A Jam-Resistant CubeSat Communications Architecture

by

Caleb Kevin Ziegler

B.S. Electrical Engineering, United States Air Force Academy, 2015

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Masters of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2017

Author ..... Signature redacted
Department of Aeronautics and Astronautics
May 25, 2017

Certified by ..... Signature redacted
Kerri Cahoy
Associate Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by ..... Signature redacted
Youssef M. Marzouk
Associate Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

A Jam-Resistant CubeSat Communications Architecture

by

Caleb Kevin Ziegler

Submitted to the Department of Aeronautics and Astronautics
on May 25, 2017, in partial fulfillment of the
requirements for the degree of
Masters of Science in Aeronautics and Astronautics

Abstract

This thesis proposes a communications system that utilizes the benefits of CubeSats to provide jam-resistant communications. The growth of CubeSats within educational communities has prompted their use in industry; both industry and academia have contributed towards making CubeSats much more capable. CubeSats can now perform many advanced missions, from technology demonstrations to Earth observation missions and science missions. Meanwhile, military satellite communications (MILSATCOM) continues to rely primarily on large, highly-capable satellites. CubeSats could augment MILSATCOM by providing many low-cost space terminals with short development times as a means to create a more robust communications suite.

The CubeSat communications architecture proposed in this thesis aims to support mobile users in hostile environments who need to relay information to a command center. Jam-resistant communications are achieved by performing ground-based beamforming (GBBF) on a radio-frequency (RF) uplink and relaying the information to a ground station via a laser communications (lasercom) downlink: each CubeSat acts as an element of a sparse antenna array. With the growth of free-space lasercom in the last decade, lasercom is now a reality on CubeSat-scale platforms. Lasercom systems have lower size, weight, and power (SWaP) compared to RF systems with similar data rates, making them a good fit for CubeSat platforms. GBBF is a special case of beamforming where each element of an antenna array relays its signal to a ground station for processing, minimizing complexity on the space terminal. Beamforming provides anti-jamming capabilites due to the spacings between elements in the array, also known as spatial diversity. This spatial diversity allows spatial filtering to occur, which modifies the array's radiation pattern to mitigate interference, add gain to the main lobe, or add multiple beams. The system is designed with the goal of minimizing cost and development time, and two ways of accomplishing this are by supporting currently fielded handheld RF transmitters and by utilizing a lasercom downlink which is being developed as part of the Nanosatellite Optical Downlink Experiment (NODE) in MIT's Space, Telecommunications, Astronomy, and Radiation Lab (STARLab).

This thesis builds on previous work done on the NODE project, specifically the waveform design for NODE. NODE is a 3U CubeSat demonstrating a lasercom downlink while in low Earth Orbit (LEO). NODE uses 200mW transmit power to obtain data rates from 8 Mbps to 80 Mbps. The Optical Communications Telescope Lab-

oratory (OCTL) at the Jet Propulsion Laboratory (JPL) and an amateur telescope will be used as optical ground stations. In order to send information to the ground station, NODE uses a waveform that provides forward error correction (FEC) and interleaving to mitigate channel effects. This thesis develops the channel coding, interleaving, modulation, and framing approach employed in the NODE waveform to provide error-free communications. A Reed-Solomon code, selected because of its performance and the existence of open-source implementations, provides error-correction capabilities. NODE uses a one-second interleaver to combat the effects of channel fading when the laser beam passes through the atmosphere. The transmitter uses pulse position modulation (PPM), an intensity modulation scheme that uses the delay of a single pulse within a symbol time to transmit information, due to the advantages in using a duty-cycled waveform with an average-power limited optical amplifier. Since the delay of the pulse conveys information for PPM, the transmitter clock must be recovered in order to properly demodulate the received waveform, and NODE uses inter-symbol guard times to encode the transmitter clock onto the waveform. Python simulations are presented showing that the channel coding, interleaving, and modulation are sufficient to obtain error-free communications with a target channel bit error rate (BER) of 1×10^{-4}. The modulator is implemented within a field programmable gate array (FPGA), and the design, validation, and testing of the modulator are described.

The feasibility of performing GBBF on RF uplinks to CubeSats in LEO, where each CubeSat acts as an element of an adaptive array, is examined. The high Doppler and large spacing between CubeSats requires the use of a space-time-frequency adaptive processor (STFAP). The STFAP consists of Doppler and delay taps, complex weights, an adaptive processor, a polyphase filter bank, and a polyphase combiner. The STFAP becomes infeasible as the Doppler and delay spread between different CubeSats increases, and analysis is used to identify scenarios where the Doppler and delay spreads seen in LEO are acceptable. Systems Tool Kit (STK) simulations are performed to analyze the Doppler and delay environment in LEO. Two CubeSat formations and multiple orientations between a user and jammer are examined to determine cases where null-forming, a special case of beamforming, is effective. A constellation is necessary to provide global coverage and maximize the effectiveness of null-forming, and two possible constellations are discussed.

Thesis Supervisor: Kerri Cahoy
Title: Associate Professor of Aeronautics and Astronautics

Acknowledgments

I would like to thank my advisor, Kerri Cahoy, for her tireless devotion to her students. She is a continual source of support, and her work ethic never ceases to amaze me.

I would like to thank my Lincoln Laboratory mentor, Brian Wolf, for guiding me every step of the way. His insights were a driving force in determining the structure of this thesis, and his mentorship was crucial in the completion of this thesis. I cannot thank Brian enough for all of the work he did supporting my graduate studies and reviewing my thesis. Additionally, I would like to thank Fred Block, Tommy Royster, and Joseph Gaeddert for discussing all aspects of satellite communications and waveform design with me. Ameya Agaskar played a huge role in explaining the fundamentals of adaptive arrays and ground-based beamforming to me, and his insights were pivotal in building chapter 4. Matthew Johnson provided hardware that enabled the completion of tests. Dave Caplan and Dave Geisler helped me understand the fundamentals of lasercom. Furthermore, I would like to thank Lincoln Laboratory Group 64, Thomas MacDonald, and Mr. Kuconis for giving me the opportunity to study at MIT and learn from the best at Lincoln Labs.

I'm grateful for being able to work with brilliant people and make great friendships during my graduate studies. Telling jokes with Linda always brought a smile to my face. Solving the world's problems with Dave and Mike provided great study breaks and more than a few laughs. Working with Rodrigo, Myron, and Joe helped me understand more about FPGAs, software development, and clock recovery. Kat and Pratik helped me get adjusted to graduate school and were the source of many interesting conversations. Moreover, I would like to thank all other members of the NODE team.

I can never give enough thanks to my family. I've been blessed with the most loving and encouraging parents, and I am eternally grateful for everything they have given me. They taught me what it means to be disciplined and shoot for the stars, and I would not be in the position I am in today if not for them. My brothers Kyle and Craig are a continual source of support, and I cherish our ability to stay close despite the distance between us.

My fiancée Megan is a blessing beyond compare. Her compassion and kindness are without measure, making her such a valuable source of support. She has cheered me on throughout my graduate studies, keeping me focused and motivated. I cannot wait to see what the future holds for us. To Betelgeuse and back.

Contents

List of Figures

List of Tables

Nomenclature

ADC	Analog-to-Digital Converter
ADCS	Attitude Determination and Control Systems
APD	Avalanche Photodiode
ASCII	American Standard Code for Information Interchange
BER	Bit Error Rate
BIST	Built-In-Self-Test
BPSK	Binary Phase Shift Keying
CINR	Carrier-to-Interference-plus-Noise Ratio
COTS	Commerical-Off-The-Shelf
D^4	Differential-Delay Differential-Doppler
DBF	Digital Beamforming
DoD	Department of Defense
DPSK	Differential-Phase-Shift-Keying
DSP	Digital Signal Processing
DSSS	Direct-Sequence Spread Spectrum
DVB-S2	Digital Video Broadcasting Second Generation
ECC	Error-Correcting Codes
EDFA	Erbium-Doped Fiber Amplifier
EIRP	Effective Isotropically Radiated Power
FBG	Fiber Bragg Grating
FEC	Forward Error Correction

FHSS	Frequency-Hopping Spread Spectrum
FIFO	First In, First Out
FPGA	Field Programmable Gate Array
FSK	Frequency-Shift-Keying
FSM	Fast Steering Mirror
FSO	Free Space Optical
GBBF	Ground-Based Beamforming
GEO	Geosynchronous Earth Orbit
HPBW	Half-Power Beamwidth
IM	Intensity Modulation
IR	Infrared
ISGT	Inter-Symbol Guard Time
JPL	Jet Propulsion Laboratory
LEO	Low Earth Orbit
LLCD	Lunar Laser Communications Demonstration
LLGT	Lunar Lasercom Ground Terminal
LLST	Lunar Lasercom Space Terminal
M-PPM	M-ary Pulse-Position Modulation
MILSATCOM	Military Satellite Communications
MMSE	Minimum Mean Square Error
MOPA	Master Oscillator Power Amplifier
NF	Noise Figure
NODE	Nanosatellite Optical Downlink Experiment
OCSD	Optical Communication and Sensor Demonstration
OCTL	Optical Communications Telescope Laboratory
OOK	On-Off Keying
OPALS	Optical Payload for Lasercomm Science

OTA	Over-The-Air
PPM	Pulse-Position Modulation
PSD	Power Spectral Density
PSL	Peak Side Lobe Level
QPSK	Quadrature Phase Shift Keying
RAAN	Right Ascension of the Ascending Node
RF	Radio-Frequency
RHCP	Right-Hand Circularly Polarized
RS	Reed-Solomon
SATCOM	Satellite Communications
SCPPM	Serially Concatenated PPM
SER	Symbol Error Rate
SINR	Signal-to-Interference-plus-Noise Ratio
SNR	Signal-to-Noise Ratio
SOF	Start of Frame
STARLab	Space, Telecommunications, Astronomy, and Radiation Laboratory
STFAP	Space-Time-Frequency Adaptive Processor
STK	Systems Tool Kit
SWaP	Size, Weight, and Power
TOSA	Transmitter Optical Sub-Assembly
UHF	Ultra High Frequency
VHDL	VHSIC Hardware Description Language
VOA	Variable Optical Attenuator

Chapter 1

Introduction

This thesis proposes a CubeSat communications system that provides jam-resistant communications. Arising from educational institutions, CubeSats are small satellites that provide short project lifecycles as a means to give students exposure to the spacecraft design cycle. However, CubeSats are becoming increasingly more prevalent in the commercial and military sector as they become more capable due to the miniaturization of components. Satellite communications (SATCOM) systems are primarily composed of large, highly-capable satellites in geosynchronous Earth orbit (GEO) and smaller, less capable satellites in low Earth orbit (LEO) that are owned by both private and government organizations. These satellites typically act as transponders to route information from one location to another. Many of these satellites, except for the more capable satellites in GEO, are susceptible to jamming as they do not employ interference suppression techniques. From a military standpoint, this is quite troubling, resulting in the Defence Advanced Research Projects Agency (DARPA) and the Department of Defense (DoD) requesting jam-resistant satellite communications systems [19, 79]. Due to their short project lifecycles, relatively low cost, and increasing capabilities, CubeSats could rise to fill the need for jam-resistant satellite communications. This chapter provides an overview of CubeSats, the SATCOM fleet, and jam-suppression techniques, ending with a discussion about jam-suppression technologies that match CubeSat design goals.

1.1 CubeSats

The term "CubeSat" refers to a class of small satellites that emerged from educational institutions. Originally coined by California Polytechnic State University (Cal Poly), San Luis Obispo, and Stanford University's Space Systems Development Laboratory (SSDL), CubeSats began as a way to reduce cost and development time while increasing accessibility to space and frequency of launches for small satellite programs [57]. CubeSats have extended beyond educational use as they are being used extensively in the commercial sector.

CubeSats are defined by their mass and form factor. CubeSats typically fall into the class of small satellites known as nanosatellites, which have a mass between 1-

10kg [33]. The CubeSat form factor is defined as a 10 cm x 10 cm x 10 cm cube, known as 1U, where multiple cubes may be combined to make larger spacecraft, such as 3U and 6U CubeSats [33].

When compared to larger, more conventional spacecraft, CubeSats have some advantages and disadvantages. As mentioned earlier, CubeSats have reduced size and development time compared to larger satellites. Their small size affords them more launch opportunities and reduces their launch costs. However, CubeSats are not as capable as larger satellites due to their limited size, weight, and power (SWaP). Also, CubeSat programs see limited mission lifetimes due to their limited propulsion and placement in LEO.

1.1.1 CubeSat Capabilities

The improvement of Attitude Determination and Control Systems (ADCS), power generation systems, small-scale propulsion systems, and communication systems have made CubeSats much more capable. Now, CubeSats perform Earth observation, conduct science experiments, test formation flight, and take part in technology demonstrations. Reaction wheels and star cameras matching the CubeSat form-factor have helped solve the ADCS problem by providing three-axis stabilization and 3° pointing accuracy [48, 70]. Triple-junction solar cells and lithium-ion batteries are providing 3U CubeSats with orbit average powers of approximately 10 W for standard solar power configurations and 20 W for more optimized solar panel configurations [22, 55]. Propulsion systems are enabling CubeSats to perform precise formation flight maneuvers, such as with CanX-4 and CanX-5 [53, 12, 8]. Furthermore, CubeSat radios have improved, with the custom X-band Planet radio providing downlinks up to 120 Mbps [49] and the commercial-off-the-shelf (COTS) L3 Cadet radio providing data rates up to 2.6 Mbps [50]. Despite the capabilites of CubeSats, to date, they have not been used to provide communications services.

1.2 Satellite Communications

While SATCOM is composed of LEO, MEO, and GEO satellites, military satellite communications (MILSATCOM) primarily relies on GEO satellites. As such, the MILSATCOM fleet lacks diversity across orbit altitudes that the commercial sector employs. Future DoD goals include using the commercial communications satellites to augment the MILSATCOM fleet [25], adding much needed diversity. A few examples of communications satellite constellations in LEO, which could be used to augment MILSATCOM, are Iridium and GlobalStar. Unfortunately, these systems may not be robust enough to be used extensively for MILSATCOM. The following subsections provide a brief overview of MILSATCOM and elaborate on the LEO communications constellations, Globalstar and Iridium. Finally, this section will end with a discussion showing that the inclusion of jam-resistant CubeSats in LEO may provide benefits for MILSATCOM.

1.2.1 MILSATCOM

The MILSATCOM fleet consists of very capable, large, and expensive communications satellites located in GEO. MILSATCOM systems are separated into three categories: wideband systems, narrowband (mobile/tactical) systems, and protected systems [46]. The wideband systems maximize capacity to provide large data rates. The Wideband Global SATCOM (WGS) system and Defense Satellite Communications System (DSCS) account for the wideband services, with WGS accounting for most of the capacity. The narrowband systems target mobile users and small ground terminals, such as vehicle, backpack, and handheld radios. The Ultra High Frequency (UHF) Follow-On (UFO) system and Mobile User Objective System (MUOS) provide the narrowband services for the DoD. The protected systems feature anti-jamming technologies, at the cost of decreased data rates, to provide reliable communications in poor, degraded environments. The Milstar and Advanced Extremely High Frequency (AEHF) compose the protected systems fleet.

The MILSATCOM fleet has some capability gaps that are not obvious. Due to the demand for communications services to mobile units, the narrowband systems tend to be oversubscribed [32]. Additionally, even though the narrowband systems were built to support handheld terminals, no handheld terminals have been developed [36, 65]. This poses a problem as the forward deployed warfighter may not get access to MILSATCOM services. Also, narrowband SATCOM does not employ significant anti-jamming technologies [32], putting tactical, mobile units at higher risk of losing communications. While the protected systems do support tactical users [32], the literature does not suggest that the protected systems support mobile users through handheld terminals. One desired capability for the MILSATCOM fleet is to be able to provide reliable communications to mobile, forward deployed units.

While MILSATCOM is very capable, building a fleet of large satellites located in GEO has its disadvantages. All of the MILSATCOM systems are billion dollar projects, and due to their size and cost, they take a very long time to go from concept to launch [14]. While in orbit, these satellites represent single points of failure, and a loss of a satellite due to adversarial actions or natural phenomenon would result in a loss of coverage in a portion of the globe or decreased ability to perform cross-links, as is the case for Milstar [32]. Additionally, due to rapidly developing technology, these systems tend to get dated, and once they do, it takes a significant amount of time to develop a more advanced satellite to fill its role.

1.2.2 LEO Missions

The Iridium and Globalstar LEO constellations provide global telecommunications services, with the Iridium constellation[1] consisting of 66 satellites and the Globalstar first and second generation fleets consisting of 48 and 24 satellites, respectively. Due to the short pass times in LEO, a constellation is required to provide global coverage. The Iridium and Globalstar satellites have the ability to support many users due to

[1]The Iridium NEXT constellation is being deployed during the writing of this thesis, so it will not be covered.

the multiple access schemes and frequency reuse they employ [69, 56, 28]. Despite their age, the Iridium[2] and Globalstar constellations are capable of augmenting the narrowband MILSATCOM systems.

Being in LEO, Iridium and Globalstar have some benefits over the current narrowband MILSATCOM systems. Due to the shorter link ranges, Iridium and Globalstar see decreased propagation delays and decreased free space path losses when compared to the MILSATCOM systems. Because of the decreased path losses, Iridium and Globalstar are able to support handheld devices while using space terminals with antennas that are much smaller than the 14 m mesh antenna used on MUOS, the newest narrowband MILSATCOM system [69, 65, 36]. Also, Iridium and Globalstar are more resilient to losses than MILSATCOM systems; a loss of a single satellite would result in a slight loss of capacity or coverage, while a loss of a MILSATCOM satellite could result in a significant loss of coverage [36]. All in all, the benefits of LEO communications systems make them a great candidate for augmenting the MILSATCOM fleet.

While LEO communications systems provide great benefits for MILSATCOM, Iridium and Globalstar were not designed with the notion of supporting military operations. Both systems were not designed to operate in the presence of jamming [9]. Additionally, Iridium and Globalstar have radically different network architectures, which would cause problems with interoperability as users on one system may not be able to communicate with users on the other system (or other MILSATCOM systems) [9, 69]. These problems make it tough for the military to rely on either system, meaning the DoD will have to field their own MILSATCOM system in LEO. The cost of such an endeavor could be just as expensive as fielding previous MILSATCOM systems since the Iridium and Globalstar systems cost 7 billion and 2.2 billion dollars, respectively [20]. Even though fielding a MILSATCOM system in LEO could be expensive, a LEO communications system would address the gaps in the MILSATCOM fleet, and CubeSats could be used to reduce cost.

1.2.3 CubeSats for Communications

The growth of CubeSat technology has made CubeSats capable of providing telecommunications services while offering a reduction in cost and development time when compared to traditional LEO communication systems. CubeSat communication systems have sufficient capacity to support multiple channels of voice data, but the capacity of CubeSat radio-frequency (RF) communication systems is worse than that of traditional LEO satellite communication systems due to the SWaP constraints placed on CubeSats. However, the low SWaP of CubeSats makes them a very cost-effective option. The total cost, including development, integration, and launch costs, of a single CubeSat typically falls between $200,000 and $2 million [84]. At that cost, a constellation of 66 CubeSats placed in a similar manner as Iridium would cost less than $132 million, which is an order of magnitude improvement over the cost of both the Iridium and Globalstar systems. Due to their low complexity and low

[2]The Iridium system even offers a special secure service for the military[69]

SWaP, CubeSats have shorter development times and see an increased amount of launch opportunities, at a reduced cost, when compared to larger satellites. Shorter development times and increased launch opportunities allow for the rapid deployment of spares and updated platforms. A constellation of CubeSats in LEO could provide communications services around the globe with a reduced cost and shorter development time than traditional LEO communication systems at the expense of capacity. The benefits of reduced cost and shorter development time may outweigh the reduction in capacity. For that reason, this thesis explores the use of a CubeSat communications system in LEO intended to provide jam-resistant communications to forward deployed soldiers using handheld radios.

1.3 Anti-Jamming Techniques

As mentioned in Section 1.2.1, MILSATCOM is in need of jam-resistant communications to mobile, tactical units, and fortunately, anti-jamming techniques may be employed by the proposed LEO CubeSat communications system. Anti-jamming techniques are electronic countermeasures that are used to enable communications in the face of intentional interference, also known as jamming. As said in [73], the goal of a jammer is to deny communications to an enemy at minimum cost under the following assumptions:

- The jammer has *a priori* knowledge of the system parameters, with the exception of codes or keys used in spread spectrum techniques.
- The signaling waveform is designed in such a way so that the jammer cannot gain disproportionate advantages by choosing a jamming waveform other than wideband Gaussian noise.

With the knowledge that complete invulnerability to jamming is impossible, the goal in designing a jam-resistant system is to make it as costly as possible for the jammer while minimizing cost for the user [73]. In designing a jam-resistant system for CubeSats, three common techniques may be employed: spread spectrum, beamforming, and laser communications.

1.3.1 Spread Spectrum

Spread spectrum techniques aim to mitigate interference by spreading a signal over a large bandwidth, and the two most common spread spectrum techniques are direct-sequence spread spectrum and frequency-hopping spread spectrum.

In direct-sequence spread spectrum (DSSS), the transmitted is spread among a much larger bandwidth than that required by the information signal through the use of a spreading signal or code signal [73]. The received signal is despread with the same spreading signal, and when in the presence of narrowband interference, the despreading spreads the interfering signal among a much larger bandwidth while recovering the narrow bandwidth of the transmitted signal, thereby decreasing the

power spectral density (PSD) of the interference. Reducing the interferer's received PSD works to improve the signal-to-interference-plus-noise ratio (SINR)[3], thereby enabling successful communications. However, the interferer's received PSD after despreading will result in a slight degradation in SINR, reducing some of the link margin.

In frequency-hopping spread spectrum (FHSS), the carrier frequency is moved to various frequencies, or channels, within the spread spectrum bandwidth, and the choice of frequency to switch to is provided by a pseudorandom sequence. Put another way, the data modulates a carrier whose frequency is pseudorandomly determined within some hopping bandwidth [73]. Assuming that the interference bandwidth is less than the hopping bandwidth, the signal is only lost when the intended signal hops to a portion of the band occupied by the interferer. Alternatively, if the interferer spreads their signal over the full bandwidth without increasing their transmit power, they reduce their PSD, limiting the effectiveness of jamming. By hopping to various frequencies pseudorandomly within some hopping bandwidth, the transmitted signal is very rarely jammed or avoids the interfering bandwidth all together. For the purposes of this thesis, DSSS and FHSS are considered to be roughly equivalent in their abilities to mitigate interference. The goal of a jammer is to maximize its received PSD within the bandwidth of interest. Given a finite amount of power, the jammer must spread the signal across a large bandwidth, accepting a low PSD, or a narrow bandwidth, gaining a large PSD. Spread spectrum techniques take advantage of the finite power in a jammer and aim to force the jammer to spread its power over a large bandwidth, reducing its PSD. In short, spread spectrum techniques use a larger amount of bandwidth as means to mitigate interference.

Spread spectrum techniques come with some advantages and disadvantages. They are widely used, and for that reason, they are fairly straightforward in implementation. As mentioned earlier, they can effectively mitigate narrowband interference. Unfortunately, this comes with some drawbacks. Spread spectrum techniques require a much larger bandwidth than the bandwidth needed for the message signal, and this can create difficulties for frequency licensing. DSSS and FHSS require additional complexity in the receiver and pose a constraint on the transmitter when compared to a system not employing either technique.

1.3.2 Beamforming

Beamforming is commonly referred to as spatial filtering. By spreading elements of an antenna array in space, the array gains spatial diversity, giving the ability to perform spatial filtering. Beamforming attempts to control the way the signals at various elements of the array constructively and destructively interfere, thereby adjusting the array's radiation pattern. Additional gain may be placed in the direction of the transmitter or receiver, or a null may be placed in the direction of an interference source. By adding gain to the main lobe of the array's radiation pattern and forming nulls

[3]SINR is used analogously in this thesis with the carrier-to-interference-plus-noise ratio (CINR), denoted as $C/(N_o + I_o)$

in the radiation pattern, a jammer's interference may be suppressed very effectively. Whereas spread spectrum techniques require an increase in bandwidth in order to achieve interference suppression, beamforming requires an antenna array distributed in space where the differences in time delay, frequency, and phase between signals at different elements in the array may be adjusted.

Ground-based beamforming (GBBF) is a special case of beamforming in which the data received from each element of the array is sent to a ground station, where the signals are processed and the information is extracted. GBBF is advantageous compared to traditional beamforming because the data can be processed on the ground, where size, weight, power, and computation resources are not limited.

Beamforming comes with some pros and cons. Since the beamformer forms a null due to spatial diversity, the beamformer can mitigate both wideband and narrowband interference, while wideband interference could cause problems with spread spectrum techniques. In contrast to spread spectrum, a beamformer does not require additional bandwidth. When a jammer is not present, the beamformer can place additional gain on the main lobe of the array radiation pattern, as shown in Equation 4.4, to gain excess link margin or obtain higher data rates. Ground-based beamforming allows a low-complexity transmitter and receiver to be used as all of the processing is pushed to the ground station. However, ground-based beamforming does come with its drawbacks. It is not a very mature technology, and the ground processing can be quite complex. A distributed array, such as the one proposed in this thesis, sees degraded performance in high-delay and high-Doppler environments, as will be explained in Chapter 4.

1.3.3 Laser Communications

Laser communications (lasercom) affords jam-resistant communications due to the use of very narrow beamwidths, providing high directivity, on transmit and receive. Due to the narrow beamwidths, most lasercom systems require the use of a beacon signal to locate and stay locked onto the receiver, and such a receive architecture may be exploited by a jammer. A lasercom system may be jammed in two ways:

1. An adversary could transmit a false beacon to prevent the transmitter from pointing at the receiver.

2. An adversary could point toward the receiver, while staying within the field of view of the receiver, and attempt to overpower the user's signal.

The narrow field of view of the optical receiver would require that a false beacon be located very close to the receiver or that the jammer be located very close to the transmitter. A jammer would need to be located very close to an Earth-based terminal to be effective, and this could be very difficult since the jammer could be denied access to the area around the terminal. Alternatively, a jammer would need to be located very close to a space-based terminal to be effective, and launching a spacecraft capable of jamming a lasercom system would take a significant amount of

time and resources. In short, lasercom is a robust anti-jamming technology because reliably jamming a lasercom system is very difficult with limited time and resources.

Lasercom can be used on uplink or downlink; however, it is not well suited for mobile, forward-deployed units. Due to the narrow beamwidths, lasercom systems have very tight pointing requirements, requiring that the terminals remain in a fixed location for the communications window or be on an inertially stabilized platform. Fielding an inertially stabilized lasercom transmitter or receiver for mobile, forward-deployed units is by no means a simple task. However, lasercom systems have demonstrated downlink and uplink capabilities between a satellite and a fixed optical ground station, as is shown in Section 1.4.2. For these reasons, this thesis only discusses the use of lasercom as a data backhaul between a spacecraft and a fixed ground station and does not envision a lasercom uplink from a mobile user.

In addition to providing jam-resistant communications, lasercom offers benefits in SWaP and data rate. As explained in [48], lasercom systems can have lower SWaP than RF systems with similar data rates, which naturally yields itself to low SWaP CubeSat platforms. Lasercom systems come with a few drawbacks despite these advantages. Lasercom systems are a new technology for CubeSats, meaning they accept additional risk. Lasercom fails when attempting to transmit through thick clouds, which can reduce the reliability of the system as a whole.

1.3.4 Design Choice

Ground-based beamforming and lasercom were chosen as the secure communications methods for the communications system proposed in this thesis. The use of both methods allows low-cost user terminals and low-SWaP CubeSats to be used as RF-to-optical relays while providing interference suppression benefits without requiring a large amount of RF bandwidth. Ground-based beamforming will be used for interference suppression on the RF uplink, and lasercom will be used for the downlink. Ground-based beamforming allows a very simple RF front-end to be used on the CubeSat, and lasercom provides sufficient data rates in order to transmit the sampled data securely to the ground station.

1.4 Lasercom

Lasercom, also known as optical communications, involves using carrier frequencies within the infrared (IR), visible, or ultraviolet (UV) spectra to send information. Due to the very high frequencies of the IR, visible, and UV spectra, large gains may be obtained with small apertures when using lasercom. The laser beams emitted from these small apertures are very directive, and for that reason, pointing them at the receiver can be a significant challenge. Fortunately, this is a solved problem for large optical terminals and is discussed in [48, 70] for small optical terminals. The high directivity of the beam results in a higher link efficiency as a higher percentage of the transmitted energy reaches the receiver compared to systems with low directivity. Also, the high directivity of the beam allows for more secure, jam-resistant commu-

nications as discussed in Section 1.3.3. Terahertz (THz) of unregulated frequency is available in the optical bands, allowing optical communications systems to use much more bandwidth than is available at RF. Compared to RF systems, optical carriers allow for a decreased aperture size, increased directivity, increased gain, and increased bandwidth. These benefits allow optical communications systems to reach data rates greater than those provided by current RF systems while reducing SWaP for transmit and receive terminals [13].

1.4.1 Direct Detection

Direct detection lasercom systems use intensity modulation (IM) schemes to encode information on a laser carrier, and the receiver only records the intensity of the signal, meaning no phase or polarization information is recovered. Such schemes are simple to implement, but they see a reduction in receiver sensitivity for the same amount of bandwidth compared to other more complex schemes, such as Differential-Phase-Shift-Keying (DPSK) [16]. However, direct detection systems are more mature than other lasercom systems due to the efforts made by the fiber-optic communications industry [40]. On-Off Keying (OOK) and Pulse-Position Modulation (PPM) are two intensity modulation formats that are commonly used for direct-detection systems, while DPSK, Phase-Shift-Keying (PSK), Frequency-Shift-Keying (FSK), and Polarization-Shift-Keying (PolSK) are modulation schemes that require more complex transmit and receive topologies, such as homodyne or heterodyne receivers [16]. Due to the simplicity and maturity of direct detection systems, this thesis focuses solely on direct detection systems. A more thorough discussion of DPSK and FSK may be found [16, 40, 41], and the design, implementation, and performance of systems using DPSK and FSK may be found in [17, 15].

1.4.2 Literature Review

Lasercom downlinks have been demonstrated successfully, and two such examples are the Lunar Laser Communications Demonstration (LLCD) and the Optical Payload for Lasercomm Science (OPALS). Additionally, lasercom downlinks are being incorporated into CubeSats, and Aerospace Corporation's AeroCube Optical Communication and Sensor Demonstration (OCSD) was the first CubeSat-scale optical terminal to make it into orbit, although the first unit experienced an anomaly that prevented the payload from being operated for validation [44]. Other lasercom downlinks are the German Aerospace Center's (DLR) Optical Space Infrared Downlink System (OSIRIS) [71] and the Small Optical TrAnsponder (SOTA) designed by Japan's National Institute of Information and Communications Technology (NICT) [66, 51], with both systems currently in orbit. Lasercom has also attracted the interest of the startups BridgeSat [2] and "analyticalspace" [1], which are designing space-based lasercom transmitters and networks of lasercom ground stations.

LLCD demonstrated laser communications from lunar orbit aboard the Lunar Atmosphere and Dust Environment Explorer (LADEE). Through development of a custom space terminal, the Lunar Lasercom Space Terminal (LLST), and a custom

ground terminal, the Lunar Lasercom Ground Terminal (LLGT), LLCD obtained downlink data rates up to 622 Mbps [13]. To obtain these high data rates, the LLST used a 0.5 W, 15 μrad downlink beam transmitted from a 10 cm aperture, and the LLGT used an array of four 40 cm telescopes coupled into an array of superconducting nanowire single photon detectors. The data rates obtained by LLCD are six times faster than previous RF communications to the moon, and the LLST is half the weight and uses a quarter of the power compared to an equivalent RF system [81]. As far as lasercom demonstrations are concerned, LLCD represents a state-of-the-art lasercom downlink for a large space terminal.

While LLCD represents a large space terminal, OPALS represents a small lasercom downlink terminal. OPALS demonstrated successful laser communications while aboard the International Space Station (ISS) in LEO. Using commercial off-the-shelf (COTS) components, OPALS completed eighteen successful links with data rates from 33-50 Mb/s [10, 64]. OPALS obtained these data rates by using a 1.08 mrad, 1.178 W beam on the space terminal and a commerical avalanche photodiode (APD) on the ground system, which used the Optical Communication Telescope Laboratory (OCTL) at the Jet Propulsion Laboratory (JPL) [64].

Aerospace Corporation's AeroCube OCSD takes full advantage of the low SWaP offered by lasercom terminals by designing a lasercom space terminal that fits inside of a 1.5U CubeSat. AeroCube OCSD is attempting to obtain data rates between 5 and 50 Mbps on downlink with an optical output power of 10 W with a full-width-half-maximum (FWHM) beamwidth of 0.35° [81]. While LLCD and OPALS used gimbaled designs to point the laser beam, AeroCube OCSD relies on the spacecraft body pointing to point the laser beam at the optical ground station [81]. AeroCube OCSD will downlink to their optical ground station located at Mt. Wilson, California. AeroCube OCSD-A, the first of three CubeSats in the OCSD program, was launched in October of 2015, but unfortunately, the spacecraft experienced an "upload anomaly" rendering the attitude control processor dead, prohibiting any lasercom demonstrations [44]. With the next two CubeSats, AeroCube OCSD will more than likely successfully demonstrate laser communications on a CubeSat platform.

This thesis uses and builds upon the lasercom system being designed in MIT's Space, Telecommunications, Astronomy, and Radiation Laboratory (STARLab) known as the Nanosatellite Optical Downlink Experiment (NODE) [48, 70, 62, 23]. For the purposes of this thesis, NODE will be used in the proposed CubeSat communications system on downlink as a way to relay information to an optical ground station, providing jam-resistant communications with sufficient capacity. NODE will be explained in more detail in Chapter 2.

1.5 Beamforming

As mentioned in Section 1.3.2, beamforming uses the spatial diversity within an array of antennas to perform spatial filtering. Spatial filtering may be viewed as adding gain or losses (nulls) to the array pattern in some predetermined manner to affect the transmission and reception of electromagnetic waves. In traditional methods,

beamforming is accomplished by phased arrays, which are antenna arrays with a very specific, predetermined spacing between elements and a variable phase shift associated with each element of the array. On transmit and receive, a phased array can shape the array's radiation pattern and steer the main lobe or nulls in a direction of interest by adjusting the phase shifts. Such a method is very popular because the array radiation pattern may be adjusted and steered electronically by changing the phase shifts rather than performing mechanical adjustments. A more thorough discussion of phased arrays may be found in [76, 78].

The development of digital signal processing (DSP) technology has brought about digtal beamforming (DBF), where the output of each element of the array is sampled and passed to a digital signal processor, which performs beamforming. This lies in contrast to the analog phase shifts and analog combining employed in a phased array. In the digital signal processor, complex weights are applied to the digital signals from each element of the array in order to adjust their amplitudes and phases, but in order for this to work properly, the complete heterodyne receivers at each element of the array must be closely matched in amplitude and phase [54]. Rather than adjusting the hardware, a calibration process can be performed on the digital data in the digital signal processor before beamforming [54], and as will be shown in Chapter 4, this process grows in complexity when the elements of the array are given large spacings and large velocities. While synchronizing the signals can be a complexity with DBF, DBF has some advantages over conventional phased arrays. DBFs can form multiple independent beams without degrading the SNR, have access to all of the information arriving at the antenna array for signal processing, and may suppress interference through applying adaptive beamforming, allowing the implementation of any algorithm expressed in mathematical form [54].

Adaptive beamforming follows from DBF as adaptive beamforming sets the complex weights in order to maximize or minimize some value. By performing optimization, adaptive arrays[4] may "adapt" to their environment [54], making them suitable for mitigating interference or assuring performance in a changing environment. Figure 1-1 shows a diagram of an adaptive array. In Figure 1-1, the beamforming network would be implemented in a digital signal processor, and the adaptive array would be implemented in a processor. The sensor array would contain the antennas, complex heterodyne receivers, and analog-to-digital converters (ADC) for each element of the array. The digital data would be processed in the beamforming network in the same manner as with a DBF. However, the adaptive processor would set the complex weights in the beamforming network in order to maximize or minimize the value of choice in the adaptive algorithm. In order to set the complex weights optimally, the adaptive array incorporates feedback from the array output. The goal of an adaptive array is to improve the reception of a desired signal while in the presence of undesired interference [60]. One advantage to adaptive arrays is that the location of the user and interference source are not needed, and depending on the adaptive algorithm used, the location of the user and interference source may be estimated from training data or other known characteristics of the user's signal that separate it from the interference

[4]To resolve ambiguities, adaptive arrays perform adaptive beamforming

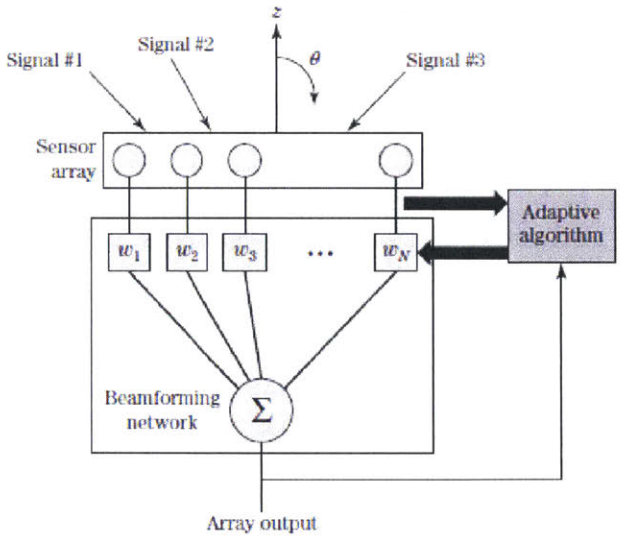

Figure 1-1: Diagram of an Adaptive Array [60]

[60]. Many different adaptive algorithms exist, and they differ in either the criteria they attempt to optimize or the approach taken to perform the optimization. Adaptive algorithms may attempt to minimize the mean square error between the received signal and a reference signal (training data), minimize the interference, maximize the SNR, or maximize the SINR, to name a few [60, 54, 11]. A few adaptive algorithms commonly used are the least mean squares algorithm, the direct sample covariance matrix inversion, and the recursive least squares algorithm [54, 60]. In Chapter 4, an adaptive beamformer will be described that uses the minimum mean square error (MMSE) criteria while applying direct sample covariance matrix inversion. A further description of adaptive arrays may be found in [54, 60, 11].

1.5.1 Ground-Based Beamforming

As mentioned earlier, GBBF refers to beamforming where the array data is transmitted from each element of an antenna array to a ground station that performs beamforming on the received data. Therefore, GBBF uses the same algorithms that adaptive arrays use. GBBF yields itself to SWaP constrained CubeSats since the processing is not performed on board the satellite. For the purposes of this thesis, GBBF will be performed on uplink only. A short description of GBBF may be found in [77].

1.5.2 Literature Review

The use of GBBF in communications links has been demonstrated successfully. The US Tracking and Data Relay Satellite Systems (TDRSS), which was launched in 1983, uses GBBF in S-band to track multiple LEO satellites at the same time [34, 7]. Each

TDRS contains 30 small antennas that can receive signals, and the signals from these antennas are sent to the ground to perform GBBF [42]. Designed and built by Space Systems/Loral, the DBSD G1 Mobile Satellite System is the first satellite to use a two-way GBBF system[80]. The DBSD G1 satellite has an S-band array for GBBF on receive and on transmit, which consists of a 12 m reflector and a 46 element feed array [80]. Terrestar and SkyTerra have begun employing GBBF techniques [34, 77], but their designs and results are absent from the literature. For all of these cases, GBBF has been performed on large satellites in GEO. GBBF has yet to be performed on small satellites in LEO.

1.6 Thesis Structure

This thesis is structured as follows. Chapter 2 provides the system architecture, which includes the formation of CubeSats acting as elements in the GBBF array and relaying the array data to the ground via a lasercom downlink. Link budgets are shown to demonstrate the feasibility of such an architecture. Chapter 3 discusses the waveform design for NODE, which includes the modulation, symbol mapping, channel coding, interleaving, and framing. Additionally, the results of waveform design simulations will be shown, and Chapter 3 concludes with the design, validation, and testing of the modulator. Chapter 4 introduces the GBBF system and shows how formations of CubeSats in a LEO constellation can provide jam suppression capabilities. The performance of GBBF with respect to interference suppression is examined for two CubeSat formations and multiple orientations between a user and jammer. Chapter 5 summarizes the thesis and identifies areas of future work.

Chapter 2

System Design

The proposed system combines the benefits of CubeSats, GBBF, and lasercom to achieve secure, affordable, compact, and capable communications. Moving the majority of the RF signal processing to the ground enables the use of low-cost, low-complexity CubeSats that act as RF-to-optical relays. Additionally, this allows a user with a handheld radio in an isolated or hostile environment to access the communications system and send information back to a command center, where optical ground stations exist. The proposed system only considers the reverse link, i.e. the link from the forward-deployed user to a centralized command center. The forward link, from the command center to the forward deployed user, is left as future work and could be provided as a broadcast-like link. The following chapter discusses the system architecture, constraints, assumptions, and requirements.

2.1 System Overview

The proposed system entails a user transmitting over RF with a handheld radio, a CubeSat in LEO receiving and sampling the RF signal then retransmitting the digital signal over an optical communications system, and an optical ground station for receiving the optical signal and performing GBBF. GBBF will be used to provide benefits for the uplink signal even though GBBF will be performed at the optical ground station.

2.1.1 Operational View

Figure 2-1 depicts the use of the communications system in an operational capacity.

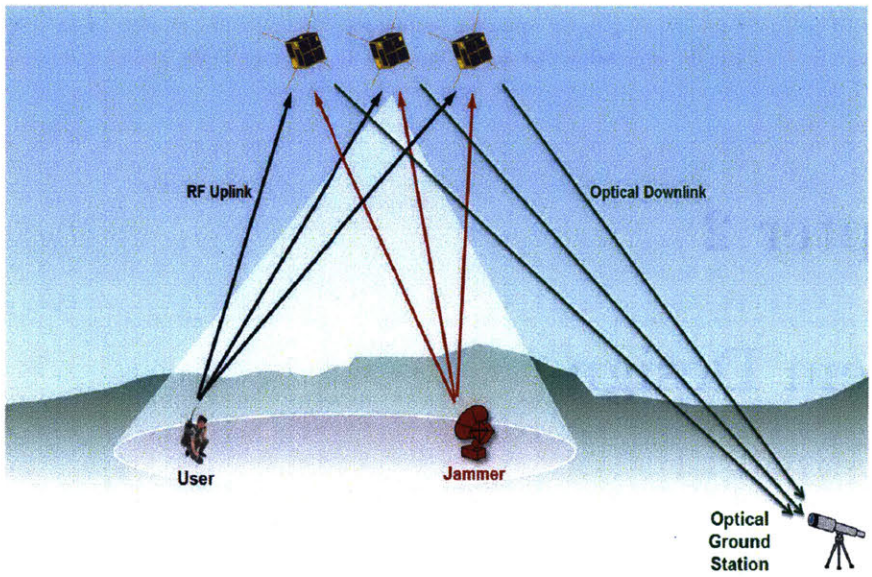

Figure 2-1: Operational View of System Architecture

As mentioned earlier, this system is asymmetric as it does not contain a forward link. The reverse link will be used for all of the communications as it is assumed the user is attempting to communicate with a command station but cannot due to the presence of a jammer. Each CubeSat acts as an element of an antenna array used for GBBF, and the presence of multiple CubeSats allows a null to be placed over the jammer, thereby mitigating the effects of the jammer and enabling successful communications.

2.1.2 System Geometry

The system geometry is shown in Figure 2-2.

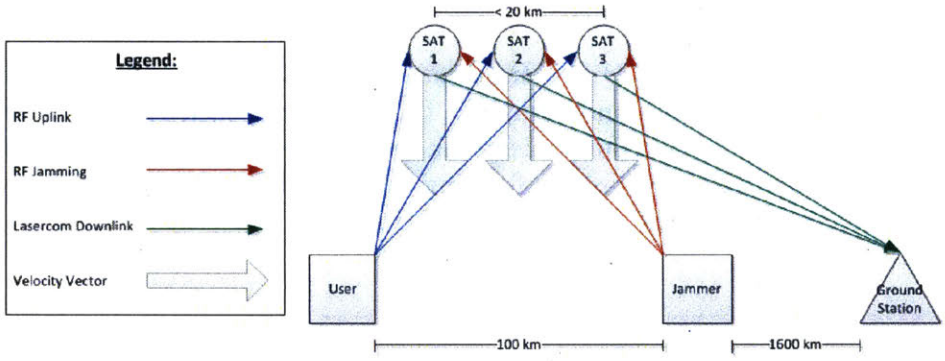

Figure 2-2: Overhead View of System Geometry

The CubeSats are assumed to be separated by no more than 10 km from the adjacent CubeSat(s) in the formation (cluster). Additionally, the velocity vector for

each CubeSat will be equal to the velocity vectors of the other CubeSats in the cluster. The user and jammer are separated by no more than 100 km to ensure that the jammer is near enough to the user to cause a significant amount of interference. The user is also no more than 1600 km from the optical ground station so that the CubeSats have line-of-sight to both the user and ground station for the majority of passes.

2.1.3 Assumptions

The following assumptions were made while designing the system:

1. Ground station stays at a fixed location
2. User location unknown to CubeSats, and jammer location unknown to both CubeSats and ground station
3. CubeSats know ground station location
4. Ground Station knows formation location
5. Spacecraft pointing sufficient for lasercom
6. Sufficient system timing onboard CubeSats
7. Formation flying between CubeSats in cluster
8. Uplink frequency limited to UHF
9. When null forming, no additional gain on mainlobe
10. Only one user attempting to communicate at any given time
11. Distance from jammer to CubeSats same as distance from user to CubeSats throughout pass

Among the assumptions above, some are intended to limit the scope of the analysis while others do not require analysis as they have been proven by demonstrations. Assumptions 1, 2, 8, 9, 10, and 11 limit the scope of the of the analysis, while the other assumptions are reasonable due to previous CubeSat missions and demonstrations. Assumptions 3 and 4 are possible with satellite-ground communications and two-line elements (TLEs). Precise pointing between a nano-satellite optical transmitter and optical ground station are laid out in [23, 70]. Sufficient system timing and timing synchronization across CubeSats in the cluster can be accomplished by chip scale atomic clocks, intersatellite communications, GPS receivers, or any combination of these. Intersatellite communications have been demonstrated by FASTRAC and CanX-4&5, and CanX-4&5 demonstrated precise relative position determination using GPS receivers [8]. Both CanX-4&5 and AeroCube-4 demonstrated formation flight between CubeSats, while CanX-4&5 proved that precise formation flying between two nano-satellites is possible [8, 52, 53].

2.1.4 Constraints

The following constraints are placed on the system:

1. RF only used on uplink, lasercom only used on downlink
2. Satellites must point at optical ground station during pass
3. Lasercom data rate will be less than 75 Mbps
4. Digital Video Broadcasting Second Generation (DVB-S2) Quasi Error Free Packet Error Rate (PER) = 1×10^{-7}
5. Minimum 10° RF elevation angle
6. Minimum 20° lasercom elevation angle
7. RF data is sampled and sampled data is transmitted over lasercom system

For the proposed communications system, RF will only be used on uplink; however, additional RF communications may be used in the form of a GPS receiver or RF transceiver, with the RF transceiver being used for inter-satellite communications (cross-links) links between CubeSats in a cluster. These cross-links could be used to coordinate maneuvers among the CubeSats in a cluster. The constraint on the lasercom data rate will be made more clear in Chapter 3. The RF uplink will use DVB-S2 for forward error correction (FEC). A minimum elevation angle of 20° is assumed in [48, 23]. In order to make the communications system compact and minimize complexity, the sampled RF data will be transmitted over the lasercom system.

2.2 Lasercom Downlink

The lasercom downlink presented in the following section is based on the Nanosatellite Optical Downlink Experiment (NODE) that was designed in [48, 70, 23, 47, 63, 62]. Chapter 3 builds on the lasercom downlink by designing and testing the waveform used in the transmitter. The lasercom module is intended to be used on a nanosatellite, and for that reason, it can fit into a volume of approximately 1U (10 cm x 10 cm x 10 cm), uses only 10 W peak power, and uses all COTS parts with custom interface boards.

2.2.1 Overview

Transmitter

The hardware for the lasercom downlink is shown in Figure 2-3.

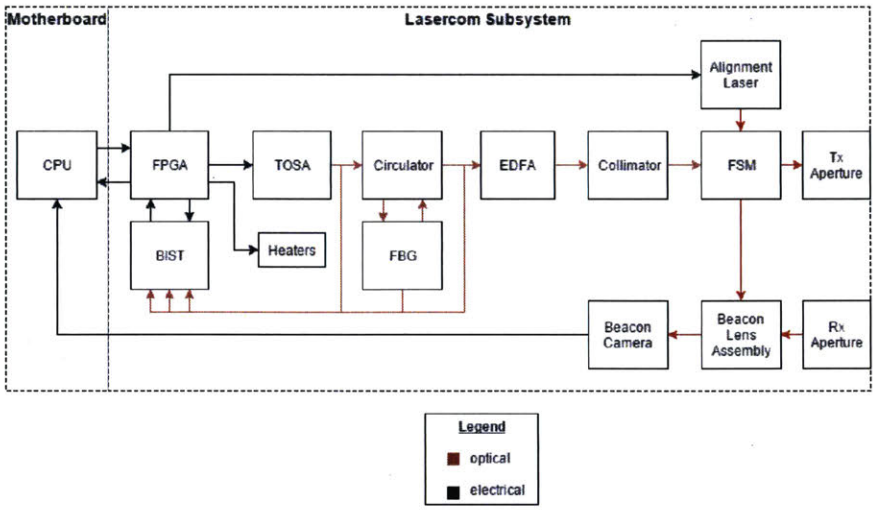

Figure 2-3: Block Diagram of Lasercom Downlink, adapted from [48, 23]

Data processed by the CPU is passed into the field programmable gate array (FPGA). The FPGA modulates the seed laser, located in the transmitter optical sub-assembly (TOSA), with the data and sends information about its status back to the CPU. Not shown in Figure 2-3 are the connections between the FPGA and all of the other boards in the lasercom subsystem; the FPGA acts as an interface between the CPU and the fast steering mirror (FSM), TOSA, heaters, and Erbium-doped fiber amplifier (EDFA). Additionally, the FPGA controls the alignment laser and interacts with the built-in-self-test (BIST) module. The laser passes through the circulator, subsequently reaching the fiber Bragg grating (FBG). The FBG reflects the passband of interest, 1550 nm $\pm$ 0.5 nm. The reflected passband is passed back through the collimator to the EDFA, where the optical signal is amplified to an average power of 200 mW. The collimator converts the optical signal in the fiber into a free space optical (FSO) signal. The FSM steers the FSO signal towards the optical ground station. In order to determine the orientation of the FSM, the alignment laser is reflected off of the FSM and into the beacon lens assembly, which passes it into the beacon camera. Additionally, the beacon signal from the ground station is passed through the beacon lens assembly and onto the beacon camera. Both the location of the alignment laser and beacon signal on the beacon camera are sent back to the CPU, where algorithms determine the proper position for the FSM. The CPU sends information concerning the FSM position to the FPGA, which in turn commands the FSM to the desired position.

The BIST module is meant to monitor the components in the lasercom system and aid in diagnosing any problems that may arise. The optical signals passed into the BIST module are inputs to photodiodes, which convert the optical signal into an electrical signal. That electrical signal is processed and passed into the FPGA for further analysis. The optical signals of interest are those that are around the FBG. Even though the FBG is considered athermal, the temperature variations expected in LEO are large enough to significantly shift the passband of the FBG, causing the

optical signal to miss the passband and be filtered completely. The optical signals being inputted into the BIST aid in locating the passband of the FBG when it shifts, and in order for the seed laser's output wavelength to align with the FBG passband, the seed laser will be operated at a different bias point, denoted by some bias current and temperature. Moreover, heaters will be used to limit the temperature variations within the lasercom subsystem, thereby limiting the amount by which the FBG passband will shift.

Receiver

Two optical ground stations will be used to demonstrate the performance of the lasercom downlink: the Optical Communication Telescope Laboratory (OCTL) and an amateur telescope. In addition to either telescope, the ground station will use a FSM, avalanche photodiode (APD), infrared (IR) camera, and signal acquisition machine. Figure 2-4 depicts the layout for the optical receiver.

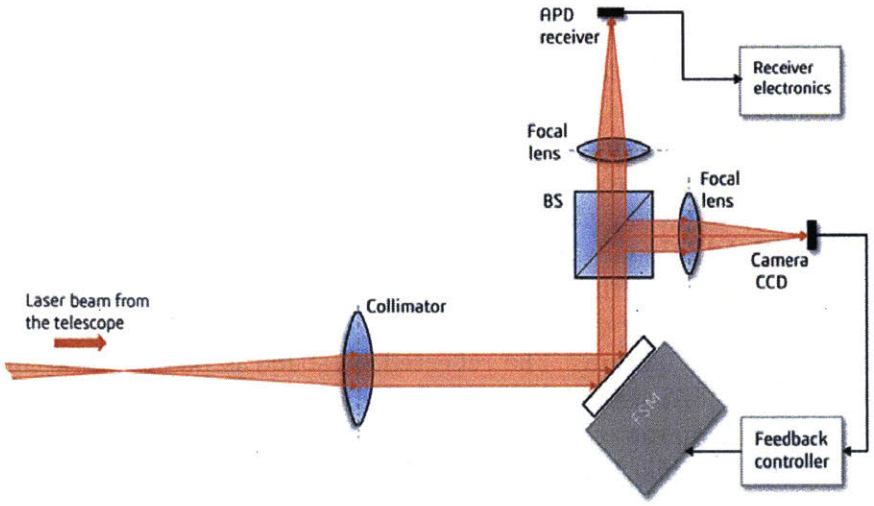

Figure 2-4: Block Diagram of Optical Receiver [23]

The telescope focuses the laser beam onto the FSM, where it is steered onto the active area of the APD. The beamsplitter (BS) directs some of the laser towards the IR camera, which provides input to the feedback controller to aid the FSM in steering the beam. The signal acquisition machine, shown as "Receiver electronics" in Figure 2-4, samples and stores the electrical signal outputted from the APD to be post-processed. Due to time and resource constraints, the receiver was chosen to be a post-processing receiver [23]. The receiver electronics are modeled after the Lunar Laser OCTL Terminal (LLOT) employed by JPL during LLCD [23, 75]. Using the amateur telescope to track a satellite has been discussed in [85], and developing the receiver optics and feedback controller are the focus of graduate student Kathleen Riesing.

2.2.2 Parameters

Transmitter

The lasercom downlink is designed to operate at 1550 nm, while the beacon signal will operate at 976 nm. The design employs a master oscillator power amplifier (MOPA) configuration, which uses the FBG to obtain a large extinction ratio ($\gtrsim$40 dB) and an EDFA to obtain an output power of 23 dBm. The following image depicts the MOPA architecture.

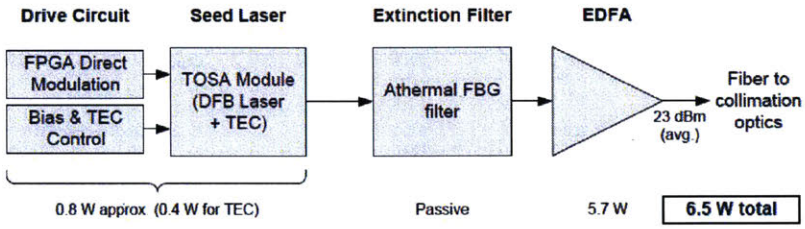

Figure 2-5: MOPA Architecture for NODE [48]

The design trade resulting in the choice of the MOPA architecture is discussed in [47, 48]. Even though the MOPA consumes 6.5 W, the entire lasercom subsystem will consume approximately 10 W. M-ary pulse position modulation (M-PPM) will be used to transmit information. The slot rate will be fixed at 200 MHz [48], and multiple PPM orders will be supported in order to allow different data rates to be used[1]. The lowest supported PPM order, 4-PPM, results in the highest data rate, 75 Mbps[2], but is the most taxing on the link budget. The highest supported PPM order, 128-PPM, results in the lowest data rate, 8.2 Mbps[2], but is the least taxing on the link budget due to increased signal-to-noise ratio (SNR).

Receiver

OCTL has a 1 m aperture [83] while the amateur telescope has a 30 cm aperture. Two APDs were chosen in [48], and those are the RDC1-NJAF and the RIP1-NJAF. Both have a 200 µm active diameter, but the RDC1-NJAF has a 300 MHz bandwidth while the RIP1-NJAF has a 1 GHz bandwidth [5]. The signal acquisition system mentioned earlier will be the Pentek Talon RTR 2729A. The Pentek Talon is capable of sampling at 3.6 GS/s with a 12-bit A/D, may contain as much as 30.7 TB of storage, and can sustain real-time recording rates up to 4.0 GB/sec, all of which make the Pentek Talon more than capable to successfully sample the signal from the APD. As operating the IR Camera, designing the feedback controller, and selecting the collimator and focal lenses are the focus of graduate student Kathleen Riesing, they will not be discussed here.

[1] A discussion of M-PPM is shown in section 3.2

[2] Data rates account for the use of inter-symbol guard times (ISGTs) and the overhead used by RS(255,239) but do not account for the acquisition sequence in section 3.5.1 as it has an insignificant impact on the data rate.

2.2.3 Link Budget

A detailed link budget for NODE, the lasercom downlink, is shown in Table 2.1.

Table 2.1: Link Budget for the initial NODE demonstration for two different ground stations: 1-m aperture OCTL, and a 30-cm aperture amateur telescope (AT). This table is provided courtesy of graduate student Emily Clements and [23]. Note that the data bit rates include overhead used for Inter-Symbol Guard Times (ISGTs) and FEC

Input Parameters	**Amateur Telescope**	**OCTL**	**Units**	**Notes**
Channel data rate	8.75	37.5	Mbps	Constant slot rate of 2.00E+08 Hz
PPM order	128	16		
Laser Transmitter				
Average optical output power	0.20	0.20	W	Unchanged transmitter architecture.
Laser Wavelength	1550	1550	nm	
Extinction ratio	42	42	dB	Kingsbury 2015 measurements [48].
Half-power beamwidth	1.33	1.33	mrad	Selected flight collimator
Receive Telescope & Optics				
Focal length	2.8	75.8	m	AT datasheet[18], OCTL paper[83]
Aperture diameter	27	100	cm	Scaled AT to account for secondary mirror.
Background Noise				
Field of View	7.14E-05	2.64E-06	rad	
Sky Spectral Radiance	6.00E-04	6.00E-04	$\frac{W}{cm^2 * SR * \mu m}$	Originally from Hemmati 2009[40].
Optical filter bandwidth	1	1	nm	
Receiver Electronics				
APD Gain	20	20	-	
Responsivity	1.0	1.0	A/W	Measured value from Kingsbury 2015[48].
Excess Noise Factor	4.3	4.3		From device sales sheet via Kingsbury 2015 [48]
Noise equivalent power	2.80E-09	2.80E-09	W	From device sales sheet via Kingsbury 2015 [48]
Noise equivalent bandwidth	3.00E+08	3.00E+08	Hz	Larger than signal bandwidth Kingsbury 2015 [48]
Link Budget Summary				
Laser avg. optical power	-7.0	-7.0	dBW	
Transmit optical losses	-1.5	-1.5	dB	Accounting for 0.3 dB per planned splice.
Transmit antenna gain	69.6	69.6	dBi	
Pointing loss	-3.0	-3.0	dB	
Path loss at 1000 km	-258.2	-258.2	dB	
Atmospheric loss	-1.0	-1.0	dB	
Receive antenna gain	114.7	126.1	dB	
Receive optics losses	-2.0	-3.0	dB	AT assumes -1 dB for beamsplitter plus miscellaneous losses; OCTL has higher losses due to more complicated optics.
Receiver Implementation loss	-3.0	-3.0	dB	Measured value from Kingsbury 2015[48]
Signal power at detector	-88.3	-78.0	dBW	
Signal power req'd, BER=1e-4	-93.2	-84.2	dBW	includes implementation loss
Margin at 1000 km	4.8	6.2	dB	Maximum range
Margin 600 km	7.6	8.9	dB	Nominal range

While the link budget does not show all PPM orders, it does show expected performance for 16-PPM and 128-PPM. As one would expect, performance is much better with OCTL than it is with the amateur telescope due to the larger aperture diameter of OCTL. The link margins are more than sufficient for nominal range and maximum range (at a 20° elevation angle). The link budget shows that 128-PPM can be used reliably with the amateur telescope while PPM orders 16-128 can be used reliably with OCTL. For the sake of brevity, a discussion of the calculations and equations used to generate the link budget will not be included in this thesis but may be found in [48, 23, 40].

2.3 RF Uplink

The RF Uplink involves the user, transmitting with a handheld radio, and the receivers on each of the three CubeSats. This section discusses the RF transmitter and receiver, then shows link budgets proving that communications are possible from a handheld radio to a CubeSat in LEO.

2.3.1 Overview

Transmitter

The transmitter consists of a mobile or handheld radio, such as the AN/PRC-152. Figure 2-6 is a picture of the AN/PRC-152.

Figure 2-6: Image of AN/PRC-152A Wideband Networking Handheld Radio

The communication system aims to be fairly agnostic to the specific transmitter being used due to the sample-and-retransmit method used in the CubeSat receivers. The transmitter must be designed to deliver sufficient C/N_o to the CubeSats such that the ground system can demodulate and decode the digital data that has been relayed via the optical downlink; however, the communications system could support various modulation formations and FEC codes if design trades make one more advantageous than another. For this reason, the AN/PRC-152A is representative of the class of terminal this communications system could support.

Receiver

The receiver onboard the CubeSat is a fairly generic receiver front-end since its goal is to sample the RF data and send it to the lasercom subsystem, which will transmit the sampled data to the ground. The block diagram in Figure 2-7 shows the receiver front-end and how it will interface with the rest of the CubeSat.

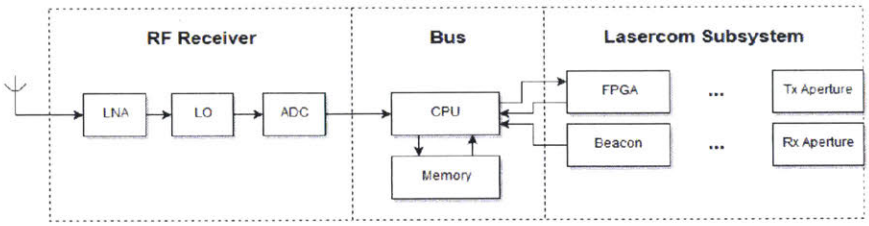

Figure 2-7: Block Diagram of RF Front-End on CubeSat

An antenna, low noise amplifier (LNA), local oscillator (LO), and ADC receive the signal, amplify it, convert it to baseband, and sample the signal. The sampled data is passed directly to the CPU and stored in memory before being downlinked by the lasercom subsystem. The receiver front end could be accomplished by a small form-factor software defined radio (SDR), such as the Ettus Research USRP B200mini.

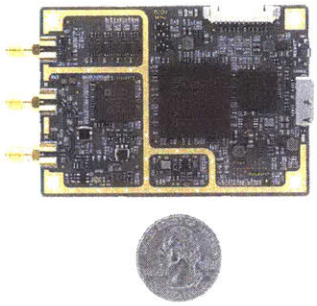

Figure 2-8: Image of Ettus Research USRP B200mini

With a frequency range of 70 MHz - 6 GHz, an instantaneous bandwidth of 56 MHz, a small form-factor, and an input for an external clock reference, the USRP B200mini could accomplish all of the tasks needed for the receiver front-end, as shown in Table 2.3.

2.3.2 Parameters

The parameters used when generating the link budgets are shown in the following subsections. Some parameters, such as transmit power and modulation scheme, may vary across different transmitters, and for that reason, the link budgets are representative of a general communications scenario to demonstrate feasibility.

Transmitter

The parameters for the transmitter are shown in Table 2.2.

Table 2.2: RF Transmitter Parameters

Parameter	Value
Center Frequency	500 MHz
Power	7 dBW
Modulation	QPSK
Double-Sided Bandwidth	100 kHz
Coding	1/2 rate DVB-S2
Antenna	Hemispherical and RHCP
Antenna Efficiency	50%

A transmit power of 7 dBW is assumed as it is the upper bound on transmit powers for mobile, handheld radios. A center frequency of 500 MHz is assumed, but another frequency within the UHF band could be chosen depending on the application of the communications system or the ability to obtain a frequency license. The FEC code is Digital Video Broadcasting Second Generation (DVB-S2), which specifies Quadrature Phase Shift Keying (QPSK) and operates between 0.7 dB and 1 dB from the Shannon limit [29]. The antenna is assumed to be hemispherical, with a 3 dBi gain, and right-hand circularly polarized (RHCP). A hemispherical antenna exempts the user on the ground from having to know the satellite's location and point the antenna. The antenna is assumed to have 50% efficiency, which is a standard efficiency for most antennas.

Receiver

Table 2.3 shows the parameters for the receiver.

Table 2.3: RF Receiver Parameters

Parameter	Value
Center Frequency	500 MHz
NF	2 dB
Modulation	QPSK
Antenna	Hemispherical and Linearly Polarized
Antenna Efficiency	50%
Antenna Temperature	290 K
Sampling	2 MS/s with 16-bit ADC

The receiver is assumed to have a 2 dB noise figure (NF) as it is the NF for the Analog Devices AD9364 RFIC transceiver [4] on the USRP B200mini [3]. As with the transmitter, an antenna efficiency of 50% is common for many antennas. Two dipoles will be mounted on opposing sides of the CubeSat in order to give the receiver a near-omnidirectional view of the ground. An antenna temperature of 290 K is common for antennas pointing towards Earth [76].

Table 2.4: Link Budget for RF Uplink assuming only one satellite and one user not in the presence of interference.

Parameter	Best Case	Worst Case	Units	Notes
Distance	600	2000	km	
Frequency	500	500	MHz	
Bandwidth	100	100	kHz	Double-Sided Bandwidth
Modulation/Coding	QPSK/DVB-S2	QPSK/DVB-S2		
Data Rate	100	100	kbps	Includes overhead for DVB-S2
Transmitter				
Transmit Gain	3	3	dBi	Hemispherical antenna pattern
Transmit Power	7	7	dBW	
Transmit Losses	-3	-3	dB	50% Antenna efficiency
EIRP	7	7	dB	
Channel				
Free Space Path Loss	-141.99	-152.45	dB	
Atmospheric Losses	-0.02	-0.2	dB	From STK Simulation
Polarization Losses	-3	-3	dB	RHCP to LP
Receiver				
Receiver Gain	3	3	dB	Monopole on poor ground plane
Receiver Losses	-3	-3	dB	50% Antenna Efficiency
Noise Factor (NF)	2	2	dB	
System Temperature (T_{sys})	28.75	28.75	dBK	
G/T	-28.75	-28.75	dB/K	
Link Information				
C/N_o	61.84	51.20	dB-Hz	
E_b/N_o	11.84	1.20	dB	
Required E_b/N_o	1	1	dB	Quasi Error Free Packet Error Rate=1×10^{-7} [29]
Link Margin	10.84	0.20	dB	

2.3.3 Link Budget

Table 2.4 shows a link budget of the RF uplink for one satellite and one user. It is assumed that no interference is present in this case.

Communications are possible for the scenario in Table 2.4 given the assumptions and constraints placed on the system. It is important to note that the data rate includes the overhead for a 1/2 rate DVB-S2 code, meaning only a data rate of 50 kbps will be available for transmitting information. The best case, or minimum distance, occurs at a 90° elevation angle whereas the worst case, or maximum distance, occurs at a 10° elevation angle.

When not in the presence of interference, the RF uplink closes with sufficient link margin. However, for the scenario of interest, there may be jamming or other sources of interference. The link budget in Table 2.5 examines the communications scenario with one satellite, one user, and one jammer with a 13 dB power advantage over the user.

Table 2.5 shows that the link fails when in the presence of interference, where I_o denotes the interferer's PSD at the receiver. The terms $C/(N_o + I_o)$ and $E_b/(N_o + I_o)$ are used to account for both the interferer and the noise when determining link margin. In many cases, such as [73], C/I_o and E_b/I_o are used as the metrics to determine link margin, but for the purposes of this thesis, $C/(N_o + I_o)$ and $E_b/(N_o + I_o)$ allow for a better comparison with tables in Chapter 4. Multiple methods can be employed to mitigate the effects of the interference, and two of those methods are explored in Chapter 4.

Table 2.5: Link Budget for RF Uplink assuming one satellite and one user while in the presence of interference.

Parameter	Best Case	Worst Case	Units	Notes
Distance	600	2000	km	
Frequency	500	500	MHz	
Bandwidth	100	100	kHz	Double-Sided Bandwidth
Modulation/Coding	QPSK/DVB-S2	QPSK/DVB-S2		
Data Rate	100	100	kbps	Includes overhead for DVB-S2
Transmitter				
Transmit Gain	3	3	dBi	Hemispherical antenna pattern
Transmit Power	7	7	dBW	
Transmit Losses	-3	-3	dB	50% Antenna efficiency
EIRP	7	7	dB	
Jammer				
Jammer Gain	3	3	dBi	
Jammer Power	20	20	dBW	
Jammer Losses	-3	-3	dB	50% Antenna efficiency
EIRP	20	20	dB	
Channel				
Free Space Path Loss	-141.99	-152.45	dB	
Atmospheric Losses	-0.02	-0.2	dB	From STK Simulation
Polarization Losses	-3	-3	dB	RHCP to LP
Receiver				
Receiver Gain	3	3	dB	Monopole on poor ground plane
Receiver Losses	-3	-3	dB	50% Antenna Efficiency
Noise Factor (NF)	2	2	dB	
System Temperature (T_{sys})	28.75	28.75	dBK	
G/T	-28.75	-28.75	dB/K	
Link Information				
$C/(N_o + I_o)$	36.99	36.84	dB-Hz	
$E_b/(N_o + I_o)$	-13.01	-13.16	dB	
Required $E_b/(N_o + I_o)$	1	1	dB	Quasi Error Free Packet Error Rate=1×10^{-7} [29]
Link Margin	-14.01	-14.16	dB	

2.3.4 Design Choices

The modulation format is QPSK because initial link budgets revealed that the system had more than sufficient link margin for Binary Phase Shift Keying (BPSK), a common modulation scheme for satellite communications, but was limited in bandwidth due to the lasercom downlink. Since QPSK has twice the data rate of BPSK given the same bandwidth at the cost of a 3 dB worse average symbol-energy-to-noise-spectral-density ratio than BPSK [87], the excess link margin and limited bandwidth drove the decision to switch from BPSK to QPSK. The bandwidth is 100 kHz so the data rate from the receiver, after accounting for oversampling and I and Q channels, must be less than the data rate provided by the lasercom downlink. The sampling rate on the receiver is 2 MS/s to allow for a 10 times oversampling and to capture both I and Q channels. A 12-bit ADC provides more than sufficient resolution and dynamic range. Using 10 times oversampling and a 12-bit ADC ensures that the sampled data being retransmitted over the lasercom downlink may be reconstructed on the ground.

Chapter 3

Lasercom Downlink

This chapter covers the waveform design for the lasercom downlink. PPM, channel coding, interleaving, framing, padding, symbol mapping (known as mapping below), the software modulator, and the FPGA implementation are discussed. Results from testing the channel coding, interleaving, and symbol mapping across a simulated channel are shown. The entire waveform design process is shown in the section labeled "A Day in the Life of a Packet". The FPGA design, which includes the firmware, memory units, and modulator, is explained through block diagrams and a finite state machine diagram. The chapter concludes with the results of tests performed on the modulator, all of which confirm that the modulator functions correctly. The concept and hardware for the lasercom downlink are outlined in [48, 23]. Additionally, the interface between the FPGA and CPU and the beginnings of the waveform design are implemented in [48]. The work presented here only covers the waveform design aspect of the lasercom transmitter and builds upon previous work done on the lasercom transmitter.

3.1 Overview

The waveform design includes the error correction, interleaving, framing, and modulation. The waveform design block diagram is shown in Figure 3-1.

The transmitter is composed of a microprocessor, FPGA, and the optical components as specified in [48]. The receiver is a post-processing receiver, where the electrical signal leaving the APD is sampled and stored in the Pentek Talon signal acquisition system. Thus, all of the operations to recover the data are performed in software. The receiver is modeled after the post-processing receiver used in [75].

3.2 Pulse Position Modulation

PPM is a modulation scheme that uses the location of a pulse within a symbol period to convey information. Each pulse location is known as a slot, and only one slot may contain a pulse, also known as an active slot, within each PPM symbol period. M-PPM has M slots within each symbol period and encodes $\log_2(M)$ bits per

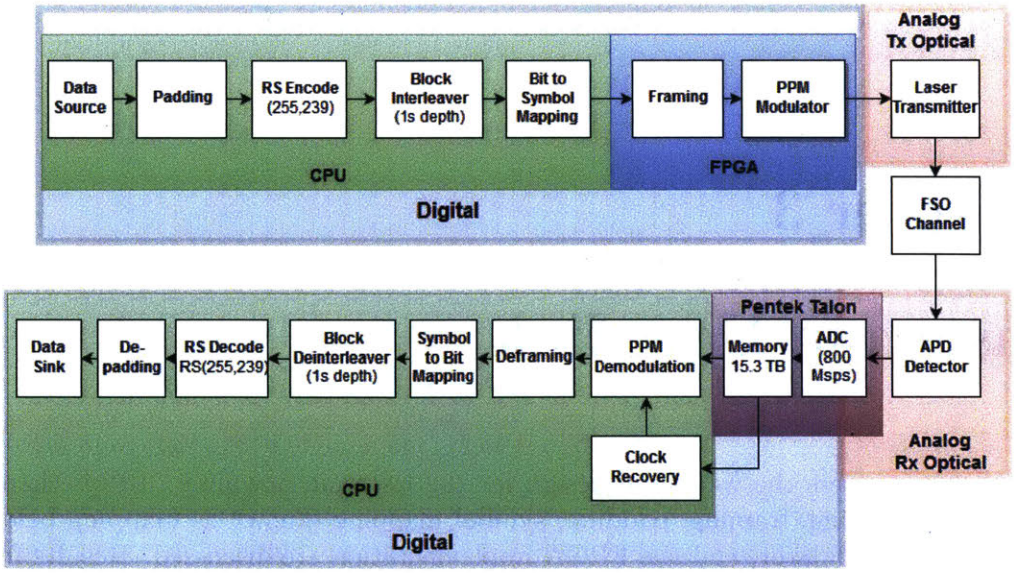

Figure 3-1: Waveform Design Block Diagram

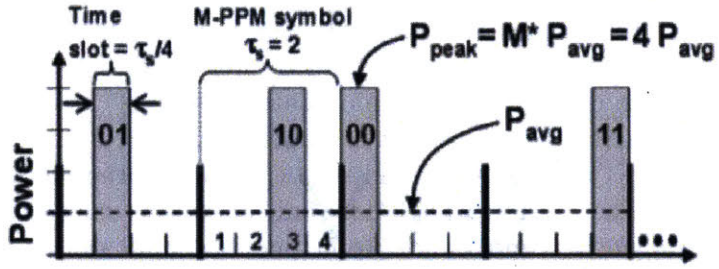

Figure 3-2: Image of 4-PPM from [16]

PPM symbol, where the bits determine the location of the active slot [37, 16]. The duty cycle of an M-PPM waveform is equal to $1/M$, making PPM a power-efficient modulation scheme. Figure 3-2 shows 4-PPM.

PPM was chosen as the modulation format for the lasercom transmitter in [48]. The lack of low-SWaP coherent lasercom systems required the use of an IM scheme. The choice of a MOPA configuration, which uses an average-power-limited EDFA, made PPM more advantageous than other intensity modulation schemes, such as OOK, due to the high peak-to-average power ratio provided by its low duty cycle.

3.3 Channel Coding

Since the optical channel is noisy and causes errors during transmission, the lasercom downlink uses channel coding. There are many different types of error-correcting codes (ECC), with Reed-Solomon codes and concatenated codes, such as serially concatenated PPM (SCPPM), being the most common ECCs for optical communications.

Reed-Solomon codes have good error-correcting capabilities and many open-source implementations exist [23]. SCPPM has been implemented successfully and would be a great choice for a lasercom transmitter as it operates within 1 dB of capacity [59]. Due to time constraints, resource constraints, and the lack of an open-source implementation for SCPPM, RS was chosen as the ECC for the lasercom transmitter.

3.3.1 Reed-Solomon

Reed-Solomon (RS) codes are a class of non-binary linear block codes, and more specifically cyclic codes, that have the largest possible minimum distance for any linear code with the same block length [73]. RS codes are described by the amount of symbols per codewords (codeword size or block length), denoted n, and the amount of message symbols per codeword, denoted k. Typically, a specific RS code will be denoted by RS(n,k). Each RS symbol has $\log_2(n+1)$ bits, meaning $n = 2^m - 1$, where m is the amount of bits per symbol. The amount of parity symbols per RS codeword is equal to $n - k$. The error correcting capability of a RS code is given by [73]:

$$t = \left\lfloor \frac{n-k}{2} \right\rfloor \tag{3.1}$$

As Equation 3.1 shows, the error correcting capability of a RS code is equal to half of the amount of parity symbols. If the location of an error is known, erasure decoding may be used to increase performance of the RS code. The erasure-correcting capability of a RS code is equal to $n - k$, which is the same as the number of parity symbols per codeword [73]. Error-correcting codes are typically compared based off of their code rate, and for RS codes, the code rate is:

$$r = \frac{k}{n} \tag{3.2}$$

The code rate is a way of categorizing codes by the amount of parity bits, or overhead, required and error-correcting capability offered. When discussing the performance of a code, the specific n and k are needed, in addition to the code rate, to accurately compare performance.

The theoretical performance of a RS code is given by [73]:

$$P_E = \frac{1}{n} \sum_{j=t+1}^{n} j \binom{n}{k} p^j (1-p)^{n-j} \tag{3.3}$$

where P_E is the probability of a RS symbol error after decoding and p is the probability of a RS symbol error before decoding. Equation 3.3 uses the binomial distribution to determine the probability of getting j errors per codeword, where j is always greater than the error-correcting capability of the code. Finding the expectation of the number of symbol errors per codeword and dividing that by the codeword size gives the probability of a RS symbol error after decoding.

As stated earlier, a RS code was chosen as the FEC code for this NODE. The

unireedsolomon codec[1] was chosen as the Reed-Solomon encoder due to it being open-source, well-documented, and universal (it is compatible with other RS codecs and allows the use of various n and k). Additionally, the codec can perform erasures decoding, which, as mentioned above, can improve the error-correcting capability of a RS code.

Since the physical layer was designed to provide a BER of 1×10^{-4} at the input to the decoder, a RS(255,239) code was chosen to provide near error-free communications [48]. It has been shown that a BER of 1×10^{-4} on the input of the decoder will amount to a BER of 1×10^{-15} at the output of the decoder [74], and a BER of 1×10^{-15} is considered to be error-free.

3.3.2 Reed-Solomon with PPM

Typically, communication systems using RS codes attempt to match the modulation symbol size and the RS symbol size so that each RS symbol maps uniquely to one modulation symbol. However, RS codes with short block lengths, which would be needed for the lower PPM orders, perform poorly [40]. The chosen RS code, RS(255,239), has a large enough block length to prevent poor performance but is not matched to the PPM orders being used. M-PPM uses $\log_2(M)$ bits to represent each modulation symbol while RS(255,239) uses 8 bits to represent each RS symbol. This represents a problem since one PPM symbol may span two RS symbols, meaning an error in the PPM symbol could cause an error in both RS symbols, leading to an RS symbol error rate (SER) that is greater than the PPM SER. Additionally, multiple PPM symbols may fit into one RS symbol, which can result in an increased RS SER given some PPM SER when the symbol errors are not correlated. Both of these cases are represented in Figure 3-3.

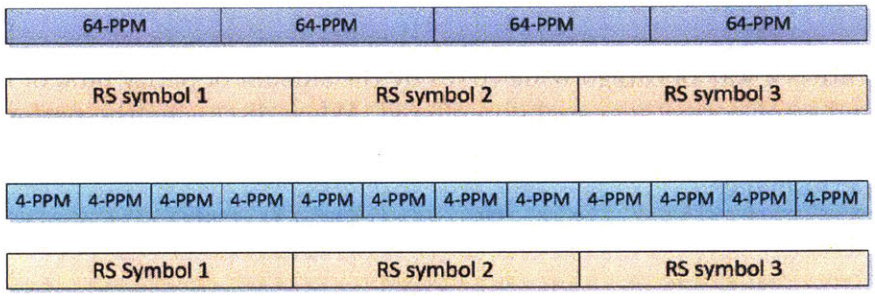

Figure 3-3: Mapping PPM symbols to RS Symbols

As Figure 3-3 shows, the modulation symbols are not matched to the RS symbols. This mismatch has been examined in [35] for an arbitrary M-ary modulation scheme used with RS encoding. For 4-PPM, four PPM symbols fit into each RS symbol, but for 64-PPM, a non-integer number of PPM symbols fit into each RS symbol, causing

[1]All of the material for the unireedsolomon module is on Github at https://github.com/lrq3000/unireedsolomon.

some overlap where one PPM symbol spans across two RS symbols. A block diagram of the communication system model used in [35] is shown in Figure 3-4.

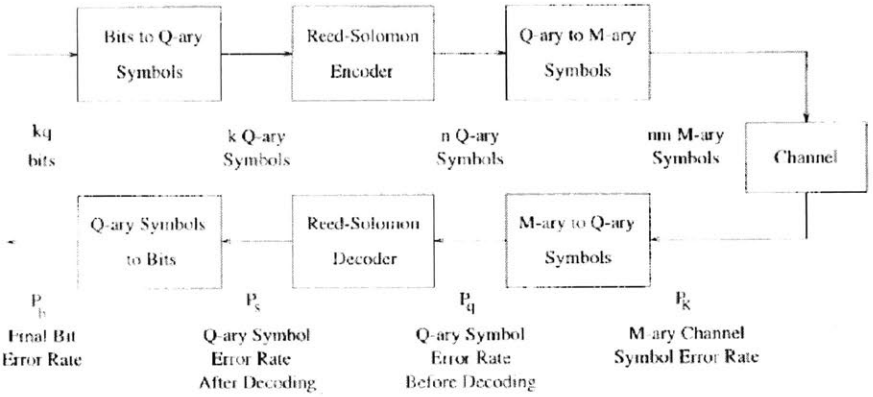

Figure 3-4: Block Diagram of RS Coded Communications System from [35]

In Figure 3-4, Q is equal to $n+1$, q is equal to $\log_2(Q)$, and M can take values from 4 to 128. For the remainder of this thesis, P_k will be shown as SER_M, P_q will be shown as SER_{RS}, P_S will be shown as SER_{DEC}, and P_b will be shown as BER_{DEC}. Table 3.1 shows the naming conventions that will be used.

Table 3.1: Table for RS Naming Conventions

Naming in [35]	Naming here	Description
P_k	SER_M	PPM symbol error rate
N/A	BER_M	PPM bit error rate
P_q	SER_{RS}	RS symbol error rate before decoding
N/A	BER_{RS}	RS bit error rate before decoding
P_S	SER_{DEC}	RS symbol error rate after decoding
P_b	BER_{DEC}	Bit error rate after decoding

The number of modulation symbols per RS symbol is denoted in Figure 3-4 as m and is given by [35]:

$$m \approx \log_2(Q)/\log_2(M) \tag{3.4}$$

The number of modulation symbols per RS symbol is helpful in calculating the SER_{RS}, from the SER_M. When m is an integer, SER_{RS} can be calculated from SER_M using the following equation given in [35].

$$SER_{RS} = 1 - (1 - SER_M)^m \tag{3.5}$$

For small values of SER_M, Equation 3.5 is given by [35]

$$SER_{RS} \approx m * SER_M \tag{3.6}$$

This multiplication occurs because only one modulation symbol needs to be in error for the entire RS symbol to be in error [35]. However, when m is not an integer, one PPM symbol may span two different RS symbols, such as the case for 64-PPM in Figure 3-3, making Equations 3.5 and 3.6 underestimates of the SER_{RS} [35]. Table I in [35] gives the multiplication factor for various combinations of Q and M and elaborates on the method to find the multiplication factors when one PPM symbol contains information from multiple RS symbols. Unfortunately, the table does not cover the case when 128-PPM is used with RS(255,239). Extrapolating and using the process from [35] yields the following equation for the conversion from SER_M to SER_{RS} for 128-PPM:

$$SER_{RS} = 2 * SER_M - SER_M^2 \tag{3.7}$$

Plotting SER_M versus SER_{DEC} with the equations given in Table I in [35] gives a theoretical estimate of performance for different PPM orders when used with RS(255,239). This plot is shown in Figure 3-5.

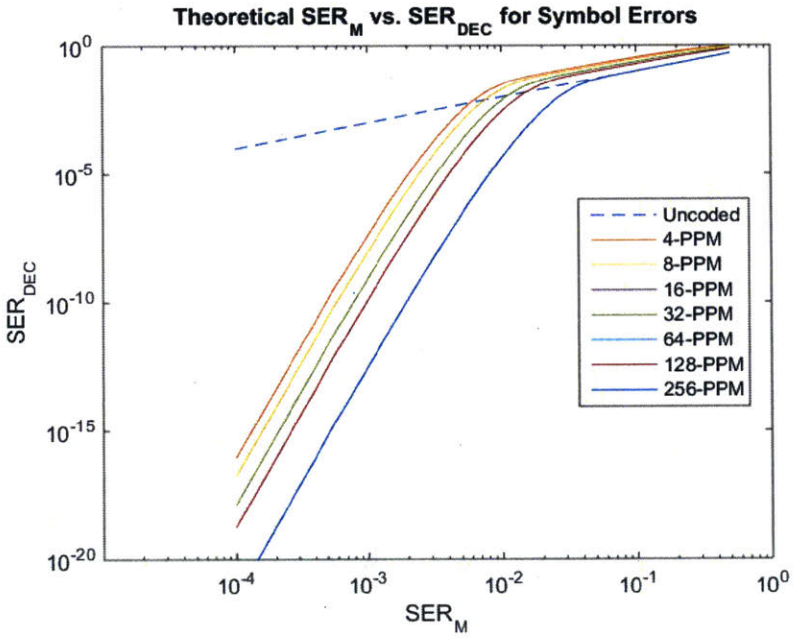

Figure 3-5: SER_M vs. SER_{DEC} for Non-Fading Channel with RS(255,239) and various PPM orders

Figure 3-5 shows that the best performance is achieved when the RS code and PPM order match in regards to the number of bits per symbol. In general, the error-correcting performance decreases as the PPM order decreases, with 16-PPM and 64-PPM being the exceptions[2].

[2]The curves for 16-PPM and 64-PPM are identical to the curve for 128-PPM, and for that reason, they are hidden by the 128-PPM curve

3.3.3 Validation and Testing

The validation and testing for the chosen RS code followed two steps. The first step involved testing the RS performance on a non-fading channel to determine if the codec could meet theoretical estimates. The second step tested the performance of the RS codec when using PPM over a non-fading channel. The performance of the RS codec with and without interleaving on a fading channel will be explained in Section 3.4. Figure 3-6 shows the tests and the components that are included in those tests.

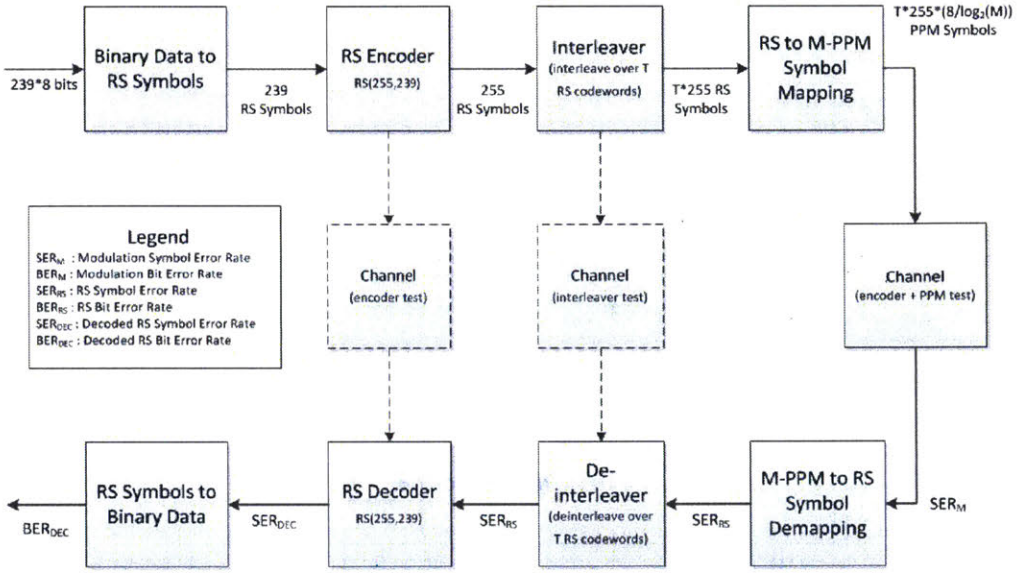

Figure 3-6: Block Diagram of RS Tests

The tests are designed to verify performance of the encoder, interleaver, and PPM symbol mapping. Additionally, the tests examine various channel effects and confirm the degradation in performance predicted in [35] by a RS order and modulation order mismatch.

Encoder Test

Simulations confirm that the RS(255,239) code will be sufficient to provide error-free communications. Figure 3-7 shows the simulation results when using RS(255,239) on the unireedsolomon codec with and without interleaving compared to the theoretical performance of RS(255,239). The near perfect correlation between the theoretical and simulation curves in Figure 3-7 confirms that the codec achieves theoretical performance, meaning the codec will be able to correct a BER_{RS} of 1×10^{-4} to a BER_{DEC} of 1×10^{-15}. The simulations were performed across a non-fading channel, and the performance in Figure 3-7 with and without interleaving is identical, which is expected for a non-fading channel. A binary channel, which either corrupts a symbol or lets one pass error-free, was used as the non-fading channel model. Figure 3-8 shows a temporal representation of the binary non-fading channel model.

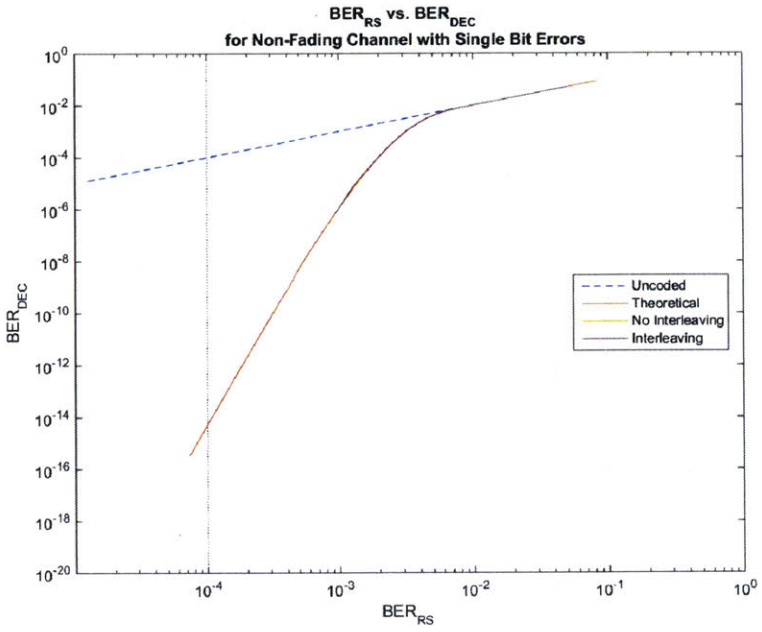

Figure 3-7: BER_{RS} vs. BER_{DEC} for RS(255,239) with Single Bit Errors

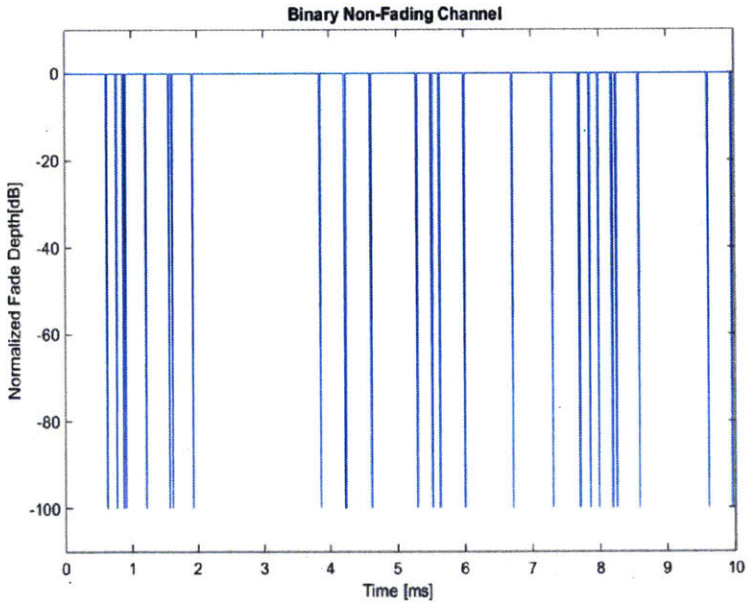

Figure 3-8: Temporal Representation of the Binary Non-Fading Channel Model

The non-fading channel model randomly distributes independent errors, as seen in Figure 3-8. Errors only occur when the attenuation reaches -100 dB, and the presence of two attenuation levels makes it a binary channel. Figure 3-7 uses the following SER

to BER conversion from [43].

$$BER_{Input} = 1 - (1 - SER_{Input})^{(1/m)} \tag{3.8}$$

$$BER_{Output} = 1 - (1 - SER_{Output})^{(1/m)} \tag{3.9}$$

where, as mentioned earlier, m is the number of bits per RS symbol, which is 8 for RS(255,239), and input and output refer to the input and output to/from the decoder. Equation 3.8 assumes that single bit errors are inserted by the channel, and the results change once that assumption is removed. Assuming the channel causes symbol errors and an M-ary orthogonal modulation scheme is used, the SER to BER conversion becomes [73]:

$$BER_{Input} = SER_{Input}\left(\frac{M}{2(M-1)}\right) \tag{3.10}$$

$$BER_{Output} = SER_{Output}\left(\frac{n+1}{2n}\right) \tag{3.11}$$

where M is the constellation size for the chosen modulation scheme and n is the codeword size used for the RS code. Figure 3-9 shows the same performance data as Figure 3-7 with the exception that the channel applies symbol errors rather than single-bit errors.

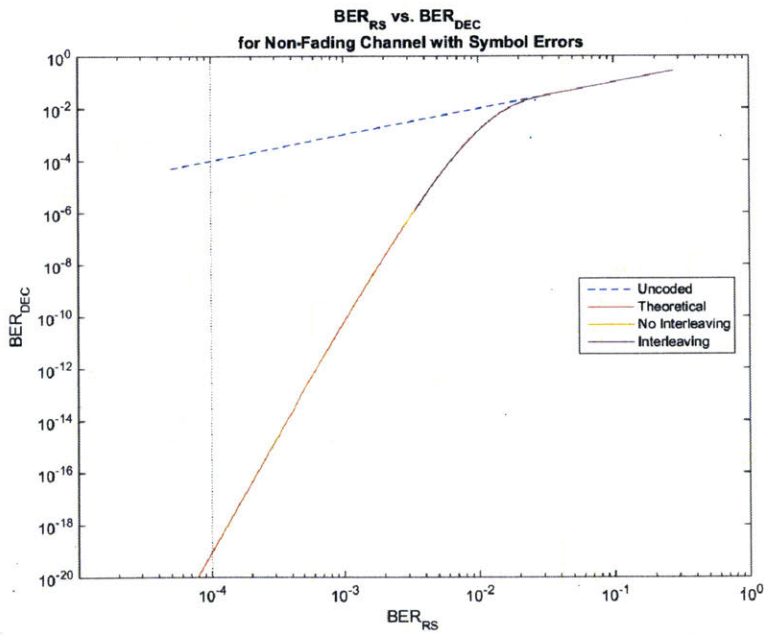

Figure 3-9: BER_{RS} vs. BER_{DEC} for RS(255,239) with Symbol Errors

As seen in Figure 3-7, the codec performs better, in terms of BER, in the presence of symbol errors rather than single-bit errors. Symbol errors result in multiple bit errors per symbol, which is seen as an improved decoder performance when comparing

BER_M to BER_{DEC}. A RS code can correct all of the bit errors within one symbol, and as more bit errors are packed into each symbol, the decoder is able to correct a higher input BER to the same decoded BER. In other words, Figure 3-7 and the performance of RS(255,239) given by [74] are upper bounds on the error-correcting capability for an RS code. Since symbol errors occur across the channel, it is not fair to assume that every symbol error will include only one bit error, and for this reason, Figure 3-9 is a more accurate measure of performance, so the remainder of the plots in this thesis will show symbol errors. Additionally, all other plots will be compared with Figure 3-9, where a channel BER of 1×10^{-4} results in a decoded BER of 1×10^{-19}. With this being said, the unireedsolomon codec achieves theoretical performance for RS(255,239).

Encoder with PPM Symbol Mapping Test

As mentioned earlier, the physical layer is designed to provide a BER of 1×10^{-4}, and with the degradation in performance due to RS to PPM symbol mapping, an additional test must be performed to determine if error-free performance is possible with RS(255,239) and PPM symbol mapping. Figure 3-10 uses the equations in Table I of [35] to show the BER for RS(255,239) when using different PPM orders.

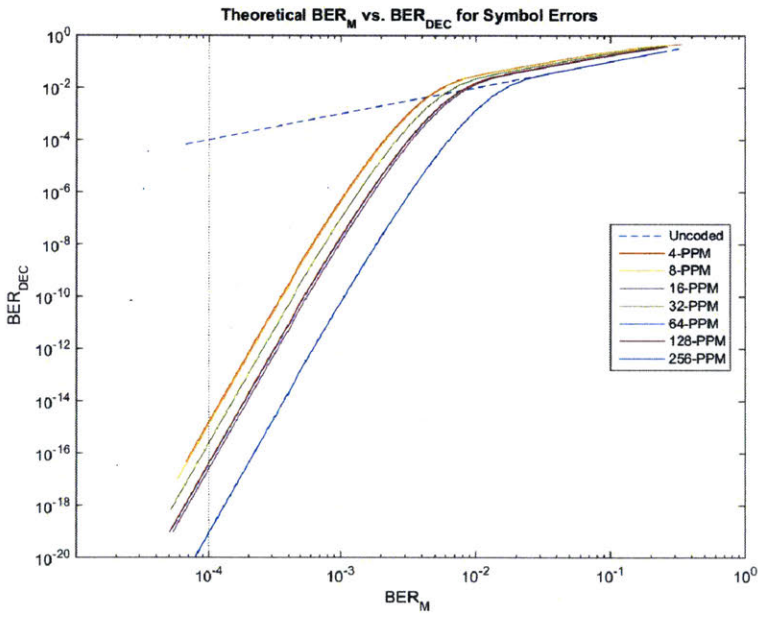

Figure 3-10: BER_M vs. BER_{DEC} for Non-Fading Channel with RS(255,239) and various PPM orders

In Figure 3-10, a BER_M of 1×10^{-4} for 4-PPM, the worst case, results in a BER_{DEC} of approximately 1×10^{-15}; therefore, the theory suggests that error-free communications can be obtained with RS(255,239) when PPM symbol mapping is used. Figure 3-11 shows the simulation results for PPM symbol mapping with

RS(255,239) on a non-fading channel, and since simulating down to 1×10^{-15} is not realistic for modern computing systems, all simulations end at 1×10^{-8}, which denotes quasi error-free communications since the maximum frame size is equal to 1×10^{7} symbols.

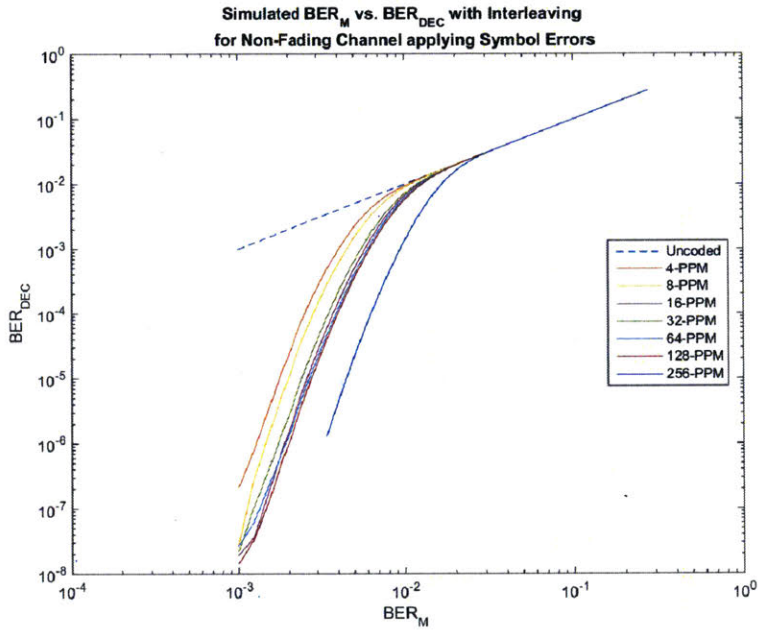

Figure 3-11: BER_M vs. BER_{DEC} for Non-Fading Channel with RS(255,239) and various PPM orders

Figure 3-11 confirms theoretical estimates from [35] and Figure 3-10 in regard to RS performance with PPM symbol mapping. As the theory suggested, 4-PPM is the worst performer while 128-PPM is the best performer. At a BER_M of 1×10^{-4}, all of the curves will be well below the quasi error-free BER_{DEC} of 1×10^{-8}, showing that the unireedsolomon codec will provide error-free performance with PPM symbol mapping. Interestingly, the simulations suggest that the theory overestimates BER_{RS}, and subsequently BER_{DEC} when converting from BER_M to BER_{RS}, and this is the most pronounced at high BERs. Figures 3-12 to 3-17 highlight the difference between theoretical estimates and simulation results by showing BER_M versus BER_{DEC} for RS(255,239) with all PPM orders.

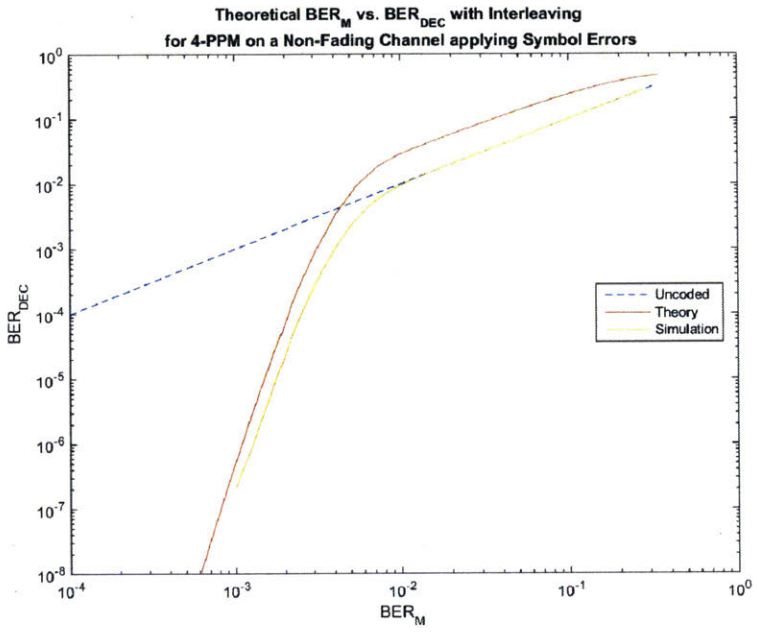

Figure 3-12: BER_M vs. BER_{DEC} for 4-PPM used with RS(255,239) on a Non-Fading Channel

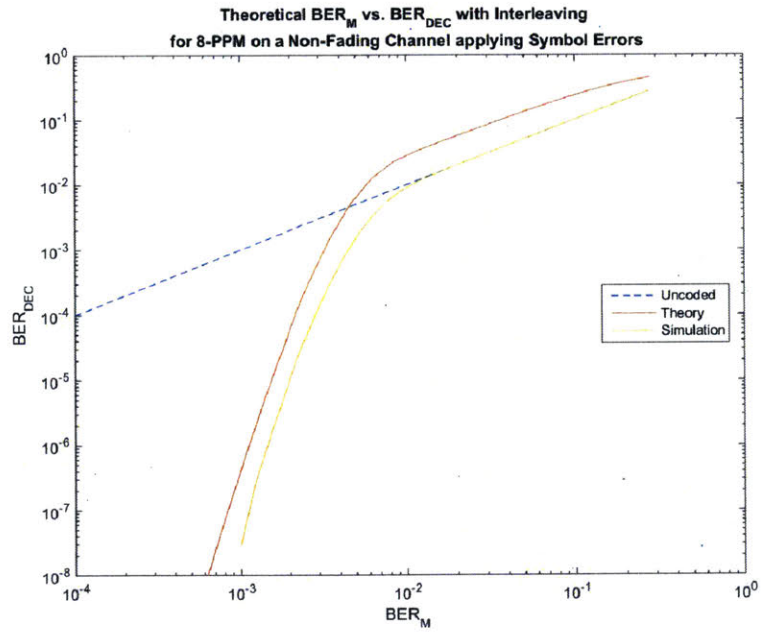

Figure 3-13: BER_M vs. BER_{DEC} for 8-PPM used with RS(255,239) on a Non-Fading Channel

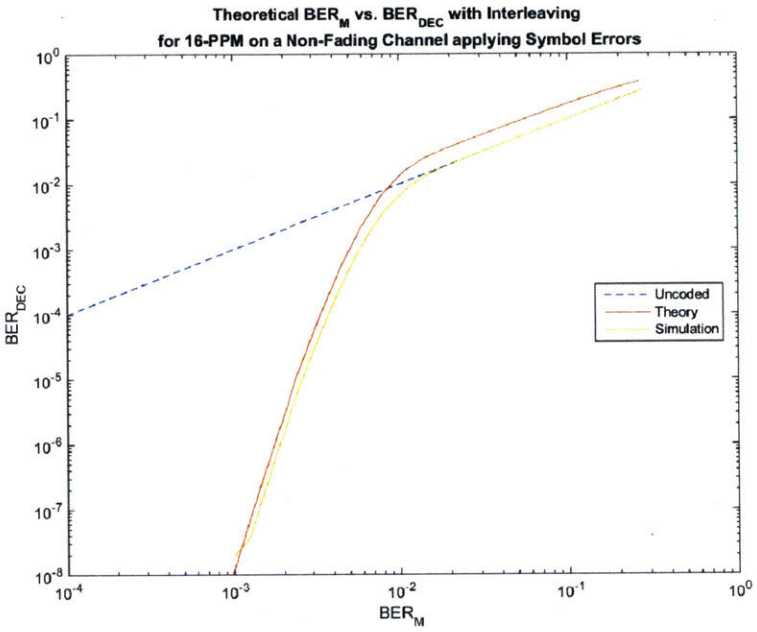

Figure 3-14: BER_M vs. BER_{DEC} for 16-PPM used with RS(255,239) on a Non-Fading Channel

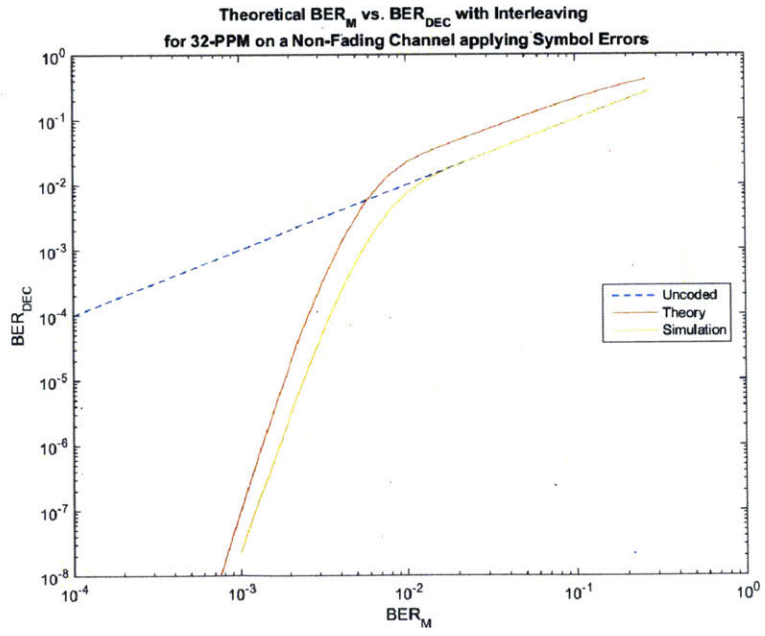

Figure 3-15: BER_M vs. BER_{DEC} for 32-PPM used with RS(255,239) on a Non-Fading Channel

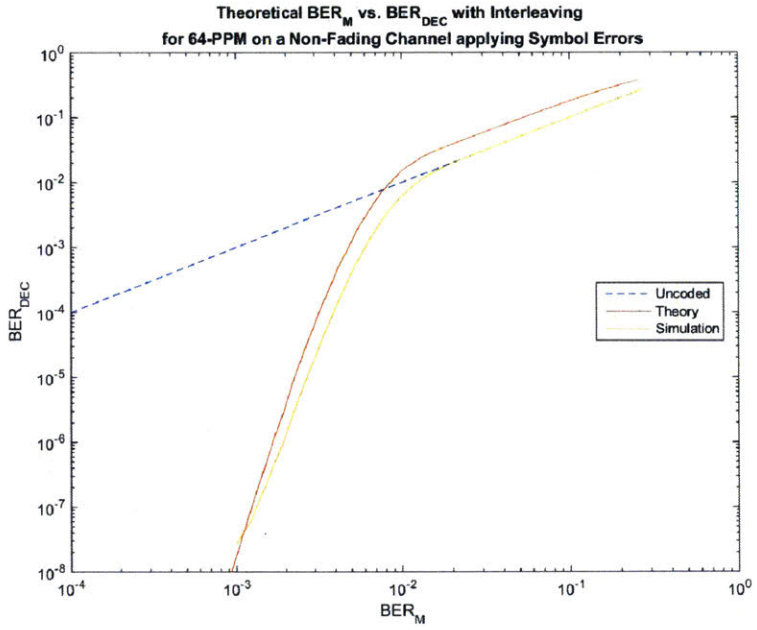

Figure 3-16: BER_M vs. BER_{DEC} for 64-PPM used with RS(255,239) on a Non-Fading Channel

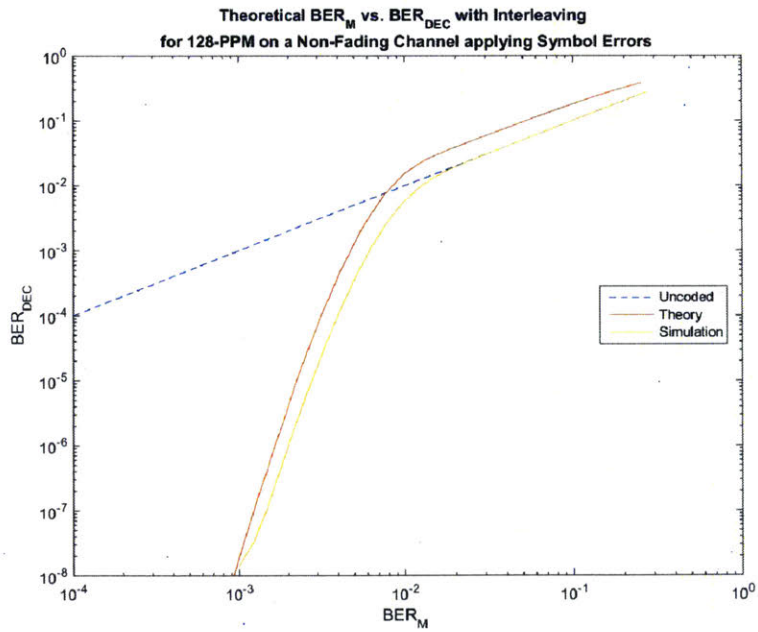

Figure 3-17: BER_M vs. BER_{DEC} for 128-PPM used with RS(255,239) on a Non-Fading Channel

As Figures 3-12 to 3-17 show, lower PPM orders see worse performance than

higher PPM orders due to PPM symbol mapping, which agrees with theoretical estimates. Moreover, tests show that the unireedsolomon codec will enable error-free communications at a channel BER of 1×10^{-4} when PPM symbol mapping is used.

3.4 Interleaving

Interleavers are typically used with error-correcting codes to distribute channel errors in a way that the error-correcting code can handle. Interleaving separates symbols from the same codeword in time in order to transform a channel with memory into a memoryless channel, which makes burst errors appear more random to the decoder [73]. Due to the characteristics of the optical channel, interleaving plays a major role in providing error-free communications. A block interleaver was chosen to perform the interleaving, and a custom block interleaver was designed in Python.

3.4.1 Channel Effects

The optical channel from LEO to ground is characterized by fades on the order of milliseconds that cause burst errors in the data [23, 48, 40, 86, 41]. Burst errors are correlated errors between symbols, and without interleaving, burst errors can result in many errors occurring in one codeword. As mentioned above, linear block codes, such as Reed-Solomon codes, can only correct a certain amount of errors per codeword, and an entire codeword may be lost if the error-correcting capability of the codeword is surpassed. Millisecond-class fades result in the loss of many Reed-Solomon codewords of length 255, which requires the use of an interleaver. An interleaving depth of one second was chosen since it is a common design practice to select an interleaving depth of 100-1000 times the coherence time of the channel fading process, which as stated above, is on the order of milliseconds [23].

3.4.2 Block Interleaver

A block interleaver was chosen over a convolutional interleaver since the interleaving will not be performed in real-time and since the satellite is assumed to have sufficient memory for interleaving. Convolutional interleavers are very attractive in many applications because they have one-half of the delay and require one-half of the memory compared to block interleavers [73]. The block interleaver could be substituted by a convolutional interleaver in future versions if memory becomes a concern and real-time processing becomes a requirement.

3.4.3 Developing Custom Block Interleaver in Python

A custom block interleaver was designed in Python. The block interleaver stores each codeword as a row then reads out the values column-by-column to perform the interleaving. In Python, this is accomplished by creating a list of lists, where each codeword is a list contained within the main list. To read out by columns, the first

element of each codeword is read out, then the second element, until the 255th element of every codeword has been read out and interleaving is complete.

3.4.4 Validation

The interleaving was validated visually with text files and gray scale images. A text file was encoded and converted into a grayscale image, where the ASCII value of each text character was converted into a grayscale value from 0 (black) to 255 (white). This image is shown in Figure 3-18.

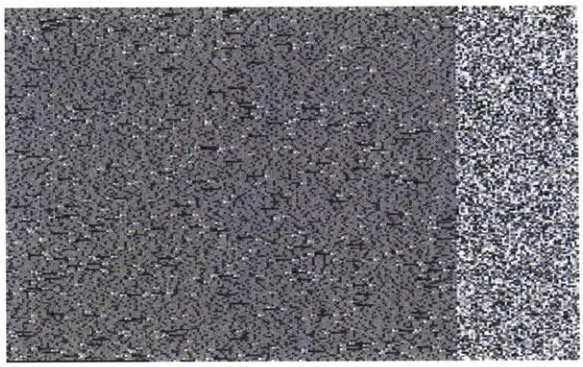

Figure 3-18: Encoded Text File

Since text characters typically only use ASCII values 32-127, they mostly compose black and gray colors, as can be seen on the left side of the image. The parity bits added by the encoder encapsulate the full range of ASCII values, so they appear brighter on the right hand side of the image. The interleaved version of Figure 3-18 is shown in Figure 3-19.

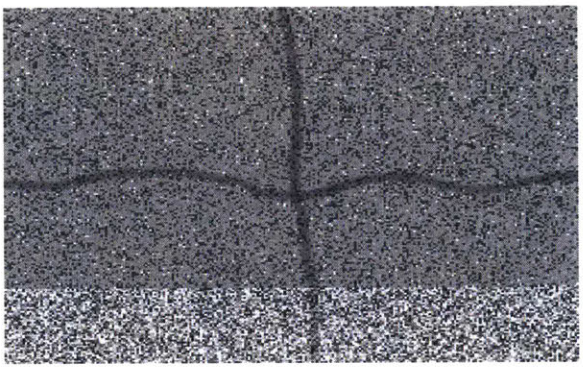

Figure 3-19: Encoded and Interleaved Text File with Errors Added

When compared to the encoded text file, the encoded and interleaved file has the rows and columns switched. This is confirmed in the fact that the parity bits are shown on the bottom of the image, meaning codewords are aligned down columns.

This acts as proof that the interleaver is working correctly because it reads out columns of the encoded file and fills them row-by-row to construct the interleaved image. Assuming row 1 is transmitted before row 2 in the interleaved image, the interleaving correctly interleaves over all codewords. The black lines on the image are errors inserted by an image editor after interleaving. Figure 3-20 shows the deinterleaved version of Figure 3-19.

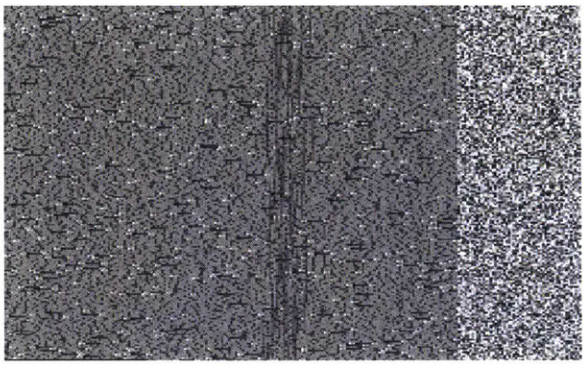

Figure 3-20: Deinterleaved and Encoded Text File with Errors Added

The deinterleaved image shows that the errors are spread across multiple codewords, which is exactly what the interleaver and deinterleaver are meant to do.

3.4.5 Testing

The interleaver test block diagram is shown in Figure 3-6, and the test is intended to demonstrate that the interleaver is essential to obtain error-free communications when errors are correlated, such as across a fading channel. Figures 3-7 and 3-9 show that theoretical performance is achieved with RS(255,239) across a non-fading channel whether or not interleaving is used. Once a fading channel is used, RS(255,239) is not able to correct the errors as well. To model the fading experienced in the optical channel, the simulations use a channel coherence time of 1 millisecond. For the purposes of testing, the fades are set to be less than or equal to 1 millisecond, meaning below a BER_{RS} of 1×10^{-3}, the length of fade is equal to the BER and above a BER_{RS} of 1×10^{-3} multiple fades occur. The fades were inserted by selecting, at random, 1 millisecond of symbols and randomly assigning each symbol one out of all possible M symbol values, resulting in a binary fading channel. A temporal depiction of the binary fading channel is shown in Figure 3-21. A fade occurs in Figure 3-21 when the attenuation drops to -100 dB, and errors are dependent, or correlated, within a fade. Figure 3-22 shows the performance of RS(255,239) on a fading channel with a coherence time of 1 millisecond. Curves are shown with and without interleaving.

Figure 3-22 demonstrates that not using an interleaver across a fading channel with millisecond class fades, such as the optical channel, is comparable to not using channel coding. Fortunately, Figure 3-22 also shows that an interleaver can successfully mitigate the effect of the fades, thereby enabling the RS performance to match

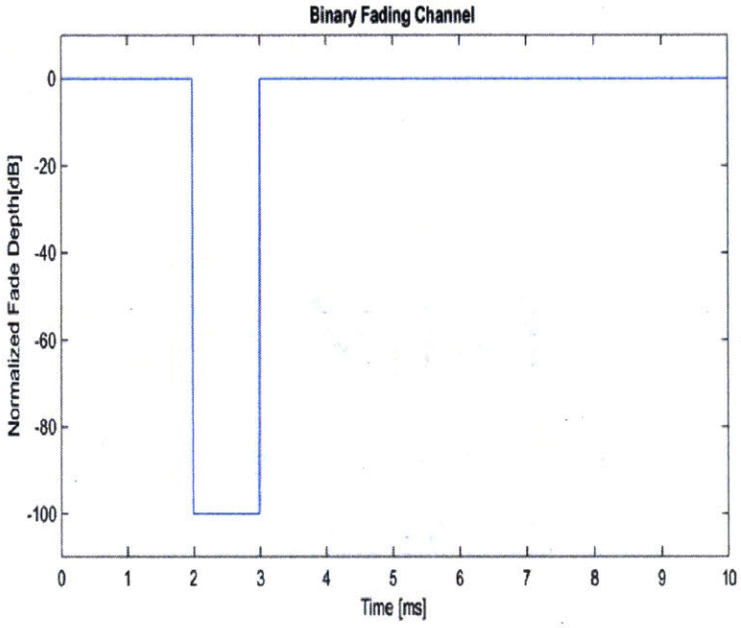

Figure 3-21: Temporal Representation of the Binary Fading Channel Model

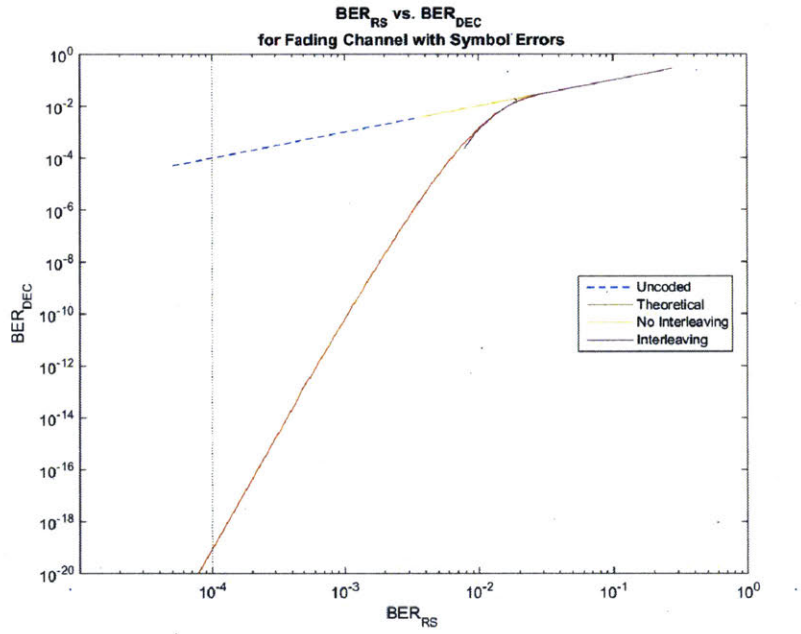

Figure 3-22: Performance of RS(255,239) for a Fading Channel with a Coherence Time of 1 Millisecond with and without Interleaving

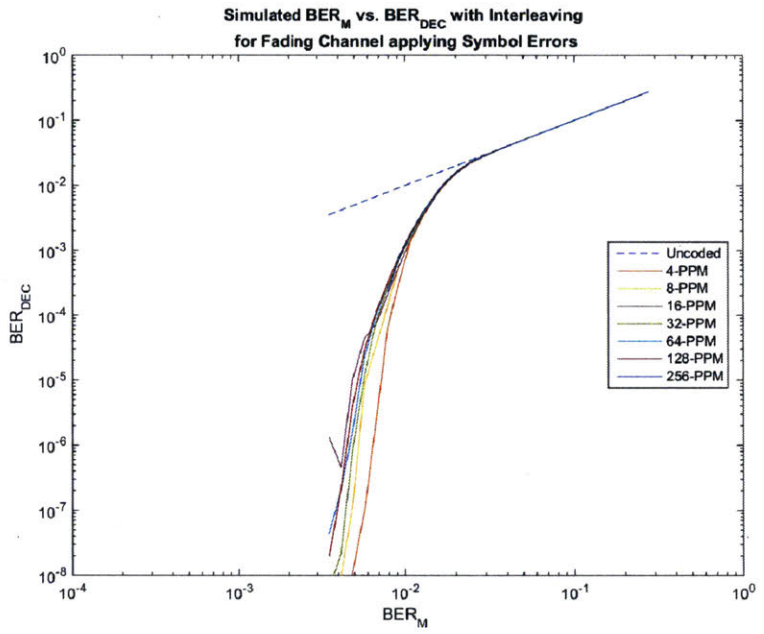

Figure 3-23: Performance of RS(255,239) for a Fading Channel with a Coherence Time of 1 Millisecond with Interleaving for all PPM Orders

that of theory. When transmitting across the optical channel, PPM symbol mapping will be used, meaning the fades will occur across PPM symbols. When a fading channel with a coherence time of 1 millisecond is applied to the PPM symbols, the negative effects of PPM symbol mapping are diminished as multiple PPM symbol errors are packed into the same (or fewer) RS symbol(s), preventing the error multiplication caused by PPM symbol mapping described in Section 3.3.2. Figure 3-23 shows the improved performance over a fading channel, and Figure 3-24 explains why the improvement in performance occurs.

The negative effects of PPM symbol mapping are reduced in Figure 3-23 because fewer RS symbol errors occur when PPM symbol errors are correlated. This happens because, as mentioned earlier, RS codes can correct all of the bit errors within one RS symbol, and when PPM symbol errors are correlated, more bit errors are packed into each RS symbol. In addition, the error multiplication explained in [35] is less severe when the PPM errors are correlated because the probability that one PPM symbol error causes two RS symbol errors decreases while the probability that one RS symbol error contains multiple PPM symbol errors increases. Correlated PPM symbol errors prevent the SER_{RS} from increasing due to PPM symbol mapping. Figure 3-24 shows this.

In Figure 3-24 each box represents a symbol, where "8-PPM" represents an 8-PPM symbol and "RS" represents an RS symbol. The SER_M is the same for both cases, but the uncorrelated error case causes 3 RS symbol errors whereas the correlated error case only causes 1 RS symbol error. Random, uncorrelated PPM symbol errors will cause at least as many RS symbol errors as correlated PPM symbol errors, but it

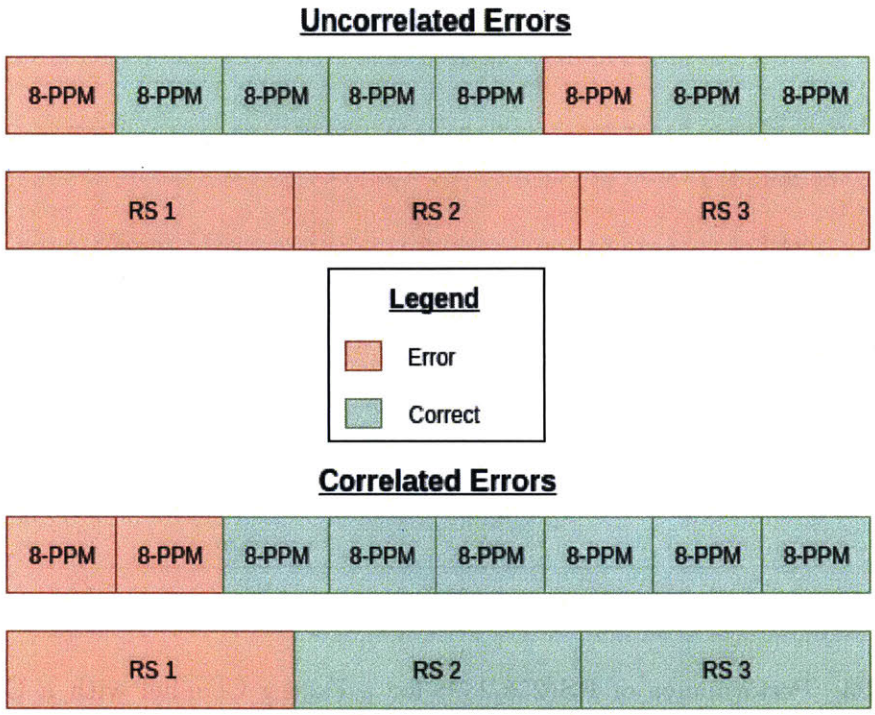

Figure 3-24: Impact on RS Symbol Errors when PPM Symbol Errors are Uncorrelated and Correlated

is more likely that uncorrelated PPM symbol errors will cause more RS symbol errors than correlated PPM symbol errors. For this reason, the negative effects of PPM symbol mapping with RS(255,239) are not as severe on a fading channel. Figures 3-25 to 3-30 show the performance of RS(255,239) with interleaving and PPM symbol mapping for all PPM orders on a fading channel with a 1 millisecond coherence time.

The jagged portion of the simulation curve in Figure 3-27 is due to uncertainty in the BER since few errors were accumulated at low BERs. The variability of a simulated BER value is inversely proportional to the number of errors accumulated, and few errors were accumulated due to limited processing power. Figures 3-25 to 3-30 show that, on a binary fading channel, the performance of each PPM order is better than the theory suggests and approaches the performance obtained when the RS codeword size is matched to the PPM order. Fortunately, the error-correcting performance of RS(255,239) will not be degraded since the optical channel is a fading channel, thereby mitigating the negative effects of PPM symbol mapping, and interleaving will be used to mitigate the effects of the fades. While promising, these results only apply to the binary fading channel model shown earlier. A more realistic channel model would include fades of various durations and would include many different attenuation values. The results shown here represent the two extremes: degraded performance on a binary non-fading channel and nominal performance on a fading-channel. Performance on a more realistic channel model would lie somewhere between these two extremes.

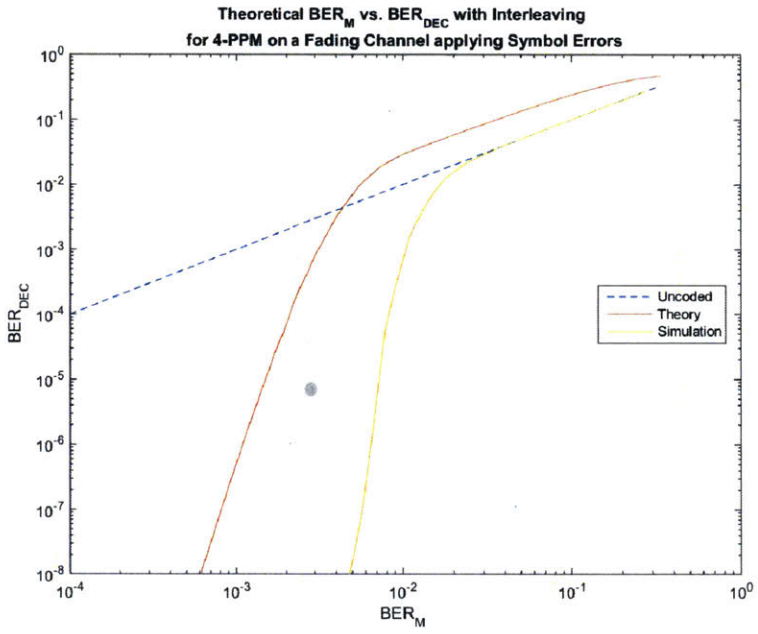

Figure 3-25: BER_M vs. BER_{DEC} for 4-PPM used with RS(255,239) on a Fading Channel with a 1×10^{-3} sec Coherence Time

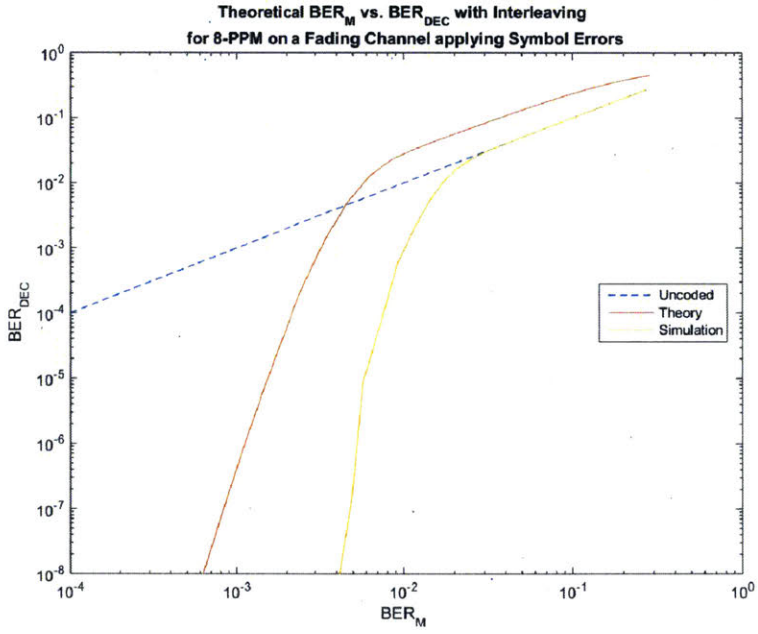

Figure 3-26: BER_M vs. BER_{DEC} for 8-PPM used with RS(255,239) on a Fading Channel with a 1×10^{-3} sec Coherence Time

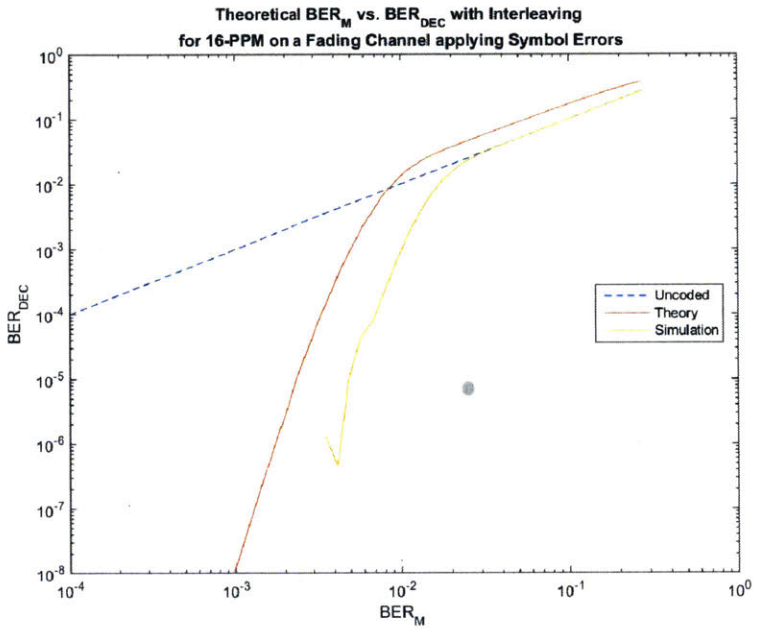

Figure 3-27: BER_M vs. BER_{DEC} for 16-PPM used with RS(255,239) on a Fading Channel with a 1×10^{-3} sec Coherence Time

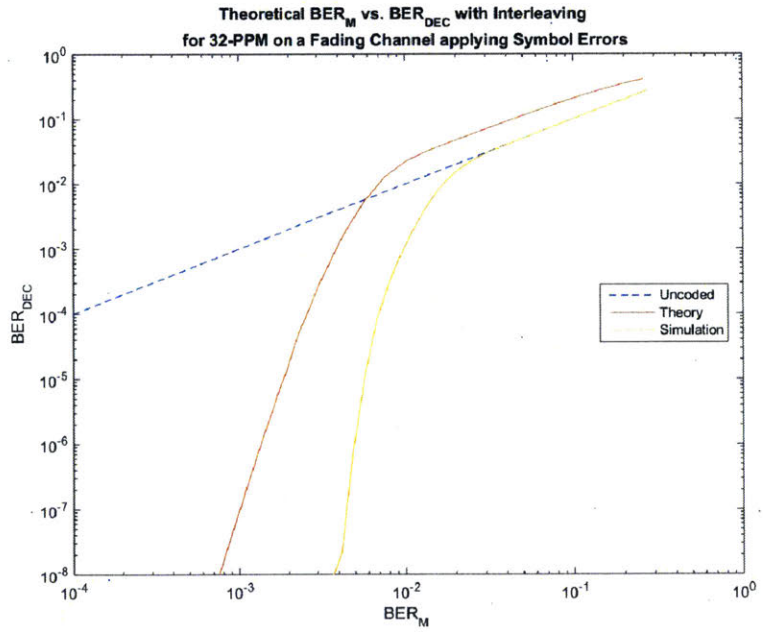

Figure 3-28: BER_M vs. BER_{DEC} for 32-PPM used with RS(255,239) on a Fading Channel with a 1×10^{-3} sec Coherence Time

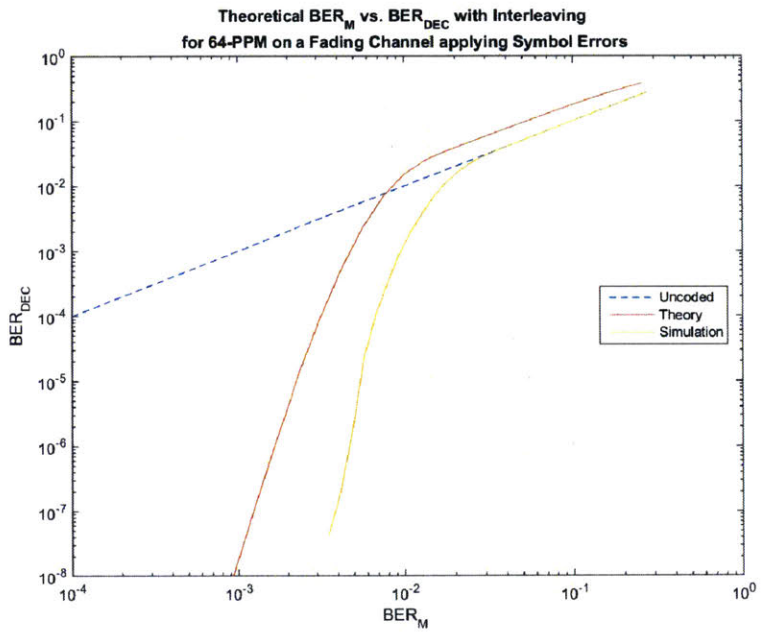

Figure 3-29: BER_M vs. BER_{DEC} for 64-PPM used with RS(255,239) on a Fading Channel with a 1×10^{-3} sec Coherence Time

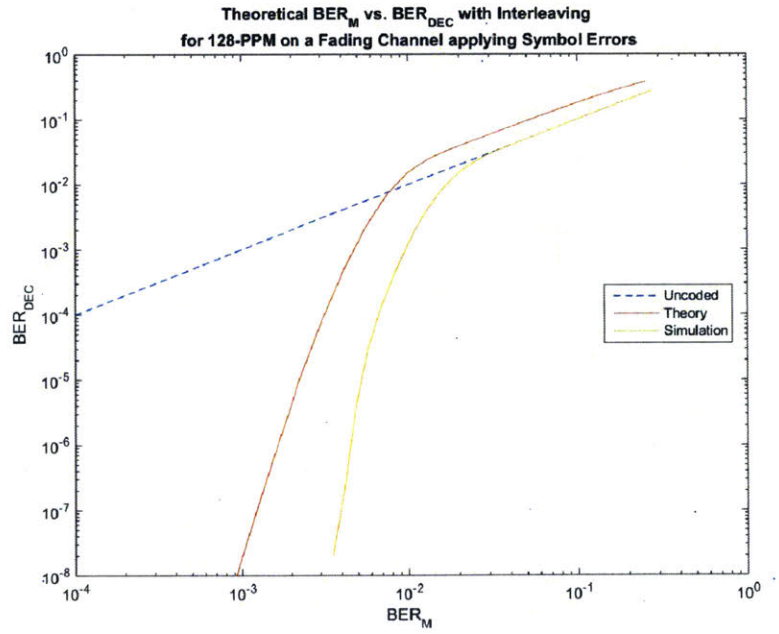

Figure 3-30: BER_M vs. BER_{DEC} for 128-PPM used with RS(255,239) on a Fading Channel with a 1×10^{-3} sec Coherence Time

3.5 Framing

Framing includes the acquisition sequence and inter-symbol guard time (ISGT) used within the NODE waveform. The acquisition sequence and ISGT are fixed for each PPM order. Since NODE interleaves over one frame's worth of data, the frame structure is set by the channel effects and the interleaving needed to mitigate those effects. The frame structure is similar to that employed in [58]. The frame structure for NODE is shown in Figure 3-31.

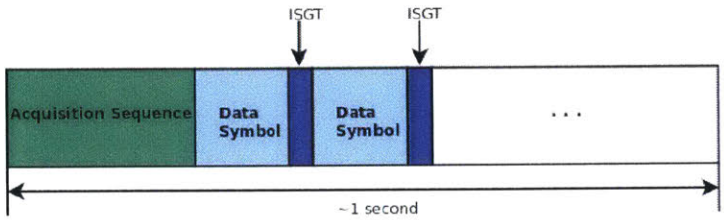

Figure 3-31: Frame Structure Employed for NODE

3.5.1 Acquisition Sequence

The acquisition sequence is used to denote the beginning of an interleaved frame of data. A desirable property for the acquisition sequence is to have good autocorrelation features[3], ensuring that the beginning of the frame is clearly denoted, even while in the presence of errors. For PPM orders greater than or equal to 16-PPM, the acquisition sequence used is the same one employed in [6, 82]. However, different acquisition sequences were generated for 4-PPM and 8-PPM as no recommendation was given in [6]. Table 3.2 shows the acquisition sequence for each PPM order.

Table 3.2: Acquisition Sequences for Various PPM Orders

PPM Order	Acquisition Sequence (Symbol Numbers)
4	2,3,2,3,0,2,2,1,3,3,0,1,0,1,1,2,1,0,2,2,3,1,1,0,3,0,0,2,2,0,0, 2,0,2,0,3,1,1,2,1,0,0,1,1,3,3,3,1,0,3,0,1,2,0,3,1,3,1,0,2,3,2,3,0
8	2,4,6,3,0,3,1,2,7,7,7,5,6,7,5,7,2,4,4,4,0,3,6,1,3,5,2,1,3,2,4,6
≥ 16	0,2,7,14,1,2,15,5,8,4,10,2,14,3,14,11

The sequence for 16-PPM, 32-PPM, 64-PPM, and 128-PPM was chosen because it has good autocorrelation features, but more specifically, it has a very narrow autocorrelation and low peak side lobe level (PSL) [82]. The autocorrelation for this sequence is shown in Figure 3-32.

Since deframing is performed after demodulation, the autocorrelation is performed across symbols rather than slots, meaning the symbol energy is accumulated when

[3]Even though there are many metrics used to denote good autocorrelation features, the peak side lobe level (PSL) and the minimum amount of symbol overlap, denoting a narrow autocorrelation, are the metrics used here.

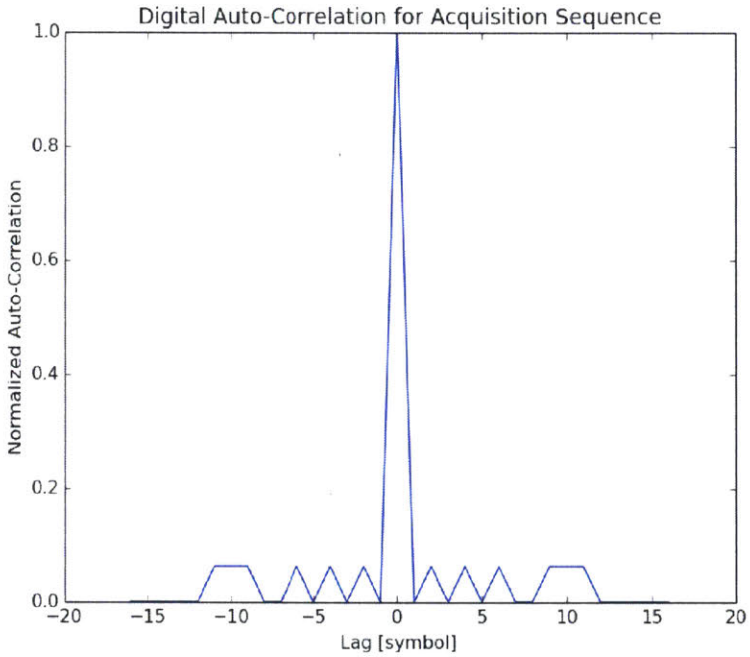

Figure 3-32: Autocorrelation for Acquisition Sequence used for PPM orders ≥ 16

two symbols match. Confirming performance in [82], Figure 3-32 shows that the autocorrelation has very low side lobes. Figure 3-33 shows a histogram of symbol overlap for the same acquisition sequence.

The symbol overlap is the number of symbols that match for any lag value in the autocorrelation. Figure 3-33 confirms that the acquisition sequence has a narrow autocorrelation since the symbol overlap is no larger than 1, except when the main peak occurs and beginning of the acquisition sequence is detected.

The acquisition sequences for 4-PPM and 8-PPM were designed to have the same probability of repeating as the sequence used for PPM orders ≥ 16 and to have good autocorrelation features. The probability that the 16-PPM sequence repeats is:

$$P(SequenceRepeats) = \left(\frac{1}{16}\right)^{16} = 54 \times 10^{-21} \tag{3.12}$$

Since a $P(SequenceRepeats)$ of 54×10^{-21} was sufficient in [82], the acquisition sequences for 4-PPM and 8-PPM were chosen to have a $P(SequenceRepeats)$ less than or equal to 54×10^{-21}. Calculations reveal that sequences of 22 symbols and 32 symbols for 8-PPM and 4-PPM, respectively, obtain a $P(SequenceRepeats)$ of 54×10^{-21}. However, to obtain better autocorrelation properties and keep the duration of the acquisition sequence the same for PPM orders 4, 8, and 16, acquisition sequences of 32 and 64 symbols are used for 8-PPM and 4-PPM, respectively. The autocorrelation for the 8-PPM acquisition sequence is shown in Figure 3-34.

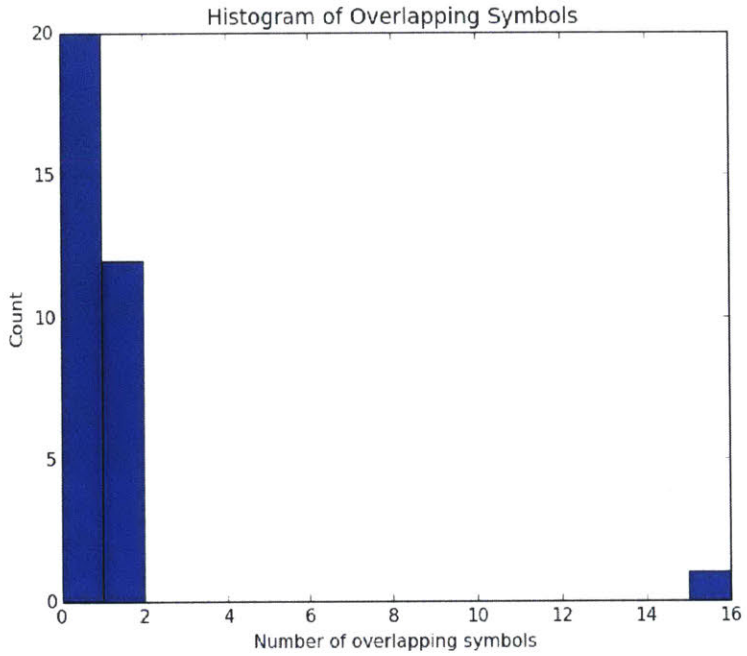

Figure 3-33: Histogram of Symbol Overlap for Acquisition Sequence used for PPM orders ≥ 16

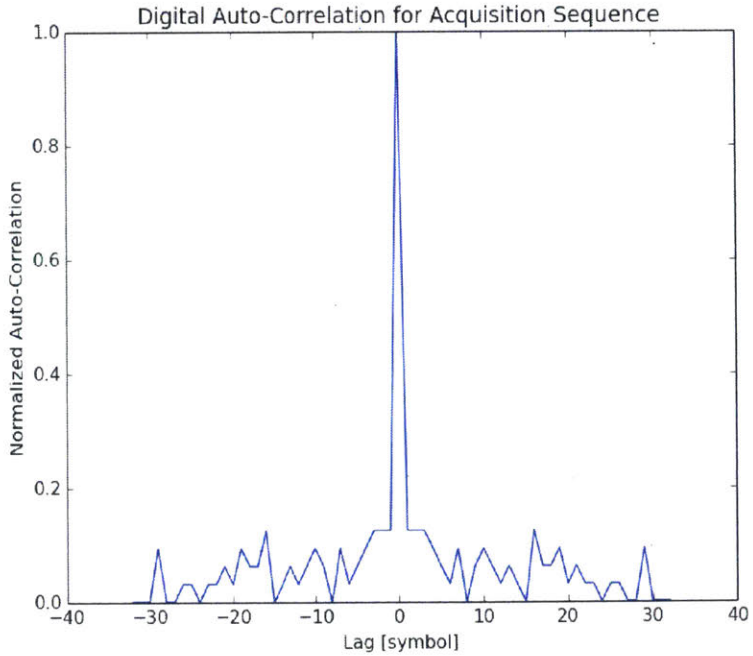

Figure 3-34: Autocorrelation for Acquisition Sequence used for 8-PPM

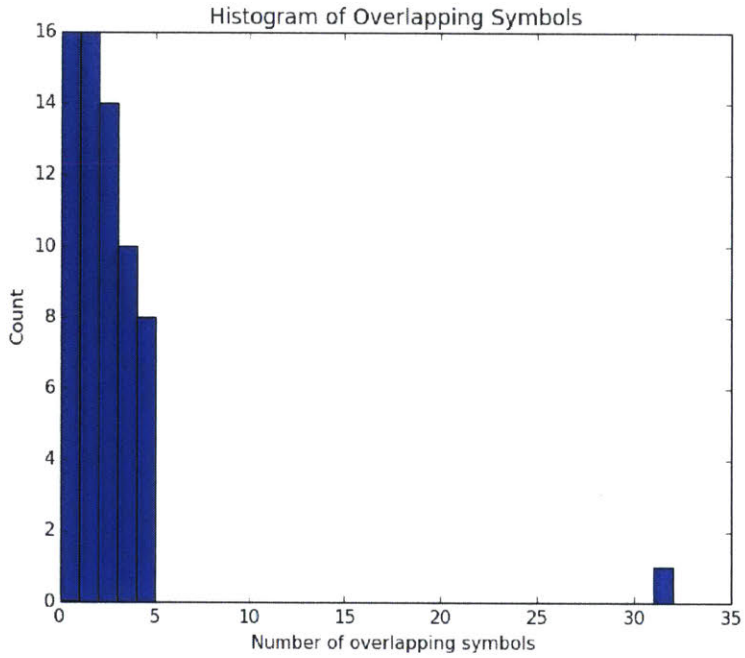

Figure 3-35: Histogram of Symbol Overlap for Acquisition Sequence used for 8-PPM

Even though the PSL is not as low as the PSL in Figure 3-32, the main lobe in the autocorrelation in Figure 3-34 is large enough to stand out from the side lobes in the presence of errors. Figure 3-35 shows the histogram of symbol overlaps for 8-PPM.

The histogram for symbol overlap in Figure 3-35 does not have a symbol overlap greater than 5 except for the main lobe. While this is not as good as that of Figure 3-33, this is acceptable since the largest overlap occurs at 0 and 1. Figure 3-36 shows the autocorrelation for the acquisition sequence used for 4-PPM.

Even though the autocorrelation in Figure 3-36 is clearly worse than that in figures 3-32 and 3-34, the PSL is still low enough for detection when in the presence of errors. Figure 3-37 shows the histogram of symbol overlaps for the 4-PPM acquisition sequence.

Unfortunately, the histogram in Figure 3-37 does not have the largest symbol overlaps at 0 and 1. However, this sequence will be good enough since there is a significant separation between the peak side lobe and the main lobe.

A random sequence generator was used to generate the acquisition sequences for 4-PPM[4]. and 8-PPM. The random sequence generator was run a fixed number of times to produce a large amount of sequences. After performing autocorrelations on the randomly generated sequences, the PSL for each sequence was recorded. Among all sequences where the maximum symbol overlap value corresponded to 0 symbol

[4]The acquisition sequence for 4-PPM uses a random sequence of length 38 appended to the sequence provided in [45] to improve the autocorrelation features.

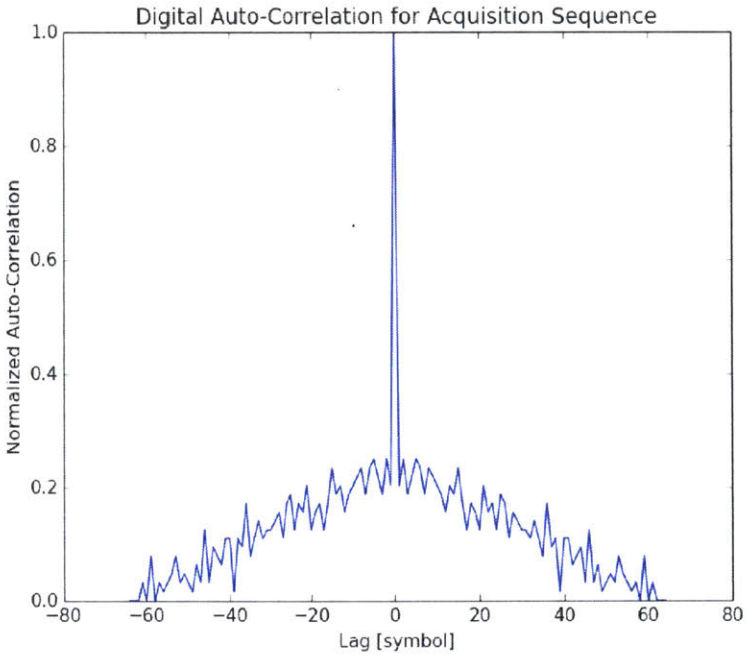

Figure 3-36: Autocorrelation for Acquisition Sequence used for 4-PPM

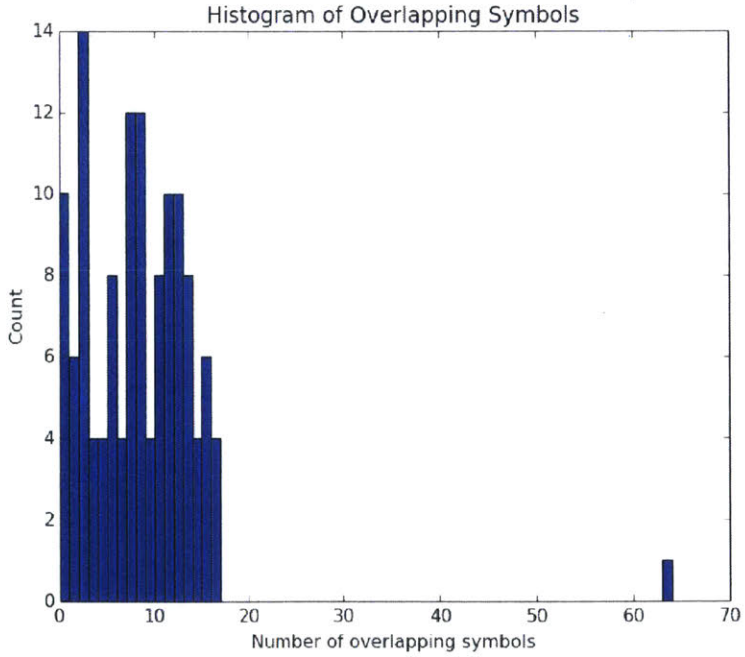

Figure 3-37: Histogram of Symbol Overlap for Acquisition Sequence used for 4-PPM

overlap, the sequence with the maximum PSL was returned. Due to the brute-force method employed and the length of the sequences, the chosen acquisition sequences are almost certainly not the best possible acquisition sequences as far as the autocorrelation metrics are concerned, but they will be sufficient for the purposes of the lasercom downlink. If testing shows that the acquisition sequences need better autocorrelation properties, a more rigorous approach to designing sequences with good autocorrelation properties is explained in [45, 72].

3.5.2 Inter-Symbol Guard Time

The NODE transmitter employs an ISGT between symbols in order to aid in clock recovery and to prevent degradation of the EDFA's gain [48, 67].

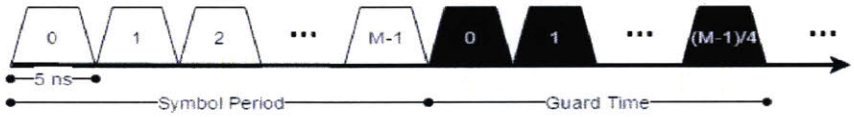

Figure 3-38: PPM Symbol with Inter-Symbol Guard Time

The ISGT is very helpful in performing symbol and slot clock recovery because a correlation can be performed with the dead time to extract the symbol clock [67]. As shown in Figure 3-38, the ISGT takes up one-quarter [6] of a symbol period for every PPM order so that the same amount of symbols may be accumulated for each PPM order when recovering the symbol clock. Additionally, the dead time prevents back-to-back pulses from arriving at the EDFA, which results in a degradation in the peak power output of the EDFA due to the long amplification period. This has been observed in the lab.

3.6 Padding

The padding function ensures that the interleaver will always interleave over one second of data. This is accomplished by appending padding bytes to the source file so that the amount of data and padding bytes combined is equal to the data contained within an integer number of frames. The depadding function is the inverse of the padding function, and both functions are shown in Figures 3-1 and 3-41. A block diagram showing the bytes added by the padding operation is shown in Figure 3-39.

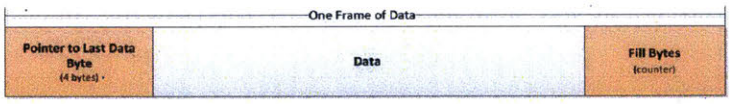

Figure 3-39: Block Diagram of Padding for One Frame of Data

The padding function accepts data, determines the number of bytes in the data, prepends a pointer to the data, known as the end-of-data pointer, which points to

the index of the last data byte, and adds bytes as a counter to pad the source file to a length equal to the amount of bytes per frame. If the data contains more bytes than can be fit into one frame, the padding function fills the first N frames completely with data then adds padded bytes to the end of the last frame. For NODE, the data is obtained from a source file that is loaded as a binary file into the padding function. The intent of the padding function is to ensure that the interleaver always has one frame's worth of data to interleave over.

The padding function must know how much data is required in a frame given the PPM order, interleave depth, and the n and k used for encoding. Since PPM is the modulation format, slots are 5ns in duration, and guard slots are one quarter of a symbol period, the bit rate out of the modulator is

$$bitrate = \frac{\log_2(M)}{(5e{-}9 * 1.25 * M)} \tag{3.13}$$

where M is the ppm order. Since there are 8 bits per RS symbol, the RS symbol rate out of the modulator is

$$symbolrate = \frac{bitrate}{8} \tag{3.14}$$

The required number of codewords per frame of data is

$$codewords = \left\lceil \frac{symbolrate * interleavedepth}{n} \right\rceil \tag{3.15}$$

Then, the number of message symbols allowed per frame is

$$messagesymbols = (codewords * k) - 4 \tag{3.16}$$

where four bytes are subtracted since the end-of-data pointer is 4 bytes long.

Validation and Testing

The padding function was validated by using three source files for each PPM order: one containing an amount of data equal to a frame of data, one containing less than a frame of data, and the last containing greater than a frame of data. A Python script, used to test the padding function, loaded the source file and called the padding function to generate one or two frames of padded data. The script read the 4 byte pointer inserted at the start of the data for each frame. Next, the script stored the data specified by the pointer and compared that data to the data in the source file. Proving both were equal was sufficient to validate the padding function.

3.7 Mapping

The bit-to-symbol mapping function converts a padded, encoded, and interleaved block of data into PPM symbols and inserts frame bytes to command the modulator to perform framing. The symbol-to-bit mapping function accepts symbol values

from the demodulation block and converts the symbols into a bitstream, which is then converted into RS symbols for deinterleaving and decoding. Both functions are shown in Figures 3-1 and 3-41. In the bit-to-symbol mapping function, a frame byte is added to the front of each frame, one frame of encoded and interleaved data is mapped to PPM symbols so that each byte includes $\log_2(M)$ significant bits, and a tracking byte is appended to each frame of PPM symbols. The frame byte denotes the start of a frame and tells the modulator to send the acquisition sequence before modulating data. The tracking byte informs the modulator that the end of the frame has been reached and puts the modulator in the tracking mode, where the modulator continually sends the 0th slot. The tracking mode is intended to ensure that energy always hits the detector to maintain spatial acquisition. Each PPM symbol is contained within one byte due to the interface between the CPU and FPGA, which only operates on bytes. The output of the bit-to-symbol mapping function is a file that may be read in Python and loaded into the FPGA, and this file will be known as the modulator (mod) file. The mod file includes frame bytes, tracking bytes, and bytes containing PPM symbols, which will be referred to as modulation bytes. Figure 3-40 is a block diagram showing the framing/tracking byte and data byte.

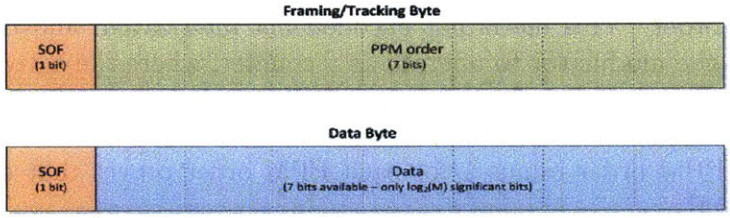

Figure 3-40: Framing/Tracking Byte and Data Bytes after Bit-to-Symbol Mapping

Since the bit-to-symbol mapping operation converts bytes with 8 significant bits to bytes with $\log_2(M)$ significant bits, the mod file is larger than the padded and encoded data. To apply the bit-to-symbol mapping operation to all of the data, the amount of data in a frame must be divisible by the number bits per PPM symbol. For example, given 128-PPM, the bit-to-symbol mapping operation needs 7 data bytes at a time in order to produce a stream of modulation bytes without introducing errors. To ensure that the number of bytes in the padded, encoded, interleaved data is divisible by the number of bits per symbol, the amount of message symbols, as shown in Equation 3.16 is adjusted. The number of codewords per frame in Equation 3.15 is continually incremented by 1 until the number of message symbols per frame in Equation 3.16 is divisible by the number of bits per symbol. This is accomplished by checking to see if the number of bits per symbol is divisible by the number of message symbols, incrementing the number of codewords by 1 if divisibility is not obtained, and calculating the number of message symbols in Equation 3.16 until divisibility is obtained.

Exactly one symbol is included in each modulation byte for PPM orders 16, 32, 64, and 128, but more than one symbol is included in each modulation byte for PPM orders 4 and 8. Three PPM symbols are included in each modulation byte for 4-

PPM while two PPM symbols are included in each modulation byte for 8-PPM. This convention is followed to pack each modulation byte with the maximum amount of PPM symbols, thereby decreasing the rate at which modulation bytes are read into the FPGA for low PPM orders. Decreasing the rate that modulation bytes are read into the FPGA is important to prevent the FIFO on the FPGA from emptying when the data rate out of the modulator is the highest.

The inverse of the bit-to-symbol mapping function is the symbol-to-bit mapping function, which takes PPM symbols and the PPM order as inputs and outputs a bitstream. By concatenating the binary representations for each PPM symbol in a frame, the PPM symbols are converted to RS symbols for the deinterleaver.

3.7.1 Validation and Testing

Validation for the bit-to-symbol mapping function is accomplished in two steps. The first step is a quick check in Python verifying the frame byte location, the values of the modulation bytes, and the size of the file. The second step is to load the mod file into a Python script, which checks that the bytes in the mod file are correct. This is accomplished by creating a source file that is equal to the length of a frame of data for each PPM order. This file is not encoded nor interleaved, and the contents of the file call for the modulator to operate as a counter, where the active slot for each symbol repeatedly progresses from 0 to the PPM order minus 1 for the duration of a frame. Showing that the mod file calls for the modulator to function as a counter from 0 to the PPM order minus 1 for each PPM order proves that the mod file is operating correctly.

The symbol-to-bit mapping function is validated with the software modulator file in Section 3.8. After stripping the acquisition sequence from one software-modulated frame, the data in the software modulation file and PPM order are passed to the symbol-to-bit mapping function. The output of the symbol-to-bit mapping function is compared to the input of the bit-to-symbol mapping function, and showing they are equal is sufficient to validate the symbol-to-bit mapping function.

The bit-to-symbol mapping and symbol-to-bit mapping functions are validated in the encoder test using PPM symbol mapping. As shown in Figure 3-6, bit-to-symbol mapping and symbol-to-bit mapping are used to send symbols across the channel and recover the binary representation for those symbols. Recovering error-free data after every block in the receive chain (symbol-to-bit mapping , deinterleaving, and decoding) at low error-rates confirms that the bit-to-symbol mapping and symbol-to-bit mapping functions are working properly together.

3.8 Software Modulator

3.8.1 Design

A software modulator is designed to mimic the performance of the modulator, which is implemented in the FPGA, for testing purposes. Implemented in Python, the

software modulator accepts a mod file as the input and outputs the symbol values corresponding to the active slot in each symbol. In other words, the output of the software modulator is a file that represents the "truth" data that the demodulator is attempting to recover. The acquisition sequence is inserted for each frame in the software modulation file, and each byte in the software modulation file corresponds to one PPM symbol. The output of the software modulator will be known as the software modulation file.

3.8.2 Validation

The software modulator is validated in two steps. The first step confirms that the software modulation file is the correct size in relation to the input mod file. The second step uses a mod file with known and periodic bytes (all zeros, all ones, or a counter) to confirm that the output perfectly creates the expected modulated sequence. Validation requires identification of the acquisition sequence in addition to confirmation that the correct PPM symbols are repeated throughout the duration of the software modulation file.

3.9 A Day in the Life of a Packet

The data goes through various steps before it modulates the laser, and this section is meant to explicitly show those steps. The block diagram in Figure 3-41 is a simplification of Figure 3-1 for the purposes of discussing each step in the waveform design.

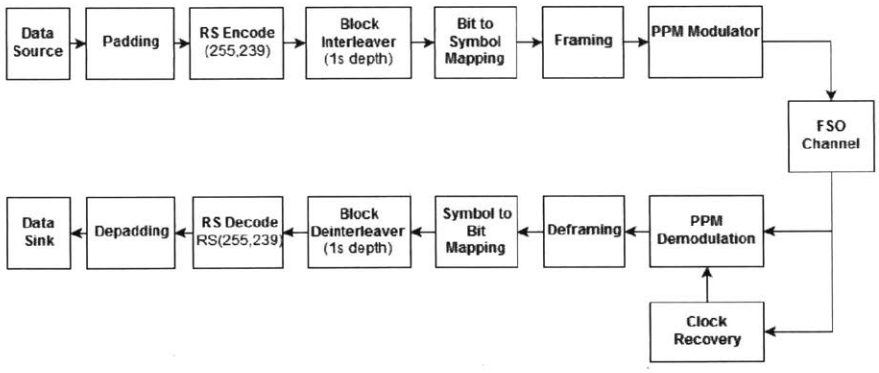

Figure 3-41: A Day in the Life of a Packet Block Diagram

Padding

The padding function prepends a pointer to the data and appends a counter to the data. The pointer is four bytes long, and its value is the index of the last data byte.

Figure 3-42: Padding Visualization

RS Encode

Encoding with RS(255,239) adds 16 parity bytes to every 239 message bytes.

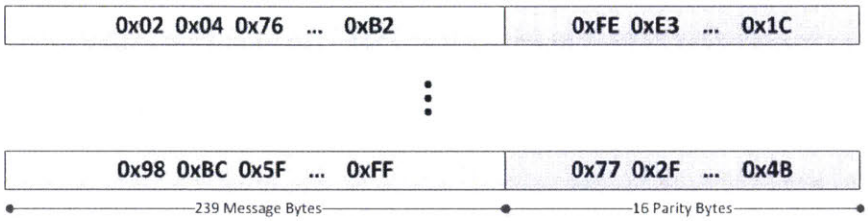

Figure 3-43: RS Encoding Visualization

Block Interleaver

Interleaving spreads out the RS symbols in each codeword to mitigate the effects of fades.

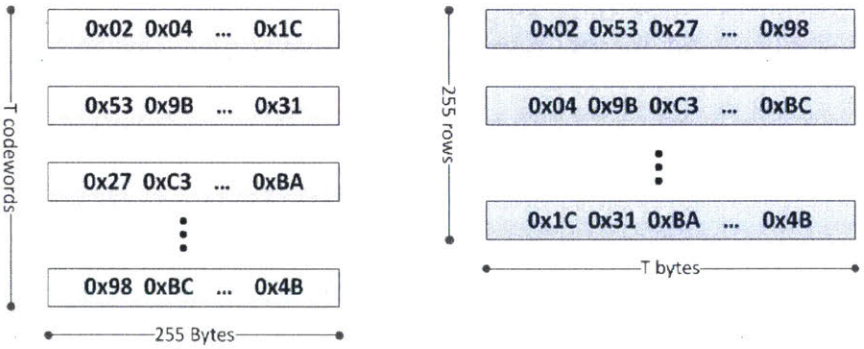

Figure 3-44: Interleaving Visualization

As displayed in Figure 3-44, codewords are written into the interleaver row-by-row and read out column-by-column. For the interleaved data, the entire first row is transmitted before the second row is transmitted.

Bit-to-Symbol Mapping

Bit-to-symbol mapping converts the encoded, interleaved data into PPM symbols and adds framing and tracking bytes. The framing and tracking bytes take the hexadecimal value "0x8F" for 16-PPM, as shown in Figure 3-45.

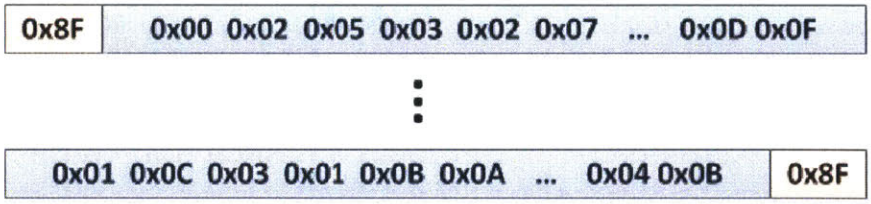

Figure 3-45: Bit-to-Symbol Mapping Visualization

Framing

Framing prepends the acquisition sequence and appends tracking symbols to the modulated data.

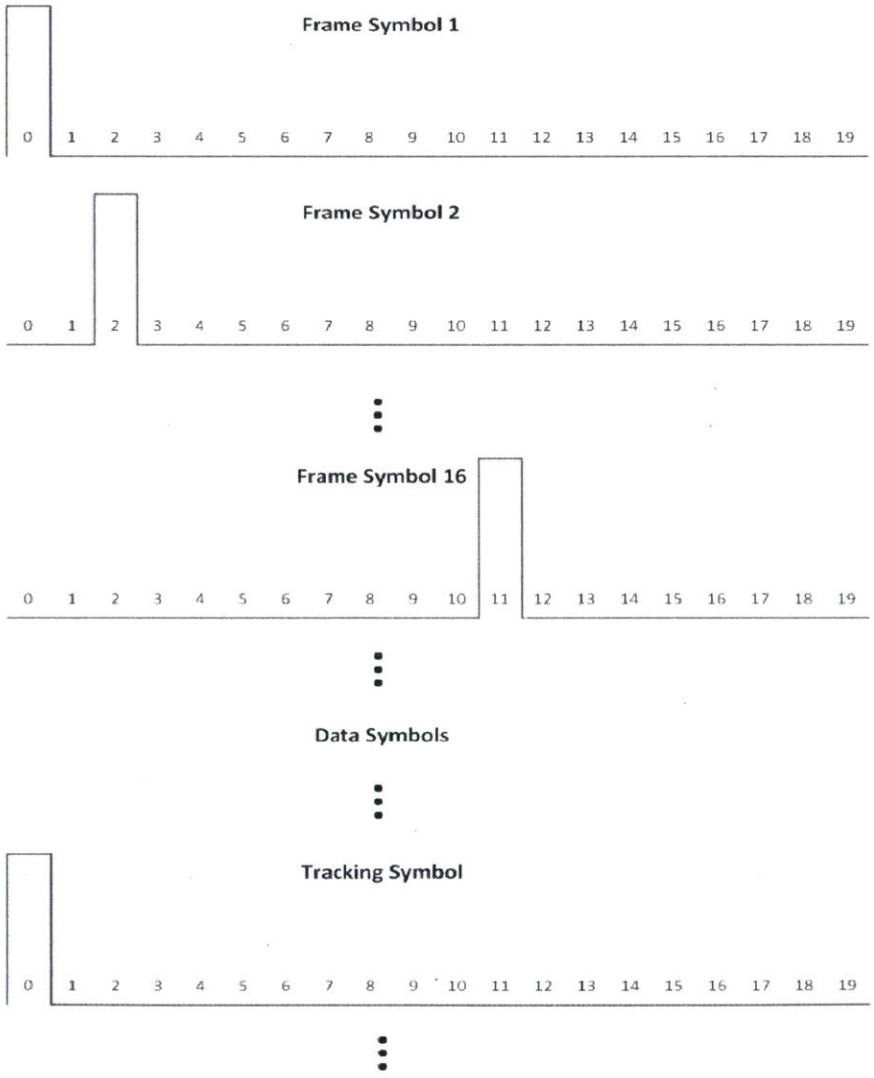

Figure 3-46: Framing Visualization

As shown in Figure 3-46, tracking symbols always send the 0th symbol when the transmitter has finished transmitting frames of data in order to aid in spatial acquisition on the receiver.

PPM Modulator

The PPM modulator is implemented in the FPGA and converts the PPM symbol numbers from the bit-to-symbol mapping into PPM symbols, which are sent directly to the laser in the form of a drive current.

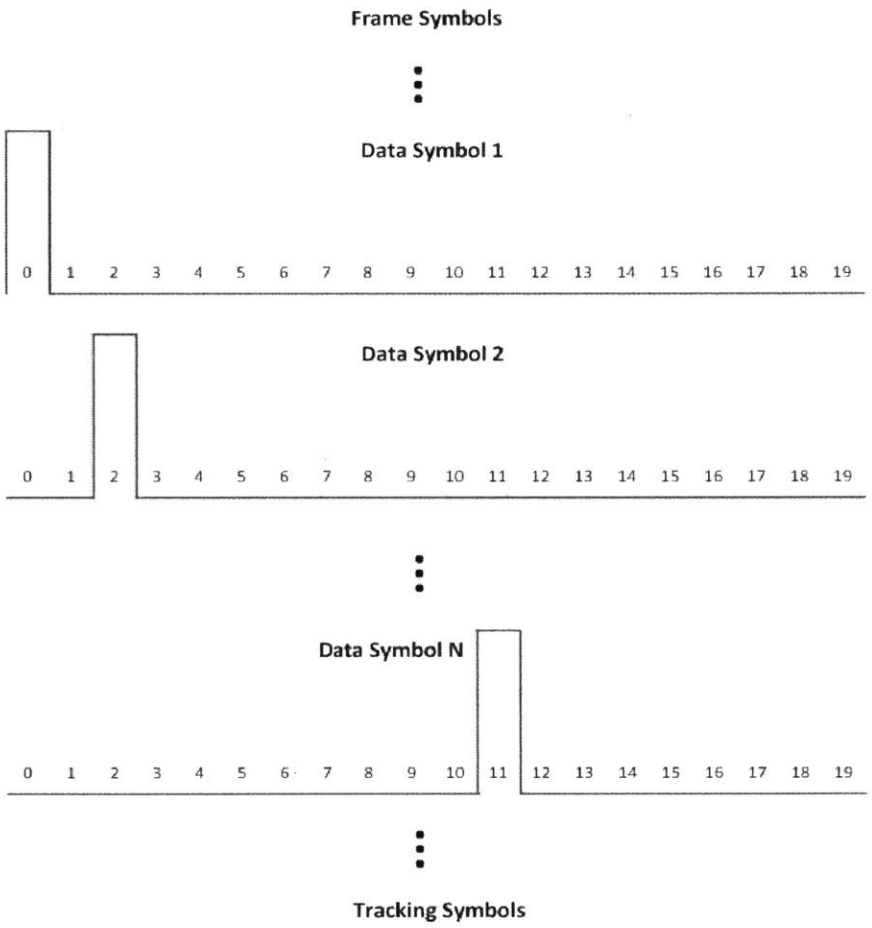

Figure 3-47: Modulation Visualization

FSO Channel

Figure 3-48 shows a sampled PPM waveform that would be fed into the demodulation and clock recovery blocks.

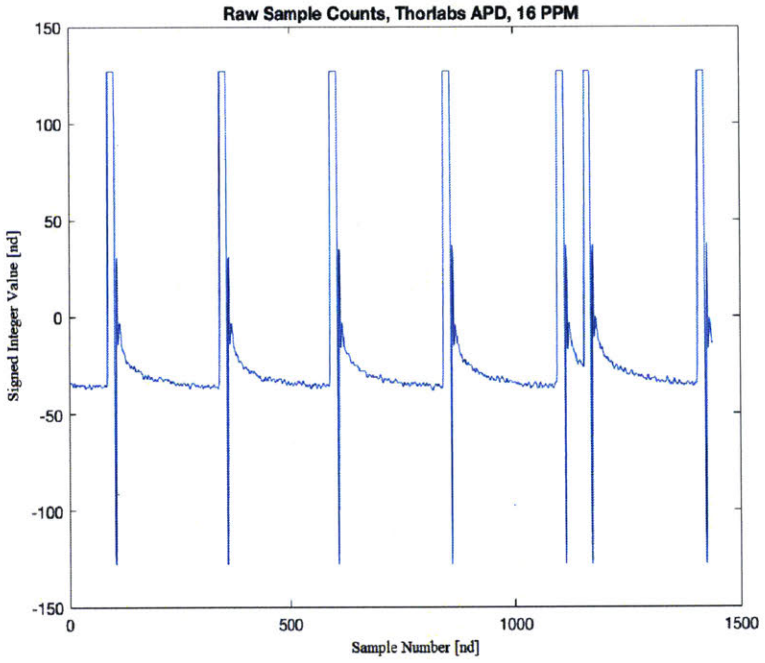

Figure 3-48: Samples Captured with Pentek Talon while Transmitting 16-PPM

PPM Demodulation and Clock Recovery

The output of the demodulation and clock recovery blocks will be PPM symbol numbers, as presented in Figure 3-49.

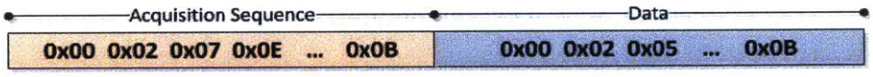

Figure 3-49: Demodulation Visualization

Deframing

Deframing uses a cross-correlation to find the acquisition sequence and removes it from the demodulated data. Additionally, deframing separates all of the demodulated data into one or more frames so that symbol-to-bit mapping and deinterleaving only occur across one frame. ground station

Figure 3-50: Deframing Visualization

Symbol-to-Bit Mapping

Symbol-to-bit mapping converts the demodulated PPM symbol numbers into a continuous bitstream, thereby making the conversion from PPM symbols to RS symbols.

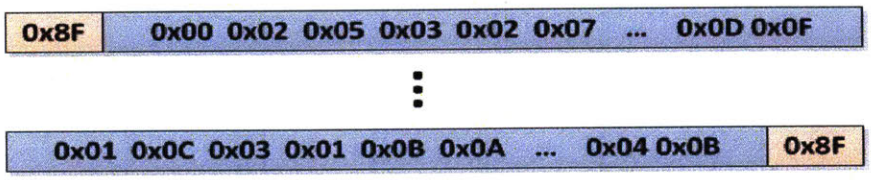

Figure 3-51: Symbol-to-Bit Mapping Visualization

Block Deinterleaver

The block deinterleaver converts the interleaved data back into blocks of codewords so that the decoder may perform decoding.

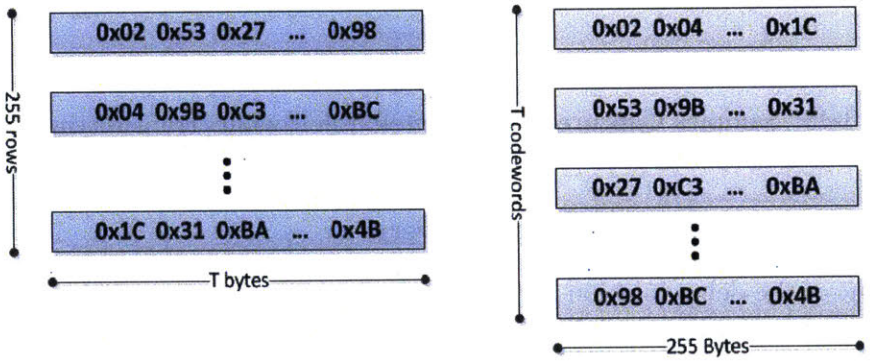

Figure 3-52: Deinterleaving Visualization

RS Decoding

Decoding corrects RS symbol errors and strips off the parity symbols. If the amount of symbol errors per RS codeword exceeds the error-correcting capability of the code, the codeword is considered in error and is released for further analysis.

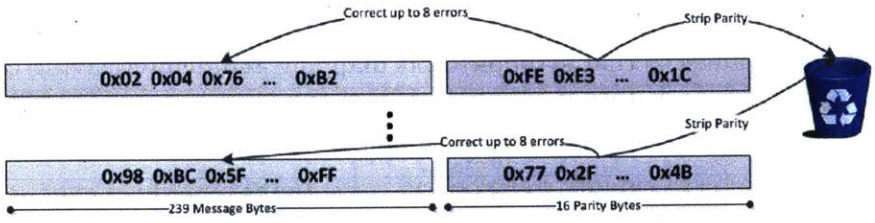

Figure 3-53: Decoding Visualization

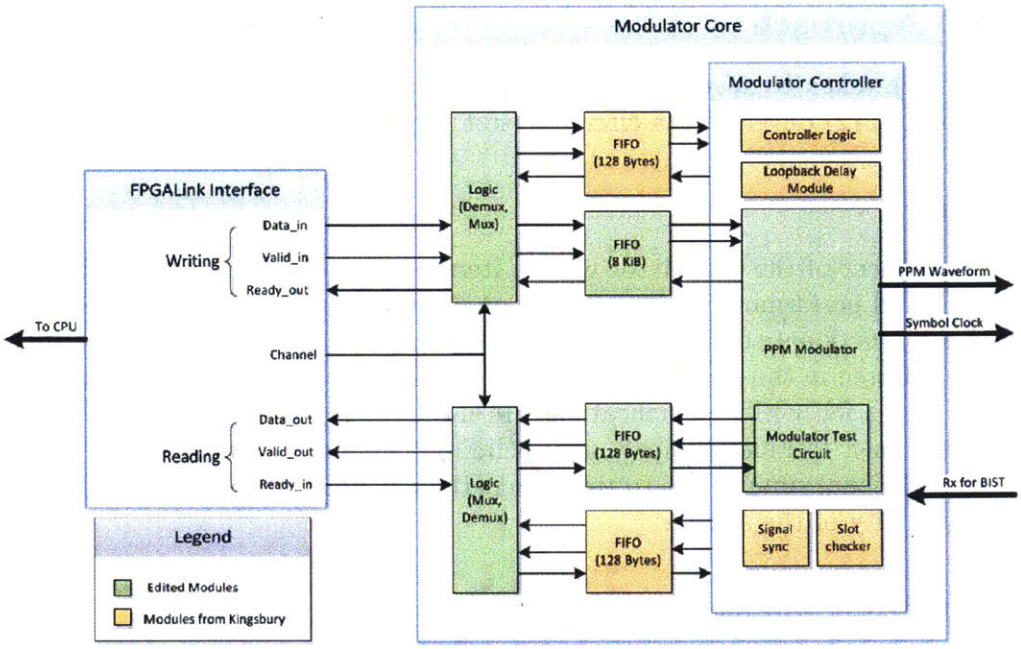

Figure 3-55: FPGA Design

Depadding

The depadding function uses the pointer to locate the last data byte, strips off the pointer and counter, then releases the data to the next layer for further analysis.

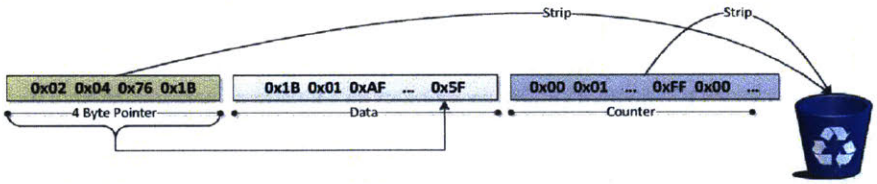

Figure 3-54: Depadding Visualization

3.10 FPGA Implementation

The FPGA implementation involves the PPM modulator, memory units, and the FPGALink interface. The FPGALink inferface is used to program the FPGA and to pass data between the CPU and FPGA. The modulator is able to perform PPM with modulation orders of 4, 8, 16, 32, 64, and 128. Also, the modulator may be put in various modes depending on the data loaded into the modulator. A block diagram of the FPGA design is shown in Figure 3-55. The FPGALink interface and modulator are connected via an 8KiB FIFO. The following sections detail the approach taken to design the modulator and leverage the FPGALink interface to successfully write data to the modulator.

3.10.1 Approach

The approach taken to develop the FPGA follow two main steps: (1) developing the modulator and (2) developing a circuit to test the performance of the modulator.

Modulator

The development of the modulator was an iterative process that involved simulating the VHDL and performing timing analysis on the synthesized design. The simulation checked for functional soundness while the timing analysis determined whether the hardware inferred in the synthesized design could accomplish the required tasks while ensuring delays fell within specifications. In most cases, registering inputs, outputs, and signals, in a technique known as pipelining, eliminated timing errors, but this required additional simulations to verify that the pipelining did not adversely impact the functionality of the design.

Modulator Test Circuit

The modulator test circuit monitors the status of the FIFO in order to keep a continuous stream of information going to the modulator; in other words, it throws a flag when the FIFO empties. Additionally, when given data that counts from 0 to the PPM order minus 1 for the duration of a frame, the modulator test circuit uses a counter from 0 to the PPM order minus 1 to check that the data being modulated is correct. If the FIFO empties or the data is incorrect, the FPGA stores the respective flag in memory and alerts the user at the end of a frame. A FIFO brings the flags from the 200 MHz domain to the 48 MHz domain, and the valid input for this FIFO is generated by an event monitor (or trigger) circuit, which pulses high upon a change in any of the input flags (empty, data error, and modulation active flags). A simplified block diagram of the modulator test circuit is shown in Figure 3-56. An iterative process was employed to design the modulator test circuit since the FIFO, valid input flag, combinatorial logic, and pipelining require multiple simulations and timing analyses. The modulator test circuit is used to validate and test the synthesized FPGA image, and this is explained in subsection 3.10.4.

3.10.2 Modulator Operation

The modulator is implemented as a finite state machine, more specifically a Moore machine. Figure 3-57 shows the finite state machine for the modulator. The finite state machine contains six states: S_RESET, S_INIT, S_PREFRAME, S_FRAME, S_TRACKING, and S_DATA. The states are described in the following subsections.

S_RESET

The modulator begins in this state on boot-up. In S_RESET, the modulator resets signals to some default value and transitions to S_INIT.

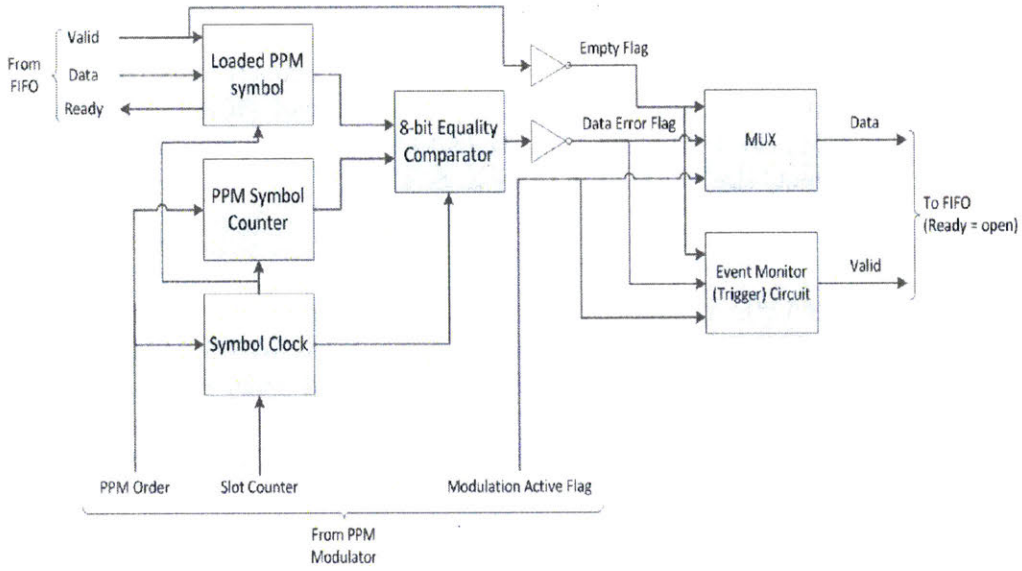

Figure 3-56: Block Diagram of Modulator Test Circuit

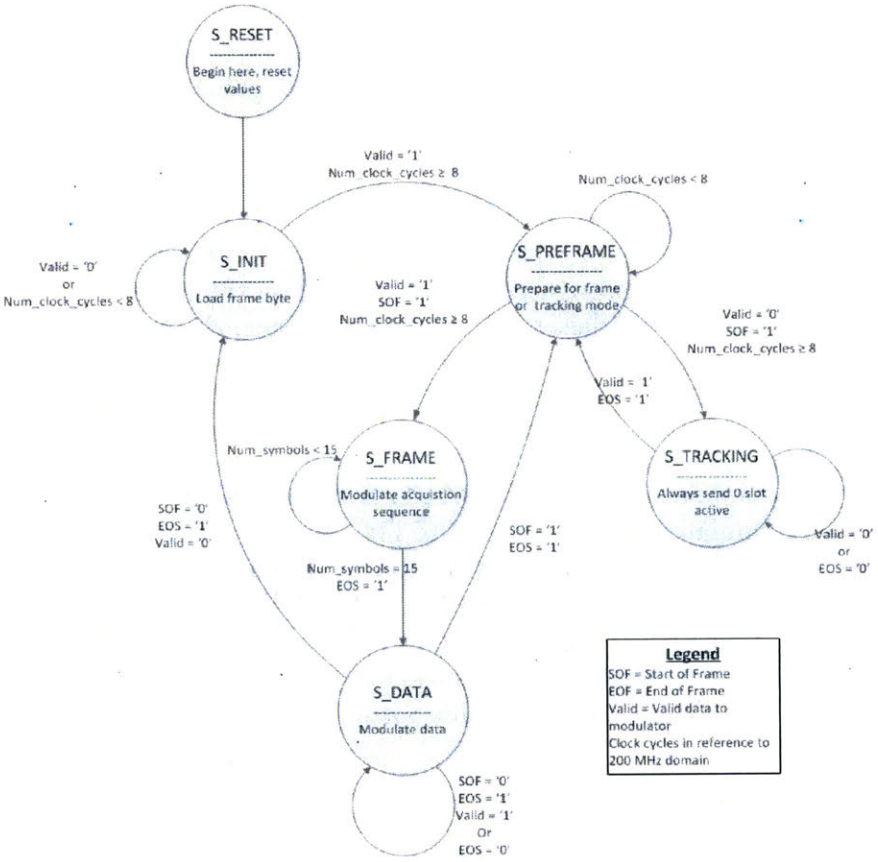

Figure 3-57: Finite State Machine for Modulator Sitting in 200 MHz Domain

S_INIT

Once in S_INIT, the modulator awaits valid data before transitioning to S_PREFRAME. If the valid flag is asserted after 8 clock cycles, the modulator accepts a new data byte from the FIFO and transitions to S_PREFRAME. If both of these conditions are not met, the modulator stays in S_INIT.

S_PREFRAME

While in S_PREFRAME, the modulator sets the PPM order and decides whether to transition to S_INIT, S_FRAME, or S_TRACKING. If the data byte loaded while in S_INIT does not have the start of frame (SOF) bit asserted, the modulator returns to S_INIT. If the SOF bit is asserted and valid is asserted, the modulator transitions to S_FRAME and reads a new data byte, but if SOF is asserted and valid is not asserted, the modulator transitions to S_TRACKING.

S_TRACKING

In S_TRACKING mode, the modulator continually sends one slot active to aid in spacial acquisition of the signal on the APD. While in S_TRACKING, the modulator always sends the 0th slot active for the PPM order loaded in S_PREFRAME. If the modulator reaches the end of a symbol and valid is asserted, it loads a new data byte and transitions into S_PREFRAME, else it stays in S_TRACKING.

S_FRAME

The modulator sends the acquisition sequence in S_FRAME. Three acquisition sequences, one for 4-PPM, one for 8-PPM, and one for 16-128-PPM, are loaded in RAM in the FPGA. One of these sequences is chosen based on the PPM order in the modulator. After modulating the last symbol in the sequence, the modulator transitions to S_DATA.

S_DATA

The transmitter modulates data in S_DATA. If valid is asserted while in S_DATA, the modulator loads a data byte for the next symbol, but if valid is not asserted, the modulator goes into S_INIT after modulating the current data byte. If the data byte loaded for the next symbol has the SOF bit asserted, the modulator transitions to S_PREFRAME after modulating the current data byte. If neither of these conditions are met, the modulator stays in S_DATA to either finish modulating the current symbol or to modulate another symbol.

3.10.3 FPGALink Interface

The interface between the CPU and FPGA is provided by an opensource project called FPGALink [21]. FPGALink provides a hardware abstraction layer for programming and communicating with the FPGA. FPGALink consists of an application

program interface (API) written in Python and FPGA cores that are written in VHSIC Hardware Description Language (VHDL). Up to 127 virtual channels, each 8 bits wide, may be used for reading and writing to the FPGA, and as shown in Figure 3-55, FIFOs are used to store the data for reading and writing. The main reason for using the FIFOs is to cross between the 48 MHz and 200 MHz clock domains. The FPGA core for FPGALink is described in the files "comm_fpga.vhdl", "harness.vhdl", and "top_level.vhdl". The most relevant API functions are flWriteChannel and flReadChannel. More information about FPGALink may be found in the user manual [21].

3.10.4 Validation and Testing

The validation and testing for the FPGA design is composed of four steps: simulations, internal testing of the synthesized FPGA image, testing of the synthesized FPGA image with an oscilloscope, and testing the FPGA image with a signal acquisition machine.

Simulations

Before attempting to synthesize the VHDL, the design was simulated at various steps to ensure functional soundness. After making changes to one component, that component was subjected to an individual simulation to critique the functionality of that component. Once that component passed simulations, the entire design was subjected to a simulation verifying that the component interfaced correctly with the entire design. The simulations tested all of the states of the finite state machine in the modulator, all transitions between states in the modulator, and various failure modes the design could experience.

Internal Testing of FPGA

Internal testing of the FPGA was performed with the modulator test circuit, as described in Section 3.10.1. This test helped identify and correct errors in the bit-to-symbol mapping function, modulator, and utilization of the FPGALink interface. This test was important for the successful demonstration of the transmitter because it caught obscure errors in the modulator and bit-to-symbol mapping functions. Additionally, this test informed the choice of packet sizes and the frequency at which to send packets to the FPGA to prevent emptying of the FIFOs.

Testing with Oscilloscope

The testing with the oscilloscope confirmed that the PPM waveform leaving the modulator could be sampled and demodulated, with the help of a symbol clock, to recover part[5] of the transmitted frame. A Python script, shown in appendix A was created to control the oscilloscope, receive sampled data from the oscilloscope, and perform

[5]Only part of the transmitted frame could be sampled due to the limited memory depth in the oscilloscope.

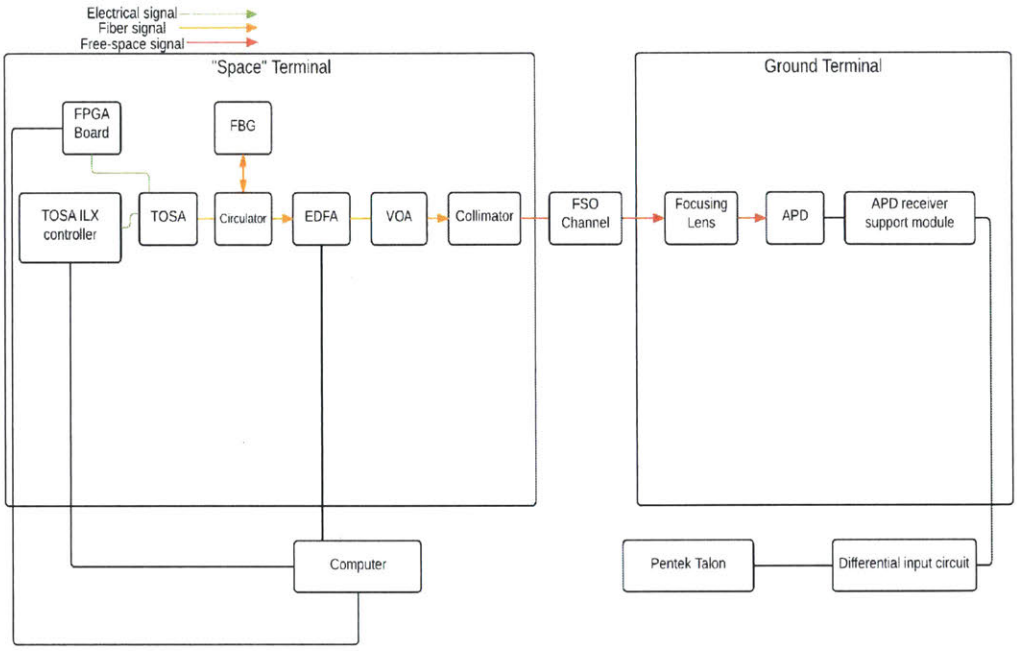

Figure 3-58: Block Diagram for the Over-The-Air Test, provided courtesy of graduate student Emily Clements

demodulation. The sampled data was demodulated with the help of a symbol clock transmitted from the FPGA. Since the sampled signal was a digital signal, no thresholding approach was employed for demodulation. The demodulated symbol values were compared to the software modulated files to determine the correctness of the modulation and demodulation. This test was made easier by transmitting a periodic waveform, such as a counter.

Testing with Signal Acquisition Machine

The Pentek Talon, as described in Section 2.2.2, was used as the signal acquisition machine in the Over-the-Air (OTA) test, the final test for the modulator. The block diagram in Figure 3-58 shows the components used in the OTA test and how they were connected during the test. The TOSA ILX Controller, more specifically the ILX Lightwave Laser Diode Controller, maintains the laser diode temperature and bias current set points. The circulator allows the reflected passband of the FBG to enter the EDFA, and the variable optical attenuator (VOA) applies attenuation to the optical signal. The VOA is not used to model channel effects, but rather, it is used to test performance at different power levels. The collimator converts the optical signal in the fiber to a FSO signal. The focusing lens focuses the laser onto the 200 µm active diameter of the APD. The APD receiver support module sets the bias point for the APD and reads out the signal from the APD. The differential input circuit converts the differential output from the APD receiver support module into a single-ended output and provides amplification. Finally, the Pentek samples the

signal on the single-ended output.

During the OTA test, the Pentek sampled the signal at 1 GS/s using 8 bits per sample. The OTA test was performed across three trials, with each trial using a different attenuation. The average optical powers at the output of the collimator were -21 dBm, -36 dBm, and -41 dBm, for trials one, two, and three, respectively. The Voxtel RIP1-NJAF APD was used for all three trials. The laser spot size on the APD may have exceeded the 200 µm active diameter, resulting in unknown losses[6] across the FSO channel. All PPM orders were used for each test. Figures 3-59 to 3-64 show the sampled waveform from trial one for PPM orders 4 to 128 with an optical transmit power of -21 dBm.

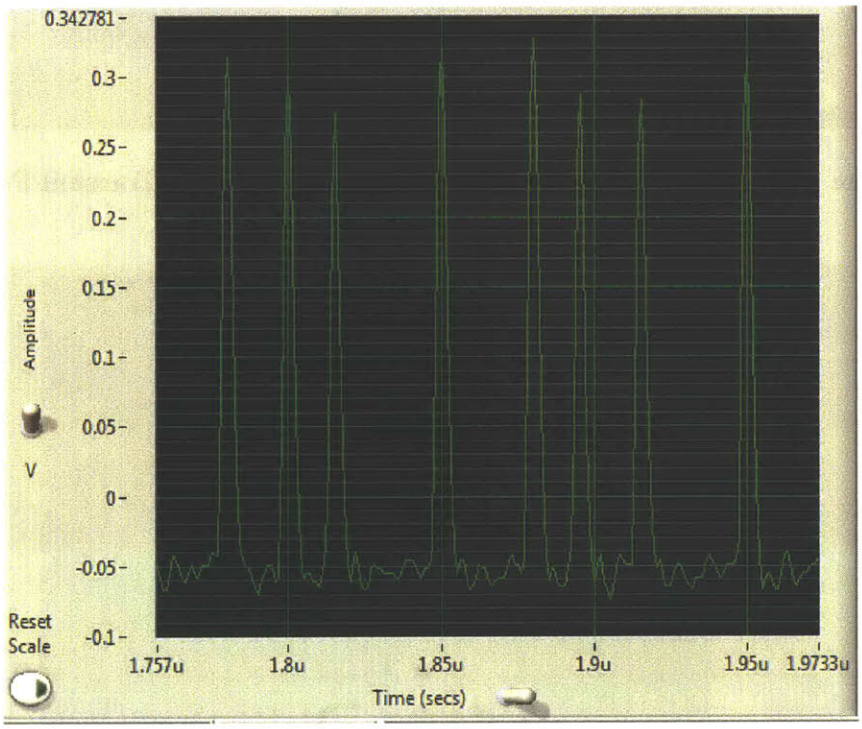

Figure 3-59: Sampled Waveform for 4-PPM with -21 dBm Transmit Power

[6]The unknown losses are due to the simple approach taken in aligning the collimator, focusing lens, and APD. Future tests with better alignment procedures and more advanced focusing/pointing mechanisms are required to validate the link budget in Table 2.1.

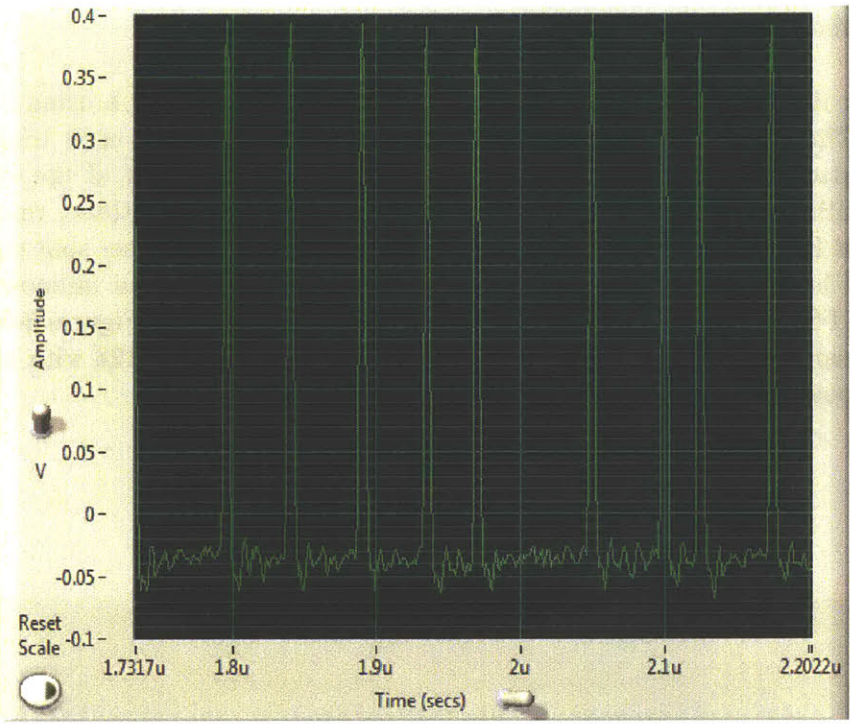

Figure 3-60: Sampled Waveform for 8-PPM with -21 dBm Transmit Power

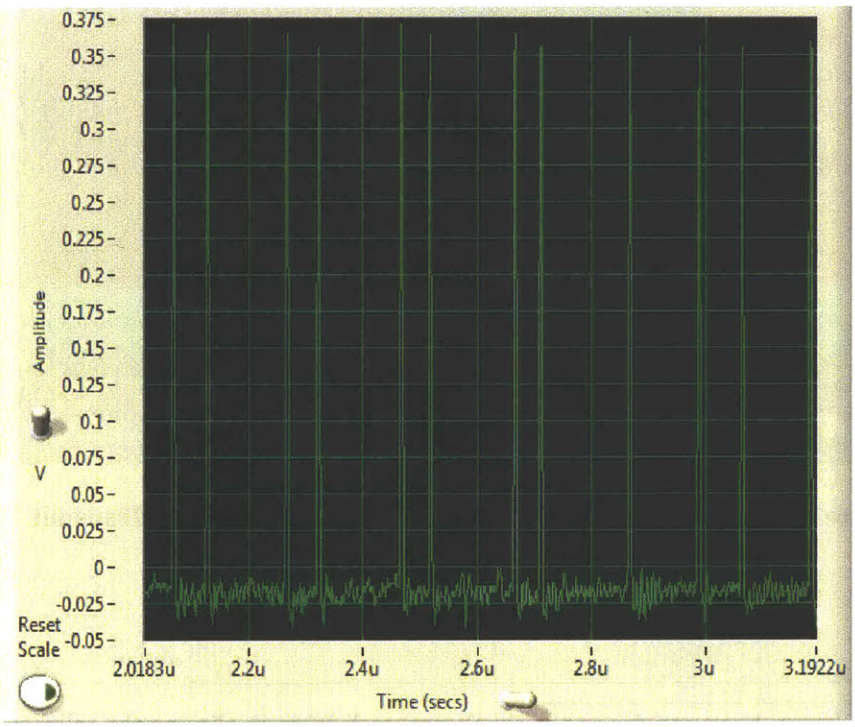

Figure 3-61: Sampled Waveform for 16-PPM with -21 dBm Transmit Power

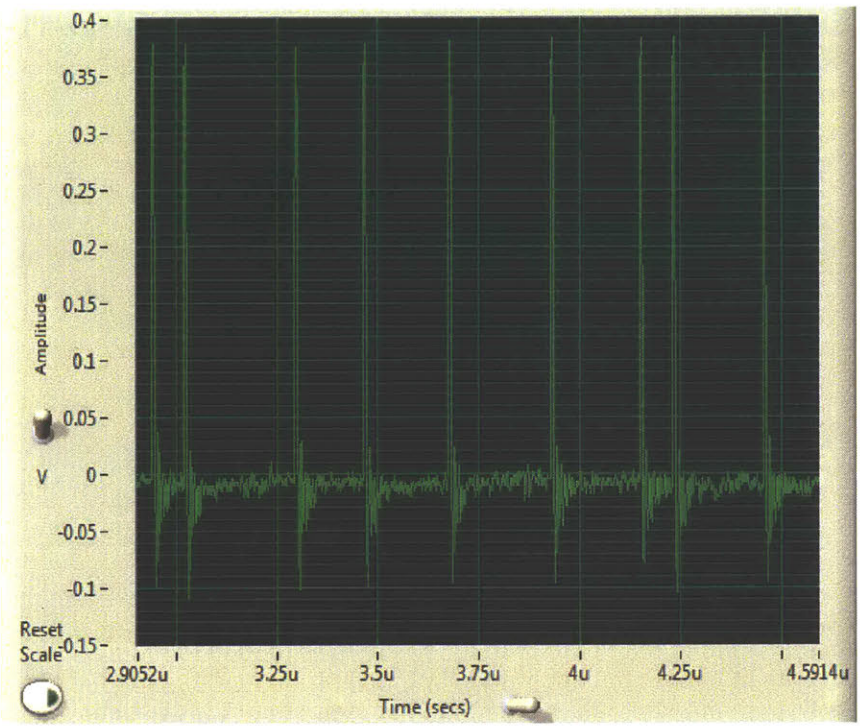

Figure 3-62: Sampled Waveform for 32-PPM with -21 dBm Transmit Power

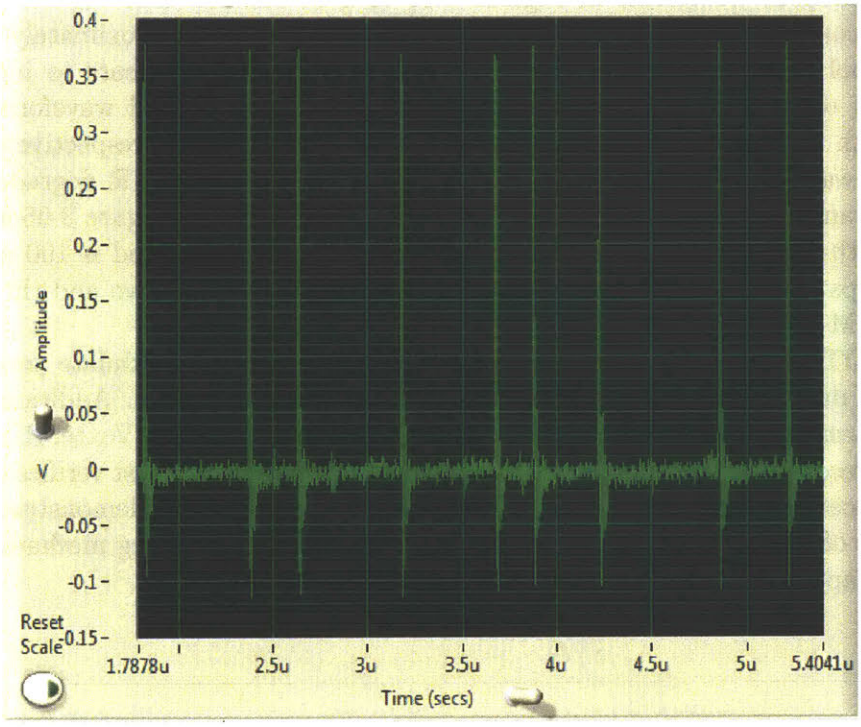

Figure 3-63: Sampled Waveform for 64-PPM with -21 dBm Transmit Power

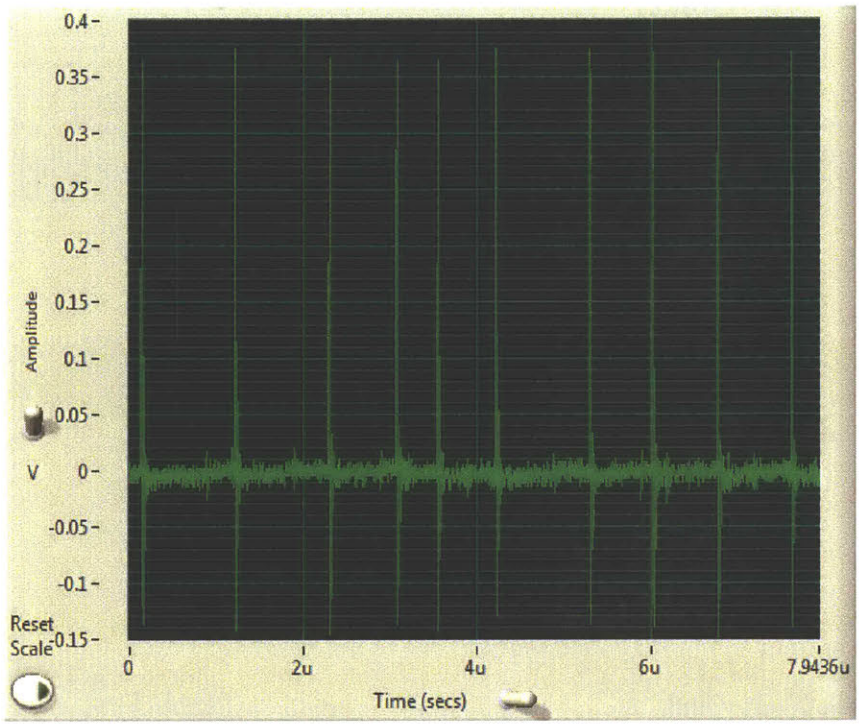

Figure 3-64: Sampled Waveform for 128-PPM with -21 dBm Transmit Power

The sampled waveforms in Figures 3-59 to 3-64 confirm that the modulator is working correctly since the average time between pulses is approximately equal to the symbol time, when including the ISGT, and the SNR appears to increase as the PPM order increases. Figures 3-65 and 3-66 show sampled waveforms of 16-PPM with optical transmit powers of -36 dBm and -41 dBm, respectively. The sampled waveforms in Figures 3-65 and 3-66 show that the SNR degrades as the optical transmit power is decreased. While the PPM pulses in Figure 3-66 are tough to see with the naked eye, the knowledge that each symbol period is 100 ns for 16-PPM helps identify the pulses. Sampled waveforms from trials two and three for all other PPM orders are omitted for brevity.

The OTA test confirms that the modulator can correctly modulate encoded, interleaved data with PPM orders from 4 to 128 while using ISGTs. Additionally, this test confirms that the Pentek Talon can sample the output of the Voxtel RIP1-NJAF APD to recover the transmitted PPM waveform. Moreover, this test verifies that FSO communications are possible with NODE, but future tests must demonstrate the capabilities of NODE by using better alignment procedures, pointing mechanisms, and focusing apparatuses.

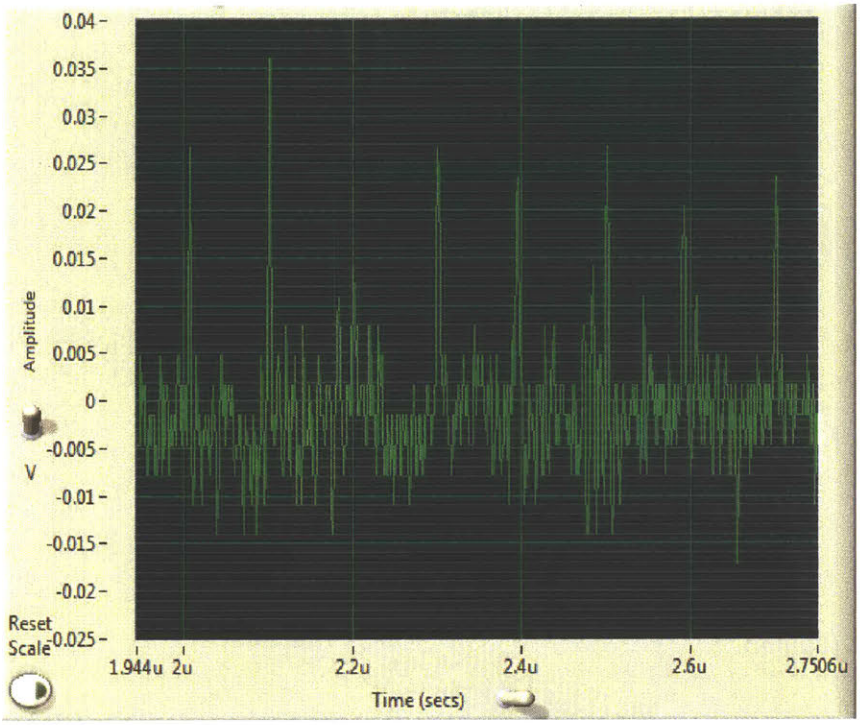

Figure 3-65: Sampled Waveform for 16-PPM with -36 dBm Transmit Power

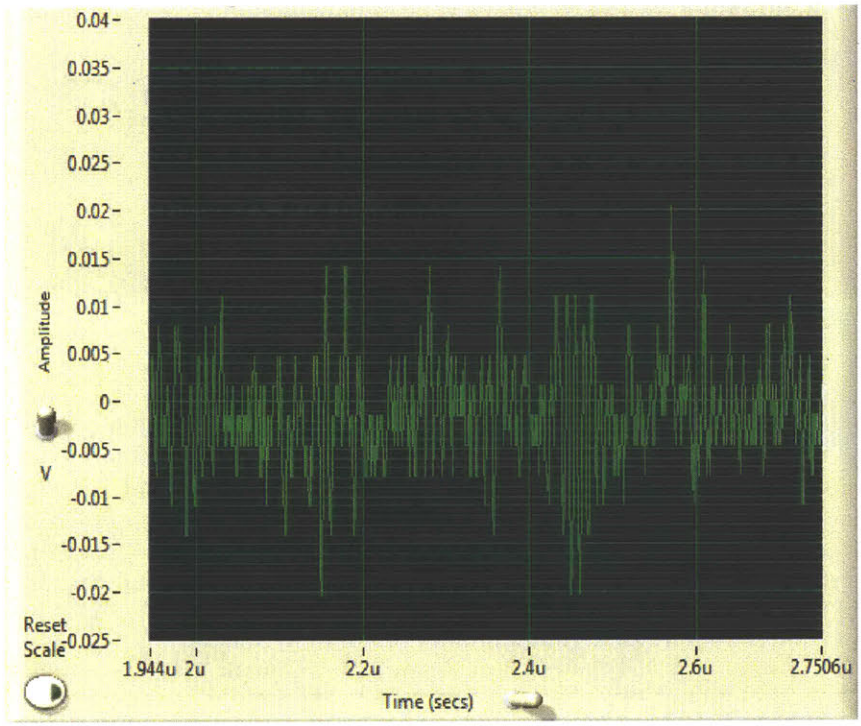

Figure 3-66: Sampled Waveform for 16-PPM with -41 dBm Transmit Power

Chapter 4

Radio Frequency Uplink

The RF uplink is outlined in Section 2.3, and the link budget is shown in Table 2.4 without interference and in Table 2.5 with interference. Table 2.5 shows that communications fail when in the presence of interference, and the interference can be mitigated with the use of a formation of CubeSats and GBBF. The operational view in Figure 2-1 shows that three CubeSats are in the formation, and each of those CubeSats acts as an element in an antenna array used for GBBF. GBBF may be used to add gain to the main lobe of radiation pattern and steer the main lobe towards the user in an attempt to gain sufficient $C/(I_o + N_o)$ to overcome the interference, and such an approach will be referred to as phase-coherent combining. Additionally,GBBF may be used to steer a null towards the jammer in an attempt to reduce the jammer's received power, thereby mitigating the interference and increasing the $C/(I_o + N_o)$. This section will show that phase-coherent combining is not sufficient to mitigate the interference from the jammer, meaning a null must be steered towards the jammer to overcome the jammer. However, the large delays and Doppler experienced in LEO make null-steering very difficult, and the following analysis explains how null-steering is possible in LEO.

4.1 Ground-Based Beamforming

As mentioned in Chapter 2, GBBF refers to processing the sampled RF data at the ground station after the data has been relayed over the lasercom downlink. The ground-based beamformer may perform phase-coherent combining or null-forming to overcome interference[1]. The ground-based beamformer uses a space-time-frequency adaptive processor (STFAP), as discussed in [39, 11], to correct for the Doppler and delay differences seen on the signal at each element of the array. The STFAP uses Doppler and delay tap lines and an adaptive processor in order to perform phase-coherent combining and null-forming. This section will give a high-level overview of the STFAP, while discussing some of the trade space in designing the STFAP, and will show the advantages and disadvantages of both phase-coherent combining and

[1]While this thesis examines phase-coherent combining and null-forming as two distinct operations, some combination of the two may be employed if the adaptive processor finds it to be optimal.

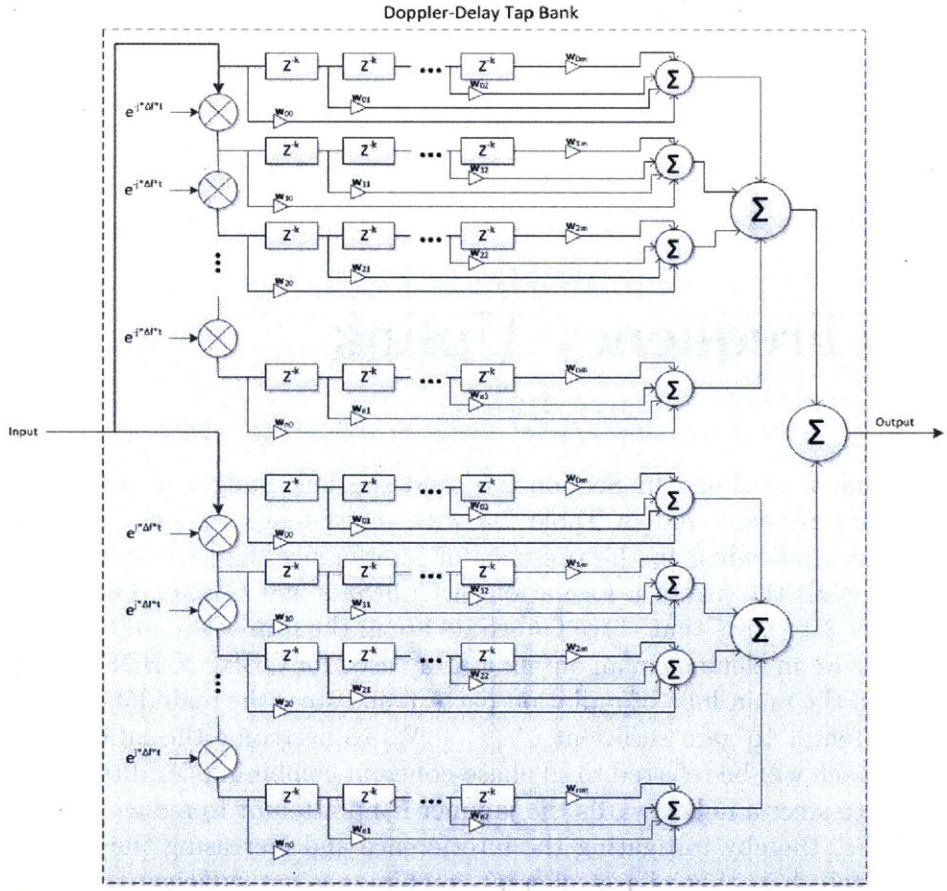

Figure 4-1: Doppler-Delay Tap Bank used for GBBF

null-forming within this architecture. Further discussion on the STFAP approach may be found in [39, 11].

4.1.1 Space-Time-Frequency Adaptive Processing

Because of the large velocities of the CubeSats and the relatively large distances between them, the delay and Doppler between each signal must be resolved in order to combine the signals at each element of the array. The transmitted waveform will contain training data in order to inform the choice of weights, and the delay and Doppler will be resolved through the use of delay and Doppler taps [11]. The delay and Doppler taps must cover the expected delay and Doppler spread[2] between signals at each element of the array. The delay and Doppler taps are arrayed in a Doppler-Delay Tap bank, as shown in Figure 4-1.

The Doppler tap lines are shown on the left of Figure 4-1 while the delay tap lines

[2]The terms Doppler spread and delay spread refer to the maximum difference in the Doppler and delay for the user signal between any two elements in the array, and a channel with Doppler and delay spread is known as a doubly dispersive channel [11].

are shown to the right. There are m Doppler taps per line whereas there are n delay taps per line. Each Doppler-delay tap combination has a complex weight associated with it, and there are a total of $2*n*m$ complex weights per Doppler-Delay tap bank. Since the difference in the Doppler between two elements in the array can be either positive or negative, the Doppler taps must cover the Doppler spread in positive and negative frequencies. The amount of frequency shift provided by each Doppler tap is related to the rate at which the complex weights are updated. The rate at which the complex weights are updated sets the frequency resolution needed to properly model the Doppler between bursts of training data, and the term Δf shown in Figure 4-1 is the frequency resolution. Put another way, the greater the rate at which the weights are updated, the coarser the frequency resolution can be in order to correct the Doppler between training bursts with adjustments to the complex weights. The number of Doppler taps needed per line is shown in the following equation.

$$DopplerTaps = \frac{DopplerSpread}{\Delta f} \tag{4.1}$$

As the update rate increases, the frequency resolution becomes more coarse, requiring the use of fewer Doppler taps. Unfortunately, the update rate of the weights is limited by the amount of training data sent and the time between bursts of training data. Enough training data must be sent to properly determine the weights and the training data must be sent at regular intervals between the data, and both of these factors limit the rate at which the weights may be updated. The number of delay taps needed per line is shown in the following equation.

$$DelayTaps = \frac{DelaySpread}{\Delta t} \tag{4.2}$$

In Equation 4.2, Δt represents the Nyquist sampling interval. The number of required delay taps may be decreased by decreasing the subchannel bandwidth, thereby increasing the Nyquist sampling interval. In Figure 4-1, each delay tap has a delay value corresponding to Δt. The channel bandwidth may be broken up into smaller subchannels with a polyphase filter bank, as shown in Figure 4-3. The subchannel bandwidth is constrained by the Doppler frequency; the subchannel bandwidth must be greater than the absolute value of the maximum and minimum expected Doppler frequency for a pass. The Doppler-Delay Tap Banks operate on one channel received from each element of the array, as shown in Figure 4-2.

The adaptive processor is contained within the channel processor, and the adaptive processor determines the weight values for all of the Doppler-Delay Tap Banks within the channel processor. The adaptive processor will be described more in Section 4.1.1. Figure 4-3 shows how the channel processor fits into the overall STFAP architecture.

The polyphase filter bank[3] is an efficient way to channelize the input signal into smaller subchannels [38]. The subchannels are inputs to the channel processors, where the Doppler-Delay Tap Banks resolve the Doppler and delay so that the adaptive

[3]A description of the polyphase filter bank is given in [38].

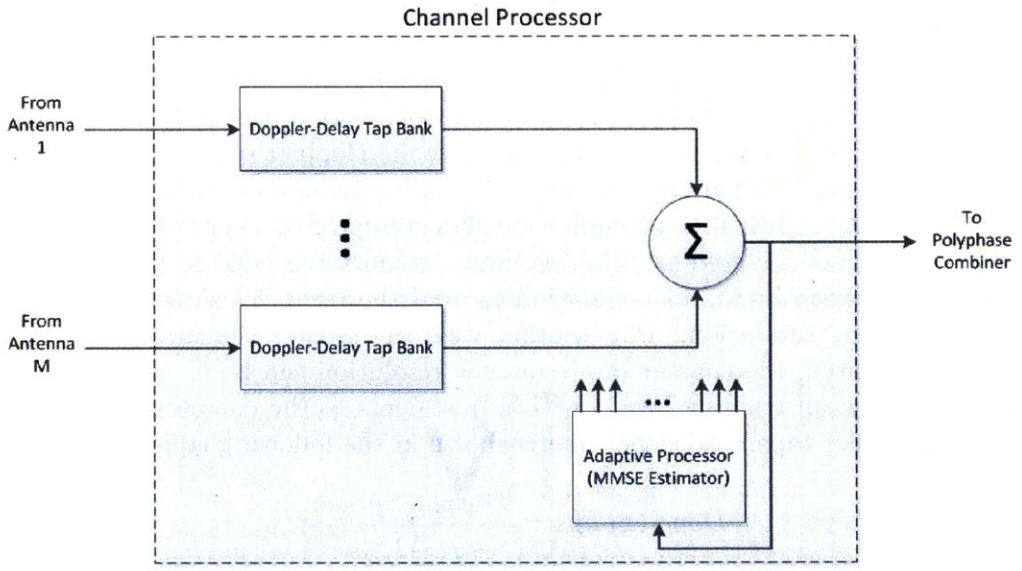

Figure 4-2: Channel Processor for GBBF

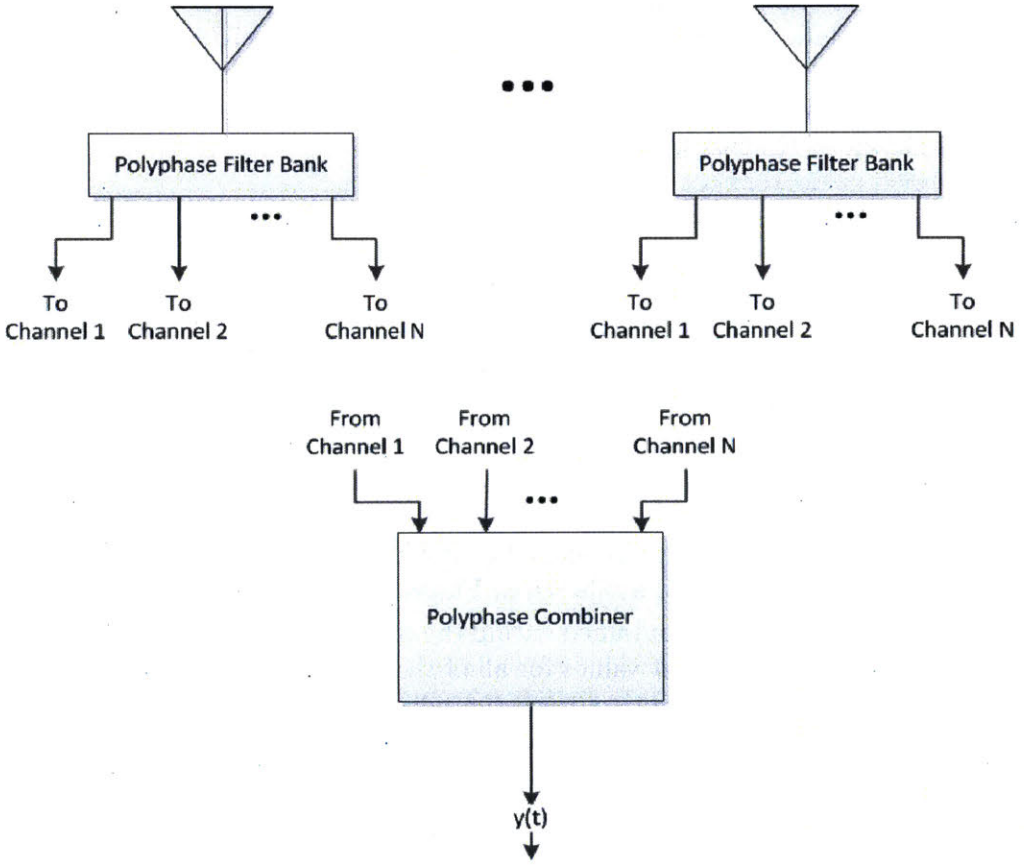

Figure 4-3: Space-Time-Frequency Processor for GBBF

processor may set the optimal complex weights based off of the training data. The combination of resolving the delay and Doppler and setting the optimal complex weights results in phase-coherent combining or null-forming, depending on which approach the adaptive processor finds optimal. The output of the channel processor is an input to the polyphase combiner, which acts as the inverse of the polyphase filter bank and combines the subchannels to generate the output signal, $y(t)$.

Adaptive Processor

The adaptive processor attempts to find the complex weights that minimize the mean-square error between the known training data and the output of the channel processor in the MMSE approach [54, 11]. Following this approach, the result is that the optimum weights, $\omega_{\mathbf{opt}}$, are given by the Wiener solution [54, 11], shown below.

$$\omega_{\mathbf{opt}} = \mathbf{R^{-1}r} \tag{4.3}$$

In Equation 4.3, R is the covariance matrix and r is the expectation of the training data multiplied by the received signal. For the STFAP, $\omega_{\mathbf{opt}}$, $\mathbf{R}$, and $\mathbf{r}$ quickly grow in dimensionality[4] as the number of Doppler taps, delay taps, and receivers increases [11]. As the covariance matrix grows in size, more samples (training data for the STFAP) are needed to estimate the covariance matrix with sufficient fidelity [68]. Increasing the dimensionality of the covariance matrix and the amount of training data needed to approximate the covariance matrix can make the STFAP infeasible [39, 11]. To make the beamformer feasible, the STFAP must be kept sparse by minimizing the amount of Doppler and delay taps in Figure 4-1 [11, 39]. Subchannelizing the received signal through use of the polyphase filter bank and maximizing the rate at which the weights are updated aim to make the STFAP sparse. A trade space exists between the Doppler-Delay Tap Banks, the subchannel bandwidth, and amount of training data used to estimate the covariance matrix, but unfortunately, this is beyond the scope of this thesis.

4.1.2 Phase-Coherent Combining

Phase-coherent combining may be used to steer the main lobe of the array's radiation pattern toward the user and place additional gain on the main lobe. Since training data is assumed to be contained within the user's signal, the signals at each element of the array can be aligned in time and frequency, which results in added gain for the user's signal after the signals from each element of the array are combined. The expected improvement in SNR due to phase-coherent combining is given by [31]:

$$E[snr] = M * SNR \tag{4.4}$$

where M is the number of elements in the array. Effectively, the observed transmitter EIRP is increased by $10\log_{10}(M)$ dB when phase-coherent combining is used.

[4]Refer to Chapter 10 in [11] for a specifics on evaluating the STFAP.

Table 4.1: Link Budget for RF Uplink assuming two satellites and one user while in the presence of interference.

Parameter	Best Case	Worst Case	Units	Notes
Distance	600	2000	km	
Frequency	500	500	MHz	
Bandwidth	100	100	kHz	
Modulation/Coding	QPSK/DVB-S2	QPSK/DVB-S2		
Data Rate	100	100	kbps	
Transmitter				
Transmit Gain	3	3	dBi	Hemispherical antenna pattern
Transmit Power	7	7	dBW	
Transmit Losses	-3	-3	dB	50% Antenna efficiency
Beamforming Gain	3	3	dB	
EIRP	10	10	dB	
Jammer				
Jammer Gain	3	3	dBi	
Jammer Power	20	20	dBW	
Jammer Losses	-3	-3	dB	50% Antenna efficiency
EIRP	20	20	dB	
Channel				
Free Space Path Loss	-141.99	-152.45	dB	
Atmospheric Losses	-0.02	-0.2	dB	From STK Simulation
Polarization Losses	-3	-3	dB	RHCP to LP
Receiver				
Receiver Gain	3	3	dB	Monopole on poor ground plane
Receiver Losses	-3	-3	dB	50% Antenna Efficiency
Noise Factor (NF)	2	2	dB	
System Temperature (T_{sys})	28.75	28.75	dBK	
G/T	-28.75	-28.75	dB/K	
Link Information				
$C/(N_o + I_o)$	39.99	39.84	dB-Hz	
$E_b/(N_o + I_o)$	-10.01	-10.16	dB	
Required $E_b/(N_o + I_o)$	1	1	dB	Quasi Error Free Packet Error Rate=1×10^{-7} [29]
Link Margin	-11.01	-11.16	dB	

Adding additional gain to the main lobe of the array's radiation pattern can be achieved if the user and jammer signals are uncorrelated at each CubeSat receiver, meaning the jammer will not see any additional gain due to phase-coherent combining. When the adaptive processor finds phase-coherent combining to be optimal, the improvement in SNR shown in Equation 4.4 extends directly to an improvement in $C/(I_o+N_o)$ or SINR. The link budgets in Tables 4.1 and 4.2 show improvements in the $C/(I_o + N_o)$ when phase-coherent combining is used for an antenna array composed of 2 Cubesats and 3 CubeSats, respectively.

Tables 4.1 and 4.2 show that phase-coherent combining may not be sufficient to overcome interference. Additionally, using a lower rate code, such as the 1/4 rate DVB-S2 code, with phase-coherent combining is not sufficient as the required E_b/N_o for the 1/4 rate DVB-S2 code is 0.75 dB [29]. Therefore, effort must be made to ensure the adaptive processor finds null-forming more optimal, when compared to phase-coherent combining, in order to overcome the interference.

4.1.3 Null-Forming

Null-forming involves placing a null at the jammer location in order to make the jammer signals received at the different elements of the array destructively interfere with each other. Unfortunately, null-forming is more constraining than phase-coherent combining as the total received signal at each element of the array, consisting of both the user and jammer signals, must be correlated with the total received signal

Table 4.2: Link Budget for RF Uplink assuming three satellites and one user while in the presence of interference.

Parameter	Best Case	Worst Case	Units	Notes
Distance	600	2000	km	
Frequency	500	500	MHz	
Bandwidth	100	100	kHz	
Modulation/Coding	QPSK/DVB-S2	QPSK/DVB-S2		
Data Rate	100	100	kbps	
Transmitter				
Transmit Gain	3	3	dBi	Hemispherical antenna pattern
Transmit Power	7	7	dBW	
Transmit Losses	-3	-3	dB	50% Antenna efficiency
Beamforming Gain	4.5	4.5	dB	
EIRP	11.5	11.5	dB	
Jammer				
Jammer Gain	3	3	dBi	
Jammer Power	20	20	dBW	
Jammer Losses	-3	-3	dB	50% Antenna efficiency
EIRP	20	20	dB	
Channel				
Free Space Path Loss	-141.99	-152.45	dB	
Atmospheric Losses	-0.02	-0.2	dB	From STK Simulation
Polarization Losses	-3	-3	dB	RHCP to LP
Receiver				
Receiver Gain	3	3	dB	Monopole on poor ground plane
Receiver Losses	-3	-3	dB	50% Antenna Efficiency
Noise Factor (NF)	2	2	dB	
System Temperature (T_{sys})	28.75	28.75	dBK	
G/T	-28.75	-28.75	dB/K	
Link Information				
$C/(N_o + I_o)$	41.49	41.34	dB-Hz	
$E_b/(N_o + I_o)$	-8.51	-8.66	dB	
Required $E_b/(N_o + I_o)$	1	1	dB	Quasi Error Free Packet Error Rate=1×10^{-7} [29]
Link Margin	-9.51	-9.66	dB	

from other elements of the array for null-forming to be effective. As the signals become uncorrelated, the performance of null-forming degrades rapidly. Since the adaptive processor ultimately chooses which approach, either phase-coherent combining or null-forming, is more optimal, efforts must be made to increase the correlation between signals at different elements so that the null-forming approach is more optimal, thereby suppressing the interference and allowing the link to close.

Differential-Delay Differential-Doppler Product

The differential-delay differential-Doppler (D^4) product [39] is representative of the correlation between channels, where a smaller D^4 product corresponds to a correlation coefficient closer to 1 and a larger D^4 product corresponds to a correlation coefficient closer to 0. The differential-delay is the difference between the time delay for the user and jammer signals at each element of the array. Likewise, the differential-Doppler is the difference between the Doppler frequency for the user and jammer signals at each element of the array. The D^4 product is the product of the differential-delay and differential-Doppler, and the largest D^4 product is the limiting case. In [39], the authors describe how performance in the presence of jamming for a certain class of GBBF algorithms, more specifically the commutative algorithms[5], degrades as the D^4

[5]A description of the difference between the commutative method and non-commutative methods are given in [39, 11]. The STFAP algorithm described here is a commutative algorithm. Noncommutative algorithms employ more complicated techniques and are beyond the scope of this thesis.

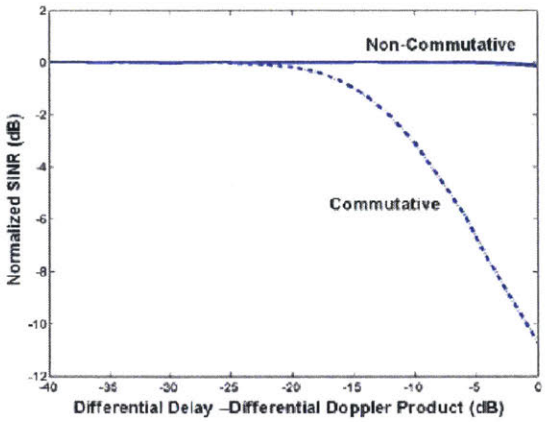

Figure 4-4: Plot of SINR vs Differential-Delay Differential-Doppler Product [39]

product increases. This degradation is caused by the increasing difficulty in forming nulls towards the interferers. Figure 4-4 from [39] shows the relationship between the SINR of the intended user and the D^4 product after beamforming.

The negative effects of the large D^4 products are mitigated slightly by the STFAP shown earlier. The signals may appear uncorrelated if the bandwidth is large with respect to the differential delay [39]. However, channelizing into smaller bandwidths gives an improvement in null depth because the jammer's differential-delay is less than the delay given for a delay tap, i.e., the delay may be viewed as a phase shift and corrected by the complex weights. As the update rate increases, the Doppler resolution is made more coarse, so the jammer's differential-Doppler may be corrected by the complex weights. When the differential-Doppler and differential-delay may be easily corrected by the complex weights after aligning the user signal, such as when there is a small D^4 product, the null depth increases as the jammer's interference may be destructively combined more effectively. For the purposes of this thesis, D^4 products below 1×10^{-3} throughout the duration of a pass will be considered sufficient for the beamformer to produce a "deep-enough" null to mitigate the interference. While the D^4 product is used as the sole metric to ascertain nulling performance, the D^4 product is only a predictor of nulling performance; put another way, the interplay between the differential-Doppler, differential-delay, and null depth is fairly complex and is not perfectly captured by the D^4 product.

4.2 Approach

Systems Tool Kit (STK) is used to analyze D^4 products for various configurations of the CubeSat formation/cluster, user, and jammer. The CubeSat formation is given various geometries in order to determine which geometry, if any, minimize the D^4 product. Additionally, different orbital parameters, such as the orbit altitude, inclination and right ascension of the ascending node (RAAN), are examined to see if they would have any impact on the D^4 product. The spacing between the user and jam-

Table 4.3: Orbital Parameters

Parameter	Value
Semi-Major Axis	600 km
Eccentricity	0
Inclination	90°
Argument of Perigee	90°
RAAN	242.5°
True Anomaly	45°

Table 4.4: User-Jammer Parameters

Parameter	Value
User Latitude	41.07°
user Longitude	-89.38°
Jammer Latitude	40.99°
Jammer Longitude	-87.01°
User-Jammer Spacing	200 km

mer, the placement of the user and jammer relative to each other, and the placement of the user and jammer in relation to the CubeSat cluster are analyzed to determine their impact on D^4 products. Even though the D^4 product is only a predictor of nulling performance, the D^4 product may be analyzed to derive performance bounds for nulling, and details are intentionally omitted for public releasability. The results of the STK simulations are shown in Sections 4.4.3 and 4.5.

4.3 Parameters

Even though the orbital parameters and user-jammer placement are adjusted to determine the impact on the D^4 product, the orbital parameters and user-jammer placement are adjusted around one set of parameters that resulted in a minimal D^4 product. The set of orbital parameters is shown in Table 4.3.

The set of parameters for the user-jammer placement is shown in Table 4.4.

The CubeSats in the cluster were given spacings (also known as inter-satellite spacing) of 0.88 km and 8.8 km in the simulations minimizing the D^4 product. The following sections will discuss the rationale for selecting the parameters and highlight the trade-offs between various combinations of parameters.

4.4 CubeSat Formations

As stated in Chapter 2, every CubeSat formation/custer is composed of three CubeSats. Three CubeSats were chosen to give each formation redundancy. Only two CubeSats are needed to form a null over the jammer, so the third CubeSat acts as redundancy. The CubeSats are separated by no more than 10 km for the adjacent

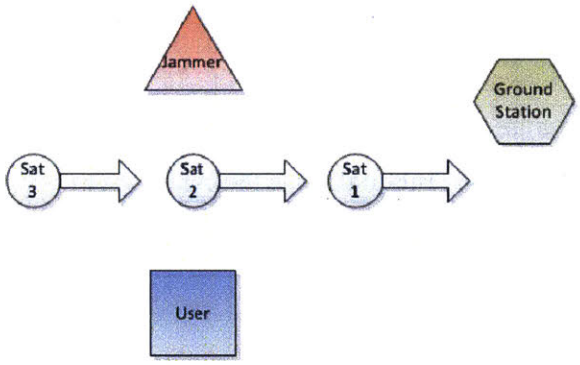

Figure 4-5: Line of Pearls Formation

CubeSat to minimize the D^4 product. A spacing larger than 10 km results in a larger D^4 product, but a spacing less than 10 km slightly, but not significantly, improves the D^4 product. When selecting a specific spacing, station-keeping and half-power beamwidth (HPBW) must be taken into consideration. Formation flying has been demonstrated for the inter-satellite spacings examined, demonstrating the feasibility of maintaining such spacing. The HPBW is determined by the spacing between CubeSats, and the half-power beamwidth (HPBW) is approximately $\frac{\lambda}{D_{total}}$, where D_{total} is the distance spanned by all three CubeSats[54]. A jammer inside the HPBW is difficult to null, but fortunately, the large spacings used here result in a HPBW of 340 µrad and 34 µrad, respectively.

The altitude for the orbit was selected to be 600 km. STK simulations show that an altitude higher than 600 km results in larger D^4 products since the differential delay increases as the altitude increases. Also, simulations reveal that an altitude lower than 600 km results in a larger differential Doppler as lower orbits have larger velocities, and thus, larger Doppler shifts. In short, an orbit altitude near 600 km is a "happy-medium" where neither the differential delay nor the differential Doppler are too large.

The orbit inclination and RAAN by themselves have no impact on the D^4 product; however, in relation to the user and jammer positions and configuration, the inclination and RAAN are significant in minimizing the D^4 product. Since the optimal RAAN and inclination rely heavily on the user-jammer placement, they will be discussed further in Section 4.5.

Analysis shows that the arrangement of the CubeSats in the formation has a significant impact on the D^4 product. Subsections 4.4.1, 4.4.2, and 4.4.3 elaborate upon two different formations to show how the formation affects the D^4 product.

4.4.1 Line of Pearls Formation

The line of pearls formation consists of the CubeSats being located in the same orbital plane but at different True Anomalies. Figure 4-5 shows the line of pearls formation. In Figure 4-5, the arrows represent the direction of spacecraft velocity.

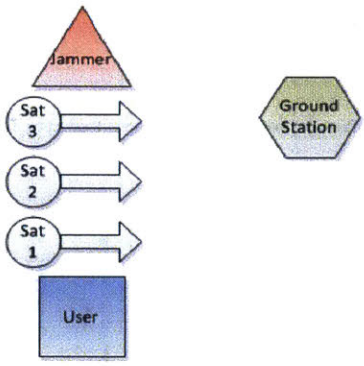

Figure 4-6: Abreast Formation

4.4.2 Abreast Formation

The abreast formation consists of the CubeSats being located on slightly different orbital planes, where the orbital planes differ only in their RAAN. Figure 4-6 shows the abreast formation. The arrows in Figure 4-6 represent the direction of spacecraft velocity.

4.4.3 Analysis

For all formations, the D^4 product tends to get significantly larger, thereby reducing the nulling depth, as the inter-satellite spacing is increased beyond 9 km. This trend is agnostic to the specific formation; all formations see decreased performance as the inter-satellite spacing is increased beyond 9 km. However, the D^4 product grows more rapidly for the line of pearls formation than the abreast formation as the inter-satellite spacing is increased beyond 9 km. Figures 4-7 and 4-8 show the D^4 product for an entire pass for the line of pearls and abreast formation, respectively, at a spacing of approximately 880 m with the parameters shown in tables 4.3 and 4.4.

When comparing plots of D^4 product for an entire pass, the curve giving the maximum D^4 product is the curve that determines the maximum attainable null-depth. Each curve shows the D^4 product between the user and jammer signals at each element of the array, or CubeSat. Figures 4-7 and 4-8 show that given a "small-enough" spacing (any spacing less than 9 km[6]), both formations perform identically with respect to the D^4 product. At an intersatellite spacing less than 9 km, all formation configurations are equal with respect to their effect on D^4 products. However, the different performance for each formation with respect to the D^4 product can be seen when the inter-satellite spacing is increased to 12 km. Figures 4-9 and 4-10 show the D^4 product for an entire pass for both formations at a spacing of 12 km with the parameters shown in tables 4.3 and 4.4.

Figures 4-9 and 4-10 show that given the same inter-satellite spacing, orbital parameters, and user-jammer parameters the abreast formation gives better D^4 products

[6]For the sake of brevity, tests showing that a spacing less than 9 km is small-enough are not included.

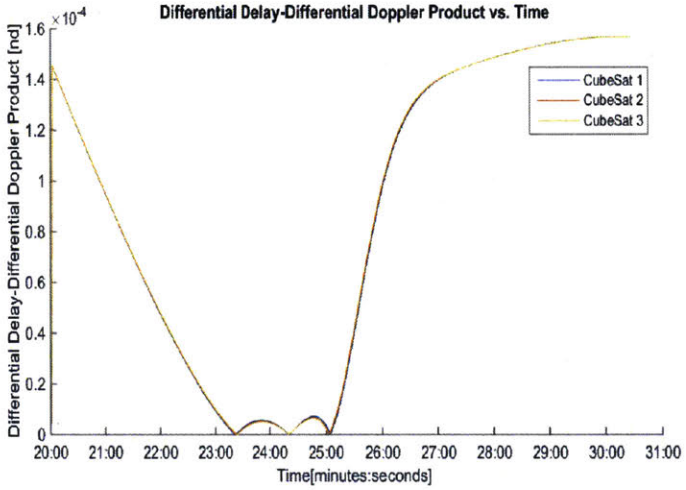

Figure 4-7: D^4 Product for an Entire Pass for the Line of Pearls Formation with an Inter-Satellite Spacing of 880 m

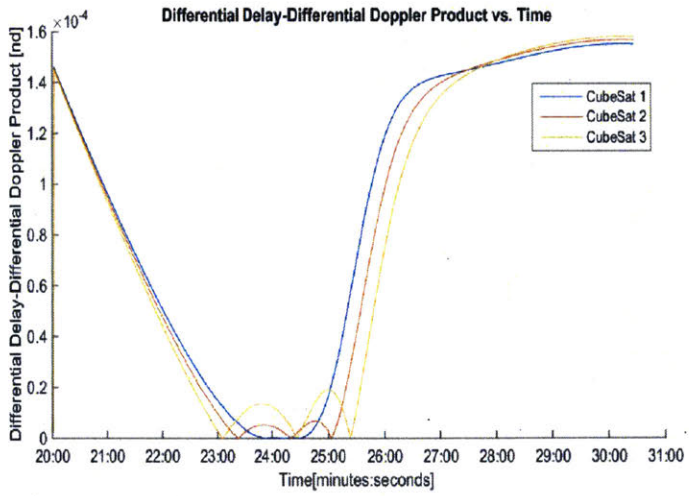

Figure 4-8: D^4 Product for an Entire Pass for the Abreast Formation with an Inter-Satellite Spacing of 880 m

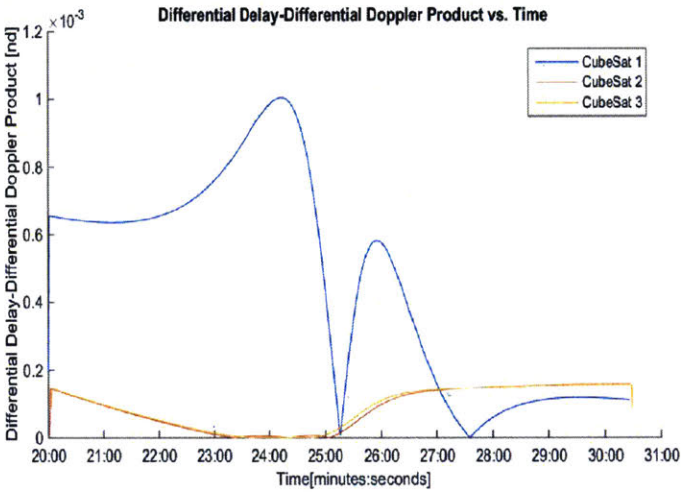

Figure 4-9: D^4 Product for an Entire Pass for the Line of Pearls Formation with an Inter-Satellite Spacing of 12 km

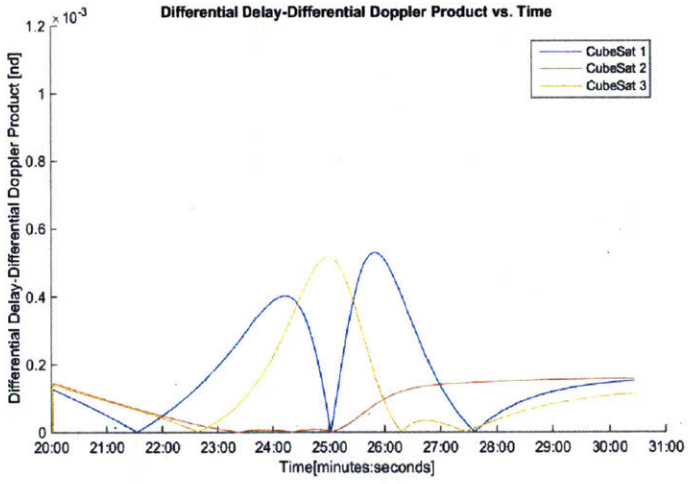

Figure 4-10: D^4 Product for an Entire Pass for the Abreast Formation with an Inter-Satellite Spacing of 12 km

than the line of pearls formation. Further analysis should define a metric describing how difficult it is to implement each formation, and this metric could include ΔV required on the rocket and payload and the difficulty in performing station-keeping. Since a larger inter-satellite spacing can give sharper nulls and relax the constraints placed on station-keeping, the superior performance of the abreast formation with larger inter-satellite spacing makes it the best choice for the proposed communications system.

4.5 User-Jammer Placement

The placement of the user and the jammer has an impact on the D^4 product. In the previous simulations, the user and jammer were given the placement that minimizes the D^4 product. The minimizing user-jammer placement is shown in Figures 4-6, 4-5, and 2-2. More specifically, the minimizing user-jammer orientation has the user and jammer lying equidistant but on opposite sides of the formation ground track, with the line connecting the user and jammer lying perpendicular to the formation's velocity vector. The values to obtain this user-jammer orientation are shown in Table 4.4. The following analysis shows that the user-jammer placement can have a significant impact on the D^4 product.

4.5.1 Analysis

In order to determine the effect of the user-jammer placement on the D^4 product, a series of simulations are performed where the user-jammer orientation is systematically adjusted to examine the effects on the D^4 product. First, the spacing between the user and jammer is increased while keeping the user and jammer equidistant from the formation's ground track. Second, the latitude of the jammer is adjusted until the D^4 product becomes unacceptable for null-forming. Third, after bringing the jammer's latitude back to the best case, the jammer's longitude is adjusted until the D^4 product becomes unacceptable for null forming. Beginning with the values shown in Tables 4.3 and 4.4, the abreast formation, and an inter-satellite spacing of 880 m, the user-jammer spacing was increased to 400 km. The resulting D^4 product for an entire pass is shown in Figure 4-11.

Figure 4-11 very nearly resembles the minimizing case shown in Figure 4-8, meaning separating the user and jammer by a larger distance while keeping them symmetric with respect to the satellite ground track has no effect on the D^4 product. Increasing the jammer's latitude results in a lack of symmetry, as shown in Figure 4-12.

In Figure 4-12, the arrow shows the increase in latitude. Keeping all parameters the same as those in Figure 4-11, with the exception of the jammer's latitude, which was increased by $0.05°$, results in the plot shown in Figure 4-13.

Figure 4-13 shows that increasing the jammer's latitude, thereby removing symmetry between the user-placement and the formation's ground track, can significantly increase the D^4 product. This leads to one of the main points concerning the user-jammer placement: the geometry and symmetry between the formation's ground

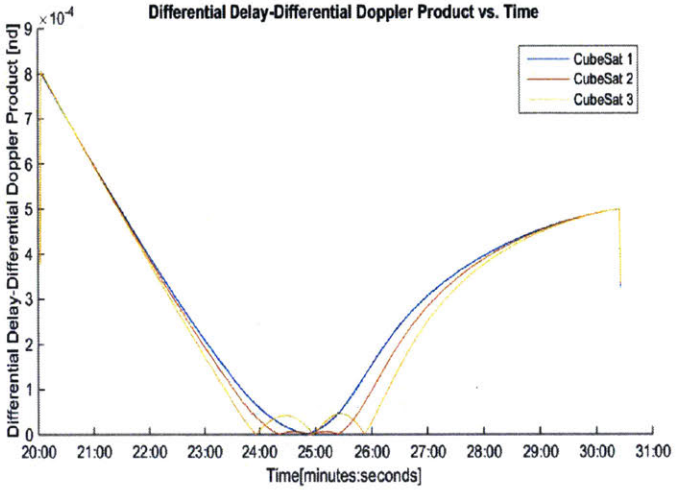

Figure 4-11: D^4 Product for an Entire Pass with a User-Jammer Spacing of 400 km and Symmetric Placement

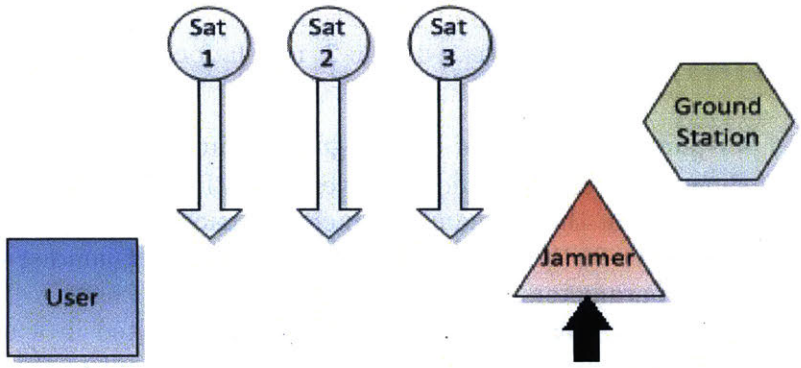

Figure 4-12: Graphic Showing Ground Placement with a User-Jammer Spacing of 400 km and Jammer Latitude increased by 0.05° in Relation to Figure 4-11

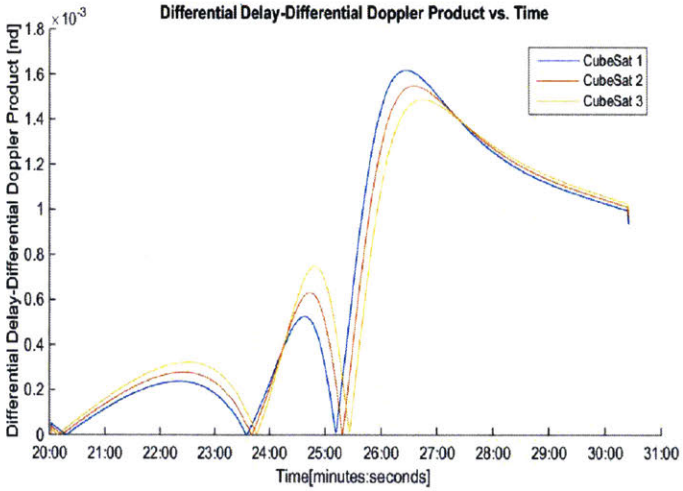

Figure 4-13: D^4 Product for an Entire Pass with a User-Jammer Spacing of 400 km and Jammer Latitude increased by 0.05° in Relation to Figure 4-11

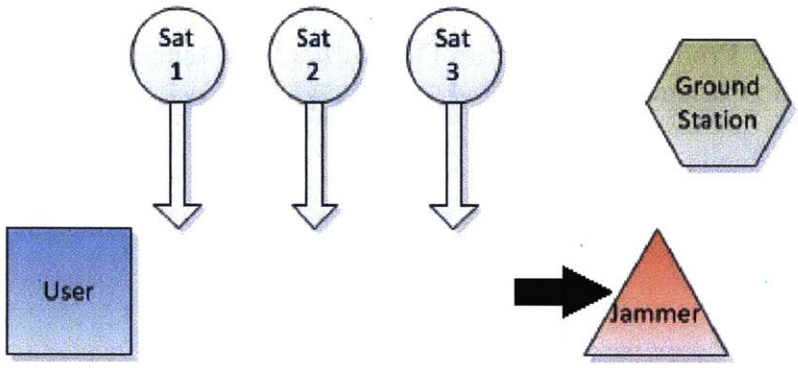

Figure 4-14: Graphic Showing Ground Placement with a User-Jammer Spacing of 426 km and Jammer Longitude increased by 0.2° in relation to Figure 4-11

track and the user-jammer orientation has a significant impact on the D^4 product.

Figure 4-13 shows that the D^4 product is very sensitive to the placement of the user and the jammer when the user and jammer are spaced out across the satellite ground track. In order to further characterize the effect of user-jammer orientation, the user and jammer parameters were reset to the parameters used in Figure 4-11, and the jammer longitude was increased by 0.2° . This resulted in a user-jammer spacing of 426 km and the user and jammer not being equidistant to the formation's ground track. The ground placement and geometry for this increased longitude are shown in Figure 4-14.

In Figure 4-14, the arrow shows the direction of increase in longitude. Figure 4-15 shows a plot of the D^4 product for a pass with the jammer longitude increased by 0.2° compared to the simulation shown in Figure 4-11.

Figure 4-15 shows that an increase in jammer longitude of 0.2° is comparable to

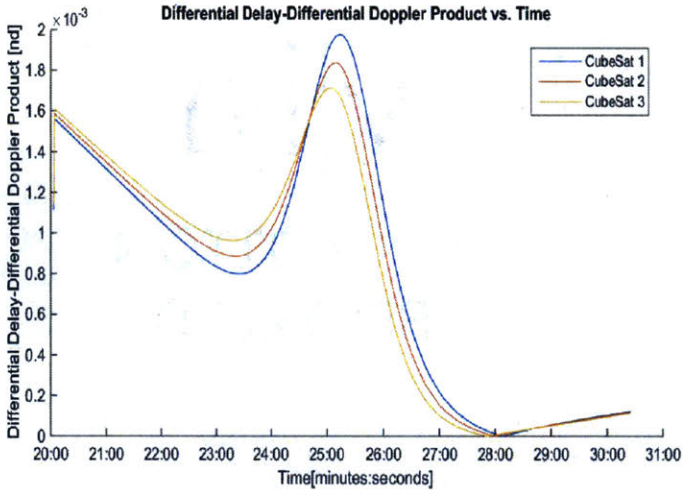

Figure 4-15: D^4 Product for an Entire Pass with a User-Jammer Spacing of 426 km and Jammer Longitude increased by 0.2° in Relation to Figure 4-11

a latitude increase of 0.05° with respect to the D^4 product. While the sensitivity of the D^4 product to these changes in latitude and longitude only applies to the specific user-jammer spacing used, the results show that an asymmetric placement of the user and jammer in relation to the formation's ground track results in an increase in the D^4 product, and subsequently a decrease in the achievable null depth. This represents a key limitation of this system: the nulling performance depends on the orientation of the user and jammer in relation to the formation's ground track. Since the system is unable to control the placement of the user and jammer, a constellation of CubeSat formations is required to provide global coverage and maximize the likelihood that the user and jammer are oriented symmetrically in relation to the satellite ground track.

4.6 Constellations

Since the user and jammer orientation cannot be controlled by the communications system, a constellation is necessary to increase the likelihood that symmetry is achieved. A constellation for this communications system must maximize two metrics: zonal (or if possible, global) coverage and the likelihood that a favorable ground geometry exists. Additionally, the constellation must minimize the number of formations used. A favorable geometry has the following characteristics, listed in order of importance in achieving sufficient null-forming performance:

1. No spacing between user and jammer along satellite ground track

2. User and jammer equidistant from satellite ground track and lying along line perpendicular to ground track

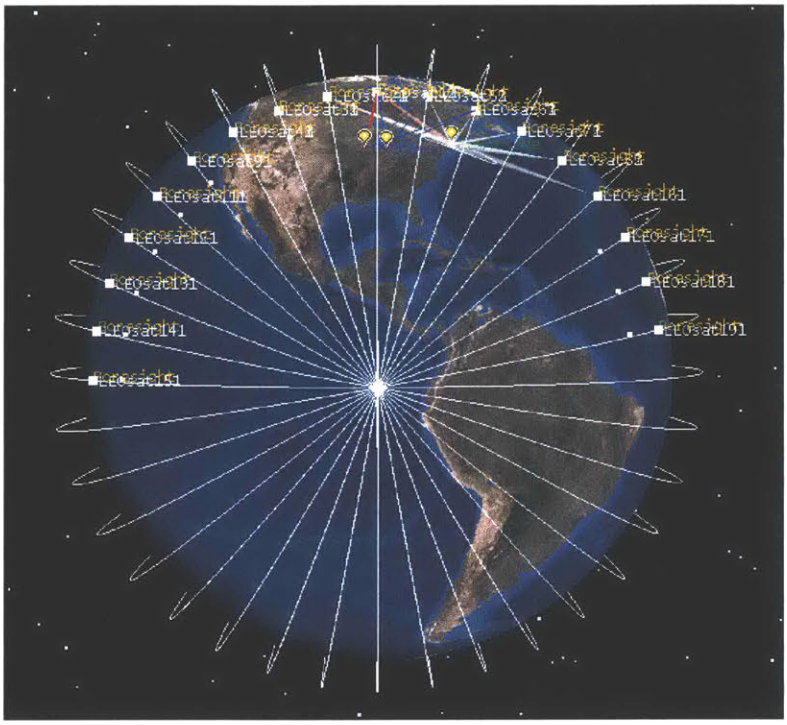

Figure 4-16: Constellation Maximizing Likelihood of a Symmetric Geometry Over Equator

For all of the simulations shown earlier, the formation passes between the user and jammer, minimizing the differential-delay when the user and jammer are spaced far apart along a line perpendicular to the satellite's ground track. When the user and jammer are spaced far apart, the formation must pass between the user and jammer in order to form sufficiently deep nulls. When the user and jammer are spaced close together, the formation may pass to the side of both the user and jammer and still null the jammer (this is fairly analogous to the case shown in Figure 4-12, but when the formation passes to one side of both the user and jammer, the user-jammer spacing must be minimized). However, since the jammer location is unknown, per the assumptions in Chapter 2, a constellation that has many formations in orbital planes that differ in RAAN and inclination or inclination only can maximize the likelihood that one formation sees a symmetric ground geometry.

Extensive research has been performed on designing satellite constellations to maximize coverage, such as the Walker and "streets-of-coverage" constellations [24]. Unfortunately, the Walker and "streets-of-coverage" constellations do not maximize zonal coverage and do not maximize the likelihood of a favorable geometry since they do not vary the orbital planes in inclination. The likelihood of a favorable geometry may be maximized by placing formations in orbital planes that intersect above one region due to their inclinations and RAANs. Figures 4-16 and 4-17 are two examples of this.

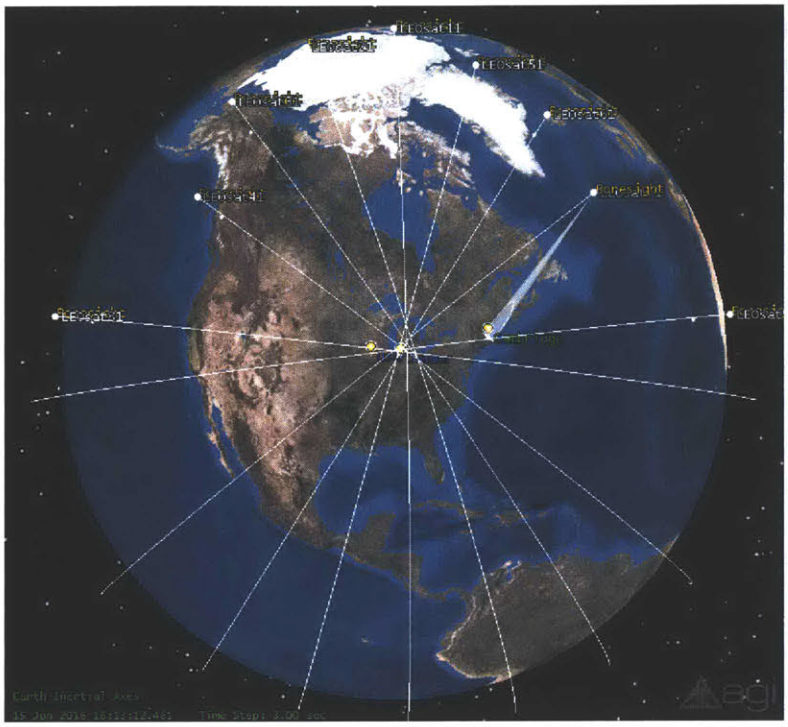

Figure 4-17: Constellation Maximizing Likelihood of a Symmetric Geometry Over United States

The constellation in Figure 4-16 uses 17 formations whereas the constellation in Figure 4-17 uses 9 formations, resulting in 51 and 27 satellites, respectively, assuming one formation per orbital plane. The area of interest is the area where the orbital planes intersect, and this region will be called the nulling region. At the nulling region, the ground geometry is most likely to be symmetric with respect to the satellite ground track. Not shown in the images is the nulling region that occurs on the opposite side of the Earth. One downside to this formation is that the orbital procession due to the J2 perturbation would cause the region to move, meaning a specific region cannot be guaranteed coverage for all time. This is a fairly serious flaw in this constellation and will require future work.

Another possible constellation is the Flower Constellation. Conceived at Texas A&M University in Dr. Daniele Mortari's lab, Flower Constellations aim to provide repeatable ground tracks [61]. A given Flower Constellation is defined by the number of petals, number of sidereal days to repeat the ground track, the number of satellites in the constellation, the orbit inclination, argument of perigee, and perigee altitude [61]. The inclination, argument of perigee, and perigee altitude are the same for each satellite in the constellation. The ability to control the repeatability of the constellation's ground track could offer a way to maximize the likelihood of obtaining a favorable geometry while providing global coverage. Additionally, multiple flower constellations, with orbits at different inclinations, could be used to maximize the likelihood of a symmetric ground geometry. Other modifications to the flower con-

stellation could result in a constellation optimized for this communications system.

Both constellations offer some advantages in positioning the formations in such a way as to offer more reliable jam-resistant communications. Since the focus of this thesis is not constellation design, the formulation of a constellation that maximizes zonal (or global) coverage, maximizes the likelihood of a favorable ground geometry, and minimizes the amount of formations will be left as future work.

Chapter 5

Summary & Future Work

5.1 Summary

This thesis shows the framework for a constellation of CubeSats in LEO aimed at providing jam-resistant communications for mobile users. The operational view, consisting of one user, one jammer, one ground station, and a constellation of CubeSat formations, where each formation contains 3 CubeSats, illustrates the system design. The system assumptions and constraints limit the scope of the problem. The lasercom transmitter and receiver from NODE [48, 23, 70, 62, 47, 63] are explained, and a link budget from [23] is shown to confirm that the laser transmitter and receiver provide sufficient downlink data rates for the proposed communications system. The parameters for the RF uplink were shown, and link budgets demonstrate that a handheld radio and a simple RF front-end may be used to communicate from the ground to the CubeSat.

Building off of the lasercom downlink from NODE, Chapter 3 shows the waveform design for NODE. PPM, RS coding, interleaving, and symbol mapping are described, and simulations demonstrate that the RS error-correction, when coupled with interleaving and symbol mapping, is sufficient to correct a channel BER of 1×10^{-4} to error-free when on a non-fading or fading binary channel. The framing system, including the acquisition sequence and the inter-symbol guard time, is presented, and autocorrelations reveal the features of the chosen acquisition sequences. The FPGA design consists of the modulator state machine, the interface between the CPU and the modulator, memory units, and the circuit used within the FPGA to test the modulator. The results show that the modulator functions correctly.

Chapter 4 shows the STFAP, which constitutes the ground-based beamformer. The inadequate performance with phase-coherent combining leads to the conclusion that null-forming is required in order to defeat the jammer. The D^4 product is discussed with regard to its effect on null-forming. Two formations are shown: the line-of-pearls formation and the abreast formation. STK simulations indicate that the abreast formation limits D^4 products when the inter-satellite spacing increases, making the abreast formation more favorable than the line-of-pearls formation. Additional STK simulations analyzing the placement of the user and jammer in relation to

the satellite ground track reveal the sensitivity of D^4 products to symmetry between the user, jammer, and satellite ground track. Two constellations, which attempt to maximize the likelihood that a favorable symmetry is seen by the formation, are proposed.

Four architectures could be used to provide jam-resistant communications for a CubeSat constellation in LEO:

1. Spread spectrum on uplink and downlink
2. GBBF on uplink and spread spectrum on downlink
3. Spread spectrum on uplink and lasercom on downlink
4. GBBF on uplink and lasercom on downlink.

The following figures of merit will be used in analyzing the various architectures:

1. Bandwidth
2. Complexity (user terminal)
3. Complexity (satellite)
4. Complexity (ground processing)
5. Size of ground station
6. Number of CubeSats required
7. Technical risk
8. Data rate
9. Scalability

The complexity of the user terminal and satellite refers to the additional signal processing and/or pointing required in the design. The complexity of ground processing pertains to the limitations associated with processing data on the ground, as seen in the STFAP. The scalability of the design is taken to involve the ability to attain higher data rates and/or support multiple users. A well-designed system should minimize the first six figures of merit while maximizing the last three. Each anti-jamming technology provides some advantages and disadvantages, as discussed briefly in Section 1.3. Tables 5.1 and 5.2 show the advantages and disadvantages of each anti-jamming technology with respect to the figures of merit. The figures of merit shown in Table 5.1 are those which are improved by the anti-jamming technology, while those shown in Table 5.2 are made worse by the anti-jamming technology.

Table 5.1: Advantages of Anti-Jamming Technology

	Uplink	Downlink
Spread Spectrum	4, 6, 7	4, 6, 7
GBBF	1, 2, 3, 9	N/A
Lasercom	N/A	1, 5, 8, 9

Table 5.2: Disadvantages Anti-Jamming Technology

	Uplink	Downlink
Spread Spectrum	1, 2, 3, 9	1, 5, 9
GBBF	4, 6, 7	N/A
Lasercom	N/A	3, 7

Architecture 1 uses spread spectrum on uplink and downlink, requiring a large amount of bandwidth on uplink and downlink. A spread spectrum bandwidth of approximate 100 MHz is required to mitigate the expected jammer with 20 dBW output power, and it is tough to acquire this much bandwidth at the lower-end of the UHF band as many handheld radios operate in that region. Spread spectrum requires an increase in complexity for the user and satellite terminals, and the use of orthogonal codes or hopping patterns requires precise timing synchronization between user terminals, further increasing the complexity of the user terminals. This increase in complexity requires the development of custom user terminals, increasing cost and development time. The limited power available on CubeSats requires the use of large, immobile ground stations or higher carrier frequencies for RF downlinks. Moreover, such a design does not offer much scalability. The uplink data rates are limited by the sizes of the apertures and the power on the spacecraft, and the large spread spectrum bandwidth required makes it tough to increase the information bandwidth on downlink. Code division multiple access (CDMA) or frequency division multiple access (FDMA) could be used with spread spectrum, but utilizing both effectively without losses would require precise timing between user transmitters on uplink and satellite transmitters on downlink. This design accepts the least amount of technical risk since it uses previously demonstrated techniques. Also, this design reduces cost as one CubeSat may be used rather than three. While this design could minimize risk and reduce cost, the complexities with the user terminal and satellite terminal, the large amount of bandwidth required at lower UHF frequencies, and the lack of scalability of this design make it less attractive.

Architecture 2 uses GBBF on uplink and spread spectrum on downlink. The satellite receiver front end would be very simple, and a currently fielded handheld transmitter, such as the AN/PRC-152 shown earlier, could be used as the user terminal. The uplink would only require the information bandwidth of 100 kHz since spread spectrum is not used. The spatial diversity in GBBF provides the opportunity for space division multiple access (SDMA), offering scalability. However, additional CubeSats will have to be used compared to architectures 1 and 3. GBBF has more complex ground processing, requiring the use of the sparse STFAP that sees degraded

performance at large D^4 products. Also, GBBF has not been demonstrated on CubeSats in LEO, so such a design accepts additional risk. Using spread spectrum on downlink requires a large amount of bandwidth, a large, immobile ground station or higher carrier frequencies, and increased complexity in the satellite transmitter. As stated with architecture 1, spread spectrum is not very scalable. The RF downlink could be a bottleneck, or require the use of unwieldy large ground stations, if the data rates on the RF-front end were increased. The support for handheld transmitters and scalability of the GBBF make this architecture a reasonable choice; however, the increase in bandwidth and limitations of RF on downlink are two downsides to this design.

Architecture 3 uses spread spectrum on uplink and lasercom on downlink. The lasercom downlink provides jam-resistance without requiring a large increase in bandwidth; as shown in Chapter 2, the lasercom downlink is capable of providing data rates as high as state-of-the-art custom RF transmitters and is scalable to much higher data rates. This allows the use of a very simple sample-and-forward method, much like a regenerative transponder, to be used on the satellite. Lasercom downlinks can use relatively small, transportable ground stations, making it easier to field lasercom ground stations. However, lasercom downlinks have not been demonstrated successfully on a CubeSat in orbit as of the writing of this thesis and cannot transmit through clouds, increasing the risk of using lasercom. A complete network of ground stations or optical crosslink capabilities could mitigate risks associated with cloud cover, thereby providing more reliable communications. Fortunately, the feasibility of optical crosslinks are being examined in MIT's STARLab, and the optimal placement of ground stations is examined in [27, 26]. Another downside to lasercom is that it increases the complexity on the satellite due to the strict pointing requirements, but the development of CubeSat capabilities and the NODE design make this less of a concern. The scalability, high data rates, and jam-resistance of lasercom make it a competitive choice as the downlink for the proposed system. Unfortunately, the downsides associated with using spread spectrum on the RF uplink, such as requiring a specific handheld transmitter, a spread spectrum bandwidth of 100 MHz at UHF, and a custom receive front-end, decrease the utility of this design.

Architecture 4 uses GBBF on uplink and lasercom on downlink. Using GBBF on uplink minimizes the complexity in the user terminals, allowing the system to be agnostic to the specific handheld transmitter used on the ground. Additionally, the use of GBBF allows for the use of a very simple, COTS RF front-end on the satellite, reducing cost and development time. The data rate for the lasercom downlink is sufficient to allow a simple sample-and-forward method on the CubeSat, simplifying the design and reducing cost. The large amount of bandwidth available at optical frequencies allows the downlinks to be separated by wavelength, giving all CubeSats access to one ground station without any significant negative effects. The relatively small and transportable optical ground stations in the NODE design are cheaper than large RF ground stations, making it easier to field a network of optical ground stations. Additionally, the GBBF uplink and lasercom downlink are scalable, allowing multiple access and increased data rates in future versions. However, as stated earlier, both GBBF and lasercom have some disadvantages, and they both share the disadvantage

of increased technical risk. GBBF requires an increased number of CubeSats and sees more complex ground processing. Lasercom requires a complex optical transmitter with strict pointing requirements. While this design has its disadvantages, the scalability of this design, the relative simplicity of the CubeSats, the low cost for the CubeSats, and the low complexity of user terminals outweigh the disadvantages, making this design a great choice to supplement MILSATCOM by providing secure communications to mobile users in hostile environments.

5.2 Future Work

5.2.1 Real-Time Lasercom Implementation

After a first successful demonstration of NODE, the transmitter and receiver will need to be revised to work in real time. This task centers on FPGA design. All of the tasks performed in the CPU in 3-1, with the exception of padding, would need to be implemented in the FPGA.

For the transmitter, this involves using Reed-Solomon cores, which Xilinx provides for free with the ISE WebPACK, in parallel to perform encoding in real-time. Implementing the interleaver would require sufficient high-performance memory in the FPGA, a switch to a convolutional interleaver, or both. The bit-to-symbol mapping function could be tricky to realize on the FPGA as different PPM orders require different pathways within the FPGA; a state machine, where the PPM order determines the state, could be used with multiple asymmetric FIFOs to accomplish bit-to-symbol mapping .

For the receiver, the sampling, clock recovery, demodulation, symbol-to-bit mapping , deframing, deinterleaving, and RS decoding would need to be performed within an FPGA. An ADC with a sampling rate > 400 MS/s and an evaluation board would need to be used in conjunction with a fairly capable FPGA, such as one that has an FPGA Mezzanine Card (FMC) Low-Pin Count (LPC) or High-Pin Count (HPC) connector, to transfer the samples from the ADC to the FPGA at Gbps data rates. Moving the clock recovery, demodulation, and deframing algorithms to the FPGA would require the implementation of the post-processing functions being developed by graduate student Joe Kusters. As with the interleaving, the deinterleaving would require sufficient memory on the FPGA, or a convolutional interleaver, in order to perform the deinterleaving. Xilinx RS decoder cores could be used for decoding, and placing decoders in parallel will allow for real-time processing. The most significant tasks remaining for the receiver involve integrating an FPGA with a high-performance ADC and implementing the clock recovery, demodulation, and deframing algorithms on the FPGA.

5.2.2 Optimization of CubeSat Constellation

A stated in Chapter 4, the goal of the CubeSat constellation is to maximize coverage and the likelihood of obtaining a symmetric geometry while minimizing the number

of formations. One way to obtain this is through the use of a flower constellation or the "pin-wheel" constellation. Unfortunately, neither of these constellations are optimized for this specific application. A constellation that varies each formation's ground track over time will provide a symmetric geometry at some point in time. Spacing formations at regular intervals in the constellation while ensuring they have different ground tracks is likely to provide global coverage and increase the chances one location sees a favorable geometry. The values to be optimized for each formation's orbit are the inclination, RAAN, and true anomaly since the orbit altitude is fixed by the link budgets.

5.2.3 Analysis of Ground-Station Placement and Crosslinks

The placement of ground stations is important in ensuring reliable communications with low latency. In order to downlink the data reliably, the ground stations must be placed in a way so that clouds and other atmospheric phenomenon do not prevent lasercom downlinks from occurring. Also, the placement of ground stations must ensure that the latency between reception of the uplink and the downlink of the data is negligible. To minimize latency, optical ground stations should be located in areas where communications coverage is desired and should be in view of the formation during a pass. The optimization must take into account the amount of cloud coverage for certain areas around the globe so as to find areas with low possibility of cloud coverage, offering more opportunities for lasercom downlinks. A ground-station placement optimization has been performed for current satellites conducting scientific, Earth observation, and weather monitoring missions in LEO by [27, 26]; however, an optimal placement would need to be determined for the chosen constellation.

5.2.4 Analysis with More Users and Jammers

While this thesis only examined the case of one user and one jammer, many real-world scenarios would expect to see multiple users and multiple jammers. An adaptive array with N elements has N degrees of freedom and can form up to $N-1$ nulls, meaning $N-1$ interference sources may be suppressed at any one time [30]. When forming beams for multiple users, an array of N elements may support up to N users[54], with each beam consuming one degree of freedom. When multiple jammers and multiple users are present, degrees of freedom may be used to null jammers or support additional users. For the three-element array shown in this thesis, three users could be supported, one user could be supported while nulling two jammers, or two users could be supported while nulling one jammer.

Even though sufficient degrees of freedom exist to form additional nulls and beams, the delay and Doppler environment is the most constraining aspect of the GBBF. As seen in this thesis, the delay and Doppler environment in LEO requires the use of a STFAP, which has to be made sparse in order to be practical. Further analysis must examine whether the STFAP will be sparse enough to be practical with the addition of more users and more jammers. The increased delay and Doppler spread coupled with additional D^4 products caused by multiple users and jammers could make the

STFAP infeasible. This could be determined by performing additional analysis in STK.

5.2.5 Analyze Crosslinks

Optical or RF crosslinks could make the communications system more robust by providing a means to route information to CubeSats that are in sight of an optical ground station. Crosslinks could loosen the constraints on the optimal placement of optical ground stations and decrease latency. However, crosslinks would add more complexities to the space terminals. The CubeSats would have to include optical receivers, which would increase the required SWaP and development time. Coordinating crosslinks between the CubeSats in one formation and the CubeSats in another formation is a tough problem and could possibly require RF crosslinks for the purposes of coordinating movements. Additionally, routing the information requires the use of sophisticated data-link and network layer protocols. An analysis on the benefits of crosslinks, assuming an optimized constellation and ground-station network, would help in determining if crosslinks are required.

5.2.6 Adding forward link

The proposed communications system only provides the reverse link, the link from the user to the command center, and not a forward link from the command center to the user. A forward link could be implemented as a broadcast-type link from a command center to provide communications to all forward-deployed mobile units. A broadcast-type link would be very simple to implement, and a trade could be performed to determine which, if any, anti-jamming techniques would be useful on such a link.

A forward link could take advantage of the CubeSat formation and perform GBBF on transmit. This link would require a lasercom uplink on the ground, an optical receiver on the CubeSats, and RF front ends on the CubeSats capable of output powers in excess of 5 W. Additionally, knowledge of the location of the user and any jammers would be required in order to perform GBBF on transmit, and multiple beams or nulls could be formed with GBBF on transmit if the user and jammer locations were known. Due to the changes in the design and additional complexities required to do GBBF on transmit, such a forward link could be a candidate feature for a second version of the proposed communications system.

Appendix A

Oscilloscope Code

```
from __future__ import division
import agilent4000x
import sys

outfile = sys.argv[1]
trange = sys.argv[2]
timerange = float(trange)
ppm_order = int(sys.argv[3])
expected\_slot = int(sys.argv[4])
# setup scope
scope = agilent4000x.Agilent4154A()
scope.cmd('*RST')
scope.cmd(':POD1:DISPlay 1')
scope.cmd(':POD1:THReshold TTL')
scope.cmd(':timebase:mode main')
scope.cmd(':timebase:range ' + trange)
scope.cmd(':timebase:delay 0')
scope.cmd(':timebase:position 0')
scope.cmd(':trigger:mode edge')
scope.cmd(':trigger:slope positive')
scope.cmd(':trigger:edge:source digital5')
scope.cmd(':waveform:source pod1')
scope.cmd(':acquire:mode rtime')
scope.cmd(':acquire:type normal')
scope.cmd(':acquire:count 2')
scope.cmd(':acquire:complete 100')
scope.cmd(':waveform:unsigned 1')
scope.cmd(':waveform:format byte')
scope.cmd(':waveform:points:mode max')

# print configuration of scope
```

```
scope.cmd('timebase:range?')
print "Time range: ", scope.resp()
scope.cmd('acquire?')
print "Acquire information: ", scope.resp()
scope.cmd('acquire:srate?')
sample_rate = float(scope.resp())
print "Sample rate: ", str(sample_rate)
scope.cmd('acquire:points?')
acq_points = int(scope.resp())
print "Number of Points: ", str(acq_points)
scope.cmd('acquire:count?')
print "Acquire count: ", scope.resp()
scope.cmd('trigger:mode?')
print "Trigger mode: ", scope.resp()
scope.cmd('trigger:slope?')
print "Trigger slope: ", scope.resp()
scope.cmd('trigger:level?')
print "Trigger source: ", scope.resp()
scope.cmd('waveform:unsigned?')
print "Waveform is unsigned: ", scope.resp()
scope.cmd('waveform:format?')
print "Waveform format: ", scope.resp()
scope.cmd('waveform:source?')
print "Waveform source: ", scope.resp()
scope.cmd(':waveform:points:mode?')
print "Waveform points mode: ", scope.resp()
scope.cmd(':waveform:type?')
print "Waveform type: ", scope.resp()

# collect data
scope.cmd(':digitize digital5,digital7')
scope.cmd(':waveform:segmented:count?')
print "waveform segmented count: ", scope.resp()
scope.cmd(':waveform:points?')
wavepoints = int(scope.resp())
print "waveform points: ", str(wavepoints)
scope.cmd('waveform:count?')
print "Waveform count: ", scope.resp()

scope.cmd(':waveform:data?')
returned_samples = scope.resp()
stripped_samples = returned_samples[10:]
print "Header is :",returned_samples[:10]
samples = []
for byte in stripped_samples:
```

```
        try:
                samples.append(ord(byte))
        except:
                print "couldn't convert:", byte

# do bitmask to separate digital 5 (sym sync) from digital 7
    (ppm sequence)
bitmask5 = 0x20
bitmask7 = 0x80
samples_dig5 = []
for byte in samples:
        samples_dig5.append((byte & bitmask5) >> 5)
samples_dig7 = []
for byte in samples:
        samples_dig7.append((byte & bitmask7) >> 7)

# determine campling rate and decimation of oscope
numpoints = sample_rate*timerange
decimation = 4
realsample_rate = sample_rate/decimation
realsamplesperslot = realsample_rate/200e6
samplesperslot = int(realsample_rate/200e6)
slotsperextrasample = 1/(realsamplesperslot - samplesperslot)
samplesbeforedelete = samplesperslot*slotsperextrasample
samplespersymbol = samplesperslot*ppm_order # add guard time
    later

# decimate so that we have an integer number of samples per
    slot
samples_dig5.append(0)
samples_dig7.append(0)
samples_sync = []
samples_ppm = []
for i,byte in enumerate(samples_dig5):
        if (i % (samplesbeforedelete+1)) == 0:
                pass
        else:
                samples_sync.append(byte)
for i,byte in enumerate(samples_dig7):
        if (i % (samplesbeforedelete+1)) == 0:
                pass
        else:
                samples_ppm.append(byte)

# Demodulation
```

```
# putting each symbol for sym_sync and ppm_signal in it's own
    list
count = 0
active_sample = 0
ppm_symbols = [[[],[]]]
for i,sample in enumerate(samples_sync):
        if samples_sync[i-1] == 0 and sample == 1:
                # start of symbol (count is distance from
                    rising edge of first sym)
                ppm_symbols.append([[sample],[samples_ppm[i
                    ]]])
                count += 1
        else:
                # not start of symbol
                ppm_symbols[count][0].append(sample)
                ppm_symbols[count][1].append(samples_ppm[i])

if (sum(ppm_symbols[0][0]) == 0):
        del ppm_symbols[0]
if (len(ppm_symbols[-1][0]) < len(ppm_symbols[-2][0])):
        del ppm_symbols[-1]

ppm_demod = []
list_sampspersym = []
underflows = 0
erasures = 0
for symbol in ppm_symbols:
        count_sym = 0
        count_ppm = 0
        slot_sym = 0
        slot_ppm = 0
        list_sampspersym.append(len(symbol[0]))
        # if length of symbol[0] is greater than
            samplesperslot*ppm_order*(1.25) + 10, declare
            erasure
        if len(symbol[0]) > (samplesperslot*(1.25*ppm_order)
            + 8):
                print('FIFO underflow error')
                underflows += int(len(symbol[0])/(
                    samplesperslot*(1.25*ppm_order)))
        # determine location of active pulse
        for i,sample in enumerate(symbol[1]):
                if ((symbol[1][i-1] == 1) and (sample == 1)
                    and (symbol[1][i+1] == 1)):
                        count_ppm += 1
```

```
                        slot_ppm += i
        try:
                ppm_offset = slot_ppm/count_ppm
                # symbol value is location for active slot
                   since symbols are aligned by sym_sync
                symbol_value = (ppm_offset)/samplesperslot
                ppm_demod.append(int(symbol_value))
        except:
                erasures += 1
                print("Erasure")
print "Demodulated sequence: \n",ppm_demod
print

avgsampperslot = (sum(list_sampspersym)/len(list_sampspersym)
   )/(1.25*ppm_order)
print "Avg samples per slot: ",avgsampperslot
print

# write to file
chr_samples_sync = [chr(sample) for sample in samples_sync]
chr_samples_ppm = [chr(sample) for sample in samples_ppm]
write_samples =  chr_samples_ppm
with open(outfile,'wb') as g:
        g.write(''.join(write_samples))

# Calculate SER
errors = 0
for symbol in ppm_demod:
        if symbol != expected_slot:
                errors += 1
errors += erasures
errors += underflows
SER = float(errors/(len(ppm_demod) + erasures + underflows))

# Print results to terminal
print "Number of samples: ", len(samples)

print "Number of erasures: ", erasures

print "Number of FIFO underflow errors: ", underflows

print "SER: ", SER

scope.cmd(":recall:setup 'cz_29Nov16'")
scope.disconnect()
```

Bibliography

[1] analyticalspace. https://www.analyticalspace.com.

[2] BridgeSat. http://www.bridgesatinc.com.

[3] USRP B200mini Series. https://www.ettus.com/content/files/USRP_B200mini_Data_Sheet.pdf. Accessed: 2017-05-09.

[4] RF Agile Transceiver: AD9364. http://www.analog.com/media/en/technical-documentation/data-sheets/AD9364.pdf, 2014. Rev. C.

[5] Voxtel Catalog rev. 06. http://voxtel-inc.com/files/Voxtel-Catalog.pdf, 2015.

[6] High Photon Efficiency Optical Communications Coding & Synchronization. Recommended Standard CCSDS XXX.0-B-0.6, CCSDS, jun 2016.

[7] Piero Angeletti, Nader Alagha, and Salvatore D'Addio. Space/ground beamforming techniques for satellite communications. In *Antennas and Propagation Society International Symposium (APSURSI), 2010 IEEE*, pages 1–4. IEEE, 2010.

[8] Saptarshi Bandyopadhyay, Rebecca Foust, Giri P Subramanian, Soon-Jo Chung, and Fred Y Hadaegh. Review of Formation Flying and Constellation Missions Using Nanosatellites. *Journal of Spacecraft and Rockets*, (0):567–578, 2016.

[9] Al Bartko, AQ Le, Thomas N Thomas, and Lenny Sheet. Perceived issues associated with military use of satellite based personal communications systems. In *Military Communications Conference, 1995. MILCOM'95, Conference Record, IEEE*, volume 3, pages 1224–1228. IEEE, 1995.

[10] Abhijit Biswas, Bogdan Oaida, Kenneth S. Andrews, Joseph M. Kovalik, Matthew Abrahamson, and Malcolm W. Wright. Optical payload for lasercomm science (OPALS) link validation during operations from the ISS , 2015.

[11] Daniel W Bliss and Siddhartan Govindasamy. *Adaptive wireless communications: MIMO channels and networks.* Cambridge University Press, 2013.

[12] Grant Bonin, Niels Roth, Scott Armitage, Josh Newman, Ben Risi, and Robert E Zee. CanX–4 and CanX–5 Precision Formation Flight: Mission Accomplished! 2015.

[13] Don M Boroson, Bryan S Robinson, Daniel V Murphy, Dennis A Burianek, Farzana Khatri, Joseph M Kovalik, Zoran Sodnik, and Donald M Cornwell. Overview and results of the lunar laser communication demonstration. In *SPIE LASE*, pages 89710S–89710S. International Society for Optics and Photonics, 2014.

[14] R. Burch. The case for disaggregation of U.S. MILSATCOM. In *2011 - MILCOM 2011 Military Communications Conference*, pages 2286–2291, Nov 2011.

[15] D. O. Caplan, R. T. Schulein, J. J. Carney, M. L. Stevens, and S. J. Spector. WDM laser transmitters for mobile free-space laser communications , 2016.

[16] David O Caplan. Laser communication transmitter and receiver design. *Journal of Optical and Fiber Communications Reports*, 4(4-5):225–362, 2007.

[17] DO Caplan, JJ Carney, JJ Fitzgerald, I Gaschits, R Kaminsky, G Lund, SA Hamilton, RJ Magliocco, RJ Murphy, HG Rao, et al. Multi-rate DPSK optical transceivers for free-space applications. In *SPIE LASE*, pages 89710K–89710K. International Society for Optics and Photonics, 2014.

[18] Celestron. CPC 1100 GPS (XLT) Computerized Telescope. `http://www.celestron.com/browse-shop/astronomy/telescopes/cpc-telescopes/cpc-1100-gps-%28xlt%29-computerized-telescope`.

[19] Joey Chang. DARPA wants more efficient, jam-resistant RF communications. `http://defensesystems.com/articles/2014/07/01/darpa-anti-jamming-rf-hermes.aspx`. Accessed: 2017-04-25.

[20] Paolo Chini, Giovanni Giambene, and Sastri Kota. A survey on mobile satellite systems. *International Journal of Satellite Communications and Networking*, 28(1):29–57, 2010.

[21] McClelland Chris. FPGALink User Manual. `https://github.com/makestuff/libfpgalink/wiki/FPGALink`.

[22] Craig Clark. Huge Power Demand... Itsy-Bitsy Satellite: Solving the Cubesat Power Paradox. `http://digitalcommons.usu.edu/smallsat/2010/all2010/19/`, 2010.

[23] Emily Clements, Raichelle Aniceto, Derek Barnes, David Caplan, James Clark, Iñigo del Portillo, Christian Haughwout, Maxim Khatsenko, Ryan Kingsbury, Myron Lee, et al. Nanosatellite optical downlink experiment: design, simulation, and prototyping. *Optical Engineering*, 55(11):111610–111610, 2016.

[24] Stefania Cornara, Theresa W Beech, Miguel Belló-Mora, and Guy Janin. Satellite constellation mission analysis and design. *Acta Astronautica*, 48(5):681–691, 2001.

[25] Col Chuck Cynamon. Military Satellite Communications. https://afcea-la.org/sites/afcea-la.org/files/AFCEA%20Brief%20Feb2010.pdf, Feb 2010.

[26] Iñigo del Portillo, Marc Sanchez, Bruce Cameron, and Edward Crawley. Optimal Location of Optical Ground Stations to Serve LEO Spacecraft. In *Aerospace Conference*, pages 1–16, 2017.

[27] Iñigo del Portillo Barrios. Optimal locations for the ground segment of optical space communications networks. Master's thesis, Massachusetts Institute of Technology, 2016.

[28] Fred J Dietrich. The Globalstar satellite cellular communication system: design and status. In *Wescon/97. Conference Proceedings*, pages 180–186. IEEE, 1997.

[29] EN ETSI. 302 307 V1. 2.1 (2009-08) European Standard (Telecommunications series) Digital Video Broadcasting (DVB). *Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (DVB-S2).*

[30] Alan J Fenn. *Adaptive Antennas and Phased Arrays for Radar and Communications.* Artech House, 2007.

[31] Benjamin Friedlander and Shimon Scherzer. Beamforming versus transmit diversity in the downlink of a cellular communications system. *IEEE Transactions on Vehicular Technology*, 53(4):1023–1034, 2004.

[32] David A Fritz, Bharat T Doshi, Andrew C Oak, Steven D Jones, Jack L Burbank, Harry L Miller, John D Oetting, Ryan M Collins, Lino A Gonzalez, and Robert A Nichols. Military satellite communications: space-based communications for the global information grid. *Johns Hopkins APL technical digest*, 27(1):32–40, 2006.

[33] Chad Frost, E Agasid, et al. Small spacecraft technology state of the art. *Moffett Field, California: NASA Ames Research Center*, 2015.

[34] G Gallinaro, S Cioni, F Vanin, O Vidal, G Huggins, M Gross, S Andrenacci, S Chatzinotas, and E Tirro. Future ground beamforming. In *Advanced Satellite Multimedia Systems Conference and the 14th Signal Processing for Space Communications Workshop (ASMS/SPSC), 2016 8th.* IEEE, 2016.

[35] T Aaron Gulliver. Matching Q-ary Reed-Solomon codes with M-ary modulation. *IEEE Transactions on Communications*, 45(11):1349–1353, 1997.

[36] Emre Gündüzhan and K. Dewayne Brown. Narrowband Satellite Communications: Challenges and Emerging Solutions. *Johns Hopkins APL technical digest*, 33(1):52–56, 2015.

[37] Jon Hamkins. *Pulse Position Modulation*, pages 492–508. John Wiley & Sons, Inc., 2007.

[38] Fredric J Harris, Chris Dick, and Michael Rice. Digital receivers and transmitters using polyphase filter banks for wireless communications. *IEEE transactions on microwave theory and techniques*, 51(4):1395–1412, 2003.

[39] Gary F Hatke, Keith W Forsythe, Andrew L McKellips, and Tri T Phuong. Space-Time-Frequency Adaptive Processor Design for Ultra-Sparse Apertures. In *2006 Fortieth Asilomar Conference on Signals, Systems and Computers*, pages 2262–2266. IEEE, 2006.

[40] Hamid Hemmati. *Near-Earth Laser Communications*. CRC Press, 2009.

[41] Hamid Hemmati and David Caplan. Optical Satellite Communications. *Optical Fiber Telecommunications Volume VIB: Systems and Networks*, page 121, 2013.

[42] Keith Hogie, Ed Criscuolo, Asoka Dissanayaka, Bruce Flanders, Haleh Safavi, and Jeff Lubelczyk. Tracking and Data Relay Satellite System Demand Access System Augmentation. `http://gsaw.org/wp-content/uploads/2015/03/2015s06hogie.pdf`.

[43] Telecommunication Standardization Sector ITU. Forward error correction for submarine systems. *Tech. Recommendation G*, 975:G709.

[44] Siegfried Janson, Richard Welle, Todd Rose, Darren Rowen, Brian Hardy, Richard Dolphus, Patrick Doyle, Addison Faler, David Chien, Andrew Chin, et al. The NASA Optical Communications and Sensor Demonstration Program: Initial Flight Results. 2016.

[45] Young-Sik Kim, Ji-Woong Jang, Sang-Hyo Kim, and Jong-Seon No. New quaternary sequences with optimal autocorrelation. In *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*, pages 286–289. IEEE, 2009.

[46] Mak King and Michael J. Riccio. Military Satellite Communications: Then and Now. `http://www.aerospace.org/crosslinkmag/spring-2010/military-satellite-communications-then-and-now/`. Accessed: 2017-04-27.

[47] RW Kingsbury, DO Caplan, and KL Cahoy. Compact optical transmitters for cubesat free-space optical communications. In *SPIE LASE*, pages 93540S–93540S. International Society for Optics and Photonics, 2015.

[48] Ryan W Kingsbury. *Optical Communications for Small Satellites*. PhD thesis, Massachusetts Institute of Technology, 2015.

[49] Bryan Klofas. High-Speed Radio Options. `http://mstl.atl.calpoly.edu/~bklofas/Presentations/DevelopersWorkshop2014/Klofas_High-Speed_Radio_Options.pdf`, Apr 2014.

[50] Bryan Klofas and Kyle Leveque. A Survey of CubeSat Communication Systems: 2009-2012. https://www.klofas.com/papers/Klofas_Communications_Survey_2009-2012.pdf, 2013.

[51] Y. Koyama, M. Toyoshima, Y. Takayama, H. Takenaka, K. Shiratama, I. Mase, and O. Kawamoto. SOTA: Small Optical Transponder for micro-satellite. In *2011 International Conference on Space Optical Systems and Applications (ICSOS)*, pages 97–101, May 2011.

[52] Herbert J. Kramer. AeroCube-4. https://directory.eoportal.org/web/eoportal/satellite-missions/a/aerocube-4. Accessed: 2017-03-14.

[53] Herbert J. Kramer. CanX-4&5 (Canadian Advanced Nanospace eXperiment-4&5). https://directory.eoportal.org/web/eoportal/satellite-missions/content/~/article/canx45. Accessed: 2017-03-14.

[54] John Litva and Titus K Lo. *Digital beamforming in wireless communications.* Artech House, Inc., 1996.

[55] Christopher John Lowe, Malcolm Macdonald, and Stephen Greenland. Through-life modelling of nano-satellite power system dynamics. In *64th International Astronautical Congress 2013*, pages Article–IAC, 2013.

[56] Kris Maine, Carrie Devieux, and Pete Swan. Overview of IRIDIUM satellite network. In *WESCON/'95. Conference record.'Microelectronics Communications Technology Producing Quality Products Mobile and Portable Power Emerging Technologies'*, page 483. IEEE, 1995.

[57] Arash Mehrparvar, D Pignatelli, J Carnahan, R Munakat, W Lan, A Toorian, A Hutputanasin, and S Lee. Cubesat design specification rev. 13. *The CubeSat Program, Cal Poly San Luis Obispo, US*, 2014.

[58] Jeffrey A Mendenhall, Lawrence M Candell, Pablo I Hopman, George Zogbi, Don M Boroson, David O Caplan, Constantine J Digenis, David R Hearn, and Ryan C Shoup. Design of an optical photon counting array receiver system for deep-space communications. *Proceedings of the IEEE*, 95(10):2059–2069, 2007.

[59] B Moision and J Hamkins. Coded modulation for the deep-space optical channel: serially concatenated pulse-position modulation. *IPN Progress Report*, 42(161):1–26, 2005.

[60] Robert A Monzingo, Randy L. Haupt, and Thomas W Miller. *Introduction to Adaptive Arrays.* Scitech publishing, second edition, 2011.

[61] Daniele Mortari, Matthew P Wilkins, and Christian Bruccoleri. The flower constellations. *Journal of Astronautical Sciences*, 52(1):107–127, 2004.

[62] Tam Nguyen. Laser Beacon Tracking for Free-Space Optical Communication on Small-Satellite Platforms in Low-Earth Orbit. Master's thesis, Massachusetts Institute of Technology, 2015.

[63] Tam Nguyen, Kathleen Riesing, Ryan Kingsbury, and Kerri Cahoy. Development of a pointing, acquisition, and tracking system for a cubesat optical communication module. In *SPIE LASE*, pages 93540O–93540O. International Society for Optics and Photonics, 2015.

[64] Bogdan V Oaida, William Wu, Baris I Erkmen, Abhijit Biswas, Kenneth S Andrews, Michael Kokorowski, and Marcus Wilkerson. Optical link design and validation testing of the Optical Payload for Lasercomm Science (OPALS) system. In *SPIE LASE*, pages 89710U–89710U. International Society for Optics and Photonics, 2014.

[65] John D. Oetting and Tao Jen. The Mobile User Objective System. *Johns Hopkins APL technical digest*, 30(2):103–112, 2011.

[66] D. H. Phung, E. Samain, N. Maurice, D. Albanesse, H. Mariey, M. Aimar, G. M. Lagarde, G. Artaud, J. L. Issler, N. Vedrenne, M. T. Velluet, M. Toyoshima, M. Akioka, D. Kolev, Y. Munemasa, H. Takenaka, and N. Iwakiri. Telecom scintillation first data analysis for DOMINO - laser communication between SOTA, onboard socrates satellite, and MEO OGS. In *2015 IEEE International Conference on Space Optical Systems and Applications (ICSOS)*, pages 1–7, Oct 2015.

[67] Kevin J Quirk, Jonathan W Gin, and Meera Srinivasan. Optical PPM synchronization for photon counting receivers. In *Military Communications Conference, 2008. MILCOM 2008. IEEE*, pages 1–7. IEEE, 2008.

[68] Irving S Reed, John D Mallett, and Lawrence E Brennan. Rapid convergence rate in adaptive arrays. *IEEE Transactions on Aerospace and Electronic Systems*, (6):853–863, 1974.

[69] M. Richharia and Leslie Westbrook. *Satellite Systems for Personal Applications: Concepts and Technology.* Wireless communications and mobile computing. Chichester, West Sussex, U.K. ; Hoboken, NJ : Wiley, 2010., 2010.

[70] Kathleen Michelle Riesing. Development of a Pointing, Acquisition, and Tracking System for a Nanosatellite Laser Communications Module. Master's thesis, Massachusetts Institute of Technology, 2015.

[71] Christopher Schmidt, Martin Brechtelsbauer, Fabian Rein, and Christian Fuchs. OSIRIS Payload for DLR's BiROS Satellite. In *International Conference on Space Optical Systems and Applications 2014*, 2014.

[72] R Scholtz and L Welch. Group characters: Sequences with good correlation properties. *IEEE Transactions on Information Theory*, 24(5):537–545, 1978.

[73] Bernard Sklar. *Digital Communications*, volume 2. Prentice Hall NJ, 2001.

[74] Leilei Song, Meng-Lin Yu, and Michael S Shaffer. 10-and 40-Gb/s forward error correction devices for optical communications. *IEEE journal of solid-state circuits*, 37(11):1565–1573, 2002.

[75] Meera Srinivasan, Kevin Birnbaum, Michael Cheng, and Kevin Quirk. A post-processing receiver for the lunar laser communications demonstration project. In *SPIE LASE*, pages 86100Q–86100Q. International Society for Optics and Photonics, 2013.

[76] Warren L Stutzman and Gary A Thiele. *Antenna Theory and Design*. John Wiley & Sons, 2013.

[77] J Tronc, P Angeletti, Nuan Song, Martin Haardt, J Arendt, and Gennaro Gallinaro. Overview and comparison of on-ground and on-board beamforming techniques in mobile satellite service applications. *International Journal of Satellite Communications and Networking*, 32(4):291–308, 2014.

[78] Barry D Van Veen and Kevin M Buckley. Beamforming: a versatile approach to spatial filtering. *IEEE assp magazine*, 5(2):4–24, 1988.

[79] Anne Wainscott-Sargent. Defense Eyes Lower Risk High-Throughput Future. `http://interactive.satellitetoday.com/via/september-2016/defense-eyes-lower-risk-high-throughput-future/`. Accessed: 2017-04-25.

[80] John Walker, Bob Day, and Sarah Xie. Architecture, implementation and performance of ground-based beam forming in the DBSD G1 mobile satellite system. In *28th AIAA International Communications Satellite Systems Conference (ICSSC-2010)*, page 8824, 2010.

[81] Richard P Welle, S Janson, D Rowen, and T Rose. Cubesat–scale laser communications. In *National Space Symposium*, 2015.

[82] MM Willis, BS Robinson, ML Stevens, BR Romkey, JA Matthews, JA Greco, ME Grein, EA Dauler, AJ Kerman, D Rosenberg, et al. Downlink synchronization for the lunar laser communications demonstration. In *Space Optical Systems and Applications (ICSOS), 2011 International Conference on*, pages 83–87. IEEE, 2011.

[83] K Wilson, A Vaughan, J Wu, D Mayes, J Maloney, and R Sobek. Preliminary characterization results of the optical communications telescope laboratory telescope. *The Interplanetary Network Progress Report*, 42:161, 2005.

[84] Kirk Woellert, Pascale Ehrenfreund, Antonio J Ricco, and Henry Hertzfeld. Cubesats: Cost-effective science and technology platforms for emerging and developing nations. *Advances in Space Research*, 47(4):663–684, 2011.

[85] Hyosang Yoon, Kathleen Riesing, and Kerri Cahoy. Satellite Tracking System using Amateur Telescope and Star Camera for Portable Optical Ground Station. 2016.

[86] Xiaoming Zhu and Joseph M Kahn. Communication techniques and coding for atmospheric turbulence channels. *Journal of Optical and Fiber Communications Reports*, 4(6):363–405, 2007.

[87] Rodger Ziemer and William H Tranter. *Principles Of Communications: Systems Modulation, and Noise.* John Wiley & Sons, 2015.