SHALLOW UNDER WATER COMMUNICATION WITH PASSIVE PHASE CONJUGATION AND ITERATIVE DEMODULATION AND DECODING

By

CHRISTOPHER CORSON KEESER

A thesis submitted in partial fulfillment of
the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

WASHINGTON STATE UNIVERSITY
School of Electrical Engineering and Computer Science

December 2008

To the Faculty of Washington State University:

The members of the Committee appointed to examine the thesis of Christopher Corson Keeser find it satisfactory and recommend that it be accepted.

Co-Chair

Co-Chair

ACKNOWLEDGMENT

I would like to thank my wife Tiffiny for being my world and my reason. I would like to thank my father for inspiring me to believe me in myself, and I would like to thank my mother for never giving up on me. Without you I would not be where I am today. Thank you.

I would also like the thank Sierra, Abby and Serenity, my three little angels who told me daily,“don’t worry daddy, you will finish on time!” They are the best children a dad could hope for.

To Dr. Belzer and Dr. Fischer. You have both been beacons of education for me. I am deeply grateful for the time and energy you have both spent in the noble pursuit of education and passing knowledge on to others. Your help and guidance have been principle in my success.

I would like to extend special thanks the Navy’s Acoustic Research Detachment at Bayview Idaho for logistical and equipment support in carrying out the lake tests. In particular, I would like to thank Doug Odell, Henry Netzer, Alan Griffits and Scott Jorgensen of the ARD.

I would like to extend my gratitude to the National Science Foundation for their financial support under NSF grant CCF-0635390, and Washington State Universities School of Electrical Engineering and Computer science for their generous fellowship. I would also like to thank Comtech AHA for their financial contributions during the summer.

SHALLOW UNDER WATER COMMUNICATION WITH PASSIVE PHASE CONJUGATION AND ITERATIVE DEMODULATION AND DECODING

Abstract

by Christopher Corson Keeser, M.S.
Washington State University
December 2008

Co-Chair: Benjamin Belzer

Co-Chair: Thomas Fischer

In this thesis we propose a system of passive phase conjugation with iterative demodulation and decoding to increase information throughput in the shallow underwater acoustic channel. The system incorporates passive phase conjugation to mitigate inter-symbol interference caused by the severe multipath environment encountered in the shallow water acoustic channel and a soft-input, soft-output demodulation and a soft-input, soft-output channel decoder in an iterative ‘turbo’ structure to improve the bit error rate performance at higher symbol rates. The system incorporates a linear estimator to estimate slowly varying phase from the channel and to account for Doppler shifts due to source or receiver motion. The system was successfully simulated and achieved a 10^{-3} bit error rate at signal to noise ratios as low as 3.2 dB with random slowly varying phase changes. The system was tested at Lake Pend Orielle in Idaho over a distance of 1.78 km in shallow water with bit error rates below 10^{-5} at 2,500 information symbols per second using a 25 kHz carrier.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

There are many applications that could utilize a high rate, high reliability communications channel in the shallow water environment such as remotely operated vehicles controlled by telemetry and sending back live video and pictures where tethers would be problematic, and underwater sensor networks for data logging and information gathering of various oceanographic datum. However, the shallow water acoustic channel has a high degree of inter-symbol interference (ISI) due to a large amount of multipath interference. This high level of ISI results in a frequency selective fading channel, which impairs high data rate communications.

There have been various solutions proposed and tested to improve the communication speed and reliability in the shallow water acoustic channel, e.g. [1, 2, 3, 4, 5]. Some of the technologies being researched include implementations of direct sequence spread spectrum modems coupled with matching pursuit algorithms that have produced data rates of 161 bps in 2 to 4 meter water over a range of 440 meters [2], phase coherent receivers operating over ranges of 2 kilometers have succeeded in transitting information at 6,700 bps using multiple receivers, quadrature phase shift keying (QPSK) and channel coding [1], passive phase conjugation and binary phase shift keying (BPSK) have been used to achieve data rates around 1.2 kbps in shallow water and orthogonal frequency division multiplexing (OFDM) have also been used to transmit data in the shallow under water acoustic channel [3].

Phase conjugation for mitigation of multipath interference in the oceanic acoustic channel was proposed by Jackson and Dowling [6]. The process involves probing the channel with a symbol, then recording the received symbol at multiple hydrophones spaced sufficiently far apart. The received symbol can either be played back in reverse from each of the hydrophones, which causes the transmitted wavefront to converge in space and time to the location of the source (active phase conjugation), or the received symbol can be used as a matched filter to reduce the effects of ISI on a transmitted symbol sequence (passive phase conjugation).

Active phase conjugation for underwater acoustics can be demonstrated using a single acoustic source and an array of hydrophones. The source transmits a signal and the received signal is recorded at each hydrophone. The recorded signals are then transmitted backwards (time-reversed) from each hydrophone. The resulting transmission converges in space and time to the origin of the transmission (the source hydrophone). This was confirmed experimentally by Kuperman et. al. during tests in the Mediterranean [7].

Passive phase conjugation is similar to active phase conjugation, except that the time reversal and signal convergence are done computationally. A source transmits a probe symbol, which is affected by the multipath environment. This signal is recorded at each receiving hydrophone. After a sufficient period of time has elapsed (so that the majority of the energy in the probe pulse has dissipated), a data sequence is transmitted. Each hydrophone then cross-correlates the received data sequence with the recorded probe pulse which helps to focus the acoustic energy in time. The result of the temporal focusing is then summed across the multiple receivers to reinforce the temporal focusing while averaging out imperfectly focused temporal side lobes. Passive phase conjugation is effectively a temporal and spatial matched filtering process which ‘undoes’ the ISI.

Due to the time-varying nature of the channel, the probing sequence must be repeated periodically in order to maintain a good estimate of the channel. This was one of the topics addressed in a paper written by Rouseff et. al. [5]; and the length of time the channel model remained satisfactory was influenced by a variety of factors and ranged from a half second to many seconds.

To transmit digital information, the transmitted probe pulse is a single symbol representing a ‘1’. This probe pulse is received and correlated with the subsequent symbol stream, summed across all receiver channels and sampled. For coherent demodulation, a positive value at the sampling time indicates a received ‘1’ and a negative value indicates a received ‘-1’. However, coherent reception of the symbol sequence requires phase tracking systems that can suffer from the fading environment. Hoeher and Lodge presented an idea which combined decoding of a channel code with soft information provided by a soft demodulator in an iterative turbo structure [8]. Their method provides a derivation of a APP DPSK demodulator. When combined with an APP channel decoder, their system allowed information to be transmitted differentially and performed within a few tenths of a dB of a coherently demodulated system. In this paper we propose a similar system which requires a reduced number of trellis states compared to [8] and subsequently reduces the computational complexity of the algorithm.

Phase conjugation is used to mitigate the ISI encountered in the underwater acoustic channel. The receiver structure was further motivated by the desire to remove complex decision feedback equalizers, that are not appropriate for the shallow water acoustic channel due to its fading nature, and phase locked loops, that can lose lock in a deep fade and can require a significant amount of time to regain phase lock. We wanted to develop a computationally simple and robust system that can be used for high rate acoustic communications in the shallow water acoustic channel to provide real time telemetry and imagery.

Our system begins with a rate 1/2 irregular repeat accumulate (IRA) low density parity check (LDPC) code. The information bits are encoded using the LDPC channel code, and the bits are randomly interleaved to remove correlation between bits. The interleaved bits are differentially encoded using differential binary phase shift keying (DBPSK) and modulated for transmission. Before transmission of the symbol sequence, a probe symbol is transmitted, followed by a short period of silence to ensure that as much as feasible of the multipath structure is sampled and modeled before the symbol sequence is transmitted. The waveforms are received by multiple hydrophones, bandpass filtered, and converted to baseband. The received probe symbol is cross correlated with the received symbol sequence at each hydrophone and the results summed together. This is then passed to an adaptive sampler which determines the sampling time and adapts to timing drift encountered when the source and receiver clocks are not perfectly matched or when there is relative motion between the source and receiver. The complex symbols are then squared to remove data dependence and a linear predictor tracks the slowly varying phase changes introduced by the channel or by Doppler shifts. The square root of the linear predictor phase is then calculated (to remove the phase doubling produced by squaring the symbols) and unwrapped to provide estimates of the channel phase. The received symbols and the estimates of the channel phase are passed to a modified BCJR algorithm for demodulation of the DBPSK sequence. The BCJR algorithm computes log likelihood ratios (LLRs) for the received symbols, de-interleaves them and passes this information to a soft-input, soft-output (SISO) LDPC decoder. The LDPC decoder iterates on the information for a fixed number of cycles, then re-interleaves the LLRs and passes the new information back to the BCJR algorithm. This cycle repeats until a maximum number of BCJR/LDPC iterations is reached, or until the LLRs have not changed at all in two iterations. The LLRs are then thresholded to recover the transmitted bits.

In this thesis we provide a system which provides good symbol rate performance for shallow water acoustic communications with a low amount of computational complexity. The system was simulated and tested in Lake Pend Oreille, Idaho, at the NAVY's Acoustic Research Detachment. In simulations, the system performed well at signal to noise ratios (SNR, measured as $\frac{E_b}{N_0}$) as low as 3.2 dB with bit errors below 10^{-3}. The system was successfully tested in shallow water over a 1.78 Km distance with information rates of 2,500 bits per second with a 25 kHz carrier and error rates below 10^{-5}. In an uncoded test, the passive phase conjugation method in conjunction with the channel estimation and soft demodulation using the BCJR algorithm provided symbol rates of 10,000 symbols per second with an error rate of 10^{-2} which can easily be corrected with a sufficient channel code.

Chapter 2

SYSTEM STRUCTURE

2.1 Abstract

This chapter discusses the structure of the modem and gives details involved in each element of the system from a signal flow viewpoint. The sections begin with the channel coding and modulation scheme, followed by reception, conversion to baseband and ISI mitigation through passive phase conjugation. Later sections deal with symbol timing recovery, channel phase estimation, demodulation using the BCJR algorithm and iterative demodulation and decoding. A simple overview of the entire system is shown in figure 2.1, and an expanded view of the transmission section can be seen in figure 2.2.

2.2 LDPC Encoding

An IRA LDPC code encodes the data for transmission over the channel. The IRA LDPC code provides near Shannon limit performance at a reduced computational complexity compared to turbo codes, and to general irregular LDPC codes.

The codes were generated using a MATLAB program that constructed the generator matrices for the different block sizes used in this thesis. The target code rate

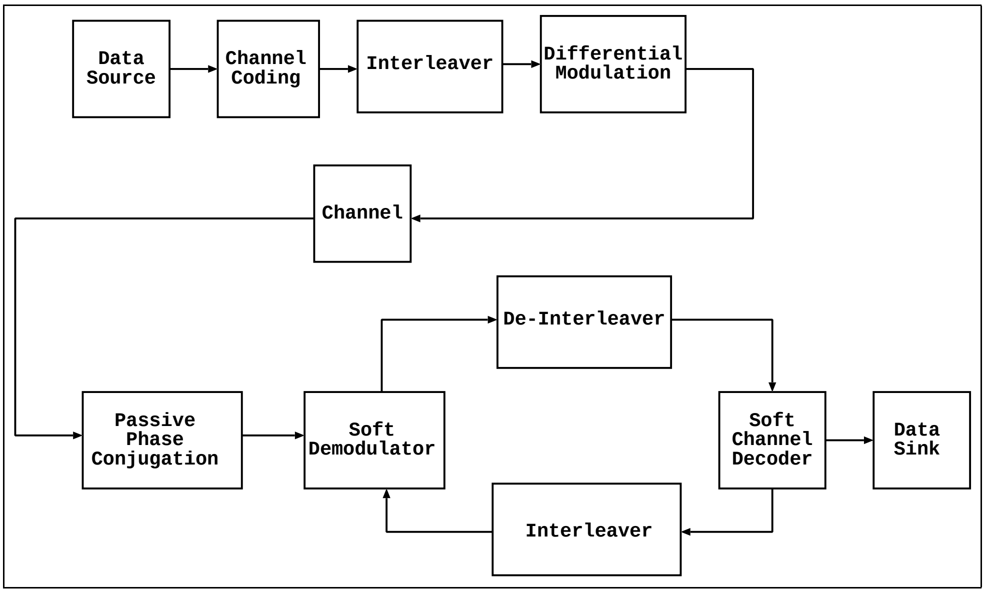

Figure 2.1: System overview block diagram

chosen is 1/2. Since extrinsic information is passed to and from the demodulator and channel decoder, a random interleaver is used to permute the order of these bits to decorrelate them from channel coding to differential modulation. Before differential encoding, a short (40 bit) synchronization sequence is prepended to the interleaved LDPC code block.

2.3 Differential Binary Phase Shift Keying

The encoded and interleaved bits are sent to a differential binary phase shift keyed modulator that encodes the bits according to equation

$$c_k + d_{k-1} \bmod 2 = d_k, \tag{2.1}$$

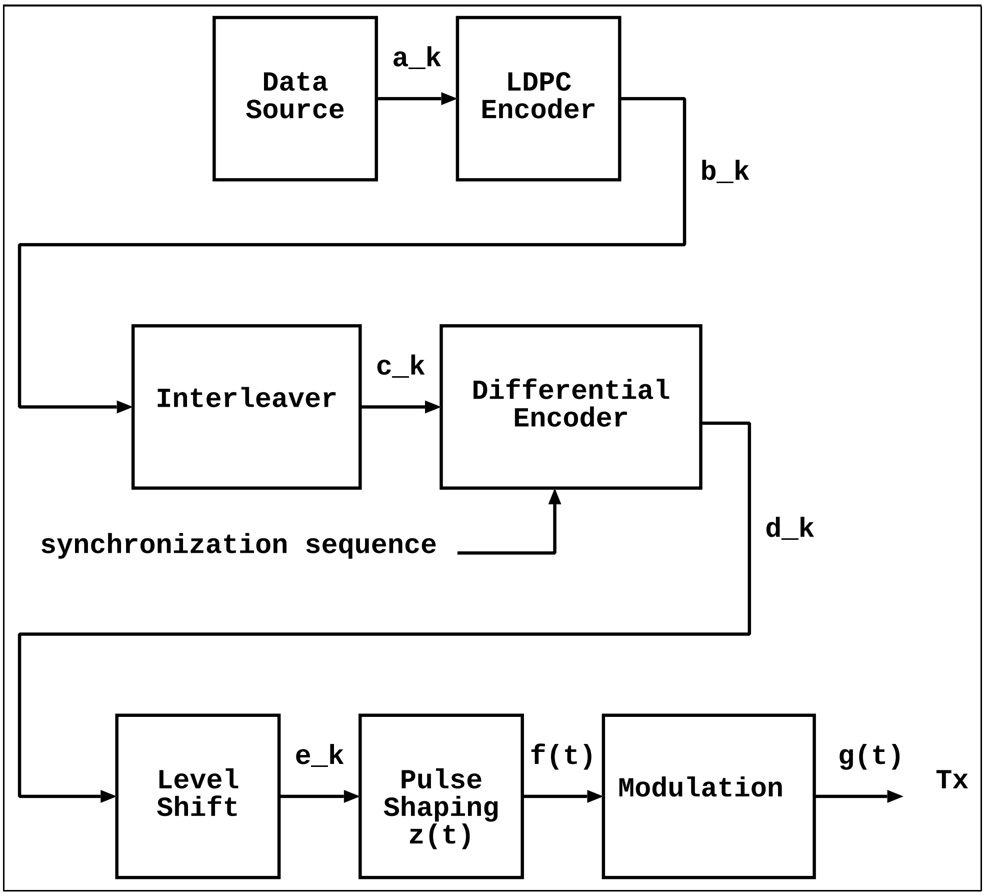

Figure 2.2: Transmitter overview block diagram

where c_k is the encoded and interleaved bit, d_k is the current output and d_{k-1} is the previous output. This differential encoding allows the data to be decoded by observing the difference in phase between consecutive symbols after modulation and also creates a trellis structure which will be exploited in the soft demodulation scheme which utilizes the BCJR algorithm. To illustrate the trellis nature of the differential encoding scheme, consider d_{k-1} as the state of the trellis code. A table of input, output and state can be constructed and is shown in table 2.1.

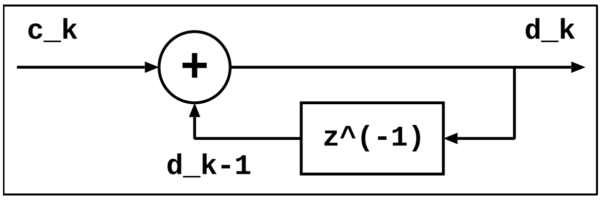

Figure 2.3: Block diagram of differential encoder

c_k input	d_{k-1} current state	d_k next state
0	0	0
1	0	1
0	1	1
1	1	0

Table 2.1: This table illustrates the relationship between the differential encoding and a convolutional code where an input and a state determine an output.

This table can then be expanded into a state transition trellis as shown in figure 2.4. Another advantage to differential encoding is that it is not necessary to coherently demodulate the received symbol stream, which permits the elimination of feedback related phase trackers that can lose synchronization and require time to regain phase lock. A small performance penalty is incurred by using DBPSK over coherently demodulated BPSK. At high SNR, DBPSK requires an extra 1 dB in SNR to perform at a similar BER to coherently demodulated BPSK.

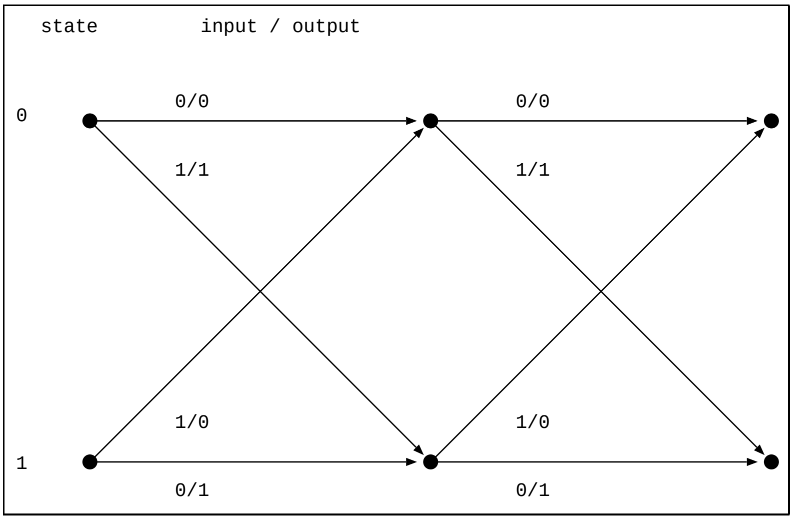

Figure 2.4: Differential encoder state transition trellis

2.4 Pulse Shaping

The differentially encoded sequence is level shifted, and converted into a series of pulses $z(t)$, where $z(t)$ is a half width square pulse. See figure 2.5.

$$z(t) = \begin{cases} 1 & \frac{-T}{4} \leq t \leq \frac{T}{4} \\ 0 & \text{otherwise} \end{cases}$$

The half width square pulse was chosen because it allows for simple symbol synchronization by observing the absolute value at the output of passive phase conjugation. The absolute value is observed over a symbol interval and the peak in magnitude corresponds to the ideal sampling time. The convolution of the square pulse with a time reversed version of itself can be seen in figure 2.6.

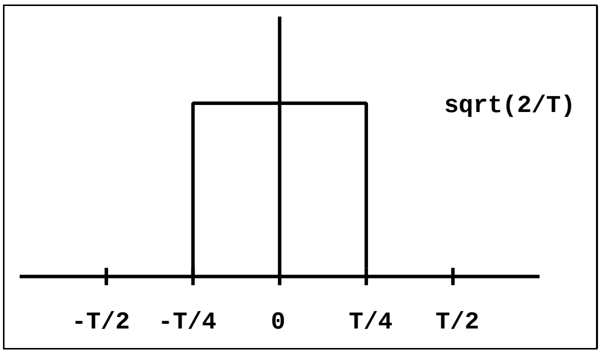

Figure 2.5: Half width square pulse

A series of pulses passed through the matched filter produces figure 2.7, and it can be seen that the peak of the matched filter corresponds to the ideal sampling time. This choice of pulse shape greatly simplifies symbol timing recovery at the receiver at the cost of less energy per bit and a higher bandwidth required to transmit the square pulse shape.

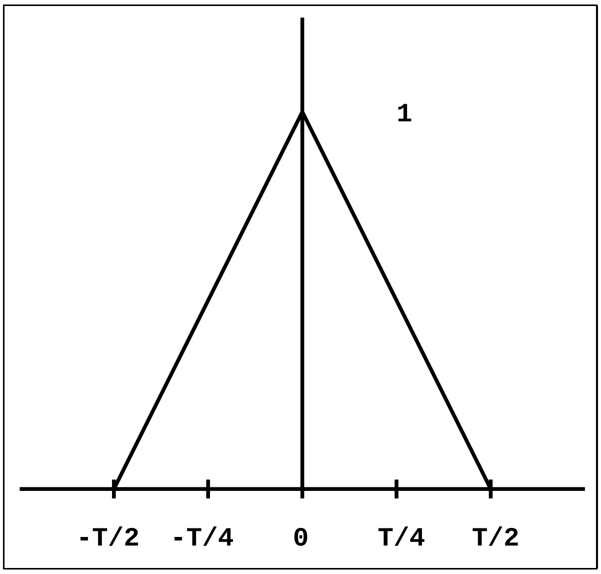

Figure 2.6: Half width square pulse convolved with a time reversed version of itself

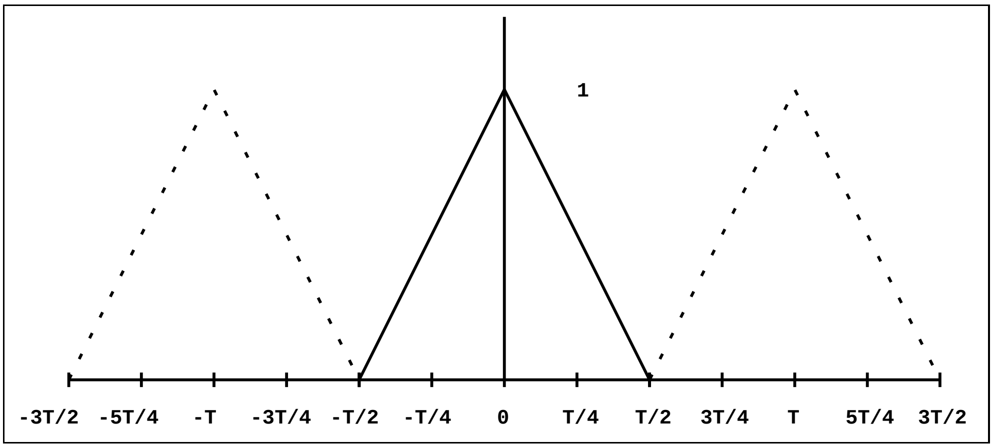

Figure 2.7: A series of multiple symbols after matched filtering

The symbol sequence is transmitted from a hydrophone with a single symbol probe pulse followed by 500 milliseconds of silence. The symbol stream is then transmitted at the full symbol rate. The signal is received by multiple hydrophones spaced more than 1 wavelength apart.

The next sections provide details about the operation of the receiver. The receiver structure can be seen in figure 2.8

2.5 Energy Detection

To extract the probe pulse and data streams from the received waveforms, a probe symbol, $z(t)\cos 2\pi f_0 t$, is cross correlated with each of the received waveforms. The result of the cross correlation is rectified and then integrated over a 2 millisecond period. The result of the integration is summed over all receiving hydrophones for each period. This provided detection of in-band energy at the specified intervals and allows determination of the beginning of the probe and data sequences within 2 milliseconds. An example of the energy analysis can be seen in figure

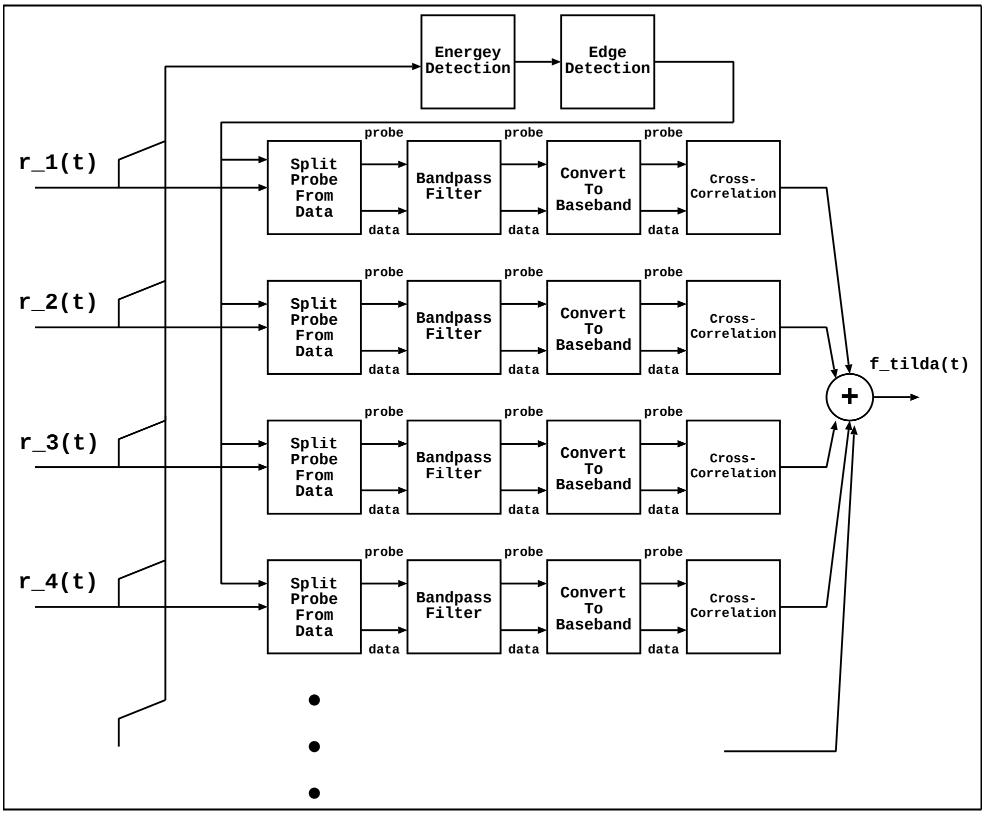

Figure 2.8: Receiver front end processing block diagram

2.9. When the detected energy exceeds a threshold, the data and probe are considered located and the probe symbol and data sequence are extracted and separated. With the energy detection process, it was possible to extract the probe and data sequences when the exact transmission time is unknown or not timed exactly.

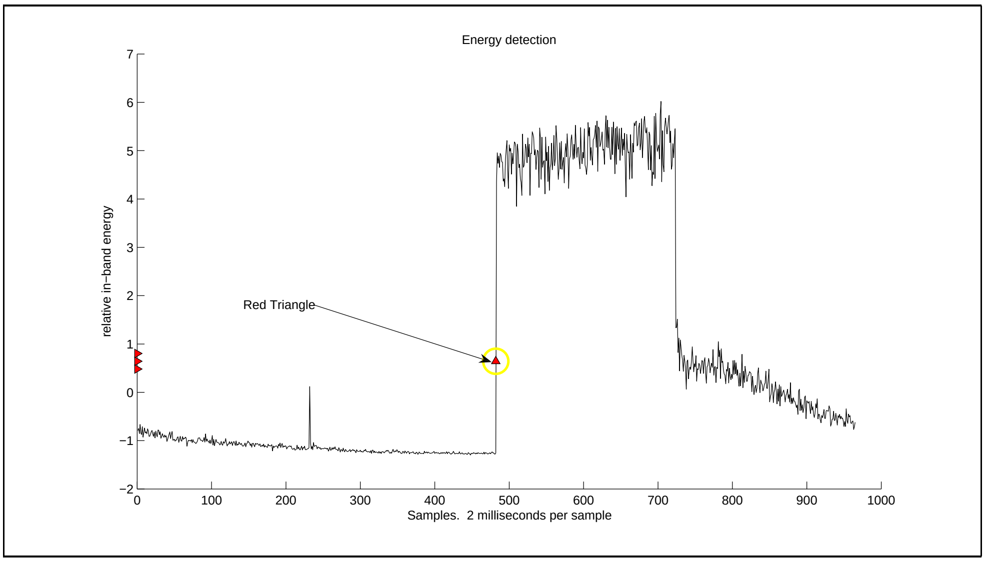

Figure 2.9: Output of energy detection. The figure illustrates the detection of the beginning of symbol block with the red, upward pointing triangle. The probe is seen as the energy spike before the symbol block. In some cases, the probe is too weak to be detected, so its location is always inferred from the location of the data block

An extracted probe sequence can be seen in figure 2.10 and the corresponding data sequence encountered after the probe sequence can be seen in figure 2.11.

2.6 Bandpass Filtering

The received signals are then bandpass filtered with cutoff frequencies of $f_c - 2f_s$ and $f_c + 2f_s$ (where f_c is the carrier frequency and f_s is $\frac{1}{\text{symbol interval}}$). This is done to remove out of band noise. The probe and data sequences shown in figures 2.10

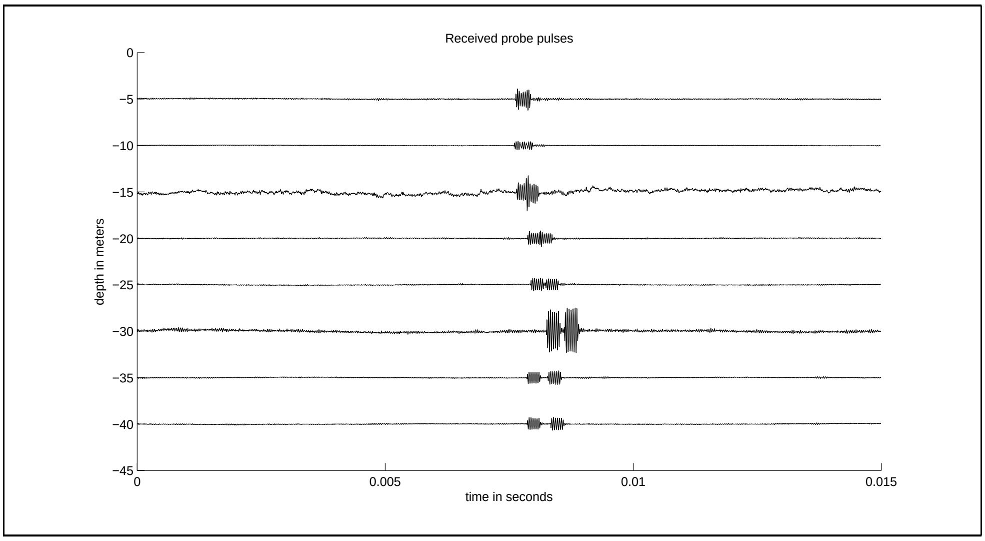

Figure 2.10: An extracted probe sequence from all 8 channels with their respective depths. Notice the multiple received symbols and the phase delay encountered across all the channels

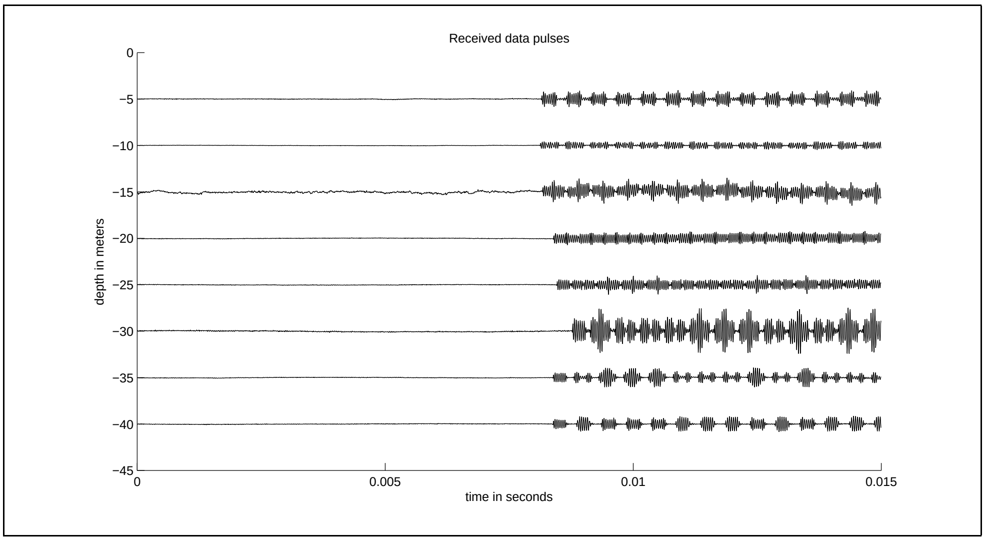

Figure 2.11: An extracted data sequence from all 8 channels with their respective depths. The intersymbol interference effects can be clearly seen at the 35 and 40 meter depths

and 2.11 can be seen after bandpass filtering in figures 2.12 and 2.13 respectively.

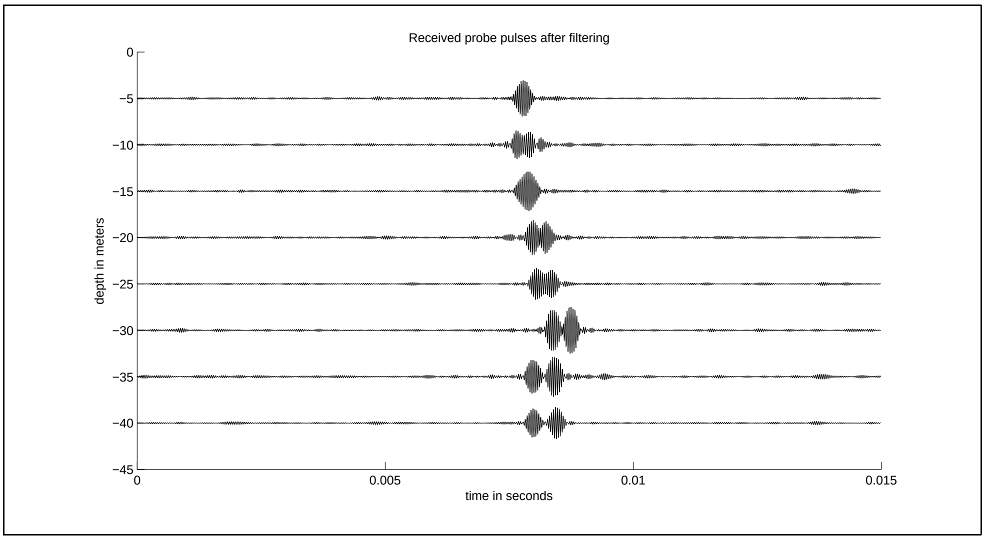

Figure 2.12: An extracted probe sequence from all 8 channels with their respective depths after bandpass filtering

2.7 Conversion to Baseband

In order to minimize complexity and speed up processing, the extracted probe and data waveforms are each converted to baseband using the Hilbert transform shown in equations (2.2) and (2.3),

$$s^{+}(t) = s(t) + \frac{j}{\pi}\int_{-\infty}^{\infty} \frac{s(\tau)}{t-\tau}\,\mathrm{d}\tau \tag{2.2}$$

$$\tilde{s}(t) = s^{+}(t)e^{-j2\pi f_0 t} \tag{2.3}$$

where $s^{+}(t)$ is the right sided spectrum or pre-envelope of $s(t)$ and $\tilde{s}(t)$ is the baseband signal or complex envelope. To convert the data to baseband, the received signals were passed through MATLAB's Hilbert() function which produced the

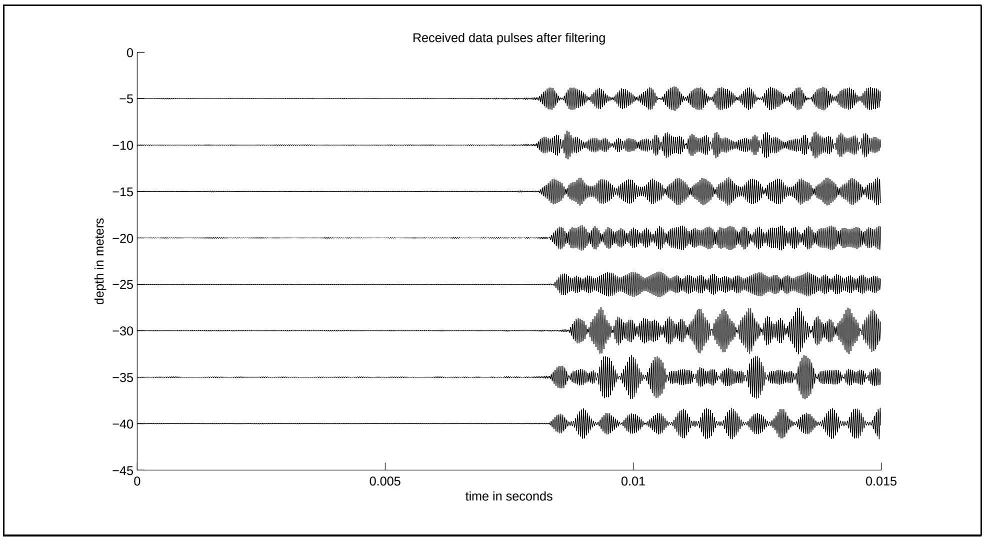

Figure 2.13: An extracted data sequence from all 8 channels with their respective depths after bandpass filtering

pre-envelope of the probe symbol and the symbol sequence. A vector of times and samples of the complex signal $e^{-j2\pi f_0 t}$ is then produced and multiplied by the pre-envelope to convert the signals to baseband. All signal processing after this point is done at baseband.

2.8 Passive Phase Conjugation

Passive phase conjugation serves to align all signals in time and remove ISI. The baseband probe signals are cross correlated with the received symbol sequence for each hydrophone channel. This produces main peaks that align temporally, and undesired temporal side lobes that are different for each channel. The results are then summed across all hydrophones, with all the main peaks contributing to the received signal and all temporal side lobes occurring at different times, resulting in an averaging that minimizes the effects of the different temporal side lobes. The resulting signal has significantly reduced ISI. Further explanation of the process can be found in [9].

The probe sequence in figure 2.12 was cross correlated with the data sequence in figure 2.13 and the result can be seen in figure 2.14. An expanded view of the beginning symbols can be seen in figure 2.15. Note that the symbol peaks are now more aligned in time. These aligned sequences are then summed together to minimize the temporal sidelobes and to reinforce the peaks. The result can be seen in figure 2.16.

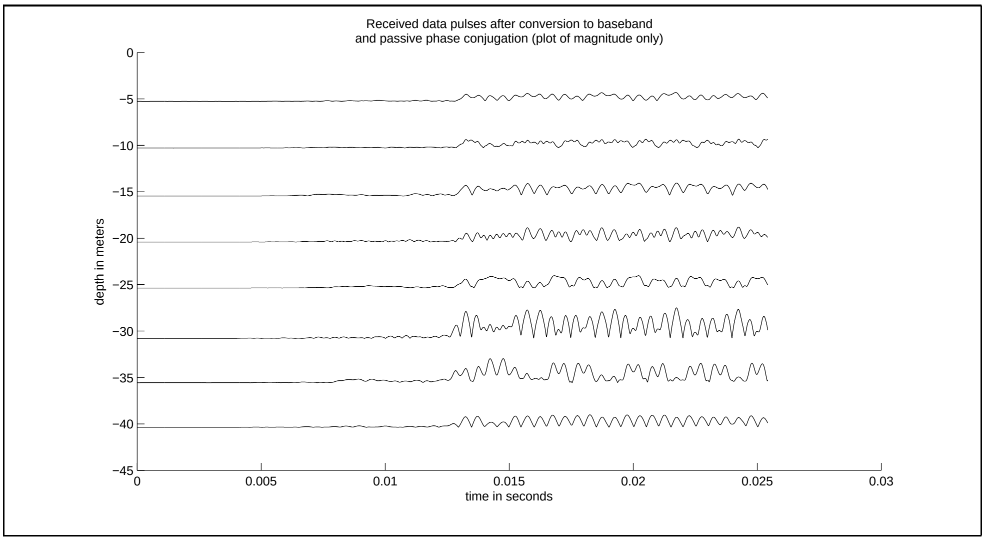

Figure 2.14: Effects of passive phase conjugation on each channel. Note that the symbols are aligned in time

2.9 First Estimate of Sampling Time

After passive phase conjugation, the signal is examined over multiple symbol intervals. The absolute value of the signal is used to determine the peak in the magnitude corresponding to the output of the matched filter (passive phase conju-

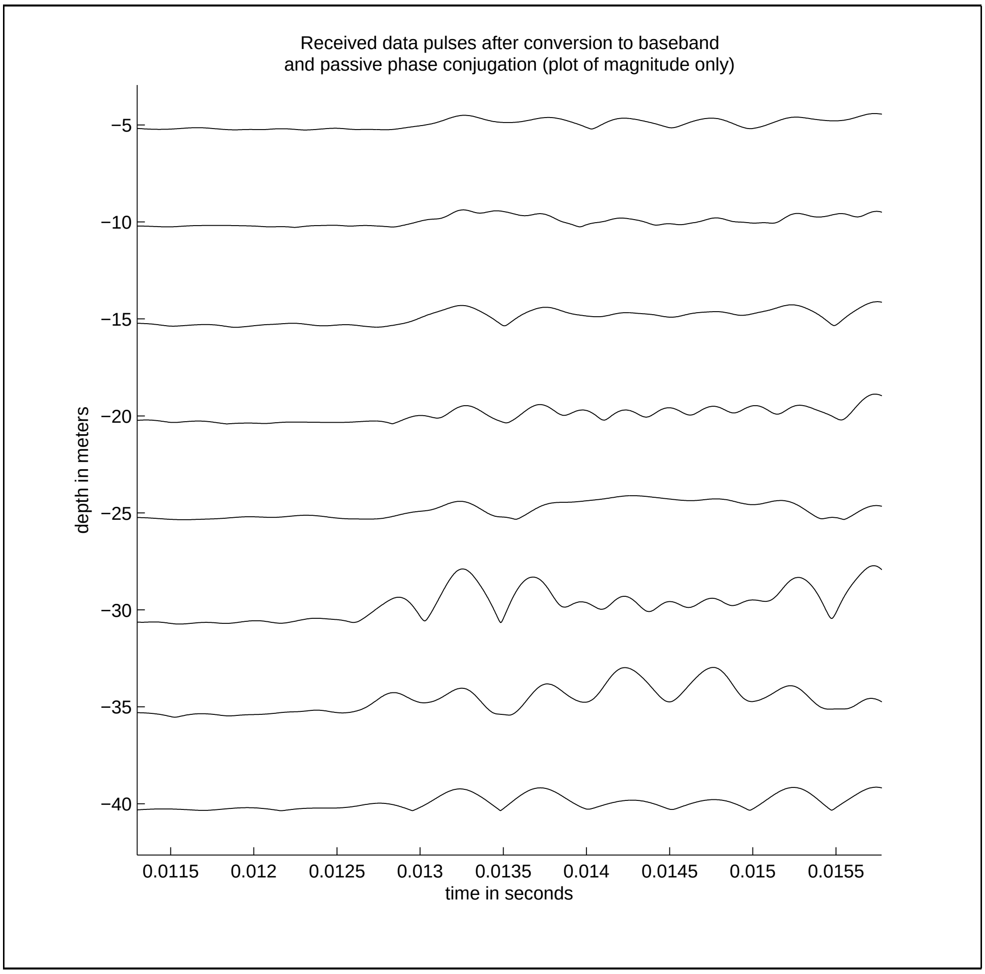

Figure 2.15: Effects of passive phase conjugation on each channel in an expanded view. Note that the symbols are aligned in time

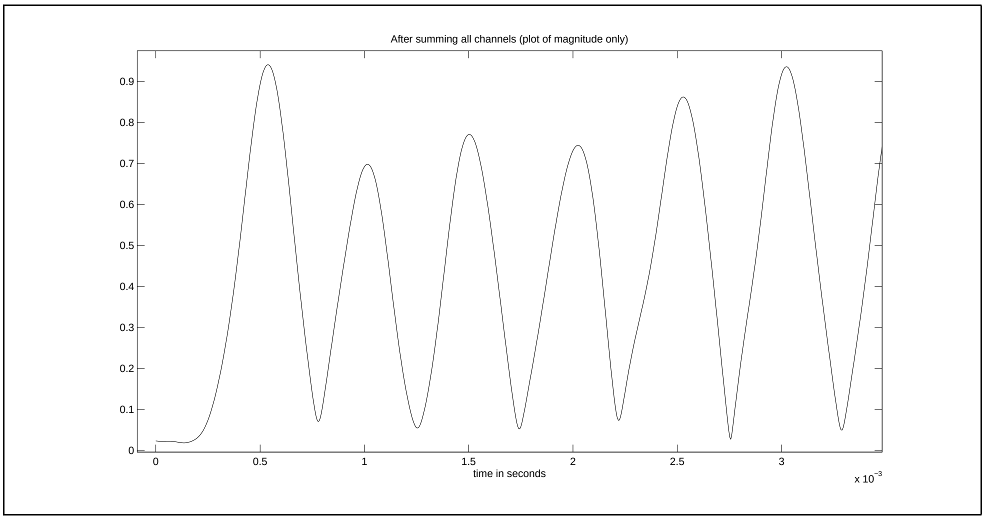

Figure 2.16: Passive phase conjugation after summing across all channels. Notice that there are clearly defined peaks at the ideal sampling time and that ISI has been minimized

gation) which maximizes the SNR. The window is shifted until a mean maximum amplitude occurs at roughly the center of the symbol interval. Symbols without a clearly defined maximum are ignored when determining a good sampling time.

The symbol sequence shown in fig 2.16 is examined for the ideal sampling time and is shown in figure 2.17

2.10 Adaptive Sampling

Once a good estimate of the ideal sampling time is determined, an adaptive sampling system tracks gradual shifts in symbol intervals and adapts the sampling time to compensate.

The averaged sampling time is used to determine a symbol window. The received signal is observed within this window, and the location of the magnitude peak within the window is recorded. The symbol is sampled at the peak if the

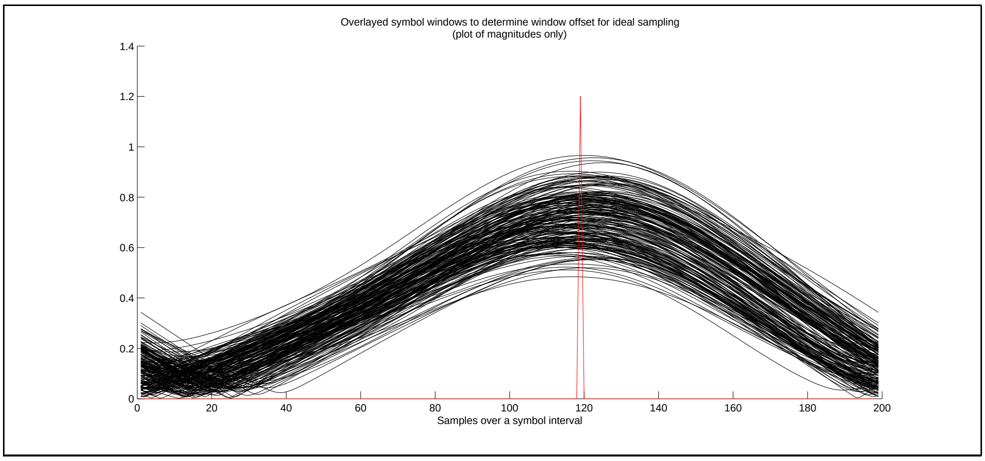

Figure 2.17: The symbols windows are overlaid using a fixed symbol rate and the mean of the position where the magnitude peak occurs is used as the first estimate of the sampling time

peak does not deviate too much from the estimated ideal sampling time, otherwise it is sampled at the estimated ideal time. The offset from the expected sampling time is accumulated in an integrating fashion, and if the accumulated shift is large enough, the estimated ideal sampling time is updated to track this gradual shift.

2.11 Locating the Synchronization Sequence

A critical step to ensure the LDPC decoder can correctly decode the symbols is knowing the proper bit positions. In case the first few symbols are missed, or the energy detector triggers incorrectly, properly locating the synchronization sequence can correct for this error.

In order to align these bit positions, the first 200 symbols of the received symbol sequence are decoded using a hard decision decoder. This decoder observes the difference in phase change between consecutive symbols and generates an estimate of the bit based on a threshold. A phase shift of $180°$ between symbols represents a ‘1’, while no change represents a ‘0’.

The known synchronization sequence is then shifted along the hard decision demodulated bits looking for a close match. The iterative demodulator and decoder could not converge on a solution when more than 1 in 7 bits was in error, therefore, if the synchronization sequence was matched with less than 1 in 7 bits in error, then the system determined that the synchronization sequence was found.

The symbols are then shifted to align them properly for the LDPC decoder before being sent to the BCJR demodulator. If the symbol sequence is missing a few of the first symbols, then the sequence is padded with zeros. If the sequence begins too early, the extraneous symbols are removed from the front of the symbol stream.

The system allows a maximum left shift of the sync sequence length divided by 2 (20 symbols in this case since the sync sequence is 40 bits long). In other words, the system will try to locate the synchronization sequence with up to 20 of the first symbols missing (received symbol sequence missing the first X symbols). The maximum right shift allowed is two times the synchronization sequence length (80 symbols in this case). In other words, the energy detection scheme can trigger as many as 80 symbols too early, and the sync sequence will still be located. Once the symbol sequence is properly aligned, the symbols are sent to the channel estimation block.

2.12 Squaring Received Symbols for Channel Phase Estimation

The BCJR algorithm needs estimates of the channel phase to properly demodulate the received complex points. A block diagram of the channel phase estimator can be seen in figure 2.18.

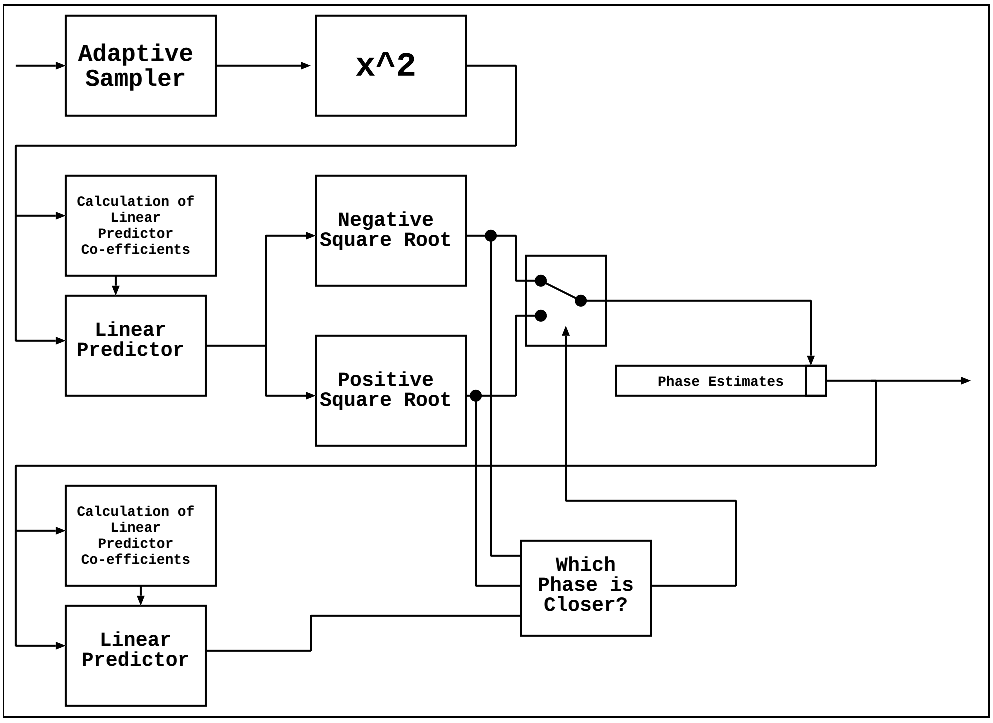

Figure 2.18: Channel estimator block diagram

The sampled complex values are squared to remove the data from the signal and leave only the channel phase. To illustrate how squaring the received points removes the data dependence, consider a symbol encoded with either 0° x_a or 180° phase x_b. The received symbol will be either r_{x_a} or r_{x_b} as shown in equation (2.4), corrupted by the multiplicative channel phase θ (ignoring additive white Gaussian noise).

$$\begin{aligned} r_{x_a} &= Be^{j0}e^{j\theta} \\ r_{x_a} &= Be^{j\pi}e^{j\theta}. \end{aligned} \tag{2.4}$$

Through algebraic manipulation

$$\begin{aligned} r_{x_a} &= Be^{j(0+\theta)} \\ r_{x_a} &= Be^{j(\pi+\theta)}. \end{aligned}$$

The received symbol is then squared

$$\begin{aligned} r_{x_a}^2 &= B^2e^{2j(0+\theta)} \\ r_{x_a}^2 &= B^2e^{2j(\pi+\theta)}. \end{aligned}$$

Distribute the 2 and split the exponentials, to get

$$\begin{aligned} r_{x_a}^2 &= B^2e^{j0}e^{j2\theta} \\ r_{x_a}^2 &= B^2e^{j2\pi}e^{j2\theta}, \end{aligned}$$

which is equivalent to

$$\begin{aligned} r_{x_a}^2 &= B^2e^{j2\theta} \\ r_{x_a}^2 &= B^2e^{j2\theta}. \end{aligned}$$

Through the manipulations it is clear that the 0 and 180 degree phase dependence is completely removed and all phase remaining is due to the channel.

It is important to note that the phase from the channel is doubled. Figure 2.19 illustrates the process. The points were generated by taking encoded bits, level shifting them, subjecting them to AWGN and a random, slowly-varying channel phase. The black points represent received symbols and the red points indicate the underlying channel phase introduced.

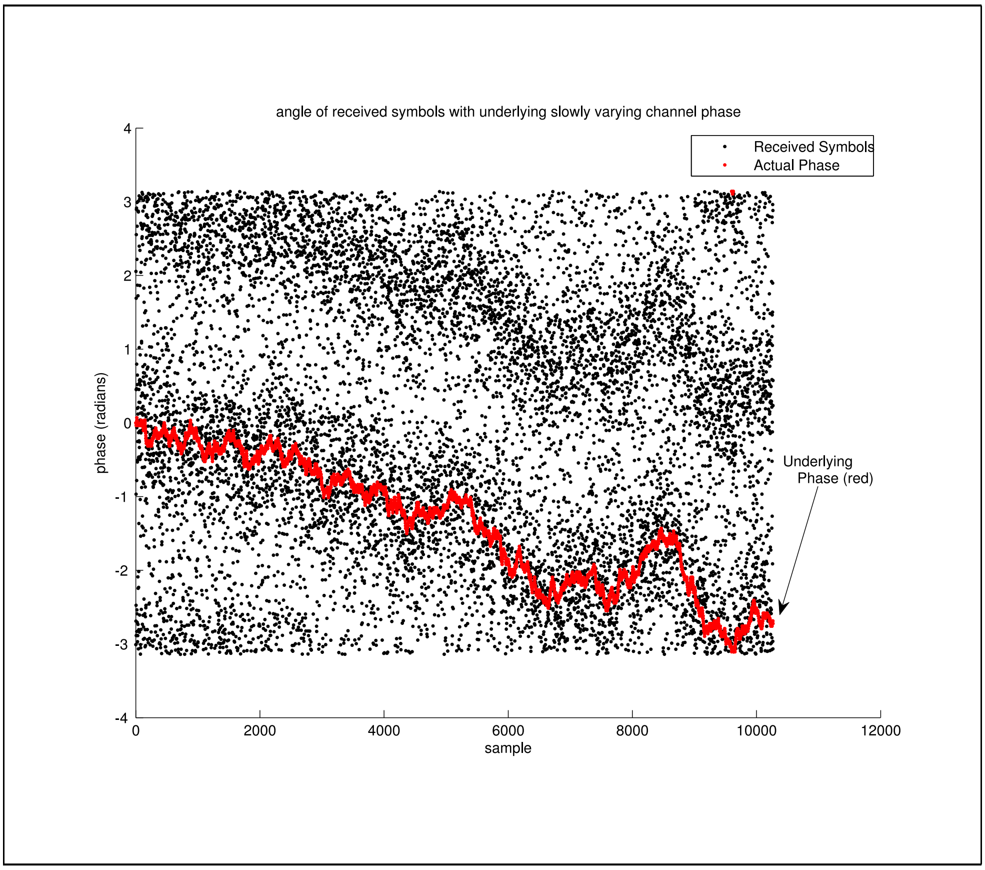

Figure 2.19: Angle of received complex points with underlying slowly varying channel phase

Figure 2.20 shows the result of squaring each symbol. The black points represent the received symbols squared, and the red points are the actual channel phase points squared.

2.13 Linear Estimation of Channel Phase

A linear estimator is constructed based on the squared samples. The estimator is used to track the slowly varying channel phase. The linear predictor allows for a continued projection of the sequence based on past received symbols.

We define the linear prediction of sequence $x(n)$ using the past P symbols (where P is the desired predictor order) and predictor co-efficients by equation (2.5).

$$\hat{x}(n) = \sum_{i=1}^{P} a_i x(n-i) \tag{2.5}$$

To determine the optimum predictor coefficients $\underline{a}$ where $\underline{a}^T = [a_1, a_2, \ldots, a_P]$, we use the method described by Makhoul [10]. First we determine the autocorrelation $\underline{r}$ of the received symbols using

$$r(k) = \sum_{n=0}^{N-1} x(n)x(n-k) \tag{2.6}$$

where N is the number of elements in the sequence $x(n)$. We then construct a matrix and vectors with the auto correlation values as defined by equations (2.7) and (2.8)

$$\underline{r}^T = \Big[r(1), r(2), \ldots, r(p) \Big] \tag{2.7}$$

$$R = \begin{bmatrix} r(0) & r(1) & r(2) & \cdots & r(P-1) \\ r(1) & r(0) & r(1) & \cdots & r(P-2) \\ r(2) & r(1) & r(0) & \cdots & r(P-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r(P-1) & r(P-2) & r(P-3) & \cdots & r(0) \end{bmatrix}. \tag{2.8}$$

Since we wish to determine the optimum coefficients to minimize the mean squared error between our received sequence and the predicted output, we need to solve equation (2.9), provided R is invertible.

$$\underline{a}^{opt} = R^{-1}\underline{r} \tag{2.9}$$

Once the prediction coefficients have been determined, a sequence of predictions based on past received symbols helps to expose the underlying slowly varying channel phase. Linear predictors are constructed separately for the real and imaginary parts of the received symbol sequence. Using the symbol sequence given in figure 2.20, a linear predictor is generated based on these squared points, and can be seen in figure 2.20 as the yellow points. For all simulations and in water tests, a predictor order of $P = 20$ was used.

2.14 Phase Estimation and Unwrapping

To remove the doubling of the angle introduced by the squaring of each value, the square root of each estimated symbol is taken. This creates a $180°$ phase ambiguity. To unwrap the phase and determine channel estimates, another linear estimator is constructed which projects the next phase estimation point into one

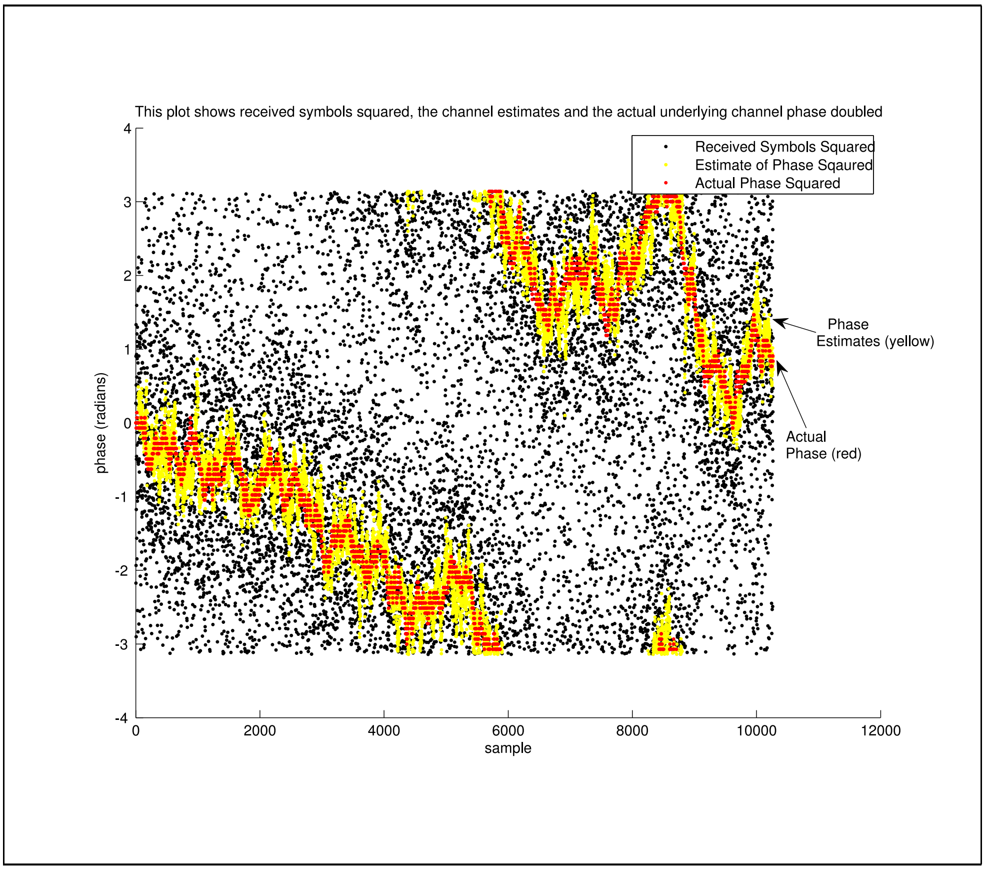

Figure 2.20: Angle of received complex points squared, channel phase squared and output of linear predictor

of the two regions $\left[\frac{\pi}{2}, \frac{-\pi}{2}\right]$ or $\left[\frac{-\pi}{2}, \frac{\pi}{2}\right]$. The two candidate points are compared to the projected point, and the point that is closer is chosen. The linear predictor is then updated using the chosen point.

The square root of each linear predictor point is computed, and shown in figure 2.21.

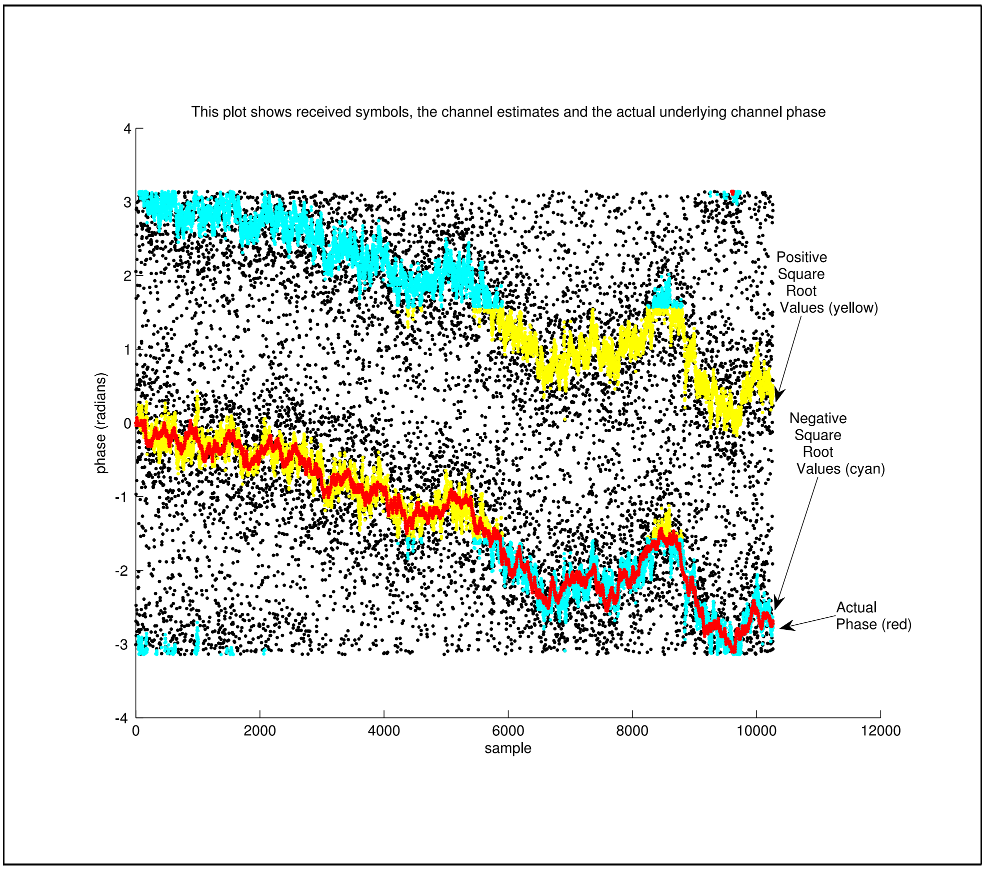

Figure 2.21: Angle of received complex points with positive and negative results of the square root. The underlying channel phase is also shown.

There is a phase unwrapping problem which must be solved in order to make use of these points. The yellow points represent the positive result of the square root operation, and the cyan points represent the negative result of the square root operation. The system begins by knowing the phase of the first few symbols (through the synchronization sequence) and then generates another linear predictor to project phase points into one of the two realms. The point that is closer to the projected point is chosen and the linear predictor is updated with the newly chosen point. This process is applied to all the symbols, resulting in an estimate of the channel phase. Figure 2.22 shows the completed channel phase estimate in yellow with the actual channel phase in red.

The linear predictor provides a sufficient estimate of the channel phase, but still suffers from $180°$ phase slips at low SNR. These phase slips would severely impair the performance of a BPSK system which requires that the phase estimate remain aligned to the actual channel phase to properly demodulate the information. The DBPSK modulation scheme can tolerate the phase slips with only a few bits received in error during the actual phase slip, and will recover and continue to demodulate correctly even with the channel estimates $180°$ out of phase.

2.15 BCJR Algorithm

The unwrapped phase linear predictor outputs are then fed to a BCJR algorithm to correct for the phase shift introduced by the channel. The BCJR algorithm, developed by Bahl, Cocke, Jelinek and Raviv [11], is a maximum a posteriori decoder for codes defined on a trellis. This recursive algorithm computes metrics for trellis transitions and trellis states. From these metrics, a log likelihood ratio (LLR) (2.10) is produced for each symbol.

$$L_a = \frac{log\big(p(u_l = +1)\big)}{log\big(p(u_l = -1)\big)} \tag{2.10}$$

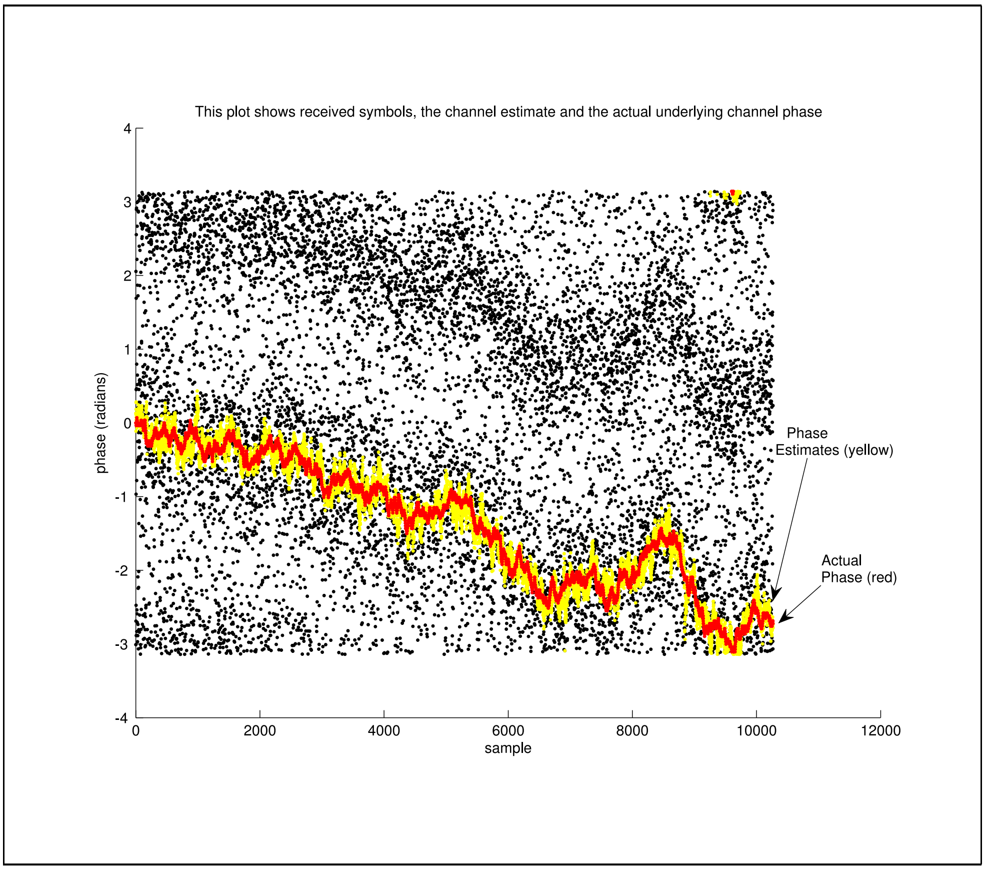

Figure 2.22: Angle of received complex points, underlying channel phase and completed channel estimates

This particular version of the BCJR algorithm was presented by [12]. It produces these metrics by first computing the transition metrics $\gamma^*(s', s)$ using equation (2.11).

$$\gamma^*(s', s) \equiv \begin{cases} u_l \frac{L_a(u_l)}{2} + \frac{L_c}{2}\mathbf{r}_l \cdot \mathbf{v}_l & l = 0, 1, \cdots, h-1 \\ \frac{L_c}{2}\mathbf{r}_l \cdot \mathbf{v}_l & l = h, h+1, \cdots, K-1 \end{cases} \tag{2.11}$$

The state transition is (s', s), where s' is the originating state and s is the next state. u_l represents the input symbol at time l. L_a is the extrinsic information provided as LLRs from an outside source (the LDPC decoder). For the first time through the loop, the L_a values are initialized to 0 when all source bits are equally likely. L_c is the channel reliability factor and can be calculated using equation (2.12)

$$L_c = 4\frac{E_s}{N_0} \tag{2.12}$$

and $\mathbf{r}_l \cdot \mathbf{v}_l$ represents the dot product between the received symbol $\mathbf{r}_l$ and the trellis output $\mathbf{v}_l$. In order to provide increased performance, the trellis can be terminated by $h-K$ bits after h bits have been encoded, where h is the number of information bits, and K is the total length of the symbol sequence.

At each state, the α and β metrics can be computed using equations

$$\alpha^*_{l+1}(s) = max^*_{s' \in \sigma_l}\left(\gamma^*(s', s) + \alpha^*_l(s')\right) \quad l = 0, 1, \ldots, K-1, \tag{2.13}$$

$$\beta^*_l(s') = max^*_{s' \in \sigma_l + 1}\left(\gamma^*(s', s) + \beta^*_{l+1}(s')\right) \quad l = K-1, K-2, \ldots, 0, \tag{2.14}$$

where $max^*(x,y)$ is defined by

$$max^*(x,y) = max(x,y) + ln(1+e^{-|x-y|}). \tag{2.15}$$

The initial and final conditions of the trellis are accounted for by using equation (2.16) for the α metrics at time $l = 0$, and equation (2.17) for the β metrics at time $l = K$.

$$\alpha_0^* \equiv \begin{cases} 0 & s = \text{initial state} \\ -\infty & \neq \text{initial state} \end{cases} \tag{2.16}$$

$$\beta_K^* \equiv \begin{cases} 0 & s = \text{ending state} \\ -\infty & \neq \text{ending state} \end{cases} \tag{2.17}$$

If the code is not tailed (i.e. $h - K = 0$), then the values in equation (2.17) would all be set to $\frac{1}{\text{number of states}}$. After computing all the α, β and γ metrics, the log likelihood ratios can be calculated using equation (2.18).

$$\begin{aligned} L(u_l) =& max^*_{(s',s)\in\sum_l^+} \left[\beta_{l+1}^*(s) + \gamma_l^*(s',s) + \alpha_l^*(s')\right] - \\ & max^*_{(s',s)\in\sum_l^-} \left[\beta_{l+1}^*(s) + \gamma_l^*(s',s) + \alpha_l^*(s')\right] \end{aligned} \tag{2.18}$$

We modified the metric by adopting the metric proposed by Lodge for a MAP demodulator [8] to accept the channel prediction from the linear predictor. The change is shown in equation (2.19)

$$\gamma^*(s',s) \equiv \begin{cases} u_l \frac{L_a(u_l)}{2} + \frac{L_c}{2}\Re\left(e^{-j\tilde{\theta}}\mathbf{r}_l \cdot \mathbf{v}_l\right) & l = 0, 1, \cdots, h-1 \\ \frac{L_c}{2}\mathbf{r}_l \cdot \mathbf{v}_l & l = h, h+1, \cdots, K-1 \end{cases} \tag{2.19}$$

where $\tilde{\theta}$ is the channel prediction. The idea is that the conjugate of the channel phase will move each point toward being purely real and remove the effect of the channel phase.

2.16 Iterative Demodulation and Decoding

The BCJR algorithm determines LLRs for each received symbol. On the first iteration, the LLRs are de-interleaved, and sent directly to the LDPC decoder. The LDPC decoder program reads in the LDPC encoding matrix and the LLRs from the BCJR algorithm and iterates on those values for a fixed number of iterations. The input LLRs from the BCJR algorithm are then subtracted from the output LLRs of the LDPC decoder to provide extrinsic information and the result is re-interleaved and fed back to the BCJR algorithm. This repeats until the LLRs from the BCJR and LDPC decoder do not change from one iteration to the next, or until a maximum number of iterations has been reached. The information bits are then recovered from threshold decoding of the LLRs. A block diagram of this process is shown in figure 2.23.

Experiments determined that 5 iterations of the LDPC code for every outer iteration on the BCJR demodulator and LDPC decoder provide the best performance. The loop terminates after a fixed number of iterations or after the LLRs from both the LDPC decoder and BCJR algorithm have not changed in two iterations.

A comparison of the methods presented above to the demodulator scheme described by Hoeher and Lodge in [8] illustrates the improvement in computational complexity provided by the linear predictor. The algorithm described in [8] re-

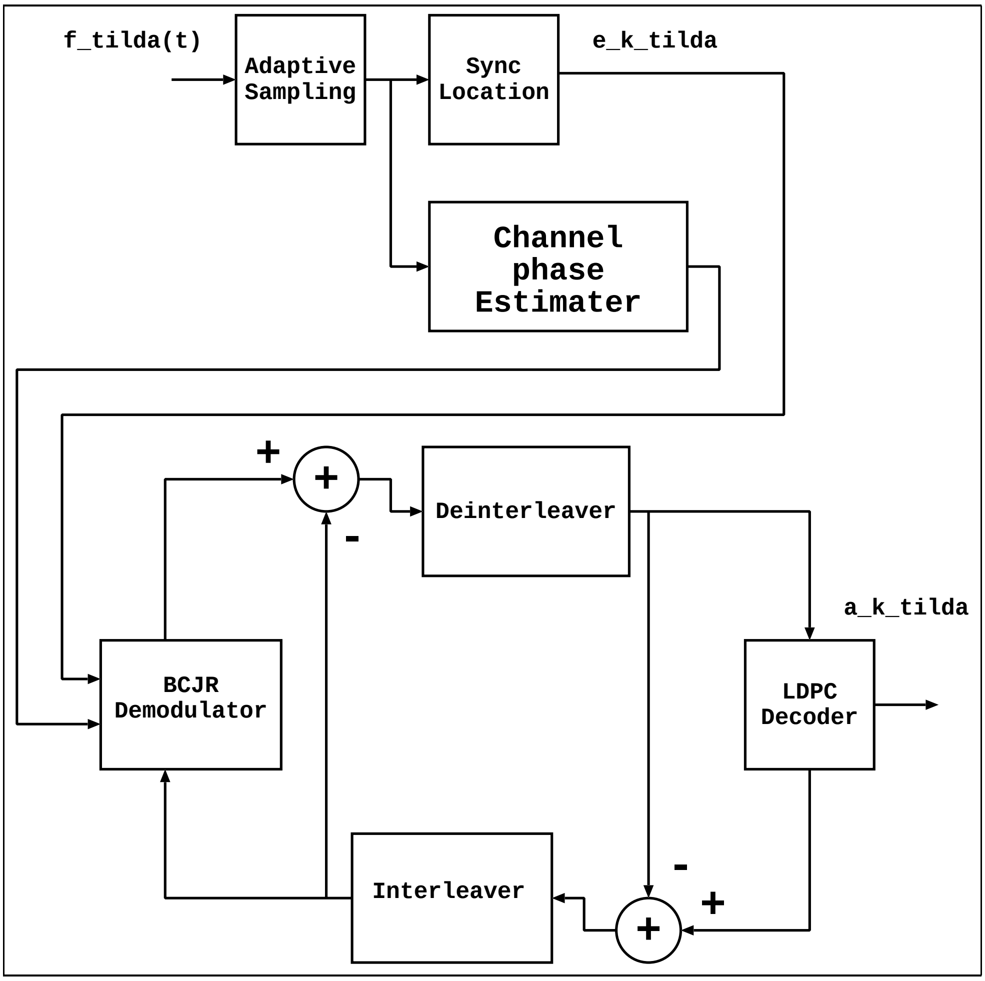

Figure 2.23: Iterative demodulation and decoding block diagram

quires that the trellis be expanded to M^N states with M branches per state. M is the number of points in the signal constellation (2 for DBPSK) and N is the desired phase predictor order. For a 20th order predictor, the algorithm as proposed requires 2^{20} or $1.048576e6$ states. This quickly becomes unwieldy for larger constellations and higher predictor orders. In our method, prediction of the channel phase is done once, and will not slow down the BCJR algorithm once completed. This is in direct contrast to exponentially increasing the computational complexity in the form of added states, which will affect performance on every pass of the BCJR algorithm.

Chapter 3

SIMULATIONS AND IN-WATER TESTS

3.1 Simulations

A simulation program was written in MATLAB to verify the performance of the iterative demodulation and decoding system. The system was designed to simulate the performance of the iterative demodulator / decoder and channel phase estimation after passive phase conjugation. The simulation generated random data and encoded it with an IRA LDPC code. The LDPC bits were randomly interleaved, differentially encoded and the data was tailed with a single bit to end the trellis in a known state. The differentially encoded bits were level shifted and subjected to a slowly varying and random phase change plus complex additive white Gaussian noise

$$Y = X * e^{j\theta(t)} + N \tag{3.1}$$

with zero mean and variance calculated using

$$\sigma^2 = \frac{1}{10^{\frac{\text{SNR}_{\text{dB}}}{10}}}. \tag{3.2}$$

$\theta(t)$ was created by integrating a random value which could take on values of $\Delta\theta$, $-\Delta\theta$ and 0. For the simulations, a $\Delta\theta$ was set to $2°$.

The received symbols are passed through the iterative demodulation and decoding system and the results are tabulated at different SNR. Two different code sizes are used with and without channel estimation. The plots of the bit error rate (BER) versus SNR are shown in figure 3.1. It is interesting to note that the performance of the system for the larger block size performed within $\frac{1}{10}$ of a dB of the system with perfect channel knowledge.

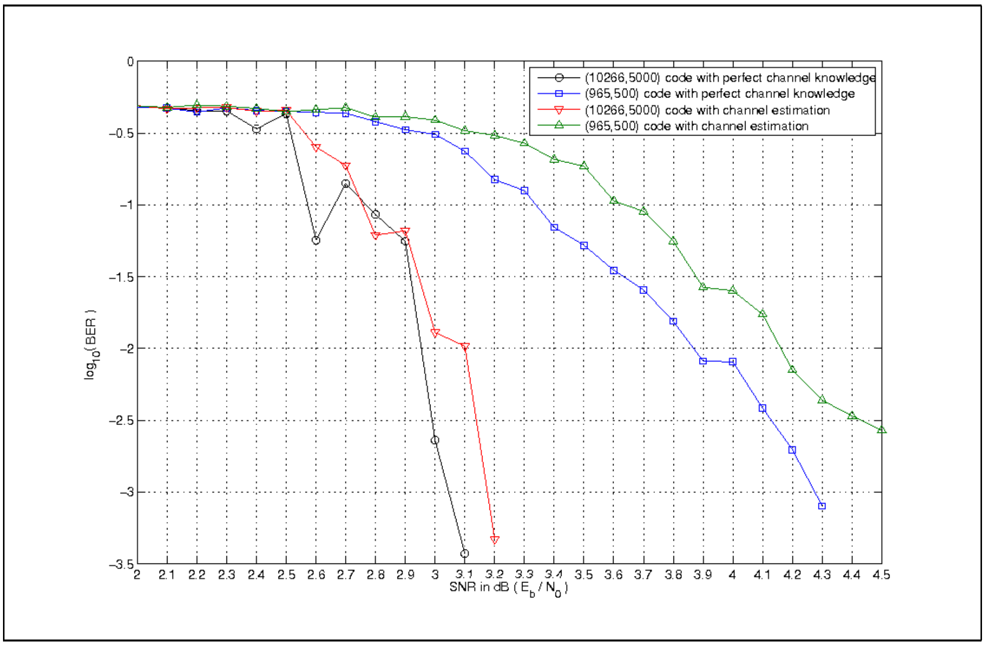

Figure 3.1: Simulated performance curves for two different block sizes with and without channel estimation

According to tests done by Hoeher and Lodge, tailing the code had a minimal impact on the performance of the system [8] and was not done for our in-water tests.

To simulate relative motion between the source and receiver, a linear phase change of $15°$ was used as the underlying phase. The system was simulated at an SNR of 3.5 dB. The received symbols are shown in figure 3.2 and a close up view of the channel phase estimates are shown in figure 3.3. There is a $180°$ phase shift between the actual channel phase and the estimated phase due to a phase slip earlier in the sequence. DBPSK is affected only during the phase slip, and is able to properly demodulate the remaining bits, even with the $180°$ phase error. With a carrier frequency of 25 Khz and a symbol rate of 5,000 symbols per second, this equates to a relative motion of 12.5 meters per second. The system corrected all but 8 bits out of 10,266 transmitted bits.

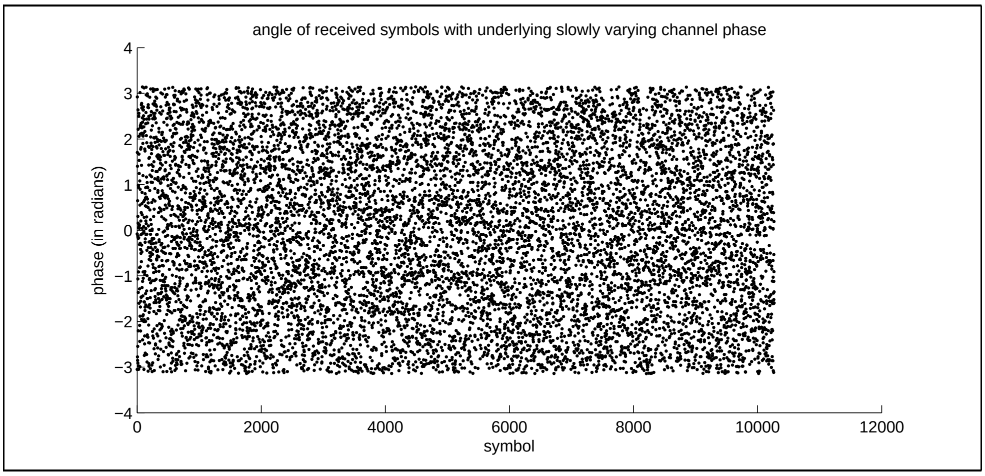

Figure 3.2: Simulated received symbols with a $15°$ linear phase change between symbols at 3.5 dB SNR

3.2 In-Water Testing

Before testing, we decided to use two symbol rates (2,000 and 5,000 symbols per second), two code block sizes (965 and 10,2666 bits) and two carrier frequencies (25 and 50 Khz). The decisions of block size and symbol rates were influenced

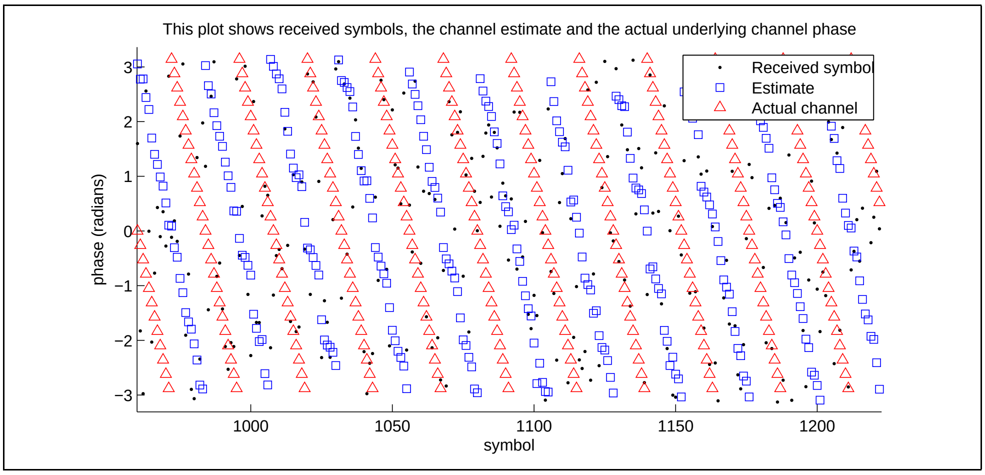

Figure 3.3: An expanded view of the simulated received symbols which shows the received symbols, underlying phase, and channel estimates

in part by a team of researchers from the University of Washington (UW) who had conducted tests using passive phase conjugation and adaptive equalizers in the Puget Sound. One of the effects studied was the amount of time after a probe pulse had been transmitted that passive phase conjugation remains effective. In the worst case scenario experienced by the UW team, a channel coherence time of 0.5 seconds was experienced [5].

The LDPC block sizes chosen were (965,500) and (10266,5000). The longer code provides better error correction performance, while the smaller block size provides lower latency and a higher probability that the block will be transmitted successfully before the channel estimation from passive phase conjugation is no longer acceptable. The information was transmitted using DBPSK yielding 1 bit per symbol. The symbol rates used were 2,000 coded symbols per second and 5,000 coded symbols per second corresponding to approximately 1,000 information bits per second and 2,500 information bits per second respectively. At 2,000 coded symbols per second, the 10,266 symbol block lasts in excess of 5 seconds and in a worst case scenario, the window of channel coherence would have been significantly shorter than the time required to transmit the entire symbol sequence. A final test of 41074 uncoded information bits transmitted at 10,000 symbols per second on a 50 Khz carrier was added to test the performance limits of the passive phase conjugation, channel estimation and BCJR soft demodulation.

Once the code blocks and symbol rates were selected, waveforms were constructed for the entire symbol sequences. These sampled waveforms were constructed using a MATLAB program and the waveforms were sampled at 1 million samples per second. The waveforms were sent to the research facility prior to our arrival for loading onto a waveform generator.

To test the system in the shallow water acoustic channel, we went to the Acoustic Research Detachment at Lake Pend Oreille, Idaho on November 20th, 2007. A small watercraft carried the waveform generator, an amplifier, and an ITC-1032, 25 kHz transmission hydrophone. The source hydrophone was suspended approximately 3 m below the surface. The watercraft was single point anchored during each test. The receiving apparatus was located on a barge anchored at 4 points approximately 1.78 Km or 0.89 Km from the transmitting hydrophone, depending on the test. The receiving array was a vertical hydrophone array consisting of eight ITC-8140 hydrophones each separated by 5 meters with the first hydrophone 5 meters below the surface. In [13] this number of hydrophones was shown to provide a significant increase in performance before limited returns were seen from increasing the number of hydrophones. The hydrophones were mounted onto 20 inch PVC risers and oriented toward the shallow end of the lake to prevent problems with acoustic shadowing caused by the cables. The eight hydrophones were held vertical by a lead weight at the end of the array. The receiving apparatus was suspended from the moon pool of the barge. The eight hydrophone channels were sampled using National Instruments PCI-4452 data acquisition cards and the data was stored on a portable hard drive. The received data was sampled at 196608 samples per second and each channel sample was stored as a signed 16 bit value.

The lake bottom profile, as determined by echosounding, can be seen in figure 3.4. The vertical axis represents depth in meters and the horizontal axis represents time. The echosounder craft traveled between two endpoints (barge and shoreline) at an approximately constant velocity in a straight line. The apparatus was a Simrad EK-60 Split-beam Echosounder. The frequency of the sounder was 120 KHz at 200 watts with a ping duration of 256 milliseconds and a ping rate of 2.5 pings per second. The top trace is the actual lake bottom profile and the lower two traces are double and triple bounce paths. The lake bottom is mostly rock and gravel with some silty places (located by places where the third echo return is faint or non-existent). Our test setup is shown in figure 3.5.

Each data set was run at a midway point, approximately .89 km from the barge, and in shallow water, approximately 1.78 km from the barge. Each data set was composed of a probe pulse, followed by 500 milliseconds of silence, then the symbol block. This sequence of probe symbol, 500 milliseconds of silence and symbol block was repeated with 1 second between the end of the block and the next probe symbol over a ten minute period. We did not run all possible combinations of distance, symbol rate, block size and carrier frequency due to time constraints. The data was collected and processed off line. A list of the tests run is shown in table 3.1.

The received waveforms were stored on a computer and used for bit error analysis. Since the waveforms were stored in their raw format (8 channels, 2 bytes

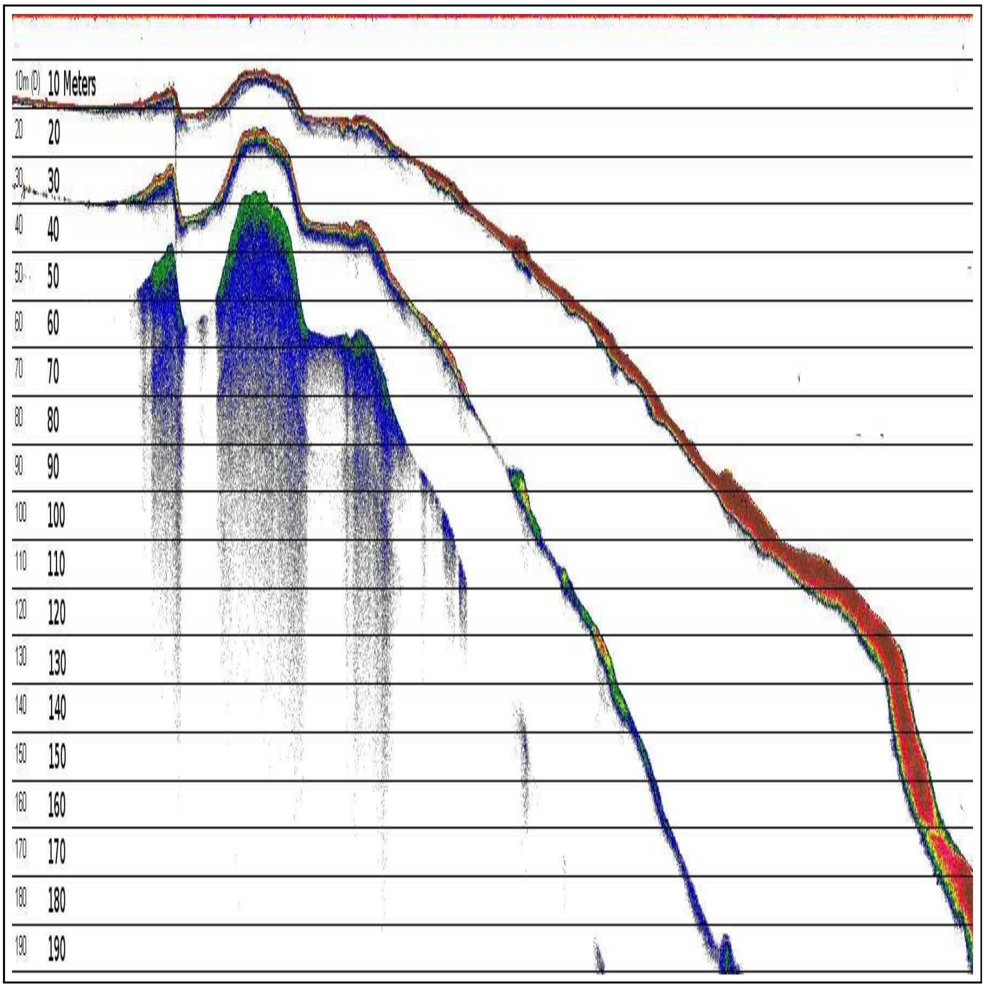

Figure 3.4: Echosounder lake bottom profile

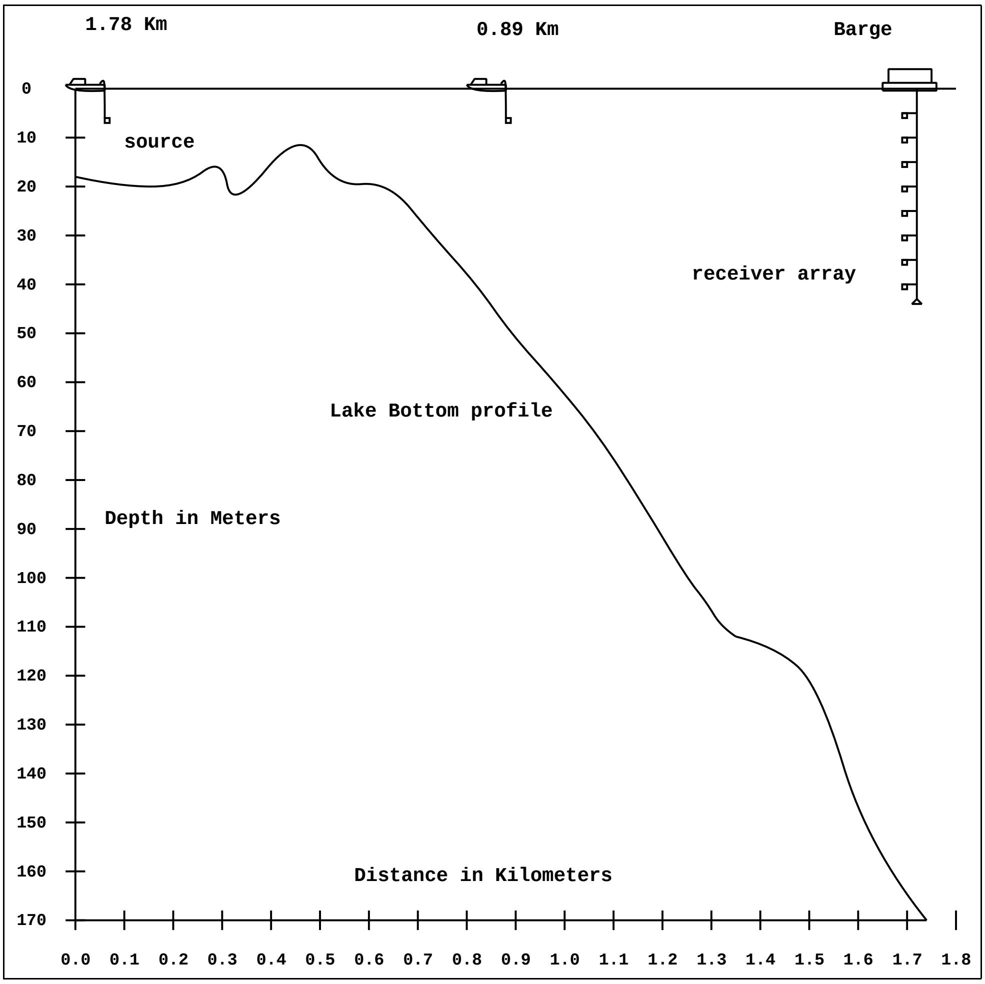

Figure 3.5: Test setup showing barge and source boat positions and relative lake bottom profile

Carrier (Khz)	Symbol Rate (symbols per second)	Block Size (coded bits)	Info Bits	Distance (Km)
25	2,000	965	500	0.89
25	2,000	10,266	5000	0.89
25	5,000	10,266	5000	0.89
25	2,000	965	500	1.78
25	2,000	10,266	5000	1.78
25	5,000	10,266	5000	1.78
50	2,000	965	500	1.78
50	2,000	10,266	5000	1.78
50	5,000	10,266	5000	1.78
50	10,000	41,074	41074	1.78

Table 3.1: This table shows the different tests that were run at Lake Pend Oreille

per channel per sample, 196,608 samples per second, for more than 10 minutes per test) the files were in excess of 1.7 GB and processing of the files was not feasible without parsing them into separate tests. A process of energy detection identical to the one described in chapter 2 was used to analyze the entire file and the result was stored as sort of table of contents for the larger file. To process a test, the energy detection file was analyzed for the specific test, then a function would seek into the large data file and extract the probe and symbol sequences. The probe and symbol sequences would go through all the steps described in chapter 2 and the results would be analyzed by comparing the data transmitted with the data decoded.

To calculate the SNR after passive phase conjugation, the energy in the noise was averaged and recorded during the periods of silence following each probe pulse. This was then subtracted from the average received energy from the symbol sequence. The ratio of received energy minus noise energy and noise energy provided and estimate of the SNR. When passing the SNR to the BCJR and the LDPC decoder, a fixed value of 7 dB was used. This number was chosen based on trial and error.

The results for the tests outlined in 3.1 were compiled and are presented in 3.2. It was discovered during the processing of our data that during two of our in water tests, there were other experiments running in the lake that caused a significant amount of interference in our acoustic band. This in-band noise is responsible for for the wide range of SNRs experienced in some of the tests.

Distance (Km)	Carrier (Khz)	Symbol Rate (symbols per second)	Blocks Sent	Block Size (coded bits)	Max SNR ($\frac{E_b}{N_0}$)	Min SNR ($\frac{E_b}{N_0}$)	$log_{10}(BER)$
0.89	25	2,000	299	965	19.23	18.26	-2.59
0.89	25	2,000	67	10,266	21.93	21.3	-4.66
0.89	25	5,000	142	10,266	20.46	19.63	-5.86
1.78	25	2,000	257	965	14.29	3.45	-2.65
1.78	25	2,000	82	10,266	15.76	13.96	-5.45
1.78	25	5,000	112	10,266	12.82	4.05	-5.02
1.78	50	2,000	277	965	17.8	13.3	-2.65
1.78	50	2,000	68	10,266	18.11	14.36	-5.37
1.78	50	5,000	137	10,266	16.12	10.52	-5.45
1.78	50	10,000	89	41,074	15.04	10.18	-2.03

Table 3.2: This table shows the results of the different tests that were run at Lake Pend Oreille

The results of the tests indicate that the methods described in this thesis are sufficient to provide a high rate communications link in a shallow water. Tests run at 5,000 coded symbols per second, resulting in 2,500 information bits per second, provided bit error rates (BERs) below 10^{-5} which is more that adequate for data transmissions which can tolerate information errors such as images and video. The BERs were calculated using received codewords which implies that even lower BERs would be achieved in practice since some of the errors will occur in the redundant parity bits.

Another exciting result is the feasibility of transmission at 10,000 symbols per second. With a sufficient code, the expected BER rate can be dropped by orders of magnitude, providing a 5,000 information bits per second channel with a low probability of errors.

Chapter 4

CONCLUSION

Passive phase conjugation in conjunction with an iterative soft demodulator and channel decoder has been shown to improve information throughput in the shallow water acoustic channel. Our system was successfully tested with symbol rates up to 5,000 coded symbols per second corresponding to 2,500 information bits per second in water with low bit error rates. The high rate test with no channel code shows that even at these higher symbol rates, the bit errors were sufficiently small to be corrected by a well designed channel code. Error tolerant information (pictures, audio and video) could even be transmitted with minimal channel coding overhead to provide real time information over an acoustic channel.

The system presented in this thesis uses simple methods to achieve high data rates. Our receiver structure can be made very computationally efficient through the use of the log-max algorithm in the BCJR algorithm and the min-sum algorithm in the LDPC decoder algorithm. Other computations required are Toeplitz matrix inversions, finite impulse response (FIR) filter implementations and cross correlations, all of which can be implemented efficiently in digital signal processing (DSP) cores and computer software. This implies that the system could be designed to operate in real time.

The in water tests showed the viability of our system and the simulations show that information throughput gains can be experienced using these simple methods. With some improvements, we believe even higher throughputs could be achieved. Measures to improve the system would include: (i) using full width square pulses or root raised cosine pulses to increase symbol energy at higher symbol rates; (ii) using an interleaver optimized for use with the system ; (iii) increasing the constellation size and implementing differentially encoded M-ary phase shift keying to increase the number of bits per symbol. We also expect that further gains could be experienced if the symbol timing and phase estimation were also part of the soft information loop.

Further gains in throughput can be experienced if multiple frequency channels are used in parallel. Using a raised cosine pulse at the output of the matched filter requires at most a bandwidth equal to the symbol rate for data transmission. With 5,000 symbols per second, the bandwidth required would be 10 Khz around the carrier. If the channels begin at 15 Khz, there would be 5 bands from 15 Khz to 55 Khz with a total throughput of 25,000 symbols per second. Using a higher order modulation scheme like QPSK and a rate 1/2 LDPC code, the information throughput would be 25,000 bits per second. Another advantage to the parallel technique would be the ability to probe each frequency channel separately. Four out of the five channels could be transmitting while the fifth channel could reprobe to update the passive phase conjugation filter.

With the evidence that further improvements can be made to increase the throughput beyond 2,500 information bits per second, this technique could be used in near shore reconnaissance with unmanned underwater vehicles (UUV) to provide real time information through a high speed acoustic link even in shallow water. Through the use of active phase conjugation, a bi-directional high speed acoustic link can be established with an UUV with a single hydrophone on the UUV and an acoustic array deployed from the larger control vehicle. This technique can also be used to provide a high speed network for sensor nodes in shallow water.

BIBLIOGRAPHY

[1] L. Freitag, M. Grund, S. Singh, S. Smith, R. Christenson, L. Marquis, and J. Catipovic. A bidirectional coherent acoustic communication system for underwater vehicles. In *OCEANS '98 Conference Proceedings*, volume 1, pages 482–486 vol.1, 1998.

[2] T. Fu, D. Doonan, C. Utley, and Hua Lee. Field testing of a spread spectrum acoustic modem with sparse channel estimation. *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 5292–5295, April 2008.

[3] M. Stojanovic. Ofdm for underwater acoustic communications: Adaptive synchronization and sparse channel estimation. *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 5288–5291, April 2008.

[4] D.B. Kilfoyle and A.B. Baggeroer. The state of the art in underwater acoustic telemetry. *Oceanic Engineering, IEEE Journal of*, 25(1):4–27, 2000.

[5] D. Rouseff, D.R. Jackson, W.L.J. Fox, C.D. Jones, J.A. Ritcey, and D.R. Dowling. Underwater acoustic communication by passive-phase conjugation: theory and experimental results. *Oceanic Engineering, IEEE Journal of*, 26(4):821–831, 2001.

[6] Darrell R. Jackson and David R. Dowling. Phase conjugation in underwater acoustics. *The Journal of the Acoustical Society of America*, 89(1):171–181, 1991.

[7] W. A. Kuperman, William S. Hodgkiss, Hee Chun Song, T. Akal, C. Ferla, and Darrell R. Jackson. Phase conjugation in the ocean: Experimental demonstration of an acoustic time-reversal mirror. *The Journal of the Acoustical Society of America*, 103(1):25–40, 1998.

[8] P. Hoeher and J. Lodge. “turbo dpsk”: iterative differential psk demodulation and channel decoding. *Communications, IEEE Transactions on*, 47(6):837–843, 1999.

[9] Paul Hursky, Michael B. Porter, Martin Siderius, and Vincent K. McDonald. Point-to-point underwater acoustic communications using spread-spectrum passive phase conjugation. *The Journal of the Acoustical Society of America*, 120(1):247–257, July 2006.

[10] J. Makhoul. Linear prediction: A tutorial review. *Proceedings of the IEEE*, 63(4):561–580, 1975.

[11] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate (corresp.). *Information Theory, IEEE Transactions on*, 20(2):284–287, 1974.

[12] Shu Lin and Daniel J. Costello. *Error Control Coding*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, second edition, 2004.

[13] H. C. Song, W. S. Hodgkiss, W. A. Kuperman, W. J. Higley, K. Raghukumar, T. Akal, and M. Stevenson. Spatial diversity in passive time reversal commu-

nications. *The Journal of the Acoustical Society of America*, 120(4):2067–2076, October 2006.