A FREQUENCY DOMAIN ADAPTIVE ALGORITHM FOR COLORED MEASUREMENT NOISE ENVIRONMENT

Tõnu Trump

Ericsson Radio Systems AB
S–164 80 Stockholm, Sweden
tonu.trump@era.ericsson.se

ABSTRACT

The problem of incorporating partial knowledge of measurement noise into a frequency domain adaptive filtering scheme is addressed.The proposed algorithm is obtained by minimizing a BLUE criterion function using the stochastic gradient method and then switching over to the frequency domain to reduce the computational complexity. The performance of the algorithm in the situations of colored measurement noise is demonstrated by means of simulations using stationary as well as speech signals.

1. INTRODUCTION

Most of the existing adaptive algorithms are designed to minimize in some sense the error signal power or some estimate of it [3]. This approach implicitly assumes that the measurement noise is white or absent, an assumption which may or may not be satisfied in practice. The assumption is likely to be satisfied if the measurement noise is dominated by thermal and quantization noises. On the other hand in the applications like voice echo cancellation the measurement noise mainly consist of background sounds in a room and the assumption of whiteness can easily be violated.

In this paper a best linear unbiased estimate (BLUE) criterion function is used as the target function in the design of an adaptive algorithm. By this approach the color of the measurement noise can be accounted for in the algorithm and a better performance can be expected.

Some earlier work in a similar direction can be found in [4] and [6], where the measurement noise is assumed to be an autoregressive (AR) process. It is then proposed to apply a whitening filter to the residual signal before it is fed back to the coefficient updates. In [6] it is assumed that the proper whitening filter is known *a priori* while an adaptive whitening is considered in [4].

The approach taken in this paper differs from these of [4] and [6] in several ways. First, we do not assume any particular model for the measurement noise (it is, though, assumed that a correlation matrix of reasonable size describes the noise properties adequately). Second, a frequency domain algorithm is developed in this paper. Third, a variant of the BLUE criterion known for its good numerical properties [8] is minimized here. Fourth, the way of obtaining information concerning the measurement noise is somewhat intermediate to these of [4] and [6]. More specifically, as the main focus in this paper is on the voice echo cancellation applications, the noise correlation can be estimated during the natural pauses in speech.

In the following the italic, bold face lower case and bold face upper case letters are used for scalars, column vectors and matrices respectively. Superscripts T, H and $\dagger$ denote transpose, Hermitian transpose and Moore–Penrose pseudo–inverse. $\mathbf{I}$ is the identity matrix and $\mathbf{0}$ is matrix of all zeros.

2. DERIVATION

Consider the criterion function

$$V = (\mathbf{X}\mathbf{h} - \mathbf{y})^T(\gamma\mathbf{R}_v + \mathbf{X}\mathbf{X}^T)^\dagger(\mathbf{X}\mathbf{h} - \mathbf{y}), \quad (1)$$

where $\mathbf{h}$ is the N vector of unknown filter coefficients, γ is a positive constant, $\mathbf{X}$ is the $M \times N$ matrix of the input signal samples $\mathbf{X} = [\mathbf{x}(t)\ldots\mathbf{x}(t-N+1)]$ with $\mathbf{x}(t) = [x_{t-M+1}, \ldots, x_t]^T$. This definition of $\mathbf{X}$ is perhaps not the most consequent one but it leads to the frequency domain algorithms which show strong similarities with the classical overlap–save frequency domain adaptive filter (FDAF) algorithms described in [1, 3, 9]. The vector $\mathbf{y} = \mathbf{X}\mathbf{h} + \mathbf{v}$ is the M vector of observed signal samples and $\mathbf{v}$ is the M vector of measurement noise with symmetric Toeplitz covariance matrix $\mathbf{R}_v$.

Derivative of the criterion function with respect to the unknown vector $\mathbf{h}$ reads

$$\begin{aligned}\frac{\partial V}{\partial \mathbf{h}} &= 2[\mathbf{X}^T(\gamma\mathbf{R}_v + \mathbf{X}\mathbf{X}^T)^\dagger\mathbf{X}\mathbf{h} - 2\mathbf{X}^T(\gamma\mathbf{R}_v + \mathbf{X}\mathbf{X}^T)^\dagger\mathbf{y} \\ &= -2\mathbf{X}^T(\gamma\mathbf{R}_v + \mathbf{X}\mathbf{X}^T)^\dagger\mathbf{e}, \quad (2)\end{aligned}$$

where $\mathbf{e} = \mathbf{y} - \mathbf{X}\mathbf{h}$. The filter coefficient estimate is thus

$$\hat{\mathbf{h}} = (\mathbf{X}^T(\gamma\mathbf{R}_v + \mathbf{X}\mathbf{X}^T)^\dagger\mathbf{X})^{-1}\mathbf{X}^T(\gamma\mathbf{R}_v + \mathbf{X}\mathbf{X}^T)^\dagger\mathbf{y}. \quad (3)$$

In a way similar to that used in [8] pp. 89–90 it can be shown that (3) is the BLUE of $\mathbf{h}$ for the case of possibly singular $\mathbf{R}_v$. It is further argued in [8] that this estimate (for the case $\gamma = 1$) has better numerical properties than the more traditional BLUE resulting from minimization of $(\mathbf{X}\mathbf{h} - \mathbf{y})^T\mathbf{R}_v^{-1}(\mathbf{X}\mathbf{h} - \mathbf{y})$ which is used in [4].

Next, let us build a gradient algorithm minimizing (1) as

$$\mathbf{e}(t) = \mathbf{y}(t) - \mathbf{X}(t)\hat{\mathbf{h}}(t) \quad (4)$$

$$\hat{\mathbf{h}}(t+1) = \hat{\mathbf{h}}(t) + \mu\mathbf{X}^T(t)(\gamma\mathbf{R}_v + \mathbf{X}(t)\mathbf{X}^T(t))^\dagger\mathbf{e}(t). \quad (5)$$

In the above equations the length of the entire signal, M has been replaced by a block length, $L < M$ in all corresponding matrix dimensions. Note, that in the case of white v, the algorithm coincides with the relaxed and regularized form of the affine projection algorithm [2] with the regularization parameter proportional to the measurement noise variance.

In this paper we are, however, interested in building a frequency domain algorithm so we proceed defining an $(N+L) \times (N+L)$ cyclically extended input signal matrix

$$\mathbf{X}_c(t) = \text{cycl}([x_{t-N-L+1}\ x_t\ x_{t-1}\ \ldots\ x_{t-N-L+2}])$$

$$= \begin{bmatrix} x_{t-N-L+1} & x_t & \ldots & x_{t-N-L+2} \\ x_{t-N-L+2} & x_{t-N-L+1} & \ldots & x_{t-N-L+3} \\ & & \ldots & \\ x_{t-L} & x_{t-L+1} & \ldots & x_{t-L-1} \\ x_{t-L+1} & x_{t-L} & \ldots & x_{t-L-2} \\ & & \ldots & \\ x_t & x_{t-1} & \ldots & x_{t-N-L+1} \end{bmatrix}$$

Note, that this is just one particular cyclic extension of several possible. The advantage of this one is that it leads to intuitively well understandable algorithms. The same extension was used in [1]. The original input signal matrix $\mathbf{X}(t)$ appears in the lower left corner of $\mathbf{X}_c(t)$ and consequently

$$\mathbf{X}(t) = [\mathbf{0}_{L\times N}\ \mathbf{I}_L]\mathbf{X}_c(t) \begin{bmatrix} \mathbf{I}_N \\ \mathbf{0}_{L\times N} \end{bmatrix} \tag{6}$$

The matrix $\mathbf{X}_c(t)$ is right–cyclic by construction and so is $\mathbf{X}_c^T(t)$. The eigendecomposition of a right–cyclic matrix (see e.g. [5]) is given by

$$\mathbf{X}_c^T = \mathbf{F}\mathbf{\Lambda}\mathbf{F}^H, \tag{7}$$

where $\mathbf{F}$ is the $(N+L)\times(N+L)$ discrete Fourier transform matrix with $\mathbf{F}_{kl} = \frac{1}{\sqrt{N+L}} exp(\frac{-j2\pi kl}{N+L})$ and $\mathbf{\Lambda}$ is a diagonal matrix formed by the discrete Fourier transform of the first row of the matrix $\mathbf{X}_c^T$ i.e.

$$\mathbf{\Lambda} = \text{diag}(\text{DFT}([x_{t-N-L+1}\ldots x_t])).$$

In the case of real valued signals the transpose and Hermitian transpose of $\mathbf{X}_c$ are the same and, hence, $\mathbf{X}_c^T = \mathbf{F}\mathbf{\Lambda}\mathbf{F}^H = \mathbf{F}^H\mathbf{\Lambda}^H\mathbf{F}$ and $\mathbf{X}_c = \mathbf{F}\mathbf{\Lambda}^H\mathbf{F}^H = \mathbf{F}^H\mathbf{\Lambda}\mathbf{F}$.

Substituting (6) and (7) into (4) and (5) we obtain

$$\mathbf{e}(t) = \mathbf{y}(t) - [\mathbf{0}_{L\times N}\ \mathbf{I}_L]\mathbf{F}^H\mathbf{\Lambda}\mathbf{F}\begin{bmatrix} \mathbf{I}_N \\ \mathbf{0}_{L\times N} \end{bmatrix}\hat{\mathbf{h}}(t) \tag{8}$$

$$\begin{aligned} \hat{\mathbf{h}}(t+1) &= \hat{\mathbf{h}}(t) + \mu[\mathbf{I}_N \mathbf{0}_{N\times L})]\mathbf{F}^H\mathbf{\Lambda}^H\mathbf{F} \\ &\quad \times \begin{bmatrix} \mathbf{0}_{N\times L} \\ \mathbf{I}_L \end{bmatrix} (\gamma\mathbf{R}_v + \mathbf{X}\mathbf{X}^T)^\dagger \mathbf{e}(t). \end{aligned} \tag{9}$$

Defining the frequency response vector of the unknown system as

$$\hat{\mathbf{f}}(t) = \mathbf{F}\begin{bmatrix} \mathbf{I}_N \\ \mathbf{0}_{L\times N} \end{bmatrix}\hat{\mathbf{h}}(t) \tag{10}$$

and introducing vector $\mathbf{g}(t)$ given by

$$\mathbf{g}(t) = \mathbf{F}\begin{bmatrix} \mathbf{0}_{N\times L} \\ \mathbf{I}_L \end{bmatrix}(\gamma\mathbf{R}_v + \mathbf{X}(t)\mathbf{X}^T(t))^\dagger\mathbf{e}(t) \tag{11}$$

we obtain

$$\mathbf{e}(t) = \mathbf{y}(t) - [\mathbf{0}_{L\times N}\ \mathbf{I}_L]\mathbf{F}^H\mathbf{\Lambda}\hat{\mathbf{f}}(t) \tag{12}$$

$$\hat{\mathbf{f}}(t+1) = \hat{\mathbf{f}}(t) + \mu\mathbf{F}\begin{bmatrix} \mathbf{I}_N \\ \mathbf{0}_{L\times N} \end{bmatrix}[\mathbf{I}_N\mathbf{0}_{N\times L}]\mathbf{F}^H\mathbf{\Lambda}^H\mathbf{g}(t). \tag{13}$$

One possibility is to stop right here approximating $\gamma\mathbf{R}_v + \mathbf{X}(t)\mathbf{X}^T(t)$ by a Toeplitz matrix and using e.g. the Levinson algorithm [5] to find $(\gamma\mathbf{R}_v + \mathbf{X}(t)\mathbf{X}^T(t))^\dagger\mathbf{e}(t)$. For the case of $L \ll N$ this approach would give a satisfactory computational complexity $(O((N+L)\log_2(N+L)) + O(L^2))$. This algorithm will be referred to as ALGO1 in the simulation study where it is used as a reference for validation of the following approximations.

However, the complexity of the Levinson algorithm is not in all cases acceptable and we proceed to find a simpler approximate algorithm. As shown in Appendix, the vector $\mathbf{g}$ can alternatively be expressed as

$$\begin{aligned} \mathbf{g}(t) &= \left(\mathbf{F}\begin{bmatrix} \mathbf{0}_{N\times L} \\ \mathbf{I}_L \end{bmatrix}(\gamma\mathbf{R}_v + \mathbf{X}(t)\mathbf{X}^T(t))\right. \\ &\quad \left.\times\ [\mathbf{0}_{L\times N}\mathbf{I}_L]\mathbf{F}^H\right)^\dagger \mathbf{F}\begin{bmatrix} \mathbf{0}_{N\times 1} \\ \mathbf{e}(t) \end{bmatrix}. \end{aligned} \tag{14}$$

We now approximate the input signal correlation matrix by a Toeplitz matrix $\mathbf{R}_x \approx \mathbf{X}(t)\mathbf{X}^T(t)$ and define a cyclic extension of the sum of the correlation matrices, $\mathbf{R}_c$, satisfying

$$\gamma\mathbf{R}_v + \mathbf{X}(t)\mathbf{X}^T(t) \approx \gamma\mathbf{R}_v + \mathbf{R}_x = [\mathbf{0}_{L\times N}\mathbf{I}_L]\mathbf{R}_c\begin{bmatrix} \mathbf{0}_{N\times L} \\ \mathbf{I}_L \end{bmatrix}.$$

Assuming that $L \leq N+1$, the first row of $\mathbf{R}_c$, obeys even DFT symmetry and hence it can be decomposed as

$$\mathbf{R}_c = \mathbf{F}^H\mathbf{D}\mathbf{F} = \mathbf{F}\mathbf{D}\mathbf{F}^H, \tag{15}$$

where $\mathbf{D}$ is a diagonal matrix with discrete Fourier transformed first row of $\mathbf{R}_c$ on the main diagonal. The main diagonal of $\mathbf{D}$ is real and symmetric by construction. The equation (14) can now be rewritten as

$$\mathbf{g}(t) = (\mathbf{K}\mathbf{D}\mathbf{K})^\dagger\mathbf{F}\begin{bmatrix} \mathbf{0}_{N\times 1} \\ \mathbf{e}(t) \end{bmatrix}, \tag{16}$$

where

$$\mathbf{K} = \mathbf{F}\begin{bmatrix} \mathbf{0}_{N\times L} \\ \mathbf{I}_L \end{bmatrix}[\mathbf{0}_{L\times N}\mathbf{I}_L]\mathbf{F}^H$$

can be seen as a window matrix. It is a right–cyclic matrix with elements of the first row given by

$$\mathbf{K}(1,n) = \frac{1}{N+L}\exp\left(\frac{-j\omega(2N+L-1)}{2}\right)\frac{\sin(\omega L/2)}{\sin(\omega/2)},$$

and $\omega = \frac{2\pi n}{N+L}$, which corresponds to a rectangular window in the time domain. Using (16) in (13) would still result in an algorithm with relatively high computational complexity and some more approximations are desirable. A particularly simple algorithm can be obtained approximating $\mathbf{K}$ by an identity matrix. In fact, this approximation results in an algorithm very similar to the self–orthogonalizing FDAF [9] with signal power at frequency bins (which is used to normalize the step sizes at respective frequencies) replaced by the weighted sum of signal and measurement noise powers.

Up to now we have assumed that the measurement noise correlation matrix, $\mathbf{R}_v$, is known. In practice this is often not the case and $\mathbf{R}_v$ has to be estimated. In voice echo cancellation applications, it is natural to initialize the estimate with $\sigma_v^2\mathbf{I}$, where σ_v^2 is some expected noise power. Note that in some applications, it may

INITIALIZATION
$\hat{\mathbf{f}} = \mathbf{0}_{(N+L)\times 1}$ $\mathbf{P}_x = \sigma_x^2 \mathbf{I}_{N+L}$ $\mathbf{P}_v = \sigma_v^2 \mathbf{I}_{N+L}$
FOR EACH NEW BLOCK OF L **INPUT SAMPLES:**
$\mathbf{\Lambda} = \mathrm{diag}(\mathrm{DFT}([x_{t-N-L+1} \dots x_t]))$ $\mathbf{e}(t) = \mathbf{y}(t) - [\mathbf{0}_{L\times N)}\ \mathbf{I}_L]\mathbf{F}^H \mathbf{\Lambda}\hat{\mathbf{f}}(t)$ $\mathbf{c}(t) = \mathbf{F}\begin{bmatrix} \mathbf{0}_{N\times 1} \\ \mathbf{e}(t) \end{bmatrix}$ If $\mathrm{Tr}(\mathbf{\Lambda}\mathbf{\Lambda}^H) > th$ then compute: $\mathbf{P}_x = \beta\mathbf{P}_x + (1-\beta)\mathbf{\Lambda}\mathbf{\Lambda}^H$ $\mathbf{D} = \mathbf{P}_x + \gamma\mathbf{P}_v$ $\mathbf{g}(t) = \mathbf{D}^\dagger \mathbf{c}(t)$ $\hat{\mathbf{f}}(t+1) = \hat{\mathbf{f}}(t) + \mu\mathbf{F}\begin{bmatrix} \mathbf{I}_N \\ \mathbf{0}_{L\times N} \end{bmatrix}[\mathbf{I}_N\,\mathbf{0}_{N\times L}]\mathbf{F}^H\mathbf{\Lambda}^H\mathbf{g}(t)$ Else compute: $\mathbf{P}_v = \beta\mathbf{P}_v + (1-\beta)\mathrm{diag}(\mathbf{c})\mathrm{diag}(\mathbf{c})^H$

Table 1: The proposed algorithm (ALGO2).

be possible to obtain a better initial estimate during the initialization phase. The estimate can then be refreshed during the natural pauses in speech i.e. while $\mathrm{Tr}(\mathbf{\Lambda}\mathbf{\Lambda}^H)$ is larger than some threshold, th . This procedure does not have any significant impact to the computational complexity as the coefficients should not be updated when the input signal power is low anyway [7]. Obviously one can update $\mathbf{D}$ directly in the frequency domain instead of first updating $\mathbf{R}_c$ and then computing the DFT.

The equations above can be applied every sample or once per a block of samples. In this paper we have chosen to apply them once every block of L samples. This implies a lower computational complexity for the price of L samples delay and somewhat slower initial convergence. The final algorithm (ALGO2) is outlined in Table 1.

3. SIMULATION RESULTS

A simulation study is conducted with the aim of comparing the performances of ALGO1, ALGO2 with each other and with that of FDAF [9]. Another aim for the study is to evaluate the performance of the algorithms with speech signals.

3.1. Stationary signals

Simulations with stationary signals are used to evaluate the validity of approximations made while going from ALGO1 to ALGO2 and to compare the performance of both of them to that of FDAF. In the context of this paper, stationarity of signals means that the statistics of the signals do not depend on time for $t > 0$.

A typical simulation result is shown in Figure 1, where the upper plot shows learning curves of the algorithms i.e. $E[(e(t) - v(t))^2]$ and the lower plot shows the corresponding weight errors, $E[\frac{\|\mathbf{h}-\hat{\mathbf{h}}(t)\|^2}{\|\mathbf{h}\|^2}]$. The curves are ensemble averages over 200 independent trials. Input signal is an AR process generated by $\frac{1}{1+0.7z^{-1}-0.1z^{-2}+0.1z^{-3}+0.01z^{-4}}$. Measurement noise is an AR

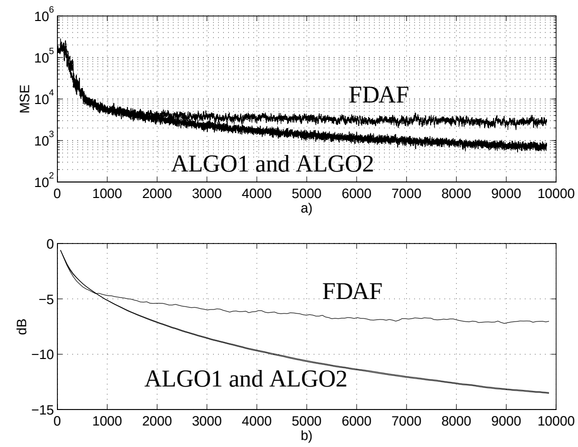

Figure 1: Results with stationary signals. a) learning curves; b) weight errors.

process generated by $\frac{1}{1-0.93z^{-1}+0.01z^{-2}}$. SNR is approximately 0dB. The true impulse response is a 64 taps FIR filter with flat frequency response and coefficients distributed uniformly in the interval $[-1/128, 1/128]$. $N = 64$, $L = 64$, $\mu = 0.04$, $\beta = 0.99$, $\gamma = 128$. As the signals are stationary, the measurement noise spectrum is estimated first (this time is not shown in the Figure above) and then used in the adaptive algorithm.

The curves corresponding to ALGO1 are not distinguishable from these of ALGO2 which indicates that the approximations have been reasonably good. Both of the algorithms outperform the FDAF.

3.2. Measured signals

Here we study the performance of ALGO2 in the acoustic echo cancellation application for the car hands-free telephone problem. Figure 2 presents the learning curves (averaged in time using an exponential window) of one typical simulation. The $N = 256$ tap echo impulse response identified in a Volvo 940 with two people sitting on the front seat of the car is used. $L = 256$, $\mu = 0.04$, $\beta = 0.99$, $\gamma = 128$. $\mathbf{P}_v$ is estimated on–line. A female speech signal is generating the echo which is then corrupted by a noise recorded in a moving car. The sampling rate is 8000 Hz so the curves in the Figure 2 correspond to a one minute time interval. The echo power is lower than the noise power most of the time but as the signals are concentrated in different frequency bands, the echo is clearly audible before the adaptive processing. FDAF improves the situation but the echo remains audible after the processing. After the initial convergence, the residual echo of the proposed algorithm is hardly audible in the noise.

Figure 3 shows the power spectra computed using samples 394000 to 410000 from Figure 2. It can be seen that the echo signal power is above the measurement noise power at most of the frequencies. Processing by FDAF attenuates echo but the residual echo power remains above measurement noise power at several frequencies. Finally, processing by ALGO2 reduces the residual echo power below the measurement noise power at all the frequencies and, hence, the masking effects of human perception make the echo inaudible.

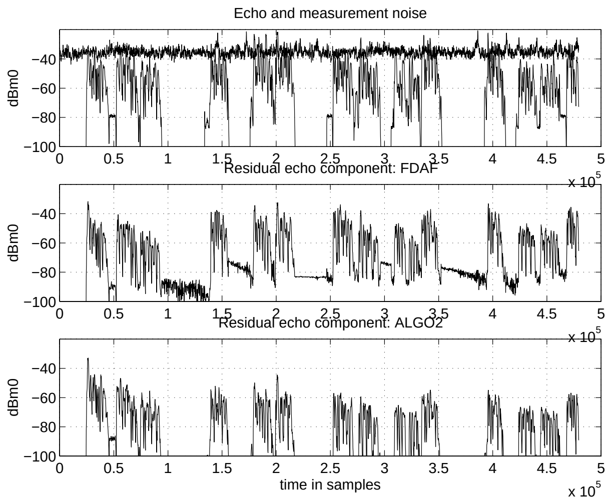

Figure 2: Results with measured signals. Upper: powers of car noise and echo; Middle: residual echo power of FDAF; Lower: residual echo power of ALGO2.

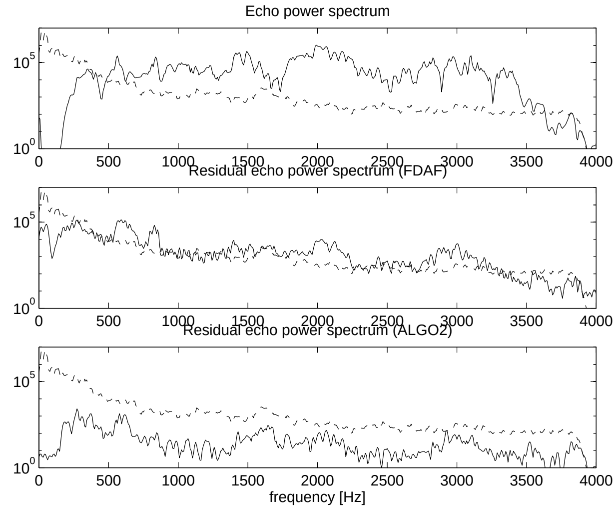

Figure 3: Power spectra. In all the plots the dashed line is the measurement noise spectrum. Upper: Echo spectrum; Middle: residual echo spectrum of FDAF; Lower: residual echo spectrum of ALGO2.

4. CONCLUSIONS

A novel adaptive algorithm has been derived by minimizing a BLUE criterion function and using eigenproperties of cyclic matrices. From several ways of cyclic extension of signal matrices the one leading to algorithms similar to the classical frequency domain adaptive filter has been used. The good performance of the proposed algorithm has been demonstrated by simulations.

5. REFERENCES

[1] G. A. Clark, S. R. Parker and S. K. Mitra, "A Unified Approach to Time– and Frequency–Domain Realization of FIR Adaptive Digital Filters," *IEEE Trans. Acoustics, Speech and Signal Processing,* Vol. 31, pp.1073–1083, Oct. 1983.

[2] S. L. Gay and S. Tavathia, "The Fast Affine Projection Algorithm," *Proc. ICASSP'95,* Detroit, vol.5 pp.3023–3026, May, 1995.

[3] S. Haykin, *"Adaptive Filter Theory,"* Third Edition, Prentice Hall, 1996.

[4] K.C. Ho, "A Minimum Misadjustment Adaptive FIR Filter," *IEEE Trans. Signal Processing,* Vol. 44, pp. 577–585, March, 1996.

[5] S. L. Marple, Jr., *"Digital Spectral Analysis with Applications,"* Prentice Hall, 1987.

[6] A. C. Orgren, S. Dasgupta, C. E. Rohrs and N. R. Malik, "Noise Cancellation with Improved Residuals," *IEEE Trans. Signal Processing,* Vol. 39, pp.2629–2639, Dec. 1991.

[7] T. Petillon, A. Gilloire and S. Theodoridis, "The Fast Newton Transversal Filter: An Efficient Scheme for Acoustic Echo Cancellation in Mobile Radio," *IEEE Trans. Signal Processing,* Vol. 42, pp. 509–517, March 1994.

[8] T. Söderström and P. Stoica, *"System Identification,"* Prentice Hall, 1988.

[9] J. J. Shynk, "Frequency–Domain and Multirate Adaptive filtering," *IEEE Signal Processing Magazine,* Jan. 1992, pp.14–37.

6. APPENDIX

Consider the equation

$$\mathbf{R}\mathbf{z} = \mathbf{e},$$

where $\mathbf{R} = \gamma\mathbf{R}_v + \mathbf{X}\mathbf{X}^T$ is a nonnegative definite $L \times L$ matrix and $\mathbf{z}$ and $\mathbf{e}$ are L vectors. The latter is clearly equivalent to

$$\begin{bmatrix} \mathbf{0}_N & \mathbf{0}_{N\times L} \\ \mathbf{0}_{L\times N} & \mathbf{R} \end{bmatrix} \begin{bmatrix} \mathbf{0}_{N\times 1} \\ \mathbf{z} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{N\times 1} \\ \mathbf{e} \end{bmatrix},$$

which can be rewritten as

$$\mathbf{F} \begin{bmatrix} \mathbf{0}_N & \mathbf{0}_{N\times L} \\ \mathbf{0}_{L\times N} & \mathbf{R} \end{bmatrix} \mathbf{F}^H \mathbf{F} \begin{bmatrix} \mathbf{0}_{N\times 1} \\ \mathbf{z} \end{bmatrix} = \mathbf{F} \begin{bmatrix} \mathbf{0}_{N\times 1} \\ \mathbf{e} \end{bmatrix}.$$

As by (11)

$$\mathbf{g} = \mathbf{F} \begin{bmatrix} \mathbf{0}_{N\times 1} \\ \mathbf{z} \end{bmatrix}$$

we may write

$$\mathbf{g} = \left(\mathbf{F} \begin{bmatrix} \mathbf{0}_N & \mathbf{0}_{N\times L} \\ \mathbf{0}_{L\times N} & \mathbf{R} \end{bmatrix} \mathbf{F}^H \right)^{\dagger} \mathbf{F} \begin{bmatrix} \mathbf{0}_{N\times 1} \\ \mathbf{e} \end{bmatrix},$$

which is equivalent to (14).