HYPER-WIDEBAND OFDM SYSTEM

A Thesis
Presented to
The Academic Faculty

by

Edward Tan

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
May 2016

HYPER-WIDEBAND OFDM SYSTEM

Approved by:

Dr. Stephen E. Ralph, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. John R. Barry
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Justin K. Romberg
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: April 27, 2016

ACKNOWLEDGEMENTS

In the course of completing my thesis, many people played a positive impact in making the journey enlightening and enjoyable. First and foremost, I would like to thank my advisor Dr. Stephen Ralph for inviting me into his research group and supporting me throughout my work. Also, special thanks to Jerrod Langston for helping me set up the wireless experiments and Andy Stark with GTRI for loaning me some of their equipment. Further thanks to Justin Lavrencik, Siddharth Varughese, Alirio Melgar, Antony Varghese, Kathleen Tokuda, and Ricky Causey for their encouragement and support. Finally, I am truly grateful to Dr. John Barry and Dr. Justin Romberg for taking time out of their busy schedules to serve on my committee.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Hyper-wideband communications represent the next frontier in spread spectrum RF systems with an excess of 10 GHz instantaneous bandwidth. In this thesis, an end-to-end physical layer link is implemented featuring 16k-OFDM with a 4 GHz-wide channel centered at 9 GHz and including features such as scrambling, low density parity check (LDPC) error coding, and frequency interleaving. No a priori channel state information is assumed; channel information is derived from the preamble and comb pilot structure. Due to the unique expansive spectral properties, the channel estimator is primarily composed of least squares channel estimates combined with a robust support vector statistical learning approach using autonomously selected parameters. The system's performance is demonstrated through indoor wireless experiments, including line-of-sight and near-line-of-sight links. Moreover, it is shown that the support vector approach performs superior to linear and cubic spline inter/extrapolation of the least squares channel estimates.

CHAPTER 1

INTRODUCTION

Since the onset of broadband Internet, perpetual demand for higher data rates has driven new infrastructure designs to handle greater capacities. As a result, wireless protocols have experienced an expansion of their spectral footprints using more advanced modulation schemes while tackling adverse channel effects and RF device constraints with more sophisticated signal processing. For mobile communication, the LTE Advanced specification spearheaded by the 3GPP standard group features an aggregate bandwidth of up to 100 MHz from Release 10 and onwards. Similarly, the IEEE standards 802.11ac-2013 and 802.11ad-2012 for wireless local area networks (WLAN) implement channel bandwidths up to 160 MHz in the 5 GHz band and 2.16 GHz in the 60 GHz band.

Due to congestion, the public and private sector have long contemplated exploring less crowded spectra just outside the crowded 2.4 and 5 GHz ISM bands. Around the early 2000s, ultra-wideband (UWB) communication entered with bandwidths ranging from 500 MHz to an implicit 7.5 GHz within the FCC's defined operating frequency range of 3.1 to 10.6 GHz. However, the FCC also imposed a -41.3 dBm/MHz power emission limit resulting in short ranges and low data rates for compliant standards, such as IEEE 802.15.4 for wireless personal area networks (WPAN).

Starting in mid-2014, DARPA and other government organizations envision hyper-wideband communications for an array of applications [1]. Desired systems

include spread spectrum with 10 GHz or more of instantaneous bandwidth while operating below 20 GHz in order to avoid severe atmospheric absorption. As can be seen in the visual comparison shown in Figure 1.1 below, the relative enormity of the hyper-wideband to traditional communication bandwidths enables resistance to jamming, interference, and detection. Naturally, the sizable expansion of instantaneous bandwidth also increases the capacity for considerably higher data rates.

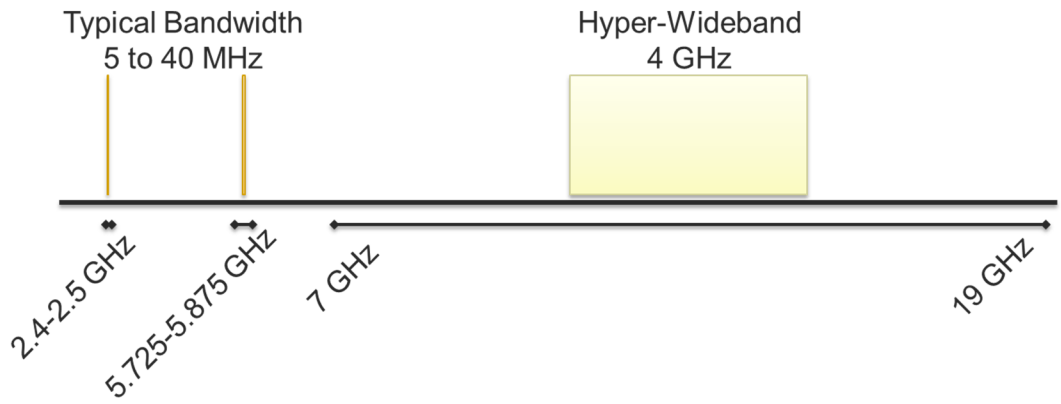

Figure 1.1. Spectral comparison

Ongoing hyper-wideband research efforts include formulating the system architecture, designing wideband electronics for transceivers, and quantifying channel propagation effects. Due to the unique spectral characteristics of hyper-wideband channels, traditional narrow-band channel estimation methods are not appropriate. Consequently, there is a need to accurately and robustly estimate the channel to counter the enhanced channel impairments including multipath and frequency selective fading. This thesis demonstrates an end-to-end wideband wireless link where channel estimation is performed using a robust support vector statistical learning approach. No *a prior* channel state information (CSI) is used and selection of the free parameters is automated based on CSI obtained during the preamble.

Previous work utilizing support vector regression (SVR) for channel estimation in OFDM has been reported for narrowband channels [2]-[4]. In differentiation from past efforts, the work in this thesis improves the method by adding a moving average component and automating the free parameters while making no assumptions of the channel. Moreover, the performance is evaluated with experimental link measurements using a 4 GHz-wide channel centered at 9 GHz and utilizing orthogonal frequency-division multiplexing (OFDM) with a preamble and comb pilot structure. Combining three similar channels would aggregate a total of 12 GHz bandwidth, e.g. spanning from 7 to 19 GHz, to qualify for DARPA's previously stated requirements.

Finally, the application of SVR in hyper-wideband channel estimation has not been previously reported. This work demonstrates an enhanced SVR method within a channel bandwidth that is several orders of magnitude larger than previous communication protocols as well as a higher carrier frequency. Furthermore, the robust nature of the proposed channel estimation scheme may be extended to any OFDM application with a comb pilot structure.

In the following chapters, a physical (PHY) layer specification is proposed following orthogonal frequency-division multiplexing (OFDM) modulation with scrambling, forward error correction, and interleaving. Afterwards, a wireless communication system is designed around the various receiver stages, including signal detection, synchronization, channel estimation, and soft error decoding. Due to distinct difficulties related to hyper-wideband spectral characteristics, particular attention shall be focused on a robust support vector statistical learning approach to estimate and equalize the channel effects.

Next, a software defined radio implementing the proposed PHY layer is codified in Matlab. Indoor wireless experiments are conducted under various configurations to evaluate the system's performance. Finally, the results are analyzed and potential future work is discussed.

CHAPTER 2

PHYSICAL LAYER SPECIFICATION

2.1. OFDM Parameters

Given the parameters of the hyper-wideband communication stated previously in the introduction, orthogonal frequency-division multiplexing (OFDM) is selected due to its numerous advantages. First, OFDM exhibits resistance to frequency selective fading and interference due to partitioning the hyper-wideband channel into numerous narrowband flat fading sub-channels combined with employing proper error coding and/or frequency interleaving. Moreover, OFDM maximizes spectral efficiency through the overlapping of the adjacent sub-channels. Furthermore, through the use of fast Fourier transform (FFT) techniques, OFDM can be implemented with high computational efficiency. Lastly, OFDM is less sensitive to timing offsets and intersymbol interference (ISI) by copying a portion at the end of the symbol and inserting it to the front to form the cyclic prefix (CP), which is also known as the guard interval (GI).

When designing the physical layer for hyper-wideband communication, a considerable amount of inspiration is derived from industry OFDM standards IEEE 802.11 a/g/n/ac [5]. However, since 802.11 specifies channel widths ranging from only 20 to 160 MHz, the proposed 4 GHz hyper-wideband channel allows OFDM system designs with overwhelmingly greater capacities leading to superior data rates but also simultaneously placing higher demands on the transceiver. The 802.11 standards support a maximum FFT size of 512, or equivalently 512 subcarriers per OFDM symbol. With an expanded bandwidth allotment by a factor of 25 or more for hyper-wideband, an FFT size

of 16k (16384) is suggested per OFDM symbol, which represent a factor of 32 or more in comparison to the listed IEEE 802.11 standards. Table 2.1 outlines a comparison between IEEE 802.11 a/g/n/ac and the proposed specification.

Table 2.1 Specification Comparison

	IEEE 802.11 a/g/n/ac				**Hyper-wideband**
Bandwidth (MHz)	20	40	80	160	4000
FFT size, N_{FFT}	64	128	256	512	16384
# data subcarriers, N_{SD}	52	108	234	468	14878
# pilot subcarriers, N_{SP}	4	6	8	16	512
# total subcarriers, N_{ST}	56	114	242	484	15390
Subcarrier spacing (kHz), Δ_F	312.5				≈244.14
FFT period (μs), T_{FFT}	3.2				4.096
Guard interval (μs), T_{GI}	0.4 or 0.8				0.256
Symbol duration (μs), T_{GI}	3.6 or 4				4.352
Constellations	2,4,16,64-QAM		2,4,16,64,256-QAM		2,4,16,64-QAM
FEC	BCC or LDPC				LDPC
Data rate (Mbps) @ 16-QAM, $R = 1/2$	28.9	60	130	260	6837.3

When selecting the OFDM parameters, it is important to note that no declaration of optimality is made in regards to the parameters set forth for any particular environment. In deciding the FFT size, the value of 16384 represents a power of 2 that achieves a similar scaling relative to the bandwidth gaps between the hyper-wideband and 802.11 standards. As a precaution, the subcarrier frequency spacing Δ_F of ≈244.14 kHz is checked to satisfy

$$f_m \ll \Delta_F \ll B_C, \tag{2.1}$$

where f_m denotes the maximum Doppler shift and B_C represents the coherence bandwidth constraints, in order to maintain a flat subcarrier channel.

First, it can be shown that the maximum Doppler shift remains much lower than Δ_F regardless of whether the target remains stationary or moving at the velocity of a commercial aircraft across the spectrum of interest. For example, assuming a carrier frequency f_c of 11 GHz and velocity v of 100 km/hr, a maximum Doppler shift f_m of 1.0185 kHz is obtained via

$$f_m = \frac{v f_c}{c}, \tag{2.2}$$

where c represents the speed of light. Therefore, the lower constraint of (2.1) is satisfied and remains true even when adjusting v to 250 km/hr and f_c to 19 GHz.

For the upper constraint of (2.1), multiple indoor empirical measurements performed at 11 GHz in various configurations obtain 90% coherence bandwidths at 19, 36, and 60 MHz [6]. Other separate indoor results at 17 GHz establish coherence bandwidths around 2.41 MHz [7]. All coherence bandwidth findings remain much greater than the subcarrier frequency spacing. On the other hand, a single carrier modulation with 4 GHz bandwidth would fail the coherence bandwidth upper constraint and result in significant frequency-selective fading.

Finally, all the IEEE 802.11 OFDM protocols support BPSK, QPSK, 16-QAM, and 64-QAM constellations with gray code mapping [5] as seen in Figure 2.1 and are investigated for hyper-wideband communications. In order to maintain the same average power, a normalization factor is introduced in increasing order as follows: 1, $1/\sqrt{2}$, $1/\sqrt{10}$, and $1/\sqrt{42}$. Although the latest IEEE 802.11 ac revision also supports 256-QAM, this particular constellation size is excluded from this thesis's scope of work for practical reasons, such as the effective number of bits (ENOB) available to contemporary D/A and A/D converters.

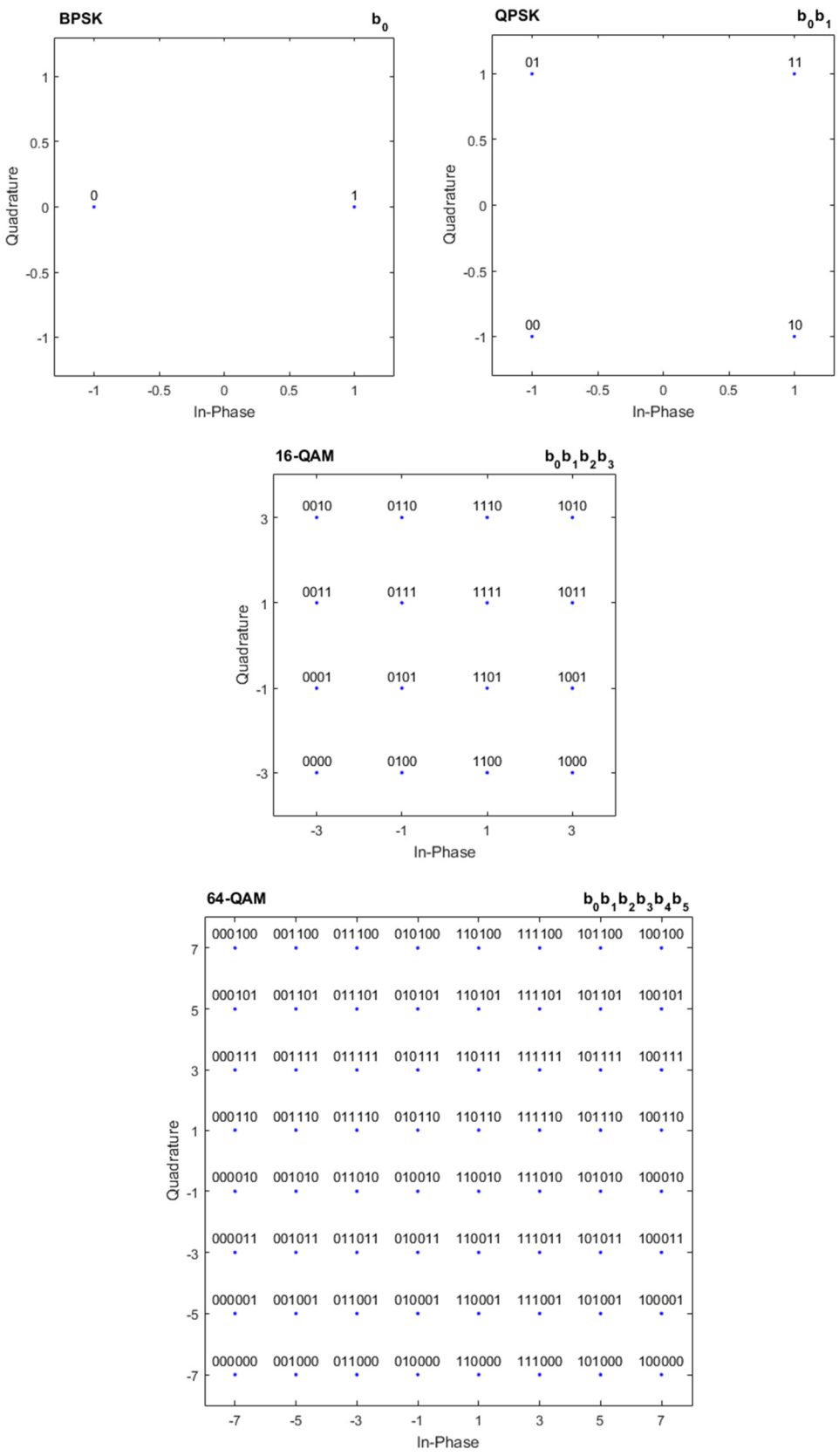

Figure 2.1. BPSK, QPSK, 16-QAM, and 64-QAM constellations

2.2. Nulls and Pilots Structure

As shown previously in Table 2.1, not all possible subcarriers are utilized. For example, in the 160 MHz specification for 802.11ac, 28 of the 512 subcarriers represent null subcarriers. Most of these null subcarriers are designated at the upper and lower frequency boundaries of the carrier to serve as guard bands against interference from adjacent channels as well as allowing roll-off for anti-aliasing filters. Additionally, there exists one or more center (DC) null subcarrier to remedy potential spurious responses in zero-IF receiver designs.

Furthermore, pilot tones occupy a selection of the utilized subcarriers for channel estimation at the receiver. In the previous example, only the remaining 468 of the original 512 subcarriers carry the data payload. The ratio of pilot to data subcarriers presents a tradeoff between obtaining greater channel information for equalization with more pilot tones and their overhead that diminishes the data throughput. For the purpose of this work, the hyper-wideband protocol shall maintain the roughly 29-to-1 data-pilot ratio implemented by the IEEE 802.11 standards for 80 and 160 MHz bandwidths.

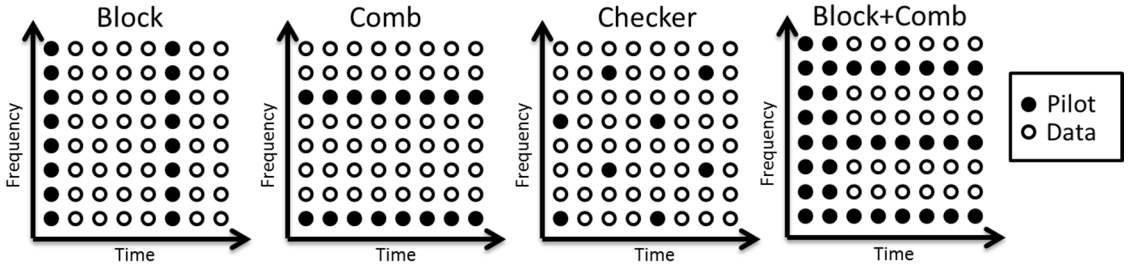

Figure 2.2. Pilot structures (not drawn to scale)

After determining the pilot-to-data ratio, there exist various arrangements to disperse the pilot tones amongst the data subcarriers, such as the block, comb, and checkered structures shown in Figure 2.2. Each structure compromises between tracking the channel with greater frequency or time precision. While the 802.11 OFDM standards

follow a combo pilot structure [5], the initial training sequences in the preamble can be considered as a couple blocks of pilot tones.

Due to OFDM's innate sensitivity to carrier frequency offset, the proposed hyper-wideband protocol shall also follow the 802.11's format of comb structure with the initial block pilot tones. For simplicity, the 512 pilot tones shall be distributed uniformly in frequency with every 30th subcarrier starting from index -7680 to 7680. The data subcarriers will fill in the gaps from subcarrier index -7709 to 7709 with the exception of 29 null subcarriers in the center from index -14 to 14. Figure 2.3 illustrates portions of the null, pilot, and data subcarrier arrangement when centered at DC on the frequency axis.

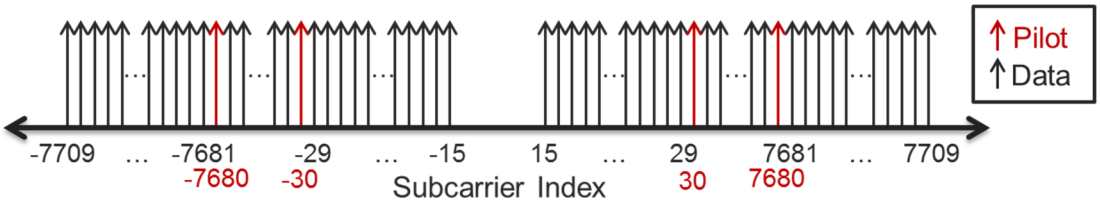

Figure 2.3. Subcarrier arrangement

2.3. Packet Framework

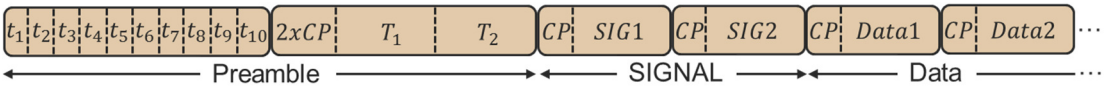

Figure 2.4. IEEE 802.11 packet framework (not drawn to scale)

In general, the packet framework at the physical layer for the hyper-wideband communications follows that of the IEEE 802.11 OFDM standards. As a brief review of the 802.11 packet structure, each packet consists of a preamble, SIGNAL symbols, and the data payload as shown in Figure 2.4. First, the preamble consists of a short and long training sequence known to the receiver. In the first part, the short training sequence comprises of 10 smaller repeating sequences each of length 16. Afterwards, the long

training sequence contains two repeating OFDM symbols with BPSK modulation, although they can be viewed as one large symbol with double the cyclic prefix duration [5].

Following the preamble, the SIGNAL field contains one or more OFDM symbols transmitted with a known modulation coding scheme that characterizes the upcoming data portion of the packet. Although its composition varies depending on the particular 802.11 specification, the SIGNAL field conveys the necessary signaling information such as packet length, constellation size, and forward error correction (FEC) coding scheme. Additionally, amendments n and ac add an 8-bit cyclic redundancy check (CRC) to ensure integrity of the received bits [5].

Next, the remaining OFDM symbols carry the data payload in the manner described by the SIGNAL field. Prior to the start of the intended data bits, the protocol inserts 16 bits representing the SERVICE field. The SERVICE field composes of 7 zero bits for initializing the scrambler based on a chosen seed and 9 reserved bits. Finally, the data bit sequence is zero-padded at the end to complete the last OFDM symbol [5].

In the hyper-wideband communications, the packet framework duplicates much of the IEEE 802.11's framework. For the preamble's short training sequence, the same 160 complex values initiate the hyper-wideband communication, albeit with a much shorter duration due to the higher signaling rate. Again, two OFDM symbols T_1 and T_2 are utilized in the long training sequence, although without the merging of both symbols' cyclic prefixes.

However, since the number of bits required to convey IEEE 802.11's SIGNAL fields represent an miniscule fraction of one full hyper-wideband OFDM symbol, the

signaling information is instead condensed into a small portion of T_2 while eliminating the need for additional SIGNAL symbols. Therefore, T_1 and T_2 are known to the receiver except for the allocation for signaling information in T_2, although the 8-bit CRC remains present for verifying these unknown bits. For the rest of the packet structure, no further changes are made and the new modified framework for hyper-wideband communications is depicted in Figure 2.5.

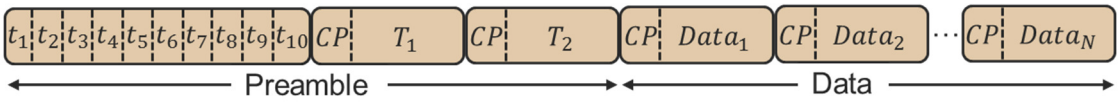

Figure 2.5. Modified hyper-wideband packet framework (not drawn to scale)

2.4. Scrambling

In comparison to single carrier modulations, one of the disadvantages of OFDM is its potential for high peak-to-average power ratio (PAPR). With OFDM, multiple subcarriers may transmit at the highest power, i.e. the constellation point with the greatest energy, during the same symbol period versus other times when transmitting at the lowest power. After adding the subcarriers during each instance, the result is a PAPR substantially higher than a single carrier's average. Consequently, modulation schemes with high PAPR require more expensive and less efficient analog circuits, such as power amplifiers, with large linear range.

In order to reduce the PAPR for OFDM, one common solution is to whiten the transmitted data bits through scrambling and thereby break any long chains of zeros or ones. IEEE 802.11 utilizes the following scrambler with generator polynomial

$$g(x) = 1 + x^4 + x^7 \tag{2.3}$$

to generate a repeating 127-bit sequence whose starting point depends on the initial seed utilized by the transmitter and can be derived by the receiver through the SERVICE field [5]. As shown in Figure 2.6, the incoming data bits are summed mod-2 with this whitening sequence to produce the scrambled output bits. The proposed hyper-wideband protocol borrows the same scrambler for its implementation.

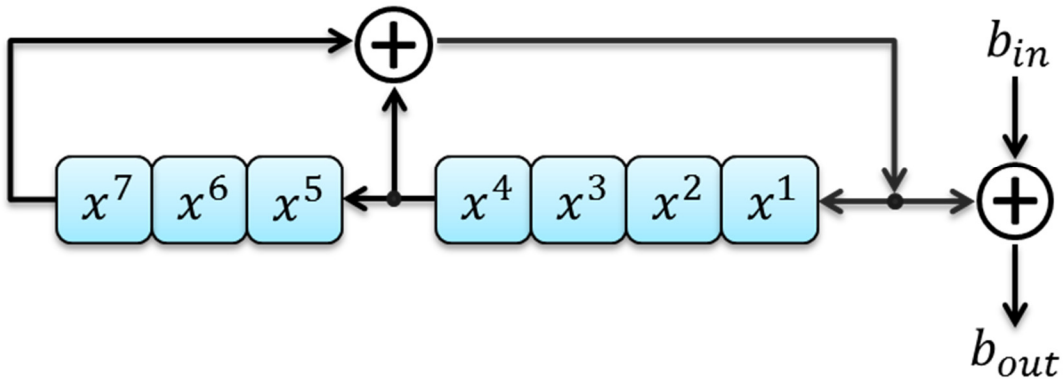

Figure 2.6. IEEE 802.11 OFDM scrambler

2.5. Forward Error Correction (FEC)

As one of the essential components in wireless communications, forward error correction enables the transmitter to encode the data payload with some form of redundancy in order for the receiver to reliably recover the message even with a limited number of errors. The two main categories of FEC include convolutional codes and block codes. Convolutional coding involves sliding polynomial(s) across the data bits to generate the coded bits. On the other hand, block coding encodes the bits into fixed-size blocks with added bits at the end of each block. Some protocols combine both by utilizing an outer block code followed by an inner convolutional code.

In the initial IEEE 802.11 a/g versions, the standard employs only binary convolutional coding (BCC), which originates largely from work performed by Viterbi in

the 60s [8]. This specific BCC implementation utilizes the following two 6th order polynomial polynomials

$$g_A(x) = 1 + x^2 + x^3 + x^5 + x^6 \tag{2.4}$$

$$g_B(x) = 1 + x + x^2 + x^3 + x^6 \tag{2.5}$$

with corresponding diagram shown in Figure 2.7. It utilizes a minimum coding rate of ½, i.e. two coded bits for every one data bit, although higher rates are supported through puncturing, i.e. selective removal of coded bits [5]. However, this thesis work focuses exclusively on the base encoding rate of ½.

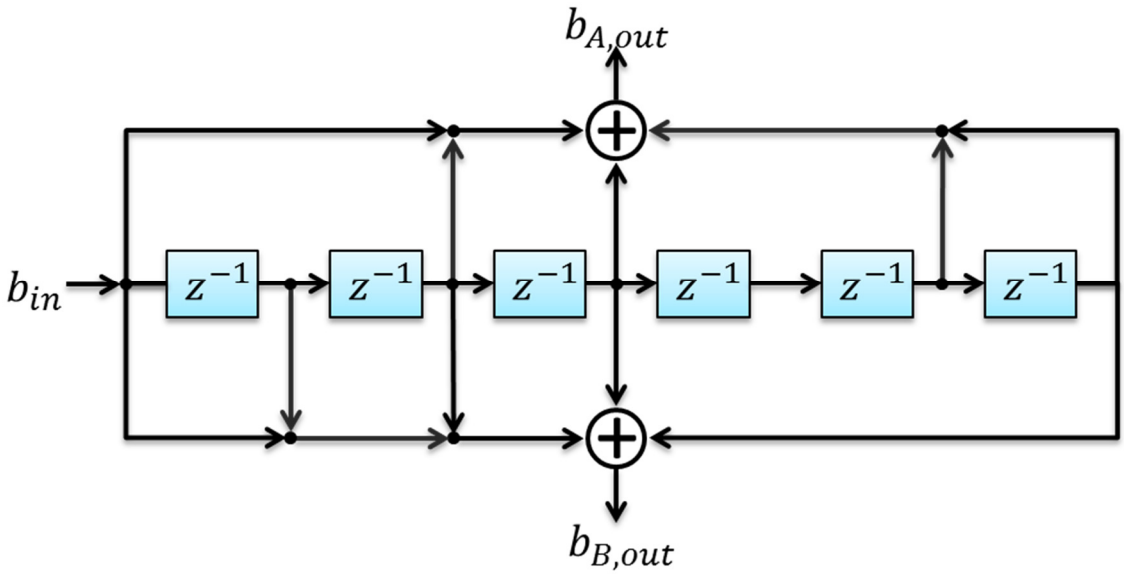

Figure 2.7. IEEE 802.11 OFDM convolutional encoder

In later amendments n and ac, the protocol allows the user to choose low density parity check (LDPC) as an alternative FEC option. Discovered by Gallager in the 60s, LDPC is a sparse linear block code approaching Shannon's capacity [9]. However, it has remained largely unused until the 2000s due to past computational hardware limitations. In the linear block code, a k x n generator matrix G generates an n-length codeword $\boldsymbol{c}$ from the k-length information word $\boldsymbol{u}$ as follows:

$$\boldsymbol{c} = \boldsymbol{u}G. \tag{2.6}$$

On the flip side, the associated $(n - k)$ x n parity check matrix H calculates the syndrome $\boldsymbol{s}$ from the received word $\boldsymbol{r}$ as follows:

$$\boldsymbol{s} = H\boldsymbol{r}^T. \tag{2.7}$$

If no errors occurred during transmission, $\boldsymbol{s}$ evaluates to an all zero vector and the first k bits of $\boldsymbol{r}$ represent the decoded message.

For the proposed hyper-wideband specification, the FEC utilizes LDPC with the parity check matrix H from the 802.11ac standard corresponding to codeword length n of 1944 and message length k of 972 for a code rate of ½ [5]. Using a subblock size Z of 81 x 81 bits, the above H is defined in Table 2.2, where an entry of -1 represents all zeros and the non-negative entries correspond to the permutation order of the identity matrix. Further discussion on decoding algorithms for both LDPC and BCC can be found in section 3.6.

Table 2.2 Parity Check Matrix

57	-1	-1	-1	50	-1	11	-1	50	-1	79	-1	1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
3	-1	28	-1	0	-1	-1	-1	55	7	-1	-1	-1	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1
30	-1	-1	-1	24	37	-1	-1	56	14	-1	-1	-1	-1	0	0	-1	-1	-1	-1	-1	-1	-1	-1
62	53	-1	-1	53	-1	-1	3	35	-1	-1	-1	-1	-1	-1	0	0	-1	-1	-1	-1	-1	-1	-1
40	-1	-1	20	66	-1	-1	22	28	-1	-1	-1	-1	-1	-1	-1	0	0	-1	-1	-1	-1	-1	-1
0	-1	-1	-1	8	-1	42	-1	50	-1	-1	8	-1	-1	-1	-1	-1	0	0	-1	-1	-1	-1	-1
69	79	79	-1	-1	-1	56	-1	52	-1	-1	-1	0	-1	-1	-1	-1	-1	0	0	-1	-1	-1	-1
65	-1	-1	-1	38	57	-1	-1	72	-1	27	-1	-1	-1	-1	-1	-1	-1	-1	0	0	-1	-1	-1
64	-1	-1	-1	14	52	-1	-1	30	-1	-1	32	-1	-1	-1	-1	-1	-1	-1	-1	0	0	-1	-1
-1	45	-1	70	0	-1	-1	-1	77	9	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	-1
2	56	-1	57	35	-1	-1	-1	-1	-1	12	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0
24	-1	61	-1	60	-1	-1	27	51	-1	-1	16	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0

2.6. Frequency Interleaving

Forward error correcting codes fail when a certain number of bit errors occur in a given length. However, the decoder may still recover the message if a high concentration of error bits are instead dispersed across the payload through interleaving. For OFDM systems, frequency-selective interference can inflict a burst of errors by affecting a contiguous subset of the subcarriers. However, frequency interleaving ensures that the errors bits are distributed rather than clumped together and therefore the bits are decoded correctly. Naturally, the performance gain of interleaving necessitates an existing FEC and renders no benefits by itself.

When designing a frequency interleaver for an OFDM system, the data bits can be visualized as a 2D matrix having N_{ROW} and N_{COL} with each entry corresponding to a frequency-time location. The interleaver then follows one or more permutation(s) similar to the transpose of the matrix so as to rearrange the ordering of the bits. For example, the interleaving algorithm for BCC in the IEEE 802.11 standard for a single spatial stream can be presented by the two permutations

$$i = N_{ROW}\left(k \bmod N_{COL}\right) + \left\lfloor \frac{k}{N_{COL}} \right\rfloor; \;\; k = 0,1, \dots, N_{CBPS} - 1 \tag{2.8}$$

$$j = s\left\lfloor \frac{i}{s} \right\rfloor + \left(i + N_{CBPS} - \left\lfloor \frac{N_{COL} * i}{N_{CBPS}} \right\rfloor\right) mod\ s; \;\; s = \max\left(\frac{N_{BPSC}}{2}, \, 1\right), \tag{2.9}$$

where N_{CBPS} denotes the number of coded bits per symbol and N_{BPSC} represents the number of bits per subcarrier [5].

Similarly, the LDPC option in 802.11 ac utilizes tone mapping permutation

$$t = d_{TM}\left(k \bmod \frac{N_{SD}}{d_{TM}}\right) + \left\lfloor \frac{k * d_{TM}}{N_{SD}} \right\rfloor; \;\; k = 0,1, \dots, N_{SD} - 1 \tag{2.10}$$

to interleave the bits, where d_{TM} denotes the distance parameter and N_{SD} represents the number of data subcarriers per OFDM symbol [5]. Additionally, the problem can be realized by the matrix interpretation using the following definitions:

$$N_{ROW} = d_{TM} * N_{BPSC} \tag{2.11}$$

$$N_{COL} = \frac{N_{SD}}{d_{TM}}. \tag{2.12}$$

However, unlike 802.11ac, d_{TM} is chosen to be 86 for the hyper-wideband protocol in order to preserve the ability for each codeword to cover the full range of frequencies. The basis for the value 86 stems from choosing an integer factor of the number of data subcarriers N_{SD} that satisfies condition

$$d_{TM} \geq \left\lceil \frac{N_{CBPS}}{N_{LDPC}} \right\rceil. \tag{2.13}$$

CHAPTER 3

SYSTEM DESIGN

3.1. Overview

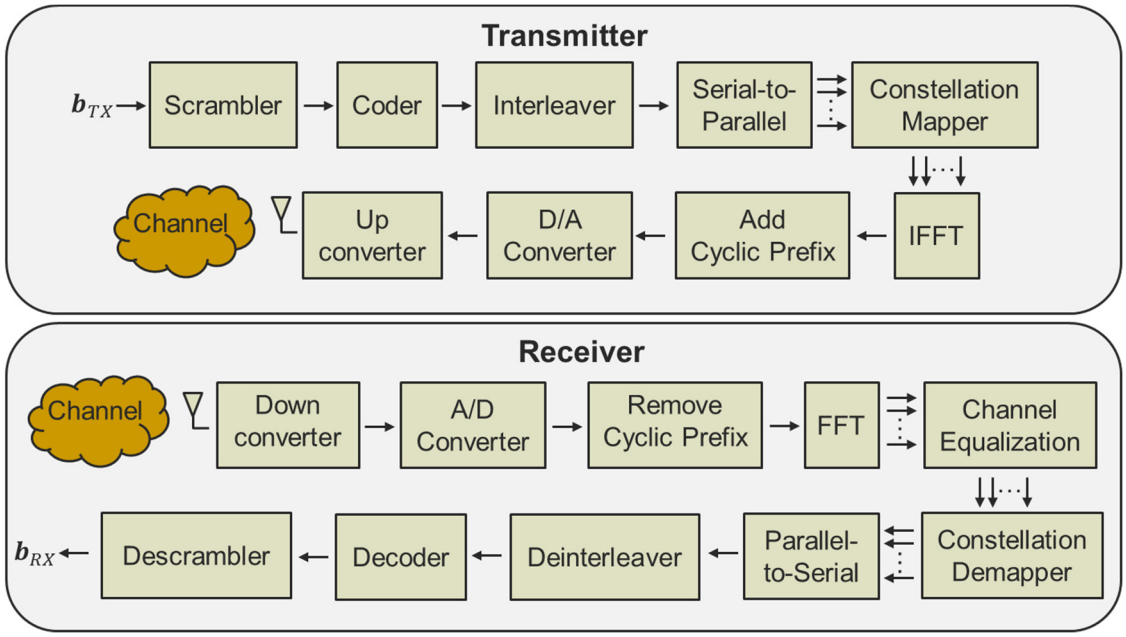

Figure 3.1. OFDM TX/RX block diagram

Overall, the basic building blocks for the hyper-wideband system remain largely identical to a typical OFDM telecommunication design illustrated in Figure 3.1. At the transmitter, the information bits pass through a scrambler, forward error correction coder, and a frequency interleaver before mapped to QAM signals on orthogonal subcarriers. After conversion from digital to analog at the front end, the baseband signal is upconverted to radio frequency (RF) and broadcasted out the antenna over the air to the receiver.

On the other end, the receiver requires greater complexity than the simple reversal suggested by the block diagram. Since the receiver does not know when a transmission

occurs, the receiver must detect the start of a valid signal and synchronize in time and frequency. Furthermore, due to signal degradation, the receiver also needs to perform channel estimation and equalization along with any necessary error correction.

3.2. Signal Detection and Timing Synchronization

With *a priori* knowledge of the transmitted preamble, a typical signal detection method involves a matched filter utilizing a window snapshot of the received signal $\boldsymbol{w}$ with the expected preamble $\boldsymbol{p}$ [10]. In order to compensate for power fluctuations in the spectral environment, both $\boldsymbol{w}$ and $\boldsymbol{p}$ are normalized to unit energy before calculating a fair correlation metric r_n between zero and one for the received samples starting at time n as formulated:

$$r_n = \left| \frac{\boldsymbol{w}^T \boldsymbol{p}^*}{\|\boldsymbol{w}\| \|\boldsymbol{p}\|} \right|. \tag{3.1}$$

Afterwards, hypothesis testing involving a threshold t selects between the null hypothesis of detecting a valid signal versus the alternative hypothesis of no valid signal

$$\begin{aligned} H_0 &: r_n \geq t \\ H_a &: r_n < t, \end{aligned} \tag{3.2}$$

where t represents some threshold. The threshold can be a fixed constant or a delayed variable calculated through statistical measurement of recent samples $\boldsymbol{x}$ as formulated:

$$\begin{aligned} t &= Q_3 + \alpha * IQR(\boldsymbol{x}) \\ IQR &= Q_3 - Q_1 \end{aligned} \tag{3.3}$$

In (3.3), Q_1 and Q_3 represent the first and third quartiles and parameter α denotes a positive multiplicative factor, e.g. 3 or 4.

Alternatively, by taking advantage of the ten repetitions within the preamble's short training sequence, a second method for calculating r_n employs a double sliding window to perform cross correlation between neighboring intervals $\boldsymbol{w_1}$ and $\boldsymbol{w_2}$ with individual length equal to the repeating sequences. Again, in order to eliminate potential power variations influencing the result, both $\boldsymbol{w_1}$ and $\boldsymbol{w_2}$ are normalized before calculating the metric, resulting in the following expression:

$$r_n = \frac{\left|\boldsymbol{w_1^T w_2^*}\right|}{\|\boldsymbol{w_1}\|\|\boldsymbol{w_2}\|}. \tag{3.4}$$

This method offers the benefit of fewer computations while also including information of the channel effects [9] assuming that the channel is relatively flat over the span of the preamble. Additionally, the double sliding window technique can trade between more computational cycles in exchange for higher decision confidence by determining the number of repeating sequences to incorporate for a single window. However, it performs extremely poorly in low signal-to-noise ratio (SNR) conditions.

Meanwhile, the receiver can also achieve timing synchronization simultaneously upon detecting a valid signal by picking the largest correlation peak r_{max} during a local duration. Depending on the matched filter length or the double sliding window width, it is important to keep track of the number of peaks or the peak hold duration in order to accurately pinpoint the starting time [9]. As a safeguard, the OFDM's guard interval provides some room for error by cyclically extending the FFT duration in order to tolerate slight timing offsets as long as they are less than the guard interval.

In order to demonstrate the two methods, simulations are conducted composing of the preamble mixed with additive white Gaussian noise (AWGN) at 10 dB and 0 dB SNR. For the matched filter, the signal is correlated against 1 to 10 of the repeating

training sequences (TS), and the double sliding window widths range from 4 to 20 ns. Correlation plots of each run at the two different SNRs are depicted in Figures 3.2 and 3.3 with a red line indicating the statistical threshold calculated at $\alpha = 3$.

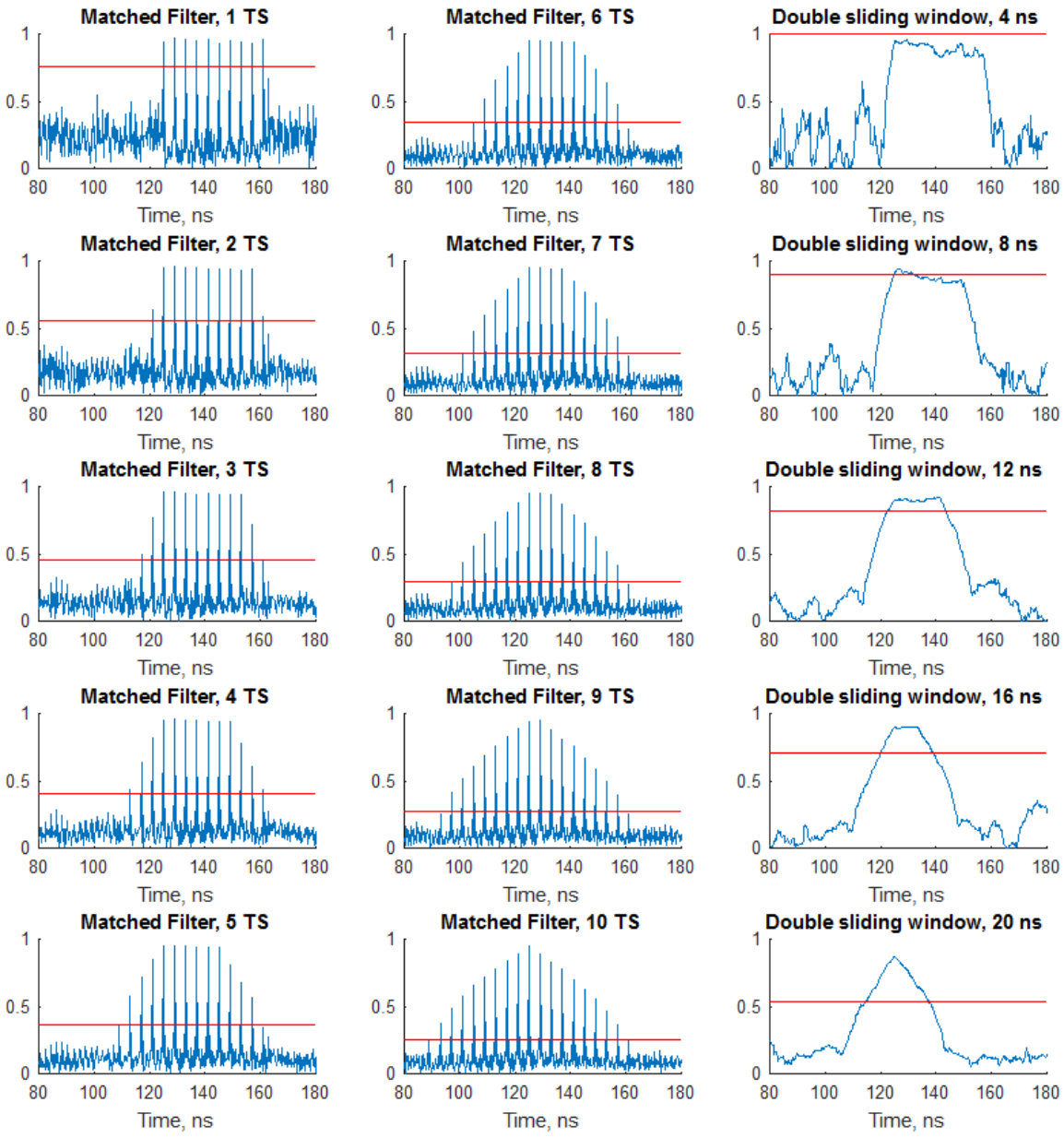

Figure 3.2. Correlation plots at 10 dB SNR

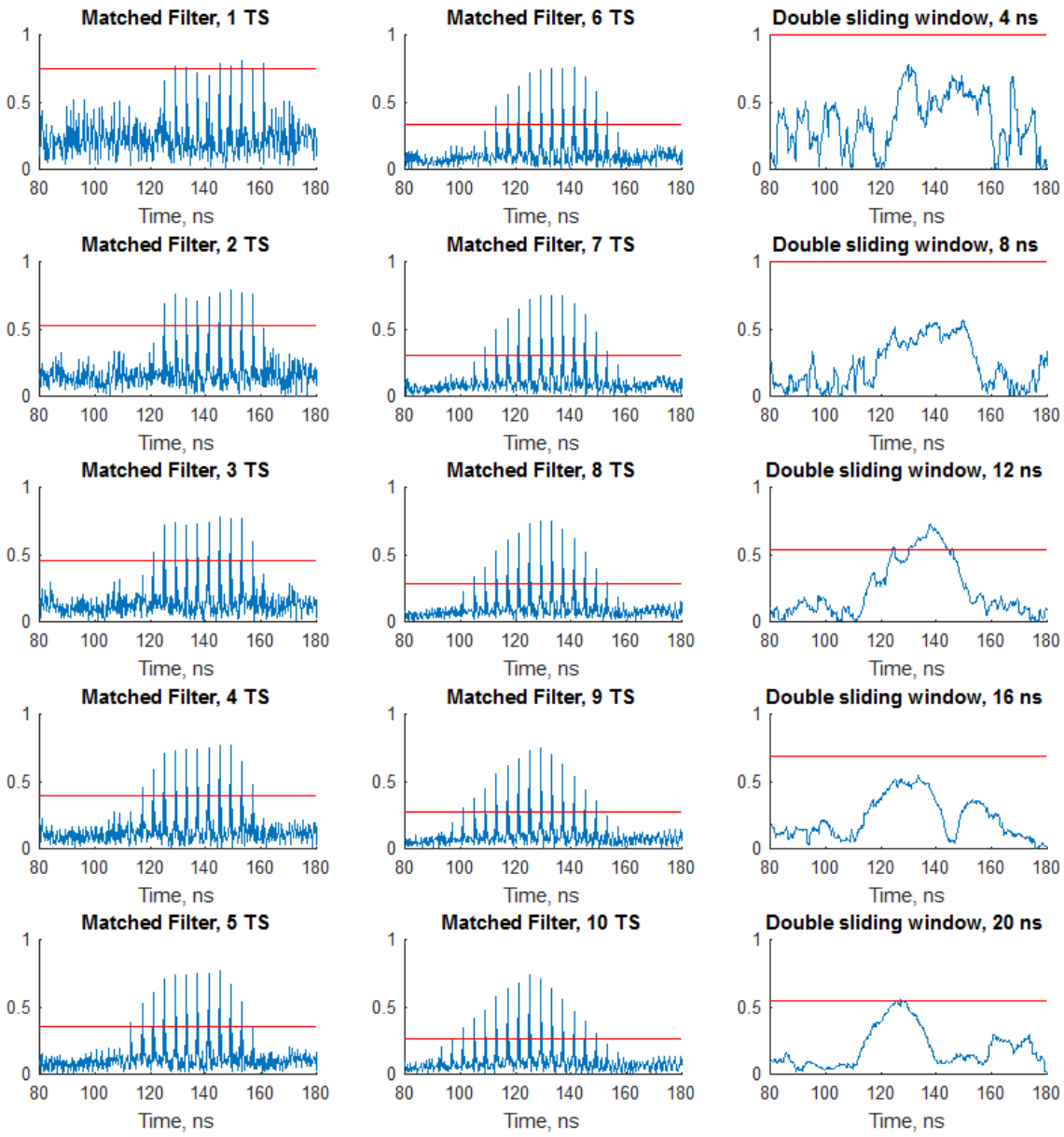

Figure 3.3. Correlation plots at 0 dB SNR

At the higher SNR of 10 dB, all the simulations perform well with definitive spikes and peaks, although the threshold factor α may require minor adjustments. Under lower SNR conditions at 0 dB, the majority of the matched filters' performances remain relatively steady. However, the double sliding window peaks become less clearly defined at the higher widths and mostly lost at the lower widths. To ensure optimal performance,

the hyper-wideband's signal detection and time synchronization employs the matched filter with all 10 repeating sequences.

3.3. Frequency Synchronization

Oftentimes, the transmitter and receiver may not be finely tuned to the same carrier frequency or sampling clock frequency. Combined with the fading effects of the channel, this frequency offset can potentially eliminate the orthogonality in OFDM and cause intersymbol interference (ISI). On the other hand, the frequency offset can usually be neglected if it is very small compared to the subcarrier spacing.

Similar to the double sliding method in (3.4), the two OFDM symbols in the preamble's long training sequence can be utilized to estimate the frequency offset by assigning $\boldsymbol{w_1}$ to the first symbol and $\boldsymbol{w_2}$ to the second. First, the phases pertaining to the signaling information subcarriers for $\boldsymbol{w_2}$ may need to be rotated by π in order to match the respective fixed values of $\boldsymbol{w_1}$. Then, instead of taking the sum of the normalized magnitudes as in (3.4), the mean of the angles is calculated by

$$\boldsymbol{\theta} = \frac{\angle(\boldsymbol{w_2} \cdot \boldsymbol{w_1^*})}{\mathrm{N_{SYM}}}$$
$$f_o = E_{trunc}[\boldsymbol{\theta}], \tag{3.5}$$

where N_{SYM} denotes the number of samples per OFDM symbol [9].

In order to ensure a more accurate measurement, the notation E_{trunc} signifies that outlier spikes in the angle correlations are truncated prior to taking the mean. Afterwards, hypothesis testing can be conducted to test the null hypothesis that a specific angle is an outlier versus the alternative hypothesis of not being an outlier as follows:

$$H_0: \theta_n \in (-\infty, t_l) \cup (t_u, \infty)$$
$$H_a: \theta_n \in [t_l, t_u] \tag{3.6}$$

Similar to the statistical algorithm employed in (3.3), t_l and t_u represent lower and upper thresholds determined by

$$t_l = Q_1 - \alpha * IQR(\boldsymbol{\theta})$$
$$t_u = Q_3 + \alpha * IQR(\boldsymbol{\theta}) \tag{3.7}$$
$$IQR = Q_3 - Q_1.$$

As an example, Figure 3.4 plots the phase offsets with the dashed lines representing t_l and t_u with $\alpha = 1.5$ and the red line indicating an average of 348.7 Hz as the frequency offset.

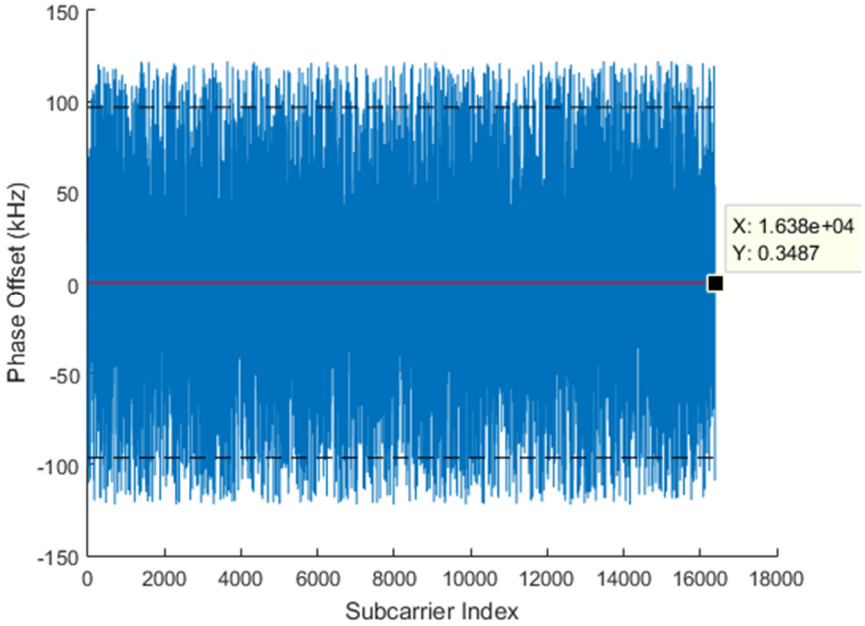

Figure 3.4. Phase offset plot

Ultimately, the benefit of this frequency offset estimation is questionable since it can only estimate up to approximately ±114.9 kHz before rolling over, which is calculated by

$$f_{o,max} = \frac{1}{2\delta t}, \tag{3.8}$$

where T_{SYM} denotes the repeating interval δt in this particular scenario. Other intervals, such as the repeating short training sequences, can be utilized to estimate higher frequency offsets. However, the resulting estimate's accuracy decreases when utilizing the short training sequence due to the shorter spacing between recurring samples. Moreover, fast fading, intersymbol interference (ISI), and timing offsets can distort any of the frequency offset estimates. On the positive side, channel equalization can correct phase offsets on a symbol by symbol basis as detailed in section 3.4.

3.4. Channel Estimation and Equalization

During signal propagation from transmitter to receiver, the radio wave experiences power decay and channel fading. Examples include multipath and scattering due to obstacles and interference from external emitters. Inadequate or mismatched RF components may also contribute to signal distortion. Moreover, due to the expansive 4 GHz hyper-wideband carrier, there exists high likelihood of frequency selective fading. Therefore, the receiver performs channel estimation and equalization in order to promote a reliable communication link.

Of the various channel estimation algorithms, the least squares (LS) estimator represents a simple but effective estimator. Using the pilot tones as references, the channel estimate $\widehat{H}_{LS}$ is obtained by

$$\widehat{H}_{LS} = X^{-1}Y \tag{3.6}$$

where X represents the known signal and Y denotes the received signal. Therefore, for each subcarrier k, (3.6) can be written as

$$\widehat{H}_k = \frac{Y_k}{X_k}. \tag{3.7}$$

This method requires no *a priori* channel state information (CSI), i.e. no assumptions about the channel are made [11].

However, with the exception of the preamble, the modulation scheme follows a comb structure with pilots only on select subcarriers throughout the duration of the packet as opposed to a block structure where pilots are assigned across all subcarriers at certain instances of time. As a result, some form of interpolation across frequency is necessary to acquire the channel estimates for the remaining subcarriers. Traditional techniques include linear interpolation or cubic spline interpolation [10], although a third method utilizing support vector regression (SVR) is explored later.

Another channel estimator relies on decision feedback to continually update original channel estimates (perhaps using LS) in a weighted fashion. In addition, the decision feedback only demands a modest increase in computational resources and does not require CSI [10]. However, the decision feedback estimator performs poorly in general since any error can quickly compound beyond recovery of the protocol's error correction scheme. On the other hand, the decision feedback estimator works extremely well when there exists reliable verification of symbols or bits received midway through the packet, such as a cyclic redundancy check (CRC) that can verify the bits received are correct.

In addition to the LS and decision feedback estimators, the minimum mean square error (MMSE) estimator takes a Bayesian approach to estimating the channel. As its name suggests, the MMSE finds the solution $\widehat{H}_{MMSE}$ that minimizes the mean square error through

$$\widehat{H}_{MMSE} = R_{HH}(R_{HH} + \sigma^2(XX^H)^{-1})^{-1}X^{-1}Y$$
$$R_{HH} = E[HH^H], \tag{3.8}$$

where R_{HH} represents the channel's autocovariance matrix and σ^2 denotes the noise variance [10]. Theoretically, the MMSE estimator outperforms LS and decision feedback estimator but its performance depends on accurate CSI. Although the long training sequences in the preamble at the beginning of each packet supplies a couple of known block data in which to accurately estimate the 2nd-order channel statistics, the information gleaned from those two OFDM symbols quickly turns obsolete in the ensuing data symbols for any fast fading environment. Therefore, the MMSE estimator is ruled out for this thesis work due to its prerequisites.

Ultimately, the hyper-wideband receiver employs a variant of LS and SVR with moving average (MA) and limited decision feedback. Starting with the preamble's known training symbol T_1, a baseline channel estimate for all subcarriers is obtained using LS. Similarly, for the remaining known signals, including most of T_2 and the pilot tones, the LS estimator obtains the precise channel for that particular time and subcarrier. For the signaling bits in T_2, decision feedback is performed for the corresponding subcarriers conditioned on the CRC passing and resulting in an updated baseline channel estimate. With the remaining OFDM data symbols, a moving average of the pilot tones in frequency act as reference points upon which a robust support vector regression learning algorithm adjusts the channel estimate baseline for equalization. A more in-depth discussion about the support vector regression algorithm is presented in the following section.

3.5. Support Vector Regression

In the 90s, Vapnik and others developed the support vector machine (SVM) at AT&T Bell Labs as a supervised learning algorithm for classification [12] and later extended the model to include regression [13]. Since then, various fields, such as biophysics [14] and finance [15], have adapted the methodology to fit their applications. In telecommunications, several similar papers [2]-[4] as early as 2006 have been written pertaining to support vector regression (SVR) for OFDM channel estimation.

To begin framing the SVR problem, let the regression model be expressed as

$$h_k = \langle \boldsymbol{w}, \phi(s_k) \rangle + b + e_k, \tag{3.9}$$

for $k = 1, \ldots, N_{ST}$, where h_k defines the complex channel estimates, $\boldsymbol{w}$ represents an m-dimensional complex weight vector, $\phi(\cdot)$ characterizes a nonlinear feature mapping function from $\mathbb{R}$ to $\mathbb{R}^m$, s_k denotes the non-null subcarrier frequencies normalized to $[0,1]$, b expresses the complex bias term, e_k represents the complex residuals, and N_{ST} denotes the total number of non-null subcarriers. Similar to previous work [2]-[4], a training data set is acquired from the pilot subcarrier frequencies and their complex channel estimates. However, the performance can be notably improved by utilizing a moving average of length 5 with two neighboring symbols ahead and behind in time (if available) in order smooth out short-term fluctuations.

As illustrated in Figure 3.5, a hyperplane is constructed with a soft margin defined by a subset of the training data, called the support vectors, shown with solid black fill. More precisely, the margin can be defined as

$$\frac{1}{2}\frac{\langle \boldsymbol{w}, \boldsymbol{x}_+ - \boldsymbol{x}_- \rangle}{\|\boldsymbol{w}\|} = \frac{\varepsilon}{\|\boldsymbol{w}\|}, \tag{3.10}$$

where $\boldsymbol{x}_+$ and $\boldsymbol{x}_-$ represent a pair of the farthest support vectors when projected onto $\boldsymbol{w}$, i.e. orthogonal to the hyperplane. Complex slack variables ξ and ξ' for above and below the hyperplane are introduced with real and imaginary components, e.g. ξ_R and ξ_I, in order to ease constraints established later.

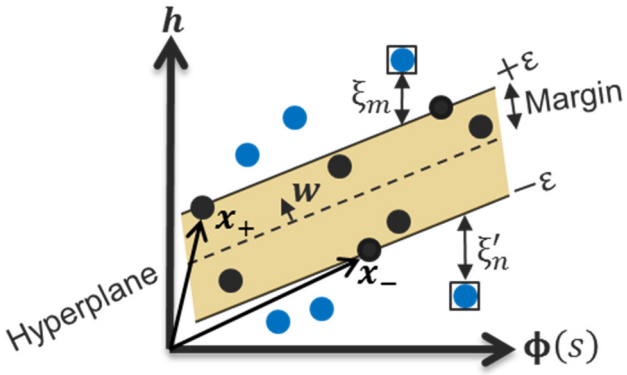

Figure 3.5. Hyperplane with soft margin defined by support vectors.

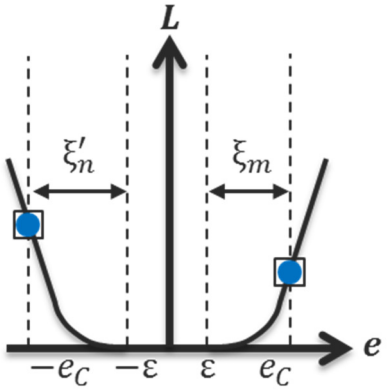

Figure 3.6. Modified Huber loss curve with two sample data points.

Additionally, a loss is prescribed to the residuals with the following modified Huber loss function

$$L(e_k) = \begin{cases} 0, & |e_k| \le \varepsilon \\ \dfrac{1}{2\delta}(|e_k| - \varepsilon)^2, & \varepsilon \le |e_k| \le e_C, \\ C(|e| - \varepsilon) - \dfrac{\delta C^2}{2}, & |e_k| \ge e_C \end{cases} \tag{3.11}$$

where $e_C = \varepsilon + \delta C$ and a model curve illustrated in Figure 3.6. Notably, the loss function consists of an ε-insensitive, quadratic, and linear region designed to generate a smooth curve starting from a flat zero and ramping up to a constant gradient. Moreover, the regions' intervals are adjustable by the free parameters ε, δ, and C [16].

In short, the objective of SVR is to maximize the margin while minimizing the overall loss. For notation, K_1 and K_3 designates the real and imaginary quadratic regions, respectively, and K_2 and K_4 designates the real and imaginary linear regions, respectively. Then, the primal problem can be formulated as minimizing $\mathcal{L}_p$ with respect to $\boldsymbol{w}$, b, ξ, and ξ′ where

$$\begin{aligned}\mathcal{L}_p = \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{1}{2\delta}\sum_{k\epsilon K_1}\left(\xi_{R,k}^2 + \xi_{R,k}'^2\right) + C\sum_{k\epsilon K_2}\left(\xi_{R,k} + \xi_{R,k}'\right) \\ + \frac{1}{2\delta}\sum_{k\epsilon K_3}\left(\xi_{I,k}^2 + \xi_{I,k}'^2\right) + C\sum_{k\epsilon K_4}\left(\xi_{I,k} + \xi_{I,k}'\right) - \sum_{k\epsilon K_{2,}K_4}\frac{\delta C^2}{2}\end{aligned} \tag{3.12}$$

and subject to constraints

$$\begin{cases}\Re\{h_k - \langle \boldsymbol{w}, \phi(s_k)\rangle - b\} \le \varepsilon + \xi_{R,k} \\ \Im\{h_k - \langle \boldsymbol{w}, \phi(s_k)\rangle - b\} \le \varepsilon + \xi_{I,k} \\ \Re\{-h_k + \langle \boldsymbol{w}, \phi(s_k)\rangle + b\} \le \varepsilon + \xi_{R,k}' \\ \Im\{-h_k + \langle \boldsymbol{w}, \phi(s_k)\rangle + b\} \le \varepsilon + \xi_{I,k}' \\ \xi_{R,k}^{(\prime)}, \xi_{I,k}^{(\prime)} \ge 0\end{cases} \tag{3.13}$$

Through Lagrangian duality, the original minimization of $\mathcal{L}_p$ can be solved by maximizing $\mathcal{L}_{\boldsymbol{d}}$ with respect to Lagrangian multipliers $\boldsymbol{\alpha_R}$, $\boldsymbol{\alpha_I}$, $\boldsymbol{\alpha_R'}$, and $\boldsymbol{\alpha_I'}$, where

$$\begin{aligned}\mathcal{L}_d = -\frac{1}{2}\boldsymbol{\psi}^H(\langle\phi(\boldsymbol{s}), \phi(\boldsymbol{s})\rangle + \delta\mathbf{I})\boldsymbol{\psi} + (\boldsymbol{\alpha_R} - \boldsymbol{\alpha_R'})^T\Re\{\boldsymbol{h}\} \\ + (\boldsymbol{\alpha_I} - \boldsymbol{\alpha_I'})^T\Im\{\boldsymbol{h}\} - \varepsilon\mathbf{1}^T(\boldsymbol{\alpha_R} + \boldsymbol{\alpha_I} + \boldsymbol{\alpha_R'} + \boldsymbol{\alpha_I'}).\end{aligned} \tag{3.14}$$

and subject to constraints

$$\begin{cases} \sum_k (\alpha_{R,k} - \alpha'_{R,k}) = \sum_k (\alpha_{I,k} - \alpha'_{I,k}) = 0 \\ 0 \le \alpha_{R,k}, \alpha'_{R,k}, \alpha_{I,k}, \alpha'_{I,k} \le C \end{cases} \tag{3.15}$$

with derivation shown in Appendix A. In (3.14), $\boldsymbol{\psi}$ is a shorthand notation that can be expanded as

$$\boldsymbol{\psi} = \boldsymbol{\alpha_R} + j\boldsymbol{\alpha_I} - \boldsymbol{\alpha'_R} - j\boldsymbol{\alpha'_I}. \tag{3.16}$$

Furthermore, ϕ in (3.14) need not be explicitly known. With Mercer's theorem, an approach known as the "kernel trick" replaces the inner product $\langle \phi(\boldsymbol{s}), \phi(\boldsymbol{s}) \rangle$ with a positive semidefinite kernel function $K(x, y)$. In this implementation's SVR, the radial basis function (RBF) is selected as a popular kernel choice with the following formula:

$$K(x, y) = exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right). \tag{3.17}$$

Using the pilot subcarriers s_k as the input parameters, the N_{SP} x N_{SP} kernel matrix $\mathbf{K}$ is constructed as follows:

$$\boldsymbol{K} = \begin{bmatrix} K(s_1, s_1) & \cdots & K(s_1, s_{N_{SP}}) \\ \vdots & \ddots & \vdots \\ K(s_{N_{SP}}, s_1) & \cdots & K(s_{N_{SP}}, s_{N_{SP}}) \end{bmatrix}, \tag{3.18}$$

where N_{SP} represents the number of pilot subcarriers. Consequently, (3.14) is rewritten as

$$\begin{aligned} \mathcal{L}_d = -\frac{1}{2}\boldsymbol{\psi}^H(\mathbf{K} + \delta\mathbf{I})\boldsymbol{\psi} + (\boldsymbol{\alpha_R} - \boldsymbol{\alpha'_R})^T \Re\{\boldsymbol{h}\} \\ + (\boldsymbol{\alpha_I} - \boldsymbol{\alpha'_I})^T \Im\{\boldsymbol{h}\} - \varepsilon \mathbf{1}^T(\boldsymbol{\alpha_R} + \boldsymbol{\alpha_I} + \boldsymbol{\alpha'_R} + \boldsymbol{\alpha'_I}), \end{aligned} \tag{3.19}$$

where $\mathbf{I}$ represents the identity matrix and $\mathbf{1}$ specifies a vector of all ones.

Utilizing the preamble's known training OFDM symbols T_1 and T_2, the SVR's free parameters C, δ, ε, and σ^2 can be automatically determined. Each parameter can be viewed as a complex value with the logic for the imaginary component's derivation

mirroring the real. Through guidance from [17] and experimental testing, the following definitions are utilized:

$$C = \max\left(\left|\overline{(\boldsymbol{h_2}/\boldsymbol{h_1})} + std(\boldsymbol{h_2}/\boldsymbol{h_1})\right|, \left|\overline{(\boldsymbol{h_2}/\boldsymbol{h_1})} - std(\boldsymbol{h_2}/\boldsymbol{h_1})\right|\right) \tag{3.20}$$

$$\delta = \frac{\sum \sigma_{\boldsymbol{y}}^2}{\sum(\overline{\boldsymbol{y}})^2} \tag{3.21}$$

$$\varepsilon = \frac{\mathrm{var}(\boldsymbol{h_2}/\boldsymbol{h_1})}{N_{SP}} \tag{3.22}$$

$$\sigma^2 = \overline{\sigma_{\boldsymbol{h}}^2} \tag{3.23}$$

where N_{SP} denotes the number of pilot subcarriers, $\boldsymbol{y_1}$ and $\boldsymbol{y_2}$ represent the received signals for T_1 and T_2 and $\boldsymbol{h_1}$ and $\boldsymbol{h_2}$ represent their LS channel estimates. For notation, $\overline{\boldsymbol{h_1}}$ denotes the mean of all subcarriers for T_1 whereas $\overline{\boldsymbol{h}}$ contains the N_{ST} means across T_1 and T_2; the same applies to $\sigma_{\boldsymbol{h_1}}$ and $\sigma_{\boldsymbol{h}}$. As a side note, for each expression involving only $\boldsymbol{h_1}$, the expression can be logically expanded to incorporate both $\boldsymbol{h_1}$ and $\boldsymbol{h_2}$. However, the slight performance gain may not be worth the additional computation time.

Finally, the optimization problem can be solved iteratively through sequential minimal optimization (SMO) to decompose the problem into the smallest sized sub-problems [18] with a working set selection utilizing second order information to choose the pairs of indices at each iteration in order to achieve fast convergence [19]. After meeting a tolerance of 0.001, the optimization results are substituted into

$$\hat{h}_m = \sum_n \psi_n K(s_m, s_n) + b, \tag{3.24}$$

for $n = 1, \ldots, N_{SP}$ to obtain the channel estimate for the corresponding normalized subcarrier frequency s_m. Then, the receiver performs channel equalization to obtain the current estimated received signal by reversing the distortion from the current OFDM

symbol's SVR channel estimate on top of the baseline channel estimate acquired during the preamble.

3.6. Error Decoding

Even with excellent channel equalization, it remains improbable to sustain a bit error rate (BER) of zero without coding overhead in wireless communications, especially in the envisioned hyper-wideband link. In section 2.5, binary convolutional coding (BCC) and low density parity check (LDPC) coding are introduced as two forward error correction (FEC) codes utilized in industry standards such as IEEE 802.11. However, while implementation of an FEC at the transmitter is largely fixed, there exists multiple ways to decode on the receiver side.

Regardless of BCC or LDPC, the QAM constellations are first demapped into soft metrics based on IQ error vectors. Figure 3.7 and Table 3.1 demonstrates an example of demapping an IQ signal following 16-QAM constellation to corresponding bit errors depending on the receiver's decision. Each metric's precision and decision condition are preserved for further decoding stages rather than formulating a hard decision or a log likelihood ratio (LLR). After demapping, the metrics are deinterleaved by reversing the permutations performed at the transmitter.

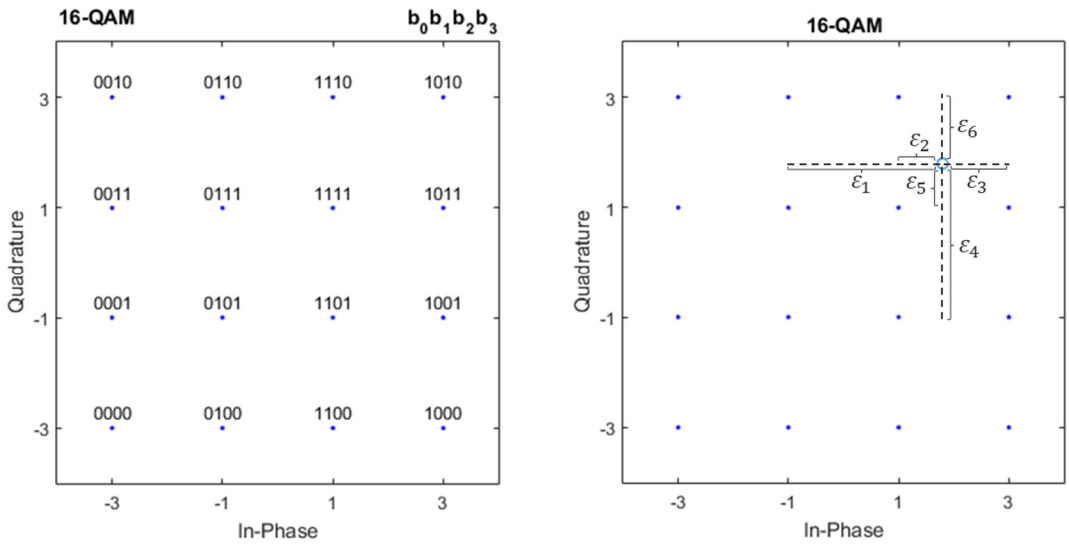

Figure 3.7. Example of demapping 16-QAM IQ signal to bit error metrics

Table 3.1 Example of Soft Bit Error Metrics After Demapping

	b_0	b_1	b_2	b_3
Decide '0'	ε_1	ε_3	ε_4	ε_6
Decide '1'	ε_2	ε_2	ε_5	ε_2

At this point, the decoding process diverges depending on the FEC chosen. For BCC, Viterbi decoding calculates the cumulative likelihood or error for every possible path and selects the most likely sequence, or the sequence with minimum cumulative error [7]. Since the polynomial order for the BCC of interest is 6, there exists 2^6, or 64, possible states at any given time, with the exception of the initial 6 input bits. This operation can be visualized via a trellis diagram as shown in Figure 3.8, where the dashed blue lines portray the possible paths for each state s at time t_i, the solid red line depicts the optimal path, and the red metrics e_i contain the accumulated error for the optimal path at time t_i.

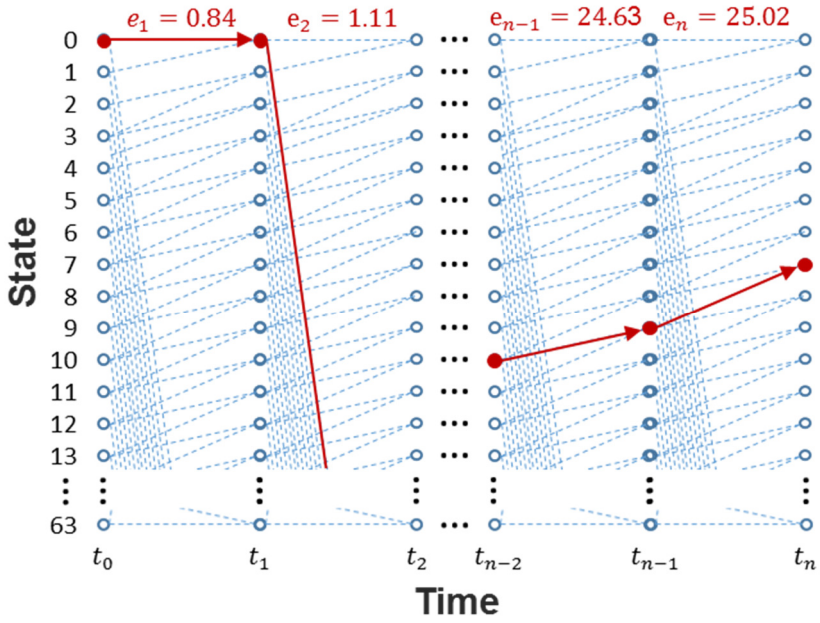

Figure 3.8. Sample trellis depicting soft Viterbi decoding for BCC

Although extremely time consuming and memory intensive, this strategy optimally decodes the input sequence. However, the receiver can drastically cut down on the number of possibilities by retaining only the best path at each of the 64 states for the present time. Additionally, this reduction eliminates the need for resources to handle collisions, i.e. keeping track of multiple paths per state.

For LDPC, an iterative algorithm known as message passing is commonly used for decoding. To begin, the input sequence of error values for deciding '0' are subtracted from the error values for deciding '1' to acquire a likelihood metric, so that a positive likelihood favors a '1' and a negative likelihood leans towards a '0'. Next, these likelihood metrics are divided into codewords for evaluation according to the parity check matrix H. As illustrated in Figure 3.9, the $(n-k)$ x n parity check matrix H can be framed as a bipartite graph with $(n-k)$ check nodes and n variable nodes, where an edge exists between check node c and variable node v when $H(c,v) = 1$.

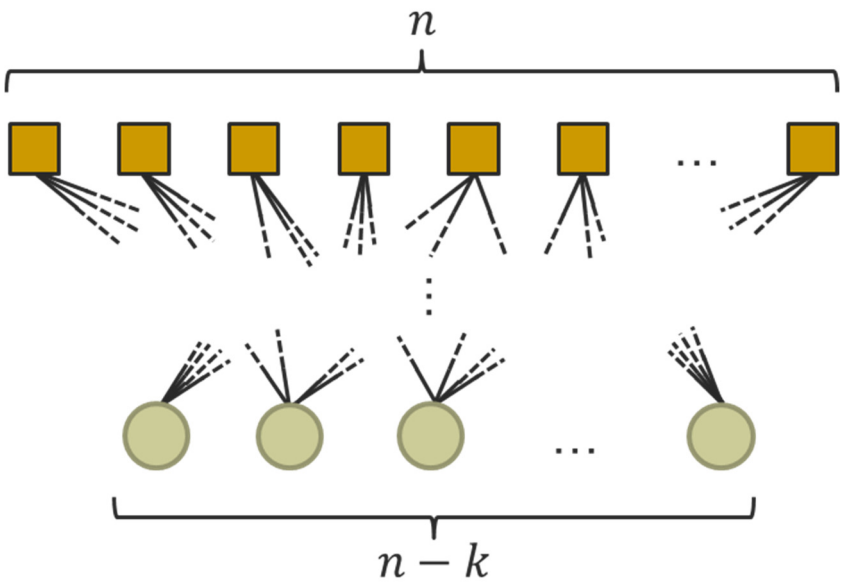

Figure 3.9. LDPC bipartite graph with $\boldsymbol{n}$ variable nodes and $(\boldsymbol{n-k})$ check nodes

The input likelihood metrics initialize the variable nodes and then the iterative process starts with the variable nodes passing their values onto the check nodes. At each check node, the parity of the likelihoods is calculated. If the parity for all check nodes is a 0, then the algorithm is done and a hard decision discloses the first k message bits from the codeword. Otherwise, each check node updates their associated variable nodes with the lowest value previously received and that value is added to the variable nodes. For clarity, although the lowest value is determined by magnitude, the updated sum includes the sign. This message passing process repeats until a parity of 0 is achieved or the algorithm reaches a pre-determined maximum number of iterations and fails [20]. For the hyper-wideband experiments, four iterations is set as the maximum.

When comparing between BCC and LDPC, LDPC's structure offers lower complexity and faster performance. In particular, the message passing and independent codeword processing lends to potential for multi-threading as opposed to the feedback requirement of BCC and sequential traceback for Viterbi decoding. Moreover, it has been

shown that long blocks for LDPC approach near-capacity performance [9] and LDPC in general performs considerably better than BCC [21]. Therefore, the wireless experiments for hyper-wideband communications focus exclusively on LDPC FEC.

CHAPTER 4

WIRELESS EXPERIMENT

4.1. Setup

In summary, the plan is to transmit and receive the 4 GHz-wide signal at a carrier frequency of 9 GHz. For this setup, the majority of the transmitter and receiver components, such as up/down-conversion and filtering, are implemented in software. As illustrated in Figure 4.1, the hardware components consist of an arbitrary waveform generator (AWG), an amplifier, a pair of quad-ridge horn antennas, and a real-time scope. During a test run, a packet containing 7 OFDM data symbols (not counting the preamble) and 39.208 μs long is generated and loaded into the AWG. The packet is then broadcasted repeatedly at the carrier frequency of 9 GHz and sampling rate of with 24 GS/s. Located 1.5 meters away at the same height, the Lecroy WaveMaster 830Zi receives and stores a 50 μs duration sampled at 40 GS/s. Unsurprisingly, data collections without a full packet are discarded.

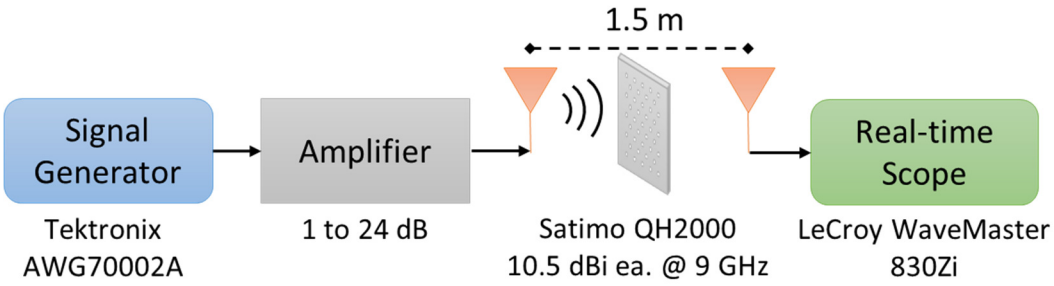

Figure 4.1. Experimental setup for conducting indoor wireless tests

An important point to note is that the Tektronix AWG70002A instrument specifies a maximum frequency of only 10 GHz, which is 1 GHz less than required for the targeted parameters. Moreover, Tektronix reports an effective number of bits (ENOB)

of 5.6 for this particular model. Therefore, the AWG is expected to play a non-trivial and detrimental impact on the signal. At the same time, this situation offers an excellent opportunity to test the channel estimation scheme. On the other end, the Lecroy WaveMaster 830Zi possesses a sufficient analog bandwidth of 20 GHz at the 40 GS/s sampling rate. In the 7 to 11 GHz frequency range of interest, the pair of Saitmo QH2000 antennas radiate from 8.5 dBi to 11 dBi with a largely smooth, linear curve.

In order to vary the signal-to-noise ratio (SNR), attenuators are added to the amplifier in the transmitter chain. Additionally, both line-of-sight (LOS) and near-line-of-sight (nLOS) experiments are conducted. For the nLOS tests, an aluminum block is placed halfway between the transmitter and receiver at equal height. The block's dimensions are 16 x 16 x 1.2 cm, and it contains drilled holes of 2.5 mm radius spaced 2.5 cm apart from each other. A series of runs at various SNRs, LOS or nLOS, and constellations (BPSK, QPSK, 16-QAM, and 64-QAM) are performed.

4.2. Results

At the end of the experiments, each packet's bit error rate (BER) is first calculated without forward error correction (FEC). Afterwards, a binary result of pass or fail is given with the inclusion of FEC. The BER vs SNR plots without FEC for the various constellations, inter/extrapolation channel estimation models, and LOS or nLOS environment are shown in Figure 4.2, where markers with solid fill represent test runs that passed with FEC.

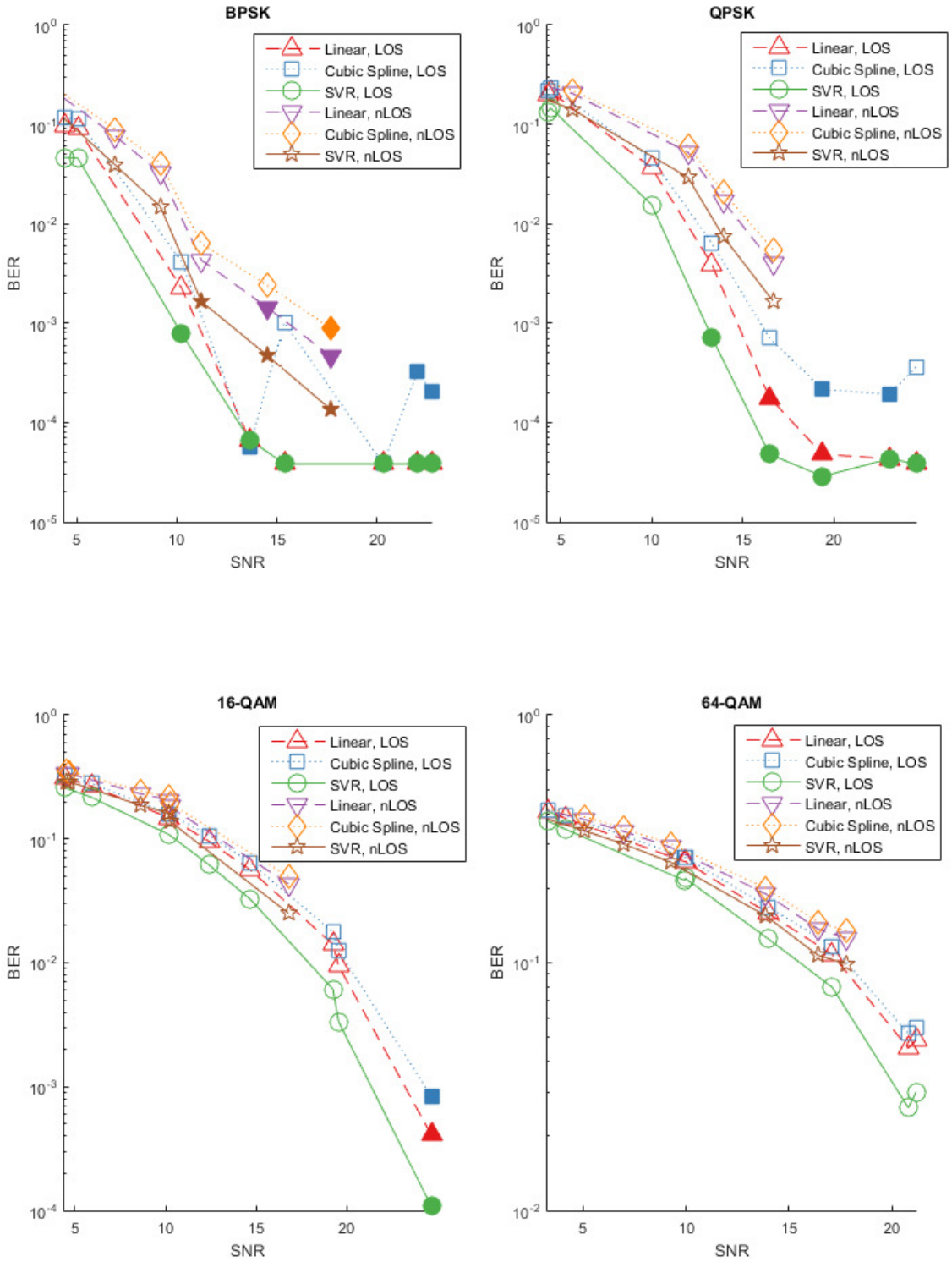

Figure 4.2. BER vs SNR plots for wireless experiments with various configurations

As can be seen, the SVR approach either matched or performed better than linear and cubic spline inter/extrapolation in both BER and number of passed packets for every data point. Moreover, the SVR's consistency in better performance than its alternatives becomes clearer at lower SNRs and higher constellations. On the other hand, the cubic spline method unquestionably proved the worst of the three.

Additionally, the plots affirm the logical trend of lower BER with higher SNR. One possible factor for the exceptions can be blamed on erroneous channel estimate extrapolation, particularly for the linear and cubic spline techniques. As a solution, the protocol can be modified so that the lowest and highest non-null subcarriers with respect to frequency carry pilot tones instead of data. However, the additional overhead is not necessary when using the SVR approach. It can also be seen that the BPSK and QPSK configurations reveal a BER floor that can be attributed to the packet's limited number of bits for calculating BERs with higher precision.

Finally, the dismal pass rate for 16-QAM is observed with only one passed packet for each of the channel estimation methods. Furthermore, zero packets passed with FEC for 64-QAM. Therefore, only BPSK and QPSK should be considered for this particular hyper-wideband communication setup. Potential solutions for improving the performance include reducing the coding rate at the cost of throughput speed, transmitting at higher power to increase SNR, or employing D/A and A/D converters with higher ENOB, although the hardware constraint is much more difficult to surpass.

For perspective, Figure 4.3 exhibits a snapshot of the channel calculated from one of the OFDM training symbols during one of the experiments with an SNR of 11.2 dB. Note that when describing the channel, elements such as the instrument's internal

electronics and the digital filters are included in addition to the over-air propagation. The numerous spikes in magnitude and phase changes are evident even at this SNR level and signifying that frequency selective fading is a significant problem to hyper-wideband communications.

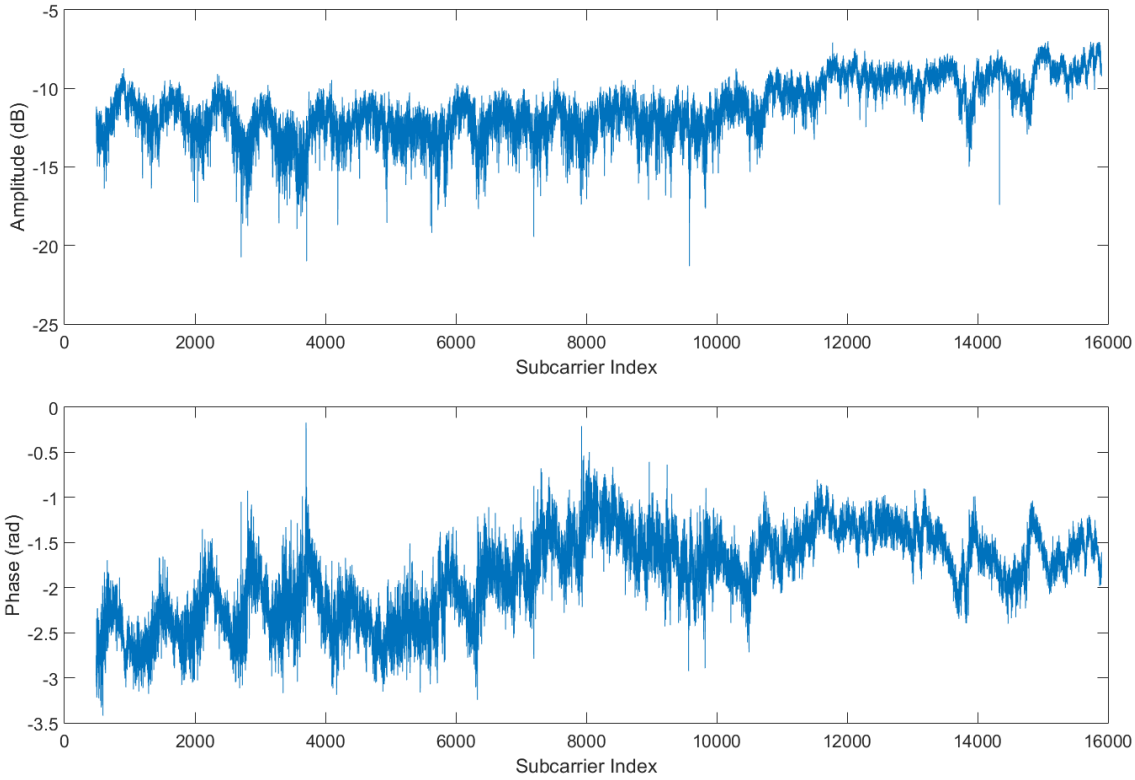

Figure 4.3. Channel spanshot for an OFDM symbol at SNR of 11.2 dB

CHAPTER 5

CONCLUSION

In this thesis, an end-to-end physical layer specification for OFDM hyper-wideband communications is designed and implemented. Additionally, the importance and difficulties in channel estimation was discussed and a robust support vector statistical learning approach was proposed. Wireless experiments were conducted with various configurations, including constellations, intra/extrapolation channel estimation methods, and LOS or nLOS environment. The results revealed that the SVR approach performed the best of the attempted methods, as well as illuminating potential pitfalls to be addressed in future efforts.

Further testing in similar and different environments should be conducted to obtain more empirical results for extending the community's knowledge in hyper-wideband communications. Additionally, future work may include exploring multiple-input and multiple-output (MIMO) to enhance capacity by exploiting multipath propagation. For implementation, significant work is required to optimize the software for target platforms of interest in order to achieve real-time performance. Finally, the SVR method to robust channel estimation can be extended beyond hyper-wideband to address similar OFDM protocols in other challenging environments, such as vehicular or underwater communications.

APPENDIX A

SUPPORT VECTOR REGRESSION DERIVATION

The purpose of this appendix is to show the derivation from minimizing $\mathcal{L}_p$ defined in (3.12) and (3.13) to maximizing its dual $\mathcal{L}_d$ defined in (3.14) and (3.15). For convenience, (3.12) and (3.13) are reproduced below as (A.1) and (A.2):

$$\begin{aligned}\mathcal{L}_p = \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{1}{2\delta}\sum_{k\epsilon K_1}\left(\xi_{R,k}^2 + \xi_{R,k}'^2\right) + C\sum_{k\epsilon K_2}\left(\xi_{R,k} + \xi_{R,k}'\right) \\ + \frac{1}{2\delta}\sum_{k\epsilon K_3}\left(\xi_{I,k}^2 + \xi_{I,k}'^2\right) + C\sum_{k\epsilon K_4}\left(\xi_{I,k} + \xi_{I,k}'\right) - \sum_{k\epsilon K_{2,}K_4}\frac{\delta C^2}{2}\end{aligned} \tag{A.1}$$

subject to

$$\begin{cases}\Re\{h_k - \langle \boldsymbol{w}, \phi(s_k)\rangle - b\} \le \varepsilon + \xi_{R,k} \\ \Im\{h_k - \langle \boldsymbol{w}, \phi(s_k)\rangle - b\} \le \varepsilon + \xi_{I,k} \\ \Re\{-h_k + \langle \boldsymbol{w}, \phi(s_k)\rangle + b\} \le \varepsilon + \xi_{R,k}'. \\ \Im\{-h_k + \langle \boldsymbol{w}, \phi(s_k)\rangle + b\} \le \varepsilon + \xi_{I,k}' \\ \xi_{R,k}^{(\prime)}, \xi_{I,k}^{(\prime)} \ge 0\end{cases} \tag{A.2}$$

Likewise, (3.14) and (3.15) are reproduced below as (A.3) and (A.5):

$$\begin{aligned}\mathcal{L}_d = -\frac{1}{2}\boldsymbol{\psi}^H(\langle\phi(\boldsymbol{s}), \phi(\boldsymbol{s})\rangle + \delta\mathbf{I})\boldsymbol{\psi} + (\boldsymbol{\alpha_R} - \boldsymbol{\alpha_R'})^T\Re\{\boldsymbol{h}\} \\ +(\boldsymbol{\alpha_I} - \boldsymbol{\alpha_I'})^T\Im\{\boldsymbol{h}\} - \varepsilon\mathbf{1}^T(\boldsymbol{\alpha_R} + \boldsymbol{\alpha_I} + \boldsymbol{\alpha_R'} + \boldsymbol{\alpha_I'}),\end{aligned} \tag{A.3}$$

where

$$\boldsymbol{\psi} = \boldsymbol{\alpha_R} + j\boldsymbol{\alpha_I} - \boldsymbol{\alpha_R'} - j\boldsymbol{\alpha_I'} \tag{A.4}$$

and subject to

$$\begin{cases}\sum_k\left(\alpha_{R,k} - \alpha_{R,k}'\right) = \sum_k\left(\alpha_{I,k} - \alpha_{I,k}'\right) = 0 \\ 0 \le \alpha_{R,k}, \alpha_{R,k}', \alpha_{I,k}, \alpha_{I,k}' \le C\end{cases}. \tag{A.5}$$

Next, the Lagrangian $\mathcal{L}_{pd}$ is formed from (A.1) with multipliers $\boldsymbol{\alpha_R}$, $\boldsymbol{\alpha_I}$, $\boldsymbol{\alpha'_R}$, $\boldsymbol{\alpha'_I}$, $\boldsymbol{\beta_R}$, $\boldsymbol{\beta_I}$, $\boldsymbol{\beta'_R}$, and $\boldsymbol{\beta'_I}$:

$$\begin{aligned}\mathcal{L}_{pd} = {} & \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{1}{2\delta}\sum_{k\epsilon K_1}\left(\xi_{R,k}^2 + \xi_{R,k}'^2\right) + C\sum_{k\epsilon K_2}\left(\xi_{R,k} + \xi'_{R,k}\right) \\ & + \frac{1}{2\delta}\sum_{k\epsilon K_3}\left(\xi_{I,k}^2 + \xi_{I,k}'^2\right) + C\sum_{k\epsilon K_4}\left(\xi_{I,k} + \xi'_{I,k}\right) - \sum_{k\epsilon K_{2,}K_4}\frac{\delta C^2}{2} \\ & + \sum_k \alpha_{R,k}\left(\Re\{h_k - \langle \boldsymbol{w}, \phi(s_k)\rangle - b\} - \varepsilon - \xi_{R,k}\right) \\ & + \sum_k \alpha_{I,k}\left(\Im\{h_k - \langle \boldsymbol{w}, \phi(s_k)\rangle - b\} - \varepsilon - \xi_{I,k}\right) \\ & + \sum_k \alpha'_{R,k}\left(\Re\{-h_k + \langle \boldsymbol{w}, \phi(s_k)\rangle + b\} - \varepsilon - \xi'_{R,k}\right) \\ & + \sum_k \alpha'_{I,k}\left(\Im\{-h_k + \langle \boldsymbol{w}, \phi(s_k)\rangle + b\} - \varepsilon - \xi'_{I,k}\right) \\ & - \sum_k \beta_{R,k}^{(\prime)}\xi_{R,k}^{(\prime)} - \sum_k \beta_{I,k}^{(\prime)}\xi_{I,k}^{(\prime)}.\end{aligned} \tag{A.6}$$

Taking the gradient of (A.6) with respect to $\boldsymbol{w}$, b, and ξ and setting them to zero results in

$$\nabla_{\boldsymbol{w}}\mathcal{L}_{pd} = \boldsymbol{w} - \sum_k \left(\alpha_{R,k} + j\alpha_{I,k} - \alpha'_{R,k} - j\alpha'_{I,k}\right)\phi(s_k) = 0 \tag{A.7}$$

$$\nabla_b\mathcal{L}_{pd} = \sum_k \left(\alpha_{R,k} - \alpha'_{R,k}\right) = \sum_k \left(\alpha_{I,k} - \alpha'_{I,k}\right) = 0 \tag{A.8}$$

$$\nabla_{\xi_{R,k}^{(\prime)}}\mathcal{L}_{pd} = \begin{cases}\frac{1}{\delta}\xi_{R,k}^{(\prime)} - \alpha_{R,k}^{(\prime)} - \beta_{R,k}^{(\prime)} = 0, & k \in K_1 \\ C - \alpha_{R,k}^{(\prime)} - \beta_{R,k}^{(\prime)} = 0, & k \in K_2\end{cases} \tag{A.9}$$

$$\nabla_{\xi_{I,k}^{(\prime)}}\mathcal{L}_{pd} = \begin{cases}\frac{1}{\delta}\xi_{I,k}^{(\prime)} - \alpha_{I,k}^{(\prime)} - \beta_{I,k}^{(\prime)} = 0, & k \in K_3 \\ C - \alpha_{I,k}^{(\prime)} - \beta_{I,k}^{(\prime)} = 0, & k \in K_4\end{cases}. \tag{A.10}$$

From (A.7), the following expressions can be obtained:

$$\boldsymbol{w} = \sum_k \left(\alpha_{R,k} + j\alpha_{I,k} - \alpha'_{R,k} - j\alpha'_{I,k}\right)\boldsymbol{\phi}(s_k) = \sum_k \psi_k \boldsymbol{\phi}(s_k)$$

$$\|\boldsymbol{w}\|^2 = \boldsymbol{\psi}^H \langle \boldsymbol{\phi}(\boldsymbol{s}), \boldsymbol{\phi}(\boldsymbol{s})\rangle \boldsymbol{\psi} \tag{A.11}$$

$$\sum_k \left(-\alpha_{R,k} - j\alpha_{I,k} + \alpha'_{R,k} + j\alpha'_{I,k}\right)\langle \boldsymbol{w}, \boldsymbol{\phi}(s_k)\rangle = -\boldsymbol{\psi}^H \langle \boldsymbol{\phi}(\boldsymbol{s}), \boldsymbol{\phi}(\boldsymbol{s})\rangle \boldsymbol{\psi}.$$

Due to (A.8), certain expressions from (A.6) can be zeroed, including

$$\begin{aligned} \Re\{b\} \sum_k \left(\alpha_{R,k} - \alpha'_{R,k}\right) = 0 \\ \Im\{b\} \sum_k \left(\alpha_{I,k} - \alpha'_{I,k}\right) = 0. \end{aligned} \tag{A.12}$$

Furthermore, Karush–Kuhn–Tucker (KKT) conditions force

$$\begin{aligned} \beta^{(\prime)}_{R,k} \xi^{(\prime)}_{R,k} = 0 \\ \beta^{(\prime)}_{I,k} \xi^{(\prime)}_{I,k} = 0, \end{aligned} \tag{A.13}$$

which implies that

$$\begin{aligned} \sum_k \beta^{(\prime)}_{R,k} \xi^{(\prime)}_{R,k} = \sum_k \beta^{(\prime)}_{I,k} \xi^{(\prime)}_{I,k} = 0 \\ \beta^{(\prime)}_{R,k} = 0 \;\; if \;\; \xi^{(\prime)}_{R,k} > 0 \\ \beta^{(\prime)}_{I,k} = 0 \;\; if \;\; \xi^{(\prime)}_{I,k} > 0. \end{aligned} \tag{A.14}$$

Rearranging terms in (A.9) and (A.10) and utilizing (A.14) results in

$$\begin{aligned} \xi^{(\prime)}_{R,k} = \delta\alpha^{(\prime)}_{R,k} \\ \xi^{(\prime)}_{I,k} = \delta\alpha^{(\prime)}_{I,k} \\ C = \alpha^{(\prime)}_{R,k} = \alpha^{(\prime)}_{I,k}. \end{aligned} \tag{A.15}$$

With the addition of (A.15), some of the expressions in (A.6) can be re-written as

$$\frac{1}{2\delta}\sum_{k\epsilon K_1}\left(\xi_{R,k}^2+\xi_{R,k}'^2\right)+\frac{1}{2\delta}\sum_{k\epsilon K_3}\left(\xi_{I,k}^2+\xi_{I,k}'^2\right)=\frac{\delta}{2}\left[\sum_{k\epsilon K_1}\left(\alpha_{R,k}^2+\alpha_{R,k}'^2\right)+\sum_{k\epsilon K_3}\left(\alpha_{I,k}^2+\alpha_{I,k}'^2\right)\right] \quad \text{(A.16)}$$

$$C\sum_{k\epsilon K_2}\left(\xi_{R,k}+\xi_{R,k}'\right)+C\sum_{k\epsilon K_4}\left(\xi_{I,k}+\xi_{I,k}'\right)=\delta\left[\sum_{k\epsilon K_2}\left(\alpha_{R,k}^2+\alpha_{R,k}'^2\right)+\sum_{k\epsilon K_4}\left(\alpha_{I,k}^2+\alpha_{I,k}'^2\right)\right] \quad \text{(A.17)}$$

$$\sum_{k\epsilon K_2,K_4}\frac{\delta C^2}{2}=\frac{\delta}{2}\left[\sum_{k\epsilon K_2}\left(\alpha_{R,k}^2+\alpha_{R,k}'^2\right)+\sum_{k\epsilon K_4}\left(\alpha_{I,k}^2+\alpha_{I,k}'^2\right)\right]. \quad \text{(A.18)}$$

Finally, by substituting, zeroing, and/or rearranging expressions obtained from (A.11), (A.12), (A.16), (A.17), and (A.18), the desired dual $\mathcal{L}_d$ of (A.3) and (A.5) can be obtained from $\mathcal{L}_{pd}$ defined in (A.6).

REFERENCES

[1] DARPA, “Broad Agency Announcement Hyper-wideband Enabled RF Messaging,” DARPA-BAA-14-34, June 30, 2014, [Online]. Available: http:// www.fbo.gov. [Accessed: Apr. 1, 2016].

[2] M. J. F. G. Garcia, J. L. Rojo-Alvarez, F. Alonso-Atienza and M. Martinez-Ramon, "Support vector machines for robust channel estimation in OFDM," in IEEE Signal Processing Letters, vol. 13, no. 7, pp. 397-400, July 2006.

[3] D. Wu and W. Fan, “A New Channel Estimation Method Based on Pilot-aided and Local Adaptive Least Squares Support Vector Regression in Software Radio OFDM System,” Artificial Intelligence, 2009. JCAI '09. International Joint Conference on, Hainan Island, 2009, pp. 349-352.

[4] A. Charrada and A. Samet, “Nonlinear complex LS-SVM for highly selective OFDM channel with impulse noise,” Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), 2012 6th International Conference on, Sousse, 2012, pp. 696-700.

[5] IEEE Std 802.11ac-2013, “Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications--Amendment 4: Enhancements for Very High Throughput for Operation in Bands below 6 GHz," Dec. 2013.

[6] M. Kim, Y. Konishi, Y. Chang and J. Takada, “Large Scale Parameters and Double-Directional Characterization of Indoor Wideband Radio Multipath Channels at 11 GHz,” in IEEE Transactions on Antennas and Propagation, vol. 62, no. 1, pp. 430-441, Jan. 2014.

[7] M. Lobeira Rubio, A. Garcia-Armada, R. P. Torres and J. L. Garcia, “Channel modeling and characterization at 17 GHz for indoor broadband WLAN,” in IEEE Journal on Selected Areas in Communications, vol. 20, no. 3, pp. 593-601, Apr 2002.

[8] A.J. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” IEEE Trans. Inform. Theory, vol. 13, pp. 260–269, Apr. 1967.

[9] R.G. Gallager, “Low density parity check codes,” IRE Trans. Inform. Theory, vol. 8, no. 1, pp. 21–28, Jan. 1962.

[10] T.M. Schmidl and D.C. Cox, “Robust Frequency and Timing Synchronization for OFDM,” IEEE Trans. Commun., vol. 45, no. 12, pp. 613-1621, Dec. 1997.

[11] Y. Shen and E. Martinez, “Channel Estimation in OFDM Systems,” Freescale Semiconductor Application Note, Rev. 0, Jan. 2006.

[12] B.E. Boser, I.M. Guyon, V.N. Vapnik, “A training algorithm for optimal margin classifiers,” Proceedings of the 5th Annual Workshop on Comp. Learning Theory, pp. 144-152, 1992.

[13] V.N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, NY, 1995.

[14] G. Camps-Valls, L. Bruzzone, J.L. Rojo-Alvarez, F. Melgani, “Robust support vector regression for biophysical variable estimation from remotely sensed images,” IEEE Geo. and Remote Sensing Letters, vol. 3, no. 3, pp. 339-343, Jul. 2006.

[15] L.J. Cao and F.E.H. Tay, “Support vector machine with adaptive parameters in financial time series forecasting,” IEEE Trans. Neural Networks, vol. 14, no. 6, pp. 1506-1518, Nov. 2003.

[16] J. L. Rojo-Alvarez, M. Martinez-Ramon, A. R. Figueiras-Vidal, A. Garcia-Armada and A. Artes-Rodriguez, “A robust support vector algorithm for nonparametric spectral analysis,” in IEEE Signal Processing Letters, vol. 10, no. 11, pp. 320-323, Nov. 2003.

[17] V. Cherkassky, Y. Ma, “Practical selection of SVM parameters and noise estimation for SVM regression,” in Neural Networks, vol. 17, no. 1, pp. 113-126, Jan. 2004.

[18] J. C. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Scholkopf, C. J. C. Burges, and A. J. Smola, editors. *Advances in Kernel Methods – Support Vector Learning*, Cambridge, MA, 1998. MIT Press.

[19] R.-E. Fan, P.-H. Chen, and C.-J. Lin, “Working set selection using second order information for training Support Vector Machines,” Journal of Machine Learning Research 6, 2005, pp. 1889–1918.

[20] F. R. Kschischang, B. J. Frey and H. A. Loeliger, "Factor graphs and the sum-product algorithm," in IEEE Transactions on Information Theory, vol. 47, no. 2, pp. 498-519, Feb 2001.

[21] A. Purkovic, N. Burns, S. Sukobok, and L. Demirekler, “LDPC vs. Convolutional Codes for 802.11n Applications: Performance Comparison”, IEEE 802.11-03/865r1, Nortel Networks, Jan. 2004.