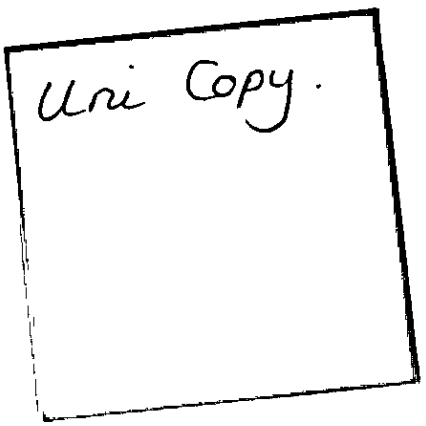
Uni Copy.

Blind Adaptive Algorithms for Channel Shortening in Wireline Multicarrier Systems

Thesis submitted to Loughborough University in candidature for the degree of Doctor of Philosophy.

Khaled Maatoug

Advanced Signal Processing Research Group
Department of Electronic and Electrical Engineering
Loughborough University
2009

ABSTRACT

In wireline multicarrier systems a cyclic prefix is commonly used to facilitate simple channel equalization at the receiver. The selection of the length of the cyclic prefix is a trade-off between maximizing the length of the channel for which inter-symbol interference is eliminated and optimizing the transmission efficiency When the length of the channel exceeds that of the cyclic prefix, adaptive channel shorteners can be used to force the effective channel length of the combined channel and channel shortener to satisfy the cyclic prefix constraint. The focus of this thesis is the design of new blind adaptive time-domain algorithms for channel shortening in wireline multicarrier systems, with good convergence properties and low computational complexity

An overview of the previous work in the field of channel shortening algorithms for use in wireline multicarrrier systems is given. Emphasis is placed on the family of property restoral algorithms, including the single lag autocorrelation minimizing (SLAM) blind adaptive algorithm, which forms the basis for the time-domain algorithms considered in the remainder of the thesis.

The relatively slow initial convergence of the SLAM blind adaptive algorithm is therefore improved by the proposal of a new variable-step SLAM algorithm and a quasi-Newton adaptive algorithm These algorithms are compared in terms of computational complexity and memory

usage so that their suitability for real-time implementation can be assessed. Simulation studies are performed on the basis of real carrier serving area (CSA) loop test channels.

A fundamentally new random lag selection-based blind adaptive channel shortening algorithm named the exponential probability generalized lag hopping sum squared autocorrelation minimizing algorithm (EGLHSAM) is then proposed which overcomes the possibility of ill-convergence in SLAM-type algorithms for particular channels. The exponential probability is chosen to represent approximately the envelope behaviour of the CSA loop test channels. The performance of EGLHSAM is assessed through simulations

Finally, the problem of decay parameter selection within the EGLHSAM algorithm is overcome by modifying the exponential probability density function employed in the random lag selection to a uniform form. This algorithm is named the GLHSAM algorithm and is demonstrated to have the capacity to match the convergence properties of the original sum squared autocorrelation minimization algorithm proposed by Martin and Johnson whilst retaining the complexity of the SLAM algorithm proposed by Nawaz and Chambers.

This thesis is dedicated to my wife and my children Omnya, Ayat and Anas.

ACKNOWLEDGEMENTS

I would like to express my thanks to those people who have contributed to making this research an invaluable experience some academically and others on a more personal level with their friendship and encouragement, especially during difficult times

Firstly ,I would like to thank my PhD supervisors Professor Jonathon Chambers and Dr Sangarapillai Lambotharan for giving me this opportunity and for their guidance throughout my research studies Special thanks go to Prof. Jonathon Chambers without his invaluable support and monitoring this thesis would have not been accomplished. He has contributed to all papers and the thesis with a major impact he has my best respect professionally and personally. I would also like to express my thanks to the administration staff at both Loughborough and Cardiff Universities, where this research was originally started. I would like to specially thank Mrs. Chris Lee from Cardiff University

I am grateful to Dr. R. K. Martin for his invaluable advices which was given to me whilst we were in Las Vegas during the ICASSP 2008 conference .

Thanks also go to my colleagues, many of whom have now become firm friends, in the DSPs Centre both at Loughborough and Cardiff.

Finally, thanks go to my family, both here and back home in Libya, who without their continued support and understanding I would not

have been able to proceed with my studies Special thanks to my wife for the unconditional support through out my studies.

PUBLICATIONS

- K Maatoug and J. A. Chambers,"A study of fast converging single lag autocorrelation minimizing algorithms for real time channel shortening in wireline systems," *Proceedings of the European DSP Education & Research Symposium Texas Instruments EDESR 2008*, Isreal, Jan. 2008.

- K. Maatoug and J A Chambers,"A generalized blind lag hopping adaptive channel shortening algorithm based upon squared autocorrelation minimization (GLHSAM)," *The Third International Conference on Systems and Networks Communication (*ICSNC 2008*)*, Malta, Oct 2008.

- K. Maatoug and J A Chambers, "A generalized blind lag hopping adaptive channel shortening algorithm based upon squared auto-correlation minimization, " *The 8th* IMA *International Conference on Mathematics in Signal Processing*, Cirencester, U K., Dec 2008.

- K. Maatoug and J.A Chambers, "Blind lag hopping adaptive channel shortening algorithm based upon squared auto-correlations minimization", ICCTA, Egypt, April 2008.

Acronyms

ADSL	Asymmetric Digital Subscriber Line
AR	Autoregressive
CMA	Constant Modulus Algorithm
CP	Cyclic Prefix
CSA	Carrier Serving Area
DAB	Digital Audio Broadcast
dB	decibel
DD-LMS	Decision-Directed-LMS Algorithm
DMT	Discrete Multitone
DSL	Digital Subscriber Lines
DVB	Digital Video Broadcast
EC-ADSL	Echo Cancelled-ADSL
EGLHSAM	Exponential Probability Generalized Lag Hopping SAM
FDM-ADSL	Frequency Division Multiplexed ADSL
FEQ	Frequency domain Equalizer

FFT	Fast Fourier Transform
FIR	Finite Impulse Response
FRODO	Forced Redundancy with Optional Data Omission
GLHSAM	Generalized Blind Adaptive Lag Hopping SAM
ICI	Inter Carrier Interference
IFFT	Inverse Fast Fourier Transform
ISI	Inter Symbol Interference
LMS	Least Mean Square
MBR	Maximum Bit Rate
MCM	Multicarrier Modulation
MERRY	Multicarrier Equalization by Restoration of Redundancy
MFB	Matched Filter Bound
MIMO	Multiple Input Multiple Output
MMSE	Minimum Mean Squared Error
MSE	Mean Squared Error
MSSNR	Maximum Shortening Signal to Noise Ratio
OFDM	Orthogonal Frequency Division Multiplexing
PTEQ	Per Tone Equalization
QAM	Quadrature Amplitude Modulation
QN-SLAM	Quasi-Newton-SLAM

RLS	Recursive Least Squares
SAM	Sum-squared Autocorrelation Minimization
SIR	Signal-to-Interference Ratio
SISO	Single Input Single Output
SLAM	Single Lag Autocorrelation Minimization
SNR	Signal to Noise Ratio
SSNR	Shortening Signal to Noise Ratio
TEQ-FB	TEQ-Filter Bank
TEQ	Time domain Equalizer
TIR	Target Impulse Response
VS-SLAM	Variable-Step SLAM
Wi-Fi	Wireless Fidelity
WiMax	Worldwide Interoperability for Microwave Access
WSS	Wide-Sense Stationary

CONTENTS

MATHEMATICAL NOTATIONS

x	Scalar quantity
$\mathbf{x}$	Vector quantity
$\mathbf{X}$	Matrix quantity
$\|\ \|$	Absolute value
$\|\|\ \|\|$	Euclidean norm
$(\ )^T$	Transpose operator
$(.)^*$	Complex conjugate operator
$(.)^H$	Hermitian transpose operator
$\mathbb{C}$	Set of complex numbers
$*$	Discrete time domain
$\Re(x)$	Real part of x
$\mathbf{I}$	Identity matrix
m **mod** n	The remainder of the integer division of m by n

$\otimes$	Kronecker product
$\lfloor\ \rfloor$	Transaction operation

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 Introduction and Motivation

Advanced wireless and wireline communication systems such as Institute of Electrical and Electronics of America IEEE 802 11/g wireless-fidelity (Wi-Fi), IEEE 802.16 wireless (WiMAX), ADSL, and ADSL2/+, have adopted multicarrier modulation (MCM) as the signaling technique either in the form of Orthogonal frequency division multiplexing (OFDM) [3] for wireless systems, or in the form of discrete multi tone (DMT) for wireline systems, due to its ability to combat the dispersive effect of the communication channel For the proper operation of MCM, a cyclic prefix (CP) which is at least as long as the length of the channel impulse response minus 1, has to be appended to the data part of the transmitted frame The CP is the last v samples of the original N samples to be transmitted. The CP is inserted between transmitted frames to combat inter symbol and inter carrier interferences (ISI and ICI) which significantly reduce the system performance. At the receiver the CP is removed and the remaining N samples are then processed by the receiver.

However, if the length of the channel is large, the throughput efficiency of the system deteriorates significantly with this additional load

of CP which does not convey user data. It is, therefore, desirable either to make v as small as possible or to choose a large data length for the transmitted frame, N Selecting large N will increase the computational complexity, system delay, and memory requirements of the transceiver. In order to overcome these problems a shorter cyclic prefix can be designed as an engineering compromise to minimize throughput loss whilst ensuring that a time domain channel equalizer (TEQ), can be used to shorten the effective channel to be no longer than the CP used Channel shortening is a generalization of equalization and the TEQ generally has not to be longer than the channel as its job is to shorten the channel to a given length, rather than shortening it to length one as is done in classic equalization. The TEQ is usually an FIR filter The focus of this thesis is to develop blind adaptive algorithms for the TEQ design. [4]

Figure (1.1) [1] shows the structure of CP and data frame Here the length of the data part of the frame is 12 while the channel is assumed to be of length 4. Hence a CP of length 3 is used Each transmitted frame will contain user data to be transmitted in the boxes labelled 4 to 15 and the last three boxes are copied to the start of the frame as a CP to combat ISI and ICI. The loss of throughout is quantified by

$$Loss\ of\ Throughput = \frac{v}{N+v} \tag{1.1.1}$$

Therefore, for the frame arrangement in Figure(1.1) the data throughput loss is 20%. Figure (1.2) [1] shows the place of the TEQ in the overall block diagram of a baseband MCM system If the TEQ weight vector w is designed to shorten the effective channel denoted by c ($\mathbf{c} = \mathbf{h} * \mathbf{w}$)

where the (*) denotes linear discrete time convolution and h is the channel vector to be of length 2, the CP will reduce to length 1 in the Figure (1.1) and loss of throughput will reduce to 7 7%

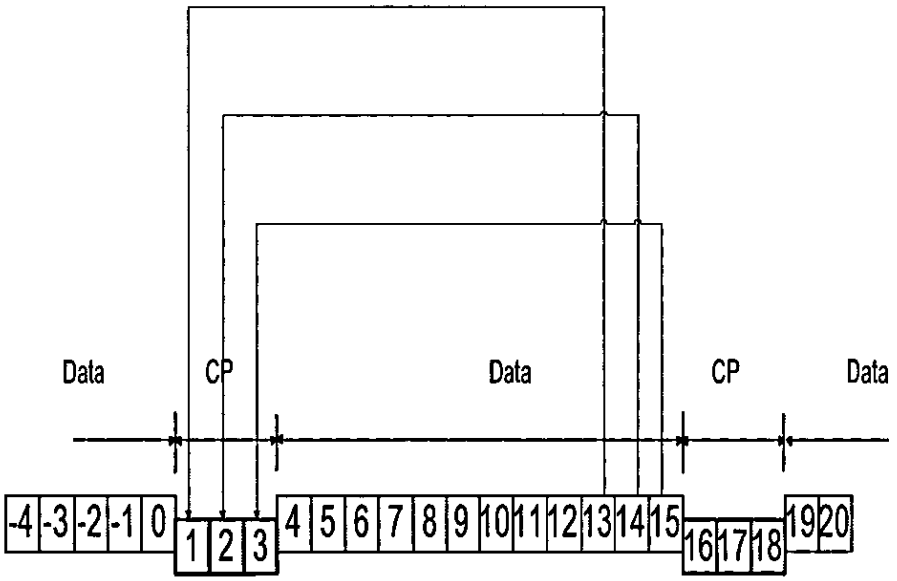

Figure 1.1. Structure of the data and cyclic prefix used in multicarrier transmission [1].

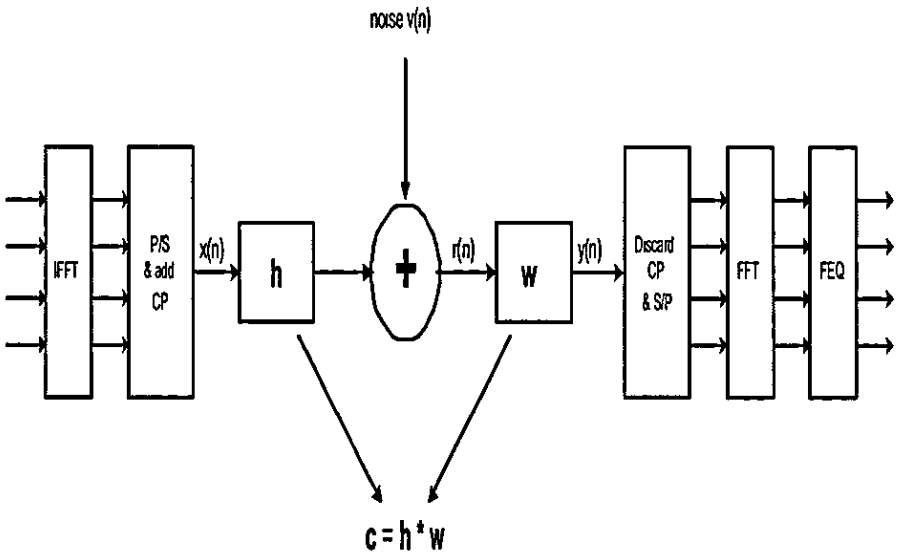

Figure 1.2. Baseband block diagram of the OFDM Transmitter and Receiver showing the channel h, TEQ w, and noise.

The baseband OFDM multi-carrier model along with the TEQ are shown in Figure (1.2). The input bits stream is first divided into blocks of N quadrature amplitude modulation (QAM) symbols. These QAM symbols are modulated onto N subchannels An efficient means to convert the N subchannels data to the time domain is to use an inverse fast Fourier transform (IFFT). The output of the IFFT is converted from parallel to serial and the CP is inserted The data are then serially transmitted. At the receiver in the baseband the ISI corrupted CP is discarded and an FFT is used to demodulate the signal. Because of the nature and length of the CP, the linear convolution between the effective channel $\mathbf{c} = \mathbf{h} * \mathbf{w}$ and the transmitted signal becomes circular. Therefore, the output of the FFT at each subchannel is the multiplication of the symbol sent on that subchannel and the frequency response of the effective channel at the subchannel plus the noise at that subchannel. Finally, the transmitted symbols are retrieved by dividing this output by the one-tap FEQs which are actually the frequency responses of the effective channel at the respective subchannels

Further examples of multicarrier communication systems include wireless local area networks (IEEE 802.11 a/g/n, HIPERLAN/2) [5], wireless metropolitan area networks (IEEE 802 16) a k.a Fixed WiMax (IEEE 802.16d) [6], IEEE Mobile WiMax 802.16/e [7], Digital Audio Broadcast (DAB) [8] and Digital Video Broadcast (DVB) [9] in Europe, satellite radio (Sirius and XM Radio) [10], and the proposed standard for multiband ultra wideband (IEEE 802.15 3a). Examples of wireline multi-carrier systems include power line communications (HomePlug) [11] and Digital subscriber lines (DSL) [12]. [13] discusses the

application of DMT signalling to high speed back plane interconnects. Tight power budgets in backplane links impose severe constraints on DMT block size and suggest the use of channel shortening filters in the system to maximise throughput OFDM in combination with MIMO technology is also being investigated for the Fourth Generation (4G) mobile phone systems [14] [15] [16] [17].

There has been extensive research in proposing TEQ algorithms. A literature survey of TEQ design methods is given in Chapter 2 However there remains need for further work to improve the convergence of the blind adaptive channel shortening algorithms This is the focus of the thesis.

1.2 Organization of the thesis

The remainder of the thesis is organized as follows Chapter 2 presents a literature survey of the channel shortening algorithms

Chapter 3 proposes techniques to improve the convergence of the SLAM algorithm The SLAM algorithm is a low complexity channel shortening approach as it minimizes the square of only a single fixed autocorrelation value. This chapter in particular details the moving average (MA) and autocorrelation (AR) implementations of the SLAM algorithm but later uses the MA implementation for faster convergence of the SLAM algorithm developed in the chapter. Two schemes are suggested to improve the convergence of the adaptive SLAM algorithm The first one variable-step-SLAM (VS-SLAM) uses a variable step at each iteration of the algorithm. The step size is selected automatically according to the value of the cost at each iteration The second scheme quasi-Newton SLAM (QN-SLAM) achieves a faster quadratic type convergence using

a Newton descent type update. The computational complexity and memory requirements of SLAM, VS-SLAM, and QN-SLAM are provided It is shown that VS-SLAM has identical complexity as SLAM, whereas QN-SLAM has quadratic complexity in the TEQ length. The proposed two algorithms are compared with SLAM by shortening 8 carrier serving area (CSA) Loop wireline channels Both proposed algorithms successfully shorten the CSA Loop channels. The channel shortening effect and the resulting TEQ designs are shown in the simulations section Achievable bit rate is used as the performance metric to assess the convergence rate of the algorithms. The details of how the achievable bit rate is calculated are provided The results show that on average VS-SLAM converges faster than the SLAM algorithm for all 8 CSA Loop channels QN-SLAM is faster than SLAM and sometimes converges earlier than the SAM algorithm. However, its response can be very noisy. The noisy convergence coupled with the very high computational complexity of the QN-SLAM algorithm makes it less useful for real time channel shortening applications. VS-SLAM appears to be the preferred algorithm, but SALM-type algorithms can suffer ill-convergence.

Chapter 4 proposes an exponential probability generalized lag hopping version of the SLAM algorithm named EGLHSAM The drawback with SLAM algorithm is that it minimizes a fixed autocorrelation value. There can be some channel impulse responses where the SLAM cost is zero but the channel impulse response is not confined to the required window length. EGLHSAM overcomes this problem by minimizing a random lag at each iteration from the available range of lags. Therefore, in a complete adaptation, it visits all the possible lags. This

reduces the possibility that EGLHSAM cost is zero but channel is not short as required resulting in a poor SIR The algorithm selects the lags with a probability matching the envelope of the impulse response of the underlying channel This increases the initial convergence rate of the EGLHSAM algorithm over that of the SLAM algorithm. The chapter gives a breakdown of the SIR formula and shows that only minimizing a fixed autocorrelation, as in SLAM, does not provide guarantee that SIR will be increased There is a possibility that few taps outside the required window are left which is against the channel shortening phenomenon The histograms of the lags simulated are shown. The EGLHSAM algorithm is compared with SLAM by shortening 8 CSA Loop wireline channels. Different decaying slopes for the lags are simulated for the EGLHSAM algorithm. It successfully shortens the 8 CSA Loop channels. The channel shortening effect and the resulting TEQ designs are shown in the simulations section. Achievable bit rate is again used as the performance metric to assess the convergence rate of the algorithms. Depending upon the decaying slope of the lags, EGLHSAM outperforms SLAM. This 'good' decaying parameter value is different for different CSA Loop channels This is a problem with the EGLHSAM algorithm where it needs the optimum decaying parameter value. It is also mentioned that using a highly decaying nature of the lags probability excludes some of the lags to be minimized. However, this is less severe problem than with the SLAM algorithm.

Chapter 5 proposes a generalized lag hopping algorithm but uses uniform probability of lag selection The algorithm is named GLHSAM and it overcomes problems with the SLAM and EGLHSAM algorithm to guarantee high SIR. This algorithm also does not need to know

the decaying parameter for every channel The GLHSAM algorithm is shown to have identical channel shortening effect as that of the SLAM and EGLHSAM algorithm The convergence rate of GLHSAM is better than SLAM and is comparable to that of EGLHSAM The convergence rate can be further increased by incorporating more lags in the update while keeping an overall uniform probability of lags selection.

Chapter 6 concludes this thesis and points out possible areas for further research

Chapter 2

LITERATURE SURVEY

2.1 Overview

The purpose of this chapter is to review the previous work in the field of channel shortening. The minimum mean square error (MMSE) method is discussed in Section 2.2. The maximum shortening signal-to-noise ratio (SSNR) method is then introduced and compared with the MMSE technique in Section 2.3. Algorithms for time-domain adaptive channel shortening are introduced in Section 2.4, the enhancement of which, is the focus of this thesis. Finally, in Section 2.5 alternative frequency domain methods are discounted due to their computational complexity.

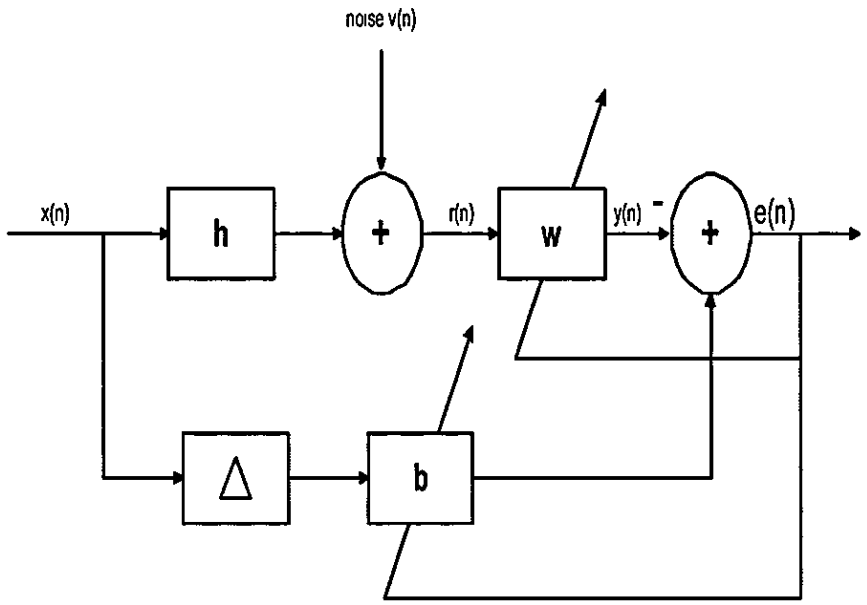

Figure 2.1. Block diagram of an MMSE channel shortening system

2.2 Minimum Mean Square Error Method

Figure 2.3 illustrates the structure of the time-domain equalizer (TEQ) design method reported in [18]. In the block diagram, $\mathbf{b} = [b_0, b_1 \cdots b_{n_b-1}]^T$ is defined as the target impulse response (TIR), where $(\,)^T$ denotes vector transpose. Also, $\mathbf{w} = [w_0, w_1, \cdot\ \cdot w_{n_w-1}]^T$, is defined as the TEQ parameter or weight vector which is designed to drive the mean squared error between the system output and the delayed output of $\mathbf{b}$ to a minimum The channel impulse response vector is represented as $\mathbf{h} = [h_0, h_1, h_2, ...h_{n_h-1}]^T$. The received signal, in vector form, therefore becomes

$$\mathbf{r_n} = \mathbf{H}\mathbf{x}_n + \mathbf{v}_n \tag{2 2 1}$$

where $\mathbf{r}_n = [r(n), r(n-1) \cdots r(n-n_w+1)]^T$, $\mathbf{x}_n = [x(n), x(n-1) \cdots x(n-n_h-n_w+1)]^T$, $\mathbf{v}_n = [v(n), v(n-1) \cdots v(n-n_w+1)]^T$ and $\mathbf{H}$ is the Toeplitz convolution matrix given by

$$\mathbf{H} = \begin{pmatrix} h_0 & h_1 & \cdots & h_{n_h-1} & 0 & \cdot\ \cdot & 0 \\ 0 & h_0 & h_1 & \cdots & h_{n_h-1} & \cdots & 0 \\ \vdots & \ddots & \cdot & \ddots & 0 & \ddots & \vdots \\ 0 & \cdots & 0 & h_0 & h_1 & \cdots & h_{n_h-1} \end{pmatrix}$$

In channel shortening, the elements of $\mathbf{x}_n$ are zero mean, unit variance, data symbols, $x(n)$, which form the input to the channel $\mathbf{h}$. Zero-mean additive white Gaussian noise is included in the model equation (2 2.1) as the elements of $\mathbf{v}_n$ All signals throughout this thesis are real-valued

The mean squared error is therefore

$$\begin{aligned} E[e^2(n)] &= E[(\mathbf{w}^T\mathbf{r}_n - \tilde{\mathbf{b}}^T\mathbf{x}_n)^2] \\ &= E[(\mathbf{w}^T\mathbf{r}_n - \tilde{\mathbf{b}}^T\mathbf{x}_n)(\mathbf{r}_n^T\mathbf{w} - \mathbf{x}_n^T\tilde{\mathbf{b}})] \\ &= \mathbf{w}^T\mathbf{R}_{rr}\mathbf{w} + \tilde{\mathbf{b}}^T\mathbf{R}_{xx}\tilde{\mathbf{b}} - 2\mathbf{w}^T\mathbf{R}_{rx}\tilde{\mathbf{b}} \end{aligned} \quad (2.2\ 2)$$

where $\tilde{\mathbf{b}} = [\mathbf{0}_{1\times\Delta}\mathbf{b}^T\mathbf{0}_{1\times s}]^T$ and Δ is the delay parameter. The terms $R_{xx} = E[\mathbf{x}_n\mathbf{x}_n^T]$, $R_{rx} = E[\mathbf{r}_n\mathbf{x}_n^T]$, and $R_{rr} = E[\mathbf{r}_n\mathbf{r}_n^T]$ are respectively the transmission signal autocorrelation, channel output/input cross correlation and the channel output autocorrelation matrices. To find the optimal MMSE solution for $\mathbf{w}$, differentiate equation (2 2 2) and equate the result to the zero vector,

$$\begin{aligned} \frac{\partial E[e^2[n]]}{\partial \mathbf{w}} &= R_{rr}\mathbf{w} - R_{rx}\tilde{\mathbf{b}} = \mathbf{0} \\ R_{rr}\mathbf{w} &= R_{rx}\tilde{\mathbf{b}} \end{aligned} \quad (2.2.3)$$

and therefore

$$\mathbf{w}_{opt} = R_{rr}^{-1}R_{rx}\tilde{\mathbf{b}} \quad (2.2.4)$$

Substituting equation (2 2.4) into equation (2.2 2)

$$E[e^2(n)] = \tilde{\mathbf{b}}^T(R_{xx} - R_{rx}^T(R_{rr}^{-1})^T R_{rx})\tilde{\mathbf{b}} \quad (2.2.5)$$

which is the minimum mean square error. Solution of equation (2.2.4) assumes knowledge of the TIR, which can be found from the system requirements A good literature survey of the MMSE TEQ design technique is reported in [19]. Most of the reported techniques in [19] focus

on minimizing the complexity of MMSE channel shortening by exploiting the structure of the terms in equation (2.2.4).

The TEQ design method for frequency division multiplexed asynchronous digital subscriber line (ADSL) (FDM-ADSL) is slightly different from that for echo cancelled ADSL (EC-ADSL). In FDM-ADSL, separate frequency bands are allocated for downstream and upstream transmission Sharp filters are employed in the analogue front end of the receiver to achieve this philosophy In EC-ADSL, overlapping spectra are used for downstream and upstream transmission whilst echo-cancelling is applied in [20]. Some researchers note that the MMSE TEQ can possess high gain in the FDM-ADSL stop-band region (the upstream transmission band) as reported in [21] The output of the discrete fourier transform, applied after the TEQ, possesses relatively high spectral side-lobes as reported in [20] The TEQ can then boost the stop band noise or the near end cross talk from the local upstream transmission dramatically which can drop the signal-to-noise ratio (SNR) of the active sub-carriers. The authors in [21] therefore modified the MMSE cost function to achieve suppression of the TEQ energy in the stop-band Their simulation results showed a more than 35 percent increase in the bit rate of the system after modification

Another issue in relation to the MMSE channel shortening method is reported in [22]. This method shortens the channel by minimizing the difference between the TIR and the effective impulse response of the system. In particular, it aims to minimize the difference inside and outside the target window. However, the difference inside the target window is supposed not to cause any ISI. Moreover, the TIR and effective impulse response generally possess larger magnitude inside the

target window than outside the target window. This means that the MMSE method initially tries to minimize the difference inside the window, which doesn't cause ISI, more than outside the target window, which causes ISI. Therefore, minimizing the MMSE to remove ISI is not always the best choice to design a TEQ for a discrete multi-tone modulation system.

2.3 Maximum Shortening Signal-to-Noise Ratio (MSSNR) Method

Generally, perfect shortening of the impulse response is not possible to achieve [23] [24] [25] [26] [27] Some energy will remain outside the largest $(v + 1)$ consecutive samples of the effective channel. The main aim is generally to drive as much as possible of the effective impulse response of the channel to remain inside (v+1) consecutive samples The MSSNR TEQ design method reported in [2] tries to maximize the ratio of the effective channel impulse response energy within a target window length (v+1) consecutive samples to the energy of the channel outside of the window. By referring to Figure (2.2), the effective channel impulse response can be written as in

$$\mathbf{h}_{eff} = \mathbf{c} = \mathbf{h} * \mathbf{w} \tag{2.3 1}$$

where the $(*)$ denotes linear discrete time convolution. The shape of the resulting impulse response of the effective channel $\mathbf{h}_{eff}$ is generally unimportant, what is important is that the SSNR be maximized The result of the MSSNR channel shortening method is illustrated in Figure (2 2) If $\mathbf{H}$ denotes the convolution matrix of the original channel $\mathbf{h}$, then the effective channel $\mathbf{h}_{eff} = \mathbf{Hw}$ as reported in [2]. The ef-

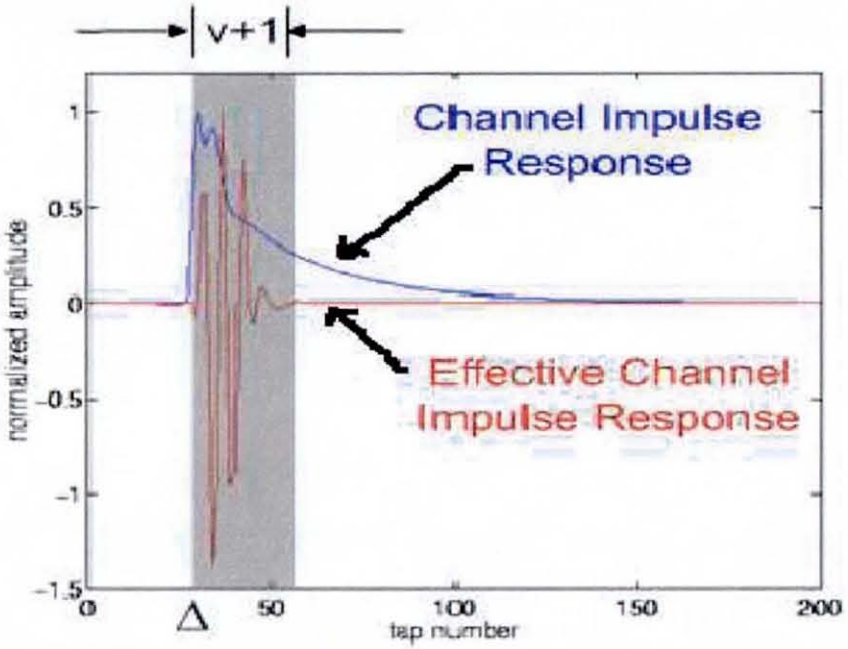

Figure 2.2. Original and shortend normalized channel using the MSSNR method, where Δ is the transmission delay [2].

fective channel can then be partitioned into two parts, the first part is the channel samples lying within the desired $(v+1)$ window which is denoted by $\mathbf{h}_{win} = \mathbf{H}_{win}\mathbf{w}$, while the second part consists of the channel samples lying outside this desired window which is denoted by, $\mathbf{h}_{wall} = \mathbf{H}_{wall}\mathbf{w}$ $\mathbf{H}_{win}$ consists of $(v+1)$ rows of $\mathbf{H}$ starting from position Δ, where Δ is the transmission delay, and $\mathbf{H}_{wall}$ consists of the remaining rows of $\mathbf{H}$. The SSNR is defined as

$$SSNR = \frac{\mathbf{w}^T\mathbf{H}_{win}^T\mathbf{H}_{win}\mathbf{w}}{\mathbf{w}^T\mathbf{H}_{wall}^T\mathbf{H}_{wall}\mathbf{w}} = \frac{\mathbf{w}^T\mathbf{B}\mathbf{w}}{\mathbf{w}^T\mathbf{A}\mathbf{w}} \tag{2.3 2}$$

The shortening is achieved by minimizing the wall energy (the denominator) while keeping the window energy (the numerator) equal to unity. If the length of the TEQ is smaller than $(v+1)$, matrix $\mathbf{B}$ is positive definite and can be decomposed by a Cholesky decomposition [28].

$$\begin{aligned} B &= \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T \\ &= (\mathbf{Q}\mathbf{\Lambda}^{1/2})(\mathbf{\Lambda}^{1/2}\mathbf{Q}^T) \\ &= (\mathbf{Q}\mathbf{\Lambda}^{1/2})(\mathbf{Q}\mathbf{\Lambda}^{1/2})^T \\ &= (\mathbf{B}^{1/2})(\mathbf{B}^T)^{1/2} \end{aligned} \tag{2.3 3}$$

where Λ is a diagonal matrix of eigenvalues of $\mathbf{B}$ and $\mathbf{Q}$ is matrix of orthonormal eigenvectors vectors. Let us denote

$$\mathbf{s} = (\mathbf{B}^T)^{1/2}\mathbf{w} \tag{2.3.4}$$

and

$$\mathbf{w} = (\mathbf{B}^T)^{-1/2}\mathbf{s} \tag{2.3.5}$$

Then substituting equation (2.3.5) into equation (2 3 2)

$$SSNR = \frac{\mathbf{s}^T\mathbf{s}}{\mathbf{s}^T\mathbf{C}\mathbf{s}} \tag{2.3 6}$$

where $\mathbf{C} = (\mathbf{B}^{-1/2})\mathbf{A}(\mathbf{B}^T)^{-1/2}$. The MSSNR TEQ method minimizes the denominator of equation (2 3 6) while setting its numerator equal to unity. This minimization gives the eigenvector $\mathbf{s}_{min}$ corresponding to the minimum eigenvalue of the matrix C. The resulting TEQ is given by equation (2 3 5)

$$\mathbf{w}_{opt} = (\mathbf{B}^T)^{-1/2}\mathbf{s}_{min} \tag{2.3 7}$$

The MSSNR method requires knowledge of the channel while it does not take into account the noise present in the channel Maximizing SSNR does not necessarily maximize the data rate [2]. The choice of the transmission delay, Δ, which gives the best SSNR is computationally expensive. There is a difference between the MMSE method and the MSSNR method. As stated before, the error definition in the MMSE method also includes the difference between the effective channel and the target channel inside the window of interest. Therefore minimizing the MSE does not necessarily minimize the effective channel wall energy When the length n_w, exceeds the length of the cyclic prefix, the matrix $\mathbf{B}$ becomes singular and $(\mathbf{B})^{-1/2}$ does not exist. In [29] it was suggested to maximize the energy inside the window i e , $\mathbf{w}^T\mathbf{B}\mathbf{w}$ while keeping the energy outside the window i.e , $\mathbf{w}^T\mathbf{A}\mathbf{w}$ equal to unity. The matrix $\mathbf{A}$ is always positive definite and the arbitrary length TEQ can be selected to obtain the required performance gains The authors of [30] investigate further the work reported in [29] in the presence of white Gaussian noise and near/far-end crosstalk Although their simulations show that a

longer length TEQ increases the SSNR; it may not necessarily improve the sub-channel SNR which is directly proportional to the data rate. This again shows the inadequacy of increasing the SSNR to maximize the bit rate of the system.

A low complexity sub-optimal divide and conquer TEQ algorithm was reported in [31]. This method separates the design of a long length TEQ into a series of two-tap TEQs The cost function in each iteration is the channel energy outside the window of interest and is changed in each iteration by the two-tap TEQ used in the previous iteration [19]. This cost function is the same as the denominator of the SSNR. The final TEQ is the convolution of all the TEQs designed at each step. This method eliminates the need for matrix inversion as in the MSSNR method and hence it is less computationally complex.

In [32] the MMSE and the MSSNR methods are compared and it was illustrated that under the assumption of white input, both the methods are equivalent. The MMSE is better than MSSNR only if implemented adaptively with an infinitesimal small step size and the noise is assumed white, then the amount of noise added is small in the formulation of its matrices.

According to [33], the part of the channel response exceeding the CP length which causes ISI and ICI depends not only on its energy but also on its distance from the guard interval. Therefore, their cost function not only includes the energy of the taps of the channel outside the window of interest, but also their distances from the time center of the original channel impulse response They use the term "delay spread equalizer" as opposed to in the MSSNR method where the expression "energy equalizer" is used. Their simulations show improvement in

that the SNR, distribution and the noise shaping by the TEQ at the sub-channels does not have notches The delay spread "equalizer" also has less sensitivity to the symbol synchronization errors. However, there is no explicit dependency on the inclusion of the channel-induced additive noise or the synchronization error in their design framework In [34], the algorithm of [33] was augmented. Their formulation of a TEQ algorithm explicitly included the noise and gave new penalizing functions for the delay spread of the effective channel. The objective function J is a convex combination of the channel shortening objective and noise-to-signal objective i e ,

$$\begin{aligned} J &= \alpha J_{short} + (1-\alpha) J_{noise} \\ &= \alpha \frac{\sum_n f(n-n_{mid})|\mathbf{h}_{eff}|^2}{\sum_n |\mathbf{h}_{eff}|^2} \\ +(1-\alpha)\frac{\sigma^2_{noise}}{\sigma^2_{signal}\sum_n |\mathbf{h}_{eff}|^2} & \end{aligned} \tag{2.3 8}$$

where $\alpha \in [0,1]$, n_{mid} is the time center of $\mathbf{h}_{eff}$, and $f(n)$ is a penalty function which penalizes the effective channel taps away from the time center n_{mid} The shortening cost function penalizes all of the taps and not only the taps outside the window. The simulations show some improvement in the data rates over those of [32] but there again notches appear in the sub-channel SNR plot. In [34], the authors extended their work to MIMO implementation and the penalizing function is also changed to take into account only the taps outside the window. The spectral flatness of the TEQ in the MSSNR cost function is included in [35] The implicit flatness measure is the distance of the effective channel impulse response $\mathbf{h}_{eff}$ from the original channel impulse response $\mathbf{h}$ The resulting TEQ does not have nulls in the frequency

domain Although this method shows lower SSNRs achieved as compared to the original MSSNR method, it results in higher data rates. The authors also recommended that the selection of the transmission delay should be to maximize the SSNR rather than to maximize the SSNR and the flatness

2.4 Property restoral blind adaptive channel shortening algorithms

In [36], a blind adaptive channel shortening algorithm based on the redundancy arising from the CP in the transmitted signal is proposed. The algorithm is called multicarrier equalization by restoration of redundancy (MERRY) [37] [38]. The following is true for the transmitted OFDM symbol in Figure 1.1

$$x[(N+v)k+i] = x[(N+v)k+N+i] \qquad i \in \{1,2, \quad ,v\} \quad (2.4\ 1)$$

where k is the symbol index. The input of the TEQ, r(n) is given by

$$r(n) = \sum_{j=0}^{L_h} h(j)x(n-j) + v(n) \qquad (2\ 4\ 2)$$

where $L_h + 1$ is the length of the channel impulse response, and $v(n)$ is the noise sample at index n The output of the TEQ, $y(n)$, is given by

$$y(n) = \sum_{j=0}^{L_w} w(j)r(n-j) \qquad (2\ 4\ 3)$$

where L_w+1 is the length of the TEQ. The ISI destroys the relationship in equation (2.4 1) as the channel that is longer than v samples will introduce energy into the sample $x[(N+v)\ k+v]$ at the receiver that

is not equal to the energy received by its dual $x[(N+v)\ k+v+N]$ at the end of the DMT frame. Ignoring the symbol index k for simplicity reasons, the cost function can be defined as

$$J_{merry}(\Delta) = E(y(v+\Delta) - y(v+N+\Delta))^2 \qquad (2\ 4.4)$$

where Δ is the transmission delay MERRY only updates once per symbol and its cost function depends on Δ It shortens the channel to v rather than $v+1$ samples. ISI free transmission is guaranteed as long as the effective channel is smaller than or equal to $v+1$. MERRY minimizes the energy outside of a length v window plus the energy of the filtered noise. In contrast, the MSSNR design minimizes the energy of the combined impulse response outside of a window of length $v+1$ without taking into account the noise MERRY is generalized to the so called forced redundancy with optional data omission (FRODO) algorithm [39]. FRODO uses more than one sample in the update rule and allows channel shortening of variable window lengths. The cost function is given by.

$$J_{frodo}(\Delta) = \sum_{i \in S_f} E(y(i+\Delta) - y(i+N+\Delta))^2 \qquad (2.4.5)$$

where $S_f \subset \{1,....v\}$. For MERRY $S_f = \{v\}$. The MERRY cost function analysis reported in [40] shows that it represents the effective channel energy outside a window of length v starting from the transmission delay Δ. It has been further recommended that if the number of comparisons made is more than one (the basic MERRY algorithm), then the "full" FRODO algorithm tries to suppress all of the channel taps except one. This is against the idea of channel shortening to a

desired window and actually works to shorten the channel to single impulse. Their simulation results also show that although using more than one term increases the convergence rate of the FRODO algorithm, it degrades its asymptotic performance. The MERRY and FRODO algorithms have also been applied to the MIMO case in [39]. Both the MERRY and FRODO cost functions depend upon the choice of the transmission delay Δ which the authors suggest to calculate by the following heuristic method

$$\Delta = \Delta_{peak} + \frac{\mathbf{L}_w}{2} \tag{2.4.6}$$

where Δ_{peak} is the delay which maximizes the energy of the un-shortened channel in a window of length $v+1$. As was mentioned earlier, the MERRY cost function represents the energy of the effective channel outside a window of length v. If there is no TEQ used, the cost function will represent the energy of the original channel outside a window of length v The index Δ_{peak} in which the energy of the channel inside the window is maximum is the index in which the energy of the channel outside the window will be minimum. Therefore Δ_{peak} can be estimated by transmitting κ symbols and evaluating [39]

$$\hat{\Delta}_{peak} = \min_{0 \le d \le s-1} \sum_{k=1}^{\kappa} \left(r(K\,k + v + d) - r(K.k + v + N + d) \right)^2 \tag{2.4.7}$$

where $s = N + v$ is the OFDM symbol duration. Substitution of equation (2.4.7) into equation (2 4 7) gives an estimate of the transmission delay Δ for MERRY and FRODO algorithms. This is a low complexity method to avoid the global search over the transmission delay parameter Δ and can be used for other TEQ methods as well The

authors in [41] [42] propose another blind, adaptive channel shortening algorithm sum squared autocorrelation minimization (SAM). SAM is based on minimizing the sum squared autocorrelation of the signal outside a window of length v at the output of the TEQ The cost function is given by:

$$J_{sam} = \sum_{l=v+1}^{L_c} \Big(R_{yy}(l)\Big)^2 \tag{2.4 8}$$

where $R_{yy}(l)$ is the autocorrelation of the output of the TEQ at lag l and L_c is the length of the effective channel ($\mathbf{c} = \mathbf{h} * \mathbf{w}$) minus one. Assuming an uncorrelated transmitted signal at the output of the IFFT block in Figure (1.2), if the channel is short, the autocorrelation of the output of the channel should also be short. The good things about SAM are, it is blind, adaptive, and independent of the transmission delay Δ. SAM converges faster than MERRY. SAM can track channel variations within a symbol because it can update once per sample while MERRY updates once every symbol. However, SAM has higher complexity than MERRY as can be seen in [41]

In [22, 43] a sub-channel SNR model is proposed

$$SNR_k = \frac{S_{x,k}|H_k^{signal}|^2}{S_{n,k}|H_k^{noise}|^2 + S_{x,k}|H_k^{ISI}|^2} \tag{2 4 9}$$

where H_k^{signal}, H_k^{noise} and H_k^{ISI} are the $k-th$ coefficients of the N point FFT of $\mathbf{h}_{win}$, $\mathbf{h}_{wall}$ and the TEQ w respectively and $S_{x,k}$ and $S_{n,k}$ are the k^th sub-channel power spectral densities of the signal and the noise before the equalizer The numerator contains the portion of the resulting transmission channel that contributes to the useful signal and the denominator includes the contribution of the ISI noise of the shortened

channel impulse response outside of the desired window. Defining the following

$$\begin{aligned} H_k^{signal} &= \mathbf{q}_k^H \mathbf{G}\mathbf{H}^1\mathbf{w} \\ H_k^{ISI} &= \mathbf{q}_k^H \mathbf{D}\mathbf{H}^1\mathbf{w} \\ H_k^{noise} &= \mathbf{q}_k^H \mathbf{F}\mathbf{w} \end{aligned} \tag{2.4 10}$$

wherein the $N \times T$ matrix $\mathbf{H}^1$ is the first N rows of the convolution matrix of the transmission channel, T denotes the length of the TEQ, and the diagonal $N \times N$ matrices $\mathbf{G}$ and $\mathbf{D}$ give the rows of the vector $\mathbf{Hw}$ corresponding to the desired $v+l$ window and outside of it, respectively, and the $N \times T$ matrix $\mathbf{F}$ when multiplied with $\mathbf{w}$ gives the TEQ vector $\mathbf{w}$ plus padding it with $N - T$ zeros. Multiplication with the vector $\mathbf{q}_k^H$ where $(\,)^H$ denotes Hermitian, or conjugate transpose gives the k^{th} coefficient, of the N point FFT. The subchannel SNR would then be·

$$\begin{aligned} SNR_k &= \frac{\mathbf{w}^T\mathbf{H}^T\mathbf{G}^T\mathbf{q}_k S_{x,k}\mathbf{q}_k^H\mathbf{G}\mathbf{H}\mathbf{w}}{\mathbf{w}^T\mathbf{F}^T\mathbf{q}_k S_{n,k}\mathbf{q}_k^H\mathbf{F}\mathbf{w} + \mathbf{w}^T\mathbf{H}^T\mathbf{D}^T\mathbf{q}_k S_{x,k}\mathbf{q}_k^H\mathbf{D}\mathbf{H}\mathbf{w}} \\ &= \frac{\mathbf{w}^T\mathbf{A}_k\mathbf{w}}{\mathbf{w}^T\mathbf{B}_k\mathbf{w}} \end{aligned} \tag{2 4 11}$$

The bit rate of the DMT system is given by

$$\begin{aligned} b_{dmt} &= \sum_{k=usedtone} log_2\Big(1 + \frac{SNR_k}{\Gamma}\Big) \\ &= \sum_{k=usedtone} log_2\Big(1 + \frac{1}{\Gamma}\frac{\mathbf{w}^T\mathbf{A}_k\mathbf{w}}{\mathbf{w}^T\mathbf{B}_k\mathbf{w}}\Big) \end{aligned} \tag{2 4.12}$$

where Γ denotes the SNR gap of the system and is assumed to be constant over all sub-channels. The maximum bit rate (MBR) algorithm

maximizes the non-linear bit rate The optimization toolbox within MATLAB was used to solve equation (2 4.12), and the matched filter bound (MFB) was achieved However, the authors concluded that the MBR method is computationally expensive. Therefore they proposed a low complexity near optimal min-ISI method. The min-ISI method reported in [22] introduces the idea of frequency weighting in the form of sub-channels. It shapes the frequency response of the TEQ. Specifically, it results in increased minimization of ISI noise on the sub-channels with higher SNRs. The simulations show that the min-ISI method achieves almost the same data rates as that of MBR method. The min-ISI TEQ is given by

$$\mathbf{w} = arg \min_{\mathbf{w}\,\mathbf{w}^T\mathbf{H}^T\mathbf{G}^T\mathbf{G}\mathbf{H}\mathbf{w}=1} \left[\mathbf{w}^T\mathbf{H}^T\mathbf{D}^T \sum_k \left(\mathbf{q}_k \frac{S_{x,k}}{S_{n,k}} \mathbf{q}_k^H\right)\mathbf{D}\mathbf{H}\mathbf{w}\right] \quad (2\,4\,13)$$

The value of the cost function increases in favour of the sub-channels with higher SNRs A small reduction in ISI power in these sub-channels will increase the bit rate. While in low SNR sub-channels, the noise is so dominant that decrease of ISI power does not have a big effect on the bit rate The min-ISI method is a generalization of the MSSNR method. The min-ISI method takes into account the frequency response of $\mathbf{h}_{wall}$ while the MSSNR method only looks at its energy.

Another interesting point to note is that the min-ISI method achieves almost 96% percentage of the matched filter bound data (MFB) rates with a TEQ length of only 3 taps. The authors then get maximum data rates with the min-ISI method using a small value of the CP and a longer TEQ. In this way they are successful in trading-off the reduction in the throughput of the system due to CP with the complexity of the

TEQ

2.5 Per Tone Equalization Scheme

In [44] an alternate equalization structure for multi-carrier systems is proposed where equalization is performed with a T-tap equalizer after the FFT for each tone/sub-carrier separately, hence the name per tone equalization (PTEQ) [45] [46] [47] [48]. A TEQ equalizes all the tones of a multi-carrier system in a combined fashion. The PTEQ scheme enables true signal-to-noise ratio (SNR) optimization to be implemented for each tone Their simulation results have compared the performance of the PTEQ scheme with the MMSE TEQ scheme. The achievable data rates are always higher with the PTEQ scheme and a smoother function of the transmission delay Δ as compared to the MMSE TEQ scheme. Therefore, PTEQ is not that sensitive to the symbol timing synchronization (Δ_{peak}) estimation error. The PTEQ scheme has very large complexity during the initialization mode For DMT-based systems, it requires initialization of T x N/2 filter taps instead of only T taps as in the TEQ scheme. This also increases the memory requirements of the PTEQ scheme as compared to the TEQ scheme The symbol timing synchronization in TEQ schemes involves searching for the optimal delay around Δ_{peak} while it is equal to Δ_{peak} in the PTEQ scheme.

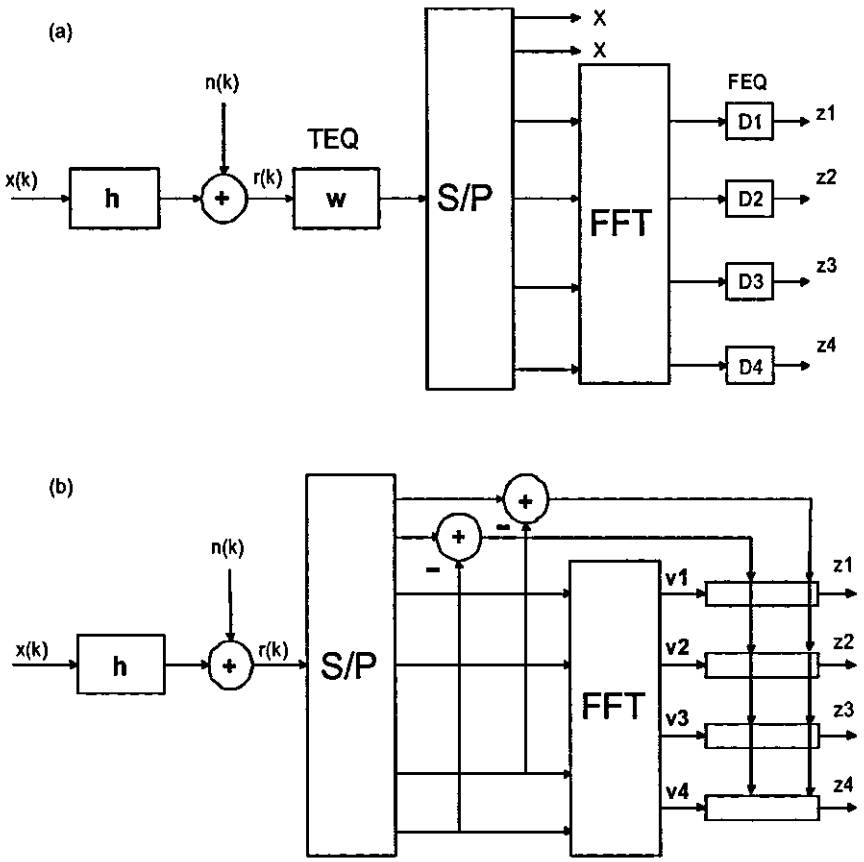

Figure 2.3. Comparison of TEQ and Per Tone structure of channel shortening.

The PTEQ scheme is generalized to MIMO in [49] PTEQ has been considered for channel shortening and equalization over doubly selective OFDM channels [50]. The non-adaptive implementation of the PTEQ scheme in [44] requires knowledge of the channel and the signal and noise statistics. Recursive least squares (RLS) and least mean square (LMS) [51] [52] adaptive implementation of the PTEQ scheme, which need training, have been suggested in [53]. The blind, adaptive version of the PTEQ scheme is discussed in [54] [55] by using the constant modulus algorithm (CMA) and the decision-directed-LMS (DD-LMS) algorithm. The per tone DD-LMS algorithm is given in [40] as

$$\begin{aligned} z_i(k) &= \bar{\mathbf{v}}_i^T(k)\mathbf{F}_i\mathbf{r}(k) \\ e_i(k) &= Q[z_i(k)] - z_i(k) \\ \bar{\mathbf{v}}_i(k+1) &= \bar{\mathbf{v}}_i(k) + \mu e_i(k)\mathbf{F}_i^*\mathbf{r}^*(k) \end{aligned} \tag{2 5.1}$$

where $i = 1, \quad , N$ is the subchannel index, $k = 1, 2, 3..$ is the symbol index, and $Q[\cdot]$ is the quantization or decision device $z_i(k)$ is the equalized output for subchannel i $\bar{\mathbf{v}}_i^T = [v_{i,T-1} \cdots V_{i,0}]$ is the T-tap reversed PTEQ equalizer for the subchannel i. The vector $\mathbf{F}_i.\mathbf{r}(k)$ contains in reverse order $(T-1)$ required difference terms extracted from the received vector $\mathbf{r}(k)$ in its first $(T-1)$ entries, and the i-th value of the FFT in its last entry

The authors of [54] suggest to use first the CMA PTEQ and then the DD-LMS PTEQ during the initialization of the equalizer. The simulation shows the characteristics of the SNR distribution on one of the subchannels as a function of the symbol timing synchronization error. The SNR distribution is relatively constant over a range of negative

synchronization error δ values and drops in magnitude for the positive synchronization errors.

The SNR improvement by the PTEQ scheme over the TEQ schemes is more pronounced at higher SNR subchannels of the unequalized channel. To find a better tradeoff between complexity and bit rate, [56] propose a dual-path TEQ scheme Two TEQ filters are designed such that one TEQ equalizes over the entire bandwidth while the other one optimizes over a selected frequency band. The dual-path TEQ structure passes the received data through two paths instead of one path Each path has its own TEQ, FFT and one-tap FEQ The selective band TEQ SNR. The TEQ that equalizes over the entire bandwidth can be designed using any of the TEQ design methods such as MMSE or MSSNR. The selective band TEQ would need to be designed using a method that allows frequency selective weighting such as Min-ISI. The simulations show a 4% increase in bit rates over a single path TEQ.

The TEQ-filter bank (TEQ-FB) of [57] is another algorithm similar to the PTEQ scheme where each subchannel has its own filter but in the time domain. After the TEQs, the transfer to the frequency domain is performed using a bank of Goertzel filters, each one tuned to the frequency of the desired subchannel and computing a single point DFT coefficient. This method may have lower memory needs than the PTEQ scheme but its computational requirements are significantly higher during data transmission mode [57] Their simulations show a slightly better performance than PTEQ.

In [58] a blind adaptive equalization algorithm for OFDM systems which exploits the null carriers present in the system is proposed. This carrier nulling algorithm is based on minimizing a quadratic criterion

based on the energy of the null carriers

$$J = \sum_{i=nullcarriers} E[|Y_i|^2] \qquad (2\,5\,2)$$

where Y_i is the received signal after the FFT in the i subchannel. A unit norm constraint is imposed on the equalizer to avoid the trivial solution This shortens the channel to a single spike i e., complete equalization The algorithm does not require the transmission of CP The use of the blind term for this algorithm is debatable, as transmission of zeros on certain carrier could be thought of as training signal consisting of zeros

The fundamentals algorithms for blind adaptive channel shortening have been introduced. the SAM-type algorithm will now be the focus of the thesis due to its improved convergence properties over MERRY and its relatively low complexity as compared to frequency domain approaches.

Chapter 3

FAST CONVERGING SINGLE LAG AUTOCORRELATION MINIMIZING ALGORITHMS FOR REAL TIME CHANNEL SHORTENING IN WIRELINE SYSTEMS

3.1 Overview

A blind adaptive channel shortening algorithm based on minimizing the squared single lag autocorrelation (SLAM) of the effective channel was recently proposed [59]. Two approaches are presented in this chapter to improve the convergence of SLAM. Their suitability for real-time implementation is a focus of the work so computational efficiency and memory requirements are considered In the first approach, a time-varying step size algorithm is derived on the basis of the work of Mathews [60] In the second approach, a quasi-Newton algorithm is derived.

Simulations studies for CSA loop wireline channels are used to confirm the utility of the schemes.

3.2 Introduction

In [59] the authors propose a low complexity algorithm called SLAM for blind channel shortening which belongs to the class of property restoral algorithms defined in [61]. The channel needs to be shorter than CP, therefore, the channel should also have a autocorrelation shorter than CP, Assuming an uncorrelated source at the transmitter, SLAM tries to fulfil this property by shortening the autocorrelation of the output data.

SLAM is an algorithm that aims to achieve channel shortening by minimizing the square of only a single autocorrelation value. It is difficult to make all channel taps zero outside the CP width window. What is possible is to maximize the SSNR of the effective channel. The initial choice of the adaptation gain has a marked effect on the convergence of SLAM due to the multimodality of the underlying cost function [1] In this work, methods are therefore investigated to automate the selection of the adaptation gain and investigate their suitability for real-time implementation.

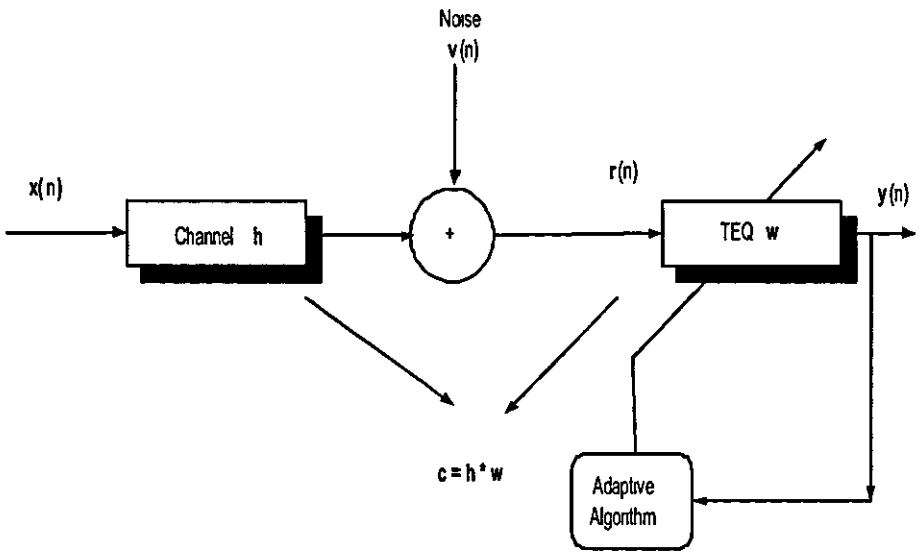

Figure 3.1. Overall baseband channel shortening system model.

3.3 System Model

The system model is shown in Figure 3.1. The signal x(n) is a white, zero mean, wide-sense stationary (W.S.S), real and unit variance source sequence which is then transmitted through the linear finite-impulse response (FIR) channel $\mathbf{h} = [h(0)h(1)...h(L_h)]^T$,v(n) is a zero mean i.i.d , noise sequence uncorrelated with the source sequence and has variance σ_v^2. The received signal $r(n)$ is

$$r(n) = \sum_{k=0}^{L_h} h(k)x(n-k) + v(n) \tag{3 3.1}$$

and $y(n)$, the output of the TEQ is given by

$$y(n) = \sum_{k=0}^{L_w} w(k)r(n-k) = \mathbf{w}^T\mathbf{r}_n \tag{3 3.2}$$

where w is the impulse response vector of the TEQ
$\mathbf{w} = [w(0)w(1)...w(L_w)]^T$, and $\mathbf{r}_n = [r(n)r(n-1)..r(n-L_w]^T$. L_h, L_c, and L_w are the order of the channel, effective channel and the TEQ respectively. It is also assumed that $2L_c \leq N$ holds, N being the FFT size [59] which is a reasonable assumption in the case of ADSL.
The focus of this work is the design of unsupervised/blind learning algorithms for the time domain equalizer (TEQ) shown in Figure 3.1, to achieve overall channel shortening, i e. essentially reducing the effective length of the combined channel c, to some design requirement.

3.3.1 SLAM algorithm

The cost function of SLAM is defined as

$$J_{SLAM} = R_{yy}(l)^2, l = v+1 \tag{3 3 3}$$

where $R_{yy}(l)$ is the autocorrelation function of the real channel output i.e. $E\{y(n)y(n-l)\}$, and v is the CP length and E denotes the statistical expectation operator.

A steepest gradient-descent type algorithm can be used to minimize J_{SLAM}, i.e.,

$$\mathbf{w}(n) = \mathbf{w}(n-1) - \mu \nabla_{\mathbf{w}} J_{SLAM}(n-1) \tag{3 3 4}$$

and μ is the step size, and $\nabla_{\mathbf{w}} J_{SLAM}(n-1)$ is the gradient of J_{SLAM} with respect to $\mathbf{w}(n-1)$. Using equation (3 3 3), it can be calculated as

$$\nabla_{\mathbf{w}} J_{SLAM}(n-1) = 2R_{yy}(l)\frac{\partial R_{yy}(l)}{\partial \mathbf{w}} \tag{3 3 5}$$

where using (5 3.2), it is written as

$$\begin{aligned} \frac{\partial R_{yy}(l)}{\partial \mathbf{w}} &= \frac{\partial E\{y(n)y(n-l)\}}{\partial \mathbf{w}} \\ &= E\{y(n)\mathbf{r}(n-l) + y(n-l)\mathbf{r}(n)\} \end{aligned} \tag{3.3 6}$$

Then, using (3 3 5) and (3 3 6), the update (3 3 4) can be written as·

$$\mathbf{w}(n) = \mathbf{w}(n-1) - 2\mu R_{yy}(l) E\{y(n)\mathbf{r}(n-l) + y(n-l)\mathbf{r}(n)\} \tag{3.3.7}$$

In the real-time implementation the $E\{.\}$ operator can be realized by either auto regressive or moving average forms. The practical update

equation of the moving average (MA) form is given as [40]

$$\mathbf{w}(k) = \mathbf{w}(k-1) - 2\mu\Big\{\sum_{n=kN_{avg}}^{(k+1)N_{avg}-1} \frac{y(n)y(n-l)}{N_{avg}}\Big\} \times \Big\{\sum_{n=kN_{avg}}^{(k+1)N_{avg}-1} \frac{y(n)\mathbf{r}(n-l)+y(n-l)\mathbf{r}(n)}{N_{avg}}\Big\} \quad (3\,3\,8)$$

where $l = v + 1$, k is the averaging block number and the averaging window length N_{avg} is the design parameter which determines the algorithm complexity and the accuracy. A fixed step size is not appropriate for situations where statistics of the measured data change, new variable step size algorithms are therefore proposed for SLAM.

Another way of implementing the SLAM algorithm is by using the auto-regressive (AR) estimates. Let

$$\mathbf{a}^n = (1-\lambda)\mathbf{a}^{n-1} + \lambda y(n) \begin{bmatrix} r(n-v-1) \\ \vdots \\ r(n-v-1-L_w) \end{bmatrix}$$

$$b^n = \mathbf{w}^T\mathbf{a}^n$$

$$\mathbf{c}^n = (1-\lambda)\mathbf{c}^{n-1} + \lambda y(n-v-1) \begin{bmatrix} r(n) \\ \vdots \\ r(n-L_w) \end{bmatrix}$$

where $0 < \lambda < 1$ is a forgetting factor and is a design parameter. Using these AR estimates and equation (5.3 2), the update rule of equation (3.3.7) can be written as

$$\mathbf{w}(n) = \mathbf{w}(n-1) - 2\mu\, b^n.\{\mathbf{a}^n + \mathbf{c}^n\} \quad (3\,3\,9)$$

The advantage of AR implementation is that the TEQ is updated at every time instant rather than at every N_{avg}^{th} time instant as is the case with the MA implementation, however, due to its straightforward form, the MA approach is used in this work.

3.4 Accelerating the convergence of SLAM

Two schemes are therefore proposed to increase the convergence rate of the SLAM algorithm, using MA implementation.

3.4.1 Variable Step SLAM (VS-SLAM)

In order to automatically update the step size μ as in [60], the update equation becomes

$$\mu(n) = \mu(n-1) - \rho \nabla_{\mu} J_{SLAM}(n-1) \tag{3.4.1}$$

where ρ is a learning rate. The gradient of J_{SLAM} with respect to μ can be implemented as:

$$\nabla_{\mu} J_{SLAM}(n-1) = 2R_{yy}(l)\frac{\partial R_{yy}(l)}{\partial \mu} \tag{3.4.2}$$

where $\frac{\partial R_{yy}(l)}{\partial \mu}$ is given by

$$\frac{\partial R_{yy}(l)}{\partial \mathbf{w}}\frac{\partial \mathbf{w}}{\partial \mu} \tag{3.4.3}$$

The first term is calculated as in equation (3.3.6) and the gradient $\partial \mathbf{w}/\partial \mu$ can be calculated by differentiating equation (3.3.4) with respect to μ:

$$\frac{\partial \mathbf{w}}{\partial \mu} = -\nabla_{w} J_{SLAM}(n-1) \tag{3.4.4}$$

Substituting equations (3 4 2), (3.4.3), and (3 4 4), into the update rule (3.4.1), the step size update is obtained as

$$\mu(n) = \mu(n-1) + 2\rho\,[R_{yy}(l)E\{y(n)\mathbf{r}(n-l) + y(n-l)\mathbf{r}(n)\}]^T \\ [R_{yy}(l)E\{y(n)\mathbf{r}(n-l) + y(n-l)\mathbf{r}(n)\}] \quad (3\,4.5)$$

3.4.2 QN-SLAM

Faster quadratic-type convergence can generally be obtained by using a Newton descent type update, which takes the form

$$\mathbf{w}(n) = \mathbf{w}(n-1) - \mu(\nabla^2_{\mathbf{w}} J_{SLAM}(n-1))^{-1}\nabla_{\mathbf{w}} J_{SLAM}(n-1) \quad (3.4\,6)$$

Equation (3.4.6) includes the second order gradient term $\nabla^2_{\mathbf{w}} J_{SLAM}(n-1)$ which is approximated in this work so as to form a Quasi-Newton algorithm

$$\nabla^2_{\mathbf{w}} J_{SLAM}(n-1) \cong 2(\nabla_{\mathbf{w}} J_{SLAM} \times (\nabla_{\mathbf{w}} J_{SLAM})^T + \\ R_{yy}(l) \times \mathbf{r}(n)(\mathbf{r}(n-l))^T + \alpha I) \quad (3.4.7)$$

where α is a parameter chosen to ensure a positive definite form; the trade-off for real time implementation is increased complexity together with the memory requirements.

3.5 Computational Complexity Comparisons

The estimated computational complexities of MA implementations of the SLAM, VS-SLAM and QN-SLAM algorithms are shown in Tables (3.1), (3.2) and (3.3) respectively. Their complexities are compared

in terms of number of multiplications and additions/subtractions per iteration of the algorithm. SLAM is shown to have complexity proportional to the averaging length N_{avg} and shortener length L_w. VS-SLAM has essentially identical complexity to SLAM, whereas QN-SLAM has quadratic complexity in Lw due to the use of the approximation to the Hessian matrix, clearly the convergence advantage of QN-SLAM is considerably offset by this increased complexity in a real-time application, and the calculation of the matrix inversion in QN-SLAM would make this situation even more accute. Likewise, in terms of estimated memory storage requirement for the three algorithms SLAM and VS-SLAM have essentially the same need, i.e. proportional to L_w, whilst QN-SLAM has a level proportional to L_w^2. In conclusion, therefore the potential convergence advantage of QN-SLAM is unlikely to justify its use in real-time applications unless further simplifications of the matrix and its inverse are introduced.

Steps	multiplications	addition subtractions
$N_{avg}, y(n-l)r_n$ terms	$N_{avg}.\{L_w+1\}$	-
$N_{avg}, y(n)r_{n-l}$ terms	$N_{avg}.\{L_w+1\}$	-
Accumulating above terms	-	$N_{avg}.\{L_w+1\}$ $+(N_{avg}-1)$ $\{L_w+1\}$
$N_{avg}, y(n)$ $y(n-l)$ terms	N_{avg}	-
Form gradient	-	$N_{avg}-1$
	1	-
	L_w+1	-
Weight update	-	L_w+1
Total complexity	$N_{avg}.\{2L_w+3\}$ $+L_w+2$	$N_{avg}\{L_w+2\}$ $+L_w$ $+(N_{avg}-1)$ $\{L_w+1\}$

Table 3.1. Estimated computational complexity for SLAM

Variable adaptation gain	Multiplication	Addition subtraction
	$L_w + 2$	$L_w + 1$
Total complexity	$N_{avg} \times \{2L_w + 3\}$ $+2L_w + 4$	$N_{avg}\{2L_w + 3\}$ $+L_w$

Table 3.2. Estimated additional computational complexity for VS-SLAM

Quasi Newton	Multiplication	Addition subtraction
	$N_{avg}\times$ $(L_w + 1)^2$ $+2(L_w + 1)^2$	$\{2N_{avg} - 1\}\times$ $\{L_w + 1\}^2$ $+2\{L_w + 1\}^2$
Total without matrix inversion	$N_{avg}\{2L_w + 3\}$ $+L_w + 2$	$N_{avg}\{2L_w$ $+3\} - 1$

Table 3.3. Estimated additional computational complexity for QN-SLAM

Algorithm	SLAM	VS-SLAM	QN-SLAM
Memory	$2(L_w + 1) + 1$	$2(L_w + 1) + 2$	$2(L_w + 1)+$ $1 + (L_w + 1)^2$

Table 3.4. Estimated memory storage requirements for the algorithms SLAM, VS-SLAM and QN-SLAM

3.6 Simulations

The standard parameters of an ADSL downstream transmission were simulated. An MA implementation was simulated for the SAM, SLAM, VS-SLAM and QN-SLAM algorithms. The value of N_{avg} was 32. The cyclic prefix had length 32. The FFT size $N_{fft} = 512$, the TEQ had 16 taps and the channels used were the eight test ADSL channels CSA loops provided at [62] [63]. The noise was chosen such that $\sigma_x^2\|\mathbf{c}\|^2/\sigma_v^2 =$ 40 dB where $\|\ \|$ denotes the Euclidean norm. Single spike initialization with the center spike of the TEQ initialized to unity was used. The step size for SAM was 5; whereas, for SLAM it was 600, to get SLAM algorithm to converge in the given number of symbols. The initial step size for VS-SLAM and QN-SLAM algorithms was also 600 The values of α and ρ were 0 1 and 1×10^4, respectively. All algorithms were compared with the maximum shortening SNR (MSSNR) solution and the matched filter bound (MFB) on capacity, which assumes no ICI. For a point-to-point system with bit loading, the achievable bit rate for a fixed probability of error (typically 10^{-7} in DSL) is the performance metric. The SNR gap Γ is given by

$$\Gamma = \Gamma_{gap} + \gamma_m - \gamma_c \tag{3 6 1}$$

The bit rate on each subcarrier is determined using noise margin $\gamma_m =$ 6dB and the coding gain $\gamma_c = 4.2$dB. The value of $\Gamma_{gap} = 9\,8$dB is used which corresponds to a probability of error 10^{-7} and the QAM modulation used across the sub carriers. The bit rate on each subcarrier

i is calculated based on

$$b_i = \log_2 \left(1 + 10^{((SNR_i - \Gamma)/10)}\right) \qquad (3\,6\,2)$$

The total bit rate is computed with the formula

$$rate = \left(\sum_{i=1}^{N_{fft}/2} b_i \right) \cdot \frac{F_s}{N_{fft} + v}$$

where $F_s = 2.208$ MHz is the sampling frequency. The achievable bit rate performance metric will be used to assess the performance of the TEQ algorithms developed in this thesis.

Figure (3 2) compares the achievable bit rates as a function of averaging block number by SAM, SLAM, VS-SLAM, and QN-SLAM algorithms for CSA loop 1. VS-SLAM outperforms SLAM in terms of maximum attained bite rate and QN-SLAM converges faster than VS-SLAM algorithm. Figure (3.3) compares the achievable bit rates as a function of averaging block number by SAM, SLAM, VS-SLAM, and QN-SLAM algorithms for CSA loop 2 Here again VS-SLAM converges faster than the SLAM algorithm. QN-SLAM converges very marginally earlier than the SAM algorithm.

Figure (3 4) compares the achievable bit rates as a function of averaging block number by SAM, SLAM, VS-SLAM, and QN-SLAM algorithms for CSA loop 3 VS-SLAM converges a little faster than the SLAM algorithm QN-SLAM converges quite abit earlier than VS-SLAM but its convergence is very noisy and most probably related to the approximation in the second derivative calculation. Figure (3.5) compares the achievable bit rates as a function of averaging block number by SAM,

SLAM, VS-SLAM, and QN-SLAM algorithms for CSA loop 4. Again VS-SLAM converges faster than the SLAM algorithm. QN-SLAM converges even earlier than the SAM algorithm.

Figure (3.6) compares the achievable bit rates as a function of averaging block number by SAM, SLAM, VS-SLAM, and QN-SLAM algorithms for CSA loop 5 VS-SLAM are SLAM are comparable in terms of convergence rate. QN-SLAM is quite noisy, though it converges quite early Figure (3.7) compares the achievable bit rates as a function of averaging block number by SAM, SLAM, VS-SLAM, and QN-SLAM algorithms for CSA loop 6. VS-SLAM is faster than SLAM algorithm while QN-SLAM is even faster than SAM and noisy, too

Figure (3 8) compares the achievable bit rates as a function of averaging block number by SAM, SLAM, VS-SLAM, and QN-SLAM algorithms for CSA loop 7. Again VS-SLAM converges faster than the SLAM algorithm. QN-SLAM converges even earlier than the SAM algorithm.

Figure (3 9) compares the achievable bit rates as a function of averaging block number by SAM, SLAM, VS-SLAM, and QN-SLAM algorithms for CSA loop 8. VS-SLAM converges a little bit faster than the SLAM algorithm QN-SLAM algorithm converges earlier than SLAM but its response is very noisy

Figure (3 10) shows the original and shortened CSA Loop 1 (top) and 2 (bottom) by the VS-SLAM algorithm. Similarly, Figures (3.11), (3.12), and (3 13) show the same for CSA Loop 3,4 and 5,6 and 7,8 respectively. Figure (3 14) shows steady state coefficients of the TEQ given by the VS-SLAM algorithm for CSA Loop 1 (left) and 2 (right) Figures (3 15), (3 16), and (3 17) show the same for CSA Loop 3,4 and 5,6 and 7,8, respectively.

Figure (3 18) shows the original and shortened CSA Loop 1 (top) and 2 (bottom) by the QN-SLAM algorithm. Similarly, Figures (3 19), (3.20), and (3 21) show the same for CSA Loop 3,4 and 5,6 and 7,8 respectively Figure (3.22) shows steady state coefficients of the TEQ given by the QN-SLAM algorithm for CSA Loop 1 (left) and 2 (right). Figures (3.23), (3.24), and (3 25) show the same for CSA Loop 3,4 and 5,6 and 7,8, respectively. All figures confirm the shortening performance of the proposed novel algorithms.

3.7 Summary

In this chapter, fast converging single lag minimizing autocorrelation algorithms for real time channel shortening have been proposed The QN-SLAM algorithm has been shown to have the fastest convergence rate however it has largest complexity and its convergence is very noisy. For real-time applications the VS-SLAM algorithm appears to behave the same as SAM. The results have been achieved by analysis on a benchmark standard test channel.

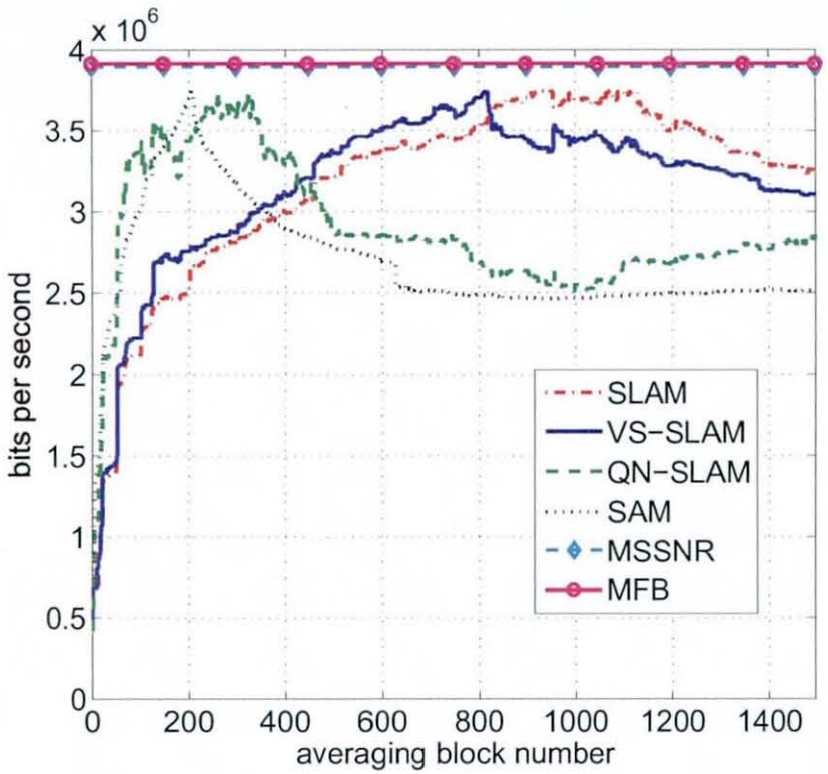

Figure 3.2. Achievable bit rate comparison of VS-SLAM and QN-SLAM with SLAM, SAM, MSSNR and MFB algorithms for CSA Loop 1.

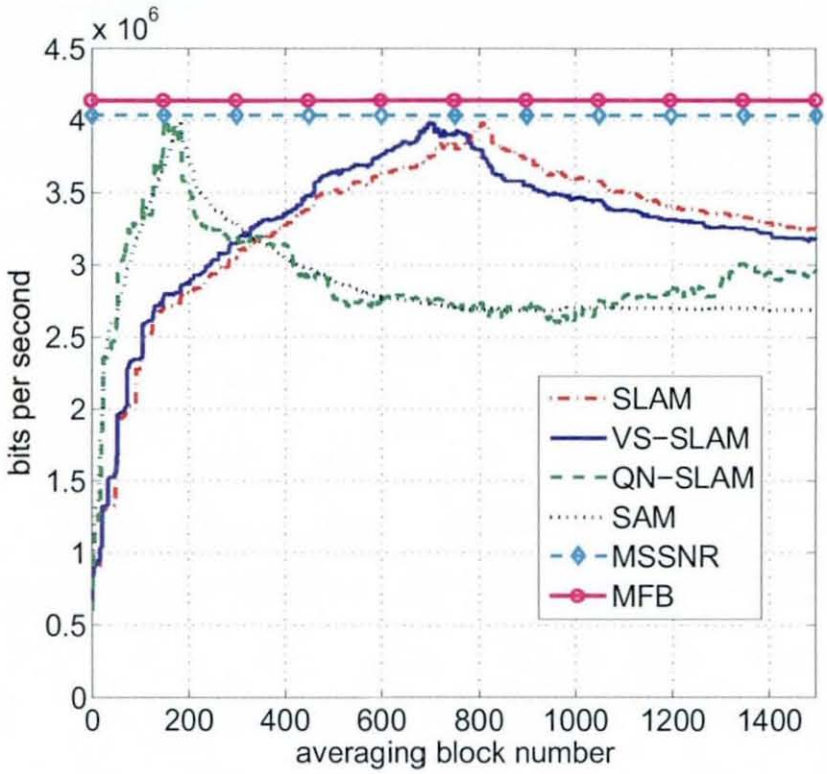

Figure 3.3. Achievable bit rate comparison of VS-SLAM and QN-SLAM with SLAM, SAM, MSSNR and MFB algorithms for CSA Loop 2.

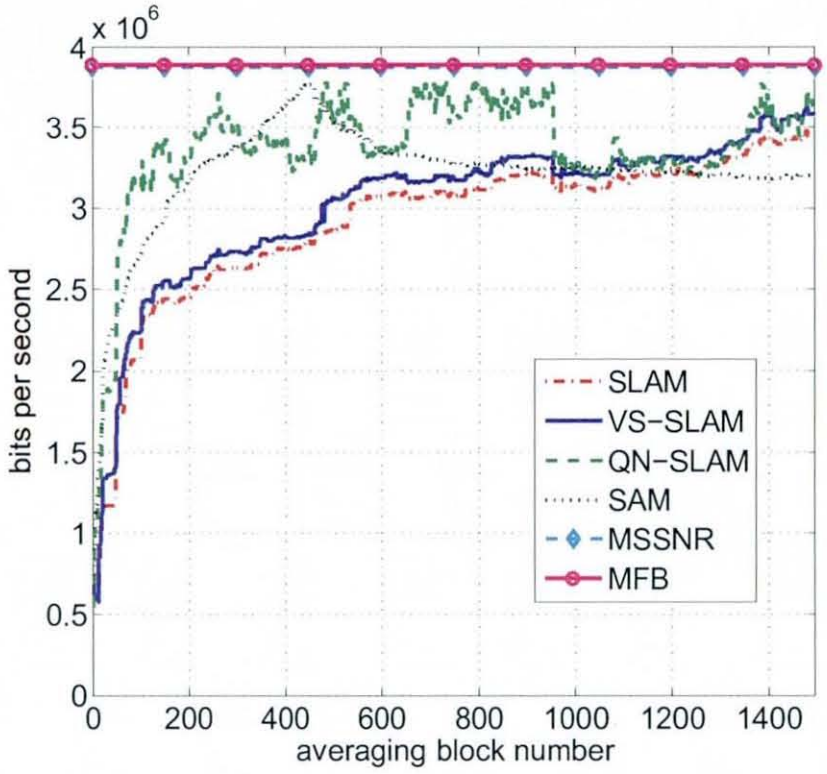

Figure 3.4. Achievable bit rate comparison of VS-SLAM and QN-SLAM with SLAM, SAM, MSSNR and MFB algorithms for CSA Loop 3.

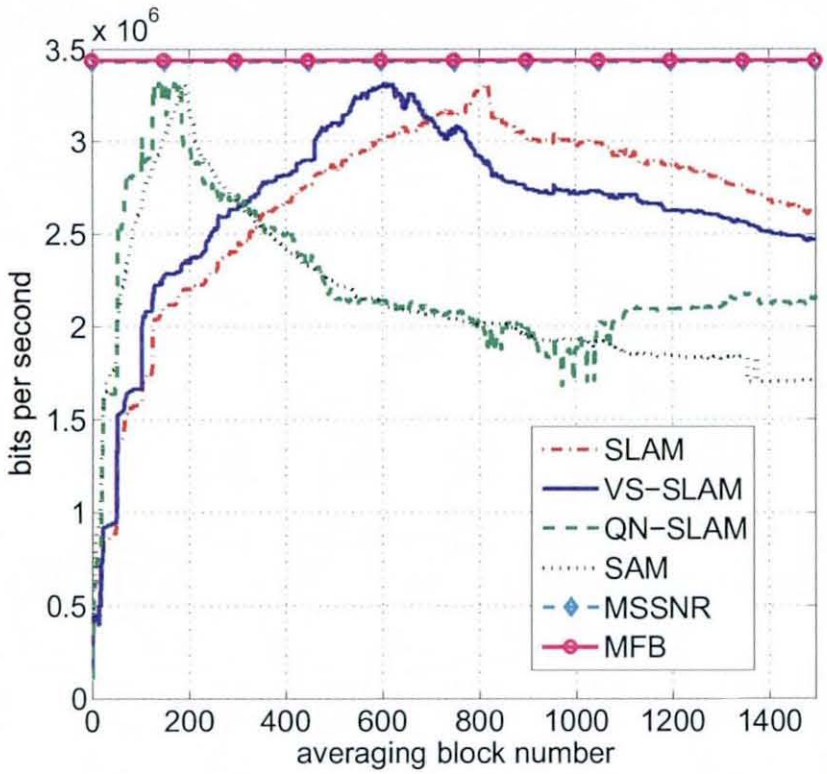

Figure 3.5. Achievable bit rate comparison of VS-SLAM and QN-SLAM with SLAM, SAM, MSSNR and MFB algorithms for CSA Loop 4.

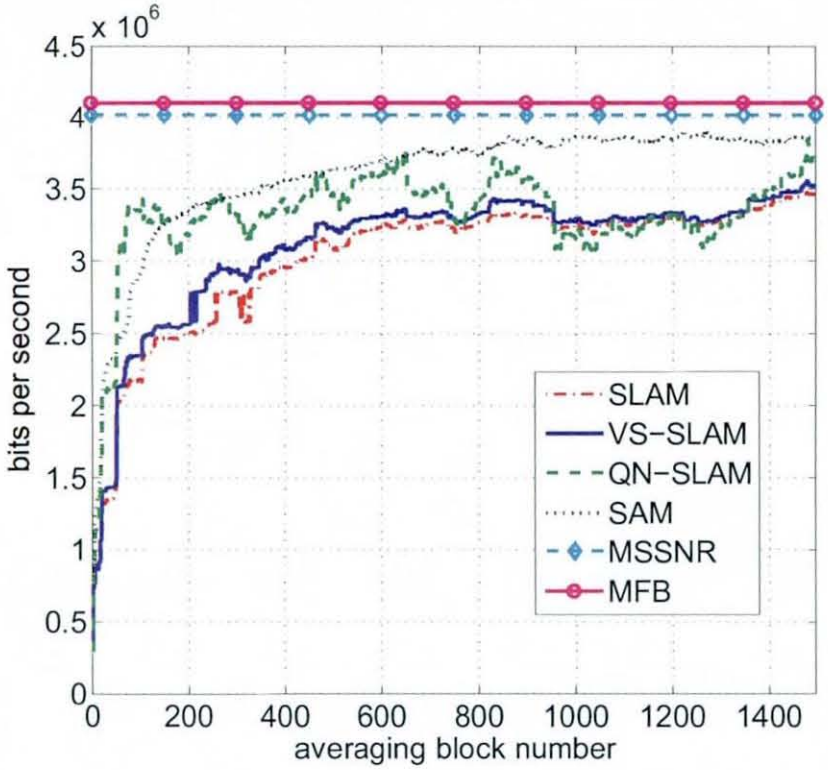

Figure 3.6. Achievable bit rate comparison of VS-SLAM and QN-SLAM with SLAM, SAM, MSSNR and MFB algorithms for CSA Loop 5.

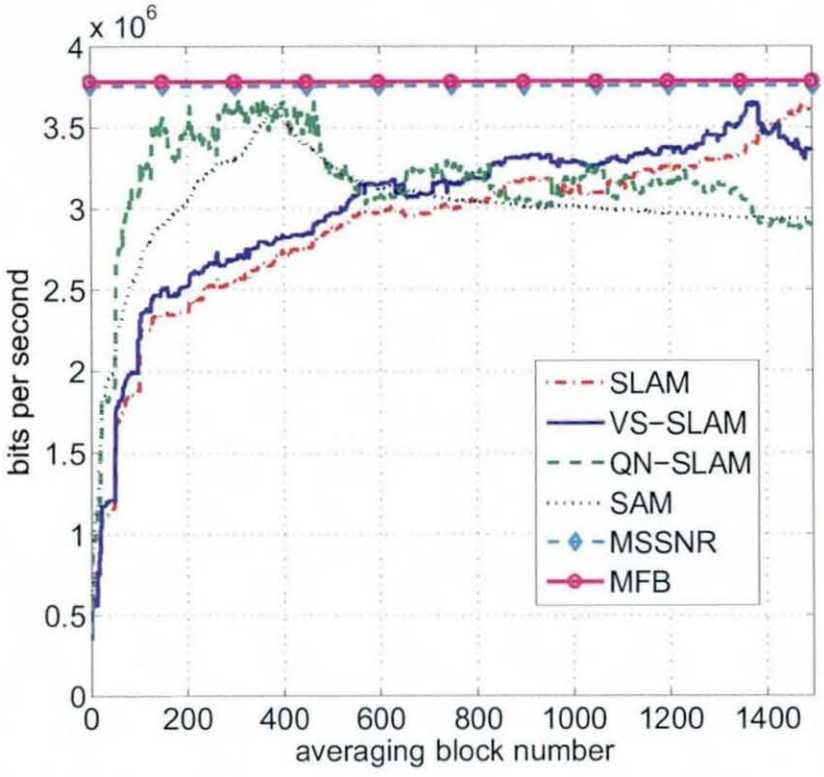

Figure 3.7. Achievable bit rate comparison of VS-SLAM and QN-SLAM with SLAM, SAM, MSSNR and MFB algorithms for CSA Loop 6.

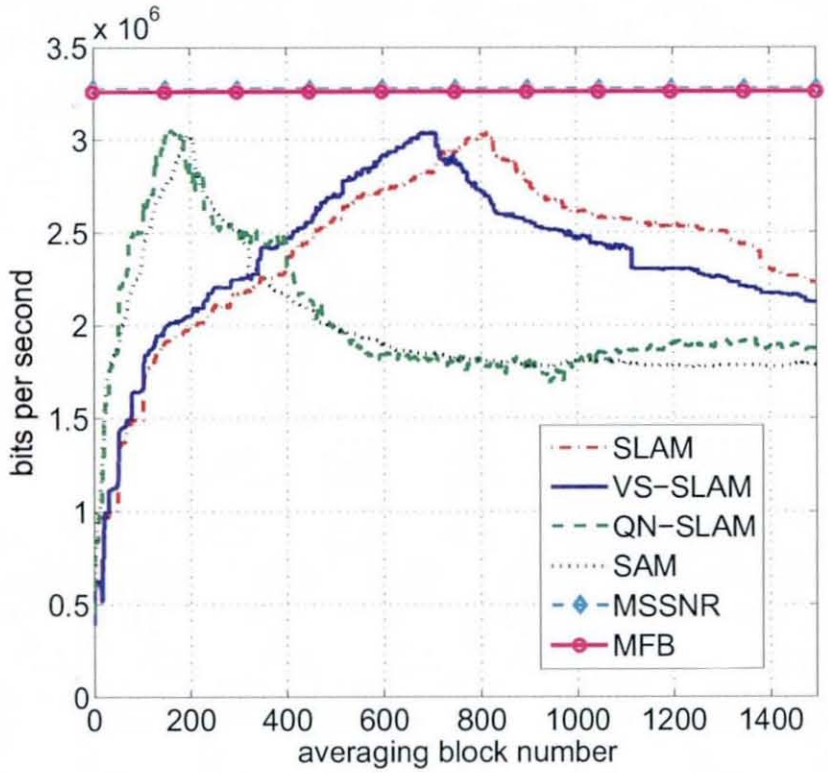

Figure 3.8. Achievable bit rate comparison of VS-SLAM and QN-SLAM with SLAM, SAM, MSSNR and MFB algorithms for CSA Loop 7.

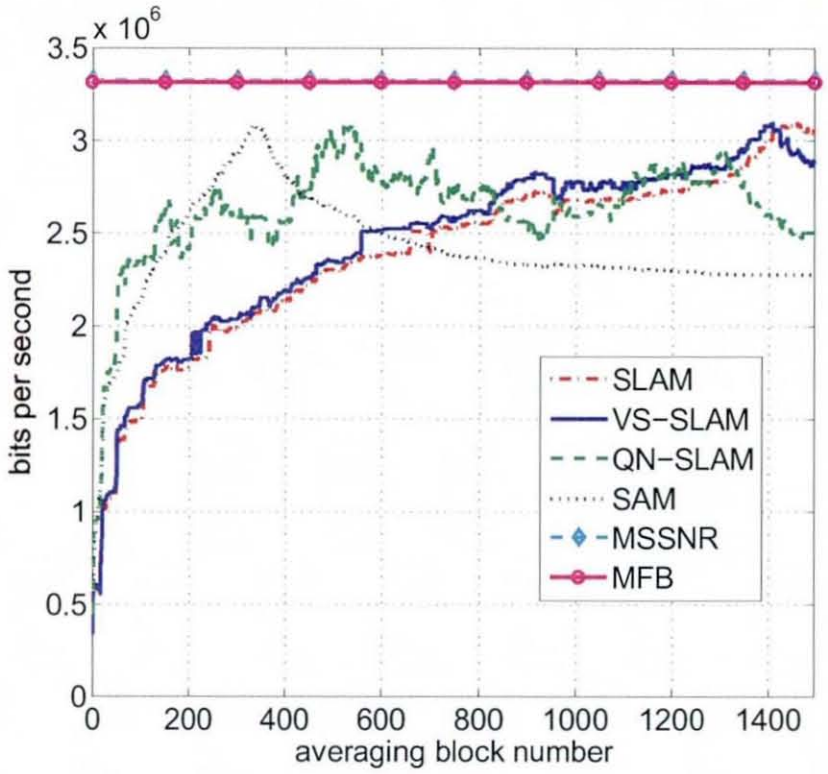

Figure 3.9. Achievable bit rate comparison of VS-SLAM and QN-SLAM with SLAM, SAM, MSSNR and MFB algorithms for CSA Loop 8.

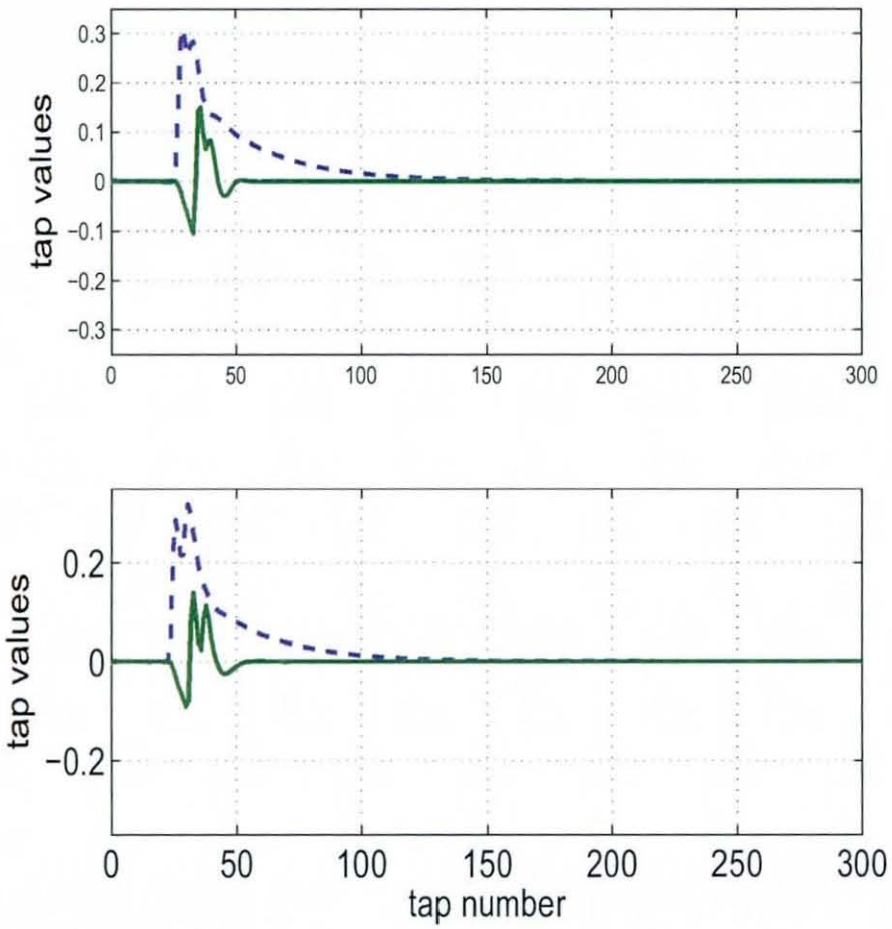

Figure 3.10. Channel shortening of CSA Loop 1 (top) and CSA Loop 2 (bottom) by VS-SLAM. Dotted and solid curves show original and the shortened channel, respectively.

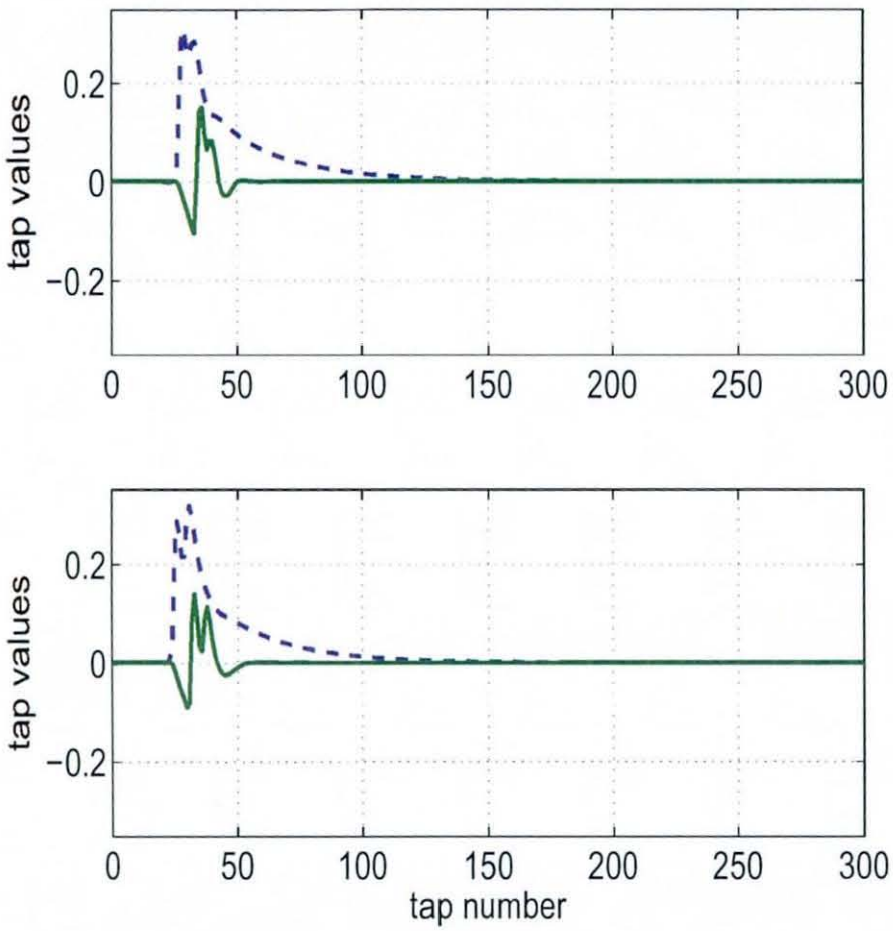

Figure 3.11. Channel shortening of CSA Loop 3 (top) and CSA Loop 4 (bottom) by VS-SLAM. Dotted and solid curves show original and the shortened channel, respectively.

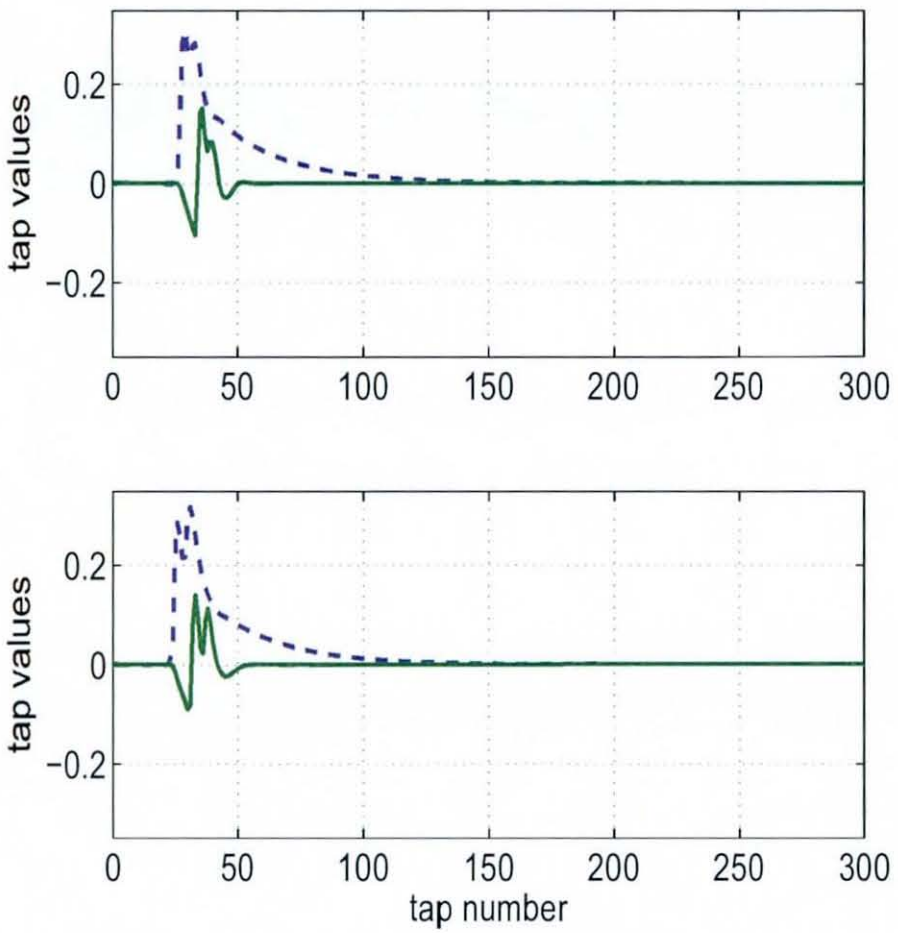

Figure 3.12. Channel shortening of CSA Loop 5 (top) and CSA Loop 6 (bottom) by VS-SLAM. Dotted and solid curves show original and the shortened channel, respectively.

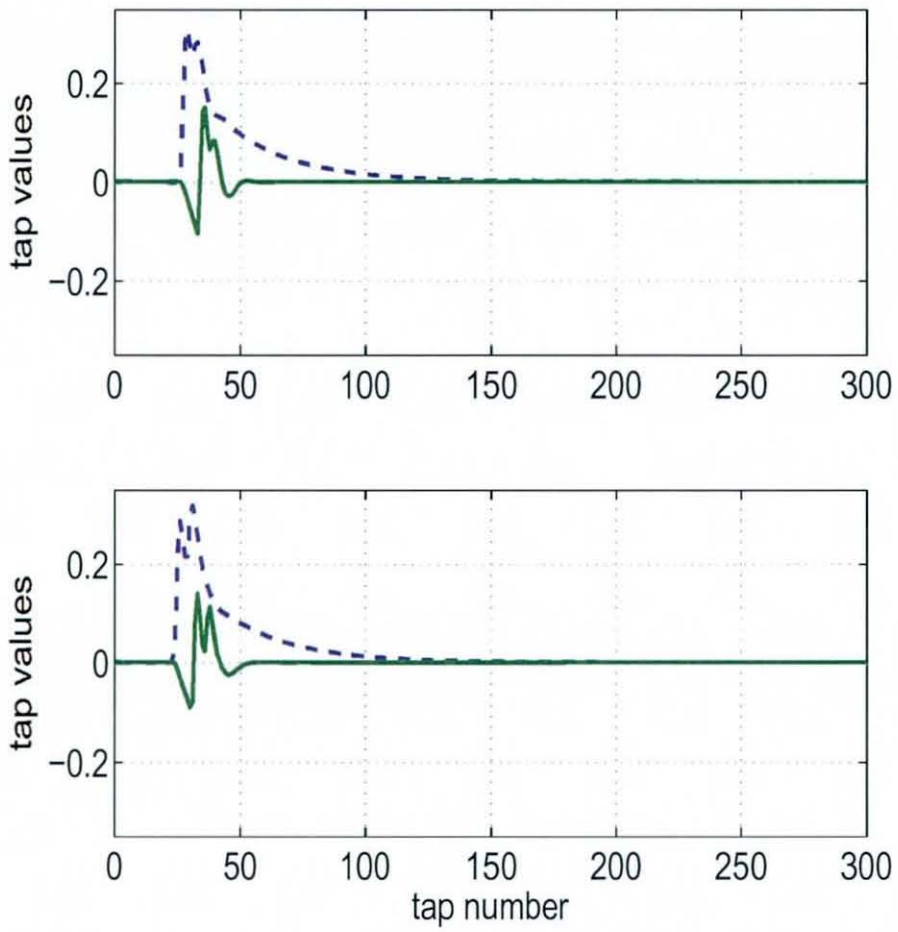

Figure 3.13. Channel shortening of CSA Loop 7 (top) and CSA Loop 8 (bottom) by VS-SLAM. Dotted and solid curves show original and the shortened channel, respectively.

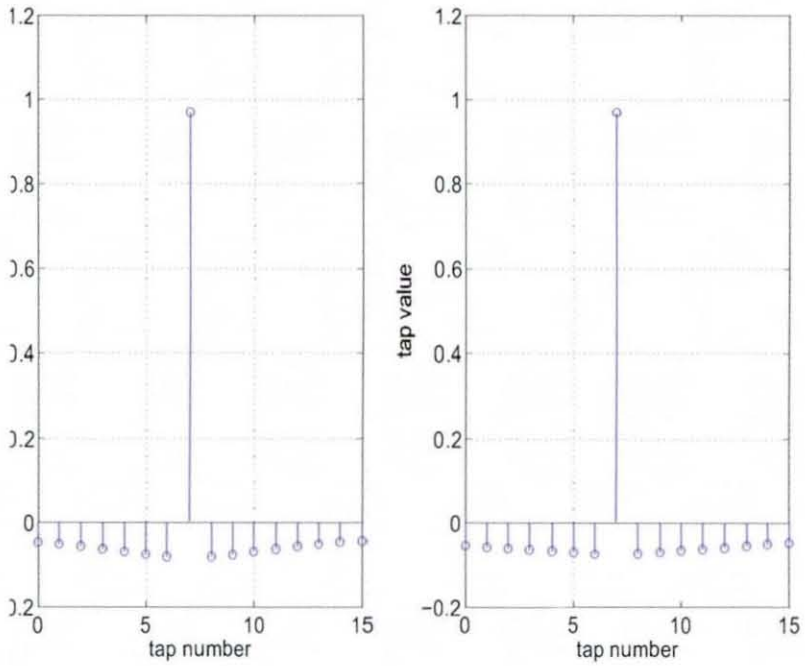

Figure 3.14. Steady state coefficients of the TEQ achieved by the VS-SLAM for CSA Loop 1 (left) and CSA Loop 2 (right).

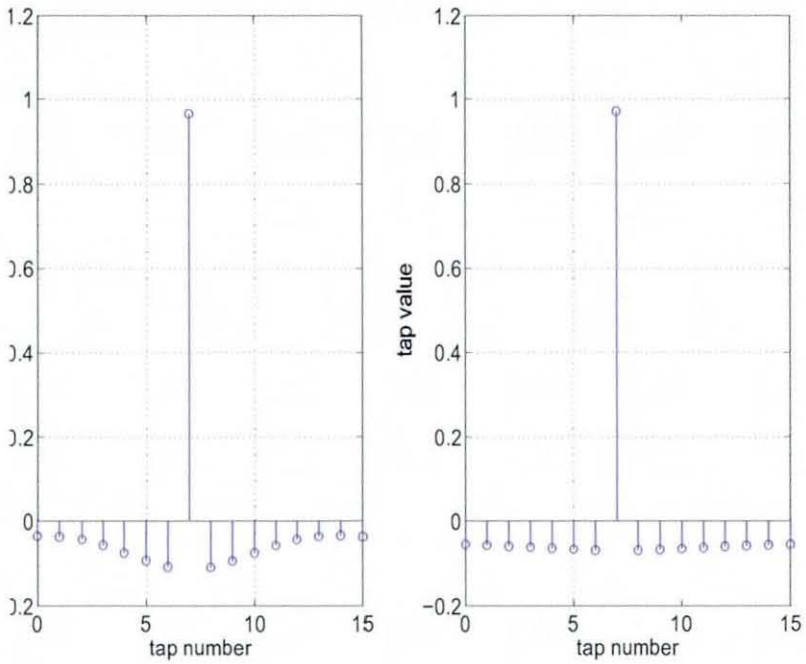

Figure 3.15. Steady state coefficients of the TEQ achieved by the VS-SLAM for CSA Loop 3 (left) and CSA Loop 4 (right).

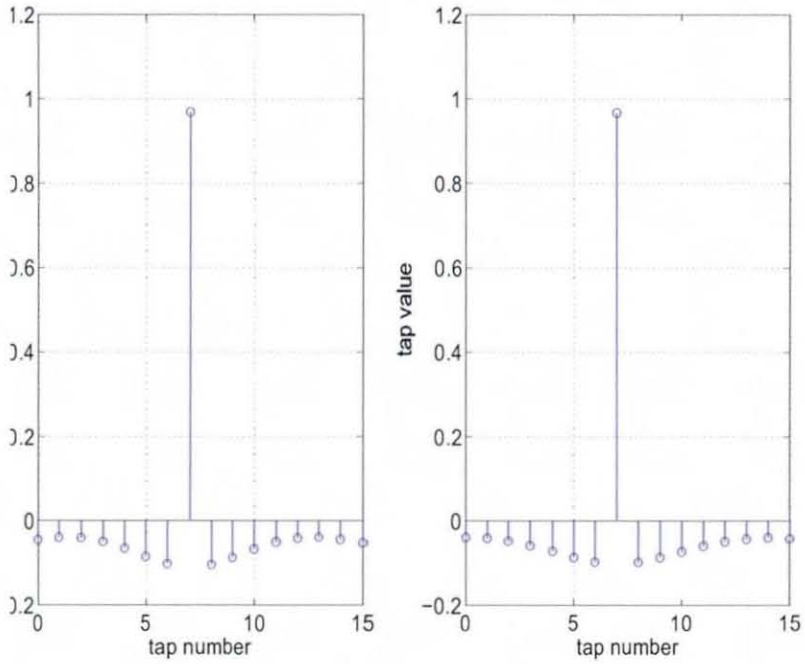

Figure 3.16. Steady state coefficients of the TEQ achieved by the VS-SLAM for CSA Loop 5 (left) and CSA Loop 6 (right).

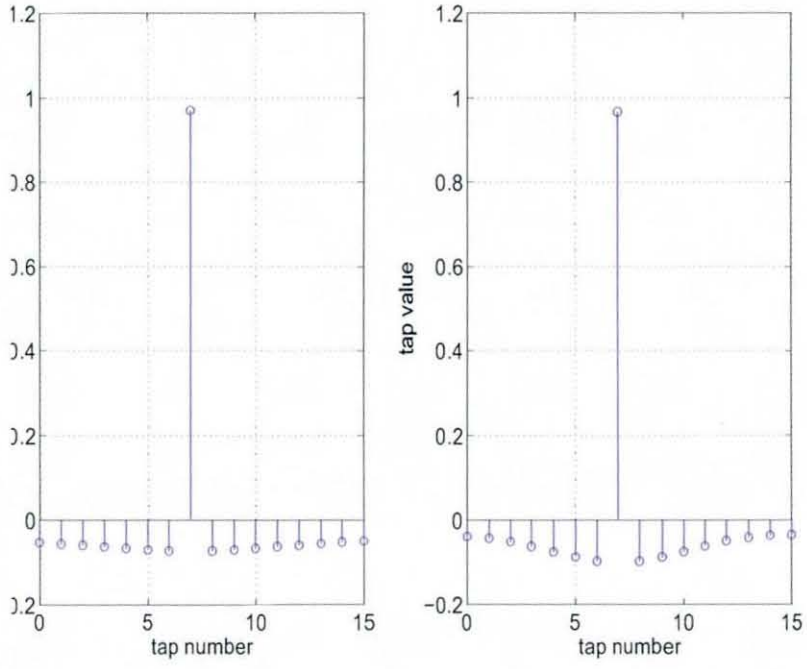

Figure 3.17. Steady state coefficients of the TEQ achieved by the VS-SLAM for CSA Loop 7 (left) and CSA Loop 8 (right).

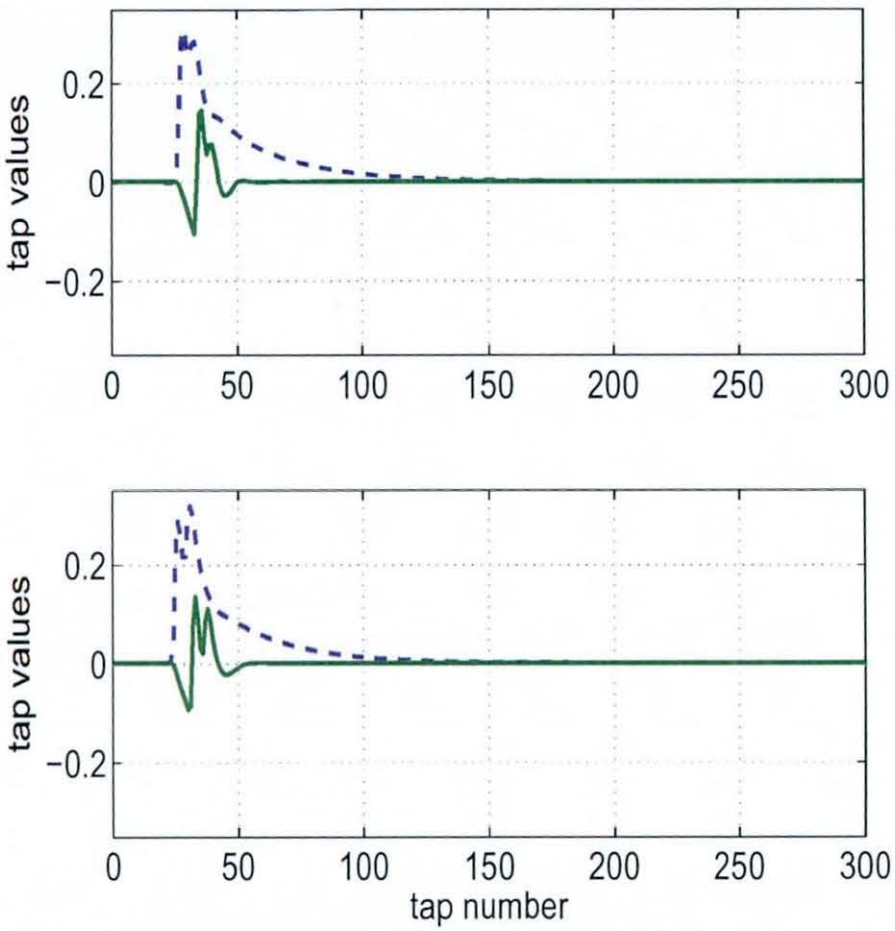

Figure 3.18. Channel shortening of CSA Loop 1 (top) and CSA Loop 2 (bottom) by QN-SLAM. Dotted and solid curves show original and the shortened channel, respectively.

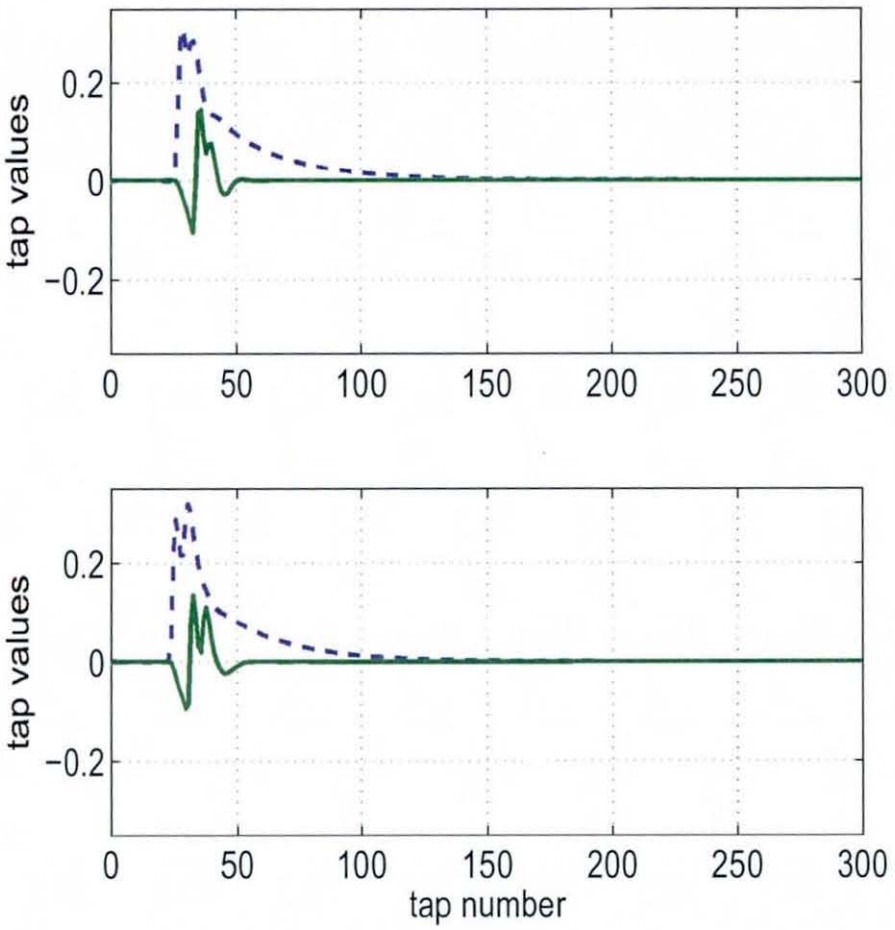

Figure 3.19. Channel shortening of CSA Loop 3 (top) and CSA Loop 4 (bottom) by QN-SLAM. Dotted and solid curves show original and the shortened channel, respectively.

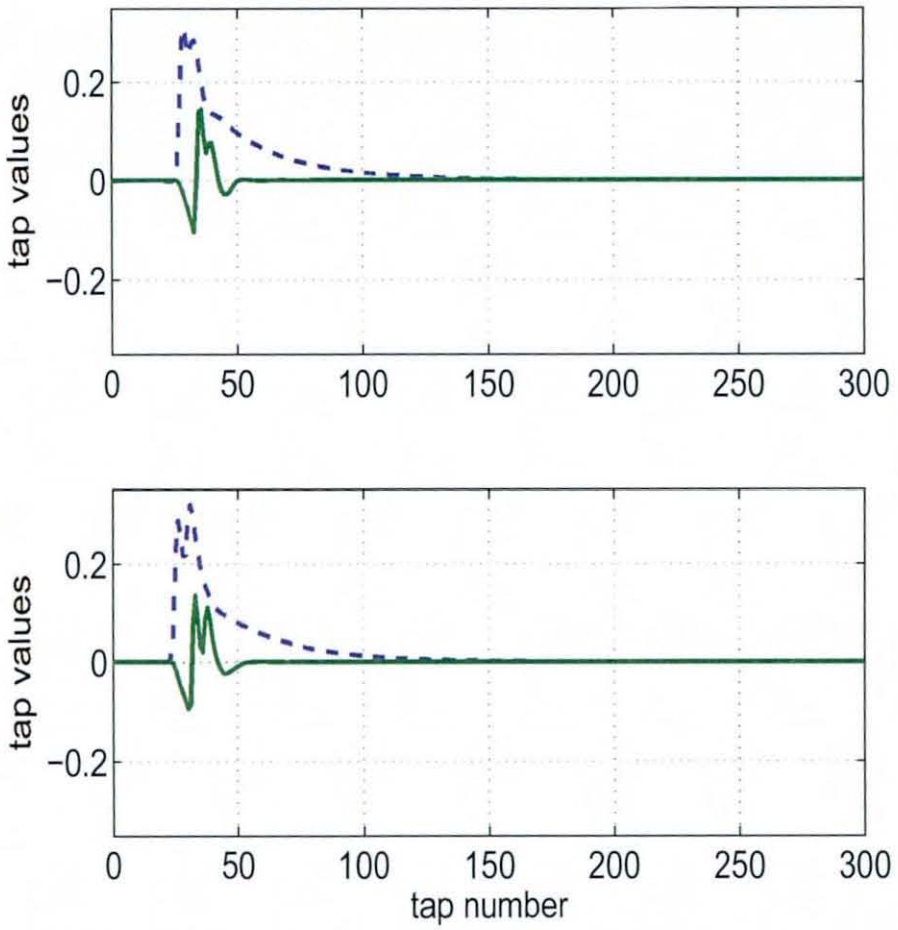

Figure 3.20. Channel shortening of CSA Loop 5 (top) and CSA Loop 6 (bottom) by QN-SLAM. Dotted and solid curves show original and the shortened channel, respectively.

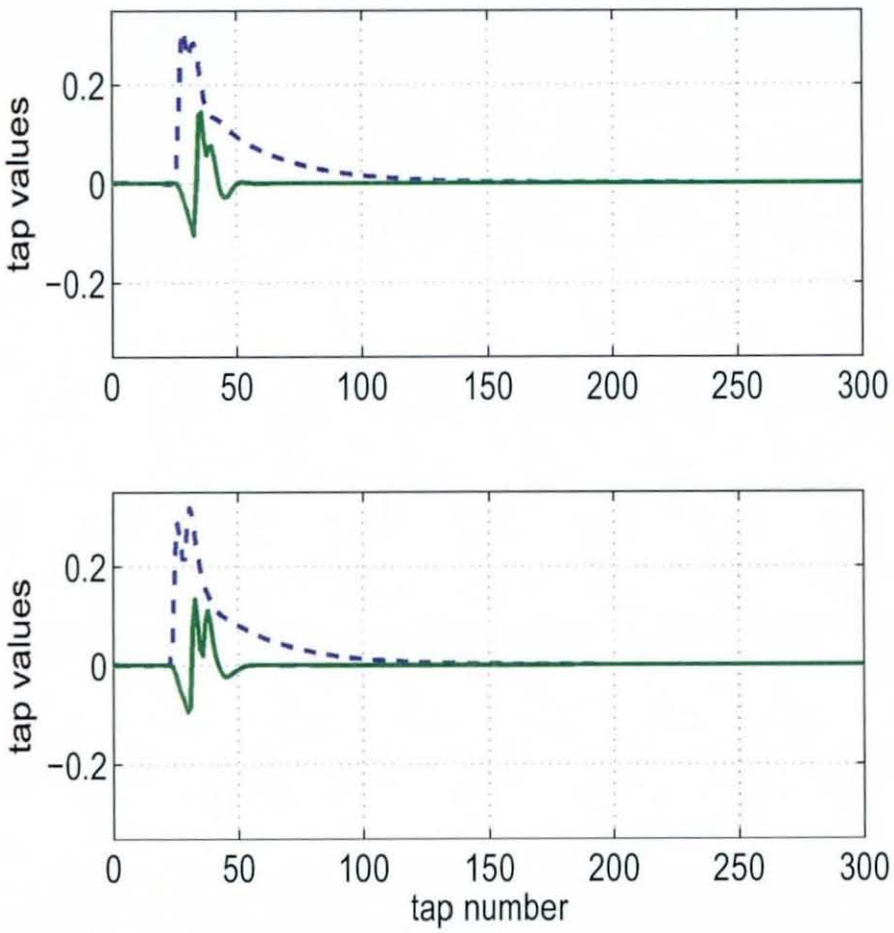

Figure 3.21. Channel shortening of CSA Loop 7 (top) and CSA Loop 8 (bottom) by QN-SLAM. Dotted and solid curves show original and the shortened channel, respectively.

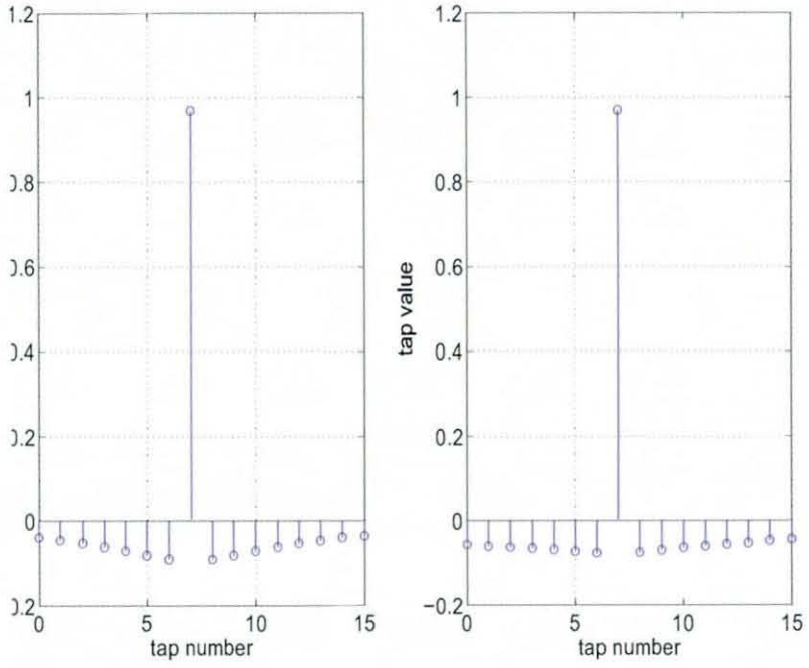

Figure 3.22. Steady state coefficients of the TEQ achieved by the QN-SLAM for CSA Loop 1 (left) and CSA Loop 2 (right).

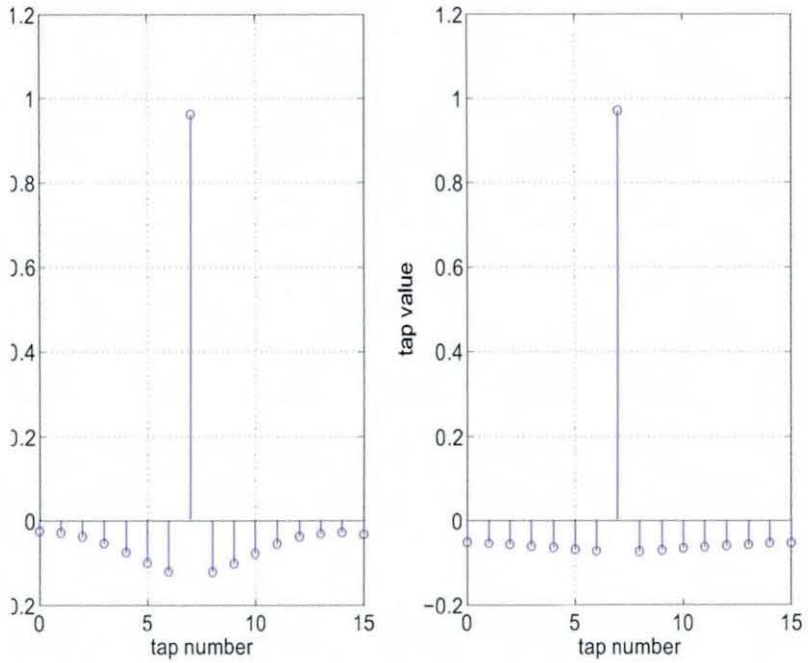

Figure 3.23. Steady state coefficients of the TEQ achieved by the QN-SLAM for CSA Loop 3 (left) and CSA Loop 4 (right).

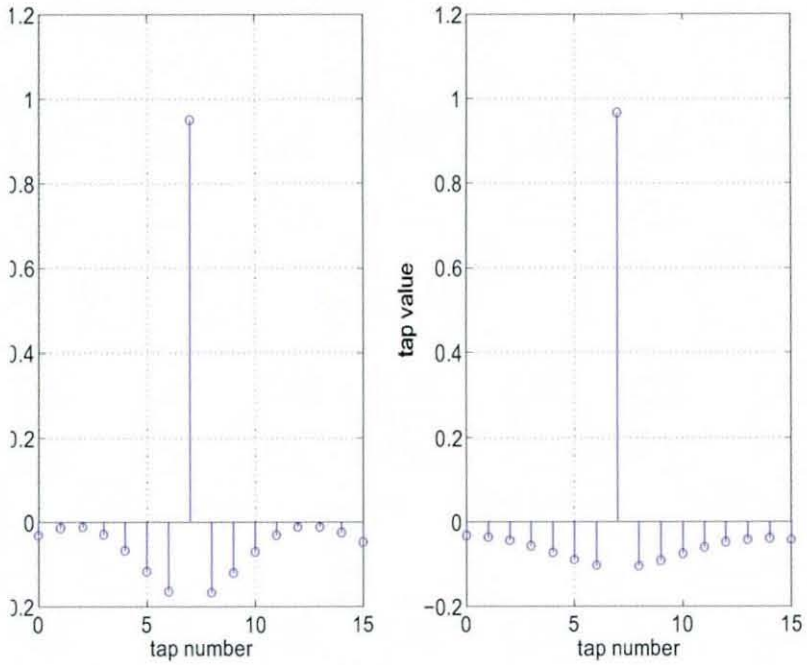

Figure 3.24. Steady state coefficients of the TEQ achieved by the QN-SLAM for CSA Loop 5 (left) and CSA Loop 6 (right).

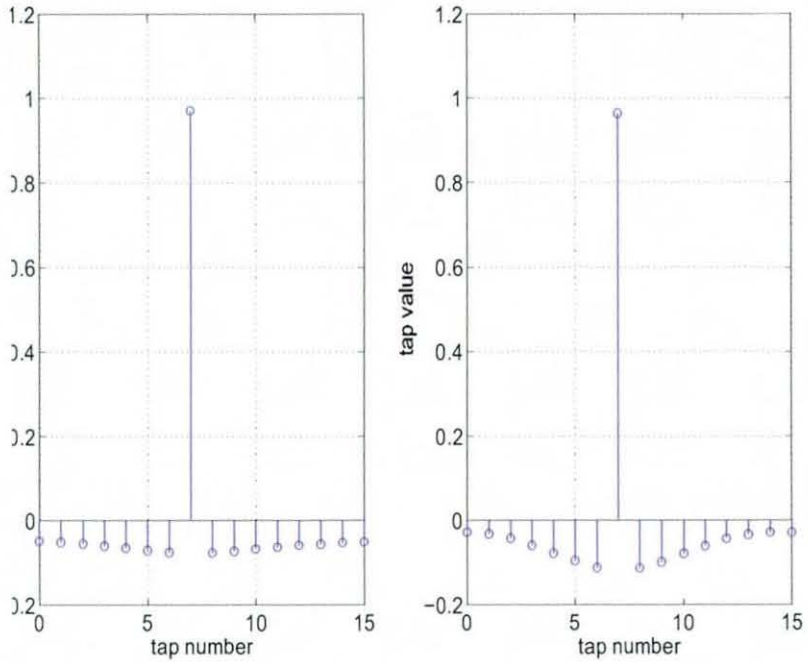

Figure 3.25. Steady state coefficients of the TEQ achieved by the QN-SLAM for CSA Loop 7 (left) and CSA Loop 8 (right).

Chapter 4

EXPONENTIAL PROBABILITY GENERALIZED LAG HOPPING SAM ALGORITHM (EGLHSAM)

4.1 Overview

An exponential probability generalized lag-hopping SAM algorithm (EGLHSAM) for channel shortening is proposed [64]. The algorithm minimizes the square of autocorrelation at one lag as for the SLAM algorithm. It differs though from SLAM algorithm in the way it selects the lag to be minimized The SLAM algorithm minimizes a fix lag whose value is greater than the cyclic prefix length. On the other hand, with EGLHSAM algorithm, the probability of selecting a lag matches approximately the envelope profile of the impulse response of the underlying channel to be shortened At each iteration a unique lag is chosen randomly from the available range so that on the average the

histogram of the lags chosen matches the impulse response of the channel. The motivation is to match the probability of selecting a lag to the nature of the underlying channel impulse response. The CSA loop channels have exponentially decaying impulse response characteristics. Therefore an exponentially decaying probability distribution is used for the selection of the lag to use within the cost function to be minimized. The simulation results show that the EGLHSAM algorithm improves the convergence of the SLAM algorithm. This algorithm provides the ability to select a level of complexity between the sum-squared autocorrelation minimization (SAM) algorithm due to Martin and Johnson and the single lag autocorrelation minimization (SLAM) algorithm, proposed by Nawaz and Chambers whilst guaranteeing convergence to high signal-to-interference-ratio (SIR)

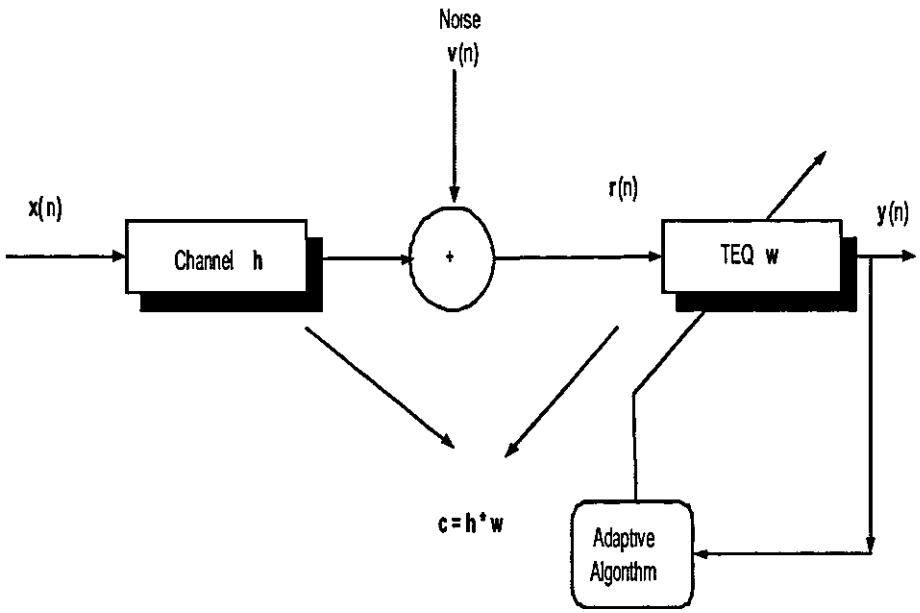

Figure 4.1. Overall baseband channel shortening system model

4.2 System Model

In order to make this chapter self contained, details of the system model are again included. The system model is shown in Figure 4.1. The signal x(n) is a white, zero mean, wide-sense stationary (W S S.), real and unit variance source sequence, typically drawn from a finite constellation, which is then transmitted through the linear finite-impulse response (FIR) channel $\mathbf{h} = [h(0)h(1)...h(L_h)]^T$, v(n) is a zero mean, i.i.d , noise sequence uncorrelated with the source sequence and has variance σ_v^2. The received signal $r(n)$ is

$$r(n) = \sum_{k=0}^{L_h} h(k)x(n-k) + v(n) \tag{4.2.1}$$

and $y(n)$, the output of the TEQ is given by

$$y(n) = \sum_{k=0}^{L_w} w(k)r(n-k) = \mathbf{w}^T\mathbf{r}_n \tag{4.2.2}$$

where $\mathbf{w}$ is the impulse response vector of the TEQ $\mathbf{w} = [w(0)w(1)...w(L_w)]^T$, and $\mathbf{r}_n = [r(n)r(n-1)...r(n-L_w]^T$. L_h, L_c, and L_w are the order of the channel, effective channel and the TEQ respectively It is also assumed that $2L_c \leq N$ holds, N being the FFT size [59] which is a reasonable assumption in the case of ADSL.

4.3 SAM and SLAM Cost Functions

The concept of SAM is based on the fact that for the effective channel to have zero taps outside a window of size $(v+1)$ its autocorrelation values must be zero outside a window of size $(2v+1)$. In SAM the auto-correlation sequence of the combined channel-equalizer impulse

response becomes

$$R_{cc}(l) = \sum_{k=0}^{Lc} c(k)c(k-l) \qquad (4\,3.1)$$

and for the shortened channel, which implies that the following must hold

$$R_{cc}(l) = 0, \quad \forall |l| > v \qquad (4\,3\,2)$$

The cost function J_{SAM} in SAM is defined on the basis of minimizing the sum-squared auto-correlation terms, i.e.,

$$J_{SAM} = \sum_{l=v+1}^{L_c} R_{cc}(l)^2 \qquad (4\,3.3)$$

SLAM exploits the fact that a single autocorrelation at a lag greater than the guard interval provides a measure of the presence of the channel outside the desired guard interval, hence minimizing only this single autocorrelation is particularly applicable to subscriber line channels which are essentially minimum phase. In SLAM the auto-correlation sequence of the combined channel-equalizer impulse response is also given by equation (5 4.1) which can be found in Chapter 5 and for a shortened channel, it follows that

$$R_{cc}(l) = 0, \quad l = v+1 \qquad (4\,3\,4)$$

In this case the cost function J_{SLAM} in SLAM is defined based upon minimizing the squared-auto-correlation of the effective channel only at lag $l = v+1$, i.e.,

$$J_{SLAM} = R_{cc}(l)^2, \quad l = v+1 \qquad (4\,3.5)$$

4.4 SIR Performance

In [65], the authors provide an expression for the signal to interference power ratio (SIR) achieved in the output $y(n)$ when the TEQ is based on the blind channel shortening metrics of SAM, SAAM, and SLAM. For non-negative lags it can written as

$$SIR = \frac{\sum_{l=-v}^{v} |R_{cc}(l)|^2}{\sum_{l=-N}^{-v+1} |R_{cc}(l)|^2 + \sum_{l=v+1}^{N} |R_{cc}(l)|^2} \tag{4.4.1}$$

It should be noted that the denominator in this expression is the SAM cost Now considering those shortened responses only which satisfy the unit energy constraint, the following relation can be derived [65]

$$\begin{aligned} SIR(dB) &= 10\log\Big(\sum_{l=-v}^{v} |R_{cc}(l)|^2\Big) - 10\log(J_s) \\ &= 10\log\Big(1 + 2\sum_{l=1}^{v} |R_{cc}(l)|^2\Big) - 10\log(J_s) \end{aligned} \tag{4 4.2}$$

where J_s denotes the SAM cost, J_{SLAM} denotes the SLAM cost, and $J_s^{'}$ denotes the SAM cost minus SLAM cost. From the second line in equation (4 4.2), it is seen that a low SAM cost can be guaranteed to a give high SIR at the output of the TEQ. Unfortunately, as stated in [65], no such result holds for SLAM. This drawback is not present in the SAM algorithm In order to overcome this problem with SLAM, the EGLHSAM algorithm is proposed

It selects the lags randomly from the whole range of lags of SAM algorithm, so that a low average EGLHSAM cost, achieved through recursive learning, provided $k \longrightarrow \infty$, guarantees to give a high SIR for

all types of channels at the output of the TEQ. A second benefit of the EGLHSAM algorithm is that it increases the convergence rate as compared to the SLAM algorithm as it selects the lags with a probability to match the impulse response characteristics of the underlying channel.

4.5 EGLHSAM Blind Adaptive Algorithm

The steepest gradient-descent algorithm to minimize the SAM cost J_{SAM} becomes

$$\mathbf{w}^{new} = \mathbf{w}^{old} - \mu \nabla_{\mathbf{w}} \sum_{l=v+1}^{L_c} (E[y(n)y(n-l)])^2 \qquad (4\,5.1)$$

where l is the lag index, μ denotes the step size, and $\nabla_{\mathbf{w}}$ represents the gradient with respect to $\mathbf{w}$. The instantaneous cost at time instant k, where the expectation operation is replaced by a moving average over a user-specified window of length N_{avg} is defined as

$$\sum_{l=v+1}^{L_c} (E[y(n)y(n-l)])^2 = \sum_{l=v+1}^{L_c} \left\{ \sum_{n=kN_{avg}}^{(k+1)N_{avg}-1} \frac{y(n)y(n-l)}{N_{avg}} \right\}^2 \qquad (4\,5\,2)$$

where N_{avg} is a design parameter and it should be large enough to yield a reliable estimate of the expectation, but no larger, as the algorithm complexity is proportional to N_{avg}. The gradient descent algorithm becomes

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mu \nabla_{\mathbf{w}} \Big(\sum_{l \;\varepsilon\; Lagset} \left\{ \sum_{n=kN_{avg}}^{(k+1)N_{avg}-1} \frac{y(n)y(n-l)}{N_{avg}} \right\}^2 \Big)$$

$$\mathbf{w}(k+1) = \mathbf{w}(k) - 2\mu \sum_{l\ \varepsilon\ Lagset} \Big\{ \sum_{n=kN_{avg}}^{(k+1)N_{avg}-1} \frac{y(n)y(n-l)}{N_{avg}} \Big\} \times \Big\{ \nabla_{\mathbf{w}} \Big(\sum_{n=kN_{avg}}^{(k+1)N_{avg}-1} \frac{y(n)y(n-l)}{N_{avg}} \Big) \Big\} \quad (4.5.3)$$

and using equation (5.3.2), the lag hopping algorithm becomes.

$$\mathbf{w}(k+1) = \mathbf{w}(k) - 2\mu \sum_{l\ \varepsilon\ Lagset} \Big\{ \sum_{n=kN_{avg}}^{(k+1)N_{avg}-1} \frac{y(n)y(n-l)}{N_{avg}} \Big\} \times \Big\{ \Big(\sum_{n=kN_{avg}}^{(k+1)N_{avg}-1} \frac{y(n)\mathbf{r}_{n-l} + y(n-l)\mathbf{r}(n)}{N_{avg}} \Big) \Big\} \quad (4\,5\,4)$$

where $l_1 ... l_{N_{LAGS}}$ within the Lagset are chosen to be individually unique and to be drawn with exponentially decaying probability from the range of available lags, initially $v+1, \ ., L_c$ The number of lags, $L_{N_{LAGS}}$, can be chosen over the range $1, .. , L_c - v$, and when $N_{LAGS} = 1$, with exponential probability, the algorithm takes the form of a lag-hopping version of SLAM, named EGLHSAM

4.6 Probability of lags selection

The range of autocorrelation lags to be included in the SAM family of cost functions is $v + 1, ..., L_c$. The SAM algorithm suggests minimizing the sum-squared autocorrelation at all of these lags. On the other hand, the SLAM algorithm takes into account only the lag at $v + 1$. EGLHSAM algorithm suggests to select from one to $L_c - (v + 1) + 1$ lags from the range, randomly but uniquely and with exponentially decaying probability of selecting the lags from the range. This has been demonstrated in Figure (4 4). The x-axis shows the available range of lags and y-axis shows their exponentially decaying probability

The goal of the EGLHSAM algorithm is to get an exponential probability of selecting the lags from the available range This purpose is achieved by the following steps on the random number *tvar* generated between 0 and 1 with uniform probability The equations are given in Matlab notation

$$tvar2 = abs(exp(a * tvar) - 1) \tag{4.6 1}$$

$$tvar2 = tvar2/abs(exp(a * 1) - 1) \tag{4 6.2}$$

The purpose of the above two lines is to shape the variable *tvar* to lie exponentially between 0 and 1 according to the parameter a. A positive value of a gives exponentially decaying behavior of *tvar* while a negative a gives exponentially increasing behavior of *tvar* The second line in the equation is for normalizing *tvar*2 between 0 and 1. The transformation from *tvar* to *tvar*2 is shown in the Figure (4.2) As before, the lag l is decided using

$$l = (v + 1) + round(tvar2 * (L_c - (v + 1))) \tag{4 6 3}$$

Figure (4 3) shows the exponentially decreasing as well as increasing probabilities of the lags during the simulations of the EGLHSAM algorithm. The values of a used are mentioned in the simulation. Figure (4.3) shows the histograms of the lags simulated for the EGLHSAM algorithm In this figure, the top left plot represents the SLAM algorithm. The other three plots represent EGLHSAM algorithm with different slopes of exponentially decaying probabilities The titles of these three plots show the parameter which controls their slope, with

smaller number suppressing more the selection of higher lags.

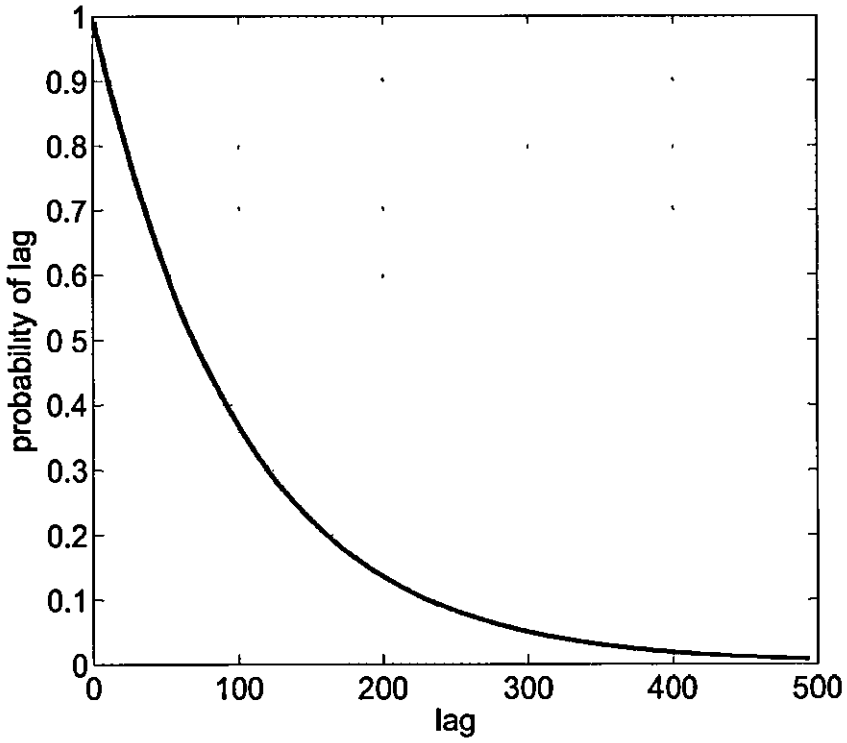

Figure 4.2. The lags and their exponentially decaying probability.

The complexity of the SLAM algorithm is about 1/500 times that of SAM for typical CSA loop channels [1]. EGLHSAM enjoys the same advantage with the SLAM compared to the SAM algorithm.

4.7 Simulations

The standard parameters of an ADSL downstream transmission were again simulated. An MA implementation was simulated for the SAM, SLAM and EGLHSAM algorithms The value of N_{avg} was 32. The cyclic prefix had length 32 The FFT size $N_{fft} = 512$, the TEQ had 16 taps and the channels used were the eight test ADSL channels CSA

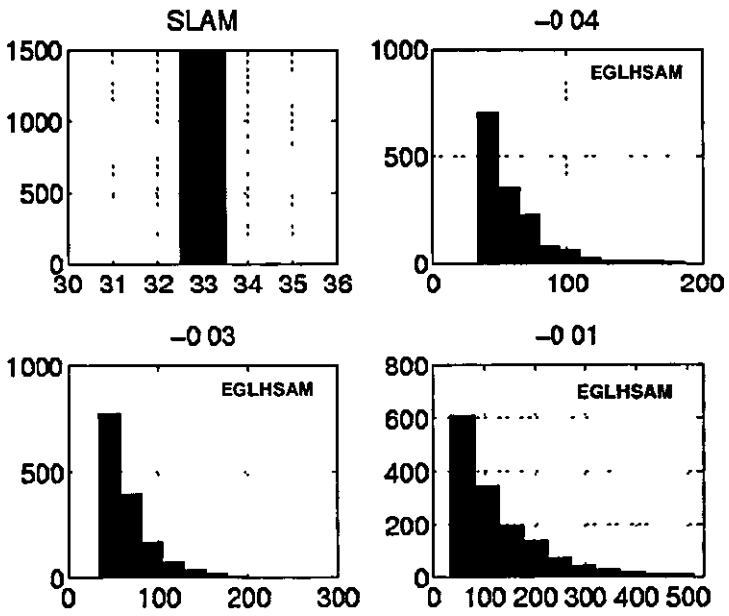

Figure 4.3. Histogram of the lags for SLAM and EGLHSAM algorithms. The values of the lags are between v+1=33 and L_c=526 The titles of EGLHSAM plots show the parameter which controls their slope, with smaller number suppressing more the selection of higher lags

loops provided at [62]. The noise was chosen such that $\sigma_x^2\|\mathbf{c}\|^2/\sigma_v^2 = 40$ dB where $\|.\|$ denotes the Euclidean norm. 89 OFDM symbols were employed. This value was chosen such that the whole range of available lags is exhausted at least 3 times by the EGLHSAM algorithm. The decaying parameter values of -0.04,-0.03 and -0.01 were employed to simulate different decaying slopes. The resulting histograms are shown in Figure (4 3). Single spike initialization with the center spike of the TEQ initialized to unity was used. The step size for SAM was 5; whereas, for SLAM and EGLHSAM, it was 600 to obtain the respective algorithms converge in the given number of symbols All algorithms were compared with the maximum shortening SNR (MSSNR) solution and the matched filter bound (MFB) on capacity, which assumes no ICI Figure (4.4).

Figure (4 5) shows the original and shortened CSA Loop 1 (top) and 2 (bottom) by the EGLHSAM algorithm with $\alpha = -0.04$. Similarly, Figure (4 6) shows the original and shortened CSA Loop 3 (top) and 4 (bottom) by the EGLHSAM algorithm with $\alpha = -0\,04$ (4 7) and (4 8) show the same for CSA Loop 5,6 and 7,8 respectively Figure (4.9) shows steady state coefficients of the TEQ given by the EGLHSAM algorithm for CSA Loop 1 (left) and 2 (right). Figures (4 10), (4.11), and (4.12) show the same for CSA Loop 3,4 and 5,6 and 7,8, respectively.

Figure (4.13) compares the achievable bit rates by SAM, SLAM, and EGLHSAM algorithms for CSA loop 1. Note that, the stopping criterion is not applied [66] where learning is stepped at peak bps. It is evident that EGLHSAM with $\alpha = -0.04$ clearly outperforms SLAM in terms of convergence rate, whereas EGLHSAM with $\alpha = -0\,03$ and

$\alpha = -0\,01$ match the convergence rate of the SLAM. Note that the curves with higher number represent lesser suppressing of higher lags but a decrease in the convergence rate is seen from blue to green and cyan curve.

Figure (4.14) compares the achievable bit rates by SAM, SLAM, and EGLHSAM algorithms for CSA loop 2 Again the increase in the convergence rate is observed for EGLHSAM algorithm with $\alpha = -0\,04$ and $\alpha = -0\,03$. But the cyan curve with $\alpha = -0\,01$ even degrades more than the SLAM algorithm.

Figure (4.15) compares the achievable bit rates by SAM, SLAM, and EGLHSAM algorithms for CSA loop 3. Again the increase in the convergence rate is observed for EGLHSAM algorithm with $\alpha = -0\,04$, $\alpha = -0\,03$ and $\alpha = -0.01$. But the green curve with $\alpha = -0\,03$ outperform the other two EGLHSAM curves in terms of convergence rate

Figure (4.16) compares the achievable bit rates by SAM, SLAM, and EGLHSAM algorithms for CSA loop 4. Again the increase in the convergence rate is observed for EGLHSAM algorithm with $\alpha = -0\,04$ and $\alpha = -0\,03$. But the cyan curve with $\alpha = -0\,01$ even degrades than the SLAM algorithm.

Figure (4.17) compares the achievable bit rates by SAM, SLAM, and EGLHSAM algorithms for CSA loop 5. All the EGLHSAM are better than SLAM in terms of convergence rate. The convergence rate of the cyan curve outperform other EGLHSAM curves.

Figure (4 18) compares the achievable bit rates by SAM, SLAM, and EGLHSAM algorithms for CSA loop 6. Again the increase in the convergence rate is observed for EGLHSAM algorithm with $\alpha = -0.04$ and $\alpha = -0\,03$. But the cyan curve with $\alpha = -0\,01$ even degrades

more than the SLAM algorithm.

Figure (4.19) compares the achievable bit rates by SAM, SLAM, and EGLHSAM algorithms for CSA loop 7 The convergence rate of the EGLHSAM algorithm with $\alpha = -0\ 04$ and $\alpha = -0.03$ are comparable with the SLAM algorithm, but the cyan curve with $\alpha = -0\ 01$ even degrades more than the SLAM algorithm

Figure (4.20) compares the achievable bit rates by SAM, SLAM, and EGLHSAM algorithms for CSA loop 8 Again the increase in the convergence rate is observed for EGLHSAM algorithm with $\alpha = -0.04$ and $\alpha = -0\ 03$. But the cyan curve with $\alpha = -0\ 01$ is comparable with that of the SLAM algorithm.

4.8 Summary

A new lag hopping blind adaptive channel shortening algorithm has been proposed The proposed EGLHSAM algorithm essentially achieves the same result in terms of reducing the effective channel length as SLAM and SAM The proposed algorithm shortens all types of channels, where SLAM does not have the capability to shorten all types of channels EGLHSAM is also intuitive to match the probability of lag selection to the impulse response of the underlying channel to improve the convergence rate. The algorithm has low complexity as for the SLAM algorithm. The simulations have revealed that the performance of the EGLHSAM algorithm for at least the ADSL downstream CSA loops 1 to 8 is better than that of the SLAM for all the channels.

The detailed discussion in the simulations section shows that the EGLHSAM converges faster than the SLAM algorithm For different CSA loops, this improvement in convergence is achieved at different val-

ues of α. In practice, however, the optimum α for the channel to be shortened cannot always be found. This fact makes the EGLHSAM algorithm specific to the channel being shortened As the value of α is decreased, more and more higher lags are suppressed and they are not minimized by the EGLHSAM algorithm. This situation resembles that of the SLAM algorithm. For certain channels, it is possible that EGLHSAM with lower value of α has zero cost, but the channel might not be shortened as not all the lags are minimized. Of course this problem is not that severe as with the SLAM algorithm In the next chapter, another lag hopping algorithm is being introduced where the exponentially decaying condition on the lag selection is being relaxed and all the lags are being randomly and uniquely chosen with uniform probability.

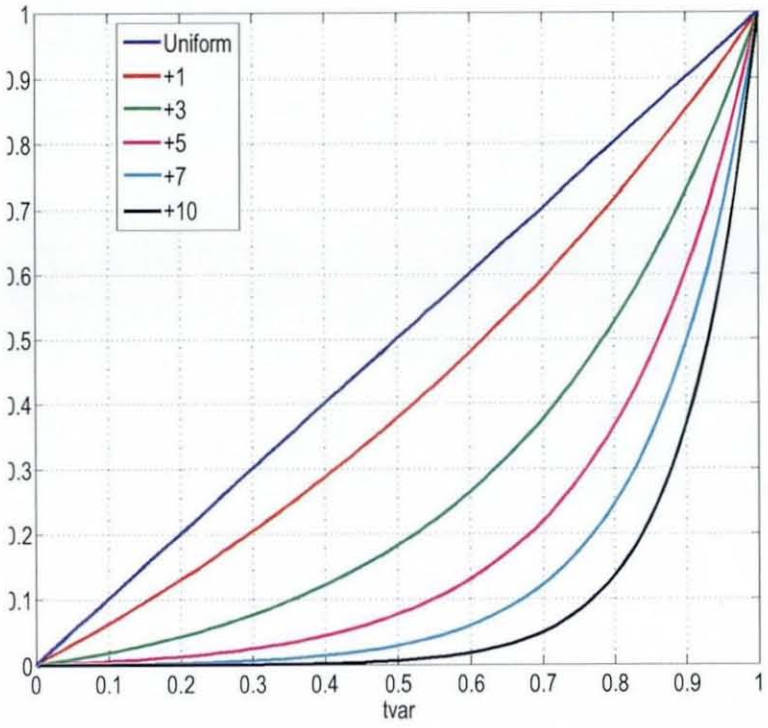

Figure 4.4. Transformation of *tvar* to *tvar*2 to get an exponential increasing probability from a uniform one.

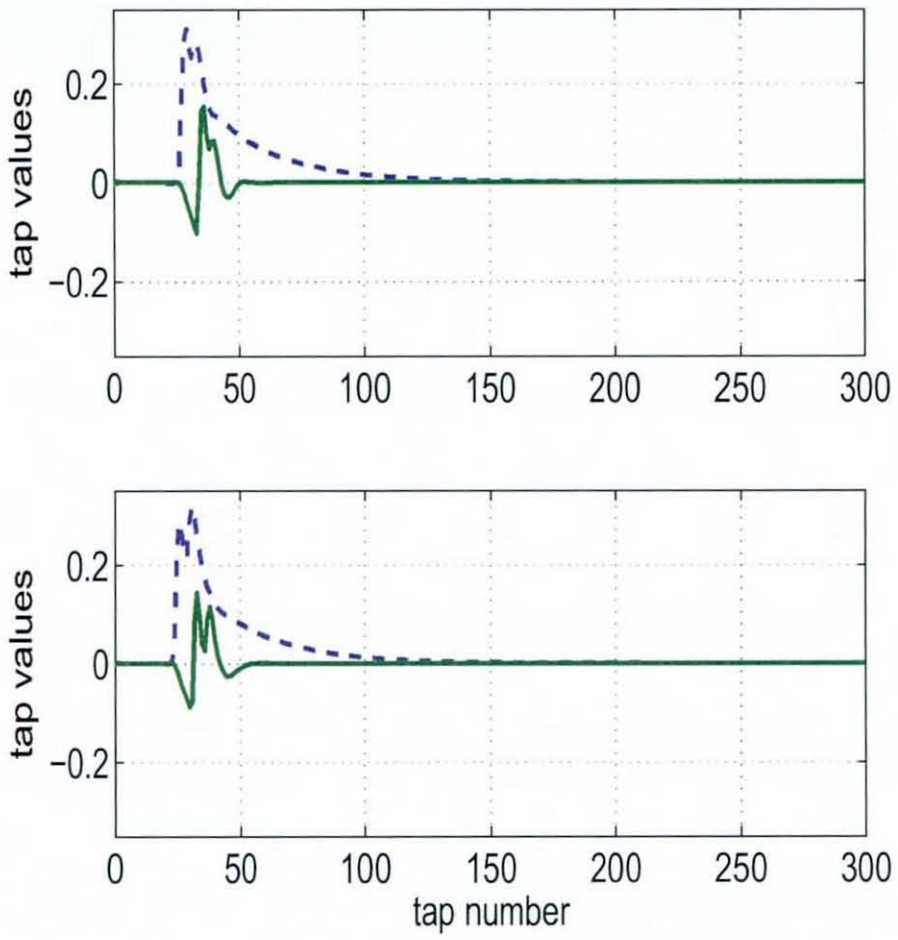

Figure 4.5. Channel shortening of CSA Loop 1 (top) and CSA Loop 2 (bottom) by EGLHSAM with $\alpha = -0.04$. Dotted and solid curves show original and the shortened channel, respectively.

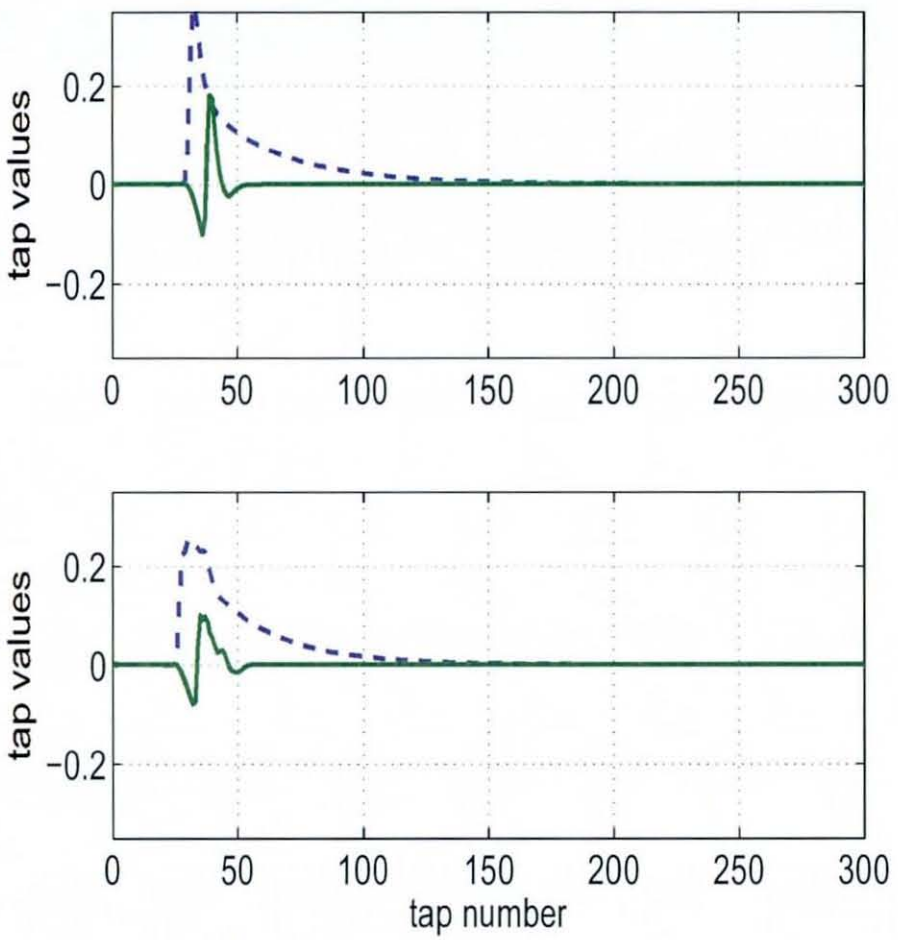

Figure 4.6. Channel shortening of CSA Loop 3 (top) and CSA Loop 4 (bottom) by EGLHSAM with $\alpha = -0.04$. Dotted and solid curves show original and the shortened channel, respectively.

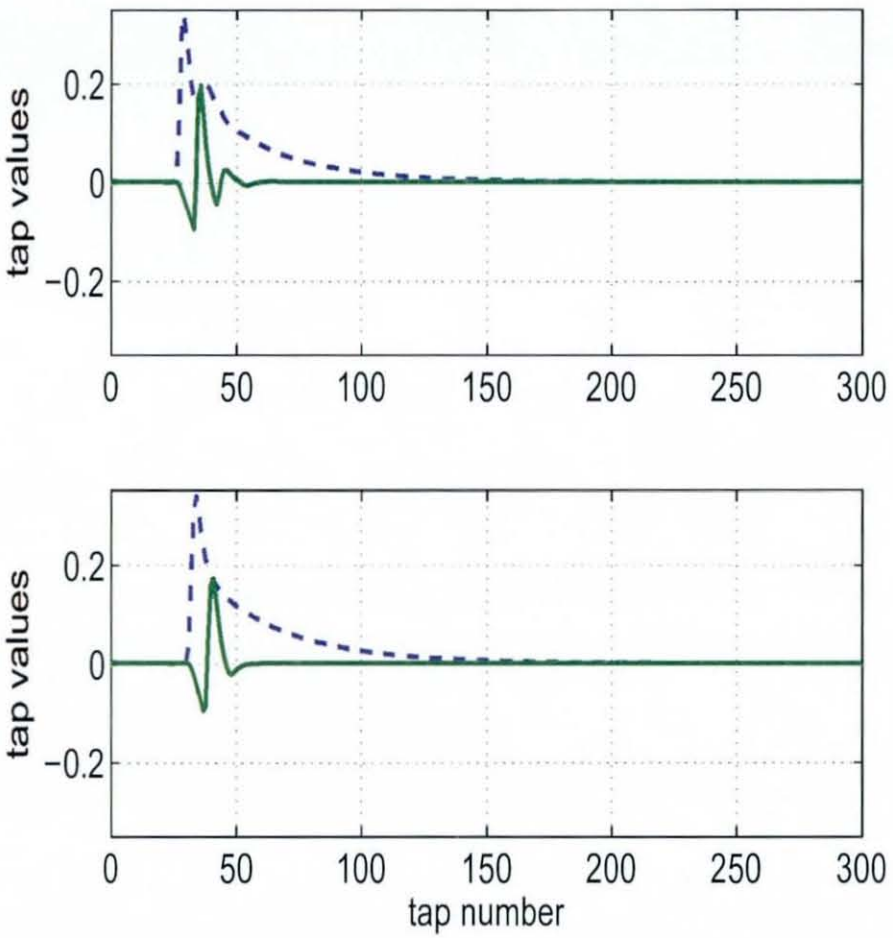

Figure 4.7. Channel shortening of CSA Loop 5 (top) and CSA Loop 6 (bottom) by EGLHSAM with $\alpha = -0.04$. Dotted and solid curves show original and the shortened channel, respectively.

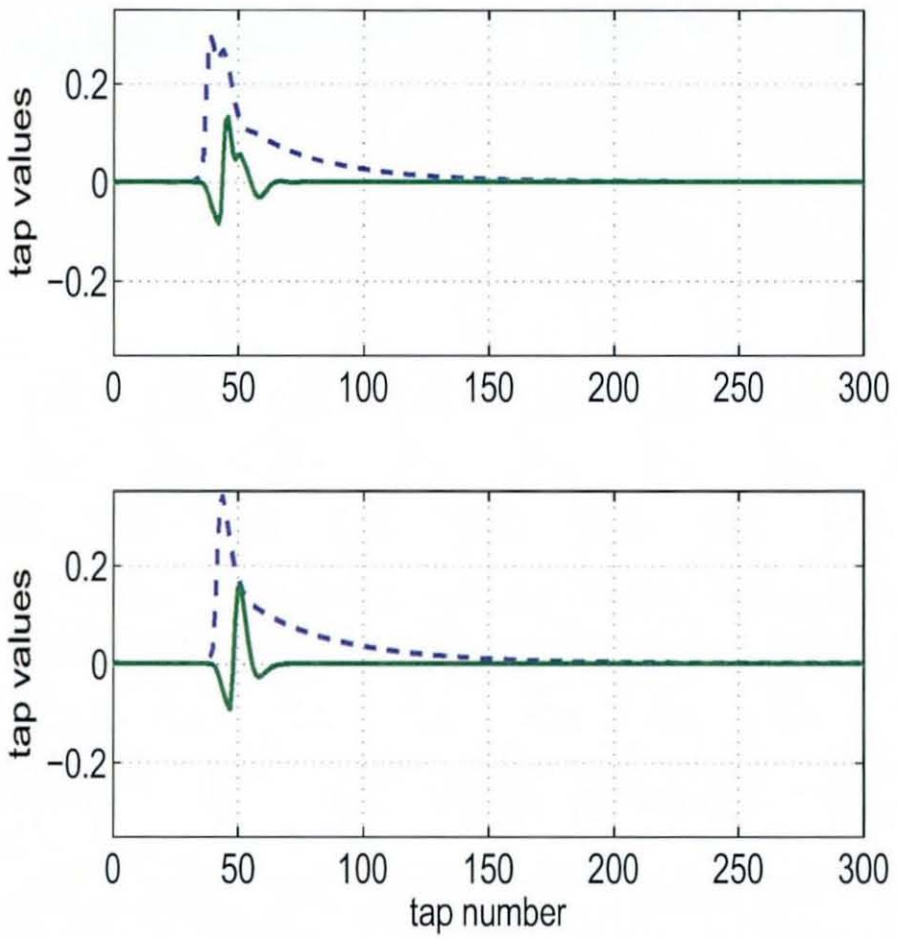

Figure 4.8. Channel shortening of CSA Loop 7 (top) and CSA Loop 8 (bottom) by EGLHSAM with $\alpha = -0.04$. Dotted and solid curves show original and the shortened channel, respectively.

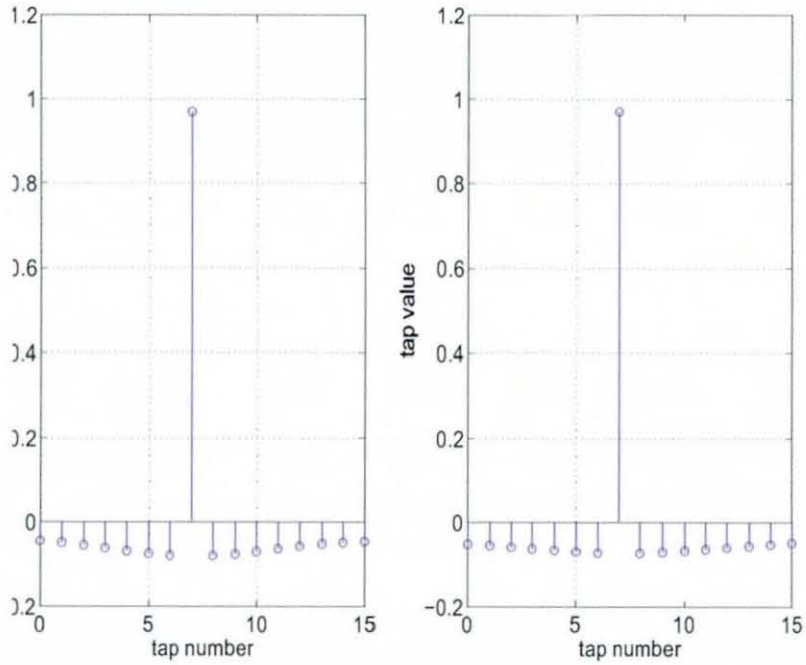

Figure 4.9. Steady state coefficients of the TEQ achieved by the EGLHSAM for CSA Loop 1 (left) and CSA Loop 2 (right).

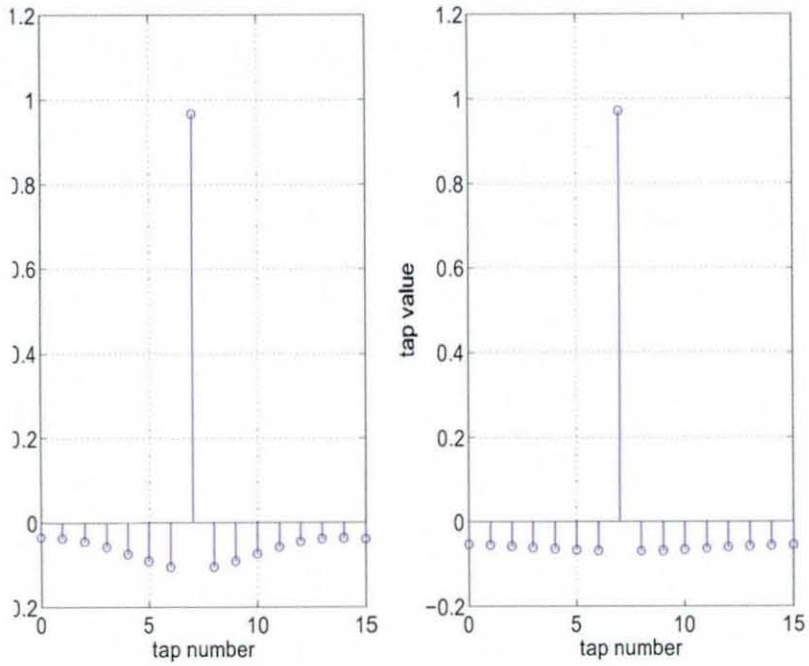

Figure 4.10. Steady state coefficients of the TEQ achieved by the EGLHSAM for CSA Loop 3 (left) and CSA Loop 4 (right).

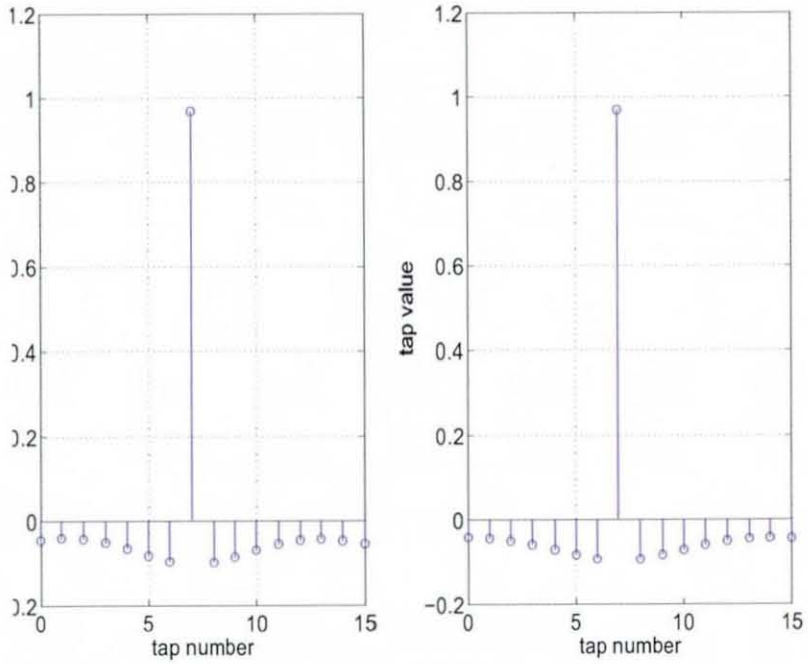

Figure 4.11. Steady state coefficients of the TEQ achieved by the EGLHSAM for CSA Loop 5 (left) and CSA Loop 6 (right).

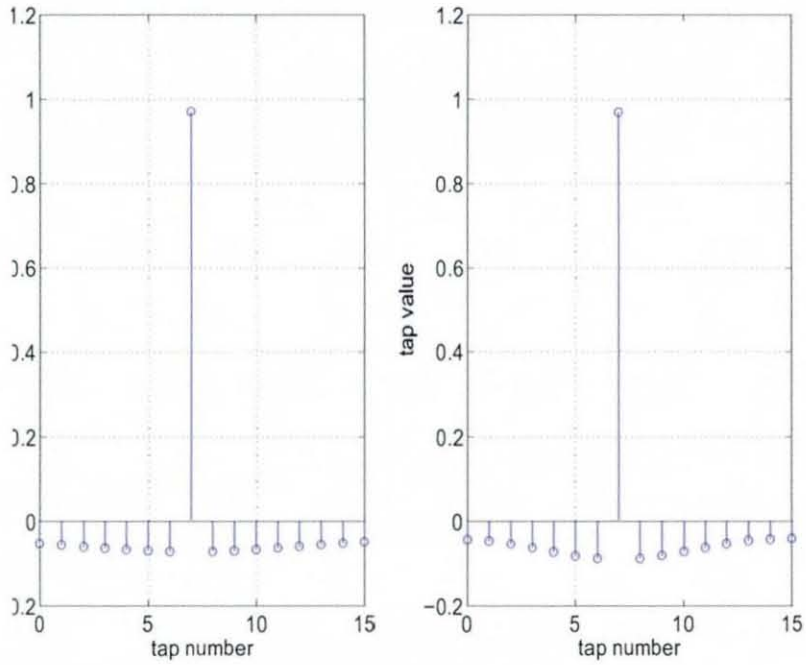

Figure 4.12. Steady state coefficients of the TEQ achieved by the EGLHSAM for CSA Loop 7 (left) and CSA Loop 8 (right).

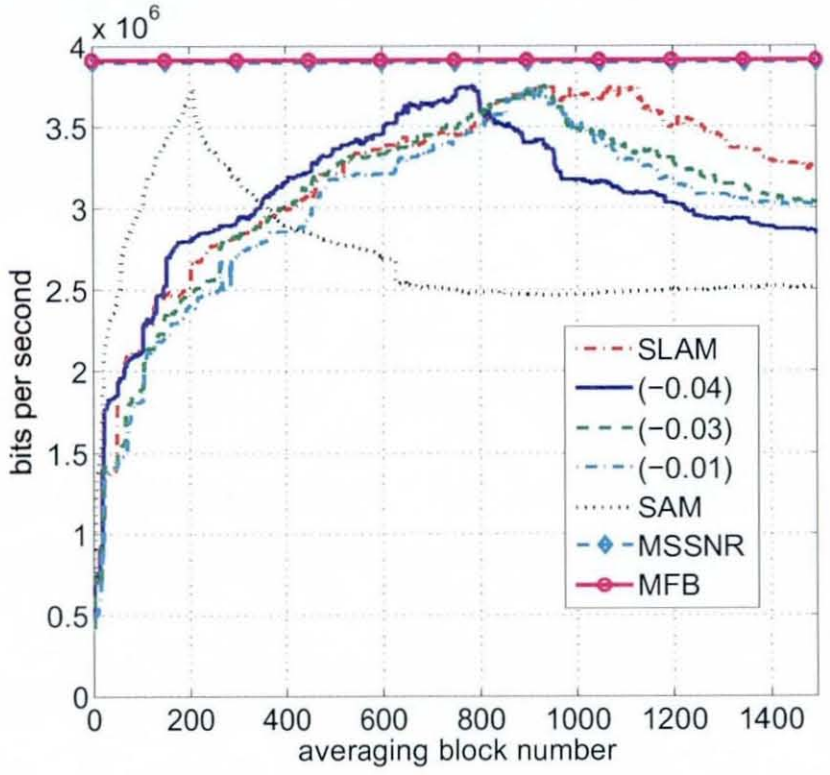

Figure 4.13. Achievable bit rate comparison of GLHSAM with 1, 15 lags with SLAM, SAM, MSSNR and MFB algorithms for CSA Loop 1.

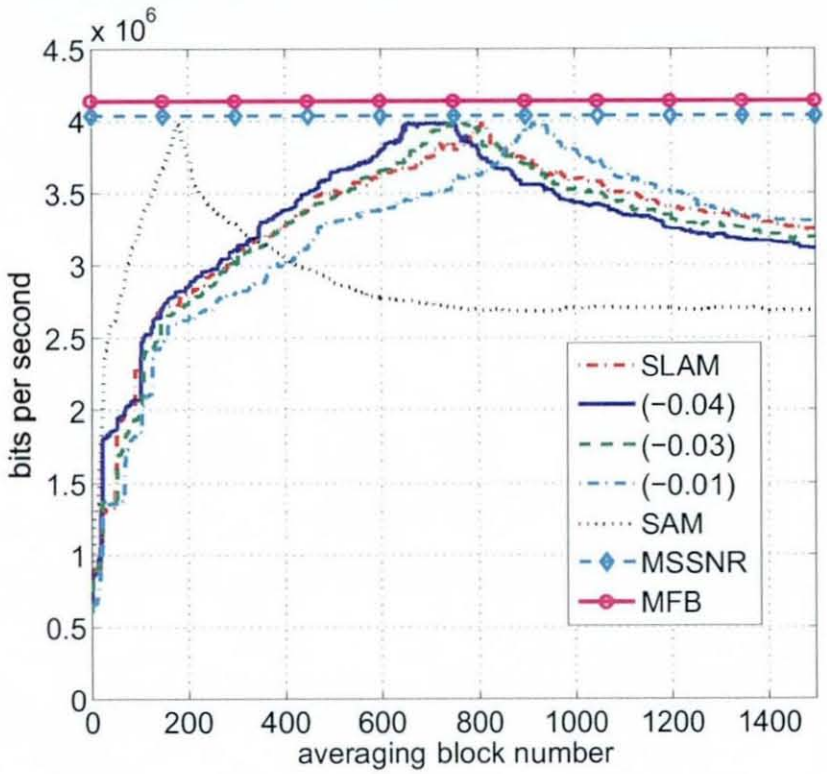

Figure 4.14. Achievable bit rate comparison of GLHSAM with 1, 15 lags with SLAM, SAM, MSSNR and MFB algorithms for CSA Loop 2.

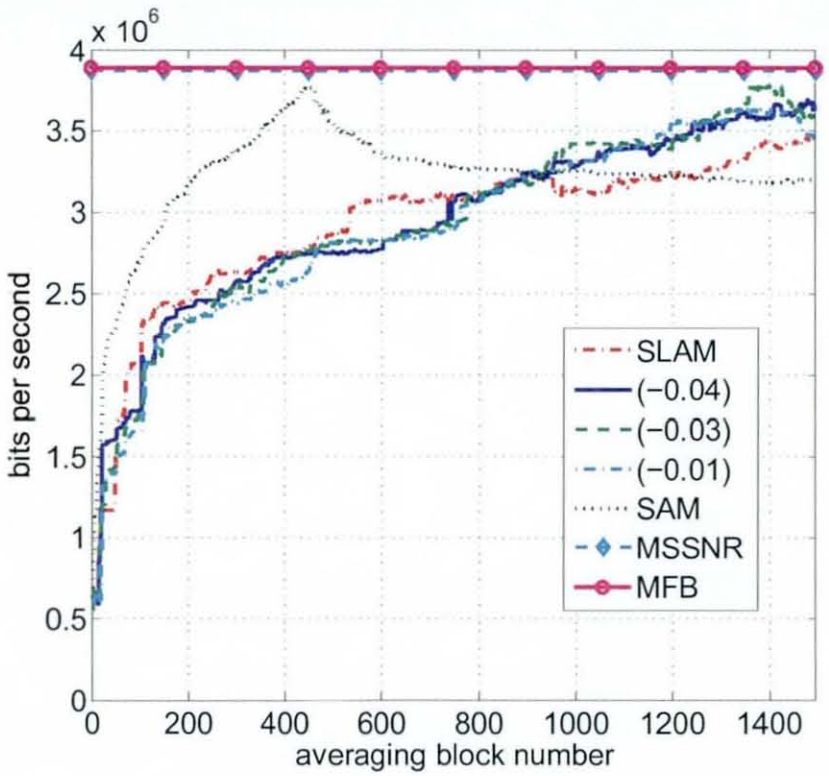

Figure 4.15. Achievable bit rate comparison of GLHSAM with 1, 15 lags with SLAM, SAM, MSSNR and MFB algorithms for CSA Loop 3.

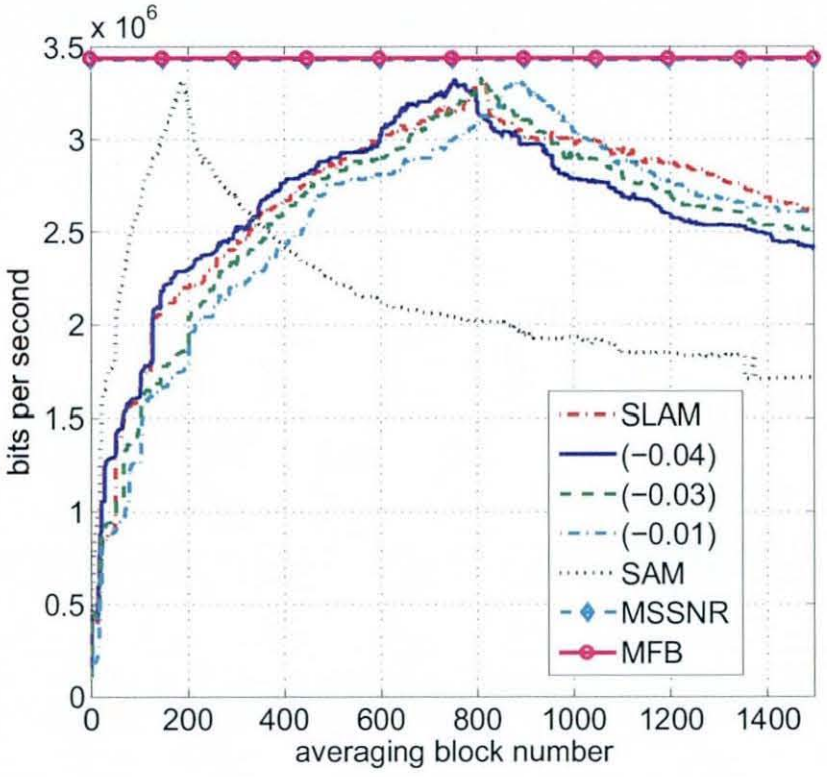

Figure 4.16. Achievable bit rate comparison of GLHSAM with 1, 15 lags with SLAM, SAM, MSSNR and MFB algorithms for CSA Loop 4.

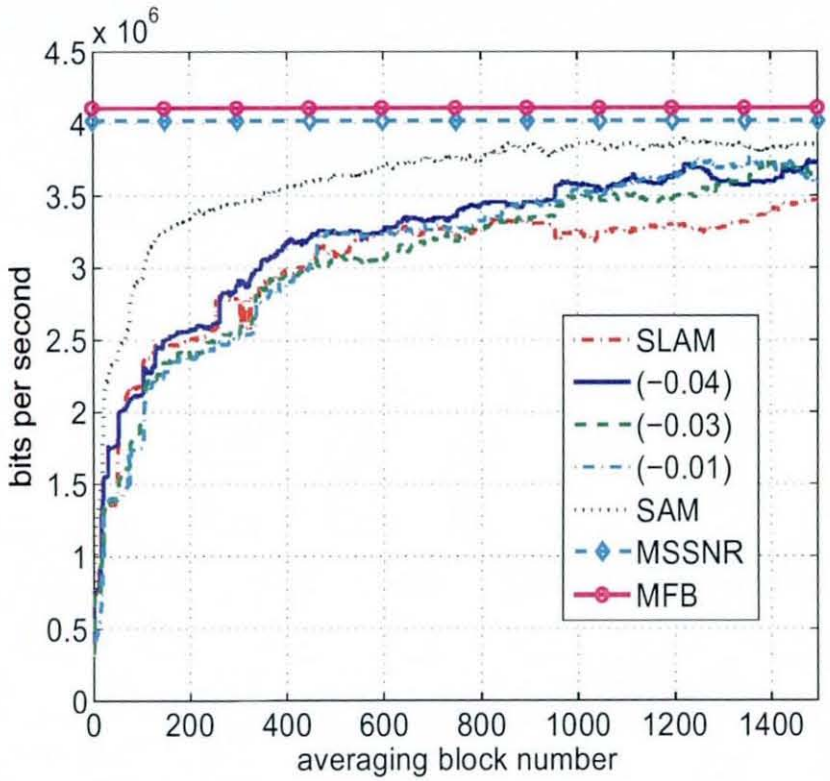

Figure 4.17. Achievable bit rate comparison of GLHSAM with 1, 15 lags with SLAM, SAM, MSSNR and MFB algorithms for CSA Loop 5.

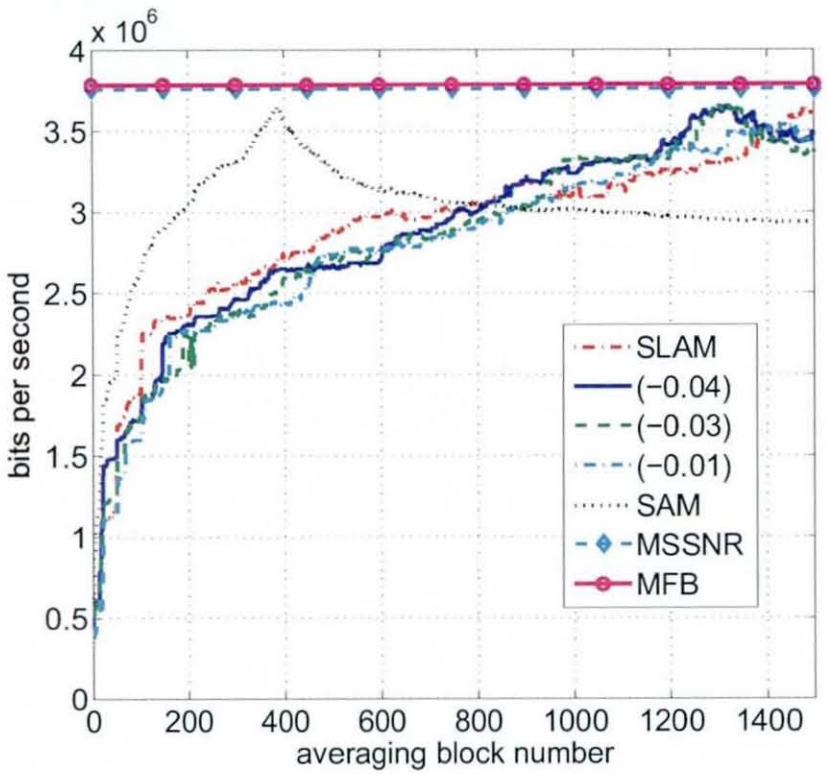

Figure 4.18. Achievable bit rate comparison of GLHSAM with 1, 15 lags with SLAM, SAM, MSSNR and MFB algorithms for CSA Loop 6.

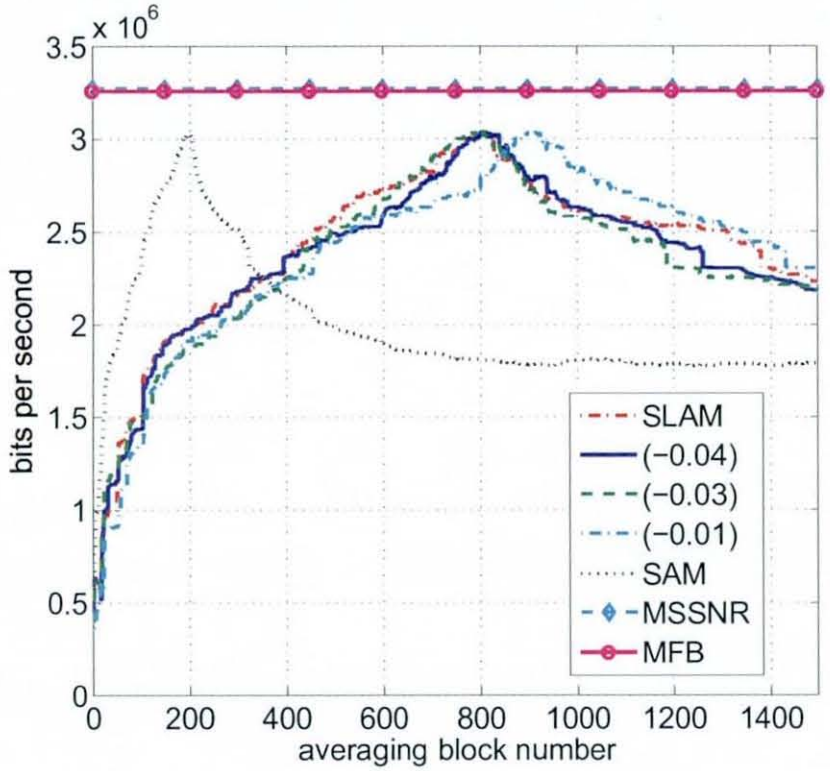

Figure 4.19. Achievable bit rate comparison of GLHSAM with 1, 15 lags with SLAM, SAM, MSSNR and MFB algorithms for CSA Loop 7.

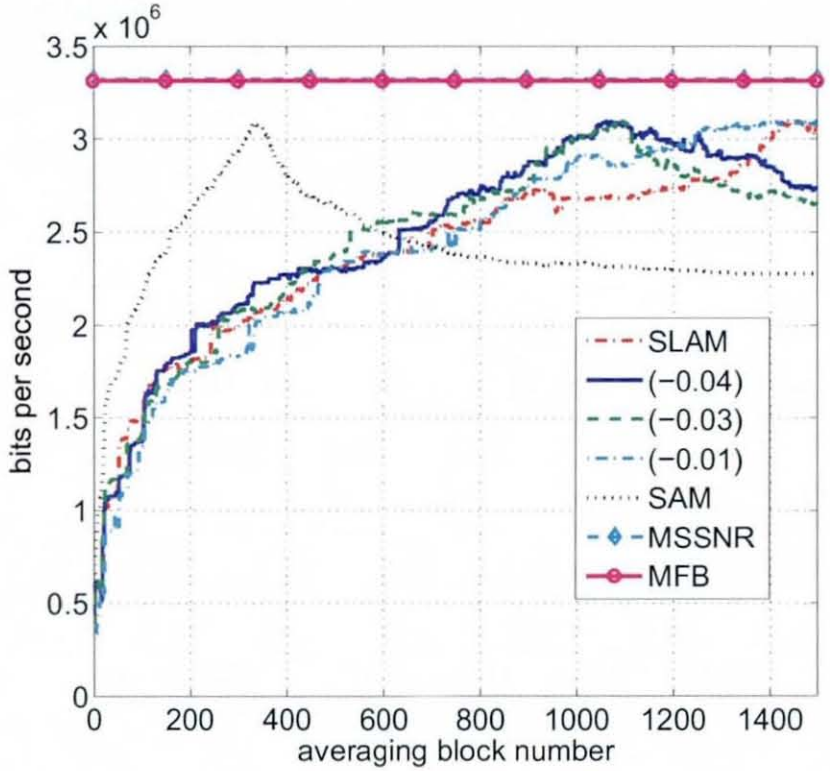

Figure 4.20. Achievable bit rate comparison of GLHSAM with 1, 15 lags with SLAM, SAM, MSSNR and MFB algorithms for CSA Loop 8.

Chapter 5

GENERALIZED LAG HOPPING SAM ALGORITHM (GLHSAM)

5.1 Overview

A generalized blind adaptive lag-hopping channel shortening SAM (GLHSAM) algorithm based upon squared autocorrelation minimization is proposed [67] This algorithm provides the ability to reduce the computational complexity of the sum-squared autocorrelation minimization (SAM) algorithm due to Martin and Johnson as in the single lag autocorrelation minimization (SLAM) algorithm, proposed by Nawaz and Chambers whilst guaranteeing convergence to high signal-to-interference-ratio (SIR). The drawback of the EGLHSAM algorithm in terms of estimating the optimum decaying parameter α is overcome in the GLHSAM algorithm At each iteration a number of unique lags are chosen randomly and uniformly from the whole available range so that on the average GLHSAM has the same cost as the SAM algorithm As, on the average, all of the available lags are chosen, the drawback of the SLAM is also overcome. The performance of the proposed GLHSAM

algorithm is confirmed through simulation studies

5.2 Introduction

A low complexity blind adaptive algorithm to design a TEQ, called sum-squared auto-correlation minimization (SAM) was proposed in [41] which achieves channel shortening by minimizing the sum-squared autocorrelation terms of the effective channel impulse response outside a window of a desired length. The drawback with SAM is that it has a significant computational complexity. SLAM [59], on the other hand, achieves channel shortening by minimizing the squared value of only a single autocorrelation at a lag greater than the guard interval The drawback with the SLAM cost, as noted in the previous chapter, is that a low value does not necessarily guarantee convergence to high SIR for all types of channels [65] As noted in the previous chapter, a low value of EGLHSAM cost also does not necessarily guarantee convergence to high SIR for all types of channels [65]. The contribution in this chapter is therefore to propose a new channel shortening algorithm with random lag selection which has complexity at each iteration as that of SLAM whilst retaining the advantage that a low GLHSAM cost does infer and guarantee high SIR too for all types of channels Plus the new algorithm also does not need to know the value of α

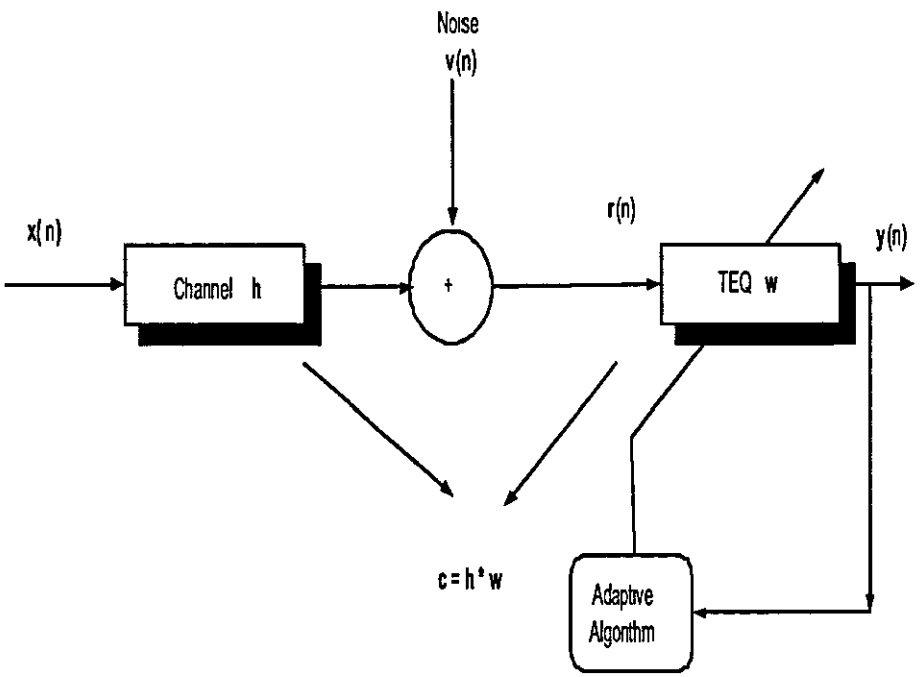

Figure 5.1. Overall baseband channel shortening system model.

5.3 System Model

For the completeness of the chapter, the system model is shown in Figure 5.1. The signal x(n) is a white, zero mean, wide-sense stationary (W.S S.), real and unit variance source sequence, typically drawn from a finite constellation, which is then transmitted through the linear finite-impulse response (FIR) channel $\mathbf{h} = [h(0)h(1)..\ h(L_h)]^T$, v(n) is a zero mean, i i d , noise sequence uncorrelated with the source sequence and has variance σ_v^2. The received signal $r(n)$ is

$$r(n) = \sum_{k=0}^{L_h} h(k)x(n-k) + v(n) \qquad (5.3\ 1)$$

and $y(n)$, the output of the TEQ is given by

$$y(n) = \sum_{k=0}^{L_w} w(k)r(n-k) = \mathbf{w}^T\mathbf{r}_n \qquad (5\ 3\ 2)$$

where $\mathbf{w}$ is the impulse response vector of the TEQ $\mathbf{w} = [w(0)w(1)...w(L_w)]^T$, and $\mathbf{r}_n = [r(n)r(n-1)..\ r(n-L_w]^T$. L_h, L_c, and L_w are the order of the channel, effective channel and the TEQ respectively. It is also assumed that $2L_c \leq N$ holds, N being the FFT size [59] which is a reasonable assumption in the case of ADSL.

5.4 SAM and SLAM Cost Functions

The notion of SAM is founded on the fact that for the effective channel to have zero taps outside a window of size $(v+1)$ its autocorrelation values must be zero outside a window of size $(2v+1)$. In SAM the auto-correlation sequence of the combined channel-equalizer impulse

response becomes

$$R_{cc}(l) = \sum_{k=0}^{Lc} c(k)c(k-l) \tag{5.4.1}$$

and for the shortened channel, the following must hold

$$R_{cc}(l) = 0, \quad \forall |l| > v \tag{5.4.2}$$

The cost function J_{SAM} in SAM is defined on the basis of minimizing the sum-squared auto-correlation terms, i.e.,

$$J_{SAM} = \sum_{l=v+1}^{L_c} R_{cc}(l)^2 \tag{5.4 3}$$

SLAM is based on the fact that a single autocorrelation at a lag greater than the guard interval provides a measure of the presence of the channel outside the desired guard interval, hence minimizing only this single autocorrelation is particularly applicable to subscriber line channels which are essentially minimum phase. In SLAM the auto-correlation sequence of the combined channel-equalizer impulse response is also given by equation (5.4.1) and for a shortened channel, the following must hold

$$R_{cc}(l) = 0, \quad l = v + 1 \tag{5 4 4}$$

In this case the cost function J_{SLAM} in SLAM is defined based upon minimizing the squared-auto-correlation of the effective channel only at lag $l = v + 1$, i e.,

$$J_{SLAM} = R_{cc}(l)^2, \quad l = v + 1 \tag{5.4.5}$$

In [65], however, it has been pointed out that minimizing (5.4.4) only does not guarantee high SIR for certain combined channel and shortener responses To overcome this problem the contribution is to generalize a lag hopping version of SLAM, where at each iteration of the learning algorithm, the lag parameter in (5 4 4) is chosen at random to lie within the range $v+1, ..., L_c$, with equal probability of selecting anyone lag, to the case of selecting randomly, but uniquely, any number of lags between 1 and $L_c - v$, so that on average the cost is identical to (5 4.3) when implemented in an adaptive learning algorithm and the speed of convergence of the algorithm compared to SLAM is likely to be improved. The computational complexity at each iteration of the algorithm could therefore be chosen between that of SLAM and SAM.

5.5 GLHSAM Adaptive Algorithm

The steepest gradient-descent algorithm to minimize the SAM cost J_{SAM} becomes

$$\mathbf{w}^{new} = \mathbf{w}^{old} - \mu \nabla_{\mathbf{w}} \sum_{l=v+1}^{L_c} (E[y(n)y(n-l)])^2 \qquad (5\ 5\ 1)$$

where l is the lag index, μ denotes the step size, and $\nabla_{\mathbf{w}}$ represents the gradient with respect to $\mathbf{w}$. The instantaneous cost at time instant k, where expectation operation is replaced by a moving average over a user-specified window of length N_{avg} is defined as

$$\sum_{l=v+1}^{L_c} (E[y(n)y(n-l)])^2 = \sum_{l=v+1}^{L_c} \left\{ \sum_{n=kN_{avg}}^{(k+1)N_{avg}-1} \frac{y(n)y(n-l)}{N_{avg}} \right\}^2 \qquad (5\ 5\ 2)$$

where N_{avg} is a design parameter and it should be large enough to yield a reliable estimate of the expectation, but no larger, as the algorithm complexity is proportional to N_{avg}. The gradient descent algorithm becomes

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mu\nabla_{\mathbf{w}}\Big(\sum_{L\varepsilon Lagset}\Big\{\sum_{n=kN_{avg}}^{(k+1)N_{avg}-1}\frac{y(n)y(n-l)}{N_{avg}}\Big\}^2\Big)$$

$$\begin{aligned}\mathbf{w}(k+1) = \mathbf{w}(k) - 2\mu &\sum_{L\varepsilon Lagset}\Big\{\sum_{n=kN_{avg}}^{(k+1)N_{avg}-1}\frac{y(n)y(n-l)}{N_{avg}}\Big\}\\ &\times\Big\{\nabla_{\mathbf{w}}\Big(\sum_{n=kN_{avg}}^{(k+1)N_{avg}-1}\frac{y(n)y(n-l)}{N_{avg}}\Big)\Big\}\end{aligned} \tag{5.5.3}$$

and using equation (5.3.2), the GLHSAM algorithm becomes:

$$\begin{aligned}\mathbf{w}(k+1) = \mathbf{w}(k) - 2\mu &\sum_{L\varepsilon Lagset}\Big\{\sum_{n=kN_{avg}}^{(k+1)N_{avg}-1}\frac{y(n)y(n-l)}{N_{avg}}\Big\}\\ &\times\Big\{\Big(\sum_{n=kN_{avg}}^{(k+1)N_{avg}-1}\frac{y(n)\mathbf{r}_{n-l}+y(n-l)\mathbf{r}(n)}{N_{avg}}\Big)\Big\}\end{aligned} \tag{5 5.4}$$

where l_1. .$l_{N_{LAGS}}$ within the Largest elements are chosen to be individually unique and to be drawn with uniform probability from the range of available lags, initially $v+1$, .., L_c. The number of lags, $L_{N_{LAGS}}$, can be chosen over the range 1, .., $L_c - v$, and when $N_{LAGS} = 1$, the algorithm takes the form of a lag-hopping version of SLAM, named GLHSAM(1) in simulations, and when $N_{LAGS} = L_c$ the algorithm is identical to SAM. The key advantage of the random lag hopping in the proposed GLHSAM algorithm is that as $k \to \infty$ since all of the lags in the SAM cost will be visited with probability tending to unity during

adaptation, the average cost which is minimized is identical to that of SAM, and thereby should retain the same convergence properties. Figure (5.2) shows the histogram of the lags minimized by the GLHSAM algorithm in the simulations.

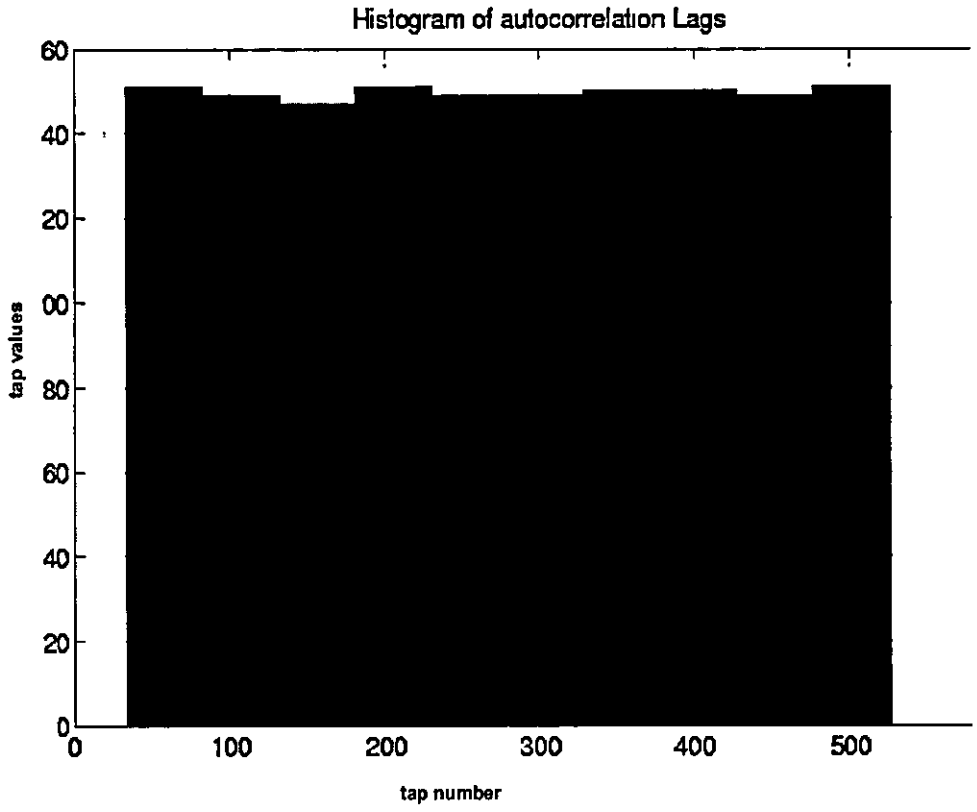

Figure 5.2. Uniform Histogram of lags minimized during the simulations of GLHSAM algorithm.

The complexity of the SLAM algorithm is about 1/500 times that of SAM for typical CSA loop channels [1] GLHSAM(1) enjoys the same advantage with the SLAM algorithm.

5.6 SIR Performance

With reference to equation (4 4 2), the GLHSAM algorithm selects lags randomly with uniform probability, so that a low average GLHSAM cost, achieved through recursive learning, will be identical to a low SAM cost, provided $k \longrightarrow \infty$ which guarantees to give a high SIR at the output of the TEQ, as on the average algorithm it employs all the lags as in SAM. This feature is absent in the EGLHSAM algorithm especially when the decaying parameter α is large

The convergence rate (and hence achievable SIR) in a given adaptation time can be increased by taking more lags in one update of the GLHSAM algorithm. To demonstrate this fact, simulations of the GLHSAM algorithm with 15 random and distinct lags are performed and convergence rate is shown to be improved in the simulations.

5.7 Simulations

The standard parameters of an ADSL downstream transmission were again simulated An MA implementation was simulated for the SAM, SLAM and GLHSAM algorithms The value of N_{avg} was 32. The cyclic prefix had length 32. The FFT size $N_{fft} = 512$, the TEQ had 16 taps and the channels used were the eight test ADSL channels CSA loops provided at [62] The noise was chosen such that $\sigma_x^2 \|\mathbf{c}\|^2/\sigma_v^2 = 40$ dB where $\|.\|$ denotes the Euclidean norm. 89 OFDM symbols were em-

ployed. This value was chosen such that the whole range of available lags is exhausted at least 3 times by the GLHSAM algorithm. Single spike initialization with the center spike of the TEQ initialized to unity was used The step size for SAM was 5; whereas, for SLAM and GLHSAM, it was 600 to get the respective algorithms converge in the given number of symbols and also to keep the results comparable with those of the previous chapter. GLHSAM(15) had a step size of 100. GLHSAM(15) converges faster than SLAM and GLHSAM(1) algorithms and a smaller step size has been chosen to show that even with a smaller step size, it converges earlier than the other two algorithms. All algorithms were compared with the maximum shortening SNR (MSSNR) solution and the matched filter bound (MFB) on capacity, which assumes no ICI

Figure (5 3) shows the original and shortened CSA Loop 1 (top) and 2 (bottom) by the GLHSAM(1) algorithm Similarly, Figures (5.4), (5 5), and (5.6) show the same for CSA Loop 3,4 and 5,6 and 7,8 respectively Figure (5.7) shows steady state coefficients of the TEQ given by the GLHSAM(1) algorithm for CSA Loop 1 (left) and 2 (right). Figures (5.8), (5.9), and (5 10) show the same for CSA Loop 3,4 and 5,6 and 7,8, respectively.

Figure (5 11) compares the achievable bit rates as a function of averaging block number by SAM, SLAM, and GLHSAM(1) and GLHSAM(15) algorithms for CSA Loop 1. GLHSAM(1) converges faster than the SLAM algorithm. As expected GLHSAM(15) is faster than GLHSAM(1) but slower than SAM. The same comments apply to Figure (5 12) which compares the achievable bit rates as a function of averaging block number by SAM, SLAM, and GLHSAM(1) and GLH-

SAM(15) algorithms for CSA loop 2.

Figure (5.13) compares the achievable bit rates as a function of averaging block number by SAM, SLAM, and GLHSAM(1) and GLHSAM(15) algorithms for CSA loop 3. Here the SLAM algorithm has not even converged yet GLHSAM(1) is again faster than SLAM and slower than GLHSAM(15) algorithm. The convergence rates of the algorithms are similarly comparable in Figure (5 14) which compares the achievable bit rates as a function of averaging block number by SAM, SLAM, and GLHSAM(1) and GLHSAM(15) algorithms for CSA loop 4.

Figure (5 15) compares the achievable bit rates as a function of averaging block number by SAM, SLAM, and GLHSAM(1) and GLHSAM(15) algorithms for CSA loop 5. Here again the SLAM algorithm has not even converged yet. GLHSAM(1) is faster than SLAM and slower than GLHSAM(15) algorithm Figure (5.16) compares the achievable bit rates as a function of averaging block number by SAM, SLAM, and GLHSAM(1) and GLHSAM(15) algorithms for CSA Loop 6. GLHSAM(1) converges faster than the SLAM algorithm. As expected GLHSAM(15) is faster than GLHSAM(1) but slower than SAM.

Figure (5.17) compares the achievable bit rates as a function of averaging block number by SAM, SLAM, and GLHSAM(1) and GLHSAM(15) algorithms for CSA Loop 7. Here the SLAM and GLHSAM(1) are comparable but GLHSAM(15) is surely faster than them. Figure (5 18) compares the achievable bit rates as a function of averaging block number by SAM, SLAM, and GLHSAM(1) and GLHSAM(15) algorithms for CSA Loop 8. GLHSAM(1) converges faster than the SLAM algorithm. As expected GLHSAM(15) is faster than GLHSAM(1) but slower than SAM.

The simulations of all the 8 CSA Loops show that GLHSAM with even one lag is faster than the SLAM algorithm. Importantly, the speed with which the GLHSAM algorithm reaches the best performances increases with the number of lags Therefore GLHSAM can use different combinations of convergence speeds and computational complexity and thereby gives the designer the maximum flexibility. It should be noted that the stopping criterion given in [59] is not used in these simulations

5.8 Summary

A new generalized lag hopping blind channel shortening algorithm has been proposed. The proposed algorithm GLHSAM essentially achieves the same result in terms of reducing the effective channel length as SLAM Importantly, the disadvantage of SLAM and EGLHSAM in terms of the SIR performance has been overcome by the proposed algorithm. The algorithm is more general than the EGLHSAM algorithm in that it does need to know the decaying parameter α The algorithm has low complexity as the SLAM algorithm and a low GLHSAM cost is also identical to a low SAM cost as on the average the proposed algorithm uses all the lags as in SAM. It is also demonstrated through simulations that the convergence performance of GLHSAM can be increased by incorporating more lags in its update. Therefore, there is a tradeoff between the complexity of the algorithm and its convergence rate.

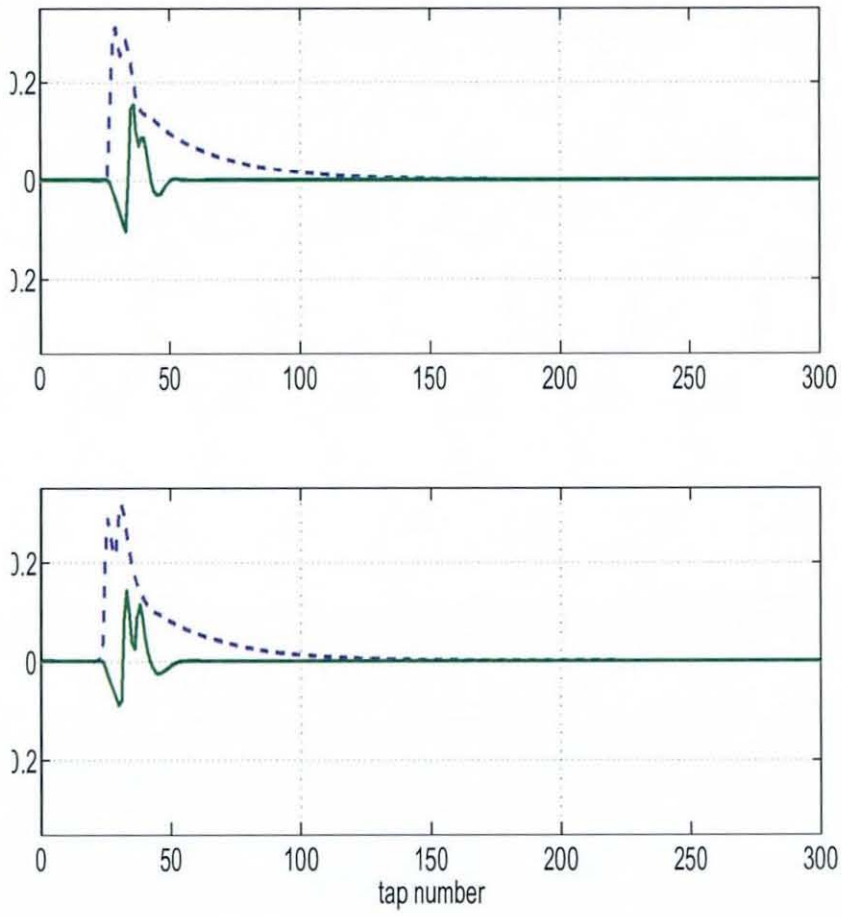

Figure 5.3. Channel shortening of CSA Loop 1 (top) and CSA Loop 2 (bottom) by GLHSAM(1). Dotted and solid curves show original and the shortened channel, respectively.

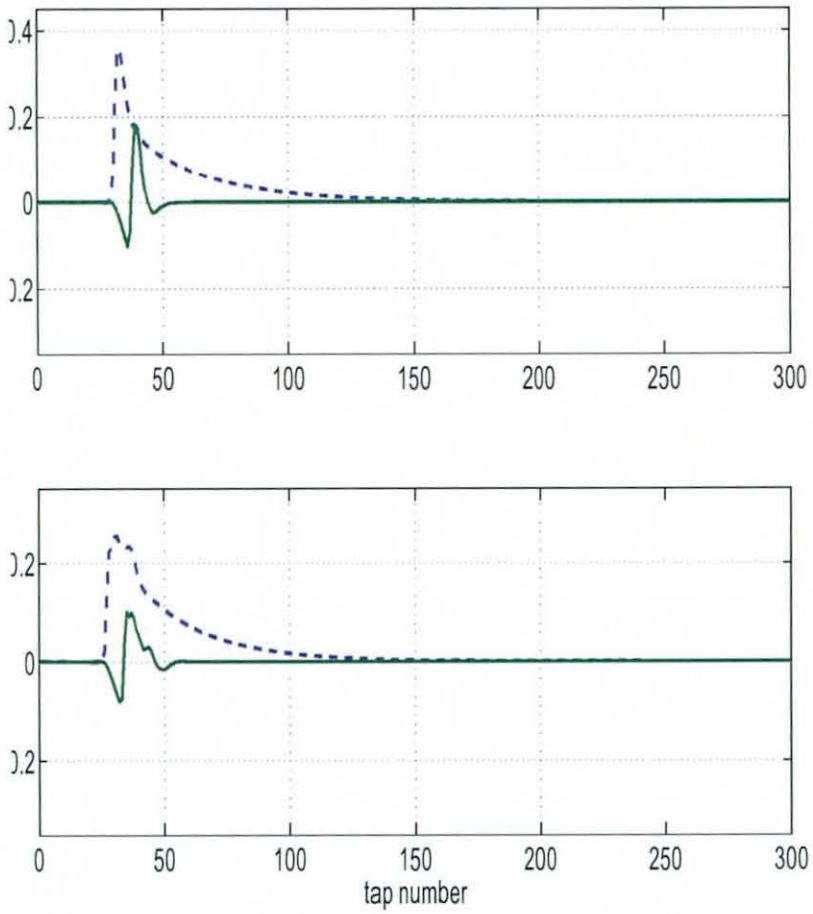

Figure 5.4. Channel shortening of CSA Loop 3 (top) and CSA Loop 4 (bottom) by GLHSAM(1). Dotted and solid curves show original and the shortened channel, respectively.

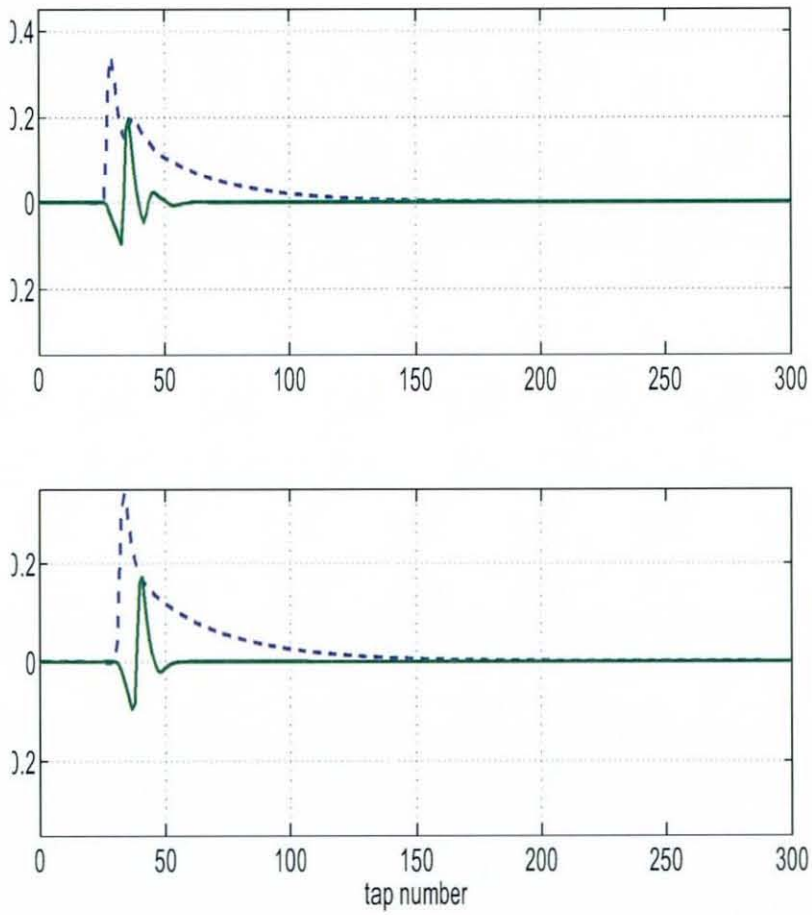

Figure 5.5. Channel shortening of CSA Loop 5 (top) and CSA Loop 6 (bottom) by GLHSAM(1). Dotted and solid curves show original and the shortened channel, respectively.

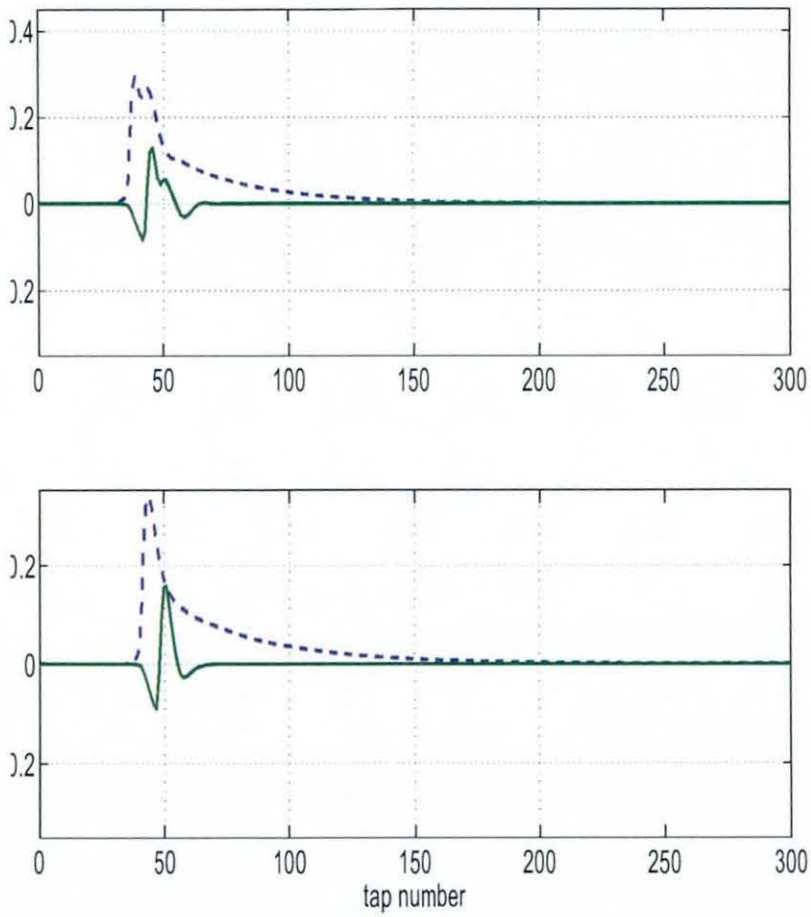

Figure 5.6. Channel shortening of CSA Loop 7 (top) and CSA Loop 8 (bottom) by GLHSAM(1). Dotted and solid curves show original and the shortened channel, respectively.

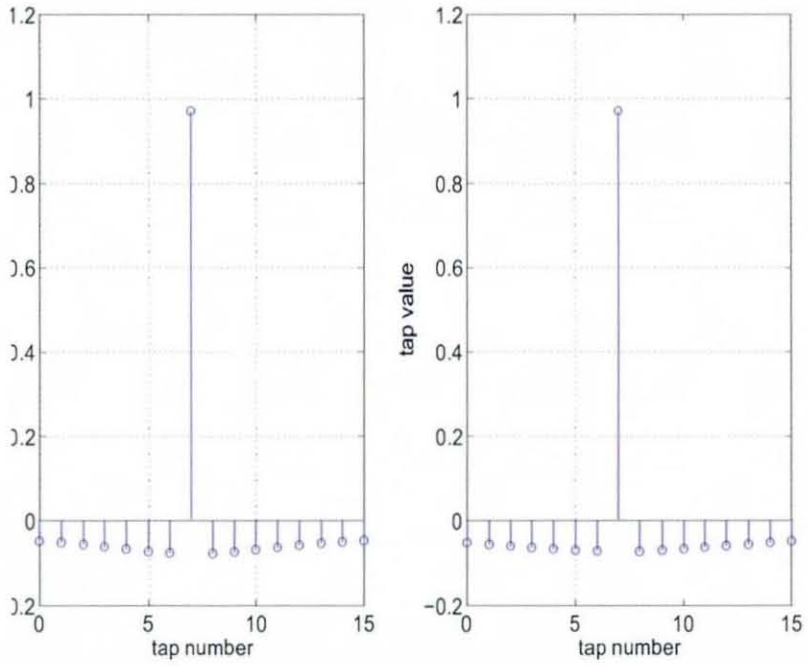

Figure 5.7. Steady state coefficients of the TEQ achieved by the GLHSAM(1) for CSA Loop 1 (left) and CSA Loop 2 (right).

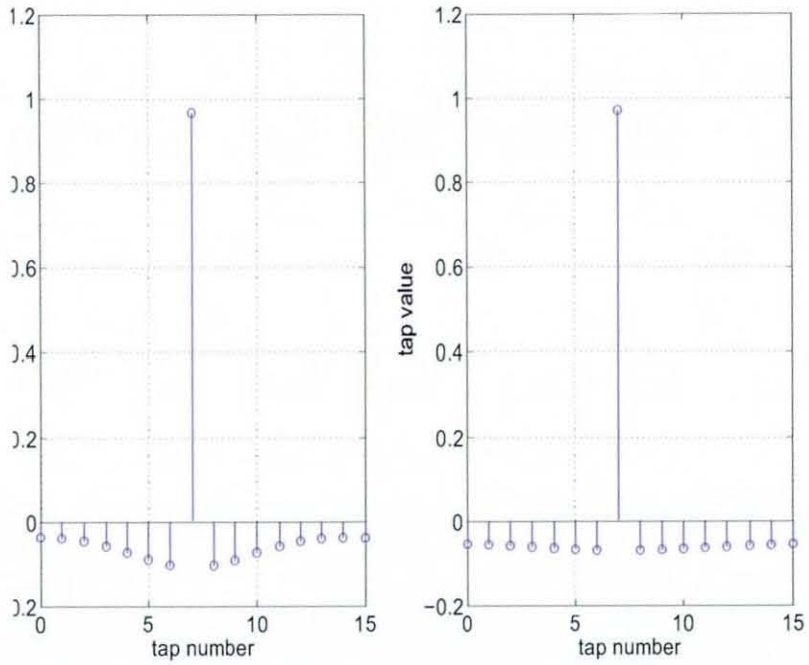

Figure 5.8. Steady state coefficients of the TEQ achieved by the GLHSAM(1) for CSA Loop 3 (left) and CSA Loop 4 (right).

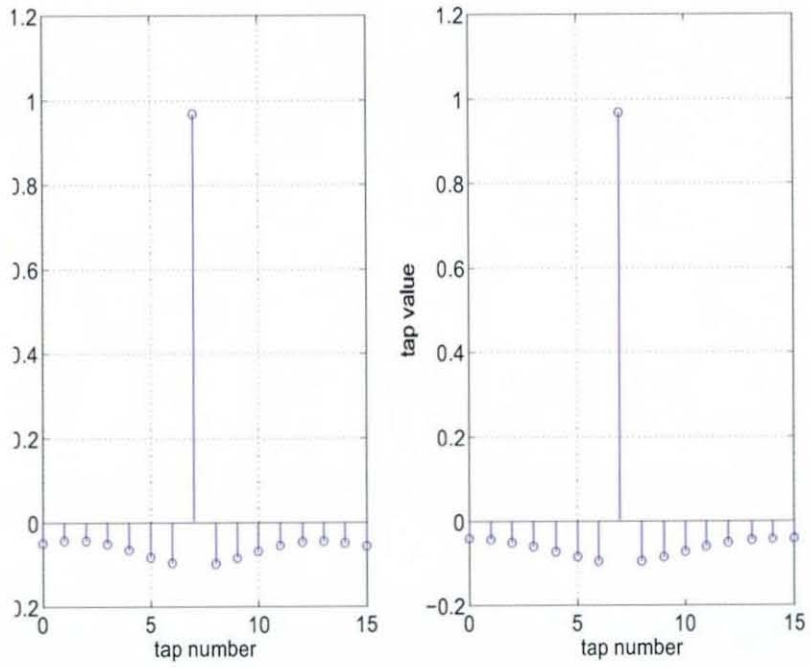

Figure 5.9. Steady state coefficients of the TEQ achieved by the GLHSAM(1) for CSA Loop 5 (left) and CSA Loop 6 (right).

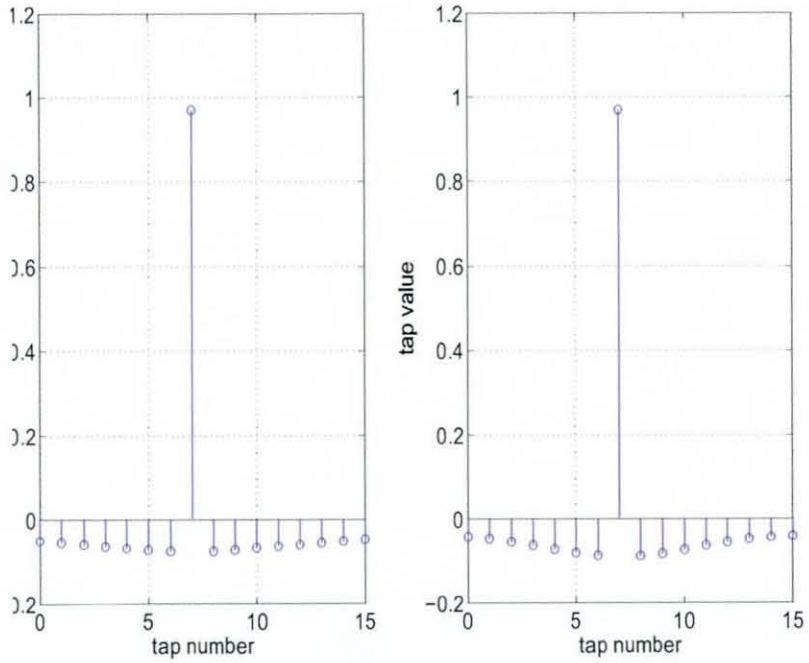

Figure 5.10. Steady state coefficients of the TEQ achieved by the GLHSAM(1) for CSA Loop 7 (left) and CSA Loop 8 (right).

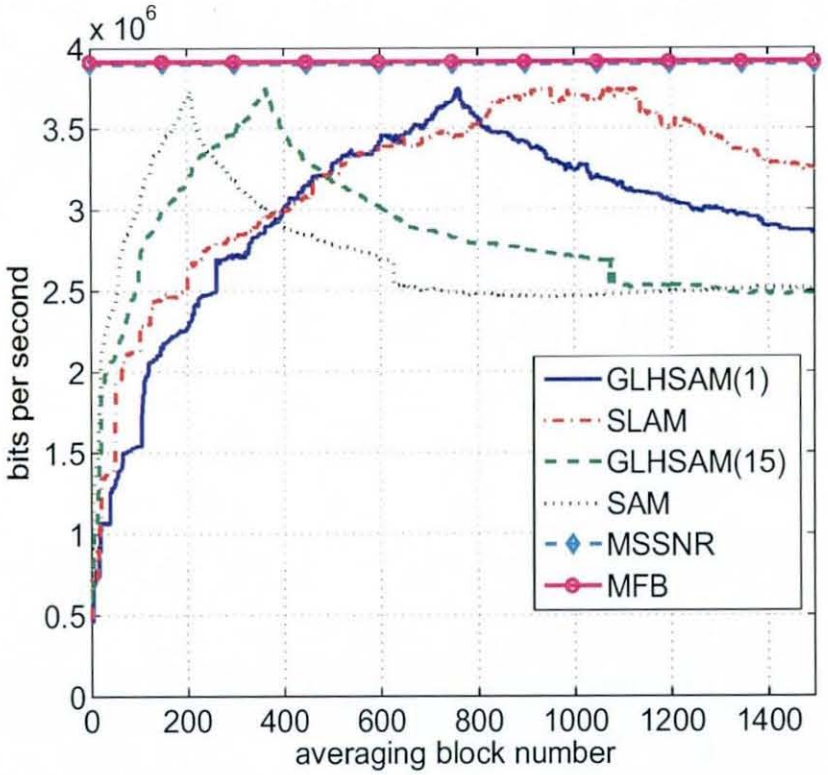

Figure 5.11. Achievable bit rate comparison of GLHSAM with 1, 15 lags with SLAM, SAM, MSSNR and MFB algorithms for CSA Loop 1.

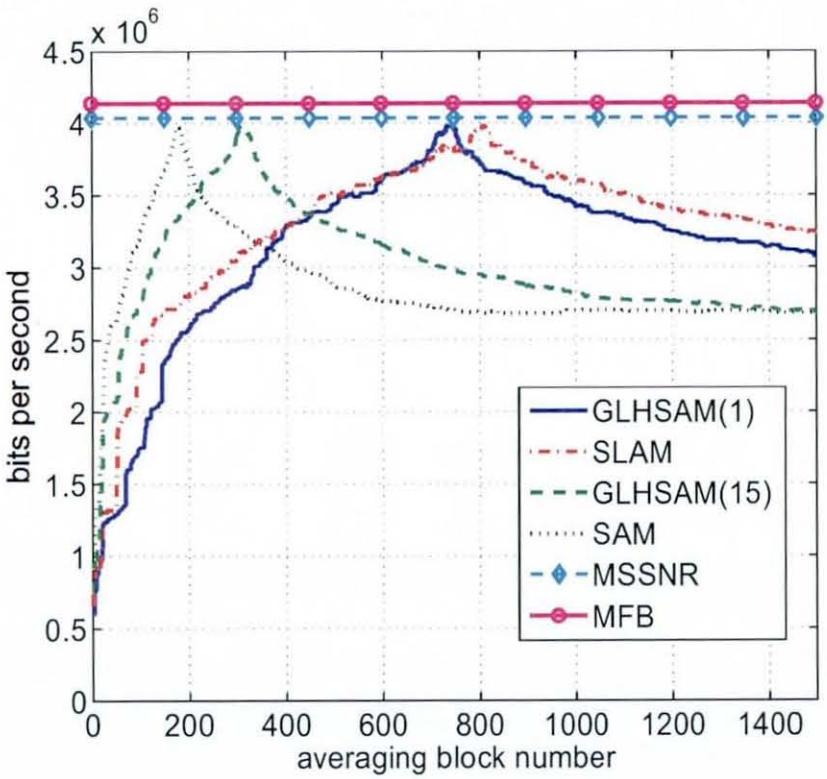

Figure 5.12. Achievable bit rate comparison of GLHSAM with 1, 15 lags with SLAM, SAM, MSSNR and MFB algorithms for CSA Loop 2.

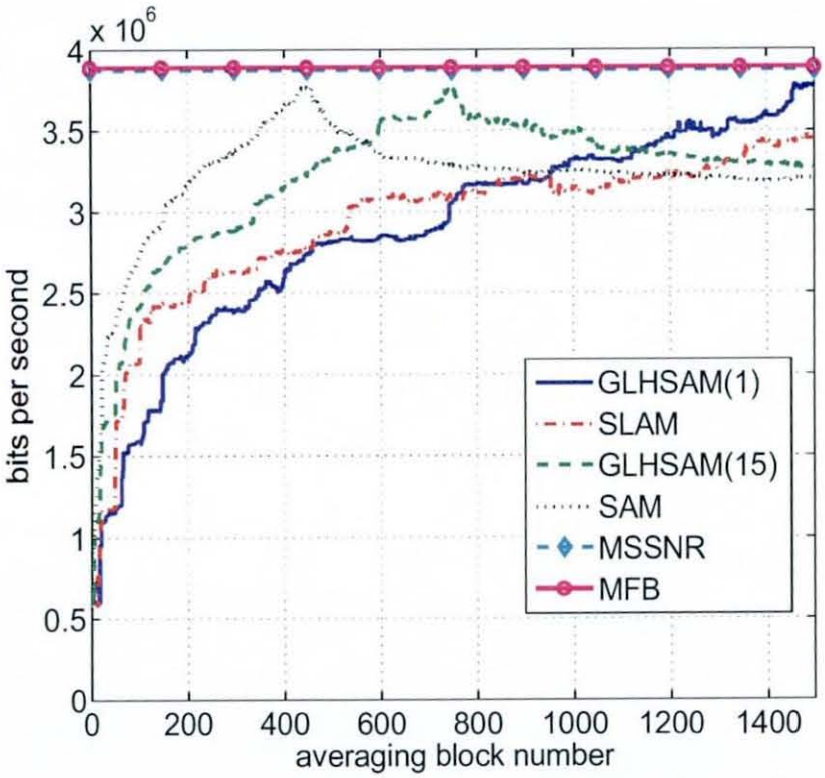

Figure 5.13. Achievable bit rate comparison of GLHSAM with 1, 15 lags with SLAM, SAM, MSSNR and MFB algorithms for CSA Loop 3.

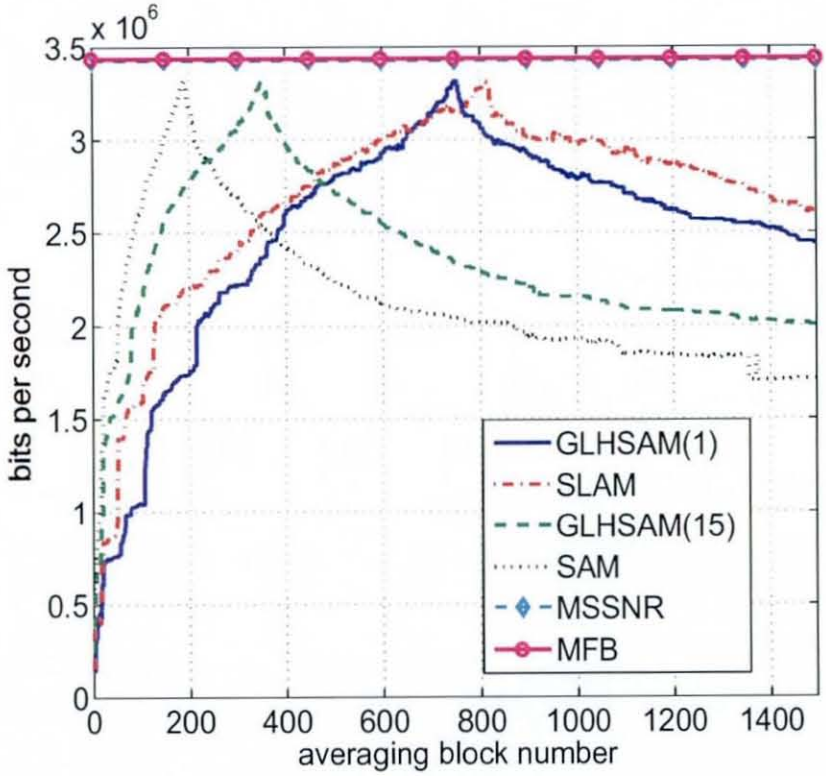

Figure 5.14. Achievable bit rate comparison of GLHSAM with 1, 15 lags with SLAM, SAM, MSSNR and MFB algorithms for CSA Loop 4.

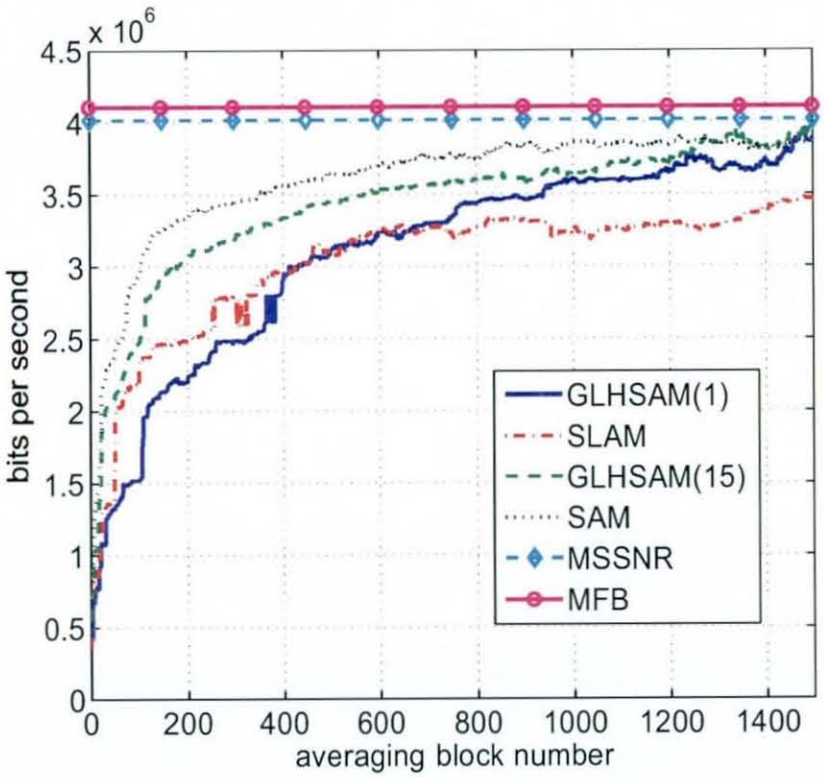

Figure 5.15. Achievable bit rate comparison of GLHSAM with 1, 15 lags with SLAM, SAM, MSSNR and MFB algorithms for CSA Loop 5.

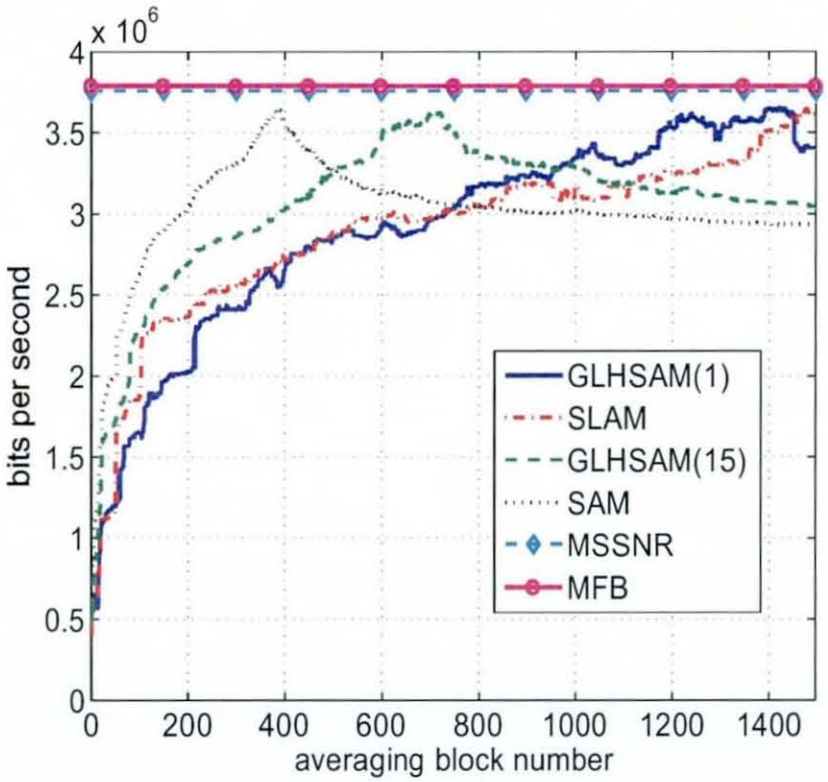

Figure 5.16. Achievable bit rate comparison of GLHSAM with 1, 15 lags with SLAM, SAM, MSSNR and MFB algorithms for CSA Loop 6.

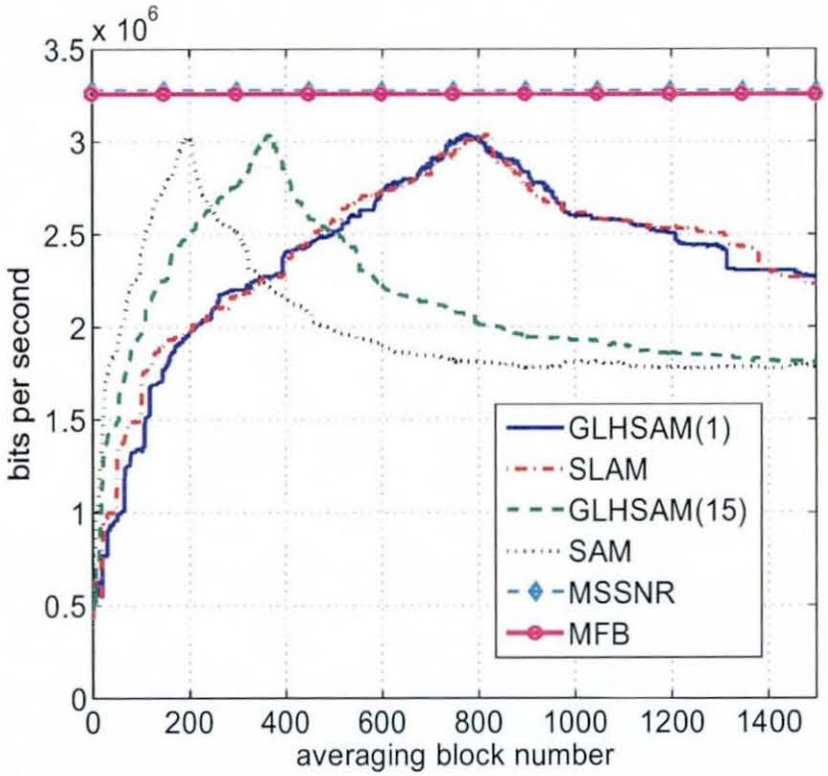

Figure 5.17. Achievable bit rate comparison of GLHSAM with 1, 15 lags with SLAM, SAM, MSSNR and MFB algorithms for CSA Loop 7.

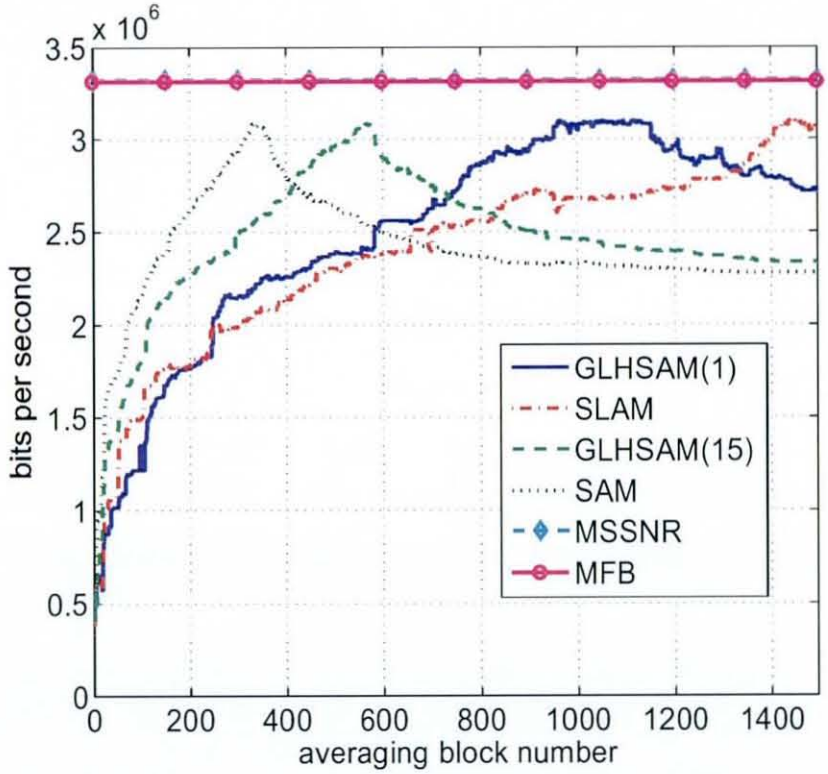

Figure 5.18. Achievable bit rate comparison of GLHSAM with 1, 15 lags with SLAM, SAM, MSSNR and MFB algorithms for CSA Loop 8.

Chapter 6

CONCLUSIONS AND FUTURE WORK

In this chapter general conclusions are drawn and suggestions for further work are given.

Chapter 3 proposes techniques to improve the convergence of the SLAM algorithm. The SLAM is a low complexity channel shortening algorithm as it minimizes the square of only a single fixed autocorrelation. This chapter details the MA and AR implementations of the SLAM algorithm but later uses the MA implementation for the faster cousins of SLAM developed in the chapter. Two schemes have been suggested to improve the convergence of the adaptive SLAM algorithm. The first one VS-SLAM uses a variable step at each iteration of the algorithm. The step size is selected automatically according to the value of the cost at each iteration. The second scheme QN-SLAM uses a faster quadratic type convergence using a quasi Newton descent type update The computational complexity and memory requirements of SLAM, VS-SLAM, and QN-SLAM are provided. It is shown that VS-SLAM has identical complexity as SLAM, whereas QN-SLAM has quadratic complexity in the TEQ length. The proposed two algorithms are compared with SLAM by shortening 8 CSA Loop wireline channels. Both

proposed algorithms successfully shorten the CSA Loop channels. The channel shortening effect and the resulting TEQ designed are shown in the simulations section Achievable bit rate is used as the performance metric to assess the convergence rate of the algorithms. The details of how the achievable bit rate is calculated are provided. The results show that on average VS-SLAM converges faster than SLAM algorithm for all 8 CSA Loop channels. QN-SLAM is faster than SLAM and sometimes converges earlier than the SAM algorithm. However, its response is very noisy. The noisy convergence coupled with very high computational complexity of the QN-SLAM algorithm makes it less useful for real-time channel shortening applications. VS-SLAM appears to be the preferred algorithm.

Chapter 4 proposes an exponential probability generalized lag hopping version of the SLAM algorithm named EGLHSAM. The drawback with SLAM algorithm is that it minimizes a fixed autocorrelation value. There can be some channel impulse responses where the SLAM cost is zero but the channel impulse response is not confined to the required window length. EGLHSAM overcomes this problem by minimizing a random lag at each iteration from the available range of lags. Therefore, in a complete adaptation, it visits all the possible lags This reduces the possibility that EGLHSAM cost is zero but channel is not short as required resulting in a poor SIR Furthermore, the algorithm selects the lags with a probability matching the envelope of the impulse response of the underlying channel This increases the convergence rate of the EGLHSAM algorithm than that of the SLAM algorithm. The chapter gives breakdown of the SIR formula and shows that only minimizing a fixed autocorrelation, as in SLAM, does not provide guarantee that

SIR will be increased. There is a possibility that a few taps outside the required window are left which is against the channel shortening phenomenon. The histograms of the lags simulated are shown. EGLHSAM algorithm is compared with SLAM by shortening 8 CSA Loop wireline channels. Different decaying slopes for the lags are simulated for the EGLHSAM algorithm. It successfully shortens the 8 CSA Loop channels The channel shortening effect and the resulting TEQ designed are shown in the simulations section. Achievable bit rate is again used as the performance metric to assess the convergence rate of the algorithms. Depending upon the decaying slope of the lags, EGLHSAM outperforms SLAM. This 'good' decaying parameter value is different for different CSA Loop channels. This is a problem with EGLHSAM algorithm where it needs the optimum decaying parameter value. It is also mentioned that using a highly decaying nature of the lags probability excludes some of the lags to be minimized However, this is less severe problem than with the SLAM algorithm.

Chapter 5 proposes a generalized lag hopping algorithm but uses uniform probability of lag selection. The algorithm is named GLHSAM and it overcomes problems with SLAM and EGLHSAM algorithm to guarantee high SIR. This algorithm also does not need to know the decaying parameter for every channel GLHSAM algorithm is shown to have identical channel shortening effect as that of SLAM and EGLHSAM algorithms. The convergence rate of GLHSAM is better than SLAM and is comparable to that of EGLHSAM. The convergence rate can be further increased by incorporating more lags in the update while keeping an overall uniform probability of lags selection.

6.1 Future Research

- The performance of the proposed algorithms need to be confirmed for the upstream channels

- Complete equalization in OFDM/ADSL requires the estimation of the FEQ as well. The proposed algorithms can be complemented by providing FEQs designs.

- A stopping criterion has not been used in these algorithms. The cost surface is very shallow for all the proposed algorithms and algorithms are needed to ensure that the solution does not diverge from the global minima

- To calculate the range of lags to be minimized, all the proposed algorithms assume the knowledge of the length of the channel In that sense, they are not truly blind. Although, EGLHSAM and GLHSAM algorithms do shorten the channels even if a reduced range of lags is minimized Therefore, a rough knowledge of the length of channel is not impractical.

- In wireline and wireless systems the channel characteristics may change due to temperature variations or due to the movement of the transmitter and/or the receiver. In this thesis, though, the channel is assumed not to change during at least one OFDM block transmission time. Such channels arise in the case of ADSL

and fixed multipath wireless channels This allows the channel shorteners to mitigate the effects of ISI which is the result of the delay spread of the wireline channel or of the length of multipath of the wireless channel greater than the value of the CP used More challenging extensions of the thesis will address the environments where channel characteristics change more rapidly with time. As with the adaptive equalizers, adaptive channel shorteners are well-suited for time varying channel environments Therefore, the adaptive channel shortening algorithms suggested in this thesis are expected to perform well in such scenarios, too

- Current IEEE 802 11 receivers do not typically employ TEQs because the expected delay spreads are not very long and throughput loss due to CP is small. Future wireless standards such as IEEE 802 11/n or WiMax may be designed for longer channels, necessitating the need for channel shortening to reduce the loss of throughput Multiple-input multiple-output (MIMO) configuration [68] [69], more aggressive coding, including a larger constellation, higher convolutional code rate, and a reduced guard interval are some of the suggestions put forward by the IEEE task groups to improve the data rate in Wi-Fi and take them to as high as 100 Mbps [70]. Channel shortening is the answer to decrease the required length of the guard interval

- Also WiMax uses a variable length guard interval [7] With a very clear channel, a small guard interval is used increasing the

throughput or the spectral efficiency. For longer channels, though, a long CP is required Channel shortening can be used for such channels to keep the spectral efficiency at the maximum while shortener block can be turned off for the clear or smaller delay spread channels. The algorithms can be extended to such scenarios.

REFERENCES

[1] R. Nawaz, "Low Complexity Channel Shortening and Equalization For Multi-Carrier Systems," Ph. D Thesis, Cardiff University, UK 2006.

[2] P J. W. Melsa, R. C. Younce, and C E Rohrs, "Impulse response shortening for discrete multitone transceivers," *IEEE Trans. Commun.*, vol. 44, pp. 1662–1672, Dec. 1996.

[3] R. V Nee and R. Prasad, *OFDM For Wireless Multimedia Communications* Boston, London· Artech House Publishers, 2000.

[4] M. de Courville, P Duhamel, P. Madec, and J. Palicot, "Blind equalization of OFDM systems based on the minimization of a quadratic criterion," *IEEE Int. Conf. on Comm., Dallas, TX*, pp. 1318–1321, May 1996.

[5] R. D. J. van Nee, G A Awater, M. Morikura, H. Takanashiand, M. A Webster, and K. W. Halford, "New high-rate wireless LAN standards," *IEEE Commun. Magazine*, vol. 37, no. 12, pp. 82–88, Dec. 1999

[6] The Inst. of Electrical and Electronics Engineers, "Air Interface for Fixed Broadband Wireless Access Systems, MAC and Additional PHY Specifications for 2-11 GHz IEEE Std. 802.16a,". 2003 Edition.

[7] D Poulin, "The challenges of WiMax design," *IET Communications Engineering Magazine*, Oct. 2006.

[8] The European Telecomm. Standards Inst., "Radio Broadcasting System, Digital Audio Broadcasting (DAB) to Mobile, Portible, and Fixed Receivers,". ETS 300 401, 19951997.

[9] The European Telecomm. Standards Inst , "Digital Video Broadcasting (DVB); Framing Structure, Channel Coding and Modulation for Digital Terrestrial Television,". ETSI EN 300 744 V1.4 1, 2001 Edition.

[10] D. H. Layer, "Digital radio takes to the road," *IEEE Spectrum*, vol. 38, pp 40–46, July 2001.

[11] S. Galli, A Scaglione, and K. Dostert, "Broadband is power: Internet access through the power line network," *IEEE Commun Magazine (special issue)*, vol 41, no. 5, pp. 82–118, May 2003

[12] T Starr, J. M. Cioffi, and P T. Silverman, *Understanding Digital Subscriber Line Technology* Englewood Cliffs NJ: Prentice-Hall, 1999.

[13] A Amirkhany, A Abbasfar, V. Stojanovic, and M. Horowitz, "Practical limits of multi-tone signaling over high-speed backplane electrical links," in *ICC 2007 Signal Processing for Communications Symposium* Glasgow, Scotland, United Kingdom June 2007.

[14] H. Bölcskei, "MIMO-OFDM wireless systems· Basics, perspectives, and challenges," *May 2006 Report.* Communication Technology Laboratory ETH Zurich 8092 Zurich, Switzerland

[15] G. L Stuber, J. R Barry, S. W Mclaughlin, Y. G Li, M. A. Ingram, and T G Pratt, "Broadband MIMO-OFDM wireless communications," *Proceedings of the IEEE*, vol. 92, pp. 271–294, Feb. 2004.

[16] Z. Wang and G. B. Giannakis, "Wireless multicarrier communications. Where Fourier meets Shannon," *IEEE Signal Processing Magazine*, vol. 17, pp 29–48, May 2000.

[17] Z. Wang, X Ma, and G. B Giannakis, "OFDM or single-carrier block transmission?," *IEEE Trans on Communications*, vol. 52, pp. 380–394, Mar. 2004

[18] J S. Chow and J M Cioffi, "A cost-effective maximum likelihood receiver for multicarrier systems," in *Proc IEEE Int. Conf Commun.*, pp 948–952 vol. 2, June, 1992.

[19] M. Milošević, "Maximizing Data Rate Of Discrete Multitone Systems Using Time Domain Equalization Design," Ph D Thesis, The University of Texas at Austin 2003

[20] J. A C Bingham, *ADSL, VDSL, and Multicarrier Modulation.* New York, US: John Wiley & Sons, Inc , 2000.

[21] J. F V. Kerchove and P. Spruyt, "Adapted optimization criterion for FDM-based DMT-ADSL equalization," in *Proc. IEEE Int. Conf. Commun* , pp 1328–1334. June 1996.

[22] G. Arslan, B. L. Evans, and S Kiaei, "Equalization for discrete multitone receivers to maximize bit rate," *IEEE Trans. Signal Processing*, vol. 49, pp. 3123–3135, Dec 2001.

[23] T. Miyajima and Z. Ding, "Multicarrier Channel Shortening Based on Second-Order Output Statistics," *IEEE Workshop on Signal Proc Advances in Wireless Comm , Rome, Italy*, Jun. 2003

[24] B Muquet, M. de Courville, and P. Duhamel, "Subspace-Based Blind and Semi-Blind Channel Estimation for OFDM Systems," *IEEE Trans on Signal Processing*, vol. 50, pp. 1699–1712, Jul. 2002

[25] D. Pal, G. N. Iyengar, and J. M Cioffi, "A New Method of Channel Shortening With Applications to Discrete Multi Tone (DMT) Systems. Part I: Theory," *submitted to IEEE Trans. on Comm.*

[26] D. Pal, G. N. Iyengar, and J. M. Cioffi, "A New Method of Channel Shortening With Applications to Discrete Multi Tone (DMT) Systems," *IEEE Int. Conf. on Comm* , vol. 2, pp. 763–768, Jun 1998.

[27] F. Romano and S. Barbarossa, "Non-Data Aided Adaptive Channel Shortening for Efficient Multi-Carrier Systems," *In Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Proc., Hong Kong SAR, China*, vol 4, pp. 233–236, Jun 2003.

[28] G. Strang, *Linear Algebra and Its Applications* Harcourt Brace Jovanovich, Publishers, San Diego, CA, 1988.

[29] C Yin and G Yue, "Optimal impulse response shortening for discrete multitone transceivers," *Electronics Letters*, vol. 34, pp. 35–36, Jan. 1998.

[30] R Schur, J Speidel, and R. Angerbauer, "Reduction of guard interval by impulse compression for DMT modulation on twisted pair cables," in *IEEE Global Telecomm. Conf.*, pp. 1632–1636. San Francisco, USA, Nov 2000.

[31] B Lu, L. D. Clark, G. Arslan, and B. L. Evans, "Fast time-domain equalization for discrete multitone modulation systems," in *Proc IEEE Digital Signal Processing Workshop.* Hunt, TX, Oct. 2000.

[32] R Schur and J Speidel, "An efficient equalization method to minimize delay spread in OFDM/DMT systems," in *Proc. IEEE Int Conf. Commun* , pp. 1481–1485 June 2001.

[33] A Tkacenko and P. P. Vaidyanathan, "Noise optimized eigenfilter design of time-domain equalizers for DMT systems," in *Proc. IEEE Int. Conf. Commun.*, pp 54–58. May 2002

[34] A. Tkacenko and P. P. Vaidyanathan, "Eigenfilter design of MIMO equalizers for channel shortening," in *Proc. IEEE Int. Conf. Commun.*, pp. 2361–2364. May 2002.

[35] M. G. Troulis and S. Sesia, "A spectrally flat time domain equalizer for rate improvement of multicarrier systems," in *Proc IEEE Int. Conf. Commun.*, pp 1803–1807. May 2002.

[36] R K. Martin, J. Balakrishnan, W. A. Sethars, and C R. Johnson, "A blind, adaptive TEQ for multicarrier systems," *IEEE Signal Processing Letters*, vol. 9, pp. 341–343, Nov. 2002.

[37] M. Nafie and A. Gatherer, "Time-Domain Equalizer Training for ADSL," *In Proc. IEEE Int Conf. on Comm , Montreal,Canada*, vol 2, pp 1085–1089, Jun. 1997.

[38] W. A. S. R. K. Martin, J. Balakrishnan and C. R Johnson, "Jr. A Blind, Adaptive TEQ for Multicarrier Systems," *IEEE Signal Processing Letters*, vol. 9(11), pp 341–343, Nov. 2002.

[39] R. K. Martin, J M. Walsh, and C R Johnson, "Low complexity MIMO blind adaptive channel shortening," in *Proc. In. Conf. on Acoustics, Speech, and Signal Processing.* Montreal, Quebec, May 2004.

[40] R. K Martin, "Blind, Adaptive Equalization for Multicarrier Receivers," Ph. D. Thesis, Cornell University, US 2004.

[41] J. Balakrishnan, R K. Martin, and C. R. Johnson, "Blind, adaptive channel shortening by sum-squared auto-correlation minimization (SAM)," *IEEE Trans. Signal Processing*, vol. 51, no 12, pp. 3086–3090, Dec. 2003

[42] J. Balakrishnan, R. K Martin, and C. R. Johnson, "Jr Blind, Adaptive Channel Shortening by Sum-squared Auto-correlation Minimization (SAM)," *IEEE Asilomar Conf on Signals, Systems, and Computers,Pacific Grove, CA*, pp. 1867–1875, Nov. 2002

[43] G Arslan, "Equalization for Discrete Multitone Transceivers," Ph D. Thesis, University of Texas at Austin, 2000.

[44] K V Acker, G. Leus, M. Moonen, O van de Wiel, and T Pollet, "Per tone equalization for DMT-based systems," *IEEE Trans Commun.*, vol 49, no 1, pp. 109–119, Jan. 2001.

[45] K. V Acker, G. Leus, M Moonen, O van de Wiel, and T Pollet, "Per tone equalization for dmt-based systems," *IEEE Trans. on Comm* , vol. 49, pp 109–119, Jan. 2003.

[46] K. V. Acker, G. Leus, M Moonen, and T. Pollet, "Rls-based initialization for per-tone equalizers in dmt receivers," *IEEE Trans. on Comm* , vol. 51, pp 885–889, Jun. 2003

[47] G C. G. Ysebaert, K Vanbleu and M. Moonen, "Adsl per-tone/pergroup optimal equalizer and windowing design," *In Proc. Thirty-Sixth Asilomar Conf on Signals, Systems, and Computers*, Nov. 2002

[48] G Ysebaert, K. Vanbleu, G. Cuypers, M. Moonen, and T. Pollet, "Combined rls-lms initialization for per tone equalizers in dmt-receivers," *In Proc. Thirty-Sixth Asilomar Conf. on Signals, Systems, and Computers*, vol. 51, pp. 1916–1927, Nov. 2003

[49] G. Leus and M. Moonen, "Per-tone equalization for MIMO OFDM systems," *IEEE Trans on Signal Processing, Special Issue on Signal Processing for MIMO Wireless Commun. Systems*, vol. 51, no. 11, pp. 2965–2975, Nov. 2003.

[50] I. Berhumi, G Leus, and M Moonen, "Per-tone equalization for OFDM over doubly-selective channels," in *Proc. Int Conf. on Commun.*, pp. 2642–2647. Paris, France, June 2004.

[51] S. C. Douglas, "Adaptive filters emplying partial updates," *IEEE Trans. on Trans Circuits Sys.*, vol. 44, pp 209–216, Mar 1997.

[52] B. Widrow, J McCool, M. G Larimore, and C. R. Johnson, "Jr. stationary and nonstationary learning characteristics of the lms adaptive filter," *Proceedings of the IEEE*, vol. 64, pp. 1151–1162, Aug. 1976

[53] K. V Acker, G. Leus, M. Moonen, and T. Pollet, "RLS-based initialization for Per-Tone equalizers in DMT receivers," *IEEE Trans. Commun* , vol. 51, no. 6, June 2003

[54] R K Martin and C. R Johnson, "Blind, adaptive, Per Tone equal-

ization for multicarrier receivers," in *Conference on Information Sciences and Systems.* Princeton University, Mar. 2002.

[55] D. N. Godard, "Self-Recovering Equalization and Carrier Tracking in Two-Dimensional Data Communication Systems," *IEEE Trans on Comm., COM-28,* pp 1867–1875, Nov. 1980.

[56] M. Ding, R. Redfern, and B L Evans, "A dual-path TEQ structure for DMT ADSL systems," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing,* pp. 2573–2576. April, 2002

[57] M. Milosevic, L. F. C. Pessoa, and B. L. Evans, "Simultaneous multichannel time domain equalizer design based on the maximum composite shortening SNR," in *Proc. IEEE Asilomar Conf on Signals, Systems and Comp.*, pp. 1895–1899. vol. 2, Pacific Grove, CA, November 2002.

[58] M. de Courville, P. Duhamel, P. Madec, and J. Palicot, "Blind equalization of OFDM systems based on the minimization of a quadratic criterion," in *Proc. Int Conf. Commun* , pp. 1318–1321. Dallas, TX, June 1996.

[59] R. Nawaz and J. A. Chambers, "A novel single lag auto-correlation minimization (SLAM) algorithm for blind adative channel shortening," in *Proc Int. Conf. on Acoustics, Speech, and Signal Proc* Philadelphia, PA, USA, Mar. 2005.

[60] V J. Mathews and Z. Xie, "A stochastic gradient adaptive filter with gradient adaptive step size," *IEEE Trans on Signal Processing,* vol. 41, pp. 2075–2087, June 1993

[61] R. K. Martin and C. R. Johnson, "Adaptive equalization Transi-

tioning from single-carrier to multi-carrier systems," *IEEE Signal Processing Magazine*, vol. 22, pp 108–122, Nov. 2005

[62] R. K. Martin. Matlab Code for Papers by R. K. Martin [Online]. Available: http://bard ece cornell edu/matlab/martin/index.html

[63] K. Sistanizadeh, "Loss characteristics of the proposed canonical ADSL loops with 100-Ohm termination at 70, 90, and 120 F," *ANSI T1E1.4 Committee Contribution*, vol. 161, Nov 1991.

[64] K. Maatoug and J. A. Chambers, "A generalized blind lag hopping adaptive channel shortening algorithm based upon squared autocorrelation minimization (GLHSAM)," *Third International Conference on Systems and Networks Communications (ICSNC)*, pp 75–78, Oct. 2008.

[65] J. M Walsh, R K. Martin, and C. R Johnson, "Convergence and performance of auto-correlation based channel shorteners," *Signals, Systems, and Computers*, 2006.

[66] R. Nawaz and J. A Chambers, "Blind adaptive channel shortening by single lag auto-correlation minimization (SLAM)," *Electronics Letters*, vol. 40, pp 1609–1611, Dec. 2004.

[67] K Maatoug and J. A Chambers, "A generalized blind lag hopping adaptive channel shortening algorithm based upon squared autocorrelation minimization," *The 8th IMA International Conference on Mathematics in Signal Processing*, Dec. 2008.

[68] S. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE Journal on Selected Areas in Communications*, vol 16, pp. 1451–1458, Oct. 1998.

[69] V. Tarokh, N. Seshadri, and A. R Calderbank, "Space-time codes for high data rate wireless communication: performance criterion and code construction," *IEEE Trans. on Information Theory*, vol 44, pp 744–765, Mar. 1998.

[70] M. S. Gast, "IEEE 802.11 networks the Definitive Guide," in *IEEE 802.11 networks, chapter 5.* available online at http://www.oreilly com/catalog/802dot112/chapter/ch15 pdf.