Design Document

**BPSK Modem with Convolutional Code**

Submitted To:

Professor Dennis Silage

Senior Design Project I and II

Temple University

College of Engineering

1947 North 12th Street

Philadelphia, Pennsylvania 19122

March 28, 2014

Prepared By:

PCG

C. Destin, B. Keith, & B. Thibodeau

*Programmable Communication Group*

Faculty Advisor: Dr. Dennis Silage

Temple University

College of Engineering

1947 North 12th Street

Philadelphia, Pennsylvania 19122

For further information, please contact Dr. Dennis Silage (email: silage@temple.edu).

|  |  |  |
| --- | --- | --- |
| **Team SD1-01** | **Programmable Communication Group** | **PCG** |
| **Team Members** | Cedric Destin, Brandon Keith, Brian Thibodeau | |
| **Advisor(s)** | Dennis Silage, PhD | |
| **Coordinator** | Thomas Sullivan, PhD | |
| **Department(s)** | Electrical and Computer Engineering | |
| **Project Title** | BPSK Modem with Convolutional Coding | |
| **Abstract** | Amateur radio satellite telemetry is the process of using the amateur radio frequency bands to transmit telemetry data from a miniaturized low-Earth orbiting satellite to a ground station. The most prevalent means of transmitting telemetry data down to Earth is not nearly as power-efficient as it could be. Inefficient power usage makes amateur satellite telemetry an expensive and esoteric hobby to get involved with. This senior design team aims to demonstrate how concatenated forward error correction (FEC) codes can make amateur satellite telemetry more power-efficient, and hence make the hobby more accessible to prospective amateur satellite operators. Specifically, we use FPGA hardware to implement a BPSK modem with a (2,1,7) convolutional encoder and Viterbi decoder. The performance of the proposed modem is evaluated using an AWGN core and a thorough budget link analysis. The comparison of BER and budget link results between traditional BPSK and FEC BPSK demonstrate the potential of the proposed solution | |
| **URL** | <https://sites.google.com/a/temple.edu/programmable-communication-group/> | |

**Executive Summary**

Table of Contents

[1. Problem 5](#_Toc379111470)

[1.1. Overall Objectives 5](#_Toc379111471)

[1.2. Historical and Economic Perspective 5](#_Toc379111472)

[1.3. Candidate Solutions 6](#_Toc379111473)

[1.3.1. Forward Error Correction: Block and Convolutional Codes 6](#_Toc379111474)

[1.3.2. Line Coding: Non Return Zero and Manchester 7](#_Toc379111475)

[1.3.3. Carrier Recovery: Squaring Loop and Costas Loop 8](#_Toc379111476)

[1.3.4. Clock and Data Recovery: Open Loop and Closed Loop Circuits 10](#_Toc379111477)

[1.4. Proposed Solution Concept 11](#_Toc379111478)

[1.5. Major Design and Implementation Challenges 11](#_Toc379111479)

[1.6. Implications of Project Success (Brandon) 12](#_Toc379111480)

[2. DESIGN REQUIREMENTS 13](#_Toc379111481)

[2.1. Functional Design Constraints (Brandon K) 13](#_Toc379111482)

[2.2. Non-Functional Design Constraints (Brandon) 14](#_Toc379111483)

[3. APPROACH 14](#_Toc379111484)

[3.1. Software Simulation Using Matlab/Simulink 14](#_Toc379111485)

[3.1.1. FEC-BPSK Modulator 15](#_Toc379111486)

[3.1.2. FEC-BPSK Demodulator: Carrier, Timing, and Data Recovery 16](#_Toc379111487)

[3.1.3. Forward Error Correction (Brandon) 24](#_Toc379111488)

[3.2. Hardware Implementation using ISE Project Navigator 31](#_Toc379111489)

[4. EVALUATION (Brandon) 32](#_Toc379111490)

[5. SUMMARY AND FUTURE WORK 32](#_Toc379111491)

[6. ACKNOWLEDGEMENTS 32](#_Toc379111492)

[7. REFERENCES 33](#_Toc379111493)

# Problem

## Overall Objectives

It has been shown that forward error correction can dramatically improve bit error rate performance (BER) in amateur packet radio satellite telemetry links (Hsiao, et. al, 2000). Additionally, it has been shown that binary phase shift-keying (BPSK) modulation is more reliable and bandwidth-efficient than audio frequency shift keying (AFSK) modulation (Hsiao, et. al, 2000). A quick survey of active amateur radio satellites reveals that many of them use FSK modulation and lack forward error correction capabilities. This senior design project aims to demonstrate how BPSK modulation with forward error correction (specifically convolutional coding) makes amateur radio satellite telemetry more accessible and reliable. Consequently, this senior design project advocates for improved robustness in amateur packet radio communication systems, specifically in those systems dealing with satellite telemetry.

Amateur packet radio satellite telemetry is often unidirectional and does not benefit from automatic repeat request (ARQ) like in other bidirectional amateur packet radio communications (Hsiao, et. al, 2000). In other words, if even one bit of an AX.25 telemetry packet is received in error, the entire packet is discarded and cannot be re-transmitted (Karn, 1994). This means that beacon signals from the amateur satellites must be transmitted with enough power to ensure that the embedded telemetry packet is received without error (de Milliano, et. al, 2010). BPSK modulation with convolutional coding outperforms AFSK modulation in terms of BER performance. This fact could benefit amateur radio satellite telemetry by improving network reliability and accessibility. The enhanced network reliability could lower overall power consumption in amateur telemetry satellites (de Milliano, et. al, 2010), resulting in two benefits: 1) reduced cost of satellite construction, and 2) making amateur telemetry satellites more technologically and financially accessible to amateur satellite operators by reducing the size, cost, and complexity of ground station antennas (Karn, 2011).

The ultimate goal of this senior design project is to demonstrate the improved network reliability (in terms of BER) and accessibility (in terms of link margin) that results from implementing BPSK modulation with convolutional coding in amateur packet radio telemetry satellites and ground stations.

## Historical and Economic Perspective

The standard digital modulation scheme used for amateur radio very-high frequency (VHF) and ultra-high frequency (UHF) operation is Bell 202 (Capitaine, et. al, 2010). Bell 202 provides AFSK modulation using 1200 Hz and 2200 Hz tones, with a resulting data rate of 1200 b/sec. It is typically used in the physical layer of the AX.25 data link layer protocol and this has been the case since the early 1980s (Karn, 1994). In 1984, when Bell 202 was a fairly new standard in the amateur radio community, Steve Goode, K9NG, performed an exhaustive bit error rate (BER) performance analysis of a standard Bell 202 modem (Goode, 1984). Goode found that 25 dB SNR at his FM receiver was necessary for high communication reliability. In other words, 25 dB SNR or higher was required to accurately receive 98% of incoming packets, which corresponded to a BER of 1.6e-5. Ralph Wallio, WORPK, figured out that with this BER, there is only a 1.603% chance of accurately receiving 117 consecutive 256-byte AX.25 packets (Wallio). Wallio concluded that “this is as Goode as it gets” and it is virtually impossible to get better results without error correction.

This poor reliability performance is not exclusive to amateur radio terrestrial communications. In 1995, it was demonstrated that error detection (not error correction) alone is not robust enough for amateur radio microsatellite communications (Hsiao, et. al, 2000). Particularly in unidirectional satellite communications, the harsh environmental conditions coupled with the microsatellite’s characteristically low transmitter power make for very unreliable telemetry data links (Hsiao, et. al, 2000). It has been demonstrated that forward error correction, specifically convolutional coding, can generally correct up to 75 percent of errors (Hsiao, et. al, 2000). It was also demonstrated that 1200 b/sec BPSK provides much more reliable transmission quality than 1200 b/sec AFSK, irrespective to whether the VHF or UHF amateur bands are used. Moreover, it was demonstrated that BPSK occupies a considerably smaller frequency bandwidth than AFSK while possessing excellent anti-interference properties. And with a general tenfold BER performance increase for both 1200 b/sec AFSK and BPSK over 144 MHz VHF, implementing forward error correction for amateur satellite telemetry was clearly demonstrated to be better than not implementing forward error correction.

In 2003, the AAU-Cubesat was one of the first pico-satellites to be launched into space. Moreover, the miniaturized satellite harbored a communication subsystem that implemented both forward error correction and interleaving over 9600 b/sec Gaussian minimum shift-keying (GMSK) AX.25 (Alminde, et. al, 2002). The enhanced robustness and data rate was justified by the fact that it had to transmit approximately 1461 kilobytes (kB) of telemetry and picture data per day. This simply would not have been possible had the satellite not utilized error correction. However, it operated at 437.9 MHz, which meant that a 2-meter radio could not receive its telemetry data. This would particularly bother Phil Karn, KA9Q, who is a strong proponent of making robust satellite telemetry links accessible to the average amateur radio operator (Karn, 2011). Karn asserts that robust telemetry links (using forward error correction) reduce the cost of satellite construction and simplify ground antennas, making amateur radio satellite telemetry much more technologically and financially accessible to amateur satellite operators (Karn, 2011).

As amateur satellite designers foresee the next generation of miniature satellites (de Milliano, et. al, 2010), and as the next generation of amateur satellites equipped with robust communication schemes continue to ascend into space, and as miniature satellites become increasingly more financially and technologically accessible to amateur satellite operators, it must be clearly demonstrated to the amateur radio community how these advancements trump the ubiquitous 1200 b/sec AFSK AX.25. Hence, to reiterate, this senior design project hopes to clearly demonstrate the performance advantages that yield from using BPSK modulation with forward error correction in amateur satellite telemetry.

## Candidate Solutions

Basic functions of a telemetry modem include baseband modulation/demodulation (line codes), passband modulation/demodulation, synchronization, and forward error correction. In the last century, many solutions have been proposed that trade performance in terms of bandwidth, transmission power and complexity. In this section we consider the following solutions:

1. Two forward error correcting codes - block codes and convolutional codes.
2. Two line codes – Non-Return to Zero (NRZ) and Manchester code.
3. Two carrier recovery circuits – Costas Loop and squaring loop.
4. Two timing and data recovery circuits – open loop and closed loop (Early-Late gate).

The subsequent sections will consider the costs and benefits of the aforementioned solutions.

### Forward Error Correction: Block and Convolutional Codes

Forward error correction (FEC) is a form of robust channel coding. It is used to correct errors that are injected into a digital communication link across a noisy propagation medium (channel). FEC codes fall into two general categories: block codes and convolutional codes.

NOTE: *At the time of writing this document, the Xilinx CORE Generator in Project Navigator ISE 14.6 only consists of one block coder/decoder pair and one convolutional coder/decoder pair. The block coding pair consists of a Reed-Solomon coder and decoder. The convolutional coding pair consists of a convolutional encoder and a Viterbi decoder.*

Deep space and satellite communication links are riddled with random errors across a very wide bandwidth (Nguyen, et. al, 2009). In addition to random errors in the satellite link, bursts of noise can corrupt an entire segment of a link resulting in burst errors (Murphy, et. al, 1994). Block codes are better suited for correcting burst errors while convolutional codes are better suited for correcting random errors (Viswanathan, 2013). A combination of block codes and convolutional codes, namely concatenating codes, are used in many systems to provide robustness against both kinds of errors (see Figure 1). This concatenating code consists of a coding chain and a decoding chain. The coding chain resides in the transmitter and consists of a Reed-Solomon encoder, followed by a block interleaver, then a convolutional encoder. The decoding chain resides in the receiver and undoes what the coding chain did. Namely, the decoding chain consists of a Viterbi (convolutional) decoder, followed by a block de-interleaver, then a Reed-Solomon decoder.



Figure 1. Top-level diagram for a concatenating code scheme consisting of a block code pair, an interleaving pair, and a convolutional code pair.

The propagation medium for space communications is modeled well by the AWGN channel (Viswanathan, 2013). Furthermore, it is understood that AWGN provides maximum bit corruption and compared to other channel models, systems that perform the best in AWGN perform the best in real-life situations (Viswanathan, 2013). Hence, this senior design project will rely solely on the AWGN channel to represent our propagation medium.

Figure 1 shows the propagation medium being modeled by the AWGN channel. The AWGN channel is a random noise channel, not a bursty noise channel. Being that convolutional coding excels at correcting random errors, it is logical that convolutional coding alone pairs well with the AWGN channel. Therefore, convolutional coding is the only forward error correction scheme used in this senior design project.

### Line Coding: Non Return Zero and Manchester

There are three criteria used for evaluating the performance of line codes: interference and noise immunity, bandwidth, and synchronization capabilities. These criteria are used for determining the appropriate line code that should be used in modems. NRZ is the most common line code as it appears naturally in digital logic. A binary one () is represented by a positive voltage while a binary zero () is represented by zero voltage. Conversely, Manchester line code represents a ‘’ by a transition from zero volts to a positive voltage during the second half of the bit period while a ‘’ is encoded as a transition from a positive voltage to zero volts during the first half of the bit period. Compared to NRZ, this means that Manchester code has two level transitions during one bit period while NRZ has only one. This presents a tradeoff between NRZ and Manchester in terms of synchronization and bandwidth. The two level transitions during each bit period means that the receiver can easily extract the transmitted clock to use for synchronization between the transmitter and receiver. The tradeoff is an increase in bandwidth due to the more frequent transitions, but how do these line codes perform in a noisy environment? The simple answer is that they perform the same. It can be shown that the energy, *E*, in NRZ and Manchester is , where is bit period. It can also be shown that each line code has a theoretical probability of bit error to be:

Since NRZ and Manchester share the same theoretical probability of bit error and contain the same energy, they both perform equally likely in a noisy environment. Thus the decision for choosing NRZ or Manchester resides in what is more important, bandwidth or synchronization? Considering our modem requires only a 1200 b/sec data rate, it was decided that synchronization was more important than bandwidth. Thus Manchester code was chosen as the line code in our modem.

### Carrier Recovery: Squaring Loop and Costas Loop

The modem demodulator is responsible for providing either coherent or non-coherent demodulation. Coherent demodulators require phase synchronization between the received signal and the locally generated oscillator. Conversely, Non-coherent demodulation does not require synchronization and makes no attempt to estimate the phase of the received signal. The advantage of non-coherent modulation is that it does not require additional hardware like phase-locked loops which are used to lock onto the incoming carrier phase (Feigin, 2002). However, the LEO-AMSAT’s we are interested in communicating with use BPSK for downlink and thus requires the design of a coherent demodulator.

The successful extraction of information from a received signal in a coherent demodulator requires both carrier and timing synchronization. Figure 2 illustrates the architecture of a typical coherent demodulator.



Figure 2. Received waveform takes two paths. First path extracts carrier for coherent demodulation and the second path recovers timing information. This architecture is based on the optimum binary receiver

The received signal from the transceiver is first processed by a band-pass filter to remove as much noise as possible and then sent to the carrier recovery circuit. Recovering the carrier is done in one of two ways, the squaring loop or the Costas loop. Each method utilizes phase-lock concepts and has its own advantages and disadvantages in terms of complexity and performance.

**Carrier Recovery using Squaring Loop**

The squaring loop is a popular choice for coherent demodulation of BPSK waveforms because it is mathematically easy to analyze and its hardware implementation is not as complex as the Costas loop. The squaring loop takes advantage of the fact that when the BPSK signal is squared, the phase offsets are removed leaving only a spectral component at twice the carrier frequency. A bandpass filter isolates the spectral component while also serving to remove any extraneous noise products resulting from the squaring. Following the band-pass filter, the signal is fed to a phase-lock loop (PLL) for phase and frequency tracking. Once the output of the voltage controlled oscillator (VCO) is locked in phase and frequency with the received signal, its frequency is divided by two. The resulting carrier is fed back to the correlator where it is mixed with the received BPSK waveform and the timing can be recovered (Nguyen & Shwedyk, 2009). The operation of the squaring loop is shown in Figure 3.



Figure 3. Squaring loop used for carrier recovery in the coherent demodulator. The Phase-Lock Loop utilizes feedback to track and lock onto in the received waveforms suppressed carrier

**Carrier Recovery using Costas Loop**

The second method for carrier recovery is the Costas Loop. Unlike the squaring loop whose only purpose is suppressed carrier reconstruction, the Costas loop is capable of synchronous data detection in addition to suppressed carrier reconstruction (Feigin, 2002). The received BPSK signal takes two paths in the Costas loop, the in-phase loop (top of Figure 4) and the quadrature loop (bottom of Figure 4). Unlike the PLL in the squaring loop, the Costas PLL uses an orthogonal carrier in the quadrature loop which allows direct demodulation of the BPSK signal without a need for squaring. The demodulated symbols are extracted from the in-phase arm once the loop is locked. This is illustrated in Figure 4.

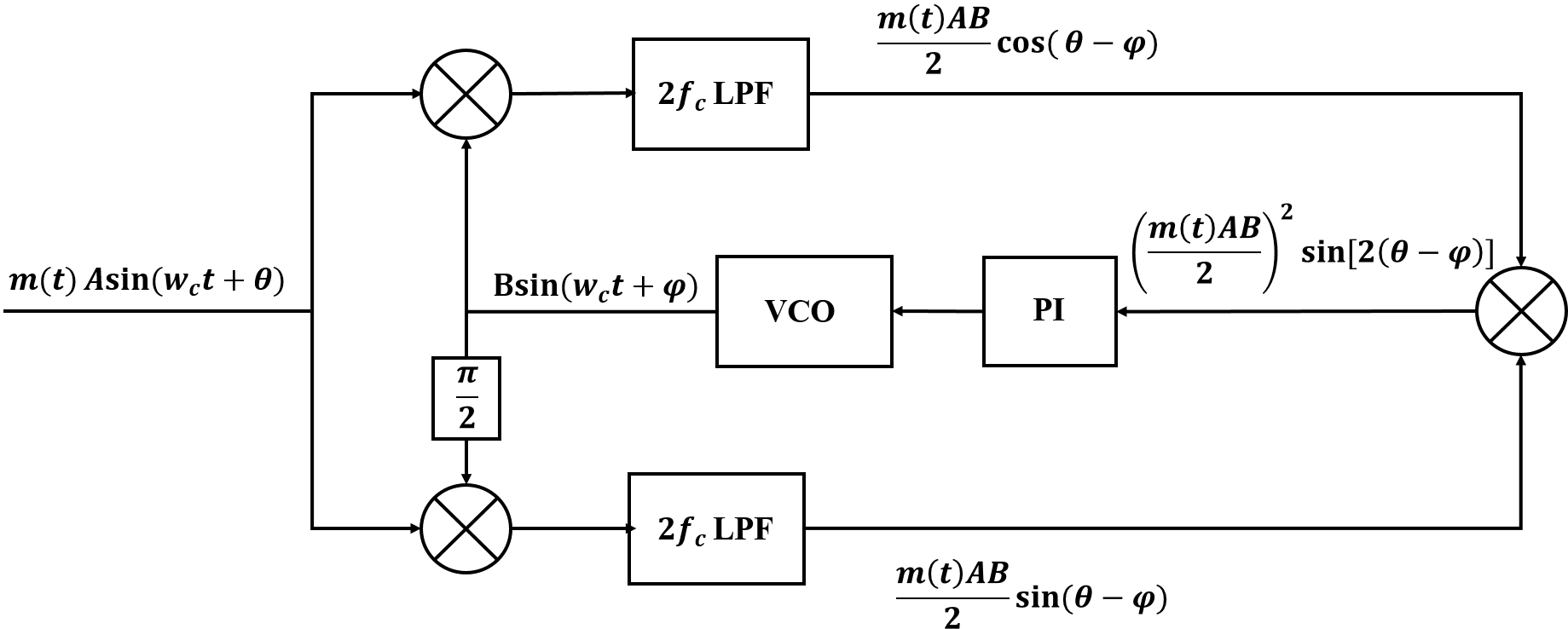


Figure 4. Costas loop used for suppressed carrier reconstruction as well as synchronous data detection.

Following a thorough analytical and experimental analysis of both the Costas loop and squaring loop in Simulink, it was decided that the Costas loop would be implemented for carrier recovery. The squaring loop demonstrated poor BER results as compared to the Costas Loop and the PLL proved unreliable to frequency steps and random 180 degree shifts. Alternatively, the Costas Loop demonstrated superior BER results and more reliable and robust carrier tracking to both phase and frequency steps (see appendix A). One of the downsides of the Costas loop is implementation of the arm filters. If these filters are not perfectly matched, then the loop’s performance is degraded. However, high speed digital circuits like FPGA’s allow the design and implementation of identical filters thus alleviating the problem.

### Clock and Data Recovery: Open Loop and Closed Loop Circuits

Timing recovery is the process of extracting a clock from the received signal so that the correct symbol determination can be made. The reason for this can be understood by recognizing that the local clock at the receiver is not synchronized with the transmitter clock and does not know when to sample the received data in order to make the correct symbol determination. In this section we consider two non-data aided architectures used for timing recovery. The first is an open loop architecture which is shown in Figure 10 (top) and the second is a closed loop architecture shown in 10(bottom). The closed loop circuit under consideration is also known as the Early-Late Gate.

In both methods it is assumed that the received signal is baseband and contains no spectral component. Thus the problem is similar to carrier phase recovery in BPSK. In the open loop method, a spectral component is created by delaying the received signal by one half a bit time and then multiplying it with the original received signal. The result of the multiplication produces a spectral component at a rate of Hz. Then a simple band pass filter can be designed to isolate the desired spectral component. Although simple to implement, the problem with the open loop method is that there is an average non-zero tracking error that reduces system performance (Nguyen & Shwedyk, 2009).

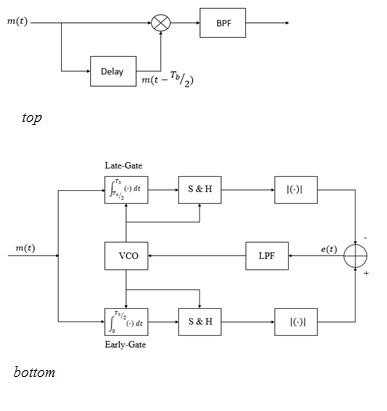


Figure 5. Open loop clock recovery circuit that produces a spectral component at 1/ .

The non-zero tracking error can be eliminated by using a closed loop timing circuit instead. The Early-Late gate is one closed loop circuit that requires three samples during each bit period. If *m(t)* is the received baseband signal from the correlator, then the early gate integrates and samples *m(t)* early while the late gate integrates and samples the *m(t)* late. The absolute value of the early and late samples are then compared to generate an error. The error signal drives a VCO which advances or retards the clock until the error is zero (Judd, 1996). When the error is zero, the output clock from the VCO is used to sample the received signal at the optimal time needed for correct symbol determination. Since the Early-Late gate synchronizer results in zero error, this method was chosen for timing recovery in our modems. A more detailed analysis of its operation is discussed in the Section 3, Approach.

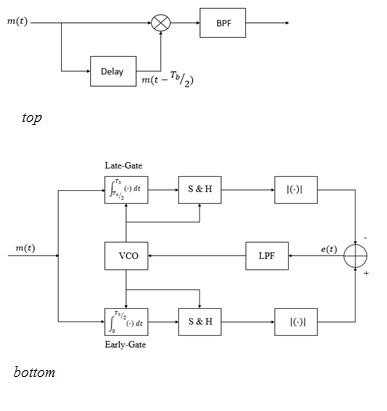


Figure 6. Architecture for the open- loop timing recovery (top) and the closed- loop (bottom)

## Proposed Solution Concept

This senior design project will consist of developing a software simulation (Simulink) and a hardware implementation (FPGA) of a 1200 b/sec FEC-BPSK modem (using soft-decision) with a (2, 1, 7) convolutional code. Synchronization in the demodulator is accomplished using the Costas Loop carry recovery circuit and the Early-late gate timing and data recovery circuit. We will gather measured BER performance data from the software simulation and the hardware implementation and compare it to the theoretical BER performance. When the measured and theoretical BER performances match closely, the software simulations and hardware implementations will be considered complete. The FPGA hardware implementation is expected to be the most important product of this senior design project.

In addition, we will perform a budget link analysis of our proposed modulation scheme along with two conventional Amateur Radio satellite modulation schemes – AFSK and BPSK. The purpose of the link analysis is to compare the accessibility (in terms of link margin) of Amateur Radio satellite telemetry links when AFSK, BPSK, and FEC -BPSK are received by a “less capable” ground station (like a FunCube satellite dongle).

## Major Design and Implementation Challenges

The BER performance analysis of our FEC-BPSK modem is dependent upon the accessibility of an AWGN channel. Although Simulink provides an AWGN channel for the software BER analysis, Xilinx does not currently support an AWGN IP core to be used for our hardware analysis. Xilinx has discontinued their AWGN Core v1.0 since the release of ISE Design Suite 11 (Xilinx, 2009).

The Costas loop carrier recovery circuit is expected to track and lock onto the transmitted carrier frequency under severe SNR conditions. Thus a major challenge is to design a robust Costas Loop that is capable of tracking changes to both phase and frequency offsets. This is traditionally accomplished using feedback control system concepts, but the Costas Loop (along with the conventional PLL) exhibit highly nonlinear behavior that make modeling a difficult task. However the Costas Loop can be linearized in terms of the phase using small angle approximations. It is important to note that the loop’s behavior to large phase and frequency offsets requires extensive simulation to ensure the desired response is achieved and stability is maintained. Similar design challenges exist for the Early-Late gate circuit as it is also a feedback control system.

The FEC ability of BPSK is intended to provide correction of random errors which may occur during transmission of data through the AWGN channel. Hence, a block encoder is expected to have an optimized code rate to enable the correction of the numerous burst of errors that may occur. Because of the complicated algebra involved, complexity in the design of the block encoder must be avoided, and such challenge poses an immediate constraint on the design project when designing the error correction. Similarly, the design of the block interleaver aims towards an interleaver with and optimized depth, (*d*.) Both of those parameters must also be considered at the receiver where decoding and de-interleaving occur to remove the coding done at the source. A very large depth will result in a slower computation at the receiver, hence, the depth must be optimized to optimize both the computation time at the receiver and the robustness of the system with respect to burst of noise and random noise.

## Implications of Project Success (Brandon)

It was hinted in Section 1.1 (Overall Objective) and Section 1.2 (Historical Perspective) that this senior design team has identified a problem within the amateur radio community. According to amateur radio operator Jeff Davis, KE9V, amateur radio has somewhat of a *lost future* (Davis, 2010). In the earlier half of the 20th century, amateur radio operators led the forefront of “discovery and experimentation” in the industries of electronics and communications. This was the case because many amateur radio operators were in fact professional electronics technicians and electronics engineers that designed and implemented the next wave of commercial and military communications. Oftentimes, the budding amateur radio operator, a *neophyte* if you will, would go on to become the next electronics repairman or electronics engineer. However, Davis highlights the fact that at some point in the past, the amateur radio community reached somewhat of a crossroads. Up to that point in time, the amateur radio community had pioneered Frequency Modulation (FM) communications over ultra-high frequency (UHF) and very-high frequency (VHF) operations, stationed repeaters throughout the land for long-distance over-air communications, and launched amateur radio satellites into the heavens which led to improved methods for space communications in addition to low-cost spacecraft manufacturing and launch. Davis highlights the fact that although the non-amateur world would go on to produce cellular technology, drastically improved over-air communications, and intelligent military digital communications, the amateur radio community as a whole decided to dwell in the past as the future marched ahead without it.

This senior design team identified one amateur radio operator and notable electrical engineer, Dr. Phil Karn, KA9Q, in his efforts to secure the future of amateur radio. Like Jeff Davis, Dr. Karn is also aware of amateur radio’s *lost future*. In a modem design article (Karn, 2011), Dr. Karn hints that making amateur radio communications more accessible to prospective amateur radio operators is one solution for securing the future of amateur radio. Specifically, in the design article, Karn identifies the fact that amateur radio satellite communications is mostly inaccessible to amateur radio operators because the equipment involved is too expensive and esoteric. Karn’s philosophy is that by making amateur radio satellite communication accessible to all amateur radio operators, school demonstrations will be more commonplace and consequently more kids will want to become amateur radio operators. It is implied that if more kids become amateur radio operators, or *hams*, amateur radio in general cannot have a *lost future*.

Hence, according to Phil Karn, one solution to securing the future of amateur radio is to make amateur radio satellite communications more accessible to kids. In order to make amateur radio satellite communications more accessible to kids, the amateur radio equipment involved in said communications must be less expensive and esoteric. By expensive and esoteric, Karn is referring to state-of-the-art software-defined radio systems and bulky antennas. This kind of equipment is regarded as being too inaccessible for the typical school demonstration of amateur radio satellite communications. Instead, Karn emphasizes the fact that a standard 2-meter single sideband (SSB) transceiver and an inexpensive antenna system should be all that is required at these school demonstrations. Satellite communications in general requires for relatively high-powered transmission of signals to overcome the high fading (energy loss) that results from an electromagnetic wave propagating through space (Sklar, 2001). In fact, free space attenuates an electromagnetic wave more than any other form of power attenuation along a satellite communication link. Hence, it is often the case that transmitted signals between amateur packet radio satellites and ground stations either deal with high transmission power to acquire a digital communication link with high data reliability or lower transmission power and low data reliability and link efficiency. It is understood that if you increase the reliability (BER) of a communication link, you can communicate with less capable ground stations (de Milliano, et. al, 2010). “Less capable” ground stations includes stations with small, portable SMA antennas and little USB modem dongles (like the FunCube dongle). Consequently, amateur radio satellite communications could become more *accessible* to prospective amateur satellite operators.

In a similar fashion as Phil Karn, KA9Q, and others (Hsiao, 2000), this senior design project aims to demonstrate that there are much more reliable digital communication schemes than are currently employed in most amateur radio satellites today. The intention of this senior design project is to provide concrete evidence that BPSK modulation with forward error-correcting codes can make amateur radio satellite communications more reliable and hence, more *accessible* to prospective amateur satellite operators. Perhaps budget link analyses of popular and prospective communication schemes, like showcased in this senior design project, would further persuade an amateur satellite designer to employ more reliable communication schemes in the ascending fleet of miniaturized amateur radio satellites.

# DESIGN REQUIREMENTS

As discussed in the section 1.4 (Proposed Solution Concept), our analysis will require the design and implementation of a 1200 b/sec FEC-BPSK modem with a (2, 1, 7) convolutional code. The functional requirements of this system is summarized in Table 1 in section 2.1 followed by a summary of each. Non-functional requirements are listed in Table 2 in section 2.2.

## Functional Design Constraints (Brandon K)

|  |  |
| --- | --- |
| **Name** | **Description** |
| Data Rate | The convolutional encoder is required to receive user data at a rate of 1200 b/sec.  The convolutional encoder is required to transmit FEC data at a rate 2 x 1200 = 2400 b/sec.  The BPSK modulator is required to receive serialized FEC data at a rate of 0.5 x 2400 b/sec = 1200 b/sec.  The BPSK demodulator is required to deliver 3-bit soft-decision data at a rate of 3 x 2400 b/sec = 7200 b/sec.  The Viterbi decoder is required to receive 3-bit soft-decision at a rate of 3 x 1200 = 3600 b/sec.  The Viterbi decoder is required to deliver user data at a rate of 1200 b/sec. |
| Symbol Rate | The BPSK modulator is required to operate at 2400 symbols/sec.  The BPSK demodulator is required to operate at 2400 symbols/sec. |
| Line Coding | Manchester Encoding is resilient to small changes in Frequency due to Doppler shift and provides easier clock extraction during demodulation. |
| Modulation/ Demodulation | BPSK modulation is done non-coherently and BPSK demodulation is done coherently. 3-bit soft-decision encoding is implemented in the output of the demodulator. |
| Operating Frequencies | The BPSK modems will modulate the data using a 4800 Hz carrier. |
| BER Performance  (Eb/N0 dB)  (over AWGN) | 0.5 @ 0 dB  0.0783 @ 1 dB  0.0071 @ 2 dB  4.28e-4 @ 3 dB  1.74e-5 @ 4 dB  4.40e-7 @ 5 dB |
| Forward Error Correction | This modem will use the (2, 1, 7) convolutional code. Viterbi decoding will operate in 3-bit (fixed point) soft-decision decoding mode. |
| Interface(s) | Modified 19-bit AWGN core (Verilog)  BERT: RS-232 (1200 b/sec, 1 parity bit, 1 stop bit, no flow control) |

Table 1. Functional design constraints for the all three systems.

## Non-Functional Design Constraints (Brandon)

These non-functional design constraints are based off of the Trenz Electronics Micromodule Spartan-6 XCSLX45-2CSG484I FPGA development board (product#: TE0630-00I) and Trenz Electronics Demo Carrier Board for Industrial Micromodule TE0300/TE0630 (product#: TE0304-00).

|  |  |  |
| --- | --- | --- |
| **Type** | **Name** | **Description** |
| Economic | Cost | FPGA Board: $204.64, Carrier Board: $81.03 |
| Environmental | Temperature | FPGA Industrial grade: -40° C to 100° C |
| Environmental | Power Consumption | Carrier Board: USB bus power supply |
| Manufacturability | Dimensions | Carrier Board: 115 x 79 mm |
| Manufacturability | Weight | N/A |

Table 2. Non-functional design constraints for test board.

# APPROACH

## Software Simulation Using Matlab/Simulink

Simulink provides a graphical design environment for rapid prototyping and simulation of the various subsystems and circuits required by the FEC-BPSK modem. Each system, subsystem, and circuit was designed using a black box approach. Not only does this allow seamless integration of our individual designs, but it also helps organize the design modularly to be carried over to FPGA. In addition, Simulink provides the tools and block sets necessary to evaluate the performance of our system under a variety of conditions. Of particular importance is the evaluation of BER performance in AWGN and phase and frequency shifts. These are conditions that all modems are expected to able to handle.

The remainder of the section is organized as follows, section 3.1.1 will consider the Simulink design of the Manchester encoder and decoder subsystems. In section 3.1.2, the modulator subsystem is discussed. Then in 3.1.3, the demodulation subsystem of BPSK is discussed by examining Costas Loop carrier recovery circuit and the Early-Late gate timing and data recovery circuit.. Each modem has slightly different requirements for the PLL and thus its application will be examined for FSK and BPSK. Closely related to PLL is the early-late gate synchronizer which is used for timing recovery of both modems. Its Simulink designed will also be discussed in section 3.1.3. Lastly, section 3.1.4 will conclude the Software simulation section by examining interleaving forward error correction and how it can be used to reduce the BER of the BPSK modem.

### FEC-BPSK Modulator

In PSK, each bit corresponds to a distinct phase of a sinusoidal carrier. For BPSK, these phases are chosen to be 0 and 180 degrees with the transmitted signal, represented mathematically by equation (6).

Then exploiting the fact that, the expression for the transmitted BPSK can be re-written as:

where is the Amplitude and is the carrier frequency of the transmitted BPSK signal. From equation (7), the Simulink model of the BPSK modulator was designed to modulate a 2400 b/sec NRZ data stream by gating two antipodal sinusoidal carriers with amplitude, and a carrier frequency, . This implementation is illustrated in Figure 7.

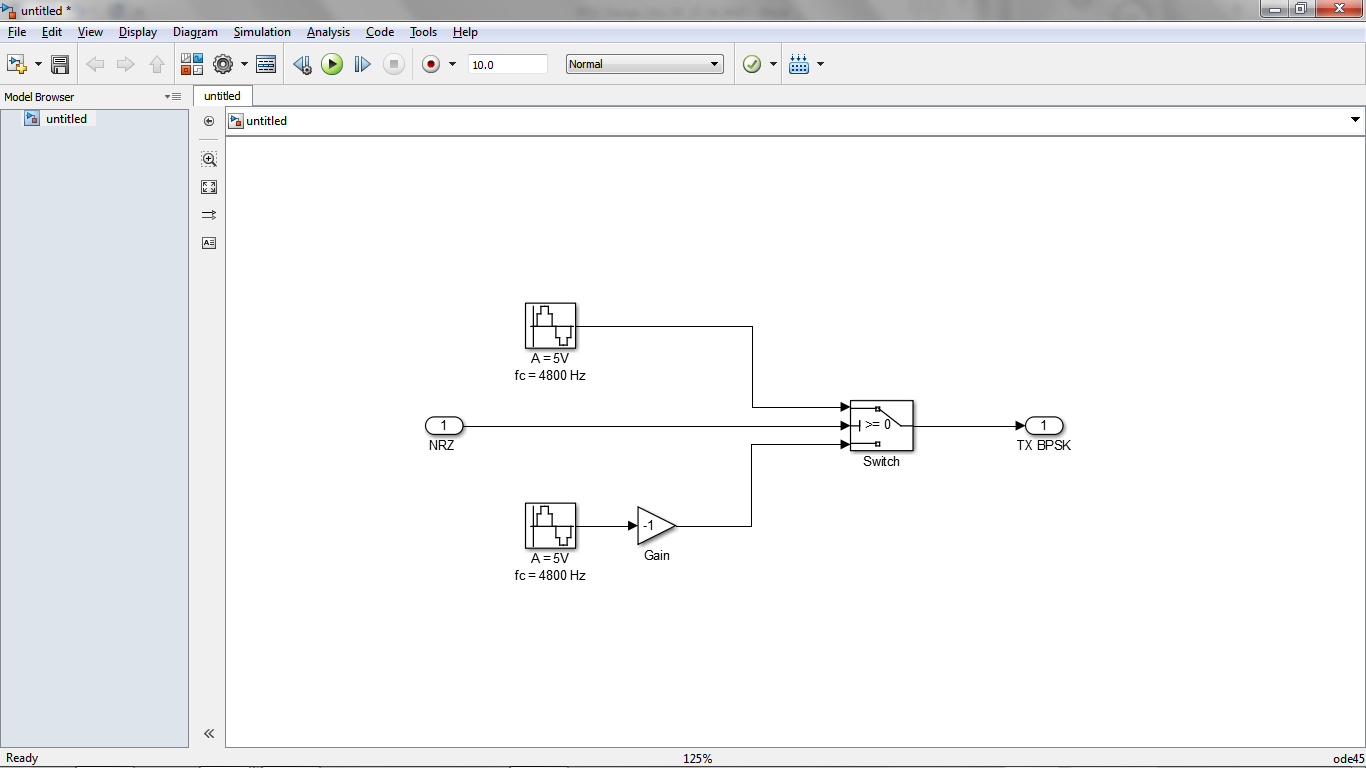
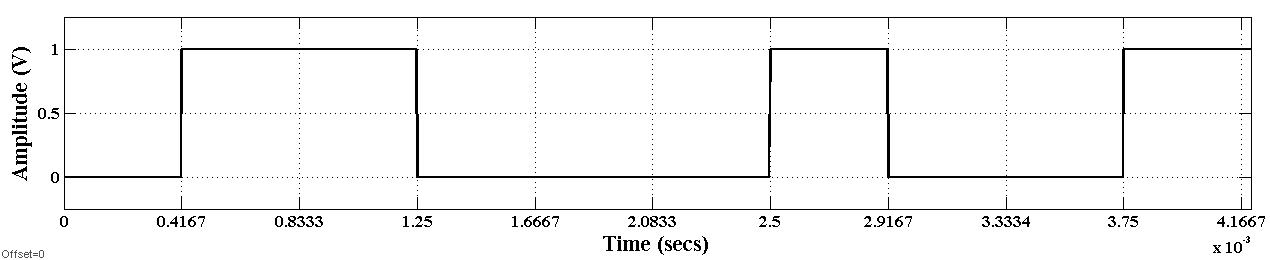


Figure 7. BPSK modulator that uses the Manchester data stream to gate two antipodal sinusoidal carriers that result in the BPSK modulated signal.

Figure 8 illustrates the operation of the BPSK modulator. When a binary ‘1’ is input to the modulator, the positive sine wave is transmitted and when a binary ‘0’ is input, the negative sine wave is transmitted.



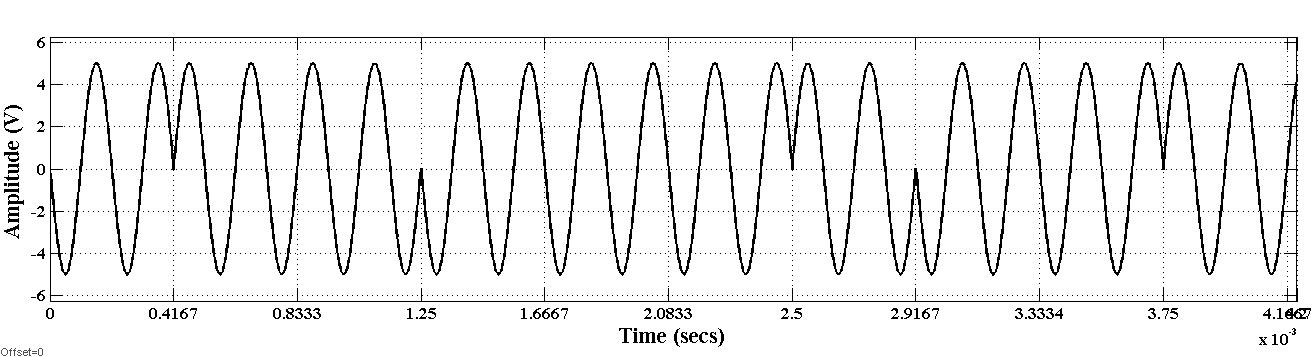


Figure 8. The transmitted BPSK signal has 180 degree phase shifts that correspond to logic level transition of the NRZ data stream.

### FEC-BPSK Demodulator: Carrier, Timing, and Data Recovery

Coherent demodulation of BPSK requires the receiver to be synchronized with the transmitter. As discussed in Section 1.3, synchronization requires two circuits’ independent circuits. The first is the Costas loop carrier recovery circuit used for suppressed carrier reconstruction and data demodulation and the second is the Early-Late used clock and data recovery. In this section we consider the theory, operation, and design of both circuits.

**Costas Loop Design for Coherent BPSK Demodulation**

The operation of the Costas Loop is explained using simple trigonometry and is illustrated in Figure 9. For simplicity, we assume the received signal is free of noise. This is only to illustrate loop operation.

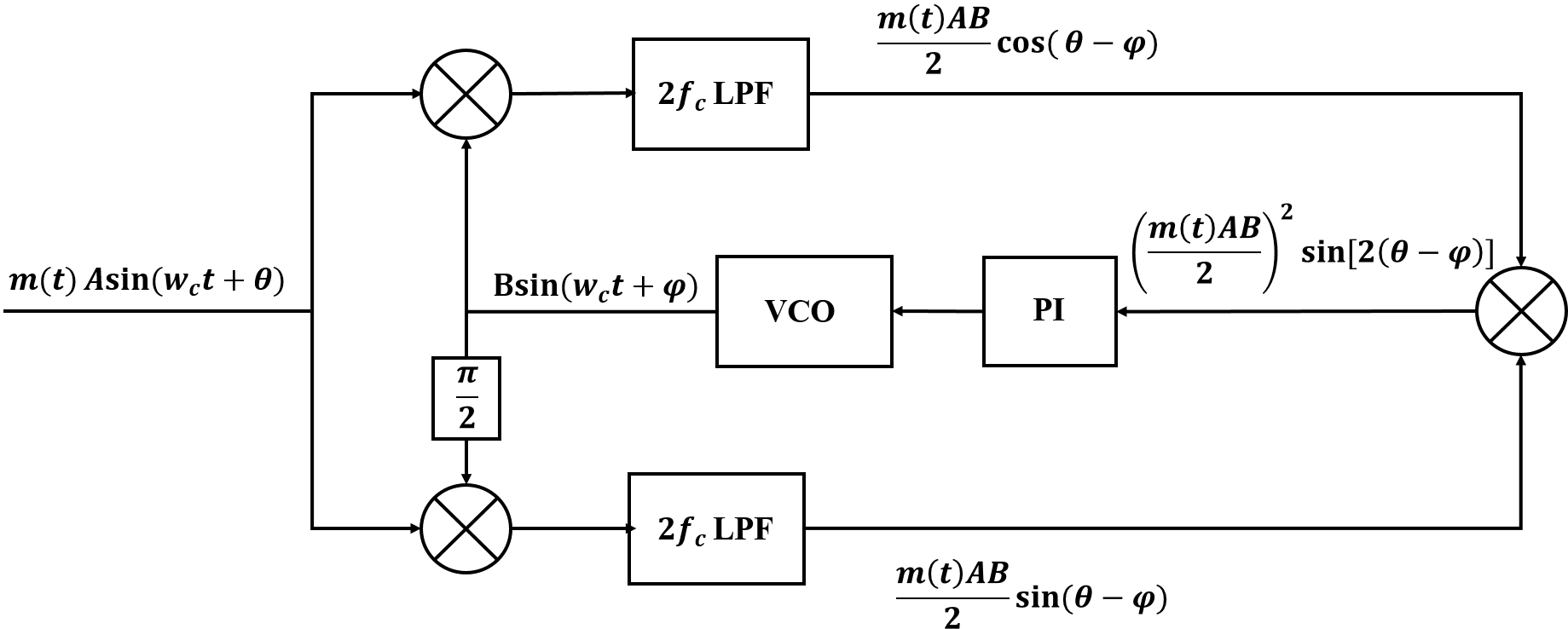


Figure 9. Time domain model of the Costas Loop carrier recovery and demodulation circuit. It’s operation is described using trigonometric identities.

The received signal, is the product of the data, and the modulating carrier with phase, . This is equivalent to equation (3) where .

, (4)

Referring to Figure 9, the received signal is split between the in-phase and quadrature arms where it is multiplied with the output of the local oscillator with amplitude, and phase, . Thus the output of the multiplier in the in-phase arm is:

While the output of the multiplier in the quadrature branch is:

Note that the VCO in the bottom arm is shifted by to produce the quadrature component. Low pass filters (LPF) with cutoff frequencies at remove the double frequency leaving only the cosine and sine of the phase difference.

The phase doubler on the right side of the Costas Loop multiplies the output of the in-phase and Quadrature LPF’s. The resulting error signal, , drives the VCO towards phase lock.

Assuming the VCO is locked in phase with the carrier of the transmitted BPSK signal (), then the output of the in-phase LPF from equation (7), reduces to leaving behind only a scaled version of the originally transmitted data. In addition, the error signal from the phase doubler is reduced to zero implying the receiver is locked in phase with the received signal, .

However, fundamental questions arise like how long will it take the loop to lock? What is the lock range of the VCO? We can define specifications for the Costas Loop by applying concepts from classical control theory. Small angle approximations make it possible to linearize the Costas Loop in terms of phase. Figure 10 presents the Laplace domain model of the loop where it is assumed that .



Figure 10. Linearized Costas Loop in the Laplace domain. Note that the filter is omitted in the linearized Laplace model because it is assumed that the double frequency component was filtered leaving only the phase error

The open loop gain is modeled by which includes the amplitude of the received signal and the gain of the arm filters. represents the Loop filter (PI filter) that adjust the error in order to increase or decrease the time to lock. The VCO is modeled by an integrator that accumulates the phase and is the VCO gain. The closed loop transfer function of the Costas Loop in Figure 10 is thus given by,

And the steady error transfer function is,

In order for the PLL to be capable of tracking a phase and frequency step, the appropriate loop filter *L*(s)must be chosen such that steady error transfer has zero error for step () and ramp () inputs. This is accomplished by application of the final value theorem to equation (15).

The results of the final value theorem tell us that the loop filter must contain an integrator so that the error at infinity goes to zero. Two common loop filters that accomplish this are the lead-lag filter and the proportional integral filter (PI). In this design, the PI filter was chosen which takes the form,

Where is the proportional gain and is the integral gain. After substituting the expression for the loop filter into the closed loop transfer function given by equation (18), the new closed transfer function describing the Costas Loop’s behavior is,

Recognizing that equation (19) is a prototypical second order transfer function, the PLL transfer function can be re-written in terms of the natural frequency and the dampening ratio (Crawford, 2008).

Where,

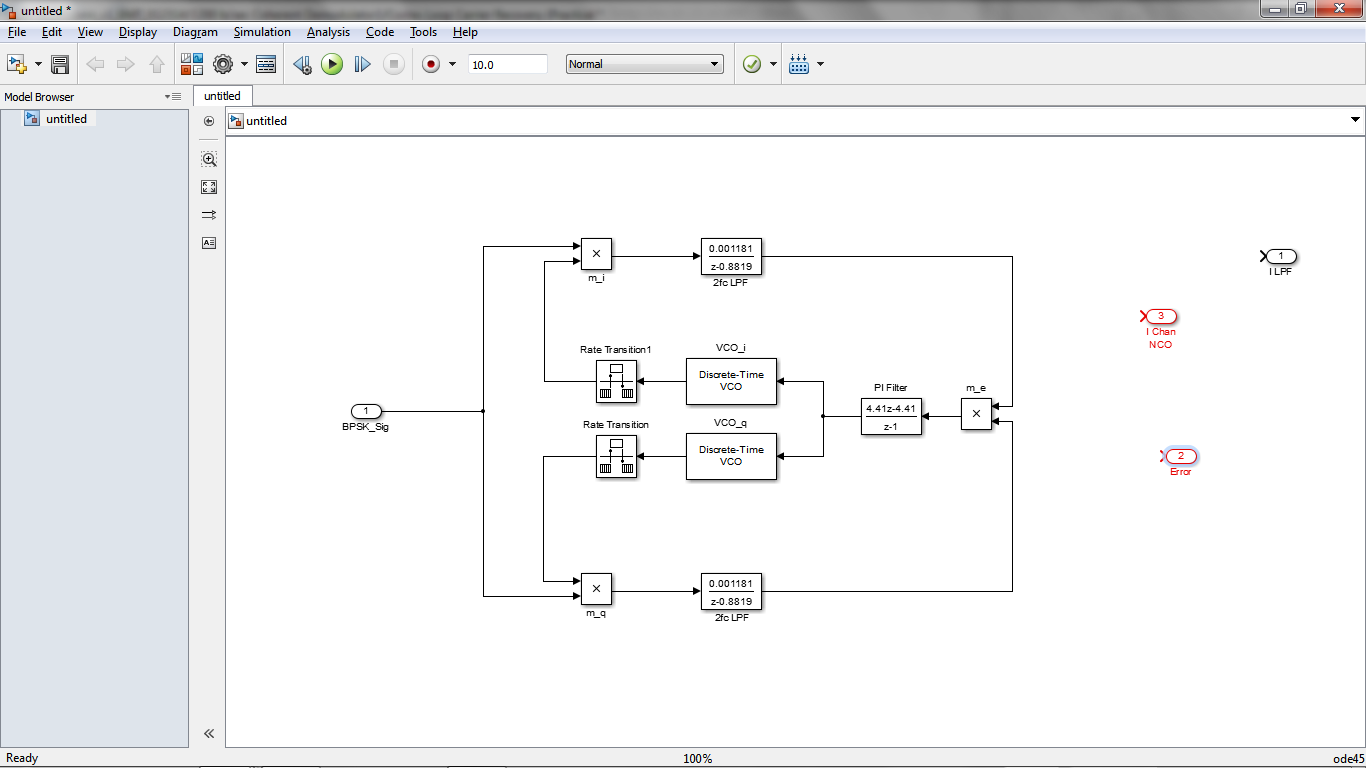


Figure 11. Simulink architecture of the Costas Loop

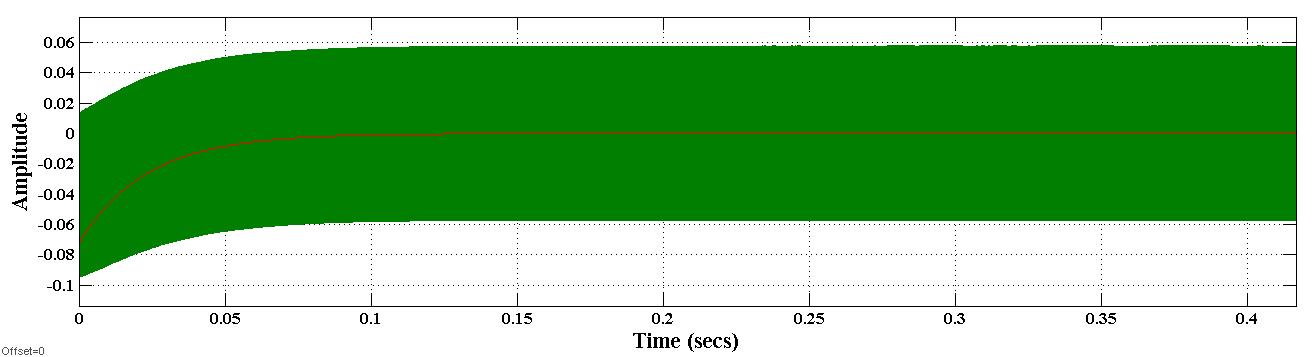


Figure 12. Simulated error of the linear model (red) and the Simulink model (green)

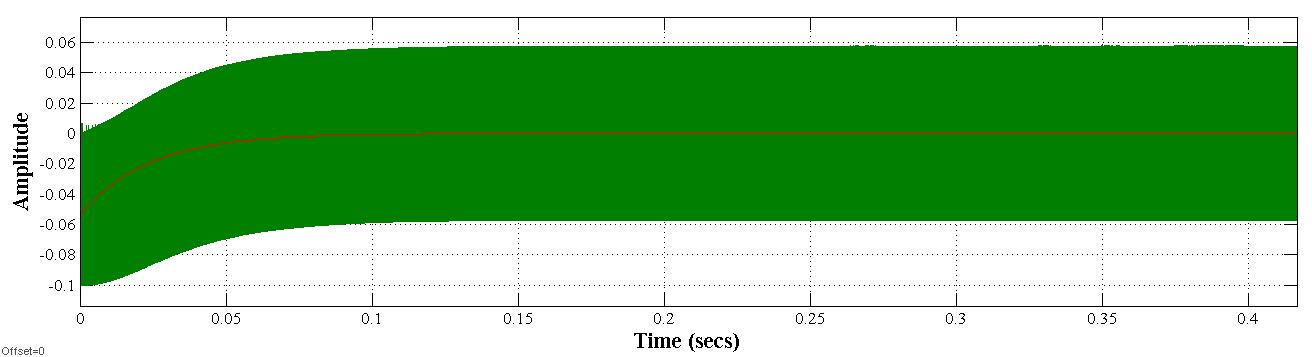


Figure 13. Simulated error of the linear model (red) and the Simulink model (green)

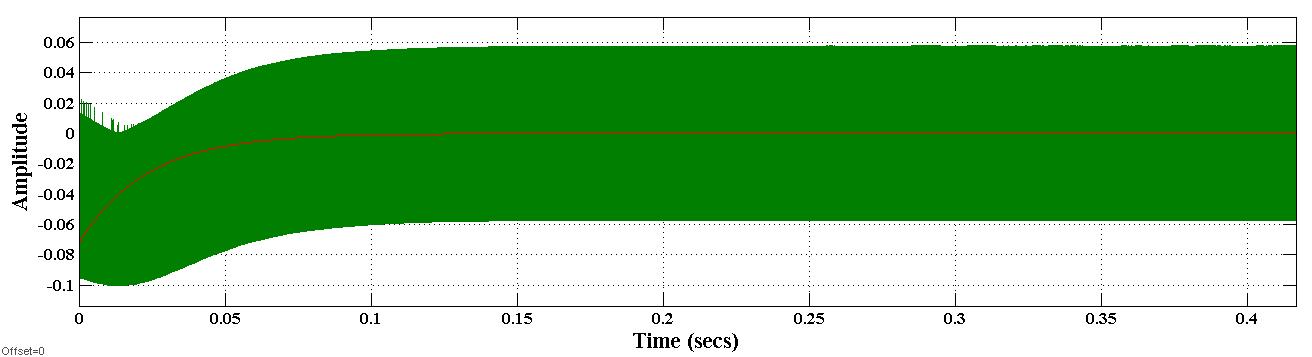


Figure 14. Simulated error of the linear model (red) and the Simulink model (green)

**Timing Recovery Using Early-Late Gate Synchronizer**

The synchronization between the transmitter and the receiver is done using the Early-Late Gate Timing Recovery circuit, among the several other circuitry available, the Early Late Gate has proven to be the ideal design in communication links. The Early-Late Gate provides the opportunity to constantly ameliorate synchronization through its feedback components and its loop filter. As section 1.3.4 explains, the error between the early clock and the late clock is to converge to zero in ideal conditions. For the error to converge to zero at a certain time, the Early Late Gate depends on its loop filter component in circuitry. This component is used to minimize the settling time of the circuit to synchronize the received signal. Therefore, based on those facts, it was found beneficial to implement a PID controller for achieving an appropriate settling time. Figure 6, would then consist of a PID controller for its loop filter, while the remaining components will remain the same.

On purpose of obtaining a zero error rate from the Early Late Gate, the early and late clocks must operate within a certain region which will allow the integrators to integrate an equal amount of energy from the two branches. This region is commonly referred to as the *correct receiver timing* while the *early/ late* *receiver timing* are known to detriment the error between the early and late gates. Figure 15 illustrates the desired condition for the three samples to operate in, note that the area integrated by the integrators will be identical. Before it locks itself into the *correct timing* the circuit first adjusts its clock using the PID controller. The controller, makes use of the Proportional and Integral gains to drive the output clock from the error signals. The gains PI controller can be found in table 3.

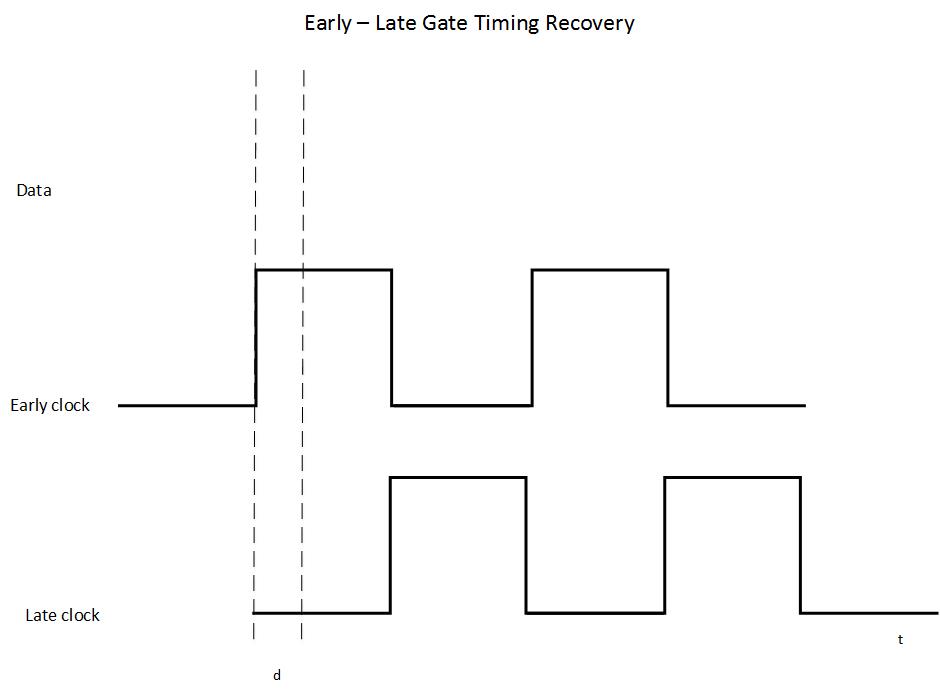


Figure 15. Ideal timing between the Early and Late branch

|  |  |
| --- | --- |
| P | I |
| 0.15 | 0.25 |

Table 3. Proportional and integral values for the Early-Late Gate’s PID controller

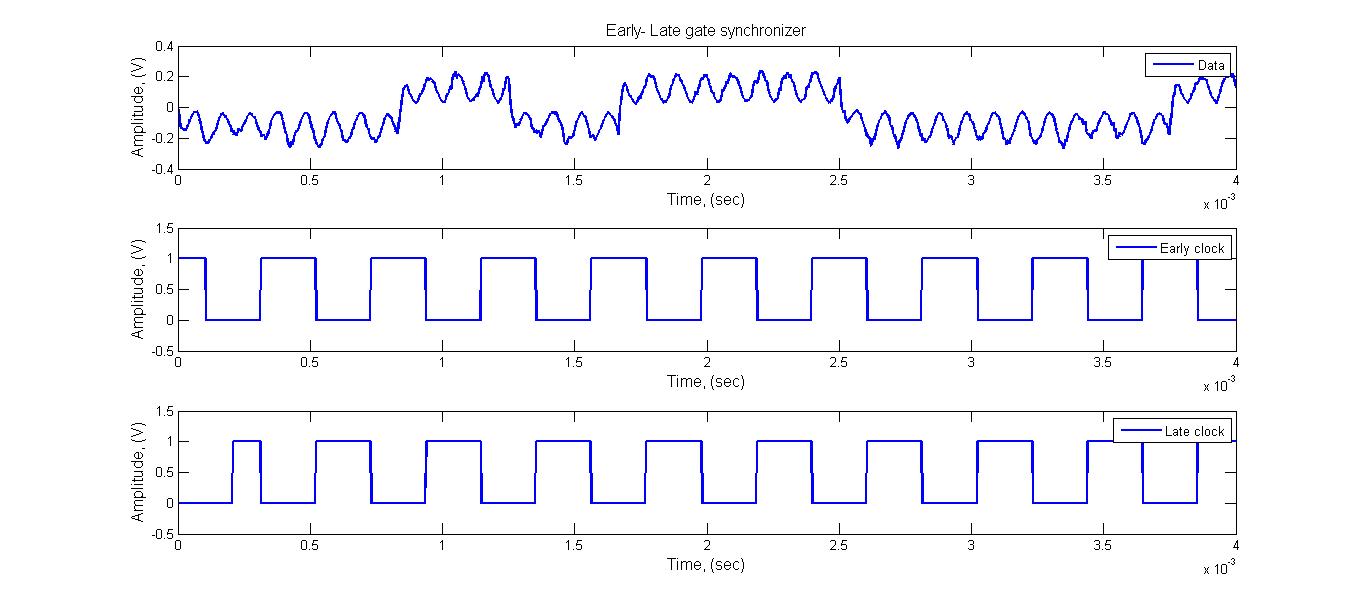


Figure 16. Early Late Gate operating within correct receiver timing

Figure # depicts the Early-Late Gate’s implementation in Matlab Simulink, since the signal is sampled at a rate of 480E+3, the demodulated signal will be composed of 200 samples hence, a delay of 100 samples on the clock (at the late gate) is equivalent to a half-bit delay and is then used to control the *integrator* and the *sample and hold blocks* of the late branch. And finally, the Voltage Controlled Clock (VCC) is implemented with a traditional VCO that is later converted into a square wave using a bang-bang controller.

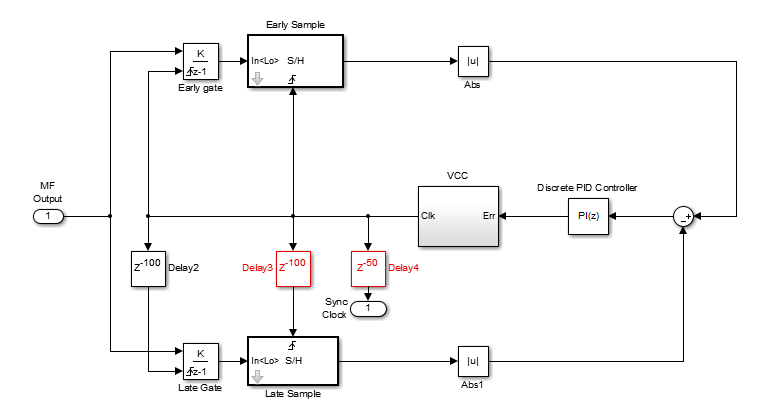


Figure 17. Simulink implementation of the Early- Late gate timing synchronizer

Within the *correct timing region,* the error between the late and early branch is expected to be zero as Figure 16 illustrates, both branch will integrate different region of the waveform, but eventually, the amount of energy will sum to the same quantity. The error is then obtained by the subtraction of the two signals, the error is kept small by the use of the PID controller which drives the clock to lock unto the edges of the signal. Figure 18 shows the error rate of the Early- Late Gate circuit, by first observation, it can be seen that the energy at the two signals are the same except for very small instances where one branch peaks more than the other. By tuning the PID controller, those instances are weighted accordingly to drive to the VCC providing the synchronized clock.

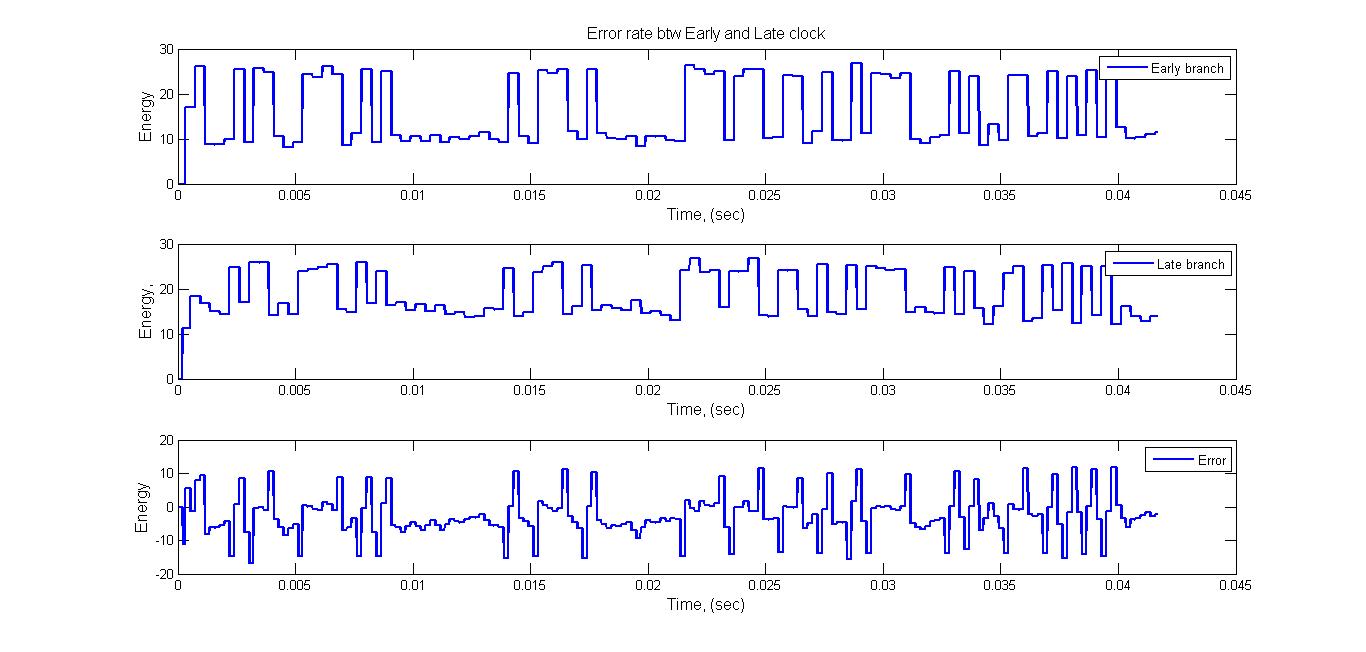


Figure 18: Error rate between the Late and Early branch

### Forward Error Correction (Brandon)

The third digital communication system to be analyzed is System C – this system comprises everything in System B plus forward error correction (FEC) and a modification to the receiver. Convolutional coding is used for FEC because of its favorable abilities to correct random errors. The modification to the receiver includes making the BPSK demodulator implement soft-decision decoding instead of hard-decision decoding. Like FEC itself, soft-decision decoding provides a beneficial *coding gain.* An explanation of this will be presented shortly as we describe System C in a clockwise fashion (see Figure 19), starting with the *Convolutional Encoder* and looping around to the *Viterbi Decoder*.

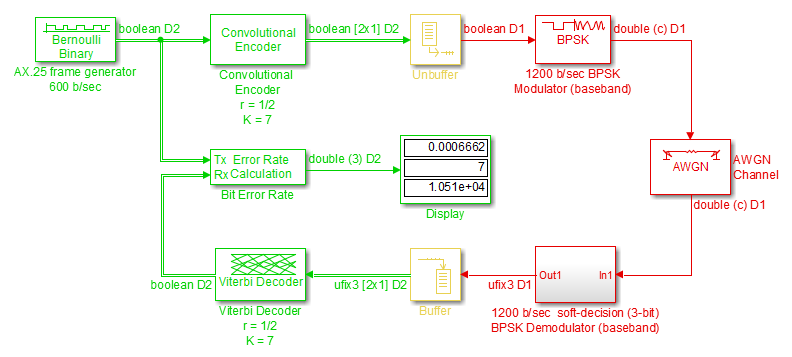


Figure 119. A system-level diagram of System C. A convolutional encoder and Viterbi decoder are used to implement convolutional coding (½ code rate, constraint length 7). Besides from the addition of forward error correction, System C differs from System B in that the BPSK demodulator at the receiver implements 3-bit soft-decision instead of hard-decision.

Figure 19 shows sample time coloring to show that there are different sampling times throughout the model. The *AX.25* *Frame Generator* outputs bits at a rate of 600 b/sec. Hence, the green represents a sampling time of 1/600 seconds. The *Convolutional Encoder* outputs twice as many bits as the *AX.25 Frame Generator*, but its output is a 2-bit signal. Hence, the sampling time after the *Convolutional Encoder* is still 1/600 seconds (green). The *Unbuffer* is used to serialize the 2-bit signal, hence the sampling rate used after the *Unbuffer* is 1/1200 seconds (red). This represents the 1200 b/sec data rate that is used by the *BPSK Modulator*. The *AWGN Channel* and *BPSK Demodulator* both use this 1/1200 second sample time, therefore those blocks are colored red. However, the output of the *BPSK Demodulator* is a 3-bit bus. This 3-bit bus does not need to be serialized, therefore no change in sample time is needed. However, the soft-decision *Viterbi Decoder* requires two 3-bit outputs from the *BPSK Demodulator* in order to execute its error-correcting procedures. Hence, we revert back to the 1/600 second sample time. This means that two 3-bit words are provided to the soft-decision *Viterbi Decoder* at a rate of 600 words/sec. The output of the *Viterbi Decoder* is a 600 b/sec serial data stream. Lastly, the bit error rate compares two 600 b/sec data streams, so the *Bit Error Rate* block and the *Display* block are both colored green. Figure 40 summarizes this information.

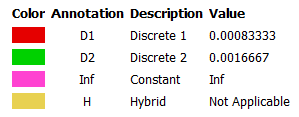


Figure 20. The sample color chart for the system-level Simulink model of System C.

**Convolutional Encoding**

The convolutional encoder holds *k* bits of input data (*k-*bit message word) to generate an *n*-bit output. A constraint length *K* represents how many *k-bit* message words are used to process the *n-bit* output of the convolutional encoder. This would be regarded as a (*n*, *k*, *K*) convolutional encoder. The convolutional encoder is composed of a shift register with *K* *k-bit* stages and *n* modulo-2 adders. As an example, Figure 21 shows a (2, 1, 7) convolutional encoder. **In fact, the (2, 1, 7) convolutional code will be used solely throughout this senior design project.** The (2, 1, 7) convolutional encoder comprises *K = 7* stages (or, *K – 1* = 6 *k*-bit delays) in its shift register and *n* = 2 modulo-2 adders. Each stage of the shift register holds *k* = 1 bits. The *code rate* of a convolutional encoder is *k/n*, so the code rate for this encoder is ½.

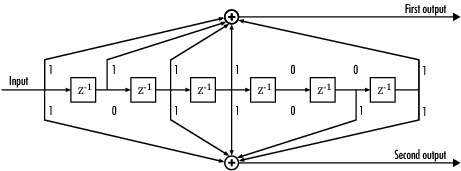


Figure 21. (Courtesy of Matlab®) A convolutional encoder (1/2 code rate, constraint length 7).

At each unit of time, *k* bits will shift to the next stage in the shift register, and *k* bits will shift into the first stage of the shift register. There are *K* stages for a group of *k* bits to shift into before it eventually shifts out of the shift register. At each unit of time, each of the *n* modulo-2 adders are sampled and these *n* bits are outputted by the convolutional encoder. In Figure 21, the top modulo-2 adder provides the first output bit and the bottom modulo-2 adder provides the second output. These two bits combined are the output of this convolutional encoder. Hence, one bit shifts into the encoder and two bits are produced by the encoder. The connections between the shift register stages and the modulo-2 adders, or the *generation matrices*, characterize the convolutional encoder. In other words, some permutations of connections have better error-correcting capabilities than other permutations of connections (Sklar, 2001).

The *trellis diagram* is widely used for showing the possible outputs of a convolutional encoder. However, in the case of the (2, 1, 7) convolutional encoder, the trellis diagram would compose 64 (2K-1 = 27-1) states. Consequently, it would not be practical or possible to display the full trellis diagram in this report. However, as a basic example, let’s see how the convolutional encoder (shown in Figure 41) would process the following:

In the convolutional encoder: 11010002

1st output: (12 + 12 + 02 + 12 + 02) % 2 = 12

2nd output: (12 + 02 + 12 + 02 + 02) % 2 = 02

2-bit output: 012

The *poly2trellis* function is used by Simulink to generate the functionality of the *Convolutional Encoder*. Hence, the command **poly2trellis(7, [171 133])** was entered in the *trellis structure* parameter field. The argument is interpreted as a convolutional encoder with constraint length 7 and whose generation matrices (or shift register connections) are described in octal code. The upper generation matrix is 1718 and the lower generation matrix is 1338. This can be verified by studying the connections shown in Figure 21.

**1200 b/sec BPSK Modulator**

Figure 19 shows a BPSK modulator block used to generate a 1200 b/sec BPSK modulated baseband signal. The serialized output of the convolutional encoder is used as the modulating signal. Hence, the input of the BPSK modulator is a digital bitstream (0 and 1) and the output of the BPSK modulator is a baseband signal with amplitudes +1 V and -1 V. The phase offset used by the BPSK modulator is 0°.

Baseband signals can replace passband signals here because the baseband optimum receiver is essentially configured the same as the passband optimum receiver (Silage, 2009).

**AWGN Channel**

Figure 19 shows the *AWGN Channel* being used to model random noise that pollutes satellite communication signals. Figure 42 shows the configurations of the *AWGN Channel* block. The signal-to-noise ratio is specified in units of the ratio of energy per bit to noise power spectral density *Eb/No (*dB)*.* This field uses the constant parameter *EbNo* to set the desired SNR value in dB.

The *Convolutional Encoder* is configured as a ½ rate encoder. For every one bit, the encoder adds another two bits. To accommodate this, and add the correct amount of noise, the *Eb/No (dB)* parameter of the AWGN block is effectively halved by subtracting 10\*log10(2).

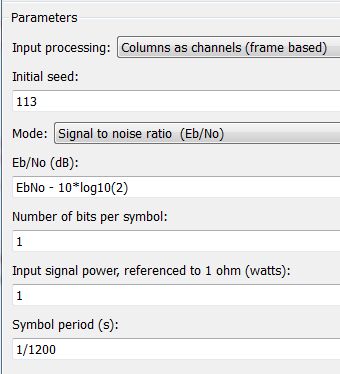
**

Figure 22. The configuration for the *AWGN Channel* block used in the

system-level Simulink model of System C.

Since BPSK modulation is used, a single bit represents a symbol. Additionally, since we are dealing with a 1 V baseband signal, the nominal input signal is 1 W. Since BPSK modulation is used, the symbol period is the same as the bit period (1/1200 seconds). Figure 23 is a time capture showing 10 bits being transmitted (yellow) and received (red).

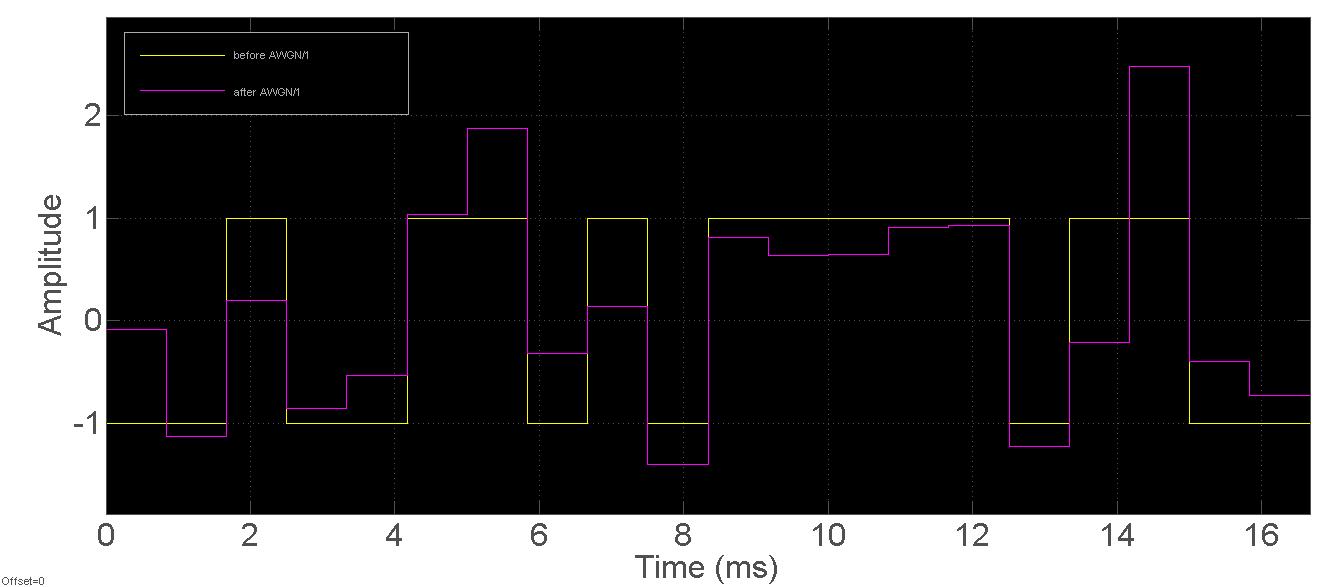


Figure 23. Output of the 1200 b/sec BPSK baseband signal through AWGN. The yellow signal represents the transmitted signal and the red signal represents the received signal (after AWGN).

**1200 b/sec BPSK Demodulator**

The BPSK demodulator is used to demodulate the noisy baseband signal output of the AWGN channel. The output of the BPSK demodulator is essentially the output of the matched filter (integrate and dump) filter in the bit recovery process. However, instead of implementing hard-decision decoding and producing a logic 0 or 1 as output, the output of the matched filter is scalar quantization (SQ) encoded into a 3-bit signal. The range of the 3-bit output of the SQ encoder represents the confidence level of the demodulated signal actually being a logic 0 or 1 (shown in Table 4). This indecisiveness justifies the term “soft-decision” decoding. Figure 22 shows the circuit used to implement this 3-bit scalar quantization encoding.

|  |  |
| --- | --- |
| **Input Value** | **Confidence** |
| 0 | Most confident zero |
| 1 | Second most confident zero |
| 2 | Second least confident zero |
| 3 | Least confident zero |
| 4 | Least confident one |
| 5 | Second least confident one |
| 6 | Second most confident one |
| 7 | Most confident one |

Table 4. This table shows the output of the *Scalar Quantization Encoder* which is used in the system-level Simulink model of System C.

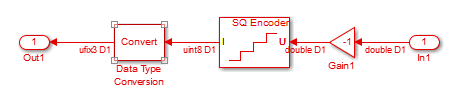


Figure 22. Scalar Quantization Encoder scalar quantizes BPSK demodulator output into a 3-bit value. This 3-bit “soft-decision” output is used by the *Viterbi Decoder.*

**Viterbi Decoding**

The soft-decision Viterbi decoder uses Hamming distance and maximum likelihood to decode the convolutional encoding. The ½ code rate convolutional code with constraint length 7 has a *free distance* *df* of 10 (Sklar, 2001). Essentially, free distance is the minimum Hamming distance between the all-zero path and any arbitrary path that diverges and remerges with the all-zero path (Sklar, 2001). It is essentially a measure of how well a convolutional code corrects errors. The higher the free distance, the better the convolutional code is at correcting random errors. A Viterbi decoder can correct up to *t* random errors (Sklar, 2001):

(18)

Hence, the ½ code rate convolutional code with constraint length 7 can correct up to 4 random errors in a received signal. The upper-bounded probability of bit error for the (2, 1, 7) convolutional code with BPSK and soft-decision decoding (Sklar writes ‘hard-decision decoding’; Brandon claims this is incorrect) is expressed as follows:

(19)

This upper-bounded probability of bit error is related to an upper-bounded coding gain (dB) expressed as follows (Sklar, 2001):

where the code rate *r* is:

The result of the coding gain inequality means that BPSK with soft-decision Viterbi decoding (‘2,1,7’ code, ½ code rate, constraint length 7) offers at most 7 dB of coding gain compared to un-coded BPSK.

The *Viterbi Decoder* uses a *traceback length* of 48, which is within the recommended range for a convolutional code with constraint length *K* = 7 (Sklar, 2001). The traceback length is a metric for path memory since the convolutional code depends on both currently inputted data and recently inputted data.

**Simulation Results**

One of the main goals of this senior design project is to implement BPSK modulation with soft-decision Viterbi decoding via both computer simulation (Simulink) and hardware implementation (FPGA). From the computer simulation models, we are able to determine if the observed BER performance results correlate with theory. If the results do compare with theory, the next goal would be to implement the models in FPGA hardware. The overall objective would be to assess the power consumption savings when using BPSK modulation with soft-decision Viterbi decoding versus just BFSK (Bell 202) modulation or BPSK modulation.

The following graph (Figure 43) shows the BER performances for several targeted amateur satellite telemetry communication schemes. The graph shows that there is approximately a 6 dB improvement to the perceived SNR when using BPSK instead of BFSK. The results of simulation for BPSK with soft-decision Viterbi decoding provides a maximum of 7 dB coding gain when compared to un-coded BPSK. This result matches with the theoretical coding gain inequality provided in the previous section. Additionally, there is approximately 2 dB of coding gain when using soft-decision Viterbi decoding instead of hard-decision Viterbi decoding (Viswanathan, 2013). The curve for BPSK with hard-decision Viterbi decoding satisfies this fact by performing 2 dB worse than soft-decision Viterbi decoding.

These BER performance results serve as the basis for determining how much power consumption is reduced by opting to use BPSK with soft-decision Viterbi decoding instead of just BFSK (Bell 202) or BPSK in amateur radio satellite telemetry communications. Namely, these results provide a basis for a *budget link analysis*, which is a procedure used to assess the overall power budget of a digital communication system. Obviously, in our case the digital communication system is an amateur satellite communications over AWGN. Moreover, it is important to determine the optimal balance point where we benefit from a generous amount of coding gain while achieving lower SNR than with un-coded BFSK or BPSK. That is, BPSK with soft-decision Viterbi decoding achieves 5.8 dB of coding gain at 11.3 Eb/N0 versus un-coded BPSK. Our job would then be to determine if a higher coding gain (upper-bounded to 7 dB reaching out to infinite SNR) is worth achieving at the expense of high power consumption.

Figure 23. BER performance for several amateur satellite telemetry communications schemes over AWGN.

**Budget Link Analysis (1200 b/sec BFSK)**

**Budget Link Analysis (1200 b/sec BPSK)**

**Budget Link Analysis (1200 b/sec BPSK with hard-decision Viterbi decoding)**

**Budget Link Analysis (1200 b/sec BFSK with soft-decision Viterbi decoding)**

## Hardware Implementation using ISE Project Navigator

Deep space and satellite communication links are riddled with random errors across a very wide bandwidth (Nguyen, et. al, 2009). In addition to random errors in the satellite link, bursts of noise can corrupt an entire segment of a link resulting in burst errors (Murphy, et. al, 1994). These channel imperfections are common in satellite communications and are modeled very well by the additive white Gaussian noise (AWGN) channel (Viswanathan, 2013). The AWGN channel is a random noise channel that makes a communication link vulnerable to random bit errors and burst errors. In general, it is understood that AWGN provides maximum bit corruption and compared to other channel models, systems that perform the best in AWGN perform the best in real-life situations (Viswanathan, 2013). Hence, this senior design project will rely solely on the AWGN channel (see Section 1.3.6) to represent the propagation medium for our three amateur radio satellite telemetry systems.

We implement the bit error rate tester (BERT) in software. The BERT consists of an AX.25 packet generation program written by us, a custom AX.25 packet comparison program written by us, and an available virtual serial terminal interface (with data logging capabilities). The BERT provides several performance metrics based off of bit error rate (BER) and packet error rate (PER). Please refer to Section 3 (Approach) for the implementation of this BERT and how it interfaces with the external FPGA board.

# EVALUATION (Brandon)

# SUMMARY AND FUTURE WORK

# ACKNOWLEDGEMENTS

# REFERENCES

Alminde, L., Bisgaard, M., Vinther, D., Viscor, T., & Ostergaard, K. Z. (2002). *Robustness of radio link between AAU-cubesat and ground station.*Unpublished manuscript.

Capitaine, T., Barrandon, L., Bourny, V., Senlis, J., Le Mortellec, A., Astier, R., et al. (2010). Robust satellite AX25 frames demodulation. Paper presented at the *The Small Satellites Systems and Services (4S) Symposium, , 31.*

Crawford, J. A. (2007). *Advanced phase-lock techniques*. Norwood, Mass.: Artech House.

Davis, J. (2010). Amateur radio's lost future. Message posted to <http://www.amateurradio.com/ham-radios-lost-future/>

de Milliano, M., & Verhoeven, C. (2010). Towards the next generation of nanosatellite communication systems.*Acta Astronautica, 66*(9–10), 1425-1433. doi:<http://dx.doi.org.libproxy.temple.edu/10.1016/j.actaastro.2009.10.034>

Feigin, J. (2002, January 1). Practical Costas Loop Design. *RF Signal Processing*, *NA*, 20-26. Retrieved September 14, 2013, from http://defenseelectronicsmag.com/site-files/defenseelectronicsmag.com/files/archive/rfdesign.com/images/archive/0102Feigin20.pdf

Goode, S. (1984). BER performance of TAPR TNC modem.*Packet Status Register,*(11), 14-15.

Hsiao, F., Liu, H., & Hsieh, S. (2000). Using forward error correction technique for microsatellite data broadcasting system.*Acta Astronautica, 46*(2–6), 203-211. doi:<http://dx.doi.org.libproxy.temple.edu/10.1016/S0094-5765(99)00224-6>

Judd, D. (1996). Data synchronization simulation using the MATHWORKS Communications Toolbox. *1996 IEEE International Conference on Communications*, *2*, 706-710.

Karn, P. (1994). *Toward new link layer protocols.* Retrieved 11/01, 2013, from <http://www.qsl.net/n9zia/newlinkpaper.html>

Karn, P. (2011). *The BPSK1000 telemetry modem for ArriSSat-1.* Retrieved 11/01, 2013, from <http://www.ka9q.net/bpsk1000.html>

Lindsey, W., & Simon, M. K. (1977). Detection of digital FSK and PSK using a first-order phase-locked loop.*Communications, IEEE Transactions on, 25*(2), 200-214. doi:10.1109/TCOM.1977.1093804

Magliacane, J. (1997). Digital OSCAR Communications. *Digital OSCAR Communications*. Retrieved December 2, 2013, from http://www.amsat.org/amsat/articles/kd2bd/Digital/

Murphy, J., Chow, E., & Markley, R. (1994). ATM service-based selective retransmission over DSN satellite links. Paper presented at the *15th AIAA International Communication Satellite Systems Conference, San Diego, USA, , 350.*

Nguyen, H., & Shwedyk, E. (2009). *A first course in digital communications* (1st ed.). Cambridge, UK: Cambridge University Press.

Proakis, J. G. (2008). *Digital communications* (5th ed.). Boston: McGraw-Hill.

Rao, K. H. S., Jamadagni, K. S., Von Allmen, L. A., & Shah, A. V. (1990). All-digital pseudo-coherent (PC) FSK modems. Paper presented at the *Digital Communications, 1990. Electronic Circuits and Systems for Communications. Proceedings, 1990 International Zurich Seminar on,*463-473. doi:10.1109/DIGCOM.1990.129390

Roelofs, E. (1987). a fsk modem for the map protocol.*Report of the Practical,*

Silage, D. (2009). *Digital communication systems using MATLABÂ® and SimulinkÂ®*. Gilroy, CA: Bookstand Pub..

Sklar, B. (2001). *Digital communications: Fundamentals and applications* (2nd ed.). New Jersey: Prentice Hall.

Viswanathan, M. (2013). In Mathuranathan V. (Ed.), *Simulation of digital communication systems using matlab* (2nd ed.). GaussianWaves: Viswanathan, Mathuranathan;.

Wallio, R. *AX.25 packet radio AFSK on FM Bit, packet and data set error rates.* Retrieved 11/07, 2013, from <http://showcase.netins.net/web/wallio/BER_Packetradiobiterrorrate.html>

Zicari, P., Corsonello, P., & Perri, S. (2008). A high flexible Early-Late Gate bit synchronizer in FPGA-based software defined radios. *4th European Conference on Circuits and Systems for Communication*, *NA*, 252-255.

1. Product SPECIFICATION
2. SOME INTERESTING RELEVANT DERIVATION