Efficient Blind Channel Shortening using Stochastic Sum-squared Autocorrelation Minimization

W. G. Dalzell and C. F. N. Cowan

Digital Communications Research Group
ECIT, Queen's University of Belfast
Queen's Road, Belfast, BT3 9DT, UK
web: http://www.ecit.qub.ac.uk/

email : gdalzell01@qub.ac.uk
email: c.cowan@ecit.qub.ac.uk

***Abstract*— Channel-shortening is a generalization of transmission channel equalization; its purpose is to reduce the length of the impulse response of a channel. One application is to Multi-Carrier Modulation (MCM) transmission systems, where a Cyclic Prefix inserted between transmitted symbols prevents Inter-Symbol Interference, provided that the channel impulse response is sufficiently short. Communication channels that have non-stationary characteristics, such as those in mobile communication, are unsuited to equalization methods that utilise training sequences and significant initial computation. Instead, methods that are blind, continuously adaptive and not computationally intensive are to be preferred. Sum-squared Autocorrelation Minimization (SAM) is a blind channel-shortening algorithm suited to MCM, but one that has a high computational load. In this paper we describe a new stochastic modification of the SAM algorithm, which delivers a significantly reduced computational load whilst retaining the level of achievable bit-rate obtained by SAM, at a cost of reduced convergence speed.**

***Keywords* – Multi-Carrier Modulation, Channel Shortening, Equalisation, OFDM.**

I INTRODUCTION

Multi-carrier Modulation (MCM) is a transmission modulation scheme that utilizes simultaneously a number of "sub-carriers" of distinct frequencies within the transmitted signal. In the form of Orthogonal Frequency Division Multiplexing (OFDM) or Discrete Multi-Tone (DMT), it is in wide operational use, being the modulation scheme used by, for example, Digital Subscriber Line (DSL), Digital Audio Broadcasting (DAB) and IEEE802.11 wireless Local Area Network.

The use by MCM of a long multi-carrier symbol diminishes its sensitivity to Inter-Symbol Interference (ISI). In addition, MCM adds a Cyclic Prefix (CP), containing redundant data, to the multicarrier symbol at transmission. The CP, by introducing a "guard interval" between consecutive symbols, further reduces the sensitivity of MCM to ISI. Nonetheless, ISI remains a problem for MCM systems where the transmission channel has a delay spread greater than the signal's CP length. For example, this is the case for standard carrier-serving area test channels widely used for ADSL transmission systems, described later. Typically these test channels have delay-spreads with significant energy of greater than 50μs in duration, whilst the signal CP length is only 14μs.

To ameliorate the problem of ISI for MCM transmission by channel equalization, it is necessary only to reduce the length of the equalized channel to $v + 1$, where v is the length of the CP. This is referred to as "channel-shortening", rather than full equalization where the equalized channel impulse response is an impulse of length 1.

A substantial amount of research has been published proposing methods for channel-shortening. For example, in 1996 Melsa *et al* [1] proposed a method called Maximum Shortening Signal to Noise Ratio (MSSNR). The effective channel impulse response is identified using a training signal, and divided into a window (where it is desirable to maximize energy) and a wall (where minimum energy is desired). The equalizer is adjusted so as to maximize the ratio of the energy in the window to the wall. In the frequency domain, Van Acker *et al* [2] proposed a "per-tone" channel-shortening method. The shortening filter **w** is implemented after transforming the signal into the frequency domain, with one complex tap per sub-carrier.

The above methods require knowledge of a training sequence in the transmitted signal, with its attendant overhead. Such an overhead increases where the channel is dynamic, and the channel-shortener must be periodically or continually updated. These methods are more suited to relatively static communications channels, such as those of a telephone network used for Digital Subscriber Line signals. For such channels an initial equalization algorithm, perhaps of high computational load, with a training period is acceptable. If, in contrast, a channel characteristic can periodically and rapidly change (such as those of mobile networks), the equalization method must be adaptive, and the additional overheads of signal training sequences and heavy computational loads are highly undesirable.

A number of "blind" methods for channel-shortening—that is methods that operate using general characteristics of the signal rather than specific content—have been proposed. In [3] a Multicarrier Equalization by Restoration of Redundancy (MERRY) scheme is described that uses the redundancy of the CP values to infer the channel-shortener values. Another scheme, Sum-squared Autocorrelation Minimization (SAM), uses the expected low autocorrelation of a MCM signal [4].

The SAM scheme has an advantage over MERRY in that it can update the channel-shortener at a per-sample rate, whereas the nature of the MERRY algorithm allows it to update at the per-symbol rate only. However, the SAM algorithm brings a very high computational load to a receiver. This high computational load has received attention, with proposed methods to reduce it in [5], by partial-updating of the shortening filter, and in [6], by sub-sampling the autocorrelation lags.

Here, we also address the high computational load of SAM. We modify SAM to produce a new algorithm, such that the shortening filter **w** is updated stochastically using instantaneous autocorrelation values, substantially reducing the amount of computation in doing so, and examine how the algorithm behaves. In the following sections we describe the system model, the basic SAM algorithm, the stochastic modification of SAM and results obtained from simulations.

II SYSTEM MODEL

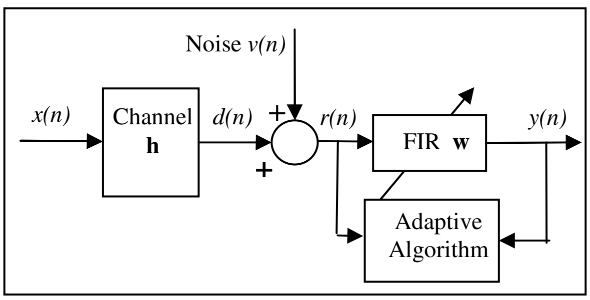

Fig. 1 Transmission System Model

The system model shown in Fig. 1 is the same model as used in previous related work, such as in [4] and [5]. The transmission channel **h** is represented as a linear finite impulse-response (FIR) filter of length L_h +1, and the channel-shortener **w** is an FIR filter of length L_w + 1. The input signal *x(n)* represents the MCM-modulated signal; added noise *v(n)* is uncorrelated with *x(n)*, is zero-mean and independent and identically distributed (i.i.d). All signals are modelled here as real.

The input data to the receiver, *r(n)*, and *y(n),* the output of the channel-shortener **w**, are given by:

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

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

The effective channel **c** is obtained by the discrete convolution of **h** and **w**, i.e. **c** = **h** * **w**; of length L_c + 1, where $L_c = L_w + L_h$.

III THE SAM ALGORITHM

The SAM algorithm operates according to the observation that a channel impulse response (IR) with zero autocorrelation for lags of greater than the signal CP length *v* will not introduce ISI. So for an effective channel to be shortened to a length of *v*+1, its IR autocorrelation should be low for autocorrelation lags greater than *v*. This leads to a cost-function for SAM: the total squared autocorrelations of the effective channel IR for lags greater than *v*:

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

where $R_{cc}(l)$ is the IR autocorrelation of **c** at a lag *l*.

This has no immediate utility, since **c** is unknown. However, in [4] the authors show that provided the transmitted signal is white, zero-mean and wide-sense stationary, autocorrelation of *y(n)* may reasonably be used instead of R_{cc}. To limit the impact of noise the signal energy must be significantly higher than noise energy or the signal CP must be longer than the shortening filter. They also show that it is preferable to meet the constraint $2.L_c < N_{FFT}$, where N_{FFT} is the size of the MCM signal FFT size. The practical cost-function then becomes:

$$J_{SAM} = \sum_{l=v+1}^{L_c} \left| R_{yy}(l)\right|^2$$

that may be directly evaluated from values of *y(n)*. The shortening filter **w** is updated using the "steepest-descent" method, and is then normalized to prevent adaptation to **w** = **0**, so that:

$$\hat{\mathbf{w}}_{k+1} = \mathbf{w}_k - \mu.\nabla_w [J_{SAM}], \qquad \mathbf{w}_{k+1} = \frac{\hat{\mathbf{w}}_{k+1}}{\|\hat{\mathbf{w}}_{k+1}\|}$$

The gradient term $\nabla_w [J_{SAM}]$ when evaluated using an average of N_{av} samples to obtain expected figures, is, assuming *y(n)* is real:

$$\nabla_w [J_{SAM}] = 2. \sum_{l=v+1}^{L_c} \left\{ \sum_{n=kN_{av}}^{(k+1)N_{av}-1} \frac{y(n).y(n-l)}{N_{av}} \right. \\ \left. . \sum_{n=kN_{av}}^{(k+1)N_{av}-1} \left(\frac{y(n).\boldsymbol{r}(n-l) + \boldsymbol{r}(n).y(n-l)}{N_{av}} \right) \right\} \quad (1)$$

It is this gradient term that causes the high computational cost of the SAM algorithm. For an Auto-Regressive (AR) implementation of the above average, updating **w** once per sample, the total cost is estimated in [4] to be $4.L_w(L_c–v)$ multiply-accumulate (MAC) operations, plus the normalization operations. If L_c = 100, L_w = 16 and v = 32, this requires approximately 4400 MAC operations per sample. At sample rates of 2Msample/s or greater, this represents a severe demand on real hardware available at the time of writing.

IV STOCHASTIC SAM

The intent of the Stochastic SAM algorithm is to reduce the computational load by removing averaging from (1), so that the gradient of J_{SAM} used to update **w** is evaluated from instantaneous autocorrelation values; errors due to use of the instantaneous values would then be averaged to zero over a large number of updates, given the limited size of individual adjustments to **w**.

A number of considerations apply to implementations of the stochastic method. Firstly, there is a computation problem in that an offset term occurs in the stochastic calculation. Inspection of (1) shows a sum of products of two averaged terms. Close examination of the two averaged terms will show that certain elements within them are correlated; however, the *expected* values of these elements are not correlated. A problem thus emerges when one attempts to use the instantaneous values of the elements, as in this stochastic algorithm, instead of the averaged values. The correlations then appear, manifested as incorrect offsets in the calculation.

A way of removing this offset effect is to prevent the correlation by retaining the averaging of one of the correlated elements, while the second remains instantaneous. Examination of (1) shows that of the two averaged terms that make up the product, the second has a much greater computation, and has thus more savings to offer when used in the instantaneous form. Accordingly the stochastic algorithm used here retains the averaged first term, as in (2). It might be described therefore as “partial-stochastic”.

The second consideration is that application of the stochastic method can plausibly be applied to the AR implementation of SAM, to a true moving-average version (both of which will update **w** once per sample), or to the average shown in (1) that updates **w** at the block average rate. On examination it turns out that the stochastic method does not provide a large computational saving to the AR or true moving-average methods. It is therefore here applied to the block average method of (1).

The stochastic version of the cost-function gradient then becomes (again assuming *y(n)* to be real):

$$\nabla_w (J_{StocSAM}) = 2. \sum_{l=v+1}^{L_c} \sum_{n=kN_{av}}^{(k+1)N_{av}-1} \left(\frac{y(n).y(n-l)}{N_{av}} \right) \\ .\big(y(n).\boldsymbol{r}(n-l) + \boldsymbol{r}(n).y(n-l)\big) \quad (2)$$

A new value will be calculated every N_{av} samples. The total number of MAC operations for an update is $(N_{av} + 4.L_w)(L_c–v)$, approximately.

For the same example system as for SAM above the computational load is approximately 200 MAC operations, per sample, for N_{av} = 32. This may be compared to the SAM figure of 4400 MAC operations. (Note that these figures assume *y(n)* to be real—the figures for complex *y(n)* are higher, but the ratio of the counts for SAM and Stochastic SAM remains the same.)

V SIMULATION RESULTS

The Stochastic SAM algorithm was simulated to evaluate its performance, using the model code available at [7] as a basis. As with previous work, the symbol FFT size is 512 samples and the CP-length 32. The 8 ADSL test channels CSA 1-8 defined in [8], and available from [9], were used. Near-end Crosstalk (NEXT) noise was selected to be the additive noise source, the results shown here all being for an SNR level of 45dB. The shortening filter has 16 taps, and is initialized as a single impulse at tap 8.

The basic SAM algorithm, using AR averaging, was simulated as a benchmark against which the performance of the stochastic algorithm could be compared. The filter update step-size μ was selected for both algorithms to obtain rapid convergence while remaining stable. For basic SAM, μ=40, and for the stochastic algorithm, μ=5; with these values the algorithms produced stable solutions over the SNR range tested, 30dB to 50dB. The averaging block size N_{av} was set to 32.

The main measure of performance is Achievable Bit-Rate (ABR) rather than Bit-Error Rate, since MCM signals typically adapt the transmitted bit-rate to that which is achievable given the SNR of the current channel. ABR is evaluated as in [4]. The speeds of convergence of the two algorithms were compared for the test channels, CSA Loops 1-8. Results shown here are for one channel only, CSA Loop 3, which is representative of the others.

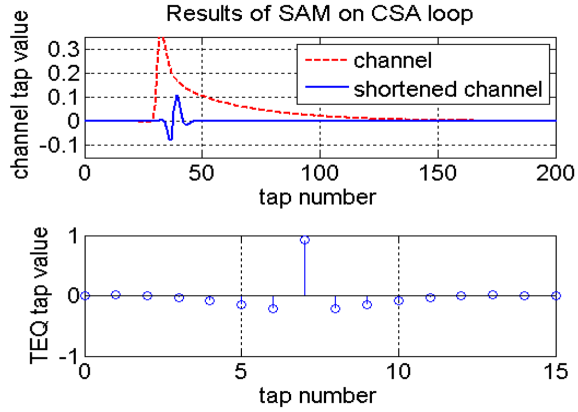

Fig. 2 Impulse Responses of CSA Loop 3, the Effective Channel after shortening, and the Shortening Filter

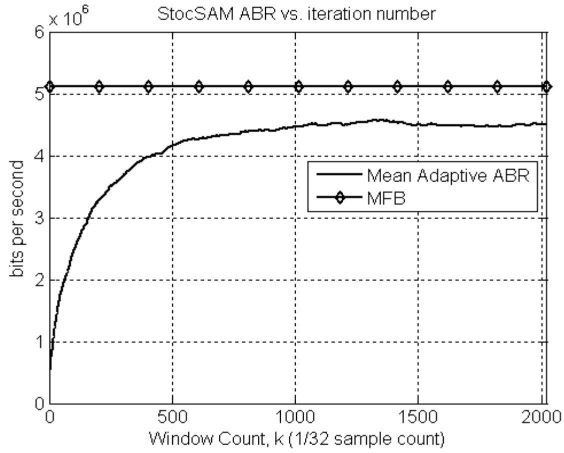

Fig. 4 Stochastic SAM algorithm: Simulated runs of 120 symbols of CSA Loop 3 –Achievable Bit-Rate

Figure 2 shows the channel IR for CSA Loop 3, where energy in the response of over 100 samples in duration may be observed. A typical IR of the channel after it has been shortened and a typical IR of the shortening filter (i.e. Time-domain Equalizer, TEQ) are also shown.

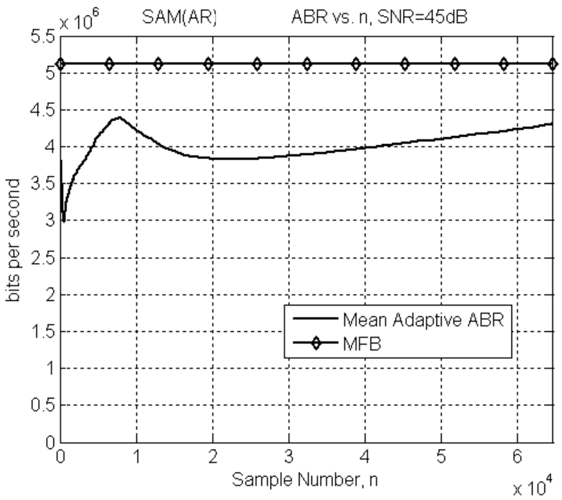

Fig. 3 Basic SAM algorithm: Simulated runs of 120 symbols, CSA Loop 3 –Achievable Bit-Rate

Figure 3 and Figure 4 show the convergence of the SAM and Stochastic SAM algorithms for simulation runs each of 120 symbols' duration. For each algorithm, the mean of 32 individual sample runs is shown, and the Matched-Filter Bound (MFB) for this SNR is included as a reference. Note that for Stoc SAM the x-axis indicates the averaging block count k, where 32 samples are in 1 block, and 1 symbol is 544 samples.

General observations from the results show (i) a significantly lower convergence speed of the stochastic algorithm, (ii) that both algorithms exhibit a performance spread between the sample runs, and (iii) no settled level bit-rate is apparent after 120 symbols (though the latter two are not visible from the graphs).

An interesting characteristic of the SAM algorithm is its long-term behaviour. Figure 5 shows the achievable bit-rate of a significantly longer simulated run of 12,000 symbols of the basic SAM algorithm, and Figure 6 the cost-function J_{SAM}, this time using a log time axis. Two behaviours are apparent. First, the cost-function is continuously reducing without converging to a settled level. Second, there is no bit-rate at which the system settles; instead the bit-rate appears to wander within limits of a converged zone. Since for SAM the achievable bit-rate is a non-linear function of the cost-function, it is not known how the bit-rate behaviour will continue beyond the end of this simulation, as the cost-function continues to change.

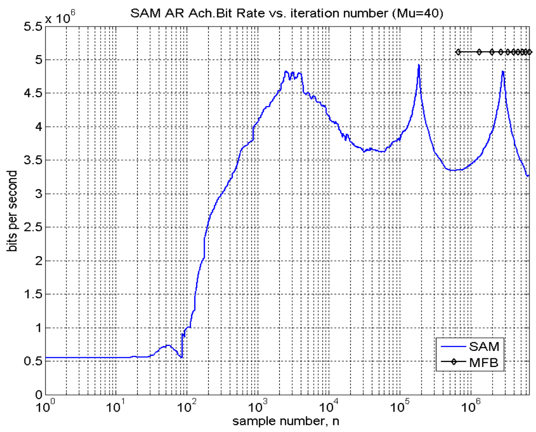

Fig. 5 Basic SAM algorithm: Simulated run of 12,000 symbols of CSA Loop 3 - Achievable Bit-Rate

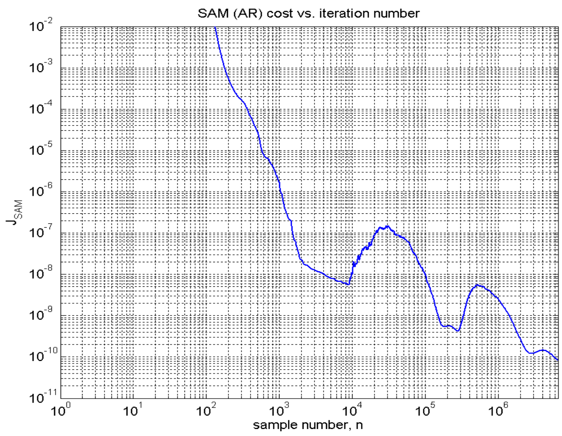

Fig. 6 Basic SAM algorithm: Simulated run of 12,000 symbols of CSA Loop 3 – Cost-Function J_{SAM}

Figures 7 and 8 show results of a similar run of the Stochastic SAM algorithm, this time with a duration of 120,000 symbols. The same features of continuously reducing cost-function and wandering of the bit-rate after initial convergence may be observed, just as with the basic SAM algorithm. Differences between the two algorithms are also apparent. The achievable bit-rate of the stochastic algorithm exhibits a noisier or jittery characteristic after initial convergence, although its average behaviour is also that of wandering. It is slower to converge—the cost-function value for the stochastic algorithm at the simulation end is similar to that of the basic algorithm (approx 10^{-10}), even though its simulated duration was 10 times longer.

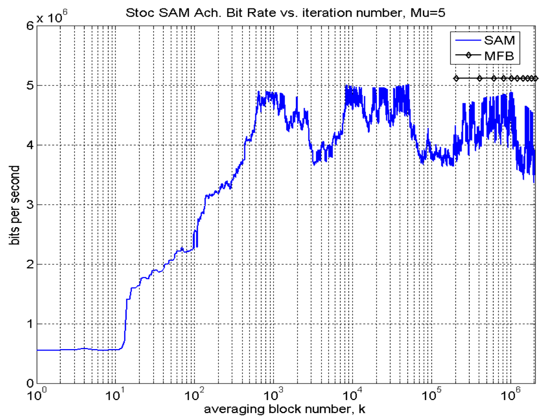

Fig. 7 Stochastic SAM algorithm: Simulated run of 120,000 symbols of CSA Loop 3 - Achievable Bit-Rate

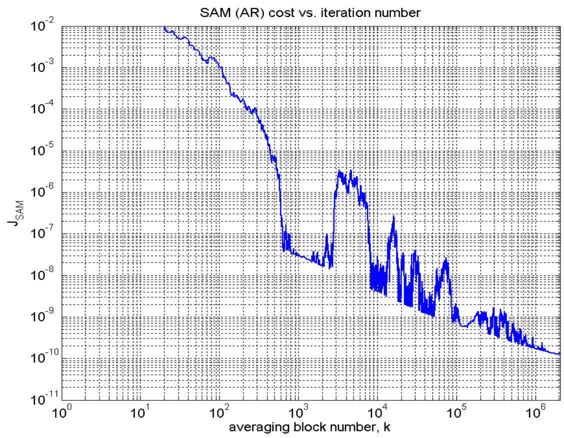

Fig. 8 Stochastic SAM algorithm: Simulated run of 120,000 symbols of CSA Loop 3 – Cost-Function J_{SAM}

Larger numbers of simulated sample runs over all channels were completed. The wandering behaviour of the achievable bit-rate after initial convergence occurred in all runs, both of the basic algorithm and the stochastic algorithm, and no difference was observed between the algorithms in the limits within which the wandering occurs. The results therefore show that achievable bit-rate of the stochastic algorithm may be considered to be the same as that of the basic algorithm.

The comparison of the convergence speed of the two algorithms showed the stochastic algorithm to be slower than the basic algorithm by a factor that lies between 30 and 40, depending on the comparison criterion used.

VI CONCLUSION

The research described here is motivated by the need to develop blind, adaptive and low-computation channel-shortening algorithms for non-stationary communication channels carrying MCM signals.

The results show the new Stochastic SAM algorithm has substantially reduced computational cost relative to that of the basic SAM algorithm, while retaining the achievable bit-rate, at a penalty of slower convergence. In the example used computational cost is reduced by a factor of approximately 20, and the convergence speed is reduced by a factor in the range of 30 to 40.

The computational cost remains high notwithstanding the gains made here; in this example approximately 200 MAC operations are required per sample by the stochastic algorithm. A potentially fruitful direction to consider is the combination of Stochastic SAM with other computational-cost reduction methods, such as partial-update [5] and lag-hopping [6], to further reduce the cost.

The SAM algorithm cost-function surface has non-global minima, and is acknowledged to deliver lower performance than other algorithms [10]. The observation here that long-term bit-rate behaviour is uncertain is a further concern for practical implementations. Nevertheless, its fast performance remains attractive. Use of the higher convergence speed of SAM in a hybrid system with other blind algorithms may be an appropriate practical exploitation of its strength, particularly if the speed can be retained while the computational cost is reduced. Such a system will then be a step towards the blind, adaptive and low-computation equalization algorithm required for mobile systems.

Acknowledgements: This work is funded by the Northern Ireland Department for Employment and Learning (DEL).

VII REFERENCES

[1] P. J. W. Melsa, R. C. Younce, and C. E. Rohrs. "Impulse Response Shortening for Discrete Multitone Transceivers." *IEEE Trans. on Comm.*, 44:1662-1672, December 1996.

[2] K. Van Acker, G. Leus, M. Moonen, O. Van de Weil, and T. Pollet. "Per-Tone Equalization for DMT-Based Systems". *IEEE Trans. on Comm.,* 49(1):109-119, January 2001.

[3] R. K. Martin, J. Balakrishnan, W. A. Sethares and C. R. Johnson, Jr. "A Blind, TEQ for Multicarrier Systems." *IEEE Signal Processing Letters*, 9(11):341-343, November 2002.

[4] J. Balakrishnan, R. K. Martin, and C. R. Johnson, Jr. "Blind, Adaptive Channel Shortening by Sum-squared Auto-correlation Minimization (SAM)." *IEEE Trans. on Signal Processing*, 51(12):3086-3093, December 2003.

[5] M. Grira, J. A. Chambers, "Random Partial Update Sum-Squared Autocorrelation Minimization Algorithm for Channel Shortening (RPUSAM)", *Proc. 3rd Intl. Symposium on Communications, Control and Signal Processing*, St. Julians, Malta, March 2008, pp. 1400-1403

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

[7] R. K. Martin, Matlab Code by R. K. Martin. [Online]. Available at: http://bard.ece.cornell.edu/matlab/martin/index.html

[8] G. Arslan, B. L. Evans, and S. Kiaei. "Equalization for Discrete Multitone Receivers To Maximize Bit Rate." *IEEE Trans. on Signal Processing*, 49(12):3123-3135, December 2001

[9] G. Arslan, M. Ding, Lu, B., Z. Shen and B. L. Evans, TEQ Design Toolbox. Univ. Texas, Austin, TX. [Online]. Avail. at: http://www.ece.utexas.edu/~bevans/projects/adsl/dmtteq/

[10] R. K. Martin, K. Vanbleu, M. Ding, G. Ysebaert, M. Milosevic, B. L. Evans, M. Moonen, and C. R. Johnson, Jr., "Unification and Evaluation of Equalization Structures and Design Algorithms for Discrete Multitone Modulation Systems," *IEEE Trans. on Signal Processing*, vol. 53, no. 10, pp. 3880–3894, Oct. 2005