Investigation of a Correlation Based Technique for Rapid Phase Synchronization in the DVB-S Standard

Francis T. Nguyen

Thesis submitted to the Faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering

Robert W. McGwier, Chair
Joseph M. Ernst
Michael Buehrer
Carl Dietrich

December 20, 2015

Keywords: DVB-S, satellite, communications, correlation
Investigation of a Correlation Based Technique for Rapid Phase Synchronization in the
DVB-S Standard

Francis Nguyen

ABSTRACT

The Direct-Video-Broadcast Satellite (DVB-S) standard is used to provide video and radio to millions of users worldwide. It is designed to provide quasi-error free satellite communications. This thesis discusses some of the limitations of the DVB-S standard, describes some attempts in related work to address these concerns, and purposes a new modification to enhance the performance and reliability of the Direct-Video-Broadcast Satellite (DVB-S) standard by using a correlator in a DVB-S receiver. In many existing receive chains, synchronization speed is slightly delayed because phase ambiguity cannot be determined and corrected until after Viterbi decoding. Using correlation against known symbols before demodulation, the phase ambiguity can be corrected prior to Viterbi decoding, thus reducing the amount of time required to synchronize the received signal. To enhance the correlator’s ability to detect the DVB-S synchronization bytes, a two byte, rather than single byte, known sync word is proposed as a modification to the standard. The motivation behind a longer sync word is to improve the standard in high noise environments. A two byte sync word provides more known information for correlation. The resulting correlation peaks are double that of when a single byte is used; this corresponds to about a 3 dB increase in SNR to provide fast signal acquisition and signal tracking in a noisy environment.
Dedication

This thesis is dedicated to my parents who have given everything so I can have this opportunity.
Acknowledgments

I would like to acknowledge the many people who helped guide me and make this thesis possible. I would like to thank my advisors Dr. Robert McGwier, for his vision and guidance through the course of my research, and Dr. Joseph Ernst, for his thorough edits of all my research endeavors and his unwavering support throughout my graduate studies. I would also like to thank my committee members, Dr. Michael Buehrer and Dr. Carl Dietrich, for their continued wisdom and insight.

Also, I would like to thank my colleagues at BIT Systems, for providing me the flexibility, understanding, and motivation to pursue this degree, and the Hume Center, for laying the foundation and providing the inspiration for all my graduate work.

Finally, thank you to my family and friends for continually supporting and encouraging me through this process. Special thanks goes out to Deirdre Kilcoyne, for being my “partner-in-crime” throughout my college education, and Claire Steindam, for putting up with me when I wasn’t easy to put up with.
List of Figures

2.1 Block diagram of the DVB-S transmitter path. .................................. 4
2.2 DVB-S frame structure after energy randomization [1]. ...................... 5
2.3 DVB-S frame structure after Reed-Solomon encoding and convolutional interleaving [1]. ................................................................. 5
2.4 Block diagram of the DVB-S receiver path [1]. .................................. 7

3.1 The autocorrelation output of two byte sync words: 0x47E2, 0x4747, 0x47B8, and 0x471D, in which the second maxima, i.e. second highest peak, of 0x47E2 is the lowest among the other potential two byte sync words making it the most suitable two byte sync word. ............................................................. 13
3.2 Interleaved frame structure with extended sync word. ....................... 14
3.3 Block diagram of the proposed receiver/correlation path. .................. 16
4.1 The cross correlation of 5 DVB-S superframes with a one byte sync word and random payload (Eb/No=15dB). ...................................................... 19

4.2 The QPSK modulated, complex symbols of 0x47$^{HEX}$ and 0xB8$^{HEX}$. ........ 20

4.3 The cross correlation of 5 DVB-S superframes with a two byte sync word and random payload (Eb/No=15dB). ...................................................... 21

4.4 The cross-correlation outputs at varying phase rotations. ......................... 22

4.5 The Type I and Type II error rates with respect to correlation thresholds at varying Eb/No for a one byte sync word. ................................................. 23

4.6 The Type I and Type II error rates with respect to correlation thresholds at varying Eb/No for a two byte sync word. ................................................. 24

4.7 The Receiver Operating Characteristic plot of a one byte sync word with varying average SNR. ................................................................. 26

4.8 The Receiver Operating Characteristic plot of a two byte sync word with varying average SNR. ................................................................. 27

4.9 Type I and Type II error rates for a one byte sync word at 0 dB and for a two byte sync word at 3 dB. ......................................................... 28

5.1 The GNU Radio flowgraph for a DVB-S transmitter. .......................... 30

5.2 The GNU Radio flowgraph for a DVB-S receiver. .............................. 31
5.3 The GNU Radio flowgraph for a DVB-S correlation detector.

5.4 The GNU Radio flowgraph for a DVB-S receiver including the correlation detector. It is a combination of the flowgraphs in Fig. 5.2 and Fig. 5.3.

5.5 The GNU Radio custom block, rotationSelector.py, to output the appropriate phase adjusted signal based on the correlation detector's output.
List of Tables

2.1 The system performance of a DVB-S transponder over 33 MHz [1]. . . . . . 4

4.1 The AUC from ROC plots of a one byte sync word and a two byte sync word. 28

5.1 The input and output descriptions for the rotationSelector block. . . . . . . 35
Chapter 1

Introduction

The purpose of this thesis is to investigate a method for enhancing the robustness of the Direct-Video-Broadcast Satellite (DVB-S) standard. DVB-S is used to transmit multimedia programming to viewers around the world via satellites. The current implementations of DVB-S receivers cannot resolve the phase ambiguity before demodulation, because the phase cannot be detected by the Viterbi decoder [1]. The symbols must be inner decoded (Viterbi decoder) to locate the sync word. An iterative process of performing the Viterbi decoding, looking for the sync word, adjusting the QPSK demodulator, and repeating if the sync word is not found, which can be tedious and resource intensive. This thesis investigates a new receiver implementation, which uses a correlator in conjunction with a two byte sync word, instead of the one byte sync word specified by the standard, in the data stream to increase the robustness of synchronization. The two byte sync word can be especially useful in high noise environments when it is difficult to achieve synchronization on the received signal. The
purpose of using correlation is to provide fast signal acquisition and continuous monitoring of channel effects, such as phase ambiguity, timing offset, and frequency shift.

This thesis begins with background on the DVB-S standard and related works. In Chapter 3, the methodology behind the algorithm and simulation is discussed. Simulation results are presented in Chapter 4 to detail the effectiveness of a two byte sync word over the original standard. Chapter 5 discusses a correlator implementation in GNU Radio. Chapter 6 concludes the thesis with a summary of the work and with descriptions of possible future work in the area.
Chapter 2

Background

This chapter details the DVB-S standard with emphasis on its frame structure. Then, the chapter takes a look at the 2nd generation of satellite broadcasting, DVB-S2, and compares it to DVB-S. Finally, related works to use of a correlator in a DVB-S receiver are examined.

2.1 DVB-S Standard

DVB-S is used worldwide to provide video and radio to millions of users worldwide over satellite. DVB-S is a mainstay in satellite broadcasting because it is designed to provide quasi-error free transmission; i.e. it provides a Bit Error Rate (BER) of between $10^{-10}$ and $10^{-11}$ at the receiver’s output if the received signal is above the carrier to noise ratio (C/N) and carrier to interference ratio (C/I) for the chosen forward error correction (FEC) [1][2]. An example of DVB-S performance over a 33 MHz transponder is provided by the European
Table 2.1: The system performance of a DVB-S transponder over 33 MHz [1].

<table>
<thead>
<tr>
<th>Bit Rate Ru (after MUX) [Mbit/s]</th>
<th>Bit Rate R’u (after RS) [Mbit/s]</th>
<th>Symbol Rate [Mbaud]</th>
<th>Convolut. Inner Code Rate</th>
<th>RS Outer Code Rate</th>
<th>C/N (33 MHz) [dB]</th>
</tr>
</thead>
<tbody>
<tr>
<td>23.754</td>
<td>25.776</td>
<td>25.776</td>
<td>1/2</td>
<td>188/204</td>
<td>4.1</td>
</tr>
<tr>
<td>31.672</td>
<td>34.368</td>
<td>25.776</td>
<td>2/3</td>
<td>188/204</td>
<td>5.8</td>
</tr>
<tr>
<td>35.631</td>
<td>38.664</td>
<td>25.776</td>
<td>3/4</td>
<td>188/204</td>
<td>6.8</td>
</tr>
<tr>
<td>39.590</td>
<td>42.960</td>
<td>25.776</td>
<td>5/6</td>
<td>188/204</td>
<td>7.8</td>
</tr>
<tr>
<td>41.570</td>
<td>45.108</td>
<td>25.776</td>
<td>7/8</td>
<td>188/204</td>
<td>8.4</td>
</tr>
</tbody>
</table>

Figure 2.1: Block diagram of the DVB-S transmitter path.

Telecommunications Standards Institute (ETSI) standard guide and is seen in Table 2.1.

## 2.1.1 Transmit Path

Fig. 2.1 shows the block diagram of a DVB-S transmitter. The concatenation of three forward error correction blocks: Reed-Solomon encoder, convolutional interleaver, and punctured convolutional encoder, provide for a high degree of error correction [1].

At the input of a DVB-S transmitter is an MPEG-2 Transport Stream (TS). Typically in the DVB-S standard, the MPEG-2 TS carries multiplexed video and audio content that is to be broadcast, but it can be used to transmit non-multimedia data packets [2]. The TS packet is 188 bytes long with the first byte being 0x47_{HEX}. 0x47_{HEX} is chosen by the MPEG-2 TS standard to minimize sync byte emulation; in that the sync word does not appear frequently in the frame outside of the sync word [4][5]. Additionally, any rotation of
the bits along the byte string will never be the same as the original value, thus providing a favorable autocorrelation result. Although the sync byte is defined within the MPEG-2 standard, the same sync word is used within DVB-S to resolve phase ambiguity and align the frames for de-interleaving.

Most MPEG-2 packets do not implement a cyclic redundancy check (CRC) to verify the payload information. Certain packets that provide the Service Information table, which provides programming information to the end user, are protected by a 32-bit CRC. However, every MPEG-2 packet has a one byte Transport Error Indicator within the header to indicate if the Reed Solomon decoder could resolve all errors within a packet. If an unrecoverable bit error is present in the packet, then the packet would be discarded.

The first stage in the transmitter is randomization in which energy spikes are dispersed. To
denote the start of de-randomization at the receiver the first sync byte of 8 MPEG-2 TS packets is inverted to 0xB8_{HEX}; the frame structure after randomization is depicted in Fig. 2.2 [1].

The randomized stream is outer encoded with a shortened Reed-Solomon encoder (Block Length, N=204; Message Length, K=188; Check Symbols, T=8) and convolutional interleaver. Each 188 byte long frame is encoded so that each frame is a 204-byte long error protected packet [3]. Each error protected packet is further encoded by a convolutional interleaver with a depth of 12; each sync byte is routed into the first interleaver branch to preserve its periodicity. The interleaver is used to distribute bursty errors among packets [6]. Fig. 2.3 shows the frame structure after interleaving [1].

The interleaved frames are inner encoded by a punctured convolutional encoder. The encoder is based on a mother rate of 1/2 and constraint length of K=7. The standard allows for varying puncture rates of 1/2, 2/3, 3/4, 5/6, and 7/8 [1][3]. Only rate 1/2 is used for the purpose of this thesis to demonstrate the effectiveness of this correlation technique; the system can be easily adapted to support other puncture rates.

Finally, square root raised cosine pulse shaping, with a roll-off factor of 0.35, is applied to the data stream and then QPSK modulated with Gray Coding for transmission [1][7].

The bit rate of an MPEG-2 stream broadcasted via satellite is based on its satellite bandwidth and selected error correction rate. For a 1/2 encoded signal on a 32 MHz satellite bandwidth at 25 MBaud, the resulting bit rate would be 23.04 Mbits per second. This corresponds to
less than one uncorrected bit error per hour [1].

2.1.2 Receive Path

The opposite steps from a DVB-S transmit chain, as described in Section 2.1.1, are applied to a typical DVB-S receiver, as seen in Fig. 2.4. First, the received satellite signal is applied to a matched filter, complementary to the pulse shaping applied at the transmitter, and then QPSK demodulated. Next, the inner decoder (i.e. Viterbi decoder) is applied to the data stream along with a de-puncturing algorithm for the signal’s given code rate.

The next step in typical DVB-S receiver designs is to use a sync decoder and look for the MPEG-2 sync byte, 0x47 \textsubscript{HEX}. The sync decoder provides alignment information for the next stage in the process, de-interleaving. It is at this step where the sync decoder attempts to resolve the phase ambiguity of the QPSK demodulator. A loopback is utilized to adjust the demodulator to successfully locate the MPEG-2 sync byte. In addition, the sync decoder uses the sync byte to align the packet for the convolutional de-interleaver.

The output of the sync decoder is applied to the convolutional de-interleaver, the reverse of the convolutional interleaver in the transmitter, to minimize any burst errors before going into the outer decoder [6]. At the outer decoder, the Reed Solomon protected frames, as
depicted in Fig. 2.3, are error corrected and each frame is now 188 bytes long. The frame structure at the outer decoder’s output is illustrated in Fig. 2.2

Before final output to a television or physical interface, the energy dispersion algorithm is removed from the byte stream, or de-randomized. The inverted sync bytes, $0xB8_{HEX}$, are used in this step to align the byte stream for the start of the removal algorithm. The inverted sync bytes are then returned to the non-inverted state, and finally an MPEG-2 TS is generated. Often times, multiple programs are multiplexed within a single stream; additional processing at the physical interface (e.g. television) de-multiplexes the stream to produces video and audio programming.

### 2.2 DVB-S and DVB-S2

The DVB-S standard was adopted for use in 1994 to deliver MPEG-2 services to satellite customers. However, the standard has limitations in that it was framed around the MPEG TS protocol; not satellite transmission [2]. It didn’t account for varying transmission parameters and channel effects on the signal. DVB-S2 was adopted in 2005 to adapt to the changing multimedia market and correct some of DVB-S’s shortcomings. DVB-S2 was geared to provide High-Definition Television as well as data services to satellite customers. DVB-S2 also introduced higher order modulation schemes and powerful Low Density Parity Check based error correction. However, most importantly, DVB-S2 implemented Variable Coding and Modulation (VCM) and Adaptive Coding and Modulation (ACM) which allows for varying
transmission protocols to maximize the transponder’s bandwidth. VCM and ACM account for fading and channel degradation which the original DVB-S standard did not [8]. The adaptive nature of DVB-S2 allows it to perform better than DVB-S, especially in non-ideal weather environments, such as rain [9].

Although DVB-S2 has performance gains over DVB-S, DVB-S is still used around the world, because it is simpler and more cost effective to maintain. This is critical in developing markets where upgrading to DVB-S2 would require swapping every customers’ set-top box to receive DVB-S2 broadcasts. As a result, the work here is still relevant to easily and cost effectively enhance the DVB-S standard.

2.3 Related Works

In traditional DVB-S receivers, a sync search scheme proposed by Zhiqiang and Yun is typically used [1][10]. Zhiqianq and Yun’s implementation leverages a finite state machine to track the location of the sync byte across the received bit stream. A confidence value is used to ensure that the detected sync words have the correct periodicity before completing synchronization [10]. The downside to their implementation is that the received signal must be demodulated and Viterbi decoded to perform bit matching. As a result, any phase ambiguity in the signal cannot be resolved until after Viterbi decoding of the signal.

Xue, Wang, and Li proposed a correlation detecting method, but for the second generation of the DVB-S standard, DVB-S2. DVB-S2 differs from DVB-S in that it is designed for higher
rate codecs. As a result, advanced forward error correcting codes and enhanced modulation schemes are used in the DVB-S2 standard. In addition, the DVB-S2 frame structure differs greatly from DVB-S; most notably in that DVB-S2 uses a Physical Layer (PL) Header frame [8][11]. However, the algorithms for frame correlation and synchronization proposed by Xue, et al. can be adapted into the DVB-S standard.
Chapter 3

Methodology

This chapter discusses the mathematical concepts behind a correlator which includes the correlation algorithm, threshold selection, and the method to resolve the phase ambiguity at the demodulator. Section 3.2 discusses the selection of a two byte sync word. Finally, in Section 3.3 the methodology behind the simulations is detailed.

3.1 Algorithm

A correlation based detector is used since it is the Maximum Likelihood detector when assuming Additive White Gaussian Noise (AWGN). Essentially, a matched filter is used to determine the likelihood at which the received data matches the known sync words. A sliding window technique is used to slide the known sync word with the received signal [12].
A threshold is needed to determine the start of the DVB-S frame from the correlation output. A threshold value is selected to avoid false detection, i.e. when there seems to be a sync word, but is not really there (Type I error), and missed detection, i.e. when there is a sync, but is not detected (Type II error). A threshold corresponding to the uniformly most powerful test for each false alarm rate is chosen to maximize the detection rate [13]. The probability of Type I error is only dependent on the distribution associated with the null hypothesis and the threshold and is therefore independent of signal strength. Type II corresponds to the alternative hypotheses at varying signal strengths for each false alarm rate.

A cost matrix analysis is performed to determine which type of error is better or worse. In this system, a missed detection is worse than a false detection. A false detection simply means that time is spent to process it; however, a missed detection results in a missed opportunity to process the signal with no means to recover it.

To resolve phase ambiguity during initial signal acquisition, each of the four possible QPSK constellations are correlated against the received symbol stream. The correlation output with the highest maximum value of the four QPSK constellation is where the QPSK constellation will be locked. The correlation continuously runs to track channel effects on the received signal.
Figure 3.1: The autocorrelation output of two byte sync words: 0x47E2, 0x4747, 0x47B8, and 0x471D, in which the second maxima, i.e. second highest peak, of 0x47E2 is the lowest among the other potential two byte sync words making it the most suitable two byte sync word.

### 3.2 Extended Sync Word

The first step in the development of an extended sync word is to choose the bytes. Each of the possible two byte sync words, with the first byte starting with 0x47\textit{HEX}, are autocorrelated, i.e. correlated against themselves [14]. The original sync word remains the first byte of the new two byte sync word in order to maintain compatibility with the rest of the DVB-S receiver and to adhere with the MPEG-II standard. The resulting autocorrelations are plotted in Fig. 3.1. It can be seen that each of the two byte sync words have the same
maximum, but $0x47E2_{HEX}$ has the lowest second maxima of the autocorrelated sync words, making it the optimal two byte sync word. While all 256 options were evaluated, only four options are shown in Fig. 3.1.

To insert it into the DVB-S frame, the adjacent byte after the original sync word after convolutional interleaving is replaced with an $0xE2_{HEX}$. Thus, the first sync word of eight DVB-S frames is $0xB8E2_{HEX}$ and the remaining seven in the superframe becomes $0x47E2_{HEX}$. The new frame structure is illustrated in Fig. 3.2. An interesting observation of the new sync word is that $0xE2_{HEX}, 1110\ 0010_{BIN}$, is simply the bit mirror of $0x47_{HEX}, 0100\ 0111_{BIN}$, thus giving the new sync word, $0100\ 0111\ 1110\ 0010_{BIN}$, a circular rotation. Circular rotation occurs when the last bit is shifted to the front of the bit string eight times, which produces the exact same byte string before any shifting. Additionally, the two byte sync word will reduce sync byte emulation because the probability of having two bytes in the data stream match the new sync word is reduced by $1/2^8$ when compared with a one byte sync word.

The reason why the new sync word is inserted into the data stream after interleaving is because the data packets are convolutionally interleaved to distribute any bit errors across different packets rather than corrupting a single packet. Within the interleaver, only the first
byte of each packet is preserved in the first interleaver branch as the sync word; the resulting bytes are interleaved with each other [6]. As a result, the second byte of the extended sync word is not inserted before the convolutional interleaver; rather, after it. A real system would have to know where to inject the second byte of the new sync word within the data stream before the interleaver.

Due to the computational difficulties attributed to the convolutional nature of the interleaver, there would need to be changes in the standard’s interleaver implementation to insert the sync byte before interleaving.

### 3.3 Simulation

In the simulation, Matlab is used to perform the cross-correlation between the transmitted data and the known sync bytes. First, a data stream must be generated and a transmit chain must be implemented. All these tasks are performed in Matlab.

#### 3.3.1 Data Stream

Since the frame structure after interleaving is known, a sample DVB-S data stream is generated based on Fig. 2.3; the payload is loaded with random 0’s and 1’s. The data stream is then transmitted by the QPSK modulator. For the two byte sync word, the new sync word replaces the original sync byte as well as the next byte, as illustrated in Fig. 3.2.
3.3.2 Transmit Chain

Since the frame structure is known after the Convolutional Interleaver, a partial DVB-S transmitter chain is implemented starting with the punctured convolutional encoder with rate $1/2$. The encoded packets are normalized to 1’s and -1’s before pulse shaping and modulated according to the DVB-S standard. In addition, Additive White Gaussian Noise (AWGN) is added to the transmitted symbols to simulate base channel distribution [12]. Similarly, the known sync byte stream, a one byte sync word and two byte sync word, must be similarly encoded and transmitted to generate the known symbols.

3.3.3 Cross-Correlation

Fig. 3.3 details the block diagram of the proposed receiver/correlation implementation. At the receiver, the received symbols are matched filtered to remove the pulse shaping. Then, the symbols are phase adjusted for each of the possible QPSK constellations. With the symbols for the received data and the symbols for the sync bytes at the appropriate
rotation, the cross-correlation between the two is performed using the ‘xcorr’ function in Matlab which adheres to the algorithm described in Section 3.1. The correlation output with the maximum peak value is then selected. The results from the cross-correlation are found in Section 4.

In this thesis, all four rotations are illustrated to demonstrate the correlator’s operation. However, system complexity and processing could be reduced by implementing only two correlations, one at 0° and another at 90°. A minimum and maximum comparison could then be applied to check and correct for 180° rotations of the correlation output.
Chapter 4

Results

This chapter details the results from MATLAB simulation. It includes the correlation output, resolving phase ambiguity, optimum threshold level, and the receiver operating characteristics (ROC) for a one byte versus two byte sync word.

4.1 Correlation Output

Fig. 4.1 shows the correlator output when the data stream is loaded with random data and correlated against a single 0x47\textsubscript{HEX} sync byte. Because the input data is random, there is a correlation peak wherever the data byte matches the sync byte. This is a problem because a byte in the data stream could be mistaken for a sync byte. Additional processing must be done to determine whether a peak in the correlator actually corresponds to a sync byte or is just random data. The upside to this correlation technique is that lag time for correlation
Figure 4.1: The cross correlation of 5 DVB-S superframes with a one byte sync word and random payload (Eb/No=15dB).

is minimized because only one byte of data is being correlated.

A spike at the first of every eight packets is missing because that packet contains the inverted sync word, 0xB8_{\text{HEX}}. Conventional wisdom would say that there should be a negative maximum in that location. However, that is not the case, because, while 0xB8_{\text{HEX}} is the bit inversion of 0x47_{\text{HEX}}, QPSK modulation does not make them symbol inversions of each other, as seen in Fig. 4.2. If the complex symbols of the inverted sync word were truly the invert of the normal sync word symbols, then the correlation would contain a negative maximum.
Fig. 4.2: The QPSK modulated, complex symbols of $0x47_{HEX}$ and $0xB8_{HEX}$.

Fig. 4.3 shows that the the correlation maximums are double with a two byte sync word, thus pushing the peaks higher above the noise floor. In addition, it is easier to distinguish the start of a frame because the chances of having a repeated $0x47_{HEX}$ in the data stream are reduced, but locating the inverted sync byte requires post-processing.
4.2 Resolving Phase Ambiguity

In the simulation, the transmitted symbols are phase rotated before adding channel noise to simulate phase ambiguity at the receiver. In this simulation’s specific case, a 180° phase adjust is applied at the transmitter to simulate phase ambiguity at the receiver for a DVB-S transmission utilizing a two byte sync word. Fig. 4.4 shows the correlation output at different phase adjustments: 0°, 90°, 180°, and 270°. An algorithm calculates which of the four rotations provides the maximum correlation value. It can be seen that Fig. 4.4c has the
maximum correlation peaks and, visually, that the frames are easily discernible within the plot. As a result, the incoming signal must be adjusted by $180^\circ$ to acquire phase lock. Using correlation to resolve phase ambiguity after the matched filter and before demodulating eliminates the need for the sync decoder, typically implemented after the Viterbi decoder, to resolve phase ambiguity. This can improve acquisition time and can provide signal tracking.
Figure 4.5: The Type I and Type II error rates with respect to correlation thresholds at varying Eb/No for a one byte sync word.

4.3 Optimal Threshold

Picking an optimum threshold is key to a correlation detector’s effectiveness. A proper threshold value must accurately locate the sync word from the correlator’s output by minimizing the rate of false detection, Type I error, and missed detection, Type II error. Figs. 4.5 and 4.6 plot Type I and Type II errors together at varying power levels for one byte sync
Figure 4.6: The Type I and Type II error rates with respect to correlation thresholds at varying Eb/No for a two byte sync word.

word and two byte sync words, respectively. The intersection of Type I and Type II errors produce an ideal threshold.

It should be noted that the Type I errors in Fig. 4.5 are the same because the Type I errors are only dependent on the null hypothesis; Type I error is independent of the strength of the signal. A similar observation is seen for the two byte sync word in Fig. 4.6; however, the Type I error is twice as wide in Fig. 4.6 than Fig. 4.5 because twice the number of bits
are correlated.

Type II errors in Figs. 4.5 and 4.6 move towards higher thresholds with increasing signal strength because the error is dependent on the average SNR of the signal.

For a one byte sync word, the threshold value is around 10 depending on the signal strength, but there is still a slight probability of error, especially at lower average SNRs. With a two byte sync word, there is a range of threshold values in which the error rates are effectively 0 and the range of thresholds is a little more than double than the threshold for a one byte sync word; the performance is characterized more closely with the Receiver Operating Characteristics (ROC) curves. Because the correlation maximums are much higher from the noise floor, the frames are easily discernible while minimizing Type I and Type II errors.

4.4 Receiver Operating Characteristics

In this section, the ROC curves of the correlation detector highlights the improvement of using a two byte sync word over a one byte sync word. The ROC curves in Figs. 4.7 and 4.8 plot the false positive rate against the true positive rate for thresholds between a value of -50 and 50 at varying average SNRs [15][16]. Comparing Figs. 4.7 and 4.8, there is about a 6 dB performance increase when using a two byte sync word which is much more evident at lower average SNRs. At about an Eb/No of 3 dB, the performance of a two byte sync byte becomes indistinguishable on this plot to a point where false positives are minimized and the probability of acquiring a true positive is almost guaranteed. For a one byte sync
Figure 4.7: The Receiver Operating Characteristic plot of a one byte sync word with varying average SNR. That point does not occur until about 9 dB and higher. This occurs at higher signal strengths because the Type II error is minimized with increasing average SNR.

The performance gain is numerically evaluated by calculating the Area Under the Curve (AUC) for a two byte sync word and a one byte sync word [17]. Using a trapezoidal integration technique, the AUC for both a two byte sync word at -3 dB and a one byte sync word at 3 dB are about equivalent. The equivalent AUC affirms what is observed through visual inspection of the ROC plots; there is a 6 dB of improvement from a one byte sync word to a two byte sync word. Table 4.1 details all the AUC values for all the ROC curves seen in Fig. 4.7 and Fig. 4.8.
Figure 4.8: The Receiver Operating Characteristic plot of a two byte sync word with varying average SNR.

Initial assumptions says that there should only be a 3 dB, rather than 6 dB, performance gain because the known information to the correlator is doubled. As a result, the correlation peaks as seen in Figs. 4.1 and 4.1 are doubled, or increased by 3 dB, from a one byte sync word to two bytes. From the plots in Fig. 4.9, it can be seen that the error rates for a one byte sync word and a two byte sync word when differed by 3 dB are vastly different, thus indicating that the initial assumption of 3 dB is incorrect. While a 3dB improvement to the height of the correlation peak is expected, there is also a reduction due to the additional averaging of the data over the longer correlation. The “noise” in this case is caused by correlation with unknown data and not is not due to AWGN from the channel.
Figure 4.9: Type I and Type II error rates for a one byte sync word at 0 dB and for a two byte sync word at 3 dB.

Table 4.1: The AUC from ROC plots of a one byte sync word and a two byte sync word.

<table>
<thead>
<tr>
<th>Eb/No</th>
<th>Type 1 Byte Sync</th>
<th>Type 2 Byte Sync</th>
</tr>
</thead>
<tbody>
<tr>
<td>-3 dB</td>
<td>0.9877</td>
<td>0.9990</td>
</tr>
<tr>
<td>0 dB</td>
<td>0.9965</td>
<td>1.000</td>
</tr>
<tr>
<td>3 dB</td>
<td>0.9991</td>
<td>1.000</td>
</tr>
<tr>
<td>6 dB</td>
<td>0.9998</td>
<td>1.000</td>
</tr>
<tr>
<td>9 dB</td>
<td>0.9999</td>
<td>1.000</td>
</tr>
<tr>
<td>12 dB</td>
<td>0.9999</td>
<td>1.000</td>
</tr>
</tbody>
</table>

It should be noted that replacing a payload byte with a two byte sync word can negatively affect the system’s output bit rate. To accommodate the second byte of the sync word, the symbol rate would have to increase which increases the noise power. To accommodate additional noise power, the bandwidth of the satellite would have to increase.
Chapter 5

Implementation

This chapter details the implementation of the proposed correlation technique in a signal processing application.

GNU Radio, an open-source signal processing suite, is leveraged in this implementation because it provides a powerful signal processing framework and an extensive library of signal processing blocks which is leveraged in this application. Because GNU Radio is an open-source application, many blocks have been already developed; many of these developed blocks are leveraged in this implementation.
5.1 DVB-S Transmitter

GNU Radio DVB-S transmitter blocks have been previously developed by Ron Economos to simulate a DVB-S broadcast [18]. The transmitter inputs an actual MPEG-2 TS, containing video and audio, and outputs complex symbols according to the DVB-S standard, which includes randomization, forward error correction, and modulation. The flowgraph is depicted in Fig. 5.1. The transmitter flowgraph is used to generate the symbols that are used for the DVB-S receiver and correlation implementation, as well as the known sync word for correlation.
5.2 DVB-S Receiver

An early GNU Radio DVB-S receiver had been designed by Edmund Tse [19]. Tse’s initial implementation leveraged the sync decoder state machine proposed in [10][20]. However, Tse’s work was written for GNU Radio Version 3.3 and required significant modifications to adapt the code to today’s GNU Radio Version 3.7. In addition, the repeating the Viterbi Decoder to search for the MPEG-2 sync word was resource intensive. This GNU Radio implementation seeks to eliminate the phase adjust callback by using correlation to resolve the demodulator’s phase ambiguity; however, the sync decoder is still used to align packet for de-interleaving using the Viterbi decoded sync word.

The receiver flowgraph without any correlation elements is shown in Fig. 5.2.

The receiver correlation is discussed in Section 5.2.1. The selection of the correlator’s output is discussed in Section 5.2.2.
5.2.1 Correlation

The Correlation Estimator block from GNU Radio is used to calculate the correlation between the received, filtered symbols and the modulated sync word. The modulated sync word is derived by inputting $0x47_{HEX}$ into the transmitter flowgraph, then set the outputted symbols as a parameter for the Correlation Estimator block. The correlation of each phase adjustment is calculated and provided to the threshold checker block. The input to the Correlation Estimator block is also provided as a tagged output stream. The tagged output stream provides timing and phase estimates which can be useful in future modifications.

The GNU Radio flowgraph of just the correlation detector is seen in Fig. 5.3 and the entire
Figure 5.4: The GNU Radio flowgraph for a DVB-S receiver including the correlation detector. It is a combination of the flowgraphs in Fig. 5.2 and Fig. 5.3.

The flowgraph combining the receiver and correlation detector is seen in Fig. 5.4

5.2.2 Threshold Selector

The GNU Radio Threshold block outputs 0 or 1 if the input signal meets the defined threshold. In the Threshold block, when the input signal approaches the High parameter, a 1 is output, and when the input signal drops below the Low parameter, a 0 is output.
The custom-built, python GNU Radio block, rotationSelector.py takes the outputs from the threshold blocks and each of rotated symbol streams and outputs the appropriate stream whose correlation output meets the threshold. For example, if the $0^\circ$ adjusted stream’s correlation meets the threshold, that stream is outputted to the rest of the DVB-S receiver. If the inputs from the thresholds change, then the output is switched accordingly, which allows for quick adaptation to the channel environment. Table 5.1 details the inputs and output of the the block illustrated in Fig. 5.5.

### 5.3 Discussion

The input stream for the correlation detector flowgraph is generated by the transmitter flowgraph, Fig. 5.1, and phase ambiguity is introduced using a Constant Multiply block
Table 5.1: The input and output descriptions for the rotationSelector block.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>in0</td>
<td>$0^\circ$ Rotated Symbol Stream</td>
</tr>
<tr>
<td>in1</td>
<td>$90^\circ$ Rotated Symbol Stream</td>
</tr>
<tr>
<td>in2</td>
<td>$180^\circ$ Rotated Symbol Stream</td>
</tr>
<tr>
<td>in3</td>
<td>$270^\circ$ Rotated Symbol Stream</td>
</tr>
<tr>
<td>thres0</td>
<td>$0^\circ$ Rotated Correlation Threshold Output</td>
</tr>
<tr>
<td>thres1</td>
<td>$90^\circ$ Rotated Correlation Threshold Output</td>
</tr>
<tr>
<td>thres2</td>
<td>$180^\circ$ Rotated Correlation Threshold Output</td>
</tr>
<tr>
<td>thres3</td>
<td>$270^\circ$ Rotated Correlation Threshold Output</td>
</tr>
<tr>
<td>out</td>
<td>Selected Symbol Stream Output</td>
</tr>
</tbody>
</table>

Before outputting to file. When executing the correlation detector flowgraph, Fig. 5.3, individually, the correlation detector is able to output the correct, phase adjusted signal based on the correlation’s output. Next, the receive flowgraph, Fig. 5.2 (without the AGC block and FIR Filter block, since those processes occur before correlation), takes the correlation detector’s output file stream and processes it through the demodulation, error correction, and de-randomization blocks. At the end of this process, an MPEG-2 TS is produced and can be played back on a computer using ffmpeg or another media player. The video output here has the same content as the input at the beginning of the transmit/receive process.

Due to computing complexities, the full correlation-receive flowgraph, Fig. 5.4, is not able to produce a full MPEG-2 TS output. The de-interleaver is able to briefly sync onto the sync word, but loops between losing the sync word and reacquiring the sync word. It is important that the de-interleaver maintains its lock on the sync word, because it is used to align the DVB-S packets for Reed-Solomon decoding. This shortcoming can be attributed to limited system resources in performing the correlations and running the rest of the DVB-S decoding.
processes. Having a system with greater resources as well as using better threading processes can improve the effectiveness of a combined correlation-receiver design.
Chapter 6

Conclusion

This paper has investigated the use of a correlator and possible additional sync byte to improve the performance of the DVB-S standard in low SNR environments. This is just one possible correlation technique that can be used to enhance the robustness of the DVB-S standard. From the two ROC plots in Figs. 4.7 and 4.8, it can be assessed that the use of a two byte sync word provides about a 6 dB improvement over the original standard’s one byte sync word. This contributes to better correlation performance in high noise environments which in turn will allow for quicker symbol acquisition and improved tracking. In addition, multiple correlators can be used to provide phase acquisition and tracking for the DVB-S signal.
6.1 Future Work

While this thesis has demonstrated the advantage of the suggested changes to the standard, future work can be done to implement a synchronizer to work in conjunction with the proposed correlator. This investigation assumed zero frequency offset and perfectly timed clocks. A full synchronizer can provide timing recovery and even frequency shift using the correlation output. The pair can provide a more robust means to track the received signal in the event of channel degradation. Additional work can also be done to examine the correlation detector’s effectiveness against more complicated channels with multi-path loss and shadowing. Furthermore, future work can be done to investigate the impacts of a two byte sync word; the additional synchronization SNR will lead to better ground coverage and resilience to shadowing. Such investigations could include if there is better coverage if the symbol rate or bandwidth was increased to accomodate a two byte sync word or how the bit rate is affected.

The correlation investigated in this thesis was performed in the time domain. A frequency domain correlation, utilizing Fourier Transform, should be investigated because it can provide timing, frequency offset, and frame detect at much lower computational complexity from order $N^2$ to order $N \log(n)$.

Since the implementation is performed completely in GNU Radio, the work in this thesis can be adapted to receive a live, over-the-air DVB-S broadcast, such as the DVB-S broadcast from the International Space Station. The extension would require hardware components,
such as a satellite dish, a low-noise block downconverter, and a Software Defined Radio (e.g. Ettus USRP) [21]. Proper pointing and frequency tuning would be required, but much of that information can be found online by enthusiasts. However, processing limitations on the host system must be considered, especially for larger bandwidth satellite feeds (i.e. signals).

The GNU Radio implementation can also be enhanced. A polyphase resampling can account for timing and carrier offsets. A Costas Loop can be implemented for second order loop fixes. A frequency domain correlation should also be implemented

The correlation known sync word could also be further examined. One possible idea is to aggregate multiple one byte sync words that are separated by a known period and use it during correlation. That technique would provide more information to the correlator and produce higher correlation maximums.

Another extension of the proposed correlator would be to use it to determine the signal’s puncture rate if it is unknown to the receiver. In this paper, only a puncture rate of 1/2 was explored. Using a similar technique to determine phase ambiguity, the puncture rate could be derived by using multiple correlators and determining the output’s maximum value using a correlation detector.
Bibliography


