University of Southern Queensland

Faculty of Sciences

Optimisation of Test Signals for Channel Identification

A thesis submitted by

Stephen Clive Braithwaite

in fulfilment of the requirements of

Doctor of Philosophy

Submitted: June, 2011

Abstract

Channel estimation is required in virtually all communication systems: wireless, optical, and electrical systems. Modern mobile wireless devices undertake channel estimation repeatedly and audio communication systems of all varieties repeatedly estimate the echo-path channel so that echos can be cancelled. The channel estimation must be carried out repeatedly because the channel will vary while communication is taking place. This dissertation examines channel estimation in both acoustic and wireless cases.

Echo cancellation has become ubiquitous because it is superior to other echo management techniques whenever the echo channel can successfully be estimated. Echo cancellation relies on the correct estimation and modelling of the echo channel. Despite much research into advanced echo cancellation techniques, there remains acoustic applications where the channel is too dynamic for existing echo channel estimation techniques to succeed. An echo cancellation simulation environment has been developed to facilitate the simulation of dynamic echo environments and the monitoring of the results and the internal state of echo cancellation algorithms. This environment enables the monitoring of the overall result and the internal state of the echo cancellation algorithms. Object oriented programming techniques are used to achieve the flexibility needed to do this for different algorithms, different stimulus or test signals, different background noise and different channels.

While most acoustic echo channels are essentially linear, speakers can introduce non linearity to an echo channel, depending on their quality. A simplified non linear echo channel model of this situation is used, and a non linear algorithm for echo cancellation

developed that estimates the parameters of this model.

Second order statistics are used in techniques for channel estimation in wireless systems. Their use is necessary, given the dynamic nature of wireless channels. To date, however, there have been few attempts to fully utilise second order statistics as part of channel estimation for acoustic systems. The optimal linear estimator of the channel is identified that fully takes the second-order statistics of the channel variation into account.

Test signals are often used in the estimation of wireless channels. They are also used in the estimation of acoustic channels. Equations characterising the optimal (in the sense of least-squares estimation error for fixed power) test signal based on the second order statistics of the channel and of the noise are developed. The power advantage of an optimal test signal over white noise as a signal is unbounded as the channel statistics vary.

Channel autocovariances may be represented by covariance matrices. The model offered by a covariance matrix is often too complicated for use in practice, however. A model of intermediate complexity has been developed that remains sufficiently flexible to encapsulate desirable features of a channel autocovariance while being significantly simpler to use. The simplified model of a channel autocovariance has been used to randomly generate autocovariance matrices that are suitable for use in the testing the method for finding optimal test signals.

Associated Publications

The following publications were produced during the period of candidature:

Braithwaite, S. and Addie, R. "Optimal choice of test signals for channel estimation". *IEEE/ICME International Conference on Complex Medical Engineering*, Gold Coast Australia, Jul 2010.

Braithwaite, S. and Addie, R. "ECOOSE: an echo cancellation object oriented simulation environment", *International Conference on Simulation Tools and Techniques for Communications, Networks and Systems and Workshops*, Rome Italy, Mar 2009.

Braithwaite, S. and Addie, R. "Non-linear echo cancellation - a Bayesian Approach." *2nd International Conference on Signal Processing and Communication Systems*, Gold Coast Australia, Dec 2008

Braithwaite, S. and Addie, R. "Optimal Choice of Test Signals for Linear Channel Estimation using Second Order Statistics", *Optimization and Engineering*, Submitted Aug 2010 (Accepted subject to minor changes.)

The following submissions were produced during the period of candidature and await a decision on publication:

Braithwaite, S. and Addie, R. "Estimation of Linear Echo Channels using Second Order Statistics", *Signal Processing*, Submitted May 2011

Certification of Dissertation

I certify that the ideas, designs and experimental work, results, analyses and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged.
I further certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.

Signature of Candidate:

Stephen Clive Braithwaite
Date:

Signature of Principal Supervisor:

Associate Professor Ron Addie
Date:

Signature of Associate Supervisor:

Associate Professor Yan Li
Date:

Acknowledgments

This thesis was typeset using LaTeX 2_ε. Many diagrams were created using Open Office. UML class diagrams were created using BOUML.

I would like to acknowledge the support of my project supervisor, Associate Professor Ron Addie, who is a never ending source of ideas, enthusiasm and wisdom. I would also like to the acknowledge the support of Associate Professor Yan Li, who was there when she was needed.

Importantly, I would like to acknowledge the support of Melita, Josephine and Clover who have tolerated my high asperations, my meagre earnings, frugal spending and all of the time that I spent studying.

Stephen Clive Braithwaite

University of Southern Queensland
June 2011

Contents

List of Figures

List of Tables

Acronyms & Abbreviations

DFT	Discrete Fourier Transform
DSL	Digital Subscriber Line
DSP	Digital Signal Processors
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
GSM	Global System for Mobile Communications
ICI	Inter-Channel Interference
IID	Independent and Identically Distributed
ISI	Inter-Symbol Interference
ITU	International Telecommunications Union
LEM	Loudspeaker Enclosure Microphone
LOS	Line Of Site
LMS	Least Mean Squares algorithm
MSK	Minimum Shift Keying
NLMS	Normalised Least Mean Squares algorithm
OFDM	Orthogonal Frequency Division Multiplexing
PCA	Principle Component Analysis
PDF	Probability Density Function
PNLMS	Proportionate Normalised Least Mean Squares algorithm
PSK	Phase Shift Keying
QAM	Quadrature Amplitude Modulation
QMF	Quadrature Mirror Filters
RMS	Root Mean Square
RLS	Recursive Least Squares algorithm

FRLS	Fast Recursive Least Squares algorithm
SNR	Signal to Noise Ratio
SVD	Singular Value Decomposition
UML	Unified Modeling Language
WAV	Waveform Audio File format

Chapter 1

Introduction

This dissertation is primarily about acoustic channel estimation. Acoustic echo can cause reverberation and distortion of otherwise pristine sound. It may create an irritating delayed image of the original sound or it may cause squealing feedback. Techniques for the management of acoustic echo include echo suppression, comb filters, frequency shifting, and echo cancellation. Echo cancellation is the most difficult technically of these techniques, because it relies on channel estimation. Despite this, it has been very successful because when it works properly, it does not introduce distortion and does not inconvenience the users.

Channel estimation is also very important in wireless communications. The medium of wireless echo is that of electromagnetic energy transmitted through air. Like acoustic echo, the channel is complicated because the electromagnetic energy arrives at a receiver by many different paths. Like acoustic echo channels, wireless channels can be very dynamic. This is especially true in the case of mobile radio devices. One mitigating factor in wireless channel estimation is that fact that the spectrum of interest normally does not extend across the entire radio spectrum, but is normally limited to a relatively narrow frequency band.

Modern wireless applications such as mobile phones, digital TV and ASDL employ Orthogonal Frequency Division Multiplexing (OFDM) which has embedded test signals which have, as one of their functions, wireless channel estimation. Some of the

most successful wireless channel estimation techniques in OFDM utilise the secondary statistics of the channel. This dissertation utilises ideas from wireless channel estimation techniques and applies them to acoustic channel estimation.

1.1 Notation

This dissertation explores many aspects of channel estimation, and as such has a wide scope. It has not been possible to exclusively reserve modern and Greek alphabetic characters for every value which must be represented. Instead, this dissertation reserves symbols for only the key values and operations used throughout the dissertation. Since the key variables require the core concepts in order to have any real meaning, we will introduce the core concepts of echo cancellation, the key variables, and the notations for the dissertation together in this section.

Echo cancellation works by estimating the echo channel. The echo canceller uses the echo channel estimate in combination with the input to predict what the echo will be. In telephonic and acoustic echo cancellation, the predicted echo is removed from the output, by subtracting it as shown in Figure 1.1.

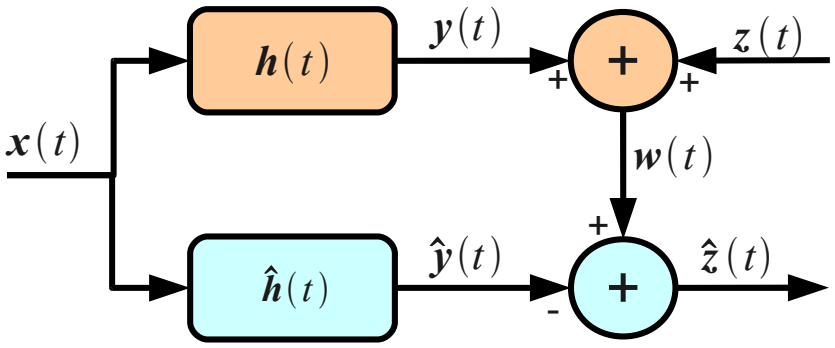

Figure 1.1: Block diagram for basic echo cancellation.

In Figure 1.1, $x(t)$ represents the original signal deliberately introduced into the echo channel. In telephonic echo cancellation, it is often referred to as the *far end signal*. In acoustic echo cancellation, it is the speaker signal. In wireless channel estimation it is referred to as the transmitted signal. In telephonic and acoustic echo cancellation, $x(t)$

is a real signal. In wireless channel estimation it is common to use a complex frequency shifted version of $x(t)$.

Most echo cancellation is linear, and in the case of telephonic and acoustic echo cancellation, the channel is most often represented in the form of an *impulse response*. Also in Figure 1.1, $h(t)$ represents the channel and $\hat{h}(t)$ is our estimation of it. Although impulse responses are infinite in theory, the amplitude eventually becomes small, and they can be truncated to a finite length in our model without sacrificing too much accuracy. The length of the impulse response will usually be denoted by n.

The impulse response is also commonly represented in the Fourier domain as $\tilde{h}(f)$ instead of $h(t)$ and in this case is referred to as the *transfer function*. The impulse response is generally real in telephonic and acoustic echo cancellation, but generally complex in wireless channel estimation. We use the term *convergence*, in echo cancellation, to describe the situation where the echo cancellation is successful, and $\hat{h}(t)$ approaches $h(t)$.

Also in Figure 1.1, $y(t)$ is the signal that emerges from the channel directly as a result of the signal $x(t)$ entering the channel, and $\hat{y}(t)$ is the estimate of $y(t)$. We will use use *hat* symbol above a value symbol, e.g. $\hat{h}(t)$, consistently throughout this dissertation to mean *the estimate of*. Similarly, we will use the *bar* symbol, e.g. $\bar{x}$, to mean *the mean of*. Sometimes we will use $\hat{v}_{k|j}$ to mean the estimate of v_k (the estimate of the k_{th} temporal instance of the scalar or vector v) based on information up to and including the j_{th} temporal instance.

Also depicted in the illustration is $z(t)$. In echo cancellation this is generally referred to as the noise. In wireless channel estimation it really is noise, i.e. something that would be eliminated, were it easily possible, as the goal is to find $x(t)$. In telephonic echo cancellation and in acoustic echo cancellation, however, $z(t)$ is often referred to as the near end signal and it is the desired output from the echo cancellation subsystem.

Also appearing in the diagram is $w(t)$, which is the combination of the near end signal combined with the echoes of the far end signal, that is:

$$w(t) = y(t) + z(t).$$

Unlike $y(t)$ and $z(t)$, $w(t)$ can be measured. In the case of telephonic and acoustic echo cancellation, $\hat{y}(t)$ is subtracted from $w(t)$ giving the result $\hat{z}(t)$ which hopefully is a faithful reproduction of $z(t)$.

Echo cancellation normally works digitally by representing quantities as discrete digital samples held in arrays, and hence this dissertation generally refers to the discrete vectors, e.g. $\mathbf{x}$, $\mathbf{y}$, $\mathbf{h}$, $\mathbf{w}$ and $\hat{\mathbf{h}}$ instead of $x(t)$, $y(t)$, $h(t)$, $w(t)$, and $\hat{h}(t)$. Note that we use bold lowercase letters to indicate vectors throughout the dissertation. Matrix versions of these values such as $\mathbf{X}$ are used occasionally throughout the dissertation. Note that that matrices are denoted by bold capital letters. Vectors and matrices are indexed from zero in this dissertation. The matrix $\mathbf{I}$ is the identity matrix. The square root of -1 is represented by i.

In the linear case, the effect of the channel upon the signal $\mathbf{x}$ can be modelled by

$$\mathbf{y} = \mathbf{x} \otimes \mathbf{h}, \tag{1.1}$$

where the $\otimes$ symbol indicates convolution. The $\otimes$ symbol will be used to indicate convolution throughout the dissertation. It is often convenient to express the samples of $x(t)$ in reverse time order so that $\mathbf{x}_0$ is the most recent sample of $x(t)$ and $\mathbf{x}_1$ is the previous sample of $x(t)$, and so on. If we use this convention, we may express the most recent discrete value of $y(t)$, y as an inner product

$$y = \mathbf{x}^T \mathbf{h}. \tag{1.2}$$

The Discrete Fourier Transform (DFT) Matrix $\mathbf{F}$ is a matrix representation of the DFT, such that if $\tilde{\mathbf{x}}$ is the DFT of x, then

$$\tilde{\mathbf{x}} = \mathbf{F}\mathbf{x},$$

where the tilde character is interpreted to mean "in the Fourier domain". In order to simplify our algebra we use the *unitary* version of the DFT and the DFT matrix so that

$$\mathbf{F}^* \, \mathbf{F} = \mathbf{I},$$

which is a very useful property. Specifically,

$$F_{j,k} = \frac{1}{\sqrt{n}} e^{(\frac{-2\pi i \, jk}{n})}. \tag{1.3}$$

The equation (1.1) may be represented in the Fourier domain, and in this case convolution in the time domain is equivalent to element by element multiplication in the Fourier domain, i.e.

$$\tilde{\mathbf{y}} = \tilde{\mathbf{x}} \diamond \tilde{\mathbf{h}}, \tag{1.4}$$

Throughout this dissertation, the $\diamond$ character indicates element by element multiplication between vectors or matrices having the same dimensions.

The channel impulse response of a communication systems under consideration generally varies with time, and hence we regard it as *random*, having a mean and an autocovariance. We use the overline or the expectation operator $E[\cdot]$ to denote the time or the ensemble average (which are equal, because we assume the whole system is ergodic):

$$\overline{h} = E[h]. \tag{1.5}$$

We denote the autocovariance matrix by $\mathbf{C}_h$, and the subscript h here indicates that this particular matrix is the covariance of the impulse response h.

$$\mathbf{C}_h = E[\mathbf{h}\ \mathbf{h}^T] - \bar{\mathbf{h}}\ \bar{\mathbf{h}}^T. \tag{1.6}$$

If we assume that the mean values of the impulse response are zero, we may write:

$$\mathbf{C}_h = E[\mathbf{h}\ \mathbf{h}^T]. \tag{1.7}$$

The subscript h indicates that this is the covariance of the impulse response. The superscript $(\cdot)^T$ is used to indicate the transpose. In the equation

$$\mathbf{C}_{\tilde{h}} = E[\tilde{\mathbf{h}}\ \tilde{\mathbf{h}}^*], \tag{1.8}$$

$\mathbf{C}_{\tilde{h}}$ is the covariance matrix of the transfer function. $\tilde{\mathbf{h}}$ is the transfer function, and since it is complex, the Hermitian transpose is used, indicated by the superscript $(\cdot)^*$. We use the superscript $(\cdot)^+$ to indicate the Moore Penrose inverse of a matrix (Moore 1920, Penrose 1955).

We use σ to denote the standard deviation. As $w(t)$, $x(t)$, $y(t)$ and $z(t)$ are zero mean signals, in general, the power in a signal is the square of the standard deviation, e.g. the power in the noise is normally denoted by σ_x^2.

The % symbol is used throughout this dissertation to indicate the remainder of integer division (Boute 1992) where, for example, $(-2)\%3 = 1$. This symbol is used instead of the more standard mod() function because it is used frequently in matrix subscripts, and it was found that under these conditions the use of the mod() function here makes the equations in question unnecessarily difficult to read.

The following functions are each used throughout this dissertation:

diag$(\cdot)$ Returns a vector that has the diagonal elements of its matrix argument as its elements.

Diag$(\cdot)$ Returns a diagonal matrix that has the elements of its vector arguments as its diagonal elements. Clearly:

$$\text{diag}(\text{Diag}(\mathbf{v})) = \mathbf{v}.$$

Circ$(\cdot)$ Returns a circulant matrix that has the elements of its vector argument as its left hand column.

tr$(\cdot)$ Returns the trace of a matrix, i.e. the sum of the elements along the main diagonal.

traces$(\cdot)$ Returns a vector that has the sum of each diagonal of its matrix argument as its elements. Specifically:

$$\text{traces}(\mathbf{X})_i = \sum_{j=0}^{n-1} \mathbf{X}_{(j-i)\%n,j} = \sum_{j=0}^{n-1} \mathbf{X}_{j,(j+i)\%n}. \tag{1.9}$$

Finally, this dissertation uses differential calculus on vector functions of vectors and on scalar functions of matrices. I have chosen from several plausible notations. One work that this dissertation refers to (Petersen & Pederson 2008) uses the partial differentiation notation as commonly found in multivariate calculus. For example, rule 57 from (Petersen & Pederson 2008) states:

$$\frac{\partial \text{tr}(\mathbf{A}\mathbf{X}^{-1}\mathbf{B})}{\partial \mathbf{X}} = -(\mathbf{X}^{-1}\mathbf{B}\mathbf{A}\mathbf{X}^{-1})^T, \tag{1.10}$$

where $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{X}$ are all matrices. However, various publications in the field of echo cancellation use $\nabla_{\mathbf{x}}$ to mean differentiation with respect to $\mathbf{x}$ as it does in vector calculus. (Cabell 1998, Widrow, McCool, Larimore & Johnson 1976, Widrow & Stearns

1985) use this notation, as does this dissertation. The following example, in which the expression ($\mathbf{Ax}$) is being differentiated with respect to the vector $\mathbf{x}$, demonstrates its use

$$\nabla_{\mathbf{x}}\mathbf{Ax} = \mathbf{A}. \tag{1.11}$$

In addition, the example above demonstrates the convention used in this dissertation for the differentiation of vector functions of vectors. It implies that if we have a vector $\mathbf{u}$ of dimension n that is some function of a vector $\mathbf{v}$ of dimension m, then $\nabla_v(\mathbf{u})$ is a matrix that has n rows and m columns. If u is scalar, then $n = 1$ and the derivative is a row vector. If v is scalar, then $m = 1$ and the derivative is a column vector. This convention is used because when adhered to, the chain rule may be expressed simply in terms of a matrix multiplication. That is, if we have a vector $\mathbf{p}$ that is a function of $\mathbf{u}$, then

$$\nabla_v\ \mathbf{p} = (\nabla_u \mathbf{p})\ \ (\nabla_v \mathbf{u}) \tag{1.12}$$

1.2 Echo Cancellation

Echo cancellation is required in a wide range of situations and the nature of the problem varies significantly. Two most important types of echo cancellation are:

telephonic echo cancellation is required to cancel an echo which occurs due to the presence of a *hybrid* in the circuit between a telephone and the network in most residential access networks. Telephonic echo cancellation has been successful enough that we may regard it as a solved problem.

acoustic echo cancellation is required to cancel an echo which occurs when a microphone and speaker are used in the same space for the purpose of communication with a remote party. The medium of acoustic echo is that of sound transmitted through air. Echo cancellation is employed in mobile phones and in videoconferencing, greatly enhancing the sound quality without most users even being aware of its use.

Acoustic echo is caused by the reflection of sound from walls, ceilings, floors, furniture and people. Acoustic echo channels are usually complicated because they

are produced by the echoes of echoes, and the sound that reaches its destination will have come via many different paths. Acoustic channels may be very dynamic, and the echo cancellation technology needs to adapt rapidly. Acoustic echo cancellation is still an ongoing topic of research because there are still situations (described in Section 1.2.1) where the current technology is not adequate. The range of frequencies of interest in echo cancellation is the entire audio spectrum ranging from approximately 100 Hz to approximately 20 KHz.

However, the history of echo cancellation begins with telephonic echo cancellation, and most of the techniques for acoustic echo cancellation were originally developed for telephonic echo cancellation and refined for acoustic echo cancellation. Therefore, this dissertation cannot ignore telephonic echo cancellation.

There are also a wealth of channel estimation techniques used in wireless technologies. Part of this dissertation aims to adopt concepts used in the wireless world for application in acoustic situations. Therefore it is necessary for this dissertation to examine some examples of dynamic channel estimation in the wireless world.

1.2.1 Acoustic Echo Challenges

As mentioned above, there are some applications where echo cancellation is highly desirable, but the echo cancellation is inadequate. Here we shall explore some examples.

Speaker Phones

Acoustic echo cancellation works best when the echo channel is relatively static and when the echo has a short time duration. The echo cancellation associated with a mobile phone, for example, enjoys a short echo channel that is fairly static, and as such works well. In stark contrast to the mobile phone, the speaker phone has a longer echo duration and one that can change dramatically whenever the user changes position. It should therefore come as no surprise that hands free telephony is one application where the current echo cancellation technology is inadequate.

This is very clearly demonstrated by a string of patent applications enabling better echo suppression for speaker phones (Arbel 1989, Staudacher & Harbhajan 1997, Faraci & Skene 1999, Freed 2004). There would be no need to resort to echo suppression if the echo cancellation was adequate. Similarly, research into spectral power subtraction where sub bands are suppressed to improve human perception of sound quality (Faller 2003) would not be needed if the echo cancellation was adequate.

Music Concerts

The echo from a music venue back into the microphone of a singer reduces the quality of the music. Low level echoes manifest themselves in the form of unwanted reverberation, and in the worst case can introduce a loud squeal if the feedback into the microphone becomes too strong. The problem of echo cancellation in this environment is made difficult because the singer moves, and each different position that the singer takes causes a different echo channel.

In the case of the spoken word, frequency shifting using the sound equipment is commonly used to prevent microphone squeal. Frequency shifting works by multiplying the signal by a constant frequency. Frequency shifting effectively adds or subtracts the multiplied frequency to all frequencies of the sound signal. Since sound cannot feed back at the same frequency, microphone squeal is stopped.

While frequency shifting may be acceptable in the case of the spoken word, such frequency shifts change the frequency ratios between notes causing any melody to be off key. Frequency shifting has a disastrous effect on music, and can not be used at a concert. In these circumstances microphone squeal may be muted by notching out or suppressing the frequency that the squeal would have occurred in. This is also not ideal since it distorts the signal received by the concert audience.

An echo cancellation technology that can adapt rapidly enough may have the potential to eliminate the need for such techniques as frequency shifting and the suppression of problem frequency sub bands.

VOIP

Voice over Internet Protocol (VOIP) communication refers to the use of the Internet to transmit interactive voice communication. VOIP provides a telephony service at a fraction of the cost of existing telephone networks although the quality of communication is significantly worse. It is widely expected that VOIP will gradually replace the traditional telephone network. In addition, facilities for VOIP communication in common use frequently make use of computers with attached or built-in audio and possibly video input/output devices.

When a headset is in use with VOIP echo cancellation is not necessary. However, the use of a headset for such telephony is regarded as an inconvenience and as such limits the spread of VOIP. Echo cancellation would remove the need for headphones.

Difficulties arise because the person speaking moves, thus altering the echo channel. Personal experience with VOIP gives testament to the fact that the echo cancellation used struggles to cope with the echoes when headsets are not used.

Recently, Skype™ has resorted to echo suppression as its one of its echo control mechanisms, which clearly demonstrates the inadequacy of the current echo cancellation technology.

Hearing Aids

Without the use of echo cancellation there is a limit to how much amplification can be used in a hearing aid. Although hearing aids that fit snugly inside the ear can reduce the acoustic feedback, a reduction in sound quality and howling will occur well before a satisfactory volume is reached for some patients. Echo cancellation technology can remove the feedback completely depending on the accuracy of the estimate of the channel. The channel is difficult to estimate, however, and the subject of echo cancellation in hearing aids is an ongoing topic of research (van Waterschoot & Moonen 2009, Spriet, Moonen & Wouters 2009).

One common method reducing feedback in hearing aids is to use a test signal when-

ever misalignment in the channel model is detected (Kates 1991, Goodings, Senebsieb, Gideon, Wilson & Hanson 1993, Engebretson & French-St.George 1993, Spriet, Proudler, Moonen & Wouters 2005). The efficiency of these test signals can be improved by optimising the test signals using the techniques developed in Chapter 6, *Optimisation of Test Signals.*

Remote Auscultation

Queensland has remote areas which are difficult to service in an economical way. Queensland Health uses telehealth as an important means of addressing the logistical problems encountered in providing quality services to remote areas (*The Uneven Diffusion of Telemedicine Services In Australia* 1998, Nooriafshar & Maraseni 2007). Queensland has over 200 Telehealth clinics. Queensland Health boasts that it is recognised internationally as having the largest and most utilised videoconferencing network of any single health organisation in the world (*Excellence in Queensland Health* 2008).

Many of the remote health clinics operate with a nurse at the remote end, that performs tests, collects documents and other details and faxes the results to the relevant specialist clinic. The videoconferencing systems are then used to facilitate an interview between the specialists and the patients (Braithwaite 2006).

This arrangement is unsatisfactory, however, for auscultation. Doctors have expressed the need to listen to the patients heart and the patient's lungs directly. It is not satisfactory for the nurse to do the listening for the specialist (Braithwaite 2006).

The obvious solution is to use the stethoscope directly using the videoconferencing system. Sadly the direct use of the stethoscope remotely has not so far been a success. Using the stethoscope directly in over the videoconferencing results in poor quality auscultation. We believe that the problem exists because the echo cancellation normally associated with videoconferencing is unsuitable for remote auscultation.

Acoustic echo cancellation is already more challenging than hybrid echo normally associated with telephony. One thing that makes it more challenging is the echo path impulse response is much longer than that associated with a hybrid echo. This longer

impulse response requires the use of more computing power.

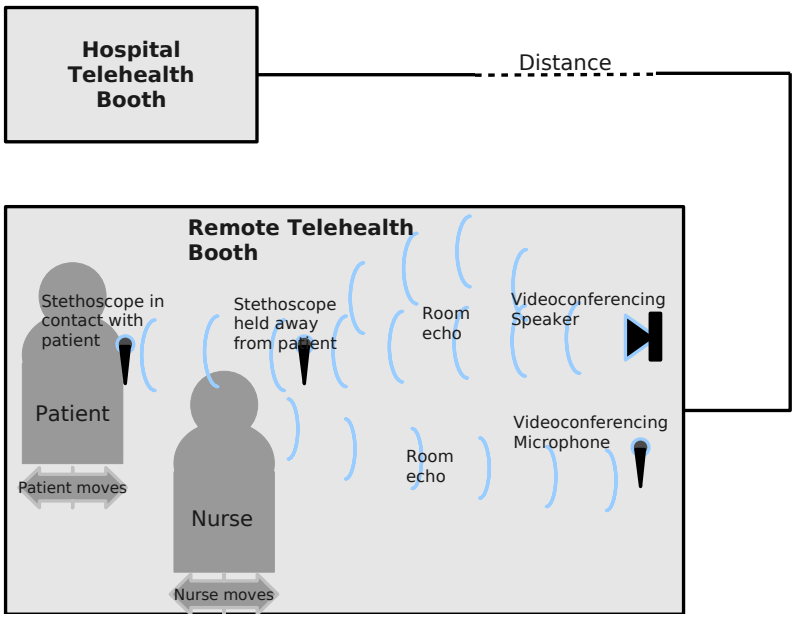

Figure 1.2: Illustration demonstrating why the impulse response of a remote auscultation is subject to rapid change.

Echo cancellation associated with a remote digital stethoscope faces further challenges. As illustrated in Figure 1.2 the stethoscope may be held away from the patient, or it may be held against the patient. The echo channel will have undergone dramatic change when the stethoscope was placed onto the patient or moved to a different location on the patient. Also, the patient changes the geometry of his body, and therefore also the echo channel whenever he or she breaths in or out. Hence, the echo cancellation needs to converge quickly. It is quite a challenge for the echo cancellation to converge quickly, however, because the auscultation sound will be present at the same time as the echo. This is a condition known as *double talk*, which is known to cause slow convergence of an echo cancellation algorithm.

One potential solution would be for the nurse to initiate a test signal whenever the stethoscope is in place and movement has ceased. Such a technique can be made substantially more efficient if the test signal is optimised. We develop a technique for the optimisation of test signals in Chapter 6, *Optimisation of Test Signals*.

1.2.2 Echo Cancellation Technology

Echo cancellation algorithms examine the input and the output of an echo channel and perform the calculations which update the model of an echo channel. Then the model is used to predict what the output from the channel will be based on the input to the channel.

Most algorithms constantly use the input and output samples to update the estimate of the echo channel at each time instant, and the cancellation performed uses the latest updated model, but block based echo cancellation work differently. Samples of input data are grouped into blocks. When enough data has arrived to fill the blocks, the blocks are processed and the working model is then updated. Most echo cancellation systems operate purely in the time domain, but echo cancellation performed in an orthogonal basis and frequency domain echo cancellation are also possibilities.

Although the echo cancellation algorithm is the core technology of echo cancellation, there are also key enabling technologies that tend to be used in practice. For example, subband echo cancellation (described in Section 2.8) provides a way of breaking down the computationally complex task of echo cancellation so that the echo cancellation algorithm has much less work to do. Double talk detection (described in Section 2.9) is another enabling technology. LMS and AP based algorithms are quite sensitive to noise, and benefit strongly from the use of a double talk detector, which detects noise (double talk), and turns off or slows down the adaptation of the echo canceller untill the noise (double talk condition) has passed.

We shall outline the main echo cancellation algorithms and enabling technologies here, and explore each more fully in Chapter 2, *Echo Cancellation*.

1.2.3 The LMS Algorithm

Echo cancellation began with the Least Mean Square (LMS) technique, which was patented in the early 1960s. It is so named because in the absence of noise, it converges to the point of least mean error in $w(t)$. Because of its inherent simplicity, its low

computational complexity O_n and its low memory requirements, it remains the most widely used echo cancellation technique (Vaseghi 1996). LMS algorithms are explored in more depth in Section 2.3, *LMS echo cancellation.*

1.2.4 The RLS Algorithm

The RLS algorithm uses data from the past to achieve rapid convergence. It is assumed that the system does not change dramatically in the short term, and data from recent samples are worth more than samples from the more distant past. Thus, the RLS algorithm uses an explicit weighting factor so that recent samples have more influence on the results than past ones. RLS algorithms are explored in more depth in Section 2.4, *The Recursive Least Squares Algorithm.*

In Chapter 6, *Optimisation of Test Signals*, we introduce a different approach to using the past which uses second order statistics of the channel from the past record to improve current estimates.

1.2.5 The AP Algorithm

The Affine Projection (AP) algorithm can be seen as a hybrid between the LMS algorithm and the RLS algorithm, in that it uses a limited amount of information from the past, depending on the *order* of the algorithm. An order one AP algorithm is identical to the NLMS algorithm. An order two AP algorithm is less efficient than the NLMS algorithm, but achieves a much faster convergence than the NLMS algorithm. An order 3 AP algorithm is less efficient than an order 2 AP algorithm, but has a faster convergence, and so on. An order ∞ AP algorithm essentially is the RLS algorithm. AP algorithms are explored in more depth in Section 2.5, *The Affine Projection Algorithm.*

1.2.6 Frequency Domain Echo Cancellation

Frequency domain echo cancellation is described in Section 2.6, *Fourier Domain Echo Cancellation.* Frequency domain echo cancellation is a somewhat natural choice for

echo cancellation because a sinusoidal wave at a given frequency entering an echo channel results in a sinusoidal wave at that same frequency, differing only in amplitude and phase. Thus we may perform echo cancellation in each frequency band individually. In the frequency domain amplitude and phase are encompassed within a single complex number, and we may determine the channel for a given frequency by means of a single division, making frequency domain echo cancellation computationally very efficient. This clearly does not utilise statistical correlations between different frequencies, however, as we do in Chapter 6, *Optimisation of Test Signals*.

1.2.7 Transform Techniques

Although the echo cancellation problem presents itself in the time domain, there may be significant advantages in converting the problem so that it is expressed in an orthogonal basis and solving in that domain. There are several approaches. These are described in Section 2.7, *Using Orthogonal Transforms*.

1.2.8 Frequency Shifting Techniques

The normal acoustic range occupies a bandwidth from as little as 10 Hz to as much as 20 kHz. Its frequency bandwidth is approximately that of its highest frequency, i.e. 20khz. Occupied wireless bandwidths, on the other hand, are very narrow when compared to the absolute transmitted frequencies.

Wireless channel estimation would be very difficult if we attempted to sample at radio frequency rates. Not only would we have a very long effective impulse response to work with at each sample, but we would have a very short time between samples, and have to process very fast in order to ready for the next sample. It is still possible to perform channel estimation on a wireless signal by frequency shifting it downwards. Frequency shifting is examined in more detail in Subsection 2.2.3, *Feedback Suppression*.

Let us imagine, for example, that we were to perform channel estimation on an FM broadcast band operating at 100.1 MHz. The lowest frequency within its effective bandwidth would be the central frequency minus 75 kHz, i.e. 100.025 MHz, and our

transmit and receive signals may be frequency shifted down by that amount. We could then perform channel estimation in the frequency range 0 to 150 kHz, remove the echo, and frequency shift our echo free result back up to the transmitted frequency range.

We can do this with acoustic echo cancellation too, by splitting the frequency range into subbands. High quality matched filters are used to divide the full bandwidth into approximately four to eight subbands. The subbands are then shifted down into the frequency range of the lowest subband. Echo cancellation is performed with short impulse responses and low sample rates, dramatically reducing the overall computational complexity of the echo cancellation task. The resulting output with the echo removed is then frequency shifted back to the original subbands and recombined into a full band. This is known as subband echo cancellation and it is covered in Section 2.8, *Subband Echo Cancellation*.

1.2.9 Double talk Detection

The term *double talk* was originally used to describe the condition where both parties are talking at once during a telephone conversation (Emling & Mitchel 1963), but from the point of view of echo cancellation, would also apply to any case where the signal $z(t)$ is non zero. Channel estimation accuracy is likely to be impaired during conditions of double talk, as explained in Subsection 2.3.4 *Analysis of the LMS Algorithm*, and because of this, a double talk detector is normally used to detect the condition so that the channel estimation may be halted or slowed. Double Talk Detectors are discussed in Section 2.9, *Double-talk Detection*.

1.3 Ideas from the Wireless World

The work described in Chapter 6, *Optimisation of Test Signals*, is primarily about the optimisation of periodic test signals using the autocovariance of the noise and the autocovariance of the channel history. Although the application of this work is universal, it is targeted at acoustic channels. There appears to be no existing publications on the optimisation of test signals based on the secondary statistics of the channel or the noise.

There are publications on the optimisation of the crest factor of periodic test signals for wireless channels and publications on the use of secondary statistics for channel estimation, also for wireless channels. We have reviewed wireless literature relevant to our work on test signals in Chapter 5, *Lessons From the Wireless World*.

1.3.1 Orthogonal Frequency Division Multiplexing

Orthogonal Frequency Division Multiplexing (OFDM) is a very successful technology, and a field of ongoing research. In OFDM, the secondary statistics of the channel are used to assist the channel estimation and we explore the techniques that are use to achieve this in Section 5.1, *Statistical Estimation of OFDM Wireless Channels*. Rather than assume that the reader already understands OFDM, we have provided sufficient background on fading wireless channels and OFDM so that the reader may understand why these techniques are used and how they work. Having provided some background, we proceed to explore Least Squares (LS) channel estimation and Linear Minimum Mean Squared Error (LMMSE) Estimation used in OFDM communications technology.

1.3.2 Peak Factor Optimisation

The test signal optimisation described in Chapter 6 finds a test signal with the most efficient use of test signal power, given knowledge of the secondary statistics, for a linear channel. So long as the channel is perfectly linear then the optimal test signal may be defined only by the signal strengths of each frequency in the Fourier domain. The optimality of the test signal does not depend on the phases of the test signal.

The channel without the transducers is generally linear with no effective limits, however, the transducers and the electrical equipment associated with the transducer generally have limits on the peak instantaneous power transmitted. It may be that a transducer is only linear within a certain prescribed range, for example. It therefore becomes desirable to choose the phases of a test signal in the Fourier domain with given frequency strengths so that it has the smallest peak instantaneous power in the time domain. The process of choosing the optimal phases in each frequency to give the smallest peak

instantaneous power of the test signal is called crest factor optimisation.

There is currently no formula that gives the optimal phases. (Schroeder 1970) developed a formula that gives significantly better results than one would expect from using randomly chosen phases. Crest factor optimisation techniques generally use Schroeder's formula as the starting point for their optimisation. We explore crest factor optimisation in more detail in Section 5.2, *Optimisation of Test Signal Crest Factors*.

1.4 Contributions to Channel Estimation

This section will outline contributions of this dissertation to channel estimation.

1.4.1 ECOOSE

Chapter 3, *Echo Cancellation Simulation*, describes the Echo Cancellation Object Oriented Simulation Environment (ECOOSE). ECOOSE is an object oriented framework for the development and testing of echo cancellation algorithms. The most important attribute of ECOOSE is flexibility:

- ECOOSE can work with static or dynamic channels.
- These channels may be linear or non-linear.
- ECOOSE permits testing using any recorded channel input, $x(t)$, and any recorded noise, $z(t)$.
- It can be used to drive adaptive online algorithms as well as block based algorithms.
- The object oriented structure of ECOOSE permits the specification of any ad-hoc parameters required by a new algorithm or added to an old algorithm in a painless way, with defaults, so that one does not have to pay attention to each parameter while one is testing.

ECOOSE is the product of three development cycles. Procedural method using a procedural language were employed in the first attempt at creating this flexible environment. It turned out to be difficult to implement the desired flexibility in a procedural way. The desired flexibility needed for the development and testing of a wide variety of algorithms was later achieved by re-implementing the test environment using object oriented technology.

1.4.2 Applying the Method of Innovations

A Hammerstein channel (Belforte & Gay 1999) consists of a linear channel preceded by a memoryless non-linear channel, as depicted in Figure 1.3. Such a channel might result from a loudspeaker that clips during periods when it is pushed beyond its linear range.

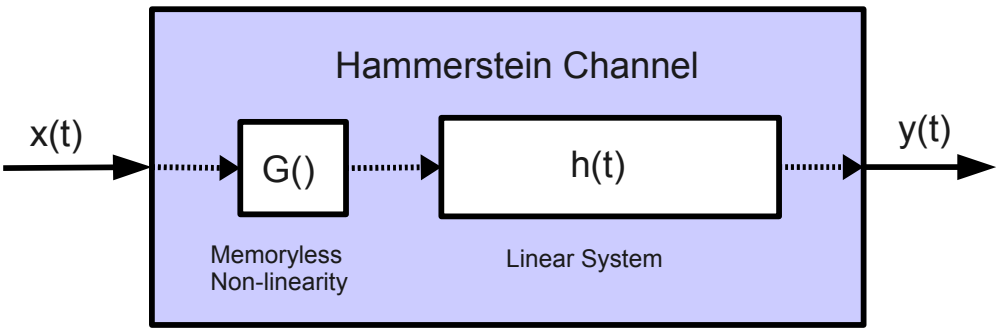

Figure 1.3: Diagram depicting a Hammerstein non linear system.

In Chapter 4, *Framework for Echo Cancellation with Non Linearities*, we describe how we modelled the memoryless non-linear part of the channel using Legendre polynomials, which have the property of being orthogonal on the interval $[-1 : 1]$ with a constant weighting function. We describe the method of innovations (Kailath 1981) and then use it to derive an algorithm to perform echo cancellation using this model of a Hammerstein channel.

The derived algorithm was implemented in ECOOSE and we perform simulations that show that the resulting algorithm converges for Hammerstein channels, whereas echo cancellation algorithms that assume a linear channel fail to converge.

1.4.3 Finding the Optimal Test Signal

Chapter 6, *Optimisation of Test Signals*, describes the development of methods for finding the optimal test signal for a fixed signal power when knowledge of the secondary statistics of the channel and of the noise are available.

Note that our test signal optimisation is different from the crest factor optimisation described in Section 5.2. Our optimisation finds the optimal frequency strengths for a given overall test signal power, and is complementary to crest factor optimisation, which finds the optimal phases given the frequency strengths of the test signal in order to minimise the peak instantaneous power.

Our test signal optimisation assumes a linear channel with no limits on the instantaneous peak power, and under that assumption the optimal signal may be described in terms of the signal strengths of each frequency in the Fourier domain, i.e. the phases of the test signal have no bearing on its effectiveness. Crest factor optimisation optimises the phases and is orthogonal and complementary to this type of optimisation.

We find the optimal estimator for a fixed test signal when we know the secondary characteristics of the channel and the noise. We define a measurable quantity which we call *coloredness* and note the performance of the optimal estimator for different noise coloredness and for different coloredness of the channel history. Covariances are called *white* if all frequencies have the same power, and called *colored* if they do not.

Two different methods of finding the optimal test signal are devised. One method of test signal optimisation is devised in the time domain. We derive a formula for the optimal test signal for a fixed estimator, subject to a constraint on the available power. Our optimisation method employs both the formula for the optimal estimator and the formula for the optimal test signal.

A key observation from the results using this method is that the optimal test signal is not unique in the time domain, but the frequency strengths associated with an optimal signal are unique. We prove that phases in the frequency domain have no bearing on the performance of a test signal. Based on this result we derive a second equation

characterising the optimal test signal with the information about the phases removed from the signal being optimised, in the frequency domain. Since the information about the phases has been removed, there are only half the number of variables to optimise.

We test our optimisation methods using a randomly generated ensemble of channel scenarios, comparing the performance of the optimal test signal with white ones. We show that if the secondary statistics of the noise or of the channel are known, then a white test signal is, in general, not the optimal test signal. We show that the advantage in accuracy obtained by using an optimal test signal may be arbitrarily large, depending on the particular situation.

1.4.4 Statistical Model of Channel Variation

The covariance matrices that we used for testing our optimisation of test signals were not generated experimentally, but were randomly created from a statistical model of possible channels. If we take into account the fact that covariance matrices are symmetric, there are effectively $\frac{n(n-1)}{2}$ elements to specify in a covariance matrix, which is too many to specify without a model of how to do so.

In order to validate the test signal optimisation methods developed in Chapter 6, it was neccesary to show that these methods work well for a randomly chosen combination of channel and noise covariances drawn from a reasonable distribution of possible channel and of noise covariances, rather than only for a specific channel.

Channel covariance matrices have properties that need to be observed in both the time domain and in the frequency domain. These properties are positive definiteness and symmetry. We need to be able to specify the diagonal in the time domain to be consistent with a dispersive impulse response as shown in Figure 2.5. We also need to be able to specify the diagonal in the frequency domain, which corresponds to the power spectral densities.

In Chapter 7, *Simplified Models for Impulse Response and Noise Covariances*, we develop models of the covariance matrices of the channel and of the noise, and devise a method of creating them with the required features in both the time domain and the

frequency domain. Our method requires the specification of the main diagonal in the time domain as well as a specification of the diagonal in the frequency domain.

1.4.5 Mathematical Identities

Appendix A, *Mathematical Results*, contains mathematical results used in the main body of the dissertation. The derivation of equations associated with the optimisation of test signals (Chapter 6) required some unfamiliar matrix identities. Some of these were found in the literature, and these are stated and referenced in Section A.2, *Identities Adopted for Optimisation of Test Signals*. Other matrix identities were developed as part of this work, and are believed to be original. They are stated and proved in Section A.1, *Identities Developed for Optimisation of Test Signals*.

1.4.6 Computer Program Source Code

There are over 6000 lines (200 pages) of Matlab 2008 source code that were developed as part of this work. is a guide to the source code. The source code has been made available, and is described in Appendix B, *Guide to Software Source Code*, but has not been included in the body or in the appendices of this dissertation.

A guide to the Matlab code associated with ECOOSE (Chapter 3) is given in Section B.1, *ECOOSE Source Code*. It includes a guide to the class hierarchy of ECOOSE and an explanation of the directory layout of the code. In addition, ready made experiments or demonstrations have been made available and there are instruction on how to execute these.

A guide to the Matlab code associated the optimisation of test signals is given in Section B.2, *Optimal Test Signal Source Code*. There is a guide to the directory layout, as well as sections that explain the core optimisation routines, the routines that manufacture test data, and routines that orchestrate experiments and produce plots. As with ECOOSE, demonstrations have been made available and there are instruction on how to execute these.

Chapter 2

Echo Cancellation

In this chapter we will provide an overview of echo cancellation, its background and its technology. We will examine the historical background of this technology, the need for echo cancellation as well as the alternatives to echo cancellation. We will also explain the basic concepts, examine the key echo cancellation technologies and the mathematics that underpins them, and explore the directions and possible impact of current research.

2.1 Telephonic and Acoustic Echo Channels

2.1.1 Echo Associated with Telephony

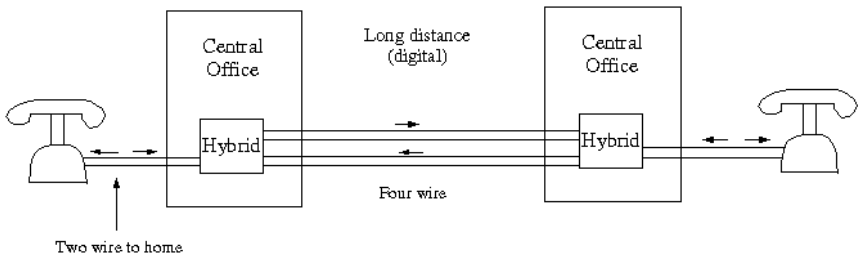

Figure 2.1: Illustration of telephone loop using 2 wire and 4 wire paths. Adapted from (Jones 2010).

Investigation into echo cancellation was initially motivated by the desire to rid long

distance telephone calls from annoying echoes. Prior to the 1930s, telephony was essentially local, and the communication delay was less than approximately 10 milliseconds. If the time duration for an echo is less than 40 ms, then the human ear will not notice the echo (Cutler 1965). The echo signal cannot readily be discerned because it is almost coincident with the listener's side-tone level(Berkley, Mitchell & Pierce 1972, Painter & Spanias 1997). The echo only became a problem when international telephony introduced one way delays of up to 80 milliseconds, for example a call between London and Hawaii (Emling & Mitchel 1963). When end to end delay is as long as this, the echo becomes annoying (Blackwell 1932).

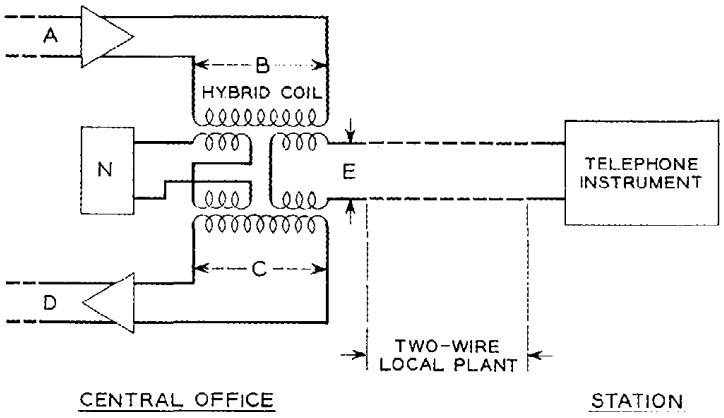

Figure 2.2: Schematic diagram of a hybrid used in telephony between two wire circuits and 4 wire circuits. Adapted from (Emling & Mitchel 1963).

There are two types of echoes associated with early telephony. One is the echo created by sound from the speaker of the handset entering the microphone. Telephones in those days had good acoustic separation between the microphone and the speaker so this sort of echo was minimal. The other type of echo is created by impedance mismatches in the electrical circuits. One major impedance mismatch that could not easily be eliminated was the conversions between the core four wire circuits and end user two wire circuits. Transmission over distances of length requires amplification, and is only practical in systems which use four wires, i.e. two in each direction. In order to save copper, telephone companies used a two wire system between the most local telephone exchanges and individual houses or offices. So at some point near the residence, usually the local exchange, a conversion between 2 wire and 4-wire communication must take place (Kelly & Logan 1966, Sondhi 1966, Gritton & D.W.Lin 1984). A high level

diagram dipicting the use of hybrids is shown in Figure 2.1, and a schematic diagram of a hybrid is shown in Figure 2.2.

The device which implements this conversion is known as a *hybrid*. The hybrid is an imperfect device, which inevitably has an impedance mismatch which creates echoes. Efforts at creating an automatic balancing system for the hybrid were reduced after it was proved that this was theoretically impossible with any linear network (Lewellyn 1950). Because the dominant echo is created at a single point, telephonic echo is generally sparse, i.e. most of the values of the impulse response are very small or zero, as illustrated in Figure 2.3.

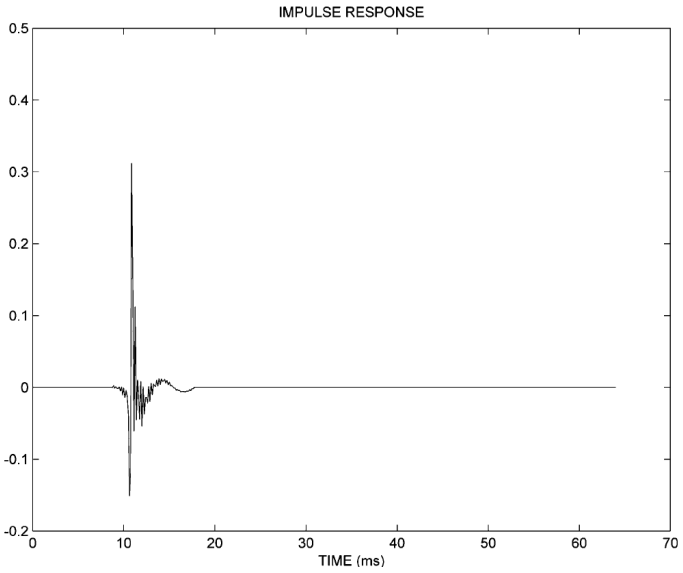

Figure 2.3: Illustration of a sparse echo channel associated with telephony. Adapted from (Jones 2010).

2.1.2 Acoustic Echo

If we have a space with loudspeakers to deliver the far end signal to our listeners and a microphone to pick up their speech as depicted in Figure 2.4, then some of the sound from the loudspeakers will be picked up by the microphone. The sound from the loudspeakers picked up by the microphone is termed acoustic echo.

Depending on the situation and on the delay, acoustic echo can seriously degrade the

sound delivered or simply cause an annoying echo. In the case of a music concert, where the delay is short, sound originating from the speakers that enters the microphone can cause reverberation and degradation of the overall sound quality, and in the extreme case cause microphone squeal. In the case of videoconferencing, where the delay is long, the echo can act as noise which prevents a clean sound and easy comprehension of the

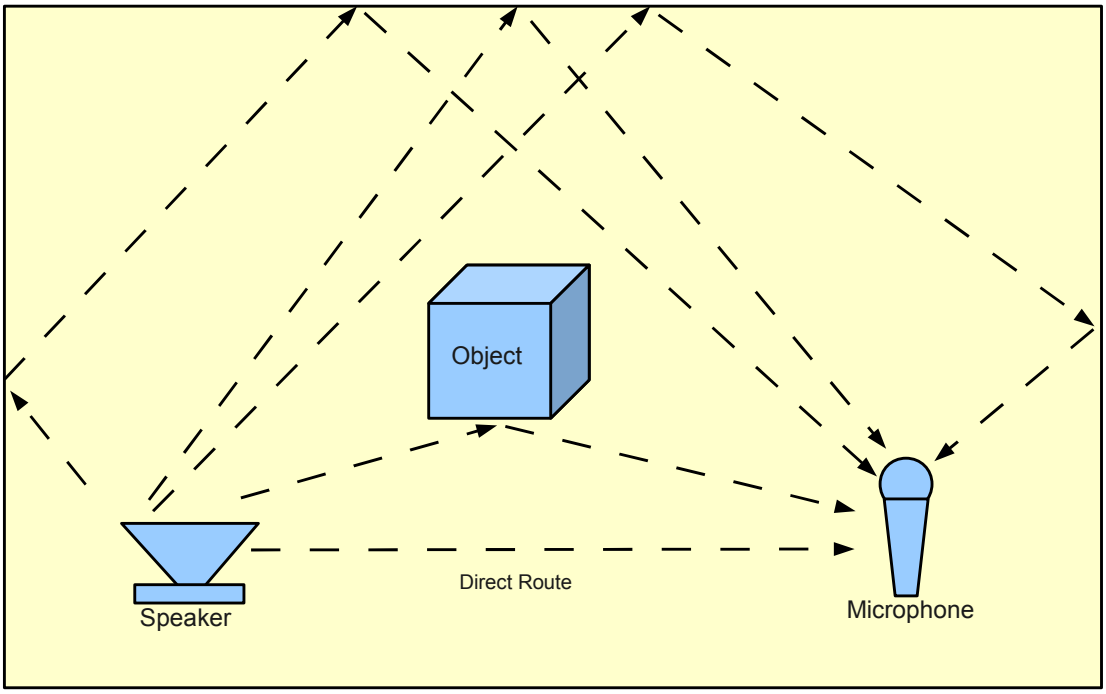

Figure 2.4: Diagram of Sound Propagation in a Loudspeaker Enclosure Microphone (LEM) Channel.

The sound from the speaker may take many different routes before it enters the microphone, as depicted in Figure 2.4. The sound impulse may take a direct route, and it may also arrive by reflecting from walls and objects and eventually reaching the microphone.

The envelope of an acoustic echo typically begins with a period of all zeroes because it takes little while for an impulse to reach the microphone by the first sound path. The impulse response then has a period of high amplitude activity as the impulse arrives at the microphone by the shortest paths. After that, the impulse reaches the microphone via paths that include several reflections, and then reflections of reflections and so on. As sound reflection absorbs a fraction of the energy, the strength of the sound reflections

die away with time, and the envelope of the impulse response dies away exponentially. In theory, the impulse response has infinite length, but in a practical sense it dies away to zero. Such an impulse response is depicted in Figure 2.5 and is termed a *dispersive* impulse. There are echo cancellation algorithms that are very efficient in identifying sparse impulse responses but do not work well with dispersive impulse responses.

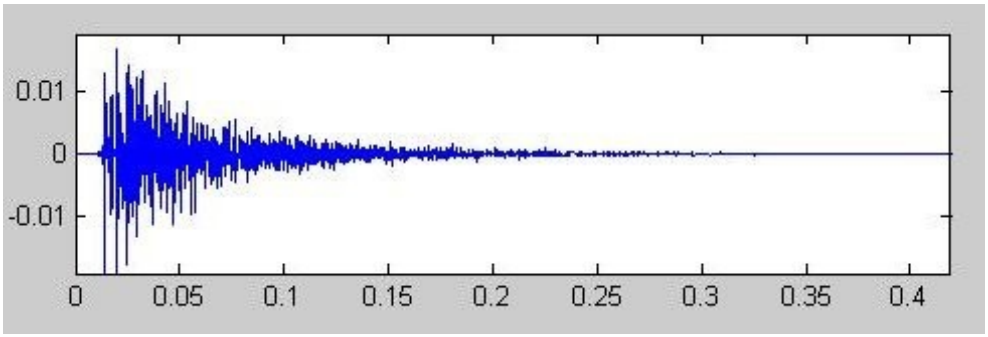

Figure 2.5: Dispersive channel. Simulated impulse response of a room. Adapted from (Lehmann 2009).

Energy at a frequency of zero is not transmitted acoustically, as explained in Subsection 7.1.3, and therefore acoustic impulse responses always have a mean of zero. As people or objects are moved within a room, then many paths from the microphone to the speaker will be altered. Such movement will cause the impulse response to be altered. If either the speaker or the microphone is moved, then all paths between the microphone and the speaker are altered. Moving the microphone or the speakers causes dramatic changes in the impulse response.

The impulse response of an acoustic echo channel may typically be in the range 50 to 300 ms and good quality cancellation can require thousands of taps (Allen & Berkley 1978). The length of the impulse response necessitates the use of a large amount of computing power and can make it impractical to use one of the more complex echo cancellation algorithms.

This is exacerbated by the fact that the echo channel may change rapidly at any time. If a door opened, or a person passed through the echo channel, the impulse response of the channel can change.

On the one hand, a faster converging algorithms is needed. Unfortunately the faster

converging algorithms tend to be the ones that require more processing power, and we cannot use those because of the long dense impulse response. Acoustic echo cancellation is is a far more challenging task because of the much longer impulse response and because the impulse response is typically more dynamic (Benesty, Gansler, Morgan, Sondhi & Gay 2001)(chapter 1). While the challenges of telephonic echo cancellation have effectively been solved, there remain important challenges for echo cancellation of audio communication systems.

2.2 Alternative Echo Management Techniques

Echo cancellation is one of the means of effectively dealing with echoes. Echo suppression, comb filters and frequency shifting are alternate technologies which may be used in order to mitigate the problem of echoes. These methods have been used historically and are still used today. We will briefly explore these alternative technologies, their advantages and their disadvantages in this section.

Echo cancellation is far more complicated and requires much greater resources than the alternative echo management techniques. Echo cancellation relies on the availability of fast cheap digital technology, requires substantial memory resources, and depending on the situation, challenges the speed of even today's digital technology. Echo cancellation was first developed in the 1960s and has been a topic of active research over the past fifty years. The most sophisticated alternative technique is frequency shifting, and we shall see in Section 2.8 that frequency shifting is one of the tools often employed as a part of practical echo cancellation.

Low tech solutions often prove to be the best, but in the case of echo management, each of the low tech solutions has major disadvantages or limitations and echo cancellation has become the dominant solution. When the channel can be adequately estimated, echo cancellation provides an almost perfect solution because the echo of the transmitted sound is removed from the desired sound without distorting it and it does so while continuing to receive the near end sound. Nevertheless, there are situations, as discussed in Section 1.2.1, where echo cancellation technology is still not adequate, and

it is in these situations where alternative technologies are still used today.

2.2.1 Echo Suppression

The hybrid echo associated with telephony inspired several schemes for echo management. Initially, the most successful was echo suppression, which had originally been developed for radio communications in the 1940s (Barney 1941). Echo suppression was adapted for telephone in the 1950s(Brady 1967).

Echo suppression determines who is speaking and disconnects the line returning to the speaker. When no party is speaking, the line is connected as normal. Under this condition, there is no sound, and therefore no echo. When only one party is speaking, the sound in the return direction is disconnected. This way, the echo does not return to the speaker. When both parties speak, the line is connected as normal. Fortunately under this condition, neither party is listening particularly well and the echo may go unnoticed (Gritton & D.W.Lin 1984).

The presence of echo suppression can cause discomfort and inconvenience to the users. One problem was that there is no sound returning to the speaker while he or she is speaking, often leading the speaker to mistakenly believe that the line is dead. Another problem with echo suppression is that interruptions are impaired. If one party is speaking and the other party interrupts, then the reconnection of the new speaker's line may not happen quickly enough and the interruption may go unnoticed (Cutler 1965).

The problems associated with echo suppression were temporarily mitigated by evolutionary improvements. These improvements included more intelligent switching strategies and having a controlled potentiometer rather than a switch (Brady 1967)(Gritton & D.W.Lin 1984). Later however, the introduction of satellite links caused long delays of up to half a second (Emling & Mitchel 1963) and the imperfections of echo suppression then became exacerbated. This encouraged research into echo management and Bell Laboratories was the first company to patent a practical echo cancelling design (Kelly & Logan 1966), known as the Least Mean Square (LMS) algorithm.

Nevertheless, echo suppression continues to be commonly used in places where echo

cancellation is inadequate. Skype, for example, uses echo suppression as one of its echo management techniques.

2.2.2 Comb Filters

Another echo management technique is that of the comb filter. The use of comb filters for echo management is still common for speakerphones and for military communications equipment.

Comb filters divide the frequency range into several subbands, with consecutive subbands being allocated alternatively to the receiving line and to the sending line (Cutler 1965). No sound gets through both filters, therefore all echoes are blocked. Frequencies that are passed by one filter are blocked by the other filter, and vice versa, as shown in Figure 2.6.

Human beings are capable of hearing the primary frequency of a tone even if that frequency is removed (Reid 2002). The human voice is rich in harmonics, and it is sufficient if some of the harmonics of the tone are passed.

When the passband frequencies are correctly selected, a lower impact on the perceived naturalness of the perceived speech results (Berkley & Courtney-Pratt 1971). When the divisions between the frequency bands are chosen at $\frac{1}{3}$ octaves, and the fundamental frequency is blocked, then the second, third, fifth and eighth harmonic will be passed, providing they lie within the telephone line frequency limits. Thus, at least some harmonics of any tone will be passed by any filter.

Comb filters provide a very effective echo management solution. Both people can talk at the same time, yet the echo is completely removed and the resulting speech is intelligible.

On the other hand, human speech passed through a comb filter, though intelligible, sounds tinny and unnatural. Also, our ability to identify someone by means of their voice is reduced.

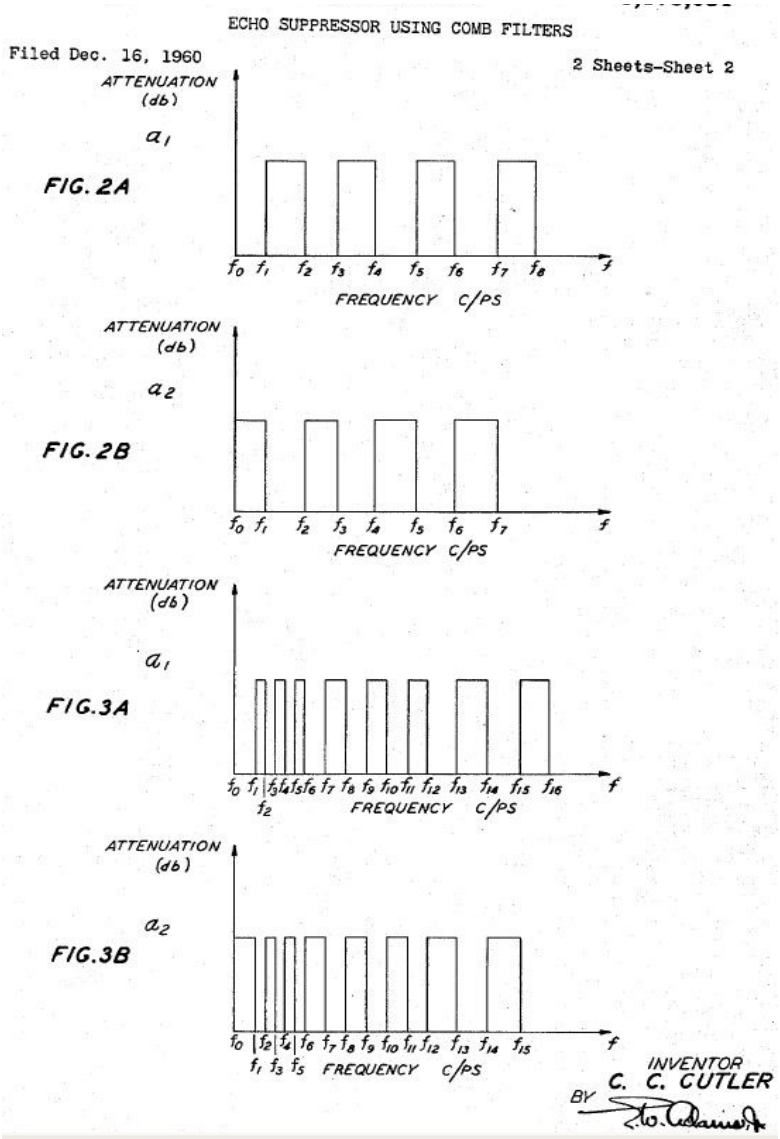

Figure 2.6: Diagram which depicts the complementary filter responses of the comb filters for the receive and the send signals. Adapted from (Cutler 1965).

As such, comb filters are typically used in applications where a channel is too dynamic for echo cancellation such as speakerphones (Phillips, Williams & Herbert 1985) and in various military radio applications.

2.2.3 Feedback Suppression

Echo at a short delay, also known as reverberation, is normally not as annoying to humans as echo with a long delay. However, if the sound from a loud speaker can re-enter a microphone and be amplified, reverberation can become intolerable. As the amplified echo of a sound becomes nearly as loud as the original sound being echoed, the reverberation becomes overwhelming. When the amplified echo of a sound becomes louder than the original sound being echoed, acoustic feedback howling otherwise known as microphone squeal, at frequencies determined by the loop delay may result.

This is common in an auditorium or concert or when using a megaphone or in hearing aids. Feedback suppression is an ongoing topic of research (Spriet, Moonen & Wouters 2010, van Waterschoot & Moonen 2011).

The open loop response is the transfer function of the channel when the feedback loop is disabled, e.g. by not feeding the microphone input back into the system. The closed loop response is the transfer function of the channel when the feedback loop is enabled. The closed loop response is a function of the open loop response, especially the feedback delay. Whenever the closed loop gain exceeds 1 for any frequency, howling feedback will occur.

Figure 2.7 helps to give an intuitive understanding of acoustic howling by showing the closed loop response when the open feedback response has a flat transfer function of unity at a delay of 2 milliseconds. It shows that the closed loop transfer function has peaks and troughs at regular intervals. As we increase the loop delay, this picture would change in a way that can be described as compression, i.e. the peaks and troughs would be much closer together.

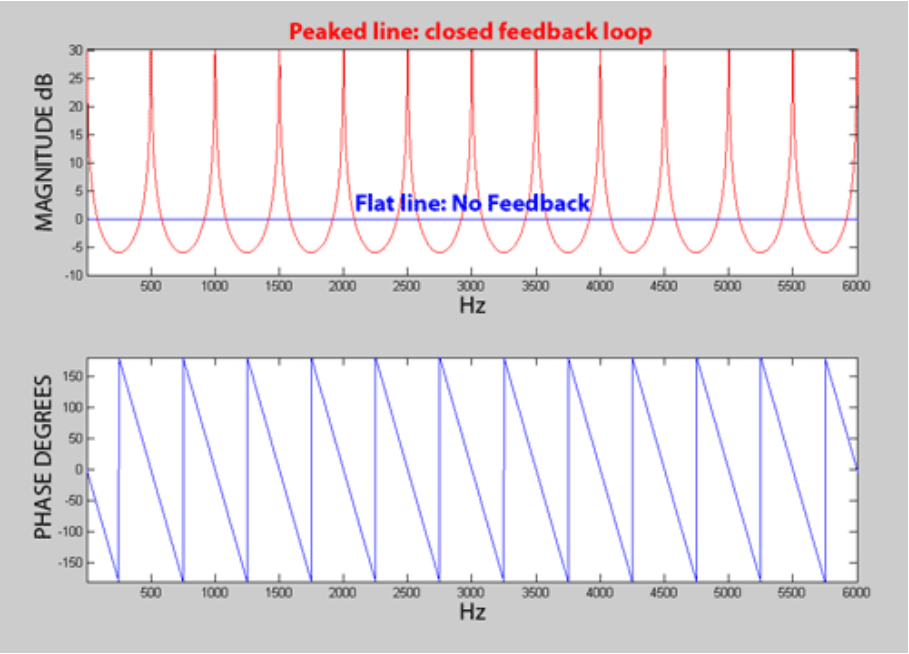

Figure 2.7: Diagram showing open (flat) and closed (peaked) loop responses at a delay of 2ms and a Gain of 1. Adapted from (Troxel 2005).

Frequency Shifting

Frequency shifting is common technique for defeating feedback in used in auditoriums, megaphones, and hearing devices (Freed & Soli 2009). Frequency shifting moves every frequency by the same amount. Frequency shifting is not used for music because it destroys the harmonic relationship of notes to each other, but is acceptable for human speech so long as the frequency is not shifted more than about 12 Hertz. Even speech, frequency shifted by more than 12 Hz, will sound distorted (Troxel 2005). If there is energy at a frequency where the gain is more than 1 then instability would normally result. But with frequency shifting, each time the energy goes through the loop it is shifted in frequency until it is eventually moved to a frequency where the loop gain is low, and the energy will be dissipated. Frequency shifting is very effective at stopping acoustic feedback howling. A frequency shift of 5 Hz will increase the amplification possible before feedback howling occurs by about 10 times (Schroeder 1964).

Frequency shifting is also an important component of subband echo cancellation, which is covered in in Subsection 2.8. It is also used in digital wireless techniques such as

OFDM, which is covered in Section 5.1.

There are many techniques of frequency shifting. The Filter Method of frequency shifting, though not the most popular method for acoustic applications, is the simplest method of frequency shifting. Other methods include the Bode method (Bode & Moog 1972), Weaver methods (Weaver 1956), Hilbert Tranform methods (Wardle 1998), and Discrete Fourier Transform (DFT) methods (Scheuing & Yang 2006), but this list is by no means exhaustive. I will describe the most basic technique, the Filter Method.

The Filter Method The frequency of a band is shifted up or down into another band by "modulating" it with the difference between actual and the desired frequency. Frequency shifting changes the frequency of a sound up or down by a fixed amount. We can represent our original signal $x(t)$by the integration of the signals at each frequency within a subband by:

$$x(t) = \int_{\omega=a}^{b} \tilde{x}(\omega)\ cos(\omega t + \phi(\omega))\ d\omega,$$

where a and b are the upper and lower bounds of the subband, $\tilde{x}(\omega)$ is the strength of the signal at frequency ω, and $\phi(\omega)$ is the phase of the signal at the frequency of ω. If we have sound in a given subband, and we wish to shift the frequency up or down by ω_s where ω_s is greater than or equal to the width of the subband, we may frequency shift a sound by first multiplying the signal by a cosine signal at the frequency of ω_s:

$$\begin{aligned} x(t)cos(\omega_s t) &= \int_{\omega=a}^{b} \tilde{x}(\omega)\ cos(\omega t + \phi(\omega))\ cos(\omega_s t)\ d\omega \\ &= \int_{\omega=a}^{b} \tilde{x}(\omega)\ cos((\omega + \omega_s)t + \phi(\omega))\ d\omega \\ &\quad + \int_{\omega=a}^{b} \tilde{x}(\omega)\ cos((\omega - \omega_s)t + \phi(\omega))\ d\omega \\ &= x_u(t) + x_l(t), \end{aligned}$$

where $x_u(t)$ is a copy of $x(t)$ shifted up by ω_s and $x_l(t)$ is a copy of $x(t)$ shifted down by ω_s. All that remains to be done is to remove the unwanted frequency by means of filtering.

In order to shift by a frequency ω_p where ω_p is less than the width of the subband, one needs to first shift up by ω_a and then down by ω_b, where $\omega_a - \omega_b = \omega_p$, and both ω_a and ω_b are greater than the width of the subband.

2.3 LMS echo cancellation

The LMS algorithm is the first echo cancellation algorithm that was developed. It is easily the simplest of all the echo cancellation algorithms as well as being the most implemented. Most echo cancellation software today are essentially implementations of algorithms that are enhancements of the LMS algorithm. For this reason, the LMS algorithm is important. In this section we consider its development, explain how it works, and analyse its performance.

2.3.1 Introduction to the LMS Algorithm

Echo cancellation was developed in the early 1960s by Bell Labs, AT&T and by the Nippon Electric Company. In Bell Telephone Laboratories, AT&T, an unpublished internal memorandum by Flanagan and de Barbeyrac documents experiments that demonstrated the feasibility of sending a test impulse through a system and using the result to synthesize a transversal filter approximating the impulse response (Sondhi 1967).

The actual impulse response of the system is rarely constant, however, and John Kelly of Bell Labs proposed the idea of using the speech signal itself in place of a test impulse in order to determine the impulse response dynamically (Sondhi 1967). A patent application for a design using these ideas was made in October 1966 (Kelly & Logan 1966). Although the patent was called "Self-Adaptive Echo Canceller", the method is now known as the Least Mean Square (LMS) echo cancellation. The patent application claims that the method achieves echo attenuation of up to 30 dB.

One month after that patent application the (Miura, Kobayashi, Sato & Nagata 1966) presented a design similar to the Flanagan and de Barbeyrac design in that it used a test pulse to discover the impulse response of the echo channel which they called a Blockless Echo Supressor (Miura et al. 1966, Miura, Sato & Nagata 1969).

As mentioned in the previous subsection, we work with discretized version of the signals shown in Figure 1.1. For the sake of describing the LMS algorithm simply, we use x to

hold the discrete samples of the input signal in reverse chronological order beginning from the present. That is

$$\mathbf{x}_j = x(t - jT) \quad : \quad j = 0...n-1 \quad ,$$

where t represents the current discrete sample time, T represents the sampling time constant and n is some arbitrary vector length greater than the effective length of the impulse response.

Then LMS algorithm may now be very simply described. We initialise our estimate of the channel, $\hat{\mathbf{h}}$ to zero. Then at each time step, we add some negative or positive value times $\mathbf{x}$ to $\hat{\mathbf{h}}$. The amount of $\mathbf{x}$ that we add will be proportional to $\hat{z}(t)$. More precisely, the update to our estimate of the channel $\hat{h}$ is given by

$$\Delta\hat{\mathbf{h}} = k\ \hat{z}\ \mathbf{x},$$

where k is an arbitrary constant between 0 and 1. At each time step the discretised estimate $\hat{y}(t)$ of $y(t)$, is given by means of the scalar product of the vectors x and h:

$$\hat{y} = \mathbf{x}^T\mathbf{h}.$$

Figure 2.8 shows a hardware implementation of the LMS algorithm in diagram form. The local connection is on the right hand side and the remote connection is on the left hand side.

The following examples may assist the reader to gain an intuitive grasp of why this works:

Example 1: We start up our algorithm in the absence of far end speech, and we receive an input $\mathbf{x}$ that is just an impulse. The channel is static. In this example $\hat{\mathbf{h}}$ will become the value of $k\mathbf{h}$. We can demonstrate this by considering the first few time steps:

- At time 0, $y(t) = h(0)$, and $\hat{y}(t) = 0$. At the same time, all values of $\mathbf{x}$ are zero, except x_0 which has the value of 1. So $\hat{h}_0$ gets kh_0 added to its current value.
- At time 1, $y(t) = h(1)$, and $\hat{y}(t) = 0$. At the same time, all values of $\mathbf{x}$ are zero, except x_1 which has the value of 1. So $\hat{h}_1$ gets kh_1 added to its current value.

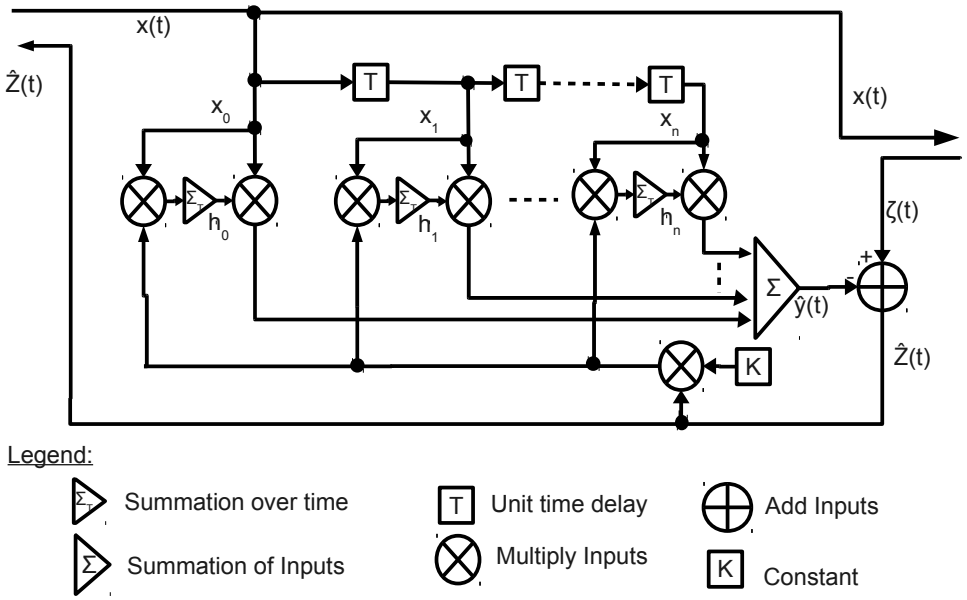

Figure 2.8: Diagram depicting a Hardware Implementation of the LMS algorithm.

- And so on, for each time discrete time step up to n, so that $\hat{\mathbf{h}} = k\mathbf{h}$.

Example 2: Our input consists of a series of such impulses n samples apart with a static channel in the absence of far end speech. Since each can each be taken separately, this would cause $\hat{\mathbf{h}}$ converge to $\mathbf{h}$ directly so that $|\mathbf{x} - \hat{\mathbf{h}}|$ would decrease as a negative exponential of time.

Example 3: Our input $\mathbf{x}$ consists of white noise, with a static channel in the absence of far end speech. Since all of the elements of $\mathbf{x}$ are uncorrelated with each other then the net interference between successive values of $\mathbf{x}$ over time would be zero, and so the system would still converge directly.

Example 4: Let our input $\mathbf{x}$ consist of white noise, and let the far end speech be uncorrelated with the input. The far end speech constantly introduces perturbations from the target solution, and this will prevent the complete convergence of $\hat{\mathbf{h}}$ to $\mathbf{h}$. Since the far end speech is uncorrelated with the input $\hat{\mathbf{h}}$ will still be an unbiased estimate of $\mathbf{h}$. If k is small then $\hat{\mathbf{h}}$ will closely approach $\mathbf{h}$, but will take a long time to do so. If k is large then $\hat{\mathbf{h}}$ will rapidly approach $\mathbf{h}$ initially, but will not closely approach it.

2.3.2 Derivation of the LMS Algorithm from the Method of Steepest Descent

The following derivation of the LMS algorithm in the absence of far end speech is based on one given in (Widrow & Stearns 1985). In it we use subscripts of $\mathbf{h}$, $\hat{z}$, and $\mathbf{x}$ to indicate the time instance, e.g. $\mathbf{x}_i$ is the value of the input vector $\mathbf{x}$ at time instance i.

If $f(\mathbf{h})$ is a real valued function, then moving in the direction $-\nabla_h(f)$ will reduce the value of f the fastest (Stewart 2006). So the method of steepest descent is

$$\hat{\mathbf{h}}_{i+1} = \hat{\mathbf{h}}_i - \mu \nabla_{\hat{h}_i}(f), \tag{2.1}$$

where μ is a constant that regulates the step size.

The function that we wish to minimise is the mean of the square of the error:

$$f(\hat{\mathbf{h}}) = E(\hat{z}^2) = E((y - \hat{y})^2).$$

If we take $\hat{z}$ to be representative of $E(\hat{z})$, then we may contemplate minimising just

$$f(\hat{\mathbf{h}}) = \hat{z}^2 = (y - \hat{y})^2.$$

Differentiating the above gives us:

$$\nabla_{\hat{h}}(f) = \nabla_{\hat{h}}(\hat{z}^2) = \begin{bmatrix} \frac{\partial \hat{z}^2}{\partial \hat{h}_1} \\ \frac{\partial \hat{z}^2}{\partial \hat{h}_2} \\ \dots \\ \frac{\partial \hat{z}^2}{\partial \hat{h}_n} \end{bmatrix} = -2\hat{z} \begin{bmatrix} \frac{\partial \hat{z}}{\partial \hat{h}_1} \\ \frac{\partial \hat{z}}{\partial \hat{h}_2} \\ \dots \\ \frac{\partial \hat{z}}{\partial \hat{h}_n} \end{bmatrix}. \tag{2.2}$$

Now, since

$$\hat{z} = y - \mathbf{x}^T \hat{\mathbf{h}} = y - \sum_{j=0}^{n-1} x_j \hat{h}_j,$$

then

$$\frac{\partial \hat{z}}{\partial \hat{h}_j} = x_j,$$

and we may substitute this into (2.2), giving:

$$\nabla_{\hat{h}}(f) = \begin{bmatrix} \frac{\partial \hat{z}^2}{\partial \hat{h}_1} \\ \frac{\partial \hat{z}^2}{\partial \hat{h}_2} \\ \dots \\ \frac{\partial \hat{z}^2}{\partial \hat{h}_n} \end{bmatrix} = -2\hat{z} \begin{bmatrix} \frac{\partial \hat{z}}{\partial \hat{h}_1} \\ \frac{\partial \hat{z}}{\partial \hat{h}_2} \\ \dots \\ \frac{\partial \hat{z}}{\partial \hat{h}_n} \end{bmatrix} = -2\ \hat{z} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} = -2\ \hat{z}\ \mathbf{x}. \tag{2.3}$$

So the method of steepest descent becomes:-

$$\hat{\mathbf{h}}_{i+1} = \hat{\mathbf{h}}_i + 2\mu\ \hat{z}_i\ \mathbf{x}_i, \tag{2.4}$$

which is our LMS algorithm where k is given by 2μ.

2.3.3 Proof that the LMS Algorithm Converges, or Remains Constant

This proof of the convergence of the LMS algorithm is based on (Kelly & Logan 1966).

With No Local Sound Let us assume initially that no one is speaking on the local telephone, i.e. that $\mathbf{z}$ is zero :-

$$\begin{aligned} \hat{z} &= y - \hat{y} \\ &= \mathbf{h} \cdot \mathbf{x} - \hat{\mathbf{h}} \cdot \mathbf{x} \\ &= (\mathbf{h} - \hat{\mathbf{h}}) \cdot \mathbf{x}. \end{aligned}$$

Thus, we may express the LMS algorithm as:

$$\Delta\hat{\mathbf{h}} = -k((\mathbf{h} - \hat{\mathbf{h}}) \cdot \mathbf{x})\mathbf{x}.$$

Let us introduce the error $\mathbf{e} = \mathbf{h} - \hat{\mathbf{h}}$. We will assume $\mathbf{h}$ is stationary, so $\Delta\mathbf{e} = \Delta\hat{\mathbf{h}}$:

$$\Delta\mathbf{e} = -k(\mathbf{x} \cdot \mathbf{e})\mathbf{x}.$$

For simplicity, let us assume that we have very small times steps and very small changes to $\hat{\mathbf{h}}$:

$$\nabla_t(\mathbf{e}) = -k(\mathbf{x} \cdot \mathbf{e})\mathbf{x}.$$

If we dot product both sides by $\mathbf{e}$, then:

$$\mathbf{e} \cdot \nabla_t(\mathbf{e}) = -k(\mathbf{x} \cdot \mathbf{e})(\mathbf{x} \cdot \mathbf{e}).$$

By applying the product rule (backwards), we get:

$$\frac{1}{2}\nabla_t(\mathbf{e} \cdot \mathbf{e}) = -k(\mathbf{x} \cdot \mathbf{e})^2.$$

Therefore

$$\nabla_t|\mathbf{e} \cdot \mathbf{e}|^2 = -2k(\mathbf{x} \cdot \mathbf{e})^2.$$

This shows that the change in the square of the error when $\mathbf{z} = 0$ is always negative or zero. If it is negative, then the error is reduced, thus cancelling $\mathbf{y}$. If it remains constant, then error stays the same.

With Local Sound When there is a voice at the local end, this desirable behavior will be disturbed. However, $\mathbf{z}$ is essentially random and uncorrelated to $\mathbf{x}$ and so according to (Kelly & Logan 1966), the addition of a voice at the near end will have only a relatively small effect.

2.3.4 Analysis of the LMS Algorithm

In this section we will analyse the stability and convergence of the LMS algorithm, and this will also allow us to draw some conclusions about the optimal step size. This analysis is based on analysis presented by (Douglas & Rupp 1998, Manolakis, Ingle & Kogon 2000, Deeba & Wood 1990).

In order to gain insight into the choice of a step size, we shall begin with a derivation of the Wiener-Hopf equation. Let us examine the mean of the square of the error, which is given by:

$$\begin{aligned} \varepsilon &= E(\hat{z}^2) \\ &= E((y - \hat{\mathbf{h}}^T\mathbf{x})^2) \\ &= E(y^2 - 2\hat{\mathbf{h}}y\mathbf{x} + \hat{\mathbf{h}}^T\mathbf{x}\mathbf{x}^T\hat{\mathbf{h}}). \end{aligned}$$

The vector $\hat{\mathbf{h}}$ is constant here, and expansion gives us:

$$\varepsilon = E(y^2) - 2\hat{\mathbf{h}}E(y\mathbf{x}) + \hat{\mathbf{h}}^T E(\mathbf{x}\mathbf{x}^T)\hat{\mathbf{h}}. \tag{2.5}$$

Let us now introduce the vector $\mathbf{p}$ and the covariance matrix $\mathbf{C}_x$:

$$\mathbf{p} = E(y\mathbf{x}) = \begin{bmatrix} E(yx_0) \\ E(yx_1) \\ \cdots \\ E(yx_n) \end{bmatrix} \tag{2.6}$$

$$\mathbf{C}_x = E(\mathbf{x}\mathbf{x}^T) = \begin{bmatrix} E(x_0x_0) & E(x_0x_1) & ... & E(x_0x_n) \\ E(x_1x_0) & E(x_1x_1) & ... & E(x_1x_n) \\ ... & ... & ... & ... \\ E(x_nx_0) & E(x_nx_1) & ... & E(x_nx_n) \end{bmatrix}. \tag{2.7}$$

Substituting these into (2.5) we get:

$$\varepsilon = E(y^2) - 2\mathbf{p}^T\hat{\mathbf{h}}. + \hat{\mathbf{h}}^T\mathbf{C}_x\hat{\mathbf{h}}. \tag{2.8}$$

This is a quadratic function in $\hat{\mathbf{h}}$. From the definition $\varepsilon = E(\hat{z}^2)$, we know that $\mathbf{C}_x$ is positive definite, and ε has a minimum when $\hat{\mathbf{h}} = \mathbf{h}$, and is positive for other values of $\hat{\mathbf{h}}$. The minimum of this quadratic occurs when the derivatives are zero.

We may differentiate (2.8) w.r.t. $\hat{\mathbf{h}}$ by application of Identity (A.14), giving:

$$\nabla_{\hat{h}}(\varepsilon) = -2\mathbf{p}^T + 2\hat{\mathbf{h}}^T\mathbf{C}_x. \tag{2.9}$$

Therefore, if we know $\mathbf{C}_x$ and $\mathbf{p}$, the following equation, kown as the Weiner-Hopf equation (Wiener 1949), gives us the $\hat{\mathbf{h}}$ at which ε is a minimum:

$$\mathbf{h} = \hat{\mathbf{h}}_{min} = \mathbf{C}_x^{-1}\mathbf{p}. \tag{2.10}$$

Substituting (2.10) into (2.8) gives us:

$$\begin{aligned} \varepsilon_{min} &= E(y^2) - 2\mathbf{p}^T\hat{\mathbf{h}}_{min} + \hat{\mathbf{h}}_{min}^T\mathbf{C}_x\hat{\mathbf{h}}_{min} \\ &= E(y^2) - 2\mathbf{p}^T\mathbf{C}_x^{-1}\mathbf{p} + (\mathbf{C}_x^{-1}\mathbf{p})^T\mathbf{C}_x\mathbf{C}_x^{-1}\mathbf{p} \\ &= E(y^2) - \mathbf{p}^T\mathbf{C}_x^{-1}\mathbf{p} \\ &= E(y^2) - \mathbf{p}^T\mathbf{h}. \end{aligned} \tag{2.11}$$

The scalar quadratic equation

$$y = ax^2 + bx + x$$

may be expressed in the form

$$y - y_0 = a(x - x_0)^2.$$

In a similar way, our quadratic (2.8) in $\hat{\mathbf{h}}$

$$\varepsilon = E(y^2) - 2\mathbf{p}^T\hat{\mathbf{h}} + \hat{\mathbf{h}}^T\mathbf{C}_x\hat{\mathbf{h}}$$

may be expressed in the form

$$\varepsilon = \varepsilon_{min} + (\hat{\mathbf{h}} - \mathbf{h})^T \mathbf{C}_x (\hat{\mathbf{h}} - \mathbf{h}).$$

proof:

$$\begin{aligned} RHS &= \varepsilon_{min} + (\hat{\mathbf{h}} - \mathbf{h})^T \mathbf{C}_x (\hat{\mathbf{h}} - \mathbf{h}), \end{aligned}$$

Substituting (2.11) for ε_{min}

$$\begin{aligned} &= E(y^2) - \mathbf{p}^T \mathbf{h} + (\hat{\mathbf{h}} - \mathbf{h})^T \mathbf{C}_x (\hat{\mathbf{h}} - \mathbf{h}) \\ &= E(y^2) - \mathbf{p}^T \mathbf{h} + \mathbf{h}^T \mathbf{C}_x \mathbf{h} + \hat{\mathbf{h}}^T \mathbf{C}_x \hat{\mathbf{h}} - \hat{\mathbf{h}}^T \mathbf{C}_x \mathbf{h} - \mathbf{h}^T \mathbf{C}_x \hat{\mathbf{h}}. \end{aligned}$$

Substituting (2.10) for $\mathbf{h}$

$$\begin{aligned} &= E(y^2) - \mathbf{p}^T \mathbf{C}_x^{-1\mathbf{p}} + (\mathbf{C}_x^{-1}\mathbf{p})^T \mathbf{C}_x \mathbf{C}_x^{-1\mathbf{p}} + \hat{\mathbf{h}}^T \mathbf{C}_x \hat{\mathbf{h}} - \hat{\mathbf{h}}^T \mathbf{C}_x \mathbf{C}_x^{-1\mathbf{p}} - (\mathbf{C}_x^{-1}\mathbf{p})^T \mathbf{C}_x \hat{\mathbf{h}} \\ &= E(y^2) - \mathbf{p}^T \mathbf{C}_x^{-1\mathbf{p}} + (\mathbf{C}_x^{-1}\mathbf{p})^T \mathbf{C}_x \mathbf{C}_x^{-1\mathbf{p}} + \hat{\mathbf{h}}^T \mathbf{C}_x \hat{\mathbf{h}} - 2\hat{\mathbf{h}}^T \mathbf{p} \\ &= E(y^2) + \hat{\mathbf{h}}^T \mathbf{C}_x \hat{\mathbf{h}} - 2\hat{\mathbf{h}}^T \mathbf{p}, \end{aligned}$$

which by equation (2.8),

$$\begin{aligned} &= \varepsilon \\ &= LHS \end{aligned}$$

QED .

Let us now introduce the the vector $\mathbf{v}$

$$\mathbf{v} = \hat{\mathbf{h}} - \mathbf{h}, \tag{2.12}$$

which permits our quadratic to be expressed as

$$\varepsilon = \varepsilon_{min} + \mathbf{v}^T \mathbf{C}_x \mathbf{v}. \tag{2.13}$$

Let us also introduce the matrix $\mathbf{Q}$ and the vector Λ. From (2.7), we can see that the matrix $\mathbf{C}_x$ is symmetric. It can therefore be diagonalized, i.e $\mathbf{C}_x$ can be expressed in the following way

$$\mathbf{C}_x = \mathbf{Q}\Lambda\mathbf{Q}^T. \tag{2.14}$$

Where Λ is a diagonal matrix of the eigenvalues of $\mathbf{C}_x$, and $\mathbf{Q}$ is a matrix of the eigenvectors of $\mathbf{C}_x$ and forms an orthonormal basis and $\mathbf{Q}^T\mathbf{Q} = I$.

Let us take the algorithm for a step in the method of steepest descent, as embodied in (2.4):

$$\hat{\mathbf{h}}_{k+1} = \hat{\mathbf{h}}_k - \mu\nabla_k(\varepsilon),$$

which, using (2.9) to substitute for $\nabla_k(\varepsilon)$, we get:

$$= \hat{\mathbf{h}}_k - \mu(2\mathbf{C}_x\hat{\mathbf{h}}_k - 2P).$$

Using (2.12) to substitute for the LHS, we get:

$$\begin{aligned}\mathbf{v}_{k+1} + \mathbf{h} &= \mathbf{v}_k + \mathbf{h} - \mu(2\mathbf{C}_x(\mathbf{v}_k + \mathbf{h}) - 2P)\\ \mathbf{v}_{k+1} &= \mathbf{v}_k - 2\mu\mathbf{C}_x\mathbf{v}_k + (2\mathbf{C}_x\mathbf{h}) - 2P),\end{aligned}$$

which, using (2.6) and (2.7) to substitute for $\mathbf{C}_x$ and P, we get:

$$\begin{aligned}&= \mathbf{v}_k - 2\mu\mathbf{C}_x\mathbf{v}_k + 2(E(\mathbf{xx})\mathbf{h} - E(\mathbf{x}y))\\ &= \mathbf{v}_k - 2\mu\mathbf{C}_x\mathbf{v}_k\\ &= (I - 2\mu\mathbf{C}_x)\mathbf{v}_k. \qquad (2.15)\end{aligned}$$

Let us introduce $\mathbf{u}$:

$$\mathbf{u} = \mathbf{Q}^T\mathbf{v}. \qquad (2.16)$$

Hence $\mathbf{u}$ is $\mathbf{v}$ expressed in the basis $\mathbf{Q}$. Substituting (2.16) into (2.15), we get:

$$\begin{aligned}\mathbf{u}_{k+1} &= (I - 2\mu\mathbf{C}_x)\mathbf{Q}\mathbf{u}_k\\ &= \mathbf{Q}^T(I - 2\mu\mathbf{C}_x)\mathbf{Q}\mathbf{u}_k\\ &= (\mathbf{Q}^T I\mathbf{Q} - 2\mu\mathbf{Q}^T R\mathbf{Q})\mathbf{u}_k\\ &= (I - 2\mu\Lambda)\mathbf{u}_k.\end{aligned}$$

It follows that

$$\mathbf{u}_{k+m} = (I - 2\mu\Lambda)^m\mathbf{u}_k. \qquad (2.17)$$

If $\hat{\mathbf{h}}$ is to converge to $\mathbf{h}$ then

$$\lim_{m\to\infty}(I - 2\mu\Lambda)^m = 0.$$

That is

$$\begin{bmatrix} \lim_{m\to\infty}(1-2\mu\lambda_0)^m & & & \\ & \lim_{m\to\infty}(1-2\mu\lambda_1)^m & & \\ & & \ldots & \\ & & & \lim_{m\to\infty}(1-2\mu\lambda_{N-1})^m \end{bmatrix} = 0.$$

Hence for convergence, the following applies for each λ_j

$$0 < \mu\lambda_j < 1$$

$$0 < \mu < \frac{1}{\lambda_j}.$$

In order for the steepest descent algorithm to converge, the step size must be greater than zero, but less than the inverse of the largest eigenvalue of $\mathbf{C}_x$.

When $x(t)$ is wide sense stationary then there is a close relationship between the eigenvalues of $\mathbf{C}_x$ and the frequency strengths within $x(t)$ (Deeba & Wood 1990). If the frequency spectrum of $x(t)$ is white, then the eigenvalues will be equal. In this case convergence will be uniform, and potentially rapid in the absence of double talk. If the frequency spectrum of $x(t)$ is colored, however, then the eigenvalues will be spread out and rapid convergence will occur in frequencies corresponding to the largest eigenvalues, but slow convergence will occur in frequencies corresponding to the smallest eigenvalues. Where a frequency is absent in the input signal $x(t)$, $\mathbf{u}$ may not converge at all. This is not ideal, but we must keep in mind that the real goal is not to minimise $\mathbf{v}$ but to minimise the mean of the square of the errors in $\hat{z}$, and where a frequency is absent in the input signal, it has no echo to cancel.

2.3.5 LMS in the Presence of Noise

(Gay & Mammone 1990) provide an example that demonstrates the effect of noise on the LMS algorithm in a very intuitive way. They use the case of the Normalised LMS algorithm with a step size $\mu = 1$, and assume that the channel is static.

Figure 2.9 depicts the operation of the NLMS algorithm with $\mu = 1$ operating with and without far end speech or noise. At any sample interval the equation

$$\mathbf{x}^T\hat{\mathbf{h}} = w \tag{2.18}$$

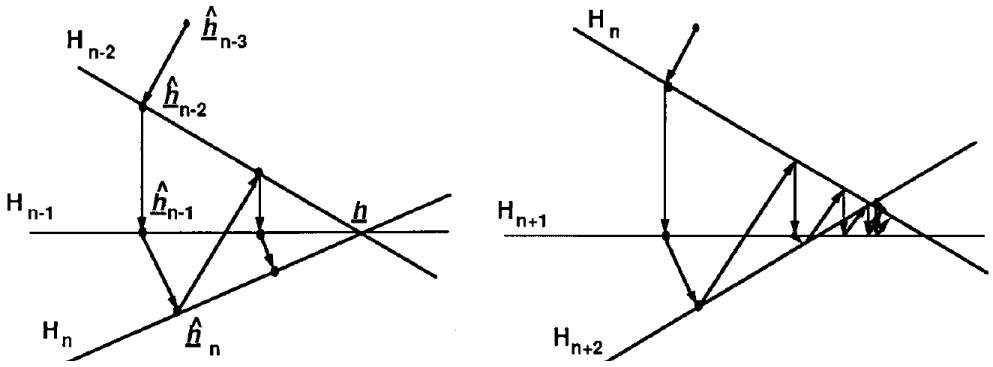

Figure 2.9: Graphical interpretations of the convergence of the NLMS algorithm with $\mu = 1$. The left hand image shows what happens when noise is absent and the right hand image shows what happens when the system is perturbed by noise. Adapted from (Gay & Mammone 1990).

defines a hyperplane. The NLMS update equation effectively projects $\hat{\mathbf{h}}$ onto the hyperplane. The diagram shows the estimate improving as a result of this projection.

In the absence of noise (2.18) becomes

$$\mathbf{x}^T \hat{\mathbf{h}} = y$$

and these hyperplanes intersect at the solution $\hat{\mathbf{h}} = \mathbf{h}$. At each step, the error, $|\hat{\mathbf{h}} - \mathbf{h}|$, becomes exponentially reduced, until in the limit $\hat{\mathbf{h}} = \mathbf{h}$.

In the presense of noise $z \neq y$ at each timestep, and as a result the hyperplanes from each timestep do not intersect at $\hat{\mathbf{h}} = \mathbf{h}$, and the LMS algorithm will not reach $\hat{\mathbf{h}} = \mathbf{h}$ in the limit.

In practice, the LMS step size will be less than the effective steps size of NLMS with $\mu = 1$, and the LMS algorithm will project $\hat{\mathbf{h}}$ only part of the way, and not actually onto the hyperplanes. Nevertheless, noise perturbs the hyperplanes, and the same thing happens, simply with shorter steps. It is often desirable to effectively remove all traces of the echo, and in order to do that the estimate $\hat{\mathbf{h}}$ needs to be a good estimate. Many of the subsequent incremental refinements of the LMS algorithm are part of an attempt to mitigate the effects of the far end noise.

2.3.6 Normalised Least Mean Squares Method

The Normalised Least Mean Squares (NMLS) method, originally published by (Nagumo & Noda 1967), is a simple but very effective enhancement of the LMS method. It differs slightly from the original LMS algorithm in that it uses a normalisation factor.

Let us take our method for updating $\hat{\mathbf{h}}$ embodied by (2.1):

$$\hat{\mathbf{h}}_{j+1} = \hat{\mathbf{h}}_j + 2\mu\ \hat{z}_j\ \mathbf{x}_j.$$

Now y and $\hat{y}$ are both proportional to $\mathbf{x}$, therefore in the absence of far end speech, $\hat{z}$ is proportional to $\mathbf{x}$. So our update is proportional to $|\mathbf{x}|^2$. Our updating is dependent on the square of the volume. At low volumes, (i.e. for a quiet talker), our algorithm could only converge very slowly, because if we made μ large in order to compensate, then at high volume (loud talker) our Λ becomes very large and our convergence could become unstable.

NLMS normalises the step size by dividing by the square of $\mathbf{x}$:

$$\hat{\mathbf{h}}_{j+1} = \hat{\mathbf{h}}_j + \frac{2\mu\ \hat{z}_j\ \mathbf{x}_j}{\mathbf{x}^T\mathbf{x}}. \tag{2.19}$$

The expression $\mathbf{x}^T\mathbf{x}$ is a near enough approximation to $E(\mathbf{x}^T\mathbf{x})$ that we may use it to evaluate convergence. For NLMS, equation (2.17) can be rewritten as:

$$\begin{aligned}\mathbf{u}'_{k+m} &= (I - 2\mu\frac{\Lambda}{E(\mathbf{x}^T\mathbf{x})})^m\mathbf{u}'_k \\ &= (I - 2\mu\frac{\Lambda}{\mathrm{tr}(C_x)})^m\mathbf{u}'_k \\ &= (I - 2\mu\frac{\Lambda}{\mathrm{tr}(\Lambda)})^m\mathbf{u}'_k.\end{aligned}$$

If $\hat{\mathbf{h}}$ is to converge to $\mathbf{h}$ then

$$\begin{bmatrix} \lim_{m\to\infty}(1 - 2\mu\frac{\lambda_0}{\sum_{j=0}^{N-1}\lambda_j})^m & & & \\ & \lim_{m\to\infty}(1 - 2\mu\frac{\lambda_1}{\sum_{j=0}^{N-1}\lambda_j})^m & & \\ & & \ldots & \\ & & & \lim_{m\to\infty}(1 - 2\mu\frac{\lambda_{N-1}}{\sum_{j=0}^{N-1}\lambda_j})^m \end{bmatrix} = 0.$$

As already mentioned, the matrix C_x is positive semi definite, and all the eigenvalues are either zero, or positive. Therefore, for convergence

$$0 \leq \mu \frac{\lambda_j}{\sum_{j=0}^{N-1} \lambda_j} \leq 1$$

for each j. So when the normalisation factor is used, our step size μ needs to be in the range $0 < \mu \leq 1$ regardless of the values in $\mathbf{x}$.

In practice, the algorithm is modified slightly to use a *regularization* term so that the algorithm remains stable during times when $\mathbf{x}$ is silent. A small value is added to the denominator so that the standard NMLS update algorithm becomes:

$$\hat{\mathbf{h}}_{j+1} = \hat{\mathbf{h}}_j + \frac{2\mu \ \hat{z}_j \ \mathbf{x}_j}{\mathbf{x}_j^T \mathbf{x}_j + \Delta}.$$

The normalisation term itself can be updated rather than recomputed, so very little extra computing overhead is associated with this enhancement of the LMS algorithm.

2.3.7 Proportionate Normalised Least Mean Squares Algorithm

Then NLMS algorithm is known to be slow for long sparse echo channels, and these are normally associated with telephonic echo and VOIP circuits (das Chagas de Souza, Tobias, Seara & Morgan 2010). The Proportionate Normalised Least Mean Squares (PNLMS) algorithm, originally published by (Duttweiler 2000), is effective wherever the impulse response is sparse, as is generally the case with the hybrid delay associated with telephonic echo. PNLMS has a different step size for each element of $\mathbf{h}$. The step size for each element of $\mathbf{h}$ is essentially proportional to the absolute of the value of that element of $\hat{\mathbf{h}}$ (within limits).

At any iteration j of the algorithm, we can define the infinity norm of the impulse response $\mathbf{h}$ of length n to be

$$l_\infty = max(h_1, \ h_2, \ ..., \ h_n).$$

And also let us define

$$l'_\infty = max(l_\infty \ , \ \delta),$$

where δ is a constant preventing l'_∞ from going to zero. Then let us define

$$g_k = max(\rho,\ l'_\infty,\ \hat{h}_k),$$

where ρ is a lower limit on the step size for any element. Without ρ, all zero elements in $\hat{\mathbf{h}}$ would not converge at all:

$$\bar{g} = \frac{1}{n}\sum_{k=1}^{n} g_k.$$

Then the formula for updating each element k of $\hat{\mathbf{h}}$ for any iteration j is given by:-

$$\mathbf{h}_{k|j} = \mathbf{h}_{k|j-1} + \mu \frac{g_{k|j}}{\bar{g}_j} \frac{\hat{z}_j \mathbf{x}_j}{\mathbf{x}_j^T \mathbf{x}_j + \Delta}.$$

Performance of PNMLS

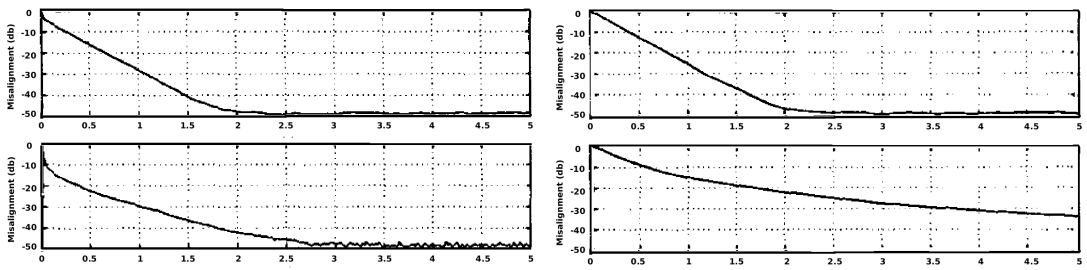

Figure 2.10: Graphs showing echo cancellation convergence obtained by simulations. The graphs in the left hand column show convergence for a sparse echo channel. The graphs in the right hand column show convergence for a dispersive channel. The graphs in the top row show convergence for NLMS. The graphs in the bottom row show convergence for PNLMS. Adopted from (Benesty & Gay 2002).

For sparse channels, PNLMS has a much faster initial convergence than does NLMS, however, after the initial convergence, PNLMS converges more slowly than NLMS. The initial faster convergence and the slower subsequent convergence is apparent in the left hand graphs of Figure 2.10. PNLMS exhibits poor performance for echo channels that are not sparse, as shown in the right hand graphs of Figure 2.10.

Enhancements of PNMLS

There have been many enhancements of PNLMS. Since acoustic channel estimation is the main focus of this dissertation, and since acoustic echo channels are normally dispersive rather than sparse, I will not attempt to document every enhancement.

(Gay 1998) proposed PNLMS++ which uses alternate NLMS and PNLMS iterations, and obtains a result where the convergence is still rapid after the initial convergence period and has a better convergence for dispersant channels. (Benesty & Gay 2002) proposed Improved PNLMS (IPNLMS) which effectively uses a weighting parameter, α, to select the proportion of update which would be given by the PNLMS method and the proportion of update given by the NLMS method. An α of -1 effectively turns the IPNLMS algorithm into the NLMS algorithm. An α of 1 effectively turns IPNLMS algorithm into the PNLMS algorithm.

The μ-law PNLMS (MPNLMS) and Sparseness Controlled PNLMS (SPNMLS) algorithms were developed by (Deng & Doroslovacki 2006), using careful analysis. The MPNLMS has the optimal convergence coefficients, but computing these in real time requires substantial computational resources. The SPNMLS algorithms is a close approximation of the MPNLMS algorithm but having a much reduced computational complexity.

The Partial Update PNLMS (PPNLMS) algorithm developed by (Dyba & Deng 2009) achieves a lower computational complexity than the NMLS algorithm by breaking the impulse response into blocks and only updating the most active blocks at each iteration, and other blocks less often than each iteration.

The Individual Activation Factor PNLMS (IAF-PNLMS) algorithm developed by (das Chagas de Souza et al. 2010) maintains an individual activation factor for each element of the discrete impulse model, rather than just using the magnitude of the impulse response element. The algorithm is new, and subject to third party verification, but according to the author's simulations it easily outperforms all of the above enhancements to the PNLMS algorithm for sparse echo channels.

2.4 The Recursive Least Squares Algorithm

The Recursive Least Squares (RLS) algorithm has an impressive performance, as shown in Figure 2.11. The main drawback with the RLS algorithm is its complexity. There are, however, Fast RLS (FRLS) algorithms (fast as in low computational complexity) that narrow the complexity gap between the RLS algorithm and the LMS method. The RLS algorithm was developed by (Godard 1974) by applying the Kalman filter (Kalman 1960) to the problem of echo channel estimation.

2.4.1 Derivation of the RLS algorithm

In this subsection we will outline a derivation of the RLS algorithm based on (Hansler & Schmidt 2004). It is much simpler than the original derivation of the RLS algorithm by (Godard 1974).

Block Based Least Mean Squares Algorithm

In this subsection we will examine the application of the standard method of least squares (Tunbull & Aitken 1932, Aldrich 1998) to the problem of determining echo channels still using the notation as illustrated in Figure 1.1. This results in a block based algorithm that is interesting because if we run the RLS algorithm on a block of data, beginning anew with each block, the RLS algorithm is essentially equivalent to the method of least squares. We will then proceed to use our resulting method of least squares to derive the RLS algorithm.

The basic echo channel model that incorporates the effect of the impulse response $h(t)$ on the input $x(t)$ and the far end noise $z(t)$ is expressed using a convolution:

$$w(t) = x(t) \otimes h(t) + z(t).$$

If we express this in discrete form, it becomes

$$\mathbf{w} = \mathbf{x} \otimes \mathbf{h} + \mathbf{z}.$$

If we express our input vector $\mathbf{x}$ in the form of a Toeplitz matrix, the convolution is replaced by a matrix multiplication. (This explained in more detail in Subsection 6.1.) Thus we may write:

$$\mathbf{w} = \mathbf{X}\mathbf{h} + \mathbf{z}.$$

The value of $\mathbf{h}$ that minimises $\mathbf{w} - \mathbf{y}$ is given by the method of least squares. The key observation here is that $(\mathbf{w} - \mathbf{y})$ is perpendicular to C, i.e. orthogonal to each column of $\mathbf{X}$:

$$\mathbf{X}^T(\mathbf{y} - \mathbf{w}) = 0. \tag{2.20}$$

One possible solution for $\mathbf{h}$ would be:

$$\mathbf{X}^T(\mathbf{X}\mathbf{h} - \mathbf{y}) = 0$$

$$\mathbf{X}^T\mathbf{X}\mathbf{h} = \mathbf{X}^T\mathbf{y}$$

$$\mathbf{h} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y},$$

assuming that $(\mathbf{X}^T\mathbf{X})$ is invertible.

Weighted Least Mean Squares Algorithm

If we assume that the real impulse response of our channel can vary, then recent results are more valid than results further in the past. An iterative method that gave equal weight to recent results and to results from the distant path could not adequately track a changing impulse response as the impulse response changed. Therefore, there needs to be a system that weights the present more than the past. The foundation for the recursive least squares algorithm is a weighted version of the method of least squares. The early formulations of the method of least squares already included a weighted version of the method of least squares.

Let us introduce a diagonal matrix $\mathbf{W}$ into (2.20), whose non zero elements contain a weighting of each sample:

$$\mathbf{X}^T\mathbf{W}(\mathbf{y} - \mathbf{w}) = 0.$$

The contribution each sample makes toward the solution is governed by the square of the corresponding element of W. We can show this by noting that the method of least

squares finds the point at which E, the sum of the squares of the errors of our estimate $\hat{\mathbf{y}}$, is a minimum:

$$E = \sum_{j=1}^{M} (\mathbf{W}_{jj}\mathbf{x}_j\hat{\mathbf{h}} - \mathbf{W}_{jj}\mathbf{y}_j)^2,$$

where M is the number of samples, and $\mathbf{x}_j$ is the transpose of the jth row of $\mathbf{X}$. Expanding gives us:

$$\begin{aligned} E &= (\sum_{j=1}^{M} \mathbf{W}_{jj}^2 \hat{\mathbf{h}}^T \mathbf{x}_j \mathbf{x}_j^T \hat{\mathbf{h}}) - (\sum_{j=1}^{M} 2\mathbf{W}_{jj}^2 \hat{\mathbf{h}}^T \mathbf{x}_j \mathbf{y}_j) + (\sum_{j=1}^{M} \mathbf{W}_{jj}^2 \mathbf{y}_j^2) \\ &= \hat{\mathbf{h}}^T (\sum_{j=1}^{M} \mathbf{W}_{jj}^2 \mathbf{x}_j \mathbf{x}_j^T) \hat{\mathbf{h}} - 2\hat{\mathbf{h}}^T (\sum_{j=1}^{M} \mathbf{W}_{jj}^2 \mathbf{x}_j \mathbf{y}_j) + (\sum_{j=1}^{M} \mathbf{W}_{jj}^2 \mathbf{y}_j^2). \end{aligned} \tag{2.21}$$

Let us introduce $\mathbf{R}$, which is a positive definite, invertible matrix, and also the vector $\mathbf{p}$:

$$\mathbf{R} = \sum_{j=1}^{M} \mathbf{W}_{jj}^2 (\mathbf{x}_j \mathbf{x}_j^T) \tag{2.22}$$

$$\mathbf{p} = \sum_{j=1}^{M} \mathbf{W}_{jj}^2 (\mathbf{x}_j \mathbf{y}_j). \tag{2.23}$$

Substituting (2.22) and (2.23) into (2.21) gives us:

$$E = \hat{\mathbf{h}}^T R \hat{\mathbf{h}} - 2\hat{\mathbf{h}}^T \mathbf{p} + (\sum_{j=1}^{M} \mathbf{W}_{jj}^2 \mathbf{y}_j^2).$$

The solution for $\hat{\mathbf{h}}$ is where E is minimum. Since this is a positive definite quadratic in $\hat{\mathbf{h}}$, it has a minimum at $\nabla_h(E) = 0$. Thus:

$$\nabla_h(E) = 2\mathbf{R}\hat{\mathbf{h}} - 2\mathbf{p},$$

and therefore:

$$\hat{\mathbf{h}} = R^{-1}\mathbf{p}. \tag{2.24}$$

From (2.22) and (2.23), we can see that the contribution to $\mathbf{R}$ and the contribution to $\mathbf{p}$ from sample j are both proportional to the square of $\mathbf{W}_{jj}$.

These weightings bring about an exponential decay of information from the past. Let us introduce a *forget factor*:

$$0 << \lambda < 1$$

and $\mathbf{W}$ becomes:

$$\mathbf{W} = \begin{bmatrix} \lambda^{M-1} & 0 & ... & 0 \\ 0 & \lambda^{M-2} & ... & 0 \\ ... & ... & ... & ... \\ 0 & 0 & ... & 1 \end{bmatrix}. \tag{2.25}$$

Updating the Matrix

The time delay involved while forming a matrix, and the numerical complexity of solving (2.24) prevents the direct application of the method of least squares from being a practical method for echo cancellation. The RLS algorithm uses continuous incremental updates to $\mathbf{R^{-1}}$, and does so in a way that the computational complexity for each sample is dominated by two matrix vector multiplies of order N.

From (2.22) and (2.25) we can see that the matrix $\mathbf{R}$ can be updated by

$$\mathbf{R}_{j+1} = \lambda^2 \mathbf{R}_j + \mathbf{x}\mathbf{x}^T.$$

But if we want to use (2.24) to find $\hat{\mathbf{h}}$, then we need to update the matrix $\mathbf{R}^{-1}$. This can be done by means of the *Woodbury Matrix Identity* (Hager 1989):

$$(\mathbf{A}^{-1} + \mathbf{u}\mathbf{v}^T)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^T\mathbf{A}^{-1}}{1 + \mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}}.$$

We can apply this formula directly in order to update $\mathbf{R}$

$$((\lambda^2\mathbf{R}^{-1}) + \mathbf{x}\mathbf{x}^T)^{-1} = (\lambda^2\mathbf{R})^{-1} - \frac{(\lambda^2\mathbf{R})^{-1}\mathbf{x}\mathbf{x}^T(\lambda^2\mathbf{R})^{-1}}{1 + \mathbf{x}^T(\lambda^2\mathbf{R})^{-1}\mathbf{x}}$$

$$\mathbf{R}_{j+1}^{-1} = (\lambda^2\mathbf{R}_j)^{-1} - \frac{(\lambda^2\mathbf{R}_j)^{-1}\mathbf{x}_j\mathbf{x}_j^T(\lambda^2\mathbf{R}_j)^{-1}}{1 + \mathbf{x}_j^T(\lambda^2\mathbf{R}_j)^{-1}\mathbf{x}_j}.$$

From (2.23) and (2.25) we can see that the vector $\mathbf{p}$ may be updated using

$$\mathbf{p}_{j+1} = \lambda^2\mathbf{p}_j + \hat{\mathbf{z}}\mathbf{x}_j.$$

A summary of the algorithm follows. The following variables are used

S: R^{-1}

ν: λ^2

u: A column vector.

v: A row vector.

- Initialise S to kI where $0 < k << 1$.
- Initialise $\mathbf{p}$ to 0.
- For each sample
  - $S \longleftarrow \nu S$
  - $\mathbf{u} \longleftarrow S\,\mathbf{x}$
  - $\mathbf{v} \longleftarrow \mathbf{u}^T$
  - $S \longleftarrow S - (\mathbf{uv})/(1 + \mathbf{vx})$
  - $\mathbf{p} \longleftarrow \nu\mathbf{p} + (\hat{z} + \hat{y})\mathbf{x}$
  - $\hat{\mathbf{h}} \longleftarrow S\,\mathbf{p}$

2.4.2 Performance of the RLS algorithm

As can been seen in Figure 2.11, the RLS algorithm features rapid convergence. In addition, it can track the channel accurately even in the presense of far end speech or noise, whereas the NLMS algorithm performs poorly in the presense of noise. These two advantages alone make it clear that the RLS algorithm is important. On the other hand, the RLS algorithm may be unsuitable for tracking dynamic channels, as it struggles to unlearn its previous knowledge of a channel.

The main problem with the RLS algorithm as described, however, is that using it to track a channel impulse response has a computational complexity of $O(n^2)$ per sample, as opposed to the NLMS algorithm which has a complexity of $O(n)$, where n is the effective length of the impulse response (Hansler & Schmidt 2004).

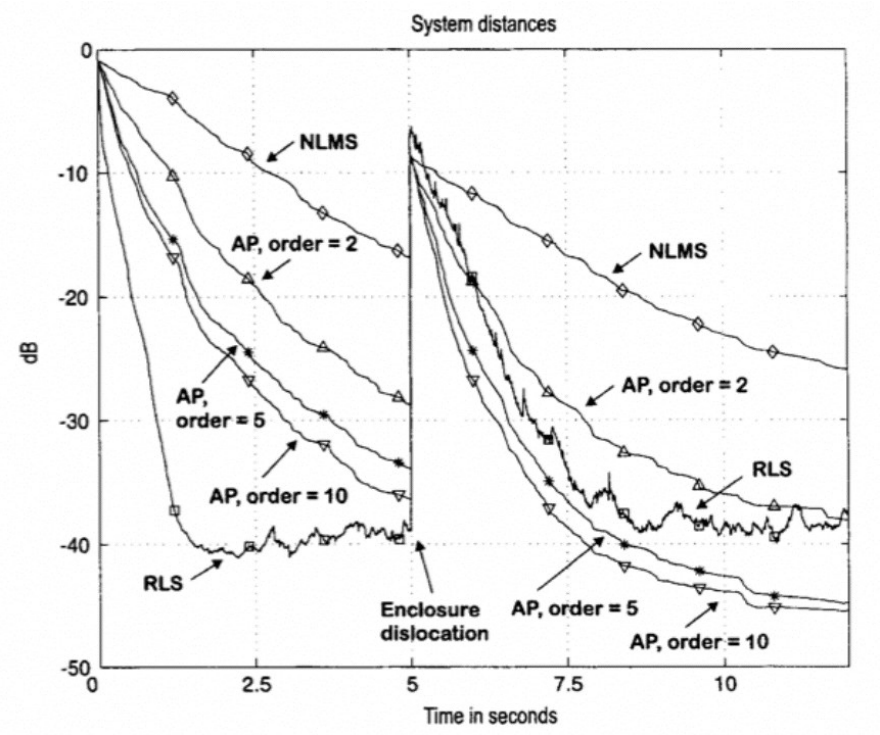

Figure 2.11: Diagram showing the convergence of the system distance for the RLS algorithm and the AP algorithm of different orders. The impulse response is changed at t=5 seconds. Adapted from (Hansler & Schmidt 2004).

It is also worth noticing that the RLS algorithm is not a Bayesian algorithm, and does not take the secondary statistics of the noise or of the channel into account. The work in Chapter 6, *Optimisation of Test Signals* shows that improvements can be made when the statistics of the noise and the channel are taken into account.

2.4.3 Fast RLS Algorithms

The Fast Kalman algorithm was applied to the problem of adaptive filtering by (Ljung, Morf & Falconer 1978). The resulting algorithm has the convergence behavior of the original RLS algorithm but a computational complexity of $O(n)$ or $11n$. Other fast RLS algorithms have been developed by (Carayannis, Manolakis & Kalouptsidis 1983, Cioffi & Kailath 1984), and these can perform adaptive equalisation with a complexity of $7n$.

These three algorithms have very attractive convergence and computational properties, but various implementers (Ljung 1983, Lin 1984, Mueller 1981) noticed unexplained and catastrophic failures in the algorithms of (Carayannis et al. 1983) and (Cioffi & Kailath 1984). By experimentation, (Lin 1984) discovered that the failures always occurred immediately after a given expression had a negative value. The algorithm of (Ljung et al. 1978) also produced meaningless values from time to time. In fact all three algorithms in their original form were rendered unusable because of their inherent numerical instability (Papaodysseus, Alexiou, Roussopoulos & Panagopoulos 2001).

The numerical problems inherent in these algorithms were analysed by (Ljung & Ljung 1985, Fabre & Gueguen i1986, Botto & Moustakides 1989), and a second generation of fast RLS algorithms, this time numerically stable, were developed. There were many different fast RLS algorithms developed, including those by (Botto & Moustakides 1989, Slock & Kailath 1988, Benallal & Gilloire 1988, Vega, Rey, Benesty & Tressens 2009).

A new technique using reduced size predictors of order m, where $m << n$, was developed by (G.V.Moustakides & S.Theodoridis 1991, P.P.Mavridis & G.V.Moustakides 1996). Fast RLS algorithms that use this technique are often called Fast Newton Transversal Filter (FNTF) Algorithms. Several fast RLS algorithms including (Benallal & Benkrid 2007, M.Arezki & D.Berkani 2009) use the technique, and they are able to

produce fast stable RLS algorithms that have a complexity of $2n + 4m \simeq 2n$. The computational complexity of these algorithms approach that of NLMS, and ought to be be a very attractive choice for implementation.

2.5 The Affine Projection Algorithm

The Affine Projection (AP) Algorithm was first published by (Ozeki & Umeda 1984). It features fast convergence and has a fast response to a dynamically changing channel.

The AP algorithm is actually a family of algorithms, each algorithm having a different *order*. Low order AP algorithms have lower computational complexity. At the limit, an order 1 AP algorithm *is* the NLMS algorithm. High order AP algorithms have faster convergence. As the order goes to infinity, the AP algorithm becomes the RLS algorithm with a forget factor of 1 (i.e. no forgetting the past).

The AP algorithm generalises the NLMS algorthm. As mentioned above, the order 1 AP algorithm *is* the NLMS algorithm. The order 2 AP algorithm is similar, but uses the current value of $\hat{z}$ and $\mathbf{x}$ as well the values of $\hat{z}$ and $\mathbf{x}$ from the previous iteration. An order m algorithm uses the values of $\hat{z}$ and $\mathbf{x}$ from the most recent m iterations.

The update for $\hat{\mathbf{h}}$ using the NLMS algorithm is given by (2.19):

$$\Delta\hat{\mathbf{h}} = 2\mu\ \mathbf{x}\ (\mathbf{x}^T\mathbf{x})^{-1}\hat{z}.$$

The update for $\hat{\mathbf{h}}$ using the AP algorithm looks very similar :-

$$\Delta\hat{\mathbf{h}} = 2\mu\ \mathbf{X}\ (\mathbf{X}^T\mathbf{X})^{-1}\hat{\mathbf{z}}, \tag{2.26}$$

where $\hat{\mathbf{z}}$ is an array of the m most recent discrete values of $\hat{z}(t)$, and the columns of $\mathbf{X}$ are the m most recent versions of $\mathbf{x}$, where m is the order of the AP algorithm. Each vector in $\mathbf{X}$ is shifted, i.e. $\mathbf{X}$ is a non-square Toeplitz matrix.

Also, in a similar manner to the NLMS algorithm, a regularization term is used in practice to avoid instability during silent periods when $\mathbf{X}$ goes to zero, so that the update algorithm becomes:

$$\Delta\hat{\mathbf{h}} = 2\mu\ \mathbf{X}\ (\mathbf{X}^T\mathbf{X} + \Delta I)^{-1}\hat{\mathbf{z}}. \tag{2.27}$$

2.5.1 Proof of Convergence of the AP algorithm

Hansler and Schmidt (Hansler & Schmidt 2004) give a proof of convergence when $\mathbf{z}$ is zero that compares the square of the instantaneous error with the square of the error at the same moment, but using the new value of $\hat{\mathbf{h}}$. If using the new value of $\hat{\mathbf{h}}$ instead of the old one reduces the absolute value of the error at each step, then the algorithm is said to converge. Since the NLMS algorithm is a special case of the AP algorithm, this is also a proof of the NLMS algorithim. The proof follows:-

Let $\hat{\mathbf{z}}$ be the actual error of the cancelled echo signal obtained by using $\hat{\mathbf{h}}$. Let $\hat{\mathbf{z}}_*$ be the error of the cancelled echo signal at the same moment, but using the updated estimate of the impulse response $\hat{\mathbf{h}} + \Delta\hat{\mathbf{h}}$ instead. Let $\Delta\hat{\mathbf{h}}$ be the change in $\hat{\mathbf{h}}$. Then:

$$\begin{aligned}
\hat{\mathbf{z}} &= \mathbf{y} - \hat{\mathbf{y}} \\
&= \mathbf{y} - \mathbf{X}^T\hat{\mathbf{h}} \\
\hat{\mathbf{z}}_* &= \mathbf{y} - \mathbf{X}^T(\hat{\mathbf{h}} + \Delta\hat{\mathbf{h}}) \\
&= \mathbf{y} - \mathbf{X}^T\hat{\mathbf{h}} - \mathbf{X}^T\Delta\hat{\mathbf{h}} \\
&= \hat{\mathbf{z}} - \mathbf{X}^T\Delta\hat{\mathbf{h}} \\
||\hat{\mathbf{z}}_*||^2 &= \hat{\mathbf{z}}_*^T\hat{\mathbf{z}}_* \\
&= [\hat{\mathbf{z}}^T - \Delta\hat{\mathbf{h}}^T\mathbf{X}][\hat{\mathbf{z}} - \mathbf{X}^T\Delta\hat{\mathbf{h}}] \\
&= ||\hat{\mathbf{z}}||^2 - \Delta\hat{\mathbf{h}}^T\mathbf{X}.\hat{\mathbf{z}} - \hat{\mathbf{z}}^T\mathbf{X}^T\Delta\mathbf{h} + \hat{\mathbf{h}}^T\mathbf{X}\mathbf{X}^T\hat{\mathbf{h}}.
\end{aligned}$$

So the change in the square of the error vector when we use the new value of $\hat{\mathbf{h}}$ is

$$||\hat{\mathbf{z}}_*||^2 - ||\hat{\mathbf{z}}||^2 = -\Delta\hat{\mathbf{h}}^T\mathbf{X}\hat{\mathbf{z}} - \hat{\mathbf{z}}^T\mathbf{X}^T\Delta\hat{\mathbf{h}} + \Delta\hat{\mathbf{h}}^T\mathbf{X}\mathbf{X}^T\Delta\hat{\mathbf{h}}.$$

Substituting for $\Delta\hat{\mathbf{h}}$ using (2.26),

$$\begin{aligned}
&= -2\mu(\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\hat{\mathbf{z}})^T\mathbf{X}\hat{\mathbf{z}} - 2\mu\hat{\mathbf{z}}^T\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\hat{\mathbf{z}} \\
&\quad + 4\mu^2(\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\hat{\mathbf{z}})^T\mathbf{X}\ \mathbf{X}^T(\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\hat{\mathbf{z}}) \\
&= -2\mu(\hat{\mathbf{z}}^T\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\hat{\mathbf{z}})^T - 2\mu\hat{\mathbf{z}}^T\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\hat{\mathbf{z}} \\
&\quad + 4\mu^2(\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\hat{\mathbf{z}})^T\ \mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\hat{\mathbf{z}} \\
&= -2\mu(\hat{\mathbf{z}}^T\hat{\mathbf{z}}^T - 2\mu\hat{\mathbf{z}}^T\hat{\mathbf{z}} + 4\mu^2\hat{\mathbf{z}})^T\hat{\mathbf{z}} \\
&= 4(\mu^2 - \mu)\hat{\mathbf{z}}^T\hat{\mathbf{z}} \\
&= 4(\mu^2 - \mu)||\hat{\mathbf{z}}||^2.
\end{aligned}$$

This is a quadratic in μ. The algorithm converges only when this is negative, which occurs in the range $0 < \mu < 1$. The algorithm converges most rapidly when this quadratic is at a minimum, i.e. when $\mu = \frac{1}{2}$.

2.5.2 Performance of the AP algorithm

As can been seen in Figure 2.11, an order 2 AP algorithm converges much more rapidly then an order 1 AP algorithm. It also achieves much better tracking and cancellation, and recovers much faster than the RLS algorithm after a dislocation. The AP algorithms of moderate order work much better than the NLMS when the $x(t)$ is frequency deficient (coloured). This has been confirmed theoretically and by simulation (Sankaran & Beex 2000).

However, an order 10 AP algorithm offers only minor improvements over an order 5 AP algorithm. Given the significantly increased complexity of higher order AP algorithms, we can recommend the AP algorithm of orders in the range 2 to 5.

2.6 Fourier Domain Echo Cancellation

An impulse response expressed in the frequency domain is known as the *transfer function*. Echo cancellation may be performed in the frequency domain with a lower computational complexity even than NLMS. One disadvantage of Fourier Domain echo cancellation is that in its original form it is a block processing algorithm, and its use imposes a blocking delay.

When the input to a linear channel is a sine (or cosine) signal, then the output from the channel is also a (delayed and attenuated) sine or cosine wave of the same frequency. Any signal and any impulse response can be represented as a linear combination of sine and cosine signals. For an input at given frequency the attenuation and phase delay completely characterises the response. Therefore, we may determine each frequency component of the channel response independently (separately from each of the other frequencies). We shall show in Chapter 6, *Optimisation of Test Signals*, that taking into

account the statistical correlation between near terms of the transfer function leads to a more efficient estimate of the channel. However, ignoring this for the moment, we will see that treating each frequency independently permits a very efficient computation of and application of the transfer function.

2.6.1 Frequency Domain Convolution

Not only can we *calculate* the impulse response efficiently in the frequency domain, but we need not convert our impulse response back to the time domain. Indeed, it is much more efficient to *apply* our transfer function directly in the frequency domain and to convert only our result y to the time domain. The application of an impulse response is a convolution in the time domain. In the frequency domain, each frequency can be treated separately, and application of the transfer function in the frequency domain involves only element by element multiplication.

2.6.2 Adaptive Processing in the Frequency Domain

Although the processing in the Fourier domain is trivial, we need to convert our input signal and our output signal from the time domain to the frequency domain and back again to the time domain. The technology that makes the Discrete Fourier Transform (DFT) a practical proposition for frequency domain echo cancellation is the Fast Fourier Transform (FFT) which was originally invented in 1805 by C.F. Gauss (Heideman, Johnson & Burrus 1985) and reinvented in 1965 by (Cooley & Tukey 1965).

There is also a temporal incompatibility with using the FFT. The time domain data arrives and is required continuously, while conversion back and forth to the frequency domain can only be performed in blocks. The frequency domain version of the LMS algorithm buffers $x(t)$ into blocks of size n, say. Clearly this buffering creates a delay of n time samples.

Figure 2.12 illustrates the frequency domain LMS algorithm. As each block of $\mathbf{x}$ becomes available, the DFT is applied producing a complex $\tilde{\mathbf{x}}$. We apply our estimated frequency domain impulse response estimate $\tilde{\mathbf{h}}$ to $\tilde{\mathbf{x}}$ giving $\tilde{\mathbf{y}}$ by using element by ele-

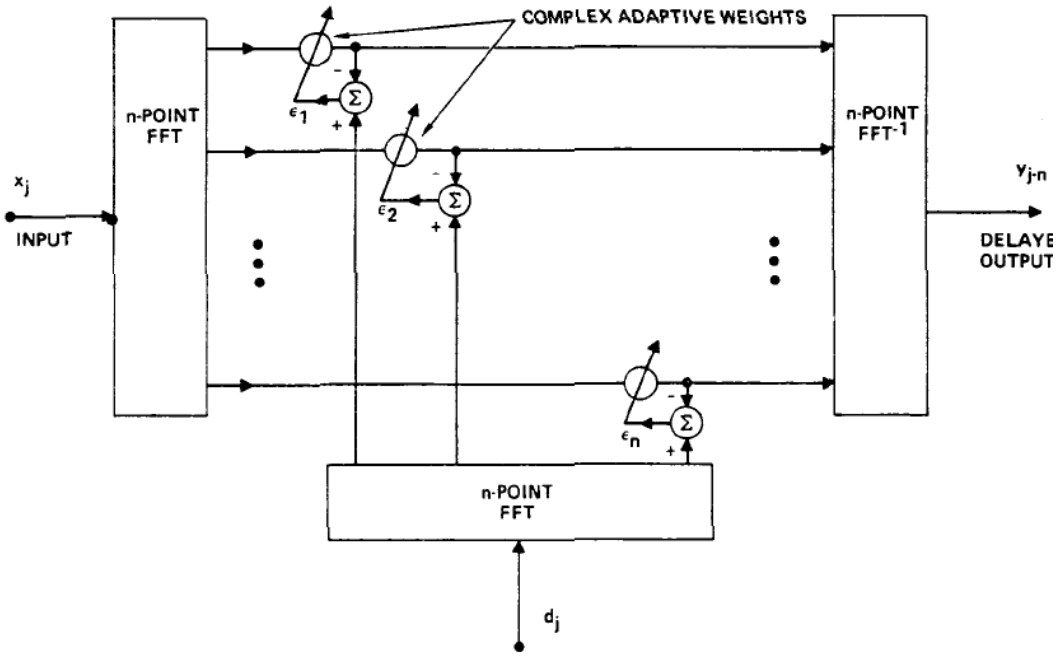

Figure 2.12: Schematic Diagram of the Frequency Domain LMS algorithm. Adapted from (Dentino et al. 1978).

ment multiplication:

$$\tilde{\mathbf{y}} = \tilde{\mathbf{h}} \diamond \tilde{\mathbf{x}}.$$

Similary and simultaneously, the estimate of the transfer function is updated in an element by element fashion as follows:

$$\Delta\tilde{\mathbf{h}} = k\ \tilde{\mathbf{w}} \diamond \tilde{\mathbf{y}}.$$

Then the inverse DFT is applied to $\tilde{\mathbf{y}}$ giving $\hat{\mathbf{y}}$ as a block.

2.6.3 The Overlap-Save Method

One detail omitted from the description of frequency domain echo cancellation so far is that the DFT only gives the correct answer for a cyclic or repeating signal $x(t)$ and $w(t)$ whose period matches the block size. In general, $x(t)$ will not be a repeating test signal, and we must therefore prepare the input and keep only results which would be correct whether or not the input signal was indeed by cyclic with the correct period. The process which pads out the input and discards unwanted results is called the overlap-save method.

Let us say that we are to process $x(t)$, in blocks of size n_x, and the array **h** has size n_h. Then the blocks which we convert to the frequency domain must have size $n_f = n_h + n_x$. For efficiency reasons, the n_f is normally chosen to be an integer power of 2. This process is illustrated in Figure 2.13.

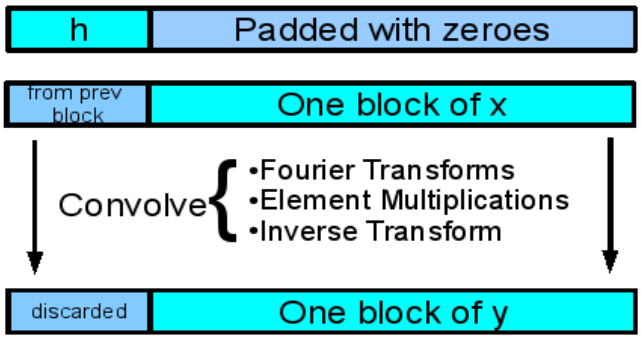

Figure 2.13: Diagram showing convolution of an impulse response with the channel input performed by conversion to and from the frequency domain.

The impulse response **h** has to be extended to size n_f by padding on the right with n_x zeroes as shown in Figure 2.13. Similarly, the blocks of $x(t)$ have to be extended to size n_f by overlapping on the left with the previous block of data (or zeroes, if this happens to be the first block).

The transform, element by element multiplication and inverse transform treats the data as a circular array. So the first n_h elements from the inverse transform are are actually the convolution of **h** with some of the previous block of **x** and some of the end of the current block of **x**, and so must be discarded. The remaining n_x elements are our result **y**.

When we process a block size of n_x then we need to convert blocks of size $n_x + n_h$. Clearly the block size needs to be large in order to perform efficient echo cancellation. However, performing the calculations in a block imposes a delay in the sound. If the block size is large, then so is the delay. So a judicious choice of n_x is necessary.

2.6.4 Performance of Adaptive Processing in the Frequency Domain

We can compare the number of floating point multiply operations used to perform the time domain version of the LMS algorithm with the number of complex multiplies used to perform the frequency domain version of the LMS algorithm. Given an impulse response of n taps, if we wish to perform n adjustments using the time domain LMS algorithm, then we need to perform n^2 multiplies and another n^2 multiplies to perform the cancellation. To do the same calculation in the frequency domain requires $\frac{3}{2}n\ log_2(n)$ complex multiplies. For acoustic echo cancellation, which may have a long effective impulse response, this difference is becomes very significant. For an impulse response of 1000 taps, the frequency domain algorithm can be 100 times faster (Dentino et al. 1978).

The method we have described here could be improved. Frequency domain methods essentially amount to a computationally efficient version of the LMS algorithm. They do not take into account the correlation between different frequencies of the channel, the total noise power, or the different frequency strengths of the noise, as we do in Chapter 6, *Optimisation of Test Signals*.

2.7 Using Orthogonal Transforms

The LMS, AP and RLS families of echo cancellation algorithms all work by directly estimating the time domain impulse response. Fourier domain echo cancellation, on the other hand is an example of a way in which we can represent an echo channel as a linear combination of orthogonal basis vectors and perform the echo cancellation in that basis. The orthogonality property of our chosen basis functions is an important property, leading to significant gains.

There are many other orthogonal bases that one might use, and each has its own set of advantages. Indeed there are too many different types of basis functions or vectors that have been used or proposed for channel estimation for us to explore them all. Fourier domain echo cancellation, for example, effectively uses the DFT matrix F as an

orthogonal basis. The DFT matrix is a somewhat "natural" choice of of an orthogonal basis because a sinusoidal input to a linear channel gives a sinusoidal input and this property leads to substantial gains. We shall examine some more interesting uses of orthogonal basis functions in this section.

2.7.1 Using Principle Component Basis Functions

(Wang, Yen & Principe 1996) and (Grant & Gay 2004*b*) proposed the use of the orthogonal vectors resulting from the Principle Component Analysis (PCA) of the historical record of a given acoustic echo channel (i.e. room or environment). The use of only the principle components as a basis would give the theoretically shortest possible representation of the impulse response. Since advanced methods such as the RLS algorithm are of complexity O_{n^2} per data sample, where n is the length of the impulse response, the use of PCA would dramatically reduce the complexity, and permit the use of advanced echo cancellation techniques.

(Grant & Gay 2004*b*), however, did not proposes the use of only the principle vectors. They proposed the use of the full rank principle component matrix. They pointed out that a non sparse impulse response as is normally found in acoustic echo cancellation becomes sparse when expressed in principle components, and therefore is suitable for use with PNLMS. The rapid convergence normally associated with PNLMS when applied to telephonic echo cancellation can thus be achieved for non sparse echo channels, by using a principle components orthogonal basis.

(Grant & Gay 2004*b*) demonstrated the use of PCA by performing simulations of a room where the speaker and microphone positions are fixed, but a certain ideal piece of furniture is moved to random positions in the room. The simulations show a superior convergence of the echo cancellation, at the expense of greatly increased computational complexity.

However, it turns out that PCA echo cancellation has a major flaw. An acoustic environment with a variable temperature cannot be adequately represented by a relatively small number of principle components in because the speed of sound varies with tem-

perature. Therefore echo paths lengthen or compress as the temperature varies and the PCA matrix for a given acoustic environment becomes very different at different temperatures. This did not become apparent from the simulations of (Grant & Gay 2004*b*) because temperature variation was not modelled as part of the simulations. (Grant & Gay 2004*a*) proposed the use of a different PCA basis for each room temperature, but the overhead of keeping many large PCA matrices greatly reduces the attractiveness of PCA echo cancellation.

PCA bases may still be useful in certain circumstances. One example would be where air conditioning keeps room temperatures constant. (Cabell 1998) provided another example by using PCA based active noise management to neutralise vibrations transmitted through the hull of an aircraft.

2.7.2 Using Laguerre Basis functions

In frequency domain echo cancellation, the impulse response is modelled by representing the impulse response as a combination of orthonormal impulse responses. The echo canceller designed by (Davidson & Falconer 1991) provides another example where the impulse response is represented by a combination of orthonormal impulse responses.

The LMS algorithm and its derivatives are implemented either physically or conceptually using a tapped delay line, which produces our input vector $\mathbf{x}$. The next element of the input vector $\mathbf{x}$ is obtained by passing the current one through an elementary delay filter shown as a unit time delay, or "T" block in Figure 2.8. The transfer function of these "T" blocks in the notation of Z transforms, is $\frac{1}{z}$.

The echo canceller of (Davidson & Falconer 1991) uses a similar tapped line, as shown in Figure 2.14, in order to produce a set of filters whose impulse responses are the discrete weighted Laguerre functions. These impulse responses have infinite length, and are orthonormal, i.e.

$$\sum_{k=0}^{\infty} L_n(k)L_m(k) = 0, \quad m \neq n$$

$$\sum_{k=0}^{\infty} L_n(k)L_m(k) = 1, \quad m = n,$$

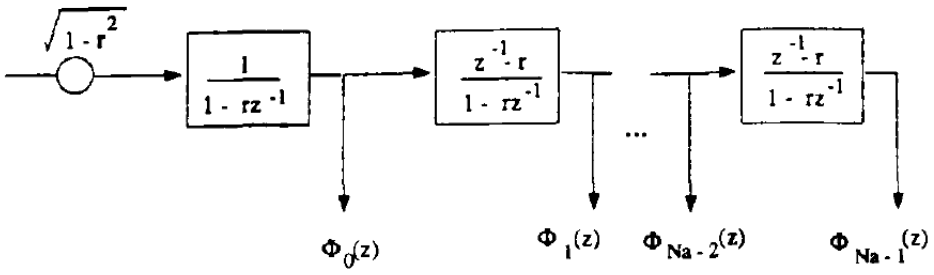

Figure 2.14: Depiction of tapped line that generates impulse responses which are the orthogonal discrete weighted Laguerre functions. Adapted from (Davidson & Falconer 1991).

where $L_n(k)$ is the Laguerre function of order n. The elementary filters of the tapped line here are not the unit delay filter $\frac{1}{z}$, but infinite impulse responses given by the transfer function

$$U(z) = \frac{z^{-1} - r}{1 - rz^{-1}},$$

where r is a parameter chosen to limit the effective length of the impulse response.

The echo cancellation proceeds much like ordinary LMS echo cancellation. The tapped line generates a vector of outputs ϕ, each of which is the input $\mathbf{x}$ convolved with the Laguerre impulse responses. The estimated channel response $\hat{y}$ is generated by combining these outputs $\phi(t)$ according to the weights ω, which take the place of the impulse response vector $\mathbf{h}$, as used in the LMS algorithm.

The advantage of this scheme over LMS is that the number of outputs from the tapped line can be substantially less than the length of the impulse response. In practice, this scheme is combined with more conventional echo cancellation, and only used to cancel the tail of a very long impulse response. This design has a very simple hardware implementation.

2.7.3 Using Arbitrary Orthogonal Basis Functions

A general scheme for performing echo cancellation (specifically NLMS and PNLMS) in an arbitrary orthonormal basis set was proposed by (Grant & Gay 2004*b*). The proposal may be summarised as follows:

Let us express the impulse response $\mathbf{h}$ as a linear combination $\mathbf{b}$ of an arbitrary orthonormal basis U:

$$\mathbf{h} = U\mathbf{b}.$$

Similarly, the input vector $\mathbf{x}$ is expressed as a linear combination $\mathbf{s}$ of the same orthonormal basis :-

$$\mathbf{x} = U\mathbf{s}.$$

The computation of the $\hat{y}$ is given by:-

$$\begin{aligned} \hat{y} &= \mathbf{x}^T\mathbf{h} \\ &= (U\mathbf{s})^T(U\mathbf{b}) \\ &= \mathbf{s}^T\mathbf{b}. \end{aligned}$$

The update of $\mathbf{h}$ using NLMS is given by

$$\begin{aligned} \Delta\mathbf{h} &= \frac{\mu\hat{z}\mathbf{x}}{\mathbf{x}^T\mathbf{x}+\delta} \\ \Delta(U\mathbf{b}) &= \frac{\mu\hat{z}(U\mathbf{s})}{(U\mathbf{s})^T(U\mathbf{s})+\delta}. \end{aligned}$$

The update of $\mathbf{b}$ using NLMS is therefore

$$\Delta\mathbf{b} = \frac{\mu\hat{z}\mathbf{s}}{\mathbf{s}^T\mathbf{s}+\delta}.$$

We can apply PNLMS, by introducing the diagonal matrix for M in the usual way:

$$M = f(\rho, \delta, \mathbf{b}),$$

and our update formula for b becomes

$$\Delta\mathbf{b} = \frac{\mu\hat{z}M\mathbf{s}}{\mathbf{s}^TM\mathbf{s}+\delta}.$$

2.8 Subband Echo Cancellation

Subband echo cancellation is not an echo cancellation algorithm, but a mthod of dividing the echo cancellation problem in order to allow reduced complexity echo cancellation using an echo cancellation algorithm.

Subband echo cancellation works by splitting the incoming signal $x(t)$ and the near end signal $w(t)$, which we will call fullband signals, into separated frequency subbands. Each separated frequency subband is then frequency shifted into the frequency range of the lowest subband, where a reduced sampling frequency may be used. Echo is then cancelled out in each subband, using standard echo cancellation techniques. The subband results are then combined into a fullband result.

Although it is possibly to perform the echo cancellation to construct a fullband $\hat{y}$, and then perform $\hat{z}(t) = w(t) - \hat{y}(t)$, subband echo cancellation is not normally performed that way. Instead, the subtraction is normally performed in the subband. Hence, the high level block diagram no longer looks like the one in Figure 1.1, but instead looks like the one in Figure 2.15. The reason for this is that we have already calculated the subband error signal $\hat{z}(t)$ in order to estimate the channel as part of the error cancellation algorithm.

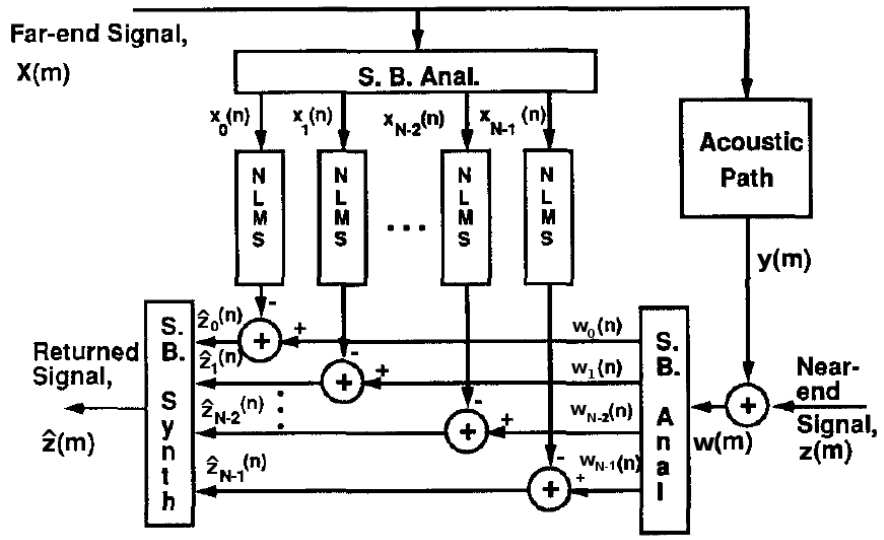

Figure 2.15: Basic block diagram of subband echo cancellation. Adapted from (Gay & Mammone 1990).

An analysis filter bank acts upon $x(t)$ in order to break the signal into m separate subbands, which are normally have equal width. An identical analysis filter bank is applied to the fullband $w(t)$ giving m separate subbands. A matching synthesis filter is used to combine the error cancelled signals $\hat{z}(t)$ from each subband and creates the

fullband $\hat{z}(t)$ (Gilloire 1987).

2.8.1 Critical Subsampling

If our sampling frequency was f_s then our resulting fullband has a frequency range of 0 to f_f, where $f_f = \frac{f_s}{2}$. If we split our fullband signal into m subbands, then our lowest subband has a frequency range of 0 to $\frac{f_f}{2m}$. The analysis filter extracts each subband from the fullband signal, and shifts each frequency band down into the frequency of the lowest frequency band. In practice, the value of m tends to be in the range 4 to 8. The downshifted subband is then subsampled at a rate of $\frac{f_s}{r}$ where r is the sampling reduction ratio. r is normally a number in the range $\frac{3}{4}m$ to m. The case where $r = m$ is known as critical subsampling.

According to some texts, critical subsampling is not recommended (Benesty et al. 2001) (Hansler & Schmidt 2004). These texts point out that the aliasing with adjacent subbands becomes a problem as r approaches m. With r significantly less then m the aliasing is much less of a problem, and it is possible to apply a standard fullband echo cancellation algorithms within each subband. Several papers focus on the use of critical subsampling, however. Some augment the analysis filter with cross terms from adjacent bands (Gilloire 1988) and some use modified versions of the fullband algorithms which expect and compensate for the resulting aliasing (Gordy & Goubran 2006).

2.8.2 The Analysis and Synthesis Filters

The quality of the analysis and synthesis filters used are critical to the quality of subband cancellation. In order to work perfectly, perfect subband filtering would be needed, i.e. infinite stopband attenuation and a zero length transition region between the pass bands and the stop bands. This would require infinite length FIR filters, requiring infinite resources and an infinite delay (Benesty et al. 2001). In practice a compromise must be found between echo cancellation quality and delay. Hence, subband filtering must necessarily impose both distortion and delay.

The technology involved in these filter banks is a topic for study on its own. Not only

must the filters have narrow transition bands, low ripple, good stop band filtering, and a relatively short length, but the analysis filter and the synthesis filters must be matched pairs. Filter banks suitable for subband cancellation have the property that a signal $x(t)$ split into subbands and immediately reconstructed by the synthesis filter gives $x(t-\tau)$ precisely, where τ is the combined time delay of the analysis and the synthesis filters. Such filters are called *perfect reconstruction filters*.

Quadrature Mirror Filters are a popular class of perfect reconstruction filters. These are designed for the breaking of a fullband into two subbands. Such filters may be applied repeatedly in order to break a fullband into subbands whose number is an integer power of two. Many papers have been written on this topic, and they are often designed using optimisation techniques (Nguyen & P.P.Vaidyanathan 1990, Kumar, Singh & Anand 2008, Johnstone 1980).

2.8.3 Computational Complexity

The reduced computational complexity results from the sampling reduction. The impulse response now has its length reduced by r, so the complexity of convolution and the complexity of adaption for each iteration is reduced by a factor of r. Also because of the subsampling, the rate of performing iterations is also reduced by a factor of r. But of course, there are now m subbands to process. Hence the overall computational complexity is reduced by a factor of approximately $\frac{r^2}{m}$ (Gilloire 1987).

In addition to the above mentioned reduction in computational complexity, there are other benefits associated with subband echo cancellation:

- Depending on how the distribution of the power spectral densities, coloured fullband excitation may well be white within each subband. As discussed in Section 2.3.4, the step size constant is generally a compromise. Where the strength of frequencies of the excitation signal are lacking, a larger step size is best. Where the strength of frequencies are strong, a smaller step size is best. So in the case of colored excitation signals, we need to choose a step size that is a compromise. If the excitation signal in each subband is white, however, the appropriate step

size will be the same for all frequencies of the subband. Hence convergence can be faster.

- Because wall coverings tend to absorb high frequencies more then they absorb low frequencies, impulse responses of high frequency subbands are generally shorter. Engineers may may well choose to model shorter impulse responses (Hansler & Schmidt 2004), thus further reducing the computational complexity.
- It is not necessary to perform echo cancellation in every subband (Benesty et al. 2001).
- Parallel processors may be used to cancel the echo in each subband so that the recuction in complexity of $\frac{r^2}{m}$ will lead to a speedup of r^2.

One downside to subband echo cancellation is that the filter banks must impose both delay and signal distortion.

2.8.4 Recent Advances in Subband Processing

- Pradham and Reddy have designed and simulated a subband echo canceller which takes advantage of the subband structure and manage to produce an algorithm which converges faster as the number of subbands m is increased (Pradham & Reddy 1999).
- Schuldt et al have proposed a modified subband structure that does not impose a processing delay (Schuldt, Lindstrom & Claesson 2008).

2.9 Double-talk Detection and Mis-adaptation Detection

The term double-talk originates from telephony, and refers to the condition where both parties are speaking concurrently. It is difficult to achieve high quality echo cancellation during periods of double-talk for reasons that have been explained in Subsection 2.3.5. If the current estimate of the channel is close to the actual channel, then further adaption during periods of double-talk will most likely cause misalignment, rather

than convergence. Most practical echo cancellation implementations use double-talk detection.

For successful echo cancellation, the echo channel has to be slowly changing, but in general, echo channels are not static. It is useful to be able to detect when the model of the echo channel is no longer accurate and further adaptation is needed.

Modern algorithms for detecting that the current estimate of the channel is out of date can also detect double-talk. These two topics are closely related and are treated together in this section.

2.9.1 General Procedure for Double-talk Detectors

In acoustic and telephonic echo cancellation, a double-talk detector is used to detect the double-talk condition. They generally adhere to the following procedure:-

1. A detection statistic θ is created by processing some of the available inputs $x(t)$, $\hat{y}(t)$, $w(t)$ and $\hat{z}(t)$ over time. The evaluation of θ may be achieved either in the time domain or the frequency domain.

2. The detection statistic is compared to a preset threshold T, and doubletalk is declared whenever $\theta < T$.

3. Once doubletalk is detected, adaption of the echo cancellation is paused or retarded, for at least a fixed period of say, t_h. Normal echo cancellation adaptation is only resumed if $\theta \geq T$ continuously over t_h seconds.

2.9.2 The Geigel Algorithm

Double-talk detectors were developed for use with echo suppression in the 1950s and studied in detail in the 1960s (Gould 1964). The Geigel algorithm was patented by Geigel in 1975 (Geigel & LaMarche 1975).

The Geigel algorithm examines the ratio between the maximum value of the far end

signal, $x(t)$ over a given time window and the instantaneous value of the near end signal $w(t)$. If this ratio is it is greater than a certain threshold T, then double-talk is detected. Specifically, $\theta = \frac{max(|x_n|,...,|x_{n-L+1}|)}{|y_n|}$. where L is the number of time constants for determining the envelope of $x(t)$. Double-talk is detected if $\theta < T$, where T is the inverse of the expected hybrid attenuation (typically in the range 1.4 - 2). For the purpose explaining the operation of the Geigel algorithm, I will refer to instances when $\theta < T$ as a double-talk trigger.

The double-talk detector will indicate a double-talk condition as soon as as a double-talk trigger occurs. The double-talk detector will continue to indicate double-talk for a period of time according to a mechanism that will be described in the next few paragraphs. There may be transient spikes on the line, which will cause a double-talk trigger, but are not considered as true double-talk conditions. The mechanism exist so that these transient spikes are treated differently to double-talk, and double-talk will be indicated for a much shorter period of time.

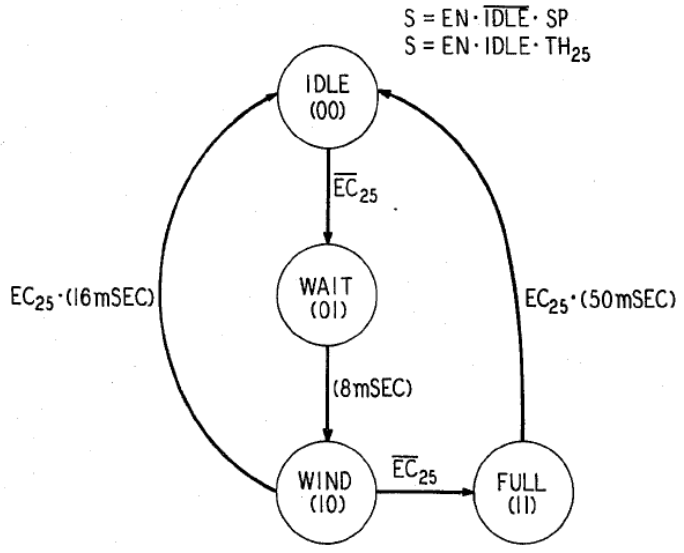

Figure 2.16: State diagram showing the operation of the Geigel algorithm. Adapted from (Geigel & LaMarche 1975).

The different states of the algorithm are shown in Figure 2.16. Here, the IDLE state is the only state that does not indicate double-talk. Double-talk is indicated during all of the other states of the double-talk detector. When we are in our IDLE state, any double-talk trigger will cause the double-talk detector to enter a WAIT state, where it will remain until a timer associated with the WAIT state expires. The timer typically

has a duration of 8 ms.

Then, it will enter a WIND (window) state, which is also associated with a timer (typically 16 ms). Any double talk trigger encountered while the detector is in the WIND state will cause the detector to enter the FULL state. Otherwise if the timer expires with no double-talk trigger then the detector will return to the IDLE state.

The FULL state also has a timer (typically 50ms), which will be restarted whenever double-talk is triggered. The double-talk detector will return to the IDLE state only when the timer expires.

2.9.3 Algorithms based on Correlation

The Geigel algorithm is both conceptually and computationally simple. It has become the industry standard for most telephonic echo cancellation implementations (Huang & Benesty 2004). For acoustic applications, however, its performance is somewhat unreliable (Cho, Morgan & Benesty 1999).

Acoustic echo cancellation requires a different class of doubletalk detection. These are normally based on estimation of the correlation between the far end signal, $x(t)$ and the signal with the echo fully or partially removed $\hat{z}(t)$. We can expect the near end signal $z(t)$ to be completely uncorrelated with the far end signal $x(t)$, while any residual uncancelled echo $y - \hat{y}$ will be correlated with x.

2.9.4 The Cross Correlation Method

The Cross Correlation (CC) method, developed by (Ye & Wu 1991), is a system that uses the correlation between $x(t)$ and $w(t)$ as a means of estimating the estimation error $|\hat{y}(t) - y(t)|$. That is, it can tell when $\hat{\mathbf{h}}$ is not accurate and the error cancellation needs more adaptation.

This method uses the average of correlations between $\hat{z}$ and the discrete values of the

input **x** going into the past:

$$\theta = \frac{1}{n}\sum_{j=0}^{n-1}\frac{E[x(k-j)\hat{z}(k)]}{\sqrt{E[x^2(k-j)]E[\hat{z}^2(k)]}}.$$

It performs this estimation using time based exponential decay as a means of tracking the recent past. This is achieved using an exponential weighting factor using λ in the range $0.9 < \lambda < 1$. The recent power in $\hat{z}$ is estimated by:

$$P^2_{\hat{z}}(n) = \lambda P_{\hat{z}}(n-1) + (1-\lambda)\hat{z}(n)^2,$$

where n is the current discrete sample time. The recent power in each element of the vector **x** are given by:

$$P^2_{x_j}(n) = \lambda P_{x_j}(n-1) + (1-\lambda)x^2(n-j).$$

The cross covariances between $\hat{z}$ and the elements of **x** are given by:

$$P_{\hat{z}x_j}(n) = \lambda P_{\hat{z}x_j}(n-1) + (1-\lambda)\hat{z}(n)x(n-j).$$

The cross correlations are given by:

$$C_j(n) = \frac{P_{\hat{z}x_j}(n)}{P_{\hat{z}}(n)\ P_{x_j}(n)}.$$

The decision variable is given by:-

$$\theta = \frac{1}{n}\sum_{j=0}^{N-1}|C_j(n)|.$$

If $\theta \leq T$, then the echo canceller is said to have converged, where T is our chosen decision threshold. Otherwise the echo canceller should continue to adapt.

This scheme provides a system to tell if the echo cancellation has converged. Used alone this scheme will not detect doubletalk. One disadvantage of this scheme is that it requires a substantial amount of computational resources.

Normalised Cross Correlation Method

A disadvantage with the Cross Correlation method in its original form is that the optimum value for the chosen decision threshold T depends on the statistics of the

signals and of the echo path. It is therefore not possible to choose a good value of T in advance. (Benesty, Morgan & Cho 2000) proposed the Normalised Cross Correlation (NCC) method. This is further refined in (Benesty et al. 2001)(chapter 6).

The term $\mathbf{h}^T\mathbf{C}_{xx}\mathbf{h}$ is the power in $y(t)$:-

$$\begin{aligned}\mathbf{h}^T\mathbf{C}_{xx}\mathbf{h} &= \mathbf{h}^T E[\mathbf{x}\mathbf{x}^T]\mathbf{h} \\ &= E[\mathbf{h}^T\mathbf{x}\mathbf{x}^T\mathbf{h}] \\ &= E[y^2] \\ &= \sigma_y^2.\end{aligned}$$

The covariance vector between $x(t)$ and $y(t)$ is given by

$$\begin{aligned}\mathbf{r}_{xy} &= E[\mathbf{x}\mathbf{y}] \\ &= E[\mathbf{x}\mathbf{x}^T h] \\ &= \mathbf{C}_{xx}\mathbf{h}.\end{aligned}$$

It is also worth noticing that

$$\begin{aligned}\mathbf{r}_{xw} &= E[\mathbf{x}(\mathbf{y}+\mathbf{z})] \\ &= E[\mathbf{x}\mathbf{y}] + E[\mathbf{x}\mathbf{z}],\end{aligned}$$

and since $z(t)$ is uncorrelated with $x(t)$,

$$\begin{aligned}&= E[\mathbf{x}\mathbf{y}] \\ &= \mathbf{r}_{xy}.\end{aligned}$$

Because $\mathbf{C}_{xx}$ is symmetrical, we may express the power in y in terms of the covariance vector:-

$$\begin{aligned}\sigma_y^2 &= \mathbf{h}^T\mathbf{C}_{xx}\mathbf{h} \\ &= \mathbf{h}^T\mathbf{C}_{xx}^T\mathbf{C}_{xx}^{-1}\mathbf{C}_{xx}\mathbf{h} \\ &= \mathbf{r}_{xw}\mathbf{C}_{xx}^{-1}\mathbf{r}_{xw}.\end{aligned}$$

The statistic proposed by (Benesty et al. 2000) is

$$\begin{aligned}\theta &= \frac{\sigma_y}{\sigma_w} \\ &= \sqrt{\frac{\sigma_y^2}{\sigma_w^2}} \\ &= \sqrt{\mathbf{r}_{xw}(\sigma_w^2\mathbf{C}_{xx}^{-1})\mathbf{r}_{xw}}.\end{aligned}$$

Clearly $0 \leq \theta \leq 1$. Furthermore, θ is not subject to changes in the echo path (Benesty et al. 2000), and we may use a given value of T for all echo cancellation situations.

(Benesty et al. 2001)(chapter 6) suggests evaluation terms required for the calculation of this statistic by means of weighted exponential decay:

$$\mathbf{r}_{xw}(n) = \lambda\mathbf{r}_{xw}(n-1) + (\lambda-1)\mathbf{x}(n)w(n)$$

$$\mathbf{C}_{xx}(n) = \lambda\mathbf{C}_{xx}(n-1) + (\lambda-1)\mathbf{x}(n)\mathbf{x}^T(n)$$

$$\sigma_w(n) = \lambda\sigma_w(n-1) + (\lambda-1)w^2(n).$$

We may observe that it is unnecessary to take the square root. We may simply use θ^2 as our statistic, and T^2 as our threshold. We may also observe that the $\mathbf{C_{xx}}$ will not change dramatically at each iteration, and that it is possible to use an updating method for $\mathbf{C}_{xx}^{-1}$ (Salmen, Schlipsing & Igel 2010).

2.10 Chapter Summary

This chapter contains a wide ranging summary of the field of echo cancellation. We have examined the core technology of echo cancellation, the LMS, RLS, AP and Frequency domain algorithms, reviewed the state of the art and analysed the most popular LMS algorithm in depth.

The material in this chapter provides the foundation for the chapters to come. Chapter 3, *Echo Cancellation Simulation*, provides an environment for the exploration of algorithms described here and for new algorithms. Chapter 4, *Framework for Echo Cancellation with Non Linearities* builds on this foundation and creates a paradigm

for the creation of echo cancellation algorithms that cope with non-linearities. Channel estimation lies at the core of echo cancellation technology, and concepts from this chapter underpin the work of Chapter 6, *Optimisation of Test Signals*.

Chapter 3

Echo Cancellation Simulation

In order to facilitate the easy exploration and development of echo cancellation algorithms, an echo cancellation simulation environment has been developed. It has been named the Echo Cancellation Object Oriented Simulation Environment (ECOOSE). The design criteria for ECOOSE collectively state that the end product shall be able to model any echo cancellation scenario using any echo cancellation algorithms, and may be summed up as *flexibility*. The required flexibility for ECOOSE was not achieved in the first, or even the second attempt, and it is important to learn from these mistakes and to understand why these attempts were unsatisfactory. This chapter provides the requirements of the project, explains why the first and the second attempts failed, and presents some sample results.

An object-oriented framework is the reusable design and implementation of a system or subsystem (Beck & Johnson 1994). It is typically implemented as a set of abstract classes which define the core functionality of the framework along with concrete classes for specific applications included for completeness (Froehlich, Hoover, Liu & Sorenson 1997). ECOOSE is an object oriented framework for experimenting with theoretical algorithms. This chapter documents the development a practical software project, and is presented in the past tense.

3.1 Requirements

ECOOSE must be a suitable environment for the development and evaluation of echo cancellation algorithms.

ECOOSE needs to be able to work with the following types of channels:-

- Static Linear Channels
- Channels with a timewise discontinuity.
- Slowly changing linear channels.
- Non linear channels.

ECOOSE needs to be able to handle the following types of algorithms:-

- Algorithms that are enhancements of the LMS algorithm.
- Algorithms that are enhancements of the RLS algorithm.
- Algorithms that are enhancements of the AP algorithm.
- Block based time domain algorithms.
- Block based frequency domain algorithms.
- Non linear algorithms.

ECOOSE should be able to generate the following types of output.

- Graphical depiction of algorithm convergence including the euclidean difference between the model and the known solution.
- Numerical indicators of obtained convergence.
- File output of obtained channel.

ECOOSE should be able to take any local or remote sound in the form a waveform audio input.

3.2 Structured Design implemented in Matlab 2007

Matlab is an attractive platform for the implementation of the algorithms because it is a high level language and makes a wide range of medium level mathematical functionality immediately available. Matlab 2007 was available at the commencement of the project. Matlab 2007 has only very limited object oriented capabilities. No references to Matlab 2007 objects are possible, and Matlab 2007 objects are not mutable. Matlab 2007 permits only objects that need to be recreated each time a value in the object is modified. These limitations presented a major impediment to object oriented programming. Thus, a design for implementation in Matlab 2007 had to be a structured one.

Interfaces for the various elements of a simulation were defined, so that they could be interchanged as needed. This earlier toolbox had the following elements as well as code that allowed the elements to be selected and run:-

- A directory of sound files in waveform audio format.
- A directory of linear channels in waveform audio format
- A directory of echo cancellation algorithms in the form of Matlab functions,
- A directory of experiment specifications or run files in the form of small Matlab functions. Each file performed one experiment or run.

This first implementation was unsatisfactory because it was not flexible enough to meet the primary goal of the project. Any experiment could be carried out, but it was necessary to meddle with the code for algorithms and other elements in order to do so. Examples of the way the first implementation lacked flexibility are as follows:

- It was difficult to separate the algorithm from the channel. Ideally, the algorithm should not already know the channel, but should have to discover the channel for itself. But progress, typically in the form of euclidean distances, needed to be monitored at each iteration, and that required knowledge of the channel. If the algorithm state variables were not to be passed out of the algorithm in

order to save the values between calls, it was necessary for the algorithm to be provided with the information about the channel and do those calculations with each iteration.

Because the algorithm and the channel were not separated, it was difficult to try out random combinations of channel and algorithm, e.g. trying a non linear channel with a linear algorithm, or a linear channel with a non linear algorithm.

- Each new algorithm seemed to impose new demands on the algorithm interface. The algorithm interface had to be expanded at each turn. In order to allow interchangeability, the changes in the interface had to be propagated back to the existing algorithms. The only alternative to this would be to have a huge, all encompassing interface with every need anticipated.

- Different algorithms required different parameters, and in order to cater for this, it was necessary to format arguments into a string and implement parsing. In practice, it was easier to change the parameters in the code itself.

- Algorithms had to be modified in a major way to model a change in the parameters of a channel during a run.

3.3 Object Oriented Programming implemented in Matlab 2008

Implementation in a 3rd generation object oriented language such as C++ was considered. Using C++ would allow us to solve all of the above problems, but we would lose the easy access to the well developed well tested mathematical primitives offered by Matlab. Finding or developing the required primitives would take time, and testing them would also take time.

Another possibility was the newly released Matlab 2008. The object oriented capabilities available in Matlab 2008 are adequate for object-oriented development (McGarrity 2008).

In Matlab 2008 it is possible to have objects that can be referenced from elsewhere in a

program and to have methods that alter only some of the properties of an object(Mathworks 2008*b*). Properties and methods can be private, protected or public, as one might expect in an object oriented language. Constructors, destructors, abstract classes, static methods and even multiple inheritance are all part of the language(Mathworks 2008*b*). Objects that are no longer referenced are automatically destroyed, which seems appropriate to a high level application-oriented language such as Matlab (although currently, memory space associated with destroyed objects is not re-used, unless one manually destroys *all* variables (Foti 2008)).

We opted to upgrade to Matlab 2008 and recreate the echo simulation environment again from scratch.

3.4 The First Object Oriented design of ECOOSE

3.4.1 Design

Figure 3.1 is a class diagram from the first object oriented design of ECOOSE, which shows the relationships between the various classes.

3.4.2 The Algorithm Classes

The algorithm classes are derived from the base algorithm class, algBase, and the subclasses specialize the algorithm. Thus the algorithm is represented by an hierarchy of classes. Note that an algorithm class gets to know the same information that a real echo cancellation algorithm would. The algorithm classes do not know the channel. It is the job of the algorithm to discover the channel. An important abstract method of the base class is called next(), which is called for each iteration and has as its arguments, the current value of x and the current value of w. The algorithm classes make their results available by means of methods. One of the abstract methods of the base algorithm class is getY(), which returns the current estimated result from the echo channel y.

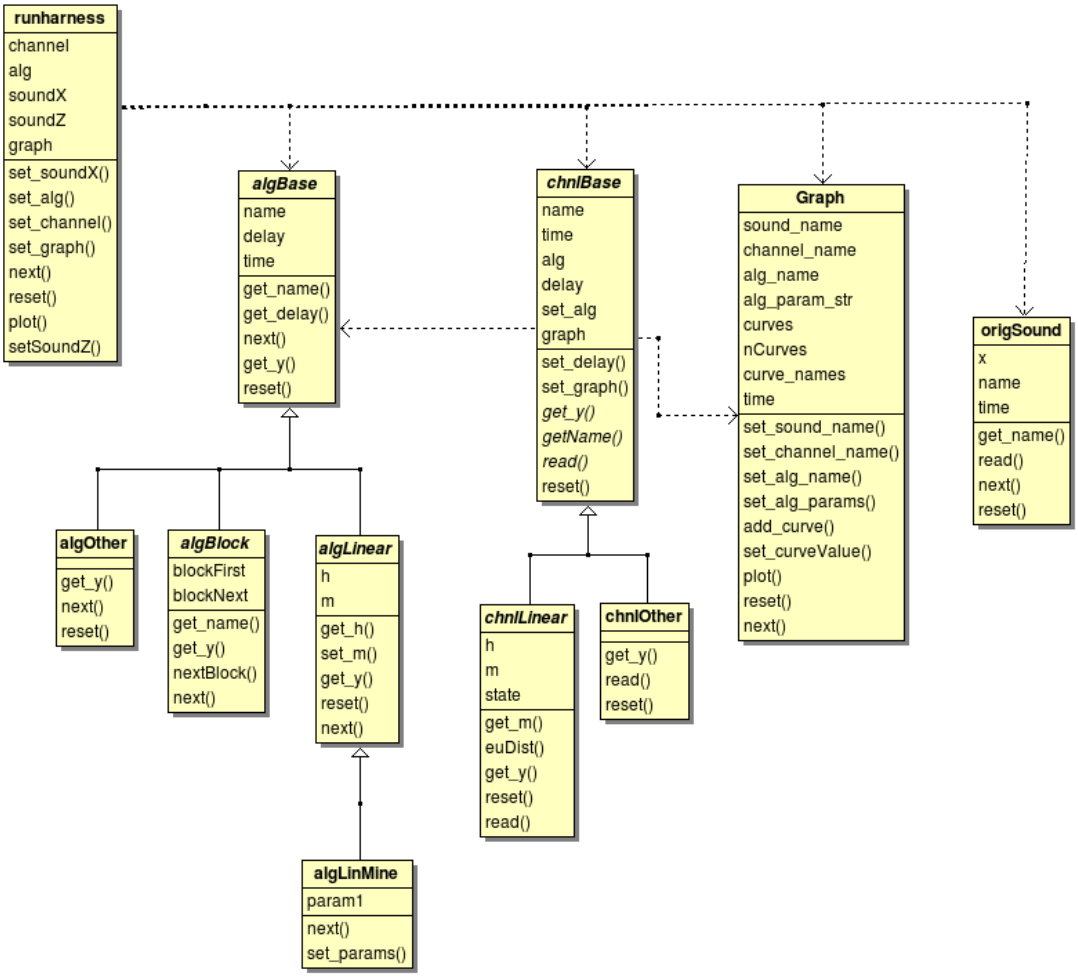

Figure 3.1: Class Diagram from the first object oriented design of ECOOSE. Created using the freeware UML toolbox BOUML (Pages 2010)

3.4.3 The Linear Algorithm Class

The linear algorithm class is a specialization of the base algorithm class. Most cancellation algorithms are linear, and are derived from this class. Since linear channels are characterised by their impulse response, the class has a method getH() which returns the impulse response, and a method setH() which allows the current value of the impulse response to be set artificially.

3.4.4 The Channel Classes

The channel class knows the characteristics of the channel, and knows how to calculate the real value of y from the values of x. The channel class calculates $w = y + z$ and provides the algorithm with x and w. The channel class uses a method provided by the algorithm to obtain the estimated value of y, and in turn has a method for report the error to output classes such as the graphing class.

3.4.5 Algorithm Delay

Block based echo cancellation algorithms, such as Frequency domain algorithms require information to be calculated as a block, and consequently impose a blocking delay. The algorithm knows how long this delay will be and has a method which returns the delay measured in discrete time periods to the caller. At each iteration, a request to the algorithm for the y value will return the estimated value of for y at that many time periods ago. The channel, having obtained the delay from the algorithm performs its comparisons with the real value of y that many time periods ago.

3.4.6 Flaws in the First Object Oriented Design

Ideally the channel class should be responsible only for the representing the channel. In the first design, however, the channel class was not only responsible for calculating the real channel response to x, it was also used to calculate and plot the differences

between the real channel characteristics and the estimated characteristics over time. This was done because the channel class is well placed to know what can be graphed, and well placed to calculate the difference between the real channel characteristics and the estimated channel characteristics, so it was convenient to have the channel class perform both tasks.

Unfortunately it became apparent that the advantages of this approach are only of small benefit, and the cost of including functions which are not really part of a channel in the channel class was too serious to be tolerated. For example, temporary code modifications would have been required in the algorithm class itself in order to create the graphs shown in Figure 4.1 for the purpose of exploring a non linear echo cancellation algorithm.

The manner in which linear and non-linear channels interface with linear and non-linear algorithms poses a challenge in object oriented design. Naturally, linear channels can interface with linear algorithms and non linear channels can interface with the non linear algorithms. The linear channel can interface with the non linear algorithm class without difficulty. The methods which are exclusive to the nonlinear algorithm class simply won't be called and that is not a problem. However, the non-linear channel cannot interface with the linear algorithm, without a workaround, because it calls methods of the non-linear algorithm that the linear algorithm does not implement.

3.5 The Second Object Oriented design of ECOOSE

In order to address the flaws described above it was necessary to take the functionality that the channel class hierarchy had in the first object oriented design, and split it into two class hierarchies as shown in Figure 3.2. One of them is a hierarchy of channel calculator classes whose purpose is to calculate y, given x. The other is a hierarchy of algorithm observer classes which are responsible for obtaining the estimated echo characteristics from the algorithm class at each time step, comparing them with known echo characteristics, and prepare, at a high level, final results such as graphs.

This algorithm observer class needs to have the same channel model that the algorithm

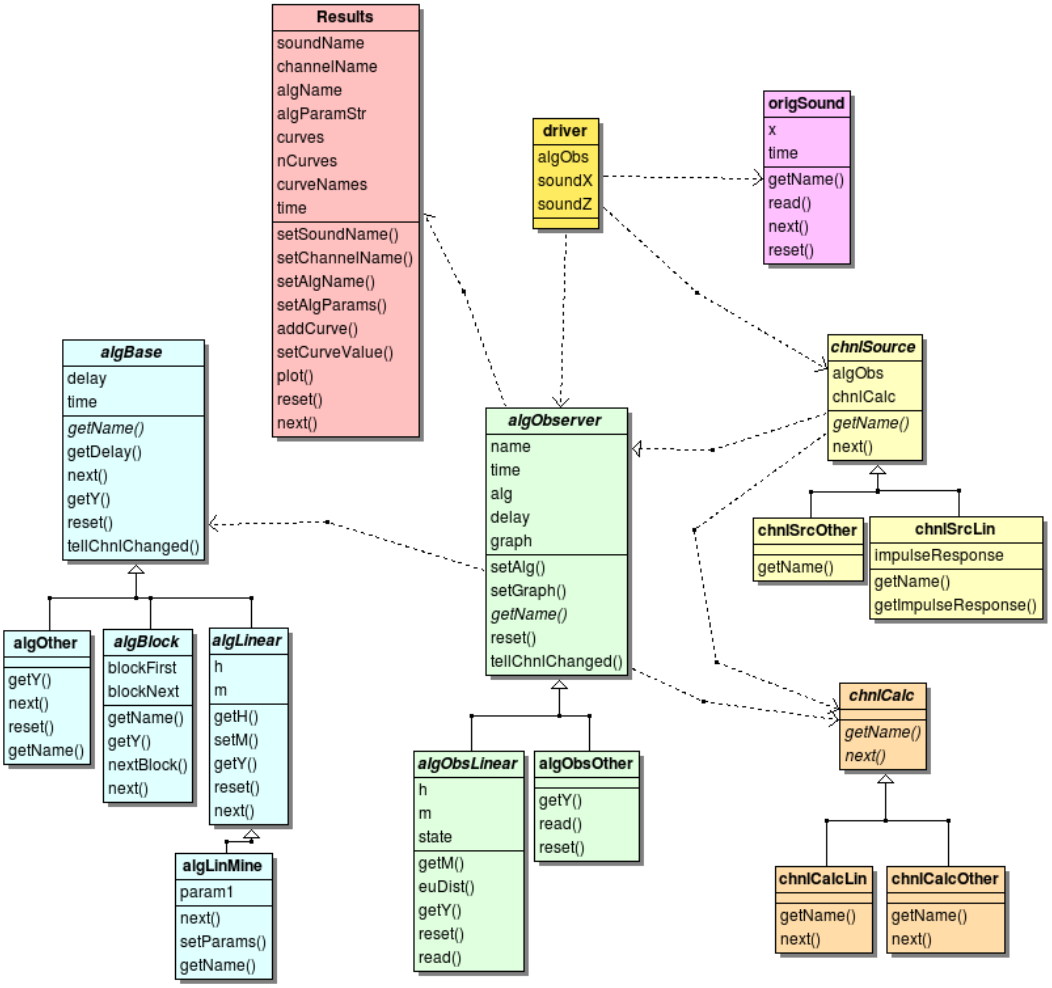

Figure 3.2: Class Diagram for the improved object oriented design for ECOOSE. Created using the freeware UML toolbox BOUML (Pages 2010)

class has and needs to know the values in that model from both the channel calculator class and the algorithm class, so that it may perform comparisons. The base observer class is able to observe all algorithm classes because it only observes the estimated $\hat{y}$, and compares it to the real y as calculated by the channel calculator class.

The linear algorithm observer class observes the estimated impulse response, $\hat{h}$ as well as estimated result, $\hat{y}$. It is able to observe all algorithms where the characterisations include the impulse response h. Note that the channel itself may be linear or non linear, now that the channel calculator class has been separated from the algorithm observer class.

As mentioned, the role of the channel calculator class is to apply the channel to x giving y. Another class is needed to track changes in the dynamic channels and to inform both the observer class and the channel of the current state of the channel whenever the channel changes. From the requirements, ECOOSE must be able to handle a channel that changes slowly, or is subject to sudden change during the course of the simulation. It is therefore highly desirable to have the true source of the channel characteristics in a single class. Failure to do this could easily result in wrong results, because the algorithm observer's understanding of the channel characteristics could easily diverge from those of the channel class. This class is called the channel source class.

Having the channel source class call methods in the algorithm observer class and the channel calculator class would not not be a flexible arrangement, because then the channel source class would have to make exactly the right calls, i.e. it would have to be matched to both the algorithm observer class and the channel calculator class, and it would end up being a channel source hierarchy of classes rather than just a standalone class. The solution given by the observer design pattern (Gamma, Helm, Johnson & Vlissides 1994) was the one implemented. The algorithm observer class and the channel calculator class subscribe to events generated by the channel source class. When the channel changes, the channel source class issues an event to all subscribers, and whenever the subscribers receive the event they call a method of the channel source class to receive the updated channel data.

3.6 Chapter Summary

The development, analysis, and comparison of echo cancellation algorithms has reached a stage where it is essential that independent and objective tests of algorithms need to be carried out in an agreed manner.

The creation of an environment for the simulation of echo cancellation proved difficult to achieve using procedural technology. It proved impractical to separate the algorithm from the channel. Accommodating the extra parameters that each algorithm may have was hard work. A dynamically changing channel would have been a programming challenge. The adoption of object oriented technology solved these problems, and also give us flexibility without the penalty of having to re-implement all the features of an audio channel or of an algorithm if most of its features are similar to one already implemented.

In developing an OO framework, as discussed in (Johnson & Foote 1988), we need to think carefully about its design so that a wide range of applications can readily be accommodated. We have explained some of the difficulties, described our designs, for our object oriented framework which we feel can adapt to the present and future needs of designers of echo cancellation algorithms.

Chapter 4

Framework for Echo Cancellation with Non Linearities

This chapter explains how to exploit the *Method of Innovations* (Kailath 1981) to perform channel identification for a channel with a non linearity using a Bayesian approach. We refer to the algorithm developed here as the Hammerstein LMS (HLMS) algorithm.

4.1 Our Problem

The non-linearity that we model is one that might be introduced by a speaker driven to its maximum amplitude, i.e. clipping, to an otherwise linear channel. This is modelled as a Hammerstein system (Belforte & Gay 1999), i.e. the combination of a memoryless non-linear function of the input and a linear system as shown in Figure 1.3. The method developed in this chapter can be applied to different nonlinear channels with form similar to this.

The non-linear function will be expressed as a polynomial so that the task of identifying the non linear function essentially became the task of determining the co-efficients of that polynomial. If we use a polynomial in its simplest form, however, the co-efficients

would not be independent. In order to improve convergence, we express the non-linearity in terms of orthogonal polynomials. We choose to use a flat orthogonality weighting function (i.e. no weighting function) and orthogonality over the segment -1 to 1, which implies the use of the Legendre polynomials (Hilbert & Courant 1924).

Let the non-linear function be denoted by $\Phi(x)$, where

$$\Phi(x) = \sum_{j=0}^{m} \alpha_j \phi_j(x)$$

where each ϕ_j denotes the Legendre polynomial of order j, and α_j denotes the corresponding coefficient that needs to be determined. The Legendre polynomials are orthogonal, and as part of our Bayesian approach, we assume that $\alpha = (\alpha_0, ..., \alpha_m)$ are a collection of independent identically distributed (IID) variables.

Let the filter tap coefficients of the linear system be denoted by h_j, in the usual manner. We also adopt, as another aspect of the Bayesian approach to model estimation, the assumption that the coefficients of the impulse response are IID. Although it is essential to adopt a priori assumptions regarding the distribution of all estimated parameters in the Bayesian approach, these assumptions are not expected to strongly influence the resulting estimations. Thus the coefficients that will be estimated from the observed data are $(\alpha_0, ..., \alpha_m; h_0, ..., h_{n-1})$.

4.2 The Method of Innovations

In this section we describe the Method of Innovations and provide a proof for it.

Description In this section we wish to show how to update an estimate as additional information arrives. The arriving information will come from a sequence of random scalar or vector variables, $v_1, v_2, ...$.

The vector or scalar u is the quantity we wish to estimate, based on the set $v_1, ..., v_k$. The estimates of u, based on the set $v_1, ..., v_k$. will be denoted by $\hat{u}_{|k}$. The estimate of v_k from $v_1, ..., v_{k-1}$ will be denoted by $\hat{v}_{k|k-1}$.

We may make a linear estimate $\hat{v}_k$ of v_k from $(v_1, ..., v_{k-1})$ in the sense that $\hat{v}_k$ is some linear combination of $(v_1, ..., v_{k-1})$. That is, in the case where v is scalar:

$$\hat{v}_{k|k-1} = (v_1, ..., v_{k-1})a,$$

or in the case where v is a vector

$$\hat{v}_{k|k-1} = a(v_1, ..., v_{k-1})^*,$$

where a is a constant vector or matrix and $\hat{v}$ is a linear estimate in the least squares sense. That is, a is characterized in the following way

$$E[(\hat{v}_{k|k-1} - v_k)^* v_j] = 0 \qquad j = 0, ..., k-1.$$

Our objective here is to be able to predict a fixed variable u from our observations of v. We assume that u is a linear combination of past values of v. We already have a prediction $\hat{u}_{|k-1}$ of u, which is based on $(v_1, ..., v_{k-1})$. Now that we have new data in the form of v_k, we wish to make a new prediction $\hat{u}_{|k}$.

Let us define the *innovation* ϵ_k by:

$$\epsilon_k = v_k - \hat{v}_{k|k-1}. \tag{4.1}$$

Now we come to the key equation of the innovations method, which we first prove then later we apply it to the case of a nonlinear channel. The kth estimate of u is given by:

$$\hat{u}_{|k} \;=\; \hat{u}_{|k-1} \;+\; E[u\epsilon_k^*]\;\; (E[\epsilon_k \epsilon_k^*])^{-1}\;\; \epsilon_k. \tag{4.2}$$

If one notices that the term on the right hand side is in fact the Linear Minimum Mean Square Estimator (LMMSE) formula (A.13) for estimation of a variable from an observed variable, then the above formula may be intuitively perceived as the previous estimation of u plus the contribution from the latest innovation.

Proof: The following proof is based on that presented in (Kailath 1981) and relies on the orthogonality principle (Luenberger 1969)(chapter 3). The orthogonality used in this proof is one between stochastic variables. If we let the space of stochastic variables be a Hilbert space by defining the inner product of u and v to be

$$u \cdot v = E[uv],$$

then we can say that u and v are orthogonal if and only if $E[uv] = 0$.

Let the kth error of estimation of u be represented by $\ddot{u}_{|k}$:

$$\ddot{u}_{|k} = u - \hat{u}_{|k}. \tag{4.3}$$

Then by the orthogonality principle, the minimum least squares estimate of u lies where $\ddot{u}_{|k}$ is orthogonal to our observed instances.

$$\ddot{u}_{|k} \perp (v_1, ..., v_k). \tag{4.4}$$

Hence, we need to show that $\ddot{u}_{|k}$ is orthogonal to $(v_1, ..., v_k)$, and we will do this by induction.

Clearly when $k = 0$, $\hat{u}_{|k} = 0$, then by definition the error is orthogonal to the empty set and orthogonality holds for $k = 0$.

Next, we need to show that if our orthogonality holds for $k - 1$ then it must also hold for k. Our *inductive hypothesis* is:

$$\ddot{u}_{|k-1} \perp (v_1, ..., v_{k-1}). \tag{4.5}$$

By expansion using (4.1):

$$\begin{aligned} E[\ddot{u}_{|k-1} v_k^*] &= E[\ddot{u}_{|k-1}(\epsilon_k + \hat{v}_{k|k-1})^*] \\ &= E[\ddot{u}_{|k-1}\epsilon_k^*] + E[\ddot{u}_{|k-1}\hat{v}_{k|k-1}^*]. \end{aligned} \tag{4.6}$$

Since $\hat{v}_{k|k-1}$ is a linear combination of $(v_1, ..., v_{k-1})$, by our inductive hypothesis,

$$= E[\ddot{u}_{|k-1}\epsilon_k^*],$$

and by expansion using (4.3),

$$= E[u\epsilon_k^*] - E[\hat{u}_{|k-1}\epsilon_k^*],$$

and by the orthogonality principle,

$$= E[u\epsilon_k^*]. \tag{4.7}$$

Now by expansion using (4.1), we get:

$$E[\epsilon_k v_k^*] = E[\epsilon_k \epsilon_k^*] + E[\epsilon_k \hat{v}_{k|k-1}],$$

and since our innovation is defined to be orthogonal to the previous estimate,

$$= E[\epsilon_k \epsilon_k^*]. \tag{4.8}$$

Taking our definition of $\ddot{u}$,

$$\ddot{u}_{|k} = u - \hat{u}_{|k},$$

and substituting (4.2) into the above,

$$\begin{aligned} &= u - \hat{u}_{|k-1} - E[u\epsilon_k^*] \ (E[\epsilon_k \epsilon_k^*])^{-1} \ \epsilon_k \\ &= \ddot{u}_{|k-1} - E[u\epsilon_k^*] \ (E[\epsilon_k \epsilon_k^*])^{-1} \ \epsilon_k. \end{aligned} \tag{4.9}$$

The first term of (4.9) is orthogonal to $(v_1, ..., v_{k-1})$ by our inductive hypothesis. The second term, a multiple of ϵ_k is orthogonal to $(v_1, ..., v_{k-1})$. Therefore $\ddot{u}_{|k}$ is orthogonal to $(v_1, ..., v_{k-1})$.

By applying the expectation $E[\cdot \times v_k^*]$ of both sides of (4.9) and expanding, we get

$$E[\ddot{u}_{|k} v_k^*] = E[\ddot{u}_{|k-1} v_k^*] - E[u\epsilon_k^*] \ (E[\epsilon_k \epsilon_k^*])^{-1} \ E[\epsilon_k v_k^*],$$

and substitution of (4.7) and (4.8) into the above gives us,

$$\begin{aligned} &= E[u\epsilon_k^*] - E[u\epsilon_k^*] \ (E[\epsilon_k \epsilon_k^*])^{-1} \ (E[\epsilon_k \epsilon_k^*]) \\ &= E[u\epsilon_k^*] - E[u\epsilon_k^*] \\ &= 0. \end{aligned}$$

Hence, $\ddot{u}_{|k}$ is orthogonal to v_k. Since $\ddot{u}_{|k}$ is orthogonal to $(v_1, ..., v_{k-1})$ and also to v_k, then $\ddot{u}_{|k}$ is orthogonal to $(v_1, ..., v_k)$, and thus we have shown that if our inductive hypothesis holds for $k-1$ then it must also hold for k.

4.3 Applying the Method of Innovations to Echo Cancellation

We now need to apply the method to the problem described in Section 4.1. We will use the variable names documented in Section 1.1 and depicted in Figure 1.1.

The stochastic variable that we may observe is w. The least squares best estimate of y based on $(w_1, ...w_{k-1})$ is denoted by $\hat{y}$. In place of ϵ_k, we use $\hat{z}_k$. Under conditions of no doubletalk, $z(t) = 0$ and $\hat{z} = y - \hat{y}_{k|k-1}$, so $\hat{z}$ is a perfect fit. Under conditions of doubletalk, we assume that $z(t)$ is uncorrelated with $h(t)$ and with $x(t)$, so it will still be valid to use $\hat{z}_k$, even though we expect doubletalk to perturb the solution as described in Subsection 2.3.5. The variables that we wish to estimate, $(\alpha_0, ..., \alpha_m; h_0, ..., h_{n-1})$ will be used in place of u. Application of (4.2) into this context gives us:

$$\hat{h}_{j|k} = \hat{h}_{j|k-1} + E[h_j, w_k]\ E[\hat{z}_k^2]^{-1} \hat{z}_k, \tag{4.10}$$

$$\hat{\alpha}_{j|k} = \hat{\alpha}_{j|k-1} + E[\alpha_j, w_k]\ E[\hat{z}_k^2]^{-1}\ \hat{z}_k. \tag{4.11}$$

Taking sub-expressions of (4.10) and (4.11) one at a time,

$$\begin{aligned} E[h_j, w_k] &= E[h_j, y_k] \\ &= E[h_j \sum_{i=1}^{n} h_i \Phi(x_{k-i})], \end{aligned}$$

recalling that we are assuming the elements of $\mathbf{h}$ to be IID,

$$\begin{aligned} &= E[h_j h_j \Phi(x_{k-j})] \\ &= \sigma^2_{h_j|k-1} \Phi(x_{k-j}). \end{aligned} \tag{4.12}$$

Also

$$\begin{aligned} E[\alpha_j, w_k] &= E[\alpha_j, y_k] \\ &= E[\alpha_j \sum_{i=1}^{n} h_i \Phi(x_{k-i})], \\ &= E[\alpha_j \sum_{i=1}^{n} h_i \sum_{s=1}^{m} \alpha_s \phi_s(x_{k-i})], \end{aligned}$$

recalling that we are assuming the elements of α to be IID,

$$= E[\alpha_j \sum_{i=1}^{n} h_i \alpha_j \phi_j(x_{k-i})]$$
$$= \sigma^2_{\alpha_j|k-1} \sum_{i=1}^{n} h_i \phi_j(x_{k-i}). \quad (4.13)$$

As the algorithms converges, $\hat{z}$ will become an accurate expression for z. Hence σ_z^2 would soon become an adequate expression for $E[\hat{z}_k^2]$ as the echo cancellation begins to converge. Substituting (4.12) into (4.10) gives:

$$\hat{h}_{j|k} = \hat{h}_{j|k-1} + \frac{\hat{\Phi}_{|k-1}(x_{k-j})\sigma^2_{h_{j|k-1}}}{\sigma_z^2}\ \hat{z}_k, \quad (4.14)$$

and substituting (4.13) into (4.11) gives:

$$\hat{\alpha}_{j|k} = \hat{\alpha}_{j|k-1} + \frac{\left(\sum_{i=0}^{n} \hat{h}_{i|k-1}\phi_j(x_{k-i})\right)\sigma^2_{\hat{\alpha}(j|k-1)}}{\sigma_z^2}\hat{z}_k. \quad (4.15)$$

Note that the derived algorithm is more like PNLMS than like NLMS, because each coefficient is given its own rate of convergence, $\sigma^2_{h_{j|k-1}}$ and $\sigma^2_{\hat{\alpha}(j|k-1)}$.

4.4 Implementation

We have applied the method of innovations and it has produced two equations for the updating of our model coefficients. In this section we describe the simplified algorithm that was implemented and tested. For simplicity, we used single constants in place of separate values of $\sigma^2_{h_{j|k-1}}$ and $\sigma^2_{\hat{\alpha}(j|k-1)}$, so that the actual implemented algorithm was more like NLMS than like PNLMS.

In the expression for determining the filter tap coefficients, (4.14), we replaced $\frac{\sigma^2_{h_{j|k-1}}}{\sigma_z^2}$ with the constant μ_1, giving

$$\hat{h}_{j|k} = \hat{h}_{j|k-1} + \mu_1 \hat{\Phi}_{|k-1}(x_{k-j})\ \hat{z}_k. \quad (4.16)$$

An appropriate value for μ_1 was determined experimentally. Notice that if Φ was replaced by a linear function, then this is the usual expression for the LMS algorithm.

In the expression for determining the coefficients of the non-linear function (4.15), we

replaced $\frac{\sigma^2_{\hat{\alpha}(j|k-1)}}{\sigma^2_{\hat{z}}}$ with the constant μ_2, giving

$$\hat{\alpha}_{j|k} = \hat{\alpha}_{j|k-1} + \mu_2 \left(\sum_{i=0}^{n} \hat{h}_{i|k-1} \phi_j(x_{k-i}) \right) \hat{z}_k. \tag{4.17}$$

An appropriate value for μ_2 was also determined experimentally.

At each iteration, (4.16) is used to update each of the n filter tap coefficients, and (4.17) is used to update each of the Legendre polynomial coefficients.

4.5 Results

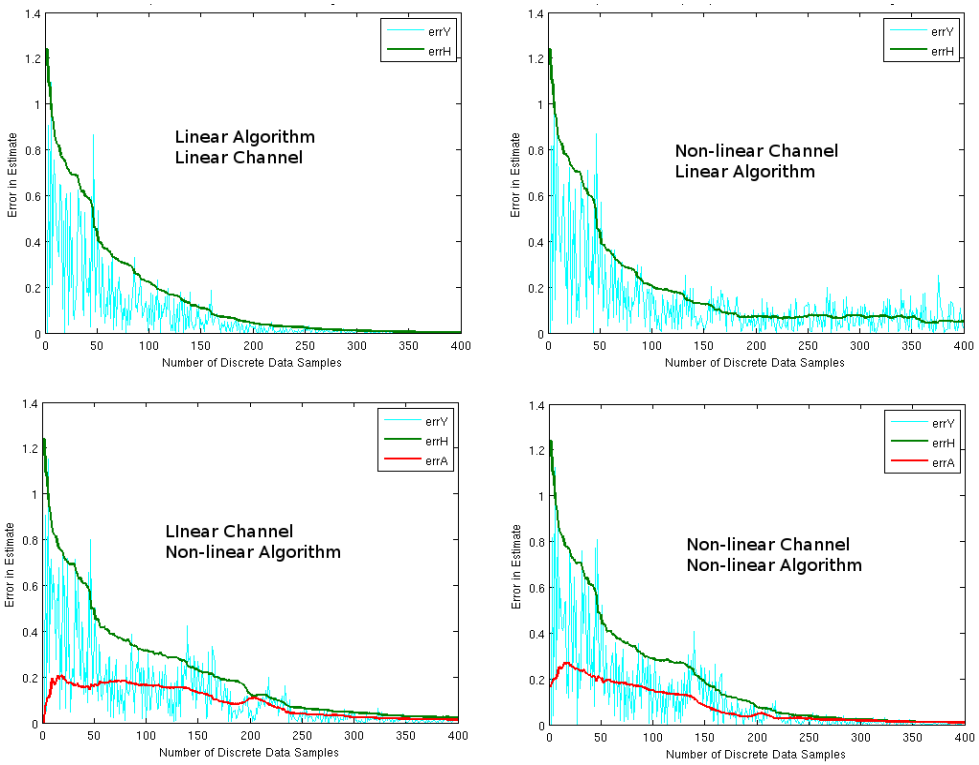

Figure 4.1: Graphs showing the Euclidean norm of the errors of the estimated coefficient in echo cancellation algorithms as the algorithms adjust over time. Created in ECOOSE.

HLMS was implemented in ECOOSE, and the results are summarised in the 4 graphs of Figure 4.1. The left hand graphs all featured linear channels. The right hand graphs all feature non-linear channels. The graphs in the top row were created using NLMS echo cancellation. The graphs in the bottom row were created the HLMS algorithm. The noisy light blue curve shows the absolute of the error in y. The dark green curve

shows the Euclidean norm of the errors in the linear tap coefficients. The red curve shows the euclidean norm of the errors in the non-linear coefficients.

For the sake of comparisons, the NLMS algorithm was applied to a linear channel. The top left graph in Figure 4.1 shows convergence of the NLMS algorithm applied to a linear channel. In it, you can see that the absolute error in $\hat{\mathbf{h}}$ and the absolute error in $\hat{y}$, and hence the echoes, converge rapidly to zero.

The top right hand graph of Figure 4.1 shows what happens if the response of channel is non linear. The response used in this experiment models "clipping" of the sort that might be exhibited by a speaker driven to its limit. Initially the NLMS makes progress, but ultimately it is unable to converge. Clearly linear echo cancellation is far from useless, but it would still be unsatisfactory in most cases.

The bottom left hand graph of Figure 4.1 shows the performance of a system when the echoes in a linear channel are being cancelled by the HLMS algorithm. Convergence to a satisfactory estimate of the channel is similar to that for the NLMS algorithm.

The bottom right hand graph of Figure 4.1 shows the performance when the echoes pass through a nonlinear channel and the echo canceller is the HLMS algorithm. The errors in both $\hat{\mathbf{h}}$ and $\hat{y}$ converge to zero, and the echoes are effectively suppressed.

The code used to simulate the HLMS algorithm is available as part of the ECOOSE code. Details on how to run the code and to generate the graphs shown in Figure 4.1 are given in Subsection B.1.3.

4.6 Chapter Summary

The method of innovations has been explained, and used to provide an adaptive NLMS-like algorithm for the estimation of a non linear Hammerstein channel. The algorithm has been implemented in ECOOSE and the results confirm the viability of the derived algorithm and the viability of applying the method of innovations to the problem of echo cancellation.

Chapter 5

Lessons From the Wireless World

In echo cancellation and in certain wireless technologies, we need to estimate linear channels. In echo cancellation, we estimate the channel in order to predict what comes out of the channel and cancel it from received input. In ODFM, GSM and other personal communication networks, we estimate the channel in order to be able to use a more efficient modulation technique. Because of the commonality, some of the ideas and concepts are from one technology are are applicable in the other technology. Despite the commonality, the differences are many in practice, and the ideas and concepts end up being applied in different ways.

Currently in the echo cancellation literature, there is very little on the optimality of test signals or the use of secondary statistics to improve channel estimation. We may find such publications on such topics in the literature relating to successful wireless technology. Hence, this chapter of the dissertation will focus on those topics in the wireless literature in order to gain ideas, understanding, and inspiration from research which was developed for wireless channel sounding and for OFDM communications.

5.1 Statistical Estimation of OFDM Wireless Channels

Orthogonal Frequency Division Multiplexing (OFDM) is a technology that is rapidly being adopted for communications. Its uses are many, but they include our mobile

phone communications, our military communications, ASDL, and more. The channel estimation techniques being developed for OFDM are of particular interest. In this section we shall examine some statistical techniques used for channel estimation in OFDM. It is necessary, however, to understand the basics of OFDM and of wireless communication in general in order to be able to understand the statistical techniques for estimating the channel, and we shall therefore focus on the background before moving to more specific details.

5.1.1 Background

In this section we will endeavour to understand the context in which ODFM operates. In order to properly appreciate the solutions offered by ODFM technology, it is necessary to understand the problems.

Although we refer to OFDM as if it was a single technology, we might also view it as a collection of modern technologies that work together to produce a superior result. These technologies include:

- The use of multiple tightly packed frequency bands, permitting the use of long symbols which better cope with multipath propagation.
- The use of orthogonal overlapping occupied bandwidths which allow the frequency bands to be packed much tighter.
- The simultaneous use of phase and amplitude modulation, which allow more data to be transmitted with each symbol.
- The use of the Fast Fourier Transform which allows efficient, cheap and miniaturised transmitters and receivers.
- Advanced statistical methods of improving the channel estimation.

Occupied Bandwidth

One way of ensuring good communication with a very high data bandwidth between two points would be use a very high frequency bandwidth. However, in general it is not possible to use an arbitrarily wide frequency bandwidth, because that would interfere with the communications of other people. The radio frequency spectrum, which extends from approximately 30Hz to 300Ghz is a very valuable resource, and there are enormous demands for allocations of the spectrum. The radio frequency spectrum is managed by the International Telecommunications Union (ITU) (*Radiocommunication Sector* 2010), which is an agency of the United Nations (UN). Economic and equitable usage requires efficient use of allocated radio spectrum.

An allocated bandwidth may be specified by a centre frequency as well as an occupied bandwidth. Typically the occupied bandwidth used in practice is one where 98% of the power lies within the allocated boundaries.

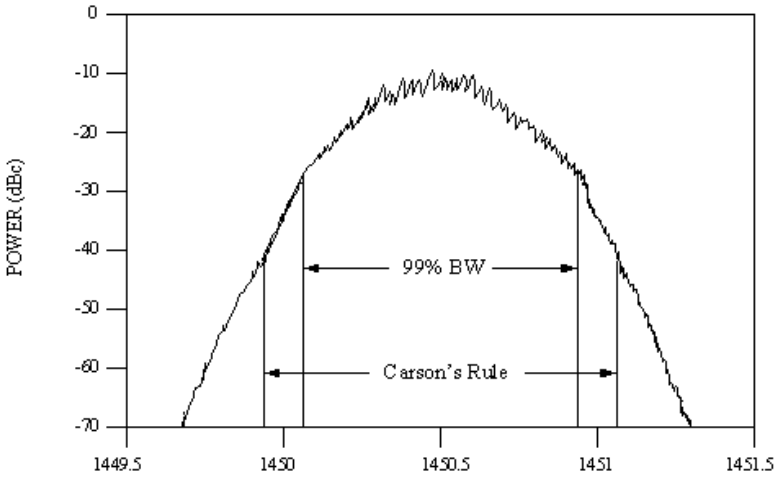

Figure 5.1: Illustration depicting spectrum which results when a signal having 6 dB/octave modulates an FM transmitter. Adapted from (*RCC IRIG 106-99: Telemetry Standards (Appendix A)* 2003).

While a carrier frequency with no modulation might theoretically occupy an infinitesimal frequency bandwidth, when data is transmitted on a carrier, the carrier must necessarily occupy a non zero bandwidth. The occupied bandwidth is a function of the frequency of the data being encoded onto the carrier frequency. Figure 5.1 shows the frequency spread of a FM carrier when modulated with a sound. The energy is greatest

near the carrier frequency, and becomes less, but never zero, as we move away from the carrier frequency.

Carson's rule is an approximation that predicts the occupied bandwidth for sinusoidal modulation. Carson's rule for frequency modulation is:

$$B = 2(f_m + f_d),$$

where B is the occupied bandwidth, f_m is the maximum modulation frequency, and f_d is the maximum frequency deviation used in the frequency modulation (Anuff & Liou 1971). Carson's rule for phase modulation is:

$$B = 2f_m(\Delta\theta + 1),$$

where $\Delta\theta$ is the peak phase deviation (Proakis & Salehi 1994). In general, for phase modulation, the occupied bandwidth is proportional to the symbol rate. Hence, for a given carrier frequency allocation, we could have n carrier bands, each carrying $2f_m$ symbols per second, or we could have $2n$ carrier bands, but they could each only carry f_n symbols per second. Either way, the total number of symbols per second and the total occupied bandwidth would remain relatively unchanged.

Multipath Propagation

Sometimes there is line of site (LOS) propagation between the transmitter and the receiver. A good example of this would be microwave links between cell phone stations using parabolic dishes. But in most practical wireless situations, the radio propagation travels to the receiver indirectly by many routes as depicted in Figure 5.2. This is termed "multipath propagation" (Parsons & Bajwa 1982).

Loss of signal under these circumstances due to signal cancellation caused by paths arriving out of phase can be classified as *Rician* fading, if there is a line of sight or dominant signal path, or *Rayleigh* fading, when there is no line of sight or dominant path.

Impulse Response Because it takes a while for signal to reach its destination, impulse responses measured from the time of transmission would have a period of all zeros.

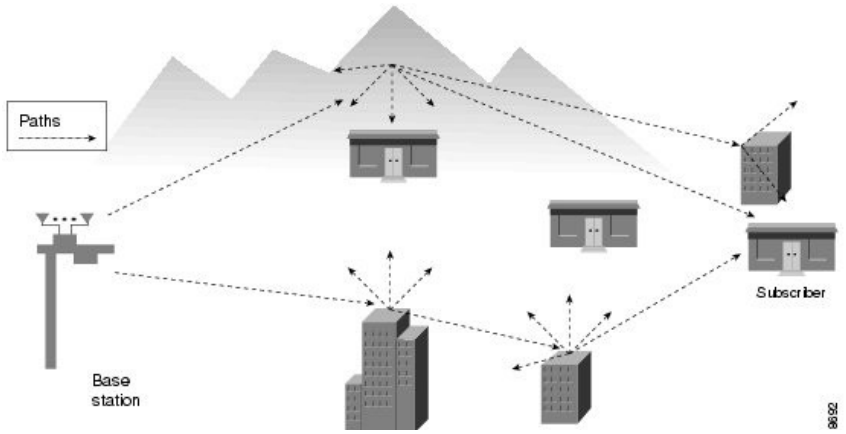

Figure 5.2: Illustration depicting multipath propagation. Adapted from (*Wireless Technologies - Cisco Documentation* 2002)

However, since it would often be difficult to know the exact time of transmission and since it would also be pointless to do so as that would entail consideration of a very long impulse response, impulse responses in the wiresless world are normally measured from the first arrival time. Then impulse responses associated with multipath propagation have an active period which has oscillations, which eventually go to zero in a way so that the envelope has an exponential decay.

Baseband Channel Impulse Response In wireless, we are generally interested in only a narrow band centred around some frequency f_c, and it is common to characterise a channel by a complex baseband channel impulse response (Rappaport 2001).

The relationship between the the baseband input $c(t)$ and the transmitted signal $x(t)$ is as follows:

$$x(t) = \mathbb{R}[c(t)e^{j2\pi f_c t}].$$

Similarly, The relationship between the the baseband output $r(t)$ and the recieved signal $y(t)$ is as follows:

$$y(t) = \mathbb{R}[r(t)e^{j2\pi f_c t}].$$

We can see that $c(t)$ corresponds to the envelope of the transmitted signal $x(t)$ and $r(t)$ corresponds to the envelope of the received signal $y(t)$.

If we ignore the factor of $\frac{1}{2}$, the baseband impulse response $h_b(t)$ is defined much like

an actual impulse response:

$$r(t) = \frac{1}{2}c(t) \otimes h_b(t).$$

Power Delay Profile The power delay profile is the squared values of the impulse response time shifted so that the line of sight time, or the first arrival of a signal is the first element (der Perre, Ness, Thoen, Bandenameele & Engels 2002).

$$p(t) = h^2(t)$$

RMS Delay Spread The RMS Delay Spread is a measure of of the spread of the impulse response. First, we define a centroid in time for the power delay profile $\bar{\tau}$ in the following way

$$\bar{\tau} = \frac{\int_{-\infty}^{\infty} t\, p(t)dt}{\int_{-\infty}^{\infty} p(t)dt},$$

then the RMS delay spread is given by (Rappaport 2001) :

$$\tau_{\text{rms}} = \sqrt{\frac{\int_{-\infty}^{\infty}(t-\bar{\tau})^2 p(t)dt}{\int_{-\infty}^{\infty} p(t)dt}}.$$

Effective Length of the Impulse Response Multipath propagation causes time dispersion of the received signal. If the effective length of the impulse response is greater than the symbol length, then the symbols will interfere with each other giving a high error rate. Hence, the symbols need to be longer than the expected length of the impulse response. Engineers use the following rule of thumb (der Perre et al. 2002) to estimate the effective length of the impulse response:

$$\Delta\tau \approx 4\tau_{\text{rms}}.$$

Time Varying Channels

It has to be added that there are other factors that limit how long a symbol should be in practice. In most applications the channel is generally not static, and symbols need to be short enough so that the channel can be regarded as static for the duration of the symbol. This consideration is especially important in mobile communications. In

the case of OFDM, it is desirable that the channel does not change substantially over a period of a single symbol. The time variability of a channel imposes a lower limit on the symbol transmission rate of a given carrier frequency.

Relative motion in the transmitter, receiver or reflectors in a multipath propagation channel causes rapid phase changes in a channel. If we limit the power loss to 3db, then we can tolerate at most a phase change of $\frac{\pi}{4}$. If we assume the very simplest model, which is that of line of site transmission with the transmitter moving directly towards or away from the reciever then our maximum symbol time is given by

$$\frac{v_c}{8v_v f_c},$$

where v_c is the speed of the carrier signal which is 3×10^8, v_v is the speed of the transmitter or receiver, and f_c is the frequency of the carrier signal. For example, the maximum symbol length for a carrier of 2.5 GHz and relative movement of 100km/h would be 0.54 milliseconds.

Relative motion in the transmitter, receiver or reflectors in a multipath propagation channel also causes a frequency change known as the doppler effect. Since OFDM features tightly packed frequency bands, frequency drifting can easily cause inter-channel interference (ICI).

Coherence Time If we observe a channel over a small enough period of time, then we may observe that the channel remained relatively constant over that period of time. Coherence Time T_c is a measure of how small that period of time is. The autocorrelation function of the channel frequency response (der Perre et al. 2002) is given by

$$\mathbf{r}_t(\Delta t) = \frac{\int_{-\infty}^{\infty} \mathbf{h}(t)\mathbf{h}^*(t+\Delta t)\, dt}{\int_{-\infty}^{\infty} |\mathbf{h}(t)|^2\, dt}$$

Clearly $\mathbf{r}_t(0) = 1$. T_c is defined to be the value of Δt where $\mathbf{r}_t(\Delta t)$ is reduced to $\frac{1}{2}$.

Coherence Bandwidth The transfer function of a given path between a transmitter and a receiver over the entire radio frequency spectrum will no doubt vary greatly. But if we take a small enough subband, then at any moment the transfer function across

that subband will be relatively flat. The frequency coherence bandwidth B_{coh} is a measure of how narrow that subband is. It effectively gives the spectral spread of signal that would see the same channel.

The correlation coefficient of the channel frequency response (Vaughan & Anderson 2003) is given by

$$\rho_f(\Delta f) = \frac{E[\tilde{\mathbf{h}}(f)\tilde{\mathbf{h}}(f+\Delta f)^*]}{E[\,|\tilde{\mathbf{h}}(f)^2|\,]}.$$

Clearly $\rho_f(0) = 1$. B_{coh} is defined to be the value of Δf where $\rho_f(\Delta f)$ is reduced to $\frac{1}{2}$.

Modulation Techniques

Amplitude Modulation Amplitude Modulation was patented in 1901 (Marconi 1901, Tesla 1900). It is depicted in Figure 5.3 and works by varying the amplitude of the carrier signal according to the amplitude of the data signal. Its use in its original form is limited now, but still ongoing.

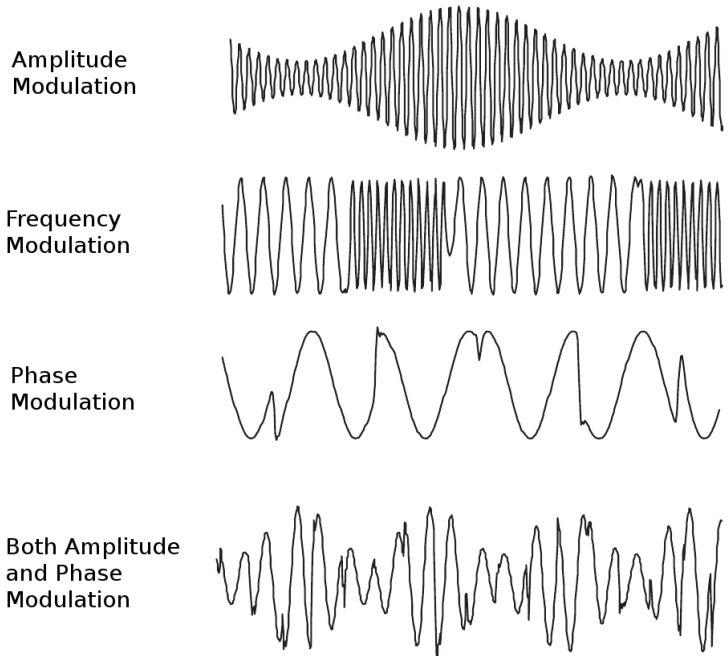

Figure 5.3: Illustration depicting the different types of wireless modulation. Adaptedfrom (Agilent Technologies 2001)

Frequency Modulation Frequency Modulation (Armstrong 1930) is depicted in Figure 5.3. It works by slightly increasing and decreasing the frequency of the carrier signal according to the amplitude of the data signal. If binary data is being transmitted, most of the energy will be centred in two frequencies, being the carrier frequency plus the frequency offset and the carrier frequency minus the frequency offset.

Phase Modulation Phase Modulation (Delaney 1958, Doelz 1949) is depicted in Figure 5.3. It is normally only used for digital communications. It works by altering the phase of the carrier signal.

Minimum Shift Keying (MSK) is a transmission system that uses phase modulation (Doelz & Heald 1958). For each bit transmitted, the phase is shifted from its current orientation either +90 degrees or -90 degrees, depending on whether the bit is a one or a zero. For each bit transmitted, there is always a change of phase and so there is no need for any other synchronisation.

Differential and Coherent Modulation Coherent Modulation encodes the data in terms of absolute phase, or absolute amplitudes. It must be remembered here, that if what we transmit is absolute, then what we receive is dependant not only on what is transmitted, but also on the channel. Hence Coherent modulation requires some reference point. There may or may not be an explicit channel estimate, but the reference point effectively constitutes an estimate of the channel.

Differential modulation, on the other hand, encodes data in the difference between the previous symbol and the current one. The previous symbol is effectively a reference for the current symbol. Hence there is no need for estimation of the channel.

A major advantage of differential modulation is that the receiver need not make an estimate of the channel, and it therefore simpler. It also removes the need for test transmissions or pilot signals. A major advantage for coherent modulation is that it requires 3dB less transmission power. It also enables the use of multi amplitude and phase Quadrature Amplitude Modulation, as described in the next paragraph (Edfors, Sandell, deBeek, Landstrom & Sjoberg 1998).

Quadrature Amplitude Modulation Quadrature Amplitude Modulation (QAM) uses phase modulation and amplitude modulation at the same time. Phase modulation and amplitude modulation are orthogonal and may used at the same time. For example, with two amplitude levels, and two phase levels, we have 4 states that can be transmitted. This is known as 4-QAM and also known as Quadrature Phase Shift Keying (QPSK). We say that 4-QAM has two bits per symbol, meaning that it allows us to send two bits of information at the same time.

We need not only have two levels of amplitude and two levels of phase. For example, if the phase shifts are 90 degrees we can use 4 different phases. Similarly, we may have 4 levels of amplitude, giving a total of 16 states. This is called 16-QAM and it transmits 4 bits per symbol.

5.1.2 OFDM

Multicarrier Modulation

As explained in Subsection 5.1.1 the use of longer symbols will give us a smaller occupied bandwidth. Hence, instead of having a single carrier frequency transporting a high data rate, we can have many carriers frequencies each transporting low data rates and can transmit approximately the same combined data rate in the same combined occupied bandwidth. The important advantage of doing this is that using longer symbols allows us to accommodate a longer effective impulse response lengths with the same amount of inter symbol interference (ISI), which is precisely what is needed under conditions of multipath propagation.

There are some practical limits to how far we can go with this, however. As explained in Subsection 5.1.1 the coherence time imposes a practical limit on how long we can make the signals. Reception would become impossible if the coherence time was shorter than the symbol time. Computational complexity imposes another practical limit on how long we can make the symbols. Another disadvantage is that effective channel latency is increased.

Orthogonal Multicarrier Modulation

Traditionally, adjacent frequencies have had to be sufficiently separated in frequency so that their occupied bandwidths do not interfere with each other. However, it is possible to move adjacent frequencies much closer together so that their frequency spreads are heavily overlapped, and still not have the channels interfere with each other. (Chang 1966) published just such a method so that frequency spreads overlap, but are orthogonal to each other in the real and imaginary parts so that they do not interfere with each other. The method can work with amplitude, frequency or phase modulation.

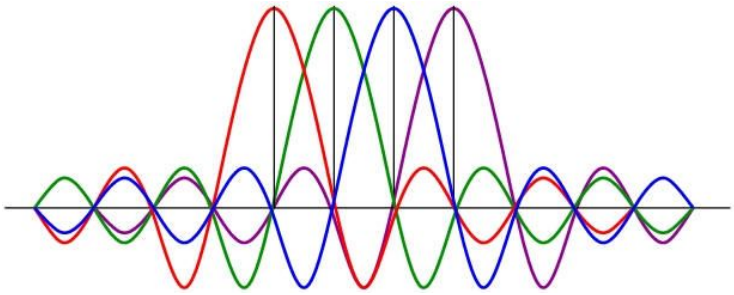

Figure 5.4: Illustration depicting the orthogonal nature of the frequency spreads of adjacent carriers in OFDM. Adaptedfrom (*OFDM in Multipath* 2007).

The frequency spread due to amplitude modulation by a square pulse, for example, is a sync function centred about the carrier frequency and with a spread proportional to the symbol rate. If the carrier frequencies are spaced at the correct distance apart, then they are orthogonal. This is illustrated in Figure 5.4 where we can see that the real parts of the green and red frequency spreads have strong positive overlap close to each other, but this is cancelled out by weak negative overlap at all other places.

Multi carrier modulation using orthogonal overlapping occupied bandwidths, is referred to as Orthogonal Frequency Division Multiplexing (OFDM). It permits the transmission of a binary order of magnitude more data to be transmitted in the same frequency range than would otherwise be possible, and successfully does this even where there is a large time dispersion due to multipath propagation.

Since radio bandwidth is a valuable resource, and since ODFM makes economical use of an allocated bandwidth, most new communication systems rely on OFDM technology.

ODFM is currently used in most modern communication systems, including broadband modems, cable TV, satellite TV, mobile phones and free to air digital audio and video broadcasting. New wireless technologies such as Long Term Evolution 3GPP (Furht & Ahson 2009), WIMAX 802.16 (Prasad & Velez 2010) and Flash-OFDM (Lee, Park, Cho, Lee & Ryu 2006) will use enhancements of OFDM technology.

The Cyclic Prefix Each symbol transmitted is prefixed by an interval that contains a cyclic extension of the time domain signal. This interval is called the *guard interval* or the *cyclic prefix*. The length of the cyclic prefix is the anticipated maximum effective length of the impulse response. Without the cyclic prefix, the multipath propagation would spread the information so that the received transmission of a given block would include information from the previous block, which would constitute inter-symbol interference, or unnecessary noise. With the cyclic prefix, the information being spread into a received block is effectively that of the same block. What gets received is effectively a cyclic convolution of the transmitted block and the impulse response. The DFT that will be applied to the received block has the property of transforming cyclic convolutions into a element by element multiplication. What we receive, therefore, is an element by element multiplication of the transfer function of the channel with the fourier representation of the transmitted signal, plus some additive noise.

Implementation of OFDM Prior to the development of efficient, fast and tiny Digital Signal Processors, spread spectrum systems had to be implemented using expensive and bulky banks of subcarrier oscillators and demodulators. There were several early spread spectrum military systems. The Kineplex data transmission system for acoustic modems (Mosier & Clabaugh 1958) was a spread spectrum system that achieved multicarrier modulation using high quality *kinematic* analog filters. Orthogonal Multicarrier wireless modulation was achieved in the KATHRYN system (Zimmerman & Kirsch 1967), using a *Fourier Transformer* implemented with ultrasonic delay line memory devices. KATHRYN was the first true OFDM system.

In 1971 Weinstein and Ebert proposed the use of the Fast Fourier Transform (FFT) as an alternative to the use of banks of analogue oscillators and demodulators (Weinstein

& Ebert 1971). The FFT had been reinvented just six years earlier in 1965 (Cooley & Tukey 1965), having been originally invented in 1805 by J.F.H. Gauss (Heideman et al. 1985).

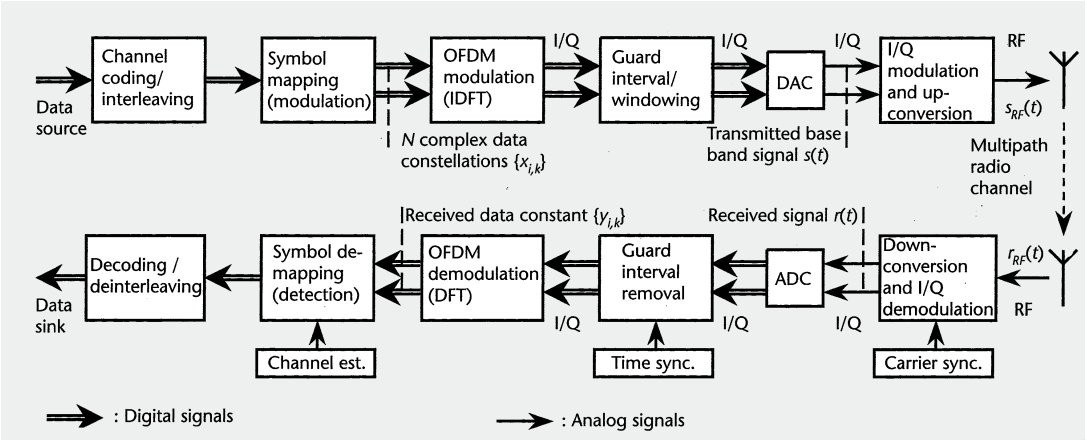

Figure 5.5: Block diagram showing the components of OFDM transmitters and receivers. Adapted from (Prasad 2004).

The transmission process is shown in Figure 5.5. A serial data stream is buffered into a block and presented as parallel data. Modulation then occurs. At this point, we are working in the frequency domain, so the "modulation" for each subcarrier essentially boils down to a complex scalar variable being assigned a complex value that encompasses both amplitude and phase. The pilot signals are inserted here too. The complex values of each frequency are then converted using an inverse FFT into time domain values, still in parallel. Then these parallel time domain values are sequenced, together with the cyclic extension, into serial form, ready for conversion from digital into an analog signal. The signal now converted into analog does not have the frequency of the carrier, but is a baseband signal as described in Subsection 5.1.1. The baseband signal is converted up to the required frequency range by frequency shifting (described in Subsection 2.2.3) and is then amplified and transmitted.

The receiver process, also shown in Figure 5.5, is similar to the transmission process but has inverse processes corresponding to those in the transmission processes in the reverse order to those in the transmission processes.

The ubiquitous use of this technology has been enabled by DFT and IDFT transforms, implemented using the fast and tiny Digital Signal Processors (DSP) that have become

available since the turn of the century.

5.1.3 Channel Estimation

Channel estimation plays an important role in ODFM. With channel estimation it is possible to use coherent phase shift keying (PSK) instead of differential PSK which would allow a 3 dB gain (Li & Stuber 2006, Shen & Martinez 2006). Techniques which utilize second order channel statistics dominate literature for channel estimation in OFDM. The utilisation of second order channel statistics is of particular interest in this dissertation.

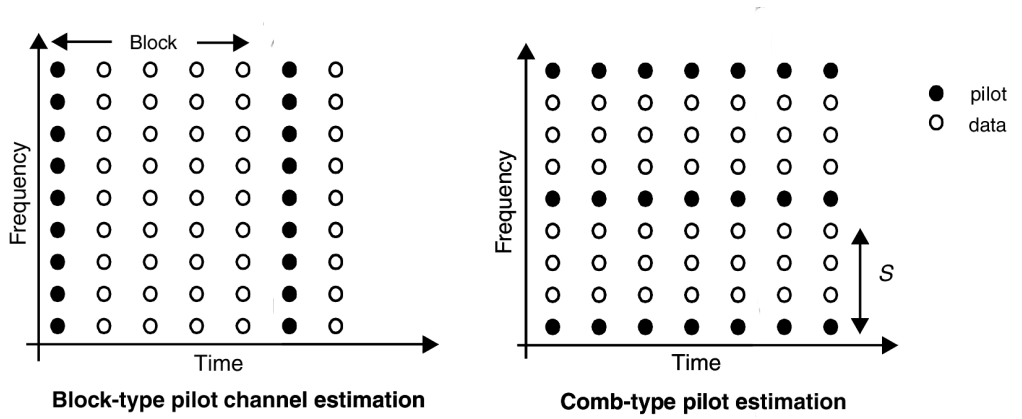

Figure 5.6: Illustration of the arrangement of pilot signals in OFDM. Adapted from (Shen & Martinez 2006).

OFDM systems generally include the use of a *pilot signal*, one of the roles of the pilot signal is that of a test signal.

Figure 5.6 shows data symbols arranged by subcarrier and by time for block type pilot estimation and for comb type pilot estimation. In block type pilot symbol arrangements, entire time slots or DFT blocks have been dedicated to pilot symbols (indicated in black). In this case the channel for each data block needs to be estimated by interpolation in time. In comb type pilot symbol arrangements, the pilot symbols have been allocated their own dedicated subcarriers. In this case the channel for each data block need to be estimated by interpolation in frequency. (Negi & Cioffi 1998) modelled the channel as first order Markov channel and demonstrated that comb type pilot pilot channel estimation is superior to block type pilot channel estimation.

Estimators

As explained in Subsection 5.1.2, the data is modulated in the frequency domain, converted to the time domain, transmitted, received and converted back in to the frequency domain. For the purpose of channel estimation, we can treat the process beginning with the inverse DFT and ending with the DFT as a black box. That is, we may treat the channel in a high level way as a channel in the frequency domain. We can demonstrate this by using a slightly simplified model (Shen & Martinez 2006):

$$\begin{aligned} \tilde{w} &= \mathbf{F}((\mathbf{F}^{-1}\tilde{\mathbf{x}}) \otimes \mathbf{h} + \mathbf{z}) \\ &= \tilde{\mathbf{x}} \diamond \mathbf{F}\mathbf{h} + \mathbf{F}\mathbf{z} \\ &= \tilde{\mathbf{x}} \diamond \tilde{\mathbf{h}} + \tilde{\mathbf{z}}. \end{aligned}$$

Least Squares Estimator Least squares estimation (LS) of the channel of an OFDM system in the frequency domain is conceptually much simpler than LS in the time domain, because the data has already been orgainised into blocks and LS can be performed with only element by element multiplication:

$$\tilde{\xi} = \tilde{\mathbf{x}} \diamond \tilde{\mathbf{h}}.$$

We may express this in the form of matrix multiplication by introducing the diagonal matrix $\tilde{\mathbf{X}} = \mathrm{Diag}(\tilde{\mathbf{x}})$. Then we have

$$\tilde{\xi} = \tilde{\mathbf{X}}\tilde{\mathbf{h}} + \tilde{\mathbf{z}},$$

and it follows that ignoring the noise, the channel estimate is given by

$$\tilde{\mathbf{h}}_L = \tilde{\mathbf{X}}^{-1}\tilde{\xi}.$$

There is no actual matrix inversion involved, however, because $\mathbf{X}$ is diagonal. The noise does have an impact on this estimate of $\mathbf{h}$, however, and the LS estimator suffers from a high mean-squared error whenever there is noise present (Shen & Martinez 2006).

Linear Minimum Mean Squared Error Estimator One way of reducing the errors is to use the Linear Minimum Mean Squared Error (LMMSE) estimator. The

(LMMSE) estimator was introduced by (Edfors et al. 1995) and takes advantage of the fact that the elements of $\tilde{h}$ are statistically correlated in time (Li & Stuber 2006). The correlations are normally expressed in the form of a correlation matrix $C_{\tilde{h}}$, which is defined by $C_{\tilde{h}} = E[\tilde{h}\tilde{h}^*]$. I will now derive the formula for the MMSE estimator, based on the derivation in (Li & Stuber 2006), beginning by explaining some of the underlying assumptions:

Relative to the entire spectrum stretching down to a frequency of zero, the frequency band of interest in OFDM is narrow, and because of this we are able to make the assumption of stationarity in the frequency domain. The assumption of stationarity implies that the matrix $C_{\tilde{h}}$ is circulant. This assumption also implies that the channel for each band, $\tilde{h_i}$, has the same variance, and for simplicity we choose the magnitude of the variance of each band is equal to be 1, i.e.

$$(C_{\tilde{h}})_{i,i} = E[\tilde{h}_i \ \tilde{h}_i^*] = 1. \tag{5.1}$$

The fact that the band of interest is narrow also permits an assumption regarding the noise. We assume here that the noise is independent and identically distributed (IID), otherwise known as white noise, and this implies that the autocorrelation matrix for the noise is diagonal and constant:

$$C_{\tilde{z}} = \rho I, \tag{5.2}$$

where ρ is the variance of the estimation error. It follows that

$$\begin{aligned} C_{\tilde{h}_L} &= E[\tilde{h}_L \ \tilde{h}_L^*] \\ &= E[(\tilde{h} + \tilde{z})(\tilde{h} + \tilde{z})^*] \\ &= E[(\tilde{h} \ \tilde{h}^* + 2\tilde{h} \ \tilde{z}^* + \tilde{z} \ \tilde{z}^*] \\ &= C_{\tilde{h}} + \rho I. \end{aligned} \tag{5.3}$$

Similarly,

$$\begin{aligned} E[\tilde{h} \ \tilde{h}_L^*] &= E[\tilde{h}(\tilde{h} + \tilde{z})^*] \\ &= E[(\tilde{h} \ \tilde{h}^* + 2\tilde{h} \ \tilde{z}^*] \\ &= C_{\tilde{h}}. \end{aligned} \tag{5.4}$$

The LMMSE estimate, $\tilde{h}_M$, is a linear function of the LS estimate, $\tilde{h}_L$. We may express this fact in the following way:

$$\tilde{h}_M = \tilde{A}\,\tilde{h}_L, \tag{5.5}$$

where $\tilde{A}$ is the *estimator coefficient matrix*.

Our problem is to minimise the error of the estimate, or equivalently minimise expectation of the square of the error of the estimate, i.e. minimise $E||\tilde{h}_M - \tilde{h}||^2$.

$\tilde{h}$, $\tilde{h}_L$ and $\tilde{h}_M$ can be viewed as vectors of zero mean random variables. The space of zero mean random variables forms a Hilbert space, if we define the inner product of two zero mean random variables, x and y, to be $E[xy]$, i.e. the correlation between them. By the orthogonality principle, the solution $\tilde{h}_M$ must occur at the point where there is no correlation between the error of the estimate $\tilde{h}_M - \tilde{h}$ and the initial estimate $\tilde{h}_L$. That is:

$$E[(\tilde{h}_M - \tilde{h})\tilde{h}_L^*] = 0. \tag{5.6}$$

Substituting (5.5) into (5.6) gives us

$$\tilde{A}E[\tilde{h}_L\tilde{h}_L^*] - E[\tilde{h}\tilde{h}_L^*] = 0. \tag{5.7}$$

Substituting (5.3) and (5.4) into (5.7) we get

$$\tilde{A}(C_{\tilde{h}} + \rho I) - C_{\tilde{h}} = 0. \tag{5.8}$$

Making $\tilde{A}$ the subject of the equation,

$$\tilde{A} = C_{\tilde{h}}(C_{\tilde{h}} + \rho I)^{-1}. \tag{5.9}$$

Equation (5.9) features shrinkage. When there is no noise, the result becomes simply $A = I$, and when there is much noise, the answer becomes like $A = \frac{1}{\rho}C_{\tilde{h}}$.

Using Singular Value Decomposition As stated in the previous subsection, the elements of $\tilde{h}$ are statistically correlated in time, and as a result A tends to have low rank. In fact, it has been shown in (Edfors, Sandell, deBeek, Wilson & Borjesson 1998) that the rank of A is L, one more than the length of the cyclic prefix.

Therefore, an alternative way to reduce the complexity would be to express the matrix A as a low rank Singular Value Decomposition (SVD) approximation $A = V\Lambda V^*$

where V contains only the vectors corresponding to the significant eigenvalues (Edfors, Sandell, deBeek, Wilson & Borjesson 1998). The LMMSE estimate may be calculated as suggested by the bracketing in the following equation:

$$\tilde{h}_M = V\,(\Lambda\,(V^*\,\tilde{h}_L)). \tag{5.10}$$

This effectively reduces the complexity of the calculation of $\mathbf{h}_{LMMSE}$ to approximately $4L^2$ operations.

The SVD decomposition of A may be expressed in terms of the SVD composition of $C_{\tilde{h}}$ (Li, Cimini & Sollenberger 1998).

$$\begin{aligned} A &= C_{\tilde{h}}(C_{\tilde{h}} + \rho I)^{-1} \\ &= (I + \rho C_{\tilde{h}}^{-1})^{-1} \\ &= (I + \rho U D^{-1} U^*)^{-1} \\ &= U(I + \rho D^{-1})^{-1} U^*, \end{aligned}$$

where D is a diagonal matrix containing the eigenvalues of $C_{\tilde{h}}$ and U are the eigenvectors of $C_{\tilde{h}}$. Therefore A and $C_{\tilde{h}}$ have the same eigenvectors (i.e. $U = V$) and each of the eigenvalues of A are given by

$$\phi_i = \frac{d_i}{d_i + \rho}, \tag{5.11}$$

where ϕ_i is the ith eigenvalue of A, and d_i is the ith eigenvalue of $C_{\tilde{h}}$.

Efficient Computation and Improvement in the Time Domain As explained in Subsection 5.1.3, the matrix $C_{\tilde{h}}$ is circulant and has the property that it is diagonalised by the inverse Fourier transform. That is, $V = F$, and we may effectively perform our SVD by means of the fast Fourier transform and its inverse, further reducing the computational complexity to approximately $M\log(M)$ operations. We can express (5.5) in the time domain:

$$\begin{aligned} \tilde{h}_M &= A\,\tilde{h}_L \\ &= F(\,(F^*AF)\,(F^*\tilde{h}_L)\,). \end{aligned}$$

Figure 5.7 illustrates this process, which consists of transformation into the time domain, truncating the impulse response to what we know to be the effective length

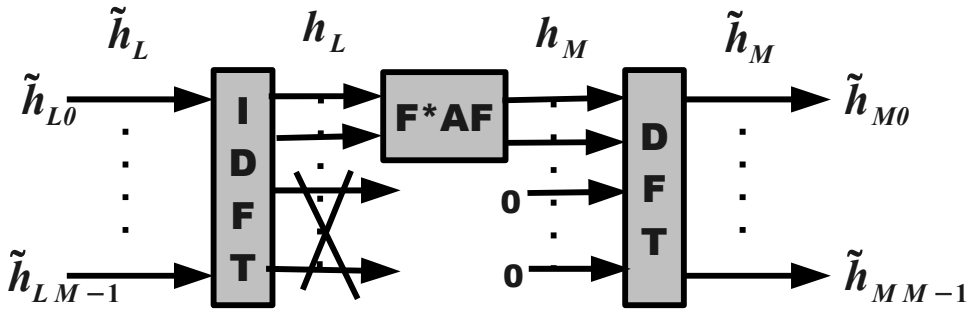

Figure 5.7: Illustration of efficient implementation of LMMSE in the time domain. Reproduced in Open Office from the original diagram in (Edfors et al. 1995).

of the impulse, multiplication by (F^*AF), and transformation back into the frequency domain. We know that values in the elements of the impulse response, h_L, after the effective length of the impulse response are due to noise. Therefore, by forcing these elements to zero, we are further improving our estimate of $\tilde{h}_M$ (Edfors et al. 1995).

5.2 Optimisation of Test Signal Crest Factors

Wideband channel sounding is often performed by measuring the response to a periodic test signal (Schroeder 1970). The channel is normally assumed to be linear. We can describe our test signal in the frequency domain:

$$x_j = \sqrt{n} \sum_{k=0}^{n-1} a_k e^{i(\frac{2\pi kj}{n} + \phi_k)}.$$

Given circumstances may require the signal strengths in each frequency, a_k, to be fixed. A white test signal, where all frequency strengths a_k are equal would be a common case. The phases at each frequency ϕ_k, however, remain to be chosen. The choice of the phases of each frequency is important. If we use a zero or a constant phase, we will typically end up with a signal that has a single huge spike. If we use random phases, the envelope of the signal may have many spikes.

In either of the above choices of the phases, the ratio of the maximum absolute value of the signal to the mean power of the signal would be high. This ratio, K_r, is known

as the crest factor (Felhauer, Baier, Konig & Mohr 1993), and is given by

$$\begin{aligned} K_r &= \frac{|x|_{\text{peak}}}{x_{\text{rms}}} \\ &= n\frac{|x|_{\text{peak}}}{\sqrt{\sum_{j=0}^{n-1} a_i^2}}. \end{aligned} \tag{5.12}$$

The transmission equipment and the receiving equipment effectively limit the maximum absolute values of the signal that may be transmitted. If these limits are exceeded, the transmission equipment may simply clip or it may become non linear. Either way, the signal becomes distorted and harmonics are generated. In this case what would ultimately get transmitted or received would be something other than the intended spectrum.

It is desirable to have a large signal to noise ratio, so we wish the average power envelope of the signal to be as high as possible. A test signal with a constant envelope will give us the highest average power envelope for a given absolute maximum instantaneous value. It is therefore desirable to keep the envelope of the test signal as constant as possible, i.e. to minimise K_r, the crest factor of the test signal, by choosing suitable values for each of the ϕ_k.

At present, there is no analytical formula which selects the phases to give a minimum crest factor. In this section we will review the rules and optimisations developed to give a low crest factor or an optimal crest factor. A comprehensive computer search of all possible combinations of each phase becomes impossible whenever there are more than a handful of frequencies involved. Another method, called the *random* method, uses a computer program to try random combinations of phases, and to retain the combination that gives the lowest crest factor.

5.2.1 Schroeder's Formula

Schroeder's may be used to derive a test signal with a good crest factor. Even when it is not used to give the final values of the phases of the test signal, it is generally used to give the initial values for an optimisation of the test signal crest factor.

(Schroeder 1970) derived a formula for the values of ϕ_k that gives a good crest factor for a subset of the general case. He constructed a frequency modulated (FM) signal that has the desired power spectrum, and derives the formula for that specific case. (Schroeder 1970) acknowledges that it is not possible in general to fit an exact FM signal to an arbitrarily prescribed power spectrum. However, it is observed that the derived rule applied to the general case gives consistently better results than the random method (der Ouderaa, Schoukens & Renneboog 1988).

Schroeder's formula for each ϕ_k is:

$$\phi_k = \phi_1 - 2\pi \sum_{j=1}^{k-1} (k-j) a_j^2 \qquad k = 2, \cdots .n, \tag{5.13}$$

and for a white test signal this reduces to

$$= -\pi \frac{k^2}{n}. \tag{5.14}$$

5.2.2 Domain Swapping Methods

(Van den Bos 1987) uses Schroeder's formula as the initial starting point for an optimisation which further improves the Crest Factor. His optimisation procedure involves an application of the DFT and an application of the inverse DFT as part of each iteration. Each iteration is as follows:

- The signal is converted to the time domain, and then the signal is converted to full amplitude square waves by replacing the instantaneous value of $x(t)$ with the sign of $x(t)$.
- Then the signal is converted back to the frequency domain. Here, the phases are retained, and the frequency strengths are replaced by the original frequency strengths.

Instead of converting to full magnitude square waves, (der Ouderaa et al. 1988) proposed a modification where the time domain signal is merely clipped to a level of something less than the peak. They recommend a level of 20% less than the peak for the first iteration and then gradually reducing the clipping level with each iteration. If the

optimisation becomes trapped in a local minima, the clipping level should be raised again.

5.2.3 L-norm Methods

The definition of the crest factor, (5.12), involves the maximum of the absolute values of the test signal in the time domain, somtimes known as the l_∞-norm of the test signal. The l_∞-norm may be seen as the limit of a family of l_p-norms. The l_p-norm of the function x(t) taken over the interval $[a, b]$ is denoted by $l_p(x)$ and is defined as:

$$l_p(x) = \left(\frac{1}{b-a} \int_a^b |x(t)|^p \, dt \right)^{\left(\frac{1}{p}\right)}.$$

Two other commonly known norms are the l_1-norm (mean) and the l_2-norm (RMS). The l_∞-norm is defined to be:

$$l_\infty(x) = \frac{lim}{p \to \infty} \, l_p(x). \tag{5.15}$$

Taken as a function of the phases, the l_∞-norm is highly non-smooth (Manchester 2009). The l_∞-norm may be approximated by (5.15) using a large value of p, however, and this approximation is smooth. The method of (Guillaume et al. 1991) uses Schroeder's formula to give the initial values for the phases, and then used the Gauss-Newton algorithm to find the local minima of $l_4(x)$. It then uses the result as a better starting point to find the local minima of $l_8(x)$, and so on, for binary logarithmic steps of p. (Manchester 2009) uses this same scheme, but has reformulated the crest factor optimisation problem in such a way to make it suitable for convex optimisation instead of using the Gauss-Newton method.

This scheme finds a local minima and in general cannot find the global minimum. Nevertheless, the performance of this scheme exceeds the performance of previous schemes. The performance of an L-norm method is compared with that of a domain swapping method in Figure 5.8. Both methods begin with Schroeder's formula. The domain swapping method initially achieves good results with little computation effort, but after some initial gains makes slow progress, whereas the L-norms method continues to make substantial further reductions as p is increased to 1024. After this point, the

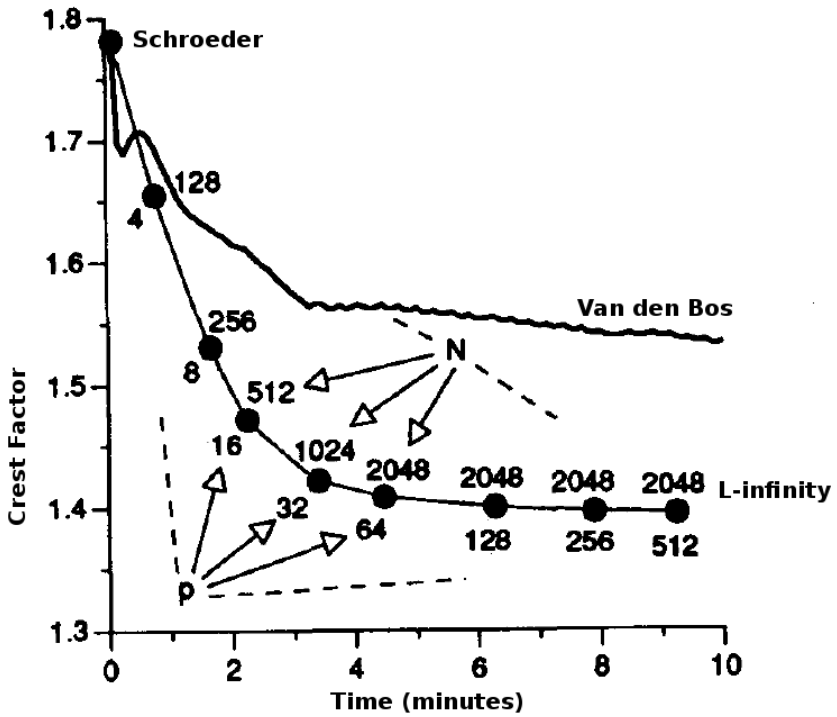

Figure 5.8: Graph comparing the method of (Guillaume et al. 1991) with the method of (Van den Bos 1987). Adapted from (Guillaume et al. 1991).

method of L-norms makes only minor improvements with increasing computational time.

5.2.4 Other Methods

(Felhauer et al. 1993) propose the systematic search of the space of phases using a computer. They note that since the resulting test signal must be real, we would have only half the number of phases to consider than we would if we considered complex test signals.

The authors fail to mention that the method would be very limited in the number of frequencies that could feasibly be optimised, but the computational complexity of this brute force approach is quite steep and this clearly could not be used for a fullband audio spectrum. If the resolution in the phases was set to $\frac{2\pi}{m}$, so that there are m possible values for each phase, then the computational complexity would be m^{n-1} where n is the number of frequencies to consider. If for example, if the resolution in the phases

was set to 5 degrees, so that there are 72 possible values for each phase, then if we had only 10 frequencies to consider, there would be approximately 3.7×10^{18} different combinations to evaluate, and if we had 100 frequencies to consider, there would be approximately 5.4×10^{185} different combinations to consider.

(Molina, Fannin & Timoney 1995) used a general minimax optimiser that is part of the Matlab optimisation toolbox. The actual Matlab routine is not specified. (der Ouderaa et al. 1988) specifically recommended against the use of a general optimisation routine because the problem is rich in local minima, and the optimisation process frequently becomes stuck.

5.3 Chapter Summary

In this chapter we have examined some aspects of wireless technology where the concepts have applications in the echo cancellation and in the optimal test signal part of this dissertation in particular. The second order statistics of wireless channels and the noise are used to impove the estimation of the wireless channels. In the next chapter, we use the second order statistics of the noise of the channel in the optimisation of test signals.

We have seen in this chapter that test signals for channel estimation are important, and that the optimal choice of test signals is a topic of interest, in wireless communication applications. Literature exists in the wireless world regarding the optimisation of the crest factors of test signals. In the next chapter we perform optimisation of the spectral densities of test signals.

Chapter 6

Optimisation of Test Signals

A time-varying linear channel may be estimated using the output from a periodic test signal. In this chapter we show how to do this in a way that takes full account of the past history of the background noise and the past history of the channel.

The use of a test signal to estimate a channel is common in wireless applications (in 802.11 and 3G, for example). Wideband channel sounding is often performed by measuring the response to a periodic test signal (Schroeder 1970). Acoustic test signals are also useful for rapid channel estimation under difficult circumstances (Goodings et al. 1993, Spriet et al. 2005).

No prior work exists in the literature concerned with the optimisation of the power spectral densities of test signals, given the secondary statistics of the channel and of the noise. There have been several published enhancements to LMS and AP echo cancellation algorithms that exploit knowledge of the noise power to adjust the rate of convergence, for example (Wei, Gelfand & Krogmeier 2001, Costa & Mermudez 2008, Mader, Puder & Schmidt 2000, Benesty, Rey, Vega & Tressens 2006). In Section 5.1 we have documented the use of secondary statistics of the channel in order to improve the channel estimation in OFDM. In Section 5.2 we have described Crest Factor Optimisation, which is the optimisation of the phases of a test signal in order to minimise peak instantaneous power given the power spectral densities of the test signal. Crest Factor Optimisation is orthogonal and complementary to the type of

optimisation developed in this chapter. It would be quite feasible to perform Crest Factor Optimisation on a test signal whose spectral densities have been found by means of optimisation developed here.

The remainder of this chapter is organised as follows: In Section 6.1 we show how a periodic test signal can be represented as a circulant matrix.

In Section 6.2 an explicit formula is obtained for the optimal linear estimator when we know the autocovariance or power spectrum of the interfering noise and the autocovariance of the echo channel variation. This is a potentially useful result on its own, that may be used for rapid channel estimation for a given periodic test signal. We define a measurable quantity *coloredness* for the noise covariance matrix and for the channel covariance matrix. We compare the optimal estimator with an estimator which does not use information about the statistics of $\mathbf{h}$ or $\mathbf{z}$ for different values of coloredness of the noise and the channel autocovariance matrices.

In Section 6.3 we develop a closed formula in the time domain for the optimal test signal for a fixed estimator given the secondary statistics of the noise and the channel. Matrix calculus identities required for the derivation of this expression for the optimal estimator are stated and proved in the appendix. These two expressions are then combined in an algorithm that iteratively finds the optimal combination of estimator and test signal in the time domain. We give details neccesary for the implementation of the algorithm.

In Section 6.4 we develop an improved implementation in the frequency domain. One may observe from the results of optimisation experiments in the time domain that the choice of a different starting point (test signal) will generally yield a different result. That is, the optimal test signal is not unique in the time domain. However, converted to the frequency domain, we can see that different optimal test signals in the time domain share the same power spectral densities but different phases. We show mathematically that the performance of a test signal is dependent only on the power spectral density and independent of the phases. We use this result to develop an iterative frequency domain method for finding the optimal test signal that uses only half the number of free variables.

In Section 6.6 we use our definition of coloredness and present results from a large number of numerical experiments that demonstrate that even with a much lower power budget the optimal test signal can achieve the same least-squares accuracy as a white test signal when either the channel covariance matrix or the noise covariance matrix is colored.

6.1 Periodic Test Signal Expressed as a Circulant Matrix

Using the variable names illustrated in Figure 1.1, the test signal $x(t)$ is the input to the channel with a finite impulse response (FIR), $h(t)$. Noise $z(t)$ is added to the channel output, $y(t)$, giving the measurable result, $w(t)$. The relationship between $x(t)$ and $y(t)$ is given by

$$y(t) = x(t) \otimes h(t). \tag{6.1}$$

We use samples in discrete time rather than with continuous functions of time, and we assume an impulse response with an effective length of n. We use the convention that x_m refers to the discrete value of x at time t_m where ($t_m = m\Delta t$) and Δt is the sampling interval. We use a similar convention for z_m, y_m and w_m. Therefore, we may write a set of n equations

$$y_i = \sum_{k=0}^{n-1} x_{i-k}\, h_k \qquad i = 0, ..., n-1.$$

This may be expressed in terms of matrix multiplication by a Toeplitz matrix $\mathbf{X}$ (Plemmons 1993), as we did in Subsection 2.4.1:

$$\mathbf{y} = \mathbf{X}\mathbf{h}, \tag{6.2}$$

where $\mathbf{X}_{i,k} = x_{i-k}$ and $\mathbf{y} = (y_0, ..., y_{n-1})^T$.

Clearly if we ignore the noise $z(t)$, we may estimate $\mathbf{h}$ from a sampling of w by

$$\hat{\mathbf{h}} = \mathbf{X}^{-1}\mathbf{w}, \tag{6.3}$$

where $\mathbf{w} = (w_0, ..., w_{n-1})^T$. This estimate is essentially the solution that would be obtained by the application of a non-Bayesian algorithm that does not utilise the statistical knowledge of the noise or the channel, such as the RLS algorithm applied to a white test signal as a block of data, as discussed in Subsection 2.4.1.

We now introduce an additional assumption, that x is periodic, which greatly simplifies the estimation procedure and enables it to be expressed readily by means of the discrete Fourier transform. We assume that the length of the repeated portion of the test signal is the same length as the modelled impulse response, n. This is not a restriction on the range of test signals which can be explored, because a signal with a repeating period shorter than the impulse response could not provide a satisfactory estimate of the channel, and if a test signal longer than the impulse response is desired we can extend the impulse response by padding with zero values until our impulse response has the desired length. The repeated portion of the test signal may be represented as a vector $\mathbf{x} = (x_0, ..., x_{n-1})^T$, and it follows that $\mathbf{X}$ is the circulant matrix given by $\text{Circ}(\mathbf{x})$.

Because our signal x is cyclic, so is y. We can estimate $\mathbf{h}$ from an accumulation of the corresponding cyclic elements of w from k sampling periods. This would have the effect of summing the noise power (i.e. increasing the effective noise strength by $\sqrt{k}$) and a summing of the test signal (i.e. increasing the test signal strength by k), thus increasing the signal to noise ratio.

6.2 The Optimal Estimator for a Given Test Signal

6.2.1 Finding the Optimal Linear Estimator

The estimate $\hat{\mathbf{h}}$ of $\mathbf{h}$ given by (6.3) is only the optimal estimator for the channel if there is no noise or if we ignore the noise. For example, if we know the noise power, then (A.23) gives a more accurate estimate because it performs *shrinkage* as does methods described in (van Houwelingen 2001, Goldstein & Brown 1978).

If we know the power spectral densities of the noise, and therefore its autocorrelation matrix $C_{\mathbf{z}} = E[\mathbf{z}\mathbf{z}^T]$, we may do better than (A.23). Further, if we know the mean of the channel $\bar{\mathbf{h}}$, and autocorrelation matrix of the channel $\mathbf{C}_h = E[\mathbf{h}\mathbf{h}^T]$, then we may do better still, as we shall show in Subsection 6.2.5. These statistical properties of $\mathbf{h}$ may be estimated by keeping a record of the channel over time. Statistically efficient

estimators of $\mathbf{C}_h$ are discussed in the next chapter. We also assume that the noise, $\mathbf{z}$, is uncorrelated with $\mathbf{x}$ and uncorrelated with the current value of the slowly varying channel, $\mathbf{h}$.

6.2.2 The Estimate of h is a linear function of w

If the random quantities in our model are Gaussian and the model is linear then by the Gauss-Markov Theorem (Stapleton 1995), the optimal unbiased least-squares estimator is linear. Let us therefore express our estimate of $\mathbf{h}$, $\hat{\mathbf{h}}$ as a linear function of our measured output signal, $\mathbf{w}$. Then we have

$$\hat{\mathbf{h}} = \bar{\mathbf{h}} + \mathbf{A}(\mathbf{w} - \mathbf{X}\bar{\mathbf{h}}), \tag{6.4}$$

where A fully incorporates our linear estimation function. The estimator has been constructed deliberately to ensure that it is unbiased, i.e. $E[\hat{\mathbf{h}}] = \bar{\mathbf{h}}$.

6.2.3 Estimation as a minimisation problem.

Least-squares estimation of $\mathbf{h}$ is achieved by minimising the sum of the squares of the errors in $\mathbf{h}$:

$$\varepsilon = E[\Big(\mathbf{h} - \bar{\mathbf{h}} - (\hat{\mathbf{h}} - \bar{\mathbf{h}})\Big)^T \Big(\mathbf{h} - \bar{\mathbf{h}} - (\hat{\mathbf{h}} - \bar{\mathbf{h}})\Big)] \tag{6.5}$$

which, substituting (6.4) into (6.5),

$$\begin{aligned} &= E[(\mathbf{h} - \bar{\mathbf{h}} - \mathbf{AXh} + \mathbf{AX}\bar{\mathbf{h}} - \mathbf{Az})^T \quad (\mathbf{h} - \bar{\mathbf{h}} - \mathbf{AXh} + \mathbf{AX}\bar{\mathbf{h}} - \mathbf{Az})] \\ &= E[\mathbf{M}], \end{aligned}$$

where

$$\begin{aligned} \mathbf{M} = &(\mathbf{h} - \bar{\mathbf{h}})^T(\mathbf{h} - \bar{\mathbf{h}}) \quad - \quad (\mathbf{h} - \bar{\mathbf{h}})^T(\mathbf{AX}(\mathbf{h} - \bar{\mathbf{h}})) \\ - &(\mathbf{h} - \bar{\mathbf{h}})^T(\mathbf{Az}) \quad - \quad (\mathbf{AX}(\mathbf{h} - \bar{\mathbf{h}}))^T(\mathbf{h} - \bar{\mathbf{h}}) \\ + &(\mathbf{AX}(\mathbf{h} - \bar{\mathbf{h}}))^T(\mathbf{AX}(\mathbf{h} - \bar{\mathbf{h}})) \quad + \quad (\mathbf{AX}(\mathbf{h} - \bar{\mathbf{h}}))^T(\mathbf{Az}) \\ - &(\mathbf{Az})^T(\mathbf{h} - \bar{\mathbf{h}}) \quad + \quad (\mathbf{Az})^T(\mathbf{AX}(\mathbf{h} - \bar{\mathbf{h}})) \\ + &(\mathbf{Az})^T(\mathbf{Az}). \end{aligned}$$

So

$$\begin{aligned}
\varepsilon &= E[(\mathbf{h}-\bar{\mathbf{h}})^T(\mathbf{h}-\bar{\mathbf{h}})] \quad - \quad E[(\mathbf{h}-\bar{\mathbf{h}})^T\mathbf{AX}(\mathbf{h}-\bar{\mathbf{h}})] \\
&- E[(\mathbf{h}-\bar{\mathbf{h}})^T\mathbf{Az}] \quad - \quad E[(\mathbf{h}-\bar{\mathbf{h}})^T\mathbf{AX}(\mathbf{h}-\bar{\mathbf{h}})] \\
&+ E[(\mathbf{h}-\bar{\mathbf{h}})^T\mathbf{X}^T\mathbf{A}^T\mathbf{AX}(\mathbf{h}-\bar{\mathbf{h}})] \quad + \quad E[\mathbf{z}^T\mathbf{A}^T\mathbf{AX}(\mathbf{h}-\bar{\mathbf{h}})] \\
&- E[(\mathbf{h}-\bar{\mathbf{h}})^T\mathbf{Az}(\mathbf{h}-\bar{\mathbf{h}})] \quad + \quad E[\mathbf{z}^T\mathbf{A}^T\mathbf{AX}(\mathbf{h}-\bar{\mathbf{h}})] \\
&+ E[\mathbf{z}^T\mathbf{A}^T\mathbf{Az}]
\end{aligned}$$

Terms containing both $\mathbf{h}$ and $\mathbf{z}$ are zero, because $\mathbf{h}$ and $\mathbf{z}$ are uncorrelated, therefore:-

$$\begin{aligned}
\varepsilon = E[(\mathbf{h}-\bar{\mathbf{h}})^T(\mathbf{h}-\bar{\mathbf{h}})] - 2E[(\mathbf{h}-\bar{\mathbf{h}})^T\mathbf{AX}(\mathbf{h}-\bar{\mathbf{h}})] \\
+E[(\mathbf{h}-\bar{\mathbf{h}})^T\mathbf{X}^T\mathbf{A}^T\mathbf{AX}(\mathbf{h}-\bar{\mathbf{h}})] + E[\mathbf{z}^T\mathbf{A}^T\mathbf{Az}]
\end{aligned}$$

Applying (A.9) we get:-

$$\varepsilon = \mathrm{tr}(\mathbf{C}_h) - 2\mathrm{tr}(\mathbf{AXC}_h) + \mathrm{tr}(\mathbf{X}^T\mathbf{A}^T\mathbf{AXC}_h) + \mathrm{tr}(\mathbf{A}^T\mathbf{AC}_z) \tag{6.6}$$

6.2.4 The Optimal Estimator A

Since (6.6) expresses the mean-squared error as a quadratic function in $\mathbf{A}$, the optimal value for $\mathbf{A}$ must be a stationarity of (6.6) considered as a function of $\mathbf{A}$. The stationary value we seek may be found by differentiating (6.6) with respect to $\mathbf{A}$, which is achieved by applying (A.10) and (A.11) to (6.6):-

$$\nabla_A(\varepsilon) = -2(\mathbf{XC}_h)^T + 2\mathbf{AXC}_h\mathbf{X}^T + 2\mathbf{AC}_z$$

Setting $\nabla_A(\varepsilon)$ to zero, we find

$$\mathbf{A}(\mathbf{XC}_h\mathbf{X}^T + \mathbf{C}_z) = (\mathbf{XC}_h)^T$$

In practice, acoustic channels do not transmit energy at a frequency of zero (DC). As an illustration, let us imagine a LEM system where energy at a frequency of zero would

have the same effect as having the position of the speaker located slightly forward or slightly backward. Hence, we should not be surprised to find that h, z, $\mathbf{C}_h$ and $\mathbf{C}_z$ do not have energy at a frequency of zero. It follows that $\mathbf{C}_h$ and $\mathbf{C}_z$ will both be singular in practice, and the optimal test signal should also have a zero DC component. Therefore, $(\mathbf{X}\mathbf{C}_h\mathbf{X}^T + \mathbf{C}_z)$ may be singular and the obvious solution

$$\mathbf{A} = (\mathbf{X}\mathbf{C}_h)^T(\mathbf{X}\mathbf{C}_h\mathbf{X}^T + \mathbf{C}_z)^{-1}$$

cannot be used. In practice $\mathbf{A}$ can be computed by using

$$\mathbf{A} = (\mathbf{X}\mathbf{C}_h)^T(\mathbf{X}\mathbf{C}_h\mathbf{X}^T + \mathbf{C}_z)^{+}. \tag{6.7}$$

6.2.5 Comparing the Optimal Estimator with a White One

Since noise with constant (flat) power spectral density is termed white, we say that the *coloredness* of the noise, c_{cz}, is $\frac{\sigma_{pz}}{\mu_{pz}}$, where σ_{pz} and μ_{pz} are the standard deviation and the mean of the power spectral density of the noise respectively. Since the diagonal of $\mathbf{C}_{\hat{z}}$ has the discrete power spectral densities, we prefer to define coloredness directly in terms of $\mathbf{C}_{\hat{z}}$:

$$c_{cz} = \frac{\sigma_{\mathbf{p}}}{\mu_{\mathbf{p}}},$$

where $\mathbf{p} = \text{diag}(\mathbf{C}_{\hat{z}})$, and μ_p is the average of the elements of $\mathbf{p}$ and σ_p is the standard deviation of the elements of $\mathbf{p}$.

Even though the impulse of the channel is not a stationary process, for the sake of convenience we will use the term *power spectral density* of C_h to refer to the total power associated with each frequency band, i.e. the values along the diagonal of $\mathbf{C}_{\hat{h}}$ in a manner analagous to that of the noise covariance matrix. Similarly, the coloredness of the channel, c_{ch}, is defined in terms of $\mathbf{C}_{\hat{h}}$:

$$c_{ch} = \frac{\sigma_{\mathbf{q}}}{\mu_{\mathbf{q}}},$$

where $\mathbf{q} = \text{diag}(\mathbf{C}_{\hat{h}})$, and μ_q is the average of the elements of $\mathbf{q}$ and σ_q is the standard deviation of the elements of $\mathbf{q}$.

The graphs shown in Figure 6.1 summarise the results of over 500 experiments in which three methods for channel estimation have each been applied to the same problem.

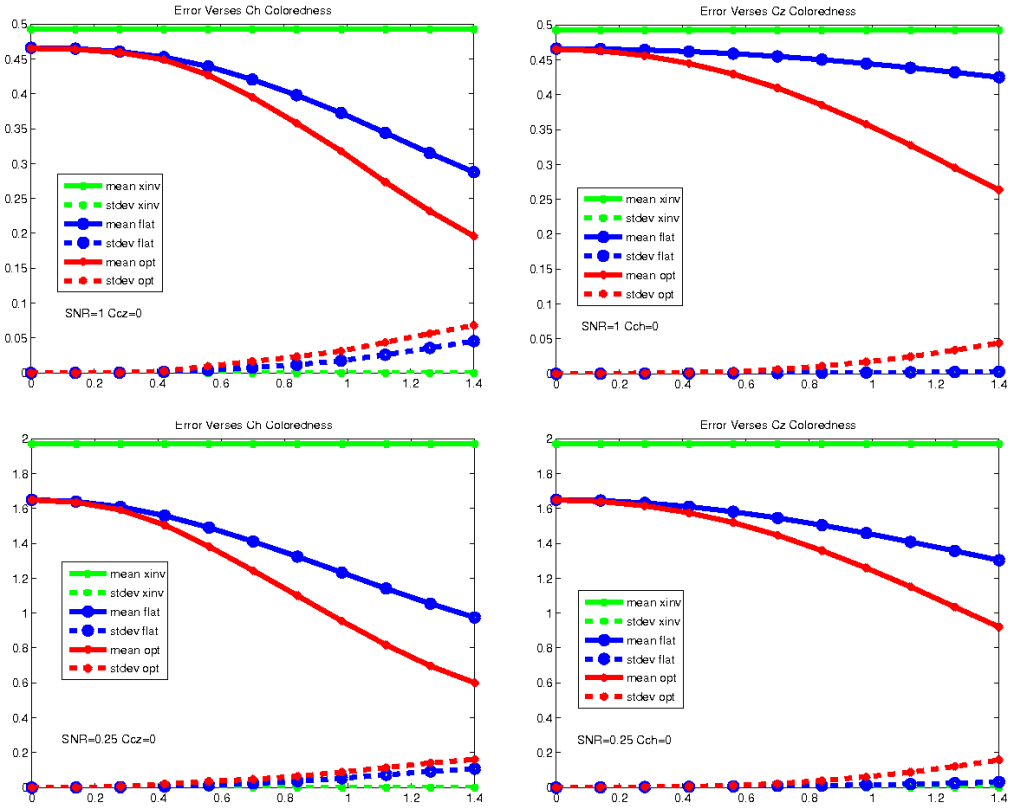

Figure 6.1: The means and standard deviations of the expected sum of the square of the square of the expected errors. Details are explained in Subsection 6.2.5. Created in Matab.

These plots demonstrate how and when the optimal estimator is better than one that does not take second order statistics into account:

The solid curves represent the *means* of the expected channel estimation errors over eleven different randomly generated channel estimation scenarios and the dashed lines represent the *standard deviations* of those same results. The channel covariance matrix in each experiment was randomly generated according to the required coloredness in the frequency domain. The noise covariance matrix was also created from a randomly generated set of power spectral densities whose mean and variance conform to the specified coloredness in the frequency domain.

The green lines show the errors for the non-Bayesian estimator given by (6.3) (e.g. the RLS algorithm described in Section 2.4 without weighting) with a white test signal. The blue lines show the errors using the optimal estimator with a white test signal. The red lines show the mean and standard deviations of the errors using the optimal estimator in combination with an optimal test signal.

The left hand graphs present the errors for varying coloredness of C_h with white noise. It shows that the errors decrease with coloredness of C_h, when the optimal estimator is used.

The right hand graphs present the errors for varying coloredness of C_z with white noise. It shows that the errors decrease with noise coloredness, when the optimal estimator is used.

The top graphs show the errors with an SNR of 1.0. The bottom graphs show the errors with an SNR of 0.25. Note that the graph shows that the errors are slightly better for the optimal estimator even when the channel autocovariance is white and the noise is white noise. This is because even with white channel and noise covariances, the optimal estimator takes the noise power into account, effectively performing shrinkage as described in Subsection A.3.5, whereas the non-Bayesian estimator cannot take the noise power into account. These graphs show that as the SNR becomes smaller, the benefit from shrinkage becomes more pronounced.

6.3 A Time Domain Method for Finding the Optimal Test Signal

Since $\mathbf{X}$ is a circulant matrix that has $\mathbf{x}$ as its first column, i.e. $\mathbf{X} = \text{Circ}(\mathbf{x})$ the optimisation under consideration (finding the best test signal) has only n free variables and should be expressed in terms of $\mathbf{x}$ rather than $\mathbf{X}$.

6.3.1 Imposing the power constraint

As we have noise present in the system, a stronger signal will give a better result. That is, unconstrained, there is no finite optimal test signal. We therefore impose the constraint that the total power in the signal $\mathbf{x}$ is less than or equal to the maximum power p of the test signal. Clearly, the optimal test signal must reside on the constraint. However, for a given value of $\mathbf{A}$ the optimal test signal may or may not reside on the constraint.

The energy constraint on the test signal can be introduced to the optimization by means of the Kuhn-Tucker theorem (Luenberger 1969)(chapter 9), which states that the constrained minimum of our objective function occurs at a point where the derivative of the constraint and the derivative of the objective function have opposite direction, that is:

$$c\nabla_x(g) + \nabla_x(\varepsilon) = 0, \tag{6.8}$$

for some positive c.

The power constraint can be expressed as

$$g(\mathbf{x}) = \mathbf{x}^T\mathbf{x} - p \leq 0, \tag{6.9}$$

and thus, the left hand term of (6.8) is

$$c\nabla_x(g) = 2c\mathbf{x}^T. \tag{6.10}$$

Focussing now on the right hand term of (6.8), we may apply (A.5) and (A.7) to (6.6),

giving

$$\begin{aligned}\nabla_x \varepsilon &= -2\ \text{traces}(\mathbf{C}_h\mathbf{A})^T + \text{traces}((\mathbf{C}_h)^T\ \mathbf{X}^T\ (\mathbf{A}^T\mathbf{A})^T)^T + \text{traces}(\mathbf{C}_h\ \mathbf{X}^T\ \mathbf{A}^T\mathbf{A})^T \\ &= -2\ \text{traces}(\mathbf{C}_h\mathbf{A})^T + 2\ \text{traces}(\mathbf{C}_h\ \mathbf{X}^T\ \mathbf{A}^T\mathbf{A})^T. \qquad (6.11)\end{aligned}$$

Substituting the LHS of (6.10), and the LHS of (6.11) into (6.8), we get

$$2c\mathbf{x} - 2\ \text{traces}(\mathbf{C}_h\mathbf{A}) + 2\ \text{traces}(\mathbf{C}_h\ \mathbf{X}^T\ \mathbf{A}^T\mathbf{A}) = 0.$$

By applying (A.8), we get

$$2c\mathbf{x} - 2\ \text{traces}(\mathbf{C}_h\mathbf{A}) + 2\mathbf{G}\mathbf{x} = 0,$$

where each element of the matrix $\mathbf{G}$ is given by

$$\mathbf{G}_{i,j} = \sum_{k=0}^{n-1}\sum_{l=0}^{n-1}(\mathbf{C}_h)_{k,l}(\mathbf{A}^T\mathbf{A})_{(l+j)\%n,(i+k)\%n.} \qquad (6.12)$$

Thus

$$(c\mathbf{I} + \mathbf{G})x = \text{traces}(\mathbf{C}_h\mathbf{A}),$$

and so

$$x = (c\mathbf{I} + \mathbf{G})^{-1}\text{traces}(\mathbf{C}_h\mathbf{A}). \qquad (6.13)$$

Whenever the unconstrained optimal value of $\mathbf{x}$ lies outside the hypersphere, the c that satisfied the constraint will be positive. Whenever the unconstrained optimal value of $\mathbf{x}$ lies inside the hypersphere, then we need to use unconstrained optimisation, and this is achieved by setting c to zero.

6.3.2 Optimisation Method

Although any combination of test signal x, and estimator, A, which satisfies (6.7) and (6.13) must be optimal, and each of these equations provides an explicit formula for the optimal estimator and the optimal test signal respectively, these equations do not directly provide a formula for the optimal *combination* of test signal and estimator. However, iterative use of these two formulae does provide us with the optimal combination and this procedure has been implemented in Matlab.

Table 6.1, which contains the results from an example optimisation, illustrates the algorithm. A major iteration of the optimisation uses the current choice of x, which I will refer to as the *starting* point, and derives the optimal estimator for the starting point using (6.7). The optimisation then finds the optimal test signal for this optimal estimator using (6.13) with the constant c chosen (by using the Newton Raphson method) to make the constraint (6.9) zero. This second point is termed here the *direction* point. In Table 6.1, the horizontal lines separate the results of each major iteration from the next major iteration. The column of the table labelled *Dir* shows the square of the expected error for the direction point of each major iteration, ε, in the rows labelled *eps*, and the distance from the starting point to the direction point in the rows labelled *dist*.

Iter		Start	Dir	Probe1	Probe2	Probe3	Probe4	Probe5
1	dist:	0	7.0151	14.030				
1	eps:	2.3205	**1.2407**	2.2223				
2	dist:	0	1.2851	2.5702	5.1404	10.281		
2	eps:	1.2047	0.9213	0.7433	**0.5901**	0.5481		
3	dist:	0	0.4366	0.8733	1.7465	3.4930	6.9860	13.9721
3	eps:	0.5901	0.5157	0.4575	0.3755	0.2939	**0.2821**	0.2612
4	dist:	0	2.7349	5.4699				
4	eps:	0.2821	**0.2319**	0.2698				
5	dist:	0	0.3633	0.7267	1.4534	2.9067		
5	eps:	0.2319	0.2237	0.2175	**0.2100**	0.2137		
6	dist:	0	0.3188	0.6375	1.2750	2.5500		
6	eps:	0.2100	0.2046	0.2005	**0.1964**	0.2089		
7	dist:	0	0.3892	0.7784	1.5569			
7	eps:	0.1964	0.1918	**0.1893**	0.1893			
...	...	...	...	...	...	...		
65	dist:	0	0.0017	0.0035	0.0070			
65	eps:	0.1623	0.1623	0.1623	0.1623			

Table 6.1: Table depicting the probe distance and the expected square of the errors of the test signal during the test signal optimisation.

Unfortunately, the distance between the starting point and the direction point is typically small, so we use the starting point and the direction point to give us a direction which we use to probe further out from the starting point. At each minor iteration,

we double the distance from the starting point, and project back onto the surface of the spherical constraint giving us the next probe point. The distance from the starting point to the probe point of each minor iteration and the expected square of the error at the probe point is shown in the columns of the table labelled *Probe*. If we have moved the maximum distance or if the expected error at the current probe point turns out to be worse than at the previous probe point or the direction point, then the previous probe point is used as the starting point for the next major iteration.

We deliberately reduce the strength of the test signal by a factor of four, prior to the first major iteration. Given that the constraint is a hypersphere that limits power, the resulting point does not touch the constraint and the initial major iterations are therefore unconstrained. This permits a short cut through the interior of the constraint sphere rather than navigating across the surface of it and results in faster initial progress. Because of this, the expected error shown in the table appears to get worse in the first iteration.

Major iterations continue until only marginal improvements are achieved. Although the number of iterations to achieve a minimisation objective accurate to within 1e-6 varied from 7 to 70, most of the gain is already achieved in the first few iterations.

Optimisation from a different starting test signal x using the same secondary statistics produces a different test signal with the same resulting expectation of the errors. When converted to the frequency domain, test signals resulting from different starting points have the same power spectral densities, but have different phases. Clearly, the resulting test signal may be subjected to crest factor optimisation.

6.3.3 Implementation Details

Creating the matrix $\mathbf{G}$ by directly implementing (6.12), would impose an unacceptable computational complexity of order n^4. However, if we recognise this equation as a 2D convolution, then we can implement it using a 2D discrete Fourier transform (DFT) and element by element multiplication in the Fourier domain. This results in a complexity of $n^2 \log(n)$.

Solving (6.9) and (6.13) together was achieved by means of the Newton-Raphson method. If we substitute (6.13) into (6.9), we obtain the power constraint as a function of c. The derivative of that function can be found as follows:-

$$\frac{d}{dc}(c\mathbf{I}+\mathbf{G}) = \mathbf{I}$$

By applying (A.12), we get:-

$$\frac{d}{dc}(c\mathbf{I}+\mathbf{G})^{-1} = -(c\mathbf{I}+\mathbf{G})^{-1}(c\mathbf{I}+\mathbf{G})^{-1}$$

$$\frac{d}{dc}x = -(c\mathbf{I}+\mathbf{G})^{-1}(c\mathbf{I}+\mathbf{G})^{-1}\text{traces}(\mathbf{C}_h\mathbf{A}) \tag{6.14}$$

If we substitute (6.13) into (6.14) we can obtain the derivative of the function that we need for the Newton-Raphson method:-

$$\frac{d}{dc}(\mathbf{x}^T\mathbf{x}-p) = -2\mathbf{x}^T(c\mathbf{I}+\mathbf{G})^{-1}\mathbf{x} \tag{6.15}$$

There are obvious refinements of our implementation, which have the potential to significantly improve computational speed. For example, the Woodbury matrix identity (Hager 1989) gives an efficient formula for the inverse of a matrix, if we already know the inverse of a similar matrix (e.g. the matrix from the last iteration). The use of this identity has not been used at this stage.

6.3.4 Comparison with the use of a General Purpose Optimiser

A general purpose nonlinear optimiser was used for the purpose of comparisons. The function to minimise was constructed by using (6.7) to express $\mathbf{A}$ as a function of $\mathbf{x}$ and then using (6.6) to express ε as a function of $\mathbf{A}$ and $\mathbf{x}$. The resulting function proved difficult to differentiate, and we leave the differentiating as a future endeavour. Fortunately, commercial non-linear optimisers are able to deduce their own gradients, although at a heavy cost in terms of computational complexity, and we permitted the optimiser to find its own gradient.

The general purpose optimisers NPSOL (Stanford Business Software 2011), CONSOLVE (Tomlab Optimization 2011*a*) and *fmincon* (MathWorks 2008*a*) are all regu-

larly updated. The optimiser *fmincon* comes with MATLAB and was therefore readily available for use. The optimiser *fmincon* is also widely used. Promotional material for CONSOLVE uses an example where CONSOLVE outperforms *fmincon*, by a factor of 100 (Tomlab Optimization 2011*b*). However, the comparisons provided are not relevant to this work. Second order derivatives were supplied to CONSOLVE in the comparison, while *fmincon* was disadvantaged by being unable to use this information. As explained above, we will not provide even the first order derivatives, therefore CONSOLVE would not enjoy the advantage of using second order derivatives.

This procedure for identifying the optimal signal converged but proved to be slow, as indicated in Table 6.5, which shows the processing times of both methods. The table elements for *fmincon* for higher orders of n are missing because the optimisation did not complete within a reasonable period of time. These timings were obtained using *Matlab* on *Linux* running on an *Intel Core 2* processor running at 1.86GHz.

n	General Optimiser T Domain	Our Method T Domain	General Optimiser F Domain	Our Method F Domain
16	1	0	2	0
32	7	1	4	1
64	57	2	13	1
128	1293	19	307	18
256	22427	163	4302	102
512	-	4065	-	4348
1024	-	55169	-	76675

Table 6.2: Table showing the timings in seconds for various problem dimensions n, for the general purpose optimiser, *fmincon* and our method, working in the time domain and in the frequency domain.

Using the general purpose optimiser did not in general result in the same test signal as the method we developed, but did produce a test signal with the same power spectral densities and the performance. Only the phases of the test signal differed. Hence, there were strong reasons to suspect that the optimality of a test signal depended only on

the spectral densities of the signal and not the phases of the signal. This inspired the creation of a frequency domain method where the optimal test signal is expressed only in terms of the spectral densities of the test signal. Actual test signals could then be created by assigning phases to each frequency.

6.4 A Frequency Domain Method for Finding the Optimal Test Signal

In this section we show that the predicted errors for a test signal are dependent on the spectral densities of the signal and insensitive to the phases of the test signal. We then proceed to create a method of discovering the optimal test signal in the frequency domain. Despite the number of apparently free variables being effectively halved in the frequency domain method, it did not converge faster than the time domain method in our experiments.

We adopt a small additional assumption which is normally adopted, namely that $\mathbf{z}$ is wide sense stationary. In addition, observe that if the noise process were to repeat periodically, outside the period during which our measurements were taken it would make no difference to the measurements. We can therefore assume that the noise is periodic in this form, and therefore $\mathbf{C}_z$ is circulant in the time domain, and its frequency domain counterpart, $\mathbf{F}^*\mathbf{C}_z\mathbf{F}$, is diagonal. In effect, we have shown by this argument that the values in the original covariance of $\mathbf{z}$ which cause the Toeplitz matrix $\mathbf{C}_z$ to be different from a circulant matrix do not enter into the calculation of the optimal estimator or the optimal test signal, and so they may be replaced by the appropriate entries which ensure that the covariance matrix is circulant.

In the case where the test signal is sent $k+1$ times and $k > 1$ measurements are recorded from the experiment, the covariance, $\mathbf{C}_z$, must be of $\{Z_k + Z_{k+N} + \cdots + Z_{k+KN}\}_{k\geq 0}$ rather than of the Z process itself. Hence, in this case also, $\mathbf{C}_z$ will be a circulant $N \times N$ matrix.

Since $\mathbf{X}$ is a circulant matrix, it is diagonalised by the discrete Fourier transform in

the following way

$$X = \sqrt{n}\,\mathbf{F}^*\mathbf{D}\mathbf{F}. \tag{6.16}$$

Where $\mathbf{D} = \mathrm{diag}(\mathbf{d})$ and $\mathbf{d}$ is the discrete Fourier transform of x (i.e. one of the columns of $\mathbf{X}$). The test signal has a zero mean.

$$\mathbf{d}_0 \;=\; 0$$

Since $\mathbf{X}$ is real valued, $\mathbf{d}$ must be skew symmetrical. So assuming an even value of n:

$$\mathbf{d}_i \;=\; \mathbf{d}_{n-i}, \quad i = 1, \ldots, \frac{n}{2} - 1.$$

This appears to transform $\mathbf{X}$ into a parameterization with $n/2$ free parameters, as desired, however in general these variables will take complex values so that there are really n independent variables.

On the other hand, we expect that the *phases* of the complex coefficients of $\mathbf{d}$ will not affect ε, so let us express the element of $\mathbf{d}$ using polar co-ordinates:

$$\mathbf{d}_k = \mu_k e^{j\theta_k}, \quad k = 0, \ldots, n-1,$$

with μ_k and θ_k real. Also let us denote

$$\Theta = \mathrm{Diag}\left((0,\ e^{i\theta_1}, \ldots, e^{i\theta_{n-1}})^T\right),$$

and

$$\mathbf{P} = \mathrm{Diag}\left((0,\ \mu_1, \ldots, \mu_{n-1})^T\right).$$

In the optimization of ε by variation of $\mathbf{d}$, we shall confirm, after a change of variable, that ε is insensitive to the values of θ_k and find an equation for the vector μ which contains $n/2$ independent variables.

Let us replace $(\cdot)^T$ by $(\cdot)^*$ in (6.6) (since the matrices in that equation are real-valued) and then use (6.16) to replace references to $\mathbf{X}$ by references to $\mathbf{D}$, giving

$$\begin{aligned}
\varepsilon \;&=\; \mathrm{tr}(\mathbf{C}_h) - 2\sqrt{n}\,\mathrm{tr}(\mathbf{A}\mathbf{F}^*\mathbf{D}\mathbf{F}\mathbf{C}_h) \\
&\qquad + n\,\mathrm{tr}((\mathbf{F}^*\mathbf{D}\mathbf{F})^*\mathbf{A}^*\mathbf{A}\mathbf{F}^*\mathbf{D}\mathbf{F}\mathbf{C}_h) \;+\; \mathrm{tr}(\mathbf{A}^*\mathbf{A}\mathbf{C}_z) \\
&=\; \mathrm{tr}(\mathbf{C}_h) - 2\sqrt{n}\,\mathrm{tr}(\mathbf{A}\mathbf{F}^*\mathbf{D}\mathbf{F}\mathbf{C}_h) \\
&\qquad + n\,\mathrm{tr}(\mathbf{F}^*\mathbf{D}^*\mathbf{F}\mathbf{A}^*\mathbf{A}\mathbf{F}^*\mathbf{D}\mathbf{F}\mathbf{C}_h) \;+\; \mathrm{tr}(\mathbf{A}^*\mathbf{A}\mathbf{C}_z),
\end{aligned}$$

which setting $\mathbf{B} = \mathbf{A}\mathbf{F}^*\theta\mathbf{F}$,

$$\begin{aligned} &= \operatorname{tr}(\mathbf{C}_h) \; - \; 2\sqrt{n}\,\operatorname{tr}(\mathbf{B}\mathbf{F}^*\mathbf{P}\mathbf{F}\mathbf{C}_h) \\ &\qquad + n\,\operatorname{tr}(\mathbf{F}^*\mathbf{P}^*\mathbf{F}\mathbf{B}^*\mathbf{B}\mathbf{F}^*\mathbf{P}\mathbf{F}\mathbf{C}_h) \; + \; \operatorname{tr}(\mathbf{A}^*\mathbf{A}\mathbf{C}_z). \end{aligned}$$

Therefore, since the frequency domain counterpart of $\mathbf{C}_z$ is diagonal, $\operatorname{tr}(\mathbf{A}^*\mathbf{A}\mathbf{C}_z) = \operatorname{tr}(\mathbf{A}\mathbf{C}_z A^*) = \operatorname{tr}(\mathbf{B}\mathbf{C}_z\mathbf{B}^*) = \operatorname{tr}(\mathbf{B}^*\mathbf{B}\mathbf{C}_z)$, and so

$$\begin{aligned} \varepsilon &= \operatorname{tr}(\mathbf{C}_h) - \; 2\sqrt{n}\,\operatorname{tr}(\mathbf{B}\mathbf{F}^*\mathbf{P}\mathbf{F}\mathbf{C}_h) \\ &\qquad + n\,\operatorname{tr}(\mathbf{F}^*\mathbf{P}^*\mathbf{F}\mathbf{B}^*\mathbf{B}\mathbf{F}^*\mathbf{P}\mathbf{F}\mathbf{C}_h) \; + \; \operatorname{tr}(\mathbf{B}^*\mathbf{B}\mathbf{C}_z). \end{aligned} \tag{6.17}$$

If we envisage ε as a function of $\mathbf{B}$, μ, and θ rather than of $\mathbf{A}$, μ, and θ, by (6.17), ε becomes insensitive to variation in θ:

$$\nabla_\theta \varepsilon \equiv 0,$$

as predicted. Since ε is clearly independent of θ, we can set θ_k to any value. We choose to set $\theta_k \equiv 0$, $k = 1, \ldots, n$, because with this choice, $d_k \equiv \mu_k$ and $\mathbf{A} = \mathbf{B}$. Hence we may write

$$\begin{aligned} \varepsilon &= \operatorname{tr}(\mathbf{C}_h) - 2\sqrt{n}\,\operatorname{tr}(\mathbf{A}\mathbf{F}^*\mathbf{D}\mathbf{F}\mathbf{C}_h) \\ &\qquad + n\,\operatorname{tr}(\mathbf{F}^*\mathbf{D}^*\mathbf{F}\mathbf{A}^*\mathbf{A}\mathbf{F}^*\mathbf{D}\mathbf{F}\mathbf{C}_h) \; + \; \operatorname{tr}(\mathbf{A}^*\mathbf{A}\mathbf{C}_z). \end{aligned}$$

So by (A.3) and (A.4),

$$\begin{aligned} \nabla_d \varepsilon &= [\; - 2\sqrt{n}\,\operatorname{diag}(\mathbf{F}\mathbf{C}_h\mathbf{A}\mathbf{F}^*) \\ &\qquad + n\,\operatorname{diag}((\mathbf{F}\mathbf{A}^*\mathbf{A}\mathbf{F}^*)\mathbf{D}(\mathbf{F}\mathbf{C}_h\mathbf{F}^*)) \\ &\qquad + n\,\operatorname{diag}((\mathbf{F}\mathbf{C}_h\mathbf{F}^*)\mathbf{D}(\mathbf{F}\mathbf{A}^*\mathbf{A}\mathbf{F}^*))\;]^T. \end{aligned}$$

We may use (A.2) to express the derivative in terms of $\mathbf{d}$ instead of $\mathbf{D}$, and at the same time, bring it to the outside of those terms,

$$\begin{aligned} \nabla_d \varepsilon &= [\; - 2\sqrt{n}\,\operatorname{diag}\mathbf{F}(\mathbf{C}_h\mathbf{A}\mathbf{F}^*) \\ &\qquad + n\,((\mathbf{F}\mathbf{A}^*\mathbf{A}\mathbf{F}^*) \diamond (\mathbf{F}\mathbf{C}_h\mathbf{F}^*)^T)\mathbf{d} \\ &\qquad + n\,((\mathbf{F}\mathbf{C}_h\mathbf{F}^*) \diamond (\mathbf{F}\mathbf{A}^*\mathbf{A}\mathbf{F}^*)^T)\mathbf{d}\;]^T. \end{aligned} \tag{6.18}$$

Elimination of Redundancy

While differentiation with respect to $\mathbf{d}$ would be desirable if $\mathbf{d}$ contained all free variables, this is not the case here. In order for the test signal $\mathbf{x}$ to be real, $\mathbf{d}$ must have

skew symmetry, and since we have chosen zero phases, $\mathbf{d}$ is real with symmetry. In addition, we are aware that the DC component of $\mathbf{d}$ has to be zero. Therefore, the number of free variables is not n, but $\frac{n}{2}$. All the variables in $\mathbf{d}$ are completely dependent on $\mathbf{d}_1 \ldots \mathbf{d}_{\frac{n}{2}}$, and we will label this sub vector as δ.

To impose the symmetry constraint on $\mathbf{d}$, let us introduce the matrix $\mathbf{Q}$ so that

$$\mathbf{d} = \mathbf{Q}\delta,$$

and

$$\nabla_\delta \, \mathbf{d} = \mathbf{Q}, \tag{6.19}$$

where $\mathbf{Q} = (q_{ij})$ is the $n \times (n/2)$ matrix defined by

$$q_{ij} = \begin{cases} 1 & i = j + 1, \\ 1 & i = (n - 1 - j), \\ 0 & \text{otherwise}, \end{cases} \tag{6.20}$$

where $j = 0, \ \ldots, \ n/2 - 1$.

Combining (6.18) and (6.19) using the chain rule gives us:

$$\begin{aligned} \nabla_\delta \varepsilon \;\; = \;\; & [\; - 2\sqrt{n} \, \mathrm{diag}(\mathbf{F}\mathbf{C}_h\mathbf{A}\mathbf{F}^*) \\ & + n \, ((\mathbf{F}\mathbf{A}^*\mathbf{A}\mathbf{F}^*) \diamond (\mathbf{F}\mathbf{C}_h\mathbf{F}^*)^T) \, \mathbf{Q}\delta \\ & + n \, ((\mathbf{F}\mathbf{C}_h\mathbf{F}^*) \diamond (\mathbf{F}\mathbf{A}^*\mathbf{A}\mathbf{F}^*)^T) \, \mathbf{Q}\delta \;]^T \, \mathbf{Q}. \end{aligned} \tag{6.21}$$

The power constraint

As already noted for the time domain optimisation, there must be a constraint on the power of the test signal. As we have noise present in the system, a stronger signal will give a better result. That is, unconstrained, there is no finite optimal test signal. We therefore impose the constraint that the total power in the signal $\mathbf{d}$ is less than or equal to the maximum power p of the test signal. Clearly, the optimal test signal must reside on the constraint. However, for a given $\mathbf{A}$ the optimal test signal may or may not reside on the constraint.

The energy constraint on the test signal can be introduced to the optimization by means of the Kuhn-Tucker theorem (Luenberger 1969)(chapter 9), which states that the constrained minimum of our objective function occurs at a point where the derivative of the constraint and the derivative of the objective function have opposite direction, i.e.

$$c\nabla_\delta(s) + \nabla_\delta(\varepsilon) = 0 \tag{6.22}$$

for some positive c.

The power constraint s can be expressed as

$$s = \mathbf{d}^T\mathbf{d} - p \leq 0, \tag{6.23}$$

and so,

$$\nabla_{\mathbf{d}}(s) = 2\mathbf{d}^T. \tag{6.24}$$

and combining (6.24) and (6.19) using the chain rule gives:

$$\begin{aligned} \nabla_\delta(s) &= 2\mathbf{d}^T\mathbf{Q} \\ &= 2\delta^T\mathbf{Q}^T\mathbf{Q}. \end{aligned} \tag{6.25}$$

Thus, substituting equation (6.21) and equation (6.25) into equation (6.22), we get

$$\begin{aligned} c\mathbf{Q}^T\mathbf{Q}\delta &= 2\sqrt{n}\,\mathbf{Q}^T\mathrm{diag}(\mathbf{F}\mathbf{C}_h\mathbf{A}\mathbf{F}^*) \\ &- n\,\mathbf{Q}^T((\mathbf{F}\mathbf{A}^*\mathbf{A}\mathbf{F}^*)\diamond(\mathbf{F}\mathbf{C}_h\mathbf{F}^*)^T)\,\mathbf{Q}\delta \\ &- n\,\mathbf{Q}^T((\mathbf{F}\mathbf{C}_h\mathbf{F}^*)\diamond(\mathbf{F}\mathbf{A}^*\mathbf{A}\mathbf{F}^*)^T)\,\mathbf{Q}\delta. \end{aligned}$$

Since c is an abitrary constant, we have allowed it to absorb the factor of two. Hence,

$$\begin{aligned} \delta &= [\, c\mathbf{Q}^T\mathbf{Q} + n\,\mathbf{Q}^T(\\ &\quad (\mathbf{F}\mathbf{A}^*\mathbf{A}\mathbf{F}^*)\diamond(\mathbf{F}\mathbf{C}_h\mathbf{F}^*)^T + (\mathbf{F}\mathbf{C}_h\mathbf{F}^*)\diamond(\mathbf{F}\mathbf{A}^*\mathbf{A}\mathbf{F}^*)^T \\ &\quad)\,\mathbf{Q}\,]^{-1} \times 2\sqrt{n}\,\mathbf{Q}^T\mathrm{diag}(\mathbf{F}\mathbf{C}_h\mathbf{A}\mathbf{F}^*), \end{aligned} \tag{6.26}$$

in which c is determined by the condition $|\mathbf{d}|^2 = p$.

6.4.1 Optimisation Method

Although any combination of test signal δ, and estimator, $\mathbf{A}$, which satisfies (6.7) and (6.26) must be optimal, and each of these equations provides an explicit formula for the optimal estimator and the optimal test signal, respectively, these equations do not directly provide a formula for the optimal *combination* of test signal and estimator, as in the case of the time domain optimisation.

However, iterative use of these two formulae does provide us with the optimal combination and this procedure has been implemented in Matlab as follows. Beginning with an initial choice for δ, (6.7) is used to find the optimal $\mathbf{A}$, and then (6.26) is used to find the optimal δ for the new value of $\mathbf{A}$. The difference between the initial δ and the new δ gives us a direction. Rather than repeat this process, which might converge quite slowly, we move in the direction indicated, taking geometrically increasing steps until the result is no longer an improvement. Referring to the above procedure as a single iteration, the first iteration usually results in a significant gain, and subsequent iterations result in diminishing gains. Iterations are performed until the gains from an iteration are no longer significant.

Solving (6.23) and (6.26) together was achieved by means of the Newton-Raphson method, for which we need the equation of the constraint, and also its gradient. If we substitute (6.26) into (6.23), we obtain the power constraint as a function of c. The gradient of the constraint s was derived as follows:-

From (6.26), we have

$$\delta = (c\,\mathbf{Q}^T\mathbf{Q} + \mathbf{G})^{-1}\mathbf{v}, \tag{6.27}$$

where

$$\mathbf{G} = n\,\mathbf{Q}^T((\mathbf{F}\mathbf{A}^*\mathbf{A}\mathbf{F}^*) \diamond (\mathbf{F}\mathbf{C}_h\mathbf{F}^*)^T + \ (\mathbf{F}\mathbf{C}_h\mathbf{F}^*) \diamond (\mathbf{F}\mathbf{A}^*\mathbf{A}\mathbf{F}^*)^T)\,\mathbf{Q}$$

and

$$\mathbf{v} = 2\sqrt{n}\,\mathbf{Q}^T\mathrm{diag}(\mathbf{F}\mathbf{C}_h\mathbf{A}\mathbf{F}^*).$$

Application of (A.12) to (6.27) gives us

$$\begin{aligned} \nabla_c \delta &= -(c\,\mathbf{Q}^T\mathbf{Q} + \mathbf{G})^{-1} \cdot \mathbf{Q}^T\mathbf{Q} \cdot (c\,\mathbf{Q}^T\mathbf{Q} + \mathbf{G})^{-1} \cdot \mathbf{v} \\ &= -(c\,\mathbf{Q}^T\mathbf{Q} + \mathbf{G})^{-1}\mathbf{Q}^T\mathbf{Q}\delta. \end{aligned} \tag{6.28}$$

Combining (6.25) and (6.28) by means of the chain rule yields the gradient for determining c by the Newton Raphson method:

$$\nabla_s c = -2\,\delta^T\,\mathbf{Q}^T\mathbf{Q}\;\,(c\,\mathbf{Q}^T\mathbf{Q} + \mathbf{G})^{-1}\mathbf{Q}^T\mathbf{Q}\delta.$$

6.4.2 Implementation using a General Purpose Optimiser

Matlab's general purpose nonlinear optimiser *fmincon* (MathWorks 2008*a*) was also used for the purpose of comparisons, as it was for the time domain optimisations. The function to minimise was constructed by using (6.7) to express A as a function of x and then using (6.6) to express ε as a function of A and x. The resulting function proved difficult to differentiate, and so we were not able to provide the gradient to the optimizer, but permitted *fmincon* to find its own gradient.

This procedure for identifying the optimal signal converged in some cases but proved to be unreasonably slow in some of these cases, as indicated in Table 6.5, which shows the processing times of both methods. The table elements for *fmincon* for higher orders of n are missing because the optimisation did not complete within a reasonable period of time. These timings were obtained using *Matlab* on *Linux* running on an *Intel Core 2* processor running at 1.86GHz.

6.5 Comparing the Frequency Domain Method with the Time Domain Method

The optimisation problem was formulated in the time domain in Section 6.2. The time domain solution follows directly from that formulation. Observing that the phases of the test signal have no effect on the test signal optimality allows us to express the test signal in the frequency domain using effectively half the number of variables used in the

time domain. One would normally anticipate that using half the number of variables would lead to a more efficient solution, and this was a major motivation for finding a frequency domain solution.

The expected improvement has been realised when using the general purpose optimiser, *fmincon*. As shown in Table , when the problem dimension n is 256, then *fmincon* takes 22427 seconds to complete when finding the solution in the time domain, and only 4302 seconds to complete when finding the solution in the frequency domain.

Our developed optimisation method, on the other hand, appears to work approximately as well in the time domain as it does in the frequency domain. As shown in Table , when the problem dimension n is 512, then our time domain method takes 4065 seconds to complete and our frequency domain method takes 4348 seconds to complete. It is not clear why the time domain method performs as well as the frequency domain method, and investigating this is a topic for future work.

6.6 Comparing Optimal Test Signals with White Ones

If one has no information about the secondary statistics of the channel history and of the noise the best strategy would be to use a test signal with a flat power spectral density, e.g. white noise. On the other hand, if we have this information available, then a test signal with a flat power spectral density is, in general, not the optimal test signal.

For each of the randomly generated channel estimation scenarios used to make the graphs in Figure 6.1, we used our optimiser to find an optimal test signal. The solid red line shows the expected error using an optimal test signal as evaluated by (6.6). Here also, each point on the line represents the mean of the results from several experiments, and the dashed line represents the standard deviation of those same results.

These graphs clearly show that an optimal test signal gives a better estimate than does a flat spectrum test signal when either the noise or the covariance of the channel is colored. The optimal channel with a completely flat $\mathbf{C}_h$ and a completely flat $\mathbf{C}_z$, is

white noise, i.e. it has a flat power spectrum density. When either or both of $\mathbf{C}_z$ and $\mathbf{C}_h$ have a non flat spectrum, better test signals than white noise are possible.

6.7 Applying the Technology

Test signals are commonly used in wireless applications, and may be used in acoustic situations where channel estimation is difficult. We have shown that the optimal test signal is superior to a white one whenever the noise or the channel history is colored. If we have the secondary statistics, we can find the optimal test signal, and use it to determine the channel. Of course, it is necessary to have estimates of the secondary statistics of the channel and the noise in order to determine the optimal test signal.

In some environments, while the echo channel may change, the second order statistics of the channel may be relatively static. An example where the secondary statistics of the channel may be static would be that of remote auscultation in telehealth clinics, described in Subsection 1.2.1. While the patients may come and go, the fixtures would be constant and the auscultation performed in the same locations.

However, reliance on fixed statistics is not a robust plan because changes in the environment could potentially cause changes in the statistics. Therefore, even in situations where the secondary statistics are static, a system that continually updates the estimate of the secondary statistics is preferable. Systems for estimating the channel statistics and the statistics of the noise, therefore, are highly desirable.

Estimation of the secondary statistics of the noise is relatively trivial, as we will show in Section 7.2. The best method for the estimation of the secondary statistics of the channel is not so obvious, however. In the next chapter, we develop a simplified model for the statistics of the echo channel that may used as the basis for an efficient method of estimating the channel statistics.

6.8 Chapter Summary

In this chapter we have derived a formula for the optimal channel estimator for a given test signal, given the secondary statistics of the channel and of the noise. The estimator performs substantially better than one that that fails to take the above statistics into account.

We have also derived a time domain formula for the optimal test signal given the secondary statistics of the channel and of the noise and used this formula, together with above mentioned estimator, to produce an algorithm for the optimal test signal.

We have noticed that resulting optimal test signals for a given set of parameters are not unique but each optimal signal shares a unique set of spectral densities. We have proven that the optimality of a test signal does not depend on the phases of the signal, and have developed an algorithm for the optimisation of a test signal in the frequency domain, that uses only the spectral densities of the signal.

We have shown that the optimal test signal for channel estimation is, in general, not white, if one has knowledge of the secondary statistics of the channel or of the noise. An optimal signal that takes into account the second order statistics of the noise and the channel outperforms a white test signal when either the channel history or the noise is coloured in the frequency domain.

Chapter 7

Statistical Model of Channel Variation

The experiments in Chapter 6 were carried out on a randomly generated ensemble of systems in which both the characteristics of statistical variation of the channel and the noise characteristics were varied. Carrying out the experiments in this way ensures that the results are not an accident due to the particular choice of channel variation characteristics or of the noise characteristics, but rather are valid systematically.

In this chapter we present the model used to generate these random combinations of channel and noise. The statistical model of channel variation is important. As well as guiding the choice of channels in the experiments, it demonstrates the existence of a statistical model of channel variation which is both realistic and parsimonious.

If we observe that channel autocovariance matrices are symmetric, then there are $\frac{n^2+n}{2}$ elements to specify, which is too many to specify in practice without the use of a simplified model. The model for $\mathbf{C}_z$ developed here needs $\frac{n}{2}$ parameters to be specified. The model for $\mathbf{C}_h$ developed here needs $\frac{3n}{2}$ parameters to be specified. We also show that we can also reduce the number of parameters to a set of 5 or so whose number does not increase with the value of n.

This is important because this model can be used as the basis for an estimation proce-

dure for the channel autocovariance which is more efficient than the obvious estimator based on averaging the observed correlations. Since the number of parameters in the complete channel autocovariance is quite large it is important to identify a better estimation procedure. Once a model with a small number of parameters has been found, an estimator for these parameters is implicitly defined by the maximum likelihood procedure, i.e. we can select the parameters by maximising the likelihood of the observations when a particular set of parameters is chosen. The task of developing this estimator has not been pursued in this work, except to the extent of demonstrating the existence of a parsimonious and realistic statistical model of channel variation.

Although an impulse response is not a stationary process, we refer to the *power spectrum densities of* C_h in this chapter. This usage has been explained in Subsection 6.2.5 and refers to the power associated with each frequency along the diagonal of $\mathbf{C}_{\hat{h}}$.

We also use the word *coloredness* in this chapter. It refers to a measurable quantity, and it has been defined in Subsection 6.2.5.

7.1 Channel Autocovariance Matrices

We will now develop a simplified model of a channel autocovariance matrix that permits us to specify the diagonal in the time domain and the diagonal in the frequency domain, subject to the constraint that the overall power in the time domain is equal to the overall power in the frequency domain. Importantly, the method that we use in order to create channel autocovariance matrices allows us to arbitrarily specify the powers in each frequency (and hence the coloredness) as the diagonal in the frequency domain. It is also clear that there will normally be significant variation in the mean power of the impulse response in the time domain (the diagonal terms of $\mathbf{C}_h$), so we seek a method which allows these terms to be specified arbitrarily.

7.1.1 The Assumption of the Zero Ensemble Average

Estimation of the mean is necessary, and the mean channel impulse response will not normally be zero, however estimation of the covariance of channel variation about the mean is a more fundamental problem. For simplicity, however, we assume that ensemble average of each element of the impulse response is zero, i.e.:

$$E[h_k] = 0, \qquad i = 0, ..., n-1.$$

Thus we may write

$$\mathbf{C}_h = E[\mathbf{h}\mathbf{h}^T].$$

7.1.2 Requirements in the Time Domain

The diagonal terms of $\mathbf{C}_h$ represent the mean power of each element of the impulse response in the time domain.

We say that the impulse response has an effective length in the time domain. This is a simplification. Acoustic impulse responses are called dispersive impulse responses and normally take the form illustrated in Figure 2.5 and described in Subsection 2.1.2. They have an envelope that decays exponentially after the peak. The effective length of the impulse response n is chosen so that the envelope approaches zero before the nth element. It follows that a channel autocovariance matrix element $\mathbf{C}_h(j,k)$ must also go to zero as j or k approach n.

In our simplest statistical model of channel variation, we use the following formula to approximate the mean power of realistic dispersive channels in the time domain:

$$\mathbf{C}_h(j,j) = aj \times e^{-bj},$$

where a and b are constants chosen to give a realistic approximation of the impulse response of a dispersive channel. This is a model with only two parameters regardless of the value of n.

The relationship between the autocovariance matrix in the time domain and the fre-

quency domain is given by:

$$\mathbf{C}_{\hat{h}} = \mathbf{F}\mathbf{C}_h\mathbf{F}^*.$$

In Subsection A.3.4 it was shown that if $\mathbf{C}_h$ is diagonal then $\mathbf{C}_{\tilde{h}}$ will be circulant.

A simplistic model of channel variation can be obtained by assuming $\mathbf{C}_h$ is diagonal, implying statistical independence between the values of $\mathbf{h}$. Since the diagonal element represent mean power, they cannot be negative. Clearly, this simplistic model is both symmetric and non-negative definite.

7.1.3 The Zero DC Transmission Assumption

We assume in this thesis that the elements of the impulse responses of acoustic channels sum to zero, i.e.:

$$\sum_{i=0}^{n-1} h_i = 0.$$

This assumption follows if we assume that the channel is linear channel and if we measure the input to the channel electrically before a speaker converts the electrical signal to sound and measure the output electrically after the microphone has converted sound to an electrical signal.

Due to linearity, the resulting output from the input of a sinusoidal waveform is a sinusoidal waveform of the same frequency. Therefore, taking a frequency domain view, we may consider the signal at a frequency of zero separately, ignoring the effect of all other frequencies.

If we transmit a DC signal to a speaker, the speaker position will be displaced slightly forward or backward from its disconnected position. The result may be considered to be equivalent to a disconnected one that has been physically located slightly forward or backward. But we know that a disconnected speaker does not transmit acoustic energy. Therefore, the response to energy at a frequency of zero is zero, and we may say that the transfer function of an acoustic channels at a frequency of zero is zero. Accordingly, our autocovariance matrices in the frequency domain have values of zero in the positions where one of the indices are zero.

7.1.4 Reqirements in the Frequency Domain

Our covariance matrix, $\mathbf{C}_h$, is real, symmetric, and non-negative definite in the time domain. As a covariance matrix, $\mathbf{C}_{\hat{h}}$ is Hermitian in the sense that

$$\mathbf{C}_{\hat{h}}(j,k) = \mathbf{C}^*_{\hat{h}}(k,j).$$

The power spectral densities (which determine coloredness) appear on the main diagonal. Because $\mathbf{C}_h$ is real and symmetric, $\mathbf{C}_{\hat{h}}$ is subject to a second axis of symmetry along the other diagonal if we ignore the elements corresponding to a frequency of zero, i.e.

$$\mathbf{C}_{\hat{h}}(n-j+k,j) = \mathbf{C}_{\hat{h}}(n-j,j-k).$$

We need to be able to specify the power spectral densities (and hence the coloredness) in the frequency domain, and it has to be done in a way that preserves the symmetries, and non-negative definiteness.

In Subsection 7.1.2 we used a two parameter model to give the diagonal for the time domain. In a similar way, it would be possible for us to define the diagonal in the frequency domain using a 2 or 3 parameter model. Therefore we can represent $\mathbf{C}_h$ using only 5 parameters regardless of the value of n.

7.1.5 A Model which is Realistic and Parsimonious

Channel variation will not, in general, exhibit the property that successive values of $\mathbf{h}$, or of $\hat{\mathbf{h}}$ are independent. Let us now outline how to construct the channel covariance matrices, $\mathbf{C}_h$ and $\mathbf{C}_{\hat{h}}$, which are consistent with each other and have specified diagonal terms.

Range transformations

It is useful to consider modifications of a channel by range transformations. For example, if we start with a channel represented using an impulse response vector $\mathbf{g}$, we can define a new channel $\mathbf{h}$ using element by element multiplication with a vector $\mathbf{f}$ either

in the time domain, $\mathbf{h} = \mathbf{f} \diamond \mathbf{g}$, or in the frequency domain, $\tilde{\mathbf{h}} = \tilde{\mathbf{f}} \diamond \tilde{\mathbf{g}}$. The class of all range transformation is not exhausted by these cases, but these are sufficient for us.

It is useful to observe that $\mathbf{C}_g$, being the autocovariance matrix of $\mathbf{g}$, must be non-negative definite. The vector $\mathbf{h}$, which is a function of $\mathbf{g}$, is a statistical process in its own right. Therefore, $\mathbf{C}_h$ is also non-negative definite and the range transformation preserves non-negative definiteness.

We can define range transformations upon the autocovariance matrix in the time domain and infer the effect in the frequency domain and vice versa. For example, if $\tilde{\mathbf{f}}$ is the function defining a range transformation in the frequency domain,

$$\begin{aligned} \mathbf{C}_{\tilde{h}} &= E[\tilde{\mathbf{h}}\tilde{\mathbf{h}}^*] \\ &= E[\tilde{\mathbf{f}} \diamond \tilde{\mathbf{g}}(\tilde{\mathbf{f}} \diamond \tilde{\mathbf{g}})^*] \\ &= \text{Diag}(\tilde{\mathbf{f}})\mathbf{C}_{\tilde{g}}\text{Diag}(\tilde{\mathbf{f}})^* \end{aligned} \tag{7.1}$$

and so

$$\mathbf{C}_h = \mathbf{F}^*\text{Diag}(\tilde{\mathbf{f}})\mathbf{F}\mathbf{C}_g\mathbf{F}^*\text{Diag}(\tilde{\mathbf{f}})^*\mathbf{F}. \tag{7.2}$$

Using the matrix identity (A.16),

$$\frac{1}{\sqrt{n}}\text{Circ}(\mathbf{f}) = \mathbf{F}^*\text{Diag}(\tilde{\mathbf{f}})\mathbf{F},$$

where $\mathbf{f} = \mathbf{F}^* \tilde{\mathbf{f}}$, we may apply it to (7.2), giving

$$\mathbf{C}_h = \frac{1}{\sqrt{n}}\text{Circ}(\mathbf{f})\ \mathbf{C}_g\ \frac{1}{\sqrt{n}}\text{Circ}(\mathbf{f})^*. \tag{7.3}$$

Method for Creating a Channel AutoCovariance Matrix

When creating a channel autocovariance matrices, we need to be able to specify the diagonal of the matrix in the time domain (as discussed in Subsection 7.1.2) and we need to be able to specify the power spectral densities (as discussed in Subsection 7.1.4). Our method permits us to specify these things in a way that is both interesting and realistic. Given the n non-negative values of the desired time domain diagonal, $\mathbf{q}$, and the a set of $\frac{n}{2}$ power spectral densities, $\mathbf{p}$, we may produce the corresponding matrix C_h as follows:

1. Create a vector $\hat{\mathbf{f}}$ of length n (a range transformation) with the correct DFT symmetry from $\mathbf{p}$:

$$\hat{\mathbf{f}}_j = \begin{cases} 0 & j = 0 \\ \sqrt{\mathbf{p}_j} & 0 < j \leq \frac{n}{2}, \\ \sqrt{\mathbf{p}_{n-j}} & \frac{n}{2} < j < n \end{cases}$$

2. Create a diagonal non-negative autocovariance matrix $\mathbf{C}_g$. The formula for $\mathbf{C}_g$ will be derived below. Since $\mathbf{C}_g$ is diagonal and non negative, it is non-negative definite.

3. Create the frequency domain autocovariance matrix corresponding to $\mathbf{C}_g$ using:

$$\mathbf{C}_{\hat{g}} = \mathbf{F}\mathbf{C}_g\mathbf{F}^*.$$

   Since $\mathbf{C}_g$ is diagonal, $\mathbf{C}_{\hat{g}}$ is circulant, as shown in Subsection A.3.4. Therefore $\mathbf{C}_{\hat{g}}$ has a constant diagonal, implying that it is a white autocovariance matrix. Since the frequency domain representation of a non-negative definite matrix in the time domain is also non-negative definite, $\mathbf{C}_{\hat{g}}$ is non-negative definite.

4. Impose the power spectral densities by applying our range transform (and hence setting the desired coloredness) using:

$$\mathbf{C}_{\hat{h}} = \mathrm{diag}(\hat{\mathbf{f}})\mathbf{C}_{\hat{g}}\mathrm{diag}(\hat{\mathbf{f}})^*.$$

   The matrix $\mathbf{C}_{\hat{h}}$ is non-negative definite because the range transformation preserves non-negative definiteness.

5. Convert the result into the time domain:

$$\mathbf{C}_h = \mathbf{F}^*\mathbf{C}_{\hat{h}}\mathbf{F}.$$

   $\mathbf{C}_h$ is non-negative definite because the time domain representation of a non-negative definite matrix in the frequency domain is non-negative definite.

As promised in step 2 above, we shall now find the diagonal matrix $\mathbf{C}_g$. We can characterise $\mathbf{C}_h$ by applying the operator $\mathrm{diag}(\cdot)$ to both sides of equation (7.3), giving:

$$n\,\mathrm{diag}(\mathbf{C}_h) = \mathrm{diag}(\mathrm{Circ}(\mathbf{f})\ \mathbf{C}_g\ \mathrm{Circ}(\mathbf{f})^*). \tag{7.4}$$

Applying (A.2) to (7.4) gives:

$$n \operatorname{diag}(\mathbf{C}_h) = (\operatorname{Circ}(\mathbf{f}) \diamond \operatorname{Circ}(\mathbf{f})) \operatorname{diag}(\mathbf{C}_g).$$

Re-arranging the above to make $\operatorname{diag}(\mathbf{C}_g)$ the subject of the equation, gives us the diagonal of our diagonal matrix $\mathbf{C}_g$:

$$\operatorname{diag}(\mathbf{C}_g) = n \operatorname{Circ}(\mathbf{f} \diamond \mathbf{f})^{-1} \operatorname{diag}(\mathbf{C}_h).$$

7.2 Noise Autocovariance Matrices

The noise $z(t)$ is assumed to be stationary. Therefore, the autocovariance matrix is circulant in the time domain and diagonal in the frequency domain, as shown in Subsection A.3.4. Hence, we need only specify the diagonal of the matrix $\mathbf{C}_z$ in the frequency domain.

Because a matrix in the time domain must be real, a symmetry is imposed in the frequency domain. Referring to the diagonal elements using the vector $\mathbf{p}$, we have

$$\mathbf{p}_j = \mathbf{p}_{n-j}, \qquad j = 1, ..., \frac{n}{2} - 1.$$

The off diagonal elements of $\mathbf{C}_{\hat{z}}$ are set to zero. For the reasons outlined in Subsection 7.1.3, we set $\mathbf{p}_0$ to zero. The diagonal elements of the matrix in the frequency domain are assigned with the power spectral densities of the noise, subject to the required symmetry. The result is converted to a circulant matrix in the time domain using the DFT.

7.3 Chapter Summary

We developed simplified models of channel and noise autocovariance matrices, which were needed in order to prepare test data for the test signal optimisation algorithms developed in Chapter 6.

The models in the form of the autocovariance matrices $\mathbf{C}_h$ and $\mathbf{C}_z$ are fully general, but they have so many parameters that they are too complex to use in practice. The

models proposed in this chapter are quite simple in comparison to the full second order models, and yet they offer the required features in both the time domain and the frequency domain.

Also, it was observed that since a realistic and parsimonious statistical model of channel variation has been identified, efficient estimation of such a model from data from real channels is readily achievable.

Chapter 8

Conclusions and Future Work

Channel estimation is very important in acoustic applications. Echo cancellation, which is used in mobile phone handsets, hands free telephones, videoconferencing and hearing aids, relies on estimation of the channel. This dissertation has made a contribution in the simulation of echo cancellation and has also devised an echo cancellation technique that works when the channel has an initial non-linearity.

Channel estimation is also very important in wireless communication techniques, including OFDM, which is used in most modern communication technologies including mobile phones, ASDL and digital wireless broadcasting. Channel estimation techniques in OFDM employ embedded test signals and use the secondary statistics of the channel to improve the channel estimation.

This dissertation has used ideas from channel estimation in wireless communications and applied them to the estimation of acoustic channels. It has used the secondary statistics of the channel and of the noise to improve the performance of acoustic channel estimation using test signals.

8.1 Echo Cancellation

Echo cancellation, which was explored in Chapter 2, *Echo Cancellation*, is one of the most important applications of acoustic channel estimation. The material explored in Chapter 2 provides a foundation for the remainder of the dissertation.

Echo Cancellation is the most challenging and technically sophisticated of the echo management techniques because it requires estimation of the echo channel. Low tech solutions to problems are often the most successful, but in the case of echo management, echo cancellation, which is a high tech solution involving mathematical algorithms for estimation of the channel and requiring cheap and powerful digital technology, has proved the most successful.

Video conferencing is one of several applications where echo cancellation is required in order to achieve a quality result. Ideally the sound from the second party must be made loud enough to be heard clearly without the use of headphones. Yet we wish only the sound that actually originated from the room to be transmitted to back to the second party. Modern echo cancellation is used for videoconferencing because it provides a superior result. The use of comb filters would induce unnatural distortion of the sound and the use of echo suppression would be an impediment to natural conversations.

8.1.1 ECOOSE

Echo cancellation is a field where new algorithms are constantly emerging. New algorithms are generally claimed by their authors to have superior convergence properties and superior computational complexity. These properties make the algorithms potential candidates for implementation. Implementers, on the other hand, should maintain a certain degree of healthy scepticism. Algorithms need to be properly evaluation and their peformance under a range of situations needs to be compared with the performance of other algorithms before the process of implementation in firmware is undertaken. ECOOSE provides a ready made framework for echo cancellation simulation that is flexible enough to permit the exploration of the algorithm's performance under a diverse range of echo cancellation situations.

Once one appreciates the object oriented framework approach used in ECOOSE, it is relatively easy to simulate and plot the performance of a new echo cancellation algorithm using ECOOSE. For example, in order to provide a new linear echo cancellation algorithm which may be used as part of ECOOSE, one need only create a new subclass of the linear algorithm class, and provide the methods that are directly related to the new algorithm.

ECOOSE and its evolution is described in Chapter 3, *Echo Cancellation Simulation*. ECOOSE has already been used to develop and evaluate the HLMS algorithm, as documented in Chapter 4.

Future Work

ECOOSE will be used to simulate, evaluate and compare emerging echo cancellation algorithms. The most recent generation of PNLMS algorithms are of particular interest. The algorithms of (Dyba & Deng 2009) and of (das Chagas de Souza et al. 2010) manage to achieve impressive convergence rates for sparse impulse responses with less computational complexity than the NLMS algorithm, according to the papers in which the algorithms were published. Independent verification of their performance would be highly desirable, and ECOOSE would be a suitable platform for their evaluation.

8.1.2 Framework for Echo Cancellation with Non Linearities

In Chapter 4, *Framework for Echo Cancellation with Non Linearities*, the Method of Innovations was applied to the case of Hammerstein channels and the resulting set of formulas, dubbed the Hammerstein LMS (HLMS) algorithm was implemented in ECOOSE. The HLMS algorithm and the NLMS algorithm converge equally well on linear channels, but on Hammerstein channels the NLMS algorithm cancels only some of the far end signal, while the HLMS algorithm is able to completely cancel it.

This work has shown the that the Method of Innovations may be applied to the problem of echo cancellation. It has demonstrated that the Method of Innovations may be used to create echo cancellation algorithms for models that incorporate memoryless non-

linearities.

Future Work

It can be expected that the method can derive an algorithm that applies when the memoryless non-linearity is at the end of a linear channel, or possibly both at the beginning and the end.

The HLMS algorithm may potentially prove useful in the area of active noise control, where Hammerstein channels arise when speakers used to cancel engine or transformer noise have nonlinearities which often causes them to produce even harmonics of the base frequency that the speakers are attempting to cancel.

8.2 Using the Statistics of the Channel and of the Noise

Whilst there are very few echo cancellation algorithms that utilise the auto-covariance of the noise or the auto-covariance of the channel, there are numerous acoustic echo cancellation algorithms which utilise the noise power and they generally assume that the noise power, σ_z^2, can be estimated during periods when the excitation $x(t)$ is silent (Wei et al. 2001, Benesty et al. 2006, Costa & Bermudez 2007, Imam, Zerguine & Deriche 2007, Zipf, Tobias & Seara 2008, Imam, Zerguine & Zerguine 2010). Clearly, the autocovariance of the noise, $\mathbf{C}_z$, may also be estimated during periods when the excitation, $x(t)$, is silent. Since successive channel estimates are continuously obtained at a high rate, the first step towards obtaining the data required to estimate second order channel statistics has already been taken.

Several schemes for OFDM make use of the second order statistics of the channel to improve the estimation of the channel (Edfors et al. 1995, Negi & Cioffi 1998, Li 2000, Srivastava, Ho, Fung & Sun 2004, Lee et al. 2006, Mehlfuhrer, Caban & Rupp 2008). We have explained the essential background information relevant to estimation of the wireless channel in OFDM and explored some of the more important schemes in Chapter 5, *Lessons From the Wireless World*.

8.2.1 Optimisation of Test Signals

In OFDM, pilot test signals are incorporated as part of the transmitted signal. OFDM receivers use those test signals in order to make an estimate of the channel, as described in Subsection 5.1.3. In the acoustic world, test signals may be used in difficult acoustic situations such as hearing aids (Long & Ling 1991, Kates 1991, Engebretson & French-St.George 1993, Spriet et al. 2005), as described in Subsection 1.2.1. The use of test signals may also provide a solution to the difficulties encountered in remote auscultation described in Subsection 1.2.1.

In Chapter 6, *Optimisation of Test Signals*, we have shown that the optimal test signal may be described in terms of the strengths in each frequency, i.e. when we assume the channel to be linear, the phases do not effect the expectation of the error of estimation. The phases are often important in practice, however. Transmission and receiving equipment, for example, are often linear only within certain limits. Given these limits, the amount of test signal power that can be transmitted is determined by the crest factor of the test signal, which is a function of the phases. We have reviewed existing optimisation techniques for improving the crest factor of a test signal in Chapter 5.

While literature exists for the optimisation of the crest factor of a test signal, no such literature exists on choosing the strengths of each frequency given the second order statistics of the noise and the second order statistics of the channel. In Chapter 6, *Optimisation of Test Signals*, we have developed prototype optimisation techniques which estimate the optimal test signal given those second order statistics. We also developed an optimal estimator for a given signal, given those same secondary statistics that is potentially useful in its own right.

Our work with optimal test signals demonstrate that the optimal test signal is not white whenever the noise is colored or the history of the channel is colored. We took the commonly accepted concept of colour in sound, which is analogous to that of colour in the visible light spectrum, and defined a measure called *coloredness* which may be applied to the near end noise and to the channel history. We showed that the amount that the optimal estimator and test signal can be improved is potentially unbounded, and depends on the coloredness and of the noise and of the channel history.

Comparison with Other Methods of Channel Estimation

It is difficult to directly compare our technique that uses test signals with channel estimation techniques such as LMS and RLS as these have different modes of operation. These methods use the operational signal for excitation, and continuously update their estimate of the channel, whereas our method uses a fixed test signal, and updates its estimate of the channel at intervals dictated by the effective length of the channel. We are currently endeavoring to make that comparison for a hypothetical channel where the history of the channel and the noise are known, by using a block based version of the RLS algorithm excited by a white noise test signal, and compare results against those using the optimal test signal and the optimal estimator.

In some instances, such as the earphone adaption systems described in (Kates 1991, Engebretson & French-St.George 1993, Goodings et al. 1993, Spriet et al. 2005), a test signal is already being used, and a more efficient test signal would be highly desirable. If the noise or the channel is colored, then this technique will produce a more efficient test signal.

Future Work

One future endeavour would be to derive an analytical expression that generates the optimal test from the second order statistics. It may be possible to tackle this problem by formulating it as an optimisation problem in the Hilbert space of stochastic channels, measurements based on such channels, and estimates of channels based on such measurements. Comparisons with results obtained from our existing optimisation routines will provide validation of our results. It is expected that the use of such an analytical expression would yield an answer in a small fraction of the time required by numerical methods.

Our optimisation finds the best test signal for a given amount of RMS power output, which implies a multidimensional sphere as our constraint, and the subsequent use of crest factor optimisation using the spectral densities given by our optimisation is an option. However, we are aware that equipment often limits the peak output power, and

clearly an optimisation that used peak power as the constraint would be a worthwhile endeavour.

8.2.2 Simplified Models for Impulse Response and Noise Covariances

In order to thoroughly test our optimisation methods, we used an ensemble of randomly generated scenarios and observed the means and the standard deviations of the performance of the optimised channels. The test signal optimisation techniques were tested using autocovariance matrices generated using pseudo-random numbers. Although the covariance matrices are fully general, they are too complex in practice to assign without having a model that enables their creation. The channel covariance matrices needed to be symmetrical, to be positive definite, to have an arbitrarily specified profile along the diagonal in the time domain and to have arbitrarily specified power spectral densities in the frequency domain.

In Chapter 7, *Statistical Model of Channel Variation*, we developed a model and a technique for the creation of a channel autocovariance matrix by specifying the diagonal in the time domain and the diagonal in the frequency domain. These diagonals, in turn, may be specified by simpler models, as shown in Chapter 7. This model may also provide a basis for efficient statistical estimation of the channel autocovariance.

We made the assumption of stationarity for the noise, which means that the noise covariance matrix was diagonal in the frequency domain. Thus, we needed only to assign the diagonal of the noise covariance in the frequency domain, and convert it to the time domain.

Future Work

The development of efficient algorithms for estimating the autocovariance matrix of a channel remains as future work. A statistical model of channel variation with a small number of parameters has been found, and the maximum likelihood procedure may be used to develop an efficient estimator using this model.

8.3 Chapter Summary

Wireless channel estimation is used in most modern wireless applications and is an important and ongoing area of research. Acoustic channel estimation is also very important and is a vital part of the echo cancellation that is found in many applications such as mobile phones, videoconferencing, telephony and earphones.

A flexible framework for the simulation of channel estimation and echo cancellation has been developed. It can simulate echo cancellation using virtually any algorithm including online or block based algorithms and any channel including dynamic and non-linear channels.

It has been demonstrated that the method of innovations can be used to create channel estimation techniques for non-linear channels. An algorithm was developed and testing has shown that it can perform echo cancellation for non-linear channels where linear echo cancellation fails.

Bayesian methods that utilise the secondary statistics of the noise and the channel have been developed for channel estimation. An optimal estimator for a given test signal has been derived. Efficient methods for finding the optimal test signal have been devised. The methods have been tested on an ensemble of randomly generated scenarios. A definition of coloredness has been proposed, and it has been shown that the developed methods easily outperform non-Bayesian methods that do not take the statistics of the channel and the noise into account whenever the noise or the channel autocovariance become more colored.

A model of statistical channel variation has been developed that permits creation of an autocovariance matrix by specifying the diagonal of the matrix in the time domain and in the frequency domain, or by the use of a simpler model that specifies these diagonals. This model was needed in order to randomly generate channel scenarios for the purpose of testing the optimisation of test signals. This model can also provide a parsimonious model which enables efficient estimation of the statistics of channel variation.

References

Agilent Technologies (2001), 'Digital modulation in communications systems an introduction', http://cp.literature.agilent.com/litweb/pdf/5965-7160E.pdf.

Aldrich, J. (1998), 'Doing least squares: Perspectives from gauss and yule', *International Statistical Review* **66**(1), 61–81.

Allen, J. & Berkley, D. (1978), 'Image method for efficiently simulating small-room acoustics', *The Journal of the Acoustical Society of America* **65**(4), 943–950.

Anuff, A. & Liou, M. (1971), 'A note on necessary bandwidth in FM systems', *Proceedings of the IEEE* **59**(10), 1522–1523.

Arbel, Y. (1989), 'Half-duplex speakerphone', US Patent Application 4879745.

Armstrong, E. (1930), 'Radio signalling system', US Patent Application 1941066.

Barney, H. (1941), 'Control of transmission in two-way signal wave transmission systems', US Patent Application 2251028.

Beck, K. & Johnson, R. (1994), 'Patterns generate architectures', *Proceedings of the European Conference on Object-Oriented Programming* pp. 139–149.

Belforte, G. & Gay, P. (1999), 'Hammerstein model identification with set membership errors', *IEEE Conference on Decision and Control, 1999* **1**(1), 592–597.

Benallal, A. & Benkrid, A. (2007), 'A simplified FTF type algorithm for adaptive filtering', *Signal Processing* **87**(5), 904–917.

Benallal, A. & Gilloire, A. (1988), 'A new method to stabilize fast RLS algorithms

in transversal adaptive filters', *International Conference on Acoustics Speech and Signal Processing* pp. 1373–1376.

Benesty, J., Gansler, T., Morgan, D., Sondhi, M. & Gay, S. (2001), *Advances in Network and Acoustic Echo Cancellation*, Springer.

Benesty, J. & Gay, S. (2002), 'An improved PNLMS algorithm', *IEEE International Conference on Acoustics, Speech, and Signal Processing* **2**(1), 1881–1884.

Benesty, J., Morgan, D. & Cho, J. (2000), 'A new class of doubletalk detectors based on cross-correlation', *IEEE Transactions on Speech and Audio Processing* **8**(2), 168–172.

Benesty, J., Rey, H., Vega, L. & Tressens, S. (2006), 'A nonparametric VSS NLMS algorithm', *IEEE Signal Processing Letters* **13**(10), 581–584.

Berkley, D. & Courtney-Pratt, J. (1971), 'Telephonic transmission using complementary comb filters', US Patent Application 3622714.

Berkley, D., Mitchell, O. & Pierce, J. (1972), 'Speech processor using multiband controlled center clipping', US Patent Application 3699271.

Blackwell, O. (1932), 'Time factor in telephone transmissions', *Transactions of the American Institute of Electrical Engineers* **51**(1), 141–147.

Bode, H. & Moog, R. (1972), 'A high-accuracy frequency shifter for professional audio applications', *Journal of the Audio Engineering Society* **20**(6), 453–458.

Botto, J. & Moustakides, G. (1989), 'Stabilising the fast kalman algorithm', *IEEE Transactions on Acoustic and Speech Signal Processing* **37**(9), 1342–1348.

Boute, R. (1992), 'The euclidean definition of the functions div and mod', *ACM Transactions on Programming Languages and Systems* **14**(2), 127–144.

Brady, P. (1967), 'Echo suppressor for communication system having transmission delay', US Patent Application 3351720.

Braithwaite, S. (2006), 'Draft problem definition for remote digital stethoscope', `http://www.sci.usq.edu.au/projects/telethascope`. [Online: accessed April 2011].

Braithwaite, S. & Addie, R. (2008), 'Ecoose home page', `http://www.sci.usq.edu.au/projects/ecoose`.

Braithwaite, S. & Addie, R. (2010), 'Testsig archive', `http://www.sci.usq.edu.au/projects/optTstSig/optsig.tgz`.

Cabell, R. (1998), A Principal Component Algorithm for Feedforward Active Noise and Vibration Control, PhD thesis, Virginia Polytechnic Institute and State University.

Carayannis, G., Manolakis, D. & Kalouptsidis, N. (1983), 'A fast sequential algorithm for least squares filtering and prediction', *IEEE Transactions on Signal Processing* **31**(6), 1394–1402.

Chang, R. (1966), 'Synthesis of band limited orthogonal signals for multichannel data transmission', *Bell System Technical Journal* **45**(10), 1775–1796.

Cho, J., Morgan, D. & Benesty, J. (1999), 'An objective technique for evaluating doubletalk detectors in acoustic echo cancelers', *IEEE Transactions on Speech and Audio Processing* **7**(6), 718–724.

Cioffi, J. & Kailath, T. (1984), 'Fast, recursive-least-squares transversal filters for adaptive filtering', *IEEE Transactions on Acoustics, Speech and Signal Processing* **32**(2), 304–337.

Cooley, J. & Tukey, J. (1965), 'An algorithm for the machine calculation of complex fourier series', *Mathematics of Computation* **19**(90), 297–301.

Costa, M. & Bermudez, J. (2007), 'A noise resilient variable step-size lms algorithm', *Signal Processing* **88**(3), 733–748.

Costa, M. & Mermudez, J. (2008), 'A noise resilient variable step size LMS algorithm', *Signal Processing* **88**(3), 733–748.

Cutler, C. C. (1965), 'Echo suppressor using comb filters', US Patent Application 3175051.

das Chagas de Souza, F., Tobias, O., Seara, R. & Morgan, D. (2010), 'A PNLMS algorithm with individual activation factors', *IEEE Transactions on Signal Processing* **58**(4), 2036–2047.

Davidson, G. & Falconer, D. (1991), 'Reduced complexity echo cancellation using orthonormal functions', *IEEE Transactions on Circuits and Systems* **38**(1), 20–28.

Deeba, A. & Wood, S. (1990), 'Convergence rate improvements for frequency domain implementations of lms adaptive filters', *Conference Record Twenty-Fourth Asilomar Conference on Signals, Systems and Computers* **2**(1), 754.

Delaney, F. (1958), 'Data phase coding system using parallel pulse injection in binary divider chain', US Patent Application 2994790.

Deng, H. & Doroslovacki, M. (2006), 'Proportionate adaptive algorithms for network echo cancellation', *IEEE Transactions on Signal Processing* **54**(5), 1794–1803.

Dentino, M., Widrow, B. & McCool, J. (1978), 'Adaptive filtering in the frequency domain', *Proceedings of the IEEE* **66**(12), 1658–1659.

der Ouderaa, E. V., Schoukens, J. & Renneboog, J. (1988), 'Peak factor minimization of input and output signals of linear systems', *IEEE Transactions on Instrumentation and Measurement* **37**(2), 207–212.

der Perre, L. V., Ness, R., Thoen, S., Bandenameele, P. & Engels, M. (2002), *Wireless OFDM Systems - How to make them work*, Kluwer, chapter Ch 2.

Doelz, M. (1949), 'Polar communication system', US Patent Application 2676245.

Doelz, M. & Heald, E. (1958), 'Minimum shift data communication system', US Patent Application 2977417.

Douglas, S. & Rupp, M. (1998), *The Digital Signal Processing Handbook*, IEEE Press, chapter 19.

Duttweiler, D. (2000), 'Proportionate normalized least-mean-squares adaptation in echo cancelers', *IEEE Transactions on Speech and Audio Processing* **8**(5), 508–518.

Dyba, R. & Deng, H. (2009), 'Partial update pnlms algorithm for network echo cancellation', *IEEE International Conference on Acoustics, Speech and Signal Processing* pp. 1329–1332.

Edfors, O., Sandell, M., deBeek, J., Landstrom, D. & Sjoberg, F. (1998), 'An introduction to orthogonal frequency division multiplexing'.

Edfors, O., Sandell, M., deBeek, J., Wilson, S. & Borjesson, P. (1995), 'On channel estimation in ofdm systems', *IEEE Conference on Vehicular Technology* **2**, 815–819.

Edfors, O., Sandell, M., deBeek, J., Wilson, S. & Borjesson, P. (1998), 'OFDM channel estimation by singular value decomposition', *IEEE Transactions on Communications* **47**(7), 931–939.

Emling, J. & Mitchel, D. (1963), 'The effects of time delay and echoes on telephone conversations', *Bell Systems Telephone Journal* **22**, 2869–2891.

Engebretson, A. & French-St.George, M. (1993), 'Properties of an adaptive feedback equalization algorithm', *Journal of Rehabilitation Research and Development* **30**(1), 8–16.

Excellence in Queensland Health (2008), Queensland Health Website.

Fabre, P. & Gueguen, C. (i1986), 'Improvement of the fast recursive least-squares algorithms via normalisation: A comparative study', *IEEE Transactions on Acoustics, Speech and Signal Processing* **34**(2), 296–308.

Faller, C. (2003), 'Perceptually motivated low complexity acoustic echo control', *Audio Engineering Society Convention Paper* .

Faraci, P. & Skene, J. (1999), 'Compact speakerphone apparatus', US Patent Application 5896461.

Felhauer, T., Baier, P., Konig, W. & Mohr, W. (1993), 'Optimum spread spectrum signals for wideband channel sounding', *Electronics Letters* **29**(6), 563–564.

Foti, D. (2008), 'Inside matlab objects in r2008a', *Matlab Digest* .

Freed, D. & Soli, S. (2009), 'Frequency shifter for use in adaptive feedback cancellers for hearing aids', US Patent Application 7609841.

Freed, E. (2004), 'Full duplex speakerphone', US Patent Application 20040240664.

Froehlich, G., Hoover, H. J., Liu, L. & Sorenson, P. (1997), 'Hooking into object-oriented application frameworks', *Proceedings of 19th International Conference on Software Engineering* pp. 491–501.

Furht, B. & Ahson, S. (2009), *Long Term Evolution: 3GPP LTE Radio and Cellular Technology*, Auerbach.

Gamma, E., Helm, R., Johnson, R. & Vlissides, J. (1994), *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison Wesley.

Gay, S. (1998), 'An efficient, fast converging adaptive filter for network echo cancellation', *Asilomar Conference on Signals, Systems and Computers* pp. 394–398.

Gay, S. & Mammone, R. (1990), 'Fast converging subband acoustic echo cancellation using rap on the we dsp16a', *International Conference on Accoustics, Speech, and Signal Processing* pp. 1141–1144.

Geigel, A. & LaMarche, R. (1975), 'Common control digital echo suppressor', US Patent Application 4029912.

Gilloire, A. (1987), 'Experiments with sub-band echo cancellers for teleconferencing', *International Conference on Accoustics, Speach and Signal Processing* pp. 2141–2144.

Gilloire, A. (1988), 'Adaptive filtering in sub-bands', *International Conference on Accoustics, Speach and Signal Processing* pp. 1572–1575.

Godard, D. (1974), 'Channel equalization using a kalman filter for fast data transmission', *IBM Journal of Research and Development* **18**(3), 267–273.

Goldstein, M. & Brown, P. (1978), 'Prediction with shrinkage estimators', *Statistics, A Journal of Theoretical and Applied Statistics* **9**(1), 3–7.

Goodings, R., Senebsieb, G., Gideon, A., Wilson, P. & Hanson, R. (1993), 'Hearing aid having compensation for acoustic feedback', US Patent Application 5259033.

Gordy, J. & Goubran, R. (2006), 'Fast system identification using affine projection and a critically sampled subband adaptive filter', *IEEE Transactions on Instrumentation and Measurement* **55**(4), 1242–1249.

Gould, R. (1964), 'Echo suppression and time delay, a subjective evaluation', *IEEE Transactions on Communication Technology* **12**(3), 74–82.

Grant, S. & Gay, S. (2004*a*), 'A mutiple principal components based adaptive filter', *Asilomar Conference on Signals, Systems and Computers* **1**, 945–949.

Grant, S. & Gay, S. (2004*b*), 'A solution space principal component based adaptive filter', *Proceedings of the IEEE International Conference on Acoustics, Speach and Signal Processing* .

Gritton, C. & D.W.Lin (1984), 'Echo cancellation algorithms', *IEEE ASSP Magazine* **1**(2), 30–38.

Guillaume, P., Schoukens, J., Pintelon, R. & Kollar, I. (1991), 'Crest-factor minimization using nonlinear chebyshev approximation methods', *IEEE Transactions on Instrumentation and Measurement* **40**(6), 982–989.

G.V.Moustakides & S.Theodoridis (1991), 'Fast newton transversal filters - a new class of adaptive estimation algorithms', *IEEE Transactions on Signal Processing* **39**(10), 2184–2193.

Hager, W. (1989), 'Updating the inverse of a matrix', *SIAM Review* **31**(2), 221–239.

Hansler, E. & Schmidt, G. (2004), *Acoustic Echo and Noise Control a Practical Approach*, Wiley.

Heideman, M., Johnson, D. & Burrus, C. (1985), *Gauss and the history of the fast Fourier transform*, Springer.

Hilbert, D. & Courant, R. (1924), *Methoden der Mathematischen Physik*, Vol. 1, Springer, chapter 2.

Huang, Y. & Benesty, J. (2004), *Audio Signal Processing for Next-Generation Multimedia Communication Systems*, Springer.

Imam, S., Zerguine, A. & Deriche, M. (2007), 'A noise constrained least mean fourth adaptive algorithm', *IEEE International Conference on Signal Processing and Communications* pp. 951–954.

Imam, S., Zerguine, A. & Zerguine, A. (2010), 'A noise constrained least mean fourth adaptive algorithm', *Signal Processing* **91**(1), 951–954.

Johnson, R. E. & Foote, B. (1988), 'Designing reuseable classes', *Journal of Object-Oriented Programming* **1**(2), 22–35.

Johnstone, J. (1980), 'A filter family designed for use in quadrature mirror filter banks', *IEEE International Conference on Acoustics, Speech, and Signal Processing* **5**, 291–294.

Jones, D. (2010), 'Adaptive Echo Cancellation', `http://cnx.org/content/m11909/latest`. [Online: accessed April 2011].

Kabal, P. (2006), 'Wave or riff wave sound file', `http://www-mmsp.ece.mcgill.ca/Documents/AudioFormats/WAVE/WAVE.html`. [Online: accessed April 2011].

Kailath, T. (1981), *Lectures on Wiener and Kalman Filtering*, Springer-Verlag, New York.

Kalman, R. (1960), 'A new approach to linear filtering and prediction problems', *Transactions of the ASME-Journal of Basic Engineering* **82**(Series D), 35–45.

Kates, J. (1991), 'Feedback cancellation in hearing aids: results from a computer simulation', *IEEE Transactions on Signal Processing* **39**(3), 553–562.

Kelly, J. & Logan, B. (1966), 'Self adaptive echo canceller', US Patent Application 3500000.

Kumar, A., Singh, G. K. & Anand, R. S. (2008), 'Near perfect reconstruction quadrature mirror filter', *International Journal of Computer Science and Engineering* .

Lee, S., Park, N., Cho, C., Lee, H. & Ryu, S. (2006), 'The wireless broadband (wibro) system for broadband wireless internet services', *IEEE Communications Magazine* **44**(7), 106–112.

Lehmann, E. (2009), 'Fast simulation of acoustic room impulse responses (image-source method)', `http://www.mathworks.com/matlabcentral/fileexchange/25965-fast-simulation-of-acoustic-room-impulse-responses-image-source-method`. [Online: accessed April 2011].

Lewellyn, F. (1950), Proof that a self-balancing hybrid cannot be constructed with linear networks, Internal document, Bell Telephone Laboratories.

Li, Y. (2000), 'Pilot-symbol-aided channel estimation for OFDM in wireless systems', *IEEE Transactions on Vehicular Technology* **49**(4), 1207–1215.

Li, Y., Cimini, L. & Sollenberger, N. (1998), 'Robust channel estimation for ofdm systems with rapid dispersive fading channels', *IEEE Transactions on Communications,* **46**(7), 902–915.

Li, Y. & Stuber, G. (2006), *Orthogonal Frequency Division Multiplexing for Wireless Communications*, Springer, chapter Ch 5.

Lin, D. (1984), 'On digital implementation of the fast kalman algorithm', *IEEE Transactions on Acoustics, Speech and Signal Processing* **32**(5), 998–1005.

Ljung, L., Morf, M. & Falconer, D. (1978), 'Fast calculation of gain matrices for recursive estimation schemes', *International Journal of Control* **27**(1), 1–19.

Ljung, S. (1983), 'Fast algorithms for integral equations and least squares identification problems', PhD Dissertation Linkoping University Sweden.

Ljung, S. & Ljung, L. (1985), 'Error propagation properties of recursive ls adaptation algorithms', *Automatica* **21**(2), 157–167.

Long, F. & Ling, G. (1991), 'Fast training echo canceller', U.S. Patent Application 4987569.

Luenberger, D. (1969), *Optimization by Vector Space Methods*, Wiley.

Mader, A., Puder, H. & Schmidt, G. (2000), 'Step-size control for acoustic echo cancellation filters an overview', *Signal Processing* **80**(8), 1697–1719.

Manchester, I. (2009), 'An algorithm for amplitude constrained input design for system identification', *IEEE Conference on Decision and Control* pp. 1551–1556.

Manolakis, D., Ingle, V. & Kogon, S. (2000), *Statistics and Adaptive Signal Processing*, McGraw-Hill, chapter 10.

Marconi, G. (1901), 'Improvements in apparatus for wireless telegraphy', GB Patent Application 7777.

M.Arezki & D.Berkani (2009), 'Advanced algorithms for adaptive filtering', *Proceedings of the WSEAS International Conference on Signal Processing, Robotics and Automation* .

MathWorks (2008*a*), 'fmincon', `http://www.mathworks.com/access/helpdesk/help/toolbox/optim/ug/fmincon.html`. [Matlab online reference].

Mathworks (2008*b*), 'Matlab classes and object-oriented programming', Matlab 2008 built in help.

Mathworks (2008*c*), 'Read microsoft wave (.wav) sound file - matlab', `http://www.mathworks.com/help/releases/R2008a/techdoc/ref/wavread.html`. [Matlab online reference].

McGarrity, S. (2008), 'Introduction to object-oriented programming in matlab', *Matlab Digest* .

Mehlfuhrer, C., Caban, S. & Rupp, M. (2008), 'An accurate and low complex channel estimator for OFDM WiMAX', *International Symposium on Communications, Control and Signal Processing* pp. 922–926.

Miura, A., Kobayashi, S., Sato, I. & Nagata, K. (1966), 'Performance of trially produced blockless echo suppressor', *Presented at the meeting of the Institute of Electronics and Communication Engineers of Japan* .

Miura, A., Sato, R. & Nagata, K. (1969), 'A blockless echo suppressor', *IEEE Transactions on Communications* **17**(4), 489–495.

Molina, A., Fannin, P. & Timoney, J. (1995), 'Generation of optimum excitation waveforms for mobile radio channel sounding', *IEEE Transactions on Vehicular Technology* **44**(2), 275–279.

Moore, E. (1920), '"on the reciprocal of the general algebraic matrix', *Bulletin of the American Mathematical Society* **26**, 394–395.

Mosier, R. & Clabaugh, R. (1958), 'Kineplex, a bandwidth efficient binary transmission system', *AIEE Trans* **76**, 723–728.

Mueller, M. (1981), 'Least squares algorithms for adaptive equalisers', *The Bell System Technical Journal* **60**, 1905–1925.

Nagumo, J. & Noda, A. (1967), 'A learning method for system identification', *Automatic Control, IEEE Transactions on* **12**(3), 282–287.

Negi, R. & Cioffi, J. (1998), 'Pilot tone selection for channel estimation in a mobile OFDM system', *IEEE Transactions on Consumer Electronics* **44**(3), 1122–1128.

Nguyen, T. & P.P.Vaidyanathan (1990), 'Structures for m-channel perfect-reconstruction fir qmf banks which yield linear-phase analysis filters', *IEEE Transactions on Acoustics, Speech and Signal Processing* **38**(3), 433–446.

Nooriafshar, M. & Maraseni, T. N. (2007), 'Telehealth system in queensland', *Proceedings of the 6th Annual Hawaii International Conference on Statistics, Mathematics and Related Fields,* .

OFDM in Multipath (2007), `http://www.s3.kth.se/signal/project_course/2007/green/FinalReport.pdf`. [Online: accessed April 2011].

Ozeki, K. & Umeda, T. (1984), 'An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties', *Electronics and Communications in Japan* **67-A**(5), 126–132.

Pages, B. (2010), 'Bouml a free uml toolbox', `http://bouml.free.fr`.

Painter, T. & Spanias, A. (1997), 'A review of algorithms for perceptual coding of digital audiosignals', *Digital Signal Processing Proceedings* **1**, 179–208.

Papaodysseus, C., Alexiou, C., Roussopoulos, G. & Panagopoulos, A. (2001), 'Quantization effects and stabilization of the fast-kalman algorithm', *EURASIP Journal on Applied Signal Processing* **1**(3), 169–180.

Parsons, J. & Bajwa, A. (1982), 'Wideband characterisation of fading mobile radio channels', *IEE Proceedings on Communications, Radar and Signal Processing* **129**(2), 95.

Penrose, R. (1955), 'A generalized inverse for matrices', *Proceedings of the Cambridge Philosophical Society* **51**(3), 406–413.

Petersen, K. & Pederson, M. (2008), The matrix cookbook, Technical report.

Phillips, I., Williams, L. & Herbert, L. (1985), 'Loudspeaking telephone instrument.', European Patent Application EP0155739.

Plemmons, R. (1993), 'Fft-based rls in signal processing', *IEEE International Conference on Acoustics, Speech, and Signal Processing* pp. 571–574.

P.P.Mavridis & G.V.Moustakides (1996), 'Simplified newton-type adaptive estimation algorithms', *IEEE Transactions on Signal Processing* **44**(8), 1932–1940.

Pradham, S. & Reddy, V. (1999), 'A new approach to subband adaptive filtering', *IEEE Transactions on Signal Processing* **47**(3), 655–664.

Prasad, R. (2004), *OFDM for Wireless Communications Systems*, Artech House.

Prasad, R. & Velez, F. (2010), *Wimax Networks*, Springer.

Proakis, J. & Salehi, M. (1994), *Communications Systems Engineering*, Prentice Hall.

Radiocommunication Sector (2010), `http://www.itu.int/net/about/itu-r.aspx`. [Online: accessed April 2011].

Rappaport, T. (2001), *Wireless Communications: Principles and Practice*, Prentice Hall.

RCC IRIG 106-99: Telemetry Standards (Appendix A) (2003), `http://www.spiraltechinc.com/otis/irig_files/IRIG_Appendix-A.htm`.

Reid, G. (2002), 'Synthesizing drums', *Sound on Sound* .

Salmen, J., Schlipsing, M. & Igel, C. (2010), 'Efficient update of the covariance matrix inverse in iterated linear discriminant analysis', *Pattern Recognition Letters* **31**(13), 1903–1907.

Sankaran, S. & Beex, A. (2000), 'Convergence behavior of affine projection algorithms', *IEEE Transactions on Signal Processing* **48**(4), 1086–1096.

Scheuing, J. & Yang, B. (2006), 'Frequency shifting for acoustic feedback reduction', *Proceedings of the European DSP Education and Research Symposium* .

Schroeder, M. (1964), 'Improvement of acoustic feedback stability by frequency shifting', *Journal of the Acoustical Society of America* **36**(9), 1718–1724.

Schroeder, M. (1970), 'Synthesis of low-peak-factor signals and binary sequences with low autocorrelation', *IEEE Transactions on Information Theory* **16**(1), 85–89.

Schuldt, C., Lindstrom, F. & Claesson, I. (2008), 'A low-complexity delayless selective subband adaptive filtering algorithm', *IEEE Transactions on Signal Processing* **56**(12), 5840–5850.

Shen, Y. & Martinez, E. (2006), Channel estimation in ofdm systems, Technical report, Freescale Semiconductor.

Slock, D. & Kailath, T. (1988), 'Numerically stable fast rls transversal filters', *International Conference on Acoustics Speech and Signal Processing* pp. 1365–1368.

Sondhi, M. (1966), 'Closed loop adaptive echo canceller using generalized filter networks', US patent 3499999.

Sondhi, M. (1967), 'An adaptive echo canceller', *The Bell System Technical Journal* **46**(3), 497–510.

Spriet, A., Moonen, M. & Wouters, J. (2009), 'Objective evaluation of feedback reduction techniques in hearing aids', *European Signal Processing Conference* pp. 1859–1863.

Spriet, A., Moonen, M. & Wouters, J. (2010), 'Objective evaluation of feedback reduction techniques in hearing aids', *Journal of the Acoustical Society of America* **128**(3), 1245–1261.

Spriet, A., Proudler, I., Moonen, M. & Wouters, J. (2005), 'Adaptive feedback cancellation in hearing aids with linear prediction of the desired signal', *Signal Processing, IEEE Transactions on* **53**(10), 3749–3763.

Srivastava, V., Ho, C., Fung, P. & Sun, S. (2004), 'Robust MMSE channel estimation in OFDM systems with practical timing synchronization', *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE* pp. 711–716.

Stanford Business Software (2011), 'Npsol 6.2', `http://www.sbsi-sol-optimize.com/asp/sol_product_npsol.htm`.

Stapleton, J. (1995), *Linear Statistical Models*, Wiley, chapter 3.

Staudacher, W. & Harbhajan, S. (1997), 'Full duplex speakerphone', US Patent Application 5657384.

Stewart, J. (2006), *Calculus and Concepts*, Thomson Brooks-Cole, chapter ch11.

Tesla, N. (1900), 'System of transmission of electrical energy', US Patent Application 645576.

The Uneven Diffusion of Telemedicine Services In Australia (1998), *Proceedings of TeleMed 98, the Sixth International Conference on Telemedicine and Telecare* .

Tomlab Optimization (2011*a*), 'consolve', http://tomopt.com/tomlab/products/base/solvers/conSolve

Tomlab Optimization (2011*b*), 'consolve', http://tomopt.com/tomlab/about/performance.php.

Troxel, D. (2005), 'Understanding acoustic feedback and suppressors', `http://www.rane.com/note158.html`. [Online; accessed August-2010].

Tunbull, H. & Aitken, A. (1932), *An Introduction to the Theory of Canonical Matrices*, Blackie and Sons.

Van den Bos, A. (1987), 'A new method for synthesis of low-peak-factor signals', *IEEE Transactions on Acoustics, Speech and Signal Processing* **35**(1), 120–122.

van Houwelingen, J. (2001), 'Shrinkage and penalized likelihood as methods to improve predictive accuracy', *Statistica Neerlandica* **55**(1), 17–34.

van Waterschoot, T. & Moonen, M. (2009), 'Adaptive feedback cancellation for audio applications', *Signal Processing* **89**, 2185–2201.

van Waterschoot, T. & Moonen, M. (2011), 'Fifty years of acoustic feedback control: State of the art and future challenges', *Proceedings of the IEEE* **99**(2), 288–327.

Vaseghi, S. (1996), *Advanced Signal Processing and digital noise reduction*, Wiley, chapter 13.

Vaughan, R. & Anderson, J. (2003), *Channels, Propagation and Antennas for Mobile Communications*, IEEE, chapter Ch 5.

Vega, L., Rey, H., Benesty, J. & Tressens, S. (2009), 'A fast robust recursive least-squares algorithm', *IEEE Transactions on Signal Processing* **57**(3), 1209–1216.

Wang, C., Yen, L. & Principe, J. (1996), 'On-line transform domain LMS algorithm implemented with PCA learning', *Proceedings of the World Conference on Neural Networks* pp. 599–602.

Wardle, S. (1998), 'A hilbert transformer frequency shifter for audio', *Proceedings of the DAFX Workshop on Digital Audio Effects* pp. 25–29.

Weaver, D. (1956), 'A third method of generation and detection of single-sideband signals', *Proceedings of the Institute of Radio Engineers* **44**(12), 1703–1705.

Wei, W., Gelfand, S. & Krogmeier, J. (2001), 'Noise-constrained least mean squares algorithm', *IEEE Transactions on Signal Processing* **49**(9), 1961–1970.

Weinstein, S. & Ebert, P. (1971), 'Data transmission by frequency-division multiplexing using the discrete fourier transform', *IEEE Transactions on Communication Technology* **19**(5), 628–634.

Widrow, B., McCool, J., Larimore, M. & Johnson, C. (1976), 'Stationary and nonstationary learning characteristics of the lms adaptive filter', *Proceedings of the IEEE* **64**(8), 1151–1162.

Widrow, B. & Stearns, S. D. (1985), *Adaptive Signal Processing*, Prentice Hall.

Wiener, N. (1949), *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*, MIT Press.

Wireless Technologies - Cisco Documentation (2002), `http://docwiki.cisco.com/wiki/Wireless_Technologies`. [Online: accessed April 2011].

Ye, H. & Wu, B. (1991), 'A new double-talk detection algorithm based on the orthogonality theorem', *IEEE Transactions on Communications* **39**(11), 1542–1545.

Zimmerman, M. & Kirsch, A. (1967), 'The an/gsc-10 (KATHRYN) variable rate data modem for hf radio', *IEE Transactions on Communictions* **15**(2), 197–205.

Zipf, J., Tobias, O. & Seara, R. (2008), 'A VSS LMS algorithm based on error autocorrelation', *European Signal Processing Conference* .

Appendix A

Mathematical Results

A.1 Identities Developed for Optimisation of Test Signals

These identities in this section have not been discovered in the literature, but were developed because they were required in order to find the the optimal test signal.

A.1.1 Identity

Let $\mathbf{D}$ be a diagonal square matrix and let $\mathbf{A}$ and $\mathbf{B}$ be arbitrary square matrices. Then

$$\text{diag}(\mathbf{ADB})_i = \sum_{k=1}^{n} \mathbf{A}_{ik}\mathbf{B}_{ki}\mathbf{D}_{kk}. \tag{A.1}$$

proof

$$(\mathbf{DB})_{kj} = \sum_{i=1}^{n} \mathbf{D}_{ki}\mathbf{B}_{ij} = \mathbf{D}_{kk}\mathbf{B}_{kj}$$

$$(\mathbf{ADB})_{ij} = \sum_{k=1}^{n} \mathbf{A}_{ik}(\mathbf{DB})_{kj} = \sum_{k=1}^{n} \mathbf{A}_{ik}\mathbf{D}_{kk}\mathbf{B}_{kj}$$

$$\text{diag}(\mathbf{ADB})_i = (\mathbf{ADB})_{ii} = \sum_{k=1}^{n} \mathbf{A}_{ik}\mathbf{B}_{ki}\mathbf{D}_{kk}$$

A.1.2 Identity

Let $\mathbf{D}$ be a diagonal square matrix and let $\mathbf{A}$ and $\mathbf{B}$ be arbitrary square matrices. Then

$$\text{diag}(\mathbf{ADB}) = (\mathbf{A} \diamond \mathbf{B}^T)\text{diag}(\mathbf{D}). \tag{A.2}$$

proof

$$(\mathbf{A} \diamond \mathbf{B}^T)_{ik} = \mathbf{A}_{ik}\mathbf{B}_{ki}$$

$$((\mathbf{A} \diamond \mathbf{B}^T)\text{diag}(\mathbf{D}))_i = \sum_{k=1}^{n}(\mathbf{A} \diamond \mathbf{B}^T)_{ik}\mathbf{D}_{kk} = \sum_{k=1}^{n}\mathbf{A}_{ik}\mathbf{B}_{ki}\mathbf{D}_{kk}$$

By Identity (A.1)

$$= \text{diag}(\mathbf{ADB})_i$$

A.1.3 Identity

Let $\mathbf{d}$ be a vector and let $\mathbf{D}$ be a diagonal square matrix, such that

$$\mathbf{D} = \text{diag}(\mathbf{d}).$$

Let $\mathbf{A}$ be any square matrix. Then

$$\nabla_{\mathbf{d}} Tr(\mathbf{DA}) = \text{diag}(\mathbf{A})^T \tag{A.3}$$

Note that this also implies

$$\nabla_{\mathbf{d}} Tr(\mathbf{BDA}) = \text{diag}(\mathbf{AB})^T$$

proof

$$\begin{aligned}\nabla_{\mathbf{d}}(Tr(\mathbf{DA})) &= \nabla_{\mathbf{d}}(\mathbf{d}_1\mathbf{A}_{11} + \mathbf{d}_2\mathbf{A}_{22} + ... + \mathbf{d}_n\mathbf{A}_{nn}) \\ &= [\mathbf{A}_{11}, \mathbf{A}_{22}, ..., \mathbf{A}_{nn}]^T = \text{diag}(\mathbf{A})\end{aligned}$$

A.1.4 Identity

Let $\mathbf{d}$ be a vector and let $\mathbf{D}$ be a real diagonal square matrix, such that

$$\mathbf{D} = \text{diag}(\mathbf{d}).$$

Let $\mathbf{A}$ and $\mathbf{B}$ be arbitrary square matrices. Then

$$\nabla_{\mathbf{d}} Tr(\mathbf{DADB}) = \text{diag}(\mathbf{ADB} + \mathbf{BDA})^T, \tag{A.4}$$

Note that this also implies

$$\nabla_{\mathbf{d}} Tr(\mathbf{CDADB}) = \text{diag}(\mathbf{ADBC} + \mathbf{BCDA})^T,$$

proof

$$(\mathbf{DA})_{pr} = \sum_{k=1}^{n} \mathbf{D}_{pk}\mathbf{A}_{kr} = \mathbf{D}_{pp}\mathbf{A}_{pr}$$

$$(\mathbf{DA})_{rq} = \sum_{k=1}^{n} \mathbf{D}_{rk}\mathbf{A}_{kq} = \mathbf{D}_{rr}\mathbf{A}_{rq}$$

$$(\mathbf{DADB})_{pq} = \sum_{r=1}^{n} (\mathbf{DA})_{pr}(\mathbf{DB})_{rq} = \sum_{r=1}^{n} \mathbf{D}_{pp}\mathbf{A}_{pr}\mathbf{D}_{rr}\mathbf{B}_{rq}$$

$$\begin{aligned} Tr(\mathbf{DADB}) &= \sum_{s=1}^{n} (\mathbf{DADB})_{ss} = \sum_{s=1}^{n}\sum_{r=1}^{n} \mathbf{D}_{ss}\mathbf{A}_{sr}\mathbf{D}_{rr}\mathbf{B}_{rs} \\ &= \sum_{s=1}^{n} \mathbf{d}_s \sum_{r=1}^{n} \mathbf{d}_r \mathbf{A}_{sr}\mathbf{B}_{rs} = \sum_{r=1}^{n} \mathbf{d}_r \sum_{s=1}^{n} \mathbf{d}_s \mathbf{A}_{sr}\mathbf{B}_{rs} \end{aligned}$$

Application of the product rule gives:-

$$\begin{aligned} \frac{\partial}{\partial \mathbf{d}_i} Tr(\mathbf{DADB}) &= \sum_{r=1}^{n} \mathbf{d}_r \mathbf{A}_{ir}\mathbf{B}_{ri} + \sum_{s=1}^{n} \mathbf{d}_s \mathbf{A}_{si}\mathbf{B}_{is} \\ &= \sum_{r=1}^{n} \mathbf{D}_{rr}\mathbf{A}_{ir}\mathbf{B}_{ri} + \sum_{s=1}^{n} \mathbf{D}_{ss}\mathbf{B}_{is}\mathbf{A}_{si} \end{aligned}$$

By applying Identity (A.1) to both of these terms, we get

$$\frac{\partial}{\partial \mathbf{d}_i} Tr(\mathbf{DADB}) = \text{diag}(\mathbf{ADB})_i + \text{diag}(\mathbf{BDA})_i$$

Therefore

$$\nabla_{\mathbf{d}} Tr(\mathbf{DADB}) = \text{diag}(\mathbf{ADB} + \mathbf{BDA})^T$$

A.1.5 Identity

Let $\mathbf{A}$ be an arbitrary real matrix. Let $\mathbf{x}$ be an arbitrary real vector, and $\mathbf{X} = \text{Circ}(\mathbf{x})$. Then

$$\nabla_{\mathbf{x}} \text{tr}(\mathbf{XA}) = \text{traces}(\mathbf{A})^T \tag{A.5}$$

proof:

$$\begin{aligned}(\mathbf{XA})_{ij} &= \sum_{k=0}^{n-1} \mathbf{X}_{i,k}\ \mathbf{A}_{k,j} \\ &= \sum_{k=0}^{n-1} \mathbf{x}_{(i-k)\%n}\ \mathbf{A}_{k,j} \\ &= \sum_{k=0}^{n-1} \mathbf{x}_k\ \mathbf{A}_{(i-k)\%n,j}\end{aligned}$$

$$\begin{aligned}\text{tr}(\mathbf{XA}) &= \sum_{l=0}^{n-1} \sum_{k=0}^{n-1} \mathbf{x}_k\ \mathbf{A}_{(l-k)\%n,l} \\ &= \sum_{k=0}^{n-1} \mathbf{x}_k \sum_{i=0}^{n-1} \mathbf{A}_{(i-k)\%n,i}\end{aligned}$$

$$\begin{aligned}[\nabla_{\mathbf{x}} \text{tr}(\mathbf{XA})]_k &= \sum_{i=0}^{n-1} \mathbf{A}_{(i-k)\%n,i} \\ &= \text{traces}(\mathbf{A})_k\end{aligned}$$

A.1.6 Identity

Let $\mathbf{A}$ be an arbitrary real matrix. Let $\mathbf{x}$ be an arbitrary real vector, and $\mathbf{X} = \text{Circ}(\mathbf{x})$. Then

$$\nabla_x \text{tr}(\mathbf{X}^T\mathbf{A}) = \text{traces}(\mathbf{A}^T)^T \tag{A.6}$$

proof:

$$\begin{aligned}(\mathbf{X}^T\mathbf{A})_{ij} &= \sum_{k=0}^{n-1}\mathbf{X}_{k,i}\ \mathbf{A}_{k,j}\\ &= \sum_{k=0}^{n-1}\mathbf{x}_{(k-i)\%n}\ \mathbf{A}_{k,j}\\ &= \sum_{k=0}^{n-1}\mathbf{x}_{k}\ \mathbf{A}_{(k+i)\%n,j}\end{aligned}$$

$$\begin{aligned}\mathrm{tr}(\mathbf{X}^T\mathbf{A}) &= \sum_{l=0}^{n-1}\ \sum_{k=0}^{n-1}\mathbf{x}_{k}\ \mathbf{A}_{(k+l)\%n,l}\\ &= \sum_{k=0}^{n-1}\mathbf{x}_{k}\sum_{l=0}^{n-1}\mathbf{A}_{(k+i)\%n,i}\end{aligned}$$

$$\begin{aligned}\left[\nabla_{\mathbf{x}}\mathrm{tr}(\mathbf{X}^T\mathbf{A})\right]_k &= \sum_{i=0}^{n-1}\mathbf{A}_{(k+i)\%n,i}\\ &= \mathrm{traces}(\mathbf{A}^T)_k\end{aligned}$$

A.1.7 Identity

Let $\mathbf{A}$ and $\mathbf{B}$ be arbitrary real matrices. Let $\mathbf{x}$ be an arbitrary real vector, and $\mathbf{X} = \mathrm{Circ}(\mathbf{x})$. Then

$$\nabla_{\mathbf{x}}\mathrm{tr}(\mathbf{X}^T\mathbf{A}\mathbf{X}\mathbf{B}) = \mathrm{traces}(\mathbf{B}^T\mathbf{X}^T\mathbf{A}^T)^T + \mathrm{traces}(\mathbf{B}\ \mathbf{X}^T\mathbf{A})^T \tag{A.7}$$

proof:

$$\mathrm{tr}(\mathbf{X}^T\mathbf{A}\mathbf{X}\mathbf{B}) = \mathrm{tr}(\mathbf{X}^T(\mathbf{A}\mathbf{X}\mathbf{B})) = \mathrm{tr}(\mathbf{X}(\mathbf{B}\mathbf{X}^T\mathbf{A}))$$

Therefore, applying Identity (A.5) and Identity (A.6) using the product rule:-

$$\nabla_{\mathbf{x}}\mathrm{tr}(\mathbf{X}^T\mathbf{A}\mathbf{X}\mathbf{B}) = \mathrm{traces}(\mathbf{B}^T\mathbf{X}^T\mathbf{A}^T)^T + \mathrm{traces}(\mathbf{B}\ \mathbf{X}^T\mathbf{A})^T$$

A.1.8 Identity

Let $\mathbf{A}$ and $\mathbf{B}$ be a arbitrary real matrices. Let $\mathbf{x}$ be an arbitrary real vector, and $\mathbf{X} = \text{Circ}(\mathbf{x})$. Then

$$\text{traces}(\mathbf{A}\ \mathbf{X}^T\mathbf{B}) = \mathbf{G}\mathbf{x} \tag{A.8}$$

where the elements of the matrix G are given by

$$\mathbf{G}_{s,k} = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1}\mathbf{A}_{i,j}\mathbf{B}_{(j+k)\%n,(i+s)\%n}$$

proof:

$$\begin{aligned}(\mathbf{A}\mathbf{X}^T)_{i,j} &= \sum_{k=0}^{n-1}\mathbf{A}_{i,k}\ \mathbf{X}_{j,k}\\ &= \sum_{k=0}^{n-1}\mathbf{A}_{i,k}\ \mathbf{x}_{(j-k)\%n}\\ &= \sum_{k=0}^{n-1}\mathbf{A}_{i,(j-k)\%n}\ \mathbf{x}_k\end{aligned}$$

$$\begin{aligned}(\mathbf{A}\ \mathbf{X}^T\mathbf{B})_{i,r} &= \sum_{j=0}^{n-1}(\mathbf{A}\mathbf{X})_{i,j}\ \mathbf{B}_{j,r}\\ &= \sum_{j=0}^{n-1}\sum_{k=0}^{n-1}\mathbf{A}_{i,(j-k)\%n}\ \mathbf{x}_k\ \mathbf{B}_{j,r}\\ &= \sum_{k=0}^{n-1}\mathbf{x}_k\sum_{j=0}^{n-1}\mathbf{A}_{i,(j-k)\%n}\ \mathbf{B}_{j,r}\end{aligned}$$

We can define traces() by

$$\text{traces}(\mathbf{P})_s = \sum_{i=0}^{n-1}\mathbf{P}_{i,(i+s)\%n}$$

for each index s. Therefore:-

$$\begin{aligned}\text{traces}(\mathbf{A}\ \mathbf{X}^T\mathbf{B})_s &= \sum_{i=0}^{n-1}\left(\mathbf{A}\ \mathbf{X}^T\mathbf{B}\right)_{i,(i+s)\%n}\\ &= \sum_{i=0}^{n-1}\sum_{k=0}^{n-1}\mathbf{x}_k\sum_{j=0}^{n-1}\mathbf{A}_{i,(j-k)\%n}\ \mathbf{B}_{j,(i+s)\%n}\\ &= \sum_{k=0}^{n-1}\mathbf{x}_k\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}\mathbf{A}_{i,j}\ \mathbf{B}_{(j+k)\%n,(i+s)\%n}\end{aligned}$$

$$
\begin{aligned}
(\mathbf{Gx})_s &= \sum_{k=0}^{n-1} \mathbf{G}_{s,k}\, \mathbf{x}_k \\
&= \sum_{k=0}^{n-1} \left(\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \mathbf{A}_{i,j} \mathbf{B}_{(j+k)\%n,(i+s)\%n} \right) \mathbf{x}_k \\
&= \sum_{k=0}^{n-1} \mathbf{x}_k \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \mathbf{A}_{i,j} \mathbf{B}_{(j+k)\%n,(i+s)\%n} \\
&= \text{traces}(\mathbf{A}\, \mathbf{X}^T \mathbf{B})_s
\end{aligned}
$$

A.2 Identities Adopted for Optimisation of Test Signals

A.2.1 Identity

Given a vector of random variables x and symmetric matrix A then

$$E(\mathbf{x}^T \mathbf{A} \mathbf{x}) = \text{tr}(\mathbf{A}\mathbf{C}_x) + \mathbf{m}^T \mathbf{A} \mathbf{m} \tag{A.9}$$

Where $\mathbf{C}_x$ is the covariance matrix of $\mathbf{x}$ and $\mathbf{m}$ is the mean of $\mathbf{x}$ ((Petersen & Pederson 2008) Identity 306).

A.2.2 Identity

Given arbitrary square matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{X}$ then

$$\nabla_{\mathbf{X}}\, \text{tr}(\mathbf{AXB}) = \mathbf{A}^T \mathbf{B}^T, \tag{A.10}$$

((Petersen & Pederson 2008) Identity 93).

A.2.3 Identity

Given arbitrary square matrices $\mathbf{A}$ and $\mathbf{X}$ then

$$\nabla_{\mathbf{X}}\, \text{tr}(\mathbf{X}\mathbf{A}\mathbf{X}^T) = \mathbf{X}\mathbf{A}^T + \mathbf{X}\mathbf{A}, \tag{A.11}$$

((Petersen & Pederson 2008) Identity 101).

A.2.4 Identity

Given an square matrix $\mathbf{Y}$ that is a function of a scalar $\mathbf{x}$ then

$$\nabla_{\mathbf{x}}(\mathbf{Y}^{-1}) \;=\; -\mathbf{Y}^{-1}\;\nabla_{\mathbf{x}}(\mathbf{Y})\;\mathbf{Y}^{-1}, \tag{A.12}$$

((Petersen & Pederson 2008) Identity 53).

A.3 Well Known Results

The identities in this section are well known, but have been proved here because it was difficult to find a suitable reference.

A.3.1 Linear Minimum Mean Square Estimator (LMMSE) of a variable given an observed variable

If we have two random variables (or vectors) $\mathbf{x}$ and $\mathbf{y}$ that are not independent, then they have a non trivial joint 2D probability distribution. If $\mathbf{x}$ is observable and we wish to estimate $\mathbf{y}$ given $\mathbf{x}$, we may do so using a linear estimator if we know the appropriate covariances:

$$\hat{\mathbf{y}} = \mathbf{C}_{xy}\mathbf{C}_{xx}^{-1}\mathbf{x} \tag{A.13}$$

proof: Let us express the linear estimation using a matrix:

$$\mathbf{y} = \mathbf{A}\mathbf{x}.$$

Then the expectation of the square of the absolute error is given by

$$\begin{aligned} E[(\mathbf{A}\mathbf{x}-\mathbf{y})(\mathbf{A}\mathbf{x}-\mathbf{y})^*] &= E[\mathbf{A}\mathbf{x}\mathbf{x}^*\mathbf{A}] - E[\mathbf{y}\mathbf{x}^*\mathbf{A}^*] - E[\mathbf{A}\mathbf{x}\mathbf{y}^*] + E[\mathbf{y}\mathbf{y}] \\ &= \mathbf{A}\mathbf{C}_{xx}\mathbf{A}] - \mathbf{C}_{xy}\mathbf{A}^* - \mathbf{A}\mathbf{C}_{xy} + \mathbf{C}_{yy} \end{aligned}$$

and since covariance matrices are Hermitian, we may factor in the following way

$$= (\mathbf{A}-\mathbf{C}_{xy}\mathbf{C}_{xx}^{-1})\mathbf{C}_{xx}(\mathbf{A}-\mathbf{C}_{xy}\mathbf{C}_{xx}^{-1})^* - \mathbf{C}_{xy}\mathbf{C}_{xx}^{-1}\mathbf{C}_{xy} + \mathbf{C}_{yy}.$$

Now all of the terms involving A are in the first term. It is known that covariance matricies are positive semidefinite. Therefore the first term is positive unless the first term is zero, and the minimum occurs when

$$\mathbf{A} = \mathbf{C}_{xy}\mathbf{C}_{xx}^{-1}.$$

Hence

$$\hat{y} = \mathbf{C}_{xy}\mathbf{C}_{xx}^{-1}\mathbf{x}.$$

A.3.2 Identity

$$\nabla_x(\mathbf{x}^T\mathbf{A}\mathbf{x}) = 2\mathbf{x}^T\mathbf{A} \tag{A.14}$$

where $\mathbf{x}$ is a real vector and $\mathbf{A}$ is a symmetric matrix.

proof:

$$\mathbf{x}^T\mathbf{A}\mathbf{x} = \mathbf{x}^T \begin{pmatrix} \sum_{i=1}^n \mathbf{A}_{1i}\ \mathbf{x}_i \\ \sum_{i=1}^n \mathbf{A}_{2i}\ \mathbf{x}_i \\ \cdots \\ \sum_{i=1}^n \mathbf{A}_{ni}\ \mathbf{x}_i \end{pmatrix} = \sum_{j=1}^n \sum_{i=1}^n \mathbf{x}_j\mathbf{x}_i\mathbf{A}_{ji}$$

$$\frac{\partial}{\partial \mathbf{x}_k}(\mathbf{x}^T\mathbf{x}) = \sum_{j=1}^n \mathbf{x}_j\mathbf{A}_{jk} - \mathbf{x}_k\mathbf{A}_{kk} + \sum_{i=1}^n \mathbf{x}_i\mathbf{A}_{ki} - \mathbf{x}_k\mathbf{A}_{kk} + 2\mathbf{x}_k\mathbf{A}_{kk}$$

$$= \sum_{j=1}^n \mathbf{x}_j\mathbf{A}_{jk} + \sum_{i=1}^n \mathbf{x}_i\mathbf{A}_{ki}$$

and by symmetry

$$\frac{\partial}{\partial \mathbf{x}_k}(\mathbf{x}^T\mathbf{x}) = 2\sum_{j=1}^n \mathbf{A}_{kj}\ \mathbf{x}_j$$

$$\nabla_x(\mathbf{x}^T\mathbf{A}\mathbf{x}) = \begin{pmatrix} \frac{\partial}{\partial \mathbf{x}_1}(\mathbf{x}^T\mathbf{A}\mathbf{x}) \\ \frac{\partial}{\partial \mathbf{x}_2}(\mathbf{x}^T\mathbf{A}\mathbf{x}) \\ \cdots \\ \frac{\partial}{\partial \mathbf{x}_n}(\mathbf{x}^T\mathbf{A}\mathbf{x}) \end{pmatrix}^T = \begin{pmatrix} 2\sum_{j=1}^n \mathbf{A}_{1j}\ \mathbf{x}_j \\ 2\sum_{j=2}^n \mathbf{A}_{2j}\ \mathbf{x}_j \\ \cdots \\ 2\sum_{j=n}^n \mathbf{A}_{nj}\ \mathbf{x}_j \end{pmatrix}^T = 2\mathbf{x}^T\mathbf{A}$$

A.3.3 Identity

$$\sum_{k=0}^{n-1} \omega^{jk} = \begin{cases} n & \omega^j = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{A.15}$$

where k is an integer and ω is known as the nth root of unity and is given by

$$\omega = e^{(\frac{2\pi i}{n})}.$$

proof:

Clearly, $\sum_{k=0}^{n-1} \omega^{jk} = n$, whenever $\omega^j = 1$, so we now need to prove that it is zero whenever $\omega^j \neq 1$.

Let S be the sums of the powers of ω^k:

$$\begin{aligned} S &= 1 + \omega^k + \omega^{2k} + ... + \omega^{(n-1)k} \\ \omega S &= \omega^k + \omega^{2k} + ... + \omega^{(n-1)k} + 1 \\ (\omega^j - 1)S &= 0 \end{aligned}$$

Therefore $S = 0$, whenever $(\omega^j \neq 1)$.

QED.

A.3.4 Identity

$$\sqrt{n}\ \mathrm{Diag}(\mathbf{Fx}) = \mathbf{F}\mathrm{Circ}(\mathbf{x})\mathbf{F}^*. \tag{A.16}$$

proof:

Let us begin with the definition (1.3) of the unitary DFT square matrix of dimension n:

$$\mathbf{F}_{j,k} = \frac{1}{\sqrt{n}}\xi^{jk}, \tag{A.17}$$

where ξ is given by

$$\xi = e^{(\frac{-2\pi i}{n})},$$

and for convenience lets us also define

$$\omega = \xi^{-1}.$$

Let us examine a key term from our LHS:

$$\sqrt{n}\,(\mathbf{Fx})_q \;=\; \sum_{p=0}^{n-1} \mathbf{x}_p \sqrt{n}\,\mathbf{F}_{q,p},$$

and by substituting (A.17) into the above, we get:

$$=\; \sum_{p=0}^{n-1} \mathbf{x}_p \xi^{qp}. \tag{A.18}$$

Each element of our LHS is given by:

$$(\text{LHS})_{q,s} \;=\; \sqrt{n}\,\text{Diag}(\mathbf{Fx})_{q,s},$$

and substuting (A.18) into the above gives:

$$=\; \begin{cases} \sum_{p=0}^{n-1}\ \mathbf{x}_p\ \xi^{qp} & q = s \\ 0 & \text{otherwise.} \end{cases}$$

Let us now define a matrix which *rotates* a vector or the columns of a matrix by pre-multiplication:

$$\mathbf{Q} = Circ(\{0, 1, 0, 0, \ldots, 0\}^T),$$

and let us define our circulant matrix Circ($\mathbf{x}$) in terms of $\mathbf{Q}$:

$$\text{Circ}(\mathbf{x}) = \sum_{p=0}^{n-1} \mathbf{x}_p \mathbf{Q}^p. \tag{A.19}$$

Since the matrix Q rotates the columns of a matrix, and since $\omega^k = \omega^{k\%n}$, we may write

$$\begin{aligned} (\mathbf{Q}^m \mathbf{F}^*)_{p,q} &= \frac{1}{\sqrt{n}}\,\omega^{(p-m)q} \\ &= \frac{1}{\sqrt{n}}\,\omega^{pq}\,\omega^{-mq}. \end{aligned} \tag{A.20}$$

From the first principles of matrix multiplication:

$$(\mathbf{F}\mathbf{Q}^m\mathbf{F}^*)_{s,q} \;=\; \frac{1}{n}\sum_{p=0}^{n-1} \mathbf{F}_{s,p}\,(\mathbf{Q}^m\mathbf{F}^*)_{p,q}$$

Substituting (A.20) into the above gives:

$$
\begin{aligned}
&= \frac{1}{n}\sum_{p=0}^{n-1} \omega^{-sp}\ \omega^{qp}\ \omega^{-mq} \\
&= \frac{1}{n}\sum_{p=0}^{n-1} \omega^{p(q-s)}\ \omega^{-mq} \\
&= \xi^{mq}\ \frac{1}{n}\sum_{p=0}^{n-1} \omega^{p(q-s)},
\end{aligned}
$$

and applying (A.15) :

$$
= \begin{cases} \xi^{mq} & q = s, \\ 0 & \text{otherwise.} \end{cases} \tag{A.21}
$$

By expanding into the left hand side below using (A.19), we get:

$$
\begin{aligned}
\mathbf{F}\text{Circ}(\mathbf{x})\mathbf{F}^* &= \mathbf{F}(\sum_{p=0}^{n-1} \mathbf{x}_p \mathbf{Q}^p)\mathbf{F}^* \\
&= \sum_{p=0}^{n-1} \mathbf{x}_p \mathbf{F}\mathbf{Q}^p\mathbf{F}^*.
\end{aligned} \tag{A.22}
$$

Finally, we have:

$$
(\text{RHS})_{s,q} = (\mathbf{F}\text{Circ}(\mathbf{x})\mathbf{F}^*)_{s,q},
$$

and substituting (A.22) into the above gives us:

$$
= (\sum_{p=0}^{n-1} \mathbf{x}_p \mathbf{F}\mathbf{Q}^p\mathbf{F}^*)_{s,q}.
$$

and substituting (A.21) into the above gives us:

$$
\begin{aligned}
&= \begin{cases} \sum_{p=0}^{n-1}\ \mathbf{x}_p\ \xi^{pq} & q = s, \\ 0 & \text{otherwise.} \end{cases} \\
&= (\text{LHS})_{s,q}.
\end{aligned}
$$

QED.

A.3.5 Utilising Knowledge of the Noise Power

If we dont know the second order statistics of the noise, but we do know the noise power, then a better estimator than $\mathbf{X}^{-1}$ is obtained by a shrinkage estimate (van Houwelingen 2001).

Let us say we have a random variable $\mathbf{x}$, and random noise, $\mathbf{z}$, is added to it before we can take a measurement $\mathbf{y}$, so that $\mathbf{y} = \mathbf{x} + \mathbf{z}$. We do not know $\mathbf{x}$ and $\mathbf{z}$, but we know that they both have zero mean, and we know their power, σ_x and σ_z.

If we use $c\mathbf{y}$ as our estimate of $\mathbf{x}$, $\hat{\mathbf{x}}$, then what is the best value of c to use? Let us take the power of the error of our estimate of $\mathbf{x}$,

$$\begin{aligned}
\varepsilon &= E[(\mathbf{x} - c\mathbf{y})^T(\mathbf{x} - c\mathbf{y})] \\
&= E[\mathbf{x}^T\mathbf{x}] - 2cE[\mathbf{x}^T\mathbf{y}] + c^2E[\mathbf{y}^T\mathbf{y}] \\
&= E[\mathbf{x}^T\mathbf{x}] - 2cE[\mathbf{x}^T\mathbf{x}] - 2cE[\mathbf{x}^T\mathbf{z}] + c^2E[\mathbf{x}^T\mathbf{x}] + 2c^2E[\mathbf{x}^T\mathbf{z}] + c^2E[\mathbf{z}^T\mathbf{z}].
\end{aligned}$$

Noting that $\mathbf{x}$ and $\mathbf{z}$ are uncorrelated and therefore $E[\mathbf{x}^T\mathbf{z}] = 0$,

$$\begin{aligned}
&= \sigma_x^2 - 2c\sigma_x^2 + c^2\sigma_x^2 + c^2\sigma_z^2 \\
\frac{d\varepsilon}{dc} &= 2(\sigma_x^2 + \sigma_z^2)c - 2\sigma_x^2
\end{aligned}$$

The best estimate of $\mathbf{x}$ is where the error of our estimate is at a minimum, so choosing c at the stationary point gives us

$$c = \frac{\sigma_x^2}{\sigma_x^2 + \sigma_z^2}.$$

Applying this principle to our case, $\mathbf{w} = \mathbf{X}\mathbf{h} + \mathbf{z}$, we can see that $\mathbf{X}^{-1}\mathbf{w}$ is no longer the best estimate of h. Knowing the noise power allows us to make a better estimate, specifically:

$$\hat{\mathbf{h}} = \frac{\sigma_w^2}{\sigma_w^2 + \sigma_z^2}\mathbf{X}^{-1}\mathbf{w}. \tag{A.23}$$

Appendix B

Guide to Software Source Code

I have made the software source code associated with this dissertation readily available for download. The code may be executed using MATLAB 2008 for evaluation or for use. I have chosen not to bind the printed software code into this dissertation. If printed the computer code would be approximately 200 pages long and it seems inconceivable that anyone would actually read the printed source code.

B.1 ECOOSE Source Code

The object oriented echo cancellation simulation environment presented in Chapter 3 is publicly available at (Braithwaite & Addie 2008) and can be readily used by anyone undertaking research in this field. Researchers who wish to propose new algorithms or analyse new ones are welcome to provide Matlab definitions of an algorithm class which can be used with the ECOOSE framework, by them or by others, thereby providing objective evidence regarding its performance.

B.1.1 The Directory Tree

The ECOOSE directory tree is described in this subsection.

Sounds The far end signal (i.e. the input to the channel) and the near end signal (i.e. the noise) are stored in Waveform Audio File Format (WAV) (Mathworks 2008*c*, Kabal 2006), in the directory *Sounds*. Only the discrete values of the sound from the WAV files are used. Once the files are in that directory, they are accessed via their file name without the ".wav" extension. The sound clips for both the near end and the far end need to be at least as long as the desired experiment.

LinChannels The linear channels are also stored as WAV files in the *LinChannels* directory. Like sounds, ECOOSE treats the channels as simply a series of discrete values. ECOOSE may use a truncated version of the channel, so the channels in these files may be longer than that actually used. Once the files are in that directory, they are accessed via their file name without the ".wav" extension.

Predists The memoryless non-linear part of a Hammerstien channel are stored in this directory. Once the files are in that directory, they are accessed via their file name without the ".mat" extension.

Classes The code for the classes that are part of the major class hierarchies are kept in this directory. The class name is the same as the file name without the ".m" extension in this directory. The major class hierarchies are depicted in Figure 3.2, and the classes belonging to the major class hierarchies are described in Subsection B.1.2.

Functions Miscellaneous code that is not part of the major class hierarchies is stored in this directory.

Runs Each file in this directory subtree represents an experiment. One simply invokes the file name without the ".m" extension from Matlab in order to run the corresponding experiment. The currently available experiments are described in Subsection B.1.3.

Results Holds the results from experiments as well as data created by one run for use in a later experiment.

B.1.2 Classes

ECOOSE has been recreated in a fully object oriented manner, and therefore a detailed description of ECOOSE must necessarily focus on its major class hierarchies. Here, we outline the classes belonging to the major class hierarchies of ECOOSE.

Channel Source Classes

Each experiment or run must instantiate a channel source class. The channel source class instantiated is the final authority on the channel state, and it is responsible for communicating that state to any other class that requires it.

chnlSrc This is the base class for all of the channel source classes. It is an abstract class and cannot be instantiated. This class has methods that implements the observer design pattern. Channel observer and channel calculator classes subscribe to a channel source class. The channel source class will notify all subscribing classes whenever the channel characteristics change, and it is then up to those classes to request the new channel information from the channel source class.

chnlSrcDummy This channel is is a subclass of *chnlSrc*. Although ECOOSE is primarily designed for evaluating echo cancellation algorithms for known channels, ECOOSE may also be used to determine an channel. In this case we do not know the channel beforehand, and we use chnlSrcDummy in this case.

chnlSrcLin This channel is is a subclass of *chnlSrc*. It is an abstract class and cannot be instantiated. In addition to the base channel source class, this class understands that the channel is represented by an impulse response. It is intended that dynamic and static linear channels inherit from this channel.

chnlSrcLinStat This class is used for static linear channels. This class may be instantiated. Most runs use this channel source class.

chnlSrcBaysStat This class is used for Hammerstein channels. It is a subclass of the *chnlSrcLinStat* class and in addition to the static linear channel, it also under-

stands the memoryless nonlinear channel, which is parameterised by Legendre Polynomial coefficients.

Channel Calculator Classes

Each experiment or run must instantiate a channel calculator class. The purpose of the instantiated channel calculator class is to calculate and provide the current value of $y(t)$ as depicted in Figure 1.1. The channel calculator class has no concept of whether a channel is static or dynamic. It merely updates its understanding of the channel whenever the channel source class sends a message to say that the channel has been updated.

chnlCalc This is the base class for all of the channel calculator classes. It is an abstract class and cannot be instantiated. This class has methods that communicate with the channel source class.

chnlCalcDummy As explained above, ECOOSE may also be used to determine an channel. In this case we do not know the channel beforehand, and we use chnlSrcDummy in this case.

chnlCalcLin This class is a subclass of *chnlCalc* and is used for all linear channels. This class may be instantiated. Most runs use this channel calculator class.

chnlCalcBays This class is a subclass of *chnlCalcLin*. It adds a memoryless nonlinear distortion to the linear channel, thus forming a Hammerstien channel.

Channel Observer Classes

Each experiment or run must instantiate a channel observer class. The purpose of the instantiated channel observer class is observe the observe and plot or save some functions of either the coefficients estimated by the algorithm class or the some functions of the difference between the estimated channel coefficients and the known values of the channel coefficients. The channel observer class has no concept of whether a channel is static or dynamic. The channel observer class must understand the model of the

channel assumed by the algorithm class. It keeps track of the known channel, getting the latest values whenever the channel source class says it has been updated. At each time step it will ask the algorithm class for the values it needs for plotting or saving.

obsBase This is the base class for all of the channel observer classes. This class has methods that communicate with the channel source class. This class may be instantiated, and has code to monitor the difference between the estimated value of y(t) and the known value of y(t).

obsLin This is a subclass of *obsBase*, and has code to monitor the difference between the estimated value of the impulse response and the known value of the impulse response.

obsBays This is a subclass of *obsLin*, and has code to monitor the difference between the estimated value of the memoryless nonlinear part of a Hammerstien channel and its known value.

Result Classes

The instantiated channel observer class expects a results object, and gives all its results to the results object. The purpose of the results object is to either plot or to save the results provided by the channel observer class.

rsltBase This is the base class for all of the results classes. It allows the observer class to create curves and to provide the data for them at each timestep. It does not actually do anything with that data.

rsltPlot This is a subclass of *rlstBase*. It plots the data collected in each curve whenever its plotting method is invoked.

rsltFile This is a subclass of *rlstBase*. It saves the data collected in each curve into a file in matlab format whenever its output method is invoked.

rsltWav This is a subclass of *rlstBase*. It saves the data collected in each curve into a file in WAV format whenever its output method is invoked.

Algorithm Classes

Each experiment or run must instantiate a algorithm class. Algorithm classes that can be instantiated are implementations of a given echo cancellation algorithm. Algorithm classes do not have information about the known channel, but have $x(t)$ and $w(t)$ as their only inputs.

algBase This is the base class for all of the algorithm classes. It is an abstract class and cannot be instantiated.

algDummy This is an algorithm class that can be instantiated, but has no actual echo cancellation algorithm. It is useful when we are not interested in testing an algorithm.

algBlock This is a subclass of *algBase*. This is the base class for all of the block based algorithm classes such as frequency division algorithms. It is an abstract class and cannot be instantiated.

algFreqLms This is a subclass of *algBlock*. It implements frequency division echo cancellation.

algLinear This is the base class for all of the time domain algorithms that use an impulse response. It is an abstract class and cannot be instantiated.

algLinNLMS This is the standard reliable NLMS algorithm. Very useful for making comparisons with more advanced algorithms.

algLinAP This is an implementation of the AP algorithm. The order of the AP algorithm is one of the configuration parameters.

algNlinBays1 This is the implementation of the HLMS algorithm as documented in Chapter 4.

B.1.3 Demonstrations

Ready made demonstrations have been distributed as part of ECOOSE. They require no arguments and may be easily invoked by name from the MATLAB command prompt,

providing one has invoked MATLAB from (or started MATLAB and set the current directory to) the main directory of ECOOSE, and then set up the execution paths for ECOOSE by issuing the command "setPaths". The following demonstrations are available:

nlmsTiny1 Plots the convergence of the NLMS algorithm excited with white noise and no doubletalk.

freqLmsTiny1 Plots the convergence of the block based frequency domain algorithm excited with white noise and no doubletalk.

nlmsUseY In this experiment, the channel is not known in advance, but is being determined. The dummy channel source and the dummy channel calculator are used, and $x(t)$ and $y(t)$ are supplied from sound files. The NMLS algorithm is used to cancel the echo and the results are plotted. Since there is considerable distortion in $y(t)$, the cancellation is not cancelled down to zero.

apSimpleTiny Plots the convergence of an order 4 AP algorithm excited with white noise and no doubletalk.

apTiny1 Plots the convergence of four different orders of AP algorithms excited with white noise and no doubletalk together on the one plot.

nlmsLinTiny1 Generates the first plot displayed in Figure 4.1. It features the NLMS algorithms peforming cancellation on a linear channel.

baysLinTiny1 Generates the second plot displayed in Figure 4.1. It features the HLMS algorithms peforming cancellation on a linear channel.

nlmsPreDistTiny1 Generates the third plot displayed in Figure 4.1. It features the NLMS algorithms peforming cancellation on a non-linear channel.

baysPreDistTiny1 Generates the final plot displayed in Figure 4.1. It features the HNLMS algorithms peforming cancellation on a non-linear channel.

B.2 Optimal Test Signal Source Code

The code that implements the theory presented in in Chapter 6 is publicly available from (Braithwaite & Addie 2010). One can view the code, and could potentially invoke the optimiser routines directly to perform an optimisation problem. There are also high level ready made routines available for easy invocation.

B.2.1 The Directory Tree

The Optimal Test Signal directory tree is described in this subsection.

Functions The core frequency domain and time domain optimisation are in this directory, along with scripts to perform sets of experiments and miscellaneous subroutines.

TimingRuns This directory contains some scripts that perform sets of experiments for the purpose of timing.

Results This directory is a place for files containing the results of experiments to reside temporarily.

B.2.2 Core Routines for Optimisation

```
function [xRslt,rslt] = optx(n, xInit, Ch, Cz, p)
```

This routine finds an optimal test signal of size n and the expected error for that test signal given the impulse response covariance matrix Ch, the covariance of the noise Cz, and the maximum test signal power p. `optx()` performs the optimisation in the time domain . The algorithm for the time domain routine `optx()` is as follows :-

- call `makexA()` to obtain the optimal estimator A using the formula (6.7).
- call `makexEps()` to obtain ε using the formula (6.6).

- Loop until the reductions in ε are smaller than a given threshold:
  - Use the class `makeX` to find a new value of x using the formulas (6.13) and (6.15).
  - call `makexA()` to obtain the optimal estimator A using the formula (6.7).
  - call `makexEps()` to obtain ε using the formula (6.6).
  - The difference between the newly derived value of x gives us a direction to move.
  - Choose points x along a line in this direction in geometrically increasing step sizes for as long as doing so reduces the value of ε.
- `optx()` monitors the current value of ε to ascertain if the algorithm is still making progress.

```
function [xRslt,rslt] = optf(n, xInit, Ch, Cz, p)
```

This routine finds an optimal test signal of size n and the expected error for that test signal given the impulse response covariance matrix Ch, the covariance of the noise Cz, and the maximum test signal power p. `optf()` performs the optimisation in the frequency domain. `optx()` and `optf()` use different formulas, and those formulas are implemented into different routines. Despite this `optx()` and `optf()` essentially use the same algorithm. Where `optx()` uses `makexEps()`, `makexA()` and `makeX()`, `optf()` uses `makefEps()`, `makefA()` and `makef()`.

B.2.3 Manufacturing the Problem

Functions that have names begin with `newCh` produce impulse response covariance matrices. The function `newChDist1()` is of particular interest, because it manufactures a covariance matrix using the method described in Section 7.1.5. Functions that have names begin with `newCz` produce the covariance of the noise. Functions that have names begin with `newX` produce initial starting points for the test signal.

B.2.4 Orchestrating Experiments and Making Plots

We refer to an invocation of the optimiser routines as an experiment. It would be very tedious to manually assemble the data for each experiment and to archive and plot the results. These tasks need to be automated, and code which performs the experiments and plots the results is described here.

exploreRuns This is a class that performs a series of runs, and saves the results into multidimensional arrays and saves them into a file. Once the data has been saved, it may be loaded into MATLAB at a later time in order to generate graphs or tables. There are several parameters that may be set by member functions, and these all have default values. The default size of the problem, n, is 64. Other parameters are varied and an experiment is performed for each combination. The defaults are as follows: The Signal to Noise Ratio (SNR) is varied from 0.1 to 10 in 10 logarithmic steps (i.e. 11 different values). The coloredness of the covariance of the noise (Czc) is varied from 0 to 2 in 10 steps. The coloredness of the covariance of the channel (Chc) is varied from 0 to 2 in 10 steps. For each Chc, the random number generator seed is varied from 10 to 21 in 10 steps. And for each random number generator seed, a initial starting point for the test signal, is randomly generated. Experiments are performed both in the frequency domain and the time domain. Hence 29282 experiments are performed if the default values are used.

mkpErr `mkpErr`, like exploreRuns, is also a class that performs a series of runs and saves the results and timings into a file for later use. `mkpErr` differs from exploreRuns, in that `mkpErr` does not generate a hypercube of data. `mkpErr` exploits an object oriented design so that it may carry out just the runs required for a given plot and has methods that read its data file output, and generates the desired plots. `mkpErr` is an abstract class and cannot be instantiated.

Generally, these plots show how results change as some parameter is varied. Subclasses of `mkpErr` must provide the method `setVP(k)` (Set Variable Parameter) which sets the protected property of `mkpErr` that is to vary according to the value of its argument

k so that that property will vary over a series of experiments in order to make the plot. In addition, the subclasses also create the covariance matrices C_h and C_z, and override default values such as the size of the problem and the range of random number generator seeds. These subclasses also have methods to set various parameters of the plot.

B.2.5 Demonstrations

Ready made demonstrations accompany this distribution. They require no arguments and may be easily invoked by name from the MATLAB command prompt, providing one has invoked MATLAB from (or started MATLAB and set the current directory to) the main directory called "Optsig", and then set up the execution paths by issuing the command "setPaths".

The plots in Figure 6.1 can be created as follows:

```
clear           % Removes old state and class definitions.
setPaths        % Sets MATLAB paths for the source code here.
mkpErr_Dist_cch.doit100()  % Make left top graph.
mkpErr_Dist_cch.doit025()  % Make left bottom graph.
mkpErr_Dist_ccz.doit100()  % Make right top graph.
mkpErr_Dist_ccz.doit025()  % Make right bottom graph.
```

Timing runs may be performed by using the ttx_runs() which will invoke the time domain optimiser, optx() with experiments that have dimensions, n, of 16, 32, 64, 128, 256, 512, 1024 in turn. This scripts prints a time stamp just before and just after each invocation of optx() so that the execution time may be evaluated. It may well take a week for these timing runs to complete. The script ttf_runs() performs the same experiments with the frequency domain optimiser optf().