GMDFα WITH ADAPTIVE RECONSTRUCTION FILTERS AND ZERO THROUGHPUT DELAY

J. Lariviere, R. Goubran

Department of Systems and Computer Engineering, Carleton University
1125 Colonel By Drive, Ottawa, Ontario, K1S 5B6, Canada

ABSTRACT

With reduction of the block size (increasing the number of subfilters) regular gmdfα can achieve low throughput delay at the expense of system performance. In situations where zero delay is desirable, we propose a new method which is not dependent on the block size. In addition, by using an adaptive reconstruction filter, further performance gains can be achieved with minimal additional computation complexity. Results from experiments performed in a conference room show an increase in the average Echo Return Loss Enhancement (ERLE) of > 2.5 dB for acoustic echo cancellation over the traditional moving average reconstruction filter.

1. INTRODUCTION

The generalized multi-delay frequency domain adaptive filter (gmdfα) has received considerable attention in echo cancellation in recent years due to several advantages of the algorithm, namely, quick convergence, high levels of cancellation and low delay [1,4,6]. Systems with long impulse responses, such as loudspeaker-room-microphone (LRM) transfer functions, are particularly well suited to gmdfα, since the algorithm subdivides the estimated system impulse response into several smaller impulse responses (resulting in low delay). Also, by performing the filter coefficient adaptation in the frequency domain (and using normalized step sizes) quick convergence is achieved regardless of the condition number (i.e., eigenvalue spread) of the correlation matrix of the input data [2,7]. The algorithm is therefore suitable for speech signals, where large condition numbers are typical.

The perceptual effect of throughput delay exacerbates the acoustic echo cancellation problem. Algorithms which use conventional subband filtering with delay being introduced through the bandpass filters or block transform filtering with large blocks often suffer from this disadvantage [3]. The gmdfα algorithm allows large impulse responses to be achieved with block processing, but where the size of each subfilter block can be reduced, at the expense of an increase in the number of subfilters. The reduction of the block size reduces the throughput delay although there is a limit beyond which there is a substantial decline in system performance due to poor frequency resolution.

This paper is organized as follows. In Section 2 we briefly review the gmdfα algorithm. In Section 3 we discuss an alternative low-delay and delayless gmdfα method. In Section 4 we discuss adaptive reconstruction filters and apply them to regular and delayless gmdfα. To conclude, in Section 5 we summarize our findings and outline directions for future work.

2. BRIEF REVIEW OF GMDFα

2.1 Generation of filter output samples

The "multi-delay" aspect of the gmdfα refers to the subdividing of a filter into multiple, delayed, subfilters. In the time domain, the convolution produces an output, y_n at discrete time n

$$y_n = \sum_{k=0}^{K-1} \boldsymbol{w}_k^T \boldsymbol{x}_{n-kR} \qquad (1)$$

where the original filter of size N is divided into K subfilters of size R, where $N=KR$ and $\boldsymbol{w}_k$ is the weight vector of subfilter k and is related to the original "full-length" weight vector coefficients by $\boldsymbol{w}_k=[w_{kR}\ w_{kR+1}\ \cdots\ w_{(k+1)R-1}]^T$. The input vector is defined as $\boldsymbol{x}_i=[x_i\ x_{i-1}\ \cdots\ x_{i-R+1}]^T$. A frequency domain version of this filter can be realized with the overlap-save or overlap-add algorithm [5], in which each subfilter and its associated block is transformed to the frequency domain, multiplied together, then inverse transformed.

Specifically, the input blocks are transformed to the frequency domain as $\boldsymbol{X}_k=\boldsymbol{F}_M\boldsymbol{x}_k^{(M)}$, $0 \le k \le K-1$, where $\boldsymbol{F}_M$ is the DFT operator of size M by M with coefficients $[\boldsymbol{F}_M]_{n,l}=\exp(-j2\pi nl/M)$, $0 \le n,l \le M-1$ and the input block is extended to a size of M to avoid circular aliasing, i.e.

$$\boldsymbol{x}_k^{(M)}=\left[x_{n-(k+1)R+1}\ \cdots\ x_{n-(k+1)R+M-1}\ \ 0\right]^T,\quad 0 \le k \le K-1 \qquad (2)$$

where M is arbitrarily taken as 2R. Similarly, the filter coefficients are transformed as $\boldsymbol{W}_k=\boldsymbol{F}_M\boldsymbol{w}_k^{(M)}$, where $\boldsymbol{w}_k^{(M)}=\left[w_{kR}\ \cdots\ w_{(k+1)R-1}\ \ \boldsymbol{0}_{1,M-R}\right]^T$ and $\boldsymbol{0}_{i,j}$ represents a zero matrix of size i by j.

Representing the inverse Fourier transform operator as $\boldsymbol{F}_M^{-1}$ and the element by element multiplication of matrices or vectors by $\otimes$, the transformed input and weight vectors produce the circularly convolved output vectors

$$\tilde{\boldsymbol{y}}_k=\boldsymbol{F}_M^{-1}\left(\boldsymbol{X}_k \otimes \boldsymbol{W}_k\right),\qquad 0 \le k \le K-1 \qquad (3)$$

which, with M=2R, must be stripped of the first R-1 elements and last element to achieve linear convolution. Since the DFT is a linear operator, the summation of the K vectors can be performed before the inverse transform (resulting in a savings of

K-1 DFTs). The resulting equation describing the output at each iteration of the filter is then

$$\hat{\boldsymbol{y}}=\begin{bmatrix}\mathbf{0}_{R-1,R-1} & \mathbf{0}_{R-1,R} & \mathbf{0}_{2R-1,1}\\ \mathbf{0}_{R,R-1} & \boldsymbol{I}_R & \\ \mathbf{0}_{1,2R-1} & & 0\end{bmatrix}\boldsymbol{F}_M^{-1}\left(\sum_{k=0}^{K-1}\boldsymbol{X}_k\otimes\boldsymbol{W}_k\right) \tag{4}$$

As suggested by equation 4, $\hat{\boldsymbol{y}}$ is effectively truncated to R elements. Each iteration of the filter will therefore shift the input and output blocks by P=R samples. Note that this results in computational savings, since the DFT of the input blocks can be re-used in the next iteration of the filter, i.e. $\boldsymbol{X}_k$ at iteration s equals $\boldsymbol{X}_{k+1}$ at iteration $s+1$, $0 \le k \le K-2$. The *generalized* multi-delay filter removes the restriction of shifting R samples at each iteration. The amount of shifting, P, is controlled by the parameter α, where

$$P=\frac{R}{\alpha}, \qquad P,\alpha \in \mathbb{I} \tag{5}$$

A number of points should be mentioned; firstly, $\alpha = 1$ corresponds to regular multi-delay filtering and results in the least computations per output sample. $\alpha > 1$ implies that more than one estimate of each output sample will be produced and hence there is a need for a *reconstruction filter* to combine the estimates. Also, at each iteration, the immediately previous iteration's DFT transformed input blocks cannot be used, rather, transformed input blocks from α iterations previous must be used. This means that the computational savings from the reuse of transformed input blocks can still be achieved, but the memory requirements increase α-fold. The advantage of $\alpha > 1$ is in terms of increased convergence speed and higher overall levels of convergence. The improved performance is generally attributed to the ability to update the filter weights more often than every R^{th} output sample.

Let $\boldsymbol{g}=[g_0\ g_1\ \cdots\ g_{\alpha-1}]^T$ be the vector of reconstruction filter weights, with the constraint $\sum_{i=0}^{\alpha-1} g_i=1$, then the final output samples are generated by

$$y_n=\sum_{i=0}^{\alpha-1} g_i\,\hat{y}_{(s-i);(j+iP)} \tag{6}$$

where $\hat{y}_{m;n}$ represents the n^{th} *element* of the m^{th} iteration output vector, $\hat{\boldsymbol{y}}_m$ and $j=remainder(sP/n)+1$ where s is the current iteration of the filter and P is defined as in (5).

2.2 Updating of filter coefficients

The weight update equations use a gradient which is a scaled cross-correlation between the input vector and the error vector. Cross-correlation, like convolution, can be performed in the frequency domain as circular correlation. With appropriate zero-padding, linear correlation is achieved by transforming the circular correlation samples to the time-domain, zeroing out the appropriate samples, then transforming the remaining samples back to the frequency domain. In practice, this step is often omitted, saving 2K DFTs in exchange for a minor degradation in performance. The resulting algorithm is referred to as the *unconstrained* gmdfα. Writing the constraint operator as

$$\boldsymbol{C}=\boldsymbol{F}_M\begin{bmatrix}\boldsymbol{I}_R & \mathbf{0}_{R,R}\\ \mathbf{0}_{R,R} & \mathbf{0}_{R,R}\end{bmatrix}\boldsymbol{F}_M^{-1} \tag{7}$$

the weight update equations are $\boldsymbol{W}_k^{s+1}=\boldsymbol{W}_k^s+\mu C\left(\boldsymbol{X}_k^*\otimes\boldsymbol{E}^s\right)$ and $\boldsymbol{W}_k^{s+1}=\boldsymbol{W}_k^s+\mu\left(\boldsymbol{X}_k^*\otimes\boldsymbol{E}^s\right)$, $0 \le k \le K-1$ for the constrained and unconstrained versions respectively, where

$$\boldsymbol{E}^s=\boldsymbol{F}_M\begin{bmatrix}\mathbf{0}_{M-R,1}\\ \boldsymbol{e}_s\end{bmatrix} \tag{8}$$

is the frequency domain version of the error vector at iteration s and * represents complex conjugation.

One of the advantages of frequency domain filtering is the ability to normalize the step size based on the power levels in each frequency bin. If we let $\boldsymbol{T}^s=\left[P_{s;1}^{-1}\ \cdots\ P_{s;M}^{-1}\right]^T$ where $P_{s;i}$ is an estimate of the power level in the i^{th} bin at iteration s, we can then write the normalized constrained and unconstrained weight update equations as $\boldsymbol{W}_k^{s+1}=\boldsymbol{W}_k^s+\mu C\left(\boldsymbol{T}^s\otimes\boldsymbol{X}_k^*\otimes\boldsymbol{E}^s\right)$ and $\boldsymbol{W}_k^{s+1}=\boldsymbol{W}_k^s+\mu\left(\boldsymbol{T}^s\otimes\boldsymbol{X}_k^*\otimes\boldsymbol{E}^s\right)$ respectively, $0 \le k \le K-1$.

3. LOW-DELAY AND DELAYLESS GMDFα

The throughput delay for regular gmdfα is directly related to the block size as $D=(R-1)/f_s$ where f_s is the sampling frequency. To illustrate the effect on system performance of reducing the block size to achieve low throughput delay, Figure 1 shows the average ERLE for a system operating on 2.3 seconds of synthetic speech as recorded in a real conference room. The filter has the equivalent of N=640 taps and block sizes are R=32, 16, 8, 4, 2 and 1, with corresponding subfilters numbering K=20, 40, 80, 160, 320 and 640 (such that N=KR). The average ERLE for NLMS is also plotted, as a reference performance level.

There are two aspects that contribute to the delay in the gmdfα algorithm, filtering and coefficient updating. Delays in filtering are due to the use of "future" samples of the input sequence (reference signal) during the blocking operation. Delays from coefficient updating are due to the use of future samples of the desired sequence (primary signal) as well as the input sequence.

Examination of equation (2) shows that $\boldsymbol{x}_k^{(M)}$ involves input sequence samples greater than n only when $k=0$, i.e. only the first subfilter makes use of future samples. This suggests the *filtering* operation can be made delayless by operating the first block in the time-domain (all other blocks continue to operate in the frequency domain). The coefficient update aspect is more problematic, since every subfilter's adaptation is based on the same error vector, defined as

$$\boldsymbol{e}_s=\left[\hat{y}_{s;0}\ \cdots\ \hat{y}_{s,R-1}\right]^T-\left[d_{sP}\ \cdots\ d_{sP-R-1}\right]^T \tag{9}$$

Figure 2 is a schematic representation of the use of the error vector in relation to the generation of the current output sample

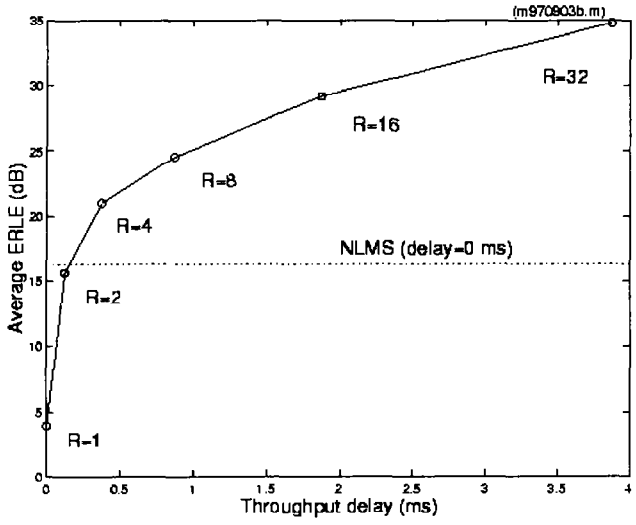

Figure 1 Average ERLE vs. throughput delay for NLMS and gmdfα with different block sizes (indicated by R).

with R=4 and α=R. Examining the generation of y_n we see the filter coefficients used to produce $\hat{y}_s$ are updated using the error vector e_{s-1} . As indicated by the vertical line representing the present time, some of the elements of e_{s-1} involve future samples. Only by basing the update on an error vector α iterations in the past, do we obtain coefficient updating with no dependence on future samples. Hence, delayless throughput requires delayed filter coefficient updating. Unfortunately, this has a direct effect on the rate

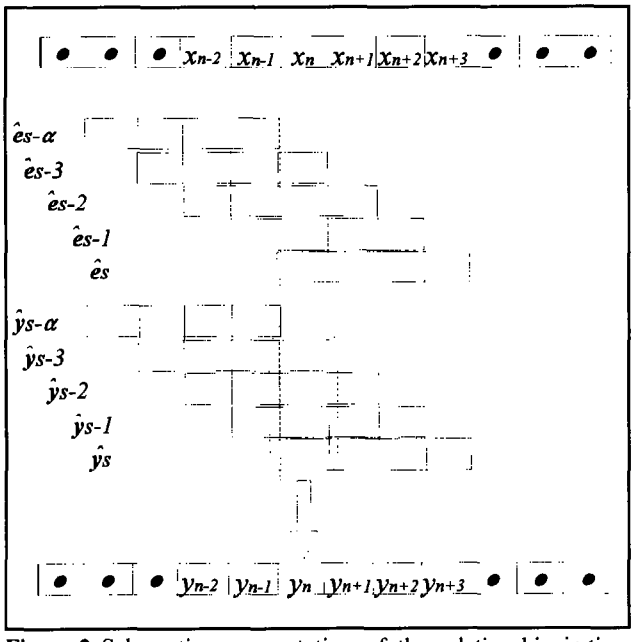

Figure 2 Schematic representation of the relationship in time between the error vector, e_s and the output sample y_n, with R=4 and α=R.

of convergence, since the step size must be decreased for stability. (The effect of "incorrect" gradient estimates cannot be immediately recognized in the next iteration.)

Figure 2 suggest an alternative method of obtaining low-delay gmdfα processing, without decreasing the block size, that is, perform the first subfilter block in the time domain and choose an appropriate coefficient update delay based on a tolerable decline in the rate of convergence. That is, the more delay that can be tolerated in the coefficient updates, the less throughput delay will result. By delaying the coefficient updates by a full α iterations, completely delayless throughput will be obtained.

Figure 3 compares the performance of the new method of obtaining delayless gmdfα and NLMS for synthetic speech recorded in a real conference room. The new method averages approximately 3 dB/sample better ERLE than the NLMS method for the first second. The major dips in the ERLE curves are due solely to speech power dropping between words and do not indicate performance degradation.

4. ADAPTIVE RECONSTRUCTION FILTERS

Further performance gains can be achieved by examining the role of the reconstruction filter, which traditionally, has been restricted to a moving average (MA) filter.

At each iteration of the gmdfα's main filter, an estimate of the output at a block of discrete time locations is made. An examination of the error variance of these estimates shows that the error variance is higher at the beginning and end of the block of estimates. This is especially evident with the *unconstrained* gmdfα algorithm, where circular correlation is used instead of linear correlation. Figure 4 plots the error variance as a function of the intra block index for synthetic speech recorded in a

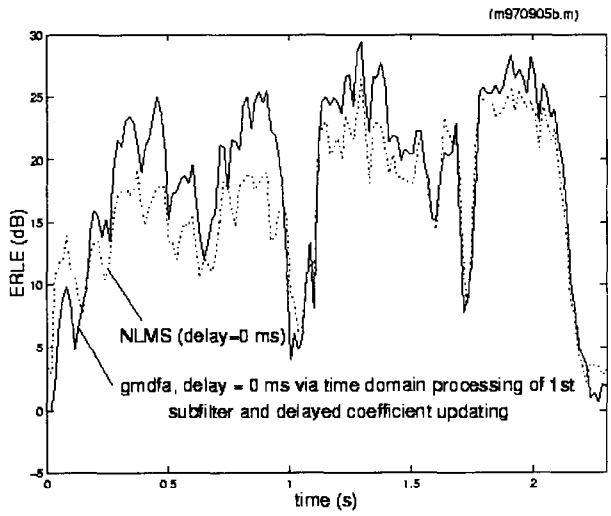

Figure 3 ERLE for delayless gmdfα and NLMS.

conference room, using differing amounts of overlap (α=1,2,4,8, R=16).

The output of the reconstruction filter at iteration s can be expressed in matrix notation as $y_n = \boldsymbol{g}^{(s)T}\boldsymbol{v}_j^{(s)}$ with the reconstruction vector, $\boldsymbol{v}_j^{(s)}$ defined as a vector of past estimates corresponding to the same time location,

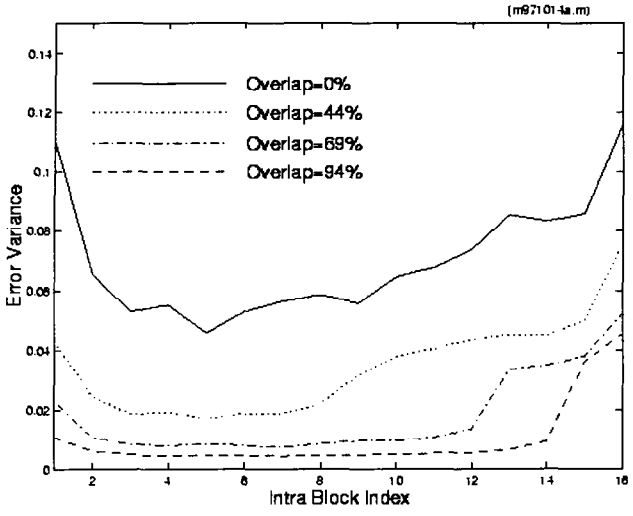

Figure 4 Error variance vs intra block index for unconstrained gmdfα.

$$\boldsymbol{v}_j^s = \left[\hat{y}_{s;j} \;\; \hat{y}_{(s-1);(j+P)} \;\; \cdots \;\; \hat{y}_{(s-\alpha+1),(j+(\alpha-1)P)}\right]^T \qquad (10)$$

A moving average filter, where each element of $\boldsymbol{v}_j^{(s)}$ is weighted by the same amount may not, therefore, produce the best final output sample, since the elements do not have the same error variance. Intuitively, it would be better to weight more heavily those estimates with the least error variance. This can be achieved through the familiar NLMS update equation

$$\boldsymbol{g}^{(s+1)} = \boldsymbol{g}^{(s)} + \frac{\mu_R \boldsymbol{v}_j^{(s)} e_n}{\boldsymbol{v}_j^{(s)T}\boldsymbol{v}_j^{(s)}} \qquad (11)$$

Figure 5 compares the performance for the two different types of reconstruction filters, using three versions of α, based on the block size. In all cases, the step size was set to 0.03125. A low step size value is needed, since the input to the adaptive reconstruction filter is almost DC (resulting in a steep error surface). As can be seen, the adaptive reconstruction method performs better than the moving average methods in almost all cases. These trials were conducted with synthetic speech recorded in a real conference room. These results were obtained using the regular (delayed) version of gmdfα. Trials were also performed combining the adaptive reconstruction filter with the delayless gmdfα. Preliminary results indicate only minor performance gains in this case.

5. SUMMARY

Two enhancements of the gmdfα algorithm were examined, delayless gmdfα and gmdfα with adaptive reconstruction filters. Zero throughput delay is possible without limiting the block size although convergence deteriorates because of the necessary step-size reduction. Using an adaptive reconstruction filter resulted in > 2.5 dB improvement in observed average ERLE . Future work will examine the ability of the adaptive reconstruction filter to handle noise present in the near end speech signal.

6. REFERENCES

[1] A. Baganne, A. Gilloire, E. Martin, P. Martineau, J.P. Thomas, "Implementation Methods of an Acoustic Echo Cancellation Algorithm GMDF," IEEE International Workshop on Acoustic Echo and Noise Control, pp. 83-86, Jun. 1995

[2] S.S. Haykin, "Adaptive Filter Theory," Prentice-Hall, Englewood Cliffs, N.J. 1991

[3] D.R. Morgan, J.C. Thi, "A Delayless Subband Adaptive Filter Architecture," IEEE Trans on Signal Processing, Vol. 43, No. 8, pp. 1819-1830, Aug. 1995

[4] E. Moulines, O.A. Amrane, Y. Grenier, "The Generalized Multidelay Adaptive Filter: Structure and Convergence Analysis," IEEE Trans on Signal Processing, Vol 43, No. 1, pp 14-28, Jan. 1995

[5] A.V. Oppenheim, R.W. Schafer, "Discrete-time signal processing," Prentice-Hall Inc. N.J., 1989

[6] J. Prado, E. Moulines, "Frequency-domain adaptive filtering with applications to acoustic echo cancellation," Ann.. Telecom., Vol 49, No 7-8, pp 414-428, 1994

[7] J.J. Shynk, "Frequency-Domain and Multirate Adaptive Filtering," IEEE Signal Processing Magazine, pp. 14-37, Jan. 1992

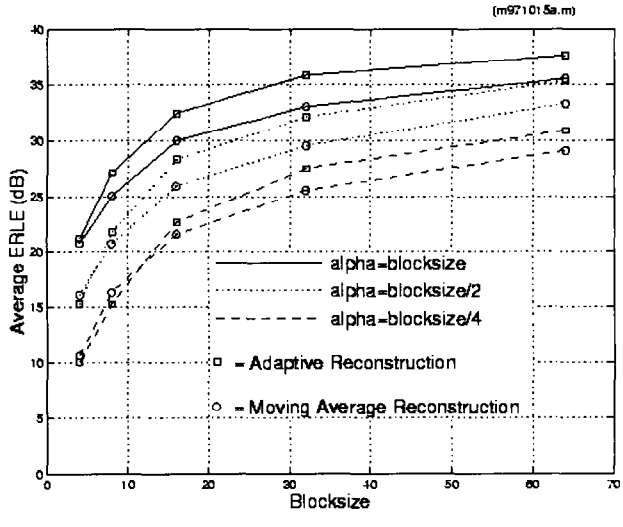

Figure 5 Average ERLE vs. Block size for adaptive reconstruction filters and moving average reconstruction filters for α=block size, block size/2 and block size/4.