# Partial Update Blind Adaptive Channel Shortening Algorithms For Wireline Multicarrier Systems

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

Mahmud Grira

Center of Digital Signal Processing
Cardiff University
2008

# ABSTRACT

In wireline multicarrier systems a cyclic prefix is generally used to facilitate simple channel equalization at the receiver. The choice 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 is greater than the cyclic prefix, adaptive channel shorteners can be used to force the effective channel length of the combined channel and channel shortener to be within the cyclic prefix constraint. The focus of this thesis is the design of new blind adaptive time-domain channel shortening algorithms with good convergence properties and low computational complexity.

An overview of the previous work in the field of supervised partial update adaptive filtering is given. The concept of property-restoral based blind channel shortening algorithms is then introduced together with the main techniques within this class of adaptive filters. Two new partial update blind (unsupervised) adaptive channel shortening algorithms are therefore introduced with robustness to impulsive noise commonly present in wireline multicarrier systems.

Two further blind channel shortening algorithms are proposed in which the set of coefficients which is updated at each iteration of the algorithm is chosen deterministically. One of which, the partial update single lag autocorrelation maximization (PUSLAM) algorithm is particularly attractive due to its low computational complexity.

The interaction between the receiver matched filter and the channel shortener is considered in the context of a multi-input single-output environment. To mitigate the possibility of ill-convergence with the PUSLAM algorithm an entirely new random PUSLAM (RPUSLAM) algorithm is proposed in which randomness is introduced both into the lag selection of the cost function underlying SLAM and the selection of the particular set of coefficients updated at each algorithm. This algorithm benefits from robust convergence properties whilst retaining relatively low computational complexity. All algorithms developed within the thesis are supported by evaluation on a set of eight carrier serving area test loop channels.

# ACKNOWLEDGEMENTS

I would like to thank my thesis advisor Prof. Jonathon Chambers, without his invaluable support and mentoring this thesis would have not been accomplished. He manages to strike the perfect balance between providing direction and encouraging independence. He has contributed to all papers and the thesis with a major impact. He has my deepest respect professionally and personally. Prof. Brian Evans of the University of Texas, Austin should also be thanked for the DMT simulation toolbox. I am moreover grateful to Dr. Andreas Jakobsson for his guidance with the issues in LaTeX. The research office staff within the School of Engineering have also been extremely helpful. I thank my wife Iman for her support during the years. She made me a happy person and gave me the extra strength and motivation necessary to get things done. Finally, I would like to thank my family especially my mother and my father for their patience and encouragement while we have been separated for many years.

# STATEMENT OF ORIGINALITY

I hereby declare that the work described in this thesis was carried out entirely by the author, in the Center of Digital Signal Processing, Cardiff School of Engineering, Cardiff University. The thesis does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university. And that to the best of my knowledge, it does not contain any materials previously published or written by any other individual except where due reference is made in the text.

# PUBLICATIONS

- M. Grira and J. A. Chambers,"Partial update single lag autocorrelation minimization for channel shortening (PUSLAM)," *Institute of Mathematics & its applications (IMA) conf. on Mathematics in Signal Processing VII*, Cirencester, UK, Dec. 2006.

- M. Grira and J. A. Chambers, "Random partial update sum-squared autocorrelation minimization algorithm for channel shortening (RPUSAM)," *15th International Conference on Digital Signal Processing*, Cardiff, UK, Jul. 2007.

- M. Grira and J. A. Chambers, "Adaptive partial update channel shortening in impulsive noise environments," *The 3rd International Symposium on Communications, Control and Signal Processing (ISCCSP 2008)*, St.Julians, Malta, Mar. 2008.

- M. Grira and J. A. Chambers, "A blind lag-hopping adaptive channel shortening algorithm based upon squared autocorrelation minimization (LHSAM)," *The 33rd International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2008)*, Las Vegas, Nevada, USA, Mar. 2008.

# LIST OF ACRONYMS

| | |
|---|---|
| **ADSL** | Asymmetric Digital Subscriber Line |
| **AWGN** | Additive White Gaussian Noise |
| **BT** | British Telecom |
| **CFO** | Carrier Frequency Offset |
| **CLT** | Central Limit Theorem |
| **CMA** | Constant Modulus Algorithm |
| **CNA** | Carrier Nulling Algorithm |
| **C-OFDM** | Coded-OFDM |
| **CP** | Cyclic Prefix |
| **CP-OFDM** | Cyclic Prefix OFDM |
| **CSA** | Carrier Serving Area |
| **DAB** | Digital Video Broadcast |
| **dB** | decibel |
| **DMT** | Discrete Multitone |

| | |
|---|---|
| **DPUSAAM** | Deterministic Partial Update Sum-Absolute Autocorrelation Minimization |
| **DS-CDMA** | Direct Sequence Code Division Multiple Access |
| **DSL** | Digital Subscriber Line |
| **DT** | Deutsche Telecom |
| **DWT** | Discrete Wavelet Transform |
| **FEQ** | Frequency domain Equalizer |
| **FFT** | Fast Fourier Transform |
| **FIR** | Finite Impulse Response |
| **FLOMs** | Fractional Lower Order Moments |
| **G-SNR** | Geometric-SNR |
| **ICI** | Inter Carrier Interference |
| **IFFT** | Inverse Fast Fourier Transform |
| **i.i.d.** | Independent Identically Distributed |
| **ISI** | Inter Symbol Interference |
| **LHSAM** | Lag-hopping Sum-squared Autocorrelation Minimization |
| **LMS** | Least Mean Square |
| **MA** | Moving Average |
| **Max-NLMS** | Maximum NLMS |

| | |
|---|---|
| **MCM** | Multicarrier Modulation |
| **MD** | Minimum Dispersion |
| **MERRY** | Multicarrier Equalization by Restoration of Redundancy |
| **MFB** | Matched Filter Bound |
| **MIMO** | Multiple-Input Multiple-Output |
| **MLSE** | Maximum Likelihood Sequence Estimation |
| **M-Max-NLMS** | Magnitude Max-NLMS |
| **MSE** | Mean Square Error |
| **MSSNR** | Maximum SSNR |
| **NLMS** | Normalized LMS |
| **OFDM** | Orthogonal Frequency Division Multiplexing |
| **pdf** | Probability Density Function |
| **PUSAM** | Partial Update SAM |
| **PUSLAM** | Partial Update SLAM |
| **PU-NLMS** | Partial Update NLMS |
| **QAM** | Quadrature Amplitude Modulation |
| **RF** | Radio Frequency |
| **RPUSAAM** | Random Partial Update Sum-Absolute Autocorrelation Minimization |

| | |
|---|---|
| **RPUSAM** | Random Partial update SAM |
| **SAM** | Sum-squared Autocorrelation Minimization |
| **SAAM** | Sum-Absolute Autocorrelation Minimization |
| **SCCP** | Single Carrier Cyclic Prefix |
| **selB-NLMS** | Selective Block NLMS |
| **seqB-NLMS** | Sequential Block NLMS |
| **SIMO** | Single-Input Multiple-Output |
| **SISO** | Single-Input Single-Output |
| **SIR** | Signal to Interference Ratio |
| **SLAM** | Single Lag Autocorrelation Minimization |
| **SLMS** | Sequential LMS |
| **SMART** | Set-Membership Recursion Techniques |
| **SM-PU-NLMS** | Set-Membership Partial Update NLMS |
| **SNR** | Signal to Noise Ratio |
| **SPU-LMS** | Stochastic Partial Update LMS |
| **SPU-NLMS** | Selective Partial Update NLMS |
| **SSNR** | Shortening SNR |
| **TEQ** | Time domain Equalizer |
| **TZ-OFDM** | Transmitter-Zero OFDM |

# LIST OF SYMBOLS

| | |
|---|---|
| $x$ | Scalar quantity |
| $\mathbf{x}$ | Vector quantity |
| $\mathbf{X}$ | Matrix quantity |
| $(.)^T$ | Transpose operator |
| $(.)^{-1}$ | Matrix inverse |
| $tr(.)$ | Matrix trace |
| $\otimes$ | Kronecker product |
| $m \mod n$ | The remainder of the integer division of m by n |
| $\lvert\cdot\rvert$ | Absolute value |
| $\lVert\cdot\rVert$ | Euclidean Norm |
| $1$ | Vector of Ones |
| $\mathbf{I}$ | Identity matrix |
| $diag$ | Diagonal elements of a matrix |
| $*$ | Discrete time convolution |
| $O(.)$ | Of order (algorithm complexity) |

| | |
|---|---|
| $\lfloor \cdot \rfloor$ | Truncation operation |

# CONTENTS

# List of Figures

# List of Tables

Chapter 1

# INTRODUCTION

In multicarrier modulation (MCM) systems, such as asymmetrical digital subscriber line (ADSL) transceivers, each symbol consists of samples to be transmitted to the receiver plus a cyclic prefix (CP) of length $v$ [3]. The CP is the last $v$ samples of the original $N$ samples to be transmitted. The CP is inserted between blocks to combat inter-symbol interference (ISI) and inter-channel interference (ICI). The length of the CP should at least be equal to the order of the channel impulse response. At the receiver the CP is removed, and the remaining $N$ samples are then processed by the receiver. Since the efficiency of the transceiver is reduced by the introduction of the CP it is therefore desirable either to make $v$ as small as possible or to choose a large $N$. Selecting large $N$ will increase the computational complexity, system delay, and memory requirements of the transceiver. The insertion of CP is shown in Figure (1.1) for the length of the channel 4 and the actual data symbol duration of 12.

To overcome these problems a short time-domain equalizer (TEQ), usually an FIR filter, can be placed in the front end of the multicarrier receiver, as shown in Figure 1.2 to shorten the impulse response of the effective channel. The length of the shortened impulse response filter and CP are usually fixed a priori and not changed from chan-

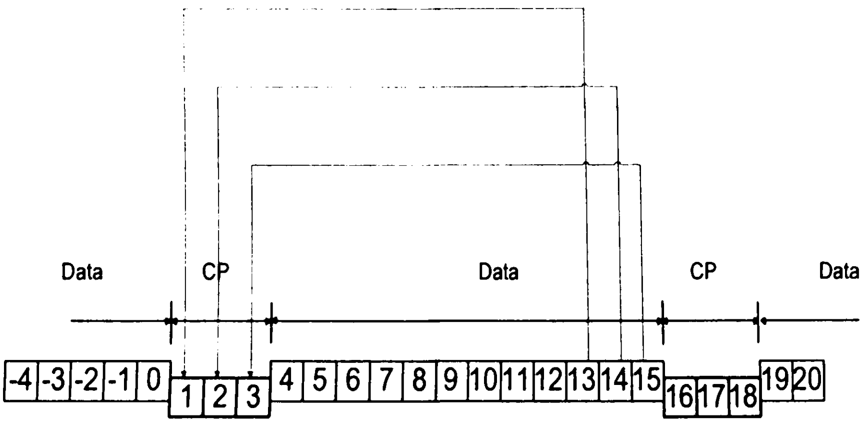

**Figure 1.1.** Insertion of a cyclic prefix in multicarrier transmission.

nel to channel. A low complexity blind adaptive algorithm to design a time-domain equalizer (TEQ), called sum-squared autocorrelation minimization (SAM) was proposed in [4] 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 [5], on the other hand, achieves channel shortening by minimizing the squared value of only a single autocorrelation at a lag greater than the CP. The drawback with SLAM is that even guaranteeing convergence of the SLAM cost to low values does not necessarily guarantee convergence to high SIRs [6]. New algorithms are therefore required with robust convergence properties and low computational complexity, and this will be the focus of this thesis.

## 1.1 Application of Channel Shortening

Channel shortening was first applied to maximum likelihood estimation (MLSE). MLSE [7] is the optimal estimation method in terms of minimizing the error probability of a sequence. Since its complexity grows

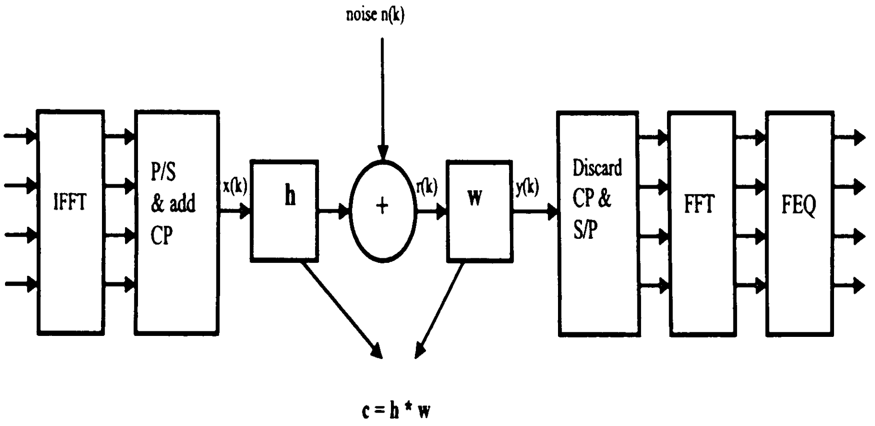

**Figure 1.2.** Multicarrier baseband system model. (I)FFT: (inverse) fast Fourier transform, P/S: parallel to serial, S/P: serial to parallel, CP: cyclic prefix, **h**: (FIR) channel of length $(L_h + 1)$, **w**: TEQ (Time domain Equalizer) of length $(L_w + 1)$, FEQ: Frequency domain Equalizer

exponentially with the channel length, a prefilter can be used to shorten the transmission channel and reduce the complexity and then applying the MLSE to the output of the shortened effective channel [8], [9]. To minimize the MSE between the target and the convolution of the channel and prefilter, one approach is to design both the prefilter and the shortened target impulse response [10], [11]. Use a decision feedback equalizer (DFE) to shorten the channel, and then apply the MLSE is another approach [12], [13]. Channel shortening has also been proposed for use in multiuser detection [14] in direct sequence code division multiple access (DS-CDMA) systems. The complexity of the MLSE grows exponentially with the number of users. "Channel shortening" can be implemented to suppress L-K of the scalar channels (channels as in SISO case) and retain the other K channels, effectively reducing the number of users from L to K. Then the MLSE can be implemented to recover the signals of the remaining K users. In this context, "channel shortening" means reducing the number of scalar channels rather than reducing the number of channel taps, and the mathematical structure is similar to channel shortening for MLSE applications [15]. Channel shortening can be used to reduce the complexity of ultra wideband systems [16]. Yet another application is in acoustics. Psychoacoustics defines the D50-measure for intelligibility of speech as the ratio of energy in a 50 ms window of the room impulse response to the total energy of the impulse response, and optimization of this measure can be performed by a channel shortener [17].

Channel shortening has found its revival and its main use is in multicarrier communication systems [18]. Examples of multicarrier communication systems include wireless local area networks (IEEE 802.11 a/g, HIPERLAN/2) [19], wireless metropolitan area networks (IEEE 802.16) [20], Digital Audio Broadcast (DAB) [21] and Digital Video Broadcast (DVB) [22] in Europe, satellite radio (Sirius and XM Radio) [23], and the proposed standard for multiband ultra wideband (IEEE 802.15.3a). Examples of wireline multicarrier systems include power line communications (HomePlug) [24] and digital subscriber lines (DSL) [25].

## 1.2 The Structure of the thesis

The remainder of the thesis is organized as follows. Chapter 2 presents a literature survey of previous work in partial update adaptive filtering techniques.

Chapter 3 studies the algorithms which attempt to restore each of the properties of the transmitted sequence that ought to be present in the equalized received sequence.

Chapter 4 proposes novel blind adaptive channel shortening algorithms, the deterministic partial update sum-absolute autocorrelation minimization (DPUSAAM) algorithm and the random partial update sum-absolute autocorrelation minimization (RPUSAAM) algorithm for multicarrier modulation systems. These algorithms are based on updating only a portion of the coefficients of the channel shortening filter at each time sample instead of the entire set of coefficients. This work is the first attempt in the field of using partial update filtering in blind adaptive channel shortening. The algorithms are also designed to be robust to impulsive noise impairment found in ADSL channels. These algorithms have low computational complexity whilst retaining essentially identical performance to the sum-absolute autocorrelation minimization (SAAM) algorithm [26]. The non-Gaussian impulsive noise has been modeled as Gaussian-mixture and as $\alpha$-stable distributions.

Chapter 5 addresses the complexity reduction and convergence issues with the SAM algorithm [4] and the SLAM algorithm [5]. The partial update method is applied to the two channel shortening algorithms which achieve the same performance whilst further reducing the computational complexity, the proposed algorithms are called the partial update SAM algorithm (PUSAM) and partial update SLAM algorithm (PUSLAM). These algorithms essentially achieve the same result in terms of reducing the effective channel length as SAM and SLAM with half the complexity. The performance advantage of the PUSAM and PUSLAM algorithms is shown on eight different carrier serving area test loops (CSA) channels and comparisons are made with the original SAM and the SLAM algorithms.

Chapter 6 addresses the complexity reduction in adaptive filter implementations, and improving the convergence which has been the problem associated with the deterministic partial update scheme in Chapter 5. In this chapter, the new random partial update sum-squared autocorrelation minimization (RPUSAM) algorithm is proposed. This algorithm has low computational complexity whilst achieving improved convergence performance, in terms of achievable bit rate, over the PUSAM algorithm with a deterministic coefficient update strategy as in Chapter 5. The performance advantage of the RPUSAM algorithm is shown on eight different carrier serving area test loops (CSA) channels and comparisons are made with the original SAM and the PUSAM algorithms. Also in this chapter a new partial update blind channel shortening algorithm is proposed. The proposed algorithm essentially achieves the same result in terms of reducing the effective channel length as SLAM. Importantly, however, the disadvantage of SLAM in terms of the SIR performance has been overcome by the proposed algorithm where the proposed algorithm has the advantage of low complexity of SLAM over SAM and also has the advantage of SAM where a low lag-hopping sum-squared autocorrelation minimization (LHSAM) cost will be identical to a low SAM cost which guarantees to give a high SIR at the output of the matched filter as on the average the proposed algorithm uses all the lags as in SAM.

Chapter 7 concludes the thesis and highlights possible areas for further research.

## Chapter 2

# LITERATURE REVIEW

### 2.1 Introduction

In this chapter, previous work in partial update adaptive filtering techniques will be reviewed. The first technique is to update one coefficient at each iteration which is called the maximum normalized least mean square (Max-NLMS) algorithm, this adaptive filter only adjusts the coefficient associated with the data element that has maximum absolute value in the filter memory at each iteration [27]. The second technique is to update a portion of the coefficients at each iteration, and those coefficients are the ones which have the largest magnitude gradient components on the error surface. Coefficients which have a small magnitude gradient component do not need to be updated as they will have little effect on the overall algorithm performance [28]. The third technique is to update entire blocks of the coefficients instead of selecting single filter coefficients for updating, thereby reducing the costs in terms of memory without losing the convergence speed. Another technique will also be studied, based on dividing the adaptive filter coefficients into small blocks and then updating a number of those blocks rather than the entire filter at every iteration, this will be achieved by using a selection criterion, which ranks the regressor vector blocks according to their squared Euclidean norms (their energy) and selects those blocks with the largest norms as the ones to be updated. Combining the data-selective updating from set-membership filtering with the reduced computational complexity from partial updating will also be studied. A new algorithm called the stochastic partial update LMS algorithm (SPU-LMS) will also be studied based on choosing which of the subset of the filter coefficients to update randomly, the motivation for which is to overcome possible convergence problems in previous schemes.

## 2.2 Overview

In [27], the author implements the maximum normalized least mean square (Max-NLMS) algorithm; it is based on updating one coefficient at every iteration. This adaptive filter only adjusts the coefficient associated with the data element that has maximum absolute value in the filter memory at each iteration. The update equation for the algorithm is given by

$$w_i(k+1) = \begin{cases} w_i(k) + \mu \frac{e(k)}{x(k-i+1)}, & \text{if } |x(k-i+1)| = \|\mathbf{x}(k)\|_\infty \\ w_i(k), & \text{otherwise} \end{cases} \tag{2.2.1}$$

where $i = 1, ...., L$, and $L$ is the length of the adaptive filter, and $k$ denotes the discrete time index. $\|\mathbf{x}(k)\|_\infty = max_{1 \leq j \leq L} \ |x(k-i+1)|$

This update is extremely simple, requiring only a single multiply, divide and add at each iteration if the maximum absolute value of the input data samples currently in the filter memory is known.

The author also introduces a new algorithm called maxlist, this algorithm is a computationally simple method requiring only $O(\log L)$ memory elements for calculating the value and position of the running maximum across a sliding data window. The algorithm exploits the shifting nature of the window, so it calculates the maximum value of the stored elements and compares it with the new element which enters the input vector at the most recent time.

Within the paper the mean square analysis of Max-NLMS is presented, the steady-state excess mean-square error (MSE) of the filter is

$$\xi_{MSE,ss} = \lim_{k \to \infty} tr\left[RE\{\mathbf{v}(k)\mathbf{v}^T(k)\}\right] \tag{2.2.2}$$

where $R = E\{\mathbf{x}(k)\mathbf{x}^T(k)\}$, $E\{\mathbf{x}(k)\} = \mathbf{0}$, $\mathbf{v}(k) = \mathbf{w}(k) - \mathbf{w}_{opt}$ , and $\mathbf{w}_{opt}$ is the optimal Wiener solution and $(.)^T$ denotes vector transpose. In [27], this expression is simplified to

$$\xi_{MSE,ss} = \sigma_x^2 L(\sigma_{v,ss}^2 + (L-1)\rho_x r_{v,ss}) \tag{2.2.3}$$

where $\sigma_x^2$ is the adaptive filter input power, $\sigma_{v,ss}^2$ is the steady-state average coefficient error power $E\{v_i^2(k)\}$, $r_{v,ss}$ is the steady-state average coefficient error cross-correlation, $E\{v_i(k)v_j(k)\}$ $i = 1, 2, ...., L$ and $j \neq i$ , and $\rho_x$ is the inter lag coefficient of the assumed correlated zero-mean Gaussian distributed input sequence, where the subscript ss denotes steady-state value.

The author also derives bounds on $\mu$ to ensure convergence of the algorithm in the mean square sense

$$0 < \mu < \frac{2 + 2g_{101}\{L - 2 - (L-1)g_{101}\}}{1 + (L-1)g_{202} + g_{101}\{L - 2 - 2(L-1)g_{101} + (L-1)(L-2)(g_{202} - g_{112})\}} \tag{2.2.4}$$

where $g_{mnp} = E\left\{\frac{x_i^m(k)x_j^n(k)}{x_{max}^p(k)}\right\}$, for $i \neq j \neq \max \neq i$.

Note that $g_{mnp}$ does not depend on the particular values of $i$, $j$, and $max$ because of the chosen input signal distributions. Note, $max$ is the index of the sample in the input vector $\mathbf{x}(k)$ which maximizes $\|x(k)\|_\infty$.

Furthermore, since $g_{101}$ is typically very small, the bounds are approximately given by

$$0 < \mu << \frac{2}{1 + (L-1)g_{202}} \tag{2.2.5}$$

And, finally, since $0 \leq g_{202} \leq 1$, the conservative bounds on the step size become

$$0 < \mu < \frac{2}{L} \tag{2.2.6}$$

which is of an identical form to that of the conventional LMS algorithm. The author compares by simulation the performance of the Max-NLMS adaptive filter to that of the LMS, sequential LMS [28], and periodic LMS [28] adaptive filters. It is shown that in terms of convergence of the coefficient error powers $trE\left\{\mathrm{v}(k)\mathrm{v}^T(k)\right\}$, where $tr(.)$ denotes matrix trace, for the four adaptive filters in a system identification task with a target filter with $L=$ 30 unity-valued FIR filter coefficients, the convergence of the Max-NLMS adaptive filter is faster than that of the periodic and sequential LMS adaptive filters. But the LMS adaptive filter outperforms the other adaptive filters; however, its complexity is approximately twice that of the other adaptive filters [27].

In [28], the author explores algorithms (the periodic LMS algorithm and the sequential LMS algorithm) for updating the coefficients of an adaptive filter by updating a portion of the coefficients at each time rather than a single coefficient as in Max-NLMS; the author compares these algorithms with the conventional LMS adaptive filter algorithm.

In the periodic LMS algorithm, one coefficient is updated at each iteration and the error is calculated once in every $L$ iterations, so the complexity is reduced, but it converges slower than LMS as is confirmed in the paper by simulation study.

The author assumes a standard FIR configuration for the adaptive filter, in which the regressor signal is the input signal.

The update equations for the periodic algorithm are given by:

$$w_i(k+1) = \begin{cases} w_i(k) + \mu e(l)x(l-i+1), & \\ & \text{if } (k+i)\text{mod}N = 0 \text{ and } l{=}N\lfloor k/N \rfloor \\ w_i(k), & \text{otherwise} \end{cases} \tag{2.2.7}$$

$$e(k) = d(k) - \mathbf{w}^T(k)\mathbf{x}(k) \tag{2.2.8}$$

where $i = 1, ...., L$, and $L$ is the length of the adaptive filter, $\mathbf{w}(k) = [w_1(k), w_2(k) .............w_L(k)]^T$ is the coefficient vector of the adaptive filter at time $k$, $\mathbf{x}(k) = [x(k), x(k-1) ..............x(k-L+1)]^T$ is the input signal vector, $d(k)$ is the desired response signal, $e(k)$ is the error signal and $\lfloor . \rfloor$ denotes the truncation operation. For $N = 1$ this algorithm reduces to the LMS algorithm and when $N = L$ it reduces to the partial update LMS adaptive algorithm.

For $N > 1$, the number of multiplies and coefficient memory accesses required for this algorithm are fewer than those required for the LMS algorithm [28].

Within the paper, [28], the evolution equation for the mean of the outer product of the coefficient error vector is given by:

$$\begin{aligned} E\{\mathrm{v}(k+N)\mathrm{v}^T(k+N)\} &= E\{\mathrm{v}(k)\mathrm{v}^T(k)\} - \mu(RE\{\mathrm{v}(k)\mathrm{v}^T(k)\} + E\{\mathrm{v}(k)\mathrm{v}^T(k)\}R) \\ &\quad +\mu^2\sigma_n^2 R + \mu^2(2RE\{\mathrm{v}(k)\mathrm{v}^T(k)\}R + Rtr[RE\{\mathrm{v}(k)\mathrm{v}^T(k)\}]) \end{aligned} \tag{2.2.9}$$

The author determines a simple expression for the steady-state value of the excess mean-square-error (MSE) by neglecting the last term, because it is much smaller than the other terms in the equation for small values of $\mu$. The resulting expression is

$$\lim_{k\to\infty} E\{\mathrm{v}^T(k)\mathrm{x}(k)^2\} = \frac{\mu\sigma_n^2 trR}{2} \tag{2.2.10}$$

which shows the dependence on adaptation gain, noise variance and tap input power.

The author also derives bounds on $\mu$ to ensure convergence of the algorithm in the mean square.

$$0 < \mu < \frac{2}{3trR} \tag{2.2.11}$$

And for independent identically distributed (i.i.d.) input signals, the evolution equation for the trace of the coefficient error correlation matrix is given by

$$trE\{\mathrm{v}(k+N)\mathrm{v}^T(k+N)\} = (1-2\mu\sigma_x^2+\mu^2(L-1)\sigma_x^4+\eta))trE\{\mathrm{v}(k)\mathrm{v}^T(k)\}+\mu^2\sigma_n^2\sigma_x^2L \tag{2.2.12}$$

where $E\{x^2(k)\} = \sigma_x^2$ and $E\{x^4(k)\} = \eta$. The steady-state excess MSE for i.i.d. input signals is

$$\lim_{k\to\infty} E\{(\mathrm{v}^T(k)\mathrm{x}(k))^2\} = \frac{\mu\sigma_n^2\sigma_x^4L}{2\sigma_x^2 - \mu((L-1)\sigma_x^2+\eta)} \tag{2.2.13}$$

which also shows the dependence on the input variance, tap filter length and noise variance. Moreover, in [28], the sequential LMS algorithm's performance was analysed. In the sequential LMS algorithm, one coefficient is updated at each time; the error is calculated for every iteration, and it is shown that its complexity is higher than that of the periodic LMS algorithm, but less than for LMS.

The update equations for the sequential LMS algorithm are given by:

$$w_i(k+1) = \begin{cases} w_i(k) + \mu e(k)x(k-i+1), & \\ & \text{if } (k-i+1) \bmod N = 0 \\ w_i(k), & \text{otherwise} \end{cases} \tag{2.2.14}$$

where $i = 1, ...., L$, and $L$ is the length of the adaptive filter.

For $N = 1$, this algorithm reduces to the LMS algorithm.

Within the paper, the author uses two types of analyses:

1- Analysis using the independence assumption as in the periodic LMS algorithm, the author expresses the algorithm again using the definition of $\mathrm{v}(k) = \mathbf{w}(k) - \mathbf{w}_{opt}$, the elements of which are given by

$$v_i(k+1) = \begin{cases} v_i(k) - \mu x(k-i+1)\mathbf{x}^T(k)\ \mathbf{v}(k) + \mu n(k)x(k-i+1), \\ \qquad\qquad \text{if } (k-i+1) \bmod N = 0 \\ v_i(k), \qquad\qquad \text{otherwise} \end{cases} \tag{2.2.15}$$

where $i = 1, \ldots., L$, and $L$ is the length of the adaptive filter.

Considering $N$ iterations of this algorithm, the coefficient error vector update is

$$\mathbf{v}(k+l) = A(k)\mathbf{v}(k) + \mathbf{b}(k) \tag{2.2.16}$$

where the elements of the $N \times N$ matrix $A(k)$ and vector $\mathbf{b}(k)$ depend only on the elements of the input and noise signals [28].

Within the paper, the vector update equation for the mean coefficient error vector is given by

$$E\{\mathbf{v}(k+N)\} = E\{A(k)\}E\{\mathbf{v}(k)\} + E\{\mathbf{b}(k)\} \tag{2.2.17}$$

as well as the coefficient error correlation matrix, given by

$$E\{\mathbf{v}(k+N)\mathbf{v}^T(k+N)\} = E\{A(k)E\{\mathbf{v}(k)\mathbf{v}^T(k)\}A^T(k)\} + E\{\mathbf{b}(k)\mathbf{b}^T(k)\} \tag{2.2.18}$$

for input signals which define the signal elements of $\mathbf{b}(k)$, that are generated from a model of the form

$$x(k) = \mathbf{a}^T\mathbf{u}(k) \tag{2.2.19}$$

where $\mathbf{a} = [a_0, a_1, ........, a_{M-1}]^T$ defines the correlation statistics of the input signal and $\mathbf{u}(k) = [u(k), u(k-1), ........., u(k-M+1)]^T$ , where $u(k)$ is a zero-mean unity variance i.i.d. signal.

2- Approximate analysis for small step sizes, in which the author rewrites the update equations for the sequential algorithm (2.2.14) as

$$w_i(k+1) = \begin{cases} w_i(k) + \mu e_{m,l} x(k-i+1) + O(\mu^2), & \\ \qquad \text{if } (k-i+1) \bmod N = 0, l = N\lfloor k/N \rfloor & \\ \qquad m = k \bmod N & \\ w_i(k), & \text{otherwise} \end{cases} \tag{2.2.20}$$

where $i = 1, ...., L$, and

$$e_j(k) = d(k+j) - \mathbf{w}^T(k)\mathbf{x}(k+j) \tag{2.2.21}$$

where $O(\mu^2)$ represents terms that are of order $\mu^2$ and higher. For small step sizes these terms can be ignored. The author derives the update equation by collecting $N$ updates for the equation given by

$$\mathbf{w}(k+N) = \mathbf{w}(k) + \mu\tilde{\mathbf{x}}(k) \otimes \mathbf{e}(k) \tag{2.2.22}$$

where $\mathbf{e}(k) = [e_0(k), e_1(k), ........., e_{(L/N)-1}(k)]^T$ is an $(L/N)$- dimensional vector of errors, where the author assumed throughout the paper that $L/N$ is an integer, $\tilde{\mathbf{x}}(k) = [x(k), x(k-N), ........., x(k-L+N)]^T$ is an $N$-dimensional decimated version of the regressor vector, and $\otimes$ denotes the Kronecker product.

Within the paper, the update for the coefficient error vector is given by:

$$\mathbf{v}(k+N) = (I_L - \mu\tilde{\mathbf{x}}(k) \otimes X^T(k))\mathbf{v}(k) + \mu\tilde{\mathbf{x}}(k) \otimes \mathbf{n}(k) \qquad (2.2.23)$$

where $X(k)$ is assumed to be $L \times L/N$ matrix and is defined as $X(k) = [\mathbf{x}(k)\mathbf{x}(k+1)....\mathbf{x}(k+(L/N)-1)]$, $\mathbf{n}(k) = [n(k),........,n(k+(L/N)-1]^T$, and $I_L$ is the $L \times L$ identity matrix.

The author takes expectations on both sides of the above equation to yield.

$$E\{\mathbf{v}(k+N)\} = (I_L - \mu E\{\tilde{\mathbf{x}}(k) \otimes x^T(k)\})E\{\mathbf{v}(k)\} \qquad (2.2.24)$$

and

$$E\{\tilde{\mathbf{x}}(k) \otimes X^T(k)\} = R \qquad (2.2.25)$$

where $R$ is again the input signal autocorrelation matrix. So (2.2.24) becomes:

$$E\{\mathbf{v}(k+N)\} = (I_L - \mu R)E\{\mathbf{v}(k)\} \qquad (2.2.26)$$

This equation is identical to that for the periodic LMS algorithm.

Then the author examines the mean-square behaviour of the sequential LMS algorithm for small step sizes. He assumes that the input signal is zero mean and either Gaussian-distributed or i.i.d. distributed with a known probability density. The update equation for

$E\{\mathbf{v}(k)\mathbf{v}^T(k)\}$ is given by

$$
\begin{aligned}
E\{\mathbf{v}(k{+}N)\mathbf{v}^T(k{+}N)\} &= E\{\mathbf{v}(k)\mathbf{v}^T(k)\} - \mu(RE\{\mathbf{v}(k)\mathbf{v}^T(k)\} + E\{\mathbf{v}(k)\mathbf{v}^T(k)\}R) \\
&\quad + \mu^2\sigma_n^2 \tilde{R} \otimes I_{(L/N)} \\
&\quad + \mu^2(2RE\{\mathbf{v}(k)\mathbf{v}^T(k)\}R + \tilde{R} \otimes F(E\{\mathbf{v}(k)\mathbf{v}^T(k)\}) \qquad (2.2.27)
\end{aligned}
$$

where $\tilde{R}$ is an $N \times N$-dimensional matrix whose $i$, $j$th value is defined by

$$[\tilde{R}]_{i,j} = r((i-j)N), \qquad (2.2.28)$$

where $r(m) = E\{x(k)x(k-m)\}$, and $F\,(.)$ is an $(L/N)\times(L/N)$ matrix-valued function whose $i$, $j$th element is

$$[F(E\{\mathbf{v}(k)\mathbf{v}^T(k)\})]_{i,j} = tr[R_{i-j}E\{\mathbf{v}(k)\mathbf{v}^T(k)\}] \qquad (2.2.29)$$

with $R(m) = E\{\mathbf{x}(k)\mathbf{x}^T(k+m)\}$.

Then the author derives the steady-state excess MSE which is approximately given by

$$\lim_{k\to\infty} E\{(\mathbf{v}^T(k)\mathbf{x}(k))^2\} = tr[RE\{\mathbf{v}(k)\mathbf{v}^T(k)\}] \qquad (2.2.30)$$

$$= \frac{\mu\sigma_n^2 L tr\tilde{R}}{2N} \qquad (2.2.31)$$

$$= \frac{\mu\sigma_n^2 trR}{2} \qquad (2.2.32)$$

The excess MSE in steady-state is approximately the same as that for the LMS adaptive filter with corresponding step size.

The author derives bounds on $\mu$ to ensure convergence of the algorithm in the mean-square sense from the update equation (2.2.27)

$$0 < \mu < \frac{2}{3tr[R]} \tag{2.2.33}$$

From [28], the author shows that the overall behavior of the sequential LMS algorithm is approximately the same as that of the periodic LMS algorithm for stationary inputs. It is also shown that the convergence rates of both algorithms are approximately *1/N*$^{th}$ that of the LMS algorithm.

In [29], the authors explore the algorithm $M$-Max NLMS that updates a portion of the coefficients at each time. These coefficients are the ones with "*larger magnitude gradient components on the error surface*" [29]

In the paper, the authors show that for LMS-type algorithms, when updating all coefficients of the adaptive filter, some coefficients have a small contribution to the error, whereas other coefficients have larger error contributions. So even if the less important coefficients are not updated at a given iteration, the algorithm performance will be hardly affected.

In the proposed algorithm, $L$ denotes the total number of coefficients at each iteration, $M$ out of $L$ which are updated. Those $M$ coefficients are the ones associated with the $M$ largest magnitude gradient components on the error surface.

The $M$-Max NLMS algorithm update equation can be written as follows:

$$w_i(k+1) = \begin{cases} w_i(k) + \frac{\mu}{\mathbf{x}^T(k)\mathbf{x}(k)} e(k)x(k-i+1) & \\ \qquad \text{if } \ i \quad \text{corresponds to one of the first} & \\ \qquad M \quad \text{maxima of} \left|x(k-i+1)\right|, i = 1, \cdots, L & \\ w_i(k) \qquad\qquad \text{otherwise} & \end{cases} \tag{2.2.34}$$

In practice, a small constant may be added to the denominator in the above equation to avoid gradient amplification which the input approaches zero [30].

The authors compare the proposed algorithm with the full-update NLMS algorithm for the same $\mu$, and show that when $M = L$, the convergence speed of the proposed algorithm approaches that of the full update NLMS algorithm. In this paper, the authors also compare the proposed algorithm with the sequential NLMS algorithm [28] in terms of complexity and convergence speed, the proposed algorithm has the same complexity overhead as the sequential NLMS, but it converges closest to the performance of NLMS. Within the paper, the algorithm is analyzed in terms of its mean square performance; to perform the analysis, the authors consider the case for $M = 1$ to show that the algorithm is guaranteed to converge to the same steady-state error as the full update NLMS for the extreme case given i.i.d. stationary zero-mean input. In [29], the authors showed that the algorithm is guaranteed to converge for the worse case of $M = 1$ for i.i.d. stationary zero-mean, where $\mu$ is chosen in the stability region and that it will converge to the same steady-state error as the full update NLMS. The authors assumed that $\mathbf{x}(k)$ is a stationary zero-mean i.i.d. sequence and they defined the coefficients error vector $\mathbf{v}(k) = \mathbf{w}(k) - \mathbf{w}_{opt}$, for $M = 1$ and $L \geq 2$, the algorithm update equation (2.2.34) becomes:

$$v_i(k+1) = \begin{cases} \left(1 - \frac{\mu}{\mathbf{x}^T(k)\mathbf{x}(k)} x^2(k-i+1)\right) v_i(k) \\ -\frac{\mu}{\mathbf{x}^T(k)\mathbf{x}(k)} \sum_{j=1,j\neq i}^{L} x(k-i+1)x(k-j+1)v_j(k) \\ +\frac{\mu}{\mathbf{x}^T(k)\mathbf{x}(k)} x(k-i+1)e^*(k) \\ \text{if } \ i \ \text{ corresponds to the maximum of} \, |x(k-i+1)| \, , i = 1, \cdots, L \\ v_i(k) \qquad\qquad \text{otherwise} \end{cases} \tag{2.2.35}$$

where $d(k) = \mathbf{x}^T(k)\mathbf{w}_{opt} + e^*(k)$, $\mathbf{w}_{opt}$ is the optimal weight vector, and $e^*(k)$ is a zero mean independent disturbance signal.

The authors assume that for high order adaptive filters $\mathbf{x}^T(k)\mathbf{x}(k) \approx L\sigma_x^2$ and from the mean error weight vector, the autocorrelation matrix governing the evolution of the mean error weight vector is $R = \frac{\sigma_x^2}{L} I$, where $I$ is the $L \times L$ identity matrix. Convergence of the proposed algorithm ($M = 1$) in the mean is therefore verified with a proper choice of the step size [29].

To derive bounds on $\mu$ to ensure full convergence of the algorithm, the authors consider the mean square error analysis of the proposed algorithm with the assumption they previously made on the input signal. Let $max$ be the index of the coefficient to be updated at time $k$, (the term $max$ ia again being used as the index of the element of the input vector $\mathbf{x}(k)$ which maximizes $\|\mathbf{x}(k)\|_\infty$) and $w_{max}(k)$ be the coefficient to be updated. The difference equation of the mean square of the $max^{th}$ coefficient for a zero mean i.i.d. input signal, can be shown from (2.2.35) to be

$$E\{v_{max}^2(k+1)\} = (1-2\widetilde{\mu}\sigma_x^2+\widetilde{\mu}^2\eta)E\{v_{max}^2(k)\}+\widetilde{\mu}^2\sigma_x^4 \sum_{j=1,j\neq max}^{L} E\{v_j^2(k)\}+\widetilde{\mu}^2\sigma_x^2\varepsilon_{min} \quad (2.2.36)$$

where $\eta = E\{x^4(k)\}$ , $\varepsilon_{min} = E\{e^{*2}(k)\}$ , and $\widetilde{\mu} = \frac{\mu}{L\sigma_x^2}$, and for a zero mean independent Gaussian input signal, $\eta = 3\sigma_x^4$. The authors assume that the sequence of updates of the coefficients is a Markov process with a uniform probability of selecting any coefficient for updating. Therefore, they have $E\{v_{max}^2(k)\} = E\{v_j^2(k)\} = c(k)$, $\forall j = 1, 2, \cdots, L$. The probability of updating any coefficient at each sample time is $\frac{1}{L}$, therefore

$$c(k+1) = \frac{1}{L}((L-1)E\{v_j^2(k+1)\} + E\{v_{\max}^2(k+1)\})j \neq \max \quad (2.2.37)$$

For $\forall j \neq \max$ , $E\{v_j^2(k+1)\} = E\{v_j^2(k)\} = c(k)$ , by substituting (2.2.36) in (2.2.37) results in

$$c(k+1) = \left(1 - 2\frac{\tilde{\mu}}{L}\sigma_x^2 + \frac{\tilde{\mu}^2}{L}[\eta + (L-1)\sigma_x^4]\right) c(k) + \frac{\tilde{\mu}^2}{L}\sigma_x^2\varepsilon_{\min} \quad (2.2.38)$$

To ensure the convergence of the algorithm in the mean square, the step size $\mu$ should be bounded by

$$0 < \mu < \frac{2L\sigma_x^4}{\eta + (L-1)\sigma_x^4} \quad (2.2.39)$$

By using (2.2.38), the authors also derive the steady-state excess MSE $\xi_{ex}(\infty)$ of the algorithm which is given by:

$$\xi_{ex,M=1}(\infty) = \frac{\mu\xi_{\min}}{2 - \frac{\mu}{L\sigma_x^4}(\eta + (L-1)\sigma_x^4)} \tag{2.2.40}$$

Within the paper, the authors show that the case of $M = 1$ and full update NLMS provide similar misadjustment when applied under the same condition and an equivalent step size is used. For related work on this topic see also [31], [32] and [33].

In [34], the authors explore algorithm selB-NLMS (selective block NLMS), which tries to combine the advantages of the selC-NLMS algorithm and seqB-NLMS. The idea is to update entire blocks of coefficients instead of selecting single filter coefficients for updating, thereby reducing the costs in terms of memory without losing the convergence speed.

In the proposed algorithm, $L$ corresponds to the total length of the filter vector and $M$ the number of filter taps to be updated at each iteration. The author assumes for simplicity that $L$ / $M$ is an integer.

The author partitions the coefficient vector $\mathbf{w}(k)$ and the excitation vector $\mathbf{x}(k)$ of the adaptive filter into $B_c$ subdivisions each of length $B_l$:

$$\begin{aligned}\mathbf{w}(k) &= [w_0(k), w_1(k), \cdots, w_{L-1}(k)]^T \\ &= [\mathbf{w}_0^T(k), \mathbf{w}_1^T(k), \cdots, \mathbf{w}_{B_c-1}^T(k)]^T\end{aligned} \tag{2.2.41}$$

$$\begin{aligned}\mathbf{x}(k) &= [x(k), x(k-1), \cdots, x(k-L+1)]^T \\ &= [\mathbf{x}_0^T(k), \mathbf{x}_1^T(k), \cdots, \mathbf{x}_{B_c-1}^T(k)]^T\end{aligned} \tag{2.2.42}$$

with

$$\mathbf{w}_i(k) = [w_{iB_l}(k), \cdots, w_{(i+1)B_l-1}(k)]^T$$

and

$$\mathbf{x}_i(k) = [x(k - iB_l), \cdots, x(k - (i+1)B_l + 1)]^T$$

The algorithm divides the excitation vector and the coefficient vector into $B_c$ blocks of length $B_l = L/B_c$ (as is shown in equations (2.2.41) and (2.2.42)). Instead of looking for the $M$ largest magnitude values, it selects $M_b = M/B_l$ blocks with the largest excitation power (energy) $\mathbf{x}_i^T(k)\mathbf{x}_i(k)$ and adapts these blocks. The algorithm update equation is given by:

$$\mathbf{w}_i(k+1) = \begin{cases} \mathbf{w}_i(k) + \mu\frac{e(k)\mathbf{x}_i(k)}{\mathbf{x}^T(k)\mathbf{x}(k)} & \\ \qquad \text{if } i \text{ belongs to the first} M_b \text{ maxima of} & \\ \qquad \mathbf{x}_i^T(k)\mathbf{x}_i(k), i \in (0, B_c - 1),\ i = 1, ...., L & \\ \mathbf{w}_i(k) \qquad \text{otherwise} & \end{cases} \tag{2.2.43}$$

where $e(k) = y(k) - \mathbf{w}^T(k)\mathbf{x}(k)$.

The author shows that by combining the two algorithms, the seqB-NLMS algorithm and the selC-NLMS algorithm, the new algorithm retains the convergence speed advantage of the selC-NLMS algorithm whilst exploiting the computational advantages of the seqB-NLMS.

In [2], the authors develop adaptive filtering algorithms with reduced computational complexity, the algorithms are based on dividing the adaptive coefficients of the filter into small blocks and updating a number of these blocks rather than the entire filter at every iteration which is similar to the previous paper [29], and is achieved by using a selection criterion, which ranks the regressor vector blocks according to their squared Euclidean norms (their energy) and selects those blocks to be updated with the largest norms.

The authors give an overview of the NLMS algorithm, and then introduce the selective partial update NLMS algorithm for a single block.

In selective partial update NLMS for a single block, the authors partition the regressor vector $\mathbf{x}(k)$ and the coefficient vector $\mathbf{w}(k)$ into $B_c$ blocks of length $B_l = L/B_c$ where $B_l$ is an integer

$$\mathbf{x}(k) = [\mathbf{x}_1^T(k), \mathbf{x}_2^T(k), \cdots, \mathbf{x}_{B_c}^T(k)]^T$$

$$\mathbf{w}(k) = [\mathbf{w}_1^T(k), \mathbf{w}_2^T(k), \cdots, \mathbf{w}_{B_c}^T(k)]^T$$

and the coefficient vector blocks $\mathbf{w}_1(k), \mathbf{w}_2(k), \cdots, \mathbf{w}_{B_c}(k)$ are the candidate subsets of $\mathbf{w}(k)$ that can be updated at discrete time instant $k$.

In the paper, the authors also write the constrained minimization problem for a single block update as:

$$\min_{1\le i\le B_c} \min_{\mathbf{w}_i(k+1)} \|\mathbf{w}_i(k+1) - \mathbf{w}_i(k)\|_2^2 \qquad (2.2.44)$$

subject to $\mathbf{w}^T(k+1)\mathbf{x}(k) = d(k)$, i.e. the a posterior error is constrained to be zero. The solution is to find the block for which the coefficient update is minimal in the squared Euclidean norm $\|.\|$ sense while satisfying the constraint $\mathbf{w}^T(k+1)\mathbf{x}(k)$ should be equal to the desired response $d(k)$.

The authors consider the minimization problem for a given block when $i$ is fixed, therefore (2.2.44) reduces to

$$\min_{\mathbf{w}_i(k+1)} \|\mathbf{w}_i(k+1) - \mathbf{w}_i(k)\|_2^2 \tag{2.2.45}$$

subject to

$$\mathbf{w}^T(k+1)\mathbf{x}(k) = d(k) \tag{2.2.46}$$

The authors solve this in a similar way to how NLMS can be derived [2] by using the method of Lagrange multipliers. The cost function to be minimized is:

$$J_i(k) = \|\mathbf{w}_i(k+1) - \mathbf{w}_i(k)\|_2^2 + \lambda(d(k) - \mathbf{w}^T(k+1)\mathbf{x}(k))$$

where $\lambda$ is a scalar Lagrange multiplier. By setting $\partial J_i(k)/\partial \mathbf{w}_i(k+1) = 0$, $i = 1, ..., B_c$ and $\partial J_i(k)/\partial\lambda = 0$ , it can be shown that

$$\mathbf{w}_i(k+1) - \mathbf{w}_i(k) - \frac{\lambda}{2}\mathbf{x}_i(k) = 0 \tag{2.2.47}$$

$$d(k) - (\mathbf{w}_i^T(k+1)\mathbf{x}_i(k) + \bar{\mathbf{w}}_i^T(k+1)\bar{\mathbf{x}}_i(k)) = 0 \tag{2.2.48}$$

where $(\mathbf{w}_i^T(k+1)\mathbf{x}_i(k)+\bar{\mathbf{w}}_i^T(k+1)\bar{\mathbf{x}}_i(k)) = \mathbf{w}^T(k+1)\mathbf{x}(k)$ and $\bar{\mathbf{x}}_i(k)$ is obtained from $\mathbf{x}(k)$ by deleting $\mathbf{x}_i(k)$, and likewise $\bar{\mathbf{w}}_i(k+1)$. Then the authors derive the equation

$$\frac{\lambda}{2} = \frac{e(k)}{\|\mathbf{x}_i(k)\|_2^2} \tag{2.2.49}$$

by substituting (2.2.47) into (2.2.48), where $\bar{\mathbf{w}}_i(k+1) = \bar{\mathbf{w}}_i(k)$ is used, i.e., only $\mathbf{w}_i(k)$ is updated.

The authors then derive the selective partial update algorithm for a single block by substituting (2.2.49) into (2.2.47) and introducing a small positive stepsize $\mu$, and also by solving the fixed block update constrained minimization problem, which is given by:

$$\mathbf{w}_i(k+1) = \mathbf{w}_i(k) + \frac{\mu}{\|\mathbf{x}_i(k)\|_2^2}\mathbf{x}_i(k)e(k), \qquad (2.2.50)$$
$$i = \arg\max_{1\leq j\leq B_c} \|\mathbf{x}_j(k)\|_2^2$$

The authors then consider updating multiple blocks; they suppose that they wish to update $B$ blocks out of $B_c$ at every iteration. And let $I_B = \{i_1, i_2, \cdots, i_B\}$ denote a $B$-subset, i.e., one having cardinality $B \triangleq |I_B|$, of the set $S = \{1, 2, \cdots, B_c\}$, and let $S$ be the collection of all such $B$-subsets, i.e., $I_B \in S$. Then the authors consider the following constrained minimization problem in order to carry out the selection of bloc

$$\min_{I_B\in S}\ \min_{\mathbf{w}_{I_B}(k+1)} \|\mathbf{w}_{I_B}(k+1) - w_{I_B}(k)\|_2^2$$

$$\begin{aligned}&\text{subject to } \mathbf{w}^T(k+1)\mathbf{x}(k) = d(k)\\ &\text{where } \mathbf{w}_{I_B}(k) = [\mathbf{w}_{i_1}^T(k), \mathbf{w}_{i_2}^T(k), \cdots, \mathbf{w}_{i_B}^T(k)]^T.\end{aligned} \qquad (2.2.51)$$

subject to $\mathbf{w}^T(k+1)\mathbf{x}(k) = d(k)$

where $\mathbf{w}_{I_B}(k) = [\mathbf{w}_{i_1}^T(k), \mathbf{w}_{i_2}^T(k), \cdots, \mathbf{w}_{i_B}^T(k)]^T$.

For $B = 1$, (2.2.51) reduces to (2.2.45). In the paper, the authors solve (2.2.51) by minimizing the cost function when $I_B$ was given and fixed, i.e.

$$J_{I_B}(k) = \frac{1}{2}\|\mathbf{w}_{I_B}(k+1) - \mathbf{w}_{I_B}(k)\|_2^2 + \lambda(d(k) - \mathbf{w}^T(k+1)\mathbf{x}(k))$$

where $\lambda$ is a Lagrange multiplier. Then the authors derive the minimization of $J_{I_B}(k)$ with respect to $\mathbf{w}_{I_B}(k+1)$ and $\lambda$ by:

$$\mathbf{w}_{I_B}(k+1) = \mathbf{w}_{I_B}(k) + \frac{1}{\|\mathbf{x}_{I_B}(k)\|_2^2}\mathbf{x}_{I_B}(k)e(k) \tag{2.2.52}$$

where $\mathbf{x}_{I_B}(k) = [\mathbf{x}_{i_1}^T(k), \mathbf{x}_{i_2}^T(k), \cdots, \mathbf{x}_{i_B}^T(k)]^T$.

The authors then obtain the NLMS algorithm for the update of $B$ blocks specified by $I_B$ after the introduction of a small positive stepsize $\mu$ (relaxation parameter)

$$\mathbf{w}_{I_B}(k+1) = \mathbf{w}_{I_B}(k) + \frac{\mu}{\|\mathbf{x}_{I_B}(k)\|_2^2}\mathbf{x}_{I_B}(k)e(k) \tag{2.2.53}$$

The block selection problem can be written to determine which blocks to update, $B$ coefficient blocks with the minimum squared-Euclidean-norm update need to be found.

$$\begin{aligned} I_B &= \arg\min_{J_B \in S} \|\mathbf{w}_{J_B}(k+1) - \mathbf{w}_{J_B}(k)\|_2^2 \\ &= \arg\min_{J_B \in S} \left\| \frac{\mathbf{x}_{J_B}(k)e(k)}{\|\mathbf{x}_{J_B}(k)\|_2^2} \right\|_2^2 \\ &= \arg\max_{J_B \in S} \|\mathbf{x}_{J_B}(k)\|_2^2 \\ &= \arg\max_{J_B \in S} \textstyle\sum_{j \in J_B} \|\mathbf{x}_j(k)\|_2^2 \end{aligned} \tag{2.2.54}$$

Then the authors found that the optimum $I_B$ to satisfy (2.2.54) is obtained by ranking the regressor vector blocks according to their squared Euclidean norms and choosing the $B$ largest blocks and that is the identical strategy suggested in [34].

The authors derive the selective partial-update NLMS (SPU-NLMS) algorithm as:

$$\mathbf{w}_{I_B}(k+1) = \mathbf{w}_{I_B}(k) + \frac{\mu}{\|\mathbf{x}_{I_B}(k)\|_2^2}\mathbf{x}_{I_B}(k)e(k)$$
$$I_B = \{i : \|\mathbf{x}_i(k)\|_2^2 \quad \text{is one of the} \quad B \quad \text{largest among} \, \|\mathbf{x}_1(k)\|_2^2, ...., \|\mathbf{x}_M(k)\|_2^2\,\} \tag{2.2.55}$$

The paper shows that for the SPU-NLMS algorithm, only one third of the filter coefficients are updated per iteration as in periodic NLMS algorithm.

The authors compare the proposed algorithm with the NLMS and the Periodic-NLMS algorithms in term of convergence performance, they show that when the block has the smallest possible length $B_l = 1$, SPU-NLMS appears to converge almost as fast as the NLMS algorithm. Also the authors compare the proposed algorithm with NLMS in term of computational complexity. Table (2.1) shows the computational complexity comparison of the NLMS and SPU-NLMS algorithms, they show that for $B_c = L$ and $B = 1$, the SPU-NLMS algorithm in (2.2.55) reduces to

$$\mathbf{w}_i(k+1) = \mathbf{w}_i(k) + \mu\frac{e(k)}{x(k-i)} \tag{2.2.56}$$
$$i = \arg\max_{0 \le j \le L-1} |x(k-j)|$$

which is the max-NLMS algorithm [27]. For $B_c = L$ and $B = L$, the SPU-NLMS algorithm becomes identical to the NLMS algorithm [2].

In the paper, the authors analyse the stability of the algorithm, they start with the persistence of excitation condition [2], they assume $B$=1, for which (2.2.55) simplifies to (2.2.50). They rewrite equation (2.2.50) as:

| | NLMS | SPU-NLMS | |
|---|---|---|---|
| | | $B_c < L$ | $B_c = L$ |
| Multiplications | $2L+2$ | $L + BB_l + 2$ | $L+B+2$ |
| Divisions | 1 | 1 | 1 |
| Comparisons | - | $O(B_c) + B_c \log_2 B$ | $\lfloor 2\log_2 L \rfloor + 2$ |

**Table 2.1.** Computational complexity of NLMS and SPU-NLMS [2]

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu(k) A_i \mathbf{x}(k) e(k), \tag{2.2.57}$$

$$i = \arg\max_{1 \le j \le B_c} \|\mathbf{x}_j(k)\|_2^2$$

where

$$\mu(k) = \frac{\mu}{\|\mathbf{x}_i(k)\|_2^2} \tag{2.2.58}$$

and $A_i$ is an $L \times L$ diagonal matrix defined by

$$A_i = diag(0, \cdots, 0, \underbrace{1, \cdots, 1, 0}_{i^{th} block}, \cdots, 0)$$

so, by using the diagonal matrix above only the $i^{th}$ block will be updated.

The desired filter response $d(k)$ is given by:

$$d(k) = \mathbf{w}_{opt}^T \mathbf{x}(k) + n(k)$$

where $\mathbf{w}_{opt}$ is the optimal coefficient vector and $n(k)$ accounts for noise, let $\mathbf{v}(k)$ denote the coefficient error vector

$$\mathbf{v}(k) = \mathbf{w}(k) - \mathbf{w}_{opt}$$

and the recursion for the coefficient error vector is

$$\mathbf{v}(k+1) = \mathbf{v}(k) - \mu(k)A_i\mathbf{x}(k)(\mathbf{v}^T(k)\mathbf{x}(k) + n(k))$$

$$i = \arg\max_{1\leq j\leq B_c} \|\mathbf{x}_j(k)\|_2^2 \tag{2.2.59}$$

By taking the statistical expectation of both sides of (2.2.59) and using the independence assumptions, and assuming that $\mu$ is a constant, the authors obtain

$$E\{\mathbf{v}(k+1)\} = (I - \mu R)E\{\mathbf{v}(k)\} \tag{2.2.60}$$

For a wide-sense stationary $x(k)$, the autocorrelation matrix $R$ is defined by

$$R = E\left\{\frac{A_i\mathbf{x}(k)\mathbf{x}^T(k)}{\|\mathbf{x}_i(k)\|^2}\middle| i = \arg\max_{1\leq j\leq B_c} \|\mathbf{x}_j(k)\|_2^2\right\} \tag{2.2.61}$$

The authors conclude that the necessary condition for the proposed algorithm to converge is that the eigenvalues of $R$ should be positive (when the eigenvalues are not positive convergence to a global minimum can not be guaranteed) and this is referred to as the persistence of excitation condition.

In the second analysis, the authors use the mean-squared error (MSE) analysis. In this analysis the authors write the coefficient error update equation as:

$$\mathbf{v}_i(k+1) = \left(I_{B_l} - \tfrac{\mu}{\|\mathbf{x}_i(k)\|_2^2}\mathbf{x}_i(k)\mathbf{x}_i^T(k)\right)\mathbf{v}_i(k) - \tfrac{\mu}{\|\mathbf{x}_i(k)\|_2^2}\mathbf{x}_i(k)\mathbf{v}^T(k)(I - A_i)\mathbf{x}(k)$$

$$-\tfrac{\mu}{\|\mathbf{x}_i(k)\|_2^2}\mathbf{x}_i(k)n(k)$$

$$i = \underset{1 \le j \le B_c}{\arg\max} \|\mathbf{x}_j(k)\|_2^2$$

where $\mathbf{v}_i(k)$ is the $i^{th}$ block of $\mathbf{v}(k)$. Under the independence assumptions, the authors derive the MSE recursion for the update coefficient block as:

$$\begin{aligned} R_{\mathbf{v}_i}(k+1) = R_{\mathbf{v}_i}(k) &- \tfrac{\mu}{B_l\sigma_x^2}(R_{\mathbf{v}_i}(k)R_{\mathbf{x}_i} + R_{\mathbf{x}_i}R_{\mathbf{v}_i}(k)) \\ &+ \tfrac{2\mu^2}{B_l^2\sigma_x^4}R_{\mathbf{x}_i}R_{\mathbf{v}_i}(k)R_{\mathbf{x}_i} + \tfrac{\mu^2}{B_l^2\sigma_x^4}R_{\mathbf{x}_i}tr(R_{\mathbf{x}_i}R_{\mathbf{v}_i}(k)) \\ &+ \tfrac{\mu^2}{B_l^2\sigma_x^4}R_{\mathbf{x}_i}tr(R_{\mathbf{x}}(I - A_i)R_{\mathbf{v}}(k)(I - A_i)) \\ &+ \tfrac{\mu^2}{B_l^2\sigma_x^4}\sigma_n^2 R_{\mathbf{x}_i} \\ & \qquad i = \underset{1 \le j \le B_c}{\arg\max} \|\mathbf{x}_j(k)\|_2^2 \end{aligned} \qquad (2.2.62)$$

where

$$R_{\mathbf{v}_i}(k) = E\{\mathbf{v}_i(k)\mathbf{v}_i^T(k) \Big| i = \underset{1 \le j \le B_c}{\arg\max} \|\mathbf{x}_j(k)\|_2^2\}$$

$$R_{\mathbf{x}_i} = E\{\mathbf{x}_i(k)\mathbf{x}_i^T(k) \Big| i = \underset{1 \le j \le B_c}{\arg\max} \|\mathbf{x}_j(k)\|_2^2\}$$

$$\sigma_x^2 = E\{\mathbf{x}^2(k)\}$$

$$\sigma_n^2 = E\{\mathbf{n}^2(k)\}$$

In (2.2.62), the authors have approximated $\|\mathbf{x}_i(k)\|_2^2$ as $B_l\sigma_x^2$.

Then the authors consider the trace of (2.2.62) since the input signal is zero-mean i.i.d. Gaussian, which is given by:

$$trR_{\mathbf{v}_i}(k+1) = \Big(1 - \tfrac{2\mu}{B_l} + \tfrac{\mu^2}{B_l^2\sigma_x^4}(2\eta + L\sigma_x^4)\Big)\, trR_{\mathbf{v}_i}(k) + \mu^2\tfrac{\sigma_n^2}{B_l\sigma_x^2},$$

$$i = \arg\max_{1\le j\le B_c} \|\mathbf{x}_j(k)\|_2^2 \tag{2.2.63}$$

where $\eta = E\{x^4(k)\}$. Then the authors remove the conditioning on $i$, which is implicit in (2.2.63), by using the block selection probabilities $p_i = \Pr\{i = \arg\max_{1\le j\le B_c} \|\mathbf{x}_j(k)\|_2^2\}$. So for a given coefficient block $i$, (2.2.63) will apply with probability $p_i$. The probability of block $i$ not being updated is $1 - p_i$ and the MSE recursion for block $i$ will be $trR_{\mathbf{v}_i}(k+1) = trR_{\mathbf{v}_i}(k)$ with probability $1 - p_i$. The authors write the MSE recursion as:

$$trR_{\mathbf{v}_i}(k+1) = p_i\Big(\Big(1 - \tfrac{2\mu}{B_l} + \tfrac{\mu^2}{B_l^2\sigma_x^4}(2\eta + L\sigma_x^4\Big)\, trR_{\mathbf{v}_i}(k) + \mu^2\tfrac{\sigma_n^2}{B_l\sigma_x^2}\Big) + (1-p_i)trR_{\mathbf{v}_i}(k)$$

$$1 \le i \le B_c$$

For zero-mean i.i.d., Gaussian input signals, the authors obtain $p_i = 1/B_c$ for all $i$, thereby yielding

$$trR_{\mathbf{v}_i}(k+1) = \Big(1 - \tfrac{2\mu}{L} + \tfrac{\mu^2}{LB_l\sigma_x^4}(2\eta + L\sigma_x^4)\Big)\, trR_{\mathbf{v}_i}(k) + \mu^2\tfrac{\sigma_n^2}{L\sigma_x^2}$$

$$1 \le i \le B_c \tag{2.2.64}$$

To ensure the stability of the recursion (2.2.64), the step size $\mu$ should be bounded by

$$0 < \mu < \frac{2B_l \sigma_x^4}{2\eta + L\sigma_x^4} = \frac{2B_l}{6 + L} \tag{2.2.65}$$

where $\eta = 3\sigma_x^4$ for Gaussian signals. Then the authors derive the nonconservative bound for large $L = B_l B_c$, where (2.2.65) can be approximated by:

$$\mu < \frac{2}{B_c} \tag{2.2.66}$$

Then the authors extend (2.2.66) to the case of multiple blocks ($B > 1$) as:

$$\mu < \frac{2B}{B_c}$$

When $B_c = B$, the stepsize is bounded by $\mu < 2$, this is consistent with the NLMS algorithm.

In [35], the authors explore the set-membership partial-update normalized least-mean square (SM-PU-NLMS) algorithm; they combine the data-selective updating from set-membership filtering with the reduced computational complexity from partial updating.

The authors start with reviewing the partial update-NLMS (PU-NLMS) algorithm and also they provide an analysis in the mean-squared sense for the convergence of the PU-NLMS algorithm as in [2].

In the paper, in set-membership filtering; the filter $\mathbf{w}$ is designed to achieve a specified bound on the magnitude of the output error. Let $\mathrm{H}(k)$ denote the set containing all vectors $\mathbf{w}$ for which the associated output error at time instant $k$ is upper bounded in magnitude by $\gamma$, i.e.,

$$\mathrm{H}(k) = \{\mathbf{w} \in R^N : \left|d(k) - \mathbf{w}^T\mathbf{x}(k)\right| \le \gamma\}$$

where $H(k)$ is referred to as the constraint set, and its boundaries are hyperplanes. Within the paper, the authors define the exact feasibility set $\psi(k)$ to be the intersection of the constraint sets over the time instants $i = 1, \ldots, k$, i.e.,

$$\psi(k) = \bigcap_{i=1}^{k} H_i$$

The authors describe the idea of set-membership adaptive recursion techniques (SMART) as a method to adapt the coefficient vector such that it will always remain within the feasible set.

Then the authors merge the ideas of partial updating and set-membership filtering to obtain the new algorithm (set-membership partial update NLMS) algorithm. The goal is to combine the advantages of set-membership filtering (SMF) and partial updating in order to obtain an algorithm with sparse updating and low computational complexity per update. The fundamental difference between SMF and partial update adaptive filtering is that for SMF if the current adaptive filter coefficients lie within a prescribed set no update will be undertaken, whereas with partial update adaptive filtering an update is made at every iteration but only a subset of coefficients is updated.

In the paper, the authors present the algorithm derivation; their approach is to find a coefficient vector that minimizes the Euclidean distance $\|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2$ subject to the constraint $\mathbf{w}(k+1) \in H(k)$ with the additional constraint of updating only $L$ coefficients. This means if $\mathbf{w}(k) \in H(k)$, the minimum distance is zero and no update is required. However, when $\mathbf{w}(k) \notin H(k)$, the new update is obtained as the solution to the optimization problem given by:

$$\mathbf{w}(k+1) = \arg\min_{\mathbf{w}} \|\mathbf{w} - \mathbf{w}(k)\|^2 .$$

subject to:

$$d(k) - \mathbf{x}^T(k)\mathbf{w} = g(k)$$

$$\tilde{\mathrm{A}}_{I_{L(k)}}(\mathbf{w} - \mathbf{w}(k)) = 0$$

where $g(k)$ is a parameter that determines a point within the constraint set $\mathrm{H}(k)$ , or it satisfies, $g(k) \leq \gamma$, and

$$\tilde{\mathrm{A}}_{I_{L(k)}} = \mathrm{I} - \mathrm{A}_{I_{L(k)}}$$

where $\tilde{\mathrm{A}}_{I_{L(k)}}$ is a complementary matrix which contains ones and zeros, the number of ones is dependent on $L$ that gives:

$$\tilde{\mathrm{A}}_{I_{L(k)}}\mathbf{w}(k+1) = \tilde{\mathrm{A}}_{I_{L(k)}}\mathbf{w}(k)$$

which means only $L$ coefficients are updated.

The authors suggest that $g(k)$ is chosen such that the update vector belongs to the closest bounding hyperplane in $\mathrm{H}(k)$, i.e.

$$g(k) = \gamma e(k)/\left|e(k)\right|$$

The authors derive the update equation in a similar way as in [2]:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu(k)\frac{e(k)\mathrm{A}_{I_{L(k)}}\mathbf{x}(k)}{\left\|\mathrm{A}_{I_{L(k)}}\mathbf{x}(k)\right\|^2}$$

The role of the matrix $\mathrm{A}_{I_{L(k)}}$ is identical to the role of the diagonal matrix which was introduced in the previous paper [2], the update occurs where only the ones in the matrix $A_{I_{L(k)}}$ exist.

The stepsize $\mu(k)$ is data dependent and given by:

$$\mu(k) = \begin{cases} 1 - \gamma / \left| e(k) \right|, \text{when} \quad \mathbf{w}(k) \notin \mathrm{H}(k), \text{i.e., if} \left| e(k) \right| > \gamma \\ 0, \qquad\qquad\qquad \text{otherwise} \end{cases} \tag{2.2.67}$$

The authors noted that $\mu(k)$ starts with high values and reduces as the error reduces, reaching zero as the maximum allowable error is approached. The authors highlight that the index set $I_{L(k)}$ specifying the coefficients to be updated is chosen as in [2], i.e., the $L$ coefficients in the input vector $\mathbf{x}(k)$ having the largest norm.

Within the paper, the authors studied the convergence issues; they assume that the coefficient error vector at instant $k$ is defined as:

$$\mathbf{v}(k) = \mathbf{w}(k) - \mathbf{w}_{opt}$$

and the desired signal is modelled as:

$$d(k) = \mathbf{x}^T(k)\mathbf{w}_{opt}$$

and the error signal is expressed as:

$$e(k) = -\mathbf{x}^T(k)\mathbf{v}(k)$$

so that the following expression gives the update equation of the norm of the coefficient error vector:

$$\begin{aligned}\|\mathbf{v}(k+1)\|^2 &= \|\mathbf{v}(k)\|^2 - \frac{1}{\|\mathrm{A}_{I_L(k)}\mathbf{x}(k)\|^2}\mathbf{v}^T(k) \times (\mu(k)\mathrm{A}_{I_L(k)}\mathbf{x}(k)\mathbf{x}^T(k) \\ &+\mu(k)\mathbf{x}(k)\mathbf{x}^T(k)\mathrm{A}_{I_L(k)} - \mu^2(k)\mathbf{x}(k)\mathbf{x}^T(k))\mathbf{v}(k) \\ &= \|\mathbf{v}(k)\|^2 - \frac{1}{\|\mathrm{A}_{I_L(k)}\mathbf{x}(k)\|^2}\mathbf{v}^T(k) \times (2\mu(k)\mathrm{A}_{I_L(k)} - \mu^2(k)\mathrm{I})\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{v}(k)\end{aligned}$$

A reduction in the coefficient error norm will occur whenever the term

$$\mathbf{v}^T(k)(2\mu(k)\mathrm{A}_{I_L(k)} - \mu^2(k)\mathrm{I})\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{v}(k)$$

is positive. The authors suggest that although the matrix

$$(2\mu(k)\mathrm{A}_{I_L(k)} - \mu^2(k)\mathrm{I})\mathbf{x}(k)\mathbf{x}^T(k)$$

has non-negative eigenvalues, there exist time instants when the coefficient error norm may increase as a result of the partial update strategy, as shown in the paper, whenever a reduction in the coefficient error norm occurs, the $\mu(k)$ that causes the largest reduction is given by:

$$\mu(k) = \|\mathrm{A}_{I_L(k)}\mathbf{x}(k)\|^2 / \|\mathbf{x}(k)\|^2$$

and achieves the largest reduction in coefficient error norm whenever a reduction occurs.

In the paper, the authors guarantee convergence with the heuristic argument that the update, even if only for a fraction of the coefficients, will point towards the optimal solution most of the time. Also the authors guarantee convergence in the mean-square sense for the case of additive measurement noise, they state that the SM-PU-NLMS algorithm converges in the mean-squared sense for zero-mean i.i.d. input signals in the presence of zero-mean additive uncorrelated noise when

$$\left(\left\|\mathrm{A}_{I_L(k)}\mathbf{x}(k)\right\|^2\right) / \left(\|\mathbf{x}(k)\|^2\right) \geq \mu(k)$$

and continue to assign a probability of update $P_e(k) = P(|e(k)| > \gamma)$, to calculate the coefficient error norm for the SM-PU-NLMS algorithm:

$$\mathbf{v}(k{+}1) = \left[\mathrm{I} - P_e(k)\mu(k)\frac{\mathrm{A}_{I_L(k)}\mathbf{x}(k)\mathbf{x}^T(k)}{\left\|\mathrm{A}_{I_L(k)}\mathbf{x}(k)\right\|^2}\right]\mathbf{v}(k){+}P_e(k)\mu(k)\frac{n(k)\mathrm{A}_{I_L(k)}\mathbf{x}(k)}{\left\|\mathrm{A}_{I_L(k)}\mathbf{x}(k)\right\|^2}$$

Then the authors derive the excess MSE under the independence assumption and assuming the additive measurement noise to be zero mean and not correlated with the white input signals by:

$$\xi(k{+}1) = \xi(k){-}\sigma_x^2{\times}E\left\{\frac{\mu(k)P_e(k)\mathbf{v}^T(k)(\mathbf{x}(k)\mathbf{x}^T(k)\mathrm{A}_{I_L(k)} + \mathrm{A}_{I_L(k)}\mathbf{x}(k)\mathbf{x}(k))\mathbf{v}(k)}{\left\|\mathrm{A}_{I_L(k)}\mathbf{x}(k)\right\|^2}\right.$$

$$\left.-\frac{\mu^2(k)P_e^2(k)\mathbf{v}^T(k)\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{v}(k)}{\left\|\mathrm{A}_{I_L(k)}\mathbf{x}(k)\right\|^2}\right\} + E\left\{\frac{\mu^2(k)P_e^2(k)n^2(k)}{\left\|\mathrm{A}_{I_L(k)}\mathbf{x}(k)\right\|^2}\right\}$$

$$= \rho_1 - \rho_2 + \rho_3 \qquad (2.2.68)$$

Then the authors rewrite $\rho_2$ by invoking the independence assumption and assuming $N$ large such that $\|\mathbf{x}(k)\|^2$ can be considered a reasonable estimate of $(N+1)E[\mathbf{x}^2(k)]$ as:

$$\rho_2 \approx \sigma_x^2 E\left\{\frac{\mathbf{v}^T(k)P_e(k)(\mathbf{x}(k)\mathbf{x}^T(k)\mathrm{A}_{I_L(k)}+\mathrm{A}_{I_L(k)}\mathbf{x}(k)\mathbf{x}^T(k))\mathbf{v}(k)}{\|\mathbf{x}(k)\|^2}\right.$$

$$\left.-\frac{P_e^2(k)\left\|\mathrm{A}_{I_L(k)}\mathbf{x}(k)\right\|^2\mathbf{v}^T(k)\mathbf{x}(k)\mathbf{x}(k)\mathbf{v}(k)}{\|\mathbf{x}(k)\|^4}\right\}$$

$$\approx \sigma_x^2 E\left\{\frac{\mathbf{v}^T(k)P_e(k)E[\mathbf{x}(k)\mathbf{x}^T(k)\mathrm{A}_{I_L(k)}+\mathrm{A}_{I_L(k)}\mathbf{x}(k)\mathbf{x}^T(k)]\mathbf{v}(k)}{(N+1)\sigma_x^2}\right.$$

$$\left.-\frac{P_e^2(k)E\left[\left\|\mathrm{A}_{I_L(k)}\mathbf{x}(k)\right\|^2\right]\mathbf{v}^T(k)\mathbf{v}(k)}{(N+1)^2\sigma_x^2}\right\} \tag{2.2.69}$$

Then the authors try to evaluate $\rho_2$ by computing the elements of matrix $B = E[\mathbf{x}(k)\mathbf{x}^T(k)\mathrm{A}_{I_L(k)} + \mathrm{A}_{I_L(k)}\mathbf{x}(k)\mathbf{x}^T(k)]$, they assume the input samples to be i.i.d., then the off diagonals will average to zero. Since $\mathrm{A}_{I_L(k)}$ will select only the $L$ values in the input vector with the largest norm, the diagonal will be an average over only the $L$ strongest components. Then the authors choose $p_i$ to denote the probability that one of the $L$ largest components contribute to the $i^{th}$ element in the diagonal. Also they choose $\{y_i\}_{i=1}^{N+1}$ to be the elements of the input vector $\mathbf{x}(k)$ sorted in magnitude such that $y_1 \leq y_2 \leq ... \leq y_{N+1}$. Then the authors calculate the diagonal elements of $B$ for a given $L$ as follows:

$$E\left\{\mathbf{x}(k)\mathbf{x}^T(k)\mathrm{A}_{I_L(k)} + \mathrm{A}_{I_L(k)}\mathbf{x}(k)\mathbf{x}^T(k)\right\}_{i,i} = 2\sum_{i=0}^{L-1} E\left\{p_i y_{N-i+1}^2\right\}$$

$$= \frac{2}{N+1}E\left\{\left\|\mathrm{A}_{I_L(k)}\right\|^2\right\}$$

where for i.i.d. signals, $p_i = 1/(N+1)$. Then the authors derive the evaluation of $\rho_2$, by substituting $p_i$ into (2.2.69) resulting in:

$$\rho_2 \approx P_e(k)(2 - P_e(k))\frac{E\left\{\left\|\mathrm{A}_{I_L(k)}\mathbf{x}(k)\right\|^2\right\}}{(N+1)^2\sigma_x^2}\xi(k) < 2\xi(k)$$

then the authors conclude that since $\rho_3$ is independent of $\xi(k)$, (2.2.68) is always stable.

Table (2.2) shows the computational complexities per update in terms of the number of additions, multiplications, and divisions for the NLMS, SM-NLMS, PU-NLMS, and SM-PU-NLMS algorithms. The authors suggest that although the PU-NLMS and SM-PU-NLMS algorithms have a similar complexity per update, the gain of applying the SM-PU-NLMS algorithm comes through the reduced number of required updates. For time instants where no updates are required, the complexity of the SM-PU-NLMS algorithm is due to filtering.

| Algorithm | Multiplications | Additions | Divisions |
|---|---|---|---|
| NLMS | $2N$+4 | $2N$+4 | 1 |
| SM-NLMS | $2N$+4 | $2N$+5 | 2 |
| PU-NLMS | $N$+$L$+3 | $N$+$L$+3 | 1 |
| SM-PU-NLMS | $N$+$L$+3 | $N$+$L$+4 | 2 |

**Table 2.2.** Computational complexity of NLMS, SM-NLMS, PU-NLMS, and SM-PU-NLMS algorithms

The authors also include in the paper simulations for a system identification application, they show that not only can the set-membership filtering adaptation algorithms, with partial updating further reduce the computational complexity when compared with the partial update NLMS algorithm, but they can also present a faster convergence for the same level of MSE [35].

In [36], the authors implement a new algorithm called the stochastic partial update LMS algorithm (SPU-LMS), it is based on choosing which of the subset of the filter coefficients to update randomly; by doing that, the authors show that divergence in nonstationary signals can be prevented by scheduling coefficient updates at random. The algorithm involves selection of a subset of size $N \ / \ P$ coefficients out of $P$ possible subsets from a fixed partition of the $L$ coefficients in the weight vector, the authors assume that the filter length $L$ is a multiple of $P$.

The authors describe the new algorithm as similar to sequential PU-LMS, the only difference is that at a given iteration, $k$, for sequential LMS (S-LMS) one of the sets $S_i, i = 1, \cdots, P$ is chosen in a predetermined fashion, whereas for SPU-LMS, one of the sets $S_i$ is sampled at random from $\{S_1, S_2, \cdots, S_P\}$ with probability $1 \ / \ P$.

The authors derive the update equation as:

$$w_j(k+1) = \begin{cases} w_j(k) + \mu e(k) x_j(k) & if \quad j \in S_i \\ w_j(k), & \text{otherwise} \end{cases} \tag{2.2.70}$$

where $e(k) = d(k) - \mathbf{w}^T(k)\mathbf{x}(k)$.

Then the authors write the above equation in a compact form as:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu e(k) I(k) \mathbf{x}(k) \tag{2.2.71}$$

where $I(k)$ is a random matrix chosen with equal probability from $\mathrm{I}(i)$, $i = 1, \cdots, P$ (where $\mathrm{I}(i)$ is obtained by zeroing out the $j$th row of the identity matrix $I$ if $j \notin S_i$).

The authors analyse the proposed algorithm in terms of uncorrelated input and coefficient vectors, deterministic signals and correlated input and coefficient vectors.

First, for the uncorrelated input and coefficient vectors, the authors assume that the desired signal $d(k)$ satisfies the condition where $d(k) = \mathbf{w}_{opt}^T\mathbf{x}(k) + n(k)$ for the stationary signal analysis of SPU-LMS. They also assume that $\mathbf{x}(k)$ is a Gaussian random vector and that $\mathbf{x}(k)$ and $\mathbf{v}(k) = \mathbf{w}(k) - \mathbf{w}_{opt}$ are independent, and $\mathbf{I}(k)$ and $\mathbf{x}(k)$ are independent of each other. They also assume that $R = E\left\{\mathbf{x}(k)\mathbf{x}^T(k)\right\}$ is block diagonal such that $\sum_{i-1}^{P} \mathrm{I}(i)R\mathrm{I}(i) = R$.

Then the authors obtain the following update equation conditioned on a choice of $S_i$ for convergence in the mean analysis.

$$E\left\{\mathrm{v}(k+1)|S_i\right\} = (I - \mu I(k)R)E\left\{\mathrm{v}(k)|S_i\right\}$$

$$= (I - \mu\mathrm{I}(i)R)E\left\{\mathrm{v}(k)|S_i\right\}$$

then the authors average over all choices of $S_i$, they obtain the following equation by making use of the fact that the choice of $S_i$ is independent of $\mathbf{v}(k)$ and $\mathbf{x}(k)$.

$$E\left\{\mathrm{v}(k+1)\right\} = \left(I - \frac{\mu}{P}R\right)E\left\{\mathrm{v}(k)\right\} \qquad (2.2.72)$$

The authors derive bounds on $\mu$ to ensure convergence of the algorithm in the mean

$$0 < \mu < \frac{2P}{\lambda_{\max}}$$

For the convergence in the mean square analysis of SPU-LMS, the authors obtain the error variance $E\left\{e(k)e^T(k)\right\}$ under the independence assumptions as:

$$E\left\{e^2(k)\right\} = \xi_{\mathsf{min}} + tr[RE\left\{\mathbf{v}(k)\mathbf{v}^T(k)\right\}]$$

where $\xi_{\mathsf{min}}$ is the minimum attainable mean square error, and is given by:

$$\xi_{\mathsf{min}} = E\left\{d^2(k)\right\} - \mathbf{r}^T R^{-1} r$$

where $R = E\left\{\mathbf{x}(k)\mathbf{x}^T(k)\right\}$ and $r = E\left\{d(k)\mathbf{x}(k)\right\}$.

Then the authors derive the evolution equation for $tr[RE\left\{\mathbf{v}(k)\mathbf{v}^T(k)\right\}]$ conditioned on choice of $S_i$ as:

$$\begin{aligned} RE\left\{\mathbf{v}(k+1)\mathbf{v}^T(k+1)|S_i\right\} &= RE\left\{\mathbf{v}(k)\mathbf{v}^T(k)|S_i\right\} - 2\mu R\mathrm{I}(i)RE\left\{\mathbf{v}(k)\mathbf{v}^T(k)|S_i\right\} \\ &+\mu^2\mathrm{I}(i)R\mathrm{I}(i)E\left\{\mathbf{x}(k)\mathbf{x}^T(k)A(k)\mathbf{x}(k)\mathbf{x}^T(k)|S_i\right\} + \mu^2\xi_{\mathsf{min}}R\mathrm{I}(i)R\mathrm{I}(i) \end{aligned} \tag{2.2.73}$$

where $A(k) = E\left\{\mathbf{v}(k)\mathbf{v}^T(k)\right\}$.

Then the authors define $\mathbf{u}(k) = Q\mathbf{v}(k)$, where $Q$ satisfies $QRQ^T = \Lambda$. By applying the definition of ***u(k)*** to (2.2.73), the authors obtain the equation:

$$\mathbf{g}(k{+}1) = \left(I - \frac{2\mu}{P}\Lambda + \frac{\mu^2}{P}\Lambda^2 + \frac{\mu^2}{P}\Lambda^2\mathbf{1}\,\mathbf{1}^T\right)\mathbf{g}(k){+}\frac{\mu^2}{P}\xi_{\mathsf{min}}\Lambda^2\mathbf{1} \tag{2.2.74}$$

where $\mathbf{g}(k)$ is a vector of diagonal elements of $\Lambda E\left\{\mathbf{u}(k)\mathbf{u}^T(k)\right\}$ , and $\mathbf{1}$ is an $L \times 1$ vector of ones.

The authors derive bounds on $\mu$ to ensure convergence of the algorithm in the mean square as:

$$0 < \mu < \frac{2}{\lambda_{\max}}$$

$$\eta(\mu) = \sum_{i=1}^{L} \frac{\mu\lambda_i}{2 - \mu\lambda_i} < 1 \tag{2.2.75}$$

which are independent of $P$ and identical to that of LMS [36].

Then the authors introduce the summed MSE difference as

$$J = \sum_{k=0}^{\infty} [\xi_k - \xi_\infty] \tag{2.2.76}$$

which is used as a measure of the convergence rate and $M(\mu) = (\xi_\infty - \xi_{\min})/\xi_{\min}$ as a measure of misadjustment. Then the authors establish the misadjustment as:

$$M(\mu) = \frac{\eta(\mu)}{1 - \eta(\mu)} \tag{2.2.77}$$

and they suggest that is the same as that of the standard LMS. Thus, they conclude that the random update of the subset has no effect on the final excess mean square error.

Then authors show that the summed MSE difference is

$$J = Ptr\{[2\mu\Lambda - \mu^2\Lambda^2 - \mu^2\Lambda^2 \mathbf{1}\ \mathbf{1}^T]^{-1}(\mathrm{g}(0) - \mathrm{g}(\infty)\} \tag{2.2.78}$$

which is $P$ times the quantity obtained for the standard LMS algorithm [36]. They conclude that for block diagonal $R$, random updating slows down convergence by a factor of $P$ without affecting the misadjustment. Furthermore, they verify that a much simpler condition $0 < \mu < \left(\frac{1}{tr\{R\}}\right)$, provides a sufficient region for convergence of SPU-

LMS and the standard LMS algorithm.

Second, for deterministic signals, the authors assume that the input signal $\mathbf{x}(k)$ is bounded, that is $\sup_k(\mathbf{x}^T(k)\mathbf{x}(k)) \leq B < \infty$, and that the desired signal $d(k)$ follows the model

$$d(k) = \mathbf{w}_{opt}^T\mathbf{x}(k)$$

they define $\mathbf{v}(k) = \mathbf{w}(k) - \mathbf{w}_{opt}$ and $e(k) = d(k) - \mathbf{w}^T(k)\mathbf{x}(k)$, and they compose a lemma.

Lemma: if $\mu < 2/B$, then $e^2(k) \rightarrow 0$ as $k \rightarrow \infty$. Where, $\overline{\{.\}}$ indicates statistical expectation over all possible choices of $S_i$, where each $S_i$ is chosen randomly with equal probability from $\{S_1, \cdots, S_P\}$.

For a positive definite matrix $A(k)$, it is stated that $A(k)$ converges exponentially fast to zero if there exits a $\gamma, 0 < \gamma < 1$ and a positive integer $K$ such that $tr\{A(k+K)\} \leq (1-\gamma)tr\{A(k)\}$ for all $k$.

And if $\mu < 2/B$ and the signal satisfies the following persistence of excitation condition, for all $k$, there exist $K < \infty$ , $\alpha_1 > 0$ and $\alpha_2 > 0$ such that

$$\alpha_1 I < \sum_{i=k}^{k+K} \mathbf{x}_i\mathbf{x}_i^T < \alpha_2 I \tag{2.2.79}$$

then $\overline{\mathbf{v}}^T(k)\overline{\mathbf{v}}(k) \rightarrow 0$ , and $\overline{\mathbf{v}^T(k)\mathbf{v}(k)} \rightarrow 0$ exponentially fast.

The authors conclude that the conditions (2.2.79) are identical to the persistence of excitation conditions for standard LMS. Therefore, the sufficient condition for exponential stability of LMS is enough to guarantee exponential stability of SPU-LMS.

Third, for correlated input and coefficient vectors, in this section the authors compare the performance of LMS and SPU-LMS in terms of stability and misconvergence when the uncorrelated input and coefficient signal vectors assumption is invalid. Here the authors analyse the stability and the performance separately. In this section the authors make the dependence of the value of $\mu$ explicit and conclude that stability and performance of SPU-LMS are similar to that of LMS.

*Result 1 (stationary Gaussian process)*, let $x(k)$ be a stationary Gaussian random process such that $E\{x(k)x(k-l)\} = r_l \rightarrow 0$ as $l \rightarrow \infty$ and $\mathbf{x}(k) = [x(k)x(k-1)\cdots x(k-n+1)]$ , then for any $p \geq 1$ , there exist constants $\mu^* > 0$ , $M > 0$ , and $\alpha \in (0,1)$ such that for all $\mu \in (0,\mu^*)$ and for all $t \geq k \geq 0$

$$\left[E\left\{\left\|\prod_{j=k+1}^{t}(I-\mu I(j)\mathbf{x}(j)\mathbf{x}(j)^T)\right\|^p\right\}\right]^{\frac{1}{p}} \leq M(1-\mu\alpha)^{t-k}$$

if and only if the input correlation matrix $E[\mathbf{x}(k)\mathbf{x}^T(k)] = R_{xx}$, is positive definite.

They continue to conclude that a necessary and sufficient condition for convergence is that the covariance matrix be positive definite. Although first analysis gives some bounds on the step size $\mu$, the authors say that they are not very reliable as the analysis is valid only for very small $\mu$.

In the mean squared analysis, the authors assume that

$$d(k) = \mathbf{x}^T(k)\mathbf{w}_{opt} + n(k)$$

The effectiveness of the method is explained in Results 2 and 3 below, where the authors compare the steady-state performance of the two algorithms for two simple scenarios where the independence assumption is violated.

*Result 2 (i.i.d. Gaussian process)*: let $\mathbf{x}(k) = [x(k)x(k-1)........x(k-L+1)]^T$, where $L$ is the length of the vector $\mathbf{x}(k)$. $\{x(k)\}$ is a sequence of zero mean i.i.d. Gaussian random variables. And $\sigma_x^2$ denotes the variance of $x(k)$ and $\sigma_n^2$ denotes the variance of $n(k)$. Then the authors assume that $n(k)$ is a white i.i.d. Gaussian noise. For LMS, they have

$$\lim_{k\to\infty} E\left\{\mathbf{v}(k)\mathbf{v}^T(k)\right\} = \mu^2\left[\frac{\sigma_n^2}{2\mu}I + \frac{L\sigma_x^2\sigma_n^2}{4}I + C\mu^{\frac{1}{2}}I\right] \tag{2.2.80}$$

and for SPU-LMS, they assume $L$ to be a multiple of $P$ and sets $S_i$ to be mutually exclusive, they have

$$\lim_{k\to\infty} E\left\{\mathbf{v}(k)\mathbf{v}^T(k)\right\} = \mu^2\left[\frac{\sigma_n^2}{2\mu}I + \frac{\frac{(L+1)P-1}{P}\sigma_x^2\sigma_n^2}{4}I + C\mu^{\frac{1}{2}}I\right]$$

then the authors note that the constant $C$ in the final mean square expression for SPU-LMS is the same as that for LMS. Therefore, for large $L$, the authors see that SPU-LMS is marginally worse than LMS in terms of misadjustment.

Then from (2.2.74), the authors obtain the vector of diagonal elements of $\lim_{k\to\infty} E\left\{\mathbf{v}(k)\mathbf{v}^T(k)\right\}$ $\mathbf{v}_d$ to be

$$\mathbf{v}_d = \mu^2\left[\frac{\sigma_n^2}{2\mu}\mathbf{1} + \frac{(L+1)\sigma_x^2\sigma_n^2}{4}\mathbf{1}\right] + O(\mu^4)\mathbf{1}$$

where $\mathbf{1}$ is an $L \times 1$ vector of ones. The authors analyse it and they obtain

$$\mathbf{v}_d = \mu^2\left[\frac{\sigma_n^2}{2\mu}\mathbf{1} + \frac{L\sigma_x^2\sigma_n^2}{4}\mathbf{1}\right] + O(\mu^{\frac{3}{2}})\mathbf{1}$$

for LMS and

$$\mathbf{v}_d = \mu^2 \left[ \frac{\sigma_n^2}{2\mu}\mathbf{1} + \frac{\frac{(L+1)P-1}{P}\sigma_x^2\sigma_n^2}{4}\mathbf{1} \right] + O(\mu^{\frac{3}{2}})\mathbf{1}$$

for SPU-LMS.

There is a third result in [36] related to spatial filtering, however this lies outside of the scope of this thesis.

The authors also include in the paper simulations which show the comparison between LMS, SPU-LMS, P-LMS, and S-LMS in terms of convergence. Therefore these comparisons are not repeated in this thesis.

Within the paper, the authors show that if the LMS algorithm converges in the mean, then so does the sequential LMS algorithm for the general case of arbitrary but fixed ordering of the sequence of partial coefficient updates. Also they conclude that S-LMS has similar convergence and steady state behaviour as LMS.

For SPU-LMS the conditions on step size for convergence in mean and mean square were shown to be equivalent to those of LMS.

The authors also verified by theory and by simulation that LMS and SPU-LMS have similar convergence criterion, and also the SPU-LMS has the same performance as P-LMS and S-LMS for stationary signals. The authors also demonstrate that choosing the coefficient to be updated randomly does not increase the final steady-state error as compared to the regular LMS algorithm.

## 2.3 Chapter Summary

In this chapter, previous work in partial update adaptive filtering techniques was reviewed. The first technique is to update one coefficient at each iteration this is called the maximum normalized least mean square (Max-NLMS) algorithm, this adaptive filter only adjusts the coefficient associated with the data element that has maximum absolute value in the filter memory at each iteration [27]. The second technique was to update a portion of the coefficients at each iteration, and those coefficients were the ones which have the largest magnitude gradient components on the error surface. Coefficients which have a small magnitude gradient component do not need to be updated as they will have little effect on the overall algorithm performance [28]. The third technique was to update entire blocks of the coefficients instead of selecting single filter coefficients for updating. Another technique was also studied, based on dividing the adaptive filter coefficients into small blocks and then updating a number of those blocks rather than the entire filter at every iteration, this was achieved by using a selection criterion, which ranked the regressor vector blocks according to their squared Euclidean norms (their energy) and selecting those blocks with the largest norms as the ones to be updated. Combining the data-selective updating from set-membership filtering with the reduced computational complexity from partial updating was also studied, the work in [35] showed that the set-membership filtering adaptation algorithms with partial updating can not only further reduce the computational complexity when compared with the partial update NLMS algorithm, but can also present a faster convergence for the same level of MSE. A new algorithm called the stochastic partial update LMS algorithm (SPU-LMS) was studied, based on choosing which of the subsets of the filter coefficients to update randomly. It was shown that for SPU-LMS the conditions on step size for convergence in the mean and mean square were shown to be equivalent to those of LMS. And also it was shown that LMS and SPU-LMS converge to similar regions within weight parameter space. Most importantly, the SPU-LMS algorithm overcomes the erratic convergence behaviour that can be observed in PU-LMS algorithm for which the update blocks are chosen deterministically [36]. Different techniques for partial update were shown in this chapter, starting from choosing one coefficient per update to selecting a block of coefficients to be updated, those blocks were chosen either in a deterministic or random manner. The purpose of including these different techniques and all the analysis was to verify that although it is well known that partial update techniques can reduce convergence speed, given sufficient time they can obtain the same accuracy measured by steady-state mean square error as the ordinary LMS algorithm. Some of those techniques will be extended and exploited in the context of channel shortening in the following chapters.

Chapter 3

# PROPERTY-RESTORAL BASED SEQUENTIAL BLIND CHANNEL-SHORTENING ALGORITHMS

In multicarrier or single-carrier cyclic prefix (SCCP) modulation, the transmitted sequence has redundancy because of the cyclic prefix. This redundancy has often been used for carrier frequency offset (CFO) estimation, where it is assumed that the channel is shorter than the cyclic prefix or that the channel is not time-dispersive. This redundancy can also be used in the property restoral sense in order to create a blind, adaptive channel shortener. In this chapter, the algorithms which attempt to restore each of the properties of the transmitted sequence that ought to be present in the equalized received sequence will be studied. Also in this chapter, the focus will be on how to develop adaptive channel shortener algorithms by using a philosophy called "property restoral". The idea is to look for and restore properties of the transmitted sequence that ought to be present in the equalized received sequence. Algorithms have been designed to restore those properties. In the multicarrier case several properties are available for creating blind, adaptive channel shorteners:

1. Cyclic prefix restoration [37].
2. Autocorrelation shortening [4,5].
3. Null-tone restoration [38,39].
4. The frequency-domain finite-alphabet methods [3].

## 3.1 Cyclic-Prefix Restoration

In paper [37], the author explores an algorithm called multicarrier equalization by restoration of redundancy (MERRY) which attempts to adapt the channel shortener with the aim of restoring the redundancy which is due to the cyclic prefix of the transmitted sequence in multicarrier or single-carrier cyclic prefix (SCCP) modulation.

In multicarrier transmission, modulation is achieved via an inverse fast Fourier transform (IFFT), and demodulation is successfully completed via an FFT. When the CP is added, the last $v$ samples are the same as the first $v$ samples in the transmitted symbol, but because of ICI and ISI in the channel, the modification to the received CP at the beginning and end of the symbol is likely to be different.

For an example, system with a data size of 8 samples and a cyclic prefix length of 2 samples. The CP is represented by $x(1)$, $x(2)$, and the symbol by $x(3)$,...., $x(10)$. Note that $x(2) = x(10)$ and $x(1) = x(9)$, but at the receiver, samples $y(2)$ and $y(10)$ would still be equal without a channel. However these samples are affected by the convolution of the channel and the input sequence. If the channel is no longer than the cyclic prefix, then the convolution for $y(10)$ only uses the $x$ data of the end of the symbol and the convolution for $y(2)$ only uses the redundant data in the prefix, making the two $y$ values equal. However, if the channel is longer than the cyclic prefix then the excess channel taps create terms that will be different in the two convolution sums. The algorithm that exploits this observation is next developed.

### 3.1.1 MERRY Algorithm

This section explains the basic MERRY algorithm. The SISO multicarrier system will be presented. Once the cyclic prefix (CP) has been added, the transmitted data complies with the relation

$$x(Mk+i) = x(Mk+i+N), \qquad i \in \{1, 2, ...., v\} \tag{3.1.1}$$

where $x$ is the source sequence to be transmitted through a linear finite impulse response (FIR) channel $h$, $k$ is the symbol (block) index, $N$ is the FFT size, $v$ is the cyclic prefix (CP) length, and $M = N + v$ is the total time-domain symbol size. The received data $r$ are obtained from $x$ by

$$r(Mk+i) = \sum_{j=0}^{L_h} h(j) \cdot x(Mk+i-j) + n(Mk+i) \tag{3.1.2}$$

and the equalized data $y$ are likewise obtained from $r$ by

$$y(Mk+i) = \sum_{j=0}^{L_w} w(j) \cdot r(Mk+i-j) \tag{3.1.3}$$

The channel $h$ has $L_h + 1$ taps, the TEQ has $L_w + 1$ taps, and the effective channel $c = h * w$ has $L_c + 1$ taps, where $L_c = L_h + L_w$.

The channel destroys the relationship in (3.1.1), because the ICI and ISI that affect the CP can be different from the ICI and ISI that affect the last $v$ samples in the symbol. Consider the example in the top of Figure (3.1), the transmitted samples $x(2)$ and $x(10)$ are exactly equal. However, at the output of the TEQ in the receiver, the interfering samples before the second sample are not all equal to their counterparts before the 10th sample. Notice that if $c(2)$, $c(3)$, and $c(4)$ were zero, then $y(2) = y(10)$. Thus, if $y(2) = y(10)$ in the mean squared error sense, then in an average sense, the channel and the CP will be equally short. Note that the last example shortens the channel to a window of $v$ taps: the first $v$ taps in the effective channel. The location of the window, and the transmission delay, can be changed by forming a different comparison. For example, as shown in the bottom of Figure (3.1), if $y(3) = y(11)$ rather than $y(2) = y(10)$, then the non-zero window of the effective channel becomes $[c1, c2]$ rather than $[c0, c1]$.

It can be noted that the channel has been shortened to $v$ taps, nevertheless a multicarrier system only requires shortening to $v + 1$ taps. However, when $v$ is large (e.g. 32 in ADSL), shortening the channel by an extra tap should have little effect on the performance.

Overall, if the effective channel has been shortened, then the last sample in the $\Delta$-delayed CP should be identical to the last sample in the symbol. The cost function that exploits this is given by

$$J_{merry}(\Delta) = E[(y(Mk + v + \Delta) - y(Mk + v + N + \Delta))^2], \quad (3.1.4)$$

$\Delta \in \{0, 1, 2, ......, M - 1\}$ where $\Delta$ is the

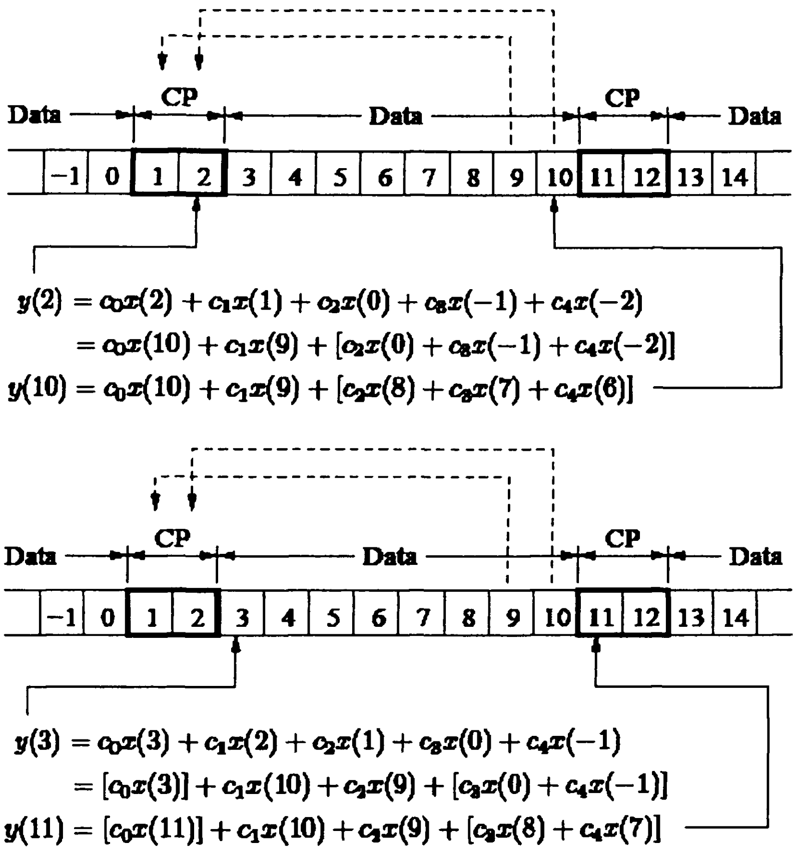
Data
CP
Data
CP
Data
−1
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
$y(2) = c_0x(2) + c_1x(1) + c_2x(0) + c_3x(-1) + c_4x(-2)$
$= c_0x(10) + c_1x(9) + [c_2x(0) + c_3x(-1) + c_4x(-2)]$
$y(10) = c_0x(10) + c_1x(9) + [c_2x(8) + c_3x(7) + c_4x(6)]$
Data
CP
Data
CP
Data
−1
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
$y(3) = c_0x(3) + c_1x(2) + c_2x(1) + c_3x(0) + c_4x(-1)$
$= [c_0x(3)] + c_1x(10) + c_2x(9) + [c_3x(0) + c_4x(-1)]$
$y(11) = [c_0x(11)] + c_1x(10) + c_2x(9) + [c_3x(8) + c_4x(7)]$

**Figure 3.1.** Illustration of the difference in the ISI at the received CP and at the end of the received symbol, delay of $\Delta = 0$. $x(i)$, $c_i$, and $y(i)$ are the transmitted data, effective channel, and TEQ output, respectively, and the bracketed terms are intended to be suppressed.

symbol synchronization parameter, which represents the desired delay of the effective channel. The choice of $\Delta$ affects the cost function, and is an important parameter in equalization [37].

A stochastic gradient descent of (3.1.4) leads to a blind, adaptive TEQ, since the transmitted data is not necessary to be known. The resulting algorithm MERRY performs a stochastic gradient descent of (3.1.4), with a constraint to avoid the trivial solution $\mathbf{w} = \mathbf{0}$ [40], [41]. For a SISO system, the basic MERRY algorithm becomes

Given $\Delta$, for symbol $k = 0, 1, 2, .....$,

$$\tilde{\mathbf{r}}(k) = \mathbf{r}(Mk + v + \Delta) - \mathbf{r}(Mk + v + N + \Delta) \qquad (3.1.5)$$

$$e(k) = \mathbf{w}^T(k)\tilde{\mathbf{r}}(k) \qquad (3.1.6)$$

$$\hat{\mathbf{w}}(k+1) = \mathbf{w}(k) - \mu e(k)\tilde{\mathbf{r}}(k) \qquad (3.1.7)$$

$$\mathbf{w}(k+1) = \frac{\hat{\mathbf{w}}(k+1)}{||\hat{\mathbf{w}}(k+1)||} \qquad (3.1.8)$$

where $\mathbf{r}(i) = [r(i), r(i-1), ....., r(i - L_w)]^T$

The norm $||.||$ can be the common $L_2$ norm, the $L_p$ norm for $p$ an integer, the norm with respect to a matrix, or any other conceivable norm. Note that MERRY is a simple vector update rule, with the added complexity of a renormalization. Because MERRY compares the CP to the end of the symbol, only one update is possible per symbol. Other implementations of the constraint include fixing one tap to unity, maintaining a channel estimate and renormalizing to enforce $||c|| = 1$ instead of $||\mathbf{w}|| = 1$, or including a penalty term in the cost function to enforce the norm constraint [37]. In [42], the authors show that MERRY can also be implemented in transmitter-zero OFDM (TZ-OFDM) systems, which is opposed to cyclic prefix OFDM (CP-OFDM) systems. TZ-OFDM systems transmit zeros during the guard period that is used for the cyclic prefix in CP-OFDM. This is equivalent to assuming that the samples in the CP $x(1)$ and $x(2)$ in Figure (3.1) are zero, rather than copies of the data. The MERRY cost function then becomes

$$J_{merry,TZ}(\Delta) = 2E[|y(Mk + v + \Delta)|^2], \tag{3.1.9}$$

$$\Delta \in \{0, 1, 2, ......, M - 1\}$$

The update equation is a stochastic gradient descent of (3.1.9) with a periodic renormalization. The advantage of MERRY is the low computational complexity but slow convergence is the main drawback with MERRY as it only updates once per symbol. The length of the shortener, $L_w$, is typically chosen as a function of the length of the channel, $L_h$ in this work a channel shortener of length 16 is used to match the work of [5]

## 3.2 Autocorrelation Shortening

In this section adaptive TEQs that rely on correlation estimates will be studied. An algorithm called sum-squared autocorrelation minimization (SAM) [4], which minimizes the sum-squared autocorrelation terms of the effective channel impulse response outside a window of a CP-length has been developed. SAM converges much faster than the MERRY algorithm but at the expense of higher complexity. SAM behaves much like the constant modulus algorithm (CMA) equalization algorithm in that it does not require the user to specify the desired delay and can adapt before carrier frequency offset (CFO) recovery is performed. Several variants of SAM have been proposed. The sum-absolute autocorrelation minimization (SAAM) algorithm [26] replaces the squares of the autocorrelation with their absolute values. Also the single lag autocorrelation minimization (SLAM) [5] aims to minimize the square of only a single autocorrelation and thereby reduce computational complexity in its realization.

### 3.2.1 SAM Algorithm

In [4], the authors explore an algorithm called blind, adaptive channel shortening by sum-squared autocorrelation minimization (SAM) for updating the coefficients of a time-domain equalizer in multicarrier modulation system. The idea is to minimize the sum-squared autocorrelation terms of the channel impulse response outside a window of a CP length.

#### System Model

The system model which is shown in Figure (3.2) is assumed. The input signal $x(k)$ is the source sequence to be transmitted through a linear finite-impulse-response (FIR) channel $\mathbf{h}$ of length $(L_h + 1)$ taps, $r(k)$ is the received signal, which will be filtered through an $(L_w + 1)$-tap TEQ with an impulse response vector $\mathbf{w}$ to obtain the output sequence $y(k)$. In this work, real signals are assumed but generalization to the complex case is straight-forward. $\mathbf{c} = \mathbf{h} * \mathbf{w}$ is denoted as the shortened or effective channel assuming $\mathbf{w}$ is in steady-state where $*$ denotes discrete time convolution. It is also assumed that $2L_c < N_{fft}$ holds, where $L_c$ is the order of effective channel and $N_{fft}$ is the FFT size [4]. The signal $n(k)$ is a zero-mean, i.i.d., noise sequence, uncorrelated with the source sequence with variance $\sigma_v^2$. The received sequence $r(k)$ is

$$r(k) = \sum_{j=0}^{L_h} h(j)x(k-j) + n(k) \tag{3.2.1}$$

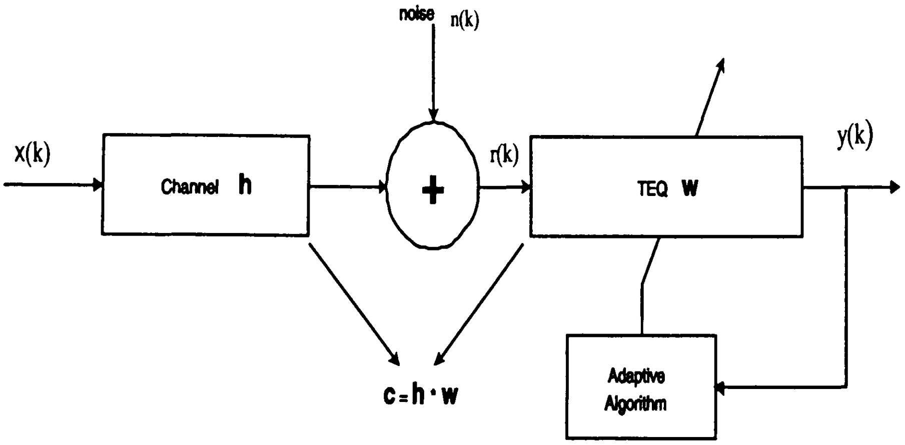

**Figure 3.2.** System model for blind adaptive channel shortening.

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

$$y(k) = \sum_{j=0}^{L_w} w(j)r(k-j) = \mathbf{w}^T \mathbf{r}_k \tag{3.2.2}$$

where $\mathbf{r}_k = [r(k) \quad r(k-1)\cdots r(k-L_w)]^T$ and $\mathbf{w}$ is the impulse response vector of the TEQ $\mathbf{w} = [w_0 \quad w_1 \quad w_2 \cdots w_{L_w}]^T$.

### SAM Cost Function

In this section the SAM cost function and its use will be studied, and how to blindly estimate it from the measured data will be shown.

The underlying idea that allows the development of SAM is that for the effective channel $\mathbf{c}$ to have taps equal to zero outside a window which have the size $(v+1)$, its autocorrelation values should be equal to zero outside a window of size $(2v+1)$. The autocorrelation sequence of the combined channel-equalizer impulse response in SAM is shown below

$$R_{cc}(l) = \sum_{j=0}^{Lc} c(j)c(j-l) \tag{3.2.3}$$

and for a shortened channel, it must satisfy

$$R_{cc}(l) = 0, \quad \forall |l| > v \tag{3.2.4}$$

Hence, one possible way of performing channel shortening is by ensuring that (3.2.4) is satisfied by the autocorrelation function of the combined response. However, this has a trivial solution when $\mathbf{c} = 0$ or equivalently $\mathbf{w} = 0$. Avoiding this trivial solution can be achieved by imposing a norm constraint on the equalizer, for instance $\|\mathbf{w}\|_2^2 = 1$, or equivalently $R_{cc}(0) = 1$.

It should be noted that it is not possible to achieve perfect nulling of the autocorrelation values outside the window of interest, because when a finite length baud-spaced TEQ is used, perfect channel shortening is not possible. This is because if the channel has $L_h$ zeros, then the effective response will always have $L_h + L_w$ zeros. If the length of the channel had decreased to, for example, $L_s < L_h$ taps, then the combined response would only have $L_s$ zeros, which contradicts the previous statement. Hence, a cost function $J_{sam}$ is an attempt to minimize (rather than nulling) the sum-square of the autocorrelation terms, is defined,

$$J_{sam} = \sum_{l=v+1}^{L_c} (R_{cc}(l))^2 \tag{3.2.5}$$

The trivial solution can be avoided by imposing a norm constraint on the TEQ i.e., $\|\mathbf{w}\|_2^2 = 1$. The TEQ optimization problem can then be stated as

$$\mathbf{w}^{opt} = \arg \min_{\|\mathbf{w}\|_2^2=1} J_{sam} \tag{3.2.6}$$

The relation between the autocorrelation sequence of the output $y(k)$

of the TEQ and the autocorrelation sequence of the effective channel is given by [4]

$$R_{yy}(l) = R_{cc}(l) + \sigma_v^2 R_{ww}(l) \tag{3.2.7}$$

Under the noiseless scenario, $R_{yy}(l) = R_{cc}(l)$, therefore equation (3.2.5) can be rewritten as

$$J_{sam} = \sum_{l=v+1}^{L_c} (R_{cc}(l))^2 = \sum_{l=v+1}^{L_c} (R_{yy}(l))^2 \tag{3.2.8}$$

In the presence of noise, (3.2.8) is only approximately true. An approximation to the cost function in (3.2.5), denoted by $\hat{J}_{sam}$ is given by

$$\begin{aligned} \hat{J}_{sam} &= \sum_{l=v+1}^{L_c} (R_{yy}(l))^2, \\ &= \sum_{l=v+1}^{L_c} (R_{cc}(l))^2 + 2\sigma_v^2 R_{cc}(l) R_{ww}(l) \\ &\quad + \sigma_v^4 (R_{ww}(l))^2 \end{aligned} \tag{3.2.9}$$

In most situations, the TEQ length $(L_w + 1)$ is shorter than the cyclic prefix length, $v$. In this case, $R_{ww}(l)$ does not exist for the stated lag in (3.2.9). Therefore, both the noise terms in (3.2.9) can be neglected. Even if the TEQ is longer than the cyclic prefix, the second and third terms which have been added are very small because of their multiplication with $\sigma_v^2$ and $\sigma_v^4$. The noise variance $\sigma_v^2$ is usually small for ADSL channels [4]. A typical value of SNR in ADSL environments is 40 dB [26]. Therefore, practically it is assumed that $\hat{J}_{sam} \cong J_{sam}$ as in Equation (3.2.5).

### Adaptive Algorithm

The steepest gradient-descent algorithm to minimize the SAM cost $J_{v+1}$ is

$$\mathbf{w} = \mathbf{w}^{old} - \mu \nabla_{\mathbf{w}} \left( \sum_{l=v+1}^{L_c} E[y(k)y(k-l)]^2 \right) \tag{3.2.10}$$

where $\mu$ denotes the step size, and $\nabla_{\mathbf{w}}$ is the gradient evaluated at $\mathbf{w} = \mathbf{w}^{old}$. The instantaneous cost function is defined, where the expectation operation is replaced by a moving average over a user-defined window of length $N_{avg}$

$$J_{v+1}^{inst}(k) = \sum_{l=v+1}^{L_c} \left\{ \sum_{n=kN_{avg}}^{(k+1)N_{avg}-1} \frac{y(n)y(n-l)}{N_{avg}} \right\}^2 \tag{3.2.11}$$

where $N_{avg}$ is a design parameter and it should be large enough to give a good estimate of the expectation, but no larger, as the algorithm complexity is proportional to $N_{avg}$. (One possible choice for block-based systems is $N_{avg} = M$, where $M$ is the total block size. This allows for one update per block, as for the MERRY algorithm). The stochastic gradient-descent SAM algorithm is then given by [4]

$$\mathbf{w}(k+1) = \mathbf{w}(k) - 2\mu \left\{ \sum_{n=kN_{avg}}^{(k+1)N_{avg}-1} \frac{y(n)y(n-l)}{N_{avg}} \right\} \times \left\{ \nabla_{\mathbf{w}} \left( \sum_{n=kN_{avg}}^{(k+1)N_{avg}-1} \frac{y(n)y(n-l)}{N_{avg}} \right) \right\} \tag{3.2.12}$$

which can be simplified to

$$\mathbf{w}(k+1) = \mathbf{w}(k) - 2\mu \left\{ \sum_{n=kN_{avg}}^{(k+1)N_{avg}-1} \frac{y(n)y(n-l)}{N_{avg}} \right\}$$
$$\times \left\{ \sum_{n=kN_{avg}}^{(k+1)N_{avg}-1} \left( \frac{y(n)\mathbf{r}_{n-l} + y(n-l)\mathbf{r}(n)}{N_{avg}} \right) \right\} \tag{3.2.13}$$

The TEQ update algorithm described in (3.2.13) will be referred to as the sum-squared autocorrelation minimization (SAM) algorithm, as it attempts to minimize the cost function described in (3.2.5).

## 3.3 Null-Tone Restoration

The presence of null tones in the transmitted data is another common property of multicarrier signals. For example, in IEEE 802.11a, 12 of the 64 tones are null tones, with 6 null tones located at the each edge of the frequency band. This provides a buffer to limit adjacent channel interference. It has also been suggested in [39] that this can be viewed as over-sampling the transmitted signal (before transmission, rather than at the receiver), since of the 64 inputs, 52 are data and 12 are zeros. A blind, adaptive channel shortening algorithm can be obtained with the aim of restoring the values of these tones to zero at the output of the FFT at the receiver [39], [43]. This results in a carrier nulling algorithm (CNA). The CNA cost function is the average power of the outputs on the generally complex tones, denoted $z_i$ that should theoretically be null.

$$J_{cna} = \sum_{i \in Null\ tones} E[|z_i|^2]. \tag{3.3.1}$$

The CNA algorithm is a constrained gradient descent of this cost function. This leads to a very simple LMS-like structure, although due to several matrix vector products, the computational complexity is slightly higher, the full details of which can be found in [44]

The CNA algorithm has much in common with the MERRY algorithm. As for MERRY, CNA can only update once per symbol. This is because the cost function is measured at the output of the FFT, once per block. Also, as with the MERRY and SAM algorithms, a constraint is needed for CNA to avoid the all-zero solution. De Courville, et al. [39] chose to implement a unit norm constraint on the channel shortener via periodic re-normalization. Assuming that the unit norm constraint is used, the CNA algorithm solves for the eigenvector corresponding to the minimum eigenvalue of the autocorrelation matrix of the outputs on the null tones [39], whereas MERRY seeks the eigenvector corresponding to the minimum eigenvalue of the autocorrelation matrix of a difference of two vectors of received samples [40]. Analysis of the CNA algorithm is difficult due to the nature of the update. The work in [39] shows that the zeroforcing equalizer (not a more generic channel shortener) minimizes the CNA cost function, hence CNA should be used in multicarrier systems that do not employ a cyclic prefix.

## 3.4 The Frequency-Domain Finite-Alphabet Methods

The time-domain data in a multicarrier system is not-finite alphabet, where as this is the case for the frequency domain data at the output of the demodulating FFT. This means that a decision-directed or constant modulus cost function can be proposed in the frequency-domain. However, now there are $N$ tones, so the cost must be summed over the $N$ outputs. For example, the frequency-domain decision-directed and constant modulus cost functions [38]

$$J_{dd} = \sum_{i \in \ tones} \beta_i E[(Q\{z_i\} - z_i)^2] \tag{3.4.1}$$

and

$$J_{cm} = \sum_{i \in \ tones} \beta_i E[(|z_i|^2 - \gamma_i)^2] \tag{3.4.2}$$

where $\beta_i$ is a designer chosen positive weight, $Q\{.\}$ is a zero memory non linearity which finds the nearest constellation point in a finite alphabet, and $\gamma_i$ is the dispersion constant, which can be selected individually for each tone. The choice of non-uniform $\beta_i's$ can be used to provide unequal error protection across the tones. The CNA algorithm can be thought of as using a special case of (3.4.1): it is a decision-directed algorithm in which the null tones will be compared to a finite alphabet that is simply the value 0, so $Q\{z_i\} = 0$ always. If the channel is shortened, the output will be QAM data on each non-null tone, but the modulus of the points will not be correct until after the bank of one-tap frequency-domain equalizers (the FEQ). Thus, the frequency-domain cost must be measured at the output of the FEQ. This means that the TEQ and FEQ, which are connected in series, will both be adapting based on the $N$ outputs of the FEQ. Typically, adaptive devices are analyzed under the assumption that each device operates independently, and this sort of adaptation of a series of elements is not well understood [45]. In [46], the authors have proposed a trained, non-adaptive design, that operates in the frequency domain. Their method maximizes the energy at the output of the pilot tones divided by the energy of the null tones. In principle, this idea could be used to create a trained, adaptive algorithm, that restores both the pilots and the null tones, as in a combination of CNA and frequency-domain LMS.

## 3.5 Chapter Summary

In this chapter, the algorithms which attempt to restore each of the properties of the transmitted sequence that ought to be present in the equalized received sequence were studied. This chapter shows that in order to create a blind, adaptive channel shortener, the redundancy which the transmitted sequence has due to the cyclic prefix in multi-carrier or single-carrier cyclic prefix (SCCP) modulation, can be used in the property restoral sense. Algorithms using a philosophy called "property restoral" were studied such as the MERRY algorithm which attempts to adapt the channel shortener with the aim of restoring the redundancy which is due to the cyclic prefix of the transmitted sequence. On the other hand, the SAM algorithm minimizes the sum-squared autocorrelation terms of the effective channel impulse response outside a window of a CP-length. This chapter also shows that the presence of null tones in the transmitted data is another common property of multicarrier signals. A carrier nulling algorithm (CNA) can therefore be derived with the goal of restoring the values of those tones to zero at the output of the receiver FFT. Algorithms based on correlation estimates such as SAM algorithm will be the focus of this thesis as these algorithms converge faster than MERRY algorithm but with a higher complexity. In this thesis, the focus will be on how the complexity of SAM and SLAM algorithms can be reduced.

Chapter 4

# ROBUST BLIND ADAPTIVE CHANNEL SHORTENING FOR IMPULSIVE NOISE ENVIRONMENTS

In this chapter novel blind adaptive channel shortening algorithms, the deterministic partial update sum-absolute autocorrelation minimization (DPUSAAM) algorithm and the random partial update sum-absolute autocorrelation minimization (RPUSAAM) algorithm are proposed for multicarrier modulation systems. These algorithms are based on updating only a portion of the coefficients of the channel shortening filter at each time sample instead of the entire set of coefficients. This work is the first attempt in the field of using partial update filtering in blind adaptive channel shortening. The algorithms are also designed to be robust to impulsive noise impairment found in ADSL channels. These algorithms have low computational complexity whilst retaining essentially identical performance to the sum-absolute autocorrelation minimization (SAAM) algorithm [26]. Simulation studies show the ability of the DPUSAAM algorithm and the RPUSAAM algorithm to achieve channel shortening and hence an acceptable level of bitrate within a multicarrier system.

## 4.1 Gaussian Noise Model

In the design and analysis of signal processing systems, the Gaussian noise model [26,47] is extensively used. The probability density function of a zero mean Gaussian model is given by

$$f_n(x) = \frac{1}{\sqrt{2\pi}\sigma} exp\left\{-\frac{x^2}{2\sigma^2}\right\}$$

where $\sigma^2$ is the variance of the distribution. The additive white Gaussian noise assumption in digital communication theory very much simplifies the design and analysis of receiver structures. The following theorem justifies the Gaussian noise assumption [48].

**Theorem 4.1.1.** ***(Central Limit Theorem, CLT)***

*Given $x_1, x_2, \ldots, x_N$ a sequence of independent identically distributed (i.i.d.) random variables with non Gaussian distribution and mean $\mu$ and variance $\sigma^2$. Then, as $N \rightarrow \infty$, the distribution of the normalized sum*

$$S_N = \frac{1}{N}\sum_{j=1}^{N} x_j$$

*converges almost surely to a Gaussian process with the same mean and variance as $x_j$* [49].

Therefore, the Gaussian noise assumption plays a basic role in formulating many of the theorems of digital communication, estimation and detection theory [49]. This is suitable in Gaussian noise environments but even mild deviations from the Gaussian assumption can have harmful effects [50–52]. Noise sources encountered in physical environments for example, urban and man-made RF noise, underwater acoustic noise, atmospheric noise, radar clutter noise and telephone circuit noise are generally non-Gaussian. They are impulsive, i.e., having higher probability of producing outliers than predicted by an additive Gaussian noise model [53–56].

## 4.2 Impulse noise in ADSL

Impulse noise is a non-stationary stochastic electromagnetic interference which consists of random occurrences of energy spikes with random amplitude and spectral content. The causes of impulse noise on the telephone line are diverse and vary from opening of the refrigerator door, when phones ring on lines sharing the same binder, and industrial electrical appliances, and transport vehicles, to atmospheric noise from electrical discharges. A number of studies by various research groups of impulses have resulted in analytical models based on the statistical analysis of over 100,000 impulses [25]. The Cook pulse model, for example, is the most widely used analytical model [57]. Cook found that the amplitude of the impulse increases with the bandwidth of the DSL system under test. This follows from the wider bandwidth of the DSL receiver filter, which means less impulse attenuation. In [58] an introduction is given to a method to simulate the amplitude, length, inter-arrival times and the spectral characteristics of the impulses. The statistics derived from observations of impulse noise on the telephone networks of British Telecom (BT) and Deutsche Telekom (DT) were used as the parameters of their model. It has also been argued that impulses defy analysis and people sometimes use representative worst case waveforms e.g., the ADSL standard [59] itself uses two measured impulses. However, in common with other researchers, Gaussian mixture and $\alpha$-stable distributions are used in this thesis for modelling the impulsive noise due to their suitability for representing practical impulsive noise [60].

### 4.2.1 Gaussian-mixture noise model

The Gaussian mixture model is an analytically simple impulse noise model [51, 61, 62]. It is popular due to its mathematical tractability. The probability density function is given by

$$f_n(x) = (1-p)f_v(x) + pf_i(x)$$

where $f_v$ is the Gaussian pdf with variance $\sigma_v^2 > 0$ and $f_i$ is the Gaussian pdf with higher variance $d^2\sigma_v^2$. The parameter $p \,\epsilon [0,1]$ is the probability of contribution of the components from this high variance distribution. The parameter $d \geq 1$ is the ratio of the standard deviations of the two variances. The effect of different shapes of Gaussian mixture noise density can be simulated to evaluate the algorithm performance simply by varying $p$ and $d$.

### 4.2.2 Properties of Stable processes

The $\alpha$-stable distribution, which can model phenomena of an impulsive nature [63], is a generalization of the Gaussian distribution and is appealing because it shares several desirable properties with the Gaussian model, such as the *stability property* and *generalized* form of the *Central Limit Theorem* [49]. In fact, $\alpha$-stable distributions can be described by their characteristic function as follows:

A random variable $x$ is said to have a stable distribution, denoted by $x \sim S_\alpha(\gamma, \beta, a)$, if and only if its characteristic function has the form [60]

$$\varphi(t) = exp\{jat - \gamma|t|^\alpha[1 + j\beta sign(t)w(t,\alpha)]\} \tag{4.2.1}$$

where

$$w(t,\alpha) = \begin{cases} tan\frac{\alpha\pi}{2}, & if\ \alpha \neq 1 \\ \frac{2}{\pi}\ log|t|, & if\ \alpha = 1 \end{cases}$$

$$sign(t) = \begin{cases} 1, & if\ t > 0 \\ 0, & if\ t = 0 \\ -1, & if\ t < 0 \end{cases} \tag{4.2.2}$$

The four parameters that describe the stable distribution are therefore $-\infty < a < \infty,\ \gamma > 0,\ 0 < \alpha \leq 2,\ -1 \leq \beta \leq 1$ [60]. In more detail,

- $\alpha$ is termed the *characteristic exponent* and determines the thickness of the tails of the distribution. Smaller values of $\alpha$ yield heavier tailed distributions and vice versa. An $\alpha = 2$ gives the Gaussian distribution. Another special case is the Cauchy distribution which corresponds to $\alpha = 1$ and $\beta = 0$.

- $\gamma$ is a *dispersion* parameter. It is similar to variance of a Gaussian distribution and equals half the variance in the Gaussian case.

- $\beta$ is the index of symmetry. When $\beta = 0$, it corresponds to a symmetric distribution around the location parameter. The resulting distribution is called a Symmetric $\alpha$-Stable (S$\alpha$S) distribution.

- $a$ which is the *location* parameter. It is the mean if $1 < \alpha \leq 2$ and the median if $0 < \alpha < 1$.

By taking the inverse Fourier transform of the characteristic function in (4.2.1) the pdf of a stable distribution can be obtained. No closed form expression exists for the stable density, except for the Gaussian ($\alpha = 2$), Cauchy ($\alpha = 1, \beta = 0$), and Pearson($\alpha = 1/2, \beta = -1$) cases [49].

If it is assumed that S$\alpha$S distributions have a zero location parameter i.e., $a = 0$, the resulting characteristic function only depends on $\alpha$ and $\gamma$, i.e.

$$\varphi(t) = exp(-\gamma|t|^{\alpha})$$

whose pdf is given by

$$S_{\alpha}(\gamma, 0, 0) = \begin{cases} \frac{1}{\pi\gamma^{1/\alpha}} \sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k!}\Gamma(\alpha k + 1) sin\left(\frac{k\alpha\pi}{2}\right)\left(\frac{|x|}{\gamma^{1/\alpha}}\right)^{-\alpha k - 1}, \\ 0 < \alpha < 1 \\ \frac{\gamma}{\pi(x^2+\gamma^2)}, \quad \alpha = 1 \\ \frac{1}{\pi\alpha\gamma^{1/\alpha}} \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k)!}\Gamma\left(\frac{sk+1}{\alpha}\right)\left(\frac{x}{\gamma^{1/\alpha}}\right)^{2k} 1 < \alpha < 2 \\ \frac{1}{2\sqrt{\gamma\pi}} exp\left(-\frac{x^2}{4\gamma}\right) \alpha = 2 \end{cases}$$

where $\Gamma(\cdot)$ is the usual Gamma function defined by

$$\Gamma(x) = \int_0^{\infty} t^{x-1}e^{-t}dt \qquad (4.2.3)$$

Figure (4.1) shows the pdfs of zero-mean S$\alpha$S distributions with different values of $\alpha$ [1]. The value of the dispersion parameter $\gamma$ is equal to unity. It can be seen that the non-Gaussian stable density functions differ from the corresponding Gaussian density in the following ways. For small values of $x$, the S$\alpha$S densities are more peaked than the normal densities. For intermediate ranges of $|x|$, the S$\alpha$S distributions have lower values than the normal density. But unlike the Gaussian

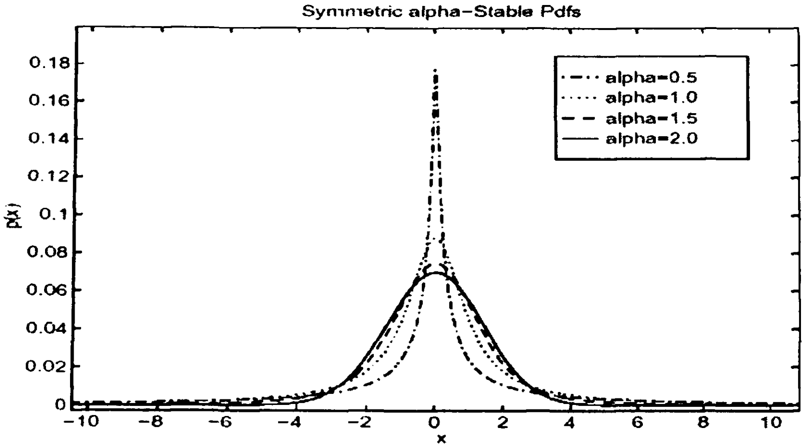

**Figure 4.1.** Effect of $\alpha$ on the pdf of an alpha-stable distribution with $\beta = 0$, $a = 0$ and $\gamma = 1$ [1].

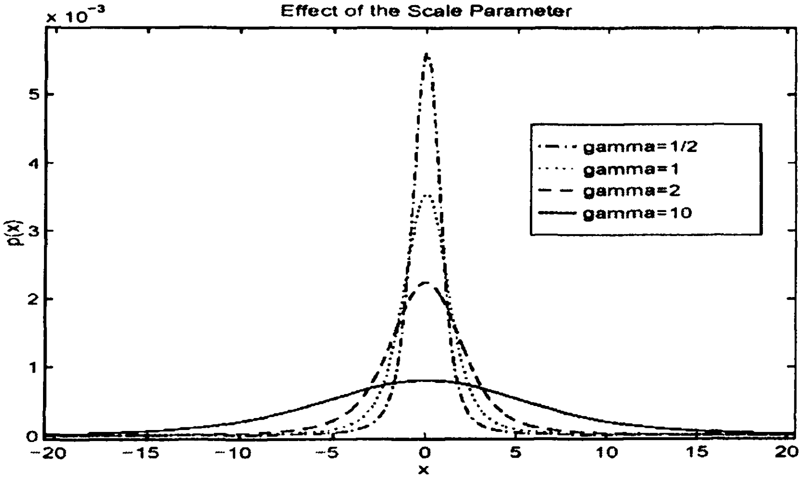

**Figure 4.2.** Effect of $\gamma$ of an alpha-stable distribution with $\beta = 0$, $a = 0$ and $\alpha = 1$ [1].

density which has exponential tails, S$\alpha$S distributions have algebraic tails. Figure (4.2) shows the effect of $\gamma$ on the pdf of a zero-mean S$\alpha$S distribution [1]. The value of the characteristic exponent $\alpha$ is equal to unity. It shows that the effect of $\gamma$ is analogous to variance in the Gaussian density case and it determines the spread of the samples around the location parameter at the respective impulsiveness as determined by the value of $\alpha$.

### 4.2.3 Fractional Lower Order Moments

The stable distributions do not possess finite second order moments except in the Gaussian limiting case. It is known that, for a non-Gaussian stable distribution with characteristic $\alpha$, only moments of order less than $\alpha$ are finite. More formally, this is stated as

**Theorem 4.2.1.** *Let $x$ be a stable random variable. If $0 < \alpha < 2$, then*

$$E|x|^p = \infty \quad if \ p \geq \alpha$$

*and*

$$E|x|^p < \infty \quad if \ 0 < p < \alpha$$

*if $\alpha = 2$, then*

$$E|x|^p < \infty \quad for \ all \ p \geq 0$$

In [60] the proof of this theorem is presented. For $0 < \alpha \leq 1$, stable processes have infinite first and higher moments; for $1 < \alpha < 2$, they have finite first moment and all moments of order $p < \alpha$; and all moments exist for $\alpha = 2$. All the moments of an S$\alpha$S random variable with $0 < \alpha < 2$ of order less than $\alpha$ are termed Fractional Lower Order Moments (FLOMs). The following preposition explains the relationships between the FLOMs of an S$\alpha$S random variable, its dispersion and its characteristic exponent [60].

**Proposition 4.2.1.** *Let $x \sim S_\alpha(\gamma, 0, 0)$. Then*

$$E(|x|^p) = C(p, \alpha)\gamma^{p/\alpha} \quad if \ 0 < p < \alpha$$

*where*

$$C(p,\alpha) = \frac{2^{p+1}\Gamma\left(\frac{p+1}{2}\right)\Gamma(-p/\alpha)}{\alpha\sqrt{\pi}\Gamma(-p/2)}$$

*is a function of $\alpha$ and $p$ and is independent of $x$.*

Most of the research in the area of modeling noise by $\alpha$-stable distributions has focused on the design of near-optimum receivers operating in impulsive noise environments, parameter estimation of linear processes, direction of arrival estimation, blind channel estimation, bearing estimation and other problems related to radar and signal modeling. Bibliographies in [1, 49] show a comprehensive list.

In [64] it was found that an $\alpha$-stable distribution is the best to describe the outliers in the noise over telephone lines which can be observed to be non-Gaussian. The value of $\alpha$ was found to be in the range $1.9 < \alpha < 2$. The concept of a *minimum dispersion* (MD) criterion for non-Gaussian stable models is introduced in [65] as a direct generalization of the MMSE criterion which is optimal for Gaussian models. The important observation from the proposition (4.2.1) is that FLOMs are related to the dispersion $\gamma$, through only a constant. Therefore the MD criterion dictates that the $p$-th lower order moment should be minimized, where $0 < p < \alpha$. The range of $\alpha$ found in [64] and mathematical convenience dictates the use of the $l_1$-norm for the case of noise on telephone lines. For ADSL channel noise, without loss of generality, a zero-mean symmetric $\alpha$-stable (S$\alpha$S) distribution is assumed, where $0 < \alpha \leq 2$ which controls the impulsiveness of the distribution.

### 4.2.4 Geometric Power of Stable Noise

The standard SNR definition based on noise variance cannot be used due to the infinite variance of stable distributions. Instead, a Geometric-SNR (G-SNR) definition has been used [66]. If $A$ is the amplitude of a signal in additive S$\alpha$S noise of geometric power $S_o^2$, then the G-SNR in dB, has the form

$$G - SNR = 10 \log \left[ \frac{1}{2C_g} \left( \frac{A}{S_o} \right)^2 \right] \tag{4.2.4}$$

where $C_g \cong 1.98$ is the exponential of the Euler constant and

$$S_o = \frac{(C_g \gamma)^{\frac{1}{\alpha}}}{C_g} \tag{4.2.5}$$

Here $\alpha$ is the characteristic exponent and $\gamma$ is the dispersion of the $S\alpha S$ noise. The normalized constant $2C_g$ in (4.2.4) ensures that for the Gaussian case ($\alpha = 2$), the definition of G-SNR coincides with that of the standard SNR. S$\alpha$S noise is generated in this work by modifying the Matlab code available at [66] which is based on the Chambers-Mallows-Stuck method [67]. Samples of S$\alpha$S noise at G-SNR of 40dB and at different values of $\alpha$ are shown in Figure (4.3). The signal amplitude is kept at unity. Plot (b) shows the impulse noise for an $\alpha = 1.99$ value close to 2, the noise samples characterized by a G-SNR possess almost the same strength as the Gaussian noise samples of plot (a) where the value of $\alpha$ is 2. Nonetheless, the concept of variance can lead to the misleading conclusion that the stable noise with $\alpha = 1.99$ has infinite strength, although this is clearly not the case in plot (b). As the value of $\alpha$ is decreased to 1.5, the noise becomes impulsive in nature having samples of larger amplitude as shown in plot (c). The number of outliers and their amplitude/strength is more visible in plot (d) where plot (c) is magnified on the y-axis.

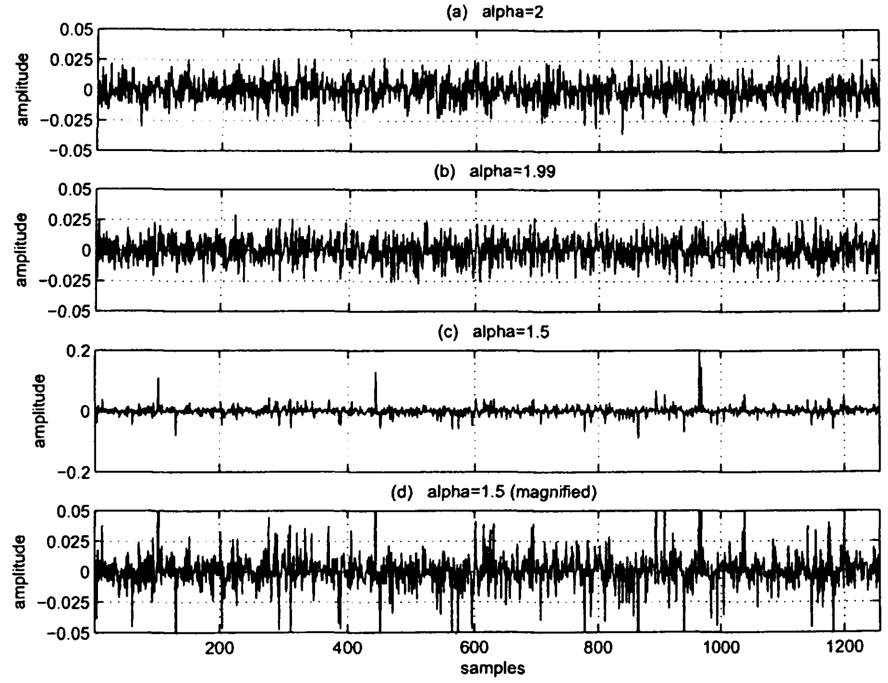

**Figure 4.3.** Gaussian and impulsive noise at GSNR=40dB. The signal amplitude is unity. (a) Gaussian noise $\alpha = 2$, (b) impulse noise $\alpha = 1.95$, (c) more impulsive noise $\alpha = 1.5$, and (d) magnified view of (c).

## 4.3 System Model

The system model shown in Figure (3.2) is used for blind adaptive channel shortening. The input signal $x(k)$ typically drawn from a finite constellation to represent the source sequence to be transmitted through a linear finite-impulse-response (FIR) channel $\mathbf{h}$ of length $(L_h + 1)$taps; $r(k)$ is the received signal, which will be filtered through an $(L_w+1)$-tap TEQ with an impulse response vector $\mathbf{w}$ to obtain the output sequence $y(k)$. The vector $\mathbf{c} = \mathbf{h}*\mathbf{w}$ is the effective channel of order $L_c = L_h+L_w$.

The symbol $*$ represents discrete time convolution and $L_h$ and $L_w$ are the orders of the channel and TEQ respectively. It is assumed that $2L_c < N$ holds, where $N$ is the FFT size [4]. The signal $n(k)$ is a zero-mean, i.i.d., noise sequence uncorrelated with the source sequence which has variance $\sigma_v^2$. The received sequence $r(k)$ is

$$r(k) = \sum_{j=0}^{L_h} h(j)x(k-j) + n(k) \tag{4.3.1}$$

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

$$y(k) = \sum_{j=0}^{L_w} w(j)r(k-j) = \mathbf{w}^T \mathbf{r}_k \tag{4.3.2}$$

where $\mathbf{w} = [w(0)w(1), \ldots, w(L_w)]^T$ is the impulse response vector of the TEQ and $\mathbf{r}_k = [r(k)\ r(k-1)\ \ldots\ r(k-L_w)]^T$.

## 4.4 SAAM

The idea of SAAM is based on minimizing the sum of the absolute values of the autocorrelation of a channel over a specific interval. This interval is outside of the region the effective channel is allowed to be non zero and is chosen to be integer values from lag $v+1$ until lag $L_c$. The cost function of SAAM is denoted $J_{v+1}$. The reasons for taking absolute values have been explained in [26] namely to mitigate large errors in the sample autocorrelation estimates. This is in contrast to the cost function of [4] based on the sum of squared autocorrelation values for the same lags. The autocorrelation sequence of the effective channel, **c** has the form

$$R_{cc}(l) = \sum_{j=0}^{L_c} c(j)c(j-l)$$

when the effective channel **c** has zero taps outside a window of size $(v+1)$, and for a shortened channel, it must satisfy,

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

Then the cost function $J_{v+1}$ in SAAM is defined based upon minimizing the sum-absolute autocorrelation terms, i.e.,

$$J_{v+1} = \sum_{l=v+1}^{L_c} |R_{cc}(l)| \qquad (4.4.1)$$

The trivial (anti) solution of $\mathbf{w} = 0$ can be avoided by imposing a norm constraint on the equalizer i.e., $\|\mathbf{w}\|_2^2 = 1$. The optimization problem can then be stated as

$$\mathbf{w}^{opt} = \arg_{\mathbf{w}} \min_{\|\mathbf{w}\|_2^2=1} J_{v+1}$$

The autocorrelation sequence of the output $y(k)$ is given by

$$\begin{aligned} R_{yy}(l) &= E[y(k)y(k-l)] \\ &= E[(\mathbf{c}^T\mathbf{x}_k + \mathbf{w}^T\mathbf{n}_k)\ (\mathbf{x}_{k-l}^T\mathbf{c} + \mathbf{n}_{k-l}^T\mathbf{w})] \qquad (4.4.2) \end{aligned}$$

where $\mathbf{x}_k = [x(k), x(k-1), \ldots, x(k-L_c)]^T$ and $\mathbf{n}_k = [n(k), n(k-1), \ldots, n(k-L_w)]^T$. The noise correlation matrix becomes

$$E[\mathbf{n}_k \mathbf{n}_{k-l}^T] = \begin{bmatrix} R_{nn}(l) & \ldots & R_{nn}(l+L_w) \\ \vdots & \ddots & \vdots \\ R_{nn}(l-L_w) & \ldots & R_{nn}(l) \end{bmatrix} \tag{4.4.3}$$

where $R_{nn}(l) = E[n(k)n(k-l)]$. The noise sequence $n(k)$ is assumed independent identically distributed (i.i.d.) as commonly assumed by researchers in this field [4], therefore, the matrix in (4.4.3) is a Toeplitz matrix with only one diagonal of nonzero entries depending upon the value of $l$, and hence becomes a shifting matrix. The matrices $E[\mathbf{x}_k \mathbf{n}_{k-l}^T] = \mathbf{0}$ and $E[\mathbf{n}_k \mathbf{x}_{k-1}^T] = \mathbf{0}$ since the signal $x(k)$ and the noise $n(k)$ are uncorrelated. If $2L_c < N$ holds, then the Toeplitz matrix $E[\mathbf{x}_k \mathbf{x}_{k-l}^T]$ has a shifting effect too. Now simplify (4.4.2) to yield [4]

$$\begin{aligned} R_{yy}(l) &= \sum_{j=0}^{L_c} c(j)c(j-l) + \sigma_v^2 \sum_{j=0}^{L_w} w(j)w(j-l) \\ &= R_{cc}(l) + \sigma_v^2 R_{ww}(l) \end{aligned} \tag{4.4.4}$$

So that the cost function in (4.4.1) can be approximated and denoted as $\hat{J}_{v+1}$

$$\begin{aligned} \hat{J}_{v+1} &= \sum_{l=v+1}^{L_c} |R_{yy}(l)| \\ &= \sum_{l=v+1}^{L_c} |R_{cc}(l) + \sigma_v^2 R_{ww}(l)| \end{aligned} \tag{4.4.5}$$

In most situations, the TEQ length $(L_w + 1)$ is shorter than the cyclic prefix length, $v$. In such situations, $R_{ww}(l)$ does not exist for the stated lags in (4.4.5). Even if the TEQ is larger than the cyclic prefix, the second term being added is very small due to its multiplication with $\sigma_v^2$. The noise variance $\sigma_n^2$ is usually small for ADSL channels [4]. Therefore, it is assumed that under practical scenarios, the hat on $J_{v+1}$ can be dropped so that $\hat{J}_{v+1} \cong J_{v+1}$. For this cost function an estimate of the length of the channel $\mathbf{h}$ is needed to determine $L_c = L_h + L_w$, which is known because the CSA test loops have nearly all of their energy in 200 consecutive taps [68]. The SAAM algorithm reaches the maximum shortening SNR (SSNR) solution of [3] under additive white Gaussian Noise (AWGN) condition, and is also robust to non-Gaussian impulsive noise environments [26].

## 4.5 Blind Adaptive Algorithm

The steepest gradient-descent algorithm to minimize $J_{v+1}$ is [26]

$$\begin{aligned} \mathbf{w} &= \mathbf{w}^{old} - \mu \nabla_{\mathbf{w}} \left( J_{v+1} \right) \\ &= \mathbf{w}^{old} - \mu \nabla_{\mathbf{w}} \left( \sum_{l=v+1}^{L_c} |E[y(k)y(k-l)]| \right) \end{aligned} \tag{4.5.1}$$

where $\mu$ is the step size and $\nabla_{\mathbf{w}}$ is the gradient evaluated at $\mathbf{w} = \mathbf{w}^{old}$. A moving average (MA) implementation is used to realize the instantaneous cost function

$$J_{v+1}^{inst}(k) = \sum_{l=v+1}^{L_c} \left| \sum_{n=kN_{avg}}^{(k+1)N_{avg}-1} \frac{y(n)y(n-l)}{N_{avg}} \right| \tag{4.5.2}$$

wherein $N_{avg}$ is a design parameter which determines a tradeoff between the algorithm complexity and a good estimate of the expectation. The steepest gradient-descent algorithm of Equation (4.5.1), therefore, can be written as a stochastic approximation as (4.5.3) which, using Equation (4.5.2), takes the form of Equation (4.5.4).

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mu \sum_{l=v+1}^{L_c} \left\{ sign\left( \sum_{n=kN_{avg}}^{(k+1)N_{avg}-1} \frac{y(n)y(n-l)}{N_{avg}} \right) \right\} \times \left\{ \nabla_{\mathbf{w}} \left( \sum_{n=kN_{avg}}^{(k+1)N_{avg}-1} \frac{y(n)y(n-l)}{N_{avg}} \right) \right\} \tag{4.5.3}$$

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mu \sum_{l=v+1}^{L_c} \left\{ sign\left( \sum_{n=kN_{avg}}^{(k+1)N_{avg}-1} \frac{y(n)y(n-l)}{N_{avg}} \right) \right\} \times \left\{ \left( \sum_{n=kN_{avg}}^{(k+1)N_{avg}-1} \frac{y(n)\mathbf{r}_{n-l} + y(n-l)\mathbf{r}_n}{N_{avg}} \right) \right\} \tag{4.5.4}$$

$$\mathbf{w}(k+1) = \frac{\mathbf{w}(k+1)}{\|\mathbf{w}(k+1)\|_2^2} \tag{4.5.5}$$

The function $sign(\cdot)$ is defined in Equation (4.2.2). To ensure that $\|\mathbf{w}\|_2^2 = 1$, the equalizer vector $\mathbf{w}$ has to be normalized at every iteration.

## 4.6 PUSAAM

As in any partial update algorithm, the aim of partial updating is to update a portion of the coefficients instead of the entire set of coefficients. The proposal here is to apply the partial update method to the SAAM algorithm and achieve the same performance whilst reducing the computational complexity, the proposed algorithms are called the DPUSAAM algorithm and the RPUSAAM algorithm.

### 4.6.1 DPUSAAM

In this algorithm the coefficients in the middle (in this simulation case, without loss of generality, there are eight) are updated $N_B - 1$ times, that is achieved by introducing a vector which contains ones in the middle and otherwise zeros, then at the $N_B^{th}$ time the outside ones are updated. The new vectors called "$mask_1$" and "$mask_2$" are created as $Mask_1 = [0000111111110000]$ $Mask_2 = [1111000000001111]$. Matrices $\mathbf{M}_i = diag(Mask_i)$, where $i = 1, 2$, are used in the update. The weight update of the DPUSAAM algorithm can therefore be written as

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mu \times \mathbf{M}_i \sum_{l=v+1}^{L_c} \left\{ sign \left( \sum_{n=kN_{avg}}^{(k+1)N_{avg}-1} \frac{y(n)y(n-l)}{N_{avg}} \right) \right\} \times \left\{ \sum_{n=kN_{avg}}^{(k+1)N_{avg}-1} \frac{y(n)\mathbf{r}_{n-l} + y(n-l)\mathbf{r}_n}{N_{avg}} \right\} \quad (4.6.1)$$

In this work $N_B = 5$, so that for $N_B - 1$ times $\mathbf{M}_i = \mathbf{M}_1$ otherwise $\mathbf{M}_i = \mathbf{M}_2$. Note, in this work the value of $L_w$ is fixed at 16 to be consistent with the work of [26]. The proposed algorithm achieves essentially the same performance as the SAAM algorithm in terms of higher bit rates and shortening the channel as will be shown in the simulation results. The advantage of the proposed algorithm is that it essentially achieves the same performance whilst updating only half of the coefficients at each iteration which implies less computational complexity at each iteration whilst retaining the advantage of the full length channel shortener. The overall complexity advantage is dependant however on the relative convergence time for the partial update algorithm as compared to the conventional adaptive algorithm. The sign function in 4.6.1 reduces the computational complexity of the implementation as compared to the SAM algorithm of [4].

### 4.6.2 RPUSAAM

The proposal here is to improve the deterministic partial update scheme to exploit improved convergence of random selection [36], and achieve performance close to SAAM. The set of indices of the coefficients of the adaptive filter is given by $\{1, 2, ..., L_w + 1\}$. This set is split into $P$ different disjoint but equal size subsets denoted $S_j$, $j = 1, ..., P$. Then, at each iteration one of these subsets is selected at random with probability $1/P$, and only those coefficients within the adaptive filter having indices from that subset are updated. The update equation can be written as in (4.6.2), where $R_i$ is a diagonal matrix with unity elements on the principle diagonal corresponding to the chosen subset $S_j$ and zeros elsewhere; and $\mathbf{w}(0)$ is initialized as for SAAM.

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mu \times R_i \sum_{l=v+1}^{L_c} \left\{ sign\left( \sum_{n=kN_{avg}}^{(k+1)N_{avg}-1} \frac{y(n)y(n-l)}{N_{avg}} \right) \right\} \times \left\{ \left( \sum_{n=kN_{avg}}^{(k+1)N_{avg}-1} \frac{y(n)\mathbf{r}_{n-l} + y(n-l)\mathbf{r}_n}{N_{avg}} \right) \right\} \quad (4.6.2)$$

Convergence analysis of these algorithms is extremely difficult due to the nonlinear dependence of the update equation on the weight vector, therefore performance assessment is made by simulation study. The complexities of the SAAM and PUSAAM algorithms have been calculated in Tables (4.1) and (4.2) respectively.

It is evident that by updating only eight coefficients i.e. $P = 2$ out of sixteen at every iteration as in DPUSAAM and RPUSAAM instead of the entire set as in SAAM, the computational complexity has reduced

| Steps | # multiplications | # additions/subtractions |
|---|---|---|
| (a)$N_{avg}$ times $y(n-l)\mathbf{r}_n$ | $N_{avg}.\{L_w+1\}$ | - |
| (b)$N_{avg}$ times $y(n)\mathbf{r}_{n-l}$ | $N_{avg}.\{L_w+1\}$ | - |
| (c)(a+b) | - | $N_{avg}.\{L_w+1\}$ |
| (d)$N_{avg}$ times $y(n)y(n-l)$ | $N_{avg}$ | - |
| (e)sum (d) outputs | - | $N_{avg}-1$ |
| (f)Sub-total for $(L_c-v)$ lags | $(L_c-v)\{N_{avg}(2L_w+3)\}$ | $(L_c-v)\{N_{avg}(L_w+2)-1\}$ |
| (g)$\mu\times$ output of (f) | $L_w+1$ | - |
| (h)$\mathbf{w}(k)$ - (g) | – | $L_w+1$ |
| (i)Total | $(L_c-v)\{N_{avg}(2L_w+3)\}+L_w+1$ | $(L_c-v)\{N_{avg}(L_w+2)-1\}+L_w+1$ |

**Table 4.1.** Number of multiplications and additions/subtractions required in the SAAM algorithm.

| Steps | # multiplications | # addition/subtractions |
|---|---|---|
| (a)$N_{avg}$ times $y(n-l)\mathbf{r}_n$ | $N_{avg}.\{L_w+1\}*$ | - |
| (b)$N_{avg}$ times $y(n)\mathbf{r}_{n-l}$ | $N_{avg}.\{L_w+1\}*$ | - |
| (c)(a+b) | - | $N_{avg}.\{L_w+1\}*$ |
| (d)$N_{avg}$ times $y(n)y(n-l)$ | $N_{avg}$ | - |
| (e)sum (d) outputs | - | $N_{avg}-1$ |
| (f)Sub-total for $(L_c-v)$ lags | $(L_c-v)\{N_{avg}(2L_w+3)\}*$ | $(L_c-v)\{N_{avg}(L_w+2)-1\}*$ |
| (g)$\mu\times R_k\times$ output of (f) | $(L_w+1)/2$ | - |
| (h)$\mathbf{w}(k)$ - (g) | – | $(L_w+1)/2$ |
| (i)Total | $(L_c-v)\{N_{avg}(2L_w+3)\}+((L_w+1)/2)$ | $(L_c-v)\{N_{avg}(L_w+2)-1\}+((L_w+1)/2)$ |

**Table 4.2.** Number of multiplications and additions/subtractions required in the PUSAAM algorithm, with $P=2$.

**In a practical realization of this algorithm these terms would have reduced complexity since the final multiplication by the vector in DPUSAAM or matrix in RPUSAAM by zero elements implies that calculation of certain quantities is redundant.*

to a half of the complexity as compared to the SAAM algorithm. Other levels of complexity reduction could be achieved with different settings for $P$.

## 4.7 Simulation Results

The Matlab code available at [69] was extended to simulate the DPUSAAM, and RPUSAAM algorithms in impulsive noise environments. The cyclic prefix had length 32, the FFT size $N_{fft} = 512$, the TEQ had 16 taps and the channel was the test ADSL channel CSA loop1 available at [68]. For simulations in $\alpha$-stable noise, the geometric-SNR (G-SNR) definition is used instead of the standard SNR definition, due to infinite variance of the S$\alpha$S distribution [66], a total of 75 OFDM symbols was used. The step size used was 0.0007, carefully chosen empirically to give best shortening performance. Importantly, the step-size for the partial update algorithm can be chosen to ensure that there is an overall complexity advantage for the partial update scheme, i.e. a larger step-size can ensure fast convergence of the algorithm. In this work, however, this issue was not considered. The dispersion of the noise for a given value of $\alpha$ is changed and the achievable bit rate is calculated. In Figures (4.4) and (4.5) the impulsive response of the original and the shortened channel with different values of $\alpha$ ($\alpha$=1.95, and 1.9, these values are between the less impulsive case of the Gaussian noise when $\alpha$=2 and the more impulsive Cauchy case when $\alpha$=1 [70], these values were chosen to show the robustness of the proposed algorithms) show that all of the algorithms are confirmed to be effective. The shortened channel has a length at least reduced by a factor of 4 as the original channel. In Figures (4.6) and (4.7) the impulse response of the original and the shortened channel for the average of eight different channels show that all the algorithms perform similarly with different channels. Such averaging is feasible as all of the channels have similar positive decay profiles and therefore the overall shortening performance is not lost in this process.

Quasi, since rigorously SNR does not exist, achievable bits per second as a function of the averaging block number are plotted at $\alpha$=1.95 and 1.9 and are shown in Figures (4.8) and (4.9), from which it can be seen that the proposed algorithms are as robust to the impulsive noise as the SAAM algorithm with half of the coefficients being updated. Careful inspection of Figures (4.8) and (4.9) reveals the improved final performance of the random update selection scheme. Note, that as shown by [15], the error performance surface for SAM-type algorithms is multimodel and the minimum of the SAM-type costs is not generally coincident with the minimum of the achievable bit rate. This observation explains the asymptotic behaviour in the figures. For the case of Gaussian-mixture modelling, the signal to Gaussian noise power was such that $\sigma_x^2\|\mathbf{h}\|^2/\sigma_v^2 = 40$ dB. This is a typical value of SNR in ADSL environments [26]. 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 $\Gamma$ is a function of a chosen probability of symbol error and the line code and is given by

$$\Gamma = \Gamma_{gap} + \gamma_m - \gamma_c \tag{4.7.1}$$

and is assumed for the system to be constant over all subchannels. This gap measures efficiency of the transmission method with respect to best

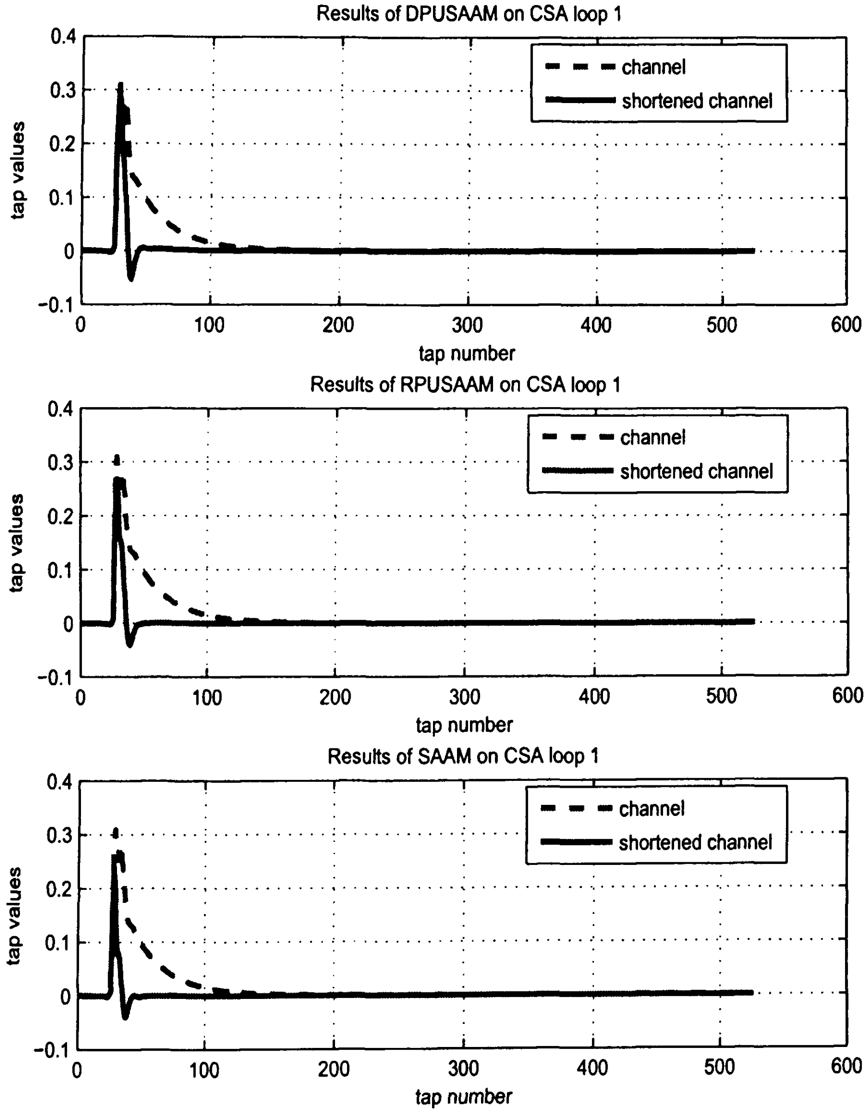

**Figure 4.4.** Original and the shortened channel in $\alpha$-stable noise environment with alpha=1.95.

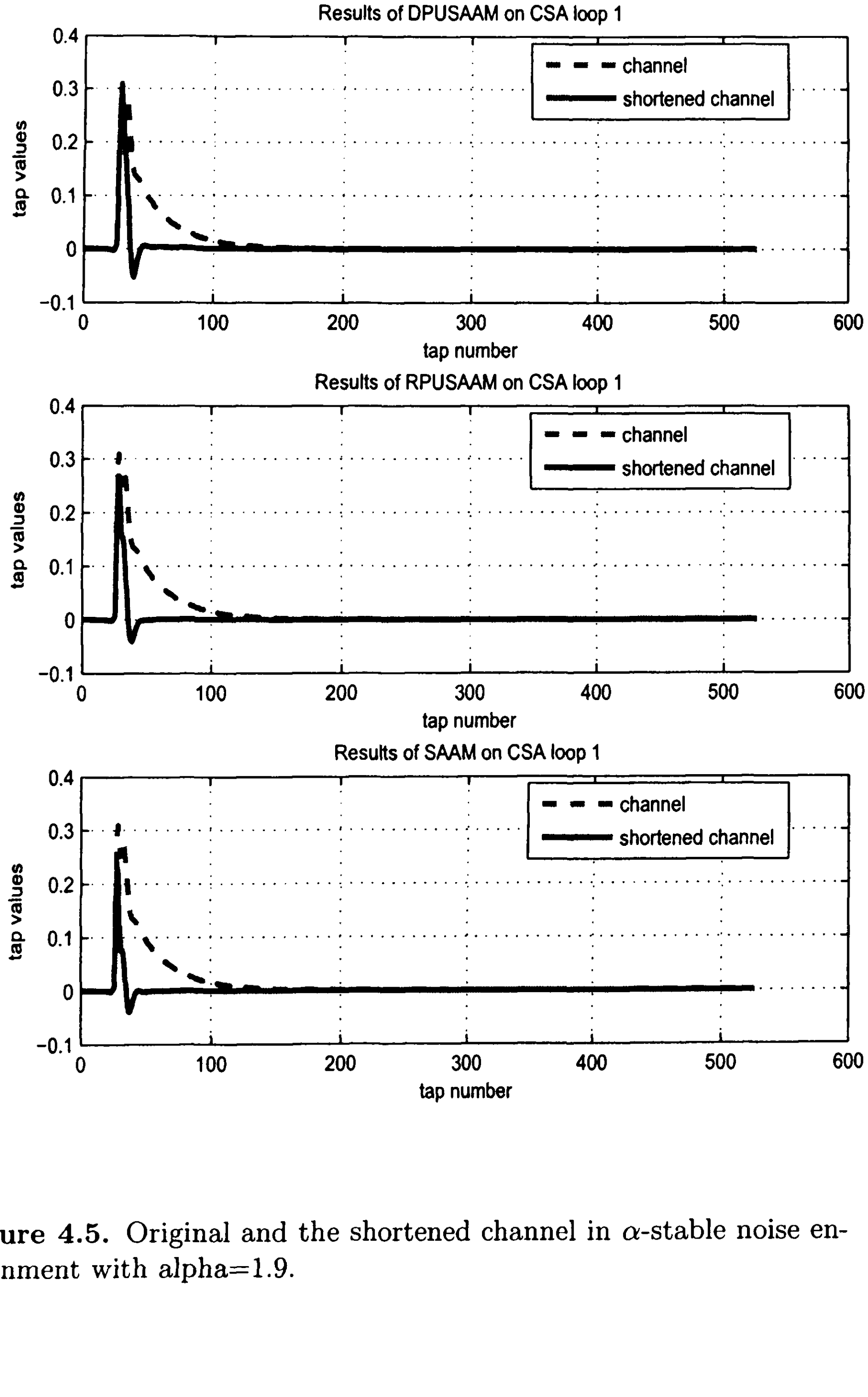

**Figure 4.5.** Original and the shortened channel in $\alpha$-stable noise environment with alpha=1.9.

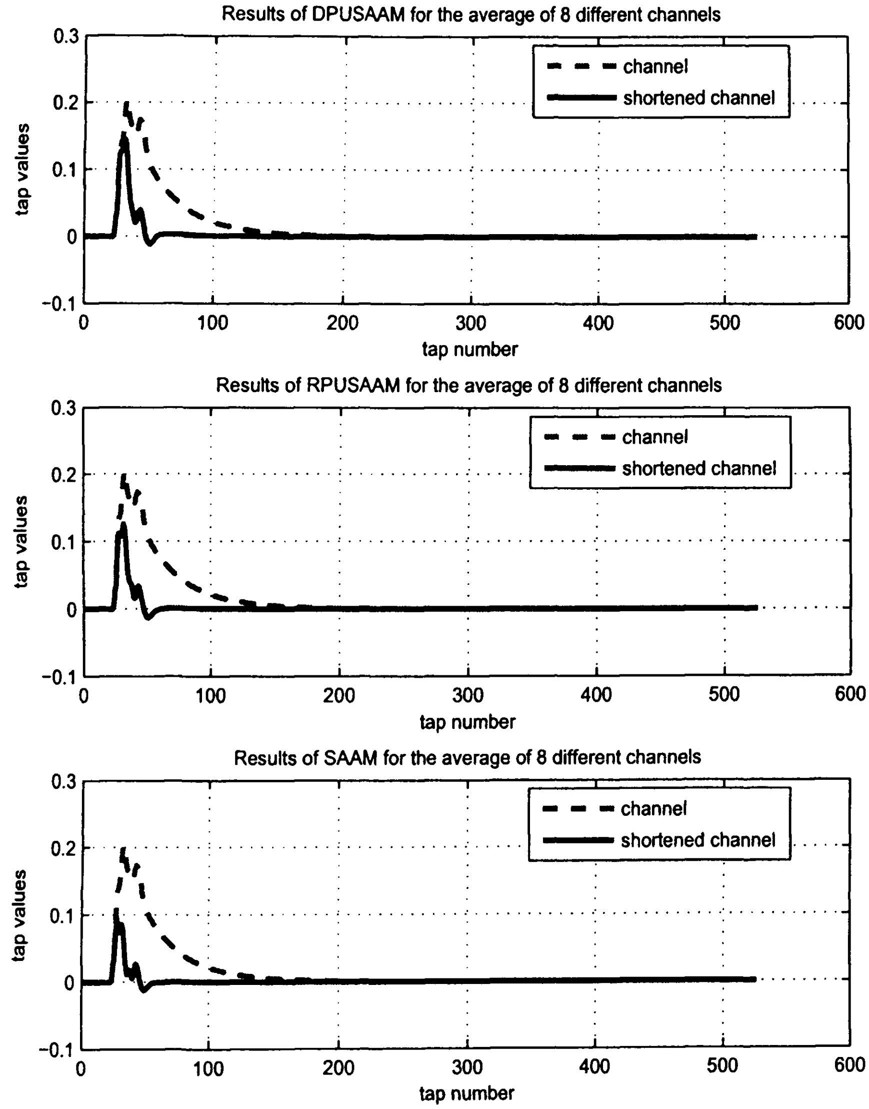

Figure 4.6. Original and the shortened channel in $\alpha$-stable noise environment with alpha=1.95 for the average of eight CSA channels.

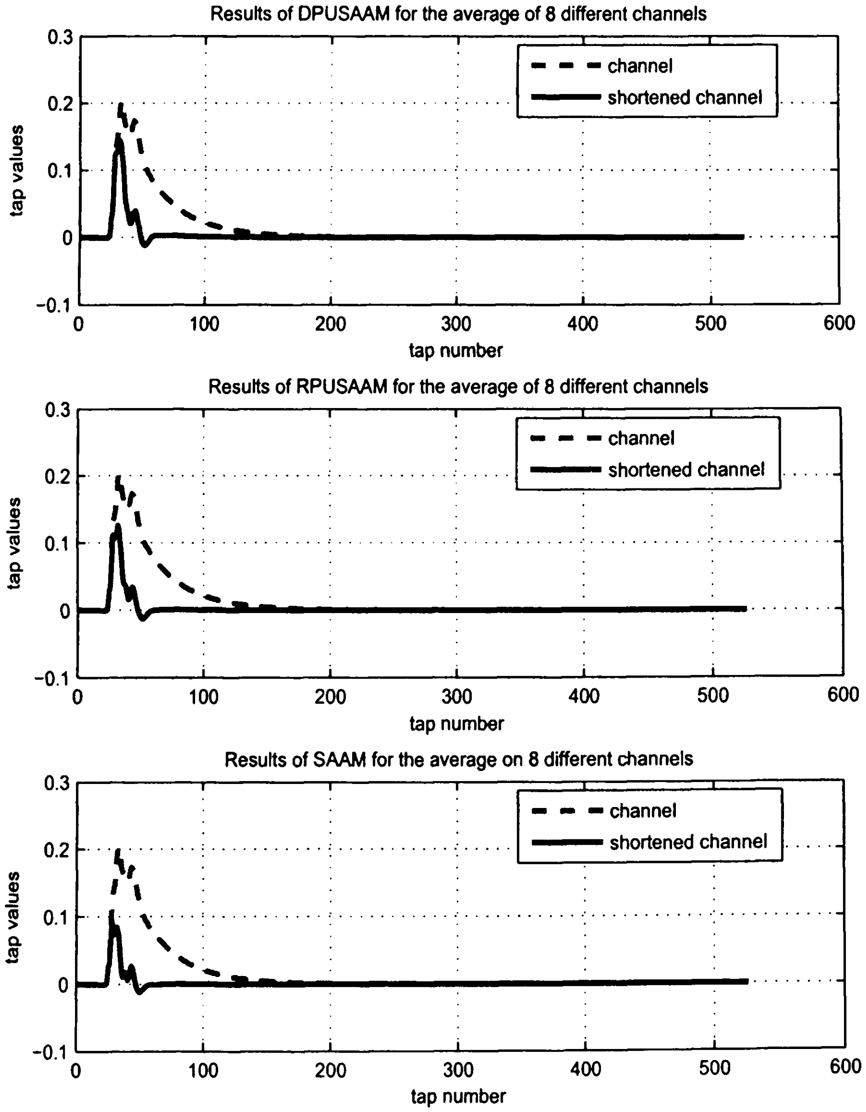

**Figure 4.7.** Original and the shortened channel in $\alpha$-stable noise environment with alpha=1.9 for the average of eight CSA channels.

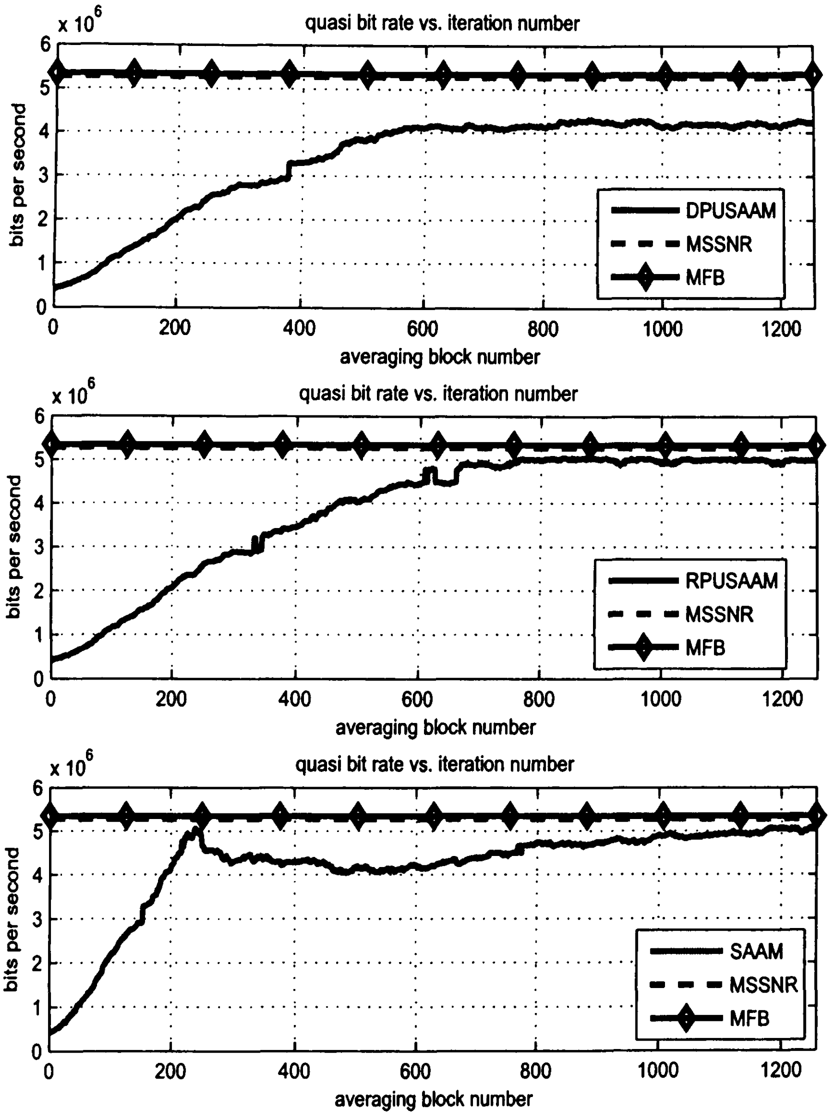

**Figure 4.8.** Quasi achievable bit rate versus averaging block number in $\alpha$-stable noise environment with alpha=1.95.

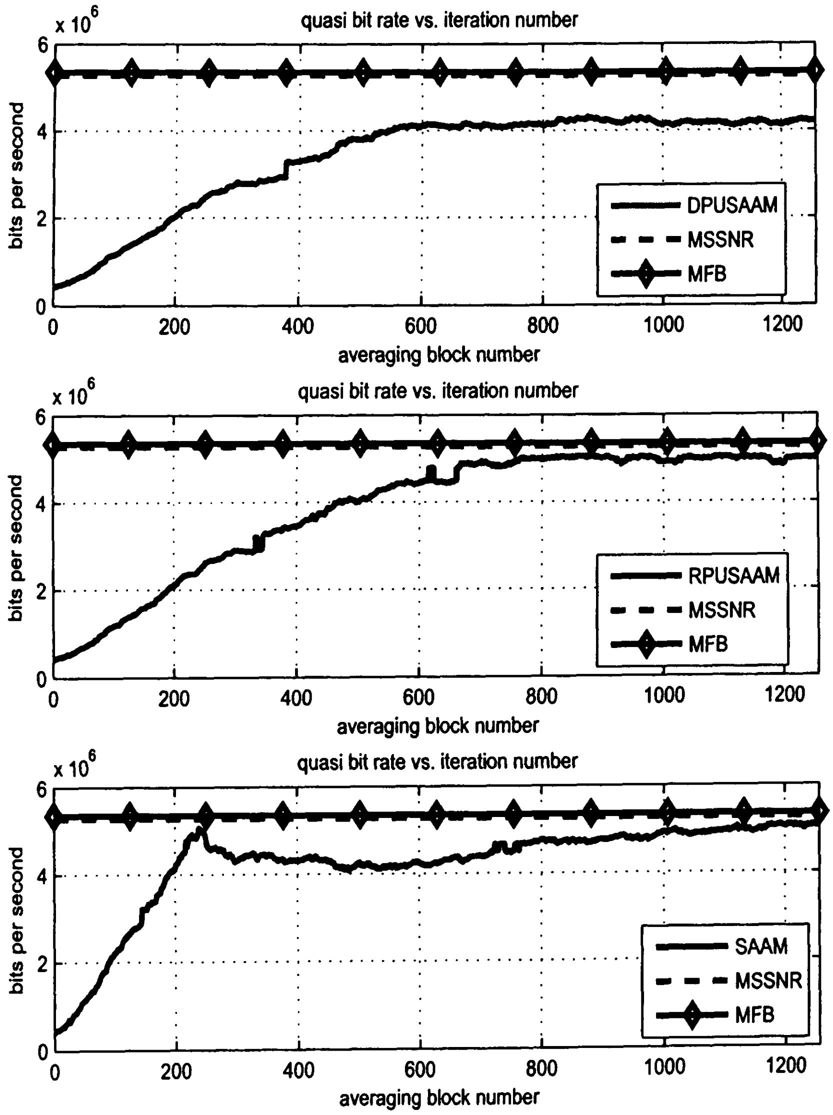

**Figure 4.9.** Quasi achievable bit rate versus averaging block number in $\alpha$-stable noise environment with alpha=1.9.

possible performance on an additive white Gaussian noise channel. The bit rate on each subcarrier is determined using the 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 QAM modulation is used across the subcarriers. The bit rate on each subcarrier $i$ is calculated based on

$$b_i = \log_2\left(1 + 10^{((SNR_i - \Gamma)/10)}\right) \tag{4.7.2}$$

The subchannel SNR, $SNR_i$ in (4.7.2) is found by using the subchannel SNR model described in (4.7.3) and includes the channel noise as well as the distortion due to ICI and ISI caused by the energy of the channel outside the $v + 1$ length. This definition can be used to assess the performance of the TEQ algorithms although it is only an approximation in an impulsive noise environment. To use this model, the maximal energy point of the shortened channel is used as the starting index of the $v + 1$ length window of the desired channel.

$$SNR_i = \frac{S_{x,i}|H_i^{signal}|^2}{S_{n,i}|H_i^{noise}|^2 + S_{x,i}|H_i^{ISI}|^2} \tag{4.7.3}$$

The bit rate is then computed with the formula

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

where $F_s = 2.208$ MHz is the sampling frequency. SAAM, DPUSAAM, RPUSAAM, and the maximum SSNR algorithm (MSSNR) of [3] are simulated. The step size used for the adaptive algorithms is 0.0007, empirically chosen to give the best performance.

The impulse response of the original and the shortened channel for SAAM, DPUSAAM, and RPUSAAM with Gaussian-mixture noise are shown in Figure (4.10), it shows that all the algorithms are confirmed to be effective. In Figure (4.11) the impulse response of the original and the shortened channel for SAAM, DPUSAAM, and RPUSAAM for the average of eight different channels shows that all the algorithms perform similarly with different channels. The effect of impulsive noise on the quasi-achievable bit rate as a function of the averaging block number is shown in Figure (4.12), it can be seen that the proposed algorithms are as robust to the impulsive noise as the SAAM algorithm with only half of the coefficients being updated. Importantly, robustness is shown both to alpha-stable and Gaussian-mixture noise as the results in Figures (4.10) and (4.11) are very similar to those in Figures (4.4) and (4.5).

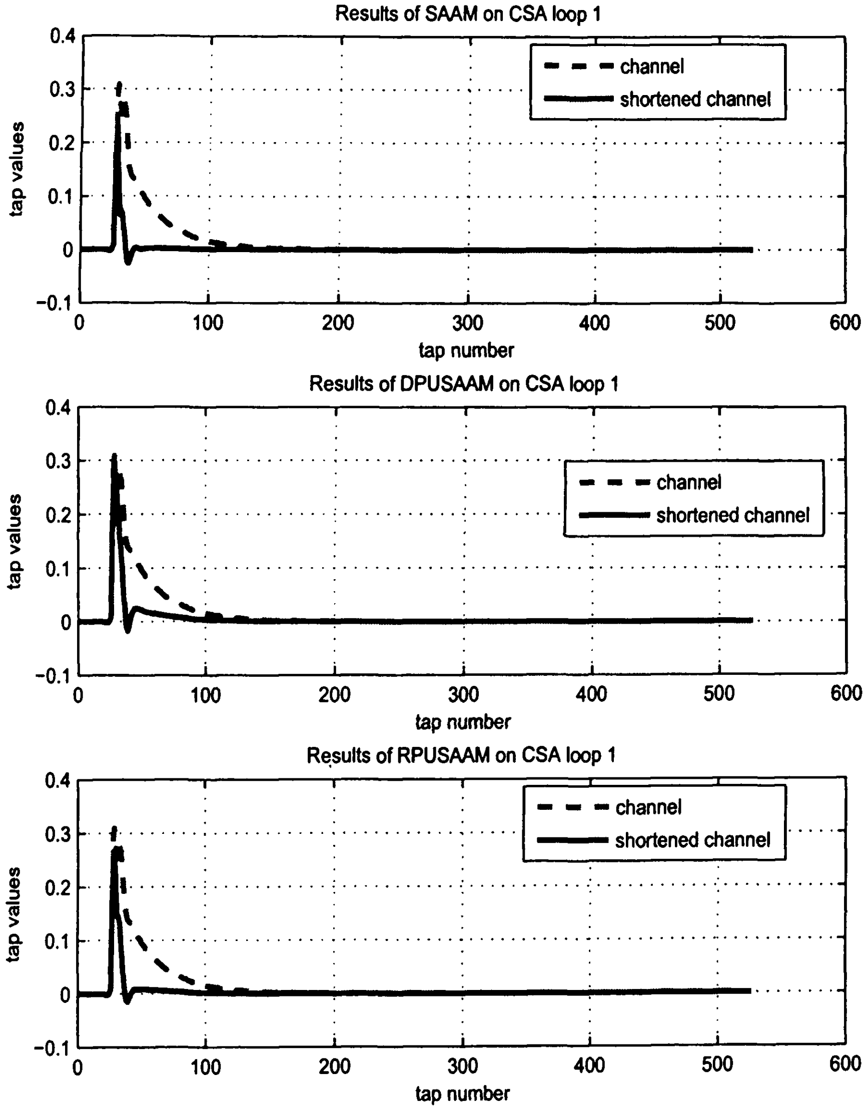

**Figure 4.10.** Original and the shortened channel for Gaussian mixture for p=0.001 and d=100

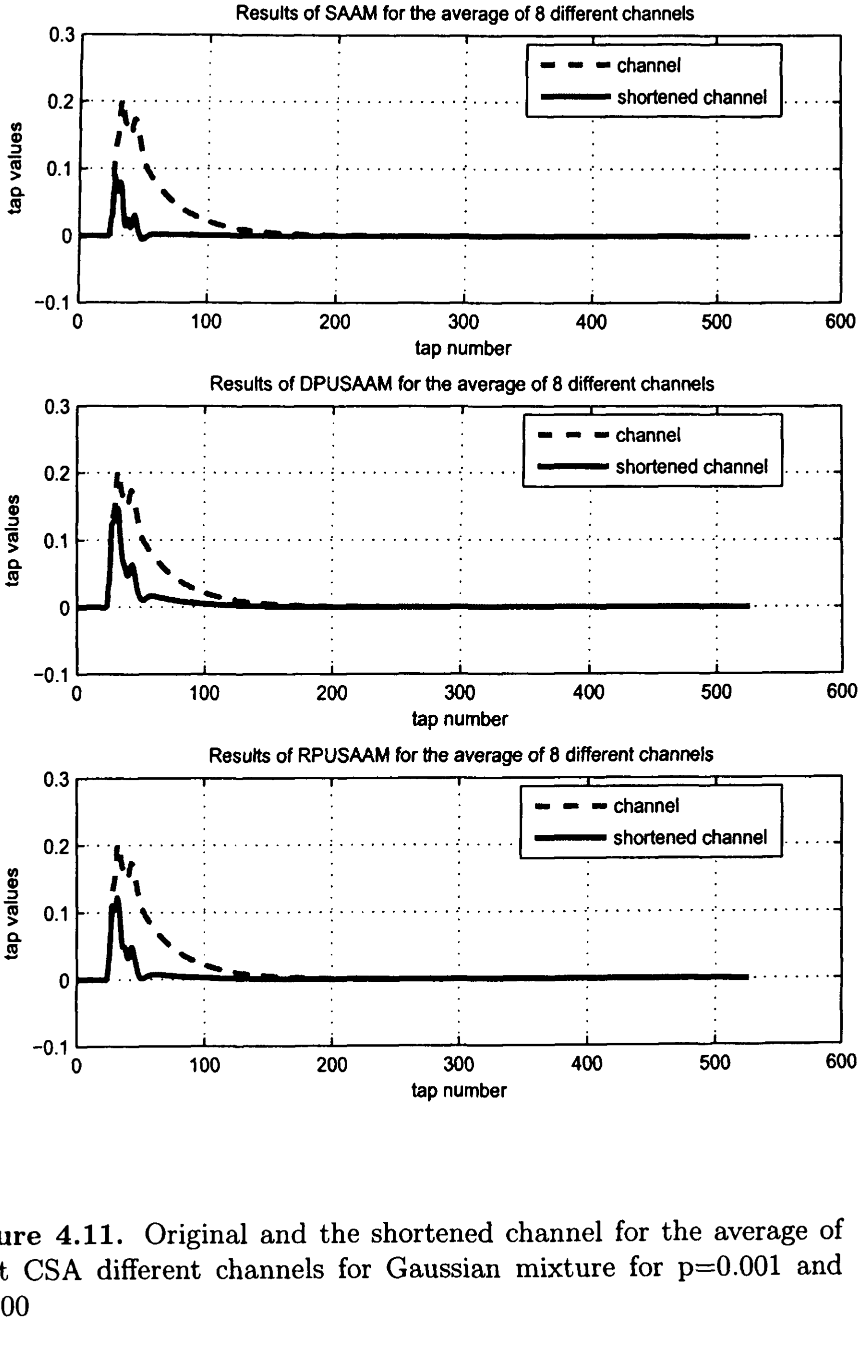

Figure 4.11. Original and the shortened channel for the average of eight CSA different channels for Gaussian mixture for p=0.001 and d=100

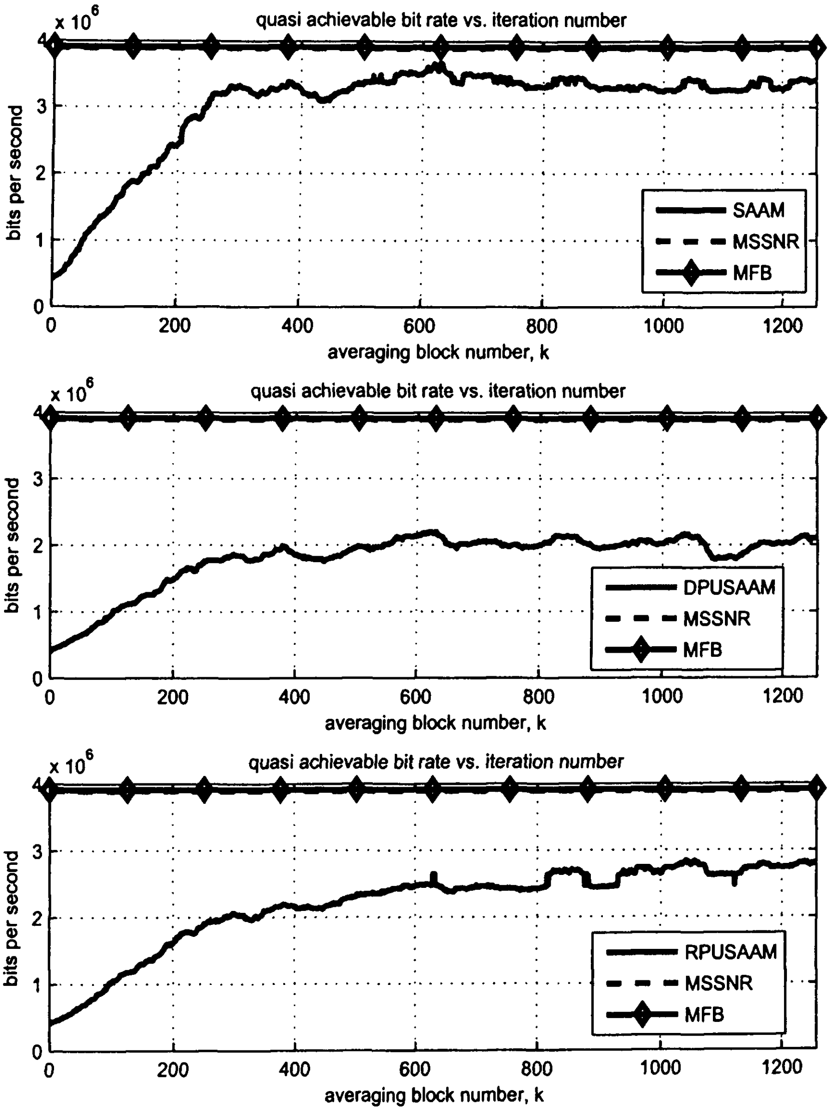

Figure 4.12. Quasi achievable bit rate versus averaging block number

## 4.8 Conclusions

The newly proposed algorithms DPUSAAM and RPUSAAM have been shown to be robust for operations in environments with a range of degrees of impulsiveness. The computational complexities of DPUSAAM and RPUSAAM are also considerably lower than SAAM. Simulation results show that DPUSAAM and RPUSAAM algorithms approach the maximum shortening signal-to-noise ratio (MSSNR) solution [3] in Gaussian noise. The DPUSAAM and RPUSAAM algorithms are also robust to additive white non-Gaussian noise.

Chapter 5

# DETERMINISTIC COEFFICIENT SELECTION IN PARTIAL UPDATE BLIND CHANNEL SHORTENING ALGORITHMS

The SAM algorithm [4] 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. The SLAM algorithm [5], 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. In this chapter, the partial update method is applied to the two channel shortening algorithms which achieve the same performance whilst further reducing the computational complexity, the proposed algorithms are called the partial update SAM algorithm (PUSAM) and partial update SLAM algorithm (PUSLAM). These algorithms essentially achieve the same result in terms of reducing the effective channel length as SAM and SLAM with half the complexity. The performance advantage of the PUSAM and PUSLAM algorithms is shown on eight different carrier serving area test loops (CSA) channels and comparisons are made with the original SAM and the SLAM algorithms.

## 5.1 System Model

The system model is shown in Figure (3.2). The input signal $x(k)$ is the source sequence to be transmitted through a linear finite-impulse-response (FIR) channel $\mathbf{h}$ of length $(L_h + 1)$ taps, $r(k)$ is the received signal, which will be filtered through an $(L_w + 1)$-tap TEQ with an impulse response vector $\mathbf{w}$ to obtain the output sequence $y(k)$. For convenience in this work real signals are assumed but generalization to the complex case is straight-forward. Denote $\mathbf{c} = \mathbf{h} * \mathbf{w}$ as the shortened or effective channel assuming $\mathbf{w}$ is in steady-state where $*$ denotes discrete time convolution. It is also assumed that $2L_c < N_{fft}$ holds, where $L_c$ is the order of the effective channel and $N_{fft}$ is the FFT size [4]. The signal $n(k)$ is a real zero-mean, independent identically distributed (i.i.d.), noise sequence, uncorrelated with the source sequence with variance $\sigma_n^2$. The received sequence $r(k)$ is

$$r(k) = \sum_{j=0}^{L_h} h(j)x(k-j) + n(k) \tag{5.1.1}$$

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

$$y(k) = \sum_{j=0}^{L_w} w(j)r(k-j) = \mathbf{w}^T\mathbf{r}_k \tag{5.1.2}$$

where $\mathbf{r}_k = [r(k) \quad r(k-1) \cdots r(k-L_w)]^T$ and $\mathbf{w}$ is the impulse response vector of the TEQ $\mathbf{w} = [w_0 \quad w_1 \quad w_2 \cdots w_{L_w}]^T$.

## 5.2 Partial Update SAM Algorithm

For the effective channel $\mathbf{c}$ to have zero taps outside a contiguous window of size $(v + 1)$, its autocorrelation values should be zero outside a window of size $2v + 1$. The autocorrelation sequence of the effective channel is given by

$$R_{cc}(l) = \sum_{j=0}^{L_c} c(j)c(j-l)$$

and for a shortened channel, it must satisfy

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

Therefore, a cost function, $J_{pusam}$, which is the same as $J_{sam}$, based upon minimizing the sum of the square values of the auto-correlation of the effective channel is suggested, i.e.,

$$J_{pusam} = \sum_{l=v+1}^{L_c} (R_{cc}(l))^2 \tag{5.2.1}$$

The trivial solution can be avoided by imposing a norm constraint on the TEQ i.e., $\|\mathbf{w}\|_2^2 = 1$. The optimization problem can then be stated as in [4]

$$\mathbf{w}^{opt} = \arg_{\mathbf{w}} \min_{\|\mathbf{w}\|_2^2=1} J_{pusam}$$

The autocorrelation sequence of the output $y(k)$ of the TEQ is related to the autocorrelation sequence of the effective channel as

$$R_{yy}(l) = R_{cc}(l) + \sigma_n^2 R_{ww}(l) \tag{5.2.2}$$

An approximation to the cost function in (5.2.1), denoted by $\hat{J}_{pusam}$ is given by

$$\begin{aligned}\hat{J}_{pusam} &= \sum_{l=v+1}^{L_c} (R_{yy}(l))^2, \\ &= \sum_{l=v+1}^{L_c} (R_{cc}(l))^2 + 2\sigma_n^2 R_{cc}(l) R_{ww}(l) \\ &\quad + \sigma_n^4 (R_{ww}(l))^2 \end{aligned} \tag{5.2.3}$$

In most situations, the TEQ length $(L_w + 1)$ is shorter than the cyclic prefix length, $v$. In this case, $R_{ww}(l)$ does not exist for the stated lag in (5.2.3). Therefore, both the noise terms in (5.2.3) can be neglected. Even if the TEQ is longer than the cyclic prefix, the second and third terms being added are very small due to their multiplication with $\sigma_n^2$ and $\sigma_n^4$. The noise variance $\sigma_n^2$ is usually small for ADSL channels [4]. Therefore, practically it is assumed that $\hat{J}_{pusam} \cong J_{pusam}$ as in (5.2.1).

### 5.2.1 Adaptive Algorithm

The steepest gradient-descent algorithm to minimize the PUSAM cost $J_{pusam}$ is

$$\mathbf{w} = \mathbf{w}^{old} - \mu \nabla_{\mathbf{w}} \left( \sum_{l=v+1}^{L_c} E[y(k)y(k-l)]^2 \right) \tag{5.2.4}$$

where $\mu$ is the step size and $\nabla_{\mathbf{w}}$ is the gradient evaluated at $\mathbf{w} = \mathbf{w}^{old}$. The instantaneous cost function is defined, where the expectation operation is replaced by a moving average over a user-defined window of length $N_{avg}$

$$J_{v+1}^{inst}(k)=\sum_{l=v+1}^{L_c}\left\{\sum_{n=kN_{avg}}^{(k+1)N_{avg}-1}\frac{y(n)y(n-l)}{N_{avg}}\right\}^2 \qquad (5.2.5)$$

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

$$\mathbf{w}(k+1)=\mathbf{w}(k)-2\mu\times\mathbf{M}(j)\times\sum_{l=v+1}^{L_c}\left\{\sum_{n=kN_{avg}}^{(k+1)N_{avg}-1}\frac{y(n)y(n-l)}{N_{avg}}\right\}$$
$$\times\left\{\nabla_{\mathbf{w}}\left(\sum_{n=kN_{avg}}^{(k+1)N_{avg}-1}\frac{y(n)y(n-l)}{N_{avg}}\right)\right\} \qquad (5.2.6)$$

where $\mathbf{M}(j)$ is a matrix which is equal to $diag(mask_j)$, (5.2.6) can be simplified to

$$\mathbf{w}(k+1)=\mathbf{w}(k)-2\mu\times\mathbf{M}(i)\times\sum_{l=v+1}^{L_c}\left\{\sum_{n=kN_{avg}}^{(k+1)N_{avg}-1}\frac{y(n)y(n-l)}{N_{avg}}\right\}$$
$$\times\left\{\sum_{n=kN_{avg}}^{(k+1)N_{avg}-1}\left(\frac{y(n)\mathbf{r}_{n-l}+y(n-l)\mathbf{r}(n)}{N_{avg}}\right)\right\} \qquad (5.2.7)$$

which takes the same form as (4.6.1) except for the $sign(.)$ function. In the PUSAM algorithm the coefficients in the middle (in the simulation case studied eight will be in the middle), that is achieved by introducing a vector which contains ones in the middle and zeros outside the middle, are updated 4 times then at the fifth time the outside ones are updated. The new vectors called Maskl and Mask2 are created as Maskl = [0000111111110000] and Mask2 = [1111000000001111]. The matrices $\mathbf{M}(i) = diag(Mask_i)$ are defined, where $i = 1, 2$, which is follows the same approach as used in the DPUSAAM algorithm, as in section (4.6.1). The partial-update SAM (PUSAM) algorithm can therefore be written as shown in (5.2.7).

In this work if $k \ mod \ 5 \neq 0$ then $\mathbf{M}(i) = \mathbf{M}(1)$ else $\mathbf{M}(i) = \mathbf{M}(2)$. Other choices of mask and update cycle period 5 are possible but in this chapter the focus is to demonstrate the basic concept.

## 5.3 Partial Update Slam Algorithm

For the effective channel $\mathbf{c}$ to have zero taps outside a contiguous window of size $(v + 1)$, its autocorrelation values should be zero outside a window of size $2v + 1$. The autocorrelation sequence of the effective channel is given by

$$R_{cc}(l) = \sum_{j=0}^{L_c} c(j)c(j - l) \tag{5.3.1}$$

and for a shortened channel, it must satisfy

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

Then the cost function $J_{puslam}$ is defined based upon minimizing the squared auto-correlation of the effective channel at lag $l = v + 1$, i.e.,

$$J_{puslam} = R_{cc}(l)^2, l = v + 1 \tag{5.3.2}$$

### 5.3.1 Adaptive Algorithm

The steepest gradient-descent algorithm to minimize the PUSLAM cost $J_{v+1}$ is

$$\mathbf{w} = \mathbf{w}^{old} - \mu\nabla_{\mathbf{w}}(E[y(k)y(k-l)])^2 \tag{5.3.3}$$

where $l$ is a single lag, $\mu$ is the step size and $\nabla_{\mathbf{w}}$ is the gradient evaluated at $\mathbf{w} = \mathbf{w}^{old}$. The instantaneous cost function is defined, where the expectation operation is replaced by a moving average over a user-defined window of length $N_{avg}$

$$J_{v+1}^{inst}(k) = \left\{ \sum_{n=kN_{avg}}^{(k+1)N_{avg}-1} \frac{y(n)y(n-l)}{N_{avg}} \right\}^2 \tag{5.3.4}$$

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

$$\begin{aligned}\mathbf{w}(k+1) = \mathbf{w}(k) - 2\mu \times \mathbf{M}(i) &\left\{ \sum_{n=kN_{avg}}^{(k+1)N_{avg}-1} \frac{y(n)y(n-l)}{N_{avg}} \right\} \\ \times &\left\{ \nabla_{\mathbf{w}} \left( \sum_{n=kN_{avg}}^{(k+1)N_{avg}-1} \frac{y(n)y(n-l)}{N_{avg}} \right) \right\}\end{aligned} \tag{5.3.5}$$

which can be simplified to

$$\mathbf{w}(k+1) = \mathbf{w}(k) - 2\mu \times \mathbf{M}(i) \left\{ \sum_{n=kN_{avg}}^{(k+1)N_{avg}-1} \frac{y(n)y(n-l)}{N_{avg}} \right\} \times \left\{ \sum_{n=kN_{avg}}^{(k+1)N_{avg}-1} \left( \frac{y(n)\mathbf{r}_{n-l} + y(n-l)\mathbf{r}(n)}{N_{avg}} \right) \right\} \tag{5.3.6}$$

In the PUSLAM algorithm the coefficients in the middle (in the simulation case studied eight will be the middle), that is achieved by introducing a vector which contains ones in the middle and zeros outside the middle, are updated 4 times then at the fifth time the outside ones are updated. The new vectors called Maskl and Mask2 are also created as Maskl = [0000111111110000] and Mask2 = [1111000000001111]. The matrices $\mathbf{M}(i) = diag(Mask_i)$ are defined, where $i = 1, 2$. The partial-update SLAM (PUSLAM) algorithm can therefore be written as shown in (5.3.6). the same strategy for selection of $\mathbf{M}_i$ is used as in PUSAM. The calculations shown in Table (5.1) clearly shows the implementation advantage of the PUSAM and PUSLAM algorithms, it is clear that the partial update algorithms PUSAM and PUSLAM have reduced the computational complexity for SAM and SLAM by half and that what was aimed to achieve. Due to the difficulty to formally analyse these algorithms their performance is assessed by simulations.

| Algorithms | # multiplications | # addition | # subtractions |
|---|---|---|---|
| SAM | $3NL(L_c - v)$ | $3NL(L_c - v)$ | 1 |
| SLAM | $3NL$ | $3NL$ | 1 |
| PUSAM | $3NL(L_c - v)/2$ | $3NL(L_c - v)/2$ | 1 |
| PUSLAM | $3NL/2$ | $3NL/2$ | 1 |

**Table 5.1.** The total number of multiplications, additions and subtractions, comparison between SAM, SLAM, PUSAM and PUSLAM.

## 5.4 Simulations

The Matlab code at [69] was extended to simulate PUSAM and PUSLAM. The cyclic prefix was of length 32, the FFT size $N_{fft} = 512$, the TEQ had 16 taps and the channel was the test ADSL channel CSA loop 1 available at [68]. The noise was set such that $\sigma_x^2 \|c\|^2 / \sigma_n^2 = 40\ dB$ where $\|.\|$ denotes the Euclidean norm; and 75 OFDM symbols were used. The step size used for PUSAM was 5 and for PUSLAM was 600. To make fair comparison between PUSAM and SAM, and between PUSLAM and SLAM, all the parameters are kept the same as in [5]. All algorithms are compared with the maximum shortening SNR solution [3], which was obtained using the code at [69], and the matched filter bound (MFB) on capacity, which assumes no ICI. The bit rate on each subcarrier is determined using noise margin $\gamma_m$ = 6dB and the coding gain $\gamma_c$ = 4.2dB. The value of $\Gamma_{gap}$ = 9.8dB is used which corresponds to a probability of error $10^{-7}$ and the QAM modulation used across the subcarriers. The SNR gap $\Gamma$ is given by

$$\Gamma = \Gamma_{gap} + \gamma_m - \gamma_c \tag{5.4.1}$$

The bit rate on each subcarrier $i$ is calculated based on

$$b_i = \log_2 \left(1 + 10^{((SNR_i - \Gamma)/10)}\right) \tag{5.4.2}$$

The bit rate was determined based on

$$R = \sum_{i=1}^{N_{fft}} \log_2(1 + SNR_i/\Gamma)$$

The remainder of the explanation relates to the figures mentioned individually. In Figures (5.1), (5.2), (5.3) and (5.4), the shortened channels are compared with the original channels and all algorithms are confirmed to be effective. The support of the shortened channel is restricted to lie within the first 50 taps. Figures (5.5) and (5.6) show the 16-tap TEQ designed after the PUSAM and the PUSLAM algorithms converge. In Figures (5.7), (5.8), (5.9) and (5.10), the achievable bits per second [15] as a function of the averaging block number, k, are plotted which show the convergence property of PUSAM and PUSLAM, best performance is achieved at approximately 900 blocks. Decrease in the bit rates after achieving the peak bit rates is clear for the SAM, SLAM, PUSAM and PUSLAM (with more blocks used to see converged behaviour) algorithms and as mentioned previously is due to the multimodular nature of the cost function and the non consistency between SAM-type costs and the achievable bit error rate. Figures (5.11) and (5.12) show the PUSAM and the PUSLAM cost versus the iteration number. The PUSAM and the PUSLAM cost function and the bit rate are a smooth function of each other i.e., the PUSAM and the PUSLAM minima and the bit rate maxima appear to be located in close proximity. All the results in these plots were for an SNR=40 dB. Figures (5.13) and (5.14) show the average performance of PUSAM and PUSLAM in term of shortening the channel for eight different CSA channels to make sure that the algorithms perform similarly with different channels.

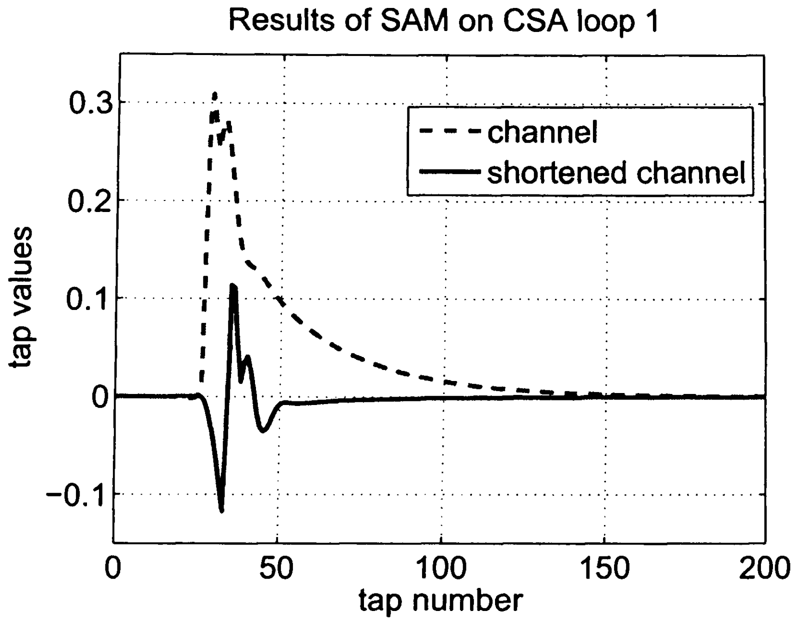

Figure 5.1. Channel (dashed) and shortened channel (solid) impulse response of SAM algorithm.

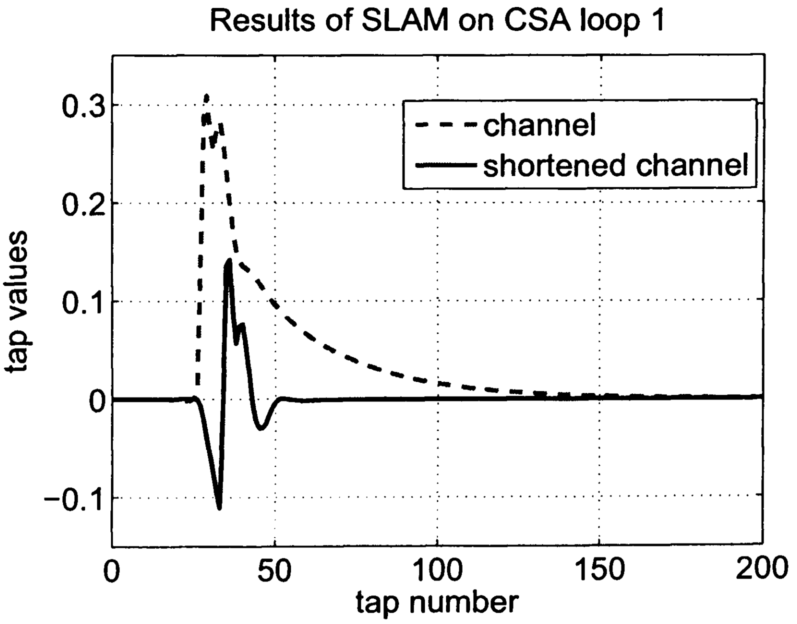

Figure 5.2. Channel (dashed) and shortened channel (solid) impulse response of SLAM algorithm.

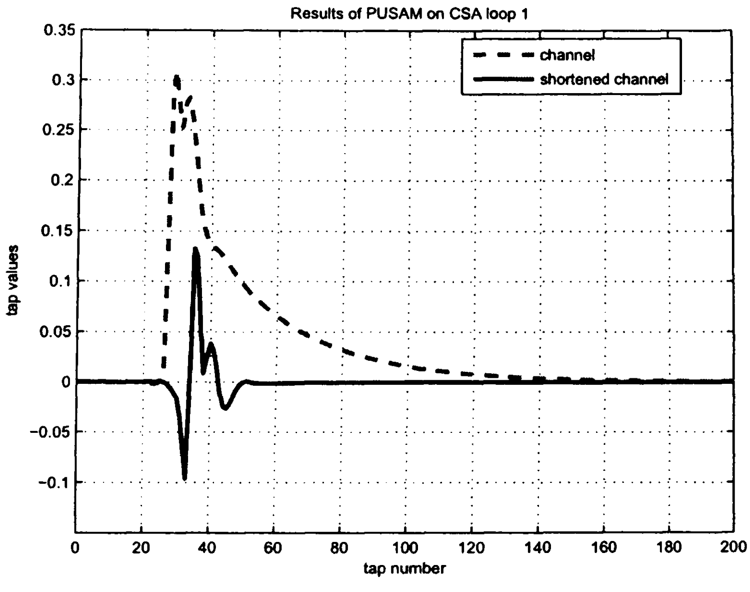

Figure 5.3. Channel (dashed) and shortened channel (solid) impulse response of PUSAM algorithm.

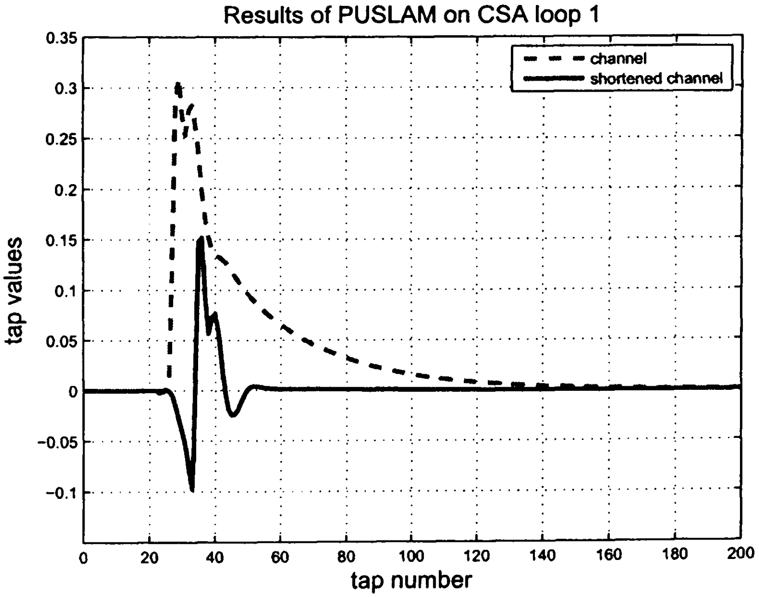

Figure 5.4. Channel (dashed) and shortened channel (solid) impulse response of PUSLAM algorithm.

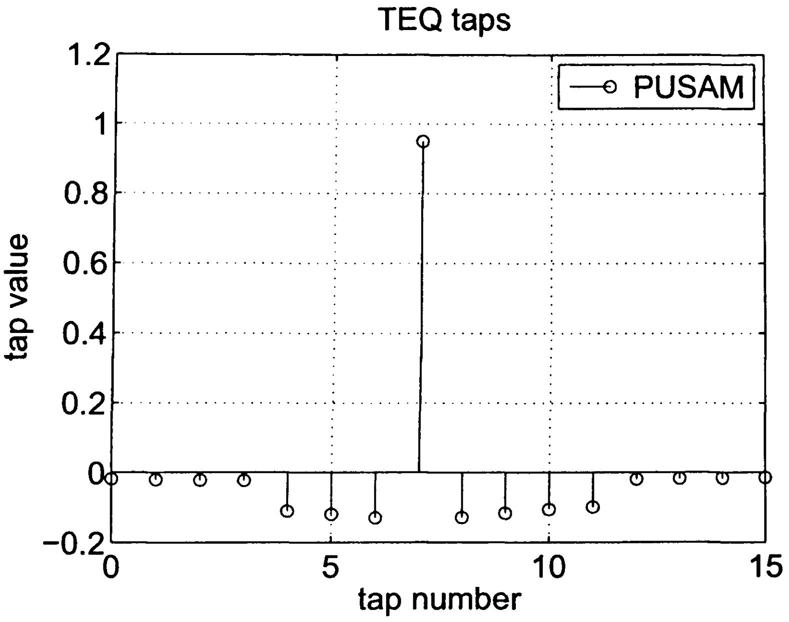

Figure 5.5. TEQ taps.

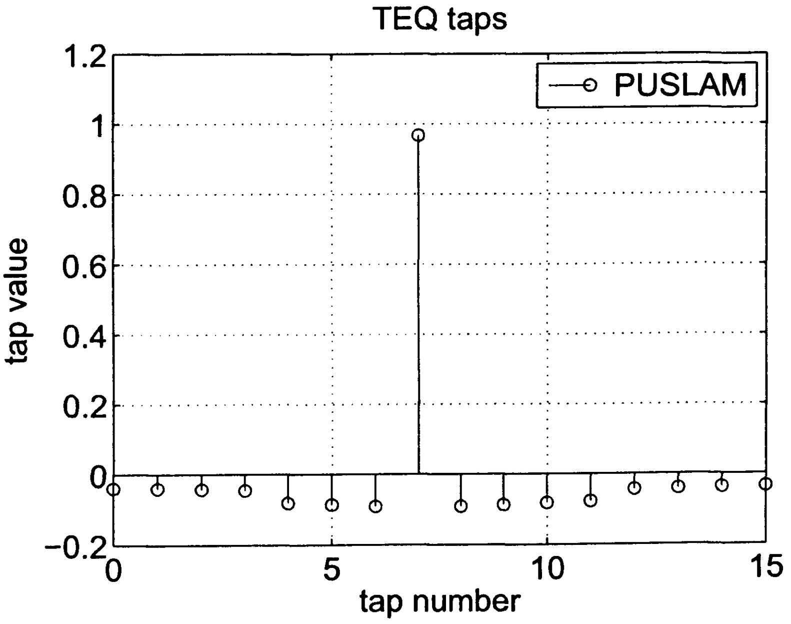

Figure 5.6. TEQ taps.

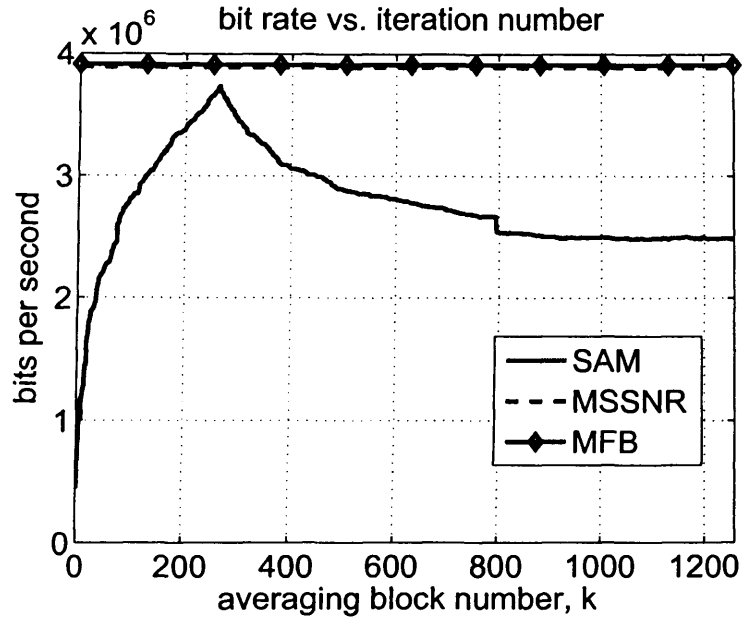

**Figure 5.7.** Achievable bit rate versus iteration number at 40 dB SNR of SAM algorithm.

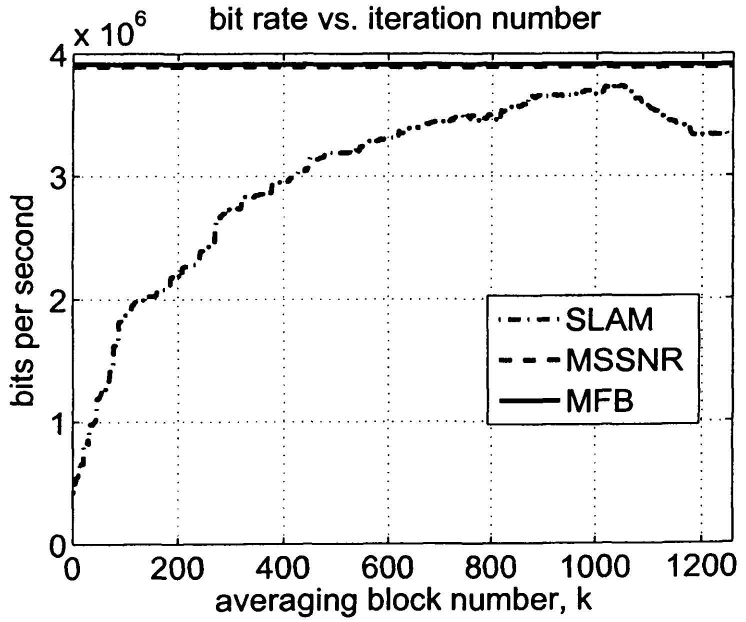

**Figure 5.8.** Achievable bit rate versus iteration number at 40 dB SNR of SLAM algorithm.

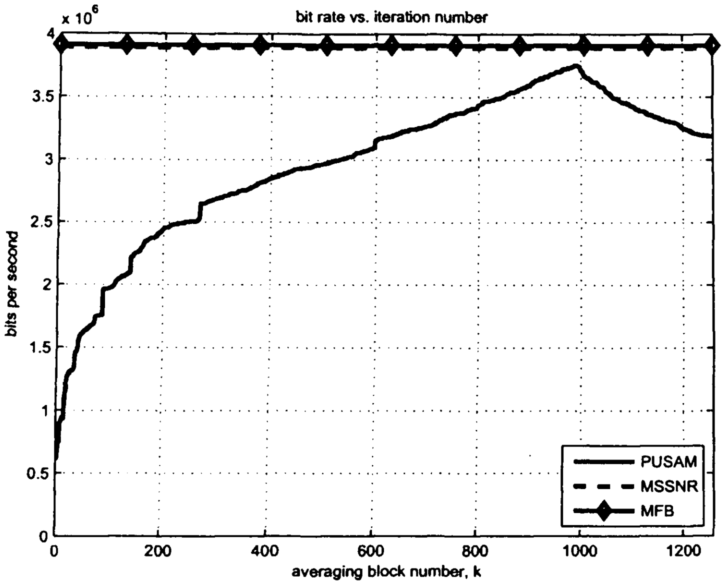

**Figure 5.9.** Achievable bit rate versus iteration number at 40 dB SNR of PUSAM algorithm.

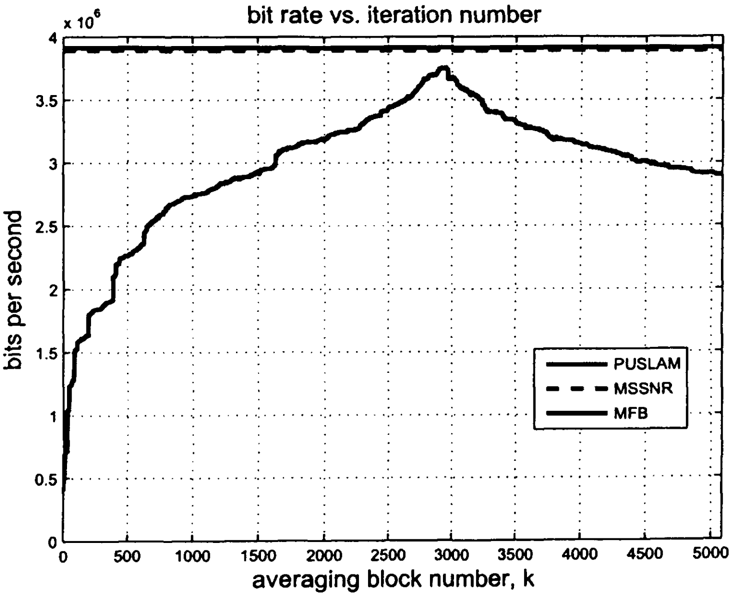

**Figure 5.10.** Achievable bit rate versus iteration number at 40 dB SNR of PUSLAM algorithm.

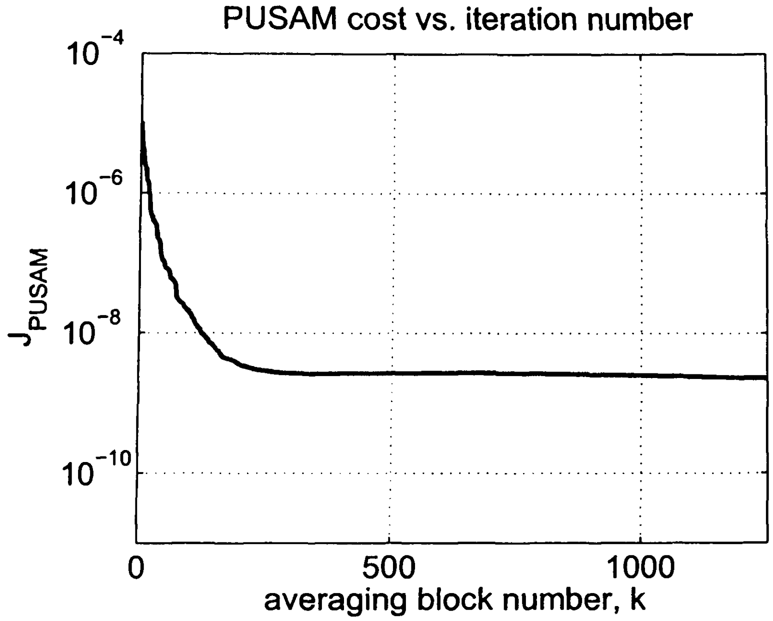

Figure 5.11. PUSAM cost versus iteration number.

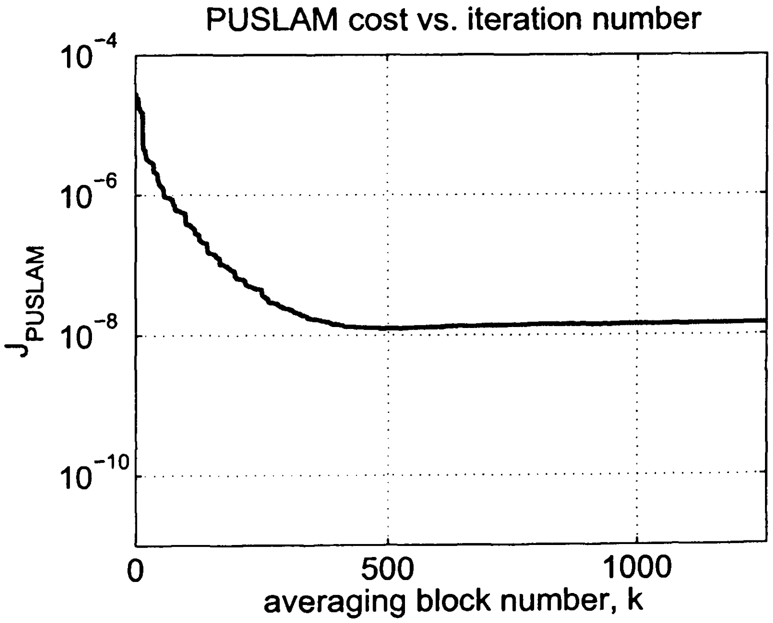

Figure 5.12. PUSLAM cost versus iteration number.

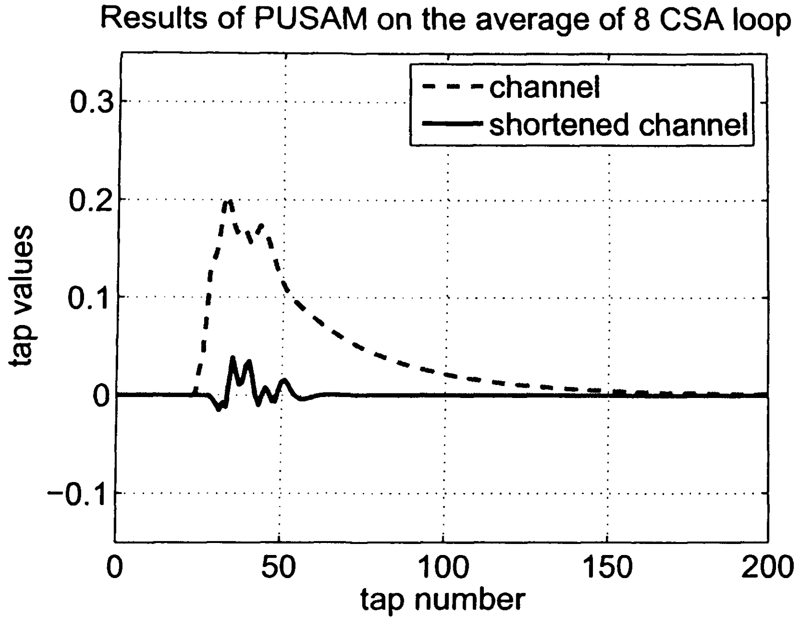

**Figure 5.13.** Channel (dashed) and shortened channel (solid) impulse response of PUSAM algorithm.

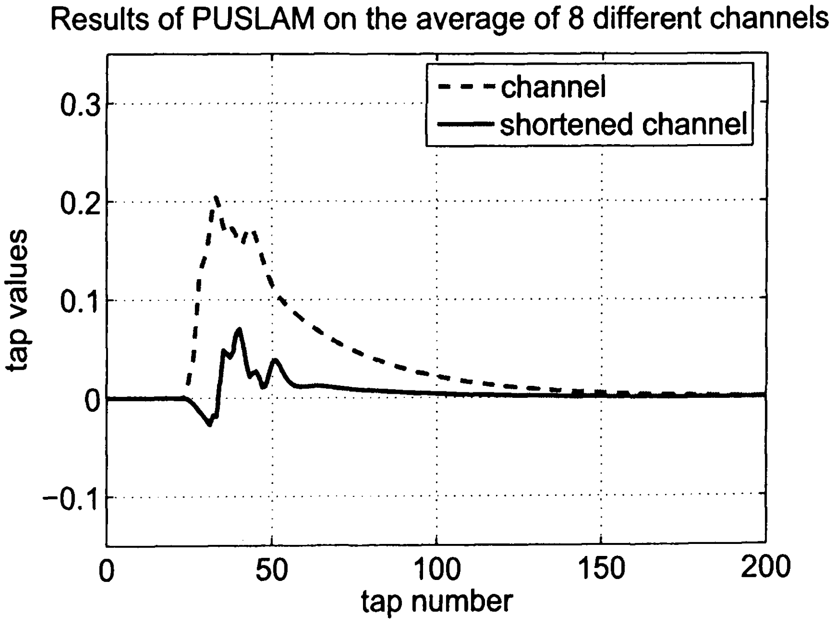

**Figure 5.14.** Channel (dashed) and shortened channel (solid) impulse response of PUSLAM algorithm.

## 5.5 Conclusions

The proposed algorithms can achieve the same performance as SAM and SLAM algorithms in terms of higher bit rates if the algorithms are stopped once the best performance is achieved as in [5] and shortening the channel as shown in the simulations results, the advantage of the proposed algorithms is that they essentially achieve the same performance whilst updating only half of the coefficients at each iteration which implies less computational complexity provided convergence time is not too long. The disadvantage of (PUSAM) and (PUSLAM) is that they can converge slower than the SAM and SLAM algorithms.

Chapter 6

# RANDOM COEFFICIENT SELECTION IN PARTIAL UPDATE BLIND CHANNEL SHORTENING ALGORITHMS

## 6.1 Random Partial Update Adaptive Filtering

Random partial updating is an effective method for reducing computational complexity in adaptive filter implementations provided the convergence time is not increased too much relative to conventional adaptive filter algorithms also it is an effective method for improve the convergence which has been the problem associated with the deterministic partial update scheme [36]. In this chapter, the new random partial update sum-squared auto-correlation minimization (RPUSAM) algorithm is proposed. This algorithm has low computational complexity whilst achieving improved convergence performance, in terms of achievable bit rate, over the PUSAM algorithm with a deterministic coefficient update strategy as in Section (5.2). The performance advantage of the RPUSAM algorithm is shown on eight different carrier serving area test loops (CSA) channels and comparisons are made with the original SAM and the PUSAM algorithms.

### 6.1.1 System model

The same system model shown in Figure (3.2) is used. The assumptions for the signal and noise in Section (5.1) are used. For convenience in this work real signals are also assumed but generalization to the complex case is straight-forward. It is assumed that $2L_c < N_{fft}$ holds, where $L_c$ is the order of effective channel and $N_{fft}$ is the FFT size [4], which means that the length of the effective channel is less than half the FFT size. The signal $n(k)$ is a zero-mean, independent identically distributed (i.i.d.), noise sequence, uncorrelated with the source sequence with variance $\sigma_n^2$. The received sequence $r(k)$ is

$$r(k) = \sum_{j=0}^{L_h} h(j)x(k-j) + n(k) \tag{6.1.1}$$

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

$$y(k) = \sum_{j=0}^{L_w} w(j)r(k-j) = \mathbf{w}^T\mathbf{r}_k \tag{6.1.2}$$

where $\mathbf{r}_k = [r(k) \quad r(k-1) \cdots r(k-L_w)]^T$ and $\mathbf{w}$ is the impulse response vector of the TEQ $\mathbf{w} = [w_0 \quad w_1 \quad w_2 \cdots w_{L_w}]^T$.

### 6.1.2 RPUSAM

For the effective channel $\mathbf{c}$ to have zero taps outside a contiguous window of size $(v+1)$, its autocorrelation values should be zero outside a window of size $2v+1$. The autocorrelation sequence of the effective channel is given by

$$R_{cc}(l) = \sum_{j=0}^{L_c} c(j)c(j-l) \tag{6.1.3}$$

and for a shortened channel, it must satisfy

$$R_{cc}(l) = 0, \forall \ |l| \ > \ v \tag{6.1.4}$$

The cost function $J_{rpusam}$ which is the same form as $J_{sam}$ is defined based upon minimizing the sum squared auto-correlation terms, i.e.,

$$J_{v+1} = \sum_{l=v+1}^{L_c} R_{cc}(l)^2 \tag{6.1.5}$$

### 6.1.3 Adaptive Algorithm

The steepest gradient-descent algorithm to minimize the RPUSAM cost $J_{v+1}$ is

$$\mathbf{w} = \mathbf{w}^{old} - \mu \nabla_{\mathbf{w}} ( \sum_{l=v+1}^{L_c} E[y(k)y(k-l)]^2) \tag{6.1.6}$$

where $\mu$ is the step size and $\nabla_{\mathbf{w}}$ is the gradient evaluated at $\mathbf{w} = \mathbf{w}^{old}$. The instantaneous cost function is defined, where the expectation operation is replaced by a moving average over a user-defined window of length $N_{avg}$

$$J_{v+1}^{inst}(k) = \sum_{l=v+1}^{L_c} \left\{ \sum_{n=kN_{avg}}^{(k+1)N_{avg}-1} \frac{y(n)y(n-l)}{N_{avg}} \right\}^2 \tag{6.1.7}$$

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

The proposal here is to improve the deterministic partial update scheme to exploit improved convergence of random selection [36] as discussed in Chapter 2, which is particularly important when minimizing non quadratic and multimodal cost functions as used in this thesis, and thereby achieve performance close to SAM for any channel. The set of indices of the coefficients of the adaptive filter is given by $\{1, 2, \ldots, L_w + 1\}$. This set is split into $P$ different disjoint but equal size subsets denoted $S_i$, $i = 1, \ldots, P$. Then, at each iteration one of these $P$ subsets is selected at random with probability $1/P$, and only those coefficients within the adaptive filter having indices from that subset are updated.

The resulting update equation can be written as in (6.1.8) where $\mathbf{M}(i)$ is a diagonal matrix with unity elements on the principle diagonal corresponding to the chosen subset $S_i$ and zeros elsewhere; and $\mathbf{w}(0)$ is initialized as for SAM. The computational complexity of this algorithm at each iteration is effectively $3NL_w(L_c - v)/P$ and therefore the computational complexity reduction is $1/P$ of the SAM algorithm.

$$\begin{aligned}\mathbf{w}(k+1) = \mathbf{w}(k) - 2\mu \times \mathbf{M}(i) \times & \sum_{l=v+1}^{L_c} \left\{ \sum_{n=kN_{avg}}^{(k+1)N_{avg}-1} \frac{y(n)y(n-l)}{N_{avg}} \right\} \\ & \times \left\{ \sum_{n=kN_{avg}}^{(k+1)N_{avg}-1} \left( \frac{y(n)\mathbf{r}_{n-l} + y(n-l)\mathbf{r}_n}{N_{avg}} \right) \right\} \end{aligned} \qquad (6.1.8)$$

The performance of this algorithm is again assessed by simulation due to the difficulty to perform mathematical analysis.

### 6.1.4 Simulations

The standard parameters of an ADSL downstream transmission were simulated as in [4]. The step size used was 5. Four subsets ($P = 4$) were used in the RPUSAM algorithm, the FFT size $N_{fft} = 512$, the TEQ had 16 taps and the channel was the test ADSL channel CSA loop 1 available at [68]. The noise was set such that $\sigma_x^2 \|c\|^2 / \sigma_n^2 = 40\ dB$ where $\|.\|$ denotes the Euclidean norm; and 75 OFDM symbols were used. All algorithms are compared with the maximum shortening SNR (MSSNR) solution, which attempts to minimize the energy outside the window of interest while holding the energy inside fixed [3], which is obtained using the code at [68], and the matched filter bound (MFB) on capacity, which assumes no ICI. The bit rate on each subcarrier is determined using noise margin $\gamma_m$ = 6dB and the coding gain $\gamma_c$ = 4.2dB. The value of $\Gamma_{gap}$ = 9.8dB is used which corresponds to a probability of error $10^{-7}$ and the QAM modulation used across the subcarriers. How the bit rate is calculated has been given in Section (5.4).

In Figures (6.2), (6.4), and (6.8), the achievable bits per second [15] as a function of the averaging block number, k , are plotted which show the improved convergence property of RPUSAM over PUSAM, best performance is achieved at approximately 350 rather than 900 blocks, which also approaches the full SAM algorithm of approximately 250 blocks. Decrease in the bit rates after achieving the peak bit rates is clear for the RPUSAM algorithm. In Figures (6.1), (6.3), and (6.5) the shortened channels are compared with the original channels and all algorithms are confirmed to be effective. The support of the shortened channel is restricted to lie within the first 50 taps. Figure (6.9) shows the average performance of RPUSAM in term of shortening the channel for eight different CSA channels to make sure that the proposed algorithm performs similarly with different channels.

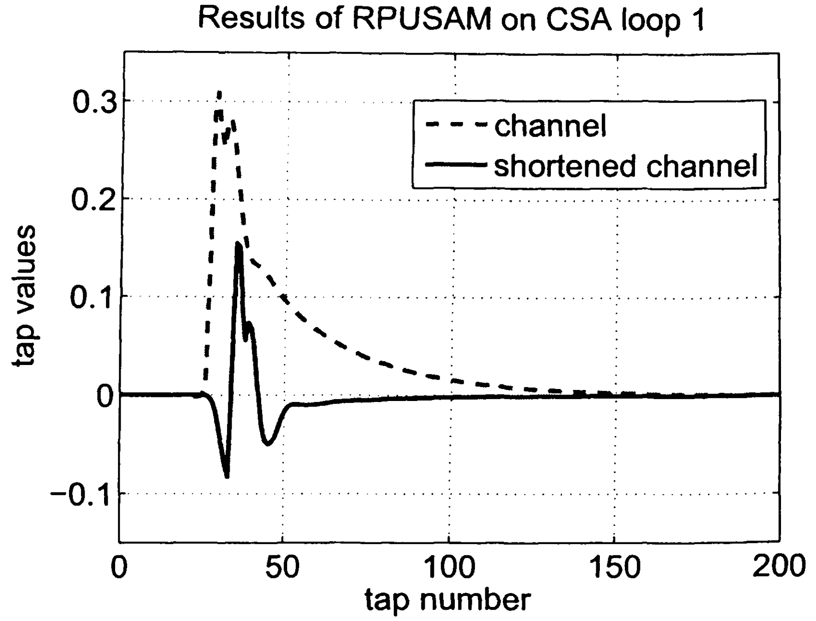

**Figure 6.1.** Channel (dashed) and shortened channel (solid) impulse response of RPUSAM

## 6.2 A Blind Lag-Hopping Adaptive Channel Shortening Algorithm (LHSAM)

Analytical results [6] showed that optimizing the single lag autocorrelation minimization (SLAM) cost does not guarantee convergence to high signal to interference ratio (SIR), an important metric in channel shortening applications. This potential limitation of the SLAM algorithm is overcome in this work whilst retaining its computational complexity advantage by minimizing the square of a single autocorrelation value with randomly selected lag. The proposed lag-hopping adaptive channel shortening algorithm based upon squared autocorre-

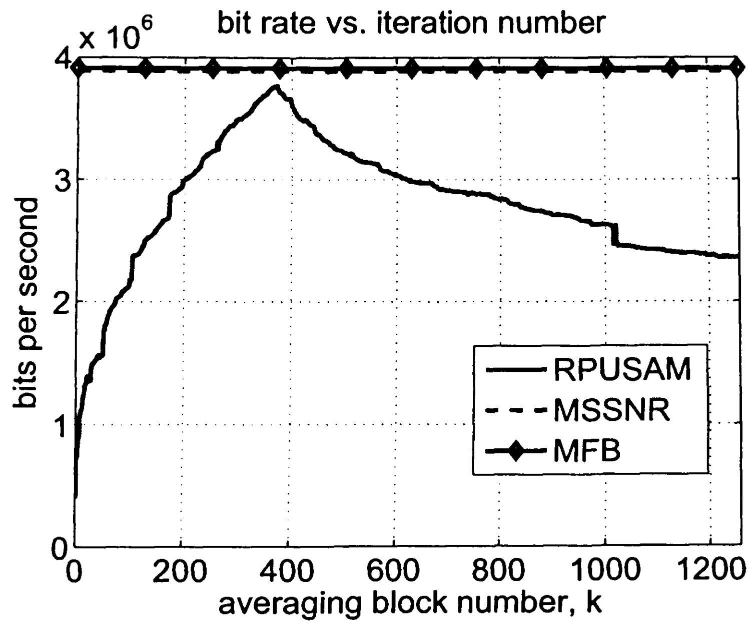

Figure 6.2. Achievable bit rate versus averaging block number at 40 dB SNR of RPUSAM

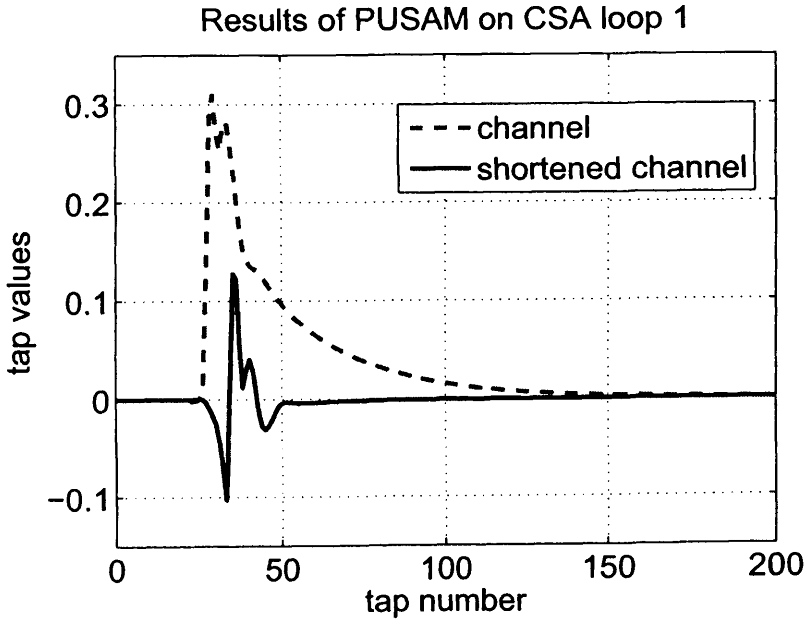

Figure 6.3. Channel (dashed) and shortened channel (solid) impulse response of PUSAM

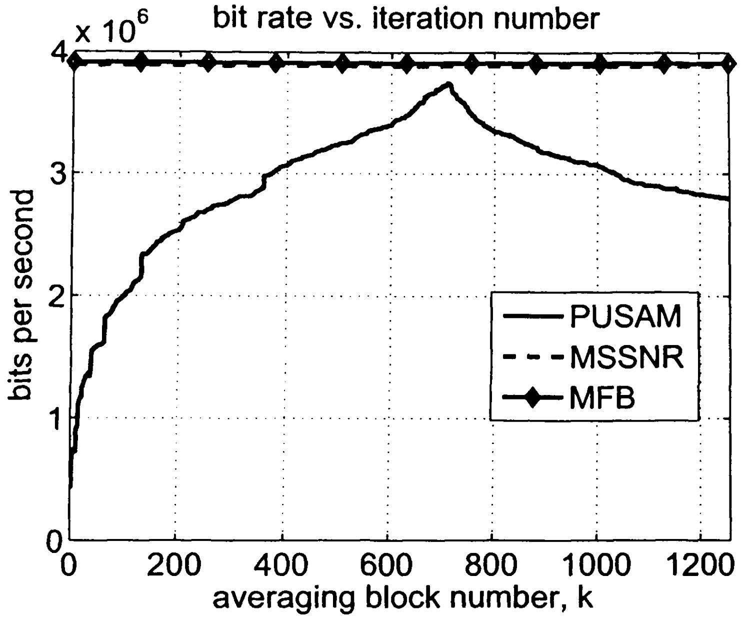

Figure 6.4. Achievable bit rate versus averaging block number at 40 dB SNR of PUSAM

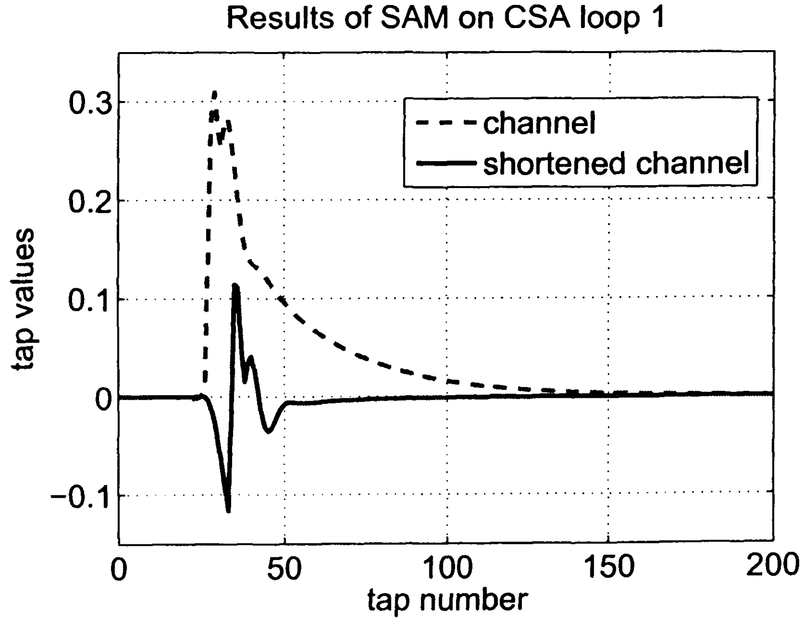

Figure 6.5. Channel (dashed) and shortened channel (solid) impulse response of SAM

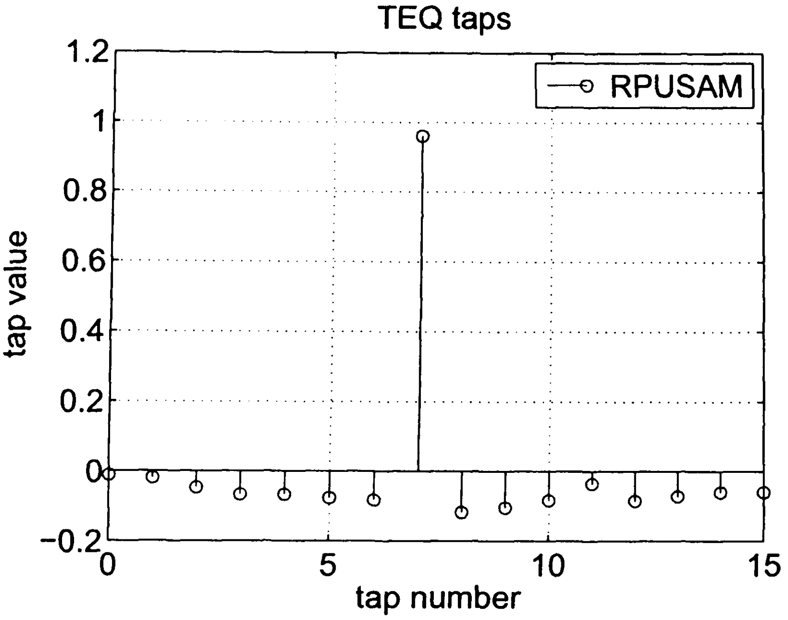

**Figure 6.6.** TEQ taps.

lation minimization (LHSAM) has, therefore, low complexity as in the SLAM algorithm and, more importantly, a low average LHSAM cost can guarantee to give a high SIR as for the SAM algorithm.

### 6.2.1 System Model

In this work, a channel shortening filter at the output of a channel is used, as shown in Figure (6.10). The case in Figure (6.10) where a single input multiple output channel (SIMO) is referred to where $L > 1$, a SISO channel is a special case when $L = 1$. This SIMO channel can be either from the use of multiple receive antennas or by over-sampling at the receiver, and $\mathbf{r}(k)$ is the received signal vector at the input to the receiver $\mathbf{r}(k) := [r_k^{(1)}, r_k^{(2)}, ...., r_k^{(L)}]$, which is the sum of some additive noise $n(k) := [n^{(1)}, n^{(2)}, ..., n^{(L)}]$ and the output signal from the transmitter $s(k)$ filtered by a channel filter $\mathbf{h}(z) :=$

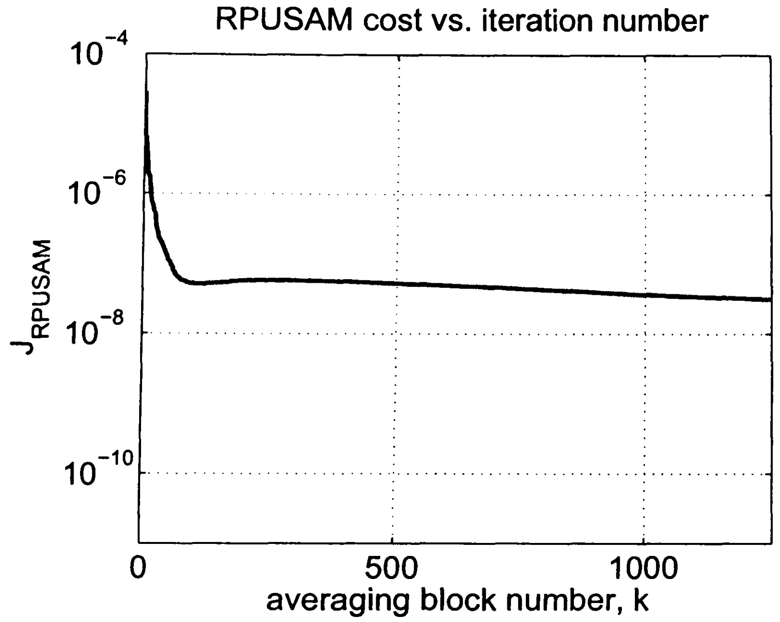

Figure 6.7. RPUSAM cost versus iteration number.

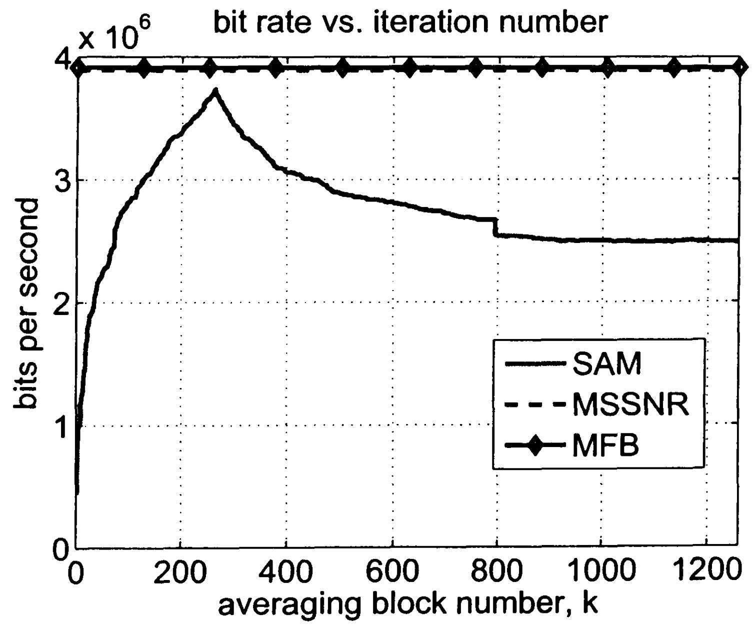

Figure 6.8. Achievable bit rate versus averaging block number at 40 dB SNR of SAM

$[\mathbf{h}^{(1)}(z), \mathbf{h}^{(2)}(z), ..., \mathbf{h}^{(L)}(z)]^T$ (where T denotes the transpose operation). Each sub-channel $\mathbf{h}^{(i)}(z)$ is modelled as a finite impulse response filter

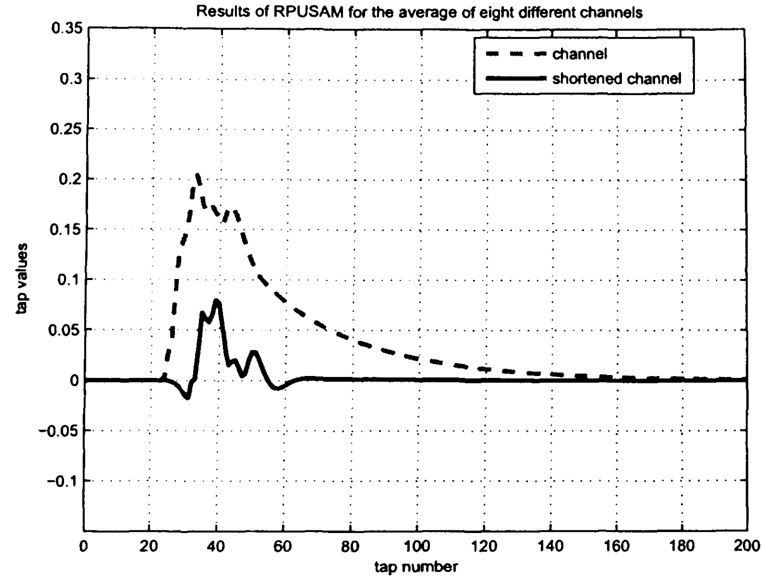

**Figure 6.9.** Channel (dashed) and shortened channel (solid) impulse response for the average of eight CSA channels of RPUSAM

of order $M$

$$\mathbf{h}^{(i)}(z) = \sum_{k=0}^{M} h_k^{(i)} z^{-k} \tag{6.2.1}$$

At the receiver, a channel shortening filter $\mathbf{w}(z) = [\mathbf{w}^{(1)}(z), \mathbf{w}^{(2)}(z), ..., \mathbf{w}^{(L)}(z)]$ processes the vector valued input $\mathbf{r}_k$ by summing the output of the channel shortening filters $\mathbf{w}^{(i)}(z)$ operating on each of the sub-channel outputs $r_k^{(i)}$. Channel shortening filters with impulse responses of order $T$ are considered, so that

$$\mathbf{w}^{(i)}(z) = \sum_{k=0}^{T} w_k^{(i)} z^{-k} \tag{6.2.2}$$

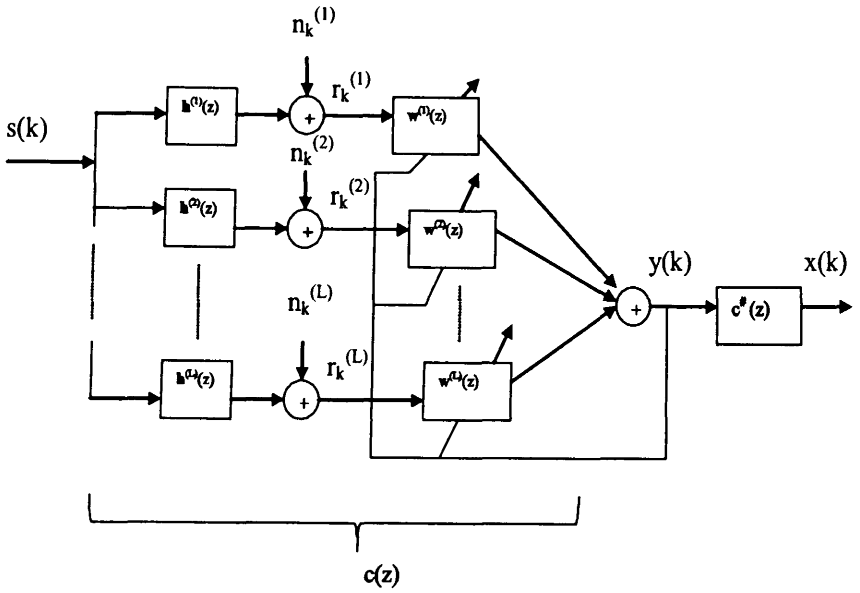

**Figure 6.10.** System model for blind adaptive channel shortening with the matched filter.

The overall effective filter $c(z)$ between the transmitted symbols $s(k)$ and the output of the channel shortener $y(k)$ can be written as

$$\mathbf{c}(z) = \mathbf{w}^T(z)\mathbf{h}(z) = \sum_{i=1}^{L} \mathbf{w}^{(i)}(z)\mathbf{h}^{(i)}(z) \tag{6.2.3}$$

Then the output $y(k)$ is processed by the receiver with a matched filter

$$\mathbf{c}^{\sharp}(z) = \sum_{k=0}^{N} cz^k = \mathbf{c}(z^{-1}) \tag{6.2.4}$$

The final output $x(k)$ is created which is passed on to the rest of the receiver. The importance of the channel shortening filter is to ensure that final transfer function $\mathbf{c}^{\sharp}(z)\mathbf{c}(z)$ between the output of the transmitter and the output of the matched filter has an impulse response which is zero outside of a window of length $2v+1$. The use of an optimal Viterbi or forward-backward sequence detector at the output of the matched filter can cause this in single carrier systems [6]. These optimal sequence detection algorithms have a high complexity in the effective impulse response length, it is often computationally infeasible to use them for long ($> 10$) effective impulse responses. In these instances, by shortening the length of the effective channel, the channel shortener allows one to reap the performance benefits of such sequence detection algorithms at a reasonable computational complexity. In multicarrier systems such as discrete multi tone (DMT) or coded orthogonal frequency division multiplexing (C-OFDM), on the other hand, one employs the channel shortener to ensure that only simple scalar equalization is required one at each bin the output of the FFT. It is not assumed either of these instances in particular. Instead, it is assumed that the signal $s(k)$ has zero mean, unit-variance, and is uncorrelated, so that $E[s(k)s(\acute{k})] = 0$ for $k \neq \acute{k}$. It is further assumed that the sub-channels $\mathbf{h}^{(i)}(z)$ have no common zeros (i.e. are co-prime), and the length of the shortening filters has been selected in a manner so that any effective transfer function $\mathbf{c}(z)$ can be created by choosing an appropriate channel shortener.

### 6.2.2 Blind Adaptive Channel Shortening Metrics

The blind channel shortening metrics of interest will be reviewed. By metric, it means a function which assigns to every combined response $\mathbf{c}(z)$ a cost. A channel shortening design according to a particular metric is similar to the combined response $\mathbf{c}(z)$ with the minimum metric that is achievable for some shortener $\mathbf{w}(z)$. The study will be focussed on channel shortening designs which operate using the autocorrelation of $y(k)$, and thus on the autocorrelation of the combined response $\mathbf{c}(z)$, whose transform may be written as

$$\mathbf{R}(z) = \mathbf{c}(z)\mathbf{c}^{\sharp}(z) = \sum_{k=-N}^{N} Q_k z^{-k} \tag{6.2.5}$$

The metrics which are considered all require an extra constraint, which is choosing to be a unit energy constraint on $\mathbf{c}(z)$, so that

$$\sum_{k=0}^{N} |c_k|^2 = R_0 = 1 \tag{6.2.6}$$

The sum squared autocorrelation metric (SAM) [4] is the sum of the autocorrelation squared outside of the window of length $2v+1$.

$$J_{SAM} = \sum_{|l| \geq v+1} |R_m|^2 = 2 \sum_{l=v+1}^{N} |R_m|^2 \tag{6.2.7}$$

The sum absolute autocorrelation metric (SAAM) [26] is similar to SAM, which is the sum of the absolute autocorrelation values outside of a window of length $2v+1$

$$J_{SAAM} = \sum_{|l| \geq v+1} |R_m|$$

$$J_{SAAM} = \sum_{|l| \geq v+1} |R_m| \tag{6.2.8}$$

The single lag autocorrelation metric (SLAM) [5] claims that it reduces the complexity of SAM designs by minimizing the absolute value of only the correlation at the lag $v+1$.

$$J_{SLAM} = |R(v+1)|^2, \tag{6.2.9}$$

[4], [26] and [5] show that $J_{SAM}$, $J_{SAAM}$, and $J_{SLAM}$ are all zero if the combined response $\mathbf{c}(z)$ has taps which are all zero except for possibly

some within a window of length $v$. The non-negative definiteness of these metrics then shows that they all have global minima for shortened $\mathbf{c}(z)$. Furthermore, $J_{SAM}$ and $J_{SAAM}$ are equal to zero only if $\mathbf{c}(z)$ has taps which are all zero except possibly within a window of size $v+1$. Thus, for SAM and SAAM the global minima are all at perfectly shortened channels. As it will be seen later, however, this is not the case for the SLAM cost. These global minima (partially) establish the utility of the SAM, SAAM, and SLAM costs. Note that these designs suffer from inherent ambiguities in terms of the combined response $\mathbf{c}(z)$ because they depend on $\mathbf{c}(z)$ only through the auto-correlation $\mathbf{R}(z)$. In particular the auto-correlation of a combined response $\mathbf{c}(z)$ remains unchanged if one replaces a zero by its conjugate inverse and re-normalizes to enforce the unit energy constraint. To see the reason of this, let the zeros of $\mathbf{c}(z)$ be $\{d_k\}$, so that

$$\mathbf{c}(z) = a_0 \prod_{k=1}^{N} (1 - d_k z^{-1}) \tag{6.2.10}$$

This gives an autocorrelation with transform

$$\mathbf{c}(z)\mathbf{c}^{\#}(z) = |a_0|^2 \prod_{k=1}^{N} (1 - d_k z^{-1})(1 - d_k z) \tag{6.2.11}$$

Now consider $\mathbf{c}_2(z)$, which is created by flipping one of the zeros over the unit circle and conjugating it, i.e. by replacing $d_1$ by $\frac{1}{d_1}$ , and then normalizing the taps so that they are unit norm, so that

$$\mathbf{c}_2(z) = b_0 \left(1 - \frac{z^{-1}}{d_1}\right) \prod_{k=2}^{N} (1 - d_k z^{-1}) \tag{6.2.12}$$

Then $\mathbf{c}_2(z)$ has an autocorrelation with transform

$$\begin{aligned}\mathbf{c}_2(z)\mathbf{c}_2^\sharp(z) &= |b_0|^2\left(1-\frac{z^{-1}}{d_1}\right)(1-d_1^{-1}z)\\ &\prod_{k=2}^{N}(1-d_kz^{-1})(1-d_kz)\\ &= \frac{|b_0|^2}{|d_1|^2}\prod_{k=1}^{N}(1-d_kz^{-1})(1-d_kz)\\ &= \mathbf{c}(z)\mathbf{c}^\sharp(z) \qquad (6.2.13)\end{aligned}$$

where the last equality followed from $\frac{|b_0|^2}{|d_1|^2} = |a_0|^2$ due to the unit energy constraint. This leads to the next section, which shows the importance of the inclusion of the matched filter $\mathbf{c}^\sharp(z)$ in the system, as in Figure (6.10).

### 6.2.3 Importance of the Matched Filter

It is assumed that the matched filter $\mathbf{c}^\sharp(z)$ was not included in Figure (6.10), so that the signal output to the rest of the (not shown) receiver chain was $y(k)$. Without the matched filter, the goal becomes to shorten the channel to $v$ non-zero taps. The performance of this system was quantified with the matched filter removed by the best delay signal to inference ratio

$$\mathsf{SIR}(\{y(k)\}) = \max_{\Delta}\frac{\sum_{k=\Delta}^{\Delta+v}|\mathbf{c}(k)|^2}{\sum_{k=0}^{\Delta-1}|\mathbf{c}(k)|^2+\sum_{k=\Delta+v+1}^{N}|\mathbf{c}(k)|^2}\ . \qquad (6.2.14)$$

It shows that the SAM, SAAM, and SLAM costs are unsuited for this system, because there are combined responses $\mathbf{c}(z)$ with costs very near to the global optimal value (0) of these costs with very high SIR. This is all due to the autocorrelation based nature of the SAAM, SAM, and

SLAM designs. In particular, unlike the auto-correlation, the quality of a combined response as a channel shortener (e.g. its SIR) changes dramatically when you flip one of its zeros over the unit circle and conjugate it. This point is perhaps best illustrated with an example. Consider a combined response $\mathbf{c}(z)$ whose zeros $d_k, k \in 1, ..., N$ are $(j = \sqrt{-1})$

$$d_k = \alpha \exp\left(j\frac{2\pi k}{N}\right), \quad k \in 1, ..., N \tag{6.2.15}$$

This gives a combined response, after unit energy normalization, of

$$\mathbf{c}(z) = \frac{1}{\sqrt{1+\alpha^{2N}}} - \frac{\alpha^N}{\sqrt{1+\alpha^{2N}}} z^{-N} \tag{6.2.16}$$

which, for $\alpha < 1$ will have a best-delay SIR of $-20\ N \log_{10}\ (\alpha)$ dB, which can be made arbitrarily large via choice of $\alpha$. As one would expect, the SAM, SAAM, and SLAM costs for this response are very low as well. In particular, the SAM cost is $10 \log_{10}\left(\frac{\alpha^{2N}}{(1+\alpha^{2N})^2}\right)$ dB and the SAAM cost is $10 \log_{10}\left(\frac{\alpha^N}{(1+\alpha^{2N})}\right)$ dB for any $v$. The SLAM cost is $-\infty$ dB for any $v < N$, and is $10 \log_{10}\left(\frac{\alpha^N}{(1+\alpha^{2N})}\right)$ dB for $v = N$. Because they depend only on the autocorrelation, the SAM, SAAM, and SLAM costs do not change if the following changes are made

$$d_1 \mapsto \frac{1}{d_1}, d_{N-1} \mapsto \frac{1}{d_{N-1}}, d_N \mapsto \frac{1}{d_N} \tag{6.2.17}$$

However, the best delay SIR changes under this transformation. The particular instance when $\alpha = \frac{1}{2}$ , $v = 1$, and $N = 9$ is shown in Figure (6.11). Here the best delay SIR was 54 dB before the translation (6.2.17), but after the translation (6.2.17) the best delay SIR becomes 1 dB. The SAM and SAAM costs remain at -54 dB and -27 dB respec-

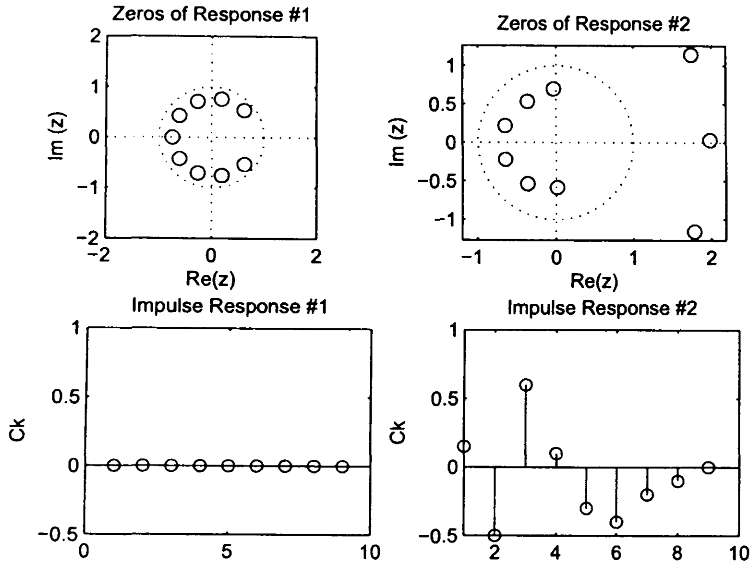

**Figure 6.11.** Two combined responses c(z) with the same autocorrelation, and thus the same SAM, SAAM, and SLAM costs (-54 dB, -27 dB, and -1 dB, respectively), but with very different best delay SIRs when no matched filter is used (54 dB versus 1dB).

tively during this translation. The SLAM cost remains at -1 dB. If, however, a matched filter is added to the receiver, then the effective impulse response between $s(n)$ and $x(k)$ is the autocorrelation of $\mathbf{c}(z)$, i.e. $\mathbf{c}(z)\mathbf{c}^\sharp(z)$. This means that the SAM and SAAM costs are minimizing the sum squared magnitude and the sum magnitude, of the taps outside of the window of length $2v+1$ in the effective response $\mathbf{R}(z)$ between $s(k)$ and $x(k)$, which is related to the signal to interference ratio of $x(k)$, as it will be shown in the next section.

Although it is clear from the argument and example that a matched filter is important for blind designs based on the autocorrelation of the combined response, it is not clear how to choose the appropriate matched filter at the receiver. In particular, as pointed out in [4], [15], [26] and [5], adaptive channel shortening filters which adaptively minimize these costs directly via choice of the channel shortener $\mathbf{w}(z)$ can be made. Because these algorithms operate directly on the received data without estimating the channel $\mathbf{h}(z)$ or the combined response $\mathbf{c}(z)$, after they have converged, although the channel may be shortened, the combined impulse response is still unknown. Thus, the requirement of a matched filter also implicitly includes the requirement that the combined impulse response $\mathbf{c}(z)$ be estimated. Alternatively, the example shown above suggests that a minimum phase requirement on $\mathbf{c}(z)$ may be sufficient, although this might require estimation of $\mathbf{c}(z)$ in order to determine if it is minimum phase. Either way, it seems that $\mathbf{c}(z)$ will have to be estimated, or designs based on auto-correlation will suffer from the ambiguities indicated above.

### 6.2.4 SIR Performance

Going back to the system depicted in Figure (6.10) with the matched filter present, a relation between the blind channel shortening metrics SAM, SAAM, and SLAM and the signal to inference power ratio in $x(k)$ is provided, which is defined to be

$$\mathsf{SIR} = \frac{\sum_{m=-v}^{v} |R_m|^2}{\sum_{m=-N}^{-v+1} |R_m|^2 + \sum_{m=v+1}^{N} |R_m|^2}$$

It can be noted that the denominator in this expression is the SAM cost, and considering only those $\mathbf{c}(z)$ which satisfy the unit energy constraint, the following relation can be obtained [6]

$$\begin{aligned}\text{SIR (dB)} &= 10\log_{10}\left(\sum_{m=-v}^{v}|R_m|^2\right) - 10\log_{10}(J_s)\\ &= 10\log_{10}\left(1+2\sum_{m=1}^{v}|R_m|^2\right) - 10\log_{10}(J_s)\\ &\geq -J_{SAM}\ \text{(dB)}\end{aligned} \tag{6.2.18}$$

So that a low SAM cost can be guaranteed to give a high SIR at the output of the matched filter. Furthermore, since

$$\begin{aligned}J_{SAAM}^2 &= \sum_{|m|\geq v}|R_m|^2 + \sum_{|i|\geq v}\sum_{|k|\geq v,k\neq i}|R_m(l)||R_i|\\ &= J_{SAM} + \sum_{|i|\geq v}\sum_{|k|\geq v,k\neq i}|R_m||R_i|\\ &\geq J_{SAM}\end{aligned}$$

then the SAAM design affords lower bound on the performance SIR

$$\text{SIR (dB)} \geq -2J_{SAAM}\ \text{(dB)} \tag{6.2.19}$$

So that a low SAAM cost also guarantees a high SIR at the output of the matched filter. Unfortunately, the SLAM design affords no such lower bound on the performance SIR as it can be seen in the next section.

Moving now to upper bounds, note that for $\mathbf{c}(z)$ satisfying the unit energy constraint, $l > 0$,

$$|R_m| \leq \max_{|eigenvalue|} \frac{1}{2} \begin{bmatrix} 0_{(N+1-m)} \times m & \mathbf{I}_m \\ 0_{m\times m} & 0_{m\times(N+1-m)} \end{bmatrix}$$
$$+\frac{1}{2} \begin{bmatrix} 0_{m\times(N+1-m)} & 0_{m\times m} \\ \mathbf{I}_m & 0_{(N+1-m)} \times m \end{bmatrix} \tag{6.2.20}$$

Denoting this maximum eigenvalue magnitude by $|\lambda_{MAX,l,N}|$, the SIR at the matched filter output among those $c(z)s$ may be upper bound obeying the unit energy constraint by

$$\text{SIR (dB)} \leq 10\log_{10}\left(1 + 2\sum_{0<m\leq v} |\lambda_{MAX,m,N}|^2\right) - J_{SAM}(\text{dB}) \tag{6.2.21}$$

Furthermore, since via the relation between the 2 and 1 norms,

$$J_{SAM} \geq \frac{J^2_{SAAM}}{(N-v)} \tag{6.2.22}$$

also it follows that

$$\text{SIR (dB)} \leq 10\log_{10}\left(1 + 2\sum_{0<m\leq v} |\lambda_{MAX,m,N}|^2\right) + 10\log_{10}(N-v) - 2J_{SAAM}(\text{dB}) \tag{6.2.23}$$

Finally, since $J_{SAAM} \geq J_{SLAM}$, the bound

$$\text{SIR (dB)} \leq 10\log_{10}\left(1 + 2\sum_{0<m\leq v} |\lambda_{MAX,m,N}|^2\right) + 10\log_{10}(N-v) - 2J_{SLAM}(\text{dB}) \tag{6.2.24}$$

shows that a high SLAM cost implies poor SIR performance.

### 6.2.5 LHSAM algorithm

The update equation of LHSAM algorithm can be written as:

$$\mathbf{w}(k+1) = \mathbf{w}(k) - 2\mu \left\{ \left( \sum_{n=kN_{avg}}^{(k+1)N_{avg}-1} \frac{y(n)y(n-l)}{N_{avg}} \right) \right\} \times \left\{ \left( \sum_{n=kN_{avg}}^{(k+1)N_{avg}-1} \frac{y(n)\mathbf{r}(n-l) + y(n-l)\mathbf{r}(n)}{N_{avg}} \right) \right\} \quad (6.2.25)$$

The key defining feature of the LHSAM algorithm is that at each iteration $k$, the lag "l" is chosen with equal probability to take on one of the values in the range of $v+1, \ldots\ldots, L_c$.

The LHSAM cost will be identical to that of the SAM cost as on the average all the lags of the SAM cost will be visited, whilst at each iteration the complexity is the same as SLAM

### 6.2.6 Simulations

The cyclic prefix used to simulate LHSAM was of length 32, the FFT size $N_{fft} = 512$, the TEQ had 16 taps and the channel was the test ADSL channel CSA loop 1 available at [68]. The noise was set such that $\sigma_x^2 \|c\|^2 / \sigma_n^2 = 40$ dB where $\|.\|$ denotes the Euclidean norm; and 75 OFDM symbols were used. To make fair comparison between LHSAM and SLAM, all the parameters are kept the same as in [5]. The step size used for SLAM and LHSAM was 600, in order to achieve convergence in approximately 1000 blocks. All algorithms are compared with the maximum shortening SNR solution [3], which was obtained using the code at [69], and the matched filter bound (MFB) on capacity, which assumes no ICI.

In the Figures (6.12), (6.13), and (6.14), the shortened channels are compared with the original channels and all algorithms are confirmed to be effective. The support of the shortened channel is restricted to lie within the first 50 taps. In the Figures (6.17), (6.18), and (6.19) the achievable bits per second as a function of the averaging block number are plotted which show the improved convergence property of LHSAM over SLAM, best performance is achieved at approximately 900 rather than 1010 blocks, due to the nature of the underlying cost function as a function of the parameters of the shortener. The bit rate was determined based on

$$BR = \sum_{i=1}^{N_{fft}} \log_2(1 + SNR_i/\Gamma)$$

The bit rate was computed using a 6-dB margin and a 4.2-dB coding gain. For more details, see [68], and for more details on how the achievable bit rate relates to SAM cost and ICI, see [4].

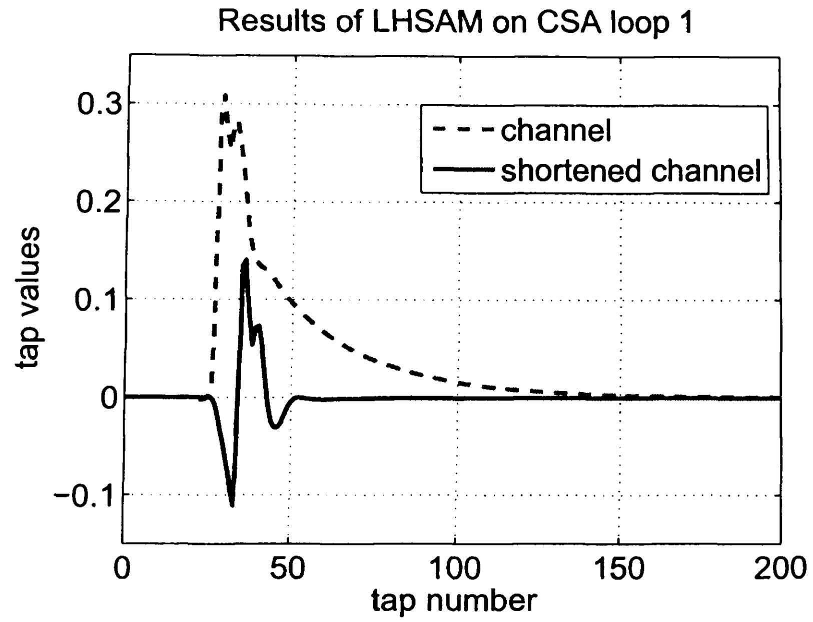

**Figure 6.12.** Channel (dashed) and shortened channel(solid) impulse response of LHSAM algorithm.

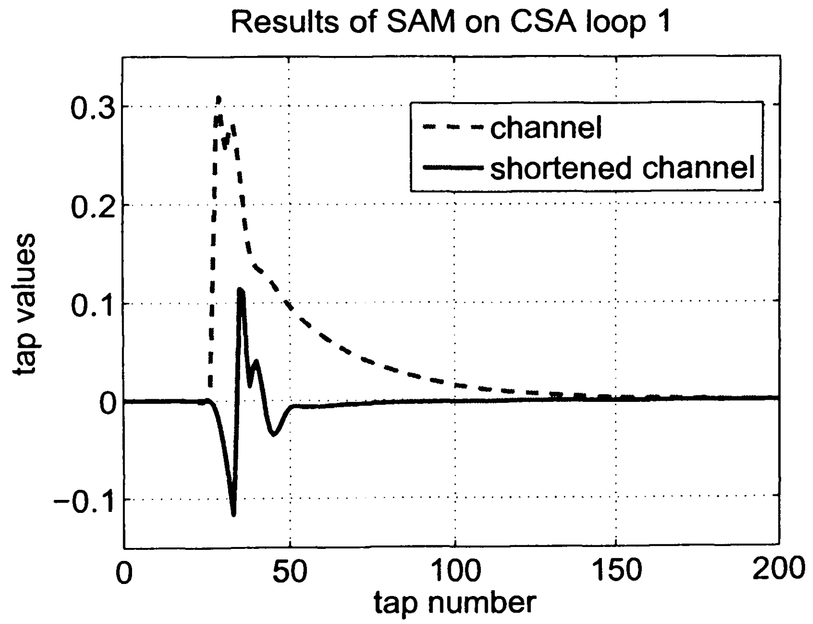

**Figure 6.13.** Channel (dashed) and shortened channel (solid) impulse response of SAM algorithm.

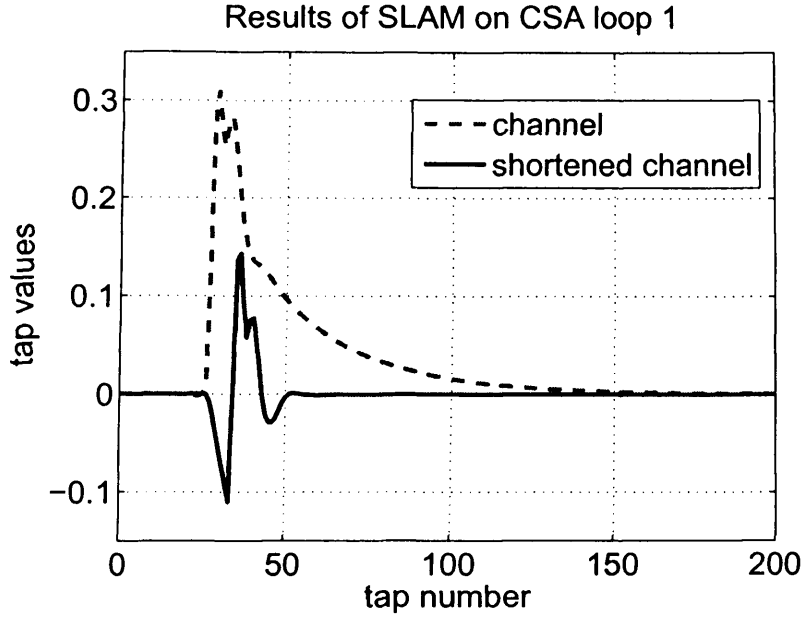

Figure 6.14. Channel (dashed) and shortened channel (solid) impulse response of SLAM algorithm.

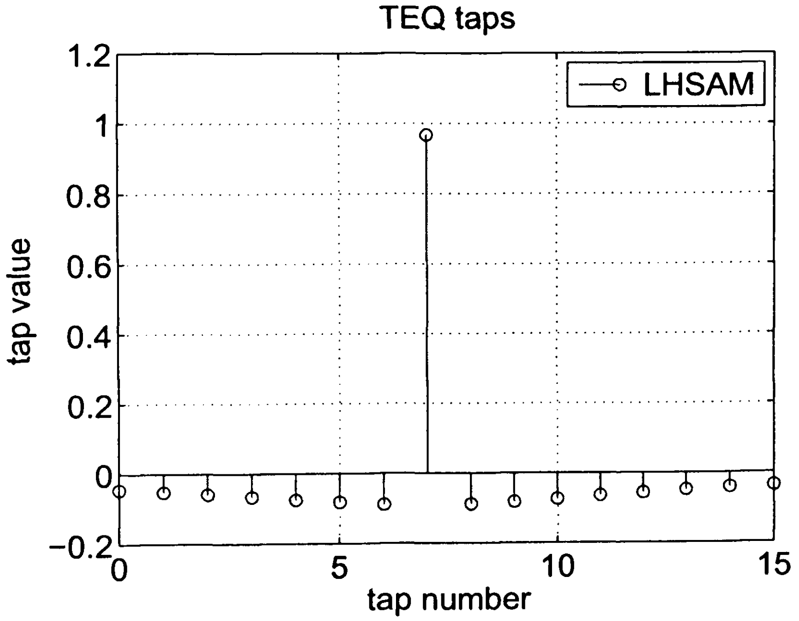

Figure 6.15. Converged TEQ taps.

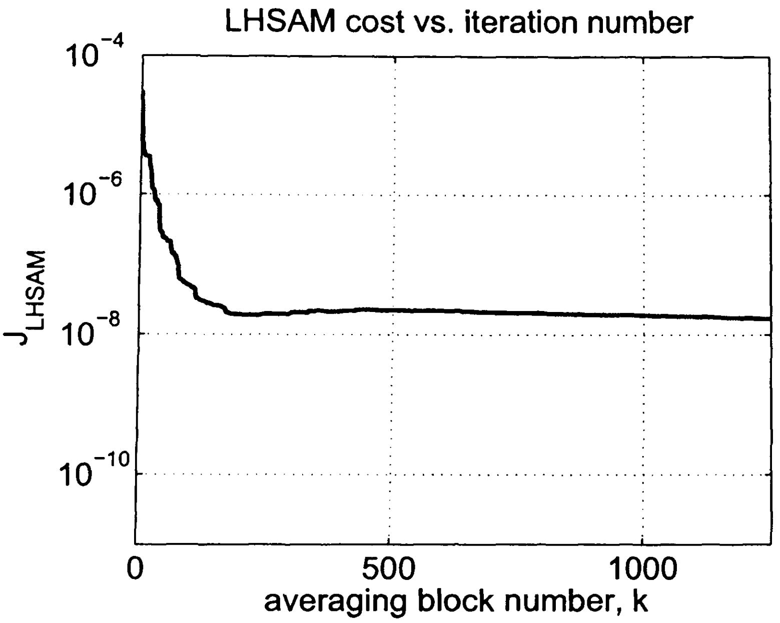

Figure 6.16. LHSAM cost versus iteration number.

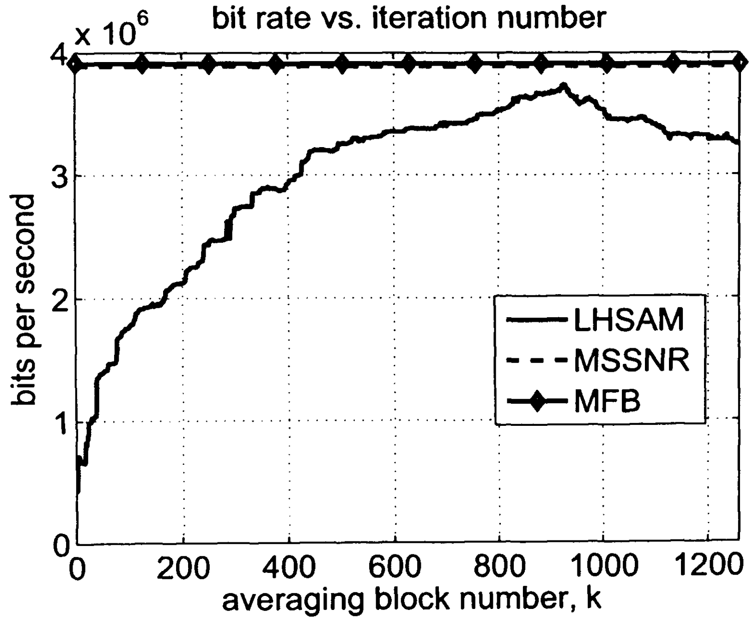

Figure 6.17. Achievable bit rate versus iteration number at 40 dB SNR of LHSAM algorithm

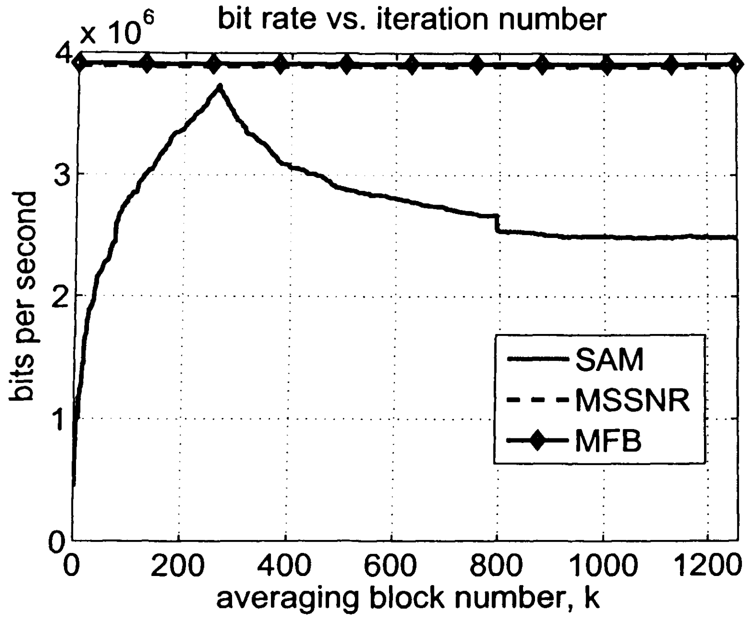

Figure 6.18. Achievable bit rate versus iteration number at 40 dB SNR of SAM algorithm

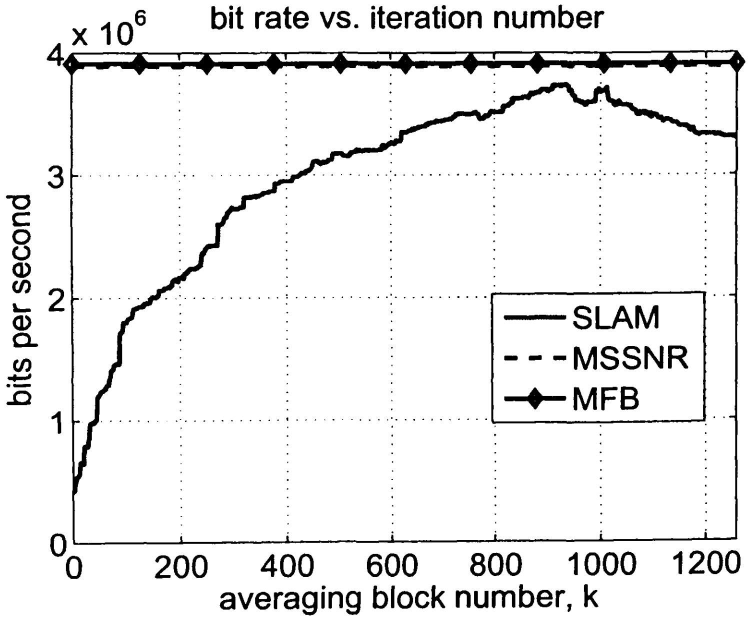

Figure 6.19. Achievable bit rate versus iteration number at 40 dB SNR of SLAM algorithm

## 6.3 Conclusion

The newly proposed RPUSAM algorithm essentially achieves the same result in terms of reducing the effective channel length as SAM and PUSAM with less complexity. The complexity reduction is achieved by only updating $N/P$ of the coefficients of the TEQ at each iteration in a random pattern; on the other hand, PUSAM updates the subsets of coefficients in a systematic fashion which degrades convergence greatly over conventional SAM. The proposed algorithm is confirmed to achieve channel shortening on a set of eight CSA channels.

Uniquely in this chapter random lag selection is introduced to mitigate the ill-convergence properties of the SLAM algorithm. The proposed algorithm achieves essentially the same result in terms of reducing the effective channel length as SLAM. Importantly, however, the disadvantage of SLAM in terms of the SIR performance has been overcome. The proposed algorithm has the same the low complexity advantage as SLAM. It also has the advantage that a low LHSAM cost will be identical to a low SAM cost which guarantees to yield a high SIR at the output of the matched filter. This is achieved as on the average all the lags of the SAM cost will be visited during convergence, whilst at each iteration the complexity being the same as SLAM.

Chapter 7

# CONCLUSIONS AND FURTHER RESEARCH

## 7.1 Conclusions

The implementation complexity of a multicarrier communication system is generally less than that of a single carrier system for the same amount of delay spread. This reduction in complexity is to a large extent due to the use of the CP which eliminates the need for an equalizer except for a single FEQ at each subchannel. However, to reduce the bandwidth efficiency loss due to insertion of CP, channel shortening or partial equalization in the form of a TEQ is introduced. The complexity of this partial equalization should, therefore, be kept low in order to keep the superiority of the multicarrier systems over single carrier systems. The throughput loss due to the insertion of the CP can further be reduced indirectly by applying channel shortening algorithms which are blind and do not need training. Furthermore, channel shortening should be made robust to the impulsive noise impairment found in ADSL channels.

Algorithms which attempt to restore each of the properties of the transmitted sequence that ought to be present in the equalized received sequence were studied. Chapter 3 shows that in order to create a blind, adaptive channel shortener, the redundancy which the transmitted sequence has due to the cyclic prefix in multicarrier or single-carrier cyclic prefix (SCCP) modulation, can be used in the property restoral sense. Algorithms using a philosophy called "property restoral" were studied such as the MERRY algorithm [37] which attempts to adapt the channel shortener with the aim of restoring the redundancy which is due to the cyclic prefix of the transmitted sequence. On the other hand, the SAM algorithm [4] minimizes the sum-squared auto-correlation terms of the effective channel impulse response outside a window of a CP-length. Chapter 3 also shows that the presence of null tones in the transmitted data is another common property of multicarrier signals, however its complexity is high. A blind, adaptive channel shortening algorithm can be derived with the goal of restoring the values of these tones to zero at the output of the receivers FFT , this results in a carrier nulling algorithm (CNA).

Chapter 4 proposes a robust blind adaptive channel shortening algorithms called DPUSAAM and RPUSAAM. These algorithms are based on updating only a portion of the coefficients of the channel shortening filter at each time sample instead of the entire set of coefficients. These algorithms are the first attempt in the field of using partial update filtering in blind adaptive channel shortening. The algorithms are also designed to be robust to impulsive noise impairment found in ADSL channels. These algorithms have low computational complexity whilst retaining essentially identical performance to the SAAM algorithm [26]. To assess the robustness of the DPUSAAM and RPUSAAM algorithms, the impulsive noise has been modelled as Gaussian-mixture and as $\alpha$-

stable distributions.

SAM [4] has relatively less complexity as compared to other channel shortening algorithms requiring matrix inversions. It converges faster than another blind adaptive channel shortening algorithm MERRY and can track channel variations within a symbol because it can update once per sample while MERRY updates once every symbol. SAM has higher complexity than MERRY. SLAM [5], 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.

Chapter 5 addresses the complexity reduction and convergence issues with SAM and SLAM algorithms. The main argument of this chapter is that effectively identical channel shortening can be achieved as SAM and SLAM whilst updating only half of the coefficients at each iteration which implies less computational complexity. The disadvantage of (PUSAM) and (PUSLAM) is that they can converge slower than the SAM and SLAM algorithms.

Chapter 6 proposed RPUSAM algorithm, which essentially achieves the same result in terms of reducing the effective channel length as SAM and PUSAM with less complexity. The complexity reduction is achieved by only updating $N/P$ of the coefficients of the TEQ at each iteration in a random pattern; on the other hand, PUSAM updates the subsets of coefficients in a systematic fashion which degrades convergence greatly over conventional SAM. The proposed algorithm is confirmed to achieve channel shortening on a set of eight CSA channels.

Also in this chapter a new partial update blind channel shortening algorithm was proposed. The proposed algorithm essentially achieves the same result in terms of reducing the effective channel length as

SLAM. Importantly, however, the disadvantage of SLAM in terms of the SIR performance has been overcome by the proposed algorithm where the proposed algorithm has the advantage of low complexity of SLAM over SAM and also has the advantage of SAM where a low lag-hopping sum-squared autocorrelation minimization (LHSAM) cost will be identical to a low SAM cost which guarantees to give a high SIR at the output of the matched filter as on the average the proposed algorithm uses all the lags as in SAM.

## 7.2 Future Research

Following the work which has been done in this thesis, a number of suggestions could be taken up as a possible future work in this area,

- Extend the application of the channel shortening algorithms presented in Chapters (4, 5 and 6) to upstream ADSL channels.
- Provide detailed convergence analysis of the proposed algorithms possibly based on an extention of the energy conservation principle [30].
- Develop faster converging versions of the proposed partial update algorithms using recursive least squares type formulations.
- Extend to case of complex data for application in multi-input multi-output systems.
- Consider application in distributed communication systems.

# BIBLIOGRAPHY

[1] E. E. Kuruoğlu, "Signal Processing in $\alpha$-Stable Noise Environments: An $L_p$-Norm Approach," Ph. D. Thesis, University of Cambridge, UK, 1998.

[2] K. Doğançay and O. Tanrikulu, "Adaptive filtering algorithms with selective partial updates," *IEEE Trans. on Circuits and Syst.II*, vol. 48, no.8, pp. 762–769, August 2001.

[3] P. J. W. Melsa, R. C. Younce, and C. E. Rohrs, "Impulse response shortening for discrete multitone transceivers," *IEEE Trans. Commun.*, vol. 44, no. 12, pp. 1662–1672, Dec. 1996.

[4] 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.

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

[6] J. M. Walsh, R. K. Martin, and C. R. Johnson, "Convergence and performance issues for autocorrelation based adaptive channel shorten-

ers," *Proc. 40th Asilomar Conf. on Signals, Systems, and Computers*, vol. 40, pp. 238–242, Nov. 2006.

[7] G. D. Forney, "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Trans. on Info. Theory*, vol. 18, pp. 363–378, May 1972.

[8] F. Magee, "A comparison of compromise viterbi algorithm and standard equalization techniques over band-limited channels," *IEEE Trans. on Comm.*, vol. 23, p. 361367, 1975.

[9] W. Younis and N. Al-Dhahir, "Joint prefiltering and MLSE equalization of space-time coded transmissions over frequency-selective channels," *IEEE Trans. on Vehicular Tech.*, vol. 51, p. 144154, 2002.

[10] N. Al-Dhahir and J. M. Cioffi, "Efficiently computed reduced-parameter input-aided MMSE equalizers for ML detection: a unified approach," *IEEE Trans. on Info. Theory*, vol. 42, no. 3, p. 903915, May 1996.

[11] D. D. Falconer and F. R. Magee, "Adaptive channel memory truncation for maximum likelihood sequence estimation," *Bell Sys. Tech. Journal*, pp. 1541–1562, Nov. 1973.

[12] D. Messerschmitt, "Design of a finite impulse response for the viterbi algorithm and decision-feedback equalizer," in *Proc. IEEE Int. Conf. on Commun.*, p. 37D.137D.5. Minneapolis, MN, June 1974.

[13] W. Lee and F. Hill, "A maximum-likelihood sequence estimator with decision-feedback equalization," *IEEE Trans. on Comm.*, vol. 25, no. 9, p. 971979, Sept. 1977.

[14] I. Medvedev and V. Tarokh, "A channel-shortening multiuser detector for DS-CDMA systems," in *Proceeding of the 53rd Veh. Tech. Conf.*, pp. 1834–1838. vol. 3, Rhodes, Greece, May 2001.

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

[16] S. I. Husain and J. Choi, "Single correlator based UWB receiver implemetation through channel shortening equalizer," in *2005 Asia-Pacific Conf. on Commun.*, pp. 610–614. Perth, Western Australia, Oct. 2005.

[17] M. Kallinger and A. Mertins, "Room impulse response shortening by channel shortening concepts," in *Proc. IEEE Asilomar Conf. on Signals, Systems and Comp.*, pp. 898–902. Pacific Grove, CA, Nov. 2005.

[18] J. A. C. Bingham, "Multicarrier modulation for data transmission: An idea whose time has come," *IEEE Commun. Magazine*, vol. 28, no. 5, pp. 5–14, May 1990.

[19] 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.

[20] 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.

[21] The European Telecomm. Standards Inst., "Radio Broadcasting

System, Digital Audio Broadcasting (DAB) to Mobile, Portible, and Fixed Receivers,". ETS 300 401, 19951997.

[22] 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.

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

[24] 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.

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

[26] R. Nawaz and J. Chambers, "Robust blind channel shortening in impulsive noise environments," *12th European Signal Processing Conf. (Eusipco), Vienna, Austria*, vol. 62, pp. 1931–1934, Sep. 2004.

[27] S. C. Douglas, "Analysis and implementation of the max-NLMS adaptive filter," *in Conf. Record od 29th Asilomar Conf. on Signals, Systems, and Computers*, vol. 1, pp. 659–663, October 1995.

[28] S. C. Douglas, "Adaptive filters employing partial updates," *IEEE Trans. on Trans Circuits Syst.*, vol. 44, no.3, pp. 209–216, March 1997.

[29] T. Aboulnasr and K. Mayyas, "Complexity reduction of the NLMS algorithm via selective coefficient update," *IEEE Trans. on Signal Processing*, vol. 47, no.5, pp. 1421–1424, May 1999.

[30] A. H. Sayed, *Adaptive Filters.* Hoboken, New Jersey, N.Y., USA: John Wiley & Sons, Inc., 2008.

[31] A. H. Khong, W.-S. Gan, P. A. Naylor, and M. Brookes, "A low complexity fast converging partial update adaptive algorithm employing variable step-size for acoustic echo cancellation," *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2008*, pp. 237 - 240, March 2008.

[32] P. A. Naylor and W. Sherliker, "A short-sort M-Max NLMS partial-update adaptive filter with applications to echo cancellation ," *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2003*, vol. 5, pp. 373 - 376, April 2003.

[33] A. H. Khong and P. A. Naylor, "Selective-Tap Adaptive Filtering With Performance Analysis for Identification of Time-Varying Systems ," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, pp. 1681 - 1695, July 2007.

[34] T. Schertler, "Selective block update of NLMS type algorithms," *in Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, ICASSP '98*, vol. 3, pp. 1717–1720, May 1998.

[35] S. Werner, M. L. R. de Campos, and P. S. R. Diniz, "Partial-update NLMS algorithms with data-selective updating," *IEEE Trans. on Signal Processing*, vol. 52, no.4, pp. 938–949, April 2004.

[36] M. Godavarti and A. O. Hero, "Partial update LMS algorithms," *IEEE Trans. on Signal Processing*, vol. 53, no.7, pp. 2382–2399, July 2005.

[37] R. K. Martin, J. Balakrishnan, W. A. Sethares, and C. R. Johnson, "A blind adaptive TEQ for muticarrier systems," *IEEE Trans. Signal Processing*, vol. 9, no. 11, pp. 341–343, Nov. 2002.

[38] R. K. Martin and C. R. Johnson, "Blind, adaptive, Per Tone equalization for multicarrier receivers," in *Conference on Information Sciences and Systems.* Princeton University, Mar. 2002.

[39] 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.

[40] 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.

[41] R. K. Martin, J. Balakrishnan, W. A. Sethars, and C. R. Johnson, "Blind, Adaptive Channel Shortening for Multicarrier Systems," *IEEE Asilomar Conf. on Signals, Systems, and Computers*, Nov. 2002.

[42] A. Scaglione, G. B. Giannakis, and S. Barbarossa, "Redundant Filterbank Precoders and Equalizers, Part I: Unification and Optimal Designs.," *IEEE Trans. Signal Processing*, vol. 47, pp. 1988–2006, July 1999.

[43] F. Romano and S. Barbarossa, "Non-Data Aided Adaptive Channel Shortening for Efficient Multi-Carrier Systems," *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, vol. 4, pp. 233–236, Apr. 2003.

[44] R. K. Martin and C. R. Johnson, "Adaptive equalization: transi-

tioning from single-carrier to multicarrier systems," *IEEE Trans. Signal Processing*, vol. 22, no.6, pp. 108–122, Nov. 2005.

[45] J. M. Walsh and J. C. R. Johnson, "Series Feedforward Interconnected Adaptive Devices," *IEEE International Conference on Acoustics, Speech, and Signal Processing Proceedings (ICASSP)*, vol. 2, pp. 445–448, May. 2004.

[46] Y. Lin, L. Liang, and S. Phoong, "A Semi-Blind Eigen Approach to Time-Domain Equalizer Design for VDSL Systems," *IEEE International Conference on Acoustics, Speech, and Signal Processing Proceedings (ICASSP)*, vol. 3, pp. 341–344, Mar. 2005.

[47] C. F. Gauss, "Gottingische gelehrte anzeigen," pp. 321–327, 1821. reprinted in Werke Bd. 4, pp. 98-117, 1880.

[48] A. Papoulis, *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill, 3rd edition, 1991.

[49] T. C. Chuah, "Robust Techniques For Multiuser CDMA Communications In Non-Gaussian Noise Environments," Ph. D. Thesis, University of Newcastle upon Tyne, UK, 2002.

[50] S. A. Kassam and H. V. Poor, "Robust techniques for signal processing: A survey," *IEEE Proc.*, vol. 73, pp. 433–481, 1985.

[51] B. Aazhang and H. V. Poor, "Performance of DS/SSMA communications in impulsive channels-Part I: Linear correlation receivers," *IEEE Trans. Commun.*, vol. 35, pp. 1179–1188, 1987.

[52] A. D. Spaulding, "Locally optimum and suboptimum detector per-

formance in a non-Gaussian interference environment," *IEEE Trans. Commun.*, vol. 33, pp. 509–517, 1985.

[53] T. K. Blankenship, D. M. Krizman, and T. Rappaport, "Measurements and simulation of radio frequency impulsive noise in hospital and clinics," in *IEEE Proc. Veh. Tech. Conf.*, pp. 1942–1946, 1997.

[54] P. L. Brocket, M. Hinich, and G. R. Wilson, "Nonlinear and non-Gaussian ocean noise," *J. Acoust. Soc. Am.*, vol. 82, pp. 1386–1394, 1987.

[55] J. H. Fennick, "Amplitude distributions of telephone channel noise and a model for impulse noise," *Bell Syst. Tech. J.*, vol. 48, pp. 3243–3263, 1969.

[56] D. Middleton, "Non-Gausian noise models in signal processing for telecommunications: New methods and results for class A and class B noise models," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1129–1149, 1999.

[57] J. Cook, "Wideband impulsive noise survey of the access network," *BT Technical Journal*, vol. 11, no. 3, pp. 155–162, 1993.

[58] I. Mann, S. Mclaughlin, W. Henkel, R. Kirkby, and T. Kessler, "Impulse generation with appropriate amplitude, length, inter-arrival, and spectral characteristics," *IEEE J. Select. Areas in Commun.*, vol. 20, no. 5, pp. 155–162, June 2002.

[59] "Asymmetric Digital Subscriber Line (ADSL) Metallic Interface," American National Standard T1.413-1995, printed from: The accompanying CDROM in [25].

[60] C. L. Nikias and M. Shao, *Signal Processing with Alpha-Stable Distributions and Applications.* New York: Wiley, 1995.

[61] H. V. Poor and M. Tanda, "Multiuser detection in impulsive channels," *Ann. Telecommun.*, vol. 54, pp. 392–400, 1999.

[62] K. Vastola, "Threshold detection in narrow-band non-Gaussian noise," *IEEE Trans. Commun*, vol. 32, pp. 134–139, 1984.

[63] A. T. Georgiadis, "Adaptive Equalisation for Impulsive Noise Environments," PhD thesis, Edinburgh University, UK, 2000.

[64] B. W. Stuck and B. Kleiner, "A statistical analysis of telephone noise," *The Bell System Technical Journal*, vol. 53, no. 7, pp. 1263–1320, Sep. 1974.

[65] B. W. Stuck, "Minimum error dispersion linear filtering of scalar symmetric stable processes," *IEEE Trans. Auto. Control*, vol. 23, pp. 507–509, 1978.

[66] J. G. Gonzalez, D. W. Griffith, and G. R. Arce, "Zero-order statistics: A signal processing framework for very impulsive environments," in *Proc. IEEE Signal Proc. Workshop on Higher Order Statistics.* Banff, Alberta, Canada, 1997.

[67] J. M. Chambers, C. L. Mallows, and B. W. Stuck, "A method for simulating stable random variables," *J. Amer. Stat. Assoc.*, vol. 71, pp. 340–346, 1976.

[68] G. Arslan and M. Ding and B. Lu and Z. Shen and B. L. Evans. MATLAB DMTTEQ Toolbox 3.1, Univ. Texas, Austin, May 10, 2003,

[Online.] Available: http://www.ece.utexas.edu/∼bevans/projects/adsl/dmtteq/dmtteq.html.

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

[70] P. G. Georgiou, P. Tsakalides, and C. Kyriakakis, "Alpha-stable modeling of noise and robust time-delay estimation in the presence of impulsive noise," *IEEE Trans. on Multimedia*, vol. 1, pp. 291–301, Sep. 1999.