Title

Iterative equalization and decoding applied to underwater acoustic communication

Permalink

Author

Sifferlen, James F.

Publication Date

2008

UNIVERSITY OF CALIFORNIA, SAN DIEGO

Iterative Equalization and Decoding
Applied to
Underwater Acoustic Communication

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy
in
Electrical Engineering
(Signal and Image Processing)

by

James F. Sifferlen

Committee in charge:

Professor William S. Hodgkiss, Chair
Professor Laurence B. Milstein, Co-Chair
Professor Kenneth Kreutz-Delgado
Professor William A. Kuperman
Professor John G. Proakis
Professor Paul H. Siegel

2008

The dissertation of James F. Sifferlen is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Co-Chair

Chair

University of California, San Diego

2008

DEDICATION

To Cheryl Lynn...

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to acknowledge my advisor, Professor Bill Hodgkiss, for his guidance, support and patience over these long years.

I also thank my co-advisor, Professor Larry Milstein, and the rest of my committee, Professor Bill Kuperman, Professor Paul Siegel, Professor Ken Kreutz-Delgado, and Professor John Proakis.

I am also indebted to the folks at MPL, in particular Dr. Heechun Song, for their help and encouragement.

I am especially grateful to the NATO Undersea Research Center (NURC), who was responsible for collecting the data used in this dissertation.

Finally, I want to thank my family and friends who have given me endless and invaluable support and encouragement over the years.

This work was supported by the Office of Naval Research under Grant N00014-05-1-0263.

Chapter 2, in full, is a reprint of the material as it appears in ”An Iterative Equalization and Decoding Approach for Underwater Acoustic Communication”, J. F. Sifferlen, H. C. Song, W. S. Hodgkiss, W. A. Kuperman, and J. M. Stevenson, *IEEE Journal of Oceanic Engineering*, Vol. 33, No. 2, April 2008, pp. 182-197. The dissertation author was the primary investigator and author of this paper.

Chapter 3, in full, is a reprint of the material as it appears in ”Iterative Equalization and Decoding of Communication Data Received from a Moving Underwater Acoustic Source”, J. F. Sifferlen, H. C. Song, W. S. Hodgkiss, W. A. Kuperman, and J. M. Stevenson, *IEEE Journal of Oceanic Engineering*, submitted 23 July 2008. The dissertation author was the primary investigator and author of this paper.

Chapter 4, in full, is a reprint of the material as it appears in ”Iterative Equalization and Decoding of Underwater Acoustic Communication Data Using Array Observations”, J. F. Sifferlen, H. C. Song, W. S. Hodgkiss, W. A. Kuperman, and J. M. Stevenson, *IEEE Journal of Oceanic Engineering*, submitted 22 October 2008. The dissertation author was the primary investigator and author of this paper.

VITA

1995	B.S.E.E., Rensselaer Polytechnic Institute, Troy, NY
1995-2000	U.S. Marine Corps
2002	M.S.E.E., University of California, San Diego
2002-2008	Research Assistant, University of California, San Diego
2008	Ph.D.E.E., University of California, San Diego

ABSTRACT OF THE DISSERTATION

Iterative Equalization and Decoding Applied to Underwater Acoustic Communication

by

James F. Sifferlen

Doctor of Philosophy in Electrical Engineering (Signal and Image Processing)

University of California, San Diego, 2008

Professor William S. Hodgkiss, Chair

Professor Laurence B. Milstein, Co-Chair

This dissertation focuses on data communication over shallow, long-range underwater acoustic (UWA) channels, which are characterized by relatively long, time-varying impulse responses and acute sensitivity to Doppler effects. The latter is a result of the slow speed of sound in water, while the former is a consequence of the waveguide nature of shallow, long-range UWA channels. A succession of novel receiver algorithms are developed which recover digital information transmitted across such channels. The first considers the case of a single, fixed source transducer transmitting information to a single, fixed receive hydrophone. The second extends the first to allow nontrivial source motion, for instance, that of an autonomous undersea vehicle. The third extends the second to allow processing of data received on an array of hydrophones.

The algorithms employ iterative detection. Iterative processing is creating a paradigm shift in digital communication, made possible by ever-increasing computational capabilities. There are two major components to the algorithms: an equalizer and a decoder. The main focus of this dissertation is the former which, because of the features of the UWA channel, and since the channel is not assumed known a priori at the receiver, entails adaptive resampling to correct for Doppler distortion, adaptive filtering to estimate the time-varying channel, and adaptive equalization to compensate for intersymbol interference produced by the long channel impulse responses. While decoding is performed using standard methods, its role is nonetheless crucial to the overall functioning of the algorithms. In fact, they rely on the iterative exchange of information between equalizer and decoder, and the improvement of that information with each iteration. Successful performance of the algorithms is demonstrated using data from at-sea experiments.

1

Introduction

This dissertation concerns the application of iterative equalization and decoding (IED) to underwater acoustic (UWA) communication. Iterative processing is an emerging paradigm in digital communication that encompasses powerful algorithms such as IED, which recover information transmitted through noisy distorting communication channels. UWA communication, on the other hand, has been an active area of research and development for decades. The focus is wireless digital communication systems (DCSs) that utilize acoustic wave propagation. Such systems are employed to communicate information reliably between submerged sources and receivers when it is impractical or undesirable to use wire connections. For example, autonomous undersea vehicles (AUVs) gather data and use these systems to transmit information to research ships at the sea surface. Having to use wire connections in such cases would be severely restrictive. The use of acoustic vice electromagnetic (EM) wave propagation is necessary because EM energy is severely attenuated in seawater. The nature of the UWA communication channel, particularly in shallow regions, and the power and hardware limitations of data transmitters like those found on AUVs, make UWA communication an ideal application for IED.

Fig. 1.1 shows the major components of conventional DCSs. At the transmitter, information bits are first encoded and mapped into symbols. Channel coding introduces redundancy and is necessary to protect against errors that inevitably occur when communicating through noisy channels. As a simple example, each information bit could be duplicated three times so that, at the receiver, a bit error occurs only if two or more of each trio of encoded bits are recovered erroneously. In some DCSs, symbol mapping and modulation are combined into a single operation. In the system considered in this dissertation, however, they are separate. Specifically, the encoded bit sequence is randomly permuted (the importance of which is discussed later), then mapped pairwise into quadrature phase-shift keying (QPSK) symbols, the set of which

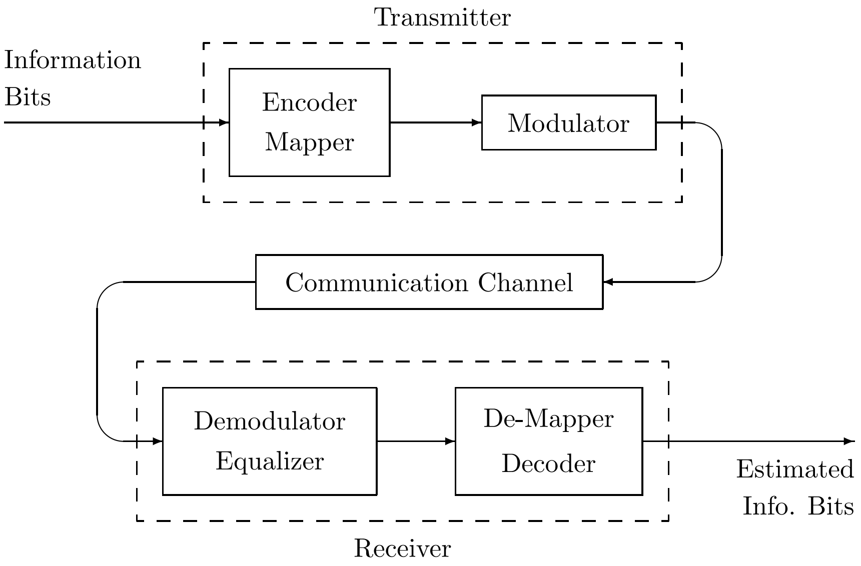


Figure 1.1: Major components of conventional digital communication systems.

form the corners of a unit square centered at the origin in the complex number plane. Modulation converts a symbol sequence into a continuous-time waveform that can physically propagate through a particular communication channel. In the present case, the transmitted signal is the product of a sinusoid at a given carrier frequency and a train of weighted and delayed pulses, the weights being the symbols. This construction is called linear modulation. At the receiver, the received wave field is first sampled. This procedure includes tasks such as detection of the presence of a communication signal and synchronization. For distorting channels, such as the UWA channel, equalization of the resulting data sequence is necessary to compensate for the distorting effects of the channel. The equalizer produces symbol estimates which are de-mapped and inverse-permuted into coded bits, then decoded into estimates of the information bits. Although not represented in the figure, a recent development in DCSs is feedback of information from the decoder to the equalizer. Simply stated, the idea is that information from the decoder can improve the equalization process, which will in turn improve the decoding process, and so on. Ideally, after some number of iterations of equalization and decoding, the receiver produces information bit estimates of higher quality than without using IED.

Fig. 1.2 illustrates the performance benefits possible with IED. The red dashed curve with dots is bit error rate (BER) versus input signal-to-noise ratio (SNR), computed via simulation, for linearly modulated uncoded binary phase-shift keying (BPSK) symbols (i.e., information bits are mapped directly into the binary symbols ± 1) transmitted over a noisy distorting UWA channel and processed a single time with the equalizer (but not the decoder, since there is no coding) of an IED algorithm presented in a later chapter. The synthetic data used to produce

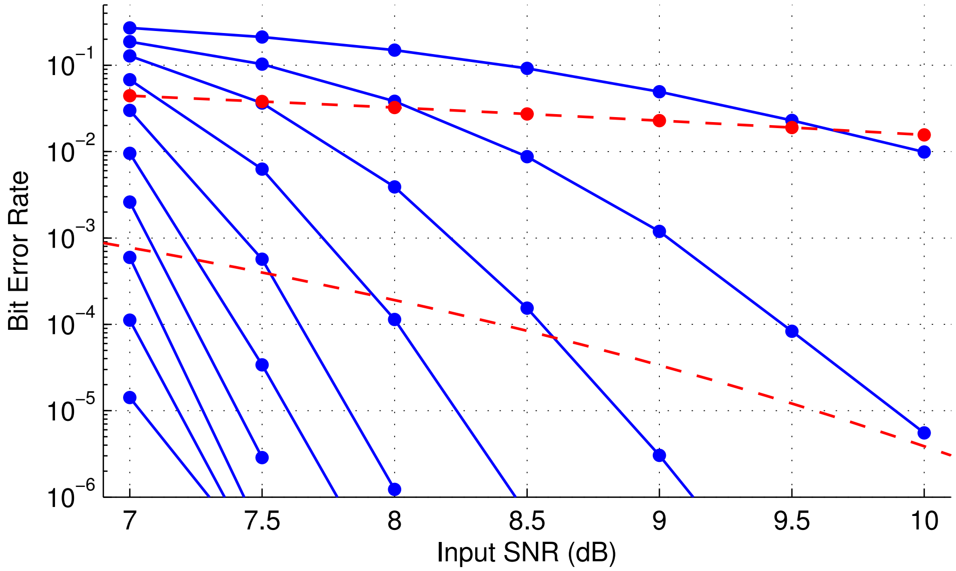


Figure 1.2: Simulation BER curves: red dashed curve (no dots) corresponds to uncoded BPSK symbols transmitted over non-distorting AWGN channel, red dashed curve with dots corresponds to uncoded BPSK symbols transmitted over a noisy distorting UWA channel with one-time equalization (no decoding), blue solid curves with dots correspond to coded QPSK symbols transmitted over a noisy distorting UWA channel. The top blue solid curve corresponds to a single equalization followed by a single decoding, the next curve below corresponds to two iterations of equalization and decoding, and so forth.

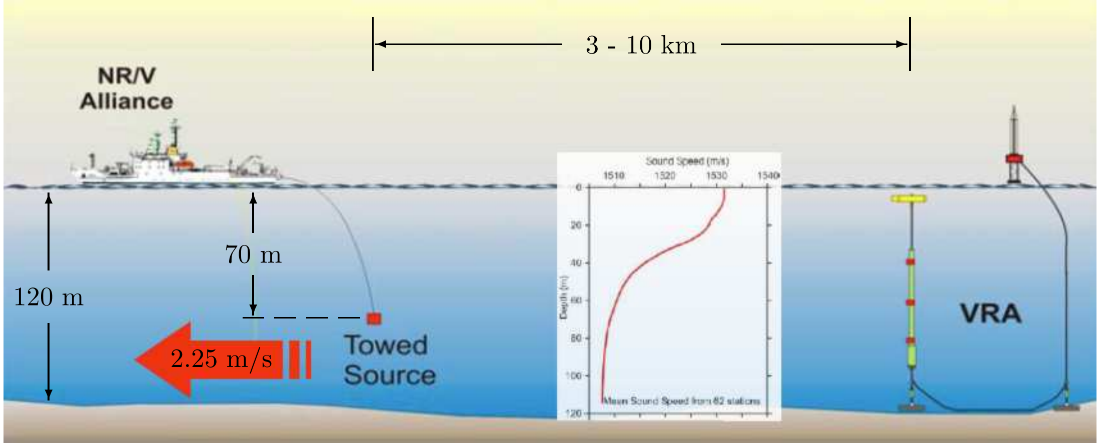
this curve was generated using measurements of an actual UWA channel, while the noise was computer-generated additive white Gaussian noise (AWGN). The blue solid curves with dots are BER versus input SNR, computed via simulation, for linearly modulated coded QPSK symbols transmitted over the same noisy distorting UWA channel and processed with the same IED algorithm for up to ten iterations: the top curve corresponds to a single equalization followed by a single decoding (that is, conventional receiver processing), the next curve below corresponds to two iterations of equalization and decoding, and so forth. The overall coding rate was one-half, so each QPSK symbol carried one bit of information on average, which is the same as the uncoded BPSK symbols. Consequently, both the uncoded BPSK and coded QPSK systems use the same transmitter power and bandwidth. The performance of the coded system with only a single iteration of equalization and decoding is worse than that of the uncoded system at lower SNRs, where the coding by itself is not strong enough to overcome the disadvantage of using the more tightly packed QPSK symbol constellation. However, this situation changes as the equalization and decoding process is iterated, as evidenced by the increasing improvement in the corresponding BER curves. Performing equalization and decoding multiple times in a cyclic manner, with symbol information passed between the two components, ultimately produces a stronger system than the comparable system that does not employ IED. After several iterations of IED the coded system performs dramatically better than both the uncoded system and the coded system with conventional detection (i.e., one-time equalization and decoding). This is the motivation for the work presented in this dissertation.

Also shown in Fig. 1.2, with the red dashed curve without dots, is BER versus input SNR for linearly modulated uncoded BPSK symbols transmitted over a non-distorting AWGN channel (i.e., a channel that simply adds noise to the communication signal). After demodulation, the binary symbols ±1 are embedded in complex AWGN with variance σ_w^2 – i.e. the SNR is $1\left/\sigma_w^2\right.$. Since the symbols are antipodal, they only see one-half of the noise variance (since the perpendicular complex dimension does not affect detection). The BER (i.e., the probability of error) can be computed analytically as $\frac{1}{2}\text{erfc}\left(\sqrt{\text{SNR}}\right)$, where erfc is the complementary error function. Comparison with the red dashed curve with dots shows the price paid in transmitting information across a distorting channel. Comparison with the blue solid curves, on the other hand, shows that this price can be overcome by using a coded system and IED. The SNR threshold for this particular example is just under 7 dB. This threshold depends upon channel characteristics: in some cases the BER curves with dots are shifted to the left, other times they are shifted to the right.

1.1 Shallow Underwater Acoustic Channels

1.1.1 Measurements & Models

UWA channels vary greatly depending upon geometrical factors such as range, ocean depth, and source and receiver depths, and environmental factors such as the variation of sound speed with depth (and range), bottom properties, surface wave conditions, and so forth [1]. For instance, a "vertical" channel in deep ocean where a source and receiver are located one above the other will have very different properties than a shallow long-range channel. The latter type of channel is considered in this dissertation. Fig. 1.3 shows the setup and key parameters of the Focused Acoustic Fields 2004 (FAF04) experiment, where the data used in this dissertation was collected. Note that the vertical receive array (VRA) of hydrophones spanned the lower two-thirds of the water column, which corresponds to the region of slowest sound speed. Since the nonuniform sound speed depth profile creates a refractive medium, and since sound bends toward regions of slower speed, the VRA spans that portion of the water column corresponding to the highest concentration of sound energy. The towed acoustic source simulated an AUV transmitting information to a research vessel at the sea surface.



Communication Signal:

- 2 - 4 kHz frequency band
- 12 kHz sampling rate
- 1 kHz symbol rate
- 9006 symbols (∼9 sec)

VRA:

- 32 sensors
- 2 m spacing
- #1 at 104 m
- #32 at 42 m

Figure 1.3: FAF04 experiment setup and key parameters. The sound speed depth profile is overlayed.

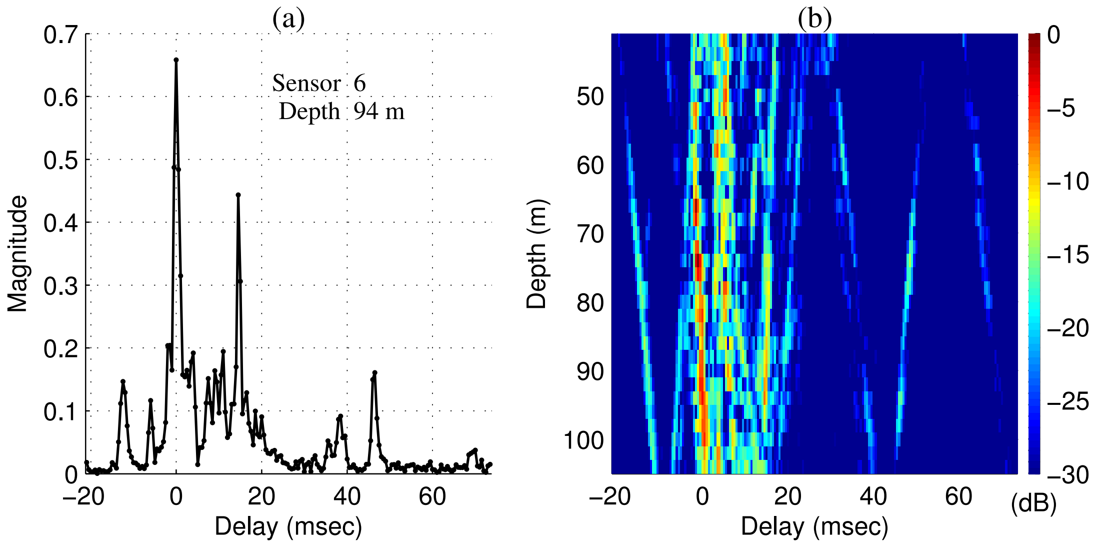


Figure 1.4: (a) Magnitude of CIR measured at 94 m depth (sensor 6, numbered from sea floor). (b) Magnitude (color scale) of CIR measured across entire 32-sensor array, spanning 42 to 104 m depth (bottom depth is 120 m).

The UWA channel is modeled as a linear time-varying system. Linearity means that if a superposition of signals is input to the system, then the output will be the superposition of system responses obtained by considering each signal independently of the others. An arbitrary band-limited signal can be decomposed into a superposition of weighted and delayed impulses, so such a system is completely described by its channel impulse response (CIR). A snapshot of typical CIRs is shown in Fig. 1.4. Part (a) shows the magnitude of the CIR measured at a depth of 94 m: the many different pulses are the result of a single transmitted pulse traveling along multiple propagation paths from the transmission source to the particular receive hydrophone. Note that the delay scale is also in units of symbol periods, since the symbol rate $R_s = 1$ symbol/msec. Consequently, the transmitted pulses, which are essentially disjoint, become significantly overlapping at the receiver: this is termed intersymbol interference (ISI). Part (b) shows the magnitudes (on a color scale) of the CIRs connecting the single source transducer to all 32 sensors of the VRA: one can clearly see the propagating wavefronts that moved through the VRA, which are nearly plane-waves since the range was much greater than the depth. When the range is much greater than the depth (by a factor of 30 to 100 for the data under consideration), the shallow UWA channel acts like a waveguide–much like a fiber-optic cable does for EM waves. The CIR in part (a) spans roughly 60 symbol periods. It is important to note that, while from a certain perspective the CIRs may seem relatively sparse, in actuality the number of multipath arrivals with significant energy is not small – i.e., they are on the order of tens of symbol periods for each

CIR. Therefore, the CIRs considered in this dissertation are neither short nor sparse. Also note that, as one can tell from the figure, the CIRs were measured at very high signal-to-noise ratios (SNRs).

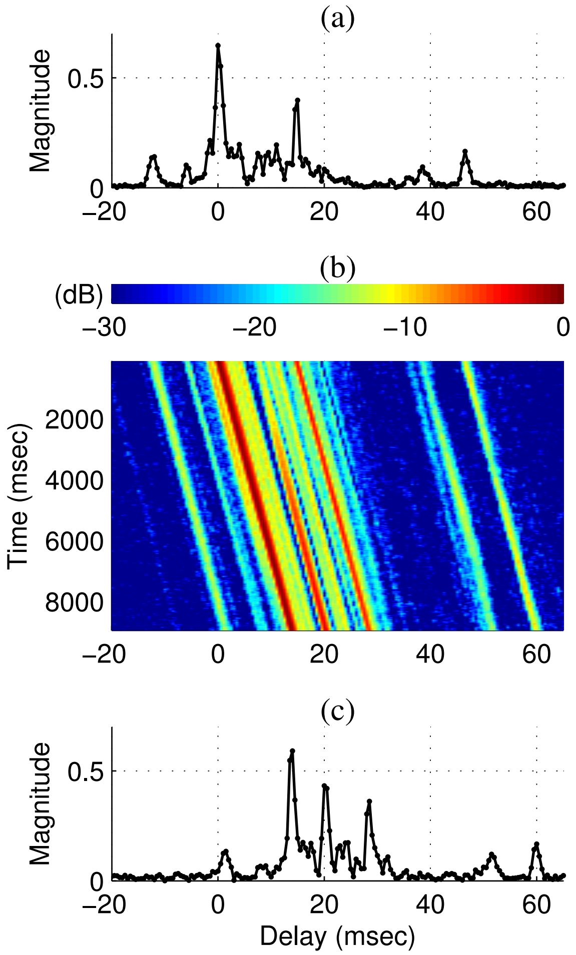


Figure 1.5: Consequence of not resampling data received on sensor at 94 m depth. (a) Magnitude of CIR measured at beginning of transmission. (b) Magnitude (color scale) of CIR measured throughout 9-second transmission. (c) Magnitude of CIR measured at end of transmission.

Since UWA channels vary with time, so do the CIRs. Relative motion of source and receiver cause the propagation paths connecting them to constantly vary in time. The motion may be inadvertant, stemming from the ceaseless undulations of the ocean itself, or intentional as in the case of an AUV which moves around while gathering information. Fig. 1.5 shows the magnitudes of CIRs measured at 94 m depth over the duration of a single 9-second transmission. Part (a) shows the CIR measured at the beginning of the transmission, part (c) shows the CIR measured at the end, and part (b) shows the CIRs measured throughout. These CIRs were measured from the received data without resampling to correct for Doppler distortion. Note that the CIRs appear to shift in delay over the course of the transmission (they are also stretched, but so little that it is undetectable). This effect, which is discussed in more detail subsequently, is a result of Doppler expansion of the received waveform relative to the transmitted waveform (since the source was moving away from the receiver). This is a critical point: the wideband nature of UWA communication signals results in Doppler compression/expansion, not just Doppler frequency shift (which would be the case with narrowband signals). That UWA signals are typically wideband is a consequence of the fact that signal attenuation increases as the square of frequency [2].

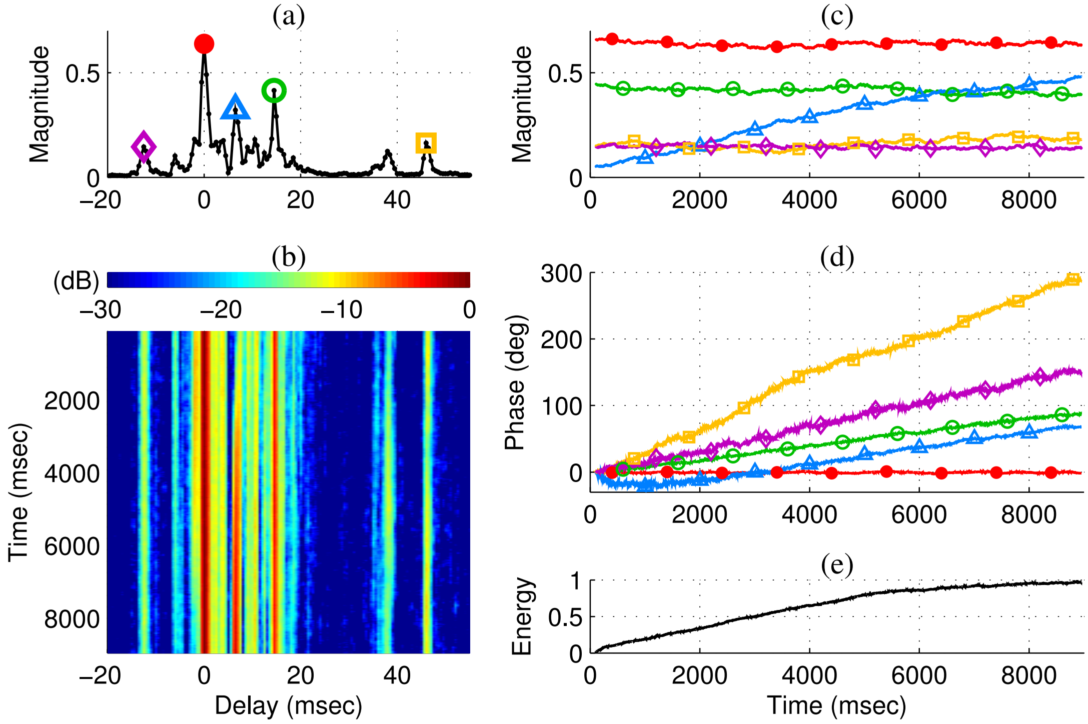


Figure 1.6: Dynamics of CIR measured from resampled data taken from sensor at 94 m depth, over duration of 9-second transmission. (a) CIR RMS values. (b) Magnitude (color scale) of CIR measured throughout transmission (compare with part (b) of Fig. 1.5). (c) and (d) Magnitude and phase, respectively, of the five time-varying CIR gains identified by colored symbols in part (a). (e) Energy in difference between each measured CIR and the first measured CIR, normalized by the maximum CIR energy.

Fig. 1.6 shows the time-varying CIR measured at the same depth (94 m) and for the same transmission as the one shown in Fig. 1.5. The important difference is that, while the latter figure shows CIRs measured without resampling the received data, the former figure shows

CIRs measured after resampling to correct for Doppler distortion. Comparison of part (b) of the two figures illustrates a crucial point: resampling the received data to compensate Doppler compression/expansion results in CIRs with drastically less time-variation than would result without resampling – and the less time variation in the CIRs, the easier it is to recover the digital information from the data. Part (a) of Fig. 1.6 shows the root-mean-square (RMS) values of the CIR gains. As can be seen from part (b), proper resampling results in multipath arrivals whose relative delays vary little over the duration of the 9-second transmission. Parts (c) and (d) show the magnitude and phase variations, respectively, of the five particular multipath arrivals indicated with colored symbols in part (a). For the most part, there is relatively little variation in the magnitudes of the CIR gains, while the phases have much more significant time variation. As one can see, the data was resampled so as to produce essentially zero phase variation on the gain of the dominant CIR arrival. Part (e) of the figure shows the energy in the difference between each measured CIR and the first measured CIR, normalized by the maximum CIR energy: although the CIR gains vary primarily in phase only, the change is enough to cause the CIR to be completely different at the end of the transmission than at the beginning. After one second the CIR changes by an amount with 20% of the maximum energy.

The frequency offsets evidenced in Fig. 1.6(d) are the result of Doppler "spread". That is, the Doppler compression/expansion is actually different for different propagation paths. Since the spread is small, the primary result is a slight frequency offset unique to each multipath. The compression/expansion is modeled as warping the continuous time variable t into $(1+\nu)\,t$, where the (generally time-varying) parameter ν is related to source and receiver speeds, as well as motion of ocean currents and surface waves and so forth. Although each multipath experiences its own Doppler "shift", it is assumed that resampling the received data sequences to compensate for an average Doppler shift results in time series accurately modeled with a slowly time-varying CIR. The signal leaving the source transducer is twice the real part of

$$s(t) \quad = \quad \exp\left(\mathrm{j}2\pi f_c t\right) \sum_n a_n h^{\mathrm{Tx}}\left(t - n\,/R_s\,\right) \tag{1.1}$$

where f_c is the carrier frequency, a_n are QPSK symbols, and $h^{\mathrm{Tx}}(t)$ is the band-limited transmit pulse. The received communication signal, after resampling, is modeled as

$$s\left(k\,/F_s\,\right) \quad = \quad \exp\left(\mathrm{j}2\pi f_c t\right) \sum_n a_n \exp\left(\mathrm{j}2\pi \frac{f_c}{R_s} \Delta\nu R_s t\right) h\left(t - \frac{n - \Delta\nu R_s t}{R_s}, t\right)\Bigg|_{t=k/F_s} \tag{1.2}$$

where F_s is the sampling frequency and $h(\tau, t)$ is the resulting time-varying CIR (τ is delay). Since the actual Doppler shift is unknown and must be estimated, the residual Doppler shift $\Delta\nu$ accounts for error in the estimation process. In the CIR measurements of Fig. 1.5(b), since no resampling was performed, the residual Doppler shift is the actual Doppler shift, which was negative, so the delays of the CIR multipaths increased as modeled by (1.2) – that is, instead of having delay n/R_s, the pulse weighted by the nth symbol has delay $(n - \Delta\nu R_s t)/R_s$. The time-varying delay $\Delta\nu R_s t$ (in symbol periods) is small compared to the corresponding phase rotation $(f_c/R_s)\,\Delta\nu R_s t$ (in cycles), which reflects the observations made earlier that the CIR gains typically have less magnitude variation than phase variation. Finally, the resampled data is modeled as

$$r_n^{(m)} \quad = \quad w_n^{(m)} + \exp\left(\mathrm{j}\phi_n\right) \sum_{l=0}^{L-1} h_{l,n}^{(m)} a_{n-l} \tag{1.3}$$

where L is the CIR span in symbol periods and, for the mth “subchannel”, $w_n^{(m)}$ is AWGN with variance $\left(\sigma_w^{(m)}\right)^2$ and $h_{l,n}^{(m)}$ is the (slowly) time-varying CIR. If the resampling rate is $2R_s$ then the sampling index $k = m + 2n$, where $m \in \{0, 1\}$ are subchannels and n is time in symbol periods. Subchannels are also created by using multiple receive sensors. The data considered in this dissertation uses both types of subchannels and, since in both cases the subchannels are assumed independent, there is no mathematical distinction and the subchannels are simply indexed as $m \in \{1, \ldots, M\}$. As described previously, the phase term, which could be absorbed into the complex-valued CIRs, reflects error in the resampling process. In particular,

$$\phi_n - \phi_{n-1} \quad = \quad 2\pi\left(f_c/R_s\right)\Delta\nu \tag{1.4}$$

where, although not explicitly denoted, $\Delta\nu$ is time-varying. Note that there is an inherent ambiguity in the model (1.3) since the phase term $\exp\left(\mathrm{j}\phi_n\right)$ could be absorbed into the time-varying complex-valued CIRs. This ambiguity is resolved during CIR estimation, as discussed in later chapters.

1.1.2 Channel Estimation

The IED algorithms described in later chapters utilize Kalman filter (KF) CIR estimation [3]. Although the channel model parameters (CMPs) for the data considered in this dissertation are deemed deterministic (but unknown and time-varying), the KF nevertheless performs effective and easily-implemented CIR estimation. The underlying model for the “standard” KF assumes an equal amount of variation on the real and imaginary parts of the gains of all CIR arrivals. However, since the gains are expected to rotate more than change in magnitude, this model is not appropriate: the resulting CIR estimates would be unnecessarily noisy. Noting that, for a given amount of phase rotation, gains with larger magnitudes will vary more than smaller ones, a modification can be made to the standard KF. This “modified” KF incorporates into its state dynamics that CIR gains will have an amount of variation proportional to their magnitudes (details of how this is accomplished are described in later chapters). The real and imaginary parts of the gains, however, would still have the same amounts of variation. An alternative is to explicitly model the gains in polar form:

$$\begin{aligned} h_{l,n}^{(m)} &= \rho_{l,n}^{(m)} \exp\left(\mathrm{j}\theta_{l,n}^{(m)}\right) && (1.5) \\ &= \left(\acute{\rho}_{l,n}^{(m)} + \Delta\rho_{l,n}^{(m)}\right) \exp\left(\mathrm{j}\left(\acute{\theta}_{l,n}^{(m)} + \Delta\theta_{l,n}^{(m)}\right)\right) && (1.6) \\ &\cong \acute{h}_{l,n}^{(m)} + \exp\left(\mathrm{j}\acute{\theta}_{l,n}^{(m)}\right)\left(\Delta\rho_{l,n}^{(m)} + \mathrm{j}\acute{\rho}_{l,n}^{(m)}\Delta\theta_{l,n}^{(m)}\right) && (1.7) \end{aligned}$$

where $\rho_{l,n}^{(m)}$ are the CIR gain magnitudes and $\theta_{l,n}^{(m)}$ are the phases. The quantities $\acute{\rho}_{l,n}^{(m)}$ and $\acute{\theta}_{l,n}^{(m)}$ are the predicted estimates of the gain magnitudes and phases, respectively, and $\Delta\rho_{l,n}^{(m)}$ and $\Delta\theta_{l,n}^{(m)}$ are the corresponding variations about the predicted values. The approximation in (1.7) yields a linear model. Details of the resulting "extended" KF (EKF) can be found in later chapters. The important point is that the EKF allows the radial and transverse components of CIR gains to be controlled independently, so they can explicitly be restricted to change in magnitude less than they rotate in phase – consistent with the model and assumptions.

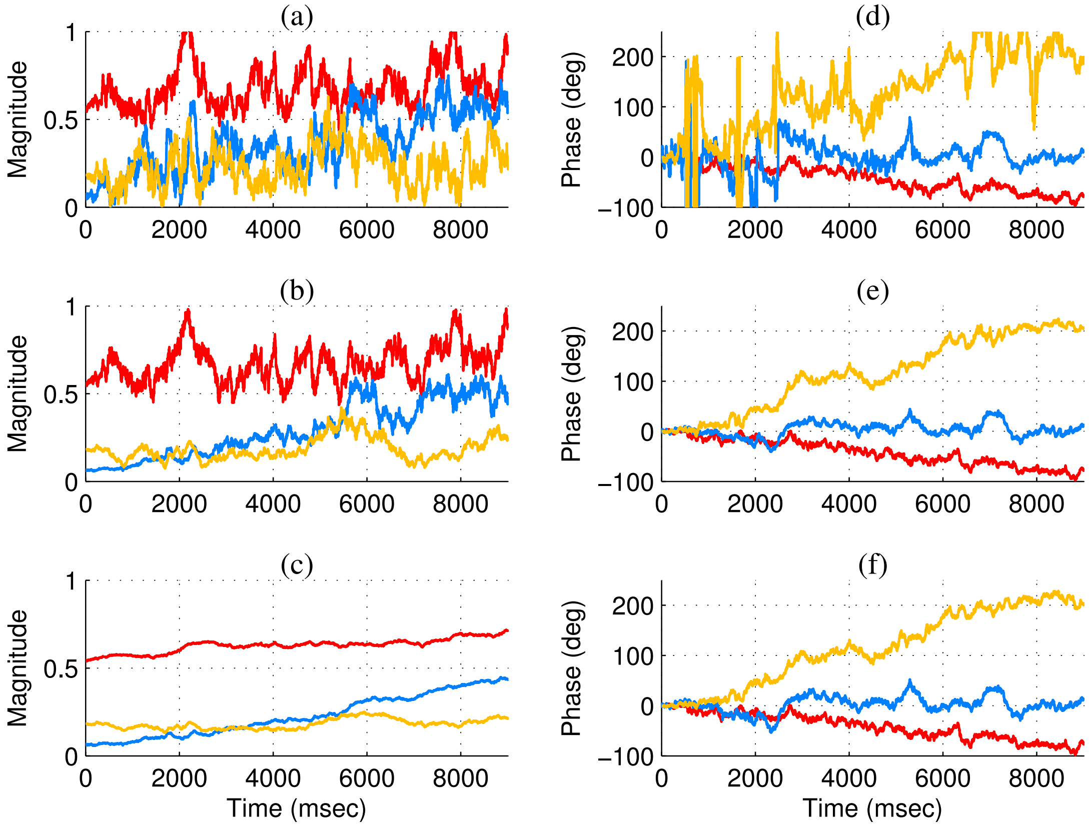


Figure 1.7: Comparison of KF implementations: (a) and (d) are "standard" KF, (b) and (e) are "modified" KF, (c) and (f) are EKF.

Figs. 1.7 and 1.8 demonstrate the performance of the various KF implementations.

The first of these shows the estimated time-varying gains of the three particular CIR arrivals indicated with a solid circle, a triangle, and a square in Fig. 1.6(a). In Fig. 1.7, parts (a) and (d) are for the standard KF, (b) and (e) are for the modified KF, and (c) and (f) are for the EKF. Parts (a),(b), and (c) show the estimated gains, while (d), (e), and (f) show the phases. For some data sequences it is better to use the modified KF during the first iteration of the IED algorithm (even though doing so theoretically allows more noise to be projected onto the CIR estimates), while for other data sequences it is better to use the EKF during the first iteration. For the data presently under consideration, the KF worked best during the first iteration. Consequently, to produce Fig. 1.7, the first iteration used the modified KF CIR estimation, while the second and subsequent iterations used one of the three KF implementations described in the previous paragraph. The results shown in the figure correspond to the second iteration. The standard KF clearly performs worse than the other two methods. Note that the modified KF produces roughly the same phase variations as the EKF, while the modified KF allows more magnitude variations than the EKF – consistent with the underlying modeling assumptions for the two implementations.

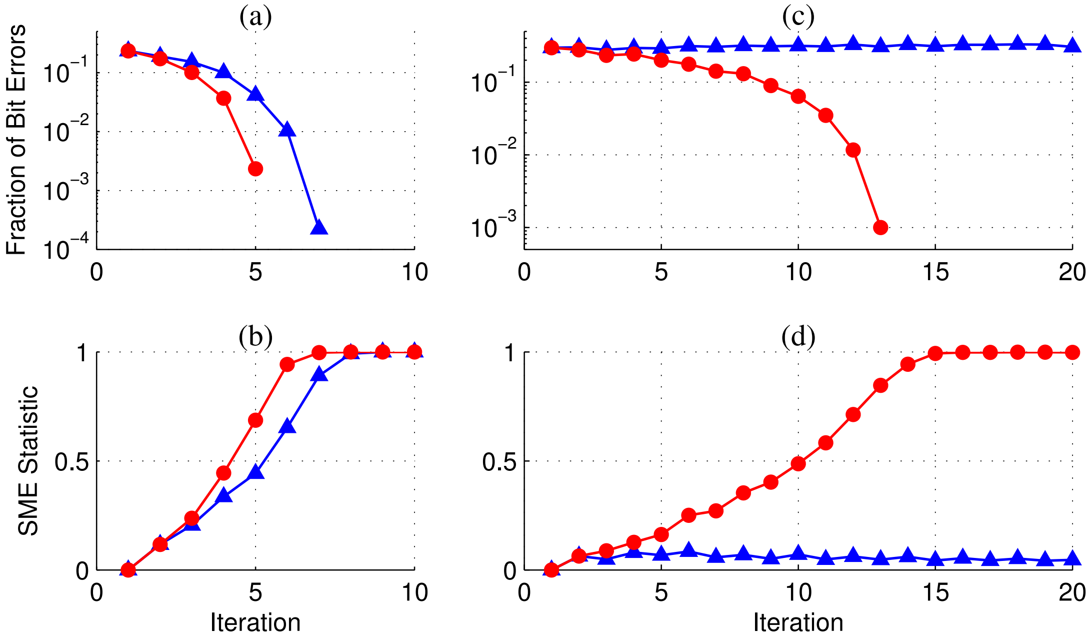


Figure 1.8: Comparison of KF implementations. (a) and (c) are fraction of bit errors. (b) and (d) are the magnitude of a symbol-quality statistic. (a) and (b) correspond to the single-sensor example, (c) and (d) to the two-sensor example. In all cases, blue curves with triangles are "modified" KF, red curves with circles are EKF.

Most important is how the various KF implementations impact the performance of the overall IED algorithm. This is shown in Fig. 1.8. Results for the standard KF are not shown, since it resulted in divergence of the algorithm for both cases. Parts (a) and (b) are for the same single-sensor case used in the preceeding paragraph, while parts (c) and (d) are for a two-sensor case (using data from the same sensor as the single-sensor case in addition to data from the sensor immediately below it). Parts (a) and (c) show the fraction of bit errors (there were 9006 information bits in each transmission). Parts (b) and (d) show the magnitude of a symbol-quality statistic that is discussed in more detail later. It is the correlation between the true symbols and symbol estimates produced by the algorithm. The important point is that the magnitude of the statistic takes on values between zero and one, with zero reflecting complete lack of symbol information and one reflecting perfect knowledge of the symbols. Results using the modified KF are indicated with blue curves with triangles, while the EKF results are indicated with red curves with circles. In both the single-sensor and two-sensor cases, using the EKF during the second and subsequent iterations yields better results than using the modified KF. This is especially so in the two-sensor case, where the IED algorithm did not even successfully recover the information bits using the modified KF.

1.2 Channel Equalization

Channel equalization is the process of compensating for the distorting effects of a communication channel. As discussed previously, there are two primary sources of distortion in the shallow UWA channel: Doppler compression/expansion and multipath interference. Nonuniform resampling is used to compensate for Doppler effects. The need for resampling was demonstrated in the previous section – without it a channel varies at a much greater rate than necessary. The idea of resampling is shown in Fig. 1.9. It is assumed that the sequence $x_k = x\left(t_k\right)$, where $x\left(t\right)$ is a band-limited continuous-time signal and the sampling times t_k are locally linear in the index k. The value $x\left(k\left/F_s\right.\right)$ is interpolated as follows. There is an index $\acute{k}$ such that $t_{\acute{k}} \leq k\left/F_s\right. < t_{\acute{k}+1}$, so $x_{\acute{k}+i} \cong x\left(t_{\acute{k}} + i\acute{T}_k\right)$, where $\acute{T}_k = t_{\acute{k}+1} - t_{\acute{k}}$ approximates the instantaneous sampling period. Consequently,

$$x\left(k\left/F_s\right.\right) \quad \cong \quad \sum_i x_{\acute{k}+i}\operatorname{sinc}\left(\left(t - t_{\acute{k}}\right)\left/\acute{T}_k\right. - i\right)\Big|_{t=k/F_s} \tag{1.8}$$

where $\operatorname{sinc}\left(\kappa\right) = \sin\left(\pi\kappa\right)/\left(\pi\kappa\right).$ Since $\operatorname{sinc}\left(\kappa\right)$ only decays as $1\left/\kappa\right.$, in practice a faster-decaying interpolation pulse is used – e.g., the Kaiser-windowed [4] sinc pulse used in the figure – so that only a small number of terms need be used in the summation.

Distortion resulting from self-interference caused by multipath propagation can be compensated in various ways. Short or sparse CIRs can be "equalized" using methods based on a

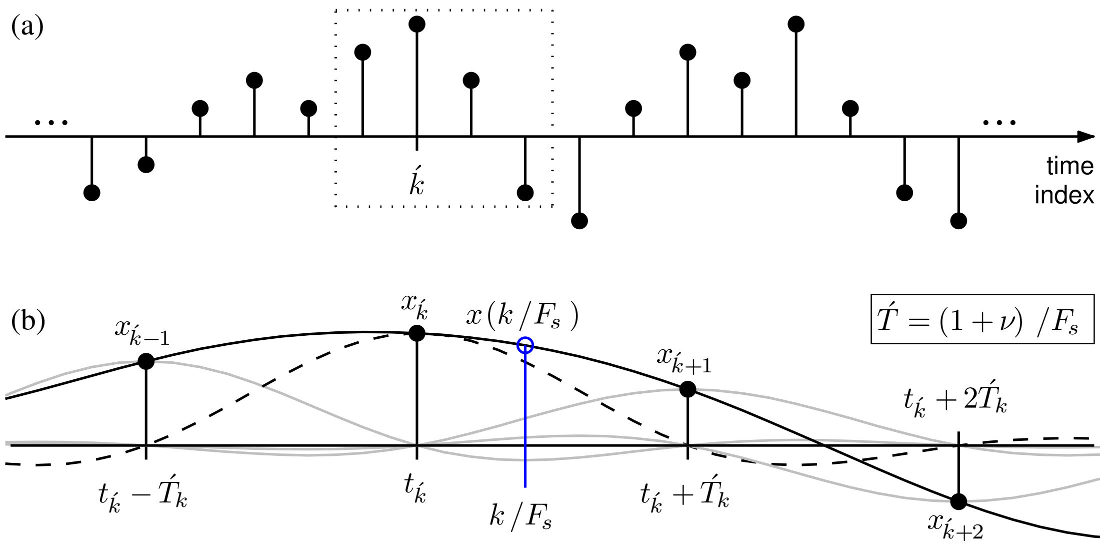


Figure 1.9: Illustration of resampling process. (a) a data sequence. (b) Details of obtaining a resampled value (blue open circle) within the dotted box of part (a): the sampled values (black solid circles) are interpolated using a superposition of weighted and shifted copies of a band-limited pulse (dashed and gray curves) to obtain a value of the underlying continuous waveform (blue open circle) at a time instant that does not correspond to one of the sampling times.

hidden Markov model (HMM) [3] of the channel [5]. Such methods are discussed in the next section. The algorithms developed in later chapters use a method of equalization based on linear filtering, which is explained presently.

For a block of data samples the model (1.3) becomes

$$\begin{aligned}
\mathbf{r}_n^{(m)} &= \left[r_n^{(m)},\ldots,r_{n+L-1}^{(m)}\right]^{\mathrm{T}} && (1.9)\\
&= \mathbf{w}_n^{(m)} + \begin{bmatrix} \exp\left(\mathrm{j}\phi_n\right) & & \\ & \ddots & \\ & & \exp\left(\mathrm{j}\phi_{n+L-1}\right) \end{bmatrix} \begin{bmatrix} h_{L-1,n}^{(m)} & \cdots & h_{0,n}^{(m)} & & \\ & \ddots & \vdots & \ddots & \\ & & h_{L-1,n+L-1}^{(m)} & \cdots & h_{0,n+L-1}^{(m)} \end{bmatrix} \mathbf{a}_n && (1.10)
\end{aligned}$$

where $\mathbf{w}_n^{(m)}$ is an AWGN vector and $\mathbf{a}_n = \left[a_{n-L+1},\ldots,a_n,\ldots,a_{n+L-1}\right]^{\mathrm{T}}$. It is assumed that $\phi_{n+i} \cong \phi_n$ and $h_{l,n+i}^{(m)} \cong h_{l,n}^{(m)}$, for $i=0,\ldots,L-1$. Consequently,

$$\mathbf{r}_n^{(m)} \cong \mathbf{w}_n^{(m)} + \exp\left(\mathrm{j}\phi_n\right) H_n^{(m)} \mathbf{a}_n \qquad (1.11)$$

where $H_n^{(m)}$ is the CIR convolution matrix in (1.10) with $h_{l,n+i}^{(m)}$ replaced by $h_{l,n}^{(m)}$. A set of linear filters $\mathbf{f}_n^{(m)}$ (expressed as row vectors) operate on the data vectors to form the filtered output

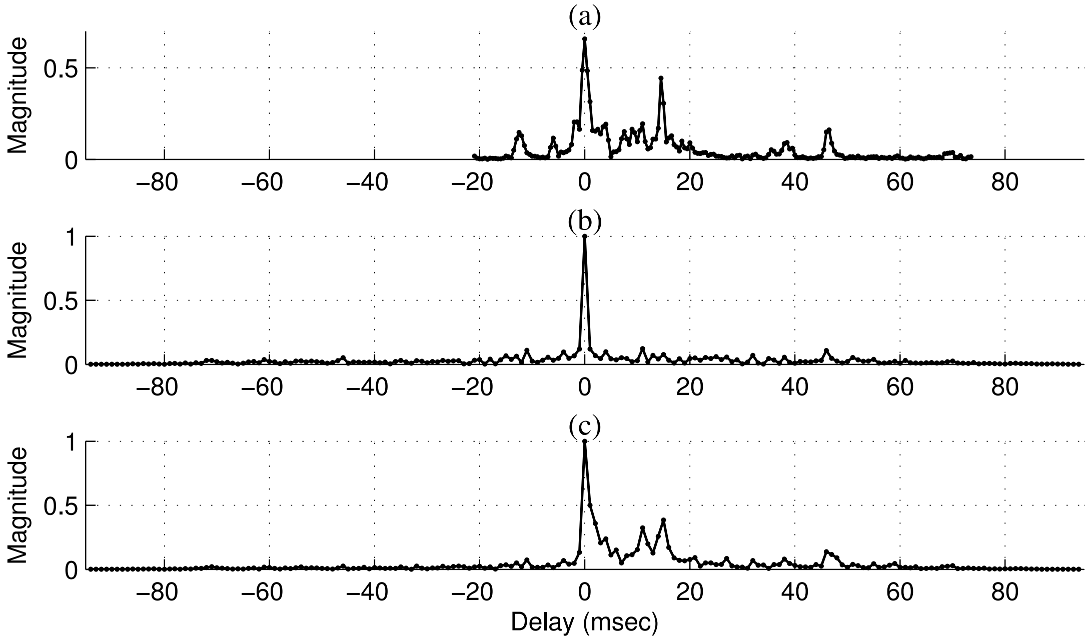


Figure 1.10: Example of EIRs for cases of LE and DFE. (a) Magnitude of CIR measured at 94 m depth. (b) Magnitude of EIR for case of applying LE to CIR of part (a). (c) Magnitude of EIR for case of applying DFE to CIR of part (a).

sequence

$$\begin{aligned} z_n &= \sum_m \mathbf{f}_n^{(m)} \mathbf{r}_n^{(m)} && (1.12) \\ &\cong u_n + \exp\left(\mathrm{j}\phi_n\right) \sum_{l=1-L}^{L-1} g_{l,n} a_{n-l} && (1.13) \\ \mathbf{g}_n &= \sum_m \mathbf{f}_n^{(m)} H_n^{(m)} && (1.14) \\ &= \left[g_{L-1,n}, \ldots, g_{0,n}, \ldots, g_{1-L,n}\right] && (1.15) \end{aligned}$$

where $u_n = \sum_m \mathbf{f}_n^{(m)} \mathbf{w}_n^{(m)}$ and the sequence $g_{l,n}$ is the equalized impulse response (EIR). The “linear” equalizer (LE) uses $\exp\left(-\mathrm{j}\phi_n\right) z_n$ as a statistic to detect symbol a_n. Consequently, the filters are chosen to minimize the noise-plus-interference (NPI) variance $\mathrm{E}\left|u_n\right|^2 + \sum_{l\neq 0} \left|g_{l,n}\right|^2$ subject to the constraint $g_{0,n} = 1$. Fig. 1.10 shows an example: part (a) is the magnitude of the CIR at depth 94 m and part (b) is the magnitude of the corresponding EIR for the LE.

In (1.13), for $l > 0$, $g_{l,n}$ multiplies symbols which preceed the current symbol a_n. Another common method of filter construction assumes previous symbols are detected without error and their contribution to the convolution sum in (1.13) can be cancelled. Letting $\hat{a}_{n-l}$, for $l > 0$, denote “hard” symbol estimates, which are assumed correct, the output of the decision feedback equalizer (DFE) and its associated hard symbol estimate are, respectively,

$$z_n^{\mathrm{DFE}} = \exp\left(-\mathrm{j}\phi_n\right) z_n - \sum_{l>0} g_{l,n} \hat{a}_{n-l} \tag{1.16}$$

$$\hat{a}_n = \arg\min_{\alpha \in \mathcal{A}} \left| z_n^{\mathrm{DFE}} - \alpha \right| \tag{1.17}$$

where $\mathcal{A}$ denotes the QPSK symbol alphabet. The DFE "forward" filters are chosen (under the assumption that previous hard symbol estimates are correct) to minimize $\mathrm{E}\left|u_n\right|^2 + \sum_{l<0} \left|g_{l,n}\right|^2$ subject to the constraint $g_{0,n} = 1$. Fig. 1.10(c) shows the magnitude of the EIR for the DFE associated with the CIR shown in part (a).

1.3 Receiver Algorithms

If an information bit sequence b is sent through a channel with model parameters $\mathcal{P}$ and produces data $\mathcal{D}$ at a receiver, then the optimal information bit estimates are

$$\hat{b}_n = \arg\max_{\beta \in \{0,1\}} \int \sum_{b : b_n = \beta} p\left(b, \mathcal{P}, \mathcal{D}\right) \mathrm{d}\mathcal{P} \tag{1.18}$$

where $p\left(b, \mathcal{P}, \mathcal{D}\right)$ is the joint probability of the information bits, the CMPs, and the received data. However, although simple to state, these optimal estimates are impossible to compute directly given the modulation and channel considered in this dissertation. The reason being that worthwhile channel coding requires the use of long information bit sequences and producing a reasonable data rate results in ISI on the order of tens of symbol periods. Consequently the resulting data pdf is too complicated. Instead, an indirect method is used to approximate the optimal bit estimates. The method is "belief propagation", in conjunction with suitable approximations, and results in applying the sum-product algorithm (SPA) to a factor graph (FG) representing the joint pdf of the bits, encoded bits, symbols, CMPs, and received data [6]. It is important to note that the overarching principle in detection of the information bits is the marginalization of a very large, yet structured, probability density.

1.3.1 Factor Graphs and the Sum-Product Algorithm

Consider a HMM with initial state x_{-1}, hidden states x_n, and observations y_n, for $n = 0, \ldots, N-1$. The joint probability of the data y and states x can be factored as $p\left(x, y\right) = p\left(x_{-1}\right) \prod_{n=0}^{N-1} p\left(x_n, y_n \left| x_{n-1}\right.\right)$. This product of conditional probabilities is depicted in the FG at the top of Fig. 1.11. The boxes, or "nodes", which represent the probabilities, are functions of the variables connected to them. The connections, typically called "edges", represent the variables. The observations y_n are constants, not variables. The dashed box at the right terminus is a function of the final state variable only. In the present case, where there is no a priori information about the final state, the dashed box can simply be regarded as unity or, if more appropriate, a Gaussian density with zero mean and large variance.

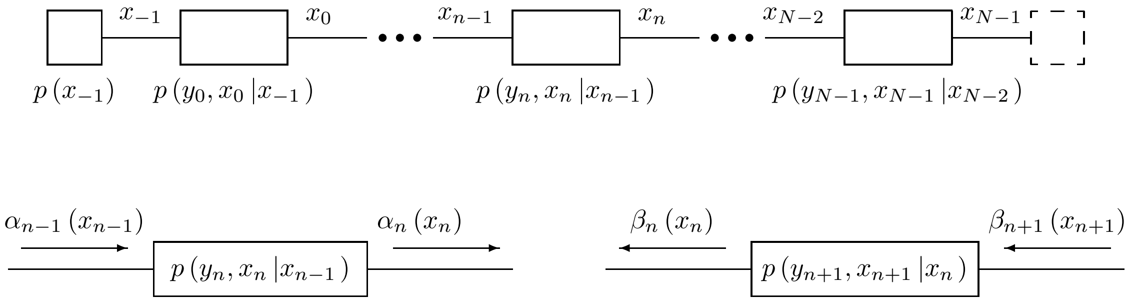


Figure 1.11: HMM FG and message flow. Top: FG. Bottom left: forward messages. Bottom right: backward messages.

There are two types of "messages" computed in this HMM example, as shown at the bottom of Fig. 1.11. The initial "forward" message is simply $\alpha_{-1}(x_{-1}) = p(x_{-1})$, and the remaining ones are computed recursively as $\alpha_n(x_n) = \sum_{x_{n-1}} p(x_n, y_n | x_{n-1}) \alpha_{n-1}(x_{n-1})$ (in this expression and elsewhere, summation should be replaced with integration where necessary, and vice versa). The initial (or final, depending upon one's point of view) "backward" message $\beta_{N-1}(x_{N-1})$ is simply the function of x_{N-1} represented by the dashed box, and the remaining ones are computed recursively as $\beta_n(x_n) = \sum_{x_{n+1}} p(x_{n+1}, y_{n+1} | x_n) \beta_{n+1}(x_{n+1})$. Note that $\alpha_n(x_n) = p(x_n, y_0^n)$ and $\beta_n(x_n) = p\left(y_{n+1}^{N-1} | x_n\right)$, where $y_{n_1}^{n_2}$ is shorthand for $y_{n_1}, \ldots, y_{n_2}$. Consequently, the joint probability of the nth state and the entire observation sequence is $p(x_n, y) = \alpha_n(x_n) \beta_n(x_n)$, i.e., the product of the corresponding forward and backward messages. Since the optimal estimate of each state is given by $\hat{x}_n = \arg\max_{x_n} p(x_n, y)$, the procedure just outlined is an efficient method of performing the marginalizations of $p(x, y)$ necessary to obtain these maximum a posteriori (MAP) estimates.

A few comments are in order. Often it is not joint probabilities, e.g., $p(x_n, y_0^n)$ and $p(x_n, y)$, that are of interest but, rather, conditional densities, e.g., $p(x_n | y_0^n)$ and $p(x_n | y)$. However, the latter are obtained from the former simply by scaling: e.g., $p(x_n | y_0^n) = p(x_n, y_0^n) / \sum_{x_n} p(x_n, y_0^n)$. The messages can be scaled accordingly, i.e., $\alpha_n(x_n) \leftarrow \alpha_n(x_n) / \sum_{x_n} \alpha_n(x_n)$. Computing the messages to within an arbitrary scale factor is usually all that is necessary anyhow. From a computational standpoint, scaling of some sort is necessary to prevent messages from becoming too small. Also, different types of messages are often used to facilitate implementation of the SPA, e.g., using the logarithm (or negative of the logarithm) of the messages described above. The SPA can be performed in other commutative semirings (addition without requirement for additive inverses and commutative multiplication) than the standard one considered here [7], e.g., the "max-sum" semiring can be used to derive the Viterbi algorithm [8]. Finally, it should be remarked that for a given system there is often more than one FG associated with it (and, in general, a given FG can lead to different instances of the SPA depending upon how the flow of messages is controlled).

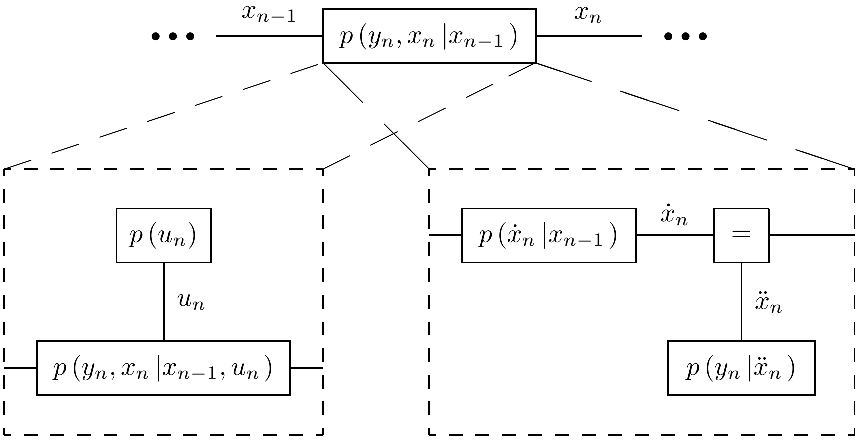


Figure 1.12: HMM FG block details. Top: FG block. Bottom left: BCJR algorithm block details. Bottom right: Kalman algorithms block details.

The HMM considered above is more than just an illustrative example, it also represents two major components of the receiver algorithms introduced in this dissertation. In particular, the BCJR algorithm [9] is used in the algorithms' decoders. It utilizes the HMM of Fig. 1.11, with details of the basic FG block shown at the bottom left of Fig. 1.12. The forward and backward messages are initialized as before, but their recursions now become

$$\begin{aligned}\alpha_n(x_n) &= \sum_{u_n}\sum_{x_{n-1}} p(x_n, y_n | x_{n-1}, u_n)\,\alpha_{n-1}(x_{n-1})\,p(u_n)\\ \beta_n(x_n) &= \sum_{u_{n+1}}\sum_{x_{n+1}} p(x_{n+1}, y_{n+1} | x_n, u_{n+1})\,\beta_{n+1}(x_{n+1})\,p(u_{n+1})\end{aligned}$$

where the "inputs" u_n are the information bits and $p(u_n)$ are their a priori probabilities, i.e., the messages flowing out of the blocks with the selfsame labels (which may be constant functions, indicating no prior knowledge about the bits). The joint probability of the observations and each state remains $p(x_n, y) = \alpha_n(x_n)\,\beta_n(x_n)$. A new type of message arises, namely, the conditional probability of the observations given each bit (i.e., the bit "likelihood")

$$p(y | u_n) = \sum_{x_{n-1}}\sum_{x_n} p(x_n, y_n | x_{n-1}, u_n)\,\alpha_{n-1}(x_{n-1})\,\beta_n(x_n)$$

These messages flow out of the $p(x_n, y_n | x_{n-1}, u_n)$ blocks and into the $p(u_n)$ blocks. The joint probability of the observations and each bit is thus computed as $p\left(u_n, y\right) = p\left(y\left|u_n\right.\right) p\left(u_n\right)$, from which MAP bit estimates are readily obtained.

The Kalman filtering and smoothing algorithms are also obtained using the HMM of Fig. 1.11. The details of the basic FG block are shown at the bottom right of Fig. 1.12, which represents the factorization $p\left(y_n, x_n\left|x_{n-1}\right.\right) = p\left(y_n\left|x_n\right.\right) p\left(x_n\left|x_{n-1}\right.\right)$. Since each state variable is used more than two times, it is necessary to create the auxiliary variables $\dot{x}_n$ and $\ddot{x}_n$ and use the special node indicated with an equality symbol, which enforces equality among all the variables connected to it – i.e., it represents the function $\delta\left(\dot{x}_n - \ddot{x}_n\right)\delta\left(\ddot{x}_n - x_n\right)$, where δ is either the Kronecker or Dirac delta function. The messages entering the equality nodes from the $p\left(y_n\left|\ddot{x}_n\right.\right)$ nodes are just these probabilities themselves. The recursion for forward messages is now computed in two steps,

$$\begin{aligned}
\dot{\alpha}_n\left(\dot{x}_n\right) &= \int p\left(\dot{x}_n\left|x_{n-1}\right.\right)\alpha_{n-1}\left(x_{n-1}\right)\mathrm{d}x_{n-1} \\
\alpha_n\left(x_n\right) &= \int\int \delta\left(\dot{x}_n - \ddot{x}_n\right)\delta\left(\ddot{x}_n - x_n\right) p\left(y_n\left|\ddot{x}_n\right.\right)\dot{\alpha}_n\left(\dot{x}_n\right)\mathrm{d}\dot{x}_{n-1}\mathrm{d}\ddot{x}_{n-1} \\
&= p\left(y_n\left|x_n\right.\right)\dot{\alpha}_n\left(x_n\right)
\end{aligned}$$

where the “prediction” messages $\dot{\alpha}_n\left(\dot{x}_n\right)$ flow out of the $p\left(\dot{x}_n\left|x_{n-1}\right.\right)$ nodes into the equality nodes. Note that $\dot{\alpha}_n\left(x_n\right) = p\left(x_n, y_0^{n-1}\right)$. If the messages are normalized, e.g., $\alpha_n\left(x_n\right) \leftarrow \alpha_n\left(x_n\right)/\left[\int \alpha_n\left(x_n\right)\mathrm{d}x_n\right]$, then the computation of forward messages constitutes Kalman *filtering*. Kalman *smoothing* is obtained by incorporating the backward messages, which are also computed in two steps in a similar manner as the forward messages. The messages leaving the equality nodes and entering the $p\left(y_n\left|\ddot{x}_n\right.\right)$ nodes are

$$\begin{aligned}
\mu\left(\ddot{x}_n\right) &= \int\int \delta\left(\dot{x}_n - \ddot{x}_n\right)\delta\left(\ddot{x}_n - x_n\right)\dot{\alpha}_n\left(\dot{x}_n\right)\beta_n\left(x_n\right)\mathrm{d}\dot{x}_{n-1}\mathrm{d}x_{n-1} \\
&= \dot{\alpha}_n\left(\ddot{x}_n\right)\beta_n\left(\ddot{x}_n\right)
\end{aligned}$$

which makes $\mu\left(x_n\right) = p\left(x_n, y_0^{n-1}, y_{n+1}^{N-1}\right)$.

All SPAs have two fundamental components: message computation and message flow. In principle, the former is simple as there is only a single rule for message computation: to compute the message leaving a node along a particular edge, multiply the node’s function by the incoming messages from the other edges and sum/integrate out all their variables. This basic procedure can be verified in the message computations described earlier. Message flow or “scheduling” is the other critical element of SPAs. In the HMM-based examples considered thus far the scheduling has been fairly obvious: messages start at the ends of the FG and proceed sequentially toward the opposite ends. In the case of the BCJR algorithm, “upward” messages are computed by taking the corresponding forward and backward messages as inputs. Typically, blocks at the periphery of a FG represent either prior probabilities or probabilities that are joint with observed data samples. The latter produce only output messages, namely the functions they represent, while messages that flow into the former are usually the desired end result of the processing, e.g., the joint probability of the data with each information bit. The FGs in the HMM cases did not possess loops – i.e., one cannot trace a path on such a graph and end up at the starting point without retracing part of the graph. For graphs without cycles the SPA is *exact*. This is evident in the BCJR and Kalman algorithms, where the messages have very precise interpretations – e.g., $\alpha_n(x_n) = p(x_n, y_0^n)$ and $\beta_n(x_n) = p\left(y_{n+1}^{N-1} | x_n\right)$. These algorithms are signal processing workhorses and invaluable in their own right. However, the SPA also forms the core of the most powerful decoding and receiver algorithms known, which all possess a common trait: they are represented by FGs with loops and their implementations are all iterative. The messages in such cases may not have such precise meanings as for the HMM-based cases, but they usually have clear interpretations. The analysis of iterative SPAs on FGs with loops is an area of active research, driven by the amazing success of such algorithms [10, 11].

1.3.2 Iterative Equalization and Decoding

The FGs of some now classic channel capacity-approaching codes are shown in Fig. 1.13. The nodes in part (c) marked with a plus sign enforce a zero-sum constraint on the variables attached to them. The dashed boxes in parts (b) and (d) represent convolutional codes (CCs), which have the HMM structure used to describe the BCJR algorithm. The oval blocks marked with Π, "permuters", perform (pseudo-)random permutations of their input bit sequences. Consequently, they are simply message routers in SPAs. They are vitally important, however: they effectively create long, random codes using easily decoded constituent codes. Long, random codes are a necessary element of capacity-appoaching codes.

The performance of the original Turbo codes [12], i.e., parallel-concatenated CCs, using iterative decoding was initially unbelievable. Their success prompted new interest in low-density parity check (LDPC) codes [13], whose iterative decoding algorithm is similarly effective. There now exist numerous other variations on this theme of powerful channel codes whose effectiveness lies in iterative decoding algorithms. An important commonality was discovered among the various algorithms, namely, that they are all instances of the SPA [14]. One crucial concept that emerged with the introduction of Turbo codes is that of "extrinsic" information: each message computed during the decoding algorithm, which can be interpreted as the probability mass function (pmf) or likelihood of a given bit or symbol, must be independent of the incoming information pertaining to that particular bit or symbol. If this condition is violated, the decoding algorithms stagnate from recycling the same information over and over. This fundamental concept is transparent in the SPA: its message computation rule produces exactly the extrin-

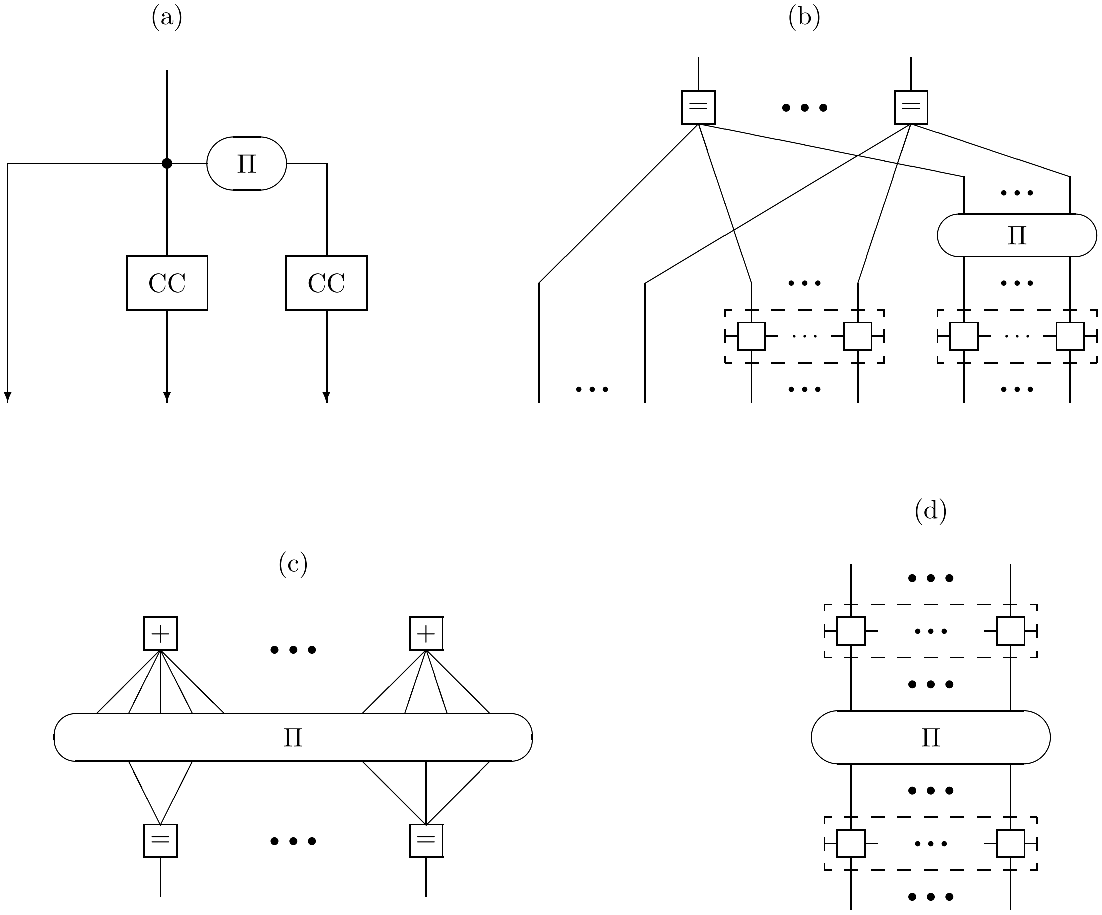


Figure 1.13: (a) and (b) Parallel-concatenated Turbo code schematic and FG, respectively. (c) LDPC Code FG. (d) Serial-concatenated Turbo code FG.

sic information – of course, proper message flow and scheduling are also necessary to avoid the recycling of old information and the accompanying ill effects.

The code depicted in Fig. 1.13(d) is of particular importance. It consists of the serial concatenation of two CCs, separated by a permuter. An analogy was made between channel codes of this type and a similar system where the "inner" CC is replaced by a channel with a nontrivial CIR, which is also a type of convolutional code. Detection algorithms that apply IED to this serial concatenation of channel code, permuter, and CIR are called "turbo" equalizers [15, 16]. Originally the channel was assumed known at the receiver. Extension to channels with parameterized models whose CMPs are not known at the receiver, and so must be estimated in conjunction with equalization, is a relatively new area of research. The case of short or sparse CIRs is fairly well developed. In such a case equalization can be performed using BCJR-type algorithms with each state performing its own separate estimation of CMPs [17]. Another approach in the case of channels with few CMPs is to expand the state to include the CMPs [18]. For example, if a hypothetical channel has a one-symbol memory and a single binary CMP, then the state can be defined as the ordered pair of the symbol in memory and the value of the single binary CMP. These two approaches, however, are not directly applicable to CIRs that are neither short nor sparse. Various methods of state reduction have been proposed – such as filtering to produce a more favorable EIR, cancelling the "tail" of the CIR/EIR using hard/soft symbol estimates, and considering only a subset of all possible states in the HMM – but it is not clear how such methods perform at SNRs low enough so that CIR estimation error is a significant issue. The focus of this dissertation is IED at low SNR of unknown, time-varying CIRs that are neither short nor sparse.

The IED algorithms developed in this dissertation are represented by the FGs and message diagrams of Fig. 1.14. Part (a) shows the major system functions: the uppermost block represents convolutionally encoding information bits, permuting the resulting sequence of encoded bits, and mapping the latter into QPSK symbols; the "channel" block represents the channel model (1.3) or (1.11); and the remaining blocks represent the Markov model (MM) structures assumed for the time-varying phase ϕ_n and CIRs $h_{l,n}^{(m)}$. The details of these latter blocks are similar to the HMM FG discussed previously. Part (b) shows details of the channel block as applied to CIR estimation. Parts (c) and (d) show message flow for equalization and CIR estimation, respectively, and part (e) shows the equivalent equalization FG after marginalizing out all variables except the current symbol and equalizer output model parameters (EOMPs). The focus of this dissertation is equalization in conjunction with channel estimation – i.e., the computation of messages that flow between the lower blocks of part (a), and messages that flow into the uppermost block from the channel block. The functions in the uppermost block, and

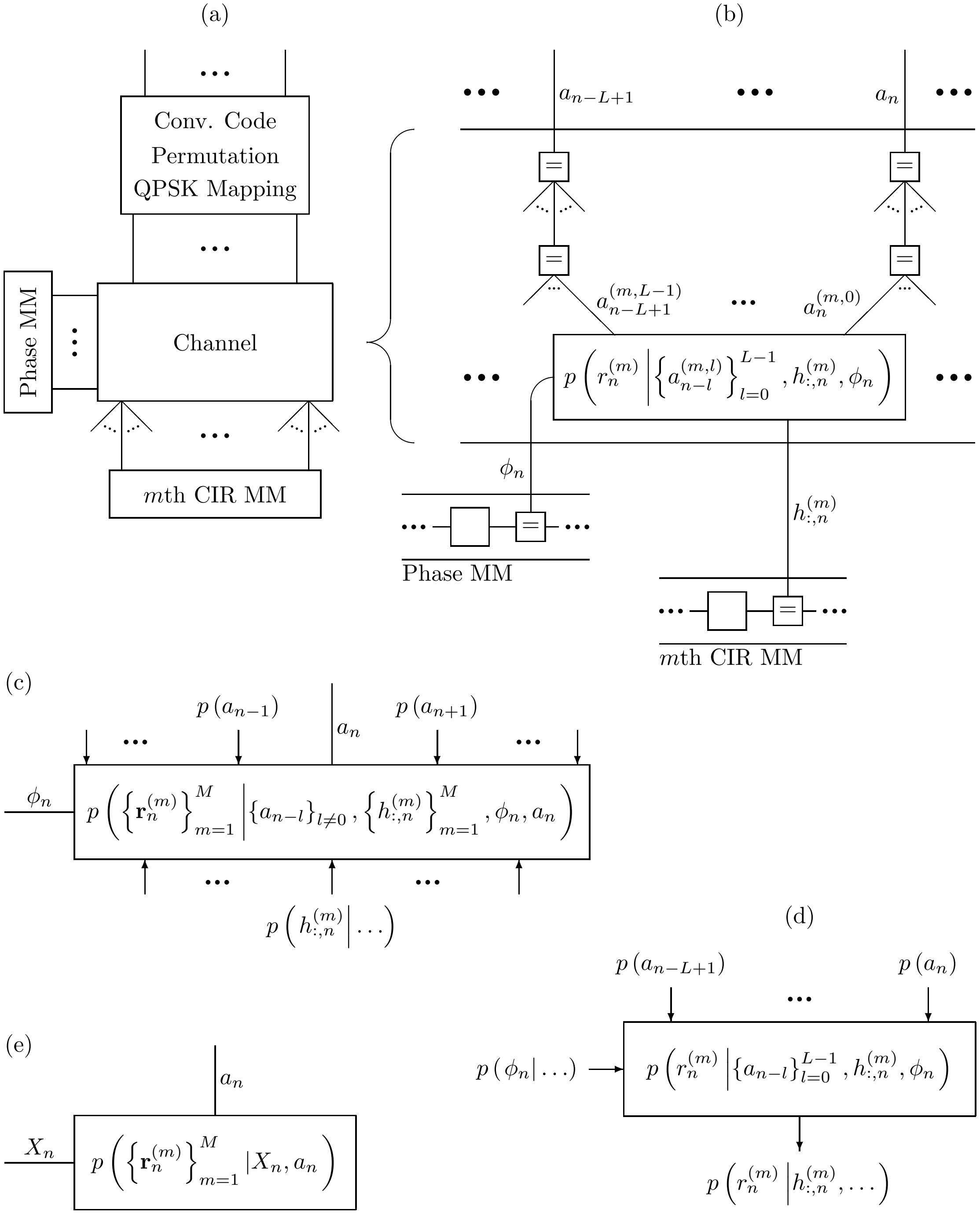

(a)
Conv. Code
Permutation
QPSK Mapping
Channel
Phase MM
mth CIR MM
(b)
a_{n-L+1}
a_n
$a_{n-L+1}^{(m,L-1)}$
$a_n^{(m,0)}$
$p\left(r_n^{(m)} \middle| \left\{a_{n-l}^{(m,l)}\right\}_{l=0}^{L-1}, h_{:,n}^{(m)}, \phi_n\right)$
ϕ_n
$h_{:,n}^{(m)}$
Phase MM
mth CIR MM
(c)
$p\left(a_{n-1}\right)$
a_n
$p\left(a_{n+1}\right)$
ϕ_n
$p\left(\left\{\mathbf{r}_n^{(m)}\right\}_{m=1}^{M} \middle| \left\{a_{n-l}\right\}_{l\neq 0}, \left\{h_{:,n}^{(m)}\right\}_{m=1}^{M}, \phi_n, a_n\right)$
$p\left(h_{:,n}^{(m)} \middle| \ldots\right)$
(d)
$p\left(a_{n-L+1}\right)$
$p\left(a_n\right)$
$p\left(\phi_n \middle| \ldots\right)$
$p\left(r_n^{(m)} \middle| \left\{a_{n-l}\right\}_{l=0}^{L-1}, h_{:,n}^{(m)}, \phi_n\right)$
$p\left(r_n^{(m)} \middle| h_{:,n}^{(m)}, \ldots\right)$
(e)
a_n
X_n
$p\left(\left\{\mathbf{r}_n^{(m)}\right\}_{m=1}^{M} | X_n, a_n\right)$

Figure 1.14: (a) IED algorithm FG. (b) Details of "Channel" block for CIR estimation. (c) Approximate equalization FG and messages. (d) CIR estimation messages. (e) Equalization FG after marginalizing out all variables except current symbol and EOMPs.

the computation of messages that flow within this block as well as down into the channel block, are standard and described elsewhere [9, 19].

No information from previous iterations is stored in the FG. During CIR estimation, therefore, symbol messages entering the channel block are simply duplicated by the equality-constraint nodes and flow into the data pdf nodes. Consequently, including the incoming phase messages, part (b) of Fig. 1.14 results in the message diagram of part (d). Since the CIRs of interest contain many multipath arrivals, direct computation of outgoing CIR messages according to the SPA is intractable. An approximation introduced by Tuchler, et al, is used instead [20, 21]. Let

$$\mathcal{N}\left(x;\bar{x},C_x\right) = \left[\pi^{L_x}\det\left(C_x\right)\exp\left(\left(x-\bar{x}\right)^{\mathrm{H}}C_x^{-1}\left(x-\bar{x}\right)\right)\right]^{-1}$$

where L_x is the length of vector x. Symbol messages $p\left(a_n\right)$, interpreted as a priori pmf's provided by the coding-permuting-mapping block, are replaced with suitable Gaussian pdf's:

$$\hat{p}\left(a_n\right) = \mathcal{N}\left(a_n;\bar{a}_n,\sigma_{\bar{a}}^2\right) \tag{1.19}$$

$$\bar{a}_n = \sum_{\alpha\in\mathcal{A}}\alpha p\left(a_n=\alpha\right) \tag{1.20}$$

$$\sigma_{\bar{a}}^2 = 1-\left\langle\left|\bar{a}_n\right|^2\right\rangle \tag{1.21}$$

where $\langle\cdot\rangle$ indicates time-averaging and (1.21) follows from the unit-variance requirement on the symbols. Tuchler's approximation allows the outgoing message in part (d) to be computed as

$$\begin{aligned}
p\left(r_n^{(m)}\middle|h_{:,n}^{(m)},\phi_n\right) &\cong \int\cdots\int\mathcal{N}\left(r_n^{(m)};\exp\left(\mathrm{j}\phi_n\right)\sum_{l=0}^{L-1}h_{l,n}^{(m)}a_{n-l},\left(\sigma_w^{(m)}\right)^2\right)\prod_{l=0}^{L-1}\hat{p}\left(a_{n-l}\right)\mathrm{d}a_{n-l}\\
&= \mathcal{N}\left(r_n^{(m)};\exp\left(\mathrm{j}\phi_n\right)\sum_{l=0}^{L-1}h_{l,n}^{(m)}\bar{a}_{n-l},\left(\sigma_w^{(m)}\right)^2+\sigma_{\bar{a}}^2\sum_{l=0}^{L-1}\left|h_{l,n}^{(m)}\right|^2\right)\\
p\left(r_n^{(m)}\middle|h_{:,n}^{(m)},\ldots\right) &= \int p\left(r_n^{(m)}\middle|h_{:,n}^{(m)},\phi_n,\ldots\right)p\left(\phi_n\middle|\ldots\right)\mathrm{d}\phi_n
\end{aligned}$$

where $h_{:,n}^{(m)}=\left[h_{0,n}^{(m)},\ldots,h_{L-1,n}^{(m)}\right]^{\mathrm{T}}$. A linear approximation is made for $\exp\left(\mathrm{j}\phi_n\right)$ to allow computation of the final message in a closed form, which is amenable to subsequent use in Kalman filter CIR estimation. The symbol mean estimates (SMEs) $\bar{a}_n$ are an alternative to the "hard" estimates $\hat{a}_n=\arg\max_\alpha p\left(a_n=\alpha\right)$.

Tuchler's approximation is also used during equalization [22, 23]. However, instead of using the FG of Fig. 1.14(b), which represents the model (1.3), we use a FG that represents the model (1.11). The corresponding message diagram is shown in part (c). The first step is to marginalize out the CIR estimates. Note that, in (1.11), the product $H_n^{(m)}\mathbf{a}_n = A_n h_{:,n}^{(m)}$, where

$$A_n = \begin{bmatrix} a_n & \cdots & a_{n-L+1} \\ \vdots & \ddots & \vdots \\ a_{n+L-1} & \cdots & a_n \end{bmatrix}$$

Consequently,

$$\begin{aligned}
& p\left(\left\{\mathbf{r}_n^{(m)}\right\}_{m=1}^{M} \middle| \{a_{n-l}\}_{l\neq 0}, \phi_n, a_n\right) \\
& \quad\cong \int\cdots\int \prod_{m=1}^{M} \mathcal{N}\left(\mathbf{r}_n^{(m)}; \exp\left(\mathrm{j}\phi_n\right) A_n h_{:,n}^{(m)}, \left(\sigma_w^{(m)}\right)^2 \mathrm{I}\right) \mathcal{N}\left(h_{:,n}^{(m)}; \acute{h}_{:,n}^{(m)}, \mathrm{Cov}\left[h_{:,n}^{(m)}\right]\right) \mathrm{d}h_{:,n}^{(m)} \\
& \quad= \prod_{m=1}^{M} \mathcal{N}\left(\mathbf{r}_n^{(m)}; \exp\left(\mathrm{j}\phi_n\right) A_n \acute{h}_{:,n}^{(m)}, \left(\sigma_w^{(m)}\right)^2 \mathrm{I} + A_n \mathrm{Cov}\left[h_{:,n}^{(m)}\right] A_n^{\mathrm{H}}\right) \\
& \quad\cong \prod_{m=1}^{M} \mathcal{N}\left(\mathbf{r}_n^{(m)}; \exp\left(\mathrm{j}\phi_n\right) \acute{H}_n^{(m)} \mathbf{a}_n, \left[\left(\sigma_w^{(m)}\right)^2 + \mathrm{trace}\left\{\mathrm{Cov}\left[h_{:,n}^{(m)}\right]\right\}\right] \mathrm{I}\right)
\end{aligned}$$

where I is an identity matrix and $p\left(h_{:,n}^{(m)} \middle| \left\{r_{n-l}^{(m)}\right\}_{l>0}, \ldots\right) = \mathcal{N}\left(h_{:,n}^{(m)}; \acute{h}_{:,n}^{(m)}, \mathrm{Cov}\left[h_{:,n}^{(m)}\right]\right)$ are the predicted CIR pdf's supplied by KF CIR estimation. Computing

$$p\left(\left\{\mathbf{r}_n^{(m)}\right\}_{m=1}^{M} | \phi_n, a_n\right) \cong \int\cdots\int p\left(\left\{\mathbf{r}_n^{(m)}\right\}_{m=1}^{M} \middle| \{a_{n-l}\}_{l\neq 0}, \phi_n, a_n\right) \prod_{l\neq 0} \hat{p}\left(a_{n-l}\right) \mathrm{d}a_{n-l}$$

using Tuchler's approximation for the incoming symbol messages is similar to the computation of $p\left(r_n^{(m)} \middle| h_{:,n}^{(m)}, \phi_n\right)$, except that the symbol a_n is not integrated out. The upshot is that the IED algorithms use a LE during the first iteration, a matched-filter with hard-symbol-estimate cancellation during later iterations, and a cross between the two during intermediate iterations. Details are provided in subsequent chapters.

To complete the equalization process, symbol and phase messages could be computed from $p\left(\left\{\mathbf{r}_n^{(m)}\right\}_{m=1}^{M} | \phi_n, a_n\right)$ by marginalizing out each of the remaining variables. However, it turns out to be more effective to proceed as follows. A model for the equalizer filter output is

$$\begin{aligned} z_n &= v_n + \exp\left(\mathrm{j}\phi_n\right) \gamma_n a_n && (1.22) \\ \omega_n &= 2 \Big/ \mathrm{E}\left|v_n\right|^2 \end{aligned}$$

where γ_n is the resulting gain and ω_n is the reciprocal of one-half of the NPI variance. Define the state vector of EOMPs $X_n^{\mathrm{T}} = [\omega_n, \gamma_n, \phi_n]$. The additional variables γ_n and ω_n are theoretically computed as part of the procedure described in the previous paragraph. However, although the error in the CIR estimates is taken into account via their covariance matrices, the resulting "theory" values of gain and reciprocal-half-NPI variance are unavoidably skewed. Using biased values for these parameters produces symbol messages of poorer quality than would be obtained using unbiased values. Consequently, we estimate these two additional EOMPs directly and thus incorporate them into the FG block shown in Fig. 1.14(e). Details of the procedure for computing the messages associated with this block are given in later chapters.

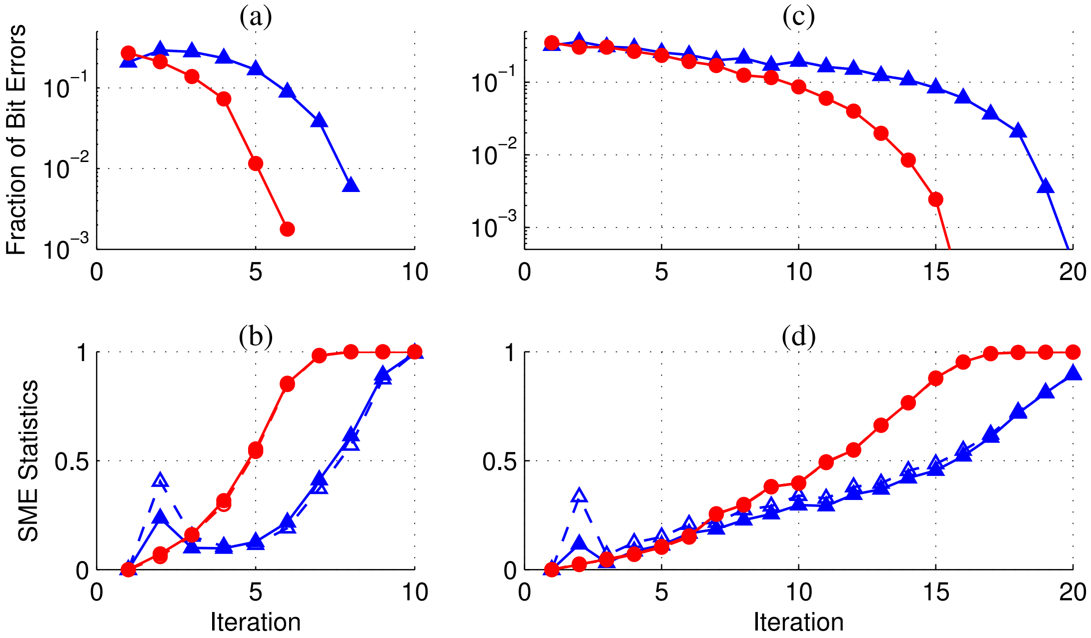


Figure 1.15: Comparison of algorithm performance using theory values (blue curves with triangles) and measured values (red curves with circles) of EOMPs. (a) and (b) Fraction of bit errors and magnitude of PSME statistic $\chi_{\bar{a}}$ (solid curves and symbols), respectively, for single-sensor processing example. (c) and (d) Fraction of bit errors and magnitude of PSME statistic $\chi_{\bar{a}}$ (solid curves and symbols), respectively, for two-sensor processing example. In (b) and (d), the dashed curves with open symbols correspond to the PSME statistic $\mathcal{E}_{\bar{a}}$ (note, they are not visible in the cases of measured values).

The benefit of including the additional EOMPs and measuring them directly, as opposed to using theory values, is shown in Fig. 1.15. Two SME statistics first need to be defined:

$$\begin{aligned} \mathcal{E}_{\bar{a}} &= \left\langle |\bar{a}_n|^2 \right\rangle \\ \chi_{\bar{a}} &= \langle a_n^* \bar{a}_n \rangle \end{aligned}$$

The first of these is simply the average energy in the SMEs. The second is the correlation between the SMEs and the true symbols. Since it involves the true symbols, the latter statistic is obviously not used within the IED algorithms. However, since the symbol sequences transmitted during the experiment are known, this statistic is a useful diagnostic to monitor performance of the algorithms. Fig. 1.15 is similar to Fig. 1.8, which shows the benefits of using EKF CIR estimation over the modified KF CIR estimation. In the present scenario, Fig. 1.15 shows the impact of using theory versus measured values on the overall performance of the IED algorithm. Parts (a) and (b) are for the single-sensor case, while parts (c) and (d) are for the two-sensor case. Parts (a) and (c) show the fraction of bit errors (there were 9006 information bits in each transmission) and parts (b) and (d) show $|\chi_{\bar{a}}|$ (recall that $|\chi_{\bar{a}}| = 0$ implies complete lack of symbol information, while $|\chi_{\bar{a}}| = 1$ reflects perfect knowledge of the symbols). Results using the theory values for γ_n and ω_n are indicated with solid blue curves with triangles, while those using the measured values are indicated with solid red curves with circles. In both the single-sensor and two-sensor cases, using the measured values yields better results than using the theory values. The improvement is more pronounced in the two-sensor case, where measuring the values directly saves five iterations of IED. Additionally, it should be the case that $\mathcal{E}_{\bar{a}} = \chi_{\bar{a}}$. Parts (b) and (d) also show $\mathcal{E}_{\bar{a}}$, with dashed blue curves with triangles indicating the use of theory values and dashed red curves with circles indicating the use of measured values. The two symbol statistics are virtually identical during all iterations of the algorithm when using measured values. The curves corresponding to using theory values, on the other hand, demonstrate that the theory values are skewed and produce a mismatch between the statistics – especially during the early iterations.

During the first iteration there is no symbol information available from the coding-permuting-mapping block to assist CMP estimation. Nevertheless, updating the estimates of the time-varying CMPs requires symbol information. This is accomplished in two ways. For one, the actual systems interleave symbols known at the receiver into the symbol sequence for the purpose of stabilizing CMP estimation. Additionally, during the first iteration only, the messages produced by the equalizer – which are estimates of the likelihoods $p\left(\left\{\mathbf{r}_n^{(m)}\right\}_{m=1}^{M}|a_n\right)$ – are fed back for CIR estimation by setting

$$p\left(a_n\right) = p\left(\left\{\mathbf{r}_n^{(m)}\right\}_{m=1}^{M}|a_n\right) \Bigg/ \sum_{a_n} p\left(\left\{\mathbf{r}_n^{(m)}\right\}_{m=1}^{M}|a_n\right)$$

which is an a posteriori probability with a neutral prior.

Near the lower SNR limit of successful performance of the IED algorithms, the symbol information provided by the coding-permuting-mapping block during the second and subsequent early iterations can have very small values for $\mathcal{E}_{\bar{a}} \cong \chi_{\bar{a}}$, that is, very little energy. Instead of using this information to estimate the CIRs, it may seem better to use the CIR estimates from the previous iteration or, as is done in the first iteration, feed back the symbol information produced by the equalizer during the current iteration to estimate the CIRs. Doing so, however, results in stagnation of the algorithm and failure to improve symbol information beyond the first iteration.

This situation reinforces the concept of extrinsic information. When the SME energy is very low, it is all the more important to employ a method of CIR estimation that uses as few degrees of freedom as possible. This is the purpose of the EKF CIR estimation, which is very well matched to the channel dynamics.

It is important to distinguish between the strength of symbol information and its reliability. Clearly $\chi_{\bar{a}}$ (its magnitude, that is – its phase is assumed to be nearly zero) is a measure of the strength of symbol information. The information is reliable, on the other hand, when $\mathcal{E}_{\bar{a}} \cong \chi_{\bar{a}}$. The latter condition is necessary to ensure unbiased CIR estimates and meeting this condition drove many of the design choices in the algorithms. It was found, for instance, that cancellation-based equalizers (i.e., DFEs or similar ones that employ cancellation with "soft" symbol estimates [24]) violate the condition and produce biased CIR estimates. This is presumably the result of recycling information. Additionally, "precoding" [25,26], i.e., the use of a rate-1 code prior to symbol mapping and modulation, which is reported to improve IED by creating a recursive inner code, was found to destroy the equality of $\mathcal{E}_{\bar{a}}$ and $\chi_{\bar{a}}$. Using precoding produces serially-concatenated CCs. Other types of channel codes, such as LDPC codes, were also found to produce inequality between the two symbol statistics. Puncturing – the discarding of encoded bits to increase code rate – also resulted in inequality and is only used in one IED algorithm where just 2% of the encoded bits are discarded. These matters all require further investigation.

Much more could be said about IED and UWA communication. The intent of this chapter, however, is merely to provide the necessary background for the IED algorithms discussed in later chapters. The next section gives overviews of these chapters.

1.4 Previews of Remaining Chapters

The remaining chapters are organized as follows: the middle three present a succession of IED algorithms, while the last concludes the dissertation and suggests areas for future research. Previews of the middle chapters are given presently.

The algorithm presented in Chapter 2 processes data transmitted from a single fixed source transducer to a single fixed receive hydrophone. The communication signal frequency band is 3-4 kHz, so the bandwidth is equal to the symbol rate of 1 kHz. The sampling frequency was 12 kHz and the transmissions were 10 seconds in length. Performance results are presented for a single transmission using data from each sensor of the 32-sensor VRA. Since there was no deliberate source motion, a channel model was used that is more restrictive than the one discussed previously in this chapter. In particular, the key assumption made is that proper (nonuniform) resampling results in a time-invariant CIR. In other words, the channel model of

(1.3) still applies, but there is no time index on the CIR multipath arrivals (and there is only a single subchannel). Training symbols, i.e., symbols known to the receiver which are inserted at the beginning of the communication signal, are used to obtain initial estimates of CMPs. When using relatively few training symbols at low SNR, a tradeoff exists between estimating too many CIR arrivals, which projects too much noise onto the resulting CIR estimate, and estimating too few CIR arrivals, which discards available signal energy and allows it to act as interference. A method referred to as CIR "framing" is developed that strikes a balance between the two ends of this spectrum. The resampling procedure used in all the IED algorithms presented in this dissertation is also developed in Chapter 2. On the other hand, as a consequence of the time-invariant CIR assumption, an alternative strategy is used to perform CMP estimation. During the first iteration the CIR is estimated using only the training symbols. The small average Doppler shift is then estimated using a non-data-aided approach rooted in the equivalence of the minimum mean-square error (MMSE) LE and the constant-modulus LE (both of which are equivalent, to within a scale factor, to the minimum-variance distortionless response (MVDR) LE discussed earlier in this chapter). Finally, the slowly-fluctuating portion of the Doppler shift is estimated via estimation of the phase in (1.3), which is performed using a Viterbi-algorithm-like procedure [3]. Chapter 2 also contains a derivation of Tuchler's LE-based equalization procedure that is a direct application of the SPA in conjunction with the Gaussian pdf approximation for symbol messages. Additionally, the case is made for measuring the EOMPs directly vice using theory values, and this is implemented in the IED algorithm. The algorithm was able to successfully process data sequences with SNR around 6 dB.

The algorithm presented in Chapter 3 extends that of Chapter 2 in two major ways. Most significantly, the new algorithm processes data transmitted from a moving source. The source had a single transducer and the algorithm processes data from a single fixed receive hydrophone. The algorithm of Chapter 3 also utilizes two subchannels, created by virtue of the fact that the communication signal frequency band is 2-4 kHz, which is twice the symbol rate of 1 kHz. The sampling frequency remains 12 kHz, but the transmissions were just over 9 seconds in length. Performance results are presented for processing data from each sensor of the 32-sensor VRA for over 50 transmissions. The new algorithm implements a CIR framing method that extends the one from Chapter 2 and performs sparse CIR estimation, that is, estimation of CIRs over non-contiguous sets of delays. A CIR measuring and modeling section, similar to but more thorough than the one in this chapter, motivates the modeling of CIR gains in polar form (1.5) and the consequent development and implementation of the EKF routine used to estimate them. The new algorithm uses pilot symbols in addition to training symbols to assist and stabilize CMP estimation. Pilot symbols are known to the receiver and are interleaved with the data bearing symbols. The channel coding, however, is performed in such a manner that the information bits are mapped into the same number of QPSK symbols – including training and pilot symbols. Tuchler's equalization procedure is implemented using the two subchannels. The procedure is split into two stages: filtering the Doppler-compensated data, then estimating the EOMPs and computing symbol a posteriori probability messages. Resampling is only performed during the first iteration. However, it is performed adaptively, in conjunction with CIR estimation and equalization, by using the estimated phase sequence ϕ_n as a control signal in accordance with (1.4). The data collected during the experiment that contained communication signals was at very high SNR. Consequently, these communication sequences were combined with suitably scaled recordings of ambient noise to yield data with an SNR of 7 dB. Results are presented for processing over 1,600 such data sequences that show high reliability and a low frequency of bit errors.

The algorithm presented in Chapter 4 is an extension of the previous one that allows the receiver to process array data. The system parameters are the same as in Chapter 3, since the same data is used – that is, the data is similarly constructed by adding scaled ambient noise recordings to high-SNR communication sequences. The algorithm processes data transmitted from a single moving source transducer to the fixed VRA of hydrophones. The use of data collected on an array should allow the receiver algorithm to operate at lower SNR per sensor than would be the case using just a single sensor. Several modifications to the single-sensor algorithm were necessary to accommodate lower SNRs on the sensors. A coherent method of signal detection is implemented in the algorithm of Chapter 4, whereas the previous algorithms were able to operate successfully using an incoherent method. The framing method of the previous algorithm, which obtained a sparse CIR estimate, is not suitable when the SNRs on the sensors are low. A different procedure is used whereby the CIR is estimated over a full fixed span, then those multipath arrivals with insufficient energy are zeroed-out prior to being used to construct equalizer filters. Also, since the linearization that is the basis for the EKF used in the previous algorithm becomes questionable at low SNRs, a similar approach that avoids the linearization is used to obtain an augmented KF. This new CIR estimation also independently controls the radial and transverse variations of the CIR gains. Tuchler's LE is implemented as a multi-sensor equalizer and performed in two stages. However, a new procedure is developed for the second stage, which performs the computation of symbol a posteriori messages and updates EOMP estimates as a joint message-passing algorithm. Since the array sensors do not all have the same SNR, in order to obtain a scalar value to describe the SNR of the entire array, the receiver input SNR is defined as the array matched-filter *output* SNR – which is equal to the sum of the SNRs on the individual sensors. The data was obtained by combining communication sequences with ambient noise recordings to yield input SNRs of 6 and 7 dB, and results are presented for processing data from subarrays of 5 and 10 sensors, taken from the top, middle, and bottom of the full 32-sensor VRA. The algorithm performed successfully at 7 dB for the 10-sensor arrays and at 6 and 7 dB for the 5-sensor arrays.

2

An Iterative Equalization and Decoding Approach for Underwater Acoustic Communication

We present an iterative approach for recovering information sent over a shallow underwater acoustic communication channel. The procedure has three main tasks: estimation of channel model parameters (CMPs), channel equalization, and decoding. These tasks are performed cyclicly until the algorithm converges. Information bits are convolutionally encoded, punctured and permuted, mapped into QPSK symbols, linearly modulated, and transmitted through a downward-refracting ocean waveguide. Training symbols are prepended to the transmitted sequence for initial estimation of CMPs. Our algorithm processes data from a single receive sensor. Data is received on a vertical array and the performance of the algorithm for each sensor in the array is examined. There is negligible Doppler spread in the received data. However, difference between transmitter and receiver clocks, as well as slight motion of the receive array, produce a non-negligible compression of the received signals. Consequently there is observable Doppler "shift". Nonuniform resampling of the data produces time series we model as the output of a linear time-invariant system. Resampling and CMP estimation are done iteratively, in conjunction with equalization and decoding. The algorithm successfully processes the data to yield few or no information bit errors.

2.1 Introduction

Each communication channel has key characteristics that drive the design of information transmission systems. The shallow underwater acoustic (UWA) channel has three. The first is relatively long channel impulse responses (CIRs) which, for the data rate used in this paper, causes intersymbol interference (ISI) to span tens of symbol periods. The second is sensitivity to Doppler effects, since the ocean is an undulating medium and has a fairly slow sound speed. The third is limited bandwidth, a result of the fact that signal attenuation increases with the square of frequency. Details of the UWA channel and a survey of the efforts to communicate through it can be found, for example, in [1] and [27–30]. The severity of these characteristics and the degree to which they impact UWA telemetry depends upon factors such as range, depth, surface interaction and conditions, sound speed depth profile, ocean floor properties, and so forth. For the data used in this paper, there is no source or receiver motion other than that which is unavoidable and a downward-refracting sound speed depth profile limits surface interaction.

It is commonly accepted that most UWA channels are "rapidly fading". Owing to factors mentioned above, and in particular when propagating signals have significant interaction with rough surface waves, this can be true. (Additionally, source/receiver motion will create a time-varying channel.) However, Yang has recently proposed that much of the perceived ocean fluctuation is in fact caused by algorithmic rather than physical phenomena [31]. While processing data transmitted through channels found by measurement to be slowly varying, he observed adaptive equalizer filters varying rapidly and giving the appearance of rapid channel fluctuations. The data used in this paper falls into such a category. If the data is "correctly" resampled to compensate for sampling clock error and source/receiver drift, then a channel model that assumes a fixed CIR is successfully employed. Failure to resample the data, or erroneous resampling, produces an apparent phase rotation of the CIR.

The majority of work in UWA communication is based on a general time-varying channel model and is designed for high SNR [32]. Equalization and CIR estimation are typically performed with the use of adaptive filters that often introduce more degrees of freedom than necessary. Furthermore, it is customary to assume that equalizers produce symbol estimates that are error-free and can be used as if they are the true symbols. Most schemes rely on correct symbol estimates for use in updating estimates of channel model parameters (CMPs) or for cancellation of ISI in decision feedback equalizers (DFEs). Additionally, the proposed receiver structures predominantly adhere to a canonical sequential processing strategy of equalization followed by decoding, with each operation performed a single time. The majority of radio communication literature either does not deal with significant ISI or circumvents the issue by using modulation formats that do not require equalization–for example, orthogonal frequency-division multiplexing

(OFDM) [33] and impulse radio [34].

In this paper we pursue alternatives to the conventional approaches. We take advantage of the true nature of our data and work within a channel model that captures the salient features of the data without excessive over-parameterization. Since we attempt to communicate at relatively low SNR, we do not assume symbol estimates are correct. Instead, we employ message-passing (MP) techniques and iterative processing [16]. Imperfect symbol knowledge is explicitly modeled and accounted for, and equalization and decoding are performed multiple times in an iterative or "turbo" manner [35, 36]. For other examples of iterative equalization and decoding applied to UWA channels the reader is refered to [37–42].

The shift from sequential to iterative processing is becoming widespread. The overwhelming success of Turbo codes, low-density parity check (LDPC) codes, Turbo equalization, and so forth, make a strong case for the use of iterative algorithms [12, 13, 15]. While initially developed for the decoding of channel codes, current research in iterative processing focuses on incorporating MP techniques into equalization and CMP estimation. That is the basic idea pursued here: decoding provides information about the transmitted symbols that can be used to improve estimation of CMPs and equalization, which can then be used to improve decoding, and so on.

Optimal equalization in the case of a known CIR with a short span is well established [43]. But the CIR of interest here is neither known nor of short duration. Various modifications of the optimum strategy have been proposed to deal with these shortcomings–notably "per survivor processing" [17]. Since it is not clear whether these approaches out-perform simpler ones that employ linear filters, one of the latter is adopted in this paper. In particular, we apply concepts pioneered by Tuchler et al [22, 23, 44] to data transmitted through an UWA channel. We use essentially the same equalizer filter, and use the same approximations to derive it, yet our derivation is simpler as it is a direct application of the "sum-product" algorithm (SPA) [6]. Furthermore, our computation of symbol a posteriori probabilities (APPs) is different and more accurate. Tuchler and others have also incorporated the use of "soft" symbol information into CIR estimation [20, 21, 45]. We follow suit, but apply the concept to a different channel model than the ones they consider.

In addition to implementing these "soft-in/soft-out" [46] equalization methods, we introduce new procedures in our receiver algorithm. Most documented work in UWA communication does not address the details of synchronization, despite the fact channel estimation is discussed. However, synchronization and CIR estimation are inextricably linked: the latter involves measuring not only the gains of impulse arrivals but their delays as well, which is precisely the subject of the former. We implement a method of simultaneous synchronization/CIR estimation that we call “framing”. Our method of phase sequence estimation is unconventional since standard approaches employing phase-locked loops (PLLs) require high SNR. We note the growing research in MP approaches to phase estimation and iterative timing recovery offer alternate approaches [47–50].

The remainder of this paper is organized as follows. In Section II we describe the Focused Acoustic Fields 2004 experiment, from which we obtained our data, and state our signal and channel models and assumptions. The receiver algorithm is then explained in Section III. In Section IV we present results from processing data collected during the experiment. We conclude the paper in Section V.

2.2 Experiment Setup, Transmission Model and Assumptions

2.2.1 Experiment Setup

The experiment was conducted during July 2004 in a shallow water region north of Elba Island, Italy. The experiment geometry is summarized in Fig. 2.1(b). A source at 110 m depth transmitted signals to a sensor array 10 km distant, through an ocean region 120 m deep. There were 32 elements in the sensor array, numbered from the bottom. The first sensor was at 104 m depth, there was 2 m spacing between sensors, and the 32nd sensor was at 42 m depth. Fig. 2.1(a) shows the measured sound speed versus depth. The experiment region was downward-refracting and, since the range was much greater than the depth, signal energy arriving at the array had little surface interaction and was concentrated toward the bottom of the sensor array.

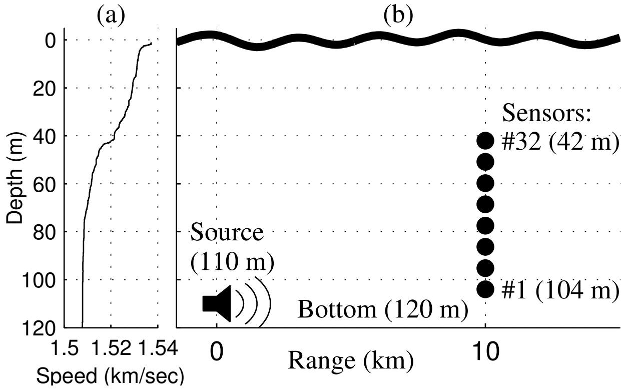


Figure 2.1: (a) Measured sound speed depth profile. (b) Experiment geometry.

2.2.2 Encoding, Permuting/Puncturing, and Mapping

The process of transforming bits b into symbols a is summarized in the Forney-style factor graph (FFG) [7, 8] of Fig. 2.2(a). Information bits $b = (b_0, \ldots, b_{N_b-1})$, $N_b = 10,000$, are encoded into bits $e = \left(e_0^{(1)}, e_0^{(2)}, \ldots, e_{N_b-1}^{(1)}, e_{N_b-1}^{(2)}\right)$ using a rate-$\frac{1}{2}$ convolutional code (CC) with maximum free distance and constraint length 5 [51]. These bits are pseudo-randomly (PR) punctured and permuted: $e \xrightarrow{\Pi} \acute{e} = \left(\acute{e}_0^{(1)}, \acute{e}_0^{(2)}, \ldots, \acute{e}_{N_d-1}^{(1)}, \acute{e}_{N_d-1}^{(2)}\right)$, $N_d = 9,800$ being the number of data-bearing symbols after QPSK mapping. The puncturing and permutation, denoted by Π, is a variation of an "S-random" permutation [52] with $S = 50$. At most one bit of each pair $\left(e_n^{(1)}, e_n^{(2)}\right)$ is punctured. Each pair $\left(\acute{e}_n^{(1)}, \acute{e}_n^{(2)}\right)$ is mapped into $a_{N_t+n} = \left(1/\sqrt{2}\right)\left[\left(2\acute{e}_n^{(1)} - 1\right) + \mathrm{j}\left(2\acute{e}_n^{(2)} - 1\right)\right] \in \mathcal{A}$, where $\mathcal{A}$ is the QPSK alphabet. Inserting $N_t = 200$ PR QPSK training symbols before the data symbols gives the transmitted symbol sequence $a = (a_0, \ldots, a_{N_s-1})$, where $N_s = N_t + N_d = 10,000$ is the total number of symbols per transmission. The symbols have zero mean, unit variance and, by virtue of the PR permutation, they are uncorrelated as well.

The scheme just described for generating a symbol sequence is statistically equivalent to that used during the experiment. Yet there is a subtle difference we now explain. The procedure stated above requires selecting the number of training symbols prior to transmission. However, we chose not to commit to a particular number of training symbols (or a specific coding scheme), so we transmitted an uncoded PR QPSK symbol sequence $\mathring{a}$. According to the symbol mapping given above, this symbol sequence corresponds to an effective punctured and permuted encoded bit sequence $\mathring{e}$. In general there is no information bit sequence and/or puncturing/permutation that will produce $b \xrightarrow{\text{CC}} e \xrightarrow{\Pi} \acute{e} = \mathring{e}$. Consequently we simply generate a PR information bit sequence, encode, puncture and permute it as stated above, and define the set $\mathcal{S} = \left\{(n, j) : \acute{e}_n^{(j)} \neq \mathring{e}_n^{(j)}\right\}$. This set represents the elements of $\mathring{e}$ that must be bit-complemented to give $\acute{e}$, and vice versa. If we insert a "scrambling" operation after the puncturing/permutation and before the mapping, which complements the bits represented by $\mathcal{S}$, then the system will transform a given block of information bits into the symbol sequence actually transmitted during the experiment. This additional operation has no effect from a statistical standpoint, and is easily incorporated in a MP algorithm.

2.2.3 Signal Models and Assumptions

In this section we state the channel model and assumptions that drive our information recovery algorithm. The data is modeled as the nonuniformly sampled output of a linear time-invariant (LTI) system. We assume, with details discussed in Appendix 2A , that proper

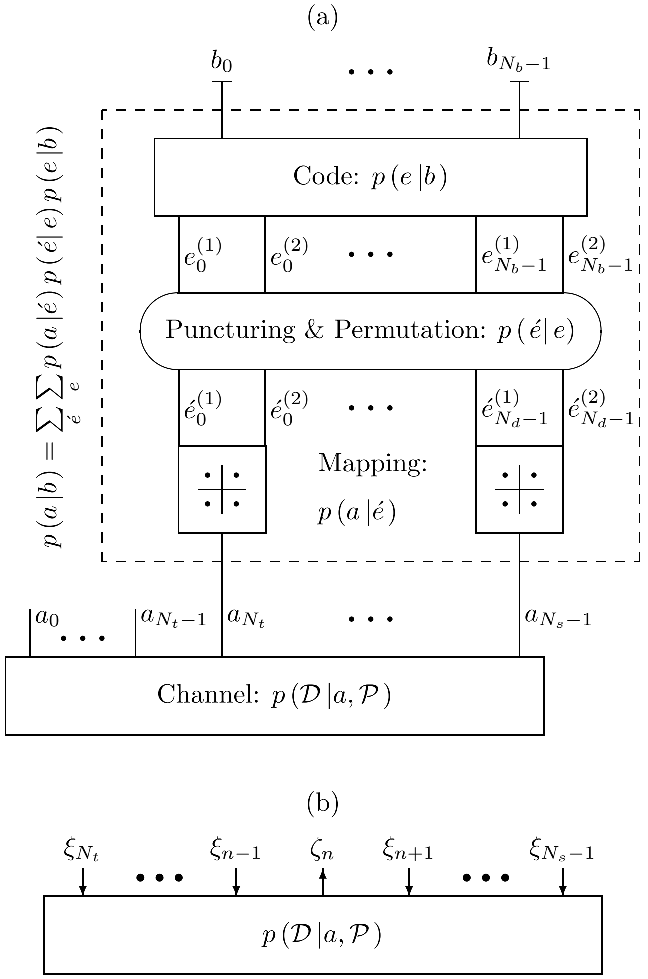

(a)
b_0
b_{N_b-1}
Code: $p\left(e\,|b\right)$
$e_0^{(1)}$
$e_0^{(2)}$
$e_{N_b-1}^{(1)}$
$e_{N_b-1}^{(2)}$
Puncturing & Permutation: $p\left(\acute{e}|\,e\right)$
$\acute{e}_0^{(1)}$
$\acute{e}_0^{(2)}$
$\acute{e}_{N_d-1}^{(1)}$
$\acute{e}_{N_d-1}^{(2)}$
Mapping:
$p\left(a\,|\acute{e}\right)$
$p\left(a\,|b\right) = \sum_{\acute{e}} \sum_{e} p\left(a\,|\acute{e}\right) p\left(\acute{e}|\,e\right) p\left(e\,|b\right)$
a_0
a_{N_t-1}
a_{N_t}
a_{N_s-1}
Channel: $p\left(\mathcal{D}\,|a,\mathcal{P}\right)$
(b)
ξ_{N_t}
ξ_{n-1}
ζ_n
ξ_{n+1}
ξ_{N_s-1}
$p\left(\mathcal{D}\,|a,\mathcal{P}\right)$

Figure 2.2: (a) Factor graph. (b) Equalizer message-passing diagram.

resampling/interpolation yields Doppler-compensated data that corresponds to a LTI system. Furthermore, we assume this resampling/interpolation procedure, which we denote using the operator $\mathfrak{R}$, can be performed with essentially no loss of information. Consequently, knowledge of the data sampling times is required. We can only use estimates, however, whose error leads to a phase-rotation of the Doppler-compensated data. (This is also discussed in Appendix 2A .) Finally, we explain our model and assumptions for this phase sequence, and for the CIR as well.

The sequence $\{2\mathrm{Re}\,[\tilde{s}_k^{\mathrm{Tx}}]\}$ is generated from the symbol sequence a, then converted to an analog signal and transmitted from a single source transducer. (We use the notation $\{x_k\}$ to denote the sequence $(\ldots, x_k, \ldots)$, where the initial and terminal indices are unspecified; the notation does not indicate the set with single element x_k.) We use linear modulation

$$\tilde{s}_k^{\mathrm{Tx}} \quad = \quad \exp\left(\mathrm{j}2\pi \frac{f_c}{F_s} k\right) \sum_l a_l h_{k-(F_s/R_s)l}^{\mathrm{Tx}}$$

with center frequency $f_c = 3.5$ kHz, sampling frequency $F_s = 12$ kHz, symbol rate $R_s = 1$ symbol/msec, and transmit pulse h_k^{Tx}. We used a pulse whose Fourier transform is a least-squares (LS) fit to a square root raised-cosine spectrum [53], having approximately 99% of its energy within a bandwidth equal to the symbol rate. Transmission through the UWA channel, sampling the received signal at a single hydrophone sensor, and filtering to remove out-of-band noise and negative-frequency signal components, results in a data sequence $\tilde{r}$ whose elements we model as

$$\tilde{r}_k \quad = \quad \tilde{w}_k + \tilde{s}\,(t_k) \tag{2.1}$$

$$\tilde{s}\,(t) \quad = \quad \exp\left(\mathrm{j}2\pi f_c t\right) \sum_l a_l h\left(t - l\,/R_s\,\right) \tag{2.2}$$

$$t_{k_0+k} \quad = \quad (1+\nu)\,\frac{k}{F_s} + \tau_k \tag{2.3}$$

where $\tilde{w}_k$ is ambient noise, $h\,(t)$ is an overall CIR, and k_0 is a reference index for the sequence of sampling times $\mathcal{T} = \{t_k\}$. The parameters ν and $\{\tau_k\}$ account for error between transmitter and receiver clocks (which is indistinguishable from Doppler "shift"), slight motion of the sensor array, and fluctuations in the ocean channel.

Fig. 2.3 shows SNR measured across the sensor array and CMP estimates for sensors 3, 21, and 32, obtained using all $N_s = 10,000$ symbols. These three sensors are typical representatives from the bottom, middle and top of the array; sensors 21 and 32 will be examined closely throughout the sequel. As sensor number increases (depth decreases), SNR tends to decrease and the CIR becomes more complicated. For sensor 3, near the ocean floor, the CIR has a dominant arrival and a short duration, while for sensor 32, at the top of the array, the CIR does not have a dominant arrival and has a much longer duration. The CIR estimates are at symbol-rate (SR)

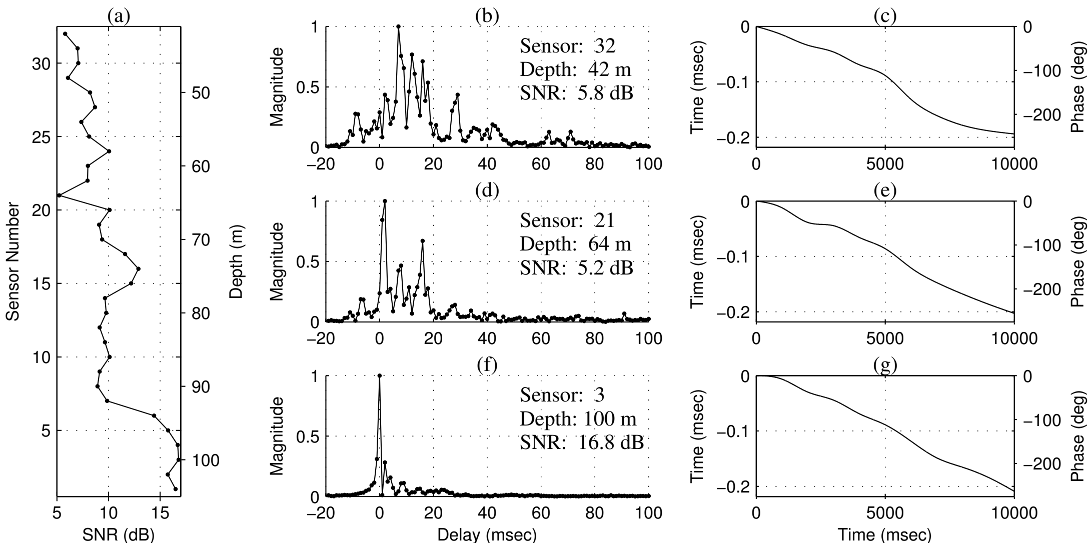


Figure 2.3: (a) SNR measured across entire array. Left axis is sensor number, right axis is depth. (b) CIR estimated for sensor 32. (c) Estimated sampling time offset $\hat{\delta}_k = \hat{t}_{k_0+k} - k\,/F_s$ for sensor 32. Right axis shows corresponding phase rotation $2\pi f_c \hat{\delta}_k$. (d,e) CIR and sampling time offset for sensor 21. (f,g) CIR and sampling time offset for sensor 3.

and all three are relative to the same synchronization, but their values are normalized independently. Also shown in the figure are estimates of the sampling time offsets $\delta_k = t_{k_0+k} - k\,/F_s$. Choosing not to resample a received data sequence would result in its CIR shifting by about 0.2 symbols over the duration of the transmission, and rotating by about 250°. These effects result from the fact

$$\begin{aligned}\tilde{s}\left(t_{k_0+k}\right) &= \tilde{s}\left(\frac{k}{F_s}+\delta_k\right)\\ &= \exp\left(\mathrm{j}2\pi\frac{f_c}{F_s}k\right)\sum_l a_l\left[\exp\left(\mathrm{j}2\pi f_c\delta_k\right)h\left(\delta_k+\frac{k}{F_s}-\frac{l}{R_s}\right)\right]\end{aligned}$$

.

We will use the notation $\acute{x}_k = \mathfrak{R}\left(\{x_k\},\{t_k\},\acute{t}_k\right)$ as follows. The input sequence $\{x_k\}$ is assumed to be the values of an underlying continuous function $x(t)$ sampled at times $\{t_k\}$. The output value $\acute{x}_k$ is the approximation of $x\left(\acute{t}_k\right)$ obtained via resampling/interpolation as discussed in Appendix 2A .

Doppler compression/expansion of the communication signal does not significantly change its frequency band. Since the bandwidth is nearly equal to the symbol rate, we only need the SR signal samples $\tilde{s}\left(n\,/R_s\right)$. Given the sampling times $\mathcal{T}$, we assume no loss of information in resampling/interpolating the data $\tilde{r}$ to obtain the following model:

$$\begin{aligned} r_n(\mathcal{T}) &= \exp\left(-\mathrm{j}2\pi \frac{f_c}{R_s} n\right) \Re\left(\tilde{r}, \mathcal{T}, \frac{n}{R_s}\right) && (2.4) \\ &\cong w_n + \sum_{l=l_1}^{l_2} h_l a_{n-l} && (2.5) \\ p(\mathcal{D}\,|a, \mathcal{P}) &\equiv \prod_n p_w\left(r_n(\mathcal{T}) - \sum_{l=l_1}^{l_2} h_l a_{n-l} \middle| \sigma_w^2\right) && (2.6) \end{aligned}$$

where w_n is additive white Gaussian noise (AWGN) with probability density function (pdf)

$$p_w\left(w_n\,|\sigma_w^2\right) = \left(\pi\sigma_w^2\right)^{-1} \exp\left(-\left|w_n\right|^2 \middle/ \sigma_w^2\right)$$

In (2.6) we consider using the pdf of the SR data $\{r_n(\mathcal{T})\}$ equivalent to using the pdf of the original data $\tilde{r}$. Therefore the CMPs are the sampling times $\mathcal{T}$, the CIR which has values h_l and is nonzero only for $l_1 \leq l \leq l_2$, and the AWGN variance σ_w^2.

We use coherent signaling, which presumes the overall channel varies relatively slowly. We incorporate this into our sampling times model in two ways. For the initial estimate, we assume $\mathcal{T}$ is piecewise-linear over blocks that span $N_\Delta = 1000$ symbols. During the iterative procedure, we update the estimate of $\mathcal{T}$ by measuring the phase sequence resulting from error in the estimation of $\mathcal{T}$ (see Appendix 2A). This phase sequence is described using cubic splines (CS) with control points placed every N_Δ symbols. Therefore, we effectively assume channel fluctuations occur at frequencies below 1 Hz. Additionally, an upper bound on the CIR length is needed, which we set to $L^{\max} = 150$ symbols.

2.3 Receiver Algorithm

The overall objective is recovery of information bits b_n. Denoting the data by $\mathcal{D}$ and the CMPs by $\mathcal{P}$, we would like to compute $\hat{b}_n = \arg\max_{b_n}\left[\max_{\mathcal{P}} p(\mathcal{D}\,|b_n, \mathcal{P})\right]$. Since exact computation is impossible we use MP techniques to obtain approximate solutions. Details of iterative equalization and decoding can be found in references given in the introduction. The main idea is that information–which can be interpreted as estimates of symbol probability mass functions (pmf's)–is cycled between an equalizer and a decoder, and the information quality improves as the process is iterated. (By "decoder" we mean everything that takes place within the dashed box in the FFG of Fig. 2.2(a).) We focus on performing equalization in conjunction with CMP estimation; standard MP routines are used to perform decoding and are straightforward applications of the SPA [9, 19]. Specifically, representing messages from the decoder by ξ and from the equalizer by ζ, we present approximations to the following:

$$\begin{aligned}
\hat{\mathcal{P}}^{(i)} &= \arg\max_{\mathcal{P}} p\left(\mathcal{D}\left|\mathcal{P}\right.\right) \\
&\cong \arg\max_{\mathcal{P}} \sum_{a} p\left(\mathcal{D}\left|a, \mathcal{P}\right.\right) \prod_{n} \xi_n^{(i-1)}\left(a_n\right) \qquad (2.7) \\
\zeta_n^{(i)}\left(\alpha\right) &\propto p\left(\mathcal{D}\middle|a_n = \alpha, \hat{\mathcal{P}}^{(i)}\right) \\
&\cong \sum_{\{a:a_n=\alpha\}} p\left(\mathcal{D}\middle|a, \hat{\mathcal{P}}^{(i)}\right) \prod_{l\neq n} \xi_l^{(i-1)}\left(a_l\right) \qquad (2.8)
\end{aligned}$$

where i is the iteration count. There is no prior symbol information available for the first iteration, i.e. $\xi_n^{(0)}\left(\alpha\right) = 1\left/|\mathcal{A}|\right.$ for $n = N_t, \ldots, N_s - 1$, so the initial CMP estimates $\hat{\mathcal{P}}^{(1)}$ are obtained using only the training symbols $a_0, \ldots, a_{N_t-1}$.

A flow chart of the receiver algorithm is shown in Fig. 2.4. Asynchronous transmissions are assumed, so the first step is to detect when a communication signal is present (vice only ambient noise). This process also provides coarse synchronization. The training symbols $\{a_n\}_{n=0}^{N_t-1}$ are then used to fine-tune synchronization and obtain an initial CIR estimate $\hat{h}^{(1)}$ and AWGN variance estimate $\left(\hat{\sigma}_w^{(1)}\right)^2$. A resampling/interpolation reference index k_0 is also chosen. These parameter estimates are used to compute a linear equalizer (LE) filter $f^{(1)}$, which is used to process SR data and obtain an initial sampling times estimate $\hat{\mathcal{T}}^{(1)}$. We then set the iteration $i = 1$ and begin the iterative procedure.

Resampling $\tilde{r}$ according to $\hat{\mathcal{T}}^{(i)}$, and shifting the resulting sequence to baseband, gives SR data $\left\{r_n^{(i)}\right\}$. This data sequence is filtered with $f^{(i)}$ to produce soft symbol estimates $\left\{z_n^{(i)}\right\}_{n=0}^{N_s-1}$, from which we compute symbol APP estimates $\zeta^{(i)}$. If $i = N_{\text{iter}}$, then the symbol APP estimates are decoded to give bit decisions $\hat{b}$ and the process is complete. Otherwise we decode the symbol APP estimates to produce symbol prior probability estimates $\xi^{(i)}$. We set $i \leftarrow i + 1$ and use $\xi^{(i-1)}$ to compute the symbol statistics

$$\begin{aligned}
\bar{a}_n^{(i)} &= \sum_{\alpha\in\mathcal{A}} \alpha\xi_n^{(i-1)}\left(\alpha\right) \\
\left(\sigma_{\bar{a}}^{(i)}\right)^2 &= 1 - \frac{1}{N_s - N_t} \sum_{n=N_t}^{N_s-1} \left|\bar{a}_n^{(i)}\right|^2
\end{aligned}$$

We then compute the CIR estimate $\hat{h}^{(i)}$. If our confidence in the symbol prior probability estimates is high enough–i.e. if the average estimated symbol variance $\left(\sigma_{\bar{a}}^{(i)}\right)^2$ is low enough–then we estimate the phase rotation $\{\theta\left(n\left/R_s\right.\right)\}$ resulting from error in the estimated sampling times $\hat{\mathcal{T}}^{(i-1)}$, which allows us to compute $\hat{\mathcal{T}}^{(i)}$; otherwise we set $\hat{\mathcal{T}}^{(i)} = \hat{\mathcal{T}}^{(i-1)}$. Finally, we estimate the AWGN variance $\left(\hat{\sigma}_w^{(i)}\right)^2$ and construct a new LE $f^{(i)}$.

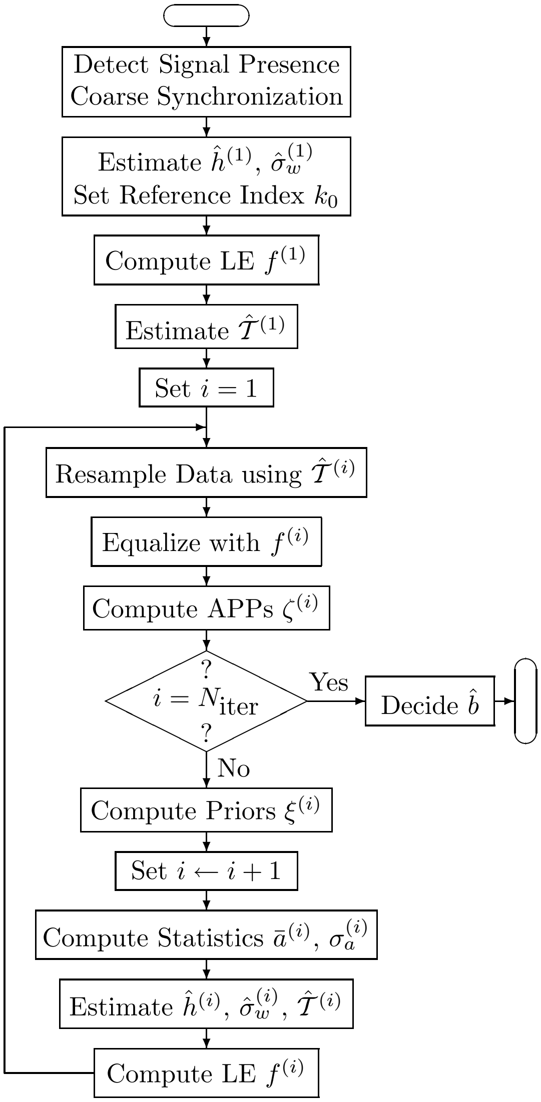

Detect Signal Presence
Coarse Synchronization
Estimate $\hat{h}^{(1)}$, $\hat{\sigma}_w^{(1)}$
Set Reference Index k_0
Compute LE $f^{(1)}$
Estimate $\hat{\mathcal{T}}^{(1)}$
Set $i = 1$
Resample Data using $\hat{\mathcal{T}}^{(i)}$
Equalize with $f^{(i)}$
Compute APPs $\zeta^{(i)}$
$i \overset{?}{\underset{?}{=}} N_{\text{iter}}$
Yes
Decide $\hat{b}$
No
Compute Priors $\xi^{(i)}$
Set $i \leftarrow i + 1$
Compute Statistics $\bar{a}^{(i)}$, $\sigma_a^{(i)}$
Estimate $\hat{h}^{(i)}$, $\hat{\sigma}_w^{(i)}$, $\hat{\mathcal{T}}^{(i)}$
Compute LE $f^{(i)}$

Figure 2.4: Information recovery flow chart.

2.3.1 Detection of Signal Presence and Coarse Synchronization

The model described by (2.5) applies when a resampling/interpolation reference index k_0 has been established. It incorporates fine synchronization into the CIR definition. During the coarse synchronization and training procedures, however, it is convenient to use a different but equivalent LTI model:

$$r^{(0)}(n) = \begin{bmatrix} r^{(0)}_n & \cdots & r^{(0)}_{n+N_t-1} \end{bmatrix}^{\mathrm{T}} \tag{2.9}$$

$$= w(n) + A(L)\,h \tag{2.10}$$

$$A(L) = \begin{bmatrix} a_0 & & \\ \vdots & \ddots & \\ a_{L-1} & \cdots & a_0 \\ \vdots & & \vdots \\ a_{N_t-1} & \cdots & a_{N_t-L} \end{bmatrix} \tag{2.11}$$

$$h = \begin{bmatrix} h_0 & \cdots & h_{L-1} \end{bmatrix}^{\mathrm{T}} \tag{2.12}$$

where $r^{(0)}_n = \exp\left(-\mathrm{j}2\pi\left(f_c/R_s\right)n\right)\mathfrak{R}\left(\tilde{r}, \left\{k/F_s\right\}, n/R_s\right) = \exp\left(-\mathrm{j}2\pi\left(f_c/R_s\right)n\right)\tilde{r}_{(F_s/R_s)n}$, $w(n)$ is noise, and the CIR is causal and L symbols long. (The entries in the upper right corner of the symbol convolution matrix $A(L)$ are zero.)

To establish coarse synchronization we pose a binary hypothesis test: the vector $r^{(0)}(n)$ is either the start of the communication sequence or it is just noise. The data pdf used to form the test statistic is

$$\begin{aligned} p\left(r^{(0)}(n)\middle|\mathcal{H}_\eta\right) &\propto \int p\left(r^{(0)}(n)\middle|\mathcal{H}_\eta, h\right)p(h)\,\mathrm{d}h \\ &\propto \int \exp\left(-\left\|r^{(0)}(n) - A(L)\,h\right\|^2/\sigma_w^2 - \|h\|^2/\sigma_h^2\right)\mathrm{d}h \\ &\propto \exp\left(-\left\|r^{(0)}(n)\right\|^2_{\left[\sigma_w^2\mathrm{I}+\sigma_h^2A(L)A(L)^{\mathrm{H}}\right]^{-1}}\right) \end{aligned}$$

where $\mathcal{H}_\eta$ are the hypotheses and the prior pdf of the CIR vector assumes uncorrelated zero-mean Gaussian elements with variance σ_h^2. (The notation $\|v\|^2_M$ indicates the quadratic form $v^{\mathrm{H}}Mv$.) The null hypothesis $\mathcal{H}_0$ corresponds to setting $A(L) = 0$. Consequently the log-likelihood ratio (LLR) is $\left\|r^{(0)}(n)\right\|^2_M$, where $M = \mathrm{I}/\sigma_w^2 - \left[\sigma_w^2\mathrm{I} + \sigma_h^2A(L)\,A(L)^{\mathrm{H}}\right]^{-1}$, and the optimal test consists of comparing this LLR with a suitable threshold. To keep computational complexity reasonable we use various approximations for M. The crudest approximation is simply $M \overset{\sim}{\propto} \mathrm{I}$. We can use the matrix inversion lemma to write $M = \left(\sigma_h^2/\sigma_w^2\right)A(L)\left[\sigma_w^2\mathrm{I} + \sigma_h^2A(L)^{\mathrm{H}}A(L)\right]^{-1}A(L)^{\mathrm{H}}$, which results in the approximation $M \overset{\sim}{\propto} A(L)\,A(L)^{\mathrm{H}}$. This latter approximation is most useful if we use the augmented matrix

$$\breve{A}\left(L\right) \quad = \quad \begin{bmatrix} a_0 & & \\ \vdots & \ddots & \\ a_{L-1} & \cdots & a_0 \\ \vdots & & \vdots \\ a_{N_t-1} & \cdots & a_{N_t-L} \\ & \ddots & \vdots \\ & & a_{N_t-1} \end{bmatrix} \tag{2.13}$$

so that the LLR, which becomes $\left\|\breve{A}\left(L\right)^{\mathrm{H}} r^{(0)}\left(n\right)\right\|^2$, can be calculated via correlation of the data sequence $r_n^{(0)}$ with the training sequence. (The entries in the upper right corner and lower left corner of $\breve{A}\left(L\right)$ are zero.) Since the threshold is proportional to the noise variance, we incorporate estimation of the latter by expressing the former as $\gamma \left\|\acute{r}^{(0)}\left(n\right)\right\|^2$, where $\acute{r}^{(0)}\left(n\right) = \begin{bmatrix} r_{n-N_g}^{(0)} & \cdots & r_{n-1}^{(0)} \end{bmatrix}^{\mathrm{T}}$ for some guard interval N_g. For a given probability of false alarm $p_{\mathrm{FA}}^{\mathrm{DOSP}}$, the parameter γ is the solution of $p_{\mathrm{FA}}^{\mathrm{DOSP}} = \mathrm{P}\left[\Phi_n \geq \gamma \left| r_n^{(0)} \overset{\mathrm{iid}}{\sim} p_w\left(r_n^{(0)} \left|\sigma_w^2\right.\right)\right.\right]$, where we define the test statistic (TS) $\Phi_n = \left\|r^{(0)}\left(n\right)\right\|_M^2 \Big/ \left\|\acute{r}^{(0)}\left(n\right)\right\|^2$.

We used the crude approximation $M \mathrel{\tilde{\propto}} \mathrm{I}$ and set $N_g = 100$, $N_t = 100$ (just for this procedure: $N_t = 200$ during training, etc.), and $p_{\mathrm{FA}}^{\mathrm{DOSP}} = 10^{-9}$, to generate Fig. 2.5. The figure shows $\left|r_n^{(0)}\right|$, $10\log_{10}\left(\Phi_n / \gamma\right)$, and a CIR estimate obtained using all $N_s = 10,000$ symbols, for sensor 32. The TS is low prior to the start of the communication signal, reaches a peak that coincides with the main energy of the CIR estimate, then becomes low again after the start of the communication signal. It becomes low again because inside the communication signal the two data vectors described above are both noise-like–albeit with higher variance–and the TS is independent of the (ambient) noise variance. Consequently we obtain coarse SR synchronization as follows: the TS becomes positive for some index n_1, we then find an index $n_2 > n_1$ for which the TS is negative again, finally we compute $\check{n}_0 = \arg\max_{n_1 \leq n < n_2} \Phi_n$. (An inverted hat over a variable, as used here and henceforth, indicates a coarse estimate of that variable.)

We note that using the approximation $M \propto \breve{A}\left(L\right)\breve{A}\left(L\right)^{\mathrm{H}}$ yields a stronger, more reliable test. However, the crude approximation works sufficiently well in the range of SNR we are considering and is very simple to implement.

2.3.2 CMP Estimation via Training

We use the model and notation introduced in the previous subsection. If the CIR length L were known, we could estimate the unknown parameters using the maximum likelihood

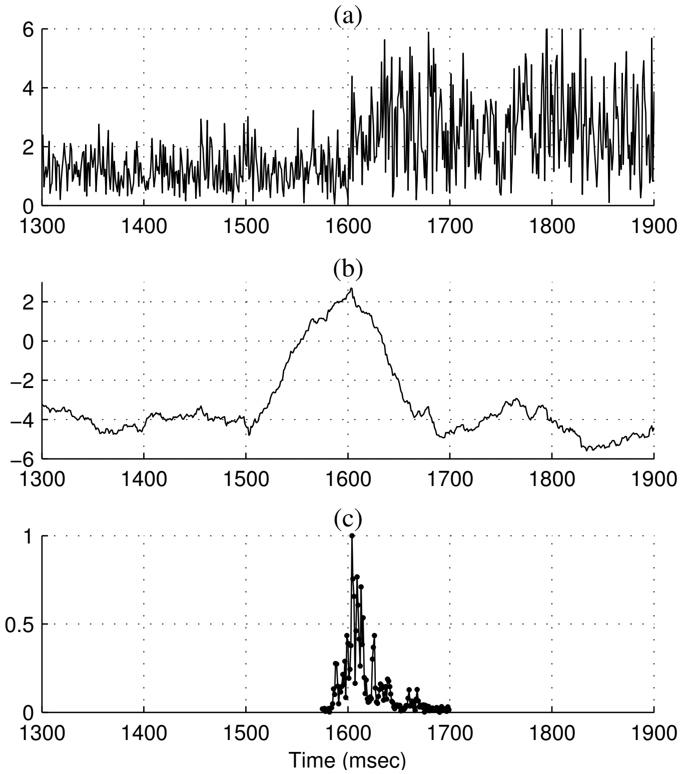


Figure 2.5: (a) Magnitude of sensor 32 initial SR data $\left|r_n^{(0)}\right|$. (b) Hypothesis test statistic. (c) Magnitude of CIR estimated using all $N_s = 10,000$ symbols.

approach–i.e. by maximizing the pdf

$$p\left(r^{(0)}\left(n\right)\left|A\left(L\right),h,\sigma_w^2\right.\right) \;=\; \left(\pi\sigma_w^2\right)^{-N_t}\exp\left(-\left\|r^{(0)}\left(n\right)-A\left(L\right)h\right\|^2/\sigma_w^2\right) \qquad (2.14)$$

which is analogous to (2.6). With L unknown, it does not make sense to simply maximize over this parameter as well. For given values of n and L, (2.14) is maximized by setting $h=\hat{h}\left(n,L\right)=A\left(L\right)^{+}r^{(0)}\left(n\right)$, where $\left(\cdot\right)^{+}$ denotes pseudo-inverse, which gives

$$p\left(r^{(0)}\left(n\right)\left|A\left(L\right),\hat{h}\left(n,L\right),\sigma_w^2\right.\right) \;=\; \left(\pi\sigma_w^2\right)^{-N_t}\exp\left(-\left\|\left[\mathrm{I}-A\left(L\right)A\left(L\right)^{+}\right]r^{(0)}\left(n\right)\right\|^2/\sigma_w^2\right)$$

Consequently (2.14) is maximized by making the subspace spanned by the columns of $A\left(L\right)$ as large as possible, which means that L should be made as large as possible. The problem with doing so, however, is that estimation error increases as L increases. This issue is not important when the number of training symbols is much larger than the anticipated CIR length. But when the training interval is relatively short, as in our case, then choosing L too large will project too much noise onto the CIR estimate, and choosing L too small will project too little signal energy onto the CIR estimate. We use the term "framing" to refer to the process of estimating the start of a CIR (i.e. synchronization), in addition to its span and the gains of its multipath arrivals.

In Appendix 2B we motivate a cost function (CF) that alleviates the problems associated with using (2.14):

$$\sigma_{\mathrm{eff}}^2\left(n,L\right) \;=\; \frac{\left\|\left[\mathrm{I}-A\left(L\right)A\left(L\right)^{+}\right]r^{(0)}\left(n\right)\right\|^2}{N_t-L}\left(1+\mathrm{tr}\left\{\left[A\left(L\right)^{+}A\left(L\right)^{+\mathrm{H}}\right]^{-1}\right\}\right) \qquad (2.15)$$

This expression is an effective noise variance. The data is modeled as $r_n=w_n+h_n*a_n$, where w_n is AWGN with variance σ_w^2, h_n is the CIR, a_n is the symbol sequence, and $*$ denotes convolution. The CIR is divided into two parts: one is estimated and the other is treated as interference. Consequently the data model becomes $r_n=\acute{w}_n+h_n^{\mathrm{sig}}*a_n$, where $\acute{w}_n=w_n+h_n^{\mathrm{int}}*a_n$. For values of (n,L) where the CF is near its minimum value, h_n^{int} contains only arrivals with relatively small gains. Consequently $h_n^{\mathrm{int}}*\left(h_{-n}^{\mathrm{int}}\right)^*\simeq\left\|h^{\mathrm{int}}\right\|^2\delta_n$ (Kronecker delta) and $\acute{w}_n$ is well appoximated as AWGN with variance $\sigma_w^2+\left\|h^{\mathrm{int}}\right\|^2$, where $\left\|h^{\mathrm{int}}\right\|^2=\sum_n\left|h_n^{\mathrm{int}}\right|^2$. The top line of (2.15) is an unbiased estimate of this noise-plus-interference variance, while the bottom line accounts for the projection of this effective noise onto the CIR estimate. Basically, some potential signal energy is sacrificed because its meager energy is outweighed by the additional error variance that would be incurred by estimating the corresponding multipath arrivals. Consequently we estimate the desired parameters by minimizing $\sigma_{\mathrm{eff}}^2\left(n,L\right)$. The upper term is made small by making the subspace spanned by the columns of $A\left(L\right)$ large, while the lower term is made small by keeping this subspace small–so we achieve the desired balance between projecting enough signal energy onto the CIR estimate, but not too much noise. To perform the minimization, we compute

$\sigma_{\text{eff}}^2(n, L)$ for $L \leq L^{\max}$ and for values of n in a window about the coarse SR synchronization index $\check{n}_0$:

$$\begin{aligned}
\left\{\hat{n}_0, \hat{L}\right\} &= \arg\min_{n,L} \sigma_{\text{eff}}^2(n, L) && (2.16)\\
\left(\hat{\sigma}_w^{(1)}\right)^2 &= \sigma_{\text{eff}}^2\left(\hat{n}_0, \hat{L}\right) && (2.17)\\
\hat{h} &= A\left(\hat{L}\right)^+ r^{(0)}(\hat{n}_0) && (2.18)
\end{aligned}$$

This procedure can be implemented efficiently using a QR decomposition of $A(L^{\max})$.

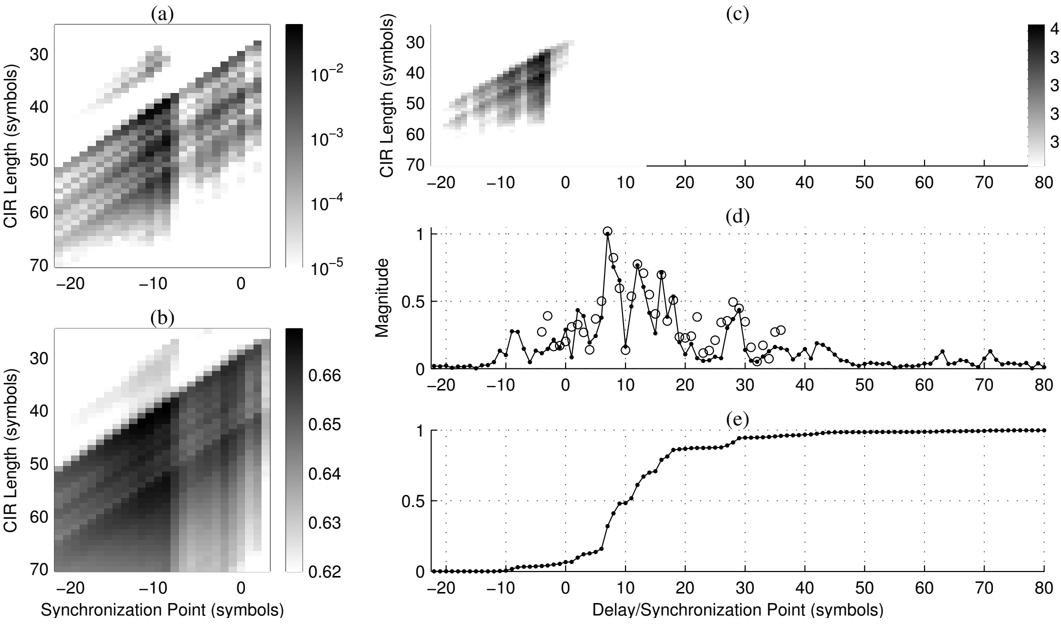


Figure 2.6: (a) Histogram of CIR length and synchronization point estimates, obtained from simulations using data generated from the CIR estimate of sensor 32 shown in Fig. 2.3(b). (b) Rate of correct symbol decisions that results from using a LE computed from CIR and noise variance estimates derived using various CIR lengths and synchronization points. (c) "Framing" CF (dB) as a function of CIR length and synchronization point for experiment data from sensor 32. (d) The corresponding CIR estimate is shown in circles, an estimate obtained using all $N_s = 10,000$ symbols is shown in dots connected by solid lines. (e) Cumulative energy of the latter CIR estimate.

We applied $\sigma_{\text{eff}}^2(n, L)$ in simulations using data generated from the sensor 32 CIR estimate of Fig. 2.3(b). Denote this "true" CIR as $\mathring{h}_l$ and its length as $\mathring{L}$. Each data sequence was generated as $r_{\mathring{n}_0+n} = w_{\mathring{n}_0+n} + \sum_{l=0}^{\mathring{L}-1} \mathring{h}_l a_{n-l}$, where $\mathring{n}_0$ is the true synchronization index, σ_w^2 was chosen to give an SNR of 5 dB, and the symbol sequence was $N_s = 10,000$ symbols long and contained the same training sequence used in the experiment. For each data sequence we computed $\left\{\hat{n}_0, \hat{L}\right\}$ from (2.16). The histogram resulting from one million trials is shown in Fig. 2.6(a). For one thousand of these trials, and for each pair $\{n_0, L\}$, we also did the following: we computed $\hat{h}\left(n_0, L\right) = A\left(L\right)^{+} r\left(n_0\right)$ and $\hat{\sigma}_w^2 = \sigma_{\text{eff}}^2\left(n_0, L\right)$, then filtered the data with an LE constructed as described in Appendix 2C , using $\sigma_a^2 = 1$ and substituting the CMP estimates for true parameter values. The average ratio of correct to total symbols for each n_0 and L is shown in Fig. 2.6(b). It is clear from the figure that the CF $\sigma_{\text{eff}}^2\left(n_0, L\right)$ selects n_0 and L so as to yield the best performance when using a LE computed from the resulting parameter estimates. In Fig. 2.6(c) we show $-10\log_{10}\left[\sigma_{\text{eff}}^2\left(n, L\right)\right]$ computed using experiment data from sensor 32. Part (d) of the figure shows the resulting CIR estimate (in circles), along with an estimate obtained using all $N_s = 10,000$ symbols. Part (e) of the figure shows the normalized cumulative sum of the energy of the latter CIR estimate. Comparing (d) and (e) shows the training CIR estimate, although quite noisy, spans an interval containing about 95% of the energy in the CIR.

We complete the training procedure by setting the resampling/interpolation reference sample k_0 to coincide with the center of the energy distribution of the CIR estimate, and setting the estimated CIR limits $\left(\hat{l}_1^{(1)}, \hat{l}_2^{(2)}\right)$ in a consistent manner:

$$\begin{aligned}
k_0 &= \frac{F_s}{R_s}\left(\hat{n}_0 + \bar{l}\right) && (2.19)\\
\bar{l} &= \left\lfloor \sum_{l=0}^{\hat{L}-1} l\left|\hat{h}_l\right|^2 \Big/ \left\|\hat{h}\right\|^2 \right\rceil && (2.20)\\
\left[\hat{h}^{(1)}_{\hat{l}_1^{(1)}}, \ldots, \hat{h}^{(1)}_{\hat{l}_2^{(1)}}\right]^{\mathrm{T}} &= \hat{h} && (2.21)\\
\hat{l}_1^{(1)} &= -\bar{l} && (2.22)\\
\hat{l}_2^{(1)} &= -\bar{l} + \hat{L} - 1 && (2.23)
\end{aligned}$$

(The notation $\lfloor\cdot\rceil$ denotes rounding the argument to the nearest integer.)

2.3.3 Initial Estimation of Received Signal Sampling Times

The sequence of sampling times $\mathcal{T}$ varies during a transmission, so it cannot be estimated via training. We use the model (2.3) for the initial estimate; we first estimate the linear trend ν, then the random fluctuations $\{\tau_k\}$. A CF of ν is developed, which is then minimized over a suitable set of values for ν. In particular, we used uniformly spaced values of ν with spacing 10^{-5}, which corresponds to a rotation of $\nu\left(360^\circ\right)\left(f_c/R_s\right)\left(N_s\right) = 126^\circ$ over the course of a transmission. Since ν is too small to be estimated accurately using $N_t = 200$ training symbols, we simply center our search about zero.

For each ν, let $\left\{\check{t}_k\left(\nu\right)\right\}$ be a rough estimate of the sequence of sampling times, with $\check{t}_{k_0+k}\left(\nu\right) = \left(1+\nu\right)k/F_s$. The equivalence (to within a complex scaling) of the minimum mean-squared error (MMSE) LE and the constant modulus (CM) LE (under the assumption of Gaussian-distributed LE output ISI) is established in [54]. Furthermore, the minimum variance distortionless response (MVDR) LE derived in Appendix 2C is simply a scaled version of the MMSE LE. Let z_n denote LE output. In [55] it is shown that the CM CF $\sum_n \left(1-|z_n|^2\right)^2$ is equivalent to the following CF:

$$J^{\mathrm{CM}} = \mu_4 \big/ \mu_2^2 \tag{2.24}$$

$$\mu_m = \frac{1}{N_s}\sum_{n=0}^{N_s-1} |z_n|^m \tag{2.25}$$

where μ_m is the sample mth-order absolute moment of the LE output. The important point is that the CM criterion and the MVDR criterion are consistent. Instead of using the CM criterion to construct a LE filter, we use it to measure the output of a LE created with the MVDR criterion. For each ν, we compute

$$r_n(\nu) = \exp\left(-\mathrm{j}2\pi\frac{f_c}{R_s}n\right)\mathfrak{R}\left(\tilde{r}, \left\{\check{t}_k(\nu)\right\}, \frac{n}{R_s}\right) \tag{2.26}$$

$$z_n(\nu) = \sum_l f_l^{(1)} r_{n-l}(\nu) \tag{2.27}$$

$$J^{\mathrm{CM}}(\nu) = \mu_4(\nu) \Big/ [\mu_2(\nu)]^2 \tag{2.28}$$

where $\mu_m(\nu)$ is (2.25) applied to the LE output $\{z_n(\nu)\}$, and $f^{(1)}$ is created according to Appendix 2C , using $\sigma_a^2 = 1$ and substituting CMP estimates for true parameter values. The LE length was chosen to be $F = 100$. We then compute $\hat{\nu} = \arg\min_\nu J^{\mathrm{CM}}(\nu)$. Resampling the data $\tilde{r}$ according to $\left\{\check{t}_k(\hat{\nu})\right\}$, then filtering with $f^{(1)}$, produces the sequence with smallest deviation from a CM, of the values considered for ν. In other words, the best value of ν will produce SR data $\{r_n(\nu)\}$ that is closest to LTI, which will produce LE output $\{z_n(\nu)\}$ that is closest to being MVDR and, based on the facts given above, is closest to being CM.

To estimate $\{\tau_k\}$, we follow a procedure motivated by Subsection 2.2.3. Using $\left\{\check{t}_k(\hat{\nu})\right\}$ to resample the data $\tilde{r}$, as opposed to using the true sampling times, is modeled as producing a rotation of the resulting SR data: $r_n(\hat{\nu}) \simeq w_n + \exp(\mathrm{j}\theta_n)\sum_l h_l a_{n-l}$. The phase θ_n is assumed slowly-varying with index n, so we treat it as constant during each LE computation. Let $z_n = z_n(\hat{\nu})$. We have the following model:

$$z_n \simeq v_n + g_0 \exp(\mathrm{j}\theta_n) a_n \tag{2.29}$$

where v_n is filtered AWGN plus residual ISI, and g_l is the convolution of the CIR h_l with the LE filter $f^{(1)}$. (In the absence of estimation error, $g_0 = 1$.) Consider the LS CF $\sum_n |z_n - g_0\exp(\mathrm{j}\theta_n)a_n|^2$ $= \sum_n |z_n|^2 + g_0^2\sum_n |a_n|^2 - 2g_0\sum_n \mathrm{Re}\left[z_n^* \exp(\mathrm{j}\theta_n) a_n\right]$. The term $\sum_n |z_n|^2$ is a constant and the term $g_0^2 \sum_n |a_n|^2$ depends only on g_0^2. We are only interested in $\{\theta_n\}$, so we define the new

CF $J^{\mathrm{VA}}(\theta,\ a) = \sum_n \mathrm{Re}\left[z_n^* \exp(\mathrm{j}\theta_n)\, a_n\right]$, where $\theta = \{\theta_n\}$. Consequently we estimate the phase sequence as

$$\hat{\theta} = \arg\max_{\theta \in \Theta} \max_{\{a_n\}_{n=N_t}^{N_s-1}} J^{\mathrm{VA}}(\theta,\ a) \tag{2.30}$$

$$= \arg\max_{\theta \in \Theta} \left\{ \sum_{n=0}^{N_t-1} \mathrm{Re}\left[z_n^* \exp(\mathrm{j}\theta_n)\, a_n\right] + \sum_{n=N_t}^{N_s-1} \max_{a_n \in \mathcal{A}} \mathrm{Re}\left[z_n^* \exp(\mathrm{j}\theta_n)\, a_n\right] \right\} \tag{2.31}$$

where Θ is a set of admissible phase sequences. The set Θ is restricted by the assumption of slowly-varying phase, and also by computational burden since there are $N_s = 10,000$ samples in each phase sequence. As illustrated in Fig. 2.7, we consider phase sequences that can be described by a trellis: the trellis "state" is the discretized phase at the beginning of each block of time, and the time intervals have length $N_\Delta = 1,000$ symbols. The phase starts at zero and either remains unchanged or increases or decreases by the amount $\theta_\Delta = 14°$ over each interval. This choice of admissible phase sequences permits solution via reverse dynamic programming [3]. This method of phase estimation is chosen over one based on use of a PLL because the latter is prone to "cycle slips" at low SNR.

As explained in Appendix 2A , we have $\hat{\theta}_n \simeq 2\pi f_c \epsilon(n/R_s)$, where $\epsilon(t)$ is sampling times estimation error as a function of time. Since we need the sampling times error at the estimated sampling times $\{\check{t}_k(\hat{\nu})\}$, we resample/interpolate to obtain

$$\hat{\tau}_k = \mathfrak{R}\left(\left\{\hat{\theta}_n/(2\pi f_c)\right\}, \{n/R_s\}, \check{t}_k(\hat{\nu})\right)$$

In this case we use linear interpolation. Finally, we compute the initial estimate of the sampling times $\hat{t}_k^{(1)} = \check{t}_k(\hat{\nu}) + \hat{\tau}_k$.

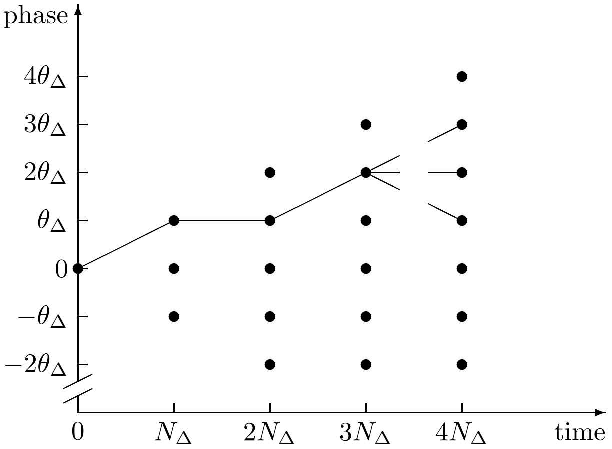


Figure 2.7: Illustration of initial phase estimation trellis.

2.3.4 Iterative Procedure

Since the ultimate objective is recovery of information bits, we consider an iteration cycle to end with the decoding task. A cycle thus consists of CMP estimation, followed by channel equalization, and is completed with decoding. The first iteration does not have the benefit of a priori symbol information other than the training sequence, so the initial CMP estimation is handled differently than during the iterative procedure. Consequently we enter the iterative procedure at the equalization step, and we find it natural to explain the procedure starting there. Initial CMP estimates $\hat{\mathcal{P}}^{(1)}$ are obtained as described previously. We let i denote the iteration, and set $i = 1$.

The goal of equalization is to compute the symbol APP estimates $\zeta_n^{(i)}$. We use (2.6) for the data pdf in (2.8), with SR data $r_n^{(i)} = \exp\left(-\mathrm{j}2\pi \left(f_c \left/ R_s\right.\right) n\right) \Re\left(\tilde{r}, \hat{\mathcal{T}}^{(i)}, n \left/ R_s\right.\right)$ obtained via the estimated sampling times $\hat{\mathcal{T}}^{(i)}$. In Appendix 2C we derive the expression $\zeta_n^{(i)}(\alpha) \propto \exp\left(-\rho^{(i)} \left|\alpha - z_n^{(i)}\right|^2\right)$, where $z_n^{(i)}$ is the output of a MVDR LE, and $\rho^{(i)}$ is its corresponding output signal to interference-plus-noise ratio (SINR). (The messages are normalized to sum to one, consistent with their interpretation as probabilities.) Input to the LE are the SR data $r_n^{(i)}$ and the estimated symbol means $\bar{a}_n^{(i)}$. The LE and its output SINR $\rho^{(i)}$ are computed from the CMP estimates $\hat{l}_1^{(i)}$, $\hat{l}_2^{(i)}$, $\hat{h}^{(i)}$, and $\hat{\sigma}_w^{(i)}$, and the symbol statistic $\left(\sigma_a^{(i)}\right)^2$. The LE output $z_n^{(i)}$ can be modeled as (2.29). Perfect CMP knowledge would give $\mathrm{e}^{\mathrm{j}\theta_n} = 1$, $g_0 = 1$, and $\sigma_v^2 = \mathrm{E}\left|v_n\right|^2 = 1 \left/ \rho^{(i)}\right.$. The CMP estimates are not perfect, however, so the SINR is not $\rho^{(i)}$ and the LE output $z_n^{(i)}$ is not unbiased. Since we have a large number of LE output samples ($N_s = 10,000$), we instead measure $\hat{g}_0$, and $\hat{\sigma}_v^2$ directly from $z_n^{(i)}$, as discussed in Appendix 2D , then compute the equalizer messages as

$$\zeta_n^{(i)}(\alpha) \quad \propto \quad \exp\left(-\left|z_n^{(i)} - \hat{g}_0\alpha\right|^2 \middle/ \hat{\sigma}_v^2\right) \tag{2.32}$$

Given equalizer messages $\zeta_n^{(i)}$, we next perform the decoding step. If $i = N_{\text{iter}} = 10$, then we compute information bit decisions and terminate the iterative procedure. (The value $N_{\text{iter}} = 10$ was obtained empirically.) Otherwise we compute symbol messages $\xi_n^{(i)}$, set $i \leftarrow i + 1$, and continue by estimating the CMPs.

From (2.7), (2.5) and (2.6), and utilizing approximations similar to those used in Appendix 2C , we obtain

$$\begin{aligned}
\hat{\mathcal{P}}^{(i)} &\simeq \arg\max_{\mathcal{P}} \sigma_w^{-2N_s^+} \exp\left(-\left\|r\left(\mathcal{T}\right)-\bar{A}h\right\|^2/\sigma_w^2\right) && (2.33)\\
h &= \left[h_{l_1},\ldots,h_{l_2}\right]^{\mathrm{T}} && (2.34)\\
\bar{A} &= \begin{bmatrix} \bar{a}_0^{(i)} & \cdots & \bar{a}_{l_1-l_2}^{(i)} \\ \vdots & & \vdots \\ \bar{a}_{l_2-l_1+N_s-1}^{(i)} & \cdots & \bar{a}_{N_s-1}^{(i)} \end{bmatrix} && (2.35)\\
r\left(\mathcal{T}\right) &= \left[r_{l_1}\left(\mathcal{T}\right),\ldots,r_{l_2+N_s-1}\left(\mathcal{T}\right)\right]^{\mathrm{T}} && (2.36)
\end{aligned}$$

where $N_s^+ = N_s + l_2 - l_1$. In the above formulation we also assume $\sigma_{\bar{a}}^2 \left\|h\right\|^2 \ll \sigma_w^2$, which becomes true as the iterative process converges successfully. Our approach to CMP estimation is cyclic [56]: we estimate the CIR using the previous sampling times estimate, then we update the estimated sampling times and noise variance using the new CIR estimate.

Synchronization and CIR length are unknown and must be updated during each iteration. Since we have a large number of symbols ($N_s = 10,000$), we first estimate the CIR over the span $\check{l}_1 = \hat{l}_2^{(i-1)} - L^{\max} + 1, \ldots, \check{l}_2 = \hat{l}_1^{(i-1)} + L^{\max} - 1$, which gives $\check{h} = \left[\check{h}_{\check{l}_1},\ldots,\check{h}_{\check{l}_2}\right]^{\mathrm{T}} = \bar{A}^+ \left[r_{\check{l}_1}^{(i-1)},\ldots,r_{\check{l}_2+N_s-1}^{(i-1)}\right]^{\mathrm{T}}$. The previous estimate had span $\hat{l}_1^{(i-1)},\ldots,\hat{l}_2^{(i-1)}$, which is contained in the new span. We "frame" the CIR estimate by establishing a threshold below which elements of $\check{h}$ are considered noise, and above which elements of $\check{h}$ are considered CIR. This is accomplished by sorting the elements of $\check{h}$ in order of increasing magnitude and calling the result $\acute{h} = \left[\acute{h}_0,\ldots,\acute{h}_{\check{l}_2-\check{l}_1}\right]^{\mathrm{T}}$, i.e. $\left|\acute{h}_l\right| \leq \left|\acute{h}_{l+1}\right|$. Let I be the mapping that takes $\check{h}$ into $\acute{h}$: $\acute{h}_l = \check{h}_{I_l}$. We model $\acute{h}$ as unbiased with AWGN with variance σ_η^2. For each index l we pose a binary hypothesis test: $\acute{h}_0,\ldots,\acute{h}_{l-1},\acute{h}_l$ are just noise, or $\acute{h}_0,\ldots,\acute{h}_{l-1}$ are noise and $\acute{h}_l$ is noise plus a CIR arrival. Given a probability of false alarm $p_{\mathrm{FA}}^{\mathrm{Frmg}}$ $(= 0.01)$, the threshold $\gamma_l = \left(p_{\mathrm{FA}}^{\mathrm{Frmg}}\right)^{-1/l} - 1$ is defined by $p_{\mathrm{FA}}^{\mathrm{Frmg}} = \mathrm{P}\left[\left|\acute{h}_l\right|^2 > \gamma_l \chi_l \middle| \acute{h}_l \overset{\mathrm{iid}}{\sim} p_w\left(\acute{h}_l \left|\sigma_\eta^2\right.\right)\right]$, where $\chi_l = \sum_{n=0}^{l-1}\left|\acute{h}_n\right|^2$. Let $\acute{\mathcal{L}} = \left\{l : \left|\acute{h}_l\right|^2 > \gamma_l \chi_l\right\}$ and $\mathcal{L} = \left\{I_l : l \in \acute{\mathcal{L}}\right\}$. The set $\mathcal{L}$ corresponds to the "significant" portion of the CIR estimate $\check{h}$. If $\max\mathcal{L} - \min\mathcal{L} < L^{\max}$ then we set $\hat{l}_1^{(i)} = \min\mathcal{L}$ and $\hat{l}_2^{(i)} = \max\mathcal{L}$. Otherwise we determine $\hat{l}_1^{(i)}$ and $\hat{l}_2^{(i)}$ such that $\hat{h}^{(i)}$ comprises the $L^{\max}$ contiguous elements of $\check{h}$ that are largest in magnitude. Finally we set $\hat{h}^{(i)} = \left[\check{h}_{\hat{l}_1^{(i)}},\ldots,\check{h}_{\hat{l}_2^{(i)}}\right]^{\mathrm{T}}$.

Define the communication signal estimate $\hat{x}_n^{(i)} = \sum_l \hat{h}_l^{(i)} \bar{a}_{n-l}^{(i)}$. If symbol information is not reliable enough, which we decide to be the case when $\left(\sigma_{\bar{a}}^{(i)}\right)^2$ is greater than the threshold $\left(\sigma_{\bar{a}}^{\max}\right)^2 = 0.25$, then we simply set $\hat{\mathcal{T}}^{(i)} = \hat{\mathcal{T}}^{(i-1)}$. (The value $\left(\sigma_{\bar{a}}^{\max}\right)^2 = 0.25$ was obtained empirically.) In this case the noise variance estimate is $\left(\hat{\sigma}_w^{(i)}\right)^2 = \sum_{n=\hat{l}_1^{(i)}}^{\hat{l}_2^{(i)}+N_s-1} \left|r_n^{(i-1)} - \hat{x}_n^{(i)}\right|^2 / \left(N_s - 1\right)$, which is an unbiased estimate when $\sigma_{\bar{a}}^{(i)} \to 0$. If $\left(\sigma_{\bar{a}}^{(i)}\right)^2 < \left(\sigma_{\bar{a}}^{\max}\right)^2$, then we are not afraid of doing more harm than good by updating our estimate of the sampling times. We do so by estimating a phase rotation of the SR data that, as explained in Appendix 2A , results from error in the previous estimate of the sampling times. Consequently we model the SR data as $r_n^{(i-1)} \simeq w_n + \exp\left(\mathrm{j}\theta_n\right)\hat{x}_n^{(i)}$. This gives the following vector model:

$$\begin{aligned}
r^{(i-1)} &= \left[r^{(i-1)}_{\hat{l}^{(i)}_1}, \ldots, r^{(i-1)}_{\hat{l}^{(i)}_2+N_s-1}\right]^{\mathrm{T}} && (2.37)\\
&= w + D\left(\hat{x}^{(i)}\right)\exp\left(\mathrm{j}\theta\right) && (2.38)\\
\hat{x}^{(i)} &= \left[\hat{x}^{(i)}_{\hat{l}^{(i)}_1}, \ldots, \hat{x}^{(i)}_{\hat{l}^{(i)}_2+N_s-1}\right]^{\mathrm{T}} && (2.39)\\
\theta &= \left[\theta_{\hat{l}^{(i)}_1}, \ldots, \theta_{\hat{l}^{(i)}_2+N_s-1}\right]^{\mathrm{T}} && (2.40)
\end{aligned}$$

where w is noise, $D\left(\cdot\right)$ is a diagonal matrix constructed from its argument, and $\exp\left[\mathrm{j}\left(\cdot\right)\right]$ is a vector whose elements are the indicated operation applied to the elements of its argument. We use cubic splines to describe the variation in the phase vector θ; this gives $\theta = C\boldsymbol{\theta}$, where $\boldsymbol{\theta} = \left[\theta_0, \theta_{N_\Delta}, \ldots, \theta_{N_s}\right]^{\mathrm{T}}$, and C is a matrix such that θ is a cubic spline interpolation of $\boldsymbol{\theta}$. Incorporating this into (2.38) motivates the LS CF $J^{\mathrm{CS}}\left(\boldsymbol{\theta}\right) = \left\|r^{(i-1)} - D\left(\hat{x}^{(i)}\right)\exp\left(\mathrm{j}C\boldsymbol{\theta}\right)\right\|^2$. We use Newton's method to minimize this CF with respect to $\boldsymbol{\theta}$. Since we anticipate phase variations about zero, that is where we evaluate the gradient and Hessian: $\nabla J\left(\mathbf{0}\right) = C^{\mathrm{T}}\mathrm{Im}\left[q\right]$ and $\nabla^2 J\left(\mathbf{0}\right) = C^{\mathrm{T}} D\left(\mathrm{Re}\left[q\right]\right)C$, where $q = D\left(\hat{x}^{(i)}\right)^{\mathrm{H}} r^{(i-1)}$. We then estimate the phase rotation, noise variance, and sampling times as follows:

$$\begin{aligned}
\check{\boldsymbol{\theta}} &= \left[\nabla^2 J\left(\mathbf{0}\right)\right]^{-1}\nabla J\left(\mathbf{0}\right) && (2.41)\\
\hat{\boldsymbol{\theta}} &= \check{\boldsymbol{\theta}} - \check{\boldsymbol{\theta}}_0 && (2.42)\\
\hat{\theta} &= C\hat{\boldsymbol{\theta}} && (2.43)\\
\left(\hat{\sigma}^{(i)}_w\right)^2 &= \frac{1}{N_s}\left\|r^{(i-1)} - D\left(\hat{x}^{(i)}\right)\exp\left(\mathrm{j}\hat{\theta}\right)\right\|^2 && (2.44)\\
\hat{t}^{(i)}_k &= \hat{t}^{(i-1)}_k + \Re\left(\left\{\frac{\hat{\theta}_n}{2\pi f_c}\right\}, \left\{\frac{n}{R_s}\right\}, \hat{t}^{(i-1)}_k\right) && (2.45)
\end{aligned}$$

The resampling in (2.45) is performed using linear interpolation–i.e. linear combination of two successive points–since the phase sequence is assumed to vary slowly.

Having completed the CMP estimation step of the iterative procedure, a new LE filter and its output SINR are computed for use in the next equalization.

2.3.5 Computational Complexity

We briefly state the computational complexity of the main components of the algorithm. Following common practice we approximate only the number of multiplications and higher-order function evaluations.

- Coarse Synchronization (Subsection 2.3.1)

  Each SR data sample requires two real multiplications (to compute squared magnitude of complex value) and one real division. (The unweighted moving-average filtering only requires additions.)

- Framing from Training (Subsection 2.3.2)

  We assume the QR decomposition $A\left(L^{\max}\right) = UT$, where U is unitary and T is upper triangular with real entries on its main diagonal. These factors are stored at the receiver along with $\left(1 + \mathrm{tr}\left\{\left[A\left(L\right)^{+} A\left(L\right)^{+\mathrm{H}}\right]^{-1}\right\}\right)/(N_t - L)$ for $L = 1, \ldots, L^{\max}$. Let N_{n_0} denote the number of synchronization points at which the framing CF is evaluated. It requires $N_{n_0}\left(2N_t + 3L^{\max}\right)$ real and $N_{n_0}L^{\max}N_t$ complex multiplications to compute the framing CF (2.15), and $L^{\max}\left(L^{\max} - 1\right)/\,2$ complex multiplications and $L^{\max}$ real divisions to compute the CIR estimate (2.18).

- Estimate of Doppler Linear Trend (Subsection 2.3.3)

  Let N_ν denote the number of values of ν at which the CF (2.28) is evaluated. Computing the N_ν resampled sequences requires $4N_\nu N_s$ complex and $N_\nu N_s$ real multiplications, and $N_\nu N_s$ real divisions. A single LE filter is constructed using at most $\mathcal{O}\left(\left(L^{\max}\right)^3\right)$ complex multiplications/divisions. It then takes about $N_\nu F N_s \cong N_\nu L^{\max} N_s$ complex multiplications to perform the filtering and about $3N_\nu N_s$ real multiplications to compute the CF.

- Initial Phase Estimation (Subsection 2.3.3)

  We give the number of computations needed for each “branch metric” evaluation (BME), then the number of BMEs needed to estimate the phase sequence. Each BME requires N_Δ evaluations of $\exp\left(\mathrm{j}\theta_n\right)$ (computing θ_n can be done with only additions) and $\left(1 + |\mathcal{A}|\right) N_\Delta$ complex multiplications, where $|\mathcal{A}| = 4$ for the QPSK alphabet. Fig. 2.7 shows the number of BMEs being equal to $3\left(N_s/\,N_\Delta\right)^2$, since two new states are created after each block. However, the figure only shows the beginning of the trellis. At the end of the trellis the number of states decreases to nine total states, since $9 \cdot 14° = 126°$ is the effective gap between the resampled sequences in the estimation of Doppler linear trend. Consequently the number of BMEs is proportional to $N_s/\,N_\Delta$, not its square, and the actual number of BMEs is $246 \simeq 25\; N_s/\,N_\Delta$. Therefore, $25N_s$ evaluations of $\exp\left(\mathrm{j}\theta_n\right)$ and $125N_s$ complex multiplications are needed to estimate the phase sequence.

- Iterative Procedure (Subsection 2.3.4)

  - Equalization

    It takes at most $\mathcal{O}\left(\left(L^{\max}\right)^3\right)$ complex multiplications/divisions to construct the LE, and about $2FN_s$ complex multiplications to filter the data. Computing symbol APPs then requires relatively few operations, although $|\mathcal{A}|\, N_s$ exponentials must be evaluated.

- De-mapping, Decoding, Mapping

  These are standard operations whose computational complexity can be found in the literature [9, 19].

- CMP Estimation

  A LS CIR estimate is computed over a wide span, then framed. The LS estimate $\check{h} = \bar{A}^{+} r$ is computed as $\check{h} = \left(\bar{A}^{\mathrm{H}}\bar{A}\right)^{-1}\left(\bar{A}^{\mathrm{H}} r\right)$, where the terms in parentheses are computed as correlations and require about $2L^{\max}N_s$ complex multiplications. The matrix inversion requires at most $\mathcal{O}\left((L^{\max})^3\right)$ complex multiplications/divisions. The framing procedure requires comparatively few operations.

- Phase Estimation via Cubic Splines

  Forming the statistic q (used to compute the gradient and Hessian) takes N_s complex multiplications. Forming the gradient and Hessian takes $11N_s$ and $\left[11 + (11)^2\right] N_s$ real multiplications. Computation of the vector of cubic spline control points requires $\mathcal{O}\left((11)^3\right)$ real multiplications/divisions. The remaining operations require relatively few computations.

Table 2.1: Number of Bit Errors for Sensors with SNR Below 8 dB.

	Sensor	21	22	26	29	30	31	32
	Depth (m)	64	62	54	48	46	44	42
	SNR (dB)	5.2	8.0	7.4	6.1	7.1	7.0	5.8
Iteration	1	1315	710	618	2033	1217	448	2679
	2	509	264	21	1110	281	12	2092
	3	159			494	6		1346
	4	32			155			739
	5	12			16			267
	6	5			10			81
	7	5						1
	8	5						
	9	5						
	10	5						

(Blank entries indicate zero errors)

2.4 Results and Discussion

Overall system performance is the most important result. Table 2.1 shows the number of bit errors, out of $N_b = 10,000$, that would result if processing were terminated after each iteration. Only sensors with SNR at or below 8 dB are included, since the data from all other sensors causes the algorithm to converge with zero bit errors after two iterations. Sensor 21, with lowest SNR of 5.2 dB, has data that converges with 5 bit errors. The other sensors have data that converges to zero bit errors. We conclude the system operates at the limit of its reliable performance for the sensors with SNR below about 6 dB.

Fig. 2.8 illustrates the point of the iterative algorithm. It shows LE output scatter plots for the first three iterations using data from sensor 24. Successive improvement of the equalization step of the processing is demonstrated via an increase in LE output SINR from 2.1 dB to 5.6 dB to 8.0 dB. Sensor 24, at 10 dB input SNR, was chosen because the scatter plots obtained from processing its data clearly illustrate the point. However, the same successful operation is achieved using data from sensors with lower SNR, even though the resulting scatter plots would show less noticeable improvement as the iterations increase.

We introduce metrics with which to assess performance of the iterative algorithm. They are functions of the iteration i, and are defined as follows:

$$M_i^{(h)} = \frac{\sum_l \left|\hat{h}_l^{(i)} - \hat{h}_l^{(N_{\text{iter}})}\right|^2}{\sum_l \left|\hat{h}_l^{(N_{\text{iter}})}\right|^2} \tag{2.46}$$

$$M_i^{(\sigma_w^2)} = \left(\hat{\sigma}_w^{(i)}\right)^2 \Big/ \left(\hat{\sigma}_w^{(N_{\text{iter}})}\right)^2 \tag{2.47}$$

$$M_i^{(\mathcal{T})} = 2\pi f_c \max_k \left|\hat{t}_k^{(i)} - \hat{t}_k^{(N_{\text{iter}})}\right| \tag{2.48}$$

$$M_i^{(z)} = \frac{\left|\frac{1}{N_s}\sum_{n=0}^{N_s-1} a_n^* z_n^{(i)}\right|^2}{\frac{1}{N_s}\sum_{n=0}^{N_s-1}\left|z_n^{(i)}\right|^2 - \left|\frac{1}{N_s}\sum_{n=0}^{N_s-1} a_n^* z_n^{(i)}\right|^2} \tag{2.49}$$

The first metric measures error in the CIR estimates. The second metric measures error in the AWGN variance estimates. The third metric measures error in the sampling times estimates, but expresses the result in terms of equivalent phase error (see Appendix 2A). The fourth metric is an estimate of the LE output SINR. The transmitted symbols are used in its computation. The numerator is the squared norm of the LS projection of the LE output sequence onto the symbol sequence; the denominator is the squared norm of the projection onto the orthogonal subspace. We also consider as a metric the number of bit errors that would result if processing were terminated after each iteration. This is the same information contained in Table 2.1. The

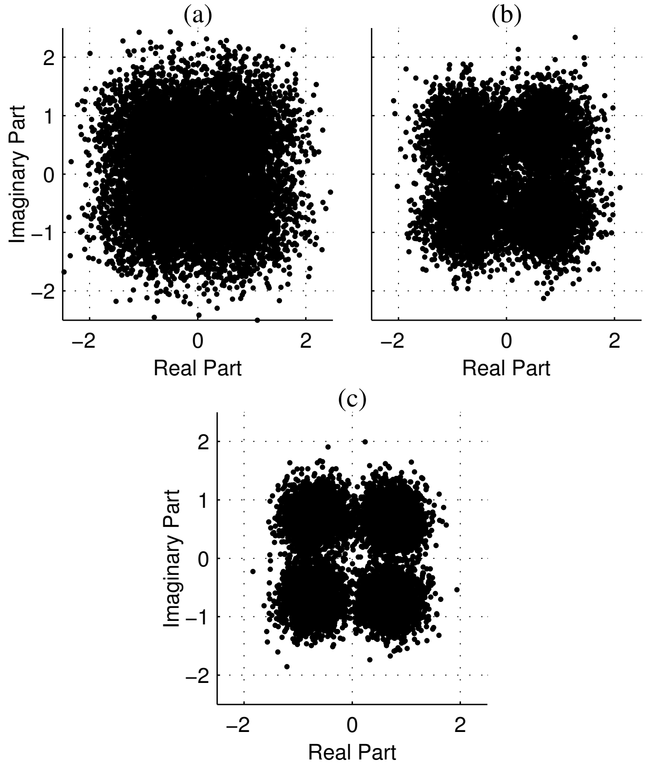


Figure 2.8: Scatter plots of LE output for sensor 24, illustrating iterative improvement in the equalization process. (a) First iteration: LE output SINR is 2.1 dB. (b) Second iteration: LE output SINR is 5.6 dB. (c) Third iteration: LE output SINR is 8.0 dB.

final metric we use is $\left(\sigma_{\bar{a}}^{(i)}\right)^2$. It is a measure of the average uncertainty in the (data-bearing) symbols at the start of each cycle. It is unity for the first iteration, reflecting an initial absence of symbol information. Ideally $\left(\sigma_{\bar{a}}^{(i)}\right)^2$ tends to zero as i increases to N_{iter}.

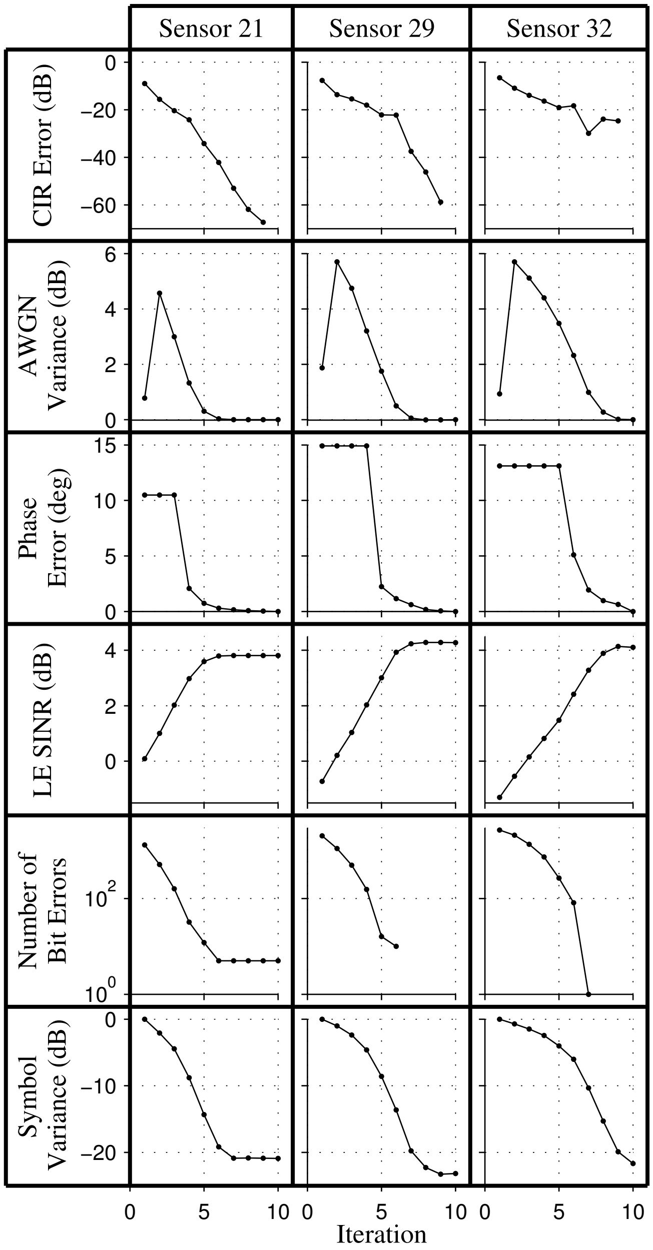
Table 2.2 shows plots of the six metrics computed using data from sensors 21, 29, and 32–the three sensors with lowest SNR. The majority of plots demonstrate successful convergence of the iterative algorithm. Even for sensor 21, the only sensor whose data converges with nonzero bit errors, every metric is well behaved as the iterations increase. The CIR estimates for sensor 32 show much more error than for the other two sensors, but the error is apparently not large enough to prevent successful recovery of the information bits. The jump in estimated AWGN variance during the second iteration is anticipated: the algorithm cannot prevent error from estimating the other parameters, and from symbol uncertainty, from distorting the AWGN variance estimate. The sampling times metrics show an initial maximum phase deviation of about 15°. This is the maximum, however, and the average phase error is likely less than 5° for the first few iterations. As the algorithm converges, the measurable phase error is negligible. It takes 3-5 iterations before $\left(\sigma_{\bar{a}}^{(i)}\right)^2$ is low enough for the sampling times estimates to be updated. Once the first updates occur, there is a large decrease in phase error, which indicates the threshold $\left(\sigma_{\bar{a}}^{\max}\right)^2 = 0.25$ is possibly too low. Finally, we observe the average symbol variance $\left(\sigma_{\bar{a}}^{(i)}\right)^2$ drops below 0.01. Therefore, equalization essentially becomes matched-filtering followed by interference cancellation using hard symbol decisions. (This is apparent in the development of Appendix 2C .) In other words, the algorithm effectively becomes a two-sided DFE whose forward filter is a matched-filter.

2.5 Conclusion

In this paper we introduced an iterative detection algorithm and applied it to UWA communication data. The data was modeled as the nonuniformly sampled output of a LTI system. Channel equalization was achieved by first interpolating/resampling the received data, then filtering the Doppler-compensated data with a LE to obtain symbol information. The symbol information was then decoded to produce improved symbol information, which was then used to improve channel equalization, and so forth. Therefore, the process had three main tasks: estimation of CMPs to enable channel equalization, equalization to produce symbol APP estimates, and decoding to produce symbol prior probability estimates. These tasks were performed cyclicly until the process converged. On the last iteration, information bit decisions were computed. The algorithm produced few or zero bit errors for the data considered.

Iterative algorithms, in particular MP or “belief-propagation” algorithms, are a growing trend in the digital communication/storage and signal processing fields. While the original algo-

Table 2.2: Performance Metrics for Sensors with Lowest SNR.



Note: There are zero bit errors for Sensor 29 after iteration six and for Sensor 32 after iteration seven.

rithms were applied to the decoding of channel codes, they have been extended to all manner of estimation and detection problems. We have applied these concepts to the problem of recovering digital information transmitted over an UWA channel. The initial "Turbo" equalization methods assumed CMPs to be known at a receiver. Iterative equalization and decoding of data sent over a channel with unknown parameters is a relatively new subject. However, the great success of established MP algorithms, along with the constant advancement in computational capability, motivates the application of iterative algorithms to the UWA communication problem.

Iterative detection is an active field of research and, in particular, we again draw attention to the work of Tuchler and his colleagues. Many aspects of our algorithm are similar to techniques employed by these and other researchers. However, we have attempted to present a comprehensive solution to our specific problem. In particular, synchronization is a critical task that is often not addressed, and we have presented an explicit procedure for performing it. Also, while it is common to model phase variation using an auto-regressive random process and employ causal PLLs for phase estimation, we have instead modeled phase variation using cubic splines with unknown control points and used noncausal estimation of the unknown parameters. Finally, given that our symbol sequence is long, we showed it is better to directly measure the gain and noise-plus-ISI variance of the equalizer output, rather than rely on indirect theoretical values. While more challenging channels exist, we have nonetheless endeavored to present a complete answer to our fixed-source/fixed-receiver problem.

2.6 Acknowledgments

This work was supported by the Office of Naval Research under Grant N00014-05-1-0263.

Chapter 2, in full, is a reprint of the material as it appears in "An Iterative Equalization and Decoding Approach for Underwater Acoustic Communication", J. F. Sifferlen, H. C. Song, W. S. Hodgkiss, W. A. Kuperman, and J. M. Stevenson, *IEEE Journal of Oceanic Engineering*, Vol. 33, No. 2, April 2008, pp. 182-197. The dissertation author was the primary investigator and author of this paper.

2A Resampling/Interpolation Model and Assumptions

There is several times over-sampling ($F_s\,/R_s\,=12$), so we assume the communication signal $\tilde{s}\,(t)$ can be reconstructed with negligible distortion via interpolation of its samples $\tilde{s}\,(t_k)$

as follows:

$$\tilde{s}(t) \cong \sum_{l=-L^{\mathrm{int}}}^{L^{\mathrm{int}}-1} \tilde{s}(t_{k-l})\, h^{\mathrm{int}}_{q(t)+lQ^{\mathrm{int}}} \tag{2.50}$$

where $t_k \leq t < t_{k+1}$ for some k, $q(t) = \lfloor Q^{\mathrm{int}}(t-t_k)/(t_{k+1}-t_k) \rceil$, Q^{int} is an upsampling factor, and h^{int}_q is an interpolating pulse that is nonzero only for $-L^{\mathrm{int}}Q^{\mathrm{int}} < q < L^{\mathrm{int}}Q^{\mathrm{int}}$. (The notation $\lfloor \cdot \rceil$ denotes rounding the argument to the nearest integer.) This interpolation expression is essentially a "polyphase" implementation [57]: it makes a linear approximation to the sampling times $(t_{k-L^{\mathrm{int}}+1}, \ldots, t_{k+L^{\mathrm{int}}})$–using $1/(t_{k+1}-t_k)$ as instantaneous sampling frequency–upsamples by the factor Q^{int}, then uses nearest-neighbor interpolation at the higher sampling rate. The pulse spectrum is ideally flat and zero-phase over the frequency band with edges $(f_c \pm R_s/2)/\left(Q^{\mathrm{int}}F_s\right)$ and has the Nyquist property $h^{\mathrm{int}}_{lQ^{\mathrm{int}}} = \begin{cases} 1, & l=0 \\ 0, & l \neq 0 \end{cases}$. We used a Kaiser-windowed sinc pulse shifted to the ideal center frequency $f_c/\left(Q^{\mathrm{int}}F_s\right)$, with $Q^{\mathrm{int}} = 100$ and $L^{\mathrm{int}} = 2$ (i.e. 4 samples were used to compute each interpolated value).

We examine the effects of resampling/interpolating the data $\tilde{r}$ using estimated sampling times $\{\hat{t}_k\}$, which is an important part of the iterative algorithm. Denote by $\hat{\tilde{s}}(t)$ the approximation of $\tilde{s}(t)$ made using (2.50) with $\{\hat{t}_k\}$ replacing the true sampling times. We assume the sampling time estimates are accurate enough that $\hat{t}_{k+1} - \hat{t}_k \cong t_{k+1} - t_k$ for some desired time t and index k with $\hat{t}_k \leq t < \hat{t}_{k+1}$. Consequently, $\hat{q}(t) = \lfloor Q^{\mathrm{int}}\left(t-\hat{t}_k\right)/\left(\hat{t}_{k+1}-\hat{t}_k\right) \rceil \cong \lfloor Q^{\mathrm{int}}\left(t+t_k-\hat{t}_k-t_k\right)/(t_{k+1}-t_k) \rceil = q\left(t+t_k-\hat{t}_k\right)$, so that

$$\hat{\tilde{s}}(t) = \sum_{l=-L^{\mathrm{int}}}^{L^{\mathrm{int}}-1} \tilde{s}(t_{k-l})\, h^{\mathrm{int}}_{\hat{q}(t)+lQ^{\mathrm{int}}} \tag{2.51}$$

$$\cong \sum_{l=-L^{\mathrm{int}}}^{L^{\mathrm{int}}-1} \tilde{s}(t_{k-l})\, h^{\mathrm{int}}_{q\left(t+t_k-\hat{t}_k\right)+lQ^{\mathrm{int}}} \tag{2.52}$$

$$\cong \tilde{s}\left(t+t_k-\hat{t}_k\right) \tag{2.53}$$

Fig. 2.9 illustrates this scenario for $t = n/R_s$: the value $\hat{\tilde{s}}(n/R_s)$ is approximately equal to

$$\tilde{s}\left(\frac{n}{R_s}+\epsilon\left(\frac{n}{R_s}\right)\right) = \exp\left(\mathrm{j}2\pi\frac{f_c}{R_s}n\right)\sum_l a_l\left[\exp\left(\mathrm{j}2\pi f_c\epsilon\left(\frac{n}{R_s}\right)\right)h\left(\epsilon\left(\frac{n}{R_s}\right)+\frac{n-l}{R_s}\right)\right]$$

where $\epsilon(t)$ is error in sampling time estimation as a function of time. Consequently there are two effects caused by error in estimating the sampling times: rotation of the CIR by the angle $\theta(t) = 2\pi f_c \epsilon(t)$ and delay of the CIR by the amount $\epsilon(t)$. Noting that a delay of ϵ symbols corresponds to a rotation of $(f_c/R_s)\,\epsilon$ cycles, and assuming the estimated sampling times are accurate enough that the rotation is well under a full cycle, the delay term can be ignored and the effect of sampling time estimation error simply treated as a rotation of the CIR. For instance, since $f_c/R_s = 3.5$ cycles/symbol, a rotation of $1/3$ cycles corresponds to a delay of less than $1/10$ symbols.

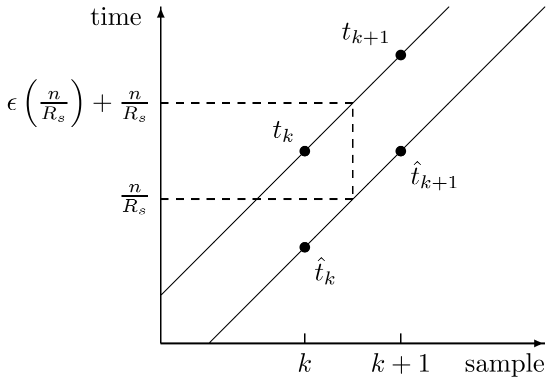


Figure 2.9: Illustration of sampling time estimation error.

2B Training "Framing" Cost Function

We use the model and notation of Subsection 2.3.2, but we drop the superscript on the SR data. The symbol convolution matrix $A(L)$ is full (column) rank, so the pseudo-inverse $A(L)^{+} = \left[A(L)^{\mathrm{H}} A(L)\right]^{-1} A(L)^{\mathrm{H}}$. Given n and L, the noise vector estimate is $\hat{w}(n,L) = r(n) - A(L)\hat{h}(n,L) = \left[\mathrm{I} - A(L)A(L)^{+}\right] w(n)$, where the last line assumes the CIR estimate is unbiased. Consequently, $\mathrm{E}\left\|\hat{w}(n,L)\right\|^2 = \sigma_w^2(N_t - L)$ and an unbiased estimate of the noise variance is

$$\hat{\sigma}_w^2 \quad = \quad \frac{1}{N_t - L}\left\|\left[\mathrm{I} - A(L)A(L)^{+}\right] r(n)\right\|^2 \tag{2.54}$$

The SR data model (2.5) can be written as $r_n = w_n + \sum_l \left(h_l - \hat{h}_l\right) a_{n-l} + \sum_l \hat{h}_l a_{n-l}$. The middle term on the right hand side represents the error incurred by using the CIR training estimate instead of the true CIR. It has zero mean and variance $\sigma_w^2 \mathrm{tr}\left\{\left[A(L)^{+} A(L)^{+H}\right]^{-1}\right\}$, where we again assume the CIR estimate is unbiased. This CIR estimation error term is independent of the ambient noise, the total effective noise variance is $\sigma_w^2\left(1 + \mathrm{tr}\left\{\left[A(L)^{+} A(L)^{+H}\right]^{-1}\right\}\right)$. Inserting the ambient noise variance estimate (2.54) into the last expression gives the result (2.15).

2C LE Derivation

We use the SR data model (2.5), but without explicitly showing the functional dependence of the SR data on the sampling times. We first make a series of definitions that are needed to derive the MVDR LE:

$$\begin{align}
\mathbf{r}_n(\delta) &= \left[r_{n+\delta-F+1},\ldots,r_{n+\delta}\right]^{\mathrm{T}} \tag{2.55}\\
&= \mathbf{w}_n + H\mathbf{a}_n(\delta) \tag{2.56}\\
H &= \left[H_{l_2+F-1-\delta}(\delta),\ldots,H_{l_1-\delta}(\delta)\right] \tag{2.57}\\
H_l(\delta) &= \left[h_{l+\delta-F+1},\ldots,h_{l+\delta}\right]^{\mathrm{T}} \tag{2.58}\\
\mathbf{a}_n(\delta) &= \left[a_{n+\delta-F+1-l_2},\ldots,a_{n+\delta-l_1}\right]^{\mathrm{T}} \tag{2.59}\\
H\mathbf{a}_n(\delta) &= \underline{H}_0(\delta)\,\underline{\mathbf{a}}_n(\delta) + H_0(\delta)\,a_n \tag{2.60}\\
\underline{H}_0(\delta) &= \left[\ldots,H_1(\delta),H_{-1}(\delta),\ldots\right] \tag{2.61}\\
\underline{\mathbf{a}}_n(\delta) &= \left[\ldots,a_{n-1},a_{n+1},\ldots\right]^{\mathrm{T}} \tag{2.62}\\
\mathcal{I}(\delta) &= \{l_1-\delta,\ldots,l_2+F-1-\delta\}-\{0\} \tag{2.63}
\end{align}$$

The parameters F and δ are the length and delay of the LE filter. The vector $\mathbf{w}_n$ is ambient noise and has covariance $C_w = \sigma_w^2\mathrm{I}$. The matrix $\underline{H}_0(\delta)$ is the matrix H with the column vector $H_0(\delta)$ removed. Note that H does not depend upon the delay δ, while $H_0(\delta)$ and $\underline{H}_0(\delta)$ do. Similarly, the vector $\underline{\mathbf{a}}_n(\delta)$ is the vector $\mathbf{a}_n(\delta)$ with the element a_n removed. As is done throughout the paper, overbar will be used to indicate expectation with respect to the symbol prior probability estimates ξ_n.

Fig. 2.2(b) shows the flow of messages in and out of the equalizer. Starting from (2.8) we obtain

$$\begin{align}
\zeta_n(\alpha) &\mathrel{\tilde{\propto}} \int\limits_{\{a_l\}_{l\neq n}} p\left(\mathcal{D}\middle|\, a_n=\alpha,\{a_l\}_{l\neq n},\mathcal{P}\right)\prod_{l\neq n} p_w\left(a_l-\bar{a}_l\,\middle|\sigma_{\bar{a}}^2\right)\mathrm{d}a_l \tag{2.64}\\
&\cong \int\limits_{\underline{\mathbf{a}}_n(\delta)} p\left(\mathbf{r}_n(\delta)\middle|\, a_n=\alpha,\underline{\mathbf{a}}_n(\delta),\mathcal{P}\right)\prod_{l\in\mathcal{I}(\delta)} p_w\left(a_{n-l}-\bar{a}_{n-l}\,\middle|\sigma_{\bar{a}}^2\right)\mathrm{d}\underline{\mathbf{a}}_n(\delta) \tag{2.65}\\
&\propto \int\limits_{\underline{\mathbf{a}}_n(\delta)} \exp\Big(-\left\|\mathbf{r}_n(\delta)-H_0(\delta)\,\alpha-\underline{H}_0(\delta)\,\underline{\mathbf{a}}_n(\delta)\right\|_{C_w^{-1}}^2 \notag\\
&\quad -\left\|\underline{\mathbf{a}}_n(\delta)-\bar{\underline{\mathbf{a}}}_n(\delta)\right\|^2/\sigma_{\bar{a}}^2\Big)\,\mathrm{d}\underline{\mathbf{a}}_n(\delta) \tag{2.66}\\
&\propto \exp\left(-\rho(\delta)\left|\alpha-z_n(\delta)\right|^2\right) \tag{2.67}\\
\rho(\delta) &= \frac{H_0(\delta)^{\mathrm{H}}Q^{-1}H_0(\delta)}{1-\sigma_{\bar{a}}^2H_0(\delta)^{\mathrm{H}}Q^{-1}H_0(\delta)} \tag{2.68}\\
Q &= C_w+\sigma_{\bar{a}}^2HH^{\mathrm{H}} \tag{2.69}\\
z_n(\delta) &= \bar{a}_n+\mathbf{f}(\delta)\left[\mathbf{r}_n(\delta)-H\bar{\mathbf{a}}_n(\delta)\right] \tag{2.70}\\
\mathbf{f}(\delta) &= \frac{H_0(\delta)^{\mathrm{H}}Q^{-1}}{H_0(\delta)^{\mathrm{H}}Q^{-1}H_0(\delta)} \tag{2.71}
\end{align}$$

The notation $\|\cdot\|_{C_w^{-1}}^2$ denotes the quadratic form $(\cdot)^{\mathrm{H}}C_w^{-1}(\cdot)$. The first approximation above converts an intractable sum into an integral that can be expressed in closed-form. The second approximation simply reduces the integral over the entire symbol sequence to an integral over symbols within a window about the target symbol a_n. The output SINR $\rho(\delta)$ is a function of filter decision delay, so we choose δ to maximize this term: $\delta^{\text{opt}} = \arg\max_\delta \rho(\delta)$. The resulting filter produces the MVDR linear estimate of the error $a_n - \bar{a}_n$, for the given filter length F. Consequently we have

$$\begin{aligned} [f_{F-1-\delta}, \ldots, f_{-\delta}] &= \mathbf{f}\left(\delta^{\text{opt}}\right) & (2.72) \\ z_n &= \bar{a}_n + \sum_l f_l \left(r_{n-l} - \bar{x}_{n-l}\right) & (2.73) \\ \bar{x}_n &= \sum_l h_l \bar{a}_{n-l} & (2.74) \end{aligned}$$

where z_n is the filter output and $\bar{x}_n$ is the “known” portion of the communication signal. As $\sigma_{\bar{a}}^2 \to 0$, we obtain a matched filter; the filter length F need not be longer that the CIR if the ambient noise is white.

2D Estimation of LE Output Statistics

We model LE output using (2.29), and set $\theta_n = 0$. The noise+ISI v_n is assumed Gaussian with variance σ_v^2. It is straightforward to show the second and fourth moments of z_n, defined by (2.25), behave as follows when $N_s \to \infty$:

$$\begin{aligned} \mu_2 &\to \sigma_v^2 + g_0^2 & (2.75) \\ 2\mu_2^2 &\to 2\sigma_v^4 + 4\sigma_v^2 g_0^2 + 2g_0^4 & (2.76) \\ \mu_4 &\to 2\sigma_v^4 + 4\sigma_v^2 g_0^2 + g_0^4 & (2.77) \end{aligned}$$

Consequently, for large N_s, $2\mu_2^2 - \mu_4 \cong g_0^4$. Therefore we estimate the LE output gain and noise+ISI variance as $\hat{g}_0 = \left(2\mu_2^2 - \mu_4\right)^{1/4}$ and $\hat{\sigma}_v^2 = \mu_2 - \hat{g}_0^2$.

Ideally, the LE output is unbiased, i.e. $g_0 = 1$, and σ_v^2 is known. CMP knowledge is imperfect, however, so this is not the case. We ran simulations to compare the accuracy of the theoretical values derived in Appendix 2C with values estimated using the expressions given above. We generated data using the sensor 21 CIR estimate of Fig. 2.3(d) and assumed perfect knowledge of the ambient noise variance σ_w^2. We also set the prior symbol variance $\sigma_{\bar{a}}^2 = 1$. Denote the true CIR by $\mathring{h}$. We avoided issues of synchronization and CIR length estimation by truncating the CIR length to 55 symbol intervals (from index -14 through 40 in the figure) and taking this to be known. We modeled the CIR estimate as $\hat{h}_l = \eta_l + \mathring{h}_l$, where η_l is AWGN with variance $\sigma_\eta^2 = \sigma_w^2 / N_{\text{trng}}$. This represents an idealized training estimate. For each trial we generated data for $N_s = 10,000$ symbols and a CIR estimate. We then computed an MVDR LE filter as described in Appendix 2C , using the CIR estimate $\hat{h}$ instead of the true value. Let $*$ denote convolution. We have the following:

$$
\begin{aligned}
r &= w + \mathring{h} * a && (2.78)\\
z &= f * r && (2.79)\\
&= f * w + \mathring{g} * a && (2.80)\\
\mathring{g} &= f * \mathring{h} && (2.81)\\
z_n &= v_n + \mathring{g}_0 a_n && (2.82)\\
v_n &= \sum_l f_l w_{n-l} + \sum_{l \neq 0} \mathring{g}_l a_{n-l} && (2.83)
\end{aligned}
$$

The actual LE gain is $\mathring{g}_0$ and the actual noise+ISI variance is $\mathring{\sigma}_v^2 = \|f\|^2 \sigma_w^2 + \sum_{l \neq 0} |\mathring{g}_l|^2$. Since these quantities vary with each trial, we are interested in the ratios $1 \left/ \mathring{g}_0 \right.$, $\hat{g}_0 \left/ \mathring{g}_0 \right.$, $\rho \left(\delta^{\mathrm{opt}}\right)^{-1} \left/ \mathring{\sigma}_v^2 \right.$, and $\hat{\sigma}_v^2 \left/ \mathring{\sigma}_v^2 \right.$, where $\hat{g}_0$ and $\hat{\sigma}_v^2$ are computed as described in the preceding paragraph. We show the results from 1 million trials in Fig. 2.10. We actually show the magnitude of the gain ratios, since the phase variation is negligible. It is clear the theoretical values are biased and have larger error variance than the estimated values. Therefore, computing equalizer messages using (2.67) with $\delta = \delta^{\mathrm{opt}}$, instead of using (2.32), would result in less reliable information being sent to the decoder.

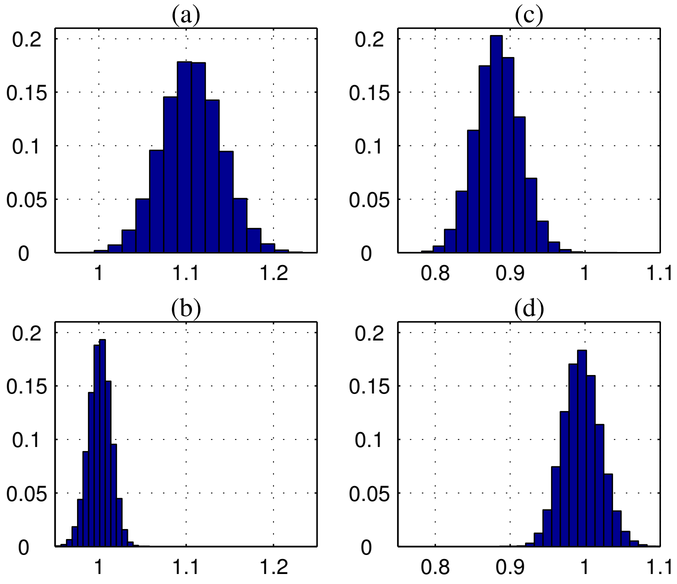


Figure 2.10: Simulation results. (a) Ratio of theoretical gain to actual gain. (b) Ratio of estimated gain to actual gain. (c) Ratio of theoretical noise variance to actual noise variance. (d) Ratio of estimated noise variance to actual noise variance.

3

Iterative Equalization and Decoding of Communication Data Received from a Moving Underwater Acoustic Source

We present a novel algorithm that recovers information transmitted from a moving source through a shallow underwater acoustic communication channel. The algorithm utilizes "turbo equalization". The three main tasks–estimating channel model parameters, equalization, and decoding–are performed iteratively until the process converges. The communication sequence is organized into a packet structure with training symbols inserted at the front and pilot symbols interleaved with data-bearing symbols. However, coding is performed in a manner such that information bits are mapped into an equal number of QPSK symbols, including training and pilot. Since we use a bandwidth equal to twice the symbol rate, our spectral efficiency is 0.5 bits/sec/Hz. Our algorithm uses data from a single receive sensor. Doppler effects caused primarily by source motion are compensated by adaptive (nonuniform) resampling. Extended Kalman filtering is used to estimate the time-varying channel impulse response. Data obtained during an at-sea experiment is combined with suitably scaled ambient noise sequences, which were also collected during the experiment. We present results of processing over 1,600 sequences at an SNR of 7 dB which show high reliability and low bit error frequency.

3.1 Introduction

A growing number of industrial, military, and marine research applications require reliable high-rate systems to receive data from remote sensors and autonomous undersea vehicles (AUVs). These data sources usually have strict power and processing limitations, while receivers such as surface ships typically do not. Consequently powerful signal processing techniques such as "turbo equalization" [15, 16] are ideally suited for underwater acoustic (UWA) communication [27–30], yet they have only just begun to receive attention in the UWA research community [37–42].

The shallow UWA channel has three dominant features: long channel impulse responses (CIRs), acute Doppler sensitivity, and limited bandwidth [1]. The communication system presented in this paper addresses these channel characteristics as follows. Coding is an integral part of any reliable communication system but it introduces redundancy, so we use QPSK symbols which compensate for this redundancy without requiring additional bandwidth. Doppler effects are particularly significant in the data used in this paper because the transmission source was moving, these effects are compensated by nonuniform resampling of the data. The dispersive nature of the shallow UWA channel necessitates equalization, and the severity of the spreading of the CIR makes equalization a critical task which we accomplish using iterative equalization and decoding (IED). Iterative algorithms that utilize the sum-product algorithm (SPA) [6] are an emerging paradigm in digital communication [7, 58]. The premise is cooperative and iterative exchange of information between algorithm components.

We employ standard coding methods and focus our attention on the equalization aspect of IED. The case of short CIRs is well developed [17, 43], the corresponding equalization methods typically employ hidden Markov models (HMMs) and consequently have complexity exponential in the CIR length measured in symbol periods. Since the CIRs we consider in this paper are prohibitively long for such methods, we adopt an approach pioneered by Tuchler et al [22, 23]. The basic idea is to approximate symbol probability mass functions (pmf's), which are computed over the finite symbol alphabet, with suitable Gaussian probability density functions (pdf's). This allows intractable summations that result from application of the SPA to be replaced with integrals that can be evaluated in closed-form. The end result is equalization performed with linear filters and, to within a multiplicative factor, the filters are equivalent to those obtained using the minimum mean squared-error (MMSE) criterion [3]. The approximation is used during CIR estimation as well, which is also developed by Tuchler and his colleagues [20, 21, 45]. The approximation leads to an interesting interpretation: instead of using symbol "hard" estimates obtained by mapping the "soft" equalizer output into the symbol alphabet, we use symbol "mean" estimates (SMEs) obtained by averaging the symbols with the estimated pmf's computed as messages in the SPA. In particular, we have prior SMEs (PSMEs) and a posteriori SMEs (ASMEs).

In this paper we present an IED algorithm with several novel aspects. First, we introduce a comprehensive strategy for synchronization and CIR arrival selection (or "sparsing"). Second, since a receiver algorithm's performance is ultimately determined by its ability to strike a balance between having enough degrees of freedom to accurately describe the data but not so many as to suffer excessively noisy channel estimates, we exploit a method of CIR estimation which achieves such a balance. Third, we show the drawback of using channel model parameter (CMP) estimates to compute subordinate quantities such as the gain and effective noise variance at the output of the equalizer, and propose an original method of estimating these quantities directly. Finally, although resampling to compensate for Doppler effects has already been addressed in the literature [59–61], our approach is particularly effective.

The remainder of this paper is organized as follows. In Section II we describe the Focused Acoustic Fields 2004 experiment, from which we obtained our data, and state our signal and channel models and assumptions. The receiver algorithm is then explained in Section III. In Section IV we present results from processing data collected during the experiment. We conclude the paper in Section V.

3.2 Experiment Setup, Models and Assumptions

3.2.1 Experiment Setup

The experiment was conducted during July 2004 in a shallow water region north of Elba Island, Italy. The experiment geometry is summarized in Fig. 3.1. The NRV Alliance towed an acoustic source–simulating an AUV–at approximately 70 m depth and transmitted communication signals through an ocean region 120 m deep to a vertical receive array (VRA). The VRA consisted of 32 sensors with 2 m spacing between elements, element #1 was at depth 104 m and #32 was at depth 42 m. Transmissions were just over 9 seconds in duration and made at 20 sec intervals. Two sets of data are considered in this paper. The first set, denoted "A", corresponds to data received while the source was moving directly away from the VRA at a speed of about 4 knots and a distance of about 7 km. There are 24 separate transmissions associated with this data set. The second set, denoted "B", corresponds to data received while the source was moving in a direction nearly perpendicular to the direction from source to VRA at a speed of about 3.5 knots and a distance of about 11 km. There are 27 separate transmissions associated with this data set. The experiment region was downward-refracting, the measured sound speed as a function of depth can be found in [62].

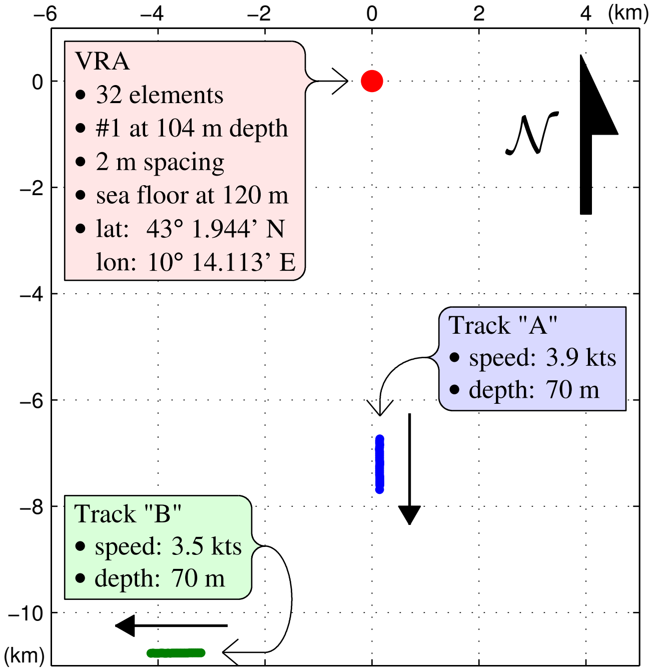


Figure 3.1: Experiment geometry.

3.2.2 Encoding, Permuting, and Mapping

The process of transforming information bits into a symbol sequence is summarized in Fig. 3.2. Information bits b are interleaved with dummy bits–i.e. bits containing no information and which are known at the receiver–and then encoded using a rate-$\frac{2}{3}$ convolutional code (CC) with a maximum free distance of 7 and a constraint length of 4 [53]. The encoded bits are S-randomly permuted [52] and mapped (pairwise) into $N_d = 7836$ QPSK symbols. The symbols are zero-mean, unit-variance and, by virtue of the pseudo-random (PR) permutation, uncorrelated. Inserting $N_t = 300$ training symbols at the beginning of the sequence and interleaving $N_p = 870$ pilot symbols throughout the remainder of the sequence produces the transmitted symbol sequence $a = (a_0, \ldots, a_{N_s-1})$, where $N_s = N_t + N_p + N_d = 9,006$. We use the same number of information bits as symbols. The pilot symbols stabilize the tracking of time-varying CMPs. As explained in [62], though not shown in Fig. 3.2(a), a "scrambling" operation is inserted between the permutation and mapping which has no effect from a statistical standpoint and is easily incorporated in a MPA.

3.2.3 Signal Models and Assumptions

We model the received signal as a noisy superposition of distorted replicas of the transmitted signal, each of which travels along a separate propagation path from source to receiver. We

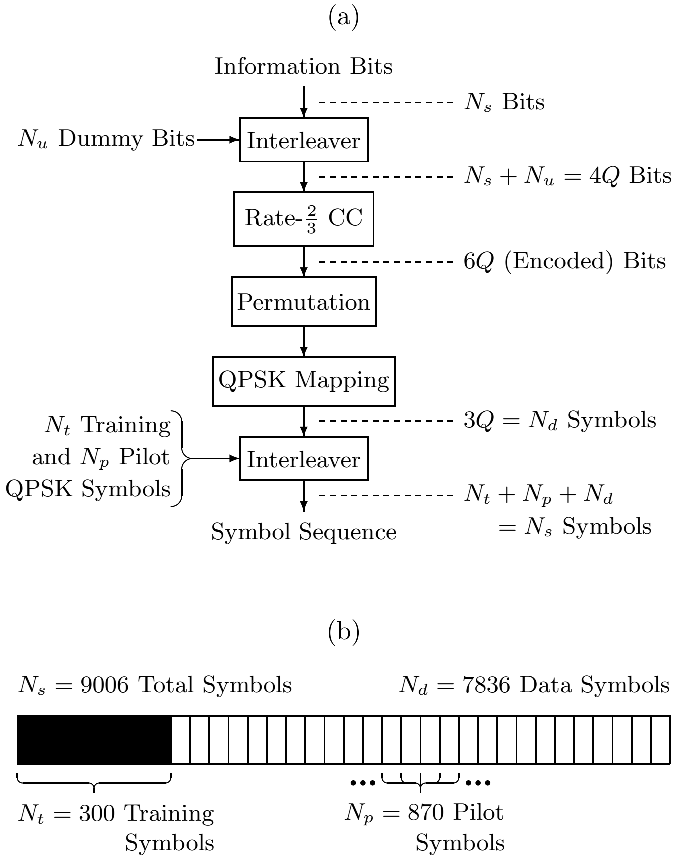


Figure 3.2: (a) Coding Scheme. (b) Symbol Sequence Organization.

use linear modulation and QPSK symbols so the transmitted signal is a train of phase-shifted pulses. Each ray-path distorts and delays the transmitted pulses independently. Source motion causes the delays to vary with time since the ray-paths change as source position changes. Moreover, source velocity effectively stretches/compresses the signal traveling along each ray-path because of Doppler effects. The amount of stretching/compression depends upon source (and receiver) speed and direction and channel properties and varies with time. For example, a source moving away from a receiver will produce a received signal which appears to be stretched compared to the transmitted signal. We assume proper resampling/interpolation produces data accurately modeled by a slowly time-varying CIR. We make working definitions of Doppler "shift" and "spread". Our algorithm can handle any Doppler shift, but a key assumption is that the Doppler spread is small.

The signal $2\mathrm{Re}\left[\tilde{s}^{(0)}\left(t\right)\right]$ is transmitted from a single transducer, with

$$\tilde{s}^{(p)}\left(t\right) \quad = \quad \exp\left(\mathrm{j}2\pi f_c t\right)\sum_{l} a_l h^{(p)}\left(t - l\,/R_s\,\right) \tag{3.1}$$

The transmit pulse $h^{(0)}\left(t\right)$ has bandwidth 2 kHz, the symbol rate $R_s = 1$ symbol/msec, and the carrier frequency $f_c = 3$ kHz. The received signal minus noise is $2\mathrm{Re}\left[\sum_{p=1}^{P}\tilde{s}^{(p)}\left(\acute{t}^{(p)}\left(t\right)\right)\right]$, where P is the number of propagation paths and $\acute{t}^{(p)}\left(t\right)$ accounts for stretching/compression caused by Doppler effects. For example, constant source and receiver velocities in freespace would result in a single path with $\acute{t}\left(t\right) = \acute{t}\left(0\right) + \left(1+\nu\right)t$, where ν is a function of the source and receiver velocities. More generally we define the Doppler "shift" $\nu^{(p)}\left(t\right) = \frac{d}{dt}\acute{t}^{(p)}\left(t\right) - 1$ and Doppler "spread" $\chi\left(t\right) = \max_{p_1,p_2}\left|\nu^{(p_1)}\left(t\right) - \nu^{(p_2)}\left(t\right)\right|$. Note these two quantities are dimensionless and if the transmitted signal were narrowband then the Doppler frequency shift would be $f_c\nu^{(p)}\left(t\right)$ and the associated frequency spread among the ray-paths would be $f_c\chi\left(t\right)$. We make no restrictions on any particular Doppler shift $\nu^{(p)}\left(t\right)$, but we assume $\chi\left(t\right) < 1\,/N_s\,$. In other words, as long as the Doppler effects on the separate ray-paths are not too different then resampling can correct for an arbitrary average Doppler shift and yield a slowly time-varying CIR. The pulses $h^{(p)}\left(t\right)$, $p = 1,\ldots,P$, are time-varying but this is not explicitly denoted (since the single variable argument represents delay). Changes to the signal frequency band resulting from Doppler effects are considered negligible.

The received signal is sampled by a single hydrophone at the rate $F_s = 12$ samples/msec, then filtered to give *complex* data $\tilde{r}_k$ in the frequency band 2-4 kHz. Anticipating the need to resample/interpolate the data to compensate for Doppler effects, we model the received data sequence as $\tilde{r}_k = \tilde{w}_k + \tilde{s}\left(\acute{t}\left(k\,/F_s\,\right)\right)$, where $\tilde{w}_k$ is noise, $\tilde{s}$ is the communication signal and $\acute{t}\left(t\right)$ accounts for an average Doppler shift. In a previous paper [62] we introduced a resampling/interpolation technique which we also use in this paper. The process is assumed information lossless. We showed that if a signal $x(t)$ is sampled at times t_k to produce the sequence $x_k = x(t_k)$, and this sequence is used in conjunction with estimated sampling times $\hat{t}_k$ to compute interpolated values $\hat{x}(t)$, then $\hat{x}(t) \cong x\left(t + t_k - \hat{t}_k\right)$. Likewise, associating $\hat{t}_k$ with $\acute{t}(k/F_s)$ and t_k with $\acute{t}^{(p)}(k/F_s)$ and defining $\tau_k^{(p)} = \acute{t}^{(p)}(k/(2R_s)) - \acute{t}(k/(2R_s))$, we approximate the resampled/interpolated communication signal as

$$\begin{aligned}
\tilde{s}\left(\frac{k}{2R_s}\right) &\cong \sum_{p=1}^{P} \tilde{s}^{(p)}\left(\frac{k}{2R_s} + \tau_k^{(p)}\right) && (3.2)\\
&= \exp\left(\mathrm{j}2\pi f_c \frac{k}{2R_s}\right) \sum_l a_l \sum_{p=1}^{P} \exp\left(\mathrm{j}2\pi f_c \tau_k^{(p)}\right) h^{(p)}\left(\tau_k^{(p)} + \frac{k-2l}{2R_s}\right) && (3.3)
\end{aligned}$$

The interpolated samples are computed at twice the symbol rate because that is the communication signal bandwidth. The mismatch $\tau_k^{(p)}$ between the actual Doppler distortion of the pth path and the average estimated distortion manifests itself by phase-rotating the pulse $h^{(p)}(t)$ by $f_c\tau_k^{(p)}$ cycles and delaying it by $R_s\tau_k^{(p)}$ symbols. Assuming the mismatch is small, i.e., assuming the sampling times $\acute{t}\left(\frac{k}{F_s}\right)$ are estimated properly, the assumption of small Doppler spread makes the former effect most significant. Consequently it is desirable to express the CIR as $\exp(\mathrm{j}\theta_{\delta,k})\, h(\delta/(2R_s))$, where δ indexes delay (and k indexes time) and again we suppress the inherent time-variation of h. The critical point is that h varies with time at a much slower rate than the phase rotations $\theta_{\delta,k}$. It is convenient to express the time index as $k = m + 2n$, $m \in \{0, 1\}$, which gives

$$\exp\left(-\mathrm{j}2\pi f_c \frac{m+2n}{2R_s}\right) \tilde{s}\left(\frac{m+2n}{2R_s}\right) = \sum_l a_{n-l} \exp\left(\mathrm{j}\theta_{m+2l,m+2n}\right) h\left(\frac{m+2l}{2R_s}\right) \qquad (3.4)$$

where a change of variable in the summation index of the convolution sum has been made.

Fig. 3.3 shows a typical CIR. It was measured using the true symbols and the SNR is above 30 dB. The received data samples, originally sampled at 12 kHz, were resampled/interpolated at approximately 2 kHz in such a manner that the phase of the strongest CIR arrival (at delay 0 msec) is approximately zero during the nine second transmission. The CIR is about 80 symbols in length and was measured using rectangular-windowed least-squares (RWLS) with a window length of 450 symbols. Part (a) shows the magnitude in dB of the time-varying CIR plotted as gray-scale intensity with delay on the vertical axis and time on the horizontal axis. The assumption of small Doppler spread is clearly met and proper resampling results in an effective CIR whose arrivals vary little in delay and magnitude. Part (b) shows magnitude versus time for five particular values of delay: those corresponding to the three strongest peaks as well as the first and last significant peaks–they are indicated with dots in part (a). The arrival at 6.5 msec, marked with triangles, is the only one to show considerable change in magnitude during the transmission. Part (c) shows phase in degrees for the same five CIR peaks. The

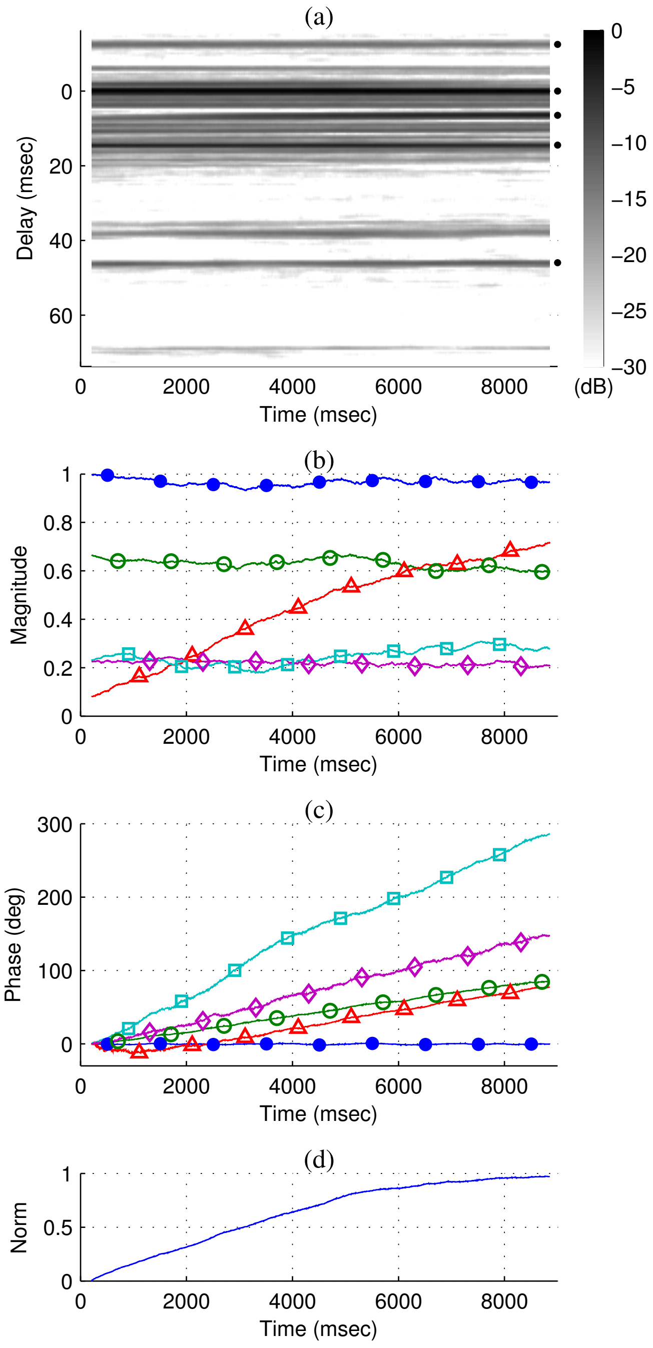


Figure 3.3: (a) Magnitude of time-varying CIR: vertical axis is delay, horizontal axis is time, gray-scale intensity is magnitude in dB. (b) Magnitude versus time for the five CIR arrivals indicated with dots in part (a). (c) Phase versus time for same five CIR arrivals. The five CIR arrivals are marked as follows: delay 0 msec with solid dots, delay 14.5 msec with circles, delay 6.5 msec with triangles, delay 46 msec with squares, delay -12.5 msec with diamonds. (d) Energy of difference between CIR estimates and initial CIR estimate, normalized by maximum CIR energy.

phases vary independently and significantly and are a manifestation of Doppler spread. Part (d) shows the energy of the difference between the time-varying CIR estimates and the earliest CIR estimate, normalized by the energy of the largest CIR estimate. After one second the CIR changes by an amount with energy equal to about 20% of the maximum CIR energy, and the CIR is completely different at the end of the transmission than at the beginning. So, while most CIR arrivals exhibit little variation in magnitude, the phase variation is sufficient to cause significant change in the CIR. The data used to generate this figure was obtained from the sensor at 94 m depth (sensor 6) at the very beginning of data set A. The measured Doppler shift did not vary significantly from an average value of -0.001543 over the duration of the data sequence. Note that resampling is entirely necessary in this case since otherwise the CIR would shift in delay by about $(-0.001543)\,(80\,\text{symbols}) = -0.123\,\text{symbols}$ and rotate in phase by about $(-0.123\,\text{symbols})\,(f_c\,/R_s\,) = -133\,\text{degrees}$ during each interval of time equal to its span, making it impossible to estimate the CIR using RWLS.

The discussion above motivates the following model:

$$r_n^{(m)} = w_n^{(m)} + \exp\left(-\mathrm{j}2\pi f_c \frac{m+2n}{2R_s}\right) \tilde{s}\left(\frac{m+2n}{2R_s}\right) \tag{3.5}$$

$$= w_n^{(m)} + \exp\left(\mathrm{j}\phi_n\right) \sum_l a_{n-l} \rho_{l,n}^{(m)} \exp\left(\mathrm{j}\theta_{l,n}^{(m)}\right) \tag{3.6}$$

$$\mathrm{E}\left|w_n^{(m)}\right|^2 = \left(\sigma_w^{(m)}\right)^2 \tag{3.7}$$

The data samples $r_n^{(m)}$ are obtained by resampling/interpolating the original data samples $\tilde{r}_k$ using estimated sampling times $\acute{t}\,(k\,/F_s\,)$. The two "subchannels", i.e., $m \in \{0,1\}$, result from resampling at twice the symbol rate. They are assumed independent because they result from resampling at a rate equal to the signal bandwidth. The noise samples $w_n^{(m)}$ are modeled as white and Gaussian. Both subchannels have the same variance σ_w^2, but we model them as possibly being different. The CIR gains $h_{l,n}^{(m)} = \rho_{l,n}^{(m)} \exp\left(\mathrm{j}\theta_{l,n}^{(m)}\right)$ are modeled in polar form to allow for different rates of change in the phases than in the amplitudes. Note that, for two nonzero vectors x and y, $\|y - x\exp(\mathrm{j}\phi)\|^2 = \|y\|^2 + \|x\|^2 - 2\mathrm{Re}\left[\left(x^{\mathrm{H}}y\right)^* \exp(\mathrm{j}\phi)\right] \geq \|y\|^2 + \|x\|^2 - 2\left|x^{\mathrm{H}}y\right|$ with equality when $\exp(\mathrm{j}\phi) = x^{\mathrm{H}}y\,/\left|x^{\mathrm{H}}y\right|$, which implies that the least-squares (LS) estimate of phase rotation between two vectors is zero if the imaginary part of their inner product is zero. We constrain the inner product between successive CIR estimates (expressed as vectors) to have zero imaginary part. This forces error in the sampling times, i.e., mismatch between the true and estimated Doppler shift, to appear in the phase ϕ_n. We estimate the phase ϕ_n at the equalizer output and use it as an error signal to control estimation of the sampling times. We also assume the span of the CIR does not exceed $L^{\max} = 100$ symbols.

3.2.4 Equalizer Output Model

The receiver algorithm uses linear filtering to perform equalization. The filters are constructed from estimates of the CIR and noise variance as described in Appendix 3B . The "equalization filter" $f_{l,n}^{(m)}$ operates on the resampled/interpolated data and the "cancellation filter" $g_{l,n}$ operates on the PSMEs $\bar{a}_n$. We assume the phase rotation ϕ_n varies slowly and we use the following equalizer output model

$$\begin{aligned} z_n &= \sum_{m=0}^{1} \sum_{l} f_{l,n}^{(m)} r_{n-l}^{(m)} && (3.8) \\ &= v_n + \exp\left(\mathrm{j}\phi_n\right) \gamma_n a_n + q_n && (3.9) \\ \omega_n &= 2 \Big/ \mathrm{E}\left|v_n\right|^2 && (3.10) \\ q_n &= \sum_{l \neq 0} g_{l,n} \bar{a}_{n-l} && (3.11) \end{aligned}$$

The (symbol-rate) sequence z_n is the output of the equalization filter and q_n is the output of the cancellation filter. The noise v_n is not white–it is the combination of filtered ambient noise, residual ISI, and additional noise caused by estimation error–but is modeled as such for tractability. It is convenient to define the parameter ω_n as the reciprocal of half this noise variance. In the absence of channel estimation error, ω_n and the gain γ_n would be known exactly. The gain γ_n is complex-valued, but its angle is smaller than can be accurately estimated so we model it as real-valued. Therefore the equalizer output model parameters (EOMPs) are the real variables $\omega_n, \gamma_n, \phi_n$.

3.2.5 SMEs and Associated Statistics

Approximating symbol pmf's with Gaussian pdf's requires SMEs and symbol variances. The SMEs are computed as

$$\bar{a}_n = \sum_{\alpha \in \mathcal{A}} \alpha \hat{p}_n\left(\alpha\right) \qquad (3.12)$$

where $\hat{p}_n\left(\alpha\right)$ are either prior probability or a posteriori probability (APP) estimates, and $\mathcal{A}$ is the QPSK alphabet. Consistent with (3.12) we could compute symbol variances as

$$\sum_{\alpha \in \mathcal{A}} \left|\alpha - \bar{a}_n\right|^2 \hat{p}_n\left(\alpha\right) = 1 - \left|\bar{a}_n\right|^2 \qquad (3.13)$$

We use a statistical interpretation instead. For PSMEs we argue that the PR permutation in the decoding process effectively gives them uniform variance equal to $\sigma_{\bar{a}}^2 = 1 - \mathcal{E}_{\bar{a}}$, where $\mathcal{E}_{\bar{a}} = \mathrm{E}\left|\bar{a}_n\right|^2 \cong \left(1 / N_d\right) \sum_{n \in \mathcal{I}_d} \left|\bar{a}_n\right|^2$, where $\mathcal{I}_d$ is the set of indices of data-bearing symbols. For ASMEs, on the other hand, there is no expectation that their variance is constant. We could filter the ASME energy $|\bar{a}_n|^2$ in order to estimate $\mathcal{E}_{\bar{a}}$, but we use a different approach. We note the signal to interference-plus-noise ratio (SINR) at the equalizer output is approximately $(1/2)\,\hat{\omega}\hat{\gamma}^2$, and we empirically compute the functional relationship between the SINR and $\mathcal{E}_{\bar{a}}$ (assuming AWGN) shown in Fig. 3.4. Consequently, during the first iteration of the algorithm when PSMEs are not available, the ASME variance is obtained directly from the EOMPs.

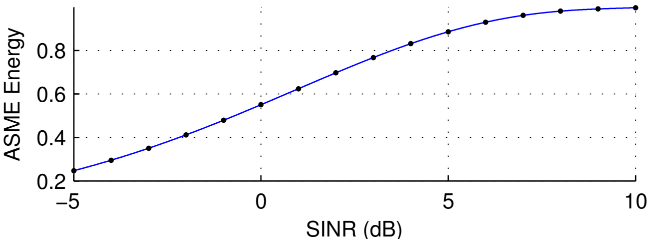


Figure 3.4: Functional relationship between SINR and ASME energy $\mathcal{E}_{\bar{a}}$.

We introduce one final statistic: $\chi_{\bar{a}} = \mathrm{E}\left[a_n^* \bar{a}_n\right]$. In principle it should be the case that $\chi_{\bar{a}} = \mathcal{E}_{\bar{a}}$. This condition would ensure unbiased CIR estimates when using SMEs. (Note that this condition cannot hold when using symbol "hard" estimates unless they are without error.) Puncturing a rate-$\frac{1}{2}$ CC–instead of using dummy bits with a rate-$\frac{2}{3}$ CC, as is done in the algorithm presented in this paper–resulted in the relationship $\chi_{\bar{a}} = (1-\lambda)\,\mathcal{E}_{\bar{a}}$, where $0.5 < \lambda < 1$ is the fraction of punctured encoded bits. For this reason we use the latter method vice the former.

3.2.6 Input SNR

Data collected during the experiment that contains communication signals is at very high SNR, so we ignore the ambient noise present in this data. Also collected during the experiment were recordings of ambient noise without any signals present. The data used in this paper was obtained by scaling the ambient noise sequences and adding them to the communication sequences to yield an SNR, as defined below, of 7 dB.

We define the *input* SNR as the SNR at the *output* of a matched-filter:

$$\begin{aligned}\mathrm{SNR}_{\mathrm{in}} &= \mathrm{SNR}_{\mathrm{MF}} \\ &= \sum_m \left\|h^{(m)}\right\|^2 / \sigma_w^2 \\ &= \sum_m \mathrm{SNR}_{\mathrm{MF}}^{(m)}\end{aligned}$$

where $\mathrm{SNR}_{\mathrm{MF}}^{(m)} = \left\|h^{(m)}\right\|^2 \Big/ \left(\sigma_w^{(m)}\right)^2$ and $\left\|h^{(m)}\right\|^2 = \sum_l \left|h_{l,n}^{(m)}\right|^2$. Since the subchannel energy $\left\|h^{(m)}\right\|^2$ technically varies with time, as does the noise variance, the definition of input SNR is used in a time-averaged sense.

Recall that each hydrophone sampled the acoustic field at approximately the rate $F_s =$ 12 samples/msec, then filtered the resulting sequence to produce a *complex* sequence in the frequency band 2-4 kHz. Let $\tilde{s}_k$ denote a communication sequence obtained in this manner, and $\tilde{w}_k$ denote an ambient noise sequence. Also, let $\langle x_k \rangle_{(k)}$ denote the time-average of the sequence x_k (with respect to the index k). Based on our signal and channel models and assumptions we have

$$\begin{aligned} \left\langle |\tilde{s}_k|^2 \right\rangle_{(k)} &\cong \frac{1}{2} \sum_{m=0}^{1} \left\langle \sum_{l} \left| h_{l,n}^{(m)} \right|^2 \right\rangle_{(n)} \\ \left\langle |\tilde{w}_k|^2 \right\rangle_{(k)} &\cong \sigma_w^2 \end{aligned}$$

where the summand is the average energy in the mth subchannel. Consequently, to generate the data used in this paper, we scale the noise sequences so that $2 \left\langle |\tilde{s}_k|^2 \right\rangle_{(k)} \Big/ \left\langle |\tilde{w}_k|^2 \right\rangle_{(k)}$ are the desired input SNR, that is, 7 dB.

3.3 Receiver Algorithm

3.3.1 Overview

The receiver algorithm utilizes IED and the focus of this paper is on equalization. We use the term "decoding" to refer to all the operations identified in Fig. 3.2(a) which transform a sequence of information bits into a symbol sequence. As shown in the figure this entails more than just decoding of the constituent CC. We interpret the messages passed between equalizer and decoder as estimates of symbol pmf's. More specifically, the messages output from the equalizer are estimates of symbol APPs–i.e. pmf's conditioned on the received data–whereas the messages output from the decoder serve as estimates of symbol prior probabilities. In this section we explain how we compute the former. Computation of the latter, and of all other messages involved with decoding, are straightforward applications of the SPA and are explained thoroughly elsewhere [9, 19] and are not discussed in this paper.

Ideally we would use only messages passed from the decoder to estimate CMPs and then perform equalization. However, this is not possible during the first iteration since no information is available from the decoder. Consequently, during the first iteration, the time-varying nature of the channel necessitates utilizing feedback of symbol APPs produced by the equalizer–i.e. messages that are passed from the equalizer to the decoder–for the purpose of propagating CMP estimates. In other words, the first iteration is necessarily performed differently from subsequent iterations.

Block diagrams of the receiver algorithm are shown in Fig. 3.5. Asynchronous transmis-

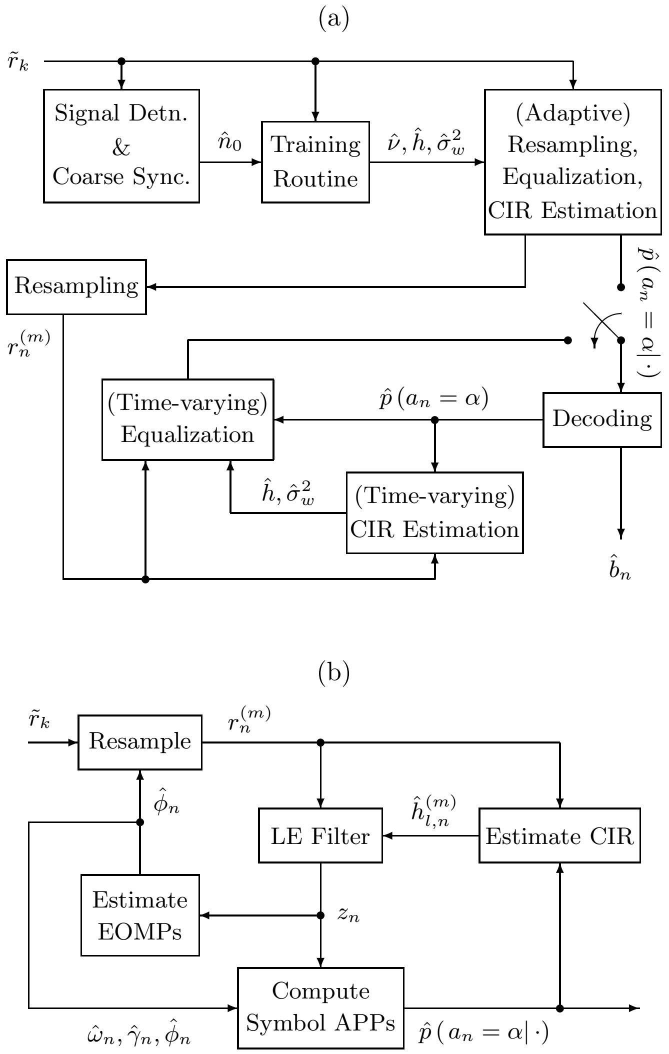

(a)
$\tilde{r}_k$
Signal Detn. & Coarse Sync.
$\hat{n}_0$
Training Routine
$\hat{\nu}, \hat{h}, \hat{\sigma}_w^2$
(Adaptive) Resampling, Equalization, CIR Estimation
Resampling
$\hat{p}\,(a_n = \alpha|\,\cdot\,)$
$r_n^{(m)}$
(Time-varying) Equalization
$\hat{p}\,(a_n = \alpha)$
Decoding
$\hat{h}, \hat{\sigma}_w^2$
(Time-varying) CIR Estimation
$\hat{b}_n$
(b)
$\tilde{r}_k$
Resample
$r_n^{(m)}$
$\hat{\phi}_n$
LE Filter
$\hat{h}_{l,n}^{(m)}$
Estimate CIR
Estimate EOMPs
z_n
Compute Symbol APPs
$\hat{\omega}_n, \hat{\gamma}_n, \hat{\phi}_n$
$\hat{p}\,(\,a_n = \alpha|\,\cdot\,)$

Figure 3.5: (a) Algorithm block diagram. (b) Details of block labeled "(Adaptive) Resampling, Equalization, CIR Estimation" in part (a).

sions are assumed, so the first step is to detect when a communication signal is present (vice only ambient noise). This process also establishes coarse synchronization–denoted by $\hat{n}_0$ in the figure. The training symbols are then used to fine-tune synchronization and obtain initial estimates of the other key CMPs–namely the Doppler shift, the CIR, and the AWGN variance. A resampling/interpolation reference index is also chosen. These CMP estimates are used to initialize the following recursion, which is illustrated in Fig. 3.5(b). The current estimate of Doppler shift controls the resampling process that produces new Doppler-corrected data at twice the symbol rate. This new data enters the equalization filter which is constructed based on the current CIR and AWGN variance estimates. The filter output is first used to update EOMP estimates, and then used in conjunction with these parameter estimates to compute a new symbol APP and ASME. The phase estimate is also used to control the Doppler shift estimate. Feedback of ASMEs is used to propagate the CIR estimate. Once APPs are obtained for all the symbols they are passed to the Decoder. Additionally, the estimated sampling times are smoothed and a final resampling is performed for use in later iterations. The sampling times could be iteratively improved after each iteration but, in practice, the refinement would be minimal and is not worthwhile.

The decoder can produce two types of output: estimates of the information bits and estimates of the symbol prior probabilities. On the last iteration of the algorithm the former is desired. Otherwise the latter is produced and used as follows. The same tasks performed in the first iteration are repeated in subsequent iterations–except for estimating Doppler shift and resampling–but they are performed sequentially instead of simultaneously. The symbol prior probability estimates are used to compute PSMEs and the PSME variance, which are then used to estimate the time-varying CIR. The channel estimate and PSME variance are used to create equalizer filters which operate on the data and PSMEs. The EOMPs are estimated using the equalizer output and the symbol prior probability estimates. Finally, new symbol APPs are computed from the EOMPs and equalizer output. There are two ways to terminate the algorithm: processing can be stopped once a prescribed number of iterations are completed, or the PSME variance can be monitored and the processing stopped when a steady-state condition is reached. The latter method can also be used to determine if the algorithm has failed and retransmission is necessary–i.e. if the PSME variance does not approach zero then the algorithm has failed.

We use the same method to detect the presence of a communication signal as is used in [62]. The method there was described for the case of symbol-rate sampling, but extends in a straightforward manner to the present case of two samples per symbol period. Consequently we do not discuss it further in this paper, other than to reiterate the procedure also establishes coarse synchronization which is needed for the training routine.

3.3.2 Training Routine

In [62] a training routine is developed which simultaneously refines coarse synchronization and produces symbol-rate estimates of the CIR and noise variance. The procedure is built around a cost function (CF)–an effective noise variance–which strikes a balance between estimating the CIR at too many delays, and thus projecting too much noise onto the signal subspace, and estimating the CIR at too few delays, which would not only be wasting signal energy but allowing it to act as interference–effectively additional noise. In this subsection we modify the approach to accomodate Doppler-distorted data that is resampled at twice the symbol rate.

We need the following definitions:

$$\begin{aligned}
r^{(m)}(\nu, n) &= \begin{bmatrix} r_n^{(m)}(\nu) & \cdots & r_{n+N_t-1}^{(m)}(\nu) \end{bmatrix}^{\mathrm{T}} && (3.14)\\
&= w^{(m)}(\nu, n) + A(\mathcal{L})\, h_{\mathcal{L}}^{(m)} && (3.15)\\
A(\mathcal{L}^{\max}) &= \begin{bmatrix} a_0 & & \\ \vdots & \ddots & \\ a_{L^{\max}-1} & \cdots & a_0 \\ \vdots & & \vdots \\ a_{N_t-1} & \cdots & a_{N_t-L^{\max}} \end{bmatrix} && (3.16)\\
h_{\mathcal{L}^{\max}}^{(m)} &= \begin{bmatrix} h_0^{(m)} & \cdots & h_{L^{\max}-1}^{(m)} \end{bmatrix}^{\mathrm{T}} && (3.17)
\end{aligned}$$

$$\begin{aligned}
\hat{h}_{\mathcal{L}}^{(m)}(\nu, n, \mathcal{L}) &= A(\mathcal{L})^{+}\, r^{(m)}(\nu, n) && (3.18)\\
\hat{w}^{(m)}(\nu, n, \mathcal{L}) &= \left[\mathrm{I} - A(\mathcal{L})\, A(\mathcal{L})^{+}\right] r^{(m)}(\nu, n) && (3.19)\\
\left[\hat{\sigma}_w^{(m)}(\nu, n, \mathcal{L})\right]^2 &= \frac{\left\|\hat{w}^{(m)}(\nu, n, \mathcal{L})\right\|^2}{N_t - |\mathcal{L}|} && (3.20)
\end{aligned}$$

$$\begin{aligned}
\left[\sigma_{\mathrm{eff}}^{(m)}(\nu, n, \mathcal{L})\right]^2 &= \left[\hat{\sigma}_w^{(m)}(\nu, n, \mathcal{L})\right]^2 \left(1 + \mathrm{tr}\left\{\left[A(\mathcal{L})^{+} A(\mathcal{L})^{+\mathrm{H}}\right]^{-1}\right\}\right) && (3.21)\\
J\left(\nu, n^{(0)}, \mathcal{L}^{(0)}, n^{(1)}, \mathcal{L}^{(1)}\right) &= \sum_{m=0}^{1} \frac{\left\| r^{(m)}\left(0, n_3^{(m)}\right)\right\|^2}{\left[\sigma_{\mathrm{eff}}^{(m)}\left(\nu, n^{(m)}, \mathcal{L}^{(m)}\right)\right]^2} && (3.22)
\end{aligned}$$

The Doppler shift ν and CIR $h_l^{(m)}$ are assumed time-invariant during the training period. The data samples $r_n^{(m)}(\nu)$ are obtained by resampling/interpolating the received data to compensate for the indicated Doppler shift. (Details of this process are given in [62].) The AWGN vector $w^{(m)}(\nu, n)$ is assumed to have statistics independent of the parameters ν, n, m. The index set $\mathcal{L}^{\max} = \{0, \ldots, L^{\max} - 1\}$ and $0 \in \mathcal{L} \subset \mathcal{L}^{\max}$. The matrix $A(\mathcal{L})$ is composed of those columns of $A(\mathcal{L}^{\max})$ corresponding to the index set $\mathcal{L}$, and $h_{\mathcal{L}}^{(m)}$ is the vector of CIR gains associated with these delays. The product $A(\mathcal{L})\, h_{\mathcal{L}}^{(m)}$ is the convolution of the training symbols with the CIR. This formulation, although notationally burdensome, allows for estimating the CIR on noncontiguous sets of delays–i.e. "sparse" CIR estimation. The CIR estimate (3.18) is LS, where $(\cdot)^{+}$ denotes pseudo-inverse, and (3.19) is the corresponding noise vector estimate whose elements have the unbiased variance estimate (3.20), where $|\mathcal{L}|$ denotes the number of elements in the set $\mathcal{L}$. The effective variance (3.21) is discussed in detail in [62]–basically the multiplicative factor accounts for the projection of noise onto the CIR estimate.

The CF (3.22) extends the CF (3.21) by accommodating multiple subchannels and allows the CIR subchannels to be estimated on different sets of delays. It would need to be evaluated for sets of values of its arguments. In particular, we would have $n_1^{(m)} \leq n^{(m)} \leq n_2^{(m)}$ for some $n_1^{(m)}, n_2^{(m)}$ derived from the coarse synchronization index $\hat{n}_0$. The value $n_3^{(m)} > n_2^{(m)}$ is such that $\left\| r^{(m)}\left(0, n_3^{(m)}\right)\right\|^2 \cong N_t \left[\left(\sigma_w^{(m)}\right)^2 + \left\|h^{(m)}\right\|^2\right]$. Consequently $J/N_t - 2$ represents an effective matched-filter SNR–i.e. the sum of the individual subchannel SNRs–which is consistent with the input SNR discussed in Subsection 3.2.6. Ideally the parameter estimates would be chosen to maximize J. However, it is not clear how to efficiently search the parameter space of this general expression, so we impose various constraints and adopt the following strategy.

We constrain $n^{(0)} = n^{(1)} = n$ and $\mathcal{L}^{(0)} = \mathcal{L}^{(1)} = \mathcal{L}$. Initially we also set $\mathcal{L} = \mathcal{L}^{\max}$. We could compute $J(\nu, n) = J(\nu, n, \mathcal{L}^{\max}, n, \mathcal{L}^{\max})$ but, given that $\left(\sigma_w^{(0)}\right)^2 = \left(\sigma_w^{(1)}\right)^2$, we instead compute

$$[\hat{\sigma}_w(\nu, n)]^2 \quad = \quad \frac{1}{2}\sum_{m=0}^{1}\left[\hat{\sigma}_w^{(m)}(\nu, n, \mathcal{L}^{\max})\right]^2 \tag{3.23}$$

over a coarse grid of values for ν, n. We do not constrain the subchannel noise variances to be equal in the CF (3.22) because in general they include effects of CIR estimation error in addition to the ambient noise variance. We maximize $[\hat{\sigma}_w(\nu, n)]^2$ over n and maximize a cubic spline (CS) fit over ν to obtain $\hat{\nu}$. At this point we compute a rough CIR estimate by which we establish a final resampling reference index k_0. Its value is not crucial, but we try to select it so that reference time $t = 0$ lies in the middle of the CIR corresponding to the first transmitted symbol. Next we resample to obtain data $r_n^{(m)} = r_n^{(m)}(\hat{\nu})$ and compute the CF $J(\hat{\nu}, n)$ over a finer grid for n, which we maximize over n to refine the synchronization index $\hat{n}_0$.

The CF $J(L) = J(\hat{\nu}, \hat{n}, \mathcal{L}_L, \hat{n}, \mathcal{L}_L)$ can be efficiently computed for the subsets $\mathcal{L}_L = \{0, \ldots, L-1\}$ by utilizing a QR decomposition of $A(\mathcal{L}^{\max})$. This motivates the following procedure. We estimate the CIR over the full span $\mathcal{L}^{\max}$ and compute the statistic $\psi_l = \max_m \left[\left|\hat{h}_l^{(m)}\right|^2 \Big/ \hat{\chi}_l^{(m)}\right]$, where $\hat{\chi}_l^{(m)}$ is an estimate of the noise variance projected onto the CIR estimate–i.e. the diagonal of $\left[\hat{\sigma}_w^{(m)}(\hat{\nu}, \hat{n}, \mathcal{L}^{\max})\right]^2 \left[A(\mathcal{L}^{\max})^{+} A(\mathcal{L}^{\max})^{+\mathrm{H}}\right]^{-1}$. We then use this statistic to reorder the columns of $A(\mathcal{L}^{\max})$ so that we can evaluate the CF using the subsets

$\mathcal{L}_L$. We basically reorder the columns of $A(\mathcal{L}^{\max})$, from left to right, corresponding to decreasing values of ψ_l. However, if we select a particular column of $A(\mathcal{L}^{\max})$, then we first place its immediately adjacent columns in the reordered matrix (unless they have already been taken). Since we assume small but nonzero Doppler spread, this additional requirement allows CIR arrivals to shift relative to one another by up to one symbol without having to update the set of delays at which the CIR is estimated. Once this procedure is completed, the CIR and the diagonal of its covariance matrix are estimated for use in the first iteration–we adjust the index sets and so forth such that the symbol convolution matrix used in the CIR estimate contains the first column of $A(\mathcal{L}^{\max})$. We also compute the noise variance estimates $\left(\hat{\sigma}_w^{(m)}\right)^2$.

3.3.3 Operations Performed During Equalization

As stated at the beginning of this section, these operations are performed simultaneously during the first iteration, then sequentially during subsequent iterations. The resampling procedure is explained thoroughly in [62].

CIR Estimation

This operation has two key elements: determining which CIR arrivals are significant, and estimating the gains of those CIR arrivals. The training routine performs both these functions together at the beginning of the first iteration, while subsequent iterations perform them separately as described in the next paragraph. During every iteration the set of delays at which the CIR is estimated is held fixed. Details of the procedure for estimating the gains of the CIR arrivals are given in Appendix 3A . The upshot is the channel model (3.6) is put in vector form and linearized about the current CIR estimate to obtain an extended Kalman filter (EKF) [3], and the complex gains are constrained to have a different rate of time variation in their phases than in their amplitudes–consistent with the model developed in Subsection 3.2.3. We incorporate the SMEs into the channel model according to the following rationale that we state symbolically: $r = w + h * (\bar{a} + \Delta a) = (w + h * \Delta a) + h * \bar{a}$, where $*$ denotes convolution and Δa represents the error in the SMEs. Consequently in the appendix we use SMEs in the symbol regression vector and $\left(\hat{\sigma}_w^{(m)}\right)^2 + \sigma_{\bar{a}}^2 \widehat{\left\|h^{(m)}\right\|^2}$ as the observation noise variance, where $\widehat{\left\|h^{(m)}\right\|^2}$ is a (theoretically) unbiased estimate of the CIR energy–i.e. it is the squared norm of the CIR estimate minus the trace of the CIR covariance matrix. For the phase random walk variance we use $\sigma_{\Delta\theta}^2 = 3 \times 10^{-4}$ and for the amplitude we use $\sigma_{\Delta\rho}^2 = \left(10^{-6}\right) \widehat{\left\|h_{\text{Trng}}^{(m)}\right\|^2}$, where $\widehat{\left\|h_{\text{Trng}}^{(m)}\right\|^2}$ is the energy in the training CIR estimate. The EKF we implement has complexity linear in the number of arrivals at which the CIR is estimated.

A different process is used for iterations after the first. The CIR is estimated over the full span of $L^{\max}$ symbols and we toggle the span so that for even iterations the first estimated delay is the same as in the previous iteration, while for odd iterations the last estimated delay is the same as in the previous iteration. We then estimate the CIR gains as explained above and in Appendix 3A , using PSMEs instead of ASMEs and using the training symbols and corresponding data for initialization. Once the time-varying CIR is estimated, we form the statistic $\psi_l = \max_m \left[(1 /|\mathcal{I}|) \sum_{n\in\mathcal{I}} \left(\left| \hat{h}_{l,n}^{(m)} \right|^2 \Big/ c_{l,n}^{(m)} \right) \right]$, where $c_{l,n}^{(m)}$ is the variance of $\hat{h}_{l,n}^{(m)}$ computed by the EKF routine and $\mathcal{I}$ is an index set which selects every 100th symbol. The statistic is compared to the threshold $\left(1 - \sigma_{\bar{a}}^2\right) [1] + \sigma_{\bar{a}}^2 [1.2]$ to select the estimated CIR arrivals to be used in constructing the equalizer filters. The procedure limits the impact of CIR estimation error by only selecting those arrivals with significant energy.

Equalization

This operation has several components. First is construction of the equalizer filters–details of which are provided in Appendix 3B . We note the "equalization" or "forward" filter, which operates on the data, is simply a linear equalizer and consequently can be computed efficiently. Also, given the rate of time-variation of the channel it is sufficient to update the equalizer filters every $L^{\max}$ symbols–i.e. every 100 msec. Since no information is available from the decoder during the first iteration there is consequently no "cancellation" filter. During subsequent iterations, however, and assuming successful performance of the algorithm, the "equalization" filter tends toward a matched-filter and the equalizer as a whole becomes essentially a soft-output two-sided decision-feedback equalizer (DFE) [53]. We incorporate CIR estimation error into the channel model according to the following rationale that we state symbolically: $r = w + \left(\hat{h} + \Delta h\right) * a = (w + \Delta h * a) + \hat{h} * a$, where Δh represents the error in the CIR estimate. Consequently in the appendix we use $\left(\hat{\sigma}_w^{(m)}\right)^2 + \mathrm{E}\left\|\Delta h^{(m)}\right\|^2$ as the observation noise variance, where the rightmost summand is the trace of Cov $\left[\hat{\boldsymbol{h}}_n^{(m)}\right]$ which is obtained from the EKF.

The second component of equalization is actually filtering the Doppler-compensated data with the "equalization" filter and filtering the PSMEs with the "cancellation" filter.

The third component is estimating the EOMPs. As noted in Subsection 3.2.4, if the CMPs were known exactly then the gain and effective noise variance at the equalizer output would also be known exactly, and could be computed as shown in Appendix 3B . Details of the procedure for estimating the EOMPs are given in Appendix 3C . The basic idea is Gaussian approximations for pdf's produce an EKF-like algorithm. The random-walk variances for the EOMPs are $\sigma_{\Delta\omega}^2 = \left(10^{-4}\right) \hat{\omega}_n^2$, $\sigma_{\Delta\gamma}^2 = 10^{-5}$, and $\sigma_{\Delta\phi}^2 = 10^{-3}$.

The final component of equalization is computing symbol APPs which are passed to the decoder. We define the equalizer "soft" output $\tilde{a}_n = \exp\left(-\mathrm{j}\hat{\phi}_n\right) z_n - q_n$ and estimate the symbol APPs as

$$\hat{p}\left(a_n = \alpha \,|\cdot\right) \quad \propto \quad \exp\left(-\frac{1}{2}\hat{\omega}_n \left|\hat{\gamma}_n \alpha - \tilde{a}_n\right|^2\right) \tag{3.24}$$

$$\propto \quad \exp\left(\hat{\omega}_n \hat{\gamma}_n \mathrm{Re}\left[\tilde{a}_n^* \alpha\right]\right) \tag{3.25}$$

where the last expression is a consequence of the symbol alphabet having constant amplitude. (The symbol APP's are normalized to have unit sum.)

Doppler Shift Updating

During the first iteration we update the Doppler Shift estimate by using proportional-integral (PI) control of the change in phase rotation $\Delta\hat{\phi}_n = \hat{\phi}_n - \hat{\phi}_{n-1}$:

$$\hat{\nu}_n \quad = \quad \hat{\nu}_{n-1} + K_{\text{prop}}\Delta\hat{\phi}_n + K_{\text{integ}}\hat{\phi}_n \tag{3.26}$$

where$K_{\text{prop}} = 10^{-4}$ and $K_{\text{integ}} = 10^{-7}$.

Final Resampling

After the first equalization is complete, the sampling times estimates are smoothed and a final resampling/interpolation of the data is performed. The sampling times estimates computed as part of the Doppler shift tracking procedure are simply smoothed forward and backward with an IIR filter, with care taken at the ends, and then adjusted using the estimated phase sequence in the manner described in Subsection III-D of [62].

3.4 Results and Discussion

The algorithm processes data received on a single sensor. There were 32 sensors in the receive sensor array and 51 total transmissions, so there are 1,632 total sequences available for processing. A summary of the results of processing all of these sequences is shown in Table 3.1. The algorithm did not fail for any of the 768 sequences from data set A, it did fail 4 times out of the 864 sequences in data set B–which is slightly less than 0.5%. By “failure” we mean the algorithm does not converge and essentially no information is extracted from a particular data packet. We consider the algorithm “successful” if it recovers the transmitted information with a small number of bit errors. For the cases of successful processing for both data sets the frequency of bit errors was less than 1 error in 100,000 bits. Fig. 3.6 shows details of the processing results for data set A. (Note: there is no data corresponding to time 7:40.) Part (a) of the figure shows the number of bit errors produced by the algorithm marked in text, with the vertical axis indicating sensor/depth and the horizontal axis indicating the start time of the transmission.

There is no apparent pattern to the distribution of errors and the maximum number of errors in any transmission is nine–or about 1 error per 1000 bits. Part (b) of the figure shows the average Doppler shift measured by the algorithm, each of the 32 sensors corresponds to a dot which is plotted against the transmission start time. The lack of variability in the Doppler shift measurements reflects the near constant velocity of the transmission source during this period of time. Fig. 3.7 shows the same processing details for data set B. In part (a) of the figure the pound signs indicate failure of the algorithm. Again there is no apparent pattern to the distribution of errors–or failures. Part (b) of the figure reflects the fact that the radial component of the source velocity increases during this time period, which agrees with the source track for data set B shown in Fig. 3.1.

Table 3.1: Summary of Performance Results

	Data Set A	Data Set B
Number of Transmissions	24	27
Total Number of Sequences	768	864
Number of Sequence Failures	0	4
Sequence Failure Rate	0%	0.5%
Number of Successful Sequences with Bit Errors	12	9
Total Number of Bit Errors for Successful Sequences	49	28
Total Number of Bits in Successful Sequences	6,916,608	7,745,160
Bit Error Frequency	7.1×10^{-6}	3.6×10^{-6}

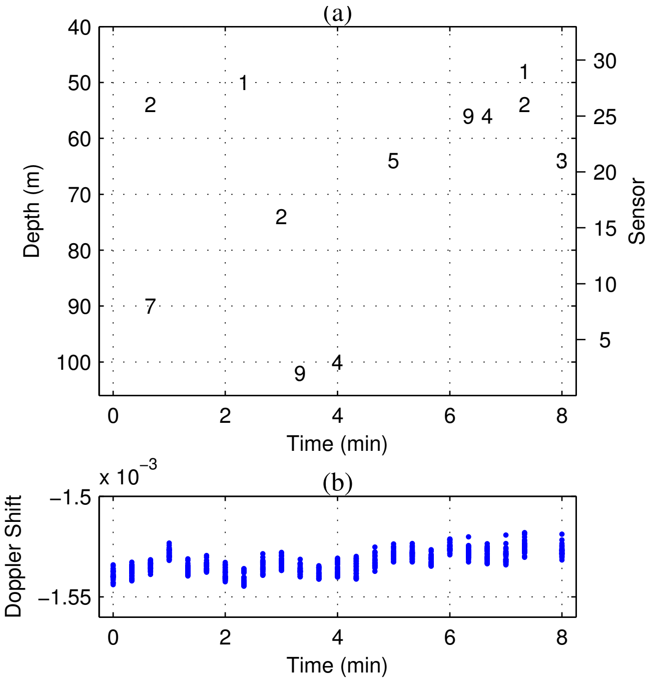


Figure 3.6: Data set A. (a) Number of bit errors marked with text versus depth/sensor on the vertical axis and transmission start time on the horizontal axis. (b) Average Doppler shift measured by the algorithm: each dot corresponds to one of the 32 sensors, horizontal axis is transmission start time.

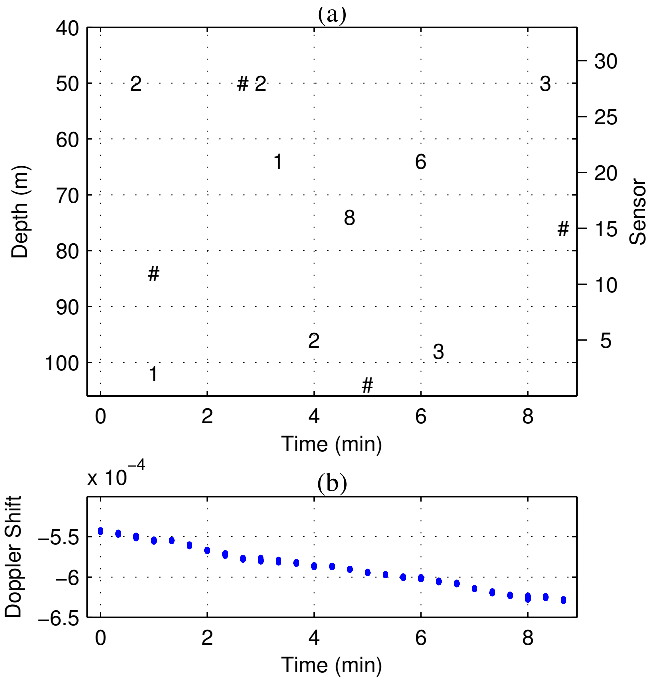


Figure 3.7: Data set B. (a) Number of bit errors marked with text versus depth/sensor on the vertical axis and transmission start time on the horizontal axis–pound signs indicate algorithm failures. (b) Average Doppler shift measured by the algorithm: each dot corresponds to one of the 32 sensors, horizontal axis is transmission start time.

The four instances of algorithm failure reported above were closely examined by monitoring various measures of system performance. It was ascertained the failures did not stem from break-down of the phase-tracking/Doppler-tracking procedure or any other specific task within the algorithm. Rather, in each of the four cases the equalizer output SINR was simply too low for the algorithm to start converging to a successful steady-state. The particular combination of channel and noise conditions produced inadequate equalizer output in these four cases. Since there is nothing to be gleaned from examining the various measures of system performance in these four cases–other than that the algorithm simply failed to iteratively improve its knowledge of the bits, symbols and CMPs–we will instead focus on a case where the algorithm was successful.

Having shown results of overall system performance for the aggragate data, we now show detailed results of processing one particular data sequence. That data sequence is the same one considered in Subsection 3.2.3 (sensor 6, at 94 m depth, at the very beginning of data set A), where its time-varying CIR was measured and discussed. This data sequence is not necessarily "typical" but, rather, appears to be close to the threshold of what the algorithm can process successfully. Fig. 3.8 elucidates this last statement. It shows the PSME variance for the second iteration of the algorithm for data set A (recall that the PSME variance is unity for the first iteration), with the vertical axis indicating sensor/depth and the horizontal axis indicating the start time of the transmission. (The corresponding graphic for data set B is essentially the same and is not provided.) Not only do the PSME variances virtually cover the entire range of possible values from zero to one, but their distribution is apparently random. Considering the particular data sequence to be considered forthwith, located at time zero and depth 94 m, its PSME variance is close to unity–indicating very weak symbol information–while those corresponding to the same sensor at later times is much lower. There is not even a pattern along this single row. In any case, the selected data sequence clearly demonstrates the performance of the iterative algorithm.

Fig. 3.9 shows initial CMP estimates produced by the training routine. Part (a) is the CF (3.23) normalized by the true noise variance, used to estimate the Doppler shift–assumed constant during the training period–together with a CS fitted through the points at which the CF is evaluated, and the extremum marked with a circle. Part (b) shows, marked with dots, the magnitude of the CIR estimate used to initialize the EKF, with a high-SNR CIR measurement shown as a solid curve for comparison. The routine discussed in Subsection 3.3.2 does a reasonable job of capturing the significant portions of the CIR. Some energy near delays -12 msec and 32 msec is missed, but it is not substantial, and there is a false detection near delay 63 msec.

Fig. 3.10(a) shows the CIR estimation error for the first and last iterations of the algorithm. The top two curves (one for each "over-sampled subchannel") are the result of applying a

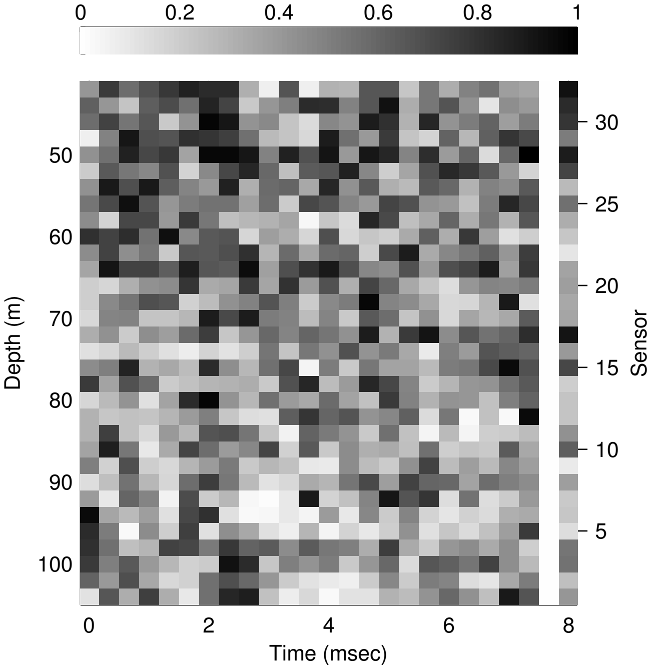


Figure 3.8: PSME variance for second iteration of algorithm for data set A: vertical axis is sensor/depth, horizontal axis is transmission start time, grayscale intensity is PSME variance.

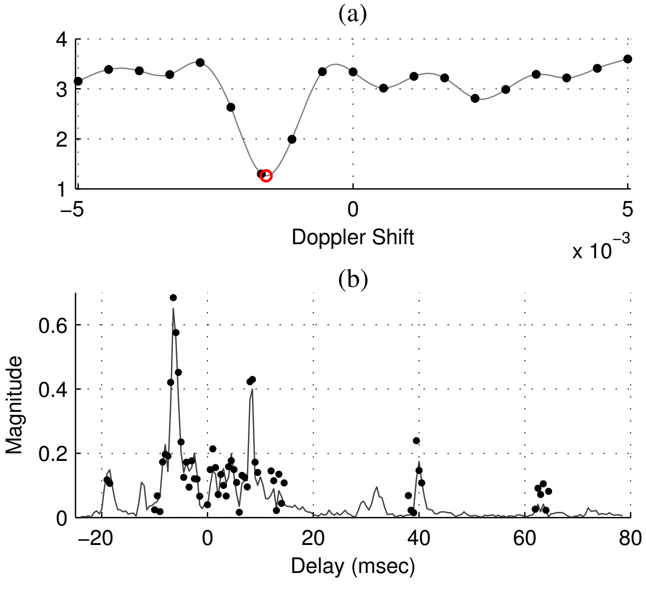


Figure 3.9: (a) Metric used to estimate Doppler shift: actual values computed are shown with dots, gray curve is a CS through the values, extremum is marked with a circle. (b) Magnitude of CIR estimate is shown with dots, gray curve is a CIR measurement computed at high SNR.

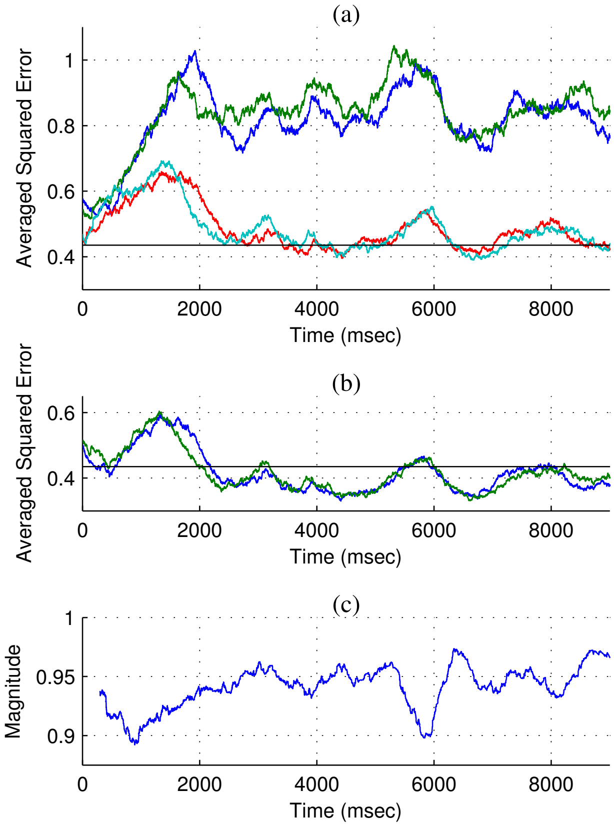


Figure 3.10: (a) Moving-average of squared-error in estimating CIR: top two curves are first iteration, bottom two curves are last iteration, straight line is average ambient noise variance. (b) Moving-average ambient noise variance measurement. (c) Magnitude of moving-average of symbol-quality statistic for last iteration.

rectangular moving-average filter of length 501 symbols to the squared-error in the CIR estimates produced during the first iteration, the bottom two curves are for the last iteration. The true symbols were used in these computations. The solid line is the (true) average ambient noise variance. Although not shown, the estimation error for intermediate iterations falls between these two sets of curves. The dramatic improvement in quality of the CIR estimates is evident and, as shown below, this improvement directly affects the strength of the symbol information. Part (b) of the figure shows moving-average measurements of the ambient noise variance (for the two subchannels) obtained directly from the associated noise sequence. Comparing these two curves with the two curves in part (a) corresponding to the last iteration, it can be seen that the CIR estimation routine tracks the channel fairly well, albeit with small estimation error. Part (c) is the magnitude of the moving-average of the ASME statistic $a_n^* \bar{a}_n$ (computed using only data-bearing symbols). It is included for comparison with the CIR estimation error of the last iteration: the major valleys in the curve from part (c) correspond to the major peaks in the curves from part (a).

Fig. 3.11 shows the performance of the EOMP tracking routine for the first and last iterations. In parts (a) and (b) there is clear improvement in the quality of the equalizer output: the bottom curves correspond to the first iteration and the top curves to the last. Part (c) shows the phase estimated during the first iteration (it is not estimated in subsequent iterations). Parts (d) and (e) show the magnitude and phase of the moving-average of the ASME statistic $a_n^* \bar{a}_n$ (computed using only data-bearing symbols). Again the improvement in equalization from the first iteration to the last is clear: the magnitude of the statistic increases from below 0.5 to above 0.9 and the phase fluctuations decrease from 15 degrees to just a few degrees.

A summary of the performance of the algorithm for the selected data sequence is shown in Fig. 3.12. Part (a) is a synopsis of the CIR estimation error produced during each iteration of the algorithm: it is the average squared-error normalized by the average noise variance. Using the true symbols we made RWLS estimates of the complex gain $\gamma_n \exp(\mathrm{j}\phi_n)$ and effective noise variance $\mathrm{E}\,|v_n|^2$ at the output of the equalizer filter. The squared magnitude of the former divided by the latter is a measure of the output SINR, and the mean of this quantity is shown for each iteration in part (b) of the figure. The SINR steadily increases with iteration until reaching a plateau just below the input SNR. Part (c) shows the average of the absolute value of the phase of the gain estimates and it too demonstrates fairly steady improvement with iteration. (The zigzag behavior is a result of the resynchronization procedure discussed in part 3.3.3 of this paper.) Part (d) of the figure shows the number of bit errors that would result if the algorithm were terminated at the end of each iteration, and part (e) shows the PSME variance for each iteration. Overall the algorithm steadily improves knowledge of the bits, symbols, and CMPs

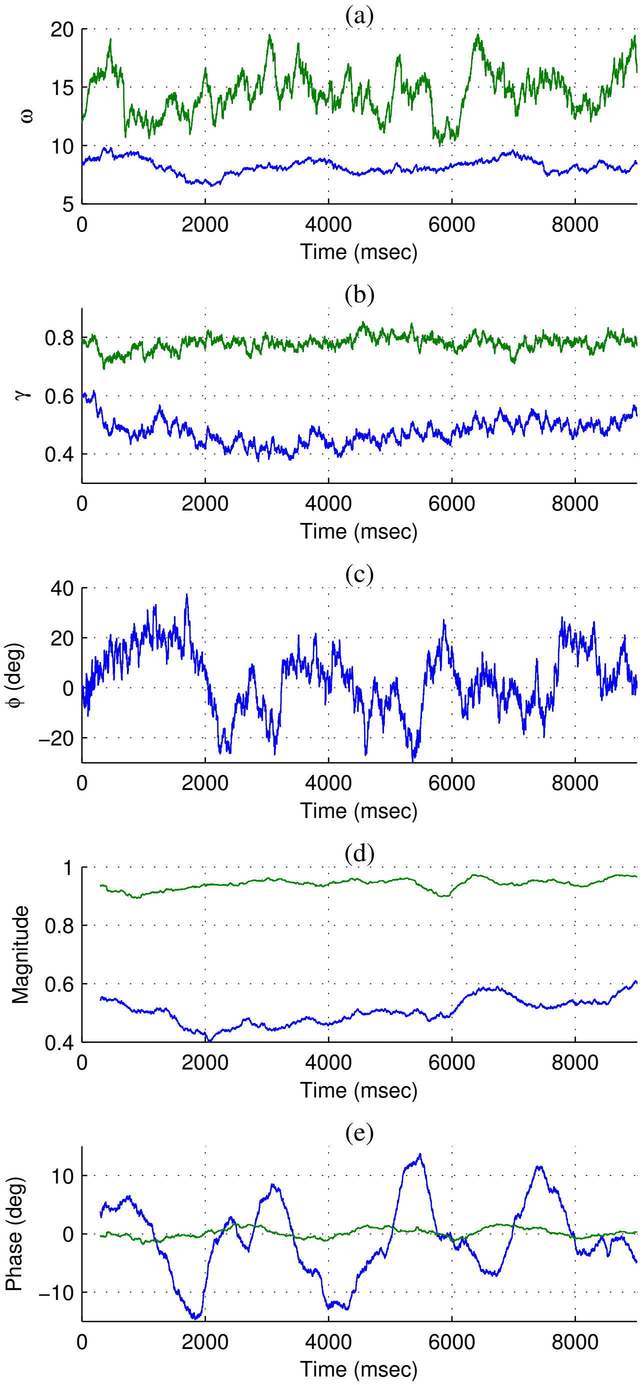
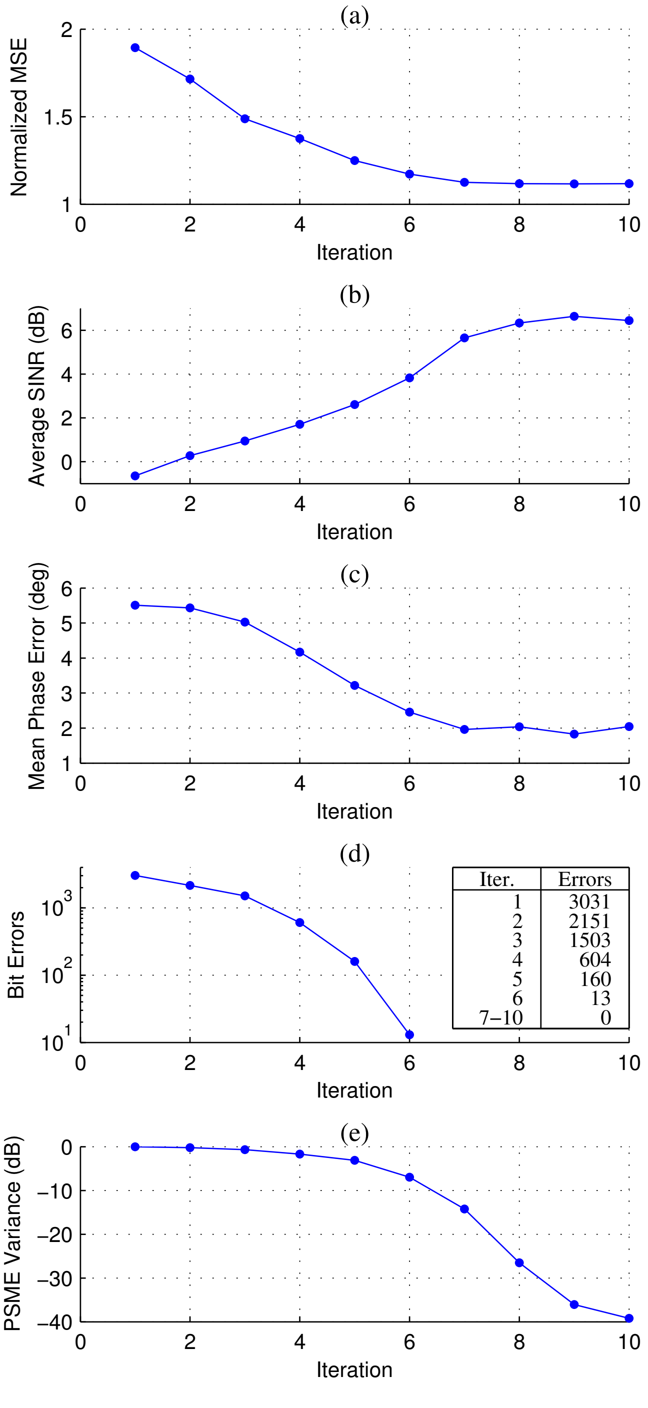


Figure 3.11: (a,b,c) EOMP estimates: in (a) and (b) bottom curve is first iteration, top curve is last iteration. (d,e) Magnitude and phase of moving-average of symbol-quality statistic: in (d) bottom curve is first iteration, top curve is last iteration.



Iter.	Errors
1	3031
2	2151
3	1503
4	604
5	160
6	13
7−10	0

Figure 3.12: (a) Mean squared-error of CIR estimation, normalized by average noise variance, versus iteration. (b) Average SINR at equalizer output versus iteration. (c) Mean absolute phase error versus iteration. (d) Number of bit errors versus iteration. (e) PSME variance versus iteration.

with each iteration.

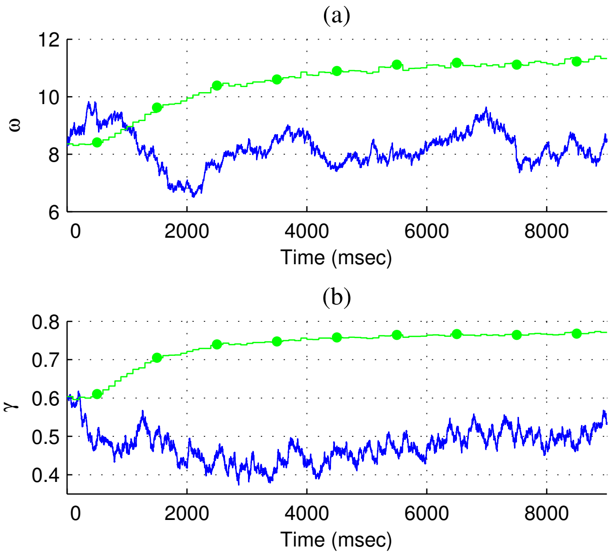


Figure 3.13: EOMPs: upper curves are theoretical values calculated from CMP estimates, lower curves are estimates obtained directly from equalizer output.

The importance of estimating the EOMPs directly as opposed to using the theoretical quantities derived in Appendix 3B is demonstrated in Fig. 3.13. Part (a) of the figure shows the noise statistic computed during the first iteration of the algorithm and part (b) shows the gain. In both parts the upper curves are theoretical values while the lower curves are direct estimates. The output SINR is approximately $(1/2)\,\hat{\omega}\hat{\gamma}^2$, which is about $(1/2)\,(11)\,(0.75)^2 = 3.1$ or 4.9 dB in the former case, and $(1/2)\,(8)\,(0.5)^2 = 1$ or 0 dB in the latter case. From measuring the output SINR using the true symbols we know the latter SINR estimate is correct. Consequently, using the theoretical values in this case would result in overestimating the SINR by about 5 dB, which would significantly reduce the reliability of the symbol APP estimates passed to the decoder.

3.5 Conclusion

In this paper we introduced an IED algorithm and applied it to communication data received from a moving UWA source. The algorithm is an instance of "turbo equalization", which represents a powerful technique in signal processing. Consequently the algorithm is well-matched to one of the main characteristics of the shallow UWA channel–namely, CIRs with large delay spread. The other characteristics–Doppler distortion and bandwidth limitations–the system addresses by resampling to compensate for the former and using QPSK modulation to efficiently utilize the latter. Additionally, we developed and implemented a CIR estimation scheme closely matched to the actual time-variation observed in the data. The algorithm has several novel aspects, which include its method of synchronization and CIR estimation, as well as its computation of symbol APPs, estimation of Doppler shift, and nonuniform resampling to compensate for Doppler distortion.

The algorithm was used to process over 1,600 data sequences and performed with a high degree of success. Only four times did the algorithm fail to recover the transmitted bit sequence with few or no errors. For the cases of successful operation, the algorithm achieved a bit error frequency of less than 1 error in 100,000 bits, with no more than 1 error per 1,000 bits in any single case. Numerous figures detail the performance of processing one particular data sequence. They clearly demonstrate the iterative improvement in knowledge of the transmitted bits and symbols and all the various model parameters needed to implement the equalization process. For the given example, the algorithm displayed steady improvement with each iteration.

3.6 Acknowledgments

This work was supported by the Office of Naval Research under Grant N00014-05-1-0263.

Chapter 3, in full, is a reprint of the material as it appears in “Iterative Equalization and Decoding of Communication Data Received from a Moving Underwater Acoustic Source”, J. F. Sifferlen, H. C. Song, W. S. Hodgkiss, W. A. Kuperman, and J. M. Stevenson, *IEEE Journal of Oceanic Engineering*, submitted 23 July 2008. The dissertation author was the primary investigator and author of this paper.

3A CIR EKF Routine

The model (3.6) is expressed in vector form as $\exp\left(-\mathrm{j}\phi_n\right) r_n^{(m)} = w_n^{(m)} + \boldsymbol{a}_n^{\mathrm{T}} \boldsymbol{h}_n^{(m)}$. We ignore phase rotation of the noise sample since it is assumed circularly symmetric, $\boldsymbol{a}_n$ is a vector of symbols and $\boldsymbol{h}_n^{(m)}$ is a vector of CIR arrivals with elements $h_{l,n}^{(m)} = \rho_{l,n}^{(m)} \exp\left(\mathrm{j}\theta_{l,n}^{(m)}\right)$. We define the “state” vector $x_n^{(m)} = \left[\boldsymbol{\rho}_n^{(m)}; \boldsymbol{\theta}_n^{(m)}\right]$, where the semicolon indicates vertically concatenating the two constituent vectors. Given a prediction of the CIR $\acute{\boldsymbol{h}}_n^{(m)}$ we make the linear approximation $\boldsymbol{a}_n^{\mathrm{T}} \boldsymbol{h}_n^{(m)} \cong \boldsymbol{a}_n^{\mathrm{T}} \acute{\boldsymbol{h}}_n^{(m)} + \left(b_n^{(m)}\right)^{\mathrm{T}} \Delta x_n^{(m)}$, where $\Delta x_n^{(m)}$ is the variation of the state vector about its predicted value $\acute{x}_n^{(m)}$. The vector $b_n^{(m)} = \left[\mathrm{D}\left(\boldsymbol{a}_n\right) \exp\left(\mathrm{j}\acute{\boldsymbol{\theta}}_n^{(m)}\right); \mathrm{j}\mathrm{D}\left(\boldsymbol{a}_n\right) \acute{\boldsymbol{h}}_n^{(m)}\right]$, where $\mathrm{D}\left(\cdot\right)$ is the diagonal matrix constructed from its vector argument and the exponential function is applied to each element of its vector argument. The following state-space model results:

$$\begin{aligned} y_n^{(m)} &= \begin{bmatrix} \mathrm{Re}\left[\exp\left(-\mathrm{j}\phi_n\right) r_n^{(m)} - \boldsymbol{a}_n^{\mathrm{T}} \acute{\boldsymbol{h}}_n^{(m)}\right] \\ \mathrm{Im}\left[\exp\left(-\mathrm{j}\phi_n\right) r_n^{(m)} - \boldsymbol{a}_n^{\mathrm{T}} \acute{\boldsymbol{h}}_n^{(m)}\right] \end{bmatrix} \qquad (3.27) \\ &\cong \ddot{w}_n^{(m)} + \left(B_n^{(m)}\right)^{\mathrm{T}} \Delta x_n^{(m)} \qquad (3.28) \\ B_n^{(m)} &= \left[\mathrm{Re}\left[b_n^{(m)}\right], \mathrm{Im}\left[b_n^{(m)}\right]\right] \qquad (3.29) \\ \ddot{w}_n^{(m)} &= \left[\mathrm{Re}\left[w_n^{(m)}\right]; \mathrm{Im}\left[w_n^{(m)}\right]\right] \qquad (3.30) \\ \mathrm{Cov}\left[\ddot{w}_n^{(m)}\right] &= \frac{1}{2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \mathrm{E}\left|w_n^{(m)}\right|^2 \qquad (3.31) \end{aligned}$$

Given a covariance matrix $\acute{C}_n^{(m)}$ of the predicted state we compute the state correction and covariance as

$$\begin{aligned} \widehat{\Delta x}_n^{(m)} &= K_n^{(m)} y_n^{(m)} \qquad (3.32) \\ C_n^{(m)} &= \acute{C}_n^{(m)} - K_n^{(m)} \left(B_n^{(m)}\right)^{\mathrm{T}} \acute{C}_n^{(m)} \qquad (3.33) \\ K_n^{(m)} &= \acute{C}_n^{(m)} B_n^{(m)} \left[\mathrm{Cov}\left[\ddot{w}_n^{(m)}\right] + \left(B_n^{(m)}\right)^{\mathrm{T}} \acute{C}_n^{(m)} B_n^{(m)}\right]^{-1} \qquad (3.34) \end{aligned}$$

where $K_n^{(m)}$ is the Kalman gain matrix. During the second and subsequent iterations of the algorithm we compute the corrected state as $\hat{x}_n^{(m)} = \acute{x}_n^{(m)} + \widehat{\Delta x}_n^{(m)}$. During the first iteration, however, we want to prevent bulk phase rotation of the CIR. Consequently we replace $\widehat{\Delta \boldsymbol{\theta}}_n = \left[\widehat{\Delta \boldsymbol{\theta}}_n^{(0)}; \widehat{\Delta \boldsymbol{\theta}}_n^{(1)}\right]$ with $\widehat{\Delta \boldsymbol{\theta}}_n - \mathbf{1}\Delta\phi_n$, where $\mathbf{1}$ (boldface) is a vector all of whose elements are one. Based on discussion in Section 3.2.3, we choose $\Delta\phi_n$ such that the inner product of the predicted and corrected CIR estimates has zero imaginary part. Since $\acute{\boldsymbol{h}}_n^{\mathrm{H}} \hat{\boldsymbol{h}}_n \cong \acute{\boldsymbol{\rho}}_n^{\mathrm{T}} \left\{\mathrm{I} + \mathrm{j}\left[\mathrm{D}\left(\widehat{\Delta \boldsymbol{\theta}}_n\right) - \mathrm{I}\Delta\phi_n\right]\right\} \hat{\boldsymbol{\rho}}_n$, where the vectors include both subchannel components (hence no superscripts) and we make use of the small angle approximation $\exp\left[\mathrm{j}\left(\cdot\right)\right] \cong 1 + \mathrm{j}\left(\cdot\right)$, then $\Delta\phi_n = \acute{\boldsymbol{\rho}}_n^{\mathrm{T}} \mathrm{D}\left(\widehat{\Delta \boldsymbol{\theta}}_n\right) \acute{\boldsymbol{\rho}}_n \Big/ \left\|\acute{\boldsymbol{\rho}}_n\right\|^2$.

We assume no predictable state dynamics so the prediction step is simply $\acute{x}_n^{(m)} = \hat{x}_{n-1}^{(m)}$ and $\acute{C}_n^{(m)} = C_{n-1}^{(m)} + \mathrm{D}\left(\left[\mathbf{1}\sigma_{\Delta\rho}^2; \mathbf{1}\sigma_{\Delta\theta}^2\right]\right)$, where $\sigma_{\Delta\rho}^2$ and $\sigma_{\Delta\theta}^2$ are the random walk variances for the magnitude and phase components of the CIR.

Since the symbol sequence input to the EKF is white the covariance matrices are nearly diagonal, which presents a natural low-complexity implementation. We simply force the covariance matrices to be diagonal and make the necessary approximations. None are required in the prediction step since the update to the covariance matrix affects only the diagonal, nor in the Kalman gain matrix equation (3.34). With $\acute{C}_n^{(m)} = \mathrm{D}\left(\acute{c}_n^{(m)}\right)$ and $C_n^{(m)} = \mathrm{D}\left(c_n^{(m)}\right)$ the covariance correction (3.33) becomes

$$c_n^{(m)} = \acute{c}_n^{(m)} - \left[K_n^{(m)} \odot \left(\acute{C}_n^{(m)} B_n^{(m)}\right)\right] [1; 1] \qquad (3.35)$$

where $\odot$ represents multiplying the two matrices element-by-element. The expression (3.35) simply computes the diagonal of the exact expression.

The training symbols and corresponding data provide initial estimates of the symbol-rate subchannel noise variances and CIRs and their covariance matrices. Letting $\hat{\boldsymbol{h}}_n^{(m)} = \hat{\boldsymbol{\rho}}_n^{(m)} \odot \exp\left(\mathrm{j}\hat{\boldsymbol{\theta}}_n^{(m)}\right) = \boldsymbol{h}_n^{(m)} + \delta\boldsymbol{h}_n^{(m)}$ and likewise for $\hat{\boldsymbol{\rho}}_n^{(m)}$ and $\hat{\boldsymbol{\theta}}_n^{(m)}$, we have $\delta\boldsymbol{h}_n^{(m)} \cong \exp\left(\mathrm{j}\boldsymbol{\theta}_n^{(m)}\right) \odot \left(\delta\boldsymbol{\rho}_n^{(m)} + \mathrm{j}\boldsymbol{\rho}_n^{(m)} \odot \delta\boldsymbol{\theta}_n^{(m)}\right)$. We further approximate the initial CIR covariance matrices as diagonal and circularly symmetric–i.e. uncorrelated real and imaginary parts–and so we obtain the following initial state covariance matrices

$$\frac{1}{2}\begin{bmatrix} \mathrm{Cov}\left[\hat{\boldsymbol{h}}_0^{(m)}\right] & 0 \\ 0 & \mathrm{D}\left(\hat{\boldsymbol{\rho}}_0^{(m)}\right)^{-2}\mathrm{Cov}\left[\hat{\boldsymbol{h}}_0^{(m)}\right] \end{bmatrix}$$

We also have need for the CIR covariance matrices throughout the iterative algorithm and compute them as $\mathrm{Cov}\left[\hat{\boldsymbol{h}}_n^{(m)}\right] \cong \mathrm{D}\left(\left[\mathrm{I}, \mathrm{D}\left(\hat{\boldsymbol{\rho}}_n^{(m)}\right)^2\right] c_n^{(m)}\right)$.

3B Equalizer Filters

The model (3.8) is expressed as $z_n = \boldsymbol{f}_n\boldsymbol{r}_n$, where $\boldsymbol{f}_n = \left[\boldsymbol{f}_n^{(0)}, \boldsymbol{f}_n^{(1)}\right]$, $\boldsymbol{r}_n = \left[\boldsymbol{r}_n^{(0)}; \boldsymbol{r}_n^{(1)}\right]$ and $\boldsymbol{f}_n^{(m)} = \left[f_{l_2^{\mathrm{EQ}},n}^{(m)}, \ldots, f_{l_1^{\mathrm{EQ}},n}^{(m)}\right]$, $\boldsymbol{r}_n^{(m)} = \left[r_{n-l_2^{\mathrm{EQ}}}^{(m)}; \ldots; r_{n-l_1^{\mathrm{EQ}}}^{(m)}\right]$. (We use semicolons to denote vertical concatenation–i.e. $[x; y] = \left[x^{\mathrm{T}}, y^{\mathrm{T}}\right]^{\mathrm{T}}$.) Similarly the model (3.6) can be expressed as $\boldsymbol{r}_n \cong \boldsymbol{w}_n + \exp(\mathrm{j}\phi_n) H_n\boldsymbol{a}_n$, where $\boldsymbol{w}_n = \left[\boldsymbol{w}_n^{(0)}; \boldsymbol{w}_n^{(1)}\right]$ is noise, $\boldsymbol{a}_n = \left[a_{n-l_2-l_2'}; \ldots; a_n; \ldots; a_{n-l_1-l_1'}\right]$, and $H_n = \left[H_n^{(0)}; H_n^{(1)}\right]$ with $H_n^{(m)}$ representing convolution matrices for the two subchannels. With these definitions we could follow the derivation given in [62] to obtain the equalizer filters by directly applying the SPA in conjunction with certain approximations. However, the results so obtained are equivalent (to within a multiplicative factor) to the MMSE equalizer filters we derive below, and we choose the latter for simplicity.

We need several additional definitions. Let the row vector e and matrix E be such that $e\boldsymbol{a}_n = a_n$ and $E\boldsymbol{a}_n = \left[a_{n-l_2-l_2'}; \ldots; a_{n-1}; a_{n+1}; \ldots; a_{n-l_1-l_1'}\right]$–i.e. E is an identity matrix with a particular row e removed. Note that $Ee^{\mathrm{T}} = 0$ (zero vector). We also use $\mathcal{E}_{\bar{a}} = \mathrm{E}\left|\bar{a}_n\right|^2$ and $\chi_{\bar{a}} = \mathrm{E}\left[a_n^*\bar{a}_n\right]$ defined in Subsection 3.2.5. Denoting the cancellation filter output as $\sum_{l\neq 0}\acute{g}_{l,n}\bar{a}_{n-l} = \acute{\boldsymbol{g}}_n E^{\mathrm{T}}E\bar{\boldsymbol{a}}_n$, the equalizer "soft" output is $\tilde{a}_n = \exp(-\mathrm{j}\phi_n) z_n - \acute{\boldsymbol{g}}_n E^{\mathrm{T}}E\bar{\boldsymbol{a}}_n$. The error can be expressed as $\tilde{a}_n - a_n \cong \boldsymbol{f}_n\boldsymbol{w}_n + (\boldsymbol{f}_n H_n - e)\boldsymbol{a}_n - \acute{\boldsymbol{g}}_n E^{\mathrm{T}}E\bar{\boldsymbol{a}}_n$, where we ignore phase rotation of the noise vector because its elements are assumed circularly symmetric. Consequently the

MMSE is obtained from

$$\begin{aligned}
\mathrm{E}\left|\tilde{a}_n - a_n\right|^2 &\cong \boldsymbol{f}_n C_w \boldsymbol{f}_n^{\mathrm{H}} + \|\boldsymbol{f}_n H_n - e\|^2 + \mathcal{E}_{\bar{a}} \acute{\boldsymbol{g}}_n E^{\mathrm{T}} E \acute{\boldsymbol{g}}_n^{\mathrm{H}} - 2\mathrm{Re}\left[\chi_{\bar{a}} \acute{\boldsymbol{g}}_n E^{\mathrm{T}} E \left(\boldsymbol{f}_n H_n\right)^{\mathrm{H}}\right] \\
&\geq \boldsymbol{f}_n C_w \boldsymbol{f}_n^{\mathrm{H}} + \|\boldsymbol{f}_n H_n - e\|^2 - \frac{\left|\chi_{\bar{a}}\right|^2}{\mathcal{E}_{\bar{a}}} \left\|\boldsymbol{f}_n H_n E^{\mathrm{T}}\right\|^2 \\
&= 1 + \boldsymbol{f}_n Q_n \boldsymbol{f}_n^{\mathrm{H}} - 2\mathrm{Re}\left[\boldsymbol{f}_n H_n e^{\mathrm{T}}\right] \\
&\geq 1 - g_{0,n}
\end{aligned}$$

where C_w is the noise covariance matrix, $Q_n = C_w + H_n H_n^{\mathrm{H}} - \left(\left|\chi_{\bar{a}}\right|^2 \big/ \mathcal{E}_{\bar{a}}\right) H_n E^{\mathrm{T}} E H_n^{\mathrm{H}}$, and $g_{0,n} = \boldsymbol{g}_n e^{\mathrm{T}}$ with $\boldsymbol{g}_n = \boldsymbol{f}_n H_n$. The first inequality above follows from completing the square in $\acute{\boldsymbol{g}}_n$ and equality is achieved by setting $\acute{\boldsymbol{g}}_n = \left(\chi_{\bar{a}}^* / \mathcal{E}_{\bar{a}}\right) \boldsymbol{g}_n$. The second inequality above follows from completing the square in $\boldsymbol{f}_n$ and equality is achieved by setting $\boldsymbol{f}_n = e H_n^{\mathrm{H}} Q_n^{-1}$.

It is straightforward to show that $\tilde{a}_n = v_n + g_{0,n} a_n$, with $v_n = \boldsymbol{f}_n \boldsymbol{w}_n + \boldsymbol{g}_n E^{\mathrm{T}} E \boldsymbol{a}_n - \acute{\boldsymbol{g}}_n E^{\mathrm{T}} E \bar{\boldsymbol{a}}_n$ and $\mathrm{E}\left|v_n\right|^2 = g_{0,n} - g_{0,n}^2$. Also, using the matrix inversion lemma [3] we can express the equalization filter as $\boldsymbol{f}_n = e H_n^{\mathrm{H}} \acute{Q}_n^{-1} \Big/ \left[1 + \left(\left|\chi_{\bar{a}}\right|^2 \big/ \mathcal{E}_{\bar{a}}\right) e H_n^{\mathrm{H}} \acute{Q}_n^{-1} H_n e^{\mathrm{T}}\right]$ with $\acute{Q}_n = C_w + \left(1 - \left|\chi_{\bar{a}}\right|^2 \big/ \mathcal{E}_{\bar{a}}\right) H_n H_n^{\mathrm{H}}$. Furthermore, if $\chi_{\bar{a}} = \mathcal{E}_{\bar{a}}$ then $\acute{\boldsymbol{g}}_n = \boldsymbol{g}_n$ and, since $\mathcal{E}_{\bar{a}} = 1 - \sigma_{\bar{a}}^2$ for PSK symbol alphabets, $\boldsymbol{f}_n = e H_n^{\mathrm{H}} \acute{Q}_n^{-1} \Big/ \left[1 + \left(1 - \sigma_{\bar{a}}^2\right) e H_n^{\mathrm{H}} \acute{Q}_n^{-1} H_n e^{\mathrm{T}}\right]$ and $\acute{Q}_n = C_w + \sigma_{\bar{a}}^2 H_n H_n^{\mathrm{H}}$.

3C EOMP Tracking

We define the "state" as $x_n = \left[\omega_n, \gamma_n, \phi_n\right]^{\mathrm{T}}$ and use the shorthand $Z_n \equiv \left\{z_l\right\}_{l=0}^n$ and likewise for Q_n. We assume AWGN at the equalization filter output. At (symbol) time $n-1$ we have $p\left(x_{n-1} \left| Z_{n-1}, Q_{n-1}\right.\right) \overset{\sim}{\propto} \exp\left(-\left(1/2\right)\|x_{n-1} - \hat{x}_{n-1}\|_{C_{n-1}^{-1}}^2\right)$, where the notation $\|v\|_M^2$ represents the quadratic form $v^{\mathrm{H}} M v$. The "prediction" step is $p\left(x_n \left| Z_{n-1}, Q_{n-1}\right.\right) = \int p\left(x_n \left| x_{n-1}\right.\right)\, p\left(x_{n-1} \left| Z_{n-1}, Q_{n-1}\right.\right) \mathrm{d}x_{n-1} \overset{\sim}{\propto} \exp\left(-\left(1/2\right)\|x_n - \acute{x}_n\|_{\acute{C}_n^{-1}}^2\right)$, where $\acute{x}_n = \hat{x}_{n-1}$ and $\acute{C}_n = C_{n-1} + \mathrm{diag}\left[\sigma_{\Delta\omega}^2, \sigma_{\Delta\gamma}^2, \sigma_{\Delta\phi}^2\right]$. This results because there are no predictable state dynamics. The "propagation" step is $p\left(x_n \left| Z_n, Q_n\right.\right) \overset{\sim}{\propto} p\left(z_n \left| x_n, q_n\right.\right) p\left(x_n \left| Z_{n-1}, Q_{n-1}\right.\right)$, where

$$p\left(z_n \left| x_n, q_n\right.\right) = \sum_{\alpha \in \mathcal{A}} p\left(a_n = \alpha\right) p\left(z_n \left| a_n = \alpha, x_n, q_n\right.\right)$$

with $p\left(z_n \left| a_n = \alpha, x_n, q_n\right.\right) \propto \omega_n \exp\left(-\left(1/2\right)\omega_n \left|z_n - \exp\left(\mathrm{j}\phi_n\right)\left(\gamma_n \alpha + q_n\right)\right|^2\right)$. Let $f\left(x_n\right) = p\left(z_n \left| x_n, q_n\right.\right)$ and $g\left(x_n\right) = \log\left[f\left(x_n\right)\right] \cong g\left(\acute{x}_n\right) + \left(x_n - \acute{x}_n\right)^{\mathrm{T}} \nabla g\left(\acute{x}_n\right) - \left(1/2\right)\|x_n - \acute{x}_n\|_{\Omega_n}^2$, where Ω_n is the positive semidefinite matrix closest (in spectral norm) to the negative of the Hessian of g evaluated at $\acute{x}_n$. Then $p\left(x_n \left| Z_n, Q_n\right.\right) \overset{\sim}{\propto} \exp\left(-\left(1/2\right)\left[\|x_n - \acute{x}_n\|_{\Omega_n}^2 - 2\left(x_n - \acute{x}_n\right)^{\mathrm{T}} \nabla g\left(\acute{x}_n\right) + \|x_n - \acute{x}_n\|_{\acute{C}_n^{-1}}^2\right]\right) \propto \exp\left(-\left(1/2\right)\|x_n - \hat{x}_n\|_{C_n^{-1}}^2\right)$, $\hat{x}_n = \acute{x}_n + C_n \nabla g\left(\acute{x}_n\right)$, $C_n = \left(\acute{C}_n^{-1} + \Omega_n\right)^{-1}$.

The gradient of g is $\nabla g = \nabla f / f$ and the negative Hessian is $\Upsilon / f + \left(\nabla g\right)\left(\nabla g\right)^{\mathrm{T}}$, where Υ is the negative Hessian of f. With the definitions $\mu\left(\gamma_n, \phi_n; \alpha\right) = \left(1/2\right)\left|\gamma_n \alpha + q_n - \exp\left(-\mathrm{j}\phi_n\right) z_n\right|^2$

and $\psi\left(x_n;\alpha\right) = p\left(a_n = \alpha\right)\exp\left[-\omega_n\mu\left(\gamma_n,\phi_n;\alpha\right)\right]$, we have $f\left(x_n\right) = \sum_{\alpha\in\mathcal{A}}\omega_n\psi\left(x_n;\alpha\right)$. Computation of the required derivatives is straightforward.

To initialize the EOMPs we assume they are constant during the training period. Consequently we have $p\left(x\left|a,z,q\right.\right) \propto \exp\left[f\left(x\right)\right]$, where $f\left(x\right) = N_t\log\left(\omega\right) - \omega\mu\left(\gamma,\phi\right)$ with $\mu\left(\gamma,\phi\right) = \left(1\left/2\right.\right)\left\|a\gamma + q - z\exp\left(-\mathrm{j}\phi_n\right)\right\|^2$, and a, z and q are column vectors of the training symbols, the first N_t equalization filter output samples and the first N_t cancellation filter output samples, respectively. Following a procedure similar to the one discussed above, and assuming $\hat{\phi} \cong 0$, we obtain $\hat{\omega} \cong \left[\left(1\left/2\right.\right)\left\|a\hat{\gamma} + q - z\right\|^2\Big/ N_t\right]^{-1}$, $\hat{\gamma} \cong \mathrm{Re}\left[a^{\mathrm{H}}\left(z - q\right)\right]\big/ N_t$, $\mathrm{Var}\left[\hat{\omega}\right] \cong \hat{\omega}^2\big/ N_t$, $\mathrm{Var}\left[\hat{\gamma}\right] \cong 1\left/\hat{\omega}\right/ N_t$, $\mathrm{Var}\left[\hat{\phi}\right] \cong 1\left/\hat{\omega}\right/\mathrm{Re}\left[\left(a\hat{\gamma} + q\right)^{\mathrm{H}}z\right]$. Instead of obtaining the desired quantities by actually computing the outputs of the equalizer filters, we instead substitute theoretical values. In the absence of channel estimation error we would ideally have $\hat{\gamma} = g_{0,0}$, $\hat{\omega} = \left[\left(1\left/2\right.\right)\mathrm{E}\left|v_n\right|^2\right]^{-1} = 2\left/\left(g_{0,0} - g_{0,0}^2\right)\right.$, and $\mathrm{Re}\left[\left(a\hat{\gamma} + q\right)^{\mathrm{H}}z\right]\Big/ N_t \to \mathrm{E}\left|g_{0,0}a_n + q_n\right|^2 = g_{0,0}^2 + \mathcal{E}_{\bar{a}}\sum_{l\neq 0}\left|g_{l,0}\right|^2$, where we use the notation from the previous appendix. The estimates become unbiased during the training period.

4

Iterative Equalization and Decoding of Underwater Acoustic Communication Data Using Array Observations

We present a receiver algorithm that processes communication data sampled on a hydrophone array. Communication signals are transmitted acoustically from the single transducer of a moving source, and propagate through a shallow underwater channel. The communication signals are created by convolutional coding of information bits, followed by pseudo-random permutation, mapping into QPSK symbols, and linear modulation. Adaptive resampling of received data sequences compensates for Doppler effects caused primarily by source motion. Multichannel equalization focuses and combines signal energy distorted by waveguide multipath propagation. Kalman filtering provides channel impulse response estimates used to construct the equalizer filters. Channel estimation and equalization are performed iteratively in conjunction with decoding, so the receiver implements “turbo equalization”. Communication signals received during an at-sea experiment are combined with ambient noise recordings to give input SNRs of 6 and 7 dB (defined as the SNR at the *output* of an array matched-filter). The results of processing over 50 transmissions using various array configurations show high reliability and low bit error frequency.

4.1 Introduction

In a previous paper [63], we introduced a novel algorithm that recovers information transmitted from a moving source through a shallow underwater acoustic (UWA) channel. That algorithm processed data received on a single hydrophone only. However, using multiple sensors should allow a receiver to operate successfully at significantly lower SNR per sensor than using just a single sensor. This should, in turn, allow a data source, such as an autonomous undersea vehicle (AUV) that has strict power limitations, to transmit signals at lower power. The reason being that a hydrophone array should be able to recombine signal energy that is dispersed by the UWA channel. Actually achieving the anticipated "array gain", however, requires several modifications to the single-sensor algorithm of [63], which is the subject of this paper.

Pulse spreading is the primary feature of the shallow UWA channel [1]. Consequently, equalization is a critical task of high-symbol-rate UWA receivers [27–30]. Iterative equalization and decoding (IED), or "turbo" equalization [15,16], comprises powerful algorithms that process data distorted by delay-spread channels. In particular, methods developed by Tuchler and his colleagues [20–23, 45] are appropriate for channels with long delay spreads. Since the channel impulse responses (CIRs) considered in this paper are on the order of tens of symbol periods, we incorporate these methods into our receiver algorithm. Another key chararcteristic of the UWA channel is acute sensitivity to Doppler effects. Doppler distortion is compensated by resampling the received data sequences. The resampling process and the overall structure of the receiver algorithm are the same as in [63], where the interested reader can find details not repeated in this paper. The present paper focuses on aspects of the new algorithm that enable processing of array data. For other examples of IED applied to UWA channels the reader is referred to [37–42].

Modifications to the single-hydrophone algorithm include the following. The incoherent method used for detection of signal presence (DOSP) in [63] does not work at low SNR, so we use a coherent method instead. Also, the extended Kalman filter (EKF) CIR estimation used in [63] is questionable at low SNR, since it requires linearization of a nonlinear function about very noisy estimates, so we use an augmented Kalman filter that does not require linearization. A method of CIR arrival selection (or "sparsing") used in [63], to reduce the impact of CIR estimation error, is similarly questionable at low SNR, so it is replaced with a more robust method. As with the former algorithm, equalization is performed in two steps: filtering and combining the Doppler-compensated data, followed by estimation of the parameters of the resulting ISI-free output and computation of soft symbol information. Slight modifications are made to the implementation of the first step, while the second step is performed using an altogether different method.

The remainder of this paper is organized as follows. In Section II we summarize the Focused Acoustic Fields 2004 experiment, from which we obtained our data, and our signal and channel models and assumptions. The receiver algorithm is then described in Section III. In Section IV we present results from processing data collected during the experiment, using arrays with five and ten sensors. We conclude the paper in Section V.

4.2 Experiment Setup, Models and Assumptions

4.2.1 Experiment Setup

The experiment was conducted during July 2004 in a shallow water region north of Elba Island, Italy. The experiment geometry is summarized in Fig. 4.1 [63]. Transmissions of duration just over 9 s were made at intervals of 20 s, from a towed acoustic source (simulating an AUV) at a depth of 70 m, through an ocean region of 120 m depth, and received on a moored vertical receive array (VRA). The VRA consisted of 32 sensors with a spacing of 2 m, numbered from the bottom, with sensor #1 at a depth of 104 m and #32 at a depth of 42 m. There are two datasets, denoted "A" and "B". The 24 transmissions of dataset A were recorded while the source was moving directly away from the VRA at a speed of about 4 knots and a distance of about 7 km. The 27 transmissions of dataset B were recorded while the source was moving nearly transverse to the VRA at a speed of about 3.5 knots and a distance of about 11 km. The sound speed environment was downward-refracting, the measured sound speed versus depth can be found in [62].

4.2.2 Encoding, Permuting, and Mapping

The process of transforming information bits into a symbol sequence is summarized in Fig. 4.2 [63]. As shown in part (a), information bits are interleaved with "dummy" bits (i.e., bits known to the receiver), encoded with a convolutional code (CC), pseudo-randomly permuted, then mapped pair-wise into QPSK symbols. Part (b) shows that the resulting sequence of N_d symbols is then interleaved with N_p pilot symbols and prepended with N_t training symbols to create the transmitted sequence of $N_s = N_t + N_p + N_d = 9,006$ symbols. We use the same number of information bits as symbols, so the data rate is equal to the symbol rate. The values of N_t and N_p for the cases of 5 and 10 sensors, along with the single-sensor values from [63], are given in Table 4.1. As explained in [62], though not shown in Fig. 4.2(a), a "scrambling" operation is inserted between the permutation and mapping which has no effect from a statistical standpoint and is easily incorporated in a message-passing algorithm.

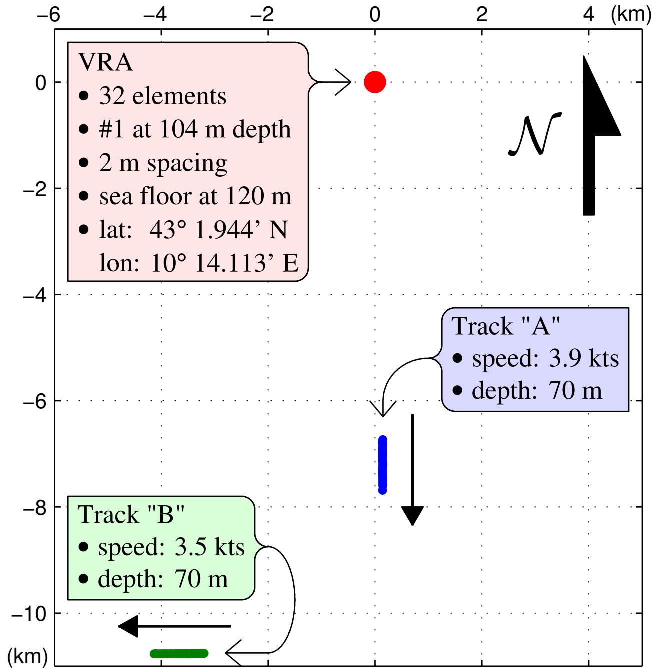


Figure 4.1: Experiment geometry.

Table 4.1: Numbers of Training and Pilot Symbols

N_{sen}	N_t	N_p	% of N_s
1	300	870	13.0 %
5	600	840	16.0 %
10	1000	801	20.0 %

Note: $N_{\text{sen}} = 1$ results reported in [63].

Table 4.2: System Parameters

Sampling Frequency	12 kHz
Center Frequency	3 kHz
Signal Bandwidth	2 kHz
Symbol Rate	1 kHz

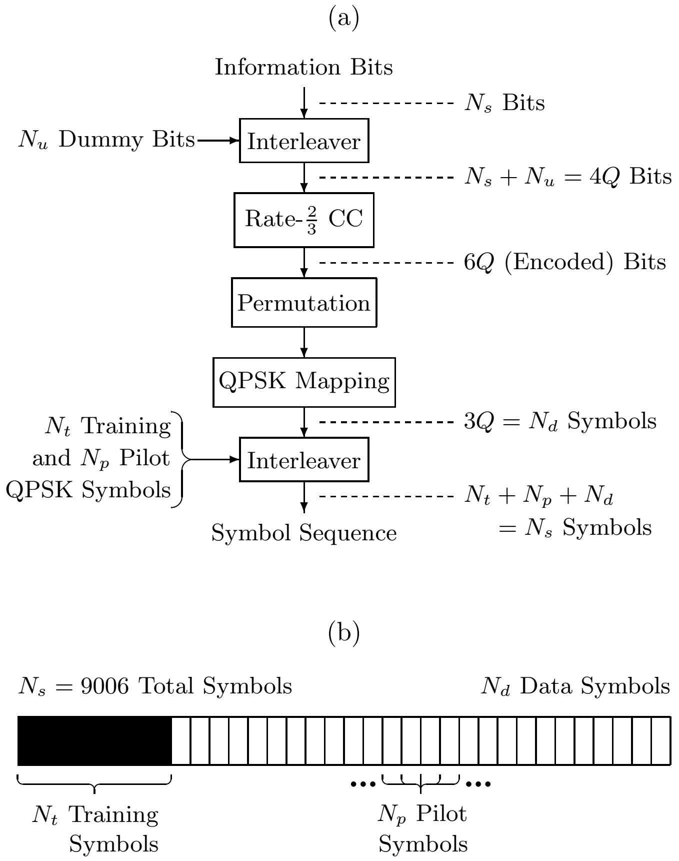


Figure 4.2: (a) Coding Scheme. (b) Symbol Sequence Organization.

4.2.3 Signal Models and Assumptions

Details of the signal and channel models and assumptions can be found in [63] and key system parameters are listed in Table 4.2. The appropriate symbol-rate (SR) model is

$$r_n^{(m)} = w_n^{(m)} + \exp\left(\mathrm{j}\phi_n\right) \sum_l h_{l,n}^{(m)} a_{n-l} \tag{4.1}$$

$$\mathrm{E}\left|w_n^{(m)}\right|^2 = \left(\sigma_w^{(m)}\right)^2 \tag{4.2}$$

where, for the mth "subchannel", $r_n^{(m)}$ is the resampled data, $w_n^{(m)}$ is AWGN, and $h_{l,n}^{(m)}$ is the CIR. As discussed in [62, 63], the phase ϕ_n is a manifestation of error in the resampling process, i.e., error in the estimation of the Doppler expansion/compression. Estimation of this phase is used as an error signal to adapt the resampling rate. There are $M = 2N_{\text{sen}}$ "subchannels", since the signal on each of the N_{sen} receive array sensors is downsampled to the rate of 2 samples per symbol during the resampling process. Since the noise is assumed both spatially and temporally white and the CIR gains $h_{l,n}^{(m)}$ are all assumed independent, there is no mathematical distinction between the two sources of the subchannels. Consequently we use the single index $m \in \{1, \ldots, M\}$ to denote them. As explained in [63], each complex CIR gain is more likely to rotate than change in magnitude. In that paper the CIR gains were modeled in polar form as $h_{l,n}^{(m)} = \rho_{l,n}^{(m)} \exp\left(\mathrm{j}\theta_{l,n}^{(m)}\right)$, and EKF channel estimation was used to constrain the magnitudes and phases to change at different rates. We use a similar procedure in this paper, discussed in Subsection 4.3.4, except that instead of magnitude and phase the CIR gains are expressed as radial and transverse components, which is more stable at low SNR. (Ambiguity between the CIR phases $\theta_{l,n}^{(m)}$ and the phase ϕ_n is resolved during channel estimation.) We also assume the span of the CIRs does not exceed $L^{\max} = 100$ symbols.

In Fig. 3 of [63] we illustrate the temporal variation of a typical CIR on a single sensor. In Fig. 4.3 of this paper we show a typical spatial variation of CIRs on the full receive array. Part (c) shows a snapshot of the CIRs connecting the source to each of the 32 receive array sensors, with delay on the horizontal axis, depth on the vertical axis, and magnitude as gray-scale intensity. The multipath wavefronts are clearly visible, although they do interfere around reference time zero. In part (b) we show the energy associated with each measured CIR. There is a fair amount of variation in signal energy as a function of depth. Part (a) shows the ambient noise variance measured on each sensor, there is also a fair amount of variation with depth.

4.2.4 Input SNR

Communication system performance is bounded by that of the matched-filter receiver under the assumption that a single symbol is transmitted (the "matched-filter bound"). If a single

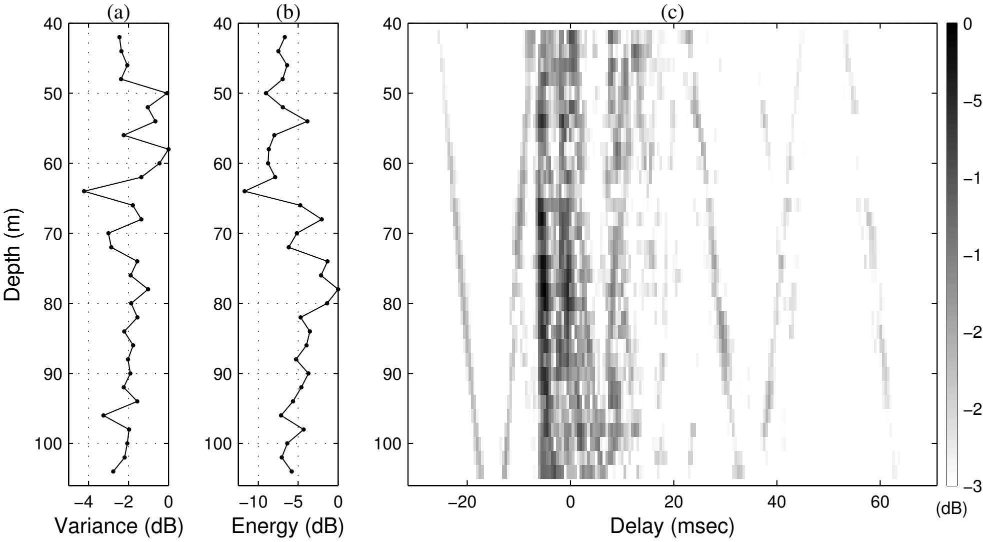


Figure 4.3: (a) Measured ambient noise variance versus depth, normalized so that the noisiest sensor has unit variance. (b) Measured CIR energy versus depth, normalized so that the strongest CIR has unit energy. (c) Magnitude of spatial CIR: horizontal axis is delay, vertical axis is depth, gray-scale intensity is magnitude in dB.

symbol is transmitted, and assuming the CIRs are constant, then the model (4.1) can be written as $r^{(m)} = w^{(m)} + h^{(m)}a$, where the subchannel data, noise and CIRs are vectors, and the symbol a is a scalar. By vertically concatenating the subchannel vectors we have $r = w + ha$, where the noise covariance C_w is block-diagonal, with subchannel covariances $\left(\sigma_w^{(m)}\right)^2 \mathrm{I}$. The matched-filter output SNR is $\mathrm{SNR}_{\mathrm{MF}} = h^{\mathrm{H}} C_w^{-1} h = \sum_m \mathrm{SNR}_{\mathrm{MF}}^{(m)}$, where $\mathrm{SNR}_{\mathrm{MF}}^{(m)} = \left\|h^{(m)}\right\|^2 \Big/ \left(\sigma_w^{(m)}\right)^2$ is the subchannel matched-filter output SNR. We adopt this quantity as our definition of the communication system's *input* SNR, which makes sense for several reasons. First, this definition forces the output SNR of our communication system to be at most equal to the input SNR. Second, the definition reduces to the only meaningful definition of input SNR in the single-channel case, i.e., the ratio of CIR energy to noise variance. Third, we deal with an array of receive sensors which do not necessarily all have the same SNR, so this definition captures the input SNR of the entire array in a scalar quantity. Fourth, as is clear in subsequent discussion, if the receiver algorithm had perfect noise variance and CIR estimates then, assuming successful convergence of the iterative procedure, this SNR would actually be achieved at the system output, making it the natural choice for comparision to output SNR.

Data collected during the experiment that contains communication signals is at very high SNR, so we ignore the ambient noise present in this data. Also collected during the experiment were recordings of ambient noise without any signals present. Using a procedure analogous to the one described in [63], we obtained the data used in this paper by scaling the ambient noise sequences and adding them to the communication sequences to yield an input SNR, as defined previously, of either 6 or 7 dB. Since the subchannel energies $\left\|h^{(m)}\right\|^2$ vary with time, as do the noise variances, the input SNR values should be considered in an average sense, that is, averaged over the duration of a single transmission.

4.2.5 Equalizer Output Model

Linear filtering and combining is used in the equalization process. The filters are constructed from estimates of the CIRs and noise variances. In the absence of channel estimation error, the gain and noise-plus-interference (NPI) variance at the filtering and combining output would be known exactly, as there are closed-form expressions for these parameters. Since this is not the case, we estimate the values of these parameters in conjunction with computing soft symbol information. The appropriate ISI-free model at the filtering and combining output is

$$z_n = v_n + \exp\left(\mathrm{j}\phi_n\right)\gamma_n a_n \tag{4.3}$$

$$\omega_n = 2 \Big/ \mathrm{E}\left|v_n\right|^2 \tag{4.4}$$

where z_n is the output, v_n is NPI (filtered ambient noise, residual ISI, channel estimation error, etc.) with variance $2\,/\omega_n$, and γ_n and ϕ_n are the magnitude and phase, respectively, of the complex gain that multiplies the symbol a_n. The noise v_n is not white, but is modeled as such for tractability. Therefore, the equalizer output model parameters (EOMPs) are the real variables $\omega_n, \gamma_n, \phi_n$.

4.2.6 SMEs and Associated Statistics

As discussed in detail in [62, 63], the soft symbol information we use can be interpreted as estimates of symbol probability mass functions (pmf's). The methods developed by Tuchler et al approximate these symbol pmf's with Gaussian probability density functions (pdf's), thus converting intractable sums that result from application of the sum-product algorithm (SPA) [6] into integrals that can be evaluated in closed-form. The approximation requires symbol mean estimates (SMEs) computed as

$$\bar{a}_n = \sum_{\alpha \in \mathcal{A}} \alpha p\left(a_n = \alpha\right) \tag{4.5}$$

where $\mathcal{A}$ is the QPSK symbol alphabet and $p\left(a_n = \alpha\right)$ is an estimated probability that symbol a_n has value $\alpha \in \mathcal{A}$. Depending upon whether the probabilities are estimates of prior or a posteriori pmf's, they are prior SMEs (PSMEs) or a posteriori SMEs (ASMEs), respectively. The SME variance $\sigma_{\bar{a}}^2 = 1 - \mathcal{E}_{\bar{a}}$, where $\mathcal{E}_{\bar{a}} = \mathrm{E}\left|\bar{a}_n\right|^2$, is the other component needed to approximate symbol pmf's with Gaussian pdf's. It is initially unity, reflecting an initial absence of information from the decoder, then converges to nearly zero as the algorithm converges successfully. Another important statistic is $\chi_{\bar{a}} = \mathrm{E}\left[a_n^* \bar{a}_n\right]$. Since this statistic requires knowledge of the true symbols, it is not used within the algorithm itself. However, we use it as a diagnostic measure of algorithm convergence: $\chi_{\bar{a}} = 0$ implies complete lack of symbol information, while $\chi_{\bar{a}} = 1$ implies total certainty. Ideally we have $\chi_{\bar{a}} = \mathcal{E}_{\bar{a}}$ at all times during the algorithm. The PSME and ASME variance is measured as described in [63].

4.3 Receiver Algorithm

4.3.1 Overview

This paper focuses on equalization, that is, the computation of symbol pmf's that are input to the decoder as part of IED. By "coder"/"decoder" we refer to all the various operations shown in Fig. 4.2(a), i.e., QPSK symbol mapping/de-mapping, permutation, and CC decoding, that transform an information bit sequence into a symbol sequence and vice versa. Application of the SPA to these operations is straightforward and is explained thoroughly in [9,19]. CIR estimation and equalization are performed simultaneoustly during the first iteration, when no information is available from the decoder and thus symbol pmf estimates computed during equalization must be fed-back for CIR estimation, then sequentially during subsequent iterations. Block diagrams of the algorithm are shown in Fig. 4.4 [63]. We now give a brief overview of the algorithm.

We assume asynchronous transmissions, so the receiver's first task is DOSP which, as indicated by $\hat{n}_0$ in part (a) of Fig. 4.4, also provides initial synchronization. Next, the training symbols are used to refine synchronization and initialize the adaptive estimation of Doppler shift (indicated by $\hat{\nu}$) and the CIRs. The following recursion is then performed on a symbol-by-symbol basis: the received data $\tilde{r}_k$ is resampled according to the Doppler shift estimate, the resulting Doppler-compensated data $r_n^{(m)}$ is equalized using multichannel filters constructed from the CIR and noise variance estimates, the equalizer output z_n is used to compute symbol a posteriori probabilities (APPs) $\hat{p}\left(a_n = \alpha \middle| \cdot\right)$ and, as a byproduct, the phase estimate $\hat{\phi}_n$ is used to update the Doppler shift estimate, and the new ASME and ASME variance are fed-back to update the CIR estimates. When APPs are obtained for all the symbols then they are passed to the decoder. Additionally, the sampling times computed during the resampling process are smoothed and a final resampling is performed for use in subsequent iterations. (Although the sampling times could be iteratively refined, in practice the refinements are insignificant and not worthwhile.)

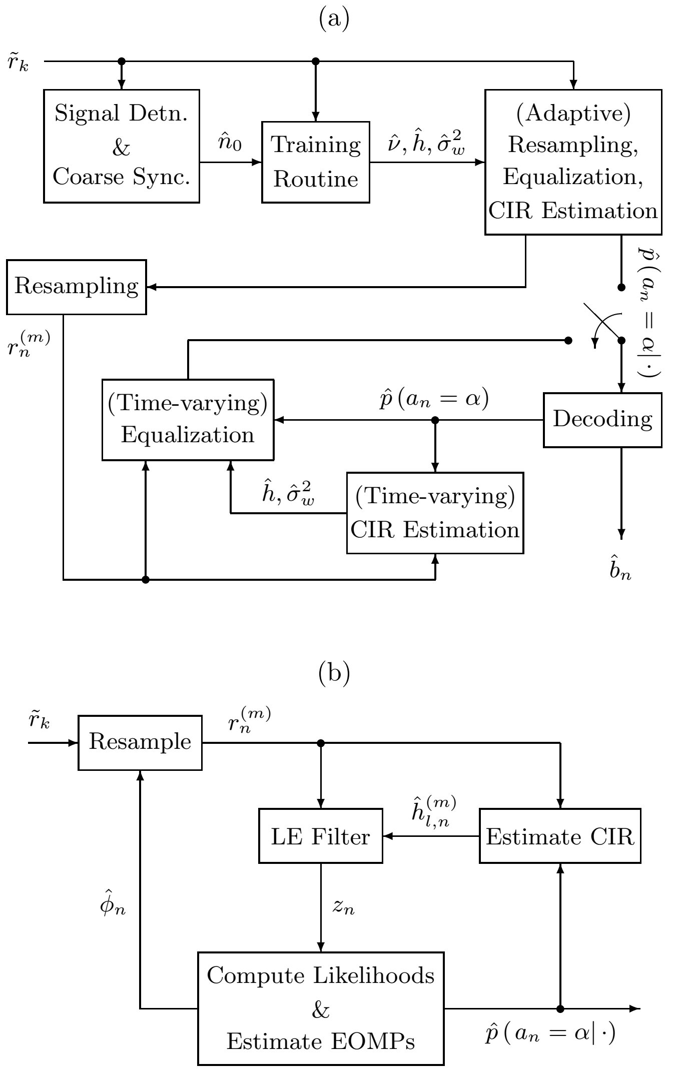

(a)
$\tilde{r}_k$
Signal Detn. & Coarse Sync.
$\hat{n}_0$
Training Routine
$\hat{\nu}, \hat{h}, \hat{\sigma}_w^2$
(Adaptive) Resampling, Equalization, CIR Estimation
$\hat{p}\,(a_n = \alpha|\cdot)$
Resampling
$r_n^{(m)}$
(Time-varying) Equalization
$\hat{p}\,(a_n = \alpha)$
Decoding
$\hat{h}, \hat{\sigma}_w^2$
(Time-varying) CIR Estimation
$\hat{b}_n$
(b)
$\tilde{r}_k$
Resample
$r_n^{(m)}$
LE Filter
$\hat{h}_{l,n}^{(m)}$
Estimate CIR
$\hat{\phi}_n$
z_n
Compute Likelihoods & Estimate EOMPs
$\hat{p}\,(\,a_n = \alpha|\cdot)$

Figure 4.4: (a) Algorithm block diagram. (b) Details of block labeled "(Adaptive) Resampling, Equalization, CIR Estimation" in part (a).

During subsequent iterations the following operations are performed sequentially: the symbol prior pmf's $\hat{p}\left(a_n = \alpha\right)$ supplied by the decoder are used to estimate the CIRs and noise variances, the prior pmf's are then used in conjunction with filters obtained from the channel estimates to equalize the Doppler-compensated data, the prior pmf's are finally used in conjunction with the equalizer output to compute symbol APPs, which are passed to the decoder. On the final iteration, information bit estimates $\hat{b}_n$ are computed instead of symbol prior pmf's.

The resampling process is explained thoroughly in [62], while details of other algorithm components can be found in [63]. The remainder of this section is devoted to describing aspects of the algorithm that are different from [62, 63] or are new.

4.3.2 Detection of Signal Presence

We develop a cost function (CF) with which to perform DOSP. The CF is derived as a binary hypothesis test statistic: either a given segment of data sequence is just noise or it additionally contains the start of a communication signal. The incoherent method used in [63] does not work for multiple sensors at low SNR. Consequently we employ a coherent method which utilizes training symbols and, as such, the CF is a function of Doppler compression/expansion as well as time. For initial synchronization we maximize out the Doppler variable and compare the resulting time series to a detection threshold. We find consecutive crossings of this threshold and search for a peak within the interval. The threshold is obtained by considering the case of false alarm. A byproduct of the procedure is a coarse estimate of Doppler compression/expansion. Both synchronization and Doppler estimation are subsequently refined during the training routine.

In model (4.1) the delay index $l \in \{l_1, \ldots, l_2\}$, where the endpoints l_1 and l_2 vary with each iteration as the algorithm attempts to find the best values for these parameters. For DOSP, we simply set $l_1 = 0$ and $l_2 = L^{\max} - 1$. We assume the Doppler effects and CIRs are essentially constant during that part of the data sequence corresponding to the first $N^{\mathrm{DOSP}} \leq N_t$ training symbols. We then have the following model:

$$\begin{aligned}
r^{(m)}(\nu, n) &= \left[\begin{array}{ccc} r_n^{(m)}(\nu) & \cdots & r_{n+N^{\mathrm{DOSP}}-1}^{(m)}(\nu) \end{array}\right]^{\mathrm{T}} && (4.6)\\
&= w^{(m)} + s^{(m)} && (4.7)\\
s^{(m)} &= A h^{(m)} && (4.8)\\
A &= \left[\begin{array}{ccc} a_0 & & \\ \vdots & \ddots & \\ a_{L^{\max}-1} & \cdots & a_0 \\ \vdots & & \vdots \\ a_{N^{\mathrm{DOSP}}-1} & \cdots & a_{N^{\mathrm{DOSP}}-L^{\max}} \end{array}\right] && (4.9)\\
h^{(m)} &= \left[\begin{array}{ccc} h_0^{(m)} & \cdots & h_{L^{\max}-1}^{(m)} \end{array}\right]^{\mathrm{T}} && (4.10)
\end{aligned}$$

where $w^{(m)}$ is AWGN with variance $\left(\sigma_w^{(m)}\right)^2$ and $s^{(m)}$ is communication signal, which is the product of the symbol convolution matrix A with the CIR $h^{(m)}$. The Doppler parameter ν represents the compression/expansion of the received signal relative to the transmitted signal. For each value of ν the data is resampled in such a manner (refer to [63]) that the samples $r_n^{(m)}(\nu)$ compensate for the Doppler effects represented by ν. Let $r(\nu, n) = \left[r^{(1)}(\nu, n); \ldots; r^{(M)}(\nu, n)\right]$, where semicolons represent vertical concatenation of the constituent vectors, and likewise construct the vectors w, s and h from their subchannel components. Define $A_M = \mathrm{I}_M \otimes A$ (Kronecker product), where I_M is the M-by-M identity matrix. We then have $r(\nu, n) = w + s$, $s = A_M h$. The covariance matrix of w is $C_w = \Sigma_w^2 \otimes \mathrm{I}_{N^{\mathrm{DOSP}}}$, where Σ_w^2 is a diagonal matrix containing the subchannel noise variances.

The pdf of the data conditioned on hypothesis $\mathcal{H}_\eta$ is

$$p(r(\nu, n) | \mathcal{H}_\eta, C_w) = \int p(r(\nu, n) | h, C_w)\, p(h | \mathcal{H}_\eta)\, \mathrm{d}h \qquad (4.11)$$

We will later substitute an estimate for C_w. The null hypothesis $\mathcal{H}_0$ has CIR prior pdf $p(h | \mathcal{H}_0) = \delta(h)$ (Dirac delta function), i.e., the CIR is certain to be zero in this case. The alternative hypothesis $\mathcal{H}_1$ has the CIR prior pdf $p(h | \mathcal{H}_1) = \mathcal{N}_c\left(h; 0, \acute{C}_h\right)$, where the latter notation indicates a circularly-symmetric complex Gaussian pdf with mean 0 and covariance $\acute{C}_h$. The data pdf is $p(r(\nu, n) | h, C_w) = \mathcal{N}_c(r(\nu, n); A_M h, C_w)$, which is identical to $p(r(\nu, n) | \mathcal{H}_0, C_w)$. Since the integrand in the alternative case is equal to $\mathcal{N}_c\left(r(\nu, n); 0, C_w + A_M \acute{C}_h A_M^{\mathrm{H}}\right) \mathcal{N}_c\left(h; \hat{h}(\nu, n), C_h\right)$, where $\hat{h}(\nu, n) = C_h A_M^{\mathrm{H}} C_w^{-1} r(\nu, n)$ and $C_h = \left(\acute{C}_h^{-1} + A_M^{\mathrm{H}} C_w^{-1} A_M\right)^{-1}$, then $p(r(\nu, n) | \mathcal{H}_1, C_w)$ is equal to the first term in this product. We let $\acute{C}_h^{-1} \to 0$, i.e., an uninformative prior. It follows that $\|r(\nu, n)\|^2_{\left(C_w + A_M \acute{C}_h A_M^{\mathrm{H}}\right)^{-1}} \to \|\hat{s}(\nu, n)\|^2_{C_w^{-1}} - \|r(\nu, n)\|^2_{C_w^{-1}}$, where $\hat{s}(\nu, n) = A_M \hat{h}(\nu, n)$. Consequently, to within an additive constant, the log-likelihood ratio (LLR) is $\|\hat{s}(\nu, n)\|^2_{C_w^{-1}} =$

$\sum_{m=1}^{M} \left\| \hat{s}^{(m)}(\nu, n) \right\|^2 \Big/ \left(\sigma_w^{(m)} \right)^2$. Because the noise is white in both space and time, the various estimates decompose into separate subchannel estimates. We have $\hat{h}^{(m)}(\nu, n) = A^{+} r^{(m)}(\nu, n)$, where $A^{+} = \left(A^{\mathrm{H}} A \right)^{-1} A^{\mathrm{H}}$ (Moore-Penrose pseudo-inverse), and $\hat{s}^{(m)}(\nu, n) = A \hat{h}^{(m)}(\nu, n) = P_A r^{(m)}(\nu, n)$, where $P_A = AA^{+}$ is the orthogonal projection matrix onto the signal subspace spanned by the columns of A. Let $\hat{w}^{(m)}(\nu, n) = r^{(m)}(\nu, n) - \hat{s}^{(m)}(\nu, n) = (\mathrm{I} - P_A) r^{(m)}(\nu, n)$ and $\left(\hat{\sigma}_w^{(m)} \right)^2 = \left\| \hat{w}^{(m)}(\nu, n) \right\|^2 \Big/ \left(N^{\mathrm{DOSP}} - L^{\max} \right)$. Note that $\mathrm{E}\left[\left(\hat{\sigma}_w^{(m)} \right)^2 \right] = \left(\sigma_w^{(m)} \right)^2$, so $\left\| \hat{w}^{(m)}(\nu, n) \right\|^2$ is proportional to an unbiased estimate of $\left(\sigma_w^{(m)} \right)^2$. Consequently our CF is

$$J^{\mathrm{DOSP}}(\nu, n) \quad = \quad \sum_{m=1}^{M} \left\| \hat{s}^{(m)}(\nu, n) \right\|^2 \Big/ \left\| \hat{w}^{(m)}(\nu, n) \right\|^2 \tag{4.12}$$

Consider the statistic $v^{\mathrm{H}} r(\nu, n)$. For a given signal vector s, the vector v which maximizes the associated SNR $\left| v^{\mathrm{H}} s \right|^2 \Big/ \mathrm{E} \left| v^{\mathrm{H}} w \right|^2$ is $v = C_w^{-1} s$, and the corresponding statistic is $s^{\mathrm{H}} C_w^{-1} r(\nu, n) = \sum_{m=1}^{M} \left(s^{(m)} \right)^{\mathrm{H}} r^{(m)}(\nu, n) \Big/ \left(\sigma_w^{(m)} \right)^2$. Using estimates in place of the unknown signal vectors and noise variances, and noting that $\left(\hat{s}^{(m)}(\nu, n) \right)^{\mathrm{H}} r^{(m)}(\nu, n) = \left\| \hat{s}^{(m)}(\nu, n) \right\|^2$, this expression becomes proportional to the CF (4.12). Thus the CF can be interpreted as matched-filter detection using estimates in place of unknown quantities. The CF can also be regarded as an estimate, to within a multiplicative factor, of the input SNR as defined in Subsection 4.2.4.

The CF (4.12) is evaluated on a coarse grid of values for ν at synchronization time increments of $L^{\max}/2$ symbol periods. The limits of the grid of values for ν depends upon the anticipated maximum source velocity, but the spacing of the values is $0.5 \left(R_s / f_c \right) / N^{\mathrm{DOSP}}$, which is like Nyquist sampling for Doppler and corresponds to a phase rotation of one-half cycle over a duration of N^{DOSP} symbols. Initially we compute $\max_{\nu} J^{\mathrm{DOSP}}(\nu, n)$ and look for a peak in the resultant time series. Once this initial synchronization is established, we curve-fit the associated function of ν to obtain a coarse Doppler estimate. To implement the procedure, we store the Gram-Schmidt decomposition $A = UT$ at the receiver, where U has orthonormal columns and T is upper triangular. Consequently $\left\| \hat{s}^{(m)}(\nu, n) \right\|^2 = \left\| U^{\mathrm{H}} r^{(m)}(\nu, n) \right\|^2$. (The CIR estimates need not be computed.) Since the CF is evaluated every $L^{\max}/2$ symbols, the computational complexity per value of ν per subchannel per symbol period is $2N^{\mathrm{DOSP}}$. Note that the processing lends itself to massive parallelization. A similar CF based on using correlation-based CIR estimates, implemented via FFTs, may be less computationally expensive at the price of reduced performance, but the issue is not considered in this paper.

To find the peak of $\max_{\nu} J^{\mathrm{DOSP}}(\nu, n)$ we employ a detection threshold. We find consecutive crossings of this threshold and search for the peak within this interval. The threshold is obtained by considering the case of false alarm, i.e., the case where $r^{(m)}(\nu, n)$ is just noise. By the properties of projection matrices, and since the noise is assumed white Gaussian, in this case the squared norms of the signal and noise vector estimates are independent chi-squared random variables (rv's) with $L^{\max}$ and $N^{\text{DOSP}} - L^{\max}$ degrees of freedom, respectively. Their ratios (independent of the noise variances) are independent and identically distributed (iid) rv's with pdf

$$p(x) \quad = \quad \frac{\left(N^{\text{DOSP}}-1\right)!}{\left(L^{\max}-1\right)!\left(N^{\text{DOSP}}-L^{\max}-1\right)!}\frac{x^{L^{\max}-1}}{(1+x)^{N^{\text{DOSP}}}} \tag{4.13}$$

. Unfortunately we know of no closed-form expression for the pdf of the sum of M of these iid rv's and, since the pdf $p\left(x\right)$ does not decay exponentially, Chernoff bound cannot be used. Using the identity $\int_0^\infty x^P \mathrm{d}x \Big/ (1+x)^Q \; = P!\,(Q-P-2)!\,/(Q-1)!$ we obtain the mean $\mu = L^{\max} \big/ \left(N^{\text{DOSP}} - L^{\max} - 1\right)$ and standard deviation

$$\sigma = \sqrt{L^{\max}\left(N^{\text{DOSP}}-1\right)/\left(N^{\text{DOSP}}-L^{\max}-2\right)}\,\Big/\left(N^{\text{DOSP}}-L^{\max}-1\right)$$

Consequently, when only noise is present, the mean and standard deviation of $J^{\text{DOSP}}\left(\nu, n\right)$ are $M\mu$ and $M\sigma$, respectively. We set the threshold at $M\left(\mu + \kappa\sigma\right)$, with $\kappa = 2.5$.

4.3.3 Training Routine

Using the notation from the previous subsection, define the CF

$$J^{\text{Trng}}\left(\nu, n\right) \quad = \quad \sum_{m=1}^{M} \left\| r^{(m)}\left(\check{\nu}, n^*\right)\right\|^2 \Big/ \left\|\hat{w}^{(m)}\left(\nu, n\right)\right\|^2 \tag{4.14}$$

Replacing $\check{\nu}$ with ν and n^* with n would result in $J^{\text{Trng}}\left(\nu, n\right) \; = \; M + J^{\text{DOSP}}\left(\nu, n\right)$, since $\left\|r^{(m)}\left(\nu, n\right)\right\|^2 = \left\|\hat{w}^{(m)}\left(\nu, n\right)\right\|^2 + \left\|\hat{s}^{(m)}\left(\nu, n\right)\right\|^2$. Instead, $\check{\nu}$ is the coarse Doppler estimate obtained in the previous subsection, and n^* is the time index $L^{\max}$ symbols after the corresponding initial synchronization index. In either case the CF (4.14) can be regarded as an affine transformation of an input SNR estimate. The reason for using this CF is that fixing the numerators of the summands eliminates additional sources of variability. During the training routine the CF (4.14) is used with all N_t training symbols vice the N^{DOSP} symbols used for DOSP.

Effective CIR estimation strikes a balance between estimating too many multipath arrivals, thus projecting too much noise onto the estimate, and estimating too few multipath arrivals, thus allowing unaccounted-for signal energy to act as interference. In other words, the typical CIR contains multipath arrivals that, for a given SNR, contain too little energy for their estimation to be worthwhile. In [63] we develop a method that incorporates the set of delays at which CIR gains are estimated as a parameter to be estimated in addition to Doppler and synchronization. In that paper the set of delays is selected during the training routine and held fixed during the first iteration. In the present case, where the SNR on each sensor is anticipated to be much lower than in the single-sensor case, we use an alternative strategy. We estimate the CIRs over the full span of $L^{\max}$ symbols then, prior to using the CIR estimates to compute equalizer filters, we set to zero those CIR gains with too little energy. Consequently, synchronization and Doppler are the two key parameters to be estimated at this point in the algorithm.

The parameter space of synchronization and Doppler is first searched, as part of DOSP, on a coarse grid of values for both parameters using the CF (4.12). This parameter space is next searched using the CF (4.14) on a finer grid of Doppler values with spacing $(15\,/360\,)\,(R_s\,/f_c\,)\,/N_t\,$, which corresponds to 15° phase rotation over the N_t training symbols, and on a slightly less coarse grid of synchronization values with a spacing of $L^{\max}\,/4\,$. We evaluate $\max_n J^{\mathrm{Trng}}\,(\nu, n)$ until we obtain three points with which to fit a parabola. Call the resulting refined Doppler estimate $\hat{\nu}$. At this point we fix the resampling reference point and time reference point as described in [63]. To fine-tune the synchronization, we evaluate $J^{\mathrm{Trng}}\,(\hat{\nu}, n)$ over a finer grid of synchronization values with spacing of $L^{\max}\,/10\,$. We then compute CIR and noise variance estimates to initialize the equalization process.

4.3.4 Channel Estimation and Equalization

There are two main pieces to the equalization process: channel estimation and soft symbol information computation, where the latter utilizes the former. The method of CIR estimation in [63] models each CIR gain in polar form, i.e., magnitude and phase, and linearizes the polar component variables about their current estimates. This linearization is questionable at low SNR, however, so we use a different implementation that is consistent with the channel model. We continue to compute symbol APPs in two steps: filtering and combining the data to remove ISI, followed by estimation of the EOMPs and computation of symbol pmf's therefrom. We briefly comment on the former step, which utilizes a multi-channel linear equalizer and is the natural extension of the method used in [62, 63], and describe a different implementation of the latter step.

CIR Estimation

According to the channel model, the CIR gains are more likely to rotate than change in magnitude. Consequently we model each CIR gain as

$$h_{l,n}^{(m)} = \exp\left(\mathrm{j}\acute{\theta}_{l,n}^{(m)}\right)\left(\acute{\rho}_{l,n}^{(m)} + \Delta\rho_{l,n}^{(m)} + \mathrm{j}\Delta\chi_{l,n}^{(m)}\right) \tag{4.15}$$

where $\acute{h}_{l,n}^{(m)} = \acute{\rho}_{l,n}^{(m)} \exp\left(\mathrm{j}\acute{\theta}_{l,n}^{(m)}\right)$ is the predicted value of the CIR gain (at time n, using data up to time $n-1$), and $\Delta\rho_{l,n}^{(m)}$ and $\Delta\chi_{l,n}^{(m)}$ are the radial and transverse components of the error, respectively. (There are no predictable CIR dynamics, so $\acute{h}_{l,n}^{(m)} = \hat{h}_{l,n-1}^{(m)}$.) This decomposition allows us to enforce the requirement that the CIR gains change in magnitude less than they rotate, but without requiring linearization of the phase term as is done in [63]. Expressing the model (4.1) in vector form, we define $\acute{r}_n^{(m)} = \exp\left(-\mathrm{j}\phi_n\right) r_n^{(m)} - \boldsymbol{a}_n^{\mathrm{T}} \acute{h}_{:,n}^{(m)}$, where $\boldsymbol{a}_n$ is the vector of symbols involved in the convolution sum and $\acute{h}_{:,n}^{(m)}$ (with a colon in place of the delay index) is the corresponding vector of predicted CIR gains. The previous decomposition then gives

$$\acute{r}_n^{(m)} = w_n^{(m)} + \begin{bmatrix} \acute{\boldsymbol{a}}_n \\ \mathrm{j}\acute{\boldsymbol{a}}_n \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \Delta\rho_{:,n}^{(m)} \\ \Delta\chi_{:,n}^{(m)} \end{bmatrix} \tag{4.16}$$

where $\Delta\rho_{:,n}^{(m)}$ and $\Delta\chi_{:,n}^{(m)}$ are the vector components of the CIR error and $\acute{\boldsymbol{a}}_n = \mathrm{D}\left[\exp\left(\mathrm{j}\acute{\theta}_{:,n}^{(m)}\right)\right] \boldsymbol{a}_n$. (We use $\mathrm{D}\left[v\right]$ to denote the diagonal matrix formed from the vector v.) If we define $y_n^{(m)} = \left[\mathrm{Re}\left[\acute{r}_n^{(m)}\right], \mathrm{Im}\left[\acute{r}_n^{(m)}\right]\right]^{\mathrm{T}}$, then we have

$$y_n^{(m)} = \begin{bmatrix} \mathrm{Re}\left[w_n^{(m)}\right] \\ \mathrm{Im}\left[w_n^{(m)}\right] \end{bmatrix} + \begin{bmatrix} \mathrm{Re}\left[\acute{\boldsymbol{a}}_n\right] & \mathrm{Im}\left[\acute{\boldsymbol{a}}_n\right] \\ -\mathrm{Im}\left[\acute{\boldsymbol{a}}_n\right] & \mathrm{Re}\left[\acute{\boldsymbol{a}}_n\right] \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \Delta\rho_{:,n}^{(m)} \\ \Delta\chi_{:,n}^{(m)} \end{bmatrix} \tag{4.17}$$

where the noise vector has covariance $\left[\left(\sigma_w^{(m)}\right)^2/2\right]\mathrm{I}$. Standard Kalman filtering gives the updates $\widehat{\Delta\rho}_{:,n}^{(m)}$ and $\widehat{\Delta\chi}_{:,n}^{(m)}$, from which we form the update $\widehat{\Delta h}_{l,n}^{(m)} = \exp\left(\mathrm{j}\acute{\theta}_{l,n}^{(m)}\right)\left(\widehat{\Delta\rho}_{l,n}^{(m)} + \mathrm{j}\widehat{\Delta\chi}_{l,n}^{(m)}\right)$. The Kalman filter's "prediction" step allows the rates of change of the radial and transverse components to be controlled separately, i.e., by adding different amounts to the diagonals of their respective covariance matrices. Since, for a given (small) amount of rotation, the change in a (complex-valued) CIR gain is proportional to its magnitude, we add $\left(10^{-5}\right)\left|\acute{h}_{l,n}^{(m)}\right|^2$ to the corresponding diagonal entry of the transverse covariance. We also do the same for the radial covariance, but use the smaller value 10^{-6} instead of 10^{-5}. Let $C_n^{(m)}$ denote the CIR covariances, which are the sum of the radial and transverse covariances computed by the Kalman filter. During the first iteration the CIR updates are adjusted as follows:

$$\widehat{\Delta h}_{:,n}^{(m)} \leftarrow \widehat{\Delta h}_{:,n}^{(m)} - \mathrm{j}\mathrm{Im}\left[\lambda_n\right] \acute{h}_{:,n}^{(m)} \tag{4.18}$$

$$\lambda_n = \frac{\sum_m \left(\acute{h}_{:,n}^{(m)}\right)^{\mathrm{H}} \left(C_n^{(m)}\right)^{-1} \widehat{\Delta h}_{:,n}^{(m)}}{\sum_m \left(\acute{h}_{:,n}^{(m)}\right)^{\mathrm{H}} \left(C_n^{(m)}\right)^{-1} \acute{h}_{:,n}^{(m)}} \tag{4.19}$$

As explained in Appendix 4A , this adjustment prevents bulk phase rotation of the CIR estimates. Consequently, potential phase rotation of the CIR gains due to error in the resampling process is absorbed in the phase ϕ_n, which is estimated after the filtering and combining equalization step. The CIR estimates are updated as $\widehat{h}_{l,n}^{(m)} = \left[1 - 0.5\left(\widehat{\Delta\chi}_{l,n}^{(m)}\right)^2 \Big/ \left|\acute{h}_{l,n}^{(m)}\right|^2\right] \acute{h}_{l,n}^{(m)} + \widehat{\Delta h}_{l,n}^{(m)}$, where the scaling of the predicted CIR gains accounts for the fact that the transverse updates do not produce pure rotation of the CIR gains.

In the preceeding formulation, we use SMEs in place of true symbols and use an estimate for the phase ϕ_n. Also, in a manner similar to that employed in [63], we use effective noise variances in place of estimates of $\left(\sigma_w^{(m)}\right)^2$. (The effective noise variances additionally account for uncertainty in the SMEs.) Instead of using exact Kalman filtering, we use and update only the diagonals of the covariance matrices. Let $c_{l,n}^{(m)}$ denote the diagonal entries of the CIR covariance matrices $C_n^{(m)}$. Before using the CIR estimates to compute equalizer filters, we first set to zero those elements with $\left|\widehat{h}_{l,n}^{(m)}\right|^2 \Big/ c_{l,n}^{(m)} > 3$. If $\widehat{h}_{l,n}^{(m)}$ were just AWGN with variance $c_{l,n}^{(m)}$, then the probability of its squared magnitude being greater than $3c_{l,n}^{(m)}$ would be 1.2×10^{-4}. In other words, we zero-out those CIR gains that contain insignificant amounts of signal energy and would only serve to worsen the equalization filters computed from the CIR estimates. The synchronization is re-estimated during each iteration after the first in a manner similar to that employed in [63].

Equalization

The model (4.1) can be expressed as $\boldsymbol{r}_n^{(m)} = \boldsymbol{w}_n^{(m)} + \exp\left(\mathrm{j}\phi_n\right) H_n^{(m)} \boldsymbol{a}_n$, where $\boldsymbol{r}_n^{(m)}$ is a vector of data samples, $\boldsymbol{w}_n^{(m)}$ is a vector of AWGN, $H_n^{(m)}$ is a CIR convolution matrix, and $\boldsymbol{a}_n$ is a vector of symbols which contains the symbol a_n in the middle. The phase ϕ_n is assumed to vary slowly enough that it is essentially constant over the time span of interest. The length of $\boldsymbol{r}_n^{(m)}$ is taken to be $L^{\max}$. By vertically concatenating the subchannel data vectors, noise vectors and CIR convolution matrices we have $\boldsymbol{r}_n = \boldsymbol{w}_n + \exp\left(\mathrm{j}\phi_n\right) H_n \boldsymbol{a}_n$, where $\boldsymbol{w}_n$ has covariance matrix $C_w = \Sigma_w^2 \otimes \mathrm{I}$. (The diagonal matrix Σ_w^2 contains the subchannel noise variances.) Let $\boldsymbol{h}_n$ be the middle column of H_n. The same procedure used in Appendix 2C , there applied to a single (sub)channel, can be used to express the row vector of equalization filters as $\boldsymbol{f}_n = \boldsymbol{h}_n^{\mathrm{H}} Q_n^{-1} \Big/ \left(\boldsymbol{h}_n^{\mathrm{H}} Q_n^{-1} \boldsymbol{h}_n\right)$, where $Q_n = C_w + \sigma_{\bar{\mathrm{a}}}^2 H_n H_n^{\mathrm{H}}$. During the first iteration, when there is no symbol information available from the decoder, the equalizer filtering and combining output is $z_n = \boldsymbol{f}_n \boldsymbol{r}_n$. During subsequent iterations the output is $z_n = \bar{a}_n + \boldsymbol{f}_n\left(\boldsymbol{r}_n - H_n \bar{\boldsymbol{a}}_n\right)$, where $\bar{\boldsymbol{a}}_n$ is a corresponding vector of PSMEs. In either case, the output z_n is modeled according to (4.3). For the first iteration the phase ϕ_n reflects error in the resampling process and is used as a control variable. For subsequent iterations the phase ϕ_n is simply the phase of the complex gain at the equalizer output. We use CIR estimates in place of the unknown true values, and we use effective noise variance estimates in place of the unknown ambient noise variances, which are computed as is done in [63].

Let $x_n = \left[\gamma_n, \phi_n\right]^{\mathrm{T}}$ and $Z_n = \{z_0, \ldots, z_n\}$. Based on the model and assumptions given in Subsection 4.2.5, we have $p\left(\omega_n, x_n \left| Z_n\right.\right) \propto p\left(z_n \left| \omega_n, x_n\right.\right) p\left(\omega_n, x_n \left| Z_{n-1}\right.\right)$, where $p\left(z_n \left| \omega_n, x_n\right.\right) = \sum_{a_n} p\left(a_n\right) p\left(z_n \left| \omega_n, x_n, a_n\right.\right)$ and

$$p\left(z_n \left| \omega_n, x_n, a_n\right.\right) = \frac{1}{\pi}\frac{\omega}{2} \exp\left(-\frac{\omega}{2}\left|z_n - \exp\left(\mathrm{j}\phi_n\right)\gamma_n a_n\right|^2\right) \tag{4.20}$$

In [63] we make an approximation to $p\left(z_n \left|\omega_n, x_n\right.\right)$ such that assuming a Gaussian density for $p\left(\omega_n, x_n \left|Z_{n-1}\right.\right)$ produces a Gaussian density for $p\left(\omega_n, x_n \left|Z_n\right.\right)$. In this paper, also assuming a Gaussian density for $p\left(\omega_n, x_n \left|Z_{n-1}\right.\right)$, a Gaussian approximation for $p\left(z_n, \omega_n, x_n \left|a_n, Z_{n-1}\right.\right) = p\left(z_n \left|\omega_n, x_n, a_n\right.\right) p\left(\omega_n, x_n \left|Z_{n-1}\right.\right)$ is made. The result is a Gaussian mixture representation for $p\left(\omega_n, x_n \left|Z_n\right.\right)$, which we then replace with a single Gaussian density with identical mean and covariance. Additionally, the Gaussian approximation for $p\left(z_n, \omega_n, x_n \left|a_n, Z_{n-1}\right.\right)$ is used to compute the symbol APP $\hat{p}\left(a_n = \alpha \right| \cdot) \propto p\left(z_n \left|a_n, Z_{n-1}\right.\right)$ by marginalizing-out the EOMPs, which is easy because the density is Gaussian. The details are provided in Appendix 4B .

4.4 Results and Discussion

There were 32 sensors in the full receive array used to collect data during the experiment. For the results presented in this paper, however, we processed data from smaller arrays of five and ten consecutive sensors, located at the top, middle, and bottom of the full receive array. The configurations of these arrays are listed in Table 4.3. Since there were 51 total transmissions, we obtained results for 153 trials for each array size. For the ten-sensor arrays, there were zero bit errors in all 153 trials at an input SNR of 7 dB (recall, the system input SNR is defined in Subsection 4.2.4 as the array matched-filter *output* SNR). At an input SNR of 6 dB, on the other hand, many trials had insufficient SNR at the equalizer output and were unable to recover the information bits with few errors. Consequently, based on this limited number of trials, we conclude that the algorithm's lower limit of operation is at an input SNR of just below 7 dB for an array size of ten sensors.

Table 4.3: Array Configurations

	Number of Sensors in Array	
	5	10
Top	Sensors: #28 - 32 Depths: 50 - 42 m	Sensors: #23 - 32 Depths: 60 - 42 m
Mid.	Sensors: #15 - 19 Depths: 76 - 68 m	Sensors: #12 - 21 Depths: 82 - 64 m
Bot.	Sensors: #1 - 5 Depths: 104 - 96 m	Sensors: #1 - 10 Depths: 104 - 86 m

For the five-sensor arrays, there were zero bit errors in all 153 trials at an input SNR of 7 dB. At an input SNR of 6 dB, while all 153 trials were processed successfully by the algorithm, there were bit errors for some of the trials. For the 72 trials in dataset A, there were 71 total bit errors and an average bit error frequency of 1.1×10^{-4}. For the 81 trials in dataset B, there were

12 total bit errors and an average bit error frequency of 1.6×10^{-5}. Fig. 4.5 shows details of the processing results for data set A (note that there is no data corresponding to time 7:40), and Fig. 4.6 shows the same processing details for data set B. Part (a) of the figures show the number of bit errors produced by the algorithm marked in text, with the vertical axis indicating array location (i.e., top, middle, bottom) and the horizontal axis indicating the start time of the transmission. There is no apparent pattern to the distribution of errors and the maximum number of errors in any transmission is nine–or about 1 error per 1000 bits. Part (b) of the figure shows the average Doppler shift measured by the algorithm, each of the 3 array locations corresponds to a dot which is plotted against the transmission start time. For dataset B the Doppler shift increases in magnitude with time and is consistent with the source track of Fig. 4.1, which shows that the radial component of the source velocity increases with time. For dataset A the Doppler shift is relatively constant, which is also consistent with its source track. The algorithm's lower limit of operation appears to be at an input SNR of nearly 6 dB for an array size of five sensors.

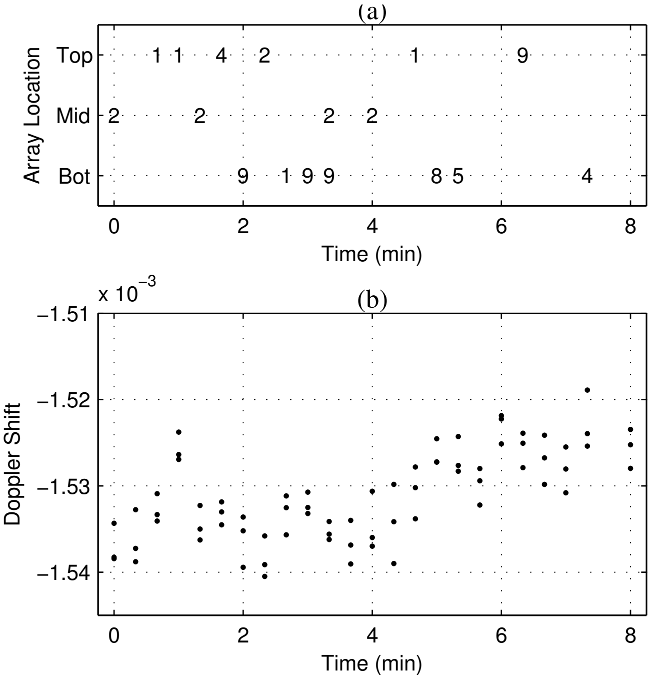


Figure 4.5: Data set A. (a) Number of bit errors marked with text versus array location on the vertical axis and transmission start time on the horizontal axis. (b) Average Doppler shift measured by the algorithm: each dot corresponds to one of the 3 array locations, horizontal axis is transmission start time.

We now discuss the processing of one particular trial, using data from the top five sensors of the full receive array, starting at the 2-minute mark of dataset A. The sensor depths

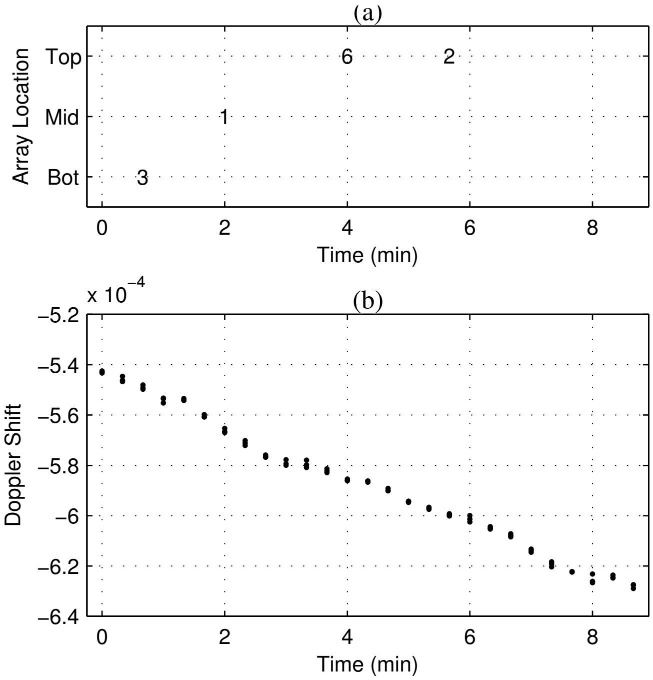


Figure 4.6: Data set B. (a) Number of bit errors marked with text versus array location on the vertical axis and transmission start time on the horizontal axis. (b) Average Doppler shift measured by the algorithm: each dot corresponds to one of the 3 array locations, horizontal axis is transmission start time.

Table 4.4: SNR on Array used in Example

Sensor #	28	29	30	31	32
Depth (m)	50	48	46	44	42
SNR (dB)	-4.7	-0.3	-0.6	-1.0	0.2

and measured SNR on each sensor is listed in Table 4.4, the input SNR is 6 dB. The CIRs on the full receive array at the start of this transmission are the ones shown in Fig. 4.3, where it can be seen that most of the CIR energy is concentrated in a span of about 25 symbol periods. Fig. 4.7 shows the equalizer performance for the first (plotted in gray) and last (plotted in black) iterations of the algorithm. Parts (a), (b), and (c) show the estimated EOMPs, there is clear improvement in the quality of the equalizer output: both the inverse noise variance and gain magnitude increase significantly, and the phase variation decreases markedly. Parts (d) and (e) show the magnitude and phase of the moving-average of the ASME statistic $a_n^* \bar{a}_n$ (computed using only data-bearing symbols). Again the improvement in equalization from the first iteration to the last is clear: the magnitude of the statistic increases from an average of about 0.5 to about 0.9 and the phase fluctuations decrease from more than 10 degrees to just a couple degrees.

A summary of the performance of the algorithm for the particular trial is shown in Fig. 4.8. Part (a) shows the decrease in CIR estimation error as the algorithm iterates: we plot the energy in the difference between the algorithm's CIR estimates and the measured CIRs (using the true symbols and noiseless data), normalized by the energy of the measured CIRs. Using the true symbols we made rectangular-window least-squares estimates of the complex gain $\gamma_n \exp(\mathrm{j}\phi_n)$ and effective noise variance $\mathrm{E}\,|v_n|^2$ of the model (4.3). The squared magnitude of the former divided by the latter is a measure of the output signal-to-interference-plus-noise-ratio (SINR), and the mean of this quantity is shown for each iteration in part (b) of the figure. The SINR steadily increases during the first five iterations, then plateaus at a value just below the input SNR. Part (c) shows the average of the absolute value of the phase of the gain estimates and it too demonstrates steady improvement as a function of iteration. Part (d) of the figure shows the number of bit errors that would result if the algorithm were terminated at the end of each iteration, and part (e) shows the PSME variance for each iteration. Overall the algorithm steadily improves knowledge of the bits, symbols, and channel model parameters until reaching a steady state.

4.5 Conclusion

In this paper we presented an IED algorithm that processes multisensor data received from a moving UWA source, and applied the algorithm to data obtained during an at-sea experiment. The same data was processed by a single-sensor IED algorithm introduced in [63]. Although that algorithm shares the same overall structure as the algorithm presented in this paper, the multisensor algorithm possesses several modifications. The method of DOSP used in [63], for example, fails at low SNR, so a stronger method was developed in this paper. Sim-

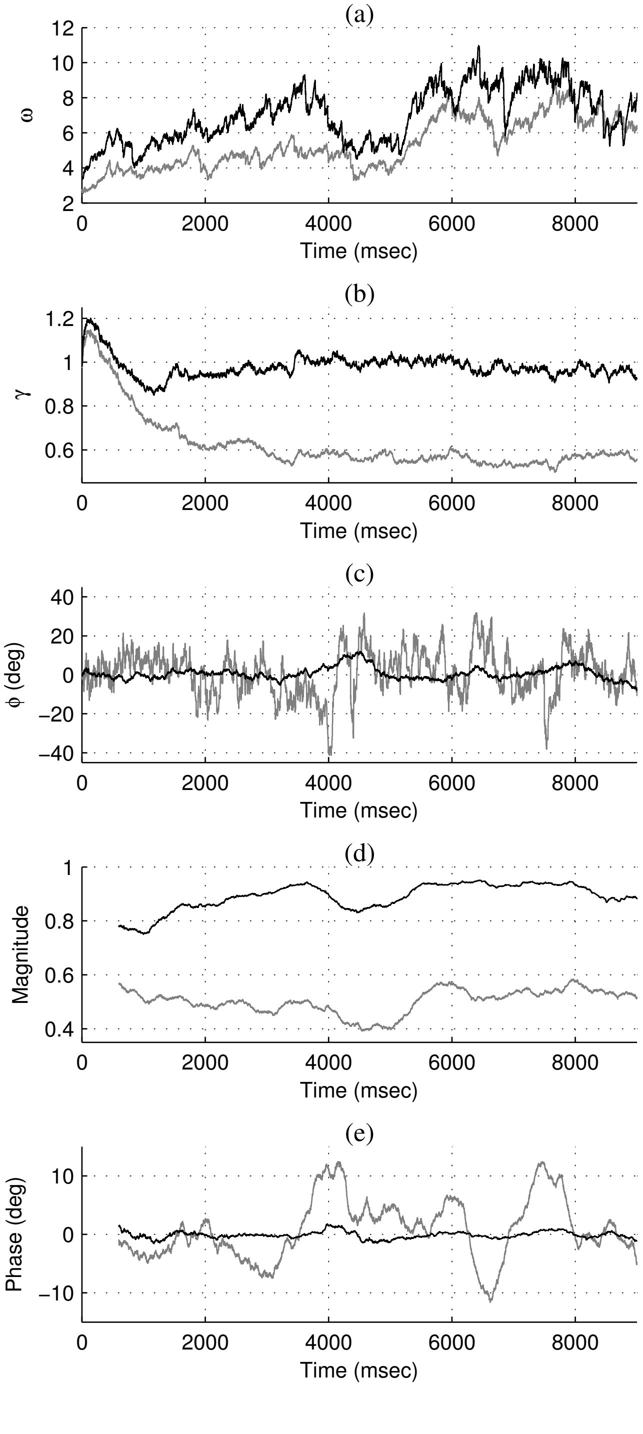
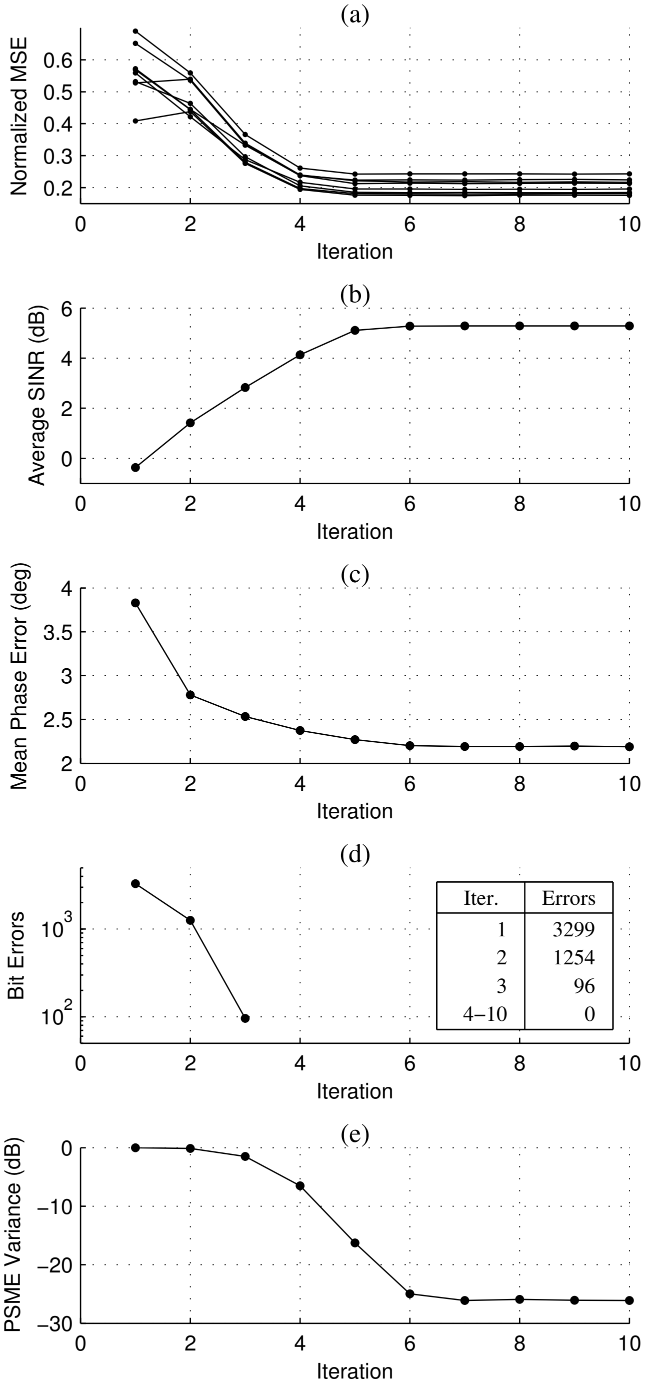


Figure 4.7: (a,b,c) EOMP estimates. (d,e) Magnitude and phase of moving-average of symbol-quality statistic. In all cases, gray curves correspond to the first iteration, black curves to the last iteration.



Iter.	Errors
1	3299
2	1254
3	96
4−10	0

Figure 4.8: (a) Normalized CIR error energy versus iteration. (b) Average SINR at equalizer output versus iteration. (c) Mean absolute phase error versus iteration. (d) Number of bit errors versus iteration. (e) PSME variance versus iteration.

ilarly, the method of CIR estimation used in [63] breaks down at low SNR, so a more robust method is employed in the present algorithm. The equalization process has also been modified, including an original method of symbol APP computation that is a closer approximation to the SPA.

The algorithm was used to process over 50 transmissions for various array configurations at various SNRs, and performed with a high degree of success. The algorithm achieved a bit error frequency of nearly 1.0×10^{-4} using five-sensor arrays at an input (i.e., matched-filter output) SNR of 6 dB, and had zero errors at 7 dB using both five- and ten-sensor arrays. Several performance plots detail the processing of one particular transmission. They clearly demonstrate the iterative improvement in knowledge of the transmitted bits and symbols and all the various model parameters needed to implement the equalization process. For the given example, the algorithm displayed steady improvement with each iteration. The lower limit of operation for the single-sensor algorithm of [63] appeared to be at an input SNR of about 7 dB, which is about the same as the lower limit of operation of the present algorithm using ten-sensor arrays. For five-sensor arrays, on the other hand, the lower limit of operation appears to be at an input SNR of about 6 dB. We conclude that the detriment of increased error in the CIR estimates, which results from operating at lower SNR per sensor, is compensated by the benefit of spatial diversity provided by using multiple sensors. In fact, for the five-sensor arrays, spatial diversity more than compensated for the increased error in the CIR estimates.

4.6 Acknowledgments

This work was supported by the Office of Naval Research under Grant N00014-05-1-0263.

Chapter 4, in full, is a reprint of the material as it appears in "Iterative Equalization and Decoding of Underwater Acoustic Communication Data Using Array Observations", J. F. Sifferlen, H. C. Song, W. S. Hodgkiss, W. A. Kuperman, and J. M. Stevenson, *IEEE Journal of Oceanic Engineering*, submitted 22 October 2008. The dissertation author was the primary investigator and author of this paper.

4A Preventing Bulk Phase Rotation of CIR Estimates

Using the notation of Subsection 4.3.4, let $\acute{\boldsymbol{h}}_n$, $\widehat{\Delta\boldsymbol{h}}_n$, and $\hat{\boldsymbol{h}}_n = \acute{\boldsymbol{h}}_n + \widehat{\Delta\boldsymbol{h}}_n$ be the vectors formed by vertically concatenating the respective subchannel vectors, and let C_n be the block-diagonal matrix formed from the subchannel CIR covariances. We define the phase rotation of $\hat{\boldsymbol{h}}_n$

relative to $\acute{\boldsymbol{h}}_n$, under the weighting C_n^{-1}, as $\widehat{\Delta\phi}_n = \arg\min_{\Delta\phi} \left\| \hat{\boldsymbol{h}}_n - \exp\left(\mathrm{j}\Delta\phi\right) \acute{\boldsymbol{h}}_n \right\|_{C_n^{-1}}^2$, where $\|v\|_M^2$ denotes the quadratic form $v^{\mathrm{H}} M v$. Consequently, $\exp\left(\mathrm{j}\widehat{\Delta\phi}_n\right) = \acute{\boldsymbol{h}}_n^{\mathrm{H}} C_n^{-1} \hat{\boldsymbol{h}}_n \Big/ \left| \acute{\boldsymbol{h}}_n^{\mathrm{H}} C_n^{-1} \hat{\boldsymbol{h}}_n \right|$, which implies that to have zero phase rotation the weighted inner-product $\acute{\boldsymbol{h}}_n^{\mathrm{H}} C_n^{-1} \hat{\boldsymbol{h}}_n$ should be a real number. Let $\overline{\Delta\boldsymbol{h}}_n = \widehat{\Delta\boldsymbol{h}}_n - \mathrm{j}\mathrm{Im}\left[\lambda_n\right] \acute{\boldsymbol{h}}_n$, $\overline{\boldsymbol{h}}_n = \acute{\boldsymbol{h}}_n + \overline{\Delta\boldsymbol{h}}_n$, and $\lambda_n = \acute{\boldsymbol{h}}_n^{\mathrm{H}} C_n^{-1} \widehat{\Delta\boldsymbol{h}}_n \Big/ \left\| \acute{\boldsymbol{h}}_n \right\|_{C_n^{-1}}^2$. Then $\acute{\boldsymbol{h}}_n^{\mathrm{H}} C_n^{-1} \overline{\boldsymbol{h}}_n = \left\| \acute{\boldsymbol{h}}_n \right\|_{C_n^{-1}}^2 + \mathrm{Re}\left[\acute{\boldsymbol{h}}_n^{\mathrm{H}} C_n^{-1} \widehat{\Delta\boldsymbol{h}}_n \right]$ is real and so $\overline{\boldsymbol{h}}_n$ has zero rotation relative to the predicted estimate $\acute{\boldsymbol{h}}_n$. Note that $\hat{\boldsymbol{h}}_n - \overline{\boldsymbol{h}}_n = \widehat{\Delta\boldsymbol{h}}_n - \overline{\Delta\boldsymbol{h}}_n = \mathrm{j}\mathrm{Im}\left[\lambda_n\right] \acute{\boldsymbol{h}}_n$. For all $\boldsymbol{h}$ such that $\acute{\boldsymbol{h}}_n^{\mathrm{H}} C_n^{-1} \boldsymbol{h}$ is real, $\left(\hat{\boldsymbol{h}}_n - \overline{\boldsymbol{h}}_n \right)^{\mathrm{H}} C_n^{-1} \left(\overline{\boldsymbol{h}}_n - \boldsymbol{h} \right) = -\mathrm{j}\mathrm{Im}\left[\lambda_n\right] \left(\acute{\boldsymbol{h}}_n^{\mathrm{H}} C_n^{-1} \overline{\boldsymbol{h}}_n - \acute{\boldsymbol{h}}_n^{\mathrm{H}} C_n^{-1} \boldsymbol{h} \right)$ is imaginary, so $\left\| \hat{\boldsymbol{h}}_n - \boldsymbol{h} \right\|^2 = \left\| \hat{\boldsymbol{h}}_n - \overline{\boldsymbol{h}}_n \right\|^2 + \left\| \overline{\boldsymbol{h}}_n - \boldsymbol{h} \right\|^2$. Therefore, $\overline{\boldsymbol{h}}_n$ is the vector closest to $\hat{\boldsymbol{h}}_n$, in the weighted LS sense, that has zero rotation relative to $\acute{\boldsymbol{h}}_n$, and so we use it in place of $\hat{\boldsymbol{h}}_n$ for the CIR update.

4B Estimation of EOMPs and Computation of Symbol APPs

We use the notation introduced in Subsection 4.3.4. Let $\mathcal{N}\left(x; \bar{x}, C\right)$ denote the real Gaussian pdf in the variable x with mean $\bar{x}$ and covariance C. Making the linear approximation $\exp\left(\mathrm{j}\phi_n\right)\gamma_n \cong \exp\left(\mathrm{j}\acute{\phi}_n\right)\left(\acute{\gamma}_n + \Delta\gamma_n + \mathrm{j}\acute{\gamma}_n\Delta\phi_n\right)$, where $\acute{\phi}_n$ and $\acute{\gamma}_n$ are predicted estimates, and letting $\acute{z}_n = \exp\left(-\mathrm{j}\acute{\phi}_n\right) z_n - \acute{\gamma}_n a_n$ and $y_n = \left[\mathrm{Re}\left[\acute{z}_n\right], \mathrm{Im}\left[\acute{z}_n\right]\right]^{\mathrm{T}}$ gives $p\left(z_n \left| \omega_n, x_n, a_n \right.\right) \cong \mathcal{N}\left(y_n; B_n^{\mathrm{T}}\Delta x_n, \mathrm{I}/\omega_n\right)$, where $\Delta x_n = \left[\Delta\gamma_n, \Delta\phi_n\right]^{\mathrm{T}}$ and

$$B_n = \begin{bmatrix} \mathrm{Re}\left[a_n\right] & \mathrm{Im}\left[a_n\right] \\ -\acute{\gamma}_n\mathrm{Im}\left[a_n\right] & \acute{\gamma}_n\mathrm{Re}\left[a_n\right] \end{bmatrix} \tag{4.21}$$

Note that, although not explicitly denoted, y_n and B_n are functions of a_n. Let $\acute{\omega}_n$ be the predicted value of ω_n and $\Delta\omega_n = \omega_n - \acute{\omega}_n$. We assume a Gaussian prior for $p\left(\omega_n, x_n \left| Z_{n-1} \right.\right)$, which can be expressed as $\mathcal{N}\left(\Delta\omega_n; 0, \acute{C}_n^{\omega}\right)\mathcal{N}\left(\Delta x_n; \acute{\psi}_n\Delta\omega_n, \acute{C}_n^{x|\omega}\right)$, where $\acute{\psi}_n = \acute{C}_n^{x\omega} \Big/ \acute{C}_n^{\omega}$ and $\acute{C}_n^{x|\omega} = \acute{C}_n^{x} - \acute{C}_n^{x\omega}\left(\acute{C}_n^{x\omega}\right)^{\mathrm{T}} \Big/ \acute{C}_n^{\omega}$. Consequently,

$$\begin{aligned} p\left(z_n, \omega_n, x_n \left| a_n, Z_{n-1} \right.\right) &\cong \mathcal{N}\left(\Delta\omega_n; 0, \acute{C}_n^{\omega}\right)\mathcal{N}\left(y_n; B_n^{\mathrm{T}}\acute{\psi}_n\Delta\omega_n, \Upsilon_n\left(\Delta\omega_n\right)\right) \\ &\quad \times\mathcal{N}\left(\Delta x_n; \widehat{\Delta x}_n\left(\Delta\omega_n; a_n\right), C_n^{x|\omega}\left(\Delta\omega_n; a_n\right)\right) && (4.22) \\ \Upsilon_n\left(\Delta\omega_n\right) &= \mathrm{I}/(\acute{\omega}_n + \Delta\omega_n) + B_n^{\mathrm{T}}\acute{C}_n^{x|\omega}B_n && (4.23) \\ \widehat{\Delta x}_n\left(\Delta\omega_n; a_n\right) &= \acute{\psi}_n\Delta\omega_n + \acute{C}_n^{x|\omega}B_n\left[\Upsilon_n\left(\Delta\omega_n\right)\right]^{-1}\left(y_n - B_n^{\mathrm{T}}\acute{\psi}_n\Delta\omega_n\right) && (4.24) \\ C_n^{x|\omega}\left(\Delta\omega_n; a_n\right) &= \acute{C}_n^{x|\omega} - \acute{C}_n^{x|\omega}B_n\left[\Upsilon_n\left(\Delta\omega_n\right)\right]^{-1}B_n^{\mathrm{T}}\acute{C}_n^{x|\omega} && (4.25) \end{aligned}$$

We say $\mathcal{N}\left(\Delta\omega_n; 0, \acute{C}_n^{\omega}\right)\mathcal{N}\left(y_n; B_n^{\mathrm{T}}\acute{\psi}_n\Delta\omega_n, \Upsilon_n\left(\Delta\omega_n\right)\right) \cong \zeta_n\left(a_n\right)\mathcal{N}\left(\Delta\omega_n; \widehat{\Delta\omega}_n\left(a_n\right), C_n^{\omega}\left(a_n\right)\right)$ for each $a_n \in \mathcal{A}$. (We do this by fitting a parabola to the logarithm of the former product, and setting $\zeta_n\left(a_n\right)=0$ if $\acute{\omega}_n+\widehat{\Delta\omega}_n\left(\alpha\right)<0$.) The term $\zeta_n\left(a_n\right)$ approximates the symbol likelihood $p\left(z_n\left|a_n, Z_{n-1}\right.\right)$, so we scale it to have unit sum and use it as soft symbol information. Let $C_n^{x|\omega}\left(a_n\right)=C_n^{x|\omega}\left(\widehat{\Delta\omega}_n\left(a_n\right);a_n\right)$ and $\widehat{\Delta x}_n\left(a_n\right)=\widehat{\Delta x}_n\left(\widehat{\Delta\omega}_n\left(a_n\right);a_n\right)$. Then

$$\begin{aligned}\widehat{\Delta x}_n\left(\Delta\omega_n;a_n\right) &\cong \widehat{\Delta x}_n\left(a_n\right)+\psi_n\left(a_n\right)\left(\Delta\omega_n-\widehat{\Delta\omega}_n\left(a_n\right)\right) &(4.26)\\ \psi_n\left(a_n\right) &= \acute{\psi}_n-\acute{C}_n^{x|\omega}B_n\left[\Upsilon_n\left(\widehat{\Delta\omega}_n\left(a_n\right)\right)\right]^{-1}B_n^{\mathrm{T}}\acute{\psi}_n &(4.27)\end{aligned}$$

Therefore, letting $\Delta X_n^{\mathrm{T}}=\left[\Delta\omega_n,\Delta x_n^{\mathrm{T}}\right]$, $p\left(\omega_n,x_n\left|a_n,Z_n\right.\right)\cong\mathcal{N}\left(\Delta X_n;\widehat{\Delta X}_n\left(a_n\right),C_n^X\left(a_n\right)\right)$, $\left(\widehat{\Delta X}_n\left(a_n\right)\right)^{\mathrm{T}}=\left[\widehat{\Delta\omega}_n\left(a_n\right),\left(\widehat{\Delta x}_n\left(a_n\right)\right)^{\mathrm{T}}\right]$, $C_n^{x\omega}\left(a_n\right)=\psi_n\left(a_n\right)C_n^{\omega}\left(a_n\right)$, $C_n^{x}\left(a_n\right)=C_n^{x|\omega}\left(a_n\right)+C_n^{x\omega}\left(a_n\right)\left(C_n^{x\omega}\left(a_n\right)\right)^{\mathrm{T}}/C_n^{\omega}\left(a_n\right)$, and

$$C_n^X\left(a_n\right)=\begin{bmatrix}C_n^{\omega}\left(a_n\right) & \left(C_n^{x\omega}\left(a_n\right)\right)^{\mathrm{T}}\\ C_n^{x\omega}\left(a_n\right) & C_n^{x}\left(a_n\right)\end{bmatrix}\qquad(4.28)$$

Also, given the symbol prior pmf $p\left(a_n\right)$, $p\left(a_n\left|Z_n\right.\right)\cong p\left(a_n\right)\zeta_n\left(a_n\right)/\left[\sum_{\alpha}p\left(a_n=\alpha\right)\zeta_n\left(\alpha\right)\right]$. Consequently, the updated EOMPs' pdf is $p\left(\omega_n,x_n\left|Z_n\right.\right)\cong\mathcal{N}\left(\Delta X_n;\widehat{\Delta X}_n,C_n^X\right)$, where

$$\begin{aligned}\widehat{\Delta X}_n &= \sum_{\alpha}p\left(a_n\left|Z_n\right.\right)\widehat{\Delta X}_n\left(a_n\right) &(4.29)\\ C_n^X &= \sum_{\alpha}p\left(a_n\left|Z_n\right.\right)\left[C_n^X\left(a_n\right)+\left(\widehat{\Delta X}_n-\widehat{\Delta X}_n\left(a_n\right)\right)\left(\widehat{\Delta X}_n-\widehat{\Delta X}_n\left(a_n\right)\right)^{\mathrm{T}}\right] &(4.30)\end{aligned}$$

For the "prediction" step of this EKF-like algorithm we add $\left(10^{-4}\right)\hat{\omega}_{n-1}^2$, $\left(10^{-5}\right)\hat{\gamma}_{n-1}^2$, and 10^{-3} to the respective diagonal entries of the covariance matrix. We assume the EOMPs are constant over a short span of training symbols and initialize the EOMPs with LS estimates as is done in [63].

5

Conclusion and Future Research

5.1 Conclusion

This dissertation has detailed the author's research efforts in the area of IED applied to UWA communication – specifically, to data corresponding to unknown, time-varying, severely delay-spread channels. The first chapter provided necessary background material about shallow UWA channels, equalization, and iterative detection. The remaining chapters presented a succession of IED algorithms, with each algorithm other than the initial one extending its predecessor. Each algorithm was developed with the mindset that an actual prototype system was being designed. The corresponding chapters all present performance results for the respective algorithms using data obtained during at-sea experiments. It was endeavored to produce receiver algorithms that can operate at low SNR. The initial algorithm (Chapter 2) processes data produced by a single, fixed source transducer and received on a single, fixed receive hydrophone. The data model (1.3) is used, but the CIRs are assumed time-invariant after proper nonuniform resampling. The next algorithm (Chapter 3) extends the initial one by incorporating source motion. Even with the use of nonuniform resampling to compensate for the consequent Doppler compression/expansion, the resulting CIRs vary significantly during the course of a transmission. This necessitated a very different approach to equalization and, of course, channel estimation. The final algorithm (Chapter 4) extends its antecedent by using data received on an array of sensors. The use of array data allows the receiver algorithm to operate at lower SNR per sensor than when using only a single sensor. This required modifications to be made to many receiver tasks to allow for the lower sensor SNRs.

The major contributions made in this dissertation are summarized as follows:

- The focal point of the research was data obtained from at-sea experiments.

- Extensive measuring and modeling of the shallow UWA channel was performed. Since the performance of a receiver algorithm that lacks the benefit of a priori channel knowledge is ultimately limited by the projection of noise onto the subspace of CMPs, obtaining a channel model that accurately describes the data with as few degrees of freedom as possible is crucially important. The models used in this dissertation, and which drove the design of the receiver algorithms, were developed with this in mind.

- Successful receiver algorithms for unknown, time-varying, severely delay-spread UWA channels were designed, implemented, and evaluated using data obtained from at-sea experiments. The algorithms were comprehensive and incorporated all necessary receiver tasks. While this may sound unremarkable, it is common practice to assume that synchronization, resampling, channel estimation, or other receiver functions have been performed before the analysis begins. Furthermore, it is also common to assume, for example, that channel or symbol estimates are without error and to use off-line channel measurements or the actual symbols when evaluating receiver performance.

- Graphical modeling and iterative message-passing were applied. Iterative processing is an emerging paradigm in digital communication which encompasses the most powerful detection and decoding algorithms known. The application of these concepts and methods in the receiver algorithms developed in this dissertation represents an effort to bring to bear the best known techniques of information recovery.

- A method of CIR "framing" was developed. Framing refers to simultaneous synchronization and CIR estimation: it incorporates not only the estimation of the gains of CIR multipath arrivals, but also the determination of which arrivals possess significant energy such that their gains should be estimated. It addresses the tradeoff between estimating the gains of too many multipath arrivals, thus projecting too much noise onto the CIR estimates, and estimating the gains of too few arrivals, thus losing signal energy and allowing the lost energy to act as interference. A cost function was introduced which quantifies this tradeoff.

- An EKF was derived for CIR estimation. Having observed that the gains of CIR multipath arrivals typically vary less in magnitude than in phase rotation, consistent with the underlying model of Doppler spread among the multipaths, the CIR gains were modeled in polar form. Expressing the gains in this manner enables independent control of the variation allowed on their radial and transverse components. Consequently, the CIR gains can be forced to change in magnitude less than they rotate, thus eliminating unnecessary degrees of freedom and, ultimately, yielding improved performance of the overall algorithms.

- The ambiguity between phase rotation caused by Doppler shift estimation error and phase rotation of the time-varying complex-valued CIR gains was resolved. Recall that Doppler shift estimation error manifests itself by phase-rotating and delay-shifting the CIRs. Consequently, if the CIR gains are allowed to phase rotate unnecessarily, then they are also shifting in delay unnecessarily. Since it is desirable to have CIRs with the least amount of time variation, the methods devised in preceeding chapters that prevent unnecessary CIR dynamics are important.

- Effective and efficient nonuniform resampling was employed. This task is essential when there is significant relative motion between source and receiver, as there is for most of the data considered in this dissertation. In such cases, assuming the motion is unpredictable and unrestricted, it is necessary to perform the resampling in an adaptive manner to track possibly time-varying Doppler shift. This was accomplished by using the phase rotation caused by Doppler shift estimation error as an error signal to control the resampling.

- Tuchler's linear filter-based equalization method was derived directly from the SPA, using the approximation of symbol pmf's by suitable Gaussian pdf's. The implementation, however, is performed in a different manner – namely, it is implemented as a two-stage procedure. The first stage is linear filtering. Although CIR estimation error is theoretically taken into account in the process, the fact is that the filters are constructed from noisy CIR estimates instead of the true unknown CIRs. It was found to be better to estimate the gain and NPI variance (or, rather, the reciprocal of half this quantity) directly than to rely on the "theory" values.

- New methods of phase tracking were created. There are various ways of estimating the phase rotation caused by Doppler shift estimation error. Some methods, for example, estimate this phase directly from the data model (1.3) using estimates of the CIRs and symbols in place of the unknown true values. The IED algorithms of this dissertation, on the other hand, employ a different method. The details are presented earlier, the upshot is that the phase is estimated after equalization filtering, in conjunction with the other EOMPs and computation of the symbol a posteriori messages.

5.2 Future Research

Numerous extensions of the research related in this dissertation are possible. The area of wireless communication, even restricted to the shallow, long-range UWA channel, represents a vast research landscape. Some suggested directions for further investigation are listed below.

- The algorithms described in previous chapters do not store CIR estimates from previous iterations. However, if the CIR estimates are smoothed and stored after each iteration, then the next iteration can subtract off the stored CIR and estimate the residual. The advantage of doing so is that the residual should exhibit much less time variation. The drawback, of course, is storage.

- Other modulation strategies could be investigated. For instance, a multicarrier approach could be taken. The algorithms considered in this dissertation employ single-carrier modulation. On the other hand is orthogonal frequency-division multiplexing (OFDM), which subdivides a given frequency band into many sub-bands and consequently uses many subcarriers. The sub-bands are ideally frequency-flat, so that the corresponding CIRs are ISI-free. Splitting the band into too many sub-bands results in a significant loss of available bandwidth, however, since OFDM implementations typically require spacing, i.e., "guard bands", between the sub-bands. A compromise would be a multicarrier approach which uses a moderate number of subcarriers and allows ISI on the order of, say, five symbol periods (as opposed to 50 or more, which is the case considered in this dissertation). The resulting "partial response" CIRs could then be equalized using HMM-based methods.

- As another example of a different modulation strategy, not exclusive of others, is the use of trellis-coded modulation (TCM) (or, to be consistent with the terminology used throughout this dissertation, trellis-coded mapping). For instance, if used in conjunction with a multicarrier approach, a HMM-based procedure could be used to perform simultaneous equalization and de-mapping using a symbol state that spans both the memory of the TCM and the CIRs. Additionally, different types of channel coding could be investigated – e.g., LDPC or Turbo codes.

- The algorithms developed in this dissertation, and future ones suggested above, could be compared with previously reported algorithms. The focus of this dissertation has been the introduction of a new approach to equalization of UWA communication data. A natural next step would be the comparison of these algorithms with existing ones.

- The research detailed in this dissertation has concentrated on implementing algorithms that recover information from data collected during at-sea experiments. An extension of this research would be analysis of the overall algorithms and their various components. Because of the complexity of the algorithms – the adaptive resampling, channel estimation, and equalization – this would most likely involve simulations and Monte Carlo methods. In particular, one could examine the symbol messages that are passed around during the algorithms.

- It would be interesting to see the algorithms developed in this dissertation applied to other communication channels with CIRs of similar length. For instance, would they be able to process data transmitted over a wideband EM radio channel?

Bibliography

[1] J. A. Catipovic, "Performance limitations in underwater acoustic telemetry," *IEEE J. Oceanic Eng.*, vol. 15, no. 3, pp. 205–216, July 1990.

[2] F. B. Jensen, W. A. Kuperman, M. B. Porter, and H. Schmidt, *Computational Ocean Acoustics.* New York: Springer-Verlag, 2000.

[3] T. K. Moon and W. C. Stirling, *Mathematical Methods and Algorithms.* Upper Saddle River, NJ: Prentice Hall, 2000.

[4] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing.* New Jersey: Prentice Hall, 1989.

[5] A. Anastasopoulos, K. M. Chugg, G. Colavolpe, G. Ferrari, and R. Raheli, "Iterative detection for channels with memory," *Proc. IEEE*, vol. 95, pp. 1272–1294, June 2007.

[6] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[7] G. D. Forney, "Codes on graphs: Normal realizations," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 520–548, Feb. 2001.

[8] H.-A. Loeliger, "An introduction to factor graphs," *IEEE Signal Processing Mag.*, vol. 21, pp. 28–41, Jan. 2004.

[9] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. 20, pp. 284–287, Mar. 1974.

[10] P. Rusmevichientong and B. Roy, "An analysis of belief propagation on the turbo decoding graph with Gaussian densities," *IEEE Trans. Inform. Theory*, pp. 745–765, Feb. 2001.

[11] Y. Weiss and W. T. Freeman, "On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs," *IEEE Trans. Inform. Theory*, pp. 736–744, Feb. 2001.

[12] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo codes," *IEEE Trans. Commun.*, vol. 44, pp. 1261–1271, Oct. 1996.

[13] R. G. Gallagher, "Low-density parity-check codes," *IRE Trans. Inform. Theory.*, vol. 8, pp. 21–28, Jan. 1962.

[14] N. Wiberg, H.-A. Loeliger, and R. Kotter, "Codes and iterative decoding on general graphs," *Europ. Trans. Telecommun.*, vol. 6, pp. 513–525, Oct. 1995.

[15] C. Douillard, M. Jezequel, C. Berrou, A. Picart, A. Didier, and A. Glavieux, "Iterative correction of intersymbol interference: Turbo equalization," *European Trans. Telecomm*, vol. 6, pp. 507–511, Oct. 1995.

[16] R. Koetter, A. C. Singer, and M. Tuchler, "Turbo equalization," *IEEE Signal Processing Mag.*, pp. 67–80, Jan. 2004.

[17] R. Raheli, A. Polydoros, and C.-K. Tzou, "Per-survivor processing: A general approach to MLSE in uncertain environments," *IEEE Trans. Commun.*, vol. 43, pp. 354–364, Feb. 1995.

[18] L. M. Davis, I. B. Collings, and P. Hoeher, "Joint map equalization and channel estimation for frequency- selective and frequency-flat fast-fading channels," *IEEE Trans. Commun.*, vol. 49, pp. 2106–2114, Dec. 2001.

[19] H.-A. Loeliger, "Some remarks on factor graphs," in *Proc. Int. Symp. on Turbo Codes*, 2003, pp. 111–115.

[20] R. Otnes and M. Tuchler, "Iterative channel estimation for turbo equalization of time-varying frequency-selective channels," *IEEE Trans. Wireless Commun.*, vol. 3, no. 6, pp. 1918–1923, Nov. 2004.

[21] S. Song, A. C. Singer, and K.-M. Sung, "Soft input channel estimation for turbo equalization," *IEEE Trans. Signal Processing*, vol. 52, no. 10, pp. 2885–2894, Oct. 2004.

[22] M. Tuchler, A. C. Singer, and R. Koetter, "Minimum mean squared error equalization using a priori information," *IEEE Trans. Signal Processing*, vol. 50, no. 3, pp. 673–683, Mar. 2002.

[23] M. Tuchler, R. Koetter, and A. C. Singer, "Turbo equalization: Principles and new results," *IEEE Trans. Commun.*, vol. 50, no. 5, pp. 754–767, May 2002.

[24] R. R. Lopes and J. R. Barry, "The soft-feedback equalizer for turbo equalization of highly dispersive channels," *IEEE Trans. Commun.*, pp. 783–788, May 2006.

[25] K. R. Narayanan, "Effect of precoding on the convergence of turbo equalization for partial response channels," *IEEE J. Select. Areas Commun.*, pp. 686–698, Apr. 2001.

[26] I. Lee, "The effect of a precoder on serially concatenated coding systems with an ISI channel," *IEEE Trans. Commun.*, pp. 1168–1175, July 2001.

[27] A. B. Baggeroer, "Acoustic telemetry–an overview," *IEEE J. Oceanic Eng.*, vol. OE-9, no. 4, pp. 229–235, Oct. 1984.

[28] M. Stojanovic, "Recent advances in high-speed underwater acoustic communications," *IEEE J. Oceanic Eng.*, vol. 21, pp. 125–136, Apr. 1996.

[29] D. B. Kilfoyle and A. B. Baggeroer, "The state of the art in underwater acoustic telemetry," *IEEE J. Oceanic Eng.*, vol. 25, no. 1, pp. 4–27, Jan. 2000.

[30] L. Freitag, M. Stojanovic, D. B. Kilfoyle, and J. C. Preisig, "High-rate phase coherent acoustic communication: A review of a decade of research and a perspective on future challenges," in *Proc. ECUA*, Delft, Netherlands, July 5–8, 2004.

[31] T. C. Yang, "Differences between passive-phase conjugation and decision-feedback equalizer for underwater acoustic communications," *IEEE J. Oceanic Eng.*, vol. 29, no. 2, pp. 472–487, Apr. 2004.

[32] M. Stojanovic, J. A. Catipovic, and J. G. Proakis, "Phase-coherent digital communications for underwater acoustic channels," *IEEE J. Oceanic Eng.*, vol. 19, no. 1, pp. 100–111, Jan. 1994.

[33] L. Litwin and M. Pugel, "The principles of OFDM," *RF Design*, pp. 30–48, Jan. 2001.

[34] M. Z. Win and R. A. Scholtz, "Impulse radio: How it works," *IEEE Commun. Lett.*, vol. 2, no. 2, pp. 36–38, Feb. 1998.

[35] S. Benedetto, D. Divsalar, and J. Hagenauer, "Guest editorial concatenated coding techniques and iterative decoding: Sailing toward channel capacity," *IEEE J. Select. Areas Commun.*, vol. 16, no. 2, pp. 137–139, Feb. 1998.

[36] J. Hagenauer, "The turbo principle: Tutorial introduction and state of the art," in *Symp. on Turbo Codes*, Brest, France, Sept. 1997, pp. 1–11.

[37] E. Sozer, J. G. Proakis, and F. Blackmon, "Iterative equalization and decoding techniques for shallow water acoustic channels," in *Proc. Oceans*, vol. 4, Honolulu, Hawaii, Nov. 5–8, 2001, pp. 2201–2208.

[38] F. Blackmon, E. Sozer, M. Murandian, J. G. Proakis, and M. Salehi, "Performance comparison of iterative/integral equalizer/decoder structures for underwater acoustic channels," in *Proc. Oceans*, vol. 4, Honolulu, Hawaii, Nov. 5–8, 2001, pp. 2191–2200.

[39] F. Blackmon, E. Sozer, and J. G. Proakis, "Iterative equalization, decoding, and soft diversity combining for underwater acoustic channels," in *Proc. Oceans*, vol. 4, Oct. 29–31, 2002, pp. 2425–2428.

[40] J. A. Flynn and J. A. Ritcey, "Turbo array receiver for underwater telemetry," in *Proc. Asilomar Conf. on Signals, Systems and Computers*, vol. 2, Nov. 7–10, 2004, pp. 1421–1425.

[41] T. Oberg, B. Nilsson, N. Olofsson, M. L. Nordenvaad, and E. Sangfelt, "Underwater communication link with iterative equalization," in *Proc. Oceans*, Sept. 2006, pp. 1–6.

[42] M. L. Nordenvaad and T. Oberg, "Iterative reception for acoustic underwater MIMO communications," in *Proc. Oceans*, Sept. 2006, pp. 1–6.

[43] Y. Li, B. Vucetic, and Y. Sato, "Optimum soft-output detection for channels with intersymbol interference," *IEEE Trans. Inform. Theory*, vol. 41, pp. 704–713, May 1995.

[44] S. Jiang, L. Ping, H. Sun, and C. S. Leung, "Modified LMMSE turbo equalization," *IEEE Commun. Lett.*, vol. 8, no. 3, pp. 174–176, Mar. 2004.

[45] M. Tuchler, R. Otnes, and A. Schmidbauer, "Performance of soft iterative channel estimation in turbo equalization," in *Proc. Int. Conf. on Commun.*, May 2002, pp. 1858–1862.

[46] G. Bauch and V. Franz, "A comparison of soft-in/soft-out algorithms for turbo detection," in *Proc. Int. Conf. Telecomm.*, June 1998, pp. 259–263.

[47] J. Dauwels and H.-A. Loeliger, "Phase estimation by message passing," in *Proc. IEEE ICC*, vol. 1, June 20–24, 2004, pp. 523–527.

[48] N. Noels, C. Herzet, A. Dejonghe, V. Lottici, H. Steendam, M. Moeneclaey, M. Luise, and L. Vandendorpe, "Turbo synchronization: an EM algorithm interpretation," in *Proc. IEEE ICC*, vol. 4, May 11–15, 2003, pp. 2933–2937.

[49] W. Zeng and A. Kavcic, “Optimal soft-output detector for channels with intersymbol interference and timing errors,” *IEEE Trans. Magn.*, vol. 39, no. 5, pp. 2555–2257, Sept. 2003.

[50] J. R. Barry, A. Kavcic, S. W. McLaughlin, A. Nayak, and W. Zeng, “Iterative timing recovery,” *IEEE Signal Processing Mag.*, pp. 89–102, Jan. 2004.

[51] K. J. Larsen, “Short convolutional codes with maximal free distance for rates $\frac{1}{2}$, $\frac{1}{3}$, and $\frac{1}{4}$,” *IEEE Trans. Inform. Theory*, pp. 371–372, May 1973.

[52] C. Heegard and S. Wicker, *Turbo Coding.* Boston, MA: Kluwer, 1999.

[53] J. G. Proakis, *Digital Communications.* New York: McGraw-Hill, 2001.

[54] C. Laot and N. L. Josse, “A closed-form solution for the finite length constant modulus receiver,” in *Proc. Int. Symp. on Info. Thy.*, Sept. 4–9, 2005, pp. 865–869.

[55] P. A. Regalia, “A finite-interval constant modulus algorithm,” in *Proc. IEEE ICASSP*, vol. 3, May 13–17, 2002, pp. III–2285–III–2288.

[56] P. Stoica and Y. Seln, “Cyclic minimizers, majorization techniques, and the expectation-maximization algorithm: A refresher,” *IEEE Signal Processing Mag.*, vol. 21, no. 1, pp. 112–114, Jan. 2004.

[57] P. Schniter. (2005, Oct. 18,) Polyphase interpolation. Connexions. [Online]. Available: http://cnx.org/content/m10431/2.11/

[58] S. Aji and R. McEliece, “The generalized distributive law,” *IEEE Trans. Inform. Theory*, vol. 46, pp. 325–343, Mar. 2000.

[59] R. A. Iltis and A. W. Fuxjaeger, “A digital ds spread-spectrum receiver with joint channel and doppler shift estimation,” *IEEE Trans. Commun.*, vol. 39, no. 8, pp. 1255–1267, Aug. 1991.

[60] B. S. Sharif, J. Neasham, O. R. Hinton, and A. E. Adams, “A computationally efficient doppler compensation system for underwater acoustic communications,” *IEEE J. Oceanic Eng.*, vol. 25, no. 1, pp. 52–61, Jan. 2000.

[61] M. Johnson, L. Freitag, and M. Stojanovic, “Improved doppler tracking and correction for underwater acoustic communications,” in *Proc. IEEE ICASSP*, vol. 1, 21–24, 1997, pp. 575–578.

[62] J. F. Sifferlen, H. C. Song, W. S. Hodgkiss, W. A. Kuperman, and J. M. Stevenson, “An iterative equalization and decoding approach for underwater acoustic communication,” *IEEE J. Oceanic Eng.*, vol. 33, no. 2, pp. 182–197, Apr. 2008.

[63] J. F. Sifferlen, H. C. Song, W. S. Hodgkiss, W. A. Kuperman, and M. Stevenson, “Iterative equalization and decoding of communication data received from a moving underwater acoustic source,” *IEEE J. Oceanic Eng.*, submitted for publication.