
# NAVAL POSTGRADUATE SCHOOL

**MONTEREY, CALIFORNIA**

# THESIS

**ELECTROMAGNETIC SENSING WITH LOW-COST SOFTWARE DEFINED RADIO**

by

Kenneth H. Liles

June 2021

Thesis Advisor: Alex Bordetsky
Second Reader: Simona L. Tick

**Approved for public release. Distribution is unlimited.**

THIS PAGE INTENTIONALLY LEFT BLANK

| REPORT DOCUMENTATION PAGE | | | *Form Approved OMB No. 0704-0188* |
|---|---|---|---|



| **1. AGENCY USE ONLY** *(Leave blank)* | **2. REPORT DATE** June 2021 | **3. REPORT TYPE AND DATES COVERED** Master's thesis |
|---|---|---|

| **4. TITLE AND SUBTITLE** ELECTROMAGNETIC SENSING WITH LOW-COST SOFTWARE DEFINED RADIO | **5. FUNDING NUMBERS** |
|---|---|
| **6. AUTHOR(S)** Kenneth H. Liles | |
| **7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)** Naval Postgraduate School Monterey, CA 93943-5000 | **8. PERFORMING ORGANIZATION REPORT NUMBER** |
| **9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)** N/A | **10. SPONSORING / MONITORING AGENCY REPORT NUMBER** |

**11. SUPPLEMENTARY NOTES** The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

| **12a. DISTRIBUTION / AVAILABILITY STATEMENT** Approved for public release. Distribution is unlimited. | **12b. DISTRIBUTION CODE** A |
|---|---|

**13. ABSTRACT (maximum 200 words)**

Sensing electromagnetic emissions for offensive and defensive purposes is becoming increasingly important, and software defined radios (SDRs) provide a wide range of electromagnetic spectrum (EMS) sensing capability. This thesis examines the applicability and effectiveness of commercial-off-the-shelf (COTS) technology for electromagnetic sensing and analysis by producing an SDR sensor network prototype. A high-level cost-effectiveness model is developed to produce insights for decision makers regarding the employment of this type of technology. Testing and experimentation suggest that SDRs may be employed as accurate EM sensors with continued research and prototype refinement.

| **14. SUBJECT TERMS** software defined radio, SDR, data fusion, EM sensing | **15. NUMBER OF PAGES** 117 |
|---|---|
| | **16. PRICE CODE** |

| **17. SECURITY CLASSIFICATION OF REPORT** | **18. SECURITY CLASSIFICATION OF THIS PAGE** | **19. SECURITY CLASSIFICATION OF ABSTRACT** | **20. LIMITATION OF ABSTRACT** |
|---|---|---|---|
| Unclassified | Unclassified | Unclassified | UU |

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

**Approved for public release. Distribution is unlimited.**

# ELECTROMAGNETIC SENSING WITH LOW-COST SOFTWARE DEFINED RADIO

Kenneth H. Liles
Captain, United States Marine Corps
BA, University of Arizona, 2013

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT**

from the

**NAVAL POSTGRADUATE SCHOOL**
**June 2021**

Approved by: Alex Bordetsky
Advisor

Simona L. Tick
Second Reader

Alex Bordetsky
Chair, Department of Information Sciences

THIS PAGE INTENTIONALLY LEFT BLANK

# ABSTRACT

Sensing electromagnetic emissions for offensive and defensive purposes is becoming increasingly important, and software defined radios (SDRs) provide a wide range of electromagnetic spectrum (EMS) sensing capability. This thesis examines the applicability and effectiveness of commercial-off-the-shelf (COTS) technology for electromagnetic sensing and analysis by producing an SDR sensor network prototype. A high-level cost-effectiveness model is developed to produce insights for decision makers regarding the employment of this type of technology. Testing and experimentation suggest that SDRs may be employed as accurate EM sensors with continued research and prototype refinement.

THIS PAGE INTENTIONALLY LEFT BLANK

# TABLE OF CONTENTS


I. INTRODUCTION....1
A. BACKGROUND AND MOTIVATION FOR RESEARCH....1
B. OBJECTIVES AND APPROACH....2
C. SCOPE....3
D. RELATED WORKS....3
E. ORGANIZATION OF THESIS....4

II. LITERATURE REVIEW, KEY TERMS AND CONCEPTS....5
A. ELECTROMAGNETIC SPECTRUM....5
B. SIGNATURE AND SPECTRUM MANAGEMENT....5
C. SOFTWARE DEFINED RADIO....6
D. DIGITAL SIGNAL PROCESSING....7
E. DATA INTEGRATION....10
F. OPEN SYSTEMS INTERCONNECTION MODEL....11
G. COST BENEFIT ANALYSIS....11

III. DESIGN....13
A. IDENTIFY HARDWARE....13
1. Main Processor....13
2. SDR....15
3. Networking Hardware....17
B. IDENTIFY SOFTWARE....18
1. GNU RADIO....19
2. Rtl-sdr, pyrtlsdr, and rtl_power....20
3. Python Code....20
C. INTEGRATING THE HARDWARE AND SOFTWARE....21
1. Main Computer and SDR....21
2. Internet Connection through 4G Card....22
3. Establishing Remote Connection....22
4. Completing the Sensor Node....23

IV. SYSTEM TESTING, EXPERIMENTATION, AND RESULTS....25
A. COMPONENT TESTING....27
B. SYSTEM TESTING....32
C. EXPERIMENTATION....39
1. Experiment Setup and Sensor Calibration....39
2. Relative Gain versus Distance function....41

3. Conducting the Experiment ..... 42
D. DISCUSSION OF TESTING AND EXPERIMENTATION RESULTS ..... 47

V. COST EFFECTIVENESS ANALYSIS ..... 49
A. INTRODUCTION ..... 49
1. Status Quo (SQ) ..... 49
2. Proposed Course of Action (COA) ..... 50
3. Stakeholders ..... 50
4. Assumptions ..... 50
5. Steps Used for the Cost Effectiveness Analysis ..... 51
B. ALTERNATIVE TO THE COA ..... 52
C. IDENTIFY COMMON UNIT OF EFFECTIVENESS ..... 52
D. IDENTIFY COST ..... 53
E. COMPUTE COST-EFFECTIVENESS RATIO ..... 53
1. Effectiveness ..... 53
2. Cost ..... 54
3. Effectiveness-Cost Ratios ..... 55
F. DISCUSSION AND RECOMMENDATION ..... 56

VI. CONCLUSION AND RECOMMENDATIONS ..... 59
A. SUMMARY ..... 59
B. LIMITATIONS ..... 60
C. RECOMMENDED FURTHER RESEARCH ..... 60

APPENDIX A. RASPBERRY PI 4B SETUP ..... 61

APPENDIX B. PYTHON CODE ..... 65

APPENDIX C. WAVESHARE SIM7600 TECHNICAL DATA AND SETUP ..... 73

APPENDIX D. LINUX SHELL COMMANDS FOR HEADLESS RASPBERRY PI ..... 75

APPENDIX E. SENSOR CALIBRATION DATA ..... 77

APPENDIX F. EFFECTIVENESS VALUE EQUATIONS ..... 87

APPENDIX G. COST-EFFECTIVENESS VALUES OVER 12 MONTHS ..... 89

**LIST OF REFERENCES** ..... **91**

**INITIAL DISTRIBUTION LIST** ..... **95**

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF FIGURES


Figure 1. The electromagnetic spectrum. Adapted from NASA (2010). .....5

Figure 2. Analog-to-digital sampling. Adapted from Smith (1997).....8

Figure 3. Effects of different sample rates. Source: Lichtman (2021). .....9

Figure 4. Raspberry Pi 3B+ (left) and 4B (right). Adapted from Hattersly (2020).....15

Figure 5. Depiction of SDRs. Source: Amazon.com (n.d.-a, n.d.-b, n.d.-c) and Ettus (2012).....16

Figure 6. Waveshare 4G circuit board for Raspberry Pi 4B. Adapted from Waveshare (2020). .....18

Figure 7. GNU Radio flow graph for component testing.....20

Figure 8. Photograph of the sensor node exterior. .....23

Figure 9. Photograph of sensor node internal hardware.....24

Figure 10. 462.550 MHz, Anritsu MS2721B at 3 feet.....27

Figure 11. 462.550 MHz, SDR test capture at 3 feet. .....28

Figure 12. 462.550 MHz, SDR test capture at 50 feet. .....28

Figure 13. 154.570MHz, Anritsu MS2721B at 3 feet.....30

Figure 14. 154.570 MHz, SDR test capture at 3 feet. .....31

Figure 15. 154.570 MHz, SDR test capture at 50 feet. .....31

Figure 16. GMRS baseline scan, Blackman-Harris windowing .....33

Figure 17. MURS baseline scan, Blackman-Harris windowing .....34

Figure 18. GMRS baseline scan, rectangular windowing.....36

Figure 19. MURS baseline scan, rectangular windowing.....37

Figure 20. GMRS test scan, rectangular windowing.....38

Figure 21. MURS test scan, rectangular windowing.....39

Figure 22. Sensor calibration location and training line. .....40
Figure 23. Scatterplot of relative gain versus distance.....42
Figure 24. Experiment results, test 1. .....44
Figure 25. Experiment results, test 2. .....45
Figure 26. Experiment results, test 2. .....46
Figure 27. COA and alternative CE ratios over 12 months.....56

# LIST OF TABLES


Table 1. Comparison between Raspberry Pi 3B+ and 4B. Adapted from Hattersly (2020). ....14

Table 2. Comparison of technical specifications of Rafael Micro R820T and Ettus USRP SDRs. Adapted from Amazon (n.d.-a, n.d.-b, n.d.-c), Ettus Research (2021), Munson (2018), Nooelec (n.d.), and Pandeya and Temple (2016). ....16

Table 3. Sensor Node Costs. ....24

Table 4. GMRS and MURS frequency, power and bandwidth. Adapted from FCC (2017a, 2017b)....25

Table 5. Baseline band survey parameters for rtl_power....32

Table 6. Second baseline and testing scan parameters for rtl_power. ....35

Table 7. Sensor node average relative gain values versus distance. ....41

Table 8. Sensor 1 and Sensor 2 captured signal Relative Gain and Estimated Range ....43

Table 9. Considered costs. Adapted from Amazon (n.d-d.) and DOD (2021). ....53

Table 10. Spectrum coverage per hour. ....54

Table 11. Spectrum coverage values....54

Table 12. Cost per single month and 6 month operations....55

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF ACRONYMS AND ABBREVIATIONS

| | |
|---|---|
| ADC | analog-to-digital converter |
| API | application programming interfaces |
| ARM | advanced reduced instruction set computer machine |
| ASIC | application specified integrated chip |
| Bn | battalion |
| C4ISTAR | Command, Control, Communications, Computers, Intelligence, and Surveillance, Target Acquisition and Reconnaissance |
| CBA | cost benefit analysis |
| CE | Command element |
| CJCS | Chairman of the Joint Chiefs of Staff |
| CLI | command line interface |
| CMC | Commandant of the Marine Corps |
| COA | course of action |
| COTS | commercial-off-the-shelf |
| CPG | Commandants Planning Guidance |
| CPU | computer processing unit |
| DARPA | Defense Advanced Research Projects Agency |
| dB | decibel |
| DMS | Digital Modernization Strategy |
| DOD | Department of Defense |
| DSP | digital signal processing |
| EM | electromagnetic |
| FCC | Federal Communications Commission |
| FFT | fast Fourier transform |
| FM | frequency modulation |
| FRS | Family Radio Service |
| ft | feet |
| GCE | ground combat element |
| GMRS | General Mobile Radio Service |

| | |
|---|---|
| GPS | Global Positioning System |
| GSG | Great Scott Gadgets |
| GSM | Global System for Mobile Communications |
| HAT | hardware attached on top |
| HTTPS | Hypertext Transfer Protocol Secure |
| Hz | hertz |
| IQ | real and imaginary |
| ISO | International Organization for Standardization |
| JEMSO | Joint Electromagnetic Spectrum Operations |
| kHz | kilohertz |
| mAh | milliampere per hour |
| MEU | Marine Expeditionary Unit |
| MHz | megahertz |
| MOS | military occupational specialty |
| MURS | Multi-Use Radio Service |
| OMB | Office of Management and Budget |
| OS | operating system |
| OSI | Open Systems Interconnection |
| Part 95 | Title 47 of the Code of Federal Regulations, Part 95 |
| RadBn | Radio Battalion |
| RAM | random access memory |
| S-2 | Intelligence section |
| SDR | Software Defined Radio |
| SFTP | Secure File Transfer Protocol |
| SIGINT | signals intelligence |
| SIM | subscriber identity module |
| SQ | Status Quo |
| SSH | Secure Socket Shell |
| TLS | Transport Layer Security |
| UHF | ultra high frequency |

| | |
|---|---|
| USB | universal serial bus |
| USRP | Universal Software Radio Peripheral |
| VHF | very high frequency |
| VPN | virtual private network |

THIS PAGE INTENTIONALLY LEFT BLANK

# ACKNOWLEDGMENTS

I would like to express my sincere gratitude to the faculty and staff members of the Naval Postgraduate School, especially Professor Alex Bordetsky, Dr. Simona Tick, and Eugene Bourakov for providing vast amounts of patience, guidance, and inspiration throughout this process. Thank you for the dedication, time and enthusiasm you placed into this thesis.

I owe my sincere thanks to my wife, Felicia, for her kindness, love, and understanding. For your personal sacrifices in support of my career, I am humbled and forever grateful.

I must also thank my father, Mark Liles, for inspiring me to apply to the Naval Postgraduate School and for patiently listening to hours of unsolicited thesis updates.

THIS PAGE INTENTIONALLY LEFT BLANK

# I. INTRODUCTION

## A. BACKGROUND AND MOTIVATION FOR RESEARCH

In the 21st century, the electromagnetic spectrum is full of information, from living rooms to battlefields, but few realize the potential that can be harnessed by listening to information as actively as we transmit across it. Emerging and mature technology have created an environment conducive to innovation and digital exploration. Devices such as radios, satellite terminals, tablets, and cell phones are used extensively by the Department of Defense (DOD) and adversarial groups alike. These devices receive and transmit electromagnetic (EM) radiation in the form of radio waves. By listening to and collecting these EM emissions, one can develop a representation of the electronic footprint of another entity or organization.

Sensing EM emissions for offensive and defensive purposes is becoming increasingly important. The 38th Commandant of the Marine Corps (CMC), General Berger, directly addressed these opportunities in the 2019 *Commandants Planning Guidance* (CPG).

> The Marine Corps confronts an increasingly complex operational environment abroad and a challenging fiscal outlook. The Marine Corps can no longer accept the inefficiencies inherent in antiquated legacy systems that put an unnecessary burden on the warfighters. We do not currently collect the data we need systematically, we lack the processes and technology to make sense of the data we do collect, and we do not leverage the data we have to identify the decision space in manning, training, and equipping the force. Where we have individual leaders and organizations that are trying to adopt the best practices in data science and data analytics, it is often accomplished through the heroic efforts of a few individuals rather than the organized and sustained effort required to transform how we sense, make sense, and act. (Commandant of the Marine Corps [CMC], 2019).

Preserving the ability to operate in a contested information network environment is described as *paramount* for the future of the Marine Corps (Headquarters, United States Marine Corps [HQMC], 2019). The Department of Defense Chief Information Officer (DOD CIO) identified the evolution of Joint Electromagnetic Spectrum Operations

(EMSO), including sensing, planning, and management, as an objective in the 2019 Digital Modernization Strategy (DMS). Information in the form of bits, bytes and waves constantly surround us, we simply need to cast our nets to collect it.

Spectrum analyzers are devices used for sensing or detecting electromagnetic signals (Gocke, 2018). Spectrum analyzers are precise pieces of equipment that provide an accurate glimpse into a small segment of the electromagnetic spectrum during a specific timeframe. The proliferation of Software Defined Radio (SDR) technology has created a unique opportunity to meet increased signal sensing and signature management requirements without significantly increasing workforce training or operator costs. The affordability provided by mass production and increased use across multiple professional domains is an opportunity the DOD should not allow to pass.

## B. OBJECTIVES AND APPROACH

This thesis examined the applicability and effectiveness of commercial-off-the-shelf (COTS) technology for electromagnetic sensing and analysis. It sought to integrate existing technology to examine the following research questions:

1. How can COTS SDR be applied as sensors to accurately depict very high frequency (VHF) and ultra high frequency (UHF) emissions as part of the future networking infrastructure?

2. How can SDR data integration be used to accurately depict VHF/UHF transmissions?

3. How can data integration contribute to distributed sensing?

4. Can COTS SDR be used to locate VHF/UHF emitter sources?

5. Do the benefits outweigh the costs in employing SDR as a networked sensor?

This thesis employed an exploratory design approach for hardware selection, testing, and integration. Upon successful integration of key hardware components, the accuracy and sensitivity of the system was tested against a calibrated spectrum analyzer. The collected data from multiple prototype sensors was transmitted to a central location

for analysis and comparison. Finally, a cost benefit analysis (CBA) was conducted comparing the SDR sensor network results to a calibrated spectrum analyzer.

## C. SCOPE

This thesis integrates existing hardware and open-source software to develop a working prototype of an SDR sensor network for use in detecting narrowband radio transmissions. This prototype is limited to use on Very High Frequency (VHF) and Ultra High Frequency (UHF) spectrum. Although some integration-based software code was written by the author, development of major system source code remains outside of the scope of this thesis.

## D. RELATED WORKS

Larsen (2007) designed and developed a mobile phone locator using a Universal Software Radio Peripheral (USRP). The author demonstrated the ability to use SDR to locate a cellular device by demodulating Global System for Mobile Communications (GSM) signal bursts and comparing the arrival times. Larsen noted a significant limitation to the study was the slow speed of general purpose hardware available at the time of writing. Recommended future research included using an upgraded SDR platform to run the sensor algorithm.

Gocke (2018) sought a means to identify and map cellular signals used to detonate improvised explosive devices. Gocke demonstrated an approach to algorithm design for cellular signal direction finding based on Friis' equation. Suggested further research includes using commercial SDRs as spectrum analyzers, arranged in both single and phased arrays. Gocke's work was conducted in the Naval Postgraduate School CENETIX Lab,

Munson (2018) also experimented with different SDR platforms to recreate cellular signals for discrete communications in the CENETIX Lab. The author successfully received, deconstructed, and reproduced cellular signals with four different SDRs. Munson noted a major limiting factor in the experimentation resided in the host computer ability to process the digital signaling software during signal reconstruction. Employing SDR sensors for pattern-based early warning was included as suggested future research.

## E. ORGANIZATION OF THESIS

The thesis is divided into six chapters. Chapter I includes the introduction, background, thesis information, and motivation for the research. This chapter states the key research questions and outlines the method in which the author approached the problem. Chapter II includes the literature review and key terms and concepts that are used throughout the thesis. It introduces higher level concepts for understanding the problem space and the methodology used to create a useable solution. Chapter III discusses the approach to selecting hardware, software and the how the sensor node was integrated. Chapter IV encompasses the sensor component and full system testing. This chapter includes the results and discussion from each test. Chapter V examines a cost effectiveness analysis of employing SDR technology as a networked sensor in tactical environments Chapter VI includes the thesis conclusion and recommendations for future research.

| | |
|---|---|
| Chapter I | Background and Motivation, Thesis Objectives and Approach, Related Works, and Thesis Organization |
| Chapter II | Literature Review, Key Terms and Concepts |
| Chapter III | Design |
| Chapter IV | System Testing, Experimentation, and Results |
| Chapter V | Cost-Effectiveness Analysis |
| Chapter VI | Conclusion and Recommendations |

# II. LITERATURE REVIEW, KEY TERMS AND CONCEPTS

## A. ELECTROMAGNETIC SPECTRUM

An understanding of the electromagnetic spectrum is fundamental to recognizing the value SDR may provide in tactical or professional application. The electromagnetic spectrum is a representation of different wavelengths of electromagnetic energy. Figure 1 depicts these wavelengths. At the lower end of the spectrum the wavelengths are long, such as radio waves. As one moves to the right of the spectrum the wavelengths become much shorter. There is a small part of the electromagnetic spectrum that many people are familiar with—the visible light range. Within this wavelength range, the human eye is able to detect electromagnetic energy.

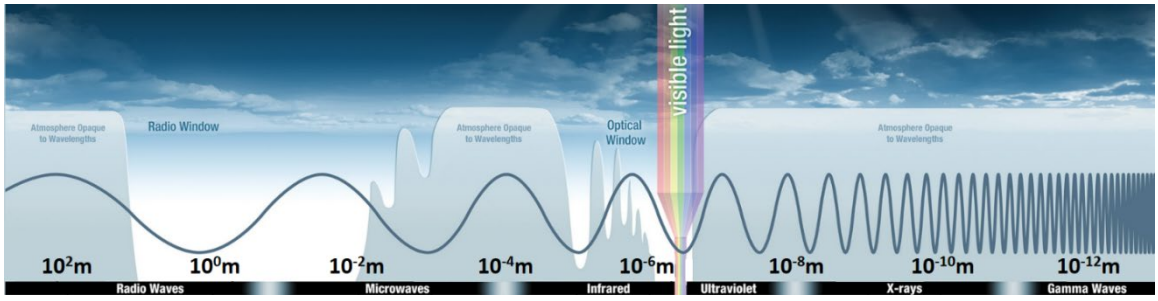


Figure 1. The electromagnetic spectrum. Adapted from NASA (2010).

Wavelength and frequency are related, and their relationship may be expressed as $f\lambda = c$, where $f$ is frequency in Hertz (cycles per second), $\lambda$ is the wavelength in meters, and $c$ is the constant for the speed of light. Because of this relationship, as the wavelength increases or decreases, the frequency will exhibit an inverse behavior. Simply put, higher frequencies have shorter wavelengths while the opposite is true for lower frequencies.

## B. SIGNATURE AND SPECTRUM MANAGEMENT

Reducing EMR emissions to prevent an adversary from locating the source location is called signature reduction. Balancing operational and mission requirements while minimizing electronic signal emissions is called signature management. Any device that

transmits radio waves can be used to profile the user, such as cellular phones, radios, and Bluetooth devices. By simply collecting metadata, such as when specific radio broadcasts were made while specific units were operating within a given area, can provide a useful repository of information for adversarial forces. Even emissions from personally-owned cellular phones can adversely affect signature management for friendly forces (Ferguson, 2020). Decreasing our own signature while increasing our ability to detect adversary signatures has significant potential for the future of warfighting organizations across the range of military operations (Gocke, 2018).

In 2020, the Chairman of the Joint Chiefs of Staff (CJCS) described the electromagnetic spectrum as "maneuver space essential for facilitating control within the operational environment [which] impacts all portions of the operational environment and military operations." Released in 2020, Joint Publication (JP) 3–85 outlines the planning guidance for Joint Electronic Spectrum Operations (JEMSO) and outlines specific actions, such as attacking and protecting the electromagnetic operating environment (CJCS, 2020). Management of the EMS is another key area JP 3-85 specifically addresses. This suggests there is an increasing importance on the ability to sense and identify friendly and adversarial EM emissions at multiple levels within the DOD.

## C. SOFTWARE DEFINED RADIO

SDRs consist of an analog-to-digital converter that digitizes and demodulates a received radio wave using software instead of hardware (Larson, 2007). Using software to digitize and demodulate signals yields much greater coverage of the EMS versus hardware-based radios. This technology paves the way for wider spectrum coverage and automation in EMS sensing and detection.

Authors Chen and Prasad (2009) extensively described the onboard components and functions of SDR systems. The SDR allows the radio to behave much like a computer, where multiple systems generated by programs run on a single piece of hardware to fulfill multiple roles. Three main types of processors are common across different SDR platforms: microprocessors, embedded processors, and a digital signal processor (DSP). Microprocessors and embedded processors control the functions of the SDR, such as

applications and networking, while the DSP provides a means to alter the base frequency of the SDR. Onboard processing power provides SDR capability to sample a high rate of analog signals and convert them to a digital signal, and vice versa. The authors also discussed spectrum sensing in the context of cognitive radio application. Cognitive radio theory involves multiband or SDR radios with the capability to sense levels of traffic across the EMS and adjust the network to a lower traffic frequency (Chen & Prasad, 2009). Although this thesis does not aim at developing an SDR cognitive radio network, SDR sensing platforms will likely contribute to advancing cognitive radio sensing.

Grayver (2013) comprehensively addresses the practical application of SDR. The author specifically included a chapter covering the disadvantages in the application of SDR. They argued that cost, power, complexity and scope should be considered when utilizing or planning to use SDR as a transmission medium. SDR implementation in high-volume, low-margin consumer devices, such as garage door openers, would likely result in an increase in cost in the end device versus utilizing a single function application specified integrated chip (ASIC) (Grayver, 2013). The increase in digital processing required by the SDR also results in a drawback. The author also argued that an increase in processing power will likely result in an increase in energy consumed. SDRs require more power than analog radio processing boards. Grayver (2013) also presents complexity of SDR operation as a disadvantage of implementing SDR technologies. While this is true, automation through computer programming may provide a solution to streamline SDR use. The author's last consideration is that SDR addresses the physical layer of the OSI model. Specifically, in a cognitive radio network, the SDR alone will not improve throughput, and requires additional cross-layer adaptations to fully realize the benefit of an SDR-based network (Grayver, 2013). These factors should be considered during the design and development phase of this thesis.

## D. DIGITAL SIGNAL PROCESSING

Digital signal processing (DSP) is the process of taking an analog signal and converting it into a digital format. In the case of this thesis, radio waves will be the signals processed. To digitize an analog signal, a computer must capture the signal and record

information regarding the signal at various points across the wave (Smith, 1997). In other words, the computer provides a glimpse of the wave's value at different times. This generates discrete variables from a continuously variable wave. As described by Smith (1997), the computer accomplishes this feat by sampling voltage with a sample-and-hold (S/H) transducer that feeds an analog-to-digital converter (ADC). The ADC assigns numerical values to the samples, a process called quantization (Smyth, 2019). Figure 2 depicts the flow of the analog signal through the ADC.

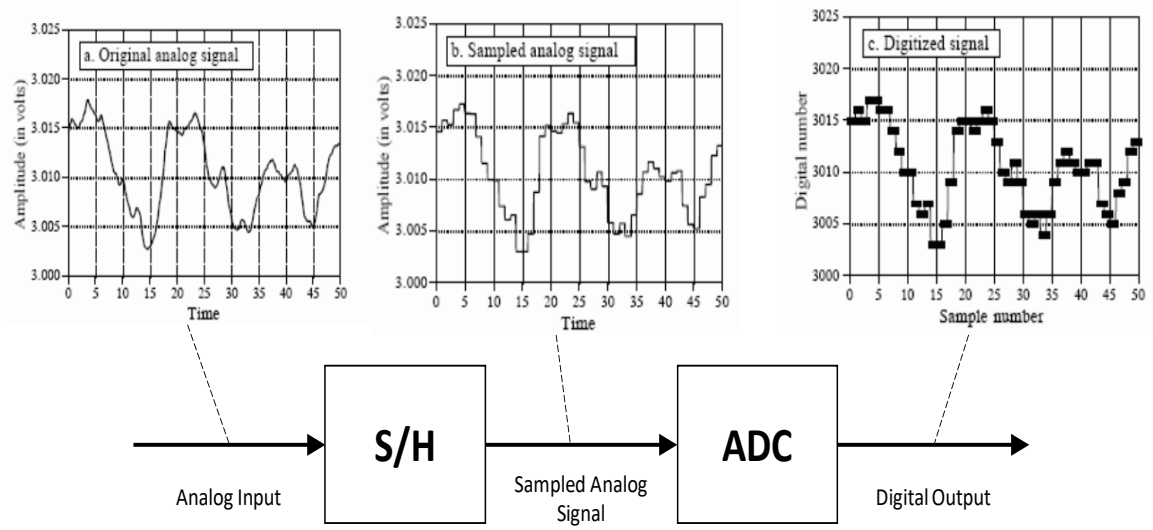


Figure 2. Analog-to-digital sampling. Adapted from Smith (1997).

Recall that frequency (*f*) represents wave cycles per second measured in Hertz (Hz). Because of this, the *sample rate*—or the speed at which the computer creates discrete variables from the continuous wave—will change depending on the frequency to be sampled. The sample rate describes the samples collected in a second and are also measured in Hz. According to the Nyquist Sampling Theorem, in order to accurately capture and reconstruct a signal the sample rate must twice as fast as the signal's highest frequency, $2f_{max}$ (Smyth, 2019). Figure 3 demonstrates the effects of sampling at less than $2f_{max}$ using a standard sinusoid wave. Note that the correct sample rate is required to accurately capture, reconstruct or analyze signals with DSP.

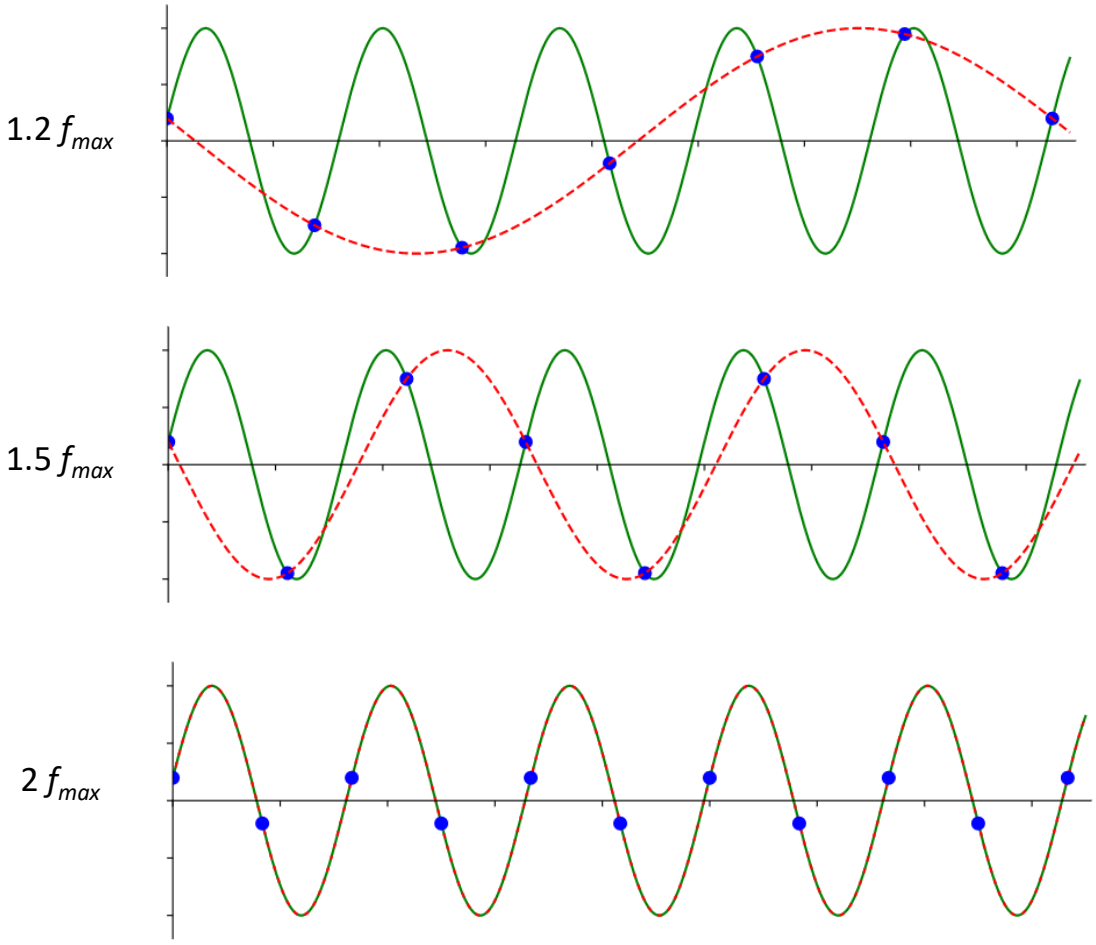


Figure 3. Effects of different sample rates. Source: Lichtman (2021).

The series of samples pictured above represent the time domain of a signal (Lichtman, 2021). The time domain represents the change in voltage of a signal (National Instruments [NI], 2016). A single signal may be comprised of one or more sine waves. According to Fourier's theorem, all waves in the time domain can be deconstructed and represented in the frequency domain as a sum of sine waves (NI, 2016; Lichtman, 2021). The frequency domain shows voltages across different frequencies within a given signal (NI, 2016). In order to go from samples measured in real and imaginary values (IQ) to voltages at specific frequencies, a computer must perform a fast Fourier transform (FFT).

Within the frequency domain, there are different windowing functions to depict the results of the FFT (NI, 2016). This is mainly due to FFT transform accuracy suffering while

depicting non-integer periods of a given signal, which can cause misrepresentation of the original signal). Different windowing functions provide different insights into the frequency domain and depend on the intended application of the analysis. For example, National Instruments (2106) suggests if one examines a frequency with other strong signals near the frequency of interest, a windowing function that offers a narrow view of the frequency of interest should be used. The Blackman-Harris and Kaiser-Bessel windowing options work well for this application. For spectrums where the noise floor is relatively constant or if the frequency range is broad, rectangular windowing (also called uniform window) is a good fit. All three of these windowing options will be used throughout this thesis in different applications.

## E. DATA INTEGRATION

Sensor data fusion technology has been employed for about six decades. Physicist Günther van Keuk worked extensively on phased-array radar data fusion for multiple German government projects since 1965 (Koch, 2013). Air traffic control and early warning and detection systems use data fusion to confirm movement of objects, both in areas with good coverage and areas where one sensor may begin to drop the signal, with a significance placed on the latter. Sensor data fusion produces a reconstruction of an underlying situation by implementing algorithms to exploit imperfect data and combine information sources (Koch, 2013). This technique is broadly employed in the Command, Control, Communications,

Computers, Intelligence, and Surveillance, Target Acquisition and Reconnaissance ($C^4$ISTAR) system based on terrestrial, airborne, and maritime sensors for producing radar picture (Koch, 2013). While radar focuses on combining sensor inputs to track objects, another application of fused data is anomaly detection (Koch, 2013). A baseline is required to detect anomalies, either through algorithm input or through automated algorithm refinement via iteration (i.e., machine learning). Bayesian algorithms are used extensively in radar data fusion, and are likely useful in determining probability density functions for SDR sensor anomaly detection.

In 2016, The Defense Advanced Research Projects Agency (DARPA) developed and tested a nuclear, biological and chemical threat detection system called SIGMA (DARPA, 2016). The system uses a technique where many devices contribute their collected data segments to build a complete picture when the data is combined. This process is also known as *sensor data fusion* (Koch, 2013). This process can gather insight on phenomena that is difficult or impossible to obtain using a single sensor, or lies beyond the technical limitations, reliability, and cost of a single sensor (Koch, 2013).

## F. OPEN SYSTEMS INTERCONNECTION MODEL

Computer networking is important to understand in order to use radio networks to pass data between devices. A computer network is a group of devices connected together to accomplish a specific task (Alani, 2014). These connections form the basis of moving information between the devices. The network must be organized, and governed by a set of rules commonly understood between the devices called protocols (Alani, 2014). In 1977, the International Organization for Standardization (ISO) established a committee to develop an overarching framework for networking standards (Alani, 2014). The result was a model that provides universal terminology and context for protocols, named the Open Systems Interconnection (OSI) reference model. The OSI model groups protocols into seven categories called layers. SDR and other transmission mediums directly influence the physical layer, but host processors on the SDR platform may be used to implement changes across the OSI model.

## G. COST BENEFIT ANALYSIS

Developing and implementing new technologies across an organization must offer potential for return on investment. Cost-benefit analysis (CBA) is one method recognized by the U.S. government to estimate the potential returns on investments. OMB Circular A-94 serves as the guideline for conducting cost-benefit analysis for federal programs and projects (White House, 1992). Identifying costs and benefits can be accomplished through monetizing the value of the benefits and the costs and weighing the outcome. This process is called net present value, and is a standard method for estimating the economic value of a federal program. Although providing both costs and benefits in monetary value is

preferred, quantifying the costs and benefits in any form may provide insights for program decision makers.

Cellini and Kee (2105) thoroughly describe one model for conducting CBA. According to the authors, a CBA is most useful when evaluating a single program, such as the case with this thesis. Special considerations must be used when identifying the stakeholders for cost-benefit analysis. These should be discussed directly when introducing the CBA model. Another key area in conducting the CBA is identifying direct and indirect costs and benefits as well as cost and benefit transfers. Cost and benefit transfers exist where a program shifts the perceived cost or benefit throughout the organization or society as a whole, instead of realizing an actual difference. An example may include manpower savings in a deployed theater due to technology proliferation, but an increase in manpower to maintain the systems and administer services elsewhere in the organization.

# III. DESIGN

This chapter describes the various factors considered for selecting the hardware, software and peripheral devices for integration into the system. It is more technical in nature and describes the challenges and solutions during the conduct of the research and writing of this thesis. The primary question this thesis seeks to answer is how to employ low-cost COTS SDR as sensors to accurately depict VHF/UHF signals as part of a sensor network. This was the guiding idea in the selection of the hardware for the system design. The system will be comprised of a few major components, namely a small, power efficient computer to run the SDR, a low-power, low-cost SDR, a power source, and a transmitting device capable of transmitting the collected information to a hub computer.

## A. IDENTIFY HARDWARE

### 1. Main Processor

The requirements for the main processing computer were centered around the maximizing processing power while operating at the minimal amount of power. This would allow the sensor to operate using commercially available rechargeable battery banks. Within a distributed sensor network, it is important for the nodes to be set up in an expeditionary manner and self-sufficient. Thus, if possible, the device should be able to provide its own power in order to eliminate the need for an external power source. Additional requirements are outlined below:

- Ability to integrate with and control low-cost SDRs
- Ability to process large amounts of collected data
- Ability to integrate with transmitting hardware/network
- Low power requirement

Previous research used laptops, cellular phones and tablets to utilize universal serial bus (USB) SDR devices (Gocke, 2018). The Raspberry Pi series of computers have been used in previous research and met varying degrees of success within similar application (Gocke, 2018). Previous research also used Raspberry Pi 3B+ boards—significant

improvements have been made between that model board and the Raspberry Pi 4B. Table 1 outlines the differences between the two boards.

Table 1. Comparison between Raspberry Pi 3B+ and 4B. Adapted from Hattersly (2020).

| | **Raspberry Pi 3B+** | **Raspberry Pi 4B** |
|---|---|---|
| **Processor** | Broadcom BCM2837B0, Quad-core Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz | Broadcom BCM2711, Quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz |
| **RAM** | 1GB LPDDR2 SDRAM | 1GB, 2GB, or 4GB LPDDR4-3200 SDRAM (depending on model) |
| **Connectivity Capabilities** | 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE | 2.4GHz and 5.0GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE |
| **Ethernet Capabilities** | Gigabit Ethernet over USB 2.0 (maximum throughput 300Mbps) | Gigabit Ethernet |
| **USB Capabilities** | 4 × USB 2.0 ports | 2 × USB 3.0 ports; 2 × USB 2.0 ports |
| **GPIO headers** | Raspberry Pi standard 40-pin GPIO header | Raspberry Pi standard 40-pin GPIO header |
| **Power Requirements** | 5V/2.5A DC | 5V/2.5A DC |

Aside from the obvious upgrades in CPU and RAM between the two models, the Raspberry Pi 4B included two USB 3.0 ports that were very important for the transmitting device selection, as the connection for data throughput used USB vs GPIO headers. This will be discussed later in part C of this section. Figure 4 depicts the Raspberry Pi 3B+ and 4B boards, respectively. At the time of writing, the Raspberry Pi 3B+ could be purchased for US$49.00 and the 4B could be purchased for US$53.90.

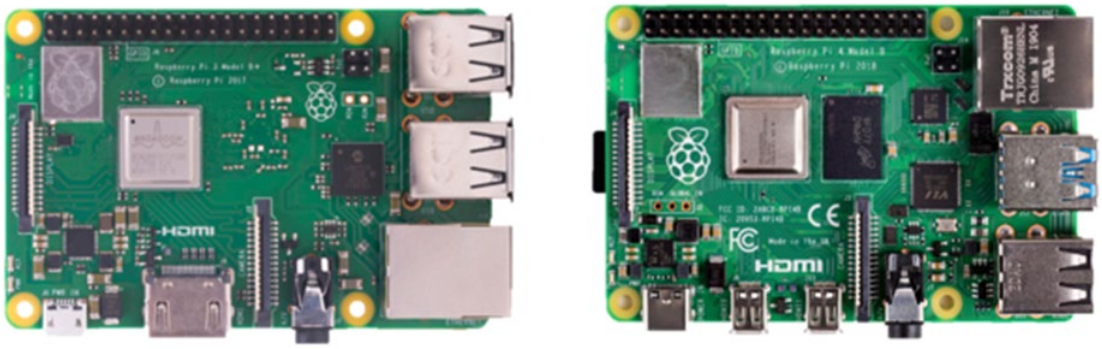

Figure 4. Raspberry Pi 3B+ (left) and 4B (right). Adapted from Hattersly (2020)

Both options were viable solutions, however previous research suggested an increase of processing power would increase the sensing ability of the SDR (Munson, 2018). With this in mind, the Raspberry Pi 4B was selected as a cost-effective and available COTS solution for this application due to its affordability, processing power and power requirements.

### 2. SDR

Four different SDRs were evaluated through literature review: the Nooelec NESDR Smart v4, RTL-SDR BLOG v3, the Great Scott Gadgets (GSG) HackRF One and the Ettus Universal Software Radio Peripheral (USRP) B205. Both the NESDR Smart v4 and the RTL-SDR were developed from the Rafael Micro R820T Digital TV Tuner. The Ettus USRP B205 uses National Instruments proprietary SDR. The HackRF One is produced by GSG in Colorado, but additional information regarding their circuit board is not available. Table 2 provides the technical specifications of both SDRs. Figure 5 depicts the types of SDRs in this chapter.

Table 2. Comparison of technical specifications of Rafael Micro R820T and Ettus USRP SDRs. Adapted from Amazon (n.d.-a, n.d.-b, n.d.-c), Ettus Research (2021), Munson (2018), Nooelec (n.d.), and Pandeya and Temple (2016).

| SDR | Cost | Frequency Range | Max Sample Rate |
|---|---|---|---|
| **NESDR Smart v4** | US$33.95 | 25 MHz to 1.7 GHz | 2.56 MS/s |
| **RTL-SDR v3** | US$27.95 | 0.5 MHz to 25 MHz | 3.2 MS/s* |
| **HackRF One** | US$319.95 | 1 MHz to 6 GHz | 20 MS/s |
| **USRP B205** | US$1,101.00 | 70 MHz to 6 GHz | 61.44 MS/s |


Figure 5. Depiction of SDRs. Source: Amazon.com (n.d.-a, n.d.-b, n.d.-c) and Ettus (2012).

From a purely technical perspective, the Ettus B205 and the HackRF one are much more capable than the mass-produced R820T SDRs. Both the Ettus and HackRF are able to send in addition to receive that contribute to their large price difference. They also include the ability to provide a reference signal input generated form within the circuit board, which is very different from the economy SDR boards.

However, the R820T based SDRs have a large repository of open-source software and supporting code libraries in high-level programming languages. In particular, the R820T is supported by many Python application programming interfaces (APIs). Python is used extensively in Raspberry Pi applications that played an important role in the selection of the SDR for integration in the system. The Ettus B205 is supported by APIs in the C+ programming language. Ettus did release Python APIs for the B205, however they were undocumented at the time of writing. In order to address the primary research question and remain within the scope of the thesis, the Nooelec NESDR Smart v4 was selected as the sensing component for the system prototype.

### 3. Networking Hardware

The networking requirements for the system were also focused on a low cost and commercially available capability to transmit large amounts of collected samples. Multiple approaches were considered to solve the problem. Using an SDR capable of receiving and transmitting would provide an avenue to transmit data. However, this would require an SDR capable of receiving information for sensing as well as receiving information for networking, or two separate SDRs with their own specific functions. Of the SDRs identified in previous section, the RTL-SDR Blog v3 and NESDR Smart v4 are receive only. They are not capable of transmitting any radio signals. The Hackaday HackRF One and the Ettus B205 are capable of both sending and receiving radio signals. Adding one of these SDRs to the system would result in a significant cost increase as well as additional computing and power requirements.

Further investigation into solutions revealed another commercial solution. The Raspberry Pi was developed in a manner that allows quick additions of peripheral equipment, including hardware. Modules that connect directly to the Raspberry Pi main

board are called hardware attached on top (HAT). Waveshare developed a HAT called the SIM7600, a fourth generation (4G) mobile networking card that integrates directly with the Raspberry Pi 4B (Waveshare, 2020). It utilizes a subscriber identity module (SIM) card from a commercial network carrier to provide access to 4G cellular networks. Once connected to the internet, a virtual private network (VPN) tunnel could be created between the sensor device and hub computer. Figure 6 displays the Waveshare 4G board.

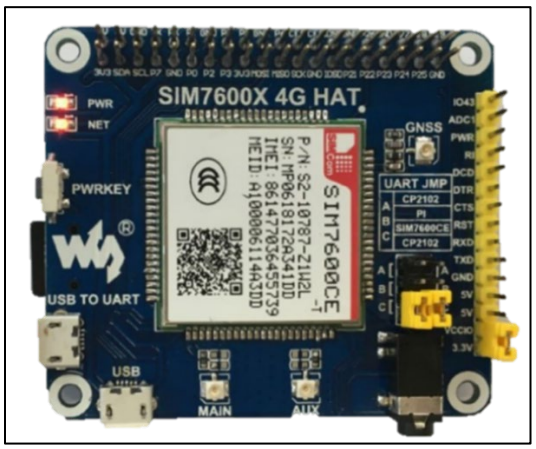


Figure 6. Waveshare 4G circuit board for Raspberry Pi 4B. Adapted from Waveshare (2020).

In addition to providing the capability to access cellular netwroks, the SIM7600 card provides Global Positioning System (GPS) and Glonass (GNSS) positioning funcitonality (Waveshare, 2020). The ability to sense EM emissions and add positioning data to the collected samples is important to the useability of a sensor node. Additional features and technical data for this device may be found in appendix C.

## B. IDENTIFY SOFTWARE

Integrating the devices relied mainly on open source software. The Raspberry Pi computer did not arrive with an operating system (OS). This allows the user to select an appropriate OS for project-specific applications. According to the manufacturer, the computer can use any advanced reduced instruction set computer machine (ARM) Linux distributions, but the recommended OS distribution is the Raspberry Pi OS available from

their website (Raspberry Pi Foundation, n.d.). The recommended OS was used for the computer, and the Python programming language was selected as the primary scripting language for the project due to its open source support and interoperability between multiple operating systems. Open source code repositories were available for R820T-based SDR receivers in both Linux and Python. A complete list of software and their repositories at the time of writing are available in appendix B.

### 1. GNU RADIO

One option for conducting DSP with a Linux OS is the GNU Radio application. This software is available open source and is a popular option for amateur radio operators and enthusiasts using SDRs to demodulate and replicate signals (GNU Radio [GNU], 2021). GNU software uses a graphical user interface to generate a Python code to operate the SDR. The software also includes compatibility for the C+ programming language and the Matlab and Octave software (GNU, 2021). It provides a wide array of DSP tools including SDR control integration, FFT manipulation, I/Q balancing and other features. GNU radio uses a flow graph and functions, called *blocks*, to control the DSP. The tools described above are standard blocks available within the GNU radio GUI. The program also provides users the ability to create their own blocks using Python. GNU Radio is used in this thesis to examine individual radio signals and provide a baseline visualization for comparison against a spectrum analyzer. Figure 7 illustrates the flow graph used to conduct the tests in this thesis.

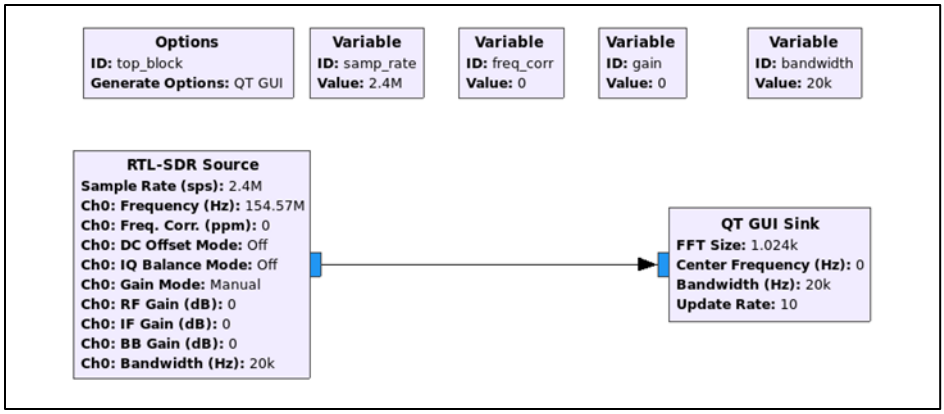


Figure 7. GNU Radio flow graph for component testing.

GNU Radio was considered for use in the automation of the sensors, but building the custom blocks required knowledge of Python programming that was outside the scope of the thesis.

### 2. Rtl-sdr, pyrtlsdr, and rtl_power

Rtl-sdr is open source software developed for SDR DSP and automation on Linux ARM operating systems. It includes a command library for Linux CLI to interface directly with the SDR via USB. The pyrtlsdr package provides Python interoperability with the rtl-sdr drivers within the Linux package. The pyrtlsdr package is used as the primary control library for the custom Python code written for this thesis in appendix B.

Rtl_power is an SDR scanning tool based on the rtl-sdr Linux packages (Keen, 2015). It allows a user to input a range of frequencies and a time limit to conduct wideband surveys. The samples provided by each pass of the SDR are measured and averaged into a bin size input by the user. This keeps the resulting comma separated value (CSV) file from reaching sizes too large to be managed by smaller computers, such as the Raspberry Pi. Users may input additional commands, such as gain and windowing options, for different project applications or for different DSP perspectives.

### 3. Python Code

This thesis included creation of Python code to integrate functions of different Python libraries for a specific purpose. Many of the available libraries and tools performed

DSP at the appropriate level to detect frequency peaks but recording the data in an orderly and manageable manner was difficult or impossible to achieve using available code alone. The program included in appendix B is based on the pyrtlsdr package and previous work by Eugene Bourakov, Research Associate at the CENETIX Lab, Naval Postgraduate School. The program iterates the SDR limit of 2 MHz scan of a frequency range and appends sample data, as well as date, time, and GPS location, to an array. This array is then written to a .csv file for further processing and analysis. The program also includes an alert notification for peak frequencies. The criteria for the peak frequency was selected as a set value based on testing. It represents any signal peak that rises above XX dB from the noise floor and highest DC spike. The alert information does not require the scanning to be complete—it is written to a log immediately and can be accessed while the scan continues. As designed, the log must be retrieved from the node using SFTP, however this information would make a good candidate for a (.JSON) message sent to a server and displayed on a screen using the sensor GPS data included in the message.

## C. INTEGRATING THE HARDWARE AND SOFTWARE

A staged approach was used in order to integrate all of the components into a functioning system. Stage 1 was based on integrating the main computer with the selected SDR and discovering how to control the SDR using command line interface (CLI). Stage 2 included integrating the networking hardware with the system and successfully accessing the internet through a cellular network. Stage 3 focused on establishing a persistent VPN between the sensor node and a hub computer and passing commands and data between the devices using Secure Socket Shell (SSH) and Secure File Transfer Protocol (SFTP). Stage 4 included encasing the sensor and associated equipment into a waterproof container.

### 1. Main Computer and SDR

At this stage, the project consisted of a single Raspberry Pi 4B and a NESDR Smart v4 SDR. The goal of this stage was to simply get the base system working. The SDR and computer serve as the heart of the sensor. Understanding how these two systems worked together set conditions to push forward in the integration project. The computer was provided the recommended Raspberry Pi OS via a micro-SD card that was imaged from an

auxiliary computer. The drivers for the R820T series SDR receivers were identified and installed from an open source repository. At this point, the computer recognized the SDR and was able to pass control commands from the CLI to the SDR, including the ability to conduct frequency band scans and record the results. The technical build instructions, including Linux repository information, may be found in appendix A.

### 2. Internet Connection through 4G Card

While the main computer and SDR were capable of connecting to the internet via 2.4 and 5 GHz wireless internet, this would severely limit the range of the sensor node and the locations available for the testing phase of this thesis. To overcome this limitation, the Waveshare SIM7600X 4G HAT peripheral for the Raspberry Pi was selected. Unlike the base computer and SDR, the documentation for the 4G HAT was not thorough. Multiple versions of the HAT are available, each with an associated geographic region. The A-H model was used for this thesis that corresponds to the North America region (Waveshare, 2020). An AT&T Inc. prepaid mobile data plan and associated SIM card was purchased for the project. Of note, only SIM cards for tablet computers were compatible with the Waveshare SIM7600X 4G HAT. The next challenge was establishing connection between the HAT and the Raspberry Pi. Unlike commercial cellular devices, the Raspberry Pi is not designed to use cellular connections as a default method for passing IP traffic. The network configuration must be changed via the CLI after each reboot in order to point the traffic to the appropriate default gateway. The SIM7600X also provided GPS functionality for the Raspberry Pi. Appendix C includes technical instructions for this stage.

### 3. Establishing Remote Connection

At this stage, the sensor node was able to control the SDR, perform scans with parameters passed through the CLI, and connect to the internet through a cellular network. Different methods exist for creating a connection between two computers. One method was to create a Linux SSH server computer and connect each node as a client to the server. This was determined to be too much of a security vulnerability for home networks while conducting research, and other options were available and considered. In order to minimize security risks, a third-party VPN service from PiTunnel was used. This service uses

Transport Layer Security (TLS) and Hypertext Transfer Protocol Secure (HTTPS) to establish a tunnel between the Raspberry Pi and other computers, providing a means to use SSH and SFTP between the devices. After the VPN was configured, the sensor node successfully received and executed commands, and transferred the collected data back to a remote computer.

### 4. Completing the Sensor Node

This stage primarily served to ruggedize the sensor node in order to proceed to the testing phase. The primary pieces of hardware enclosed in a watertight Pelican 1060 case and a 20,000 milliampere per hour (mAh) rechargeable battery was added to power the device. Figure 8 and Figure 9 depict the completed sensor node with all components labelled.

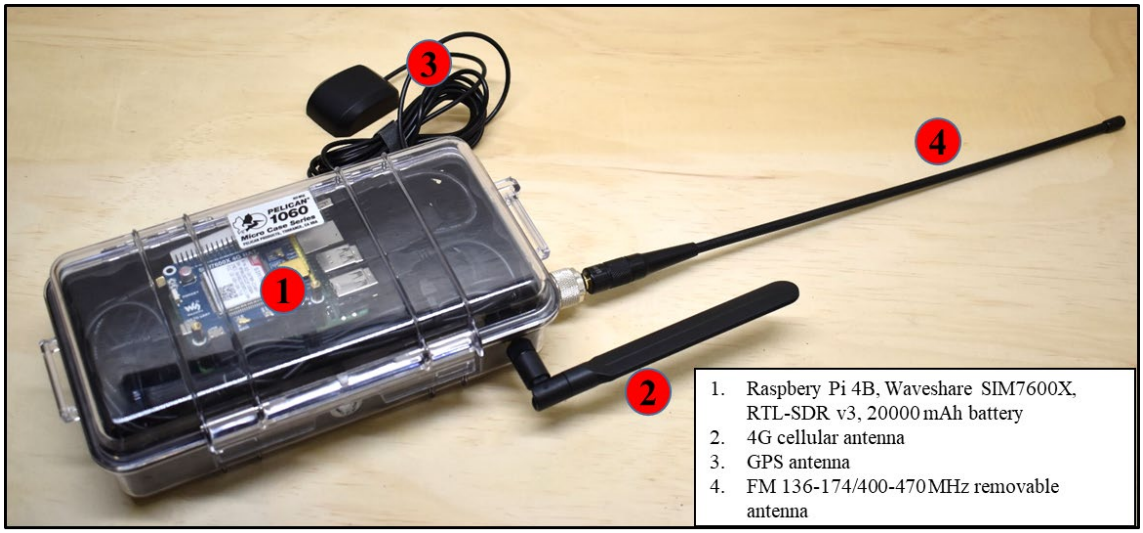


Figure 8. Photograph of the sensor node exterior.

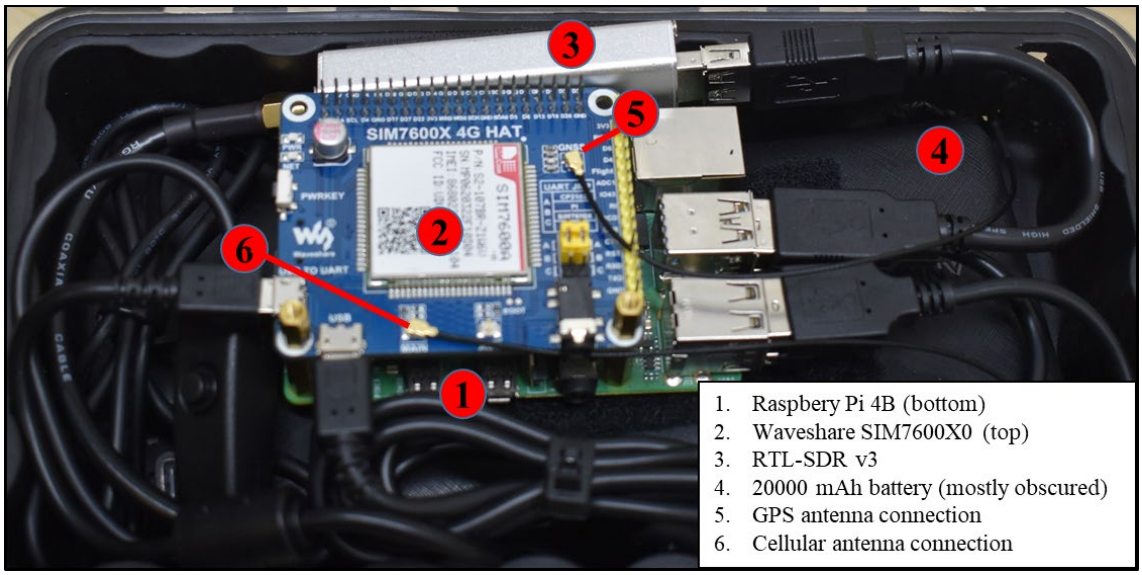


Figure 9. Photograph of sensor node internal hardware.

The pre-tax cost for each component and for a complete sensor node is provided in Table 3. A second sensor unit was built, but did not include the 4G HAT or AT&T service due to fiscal constraints.

Table 3. Sensor Node Costs.

| Component | Cost (US$) |
|---|---|
| Raspberry Pi 4B | 53.90 |
| Waveshare SIM7600X | 72.89 |
| RTL-SDR v3 | 27.95 |
| Anker PowerCore 20000mAh battery | 41.99 |
| Pelican 1060 Case | 23.05 |
| Cables, connectors and antenna | 29.88 |
| **Total** | **250.56** |

# IV. SYSTEM TESTING, EXPERIMENTATION, AND RESULTS

This thesis was designed and conducted during the height of the COVID-19 pandemic. The pandemic impacted the resources available for research and all testing was restricted to private home and field environments. To overcome challenges and provide meaningful testing, a personal General Mobile Radio Service (GMRS) license was procured in order to abide by Federal Communications Commission (FCC) rules and regulations. The license allows users to transmit up to 2 watts of power on the Multi-Use Radio Service (MURS) frequencies from 151–154 MHz, and up to 5 watts in the GMRS frequency range of 462–467 MHz (Federal Communications Commission [FCC], 2017a, 2017b). Alternative testing without a license restricts a user to 2 watts on preset channels within the Family Radio Service (FRS) frequency range only. In addition to the license, only radios certified to operate under Title 47 of the Code of Federal Regulations, Part 95 (Part 95) may be used to transmit radio signals within the GMRS range of frequencies (FCC, 2017a). A Part 95 certified TERA TR505 handheld radio (FCC ID: 2ACK8TR505D) was used for all tests contained within this thesis. The GMRS and MURS information is presented in Table 4.

Table 4. GMRS and MURS frequency, power and bandwidth. Adapted from FCC (2017a, 2017b).

| **GMRS** | | | | | |
|---|---|---|---|---|---|
| **Frequency (MHz)** | **Power (W)** | **Bandwidth (kHz)** | **Frequency (MHz)** | **Power (W)** | **Bandwidth (kHz)** |
| **462.55** | 5 | 20 | **462.725** | 50 | 20 kHz |
| **462.5625** | 5 | 20 | **467.55** | 50 | 20 |
| **462.575** | 5 | 20 | **467.5675** | 50 | 12.5 |
| **462.5875** | 5 | 20 | **467.575** | 50 | 20 |
| **462.6** | 5 | 20 | **467.6125** | 50 | 12.5 |
| **462.6125** | 5 | 20 | **467.6** | 50 | 20 |
| **462.625** | 5 | 20 | **467.6625** | 50 | 12.5 |

| GMRS | | | | | |
|---|---|---|---|---|---|
| **Frequency (MHz)** | **Power (W)** | **Bandwidth (kHz)** | **Frequency (MHz)** | **Power (W)** | **Bandwidth (kHz)** |
| **462.6375** | 0.5 | 20 | **467.625** | 50 | 20 |
| **462.65** | 0.5 | 20 | **467.7125** | 50 | 12.5 |
| **462.6625** | 0.5 | 20 | **467.65** | 50 | 20 |
| **462.675** | 0.5 | 20 | **467.5875** | 50 | 12.5 |
| **462.6875** | 0.5 | 20 | **467.675** | 50 | 20 |
| **462.7** | 0.5 | 20 | **467.6375** | 50 | 12.5 |
| **462.7125** | 0.5 | 20 | **467.7** | 50 | 20 |
| **462.725** | 50 | 20 | **467.6875** | 50 | 12.5 |
| **467.55** | 50 | 20 | **467.725** | 50 | 20 |
| **MURS** | | | | | |
| **Frequency (MHz)** | **Power (W)** | **Bandwidth (kHz)** | **Frequency (MHz)** | **Power (W)** | **Bandwidth (kHz)** |
| **151.82** | 2 | 11.25 | **154.57** | 2 | 20 |
| **151.88** | 2 | 11.25 | **154.60** | 2 | 20 |
| **151.94** | 2 | 11.25 | **N/A** | N/A | N/A |

The testing and experimentation of the system design was approached in phases. The first phase tested the basic components of the sensor node—the Raspberry Pi and the SDR—against the spectrum analyzer. The next phase examined the performance of the entire system in a band survey. Lastly, two sensor nodes were tested for variations between the two complete systems and their data outputs were analyzed for findings. For all testing and experimentation, an Anritsu MS2721B spectrum analyzer provided by the Naval Postgraduate School was used as a control variable. The sensor node design served as the test variable. All devices used the same model of frequency modulation (FM) antennas tuned for 136–174/400-470MHz

## A. COMPONENT TESTING

The first phase of testing consisted of comparing the how the SDR and Raspberry Pi captured radio signals compared to the Anritsu MS2721B. In order to test this, GNU radio onboard the Raspberry Pi was employed to receive and display a transmission while the spectrum analyzer was used to display the same signal at 462.550 MHz. The test was conducted within an office and the transmitting and receiving devices were roughly 3 feet apart. The spectrum analyzer was centered slightly higher than 462.550 MHz (in order to produce a quality image) and the window width was 2 Mhz. The windowing data for the MS2721B was not available within the equipment manual, however similar models such as the MS2090A use Kaiser-Bessel window function (Anritsu, 2019). The results from the MS2721B suggest the windowing is Kaiser-Bessel and the frequency separation appears to be consistent with the same windowing function. The TERA radio was programmed to transmit on 426.550 at 2 watts. The control test results depict a clear peak visible at 462.550MHz. Figure 10 illustrates the screen capture from the spectrum analyzer.

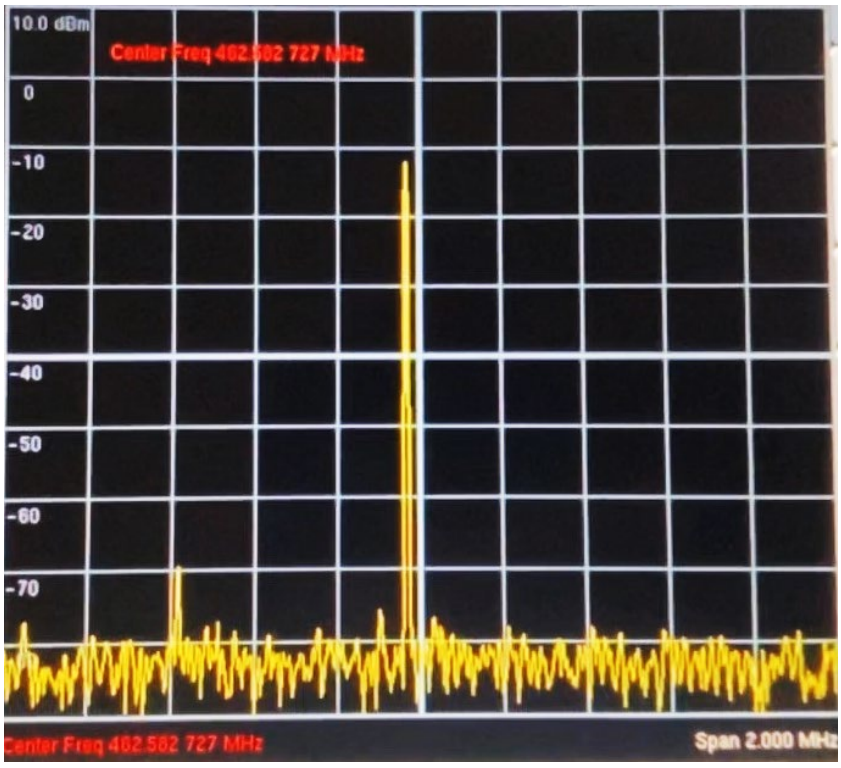


Figure 10. 462.550 MHz, Anritsu MS2721B at 3 feet

The SDR was then tested centered at the test frequency, with gain set at 0, and Kaiser windowing for visualization. Figure 11 depicts the test results. Similar results were produced, however the SDR appeared to display additional, mirrored frequencies both above and below the center frequency. A second test was conducted with the same parameters, except at a distance of 50 feet. Figure 12 shows the resulting capture.

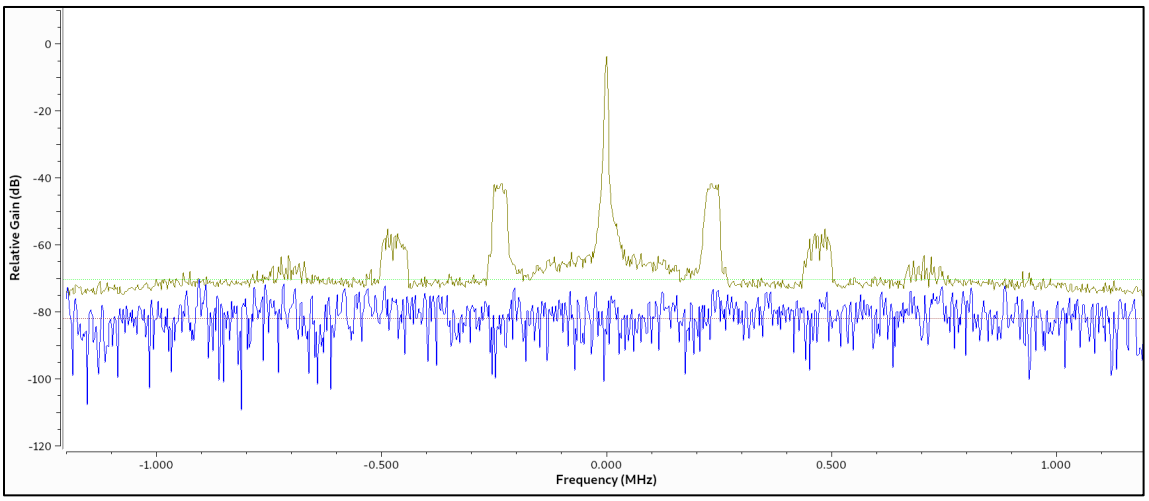


Figure 11. 462.550 MHz, SDR test capture at 3 feet.

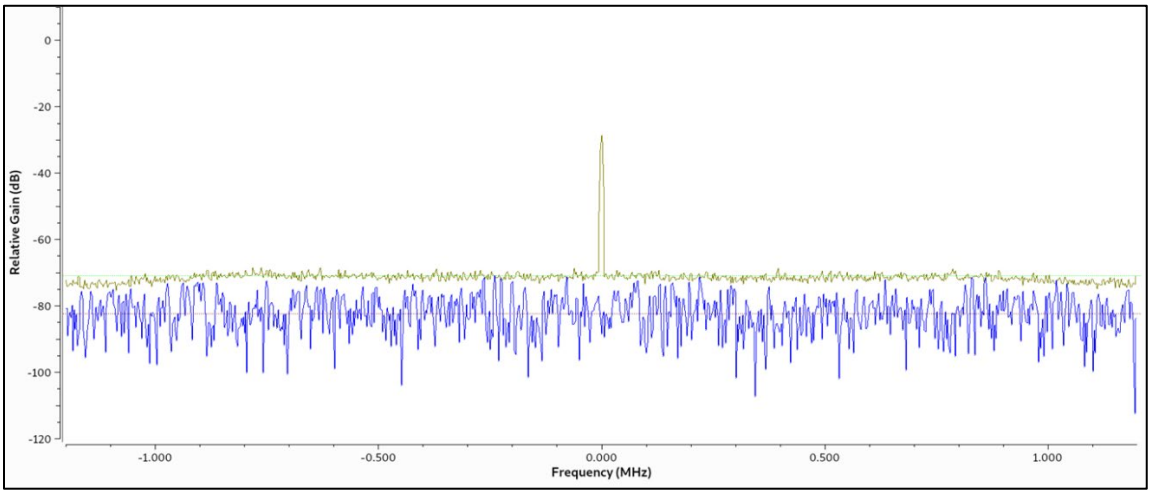


Figure 12. 462.550 MHz, SDR test capture at 50 feet.

The result shows an elimination of the mirrored frequencies within the SDR capture. The difference may be due the lack of an attenuator onboard the SDR and the close proximity between the transmitting and receiving device. According a post from the SDR manufacturer on their forum, this phenomenon is the result of the SDR receiving too much power (RTL-SDR, 2018).

The same series of tests were conducted within the VHF band using MURS frequencies. The thesis examines the feasibility of using COTS for both UHF and VHF sensing, and these tests examine the performance of the Raspberry Pi and SDR in the lower range of the EM spectrum.

In order to conduct a test within the MURS frequency range, an appropriate frequency has to be selected from Table 4. In order to reduce dependent variables, a frequency that provided 20 kHz bandwidth was needed. For this series of tests, 154.570 MHz was selected and the TERA radio was programmed with the test frequency and transmitting power of 2W. The first test used the spectrum analyzer to establish the control capture. The spectrum analyzer was centered at 154.00 MHz in order to provide a clear picture of the signal capture. Figure 13 depicts the result.

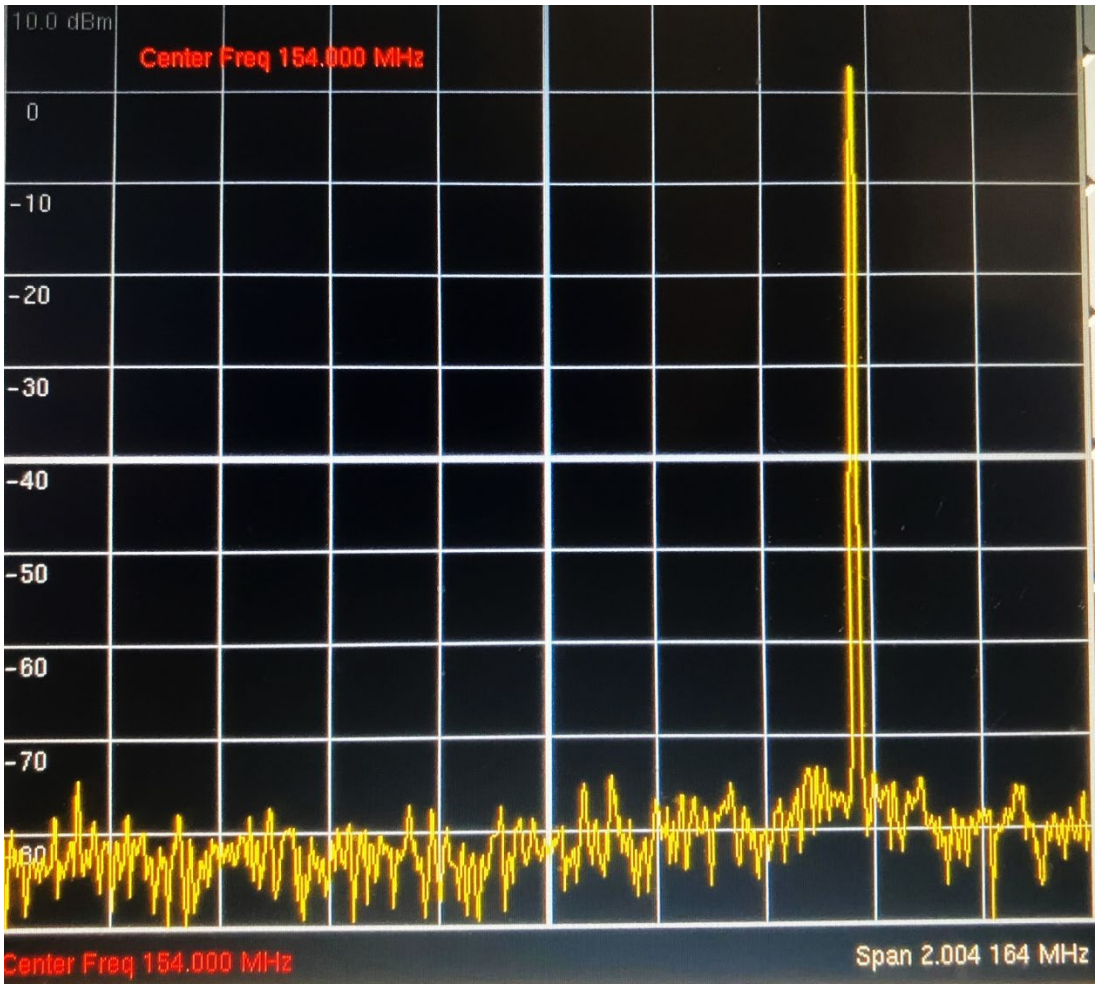


Figure 13. 154.570MHz, Anritsu MS2721B at 3 feet

The second test consisted of using the Raspberry Pi and SDR to capture the same signal. To remain consist with the first set of tests, the SDR was set at 0 gain and Kaiser windowing. Figure 14 and Figure 15 depict the test results.

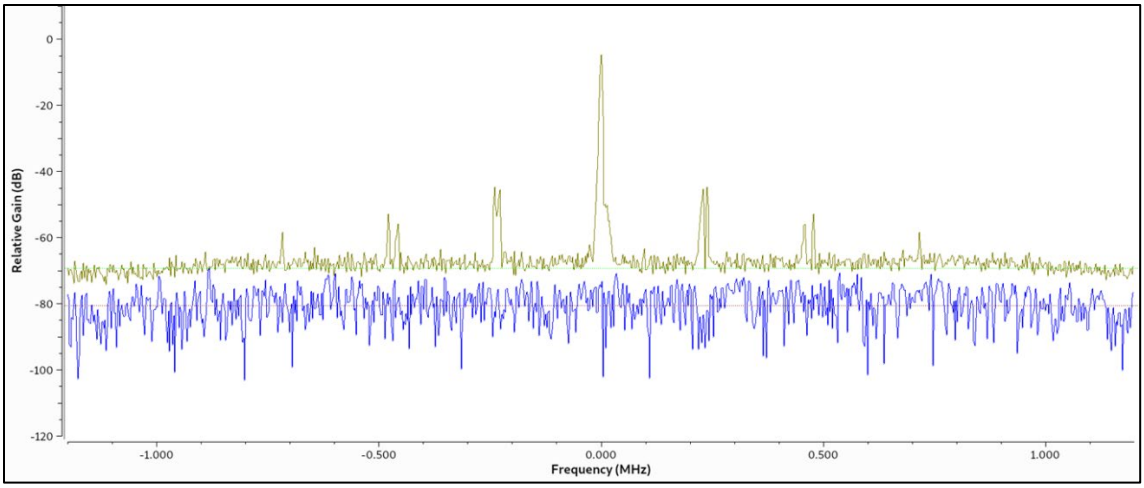


Figure 14. 154.570 MHz, SDR test capture at 3 feet.

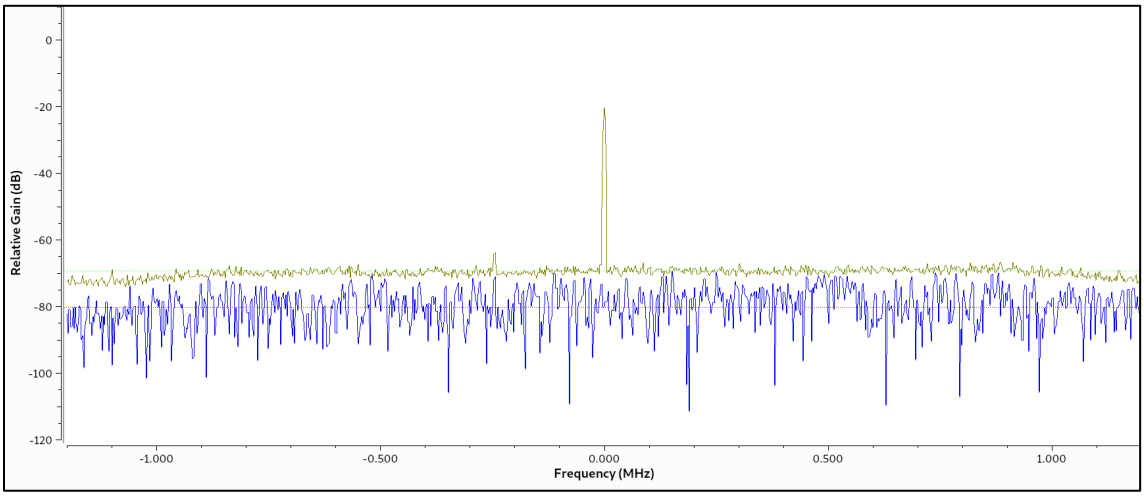


Figure 15. 154.570 MHz, SDR test capture at 50 feet.

The results from the VHF series of tests appear to be similar to the results of the UHF tests. At 3 feet, the SDR receives too much power and displays mirrored frequency readings above and below the test frequency. Moving the TERA radio further away corrected the issue.

These test results suggest the SDR is capable of accurately capturing signals within a 2 MHz wide frequency band. The test also suggests that when the transmitting device is in close proximity of the SDR, the accuracy of the captured signal degrades. It is also worth noting the difference between the maximum relative gain between the spectrum analyzer

and the SDR. The relative gain is measured in decibels (dB), which is a unit-less ratio. The difference between the noise floor and the highest dB of the target frequency for both the spectrum analyzer and the SDR were around 70dB, comparing only the 3 feet test as loss is expected as distance from the transmitting device increases. For this thesis, these tests suggest that the Raspberry Pi and SDR are capable of capturing a VHF or UHF signal for further processing.

## B. SYSTEM TESTING

The next phase focused on testing the assembled sensor node. After assembly, the tests described in Section A of this chapter were performed with no substantially different outcomes. Because the tests demonstrated the sensors ability to accurately depict signals, the next phase would use the sensor to conduct a survey of the GMRS and MURS bands and display the results. Both bands are available to use by the general public and a baseline capture of both bands were required before testing with the TERA radio. For the baseline band survey, the rtl_power program was used with the parameters in Table 5.

Table 5. Baseline band survey parameters for rtl_power.

| | GMRS | MURS |
|---|---|---|
| Frequency Range | 462-468 | 151-154 |
| Bin Size | 10 k | 10 k |
| Gain | 0 | 0 |
| Duration | 2 hour | 2 hour |
| Windowing | Blackman-Harris | Blackman-Harris |

Note that Kaiser windowing is not an available option in the rtl_power program. Blackman-Harris windowing was used due to its availability within the program and its similarity to Kaiser windowing in terms of side lobe compression (NI, 2016). However, the Blackman-Harris windowing was listed as an experimental option for the rtl_power program, while default windowing for the program is rectangular (Keen, 2015). The baseline scans are presented in Figures 16 and 17.

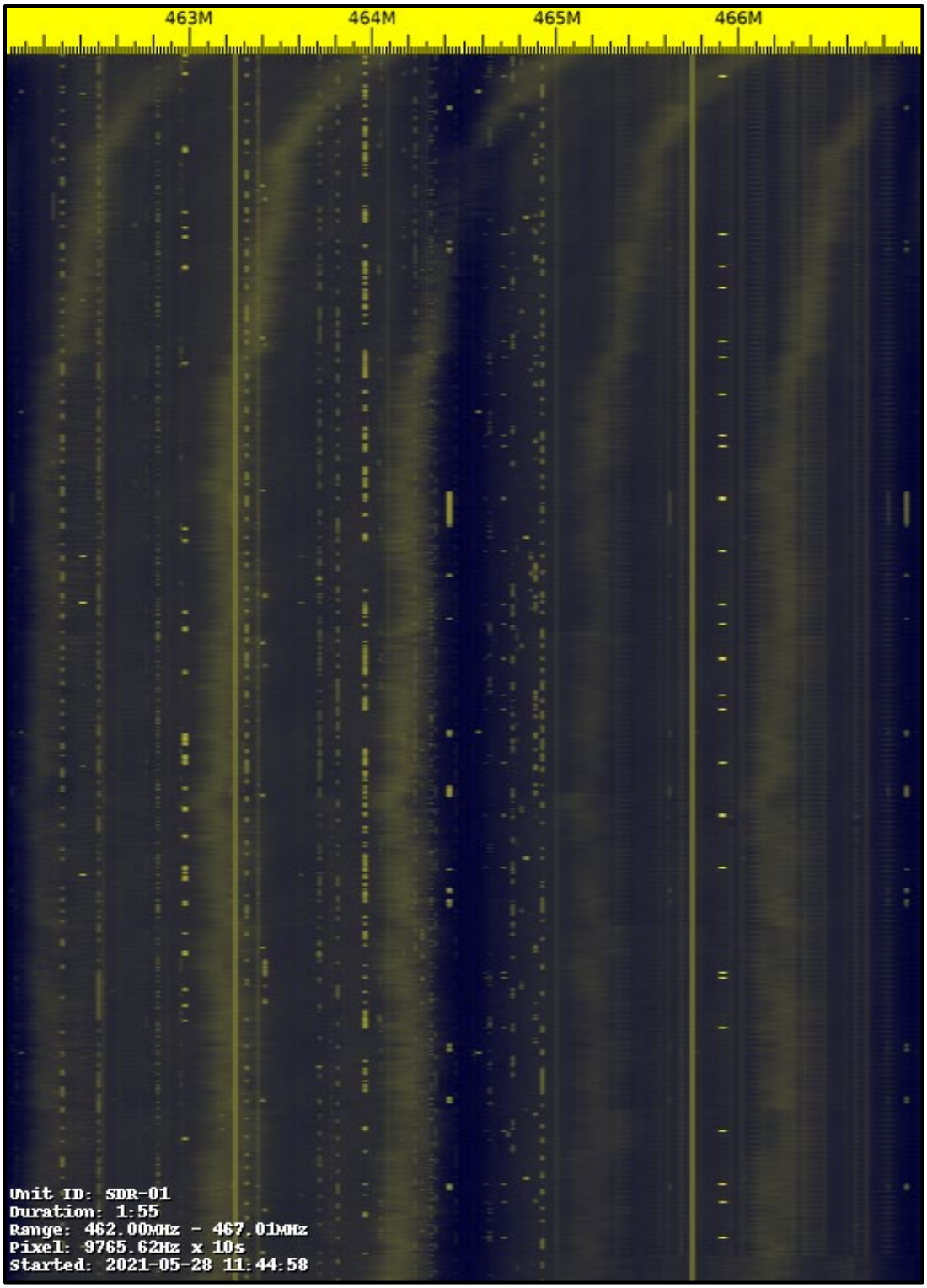


Figure 16. GMRS baseline scan, Blackman-Harris windowing

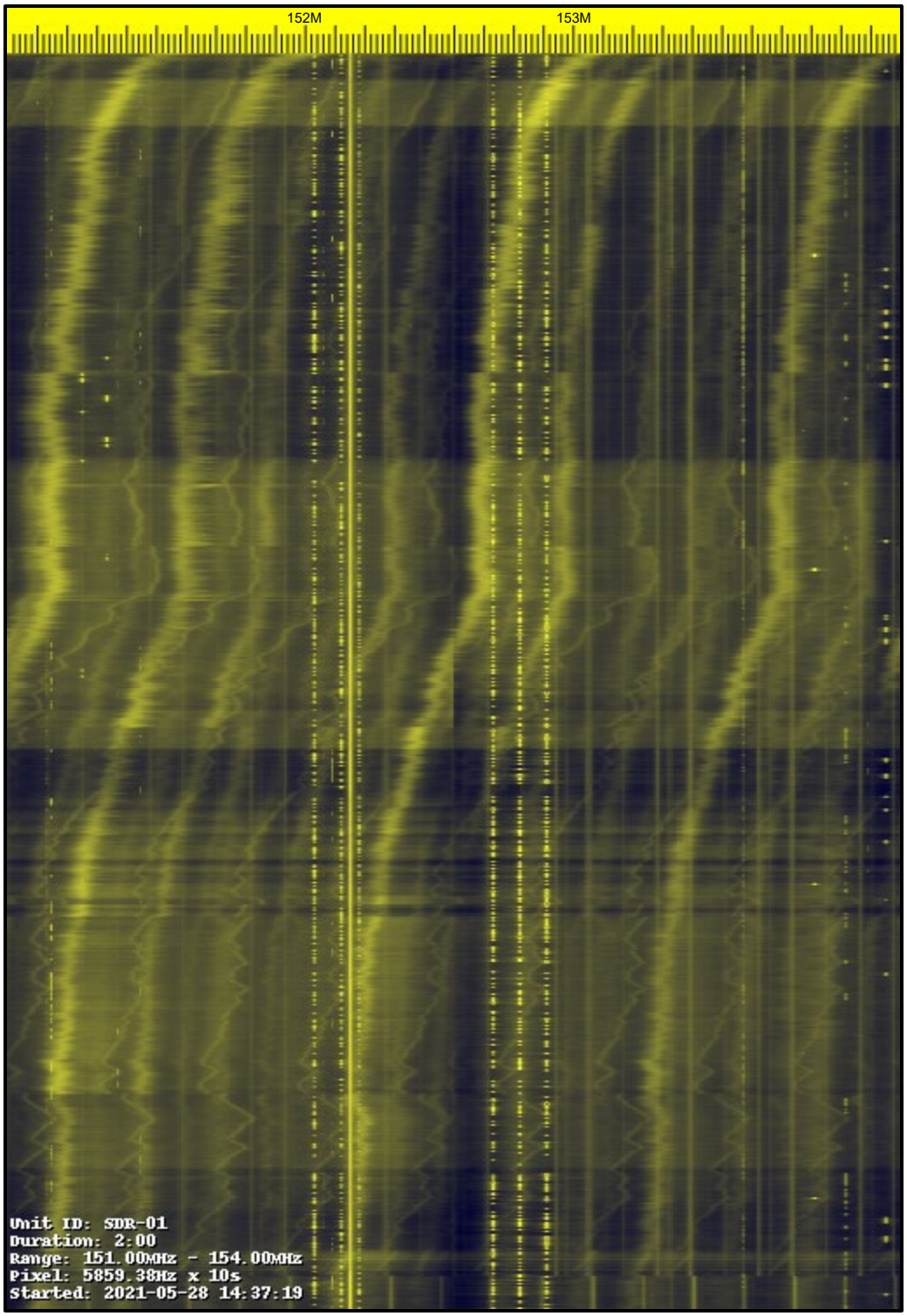


Figure 17. MURS baseline scan, Blackman-Harris windowing

The resulting band surveys reflected a large amount of distortion. The distortion appears as vertical lines that move back and forth across the horizontal axis. The distortion appears to be more prominent in the MURS frequency ranges (VHF) than in the GMRS ranges (UHF). It is also worth noting that the rtl_power program did not automatically label the frequency axis within the MURS scan because the range was too narrow. In order to provide clarity, labels were manually added to the image. A second set of baseline scans were conducted using rectangular windowing to attempt to minimize distortion and are presented in Figure 18 and Figure 19. The parameters for the scans and the resulting scans are presented in Table 6.

Table 6. Second baseline and testing scan parameters for rtl_power.

| | **GMRS** | **MURS** |
|---|---|---|
| Frequency Range | 462-468 | 151-157 |
| Bin Size | 15 k | 15 k |
| Gain | 0 | 0 |
| Duration | 2 hour | 2 hour |
| Windowing | Rectangular | Rectangular |

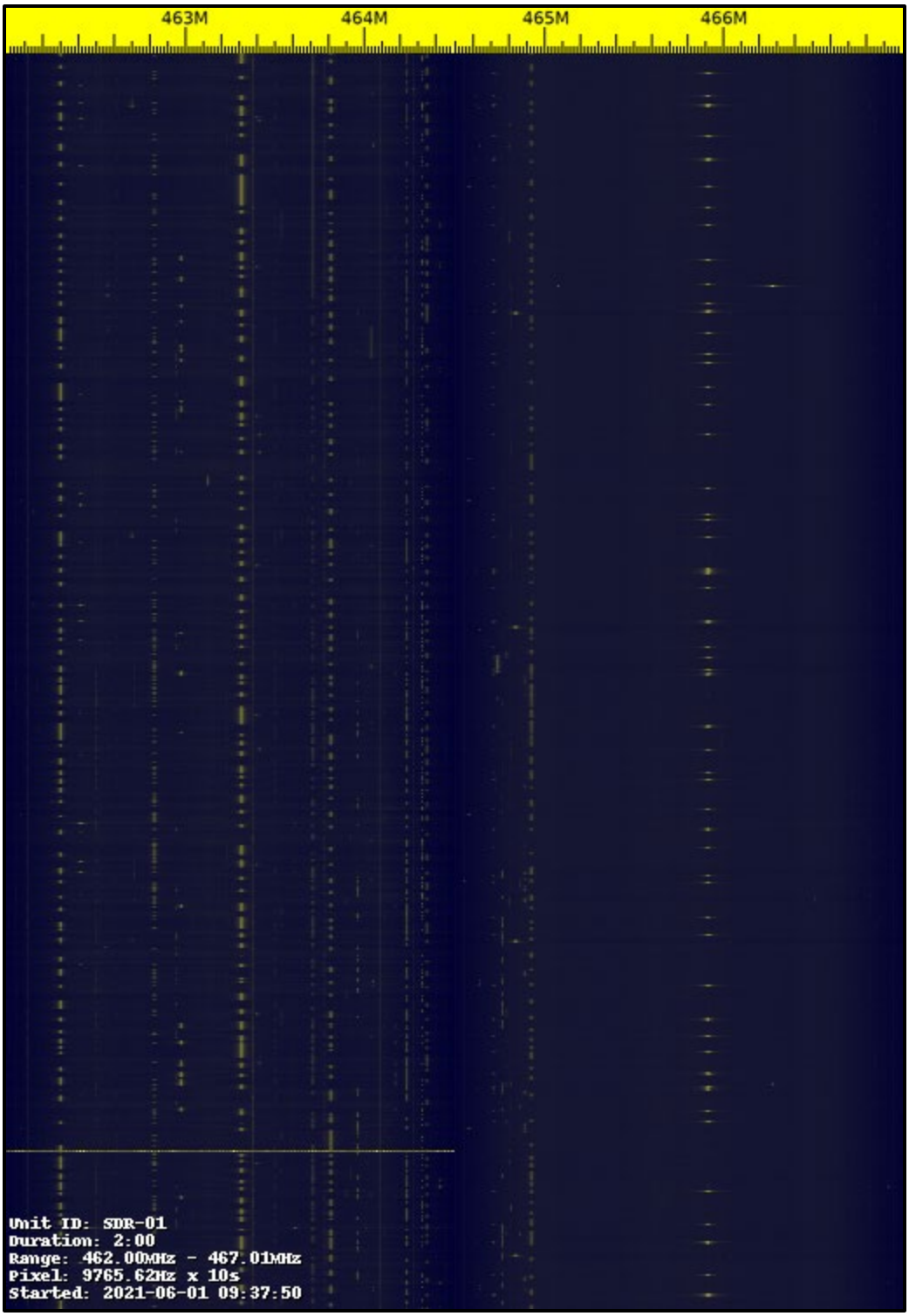


Figure 18. GMRS baseline scan, rectangular windowing

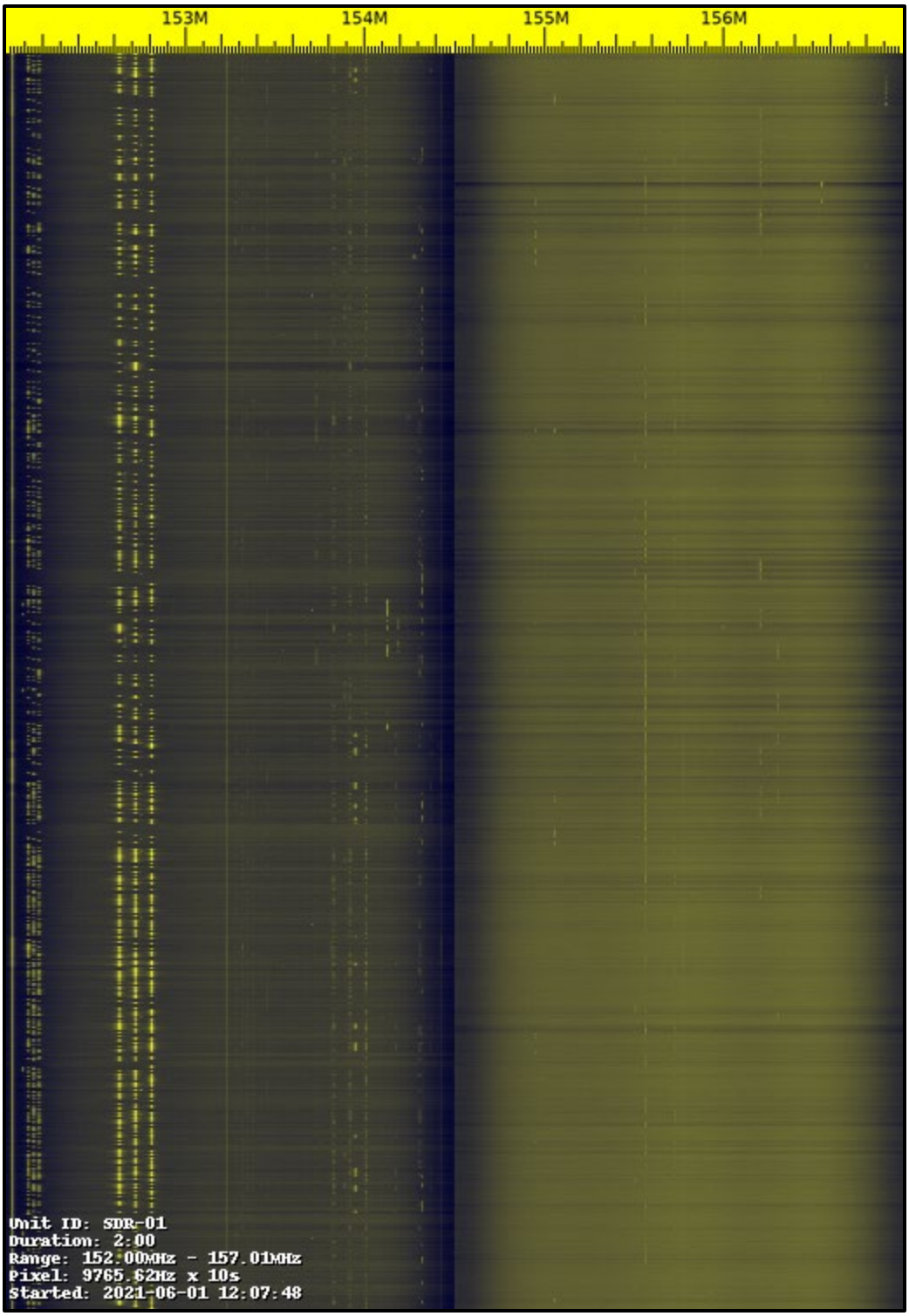


Figure 19. MURS baseline scan, rectangular windowing

The second part of this test was to conduct the shorter scans while transmitting on a known frequency and at a known interval. For the GMRS test, 462.550 MHz was used as the known frequency. The TERA radio was programmed with the test frequency and set at 2W power. The MURS test used 151.88 MHz and transmitted at 2W as well. After the rtl_power program was started, the radio was keyed every ten minutes for a duration of three seconds at 50 feet from the sensor. Red arrows depict the transmissions received by the sensor. Figure 20 and Figure 21 depict the results.

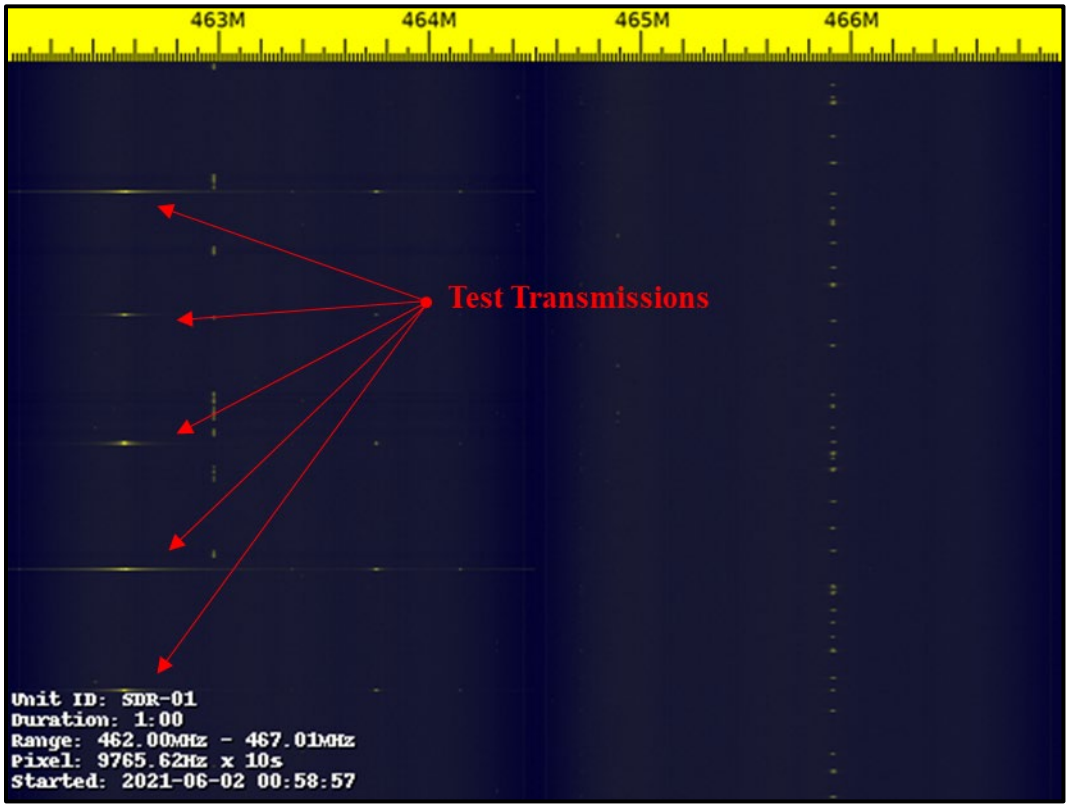


Figure 20. GMRS test scan, rectangular windowing.

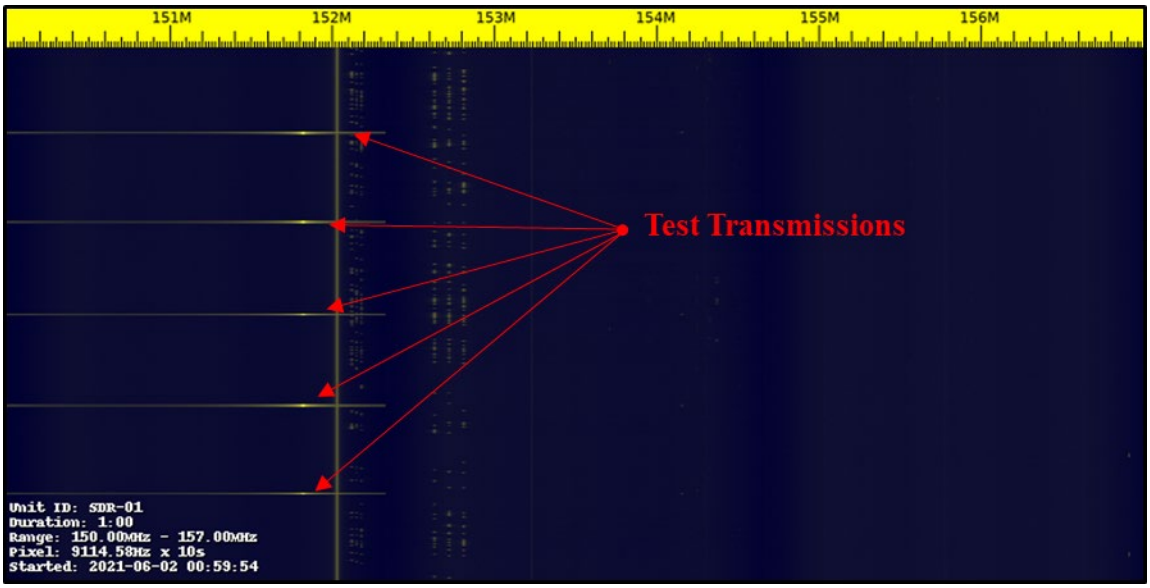


Figure 21. MURS test scan, rectangular windowing.

Both of the tests clearly depict the set of control transmissions. The signals appear to be strongest at the center frequency and show distortion along the horizontal axis. This might be caused by an excess amount of power being received at 50 feet when compared to the lower noise floor of a long duration scan. Additional public signals are seen on the test scan that correspond to signals captured in the baseline scans for both frequency ranges.

## C. EXPERIMENTATION

The experiment phase of this thesis will test the sensor nodes effectiveness at determining transmitter range based on relative gain. This experiment will use a radio operating at a known frequency and transmitting power. The sensor will record the received signal strengths from the radio as the radio moves away from the sensor and transmits at known distances and for known periods of time. The resulting data will be used to create an average relative gain for each distance. The average values will be used to create a function to determine distance based on relative gain at the receiving sensor.

### 1. Experiment Setup and Sensor Calibration

In order to determine the range, the sensors needed to be calibrated using known distances, frequencies and power outputs. To accomplish this, a parking lot aboard Naval

Postgraduate School with minimal line of sight obstructions and moderate distance was selected. A sensor was emplaced, and a line free of obstructions was determined. A 50-foot cord was used to measure out 14 radio transmitting locations, totaling 700 feet. Figure 22 depicts the sensor location and calibration line.


Figure 22. Sensor calibration location and training line.

The sensor used the code in appendix B in order to record the data for processing. The program iterated through 461.00 MHz to 468 MHz and recorded samples with 10 kHz spacing. The gain was set to 20 for the training and experiment. The TERA radio maintained the GMRS parameters used throughout the thesis, transmitting on 462.550 MHz at 2W. The transmitter was carried along the training line away from the sensor and transmitted between two and five seconds every 50 feet. At the 700-foot mark, the radio was keyed for 10 seconds to help identify the turnaround location in the data. The radio was used in an identical manner while returning to the sensor in order to maximize the collected data per training round. A total of three calibration rounds were completed. The dataset can be found in appendix E of this thesis.

### 2. Relative Gain versus Distance function

The data collected by the sensor node was sorted to find the transmissions and paired with the corresponding radio transmitting point. During sensor calibration, the radio was deliberately transmitted in longer durations in order to provide multiple data points for each distance. The resulting values were averaged for each test, and all the test values were averaged to create a single table of relative gain values compared to distance. These values are listed in Table 7.

Table 7. Sensor node average relative gain values versus distance.

| Relative Gain (dB) | Distance (ft) |
|---|---|
| 76.353 | 50 |
| 75.210 | 100 |
| 75.451 | 150 |
| 74.762 | 200 |
| 72.814 | 250 |
| 36.255 | 300 |
| 29.916 | 350 |
| 8.948 | 400 |
| 10.410 | 450 |
| 4.178 | 500 |
| 6.674 | 550 |
| 2.758 | 600 |
| 3.443 | 650 |
| 1.631 | 700 |

This data appears to progress in a non-linear manner. This may be due to partial obstructions of the radiated signal, such as a tree or light post. The data is provided in the form of a scatterplot with smoothed lines in Figure 23.

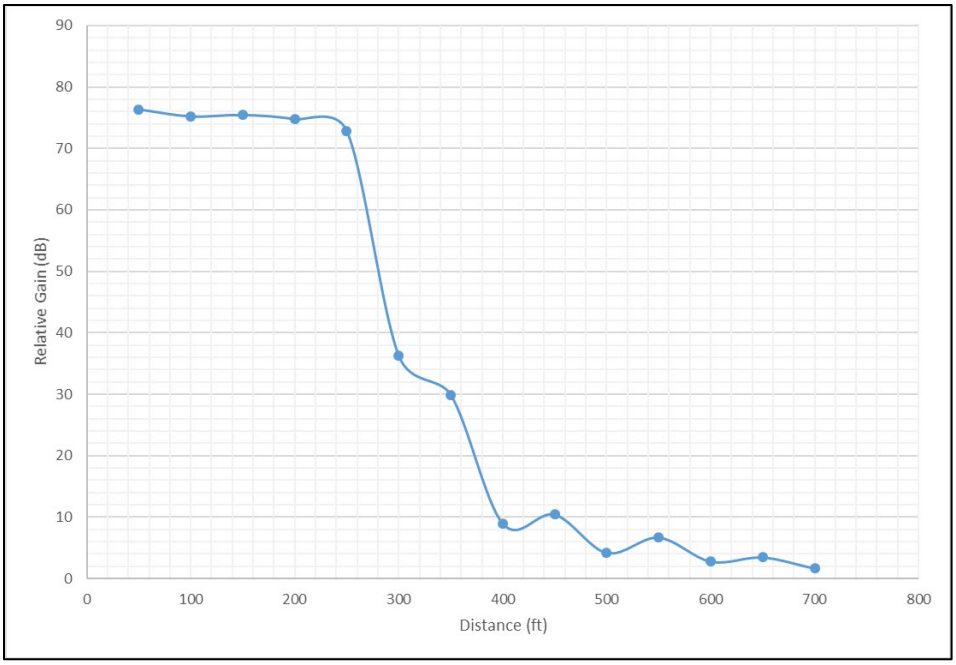


Figure 23. Scatterplot of relative gain versus distance.

### 3. Conducting the Experiment

The experiment was designed to test whether multiple low-cost SDR sensor nodes could be used to estimate a signal origin by comparing relative gain of a known frequency and power evaluated at two locations. This test used two sensor nodes spaced 600 feet apart in the same location as the sensor calibration aboard Naval Postgraduate School. The sensors were activated and began scanning the GMRS range with the same program and parameters as the calibration stage. A TERA radio programmed for 462.550 MHz and 2W transmitted a signal in three different locations with an unknown distance. The test transmission location GPS coordinates, as well as the sensor locations, were recorded and plotted on a map using onXmaps mapping software. The estimated distances were added to the onXmap as a shaded radius ring. Under perfect conditions, the test transmission should fall within the shared area between both sensors. The signal data from the two sensors were averaged from signal capture and is presented in Table 8.

Table 8. Sensor 1 and Sensor 2 captured signal Relative Gain and Estimated Range

| Test Number | Sensor 1 | | Sensor 2 | |
|---|---|---|---|---|
| | Average Relative Gain (dB) | Distance (ft) | Average Relative Gain (dB) | Distance (ft) |
| Test 1 | 11.656 | ~ 400 | 0.881 | >700 |
| Test 2 | 69.376 | ~ 250 | 2.046 | ~675 |
| Test 3 | 73.965 | ~ 225 | 0.372 | >700 |

The values were compared to the averaged calibration values. The ranges included in the table are estimated based on the calibration table and scatterplot smooth line estimate. The resulting map plots are presented in Figures 24–26. The limit for the radius for sensor 2 is set at 700 feet for tests 1 and 2 as the values are out of the bounds of data evaluated by this thesis.

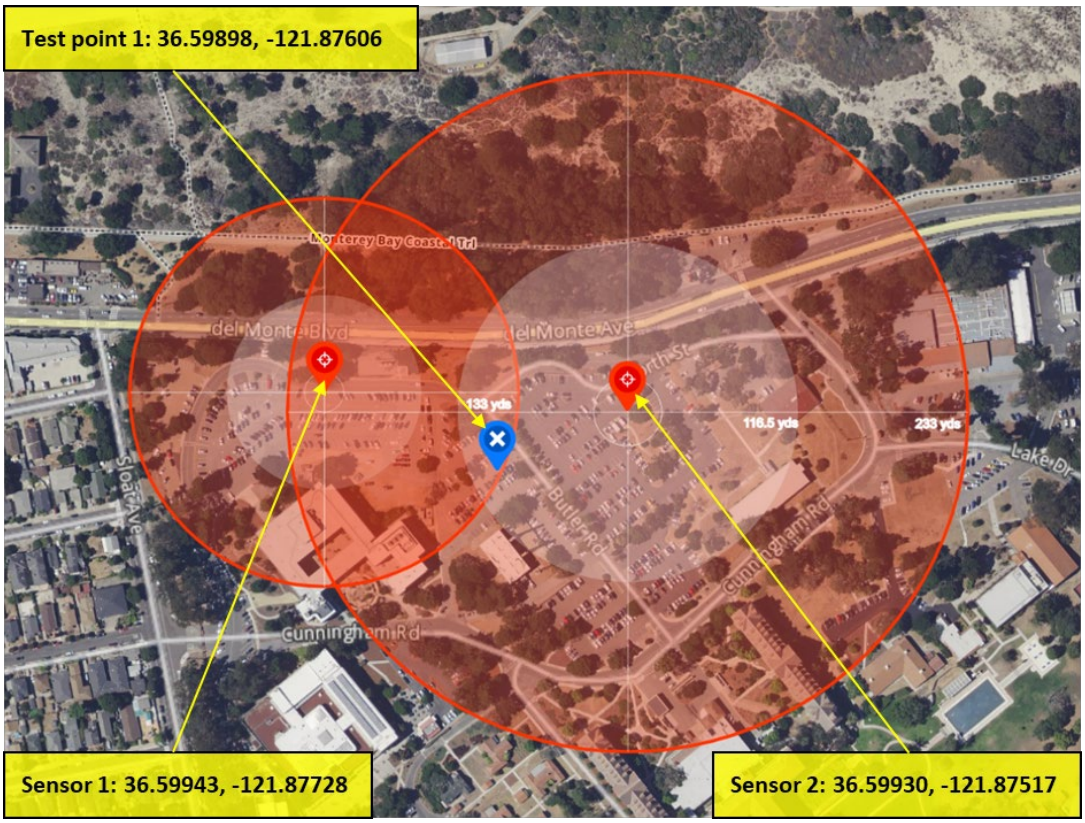


Figure 24. Experiment results, test 1.

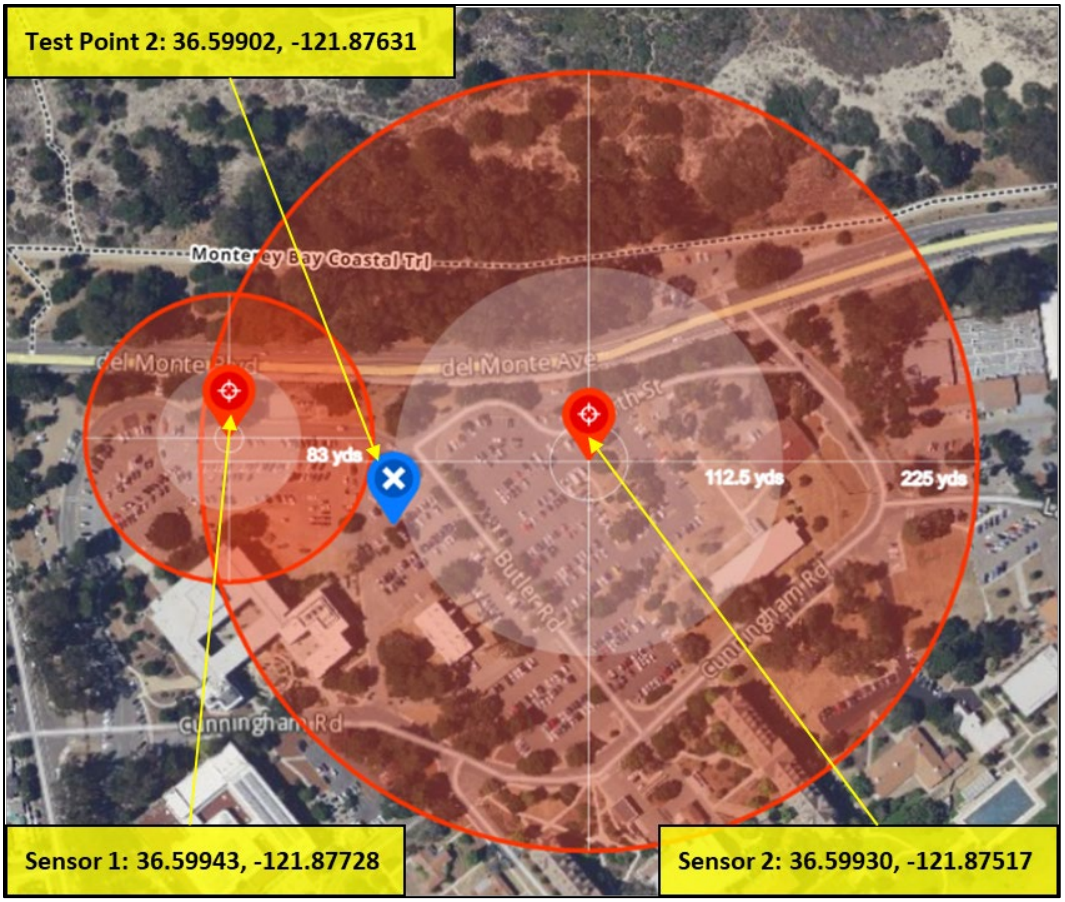


Figure 25. Experiment results, test 2.

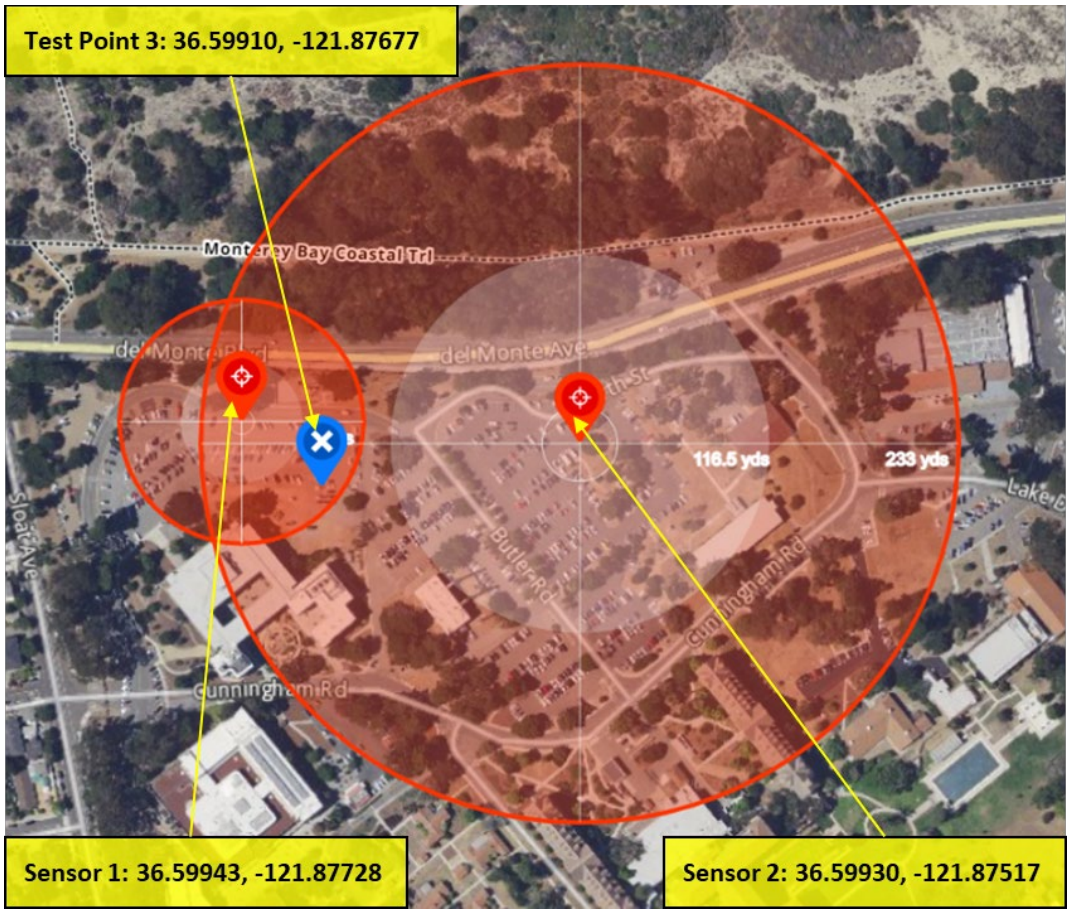


Figure 26. Experiment results, test 2.

Test 1 and test 3 depict the test transmission within the range of the sensor. Only in test 2 does the test transmission fall outside of the sensed range of either sensor. The tests also suggest that sensor 1 received more accurate relative gain of the test signal than sensor 2. This may be explained by a few factors. The line of sight was maintained to both sensors during each test transmission, but there was a clearer line to sensor 1. There was a row of trees between the test transmission location and sensor 2. These trees were adequately spaced to avoid obstruction in line of sight, but they may have impacted the total signal power absorbed by the sensor 2 antenna. Sensor 1 was located closer to more light poles than sensor 2. The metal light poles may have reflected more power to the sensor 1 antenna.

## D. DISCUSSION OF TESTING AND EXPERIMENTATION RESULTS

The RTL-SDR appears to be a very capable and cost effective method for capturing information from the EM spectrum. While using direct sampling of a known signal, the RTL-SDR is more than capable of accurately depicting frequency modulated voice signals. The 2 MHz sampling size can be employed in ways that broad coverage of the spectrum can be analyzed for strong signals. With regard to the host processor, the Raspberry Pi 4B was capable of processing the large quantities of samples produced by the SDR. At no point during the development, testing, or experimentation phase did the Raspberry Pi freeze or fail to run any processes. It is worth noting that the drivers for the R820T would not always shut down after the SDR completed a task. This resulted in an error that required manually shutting down the drivers from the CLI.

For use as a wideband scanning device, the complete sensor node appears to accurately depict various transmissions across a 10 MHz band. The test scans suggest that the accuracy of the received power decreases at close ranges, which is consistent with the individual component tests of the RTL-SDR. Another limitation of the device is the 2 MHz scanning width of the RTL-SDR. In order to conduct a scan, the SDR must iterate though 2 MHz sections, which may result in a loss of accurate spectrum coverage in terms of the time. This is especially true for short transmissions or anti-jamming techniques, such as frequency hopping. Additionally, transmitting entities may use terrain to block or reduce line of sight to the sensor resulting in lower received signal power, or no received signal at all. This effect may be reduced if the sensor is placed at higher elevation or on a raised structure, such as an antenna mast. Use of multiple sensors placed in strategic locations may also reduce missed signal captures from terrain masking. By communicating through local cellular networks, the sensor itself does not emit a unique signal in an urban environment. This may increase the sensor's survivability against similar technology by providing EM camouflage in a populated area.

The experiment based on determining transmitter distance from the sensor node yielded interesting results. Two of the three tests depicted the test transmission within the radius of both sensors range estimate rings. One of the three tests depicted the test transmission in only one of the sensor range estimate rings. There were many known

variables used in the experiment that would likely not be available in a tactical scenario. The experiment was based on the received signal strength, which can change depending on many factors. First, the received signal strength is measured in values relative to other signal strengths. This ratio can change significantly by changing the gain that the SDR processes samples. The distance equation is also based on transmitting at a known power. Transmitting at a higher power and a further distance can result in similar received signal strength ratios as a radio transmitting at a lower power but closer to the sensor. The received signal also depends on clear line of sight to the transmitting device. The experiment parameters sought to minimize any obstructions in the line of sight, such as buildings, foliage, or terrain. Obstructions can also reflect power, resulting in inaccurate measurements. All of these factors will severely affect the received signal strength.

# V. COST EFFECTIVENESS ANALYSIS

## A. INTRODUCTION

A key factor behind the motivation for this thesis is the low cost associated with the proliferation of advanced technology. In order to highlight the opportunity behind the investment into SDR research, this thesis includes a high-level cost-effectiveness analysis of employing a network of SDR based EM sensors across a Marine Expeditionary Unit (MEU) Ground Combat Element (GCE) Infantry Battalion (Bn) Rifle fire teams. The alternative approach considers the cost associated with employing Radio Battalion (RadBn) augments for tactical signals intelligence (SIGINT). The design of both this chapter and the analysis are based on the models presented by Boardman, Greenberg, Vining, and Weimer (Boardman et al.) (2018).

### 1. Status Quo (SQ)

No EM sensing capability exists organic to the MEU GCE Infantry Bn (MCRP 1–10.1). In order to conduct any EM sensing, the Infantry Bn must request support from the MEU Command Element (CE). This support is generally conducted through augmenting forces, either individual Marines or a small group of Marines assigned in support of a unit. The size of the RadBn augment to a standard MEU is not established by doctrine. It will depend on the needs of the MEU and anticipated mission (MCRP 1–10.1). Within a standard MEU, there is one Infantry Bn. Within the Infantry Bn, there a total of:

- 3 Infantry Companies (3 per Infantry Bn)
- 9 Rifle Platoons (3 per Rifle Co)
- 27 Rifle Squads (3 per Rifle Platoon)
- 81 Rifle Fire Teams (3 per Rifle Squad)

There are three RadBns, each with a single MEU support company (MCRP 1–10.1). The size of the support company is not annotated in current Marine Corps publications.

### 2. Proposed Course of Action (COA)

The COA is to provide one networked EM sensor to each fireteam within a MEU Infantry Bn. The Infantry Bn Intelligence section (S-2) would maintain and operate the hub node that receives the spectrum analysis reports and alerts from each device.

### 3. Stakeholders

The stakeholders in this model include individual Marines, the MEU GCE, the USMC and the U.S. taxpayer. Arguably, the Marines in the deployed environment have the most at stake. Their safety and ability to conduct their missions efficiently depend on the equipment, systems, and procedures provided to them. The success of the MEU GCE depends upon the success of the Marines executing their mission. The MEU GCE Commander has the authority to employ Marines and equipment in the manner they find appropriate. In addition to this authority, the GCE Commander is responsible for the Marines and equipment under their charge. The USMC is a large stakeholder in this cost-effectiveness analysis. The Marine Corps must invest into new technologies to maintain warfighting capability, but it also must do so on a limited budget. Last, but certainly not least, the U.S. taxpayer is a stakeholder. The money they entrust to the government in the form of taxes should produce a return to the taxpayers. In this model, that return may be viewed as a very small increase to national defense.

### 4. Assumptions

This model was developed on multiple assumptions. They are presented in the following list:

- The sensor node prototype in this thesis is assumed to have been refined into a working system capable of capturing EM signals of interest.
- The RadBn augments aboard a standard MEU are capable of providing at least six SIGINT MOS Marines (2629s) and three spectrum analyzers per GCE Infantry Bn.

- Each RadBn MEU Support Company is capable of simultaneously supporting all assigned MEUs.
- The three RadBn augments are in the E-4 paygrade with three years' active service.
- The Bn S-2 section processes all collected signals regardless of source.
- Costs for basic training and military occupational specialty (MOS) school are not considered.
- The same spectrum analyzer used throughout this thesis will be employed in the alternative.
- The increase in total collected signals per number of units in an area is a fixed, non-exponential increase.
- No additional training is needed for Infantry Bn Rifle fireteams or Infantry Bn S-2 to employ the SDR sensor system.
- Opportunity costs will not be considered.
- Discount rates will not be applied.

### 5. Steps Used for the Cost Effectiveness Analysis

The steps used to conduct the cost effectiveness analysis are based on the cost benefit analysis model by Boardman et al. (2018). This analysis does not seek to monetize the impacts of the SQ or COA and alternatives. As such, the steps have been modified to examine the relationship between the COA and alternatives within the scope of this thesis and the assumptions stated in this chapter.

1. Specify the alternative.
2. Identify common unit of effectiveness.
3. Identify costs.

4. Compute cost-effectiveness ratio.

5. Make a recommendation.

## B. ALTERNATIVE TO THE COA

The alternative to the proposed COA includes utilizing the MEU CE RadBn 2629s for each MEU GCE Infantry Bn. This would result in a single RadBn augment for each GCE Rifle company. This individual Marine is highly specialized and provides significant SIGINT capabilities in addition to EM sensing provided by the SDR sensor. However, for the purpose of this analysis the capabilities beyond EM sensing will not be considered.

## C. IDENTIFY COMMON UNIT OF EFFECTIVENESS

In order to compare measures of effectiveness, a common unit of effectiveness must be established (Boardman et al., 2018, Chapter 18, Para 2). Signals intelligence is the common function between both the RadBn augment and the SDR sensor, so the unit of effectiveness for this analysis will be measured in arbitrary *spectrum coverage* units given by the following equation:

$$SC = B * S_A * (1 + ((N_u - 1) * L))$$

where *SC* is spectrum coverage, *B* is spectrum coverage per hour, $S_A$ is estimated signal accuracy factor, $N_u$ is the number of units employed at a given time, and *L* is an estimated location factor. *SC* is an arbitrary unit representing the *effectiveness units* as described by Boardman et al. (2018 Chapter 18, Para 6). *B* is a unit less value depicting the ratio of spectrum coverage per hour. It is important to recognize the difference in accuracy between the spectrum analyzer and the SDR sensor, therefore the $S_A$ variable was included to capture the difference. This value is an estimate and should be updated if additional research into the SDR sensor is completed. $N_u$ represents the number of units in an area, while *L* provides a factor to increase the amount of collected signal based on the number of sensors. The latter half of the equation given by $(1 + ((N_u - 1) * L))$ provides a method of only providing an increase by *L* if the number of units is greater than one.

## D. IDENTIFY COST

There are a large number of marginal costs and cost savings associated with the COA, when compared to the status quo. Some of the monetized costs tied to the COA and alternative include the acquisition cost per device, networking support, and storing the data. This thesis includes a high-order estimate of cost effectiveness that incorporates only the largest cost elements, while the less consequential costs are not estimated. This analysis will estimate the costs of the SDR sensors and the spectrum analyzer. Accurate pricing information was not available for the model of spectrum analyzer used in this thesis and a comparable alternative was selected.

The salaries of the 2629s will be included, all in 2021 constant dollars. The salaries of the S-2 section personnel will not be included in this analysis; it is assumed that they will process the signal intelligence regardless of the source. The costs of each item are presented in Table 9.

Table 9. Considered costs. Adapted from Amazon (n.d-d.) and DOD (2021).

| Item | Cost |
|---|---|
| Monthly Salary, E-4 with 3 years' service | $2507.10 |
| Anritsu MT8221B | $7,450 |
| SDR Sensor Node | $250.56 |

## E. COMPUTE COST-EFFECTIVENESS RATIO

### 1. Effectiveness

The effectiveness calculation will follow the equation established earlier. For this analysis, the $S_A$ value for the spectrum analyzer will be set to 1. This represents 100% accuracy. The $S_A$ value for the SDR sensor will be set at .95, or 95% the accuracy of spectrum analyzer. According to the manufacturer the Anritsu MT8221B is capable of scanning 30 MHz of the EM spectrum in .01 seconds, while the SDR sensor is limited to 2 MHz at approximately .026 seconds (Anritsu, 2013). *B* is determined for both the spectrum analyzer and SDR sensor and presented in Table 10.

Table 10. Spectrum coverage per hour.

| | Anritsu MT8221B | SDR Sensor |
|---|---|---|
| Spectrum coverage (MHz) | 30 | 2 |
| Scan Time (seconds) | .01 | .026 |
| Spectrum coverage per hour (MHz) | 2.77E5 | 1.08E7 |
| Spectrum Coverage per hour as unit less value | 2.77 | 108 |

With these values, the *SC* can be determined. Table 11 lists the *SC* values for the identified parameters. The calculations for these values can be found in appendix F.

Table 11. Spectrum coverage values.

| | *SC* value per hour | *SC* value per month, continuous operation | *SC* value per 6 months, continuous operation |
|---|---|---|---|
| (1) SDR sensor | 2.1 | 1,512 | 9,072 |
| (81) SDR sensors | 189.3 | 136,296 | 817,776 |
| (1) MT8221B spectrum analyzer | 108 | 77,760 | 466,560 |
| (3) MT8221B spectrum analyzer | 345.6 | 248,832 | 1,492,992 |

### 2. Cost

The costs of the devices will be presented in the form of running the COA parameters against the alternative parameters, which includes 81 SDR sensors and 3 spectrum analyzers. The analysis includes the cost for running the both the COA and alternative for one month and six months.

The salaries for the 2629s are based on a deployed environment with a 30 workday month. This analysis considers an average deployed working day as 12 hours. Because of this, a total of two 2629s are required to continuously operate a single spectrum analyzer. Table 12 includes these costs.

Table 12. Cost per single month and 6 month operations.

| | Cost/Unit | Quantity Cost | Cost to operate for 1 month | Cost to operate for 6 months |
|---|---|---|---|---|
| (81) SDR Sensors | $250.56 | $20,295.36 | $20,295.36 | $20,295.36 |
| | | **TOTAL** | **$20,295.36** | **$20,295.36** |
| (3) MT8221B | $7450.00 | $22,350.00 | $22,350.00 | $22,350.00 |
| (6) RadBn 2629s | $2507.10 | $15,042.60 | $15,042.60 | $90,255.60 |
| | | **TOTAL** | **$37,392.60** | **$112,605.60** |

### 3. Effectiveness-Cost Ratios

This section examines the ratio as an effectiveness-cost ratio by dividing the effectiveness units of the COA and alternative by their respective costs (Boardman et al., 2018, Chapter 18, Para 2). The COA proposes equipping the 81Rifle fire teams with the SDR sensors. The alternative includes six 2629s using three MT8221B spectrum analyzers. The ratios are given as:

$$\boldsymbol{COA_{1Month}} = \frac{\$20{,}295.36}{136{,}296} = \mathbf{0.149}$$

$$\boldsymbol{Alt_{1Month}} = \frac{\$37{,}392.60}{248{,}832} = \mathbf{0.150}$$

$$\boldsymbol{COA_{6Months}} = \frac{\$20{,}295.36}{817{,}776} = \mathbf{0.025}$$

$$\boldsymbol{Alt_{6Months}} = \frac{\$112{,}605.60}{1{,}492{,}992} = \mathbf{0.075}$$

The resulting cost-effectiveness ratios from the model suggest that the COA is competitive with the alternative. It is possible that over a longer test period the COA may outperform the alternate in terms of cost-effectiveness ratio. The ratios were calculated for a period of 12 months using the same parameters. The effectiveness values, costs, and ratios for the 12-month period are included in appendix G. The data was input into a scatterplot and the data points connected using smoothed lines. This graph is depicted in Figure 27.

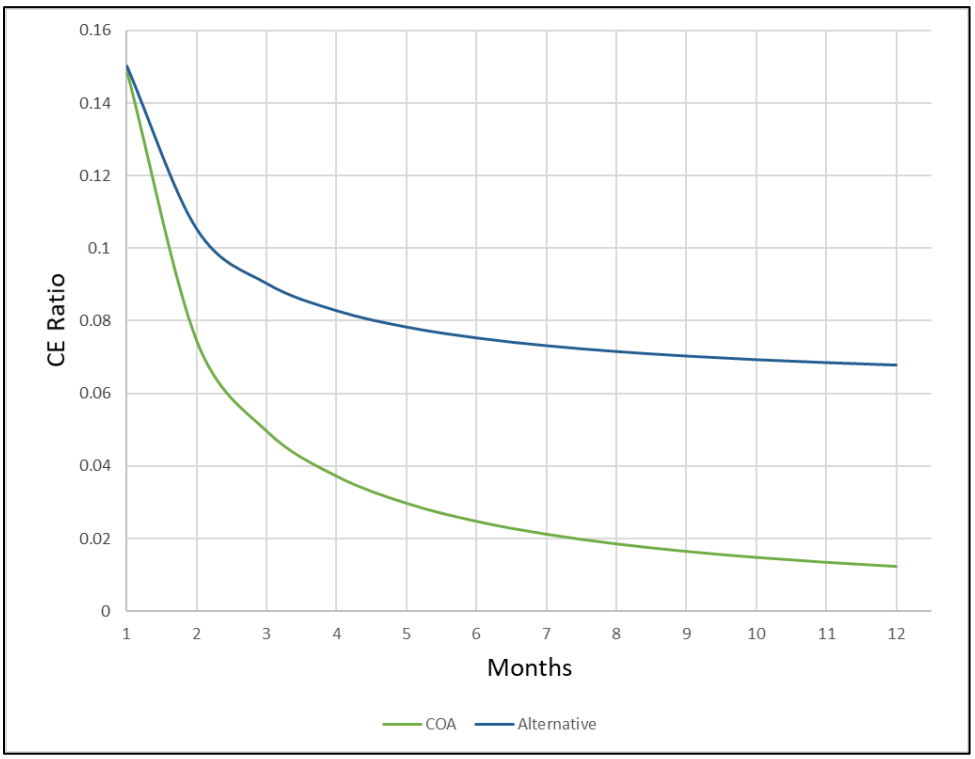


Figure 27. COA and alternative CE ratios over 12 months.

## F. DISCUSSION AND RECOMMENDATION

This model provides a high-level cost-effectiveness analysis comparing specific attributes of the SDR sensor employment and alternative options. Although the model suggests the SDR sensor employment might provide a lower ratio of cost for effectiveness, many items should be investigated or researched further to provide a more accurate perspective. The signal accuracy factor of the SDR sensor compared to the spectrum analyzer is based on test results conducted within this thesis and should be further tested to

determine a higher fidelity factor. Along the same lines, additional research should be conducted to determine a more accurate estimate of the increase of signals captured based on the number of devices collecting signals and their locations.

Many other factors were not considered by this analysis. Cost savings through automating the signals collection and analysis process may be realized with further testing. This may reduce the number of Marines required to deploy in a hostile area, which may ultimately lead to a reduction in casualties suffered. Other factors include the cost of cyber security risks introduced by remotely operating the sensor network.

THIS PAGE INTENTIONALLY LEFT BLANK

# VI. CONCLUSION AND RECOMMENDATIONS

## A. SUMMARY

The ability to sense activity within the EM spectrum is becoming increasingly important. The proliferation of technologies that use the EM spectrum has increased significantly, both in warfighting and domestic domains. As adversaries of the United States invest in these technologies, the ability to detect them should be approached with equivalent or greater interest.

This thesis explored the opportunities presented by low-cost and widely available technology to reach into the expanse of invisible information contained in the EM spectrum. The SDR sensor prototype demonstrated the capability of these COTS devices by remotely capturing and displaying signals from the GMRS and MURS frequency ranges. As developed, the SDR sensor may be useful in conducting wideband spectrum surveys to identify signals of interest or to develop a database of signal patterns from a unit conducting operations. This may be particularly beneficial to organizations using direction finding techniques. The signals of interest and signal patterns may increase the overall effectiveness in located the sources of transmission. By employing 4G cellular communications standards, the SDR sensor may blend into urban environments. This type of technology could easily collect signal information for long durations without causing unusual spectrum signatures within a heavily populated area.

The experimentation using relative gain to identify the location of a signal source did not suggest a high level of accuracy. This is likely due to a multitude of factors that affect the received power level. There are other parameters that can be used to determine signal source location that are more reliable, such as comparing the time the signal is received between multiple sensor nodes.

The benefits of using this technology are abundant. Introducing multiple sensors across a space will likely yield more information than a single source. Networking these devices paves the way for process automation, such as automated signal processing and

anomaly detection. This may increase our effectiveness at EM sensing while also driving down costs to conduct the sensing.

## B. LIMITATIONS

The testing and experimentation using the prototype used a small portion of the EM spectrum. The ability to use a lab or travel for research purposed was restricted due to the COVID-19 pandemic during the writing of this thesis. These restrictions required all thesis research and testing to be conducted within a personal domicile. All testing took place within publicly available frequency bands and with equipment and transmitting power authorized by the FCC.

The testing, calibration, and experiment did not take place within a controlled environment. Future research within a controlled laboratory environment may yield more accurate results.

## C. RECOMMENDED FURTHER RESEARCH

With additional research time, the next area of focus for this thesis would include integrating GPS-updated time across the sensor network in order to test the ability of the sensors to compare received signal timestamps. Additional areas of focus would seek to answer the following questions:

- Can COTS SDRs compare GPS-updated universal time of signals received between units?
  Can COTS SDRs be used to conduct direction finding?
- Can COTS SDRs be employed to detect drones or drone swarms?
- Can multiple SDRs be utilized on a single sensor node to increase the scanning capability of the device?
- Can COTS SDRs be employed to detect phase-shifts within captured signals?
- Can the SDR sensor prototype detect frequency hopping transmissions?
- What effect do high transmitting power devices have on COTS SDRs?

# APPENDIX A. RASPBERRY PI 4B SETUP

Change host name to desired naming convention

1. Sudo nano /etc/hosts
2. Sudo nano /etc/hostname

i) Add name from part a

3. Reboot

Install following on Raspbian:

4. Gedit

i) Sudo apt-get install gedit

6. Python3.7

i) Sudo apt-get install python3.7

Rtl-sdr driver install (Mead, 2015)

7. pi@raspberrypi ~ $ sudo raspi-config

i) Choose option 1 to "Expand Filesystem" - Ensures that all of the SD card storage is available to the OS

ii) Choose Finish and reboot

8. pi@raspberrypi ~ $ sudo apt-get update
9. pi@raspberrypi ~ $ sudo apt-get upgrade
10. pi@raspberrypi ~ $ sudo mv no-rtl.conf /etc/modprobe.d/
11. pi@raspberrypi ~ $ sudo apt-get install git-core
12. pi@raspberrypi ~ $ sudo apt-get install git
13. pi@raspberrypi ~ $ sudo apt-get install cmake
14. pi@raspberrypi ~ $ sudo apt-get install libusb-1.0-0-dev
15. pi@raspberrypi ~ $ sudo apt-get install build-essential

16. pi@raspberrypi ~ $ git clone git://git.osmocom.org/rtl-sdr.git
17. pi@raspberrypi ~ $ cd rtl-sdr/
18. pi@raspberrypi ~/rtl-sdr $ mkdir build
19. pi@raspberrypi ~/rtl-sdr $ cd build
20. pi@raspberrypi ~/rtl-sdr/build $ cmake ../ -DINSTALL_UDEV_RULES=ON
21. pi@raspberrypi ~/rtl-sdr/build $ make
22. pi@raspberrypi ~/rtl-sdr/build $ sudo make install
23. pi@raspberrypi ~/rtl-sdr/build $ sudo ldconfig
24. pi@raspberrypi ~/rtl-sdr/build $ cd ~
25. pi@raspberrypi ~ $ sudo cp ./rtl-sdr/rtl-sdr.rules /etc/udev/rules.d/
26. pi@raspberrypi ~ $ sudo reboot
27. pi@raspberrypi ~ $ sudo apt-get install libasound-dev
28. pi@raspberrypi ~ $ sudo apt-get install libpulse-dev
29. pi@raspberrypi ~ $ wget http://www.aishub.net/downloads/aisdecoder-1.0.0.tar.gz
30. pi@raspberrypi ~ $ tar zxvf aisdecoder-1.0.0.tar.gz
31. pi@raspberrypi ~ $ cd aisdecoder-1.0.0/
32. pi@raspberrypi ~/aisdecoder-1.0.0 $ mkdir build
33. pi@raspberrypi ~/aisdecoder-1.0.0 $ cd build/
34. pi@raspberrypi ~/aisdecoder-1.0.0/build $ cmake ../ -DCMAKE_BUILD_TYPE=Release
35. pi@raspberrypi ~/aisdecoder-1.0.0/build $ make
36. pi@raspberrypi ~/aisdecoder-1.0.0/build $ sudo cp aisdecoder /usr/local/bin

37. pi@raspberrypi ~/aisdecoder-1.0.0/build $ cd ~

38. pi@raspberrypi ~ $

Install following from PIP3

39. Numpy

i) Pip3 install numpy

40. Matlibplot

i) Pip3 install matlibplot

41. pyrtlsdr

i) Pip3install pyrtlsdr

Blacklist the RTL-SDR from the kernel control

42. Sudo gedit ban-rtl.conf

43. “blacklist dvb_usb_rtl28xxu”

44. Ctrl+S to save file

Reboot

THIS PAGE INTENTIONALLY LEFT BLANK

# APPENDIX B. PYTHON CODE

```
'''
Iterative Spectrum Scanner, GPS Enabled
Kenneth H. Liles
NPS, Curr 870/ITM
2020-2021

This program was designed to collected GMRS bandwidth
signals using a Rafael Micro R820T/2 SDR. This version
includes GPS coordinates (lat/long) in the alert
output.

Based on rtlsdr drivers for R820T/2 commands.

GPS integration based on work by
Eugene Bourokov, CENETIX Lab, NPS, 2019.

'''


#***** IMPORTS ***********
import numpy as np
import matplotlib.pyplot as plt
from rtlsdr import RtlSdr
import os
import sys
from datetime import datetime
import time
import logging
import serial


#****** FUNCTION DEFINITIONS *******

#This clears up an issue with an active kernel driver or device being used in another instance of librtlsdr.
def clear_sdr_kernel():
    os.system("sudo modprobe -r dvb_usb_rtl28xxu")
    print("Kernel clear order issued.")
clear_sdr_kernel()

def make_array():
    main_array = np.zeros((1, 513))
```

```
    return main_array


def get_gps():
    port = "/dev/ttyUSB2"
    ser = serial.Serial(port, baudrate = 9600, timeout = 0.5)
    if(ser.isOpen() == False):
        print("GPS serial failed.")
        logger.info("GPS serial failed.")
        pass
    while True:
        data = ser.readline()
        data = str(data)
        if "$GPGGA" in data[0:10]:
            s = data.split(",")
            if '0' in data[7]:
                print ("no satellite data available")
                return
            lat = decode(s[2])
            dirLat = s[3]
            lon = decode(s[4])
            dirLon = s[5]
            gps_data = ("Lat: %s(%s) Long: %s(%s)" %(lat, dirLat, lon, dirLon))
            return gps_data

def decode(coord):
    v = coord.split(".")
    head = v[0]
    tail =  v[1]
    deg = head[0:-2]
    min = head[-2:]
    return deg + "deg " + min + "." + tail + "min"

def rtl_scan(center_freq, freq_corr_ppm, main_array):
    psd = plt.psd
    sdr.sample_rate = 2.4e6  # Hz

    sdr.gain = 37
    sdr.center_freq =  center_freq*1e6
    samples = sdr.read_samples(256*1024)

    now = datetime.now()
    datetime_array = np.array([[now.timestamp()],[111]])
    scan_psd_array = psd(x = samples, NFFT=512 , Fs=sdr.sample_rate/1e6,
Fc=sdr.center_freq/1e6)
```

```
    np.set_printoptions(threshold=np.inf)
    scan_psd_array = np.array(scan_psd_array)


    #Following line is taken from the FFT value before the the value is divided
    #by the frequency. The .15 and 400 values are band and gain specific and will need to
be changed if the
    #freq band is altered.

    temp = np.where((scan_psd_array[0] > .15) & (scan_psd_array[0] < 400))
    alert_pwr = scan_psd_array[0, [temp]].copy()
    alert_freq = scan_psd_array[1, [temp]].copy()
    alert_array = np.vstack([alert_pwr, alert_freq])

    if alert_array.size > 0:
        print(alert_array)
        logger.info(alert_array)
    else:
        print("no alerts")

    scan_psd_array = np.concatenate((datetime_array, scan_psd_array), axis=1)

    main_array = np.vstack([main_array, scan_psd_array])
    return main_array

def log_array(main_array):
    now = datetime.now()
    dt_string = now.strftime("%Y%m%d %H:%M")
    np.savetxt("SCAN{}".format(dt_string), main_array, delimiter = ",")


#******* VARIABLE INITILIZATIONS ********
main_array = make_array()
scan_time_input = 300 #About 30 passes per minute: 150 for 5 min, 300 for 10 min
now = datetime.now()
dt_string = now.strftime("%Y%m%d %H:%M")
logging.basicConfig(filename="alert_log{}".format(dt_string),          format='%(asctime)s
%(message)s', filemode='a')
logger=logging.getLogger()
logger.setLevel(logging.INFO)


#***** MAIN CODE ******
sdr=RtlSdr()
```

```
freq_corr_ppm = 00  #freq_corr_ppm  PPM. This allows the user to correct the frequency
drift from the oscillating crystal
gpsdata = get_gps()

print(gpsdata)
logger.info(gpsdata)

for i in range(scan_time_input):
    center_freq = 462
    for i in range(3):
        main_array = rtl_scan(center_freq, freq_corr_ppm, main_array)
        center_freq = center_freq + 2


log_array(main_array)
print("Scan complete. Exiting program in 5s")
time.sleep(5)
exit()
```

```
'''
Iterative Spectrum Scanner, CSV Optimized
Kenneth H. Liles
NPS, Curr 870/ITM
2020-2021

This program was designed to collected GMRS bandwidth
signals using a Rafael Micro R820T/2 SDR. This version
optimizes the output array for post-scan .csv
manipulation.

Based on rtlsdr drivers for R820T/2 commands.
'''


#***** IMPORTS ***********
import numpy as np
import matplotlib.pyplot as plt
from rtlsdr import RtlSdr
import os
from datetime import datetime
import time



#****** FUNCTION DEFINITIONS *******

#This clears up an issue with an active kernel driver or device being used in another instance of librtlsdr.
def clear_sdr_kernel():
    os.system("sudo modprobe -r dvb_usb_rtl28xxu")
    print("Kernel clear order issued.")
clear_sdr_kernel()

def make_array():
    main_array = np.zeros((1, 769))
    return main_array

def make_small_array():
    now = datetime.now()
    small_array = np.array([now.timestamp()])
    return small_array

def main_array_add(main_array, small_array):
    main_array = np.vstack([main_array, small_array])
```

```
    return main_array

def get_freq_headers(center_freq, small_array):
    psd = plt.psd
    sdr.sample_rate = 2.4e6  # Hz
    sdr.gain = 37
    sdr.center_freq =  center_freq*1e6
    samples = sdr.read_samples(256*1024)
    scan_psd_array = psd(x = samples, NFFT=256 , Fs=sdr.sample_rate/1e6,
Fc=sdr.center_freq/1e6)
    np.set_printoptions(threshold=np.inf)
    scan_psd_array = np.array(scan_psd_array)
    small_array = np.concatenate((small_array, scan_psd_array[1]))
    return small_array

def rtl_scan(center_freq, small_array):
    psd = plt.psd
    sdr.sample_rate = 2.4e6  # Hz
    sdr.gain = 20
    sdr.center_freq =  center_freq*1e6
    samples = sdr.read_samples(256*1024)
    scan_psd_array = psd(x = samples, NFFT=256 , Fs=sdr.sample_rate/1e6,
Fc=sdr.center_freq/1e6)
    np.set_printoptions(threshold=np.inf)
    scan_psd_array = np.array(scan_psd_array)

    #Following line is taken from the FFT value before the the value is divided
    #by the frequency. The .15 and 400 values are band and gain specific and will need to
be changed if the
    #freq band is altered.

    temp = np.where((scan_psd_array[0] > .15) & (scan_psd_array[0] < 400))
    alert_pwr = scan_psd_array[0, [temp]].copy()
    alert_freq = scan_psd_array[1, [temp]].copy()
    alert_array = np.vstack([alert_pwr, alert_freq])

    if alert_array.size > 0:
        print(alert_array)
    else:
        print("no alerts")
    small_array = np.concatenate((small_array, scan_psd_array[0]))
    return small_array

def log_array(main_array):
    now = datetime.now()
```

```
    dt_string = now.strftime("%Y%m%d %H:%M")
    np.savetxt("SCAN{}".format(dt_string), main_array, delimiter = ",")


#******* VARIABLE INITILIZATIONS ********

main_array = make_array()
scan_time_input = 300 #About 30 passes per minute: 150 for 5 min, 300 for 10 min
small_array = make_small_array()
center_freq = 462
headercount = 0

#***** MAIN CODE ******
sdr=RtlSdr()
sdr.freq_correction = 20
print("Building headers...")
for i in range(3):
        small_array = get_freq_headers(center_freq, small_array)
        center_freq = center_freq + 2
        headercount = headercount + 1
        if headercount > 2:
            main_array = main_array_add(main_array, small_array)
        else:
            print("\n")
print("Scanning...")

for i in range(scan_time_input):
    small_array = make_small_array()
    center_freq = 462
    count = 0
    for i in range(3):
        small_array = rtl_scan(center_freq, small_array)
        center_freq = center_freq + 2
        count = count + 1
        if count > 2:
            main_array = main_array_add(main_array, small_array)
        else:
            print("\n")

log_array(main_array)
print("Scan complete. Exiting program in 5s")
time.sleep(5)
exit()
```

THIS PAGE INTENTIONALLY LEFT BLANK

# APPENDIX C. WAVESHARE SIM7600 TECHNICAL DATA AND SETUP

The following information is provided in the SIM7900X A-H manual (Waveshare, 2020)

Band: LTE-FDD B1/B3/B5/B7/B8/B20
LTE-TDD B38/B40/B41
Generation: 4G
Transmit Pwr: 0.25W
Data speed: LTE CAT 4; Uplink <50Mbps, Downlink < 150 Mbps

## Installation

The SIM7600X A-H card must be connected with a data-rated USB cable from a USB 3.0 port to the USB port on the 4G card. For GPS, a data-rated USB cable must connect to a USB2.0 (or higher) port from the Raspberry Pi to the UART port on the 4G card.

The following instructions were only tested with AT&T service.

The following is based on an internet forum post (mkrzysztofowicz, 2019).

1. pi:~$ sudo apt-get update && sudo apt-get install libqmi-utils udhcpc
2. pi:~$ sudo qmicli -d /dev/cdc-wdm0 --dms-set-operating-mode='online'

Check to ensure it is powering on:

3. qmicli -d /dev/cdc-wdm0 --dms-get-operating-mode

It should report "online." If not, reboot and repeat the above.

4. pi:~$ sudo qmicli -d /dev/cdc-wdm0 --device-open-proxy --wds-start-network="ip-type=4" --client-no-release-cid
5. pi:~$ sudo qmicli -d /dev/cdc-wdm0 -w
6. pi:~$ sudo ip link set wwan0 down
7. pi:~$ echo 'Y' | sudo tee /sys/class/net/wwan0/qmi/raw_ip
8. pi:~$ sudo ip link set wwan0 up
9. pi:~$ sudo ip link set wlan0 down

THIS PAGE INTENTIONALLY LEFT BLANK

# APPENDIX D. LINUX SHELL COMMANDS FOR HEADLESS RASPBERRY PI

**Disable 4G Connection**:

```
#!/bin/bash

cd
sudo ifconfig wwan0 down
sudo ifconfig wlan0 up
```

**Enable 4G Connection**:

```
#!/bin/bash

ip link set wlan0 down
sudo ip link set wwan0 down
echo 'Y' | sudo tee /sys/class/net/wwan0/qmi/raw_ip
sudo ip link set wwan0 up
sudo qmi-network  /dev/cdc-wdm0 start
sudo udhcpc -i wwan0
ip a s wwan0
exit
```

**Run gmrs_10m.py:**

```
#!/bin/bash

cd
python3 gmrs_10m.py
exit
```

NOTE: This is a simple script to run any python code via CLI. If the Raspberry Pi does not have a screen (called “headless”), the shell command will need to be run using:

pi@raspberrypi~ nohup ./[example.sh] &

in order to keep the Raspberry Pi from killing the shell and associated processes. It proved useful to make a shell script to start these processes through the Linux crontab process scheduler on reboot.

THIS PAGE INTENTIONALLY LEFT BLANK

# APPENDIX E. SENSOR CALIBRATION DATA

| Test 1 | | | | |
|---|---|---|---|---|
| **Date/Time** | **Frequency (MHz)** | | | **Distance (ft)** |
| | 462.544 | 462.553 | 462.563 | 50 |
| 6/2/2021 22:04:47.164 | 45.438 | 76.175 | 7.272 | 50 |
| 6/2/2021 22:04:49.969 | 45.484 | 76.146 | 7.250 | 50 |
| 6/2/2021 22:04:50.848 | 45.514 | 76.135 | 7.242 | 50 |
| Average | 45.479 | 76.152 | 7.255 | 50 |
| 6/2/2021 22:05:01.441 | 45.176 | 75.944 | 7.270 | 100 |
| 6/2/2021 22:05:02.732 | 45.309 | 75.800 | 7.217 | 100 |
| Average | 45.243 | 75.872 | 7.244 | 100 |
| 6/2/2021 22:05:13.052 | 44.312 | 74.105 | 7.042 | 150 |
| 6/2/2021 22:05:14.249 | 44.829 | 74.777 | 7.091 | 150 |
| Average | 44.571 | 74.441 | 7.066 | 150 |
| 6/2/2021 22:05:26.306 | 44.582 | 74.435 | 7.063 | 200 |
| 6/2/2021 22:05:28.905 | 43.729 | 73.308 | 6.992 | 200 |
| Average | 44.156 | 73.872 | 7.028 | 200 |
| 6/2/2021 22:05:39.155 | 41.722 | 69.480 | 6.568 | 250 |
| 6/2/2021 22:05:41.525 | 41.626 | 69.797 | 6.670 | 250 |
| Average | 41.674 | 69.639 | 6.619 | 250 |
| 6/2/2021 22:05:57.918 | 1.431 | 2.397 | 0.228 | 300 |
| 6/2/2021 22:05:58.604 | 5.082 | 8.458 | 0.799 | 300 |
| Average | 3.256 | 5.427 | 0.514 | 300 |
| 6/2/2021 22:06:10.949 | 30.226 | 50.292 | 4.751 | 350 |
| 6/2/2021 22:06:11.979 | 29.690 | 49.434 | 4.672 | 350 |
| Average | 29.958 | 49.863 | 4.711 | 350 |
| 6/2/2021 22:06:28.477 | 2.153 | 3.606 | 0.343 | 400 |
| 6/2/2021 22:06:30.972 | 2.016 | 3.354 | 0.317 | 400 |
| Average | 2.085 | 3.480 | 0.330 | 400 |
| 6/2/2021 22:06:44.013 | 0.761 | 1.271 | 0.121 | 450 |
| 6/2/2021 22:06:45.183 | 0.961 | 1.599 | 0.151 | 450 |
| Average | 0.861 | 1.435 | 0.136 | 450 |
| 6/2/2021 22:06:58.498 | 0.086 | 0.143 | 0.014 | 500 |
| 6/2/2021 22:07:00.657 | 0.139 | 0.232 | 0.022 | 500 |
| Average | 0.113 | 0.188 | 0.018 | 500 |
| 6/2/2021 22:07:17.514 | 1.157 | 1.955 | 0.189 | 550 |
| 6/2/2021 22:07:19.269 | 3.947 | 6.598 | 0.627 | 550 |
| 6/2/2021 22:07:20.636 | 2.314 | 3.856 | 0.365 | 550 |
| Average | 2.473 | 4.136 | 0.393 | 550 |
| 6/2/2021 22:07:32.052 | 0.794 | 1.318 | 0.124 | 600 |
| 6/2/2021 22:07:33.612 | 1.589 | 2.651 | 0.251 | 600 |

| | | | | |
|---|---|---|---|---|
| Average | 1.192 | 1.985 | 0.188 | 600 |
| 6/2/2021 22:07:46.545 | 2.305 | 3.835 | 0.362 | 650 |
| 6/2/2021 22:07:48.026 | 3.920 | 6.509 | 0.613 | 650 |
| 6/2/2021 22:07:49.284 | 4.254 | 7.116 | 0.678 | 650 |
| Average | 3.493 | 5.820 | 0.551 | 650 |
| 6/2/2021 22:08:01.096 | 0.257 | 0.429 | 0.041 | 700 |
| 6/2/2021 22:08:02.827 | 0.714 | 1.190 | 0.113 | 700 |
| 6/2/2021 22:08:04.234 | 0.507 | 0.842 | 0.079 | 700 |
| 6/2/2021 22:08:05.889 | 0.392 | 0.655 | 0.063 | 700 |
| 6/2/2021 22:08:07.914 | 0.456 | 0.762 | 0.073 | 700 |
| 6/2/2021 22:08:08.299 | 0.171 | 0.285 | 0.026 | 700 |
| Average | 0.416 | 0.694 | 0.066 | 700 |
| 6/2/2021 22:08:23.108 | 0.302 | 0.505 | 0.048 | 650 |
| 6/2/2021 22:08:25.273 | 0.473 | 0.787 | 0.075 | 650 |
| 6/2/2021 22:08:26.433 | 0.687 | 1.149 | 0.109 | 650 |
| Average | 0.487 | 0.814 | 0.077 | 650 |
| 6/2/2021 22:08:38.938 | 1.566 | 2.618 | 0.249 | 600 |
| 6/2/2021 22:08:40.157 | 2.835 | 4.739 | 0.451 | 600 |
| Average | 2.201 | 3.679 | 0.350 | 600 |
| 6/2/2021 22:08:53.495 | 0.130 | 0.216 | 0.021 | 550 |
| 6/2/2021 22:08:54.926 | 0.119 | 0.198 | 0.019 | 550 |
| Average | 0.124 | 0.207 | 0.020 | 550 |
| 6/2/2021 22:09:05.251 | 8.137 | 13.516 | 1.274 | 500 |
| 6/2/2021 22:09:07.956 | 6.599 | 11.027 | 1.046 | 500 |
| 6/2/2021 22:09:09.941 | 2.703 | 4.487 | 0.422 | 500 |
| Average | 5.813 | 9.677 | 0.914 | 500 |
| 6/2/2021 22:09:20.748 | 14.389 | 23.989 | 2.274 | 450 |
| Average | 14.389 | 23.989 | 2.274 | 450 |
| 6/2/2021 22:09:30.885 | 5.077 | 8.489 | 0.808 | 400 |
| 6/2/2021 22:09:31.908 | 10.850 | 18.012 | 1.697 | 400 |
| | 7.963 | 13.251 | 1.253 | 400 |
| 6/2/2021 22:09:42.001 | 21.318 | 35.634 | 3.386 | 350 |
| 6/2/2021 22:09:44.608 | 17.826 | 29.687 | 2.806 | 350 |
| Average | 19.572 | 32.660 | 3.096 | 350 |
| 6/2/2021 22:09:52.851 | 26.806 | 44.619 | 4.217 | 300 |
| 6/2/2021 22:09:54.639 | 30.367 | 50.428 | 4.750 | 300 |
| Average | 28.586 | 47.523 | 4.483 | 300 |
| 6/2/2021 22:10:01.507 | 44.523 | 74.434 | 7.084 | 250 |
| 6/2/2021 22:10:03.169 | 44.288 | 73.567 | 6.933 | 250 |
| Average | 44.406 | 74.001 | 7.009 | 250 |
| 6/2/2021 22:10:14.844 | 44.903 | 74.691 | 7.050 | 200 |
| 6/2/2021 22:10:16.244 | 45.103 | 74.923 | 7.059 | 200 |
| 6/2/2021 22:10:17.986 | 43.538 | 72.856 | 6.745 | 200 |
| Average | 44.515 | 74.157 | 6.951 | 200 |

| | | | | |
|---|---|---|---|---|
| 6/2/2021 22:10:27.125 | 45.440 | 75.609 | 7.137 | 150 |
| Average | 45.440 | 75.609 | 7.137 | 150 |
| 6/2/2021 22:10:36.205 | 45.726 | 75.930 | 7.155 | 100 |
| Average | 45.726 | 75.930 | 7.155 | 100 |
| 6/2/2021 22:10:44.434 | 45.717 | 75.919 | 7.150 | 50 |
| Average | 45.717 | 75.919 | 7.150 | 50 |

| **Test 2** | | | | |
|---|---|---|---|---|
| **Date/Time** | **Frequency (MHz)** | | | **Distance (ft)** |
| | 462.544 | 462.553 | 462.563 | 50 |
| 6/2/2021 22:15:06.791 | 45.518 | 76.105 | 7.234 | 50 |
| 6/2/2021 22:15:08.256 | 45.275 | 76.256 | 7.328 | 50 |
| Average | 45.397 | 76.181 | 7.281 | 50 |
| 6/2/2021 22:15:20.795 | 45.290 | 76.252 | 7.317 | 100 |
| 6/2/2021 22:15:22.293 | 45.464 | 76.135 | 7.251 | 100 |
| Average | 45.377 | 76.193 | 7.284 | 100 |
| 6/2/2021 22:15:32.018 | 45.094 | 75.764 | 7.258 | 150 |
| 6/2/2021 22:15:33.193 | 45.198 | 75.679 | 7.211 | 150 |
| 6/2/2021 22:15:35.336 | 45.140 | 75.981 | 7.081 | 150 |
| Average | 45.144 | 75.808 | 7.183 | 150 |
| 6/2/2021 22:15:46.088 | 44.518 | 74.610 | 7.114 | 200 |
| Average | 44.518 | 74.610 | 7.114 | 200 |
| 6/2/2021 22:15:58.326 | 43.632 | 73.192 | 6.993 | 250 |
| 6/2/2021 22:16:00.229 | 43.939 | 73.669 | 7.029 | 250 |
| 6/2/2021 22:16:01.517 | 43.771 | 73.681 | 6.867 | 250 |
| Average | 43.780 | 73.514 | 6.963 | 250 |
| 6/2/2021 22:16:12.612 | 20.212 | 34.029 | 3.271 | 300 |
| 6/2/2021 22:16:13.695 | 18.415 | 30.802 | 2.931 | 300 |
| Average | 19.314 | 32.415 | 3.101 | 300 |
| 6/2/2021 22:16:24.648 | 14.166 | 23.711 | 2.258 | 350 |
| 6/2/2021 22:16:26.589 | 16.825 | 28.133 | 2.675 | 350 |
| 6/2/2021 22:16:27.998 | 11.891 | 19.935 | 1.905 | 350 |
| Average | 14.294 | 23.926 | 2.279 | 350 |
| 6/2/2021 22:16:39.952 | 5.117 | 8.587 | 0.820 | 400 |
| 6/2/2021 22:16:40.274 | 4.962 | 8.324 | 0.794 | 400 |
| Average | 5.040 | 8.455 | 0.807 | 400 |
| 6/2/2021 22:16:51.381 | 0.949 | 1.587 | 0.151 | 450 |
| 6/2/2021 22:16:53.373 | 1.305 | 2.185 | 0.208 | 450 |
| 6/2/2021 22:16:54.078 | 2.531 | 4.232 | 0.403 | 450 |
| Average | 1.595 | 2.668 | 0.254 | 450 |
| 6/2/2021 22:17:06.354 | 2.205 | 3.698 | 0.353 | 500 |
| 6/2/2021 22:17:08.045 | 5.416 | 9.100 | 0.872 | 500 |
| Average | 3.810 | 6.399 | 0.612 | 500 |

| | | | | |
|---|---|---|---|---|
| 6/2/2021 22:17:19.993 | 3.285 | 5.526 | 0.530 | 550 |
| 6/2/2021 22:17:20.225 | 7.795 | 13.127 | 1.260 | 550 |
| Average | 5.540 | 9.326 | 0.895 | 550 |
| 6/2/2021 22:17:33.783 | 1.969 | 3.294 | 0.313 | 600 |
| 6/2/2021 22:17:35.542 | 2.078 | 3.500 | 0.327 | 600 |
| Average | 2.024 | 3.397 | 0.320 | 600 |
| 6/2/2021 22:17:48.205 | 3.076 | 5.147 | 0.490 | 650 |
| 6/2/2021 22:17:50.122 | 2.289 | 3.858 | 0.360 | 650 |
| Average | 2.683 | 4.503 | 0.425 | 650 |
| 6/2/2021 22:18:01.987 | 1.134 | 1.904 | 0.182 | 700 |
| 6/2/2021 22:18:02.565 | 1.471 | 2.464 | 0.235 | 700 |
| 6/2/2021 22:18:04.636 | 1.922 | 3.218 | 0.307 | 700 |
| 6/2/2021 22:18:05.557 | 1.754 | 2.940 | 0.280 | 700 |
| 6/2/2021 22:18:07.248 | 1.269 | 2.124 | 0.202 | 700 |
| 6/2/2021 22:18:08.289 | 1.412 | 2.362 | 0.225 | 700 |
| 6/2/2021 22:18:10.752 | 1.706 | 2.873 | 0.268 | 700 |
| Average | 1.524 | 2.555 | 0.243 | 700 |
| 6/2/2021 22:18:23.083 | 1.139 | 1.915 | 0.184 | 650 |
| 6/2/2021 22:18:24.744 | 1.539 | 2.578 | 0.245 | 650 |
| Average | 1.339 | 2.247 | 0.215 | 650 |
| 6/2/2021 22:18:36.038 | 3.766 | 6.301 | 0.599 | 600 |
| 6/2/2021 22:18:38.648 | 3.119 | 5.252 | 0.504 | 600 |
| 6/2/2021 22:18:39.756 | 0.632 | 1.060 | 0.101 | 600 |
| Average | 2.505 | 4.204 | 0.401 | 600 |
| 6/2/2021 22:18:52.759 | 6.530 | 10.984 | 1.053 | 550 |
| 6/2/2021 22:18:54.868 | 4.012 | 6.730 | 0.639 | 550 |
| Average | 5.271 | 8.857 | 0.846 | 550 |
| 6/2/2021 22:19:06.128 | 0.578 | 0.971 | 0.093 | 500 |
| 6/2/2021 22:19:07.897 | 0.145 | 0.244 | 0.023 | 500 |
| Average | 0.362 | 0.608 | 0.058 | 500 |
| 6/2/2021 22:19:20.178 | 1.017 | 1.713 | 0.164 | 450 |
| 6/2/2021 22:19:22.185 | 1.283 | 2.162 | 0.208 | 450 |
| Average | 1.150 | 1.938 | 0.186 | 450 |
| 6/2/2021 22:19:34.347 | 3.609 | 6.055 | 0.578 | 400 |
| 6/2/2021 22:19:36.887 | 3.208 | 5.410 | 0.520 | 400 |
| Average | 3.409 | 5.733 | 0.549 | 400 |
| 6/2/2021 22:19:48.613 | 20.971 | 35.413 | 3.411 | 350 |
| 6/2/2021 22:19:50.475 | 22.932 | 38.554 | 3.693 | 350 |
| Average | 21.952 | 36.984 | 3.552 | 350 |
| 6/2/2021 22:20:18.077 | 20.113 | 33.796 | 3.233 | 300 |
| 6/2/2021 22:20:20.648 | 22.501 | 37.700 | 3.593 | 300 |
| Average | 21.307 | 35.748 | 3.413 | 300 |
| 6/2/2021 22:20:33.296 | 44.420 | 74.607 | 7.138 | 250 |
| 6/2/2021 22:20:34.056 | 44.326 | 74.268 | 7.081 | 250 |

| | | | | |
|---|---|---|---|---|
| Average | 44.373 | 74.438 | 7.110 | 250 |
| 6/2/2021 22:20:46.231 | 44.696 | 75.425 | 7.260 | 200 |
| 6/2/2021 22:20:48.048 | 44.731 | 75.496 | 7.068 | 200 |
| Average | 44.713 | 75.460 | 7.164 | 200 |
| 6/2/2021 22:20:57.664 | 45.006 | 75.451 | 7.208 | 150 |
| Average | 45.006 | 75.451 | 7.208 | 150 |
| 6/2/2021 22:21:05.235 | 38.065 | 64.133 | 6.160 | 100 |
| 6/2/2021 22:21:07.643 | 45.417 | 76.131 | 7.262 | 100 |
| Average | 41.741 | 70.132 | 6.711 | 100 |
| 6/2/2021 22:21:15.931 | 45.199 | 76.279 | 7.349 | 50 |
| 6/2/2021 22:21:17.037 | 45.387 | 76.167 | 7.275 | 50 |
| Average | 45.29 | 76.22 | 7.31 | 50 |

| **Test 3** | | | | |
|---|---|---|---|---|
| **Date/Time** | **Frequency (MHz)** | | | **Distance (ft)** |
| | 462.544 | 462.553 | 462.563 | 50 |
| 6/2/2021 22:33:33.486 | 44.403 | 76.893 | 7.666 | 50 |
| 6/2/2021 22:33:35.912 | 44.396 | 76.904 | 7.667 | 50 |
| 6/2/2021 22:33:36.938 | 44.544 | 76.786 | 7.612 | 50 |
| Average | 44.448 | 76.861 | 7.648 | 50 |
| 6/2/2021 22:33:45.381 | 44.360 | 76.316 | 7.541 | 100 |
| 6/2/2021 22:33:47.717 | 44.323 | 76.385 | 7.563 | 100 |
| Average | 44.342 | 76.351 | 7.552 | 100 |
| 6/2/2021 22:33:55.902 | 43.720 | 75.229 | 7.438 | 150 |
| 6/2/2021 22:33:57.172 | 43.736 | 75.286 | 7.445 | 150 |
| Average | 43.728 | 75.258 | 7.442 | 150 |
| 6/2/2021 22:34:09.072 | 43.263 | 74.498 | 7.369 | 200 |
| 6/2/2021 22:34:11.858 | 42.858 | 74.270 | 7.211 | 200 |
| Average | 43.061 | 74.384 | 7.290 | 200 |
| 6/2/2021 22:34:21.098 | 41.217 | 71.194 | 7.069 | 250 |
| Average | 41.217 | 71.194 | 7.069 | 250 |
| 6/2/2021 22:34:30.278 | 10.047 | 17.396 | 1.734 | 300 |
| 6/2/2021 22:34:31.246 | 15.411 | 26.696 | 2.662 | 300 |
| Average | 12.729 | 22.046 | 2.198 | 300 |
| 6/2/2021 22:34:40.967 | 6.257 | 10.767 | 1.064 | 350 |
| 6/2/2021 22:34:42.338 | 11.237 | 19.337 | 1.911 | 350 |
| Average | 8.747 | 15.052 | 1.488 | 350 |
| 6/2/2021 22:34:52.352 | 3.863 | 6.690 | 0.667 | 400 |
| 6/2/2021 22:34:53.525 | 2.948 | 5.111 | 0.496 | 400 |
| Average | 3.406 | 5.901 | 0.582 | 400 |
| 6/2/2021 22:35:03.333 | 4.827 | 8.336 | 0.828 | 450 |
| 6/2/2021 22:35:05.245 | 4.111 | 7.108 | 0.707 | 450 |
| Average | 4.469 | 7.722 | 0.767 | 450 |
| 6/2/2021 22:35:17.946 | 1.996 | 3.461 | 0.345 | 500 |

| | | | | |
|---|---|---|---|---|
| 6/2/2021 22:35:18.743 | 2.189 | 3.795 | 0.368 | 500 |
| Average | 2.092 | 3.628 | 0.357 | 500 |
| 6/2/2021 22:35:30.784 | 8.045 | 13.881 | 1.377 | 550 |
| 6/2/2021 22:35:32.649 | 9.402 | 16.307 | 1.630 | 550 |
| Average | 8.723 | 15.094 | 1.504 | 550 |
| 6/2/2021 22:35:43.299 | 0.822 | 1.425 | 0.142 | 600 |
| 6/2/2021 22:35:45.149 | 0.718 | 1.246 | 0.121 | 600 |
| Average | 0.770 | 1.336 | 0.132 | 600 |
| 6/2/2021 22:35:56.045 | 1.847 | 3.190 | 0.317 | 650 |
| 6/2/2021 22:35:58.025 | 2.593 | 4.486 | 0.447 | 650 |
| 6/2/2021 22:36:00.364 | 2.493 | 4.296 | 0.425 | 650 |
| Average | 2.311 | 3.991 | 0.396 | 650 |
| 6/2/2021 22:36:11.077 | 1.427 | 2.467 | 0.245 | 700 |
| 6/2/2021 22:36:12.461 | 0.745 | 1.290 | 0.129 | 700 |
| 6/2/2021 22:36:14.505 | 0.786 | 1.353 | 0.134 | 700 |
| 6/2/2021 22:36:16.588 | 0.930 | 1.613 | 0.161 | 700 |
| 6/2/2021 22:36:17.572 | 0.869 | 1.502 | 0.149 | 700 |
| Average | 0.952 | 1.645 | 0.164 | 700 |
| 6/2/2021 22:36:29.385 | 2.400 | 4.155 | 0.414 | 650 |
| 6/2/2021 22:36:31.418 | 1.397 | 2.410 | 0.239 | 650 |
| Average | 1.899 | 3.283 | 0.327 | 650 |
| 6/2/2021 22:36:44.006 | 1.546 | 2.669 | 0.265 | 600 |
| 6/2/2021 22:36:46.899 | 1.095 | 1.888 | 0.187 | 600 |
| 6/2/2021 22:36:48.605 | 0.741 | 1.287 | 0.125 | 600 |
| Average | 1.128 | 1.948 | 0.192 | 600 |
| 6/2/2021 22:36:59.193 | 2.136 | 3.681 | 0.365 | 550 |
| 6/2/2021 22:37:01.136 | 0.674 | 1.165 | 0.116 | 550 |
| Average | 1.405 | 2.423 | 0.240 | 550 |
| 6/2/2021 22:37:12.933 | 3.733 | 6.436 | 0.638 | 500 |
| 6/2/2021 22:37:14.294 | 1.558 | 2.699 | 0.269 | 500 |
| Average | 2.646 | 4.567 | 0.453 | 500 |
| 6/2/2021 22:37:43.173 | 6.349 | 10.968 | 1.090 | 450 |
| 6/2/2021 22:37:44.446 | 6.241 | 10.739 | 1.062 | 450 |
| Average | 6.295 | 10.854 | 1.076 | 450 |
| 6/2/2021 22:37:56.371 | 14.378 | 24.709 | 2.439 | 450 |
| 6/2/2021 22:37:58.985 | 9.589 | 16.586 | 1.650 | 400 |
| 6/2/2021 22:37:59.337 | 5.413 | 9.314 | 0.920 | 400 |
| Average | 9.793 | 16.870 | 1.670 | 400 |
| 6/2/2021 22:38:11.838 | 12.420 | 21.480 | 2.140 | 350 |
| 6/2/2021 22:38:13.529 | 11.931 | 20.539 | 2.032 | 350 |
| Average | 12.176 | 21.009 | 2.086 | 350 |
| 6/2/2021 22:38:24.793 | 9.926 | 17.144 | 1.703 | 300 |
| 6/2/2021 22:38:26.575 | 6.828 | 11.749 | 1.161 | 300 |
| Average | 8.377 | 14.446 | 1.432 | 300 |

| | | | | |
|---|---|---|---|---|
| 6/2/2021 22:38:37.901 | 43.250 | 74.368 | 7.342 | 300 |
| 6/2/2021 22:38:39.886 | 42.679 | 73.832 | 7.359 | 250 |
| Average | 42.964 | 74.100 | 7.351 | 250 |
| 6/2/2021 22:38:52.599 | 43.970 | 76.130 | 7.591 | 200 |
| 6/2/2021 22:38:53.936 | 44.155 | 76.052 | 7.524 | 200 |
| Average | 44.063 | 76.091 | 7.557 | 200 |
| 6/2/2021 22:39:06.171 | 43.985 | 76.150 | 7.597 | 150 |
| 6/2/2021 22:39:08.221 | 43.890 | 75.945 | 7.565 | 150 |
| 6/2/2021 22:39:10.075 | 44.093 | 76.320 | 7.430 | 150 |
| Average | 43.989 | 76.138 | 7.530 | 150 |
| 6/2/2021 22:39:22.582 | 44.490 | 76.738 | 7.606 | 100 |
| 6/2/2021 22:39:23.845 | 44.357 | 76.829 | 7.670 | 100 |
| Average | 44.424 | 76.783 | 7.638 | 100 |
| 6/2/2021 22:39:37.039 | 44.511 | 76.781 | 7.613 | 50 |
| Average | 44.511 | 76.781 | 7.613 | 50 |

| **Combined Test Average Values** | | | |
|---|---|---|---|
| 462.544 | 462.553 | 462.563 | Distance (ft) |
| 45.48 | 76.15 | 7.25 | 50 |
| 45.72 | 75.92 | 7.15 | 50 |
| 45.397 | 76.181 | 7.281 | 50 |
| 45.293 | 76.223 | 7.312 | 50 |
| 44.448 | 76.861 | 7.648 | 50 |
| 44.511 | 76.781 | 7.613 | 50 |
| **45.141** | **76.353** | **7.376** | **Average** |
| 45.24 | 75.87 | 7.24 | 100 |
| 45.73 | 75.93 | 7.16 | 100 |
| 45.377 | 76.193 | 7.284 | 100 |
| 41.741 | 70.132 | 6.711 | 100 |
| 44.342 | 76.351 | 7.552 | 100 |
| 44.424 | 76.783 | 7.638 | 100 |
| **44.475** | **75.210** | **7.264** | **Average** |
| 44.57 | 74.44 | 7.07 | 150 |
| 45.44 | 75.61 | 7.14 | 150 |
| 45.144 | 75.808 | 7.183 | 150 |
| 45.006 | 75.451 | 7.208 | 150 |
| 43.728 | 75.258 | 7.442 | 150 |
| 43.989 | 76.138 | 7.530 | 150 |
| **44.646** | **75.451** | **7.261** | **Average** |
| 44.16 | 73.87 | 7.03 | 200 |
| 44.51 | 74.16 | 6.95 | 200 |
| 44.518 | 74.610 | 7.114 | 200 |
| 44.713 | 75.460 | 7.164 | 200 |
| 43.061 | 74.384 | 7.290 | 200 |

| | | | |
|---|---|---|---|
| 44.063 | 76.091 | 7.557 | 200 |
| **44.171** | **74.762** | **7.184** | **Average** |
| 41.67 | 69.64 | 6.62 | 250 |
| 44.41 | 74.00 | 7.01 | 250 |
| 43.780 | 73.514 | 6.963 | 250 |
| 44.373 | 74.438 | 7.110 | 250 |
| 41.217 | 71.194 | 7.069 | 250 |
| 42.964 | 74.100 | 7.351 | 250 |
| **43.069** | **72.814** | **7.020** | **Average** |
| 3.26 | 5.43 | 0.51 | 300 |
| 28.59 | 47.52 | 4.48 | 300 |
| 19.314 | 32.415 | 3.101 | 300 |
| 21.307 | 35.748 | 3.413 | 300 |
| 12.729 | 22.046 | 2.198 | 300 |
| 43.250 | 74.368 | 7.342 | 300 |
| **21.407** | **36.255** | **3.509** | **Average** |
| 29.96 | 49.86 | 4.71 | 350 |
| 19.57 | 32.66 | 3.10 | 350 |
| 14.294 | 23.926 | 2.279 | 350 |
| 21.952 | 36.984 | 3.552 | 350 |
| 8.747 | 15.052 | 1.488 | 350 |
| 12.176 | 21.009 | 2.086 | 350 |
| **17.783** | **29.916** | **2.869** | **Average** |
| 2.08 | 3.48 | 0.33 | 400 |
| 7.96 | 13.25 | 1.25 | 400 |
| 5.040 | 8.455 | 0.807 | 400 |
| 3.409 | 5.733 | 0.549 | 400 |
| 3.406 | 5.901 | 0.582 | 400 |
| 9.793 | 16.870 | 1.670 | 400 |
| **5.282** | **8.948** | **0.865** | **Average** |
| 0.86 | 1.43 | 0.14 | 450 |
| 14.39 | 23.99 | 2.27 | 450 |
| 1.595 | 2.668 | 0.254 | 450 |
| 1.150 | 1.938 | 0.186 | 450 |
| 4.469 | 7.722 | 0.767 | 450 |
| 14.378 | 24.709 | 2.439 | 450 |
| **6.140** | **10.410** | **1.009** | **Average** |
| 0.11 | 0.19 | 0.02 | 500 |
| 5.81 | 9.68 | 0.91 | 500 |
| 3.810 | 6.399 | 0.612 | 500 |
| 0.362 | 0.608 | 0.058 | 500 |
| 2.092 | 3.628 | 0.357 | 500 |
| 2.646 | 4.567 | 0.453 | 500 |
| **2.473** | **4.178** | **0.402** | **Average** |

| | | | |
|---|---|---|---|
| 2.47 | 4.14 | 0.39 | 550 |
| 0.12 | 0.21 | 0.02 | 550 |
| 5.540 | 9.326 | 0.895 | 550 |
| 5.271 | 8.857 | 0.846 | 550 |
| 8.723 | 15.094 | 1.504 | 550 |
| 1.405 | 2.423 | 0.240 | 550 |
| **3.923** | **6.674** | **0.650** | **Average** |
| 1.19 | 1.98 | 0.19 | 600 |
| 2.20 | 3.68 | 0.35 | 600 |
| 2.024 | 3.397 | 0.320 | 600 |
| 2.505 | 4.204 | 0.401 | 600 |
| 0.770 | 1.336 | 0.132 | 600 |
| 1.128 | 1.948 | 0.192 | 600 |
| **1.637** | **2.758** | **0.264** | **Average** |
| 3.49 | 5.82 | 0.55 | 650 |
| 0.49 | 0.81 | 0.08 | 650 |
| 2.683 | 4.503 | 0.425 | 650 |
| 1.339 | 2.247 | 0.215 | 650 |
| 2.311 | 3.991 | 0.396 | 650 |
| 1.899 | 3.283 | 0.327 | 650 |
| **2.035** | **3.443** | **0.332** | **Average** |
| 0.42 | 0.69 | 0.07 | 700 |
| 1.524 | 2.555 | 0.243 | 700 |
| 0.952 | 1.645 | 0.164 | 700 |
| **0.96** | **1.63** | **0.16** | **Average** |

THIS PAGE INTENTIONALLY LEFT BLANK

# APPENDIX F. EFFECTIVENESS VALUE EQUATIONS

SC of (1) SDR sensor:

$$SC = 2.24 * .95 * \Big(1 + \big((1 - 1) * 1.1\big)\Big) = 2.1$$

SC of (81) SDR sensors:

$$SC = 2.24 * .95 * \Big(1 + \big((81 - 1) * 1.1\big)\Big) = 189.3$$

SC of (47) SDR sensors:

$$SC = 2.24 * .95 * \Big(1 + \big((55 - 1) * 1.1\big)\Big) = 109.8$$

SC of (1) MT8221B spectrum analyzer:

$$SC = 108 * .95 * \Big(1 + \big((1 - 1) * 1.1\big)\Big) = 108$$

SC of (3) MT8221B spectrum analyzer:

$$SC = 108 * .95 * \Big(1 + \big((3 - 1) * 1.1\big)\Big) = 345.6$$

THIS PAGE INTENTIONALLY LEFT BLANK

# APPENDIX G. COST-EFFECTIVENESS VALUES OVER 12 MONTHS

| SC | | | Cost | | | CE Ratio | | |
|---|---|---|---|---|---|---|---|---|
| The | SC(COA) | SC(Alt) | Month | Cost (COA) | Cost (Alt) | Month | COA | Alt |
| 1 | 136296 | 248832 | 1 | 20295.36 | 37392.6 | 1 | 0.148906 | 0.150272 |
| 2 | 272592 | 497664 | 2 | 20295.36 | 52435.2 | 2 | 0.074453 | 0.105363 |
| 3 | 408888 | 746496 | 3 | 20295.36 | 67477.8 | 3 | 0.049635 | 0.090393 |
| 4 | 545184 | 995328 | 4 | 20295.36 | 82520.4 | 4 | 0.037227 | 0.082908 |
| 5 | 681480 | 1244160 | 5 | 20295.36 | 97563 | 5 | 0.029781 | 0.078417 |
| 6 | 817776 | 1492992 | 6 | 20295.36 | 112605.6 | 6 | 0.024818 | 0.075423 |
| 7 | 954072 | 1741824 | 7 | 20295.36 | 127648.2 | 7 | 0.021272 | 0.073284 |
| 8 | 1090368 | 1990656 | 8 | 20295.36 | 142690.8 | 8 | 0.018613 | 0.07168 |
| 9 | 1226664 | 2239488 | 9 | 20295.36 | 157733.4 | 9 | 0.016545 | 0.070433 |
| 10 | 1362960 | 2488320 | 10 | 20295.36 | 172776 | 10 | 0.014891 | 0.069435 |
| 11 | 1499256 | 2737152 | 11 | 20295.36 | 187818.6 | 11 | 0.013537 | 0.068618 |
| 12 | 1635552 | 2985984 | 12 | 20295.36 | 202861.2 | 12 | 0.012409 | 0.067938 |

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF REFERENCES


Alani, M. (2014). *Guide to OSI and TCP/IP Models* (1st ed. 2014.). https://doi.org/10.1007/978-3-319-05152-9

Amazon.com (n.d.-a). *RTL-SDR Blog R820T2*. Retrieved from https://www.amazon.com.

Amazon.com. (n.d.-b). *Nooelec NESDR smart v4 SDR*. Retrieved from https://www.amazon.com.

Amazon.com. (n.d.-c). *HackRF one software defined radio*. Retrieved from https://www.amazon.com.

Amazon.com. (n.d.-d). *Anritsu MT8221B*. Retrieved from https://www.amazon.com.

Anritsu Company. (2013). *Technical data sheet BTS master*. Retrieved from https://dl.cdn-anritsu.com/en-us/test-measurement/files/Brochures-Datasheets-Catalogs/datasheet/11410-00442R.pdf

Anritsu Company. (2019). *Understanding key real-time spectrum analyzer specifications*. Retrieved from https://dl.cdn-anritsu.com/en-us/test-measurement/files/Technical-Notes/White-Paper/11410-01138B.pdf

Boardman, A., Greenberg, D., Vining, A, Weimer, D. (2018). *Cost-benefit analysis: Concepts and practice*. [Kindle version]. Cambridge University Press

Cellini, S. & Kee, J. (2015). Cost-effectiveness and cost-benefit analysis. In K. Newcomer, H. Hatry and J. Wholey (Eds.), *Handbook of Practical Program Evaluation* (4th ed., pp. 636–672). Jossey-Bass.

Chen, K., & Prasad, R. (2009). *Cognitive radio networks*. https://doi.org/10.1002/9780470742020

Defense Advanced Research Projects Agency. (2016*). Ushering in a new generation of low-cost, networked, nuclear-radiation detectors*. Retrieved from https://www.darpa.mil/news-events/2016-08-23

Department of Defense. (2019). *Digital modernization strategy*. Washington, DC: Government Printing Office. Retrieved from https://media.defense.gov/2019/Jul/12/2002156622/-1/-1/1/dod-digital-modernization-strategy-2019.pdf

Department of Defense. (2021). *Military basic pay table*. Retrieved from https://militarypay.defense.gov/

Ettus Research. (2021). *USRP B205mini-i*. Retrieved from https://www.ettus.com/all-products/usrp-b205mini-i/

Federal Communications Commission. (2017a). *General Mobile Radio Service*. Retrieved on April 14, 2021 from https://www.fcc.gov/general-mobile-radio-service-gmrs

Federal Communications Commission. (2017b). *Multi-Use Radio Service*. Retrieved on April 14th, 2021 from https://www.fcc.gov/wireless/bureau-divisions/mobility-division/multi-use-radio-service-murs

Ferguson Dale. (2020). Signature management. *Marine Corps Gazette*, 104(9), 53–53.

Freeman, R. (2007). *Radio system design for telecommunications* (3rd ed.). https://doi.org/10.1002/0470050446

GNU Radio. (2021). *About GNU radio*. https://www.gnuradio.org/about/

Gocke, M. (2018). *Identification and mapping of cellular telephones to defeat radio-controlled IEDs* [Master's thesis, Naval Postgraduate School]. NPS Archive: Calhoun. http://hdl.handle.net/10945/59666

Grayver, E. (2013). *Implementing software defined radio* (1st ed. 2013.). https://doi.org/10.1007/978-14419-9332-8

Haslett, C. (2008). *Essentials of radio wave propagation*. Cambridge: Cambridge University Press.

Hattersly, L. (2020). *Raspberry Pi 4 vs Raspberry Pi 3B+*. Retrieved from https://magpi.raspberrypi.org.

Headquarters United States Marine Corps. (2015). *Organization of the United States Marine Corps* (MCRP 1–10.1). Washington, DC.

Headquarters United States Marine Corps. (2016). *Marine Corps operating concept: How an expeditionary force operates in the 21st century*. Washington, DC.

Headquarters United States Marine Corps. (2019). *Commandant's planning guidance*. Washington, DC.

Keen, K. (2015, December 31). *Rtl_power*. https://github.com/keenerd/rtl-sdr/blob/master/src/rtl_power.c

Keen, K. (2017, March 26). *Heatmap.py*. https://github.com/keenerd/rtl-sdr-misc/blob/master/heatmap/heatmap.py

Koch, W. (2013). *Tracking and sensor data fusion: Methodological framework and selected applications* (2014th ed.). https://doi.org/10.1007/978-3-642-39271-9

Larsen, I. (2007). *Design and implementation of a mobile phone locator using software defined radio* [Master's thesis, Naval Postgraduate School]. NPS Archive: Calhoun. http://hdl.handle.net/10945/3256

Lichtman, M. (2021). *PySDR: A Guide to SDR and DSP using Python*. Retrieved from https://pysdr.org.

National Aeronautics and Space Administration, Science Mission Directorate. (2010). *Introduction to the Electromagnetic Spectrum*. Retrieved on May 5th, 2021, from NASA Science website: http://science.nasa.gov/ems/01_intro

National Instruments. (2016). *Understanding FFTs and Windowing*. Retrieved from https://download.ni.com/evaluation/pxi

Nooelec (n.d.). *Nooelec NESDR SMART v4 SDR*. Retrieved from https://www.nooelec.com/store/sdr/sdr-receivers/nesdr/nesdr-smart-sdr.html

Mkrzysztofowicz. (2019, April 03). Re: 4G Hat [Comment on the blog post Re: 4G Hat]. https://www.raspberrypi.org/forums/viewtopic.php?t=224355

Mead, J. (2015, March 19). *Install.md*. https://gist.github.com/floehopper/

Munson, J. (2018). *Open-source tools for covert communications in a contested information environment* [Master's thesis, Naval Postgraduate School]. NPS Archive: Restricted.

Pandeya, N, & Temple, N. (2016). *About USRP Bandwidths and Sampling Rates, Application Note 177*. Retrieved from https://kb.ettus.com/

PiTunnel. (2021). *Service Features*. Retrieved April 8, 2021 from https://www.pitunnel.com/#features

Raspberry Pi Foundation. (n.d.). *Documentation: Raspberry Pi OS*. Retrieved May 14, 2021 from https://www.raspberrypi.org/documentation/raspbian/

RTL-SDR Blog. (2018, 27 Jan). *Re: Multiple false signals*. https://www.rtl-sdr.com/forum/viewtopic.php?t=2092

Smith, S. (1997). *The Scientist and Engineers Guide to Digital Signal Processing*. California Technical Publishing. San Diego, CA.

Smyth, T. (2019). *Fundamentals of Digital Audio*. University of California, San Diego. Retrieved from http://musicweb.ucsd.edu/

Waveshare. (2020). *SIM7600CE-T/E-H/A-H/SA-H/G-H 4G Modules*. Retrieved May 14, 2021 from https://www.waveshare.com/wiki/SIM7600CE_4G_HAT

THIS PAGE INTENTIONALLY LEFT BLANK

# INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California