THE ITERATED PARTITIONED BLOCK FREQUENCY–DOMAIN ADAPTIVE FILTER FOR ACOUSTIC ECHO CANCELLATION

Koen Eneman *Marc Moonen*

ESAT – Katholieke Universiteit Leuven
Kasteelpark Arenberg 10, B–3001 Leuven – Belgium

email : koen.eneman@esat.kuleuven.ac.be marc.moonen@esat.kuleuven.ac.be

ABSTRACT

For high quality acoustic echo cancellation long echoes have to be suppressed. Classical LMS–based adaptive filters are not attractive here as they are suboptimal from a computational point of view. Multirate adaptive schemes such as the partitioned block frequency–domain adaptive filter (PBFDAF) are a good alternative and are widely used in commercial echo cancellers nowadays.
In this paper the PBFDRAP adaptive filter is analyzed, which combines frequency–domain adaptive filtering with so–called "row action projection". Fast versions of the algorithm can be derived and it will be shown that the PBFDRAP outperforms the PBFDAF in a realistic echo cancellation setup.

1. INTRODUCTION

For high quality acoustic echo cancellation long echoes have to be suppressed. Acoustic echo paths are characterized by FIR filters with lengths up to 250 ms. Filters clocked at a rate of, say, 10 kHz then require several thousands of filter taps to be identified.
Of all existing adaptive algorithms the Least Mean Square algorithm may be best known. LMS–based algorithms have a complexity that is linear in the filter length, but they suffer from a rather slow convergence for signals with a colored spectrum such as speech.
More efficient implementations of the LMS algorithm are available, often relying on frequency–domain techniques [8]. The so–called Partitioned Block Frequency–Domain Adaptive Filter (PBFDAF) [7] is used in commercial echo cancellers nowadays.
In this paper the PBFDRAP will be discussed, which is a combination of the PBFDAF algorithm and so–called "row action projection" (RAP). A good reference on Row Action Projection is [5]. Fast versions of the PBFDRAP algorithm can be derived and it will be shown that the PBFDRAP outperforms the PBFDAF in a realistic echo cancellation setup.

This research work was carried out at the ESAT laboratory of the Katholieke Universiteit Leuven, in the frame of the Belgian State Interuniversity Poles of Attraction Programme – IUAP P4–02 (1997–2001) : Modeling, Identification, Simulation and Control of Complex Systems and Concerted Research Action GOA–MEFISTO–666 of the Flemish Government and IWT project MUSETTE ('Multimicrophone Signal Enhancement Techniques for handsfree telephony and voice controlled systems'). The scientific responsibility is assumed by its authors.

2. PARTITIONED BLOCK FREQUENCY–DOMAIN RAP

As a cheaper alternative to the LMS algorithm, the frequency–domain adaptive filter (FDAF) was introduced [8]. The FDAF is a block based adaptive filter, which is a direct translation of Block–LMS to the frequency domain. It appears that the FDAF algorithm is computationally attractive only if the block length has the same order of magnitude as the adaptive filter length N. In practice however, this leads to unacceptable input/output delays.
By splitting the N–taps fullband adaptive filter $\hat{w}^{(n)}[k]$ in equal parts and transforming to the frequency domain, a mixed time/frequency–domain adaptive filter can be obtained, called the Partitioned Block Frequency–Domain Adaptive Filter (PBFDAF) [7]. The parameters of the PBFDAF can be chosen such that a cheap adaptive filter with acceptable processing delay is obtained. The PBFDAF is widely used in commercial echo cancellers nowadays.
Stepping several times through the filter and weight updating part of the PBFDAF algorithm leads to an improved error and weight update. This extension of the PBFDAF algorithm which iterates several times on the same block of data will be called the partitioned block frequency–domain RAP adaptive filter (PBFDRAP). RAP stands for Row Action Projection and was initially proposed as an improvement to the LMS algorithm. A good reference on Row Action Projection is [5].
The update equations defining the overlap–save PBFDRAP are :

$$\mathbf{X}_p^{(n)} \stackrel{\forall p}{=} \mathrm{diag}\left\{ \mathbf{F} \begin{bmatrix} x[(n+1)L - pP - M + 1] \\ \vdots \\ x[(n+1)L - pP] \end{bmatrix} \right\}, \qquad (1)$$

$$\mathbf{d}^{(n)} = \begin{bmatrix} \mathbf{0} \\ \mathbf{d}_n \end{bmatrix} \begin{matrix} \updownarrow P-1+\sigma \\ \updownarrow L \end{matrix}, \quad \mathbf{d}_n = \begin{bmatrix} d[nL+1] \\ \vdots \\ d[(n+1)L] \end{bmatrix} \qquad (2)$$

for $r = 1$ to R do {

$$\mathbf{y}^{(n,r)} = \begin{bmatrix} \mathbf{0}_{P-1+\sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{N}{P}-1} \mathbf{X}_p^{(n)} \mathbf{w}_p^{(n,r)} \qquad (3)$$

$$\mathbf{e}^{(n,r)} = \mathbf{d}^{(n)} - \mathbf{y}^{(n,r)} \qquad (4)$$

$$\mathbf{w}_p^{(n,r+1)} \stackrel{\forall p}{=} \mathbf{w}_p^{(n,r)} + \mathbf{G}\boldsymbol{\Delta}\mathbf{X}_p^{(n)*}\mathbf{F}\mathbf{e}^{(n,r)}, \qquad (5)$$

}

in which $0 \leqslant p \in \mathbb{Z} \leqslant \frac{N}{P} - 1$ (assuming that $\frac{N}{P} \in \mathbb{Z}$), $\mathbf{F}$ is the $M \times M$ DFT matrix and n is the block index. Each

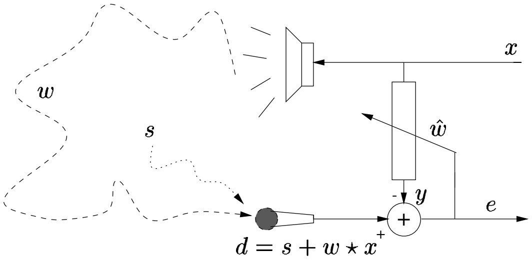

Figure 1: Echo cancellation setup

time Eq. 1–5 are performed L new far–end samples x are taken in, and L new filter output samples e are produced (see figure 1). Vector $\mathbf{d}_n$ contains the L most recent values of near–end signal d. Vectors $\mathbf{w}_p^{(n,r)}$, $\mathbf{y}^{(n,r)}$ and $\mathbf{e}^{(n,r)}$ are respectively the adaptive filter weights, the reconstructed echo and the error output at iteration step r. The PBFDRAP reduces to the PBFDAF if $R = 1$. Further,

$$\mathbf{w}_p^{(n+1,1)} \overset{\forall p}{=} \mathbf{w}_p^{(n,R+1)}, \quad 0 \leqslant p \in \mathbb{Z} \leqslant \frac{N}{P} - 1 \tag{6}$$

and

$$\hat{\boldsymbol{w}}_p^{(n)} \overset{\forall p}{=} \begin{bmatrix} \hat{w}^{(n)}[pP] \\ \vdots \\ \hat{w}^{(n)}[(p+1)P-1] \end{bmatrix}, \tag{7}$$

$$\mathbf{w}_p^{(n,1)} \overset{\forall p}{=} \mathbf{F} \begin{bmatrix} \hat{\boldsymbol{w}}_p^{(n)} \\ \mathbf{0} \end{bmatrix} \begin{matrix} \updownarrow P \\ \updownarrow M-P \end{matrix} \tag{8}$$

$\hat{w}^{(n)}[k]$ is the equivalent N–taps fullband adaptive filter for block n. L is called the block length, and hence the corresponding input/output delay of the PBFDRAP is $2L - 1$. Further, $\mathbf{\Delta} = \text{diag}\{\boldsymbol{\mu}_s^{(n)}\}$ contains subband dependent stepsizes and $M = P + L - 1 + \sigma$, with $\sigma \in \mathbb{Z} \geqslant 0$.
If P is divisible by L (which is typically the case in practice), $\mathbf{X}_p^{(n)} = \mathbf{X}_0^{(n-pP/L)}$ and hence equation 1 requires only 1 DFT operation, namely the one that corresponds to $p = 0$. The other $\mathbf{X}_p^{(n)}$ can be recovered from previous iterations. In most practical applications $P = L$, $\sigma = 1$ and M is a power of 2.
There exist two variants of this algorithm, called the constrained and the unconstrained PBFDRAP. Matrix $\mathbf{G}$ in equation 5 defines the type of updating : for the unconstrained PBFDRAP, $\mathbf{G} = \mathbf{I}_M$, for the constrained PBFDRAP

$$\mathbf{G} = \mathbf{F} \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{M-P} \end{bmatrix} \mathbf{F}^{-1}. \tag{9}$$

The unconstrained updating requires 3 DFTs per iteration, provided P is divisible by L, whereas the constrained PBFDRAP is more expensive, having an extra $\frac{2N}{P}$ DFTs to compute. The latter on the other hand has better convergence properties.
By introducing stepsize normalization the convergence can be improved. As the PBFDRAP takes on the form of an oversampled subband adaptive filter [3], applying different stepsizes to each subband, depending on the subband energy, enhances the convergence.
An ambiguity can occur with the unconstrained PBFDRAP algorithm if $\sigma > 0$: it is not guaranteed that the filter weights maintain the initial format set in Eq. 8. A random value ϵ can appear in Eq. 8 as an extra $(P+1)$–th element of $\hat{\boldsymbol{w}}_p^{(n)}$ for instance, as long as it is compensated for at the first element of $\hat{\boldsymbol{w}}_{p+1}^{(n)}$ [3]. This ambiguity can be easily compensated for by slightly changing Eq. 2 and 3, i.e. computing $\mathbf{e}^{(n,r)}$ based on $L + \sigma$ instead of L signal samples :

$$\mathbf{d}^{(n)} = \begin{bmatrix} \mathbf{0} \\ \mathbf{d}_n \end{bmatrix} \begin{matrix} \updownarrow P-1 \\ \updownarrow L+\sigma \end{matrix}, \quad \mathbf{d}_n = \begin{bmatrix} d[nL - \sigma + 1] \\ \vdots \\ d[(n+1)L] \end{bmatrix} \tag{10}$$

$$\mathbf{y}^{(n,r)} = \begin{bmatrix} \mathbf{0}_{P-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{L+\sigma} \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{N}{P}-1} \mathbf{X}_p^{(n)} \mathbf{w}_p^{(n,r)}. \tag{11}$$

The other equations remain unchanged and the additional algorithmic cost is almost negligible.
The PBFDRAP algorithm iterates R times on the same block of data $(\mathbf{X}_p^{(n)}, \mathbf{d}_n)$. Hence, an improved weight update $\mathbf{w}_p^{(n,R+1)}$ and a smaller a–posteriori error output $\mathbf{e}^{(n,R)}$ are obtained. If the number of iteration steps R is increased the algorithm further reduces the a–posteriori error by exploiting the signal characteristics, i.e. adapting the filter coefficients towards a solution that minimizes $\mathbf{e}^{(n,R)}$ for data block $(\mathbf{X}_p^{(n)}, \mathbf{d}_n)$, rather than trying to improve the model of the unknown system w as such. Therefore, a small a–posteriori error $\mathbf{e}^{(n,R)}$ does not necessarily imply a good echo path modelling. Hence, the quality of the model approximation should be evaluated by plotting the norm between the unknown system w and the model $\hat{w}^{(n)}$ or, if w is not known, by looking at the time evolution of the so–called a–priori error $\mathbf{e}^{(n,1)}$.

3. ON ITERATING THE PBFDRAP

It was shown [6] that the limiting result of reiterating the LMS algorithm leads to the normalized LMS algorithm. The same sort of derivation can be done for the PBFDRAP algorithm to check what happens if the number of iteration steps R goes to infinity.

3.1. Unnormalized and globally normalized unconstrained PBFDRAP

For the unnormalized unconstrained PBFDRAP, $\mathbf{G} = \mathbf{I}_M$ and $\mathbf{\Delta}$ is defined as $\mu \mathbf{I}_M$, with μ a positive constant. More generally, $\mathbf{\Delta} = \mu_n \mathbf{I}_M$, with μ_n a block dependent stepsize. For instance, if[1]

$$\mu_n = \frac{\mu}{\frac{1}{M}\sum_{m=0}^{M-1}\sum_{p=0}^{\frac{N}{P}-1} \mathbf{X}_p^{(n)}(m)\mathbf{X}_p^{(n)^*}(m)} \tag{12}$$

the algorithm performs a global normalization, leading to better convergence if signals with a high dynamic range (such as speech) are involved.
In [2] it is proven that if the number of iteration steps R goes to infinity, both the unnormalized and globally normalized unconstrained PBFDRAP approach a normalized version of the unconstrained PBFDAF that is based on projected subband energies and adapted with maximum stepsize $\mu_n = 1$, i.e.

$$\mathbf{w}_p^{(n,\infty)} \overset{\forall p}{=} \mathbf{w}_p^{(n,1)} + \mathbf{X}_p^{(n)^*} \left(\boldsymbol{\mathcal{P}} \sum_{p=0}^{\frac{N}{P}-1} \mathbf{X}_p^{(n)} \mathbf{X}_p^{(n)^*} \right)^{-1} \mathbf{F}\mathbf{e}^{(n,1)}, \tag{13}$$

[1] $\mathbf{X}_p^{(n)}(m)$ is the m–th diagonal element of matrix $\mathbf{X}_p^{(n)}$.

in which $\mathcal{P}$ is a projection matrix, depending on the far–end signal x [2].

3.2. Subband–normalized unconstrained PBFDRAP

The PBFDRAP may be considered as a subband adaptive system [3]. The convergence can be optimized by applying a different stepsize to each of the subband adaptive filters. Therefore, matrix $\mathbf{\Delta}$ is typically chosen as follows :

$$\mathbf{\Delta} = \mu \left(\sum_{p=0}^{\frac{N}{P}-1} \mathbf{X}_p^{(n)} \mathbf{X}_p^{(n)*} \right)^{-1} . \tag{14}$$

In this way in each subband the adaptation stepsize μ is divided by the subband energy. It is proven (see [2]) that the subband–normalized unconstrained PBFDRAP converges to the subband–normalized unconstrained PBFDAF with $\mu = 1$: increasing the number of iteration steps R has the same effect as applying a larger stepsize μ.

3.3. Unnormalized and globally normalized constrained PBFDRAP

In [1] the relation between two adaptive filtering techniques was examined : iterated Block–LMS and the Partial Rank Algorithm (PRA). The PRA is a block version of the Affine Projection Algorithm (APA) [5] and can be interpreted as a block–normalized version of the Block–LMS algorithm [1]. It was shown that iterated Block–LMS approaches PRA with maximum stepsize $\mu = 1$, if the number of iteration steps goes to infinity [1]. As the unnormalized constrained PBFDRAP is an exact representation of iterated Block–LMS in the frequency domain, it will also approach the PRA with maximum stepsize, if R goes to infinity. In [2] it was verified more formally that the unnormalized and also the globally normalized constrained PBFDRAP algorithm approach the PRA.

3.4. Subband–normalized constrained PBFDRAP

The same sort of derivation can be made for the subband–normalized constrained PBFDRAP to check what happens if R goes to infinity. Stability problems arise however in this case as the algorithm may diverge for large values of R. Starting from the weight update equation (Eq. 5) the limit for $R \to \infty$ can be computed, but for some data sets it appears that the algorithm will convergence to an unstable filter [2]. However, this does not automatically imply divergence of the algorithm for finite R, but indicates that iterating this algorithm is expected to invoke stability problems. It seems that also in this case increasing the number of iteration steps R has the same effect as applying a larger stepsize μ (cf section 3.2).

4. FAST PBFDRAP

At first sight iterating R times on the same block of data is rather expensive, the implementation cost being almost R times higher. Fast implementations were derived for the PBFDRAP algorithm [4] making a high number of iteration steps more attractive. Some of the fast implementations are specifically tuned towards the unconstrained PBFDRAP or are suitable for constrained updating only. Other versions are applicable to both update schemes.
For example, the results of a cost comparison between the standard unnormalized unconstrained PBFDRAP (Eq. 1–5) and a fast version of the unconstrained PBFDRAP are shown in figure 2. Both the standard and the fast imple-

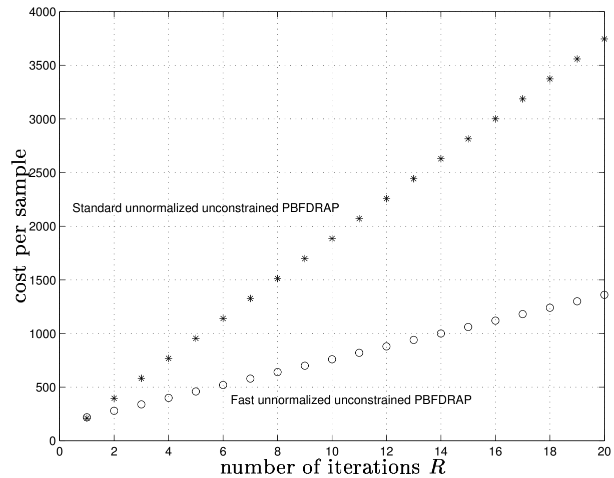

Figure 2: Cost standard unnormalized unconstrained PBFDRAP (Eq. 1–5) vs. fast unnormalized unconstrained PBFDRAP for a realistic parameter setting ($M = 256$, $P = L = 128$, $N = 1024$)

mentation give the same output, the fast algorithm being significantly cheaper. Iterating twice with the standard approach for instance is as expensive as iterating 4 times with the fast algorithm.

5. SIMULATION EXAMPLES

A lowpass noise signal having the spectral characteristics of speech was fed into a loudspeaker and recorded with a microphone in a controlled stationary laboratory environment. No near–end signal s was added on top (see figure 1).
In a first experiment the unknown acoustic path was estimated using the globally normalized PBFDRAP (cf. Eq. 12), varying the type of updating (unconstrained, constrained) and the number of iteration steps R. The other parameters were kept constant and were chosen as follows : $L = P = 128$, $M = 256$, $N = 1024$. Eq. 10 and 11 were employed to compensate for the ambiguity which can occur (see section 2). For each of the algorithms here presented the optimal stepsize was computed which maximizes the convergence speed. This stepsize was divided by 10 and applied to the adaptive filter, in order to simulate the performance of the algorithms in the presence of double–talk. During double–talk the local signal source s is active (see figure 1). If double–talk is detected the adaptation constant μ must be set immediately to zero in order to freeze the coefficients of the adaptive filter. If adaptation is not switched off the filter coefficients easily drift away from their Wiener solution, leading to a bad model and a larger error output e. If the adaptation constant μ is kept significantly smaller than the theoretical optimum large deviations of the filter coefficients can be avoided. The time evolution of the a–priori error output e for each of the adaptive filters is shown in figure 3.
In a second experiment the unknown acoustic path was estimated using the subband–normalized PBFDRAP (cf.

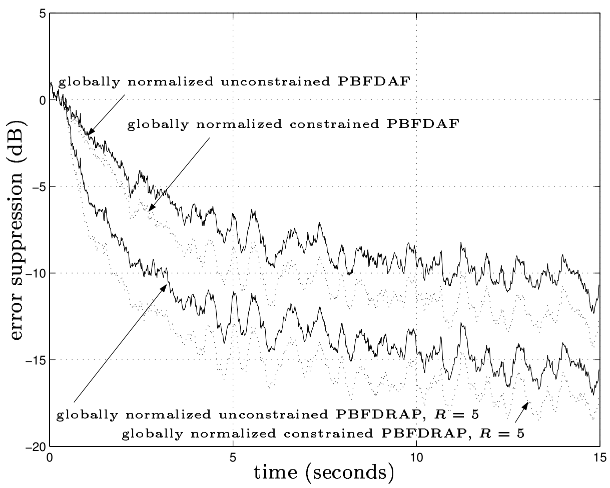

Figure 3: The globally normalized PBFDRAP was used to estimate an unknown stationary acoustic path, varying the type of updating (unconstrained, constrained) and the number of iteration steps R. The following parameters were applied : $L = P = 128$, $M = 256$, $N = 1024$. The time evolution of the a–priori error output e is shown for each of the adaptive filters.

Eq. 14), varying the type of updating (unconstrained, constrained) and the number of iterations R. Also in this experiment $L = P = 128$, $M = 256$, $N = 1024$. The ambiguity was compensated for and the adaptation stepsizes were kept an order of magnitude below the optimum. The time evolution of the a–priori error output e for each of the adaptive filters is shown in figure 4.

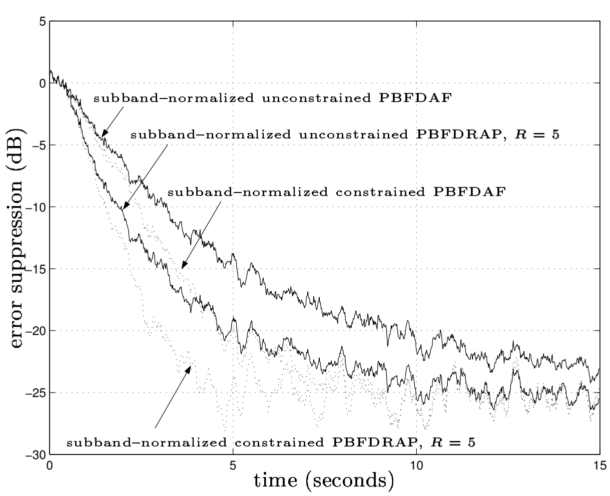

Figure 4: The subband–normalized PBFDRAP was used to estimate an unknown stationary acoustic path, varying the type of updating (unconstrained, constrained) and the number of iteration steps R. The following parameters were applied : $L = P = 128$, $M = 256$, $N = 1024$. The time evolution of the a–priori error output e is shown for each of the adaptive filters.

It is clear that by increasing the number of iteration steps R the performance of the PBFDAF can be improved. As the input signal is colored, the subband–normalized PBFDRAP convergences faster than the globally normalized PBFDRAP. Although stability is not guaranteed for $R \to \infty$, the subband–normalized constrained PBFDRAP offers the best performance for real signals and realistic values of R. Finally, if the a–posteriori errors are plotted instead of the a–priori errors extra echo enhancement can be obtained. However, during double–talk a–priori errors should be passed to the output to avoid near–end signal cancellation.

6. CONCLUSIONS

In this paper the PBFDRAP was discussed, an adaptive filtering algorithm which combines row action projection and partitioned block frequency–domain adaptive filtering. For some parameter settings the PBFDRAP algorithm approaches well–known adaptive filtering algorithms such as the PRA. Fast versions of the algorithm can be derived, leading to a reduced algorithmic complexity. Based on simulation results it was shown that in a realistic signal enhancement setup such as acoustic echo cancellation, the PBFDRAP can be employed to obtain improved system estimates outperforming the standard partitioned block frequency–domain adaptive filter.

7. REFERENCES

[1] K. Eneman, S. Doclo and M. Moonen. *A Comparison between Iterative block–LMS and Partial Rank Algorithm.* Technical Report 99-52 (ftp.esat.kuleuven.ac.be/pub/sista/eneman/reports/00-52.ps.gz ESAT–SISTA, Katholieke Universiteit Leuven, Belgium, March 1999.

[2] K. Eneman and M. Moonen. *On Iterating the Partitioned Block Frequency–Domain Adaptive Filter.* Technical Report 00-127 (ftp.esat.kuleuven.ac.be/pub/sista/eneman/reports/00-127.ps.gz) ESAT–SISTA, Katholieke Universiteit Leuven, Belgium, December 2000.

[3] K. Eneman and M. Moonen. *Hybrid Subband/Frequency–Domain Adaptive Systems.* Signal Processing, 81(1):117–136, January 2001.

[4] K. Eneman and M. Moonen. *Iterated Partitioned Block Frequency–Domain Adaptive Filtering for Acoustic Echo Cancellation.* Submitted for publication, May 2001.

[5] S. Gay. *Fast Projection Algorithms with Application to Voice Echo Cancellation.* PhD thesis, Rutgers, The State University of New Jersey, New Brunswick, New Jersey, USA, October 1994.

[6] D. Morgan and S. Kratzer. *On a Class of Computationally Efficient, Rapidly Converging, Generalized NLMS Algorithms.* IEEE Signal Processing Letters, 3(8):245–247, August 1996.

[7] J. Páez Borrallo and M. García Otero. *On the implementation of a partitioned block frequency domain adaptive filter (PBFDAF) for long acoustic echo cancellation.* Signal Processing, 27:301–315, June 1992.

[8] J. Shynk. *Frequency-Domain and Multirate Adaptive Filtering.* IEEE Signal Processing Magazine, 9(1):15–37, January 1992.