Spring 2015

Software Defined Radio Implementation Of Ds-Cdma In Inter-Satellite Communications For Small Satellites

Jr. Frank Pinto
North Carolina Agricultural and Technical State University

Software Defined Radio Implementation of DS-CDMA in Inter-Satellite Communications for Small Satellites

Frank Pinto Jr.

North Carolina A&T State University

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Department: Electrical & Computer Engineering

Major: Electrical Engineering

Major Professor: Dr. Fatemeh Afghah

Greensboro, North Carolina

2015

The Graduate School
North Carolina Agricultural and Technical State University

This is to certify that the Master's Thesis of

Frank Pinto Jr.

has met the thesis requirements of
North Carolina Agricultural and Technical State University

Greensboro, North Carolina
2015

Approved by:

Dr. Fatemeh Afghah Major Professor	Dr. Numan Dogan Committee Member
Dr. William Edmonson Committee Member	Dr. John Kelly Department Chair

Dr. Sanjiv Sarin
Dean, The Graduate School

Biographical Sketch

Frank Pinto Jr. is a 24 year old from Oxon Hill, MD. In 2008 he attended Virginia State University as a freshmen majoring in Computer Engineering and graduated with his Bachelors of Science degree in 2012. He served as the historian for the National Society of Black Engineers (NSBE) club and was also a member of IEEE. In May 2011 he interned with NASA at the Goddard Space Flight Center in Maryland and provided research on RF antennas used for space flight testing. The following summer of 2012 he interned with Oak Ridge National Laboratory and conducted bio-medical imaging research covering medical image analysis, data visualization, eye tracking, and human computer interfaces. From his research there he was able to produce a publication of his research in the Medical Physics Journal. In the summer of 2013 he also interned at Eglin Air Force Base as a contractor engineer and worked with digital image processing. Finally in the summer of 2014 he interned with the National Security Agency in which he served on a team working on Mobile Software Defined Radio. In the fall of 2013 he was admitted into the Electrical & Computer Engineering program at North Carolina A&T State University. His research interests lies in wireless communications, circuits, and telecommunications. Currently he holds an overall GPA of 3.55.

Acknowledgments

First and foremost I would like to thank God because none of this would be possible without him. I would like to thank my parents for their continued support in my education. Thanks to Dr. Kelly and Dr. Kim for words of encouragement when needed. I would like to thank my advisor, Dr. Fatemah Afghah, for her constant support and patience with me when it came to learning more about this research. Without her help I would have not made it this far in my research. I would also like to thank my committee, Dr. Edmonson and Dr. Dogan for their valuable comments. My colleagues Radhika Radhakrishnan, Christopher Oseigbu, and Nima Namvar, I thank you all for making this research experience a great one. Lastly, I would like to thank the GNU Radio community for their assistance of my research when I needed help with the software usage. This project was partially supported by New Investigator Award funding from NC Space Grant under award No. NNX10A168H.

Table of Contents

List of Figures

List of Tables

List of Acronyms

ADC – Analog Digital Converter

AES – Advanced Encryption Standard

AWGN – Additive White Gaussian Noise

BER – Bit Error Rate

BPSK – Binary Phase Shift Keying

CDMA – Code Division Multiple Access

CLIDE - Cross Link Integrated Development Program

COM DEV- Communications Development

COTS - Commercial-Off-The-Shelf

CPU – Central Processing Unit

CRC – Cyclic Redundancy Check

CSI – Channel Side Information

DAC – Digital Analog Converter

DDC - Digital Down Converter

DICE - Dynamic Ionosphere CubeSat Experiment

DSA - Dynamic Spectrum Access

DS CDMA – Direct Sequence Code Division Multiple Access

DSP – Digital Signal Processor

DSS – Distributed Space System

DUC – Digital Up Converter

FDMA – Frequency Division Multiple Access

FFT – Fast Fourier Transform

FOB – Security token

FPGA – Field Programmable Gate Array

FSK – Frequency Shift Keying

FTP – File Transfer Protocol

GEO – Geostationary Earth Orbit

GNSS – Global Navigation Satellite System

GPS – Global Positioning System

GPSDO – Global Positioning System Disciplined Oscillator

GRC – GNU Radio Companion

GSM – Global System for Mobile

HTTP – Hypertext Transfer Protocol

IF – Intermediate Frequency

ISL – Inter Satellite Link

ISM – Industrial Scientific and Medical

ISS – International Space System

JHU APL – John Hopkins University Applied Physics Laboratory

LEO – Low Earth Orbit

MAC – Media Access Control

MEO – Medium Earth Orbit

MIMO – Multiple Input Multiple Output

MPSK – Minimum Phase Shift Keying

MQAM – M-ary Quadrature Amplitude Modulation

NASA – National Aeronautics and Space Administration

NTP –Network Time Protocol

OFDM – Orthogonal Frequency Division Multiplexing

OSI – Open Systems Interconnection

PLL – Phase Locked Loop

PSK – Phase Shift Keying

RF – Radio Frequency

SCAN - Space Communications and Navigation

SNR – Signal to Noise Ratio

SBX – Transceiver name

SDR – Software Defined Radio

SLV – Small Launch Vehicle

STRS – Space Telecommunication Radio System

TCP/IP – Transmission Control Protocol/Internet Protocol

TDMA – Time Division Multiple Access

TFTP - Trivial File Transfer Protocol

UDP/IP – User Datagram Protocol/Internet Protocol

UHD – USRP Hardware Driver

UHF – Ultra High Frequency

USRP – Universal Software Radio Peripheral

VHDL – VHSIC Hardware Description Language

VHF – Very High Frequency

WiMax – Worldwide Interoperability for Microwave Access

Abstract

The increased usage of CubeSats recently has changed the communication philosophy from long-range point-to-point propagations to a multi-hop network of small orbiting nodes. Separating system tasks into many dispersed satellites can increase system survivability, versatility, configurability, adaptability, and autonomy. Inter-satellite links (ISL) enable the satellites to exchange information and share resources while reducing the traffic load to the ground. Establishment and stability of the ISL are impacted by factors such as the satellite orbit and attitude, antenna configuration, constellation topology, mobility, and link range. Software Defined Radio (SDR) is beginning to be heavily used in small satellite communications for applications such as base stations. A software-defined radio is a software program that does the functionality of a hardware system. The digital signal processing blocks are incorporated into the software giving it more flexibility and modulation. With this, the idea of a remote upgrade from the ground as well as the potential to accommodate new applications and future services without hardware changes is very promising. Realizing this, my idea is to create an inter-satellite link using software defined radio. The advantages of this are higher data rates, modification of operating frequencies, possibility of reaching higher frequency bands for higher throughputs, flexible modulation, demodulation and encoding schemes, and ground modifications. However, there are several challenges in utilizing the software-defined radio to create an inter-satellite link communication for small satellites. In this paper, we designed and implemented a multi-user inter-satellite communication network using SDRs, where Code Division Multiple Access (CDMA) technique is utilized to manage the multiple accesses to shared communication channel among the satellites. This model can be easily reconfigured to support any encoding/decoding, modulation, and other signal processing schemes.

CHAPTER 1

Introduction

In the upcoming years, space observation will have seen a decline in cost and size of satellites. There is growing interest in replacing large satellites by much smaller satellites, since large satellites are costly, hardware consuming, and produced long wait times for launch. The current usages of small satellites are attractive due to reduced build time, more frequent launch opportunities, larger variety of missions, more rapid expansion of the technical and/or scientific knowledge base, and greater involvement of small industries and universities [1].

Large number of small satellites can be deployed in various configurations, which fall into a general class called Distributed Space Systems (DSS). There are different types of formation flying patterns depending on the separation between satellites and intended applications. The three most common types of formations are: trailing or leader-follower, cluster, and constellation [2]. Fractionated spacecraft's and satellite swarms are the new technologies for future space missions, which are also subsets of the DSS. A satellite swarm is defined as sets of agents that are identical and able to self-organize to communicate directly or indirectly and to achieve a mission objective by their collective behavior. The various small satellite configurations including clusters, constellations, and swarms provide affordable solutions to perform scientific and technological missions. As objectives of these missions become more ambitious, we still have to address numerous issues such as supporting multiple signals, and increasing data rates over reliable inter-satellite and ground links to Earth. Also, as the number of CubeSats orbiting the Earth is increasing, there is a shortage of available frequencies leading to further regulatory issues. The dynamic nature of the space environment may lead to failure of nodes or change in network topology and hence the overall architecture should adapt according to the change in system dynamics. Existing communication systems cannot fully address these challenges. One of the possible strategies to solve these issues is by equipping satellites with a Software Defined Radio (SDR) because SDR facilitates various software implementations, which enable an adaptive and reconfigurable communication system. There would need to be an adaptive SDR architecture using a field programmable gate array (FPGA), which could be easily incorporated into a small satellite design. Software Defined Radio (SDR) is beginning to be used in small satellite communications for applications such as base stations. Software Defined Radio is simply a system that implements all its baseband functionalities in software instead of having physical hardware components.

In space communications there are times when on board components may fail or require some modification for the receiver on the ground station. If such a receiver has been implemented in hardware, the cost of replacing or upgrading will be costly. However, if the system has been implemented in an SDR platform, simply updating the software to the needed parameters produces re-configurability. Software Defined Radio is a key area to realize various software implementations that enable an adaptive and reconfigurable communication system without changing any hardware device or features [1]. As a whole SDR minimizes hardware to provide faster modifications, prototyping, lower costs, greater flexibility, and reusability.

Another key application of SDRs is its adaptability. An ISL equipped with SDR can enable changes in protocols for different frequency bands, such as Ka band or X band, for high data rate applications and higher bandwidth. With SDRs, a simple modification to the code as well as the Field Programmable Gate Array (FPGA) board can reflect this. FPGAs consist primarily of reconfigurable logic elements; some even support a dynamic reconfiguration in which one could substitute components between one another as needed without reprogramming

[27]. Therefore with FPGA-based SDR, performance of modulation swap schemes in real time as well as saving time and money is achievable. The firmware on the FPGA can also be remotely updated for installation of new blocks and removal of unused blocks. This will allow the system to support new protocols while the small satellite is in orbit. National Aeronautics and Space Administration's (NASA) Cross Link Integrated Development program (CLIDE) has developed technologies to facilitate new satellite-to-satellite communications for mesh connectivity and ad hoc networking [15]. The Space Communications and Navigation (SCaN) Test-bed is an experimental communications system using SDRs and provides the capability for S Band, Ka-Band, and L-Band communication with space and ground assets. Launched in July 2012 to operate on an external truss on the International Space System (ISS) [15], the SCaN Test-bed provides experimenters an opportunity to develop applications for communication and to advance the understanding of operating SDRs in space.

In this study, we propose a novel inter-satellite communication model for a network of small satellites based on implementing a Direct Sequence Code Division Multiple Access (DS-CDMA) protocol in GNU Radio. Code Division Multiple Access is a spread spectrum technique that can enable privacy due to use of separate codes for each user, reduce multi-path effects and is immune to interference effects [15]. The performance of this model has been evaluated and compared for different modulations (BPSK and QPSK), channel coding techniques (un-coded and convolutional coding), and modeled for different communication channels based on Additive White Gaussian Noise (AWGN), Rayleigh, and Rician fading noise types. A single Universal Software Radio Peripheral (USRP) serves as the mother-ship node while the other USRPs serve as the daughter-ship node. Each USRP will also be connected to a laptop using a gigabit Ethernet cable. We also assume that the system is broadcasting to all the members in the cluster. In this work, we used an open source hardware and software to make it adaptable for other researchers to utilize this model in their projects. This model can be easily reconfigured to support any encoding/decoding, modulation, and other signal processing schemes.

1.1 Motivation

The motivation for this project is to study more about what Software Defined Radio (SDR) has to offer in small satellites. SDR is a system that implements all of their baseband functionalities in software. This makes the SDR able to overcome hardware constraints imposed by standard specific hardware [23]. Table 1 shows the comparison between the traditional radios and the software defined radios. Compared to traditional radios, software defined radio is a better choice due to its versatility and inexpensive products allowing researchers and educational institutions to better understand wireless communication.

Table 1

Comparison of Traditional Radios and Software Defined Radios

Traditional Radios	Software Defined Radios
Pros: • Limited processing and thus selection of processor/controller/ADC is less critical • Cheap and readily available	Pros: • Flexible Design: Multi-band/Multi-Mode • Software based reconfigurable platform • Upgradable during mission lifetime

Table 1

Cont.

Cons:	Cons:
• Fixed Design: Single Band/Single-Mode • Complexity in hardware • More analogous components • Cross talk between the narrow bands due to aging	• Complexity in software • Vulnerable to software threats • Faster FPGAs and DSPs and larger bandwidth ADCs are required • Power Consumption

The popularity of SDRs and the numerous ways they can be used for is very attractive. From building Global System for Mobile (GSM) networks, tracking ships and planes, military, amateur and home use, intercepting signals such as a car security token (FOB) or restaurant pager [50]. All these things have been accomplished using software-defined radio and its usage for other applications such as small satellites has fascinated researchers and companies.

The inter-satellite links in Low Earth Orbit (LEO) are another motivation behind this project. LEO's have advantages over Geostationary Earth Orbit (GEO) and Medium Earth Orbit (MEO) [51] for small satellites and amateur users. Table 2 shows the three orbit altitudes and a comparison of their parameters. The effect of having low propagation delays and power requirements thus having low orbit altitudes for low-cost and light weight satellite is seen as attractive. As well as the ability to cover land, sea, and air based users; low Earth Orbit system has increased signal strength due to its proximity to the earth's surface. LEO is advantageous in sending small satellites into orbit, running the experiments, and returning them in a relatively short time as well as ISL's in LEO utilizing Radio Frequency (RF) or laser/optical transmission technologies.

Table 2

Orbit Parameters Comparison

Specifications	Low Earth Orbit	Medium Earth Orbit	Geostationary Earth Orbit
Height (km)	180 – 1,200	2,000-35,876	36,100-46,000
Path Loss (dB)	5-10	64-100	230-280
Spot beam handover (min)	5-10 mins	3 hours	None
Ground velocity (km/s)	1-2	2-4	None
Satellite handover time	7.5-7.09	2.5-2	0
Propagation delay (ms)/ Ground to Ground	150-164	180-184	~190-200
Doppler (kHz)	~40	~30	0

1.2 Project Objectives

The purpose is to design and implement an optimal inter-satellite communications using Software Defined Radio for a distributed wireless sensor network of small satellites. Using an SDR for our inter-satellite link will help enable higher data rates, re-configurability for hardware failure or changes, remote upgrades for future frequency and allocation changes, cognitive and adaptive operation, and multi-mode operation. The question of how we can model an Inter-

Satellite Link for a distributed Wireless Sensor Network by utilizing Software Defined Radio has been designed and implemented. The optimization of the ISL has been achieved by designing a DS-CDMA communication using SDR. Figure 1 shows a diagram of the Galileo Atmospheric Data Enhancement Mission, GADEM, used for measuring atmospheric data to improve Galileo navigation data by using two K-band frequencies [52]. The three satellites are utilizing a K-band inter-satellite link that would be used in different scenarios (Galileo-to-Galileo link, Galileo-to-LEO link, and Galileo to ground station link, etc.)

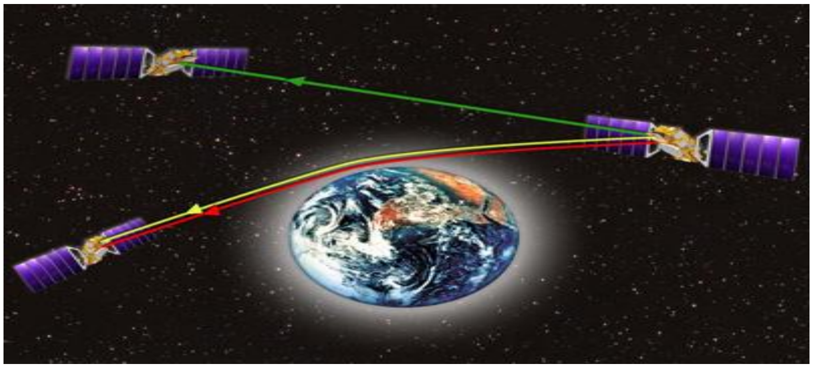


Figure 1. Galileo atmospheric data enhancement mission inter-satellite links [52].

1.3 Scope and Significance of Research

Software-defined radio is attractive to small satellites because of its various software implementations, which can enable an adaptive and reconfigurable communication system without modifications to the hardware device itself. Inter-satellite links that are software-defined will provide several opportunities such as high-speed data cross-links and simple frequency upgrades.

However, several problems are faced when trying to integrate the software-defined radio with small satellites such as which multiple access protocol will be the best for the inter-satellite communication, the type of software defined radio architecture, antenna issues, payloads, interference, wireless standards types, inter-satellite link frequency, and the topology. Current challenges included constraints limiting the application of small satellites both at the transmitting and receiving end, for example, limited power, mass, antenna size, intermittent communication link, etc. The main challenge therefore lies with power consumption. With enough power available, the small satellite can support high frequency inter-satellite links ranging from 30 MHz (Very High Frequency) to 40 GHz (Ka band) for transmission and reception [1]. Table 3 shows the frequency ranges that are appropriate for inter-satellite communications. Factors such as mission requirements, hardware constraints, and technical characteristics play an important role in choice of frequency bands. Table 4 shows some of the tradeoffs for the maximum data rate of different frequency allocations.

Table 3

Frequency Allocation Candidates for Inter-Satellite Communications [38]

Band	Frequency Range	Service	Examples
S	2025-2110 MHz 2200-2290 MHz	SRS SRS	PRIMSA TPF
Ku	13.75-14.3 GHz 14.5-15.35 GHz	SRS SRS	
Ka	22.55-23.55 GHz 25.25-27.5 GHz 32.3-33.4 GHz	ISS ISS ISS, RNSS	Iridium GRACE (K/Ka band) StarLight
W	59-64 GHz 65-71 GHz	ISS ISS	

Table 4

Bandwidth and Data Rate Equivalences [38]

Bandwidth	Maximum Data Rate	Recommended Frequency Allocation
Narrow	<100 kbps	S
Medium	100 kbps – 1 Mbps 1 Mbps – 10 Mbps	S, Ku Ku, Ka
Wide	10 Mbps – 100 Mbps >100 Mbps	Ku, Ka Ku, Ka, W

The significance of this research is to allow us to see how software defined radio can be of usage to small satellites with its endless possibilities. I have viewed other small satellite missions and current SDR small satellite implementations to determine if this is appropriate for the small satellites. To best of my knowledge, this research is the first work to study and implement the utilization of SDRs for inter-satellite communications. Therefore, the results of this research can open up a path to utilize SDRs in similar projects and other researchers can adapt this model to their needs, and create derivative solutions for them. Testing the software defined radio with the small satellites to prove its additive benefits compared to not equipping it with one will be a great achievement to the government and space community. Software defined radios have been around for over 20 years now but have only recently been seen in an increased usage by not only the government and industry, but also by hobbyist. This being said, the significance of this research is to see how we can have small satellites perform and communicate in space with minimal human intervention as possible. As well as see the potential to accommodate new applications and services without hardware changes. Furthermore, existing radios can soon be replaced by SDRs to eliminate the need for costly hardware replacements as the spectrum bandwidth is continually reassigned.

1.4 OSI Model Overview

A communication system is complex due to the many function it employs; therefore a reference model is used for better explanation. The Open System Interconnection (OSI) reference model divides a network system into different layers with different tasks. It simply serves as a reference for the communication between different computer systems connected in a network [2]. The OSI model will be used as a framework for the inter-satellite communication for small satellite systems. For an optimum communication, several communication components in the 7 layers of OSI model needs to be considered simultaneously.

Figure 2 serves as the OSI model containing the seven layers each with well defined functions. The Physical Layer and Data Link Layer serve as the focus of our work. In the proposed SDR test-bed, the physical layer contains the USRP N210, which provides the transmission of raw bits over the antenna. The physical layer, related to the trade-offs among several system drivers such as power, frequency, data rate, and bandwidth, has been found to be of primary concern, since it has the largest role to play when it comes to reliable and efficient communication via ISL [38].

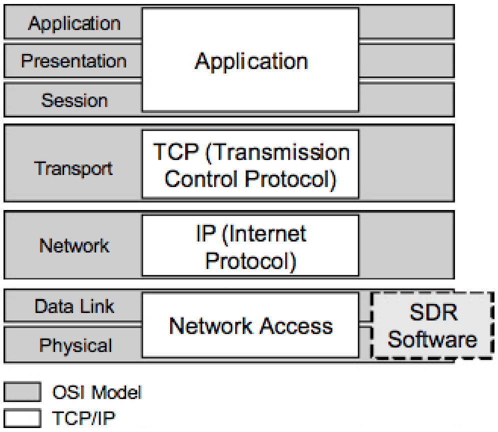


Figure 2. OSI Model of the software architecture layers [14].

Determining the optimum modulation for the inter-satellite communication systems play an important role. The type of modulation must be determined according to the mission and system parameters as well as whether the transmission type is based on a coherent or non-coherent system. Coherent systems like BPSK offer advantages in requiring less amount of power to support a given data throughout and error rate, whereas non coherent systems like Frequency Shift Keying (FSK) have poor performance in received power to support the same data throughput and error rate as a coherent system [3]. Because of power consumption being the main constraint in satellite communication systems, coherent systems are often seen as the most desirable. The disadvantage in the coherent system however is the need for a local reference frequency. For instance, in the receiver a Phase-Locked Loop (PLL) tracks the frequency and phase of the received signal. During this time a loss of communication occurs from the loop attempting to acquire and lock to the incoming signal. Non-coherent systems however do not experience this due to a net loss in data throughput because the link is not available 100% of the time [3]. Coherent systems also have several advantages mainly related to increased receiver sensitivity; compatibility with complex modulation formats such as Minimum Phase Shift

Keying (MPSK), M-ary Quadrature Amplitude Modulation (MQAM), etc., and better bit error rate detection [56].

In this study we assume a coherent system with transmission of bits modulated using BPSK and QPSK as well as exhibiting un-coded and convolutional coding techniques. The carrier frequency utilized is 2.4 GHz with AWGN, Rayleigh, and Rician Fading channel models in the case of Channel Side Information (CSI) being known at the receiver. The data link layer serves as providing reliable data transfer across the physical link and this link is also responsible for the Medium Access Control, commonly referred to as the Media Access Control (MAC) layer. The MAC layer ensures error free data transmission and provides channel-accessing schemes for several nodes within a multiple access network using a shared medium [2]. The multiple access technology types, Code Division Multiple Access (CDMA), Time Division Multiple Access (TDMA), Frequency Division Multiple Access (FDMA), for designing an optimum inter-satellite link should be considered deeply due to its effect on bandwidth in several ways such as in the frequency allocation part mentioned in section 1.3. In this study the selection of CDMA is chosen due to factors such as multiple simultaneous transmission of signals, improved ranging accuracy from Global Navigation Satellite System (GNSS) technology and insensitivity to other satellites joining in and out the system. Further advantages and limitations including the near far problem and multiple access interference will be discussed more in section 3.2.

1.5 Summary of Conceptual Framework and Methodology

The conceptual framework for the project is as follows:

- Determine the target design parameters (ex: number of clusters)

- Establish connection between Universal Software Radio Peripheral (USRP), GNU Radio, and PC’s.
- Integrate the CDMA multiple access protocol in GNU Radio to serve as the inter-satellite communication.
- Test different channel models (AWGN, Rayleigh, Rician) and modulation types Binary Phase Shift Keying (BPSK) and Quadrature Phase Shift Keying (QPSK) with signal spread by Walsh Hadamard codes.
- Perform the tests using the Radio Frequency Network Channel Emulation Simulation Tool.

During this project, I designed and implemented a cross-layer optimum inter-satellite communications model. A novel inter-satellite communication model for a network of small satellites based on implementing a Direct Sequence Code Division Multiple Access (DS-CDMA) protocol in GNU Radio has been developed. BPSK and QPSK modulations have been tested on the transmitted signal. Simulation was tested on channel models including AWGN, Rayleigh, and Rician. The DS-CDMA model follows a synchronous system therefore Walsh Hadamard codes is employed for our system. The model uses 8 chips per symbol with two orthogonal codes for training and data as the spreading code.

1.6 Thesis Outline

The thesis consists of five chapters. The thesis begins with an introduction to inter-satellite communications in small satellite networks, current challenges and the potential benefits of utilizing SDRs in inter-satellite communications. The motivation of the project, project objectives, as well as the research question has all been presented in the introduction, which is in Chapter 1. In Chapter 2, I explore the previous scholarly work on small satellites and software-defined radio. I also address the following issues: significant conceptual frameworks in the field, background information on research methodologies and areas of inquiry that have been treated thoroughly. The methodology is discussed in Chapter 3, which addresses the following areas: Rational for Methodology, Conceptual Approach, Data Gathering Methods, and Validity and Limitations of Data. In Chapter 4, I present the results from the data acquired through my research methods and provide analysis of the significant findings and observations. Finally in Chapter 5, I present a discussion of the overall research, explore the significance of the results, and discuss any future research.

CHAPTER 2

Literature Review

This chapter provides a background to the technologies utilized in this thesis, including literature review on small satellites, the missions and constellations, an introduction on SDR and its capabilities, applications in cognitive radio network, and SDRs in small satellites. The inter-satellite communication channel is also discussed in terms of noise, fading, and Doppler. The reasoning in choosing GNU Radio due to its low cost, re-configurability, it being open source software and its performance to work alongside the RF hardware is discussed. The small satellites, such as CubeSats, could be ideal for an SDR system and combined with their inter-satellite links could possibly increase data rates [33]. Due to the movement of LEO satellites along orbits, communication devices cannot be optimized for static geometric parameters only; otherwise the performance of ISL communication between satellites will degrade with time [53]. The constant changing of position of the LEO satellite and effects of noise and fading can cause the geometric parameters of the ISL to vary significantly.

2.1 Significant Conceptual Framework

The usages of software defined radios are being used to reduce nonrecurring development costs and time-to-market for new consumer product derivatives. For small satellite communications the SDR is being used for both the space and ground links. Software Defined Radio (SDR) in small satellites offers the opportunity for cognitive and adaptive operation, multi-mode operation, radio reconfiguration, remote upgrade as well as the potential to accommodate new applications and services without hardware changes. Furthermore, existing radios can soon be replaced by SDRs to eliminate the need for costly hardware replacements as the spectrum bandwidth is continually reassigned. Together with small satellites this approach is helpful because there is a scope of developing a system, which is compatible with more than one mobile communication standard. Using reconfigurable hardware and swapping the software for different technologies can accomplish this.

2.2 Current Implementations of SDRs in Small Satellites

Currently several Software Defined Radios are being optimized in small satellites. Communications Development (COM DEV) is showing its usage currently as it is being applied to the payloads of the Canadian M3MSat [33]. The SDR is capable of concurrently processing sampled received Intermediate Frequency (IF) signals, and generating and transmitting suitably modulated IF signals, to and from the RF/IF front end. The Central Processing Unit (CPU) runs RTEMS (Real-Time Executive for Multiprocessor Systems) V4.8 RTOS (Real Time Operating System). A BSP (Board Support Package) is provided with the SDR board to enable COM DEV to develop their own specific algorithms running on the CPU and Digital Signal Processor (DSP). There is also a significant level of research ongoing into the development of technologies that are enabled by SDR in small satellites, such as cognitive radio and dynamic spectrum access (DSA) and cross-layer techniques [13].

Current Global Positioning System (GPS) technologies for satellite navigation are relatively costly, heavy, and use a high amount of power, thus making small satellites to be at a disadvantage. However in [17], proposed is a low cost, low weight, low power GPS navigation system to support smaller satellites. The TIDGET based receiver design takes brief snapshots of GPS data from three different views on the satellite and downlinks the data to the ground station SDR for processing.

Cadet Nanosat Radio by L3 communications was developed for Dynamic Ionosphere CubeSat Experiment (DICE) as a half-duplex Ultra High Frequency (UHF) radio system to provide high data rate communications system for a CubeSat, by making using of SDR technology [14].

Altera Corporation is one of the pioneers of field programmable logic solutions. The high-performance DSP blocks, embedded Nios II processors, and logic elements of Altera's Stratix series FPGAs make them ideal for adaptive beam forming applications [19]. Because Altera FPGAs are remotely upgradeable, they reduce the risk involved with designing for evolving industry standards while providing the option for the gradual deployment of additional transmit diversity schemes. For security, the FPGA configuration bit stream is transferred to the FPGA in Advanced Encryption Standard (AES) encrypted form, with the FPGA containing a crypto key and a decryption module. Other FPGAs in [27], the Xilinx Spartan3A-1400, is used instead of the default FPGA in the USRP for more configurable logic elements. The usage of VHSIC Hardware Description Language (VHDL) for the firmware instead of Verilog makes it attractive due to its high usage in government and other public sector projects and its additional layer of security.

Vulcan Wireless Inc. has developed two SDRs optimized for usage in satellites. The first being the CubeSat SDR, which provides access to a wide variety of communication protocols and a data rate of up to 10 Mbps at S-Band [19]. The second, MicroBlackbox Transponder, offers fewer protocols and a lower data rate [19]. These two systems support numerous S-Band frequencies (2-4 GHz) and work with a variety of communication protocols and encryption schemes [34]. Another SDR used for the CubeSat, which is also FPGA based, is the Firehose Adaptive Software Defined Radio, designed by Adaptive Radio Technologies LLC [39]. This system features in flight programming, and programmable transmit and receive paths. However, just like the Vulcan Wireless systems, this SDR does not support the Space Plug-and-Play

Avionics (SPA) protocol for plug and play operation and does not use open source hardware or software. The SPA was developed to introduce the concept of plug and play components to small form-factor satellites, which would be advantageous for quick swapping in our small satellites since a majority of CubeSats do not contain interchangeable parts [27].

A low power, low mass, modular, multi-band SDR has been developed by John Hopkins University Applied Physics Laboratory (JHU/APL), for NASA, for use in space borne communications, navigation, radio science, and sensor applications, with terrestrial and airborne applications as well [44]. The SDR offers an S- or X-Band receiver and up to 1.3 Mbps uplink data rate. The reprogrammable SDR has flexible modulation and encoding schemes as well as flexible transmit and receive frequencies and turnaround ratio. The SDR is capable of transmit data rates up to 100 Mbps with Phase Shift Keying (PSK) and Quadrature Amplitude Modulation (QAM) modulation types. Its compatibility with NASA's Space Telecommunication Radio System (STRS) architecture helps promote the wide use of this transceiver throughout NASA community [44].

In regards to the implementation of SDR's for ISL, Tethers Unlimited SWIFT-RelNav [33] is a SDR RF-based system that provides relative range and attitude determination capabilities as well as inter-satellite communications. The SDR RelNav provides range sensing between satellites to better than 10 cm accuracy, inter-satellite crosslink data rates at ≥ 12 Mbps, Bit Error Rates of $<10^{-6}$, and timing/frequency synchronization to better than 1 ns, 0.1ppb. The SDR application in the ISL enables this system to perform ISL communication up to 10 km in range [33]. This SDR RF-based system proves to enable high data rates in satellites as well as operate in Ku and X bands. High data rate communications could revolutionize space and science exploration.

2.3 Software Defined Radio

The term "software defined radio" was coined in 1995 by Stephen Blust, who published a request for information from Bell South Wireless at the first meeting of the Modular Multifunction Information Transfer Systems (MMITS) forum in 1996, organized by the USAF and DARPA around the commercialization of their SPEAKeasy II program [46]. Joseph Mitola, who is known for coining the term cognitive radio and software radio, objected to Blust's term, but finally accepted it as a pragmatic pathway towards the ideal software radio [45].

Traditional radios consist entirely of specialized hardware limiting cross-functionality and modification through physical intervention. Any type of modification would cause the entire system to be redesigned and implemented in hardware, which is costly and time consuming. Software Defined Radio follows a software-based approach, which removes the drawbacks of conventional radio and allows more advantages of the software. As shown in Figure 3, in this system, nearly all the baseband signal processing on both the transmission and receiving end is performed in the software domain. The idea is to move the digital domain (demodulation, modulation, decoding, etc.) as close as possible to the antenna in order to achieve the ideal SDR scenario. In summary, the ultimate software defined radio would be lightweight, consume very little power, requires no external antenna [21], accepts fully programmable traffic, support broad range of frequencies and configure itself.

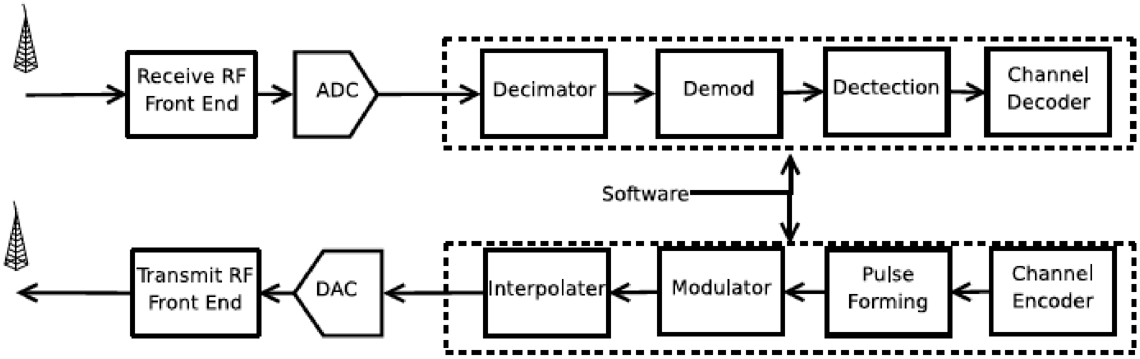


Figure 3. Block diagram of typical SDR system with digital domain in software while analog in hardware [17].

In a Software Defined Radio system, the Analog-to-Digital (ADC) and Digital-to-Analog converter (DAC) converts signals to and from radio frequency front-end. The RF front end is used to down convert the signal to the lower frequency called an Intermediate Frequency (IF). The ADC will digitize signals and pass it to the baseband processor for further processes such as demodulation, channel coding, source coding and etc.

The front end of the SDR is the RF section, which has the role of transmitting and receiving signals at the specific working frequency band of the SDR model. At the receive path RF section, the amplification of the signal is performed and mixed with the local oscillator to down convert it the Intermediate Frequency (IF). In the transmit path the RF section then receives the signal from the IF section and converts it to the required frequency. After conversion of the signal to the required frequency, matching circuitry is performed for the maximum power transfer and obtained to the antenna for transmission of the signal. The IF section of the receive side is where the ADC is performed and the Digital Down Converter (DDC) performs the signal demodulation and processing before it passes to the baseband processing. Similarly at the transmission side it performs the DAC and the Digital Up Converter (DUC), which executes the modulation on the carrier and converts the digital to analog signal for radio transmission.

The first U.S. Military software defined radio system, SpeakEasy, was developed between 1992 and 1995 and it required powerful digital signal processors (DSP) and Field Programmable Gate Arrays (FPGAs) together. At the time these radio systems were so huge and bulky that they could fit in the back of a truck [24]. Today advances in DSP and FPGA technology have enabled engineers and researchers to develop much smaller SDR platforms. With the addition of increased clock speeds for faster real time processing of data, more complex coding and modulation schemes are done by SDRs.

Research in the applications of SDRs in spectrum sensing and cognitive radio has increased in the past few years. In [47] the authors describe spectrum sensing detecting the availability of the radio frequency spectrum in a real-time fashion, which is essential and vital to cognitive radio. A small form factor software SDR development platform has been employed to implement a spectrum-sensing receiver using the Fast Fourier Transform (FFT) averaging ratio algorithm for spectrum sensing. The authors were able to successfully demonstrate real time spectrum sensing on a hardware platform with controllable primary users. The results were conducted and proved that the FFT averaging ratio algorithm using SDR is effective.

Authors in [48] also use the SDR to study spectrum sensing by employing an energy detection sensing using GNU Radio and the USRP. They present the viability of using energy detections algorithm on GNU Radio –USRP to sense and detect the spectrum frequency. The algorithm is based on magnitude squared of the FFT output, which is optimal for detecting any unknown zero mean constellation signals. The radio frequency energy in the channel is then measured to determine whether the channel is occupied or not. The results show the energy detection algorithm using GNU Radio and the USRP is able to sense and detect the spectrum frequency range.

2.4 Universal Software Radio Peripheral

The Universal Software Radio Peripheral, also known as USRP, is a flexible open source and low cost platform for SDR developed by Virginia Tech graduate, Matt Ettus and his development team at Ettus Research [32]. The USRP is simply used to create a connection between the RF-world and the PC. Figure 7 describes the USRP in this study consisting of a motherboard with Xilinx Spartan 3A-DSP 3400 FPGA, 2 pairs of DAC's and ADC's, digital down converters and up converters with programmable interpolation rates, and a daughterboard functioning as a RF front-end. The USRP interfaces with the computer using a Gigabit Ethernet. The motherboard model we will be using is the USRP N210, which is where all the circuitry is integrated. Figure 5 shows the N210 model that comes with a Multiple Input Multiple Output (MIMO) port, which can be used to connect multiple USRP systems, and optional utilization of an external reference clocking option for precise synchronization is supported. The Ethernet interface can sustain simultaneously transmitting of up to 50 MHz of bandwidth in and out of the radio. The product architecture includes a 100 MS/s dual ADC sample rate, 400 MS/s dual DAC sample rate, and the USRP N210 can stream up to 50 MS/s to and from host applications.

If the N210 has a fast Ethernet controller (100 Mbps), care should be taken while setting the data rate between the USRP and computer because the maximum data rate is restricted due to the fast Ethernet controller causing the data rate mentioned above to not be achievable. If the computer is equipped with a fast Ethernet controller then the maximum data rate will be lowered by ten times compared to a gigabit Ethernet. The maximum effective data rate will be 95.46 Mbps, which gives you a maximum sample rate of 2.98 MSPS and a minimum sample rate of

195.3125 KSPS. An optional Global Positioning System Disciplined Oscillator (GPSDO) module can also be used to discipline the USRP N210 reference clock to within 0.01 ppm of the worldwide GPS standard [5]. The chosen RF front end, which is called a daughter board, for this research was the SBX daughterboard. Figure 4 shows the SBX daughterboard, which is a wide bandwidth transceiver that provides up to 100 mW of output power, and a typical noise figure of 5 dB. The local oscillators for receive and transmit chains operate independently, which allows dual-band operation. The SBX is MIMO capable, provides 40 MHz of bandwidth and is ideal for applications requiring access to a variety of bands in the 400 MHz-4400 MHz range. Example application areas include Wi-Fi, Worldwide Interoperability for Microwave Access (WiMax), S-band transceivers and 2.4 GHz Industrial Scientific and Medical (ISM) band transceivers [5]. Figure 6 shows the antenna used in the research, which is a VERT2450 Dual Band (2.4 to 2.48 GHz and 4.9 to 5.9 GHz) omni-directional vertical antenna, at 3dBi Gain.

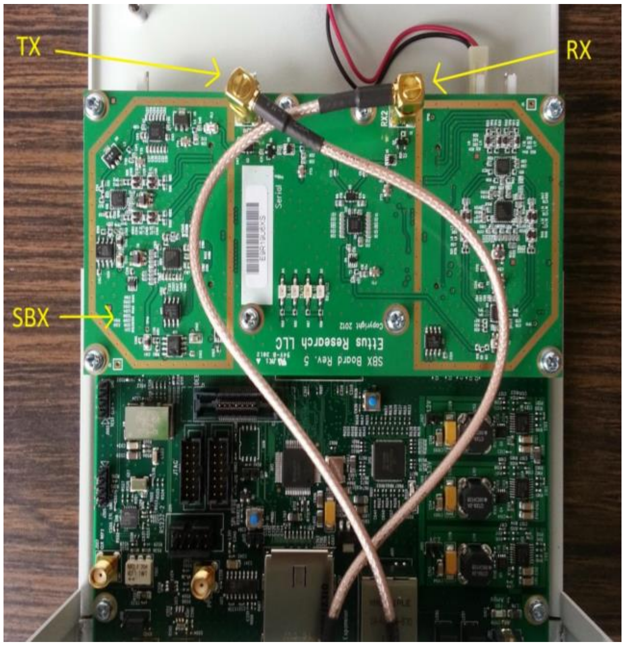


Figure 4. SBX USRP daughterboard (400 MHz-4.4 GHz).

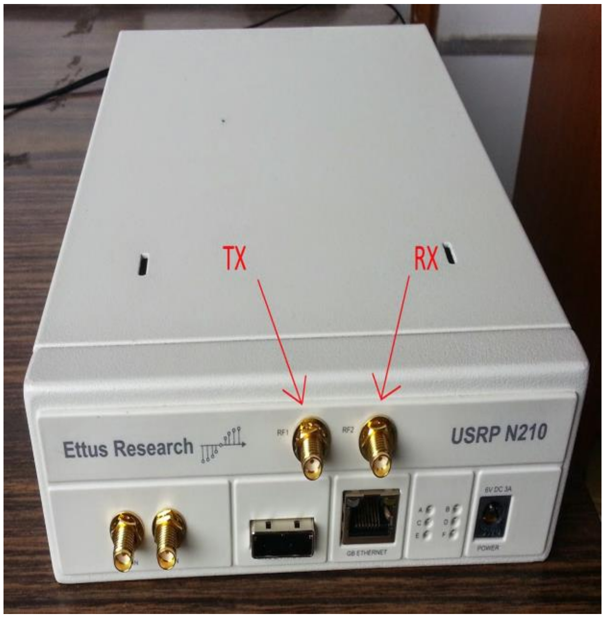


Figure 5. USRP N210 model.

Figure 6. VERT2450 vertical antenna (2.4-2.5 and 4.9-5.9 GHz) dual band.

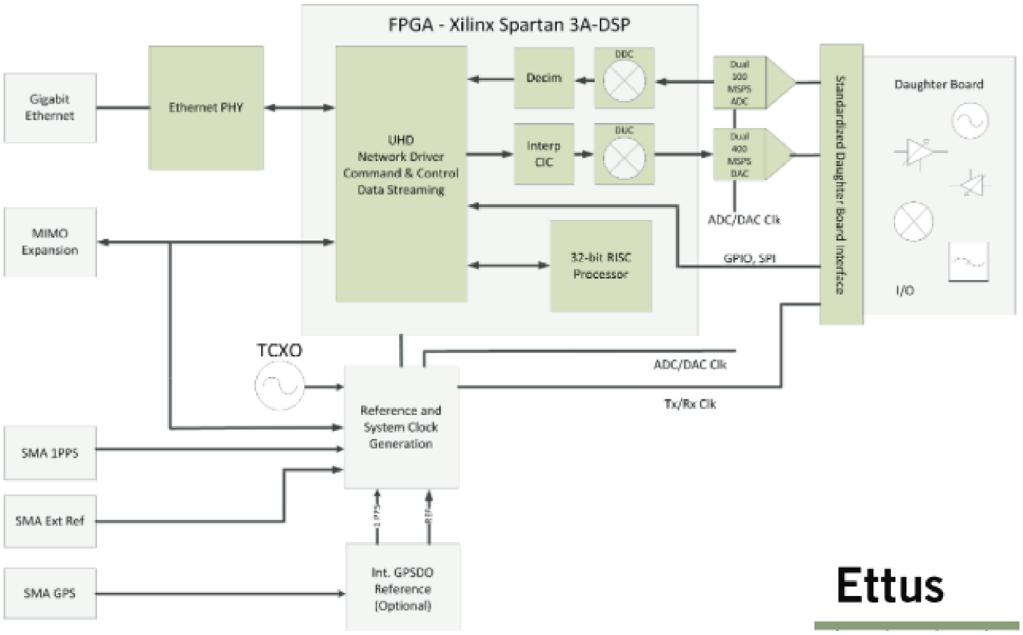


Figure 7. Xilinx Spartan 3A DSP FPGA diagram [41].

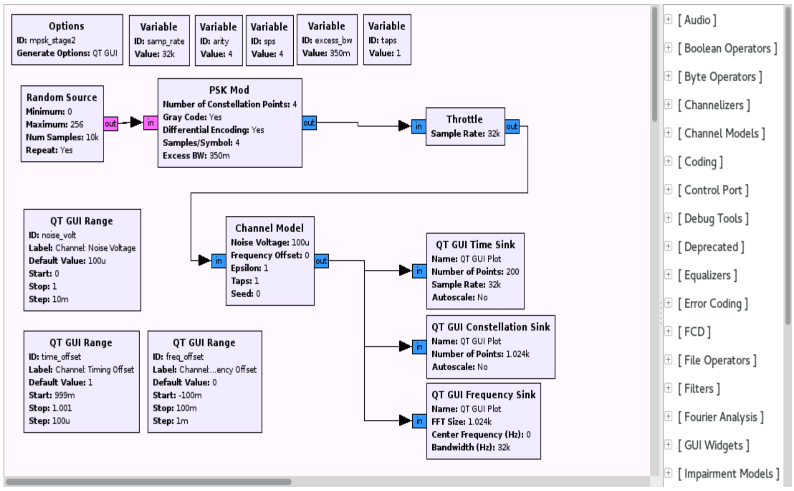
2.5 GNU Radio

GNU Radio is used as the simulation tool to better understand the working of the existing/generated filters, channels codes, equalizers, demodulators, and other processing blocks using pre-recorded or generated data. Each block has a predefined number of input/output interfaces and performs one or more communication functions in the software domain. The blocks can also be modified, upgraded, or even implemented independently, without interfering with the entire communication chain.

The wide usage of GNU radio in several research studies, including NASA, as well as it being open source prompted my interest in using it. The GNU radio architecture [20] is an open-source initiative, where the signal processing is carried out on GPP computers. GNU radio is adapted to the Universal Serial Radio Peripheral (USRP), which converts between base band and RF signals. The signal processing blocks are written in C++ and the graphs are connected using the Python programming language, while the GUI runs on Linux machines. The Simplified Wrapper and Interface Generator (SWIG) tool is used to create the interface between C++ and Python. In the case of the Python interpreter running a flow graph, it calls the corresponding objects and functions from the C++ shared libraries through the SWIG tool.

The recommended operating system for building GNU Radio is Linux, but can also be built on Microsoft Windows using one of the Linux-like environments such as Cygwin or MinGW/MSYS, as well as on MAC OS and NetBSD.

GNU Radio also comes with a handy GUI called The GNU Radio Companion (GRC), which is used to develop GNU Radio applications. Josh Blum, during his studies at John Hopkins University in 2006-2007, developed the GRC in an attempt to make GNU Radio easier to use. In 2009 he distributed the GRC as a free software for the October 2009 Hackfest. After that, the GRC was officially bundled with the GNU Radio software distribution in the 3.2.0 release [20]. With the GRC, users can graphically see the premade blocks they are trying to use, adjust their variables with sliders and text fields, as well as connect them together using the onscreen wires.



Options
ID: mpsk_stage2
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 32k

Variable
ID: arity
Value: 4

Variable
ID: sps
Value: 4

Variable
ID: excess_bw
Value: 350m

Variable
ID: taps
Value: 1

Random Source
Minimum: 0
Maximum: 256
Num Samples: 10k
Repeat: Yes

PSK Mod
Number of Constellation Points: 4
Gray Code: Yes
Differential Encoding: Yes
Samples/Symbol: 4
Excess BW: 350m

Throttle
Sample Rate: 32k

QT GUI Range
ID: noise_volt
Label: Channel: Noise Voltage
Default Value: 100u
Start: 0
Stop: 1
Step: 10m

Channel Model
Noise Voltage: 100u
Frequency Offset: 0
Epsilon: 1
Taps: 1
Seed: 0

QT GUI Time Sink
Name: QT GUI Plot
Number of Points: 200
Sample Rate: 32k
Autoscale: No

QT GUI Constellation Sink
Name: QT GUI Plot
Number of Points: 1.024k
Autoscale: No

QT GUI Range
ID: time_offset
Label: Channel: Timing Offset
Default Value: 1
Start: 999m
Stop: 1.001
Step: 100u

QT GUI Range
ID: freq_offset
Label: Channel:...ency Offset
Default Value: 0
Start: -100m
Stop: 100m
Step: 1m

QT GUI Frequency Sink
Name: QT GUI Plot
FFT Size: 1.024k
Center Frequency (Hz): 0
Bandwidth (Hz): 32k

Figure 8. PSK Modulation flow graph example in GRC.

The flow graph shown in Figure 8 is stored as an XML file. When ran by the user, GNU Radio parses the XML file and uses it to create Python code that performs the signal processing operations described by the blocks in the flow graph. In this flow graph the simulation is focused on receiving digitally modulated signals to better understand the many issues involved with what goes on when transmitting and receiving real signals through real hardware and channel effects. The first stage describes the transmitting of the signal in which a stream of bits is modulated onto a complex constellation by using the PSK Mod block. The second stage describes the addition of a channel model to the signal, in this case AWGN. With the addition of the sink and range blocks, the user is able to play with the concepts of additive noise, frequency offset, and timing offset. The sliders also allow the user to adjust the settings.

Overall, GNU Radio is showing its influence in the software defined radio community. The advantages of GNU Radio can be as follows [23]:

- As a hybrid Python/C++ system, the primitive signal processing blocks are written in C++, which is easy to optimize. Using Python, all the graph constructions, non-performance critical operations can be performed. The underlying runtime system is easy to manipulate.
- The program can be reconfigured even during the runtime by changing the parameters of signal processing blocks.
- The GNU Radio also offers Graphical User Interface (GUI) which makes it visible of the data stream in either time or frequency domain.

2.6 Computational Capabilities for the FPGA Board

The FPGA boards play an important role in the software-defined radio seeing as this is where all the re-configurable logic elements, filtering, and digital signal processing is being done. When integrated with small satellites the FPGA must have certain computational capabilities and required power. Electrical, temperature as well as mechanical and radiation performance need to be considered prior to making a choice of FPGA for small satellites. Based on the data obtained [8], a combinational use of SRAM-FPGAs (multi-chip redundant) and Flash-FPGAs (voting element) for mitigating radiation effects was conceptualized. The analysis illustrates that a multi-redundant system based on SRAM-FPGAs together with a Flash-FPGA based voter provides a sufficiently high reliability for Low Earth Orbit (LEO) missions against radiation effects.

SRAM FPGAs are reprogrammable in the field, offering designers the ability to reconfigure the small satellite without retrieving it. The FPGA should be able to perform on board data processing, while simultaneously reducing the satellites data processing energy consumption. However the limited power budget in the CubeSat and other small satellites makes this integration of FPGA's a challenging task. Typically, the 1U, 2U, and 3U CubeSats maximum power budgets range from 1 to 2.5 Watts, 2 to 5 Watts, and 7 to 20 Watts of power dissipation, respectively [10]. The present FPGA board used in the USRP N210 is designed around the Xilinx Spartan-3A DSP 3400 FPGA. It utilizes a 256 MB DDR2 SDRAM, 256 Mbits flash, 9 Mbits ZBT SRAM, 32 Mbits Platform Flash, 16 Mbits EEPROM, 10/100/100 Ethernet, RS-232 Serial Port, -40 to 100 °C operating temperature, footprint area of 19 x 19 mm, and power consumption ranging from 1 to 2.5 Watts.

2.7 SDR Re-configurability

In space communications there are times when on board components may fail or require some modification for the receiver on the ground station. If such a receiver has been implemented in hardware, the cost of replacing or upgrading this component will be costly. For components in space already that may require modifications or an upgrade, it will be impossible to implement this while already up in space unless you bring it back to earth. However, if your system has been implemented in an SDR platform, the re-configurability is a simple upgrade. Software Defined Radio (SDR) is a key area to realize various software implementations that enable an adaptive and reconfigurable communication system without changing any hardware device or features [1]. Re-configurability is the main desire for the software defined radio system. For instance, the same SDR receiver can be used to receive the data from different space missions if the communication specifications for that specific target mission are given. The same can be done with an SDR transceiver and all we would need to do is simply update the software code.

To implement ISL link to utilize a different frequency band such as Ka or X band for high data rate applications and gain high bandwidth, we could simply change some code as well as the FPGA board to reflect this. FPGAs consisting primarily of reconfigurable logic elements, some even support a dynamic reconfiguration in which we could swap components as needed without reprogramming [27]. Therefore with the FPGA based SDR we could perform modulation swap schemes in real time, saving time and money. Using this FPGA Technology we can design our small satellite to support multiple digital signal processing blocks (such as modulation or source coding) and swap them as needed. The firmware on the FPGA can also be remotely updated for installation of new blocks and removal of unused blocks. This will allow the system to support new protocols while the small satellite is in orbit. GNU Radio comes with several pre-built modulation and encoding packages that can support our small satellite and the ISL. The user would have to modify it to the mission specifications and parameters of course.

2.8 Areas of Inquiry Contributing Research

2.8.1 Software defined radio & gnu radio. When it comes to SDR and its counterpart GNU Radio, the increased popularity in the satellite communication world, RF world, and Amateur Radio users, has brought about the idea of how the software can be used to improve various applications. There are current applications being used today, such as USB TV Tuners [18], which in turn can see increased productivity with the implementation of Software Defined Radio. The GNU Radio software is currently being used academically, commercially, and for research purposes as well as real time radio communication purposes. The idea of GNU Radio being a free open source software development toolkit enables anyone to incorporate their own designs thus making their project available to the community. It is widely used in hobbyist, academic and commercial environments to support wireless communications research as well as to implement real world radio systems. Some of the GNU Radio projects being conducted, which are in progress, now include [49]:

1. GNU Radio for TDMA waveforms and applications.
2. GNU Radio in RADAR system for broadcast of Television as its signal source.
3. GNU Radio used in radio astronomy.
4. GNU Radio for Amateur radio transceivers.
5. GNU Radio for distributed measurement of spectrum utilization.
6. GNU Radio for RFID detectors and readers.

7. GNU Radio with software Global Positioning System (GPS).

8. GNU Radio for Multiple Input Multiple output Processing (MIMO).

9. GNU Radio for weather reception signals.

The first project discusses a TDMA based cooperative sensing using GNU Radio for Cognitive Radio applications. The GNU Radio with software GPS is a project in which an open source Galileo E1 software receiver has been implemented. GNU Radio equipped with a RTL2832 dongle has also been used in several projects to receive and decode National Oceanic and Atmospheric Administration (NOAA) signals. The GNU Radio software was used to output files and by using decoding software, produce images.

2.8.2 Small satellites. In this section a description of small satellites, the constellation types and missions is given. Small satellites began their usage with universities due to their limited budget and interest in satellite research. From here, it gained popularity from government agencies, research laboratories, and even private/industrial sectors. The cost and time invested in developing just a single large satellite fueled the innovation for the development of small satellites, which in turn saved time, money and resources. With the advancements in technology and microelectronics, small satellites are able to take advantage of Commercial-Off-The-Shelf (COTS) components to integrate with the small satellites. From this the time taken in developing the satellite to launching the satellite is reduced dramatically.

Table 5

Satellite mass classification table

Satellite Classification	Mass

Table 5

Cont.

Large Satellite	>1000 kilograms
Medium Sized Satellite	500-1000 kilograms
Mini Satellite	100-500 kilograms
Micro Satellite	10-100 kilograms
Nano Satellite	1-10 kilograms
Pico Satellite	0.1-1 kilograms
Femto Satellite	<100 grams

Small satellites are classified into several classes based on their mass. Table 5 provides a listing of the small satellite size classification. For instance a Nano-Satellite is less than 10 Kg in mass and a Pico-Satellite is less than 1 Kg in mass. CubeSats are a type of small satellite being built by various universities and researchers due to its size, feasibility to build, and impact on satellite research. Researchers at Cal Poly and Stanford first developed them to help universities worldwide to perform space science and exploration feasibly. Their dimensions are based on a 10 cm x 10 cm x 10 cm cube (1 U) and can be configured in the following way: 1U, 2U, 3U, or 6U. Total launch cost for small satellites are under a few million dollars in comparison to $200-1000 million for a full-sized one. The Boeing launch vehicle aimed to launch small payloads of 45 Kg, with cost as low as $300,000 per launch, using their Small Launch Vehicle (SLV) concept, which could be in service by 2020 [3]. The minimum price of a pico-satellite (the size of a soda can) launch is $12,000 [4].

The two most regularly used constellations in literature for LEO networks according to [53] are the Walker Star Constellations and the Walker Delta Constellations, named after J.G. Walker, who originally dubbed the two types as Star and Delta pattern constellations. In star constellations, also called Polar constellation, the inclination i~90° is the same for all the orbit planes, P, and N is the number of satellites in the constellation. N = N/P denotes the number of satellites per orbit plane. With the satellites evenly spaced in orbit, satellites can handover connections to the following satellites [53]. In order to provide continuous coverage between counter-rotating satellites, the spacing between satellites must be smaller than 2Δ [11]. In Delta constellation, the orbital planes, P, are equally inclined with inclination less than 90° and their ascending nodes are equally spaced along the full 360° of the equatorial plane [11], [12]. The polar view in this type of constellation forms a triangular orbital shape consisting of three planes or in this case a Greek delta shape Δ is formed around the poles [53].

Instead of using a single large satellite, various smaller satellites are being deployed in a constellation or in clusters for various distributed space science missions because of their potential to perform coordinated measurements of remote space. This helps to provide greater spatial and temporal resolution of the target. Examples of multiple satellite missions with inter-satellite communications are Iridium, Orblink, Teledesic, Proba-3, Edison Demonstration of Smallsat Networks (EDSN) mission, ESPACENET, NASA's Autonomous NanoTechnology Swarm (ANTS), and QB-50 mission [2].

Areas of inquiry for small satellites using software defined have included space and power limitations, data rates, radio frequency allocation, ISL optimization using IEEE 802.11, and antenna design for ISL.

Small satellites, mainly CubeSats, are currently being deployed in space for various data measurements and space exploration. The majority of CubeSats are developed at academic institutions in countries all over the world [28]. These CubeSats contain CMOS cameras [29], gamma ray detectors [30], GPS Receivers [31], and other scientific instruments and devices. These CubeSats, if the 1U size, are 10 x 10 x 10 cm in dimensions therefore not having enough space to hold certain payloads such as a FPGA SDR. This has been a major concern for companies interested in implementing this in the CubeSats due to the CubeSats small size and power consumption. The USRP motherboard is much too large to fit into the CubeSat form factor and in [27] the author proposes a modular FPGA based SDR for it. This CubeSat SDR includes the FPGA firmware and a newly developed motherboard developed by Configurable Space Microsystems Innovative & Applications Center (COSMIAC) at the University of New Mexico. As a whole, this paper [27] presents a flexible, plug and play SDR system for proper placement in CubeSat form factor satellites. Based on the USRP, this new CubeSat SDR provides CubeSat engineers with an easy to use SDR that is compatible with the GNU Radio software and Space Plug-and-Play Avionics (SPA) protocol. In regards to power consumption inquiries, in [27] the author proposes in the future to eliminate the dependency on GNU Radio in order for the satellites to use the CubeSat SDR system without including a General Purpose Processor (GPP) system.

2.8.3 Inter-satellite communication. The inter-satellite link (ISL) allows the satellites to exchange information and share resources to achieve the performance goal while reducing the traffic load to ground. The question as to rather small satellites with SDR can enable increased data rates and address frequency allocation changes have been challenged. The majority of CubeSat projects in the last eight years, and that are planned to launch in the next two years still utilize transceivers and beacons downlinking in the UHF band with no ISL [34]. In the near future, CubeSat programs could use higher frequencies in either the S-band, C-band or X-Band this further reducing the size and mass of the transceiver and the antenna, and gain additional bandwidth to support payloads with significant data downlink requirements. With the future frequency allocation changes coming near, researchers are wondering how the current RF systems will be able to adapt to these changes. The author in [1] believes the implementation of a SDR architecture utilizing a FPGA and RF transceiver will be able to solve back end and front end challenges thereby enabling reception of multiple signals and high data rates. The author also believes this SDR architecture will enable an adaptive and reconfigurable communication system without any hardware changes thus enabling the satellites to adapt to the changing frequencies.

Inter-satellite links for small satellites typically employ omnidirectional antennas due to the constant repositioning of the satellite, as well as the low data rate (<1Mbps). However omni-directional antennas enable security flaws such as eavesdropping by unauthorized ground stations and satellites outside the network but within constellation range. Power also gets radiated in all directions instead of in one direction, in this case the direction of the receiver. The question has arisen as to whether other types of antennas such as smart antennas would prove better for the satellite. However, in [36] a self-steering array that permits secure ISL communication between small satellites moving randomly in space is proposed and proven by the author. Self-steering antennas (also called retrodirective) are able to sense the direction of an incoming radio transmission and send a reply in that same direction, without the complexities associated with phase shift in conventional phased arrays or digital signal processing in smart antennas [36]. Power consumption is minimized as well as the improvement of the communication link budget and network security is improved. The simple power efficient characters of retro directive antennas are extremely applicable to small satellites.

The idea of using other physical layer models for space communications, primarily for ISL has been discussed but never implemented. The terrestrial usage of IEEE 802.11 is one of the fastest growing technologies in the area of wireless communications and networking [9]. However the restricted terrestrial range has placed a hold for those interested in using it for space usage. The idea in [35] focuses on testing the suitability of Commercial-Off-The-Shelf communication protocols, primarily the IEEE 802.11 (Wi-Fi) standard. The author assumes, due to the usage of IEEE 802.11 strictly for terrestrial usage for outdoor distances of only 300 meters, a modification for increased range version for space inter-satellite links can be developed. Redefinition of the IEEE 802.11 inter-frame for increased range is observed. This is done first by looking into the 3 types of inter-frame spacing defined in 802.11, namely the Short Inter Frame Space (SIFs~10µS), DCF Inter Frame Space (DIFs~ 50µS) and PCF Inter Frame Space (PIFs ~ 28µS). In [53] he focuses on the SIF, which is used to separate transmissions belonging to a single dialog like ACK, RTS and CTS. By exploiting the fact that the inter frame time out intervals are not explicitly specified in 802.11 and it is meant for indoor use of 100 meters range, the effect of increasing delay on the MAC and the physical layer, as the range increases up to 10 km were investigated. The results below in Figure 9, from [35] prove that IEEE 802.11 is feasible for ISL range for up 10 km by redefining the inter-frame delays without any modifications to the standard. This would be ideal for short range LEO clusters in formation flying missions while IEEE 802.16 would be ideal for ISL lengths of few hundred kilometers [54].

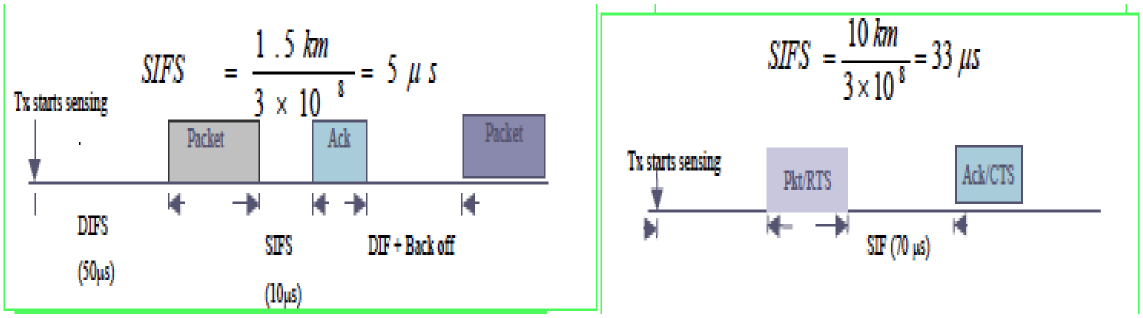


Figure 9. IEEE 802.11 control inter-frame timing (left) and modifications to the inter-frame control signals for increased range (right) [35].

The inter-satellite link is commonly used for navigation of the spacecraft and exchanging of data between one another, therefore performing pseudo-range measurements is a common way to realize relative navigation. The feasibility of using 2-4GHz as future GNSS frequencies is discussed in [55] and concludes that frequency allocation has an effect on the pseudo-range error budget, including ionosphere effects, carrier multipath and the signal acquisition process. Direct absorption at low altitudes and reflection at higher altitudes only occurs for signals below about 30 MHz [61]. For inter-satellite communications free space propagation usually occurs where there are no obstructions between the satellite links. However in some of these satellite links multipath fading could occur due to blockage resulting from high-rise when the satellite is seen at a low elevation angle or conditions in the atmosphere from clouds, rain and refraction from atmospheric layers and weather conditions. Ionosphere effects are inversely proportional to the square of the carrier frequency, and carrier multipath is inversely proportional to the carrier frequency. Therefore, a higher frequency allocation helps to reduce the errors caused by ionosphere path delay and to mitigate the carrier multipath as well. However due to higher maximum Doppler shifts at higher frequency, the Doppler search region increases, which negatively influences signal acquisition. If we assume identical code length, signal acquisition will then take longer at high frequency bands. The author in [54] compares attenuation and Doppler shift characteristics of the ISM band frequencies, 2.4 GHz and 5.5 GHz. It is observed that the transmit frequency of 2.4 GHz exhibits better attenuation of over 10 dBs compared to the 5.5 GHz frequency when used between satellites in different orbital planes.

Software Defined Radio technology today is required to handle multiple waveforms, modulation techniques, pulse shaping techniques and transmit power. The choice in deciding which modulation scheme to employ affects the design of the inter-satellite communication channel in terms of robust performance in fading, Doppler frequency and spectral efficient modulation for least amount of interference for adjacent and neighboring channels. The effect of co-channel and adjacent channel interference is an important factor in evaluating potential modulation schemes for our inter-satellite communications link. In [22], it is observed that MSK scheme has a large advantage over the amplitude modulation and phase modulation schemes, when no post modulation is employed. Non-coherent FSK and BPSK show the minimum degradation from ideal performance, while the 8-ary and 16-ary schemes show maximum degradation [22].

CHAPTER 3

Methodology

3.1 Rationale

The overall rationale for the implementation of software defined radio for inter-satellite communication is flexibility and re-configurability. Constellations, clusters, and formations are quickly becoming utilized for performance of scientific and technological missions in a more affordable way. The motivation of these missions intensifies the challenges of supporting multiple signals, increasing data rates over inter-satellite links, and supporting the change of future communication standards. The solution proposed is the use of SDR as a reconfigurable and adaptive tool for designing the ISL. The benefits of SDR in small satellites such as, cognitive and adaptive operation, multi-mode operation, radio reconfiguration, and remote upgrade are some of the elements the ISL will be able to exhibit. For the use of this study we employ the USRP N210 by Ettus Research, due to a wider range of frequencies offered by the N210. GNU Radio has also been adapted to the USRP N210 therefore making it an open source alternative for LabVIEW [25]. By the very nature of open source software, GNU Radio makes it possible to integrate several other open source software systems and to further increase the capabilities of the user’s program design.

3.2 Conceptual Approach

In this research, I proposed and implemented an optimum inter-satellite communications link for a network of multiple satellites using DS-CDMA by incorporating software-defined radio. A brief overview of the physical layers, mainly the modulation types, and an overview of the multiple access techniques are discussed in detailed. The reasoning as to why CDMA is the best option for our multiple access technology is discussed as well as limitations of CDMA for inter-satellite communication.

3.2.1 BPSK. In BPSK, the carrier signal's phase varies between two values according to the modulating signal. BPSK is also called 2-PSK for the two values have a 180° difference [56]. One phase represents a binary 1, and the other phase represents a binary 0. As the input digital signal changes state (from a 1 to a 0 or from a 0 to a 1), the phase of the output carrier shifts between two angles that are separated by 180°. If the sinusoidal carrier has amplitude A_c and energy per bit $E_b = ½ (A_c)^2 T_b$ (bit duration), then the transmitted BPSK signal is given by:

$$s_o(t) = \sqrt{\frac{2E_b}{T_b}}\cos(2\pi f_c t + \pi) = -\sqrt{\frac{2E_b}{T_b}}\ cos(2\pi f_c t)\ for\ binary\ \text{"0"}$$

$$s_1(t) = \sqrt{\frac{2E_b}{T_b}}\ cos(2\pi f_c t)\ for\ binary\ \text{"1"} \qquad (1)$$

This modulation type is a coherent system which requires the least amount of power to support a given data throughput and error rate therefore making it beneficial due to the possibility of more efficient usage of bandwidth (higher data rate).

3.2.2 QPSK. Quadrature Phase Shift Keying (QPSK), similar to BPSK, offers twice the bandwidth efficiency of BPSK. For every single modulation symbol two bits are transmitted. The carrier phase takes on four equally spaced values such as 0, π/2, π, 3 π/2. The two modulated signals, each of which can be considered to be a BPSK signal, are summed to produce a QPSK signal. QPSK transmitters and receivers are more complicated than the ones used for BPSK. Similar to BPSK, there are phase ambiguity problems at the receiving end, and differentially encoded QPSK is often used in practice. QPSK has two dimensional constellation diagrams with four points with the distance between adjacent points in the constellation given by $\sqrt{2E_s}$. Each symbol consists of two bits therefore $E_s = 2E_b$, then the distance between two neighboring points in the QPSK constellation is given by $2\sqrt{E_b}$ [22].

3.2.3 CDMA. The three basic techniques for sharing links in distributed spacecraft systems are FDMA, TDMA and CDMA [38]. In this study we focus on CDMA due to its useful properties: multiple user access, multipath tolerance, and accurate ranging. In section 3.2.4 we will discuss why CDMA is chosen over the other two multiple access technologies.

It is important to identify which inter-satellite communication network architecture will be implemented for the small satellites. Code Division Multiple Access (CDMA) is a spread spectrum technology, in which a user is assigned a unique pseudo-random code (PN). The narrowband message is multiplied by a larger bandwidth signal thus enabling all users in the CDMA system to utilize the same frequency band and transmit simultaneously. The transmitted signal is then recovered by correlating the received signal with the PN code used by the transmitter. Figure 10 shows a basic CDMA scheme in which that data to be transmitted (a) is spread before transmission by modulating the data using a PN code. This broadens the spectrum as shown in (b). The process gain in this figure is 100 as the spread spectrum is 125 times greater than the data bandwidth. In (c) the actual received signal is viewed consisting of the required signal with background noise and interference from any other CDMA users or radio sources. In (d) the received signal is recovered by multiplying the signal by the original spreading code. The wanted signal here has been filtered thus removing the wide spread interference and noise signal.

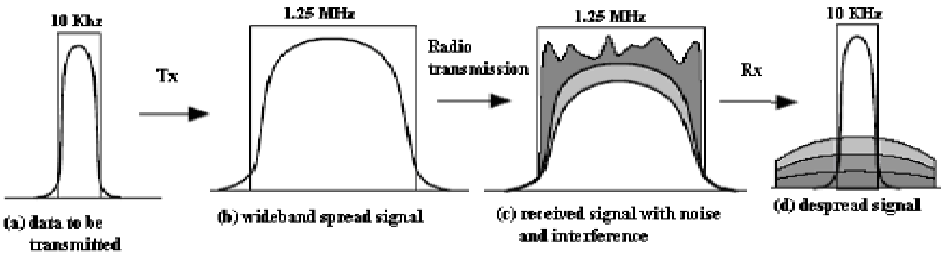

10 Khz
Tx
1.25 MHz
Radio transmission
1.25 MHz
Rx
10 KHz
(a) data to be transmitted
(b) wideband spread signal
(c) received signal with noise and interference
(d) despread signal

Figure 10. Basic CDMA generation [6]

3.2.4 Direct spread spectrum sequence. The data communication mostly adopts a type of spread spectrum modulation technique named Direct Sequence Spread Spectrum (DSSS) due to its robustness against fading, multiple access capability and anti-interference capability. Figure 11 shows a DSSS block diagram consisting of a transmitted signal, which is PSK, modulated, spread at the receiver with the PN sequence, and dispread with the same PN sequence.

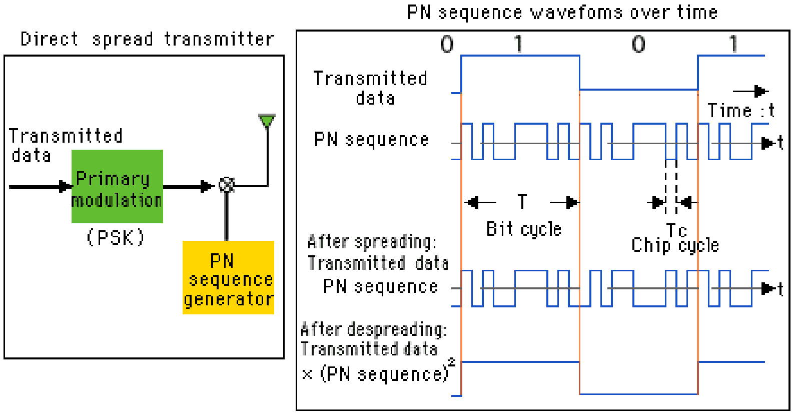


Figure 11. DSSS spreading and de-spreading [6].

To provide a practical solution for adaptive optimal inter-satellite communications in a network of small satellites, in this work I propose to use a software defined implementation of Direct Sequence Code Division Multiple Access (DS-CDMA) technique. The performance of the entire system largely depends on the design of the Multiple Access (MAC) protocol. The basic function of a MAC protocol is to avoid collision by arbitrating the access of the shared medium among the nodes in the network [13]. There are two different types of MAC protocol: contention based and collision free protocols. Numerous contention based protocols have been proposed in the literature [15], for example, ALOHA, CSMA (Carrier Sense Multiple Access), BTMA (Busy Tone Multiple Access), ISMA (Idle Signal Multiple Access), etc. The collision free protocols ensure that collision of data packet never occurs. Some of the basic protocols of this type are TDMA (Time Division Multiple Access), FDMA (Frequency Division Multiple Access) and CDMA (Code Division Multiple Access). Authors in [15, 16] propose to use Carrier Sense Multiple Access with Request-to-Send and Clear-to- Send protocol (CSMA/CA/RTS/CTS) for various formation flying patterns of small satellites. However, it is concluded that this protocol cannot be used for missions that require real to near real time communication between the satellites. In this project, I propose to use collision free protocols in particular DS-CDMA since it address the design needs of a large scalable network and provides high throughput. Frequency Division Multiple Access (FDMA), in which the radio spectrum is divided among different users, is not an economic choice for inter-satellite communication, as it requires a large bandwidth [16]. Time Division Multiple Access (TDMA) restricts fixed time slots to each user and has strict timing synchronization; hence it is not a practical option for heterogeneous small satellites networks with various data transmission rates [3] because transmission simultaneously at different times is required. Also for TDMA, the mother satellite may not be able to cover the whole system within its transmission range because of the low transmission power and time scheduling will be difficult in a scalable network. In TDMA the signal is broken up by timeslots.

The USRP N210 utilizes only one daughter card therefore you would have to switch between transmit and receive mode. The switching could take time, which would impact the ability to cleanly transmit in a specific timeslot. The complexity of the system could limit others from usage due to its various modules, blocks, coding, and information one must know in order to have a feel for the system. An alternative option for multiple accesses could be the implementation of IEEE 802.11 physical layer and MAC layer integrated with smart antennas for ISL applications. The author in [53] investigates the usage of terrestrial smart antenna integration techniques with both the physical and MAC layers in order to increase the capacity while minimizing interference and collision in close proximity links. Carefully putting all these options into consideration and seeing which technique best matches our proposed system, Code Division Multiple Access (CDMA) is chosen for this project. It enables multiple users to transmit simultaneously using their own Pseudo Random Noise (PRN) codes or their truly orthogonal codes [2].

A CDMA system has several limitations like cross correlation error and the near-far effect. Cross correlation results from the non-perfect orthogonal PRN codes. This will not be an issue in most current small satellites applications, noting the limited number of satellites in a cluster [38]. The near-far effect can be mitigated by appropriate power control mechanism. In [38] the authors suggested protocol requires strict time synchronization, which can be achieved by using GPS/GNSS clock or clock synchronization in conjunction with a highly accurate on-board clock.

3.2.5 Test-bed. In this study, we designed a novel inter-satellite communication model for a network of small satellites based on implementing a DS-CDMA protocol in GNU Radio. The performance of this model has been evaluated and compared for different modulation (BPSK and QPSK) and channel coding techniques (un-coded and convolutional coding). The signal is tested in AWGN channel model, slow Rayleigh channel model, and slow Rician channel model. We assume a distributed mesh topology which supports direct interaction among all distributed assets, and provide real time communication to each satellite. Figure 12 illustrates one USRP to act as the master node to serve as the control point or data collector point for the formation. The other USRPs will serve as the slave nodes. Implementation of the system is achieved using GNU Radio, which provides a GUI to visually see the flow graph connections and blocks. Programming of these approaches was done in C++ and Python, which is the programming language GNU Radio utilizes. GNU Radio, an open source hardware and software, makes it adaptable for other researchers to utilize this model in their projects. This model can be easily reconfigured to support any encoding/decoding, modulation, and other signal processing schemes. To the best of the author’s knowledge, this is the first work to study implementation of CDMA with SDRs for inter-satellite communications.

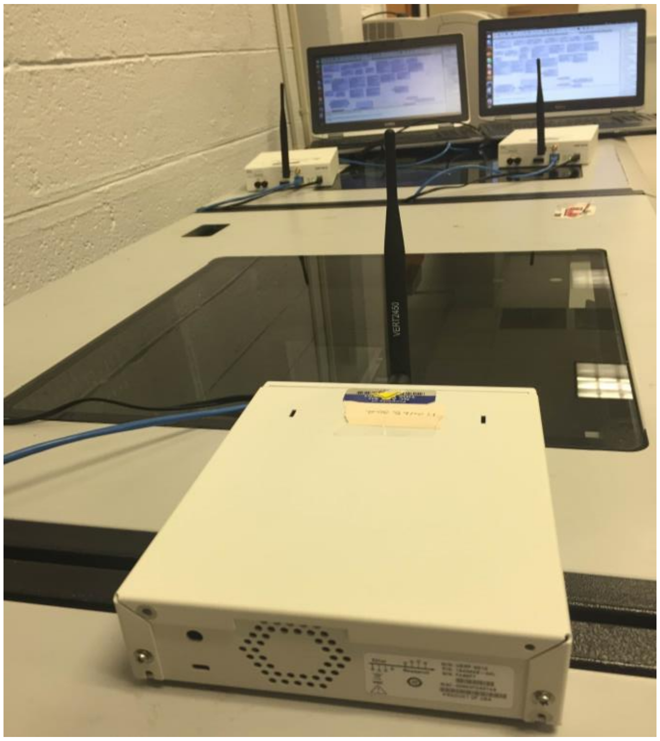


Figure 12. USRP system setup.

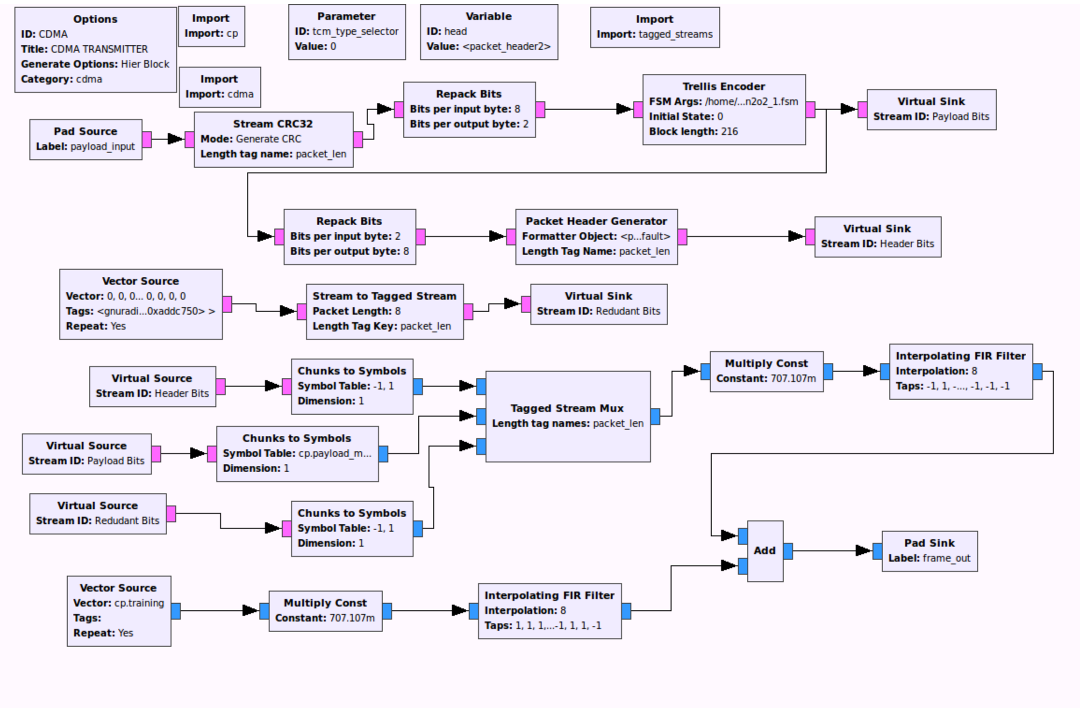
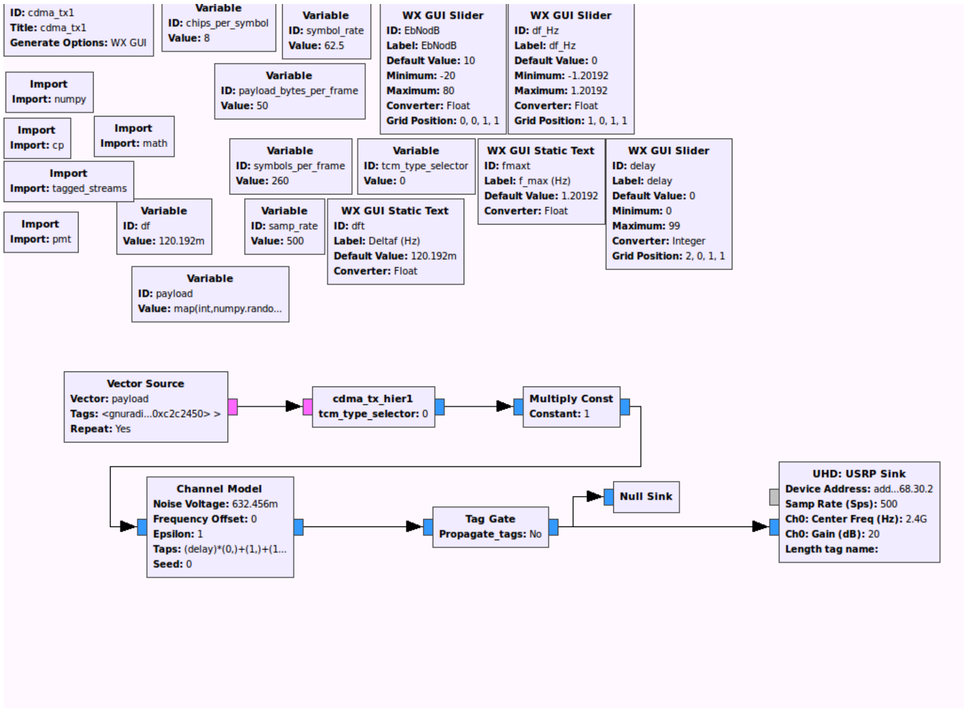


Figure 13. System diagram of hierarchical transmit block.

In GNU Radio, sometimes it makes sense to combine several blocks into one single block. For instance, if you have several applications which all have a common signal processing component consisting of several blocks. These blocks are combined into a single block, which in turn is used in the application of the overall transmitter block design. In this CDMA system this type of configuration is called a hierarchical block. Figure 13 shows the CDMA transmitter hierarchical block. In the CDMA system, the input is a vector of payload bytes (0 to 255 per byte), which then uses cyclic redundancy code (CRC) bits for each input data frame and appends them to the frame. Header bits are generated from the coded payload streams and modulated by users choice. Coded payload bytes are repacked to symbols representing *m* bits, which is the number of bits per symbol for the modulation type, and modulated using a 2^m*-ary* modulation. Modulated header and payload streams are then multiplexed. The multiplexed payload stream is spread by the spreading code sequence and scaled appropriately. The training symbols are also spread by a code sequence (orthogonal to the first spreading code) and scaled accordingly. The two channels are superimposed to form a frame to be transmitted. From this flow graph we construct a block, which is used as part of the flow graph connection for the overall transmitter system. Figure 14 illustrates the overall transmitter flow graph system including the source payload, the "cdma_tx_hier" block, which produces the CDMA frames, the channel model to simulate channel, tags, and the USRP sink block.



ID: cdma_tx1
Title: cdma_tx1
Generate Options: WX GUI

Variable
ID: chips_per_symbol
Value: 8

Variable
ID: symbol_rate
Value: 62.5

WX GUI Slider
ID: EbNodB
Label: EbNodB
Default Value: 10
Minimum: -20
Maximum: 80
Converter: Float
Grid Position: 0, 0, 1, 1

WX GUI Slider
ID: df_Hz
Label: df_Hz
Default Value: 0
Minimum: -1.20192
Maximum: 1.20192
Converter: Float
Grid Position: 1, 0, 1, 1

Import
Import: numpy

Variable
ID: payload_bytes_per_frame
Value: 50

Import
Import: cp

Import
Import: math

Variable
ID: symbols_per_frame
Value: 260

Variable
ID: tcm_type_selector
Value: 0

WX GUI Static Text
ID: fmaxt
Label: f_max (Hz)
Default Value: 1.20192
Converter: Float

WX GUI Slider
ID: delay
Label: delay
Default Value: 0
Minimum: 0
Maximum: 99
Converter: Integer
Grid Position: 2, 0, 1, 1

Import
Import: tagged_streams

Import
Import: pmt

Variable
ID: df
Value: 120.192m

Variable
ID: samp_rate
Value: 500

WX GUI Static Text
ID: dft
Label: Deltaf (Hz)
Default Value: 120.192m
Converter: Float

Variable
ID: payload
Value: map(int,numpy.rando...

Vector Source
Vector: payload
Tags: <gnuradi...0xc2c2450> >
Repeat: Yes

cdma_tx_hier1
tcm_type_selector: 0

Multiply Const
Constant: 1

Channel Model
Noise Voltage: 632.456m
Frequency Offset: 0
Epsilon: 1
Taps: (delay)*(0,)+(1,)+(1...
Seed: 0

Tag Gate
Propagate_tags: No

Null Sink

UHD: USRP Sink
Device Address: add...68.30.2
Samp Rate (Sps): 500
Ch0: Center Freq (Hz): 2.4G
Ch0: Gain (dB): 20
Length tag name:

Figure 14. System diagram of overall transmitter block including hierarchical block.

On the other hand the "cdma_rx_hier" flow graph shown in Figure 15, when compiled, forms the "cdma_rx_hier block. Figure 16 shows the overall system outside of the hierarchical block containing the USRP source, which receives the spread signal that gets inputted to the "cdma_rx_hier block. Here the beginning of the frame is acquired as well as the frequency offset. The received samples are frequency corrected, input to matched filters, de-spread, phase estimated by a PLL, de-rotated, demodulated, and finally the data symbols are CRC decoded.

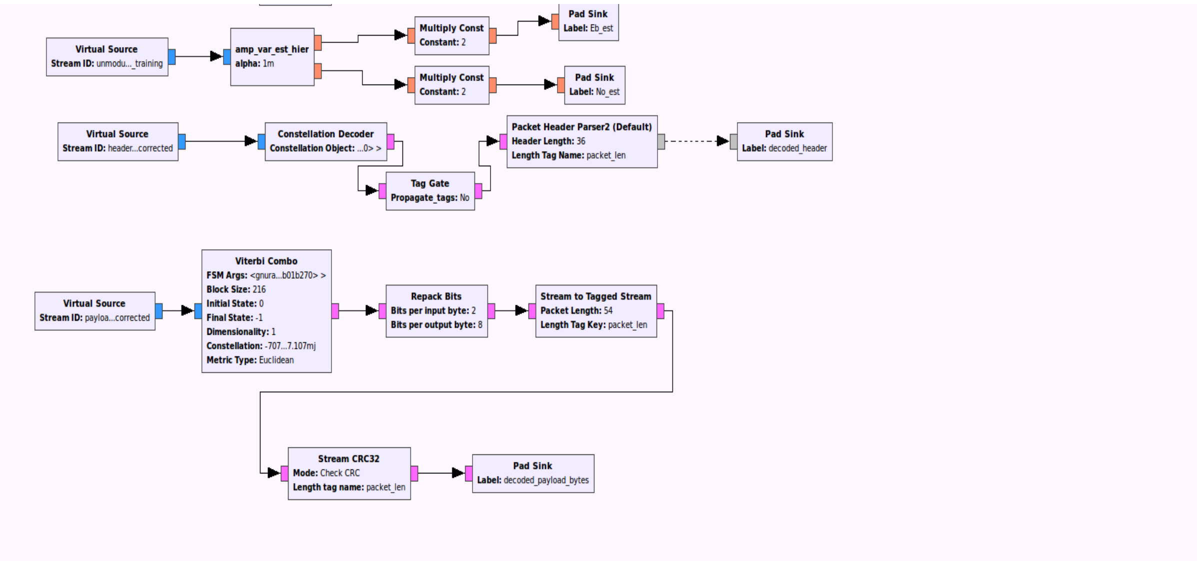


Figure 15. System diagram of receiver hierarchical block.

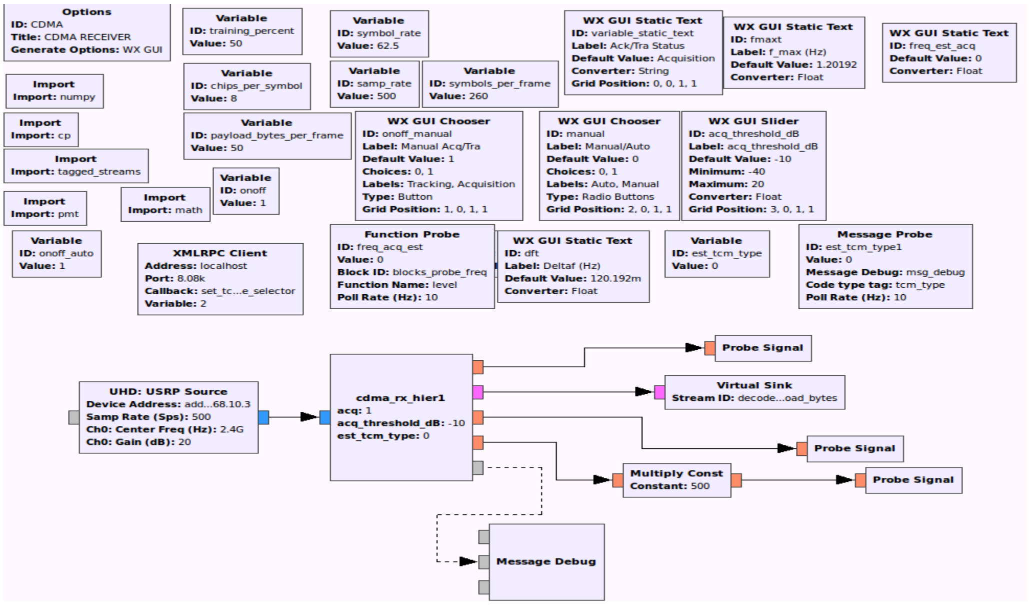


Figure 16. System diagram of overall receiver block including hierarchical block.

3.3 Data Gathering Methods

In this section, we describe the methods used in deriving the data. A brief description of the hardware and software used to gather our data will also be discussed. Refer to Appendix A for the detailed installation of each software and hardware component.

3.3.1 UHD Blocks. The USRP Hardware Driver software (UHD) is the hardware driver for all USRP devices. Developed by Ettus Research, it is compatible with all USRP software defend radios. UHD allows development on multiple operating systems including Linux, Windows, and MacOS. The goal of the UHD software is to provide a host driver and API for current and future Ettus Research products. Users will be able to use UHD software standalone or with third- party applications, such as: GNU Radio, LabView, MATLAB, Simulink, OpenBTS, Iris, RedHawk, and Amarisoft LTE eNodeB [42]. In order for GNU Radio and the USRP device to communicate there are a set of blocks in the gr-uhd component directory, which are the UHD: USRP source block and the UHD: USRP sink block. These two blocks will be the main blocks used in our project for the transmission and reception of signals, mainly seen in the flow graphs. The UHD: USRP sink blow serves as the block that accepts transmitted data from upstream processing blocks. This block will read a stream and transmit the baseband signal. On the other hand, the UHD: USRP source block provides received data to downstream processing blocks. It does this by producing baseband samples by sampling the RF on a selected antenna at a particular frequency, sample rate, and gain. In typical Linux installations the UHD path is /usr/local/share/uhd. This is where a majority of the uhd examples reside in.

3.3.2 WX GUI blocks. The GNU Radio Companion (GRC) GUI app uses a WX toolkit for evaluation of a signal to be displayed graphically in time and frequency domain. This toolkit includes real world applications usually used for analyzing signals to be embedded in the GUI.

This is all done by wxPython, which is a GUI toolkit for the Python programming language. It allows Python programmers to create programs with a robust, highly functional graphical user interface, simply and easily [43]. In GNU Radio these tools made from the wxPython are used to aid in the visualization of our signals and can be further analyzed with one's own external signal analyzer or oscilloscope. The WX GUI FFT sink block serves as the spectrum analyzer and with GNU Radio installed provides a fast Fourier transformation (FFT) of a signal received with a USRP device and daughterboard. The WX GUI Scope sink block serves as the oscilloscope in which the user can view the signal waveforms in the time domain as well as view the characteristics of an IQ (quadrature) signal in the XY mode. Other important WX GUI blocks used in this project include the WX GUI Constellation Sink block, WX GUI Number Sink block, and WX GUI Waterfall Sink block.

3.3.3 Components. The list below demonstrates the main hardware and software that are used in the project.

Hardware:

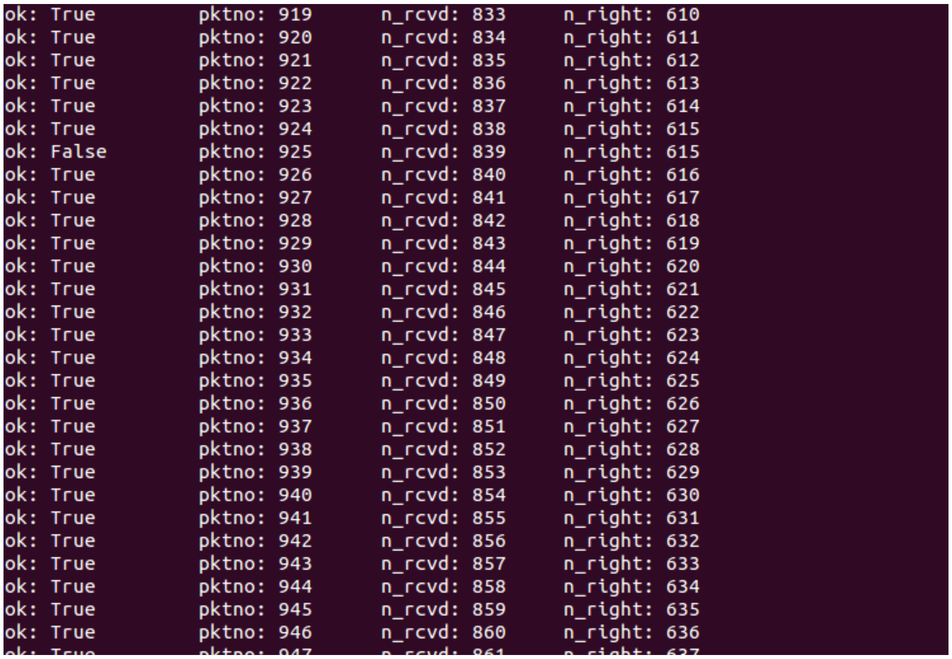
- Dell Latitude E6530 laptop with Intel Core i7 Processor
- USRP N210
- SBX daughterboard (400 MHz – 4.4 GHz)
- VERT 2450 Antenna (2.4-2.5 and 4.9-5.9 GHz)
- RSA 5103B Real-Time Signal Analyzer
- MSO 5240B Mixed Signal Oscilloscope

Software:

- Linux OS: Ubuntu 12.04 LTS 64-bit
- GNU Radio 3.7.6.1

3.3.4 Functional testing of the USRP. To begin the implementation of the DS-CDMA system using the software defined radio and GNU Radio, the first thing to test is the capability of transmitting and receiving on the USRP N210. This is done first to ensure that both radios are able to function and talk to one another without any issues. Both USRP N210 radios were equipped with a VERT2450 antenna on the TX/RX side of the radio. The radios both have a TX/RX port and an RX2 port. The TX/RX port enables transmission and reception at the same time, while the RX2 port enables only reception. Therefore in this case, for the project, both antennas were placed in the TX/RX port. To confirm multiple devices connected to separate gigabit Ethernet interfaces, we assign each Ethernet interface a separate subnet, and assign the corresponding device an address in that subnet.

The GNU Radio Companion (GRC) comes equipped with some example scripts used to test our radios. These files are in the folder named "gnuradio/gr-digital/examples", which have test scripts for Orthogonal Frequency Division Multiplexing (OFDM) transmission and narrowband transmission. We choose the OFDM directory to test our radios by setting one as the transmitter and the other as a receiver. Both OFDM directories come with a pre-generated python script called "benchmark_tx.py" and "benchmark_rx.py", which allows the transmission and reception of OFDM signals with the type of modulation the user wants to use. The script displays the status of the received signal from the transmitter. If the payload passes the Cyclic Redundancy Check (CRC) "ok" will be "True" otherwise it will be "False". "Pktno" displays the data packet number encoded in the header of each payload for labeling and tracking, starting from zero. "N_rcvd" is a counter to keep the track of the total number of received data packets regardless of whether the payload passes the CRC check or not. "N_right" is a counter used to keep track of the total number of correctly received data packets (payload passes the CRC check). This particular script allows the user to test their transmitter and receiver for successfully real time transmission and reception of packets over the air. On one side, the laptop with the SDR acting as the transmitter has the command “./benchmark_tx.py –f 2.4G”, while the laptop acting as the receiver has the command, “./benchmark_rx.py –f 2.4G” to receive these packets. Essentially we are telling the transmitter to transmit packets over the air with a frequency of 2.4 GHz. By default the packet size is 400, Megabytes are set to 1.0, transmit amplitude set to 0.1, bandwidth set to 500000, and modulation set to BPSK. Figure 17 shows the output of the received program after executing the command.



```
ok: True        pktno: 919      n_rcvd: 833     n_right: 610
ok: True        pktno: 920      n_rcvd: 834     n_right: 611
ok: True        pktno: 921      n_rcvd: 835     n_right: 612
ok: True        pktno: 922      n_rcvd: 836     n_right: 613
ok: True        pktno: 923      n_rcvd: 837     n_right: 614
ok: True        pktno: 924      n_rcvd: 838     n_right: 615
ok: False       pktno: 925      n_rcvd: 839     n_right: 615
ok: True        pktno: 926      n_rcvd: 840     n_right: 616
ok: True        pktno: 927      n_rcvd: 841     n_right: 617
ok: True        pktno: 928      n_rcvd: 842     n_right: 618
ok: True        pktno: 929      n_rcvd: 843     n_right: 619
ok: True        pktno: 930      n_rcvd: 844     n_right: 620
ok: True        pktno: 931      n_rcvd: 845     n_right: 621
ok: True        pktno: 932      n_rcvd: 846     n_right: 622
ok: True        pktno: 933      n_rcvd: 847     n_right: 623
ok: True        pktno: 934      n_rcvd: 848     n_right: 624
ok: True        pktno: 935      n_rcvd: 849     n_right: 625
ok: True        pktno: 936      n_rcvd: 850     n_right: 626
ok: True        pktno: 937      n_rcvd: 851     n_right: 627
ok: True        pktno: 938      n_rcvd: 852     n_right: 628
ok: True        pktno: 939      n_rcvd: 853     n_right: 629
ok: True        pktno: 940      n_rcvd: 854     n_right: 630
ok: True        pktno: 941      n_rcvd: 855     n_right: 631
ok: True        pktno: 942      n_rcvd: 856     n_right: 632
ok: True        pktno: 943      n_rcvd: 857     n_right: 633
ok: True        pktno: 944      n_rcvd: 858     n_right: 634
ok: True        pktno: 945      n_rcvd: 859     n_right: 635
ok: True        pktno: 946      n_rcvd: 860     n_right: 636
ok: True        pktno: 947      n_rcvd: 861     n_right: 637
```

Figure 17. Received packet status of the OFDM transmission benchmark script.

3.4 DS-CDMA Implementation in GNU Radio

Table 6 shows design parameters for the simulations, which can also represent certain small satellite requirements. LEO is the orbital altitude used with the number of clusters to be about 3 or 6, each cluster having 3 satellites. This cluster formation flying employs a distributed mesh topology. In the case of a satellite failure, it can be replaced by another satellite. An orbital velocity of 3 km/s with the separation distance between satellites in different orbits to be no wider than 2 km is chosen as well. S-band is chosen as the frequency for the ISL due to achievable high data rates, which in turn assists in faster exchange of data. The transmission range of the ISL is said to be between 10-20 km with data rates of 3-8 Mbps. We assume an acceptable transmission delay of 150-300 ms with 100,000 numbers of packets to be simulated. Due to FPGA's power consumption, typically the 1U CubeSats maximum power budget range is from 1 to 2.5 Watts (W), while the 2U is 2W to 5W. The USRP N210 FPGA consumes 1 to 2.5 Watts [5]; therefore we choose the transmission power to range from 500mW to 3W. System performance of the proposed satellite network using DS-CDMA is investigated with BPSK and QPSK modulation over AWGN, Rayleigh, and Rician channels.

Table 6.

Simulation parameters.

Design Parameters	Value
Transmission Power for TX	500 mW – 3W
Transmission Power for RX	200 mW – 500 mW
Frequency	S Band/ISM 2.4 GHz
Orbit Altitude	LEO (300 km – 500 km)
Orbit Shape	Circular (Polar)
Orbit Velocity	3 km/s
Topology	Distributed Mesh
Separation Distance Between Satellites in Different Orbits	No wider than 2 km

Table 6

Cont.

Number of Satellites per cluster	3
Number of Clusters	3 - 5
ISL Transmission Range	10 km – 20 km
ISL Data Rates	3- 8 Mbps
Acceptable Transmission Delay	150 – 300 ms
Number of Packets	100,000

At the transmitter, the data is generated from a vector random source and modulation is done by either BPSK or QPSK to map the bits to symbols. The DS-CDMA model follows a synchronous system whereby Walsh Hadamard codes is employed for our system. The model uses 8 chips per symbol with two orthogonal codes for training and data as the spreading code. The data rate employed is 500 Kbps, which results in a chip rate of 4,000 Kchips/s. The carrier frequency for the test bed is 2.4 GHz. Additive White Gaussian Noise (AWGN), Rayleigh, and Rician Fading channels are implemented in this research in the case of Channel Side Information (CSI) being known at the receiver. Walsh Code will be used for generating the orthogonally codes, in which each cluster has their own unique set of codes to avoid interference from neighboring clusters. An 8 length Walsh-Hadamard Code [7] is used, thus giving the system the ability to handle 8 different users at a given time. At the receiver the hierarchical block will take the received samples as input, frequency correct them, “chop” the samples according to the generated flags, input them to two matched filters, perform the dispreading, de-rotate the data symbols, which are then demodulated using a constellation decoder and a Viterbi Decoder. Finally the symbols are parsed and the data symbols are CRC decoded and output. This SDR CDMA module is intended to build a parameterized CDMA physical layer for usage in GNU Radio. The spreading, modulations, framing parameters, etc. are set by the user, from a global parameter file, thus making the module reconfigurable.

CHAPTER 4

Results

4.1 Bit Error Rate Calculation

Bit Error Rate (BER) is a performance measurement specifying the number of corrupted bits as they are transmitted from its source to its destination. Several factors that can affect BER are bandwidth, signal-to-noise ratio, and transmission speed and transmission medium. Authors in [59] propose using the Standard Gaussian Approximation to calculate BER performance for a binary DS-CDMA systems operating in AWGN for a synchronous system. The authors consider for QPSK and BPSK modulation schemes, the relation between bit error probability and signal to noise ratio (E_b/N0) over AWGN channel in absence of interferers to be expressed by:

$$P_e = Q(SNR), SNR = [\left(\frac{2Eb}{N0}\right)^{-1} + \frac{K-1}{2N}]^{-1/2} \qquad (1)$$

Where K is the number of active users in the system, N represents the number of chips per data symbol, E_b is the energy per bit and N_0 is the noise power spectral density.

When considering a slow Rayleigh fading channel with random sequences, the authors in [60] also propose using the Standard Gaussian Approximation. The authors assume a matched filter receiver and all transmitted signals to have the same transmission power of 2 Watts therefore the SNR does not appear in the formula. The approximation is thus expressed by:

$$P_e^{SYNC} = \frac{1}{2}\left[\!\left[1 - \frac{1}{\sqrt{1+\frac{N_0}{4N}+\sum_{k=2}^{K}\rho_{k1}^{2}}}\right]\!\right] \qquad (2)$$

Where the cross correlation coefficient between the k^{th} user and user 1's signature sequences is represented by ρ_{k1}. The transmitted signal is also assumed to experience an Additive White Gaussian Noise with a two-sided power spectral density of $N_0/2$.

The average bit error rate in DS-CDMA systems under the Rician fading environment on the assumption of the frequency non-selective fading is also calculated with BPSK modulation. The DS-CDMA model was simulated on a AWGN, Rayleigh, and Rician channel model to show the effect of the channels BER performance with respect to signal to noise ratio. The system follows a synchronous CDMA using Walsh Hadamard orthogonal code as the spreading code. For this binary direct sequence system the chip signal is rectangular as well.

4.2 BER Performance of DS-CDMA on AWGN and Rayleigh Channel

Figure 18 shows the simulated and theoretical BER performance for a two-user case with BPSK modulation over AWGN and Rayleigh fading using equation 1 and 2. The channel has also been modeled with free space in cases with long distances among the satellites [24]. It is assumed that the receiver has perfect knowledge of the channel condition. DS-CDMA performance effect on AWGN and Rayleigh Fading Channel for a two-user system sending 100,000 bits on the channel is observed. The users data are spread using two Walsh codes of length 8 that are orthogonal and have good autocorrelation properties. The signal is then modulated by Binary Phase Shift Keying (BPSK) modulation. After the signal has been received it is then dispread and demodulated. Figure 18 shows the BPSK simulated AWGN and simulated Rayleigh bit error rates were very close to the theoretical values. As the SNR increases the bit error rate decreases and when the SNR decreases there is an increased in bit error rate.

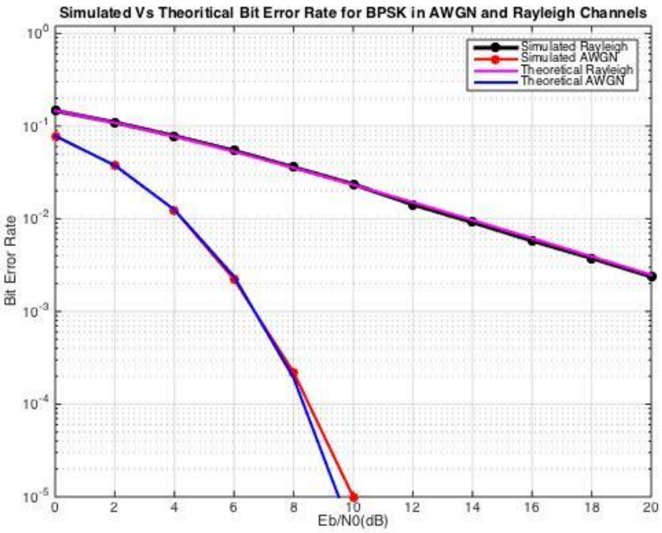


Figure 18. Performance of DS-CDMA BPSK modulation in AWGN & Rayleigh environment.

4.3 BER Performance of DS-CDMA on AWGN and Rayleigh Channel

Figure 19 shows the simulated and theoretical BER performance for a two-user case with QPSK modulation over AWGN and Rayleigh fading. The channel has also been modeled with free space in cases with long distances among the satellites [24]. It is assumed that the receiver has perfect knowledge of the channel condition. DS-CDMA performance effect on AWGN and Rayleigh Fading Channel for a two-user system sending 100,000 bits on the channel is observed. The users data are spread using two Walsh codes of length 8 that are orthogonal and have good autocorrelation properties. The signal is then modulated by Quadrature Phase Shift Keying (QPSK) modulation. Figure 19 shows the QPSK simulated AWGN and simulated Rayleigh bit error rates were very close to the theoretical values. As the SNR increases the bit error rate decreases and when the SNR decreases there is an increased in bit error rate.

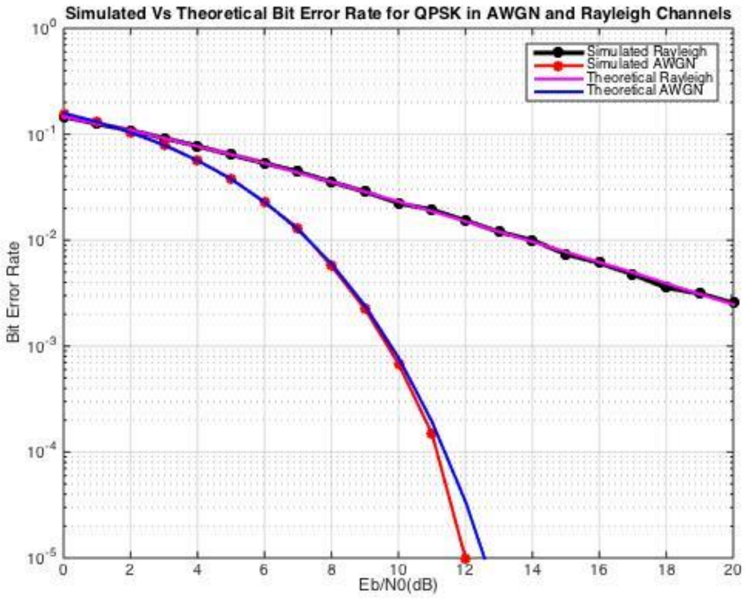


Figure 19. *Performance of DS-CDMA QPSK modulation in AWGN & Rayleigh environment.*

4.4 BER Performance of DS-CDMA on Rician Channel

Fading becomes Rice-distributed when a line of sight path is introduced into a Rayleigh Fading environment. Figure 20 illustrates, in this case, the bit error rate for the theoretical and simulated BPSK modulated signal in a flat fading Rician channel for two users. The simulated BER for two users is close to the theoretical BER with the bit error rate decreasing as the signal to noise ratio increases.

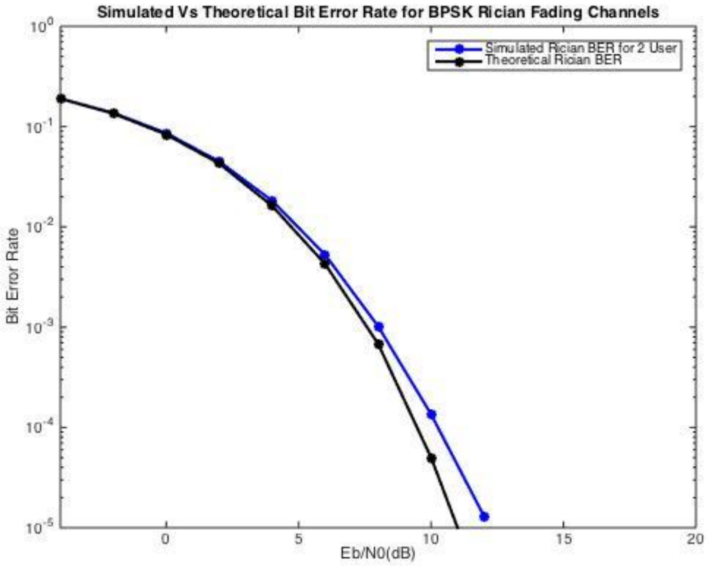


Figure 20. Performance of DS-CDMA BPSK modulation in Rician environment.

4.5 BER Performance of QPSK on AWGN Channel for Multiple Users

Figure 21 shows the Bit Error Rate performance in an AWGN communication channel for 1, 2, 5, 7 users. The figure shows the growth of the BER as more users are introduced to the system. For each user introduced to the system there is a decrease in bit error rate as their SNR increases. Quadrature Phase Shift Keying (QPSK) modulation is employed for this test.

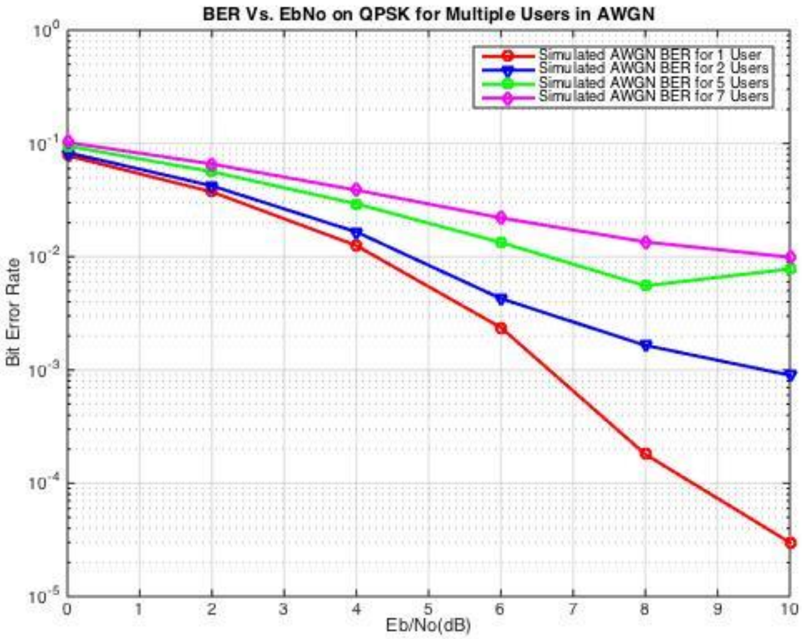


Figure 21. Performance of DS-CDMA QPSK modulation on AWGN for users 1,2,5,7.

4.6 BER Performance of QPSK on Rayleigh Channel for Multiple Users

Figure 22 shows the Bit Error Rate performance in a Rayleigh communication channel for 2, 4, 5, 7 users. The figure shows the growth of the BER as more users are introduced to the system. For each user introduced to the system there is a decrease in bit error rate as their SNR increases. When more users are introduced to the system and have a small SNR values, their BER increases. Quadrature Phase Shift Keying (QPSK) modulation is employed for this test.

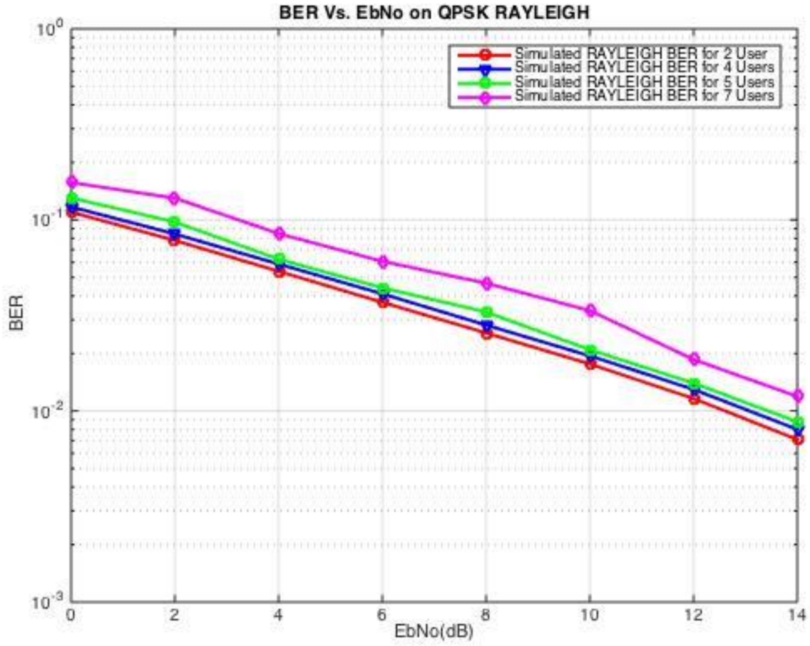


Figure 22. Performance of DS-CDMA QPSK modulation on Rayleigh channel for users 2,4,5,7.

CHAPTER 5

Discussion and Future Research

5.1 Conclusion

Using Software Defined Radio technology, I have designed and implemented a DS-CDMA system for optimization of an inter-satellite link for small satellites. The experimental result using the implemented system clarifies the theoretical and simulated performances of the transmitted and received signals by their bit error rate measurements.

5.2 Further Investigation

In the future it would be wise to think about the security aspect of the ISL and overall distributed wireless network. The security of the network plays a major role because the signals can easily be received by other hardware not in the network. Interferences coming from the external world as well are an issue. The security of this network could possibly be maintained by having cryptographic algorithms for instance. The cryptanalysis techniques developed later may render the current security evaluation insecure. Another solution for further investigation is through digital certification, which is a way of assuring that a public key is actually from the correct source. The Digital Certificate is digitally signed by a trusted third-party [37]. Due to the digital certificate now being a signed data file itself, its authenticity can be determined by verifying its digital signature.

Inter-satellite links generate excess interference to the space-to-ground link as the distance between two satellites decreases thus a power control technique is needed to support an optimal power level at all separation distances. We could introduce a power control algorithm for inter-satellite links that would implement dynamically adjustable power attenuation in the transmitters to provide a minimum signal to interference power density ratio that satisfies the signal acquisition and tracking requirements at the receiver. Designing of a signal to interference balancing algorithm for a closed loop power control in inter-satellite links could prove to show substantial power savings [58].

The self-steering features of retro-directive antennas make them attractive for secure small satellite ISL applications. The high directivity of the inter satellite communication link makes it suitable for security sensitive missions. Currently, because of this, the University of Hawaii [36] is investigating this problem as its contribution to the University Nanosat Program, sponsored by the Air Force Office of Scientific Research and administered through the Air Force Research Labs and NASA Goddard Space Flight Center.

Determining the optimum path for data packets being transmitted between a sender and receiver can be challenging, and one must consider the transmission power and time. Multiple hops are potentially applicable to communicate over long distances that are greater than the normal transmit power can support [38]. From this, data can be relayed through the satellites to achieve the desired end to end communication objective that cannot be achieved by utilizing a direct, line of sight inter-satellite communication. For future continued testing we plan to use a reactive scheme, which focuses on on-demand routing, in which a satellite attempts to find an optimal path to its destination only when the need to have a connection is established. We plan to impose either the Probabilistic Forwarding Protocol or Bandwidth Delay Satellite Routing as the network routing protocol and implement this into GNU Radio. According to [57], the Probabilistic Forwarding Protocol shows that it is very efficient and manages to propagate data very close to the sink, thus it is robust while the Bandwidth Delay Satellite Routing proves it can adapt according to link situations and choose alternate paths, thus improving the overall system performance. We also plan on integrating the IEEE 802.11 standard into GNU Radio and redefining the inter-frame delays for increased space range. The IEEE 802.11 standard is chosen in order to test the suitability of commercial-off-the-shelf communication protocols. Finally we would like to design the model for an increased number of users. Extending the model to more users will enable us to perform more testing and verifications for the system using different diversity combining techniques such as Equal Gain Combining and Maximal-ratio combining for an improved signal.

References

[1] Maheshwarappa M., Bridges C., “Software Defined Radios for Small Satellites”, NASA/ESA Conference on Adaptive Hardware and Systems (AHS), Surrey Space Centre, 2014.

[2] Radhakrishnan R., Edmonson W., Afghah F., Chenou J., Zeng Q., Osorio R., “Optimal Multiple Access Protocol for Inter-Satellite Communication in Small Satellite System, 4S Symposium”, 2011.

[3] Mitchell G., “Determining Optimum Modulation for Inter-Satellite Communication Systems”, 16th Annual AIAA/USU Conference on Small Satellites, SSC02-II-4.

[4] Rohi H., Hajghassem H., Hoseini M., “CDMA in Low Earth Orbit Satellites”, MALEK ASHTAR University of Technology, Tehran Iran.

[5] Products. [Online]. Available: http://www.ettus.com/product

[6] Hashmi M., Dhakad P., Nagaria B., “Design and Analysis of DS-CDMA Rake Receiver Simulator for Wireless Communication”, IEEE Vol. 978-1-4244-9190-2/11, 2011.

[7] Viswanathan M., “Simulation of Digital Communication Systems Using Matlab”, Second Edition, Ebook. 2013.

[8] Hartenstein R., Keevalil A., “Field-Programmable Logic and Applications. From FPGAs to Computing Paradigm”, Lecture Notes in Computer Science (Book 1482), Springer, 1998 edition.

[9] Sithirasenan E., Muthukkumarasamy V., Powell D., “IEEE 802.11i WLAN Security Protocol- A Software Engineer’s Model”, Proc. Of 4th AusCERT Asia Pacific Information Technology Security Gold, 2005.

[10] Chen B. and Yu L., “Design and Implementation of LDMA for Low Earth Orbit Satellite Formation Network”, IFIP 9th International Conference, 409-413, 2011.

[11] Galtier J., "Geographical reservation for guaranteed handover and routing in low earth orbit constellations”, Proceedings Workshop de Comunicacao Sem Fio (WCSF '99), Belo Horizonte, Brazil, July 1999.

[12] Sundaramoorthy P.P., Gil E. Verhoeven C., "Systematic Identification of Applications for Cluster of FemtoSatellites," 61st International Astronautical Congress, Prague, CZ, 2010.

[13] Zhang D., Liu S., Yin M., “A Satellite Routing Algorithm Based on Optimization of Both Delay and Bandwidth”, Key Lab of Broadband Wireless Communication and Sensor Network Technology, Nanjing University, Sept 2011.

[14] Riyanto B., Langi A., Kurniawan A., Marpanaji E., Mahendra A., Liung T., “Software Architecture of Software Defined Radio (SDR)”, ITB Research Center on ICT.

[15] Radhakishnan R., Zeng Q.A., and Edmonson W.W., “Inter-satellite Communications for Small Satellite Systems," International Journal of Interdisciplinary Telecommunications and Networking, Vol. 5, No. 3, pp. 11{24}, 2013.

[16] Radhakishnan R., Edmonson W.W., and Zeng Q.A., “The Performance Evaluation of Distributed Inter-satellite Communication Protocols for Cube Satellite Systems," The 4th Design Development and Research Conference, South Africa, 2014.

[17] Li Y., He Z., Voigt T., “A Software Radio-Empowered Sensor Network”, Swedish Institute of Computer Science.

[18] Using a DVD USB Stick as a SDR Receiver. [Online].
Available: http://sdr.osmocom.org/trac/wifi/rtl-sdr.

[19] Vulcan Wireless SDR Products. [Online]. Available from http://www.vulcanwireless.com/ Products, April 2011.

[20] Blossom E., 2014, September 5. *GNU Radio.* [Article].

Available: http://en.wikipedia.org/wiki/GNU_Radio.

[21] Wolf M., "High Performance Embedded Computing: Applications in Cyber Physical Systems and Mobile Computing", Second Edition 2014.

[22] Bhambare R., Raut R., "A Survey on Digital Modulation Techniques for Software Defined Radio Applications", IRACST Communications Conference, Vol.3, No3, June 2013.

[23] Lynaugh K. and Davis T., "Software Defined Radios for Small Satellites", ReSpace/MAPLD 2010 Conference, New Mexico, November 2010.

[24] Castro T., "Antenna System Design for OLFAR's Inter-Satellite Link", University of Twente, October 2012.

[25] Patil J., Dubey B., Moudgalya K., Peter R., "GNU Radio, Scilab, Xcos, and COMEDI for Data Acquisition and Control: An open source alternative to LabVIEW", International Federation of Automatic Control, July 2012

[26] [Online]. Available :www.whatis.techtarget.com/Nyquist-Theorem

[27] Oliveri S., Aarestad J., Pollard H., Erwin R., "Modular FPGA Based Software Defined Radio for CubeSats", IEEE ICC-Selected Areas in Communications Symposium, 2012.

[28] David L., "CubeSats: Tiny Spacecraft, Huge Payoffs." [Online]. Available from http://www.space.com/308-cubesats-tiny-spacecraft-huge-payoffs.html,2004

[29] "University of Tokyo CubeSat Mission." [Online]. Available from http://www.space.t.u-tokyo.ac.jp/cubesat/mission/index-e.html, April 2011.

[30] "AAUSAT II CubeSat." Available from http://www.aausatii.space.aau.dk/eng, April 2011.

[31] Lyke J., Fronterhouse D., Cannon S., Lanza D., and Byers W.T., "Space Plug-in-Play Avionics," in 3rd Responsive Space Conference, (Los Angeles, CA), AIAA, April 2005.

[32] Ettus M. USRP User's and Developer's Guide. [Online]. Ettus Research LLC, 2005

[33] Rel Systems 2013, August 3. [Online]. Available: http://www.tethers.com/SpecSheets/Brochure_Swift_RelNav

[34] Muri P., McNair J., “A Survey of Communication Sub-Systems for Inter-Satellite Linked Systems and CubeSat Missions”, University of Florida Electrical & Engineering Department, Journal of Communications, Vol 7, April 2012.

[35] Kwadrat C., Horne W., “Inter-satellite Communication Considerations and Requirements for Distributed Spacecraft and Flying Formation Systems”, 2002

[36] Murakami B., Ohta A., “Self-Steering Antenna Arrays for Distributed Pico Satellite Networks”, University of Hawaii at Manoa.

[37] Ulversoy T., “Software Defined radio challenges and opportunities”, in IEEE Communications Survey & Tutorials, vols. 12, no. 4, 2010.

[38] Sun R., Maessen D., Guo J., Gill E., “Enabling Inter-Satellite Communication and Ranging for Small Satellites, Delft University of Technology”, Kluyverweg.

[40] Corgan J. [Online]. Available: http://www.gnuradio.org/redmine/projects/gnuradio

[41] Blossom E., USRP Hardware. [Online]. Available: http://files.ettus.com/manual/page_usrp2.html

[42] GNU Radio UHD Website http://code.etus.com/redmine/ettus/projects/uhd/wiki

[43] wxPython Toolkit description. [Online]. Available http://www.wxpython.org/what.php

[44] Haskins C., Millard W., Angert M., Adams N., Srinivasan D., “The Frontier Software Defined Radio: Mission Enabling, Multi Band, Low Power Performance”, 61ST International Astronautically Congress 2010, Paper 8922.

[45] Mitola J., “The Software Radio Architecture”, Communications Magazine, IEEE, Vol 33, 1995.

[46] Bose V., "The Impact of Software Radio on Wireless Networking", ACM SIGMOBILE Mobile Computing and Communications Review, Volume 3 Issue 1, January 1999, Pages 30-37

[47] Chen Z., Guo N., Qui R., "Demonstration of Real Time Spectrum Sensing for Cognitive Radio", IEEE, Volume 14, Pages 9150917, 2010

[48] Sarijari M., Marawanto A., Fisal N., Kamilah S., Rashid R., Satria M., "Energy Detection Sensing based on GNU Radio and USRP: An Analysis Study", IEEE 9th Malaysia International Conference on Communications 15-17, December 2009

[49] Rondeau T. (2012, February 2) GNU Radio Projects Explained. Retrieved March 1, 2015, Available: http://www.gnuradio.org/redmine/projects/gnuradio/wiki/AcademicPapers

[50] Seeber B. (2014 August 24)" Hacking Restaurant pagers, Spectrum Monitoring on the Road, Cracking RDS Encryption & Dominating the CyberSpectrum". Retrieved March 1, 2015, Avaialble: http://www.trondeau.com/grcon14-presentations/

[51] Nogueira, M., "The Benefits of Low Earth Oribiting Satellite Technology for the Internaitonal Community: Can the Potential be Realized", Indiana Journal of Global Legal Studies, Volume 5, April 1998.

[52] Guyader E., (2006 March 3) Galileo Atmospheric Data Enchanment Mission. [Online]. Avaiable: http://www.gsa.europa.eu/galileo-atomspheric-data-enhancement-mission.

[53] Sidibeh K., "Adapatation of the IEEE 802.11 Protocol for Inter-Satellite Links in LEO Satellite Networks", Surrey Space Centre, April 2008.

[54] Mobile WiMAX - Part I: A Technical Overview and Performance Evaluation. [Online]. Available: http://www.wimaxforum.org.

[55] Rodriguez J.A., Wallner S., Hein G.W., Eissfeller B., “A vision on new frequencies, signals, and Concepts for future GNSS systems”, Proceedings of the International Technical Meeting of the Institute of Navigation (ION GNSS), Fort Worth, Texas, 25-28 September, 2007.

[56] Camatel S. and Ferrero V., "Homodyne Coherent Detection of ASK and PSK signals performed by a Subcarrier Optical Phase-Locked Loop," IEEE Photonics Technology Letters, vol. 18, no. 1, pp. 142-144, Jan. 2006.

[57] Zhang D., Liu S., Yin M., “A Satellite Routing Algorithm Based on Optimization of Both Delay and Bandwidth, Key Lab of Broadband Wireless Communication and Sensor Network Technology”, Nanjing University, China, Sept 2011.

[58] Koskie S., Gajic Z., “SIR Based Power Control Algorithms for Wireless CDMA Networks: An Overview”, International Journal for Information and Systems Sciences, Vol. 4, No. 2, pp. 204-218, 2007.

[59] Geraniotis E and Ghaffari B., “Performance of Binary and Quaternary Direct-Sequence Spread-Spectrum Multiple-Access Systems with Random Signature Sequences,” IEEE Transactions on Communications, vol. 39, no. 5, pp. 713–724, May 1991.

[60] Cheng J and Beaulieu N., “Accurate DS-CDMA Bit-Error Probability Calculation in Rayleigh Fading,” IEEE Transactions on Wireless Communications, vol. 1, no. 1, pp. 3–15, January 2002.

[61] Satellite Communications and Space Weather [Online].

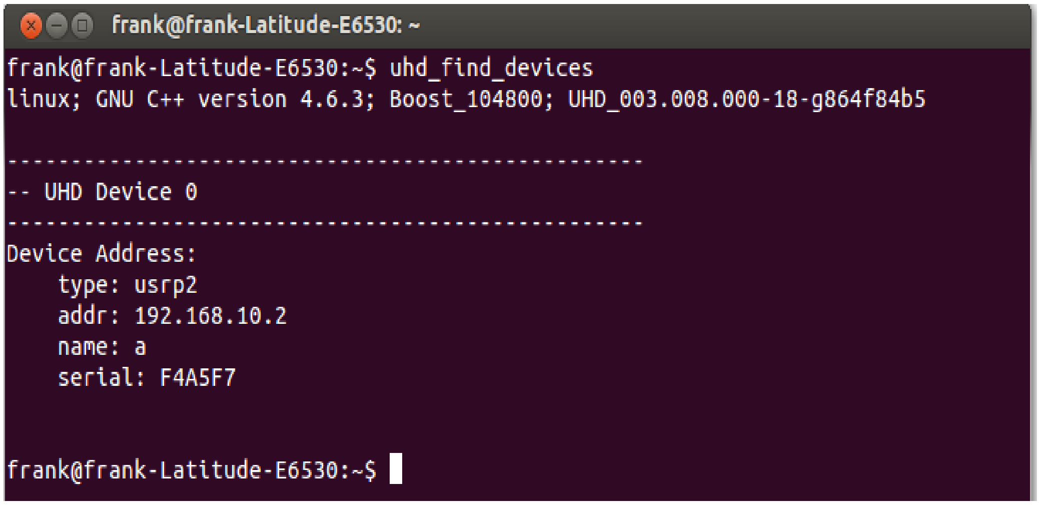
Available: https://www.ips.gov.au/Educational/1/3/2

Appendix A

GNU Radio Installation Guide

GNU Radio can be used on 4 different operating systems with Linux being the preferred option. One can install it on Linux, Windows, Mac O X, or from a bootable DVD with GNU Radio pre-installed. The method for our project was through Linux using Ubuntu 12.0.4. The first stage is to install Ubuntu onto its own hard drive partition to ensure full compatibility with GNU Radio. I suggest you visit the GNU Radio homepage [40] and scroll down to the download and build guide sections. From there download an ISO image for Ubuntu. The one we used in the project was an Ubuntu Desktop 64 bit. After download is complete, burn it to a DVD as an ISO image, insert the Ubuntu DVD into the computer's DVD drive for installation.

After the installation of Ubuntu the next step is to download and install Pybombs. Pybombs is the new GNU Radio install management system for resolving all the needed dependencies and creating you own custom blocks. The simple way to do is by opening up your terminal in Ubuntu by pressing and holding the "Ctrl, Alt, and T" keys. From here enter in the terminal "git clone git://github.com/pybombs/pybombs", press enter, then type "cd pybombs", press enter again and type "./pybombs install gnuradio". This operation installs the latest GNU Radio straight from GitHub into the pybombs directory. After the installation is finished you can create an environment file (located in $prefix/setup_env.sh) for your installation by running the command "./pybombs env". Source this file (replace $prefix with the prefix of your recently finished pybombs installation) with "source $prefix/setup_env.sh". Now you can run any GNU Radio tools from this shell simply by typing "gnuradio-companion &" from any terminal opened. Setting up the USRP N210 to communicate with the laptop requires connecting the gigabit Ethernet cable cord from the laptops Ethernet port to the USRP Ethernet port as well. The RF1 side represents transmission and reception of signal at the same time while the RF2 side represents only reception of signal. The USRP communicates at the IP/UDP layer and the default address for it is 192.168.10.2. In order to make it work with a laptop, the laptop should also be assigned an IP address in the same subnet. The procedure to this and if the user wants to change the default USRP address to their own can be seen in [41]. To ensure proper connection between the USRP and laptop is made, run this code “uhd_find_devices” from the terminal. Figure 23 of the USRP information or an image similar to it should be seen. This lets you know the USRP has been recognized and displays its information. The name, address, and serial of the USRP can all be changed in [41].



```
frank@frank-Latitude-E6530:~$ uhd_find_devices
linux; GNU C++ version 4.6.3; Boost_104800; UHD_003.008.000-18-g864f84b5

--------------------------------------------------
-- UHD Device 0
--------------------------------------------------
Device Address:
    type: usrp2
    addr: 192.168.10.2
    name: a
    serial: F4A5F7


frank@frank-Latitude-E6530:~$ 
```

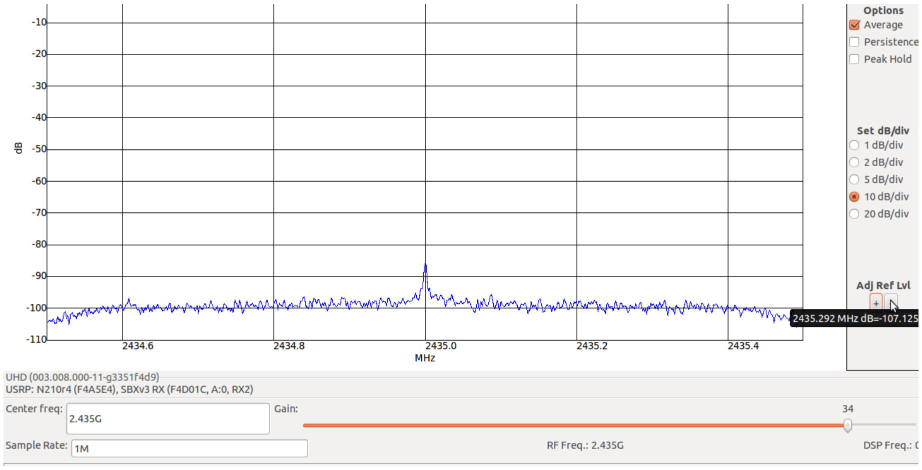
Figure 23. Command prompt output of the USRP device information.

Check the LED’s to see what each are doing. LED A represents transmission, LED B lets you know the MIMO cable link is connected, LED C is for receiving, LED D lets you know the firmware is loaded, LED E is the reference clock, and LED F shows the CPLD is loaded. The benchmarks where the uhd examples are located are in the cd/usr/local/share/gnuradio/examples folder.

Appendix B

OFDM Spectrum Testing

This test allows us to further verify the functionality of our hardware and the GNU Radio software. Just as before on the transmitter side and receiver we connect the VERT 2450 antennas to the port of the TX/RX on the USRP N210. GNU Radio comes with its own spectrum analyzer, which can be enabled from the GNU Radio Companion GUI or ran from the terminal. For this test we simply call the spectrum analyzer from the terminal with the command "uhd_fft –f 2.435G" on the laptop serving as the receiver. This command is a helpful tool for testing the receive functionality of a USRP radio. With the command it enables the SDR to prepare itself to receive a signal with a 2.4 GHz center frequency. Figure 24 below shows the spectrum of the received signal at 2.434 GHz with a sample rate of 1 M.



2435.292 MHz dB=-107.125

Options
☑ Average
□ Persistence
□ Peak Hold

Set dB/div
○ 1 dB/div
○ 2 dB/div
○ 5 dB/div
● 10 dB/div
○ 20 dB/div

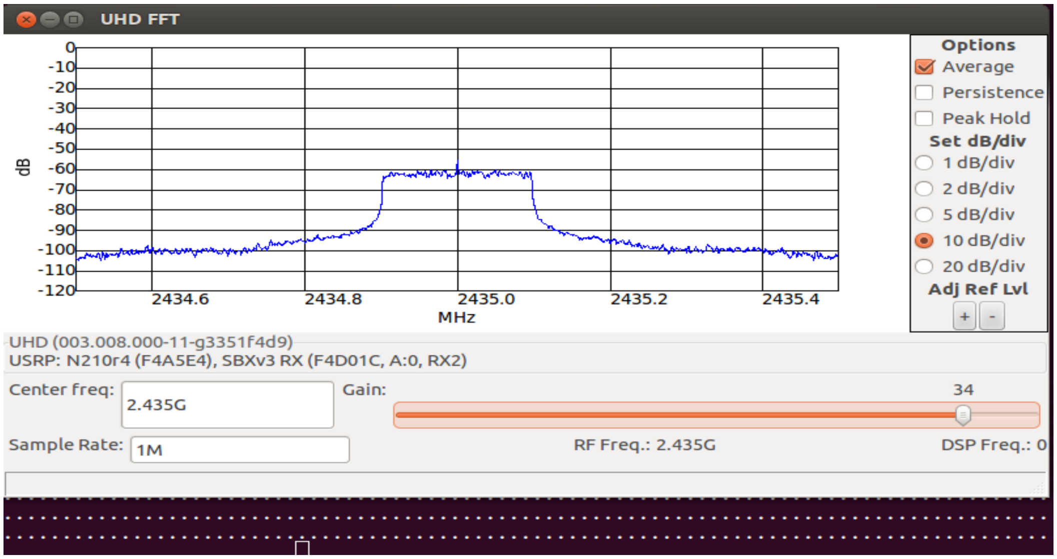
Adj Ref Lvl

UHD (003.008.000-11-g3351f4d9)
USRP: N210r4 (F4A5E4), SBXv3 RX (F4D01C, A:0, RX2)

Center freq: 2.435G
Gain: 34
Sample Rate: 1M
RF Freq.: 2.435G
DSP Freq.: (

Figure 24. Spectrum analyzer of a signal at 2.435 GHz.

From here, on the laptop serving as the transmitter, we input the follow command in the terminal, "./benchmark_tx.py –f 2.435g –m qpsk". This command prepares for the transmission of an OFDM signal with QPSK modulation to be received and viewed on the GNU Radio spectrum analyzer. After executing this command, Figure 25 shows the transmitted signal and verifies that the system can successfully transmit and receive signals with no issues.



Options
☑ Average
□ Persistence
□ Peak Hold
Set dB/div
○ 1 dB/div
○ 2 dB/div
○ 5 dB/div
● 10 dB/div
○ 20 dB/div

UHD (003.008.000-11-g3351f4d9)
USRP: N210r4 (F4A5E4), SBXv3 RX (F4D01C, A:0, RX2)

Center freq: 2.435G
Gain: 34
Sample Rate: 1M
RF Freq.: 2.435G
DSP Freq.: 0

Figure 25. OFDM received signal with QPSK modulation.