Practical Comparison of Adaptive Filter Algorithms

Glenn E. Johnson, Robert A. Muir, Joseph M. Scanlan, and William M. Steedly

The Analytic Sciences Corporation (TASC)
Reston, VA 22090

Abstract

Linear filters can be designed to pass signals of interest and suppress undesired signal components. Wiener [1] originated work to determine conditions for an optimal linear filter. This paper reviews the problem of optimally reconstructing one signal as a filtered version of a second signal. This problem has applications in noise reduction, echo or interference cancellation from a signal of interest, system modeling, and equalization. Also included is an overview of methods to implement optimal, adaptive linear filters. The focus of the paper is on contrasting two methods for the particular problem of restoring speech signals in loud audio backgrounds. The discussion includes consideration of the many problems encountered in applications of adaptive processing. Trade-offs include sacrificing steady-state MSE for adaptation rate, controlling errors introduced by round-off, selecting filter order, and operation count (complexity).

1 Introduction

In the joint process interference cancellation problem, illustrated in Figure 1, a desired signal is corrupted by a linearly filtered version of an available noise reference. An adaptive filter is employed to recover the desired signal with a minimum of distortion. Recovering a desired signal from a corrupted copy is one of the many successful applications of adaptive filtering, applications which include noise reduction, echo cancellation, system modeling, and equalization. Wiener [1] originated work to define optimality conditions and to construct optimal linear filters. This paper contrasts several methods of signal reconstruction using the example of recovering speech from noise (audio background) which was sampled from a reverberant environment. Satisfactory solutions to this task required consideration of filters with thousands of taps, adapting to nonstationary environments, and required

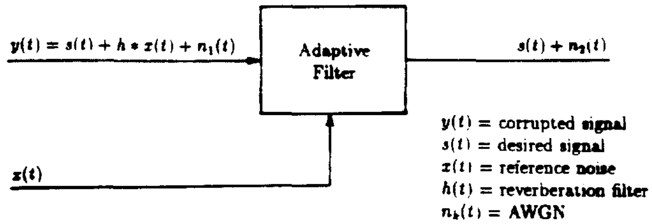

Figure 1: Joint process interference cancellation.

resolutions for a number of design considerations including trading increases in steady-state MSE for more rapid rates of adaptation, mollifying errors introduced by truncation (*e.g.*, 22-bit or 32-bit word sizes), and selection of filter order.

2 Adaptive filter methods

The optimality condition applied by Wiener was to minimize the mean squared error (MSE) between the recovered signal and the desired signal. Referring again to Figure 1, the Wiener solution to the joint process estimation problem is the filter, W, which minimizes the mean value,

$$\text{MSE} = E(y(t) - s(t) - W * x(t))^2 \tag{1}$$

The Wiener, minimal MSE filter is,

$$\begin{aligned} W &= (Exx^T)^{-1}(Exy^T) \\ &\equiv R_{xx}^{-1} R_{xy}. \end{aligned} \tag{2}$$

Sampled time, vector notation has been adopted together with expressing time in units of the sample period, $T_s = 1$. A vector x has N elements, $x(t-k)$ for $k = 0, 1, \ldots, N-1$. N is the length of the impulse

1058-6393/95 $4.00 © 1995 IEEE

response of the filter, W, used to reconstruct the corrupted signal. With desired signal and noise uncorrelated, an optimal (MSE) filter, W, can be constructed without regard for the statistics, in particular the time dependence, of the signal and noise. If the covariance is singular, a pseudo-inverse is selected, with the Moore-Penrose solution a natural choice. For jointly normal processes, the Wiener filter provides the maximum likelihood estimate while in general, equation (2) provides the best linear filter for signal reconstruction.

Several results are common to all methods used to estimate the filter, W. First, minimizing MSE requires that estimates for R_{xx} and R_{xy} be at least as long as the impulse response of the reverberation filter, h. Denoting the impulse response length of the reverberation filter as H, the Wiener filter for a joint process interference cancellation problem is $W(\ell) = h(\ell)$ for $\ell \leq \min(N, H)$.

Minimizing MSE requires that $N \geq H$, and efficiency suggests that N be minimized. If $N < H$, then the MSE includes a term, $E|\sum_{k=N+1}^{H} h(k)x(t-k)|^2$, that vanishes if N is increased to H. For IIR reverberation filters, H is cutoff at some effective length. Selecting a filter length which is significantly longer than the reverberation filter, h, carries the costs of unnecessary computations and additional round-off error.

Every adaptive filtering approach includes a method to estimate the covariances, R_{xx} and R_{xy}, from windows of data samples. Obtaining a number of samples to estimate the covariances requires ergodicity, that time averages approximate ensemble averages. If the processes, x and y, are wide sense stationary, then consistent estimates for the covariances are continually enhanced with the accumulation of data samples. And, although noise processes are generally nonstationary, the joint statistics often vary sufficiently slowly that methods applicable to stationary processes can be usefully employed. Using short term mean values for covariance estimates, the adaptive filter will track slow time variations of the noise.

There is a trade-off between enhancing performance by including more samples in the covariance estimates, and degrading performance by including stale covariance samples in the estimates. Pseudostationarity of the noise is required to even form reliable estimates of R_{xx} and R_{xy}. Nonstationarity results both from dynamics of the noise, $x(t)$, and from time variation of the reverberation filter, h. Achieving rapid adaptation improves performance, and increases the range of situations in which the adaptive filter is effective.

Adaptive filtering approaches, distinguished by convergence rates, steady state error, and implementation complexity, require design selections for the forgetting time, and the length of the filter. For discussion, adaptive filtering approaches have been grouped into four categories: sample matrix inversion (SMI); time domain least mean square (LMS); vector μ LMS; and recursive least squares (RLS).

2.1 Sample matrix inversion (SMI)

A straightforward adaptive filter implementation includes a covariance estimator together with a matrix inversion algorithm [2]. One Toeplitz covariance estimate with exponential forgetting is,

$$\hat{R}_{xx}^{(n+1)}(|j-k|) = \lambda\, \hat{R}_{xx}^{(n)}(|j-k|) + (1-\lambda)\, x(n)x(n-|j-k|) \tag{3}$$

with $\lambda = \exp(-T_s/\tau)$ and τ is the time constant for forgetting. The estimate for the Wiener filter is,

$$\hat{W} \equiv \hat{R}_{xx}^{-1}\hat{R}_{xy} \tag{4}$$

The covariance matrix estimate can be inverted by any of a variety of methods, including Gaussian elimination, order recursion, or matrix diagonalization with scalar divisions. However, since the covariance matrix has Toeplitz symmetry, it is more efficient to use the Levinson recursion [3] which is $O(N^2)$.

The SMI procedure is basically data block oriented, and requires $O(N)$ operations per data sample if N new data samples are accumulated before each matrix inversion. In this case, the filter updates only every Nth sample, and updating every new data sample makes this approach $O(N^2)$ per data sample.

2.2 Time domain least mean square (LMS)

Jacobi iteration is an approach to solve linear problems which originated in the 19th century. The fixed point of the iteration,

$$W^{(n+1)} = \mu R_{xy} + (1 - \mu R_{xx})W^{(n)} \tag{5}$$

is the Wiener solution for the optimal noise reduction filter. This iteration converges if R_{xx} is a covariance matrix and the gain satisfies,

$$0 < \mu < 2/\lambda_{max} \tag{6}$$

with λ_{max} the maximum eigenvalue of R_{xx}.

Substitution of the current data as estimates for R_{xy} and R_{xx} results in one form of the LMS procedure [4]. Rewritten for computational efficiency, the LMS algorithm is $O(N)$ operations per data sample,

$$\hat{W}^{(n+1)} = \alpha\hat{W}^{(n)} + \mu\left(y(n) - \hat{W}^{(n)} * x\right) x \tag{7}$$

This procedure effectively both estimates and inverts the covariance matrix. This procedure can be related to gradient descent on the MSE surface. Convergence of this stochastic gradient descent algorithm requires additional conditions on μ. For example, in jointly normal, quickly decorrelating noise, convergence of the MSE, $\|W - \hat{W}\|^2$, requires that,

$$\mu < \frac{2}{2\lambda_{max} + \text{Trace}(R_{xx})} \tag{8}$$

a stronger condition than that required for convergence in the mean. For covariance matrices, the trace is the sum of eigenvalues.

To mollify arithematic truncation problems, a leak parameter, $\alpha \leq 1$, is often included in LMS. This modification is required for nearly singular covariance matrices (matrices with a large ratio of maximum to minimum eigenvalues). However, any leak parameter value other than unity introduces a bias error to the Wiener filter estimates.

This readily implemented LMS procedure suffers two defects. First, the estimate never fully converges. Since the latest data is used to adjust the last filter estimate, the noisy gradient estimate continually perturbs the filter estimate from the Wiener solution. This steady state error is reduced by lowering the gain, μ, which in turn slows adaptation. This tradeoff, adaptation rate for steady state error reduction, is a general feature of adaptive filters, while the noisy gradient is peculiar to LMS.

The second defect is that adaptation rates are limited by the rate of convergence of the iteration. Convergence slows dramatically for covariance matrices with large eigenvalue disparities. The convergence rate can be characterized by the rate at which $(1 - \lambda_{min}/\lambda_{max})^n$ approaches zero as n increases. To converge substantially requires $O(\lambda_{max}/\lambda_{min})$ iterations, and this can be much slower than the dynamics of the noise.

2.3 Vector μ LMS

The vector μ LMS procedure speeds convergence for many noise processes. Every covariance matrix is unitary similar to a positive, diagonal matrix. Applying the diagonalizing transform to (5) results in,

$$UW^{(n+1)} = \mu UR_{xy} + (1 - \mu D_{xx})UW^{(n)} \tag{9}$$

with $D_{xx} = UR_{xx}U^{-1}$ a diagonal matrix. The entries along the diagonal are the eigenvalues of R_{xx}.

Now, since the equation for estimates of UW are decoupled, N one dimensional iterations with independent weights are implemented. Weight update coefficients, μ_k, are selected independently to maximize convergence rate in each of the N uncoupled iterations, $\mu_k \approx 1/D_{xx}(k,k)$. The Wiener filter is obtained by inverse transforming the estimates for UW.

Generally, the unitary transformation, U, is unknown and procedures to construct U given R_{xx} are at least $O(N^2)$. However, covariance matrices have a Toeplitz symmetry, and large, banded Toeplitz matrices are approximately circulant. The importance of this identification is that circulant matrices are diagonalized by the discrete Fourier transform (DFT) matrix. In this event, the unitary transformation is both known, and efficiently evaluated.

The equivalence of banded Toeplitz with circulant matrices is at best approximate. There are cases (anticirculant tones) for which the convergence properties of the vector μ LMS algorithm are worse than those of the time domain LMS.

If the current data samples are substituted as estimates for R_{xx} and R_{xy}, the resulting procedure is a vector μ LMS algorithm. Care is required to avoid circular convolution when linear convolution is intended [5].

Consistent with the approximations inherent to the method, the gains can be controlled according to, $\mu_k < 1/\zeta|\tilde{x}(k)|^2$ with $\zeta > 1$ a safety factor. The spectral estimate, $|\tilde{x}(k)|^2$, is a mean square (accumulated with forgetting) of the DFT of the input data.

The results of applying the frequency domain LMS algorithm as a noise cancellor to process speech have shown a substantial increase in speed of convergence, immunity to eigenvalue disparity, and a more uniform spectral cancellation in comparison to the time domain LMS algorithm.

This procedure, which potentially accelerates convergence of LMS, is $O(N \log N)$ per data sample. This procedure is typically implemented using a 5-FFT algorithm, data blocks of length N, with 50% block overlaps. In this implementation, the procedure is $O(\log N)$ per data sample, but slows adaptation rate.

2.4 Recursive least squares (RLS)

The RLS algorithms solve (4) each update. Conceptually, the RLS procedures are a return to SMI. The exact least squares filter is derived from each update using current covariance estimates. RLS efficiently produces the most rapidly converging adaptive filters.

Recursive least squares (RLS) algorithms include fast transversal filter (FTF), fast Kalman, and lattice algorithms [6, 7]. RLS algorithms achieve algorithmic efficiency by developing temporal and filter order updates for minimal MSE solutions to (1). Streamlined procedures to update a filter for the next data sample or to increase filter order by one are used.

Since a matrix inversion is involved, instability from numerical truncation should be anticipated, particularly if the covariance matrices have large eigenvalue disparities. However, constraints, normalizations, and rescue procedures mitigate the effects of truncation error. In the application of Shensa's [8] RLS algorithm reported in this paper, constraining an intermediate value to positive values by taking magnitudes cured instability problems.

For the joint process interference cancellation task, two operations are accomplished by the RLS algorithms. In the "lower" filter, the covariance of the noise reference is estimated and diagonalized,

$$\hat{D}_{xx} = U\hat{R}_{xx}U^{-1} \tag{10}$$

The RLS procedure achieves this diagonalization in $O(N)$ operations per sample, the same complexity as Levinson recursion. In the "upper" filter, a cross covariance estimate is used to construct the filter,

$$U\hat{W} = \hat{D}_{xx}^{-1}U\hat{R}_{xy} \tag{11}$$

The transformed filter is developed from the transformed cross covariance by N scalar divisions.

The RLS algorithms require $O(N)$ operations per data sample, making the procedures comparable in complexity to LMS (within a factor of 1.6 to 3.5 for FTF [6]) and, comparable to an update once per data block, Levinson recursion SMI.

3 Results

The experiment consisted of reconstructing a speaker's voice corrupted by additive audio noise. Cases with both a moving and non-moving speaker were conducted. For the case with motion, the speaker rapidly paced and waved his arms to create a nonstationary reverberant environment. The experiment was conducted in a 20×12 foot living room with an 8 foot ceiling. The acoustic absorbance was 15%. A sound system driven by a audio CD player was placed along the longer wall. The audio CD track was also recorded directly for use as the noise reference. A microphone was placed in the room center, and the room included a typical complement of furniture.

The desired signal is a speaker's voice which is to be separated from the audio noise. A recording from the microphone together with the isolated audio reference are the inputs to the adaptive filters. The noise on the microphone recording differs from the isolated audio reference due to the room acoustics. The room is modeled as a linear filter.

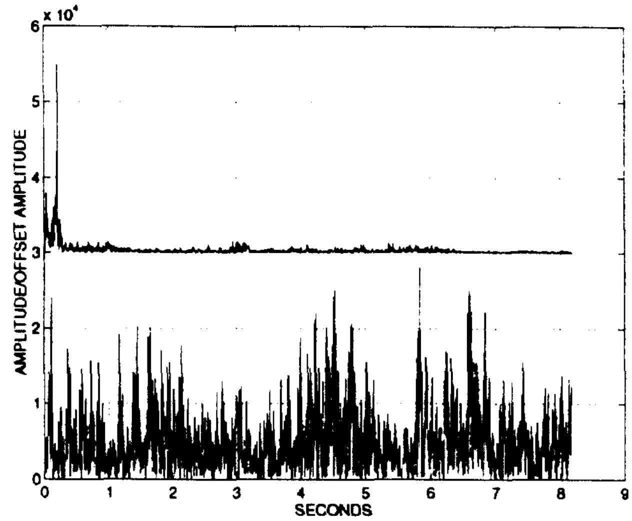

Figure 2: Initial processed output for the vector μ LMS (lower) and RLS (upper) algorithms for the non-moving speaker case.

The criteria used to assess speech enhancement was intelligibility of the processed audio in comparison to the unprocessed recording. Two minute recordings were made at 11,025 samples per second with a 16-bit resolution. The recording included AGC to maintain an optimal A/D loading.

The speech was unintelligible on the unprocessed recording for both the non-moving and moving speaker. The SIR was approximately -20 dB. After processing with both the vector μ LMS and the RLS, speech was intelligible. Noise reduction was about 25 and 15–20 dB for the vector μ LMS and about 28 and 20–25 dB for the RLS for the non-moving and moving speaker cases, respectively. Figures 2 and 3 show the initial rate of convergence for the two techniques for both cases. The initial convergence rates of the RLS were orders of magnifute times faster for the two cases. Figures 4 and 5 show the PSDs for the last 10 s of the two processed outputs for the non-moving and moving speaker cases, respectively. These figures show the extent to which noise cancellation is uniform across frequencies.

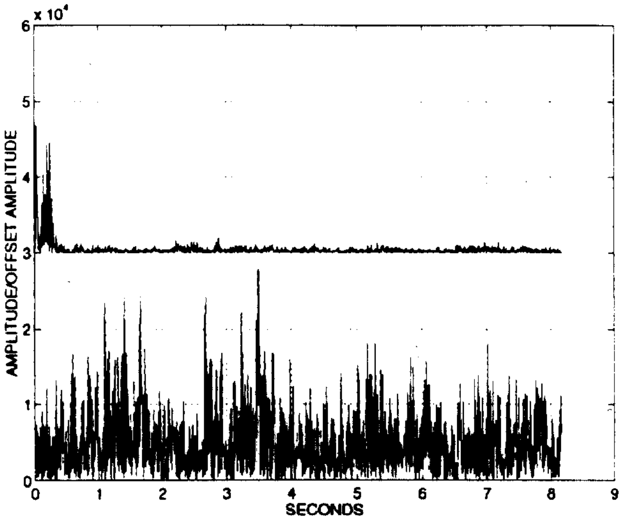

Figure 3: Initial processed output for the vector μ LMS (lower) and RLS (upper) algorithms for the moving speaker case.

4 Conclusions

The results illustrate the more rapid convergence of the RLS algorithm compared to the convergence rate of the vector μ LMS algorithm. As a result, the RLS algorithm had better noise cancellation in the nonstationary environments. As shown in Figures 4 and 5, the RLS algorithm has better uniform noise cancellation across frequencies.

References

[1] Wiener, N., *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*, MIT Press, Cambridge, Mass., 1942.

[2] Reed, I.S., J.D. Mallett, and L.E. Brennan, "Rapid Convergence Rates in Adaptive Arrays," *IEEE Trans. on Aerospace and Electronics Systems*, Vol. AES-10, November 1974, pp. 853-863.

[3] Levinson, N., "The Wiener RMS Error Criterion in Filter Design and Prediction," *J. Math. Phys.*, Vol. XXV/4, 1947, pp. 261-278.

[4] B. Widrow, and M.E. Hoff, "Adaptive Switching Circuits," *IRE WESON Conv. Rec. Pt. 4*, 1960, pp. 96-104.

[5] Ferrara, E.R., "Fast Implementation of LMS Adaptive Filters," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-28, No. 4, pp. 474-475, August, 1980.

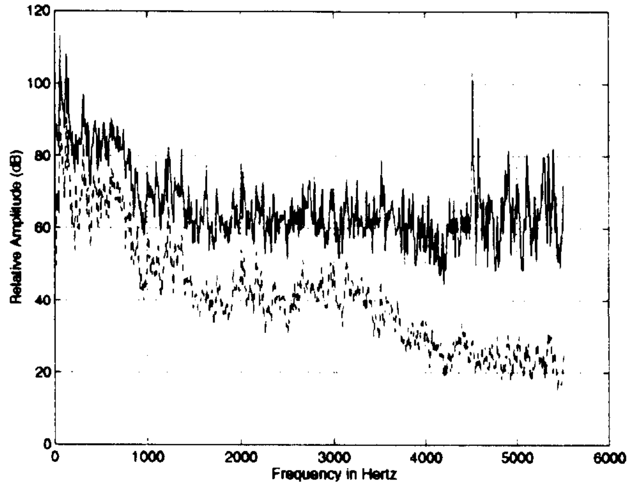

Figure 4: PSDs for the vector μ LMS (solid) and RLS (dashed) algorithms for the non-moving speaker case.

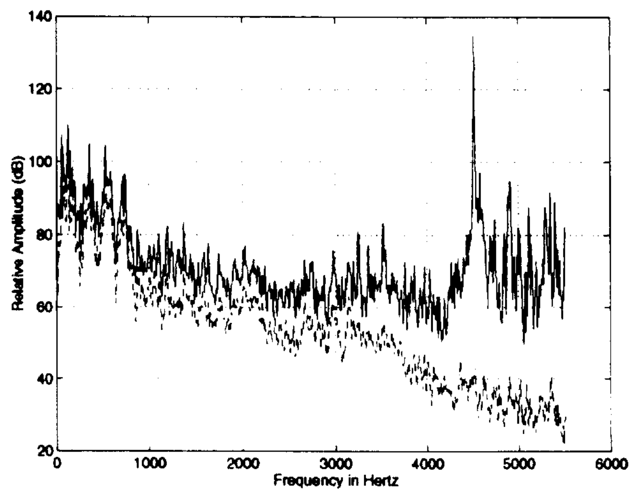

Figure 5: PSDs for the vector μ LMS (solid) and RLS (dashed) algorithms for the moving speaker case.

[6] Cioffi, J.M., and T. Kailath, "Fast, Recursive-Least-Squares Transversal Filters for Adaptive Filtering," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-32, April 1984, pp. 304-337.

[7] A.H. Sayed, and T. Kailath, "A State-Space Approach to Adaptive RLS Filtering," *IEEE Signal Processing Magazine*, July 1994, pp. 18-60.

[8] Shensa, M.J., "Recursive Least Squares Lattice Algorithms, A Geometric Approach," *IEEE Trans. on Automatic Control*, Vol. AC-26, June 1981, pp. 695-702.