



This item was submitted to Loughborough's Institutional Repository (<https://dspace.lboro.ac.uk/>) by the author and is made available under the following Creative Commons Licence conditions.

 **creative commons**
C O M M O N S D E E D

Attribution-NonCommercial-NoDerivs 2.5

You are free:

- to copy, distribute, display, and perform the work

Under the following conditions:

 **Attribution.** You must attribute the work in the manner specified by the author or licensor.

 **Noncommercial.** You may not use this work for commercial purposes.

 **No Derivative Works.** You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the [Legal Code \(the full license\)](#).

[Disclaimer](#) 

For the full text of this licence, please go to:
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

Random Partial Update Sum-Squared Autocorrelation Minimization Algorithm for Channel Shortening (RPUSAM)

M. Grira

Center of Digital Signal Processing
Cardiff University, Queen's Buildings
PO Box 925, Cardiff CF24 0YF
Wales, UK
Email: griram@cf.ac.uk

J. A. Chambers

Advanced Signal Processing Group
Dept. of Electronic and Electrical Engineering
Loughborough University
Leicestershire LE11 3TU, UK
Email: J.A.Chambers@lboro.ac.uk

Abstract—Partial updating is an effective method for reducing computational complexity in adaptive filter implementations. In this work, a novel 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 a partial update sum-squared auto-correlation minimization (PUSAM) algorithm with a deterministic coefficient update strategy. 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.

I. INTRODUCTION

In multicarrier modulation systems (MCM), 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 [1]. The CP is formed as 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) and to ensure the effective channel matrix is cyclic so that it is diagonalized by the discrete Fourier transform. 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 however increase the computational complexity, system delay, and memory requirements of the transceiver.

In practice, therefore, due to the potential variability of the impulse response length of the channel, to achieve reasonable efficiency a large v and large N have to be chosen. To overcome these problems a short adaptive time-domain equalizer (TEQ), usually a finite impulse response (FIR) filter, is typically placed ahead of the receiver. The purpose of this filter is to shorten the impulse response of the effective channel to be shorter than the length of the CP. A blind adaptive algorithm to design such a TEQ, called sum-squared auto-correlation minimization (SAM) was proposed in [2], 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 relatively high computational complexity as compared to simple adaptive algorithms such as least mean square (LMS) algorithm. In this work, we propose a new partial update algorithm which reduces the computational complexity of SAM by only updating a subset of the filter coefficients at each iteration. Those filter coefficients that are updated at each iteration are selected at random from the total set of filter coefficients so as to mitigate convergence problems found in partial update algorithms based on deterministic selection of the coefficients to update within the adaptive filter [3]. This is the first time that partial update adaptive filtering has been used in the context of blind adaptive channel shortening.

II. SYSTEM MODEL

The system model for blind adaptive channel shortening is shown in Figure (1). The input signal $x(n)$, typi-

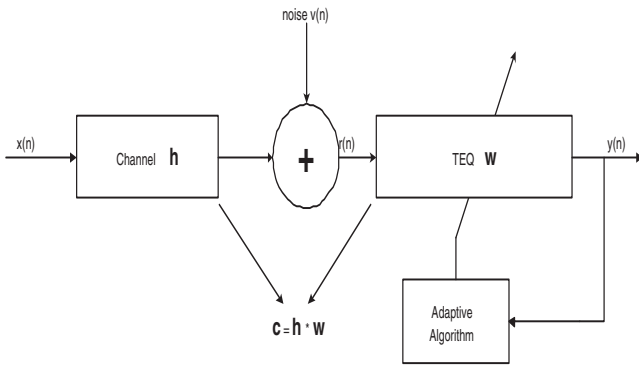


Fig. 1. System model for blind adaptive channel shortening.

cally drawn from a fixed constellation, and assumed for simplicity in development to be real in this work, is the source sequence to be transmitted through a linear finite-impulse-response (FIR) channel \mathbf{h} of length $(L_h + 1)$ taps. The received signal, $r(n)$, is filtered through an $(L_w + 1)$ tap TEQ with an impulse response vector \mathbf{w} to obtain the output sequence $y(n)$. We denote $\mathbf{c} = \mathbf{h} * \mathbf{w}$, where $*$ represents discrete time convolution, as the effective channel-equalizer impulse response vector of length $(L_c + 1)$ taps, where $L_c = L_w + L_h$. We assume that the relation $2L_c < N$ holds for multicarrier (or block-based) systems, i.e., the combined channel has length less than half the FFT (or block) size. The signal $v(n)$ is a zero-mean, i.i.d., discrete time noise sequence uncorrelated with the source sequence and has variance σ_v^2 . The received sequence $r(n)$ is

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

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

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

where $\mathbf{r}_n = [r(n) \ r(n-1) \ \dots \ r(n-L_w)]^T$ is the data vector of the TEQ and \mathbf{w} is the impulse response vector of the TEQ $\mathbf{w} = [w_0 \ w_1 \ w_2 \ \dots \ w_{L_w}]^T$, and $(\cdot)^T$ denotes vector transpose.

III. THE SAM ALGORITHM

The underlying idea that allows the development of SAM is that for the effective channel \mathbf{c} to have zero taps outside a window of size $(v+1)$, its autocorrelation values should be zero outside a window of size $(2v+1)$. In SAM the auto-correlation sequence of the combined

channel-equalizer impulse response is given by

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

and for a shortened channel, it must satisfy

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

The cost function J_{v+1} in SAM is then defined based upon minimizing the sumsquared auto-correlation terms, i.e.,

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

The update equation for the SAM algorithm based upon a block stochastic gradient descent minimization of (3) and a moving average estimate of the underlying statistics [4] can be written as can be seen in (1).

where k denotes the iteration index of the learning algorithm, and $\mathbf{w}(0) = \mathbf{e}_{L_w+1}$, where \mathbf{e}_m is a vector of zeros except for a unity value at element m , L_w is assumed odd for convenience. The computational complexity of this algorithm at each update iteration k is of the order of $3NL_w(L_c - v)$ multiplications and additions. The contribution in this work is to use a partial update strategy to reduce this complexity whilst retaining performance close to that of the original SAM algorithm.

IV. RANDOM PARTIAL UPDATE CHANNEL SHORTENING ALGORITHM

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 at each iteration. Our proposal here is to improve the deterministic partial update scheme to exploit improved convergence of random selection [5] which is particularly important when minimizing non quadratic and multimodal cost functions, and thereby achieve performance close to SAM.

The set of indices of the coefficients of the adaptive filter is given by $\{1, 2, \dots, L_w + 1\}$. This set is split into P different disjoint but equal size subsets denoted S_j , $j = 1, \dots, P$. 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 can be seen in (2)

where $M(j)$ is a diagonal 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 SAM.

$$\mathbf{w}(k+1) = \mathbf{w}(k) - 2\mu \times \sum_{l=v+1}^{L_c} \left[\left\{ \sum_{n=kN}^{(k+1)N-1} \frac{y(n)y(n-l)}{N} \right\} \times \left\{ \sum_{n=kN}^{(k+1)N-1} \left(\frac{y(n)\mathbf{r}(n-l) + y(n-l)\mathbf{r}(n)}{N} \right) \right\} \right] \quad (1)$$

$$\mathbf{w}^{k+1} = \mathbf{w}^k - 2\mu \times M(j) \times \sum_{l=v+1}^{L_c} \left[\left\{ \sum_{n=kN}^{(k+1)N-1} \frac{y(n)y(n-l)}{N} \right\} \times \left\{ \sum_{n=kN}^{(k+1)N-1} \left(\frac{y(n)\mathbf{r}(n-l) + y(n-l)\mathbf{r}(n)}{N} \right) \right\} \right] \quad (2)$$

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.

V. SIMULATION

The Matlab code at [6] was extended to simulate RPUSAM. The cyclic prefix was of length 32, the FFT size $N_{fft} = 512$, the TEQ had 16 taps and the channels were the test ADSL channel CSA loops 1-8 available at [7]. The noise was set such that $\sigma_x^2 \|\mathbf{c}\|^2 / \sigma_v^2 = 40dB$ where $\|\cdot\|$ denotes the Euclidean norm; and 75 OFDM symbols were used. The step size used was 5, and the stopping thresholds for learning were 0.0625 for RPUSAM, 0.0975 for PUSAM, and 0.062 for SAM [8], parameters chosen to determine when learning should cease as convergence has been reached. Four subsets ($P = 4$) were used in the RPUSAM algorithm. In Figures (2, 3, and 4), the performances of RPUSAM, PUSAM, and SAM 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 [1], which is obtained using the code at [7], as is the matched filter bound (MFB) on capacity, which assumes no ICI. In the top of the figures the achievable bits per second [4] 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. In the bottom plots the shortened channels are compared with the original channels and all algorithms are confirmed to be effective. Figure (5) shows the average performance of RPUSAM in term of shortening the channel for eight different CSA channels to make sure that our algorithm performs similarly with different channels.

VI. CONCLUSION

The 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

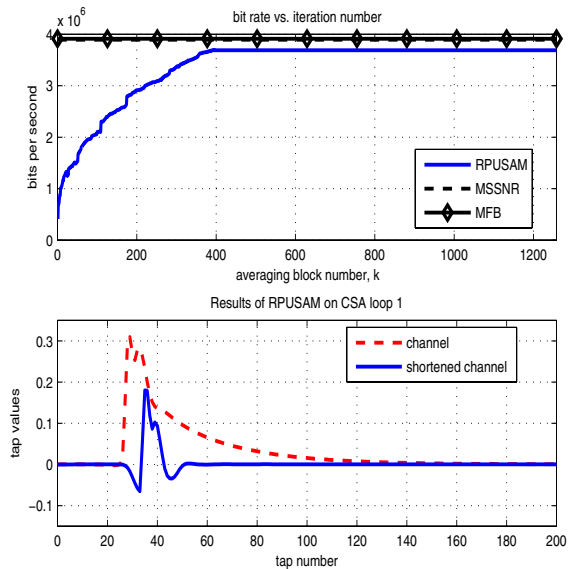


Fig. 2. A achievable bit rate and channel (dashed) and shortened channel (solid) impulse response of RPUSAM

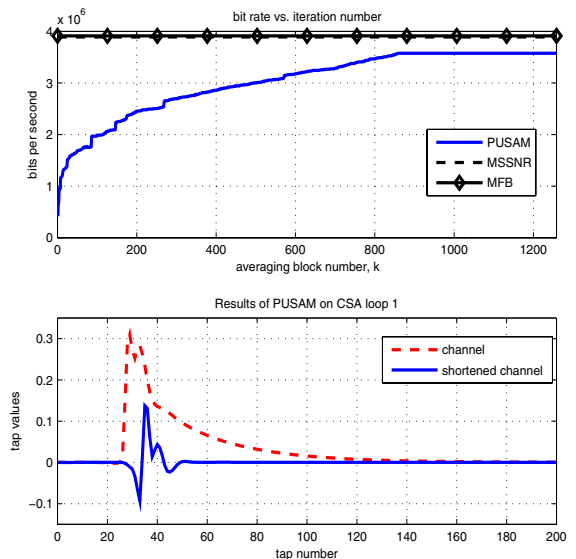


Fig. 3. A achievable bit rate and channel (dashed) and shortened channel (solid) impulse response of PUSAM

at each iteration in a random pattern; on the other hand, PUSAM updates the subsets of coefficients in a

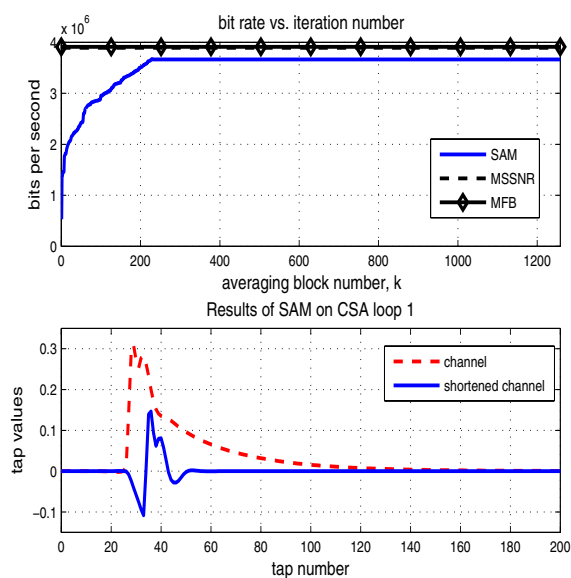


Fig. 4. A achievable bit rate and channel (dashed) and shortened channel (solid) impulse response of SAM

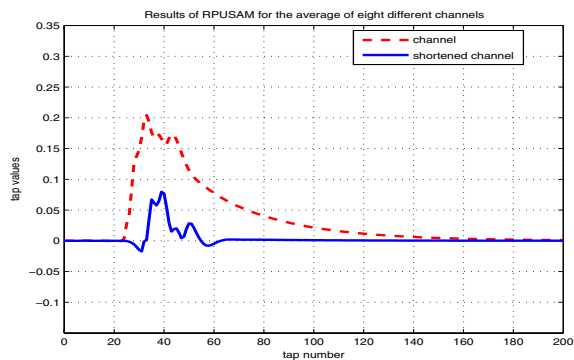


Fig. 5. Channel (dashed) and shortened channel (solid) impulse response for the average of eight CSA channels

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.

REFERENCES

- [1] J. W. Melsa, R.C. Younce and C.E. Rohrs, Impulse response shortening for discrete multitone transceivers, *IEEE Transactions on Communications*, (12) (1996) pp. 1662-1672.
- [2] J. Balakrishnan, R.K., Martin and C.R. Johnson Jr, Blind, adaptive channel shortening by Sum-Squared Auto-Correlation Minimization (SAM), *IEEE Transactions on Signal Processing*, (12) (2003) pp. 3086-3093.
- [3] M. Grira and J. A. Chambers, Partial Update Single Lag Autocorrelation Minimization for Channel Shortening, *IMA Mathematics in Signal Processing*, Cirencester, 18-20 December 2006, pp. 122-126.

- [4] R. K. Martin, Blind, Adaptive Equalization for Multicarrier Receivers, Ph. D. Thesis, Cornell University, US, 2004.
- [5] M. Godavarti and A. O. Hero, Partial Update LMS Algorithms, *IEEE Transactions on Signal Processing*, (4) (2005) pp. 2382-2399.
- [6] R. K. Martin, Matlab Code for Papers by R. K. Martin. [Online]. Available at: <http://bard.ece.cornell.edu/matlab/martin/index.html>.
- [7] G. Arslan, M. Ding, Lu, B., Z. Shen and B. L. Evans, TEQ Design Toolbox. Univ. Texas, Austin, TX. [Online]. Available at: <http://www.ece.utexas.edu/bevans/projects/adsl/dmtteq/dmtteq.html>
- [8] R. Nawaz and J. A. Chambers, Blind Adaptive Channel Shortening by Single Lag Auto-correlation Minimization, *Electronics Letters*, (25) (2004) pp.1609-1611.