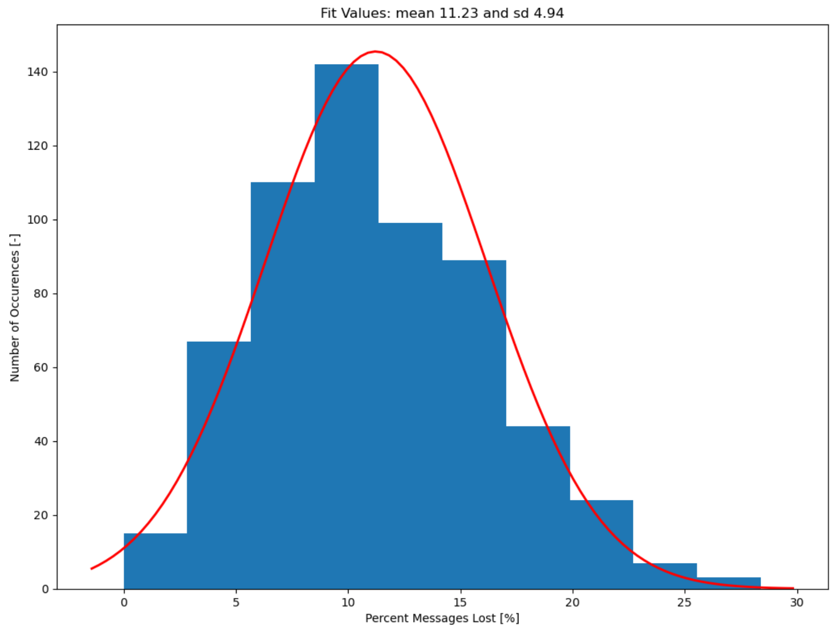


Figure 31: 600 run Overlap Histogram for 25 drones

The mean is at 11.23 % of messages overlapped, with a standard deviation of 4.94.

5.5.3 Suburban Airspace: 0.5 drones per km^2

A suburban airspace could be considered to have 50 drones in the simulation area. Here the result is seen in Figure 32

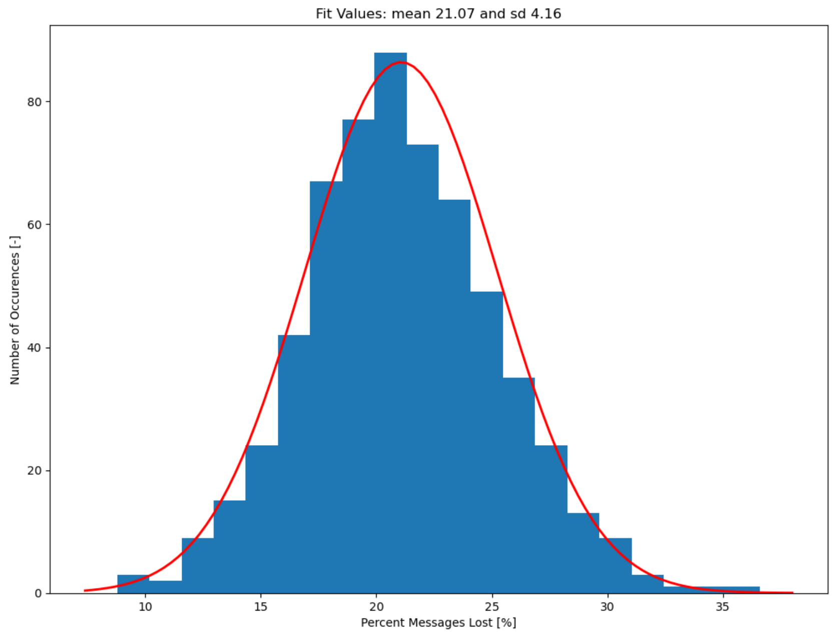


Figure 32: 600 run Overlap Histogram for 50 drones

The mean is at 21.07 % of messages overlapped, with a standard deviation of 4.16.

5.5.4 Urban Airspace: 1 drone per km^2

An urban airspace could be considered to have 100 drones in the simulation area. Here the result is seen in Figure 32.

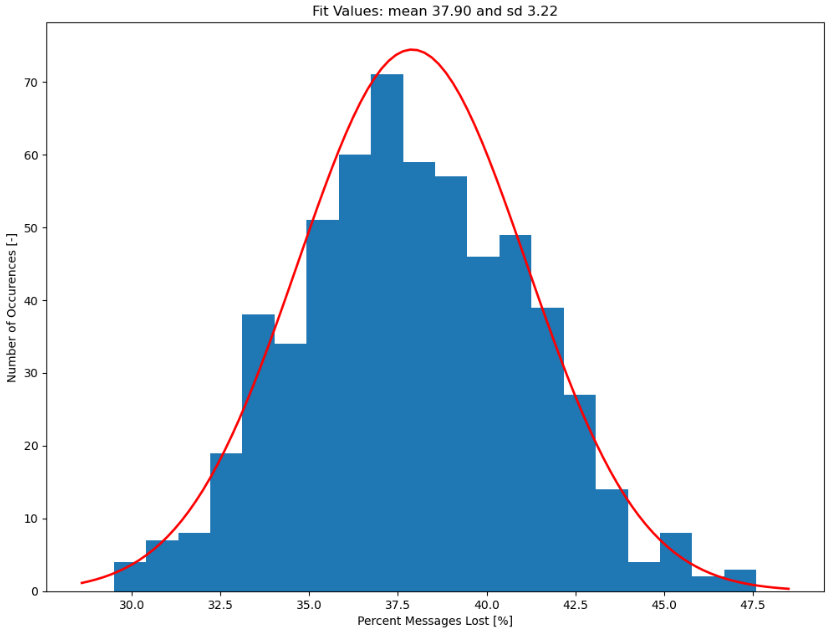


Figure 33: 600 run Overlap Histogram for 100 drones

The mean is at 37.9 % of messages overlapped, with a standard deviation of 3.22.

5.5.5 Busy Urban Airspace: 1.5 drones per km^2

Now take a busy urban airspace, with 150 drones active in the simulation area. The result is seen in Figure 34.

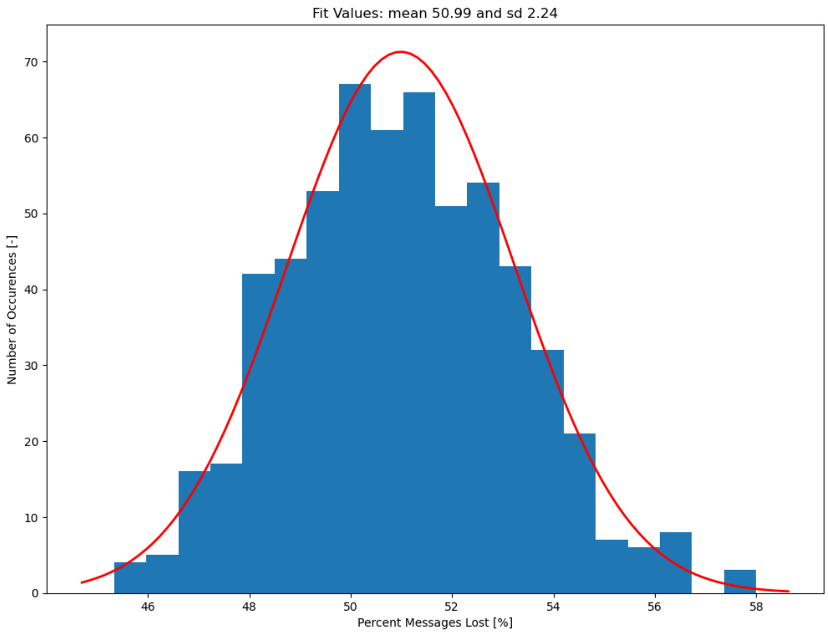


Figure 34: 600 run Overlap Histogram for 150 drones

The mean is at 51 % of messages overlapped, with a standard deviation of 2.24. This is becoming critical, as half the messages that are sent out are overlapping with a minimum of one other message in time and potentially being lost.

5.5.6 Message Overlap In Time: Overview and Comments

To summarise, the outcome of the preliminary message overlap analysis shows the obvious - as more drones are added within the same area, the probability of message overlap increases. While the histograms depicting the proportion of overlapped messages help to visualise the system performance, this metric is incomplete. For each run, the drones most impacted by overlap need to be singled out, and a comprehensive "real update rate" metric will be computed. In reports such as [9], the performance metric referenced is the "98.5% 5-second probability of update range". While range is less important for drones since they operate at low altitudes and are expected to be largely found in urban areas, such a metric could also be constructed. Instead of using a 5-second update as a benchmark, a drone ADS system will benefit from a higher update rate due to closer proximity between drones during operation.

Additionally, the message overlap rate of 51% seen in is alarming. This warrants a sensitivity study - if such dense usage is to be had, one of the following actions need to be taken:

- Utilize a higher data rate than 100 kbps to shorten message length in time
- Shorten bit length of message
- Reduce the message update rate

6. Further Work

Now that the preliminary results have been compiled, the work that will be done in the last segment of the thesis will be outlined below. This will be done in three parts, namely through work related to the range consequences of message overlap, work on the time overlap evaluation and capacity study, finally closing in a section on the hardware verification/proof of concept.

6.1. Work to be Done on Range Consequence Analysis

As seen in subsection 5.4, there are currently not many cases studied due to the time-intensive nature of the analysis. This will be expanded with a look at the following cases:

- Two well-spaced drones with drastically different transmit powers (0.1 W and 0.5 W)
- Three drones with similar power
- Two drones in close proximity (varied and similar transmitter powers)
- Drone far from noise source
- Drone close to noise source

The first three cases are self-explanatory - but will provide a greater insight into the extreme scenarios for message overlap. It is expected that in close proximity, overlap will lead to almost total message loss. Likewise, the range for the drone with the significantly lower transmitter power will be impacted strongest - only the extent remains to be seen.

A larger amount of work will be dedicated to lower-power noise sources operating on the same frequencies. While in information about the frequency occupancy on the unlicensed frequencies is difficult to come by, several scenarios will be considered. The parameters to be varied are the duty cycle of the noise sources, number of noise sources present, and noise source power.

6.2. Work to be Done on Time Overlap Analysis

In subsection 5.5, the time overlap analysis preliminary outcome is presented. As a key metric, it utilises the percentage of messages that have overlapped with another message. While it clearly shows the averages of the messages lost per scenario, and can point to system limits in terms of messages 'lost', a more transparent set of metrics is currently being implemented, and more cases are being considered. Thus, the next steps here are:

- Evaluate the impact of noise sources on same frequency
- Evaluate more cases (more drones, less drones)
- Match overlap rate to a probability distribution, and ensure statistical significance
- Evaluate real update rates for individual drones
- **Perform sensitivity study (message length, update rate, number of drones) and find system capacity/limit**

6.3. Work to be Done on Hardware Testing

The final step will be to test whether such a system is feasible on off-the-shelf hardware. This will utilise a Raspberry Pi Zero, in conjunction with a transceiver board and antenna. How exactly the encoding will be translated to MicroPython is yet to be determined.

7. Conclusions

This report has covered several major topics. Firstly, a literature search and analysis has been performed. The literature collected has been divided into three categories:

1. General Air Traffic Management, ADS-B and Electromagnetic Wave/Digital Communication Systems Theory
2. Drone-specific Automatic Dependent Surveillance - Broadcast (ADS-B) Implementation Literature
3. Literature About Other Types of Drone Surveillance Systems

From this, it was established that the utility of research into the implementation, design and limitations of a drone Automatic Dependent Surveillance system is most definitely justified. While passive systems exist, active "co-operative" surveillance systems where the drone participates in the process are limited. Analysing the SESAR U-Space goals, it appears clear that such a system must be developed, and that the underlying goals are only achievable through co-operative systems.

The goal of this thesis has been re-focused, and is now to construct a Python simulation that can evaluate the capacity of a drone ADS system, ensuring that it can be implemented acceptably in a given 10-by-10 kilometer urban area. This will be done in terms of metrics such as the real system update rate, which takes into account message overlap at the receiver from different drones, as well as the real transmitter range as a function of signal to noise ratio. This step is close to completion, with the simulation code running and yielding data outputs. At this stage, the parts that remain are a comprehensive sensitivity study and more case studies for the message overlap spatial and time scenarios.

The second goal is to provide design constraints, as well as recommendations on carrier frequency, data format and other system parameters, taking into account the capacity and sensitivity study results, as well as the theoretical background from the literature.

Lastly, the proof of concept on hardware will be attempted. This will aim demonstrate that a low-power low-size off-the-shelf implementation for Drone ADS is possible, and potentially encourage further study into the topic.

References

[1] Mode S downlink aircraft parameters implementation and operations guidance document edition 2.0. https://www.icao.int/APAC/Documents/edocs/APX.%20N%20-%20Mode%20S%20DAPsImplementation%20and%20Operation%20GuidanceDocument2.0-20200513.pdf.

[2] Drones: reform of EU aviation safety, Sep 2021. URL https://www.consilium.europa.eu/en/policies/drones/.

[3] Jerôme Bodart. MICA workshop, Sep 2021. URL https://www.icao.int/MID/Documents/2019/MICA/MICA-MID%20-%20WP%2002%20-%20Mode%20S%20Surveillance%20Principle.pdf.

[4] Daniele Cavallo. Guided and Wireless EM Transfer (EE3120TU) Lectures. TU Delft, 2018.

[5] Michael Guterres, Stanley Jones, Greg Orrell, and Robert Strain. ADS-B Surveillance System Performance With Small UAS at Low Altitudes. doi: 10.2514/6.2017-1154. URL https://arc.aiaa.org/doi/abs/10.2514/6.2017-1154.

[6] Prof.dr.ir. J.M. Hoekstra and Dr.ir. J. Ellerbroek. AE4321-15 Air Traffic Management. TU Delft.

[7] ICAO. Technical provisions for mode s services and extended squitter.

[8] Gerard Janssen. Digital Communication Systems (EE3115TU) Lectures. TU Delft, 2018.

[9] Petr Jonas, M. Jancik, S. Holoda, and J. Bodart. Impact of SUAS equipped with ADS-B on 1090 MHz environment. pages 63–67, 2020. doi: 10.23919/NTCA50409.2020.9291095.

[10] SESAR. Ground Based Technologies For A Real-Time Unmanned Aerial System Traffic Management System (UTMS) - CLASS. . URL https://www.sesarju.eu/projects/class.

[11] SESAR. CLASS - Report Document Repository. . URL https://cordis.europa.eu/project/id/763719/results.

[12] SESAR. SESAR: European Drones Outlook Study. . URL https://www.sesarju.eu/sites/default/files/documents/reports/European_Drones_Outlook_Study_2016.pdf.

[13] SESAR. SESAR Smart ATM - U-Space. . URL https://www.sesarju.eu/U-space.

[14] S. Siewert, M. Andalibi, S. Bruder, J. Buchholz, D. Chamberlain, A. Erno, T. Shiroma, and D. Stockhouse. Comparison of radar, passive optical with acoustic, and fused multi-modal active and passive sensing for uas traffic management compliance and urban air mobility safety. volume 1 PartF, 2020. doi: 10.2514/6.2020-1456. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85091891132&doi=10.2514%2f6.2020-1456&partnerID=40&md5=87e976a9ad3bb79c62c9c8f705c04ed5.

[15] J. Sun and J. M. Hoekstra. Analyzing Aircraft Surveillance Signal Quality at the 1090 Megahertz Radio Frequency, 2020.

[16] J. Sun, H. Vû, J. Ellerbroek, and J. M. Hoekstra. pymodes: Decoding mode-s surveillance data for open air transportation research. IEEE Transactions on Intelligent Transportation Systems, 2019. ISSN 1524-9050. doi: 10.1109/TITS.2019.2914770.

[17] Junzi Sun. The 1090 Megahertz Riddle: A Guide to Decoding Mode S and ADS-B Signals. TU Delft OPEN Publishing, 2 edition, 2021. ISBN 978-94-6366-402-8. doi: 10.34641/mg.11.

[18] Sasha Vlaskin. Project Plan: Designing an Automatic Dependent Surveillance System for Drones. 2021.