CHANNEL AND NOISE VARIANCE ESTIMATION IN A SPREAD SPECTRUM COMMUNICATION SYSTEM

by

Tarek M. Haddadin

A thesis submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Electrical and Computer Engineering

The University of Utah

December 2016

The University of Utah Graduate School

STATEMENT OF DISSERTATION APPROVAL

The dissertation of **Tarek M. Haddadin**

has been approved by the following supervisory committee members:

Behrouz Farhang-Boroujeny ,	Chair(s)	**18 Aug 2016** Date Approved
Rong Rong Chen ,	Member	**18 Aug 2016** Date Approved
Neal Patwari ,	Member	**18 Aug 2016** Date Approved

by **Gianluca Lazzi** , Chair/Dean of

the Department/College/School of **Electrical and Computer Engineering**

and by **David B. Kieda** , Dean of The Graduate School.

ABSTRACT

Filter Bank Multicarrier (FB-MC) is a technique similar to Orthogonal Frequency Division Multiplexing (OFDM), used to divide the spectrum of a transceiver into multiple subcarriers or channels. When a single symbol is repeated across all subcarriers, its energy is spread across the entire spectrum. This is referred to as Filter Bank Multicarrier Spread Spectrum (FB-MC-SS). The design of a preamble or training sequence used in the packet construction of a FB-MC-SS transceiver system is explored in this thesis. The preamble is used to acquire an estimate of the channel impulse response and noise variance for each subcarrier. This information is then used to undo the effect of the channel and perform Maximum Ratio Combining (MRC) across all subcarriers. An alternating $\{+1, -1\}$ sequence has been previously proposed for its implementation simplicity. An alternating $\{+1, -1\}$ sequence leads to detection advantages as a result of the impulse response of the matched filter. An alternating $\{+1, -1\}$ sequence also presents many disadvantages. Mainly, the sequence is susceptible to interference because of its distinct frequency. The alternating $\{+1, -1\}$ sequence also has a higher probability of detection by unauthorized users. To combat these deficiencies of the alternating $\{+1, -1\}$ sequence, pseudorandom sequences are explored in this thesis. The goal of the pseudorandom sequence is to gain robustness without forfeiting the packet's detectability by intended receivers. Pseudorandom Polyphase and Maximum Length Binary sequences are explored as randomized preambles. Both the alternating $\{+1, -1\}$ sequence and the pseudorandom sequence are implemented separately in the FB-MC-SS transceiver on a Xilinx FPGA to compare resource utilizations. Pseudorandom Polyphase preamble sequences lead to robust channel frequency response and noise variance estimation in interfered environments. Although alternating $\{+1, -1\}$ sequence leads to straightforward packet detection and simple FPGA implementation, the susceptibility of an alternating $\{+1, -1\}$ preamble to interference makes a pseudorandom preamble sequence more desirable.

CONTENTS

LIST OF FIGURES

Figures

LIST OF TABLES

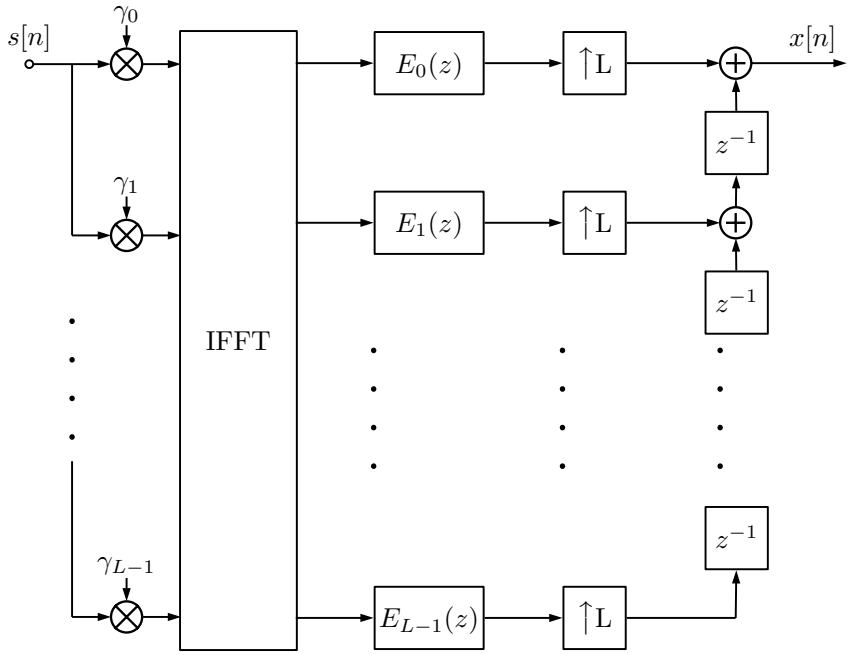
Tables

CHAPTER 1

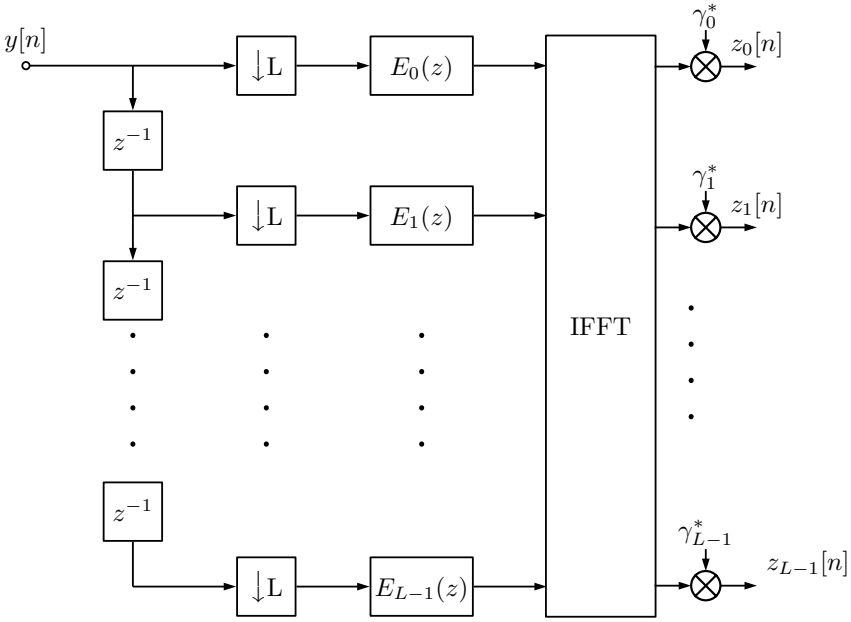
INTRODUCTION

The process of intentionally spreading the energy of transmission in a communication link is commonly referred to as spread spectrum communication. Spread spectrum techniques such as Direct Sequence Spread Spectrum (DS-SS), Frequency Hopping Spread Spectrum (FH-SS), and Multicarrier Spread Spectrum (MC-SS) are widely used [8]. Applications of spread spectrum communication include antijamming systems, low probability of detection systems, code division multiplexing, and cognitive radio control channels. MC-SS has been shown to be more robust to narrow band and partial band interference [5], [1]. Orthogonal Frequency Division Multiplexing (OFDM) is a developed multicarrier modulation technique with an efficient implementation [2]. Filter Bank Multicarrier (FB-MC) improves on OFDM by using a bank of filters to further band limit the subcarriers. When data are multiplexed across the subcarriers, it is known as frequency division multiplexing (FDM). FDM allows for a higher data rate while maintaining a fixed symbol rate. A single symbol can also be repeated across all the subcarriers to spread the energy of the symbol and employ the multicarrier system as a spread spectrum system. When used in spread spectrum communication, OFDM and FB-MC are referred to as OFDM-SS and FB-MC-SS, respectively.

A FB-MC-SS structure was implemented in [13] and proposed as an underlay control channel in cognitive radios. The implemented filter bank structures are presented in Figure 1.1. The transmit filter bank is referred to as the Synthesis Filter Bank (SFB). The receive filter bank is referred to as the Analysis Filter Bank (AFB). The work performed throughout this thesis expands on the FB-MC-SS structure presented in [13] when the data being transmitted are packetized, specifically focusing on achieving reliable estimates of the channel frequency response and noise variance for each subcarrier. These estimates are acquired using the packet's preamble sequence, which is known to the receiver as a priori



(a) Synthesis filterbank block diagram



(b) Analysis filterbank block diagram

Figure 1.1: Filterbank block diagrams

knowledge.

Although FB-MC is not quite as developed as OFDM, there has been some research in the area to develop an efficient filter bank structure using multirate signal processing techniques. The basic idea behind FB-MC is to band limit each subcarrier using a frequency shifted version of a low pass prototype filter $h[n]$. The prototype filter is frequency shifted to each respective subcarrier frequency to create a band pass filter located at the subcarrier's center frequency. An example of a single subcarrier data path can be seen in Figure 1.2. The data path shown in Figure 1.2 can be repeated N times to create a bank of N filters where f_k is the center frequency of the k^{th} filter. Although this process does perform the necessary operations for FB-MC, it has an unnecessarily large complexity due to the repetition of the prototype filter for each subcarrier.

A FB-MC-SS system is characterized by a pulse-shaping filter that is defined as

$$g[n] = \sum_{k=0}^{N-1} h_k[n] \tag{1.1}$$

where N is the number of subcarriers, γ_k is the spreading gain for the k^{th} subcarrier (in the form of $\gamma_k = e^{j\theta_k}$), f_k are the normalized subcarrier frequencies and

$$h_k[n] = \gamma_k h[n] e^{j2\pi f_k n}\,. \tag{1.2}$$

The square-root raised-cosine pulse shape with a roll-off factor of $\alpha = 1$ is proposed in [13] for the prototype filter $h[n]$. An example of the magnitude response of $g[n]$ can be seen in Figure 1.3 when $N = 8$.

Efficient implementation of the filter bank is obtained by using polyphase components of the prototype filter $h[n]$ and taking advantage of some multirate signal processing prop-

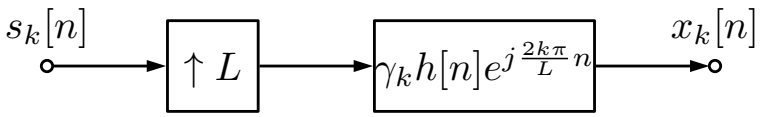


Figure 1.2: Single subcarrier data path

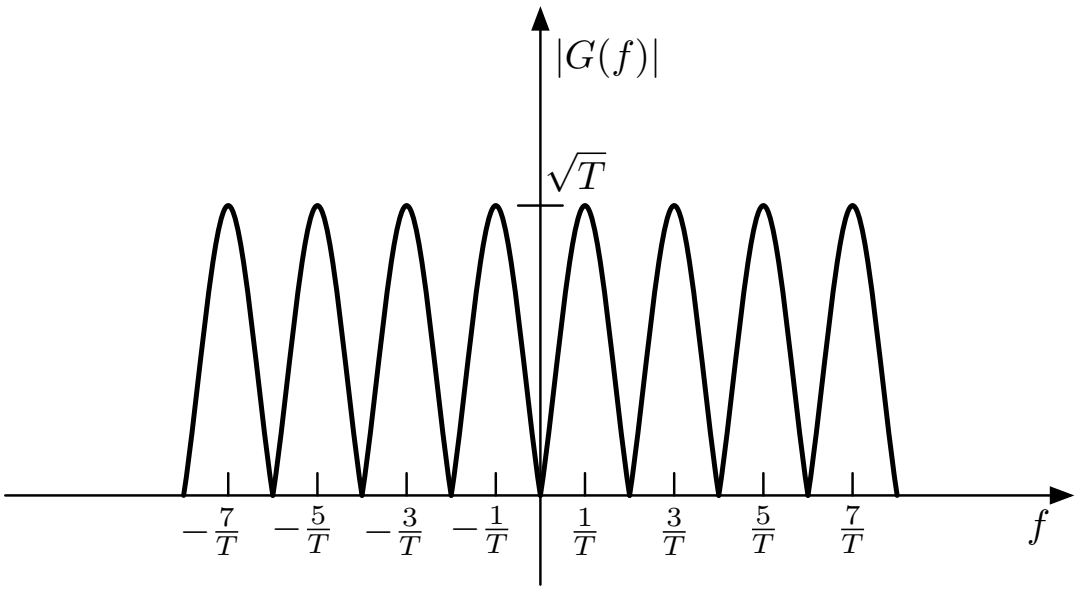


Figure 1.3: An example plot of $|G(f)|$

erties to minimize the number computations per sample. The set of complex multipliers used for modulation can also be replaced with an IFFT to further reduce computational complexity. The filter banks presented in Figure 1.1 take advantage of these computational improvements. For further details of the polyphase filter bank derivation, see [4] and [9].

1.1 Packet Detection and Timing Recovery

To choose the correct preamble sequence, the characteristics of the matched filter impulse response are studied. Packet detection and timing recovery must be performed before the AFB to maintain the correct sample timing. The received signal $y[n]$ is match filtered before being passed to the packet detection and timing recovery algorithm. This process can be seen in Figure 1.4. The transmit wide band filter, $g[n]$, has already been presented in Equation (1.1). The overall system response, excluding the channel, is given by

$$\eta[n] = g[n] \star g^*[-n] \tag{1.3}$$

[13] shows further derivations of $\eta[n]$. Following these derivations tell us that the matched filter response has three main peaks when the prototype filter is a square-root raised cosine with roll-off factor $\alpha = 1$. The main peak of the impulse response is located at $t = 0$ and has an amplitude of N. The remaining peaks are located at $\pm\frac{L}{2}$ with amplitude $-\frac{N}{2}$, where L

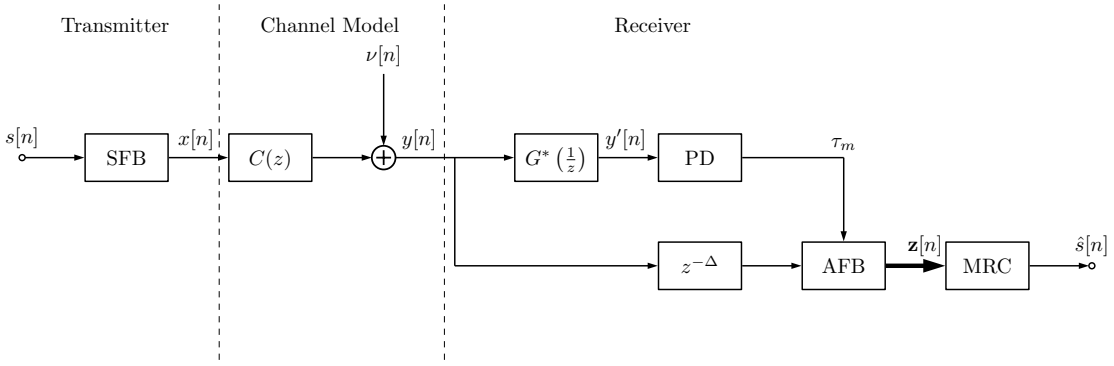


Figure 1.4: System model block diagram

is the number of samples per symbol. The impulse response $\eta[n]$ can be seen in Figure 1.5. When transmitting multiple BPSK symbols, the smaller, half symbol peaks located at $\pm\frac{L}{2}$ of consecutive symbols add together. If the consecutive symbols are alternating, these half symbol peaks add deconstructively. If the consecutive symbols are nonalternating, these half symbols peaks add constructively and create a peak with an amplitude of N located at odd integer multiple of $\frac{L}{2}$. An example of two alternating symbols and two nonalternating symbols can be seen in Figure 1.6.

The above examples can be expanded beyond BPSK symbols. Let the first transmitted symbol be s_1 and the second transmitted symbol be s_2. Both symbols are complex valued symbols located on the unit circle, meaning $|s_i| = 1$. The half symbol peak can be calculated as $\left(-\frac{N}{2}s_1 - \frac{N}{2}s_2\right)$. The magnitude of the half symbol peak can be expressed as the following

$$\sqrt{\left|\frac{N}{2}s_1\right|^2 + \left|\frac{N}{2}s_2\right|^2 + 2\left|\frac{N}{2}s_1\right|\left|\frac{N}{2}s_2\right|\cos(\theta)} \tag{1.4}$$

where θ is the phase difference between s_1 and s_2 and is in the range $[0, \pi]$. Since the transmitted symbols are located on the unit circle, Equation (1.4) can be simplified as

$$N\cos\left(\frac{\theta}{2}\right) \tag{1.5}$$

The packet detection algorithm searches for the symbol peaks. It is important to be aware

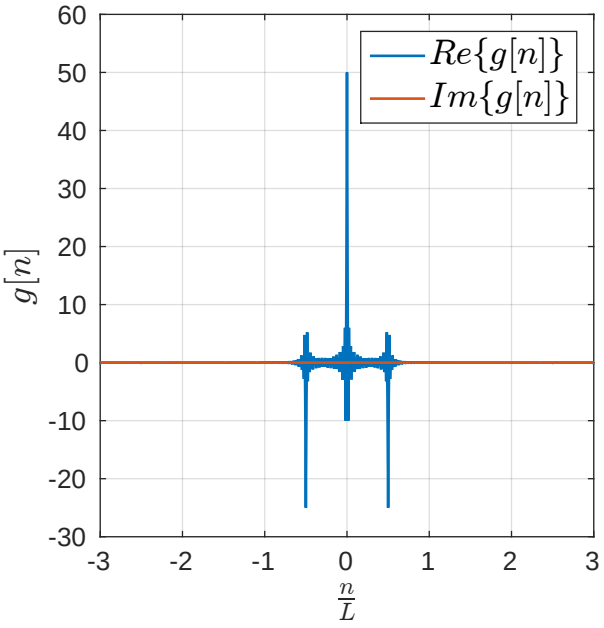


Figure 1.5: System response $\eta[n]$

of the half symbol peaks because they have the potential to be as large as the symbol peaks, making packet detection difficult to accomplish. The goal of the packet detection is to recognize the presence of the symbol peaks of the preamble sequence and notify the packet boundaries to the remaining components of the receiver. The symbol peaks are also used to correct the sample timing offset before the signal is decimated through the process of the AFB.

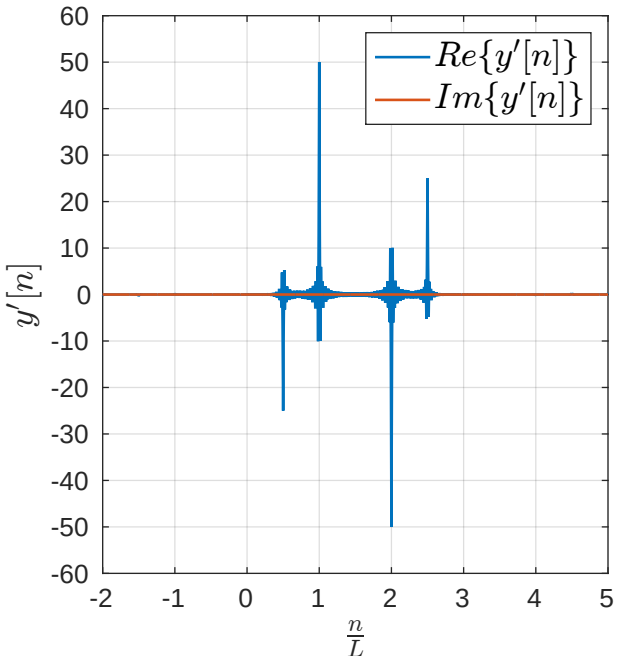
The packet detection starts with filtering the received signal $y[n]$ through the matched filter $g^*[-n]$. This gives

$$y'[n] = y[n] \star g^*[-n]\,. \tag{1.6}$$

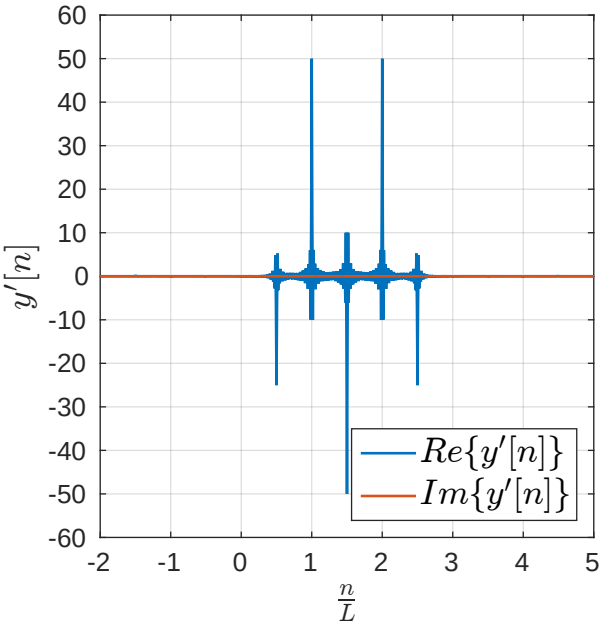
Once the signal is filtered by the matched filter, the magnitude squared of the signal is calculated, giving us

$$y'_2[n] = |y'[n]|^2\,. \tag{1.7}$$

For every L samples, the maximum sample y_m, of $y'_2[n]$, is found. The index location $\tau_m = \mathrm{mod}(n, L)$, of y_m is stored in a M length vector $\boldsymbol{\tau}$. When a packet is present, these



(a) System response (alternating symbols)



(b) System response (nonalternating symbols)

Figure 1.6: System response to alternating and nonalternating symbols

maximum values relate to the symbol peaks. When a packet is not present, these maximum values will appear at each index randomly with equal probability. Once K values in the M length vector $\boldsymbol{\tau}$ have the same index value, it is decided that a packet is present with a sample timing offset equal to that same index value. This packet detection process occurs in the block labeled PD in Figure 1.4. The received signal, $y[n]$, is also delayed by Δ, where Δ is the time required to perform matched filtering and packet detection. After a packet is detected, $y[n-\Delta]$ can be passed through the AFB with the corrected sample timing phase as shown in Figure 1.4. This packet detection and timing recovery algorithm is discussed in more detail in [12] and an implementation of the packet detection is presented in [7].

1.2 Maximum Ratio Combining

After the signal passes through the AFB, it is desired to combine the subcarriers. An equal weighted sum across all subcarriers after channel equalization is the most obvious way to combine the subcarriers. Channel equalization and an equal weighted sum can be written as

$$\hat{s}[n] = \frac{1}{N}\hat{\mathbf{C}}^H\mathbf{z}\,, \tag{1.8}$$

where $\mathbf{z} = \left[z_0[n], z_1[n], \ldots, z_{N-1}[n]\right]^T$, and

$$\hat{\mathbf{C}} = \begin{bmatrix} \hat{C}_{BB}\left(e^{j2\pi f_0}\right) \\ \hat{C}_{BB}\left(e^{j2\pi f_1}\right) \\ \vdots \\ \hat{C}_{BB}\left(e^{j2\pi f_{N-1}}\right) \end{bmatrix}. \tag{1.9}$$

A more valuable technique is to assign a weight to each subcarrier according to the SNR of the subcarrier, that is, assigning higher weight to subcarriers with higher SNR and lower weight to subcarriers with lower SNR. This combining of subcarriers is referred to as Maximum Ratio Combing (MRC) in [11]. To achieve maximum ratio combining, it is necessary to estimate the channel frequency response and the noise variance of each subcarrier. Replacing $\frac{1}{N}\hat{\mathbf{C}}$ in Equation (1.8) with $\mathbf{w}_o$, we arrive at

$$\hat{s}[n] = \mathbf{w}_o^H\mathbf{z}\,, \tag{1.10}$$

where

$$\mathbf{w}_o = \frac{1}{\sum_{k=0}^{N-1} \frac{\left|\hat{C}_{BB}\left(e^{j2\pi f_k}\right)\right|^2}{\hat{\sigma}_k^2}} \begin{bmatrix} \frac{\hat{C}_{BB}\left(e^{j2\pi f_0}\right)}{\hat{\sigma}_0^2} \\ \frac{\hat{C}_{BB}\left(e^{j2\pi f_1}\right)}{\hat{\sigma}_1^2} \\ \vdots \\ \frac{\hat{C}_{BB}\left(e^{j2\pi f_{N-1}}\right)}{\hat{\sigma}_{N-1}^2} \end{bmatrix}, \tag{1.11}$$

$\hat{C}_{BB}\left(e^{j2\pi f_k}\right)$ is an estimate of the equivalent baseband channel frequency response at the frequency f_k and $\hat{\sigma}_k^2$ is an estimate of the variance of the channel noise for the k^{th} subcarrier. The technique used to obtain these estimates varies depending on the preamble sequence used. Details on calculating these estimates are discussed in detail in Chapter 2 and Chapter 3 for the alternating $\{+1, -1\}$ sequence and pseudorandom sequence, respectively. A comparison between both techniques is discussed in Section 5.1 and future work suggestions are given in Section 5.2.

1.3 Thesis Contribution

A FB-MC-SS FPGA implementation is expanded to a packetized format. An alternating $\{+1, -1\}$ sequence and pseudorandom sequences are explored as choices for the packet's preamble. The preamble is used to detect the presence of a packet and acquire estimates for the channel frequency response and noise variance. Details on detecting the preamble are discussed in Section 2.1 for an alternating $\{+1, -1\}$ preamble and in Section 3.1 for pseudorandom sequences. Details on estimating the channel frequency response and noise variance are discussed in Section 2.2 for an alternating $\{+1, -1\}$ sequence and in Section 3.3 for pseudorandom sequences. FPGA implementation using the alternating $\{+1, -1\}$ preamble sequence is presented in [7]. Both the alternating $\{+1, -1\}$ sequence and pseudorandom sequence are implemented in FPGA.

CHAPTER 2

ALTERNATING $\{+1, -1\}$ PREAMBLE SEQUENCE

An alternating $\{+1, -1\}$ preamble sequence is explored in this chapter. An alternating $\{+1, -1\}$ may be desired because of better detection attributes and simpler implementation that leads to lower FPGA resource utilization. The effects of an alternating $\{+1, -1\}$ sequence to the packet detection are analyzed in Section 2.1. Estimating the channel frequency response and noise variance of each subcarrier using the alternating $\{+1, -1\}$ preamble sequence is presented in Section 2.2. Block diagrams for estimating the channel frequency response and noise variance are also presented in Section 2.2. The impact of interference on the estimates is explored in Section 2.3.

2.1 Detection of Alternating $\{+1, -1\}$ Sequence

Referring back to Chapter 1, one can see the impact of an alternating symbol sequence to the system response. When transmitting alternating $\{+1, -1\}$ symbols, the phase difference, θ, between any two consecutive symbols is always equal to π. Referring back to Equation (1.5), it can be seen that the magnitude at the half symbol timing will always be 0. This tells us that the amplitude at all odd multiples of $\frac{L}{2}$ will be 0 during the preamble sequence. This can be seen when looking at the system response of 20 alternating $\{+1, -1\}$ symbols in Figure 2.1. When the half symbol peaks add deconstructively, the only remaining peak in a symbol period of L samples is the symbol peak. When performing the packet detection algorithm, y_m will be detected at the symbol peak with very high probability. Hence, the packet detection performs with the best results when the preamble sequence is an alternating $\{+1, -1\}$. The probability of missed detection p_{md}, and the probability of false alarm p_{fa}, can be tuned by setting the parameters K and M. A packet is detected when K out of the last M peaks have the same τ_m as discussed in Section 1.1. The value p_{md} decreases with the ratio $\frac{K}{M}$. On the other hand, to minimize p_{fa}, $\frac{K}{M}$ should be increased.

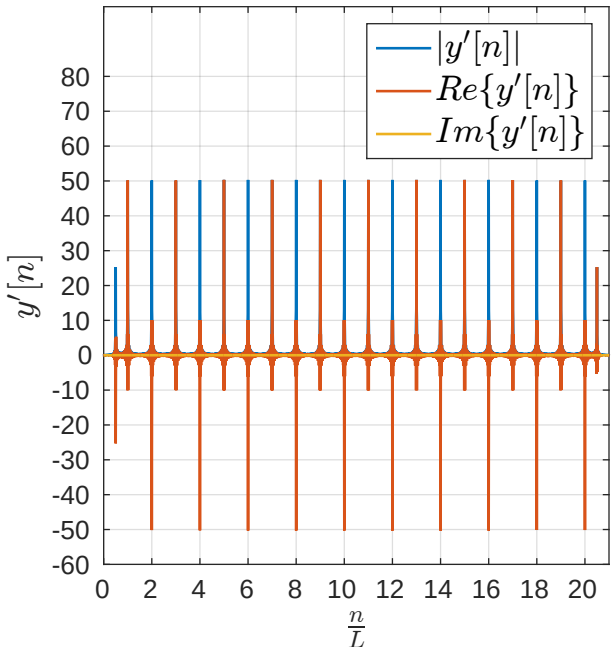
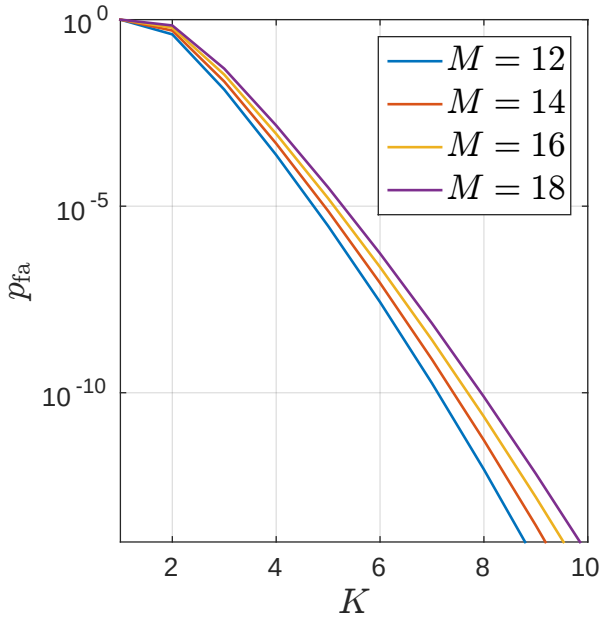


Figure 2.1: System response, alternating $\{+1, -1\}$ symbols

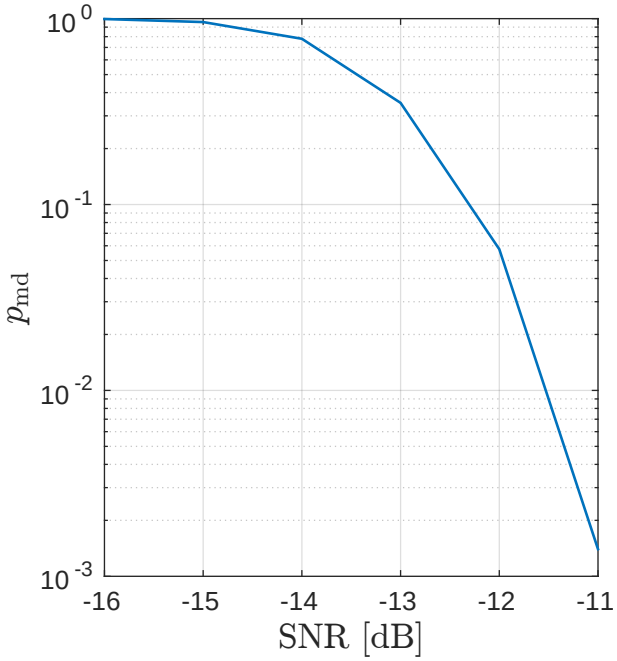
Increasing M and K while maintaining a constant p_{fa} improves p_{md}. Both p_{md} and p_{fa} are plotted in Figure 2.2. A Monte Carlo simulation with $M = 16$ and $K = 10$ is used to plot the p_{md}. A closed form equation presented in [12] is used to plot p_{fa} for a range of M and K values.

2.2 Channel and Noise Variance Estimation Based off Alternating $\{+1, -1\}$ Preamble

After the preamble has been detected using the detection algorithm discussed above, the remaining symbols in the alternating $\{+1, -1\}$ preamble sequence are used to estimate the channel frequency response, $\hat{C}_{BB}\left(e^{j2\pi f_k}\right)$, and the noise variance, $\hat{\sigma}_k^2$, for all subcarriers, where k is the subcarrier index. The outputs of the analysis filter bank $z_k[n]$ are used to obtain these estimates. The symbols transmitted are $s[n] = \pm 1$. Figure 1.4 shows a block diagram of the operations performed on $s[n]$. The symbols $s[n]$ are modulated by the Synthesis Filter Bank (SFB). The transmitted signal $x[n]$ is then filtered through the channel $c[n]$ to obtain $y[n]$. The channel model includes the process of converting the digital signal to an analog signal, frequency shifting the analog signal to a carrier frequency, transmitting the signal across an unknown wireless channel, frequency shifting the received



(a) Probability of false alarm



(b) Probability of missed detection

Figure 2.2: Alternating $\{+1, -1\}$ detection probabilities

signal back to baseband, and resampling the received signal. The received signal $y[n]$ can then be demodulated by the Analysis Filter Bank (AFB). This process results in the received subcarrier symbols $z_k[n]$ following a distribution according to

$$z_k[n] \sim \begin{cases} \mathcal{CN}\left(C_{BB}\left(e^{j2\pi f_k}\right), \sigma_k^2\right) & \text{for } \bmod(n,2) = 0 \\ \mathcal{CN}\left(-C_{BB}\left(e^{j2\pi f_k}\right), \sigma_k^2\right) & \text{for } \bmod(n,2) = 1 \end{cases}. \tag{2.1}$$

2.2.1 Channel Estimator

The Minimum Variance Unbiased (MVUB) estimator will be used to acquire an estimate of the channel frequency response $\hat{C}_{BB}\left(e^{j2\pi f_k}\right)$. In [10], the MVUB for the mean of a Gaussian random variable is shown to be the sample mean. First, every other sample is sign inverted to phase align all the symbols. Once the symbols are phase aligned, they can be represented by a single distribution, $\sim \mathcal{CN}\left(C_{BB}\left(e^{j2\pi f_k}\right), \sigma_k^2\right)$. The sample mean can then be calculated as the MVUB estimator for the channel frequency response. The process of phase aligning the symbols and calculating the sample mean is mathematically written as

$$\hat{C}_{BB}\left(e^{j2\pi f_k}\right) = \frac{1}{P}\sum_{n=0}^{P-1}\left[z_k[n]e^{j\pi n}\right] \tag{2.2}$$

where P is the preamble symbols remaining after packet detection. This process can also be seen visually using the block diagram shown in Figure 2.3.

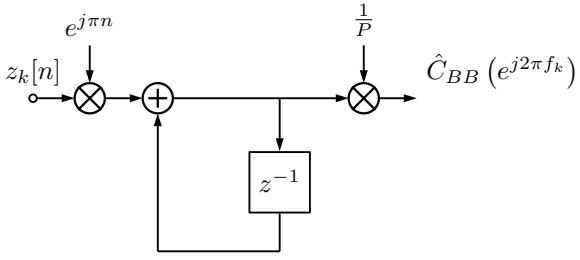


Figure 2.3: Channel frequency response estimation block diagram for alternating $\{+1, -1\}$ preamble

2.2.2 Noise Variance Estimator

Following a similar technique in Section 2.2.1, the MVUB can be developed for the variance of $z_k[n]$. For the channel estimate, the symbols were constructively added together to gain an estimate for the mean. For the noise variance estimate, the symbols are deconstructively added together to leave a signal, $\nu'_k[n]$, which is only affected by the channel noise. The noise variance can then be calculated using $\nu'_k[n]$. Since $z_k[n]$ is a sequence with alternating sign, adjacent samples can be summed to removed the mean, giving

$$\nu'_k[n] = z_k[n] + z_k[n-1]\,. \tag{2.3}$$

Since $z_k[n]$ and $z_k[n-1]$ are uncorrelated,

$$\nu'_k[n] \sim \mathcal{CN}\left(0, 2\sigma_k^2\right)\,. \tag{2.4}$$

Once the channel is removed from the $z_k[n]$, the mean of $\nu'_k[n]$ is known to be zero. When the mean of a normal distributed random variable is known to be 0, the MVUB estimate for its variance is

$$\hat{\sigma}_k^{\prime 2} = \frac{1}{P}\sum_{n=0}^{P-1} |\nu'_k[n]|^2\,. \tag{2.5}$$

The variance of $\nu_k[n]$ is thus obtained as

$$\hat{\sigma}_k^2 = \frac{\hat{\sigma}_k^{\prime 2}}{2}\,. \tag{2.6}$$

The block diagram showing the process of calculating the noise variance estimate can be seen in Figure 2.4.

2.3 Interference Issues

An alternating $\{+1, -1\}$ sequence leads to interference issues when the interference has a frequency comparable to the symbol frequency. These problems occur due to the distinct frequency of the alternating $\{+1, -1\}$ preamble sequence. Under an interference scenario, the received signal can be written as

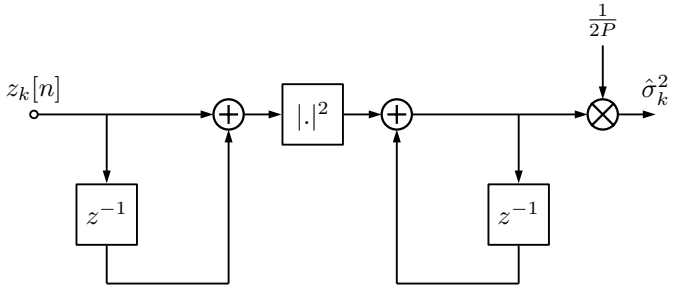


Figure 2.4: Noise variance estimation block diagram for alternating $\{+1, -1\}$ preamble

$$y[n] = x[n] \star c[n] + \nu[n] + \beta[n] \tag{2.7}$$

where $\beta[n]$ is some periodic interference. Ideally, an interference signal should increase the value for the noise variance estimate, which would lower the weight for the subcarrier during the MRC process. When the interference frequency is some odd multiple of $\frac{1}{2L}$, the channel estimate will be negatively impacted and thus corrupts the tap weights presented in Equation (1.11). When the interference frequency is some multiple of $\frac{1}{L}$, the noise variance estimate will not reflect any presence of an interference signal and the respective subcarrier will not be weighted accordingly. To gain a better understanding of each scenario, two examples are presented.

2.3.1 Example 1

An example with $\beta[n] = Ae^{(2j\pi n f_\beta + \phi)}$ is presented first to show that an interference signal with a frequency of $f_\beta = \frac{1}{2L}$ negatively impacts the channel estimate $\hat{C}_{BB}\left(e^{j2\pi f_k}\right)$ and, as a result, corrupts the tap weights $\mathbf{w}_o$. When interference is present, the AFB output is

$$z_k[n] = C_{BB}\left(e^{j2\pi f_k}\right) s[n] + \nu_k[n] + \beta'[n]\,, \tag{2.8}$$

where

$$\beta'[n] = |G(f_\beta)| A e^{(2j\pi n f_{\beta'} + \phi)}\,, \tag{2.9}$$

$f_{\beta'} = L \times f_\beta$ due to the decimation by L in the AFB, and $\nu_k[n]$ is the subcarrier noise at the output of the AFB. For the current example $f_{\beta'} = \frac{1}{2}$, meaning $\beta'[n]$ is some alternating

$\pm|G(f_\beta)|A$ sequence. The constant $|G(f_\beta)|A$ is a combination of initial amplitude A of the interference, and the gain of the filter at $f = f_\beta$. Following the same discussion presented in Section 2.2, the distribution in Equation (2.1) can be rewritten as

$$z_k[n] \sim \begin{cases} \mathcal{CN}\left(C_{BB}\left(e^{j2\pi f_k}\right) + |G(f_\beta)|A, \sigma_k^2\right) & \text{for } \bmod(n,2) = 0 \\ \mathcal{CN}\left(-C_{BB}\left(e^{j2\pi f_k}\right) - |G(f_\beta)|A, \sigma_k^2\right) & \text{for } \bmod(n,2) = 1 \end{cases} \tag{2.10}$$

Calculating the channel estimate using Equation (2.2) results in the additional term $|G(f_\beta)|A$, from the interference $\beta'[n]$, being estimated as part of the channel frequency response for the given subcarrier. This gain of $|G(f_\beta)|A$ will be carried over into Equation (1.11) and affect the weight of the subcarrier in the MRC. The effect of the interference on the tap weights can be seen in Figure 2.5 for an ideal channel $c[n] = \delta[n]$. Since the channel estimate is increased due to $|G(f_\beta)|A$, the weight of the interfered subcarrier is increased instead of being decreased.

2.3.2 Example 2

For the second example, a similar $\beta[n]$ is used. The frequency $f_\beta = \frac{1}{L}$ is adjusted to demonstrate the second problem that may occur during periodic interference with an alternating $\{+1, -1\}$ preamble sequence. After the decimation in the AFB, $\beta'[n]$ will have a normalized frequency of $f_{\beta'} = 1$, i.e. $\beta'[n] = |G(f_\beta)|A$. Accordingly, when calculating $\hat{C}_{BB}\left(e^{j2\pi f_k}\right)$ according to Equation (2.2), the effect of the interference is completely removed. During interference, $z_k[n]$ follows the form shown in Equation (2.8). When calculating $\hat{\sigma}_k^2$ according to Equation (2.5) and Equation (2.6), the presence of interference $\beta[n]$ has no effect since $f_{\beta'} = 1$. Accordingly, the interference has no affect on $\mathbf{w}_o$ as demonstrated in Figure 2.6 and the interference rejection capability of FB-MC-SS will be lost.

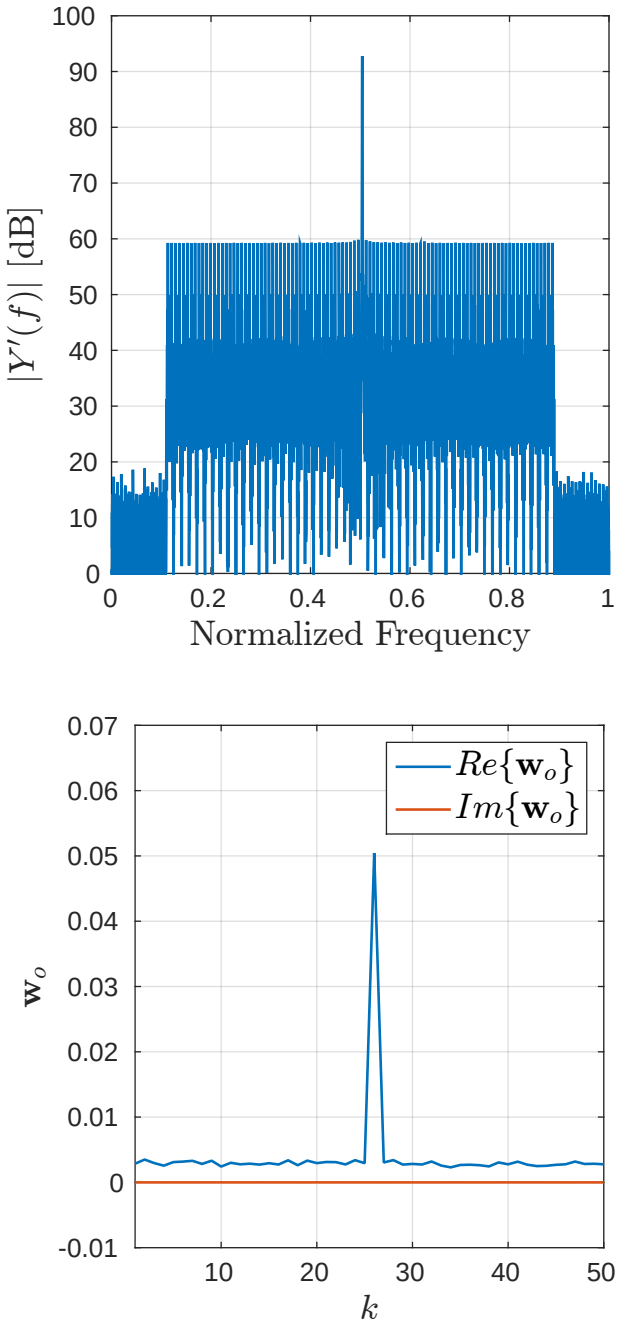


Figure 2.5: Alternating $\{+1, -1\}$, example 1 interference

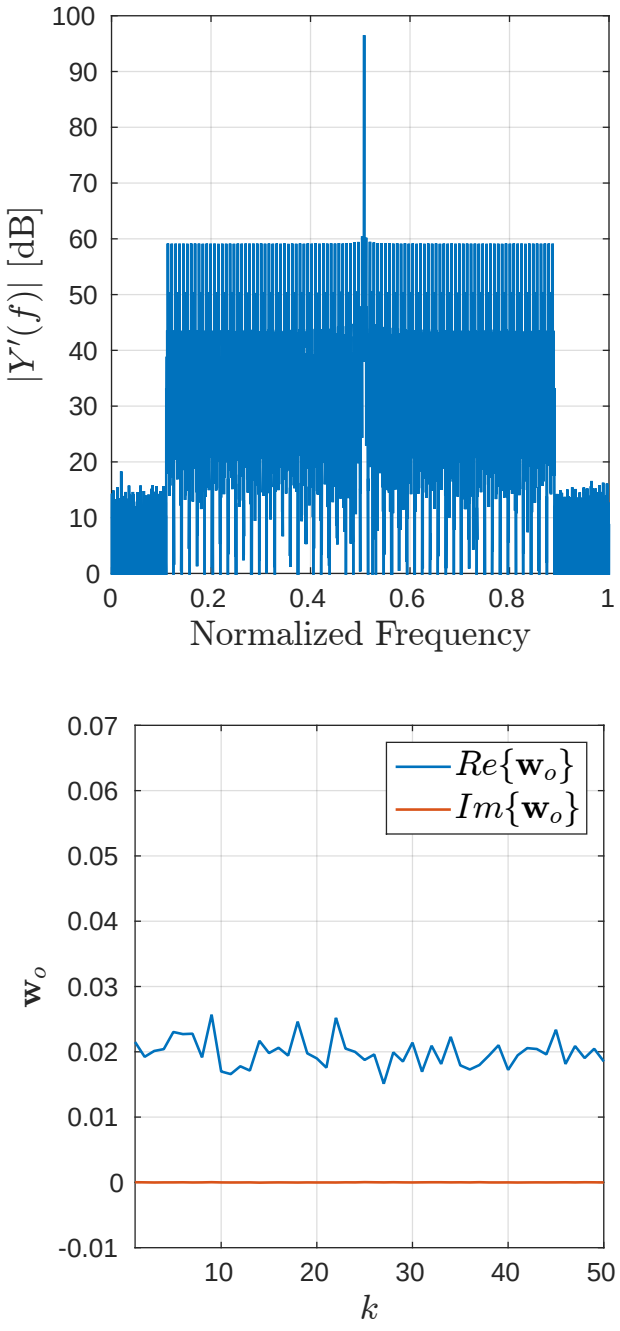


Figure 2.6: Alternating $\{+1, -1\}$, example 2 interference

CHAPTER 3

PSEUDORANDOM PREAMBLE SEQUENCE

To improve on the channel frequency and noise variance estimates, specifically the interference issues discussed in Section 2.3, the use of a pseudorandom preamble is explored in this chapter as an alternative to the alternating $\{+1, -1\}$ preamble sequence. Although a pseudorandom preamble introduces complications in the detection process and increases the implementation complexity, it significantly increases the robustness of the estimators of channel frequency response and noise variance. The detection process of a pseudorandom sequence is presented in Section 3.1. A pseudorandom preamble requires the addition of a packet alignment process between the packet detection and the subsequent steps that involve channel frequency response estimation and noise variance estimation. The packet alignment process is discussed in Section 3.2. The channel frequency response and noise variance are estimated in a similar but more complicated manner. These are presented in Section 3.3. The work discussed in this chapter is limited to two types of pseudorandom sequence, Maximum Length (ML) sequence [6] and polyphase sequence [3]. These pseudorandom sequences have correlation traits that are desirable for the packet alignment. Any ML sequence $\mathbf{p}_{\mathrm{ml}}$ of length N_{p} has an autocorrelation equal to N_{p} for the lag of zero. That is,

$$\mathbf{p}_{\mathrm{ml}}^{T}\mathbf{p}_{\mathrm{ml}} = N_{\mathrm{p}}\,. \tag{3.1}$$

Also, the correlation between $\mathbf{p}_{\mathrm{ml}}$ and a cyclically shifted version of it is equal to -1. That is,

$$\mathbf{p}_{\mathrm{ml}}^{T} C\left(\mathbf{p}_{\mathrm{ml}}, l\right) = -1 \quad \text{for} \quad l \neq N_{\mathrm{p}} \tag{3.2}$$

where $C(.\,, l)$ represents a cyclic shift of l shifts. Similarly, a polyphase sequence, $\mathbf{p}_{\mathrm{p}}$, has

correlations

$$\mathbf{p}_{\mathrm{p}}^{H}\mathbf{p}_{\mathrm{p}} = N_{\mathrm{p}}\,, \tag{3.3}$$

and

$$\mathbf{p}_{\mathrm{p}}^{H}C\left(\mathbf{p}_{\mathrm{p}}, l\right) = 0 \quad \text{for} \quad l \neq N_{\mathrm{p}}\,. \tag{3.4}$$

3.1 Detection of Pseudorandom Sequence

Referring back to Chapter 1, one can study the impact of a pseudorandom preamble sequence on the system response. When transmitting a ML sequence, the phase difference θ between two symbols is either π or 0. When $\theta = \pi$, Equation (1.5) is calculated to 0. When $\theta = 0$, Equation (1.5) is calculated to N. This tells us that the amplitude at odd multiples of $\frac{L}{2}$ will either be 0, or N. This can be seen when looking at the system response of a ML sequence in Figure 3.1. Alternatively, when a polyphase sequence is transmitted, the phase, θ, between two consecutive symbols can take any value in the range $[0, \pi]$. According to Equation (1.5), this results in a half symbol peak with a magnitude in the range $[0, N]$. The matched filter response of a transmitted polyphase sequence can be

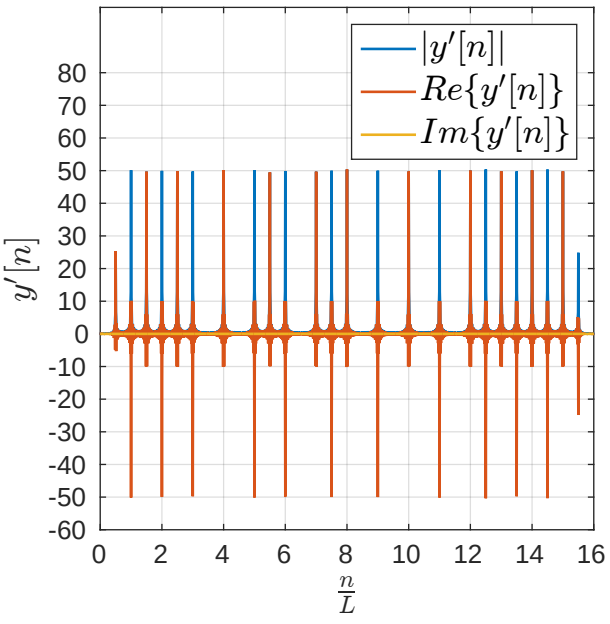


Figure 3.1: System response, ML sequence

seen in Figure 3.2. Unlike the alternating $\{+1, -1\}$ preamble sequence, the pseudorandom sequences have peaks present at odd multiples of $\frac{L}{2}$. These peaks make it more difficult for a detection to occur because y_m has the potential to be detected at $\frac{L}{2}$ instead of the correct timing of L. This problem can be combated with both the ML and polyphase sequences by carefully selecting the sequence parameters. The ML sequence can be chosen such that at the start of the sequence, there are more alternating symbols. Since the symbol peak is present for every symbol and the half symbol peak is only present when two consecutive symbols are nonalternating, the probability that the correct symbol timing is chosen is increased at the beginning of the preamble. For the polyphase sequence, the phase difference θ between consecutive symbols can be maximized and limit the chance of detecting the peak at the half symbol timing. When using a polyphase sequence, the half symbol peaks rarely have a value as large as N. This increases the probability that the y_m will be detected at the symbol timing and makes polyphase sequences a more desirable pseudorandom sequence over ML sequences. The p_{md} is presented for ML and polyphase sequences in Figure 3.3 using the same parameters of $M = 16$ and $K = 10$ discussed in Section 2.1. Since p_{fa} is not dependent on the transmitted signal it remains the same as Figure 2.2.

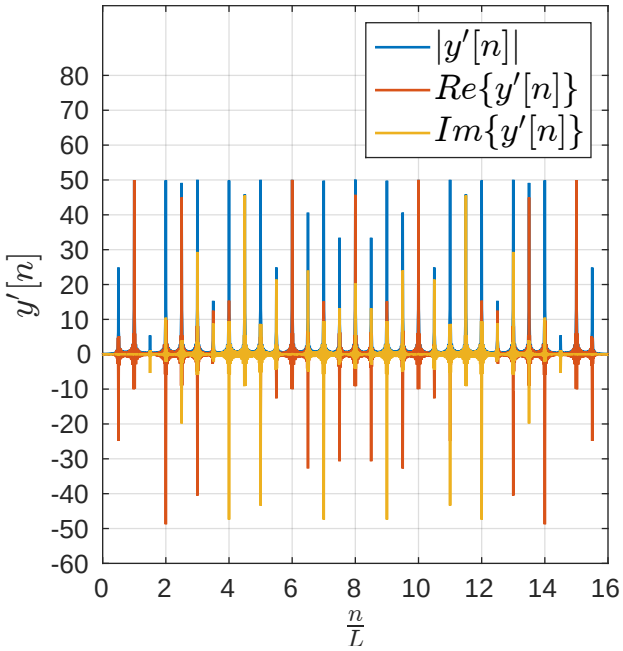


Figure 3.2: System response, polyphase sequence

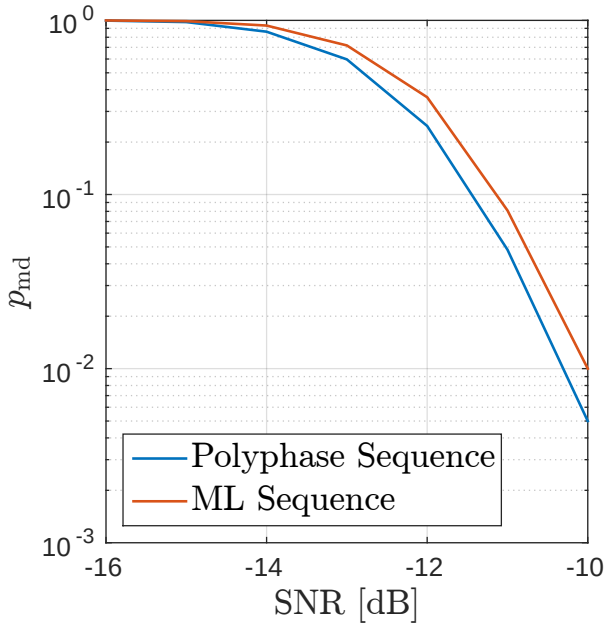


Figure 3.3: Probability of missed detection for pseudorandom preambles

3.2 Packet Alignment

Packet detection can detect a packet as early as the K^{th} symbol of the packet; under noisier conditions, the detection can occur a few symbols later. To ensure correct detection and packet processing, packets are enforced to be detected within the first P_1 symbols. Any detection later than P_1 symbols results in a missed detection. When using an alternating $\{+1, -1\}$ preamble sequence, this detection ambiguity results in a chance that the channel frequency response be estimated with a π phase ambiguity. If the first symbol detected in the packet detection is a $+1$, $\hat{C}_{BB}\left(e^{j2\pi f_k}\right)$ will be estimated correctly. If the first symbol detected in the packet detection is a -1, $\hat{C}_{BB}\left(e^{j2\pi f_k}\right)$ will be estimated with a π rotation. To correct this π rotation, a symbol sequence called the start of frame (SOF) is transmitted between the preamble and the payload. The SOF is known a priori to the receiver. The SOF is used to indicate the start of the payload and is used to recognize if a π rotation has occurred. If the SOF sequence has a π rotation, it is corrected throughout the remaining symbols of the packet. Since a pseudorandom sequence cannot be used to estimate the channel and noise variance if there is any ambiguity in the symbol timing, packet alignment is necessary. The packet alignment ensures that the channel and noise variance estimates

start on symbol P_1 for any detection between symbols K and P_1. The preamble is structured into two sequences to allow for this packet alignment. First, there are two periods of a sequence of length P_1. Second, there is a single period of a sequence of length P_2. The total preamble sequence length after packet detection is $P = P_1 + P_2$ because the first P_1 symbols are used for detection. If $\mathbf{p}_1$ is the sequence of length P_1 and $\mathbf{p}_2$ is the sequence of length P_2, the entire preamble $\mathbf{p}$ is structured as

$$\mathbf{p} = [\mathbf{p}_1 \, \mathbf{p}_1 \, \mathbf{p}_2] \tag{3.5}$$

To align the packet, the output of the matched filter $g[n]$ is decimated by L at the symbol timing τ_m, and the first P_1 symbols are used to obtain the sequence $\hat{\mathbf{p}}_1^*$. A dot product is computed between $\hat{\mathbf{p}}_1$ and every possible cyclic shift of $\mathbf{p}_1$.

$$\mathbf{r}_{\hat{p}_1 p_1} = \hat{\mathbf{p}}_1^H C(\mathbf{p}_1, l) \quad \text{for} \quad l = 0, 1, 2, \ldots, P_1 - 1 \tag{3.6}$$

The $\arg\max(\mathbf{r}_{\hat{p}_1 p_1})$ results in the index of symbol P_1 regardless of which symbol caused the packet to be detected. This index is used as the start index for channel frequency response and noise variance estimates for pseudorandom preamble sequences.

3.3 Channel and Noise Variance Estimation Based off Pseudorandom Preamble

After the packet has been aligned correctly, the remaining preamble symbols are used for channel frequency response and noise variance estimation. A similar process used for the alternating $\{+1, -1\}$ preamble sequence is used for the pseudorandom sequence. The transmitted symbols are $s[n] = e^{j\phi}$, where ϕ is some phase in the range $[0, 2\pi]$. The symbols still follow the process presented in Figure 1.4. The AFB output, $z_k[n]$, is used for both channel frequency response and noise variance estimates. The techniques used for each estimates are explained in the following subsections.

3.3.1 Channel Estimator

The MVUB estimator is used to gain an estimate for the channel frequency response of each subcarrier k. For an alternating $\{+1, -1\}$ preamble sequence, every other symbol was sign inverted to gain a distribution according to

$$z_k[n]e^{j\pi n} \sim \mathcal{CN}\left(C_{BB}\left(e^{j2\pi f_k}\right), \sigma_k^2\right) \tag{3.7}$$

Similarly, a pseudorandom sequence is demodulated to baseband by multiplying it by its complex conjugate to obtain the same distribution

$$z_k[n]p^*[n] \sim \mathcal{CN}\left(C_{BB}\left(e^{j2\pi f_k}\right), \sigma_k^2\right) \tag{3.8}$$

where $p[n]$ is sample n of $[\mathbf{p}_1\, \mathbf{p}_2]$. Once the distribution in Equation (3.8) is obtained, the channel frequency response can be estimated by calculating the sample mean, similarly to Equation (2.2). A block diagram of the channel frequency response estimation is presented in Figure 3.4.

3.3.2 Noise Variance Estimator

Following a technique similar to the one in Section 3.3.1, the MVUB is found for the variance $z_k[n]$. For the channel estimate, the symbols were constructively added together to acquire an estimate for the mean. For the noise variance estimate, the symbols impact is removed from $z_k[n]$ so the variance can be estimated. For an alternating $\{+1, -1\}$ sequence, this was done by simply summing adjacent symbols, as shown in Equation (2.3). For pseudorandom preamble sequences, $z_k[n]$ must first be modulated to have the opposite polarity as $z_k[n-1]$ before they can be summed. One possible method is to rewrite Equation (2.3) as

$$\nu_k'[n] = z_k[n]p^*[n](-p[n-1]) + z_k[n-1]\,. \tag{3.9}$$

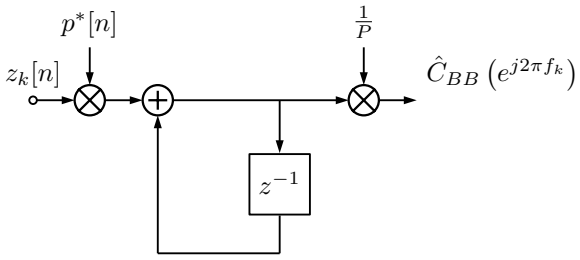


Figure 3.4: Channel frequency response estimation block diagram for pseudorandom preambles

Once $\nu'_k[n]$ is obtained, the remaining steps of the noise variance estimation of an alternating $\{+1, -1\}$ sequence can be followed to obtain the MVUB estimate of the variance. These steps are presented in Equation (2.5) and Equation (2.6). The updated block diagram for estimating the variance when using a pseudorandom preamble can be seen in Figure 3.5. When using a polyphase sequence, the multiplications in Equation (3.8) and Equation (3.9) are complex multiplications, requiring four multipliers each. When using an ML sequence, Equation (3.8) and Equation (3.9) only require a sign inversion that does not require any multipliers. This is a consideration to keep in mind when deciding between ML or polyphase sequences. If FPGA multiplier utilization is limited, a ML sequence may be a more desirable choice for the preamble.

3.4 Interference Enhancements

The aim of using a pseudorandom preamble sequence is to make the channel frequency response and noise variance estimates more robust to interference. By using a pseudorandom sequence, the distinct frequency of alternating $\{+1, -1\}$ is removed. This allows for more reliable estimation of the channel frequency response and noise variance under interference with all types of interference frequencies. The resource utilization is increased as a trade off of increasing the robustness of the estimations. The robustness of the estimates will be shown by following the same two examples discussed in Section 2.3 and Equation (2.7) is used to represent the channel model under interference.

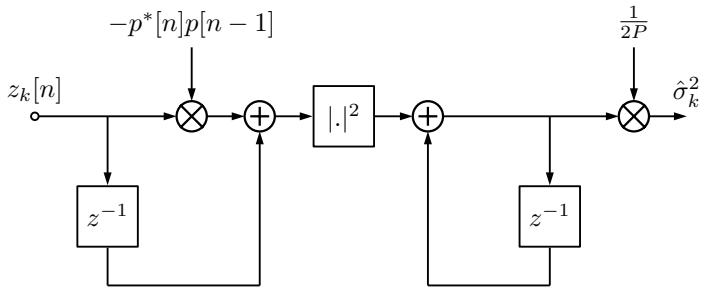


Figure 3.5: Noise variance estimation block diagram for pseudorandom preambles

3.4.1 Example 1

As in the example presented in Section 2.3, $\beta[n] = Ae^{(2j\pi n f_\beta + \phi)}$. In Section 2.3, when using an alternating $\{+1, -1\}$ preamble, the channel estimate was increased when $f_\beta = \frac{1}{2L}$. The increased channel estimate results in a high weight for the subcarrier with the interference present, which is the opposite of the desired effect. It is desired to decrease the weight of a subcarrier with interference present. Plugging the interfered AFB output from Equation (2.8) into Equation (3.8) gives

$$z_k[n]p^*[n] \sim \mathcal{CN}\left(C_{BB}\left(e^{j2\pi f_k}\right) + \epsilon_\beta, \sigma_k^2 + \sigma_\beta^2\right) \tag{3.10}$$

where ϵ_β is some error from the interference $\beta[n]$. Although the interference still introduces an error to the channel estimate, it is more important to note that the noise variance increases when using a pseudorandom preamble. The large variance estimate will minimize the weight of the subcarrier and the interferer will have minimal effect on the estimated symbol $\hat{s}[n]$. In the case of an alternating $\{+1, -1\}$ sequence, $\nu'_k[n]$ was obtained by summing adjacent symbols, as shown in Equation (2.3). For the case with interference present, the interference from the adjacent symbols deconstructs, thus removing the effect of the interference in $\nu'_k[n]$. When the variance is estimated according to Equation (2.5) and Equation (2.6), it will not be increased due to the interference. When a pseudorandom preamble is used, instead of summing adjacent symbols according to Equation (2.3), Equation (3.9) is used. The modulation of $z_k[n]$ in Equation (3.9) ensures that the effect of $\beta'[n]$ remains present in $\nu'_k[n]$. Using Equation (2.5) and Equation (2.6) to acquire the estimate $\hat{\sigma}_k^2$ results in a large value, which is an estimate for $(|G(f_\beta)|A)^2 + \sigma_k^2$. By using a pseudorandom preamble sequence, the effect of $\beta'[n]$ was maintained in $\nu'_k[n]$, which increased the estimate of $\hat{\sigma}_k^2$ by $(|G(f_\beta)|A)^2$. A large value for the estimate $\hat{\sigma}_k^2$ results in a low weight, as seen in Equation (1.11). This can also be seen by the received spectrum and calculated tap weights in Figure 3.6.

3.4.2 Example 2

Similar to Section 2.3.2, the frequency f_β is set to $\frac{1}{L}$ to demonstrate the enhancements of a pseudorandom preamble. For $\frac{1}{L}$, $\beta'[n] = |G(f_\beta)|A$. When an alternating $\{+1, -1\}$ sequence was used, $\beta'[n]$ simply adjusted the mean of $\nu'_k[n]$ and had no effect on the variance.

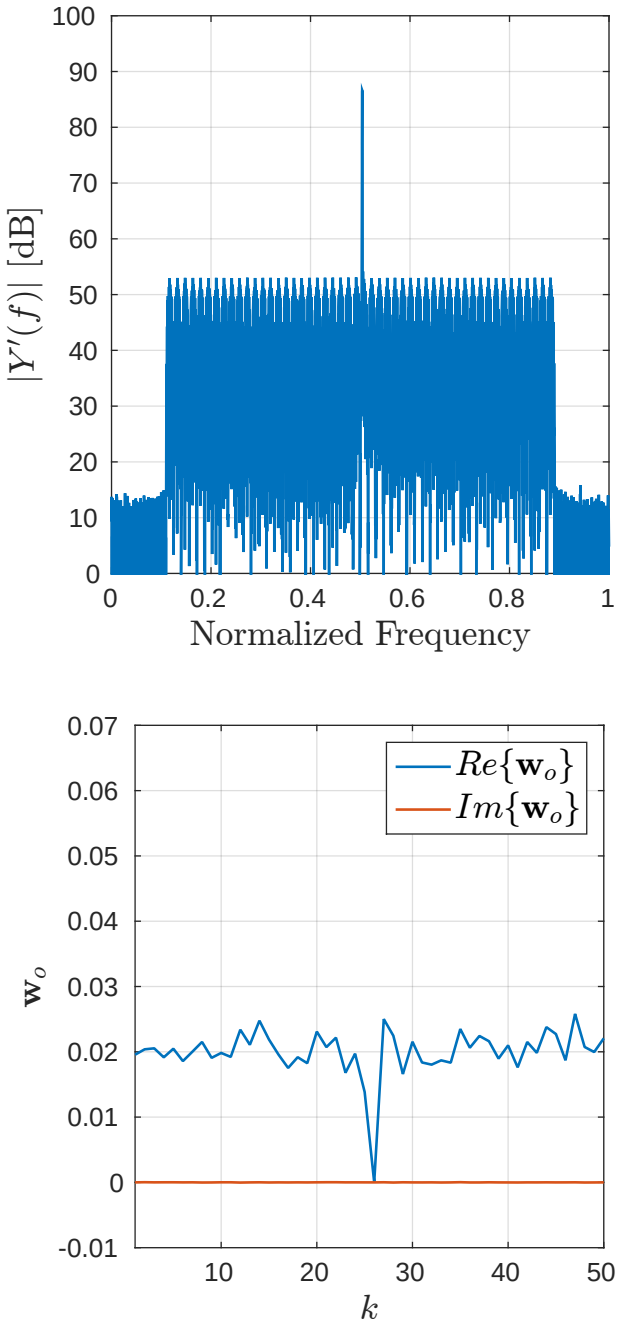


Figure 3.6: Pseudorandom, example 1 interference

It is desired for the variance to increase under interference to lower the weight given to the subcarrier. Since adjacent values of $z_k[n]$ are not simply summed, as in Equation (2.3), when using Equation (3.9) for a pseudorandom preamble, $\beta'[n]$ does affect the variance of $\nu'_k[n]$. Similar to Section 3.4.1 above, the variance estimate $\hat{\sigma}_k^2$ estimates $(|G(f_\beta)|A)^2 + \sigma_k^2$. This large interference value will minimize the weight of the interfered subcarrier during MRC, as shown in Equation (1.10). These results can be seen in Figure 3.7.

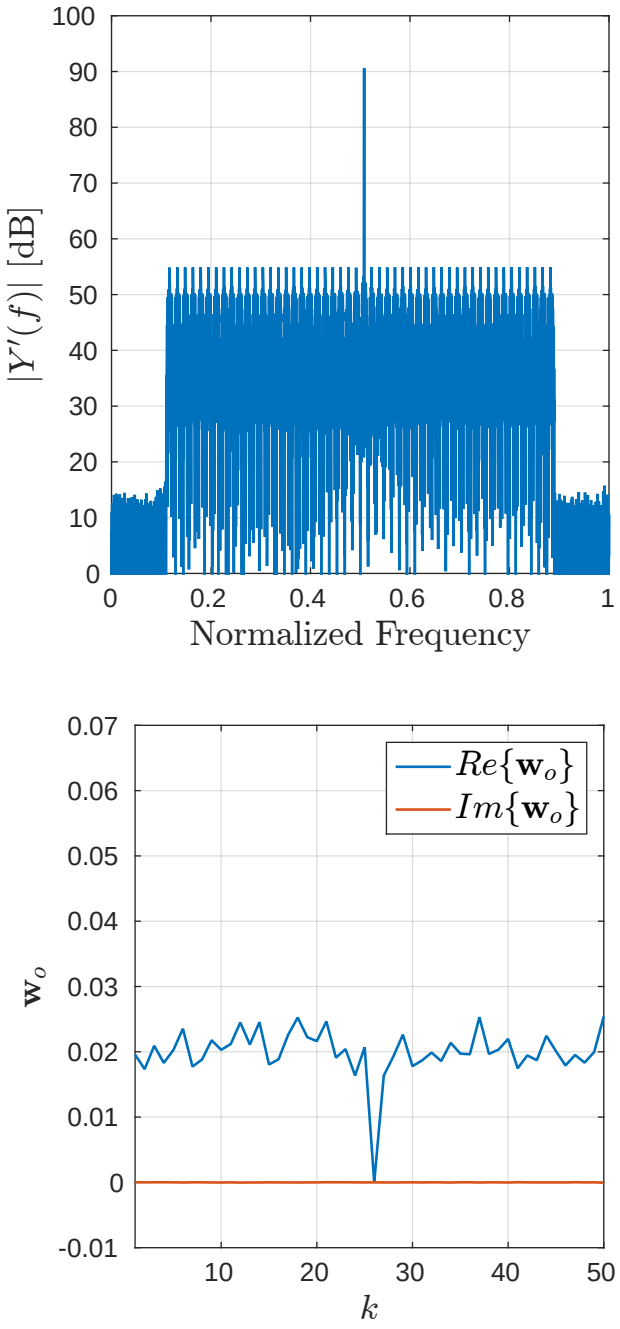


Figure 3.7: Pseudorandom, example 2 interference

CHAPTER 4

FPGA IMPLEMENTATION

The resource utilization of the FPGA implementation for each preamble sequence will be discussed in this chapter. An alternating $\{+1, -1\}$ sequence and a polyphase sequence were implemented on a Xilinx ZC706 evaluation board featuring a XC7Z045 Zynq System on Chip (SoC). An alternating $\{+1, -1\}$ sequence leads to the lowest resource utilization. When using an alternating $\{+1, -1\}$ preamble sequence, the multiplication in Equation (2.2) can be implemented using a two's complement circuit thus allowing dedicated multiplier resources to be saved. An alternating $\{+1, -1\}$ preamble sequence further saves on multiplier resources by not requiring the packet alignment circuit. A ML sequence also saves on multiplier resources by utilizing a two's compliment for the multiplication in Equation (3.8) and Equation (3.9), but the ML sequence does require the packet alignment circuit. When a ML sequence is used, the multiplications used in Equation (3.6) for the packet alignment process can also be implemented as a two's compliment. The polyphase pseudorandom sequence is the most resource-intensive preamble sequence. Since the polyphase sequence is comprised of complex symbols, multipliers must be used to compute the multiplications in Equation (3.8), Equation (3.9), and Equation (3.6). To further understand the resource utilizations of each preamble sequence, a two's compliment circuit and a complex multiplier circuit are presented in Figure 4.1 and Figure 4.2, respectively. The two's compliment circuit only requires a single register delay, while the complex multiplier requires a delay of four registers.

4.1 System Resources

The final FB-MC-SS system resources are shown in Table 4.1 and Table 4.2 for an alternating $\{+1, -1\}$ preamble sequence and a polyphase preamble sequence, respectively. The most significant figure to note in the tables is the DSP resource utilization. The DSP slice is the Xilinx resource used to perform binary multipliers and accumulators; therefore,

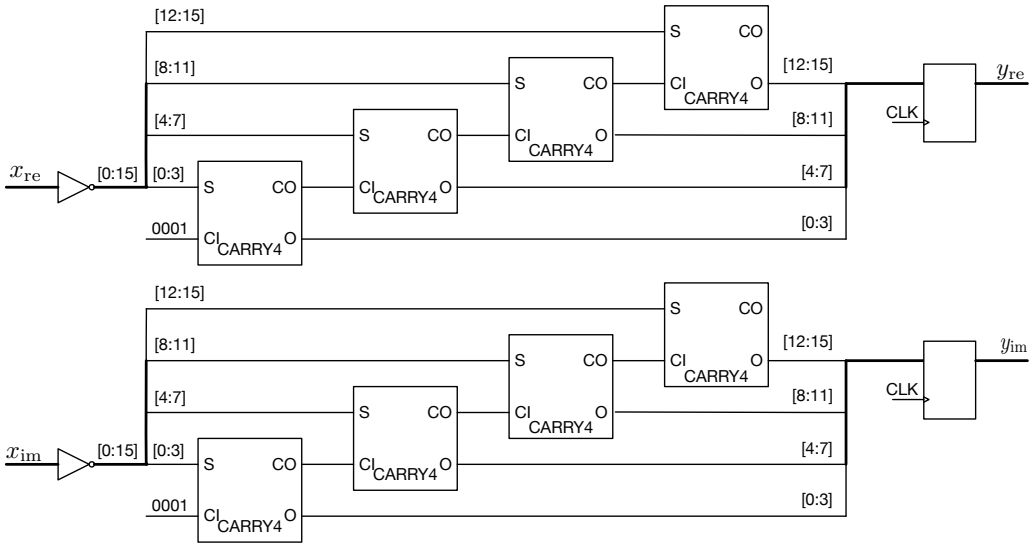


Figure 4.1: Two's compliment circuit

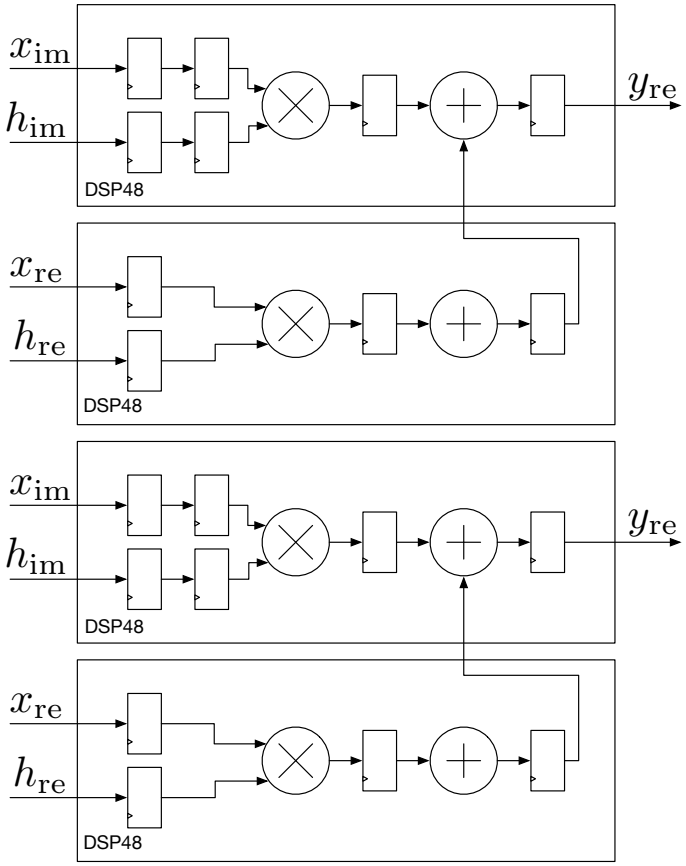


Figure 4.2: Complex multiplier circuit

Table 4.1: FB-MC-SS alternating $\{+1, -1\}$ preamble system utilization.

Resource	Utilization	Available	Utilization %
Slice LUTs	117573	218600	53.78
Slice Registers	97120	437200	22.21
Memory	106	545	19.45
DSP	374	900	41.56
IO	128	444	28.83
Clocking	11	32	34.38

Table 4.2: FB-MC-SS polyphase preamble system utilization

Resource	Utilization	Available	Utilization %
Slice LUTs	132960	218600	60.82
Slice Registers	113238	437200	25.90
Memory	112	545	20.55
DSP	504	900	56.00
IO	128	444	28.83
Clocking	11	32	34.38

it is the most valuable resource in Digital Signal Processing FPGA applications. The system with the polyphase sequence preamble utilized a much higher number of DSP slices due to the complex multiplication required in Equation (3.8), Equation (3.9), and Equation (3.6). Both Equation (3.8) and Equation (3.9) require a single complex multiplication circuit. Computing Equation (3.6) requires an array of P_1 complex multiplications.

CHAPTER 5

CONCLUSION AND FUTURE WORK

The work performed throughout this thesis is summarized in Section 5.1 and the possibilities of future work are discussed in Section 5.2.

5.1 Conclusion

This thesis performed a thorough study of the various choices of preamble sequence in a filterbank multicarrier spread spectrum (FB-MC-SS) system. The most straightforward preamble choice leads to an alternating $\{+1, -1\}$ sequence. The best packet detection traits occur for an alternating $\{+1, -1\}$ preamble. An alternating $\{+1, -1\}$ sequence also leads to better resource utilization. The shortcomings of an alternating $\{+1, -1\}$ sequence were also presented. Due to the distinct frequency present in the sequence, it was shown that channel frequency response and noise variance estimates can be easily corrupted by interference when the interference frequency is some multiples of the symbol period. Corrupted channel frequency response and noise variance estimates result in poor maximum ratio combining. A pseudorandom preamble sequence was then proposed to overcome the limitations presented by an alternating $\{+1, -1\}$ sequence. Two types of pseudorandom sequences were explored; ML sequence and polyphase sequence. Although polyphase sequence require more FPGA resources, it was still found that polyphase sequences are more desirable than ML sequences. Polyphase sequences have better correlation traits that enhance the packet alignment process. Detection of polyphase sequence is preferred over ML sequence because the half symbol peaks are always smaller than the symbol peaks. The half symbol peaks for ML sequences are equal in magnitude to the symbol peaks, making packet detection more difficult. Since ML sequences are a sequence of binary $+1$ and -1, packet alignment, channel frequency response estimation, and noise variance estimation can be performed without multipliers. ML sequences may be the only option if there is a shortage of multiplier recourses. The algorithms described in Chapter 3 have also been implemented in a Xilinx FPGA to verify

the resource utilization.

5.2 Future Work

The work discussed throughout this thesis was focused on packetizing an FB-MC-SS system and designing a reliable preamble sequence that could be used for packet detection and estimating channel information. Further research can be performed to expand on the work of this thesis. Although the polyphase sequence works quite well, it may be beneficial to explore other pseudorandom sequences. The goal of a pseudorandom code is to minimize the half symbol peaks, have good correlation properties, and maintain accurate estimates during interference. Polyphase sequences have smaller half symbol peaks than ML sequence, but a new pseudorandom sequence design could minimize the half symbol peaks further without sacrificing correlation properties or robustness to interference.

The preamble is used for packet detection, and channel frequency response and noise variance. Packet detection occurs within the first P_1 symbols of the preamble and the remaining P symbols are used for channel frequency response and noise variance estimation. P is significantly larger than P_1 because the channel frequency response and noise variance require many symbols to acquire an accurate estimate. Techniques to shorten the preamble length, specifically P, should be explored to maximize the efficiency of a packet. Denoising the channel frequency response should be explored to acquire an estimate with an equivalent minimum mean square error with less symbols. However, shortening the number of preamble symbols required to obtain estimation of the noise variance presents more difficulties that lead to a good area for further research.

REFERENCES

[1] K. Cheun, K. Choi, H. Lim, and K. Lee, *Antijamming performance of a multicarrier direct-sequence spread-spectrum system*, IEEE Transactions on Communications, 47 (1999), pp. 1781–1784.

[2] Y. S. Cho, J. Kim, W. Y. Yang, and C. G. Kang, *Introduction to OFDM*, Wiley-IEEE Press, 2010, pp. 111–151.

[3] D. Chu, *Polyphase codes with good periodic correlation properties (corresp.)*, IEEE Transactions on Information Theory, 18 (1972), pp. 531–532.

[4] B. Farhang-Boroujeny, *Signal processing techniques for software radios*, Lulu Publishing House, 2nd ed., 2010.

[5] B. Farhang-Boroujeny and C. Furse, *A robust detector for multicarrier spread spectrum transmission over partially jammed channels*, IEEE Transactions on Signal Processing, 53 (2005), pp. 1038–1044.

[6] S. W. Golomb and G. Gong, *Signal design for good correlation: For wireless communication, cryptography, and radar*, Cambridge University Press, New York, NY, USA, 2004.

[7] T. Haddadin, S. A. Laraway, A. Majid, T. Sibbett, D. L. Wasden, B. F. Lo, L. Landon, D. Couch, H. Moradi, and B. Farhang-Boroujeny, *An underlay communication channel for 5G cognitive mesh networks: Packet design, implementation, analysis, and experimental results*, in 2016 IEEE International Conference on Communications Workshops (ICC), May 2016, pp. 498–504.

[8] S. Hara and R. Prasad, *Overview of multicarrier cdma*, IEEE Communications Magazine, 35 (1997), pp. 126–133.

[9] F. J. Harris, *Multirate signal processing for communication systems*, Prentice Hall PTR, Upper Saddle River, NJ, USA, 2004.

[10] S. M. Kay, *Fundamentals of statistical signal processing. [Volume I]. , Estimation theory*, Prentice Hall signal processing series, Prentice Hall, Upper Saddle River (N.J.), 1993. Autre rimpr. 2013.

[11] J. G. Proakis, *Digital communications*, McGraw-Hill series in electrical and computer engineering, McGraw-Hill, New York, 1995.

[12] T. Sibbett, H. Moradi, and B. Farhang-Boroujeny, *Design and analysis of a novel timing acquisition method for spread-spectrum systems.* (in press), 2016.

[13] D. L. Wasden, H. Moradi, and B. Farhang-Boroujeny, *Design and implementation of an underlay control channel for cognitive radios*, IEEE Journal on Selected Areas in Communications, 30 (2012), pp. 1875–1889.