On the Partial Haar Dual Adaptive Filter for Sparse Echo Cancellation

Patrick Kechichian

Department of Electrical & Computer Engineering
McGill University
Montreal, Canada

September 2006

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Master of Engineering.

2006/09/28

Abstract

Typical sparse echo cancellers employ adaptive filtering algorithms that update only a small number of filter coefficients that produce the actual echo. Usually, these algorithms provide increased convergence speed at the cost of an increase in computational complexity for locating these significant filter coefficients. Recently, a coupled echo canceller was proposed that uses two short adaptive filters in tandem. The first adaptive filter operates in the partial Haar domain and is solely used to estimate the location of the channel's dispersive region. A short time-domain filter is then centred around this estimate to cancel echo. Using two short filters instead of one long filter not only reduces computational complexity, while substantially increasing the convergence speed of the echo canceller.

The focus of this thesis is twofold. First, it analyzes the partial Haar echo canceller and attempts to clarify some issues with its implementation. Second and foremost, it identifies and proposes feasible solutions to three inherent weaknesses of the coupled echo canceller. These include alleviating the adverse effect caused by the shift-variant property of wavelets, improving the tracking performance of the coupled echo canceller in response to abrupt changes in the echo path impulse response, and extending the original echo canceller to support the cancellation of multiple echoes. Simulations support the resulting improvements when each of the proposed solutions is incorporated into the coupled echo canceller.

Sommaire

Les annuleurs d'écho non denses typiques utilisent des algorithmes de filtrage adaptatif qui mettent à jour uniquement un nombre restreint de coefficients de filtre qui produisent l'écho réel. Habituellement, ces algorithmes se caractérisent par une vitesse de convergence plus rapide au détriment d'une augmentation de la complexité informatique pour localiser les coefficients de filtre signifiants. Récemment, on a proposé un annuleur d'écho couplé qui utilise deux filtres adaptatifs courts en tandem. Le premier filtre adaptatif fonctionne dans le domaine partiel de Haar et est employé pour estimer l'endroit où le canal présente une caractéristique dispersive. Un filtre court dans le domaine du temps est alors centré sur cette région pour annuler l'écho. L'utilisation de deux filtres courts au lieu d'un seul long filtre réduit non seulement la complexité informatique, mais elle augmente également la vitesse de convergence de l'annuleur d'écho considérablement.

Deux points principaux sont abordés dans cette thèse. D'abord, nous analysons l'annuleur d'écho partiel de Haar et essayons de clarifier quelques problèmes reliés à sa mise en oeuvre. En second lieu, nous identifions et proposons des solutions appropriées à trois faiblesses inhérentes de l'annuleur d'écho couplé. Celles-ci comprennent la réduction des effets nuisibles causés par la variation due au décalage des ondelettes, l'amélioration de la capacité de poursuite de l'annuleur d'écho couplé suite à un changement brusque de la réponse impulsionnelle de l'écho, et la modification de l'annuleur d'écho original pour permettre l'annulation des échos multiples. Lorsque chacune des solutions proposées est incorporée à l'annuleur d'écho couplé, des simulations confirment les améliorations obtenues.

Acknowledgments

I would like to thank my supervisor, Professor Benoît Champagne for his financial support and motivating guidance throughout the course of my research and the writing process of this thesis. I would also like to thank the department of Electrical and Computer Engineering at McGill University for their differential fee waiver program, which was a big help. I am thankful to François Duplessis-Beaulieu for providing a French translation of the abstract. Of course, the lab environment and the people that make up that environment also positively impacted my work. Therefore, for feeding the ever-voracious 'vortex' (an essential component of anyone's graduate studies), I would like to thank the following members of the TSP labs: Clarence Wong, Denis Tran, Frederic Thouin, Garrick Ing, Benoît Pelltier, Robert Kwun, Eric Plourde, and Wei Chu.

Finally, I could never have reached this point of accomplishment if it were not for my parents and my brother, Philippe, who have always stood behind me in whatever academic endeavours I wished to follow, from pursuing my Bachelor's degree at McGill University, to continuing into my Graduate studies.

Contents

List of Figures

List of Tables

List of Acronyms

CDF	Cumulative Distribution Function
CWT	Continuous Wavelet Transform
DMT	Discrete Multitone Modulation
DST	Dempster-Shafer Theory
DSmT	Dezert-Smarandache Theory
DWT	Discrete Wavelet Transform
ERL	Echo Return Loss
FIR	Finite Impulse Response
IT-PHDAF	Improved Tracking Partial Haar Dual Adaptive Filter
ITU	International Telecommunication Union
LMS	Least Mean Square
LS	Least Squares
LTI	Linear Time Invariant
ME-PHDAF	Multiple Echo Partial Haar Dual Adaptive Filter
MFLOPS	Million Floating Point Operations per Second
MRA	Multi-resolution Analysis
MSE	Mean Squared Error
NLMS	Normalized Least Mean Squares
PDM	Peak Discernibility Measure
PDF	Probability Density Function
PHDAF	Partial Haar Dual Adaptive Filter
PNLMS	Proportionate Normalized Least Mean Squares
PSTN	Public Switched Telephone Network
RLS	Recursive Least Squares

RDWT	Redundant Discrete Wavelet Transform
SCE-PHDAF	Suboptimal Context Escaping Partial Haar Dual Adaptive Filter
SNR	Signal-to-Noise ratio
STWQ	Scrub Taps Waiting in a Queue
VoIP	Voice over Internet Protocol

Chapter 1

Introduction

This chapter is divided into four parts. First, the telephone network environment is introduced as a backdrop to the problem of line echo and its implications. This is followed by a review of sparse echo cancellers that have been developed over recent years, including a specific echo canceller which will be the focus of this work. Section 1.3 discusses the scope of this research and the contributions made. The chapter concludes with an overview of the subsequent chapters of this text.

1.1 Line Echo in Voice Communications

The presence of echo has been and is still commonplace in today's ever-expanding communication infrastructure. In a telephone call scenario, the echo phenomenon can be described by a caller as hearing a duplicate of his or her voice delayed in time. Depending on the delay of the duplicate signal, the echo can be characterized as nearly imperceptible for small delays to obstructing conversations when the delay is longer.

The most well known occurrence of echo is in telephone networks. A user's phone is connected to the local exchange or local telephone company (also termed the central office) via a twisted pair of copper wires called the subscriber loop, terminating in a line circuit, that connects to the public-switched telephone network (PSTN) [41]. The line circuit includes a device called a hybrid that converts the twisted pair running from a user's premises into a four-wire connection, where each pair of wires is separately used for transmit and receive signals.

An impedance mismatch at the far-end hybrid causes some of the transmitted signal to leak into the near-end caller's receive path and is perceived as a reflection (see Fig. 1.1 with speaker A at the near-end and speaker B at the far-end) [48]. If the distance between callers is short, the round-trip echo delay is small and the reflection is perceived as either a slight amplification or soft reverberation. As the distance increases (delay $>$ 16 ms), this reverberation becomes an echo and can severely impede the conversation. The fact that most long distance calls are routed via satellites only makes matters worse, since the round-trip echo delay can reach up to 600 ms over these connections.

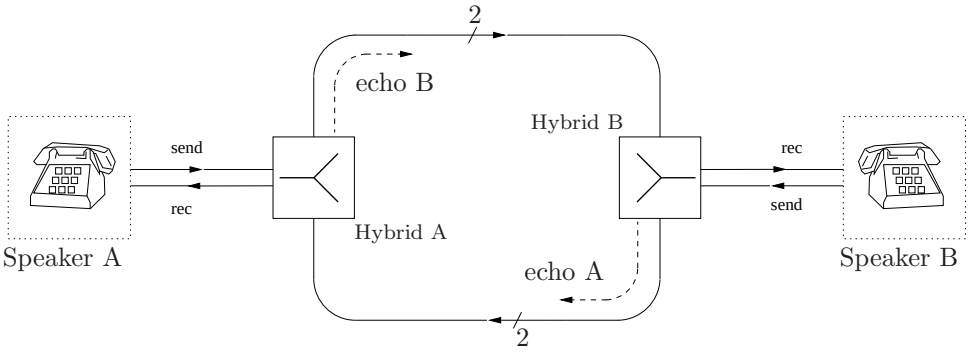


Fig. 1.1 Telephone network and line echo.

In the past, echo suppressors were installed on both the incoming and outgoing speech paths to solve the echo problem. These devices suppress echo by classifying the signal on the receive path as either speech or echo [48]. If the signal is classified as echo, then it is attenuated. Usually, the classification is based on the amplitude of the signal, which frequently leads to false classifications. This is largely one of the reason why echo suppressors have had limited success in practice.

Echo cancellers employing adaptive filters, on the other hand, have been shown to be more successful at cancelling line echo. By adaptively identifying the echo path impulse response, the echo canceller can create a replica of the echo and therefore cancel it from the received signal. The convergence speed of an adaptive filter to its steady-state coefficients can vary, however, and depends on many factors including the input correlation matrix and whether the echo path response is stationary or not. In the latter case, the echo canceller should be capable of tracking rapid changes in the impulse response.

Another hindrance to an echo canceller is double-talk, which occurs when both users are speaking simultaneously. Double-talk corrupts the desired signal required to cancel the echo, and negatively affects the steady-state mode of the adaptive filter's operation. Although it is not too common for people to be speaking simultaneously during a phone call, the control logic should inhibit the echo canceller when double-talk does occur. Double-talk detection will not be considered in this thesis.

An interesting fact about advancements in improving communication quality has been an increase in the adverse effects of echo. By adding coders and signal processing blocks into the line circuit, processing delays ranging from 80–100 ms have been introduced into the round trip delay of echo [48]. Therefore, line echo that was previously perceived as a slight amplification or reverberation can now be distinctly heard as an echo. This byproduct of increased processing delays is making the effects of echo even worse for Voice over Internet Protocol (VoIP) telephones, which are deployed onto the existing telephone infrastructure. In addition to the processing delays already present in the network, these telephones also require buffering delays for the packetization of speech, not to mention delays resulting from the sharing of network resources with other data packets [34]. One of the well known problems encountered in VoIP is known as 'initial echo', which is echo experienced at the beginning of a phone call, while the echo canceller is still converging, allowing the reflection of residual echo back to the speaker [26].

In addition to affecting voice communications, echo also impedes the introduction of transmission methods such as discrete multitone modulation (DMT), which divides the transmission channel into a set of orthogonal subchannels [9, 25]. Echo cancellation is usually not necessary when different sets of subchannels are used for up- and down-links. However, full-duplex transmission, which can significantly increase the obtainable data rates, requires echo cancellation.

An example of a network impulse response is shown in Fig. 1.2. The response contains a flat delay region or bulk delay equal to the round-trip time of the transmitted and reflected signal (region A) . The group of non-zero coefficients following the flat delay region is termed the dispersive region (region B), which produces the echo. Region C is a zero-energy zone, i.e. all impulse response coefficients are zero.

Besides line echo, another type of echo known as acoustic is present in hands-free

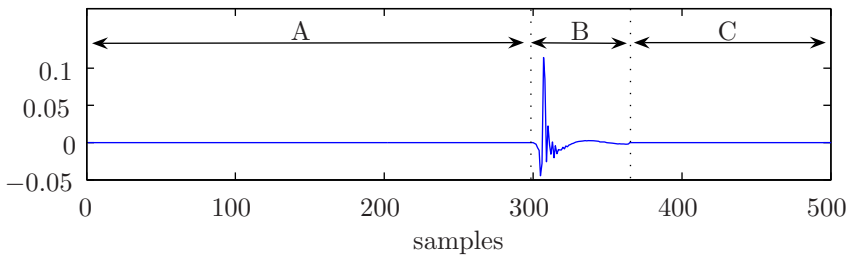


Fig. 1.2 Typical network impulse response.

communication. This echo results from the multiple reflections of sound emitted by the loudspeaker (in three-dimensional space), which reach a microphone with different delays [7]. Unlike acoustic echo, network or line echo involves specific reflection points (impedance mismatches) placed over longer distances, along a *one-dimensional* transmission medium. This makes echo path impulse responses in networks inherently sparse compared to acoustic echo.

Although both types of echo affect telephone conversations, this thesis will only be concerned with line echo. The main focus of this work will be based on an efficient and fast echo canceller that takes advantage of the sparse characteristic of echo path impulse responses in telephone networks.

1.2 Literature Review

The literature abounds with adaptive filtering algorithms that exploit the sparse characteristics of line echo [33]. Most algorithms are based on finding ways to determine which filter coefficients are actually associated with the echo, and then adapting only these coefficients.

One of the earliest works on sparse echo cancellation is by Duttweiler [16], where the input and desired signals are bandpass-filtered and decimated, and used by a short (i.e. small number of coefficients) adaptive filter to locate the impulse response's dispersive region. If D is the decimation factor, then this approach requires only $1/D$ as many taps as a traditional echo canceller. A shorter filter operating at the original sampling rate is then centred around the dispersive region to cancel the echo. Using this approach, only two short adaptive filters are required, compared to one long

filter, reducing complexity and increasing convergence speed. However, Duttweiler's echo canceller suffers from many drawbacks as remarked by [4]. The most obvious drawback stems from the observation that the convergence time of the echo canceller depends on the decimated adaptive filter, which is operating D-times slower than the rate of incoming data. Therefore, the time required to correctly estimate the delay can be quite long. Secondly, in speech applications, the bandlimiting operation can remove important frequency components in the signals, preventing proper convergence of the subsampled adaptive filter.

An adaptive multiple echo canceller is proposed in [50]. It uses a full-length primary adaptive filter in parallel with a group of short secondary adaptive filters. A monitor/control unit switches between the output of these two classes of filters, depending on specific conditions. The full-length filter is used to track multiple dispersive regions and initially cancels echo, while each short adaptive filter is centred around these regions once the full-length filter has sufficiently converged. One of the biggest problems with this approach is the added complexity of the monitor/control unit, in addition to the extra hardware and power requirements of a full length adaptive filter.

In [46], an algorithm based on the so-called "Scrub Taps Waiting in a Queue" (STWQ) approach [27] for sparse channel impulse responses is proposed that uses a two-stage adaptation process. The first part of the process estimates the flat delay region, and the second part consists of adapting those filter coefficients using a constrained tap-position control. This constrained control puts a limit on which filter taps can be updated based on their relative position within the dispersive region.

Probably one of the most well-known class of sparse echo cancellers are based on the Proportionate Normalized Least Mean Squares (PNLMS) algorithm [17] and its ubiquitous variants. These include the improved PNLMS (IPNLMS) [3], the PNLMS++ [19], and the improved IPNLMS [11]. These algorithms allocate individual step-size gains in proportion to the magnitude of each filter coefficient. Therefore, coefficients with large magnitudes converge quickly to their steady-state values, while coefficients in the flat delay region that would normally only contribute to tap-weight noise are updated using very small step-sizes. The only drawbacks of this algorithm are its increased computational complexity and its degradation in performance as the

length and/or number of dispersive regions increases.

Other attempts at improving sparse echo cancellation rely on using orthogonal transforms such as the wavelet transform on the input data. In [24], the authors propose using a subset of Haar wavelet coefficients to detect the significant channel coefficients. Then, by exploiting the hierarchical structure of the dyadic wavelet expansion, the locations of these significant coefficients are used to activate coefficients in the remaining subsets that share the same non-zero time-support. Therefore, only significant filter coefficients are adapted, increasing convergence speed and reducing computational complexity.

More recently, Bershad and Bist [4] have proposed a novel way of cancelling sparse echo, using a coupled echo canceller consisting of two short adaptive filters similar to Duttweiler's approach [16]. The first partial Haar filter operates on only a subset of input Haar coefficients, and is used by a peak delay estimator to determine the location of the echo path's dispersive region. Unlike Duttweiler's echo canceller [16] which requires the design of complex bandpass filters, the Haar wavelet transform is simpler and just as amenable to digital signal processors. The second time-domain filter is then centred around this location to actually cancel the echo. In cases where the bulk delay is very large and a traditional echo canceller would require a large number of taps, this new method provides a significant reduction in computation and memory requirements. Most importantly, by reducing the number of filter taps, the convergence speed of the overall echo canceller is increased.

1.3 Research Objectives and Contribution

Although some of the previously-mentioned sparse adaptive filtering algorithms have shown promising results, the coupled echo canceller proposed by Bershad and Bist [4] uses a more novel method of incorporating channel-monitoring for sparse echo cancellation. Because channel-monitoring is performed in parallel with the actual echo cancellation process, this can provide greater flexibility from a researcher's point of view in experimenting with new ideas to improve the overall echo canceller. With its significant increase in convergence speed and reduced complexity, the coupled echo canceller still suffers from a few drawbacks, however.

First, the shift-variant property of wavelet transforms makes the performance of the partial Haar adaptive filter highly dependent on the echo path bulk delay. As a result, the specific value of the bulk delay can affect the amount of time it takes the peak delay estimator to correctly estimate the location of the dispersive region, significantly affecting the overall convergence speed of the echo canceller.

Second, the promising results shown by [4] that support the increased convergence speed are only valid for stationary channels. The reason for this is the simulations that produced these results assume that the filter taps are initially set to zero. In non-stationary environments, where the bulk delay of the echo path impulse response can abruptly change, the amount of time required by the peak delay estimator to locate new dispersive regions can be very large.

Finally, the coupled echo canceller has only been analyzed and developed for the case of a single dispersive region. In today's communication networks, interfaces between different transmission media such as copper to fibre optic cables can also produce unwanted echo [48]. Consequently, it is not uncommon for an echo path impulse response to contain multiple dispersive regions. It would therefore be necessary to modify the original coupled echo canceller [4] to accommodate multiple echoes.

In addition to some implementation issues, this thesis looks at the above problems, and investigates possible solutions for each. More specifically,

- To deal with the shift-variant property of wavelet transforms, a peak *tendency* estimator is proposed that is based on non-Bayesian evidence theory and fuzzy inference. The estimator categorizes a peak's magnitude behaviour as either increasing or decreasing. When the current peak is categorized as decreasing and displaying jitter, then a different set of transformed input coefficients is used to drive the partial Haar adaptive filter to a Wiener solution with a larger peak. Deciding when to use a different transformed input vector is based on a schedule of trial periods.

- The proposed peak tendency estimator is also used to improve the tracking performance of the coupled echo canceller in situations where an abrupt change in an echo path impulse response's bulk delay occurs. Because the partial Haar adaptive filter is not cancelling echo directly, it can be reset to zero whenever a

change in bulk delay is detected as a decrease in the current peak magnitude. The old peak position is used to offset the short time-domain filter until a new peak has been found. As Bershad and Bist have shown [4], the peak delay estimator usually locates peaks faster when the partial Haar adaptive filter is reset to zero.

- A multiple echo canceller similar to [50] is proposed that assigns multiple agents to overlapping regions of the partial Haar adaptive filter, with each agent performing peak tendency estimation. When true peaks are found, their locations are relayed to a central coordinator that activates or deactivates the coefficients of a full-length time-domain filter used to cancel echo.

The performance gains of the above contributions are supported by simulation. Each contribution to the echo canceller is tested using ITU-T G.168 [1] hybrid impulse responses. The simulations are divided into two classes and range from specific test-scenarios to randomly generated cases. Each class of tests is required to understand the proposed algorithms in detail while at the same time establishing their performance in general. It is shown that each of the proposed solutions can substantially improve the coupled echo canceller's performance.

1.4 Thesis Overview

This text is organized as follows: Chapter 2 begins with an overview of wavelets and adaptive filtering theory. The aim of this chapter is to focus and narrow the breadth of these fields within the scope of the coupled echo canceller.

Chapter 3 provides the analysis of the coupled echo canceller proposed by Bershad and Bist [4], followed by an in depth critique of the different problems associated with this echo canceller.

Proposed solutions to these problems are developed in Chapter 4. These include ways of mitigating the effects of shift variance, improved tracking of abrupt changes in bulk delays, and extending the echo canceller to accommodate multiple dispersive regions.

Chapter 5 provides a series of simulation experiments to justify the advantages of

the proposed solutions. These simulations are accompanied by discussions pertaining to the results.

Finally, Chapter 6 concludes the text with a brief summary of the work, in addition to possible future work and improvements to the proposed algorithms.

Chapter 2

Background

This chapter provides the theoretical background necessary to understand the ensuing discussion in Chapter 3 of the coupled algorithm proposed by Bershad and Bist [4]. Because the coupled algorithm makes use of the Haar transform, the first section provides a brief introduction to wavelets. The second section discusses basic adaptive filtering theory, with particular emphasis on the LMS and NLMS algorithms.

2.1 A Brief Introduction to Wavelets

The theory of wavelets and their mathematical formulation have been well established since the early 1900s [8]. However, it has only been after the advent of improved processing power that the full potential and flexibility of wavelets have been realized in digital signal processing. In fact, one of the first algorithms utilizing wavelets for signal analysis dates back to the work of Stephane Mallat [32] in the 1980s.

Wavelets can be viewed as functions that obey certain mathematical constraints to represent other functions (signals). In addition to the time-frequency descriptive provided by wavelet transforms, these mathematical constraints also make possible the notion of multiresolution analysis. In other words, by properly scaling (i.e. dilating or contracting) and shifting a wavelet function along the time-axis, it is possible to describe the same signal with varying degrees of detail.

Unlike the class of Fourier expansions that decompose a signal in terms of sinusoids with infinite time-support, wavelet functions are not restricted to any one class of basis

functions, and also include functions with finite time-support. This makes it possible to construct wavelet functions to obtain signal representations that are sparse or that emphasize certain desired properties of a signal [38]. Wavelet transforms also provide a dual time-frequency description of a signal which is absent from Fourier analysis.

Applications of wavelets in signal processing are seemingly endless, and include image processing, time-series analysis, sound synthesis, and data compression [21]. In applications such as image denoising, an image is decomposed into a set of wavelet coefficients, and those coefficients with magnitudes less than a specified threshold are set to zero. A denoised version is then constructed using the inverse wavelet transform of the resulting coefficients. Wavelets enjoy even more widespread use in data compression, where their sparse representations allow the storage of a signal's information in very few coefficients [38].

The following section discusses the fundamentals of wavelets and their properties. This discussion is limited to the Discrete Wavelet Transform (DWT), since its analysis leads to practical and efficient applications in digital signal processing.

2.1.1 Discrete Wavelet Transform

The DWT arose from a necessity to overcome some of the inherent difficulties associated with the Continuous Wavelet Transform (CWT) [31]. These difficulties include redundancy, and an uncountable number of wavelets resulting from continuous scaling and translation variables. More importantly, however, the CWT does not lend itself to any fast algorithms for computing the transform [31]. Although discrete wavelets are continuous functions of time, they can only be scaled and translated in discrete steps, such as the dyadic DWT which uses scaling steps of size 2 and will be considered here.

A. *Scaling Function*

The set of scaling functions is defined as

$$\varphi_{j,k}(t) = 2^{j/2}\varphi(2^j t - k), \qquad j,k \in \mathbb{Z}, \quad \varphi \in L^2(\mathbb{R}), \tag{2.1}$$

where $\varphi(t)$ is the scaling function from which all others are derived and $L^2(\mathbb{R})$ rep-

resents the set of functions (of real variables) with well-defined square integrals [8]. This function has two properties. First, it is scalable in time ($2^j t$ term), and second, at each scale, the set of functions include integer-translates k of $\varphi(2^j t)$. A small scale index j corresponds to a coarse time-resolution, while a large scale index reflects finer time-resolution[1]. In addition, for the sake of simplicity, the scale index j will only assume non-negative values. The set of scaling functions is two-dimensional, as it represents a given localization in time determined by k at a given scale index j, hence the term time-frequency resolution associated with wavelets.

The time support of the function $\varphi(t)$ is usually normalized to the interval $[0, 1)$. As a result, the value of k ranges from 0 to $2^j - 1$. As the scale index increases (finer time resolution), the number of possible translations increases exponentially. At a given scale index j, the subspace spanned by these functions over translations k is denoted by

$$V_j = \text{span}\{\varphi_{j,k}(t)\}. \tag{2.2}$$

Multiresolution analysis is made possible by requiring that

$$V_j \subset V_{j+1}. \tag{2.3}$$

This means that those coefficients generated by expanding a signal at a coarser scale can be constructed from coefficients at a finer scale. The expression that relates the two is known as the multiresolution analysis (MRA) equation

$$\varphi(t) = \sum_k h_0(k)\sqrt{2}\varphi(2t - k), \tag{2.4}$$

where the $h_0(k)$ denote scaling filter coefficients (more on this in the next section). The above equation provides a recursion to construct the scaling function at a scale index j from the next finest scale index $j + 1$.

[1] this is a consequence of defining the scaling factor as 2^j instead of 2^{-j}

B. *Wavelet Function*

The wavelet function is given by

$$\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k), \qquad j,k \in \mathbb{Z}, \tag{2.5}$$

where $\psi(t)$ is known as the mother wavelet, and the variables j and k again represent the scale and translation indices, respectively. At a given scale index j, the subspace spanned by these wavelet functions is denoted by

$$W_j = \text{span}\{\psi_{j,k}(t)\}. \tag{2.6}$$

The set of wavelet functions span the differences between the spaces spanned by successive scaling functions $\varphi(2^j t)$ and $\varphi(2^{j+1}t)$. By construction, this space is related to V_j, and satisfies the following condition,

$$V_{j+1} = V_j \oplus W_j, \tag{2.7}$$

where $\oplus$ denotes a direct sum of subspaces. Equation (2.7) implies that

$$V_j = V_0 \oplus W_0 \oplus W_1 \oplus \cdots \oplus W_{j-1}. \tag{2.8}$$

Furthermore, one can conclude that in order to span V_j the initial subspace V_{j_0} can have an arbitrary scale $0 \leq j_0 < j$. Therefore, V_j is can also be represented by

$$V_j = V_{j_0} \oplus W_{j_0} \oplus W_{j_0+1} \oplus \cdots \oplus W_{j-1}. \tag{2.9}$$

This makes sense intuitively because from (2.7), the span of the initial scaling functions contain the spans of all the scaling functions with scale indices less than j_0, i.e.

$$V_0 \subset V_1 \subset \cdots \subset V_{j_0}. \tag{2.10}$$

As a result of (2.7) and (2.9), any function $g(t) \in L^2(\mathbb{R})$ can be written as a sum of

time-shifted versions of an arbitrary scaling function and a set of wavelet functions,

$$g(t) = \sum_k c_{j_0}(k)\varphi_{j_0,k}(t) + \sum_{j=j_0}^{\infty}\sum_k d_j(k)\psi_{j,k}(t), \tag{2.11}$$

where $c_j(k)$ and $d_j(k)$ are the scaling and wavelet (or difference) coefficients, respectively. To make the calculation of the wavelet coefficients tractable, an important requirement is for the wavelet and scaling functions to be orthogonal,

$$\langle \varphi_{j,k}(t), \psi_{j,l}(t) \rangle = \int \varphi_{j,k}(t)\psi_{j,l}(t)dt = 0, \qquad \forall j, k \neq l \in \mathbb{Z} \tag{2.12}$$

and for the wavelet functions, whether at the same or different scale, to be orthogonal

$$\langle \psi_{j,k}(t), \psi_{j',l}(t) \rangle = \int \psi_{j,k}(t)\psi_{j',l}(t)dt = 0, \qquad \begin{array}{l} \forall j = j', k \neq l \in \mathbb{Z} \\ \forall (j,k) \neq (j',l) \in \mathbb{Z}^2 \end{array} \tag{2.13}$$

Furthermore, if the scaling and wavelet functions have unit norm in addition to the conditions in (2.12) and (2.13), then Parseval's relation holds true for the DWT, i.e.

$$\int |g(t)|^2 dt = \sum_k |c_{j_0}(k)|^2 + \sum_{j=j_0}^{\infty}\sum_k |d_j(k)|^2. \tag{2.14}$$

Finally, the wavelet function also possesses a MRA equation that allows for the construction of a coarser wavelet from the scaling function at the next higher scale

$$\psi(t) = \sum_k h_1(k)\sqrt{2}\varphi(2t-k), \tag{2.15}$$

where the wavelet filter, $h_1(k)$ is related (by the requirement of orthogonality) to the scaling filter in (2.4) by

$$h_1(k) = (-1)^k h_0(1-k). \tag{2.16}$$

Two other very important properties of wavelet transforms are the regularity and admissibility conditions. The regularity conditions dictate that a wavelet function's energy should be concentrated in both frequency and time domains, in addition to displaying some smoothness in these domains [31]. The admissibility requirements

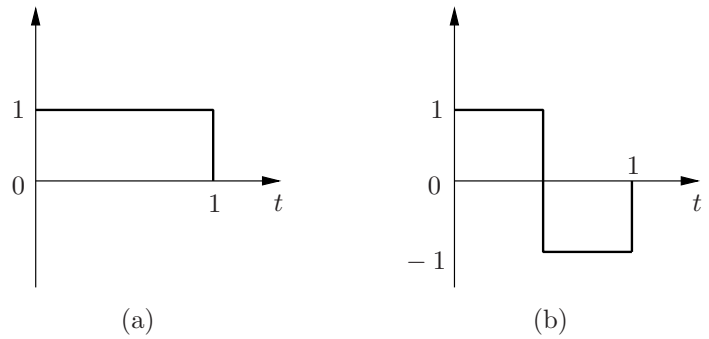
state that the Fourier transform of the wavelet function is zero at the zero frequency (the time average of the wavelet function is zero). From a spectral perspective then, it is now apparent why scaling functions are used in conjunction with wavelets. Scaling functions fill in the low-frequency 'spectral gap' resulting form the admissibility conditions (an admissibility condition for scaling functions requires that their zero-moment be non-zero). Otherwise, filling this spectral gap could only be achieved in the limit of an infinite number of wavelet functions [31]. Therefore, wavelet functions can be seen as band-pass filters while scaling functions correspond to low-pass filters.

Of course, all these constraints are meaningless unless there actually exists a set of scaling and wavelet functions that satisfy the above constraints. The next section introduces the Haar wavelet and its corresponding scaling function. Both functions are probably the most basic of all wavelet/scaling functions.

2.1.2 An Example: The Haar Wavelet

The Haar wavelet, introduced by Alfred Haar in his 1909 thesis, is probably the most basic of all wavelets [8]. The scaling and wavelet functions consist of shifted and scaled versions of a square wave. The functions $\varphi(t)$ and $\psi(t)$ are shown in Fig. 2.1 normalized to the interval $[0, 1)$. The equations for each are

$$\varphi(t) = \begin{cases} 1, & t \in [0,1) \\ 0, & t \notin [0,1) \end{cases} \qquad \psi(t) = \begin{cases} 1, & t \in [0,1/2) \\ -1, & t \in [1/2,1) \\ 0, & t \notin [0,1) \end{cases} \tag{2.17}$$



(a) (b)

Fig. 2.1 Haar (a) scaling and (b) wavelet functions.

2.1.3 Filter Bank/Transform Matrix Interpretation of Wavelets

A consequence of the two MRA equations in (2.4) and (2.15) makes calculating the required scale/wavelet coefficients amenable to a filter bank realization [32]. In fact, only two types of linear time-invariant filters with impulse responses $h_0(-n)$ and $h_1(-n)$ are required, in addition to 2:1 decimators. The output of the cascade connection of $h_1(-n)$ and the decimator at scale j produces the wavelet coefficients d_j, while the output of the cascade of $h_0(-n)$ and the decimator produce the scaling coefficients c_j. From a spectral perspective, the filters $h_0(-n)$ and $h_1(-n)$ correspond to low-pass and high-pass filters, respectively.

By cascading another stage of the above filters at the output c_j, the wavelet and scaling coefficients corresponding to the next highest scale (coarse) can be obtained. These stages can then be added iteratively with each new stage producing wavelet and scaling coefficients at a coarser scale. This bank of filters forms the analysis tree shown in Fig. 2.2. The iterated bank of filters has a very significant spectral interpretation as well. At the first stage, the input signal's spectrum is divided into a low frequency spectrum and a high frequency spectrum. The next stage then similarly divides the low-frequency spectrum into a corresponding high-frequency spectrum and low-frequency spectrum. This is shown graphically in Fig. 2.3.

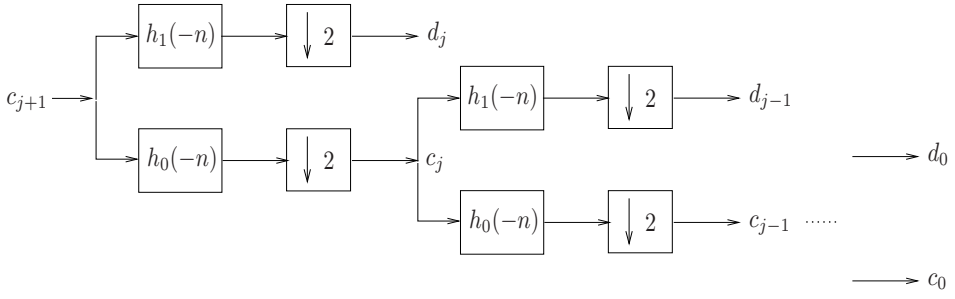


Fig. 2.2 Filter bank implementation of the Wavelet Transform.

One might wonder where the translation index k fits into the filter bank of Fig. 2.2. To understand this, one has to first realize that the above iterated filter bank does not make explicit use of the *continuous-time* functions, $\varphi(t)$ and $\psi(t)$. In fact, this filter bank is calculating the DWT of a discrete-time signal. Recall from (2.4) and (2.15) that the wavelet and scaling coefficients at a certain scale can be computed using

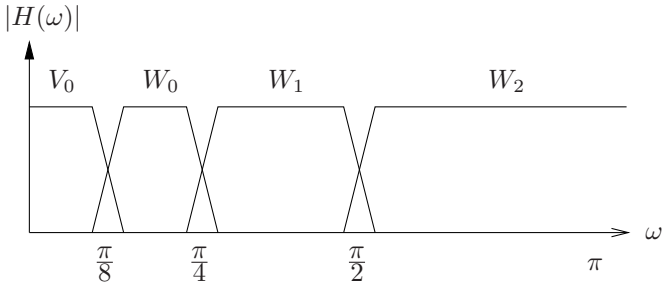


Fig. 2.3 Frequency bands corresponding to a dyadic wavelet decomposition.

the scaling coefficients of an immediate prior scale. The discrete-time input samples can therefore be interpreted as outputs of a prior (non-existential) low-pass filter corresponding to scale $j+1$ (see Fig. 2.2). Returning to the issue of the translation index k, each terminal branch in Fig. 2.2 produces a sequence of wavelet/scaling coefficients corresponding to the different translations k at every j. Of course, for each pair of wavelet coefficients produced at scale j, one wavelet coefficient, is produced at scale $j-1$. This is a result of the 2:1 downsampling operation.

Another method of visualizing the wavelet transform is with the use of an N-by-N transform matrix, whose first set of rows correspond to the scaling basis vector at a predetermined initial scale, j_0. If $j_0 = 0$, the remaining rows consist of sets of rows corresponding to the wavelet basis vector beginning with scale index $j = 0$ up to $j_{\max} - 1 = \log_2 N - 1$. Each subset of rows corresponds to a group of these wavelet basis vectors at a given scale, where each row consists of the scaling/wavelet functions at a different translational index $k = \{0, \ldots, 2^j - 1\}$.

As an example, consider the simple 8-by-8 Haar transform matrix H shown below, where the rows corresponding to a given scale have been partitioned into subsets.

$$\mathbf{H} = 2^{-3/2} \left(\begin{array}{cccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \hline \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ \hline 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{array} \right) \tag{2.18}$$

The number of rows (or vectors) within each subset is equal to 2^j, where the first two rows correspond to the scaling and wavelet basis vectors at scale index $j = 0$. In addition, the vectors in each subset have a time support of length $N/2^j$. It is interesting to note that the orthogonality conditions in (2.12) and (2.13) carry over to the discrete wavelet and scaling basis vectors in (2.18).

Although both filter-bank and matrix interpretations produce the same wavelet coefficients, there is one key difference and it plays an essential role in the coupled echo canceller. When all the discrete-time samples of a length N input signal have been filtered using the iterated filter bank with $\log_2 N$ stages in Fig. 2.2, then one scaling coefficient and $N-1$ wavelet coefficients are produced. However, in the matrix transform interpretation, every new sample shifted into an input vector $\mathbf{x}$ produces a new output vector $\mathbf{Hx}$.

2.1.4 Shift-Variance of the Wavelet Transform

Although wavelets have proven to be a significant tool in many applications, they do suffer from some disadvantages. One of the main disadvantages is their lack of shift invariance. To be more specific, the DWT coefficients of a signal are not simply related to those of the same signal shifted by one sample. This property can be seen by either looking at the matrix representation in (2.18) or at the filter-bank in Fig. 2.2 which includes decimation operations that make the overall process time-variant.

The transform is also shift-variant across subbands (or scale). Shifting the input redistributes the energy across different scales, and this is mainly due to the fact that the wavelet system uses critically sampled subband decompositions, i.e. no redundancy is introduced as a result of violating the Nyquist criterion. This can be achieved because of the way aliasing artifacts are cancelled in the inverse transform [43].

2.2 Least Mean Square Algorithm

This section presents and develops the famous least mean square (LMS) algorithm, originally introduced by Widrow and Hoff [23, 22]. The discussion will focus solely on the system identification problem shown in Fig. 2.4. To begin, some notation is in

order:

n	discrete time index
N	length of input data vector and adaptive filter
$\mathbf{u}(n)$	input data vector (of length N)
$y(n)$	adaptive filter output signal
$d(n)$	desired response
$e(n)$	error signal
$\nu(n)$	measurement/plant noise
$\mathbf{w}(n)$	adaptive weight vector (of length N)
$\mathbf{w}'$	impulse response of unknown system (of length N')

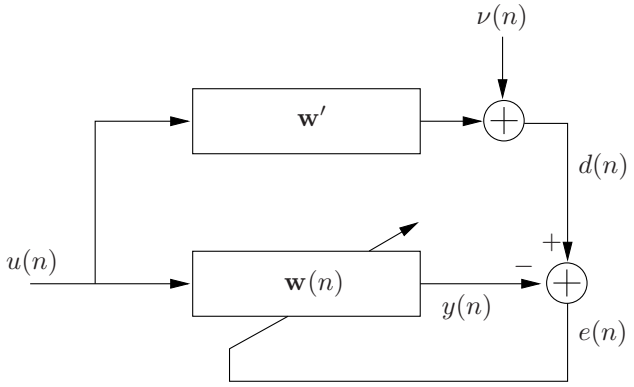


Fig. 2.4 System identification using an adaptive filter.

2.2.1 Wiener-Hopf Equation

The adaptive filter shown in Fig. 2.4 is in the form of a linear combiner or finite impulse response (FIR) filter with an adjustable weight (or coefficient) vector $\mathbf{w}(n)$. For simplicity, it will be assumed that all the data is real-valued, including the input and the tap-weights (filter coefficients). In addition, the unknown weight vector $\mathbf{w}'$ is assumed to be time-invariant. The goal of an adaptive filtering algorithm is to find a weight vector $\mathbf{w}$ so as to minimize a certain cost function. A weight vector that satisfies this condition is termed the 'best' weight vector.

The input data vector at time n is given by

$$\mathbf{u}(n) = [u(n), u(n-1), \ldots, u(n-N+1)]^T, \tag{2.19}$$

and the weight vector is denoted by

$$\mathbf{w} = [w_1, w_2, \ldots, w_N]^T. \tag{2.20}$$

Each input sample is multiplied with its respective tap-weight, and the results are added to produce the output

$$y(n) = \mathbf{w}^T\mathbf{u}(n) = \sum_{k=1}^{N} w_k u(n-k+1). \tag{2.21}$$

The error at time n is

$$e(n) = d(n) - y(n) = d(n) - \mathbf{w}^T\mathbf{u}(n), \tag{2.22}$$

where

$$d(n) = \mathbf{w'}^T\mathbf{u}(n) + \nu(n) \tag{2.23}$$

is the output to the true system (also called the plant) to be identified by the adaptation process plus an additional term that can be attributed to plant or measurement noise denoted by $\nu(n)$, assumed to be independent of $\mathbf{u}(n)$.

Here, the cost function to be minimized by the best weight vector is the mean squared error (MSE),

$$\begin{aligned} J &= E[e^2(n)] \\ &= E[d^2(n) - 2d(n)\mathbf{w}^T\mathbf{u}(n) + \mathbf{w}^T\mathbf{u}(n)\mathbf{u}^T(n)\mathbf{w}] \\ &= \sigma_d^2 - 2\mathbf{w}^T\mathbf{p} + \mathbf{w}^T\mathbf{R}\mathbf{w} \end{aligned} \tag{2.24}$$

where σ_d^2 denotes the variance of the desired response $d(n)$, $\mathbf{p} = E[d(n)\mathbf{u}(n)]$ is the cross-correlation between the input data vector and the desired signal, and $\mathbf{R} = E[\mathbf{u}(n)\mathbf{u}^T(n)]$ is the input autocorrelation matrix. The above equation shows that the MSE is a quadratic function of the tap-weight vector, $\mathbf{w}$. Assuming that $\mathbf{R} > 0$, this implies that $J(\mathbf{w})$ exhibits a unique minimum which can be found by following the negative gradient of $J(\mathbf{w})$ with respect to $\mathbf{w}$. Differentiating (2.24) with respect

to $\mathbf{w}$,

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = -2\mathbf{p} + 2\mathbf{R}\mathbf{w}, \tag{2.25}$$

and setting the result to zero, the famous Wiener-Hopf equation is obtained for the optimal filter weights,

$$\mathbf{w}_o = \mathbf{R}^{-1}\mathbf{p}. \tag{2.26}$$

If $N \geq N'$, it can be shown that the Wiener solution $\mathbf{w}_o$ corresponds to the true weight vector of the unknown system $\mathbf{w}'$ padded with $N-N'$ zeros, i.e. $\mathbf{w}_o = [\mathbf{w}'^T \mathbf{0}]^T$. In the sequel, it will be assumed that this condition is satisfied and the adaptive filter length is sufficient to model $\mathbf{w}'$, so that the symbol $\mathbf{w}'$ can be replaced by $\mathbf{w}_o$.

When $\mathbf{R}$ is nonsingular (as is usually the case), the MSE expression in (2.24) can be written as

$$J(\mathbf{w}) = \sigma_d^2 - \mathbf{p}^T\mathbf{R}^{-1}\mathbf{p} + (\mathbf{w} - \mathbf{R}^{-1}\mathbf{p})^T\mathbf{R}(\mathbf{w} - \mathbf{R}^{-1}\mathbf{p}). \tag{2.27}$$

Therefore,

$$J_{\min} = \min_{\mathbf{w}} J(\mathbf{w}) = \sigma_d^2 - \mathbf{p}^T\mathbf{R}^{-1}\mathbf{p} \tag{2.28}$$

is satisfied for $\mathbf{w} = \mathbf{w}_o = \mathbf{R}^{-1}\mathbf{p}$. Expanding σ_d^2,

$$\begin{aligned} \sigma_d^2 &= E[d^2(n)] \\ &= E[(\mathbf{w}_o^T\mathbf{u}(n) + \nu(n))^2] \\ &= \sigma_\nu^2 + \mathbf{w}_o^T\mathbf{R}\mathbf{w}_o, \end{aligned} \tag{2.29}$$

where σ_ν^2 is the noise power and the independence of $\nu(n)$ and $\mathbf{u}(n)$ has been assumed. Therefore (2.28) becomes

$$\begin{aligned} J_{\min} &= \sigma_\nu^2 + \mathbf{w}_o^T\mathbf{R}\mathbf{w}_o - \mathbf{p}^T\mathbf{R}^{-1}\mathbf{p} \\ &= \sigma_\nu^2. \end{aligned} \tag{2.30}$$

In the ideal case, at least, the minimum MSE reduces to the measurement noise power.

In general then, (2.24) can be written as

$$J(\mathbf{w}) = J_{\min} + (\mathbf{w} - \mathbf{w}_o)^T \mathbf{R}(\mathbf{w} - \mathbf{w}_o). \tag{2.31}$$

2.2.2 LMS Algorithm

The LMS algorithm is based on the steepest-descent algorithm [22],

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \frac{1}{2}\mu \frac{\partial J(n)}{\partial \mathbf{w}(n)}, \tag{2.32}$$

where the tap-weights are now a function of time. Substituting (2.25) in (2.32), one obtains the steepest descent algorithm's update equation,

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu[\mathbf{p} - \mathbf{R}\mathbf{w}(n)]. \tag{2.33}$$

Note that this is a deterministic equation since the quantities $\mathbf{p}$ and $\mathbf{R}$ are produced by expectation operators.

The LMS algorithm on the other hand, approximates $\mathbf{p}$ and $\mathbf{R}$ with their instantaneous estimated values. As a result, the LMS update equation is given by

$$\begin{aligned} \mathbf{w}(n+1) &= \mathbf{w}(n) + \mu[d(n)\mathbf{u}(n) - \mathbf{u}(n)\mathbf{u}^T(n)\mathbf{w}(n)] \\ &= \mathbf{w}(n) + \mu\mathbf{u}(n)[d(n) - \mathbf{u}^T(n)\mathbf{w}(n)] \\ &= \mathbf{w}(n) + \mu\mathbf{u}(n)e(n). \end{aligned} \tag{2.34}$$

Because instantaneous, and therefore random values of the quantities $\mathbf{p}$ and $\mathbf{R}$ are used, the LMS algorithms is a stochastic gradient algorithm.

2.2.3 Properties of the LMS Algorithm

Although the recursive form of the LMS filter in (2.34) is straightforward, its convergence analysis is quite complicated (convergence usually refers to the behaviour of the adaptive filter as it reaches its optimum solution). This can be seen by finding

the closed-form expression of (2.34), assuming $\mathbf{w}(0) = \mathbf{0}$,

$$\mathbf{w}(n) = \mu \sum_{i=0}^{n-1} e(i)\mathbf{u}(i). \tag{2.35}$$

The filter output $y(n)$ can then be written as

$$\begin{aligned} y(n) &= \mathbf{w}^T(n)\mathbf{u}(n) \\ &= \mu \sum_{i=0}^{n-1} e(i)\mathbf{u}^T(i)\mathbf{u}(n), \end{aligned} \tag{2.36}$$

where it is clear that $y(n)$ is a non-linear function of the input vectors $\mathbf{u}(j)$, $(j = 0, 1, \ldots, n)$. The most common form of analysis of the LMS algorithm is based on small step-size theory [22] which approximates the LMS weight error vector $\boldsymbol{\epsilon}(n) = \mathbf{w}_o - \mathbf{w}(n)$, and its corresponding recursion

$$\boldsymbol{\epsilon}(n+1) = [\mathbf{I} - \mu\mathbf{u}(n)\mathbf{u}^T(n)]\boldsymbol{\epsilon}(n) - \mu\mathbf{u}(n)e_o(n), \tag{2.37}$$

with its zero-order recursion

$$\boldsymbol{\epsilon}_0(n+1) = (\mathbf{I} - \mu\mathbf{R})\boldsymbol{\epsilon}_0(n) + \mathbf{f}_0(n), \tag{2.38}$$

where $e_o(n) = d(n) - \mathbf{w}_o^T\mathbf{u}(n)$, is the output error corresponding to the Wiener solution, and $\mathbf{I}$ is an N-by-N identity matrix. The second term in (2.37), $\mathbf{f}_0(n) = -\mu\mathbf{u}(n)e_o(n)$, is called the driving force. The eigendecomposition of the input correlation matrix $\mathbf{R}$ is given by $\mathbf{R} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^H$, where $\boldsymbol{\Lambda}$ is the diagonal matrix of eigenvalues, and $\mathbf{Q}$ is an orthogonal matrix of the respective eigenvectors. The transformed weight error vector $\mathbf{b}(n) = \mathbf{Q}^T\boldsymbol{\epsilon}_0(n)$ consists of scalar entries known as the natural modes of the filter. The transformed recursion in (2.38) becomes

$$\mathbf{b}(n+1) = (\mathbf{I} - \mu\boldsymbol{\Lambda})\mathbf{b}(n) + \boldsymbol{\phi}(n), \tag{2.39}$$

where $\boldsymbol{\phi}(n) = \mathbf{Q}^T\mathbf{f}_0(n)$. If $\mathbf{b}(n+1)$ is decomposed into its individual components, b_k, then the expression for the k^{th} natural mode is,

$$b_k(n+1) \;=\; (1-\mu\lambda_k)b_k(n) + \phi_k(n). \tag{2.40}$$

A closed-form equation can be derived,

$$b_k(n) \;=\; (1-\mu\lambda_k)^n b_k(0) + \sum_{i=0}^{n-1}(1-\mu\lambda_k)^{n-i-1}\phi_k(i), \tag{2.41}$$

whose mean value and mean squared value are

$$E[b_k(n)] \;=\; b_k(0)(1-\mu\lambda_k)^n \tag{2.42}$$

$$E[|b_k(n)|^2] \;=\; \frac{\mu J_{\min}}{2-\mu\lambda_k} + (1-\mu\lambda_k)^{2n}\Big(|b_k(0)|^2 - \frac{\mu J_{\min}}{2-\mu\lambda_k}\Big), \tag{2.43}$$

respectively [22]. From this expression, one can determine the range of values of the step-size μ so that the mean value of the natural modes decay to zero,

$$-1 < 1-\mu\lambda_k < +1, \quad \forall k. \tag{2.44}$$

The corresponding range for μ is

$$0 < \mu < \frac{2}{\lambda_{\max}}, \tag{2.45}$$

where $\lambda_{\max}$ and $\lambda_{\min}$ correspond to the maximum and minimum eigenvalues, respectively. The average time constant of the LMS is a useful indicator of the convergence time of the algorithm. The constant is derived from the average eigenvalue, defined as $\lambda_{\text{av}} = \frac{1}{N}\sum_{k=1}^{N}\lambda_k$, and is given by

$$\tau_{\text{av}} \approx \frac{1}{2\mu\lambda_{\text{av}}}. \tag{2.46}$$

For the steepest descent case, this time constant defines the amount of time it takes on average for a natural mode to decay to $1/e$ of its initial value, $b_k(0)$. In fact, it can

be shown [22] that the time constant τ of the filter ranges from

$$\frac{-1}{\ln(1-\mu\lambda_{\max})} \leq \tau \leq \frac{-1}{\ln(1-\mu\lambda_{\min})}. \tag{2.47}$$

As a result, the convergence time of the LMS algorithm is a function of the eigenspread (also known as the condition number), $\lambda_{\max}/\lambda_{\min}$ of the input autocorrelation $\mathbf{R}$.

To derive an MSE equation similar to (2.31) within the context of the LMS algorithm, $\mathbf{w}$ is replaced by a time-varying form $\mathbf{w}(n)$. Applying the eigenvalue decomposition $\mathbf{R} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^H$, the expected value of the squared error, given $\mathbf{w}(n)$, is

$$E[e^2(n)|\mathbf{w}(n)] = J_{\min} + \sum_{k=1}^{N} \lambda_k |b_k(n)|^2. \tag{2.48}$$

The above equation, however, is stochastic because it depends on the random quantities $b_k(n)$, $(k = 1, 2, \ldots, N)$. In order to obtain an expression for the mean square error, the expectation of (2.48) is taken with respect to $\mathbf{w}(n)$ resulting in the expression

$$\begin{aligned} J(n) &\approx J_{\min} + \mathrm{tr}[\mathbf{R}\mathbf{K}_0(n)] \\ &\approx J_{\min} + \sum_{k=1}^{N} \lambda_k E[|b_k(n)|^2], \end{aligned} \tag{2.49}$$

where $\mathbf{K}_0(n) = E[\boldsymbol{\epsilon}_0(n)\boldsymbol{\epsilon}_0^T(n)]$ is the correlation matrix of the zero-order weight error vector and using (2.43), the steady-state MSE is given by

$$J(\infty) \approx J_{\min}\Big(1 + \frac{\mu}{2}\sum_{k=1}^{N} \lambda_k\Big). \tag{2.50}$$

In the steepest descent case, where the driving force $\boldsymbol{\phi}(n)$ is absent from the natural mode recursion [22], the MSE in (2.48) decays to $J_{\min}$ as the modes decay to zero. However, the stochastic nature of the LMS produces a larger steady-state MSE, as shown in the above equation.

2.2.4 Normalized LMS Algorithm

One of the drawbacks of the LMS algorithm is gradient noise amplification. From (2.34), the direction and magnitude of the tap update vector $\mu\mathbf{u}(n)e(n)$ is proportional to the input data vector $\mathbf{u}(n)$ [22]. Therefore, any large increase in $\mathbf{u}(n)$ causes an undesirable increase in the LMS update equation (2.34).

To overcome this problem, the normalized LMS algorithm (NLMS) normalizes the weight update term in (2.34) by $||\mathbf{u}(n)||^2$,

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu}{\|\mathbf{u}(n)\|^2}\mathbf{u}(n)e(n). \tag{2.51}$$

The above equation satisfies the principle of minimal disturbance [22] which states that the squared Euclidian norm of the tap-weight update vector $\|\boldsymbol{\Delta}\mathbf{w}(n+1)\|^2 = \|\mathbf{w}(n+1) - \mathbf{w}(n)\|^2$ should be minimized, subject to the constraint that the posterior error $\tilde{e}(n) = d(n) - \mathbf{w}^T(n+1)\mathbf{u}(n)$ equals zero.

To analyze the stability of the NLMS, the second moment convergence bounds on the step-size will be used [2]. Unlike the first moment weight-convergence bounds found in (2.45), the second moment bounds on the step size are given by

$$0 < \mu < \frac{2}{\mathrm{tr}(\mathbf{R})}. \tag{2.52}$$

Because of the normalization factor $\|\mathbf{u}(n)\|^2$, the resulting Wiener-Hopf equation has the form,

$$\bar{\mathbf{w}}_o = E\Big[\frac{\mathbf{u}(n)\mathbf{u}^T(n)}{\|\mathbf{u}(n)\|^2}\Big]^{-1} E\Big[\frac{\mathbf{u}(n)d(n)}{\|\mathbf{u}(n)\|^2}\Big] = \bar{\mathbf{R}}^{-1}\bar{\mathbf{p}}, \tag{2.53}$$

where a term $\bar{a}$ denotes the normalized version of that term. As a result, $\mathrm{tr}(\bar{\mathbf{R}}) = 1$ and the second moment convergence bounds for the NLMS algorithm are given by [2]

$$0 < \mu < 2. \tag{2.54}$$

The optimal step-size is usually chosen as $\mu_{\mathrm{opt}} = 1$, and corresponds to the maximum non-oscillatory decay of the natural modes in (2.42).

Chapter 3

Coupled Echo Cancellation

This chapter focuses on the coupled echo canceller proposed by Bershad and Bist [4]. The first section introduces the echo canceller, and establishes formulae for its update equations and convergence properties. The second section provides an in depth analysis of the echo canceller, targeting its major weaknesses. The last section then analyzes some of the implementation issues associated with the coupled echo canceller.

3.1 Coupled Adaptation of Bershad and Bist

As mentioned in the introduction, recent sparse adaptive algorithms such as the PNLMS and its variants exploit the sparse characteristic of network echo paths by making modifications to the tap-weight update equations of the NLMS filter. Although these algorithms are designed to operate optimally in sparse environments[1], they still require that all filter coefficients be updated. Bershad and Bist have proposed a coupled setup using two short adaptive filters operating in parallel [4]. The first filter operates in the transform domain on only a subset of input Haar coefficients, and is used by a peak delay estimator to determine the location of the channel's dispersive region. A second time-domain filter is centred around this location to actually cancel the echo.

[1] In fact, as the number of non-zero filter coefficients increases, the PNLMS converges slower than the NLMS [3].

There are numerous advantages to using shorter filters. These include faster convergence since fewer parameters (in this case filter coefficients) need to be estimated, and a reduction in computational and memory requirements. As a result, the coupled echo canceller makes use of only two short adaptive filters, at the additional cost of computing the partial Haar transform of the input.

3.1.1 Structure of the Partial Haar Dual Adaptive Filter

The structure of the coupled echo canceller (also known as the partial Haar dual adaptive filter) is shown in Fig. 3.1. The upper branch, consists of a partial Haar transform matrix denoted by $\mathbf{H}_q$ of size $q \times N$ and a length-q ($\leq N$) partial Haar adaptive filter. The term partial Haar reflects the fact that the transform only consists of a subset of Haar basis vectors corresponding to a scale index j, where $q = 2^j$. These basis vectors span the sample interval of length N, which is set to match the maximum length of the unknown echo path impulse response. At time sample n, a new sample $u(n)$ is shifted into the input data vector (2.19) of length N and a new transformed input vector $\mathbf{z}(n) = \mathbf{H}_q\mathbf{u}(n)$ of length q is calculated. The partial Haar adaptive filter

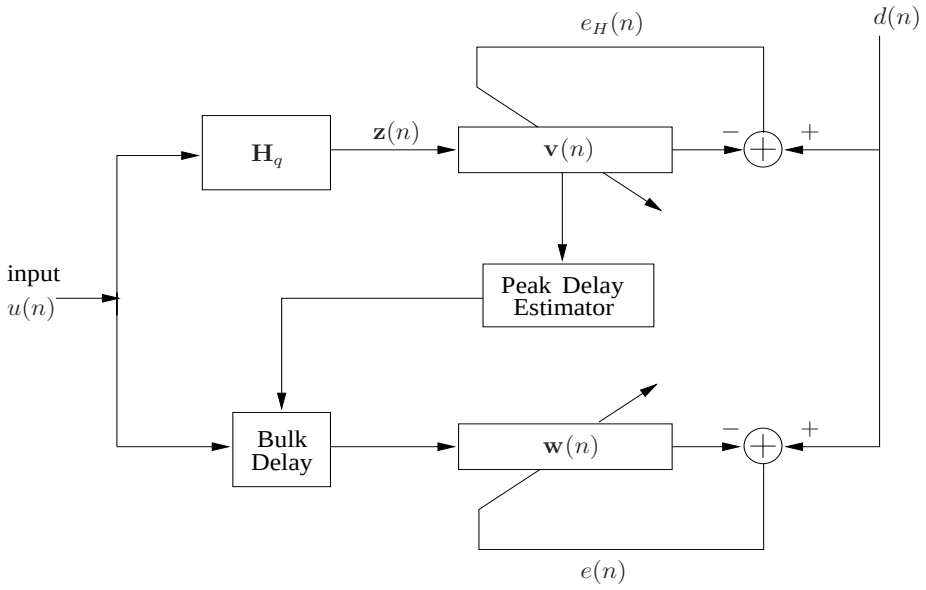


Fig. 3.1 Coupled echo canceller.

$\mathbf{v}(n)$ of length q is a scaled-down version of the full echo path impulse response, and

is used solely to track changes in the echo path impulse response. The peak delay estimator tracks the location of the dispersive region by locating the peak magnitude of the partial Haar impulse response.

In the lower branch of the echo canceller, the estimated location of the dispersive region is used to offset (using an appropriate bulk delay) a short time-domain filter $\mathbf{w}(n)$ of length L so that it is properly centred around the dispersive region. The length of the time-domain filter is set to match the longest expected dispersive region in the echo path.

3.1.2 Effect of the Complete Haar Transform on the Wiener Solution

This subsection deals with the effect of the complete Haar transform on the Wiener solution. Let $\mathbf{H}_N$ denote the orthogonal $N \times N$ Haar transform matrix, i.e. $\mathbf{H}_N^T\mathbf{H}_N = \mathbf{H}_N\mathbf{H}_N^T = \mathbf{I}_N$, and $\mathbf{z}(n) = \mathbf{H}_N\mathbf{u}(n)$ is the transformed input vector of length N. The optimal Haar weight vector is denoted by $\mathbf{v}_o$ (in this case, the length of $\mathbf{v}_o$ is N). The resulting Wiener-Hopf equation is given by

$$\mathbf{R}_{zz}\mathbf{v}_o = \mathbf{p}_z, \tag{3.1}$$

where $\mathbf{R}_{zz} = E[\mathbf{z}(n)\mathbf{z}^T(n)] = \mathbf{H}_N\mathbf{R}_{uu}\mathbf{H}_N^T$ represents the autocorrelation matrix of $\mathbf{z}(n)$, and $\mathbf{p}_z = E[\mathbf{z}(n)d(n)] = \mathbf{H}_N\mathbf{p}_u$ is the cross-correlation vector between $\mathbf{z}(n)$ and the desired signal $d(n)$. Solving for $\mathbf{v}_o$,

$$\begin{aligned}
\mathbf{v}_o &= \mathbf{R}_{zz}^{-1}\mathbf{p}_z \\
&= \{\mathbf{H}_N\mathbf{R}_{uu}\mathbf{H}_N^T\}^{-1}\mathbf{H}_N\mathbf{p}_u \\
&= \mathbf{H}_N\mathbf{R}_{uu}^{-1}\mathbf{H}_N^T\mathbf{H}_N\mathbf{p}_u \\
&= \mathbf{H}_N\left[\mathbf{R}_{uu}^{-1}\mathbf{p}_u\right] \\
&= \mathbf{H}_N\mathbf{w}_o.
\end{aligned} \tag{3.2}$$

As a consequence of the orthogonality property of the Haar transform matrix, the Wiener solution corresponding to the transformed input data vector is simply the Haar transform of the time-domain Wiener solution. In fact, it is possible to recover the optimal time-domain Wiener solution by pre-multiplying (3.2) by $\mathbf{H}_N^T$.

3.1.3 Effect of the Partial Haar Transform on the Wiener Solution

It was seen in the previous section that by transforming the input with the set of Haar basis vectors, the Wiener solution is also transformed. When only a subset of Haar basis vectors at a scale index j is used to transform the input, the situation is somewhat different. Unlike the complete Haar transform matrix, $\mathbf{H}_q$ $(q < N)$ is only row-wise orthogonal, i.e. $\mathbf{H}_q\mathbf{H}_q^T = \mathbf{I}_q \neq \mathbf{H}_q^T\mathbf{H}_q$.

The terms $\mathbf{R}_{zz}$ and $\mathbf{p}_z$ have the same interpretation as in the previous section, except all quantities are now based on the partially transformed input $\mathbf{z}(n) = \mathbf{H}_q\mathbf{u}(n)$ $(q < N)$. The resulting Wiener-Hopf equation becomes

$$\begin{aligned}
\mathbf{v}_o &= \mathbf{R}_{zz}^{-1}\mathbf{p}_z \\
&= \{E[\mathbf{z}(n)\mathbf{z}^T(n)]\}^{-1}E[\mathbf{z}(n)d(n)] \\
&= \{E[\mathbf{H}_q\mathbf{u}(n)\mathbf{u}^T(n)\mathbf{H}_q^T]\}^{-1}E[\mathbf{H}_q\mathbf{u}(n)d(n)] \\
&= \{\mathbf{H}_q\mathbf{R}_{uu}\mathbf{H}_q^T\}^{-1}\mathbf{H}_q\mathbf{p}_u. \qquad (3.3)
\end{aligned}$$

The above equation cannot be simplified any further, mainly because the partial Haar transform matrix is only row-wise orthogonal. If one assumes, however, that the input is white with $\mathbf{R}_{uu} = \sigma_u^2\mathbf{I}_N$, then

$$\begin{aligned}
\mathbf{v}_o &= \{\mathbf{H}_q\sigma_u^2\mathbf{H}_q^T\}^{-1}\mathbf{H}_q\mathbf{p}_u \\
&= \mathbf{H}_q\mathbf{p}_u/\sigma_u^2 \\
&= \mathbf{H}_q\mathbf{w}_o. \qquad (3.4)
\end{aligned}$$

The result is similar to the complete Haar transform, provided the input samples are uncorrelated. However, it is not possible to recover the original Wiener solution. In addition, it is interesting to observe that no transformation of the desired signal $d(n)$ is necessary.

Now that the effect of the partial Haar transform on the Wiener solution has been investigated, consider its effect on the minimum mean square error $J_{\min}$, which is calculated by taking the expected value of the minimum squared error, $e_{Ho}(n)$, of the

partial Haar adaptive filter,

$$\begin{aligned} J_{\min} &= E[e_{Ho}^2(n)] \\ &= E[\big(d(n) - \mathbf{v}_o^T\mathbf{z}(n)\big)^2] \\ &= E[\big(\mathbf{w}_o^T\mathbf{u}(n) + \nu(n) - \mathbf{v}_o^T\mathbf{z}(n)\big)^2] \\ &= E[\big(\nu(n) + \mathbf{w}_o^T\mathbf{u}(n) - \mathbf{w}_o^T\mathbf{H}_q^T\mathbf{H}_q\mathbf{u}(n)\big)^2] \\ &= E[\big(\nu(n) + \mathbf{w}_o^T\big[\mathbf{I}_N - \mathbf{H}_q^T\mathbf{H}_q\big]\mathbf{u}(n)\big)^2], \end{aligned} \tag{3.5}$$

Assuming the input $\mathbf{u}(n)$ is white and uncorrelated with the measurement noise $\nu(n)$,

$$\begin{aligned} J_{\min} &= \sigma_\nu^2 + \sigma_u^2\mathbf{w}_o^T(\mathbf{I}_N - \mathbf{H}_q^T\mathbf{H}_q)(\mathbf{I}_N - \mathbf{H}_q^T\mathbf{H}_q)^T\mathbf{w}_o \\ &= \sigma_\nu^2 + \sigma_u^2\mathbf{w}_o^T(\mathbf{I}_N - \mathbf{H}_q^T\mathbf{H}_q)\mathbf{w}_o \\ &= \sigma_\nu^2 + \sigma_u^2(\|\mathbf{w}_o\|_2^2 - \|\mathbf{v}_o\|^2). \end{aligned} \tag{3.6}$$

Unlike the optimal steepest descent case in (2.30), the minimum MSE of the partial Haar filter contains the additional term $\sigma_u^2(\|\mathbf{w}_o\|^2 - \|\mathbf{v}_o\|^2) \geq 0$. Intuitively, this makes sense, since the partial Haar transform spans only a subspace of the N-dimensional space, and so the resulting *excess* MSE can be attributed to the partial Haar transform's rank deficiency, i.e. to those basis vectors not included in the partial transform. Note that when $q = N$, the second term on the righthand side of (3.6) vanishes.

Bershad and Bist [4] argue that this increase in minimum MSE is not very significant since the partial Haar-domain filter is only being used to locate a dispersive region. However, as will be seen in Section 3.2 this can deteriorate the echo canceller's convergence performance.

3.1.4 Transient Behaviour of the Partial Haar Adaptive Filter

For the following and subsequent sections, six assumptions made by [4] are used to simplify the analysis. These include the Independence Theory assumptions,

A1. The sequence of tap input vectors $\mathbf{u}(n)$, $\mathbf{u}(n-1)$,..., are independent.

A2. $\mathbf{u}(n)$ is independent of $d(k)$ for $k < n$.

A3. $d(n)$ depends on $\mathbf{u}(n)$ but not on $d(k)$, for $k < n$.

Additional assumptions are used to simplify the analysis,

A4. The sequences of input samples $u(k)$ and desired responses $d(m)$ are zero-mean jointly Gaussian.

A5. $\mathbf{u}(n)$ is stationary and white, i.e. $E[\mathbf{u}(n)\mathbf{u}^T(n)] = \sigma_u^2\mathbf{I}_N$.

Finally, to make the analyze of the peak delay estimator easier, it is assumed that

A6. The partial Haar weight vector, $\mathbf{v}(n)$, is a jointly Gaussian random vector.

The partial Haar LMS weight recursion formula adapted from (2.34) is given by

$$\mathbf{v}(n+1) = \mathbf{v}(n) + \mu e_H(n)\mathbf{z}(n) \tag{3.7}$$

where

$$e_H(n) = d(n) - \mathbf{z}^T(n)\mathbf{v}(n). \tag{3.8}$$

Replacing (3.8) into (3.7) yields

$$\mathbf{v}(n+1) = [\mathbf{I}_q - \mu\mathbf{z}(n)\mathbf{z}^T(n)]\mathbf{v}(n) + \mu d(n)\mathbf{z}(n), \tag{3.9}$$

Taking the expectation of both sides of (3.9), and using A1 and A5 along with (3.3),

$$E[\mathbf{v}(n+1)] = [1 - \mu\sigma_u^2]E[\mathbf{v}(n)] + \mu\sigma_u^2\mathbf{v}_o. \tag{3.10}$$

The closed-form expression for the solution of the above equation is

$$E[\mathbf{v}(n)] = (1 - \mu\sigma_u^2)^n\mathbf{v}(0) + [1 - (1 - \mu\sigma_u^2)^n]\mathbf{v}_o. \tag{3.11}$$

Therefore as n increases and provided that $0 < \mu < 2/\sigma_u^2$, the expected weight vector approaches the optimal partial Haar weight vector $\mathbf{v}_o$.

Next, the expression for the weight vector covariance matrix $Q(n)$ is obtained so as to characterize the correlation between the jointly Gaussian tap-weights (from A6):

$$\begin{aligned} Q(n) &= E[\mathbf{v}(n)\mathbf{v}^T(n)] - E[\mathbf{v}(n)]E[\mathbf{v}^T(n)] \\ &= E[\mathbf{v}(n)\mathbf{v}^T(n)] - [1-(1-\mu\sigma_u^2)^n]^2\mathbf{v}_o\mathbf{v}_o^T, \end{aligned} \tag{3.12}$$

where it has been assumed that $\mathbf{v}(0) = \mathbf{0}$. It is shown in [4] that

$$Q(n) = [\alpha(n)+\gamma(n)]\mathbf{I}_q + [\beta(n)-(1-\mu\sigma_u^2)^{2n}]\mathbf{v}_o\mathbf{v}_o^T, \tag{3.13}$$

where

$$\rho(n) = (1-2\mu\sigma_u^2+\mu^2\sigma_u^4(q+2))^n \tag{3.14}$$

$$\alpha(n) = \left(\frac{\mu J_{min}}{2-\mu\sigma_u^2(q+2)}\right)[1-\rho(n)] \tag{3.15}$$

$$\beta(n) = (1-2\mu\sigma_u^2+2\mu^2\sigma_u^4)^n \tag{3.16}$$

$$\gamma(n) = \left(\frac{\mathbf{v}_o^T\mathbf{v}_o}{q}\right)[\rho(n)-(1-2\mu\sigma_u^2+2\mu^2\sigma_u^2)^n]. \tag{3.17}$$

For large n and $0 < \mu < 2/(q+2)\sigma_u^2$, the covariance matrix converges to the first term on the righthand side of (3.13),

$$Q(n) \approx [\alpha(n)+\gamma(n)]\mathbf{I}_q, \tag{3.18}$$

and the tap weights of the adaptive filter are uncorrelated. The above result simplifies the analysis of the peak delay estimator in the next section.

3.1.5 Peak Delay Estimator

The peak delay estimator plays a central role in the coupled echo canceller. It estimates the location of the dispersive region, allowing the short-time domain filter to be centred around it. Although the short time-domain filter can tolerate a small degree of jitter around a true peak location, false peak detections can cause the greatest convergence problems for the short-time adaptive filter.

By A6, the partial Haar-domain tap weights are assumed to be independent Gaus-

sian variables with probability density function (PDF)

$$\phi(v) = \frac{1}{\sqrt{2\pi}\sigma_w} e^{-v^2/2\sigma_w^2}. \tag{3.19}$$

The analysis by Bershad and Bist [4] further assumes that all the filter coefficients have variance σ_w^2 (from (3.18))) and zero mean, except for the tap weight corresponding to the peak which has a mean value of m. Note that both m and σ_w^2 are functions of time. This weight is assumed to be Gaussian with a non-zero mean value corresponding to the peak's tap-weight value. Let v_i correspond to the peak tap weight, so the remaining weights are $v_1, v_2, \ldots, v_{i-1}, v_{i+1}, \ldots, v_q$. The PDF can then be divided into two groups,

$$f(v) = \begin{cases} \phi(v), & l \neq i, \quad 1 \leq l \leq q \\ \phi(v-m), & l = i \end{cases} \tag{3.20}$$

Given that the value of the peak is Z_i, the probability that the absolute value of this peak is greater than the remaining filter coefficients is
$P\{|v_i| > |v_l| \text{ for all } l \in \{1, 2, \ldots, q\}, l \neq i | v_i = Z_i\}$

$$\begin{aligned} &= \prod_{\substack{l=1 \\ l \neq i}} P\{|v_i| > |v_l| | v_i = Z_i\} \\ &= \prod_{\substack{l=1 \\ l \neq i}}^{q} \int_{-Z_i}^{Z_i} \phi(v) dv \\ &= \left[\int_{-Z_i}^{Z_i} \phi(v) dv \right]^{q-1} \\ &= \left[\frac{1}{\sqrt{2\pi}\sigma_w} \int_{-Z_i}^{Z_i} \exp\left(-\frac{v^2}{2\sigma_w^2} \right) dv \right]^{q-1}. \end{aligned} \tag{3.21}$$

The probability of correct detection then becomes (integrating the expression in (3.21) over the PDF corresponding to Z_i)
$P\{|v_i| > |v_l| \text{ for all } l \in \{1, 2, \ldots, q\}, l \neq i\}$

$$= \frac{1}{\sqrt{2\pi}\sigma_w} \int_{-\infty}^{\infty} \exp\left(-\frac{(Z_i - m)^2}{2\sigma_w^2} \right) \left[\frac{1}{\sqrt{2\pi}\sigma_w} \int_{-Z_i}^{Z_i} \exp\left(-\frac{v}{2\sigma_w^2} \right) dv \right]^{q-1} dZ_i. \tag{3.22}$$

3.2 Critical Analysis of the Haar-domain Adaptive Filter

This section provides an in depth look at the significant drawbacks of the coupled echo canceller [4]. These include its dependence on the echo path's bulk delay, the problem of tracking dispersive regions in non-stationary environments, and the lack of provision for locating multiple dispersive regions.

3.2.1 Effect of the Bulk Delay on the Partial Haar Impulse Response

In Section 3.1.3 the effect of transforming a white input signal (prior to filtering) with the partial Haar transform on the optimal Wiener solution was studied. It was shown that the resulting Wiener solution is simply the partial Haar transform of the time-domain Wiener solution. Recall from Section 2.1, however, that one of the main disadvantages of wavelet transforms is their lack of shift invariance. In other words, depending on the bulk delay of the dispersive region, the transformed Wiener solution can be quite different, changing the peak delay estimator's performance.

To see why the wavelet transform is not shift invariant, consider the signal

$$x(n) = h_{j,1}(n). \tag{3.23}$$

where $h_{j,1}(n)$ represents the second row (translation index starts at $k = 0$) of the $2^j \times N$ submatrix of the complete Haar transform matrix corresponding to a scale index j. Let $N = 32$ and $j = 3$ (or $q = 2^3 = 8$), so that the non-zero time-support of each $h_{3,k}(n)$ (for $k = 0, 1, \ldots, 2^j - 1$) is $N/q = 4$ samples. The transform coefficients obtained by projecting the signal $x(n - l)$ onto this subset of basis vectors is denoted by $d_j^l(k)$ for $l = \{0, 1, \ldots, N/q - 1\}$. Figure 3.2 shows each resulting set of transform coefficients $d_3^l(k)$ for $0 \leq k, l \leq 3$.

One can note that in the case of $d_3^0(k)$, the single non-zero coefficient occurs at $k = 1$. This is expected since $x(n)$ equals the basis vector $h_{3,1}(n)$, and so $d_3^0(1) = \|x(n)\|^2$. In addition, because the set of basis vectors are orthonormal, the resulting norm is unity. If $l = 1$, the time-support of $x(n - 1)$ is no longer restricted to a single basis vector. Instead, the last sample of $x(n - 1)$ leaks into the non-zero time-support of the nearby vector $h_{j,2}(n)$. As a result, the energy of $x(n)$ is now distributed over two wavelet coefficients and so both $d_3^1(1)$ and $d_3^1(2)$ are non-zero in this case.

Furthermore, by Parseval's relation (2.14), $|d_3^1(1)|^2, |d_3^1(2)|^2 < 1$. In other words, the peak magnitude for this case is smaller than for $d_3^0(k)$.

As l increases, the energy of $x(n-l)$ is distributed across both the second and third wavelet basis vectors, $h_{3,1}$ and $h_{3,2}$. When $l = 4$, $x(n-4) = h_{j,2}(n)$, the subsequent observations for $l = \{4, 5, 6, 7\}$ are identical to the case for $l = \{1, 2, 3, 4\}$ in Fig. 3.2. In other words, although the DWT is not strictly shift invariant [8, 43], it is periodically invariant with a period of N/q.

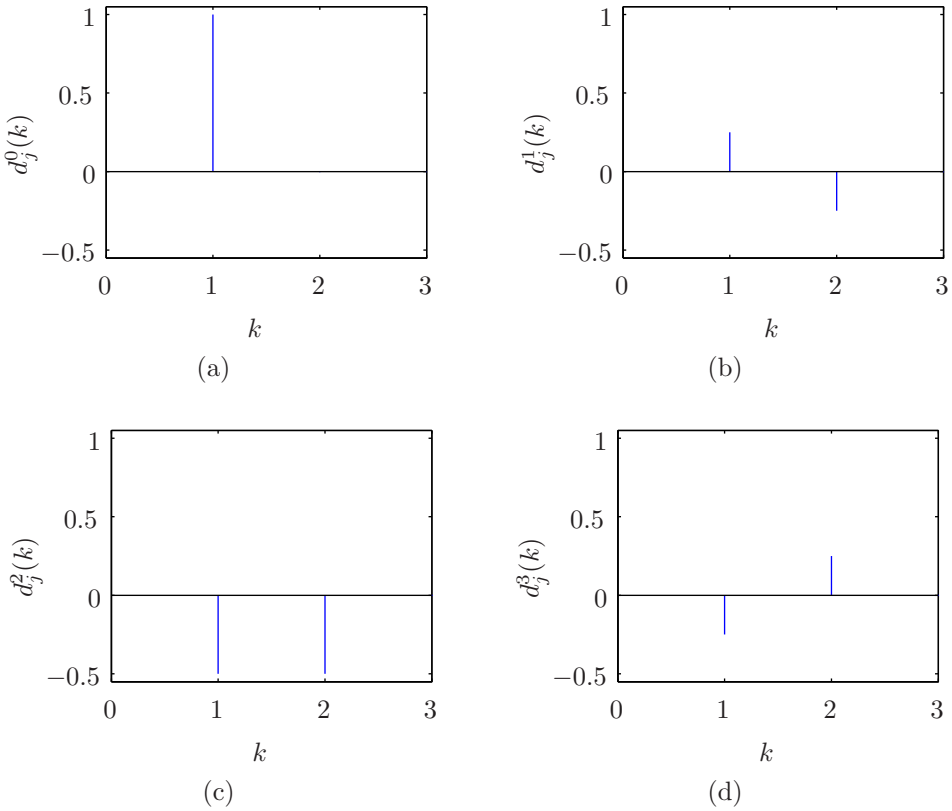


Fig. 3.2 Examining the lack of shift invariance of the wavelet transform for $N = 32$, $q = 8$, corresponding to (a) $l = 0$ (b) 1 (c) 2 (d) 3 sample(s).

From the above observations, one can conclude that for a given echo path impulse response, the partial Haar Wiener solution depends on the bulk delay. In addition, it turns out that for $q = 2^j$, there actually exist N/q *unique* partial Haar transformed impulse responses. Furthermore, the respective peak magnitudes of each of these transformed impulse responses can be different, and consequently the amount of time

it takes the peak delay estimator to correctly estimate the location of a dispersive region varies. This can significantly affect the overall convergence speed of the coupled echo canceller. For example, consider the set of $N/q = 1024/256$ partial Haar transforms of an echo path impulse response containing an ITU-T G.168 hybrid impulse response $m_5(n)$ in Fig. 3.3. One can clearly observe that the third case displays

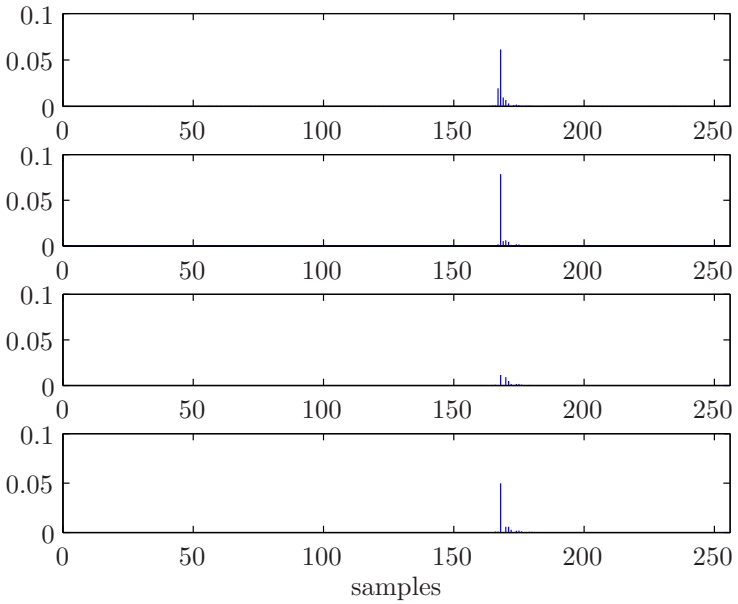


Fig. 3.3 Set of transformed impulse responses corresponding to $m_5(n)$.

a diminished peak. In addition, as a result of measurement noise and the effect of the partial Haar transform on the minimum MSE in (3.6), the peak delay estimator takes a longer time to locate small peaks. To support this claim, consider the family of CDF curves in Fig. 3.4 corresponding to the time to correctly locate the dispersive region for four different bulk delays (each curve is identified by its peak magnitude). It is apparent from this figure that it takes the peak delay estimator a larger number of input samples to correctly locate a smaller peak. For example, for the curve corresponding to a peak magnitude of 0.0166, the peak delay estimator can barely locate the peak after 10^3 input samples. An algorithm will be proposed in the next chapter to escape such suboptimal cases.

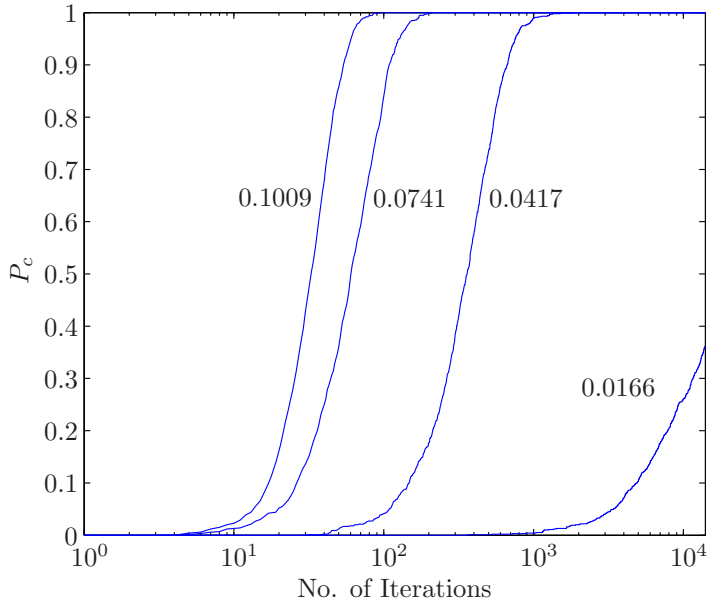


Fig. 3.4 Probability of correctly (P_c) estimating peak delays.

3.2.2 Assessing Tracking Capability

Bershad and Bist [4] prove the merits of using their coupled configuration by calculating the echo canceller's mean time to correctly locate a dispersive region. It is shown on average that this time does not exceed more than 200 samples at 8 kHz when employing a partial Haar NLMS algorithm. These results, as promising as they may appear, only represent the case when the partial Haar adaptive filter has been initialized to zero, i.e. $\mathbf{v}(0) = \mathbf{0}$, and the echo path is stationary. They do not consider what happens when the echo path impulse response abruptly changes during the course of the echo canceller's steady-state operation. The latter case will be considered in this section.

To begin, consider the number of samples required for the average filter coefficient to converge to its Wiener solution. The analysis is similar to [12]. Taking the expected value on both sides of the LMS update equation in (3.7),

$$E[\mathbf{v}(n+1)] = E[\mathbf{v}(n)] + \mu[\mathbf{p}_z - \mathbf{R}_{zz}\mathbf{v}(n)], \tag{3.24}$$

where $\mathbf{p}_z = \mathbf{R}_{zz}\mathbf{v}_o$. Replacing $\mathbf{p}_z$ with $\mathbf{R}_{zz}\mathbf{v}_o$, and solving for the mean weight error

vector $E[\mathbf{v}(n+1)] - \mathbf{v}_o$,

$$E[\mathbf{v}(n+1)] - \mathbf{v}_o = (\mathbf{I}_q - \mu\mathbf{R}_{zz})(E[\mathbf{v}(n)] - \mathbf{v}_o). \tag{3.25}$$

Then the closed-form expression for $E[\mathbf{v}(n)] - \mathbf{v}_o$ is

$$E[\mathbf{v}(n)] - \mathbf{v}_o = (\mathbf{I}_q - \mu\mathbf{R}_{zz})^n(\mathbf{v}(0) - \mathbf{v}_o). \tag{3.26}$$

Assuming $\mathbf{v}(0) = \mathbf{0}$, the above can be rewritten as

$$E[\mathbf{v}(n)] = [\mathbf{I}_q - (\mathbf{I}_q - \mu\mathbf{R}_{zz})^n]\mathbf{v}_o. \tag{3.27}$$

If the input $u(n)$ is white, i.e. $E[\mathbf{u}(n)\mathbf{u}^T(n)] = \sigma_u^2\mathbf{I}_N$, then $\mathbf{R}_{zz} = E[\mathbf{z}(n)\mathbf{z}^T(n)] = \mathbf{H}_q E[\mathbf{u}(n)\mathbf{u}^T(n)]\mathbf{H}_q^T = \sigma_u^2\mathbf{I}_q$ (recall from subsection 3.1.3 that $\mathbf{H}_q\mathbf{H}_q^T = \mathbf{I}_q$), and

$$E[\mathbf{v}(n)] = [1 - (1 - \mu\sigma_u^2)^n]\mathbf{v}_o. \tag{3.28}$$

For the above equation to converge in the mean, the range of $\mu\sigma_u^2$ is constrained to the open interval $(0, 2)$. However, only the subinterval $(0, 1)$ will be considered here since it represents a monotonic rather than oscillatory convergence of the filter tap weights.

The convergence of the mean l^{th} tap weight $\mathrm{E}[v_l(n)]$ $(1 \leq l \leq q)$ to a threshold γ_l $(0 < \gamma_l < |v_{ol}|)$ can be expressed as

$$|E[v_l(n)]| = |1 - (1 - \mu\sigma_u^2)^n||v_{ol}| \geq \gamma_l. \tag{3.29}$$

Since only the range $(0, 1)$ of $\mu\sigma_u^2$ is being considered, it follows that $|1-(1-\mu\sigma_u^2)^n| = 1 - (1 - \mu\sigma_u^2)^n$ for all n, and so (3.29) can be rewritten as

$$(1 - \mu\sigma_u^2)^n \leq 1 - \gamma_l/|v_{ol}|. \tag{3.30}$$

Solving for the smallest integer value that satisfies (3.30),

$$n = \left\lceil \frac{\ln\left[|v_{ol}|/(|v_{ol}| - \gamma_l)\right]}{\ln\left(1 - \mu\sigma_u^2\right)^{-1}} \right\rceil. \tag{3.31}$$

Because the denominator in (3.31) is fixed, the overall number of samples required for the l^{th} filter coefficient to converge depends on the difference $|v_{ol}| - \gamma_l$. Therefore, as this difference decreases (or as γ_l approaches $|v_{ol}|$), n increases. In addition, if the difference is kept constant for all filter taps, i.e. $\epsilon = |v_{ol}| - \gamma_l$, then n only depends on the magnitude $|v_{ol}|$. In this case, the larger the optimal magnitude, the longer it takes the filter coefficient to converge. Consequently, for small ϵ, the overall convergence time of the adaptive filter is equal the longest coefficient convergence time.

One might argue that by using a relative threshold defined as $\gamma_l = |v_{ol}|(1-\epsilon)$, the dependence of (3.31) on $|v_{ol}|$ is lost. In other words, this measure leads one to conclude that the convergence times of all filter coefficients are equal. However, considering the difference $|v_{ol}| - |v_{ol}|(1-\epsilon) = |v_{ol}|\epsilon$, one realizes that this value depends on $|v_{ol}|$. Therefore, the definition of convergence here will be related to an absolute threshold.

Because this thesis is concerned with sparse impulse responses, most filter coefficients will be zero, and therefore, it is more interesting to comment on the behaviour of the maximum peak magnitude. Consider, therefore, a maximum peak magnitude at index i, i.e. ($|v_{oi}| > |v_{ol}|$ for all $l \in \{1, 2, \ldots, q\}, l \neq i$). Let $\gamma_o < |v_{oi}|$ denote a threshold that lies in a small vicinity of $|v_{oi}|$, and let n_o denote the minimum number of samples required for $E[v_i(n)]$ to exceed γ_o. Similarly, let γ_m denote the minimum threshold such that $\gamma_m > |v_{ol}|$ for all $l \neq i$, and n_m is the number of samples required for $E[v_i(n)]$ to exceed γ_m. Since $\gamma_m \leq \gamma_o$, one can conclude from (3.31) that this implies $n_m \leq n_o$.

Now consider a non-stationary channel, where the echo path impulse response has remained constant for some time, the partial Haar adaptive filter has converged, and the current dispersive region is found. Initially, the time for locating the dispersive region will correspond to the case in the previous paragraph. However, what happens when there is an abrupt change in the echo path impulse response, particularly with respect to the bulk delay? In this case, the mean time to locate the new dispersive region can be longer, and is actually related to the magnitude of the old peak.

To be more specific, consider the following example: at time $n = n'$, the echo path impulse response abruptly changes from $\mathbf{v}_{o1}$ to $\mathbf{v}_{o2}$, with peak magnitudes $|v_{oj}^1|$ and $|v_{ok}^2|$, respectively, where $|v_{ok}^2| = \beta|v_{oj}^1|$ ($\beta > 0$) and $j \neq k$, i.e. the bulk delay has changed. The analysis will proceed using (3.26) as a starting point. For the j^{th} filter

tap is

$$E[v_j(n+n')] - v_{oj}^2 = (1-\mu\sigma_u^2)^n (v_j(n') - v_{oj}^2). \tag{3.32}$$

In this case, $v_j(n') \approx v_{oj}^1$ and the optimal tap weight $v_{oj}^2 = 0$, and so the above equation becomes

$$E[v_j(n+n')] = (1-\mu\sigma_u^2)^n v_{oj}^1. \tag{3.33}$$

For the k^{th} filter tap, $v_k(n') = 0$, and so its weight error update equation becomes

$$E[v_k(n+n')] = [1-(1-\mu\sigma_u^2)^n] v_{ok}^2. \tag{3.34}$$

The time to properly estimate the new peak position corresponding to $\mathbf{v}_{o2}$ is equal to the time required for new mean peak magnitude to just exceed the old mean peak magnitude, i.e. when $E[v_j(n+n')] \leq E[v_k(n+n')]$. Therefore, unlike the case considered by [4], where the peak only has to compete with neighbouring coefficient noise, in non-stationary environments, new peaks also have to compete with the presence of older peaks. The time at which the peak detector begins tracking the new peak can then be found

$$\begin{aligned} E[v_j(n+n')] &\leq E[v_k(n+n')] \\ (1-\mu\sigma_u^2)^n |v_{oj}^1| &\leq [1-(1-\mu\sigma_u^2)^n] |v_{ok}^2| \\ (1-\mu\sigma_u^2)^n |v_{oj}^1| &\leq \beta[1-(1-\mu\sigma_u^2)^n] |v_{oj}^1| \\ n &= \left\lceil \frac{\ln(1+1/\beta)}{\ln(1-\mu\sigma_u^2)^{-1}} \right\rceil. \end{aligned} \tag{3.35}$$

The number of samples required for the initial peak $v_j(n)$ to exceed a threshold γ_m, such that $v_j(n)$ is characterized as the global maximum can be found by replacing l with j and γ_k with γ_m in (3.31). Then comparing (3.35) and (3.31), the number of samples required by the new peak magnitude to exceed the old peak is greater than the initial time to locate the old peak when

$$1 + 1/\beta \geq |v_{oj}^1| / (|v_{oj}^1| - \gamma_m) \tag{3.36}$$

or

$$\beta \leq (|v_{oj}^1| - \gamma_m)/\gamma_m. \tag{3.37}$$

In other words, if the ratio between the magnitudes of the new peak and old peak (β) is less than or equal to the relative difference between the old peak magnitude, $|v_{oj}^1|$, and γ_m, then the time to locate a new peak after an abrupt change in impulse response is greater than or equal to locating the old peak initially with $\mathbf{v}(0) = \mathbf{0}$. This is often the case in practice.

The question remains, then, as to whether there is a way to speed up the tracking ability of the coupled configuration in non-stationary environments. Fortunately, the answer is yes. In Section 4.4, an algorithm is developed that provides significant improvements in tracking speed.

3.2.3 Cancelling Multiple Echoes

The peak delay estimator analyzed in [4] assumes the presence of a single dispersive region. This greatly simplifies the estimation process because the adaptive filter coefficient associated with a global maximum directly indicates the location of this region. In its present form, the coupled echo canceller of Bershad and Bist [4] does not support multiple echoes, and this section will look at the main requirements necessary to make it possible.

A major difficulty arises from the fact that knowledge about the number of dispersive regions in a channel is usually unavailable *a priori*. Therefore, in a multiple peak delay estimation scenario, the task not only involves finding local maxima from the partial Haar filter, but also requires the classification of these maxima as peaks. In addition, assumptions on the maximum number of dispersive regions will need to be made to make the multiple echo canceller practically realizable.

A second requirement for such an adaptive filter is concerned with the shift-variant property of wavelets. In Section 3.2.1, this problem was analyzed from the point of view of a single dispersive region. It was seen that, depending on the bulk delay, the partial Haar Wiener solutions can display peaks of varying magnitude, and this in turn determines how fast a dispersive region is located. The same is true for multiple echoes, and since the number of dispersive regions may not be known initially,

some filter coefficients might be falsely categorized as coefficient noise, and therefore ignored. This can adversely affect the echo canceller's performance.

Finally, an algorithm that cancels multiple echoes will also require a dynamic mechanism to allocate resources to dispersive regions that have been correctly located. These resources may include a set of short time-domain adaptive filters or a method for activating/deactivating regions of a full-length adaptive filter in response to locating new echoes.

Obviously, the task of multiple echo cancellation within the framework of [4] will require working around these problems in order to perform correctly and efficiently. In Section 4.5, an attempt at incorporating multiple echo cancellation into the coupled echo canceller will be pursued.

3.3 Implementation Issues

3.3.1 Positioning the Short Time-Domain Filter

Echo cancellers that employ two short adaptive filters operating in tandem [4, 16], usually centre a short time-domain filter around a dispersive region, based on its estimated location. The term 'centre' is somewhat vague and requires further clarification. From an ideal perspective, the term 'centring' means offsetting the short time-domain adaptive filter so that it is correctly aligned with the dispersive region.

One has to acknowledge, however, that any centring of the short time-domain filter is based on a peak location estimate from an adaptive filter operating in the partial Haar domain. Therefore, there are two factors that can potentially affect the proper centring of the short time-domain adaptive filter. First, the bulk delay estimate is based on the location of a peak, which usually results in an offset that is larger than the true bulk delay. In addition, the location of a peak depends on which set of partial Haar basis vectors is used, as each set emphasizes a specific frequency band. Therefore, depending on the particular spectral-temporal characteristic of the dispersive region, a peak's location can change simply by using a different subset of wavelet coefficients.

Second, the partial Haar adaptive filter is subsampled by an amount that depends on its length q, requiring the peak delay estimate be appropriately scaled before

centring the short time-domain filter. Therefore, the task of centring the short time-domain adaptive filter is not straightforward.

Let B denote the value of the true, but unknown bulk delay (in samples) of the time-domain echo path impulse response, and D represents a peak delay estimate,

$$D = (i-1)N/q + 1, \tag{3.38}$$

where $i \in \{1, 2, \ldots, q\}$ represents the filter-tap index corresponding to the peak and the term N/q is required due to the fact that the partial Haar adaptive filter is sub-sampled. A better estimate of the bulk delay that takes into account the previously-mentioned effects is given by

$$\widehat{B} = D - \lceil fL \rceil, \qquad D > \lceil fL \rceil. \tag{3.39}$$

The term $fL > 0$ represents a correction term, and L denotes the length (in samples) of the short time-domain adaptive filter. The value of f $(0 < f < 1)$ determines what fraction of the short time-domain filter length is subtracted from the peak delay estimate, D, to properly centre it around the dispersive region.

Finding a suitable value of f, requires an understanding of the possible effects of improperly setting it. If the resulting bulk delay estimate $\widehat{B}$ is too small, then the short time-domain filter will only contain the head of the impulse response, discarding the tail section. On the other hand, if the bulk delay is too large, then a portion of the head is omitted from the short time-domain adaptive filter. Usually the latter case is more detrimental to echo cancellation since the head of the echo path impulse response contains the most power.

An example of an ITU-T G.168 [1] hybrid impulse response is shown in Fig. 3.5. Consider a scenario utilizing this impulse response, where $N = 1024$ and the true bulk delay is $B = 600$ samples. Three partial Haar adaptive filters of lengths $q = \{256, 128, 64\}$ are used to test the effect of using different values of f in the [0,1] interval. The graph in Fig. 3.6 shows the squared weight-error (Euclidean) norm between the short time-domain adaptive filter $\mathbf{w}$ after convergence (plus the estimated bulk delay $\widehat{B}$) and the true impulse response $\mathbf{w}_o$, for $f \in [0, 1]$. As f increases, the bulk delay $\widehat{B}$ decreases. Figure 3.6 shows that for a given q, no unique value of f

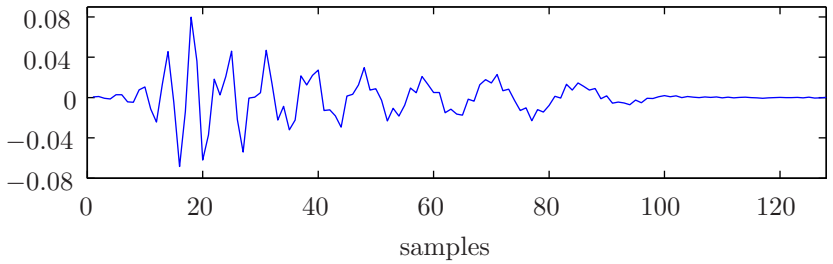


Fig. 3.5 ITU-T G.168 hybrid impulse response.

exists that makes $\|\mathbf{w} - \mathbf{w}_o\|^2 \approx 0$. This result shows that the coupled echo canceller is generally robust to any delay estimation jitter around the true peak. In addition, the intervals of zero squared norm change for different q. It is also interesting to note that these intervals shift as the length q decreases (scale decreases). For example, for $q = 256$, a permissible interval of f is $(0.05, 0.37)$, while for $q = 128$ the interval is $(0.2, 0.53)$.

Therefore, it seems that the rule of thumb for setting f is the following: if q is large (larger scale), then $f \approx 0.25$ while for smaller q, $f \approx 0.5$. For simulation purposes (as will be seen in Chapter 5) it is imperative to select a proper value of f in order to obtain reliable data. Of course, the above conclusions only serve as guidelines and are based on empirical results.

3.3.2 Partial Haar NLMS Algorithm

For the most part, the stochastic behaviour of the partial Haar-domain adaptive filter is analyzed with respect to the LMS algorithm in [4], and a majority of the simulations utilize the LMS algorithm with a fixed step size. In Chapter 2, it was shown that the LMS suffers from gradient noise amplification when the input samples become too large. The goal of this section, therefore, is to discuss how the NLMS algorithm can be applied to the partial Haar domain, beginning with the *complete* Haar NLMS algorithm:

$$\mathbf{v}(n+1) = \mathbf{v}(n) + \boldsymbol{\Delta}\mathbf{v}(n), \tag{3.40}$$

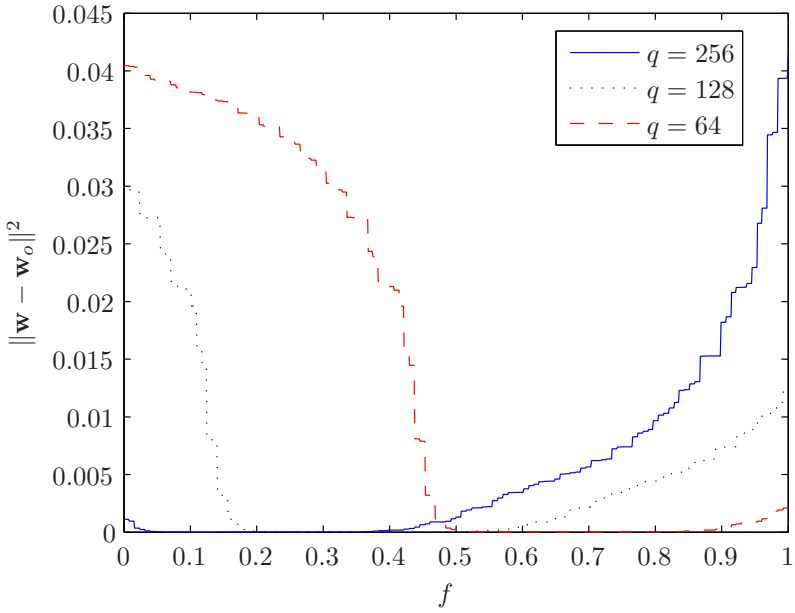


Fig. 3.6 The mean-squared weight error plotted for three different partial Haar adaptive filter lengths $q = 256, 128, 64$.

with

$$\mathbf{\Delta v}(n) = \frac{\mu}{\|\mathbf{z}(n)\|^2}\mathbf{z}(n)e_H(n) \tag{3.41}$$

$$e_H(n) = d(n) - \mathbf{z}^T(n)\mathbf{v}(n), \tag{3.42}$$

where $\mathbf{z}(n) = \mathbf{H}_N\mathbf{u}(n)$ is the transformed input vector and $\mathbf{v}(n)$ is the weight vector. Because $\mathbf{H}_N$ is an orthogonal matrix ($\mathbf{H}_N^T\mathbf{H}_N = \mathbf{H}_N\mathbf{H}_N^T = \mathbf{I}_N$), it follows that $\|\mathbf{z}(n)\|^2 = \mathbf{z}^T(n)\mathbf{z}(n) = \mathbf{u}^T(n)\mathbf{H}_N^T\mathbf{H}_N\mathbf{u}(n) = \|\mathbf{u}(n)\|^2$. In addition, the convergence behaviour of both filters is the same, since the eigenvalues of the transformed input correlation matrix are equal to the eigenvalues of the original autocorrelation matrix. This can be shown by observing the characteristic equation,

$$\begin{aligned} 0 &= \det[\lambda\mathbf{I} - \mathbf{R}_{zz}] \\ &= \det[\mathbf{H}_N(\lambda\mathbf{I} - \mathbf{R}_{uu})\mathbf{H}_N^T] \\ &= \det[\lambda\mathbf{I} - \mathbf{R}_{uu}]. \end{aligned} \tag{3.43}$$

An interesting observation can be made when analyzing the term $\mathbf{z}(n)e_H(n)$ in (3.41). First, expand $e_H(n)$ as

$$e_H(n) = d(n) - \mathbf{u}^T(n)\mathbf{H}_N^T\mathbf{v}(n). \tag{3.44}$$

Since $\mathbf{v}(n) = \mathbf{H}_N\mathbf{w}(n)$, where $\mathbf{w}(n)$ is the time-domain weight vector, this implies that

$$e_H(n) = d(n) - \mathbf{u}^T(n)\mathbf{w}(n), \tag{3.45}$$

which is the same as the time-domain error.

The partial Haar NLMS recursion formula is given by

$$\mathbf{v}_q(n+1) = \mathbf{v}_q(n) + \boldsymbol{\Delta}\mathbf{v}_q(n), \tag{3.46}$$

where $\mathbf{z}_q = \mathbf{H}_q\mathbf{u}(n)$ and

$$\boldsymbol{\Delta}\mathbf{v}_q(n) = \frac{\mu}{\|\mathbf{z}_q(n)\|^2}\mathbf{z}_q(n)e_q(n) \tag{3.47}$$

$$e_q(n) = d(n) - \mathbf{z}_q^T(n)\mathbf{v}_q(n). \tag{3.48}$$

Writing $e_q(n)$ in terms of $e_H(n)$,

$$\begin{aligned} e_q(n) &= d(n) - \mathbf{z}_q^T(n)\mathbf{v}_q(n) + \mathbf{z}^T(n)\mathbf{v}(n) - \mathbf{z}^T(n)\mathbf{v}(n) \\ &= e_H(n) + \bar{y}_q(n), \end{aligned} \tag{3.49}$$

where $\bar{y}_q(n)$ is defined as

$$\bar{y}_q(n) \triangleq \mathbf{z}^T(n)\mathbf{v}(n) - \mathbf{z}_q^T(n)\mathbf{v}_q(n). \tag{3.50}$$

Replacing (3.49) in (3.48), the partial Haar weight update vector becomes

$$\boldsymbol{\Delta}\mathbf{v}_q(n) = \frac{\mu}{\|\mathbf{z}_q(n)\|^2}\mathbf{z}_q(n)(e_H(n) + \bar{y}_q(n)). \tag{3.51}$$

Taking the expectation of the squared-Euclidean norm of (3.41) and (3.51) given

$\mathbf{u}(n)$, and assuming $e_H(n)$ and $\bar{y}_q(n)$ are uncorrelated, yields

$$E[\|\mathbf{\Delta v}\|^2] = \frac{\mu^2}{\|\mathbf{u}(n)\|^2} E[e_H^2(n)] \tag{3.52}$$

$$E[\|\mathbf{\Delta v}_q\|^2] = \frac{\mu^2}{\|\mathbf{z}_q(n)\|^2} (E[e_H^2(n)] + E[\bar{y}_q^2]). \tag{3.53}$$

As a result, $E[\|\mathbf{\Delta v}_q\|^2] \geq E[\|\mathbf{\Delta v}\|^2]$. Indeed, there are two factors that contribute to this relation: first, the extra term $E[\bar{y}_q^2]$ on the righthand side of (3.53), and second, the fact that $\|\mathbf{z}_q(n)\|^2 \leq \|\mathbf{u}(n)\|^2$. The latter relation's proof is given by

$$\begin{aligned} \|\mathbf{u}(n)\|^2 &= \|\mathbf{H}_N \mathbf{u}(n)\|^2 \\ &= |c_0(n)|^2 + \|\mathbf{H}_1 \mathbf{u}(n)\|^2 + \ldots + \|\mathbf{H}_{N/4} \mathbf{u}(n)\|^2 + \|\mathbf{H}_{N/2} \mathbf{u}(n)\|^2 \\ &\geq \|\mathbf{H}_q \mathbf{u}(n)\|^2 \\ &= \|\mathbf{z}_q(n)\|^2, \qquad 1 \leq q \leq N/2, \end{aligned} \tag{3.54}$$

where $c_0(n)$ represents the Haar scaling coefficient ($j = 0$) at time n.

The main conclusion to be drawn from the above analysis is that since $\bar{y}_q(n)$ is unavailable, one possible way to decrease (3.53) is to use the normalizing factor $\|\mathbf{u}(n)\|^2$ instead of $\|\mathbf{z}_q(n)\|^2$. This is the approach taken in this work.

3.3.3 Complexity Analysis of the Coupled Echo Canceller

Referring to the block diagram in Fig. 3.1, the coupled echo canceller is composed of three main units: partial Haar transform, partial Haar adaptive filter, and short time-domain adaptive filter.

In [4] there is no clear indication of how the partial Haar transform of the input data block is computed. The transform is introduced in matrix form, and it leaves one with the impression that the transformed data block is calculated using the matrix-vector multiplication $\mathbf{z}(n) = \mathbf{H}_q \mathbf{u}(n)$. Furthermore, one can recall from Section 3.1.1 that this transformation needs to be computed every iteration, i.e. for $\mathbf{z}(n)$, $\mathbf{z}(n+1)$,..., which can be computationally expensive.

The rows of the $q \times N$ transform matrix $\mathbf{H}_q$ each have a non-zero time support of length N/q samples. The computational cost of calculating the transformed input

data vector using the multiplication $\mathbf{H}_q\mathbf{u}(n)$ can be broken down as follows. For each basis vector, $N/q-1$ additions and N/q multiplications are required, which results in a total of $N-q$ additions and N multiplication operations. The number of multiplications can be reduced to 1 multiplication per iteration, resulting in a total of $N-q+1$ arithmetic operations per input sample[2].

The partial Haar adaptive filter employing the NLMS algorithm requires $2q+3$ multiplications, $2q+2$ additions, and 1 division operation per input sample. The normalizing factor is assumed to be calculated using the recursion $\|\mathbf{u}(n)\|^2 = \|\mathbf{u}(n-1)\|^2 - u^2(n-N) + u^2(n)$, which only requires 2 multiplication and addition operations per input sample.

The length-L short time-domain adaptive filter requires $2L+2$ multiplications and $2L$ additions. The reduced number of multiplications and additions results from the fact that the normalizing factor of the short-time domain filter can be approximated as $L\|\mathbf{u}(n)\|^2/N$.

Therefore, the total computational complexity of the coupled echo canceller using standard vector-matrix multiplication to calculate the partial Haar transform of the input is

$$\underbrace{4q+6}_{\text{PH-NLMS}} + \underbrace{4L+3}_{\text{ST-NLMS}} + \underbrace{N-q+1}_{\mathbf{H}_q\mathbf{u}(n)} = N+4L+3q+10, \tag{3.55}$$

in addition to the overhead of locating the peak[3]. As an example, consider a data block of length $N=1024$ using a length $q=128$ adaptive filter and a short time-domain adaptive filter of length $L=128$ at a sampling rate of 8 kHz. If the echo canceller is implemented on a floating point DSP, then the chip would have to be capable of 15.4 million floating point operations per second (MFLOPS). Since there is little margin for reducing the computational complexity of the two adaptive filters, the first section of Chapter 4 will be devoted to finding a more efficient way of calculating the partial Haar transform.

[2]Because the partial Haar wavelet functions consist of scaled ± 1s (see one of the subset of rows in equation 2.18), the number of multiplications per iteration can be reduced to 1 if the same scaling factor is applied to the output of the partial Haar adaptive filter.

[3]Depending on what method is used for peak location, this overhead can entail up to $q-1$ comparison operations to locate the peak using a brute force method. This work will make no assumptions concerning how the maximum peak is found.

Chapter 4

Improving the Coupled Echo Canceller

This chapter begins with a section on efficiently calculating partial Haar coefficients. This is followed by a brief introduction to non-Bayesian evidence theory and fuzzy logic systems. The chapter then responds to the three issues identified and discussed in Section 3.2, which include the dependence of the peak delay estimator's performance on the bulk delay, improved tracking performance, and the cancellation of multiple echoes.

4.1 Efficient Calculation of the Partial Haar Transform Coefficients

In Section 3.2.1, it was observed that although the wavelet transform is not exactly shift-invariant, it does exhibit periodic shift invariance. At time n, the transformed input data vector is given by $\mathbf{z}(n) = \mathbf{H}_q\mathbf{u}(n)$. Consider the length $N + N/q$ data vector $\mathbf{s}(n)$, which at time $n + N/q$ is given by

$$\mathbf{s}(n + N/q) = \begin{pmatrix} \mathbf{u}'(n) \\ \mathbf{u}(n) \end{pmatrix}, \tag{4.1}$$

where

$$\mathbf{u}'(n) = [u(n + N/q), u(n + N/q - 1), \ldots, u(n + 1)]^T. \tag{4.2}$$

To compute the Haar transform of $\mathbf{s}(n + N/q)$, a partial Haar transform matrix $\mathbf{H}'_q$ of size $(q + 1) \times (N + N/q)$ is constructed using $\mathbf{H}_q$ as follows

$$\mathbf{H}'_q = \begin{pmatrix} \mathbf{h} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_q \end{pmatrix}, \tag{4.3}$$

where $\mathbf{h}$ is a length-N/q row vector containing the non-zero portion of any row of $\mathbf{H}_q$ (see (2.18)). The length $q + 1$ transformed input data vector $\mathbf{z}'(n + N/q) = \mathbf{H}'_q\mathbf{s}(n + N/q)$ is given by

$$\mathbf{z}'(n + N/q) = \begin{pmatrix} \mathbf{h} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_q \end{pmatrix} \begin{pmatrix} \mathbf{u}'(n) \\ \mathbf{u}(n) \end{pmatrix} = \begin{pmatrix} \mathbf{h}\mathbf{u}'(n) \\ \mathbf{z}(n) \end{pmatrix}. \tag{4.4}$$

Notice that the above calculation makes use of $\mathbf{z}(n)$, an input vector transformed N/q samples ago. In fact, the new transformed input vector, $\mathbf{z}(n + N/q)$, corresponds to the first q elements of $\mathbf{z}'(n + N/q)$. Therefore, if $\mathbf{z}(n)$ is stored,

$$\mathbf{z}(n + N/q) = \begin{pmatrix} \mathbf{h}\mathbf{u}'(n) \\ \big[\mathbf{I}_{q-1}, \mathbf{0}\big]\mathbf{z}(n) \end{pmatrix}, \tag{4.5}$$

where $\mathbf{I}_{q-1}$ is the $(q - 1) \times (q - 1)$ identity matrix and $\mathbf{0}$ is a zero column vector of length $q - 1$.

Equation (4.5) provides an efficient way of calculating the partial Haar transform of a new input data vector by using the first $q - 1$ entries of a previously transformed vector, and requires the calculation of only a single partial Haar coefficient per iteration. Therefore, to exploit the relationship in (4.5) requires the development of an iterative algorithm, the matrix version of which will be developed below for times

$n + N/q - 1$ to $n + N/q$. Let

$$\mathbf{Z}(n + N/q - 1) = \begin{pmatrix} \mathbf{z}^T(n + N/q - 1) \\ \mathbf{z}^T(n + N/q - 2) \\ \vdots \\ \mathbf{z}^T(n) \end{pmatrix} \tag{4.6}$$

denote the transformed input matrix of size $N/q \times q$. At time $n + N/q$, a new partial Haar transform coefficient is computed, $z(n + N/q) = \mathbf{hu}'(n)$. The first $q - 1$ entries of the last row are reused and assigned to the last $q - 1$ entries of the first row. The last row is then discarded and the N/q most recent rows are shifted down into the matrix. The new transformed input matrix becomes

$$\mathbf{Z}(n + N/q) = \begin{pmatrix} z(n + N/q), \mathbf{z}^T(n)[\mathbf{I}_{q-1}, \mathbf{0}]^T \\ \mathbf{z}^T(n + N/q - 1) \\ \vdots \\ \mathbf{z}^T(n + 1) \end{pmatrix}, \tag{4.7}$$

The above can be seen as a vector form of the à Trous algorithm [42, 30] which is also known as the Redundant DWT (RDWT)[1]. It requires only $N/q-1$ addition operations and 1 multiplication per input sample at the expense of increased storage. Using this form of the partial Haar transform reduces the overall computational complexity from (3.55) to give the final complexity of

$$\underbrace{4q + 6}_{\text{PH}-\text{NLMS}} + \underbrace{4L + 3}_{\text{ST}-\text{NLMS}} + \underbrace{N/q}_{\mathbf{H}_q\mathbf{u}(n)} = N/q + 4L + 4q + 9 \tag{4.8}$$

arithmetic operations per iteration. If, for example, $N = 1024$ and $q = 128$, (4.8) reduces (3.55) by 889 arithmetic operations per iteration, and at 8 kHz, the resulting processor only requires 8.33 MFLOPS, a reduction of almost fifty percent!

Some might argue that the last sample of each of the last $N/q - 1$ rows of the transformed input matrix can be discarded to save memory, and this is true. However, each of these stored vectors plays a significant role in increasing the coupled echo

[1] It can also be seen as a classic polyphase implementation for a filter bank.

canceller's flexibility. At time $n + N/q$, the first row of $\mathbf{Z}$ in (4.7) corresponds to the current partial Haar transform of the input. The second row corresponds to the transformed input vector of the previous iteration. This row is also equivalent to using a shifted form of the transform at time $n + N/q$, with a shift of one sampleThe third row, corresponds to a partial Haar transform shifted right by two samples, and so forth. Therefore, from Section 3.2.1, depending on which row is used as an input to the partial Haar adaptive filter, the filter will converge to one of N/q partial Haar Wiener solutions (as seen in Section 3.2.1). In other words, each row of $\mathbf{Z}(n)$ corresponds to one of the N/q shifted partial Haar transforms. Although the above algorithm trades computational cost with an increase in memory, this memory can be put to good use.

Recall from Section 3.2.1 that depending on the bulk delay, the partial Haar adaptive filter can converge to one of N/q transformed Wiener solutions, each containing a different peak magnitude. For peaks with small magnitudes the amount of time required by the peak delay estimator to correctly estimate the location of a dispersive region was shown to be very long in some cases, slowing down the convergence of the echo canceller. The formulation in (4.7) effectively provides a way of selecting an alternate partial Haar input, and consequently a different Wiener solution. If the peak delay estimator is exhibiting too many false peak detections and jitters in the peak delay estimates, then a different transformed input can be used to steer the partial Haar adaptive filter towards a Wiener solution with a larger peak magnitude. The issue of when to decide to make such a change is discussed and developed in the next sections.

Before delving into the heart of this chapter, a definition is in order. A *shift context* (which will simply be referred to as a context) will be defined as which row of the transformed input matrix in (4.7) is being used to drive the partial Haar adaptive filter (the first row corresponds to the first context, and so forth). With respect to the time-domain impulse response's bulk delay, a context will be categorized as optimal if the resulting partial Haar Wiener solution has a large peak, which allows the peak delay estimator to quickly locate the peak (as shown in Fig. 3.4). The opposite is true for suboptimal contexts.

4.2 Preliminary Background

4.2.1 Non-Bayesian Evidence Theory

There are two factors that can make applying a probabilistic model of peak classification to adaptive filter tap weights difficult. First, the steady-state peak magnitude is not known *a priori*, and second, the coefficient variance which is governed by equations (3.14)-(3.18) depends on $J_{\min}$ and $\mathbf{v}_o$ which are unknown. Although these parameters can be estimated during the filter's steady-state operation, this can be costly. More importantly, as discussed in Section 3.2.2, the average time to detect a peak is usually much smaller than the convergence time of the filter, making estimation superfluous during steady-state.

Bayesian methods of inference or detection assume an underlying probability model in addition to prior distributions [20],[6],[37]. Belief function theory or evidence theory, on the other hand, provides a non-Bayesian formalism to deal with combining different sources of beliefs as well as quantifying uncertainty. Applications of evidence theory include statistical inference, medical diagnostics, expert systems, and risk/decision analysis [29]. The subsections below attempt to provide an overview of belief function theory and the pertinent rules of combination.

A. *Basic Belief Masses*

The frame of discernment or universe of discourse [13],

$$\Theta = \{\theta_i, i = 1, ..., n\} \tag{4.9}$$

is defined as the finite (only the finite case will be considered here) set of hypotheses[2] or basic elements of the frame. Furthermore, let $\mathscr{R}$ denote the Boolean algebra[3] of subsets of Θ with respect to the union and intersection operations. The choice of $\mathscr{R}$ determines the exclusivity of the elements of Θ. As will be seen later, this affects the way sources of beliefs are combined.

The basic units of belief in evidence theory are known as basic belief masses

[2]Such as a set of propositions: {"It rained today","It snowed today",...}.
[3]For convenience in notation the term $\{\theta_i\} \in \mathscr{R}$ is identified as θ_i.

(bbm), which are computed using basic-belief assignments (bba) (also termed the m-function). Basic belief assignments are unique to a given point of reference, source, or evidential corpus [45] (which could represent a person or sensor, for example). This assignment determines the distribution of an initial unitary belief mass among the propositions of $\mathscr{R}$. Mathematically, the bba is defined as

$$\begin{aligned} \sum_{A \in \mathscr{R}} m(A) &= 1, \\ m(\emptyset) &= 0. \end{aligned} \tag{4.10}$$

where $\emptyset$ denotes the empty set. The first equation states that the sum of the bbas over all subsets of the Boolean algebra sum to 1, while the second equation ensures that the empty set has no belief mass assigned to it. The m-function usually includes a subscript to indicate the evidential corpus to which the assignment belongs.

Most interpretations of evidence theory such as Smets' Transferable Belief Model (TBM) [45] are solely based on justifiable evidence. For example, consider a frame of discernment given by $\Theta = \{\theta_1, \theta_2\}$ with $\mathscr{R} = \{\emptyset, \theta_1, \theta_2, \Theta\}$. Initially, $m(\Theta) = 1$. The availability of evidence leading to an assignment $m(\theta_1) = 0.5$ does not imply that $m(\theta_2) = 1 - m(\theta_1) = 0.5$ (although this is required by one of the axioms of probability [36]). All it signifies is that a value of 0.5 from the original belief mass $m(\Theta) = m(\{\theta_1, \theta_2\})$, is transferred to θ_1, and the remaining 0.5 remains allocated to $m(\Theta)$ ($m(\theta_2)$ is still 0). Those bbms with $m(A) > 0$ are termed focal propositions or elements of Θ.

B. *Dezert-Smarandache Theory*

Dezert-Smarandache Theory (DSmT) provides a framework for combining different, sometimes conflicting sources of beliefs. Unlike Dempster-Schafer theory (DST), which assumes the exclusivity of hypotheses, i.e. $\theta_1 \cap \theta_2 = \emptyset$, DSmT extends DST by being able to deal with cases where non-exclusive hypotheses can exist [14].

There exist two models for combining beliefs in DSmT theory: the free DSm model and the hybrid DSm model [15]. Only the free DSm model will be considered here. This model assumes an exhaustive set of elements in Θ, but makes no additional

assumptions concerning the exclusivity of these elements, i.e. it is not assumed that $\theta_i \cap \theta_j = \emptyset$ for $i \neq j$. This makes DSmT perfect for applications that include fuzzy variables, where membership functions can overlap (membership functions will be discussed in Section 4.2.2).

Unlike DST, which is based on the power set, the free DSmT model is based on what Dezert [15] terms the hyper-power set, denoted by D^Θ, which includes the set of all subsets that can be constructed from the elements of Θ using the $\cap$ and $\cup$ operators[4]. The corresponding cardinality of D^Θ is bounded above by $2^{2^{|\Theta|}}$. The free DSm combination rule is given by

$$m(C) = \sum_{\substack{A,B \in D^\Theta, \\ A \cap B = C}} m_1(A) m_2(B). \tag{4.11}$$

The DSmT rule of combination seems expensive if one considers the cardinality of the hyper-power set D^Θ. However, in most applications, this computational cost is reduced since the number of focal elements is much smaller than $|D^\Theta|$. In addition, if the cardinality of Θ is small, the complexity is further reduced.

C. *Pignistic Transform*

Most proponents of belief-based evidence theory agree that basic belief masses have to be converted into probabilities before using them in decision-making processes. Smets' [44] argues that the framework of belief functions consists of two levels. The credal (of beliefs) level where basic belief masses are distributed, combined and updated, and a Pignistic or decision-making level where belief masses are converted into probabilities.

In DST-like frameworks such as Smets' TBM [45], the original Pignistic Transform is given by

$$P\{\theta_i\} = \sum_{A \in P^\Theta} \frac{|\theta_i \cap A|}{|A|} m(A), \qquad A \in P^\Theta, \tag{4.12}$$

which maps $\theta_i \in \Theta$ into a number $P\{\theta_i\} \in [0, 1]$, where $P\{\cdot\}$ is the Pignistic probability function, and P^Θ is the power-set defined over Θ. Here, an earlier version of a

[4]For example, if $\Theta = \{\theta_1, \theta_2\}$, then $P^\Theta = \{\emptyset, \theta_1, \theta_2, \Theta\}$ and $D^\Theta = \{\emptyset, \theta_1, \theta_2, \Theta, \theta_1 \cap \theta_2\}$ with $|D^\Theta| = 5$.

Pignistic Transform developed by Dezert in the context of DSmT will be used [13]. This transform is defined as

$$P\{\theta_i\} = \sum_{A \in D^\Theta} \alpha_{\theta_i}(A) m(A). \tag{4.13}$$

where the $\alpha_{\theta_i}(A)$ are weighting coefficients that represent the inclusion or exclusion of a proposition A in θ_i. In the case of $\Theta = \{\theta_1, \theta_2\}$, the Pignistic probabilities are given by

$$P\{\theta_1\} = m(\theta_1) + 0.5m(\theta_1 \cup \theta_2) + 0.5m(\theta_1 \cap \theta_2) \tag{4.14}$$

$$P\{\theta_2\} = m(\theta_2) + 0.5m(\theta_1 \cup \theta_2) + 0.5m(\theta_1 \cap \theta_2). \tag{4.15}$$

The coefficients 1 and 0.5 can be found by referring to the Venn diagram in Fig. 4.1. For example, the proposition $\theta_1 \cap \theta_2$ consists of one unit a_2 which is shared by both

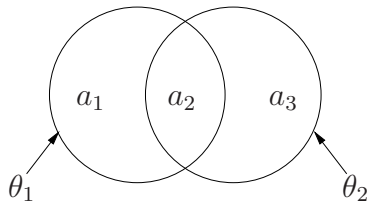


Fig. 4.1 Venn diagram representing the free DSm model.

θ_1 and θ_2 and is assigned a weight of 1/2. Therefore, $\alpha_{\theta_1}(\theta_1 \cap \theta_2) = \dfrac{1/2}{|\{a_2\}|} = 0.5$. Similarly, $\alpha_{\theta_2}(\theta_1 \cap \theta_2) = 0.5$. Proposition $\theta_1 \cup \theta_2$ consists of three units a_1, a_2 and a_3. Therefore, $\alpha_{\theta_1}(\theta_1 \cup \theta_2) = \dfrac{1/2 + 1 + 0}{|\{a_1, a_2, a_3\}|} = 0.5$, where the 1/2-term is again due to the sharing of a_2 between θ_1 and θ_2, the 1-term occurs because a_1 is only a subset of θ_1, and the 0-term occurs because a_3 is not a subset of θ_1.

4.2.2 Basics of Fuzzy Set and Systems Theory

In most cases, bbms are assigned by experts in a given field, making the task of assigning beliefs unrealistic in the absence of these experts. Instead, as presented in [47], this work will make use of a fuzzy interface to generate bbms. This interface incorporates expert opinion in the form of fuzzy membership functions.

In contrast to crisp sets where an element is either included or excluded from a set, fuzzy sets define an element's degree of set membership [35]. This degree of membership is determined by a membership function $\mu_A(x)$, where x is a crisp value in a universe of discourse[5] (discrete or continuous) and A is a fuzzy set that represents a linguistic description. For example, consider a universe of discourse $\mathcal{X}$ which equals the temperature reading of a thermometer between 0 °C and 100 °C. A crisp temperature reading is denoted by x, where $x \in \mathcal{X}$. Now consider the fuzzy sets represented by the linguistic terms 'cold','moderate','warm', and 'hot'. A membership function $\mu_l(x), l \in \{\text{cold}, \text{moderate}, \text{warm}, \text{hot}\}$, is associated with each set and determines the membership grade of x with respect to that set.

In mathematical terms, whereas the value of a crisp membership function is $\mu_{crisp}(x) \in \{0, 1\}$ depending on whether $x \in A$ or not, the value of a fuzzy membership function, $\mu_{fuzzy}(x)$, lies in the interval $[0, 1]$. The degree of membership (or membership grade) can vary from no membership at all ($\mu_{fuzzy}(x) = 0$) to full membership ($\mu_{fuzzy}(x) = 1$). Depending on the application, membership functions can vary in shape and support [49]. For the present application, piecewise-linear membership functions will be used.

One of the most important properties of fuzzy systems is their rule-base and inference mechanism. The rule-base consists of a set of IF-THEN (antecedent-consequent) statements that represent some form of prior human knowledge that describes the behaviour of a system. Unlike traditional logic, fuzzy inferences can be made even when rules are only partially satisfied [28, 35]. This is in stark contrast to crisp logic, where a rule is satisfied only if a premise of interest matches an implication's antecedent exactly [35].

At the heart of a fuzzy inference system is a fuzzy composition rule such as Zadeh's max-min rule [35]

$$\mu_{C'}(y) = \max_{x \in \mathcal{X}} \min_{i} (\mu_{A'}(x), \mu_{A_i \times C_i}(x, y)), \qquad y \in \mathcal{Y}, \tag{4.16}$$

where the fuzzy sets A_i and C_i respectively correspond to the antecedent and consequent $(A \rightarrow C)$ of the rule-base's i^{th} rule. The apostrophe is used to signify

[5] This universe of discourse should not be confused with the frame of discernment in Section 4.2.1. Here, the universe of discourse is just a membership functions' domain of crisp inputs

that the antecedent A_i of a relation might only be partially satisfied by a fuzzy set A', resulting in a different consequent C'. The term $\mu_{A\times C}(x,y)$ is a fuzzy relation defined as a two dimensional membership function over the cartesian product, i.e. $\mu_{A\times C} : \mathcal{X} \times \mathcal{Y} \rightarrow [0,1]$.

Fuzzy relations provide a measure for the degree of association between fuzzy sets, and are derived from the IF-THEN rules contained in the rule-base. Depending on how implications are defined, the mathematical form of $\mu_{A\times C}$ is different. In this thesis, the relation between the antecedent and consequent of the i^{th} rule will be defined as

$$\mu_{A_i\times C_i}(x,y) = \min(\mu_{A_i}(x), \mu_{C_i}(y)), \qquad x \in \mathcal{X}, y \in \mathcal{Y}, \tag{4.17}$$

which is also one of many fuzzy intersection rules. Using the above formula, Zadeh's max-min compositional rule can be rewritten as [28]

$$\mu_{C'}(y) = \max_i \min\{\max_{x\in\mathcal{X}} \min(\mu_{A'}(x), \mu_{A_i}(x)), \mu_{C_i}(y)\}, \qquad y \in \mathcal{Y}. \tag{4.18}$$

The reformulation in (4.18) provides a simple interpretation of the max-min compositional rule in (4.16) given by [35]

$$\boldsymbol{\mu}_{\mathcal{Y}}(y) = \boldsymbol{\mu}_{\mathcal{X}}(x) \circ \mathbf{G}, \tag{4.19}$$

where $\boldsymbol{\mu}_{\mathcal{X}}(x)$ and $\boldsymbol{\mu}_{\mathcal{Y}}(y)$, are vectors containing membership grades of the fuzzy sets belonging to the universes of discourse $\mathcal{X}$ and $\mathcal{Y}$, respectively, and $\circ$ denotes the max-min composition operation. $\mathbf{G}$ is a relational matrix that can be constructed from (4.18) and the membership functions $\mu_{A_i}(x)$,$\mu_{C_i}(x)$. Details for constructing $\mathbf{G}$ will be delayed to the next section when the peak tendency estimator is developed.

Equation (4.19) provides an alternate interpretation of (4.16) as a form of vector-matrix multiplication. However, unlike traditional vector-matrix multiplication, the multiplication and addition operations are replaced by the min and max operators, respectively (other compositional rules also exist, such as the max-product compositional rule).

4.3 Escaping Suboptimal Contexts

In Section 3.2.1, it was shown that each of the N/q contexts results in a Wiener solution that exhibits a different peak magnitude. For contexts that produce peaks of sufficient magnitude, the peak delay estimator can quickly locate a peak as its magnitude exceeds those of its neighbouring coefficients. However, in some instances the filter is operating in a context that produces a very small peak, which during the course of the partial Haar filter's convergence can not be distinguished from the neighbouring coefficient noise. This drastically increases the time required to correctly estimate the location of a dispersive region. Sometimes, the correct bulk delay can not be found due to the persistent jitter present in the delay estimate.

Efforts to design shift-invariant wavelets or wavelets capable of providing improved peak detection to solve this problem can still suffer from the above problem. A good example of this is the complex wavelet transform that uses two wavelet functions that form a Hilbert transform pair. The magnitude of the resulting complex coefficients is almost shift-invariant [39, 40], but nevertheless, can also result in Wiener solutions with very small peaks.

In the following sections, an algorithm is developed that deals with the problem of suboptimal contexts, and consists of two stages. First, it determines when the filter is operating in an optimal context based on a peak's behaviour, and second, if the current context is classified as suboptimal, it uses a form of scheduling to decide when to escape the current context. Tackling the second problem is more difficult than the first, because making the decision about when to use a different context is difficult in a framework such as adaptive filtering. Unlike classical signal detection theory, the convergence behaviour of adaptive filters includes transients. This makes it difficult to make reliable decisions.

4.3.1 Quantifying a Peak Discernibility Measure

The terms 'optimal' and 'suboptimal' that describe the peak delay estimator's performance in a given context can be associated with some properties of the partial Haar adaptive filter. These properties mainly include fast estimates of a true peak's location, in addition to absence of jitter in these estimates. Peak jitter is commonly

associated with the transient portion of the filter's learning curve where the prominent filter coefficient magnitudes have not had time to surpass neighbouring tap weights. Therefore, in most cases, the absence of peak jitter indicates the prominence of a peak tap-weight.

False peak detections occurring during steady-state operation of the partial Haar filter can be attributed in part to the low peak magnitude (relative to neighbouring coefficients) of the corresponding context. In severe suboptimal cases, the peak delay estimator is sometimes tracking nothing more than coefficient noise.

One can therefore conclude that a measure of 'optimality' should include both stability of the maximum peak location as well as the fast discernibility of the peak. In addition, since the average time to detect a peak is usually much smaller than the convergence time of the partial Haar filter (as shown in Section 3.2.2), one would like to correctly locate a peak before the filter has converged. Usually, the magnitude itself is a good indicator of the stability of a peak's location, although simply relying on magnitude lends no information about discernibility, which is a relative property that should take into account the magnitudes of neighbouring filter coefficients.

The proposed peak measure makes use of two observations. First, searching for a maximum peak using a distributed approach can result in lower latency. Second, local maxima found by each of these distributed units can be used as parameters to measure the discernibility of a global maximum. In this work, a peak discernibility measure is calculated as follows: partition the filter coefficients into three contiguous groups of roughly the same size; find the maximum peak magnitude for each of the three groups; the global maximum is found by selecting the maximum among the three maxima. The peak discernibility measure (or PDM) is defined as

$$\mathrm{PDM} = 1 - \frac{c_{\min}}{c_{\max}}, \tag{4.20}$$

where $c_{\min}$ is the minimum among the three maxima found, and $c_{\max}$ corresponds to the global maximum that is fed to the peak detector.

As the global peak magnitude increases, $c_{\max}$ increases and the PDM approaches 1. Similarly, if the global maximum is comparable to $c_{\min}$, then the PDM is close to 0. The block diagram for calculating the PDM is shown in Fig. 4.2.

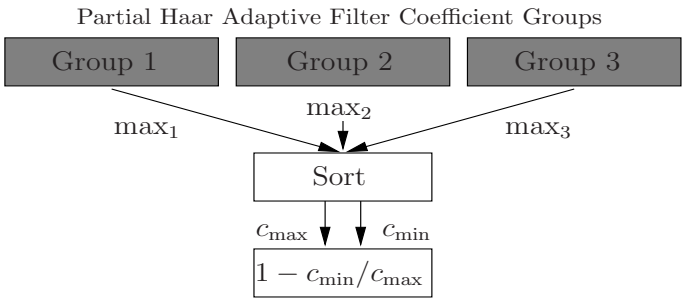


Fig. 4.2 Block diagram of peak discernibility measure (PDM) calculation.

The reason for partitioning the filter into three groups rather than two, can be explained as follows. If a partial Haar-domain dispersive region falls directly on the boundary separating two partitions, it is highly possible that each partition's maximum will be associated with the dispersive region. If the partial Haar transform of the dispersive region contains multiple peaks, this can produce an invalid PDM, since $c_{\min}$ does not represent tap-weight noise. On the other hand, when the filter is divided into three partitions, there is an extra degree of freedom in that $c_{\min}$ is usually not associated with the dispersive region, and therefore results in a more reliable PDM.

4.3.2 Constructing a Fuzzy Interface

By properly defining membership functions for characterizing the presence of a small or large peak, a system can be developed for tracking peak magnitude tendencies as either increasing or decreasing by combining fuzzy set theory and the DSmT rule of combination. This information can be used to solve many of the problems associated with the coupled echo canceller of Section 3.1. In addition, fuzzy theory and belief functions can be combined to explicitly quantify uncertainty. The fuzzy interface considered here is similar to [47] and consists of two membership functions, S and L, which correspond to the linguistic variables 'small' and 'large', respectively. The universe of discourse is the $[0, 1]$ interval corresponding to possible values of the PDM.

There are numerous methods, both online, and offline, for constructing membership functions [18]. Here, an offline approach based on cumulative distribution functions (CDFs) of the PDM values in various situations is considered. This approach constructs piecewise-linear membership functions to approximate CDFs whose

closed-form expressions do not exist or are difficult to derive. Zadeh has noted that membership functions do not need to be precise, and this somewhat empirical method is not unreasonable when peak tendency is being estimated [51].

The two CDFs considered here are the following: $\phi_s(x)$ corresponds to the values of the PDM at the instant when peak delay is correctly estimated, and $\phi_n(x)$ corresponds to those PDM values before peak delay jitter occurs (the subscripts s and n refer to signal and filter coefficient noise, respectively). The ideal membership functions are derived as

$$\mu_S(x) = 1 - \phi_n(x) \tag{4.21}$$

$$\mu_L(x) = \phi_s(x), \tag{4.22}$$

where $x \in [0, 1]$ is a possible value of the PDM in (4.20). The reason for using $1 - \phi_n(x)$ has to do with the fact that if a PDM x' causes a peak delay jitter, then it is highly possible that a value $x < x'$ will cause jitter as well. The CDFs $\phi_s(x)$ and $\phi_n(x)$ are found by running a large number of simulations over varying SNRs (10 to 60 dB in increments of 10 dB) and ITU-T G.168 impulse responses [1]. The resulting membership functions are shown in Fig. 4.3, together with their piecewise linear approximations. Observing these membership functions, one notes that their intersection point occurs somewhere around a PDM value of 0.2. In addition, their slope magnitudes are relatively large in this region. In practice, it is advisable to take a more pessimistic approach to constructing membership functions, largely due to uncertainties. This can be accomplished by reducing the slopes of the piecewise linear functions, and shifting the intersection point towards a larger PDM value. The resulting membership functions are shown in Fig. 4.4. Each employs a pair of different linear functions. Not only does such an interface produce better results in uncertain environments, but by spacing out the two membership functions, the categories of small and large peak magnitudes are spaced further apart. This reduces false classifications and allows for smoother tracking of transitions between the two extremes.

The fuzzy interface can now be used to find bbms corresponding to an input PDM

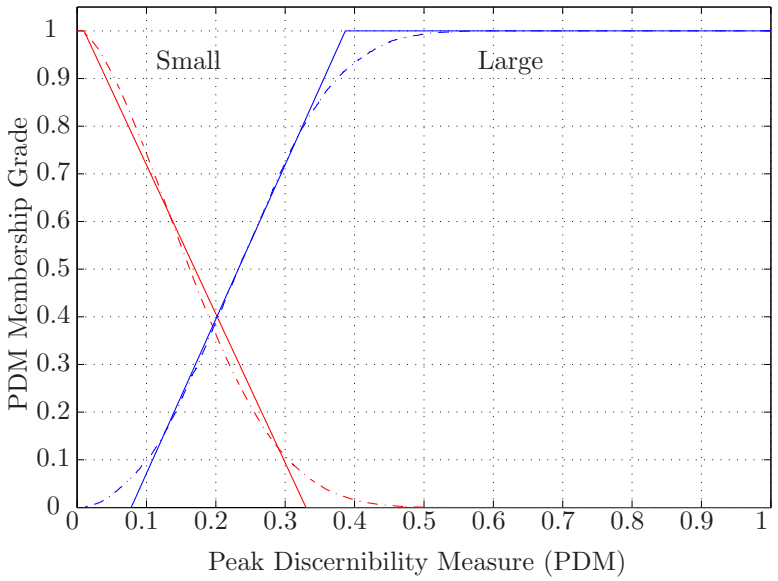


Fig. 4.3 Linear piecewise membership functions derived from $\phi_s(x)$ and $1 - \phi_n(x)$ (based on simulation).

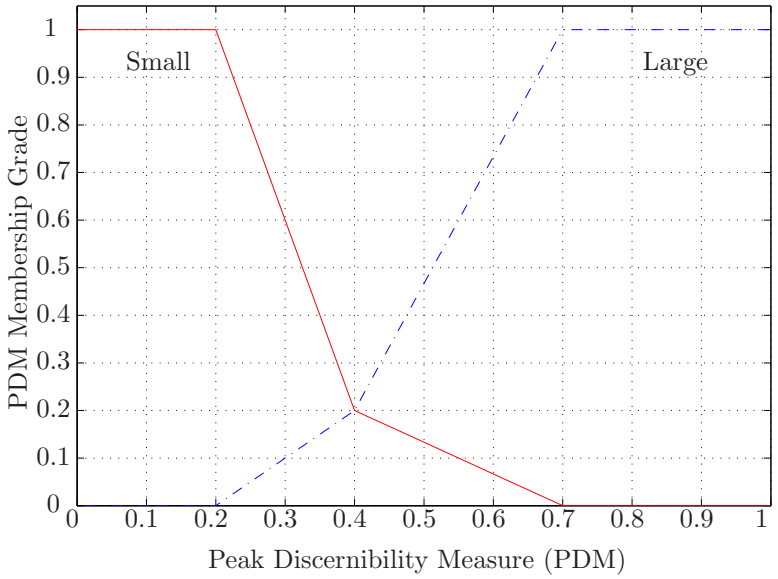


Fig. 4.4 A robust approximation of membership functions.

value x',

$$m(L) = \mu_L(x') \tag{4.23}$$
$$m(S) = \mu_S(x') \tag{4.24}$$
$$m(S \cup L) = 1 - m(S) - m(L). \tag{4.25}$$

The bbm value, $m(S \cup L)$, results from the fact that fuzzy membership grades do not necessarily sum up to 1 for a given x, and this term represents uncertainty [5].

4.3.3 Suboptimal-Context Escape Algorithm

The goal of the suboptimal-context escape algorithm is to categorize the tendency of the peak magnitude, in order to make a decision about the operating context of the partial Haar adaptive filter. If the peak is classified as noise for a certain amount of time, then the context should be changed.

The algorithm makes use of an architecture similar to that of [47] where DSmT, together with fuzzy inference, is used to track the tendencies of a moving target in passive sonar based on the amplitude of a signal emitted by the target. The adopted system is shown in Fig. 4.5, and makes use of two peak magnitude behavioural models.

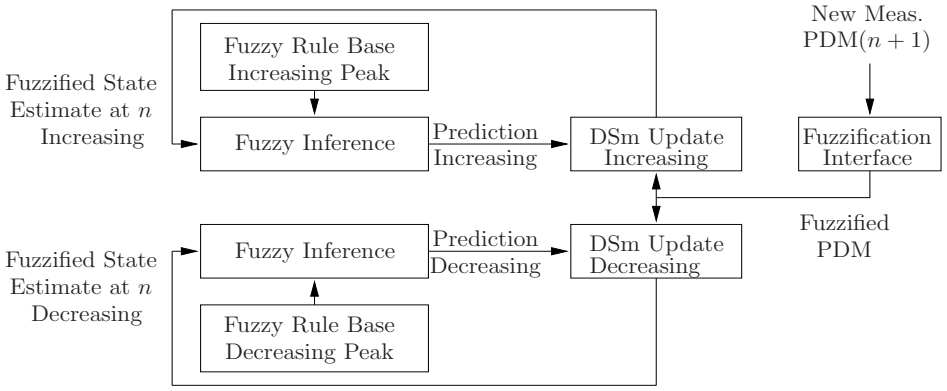


Fig. 4.5 Block diagram of Peak Tendency Estimator.

The first model (top) is that of a peak magnitude that increases over time, and results in the proper detection of the dispersive region. The second model (bottom)

corresponds to a decreasing peak magnitude, which can be used to detect filter coefficient noise. Each behavioural model consists of its own rule-base that describes the peak magnitude's tendency. For a true peak, the transition of the peak magnitude increases over time with a transition $S \rightarrow S \rightarrow L \rightarrow L$, while a decreasing (small) peak is characterized by the transition $L \rightarrow L \rightarrow S \rightarrow S$. The respective rule-bases are shown in Table 4.1.

Table 4.1 Fuzzy Rule Bases

Rule No.	Increasing PDM	Decreasing PDM
1:	If PDM$(n) = S$ then PDM$(n+1) = S$	If PDM$(n) = L$ then PDM$(n+1) = L$
2:	If PDM$(n) = S$ then PDM$(n+1) = L$	If PDM$(n) = L$ then PDM$(n+1) = S$
3:	If PDM$(n) = L$ then PDM$(n+1) = L$	If PDM$(n) = S$ then PDM$(n+1) = S$

The entries of relational matrices in Table 4.2 are found by using the rule-base in Table 4.1 and the fuzzy interface in Fig. 4.4 (for completeness, the sets $S \cup L$ and $S \cap L$ have been included, although they can be ignored in general). For an entry $g_{A' \rightarrow C'}$, where $A', C' \in \{S, L\}$, one of two things can happen. Either A' and C' exactly match the antecedent (A) and consequent (C) of a rule, or only C' matches a rule's consequent, i.e., $A' \neq A$. For example, the entry $g_{S \rightarrow S}$ in Table 4.2, matches the first rule of the increasing PDM rule-base in Table 4.1, i.e. $A' = A$ and $C' = C$. Therefore, the term $\max_{x \in \mathcal{X}} \min(\mu_{A'}(x), \mu_{A_i}(x))$ in (4.18) reduces to

$$\max_{x \in \mathcal{X}} \min(\mu_S(x), \mu_S(x)) = \max_{x \in \mathcal{X}} (\mu_S(x)) = 1. \tag{4.26}$$

Hence, $[g_{S \rightarrow S}] = 1$, as shown in the table. Now consider what happens when the implication $A' \rightarrow C'$ is not present in the rule-base. In such a case, one has to match C' with the consequent of a rule. For example, $L \rightarrow S$ is not present in the increasing PDM rule-base (Table 4.1). Therefore, one has to utilize a rule whose consequent equals S, which corresponds to the first rule. In this case,

$$\max_{x \in \mathcal{X}} \min(\mu_L(x), \mu_S(x)) = \max_{x \in \mathcal{X}} (\mu_{L \times S}(x)) = 0.2, \tag{4.27}$$

and the corresponding entry $g_{L \rightarrow S} = 0.2$.

At time n, the peak magnitude is characterized by the pair of fuzzy PDM values

Table 4.2 Fuzzy graphs corresponding to two models of peak behaviour.

(a) Increasing Peak, $\mathbf{G}^{\text{inc}}$

$n \rightarrow n+1$	S	$S \cup L$	L	$S \cap L$
S	1	0	1	0
$S \cup L$	0	0	0	0
L	0.2	0	1	0
$S \cap L$	0	0	0	0

(b) Decreasing Peak, $\mathbf{G}^{\text{dec}}$

$n \rightarrow n+1$	S	$S \cup L$	L	$S \cap L$
S	1	0	0.2	0
$S \cup L$	0	0	0	0
L	1	0	1	0
$S \cap L$	0	0	0	0

corresponding to the two models $\boldsymbol{\mu}^{\text{inc}}(n|n)$ and $\boldsymbol{\mu}^{\text{dec}}(n|n)$, where

$$\boldsymbol{\mu}^{M}(n|n) = [m(S), m(S \cup L), m(L), m(S \cap L)]^{M}, \qquad M = \text{inc or dec.} \tag{4.28}$$

Using (4.16), a state prediction of each model can be obtained,

$$\boldsymbol{\mu}^{\text{inc}}(n+1|n) = \boldsymbol{\mu}^{\text{inc}}(n|n) \circ \mathbf{G}^{\text{inc}} \tag{4.29}$$

$$\boldsymbol{\mu}^{\text{dec}}(n+1|n) = \boldsymbol{\mu}^{\text{dec}}(n|n) \circ \mathbf{G}^{\text{dec}}. \tag{4.30}$$

Each prediction requires normalization for use in the DSmT combination step.

At time $n+1$, a new vector of fuzzy PDM values is obtained from (4.23)-(4.25), and is denoted by $m_{n+1} = [m(S), m(S \cup L), m(L), m(S \cap L))]$. This new input is then combined with $\boldsymbol{\mu}^{\text{inc}}(n+1|n)$ and $\boldsymbol{\mu}^{\text{dec}}(n+1|n)$ using the DSmT rule of combination in (4.11) to find the updated fuzzy state vectors $\boldsymbol{\mu}^{\text{inc}}(n+1|n+1)$ and $\boldsymbol{\mu}^{\text{dec}}(n+1|n+1)$, respectively.

Making a decision about the peak magnitude's tendency then involves converting each set of updated bbms into their Pignistic probabilities P^{M} from (4.14)-(4.15) with $\theta_1 = S$ and $\theta_2 = L$. At any time instant, a decision about the correct behavioural model is based on the model with the smallest entropy given by

$$H^{M}_{\text{pig}}(P^{M}) = -\sum_{A \in \{S,L\}} P^{M}\{A\} \ln(P^{M}\{A\}), \tag{4.31}$$

where $P^{M}\{A\} \ln(P^{M}\{A\}) = 0$ for $P^{M}\{A\} = 0$. The peak-tendency estimation algorithm is shown in Fig. 4.6.

Escaping a suboptimal context, however, also requires deciding when to act on the

Given : $\boldsymbol{\mu}^M(n|n), \mathbf{G}^M, M = \{\text{inc}, \text{dec}\}$.
Input : $\text{PDM}(n+1)$.

`// new observation: 2 comp., 4+, 2×`
$m_{n+1}(S) = \mu_S(\text{PDM}(n+1))$;
$m_{n+1}(L) = \mu_L(\text{PDM}(n+1))$;
$m_{n+1}(S \cup L) = 1 - m_{n+1}(S) - m_{n+1}(L)$;
$m_{n+1}(S \cap L) = 0$;
for *each model* $M = \{\text{inc}, \text{dec}\}$ **do**
`// prediction-normalization (`$m_{\text{pred}}(S \cup L) = m_{\text{pred}}(S \cap L) = 0$`):3 comp., 1+, 1/`
$\boldsymbol{\mu}^M(n+1|n) \triangleq [m_{\text{pred}}(S), m_{\text{pred}}(S \cup L), m_{\text{pred}}(L), m_{\text{pred}}(S \cap L)] = \boldsymbol{\mu}^M(n|n) \circ \mathbf{G}^M$;
$\boldsymbol{\mu}^M(n+1|n) = \dfrac{\boldsymbol{\mu}^M(n+1|n)}{m_{\text{pred}}(S) + m_{\text{pred}}(L)}$;
`// state updating using (4.11): 6×, 3+`
$m_{\text{upd}}(S) = m_{\text{pred}}(S)m_{n+1}(S) + m_{\text{pred}}(S)m_{n+1}(S \cup L)$;
$m_{\text{upd}}(S \cup L) = 0$ $m_{\text{upd}}(L) = m_{\text{pred}}(L)m_{n+1}(L) + m_{\text{pred}}(L)m_{n+1}(S \cup L)$;
$m_{\text{upd}}(S \cap L) = m_{\text{pred}}(S)m_{n+1}(L) + m_{\text{pred}}(L)m_{n+1}(S)$;
`// new state vector`
$\boldsymbol{\mu}^M(n+1|n+1) = [m_{\text{upd}}(S), m_{\text{upd}}(S \cup L), m_{\text{upd}}(L), m_{\text{upd}}(S \cap L)]$;
`// Pignistic transform (4.14)-(4.15): 2+, 2×`
$P^M\{S\} = m_{\text{upd}}(S) + 0.5m_{\text{upd}}(S \cup L) + 0.5m_{\text{upd}}(S \cap L)$;
$P^M\{L\} = m_{\text{upd}}(L) + 0.5m_{\text{upd}}(S \cup L) + 0.5m_{\text{upd}}(S \cap L)$;
`// Pignistic entropy (4.31): 1 comp.`
$H^M_{\text{pig}} = -P^M\{S\}\ln(P^M\{S\}) - P^M\{L\}\ln(P^M\{L\})$;
end

Fig. 4.6 Peak-tendency estimation algorithm.

information provided by the above system. Particularly in the case of the partial Haar adaptive filter, this can prove to be difficult and should be based on the convergence behaviour of the adaptive filter. This, in turn, depends on a multitude of factors such as the input statistics, channel conditions, and most importantly, the context within which the filter is operating. In [24], this time is set to some multiple of the average time constant of the adaptation algorithm, to give the filter sufficient time to converge.

The approach developed here is slightly more involved due to the fact that there are N/q contexts. Therefore, instead of waiting for a certain number of time samples until the filter converges, the proposed method uses a schedule of N/q waiting times to sequentially test each context. The waiting times are derived so as to reduce the

average time spent waiting for a context that allows for the successful estimation of a dispersive region's location.

The above problem, is not unique to suboptimal context escaping, but can be generalized to a theory of testing a finite set of alternatives. For example, consider being given a problem with a unique solution S, and a reward R for solving the problem. Assume there are J different ways this problem can be solved, and each one leads to this one solution, i.e. all approaches produce the same reward. However, each approach A_j ($j = 1, \ldots, J$), is associated with a random cost (with respect to time) C_j with distribution F_j. In other words, each approach requires a different amount of time to reach the solution. Although the different approaches A_j are known, the unique mapping $M : A_i \rightarrow F_j$ between each approach and its corresponding cost distribution is unknown.

The problem is to find an optimal schedule consisting of J trial periods $\boldsymbol{\tau} = \{\tau_1, \ldots, \tau_J\}$ to sequentially attempt each alternative so that the average time spent reaching the solution is minimized, subject to the constraint that the probability of eventually finding the solution using this schedule is greater than or equal to some confidence threshold. The problem is further restricted in that once an alternative has been attempted without success for a period of time greater than the current trial period, it is discarded and never revisited again.

Applying the above analysis to suboptimal-context escaping is not straightforward. The main reason is that not only is the mapping between contexts and their distributions unknown, but the distributions are also unknown. Therefore, the best that can be done is to find a schedule based on some prior knowledge about correct estimation times. In other words, this schedule should ensure that the partial Haar adaptive filter stay in an optimal context, while only remaining in suboptimal contexts as a last resort.

A schedule that satisfies the above requirements consists of a sequence of non-decreasing trial periods, i.e. $\tau_1 \leq \tau_2 \leq \ldots \leq \tau_N$. The non-decreasing property is critical here because the number of contexts that can successfully lead to a correct estimate of the peak delay should increase after each failed attempt. At the same time, beginning the schedule with shorter waiting times escapes any suboptimal contexts earlier in the peak delay estimation process, while at the same time successfully staying

in optimal contexts.

The individual trial times or periods of the scheduling policy remain to be determined. The first trial period should be related to the number of samples required to properly detect the global peak in an optimal context. As will be seen in Chapter 5, optimal contexts usually require less than a 200 samples to correctly detect a peak using an NLMS algorithm with a step-size $\mu = 1$ and a sampling rate of 8 kHz. Therefore, the first trial period can be set to approximately 150 samples.

Setting the final trial period (N/q) is usually based on the amount of time willing to be spent operating in a worst-case context. This time period should not be set too long, and can be set somewhere between 400–500 samples. The remaining trial periods can be uniformly distributed between the minimum and maximum trial periods. To summarize, the algorithm begins by monitoring the peak tendency using the smallest trial period. If the peak tendency is classified as decreasing for a period of time greater than this trial period and jitter occurs in the peak delay estimate, then the filter is reset to zero and a new context is attempted for the next trial period. This is repeated until the peak delay is correctly estimated. One issue has been ignored, and that is what happens when none of the contexts provide a correct peak using the given schedule? The approach used here re-tests all contexts in the same order as the previous test, beginning with the second-shortest trial time. The suboptimal-context escape algorithm is shown in Fig. 4.7.

4.4 Improved Tracking of the Partial Haar Adaptive Filter

In Section 3.2.2, the tracking of a dispersive region after an abrupt change in the echo path impulse response was seen as a competition between filter coefficient magnitudes. Changes in an echo path impulse response can be associated with a change in the bulk delay, or a phase roll, where the impulse response coefficients change signs [50].

When the partial Haar adaptive filter is initialized to zero, a peak's magnitude only has to compete with the low-magnitude coefficient noise of the surrounding taps, which makes its detection easy and fast. In the event of an abrupt change in the echo path impulse response, however, a new peak might have to compete with the decreasing magnitude of an old peak. Therefore, although the echo path impulse

Given: $H_{\text{pig}}^{\text{inc}}$ and $H_{\text{pig}}^{\text{dec}}$ from peak tendency estimator (see Fig. 4.6),
schedule, $\boldsymbol{\tau} = \{\tau_1, \tau_2, \ldots, \tau_{N/q}\}$,
partial Haar peak indices at times n and $n+1$: $index(n)$, $index(n+1)$,
peak-tendency counters: counter_inc and counter_dec,
context variable, context.

Initial: (at $n = 0$), counter_inc = counter_dec $= 0$, $k = 1$, $T = \tau_1$, context $= 1$, $\mathbf{v}(n) = \mathbf{0}$.

```
// at time n+1...
if H_pig^dec < H_pig^inc then
    counter_dec = counter_dec + 1;
else
    counter_inc = counter_inc + 1;
end
// if the current trial time is exceeded and peak jitter occurs
// ∧ denotes the logic-and operation and mod(·) the modulus operation
if (counter_dec ≥ T) ∧ (|index(n+1) − index(n)| > 10) then
    v(n+1) = 0;
    k = min(k+1, N/q);
    T = τ_k;
    context = mod(context, N/q) + 1;
    // re-testing if all trial times have failed
    if k == N/q then
        k = 1;
    end
    counter_dec = 0;
    counter_inc = 0;
end
// schedule and counter resetting
if counter_inc ≥ T then
    k = 1;
    T = τ_1;
    counter_dec = 0;
    counter_inc = 0;
end
```

Fig. 4.7 Suboptimal-Context Escape Algorithm.

response has changed, the peak delay estimator will take longer to locate the peak corresponding to the new impulse response. The goal of this section is to reduce the amount of time it takes the peak delay estimator to track new dispersive regions in these situations.

One of the key properties of the coupled echo canceller structure in Fig. 3.1 is that the partial Haar filter is not being used to actually cancel echo. This allows some degree of flexibility in manipulating the operation of the filter, particularly for purposes of enhanced tracking performance.

Analyzing equations (3.33) and (3.34) in the event of a change in the echo path impulse response, one finds that although the location of the new peak as well as its steady-state magnitude is unknown, the new steady-state magnitude of an old peak is approximately zero. Therefore, if a decrease in the current peak's magnitude is detected (which usually signals a change in the echo path impulse response), it may be simpler to reset the whole partial Haar filter to $\mathbf{v}(n) = \mathbf{0}$. This allows for the new peak to solely compete with the low-magnitude coefficient noise of its neighbouring taps instead of the decreasing magnitude of the previous peak. As a result, the performance gains shown by Bershad and Bist [4] in the stationary case can be extended to cases where abrupt changes in the echo path impulse response occur.

In order to prevent the algorithm from constantly resetting the partial Haar adaptive filter, the reset operation should only be performed every T_{RS} samples. In addition, a reset is only deemed necessary when a decrease in magnitude is detected for a peak whose tendency has been in an increasing state for an amount of time greater then T_{inc}. It is usually the case that $T_{RS} < T_{\text{inc}}$.

One remaining problem, however, is that the improved tracking algorithm might falsely detect a change in the impulse response. This is due to the stochastic driving force $\phi_k(n)$ in (2.40) that causes the filter coefficients to display a form of Brownian motion around the optimal solution [22, 36]. In order to prevent these false alarms, the algorithm cannot simply reset the filter to zero. Instead, it also stores the position of the previous peak before resetting the filter, and feeds this position to the bulk delay unit. Once the new peak tendency has been increasing for an amount of time equal to T_{inc}, the algorithm uses the location of the new peak to centre the short time-domain

filter. The value of T_{inc} is set so as not to cause jitter in the peak delay estimate being fed to the bulk delay in the case of a false alarm.

Figure 4.8 shows the improved tracking algorithm combined with the suboptimal-context escaping algorithm.

4.5 Distributed Peak Detection for Cancellation of Multiple Echoes

Cancellation of multiple echoes can be a difficult task, particularly since no prior information is available concerning the number of dispersive regions. The method proposed here is based on a coordinator-multi-agent architecture similar to [10].

The proposed multiple echo canceller is based on two assumptions,

1. There is an average bound on the length of dispersive regions in the time domain.
2. The delay between reflections is larger than this upper bound so that almost no overlap occurs between the dispersive regions of the channel impulse response.

The first assumption is reasonable and is backed by empirical data [1]. Most network hybrid impulse responses range from 3–12 ms, or a corresponding length of 24–96 samples at a sampling rate of 8 kHz [1]. To simplify the current analysis and the ensuing simulations in Chapter 5, the length of the dispersive region will be taken as $L = 128$ samples at 8 kHz, which is the assumed length of the short time-domain filter. The second assumption prevents problems where a delay between echoes is such that there is a 30–50 percent overlap between dispersive regions, in which case assumption one is violated, since the combined echoes can interpreted as one longer dispersive region. A block diagram of the multiple echo canceller is shown in Fig. 4.9.

The partial Haar adaptive filter is divided into N/L partitions, each of length qL/N, where L is the expected length of a dispersive region in the time-domain (as well as the length of the short time-domain adaptive filter). Associated with each partition is a region that *overlaps* with its left-hand neighbouring partition. Of course, the first (leftmost) region simply corresponds to that partition.

The amount of overlap is set so as to create a region-length that is divisible by 3. The second assumption stated at the beginning of this section ensures that at

Given: See given data in Fig. 4.7,
rerouting variables peak_reroute, T_{RS}, T_{inc} and counter,
maximum-magnitude tap-weight at time $n+1$, $v_{\max}(n+1)$,
current overall maximum tap-weight magnitude and position, max_peak and max_index.
Initial: (at $n=0$), $\mathbf{v}(0)=\mathbf{0}$, counter_inc = counter_dec = 0, $k=1$, $T=\tau_1$, context = 1, max_peak = 0, max_index = 1.

```
// at time n+1 ...
```
insert : lines 1-16 from Fig. 4.7

```
// schedule and counter resetting
```
if counter_inc $\geq T$ **then**
    $k = 1$;
    $T = \tau_1$;
end

```
// re-routing of peak location:  old vs. new
```
if (peak_reroute == 1) $\wedge$ (counter_inc $< T_{\text{inc}}$) **then** `// use old peak position`
    $index(n+1)$ = max_index;
else `// use new peak position`
    peak_reroute = 0;
    counter = counter + 1;
end

```
// poll the maximum peak magnitude for any decrease
```
if counter $\geq T_{RS}$ **then**
    counter = 0;
    if $|v_{\max}| \geq$ max_peak **then**
        max_peak = $|v_{\max}|$;
    end
    if ($|v_{\max}| <$ max_peak) $\wedge$ (counter_inc $\geq T_{\text{inc}}$) **then**
        peak_reroute = 1;
        $k = 1$;
        $T = \tau_1$;
        counter_dec = 0;
        counter_inc = 0;
        $\mathbf{v}(n+1) = \mathbf{0}$;
        max_index = $index(n+1)$;
        max_peak = 0;
    end
end

Fig. 4.8 Improved tracking algorithm with suboptimal-context escaping.

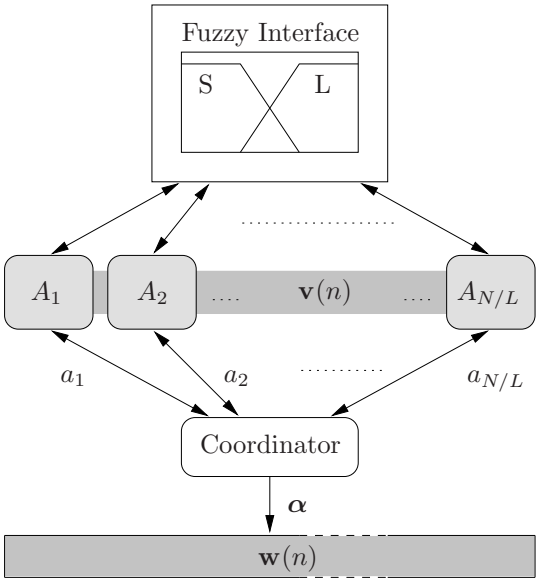


Fig. 4.9 Multiple echo cancellation system architecture.

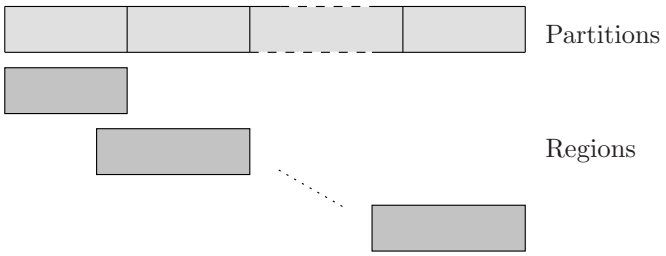


Fig. 4.10 Illustrating the difference between partitions and overlapping regions.

most one dispersive region can be located within any given region. The reason for including such a constraint will become clearer later when the issue of dispersive regions occurring near boundaries of partitions is considered. The difference between partitions and regions is shown in Fig. 4.10.

For example, in the case of a length $N = 1024$ time-domain filter (see Fig. 4.9) and corresponding partial Haar filter of length $q = 256$, the number of partitions is $N/L = 1024/128 = 8$ where each partition is of length $q/8 = 256/8 = 32$. To make the region-lengths divisible by 3, each region contains its respective partition and 7 samples from its left-hand neighbour's partition, resulting in a region length of $32 + 7 = 39$ which is divisible by 3.

Associated with each region is an agent A_i that runs a distributed version of the suboptimal-context escape algorithm. Although similar to the single-echo version of the algorithm, the distributed form departs slightly in its calculation of the PDM, which is given by

$$\mathrm{PDM}_i = 1 - \frac{\min(c_{i\,\max}, \tilde{c})}{c_{i\,\max}}. \tag{4.32}$$

Unlike the case of a single dispersive region in Section 4.3.3, each agent sends $c_{i\,\min}$ to the coordinator (see Fig. 4.9) where $\tilde{c}$ is calculated as $\tilde{c} = \max_i c_{i\,\min}$. The term $\min(c_{i\,\max}, \tilde{c})$ is necessary because $\tilde{c} > c_{i\,\max}$ in some cases, making the original PDM negative. Part of the motivation for using cooperative, rather than independent agents has to do with conserving the global characteristic of the PDM.

Escaping suboptimal contexts is complicated by the presence of multiple echoes, since each agent has control over only a finite number of tap weights. Therefore, if a certain agent requires a context switch, only its corresponding partition's context should be changed. This is where the efficient calculation of the partial Haar coefficients from Section 4.1 comes into play, since each set of coefficients is readily available.

In case a peak occurs near a boundary between two partitions, it would be necessary for the two partitions sharing this boundary to remain in the same context. This is because depending on the contexts at the boundary of two partitions, the partial Haar transform matrix can lose its row-orthogonality, since basis functions can overlap at these boundaries depending on the contexts selected. There are two cases that can result from two neighbouring partitions operating in different contexts.

In the first case, the context number of an agent's partition is greater than or equal to the context number of its left neighbour, in which case no overlap occurs at the boundary. In the second case, the context number is less than its left neighbour's number, resulting in overlap and smearing of the transformed Wiener solution in (3.4) at the boundary. Therefore, overlapping regions ensure the same operating context for two contiguous agents that share a peak near their boundary.

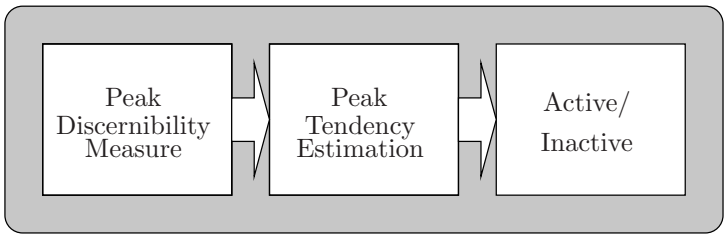


Fig. 4.11 Multiple echo cancellation agent architecture.

Figure 4.11 shows the architecture of each agent, which consists of three stages that are repeated every iteration. The first step consists of calculating the PDM associated with the peak located in each agent's region. Depending on the dominant peak behavioural model, each agent is classified as active or inactive. This information is stored in a length-N/L indicator vector $\mathbf{a}$ of active agents. If $a_i = 1$, then agent A_i is classified as active; otherwise if $a_i = 0$, the agent is deemed inactive. If agent i's peak tendency has been in an increasing state for more than T_{inc} samples, then a_i is set to 1, and the agent is classified as active. T_{inc}, is used to prevent false or transient classifications of agents.

Using information about the state of an agent (active or inactive), the central coordinator does one of two things. If none of the agents are active, i.e. $a_i = 0$ for all i, then the coordinator signals for all N time-domain adaptive filter coefficients to be updated. If, however, at least one of the agents is active, then the coordinator uses the peak locations of those agents to update only certain tap-weights of the time-domain adaptive filter. The activated tap-weights are centred around the peak delay estimates provided by each of the active agents. Therefore, the NLMS update equation of the time-domain adaptive filter is modified to

$$\mathbf{w}(n+1) = \text{diag}(\boldsymbol{\alpha})[\mathbf{w}(n) + \frac{\mu}{\|\text{diag}(\boldsymbol{\alpha})\mathbf{u}(n)\|^2} e(n)\mathbf{u}(n)]. \tag{4.33}$$

where $\boldsymbol{\alpha}$ is a length-N indicator vector whose elements α_i are used to either activate specific regions of the time-domain adaptive filter, or to simply update the entire adaptive filter.

Unfortunately, the improved tracking algorithm in Section 4.4 cannot be incorporated into the multiple echo canceller. In the case of multiple echo cancellation, a

Given: T_{inc}, $\boldsymbol{\tau} = \{\tau_1, \tau_2, \ldots, \tau_{N/q}\}$,
For each agent j,
peak indices at times n and $n+1$: $index(j,n)$, $index(j,n+1)$
contexts: context(j)
peak-tendency counters: counter_inc(j) and counter_dec(j).
Initial: (at $n=0$), counter_inc(j) = counter_dec(j) $= 0$, $k(j) = 1$, $T(j) = \tau_1$,
context(j) $= 1$ for all j.

`// at time` $n+1\ldots$
the coordinator returns $\tilde{c}$ *from the broadcasted values* $c_{i\,\min}$ *of each agent*;
$\mathbf{a} = \mathbf{0}_{N/L}$;
$\boldsymbol{\alpha} = \mathbf{0}_N$;
for *each of the* N/L *agents* **do**
calculate PDM($n+1$) *according to* (4.32);
insert : code from Fig. 4.6
insert : lines 1-16 from Fig. 4.7
if counter_inc(j) $\geq T(j)$ **then**
$k(j) = 1$;
$T(j) = \tau_1$;
end
`// activate or deactivate an agent`
$a_j = 0$;
if counter_inc(j) $\geq T_{\text{inc}}$ **then**
$a_j = 1$;
set the appropriate elements of $\boldsymbol{\alpha}$ *to* 1 *using (3.39)*;
end
end
`// if none of the agents are activated`
if $\sum a_i == 0$ **then**
$\boldsymbol{\alpha} = \mathbf{1}$;
end
use (4.33) to update the time-domain adaptive filter;

Fig. 4.12 Multiple echo canceller.

detected change in the current peak can also be associated with the disappearance of the current reflection in the given parition. Therefore, resetting a single partition is not feasible.

Chapter 5

Simulation Results and Discussion

Chapters 3 and 4 looked at the major drawbacks of the echo canceller introduced by Bershad and Bist [4], and proposed feasible solutions to these problems. The current chapter uses simulation experiments to build upon these critiques and proposals in two ways. First, it exemplifies the argued pitfalls of the original system, which include convergence and tracking issues. At the same time, the merits of the proposed algorithms in tackling these issues are shown. In addition to simulations, this chapter also analyzes the computational complexities of two of the main algorithms in Chapter 4.

5.1 Experimental Methodology

The following section introduces simulation parameters and the various filtering algorithms that will be used to compare with the algorithms proposed in Chapter 4.

5.1.1 Simulation Data

The set of hybrid impulse responses used in the following simulations are taken from Annex D of the ITU-T G.168 Recommendation for digital network echo cancellers [1]. There are eight hybrid impulse responses $m_i(n)$ (for $i = 1, 2, \ldots, 8$) of lengths L_i that range from 64 to 128 samples at 8 kHz. It is assumed that $m_i(n) = 0$ for $n \notin \{0, 1, \ldots, L_i - 1\}$. Impulse responses $i = 1$ to 4 were generated from a network hybrid simulator, while the last four consist of measured impulse responses from telephone networks in North America.

The echo path impulse responses $g(n)$ are generated according to [1]

$$g(n) = (10^{-\mathrm{ERL}/20} k_i) m_i(n - \Delta), \tag{5.1}$$

where ERL is the echo return loss, k_i is a scaling factor that depends on $m_i(n)$, and Δ represents the bulk delay. In addition, the Recommendation provides guidelines for testing echo cancellers, which includes different values of various parameters including the ERL. For the purpose of the presented simulations an ERL of 15 dB will be assumed for all hybrid impulse responses. Furthermore, ERL is not adjusted with bulk delay so as not to over-complicate the discussion. The value of Δ will be set so that the length of the bulk delay plus the dispersive region does not exceed 1024 samples.

The length of the partial Haar adaptive filter will be $q = 256$, unless stated otherwise. Longer filters ($q = 512$) will not be considered here since their use is computationally expensive, while the resulting performance of shorter filters is simply not up to par with intermediate cases ($q = 128$ or 256). The short time-domain adaptive filter length is set to $L = 128$ to match the longest dispersive region. When using a full time-domain adaptive filter in the multiple echo case, its length is set to $N = 1024$ to match the longest possible echo path impulse response.

The input $u(n)$ is a zero-mean Gaussian process with unit variance. The additive measurement noise $\nu(n)$ is also a white Gaussian process of zero-mean and variance $\sigma_\nu^2 = 10^{-\mathrm{SNR}/10}$. The samples of $\nu(n)$ are also assumed to be uncorrelated with the input. The SNR is set to 30 dB, unless stated otherwise. All simulation data and programs were generated and written in MATLAB®.

A note should be made concerning the terms 'optimal', 'best', 'suboptimal', and 'worst' associated with the partial Haar adaptive filter's (shift) context. Optimal and suboptimal contexts correspond to those combinations of contexts and bulk delays that produce fast and slow localizations of dispersive regions, respectively. The best and worst contexts of a given impulse response produce the fastest and slowest localization of a dispersive region, respectively. The above categories will only be used when the input is uncorrelated and Gaussian, and are obtained by applying the N/q shifted-versions of the partial Haar transform to an echo path impulse response of interest. The contexts can then be sorted according to the resulting partial Haar

peak magnitudes, where the largest magnitude corresponds to the best context, for a given impulse response.

5.1.2 Adaptive Filtering Algorithms

The original coupled echo canceller, which is also termed the partial Haar dual adaptive filter in [4], will be abbreviated as PHDAF. When validating the performance of each of the proposed improvements, different filtering algorithms will be used used for comparison. Of course, deciding which algorithms to benchmark against depends on which modification is being tested.

When testing the merits of escaping suboptimal contexts, the proposed suboptimal-context escape algorithm (SCE-PHDAF) will be compared to both the PHDAF, in addition to the NLMS. In this case, the NLMS will simply serve as a reference. In the case of studying the performance of the improved tracking algorithm with sub-optimal context escaping (IT-PHDAF), the learning curves of the IT-PHDAF will be compared to both the PHDAF and NLMS.

Although the PHDAF does not support multiple echoes, the multiple echo canceller's (ME-PHDAF) performance will be compared to it in the case of a single echo. In addition to comparing the ME-PHDAF to the NLMS algorithm, an ideal-NLMS (INLMS) algorithm will also be used to provide an optimal performance bound on the ME-PHDAF for the multiple echo case. The INLMS algorithm is the standard NLMS algorithm, except in that it adapts only those filter coefficients centred around the locations of the true impulse response's dispersive regions. This corresponds to the case where prior knowledge about the echo locations exists.

The step-size used here is $\mu = 1$ for all forms of the NLMS algorithm (these include all forms of the partial Haar adaptive filter, the INLMS, and short-time domain filters), and corresponds to the optimal step-size of the NLMS algorithm. At the beginning of each run, the input data vector is initialized with the first N input samples.

Concerning the calculation of the normalizing factor $\|\mathbf{u}(n)\|^2$ in Section 3.3.2, some remarks are in order. First, this power term is calculated using an efficient recursion on the input data block

$$\|\mathbf{u}(n)\|^2 = \|\mathbf{u}(n-1)\|^2 - u^2(n-N) + u^2(n). \tag{5.2}$$

The above equation may suffer from linear error buildup as n increases. However, for the purposes of these simulations, it was found that the simulation lengths used were not sufficiently long enough to cause instability.

5.2 Suboptimal-Context Escaping

Studying suboptimal-context escaping requires the use of specific examples, in addition to simulations that can reliably support the merits of using the SCE-PHDAF in general. The simulations shown below all use an input data vector of length $N = 1024$ with a partial Haar adaptive filter of length $q = 256$ and a schedule (see 4.3.3), $\boldsymbol{\tau} = \{150, 250, 300, 400\}$. The NLMS algorithm is used as a basis of comparison for both the proposed algorithm and PHDAF. The initial context is set to 1, which corresponds to no shifting of the partial Haar basis vectors.

5.2.1 Specific Cases

Figures 5.1(a) and (b) show the learning curves corresponding to an echo path impulse response using ITU-T G.168 hybrid model $m_5(n)$ under the best and worst bulk delays (for the initial context), respectively. The curves represent an ensemble of 200 runs for each simulation.

For the best-case bulk delay (Fig. 5.1(a)), both the SCE-PHDAF and PHDAF show identical learning curves reaching steady state at around $k = 750$ compared to the NLMS which converges at around $k = 5000$. Of course, this is related to the fact that the NLMS adapts a far larger number of coefficients (1024 compared to 128). In the case of a worst-case bulk delay (Fig. 5.1(b)), the PHDAF never seems to reach steady-state, while the proposed SCE-PHDAF converges much faster, nearly as well as in the optimal case, requiring about $k = 1000$ samples to converge.

5.2.2 General Case

The previous learning curves have shown promising performance gains for the SCE-PHDAF in specific cases. However, to fully accept the above results requires a more exhaustive test. Figure 5.2 shows the learning curves averaged over a thousand different echo path impulse responses. Each run randomly selects one of eight ITU-T

G.168 hybrid impulse response with equal probability, and uniformly selects a bulk delay in the interval [0,895]. The performance of the proposed echo canceller does not appear as impressive as in the specific cases, although it is still faster than the PHDAF. Again, the NLMS performs the poorest of all.

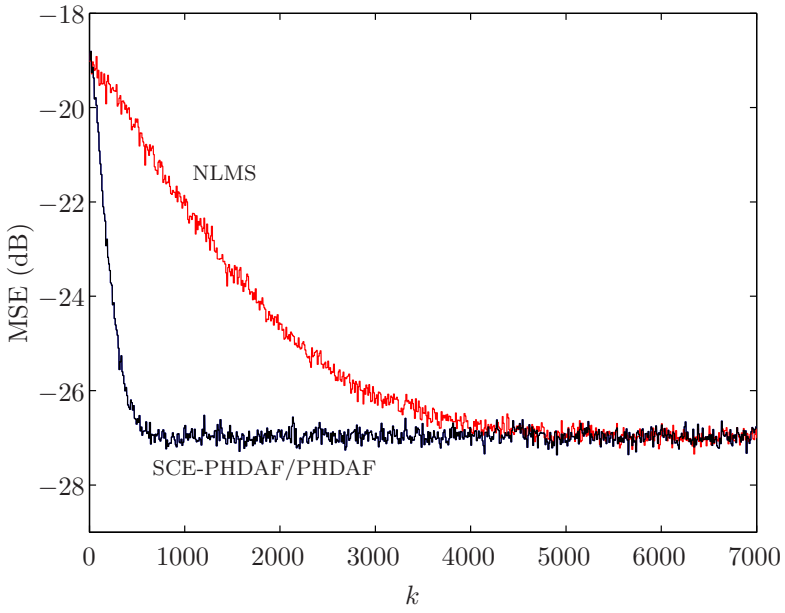
In addition to learning curves, the mean time for each echo canceller (proposed and PHDAF) to correctly estimate the location of a dispersive region was compared for different SNRs. Each row in Table 5.1 consists of the average and standard deviation over 500 random runs similar to the general case. In all cases, the proposed algorithm finds the dispersive region faster (the mean time is smaller) and more consistently (the standard deviation is much smaller). Although both PHDAF and the proposed echo canceller display similar mean times to convergence at very low SNR, the standard deviation of the PHDAF is three times larger.

5.2.3 Discussion

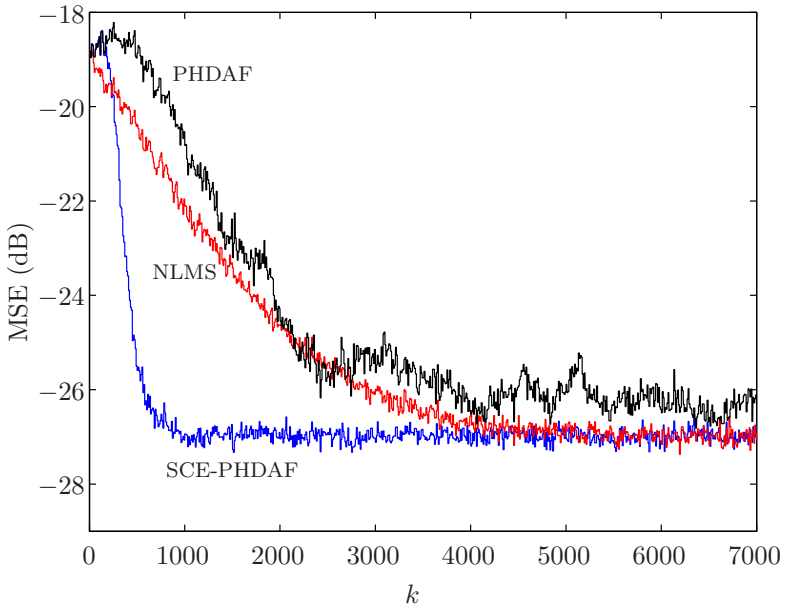
Three major observations need to be made concerning the results in Figs. 5.1-5.2. First, the specific simulations in Fig. 5.1 show that the proposed algorithm adds flexibility to the PHDAF in that it does not get trapped in suboptimal contexts. At the same time the proposed algorithm remains in optimal contexts. This behaviour may significantly increase the convergence speed of the echo canceller, as observed.

Second, the results in Fig. 5.2 also provide additional insights into the wavelet transform. It turns out that out of the 1000 runs, only 6.4% actually consisted of a worst-case combination of bulk delay and hybrid impulse response. This explains why the PHDAF and the proposed algorithm's learning curves in Fig. 5.2 are somewhat similar. However, in specific cases, like Fig. 5.1(b), the difference between the PHDAF and SCE-PHDAF is substantial, with the SCE-PHDAF showing great improvements.

Finally, Table 5.1 reveals the robustness of using a fixed schedule $\boldsymbol{\tau}$ together with a peak tendency estimator for different values of SNR. Of course, if the SNR does not change much over a specific channel, then schedules can be constructed specifically for those cases.



(a)



(b)

Fig. 5.1 ITU-T hybrid response $m_5(k)$ learning curves using a: (a) best and (b) worst-case bulk delay for the initial context used. (SNR = 30 dB)

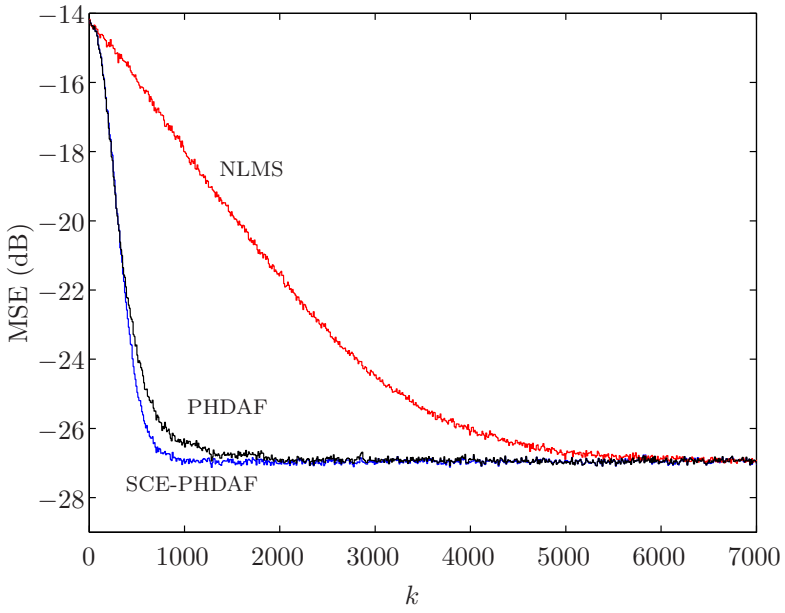


Fig. 5.2 Learning curves over 1000 runs, with randomly-selected (equiprobable) hybrid impulse responses and uniformly-selected bulk delays. (SNR = 30 dB)

Table 5.1 Comparison of mean times and standard deviation to correctly estimate the peak delay for different SNRs.

	SCE-PHDAF		PHDAF	
SNR (dB)	Mean	Std.	Mean	Std.
30	91.5	75.4	121.1	203.0
20	107.7	86.4	214.5	664.0
15	167.4	138.3	362.7	1067.7
10	421.4	387.1	531.7	1177.2

5.3 Improved Tracking of Changes in an Echo Path

The simulations described below consist of four scenarios where an abrupt change in the bulk delay, and possibly the hybrid impulse response occurs around time $k = 5200$. The waiting periods T_{RS} and T_{inc} are set to 32 and 128 samples, respectively. Again, the initial context in Fig. 4.8 is set to 1. Each scenario is categorized according to the optimality of its bulk delays (before and after the change) with respect to the initial shift context. For example, 'best-to-best' denotes a change from a best bulk delay to another best-case bulk delay relative to the initial context. The adaptive filtering parameters are the same as in the previous section.

5.3.1 Results

Best-to-best (Fig. 5.3(a)): Initially, both proposed IT-PHDAF and PHDAF converge very fast, similar to the previous section. However, after an abrupt change in bulk delay occurs, the IT-PHDAF resets the partial Haar filter, resulting in a convergence rate similar to the initial case. In contrast, the PHDAF requires almost 1500 samples to converge.

Worst-to-worst(Fig. 5.3(b)): Again, the IT-PHDAF converges quickly both initially and after a change in bulk delay, while the convergence characteristics of the PHDAF are poor in both stages.

Best-to-worst(Fig. 5.4(a)): Initially, both echo cancellers converge optimally. However, after the abrupt change in bulk delay, the PHDAF requires much more time to begin to converge after a sudden change in the bulk delay. This plateau-region extends for almost 2000 samples as compared to only about 1000 samples in Fig. 5.3(a). The IT-PHDAF, on the other hand, converges faster and its learning curve behaves similarly to the case in Fig. 5.3(a).

Worst-to-best(Fig. 5.4(b)): Again, the initial learning curves' behaviour is similar to the initial case in Fig. 5.3(b). However, at the transition point, the IT-PHDAF is slightly slower than the PHDAF, although both converge very quickly (requiring less than 1000 samples).

5.3.2 Discussion

The above results can be explained using the theory presented in Section 3.2.2. Only the behaviour at the transition points will be discussed since the initial behaviours were already explained in Section 5.2.

In Figures 5.3(a) and 5.4(a), the PHDAF's learning curve exhibits a plateau after the change in bulk delay. The length of this plateau equals the amount of time it takes the new peak's magnitude to exceed the decreasing magnitude of the old peak. The plateau region is slightly longer in Fig. 5.4(a) (which corresponds to a best-to-worst transition) because the time required by the peak delay estimator to track the new peak (which is very small) equals the amount of time it takes the old peak to become buried in coefficient noise, unlike the case in Fig. 5.3(a) where a new peak emerges while the old peak is decreasing. This can also be explained using (3.37), where $\beta \approx 0$ for the abrupt change in Fig. 5.4(a), and therefore, tracking this change takes longer.

In both of the above cases, the IT-PHDAF converges much faster because it resets the partial Haar adaptive filter once it detects a decrease in old peak's magnitude. Its converge time is still not as fast as in the initial case, however, because of T_{inc}. To elaborate, the new peak has to be categorized as increasing for T_{inc} samples before its index is accepted. During this time, the bulk delay unit continues to use the old peak location to offset the short time-domain filter.

In the worst-to-worst case of Fig. 5.3(b), it is interesting to observe that the PHDAF's learning curve does not exhibit a plateau as in Fig. 5.3(a). The reason for this, is that since the initial bulk delay is a worst-case delay, there is not much competition between the peak magnitudes before and after the transitions, and so they behave similarly.

In the final worst-to-best case, the PHDAF converges slightly faster after the abrupt change in bulk delay than the proposed algorithm. The reason is that a worst-case context is almost equivalent to a filter initialized to zero, i.e. $\mathbf{v}(n) \approx \mathbf{0}$. Once the bulk delay changes to an optimal value with respect to the initial context, the emergence of a new peak does not have to compete with the presence of an old peak.

There are two reasons why the IT-PHDAF requires a slightly longer amount of time to converge in the worst-to-best case. First, the improved tracking algorithm requires that a new peak's tendency be in an increasing state for T_{inc} samples until

it switches to the new peak's location. And second, the context escaping algorithm initially makes a context switch in response to the worst-case bulk delay. After the abrupt change in echo path impulse response, the new context might not be optimal with respect to the new bulk delay as the initial context.

5.3.3 Computational Complexity

The bulk of the suboptimal-context escaping and improved tracking algorithms' computational complexity lie with the peak tendency estimator of Fig. 4.6. Table 5.2 shows the number of arithmetic operations per iteration required by the peak tendency estimator, the suboptimal-context escaping echo canceller, and the improved tracking echo canceller (which includes suboptimal-context escaping). Since only the incremental increase in complexity over the PHDAF is of interest here, the complexity values of the first three columns of Table 5.2 (PTE SCE-PHDAF, IT-PHDAF) do not include the complexity associated with the PHDAF. The values in the second and third columns of Table 5.2 include the respective values corresponding to the peak tendency estimator. The last column is used as a reference to compute the percentage increase of incorporating either suboptimal-context escaping or improved tracking algorithms into the PHDAF.

Arithmetic operations associated the the PTE can be reduced drastically since many bbm terms such as $m_{n+1}(S \cap L)$, $m_{\text{pred}}(S \cup L)$, $m_{\text{pred}}(S \cup L)$, $m_{\text{pred}}(S \cap L)$, and $m_{\text{upd}}(S \cap L)$ (see Fig. 4.6) are zero throughout the PTE's operation. Calculating the new PDM value requires one division and addition operation. The remaining number of arithmetic operations are shown in Fig. 4.6 for each stage.

It appears that calculating the Pignistic entropies in (4.31) requires the use of two logarithm operations which are computationally expensive. However, the SCE-PHDAF and IT-PHDAF only need to measure the peak tendency (see line 15 of Fig. 4.7), which only involves a comparison of the Pignistic entropies $H_{\text{pig}}^{\text{inc}}$ and $H_{\text{pig}}^{\text{dec}}$. This calculation makes no direct use of the actual values. Therefore, an alternate measure that preserves the relationship between the two Pignistic entropies and requires only a single comparison operation is given by

$$\tilde{H}_{\text{pig}}^{M} = \min(P^{M}\{S\}, P^{M}\{L\}), \qquad M = \{\text{inc}, \text{dec}\}. \tag{5.3}$$

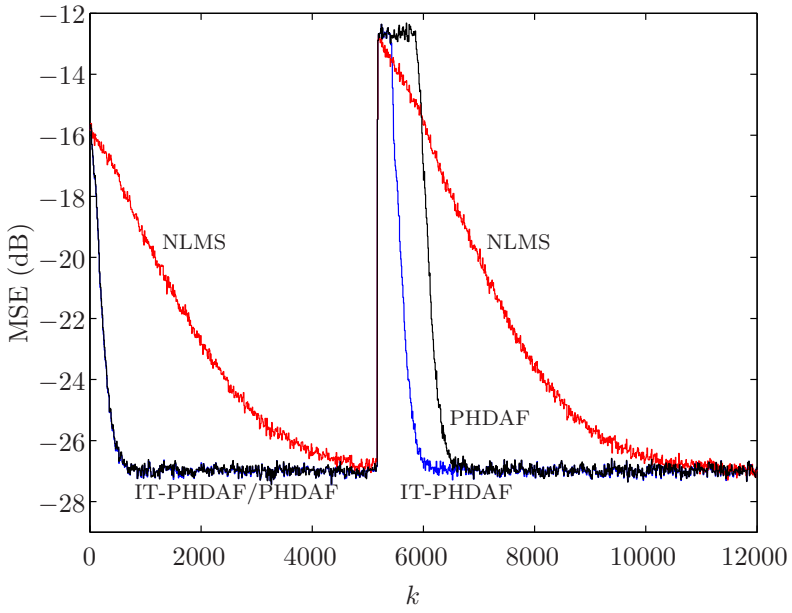
From information theory, the entropy of a source of two symbols is maximum when the occurrence of these symbols is equiprobable [38]. As the probability of one symbol decreases, the entropy of the source decreases. Therefore, the peak tendency model with the smallest minimum probabilityalso has the smallest entropy associated with its Pignistic probabilities.

To compare the percentage increase in complexity of the IT-PHDAF to that of the PHDAF employing an efficient partial Haar transform as in Sec. 4.1, consider an example. Letting $N = 1024$, $q = 256$, and $L = 128$, the percentage increase in complexity is $61/1548 = 3.94\%$, which is an acceptable amount.

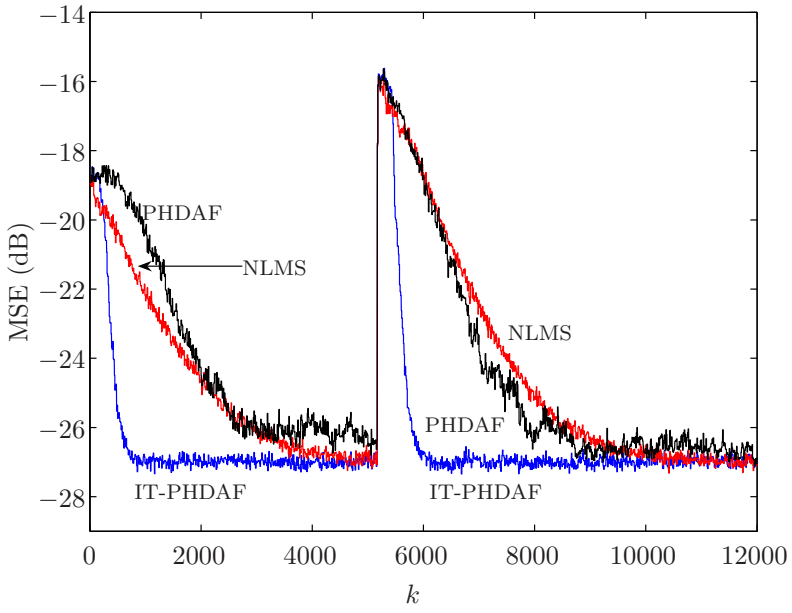
The above results have therefore shown that either suboptimal-context escaping or improved tracking (with suboptimal-context escaping) can considerably improve the performance of the coupled echo canceller without a drastic increase in computational complexity.

Table 5.2 Number of arithmetic operations per iteration - single dispersive region

	PTE	SCE-PHDAF	IT-PHDAF	PHDAF
Addition	17	18	18	$2q + 2L + N/q + 1$
Multiplication	18	18	18	$2q + 2L + 6$
Division	3	3	3	1
Comparison	11	16	22	0

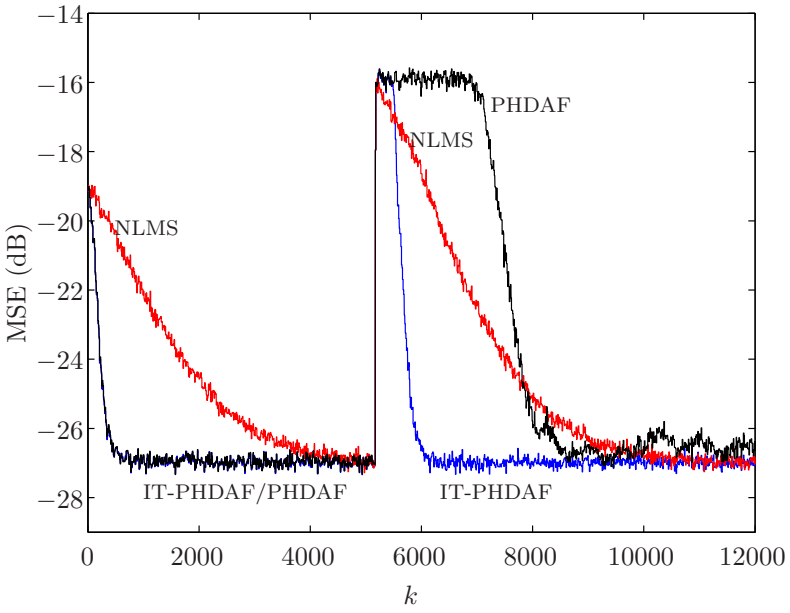


(a)

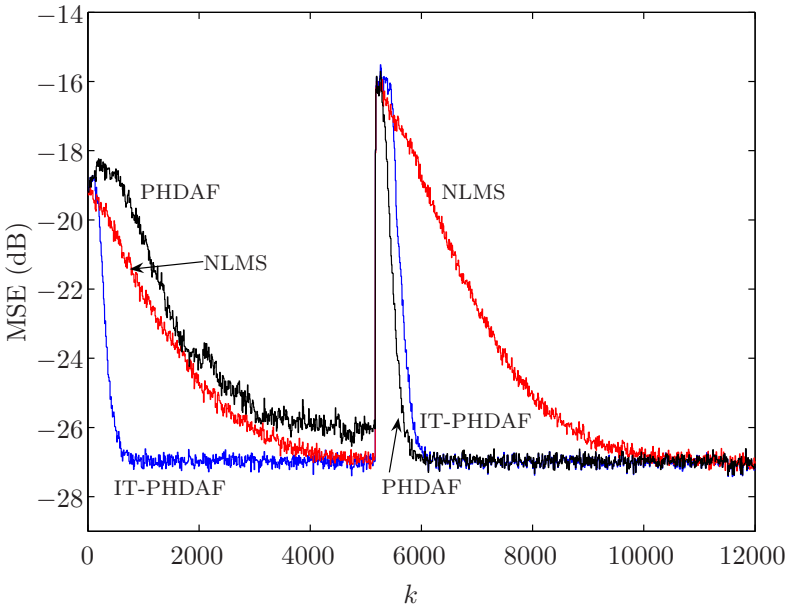


(b)

Fig. 5.3 Tracking behaviour of the IT-PHDAF compared to the PHDAF and NLMS algorithm: (a) Best-to-Best change in bulk delay for the initial context used; (b) Worst-to-worst change in bulk delay. (SNR = 30 dB)



(a)



(b)

Fig. 5.4 Tracking behaviour of the IT-PHDAF compared to the PHDAF and NLMS algorithm: (a) Best-to-worst case change in bulk delay for the initial context used; (b) Worst-to-best context change in bulk delay. (SNR = 30 dB)

5.4 Multiple Echo Cancellation

The multiple echo canceller (ME-PHDAF) of Section 4.5 is tested using a partial Haar adaptive filter of length $q = 256$, and a full time-domain adaptive filter of length $N = 1024$. Again, the dispersive regions length is assumed to be $L = 128$. This requires $1024/128 = 8$ agents operating in parallel to classify and locate dispersive regions. As mentioned in Section 4.5, although the suboptimal-context escaping algorithm is used by each agent, the improved tracking algorithm can not be integrated into the multiple echo canceller. The number of samples a peak has to be characterized as increasing before its corresponding agent is categorized as activate equals $T_{\text{inc}} = 128$ samples in Fig. 4.12.

5.4.1 Results

Cancelling multiple echoes is not directly supported by the PHDAF. Nevertheless, the learning curves in Figures 5.5(a) and (b) compare the ME-PHDAF's performance with a single dispersive region for the two cases in Fig. 5.1. Each learning curve was averaged over 200 runs. In the case of more than one dispersive region, the PHDAF will not converge.

For the best-case bulk delay in Fig. 5.5(a), the PHDAF is slightly faster than the ME-PHDAF, although both echo canceller's require only around 1000 samples to converge. However, again, the distributed form of the suboptimal-context escaping algorithm allows the ME-PHDAF to converge much faster in a worst-case bulk delay in Fig. 5.5(b). One can also observe that the steady-state MSE of the ME-PHDAF is slightly lower than that of the PHDAF.

Figure 5.6(a) shows the performance of the multiple echo canceller when the number of dispersive regions in the echo path impulse response is 2. The learning curves are averaged over 600 runs consisting of bulk delays and dispersive regions which are selected according to

$$h(n) = g_1(n) + g_2(n), \tag{5.4}$$

where $h(n)$ is the echo path impulse response, and

$$\begin{aligned} g_1(n) &= (10^{-15/20}k_i)m_i(n-\Delta_1) \\ g_2(n) &= (10^{-15/20}k_j)m_j(n-\Delta_2) \qquad i,j \text{ equiprobable in } \{1,2,\ldots,8\} \\ \Delta_1 &= 320+r_1 \\ \Delta_2 &= 640+r_2 \qquad r_1,r_2 \text{ equiprobable in } \{0,1,2,3\}. \end{aligned} \tag{5.5}$$

The INLMS filter represents the ideal case where prior knowledge about the location of these 2 dispersive regions is available, and provides the best possible performance in this case.

Two observations can be made from the above simulation. First, for $k < 500$, the ME-PHDAF's learning curve follows the NLMS, after which its convergence speed increases, reaching the INLMS curve's steady-state MSE at around $k = 2500$. Second, the multiple echo canceller displays a lower steady-state MSE again.

Figure 5.7(a) shows the performance of the multiple echo canceller when the number of dispersive regions in the echo path impulse response is 3. The learning curves were averaged over 1000 runs, and involved random bulk delays as well as randomly-selected dispersive regions given by

$$h(n) = \sum_{i}^{3} g_i(n), \tag{5.6}$$

where

$$\begin{aligned} g_i(n) &= (10^{-15/20}k_{\gamma_i})m_{\gamma_i}(n-\Delta_i), \gamma_i \text{ equiprobable in } \{1,2,\ldots,8\} \\ \Delta_1 &= 320+r_1 \\ \Delta_2 &= 640+r_2 \\ \Delta_3 &= 890+r_3 \qquad r_1,r_2,r_3 \text{ equiprobable in } \{0,1,2,3\}. \end{aligned} \tag{5.7}$$

For three dispersive regions, the learning curves converge slower for both INLMS and ME-PHDAF, and the MSE slightly increases for samples $k < 1000$ for the proposed algorithm.

5.4.2 Discussion

Three main observations can be made concerning the multiple echo canceller. The first concerns Fig. 5.5(a), and involves explaining why the ME-PHDAF is slightly slower than the PHDAF in this case. This phenomenon is a result of the ME-PHDAF's waiting of T_{inc} samples before it activates an agent and switches from a full time-domain adaptation (which follows the NLMS learning curve) to a single dispersive region (learning curve is similar to the PHDAF).

Second, in Fig. 5.6(a), it is observed that the ME-PHDAF's learning curve follows that of the NLMS algorithm initially. The reason is that the multiple echo canceller updates all $N = 1024$ taps of the time-domain echo canceller whenever none of the agents signals that it has correctly found a peak. In Fig. 5.7(a), the MSE actually increases initially as a result of initially locating fewer than 3 dispersive regions.

A third observations is the lower MSE values obtained by using the ME-PHDAF. To explain this, one has to remember that the ME-PHDAF does not make any assumptions about the number of occurring dispersive regions, (although assumptions made during the development of the canceller set an upper bound on the maximum number of dispersive regions to N/L). Therefore, during each simulation run, the number of active agents can change even after the partial Haar adaptive filter has converged. To appreciate this, consider the bar graph in Fig. 5.6(b), which shows the distribution over 600 random runs (see (5.4) and (5.6)) with respect to the dominant number of activated agents (this data excludes instances where the entire time-domain filter is adapted). Based on this graph, one can observe that the dominant number of activated agents ranges from 2 to 6 with the most frequent numbers equal to 2 and 3. The reason for this is that one of the dispersive regions is intentionally placed near the border of two partitions.

Figure 5.7(b) shows a similar distribution for the case of 3 dispersive regions. In this case, a dominant number of dispersive region for a run is $2 < 3$. Fortunately this case only occurs with probability 0.001.

Many of the sparse echo cancellers reviewed in Section 1.2, have shown a degradation in performance (sometimes worse than the NLMS) as the number of dispersive regions increases. The ME-PHDAF, on the other hand, performs better than the NLMS algorithm as the above results have shown. The cost of obtaining this im-

provement is analyzed in the next section.

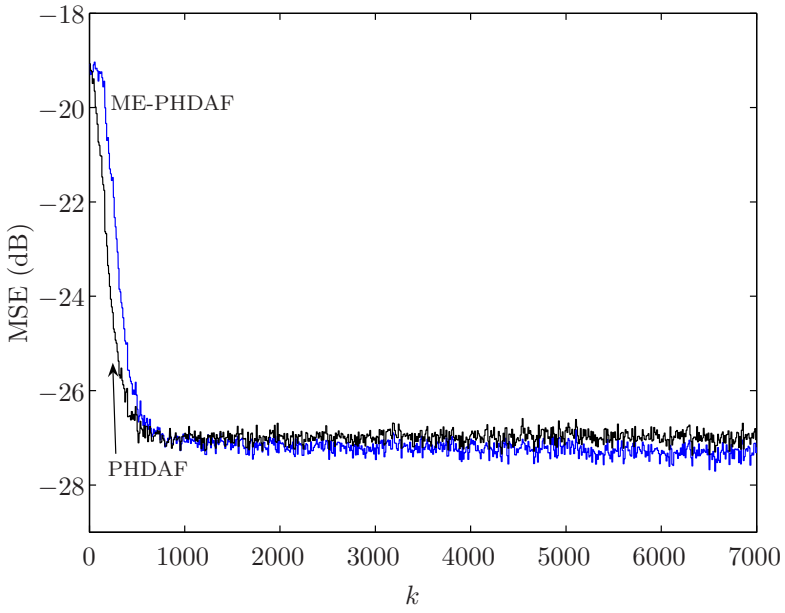
5.4.3 Computational Complexity

The computational complexity of the ME-PHDAF is shown in Table 5.3. For brevity, the results do not include the added complexity of the partial Haar adaptive filter of length q. The order of complexity of the ME-PHDAF is around N/L times the complexity of the SCE-PHDAF in Table 5.2. For example, if $N = 1024$ and $L = 128$, this results in an eight-fold increase in the number of arithmetic operations, which might be considered unacceptable. To reduce the computational load, the ME-PHDAF can be programmed to allow only one agent to estimate its peak tendency every input sample. In other words, the agents time-share the peak tendency estimator. This also requires scaling the trial periods in the schedule $\boldsymbol{\tau}$ by N/L and relevant counters in Fig. 4.12. The number of arithmetic operations per iteration required by this efficient form of the multiple echo canceller (the ME-PHDAF-E) is also shown in Table 5.3.

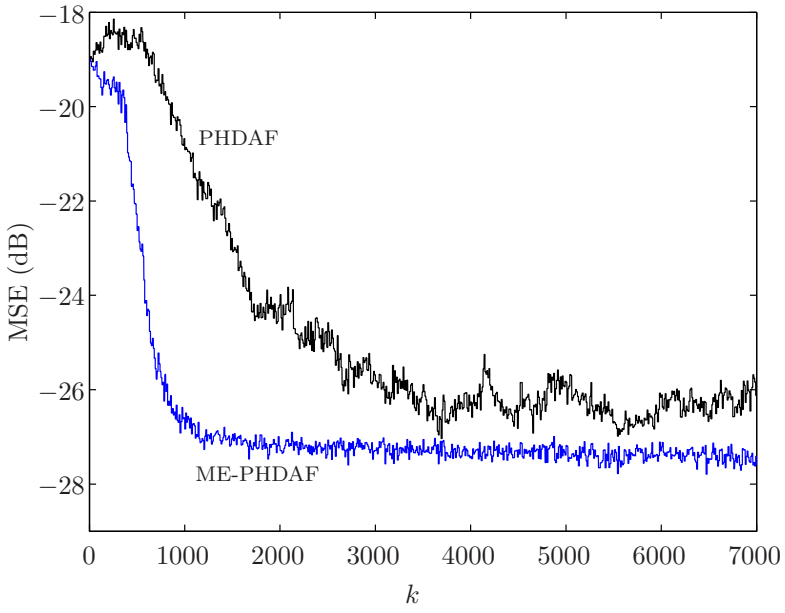
To compare the complexity of the ME-PHDAF to that of the NLMS, consider an example. Letting $N = 1024$, $q = 256$, and $L = 128$, the percentage increase in complexity per iteration over the NLMS algorithm is $(1034 + 455)/4102 = 36\%$, which is quite significant. The value 1034 is the cost of the partial Haar adaptive filter and transform. Comparing the ME-PHDAF-E to the NLMS under the same conditions yields a percentage in complexity of 26.5%. The larger part of the increase in percentage of the multiple echo canceller can be attributed to using a full time-domain adaptive filter of length N. The above complexity is still less than the multiple echo canceller proposed in [50], which uses a length N adaptive filter to track the echo path impulse response instead of a short partial Haar adaptive filter.

Table 5.3 Number of arithmetic operations per iteration - multiple dispersive regions

	ME-PHDAF	ME-PHDAF-E	NLMS
Addition	$18N/L$	18	$2N + 2$
Multiplication	$18N/L$	18	$2N + 3$
Division	$3N/L$	3	1
Comparison	$18N/L - 1$	$16 + N/L$	0

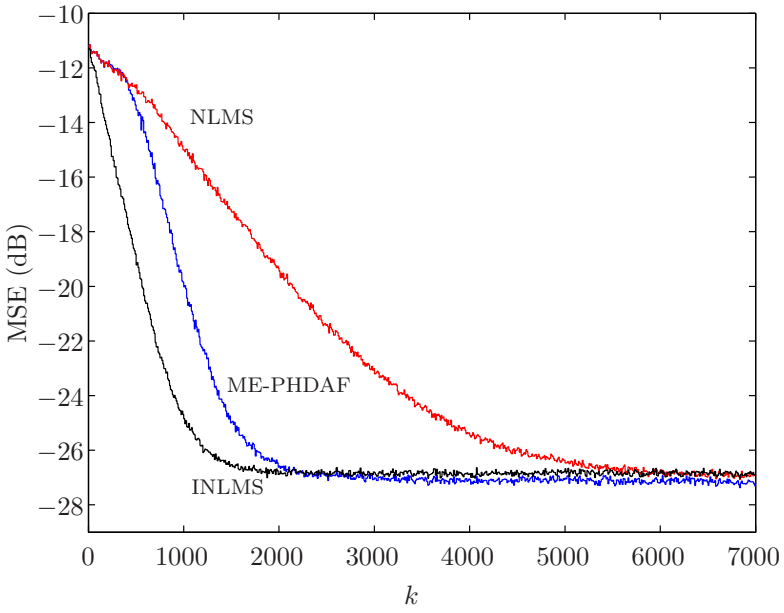


(a)

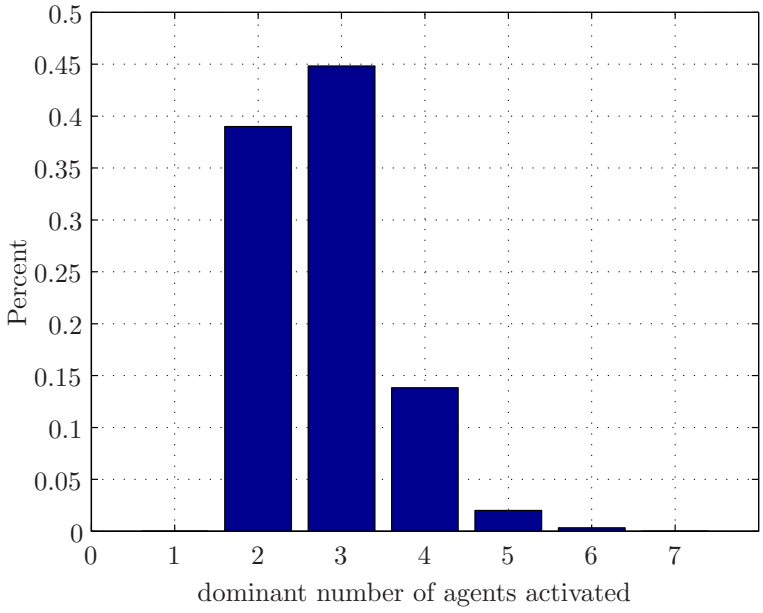


(b)

Fig. 5.5 ITU-T hybrid response $m_5(k)$ learning curves using a multiple echo canceller and the PHDAF: (a) best-case bulk delay; (b) worst-case bulk delay. (SNR = 30 dB)

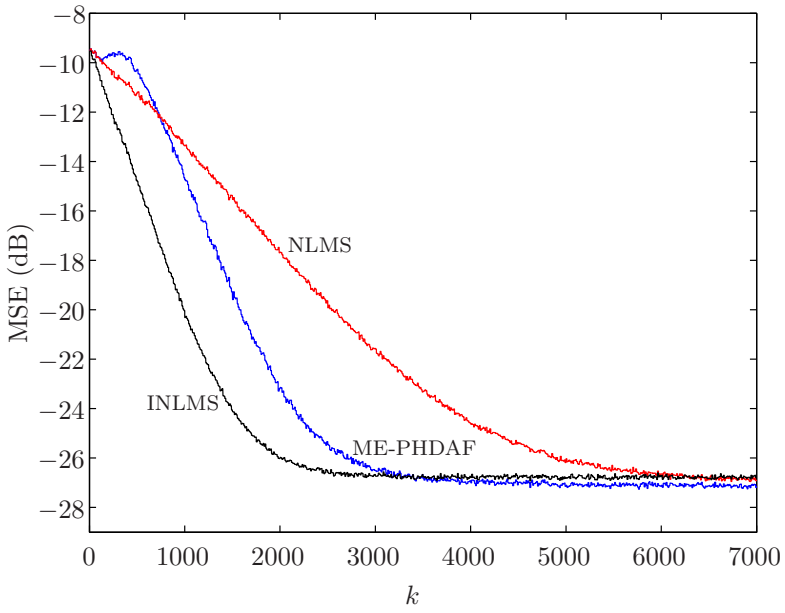


(a)

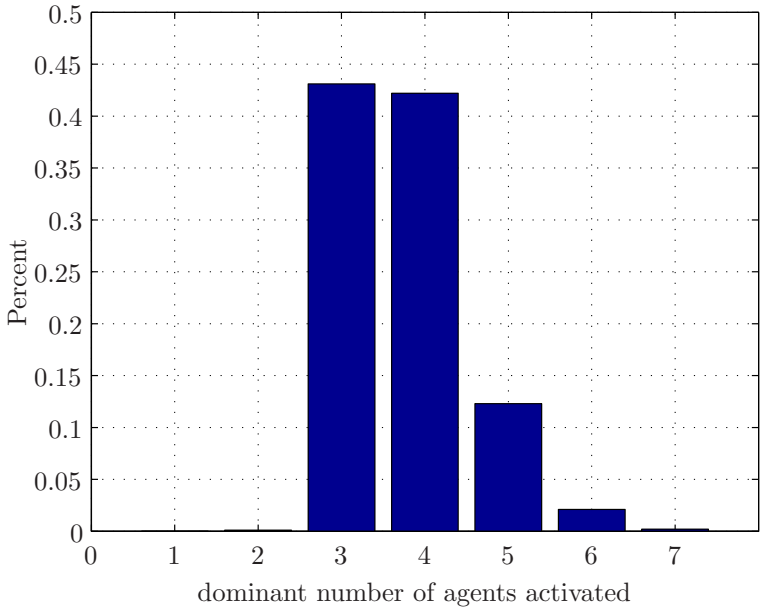


(b)

Fig. 5.6 (a) Learning curves over 600 runs using two random hybrid impulse responses and random bulk delays (SNR = 30 dB); (b) Corresponding percentage of dominant number of agents activated in 600 simulations.



(a)



(b)

Fig. 5.7 (a) Learning curves over 1000 runs using three random hybrid impulse responses and random bulk delays (SNR = 30 dB); (b) Corresponding percentage of dominant number of agents activated in 1000 simulations.

Chapter 6

Conclusion

6.1 Thesis Overview and Summary of Results

This thesis has looked at some of the issues of sparse echo cancellation utilizing a coupled configuration that employs the partial Haar wavelet transform [4]. It began by analyzing the convergence properties of the coupled configuration, and clarified some implementation issues which include positioning the short time-domain adaptive, a partial Haar NLMS algorithm, and the overall complexity of the original coupled echo canceller [4].

Furthermore, the bulk of this work has presented three major weaknesses of the coupled echo canceller [4]. First, the partial Haar adaptive filter converges to one of N/q Wiener solutions with different peak magnitudes, depending on a dispersive region's bulk delay. This can seriously hinder the peak delay estimator from correctly centring the short time-domain adaptive filter around the dispersive region. It was shown that for small peaks, it takes the peak delay estimator longer to correctly locate a dispersive region. In pathological cases, the peak delay estimator may take an extremely large number of input samples to locate a dispersive region. Second, it was analytically shown that in most instances, the peak delay estimator takes much longer to track an abrupt change in the echo path impulse response than to initially locate a peak. Finally, the coupled echo canceller proposed by Bershad and Bist [4] can not properly cancel multiple echoes.

To support the improvements developed to counter each of these drawbacks, a se-

ries of simulation tests were run in MATLAB®. Different ITU-T G.168 hybrid impulse responses were used to construct echo path impulse responses to test the proposed algorithms. These algorithms included the suboptimal context escaping algorithm (SCE-PHDAF), the improved tracking algorithm (IT-PHDAF), and a multiple echo cancelling version (ME-PDHAF) of the original coupled echo canceller.

The suboptimal context escape algorithm did not seem to provide dramatic improvements in convergence speed in the general case, since the worst-case contexts only made up a small percentage of the simulation runs. It was shown that for specific cases, the coupled echo canceller employing suboptimal context escaping was much faster than the original coupled echo canceller. In addition, the coupled echo canceller employing suboptimal context escaping was shown to out-perform the original echo canceller as the SNR dropped from 30 to 10 dB.

The improved tracking algorithm, which includes suboptimal context escaping was tested under four scenarios of abrupt change in the echo path impulse response. Each scenario depended on the optimality of the bulk delays (before and after the change) with respect to the initial context. In all but one scenario, the proposed improved tracking algorithm was shown to be faster than the original coupled echo canceller at a small increase in computational complexity (the bulk of this complexity was attributed to the peak tendency estimator). In these cases, the speed up in tracking ranged from approximately 500 to 2000 input samples. The case where the proposed algorithm was shown to be slower was attributed to the waiting time T_{inc} discussed in Section 4.4.

The multiple echo canceller was also shown to be sufficiently faster, as compared to the NLMS algorithm. The multiple echo canceller was first tested on an echo path impulse response consisting of a single dispersive region. It was shown that for a worst-case bulk delay, the multiple echo canceller converged much faster than the original coupled echo canceller [4], while for the best-case bulk delay the proposed echo canceller was slightly slower due to T_{inc}. In the case of two and three dispersive regions, the multiple echo canceller required around 500 input samples to converge to the same steady-state MSE as an ideal NLMS algorithm, which had prior knowledge about the locations of the dispersive region. Although much faster than the NLMS algorithm to reach steady-state, the multiple echo canceller was shown to be costly in

terms of arithmetic operations. An efficient form was therefore mentioned which can result in similar performance to the original multiple echo canceller, but at a lower computational cost.

6.2 Future Work

This thesis has tried to look at adaptive signal processing in a different way, combining belief-based evidence theory with fuzzy inference to incorporate a form of 'self-awareness' into an echo canceller. Incorporating forms of expert knowledge can help devices such as echo cancellers make decisions about their performance in different environments. And, if there is support to do so, these devices can then change the way they function to improve their performance within a specific environment without the aid of an external operator.

Foreseeable work with regards to the coupled echo canceller in addition to the proposed algorithms include:

- A more robust approach to decision making in suboptimal-context escaping that does not rely on a set of trial periods, but on some characteristic of the echo canceller's performance. This might require some assumptions about the noise power, MSE, etc.

- Increasing the flexibility of the multiple echo canceller so that it does not have to rely on one or more of the assumptions listed in Section 4.5.

- Extending the echo canceller with a subsystem that uses a form of spectral correlation to determine which subset of partial Haar basis vectors is suitable for a given echo path impulse response. This is especially important for impulse responses that are not rich in spectral content [24].

- Analyzing the coupled echo canceller with other classes of adaptive filter algorithms and different combinations thereof for the partial Haar and short time-domain adaptive filter.

Of course, one of the challenges faced with realizing some of the above proposals will be keeping the added computational costs as low as possible.

References

[1] "Digital network echo cancellers, ITU-T Recommendation G.168," International Telecommuniation Union, Tech. Rep., Aug. 2004.

[2] P. An, M. Brown, and C. Harris, "On the convergence rate performance of the normalized least-mean-square adaptation," *IEEE Trans. Neural Networks*, vol. 8, pp. 1211–1214, Sept. 1997.

[3] J. Benesty and S. L. Gay, "An improved PNLMS algorithm," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, vol. 2, May 2002, pp. 1881–1884.

[4] N. J. Bershad and A. Bist, "Fast coupled adaptation for sparse impulse responses using a partial Haar transform," *IEEE Trans. Signal Processing*, vol. 53, no. 3, pp. 966–976, Mar. 2005.

[5] J. R. Boston, "A fuzzy model of signal detection incorporating uncertainty," in *Proc. Second IEEE Conf. on Fuzzy Systems*, vol. 2, mar 1993, pp. 1107–1112.

[6] ——, "A signal detection system based on Dempster-Shafer theory and comparison to fuzzy detection," *Int. Journal Inform. and Security*, vol. 9, pp. 13–57, Feb. 2000.

[7] C. Breining, P. Dreiseitel, E. Hänsler, A. Mader, B. Nitsch, H. Puder, T. Schertler, G. Schmidt, and H. Tilp, "Acoustic echo control: An application of very-high-order adaptive filters," *IEEE Signal Processing Magazine*, pp. 42–69, July 1999.

[8] C. S. Burrus, R. A. Gopinath, and H. Guo, *Introduction to Wavelets and Wavelet Transforms, A Primer*, 1st ed. Prentice Hall, Inc., 1998.

[9] J. M. Cioffi and J. A. C. Bingham, "A data-driven multitone echo canceller," *IEEE Trans. Communications*, vol. 42, pp. 2853–2869, Oct. 1994.

[10] C. Cruz, D. A. Pelta, A. S. Royo, and J. L. Verdegay, "Soft computing and cooperative strategies for optimization," in *Proc. IEEE mid-summer Workshop on Soft Computing in Industrial Applications*, Espoo, Finland, June 2005.

[11] J. Cui, P. A. Naylor, and D. T. Brown, "An improved IPNLMS algorithm for echo cancellation in packet-switched networks," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, vol. 4, May 2004, pp. 141–144.

[12] H. Deng and M. Doroslovački, "Improving convergence of the PNLMS algorithm for sparse impulse response identification," *IEEE Signal Processing Letters*, vol. 12, no. 3, pp. 181–184, Mar. 2005.

[13] J. Dezert, "Foundations for a new theory of plausible and paradoxical reasoning," *Int. Journal Inform. and Security*, vol. 9, pp. 13–57, 2002.

[14] J. Dezert, F. Smarandache, and M. Khoshnevisan, "Counter-examples to Dempster's rule of combination," in *Advances and Applications of DSmT for Information Fusion (Collected Works)*, J. Dezert and F. Smarandache, Eds. American Research Press, 2004.

[15] ——, "Presentation of DSmT," in *Advances and Applications of DSmT for Information Fusion (Collected Works)*, J. Dezert and F. Smarandache, Eds. American Research Press, 2004.

[16] D. L. Duttweiler, "Subsampling to estimate delay with application to echo cancelling," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 31, pp. 1090–1099, Oct. 1983.

[17] ——, "Proportionate normalized least-mean-squares adaptation in echo cancelers," *IEEE Trans. Speech and Audio Processing*, vol. 8, no. 5, pp. 508–518, Sept. 2000.

[18] T. Frantii, "Timing of fuzzy membership functions from data," Ph.D. dissertation, Oulu University, June 2001. [Online]. Available: http://herkules.oulu.fi/issn03553213/

[19] S. L. Gay, "An efficient, fast converging adaptive filter for network echo cancellation," in *Proc. Thirty-Seventh Asilomar Conf. on Signals, Systems, and Computers*, vol. 1, Nov. 1998, pp. 394–398.

[20] P. H. Giang and P. P. Shenoy, "Statistical decisions using likelihood information without prior probabilities," in *Uncertainty in Artificial Intelligence*, A. Darwiche and N. Friedman, Eds. Morgan Kaufmann, San Fransisco, 2002, pp. 170–178.

[21] A. Graps, "An introduction to wavelets," *IEEE Computational Science and Engineering*, vol. 2, pp. 50–61, 1995.

[22] S. Haykin, *Adaptive Filter Theory*, 4th ed. Prentice Hall, Inc., 2002.

[23] S. Haykin and B. Widrow, *Least-Mean-Square Adaptive Filters*. John Wiley and Sons Inc., 2003.

[24] K. C. Ho and S. D. Blunt, "Rapid identification of a sparse impulse response using an adaptive algorithm in the Haar domain," *IEEE Trans. Signal Processing*, vol. 51, no. 3, pp. 628–638, Mar. 2003.

[25] M. Ho, J. M. Cioffi, and J. A. Bingham, "Discrete multitone echo cancelation," *IEEE Trans. Communications*, vol. 44, pp. 817–825, July 1996.

[26] J. James, B. Chen, and L. Garrison, "Implementing VoIP: A voice transmission performance progress report," in *IEEE Communications Magazine*, July 2004, pp. 36–41.

[27] S. Kawamura and M. Hatori, "A tap selection algorithm for adaptive filters," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, Apr. 1986, pp. 2979–2982.

[28] G. J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall, New Jersey, 1995.

[29] J. Kohlas and P. A. Monney, "Theory of evidence - a survey of its mathematical foundations, applications and computational anaylsis," *ZOR- Mathematical Methods of Operations Research*, vol. 39, pp. 35–68, 1994.

[30] J. Liang and T. W. Parks, "A translation-invariant wavelet representation algorithm with applications," *IEEE Trans. Signal Processing*, vol. 44, pp. 225–232, Feb. 1996.

[31] S. Mallat, *A Wavelet Tour of Signal Processing*. Academic Press, 1999.

[32] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, pp. 674–693, July 1989.

[33] R. K. Martin, W. A. Sethares, R. C. Williamson, and C. R. J. Jr., "Exploiting sparsity in adaptive filters," *IEEE Trans. Signal Processing*, vol. 50, no. 8, pp. 1883–1894, Aug. 2002.

[34] H. McKinlay, "A legacy of echo and silence," in *Telecomm. Quality of Services: The Business of Success, QoS 2004, IEE*, Mar. 2004, pp. 109–113.

[35] J. M. Mendel, "Fuzzy logic systems for engineering," *Proc. of the IEEE*, vol. 83, no. 3, pp. 345–377, Mar. 1995.

[36] A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Processes*, 4th ed. McGraw-Hill Science/Engineering/Math, 2001.

[37] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, 1988.

[38] K. Sayood, *Introduction to Data Compression*, 2nd ed. Morgan Kaufmann Publishers, 2000.

[39] I. W. Selesnick, "Hilbert transform pairs of wavelet bases," *IEEE Signal Processing Letters*, vol. 8, pp. 170–173, June 2001.

[40] ——, "The design of approximate hilbert transform pairs of wavelet bases," *IEEE Trans. Signal Processing*, vol. 50, pp. 1144–1152, May 2002.

[41] K. Shenoi, *Digital Signal Processing in Telecommunications*. Prentice Hall, Inc., 1995.

[42] M. J. Shensa, "The discrete wavelet transform: Wedding the à Trous and Mallat algorithms," *IEEE Trans. Signal Processing*, vol. 40, pp. 2464–2482, Oct. 1992.

[43] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger, "Shiftable multiscale transforms," *IEEE Trans. Inform. Theory*, vol. 38, pp. 587–607, Mar. 1992.

[44] P. Smets, "Decision making in the TBM: the necessity of the pignistic transformation," *Int. Journal of Approximate Reasoning*, vol. 38, pp. 133–147, Feb. 2005.

[45] P. Smets and R. Kennes, "The transferable belief model," *Artificial Intelligence*, vol. 66, no. 2, pp. 191–234, Apr. 1994.

[46] A. Sugiyama and S. Ikeda, "A fast convergence algorithm for adaptive FIR filters with sparse taps," *IEICE Trans. Fundamentals*, vol. E77-A, pp. 681–686, Apr. 1994.

[47] A. Tchamova, T. Semerdjiev, and J. Dezert, "Estimation of target behavior tendencies using DSmT," in *Advances and Applications of DSmT for Information Fusion (Collected Works)*, J. Dezert and F. Smarandache, Eds. American Research Press, 2004.

[48] S. V. Vaseghi, *Advanced Signal Processing and Digital Noise Reduction*. John Wiley and Sons Ltd. and B.G. Teubner, 1996.

[49] R. R. Yager and D. P. Filev, *Essentials of Fuzzy Modeling and Control*. John Wiley and Sons, Inc., 1994.

[50] P. C.-W. Yip and D. M. Etter, "An adaptive multiple echo canceller for slowly time-varying echo paths," *IEEE Trans. Communications*, vol. 38, pp. 1693–1698, Oct. 1990.

[51] L. A. Zadeh, "The role of fuzzy logic in the management of uncertainty in expert systems," *Fuzzy Sets and Systems*, vol. 11, no. 1, pp. 199–227, 1999.