Summer 2016

Analysis and Implementation of Communications Systems for Small Satellite Missions

Jason S. Harris
Old Dominion University, sharr113@odu.edu

ANALYSIS AND IMPLEMENTATION OF COMMUNICATIONS SYSTEMS FOR SMALL SATELLITE MISSIONS

by

Jason S. Harris
B.S. December 2014, Old Dominion University

A Thesis Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

ELECTRICAL AND COMPUTER ENGINEERING

OLD DOMINION UNIVERSITY
August 2016

Approved by:

Dimitrie C. Popescu (Director)

Lee A. Belfore (Member)

Robert Ash (Member)

ABSTRACT

ANALYSIS AND IMPLEMENTATION OF COMMUNICATIONS SYSTEMS FOR SMALL SATELLITE MISSIONS

Jason S. Harris
Old Dominion University, 2016
Director: Dimitrie C. Popescu

Nano satellites are becoming more and more popular space platforms due to their relatively low cost. Constellations of many of these small satellites are being launched for scientific and research purposes. This thesis has examined implementing a communications system for small satellites that can be used to maintain constant contact with satellites as they orbit the Earth. It analyzes the various components of a small satellite and how they integrate. It then discusses the different abstraction layers that will be required in order to support the same software architecture across various types of hardware. An orbital analysis was performed to define the requirements for acquisition and loss of signal. Due to the ever increasing threat from space debris, a simulation using a high performance computing system to determine satellite threats was conducted. The thesis concludes with a communications analysis followed by a case study.

ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Dimitrie C. Popescu for his patience and guidance over the years that I have worked with him. Dr. Popescu first introduced me to software defined radios, and it has provided me with many valuable opportunities. Without his support, this thesis would have taken much longer to complete. I would also like to thank Dr. Robert Ash and Dr. Lee Belfore for their assistance throughout my academic career and for being on my thesis committee. Dr. Ash taught me that space is something that is within reach of the average person as long as they are willing to put their mind to it. Dr. Belfore has provided me with a wealth of knowledge that I have used when solving various problems that I have encountered throughout my academic and professional careers.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

Small satellites are becoming increasingly popular due to their relatively low cost and their ease of manufacture. Due to these characteristics, they are relatively disposable and are being used to perform research and collect data that was previously too expensive to consider. In order to be able to properly support these small satellite missions, a network of ground stations must be implemented in order to provide maximum communications coverage. This thesis analyzes the communications channels required in order to properly implement a network of ground stations.

Small satellites are generally defined as satellites with masses smaller than 500 kg. Within this group, there is a subgroup called nano-satellites. These are generally defined as satellites with masses between 1 and 50 kg. Within the nanosat group, there is a smaller group called Cube Satellites. Cube satellites or CubeSats are a subset of small satellites that have a modular structure with one unit (1U) implied by a cube with dimensions of 10 cm × 10 cm × 10 cm and mass of up to 1 kg [1, 2]. CubeSats can be built in various modular sizes such as 1U, 2U, 3U, and 6U, and they can be placed in orbit as secondary payloads at a fraction of the cost of a traditional satellite system, thus bringing space science experimentation within reach for universities. CubeSats are also part of the NASA Centennial Program and its associated Centennial, Challenges, through the Cube Quest Challenge that was issued in 2014 seeking to develop and test subsystems necessary to perform deep space exploration using small spacecraft. As part of Cube Quest, demonstration of deep-space communication capabilities of CubeSats is expected. The implementation of the communication sub-system for a CubeSat mission is a significant endeavor, since this is a critical component that lays the foundation for the types of experiments a planned mission can pursue and determines the amount of information that can be transmitted back to Earth for analysis and further processing.

CubeSat missions may consist of a single satellite launched and operated individually for basic science experiments, or they can include multiple CubeSats that are deployed in clusters to operate in swarm-like formations as is the case for the QB50 mission [3], whose launches were planned initially for late 2015 or early 2016 [4]. In the constellation case, the satellite collects data related to the particular science experiment, performs some basic processing on this data, and then transmits the data using a radio transceiver to a ground station where it is further processed and interpreted. In the latter case, the satellites establish also inter-satellite links [5] and set up a wireless network over which they share observed scientific data along with ancillary information (position, timing, etc.) which enables them to perform joint/distributed processing of the data. Thus, in this case CubeSats may require two or more radio transceivers, one for the ground station link and another one for the inter-satellite connections.

Recently, the use of software-defined radios (SDRs) has been proposed to implement the communications sub-system for CubeSats [6, 7, 8]. This approach has been prompted by the emergence of software defined electronics [9], which offers flexible implementations for modern telecommunication and measurement systems and enables reconfigurability of a given electronic system through software and programming. SDRs have been successfully used since the late 1990s to improve interoperability of the various commercial radio systems, and to reduce development and deployment costs [10, 11].

The supply chain for small satellite hardware is starting to increase with the growth of vendors such as *Clyde-Space* and *Pumpkin Incorporated*. Both of these vendors sell various CubeSat kits, along with a range of components for ease in fabrication. There are also launch options available from several vendors that are reasonably priced. A CubeSat kit and launch from Inter-orbital Systems is quoted as low as $12,500 [12].

1.1 SMALL SATELLITE MISSIONS

Companies such as *Planet Labs* and *Spire* currently operate constellations of small satellites that are capable of global coverage. Currently, *Planet Labs* has currently launched over a hundred satellites (although most have already been retired), [13] and the company's goal is to be able to image the entire Earth once every 24 hours. In order to do this, they intend to maintain 100 to 150 satellites in sun synchronous orbits. As the Earth rotates beneath those satellites, they will act like line scanners imaging everything below them.

Spire is a company primarily interested in data collection using small satellites [14]. They started by launching a constellation of satellites to track the movements of all major cargo ships over the oceans. Large ships carry an Automatic Identification System (AIS) transponder that broadcasts information about their identification and course to nearby ships. This is useful for enabling ships to avoid collisions as well as for goverments to be able to monitor ships off of their coasts. The problem with AIS is that it operates in the very high frequency (VHF) radio spectrum. Since VHF transmissions can only be received via line of sight, when a ship moves over the horizon from an AIS receiver it can no longer able to be monitored. Now, Spire's constellation of satellites is able to receive their signals from anywhere in the world and relay the information to interested parties.

Spire will soon be launching a new project called Stratos which is a constellation of satellites that will use GPS radio occultation in order to monitor characteristics of the Earth's atmosphere for weather prediction [15]. The satellites measure the characteristics of GPS signals that they receive from GPS satellites on the opposite side of the Earth. By measuring how the GPS signals refract, the temperature and humidity levels of the Earth's atmosphere can be calculated. Spire anticipates that their system will produce roughly ten times more weather data than all of the current publicly funded meteorological satellites currently in orbit.

There are many universities that either are already launching small satellites or are planning on launching them. One of the main projects that is currently in development is that of the QB50 program which aims to launch 50 satellites into low Earth orbit [16]. The main scientific goals of the project are to gather measurements in the lower thermosphere which is the part of the atmosphere that is understood the least.

There are other smaller scale missions planned for academic purposes such as the LAICE project which is a collaboration between Virginia Tech and the University of Illinois [17]. This particular project aims to obtain atmospheric data and understand how it affects the propagation of radio waves. Another group that builds satellites for education purposes is the Radio Amateur Satellite Corporation (AMSAT). AMSAT primarily designs satellites for amateur radio operators to communicate with each other. Currently, there are approximately 10 satellites in orbit that AMSAT is affiliated with although they are not always functional.

1.2 THESIS ORGANIZATION AND CONTRIBUTION

With the increase in small satellites, there needs to be a way to communicate with them effectively. There is currently no global public ground station implementation available for companies or universities to use. The goal of this thesis is to develop an architecture that can allow multiple ground stations to be distributed around the world and remain in contact with a satellite for the maximum amount of time. Previous work has been done in this area such as Zachary Leffke's thesis entitled "Distributed Ground Station Network for CubeSat Communications," but an actual implementation has not been discussed in sufficient detail [18].

In order to implement a system of this scale and complexity, many different systems need to be analyzed. This thesis begins by analyzing the components of a small satellite mission and how they fit together. The different components are introducedin terms of how they operate at a system level, along with some of their operating characteristics. The thesis then discusses programming languages and their characteristics, elaborating on why some programming languages may be better suited than others due to their performance characteristics and portability. Communications software and its performance is discussed along with the required software for controlling ground stations.

No discussion about satellite architecture is complete without a discussion of satellite characteristics and their orbits. There are several applications that can be used for predicting the orbits of satellites; however, there is not much documentation concerning how it is done. This thesis describes how the simplified perturbations models can be used to predict the orbits of satellites and then transform the coordinate system to compensate for the rotation of the Earth. With the increasing number of objects being launched into space, space debris (also known as "space junk") is beginning to be more and more of a problem. In order to objectively determine the seriousness of this problem, a simulation was run using a high performance computing cluster to determine how many objects are within 100 km of other objects orbiting in space and to attempt to predict collision probabilities between objects.

This thesis then analyzes the communications channel between a satellite and the ground station. Little work has been done prior to this thesis on how the communications channel changes as the satellite approaches and then departs from ground station contact. An analysis of the path loss and doppler effect for various frequencies during the satellite pass has been performed. An analysis of an inter-satellite communications system has been performed as well, and an equation model was developed to show how the path loss for different numbers of satellites in a constellation can be determined.

Some implementation aspects are then discussed. Software defined radio technology, discussed in section 6.1, is becoming more and more popular with satellite communications systems due to its flexibility. Due to the low power output from software defined radios, a discussion of power amplifiers and how they can be designed is presented.

In order to improve the performance of the communications system, modulation methods requiring a lower signal to noise ratio can be used. This thesis concludes with a network implementation that describes the physical and network level protocols that could be used as a basis for a small satellite communications system.

CHAPTER 2

COMPONENTS OF A SMALL SATELLITE MISSION

There are several major components that comprise a small satellite mission. They are the satellite, ground stations, software applications and the controller. In a typical setup, there is only one satellite and one ground station. In order to expand this to a network, at least one more ground station must be added as well as a controller and application packages.

A CubeSat platform capable of satisfying requirements for scientific measurements in low Earth orbit (LEO) should include all subsystems needed to support and power the science instruments as well as to transmit the collected experiment data to a ground station for further processing, analysis, and archiving. While the physical configuration of CubeSats depends on the actual science experiment, the main components of a CubeSat system are independent of the specific CubeSat science missions and are outlined in the block diagram shown in Figure 1.

Besides the science instruments, which are supposed to provide the data related to the observed parameters, a CubeSat system includes a power sub-system with solar panels and batteries to power up the CubeSat, the on-board computer performing data acquisition and processing, and the communications sub-system for providing the link with the mission ground station as well as links with other CubeSats that may be part of the mission.

When focusing on the design of the communication sub-system of a CubeSat, one should note first that because they are placed in low Earth orbit, CubeSats are within range of a ground station only for a limited time duration, which is usually of the order of a few minutes. For a successful CubeSat mission this duration must enable the information exchange between the CubeSat and the ground station, and determines the amount of information to be exchanged as well as the data rate and the parameters of the digital modulation scheme used for the satellite-to-ground communication link.

2.1 SATELLITE BUS

The purpose of the satellite bus is to provide a platform for the main science experiment payload. A proper satellite will provide enough power for the payload to operate for the

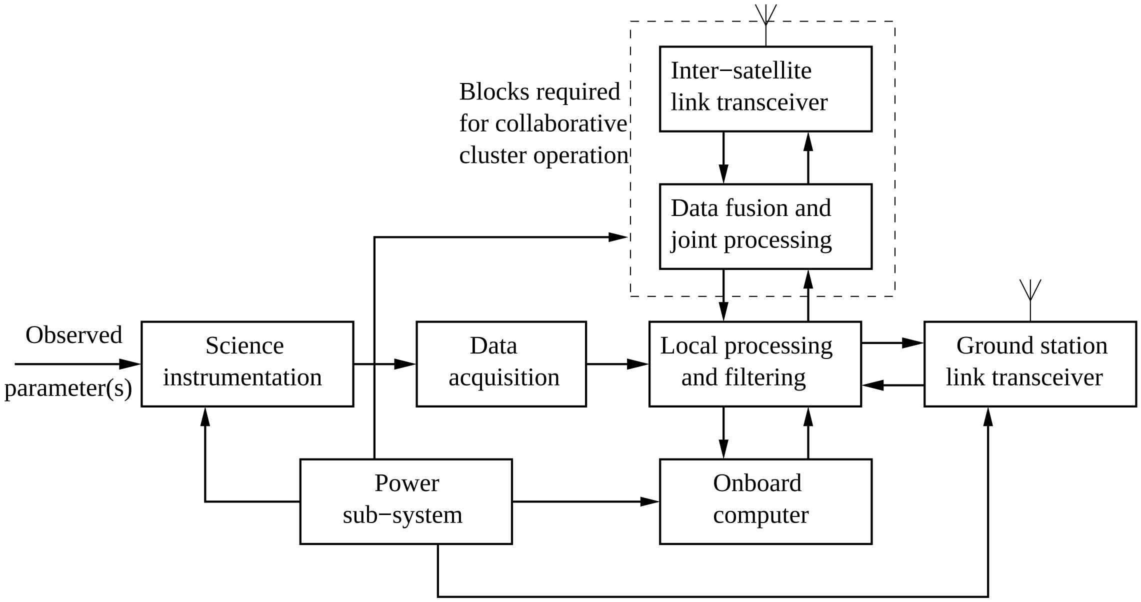

FIG. 1: Block diagram outlining the main components of a CubeSat system

entire duration of the mission. The satellite may generate power using solar cells, utilize batteries or use a combination of both. The satellite also provides a communications system for the payload. The communications system is primarily used to send and receive data related to the main experiment as well as sending status information related to the power subsystem.

The satellite bus also acts as a structural shell for rigidly-mounting internal components. Due to the vacuum of space, the satellite can only exchange thermal energy with its surroundings by radiation. In order to keep certain integrated circuits and other components cool, heat pipes and other techniques must be used [19]. Certain types of integrated circuits (such as memory) are susceptible to bit errors due to cosmic radiation [20]. The satellite must be designed to shield sensitive components as well as possible. In the event that a major failure of the control system occurs, a watchdog timer should be able to reset the control system.

Small satellites can use a wide range of digital processors or single-board computers as their control systems. The limiting factors that determines which type of control system can be used are primarily power consumption and size. Small satellites that require a full embedded system can use PC/104 based hardware which is small enough to fit into a 1U CubeSat [21]. It is possible to get PC/104 single board computers that contain the latest processors such as the Intel I7 [22]. Satellites that do not need as much processing power can use a smartphone or a basic microcontroller.

2.2 TRANSCEIVERS

Different types of radios can be used at the ground stations which serves as the systems for receiving and sending information to and from the satellite. Due to the limited space and power on a nano satellite, the ground station is generally designed to be much more powerful and with better equipment in order to minimize the path loss and increase the signal to noise ratio. The type of radio that most people are familiar with is the conventional radio. The other type of radio that may be used is a software defined radio. Each system has its advantages and disadvantages.

Conventional radios are capable of operating as standalone pieces of equipment and do not require special support components or accessories. Most of the signal processing, such as modulation and demodulation, is done using dedicated hardware which means that it is generally not changeable to support different features or modulation methods. Some conventional radios can be changed to operate at other frequencies through the use of add-on modules.

An image of an Icom IC-910 is shown in Figure 2. This is a standard conventional radio often used by Amateur Radio operators for communicating with satellites because it supports AM, FM, single sideband (SSB) and digital modulation methods. It is natively capable of transmitting on the 2 meter amateur radio band as well as the 70 cm amateur radio band. An aftermarket module can be added to allow the radio to operate on the 23 centimeter (1.2 GHz) band.

A typical conventional radio receiver uses a technique called super-heterodyne conversion. These systems use an intermediate frequency to which the received signal is initially converted. A demodulator tuned at the intermediate frequency then down converts the demodulated signal to baseband for reception and a modulator to convert the signal at baseband to the IF for transmission. The modulator and demodulator are implemented in hardware and are designed to either be fixed with respect to bandwidth, or to support several pre-defined bandwidths using mechanical switches. This means that if a user wishes to change modulation methods or bandwidths, they would have to change the physical hardware. A block diagram of a conventional radio is shown in Figure 3.

The latest development in radio technology is Software Defined Radio (SDR). This system performs all modulation and demodulation using software instead of hardware. Software defined radios also convert the baseband signal directly to the radio frequency which removes the intermediate frequency that conventional radios use. Software defined radios also natively utilize the in-phase (I) and quadrature (Q) components of a signal since they

FIG. 2: Icom IC-910 conventional radio that is primarily used by Amateur Radio Operators for Satellite Communications.

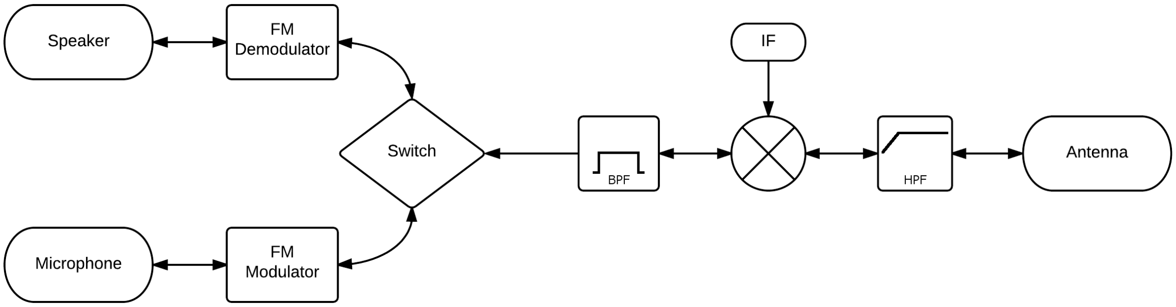

FIG. 3: Block diagram of a super-heterodyne transceiver. This type of circuit is used in conventional radios and implemented in hardware.

generally deal with digital communications systems although they can be used for analog communications as well. An example of a Software Defined Radio is the USRP B200 by Ettus Research as shown in Figure 4.

In a software defined radio, the received signal passes through a splitter and enters a low pass filter followed by a mixer. The in-phase component of the signal is extracted by mixing the signal directly with a local oscillator which converts it to a baseband signal. The in-phase component then passes through a low pass filter and into the analog to digital converter. The quadrature component is extracted by mixing the received signal with the local oscillator that has a 90 degree phase shift. The quadrature component at baseband then passes through a low pass filter and into an analog to digital converter. A block diagram

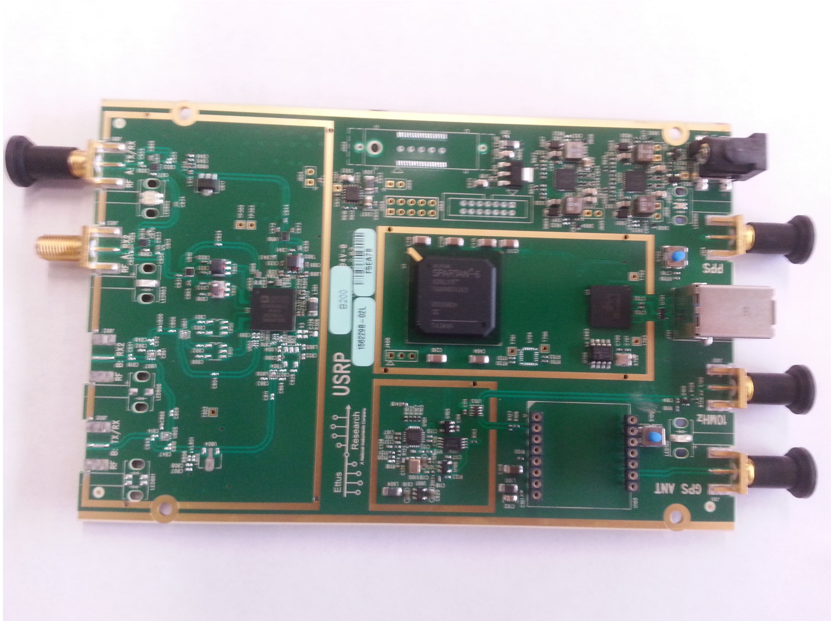

FIG. 4: USRP B200 software defined radio developed by Ettus Research.

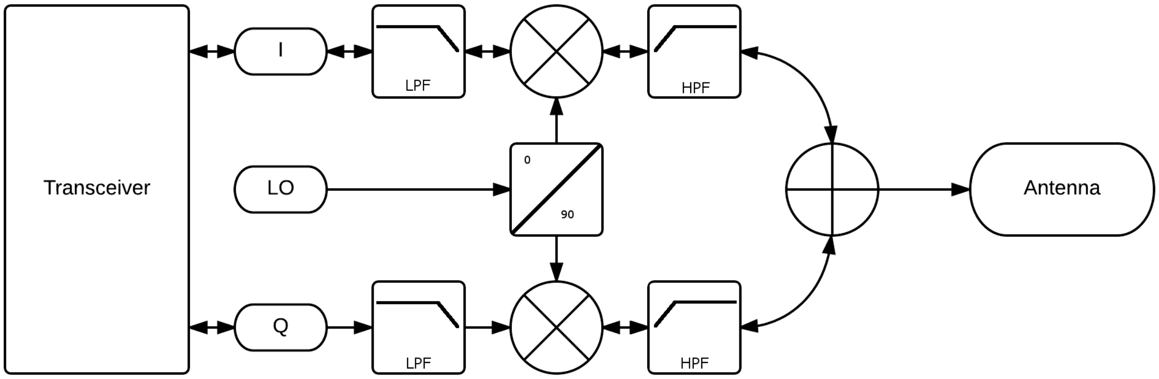

FIG. 5: Block diagram of a Software Defined Radio which uses direct conversion. This differs from Figure 3 in that it does not have an intermediate frequency stage.

of a standard software defined radio is shown in Figure 5. An example of an analog front end integrated circuit that could be used with this block diagram is the Analog Devices AD9860 [23] which supports two 64 million sample per second (MSPS) analog to digital converters and two 128 MSPS digital to analog converters. This device will be discussed in greater detail in the implementation chapter later in this thesis.

Most software defined radios use wide frequency range phase locked loops (PLLs) in order to have a large tuning range. It is common for most software defined radios to be able to operate over a range a several gigahertz [24, 25]. They are very versatile and some platforms such as the WARP board are used to prototype wireless standards [26]. This makes them very flexible and capable of supporting a wide range of satellite missions.

2.3 GROUND STATIONS

In order to implement a network, there must be more than one ground station. The goal of any network of ground stations is to enable a satellite to remain in contact with mission control for as long as possible. In order to do this the locations of ground stations are generally chosen where there is no existing coverage.

Each ground station will require at a minimum a computer, Internet connection, antenna and receiver. This setup allows a ground station to receive data from any satellite that is passing overhead and transmitting information. Since there is no transmitter involved, this rudimentary ground station setup could be placed anywhere on Earth and would not require licensing. An example of an application for this would be to receive data from weather satellites. If bidirectional communication is required, the receiver will have to be replaced with a transceiver that is capable of transmitting; those systems would need to be properly licensed with the appropriate governmental entity.

Most ground stations support different types of antennas in order to be able to cover different frequency bands. For reception, a wide band antenna such as a discone can be used, depending on the signal strength transmitted from the satellite. Since a discone antenna provides very little gain, only satellites that transmit at higher power levels can be received. The ground station at Old Dominion University has a Hustler DCX discone antenna which can cover 50 MHz to 950 MHz. For transmission, directional antennas should be used in order to increase the signal to noise ratio. The ground station at Old Dominion University has a beam antenna for 144-148 MHz and a beam antenna for 430 MHz to 450 MHz. These antennas can also be used to receive signals from satellites that have weak signal levels.

2.4 GROUND STATION CONTROLLER

The controller is the system that performs all scheduling, distribution of application packages, antenna control, status monitoring and receives all data from the ground stations. The term ”controller” is an abstract term that could refer to a single server or many servers depending on the number of satellites and the number of ground stations. The ground station controller will provide a unified interface to enable operators to orchestrate the overall satellite mission.

The controller will collect and process possibly terabytes of information daily. As such, it will need to be in a location where it has access to a very high bandwidth Internet connection. It can be anticipated that all radio recordings received from satellites will be sent back to the controller for later processing to potentially be used for future research. Since the system will be executing all of the calculations for the direction of the antennas as well as the scheduling, it will require very high performance processors. A perfect environment in which to operate the ground station controller would be a high performance computing cluster due to the massive amounts of storage and large amount of available processing power.

The ground station will generally be connected to the controller via an Internet connection. An analysis must be performed to be able to determine what kind of applications are feasible when there is a considerable geographic distance between the ground station and controller, causing a great deal of network latency.

If the system is primarily downloading or uploading data, then latency does not cause many issues. However, if there are many small messages that need to be exchanged between the satellite and controller, latency can become a significant problem. The latency between a computer at Old Dominion University and several other continents was determined and is shown in Table 1.

The time needed to transmit a message of length L and at a rate R is simply calculated as L/R. For example, if the data rate between the ground station and a satellite is 9600 baud, the amount of time needed to transmit an N bit message is $N/9600$. This calculation can be used to determine the efficiency of the communications system due to the round trip time (RTT) between the controller and the ground station. If the RTT is relatively high, packets sent between the controller and the ground station should be relatively large. In order to illustrate this, the size of the message that is equal to the round trip latency to each of the sites in Table 1 and causes the throughput to be 50% at a baud rate of 9600 was calculated. To calculate message size, the round trip time is simply multiplied by the data rate. The resulting message size would be about 320 bytes for South Africa, 324 bytes for

Location	Round Trip Latency (ms)
South Africa	267 ms
Australia	288 ms
Japan	195 ms
United Kingdom	96 ms

TABLE 1: Table of network latencies between a computer at Old Dominion University and several other locations on the Earth.

Australia, 115 bytes for the United Kingdom and 234 bytes for Japan. In order to maintain an efficiency greater than 50%, the packets need to be larger than those minimum values.

CHAPTER 3

GROUND STATION SOFTWARE APPLICATIONS

In general it can be assumed that each ground station will differ greatly in hardware makeup. In order to provide a homogeneous environment for data collection and communications software, the hardware at each ground station must operate in a robust manner. The primary software applications that need to be programmed are the communications software package and the antenna control software package. This section first discusses different programming languages and why different ones should be used. It then discusses the requirements for the communications software package and the antenna control application.

3.1 PROGRAMMING LANGUAGES

Different processor architectures may be employed at each ground station. Therefore, if an application is compiled for a particular instruction set, it may not be able to run on all of the ground stations. Presently, the majority of desktop computers utilize the x86 instruction set. However, there are a growing number of ARM instruction set based computers that are being used to replace x86 based desktops due to the former's higher power efficiency. There are two main techniques that can be used to write software that can run on any processor architecture without having to be rewritten/recompiled. The first technique that can be used is to write the software in a scripting language. Alternatively, a language that compiles to bytecode and executes on a virtual machine can be employed.

If a common scripting language is chosen, the interpreter could be installed on all of ground station computers. There are currently two major scripting languages used by the majority of communications and digital signal processing (DSP) software projects. These languages are MATLAB and Python. Both languages have their advantages and disadvantages. MATLAB is taught in many colleges with Electrical and Computer Engineering departments which means that most universities have students who are able to utilize it. MATLAB also automatically uses algorithms to maximize the performance of common mathematical functions which means that its performance for digital signal processing applications is relatively efficient. The downside is that it is proprietary, and a license must be purchased. MATLAB also does not maintain backwards compatibility between successive

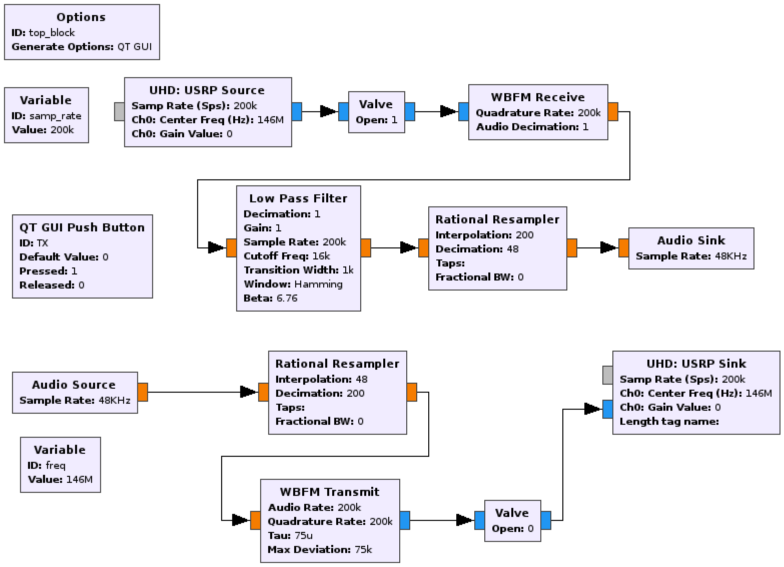

FIG. 6: Block diagram of a FM transceiver designed using GNU Radio which operates at 146 MHz.

versions. This means that the code for applications packages would need to be updated constantly. Python is an open source programming language currently gaining momentum for use with digital signal processing and communications code. Since it is open source it does not cost anything to install and use. It maintains backward compatibility for all code in the major version numbers (the current version of Python is version 3). The downside of Python is that it is not natively optimized for mathematical operations. This means that it would be more difficult to achieve performance requirements. In order to account for this, the popular software defined radio toolkit GNU Radio, which is written in Python, uses extensions written in the C programming language and assembly language in order to improve performance. GNU Radio allows a user to program a software defined radio using a graphical programming language similar to National Instruments LabView. It supports several different software defined radio platforms such as the USRP, RTL-SDR and HackRF platforms. A block diagram example of a FM transceiver using GNU Radio is shown in Figure 6.

Another technique that can be used is to write the applications packages in a language that can be compiled to run on a virtual machine. The most popular current languages meeting this specification are Java and C Sharp (C#). Currently, there are very few digital signal processing or communications software packages that have been developed with either language. The only mainstream software defined radio program written in either language is SDR# which is written in C# and is not open source. A major advantage of these languages is that they can execute much faster than scripting languages due to the use of just in time compilation (JIT). When JIT is used, the bytecode is converted into the instruction set that the processor uses natively and can achieve the performance of native applications written in C/C++ or Fortran. There is also a compiler for Java called GCJ which can compile straight to machine code [27].

3.2 COMMUNICATIONS SOFTWARE

The communications software will be the most time critical code in the entire system. The software will have to be able to receive signals from the satellite, process them and then send a response immediately in order to maximize the efficiency of the communications channel. If software defined radios are used, it is common for them to send and receive millions of samples per second to data converters (DACs and ADCs) which must be modulated and demodulated using software. If the receiver software is not fast enough, the receive buffer will fill and data will be lost. If the transmit buffer is not kept filled because the transmit code has not been optimized, jitter will result in the transmit signal. Since jitter in the transmit signal can act like an impulse in the time domain (and therefore a sinc function in the frequency domain), wide-band noise can be generated that could exceed the bandwidth of the communications channel and cause interference with other systems. To prevent any of these issues from occurring, the software will have to be heavily optimized.

Since most digital signal processing software applies similar operations to the signal samples, single instruction multiple data (SIMD) instructions can be used on the processor. These instructions use a single instruction on the processor to operate on multiple data values. For example, instead of using four square root instructions to find the square root of four different values, a SIMD instruction could be used on the processor to execute them in parallel. Most basic linear algebra subprograms (BLAS), such as LAPACK, make use of SIMD instructions and would be easy to implement in existing software [28]. There is also a library called VOLK that is specifically written for signal processing code [29]. VOLK was developed originally for use with GNURadio, but is now a separate package. If any new signal processing software is written, VOLK could be used as a starting point to make the code run as fast as possible.

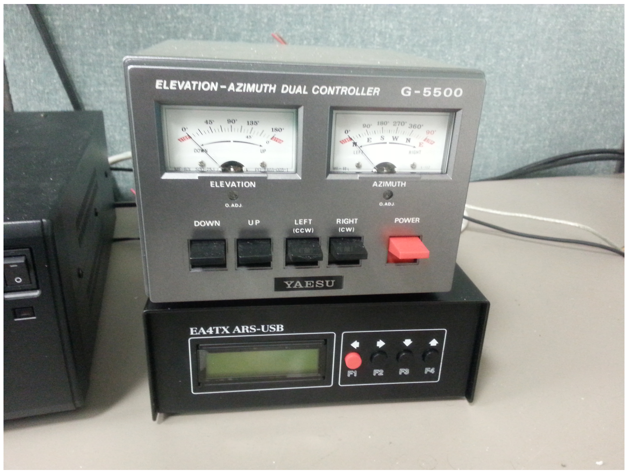

FIG. 7: Image of a Yaesu G-5500 rotator controller with a EA4TX computer interface. The G-5500 controls the motors on the rotator and the computer interface allows the computer to communicate with the controller.

3.3 ANTENNA CONTROL SOFTWARE

Three major components are required for a modern antenna setup. They are the rotator controller, the rotator and the computer interface. The rotator consists of the motors used to steer the antenna in appropriate directions. These systems often operate using alternating current due to the high voltage losses over long distances that occur when using direct current. It is common for the cables connecting the rotator to the rotator controller to be more than one hundred feet long. In order to command the rotator controller using a modern computer, an interface is needed. The computer interface is generally a separate piece of hardware that translates the relay logic used by the rotator controller into signals that are computer compatible inputs and outputs. This is generally done by using a virtual serial port over a Universal Serial Bus (USB) connection. For systems with only one ground station operated manually, a piece of open source software called GPredict is capable of tracking satellites and positioning the antennas [30]. Unfortunately, it cannot be controlled using another program for automatic scheduling and operation.

An image of the rotator controller and computer interface that is installed presently at Old Dominion University is shown in Figure 7. It is capable of controlling both the azimuth and elevation of all the antennas. Most rotator controllers operate using relay logic; thus, a computer interface box is required. The EA4TX computer interface, shown in Figure 7, allows a computer to employ commands sent and received over a USB interface to operate the rotator controller.

Each vendor's rotator controllers and computer interfaces support their own proprietary command set. The software for the antenna control would have to be written so that it is modular. This would be a perfect opportunity to use an object oriented programming language. A class could be instantiated in the software application that would represent the basic instructions which are to obtain or set the azimuth or elevation of the antennas. That class could then be inherited by specific classes written for each vendor in order to abstract the different control mechanisms.

CHAPTER 4

ORBITAL ANALYSIS AND PROPAGATION

Orbital mechanics must be utilized in order to enable realistic assessments of satellite-ground station performance. The primary pieces of information that need to be known is when a satellite will pass over a ground station, at what range, and for how long. This information can be used with the communications system analysis in order to determine if bandwidths are large enough to permit mission success.

4.1 ORBIT PREDICTION & DETERMINATION

Before an analysis of the satellite's trajectory can be performed, the orbital parameters must be known. In order to determine these parameters, the orbit of the satellite must be obtained using either radar or optical tracking. Once the orbital parameters are known for a reasonable period of time, a mathematical model can be used to propagate the satellite position so that it can be used for determining future satellite positions.

The parameters for the orbit of a satellite can be calculated using information from radar or optically using telescopes. The orbital plane of a satellite can be determined from only two discrete position vector measurements with respect to the Earth-Centered, Earth-Fixed coordinate system which will be described later in this section. If **EA** is a vector from the center of the Earth to position of a satellite at one instance in time and **EB** is a vector from the center of the Earth to a satellite at another instance of time, then the normal vector **N** can be found by calculating the cross product of **EA** and **EB**. The equation for the plane can then be defined as follows:

$$\boldsymbol{N_i}X + \boldsymbol{N_j}Y + \boldsymbol{N_k}Z = 0 \tag{1}$$

where X,Y and Z are a set of coordinates that, when substituted, result in the equation equaling zero. For clarity, a diagram is shown in Figure 8. The six elements required to specify the satellite orbit can be determined from three position vector measurements. This technique is called Angles-Only Orbit Determination and uses an iterative process [31].

The primary propagation models for satellites orbiting the Earth are simplified perturbation models. These models include the Simplified General Perturbations model (SGP)

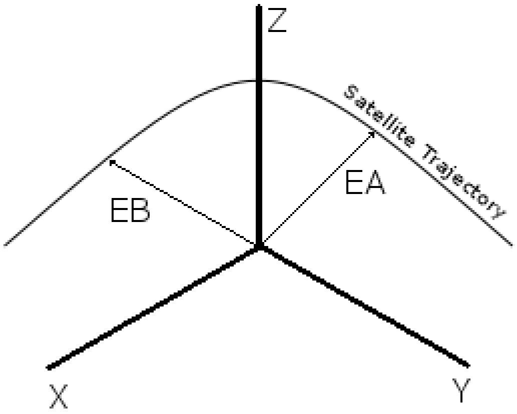

FIG. 8: Diagram of a satellite's trajectory and two position vectors that can be used to calculate its orbital plane. The origin is set to be the center of the Earth using an ECEF coordinate system.

and the Simplified Deep Space Perturbations model (SDP). The SGP model is used for objects that have an orbital period less than 225 minutes, and the SDP model is used for any object that has an orbital period greater than 225 minutes. These models work by taking the position of a satellite and its orbital parameters to determine where it will be relative to a known time. The parameters for these models are primarily generated using radar observations made by the North American Aerospace Defense Command (NORAD).

The output coordinates of the simplified perturbation models are with respect to an Earth-Centered, Earth-Fixed (ECEF) coordinate system. This coordinate system puts the origin at the center of the Earth and aligns the x-axis with 0 degrees latitude and longitude. This particular coordinate system can be difficult to work with since it doesn't map to the rotating Earth surface. Therefore, it is useful to translate the coordinates of a satellite to another system such as latitude and longitude to represent the geographic location.

In order to calculate the Longitude, Equation 2 is used where GMST is the Greenwich Mean Sidereal Time. The calculation for GMST is shown in Equation 3 where D is the fractional number of days since January 1, 2000 at 12pm GST [32].

$$Longitude = \arctan\left(\frac{Y}{X}\right) - GMST \tag{2}$$

$$GMST = 18.697374558 + 24.06570982441908 * D \tag{3}$$

4.2 ACQUISITION AND LOSS OF SIGNAL

Acquisition and loss of signal are generally estimated to be when a satellite rises and falls five degrees above the horizon, respectively. If the coordinates of both the ground station and the satellite are known, vector calculus can be used to determine when the satellite is above the horizon. If $\hat{\mathbf{G}}$ is a three dimensional point which represents the ground station location relative to the ECEF frame and $\hat{\mathbf{S}}$ is the position of a satellite, a vector $\hat{\mathbf{H}}$ can be calculated using those two points as shown in Equation 4. The angle between $\hat{\mathbf{G}}$ and $\hat{\mathbf{H}}$ can be found using the dot product and represents the elevation angle of the satellite with respect to the ground station is shown in Equation 5.

$$\hat{H} = \hat{S} \cdot \hat{G} \tag{4}$$

$$\theta = \arccos \frac{\hat{G} * \hat{H}}{|G||H|} \tag{5}$$

Propagating the orbit of satellites can be computationally expensive. High Performance Computing (HPC) can be used in order to perform different types of analysis on the orbits of the satellites being monitored. The primary use for HPC is the scheduling of satellites for the different ground stations. The orbit of each satellite that is capable of communicating with the ground station network must be propagated to within a second each day in order to determine if or when it will pass within range of any of the ground stations. A secondary objective is monitoring all other known satellites and nearby debris in orbit to determine if they pose a threat to any of the satellites in the constellation.

4.3 GROUND STATION SCHEDULING

Scheduling the operation of the ground stations is computationally expensive and is an excellent task to perform on a high performance computing cluster. In order to properly schedule the operations of a ground station, several steps need to be performed.The first step is to propagate the orbits of all the satellites in the constellation for every second of the day and determine their position. In order to do this, the satellite should be propagated with respect to time and converted to ECEF coordinates. In order to do this, the two line elements (TLEs) that describe the orbit of the satellite from NORAD must be downloaded. This should be done just before the propagation models are run to reduce error. Then, the simplified perturbation models must be used to determine where the satellite is to the nearest second and convert it to the ECEF coordinate system. Since the position of the satellite must be known within one second, 86,400 simulation steps must be executed for each satellite for every day.

The next step is to determine when the specific satellite will pass over a given ground station and for how long. In order to do this, the location of each ground station must be converted to ECEF coordinates and Equation 5 employed to determine which satellites each ground station will be able to communicate with.

Finally, if more than one satellite passes within range of a ground station at the same time a tie breaking technique must be used. In order to perform this process efficiently, a set of rules must be established. The first rule is to rank the satellites by how long they will be in range of the ground station. A satellite with a higher priority than another satellite will preempt that communication priority. In order to maximize the efficiency of this technique, a second rule must be implemented to check for unacceptably small communications windows and remove those contacts. This is due to the fact that in the event that there are three or more satellites it is possible for a satellite with a priority in the middle to only be given a very small communications slot because it preempted a lower satellite, but a higher priority satellite preempts it within a short period of time. If different satellites employ different communications rates, priority can be given to the satellite that can transmit the most data. If a satellite was only within range of a ground station for 5 minutes but had a higher data rate than a satellite that would be within range for 7 minutes, it could be given a higher priority. A system that implements this technique will need some additional logic to make sure that satellites that communicate at lower data rates aren't always preempted if the system is working with a large number of satellites.

4.4 COLLISION DETECTION AND AVOIDANCE

On February 10th, 2009 two satellites collided at approximately 800 km in altitude. The satellites in question were the Iridium 33 satellite with the Russian Cosmos 2251 [33]. The collision of these satellites resulted in thousands of pieces of small debris that are now orbiting the Earth. This event indicated that collisions between satellites are possible and can cause major issues to other satellites. Collisions between objects tracked by NORAD in space are lethal. Therefore, it is desirable to forecast collisions or near misses before they happen. Although small satellites generally lack propulsion systems, measures can be taken to prioritize that as much data as possible be recovered from the satellite before it is lost.

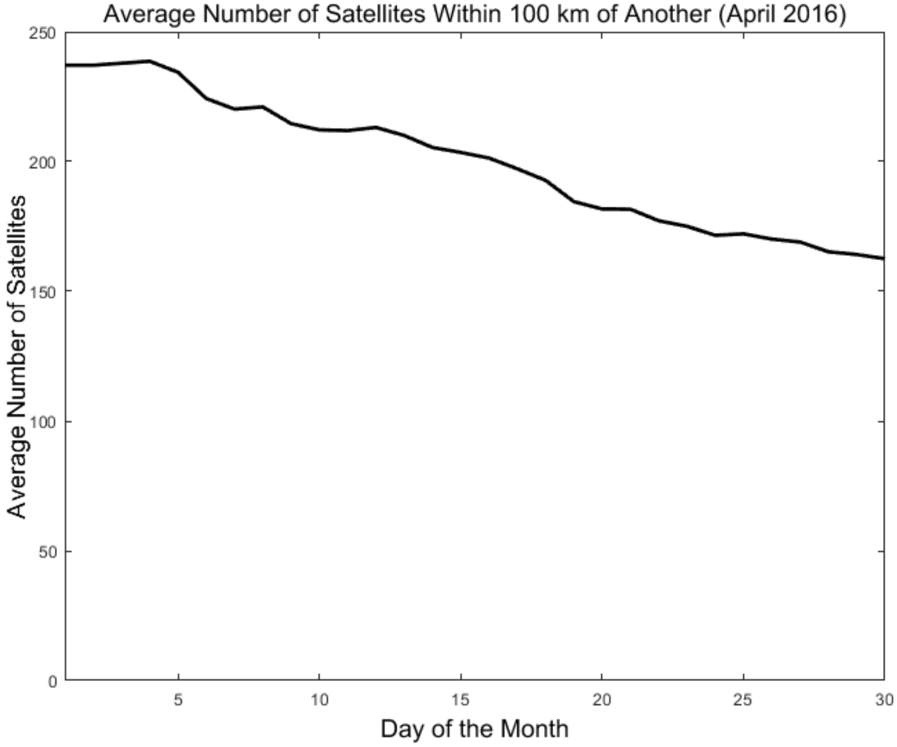

FIG. 9: Plot showing the average number of satellites that are within 100km of any other satellite for the month of April 2016.

In order to estimate how close satellites operate to one another, a simulation was performed to determine how many satellites were operating within 100 km of each other. While on the Earth, 100 km seems like a large distance, it is actually relatively close for satellites to be passing in the vastness of space. A model was developed to estimate and forecast collision probabilities. If the number of comparisons that need to be performed when determining collisions is C and N is the number of identified satellites that should be examined, the number of comparisons is:

$$C = \frac{N^2}{2} - \frac{N}{2} = \frac{N(N-1)}{2} \tag{6}$$

The simulation was written in Python and run on the Turing high performance computing cluster at ODU. It simulated the orbits of 2843 satellites every second for the month of April 2016 using the SGP4 algorithm and then compared their proximities. Any satellite within 100 km of another satellite was logged to a text file for further processing. The data sets were then loaded into a Postgres database for analysis. The simulation for each day took approximately four hours and forty five minutes using 16 processor cores on an Intel

Xeon CPU E5-2698 v3 running at 2.30GHz which is equivalent to approximately 76 CPU hours. It also performed approximately $1.05 * 10^{13}$ comparisons for the simulated month.

A chart of the average number of satellites within range of another satellite is shown in Figure 9. The number of satellites decreases over the period of a month due to the perturbations of the satellite orbits. The orbital parameters vary greatly over a period of several days, so they should be updated daily and any application that is used for satellite prediction should always use the latest orbital elements.

The simulation picked up TERRASAR-X and TANDEM-X passing within several hundred meters of each other. This was expected since they are twin satellites and are designed to operate in a close formation [34]. This shows that the simulation is valid and is working.

CHAPTER 5

COMMUNICATIONS ANALYSIS

In this chapter, an analysis of the operation and implementation of a small satellite mission with respect to the communications system was performed. Specifically, the requirements associated with the ground station that communicates with the CubeSat mission and a framework is presented that can be used to determine the time interval during which a CubeSat will be within line of sight of the ground station. A link budget analysis is presented as well as an outline of implementation aspects for CubeSat missions.

Assuming that the satellite occupies an overhead orbital trajectory as illustrated schematically in Figure 10, the duration, T, of a CubeSat pass over the ground station is obtained by dividing the arc length corresponding to its trajectory by the tangential speed of the satellite, and is given by

$$T = \frac{\left[2 \arccos\left(\frac{R_E}{R_E+h}\right)\right](R_E+h)}{v} \tag{7}$$

where $R_E = 6,371$ km is the radius of the Earth, h is the altitude of the CubeSat trajectory, and v is the circular velocity of the satellite. The values of T for orbital altitudes typical for CubeSat missions are given in Table 2.

From Figure 10 one can also see that the distance d between the CubeSat and the ground station varies as a function of the elevation angle φ of the satellite above the horizon, and using the geometry of the trajectory in Figure 10 d is given by

$$d = \sqrt{(R_E+h)^2 - R_E^2\cos^2\varphi} - R_E\sin\varphi. \tag{8}$$

For illustration purposes, Figure 11 shows the variation of the distance, d, between a CubeSat at different low Earth orbit altitudes and the ground station, as a function of the elevation angle φ. We note that while Figure 11 shows the elevation angle ranging from 0° to 180°, in practical scenarios the range of the elevation angle for a LEO satellite is about 160°, starting from at least 10° to no more than 170°, since for elevations outside this practical range, the probability of having line-of-sight visibility tends to zero due to

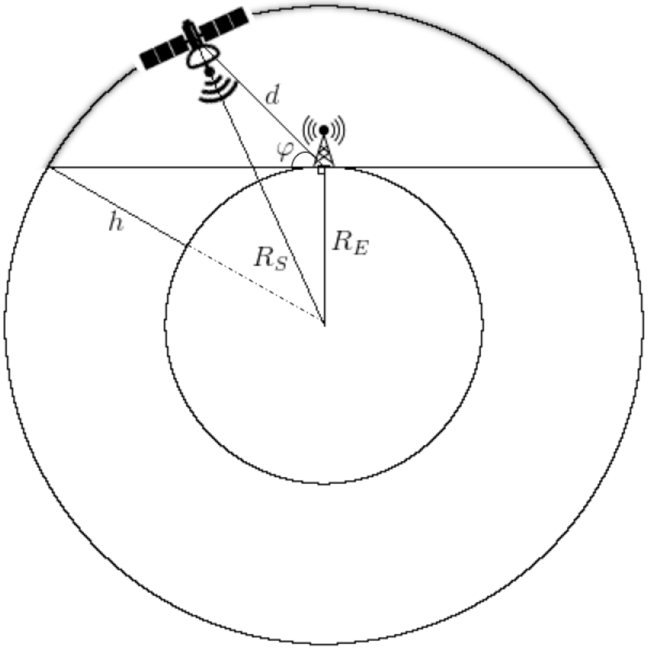

FIG. 10: CubeSat trajectory in low Earth orbit assuming that the orbit is completely circular. Using this figure and some basic geometry, the time that the satellite will be over the ground station can be determined. The parameters d and φ can be used to determine the Doppler shift and acquisition of signal respectively.

obstructions. Thus, for elevations in the practical range, the maximum distance between the LEO satellite and the ground station is summarized in Table 3.

5.1 CUBESAT-TO-GROUND STATION RADIO LINK

The orbital parameters of the CubeSat trajectory influence the characteristics of the radio link connecting the CubeSat system to the ground station and determine its performance parameters. Specifically, path loss affects the power of the transmitted signal and is characterized using a free-space propagation path loss model [36]

$$L_p = 20 \log_{10} \left(\frac{4\pi d f}{c} \right), \tag{9}$$

where d is the distance between the CubeSat and the ground station, given by Equation (8) and shown in Figure 11; f is the transmission frequency; and $c = 300,000$ km/s is the speed of light. The path loss depends on the frequency used by the CubeSat communication system, as well as on the distance between the CubeSat and the ground station, and is illustrated for different frequencies in Figure 12. We note that there is significant variation

TABLE 2: CubeSat visibility times at different altitudes for overhead trajectory [35].

Flying altitude h	Tangential speed v	Visibility time T
200 km	7.784 km/s	6.9 min
300 km	7.726 km/s	8.6 min
350 km	7.697 km/s	9.4 min

TABLE 3: Maximum distance between CubeSats and ground station at different altitudes for overhead trajectory.

Flying altitude h	Maximum distance from ground station
200 km	800 km
300 km	$1,200$ km
350 km	$1,400$ km

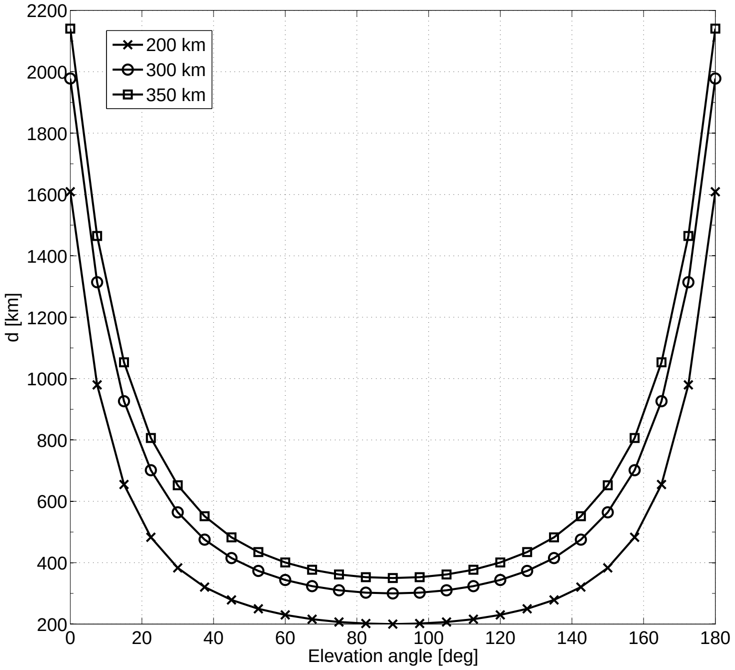

FIG. 11: Variation of distance d from ground station versus elevation above the horizon.

of the path loss affecting the CubeSat-to-ground station radio link while the CubeSat is visible from the ground station. As can be observed from Figure 12, the path loss is at a minimum when the CubeSat is at zenith in its trajectory (90° elevation angle) and is at a maximum when the CubeSat rises/sets over the horizon (0°/180° elevation angles).

From Figure 12 we note that, as was expected, the lowest path loss values correspond to the VHF band (144 MHz) while the highest path losses correspond to the L-band (1,265 MHz). We also note that, for a given frequency band, the path loss magnitude depends also on the CubeSat trajectory altitude.

Another aspect that must be considered in the study of the CubeSat-to-ground radio link is the shift in signal frequency implied by the Doppler effect, also referred to as the Doppler shift [36], which is given by

$$f_D = f\frac{v}{c}\cos\varphi. \tag{10}$$

The variation of the Doppler shift frequency as a function of elevation angle, φ, above the horizon is shown in Figure 13. As can be seen, the Doppler shift is minimal for the VHF band (around 3.7 kHz for 144 MHz), increases for the UHF band (11.1 kHz for 433 MHz), and becomes significant for the L-band (32.5 kHz for 1,265 MHz). Depending on the bandwidth allocated to the CubeSat-to-ground station link, correcting the Doppler shift may be necessary to avoid interference to adjacent bands, and will affect the complexity of the communications sub-system for CubeSat missions [37].

5.2 LINK BUDGET ANALYSIS

A power budget analysis to develop a representative link budger for nanosatellite CubeSat missions is based on a case study of a meteorological science mission [38]. According to [38], for a highly adaptive picosatellite with a calculated in-orbit core bus power of 0.864 W, a fraction of 5% of the available power has been allocated to the communications sub-system, that is 0.043 W or 16.33 dBm. The core bus power value scales up by a factor of 10 for a highly adaptive nanosatellite, with a mass of 10 kg to 8.64 W, of which the same fraction of 5% is allocated to the communications sub-system, that is 0.43 W or 26.33 dBm.

By extrapolation, one can assume that, depending on the size of the CubeSats, the power allocated to the communications sub-system:

- $P_t = 1 \times 0.043 = 0.043$ W= 16.3 dBm for a 1U CubeSat,
- $P_t = 3 \times 0.043 = 0.129$ W= 21.1 dBm for a 3U CubeSat,
- $P_t = 6 \times 0.043 = 0.258$ W= 24.1 dBm for a 6U CubeSat.

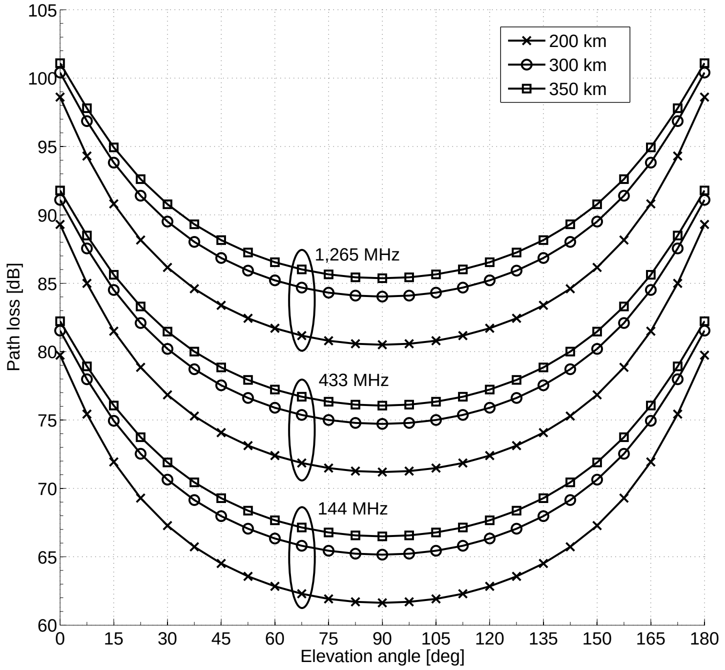

FIG. 12: CubeSat-to-ground station path loss variation as a function of its elevation angle above the horizon for different frequencies and altitude values.

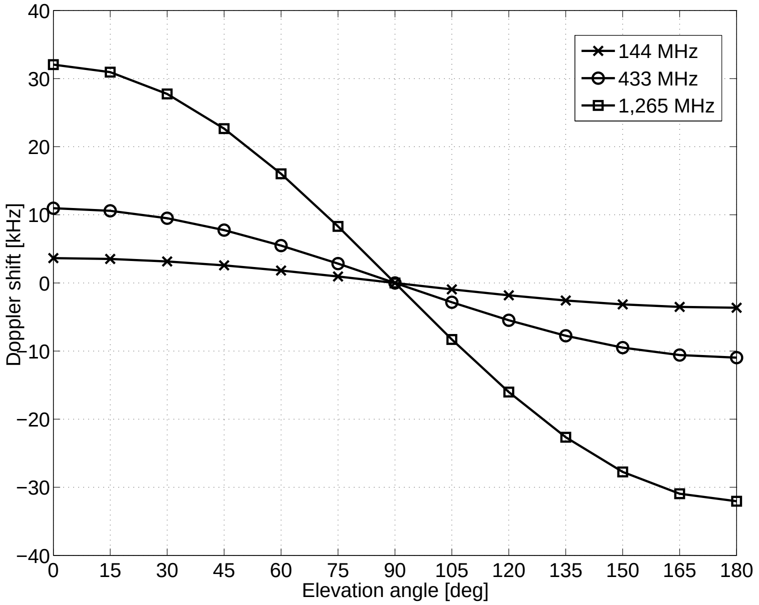

FIG. 13: Doppler shift corresponding to the CubeSat signal at the ground station. The tangential speed of the assumed CubeSat is $S = 7.7$ km/s.

To determine the value of the received power at the ground station receiver, P_r, we use the free-space propagation model [36]

$$P_r = P_t + G_t + G_r - L_p \tag{11}$$

where power values (P_t), antenna gains (G_t & G_r), and path losses (Lp) are expressed in dBm, dBi, and dB units, respectively. For a conservative link budget analysis we assume that omnidirectional antennas were used in the CubeSat and the ground station, having gains equal to 0 dBi, along with a noise floor of 90 dBm. Using Equation (11) and the path loss values shown in Figure 12, following results were obtained:

- For a 1U CubeSat, with a transmit power budget $P_t = 16.3$ dBm, the maximum path loss at the highest altitude of 350 km is $L_p = 102$ dB, 92 dB, and 82 dB for L-band, UHF, and VHF respectively, and implies corresponding received power values at the ground station $P_r = -85.7$ dBm, 75.7 dBm, and -65.7 dBm, for L-band, UHF, and VHF respectively. With noise floor at 90 dBm at the ground station receiver we obtain SNR values of 4.3 dB, 14.3 dB, and 24.3 dB for L-band, UHF and VHF, respectively.

- For a mission using a 3U CubeSat with a transmit power budget $P_t = 21.1$ dBm the received power at the ground station is $P_r = -80.9$ dBm, -70.9 dBm, and -60.9 dBm, for L-band, UHF, and VHF respectively, with corresponding SNR values for 90 dBm noise floor at the ground station receiver of 9.1 dB and 19.1 dB, and 29.1 dB for L-band, UHF, and VHF, respectively.

- For a 6U CubeSat mission for which P_t is 3 dBm above that of a 3U CubeSat, the P_r and SNR figures will be 3 dBm above the corresponding values for the 3U CubeSat.

It should be noted that, with the variable path losses implied by the varying distance between the CubeSat transmitter and the ground station receiver, the SNR varies during the transmission. To maintain a constant SNR at the ground station receiver, the CubeSat transmitter could dynamically adjust its transmit power while it was within the radio range of the ground station by receiving power level status messages from the ground station. Transmitting at a constant SNR that matches the required SNR value will also ensure that the CubeSat radio transmits with the minimum amount of power and will also contribute to increasing the lifetime of the mission by conserving battery life. Power adaptation can be accomplished by using trajectory data to adjust the transmit power P_t of the CubeSat radio once the CubeSat becomes visible to the ground station and establishes a radio link with it.

While it is not feasible to use directional antennas in the CubeSat transceiver, they should be considered for use at the ground station, since their use is beneficial to the CubeSat-to-ground station radio link. In addition, the use of pre-amplifiers along with more sensitive receiver front-end boards that can distinguish signals at noise floors of -100 dBm or lower can further enhance the performance of the radio link. This shows that SNRs of the order of $10 - 20$ dB can be achieved; when used in conjunction with digital modulation schemes and error correction codes, this implies acceptable bit error rate levels.

5.3 INTER-SATELLITE COMMUNICATIONS LINKS

Communications between satellites may or may not be similar to communicating with a ground station, depending on the configuration of the cluster. If the satellites in a constellation maintain constant distances from each other there will be no Doppler effect. The path loss can be expected to be less over distances between satellites than for similar distances between a satellite and ground station due to negligible atmospheric attenuation, although it may only be a few dB. However, it would take a large number of satellites in the same orbital plane to be able to maintain distances between satellites that are the same distance as orbital altitude.

In order to prove this, the law of cosines can be used to approximate the distance between satellites in the same orbital plane. A simplified version of the law of cosines, which takes two parameters to calculate the distance, is given in Equation 12. If R is the radius from the center of the Earth to the satellite and N is the number of satellites in the constellation

$$Distance = R\sqrt{2\left(1 - \cos\frac{2\pi}{N}\right)} \tag{12}$$

An example showing the distance between satellites at an altitude of 350 km is shown in Figure 14. The distance between satellites decreases rapidly until there are on the order of 120 satellites in the constellation. With that population, the distance between satellites is 352.2 kilometers, approximately the same distance between adjacent satellites in the constellation and a ground station when the satellite is directly overhead.

Using the path loss equation (Equation 11), the path loss between satellites can be calculated for 144 MHz, 433 MHz and 1265 MHz and is displayed in Figure 15. As shown in the figure, the choice of frequency is critical. The path loss at 1265 MHz with a large number of satellites never reaches the path loss at 144 MHz with a small number of satellites. At 120 satellites in the constellation with a frequency of 1265 MHz, the path loss is about 150 dB, which is rather substantial. Due to the limited space on a small satellite, large high-gain antennas can't be employed, so in order to increase the received signal to approximately -100 dBm, which is the noise floor for most receivers, 50 dBm of power would have to be used. This would require a 100 watt transmitter on each satellite, infeasible given the power budget. However, employing a frequency of 144 MHz with the same number of satellites, the path loss is only 126 dB which could be compensated with a one watt transmitter and is much more feasible. Using a frequency of 433 MHz, the path loss is about 136 dB and a 10 watt transmitter could be used, although this would be at the top end of expected power budgets.

Although Planet Labs does not explicitly state why they plan to use 100 to 150 satellites, they are probably choosing those numbers based on the above analysis [13]. If they keep the number of satellites in their constellation between 100 and 150 satellites the path loss will be low enough for their satellites to communicate with each other. If they have at least one satellite within range of a ground station at all times, they would be able to communicate with every satellite in the constellation if their satellites are networked together.

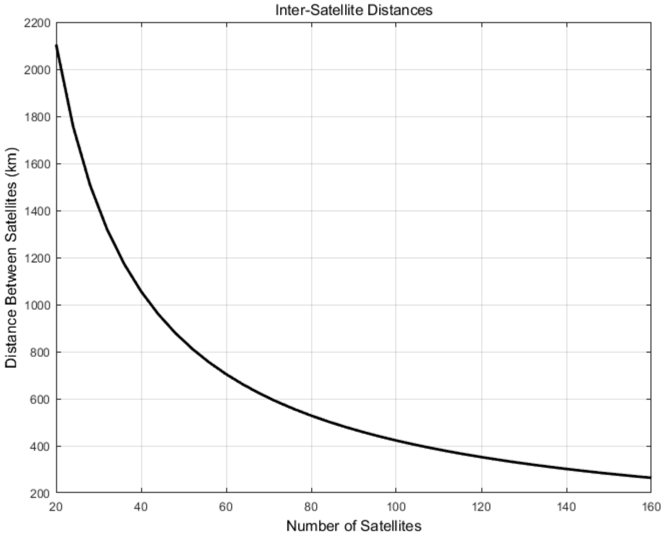

FIG. 14: Distances between satellites in the same orbital plane when given the number of satellites in the constellation.

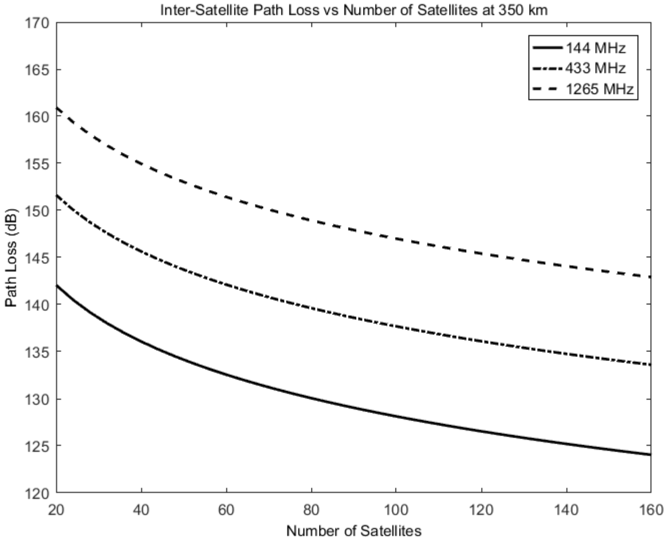

FIG. 15: Inter-satellite path loss for several frequencies based on the number of satellites in the constellation where the orbital altitude is 350 km.

CHAPTER 6

IMPLEMENTATION ASPECTS

In order to enable cost-effective implementations and provide access to space communication capabilities for university teams launching CubeSat missions, the use of commercial off-the-shelf (COTS) components is preferred. Those systems can then be modified for use in space. COTS products can be a fraction of the cost of space rated equipment, and since CubeSats are generally only in orbit for relatively short periods of time, that equipment can often operate successfully for the duration of the mission.

6.1 SOFTWARE DEFINED RADIO

An important consideration when implementing a transceiver that will be used for a CubeSat is the requirements of the planned digital modulation scheme to be used by the communications sub-system, which is determined by the specific requirements of the science experiment to be performed by the spacecraft. Specifically, knowing the amount of information, I, to be transmitted to the ground station during one trajectory pass, along with the time, T, that CubeSat is visible from the ground station, one can determine the required data rate, R, on the required CubeSat-to-ground station radio link as ($R = I/T$ bits/s).

A digital modulation method suitable for CubeSat missions is frequency shift keying (FSK), a power efficient scheme that has been used with other low-power, low-data rate satellite applications (such as global paging via satellite [39]). We note that the uncoded bit error probability for binary FSK modulation over land-based mobile satellite channels is on the order of 10^{-2} at 15 dB SNR, which can be further improved by using appropriate coding techniques [39]. In addition, binary FSK has the advantage of very simple implementation using a basic micro-controller as modulator and demodulator [40]. This makes it ideal for use in the satellite side of the communication sub-system of a CubeSat mission, for transmitting information to the ground station, since in this case simple hardware with low power requirements are usually preferred. However, the supported data rate is low, which makes the use of binary FSK less practical for transferring large amounts of data such as images or video, during the short time intervals a CubeSat is visible from the ground station.

To implement more versatile transceivers that support additional modulation schemes and can be incorporated onboard the CubeSats, software defined radios that support field programmable gate arrays (FPGAs) should be used [7]. FPGAs can be easily reprogrammed to support different requirements and enable the implementation of versatile transmitters and receivers for CubeSat missions. They also have the potential to reduce system cost and development time while providing significant flexibility in terms of the modulation and demodulation schemes that are available. Multiple choices for implementing transceivers for both the CubeSat and the ground station are available using this strategy. These include flexible boards with transmit-receive capabilities such as the BladeRF [25], which is moderately priced, the Universal Software Radio Peripheral (USRP) [41], which is more expensive but has emerged as the leading choice for SDR educational activities [42], or the more costly and powerful Wireless Open-Access Research Platform (WARP) [26, 43]. For example, the USRP B200, manufactured by Ettus Research (a National Instruments company), provides a fully integrated, single board radio platform with continuous frequency coverage from 70 MHz to 6 GHz and up to 56 MHz of real-time bandwidth, and is a cost effective alternative to the more expensive amateur radio equipment (such as ICOM 910-H radios) for implementing the ground station end of the communication sub-system for CubeSat missions.

The basic building blocks for a software defined radio are: a computer, FPGA and an analog to digital converter for a receiver. If a transceiver is required, a digital to analog converter can be added. The FPGA acts as an interface between the computer and the analog to digital converter. For systems such as the USRP B200 and the BladeRF, the FPGA packages the samples into the appropriate order and transfers them to the computer over a USB connection. The FPGA may also perform some digital signal processing tasks such as re-sampling, but the majority of the signal processing is performed with the computer. The primary constraint in software defined radio systems is the number of samples per second that they can process. It is common to find software defined radios that are capable of performing hundreds of millions of samples per second such as the USRP X310 [44]. In order to process data at this rate, the computer needs to have a high performance processor, and the interconnect between the computer and the software defined radio needs to be able to support the data rate. For sample rates in the millions of samples per second, a variant of USB is usually capable of providing enough throughput.

The sampling parameters are determined by the data converters. Due to the Nyquist rate, the data converters must run at at least twice the sampling rate of the maximum frequency that must be transmitted or received. The sample size in bits specifies the resolution of the data converter. This parameter is important since it determines the dynamic range of the data converter, which controls the accuracy of the converter relative to the original signal. For example, the original USRP used an Analog Devices AD9862 which provided a 12 bit ADC operating at a rate of 64 million samples per second (MSPS) and a 14 bit DAC operating at a rate of 128 MSPS [23]. This means that it was theoretically capable of receiving a signal up to 32 MHz and transmitting a signal up to 64 MHz. The voltage reference is set to 2 volts peak-to-peak; thus, the ADC has a resolution of $2/2^{12} = 488$ *microvolts* and the DAC has a resolution of $2/2^{14} = 122$ *microvolts*.

Programming and configuration of the SDRs may be accomplished using open-source software development toolkits such as GNU Radio [45, 46], as well as software packages for system design, algorithm development, simulation and data visualization such as MATLAB and LabVIEW [42]. Currently, MATLAB and LabVIEW can be used only in conjunction with USRP SDRs, while GNU radio can be used both as a simulation tool, without any hardware, and as a programming tool for multiple SDRs boards including USRP, BladeRF, and others.

6.2 POWER AMPLIFIERS

Another challenge when designing a small satellite mission is the power output of the transceivers. On the ground station, physical space and access to power is hardly restricted. This means large power amplifiers can be installed in order to be able to send a signal to reach the satellite. The issue is that physical space and power is at a premium on a small satellite. Power amplifiers generate large amounts of heat and due to the limited heat transfer rate in the vacuum of space, these devices could damage the electronics.

Most small satellites transmit at most a few watts of power. This is feasible and does not require too much power or produce too much heat. The power output of most software defined radio boards is approximately 10 dBm which is 0.01 watts. The goal of most amplifiers on small satellites is to boost the output power to around 30 to 35 dBm (which is between one and three watts). In order to reach these levels, the amplifier needs to provide a gain of about 20 dB.

Currently, there is no off-the-shelf amplifier that can be used directly for a Cubesat mission without modification. If an amplifier is required, it would have to be designed using discrete components. A generic schematic that can be used when designing an RF amplifier is shown in Figure 17 where V1 is the input signal and R4 represents an antenna load. In order to remove the DC bias for both the input and output, capacitors C1 and C2 are used. These capacitors are designed with a low capacitance in order to act as high pass

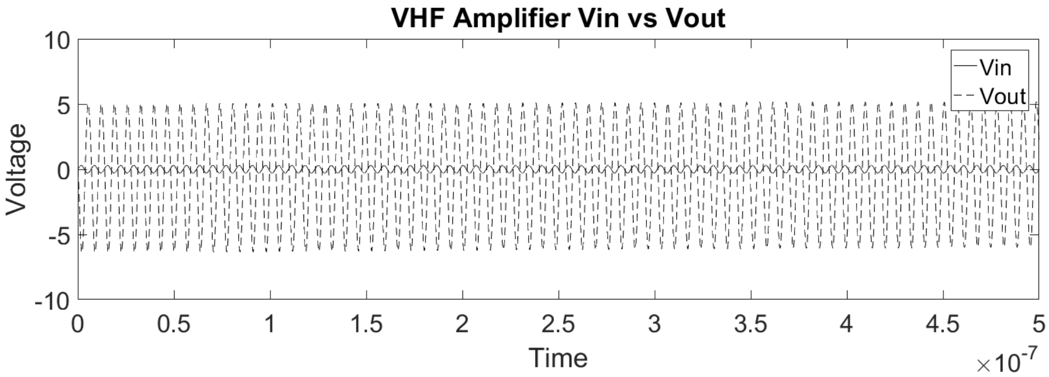

FIG. 16: Traces of Vin and Vout for the VHF amplifier simulation.

filters. The MOSFET M1 is the active element that performs the amplification. In order to provide power to the drain of the MOSFET, R1 and L1 are used. These components act together to form a low pass filter that causes the amplifier RF to flow through the capacitor C2. R1 sets the maximum current that can flow through the drain of the MOSFET and acts as a protection device. R2 and R3 are used to set the gate bias of the MOSFET which is typically set to the linear region of the amplifier. Since this is a common source amplifier configuration, the analog voltage gain is approximately $-g_m * R_1$ where g_m is the transconductance of the MOSFET [47].

Most RF MOSFET manufacturers (such as Renesys and Fairchild Semiconductor) provide reference schematics that can be used with their devices as a starting point for building an RF amplifier. An example of a MOSFET that could be used for a small satellite amplifier is the RD15HVF1 by Mitsubishi Electric. It is capable of outputting up to 15 watts at up to 520 MHz and Mitubishi Electric provides a reference design [48]. A VHF amplifier based off of the RD15HVF1 was designed and simulated using SPICE circuit simulation software. A schematic of the amplifier is shown in Figure 18. In order to perform the simulation, a RF signal with an amplitude of 0 dBm and frequency of 146 MHz was inserted into the amplifier. The signal was amplified to an output level of approximately 25 dBm as shown in Figure 16.

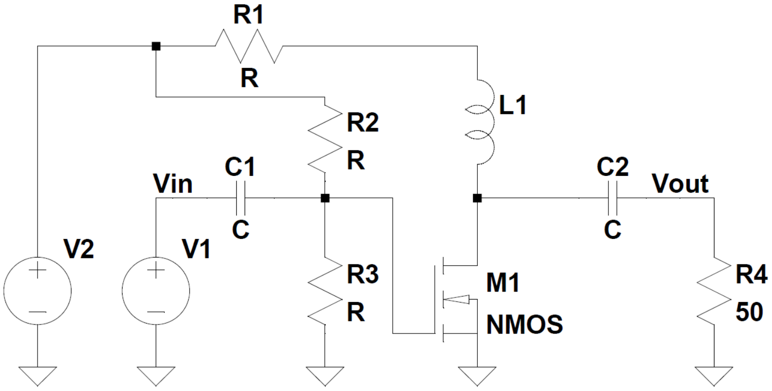

FIG. 17: Schematic for a generic RF amplifier. The components are dependent on the frequency of operation and output power level required. The RF input is inserted at Vin and the amplifier output is present at Vout.

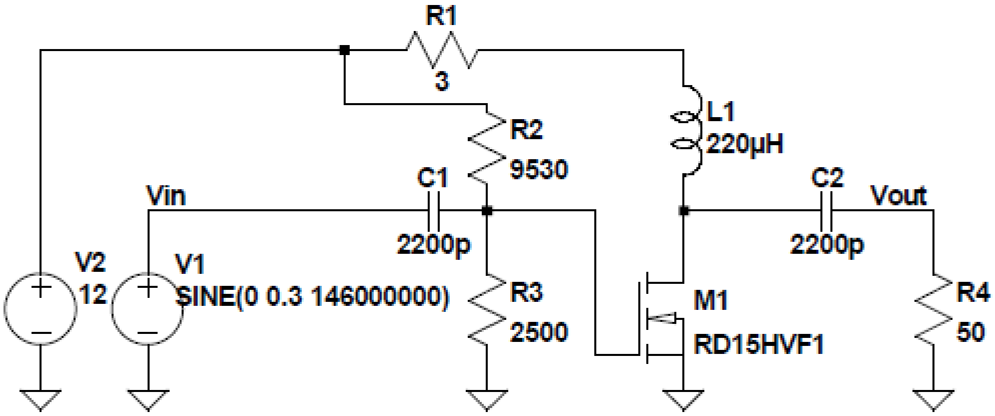

FIG. 18: Schematic for an RF amplifier based on the RD15HVF1 MOSFET from Mitsubishi Electric.

CHAPTER 7

NETWORK IMPLEMENTATION

Over the past several years, a great deal of effort has gone into the development of small satellites. Their relatively low cost makes it possible for universities to build and launch their own space experiments. They are capable of being in orbit for as little as several months or as long as several years, so there is a pretty decent return on investment. Small satellites are popular enough now that it is possible to buy a starter kit and a launch from several commercial vendors, and it is possible to go from a conceptual satellite design to orbit in less than a year. This is a short enough time frame that it is possible to iterate from a prototype satellite to a mature product in only a few years, and several different designs can be tested concurrently. However, even though there have been major advances in small satellite technology, there are still many issues that exist when designing them. Keeping mass and power budgets balanced is very important and still very challenging. It is important to recognize that all of the hard work that goes into preparing and designing the satellite will be lost if the communications system fails to perform as expected. The purpose of this thesis had been to explore a communications system architecture which can be successfully implemented on a small satellite. A small satellite communications system needs to be extremely reliable. Retransmissions are costly due to the extra power required for retransmissions as well as the relatively high latency between a satellite and the ground station. In order to reduce the need for retransmissions, forward error correction is implemented. One of the applications that is examined in this paper is using a satellite with multiple ground stations. These ground stations could be sensors or controls located in remote parts of the world. The communications system that will be implemented in this case study is designed to support multiple devices per channel by assigning them different data link layer addresses. This technique allows multiple stations to share the same RF channel, which may increase the efficiency of the channel due to a higher utilization ratio.

7.1 PREVIOUS WORK

One of the first topics that this section explores is the Internet of Things and how many devices are currently being connected to the internet. This is followed by techniques that are currently used to send data across wireless links. The MAC layer is explored first by looking at the AX.25 protocol. A modulation method for AX.25 is discussed next by exploring different types of frequency shift keying (FSK). A discussion of a satellite communications link concludes this section.

The world is progressing to a point where most electronic devices are connected to the internet and will be able to be controlled remotely. Most devices will be located in urban or household environments where they will have access to wireless access points. Other devices will be located in remote regions of the world where internet connectivity will be limited or uavailable. Traditionally, this connectivity has been accomplished using supervisory control and data acquisition (SCADA) systems communicating over two-way terrestrial radio links [49]. These systems usually operate on a trunked radio system shared with other mobile radio users such as the police, fire departments and other civil services. These systems are designed for high latencies (due to having to wait to access the shared channel) as well as low throughputs (the modulation methods must be compatible with a standard voice channel). This makes these devices perfect for using with a satellite communications system. Some of these systems are capable of communicating with each other without having to wait for user input. This type of traffic is called machine-to-machine traffic (M2M) [49]. A common example is the credit card processing system on a vending machine. Once the user inserts their card, the vending machine uses a wireless adapter (generally either a WiFi or cellular adapter) to connect to the internet and process the payment. Since there is no user input in order to verify and process the payment information, it is considered to be M2M traffic. The vending machine can also autonomously send inventory information to a central location so that operators are able to see if it needs to be restocked. This same ability can be applied to machines in remote regions using a satellite communications system. Valves on oil pipelines in remote regions could be closed or opened remotely by operators in urban areas on the other side of the world. Valves could be closed automatically if oil leaks are detected tens or hundreds of miles. This same technique could apply to controlling the electric grid [49], pipelines for the public drinking water supply and monitoring maritime vessels.

Encapsulating IP frames inside of AX.25 is possible and relatively straightforward [50]. This makes it relatively easy to use existing software over an AX.25 link. Due to the relatively slow bandwidth that is possible over a standard radio link, it does not function very well for high bandwidth tasks such as browsing the internet. However, for sending small amounts of information, it is perfectly suitable. Remote sensors could send information such as temperature or pressure in binary format. If 32 bit integers are used for all values, it would require very modest data transmissions to be sent across the link. A group of amateur radio operators in New York successfully used an AX.25 link to replicate a MySQL database [50]. The distance between the two sites was approximately 1.6 miles. The database stored troop information for the Boy Scouts Jamboree such as the troop name, scout master, number of scouts and their location. The group also tested a blogging application with similar success. This shows that using AX.25 to encapsulate IP frames can be very versatile. Just as the link was capable of being used to replicate a MySQL database without altering any code, an AX.25 link between a ground station and satellite (or a sensor to a ground station via a satellite) could be used to transport IP packets for other network programs. For example, if an AX.25 link was set up between a ground station and a satellite, a user could use secure shell (SSH) to remotely connect to the satellites terminal as well as download or upload files to the satellite at the same time. Several different low bandwidth network programs could run on the satellite computer without requiring any interference with each other or inherent lag.

There has already been research done using AX.25 frames modulated via FSK to transmit data for satellite communications [51]. It is relatively simple to generate the symbols needed for transmission using a variety of hardware including field programmable gate arrays [51], making it very easy to implement and test. FSK can send multiple bits per symbol by using multiple frequencies to increase throughput. FSK-2 (where only one bit is sent per symbol) is the most common form of FSK, however, FSK-4 uses four frequencies to transmit two bits per symbol. FSK has been used by satellites before, but it requires a higher signal to noise ratio than other modulation methods such as binary phase shift keying (BPSK). Figure 19 shows the bit error probability for the two when given signal to noise ratio. Since both FSK-2 and BPSK send the same number of bits per symbol (which is one), it makes sense to use BPSK because it can perform better with respect to noise [52].

Most analysis of communications systems uses the basic Friis equation for calculating power at the receiver [53]. This is a very useful equation; however, there are often other factors that cause signal loss in the real world such as multipath fading and signal absorption due to the atmosphere. Multipath fading can be a problem depending on the environment. If the environment is rural it is unlikely that there will be any major issues. An urban environment has more objects that the signal can reflect off of and more sources of noise. Shadowing can also be a problem if the receiver is located next to a large building, trees or a mountain where it cannot obtain a direct line of sight to the satellite. Signal absorption can be a problem depending on the frequency [53]. Ionospheric absorption can vary from 0.1 dB

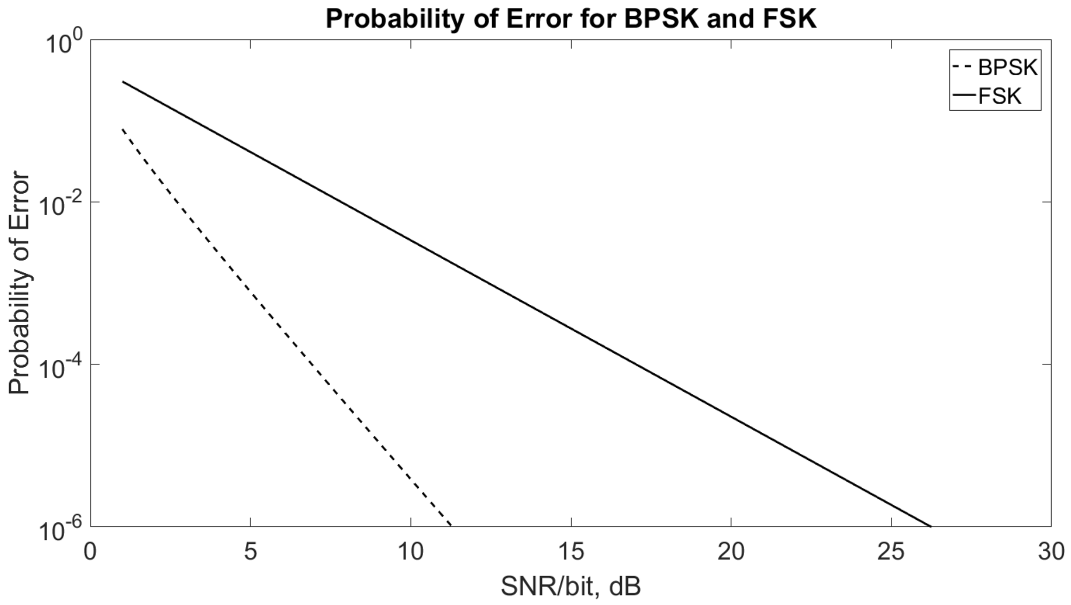

FIG. 19: Probability of error for FSK and BPSK when given signal to noise ratio

to several dB depending on the time of day (it is highest in the locations nearest the Sun) for frequencies less than 1 GHz [53]. Absorption can also be caused due to high humidity, however, it is usually less than 1dB [53]. Several different probability models can be used to simulate these effects. A Markov chain can be used to simulate whether shadowing is occurring on the receiver. Depending on its state, either a Rician or Rayleigh process can be used to simulate the fading [53].

7.2 ANALYSIS AND SIMULATION

Using AX.25 with FSK has been analyzed in the past. The purpose of this section was to explore the use of binary phase shift keying instead of frequency shift keying. The goal was to be able to determine the maximum altitude at which the communications system could operate using a specific set of parameters. The parameters that have been used for this analysis are typical of those used for a small satellite mission. They are based on actual devices so that the simulations could be relatively useful for those who wish to setup their own satellite missions. Most small satellites have a very small power budget. They do not have much area for solar cells and large high-current batteries have large masses. This simulation assumed a transmit power from the satellite of 1 Watt. There is not enough area on the satellite for a directional antenna, so a gain of 0 dBi was used to represent the worst possible scenario. The receiver parameters have been modeled on the hardware installed for the ground station at Old Dominion University in Norfolk, Virginia. The noise floor of the receiver is based on the USRP platform and is approximately -100 dBm. The antenna is a circular polarized antenna from M2 Antenna (model number 436CP30). This antenna has a gain of approximately 15.5 dB at about 436 MHz. The coaxial cable from the antenna to the receiver is hardline. In order to simulate this part of the system, 3.5 dB of loss was added to account for the loss in the coax cable. A satellite passing overhead at different altitudes was simulated, and the bit error rate and packet error rate was calculated. The altitude was simulated from 0 meters to 1500 kilometers using MATLAB. A frequency of 433 MHz was chosen since that is inside the amateur radio 70 cm band as well as the ISM band.

An analysis of the physical layer will be performed in order to indicate the modulation method. In order to reduce errors, forward error correction is added. The channel is simulated with a bit rate of 9600 bits per second and the results are discussed below.

In order to simulate the performance characteristics of a satellite communications system, the physical layer needs to be simulated first. The physical layer in this case is binary phase shift keying which can be modulated using a single carrier with or without a 180 degree phase shift. In a physical circuit, this can be done using a voltage controlled oscillator to a phase shifter. However, for simulating the signal in MATLAB, the following function may be used.

$$s(t) = \begin{cases} 0 & cos(2\pi f_c t) \\ 1 & cos(2\pi f_c t + \pi) \end{cases} \tag{13}$$

Demodulating the BPSK symbols is relatively straightforward as well. Two matched filters can be used in order to determine the phase of the received signal. From there, a comparator can be used to choose the correct binary value. A block diagram of this is shown in Figure 20.

The next layer that needs to be developed is the data link layer. In this case, it is going to be AX.25. The frame format for this layer is very simple and is shown in Table 7.2. In order to simulate this communications system, only unnumbered frames are used. These frames do no need to be acknowledged (which would significantly slow the throughput); therefore, if data needs to be retransmitted, it must be detected and requested at a higher protocol level.

Since unnumbered frames are used, the control field is always set to 0x03. The frame check sequence (FCS) is the cyclic redundancy check (CRC) 16 bit of the data that is to

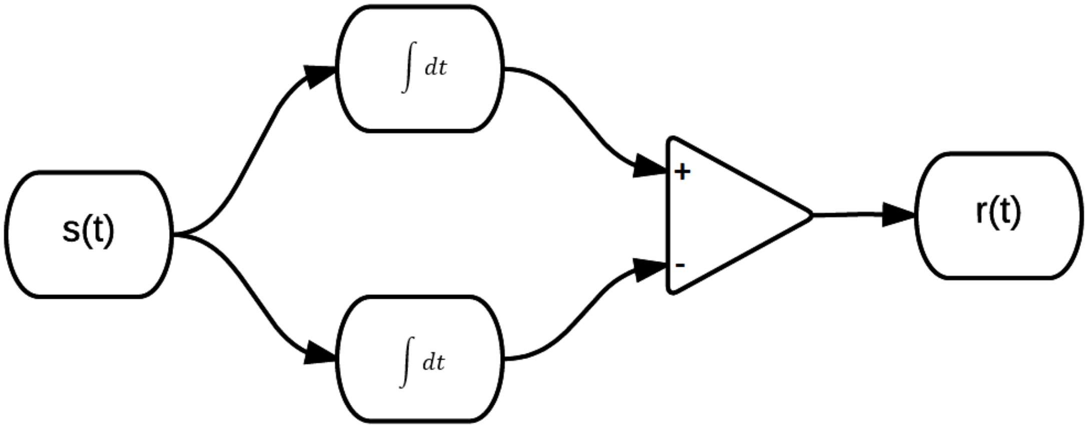

FIG. 20: Block diagram of a BPSK demodulator. The received signal is split and goes through two matched filters. One matched filter is phase shifted 180 degrees from the other. A comparator is used to see which binary bit is produced.

Flag	Address	Control	Payload	FCS	Flag
1 byte	14 bytes	1 byte	1-237 bytes	2 bytes	1 byte
0x7E					0x7E

TABLE 4: Format for an AX.25 frame. The maximum frame size is 256 bytes.

be transmitted. It is appended to the end of the packet and followed by the flag sequence. The address field is specified as shown in Table 5. Both the source and destination are 7 bytes long. The first six bytes specify the amateur radio call-sign of the sender and receiver. The 7th byte represents the SSID so that each device can have its own address based off the operator's call-sign.

The example is given using the authors call-sign. Once the packet was formatted, Hamming 7,4 forward error correction was applied in order to reduce the effect of bit errors. For encoding, the four data bits were multiplied using a Hamming matrix. For decoding, the received bits were multiplied using another matrix to obtain a syndrome vector. If any elements of the syndrome vector were not zero, then the bit that was in error was flipped

Address													
Destination (7 Bytes)							Source (7 Bytes)						
K	J	4	I	W	X	0	K	J	4	I	W	X	1

TABLE 5: Address format for the AX25 frame. The last byte in the source and destination address is the service set identifier (SSID)

in order to correct the error. The received bits were then multiplied by another Hamming matrix to obtain the decoded bits.

The channel is simulated by calculating the free space path loss and then adding additive white Gaussian noise. The free space path loss is calculated using the Friis transmission equation, shown as Equation 11. The path loss is used to reduce the transmitted signal to the appropriate level for reception. The additive white Gaussian noise is then simulated at -100 dBm and added to the signal. This simulates the sensitivity of the receiver. This -100 dBm figure is based on the specifications of the USRP software defined radio platform.

7.3 RESULTS

A simulation was run between 0 meters and 1500 kilometers. Figure 21 shows the bit error rate with respect to distance, and Figure 22 shows the packet error rate with respect to distance. They both start to increase around 800-900 kilometers where the packet error rate reaches 50% at approximately 1000 kilometers.

The system used binary phase shift keying for the modulation method and AX.25 frames for the data link layer. Hamming 7,4 forward error correction was used in order to reduce the impact of bit errors. This system worked well until a distance of approximately 800 kilometers. Since the parameters used for the simulation were based on actual hardware at Old Dominion University, it is expected that similar results would be obtained in actual operation.

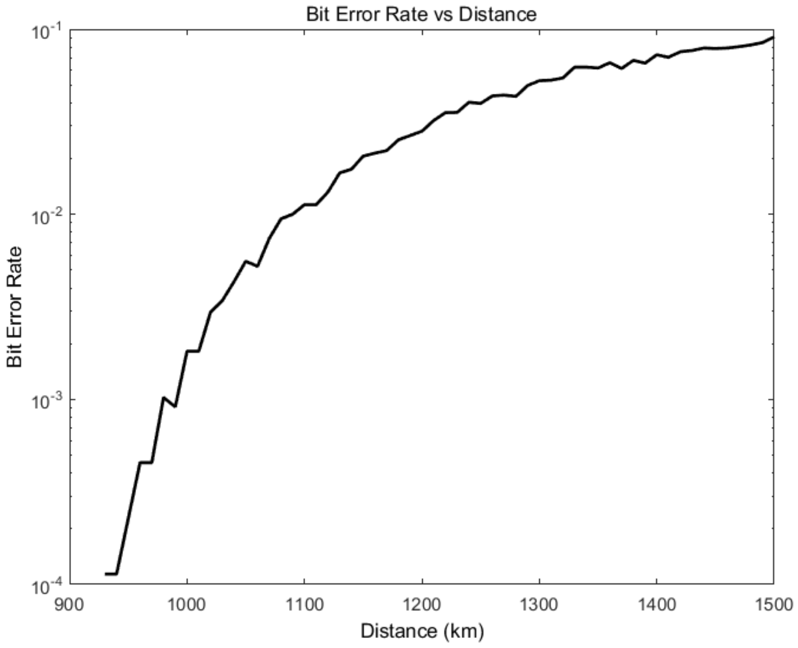

FIG. 21: Bit error rate as distance increases from 900 to 1500 km

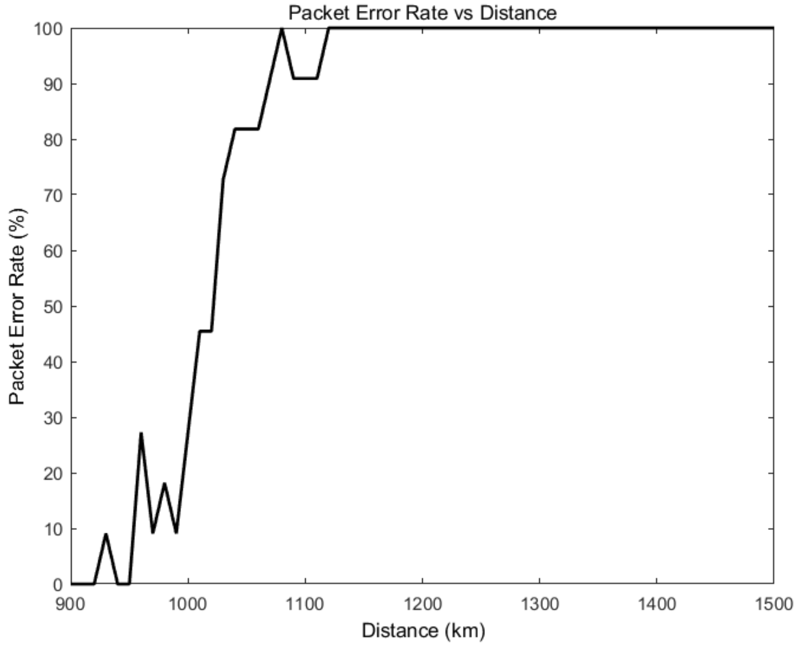

FIG. 22: Packet error rate as distance increases from 0 to 1500 km

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

In this thesis, a comprehensive analysis of a global ground station communications system for small satellites was performed. This thesis began by introducing the individual components of a small satellite mission which included the radios and antenna equipment that is needed on the ground as well as the satellite hardware that will orbit the Earth. The various pieces of software required in order to interface the different components of the system as well as possible choices for programming languages were discussed. The thesis then examined the communication channels that would have to be considered between the ground stations and the satellites passing overhead. A discussion of software defined radios was included due to their increasing prevalence in not only small satellite communications systems but also in all communications systems in general. In order to increase the power transmitted from software defined radios, RF amplifiers must be employed. Analysis and implementation of a VHF amplifier was performed. The thesis then concluded with a network implementation utilizing binary phase shift keying at the physical layer between a CubeSat and ground station which could be implemented for a small satellite communications mission.

Work will shortly begin that is based on this thesis to implement a global ground station network that can be used by Old Dominion University and its collaborators. An implementation will be designed and tested at ODU to verify that it is operating properly. From there, it is anticipated that the system will be expanded to include other university ground stations in the Commonwealth of Virginia so that they will all be linked and complement each other by supporting different transmission frequencies due to the choice of antennas at each site. It is hoped that if the system proves to be a success with several ground stations in the continental United States, then ground stations in other countries can be added. This system will be the first implementation of its type and should provide for many new research opportunities related to small satellites, their operations and communications in general. It is expected that the work done on implementing this system will be beneficial to others currently interested in launching their own small satellite missions.

Several different RF amplifiers will be designed for various frequency bands in order to maximize the power that the software defined radios at Old Dominion University can transmit. A simulator will be developed which will be able to dynamically simulate the path loss between a satellite and the ground station so that all of the hardware and software can be tested before it needs to perform in production. It is hoped that with the development of these pieces of hardware, more research can be performed with respect to maximizing the performance of the communications channel between the satellite and the ground station.

BIBLIOGRAPHY

[1] S. Waydo, D. Henry, and M. Campbell, “CubeSat Design for LEO-Based Earth Science Missions,” in *Proceedings 2002 IEEE Aerospace Conference*, vol. 1, Big Sky, MT, March 2002, pp. 435–445.

[2] California Polytechnic State University, “CubeSat Design Specifications,” http://waww.cubesat.org/images/developers/cds_rev13_final.pdf.

[3] C. Kilic, T. Scholz, and C. Asma, “Deployment Strategy Study of QB50 Network of CubeSats,” in *Proceedings* 6^{th} *International Conference on Recent Advances in Space Technologies – RAST 2013*, Istanbul, Turkey, June 2013, pp. 935–939.

[4] The QB50 Project, “An International Network of 50 CubeSats: Mission Objectives,” http://www.qb50.eu/index.php/project-description-obj/mission-objectives.

[5] A. Budianu, T. J. W. Castro, A. Meijerink, and M. J. Bentum, “Inter-Satellite Links for CubeSats,” in *Proceedings 2013 IEEE Aerospace Conference*, Big Sky, MT, March 2013, pp. 1–10.

[6] M. R. Maheshwarappa and C. P. Bridges, “Software Defined Radios for Small Satellites,” in *Proceedings* 9^{th} *NASA/ESA Conference on Adaptive Hardware and Systems – AHS 2014*, Leicester, United Kingdom, July 2014, pp. 172–179.

[7] S. J. Olivieri, J. Aarestad, L. Pollard Howard, A. M. Wyglinski, C. Kief, and R. Scott Erwin, “Modular FPGA-Based Software Defined Radio for CubeSats,” in *Proceedings 2012 IEEE International Conference on Communications – ICC’12*, Ottawa, Canada, June 2012, pp. 3229–3233.

[8] X.-N. Yang, J.-L. Xu, and C.-Y. Lou, “Software-Defined Satellite: A New Concept for Space Information System,” in *Proceedings Second International Conference on Instrumentation, Measurement, Computer, Communication and Control – IMCCC 2012*, Harbin, China, December 2012, pp. 586–589.

[9] G. Kolumban, T. I. Krebesz, and F. C. M. Lau, “Theory and Application of Software Defined Electronics: Design Concepts for the Next Generation of Telecommunications and Measurement Systems,” *IEEE Circuits and Systems Magazine*, vol. 12, no. 2, pp. 8–34, Second quarter 2012.

[10] W. H. W. Tuttlebee, “Software Defined Radio: Facets of a Developing Technology,” *IEEE Personal Communications Magazine*, vol. 6, no. 2, pp. 13–18, April 1999.

[11] F. K. Jondral, “Software-Defined Radio – Basics and Evolution to Cognitive Radio,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2005, no. 3, pp. 275–283, August 2005, special Issue on Reconfigurable Radio for Future Generation Wireless Systems.

[12] I. Systems. (2016, 5) Cubesat_1. Interorbital Systems. [Online]. Available: http://www.interorbital.com/interorbital_06222015_006.htm

[13] Planel Labs Inc. (2016, 1) Press - Planet Labs. Planet Labs Inc. [Online]. Available: https://www.planet.com/

[14] Spire. (2015, 3) Satellite based data for maritime and weather tracking — Spire Global. Spire. [Online]. Available: http://www.spire.com/

[15] (2016, 4) Spire Stratos - Satellite Weather Data — Spire. Spire Global, Inc. [Online]. Available: https://spire.com/products/stratos/

[16] H. Bedon, C. Negron, J. Llantoy, C. Nieto, and C. Asma, “Preliminary internetworking simulation of the qb50 cubesat constellation,” in *Communications (LATINCOM), 2010 IEEE Latin-American Conference on*, Sept 2010, pp. 1–6.

[17] (2013, 5) Space@VT to lead new NSF satellite study to improve communications — www.eng.vt.edu. College of Engineering, Virginia Tech. [Online]. Available: https://www.eng.vt.edu/news/spacevirginia-tech-lead-new-nsf-satellite-study-improve-communications

[18] Z. J. Leffke, “Distributed ground station network for cubesat communications,” Master’s thesis, Virginia Polytechnic Institute and State University, 12 2013.

[19] M. Parlak and R. J. McGlen, “Cooling of high power active phased array antenna using axially grooved heat pipe for a space application,” in *Recent Advances in Space Technologies (RAST), 2015 7th International Conference on*, June 2015, pp. 743–748.

[20] S. C. Rogers, “Radiation damage to satellite electronic systems,” *IEEE Transactions on Nuclear Science*, vol. 10, no. 1, pp. 97–105, Jan 1963.

[21] (2008, 10) PC/104 specification version 2.6. PC/104 Embedded Consortium. [Online]. Available: http://pc104.org/wp-content/uploads/2015/02/PC104_Spec_v2_6.pdf

[22] (2016, 4) PC/104 Intel Core I7 by RTS. RTD Embedded Technologies, Inc. [Online]. Available: http://www.rtd.com/i7/

[23] (2002, 12) AD9860/AD9862 (rev. 0). Analog Devices, Inc. [Online]. Available: http://www.analog.com/media/en/technical-documentation/data-sheets/AD9860_9862.pdf

[24] (2015, 3) Ettus Research - product detail. Ettus Research, A National Instruments Company. [Online]. Available: "http://www.ettus.com/product/details/UB200-KIT"

[25] Nuand, "bladeRF," http://www.nuand.com.

[26] Rice University, "WARP: Wireless Open-Access Research Platform," http://warp.rice.edu.

[27] (2016, 6) GCJ: The GNU Compiler for Java - GNU project - Free Software Foundation (FSF). Free Software Foundation. [Online]. Available: https://gcc.gnu.org/java/

[28] (2016, 4) Lapack – linear algebra package. Netlib. [Online]. Available: http://www.netlib.org/lapack/

[29] N. West. (2016, 4) Vector optimized library of kernels. [Online]. Available: http://libvolk.org/

[30] A. Csefe. (2016, 5) Gpredict: Free, real-time satellite tracking and orbit prediction software. [Online]. Available: http://gpredict.oz9aec.net/

[31] J. E. Prussing and B. A. Conway, *Orbital Mechanics*. Oxford University Press, 1993, p. 172.

[32] U. S. N. Observatory. (2016, 1) Approximate sidereal time. United States Naval Observatory. [Online]. Available: http://aa.usno.navy.mil/faq/docs/GAST.php

[33] (2016, 3) NASA - NSSDCA - spacecraft - details. National Aeronautics and Space Administration. [Online]. Available: http://nssdc.gsfc.nasa.gov/nmc/spacecraftDisplay.do?id=1997-051C

[34] (2016, 4) Microwave and Radar Institute - Tandem-X - a new high resolution interferometric sar mission. DLR. [Online]. Available: http://www.dlr.de/hr/en/desktopdefault.aspx/tabid-2317/3669_read-5488/

[35] W. J. Larson, J. R. Wertz, and (editors), *Space Mission Analysis and Design*, 3rd ed. Norwell, MA: Kluwer Academic Publishers, 2005.

[36] A. Goldsmith, *Wireless Communications.* New York, NY: Cambridge University Press, 2005.

[37] Q. Liu, "Doppler Measurement and Compensation in Mobile Satellite Communications Systems," in *Proceedings 1999 IEEE Military Communications Conference – MILCOM*, vol. 1, Atlantic City, NJ, November 1999, pp. 316–320.

[38] S. C. Ekpo and D. George, "A Systems Engineering Analysis of Highly Adaptive Small Satellites," *IEEE Systems Journal*, vol. 7, no. 4, pp. 642–648, December 2013.

[39] R. A. Khalona, "Performance of M-ary FSK Modulation in a Land Mobile Satellite Communication Channel," in *Proceedings Third Annual IEEE International Conference on Universal Personal Communications – ICUPC 1994*, San Diego, CA, September 1994, pp. 430–434.

[40] *FSK Modulation and Demodulation with the MSP430 Microcontroller*, Texas Instruments, 12 1998. [Online]. Available: http://www.ti.com/lit/an/slaa037/slaa037.pdf

[41] Ettus Research, "Universal Software Radio Peripheral," http://www.ettus.com/site/about.

[42] T. B. Welch and S. Shearman, "Teaching Software Defined Radio Using the USRP and Labview," in *Proceedings 2012 IEEE International Conference on Acoustics, Speech, and Signal Processing – ICASSP 2012*, Kyoto, Japan, March 2012, pp. 2789–2792.

[43] Mango Communications, "Products," http://mangocomm.com/products.

[44] (2016, 4) USRP x310 high performance software defined radio (sdr) - Ettus Research. Ettus Research, A National Instruments Company. [Online]. Available: "https://www.ettus.com/product/details/X310-KIT"

[45] GNU Radio, "Documentation," http://www.gnuradio.org.

[46] “The Comprehensive GNU Radio Archive Network,” http://www.cgran.org.

[47] A. S. Sedra and K. C. Smith, *Microelectronic Circuits.* Oxford University Press, 2010, p. 296.

[48] Mitsubishi Electric, “RD15HVF1,” 11 2011.

[49] M. D. Sanctis, E. Cianca, G. Araniti, I. Bisio, and R. Prasad, “Satellite communications supporting internet of remote things,” *IEEE Internet of Things Journal*, vol. 3, no. 1, pp. 113–123, Feb 2016.

[50] T. R. Dean and S. S. Hamilton, “Information dissemination over low-bandwidth links,” in *Systems, Applications and Technology Conference (LISAT), 2011 IEEE Long Island*, May 2011, pp. 1–5.

[51] S. M. Dilek, A. Ayranci, A. eker, O. Ceylan, and H. B. Yagci, “Ax.25 protocol compatible reconfigurable 2/4 fsk modulator design for nano/micro-satellites,” in *Telecommunications Forum (TELFOR), 2012 20th*, Nov 2012, pp. 416–419.

[52] J. G. Proakis and M. Salehi, *Fundamentals of Communications Systems.* Upper Saddle River, New Jersey 07458: Pearson Education, Inc, 2005.

[53] E. Lutz, “Modelling of the satellite communications channel; with emphasis on the land mobile satellite channel,” in *Satellite Telecommunications (ESTEL), 2012 IEEE First AESS European Conference on*, Oct 2012, pp. 1–7.

VITA

Jason S. Harris
Department of Electrical and Computer Engineering
Old Dominion University
Norfolk, VA 23529

EDUCATION:
B.S.E.E Fall 2014
Old Dominion University
Norfolk, VA

PUBLICATIONS:
Discharge characteristics of lithium-polymer batteries
Jason Harris, Dimitrie C. Popescu
IEEE SOUTHEASTCON 2014

An Experimental Study of InGaAs Photovoltaic Cell to Solar Exposure in Outer Space Environment
Jason S. Harris, Dimitrie C. Popescu, Christopher G. Bailey
IEEE SOUTHEASTCON 2016

Designing the Communication Sub-System for Nanosatellite CubeSat Missions:
Operational and Implementation Perspectives
Otilia Popescu, Jason S. Harris, Dimitrie C. Popescu
IEEE SOUTHEASTCON 2016