WIRLESS AVIONICS INTRA-COMMUNICATIONS SYSTEMS AND BAND SHARING

A Thesis

by

JOSHUA THOMAS RUFF

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Co-Chairs of Committee,	Dr. Gregory H. Huff
	Dr. Jean-Francois Chamberland
Committee Members,	Dr. Robert D. Nevels
	Dr. Darren Hartl
Head of Department,	Head of Department

December 2017

Major Subject: Electrical Engineering

1. INTRODUCTION

1.1 Motivation

Over the past two decades, wireless digital communication systems have become ubiquitous in the public life. As the technologies have become more proven, a broad range of players in the aerospace industry developed a significant interest in deploying these systems to electronics on airplanes. Specifically, these companies are interested in radiocommunication between devices on a single aircraft related to the regularity and safety of flight, rather than communications outside an aircraft or for passenger entertainment [1].

Avionics manufacturers are interested in the development, sale and deployment of sensors and devices in areas on a plane that were difficult or impossible to reach with wireless systems. Examples might include placing sensors to monitor a landing gear or internal to an engine, where rotating parts make monitoring difficult [1]. Airframers, Aircraft OEMs and Airlines all have a vested interest in any development which could reduce the amount of copper wiring on planes, thus reducing weight and fuel costs [2]. Regulators are interested in wireless avionics systems from a safety perspective. Critical avionics systems have redundant paths wired in in case of failure, and some or all of these redundancies may eventually be replaced with wireless ones [2] [1]. This type of dissimilar redundancy is always appealing from a safety perspective. The classic example of an engine fire which destroys the physical connection to a controller (and so can't be shut off) demonstrates the utility of a wireless backup [1].

1.2 History

With these diverse motivations spanning across the industry, various aerospace companies sponsored a series of working groups to implement wireless communications systems on aircraft. This systems used in this class of applications were alternatively called Wireless Sensing Networks (WSN's) in early literature, and Wireless Avionics Intra-Communications (WAIC) systems later on. WAIC related projects were sponsored and conducted through the Aerospace Vehicle Systems

Institute (AVSI), which also directed this project. AVSI projects are funded through independent grants known as Authorizations For Expenditure (AFEs)

Three AVSI projects directly relate to WAIC: AFE 56, AFE 73, and AFE 76. AFE 56 studied the feasibility of potential WAIC systems and investigated the suitability of various bands to WAIC applications. AFE 73 took the analysis done by AFE 56 and used the work to advocate to regulators for spectrum allocation for WAIC.

1.3 Project Goals

This work was funded through AVSI and managed under AFE 76. The goal of this project is to perform a band sharing study for WAIC with radio altimeters, and to develop a prototype for WAIC systems. The technical challenges in this project directly result from both the technical studies and the inherently political interactions with regulators performed under its predecessors. Because of this, a brief summary of the work done by the two preceding AVSI projects will be presented here, emphasizing the portions of each most relevant to this study.

1.4 WAIC Feasibility Study

The WAIC Feasibility study was conducted through AFE 56, and the results of this study were published in [3]. The report is summarized here for background with significant focus placed on the search for a suitable WAIC band.

AFE 56 had three primary goals [3]:

- ”Identify, Characterize and prioritize the most significant obstacles currently impeding widespread use of wireless communication in flight-critical functions”
- ”Evaluate the current aircraft RF certification process and suggest possible modifications or changes”
- ”Identify the most promising avenues to certify reliable and robust wireless intra-aircraft data transmission”

Toward these ends, investigations were performed into the certification process, suitable spectrum bands, and security concerns related to the implementation of WAIC systems.

1.4.1 Certification

Any device on an aircraft must go through a regulatory certification process, which functions as a way for regulatory bodies to declare the device airworthy [3]. Both civilian and military aircraft are subjected to various certification processes [3]. The AFE 56 working group surveyed the various standards imposed by the DoD, FAA, and ICAO (International Civil Aviation Organization), as well as international treaties governing the aviation industry. The committee took an in depth look at the flight clearance process in use at various agencies [3].

The AFE 56 working group then looked at the specific challenges brought to the forefront by wireless systems. The primary concerns for potential WAIC systems involved the sharing of spectrum with other legal occupants of the band, as well as intentional and unintentional interferers [3]. It was determined that a certification process for WAIC systems would need to account for and provide mitigation strategies for each of these various potential interferers to pass certification. Information security would also need to be guaranteed for critical systems. These considerations would drive the band selection process for WAIC [3].

1.4.2 The Search for a Suitable WAIC Band

Prior to beginning the search for a suitable band, members of the project management committee held discussions with the FCC to gain insight on the regulator's perspective on allocations for potential WAIC systems. Firstly, FCC staff reccommended AVSI pursue an international spectrum allocation before focusing on domestic rule-making. Secondly, the FCC placed significant emphasis on the importance of "picking a winner" as quickly as possible in the frequency selection process [3].

This recommendation was made in light of experience with previous international radio projects. American industry previously coordinated a global effort to upgrade the Weather Fax system which was delayed by more than two years and ultimately only partially successful. The FCC ultimately pinned these issues on the failure of American industry to ”*socialize* their specific solution” with key international players in the international spectrum allocation process [3]. The lessons from this failure would play as important of a role in the evaluating potential WAIC band as technical considerations. The band would need to be one which aerospace could get.

Initially, the committee considered the Industrial, Scientific, and Medical, or ISM bands. ISM bands are subjected to limited regulations, and were quickly eliminated from consideration for WAIC devices [3]. A wide variety of commercial devices already occupy this band, and these devices are allowed to radiate at relatively high powers. Because of the high operating powers, users are afforded no regulatory protection from hamful interference, a condition which would be unacceptable for the safety focused aerospace industry [3]. For this reason the 915 MHz, 2.4 GHz, 5.8 GHz, 24 GHz, and 61 GHz bands were eliminated from consideration for WAIC devices [3].

To find a suitable alternative, the committee stepped through both the US and international tables of frequency allocations. The committee rated alternatives according to two goals. The first was electromagnetic compatibility with wireless sensor applications [3]. The second goal was that a suitable band already be allocated or have potential to be allocated in a manner compatible with WAIC desired properties [3].

A series of criteria were used to rate the suitability of potential alternatives. A band already primarily allocated to an aeronautical service was considered beneficial from the political perspective of spectrum allocations. Benign co-primary users were considered essential [3]. The less sensitive other occupants are to the minimal level of interference from on-aircraft wireless systems, the better. Bands which possessed common allocations across international regions were also considered beneficial, to ease the process of getting approval for WAIC use of the band [3].

It was considered critical that WAIC systems be sufficiently isolated from ISM and unlicensed allotments. The relatively uncontrolled emitters were considered a significant threat to on aircraft wireless [3]. Isolation from terrestrial point to point systems was also considered critical. These systems introduce the possibility of impinging extremely high radiated power levels onto aircraft that pass through [3]. Although this risk is limited to low altitudes, it constitutes a significant safety hazard that can be avoided by the choice of band [3]. A final consideration for allocations is isolation from Satellite (Earth to Space) allocations. Up-link sites require significant RF power to maintain, and therefore consist of a safety hazard similar to point to point systems [3].

1.4.3 Candidate Bands

Based on these criteria, the AFE 56 committee performed a review of major candidate bands for WAIC systems. The committee provided a synopsis of relevant characteristics of each candidate band and rated the band according to it's suitability. AVSI performed this process with a goal of helping future working groups to prioritize future efforts at reserving spectrum allocation.

1.4.4 Channel Modeling and Security

Finally, the AFE56 committee surveyed two more obstacles to a finalized WAIC implementation. The committee looked at channel modeling for wireless sensor networks and gave an overview of security concerns.

Any implementation of WSN's on aircraft has the potential to be critical to the safety of flight. Because of this, the committee stressed the importance of developing a validated channel model for the band and air-frames in use [3]. The channel models would allow for the incorporation of the physical propagation characteristics of the wireless signals in various aircraft and could be used to improve the reliability of real WAIC designs. Because of this, the committee provided an overview of channel modeling efforts in their report and made recommendations for an approach to channel modeling efforts that might follow a new WAIC allocation [3].

Lastly, the committee commissioned a follow up investigation which looked into the security concerns associated with WAIC systems. A report [4] was commissioned through the University of Minnesota, which aimed to analyze the various potential threats to wireless networks on aircraft. The threat vectors considered included physical layer attacks such as jamming, as well as higher layer attacks such as distributed denial of service and hacking risks. The report then looked at potential mitigation strategies for each of these threat vectors. The solutions listed were meant to be a comprehensive overview, but to acquire certification each device would have to provide a detailed overview of their implementation to the relevant certification authority [4].

1.4.5 Summary and Conclusions

The AFE 56 project committee performed a feasibility study for wireless sensing network on aircraft. The committee first looked at the existing path to certification for instrumentation, and came to the conclusion that this path would work for WSN's as well, provided that the applicant for certification perform the necessary extra step of explaining to the FAA the added risks of the wireless device and how these risks were mitigated [3]. The committee then performed an in depth survey of potential bands for WSN use, summarizing the desirable characteristics possessed by any candidate band. The committee provided an in depth overview of the pros and cons of each serious candidate for WAIC allocation, a brief summary of which has been included in this report for reference. Finally, the committee looked at potential channel modeling techniques and security concerns associated with wireless systems on aircraft and outlined how these would need to be addressed for a real WAIC implementation [4].

The committee came to the conclusion that although there were numerous hurdles in the way of fully realizing WAIC systems, WAIC systems were feasible and these challenges could be overcome with industry expenditure and effort. The tasks necessary for WAIC implementation were as follows:

- Acquire spectrum for WAIC use
- Perform a band sharing study for WAIC and existing band occupants
- Develop industry standards for channel modeling of air-frames
- Develop industry standards for addressing security concerns for wireless networks on aircraft
- Work with regulators to develop a streamlined certification process for wireless sensing networks once these standards are developed

With the feasibility study complete, the AVSI partners moved on to the task of acquiring spectrum for wireless networks on aircraft.

1.5 Selecting a Suitable WAIC Band

In AFE 73, AVSI followed up on the work completed in AFE 56. The list of candidate bands was narrowed down to three at the World Radio Conference in 2015 (WRC-15), which then were the subject of a compatibility study. This process concluded with the final selection of a final WAIC band which was approved at WRC-15.

1.5.1 Assessment of Bands between 960 MHz and 15.7 GHz

At WRC-12, the ITU recommended in Resolution 423 that bands from 960 MHz to 15.7 GHz be considered for potential WAIC allocation [5]. The ITU eliminated bands below this range because the antenna size requirements were incompatible with WAIC implementation requirements. Bands above this range were to be considered only after all possibilities in this range were exhausted.

The ITU took all bands with existing allocations to aeronautical mobile or aeronautical radionavigation services into consideration. Bands in this range which fit this criteria were considered in an initial assessment. The purpose of the initial assessment was to eliminates bands with undue burden of a regulatory or technical nature. Technical burdens could involve an excessive number or difficulty of necessary band-sharing studies, while regulatory burdens could involve co-occupants of a band outside of the aerospace industry. After the initial assessment, the 2.7-2.9 GHz, 4.2-4.4 GHz, and 5.35 - 5.46 GHz bands were considered the three most promising candidates for a potential WAIC Allocation. This recommendation precipitated a theoretical compatibility study for these bands at WRC-15 [5].

1.5.2 Compatibility Study

The three bands selected in the initial assessment for more detailed analysis underwent a theoretical compatibility study. Existing ground based radar systems in the 2.8 and 5.4 GHz bands were found to be incompatible with the requirements for WAIC implementations, which left the 4.3 GHz band as the only remaining option. The ITU recommended this band for allocation for WAIC Systems [5].

1.5.3 Relevant WRC-15 Allocations

The 2015 World Radio Conference (WRC-15) made changes to the spectrum allocations in and around the radio altimeter (RA) band [6]. The 4.2-4.4 GHz band received an allocation for WAIC systems pending the experimental verification of compatibility. Additionally, new allocations for 5G systems in the 3.7 GHz (3600-4200 MHz) and 4.5 GHz (4400 - 4900 MHz) bands directly adjacent to the altimeter band would lead to experimental band-sharing studies.

1.6 Overview of Radio Altimeter Functionality

The allocation of the 4.3 GHz band for WAIC necessitates an experimental study on the effects of interference on radar altimeters. This section provides background information on the functionality of radar altimeters which is relevant to the experimental design and setup.

1.6.1 Basic Overview and Applications

The 4.2-4.4 GHz band was previously allocated exclusively to radio altimeters and transponder systems associated with altimetry. The altimeter functions to actively and continuously provide height measurements of an aircraft above the surface of the Earth [7]. The highest degree of accuracy is expected in the approach, landing, and climb phases of flight. This accuracy must be maintained through all types of ground reflectivity. The height measured by an altimeter has a variety of uses in safety critical systems. The height functions as an input The Terrain Awareness Warning System, which gives the pilot a ”Pull Up” warning at a predetermined unsafe altitude and descent rate. The height from altimeters also functions as input for Collision Avoidance, Weather, Navigation, and Autopilot systems. Radio Altimeters are expected to operate in these functions through the lifetime of the Aircraft they are installed on, which results in Altimeters used in excess of 30 years.

1.6.2 Calculating the Height From a Time Delay

There are two primary types of altimeters in use today. The first are Frequency Modulated Carrier Wave (FMCW) Altimeters [7]. FMCW altimeters use a transmitter and receiver with separate

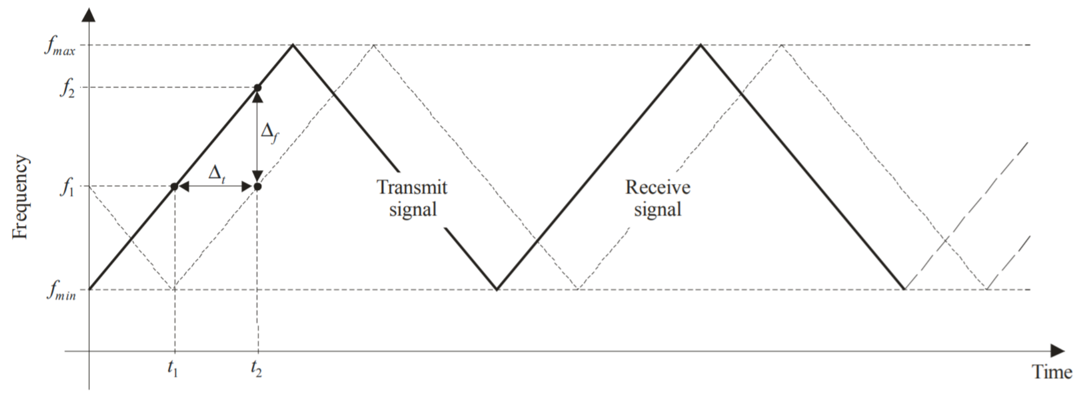

Figure 1.1: FMCW Waveform from ITU-R M.2059-0 [7]

antennas. The signal from the transmit antenna travels to the ground, is reflected, and returns to the aircraft. Due to the constant propagation speed, the return time of the signal is proportional to the height of the aircraft above the surface.

The signal travel time is based on the return of a signal of the same frequency as the transmit signal [7]. One method for calculating the travel time of a signal involves taking the difference between the frequency of the return signal at the current time and the frequency of the transmit signal at the current time, Δf. As shown in Figure 1.1, given a constant waveform, the return time of a signal is:

$$\Delta t = \frac{\Delta f}{df/dt}$$

Once Δt is calculated, it the height can be determined using the speed of light:

$$H = \frac{c}{2\Delta t}$$

While not relying on a frequency modulated waveform, pulsed radar altimeters use a series of discrete pulses to track the current height of the aircraft. The Δt between two pulses is used to calculate the height in the same manner that an FMCW altimeter does. Thus, for any altimeter under test in a lab setting, the time delay of the signal between the Altimeter TX and RX must be simulated to provide an accurate representation of real conditions.

1.6.3 Attenuation of the Altimeter Signal in Free Space

A signal traveling from Altimeter Transmit and back to receive passes through multiple different sources of gain and attenuation [8]. There is attenuation from cable losses, gain from the TX antenna, free space path loss as the signal travels toward the ground, loss from the scattering of the signal by the ground, path loss of the return signal, a gain from the receive antenna, and finally the attenuation from return cable losses. The combination of each of these gains and losses comprises the external loop-loss L for a signal leaving an aircraft. DO-155 defines the loop loss as the ratio of the power received by the RX antenna, P_R to the power sent by the transmit antenna, P_T.

$$L = \frac{P_R}{P_T}$$

The DO-155 standards specify loop loss for different heights, standardized antennas, ground scattering environments, and standardized cable attenuations, and expands the formula shown here to derive these [8]. Like the time delay, any Altimeter lab setup has to simulate the DO-155 attenuation for various heights to be realistic.

1.6.4 Conclusions

Radio Altimeters are a safety critical system in any aircraft, the output of which is used by other important airborne systems. Altimeters use the time it takes a signal to travel to the ground and back to calculate the height of an aircraft off the ground, and must be able to pick up a return signal which has been attenuated significantly depending on the height. To test radio altimeters in a lab setting, both the time delay and attenuation experienced by a real signal must be simulated.

2. METHODS

2.1 Basic Altimeter Test Bed Setup

In addition to laying out performance standards for radar altimeters, the DO-155 standards [8] also specify a basic test setup for verifying an altimeter is functioning properly. Figure 2.1 shows the diagram of this test setup, which connects the RF terminals of the altimeter to an *altitude simulator* and connects a separate device to read altitude data.

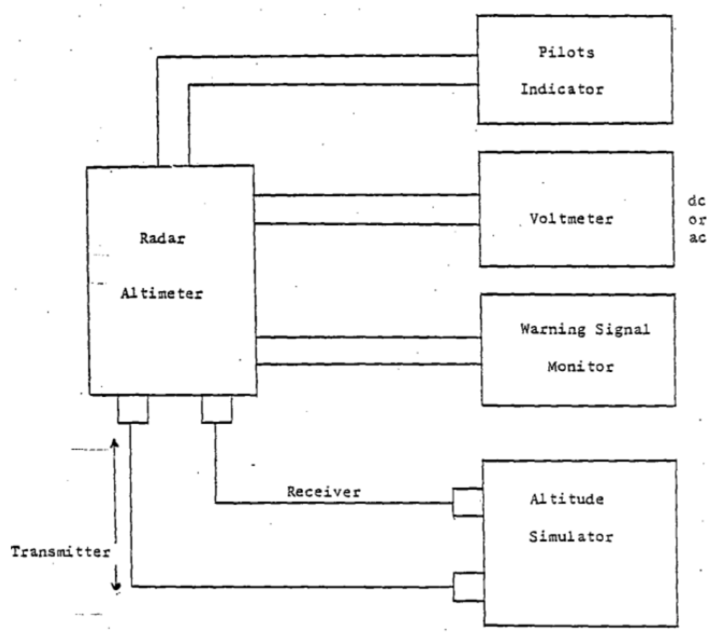

Figure 2.1: Basic Altimeter Test Setup from DO-155 [8]

The standards elaborated on the necessary characteristics of the most critical part of the testbed, the altitude simulator. The altitude simulator needed to ”consist of variable and fixed RF attenuators” [8] to simulate the loop loss an altimeter experiences aboard an aircraft (see section 1.6.3). The altitude simulator also needed a length of ”coaxial cables or other suitable delays” [8] to simulate the physical time delay experienced by an altimeter signal between the transmitter and receiver (see section 1.6.2). To complete the test setup, the altitude simulator directed the attenuated and delayed RF energy from the transmitter fed back into the receiver.

Additionally, the standards specified that any test equipment must account for cross coupling between transmitting and receiving antennas. DO-155 emphasized that the altitude simulator should achieve the desired altitude within 1% and the correct attenuation within 2.5dB [8].

2.2 Modified Altimeter Test Setup

AVSI designed a modified version of the altimeter test setup specified by DO-155, shown in Figure 2.2. The modifications allow the controlled injection of interference into the line after the altimeter signal passes through the altitude simulator.

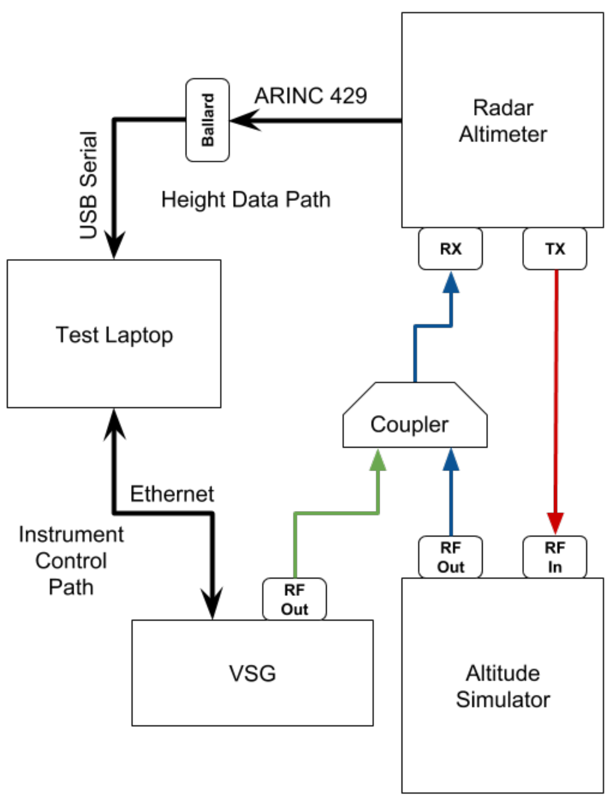

Figure 2.2: Modified Altimeter Test Setup

2.2.1 Reading the Altimeter Output

The altimeter outputs labeled height data on a standardized ARINC 429 cable configuration. The modified setup uses a Ballard ARINC device to convert the data from ARINC 429 to USB serial format, providing each data point with a time stamp. On the test laptop, Ballard CoPilot software reads the serial data and provides a display which allows the real time monitoring of all altimeter output and labels.

The labels are critical because some data points may be labeled NCD (No Computed Data) when conditions are insufficient for a reliable height measurement. CoPilot software also allows for the easy export of test data to Microsoft Excel documents for post processing.

2.2.2 Implementing the Altitude Simulator

2.2.2.1 Time Delay

Different test altitudes require the use of different methods of delaying the RF energy output by the altimeters. For higher altitude tests, spools of fiber optic cables create a time delay. The RF output from the altimeter transmitter was fed by coax connection to the fiber optic transceiver, which could either pass the signal to a single fiber optic spool or a series of cascaded spools to achieve a desired height. This setup contained optical spools of 500, 1000, 2000, and 4500 feet, each of which could be used individually or in conjunction with any or all of the other spools to implement a delay.

The optical transceiver and cascaded spools also contribute an attenuation to the loop loss which varies based on the number of spools cascaded. A single spool setup has an attenuation verified experimentally to be 29dB, with an additional 2dB loss added for each additional cascaded spool.

Later tests modified this delay setup to test an altimeter in takeoff and landing scenarios. The much lower height in these scenarios made spools of coax sufficient tor the delay instead of fiber optic cables. Two coax spools provided a height of 40 ft and 95 ft for testing these scenarios, with a 6dB and 36dB attenuation contributed to the loop loss respectively.

Height	Loop Loss
40ft	76 dB
95ft	84 dB
500ft	100 dB
1500ft	109 dB
3000ft	116 dB
5000ft	120 dB
8000ft	124 dB

Table 2.1: DO-155 Loop Losses

2.2.2.2 Achieving Standard Loop Losses

DO-155 specifies loop loss for various heights and antenna types. Table 2.1 lists the loop loss used for each height in these tests. To achieve the Loop Losses specified by DO-155 standards for each height, the attenuation inherent in the delay method used for each height must be taken into account.

Once the attenuation from the delay line is subtracted from the loop loss, 10, 20 and 30dB Pasternack fixed attenuators inserted into the setup get within 10dB of the desired loop loss. These are located strategically within the setup in part to protect the fiber optic transceiver from damage. Finally, a step attenuator capable of 1 to 11 dB is used to achieve the desired loop loss with a 1 dB precision.

2.2.3 Generating Interference Signals

A Rhode and Schwarz SMU200A Vector Signal Generator (VSG) is used to generate simulated WAIC signals of varying modulation types, bandwidths and power levels. The VSG has a SCPI interface which allows an external computer to control any functionality on the instrument through commands sent over either a serial or an Ethernet connection.

The AFE76 Project Management Committee (PMC) chose the modulation formats to represent theoretical WAIC interference. The PMC chose to subject the altimeters to an MSK waveform, as well as OFDM waveforms of varying bandwidths and dual versions of both waveforms. Each waveform used ”junk” data to modulate the carrier, which consisted of randomly generated ones

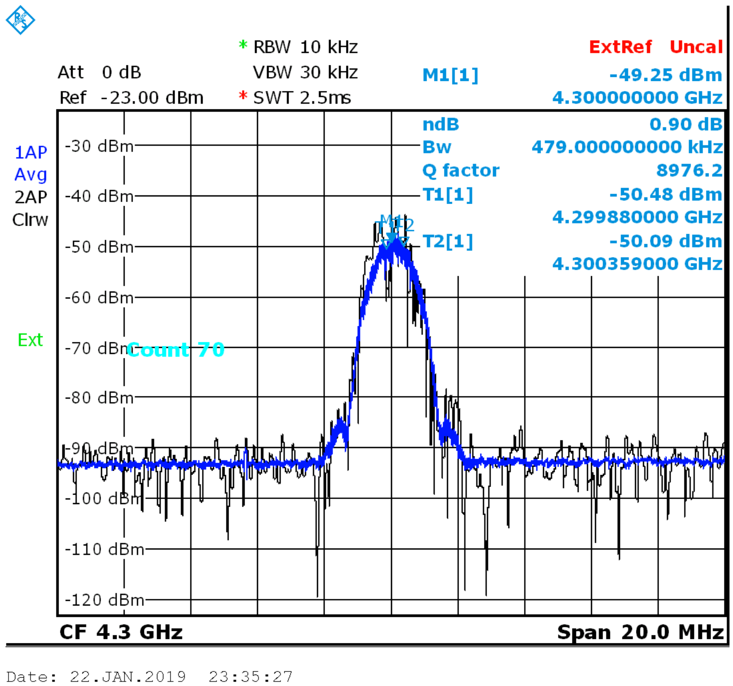

Figure 2.3: MSK Waveform at 4.3 GHz Pictured on a Spectrum Analyzer

and zeros with equal probability of either; this closely replicates the performance of an optimized communication system [**?**]

2.2.3.1 MSK Waveform

MSK or Minimum-Shift Keying is a type of modulation format which can be considered as a form of Phase-Shift Keying (PSK) or as a special case of Frequency Shift Keying (FSK) [9]. The frequency separation of an MSK signal, Δf is:

$$\Delta f = \frac{1}{2T}$$

and thus it has a modulation index of 1/2. This is ”the minimum frequency separation for orthogonality of the two sinusoids” [9]. An MSK waveform is shown on the spectrum analyzer screen-cap in Figure 2.3.

MSK modulation is available natively in the VSG software through the *custom waveform* interface. The Rhode and Schwarz VSG provides a number of common modulation options in this interface for the user to choose from. This simplifies the waveform generation for this case, as every in-built functionality on the VSG has a SCPI command corresponding to it.

2.2.3.2 OFDM Waveform

Orthogonal Frequency-Division Multiplexing or OFDM attempts to achieve an efficient, wide-bandwidth communication system by ”dividing the channel bandwidth into equal-bandwidth sub-channels, where the bandwidth of each sub-channel is sufficiently narrow so that the frequency response... of sub-channels are nearly ideal” [9]. An OFDM waveform is shown on the spectrum analyzer screen-cap in Figure 2.4. One way to view the OFDM waveform is as a series of MSK

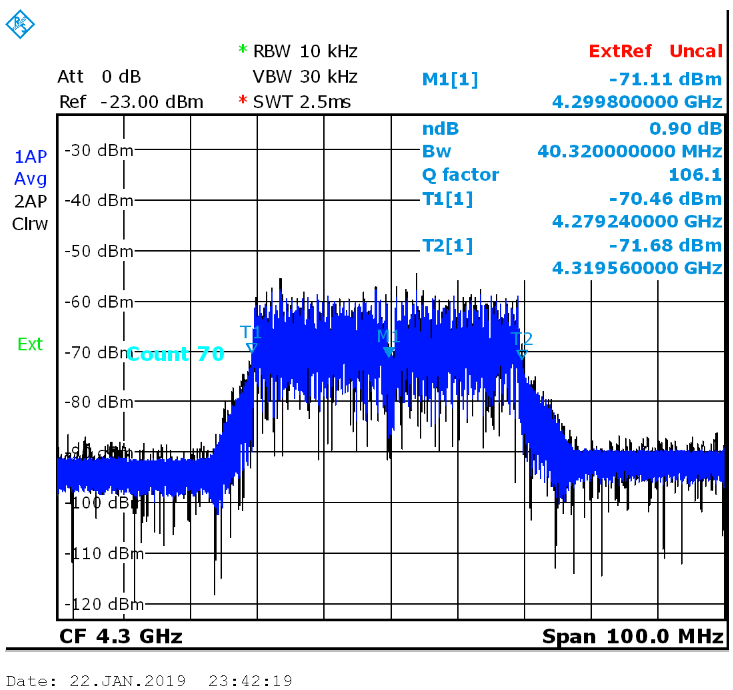

Figure 2.4: MSK Waveform at 4.3 GHz Pictured on a Spectrum Analyzer

waveforms spread out orthogonally along a desired bandwidth. From this perspective, MSK can be seen as the minimum bandwidth version of a system based on OFDM, and the altimeter can thus be subjected to wider and wider bandwidth systems to show the response to a greater number of WAIC devices on (or external to) an aircraft. This aspect, as well as the similarity of an OFDM signal to LTE systems made it a very attractive option for testing the impact of WAIC.

Generating OFDM signals with the VSG was a more involved process than generating MSK, because the functionality for creating OFDM was not available natively in the VSG software. Because of this, OFDM had to be generated through the VSG's *Arbitrary Waveform Generator*. The arbitrary waveform generator enables the VSG to generate any waveform from IQ data stored in a file on the VSG. The SCPI command for generating an arbitrary waveform must include the file path for this IQ data.

This enabled an OFDM waveform to be specified by a Matlab script which exports raw IQ data, which is then converted to a form readable by the VSG through proprietary Rhode and Schwarz software. The conversion software allowed the user to adjust the clock rate, thus adjusting the bandwidth of the signal used. The bandwidth could then be measured with a spectrum analyzer to verify the process has yielded the expected waveform. The bandwidth of the OFDM waveform was defined as being 6 dB down from the peak power.

2.2.3.3 Dual Waveforms

The VSG allows full control of an RF generator along with two baseband generators. The RF generator gives the user control of RF carrier frequency as well as the output power level of the carrier in dBm. The baseband generators allow the modulation of two potentially unique waveforms onto the carrier wave, with a possible offset frequency from the center. Both dual MSK waveforms and dual OFDM waveforms of bandwidth less than 40 MHz were tested using a +/-20 MHz offset between them.

Dual waveforms were considered an important option for testing due to a special characteristic of some altimeters at low altitudes. As the plane descends, the sweep rate of an altimeter increases to give more frequent readings at a more safety critical phase of flight. This is done by significantly reducing the Δf covered by the altimeter FMCW. The effects of this offset were thus investigated to determine whether WAIC's impact could be minimized at critical altitudes by leaving the center of the band clear.

2.3 Python Test Software

Python code written by the author pieces together the various parts of this setup into a complete test bench. Different Python scripts handle different functions necessary to the test bench. These include:

- Creating a database to transfer data between different Python Scripts
- Interface with the VSG to generate interference signals
- Parse the CoPilot log
- Mapping time stamped height measurements to time stamped interference signals
- Plotting and analyzing the results

The software uses Standard Commands for Programmable Instruments or SCPI to control the vector signal generator. In this test bed, the SCPI instructions are processed through an object-oriented hierarchy shown in Figure 2.5. The super-class, $SCPI$ interfaces directly with all lab equipment. The subclass, called *RS_Signal_Generator* in this implementation, contains python functions associated with all instrument specific commands. The helper functions from $SCPI$ send and receive communication with the instrument. Finally, the main loop exists at the highest level, which times the calls of different instrument commands and creates a database to store them. The idea behind this abstraction is to prevent the main loop from dealing with any raw SCPI command syntax– this should only be handled by one of the instrument classes.

2.3.1 Test Main Loop

The highest level of this design is the test main loop. The test main loop creates the SQLite database which stores all important information for easy transfer between the different python scripts necessary for the test bed. This program also contains variables for various test parameters, which are stored in an SQLite database for easy reference and sometimes directly control the sequence of a test. Finally, this program loops through the sequence interference signals specified by

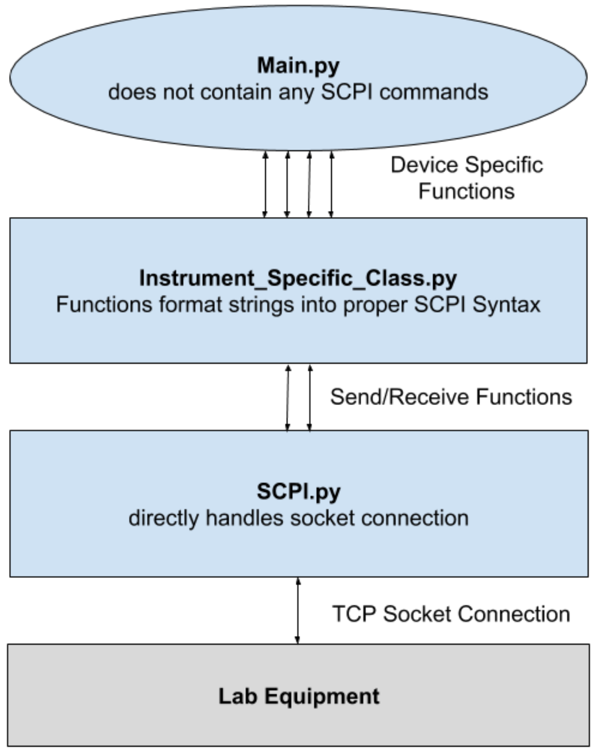

Figure 2.5: SCPI Class Hierarchy

the various test parameters, and sends the commands to the VSG to generate them. The commands sent to the VSG are time stamped as precisely as possible, and recorded in the *Generated Signals* table in the database.

Certain test parameters are stored for reference or calculation but do not directly affect the sequence of interference signals to be generated. These include the altimeter make and model, the nominal height of the test setup, the loss experienced by interference signals traveling to the altimeter RX, and the loop loss used in the setup. Other parameters directly control the sequence of the test, including interference on and off times, power levels to be used, modulation formats to be tested, and RF carrier frequencies to be used.

2.3.1.1 *Nominal Height vs Correct Height*

Nominal height is the term used to refer to the approximate altitude of the test setup experienced by different altimeters. The best way to illustrate this is with an example. The smallest spool of fiber optic cable may have a physical length of around 1500 feet, and altitude readings with no interfernce may vary from around 485 feet to around 515 feet. Because a single height needs to be chosen to derive the Loop Loss, and the spool was expected to provide approximately 500ft of delay, the test setup using only this spool in the altitude simulator is said to have a *nominal height* of 500ft.

The difference between measured height and the nominal height of the setup varies between the different altimeters, and is primarily a result of different calibration settings for each altimeter. The calibration procedure is an important part of installing an altimeter onto an aircraft. When an aircraft is on the ground, the TX and RX antennas used by the altimeter are naturally several feet off the ground, in line with the airframe. Additionally, there are standardized delays from the altimeter TX port to the TX antenna and from the RX antenna to the RX port. These are known as Aircraft Internal Delays (AIDs). To compensate for the varying heights of airframes, as well as the AID installation, avionics manufacturers developed a calibration procedure so that each altimeter could be programmed upon installation to output an altitude of 0ft when the plane is on the ground.

Because these tests are only concerned with *height error*, rather than the most precise height measurement possible in the setup, this calibration is not corrected for in test procedures; *nominal height* is only used to set the loop loss. Instead a variable called *correct_height* is calculated in post processing as the median altitude with no interference. Any height error attributable to interference is measured as a distortion from this correct height.

2.3.1.2 *Sequence Control*

The primary purpose of the test main loop is to subject an altimeter to various modulation formats, gradually stepping up the power of each until the altitude readings from an altimeter are distorted or broken. The main loop determines the type of modulation, power level, and timing,

ID	Altimeter	Start Time	End Time	Modulation	Carrier Freq	Power	RF State
1	Alt A	19:23:06	19:23:07	MSK	4300 MHz	-10 dBm	OFF
2	Alt A	19:23:07	19:23:08	MSK	4300 MHz	-10 dBm	ON

Table 2.2: Example Interference Signals Table

and as each signal is turned on or off by the VSG, stores the parameters for the signal in the *Interference Signals* table for use in post processing, an example of which is shown in Table 2.2. Each unique modulation format and power combination will have two entries in the *Interference Signals* table, corresponding to the RF ON and RF OFF states of the VSG.

A variety of parameters controls the progression of different interference signals. The *interference_duration* and *signal_off_duration*, define the length of time the altimeter will be subjected to a particular interference signal, as well as the length of time the altimeter will have to recover from any error caused by the previous signal. Throughout the main loop, each signal's Start Time and End time is calculated using interference duration variables.

A range of RF powers is specified using *power_min*, *power_max*, and *power_step*. For a given modulation format, the main loop iterates through each power level in this range, subjecting the altimeter to this interference power. This allows the user to step through increasing power with as much granularity as is desired for the test

The final variable which is important to the progression of a test is a list called *modulation_formats*, which contains strings corresponding to the different modulations the VSG will generate. MSK and OFDM signals of varying bandwidths will be listed here. The main loop iterates through each string in this list, passing the string to helper functions.

The first helper function is a lookup which calls the proper function in the VSG class corresponding to a specific modulation format string. The second helper function gives the option for different modulation formats to be put on different carriers, or a list of different carrier functions. Iterating through different carriers for different modulation formats proved to be a critical functionality in later tests.

ID	Timestamp	RF State	Power	PEP	Carrier	Offset 1	Custom 1	Arb 1
1	11:49:39.96	1	-3 dBm	4.48	4300 MHz	0	0	1
2	12:09:40.26	0	-3 dBm	4.48	4300 MHz	0	0	1

Table 2.3: Example VSG State Table

2.3.1.3 Precision of Timing Commands

During initial tests of the main loop, problems occurred which prevented the *interference_duration* and *signal_off_duration* from precisely controlling the duration and recovery time of an interference signal. This section provides an overview of the different timing issues encountered over the course of these tests, as well as the approaches taken to mitigate each issue.

The first and most serious issue encountered while running these tests were termed *hanging delays*. These occurred when the main loop progressed more or less as expected, but the vector signal generator would ”hang” in its previous state for a significant period of time after the command was sent.

First, the program was modified and a new table was added to the database to store the Vector Signal Generator state after each set of commands. Since the purpose of this table was only to be used for debugging, entries were kept as simple as possible. For example, a 1 in *RF State* corresponds to RF ON, while conversely a 0 in *RF State* corresponds to the RF OFF setting. The *Power* and *Carrier* columns correspond with the same values in Table 2.2. *Offset 1*, *Custom 1*, and *Arb 1* correspond to the state of the first Baseband generator, telling whether the *Custom Waveform Generator* corresponding to a MSK Waveform (see Section 2.2.3.1) or the *Arbitrary Waveform Generator*(see Section 2.2.3.2) corresponding to an OFDM waveform are active respectively.

For the purposes of the timing problem, these attributes serve as a fingerprint, allowing an investigator to correlate each row in the VSG State Table to a row in the Interference Signals table. While information here may seem redundant, this value differs notably from the values in the interference signals table in that the VSG state table is only populated *with values received from the Vector Signal Generator*. Because of this, this table allows for a simple method for looking through the data to find at which point the VSG might ”hang” in its current state rather than switch to the next signal as specified in the interference signals table.

Using this method, different theories for potential causes of the hanging delays were investigated. The first potential problem was dropped or delayed packets. During the initial setup, instruments were connected to one another over the TAMU network. While this functioned at a high level, upon packets had an unnecessarily long mean travel time, and significant outliers existed during times of heavy traffic. These issues were solved by moving the VSG and controller laptop onto their own local network.

While the local network solved the biggest chunk of hanging delays, it did not eliminate them entirely. These were attributed to a slowdown in the processor handling the sequence of Python commands.These remaining delays would only occur at times of extremely heavy CPU usage (possibly caused by a memory leak in the old Ballard driver), and were eliminated almost completely by rebooting the controller laptop more frequently. Later on, an upgrade to the Ballard driver reduced these even further.

While the hanging delays were investigated, the VSG State table revealed another source of imprecision in timing commands. It was noticed that the VSG State would not change until nearly a second longer than the *Interference Duration* or *Signal off Duration* parameters specified. This drift occurred because the main loop used Pythons in built *Sleep()* function to handle timing. The sleep function used the exact interference duration passed as a parameter. After the program ’woke up’ the commands to set the next interference signal up would be called, thus introducing a small delay beyond the specified duration.

This problem was dealt with by replacing the *Sleep()* function with a custom function called *Wait_Until()*. This new function takes the signal end time as a parameter (See Table 2.2), subtracts the current time from the end time, and uses the new value as the parameter for Python’s in-built sleep function. This allows the next end time to be calculated in advance by adding the *Interference Duration* or *Signal off Duration* to the previous signal’s end time, and storing the previous signal’s end time as the start time for the next signal. [This section definitely could use some sort of diagram...]

Using the *wait until* method to time commands offered important new flexibility to the main loop. This method allows writing signals to a database, setting up the baseband generators, and pinging the VSG for the VSG State information all to occur while the VSG is in an RF OFF state. While the VSG settings can not be adjusted in RF ON while maintaining the integrity of the signal, writing to the database and pinging the VSG for state information can also be done during this state, and the *wait until* function will still wake the program at the proper time to turn the VSG on or off. The implementation of the *wait until* function eliminated the drift problem, and allowed for the *Interference Duration* and *Signal Off Duration* parameters to precisely control the signal on and off time.

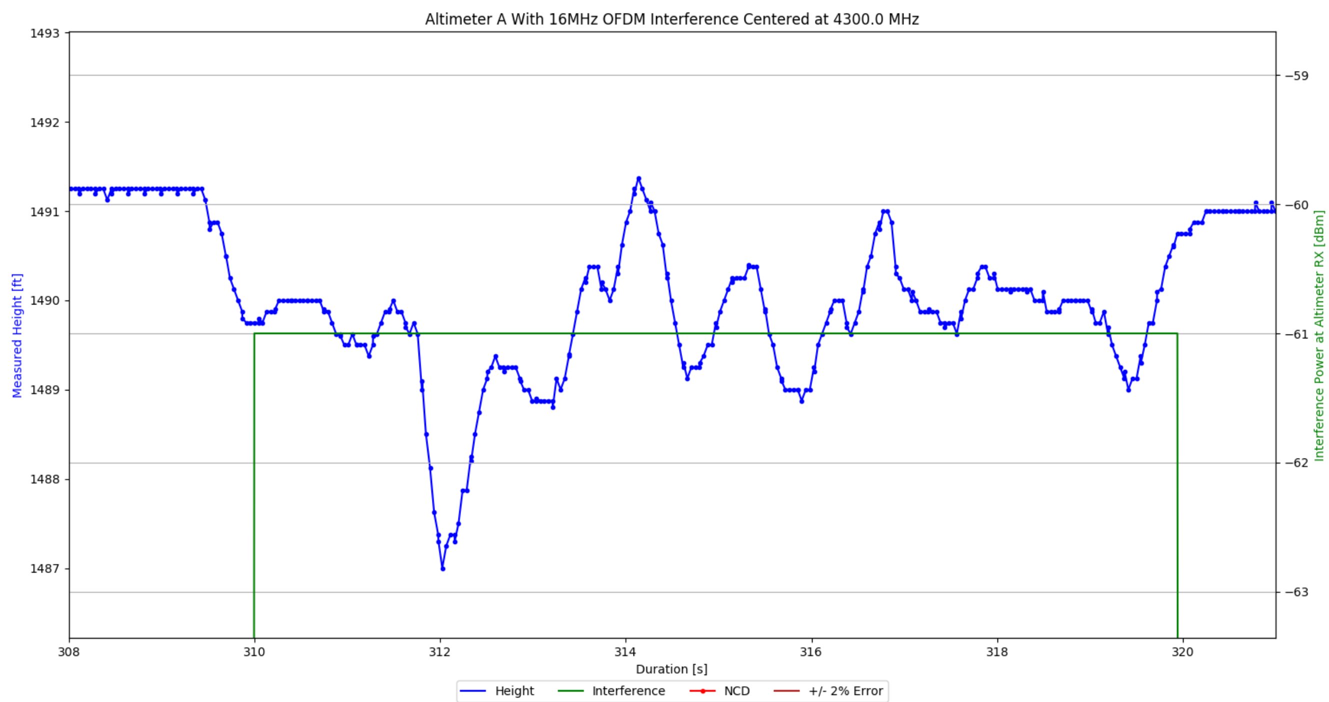

Figure 2.6: Height Plot Showing Sync Error

The final problem encountered involved the synchronization of time stamps of data between the controller laptop and the Ballard ARINC device. The Ballard internal clock drifted from about half a second ahead of the controller laptop clock to about half a second behind. This led to data sets such which looked like Figure 2.6, where distortion from the correct height would appear a split second before the interference signal was turned on. Additionally, due to averaging techniques used in the altimeter's signal processing, there would be several data points during which the altitude would recover from the distortion caused by any interference.

While this issue could not be fixed completely, several different approaches were taken to mitigate it. Firstly, the Ballard used in initial testing was an older unit, which meant it was more prone to clock errors. To address this, the clock was driven by an external IRIG-B timecode signal, rather than the on board IRIG [10]. A program called NMEA time generated the IRIG signal through the controller laptop's audio jack, which was then fed into the Ballard input. These changes would not completely synchronize the Ballard, but it would limit the drift of one clock away from another throughout a test.

Later on, when a new Ballard was purchased for this project, the external IRIG signal was no longer needed, as the updated Ballard driver provided adequate synchronization over USB. Another technique used to prevent any loss of synchronization was to periodically power cycle the Ballard to ensure that the clock is reset. Finally, to minimize the impact of any data points not captured in the RF ON interval by post processing, long interference durations were used to ensure that the impact of any missed points on the true average was minimized.

2.3.2 VSG Class

The VSG Class allows the main loop to create an object corresponding to each VSG connected to the network in the lab setup. This object then has access to functions which perform every operation needed by the main loop. The VSG Class inherits functionality from *SCPI.py* common to every SCPI programmable instrument. The VSG class provides functionality for controlling both the RF generator, as well as the VSG's two baseband generators.

The general purpose of each function in the VSG class is to construct the string for each SCPI command corresponding to that function, and send the command using the inherited SCPI *send* function. For example, the following SCPI command sets Baseband A to MSK modulation:

"SOUR1:BB:DM:FORM MSK"

"SOUR1" corresponds to baseband generator A, where "SOUR2" would correspond to baseband generator B. Different SCPI commands have different parameters like this which the VSG class takes in as a normal Python variable and inserts into the string.

The VSG class provides several functions which perform this task for the RF Generator. RF ON and RF OFF commands are self explanatory, while the *set_carrier* function lets the user choose a carrier frequency as well as the RF power setting.

Similarly, the VSG class offers functions which will set the custom modulation format of either baseband generator (for setting MSK), or for setting the arbitrary waveform generator in each baseband generator (for setting OFDM).This class also provides functions for enabling and disabling the baseband generators, which is necessary for switching between dual and single modulation formats.

Lastly, the VSG class will have functions for setting a dual MSK or dual OFDM waveform. These will make two calls to the single MSK or OFDM function within VSG, simply passing a different source parameter to specify a different baseband generator.

2.3.3 SCPI Class

SCPI.py contains the 'lowest' level of programming in the SCPI hierarchy shown in Figure 2.5. The goal of this abstraction was to implement a small set of core functionality that would be needed for any instrument capable of running SCPI commands. The SCPI class handles the TCP connection, sends and receives all commands and queried data from the instrument, and contains a few other universal instrument commands.

The first task handled by SCPI is the initialization of the socket connection between the controller laptop and the instrument. The constructor takes in the IP address of the instrument and the port number, and uses the Python socket class to initialize the connection. When the experiment is over, the SCPI *close* function which handles the proper termination of the socket connection.

Next, the SCPI class creates custom send and receive commands inherited by every instrument object. The send command handles the proper encoding of any SCPI command string before calling the native socket class *send* function. The receive command decodes a received string from the instrument, and parses the string into the different array elements delimited by commas in SCPI syntax. The receive command is meant to be called within any subclass command which *queries* the instrument for information, as the data types received will vary depending on the query.

Finally, there are a few other commands universal to almost every SCPI controlled device. The *reset* command ensures the instrument begins each experiment from a clean state. The *operation complete* query is called between commands to ensure that no commands are executed out of order. Last, the *identify* query returns the instrument make and model information, which is useful when first setting up an instrument for automation.

2.3.4 Post - Processing

After the test sequence is completed, a series of other Python scripts are used to post process the newly generated data. Time stamped altitudes must be exported from the Ballard CoPilot software into an Excel document, and a python script has to read the data from this log into the SQLite database. A second program uses the time stamps from both the logged altitude table and the generated signals table to create a full data set, where each altitude measurement is labeled with the type and magnitude of interference the altimeter was subjected to at that time. Finally, several Python scripts take this full dataset and use it to generate plots which aid in the interpreting of the results.

2.3.4.1 Parsing The Copilot Log

The first post processing script parses the Ballard Copilot log into the SQLite database so that it can be used by other scripts. The CoPilot export is in an excel file formatted like Table 2.4. This table is read by python using SQLite's built in *read_excel* function. Most of the data is then added to the SQLite database as is, but the *Value* string is stripped of units and cast as a float so that the numerical value can be easily used by other scripts.

The most important columns are the *Value*, *Time*, and *Activity* label. The Activity label reads 'SSM ERROR' whenever the altimeter does not receive a strong enough signal to make a reliable

Item #	Ch#	Lbl#	Value	Time	Activity	Name
0	0	165	509 feet	26:45.4	LO	Radio Height
1	0	165	509 feet	26:45.5	LO SSM ERROR	Radio Height

Table 2.4: Example Copilot Export

ID	Timestamp	Ch#	Carrier	Altimeter	Height	Modulation	Power	Status	RF
0	13:54:40.5	0	4300	Alt A	506	80 MHz OFDM	7	LO	ON
1	13:54:40.6	0	4300	Alt A	506	80 MHz OFDM	9	LO	OFF

Table 2.5: Example Full Dataset Table

altitude calculation. Points with this flag are called NCD or Non-Computed Data.

2.3.4.2 Mapping Interference Signals to Copilot Data

The next post processing script maps each of the Altitude data points from the copilot log (Table 2.4), and maps them to the interference signal (Table 2.2) active at that time. The method for mapping them is simple: if the time stamp for the data point is greater than an interference signal's *start time* and less than the same interference signal's *end point*, the altitude data is labeled with the corresponding modulation format, RF Power, RF State, and Altimeter under test information from the interference signals table. This process is repeated until all data points have been mapped.

2.3.4.3 Plotting the Data

This section will cover the plotting scripts in depth, or it will mention them briefly while they are covered in depth in the results section.

2.4 Setup for Testing Wider Bandwidth Interference

This section covers how the Modified setup from Figure 2.2 is again changed to allow for Wider Bandwidth OFDM signals, with the primary goal being to fill the entire 200 MHz altimeter band with simulated WAIC Interference. A secondary goal of this process is to test the effects of proposed Cellular networks in adjacent bands to the altimeter band while still maintaining simulated

WAIC Interference.

2.5 Setup for WAIC plus Altimeter Interference

This section covers another set of modifications to the setup from Figure 2.2, which are designed to simulate the interference from other altimeters. The test scenario of an aircraft approaching an airport (and thus other altimeters) was determined to be the most critical. A set of Voltage Controlled Oscillators (VCO's) were controlled by function generators with a waveform similar to that seen in Figure 1.1.

REFERENCES

[1] D. Redman, "WAIC Overview and Application Examples," 2011.

[2] H. Canaday, "War on wiring," *Aerospace America*, May 2017.

[3] Ferrell, "Feasibility of Intra-Aircraft Wireless Sensors," Final Report AFE 56, Aerospace Vehicle Systems Institute (AVSI), May 2007.

[4] A. Tewfik, M. Heimdahl, N. Hopper, and K. Yongdae, "University of Minnesota Final Report," AFE 56s1 Final Report, University of Minnesota, Minneapolis, Minnesota, Mar. 2009.

[5] "Consideration of the Aeronautical Mobile (route), Aeronotical Mobile, and Aeronautical Radionavigation Services Allocations to Accommodate Wireless Avionics Intra-Communication," ITU Reccomendation ITU-R M.2318-0, International Telecommunication Union, Geneva, Switzerland, Nov. 2014.

[6] "Final Acts of the World Radiocommunication Conference," Final Acts WRC-15, International Telecommunication Union, Geneva, Switzerland, 2015.

[7] "Operational And Technical Characteristics and Protection Criteria of Radio Altimeters Utilizing the Band 4 200 - 4 400 MHz," ITU Recommendation ITU-R M.2059-0, International Telecommunication Union, Geneva, Switzerland, Feb. 2014.

[8] "Minimum Performance Standards Airborne Low-Range Radar Altimeters," Industry Standard DO-155, Radio Technical Commission for Aeronautics, Washington, D.C., Nov. 1974.

[9] J. G. Proakis and M. Salehi, *Communication Systems Engineering*. Upper Saddle River, New Jersey: Prentice Hall, second ed., 2002.

[10] "Overview of IRIG-B Time Code Standard," Technical Note IRIG Standard 200-04, Cyber Sciences, Murfreesboro, TN, Sept. 2017.