Preliminary Design and Capacity Study of Automatic Dependent Surveillance for Drones

Vlaskin, A.; Sun, Junzi; Hoekstra, J.M.

Publication date
2022

Document Version
Final published version

Citation (APA)
Vlaskin, A., Sun, J., & Hoekstra, J. M. (2022). *Preliminary Design and Capacity Study of Automatic Dependent Surveillance for Drones*. 1-8. Paper presented at 12th SESAR Innovation Days, Budapest, Hungary.

Preliminary Design and Capacity Study of Automatic Dependent Surveillance for Drones

Aleksandr Vlaskin, Junzi Sun, Jacco Hoekstra
Control and Simulation, Faculty of Aerospace Engineering
Delft University of Technology, the Netherlands

***Abstract*—The consumer drone sector is expected to grow rapidly in the coming decades. In Europe alone, some predictions show as many as seven million drones will be flying by 2050. This poses a challenge for surveillance. In this paper, we study an Automatic Dependent Surveillance system concept similar to the one for current aircraft surveillance, which allows the drone to broadcast information about itself without external input. The study's main contents are threefold. The first consists of recommendations made based on literature. Then, we perform 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. We have demonstrated that such a system is indeed feasible. However, the carrier frequency and code allocation must be changed to prevent interference with the current aircraft's automatic surveillance system. The simulation and capacity study tests the limitation of such a system in high-density scenarios, and provide recommendation for additional work on hardware, format, and modulation techniques to enable such a system. Finally, the hardware test shows that an inexpensive commercial-of-the-shelf implementation with a range of approximately 200 meters is possible, on hardware drawing less than five Watts of power.**

Keywords: Automatic Dependent Surveillance, UAV, U-Space, ADS-B

I. Introduction

The global consumer drone sector is entering a phase of rapid growth, with seven million consumers and 400,000 commercial drones/Small Unmanned Aerial Systems (SUAS) expected to be flying by 2050 in European airspace alone [1]. With new applications such as parcel delivery [2] becoming more viable, drone densities in urban areas will grow, leading to a greater risk of collision between drones and aircraft, 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 [3]), 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, and 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 and it is difficult to implement for drones. Secondly, the current 1090 MHz scenario is threatened by heavy amounts of congestion and transponder over-interrogation [4]. 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.

The main objective of this research is to study the possibility of constructing such an ADS system for drone and tests potential capability and limitations through software simulation and preliminary hardware demonstrations.

II. Background and Previous Work

There are three main literature facets identified for this topic: information on Mode S and ADS-B, the literature on ADS-B implementations for drones, and studies on other drone surveillance systems.

A. ADS-B and Mode S literature

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 most important contents are the ADS-B message specifications and elements which can be used as a reference for the drone ADS.

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 contain 112 bits of information [5], [6]. This structure is shown in Figure 1:

Fig. 1. Mode S Message Structure [6]

where DF refers to the downlink format, CA is the transponder capability, ICAO is the unique 24-bit address, ME is the message field, and PI is the parity index.

B. ADS-B implementation on Drones

An important facet to consider was the evaluation of the effects of direct ADS-B implementation on drones. Studies [7] and [8] evaluate the effects of direct adoption of ADS-B on drones/small unmanned aerial systems. While the 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. Study [7] 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.

C. Related Initiatives, Developments, and Systems

The main project to integrate drone traffic and surveillance across Europe comes from SESAR Joint Undertaking. Within SESAR, the principal of these is U-Space. U-Space is defined to be a set of new services, which all rely on data digitalization and functional automation for unmanned vehicles. This set of services will focus on the safe and efficient integration of dense drone traffic into existing airspace [9]. The main features that were of interest to the study are foundation services that are intended to cover registration and identification of drones, and the advanced set of services that 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.

The U-space initiative encapsulates projects such as CLASS (The Clear Air Situation for uaS) [10], which aim to demonstrate and test potential technologies to enable 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 [10].

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

III. Methodology

In this section, we outline our main methodologies for the simulation, including the power matrix overlap simulation and the signal timing simulation.

A. 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. For more efficient computation, each coordinate is taken as an (X, Y) coordinate into a matrix comprised of nodes on the 10×10 km grid. The resolution can be varied, but the optimum between performance and the number of nodes is found at 1000 nodes per axis or a 10 m grid distance.

For the simulation, it is possible to either define the number of drones and their parameters or 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 and produces an SNR matrix.

For the computation of SNR, the following power equation is often used to account for space loss [11]:

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

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 has several unknowns, namely the directivities, and therefore a model fitted from experimental data is used [12]. This is seen in Equation 2 below.

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

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 the transmitter to receiver. To elaborate, k consists of the following:

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

This result in Equation 3 is especially important, as the SNR model is fitted using data from conventional ADS-B messages [12]. 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 the red line of Figure 2. Here, a low SNR signal is still present at a 200-kilometer range, with a 20 dB value at around 40 kilometers.

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) [13], and therefore a power of 10 mW. The k term is therefore adjusted such that

$$k_d = k - k_{adj} \tag{4}$$

If the correction term is used, the SNR curve for drones (in blue) for a 0.1 W transmitter is visualized in blue next alongside the red curve for aircraft transponders in Figure 2.

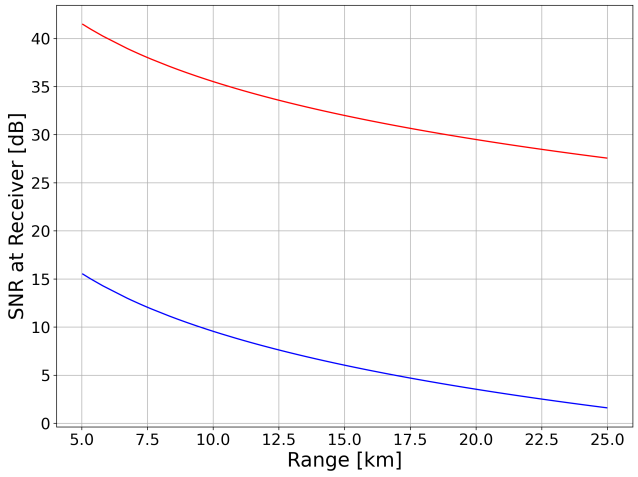

Fig. 2. Models representing the relationship between signal-to-noise ratio and distance to receiver for aircraft Mode-S transponders (red) and drone transceivers (blue)

It is worth noting the model may not fully representative of the real-world environment where the frequency is different from 433 MHz and the noise floor is different from our model.

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:

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

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. This SNR matrix can be seen presented as a heatmap in Figure 3.

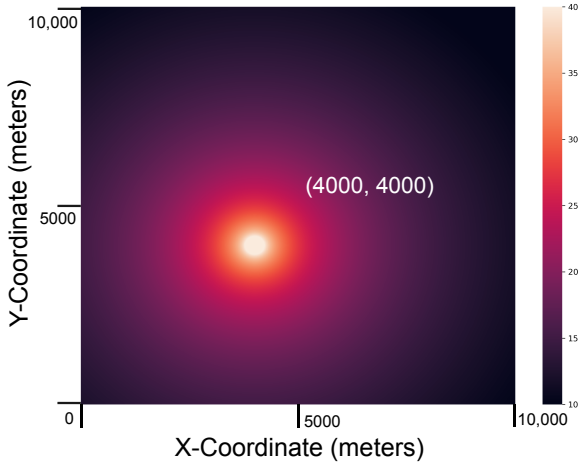

Fig. 3. Signal-to-noise ratio for a single drone at different locations in the grid

When the transmissions of two drones are overlapping, 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:

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

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.

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

B. Message Overlap in Time

Next, we explain the methodology of message overlap analysis. 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 4.

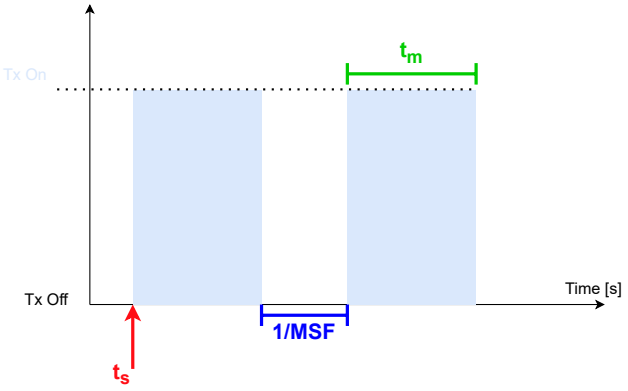

Fig. 4. Ideal model of the timing for the message transmission (not to scale)

In the figure, the light blue peaks represent the messages. Note that the message duration in time t_m depends on two other parameters, with the relation shown in Equation 7.

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

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 variations (called *jitter*), meaning that the clock edges will be offset from their desired locations [14]. In this case, it was deemed interesting to investigate whether artificially induced message time jitter would be useful to mitigate message overlap in time. With jitter, the result of message transmission is illustrated in Figure 5.

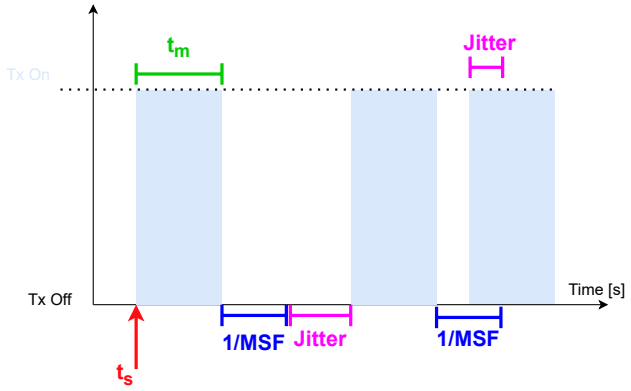

Fig. 5. Model of the timing for the message transmission with jitters (not to scale)

Once the jitter is added, the messages are no longer spaced uniformly in time. It is also worth noting that jitter can be a negative value, as shown in the third message in the figure.

To simulate this, we first consider a user-defined drone density, with message duration, maximum jitter, and message repetition frequency. The simulation model then computes every start and end time of each message, and the jitter is then added. The update rates are calculated across a 5-second simulation window, across several runs to ensure that the results are smoothed for the sensitivity study.

C. 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 another important part of this study.

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 comfortably fit inside any popular consumer drone. There were many options considered, but ultimately the choice was to use a microcontroller coupled with a transceiver module. The Pi Pico is therefore selected and trialed with several different transceiver modules to find the best working combination.

This decision was taken as microcontrollers are closer to lower-level hardware, utilizing low power. It is easy to set up, as the software used (Micropython) is based on Python. The transponder module already handles modulation and simply requires a UART [15] data input to transmit a signal.

The complete setup consists of two Pi Pico microcontrollers coupled to Pimoroni Pico Explorer bases, two HC-12 [16] 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.

After this, a Micropython program is developed, 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. The full experiment assemble is shown in Figure 6

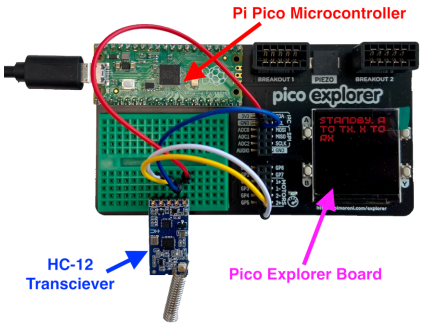

Fig. 6. Single Tx/Rx Unit

It may appear counter-intuitive to connect the **Tx** port to the **Rx** port, but this is done in accordance with the UART protocol [15]. Also note that the SET pin on the HC-12 [16] is not used, as the module works out of the box and the parameters are deemed sufficient for the application. The same procedure applies to be transmitter and receiver, which also means they are interchangeable.

IV. Experiments and Results

We conduct the simulations for both spatial and time overlap with different conditions. Simulation data is collected and used for this analysis. This section first presents the consequences of signal overlap on the SNR matrices of the transmitters involved and followed by the time overlap (garbling) sensitivity analysis.

A. 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 the power-range model proposed in this study, and we then subtract the power of one from the other to obtain the relative SNR.

The first use case demonstrates two drones that are 11,300 meters apart, with 10 mW transmitters each (as per [13], located at (1000,1000) and (9000,9000) respectively. The regions of the 10×10 km where the signal exceeds the signal-to-noise ratios (SNR) of 10, 15, and 20 dB are plotted for

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

both clean transmission and when both drones are transmitting simultaneously.

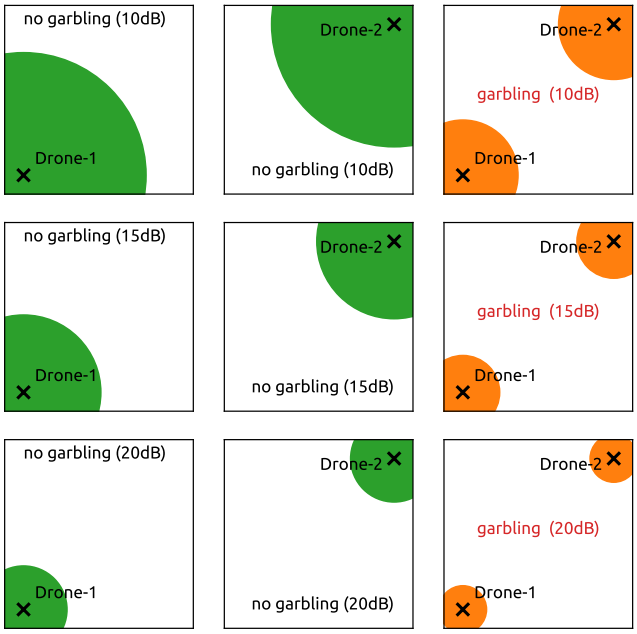

Fig. 7. Decodable range for two drones transmitting at the same power. The signal-to-noise ratio for decoding in each row are set to 10dB, 15dB, and 20dB.

Figure 7 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 green and orange zones are the areas with SNR values past the threshold, whereas the white areas are below the threshold. The rightmost figure 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 where neither SNR is within the set bounds. This is presented as a table in Table I.

TABLE I
DECODABLE RANGES (IN METERS) FOR TWO DRONES WITH THE SAME TRANSMISSION POWER

	Threshold	**Clean**	**Overlap**
Drone 1 (10 mW)	10 dB	6495	2930
	15 dB	4098	1952
	20 dB	2304	1256
Drone 2 (10 mW)	10 dB	6495	2930
	15 dB	4098	1952
	20 dB	2305	1256

Now, it is worth investigating the effects of a large gap in transmitter power. Assume a 10 mW drone located at (4000,4000) overlaps with a 1 W transmitter at (6000,6000). Figure 8 shows the result of this analysis.

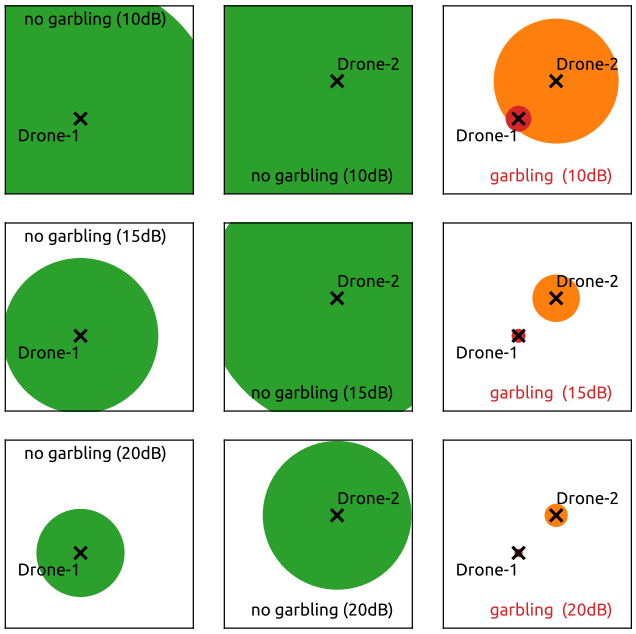

Fig. 8. Decodable range for two drones transmitting at the different powers. The signal-to-noise ratio for decoding set for each row are 10dB, 15dB, and 20dB. When signal garbling occurs, Drone 1 the decodable range for Drone 1 almost not existing.

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

TABLE II
DECODABLE RANGES (IN METERS) FOR LARGE POWER DIFFERENCE BETWEEN TWO DRONES

	Threshold	**Clean**	**Overlap**
Drone 1 (10 mW)	10 dB	7279	646
	15 dB	4098	329
	20 dB	2305	175
Drone 2 (1W)	10 dB	8485	3282
	15 dB	6964	1222
	20 dB	3916	578

Here, we can observe 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 more powerful (such as 1 W) transmitters will cause significant performance degradation to the system if left unchecked.

We also evaluate several other cases, but the general outcome is that the results of overlap are expected to be severe in terms of message loss, and need to be avoided. 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 are likely in dense urban areas.

B. Overlap in Time (Garbling) Analysis

For the time overlap, the overall goal is to investigate what kind of parameters and drone densities would allow for the nominal update rate to be kept (i.e., keep the message overlap in time to a minimum). The simulation is run for a variety of scenarios and the results are analyzed. Figure 9 show hows the overlapping of messages occurs with different density of drones.

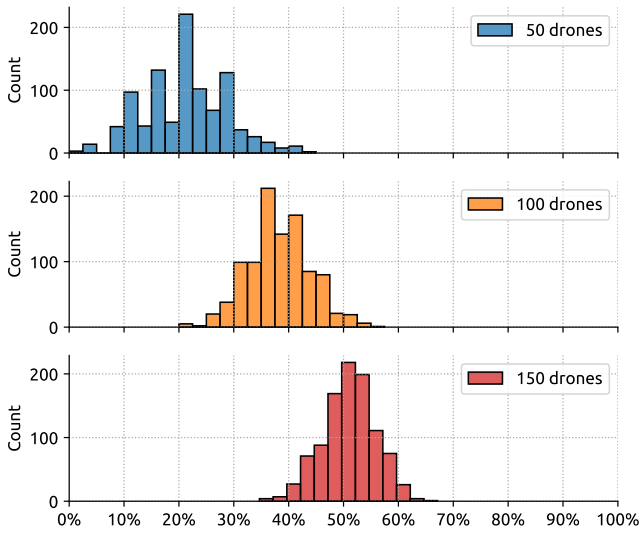

Fig. 9. Percentage of overlapping of messages at different densities of drones

Note that all of these plots utilize a 1.2 millisecond message time (based on the best-case COTS 100 kbps data rate quoted by SeedStudio Grove Serial RF Pro [17]) and for consistency, no jitter is used. We can see 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. Instead of focusing on total error, it is best to focus on maximizing a performance metric - the update rate.

In the following simulations, we further investigate three parameters and their effects on this performance metric. The parameters are the number of drones, the message duration, and the maximum jitter.

Firstly, the number of drones is varied with all other parameters remaining constant. The results are shown in Figure 10. We can observe a clear linear trend, where the update rate drops with the increasing drone density. The update rate drops to below 1.2 updates/second when the number of drones is beyond 100.

Next, we vary the message duration 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 11. Here, the trend is clear. A longer message decreases the update rate, as the probability of overlap with another message grows.

Finally, the jitter is varied. The result of its increase for 110 drones and a 1.2-millisecond message is seen in Figure 12. There is a linearly increasing trend which 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.

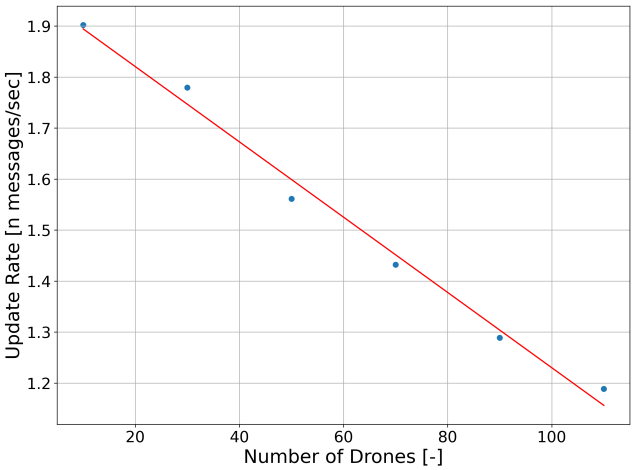

Fig. 10. Varying Number of Drones [-] with Max Jitter [x ML] = 0.1 Message Length in Time [s] = 0.0012

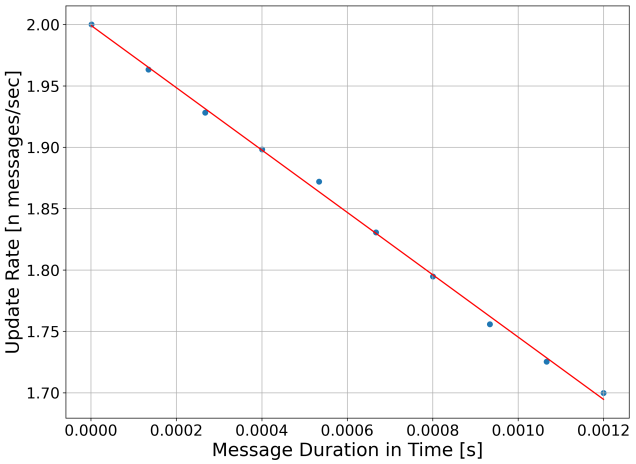

Fig. 11. Varying Message Duration with Max Jitter [x ML] = 0.2 Number of Drones = 110

C. Hardware Testing

The hardware test focuses on testing the range of the system, 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

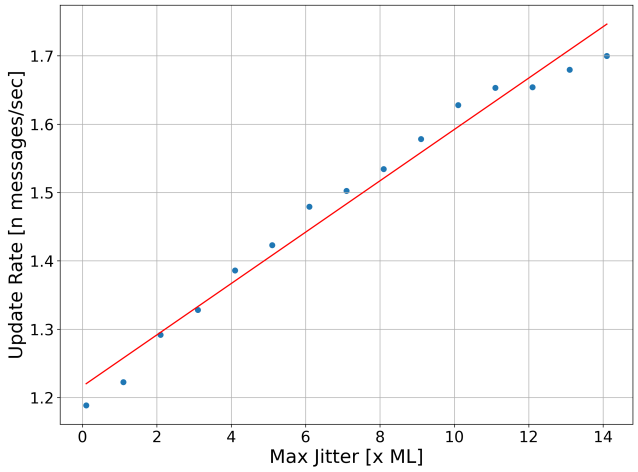

Fig. 12. Varying Max Jitter [x ML] with Message Duration in Time [s] = 0.0012 Number of Drones [-] = 110

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, which are 1) the Message Receive Indicator, 2) Correctly Received Message Rate, and 3) Total Correctly Received Messages.

Screen 1 is displayed by default when no messages are received for a long period, 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 the campus of TU Delft, with the receiver 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 13.

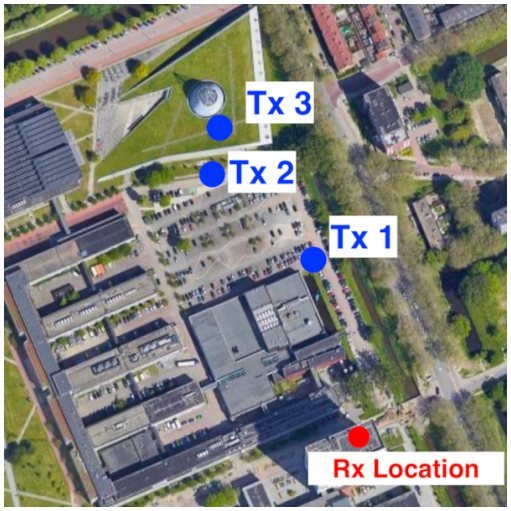

Fig. 13. Map of Rx and Tx positions

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 III.

TABLE III
PERFORMACEN FROM THE HARDWARE TEST

	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**

While several messages were 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 has a range of 1-kilometer [16], which is not the case. This is also possibly due to noise, but checking the spectrum on a Software-Defined Radio (SDR) [18] yielded no significant noise signals around the 433 MHz frequencies.

V. Discussions and 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 recommendations for future research.

A. Message format and simulation parameter selection

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 this paper, it is assumed that the 112-bit Mode-S format was retained. It is recommended to determine what to do regarding format through further research work. While this could be changed for 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 starts.

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 duplication with a 24-bit address is very low.

Another recommendation is to adjust the position encoding algorithm from the current Compact Position Reporting (CPR) used in ADS-B [6]. To correctly decode a CPR-formatted position, both an odd and even message are needed [6]. This could be simplified, as drone ADS ranges are expected to be smaller and a reference position of the receiver can be used instead (if encoded on-drone).

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. It is strongly recommended

to use a transmitter superior to HC-12 [16] 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.

B. System capacity and technological constraints

Transmitter power regulations for these drones stipulate a maximum of 10 mW [13]. It is recommended that the manufacturers select antennae of similar power. Otherwise, the message overlaps heavily impacts the lower-power drone with greater severity, as seen in Figure 8.

Jitter concerning 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.

C. Future research

Based on this study, we identify the following potential avenues of research. 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 metropolitan area.

This work assumed that most of the ADS-B and Mode S message formats would carry over to the drone ADS system, with some changes in the message content. However, a format could be developed from scratch. While keeping the Mode S format may seem a simple choice, it is not tailored for SUAS use. Therefore, it is proposed to design a specific drone ADS message format in the future, taking into account drone-specific needs.

The test hardware presented in this paper demonstrates that it is indeed possible to construct a simple ADS system from off-the-shelf components. The message is validated for correct reception but is not decoded from the hex format - if needed, this needs to be implemented in the future.

Lastly, the testing boards have the capability for sensor interfaces. It would be beneficial to add a GNSS antenna or a barometer and encode and transmit the results to further simulate the ADS system.

VI. Conclusion

This paper shows that the implementation of an Automatic Dependent Surveillance-Broadcast system is possible for the average consumer drone. However, a reduction in transmitter power from what is seen on aircraft transponders means that the message detection range can drop significantly to 10km or below. Furthermore, the carrier frequency must be changed from 1090 MHz to industrial, scientific, and medical (ISM) bands or dedicated frequencies.

In this study, simulations were constructed with two components, one which showed the extent of signal-to-noise ratio degradation on message overlap in time, and the latter the extent of message overlap in time for a given system. The first component showed that messages overlapping in time have strongly adverse effects on both messages, as expected. While two equivalent transmitters at distances of 11 km only reduce the 10 dB SNR 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.

The sensitivity and capacity analysis showed that the message duration is the parameter that constrains performance in high-density usage. We found that a jitter can be used to mitigate the amount of signal garbling and increase the effective update rates for every drone.

Finally, the hardware testing shows that building such a system can be done on a low budget with COTS components.

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] SESAR Joint Undertaking, "European drones outlook study: unlocking the value for europe.," 2017.

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

[3] "Drone Detection Radar, robin." https://www.robinradar.com/drone-detection-radar. Accessed: 09-06-2022.

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

[5] "Mode S downlink aircraft parameters implementation and operations guidance document edition 2.0," 2020.

[6] J. Sun, *The 1090 Megahertz Riddle: A Guide to Decoding Mode S and ADS-B Signals*. TU Delft OPEN Publishing, 2 ed., 2021.

[7] P. Jonas, M. Jancik, S. Holoda, and J. Bodart, "Impact of SUAS equipped with ADS-B on 1090 MHz environment," pp. 63–67, 2020. cited By 0.

[8] M. Guterres, S. Jones, G. Orrell, and R. Strain, *ADS-B Surveillance System Performance With Small UAS at Low Altitudes*.

[9] C. Barrado, M. Boyero, L. Brucculeri, G. Ferrara, A. Hately, P. Hullah, D. Martin-Marrero, E. Pastor, A. P. Rushton, and A. Volkert, "U-space concept of operations: A key enabler for opening airspace to emerging low-altitude operations," *Aerospace*, vol. 7, no. 3, p. 24, 2020.

[10] SESAR, *Ground Based Technologies For A Real-Time Unmanned Aerial System Traffic Management System (UTMS) - CLASS*.

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

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

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

[14] D. E. Comer, *Computer Networks and Internets*. Pearson, 6th ed., 2014.

[15] E. Peña and M. G. Legaspi, "Uart: A hardware communication protocol understanding universal asynchronous receiver/transmitter," *Analog Dialogue [online]*, vol. 4, no. 54, pp. 1–5, 2020.

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

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

[18] C. Laufer, *The Hobbyist's Guide to the RTL-SDR: Really Cheap Software Defined Radio*. Wiley series in software radio, CreateSpace Independent Publishing Platform, 2015.