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

December 2, 2013

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 re-create the power-inefficient satellite link (1200 b/sec AFSK over AWGN), we then change its basic baseband modulation scheme to improve power-efficiency (1200 b/sec BPSK over AWGN), and then we strategically incorporate concatenated error correction codes to make the link even more power-efficient (1200 b/sec BPSK with concatenated FEC over AWGN). Our overall result will be a performance evaluation of these three scenarios. | |
| **URL** | <https://sites.google.com/a/temple.edu/programmable-communication-group/> | |

**Executive Summary**

Table of Contents

[1. Problem 5](#_Toc373771118)

[1.1. Overall Objectives 5](#_Toc373771119)

[1.2. Historical and Economic Perspective 5](#_Toc373771120)

[1.3. Candidate Solutions 6](#_Toc373771121)

[1.3.1. Forward Error Correction: Block and Convolutional Codes 6](#_Toc373771122)

[1.3.2. Line Coding: Non Return Zero and Manchester 7](#_Toc373771123)

[1.3.3. BPSK Carrier Recovery 8](#_Toc373771124)

[1.3.4. Coherent and Non-Coherent BFSK Modulation (Cedric) 10](#_Toc373771125)

[1.3.5. Coherent and Non-Coherent BFSK Demodulation (Cedric) 11](#_Toc373771126)

[1.3.6. BPSK and BFSK Timing Recovery 13](#_Toc373771127)

[1.4. Proposed Solution Concept 14](#_Toc373771128)

[1.5. Major Design and Implementation Challenges 15](#_Toc373771129)

[1.6. Implications of Project Success (Brandon) 16](#_Toc373771130)

[2. DESIGN REQUIREMENTS 17](#_Toc373771131)

[2.1. Functional Design Constraints (Brian) 18](#_Toc373771132)

[2.2. Non-Functional Design Constraints (Brandon) 18](#_Toc373771133)

[3. APPROACH 19](#_Toc373771134)

[3.1. Software Simulation Using Matlab/Simulink 19](#_Toc373771135)

[3.1.1. Manchester Encoder/Decoder 19](#_Toc373771136)

[3.1.2. Modulation of FSK and BPSK 22](#_Toc373771137)

[3.1.3. Coherent Demodulation of FSK and BPSK using Phase Lock Loops 24](#_Toc373771138)

[3.1.4. Concatenated FEC codes (Brandon) 36](#_Toc373771139)

[3.2. Hardware Implementation using ISE Design Suite 42](#_Toc373771140)

[4. EVALUATION (Brandon) 42](#_Toc373771141)

[5. SUMMARY AND FUTURE WORK 43](#_Toc373771142)

[6. ACKNOWLEDGEMENTS 43](#_Toc373771143)

[7. REFERENCES 44](#_Toc373771144)

# 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 (line codes), passband modulation, 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 two common types of forward error correction - block codes and convolutional codes, two line codes – Non-Return to Zero (NRZ) and Manchester code, and finally coherent and non-coherent demodulation techniques used for BPSK and FSK. This includes solutions for carrier recovery and timing recovery.

### 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 our 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 in NRZ and Manchester is where is defined as the bit period. It can also be shown that each line code has a theoretical probability of bit error to be:

Since they both 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 the line code that is used in both the FSK and BPSK modem is the Manchester coding.

### BPSK Carrier Recovery

The 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. It’s mathematically easy to analyze and its hardware implementation is not as complex as the Costas loop. As the name implies, the received signal is squared to remove any phase offsets and then processed by a band-pass filter to remove as much noise as possible. After 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 mixer where it is mixed with the received waveform and the timing can be recovered (Nguyen & Shwedyk, 2009). The operation of the squaring 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). One of its disadvantages is its mathematical complexity compared to the squaring loop, but in terms of hardware components needed for complete coherent demodulation, they both require approximately the same amount.



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

Coherent modulation utilizing the Costas loop would require one band-pass filter, three low-pass filters, three multipliers and a VCO. Likewise, the squaring loop would also require one band-pass filter, three multipliers (including the squarer) and a VCO. Instead of three low-pass filters needed by the Costas, the squaring loop only requires two. Note also that the squaring loop requires a flip-flop for frequency division, but with today’s FPGA’s, a single flip-flop is negligible. An alternative to this is a phase lock loop configured as a fractional-N-frequency synthesizer (Crawford, 2008).

Following analytical and experimental analysis of both the Costas loop and squaring loop in Simulink, it was decided that we implement the squaring loop for BPSK carrier recovery. The mathematical complexity of the Costas loop proved difficult to derive an accurate transfer function that described the loop’s behavior. Conversely, the squaring loop uses a phase lock loop in which literature was rich and plentiful. We were able to derive a transfer function that accurately modeled the phase lock loop’s behavior. This is important because it allows us to confidently compensate for phase and frequency offsets in addition to tuning transient performance that minimize bit error.

### Coherent and Non-Coherent BFSK Modulation (Cedric)

The BFSK modem abides by the Bell 202 standard which uses frequencies of 1200 Hz for Mark (*b0*) and 2200 Hz for Space (*b1*). Following this protocol, the phase of the signal can be implemented either coherently or non-coherently. A coherent modulation (continuous phase modulation) implies that the phases of the two tones representing the data are always the same, which inherently prevents discontinuous jumps between a Mark and Space. Conversely, non-coherent FSK modulates the two signal waveforms without any effort to match the two signals’ phases, hence the modulated signal may experience discontinuous jumps in its phase component.

Coherent FSK modulators tend to consist of several complicated components, such as memory to ‘remember’ the previous phase of the signal (Rao et. al, 1990). Therefore they are not commonly used to avoid unnecessary loss of power although they yield a better BER performance (Rao et. al, 1990). On the other hand non-coherent FSK modulation is simpler to implement and is commonly used in several transmission system despite its lower BER performance compared to the coherent modulation.

**Non-coherent modulation**

As previously mentioned, coherent modulation requires continuous phase of the modulated signal, which can involve complicated hardware or algorithms. As a result, it is common to ignore the phase of the signals and directly modulate the two signals. This implies that the phase modulated signal will be subject of random variations. The non-coherent modulator can be implemented using the two sinusoidal wave generators (two sine functions) and a multiplexer controlled by the input data. Switching between the frequencies will generate a BFSK waveform with a bit period equal to the periodicity of the switches.



Figure 5. BFSK modulator used in non-coherent modulators. The data controls the output of the multiplexer at data’s baud rate.

**VCO Coherent modulation**

Non-coherent waveforms originate from the fact that two totally different sources are used to modulate the data, therefore the phase resulting from the modulator varies as the signal is altered through the two tones. Using a single source to modulate will maintain a continuous phase as expected. Voltage Controlled Oscillators are commonly used to provide a continuous phase, and generate a sinusoidal wave based on the input control signal. The sinusoidal output from the VCO is described in Equation 2, where the quiescent frequency is the free running frequency (quiescent frequency) output of the VCO,



Figure 6. Coherent modulator for BFSK. The data controls the output of the VCO through Equation. 2

### Coherent and Non-Coherent BFSK Demodulation (Cedric)

The Bell 202 Protocol is quite complex due to the frequency deviation and the ratio between the data rate and carrier frequency. The Bell 202 modem uses frequencies with a small frequency deviation from the carrier frequency to represent it binary data ‘’ and ‘’ where the tones selected are 1200 Hz and 2200 Hz. The frequencies selected to represented two symbols result in a signal space that is difficult to optimize since the frequencies are not orthogonal as the minimum frequency separation is denoted in equations (3) and (4) below (Nguyen, et. al, 2009).

**Coherent demodulation**

Similar to the modulator, the demodulator can be categorized into a coherent modulator and a non-coherent demodulator. In the coherent demodulator, the phase of the modulated signal is either known or is extracted prior to demodulation. Several methods are used to extract the phase of the modulated waveform, such as the phase-lock loop or more complicated systems as illustrated in Figure 7.



Figure 7. Coherent demodulator for BFSK, where the signal is assumed to have a constant phase

The coherent demodulator in Figure 7 uses two parallel branches for matching the Space and Mark onto the two orthonormal basis functionsand. Finally, using the appropriate threshold and decisions, the bits can be recovered using Maximum Likelihood. The correlation receivers or matched filters and are designed to be orthonormal to each other, and at a frequency corresponding to the Mark and Space signals. The process of using matching filter results in an optimum demodulator in terms of BER and can even be reduced to a single correlation receiver according to the relationship shown in Equation (5).

This is illustrated by the block diagram in Figure 7.

**Coherent demodulation (PLL)**

****

Figure 8. Coherent demodulator for BFSK using PLL.

The use of a phase lock loop is also a valid method for demodulating FSK. The PLL has been integrated in several radio for demodulating FM and can also serve to demodulate FSK signals. In the case of non-coherent signals, the PLL acts as an estimator of the frequencies and phases (Rao et. al, 1990) by rapidly matching the output of the VCO, the PLL is used to appropriately estimate the correlation between the signal and the output of the VCO.

****

Figure 9 .Non-Coherent demodulation

In the case of a signal with discontinuous phase, non-coherent demodulation is regarded as the ideal demodulator in FSK modulation. The advantage of non-coherent demodulator come from their ability to ignore the phase change contained in the signal. Matched filters are still utilized however, an envelope detector is present in each branch after each tone’s matched filters.

The matched filters are configured with the same objective as the coherent receiver, and the use of the envelope detector removes the phase changes. The performance of the non-coherent demodulator results in performances that closely approach the performance of the optimum coherent receiver. (Linsey et. al, 1977)(Rao et. al, 1990)



Figure 10. Non-coherent demodulator for BFSK, where the Mark and Space filters are centered at 2200Hz and 1200Hz respectively.

### BPSK and BFSK Timing Recovery

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 timing recovery circuit is 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.

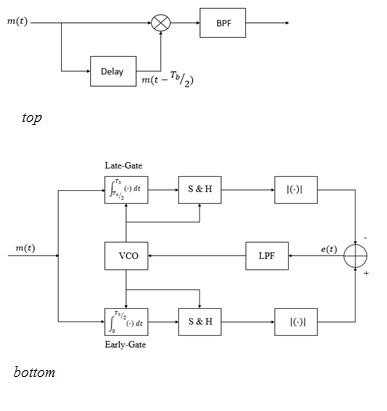


Figure 11. Architecture for the open- loop timing recovery (top) and the closed- loop (bottom)

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). This problem is alleviated by using a closed loop feedback architecture shown in Figure 10 (bottom). The principle idea behind the Early-Late synchronizer is to sample the received signal at different times and compare the two 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 detailed analysis of its operation is discussed in the Approach.

## Proposed Solution Concept

This senior design project will primarily consist of developing a software simulation (Simulink) and a hardware implementation (FPGA) of a 1200 b/sec BPSK modem (using soft-decision) with a (2, 1, 7) convolutional code. We will gather measured BER performance data and compare it to the theoretical BER performance. When the measured and theoretical performances match closely, the SW simulations and HW implementations will be considered complete. The FPGA hardware implementation is expected to be the most important product of this senior design project.

Additionally, we will be performing a budget link analysis of three amateur radio satellite telemetry systems. The purpose of this is to compare the accessibility (in terms of link margin) of amateur radio satellite telemetry links when AFSK (System A), BPSK (System B), or BPSK with convolutional coding (System C) is received by a “less capable” ground station (like a FunCube satellite dongle). The first system (System A) will replicate the many amateur radio satellite telemetry links that exist today – 1200 b/sec Bell 202 modulation without FEC. This FSK modem will perform non-coherent modulation and coherent demodulation.

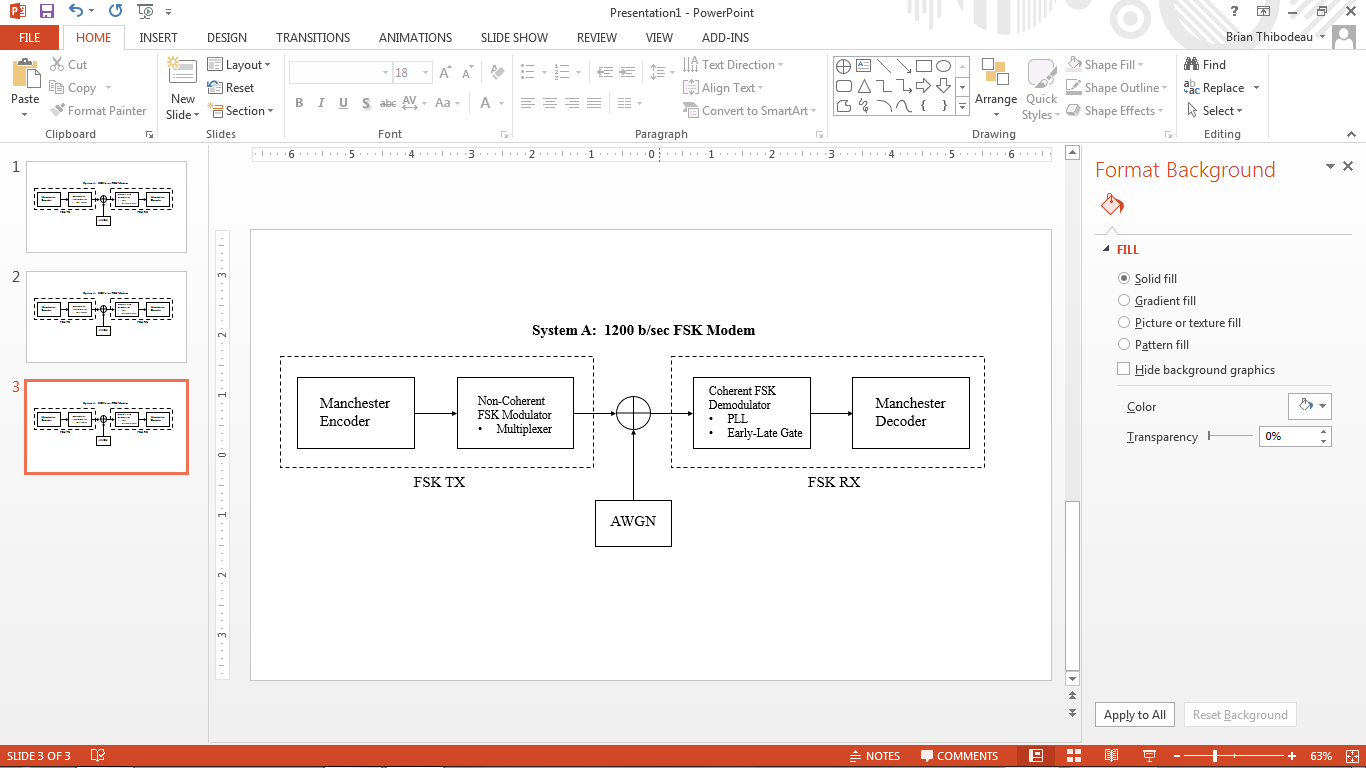


Figure 12. Common FSK link in Amateur Radio Satellite station

The second system (System B) will exploit the fact that BPSK modulation is better than AFSK in amateur VHF and UHF operations (Hsiao, et. al, 2000) – 1200 b/sec BPSK *without* FEC. This system will also perform Manchester encoding and decoding in addition to suppressed carrier reconstruction using a squaring for suppressed carrier reconstruction. Like the FSK demodulator, timing recovery is also accomplished through the Early-Late gate synchronizer.

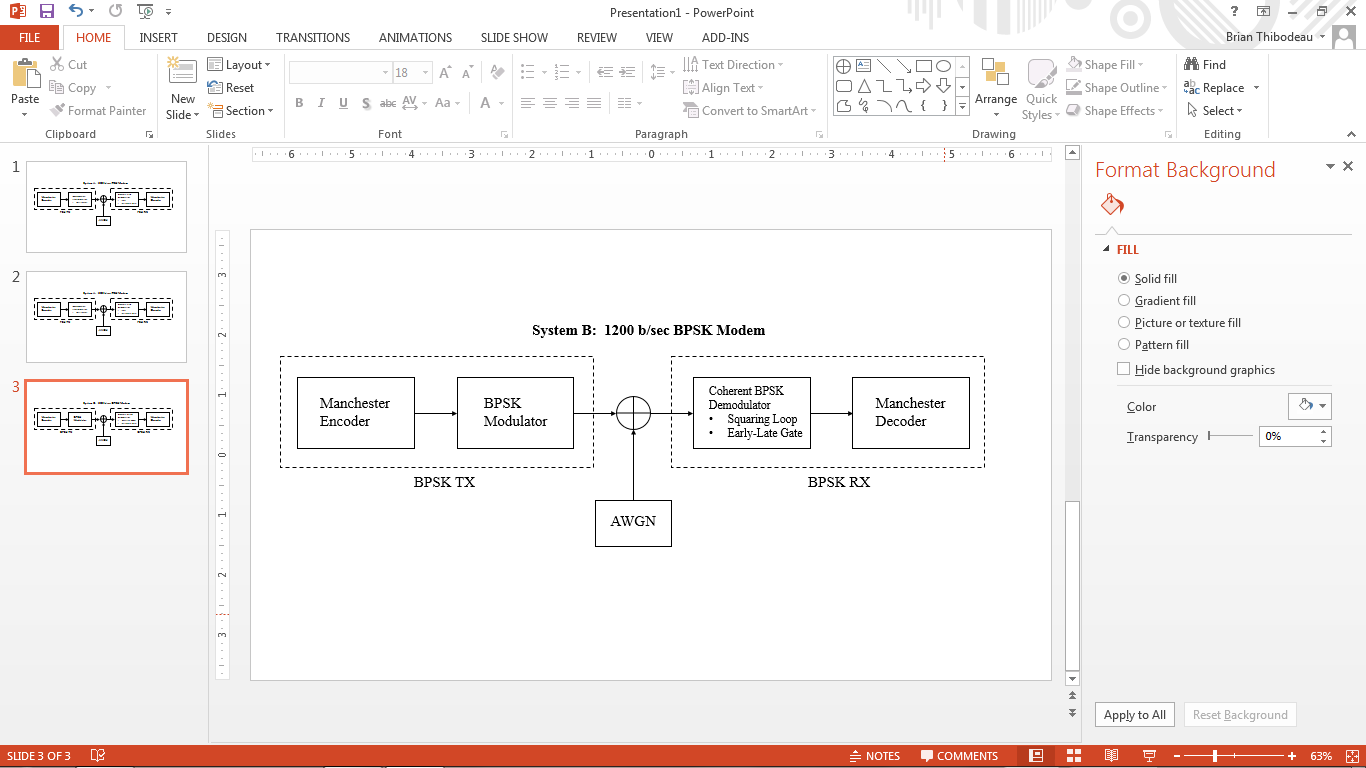


Figure 13. Common BPSK link, implemented in System B

The third system (System C) will be a robust version of system B – 1200 b/sec BPSK *with* FEC. FEC correction will be accomplished by use of the (2, 1, 7) convolutional code. This system will be implemented in FPGA hardware.

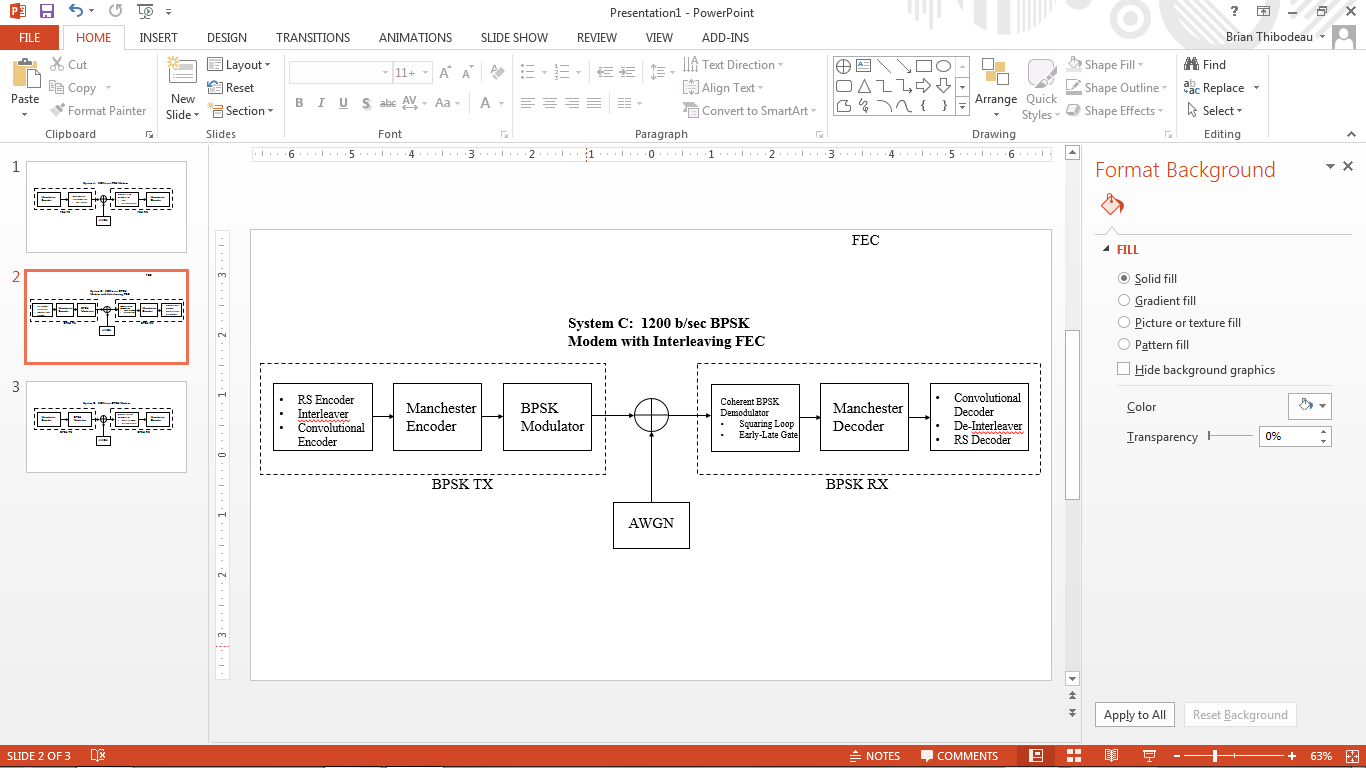


Figure 14. Proposed BPSK w/ FEC link, implemented in System C

## Major Design and Implementation Challenges

The three systems discussed in Section 1.4 (Proposed Solution Concept) will each have their specific requirement to complete the intended analysis of the telemetry transmission. The general implementation of each system on FPGA is intended to be simulated under common signal distortion. The Additive White Gaussian Noise channel will be used for all three systems on purpose of evaluating their respective performances when used in satellite communication. Hence, an Intellectual Property from Xilinx will be ideal to fully complete the analysis of each systems.

In terms of System A (FSK modem,) a coherent demodulator which follows that of the common Bell 202 modems is implemented to analyze the past designs to the more innovative systems (System B and C). The demodulator is expected to resemble the present methods used in Amateur Radio Satellite operators, using discrete components implemented in FPGA. The coherent PLL demodulator is then designed using controls theory to demodulate signals with similar SNR as previous modems under the standard Bell 202 protocol.

The BPSK components of System B and C is expected to demodulate the incoming BPSK signal using the Squaring Loop, designed using controls theory. The squaring loop is expected to regenerate the carrier frequency under severe SNR conditions, hence the senior design project faces the challenges of designing a robust demodulator using the squaring loop for BPSK demodulation. The PLL is also expected to endure frequency changes that may be as large as double the carrier frequency used, therefore the ramp response of the PLL’s transfer function will also need to be minimized. Similarly the acquisition time (time needed for the PLL to attain the target frequency) will need to be minimized to reduce the loss data.

The error correction section of System C is intended to feature correction for burst of errors and random errors which may occur during transmission of data. 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.

The Systems A, B, and C all need to include data recovery ability to fully recover the baseband signals. In all systems, data recovery needs to correlate, sample depending on the recovered clock of the transmitter. Such synchronization process is done using the Early-Late synchronization technic on all three receivers of the Systems.

## 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 BPSK modem with the (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 approach for rapid prototyping and simulation of the various subsystems required by each of the modems. Thus the FSK and BPSK modems were first designed in Simulink which enables us to gain a high level operational understanding of each modem. In addition, Simulink provides the tools and blocks necessary to evaluate the performance of our system under a variety of conditions. Of particular importance was the evaluation of system performance due the effects of 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 that is used in both the FSK and BPSK modems. In section 3.1.2, the FSK and BPSK modems design is discussed. Then in 3.1.3, coherent demodulation of both FSK and BPSK is discussed by examining the phase lock loop. 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.

### Manchester Encoder/Decoder

The binary digits from the computer (TNC) are abstract values and need to be converted to tangible waveforms. In wireless communication, Manchester code has established itself as a standard signaling technic among the several others. Signal techniques are chosen depending on several criteria among those criteria, synchronization is an indispensable component of the receiver. Being that Manchester code contains such criteria improves the synchronization process, hence it is referred as a self-clocking signaling technic.

Manchester code is also not a complicated signal scheme to implement and needs few components to obtain a self-clocking behavior. This ease to implement has attracted the Amateur Radio community and also has made itself the standard protocol for Ethernet. The IEEE standard protocol maps the binary values and into negative and positive edges of a square waveform only during the falling edge of the clock. Therefore, transitions at the positive edges of the clock contain no information, Figure 15 illustrates the protocol from the IEEE 802.3 protocol where and.

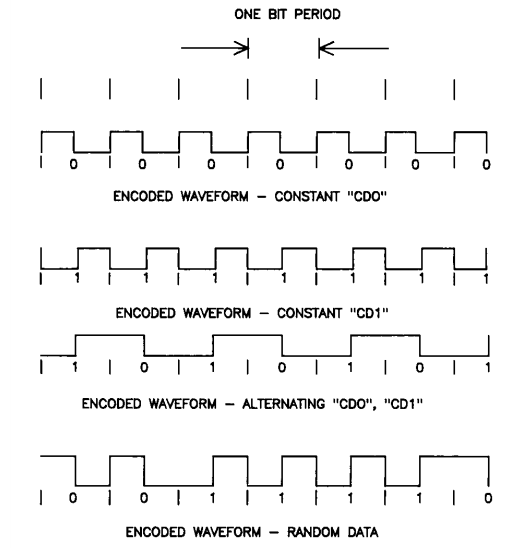


Figure 15. Manchester code based on the IEEE 80.3 standard for Ethernet (IEEE, 2012)

Implementing the Manchester code can be done either using switches or the XOR logical operator () as depicted in the Figure 15(left). Therefore Manchester code is implemented in Matlab Simulink by XORing a stream of random data with the transmitter's clock which for the BPSK modem and BFSK modem has a bit rate of 1200 b/sec. Further modification can also be done by altering the magnitude values of the signal waveform, either

Where the latter is referred as the Manchester code- Leveled.

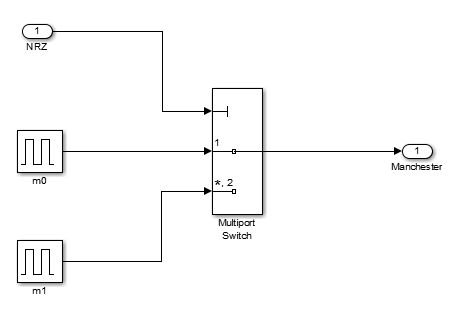
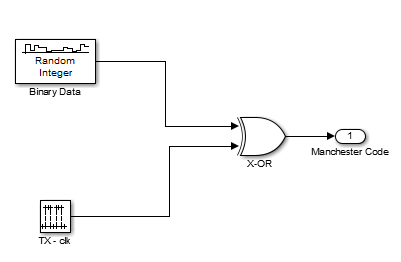


Figure 16. Manchester encoder using an XOR gate (left) Multiplexer (right)

At the receiver, the data must also be decoded to extract the information from the coded signal. The decoder is done using the Manchester code basis function denoted as. Decoding the Manchester code yields very accurate results due its efficient geometrical properties. The correlation coefficient between the two signaling waveforms equals to, hence the Manchester encoding provides a maximum distance between the waveforms as illustrated by (Nguyen et al). In the modems, the decoding is also done following the same approach, where the product between the encoded waveform and the extracted clock is used to recover the binary digits.

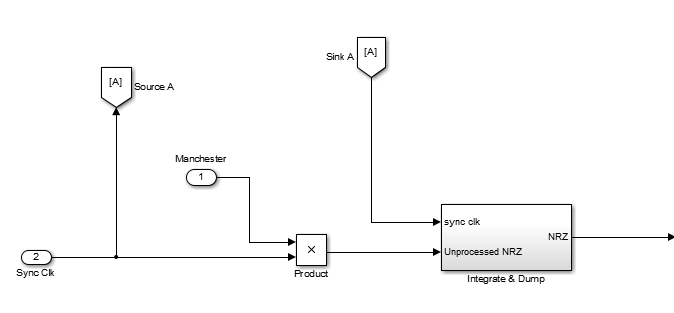


Figure 17. The decoder for the baseband Manchester is illustrated above, where the product block and the integrate and dump make up the corellation receiver for the Manchester code.

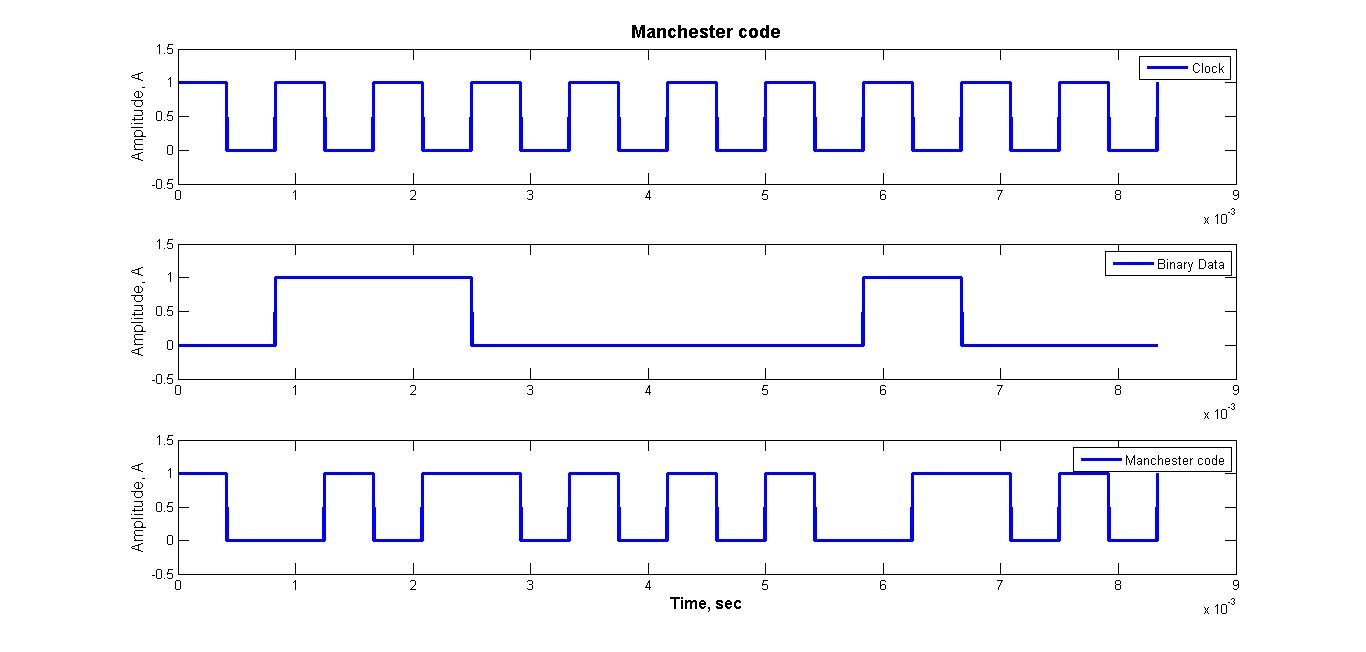


Figure 18. Results of the Manchester code

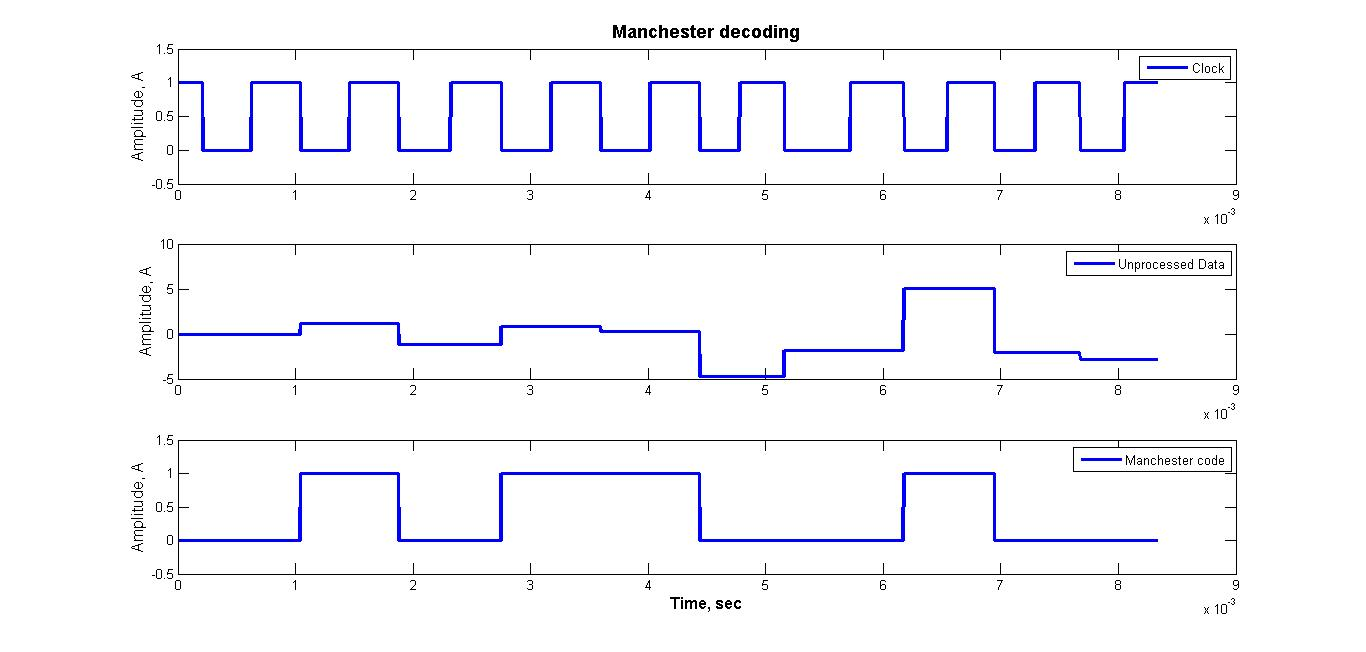


Figure 19. Results of the Manchester decoding

### Modulation of FSK and BPSK

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 as shown in 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 can be designed to gate two antipodal sinusoidal carriers with and. This implementation is illustrated by the Simulink model in Figure 20.

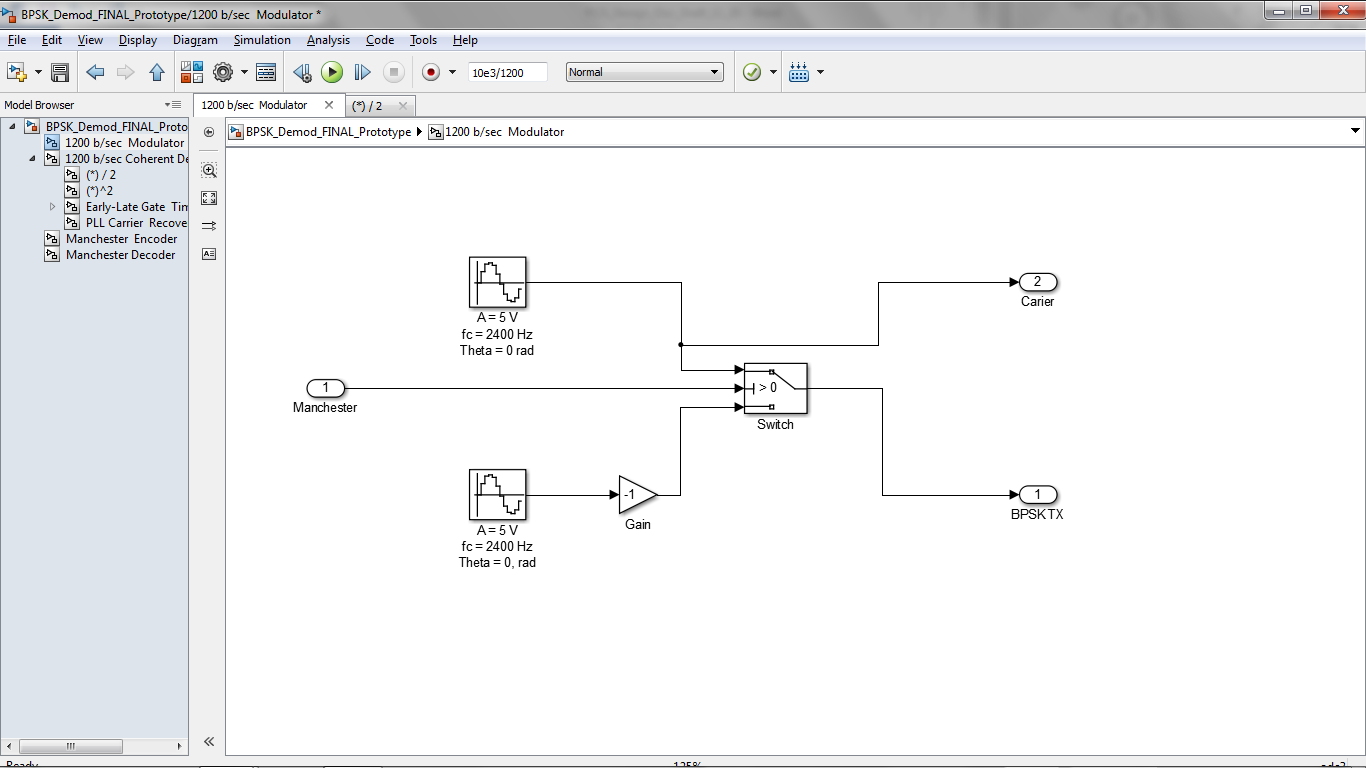


Figure 20. BPSK modulator that uses the Manchester data stream to gate two antipodal sinusoidal carriers that result in the BPSK modulated signal.

Figure 21 below presents one particular simulation that illustrates the operation of the BPSK modulator. When the input Manchester code is ‘high’, the positive sine wave is transmitted and when the input is ‘low’, the negative sine wave is transmitted.

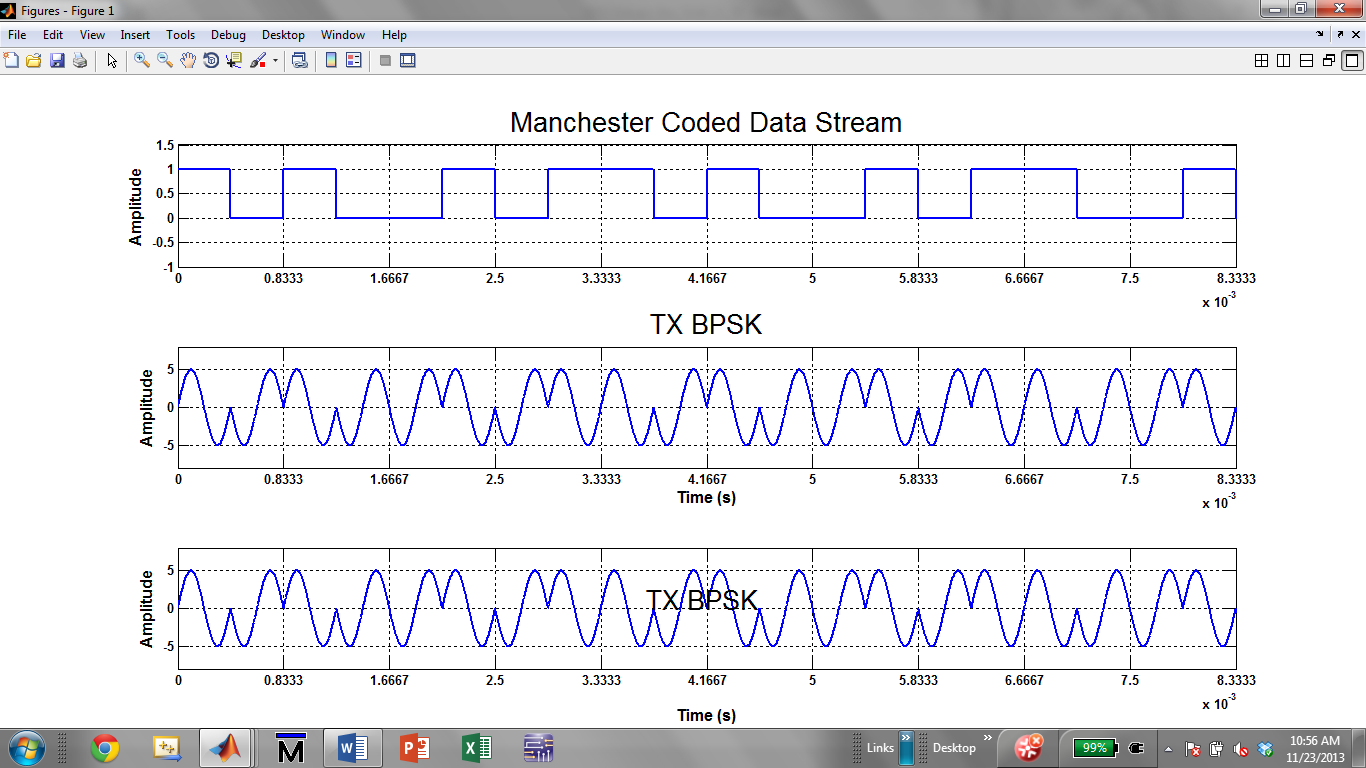


Figure 21. The transmitted BPSK signal has 180 degree phase shifts that correspond to logic level transition of the Manchester coded data stream.

**FSK Modulation**

FSK Modulation may be implemented either coherently or non-coherently where the linearity between the tones dictates the type of modulation. The non-coherent modulator which is the simpler of the two is obtained by simply gating the Mark and Space with the Manchester coded waveform. Therefore the multiplexer of Figure 5 can be used to obtain the behavior of the non-coherent modulator. In Simulink, the modulator is obtained using a Switch to obtain through the tone corresponding to the data transmitted. The signal is described to have a random phase () modeled as a uniform random variable from at the start of the bit time. Although non-coherent modulation may yield the lower error probability, engineers have favored the non- coherent modulator because of its simplicity and lack of complicated analog components (Wilk, 1995.) Therefore, the FSK modulation has been represented using the following equation to illustrate the relation between the phase and the frequency shift.

By directly gating two different frequencies an FSK signal is generated according to the Bell 202 protocol which states:

The Bell 202 protocol can be illustrated using Figure 22 top, which is obtained in Simulink using the switch illustrated in Figure 22, bottom

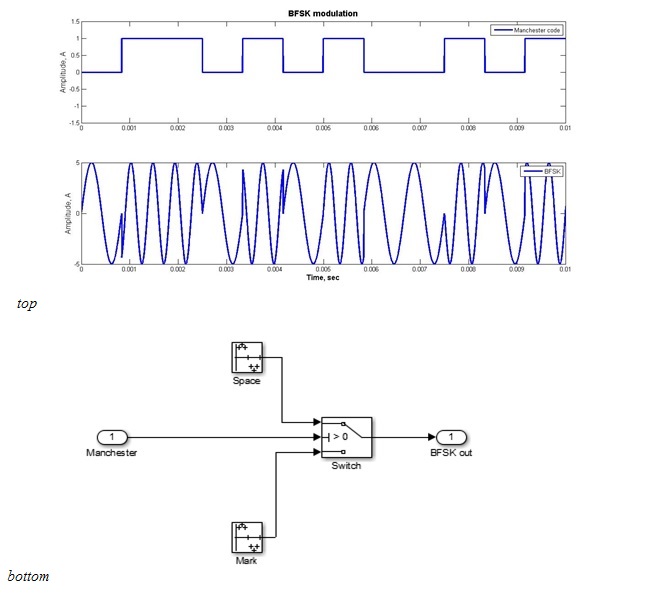


Figure 22. BFSK non-coherently modulated using switches

### Coherent Demodulation of FSK and BPSK using Phase Lock Loops

**PLL Design for Coherent FSK Demodulation**

In every communication system demodulation is an essential element to complete the system. Recovering the data is done either coherently or non-coherently depending on the technic used to modulate the two frequencies (1200 Hz and 2200 Hz). Because the receiver has no information on the transmitted data (in terms of the phase) the receiver must consider the transmitted phase of the transmitted signal. The non-coherent demodulator of Figure 23 ignores the phase of the signal using two branches to demodulate. The non-coherent demodulator is implemented only with filters, therefore can be easy to implement in terms of setting and adjusting the filters (Mark - 2200 Hz and Space 1200 Hz.) However, inter-symbol interference is certain to occur between frequencies, which will evidently complicate the demodulation process at the receiver. Hence the demodulator of the BFSK modem will consist of a coherent demodulator using a PLL to keep track of the random phases the signal may undergo. As discussed by (Lindsey et al) the PLL estimates the frequency of the signal and outputs the correlation between the tone and the running frequency of the VCO.

To include the PLL onto the modem, the non-linearity of the PLL was modelled around its three components the Phase detector, the Loop Filter, and the VCO. The PLL is then made linear with the assumption that the phase difference between the transmitted signal and the output signal from the VCO is small. Then the PD which is implemented using a multiplier approximated to be only the difference between the signals with a gain and then passed through a loop filter with wide enough to pass the modulated frequencies. The output of the loop filter is taken as the demodulated FSK and fed back to drive the VCO.

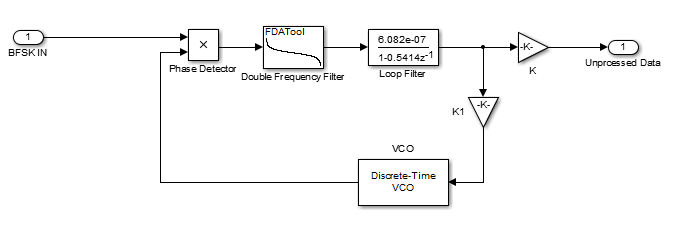


Figure 23. Model of the PLL for BFSK demodulation in Simulink

In the demodulator, the output of the loop filter is the demodulated FSK signals, analyzing the PLL, using basic modeling technics the system can be simplified to a single transfer function. The transfer function of the linearized PPL model is used to optimize the demodulation using basic controls concepts. The loop filter can be implemented in several different methods either lead-lag, active filters or a simple low-pass filter. In our BFSK modem, the loop filter was designed using the low-pass filter because of the PLL FSK demodulator's frequency response. Therefore the PLL is modeled by the following equation:

Where is the transfer function of the desired Loop filter. Then replacing with the transfer function of a Low pass filter yields the desired transfer function of the FSK PLL.

Second order transfer functions are often represented using the mechanical terms ζ, the damping ratio and ωn the natural frequencies.

Optimizing the system in terms of the settling time we see that the loop filter must have a wide bandwidth

to obtain an appropriate settling time since:

Therefore, the cut-off frequency of the low-pass filter was set at frequency 20 times greater that of the   
bit rate:

The parameters of the equation are found as follow, where the damping ratio is chosen to be = 0.707, and the natural frequency is calculated to be:

A step response can be done to evaluate the parameters of the design. Where a step on the system corresponds to an abrupt change of frequencies, while the bode plot of the system is illustrated in the Figure 24

Machine generated alternative text:
80
75
70
65
co 60
C)
t
D 
4-D
C
a’
50
45
40
35
30
10
Bode Diagram
‘ ! !!!
!
!
!!
! !
¡;
j
ji
j
¡
; ¡
jijj••••••••j•••••
10’
1
Frequency (radis)
1
1

Figure 24. Bode plot of the PLL for FSK demodulation

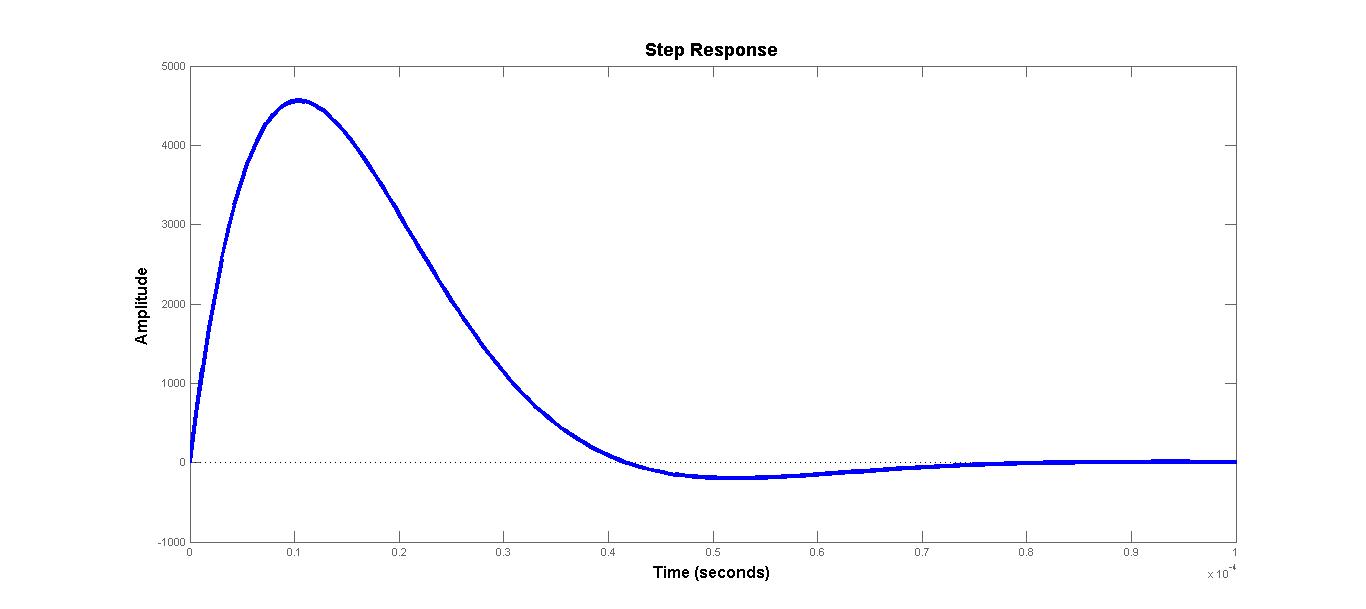


Figure 25. Step response of the PLL for FSK demodulation

The completed PLL can then be included into the BFSK demodulator using a discrete VCO, a multiplier for the phase detector and a loop filter. The demodulated data then passed to an envelope detector to further process the data and is finally recovered using the Early-Late Gate method for data recovery.

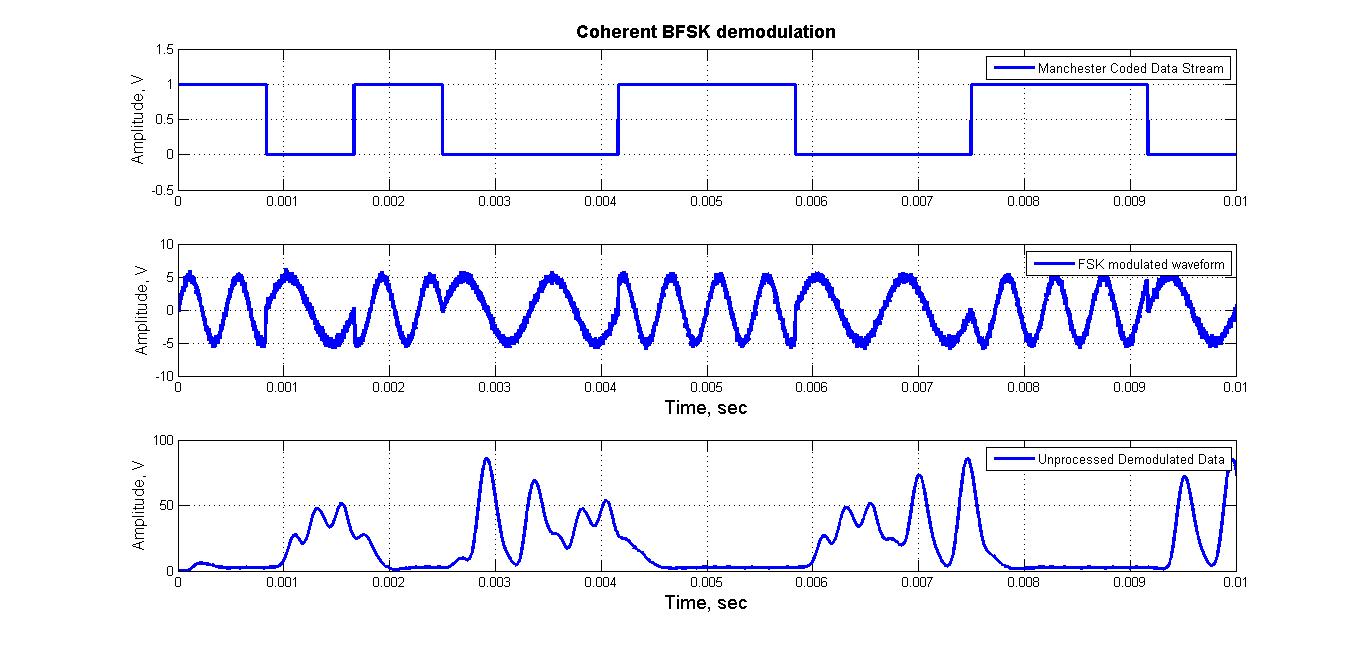


Figure 26. Unprocessed data from the PLL Coherent demodulator

**PLL Design for Coherent BPSK Demodulation**

The Simulink model of the squaring loop demodualtor that was introduced in section 1 is shown in Figure 27. The received signal is first passed through an IIR peaking filter with a single peak at 2400 Hz and a bandwidth of 10/. This corresponds to the fifth null bandwidth of the BPSK signal and the point in which 98% of the total power is captured (Silage, 2009). This filter acts to remove as much noise as possible without loss of information.

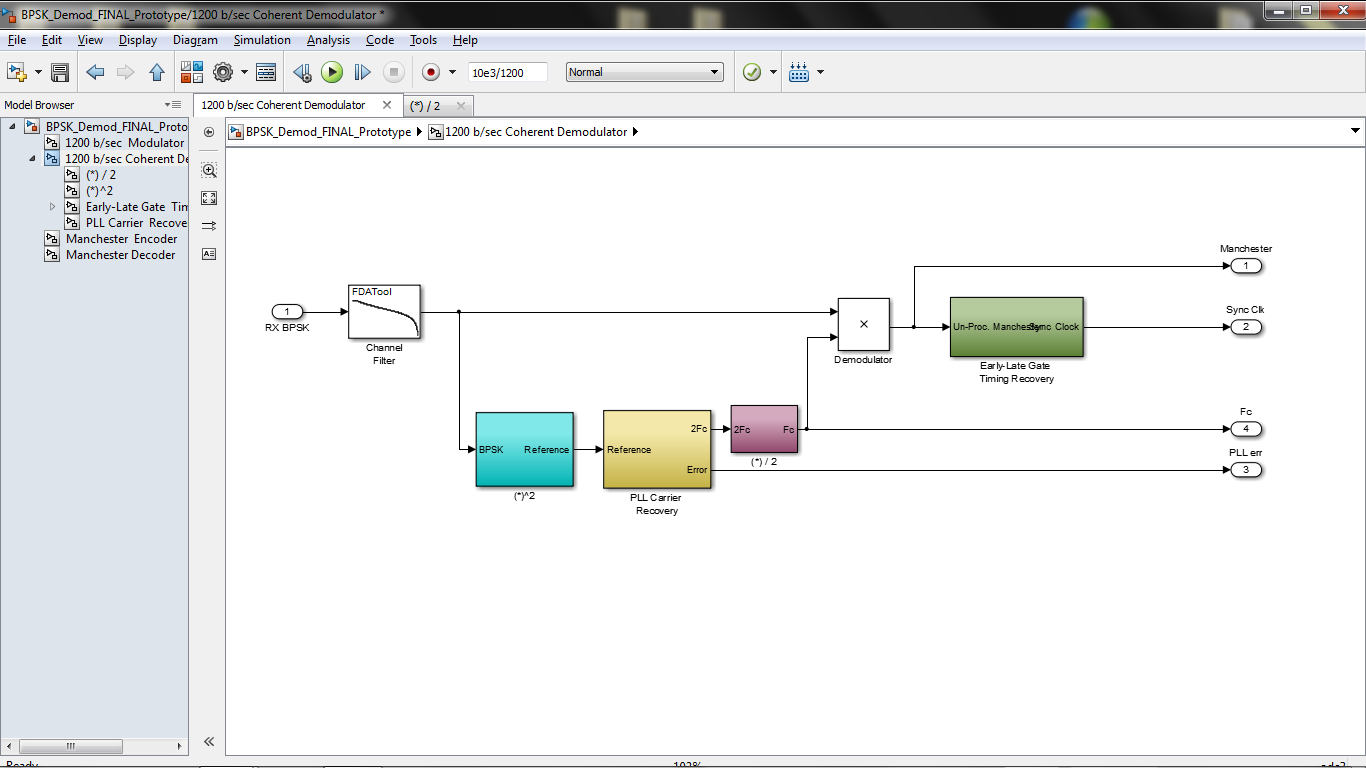
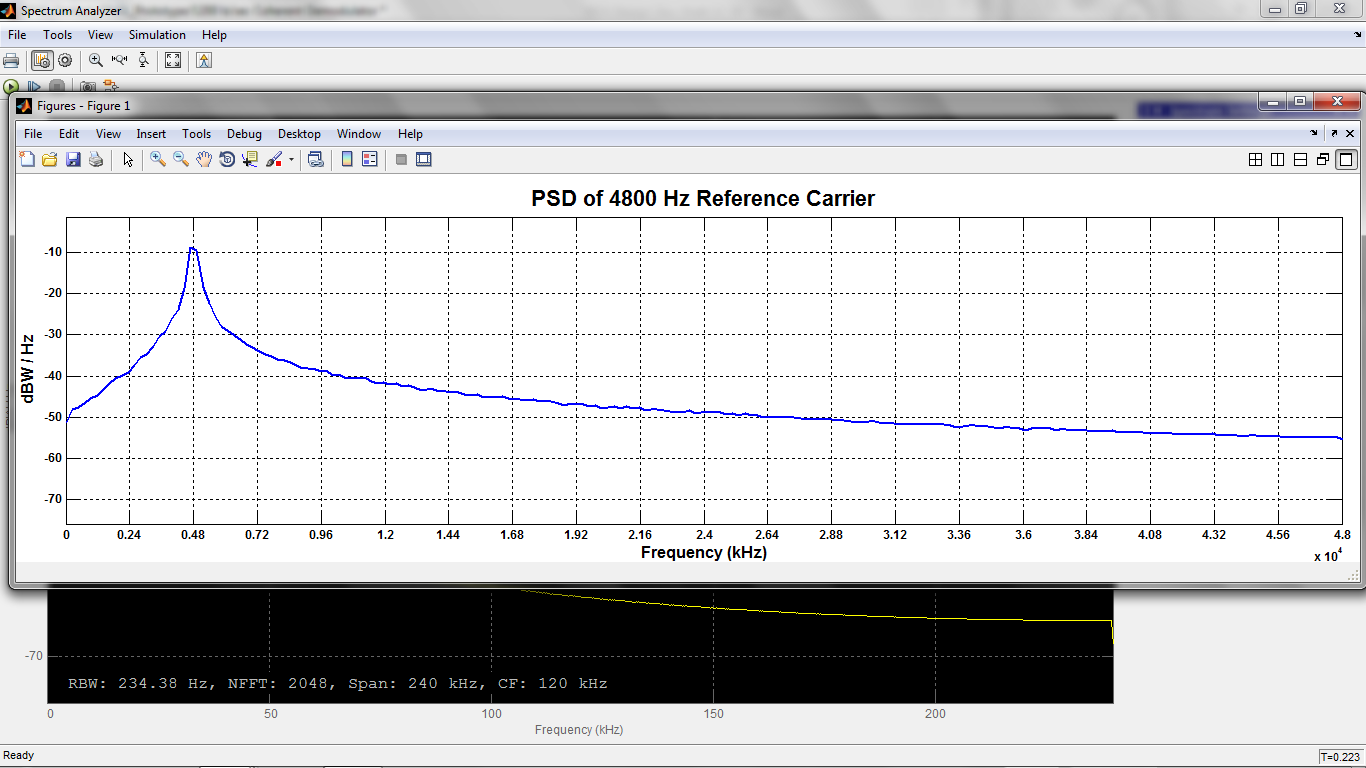


Figure 27. Simulink model of the Squaring loop Demodulator. The carrier of the received signal is reconstructed using a PLL where it is used for coherent demodulation. The early-late circuit regenerates the clock for symbol timing.

Unlike FSK and ASK where the received signal contains a spectral component at the carrier frequency for the PLL to lock onto to, the PSK spectral component of the carrier is suppressed within the PSD of the received signal. The solution to this problem is to square the received signal which produces a spectral component at (Nguyen & Shwedyk, 2009). This is easily shown using trigonometric identities. If the received signal is where is AWGN, then,

The DC component and the noise in equation (3) are filtered using a narrow bandpass filter with a center frequency of and a Q of 100. The resulting PSD after squaring and filtering is displayed in Figure 27.



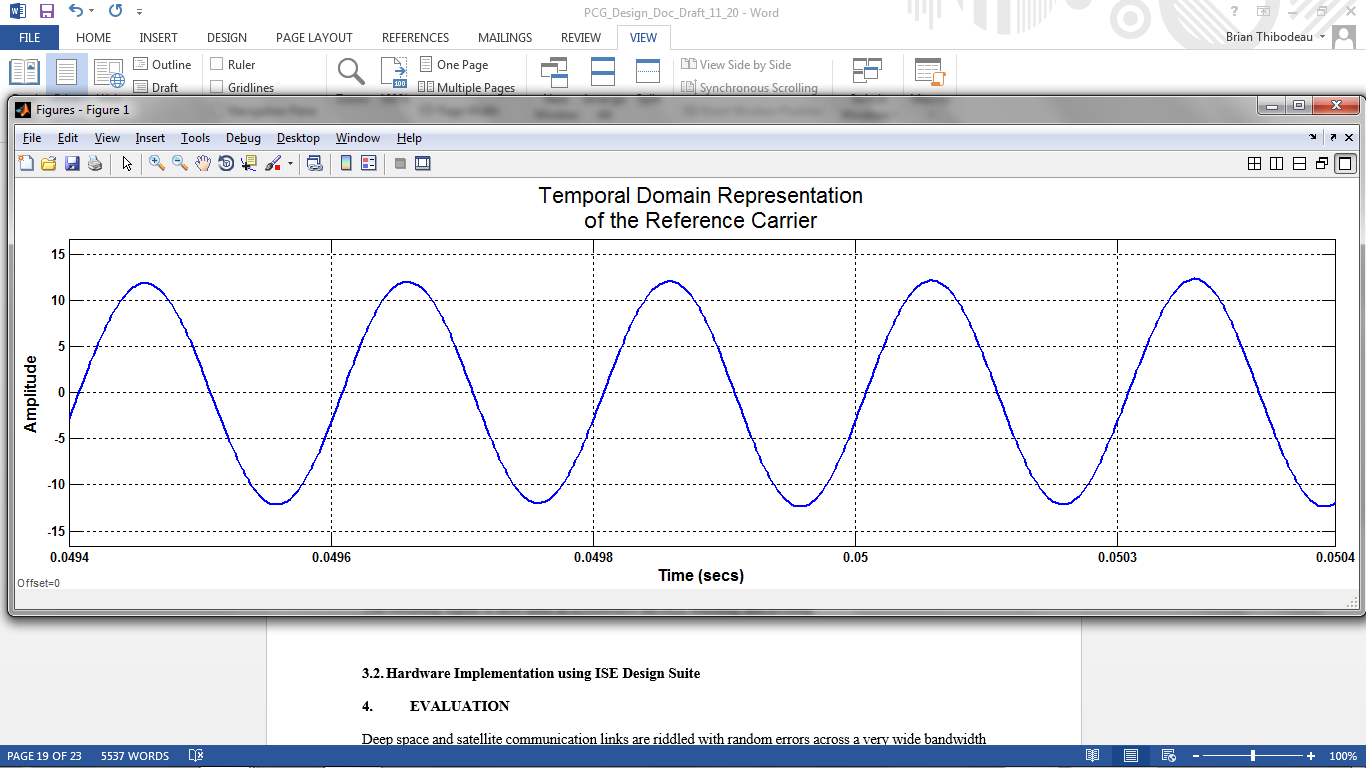


Figure 28. (Top) Power Spectral Density of the 4800 Hz reference carrier and (Bottom) the temporal domain signal of the reference carrier

The reference carrier is shown on the bottom of Figure 27 and is forwarded to the PLL where the phase and frequency of the reference carrier is estimated.

**Phase and Frequency Estimation using a PLL**

All PLL’s are composed of three components, the phase detector, loop filter, and the VCO. Its operation can be understood by considering the PLL in Figure 29. The Phase detector includes the multiplier, phased detector gain and filter. The reference carrier at is multiplied by the locally generated sinusoid which after low pass filtering at produces the phase error. Note that the PLL analysis assumes that the reference frequency is approximately the same as the VCO frequency. Any minor differences are considered in the phases and of the reference and VCO respectively (Nguyen & Shwedyk, 2009).

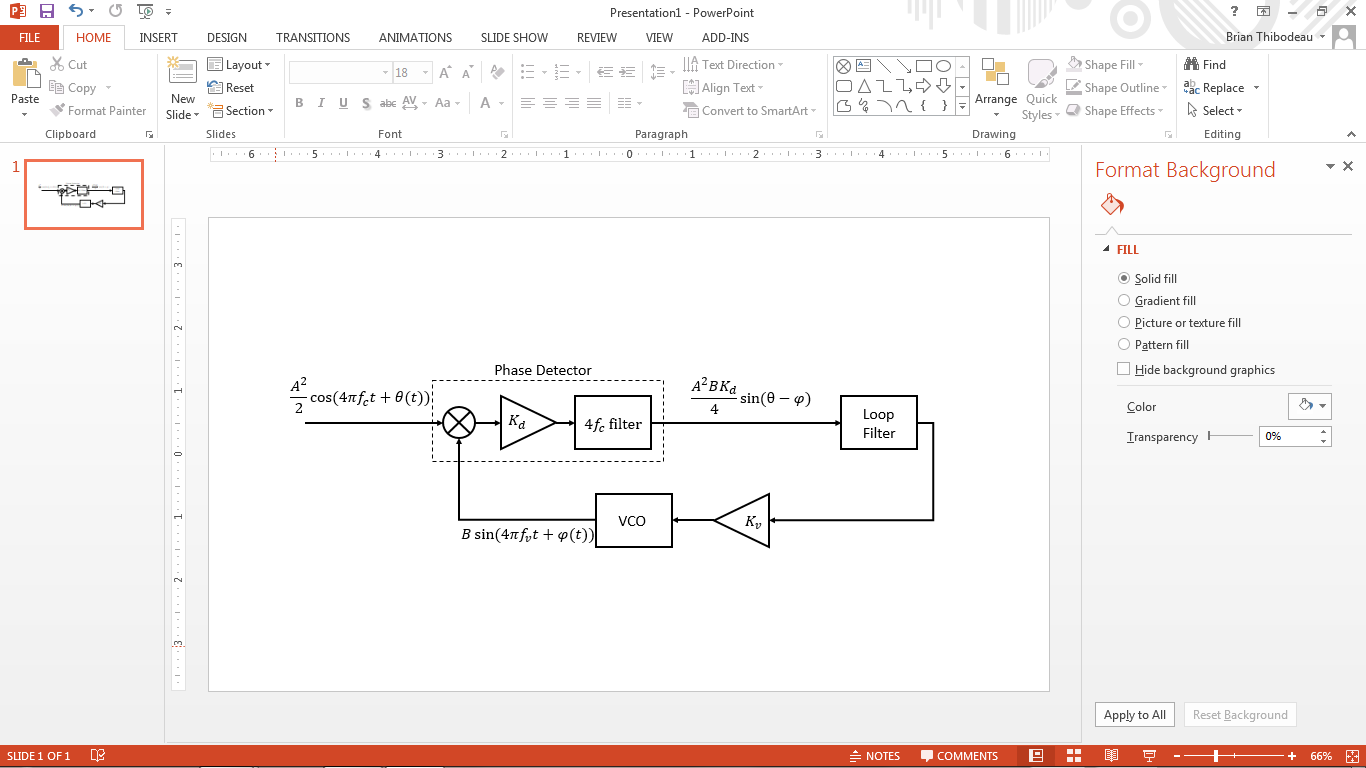


Figure 29. Time domain model of the PLL used for carrier frequency and phase estimation

The multiplication in the phase detector makes the PLL highly non-linear, but if the phase error, , is small than the output of the phase detector can be approximated as, . Under this approximation, the linearized PLL in terms of the phase is shown in Figure 29.



Figure 30. Linearized PLL 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

Once the PLL has been linearized by making the assumption that the phase error is small, conventional controls system theory can be applied. Thus the closed loop transfer function of the PLL in Figure 29 is 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). This results in the following:

The results of the final value theorem imply 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 loop filter expression into the closed loop transfer function given by equation (18), the new closed transfer function describing the PLL’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,

Using the above results, two Simulink models were developed for the design and analysis of the PLL that will be used for phase and frequency estimation in the carrier recovery circuit. The first model is the time domain model that will be implemented in hardware and the second is the linearized model to demonstrate that equations 20, 20.1, and 20.2 can be used to approximate the response of the time domain model.

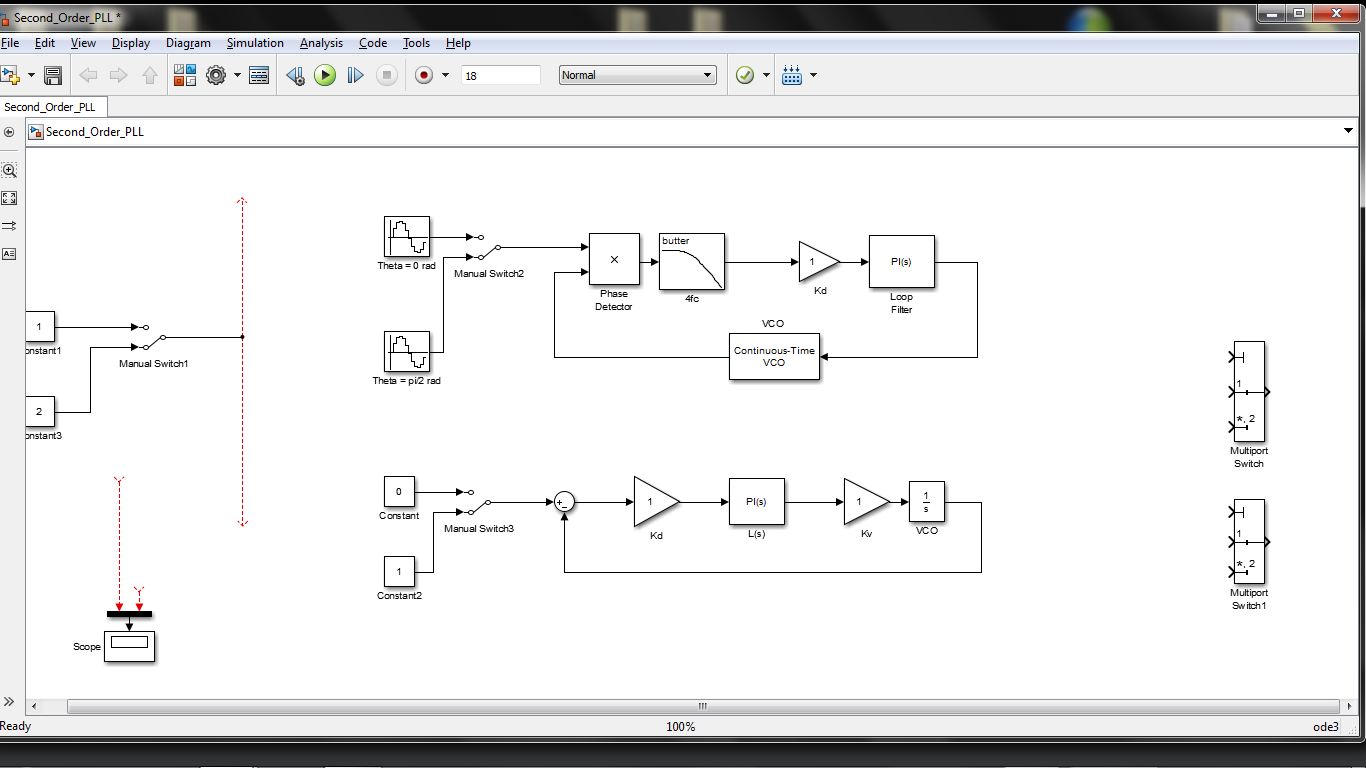


Figure 30. (Top) Time domain model of the PLL used in the carrier recovery circuit and (Bottom) linearized PLL.

The BPSK modem is designed to operate at 1200 b/sec in order to meet the requirements of LEO-AMSATS. Because of the possibility of sudden phase shifts due to AWGN in the channel, the PLL must be designed to respond quickly to these changes. Thus the PLL requires a settling time less than ms in order to minimize the number of bits lost during acquisition. From classical control theory, the settling time of a second order underdamped system can be approximated as . If the dampening ratio, is chosen to be 0.8 so as to minimize the overshoot, it can be shown from equations 9.1 and 9.2 that the natural frequency, should be 6000 rad/s. The resulting phase detector, VCO, proportional and integral gains that satisfy the requirements of the settling time and dampening ration are listed in Table 3.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 1 | 1 | 9600 |  |

Table 3. Phase detector, VCO, proportional and integral gains

In order to verify that the desired settling time is achieved, the gains in Table 4 were appropriatly substitued in the Simulink models illustrated in Figure 30 and simulated. Note that the time domain model simulated a phase step by using two sine wave blocks with one having a 90 degree phase shift with respect to the other. Both had unity amplitude and a frequency of 4800 Hz. The 4800 Hz frequency represents the squared reference carrier that was discussed earlier. The low pass filter in the phase dectector is first order with a cutoff frequency 9600 Hz.

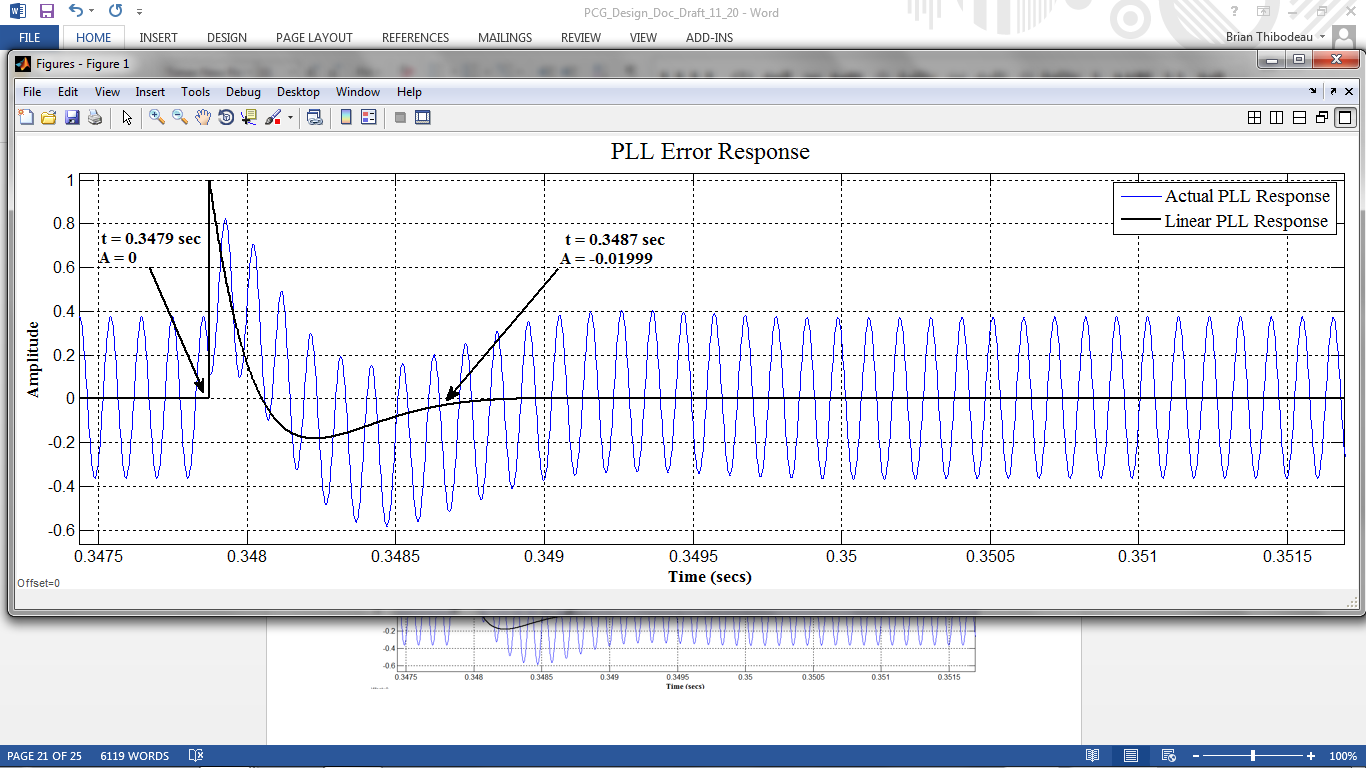


Figure 31. Comparison of the actual PLL error response to the linear PLL response

In Figure 31, the blue sinusoidal response is the actual PLL response to a phase step while the solid black line is the error response of the linear model. The step occurred at and reached 2 percent of its final value at resulting in a settling time of 0.8 ms. It can be seen that the linear PLL response closely approximates that of the actual PLL response. Lastly, the PLL was subject to extreme case of a 4800Hz frequency step. This is twice the frequency of the nominal reference carrier. As illustrated by Figure 31 below, the PLL still manages to lock onto the reference carrier. It does so at the cost of long acquisition and lock time. The time to lock in the extreme case of a 4800 Hz frequency step is approximately 15.8 msec. Although it’s unlikely the PLL will be subject to such large frequency steps, it demonstrates the robustness of the design.

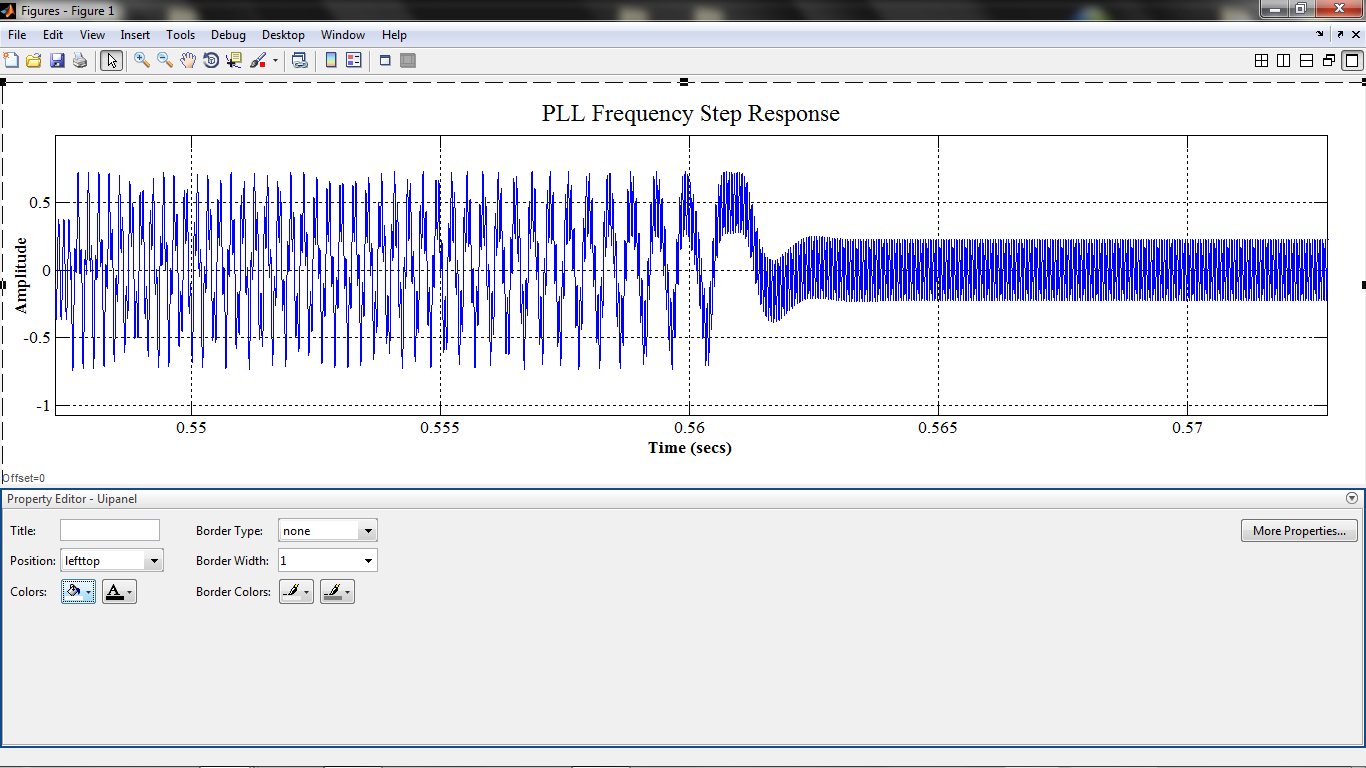


Figure 32. PLL frequency step response to a 4800 Hz step. Time to lock after frequency step is approximately 15.8 msec.

After the PLL locks onto the reference carrier, the output is taken from the VCO which is at twice the frequency of the transmitted carrier. Dividing the VCO output by two is accomplished using a D-Type flip configured as a frequency divider. In Simulink this requires converting the output of the VCO to a square wave so it can be fed to the clock input of the D-flip flop. The output of the D-flip flop is pass through a narrow band-pass filter with center frequency of 2400 Hz and a Q of 10. Figure 33 compares the 4800 Hz square wave from the PLL to the output of the divide by two flip flop.

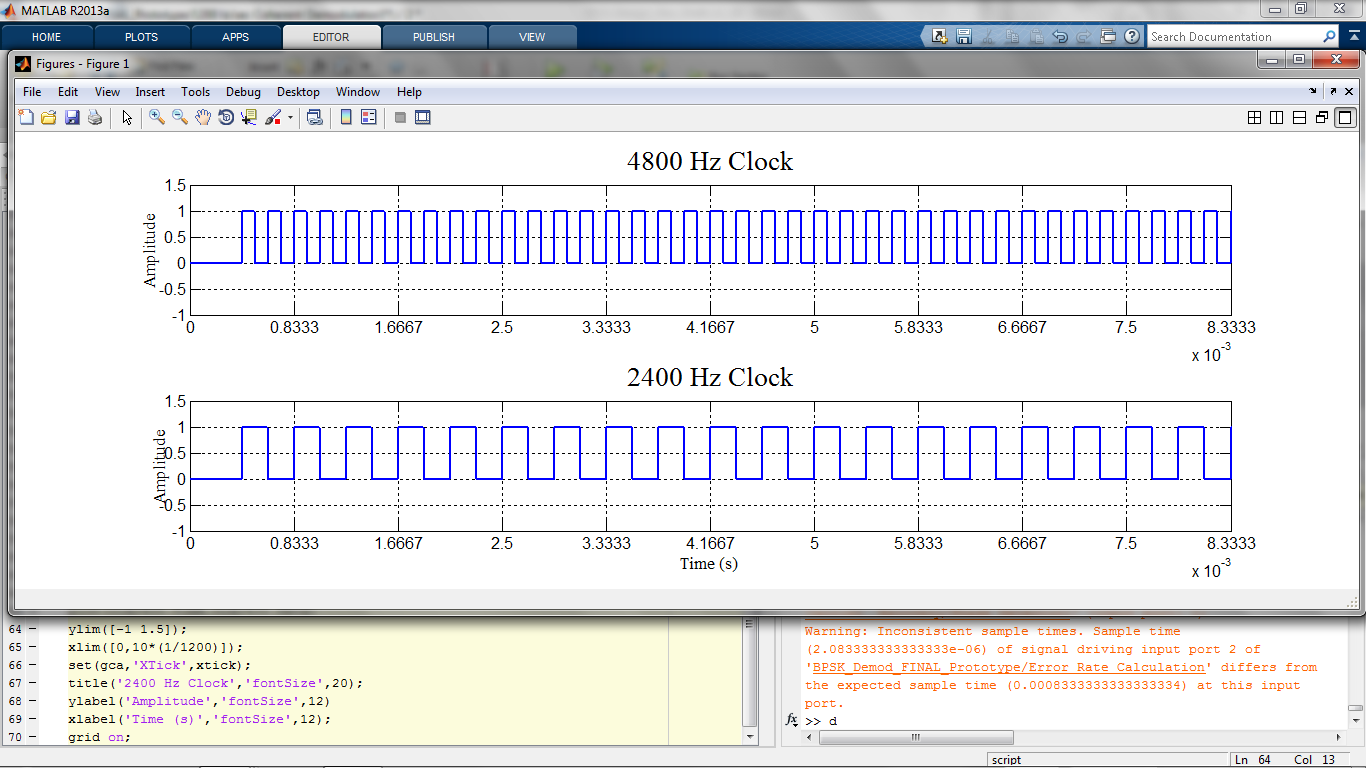


Figure 33. (Top) 4800Hz carrier reference from the PLL after being converted to a square wave. (Bottom). 2400 Hz in phase carrier following the divide by two flip-flop.

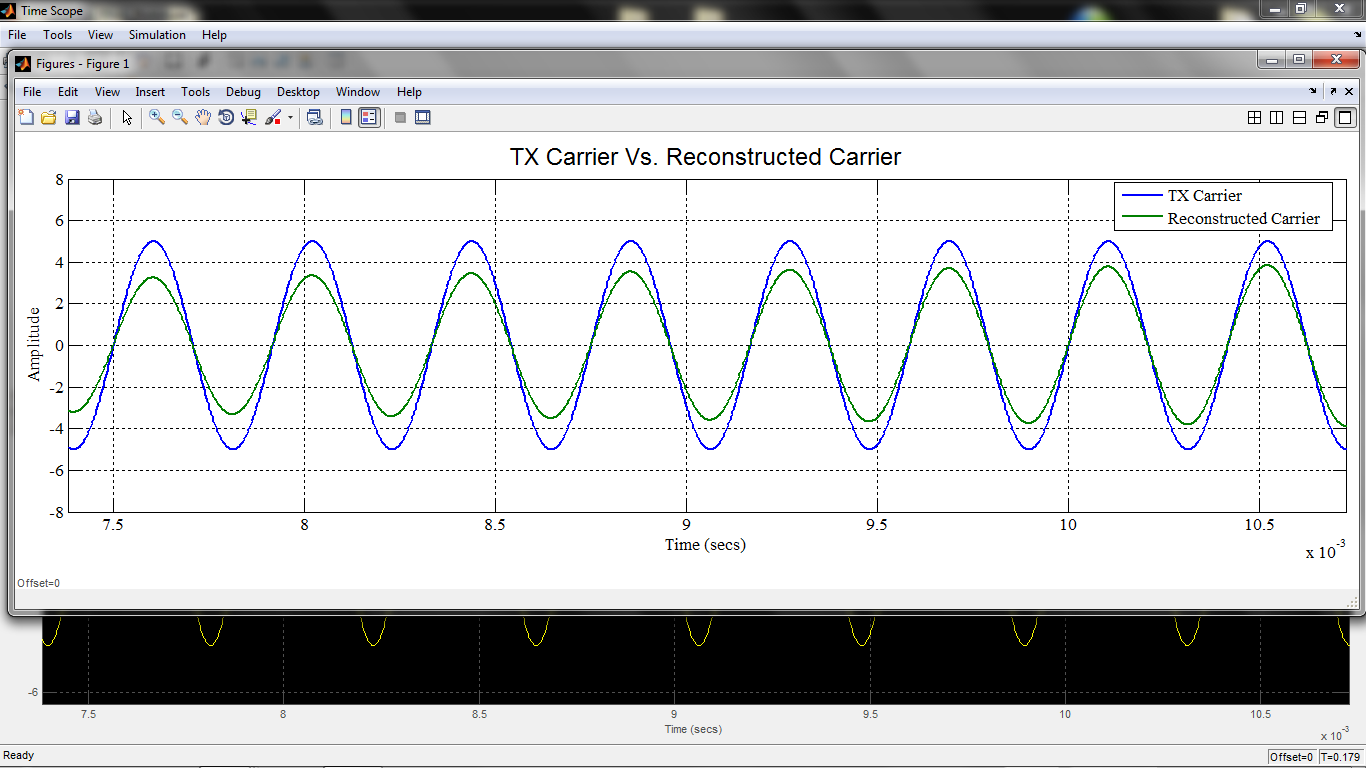


Figure 34. Transmitted carrier superimpose onto the reconstructed carrier demonstrating the PLL’s ability to track the phase and frequency of the received signal.

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

Once the carrier is reconstructed, it is sent to the correlator where it is mixed with the received BPSK signal. The output of the multiplier is the sampled baseband data that must be sampled at the appropriate time in order for correct symbol determination. This is accomplished by extracting a clock from the recovered baseband data. This process was made easier by the Manchester encoding that took place in the modulator. Figure 35 illustrates the Manchester data recovered from the mixing of the BPSK signal and the reconstructed carrier.

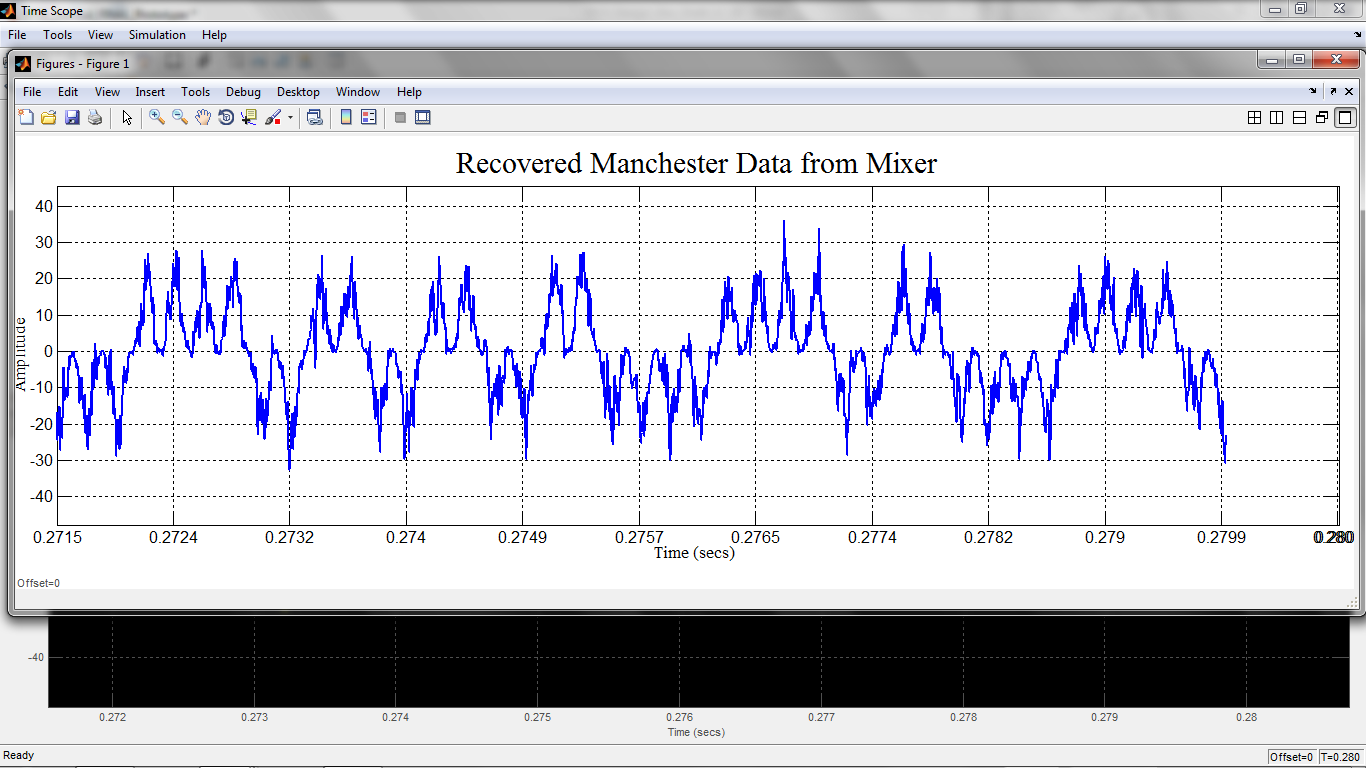


Figure 35. Recovered Manchester data that is forwarded to the early late-gate synchronizer for clock extraction.

The method that was chosen for timing recovery was the closed loop early-late synchronizer. This is a closed loop feedback system similar in operation to a PLL. It continuously tracks and adjusts a local oscillator until the output clock is synchronous with the recovered Manchester data. Its operation is can be explained more clearly by considering the Simulink model of the Early-Late gate circuit.

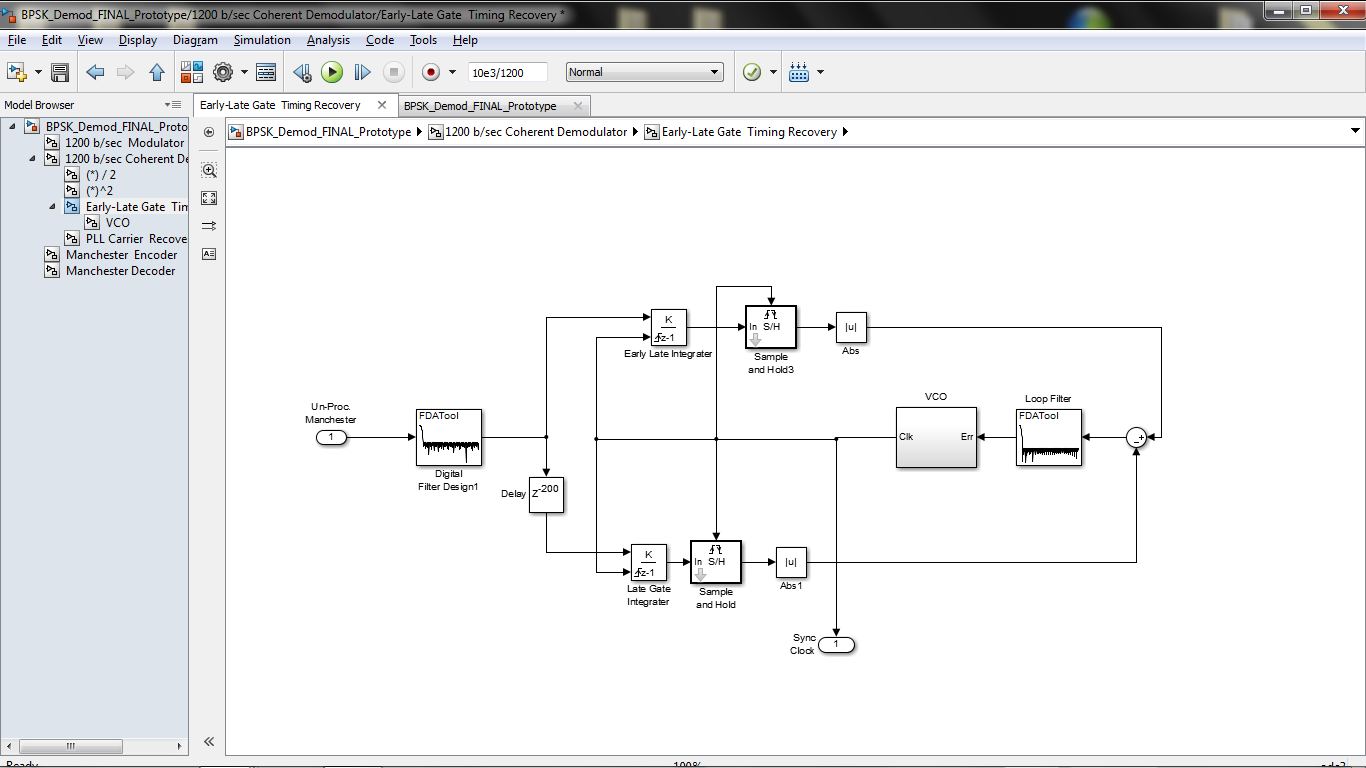


Figure 36. Early-Late Gate synchronizer used for clock extraction

A low pass filter is placed before the Early-Late gate synchronizer so as to minimize the amount of noise in the circuit. The sampled Manchester data takes two paths after being low pass filtered, the first is the Early gate branch (top branch) and the second is the late gate branch (bottom branch). The Early gate branch starts accumulating energy on the rising edge of the VCO clock and is sampled at the falling edge. Conversely, the Late gate branch begins accumulating energy on the falling edge of the clock and samples on the rising edge. Both the early and late branches rectify the samples thus producing positive voltages so that an error signal is derived when the two branches are subtracted.

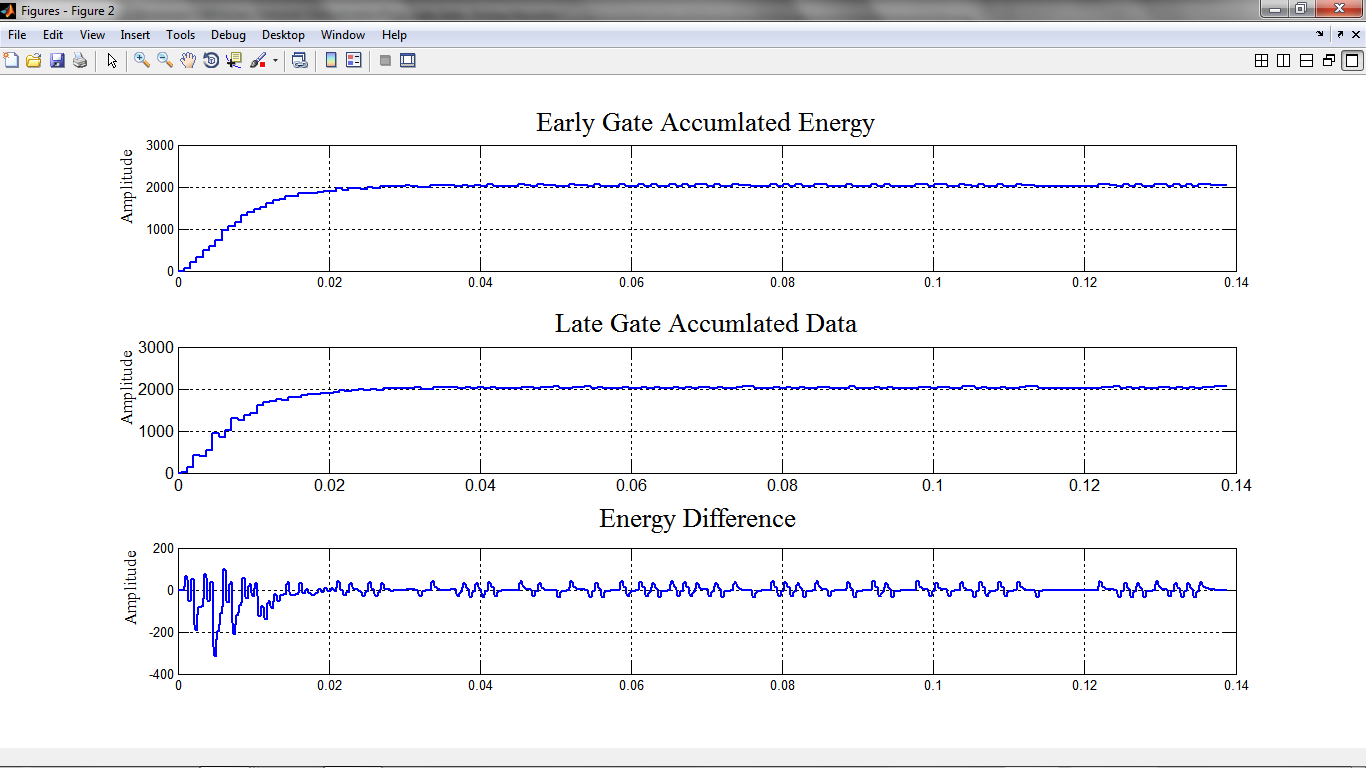


Figure 37. Energy accumulated from the early compared to the energy accumulated from the late gate. The difference in accumulated energy between the early and late gate is used to drive the VCO towards a synchronous clock

Figure 37 depicts the accumulated energy of the early and late gate branches in a addition ot the error signal that feeds the VCO. The results show that in the first 20 msec the early late gate sychronizer is tracking the received signal. This is indicated by the large energy difference in the bottom plot of Figure 37. The transient effect depicted in the accumalted energies in each branch are likely the cause of the start up time of the various filters in the reciever. The output clock of the VCO is compared to the received signal in Figure 38 to illustrate the phase coherency. Since this clock is now sychronized with the transmitted signal, it can be used to sample the recovered manchester data in order to make the correct symbol determination. Note that this subsystem is applicable to both the FSK and BPSK modems since they both use the same bit rate of 1200 b/sec.

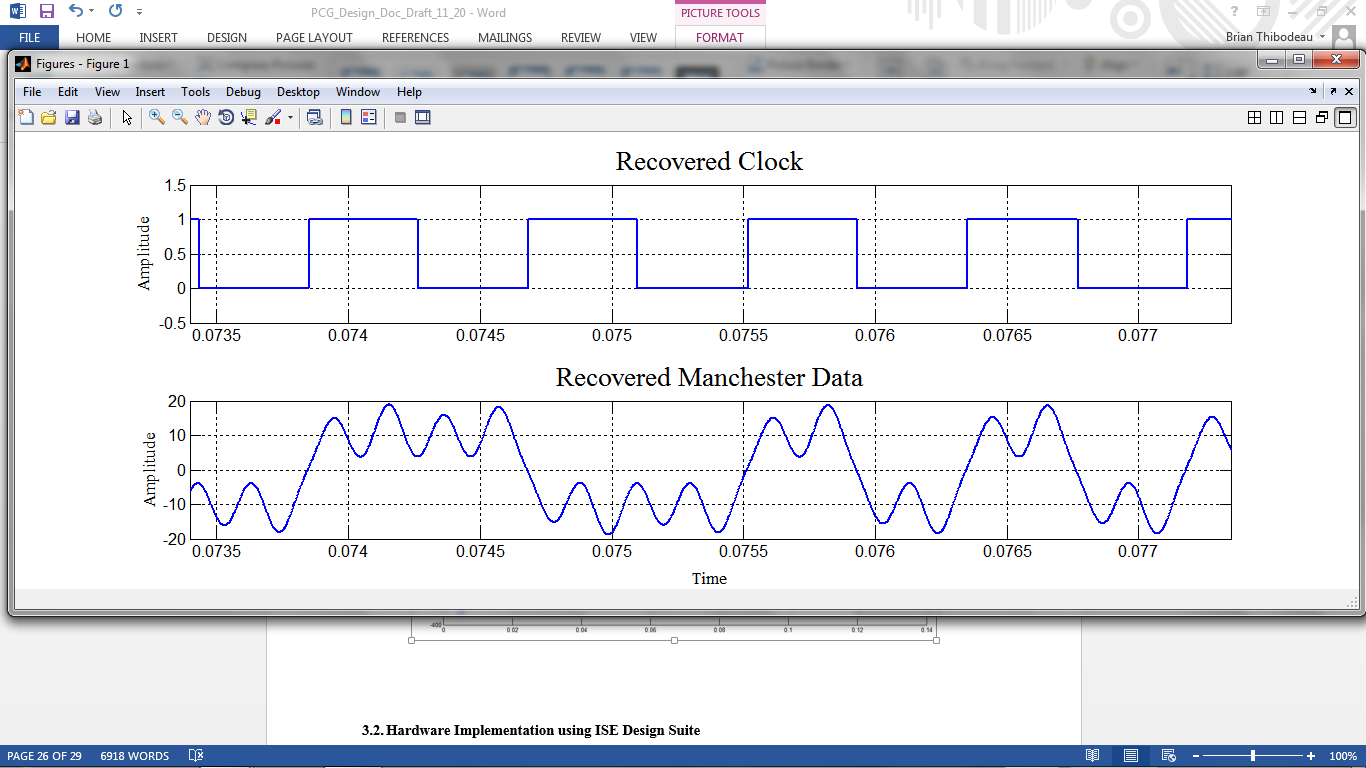


Figure 38. The recovered clock after synchronization is compared to the received Manchester data stream to demonstrate phase coherently

### 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 39), starting with the *Convolutional Encoder* and looping around to the *Viterbi Decoder*.

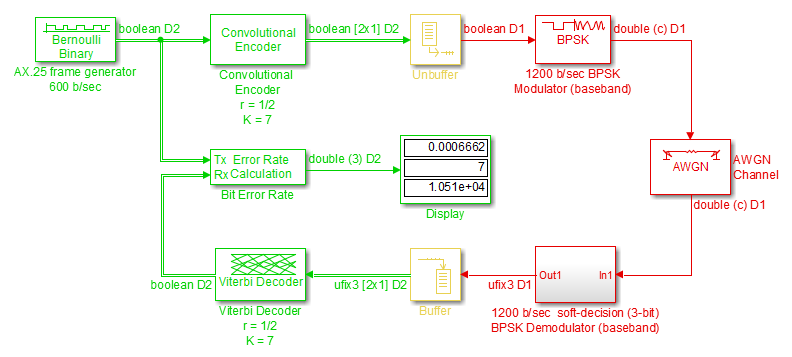


Figure 39. A system-level diagram of System C. A convolutional encoder and Viterbi decoder are used to implement convolutional coding (½ code rate, constraint length 7). Besides from the addition of forward error correction, System C differs from System B in that the BPSK demodulator at the receiver implements 3-bit soft-decision instead of hard-decision.

Figure 39 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.

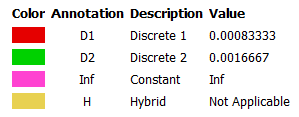


Figure 40. The sample color chart for the system-level Simulink model of System C.

**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 41 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.

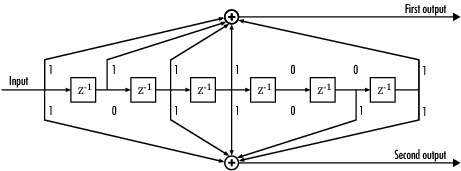


Figure 41. (Courtesy of Matlab®) A convolutional encoder (1/2 code rate, constraint length 7).

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 41, 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 41.

**1200 b/sec BPSK Modulator**

Figure 39 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 39 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).

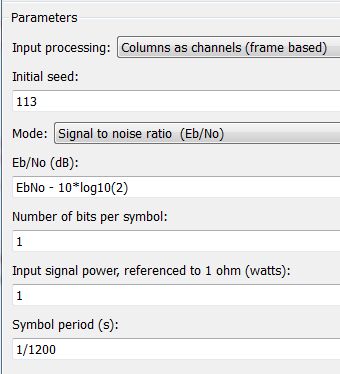
**

Figure 42. 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 43 is a time capture showing 10 bits being transmitted (yellow) and received (red).

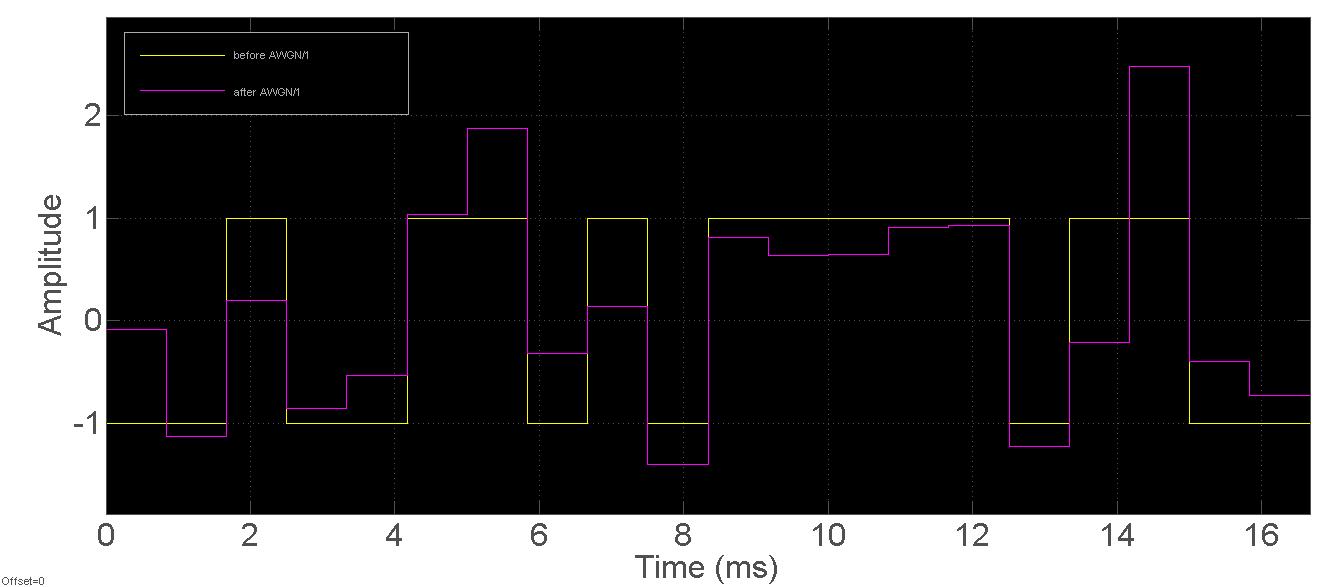


Figure 43. Output of the 1200 b/sec BPSK baseband signal through AWGN. The yellow signal represents the transmitted signal and the red signal represents the received signal (after AWGN).

**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 42 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.

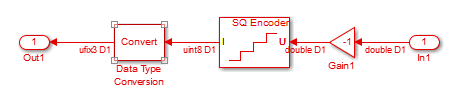


Figure 42. Scalar Quantization Encoder scalar quantizes BPSK demodulator output into a 3-bit value. This 3-bit “soft-decision” output is used by the *Viterbi Decoder.*

**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):

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:

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 43. 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 Design Suite

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