FAST BLIND CHANNEL SHORTENING USING A PREDICTION-ERROR FILTER AIDED BY 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
phone: +(44) 28 9097 1700, fax: +(44) 28 9097 1702, email: gdalzell01@ qub.ac.uk, c.cowan@ ecit.qub.ac.uk

ABSTRACT

A hybrid algorithm for blind adaptive channel-shortening of ADSL communication channels is here proposed. The prediction-error filter is a well-known technique that can equalize minimum-phase channels for Multi-Carrier Modulation (MCM) modulated signals. Another well-known algorithm, Sum-Squared Autocorrelation Minimization (SAM), also suited to blind adaptive channel-shortening of MCM signals, is used to aid the prediction-error filter. SAM exhibits fast convergence, but has high computational cost and an unstable behaviour. The objectives of the hybrid algorithm are fast convergence and stable steady-state behaviour for modelled ADSL channels from one channel-shortening algorithm; we show the performance of the hybrid fulfils the objectives.

Index Terms— Multi-Carrier Modulation, Channel-Shortening, Equalization, Linear Predictor, Prediction-Error Filter

1. INTRODUCTION

Communications channel equalization may be performed "blind", i.e. without knowledge of the transmitted signal data content, by use of expected general properties of a signal that are "restored" by an equalization process.

MCM schemes employ mutually-orthogonal sub-carriers, resulting in signals with the property (that may be restored) of low auto-correlation. By including a Cyclic Prefix (CP) between symbols, there is mitigation of inter-symbol interference (ISI). Since a CP absorbs channel bandwidth, partial channel equalization—channel shortening—remains of interest to minimize the CP length.

The SAM algorithm [1] is a well-known method of shortening MCM channels, which explicitly uses the low-autocorrelation property. The Linear Predictor in the form of a Prediction-Error Filter is also well-known as an equalizer, implicitly using signal autocorrelation. It may be configured to channel-shorten, rather than equalize, by extending the prediction delay [2]. The predictor accurately equalizes channel poles, but does not converge as quickly as SAM.

Here, we propose a channel-shortener for ADSL channels composed of a prediction-error filter aided by a SAM shortening filter. The intent is to obtain the strengths of each algorithm while minimizing the weaknesses—thus obtaining the high convergence speed of SAM while reducing its high computational cost, and obtaining the steady converged performance of the predictor rather than the unstable behaviour of SAM. ADSL channels, the target application for shortening, are mainly minimum-phase to which SAM and a forward linear predictor are suited—unlike wireless channels that can readily be mixed-phase. The ADSL test channels are those used by previous research in this area (e.g. [1], [2], [6], [10]).

2. THE LINEAR PREDICTOR

The Linear Predictor is an algorithm that predicts the value of a sample of a sequence from a linear combination of the other samples; a Forward Linear Predictor (FLP) predicts the most recent sample of the sequence from older samples. For an input signal x, the predicted value of $x(n)$, $\hat{x}(n)$, as described in [3] is given by:

$$\hat{x}(n) = -\sum_{k=1}^{L} a_p(k)x(n-k)$$

and the prediction error is:

$$\varepsilon_p(n) = x(n) - \hat{x}(n) = x(n) + \sum_{k=1}^{L} a_p(k)x(n-k)$$

Solutions of the coefficients may be static or adaptive; an adaptive solution may be engineered using the steepest-descent method or the least-mean-squares (LMS) method, in both cases minimizing the mean-square of $\varepsilon_p(n)$.

Implicitly, the predictor detects and removes autocorrelation of the input signal, and is often referred to as a *whitening filter*. It performs channel equalization as follows. If an independent and identically distributed (i.i.d) data sequence $d(n)$ is passed through a linear channel $\mathbf{h}$, its output $x(n)$ becomes autocorrelated depending on the impulse response (IR) of $\mathbf{h}$. The predicted output $\hat{x}(n)$ is obtained from x using coefficients derived from the observable autocorrelation of x; thus $\varepsilon_p(n) = x(n) - \hat{x}(n)$ converges towards $x(n)$ with the correlation terms introduced by **h** removed. The predictor may therefore be seen as an equalizer of **h**, where the prediction error signal $\varepsilon_p(n)$ is the system output, (i.e. a Prediction-Error Filter), provided the original transmitted signal has low autocorrelation.

An example of the use of a prediction-error filter is as part of an acoustic equalization scheme in [4]. A more complex, oversampling, version of the linear predictor exploiting cyclostationarity has been proposed in [5], allowing phase information to be retained by the equalizer.

The zeros of a forward linear predictor will be minimum-phase; similarly the zeros of backward linear predictor (BLP) will be maximum-phase [3]. The consequence is that a prediction-error filter will be unable to fully equalize a mixed-phase channel; an FLP will introduce minimum-phase zeros where there should be maximum-phase zeros, and a BLP will introduce maximum-phase zeros where there should be minimum-phase zeros. For ADSL channels, mixed-phase but mainly minimum-phase, the prediction-error filter based on an FLP will equalize effectively, as demonstrated for a set of representative ADSL test channels [2].

Further, the predictor may readily be made to perform channel shortening by adding additional delay to the filter input signal [2]. Introducing the delay means the autocorrelation between samples separated by less than the delay is not used in establishing the filter coefficients, and so autocorrelation for lags less than the delay will not be removed. Thus, the channel impulse response will not be equalized for terms of less than the delay.

The computation cost of an adaptive LMS-type solution to the predictor filter coefficients is $2.L_{wLP}$ Multiply-Accumulate (MAC) operations per update*,* where L_{wLP} is the number of active filter elements. This is substantially lower than for SAM.

3. SUM-SQUARED AUTOCORR. MINIMIZATION

The SAM algorithm [1] explicitly uses the low expected autocorrelation of an MCM signal as the general property to restore. The SAM shortening filter, $\mathbf{w_s}$, is adapted using the steepest descent method. The SAM cost function is the total squared autocorrelation of the filter output $y(n)$ for lags greater than the signal CP length v:

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

where $R_{yy}(l)$ is the autocorrelation of $y(n)$ at a lag l. The mechanism for channel-shortening is the ignoring of signal autocorrelation for lags of less than v. The filter update then is $\hat{\mathbf{w}}_{S,n+1} = \mathbf{w}_{S,n} - \mu_S.\nabla_{wS}\left[J_{SAM}\right]$, and $\mathbf{w_s}$ is normalized upon update to prevent adaptation to **0**. Obtaining $\nabla_{wS}\left[J_{SAM}\right]$ is computationally complex, described in [1].

SAM has the advantage of rapid convergence. It has two disadvantages that are relevant here. First, it has a very high computational cost. For an Auto-Regressive (AR) implementation updating the filter at the sample-rate, the cost is estimated in [1] to be $4.L_w(L_h-v)$ multiply-accumulate (MAC) operations, plus normalization operations, where L_w is the filter length and L_h is the channel impulse-response length. If $L_h = 100$, $L_w = 16$ and $v = 32$, approximately 4400 MAC operations per sample are required to adapt the filter.

Some modifications to SAM have been proposed to reduce its cost (e.g. Lag-hopping [6] and Stochastic SAM [7]), but they also reduce its convergence speed.

The second disadvantage may be observed in the behaviour of SAM in Figure 1, showing the achievable bit-rate (ABR) of an MCM signal in a modelled ADSL channel (Ch3), for a test scenario of duration of $3.2\text{x}10^6$ samples (6000 symbols) and SNR=40dB. After a rapid convergence to near-MFB (Matched-Filter Bound) performance, the ABR does not settle at a stable level, but "wanders" between peaks and lower values in a cyclic manner.

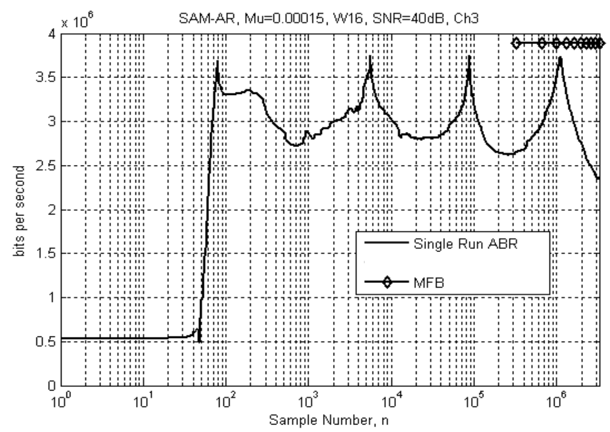

Figure 1 SAM Bit-Rate Wandering

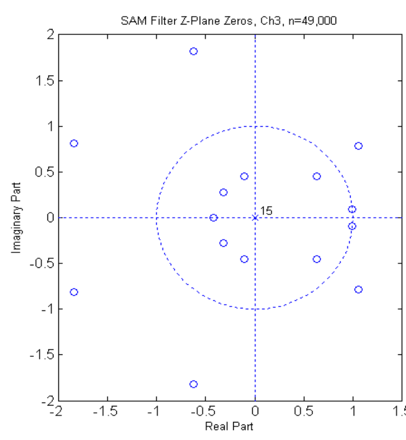

Figure 2 SAM Filter z-plane Zeros

Observation shows this unstable ABR effect to occur for each modelled channel, and that the cause is the occurrence in the shortening filter of z-plane zeros near the unit circle (UC) at lower frequencies. The zeros cause dips in the frequency response of the effective channel, lowering the SNR and traffic capacity of subcarriers in the vicinity. It is assumed here that the cause of the UC zeros relates to the multi-modal nature of the SAM cost-function surface.

An example is shown in Figure 2. For the test run of Figure 1, the filter z-plane zeros are displayed for the snapshot in time n=49,000, corresponding to a dip in ABR. Two zeros marginally inside the UC occur. These partially equalize one of the poles of the channel, but additionally cause a notch in the frequency response. As a result about 50 low-frequency sub-carriers of the transmitted signal have substantially reduced data capacity, zero in some sub-carrier instances.

4. SAM-AIDED PREDICTION-ERROR FILTER

The prediction-error filter converges to a stable channel equalizer, effective for the mainly minimum-phase ADSL channels, but with convergence speed significantly less than SAM. The motive in this work is to consider a filter that exploits both the speed of SAM and the steady solution of the predictor, retaining the stable equalization but to use SAM to increase the convergence speed.

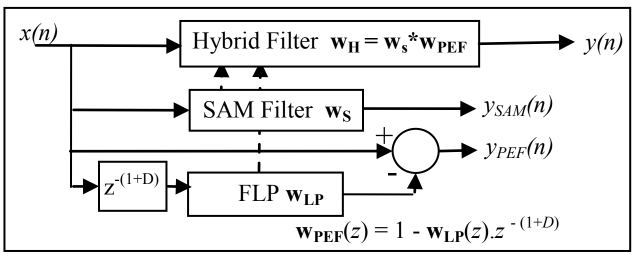

Figure 3 Hybrid FLP/SAM Shortener

The SAM algorithm may aid the FLP shortener in a number of plausible ways. In this work two methods were found to be effective: (i) a shortening filter that is the convolution of independent SAM and FLP filters; (ii) an FLP filter in series with a SAM filter. The results of method (i) will be described here.

The scheme is shown in Figure 3, where a SAM filter $\mathbf{w_S}$ and a predictor filter $\mathbf{w_{LP}}$ are updated independently, and the filters then combined to make the working hybrid filter $\mathbf{w_H}$. The SAM filter is updated and normalized per sample as:

$$\widehat{\mathbf{w}}_{S,n+1} = \mathbf{w}_{S,n} - \mu_S . \nabla_{wS} [J_{SAM}], \qquad \mathbf{w}_{S,n+1} = \frac{\widehat{\mathbf{w}}_{S,n+1}}{\left\| \widehat{\mathbf{w}}_{S,n+1} \right\|}$$

using the AR method described in [1], where μ_S is the SAM step coefficient and $\nabla_{wS} [J_{SAM}]$ is the vector gradient of the SAM cost function w.r.t. the filter coefficients.

The prediction error filter output is:

$$y_{PEF}(n) = x(n) - \sum_{k=1+D}^{L_{wLP}} w_{LP}(k) x(n-k)$$

where $y_{PEF}(n)$ is both the filter output and the prediction error, L_{wLP} is the filter length, and D is the additional predictor input delay described in [2] for channel shortening. The predictor filter update, using an LMS-type stochastic method, is:

$$\mathbf{w}_{\mathrm{LP,n+1}} = \mathbf{w}_{\mathrm{LP,n}} + 2.\mu_{LP}.y_{PEF}(n).\mathbf{x}(\mathrm{n} - (1+D))$$

The effective prediction error filter is then (in z-transform form): $\mathbf{w_{PEF}}(z) = 1 - \mathbf{w_{LP}}(z).z^{-(1+D)}$, and the hybrid filter $\mathbf{w_H}$ is: $\mathbf{w}_{\mathbf{H},n+1} = \mathbf{w}_{\mathbf{S},n+1} * \mathbf{w}_{\mathbf{PEF},n+1}$, where "*"= convolution.

Channel-shortening may be explicitly included in the hybrid filter by introducing a "don't-care" region in the predictor, using values of the delay D greater than zero. It should also be noted that use of both SAM and predictor filters will introduce redundancy, so that additional zeros will occur in the hybrid filter that lengthen the effective channel. Finally, any maximum-phase component of the channel will not be equalized, but the predictor will introduce minimum-phase zeros for that component. The equalized channel impulse response will thus include, in addition to the channel's equalized terms, (i) non-equalized maximum-phase components and resulting minimum-phase zeros, (ii) redundant zeros where SAM and the predictor overlap and (iii) the explicit "don't care" terms related to the delayed predictor. Because the introduction to the predictor of extra delay reduces its ability to equalize, the shortening prediction delay D is here kept to a value of 4 taps, much less than the CP length.

There is scope to reduce the size of the SAM filter, and make significant computation cost savings. The main function of SAM in this scheme is to make an initial rapid convergence, and a small number of taps can do this for a limited portion of the channel transfer function.

5. SYSTEM MODEL AND SIMULATION

The system model is the same as foregoing work such as [1] and [2], shown in Figure 4. The transmission channel **h** is represented as a linear finite impulse-response (FIR) filter of length L_h +1. The input signal $d(n)$ is the MCM-modulated signal; the added noise $v(n)$ is uncorrelated with the channel output, is zero-mean and i.i.d.; signals are modelled here as real. The effective channel **c** is obtained as **c** = **h** * **w**; of length $L_c + 1$, where $L_c = L_{wH} + L_h$.

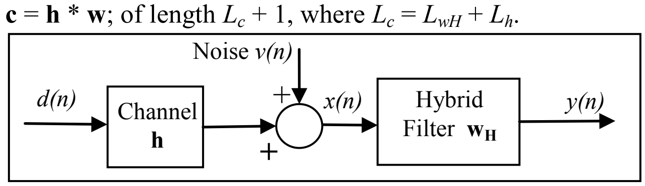

Figure 4 Transmission System Model

General aspects of the model remain as with related previous work. The modelled MCM signal symbol FFT size is 512 samples and the CP-length 32; the signal, being real, thus has 256 sub-carriers. The 8 ADSL model test channels CSA 1-8 were used, available from [9] and described in [10], chosen to allow comparison with earlier work such as [1]. The channels are mixed-phase with mainly minimum-phase components. Near-End Crosstalk (NEXT) noise is the additive noise source type, and a range of Signal-to-Noise (SNR) values from 40dB to 20dB used.

The model code available from [8] was the original basis for this work. (A correction is here included in the signal-to-noise calculation, true modelled SNR is higher than the value indicated in the earlier code, such as used in [2].)

Achievable Bit-Rate (ABR) is the measure of effectiveness, since MCM signals typically adapt the bit-rate to that which is achievable given the SNR of the current channel; it is evaluated as in [1]. Results shown here are for SNR values of 40dB and 20dB. The selected filter lengths are L_{wS} = 7 taps, L_{wLP} = 14 taps (including D). The value chosen for D is 4 taps. For SNR=40dB: $\mu_S = 1.5\text{x}10^{-4}$, $\mu_{LP} = 2\text{x}10^{-5}$; for SNR=20dB: $\mu_S = 7.5\text{x}10^{-5}$, $\mu_{LP} = 1\text{x}10^{-5}$. The centre-tap of $\mathbf{w_s}$ is initialized to 1, $\mathbf{w_{LP}}$ is initialized to $\mathbf{0}$.

The computation cost in MAC-operations per update to adapt the filter is $2.L_{wLP}$ (Predictor), $4.L_{wS}(L_h{-}v)$ (SAM), and $L_{wLP}.L_{wS}$ (convolution). Assuming L_h = 100 and v = 32, the total is 1994 MAC-operations/sample. The filter is updated once per sample.

4. SIMULATION TESTS AND RESULTS

Selected results are shown here for ADSL test channel 3, which are representative of results for all 8 test channels.

In Figure 5 and Figure 6 ABR is shown for short runs of 120 symbols, for SNR of 40dB and 20dB respectively. The ABR of the hybrid solution under test is compared with ABR for a SAM filter (16-tap) and a delayed prediction-error-filter (10-taps plus 4-tap delay), with the Matched-Filter Bound (MFB) for the channel shown as a limit. The values shown are the mean value for 16 runs of each type. Figure 7 shows the channel and equalized IRs, plus the equaliser IR and its component IRs. The ABR of a longer-duration scenario (6000 symbols) of the aided solution, for a single run at 40dB, is shown in Figure 8. (Note: 544 samples = 1 Symbol.) Observations from the results are:

Convergence Speed. The aided solution converges slightly more slowly than the SAM solution, but significantly more quickly that the unaided predictor. The convergence time described here is the time taken for ABR to reach 80% of its steady value.

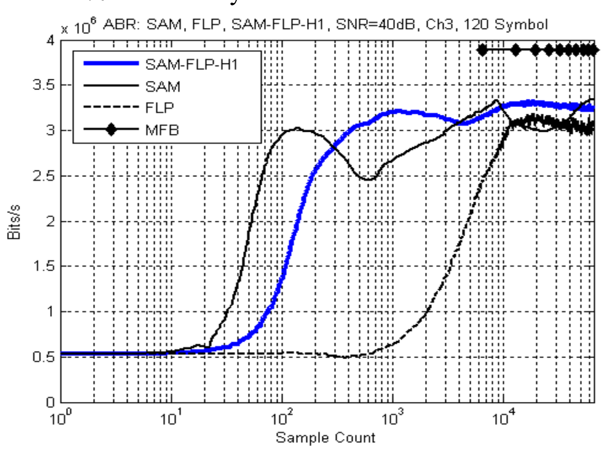

Figure 5 40dB – ABR Algorithm Comparison

At SNR=40dB, the ABR convergence time is less than the duration of one symbol—typically about half of a symbol. The improvement factor over the unaided predictor varies between 15 and 40, averaging 25 over the 8 test channels. At SNR=20dB the convergence time for most channels remains less than one symbol duration, and the improvement factor ranges from 10 to 40. For 2 particular channels, the convergence is markedly slower, of about 8 symbols' duration, and only twice the speed of the unaided solution. The average improvement factor is 16. The improvement due to aiding is thus clear at both 40dB and 20dB, though it is less at the lower SNR value.

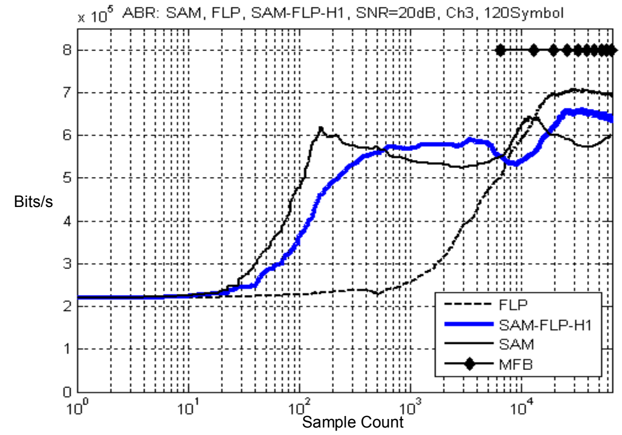

Figure 6 20dB – ABR Algorithm Comparison

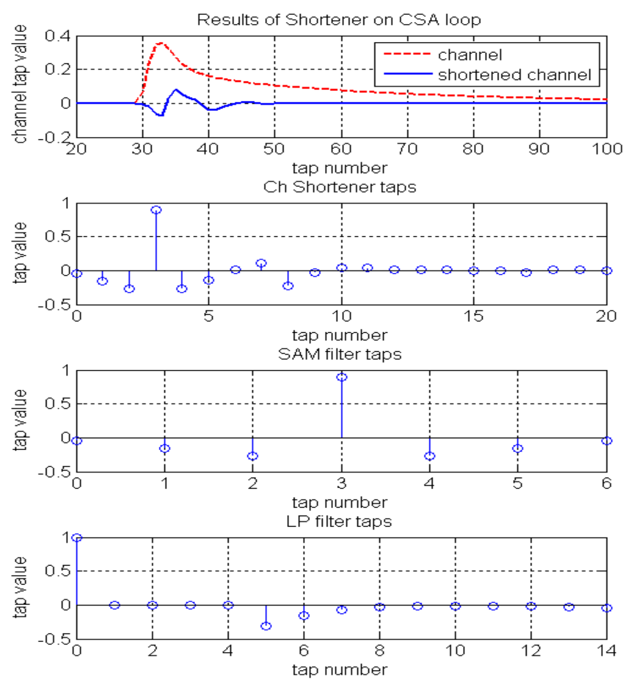

Figure 7 Channel and Equalizer IRs

Converged ABR. The prediction-error filter maintains a steady level of ABR once converged, unlike SAM which has the unstable wandering behaviour of ABR described earlier. As may be seen in Figure 8, some of this behaviour remains present in the aided predictor algorithm. The SAM solution continues to produce temporary zeros close to the unit-circle, causing dips in the hybrid filter frequency response that reduce the channel capacity at that frequency.

The result is that though the average ABR of the aided-predictor is higher than that of SAM, it is lower than that of the unaided predictor.

Mean ABR results of the unaided predictor (10-tap filter), aided-predictor and SAM (16-tap filter) algorithms were obtained for the 8 ADSL test channels, for test runs of 6000-symbol duration. The unaided predictor results are used as the reference, the SAM and aided-predictor results are compared to them. At 40dB SNR, the aided-predictor mean ABR was 3% lower than the unaided predictor mean ABR; SAM was 13% lower. At 20dB, the aided-predictor mean ABR was 6% lower; SAM was 22% lower. The figures are averages: variation among the channels occurred. For one channel (Ch7) SAM had higher mean ABR than the aided-predictor; for Ch2, at 40dB SNR, mean ABR was the same for all three algorithms.

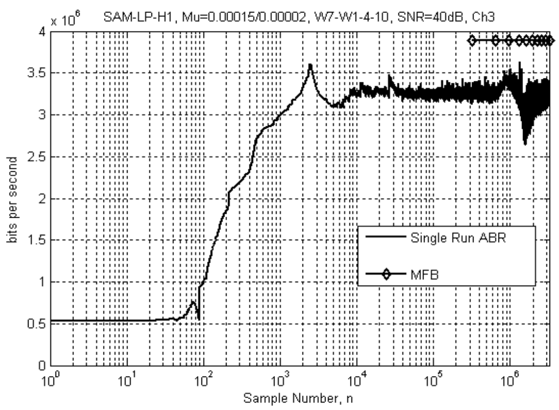

Figure 8 ABR, Ch3, 6000-Symbol Scenario Duration

5. CONCLUSION

Aiding the linear predictor using the SAM algorithm is shown here to provide a significant increase in the convergence speed of equalizing a modelled ADSL channel. The average improvement factor is about 25 and 16 for 40dB and 20dB scenarios respectively. Compared to SAM, the aided predictor requires lower computation and the unstable ABR effect of SAM is reduced.

There remain two drawbacks of introducing the SAM-aiding. The first is the reduction in the steady-state ABR performance of the aided-predictor (3% at 40dB, 6% at 20dB), due to the behaviour of SAM in placing equalizer z-plane zeros close to the unit-circle. This also occurs where aiding is implemented by placing a SAM shortener in series with a predictor. It is suggested here that in a practical channel-shortener the SAM zeros effect may be prevented by stopping SAM adaptation after the initial convergence, potentially by observing the cost function of the predictor, and terminating SAM adaptation when it settles. A more complex solution is to calculate SAM equalizer zeros in a background process, and revert to earlier SAM solutions when z-plane zeros begin to drift toward the unit-circle.

The second drawback is that the computational cost of the SAM-aiding, while less than half of SAM, remains high. Assuming channel length L_h = 100, the cost is about 2000 MAC/sample, compared to 20 MAC/sample for the unaided predictor. It may be possible to ameliorate the cost further by using fewer SAM filter taps, or by decreasing the length of the channel over which ACF is measured. In particular, for primarily minimum-phase channels, the use of a symmetric SAM filter is inefficient.

For an application where equalization convergence time is unimportant, the extra cost of the aided solution over a straightforward prediction-error filter will be unattractive. However, where rapid convergence—say within one MCM-signal symbol—is necessary, the SAM-aided predictor will deliver a solution.

6. REFERENCES

[1] 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.

[2] W.G. Dalzell and C.F.N. Cowan, “Blind Channel Shortening of ADSL Channels with a single-channel Linear Predictor”, *19th European Signal Processing Conference*, Barcelona, Aug-Sept 2011

[3] J. G. Proakis and D. G. Manolakis. *Digital Signal Processing, Principles, Algorithms and Applications.* Pearson Education Inc., Upper Saddle River, N.J., 2007.

[4] M. Kallinger, and A. Mertins, “Room Impulse Response Shortening by Channel Shortening Concepts”, *Thirty-Ninth Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, USA, Oct-Nov 2005, pp. 898-902

[5] C. B. Papadias and D. T. M. Slock. “Fractionally Spaced Equalization of Linear Polyphase Channels and Related Blind Techniques Based on Multichannel Linear Prediction,” IEEE Trans Signal Processing, vol 47, pp 641 – 654, March 1999

[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] W.G. Dalzell and C.F.N. Cowan, “Efficient Blind Channel Shortening using Stochastic Sum-squared Autocorrelation Minimization”, *22nd IET Irish Signals and Systems Conference*, Dublin, June 2011

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

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

[10] G. Arslan, B. L. Evans, and S. Kiaei. “Equalization for Discrete Multitone Receivers To Maximize Bit Rate.” *IEEE Trans. on Signal Processing*, 49(1+2):3123-3135, December 2001