Design Document

Concatenated Codes in

Amateur Radio Satellite Telemetry

Submitted To:

Professor Dennis Silage
Senior Design Project I and II
Temple University
College of Engineering
1947 North 12th Street
Philadelphia, Pennsylvania 19122

nice work!

investigate continuous phase modulation (CPM)

PCG

December 2, 2013
Prepared By:

B. Keith, B. Thibodeau, and C. Destin
Faculty Advisor(s): Dr. Dennis Silage
Programmable Communication Group
Temple University
College of Engineering
1947 North 12th Street
Philadelphia, Pennsylvania 19122

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

EXECUTIVE SUMMARY

1. PROBLEM

NOTE that this document may contains redundant information, but for purpose of documentation, all knowledge has been put on paper with mentality that is better to have more than less. In the future, material will be removed, replaced, or re-organized as seen fit. Furthermore, as a DRAFT design document, much work is still needed to adhere to the senior design formatting requirements. This includes Figures, equations, fonts, first use of acronyms etc. We are aware of this.

1.1. Overall Objectives (Brandon)

It has been shown that forward error correction dramatically improves 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). Being that average amateur satellite telemetry benefits from neither of these facts, this senior design project aims to demonstrate the degree to which forward error correction and interleaving techniques with BPSK modulation can improve the reliability of the average amateur satellite telemetry link. 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~~ (simplex) and does not benefit from automatic repeat request (ARQ) like in other ~~bidirectional~~ (duplex) ~~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 forward error correction combined with interleaving can supersede AFSK, resulting in greatly improved network reliability and power-efficiency in amateur packet radio satellite telemetry. 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).

Hence, the ultimate goal of this senior design project is to demonstrate the improved network reliability and power-efficiency that results from implementing forward error correction and interleaving with BPSK modulation in amateur packet radio telemetry satellites and ground stations.

1.2. Historical and Economic Perspective (Brandon)

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 at least 25 dB of FM receiver quieting (25 dBQ) was necessary for high communication reliability. In other words, 25 dBQ or greater 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

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. Hence, the FEC engine will be limited to using these channel code pairs.

In Section 1.3.5, we discussed that the satellite communication link is vulnerable to random errors and burst errors. 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 a two-level coding system, are used in many systems to provide robustness against both kinds of errors (see Figure 9). This two-level coding system 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 an 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 de-interleaver, then a Reed-Solomon decoder.

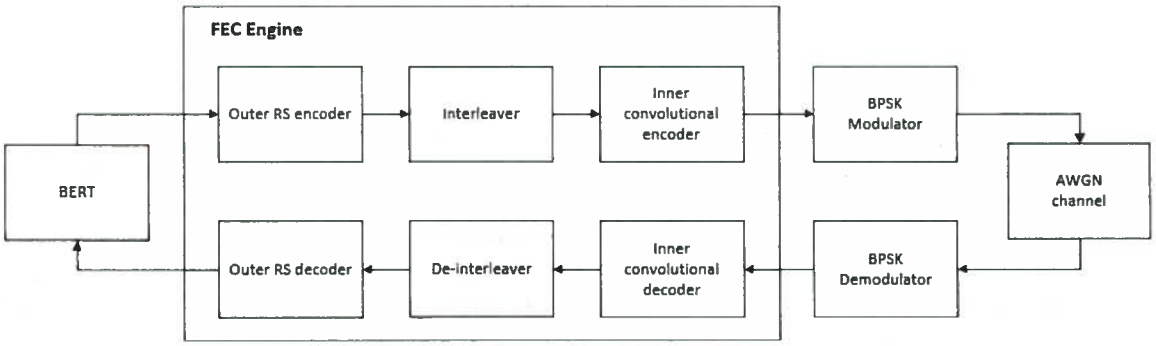


Figure 9. Top-level diagram for the FEC engine (System C) consisting of a block code pair, an interleaving pair, and a convolutional code pair.

1.3.2. Line Coding: Non Return Zero and Manchester

1.3.3. BPSK Carrier Recovery (Brian)

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. 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 1 illustrates the architecture of a typical coherent demodulator.

Design Document

Concatenated Codes in Amateur Radio Satellite Telemetry

Submitted To:

Professor Dennis Silage
Senior Design Project I and II
Temple University
College of Engineering
1947 North 12th Street
Philadelphia, Pennsylvania 19122

nice work!

investigate continuous phase modulation (CPM)

PCG

December 2, 2013
Prepared By:

B. Keith, B. Thibodeau, and C. Destin
Faculty Advisor(s): Dr. Dennis Silage
Programmable Communication Group
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	Brandon Keith, Brian Thibodeau, Cedric Destin	
Advisor(s)	Dennis Silage	
Coordinator	Thomas Sullivan	
Department(s)	Electrical and Computer Engineering	
Project Title	Concatenated Codes in Amateur Radio Satellite Telemetry	
Abstract	TBD	
URL	https://sites.google.com/a/temple.edu/programmable-communication-group/	

EXECUTIVE SUMMARY

Table of Contents

1. PROBLEM

NOTE that this document may contains redundant information, but for purpose of documentation, all knowledge has been put on paper with mentality that is better to have more than less. In the future, material will be removed, replaced, or re-organized as seen fit. Furthermore, as a DRAFT design document, much work is still needed to adhere to the senior design formatting requirements. This includes Figures, equations, fonts, first use of acronyms etc. We are aware of this.

1.1. Overall Objectives (Brandon)

It has been shown that forward error correction dramatically improves 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). Being that average amateur satellite telemetry benefits from neither of these facts, this senior design project aims to demonstrate the degree to which forward error correction and interleaving techniques with BPSK modulation can improve the reliability of the average amateur satellite telemetry link. 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~~ (simplex) and does not benefit from automatic repeat request (ARQ) like in other ~~bidirectional~~ (duplex) ~~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 forward error correction combined with interleaving can supersede AFSK, resulting in greatly improved network reliability and power-efficiency in amateur packet radio satellite telemetry. 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).

Hence, the ultimate goal of this senior design project is to demonstrate the improved network reliability and power-efficiency that results from implementing forward error correction and interleaving with BPSK modulation in amateur packet radio telemetry satellites and ground stations.

1.2. Historical and Economic Perspective (Brandon)

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 at least 25 dB of FM receiver quieting (25 dBQ) was necessary for high communication reliability. In other words, 25 dBQ or greater 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 alone is not robust enough for amateur radio microsatellite communications (Hsiao, et. al, 2000). Particularly in simplex satellite communications, the harsh environmental conditions coupled with the microsatellite's characteristically low transmitter power made for very unreliable telemetry data links (Hsiao, et. al, 2000). It has been demonstrated that forward error correction, specifically convolutional encoding and decoding, 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 detection and correction. However, it operated at 437.9 MHz, meaning that it was particularly difficult for the average amateur radio operator with a 2-meter radio transceiver to 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 forward error correction and interleaving schemes with BPSK modulation in amateur satellite telemetry.

1.3. Candidate Solutions (Brian)

Typical functions of a modem include forward error correction, source encoding, modulation, demodulation and source decoding. 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.

1.3.1. Forward Error Correction: Block and Convolutional Codes (Brandon)

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. FEC codes fall into two general categories: block codes and convolutional codes. It is important to note that 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. Hence, the FEC engine will be limited to using these channel code pairs.

In Section 1.3.5, we discussed that the satellite communication link is vulnerable to random errors and burst errors. 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 a two-level coding system, are used in many systems to provide robustness against both kinds of errors (see Figure 9). This two-level coding system 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 an 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 de-interleaver, then a Reed-Solomon decoder.

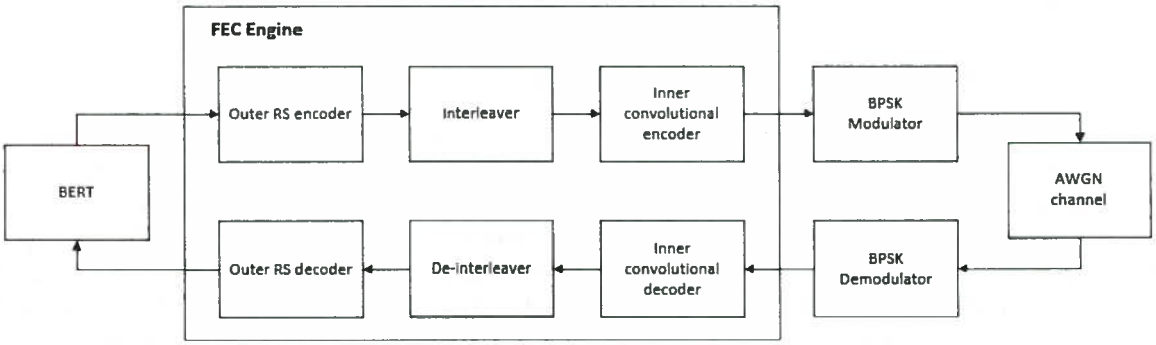


Figure 9. Top-level diagram for the FEC engine (System C) consisting of a block code pair, an interleaving pair, and a convolutional code pair.

1.3.2. Line Coding: Non Return Zero and Manchester

1.3.3. BPSK Carrier Recovery (Brian)

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. 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 1 illustrates the architecture of a typical coherent demodulator.

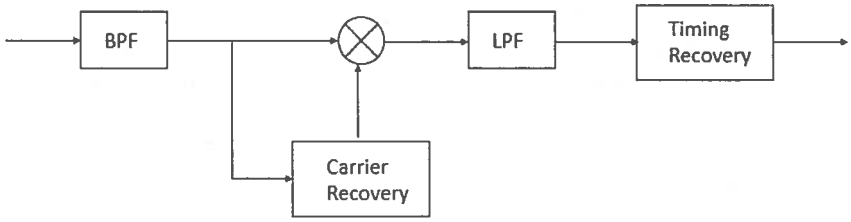


Figure 1. Received waveform takes two paths. First path extracts carrier for coherent demodulation and the second path recovers timing information. This architecture is based on the optimum binary receiver

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 bandpass 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. The operation of the squaring is shown in Figure 2.

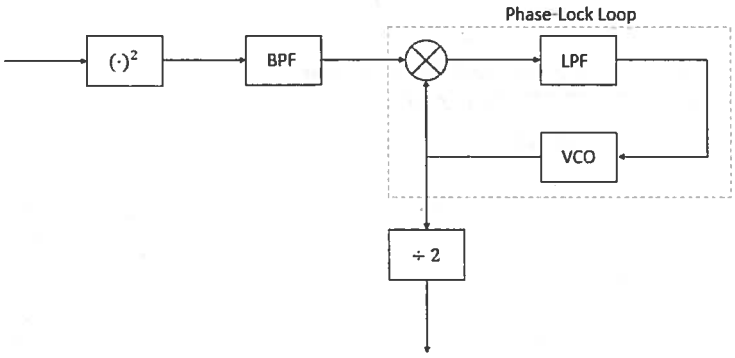


Figure 2. Squaring loop used for carrier recovery in the coherent demodulator. The Phase-Lock Loop utilizes feedback to track and lock onto in the received waveforms suppressed carrier

Carrier Recovery using Costas Loop

Another method for carrier recovery was proposed by John P. Costas in his 1957 paper, *Synchronous Communication*. 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. 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.

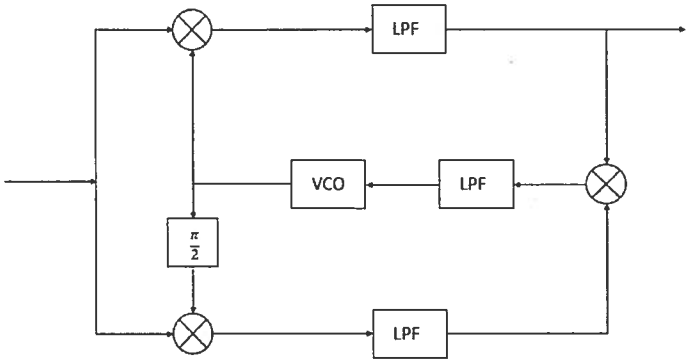


Figure 3. Costas loop used for suppressed carrier reconstruction as well as synchronous data detection.

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. The decision for implementing the squaring loop versus the Costas loop will ultimately be decided by their tracking and locking performance in the presence of noise and Doppler shifts (See section 1.5, Major Design and Implementation Challenges).

1.3.4. Coherent and Non-Coherent BFSK Modulation (Cedric)

The BFSK modem abides by the Bell 202 standard which uses frequencies of 1200 Hz for Mark (b_0) and 2200 Hz for Space (b_1). 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' phase, hence the modulated signal may experience discontinuous jumps in phase.

Coherent FSK modulators tend to consist of several complicated components, and therefore 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 modulation despite its BER performance compared to the coherent modulation. However, with the technological development, coherent modulation can surely be implemented with as much efficiency as the non-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 or two sine functions and a multiplexer controlled by the input data *m(t)*. Switching between the frequencies will generate a BFSK waveform with a bit period equal to the periodicity of the switches.

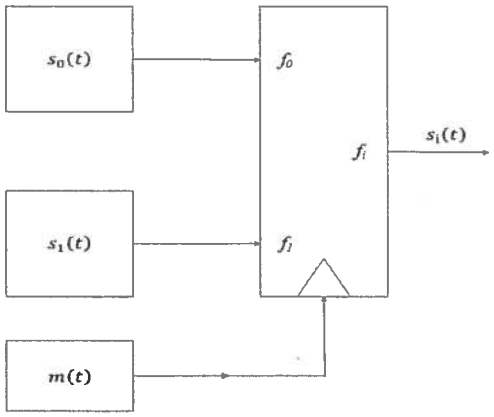


Figure 4. BFSK modulator used in non-coherent modulators. The data *m(t)* controls the output of the multiplexer at data's baud rate.

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 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.

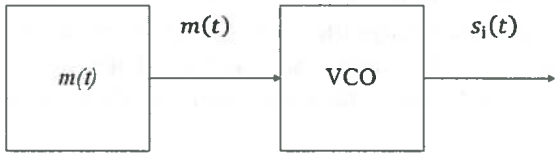


Figure 5. Coherent modulator for BFSK. The data *m(t)* controls the output of the VCO through Eq. 3

$$v_{vco} = \sin\left(2\pi t\left[f_Q + k \cdot u(t)\right]\right) {}^{Hz}/_{V} \quad (1)$$

1.3.5. Coherent and Non-Coherent BFSK Demodulation (Cedric)

The Bell 202 Protocol is quite complex due to the frequency deviation f_d 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 f_c to represent it binary data b_0 and b_1 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 (2) and (3) below (Nguyen, et. al, 2009).

$$\Delta f_{coherent} = \frac{1}{2T_b} \quad (2)$$

$$\Delta f_{non-coherent} = \frac{1}{T_b} \quad (3)$$

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 loops or more complicated systems as illustrated in Figure 6.

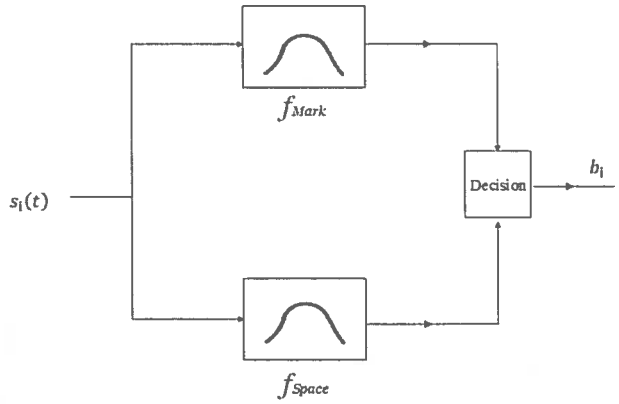


Figure 6. Coherent modulator for BFSK. The data *m(t)* controls the output of the VCO through Eq. 3

The coherent demodulator in Figure 6 uses two parallel branches for matching the Space and Mark unto the two orthonormal basis functions φ_1 andφ_2. Finally, using the appropriate threshold and decisions, the bits can be recovered using Maximum Likelihood. The correlation receivers or matched filters φ_1 and φ_2 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 using the following relationship:

$$\hat{\varphi}(t) = \frac{\varphi_1(t) - \varphi_2(t)}{\sqrt{E_1 - 2\sqrt{E_1 E_2} - E_2}} \quad (4)$$

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)

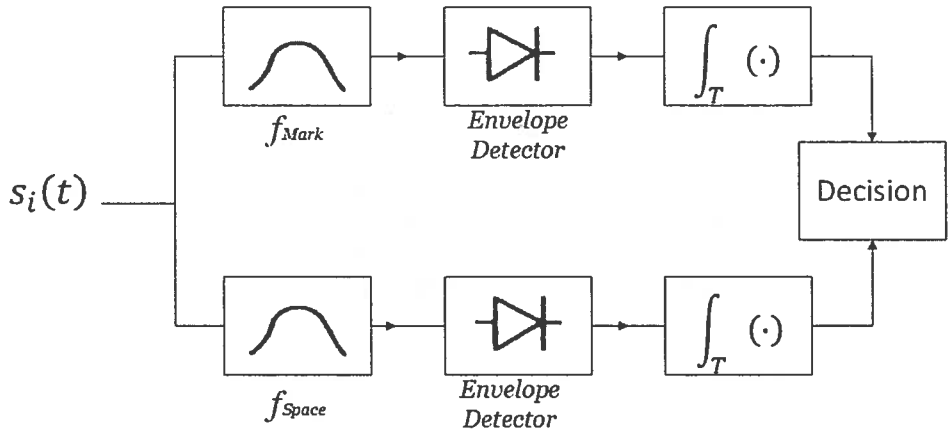


Figure 7. Non-coherent demodulator for BFSK, where the Mark and Space filters are centered at 2200Hz and 1200Hz respectively.

Non-coherent demodulation (PLL)

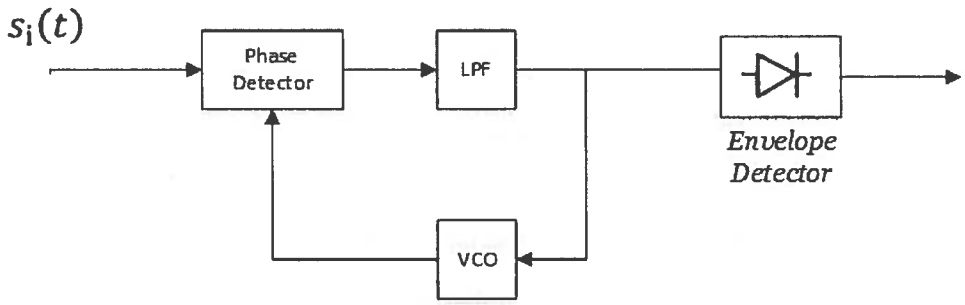


Figure 8. Non-coherent demodulator for BFSK using PLL.

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 (.) 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.

1.3.6. BPSK and BFSK Timing Recovery

This section will introduce and discuss two common circuits for timing recovery. The first is the open loop architecture that delays the received signal by one half a bit time and then XOR's it with the non-shifted received signal. The second candidate solution is the closed loop early-late gate synchronizer.

1.4. Proposed Solution Concept (Brandon 80% /Brian 20%),

This senior design project will determine the telemetry packet error rate (PER) performance and the coding gain due to implementing forward error correction in amateur radio satellite telemetry. These two parameters, respectively, will allow us to compare the reliability and power-efficiency between using and not using FEC in amateur radio satellite telemetry. This implies that we will be comparing several digital communication systems. In this senior design project, three digital communication systems will be

developed for modeling amateur radio satellite telemetry. The first system (System A) will replicate most amateur radio satellite telemetry links that exist today – 1200 b/sec Bell 202 modulation without FEC. This FSK modem will perform Manchester encoding and coherent modulation. Demodulation in the FSK modem will also be done coherently through the use of a PLL.

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 squaring loop circuit. Timing recovery is done through the early-late gate circuit which extracts the clock embedded within the Manchester code.

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 both block and convolutional coding. The addition of interleaving between the block and convolutional encoding will further improve the reliability of the BPSK modem

1.5. Major Design and Implementation Challenges

1.6. 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, Phil Karn, KA9Q, in his efforts to secure the future of amateur radio. Like Jeff Davis, Phil Karn is also aware of amateur radio's *lost future*. In a modem design article (Karn, 2011), 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 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 of a communication link, you can consequently get away with communicating at a lower signal-to-noise (SNR) ratio (Sklar, 2001). This results in lower transmitted power between an amateur radio satellite and ground station. Being that the power amplifier of the transmitter utilizes the most power of an amateur radio satellite, the lower power requirement could result in cutting the cost of satellite construction and simplify the ground antennas (Karn, 2011). Consequently, amateur radio satellite communications would 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 power-efficient 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 and concatenated error-correcting codes can make amateur radio satellite communications more power-efficient and hence, more *accessible* to prospective amateur satellite operators. Perhaps a simple BER performance analysis of popular and prospective communication schemes, like showcased in this senior design project, would further persuade an amateur satellite designer to employ more power-efficient communication schemes in the increasing fleet of miniaturized amateur radio satellites.

2. DESIGN REQUIREMENTS

2.1. Functional Design Constraints (Brian)

Name	Description
Bit Rate	Provides 1200 b/sec data rate to meet LEO-AMSAT telemetry requirements
FEC	
Modulation Type	Supports BPSK and FSK modulation and Demodulation
Operating Frequencies	The modulator and demodulator will provide operation between 1200 Hz and 2400 Hz in accordance with the Bell 202 standard
Interface	
Signal to Noise (SNR)	
Bit Error Rate	

Table 1. Functional design constraints for the GADGET system.

2.2. Non-Functional Design Constraints (Brian)

Type	Name	Description
Economic	Cost	
Environmental	Temperature	
Environmental	Power Consumption	
Manufacturability	Dimensions	
Manufacturability	Weight	

Table 3. Non-functional design constraints for the GADGET system.

3. APPROACH

Here, will be an introduction and summary of our design approach from high-level Simulink blocks to lower level hardware realizable Xilinx blocks, and then finally delving into hardware implementation. The approach is divided into TWO major sub-sections, *Software Simulation Using Matlab/Simulink* and *Hardware Implementation Using Xilinx ISE Design Suite.* The first subsection WILL be completed by DEC 2. The second sub-section will be completed in SDII.

3.1. Software Simulation Using Matlab/Simulink

3.1.1. FSK Modem (Cedric)

The binary digits from the computer (TNC) are abstract values and need to be converted to tangible waveforms. In Wireless communication, Manchester coding has established itself as a standard signaling technic among the several others. Signal technics 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 being, hence it may be referred as a self-clocking signaling technic.

Manchester code is also not a complicated signal scheme to implement and needs few components to obtain the self-clocking behavior. Hence, Manchester code has gained a great amount of popularity among communication engineer being implemented in various Amateur Radio communication and also has been a standard protocol for Ethernet. The IEEE standard protocol maps the binary values $b_0 = 0$ and $b_1 = 1$ 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 # illustrates the protocol from the IEEE 802.3 protocol where $b_0 = CD0$ and $b_1 = CD1$.

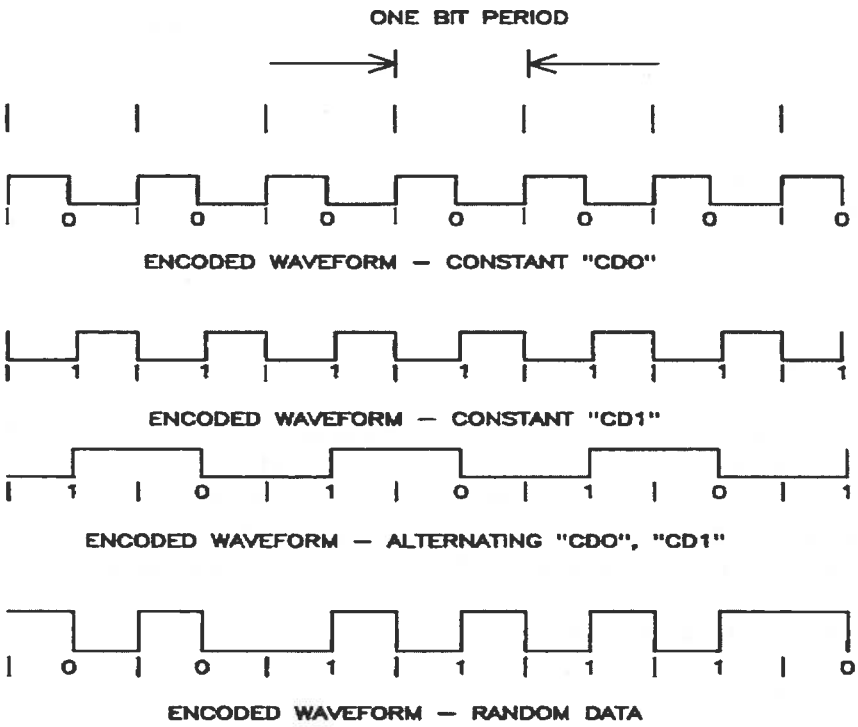


Figure #: Followed protocol of the Manchester code based on the IEEE 80.3 Ethernet communication

Implementing the Manchester line code can be done either using switches or the XOR logical operator ($\otimes$) as depicted in the Figure # + 1. 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 bps. Further modification can also be done by altering the magnitude values of the signal waveform, either

$$m(t) = \{0\,;\, \mathrm{A}\}\,\mathrm{V}$$
$$m(t) = \{-A\,;\, A\}\,\mathrm{V}$$

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

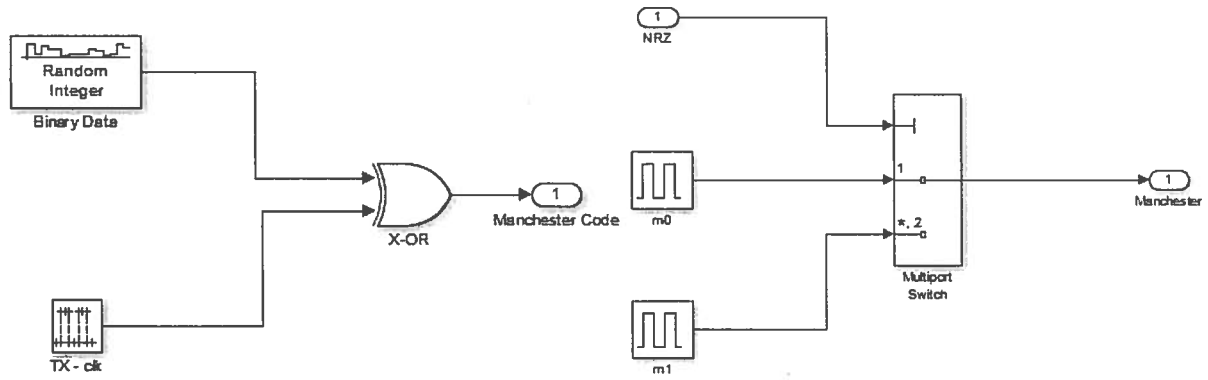


Figure # + 1: Manchester encoder using an XOR gate (left) Multiplexer (right)

At the receiver, the data must also be decoded using the Manchester code basis function denoted as $\varphi(t)$. Decoding the Manchester code yields very accurate results due its efficient decoding properties. The correlation coefficient between the two signaling waveforms equals to $\rho = -1$, 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.

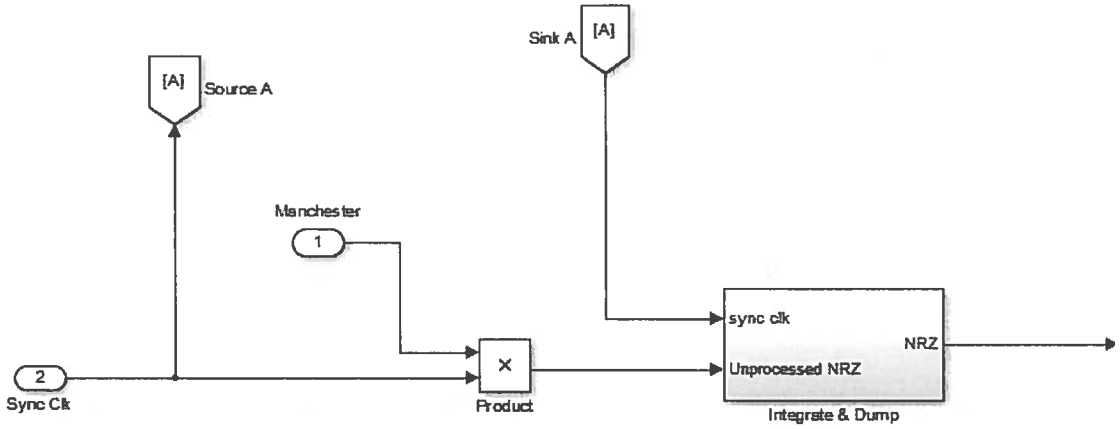


Figure # + 2: The decoder for the baseband Manchester is illustrated above, where the product block and the integrate and dump make up the corellation receiver for the Manchester code.

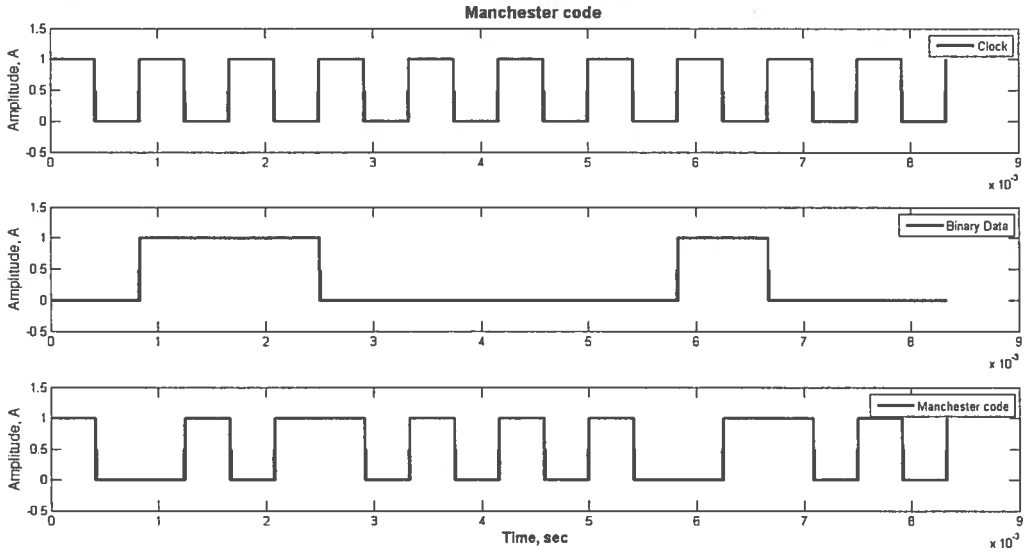


Figure # + 3: Results of the Manchester code

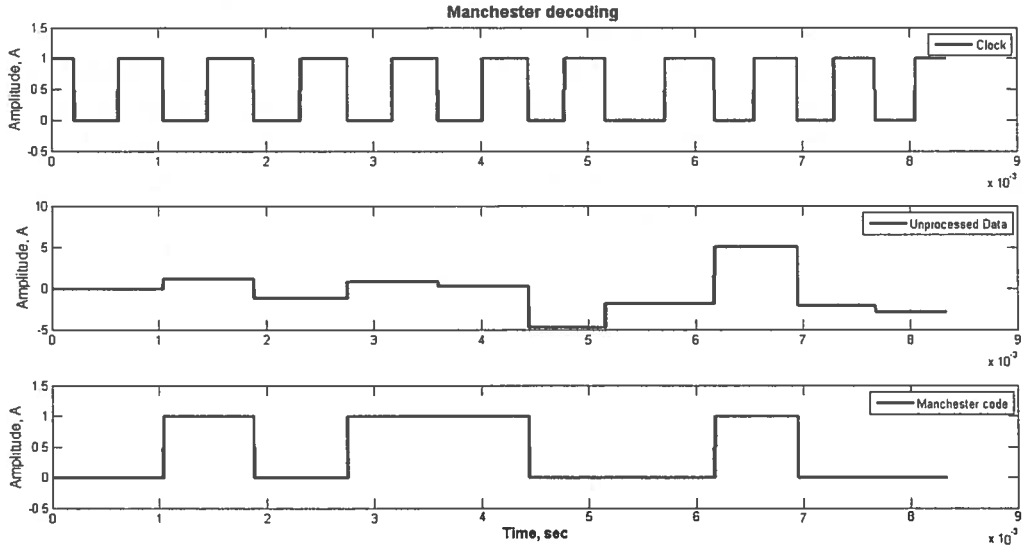


Figure # 4: Results of the Manchester decoding

3.1.1.1 Modulation

Modulation may be implemented either coherently or non-coherently where the linerarity 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 # - 4 can be used to obtain the behavior of the non-coherent modulator. In Simulink, the modulator is obtained using a switch to let through the tone corresponding to the data transmitted. The phase is described to have a random phase modeled as uniform from $[-\pi, \pi]$.

In contrast the coherent modulator may be more complex to implement since a continuity is desired between every transitions. Although a coherent signaling may be complex, a superior error rate is gained compared to that of the non-coherent signal scheme, which by definition is defined as

$$P_{error} = Q\left(\sqrt{E_b/N_0}\right) \text{ Eq \#}$$

Obtaining continuity in the modulated involves "remembering" the phase of the previous tone and may require memory components. An alternative to this complicated methods is to use a Voltage Controlled Oscillator (VCO) to modulate the incoming Manchester coded waveform. The modulator can then be defined as:

$$s(t) = A \cdot \cos(w_{VCO}t + 2\pi \int_0^{\tau} u(\tau)\, d\tau \quad \text{Eq \# + 1}$$

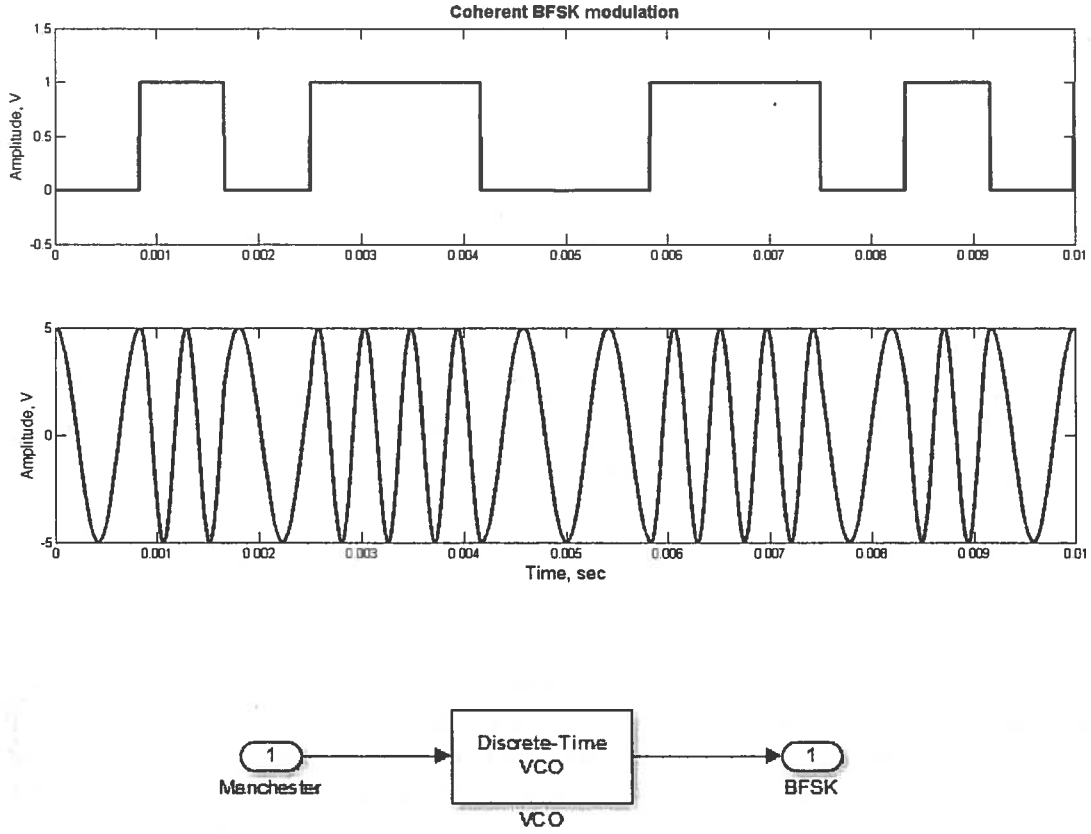
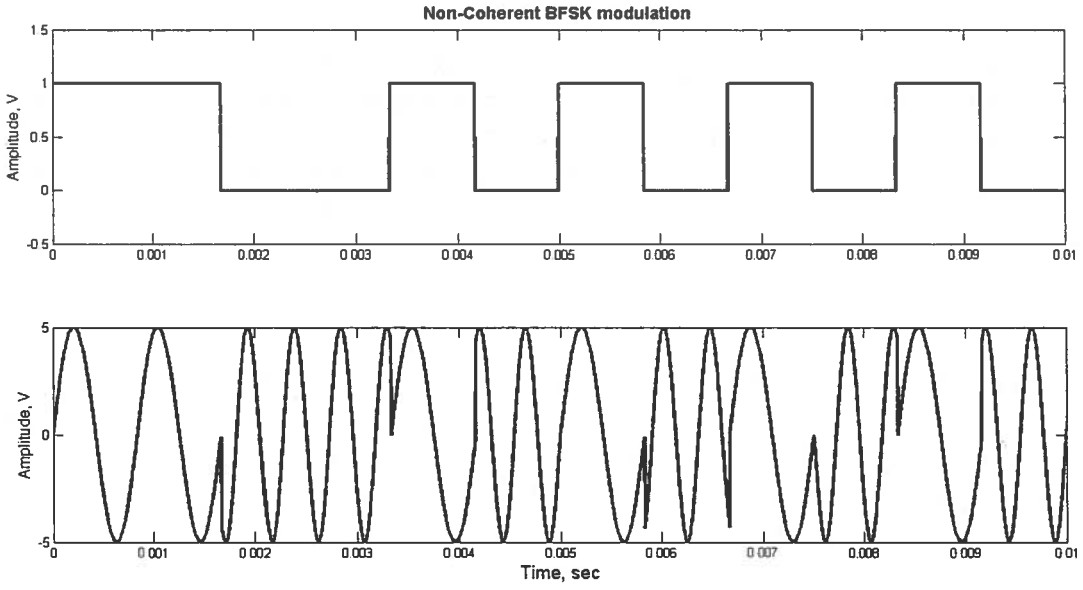


Figure #: BFSK coherently modulated using a VCO



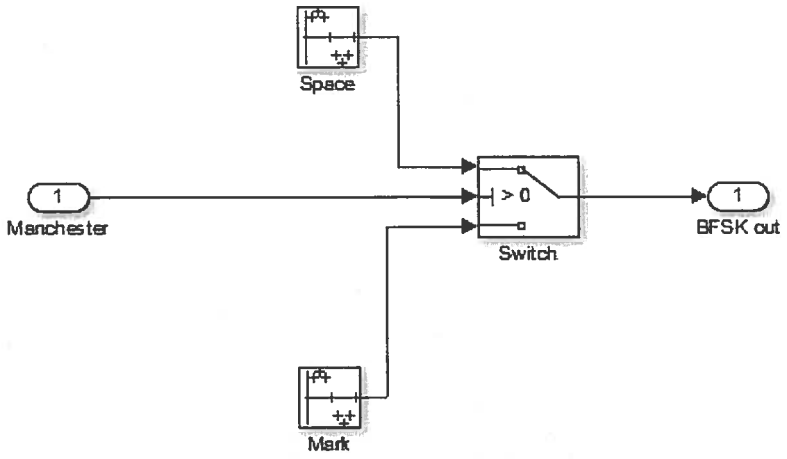


Figure #: BFSK non-coherently modulated using switches

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 us to modulate the two frequencies. 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 # 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 which will complicate the demodulation process at the receiver. Hence the demodulator of the BFSK modem will consist of a coherent demodulator using a PLLL to keep track of the random phases the signal may undergo. As discussed by (Lindsey et al) the PLL estimates the frequency of 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 K_D 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.

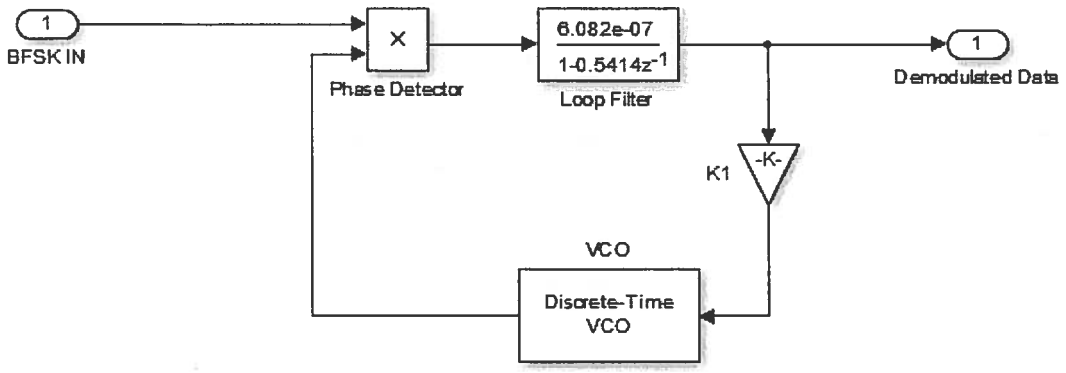


Figure #: Model of the PLL for BFSK demodulation in Simulink

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 $F(s)$ 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 as the following:

$$H(s) = \frac{s \cdot K \cdot F(s)}{s + K \cdot F(s)}$$

As stated in numerous literature (Gardner) the loop filter must contain all of the modulated frequencies. Simplifying the equation above

$$H(s) = \frac{s \cdot K\left({}^{1}/_{s\tau_1} + 1\right)}{s + K\left({}^{1}/_{s\tau_1} + 1\right)}$$

$$H(s) = \frac{s}{s^2 + {}^{s}/_{\tau_1} + {}^{K}/_{\tau_1}}$$

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

$$H(s) = \frac{s \cdot {\omega_n}^2}{s^2 + 2\zeta\omega_n s + {\omega_n}^2}$$

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:

$$2\zeta\omega_n = \frac{1}{\tau_1} = \omega_{cut}$$

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

$$\omega_{cut} = 20 \cdot 1200\ {}^{rad}/_{sec} = 24000\,{}^{rad}/_{sec}$$

The parameters of the equation are found as follow:

Where the damping ratio is chosen to be $\zeta = 0.707$, and the natural frequency is calculated to be:

$$\omega_n = {}^{\omega_{cut}}/_{(2 \cdot 0.707)} = 1.0665e + 05\, {}^{rad}/_{sec}$$

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 # + 1

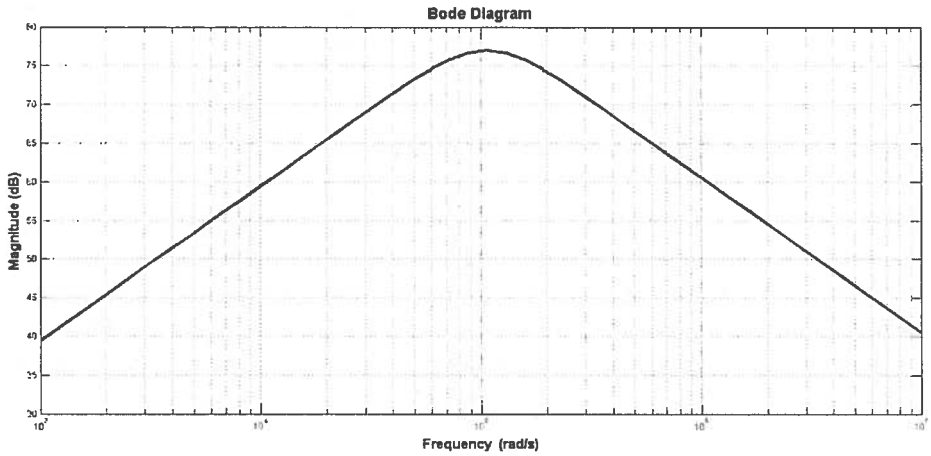


Figure # + 1: Bode plot of the PLL for FSK demodulation

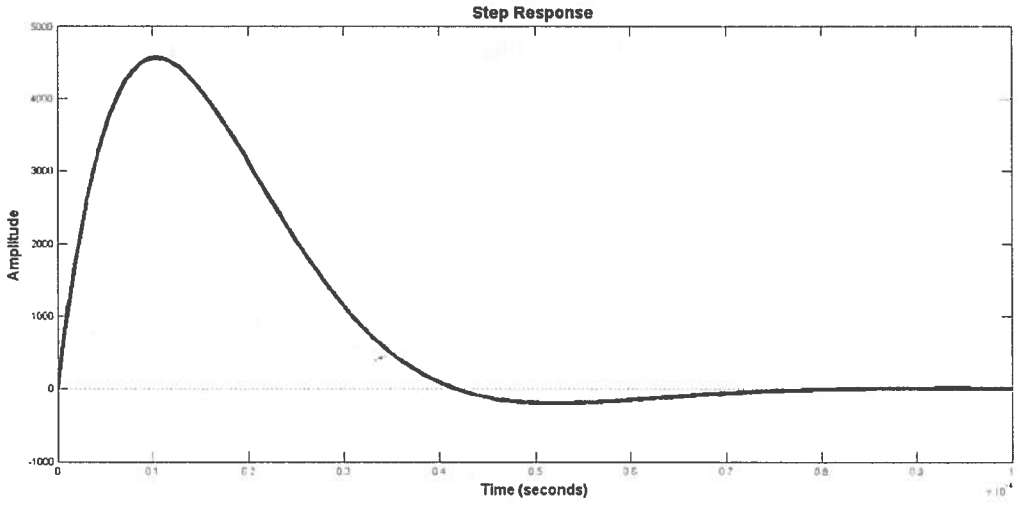


Figure # + 2: Step response of the PLL for FSK demodulation

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.

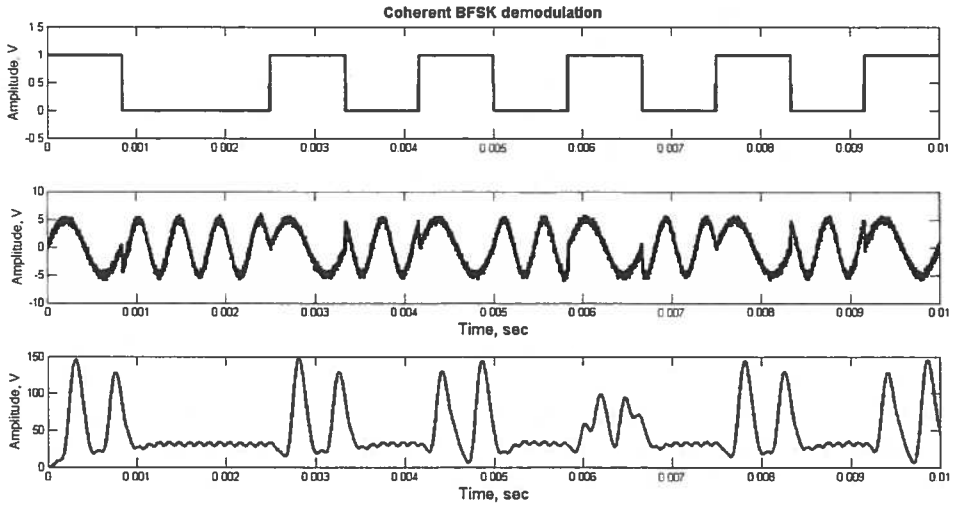


Figure # + 3: Unprocessed data from the PLL Coherent demodulator

3.1.2. BPSK Modem (Brian)

The BPSK modem was designed to accomplish Manchester encoding, Modulation, and coherent Demodulation. As discussed earlier, coherent demodulation of BPSK requires that the phase and frequency of the carrier be known in order to properly demodulate a BPSK signal. In this design, the squaring loop was chosen for its operational and mathematically simplicity. Correct demodulation of BPSK also requires a synchronized clock that controls the integrate and dump in the correlator. This clock is extracted using the closed loop early-late circuit. The locking of the early-late is improved because of the self-clocking characteristics of the Manchester codes.

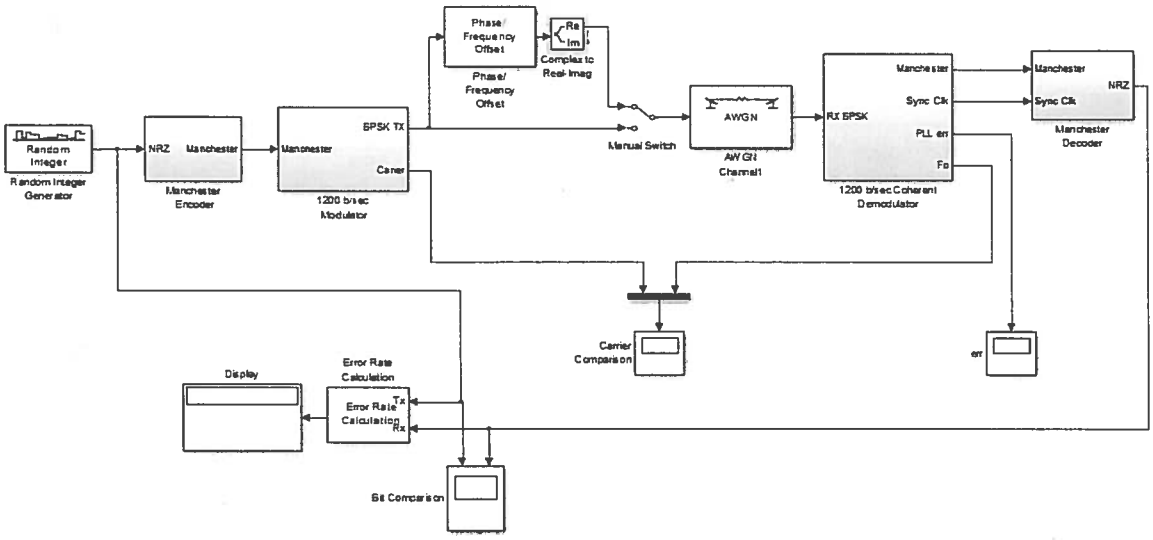


Figure 9. Simulink model used for design and evaluation of the BPSK modem. Major systems include Manchester encoder, 1200 b/sec Modulator, 1200 b/sec coherent demodulator, and Manchester decoder.

The rest of this section will be dedicated to discussing the design and performance of the primary systems that compose that BPSK modem. This will begin with the Manchester encoder, followed by the BPSK

modulator and then towards demodulation which will be broken down into carrier recovery and timing recovery. As carrier and timing recovery are the most critical functions of the modem, the majority of the discussion will be spent thoroughly examining their operation and performance. The design approach for the BPSK modem will conclude with the Manchester decoder. Note that all simulations in the following sections were performed using a simulation frequency of 480 kHz. This was chosen because it is an integer multiple of the carrier frequency which was designed to be 2400 Hz. This will be discussed further in the appropriate section.

Manchester Encoder

Although there are many ways to produce Manchester code in Simulink, it was decided to use the conventional XOR method in order to keep the design as hardware realizable as possible (Nguyen & Shwedyk, 2009). In this method, the input NRZ data stream is XOR'd with the TX clock and the output of the XOR gate is the Manchester encoded data stream. Figure 10 illustrates the Simulink model and figure 11 demonstrates the operation of the Manchester encoder through use of a timing diagram.

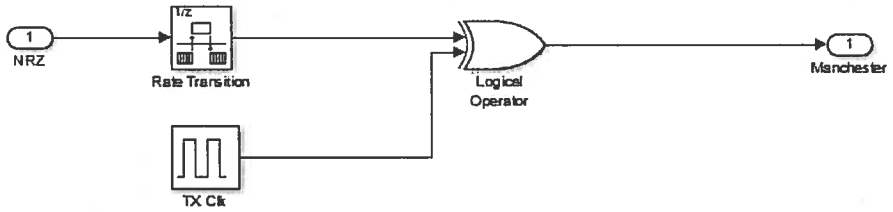


Figure 10. Exploded view of the Manchester Encoder subystem.

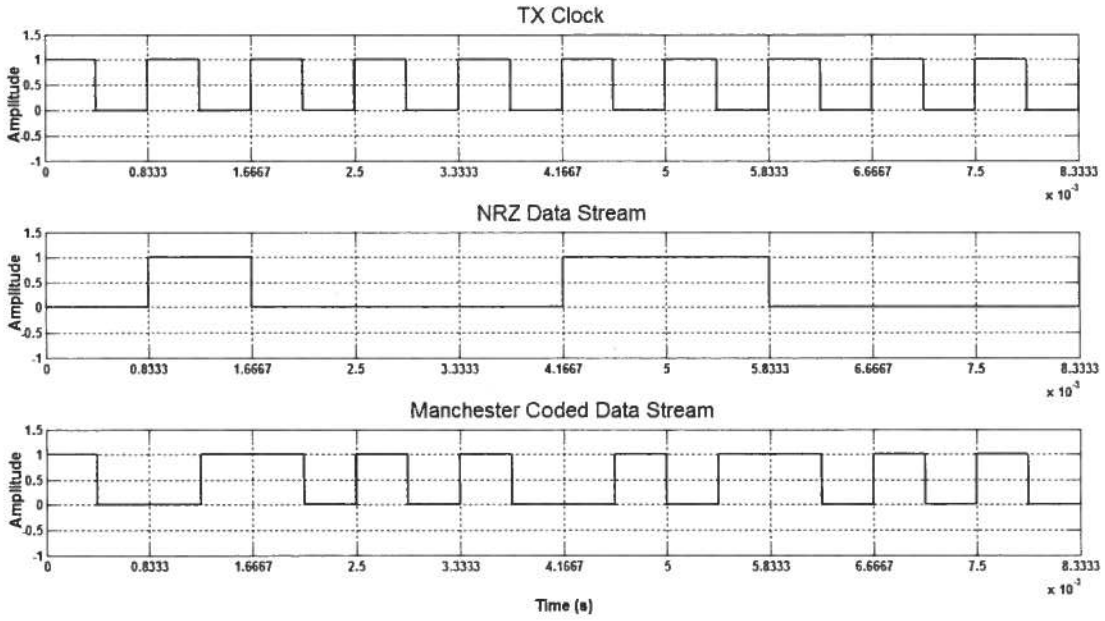


Figure 11. Timing diagram illustrating the operation of the Manchester encoder.

From Figure 11 it is seen that a logic '0' is encoded as a high to low transition during the first half of the

bit period and a logic '1' is encoded as a low to high transition during the second half of the bit period (IEEE std 802.3, 2012).

1200 b/sec 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 $s(t)$, represented mathematically as:

$$s(t) = \begin{cases} A\sin(2\pi f_c t), & logic\ '1' \\ A\sin(2\pi f_c t + \pi), & logic\ '0' \end{cases}. \quad (1)$$

Then exploiting the fact that $\sin(2\pi f_c t + \pi) = -\sin(2\pi f_c t)$, the expression for the transmitted BPSK can be re-written as:

$$s(t) = \begin{cases} A\sin(2\pi f_c t), & logic\ '1' \\ -A\sin(2\pi f_c t), & logic\ '0' \end{cases}, \quad (2)$$

where A is the Amplitude and f_c is the carrier frequency of the transmitted BPSK signal. From equation (2), the Simulink model of the BPSK modulator was designed to gate two antipodal sinusoidal carriers with $A = 5\ V$ and $f_c = 2400\ Hz$. This is shown in Figure 12.

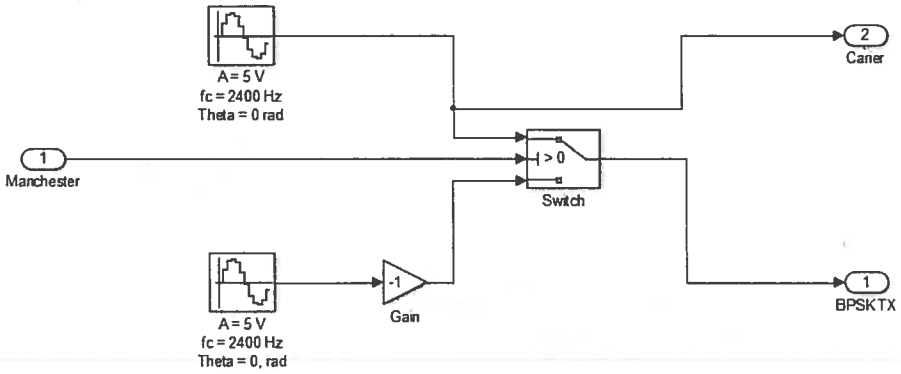


Figure 12. BPSK modulator that uses the Manchester data stream to gate two antipodal sinusoidal carriers that result in the BPSK modulated signal.

Figure 13 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.

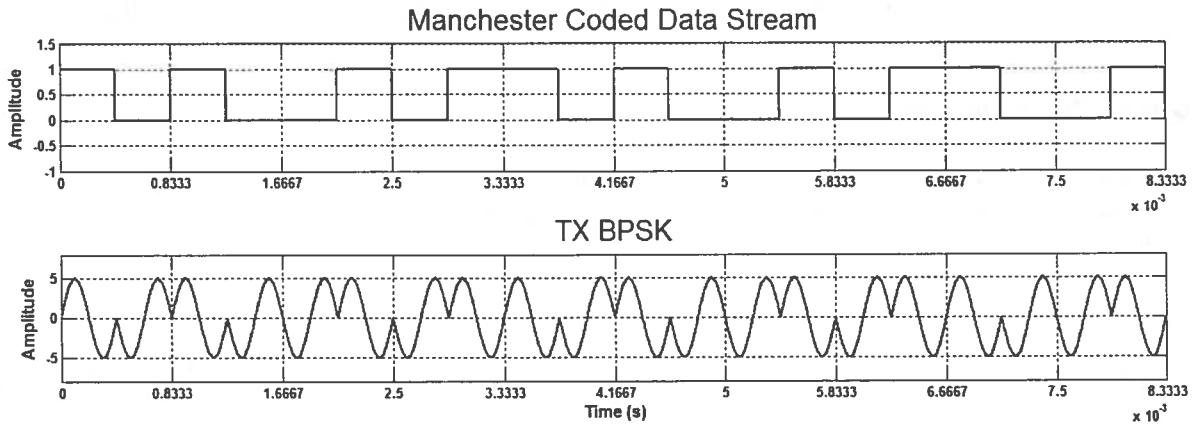


Figure 13. The transmitted BPSK signal has 180 degree phase shifts that correspond to logic level transition of the Manchester coded data stream.

1200 b/sec Coherent Demodulator

The Simulink model of the squaring loop demodualtor that was introduced in section 1 is shown in figure 14. The received signal is first passed through an IIR peaking filter with a single peak at 2400 Hz and a bandwidth of $10/T_b = 24$ kHz. 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.

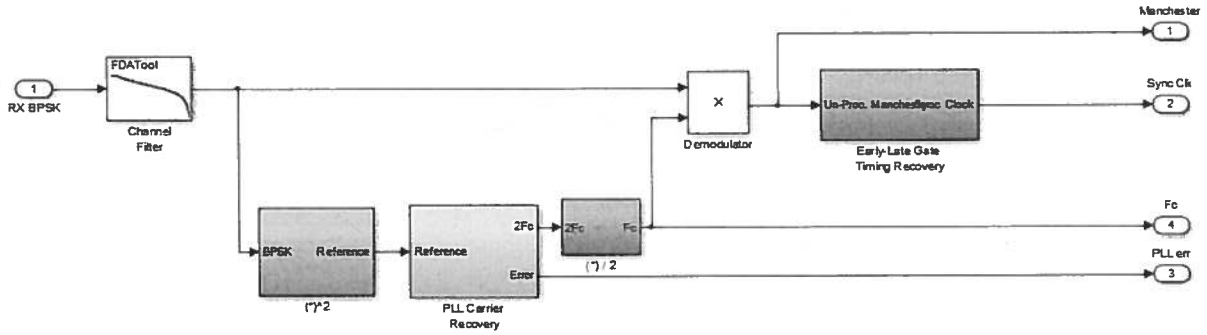


Figure 14. Simulink model of the Squaring loop Demodulator. The carrier of the received signal is reconstructed using a PLL where it is used for coherent demodulation. The early-late circuit regenerates the clock for symbol timing.

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 $2f_c$ (Nguyen & Shwedyk, 2009). This is easily shown using trigonometric identities. If the received signal is $r(t) = s(t) + w(t)$ where $w(t)$ is AWGN, then,

$$r^2(t) = A^2\sin^2(2\pi f_c t) + \textbf{noise terms}$$

$$= \frac{A^2}{2} - \frac{A^2}{2}\cos(4\pi f_c t) + \textbf{noise terms} \qquad (3)$$

The DC component and the noise in equation (3) are filtered using a narrow bandpass filter with a center frequency of $2f_c = 4800\ Hz$ and a Q of 100. The resulting PSD after squaring and filtering is displayed

in Figure 15.

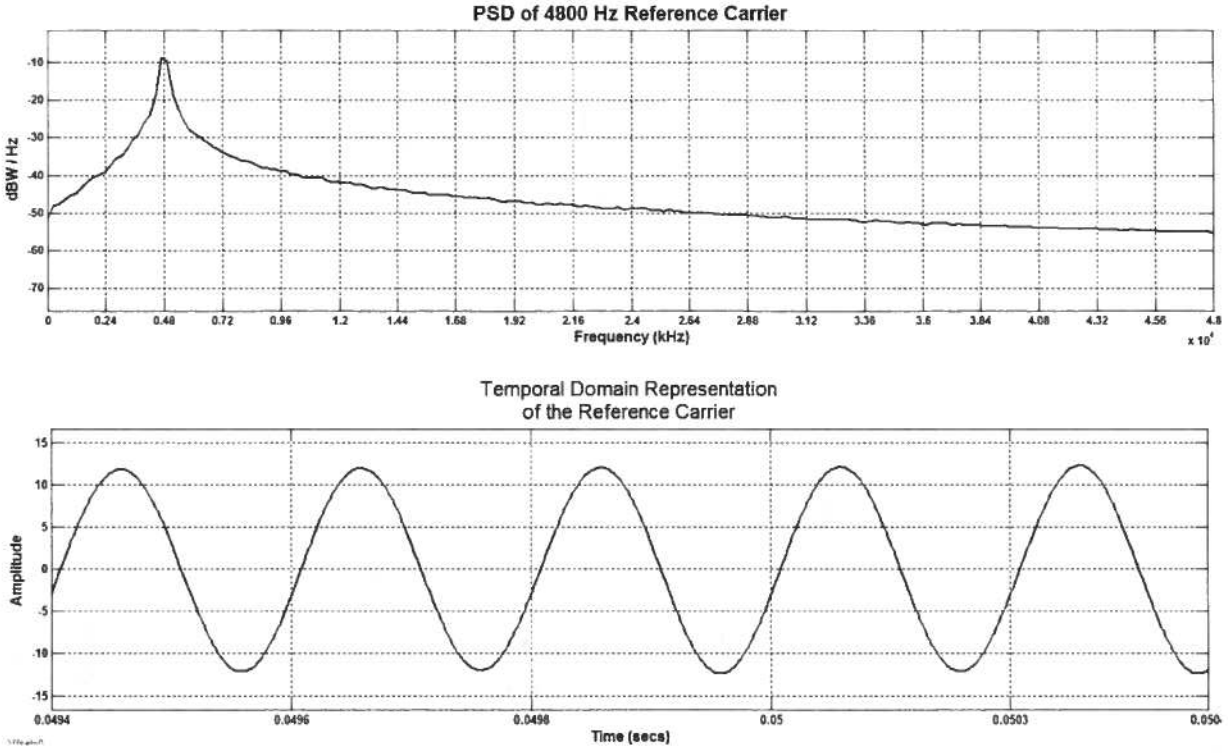


Figure 15. (Top) Power Spectral Density of the 4800 Hz reference carrier and (Bottom) the temporal domain signal of the reference carrier

The reference carrier is shown on the bottom of Figure 15 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. It's operation can be understood by considering the PLL in figure 16. The Phase detector includes the multiplier, phased detector gain K_d and $4f_c$ filter. The reference carrier at $2f_c$ is multiplied by the locally generated sinusoid which after low pass filtering at $4f_c$ 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 θ and φ of the reference and VCO respectively (Nguyen & Shwedyk, 2009).

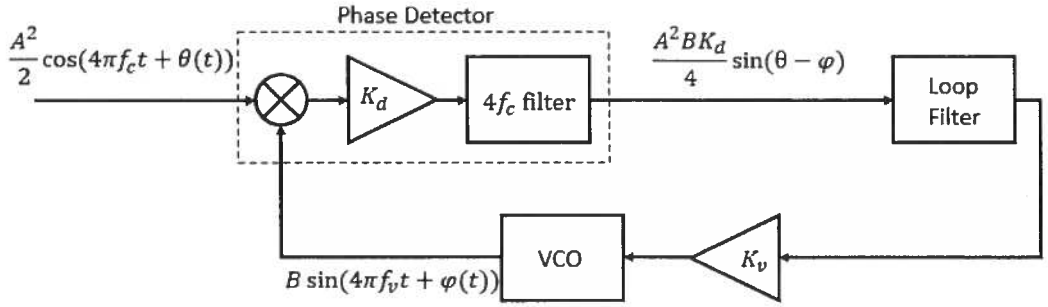


Figure 16. Time domain model of the PLL used for carrier frequency and phase estimation

The multiplication in the phase detector makes the PLL highly non-linear, but if the phase error, $\theta - \varphi$, is small than the output of the phase detector can be approximated as, $\sin(\theta - \varphi) \approx \theta - \varphi$. Under this approximation, the linearized PLL in terms of the phase is shown in Figure 17.

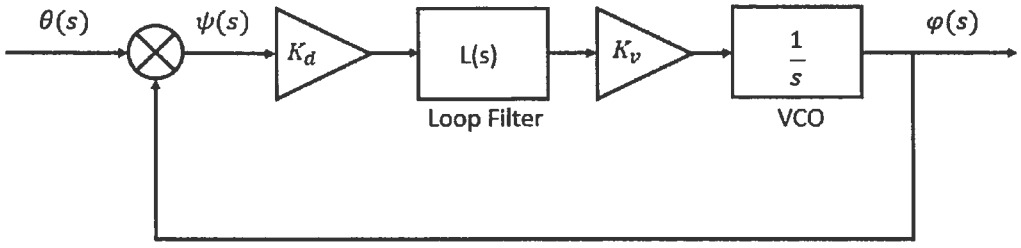


Figure 17. Linearized PLL in the Laplace domain. Note that the $4f_c$ filter is omitted in the linearized Laplace model because it is assumed that the double frequency component was filtered leaving only the phase error

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 17 is given by,

$$H(s) = \frac{\theta(s)}{\varphi(s)} = \frac{K_d K_v L(s)}{s + K_d K_v L(s)}, \tag{4}$$

And the steady error transfer function is,

$$\varphi(s) = \frac{s\theta(s)}{s + K_d K_v L(s)}. \tag{5}$$

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 ($\theta(s) = 1/s$) and ramp ($\varphi(s) = 1/s^2$) inputs. This is accomplished by application of the final value theorem to equation (5). This results in the following:

Step Input: $$\varphi(\infty) = \lim_{s\to 0} \frac{s^2\left(\frac{1}{s}\right)}{s + K_d K_v L(s)} = 0$$

Ramp Input: $$\varphi(\infty) = \lim_{s\to 0} \frac{s^2\left(\frac{1}{s^2}\right)}{s + K_d K_v L(s)} = \frac{1}{K_d K_v L(s)}$$

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,

$$L(s) = K_P + \frac{K_I}{s} \tag{6}$$

Where K_P is the proportional gain and K_I is the integral gain. After substituting the loop filter expression into the closed loop transfer function given by equation (6), the new closed transfer function describing the PLL's behavior is,

$$H(s) = \frac{K_d K_v K_P s + K_d K_v K_I}{s^2 + K_d K_v K_P s + K_d K_v K_I} \quad . \tag{7}$$

Recognizing that equation (7) is a prototypical second order transfer function, the PLL transfer function can be re-written in terms of the natural frequency w_n and the dampening ratio ξ (Crawford, 2008).

$$H(s) = \frac{w_n\left(1 + \frac{2\xi}{w_n}s\right)}{s^2 + 2\xi w_n + w_n^2} \tag{9}$$

Where,

$$w_n = \sqrt{K_d K_v K_I} \tag{9.1}$$

$$\xi = \frac{K_d K_v K_P}{2w_n} \tag{9.2}$$

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 9, 9.1, and 9.2 can be used to approximate the response of the time domain model.

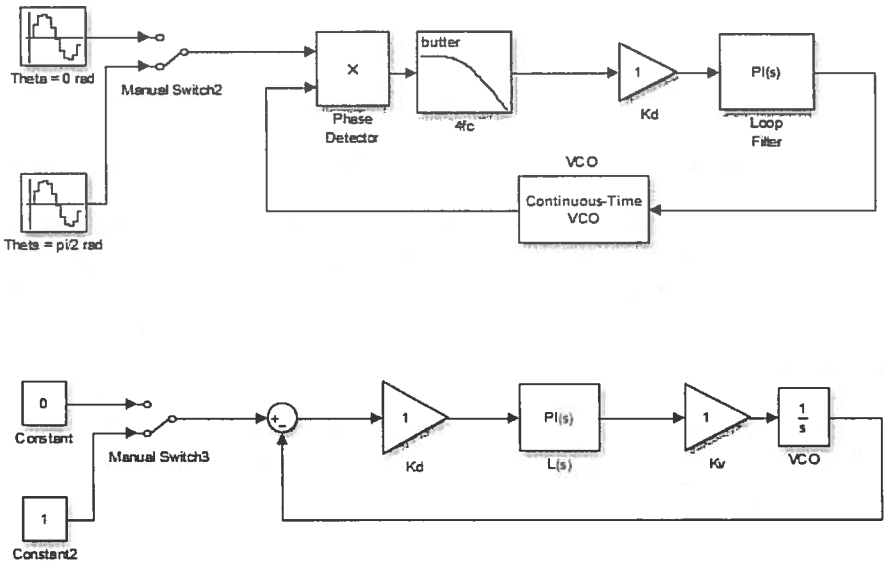


Figure 18. (Top) Time domain model of the PLL used in the carrier recovery circuit and (Bottom) linearized PLL.

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 $T_b = 0.833$ 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 $T_s = 4/\xi w_n$. If the dampening ratio, ζ 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, w_n 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 4.

K_d	K_v	K_P	K_I
1	1	9600	3.6×10^7

Table 4. 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 18 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 $4f_c$ low pass filter in the phase dectector is first order with a cutoff frequency 9600 Hz.

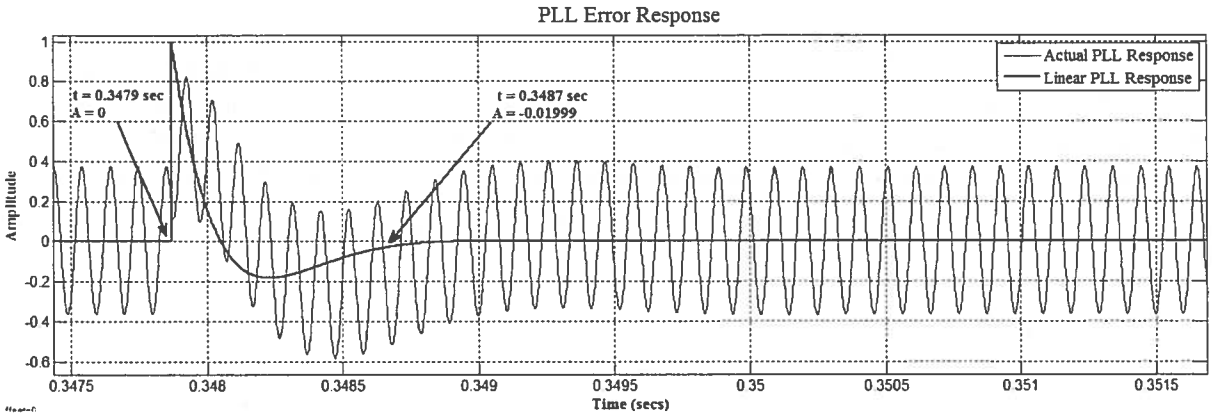


Figure 19. Comparison of the actual PLL error response to the linear PLL response

In figure 19, 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 $t = 0.3479s$ and reached 2 percent of its final value at $t = 0.3487s$ 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 20 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 ms. Although it's unlikely the PLL will be subject to such large frequency steps, it demonstrates the robustness of the design.

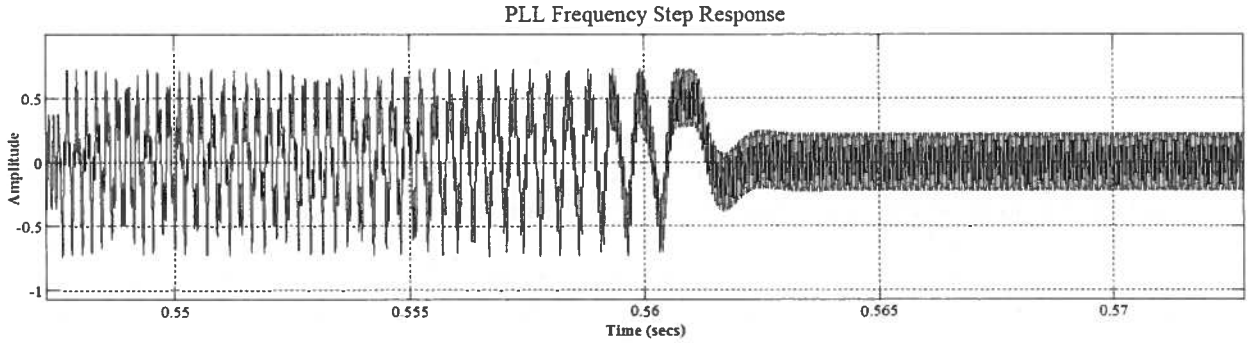


Figure 20. PLL frequency step response to a 4800 Hz step. Time to lock after frequency step is approximately 15.8 ms.

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 bandpass filter with center frequency of 2400 Hz and a Q of 10. Figure 21 compares the 4800 Hz square wave from the PLL to the output of the divide by two flip flop.

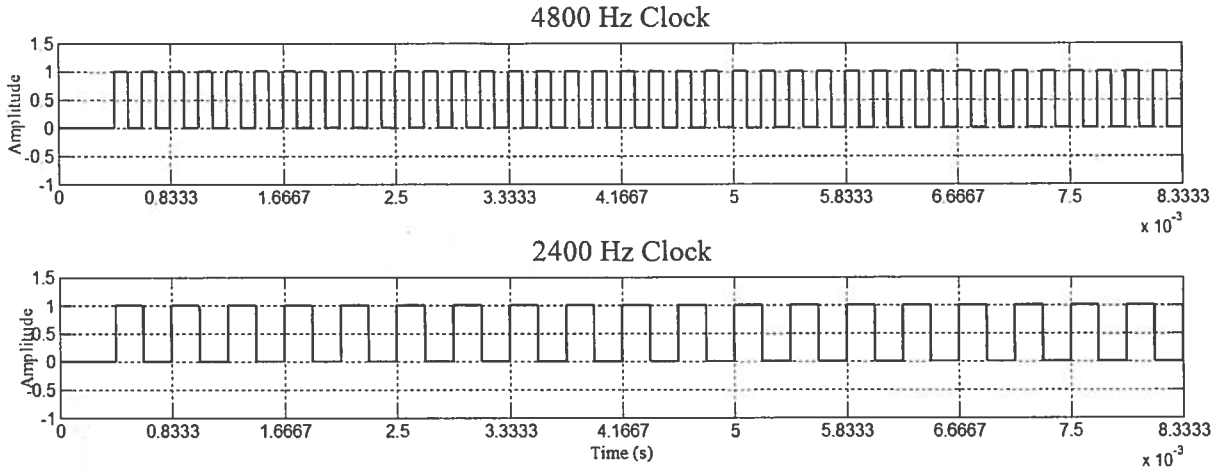


Figure 21. (Top) 4800Hz carrier reference from the PLL after being converted to a square wave. (Bottom). 2400 Hz in phase carrier following the divide by two flip-flop.

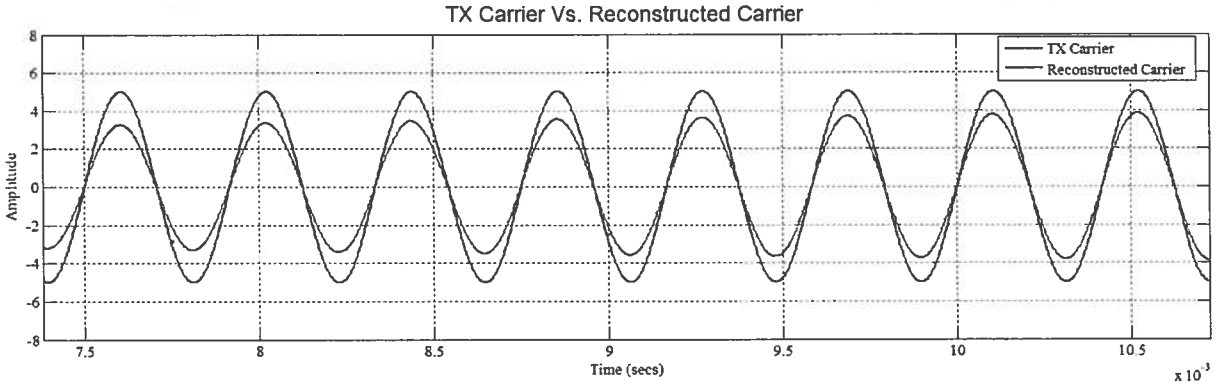


Figure 22. Transmitted carrier superimpose onto the reconstructed carrier demonstrating the PLL's ability to track the phase and frequency of the received signal.

Early-Late Gate Timing Recovery

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 23 illustrates the Manchester data recovered from the mixing of the BPSK signal and the reconstructed carrier.

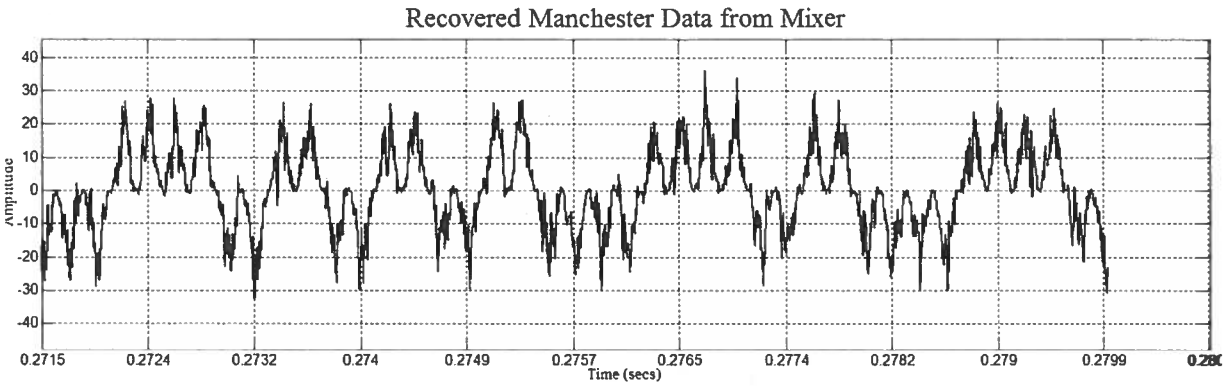


Figure 23. Reovered Manchester data that is forwarded to the early late-gate synchronizer for clock extraction.

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 tacks 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.

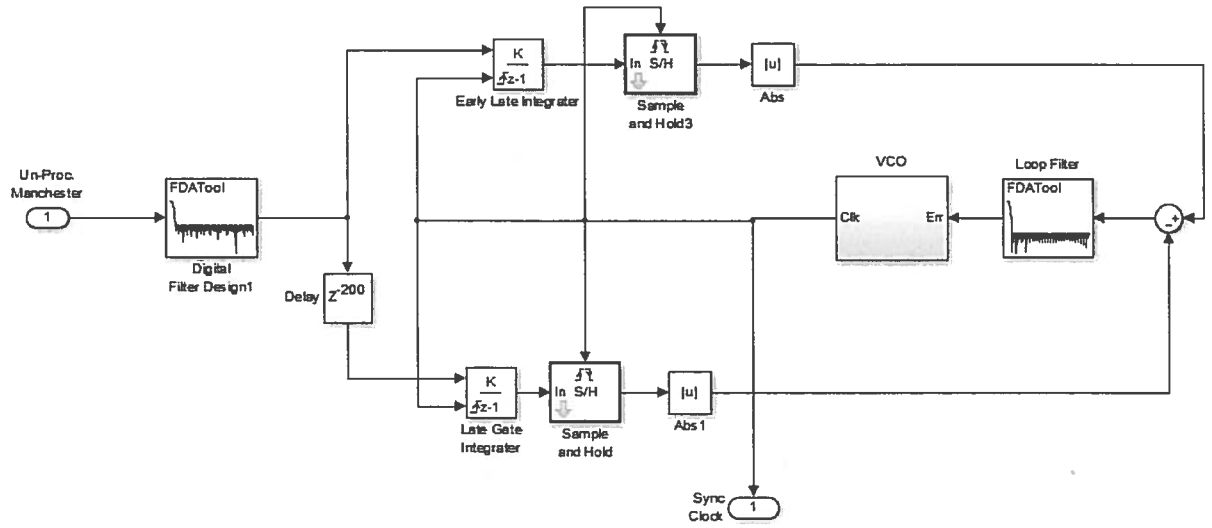


Figure 22. Early-Late Gate synchronizer used for clock extraction

Note that the early late gate in figure 22 requires more extensive simulation and research. There are many sources in literature that discuss the popularity and the conceptual overview of the early-late gate, but few discuss implementation and simulation. As such, confidence in the proper operation of the early-late gate in figure 22 is moderate. Some simulations like those shown on the next page illustrate that the early-late gate is functioning properly, even with AWGN as low as 10dB and phase steps up to 45 degrees. However, a simulation that produces results showing that the clock is out of synch with the received waveform has yet to appear. Because a counter example cannot be found, confidence in its proper operation and implementation is not high. More work is required.

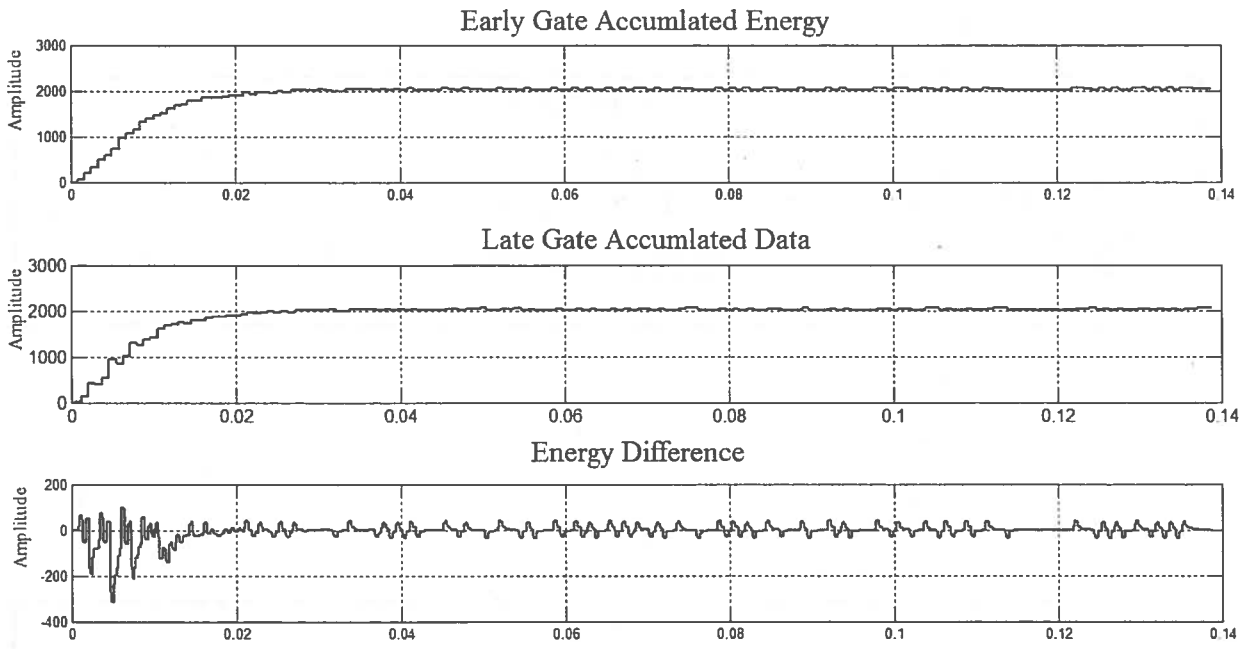


Figure 23. Energy accumulated from the early compared to the energy acculated from the late gate. The difference in accumulated energy betweent he early and late gate is used to drive the VCO towards a synchronous clock

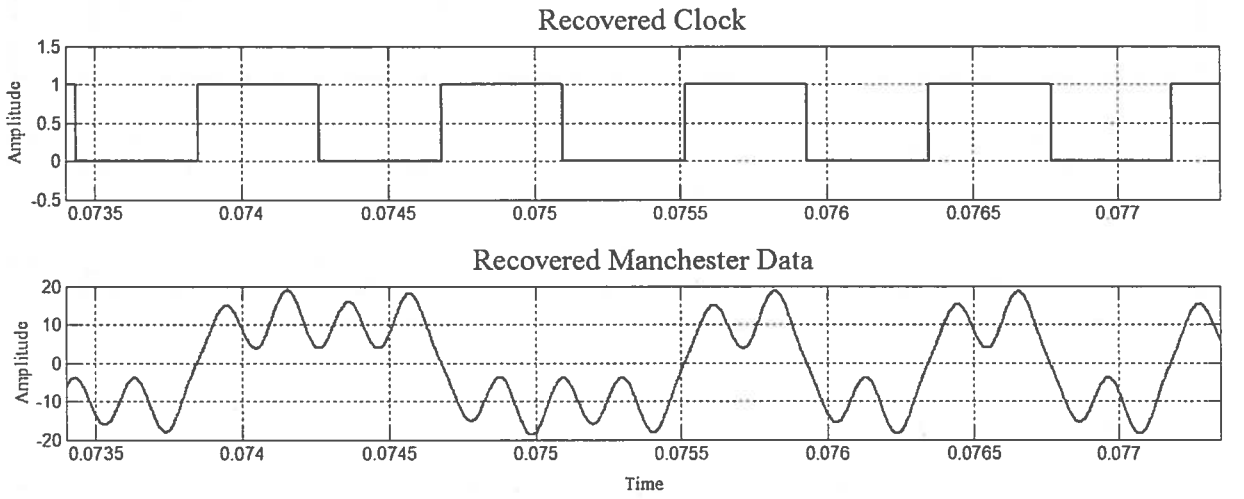


Figure 24. The recovered clock after synchronization is compared to the received Manchester data stream to demonstrate phase coherently

3.1.3. Concatenated FEC codes (Brandon)

The third digital communication system to be analyzed is System C – which comprises everything in System B with the addition of a forward error correction (FEC) engine and a modification to the BPSK modem. Specifically, the modification includes making the BPSK demodulator implement soft-decision decoding instead of hard-decision decoding. This change will prove to be beneficial to the overall SNR of the digital communication system. An explanation of this will appear shortly as we describe the FEC engine in a clockwise fashion (see Figure 3.1.3.1) starting with AX.25 frame generation (top left) and looping back

around to packet error rate calculation (bottom left).

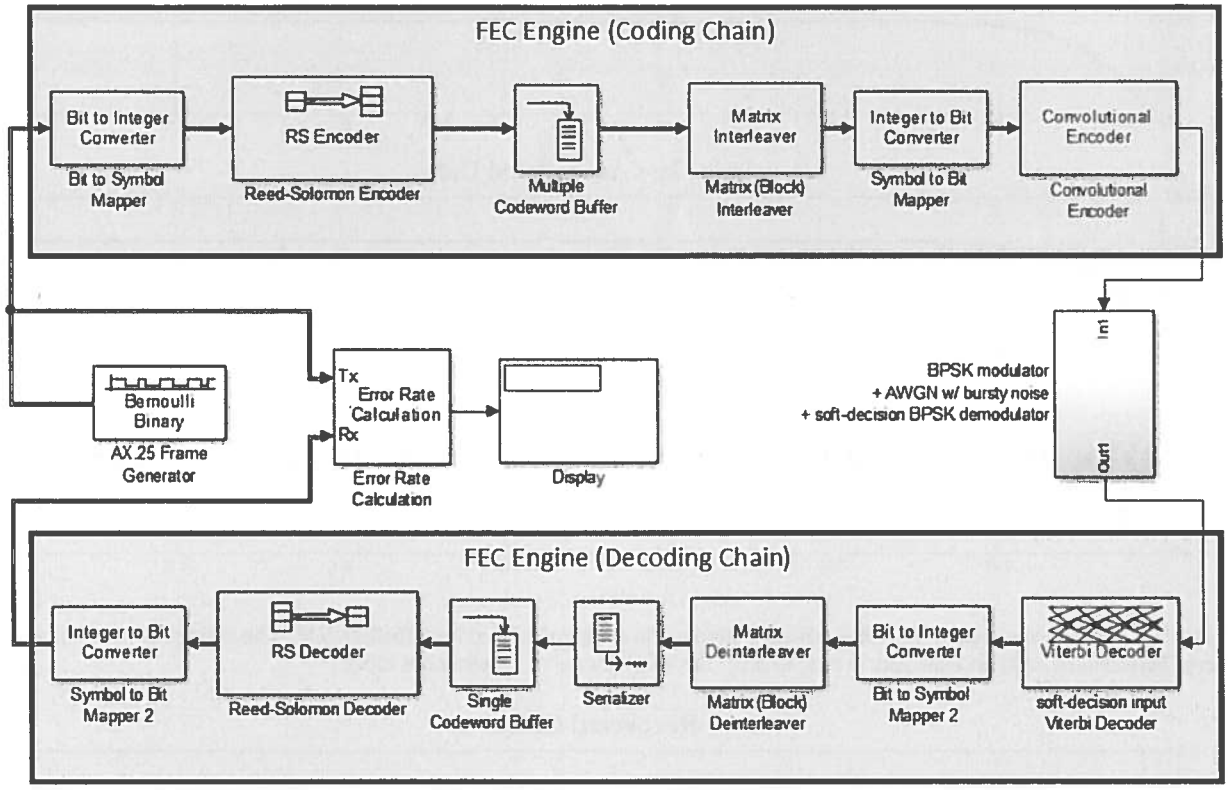


Figure 3.1.3.1. A system-level diagram depicting System C with the FEC engine. The engine is comprised of a coding chain and a decoding chain. Besides from the addition of the FEC engine, System C differs from System B in that the BPSK demodulator implements soft-decision decoding instead of hard-decision decoding.

The FEC engine comprises a concatenation of two forward error correcting codes. As discussed in Section 1, one of the codes will be a block code (correcting burst errors) while the other code will be a convolutional code (correcting random errors). The next text segments will elucidate the reasoning for this concatenation of FEC codes.

Block Encoding

One of the general categories of forward error correcting codes is *block codes*. Reed-Solomon (RS) codes, one form of block codes, perform exceptionally well in correcting burst errors in a received signal. This senior design project elected to incorporate an RS code into the FEC engine for the sole purpose of correcting burst errors that seep into the received bit stream. Figure 3.1.3.2 highlights the section of the FEC engine dedicated to block *encoding*.

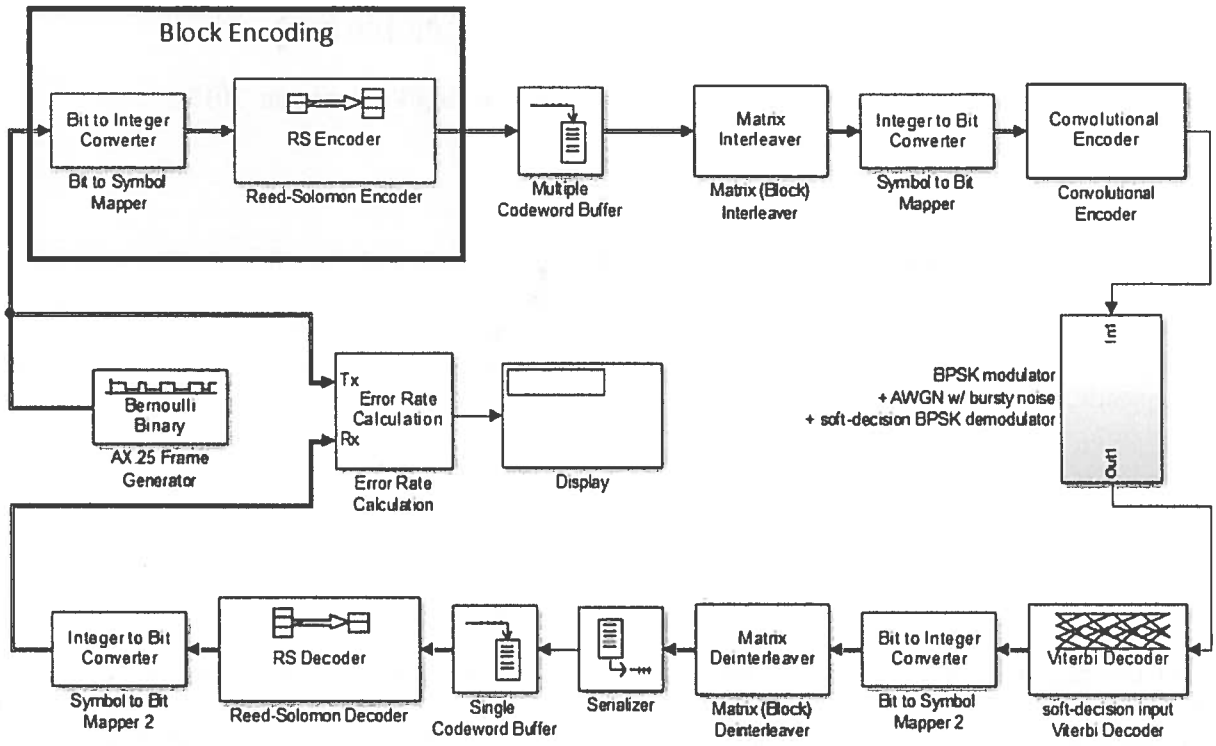


Figure 3.1.3.2. Highlights the block encoding unit of the FEC engine.

At this point, one may ask themselves how does block *encoding* work to correct burst errors. To answer such a question, we must understand that once block *encoding* is done, it must be undone at some point – this is block *decoding*. Block *decoding* will be discussed shortly, but let us first examine how block *encoding* works and how we use Simulink to simulate its functionality. It should be noted that this will be a high-level explanation that circumvents the detailed implementation of block coding. Specifically, a thorough description of Reed-Solomon codes requires an involvement of abstract algebra, specifically Galois fields (Sklar, 2001). In this regard, the curious reader is recommended to visit the excellent mathematical treatment provided in the digital communications textbook entitled *Digital Communications: Fundamentals and Applications (2nd Edition)* by Bernard Sklar. Additionally, an RS code can be realized using a linear feedback shift register (LSFR), but in the interest of time, this senior design project has elected to utilize intellectual property cores in order to bypass this design step.

Let us first begin the explanation by imagining a bit stream. The block encoder deals with correcting *symbol* errors, not single bit errors. For instance, a group of 3 bits could be abstracted to one of eight symbols (0 through 7). This functionality is represented by the *Bit to Symbol Mapper* in Figure 3.1.3.2. The symbol is then operated on by the Reed-Solomon encoding process. This is represented by the *Reed-Solomon Encoder* block in Figure 3.1.3.2. Essentially, the Reed-Solomon encoder attaches a set of parity (or redundancy) symbols to the end of a collection of symbols (known as a message word) (Viswanathan, 2013). A RS code converts *k* symbols (the message word) into a codeword, or *block*, consisting of *k symbol*s. The RS encoder essentially extends the message word with *n-k* parity symbols. This is known as an (*n*, *k*) RS code. The following depicts a (7, 3) RS code which converts a 3-symbol message word into a 7-symbol codeword (*block)*:

$$[5\ 2\ 3] \rightarrow [5\ 2\ 3\ 5\ 4\ 4\ 2]$$

The inputted bit stream representation of this would look like the following:

$$[101\ 010\ 011] \rightarrow [101\ 010\ 011\ 101\ 100\ 100\ 010]$$

The (n, k) RS code has an error-correcting capability (t) expressed as (Viswanathan, 2013):

$$t \leq \frac{n-k}{2}$$

Put differently, the (n, k) RS code can correct up to t symbol errors in a given codeword. For instance, let us polute the previous codeword example with symbol errors. The first and second of the matrices below show one and two symbol errors, respectively, that are correctable by a (7, 3) RS code. However, the third matrix shows three symbol errors which is greater than the error-correcting capability of a (7, 3) RS code. Consequently, the code fails to correct the symbol errors in the third matrix.

[5 **1** 3 5 4 4 2] correctable

[5 **1** 3 **0** 4 4 2] correctable

[5 **1** 3 **0** 4 4 **7**] not correctable

We see that regardless of if one or two symbols were received in error, the (7, 3) RS code could correct the symbol errors. One can imagine a lengthier RS code, such as the (255, 235) RS code, possesses an even more lenient error-correcting capability. In other words, the (255, 235) RS code can correct up to 5 symbol errors within a codeword. If the (255, 235) RS code deals with 8-bit symbols, this means that up to five contiguous symbol errors (up to 40 contiguous bits) are correctable. This elucidates the power of RS codes in correcting for long strings of received symbol errors (burst errors) caused by bursty noise in a propagation medium.

The performance of RS codes are a function of their symbol size (in bits), redundancy, and code rate (Sklar, 2001). One can easily imagine an RS code to be more successful at correcting errors the larger the codeword is, which means that a give burst error is relatively smaller (and hence more correctable). Hence, the larger the symbol size of a RS code, the larger the codeword is, and consequently the better the RS code performs. The code rate of an RS code is the ratio of symbols that comprise a message word and a codeword. Hence, the code rate is expressed as (Viswanathan, 2013):

$$code\ rate = \frac{k}{n}$$

When the code rate is high, the number of symbols that comprise a message word (k) and a codeword (n) are fairly close in value. The number of added redundancy symbols is fairly low. Contrarily, when the code rate is low, the number of added redundancy symbols is fairly high. This high number of redundancy symbols equates to a high computational complexity of the RS code and a higher bandwidth requirement (Sklar, 2001). However, a large number of redundancy symbols results in better error-correcting performance. Consequently, this senior design project will aim to optimize the symbol size, redundancy, and code rate for the purpose of increasing error correction capabilities without putting too much demand on hardware or bandwidth resources.

Block Interleaving

One can imagine that there are instances where a burst error is too extensive for a given RS code to correct. To increase the chances of the RS code receiving a sufficiently short burst error, we can essentially mix up the codewords from the RS encoder and then transmit the mixed information. This way, when an overly

extensive burst error occurs over the propagation medium, the receiver can put the mixed information stream back into un-mixed sequence, which essentially splits the extensive burst error into a disjointed series of smaller, correctable burst errors. This technique is known as *block interleaving*. Figure 3.1.3.3 shows the Simulink blocks of System C responsible for block interleaving.

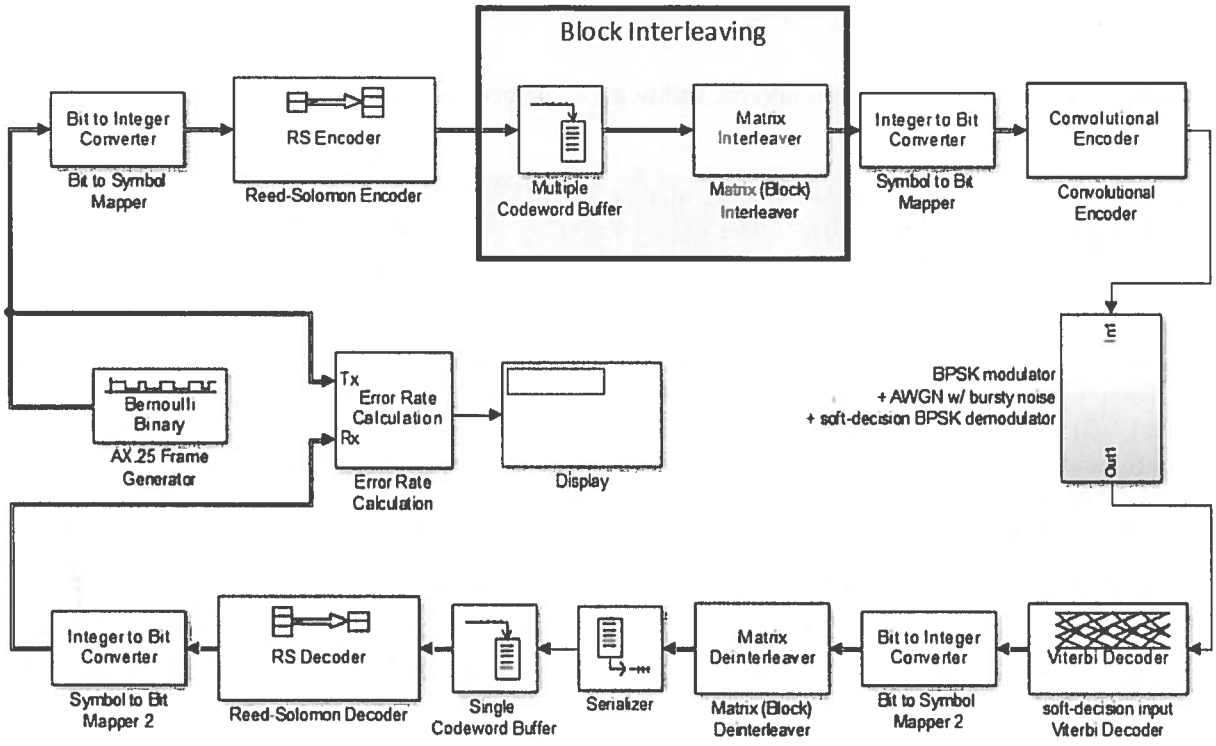


Figure 3.1.3.3 Highlights the block interleaving unit of the FEC engine.

Block interleaving simply consists of a single (d x n) matrix permutation. The matrix consists of d rows (the *interleaver depth*) of the n-symbol wide codewords (blocks) generated by the RS encoding process. Hence, the matrix consists of n columns. The sole purpose of the *Multiple Codeword Buffer* in Figure 3.1.3.3 is to accumulate codewords (blocks) and provide the block interleaver with a matrix of codewords (blocks). Let the following table represent a codeword (block) matrix:

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35

Table 1. Depicting a block matrix with five blocks (codewords). Each block comprises a single row and are colored differently to illustrate this point. There are seven columns to illustrate the point that we are dealing with seven-symbol wide blocks (codewords). The *interleaver depth* of this block interleaver is obviously 5 because there are five rows.

Each block is written into the block matrix *row-by-row* (e.g. from the top to bottom). The magic of block interleaving consists of the fact that the matrix is transmitted by reading the matrix *column-by-column* (e.g.

from the left to right). For instance, the matrix of Table 1 may be filled as follows:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35

The matrix will be *block interleaved* into the following sequence and transmitted further down the FEC encoding chain:

1	8	15	22	29	2	9	16	23	30	3	10	17	24	31	4	11	18	25	32	5	12	19	26	33	6	13	20	27	34	7	14	21	28	35

Let us now demonstrate the power of block interleaving in augmenting the Reed-Solomon error-correcting capabilities. Let us assume that (7, 3) RS code generated the codewords in the matrix of Table 1. We know that the (7, 3) RS code can correct up to two symbol errors in a given codeword. Let us imagine that we elected to not use a block interleaver at all and just transmit the first of first of two matrices shown above. By a stroke of pure bad luck, let us assume that the *entire* third codeword gets corrupted by bursty noise. The result is shown in the following matrix (the red values represent burst error):

1	2	3	4	5	6	7	8	9	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	22	23	24	25	26	27	28	29	30	31	32	33	34	35

We see that there are seven symbol errors within a codeword and the (7, 3) RS code cannot correct for this many symbol errors. Let us block interleave the symbol stream this time around:

1	8	15	22	29	2	9	16	23	30	3	10	17	24	**31**	4	**11**	**18**	**25**	**32**	5	12	19	26	33	6	13	20	27	34	7	14	21	28	35

Let us now de-interleave the symbol stream:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	**18**	19	20	21	22	23	**24**	**25**	26	27	**28**	29	30	**31**	**32**	33	34	35

We can see now that no single codeword consists of more than two symbol errors. Hence, the (7, 3) RS code would succeed in correcting for the entire burst error. One can easily imagine how increasing the interleaver depth can spread burst errors apart even further. In general, if a propagation channel causes b symbol errors in contiguous fashion, then the interleaver depth (d) is calculated as follows:

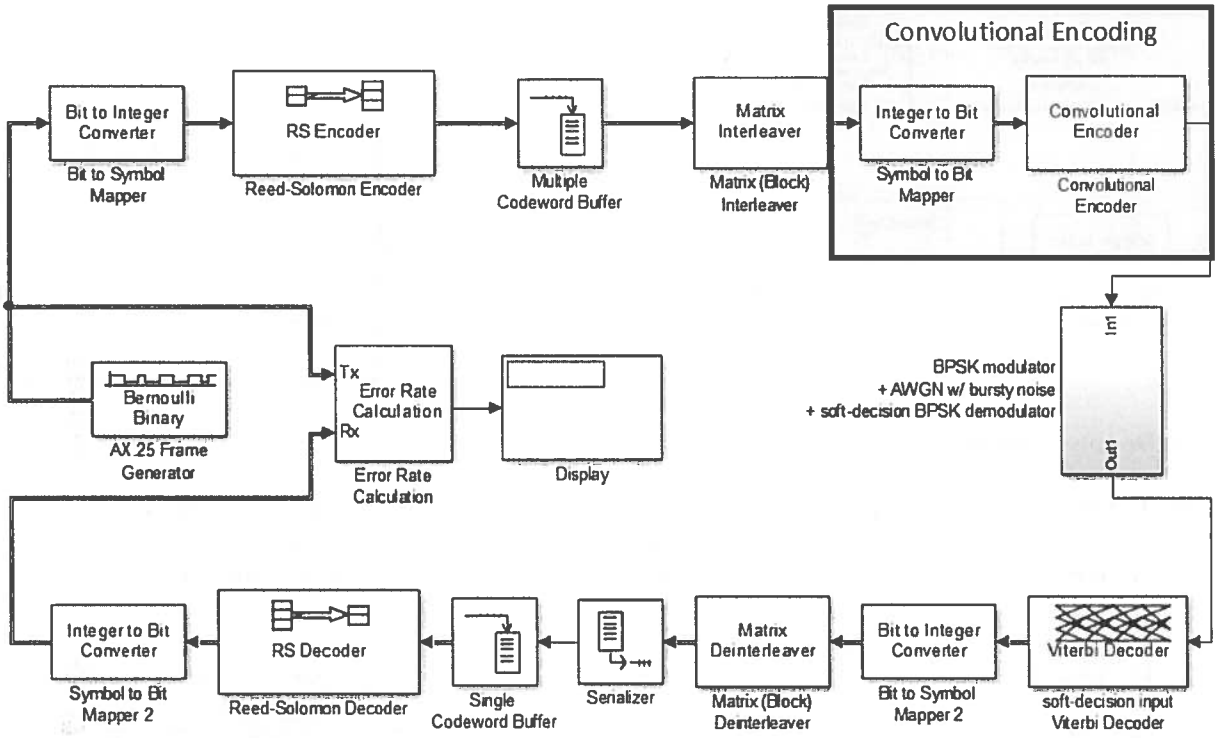
$$d > \frac{b}{t}$$

where t is the maximum number of symbol errors within a single codeword that a given RS code can correct.

However, the disadvantage of increasing the interleaving depth is that besides from using slightly more hardware resources, the time required by *Multiple Codeword Buffer* for filling the block matrix increases as well. The higher the interleaving depth, the higher the delay in the digital communication system. This senior design team must be cognizant of this during the design of System C.

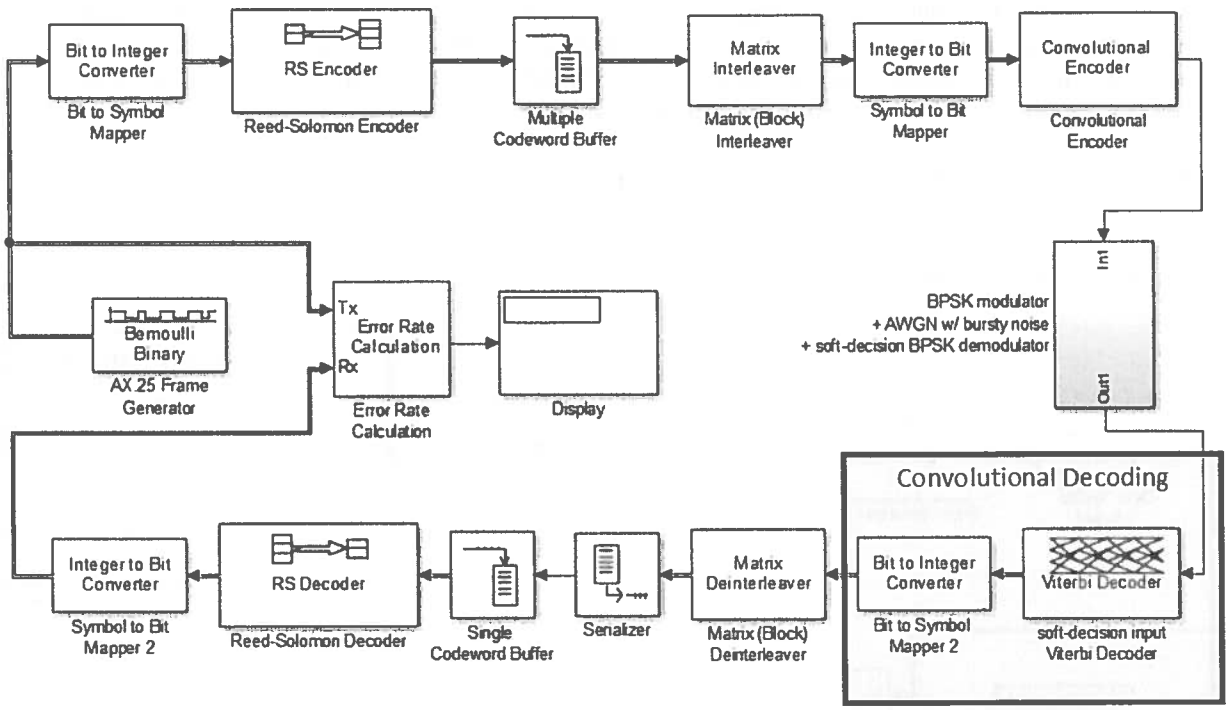
Convolutional Encoding

[TODO (TBC by 12/02/13): A thorough, high-level explanation of the general (n, k, L) convolutional code will be discussed here.]



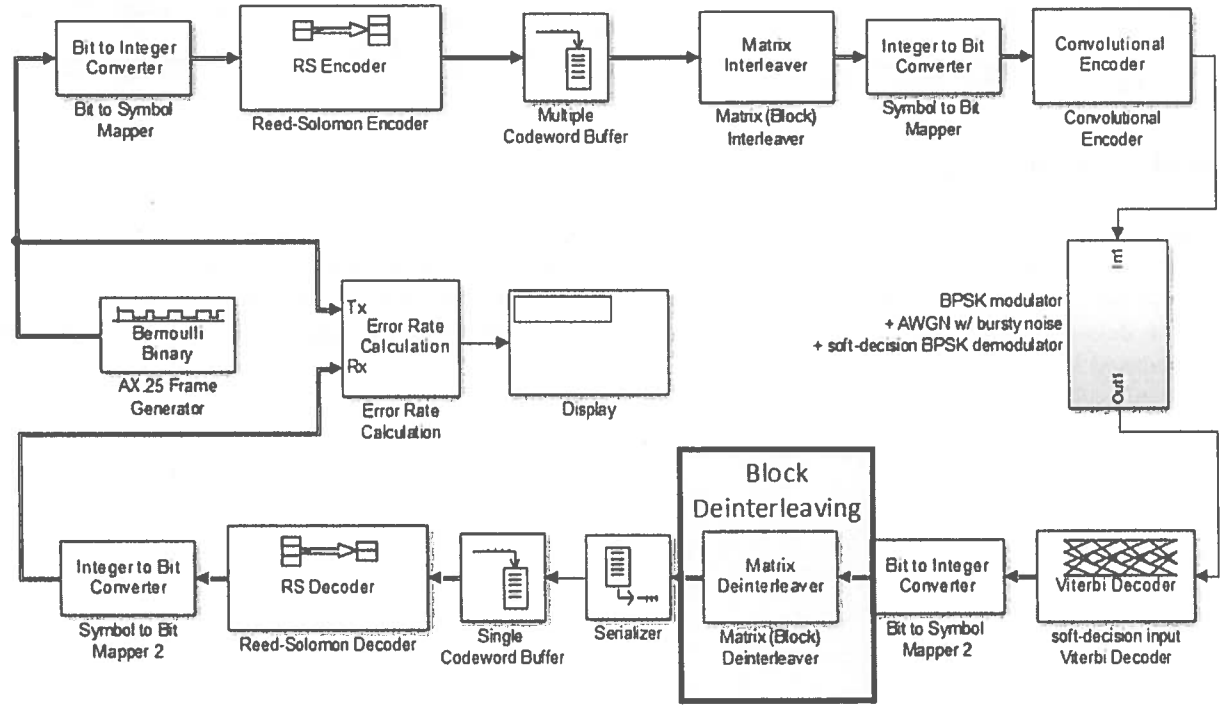
Convolutional Decoding

[TODO (TBC by 12/02/13): Since the discussion of the FEC engine functionality is limited to high-level descriptions only, this section will emphasize the inverse functionality provided in the Convolutional Encoding section. However, it is necessary to provide reasoning for using soft-decision decoding instead of hard-decision decoding in the BPSK demodulator. That reasoning will be discussed in this section.]



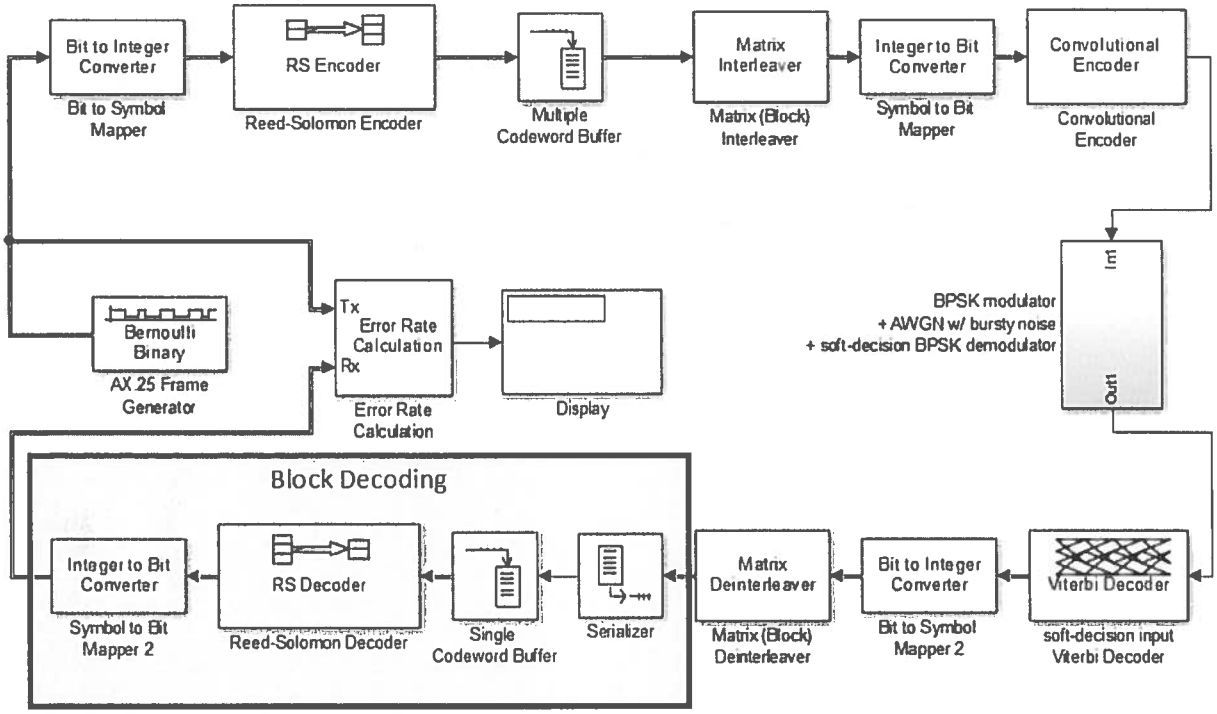
Block De-interleaving

[TODO (TBC by 12/02/13): This section will literally emphasize the fact that the block de-interleaver performs the inverse functionality of the block interleaver. In fact, the examples provided in the Block Interleaving section already hinted as to how block de-interleaving works.]



Block Decoding

|TODO (TBC by 12/02/13): This section will emphasize high-level inverse functionality of the Block Coding section. The Serializer and Single Codeword Buffer will be briefly explained.|



3.2. Hardware Implementation using ISE Design Suite

4. EVALUATION (BRANDON)

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.

5. SUMMARY AND FUTURE WORK

6. ACKNOWLEDGEMENTS

7. 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.*

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

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

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

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.

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,*

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

APPENDIX A: PRODUCT SPECIFICATION

APPENDIX B: SOME INTERESTING RELEVANT DERIVATION