ECHO CANCELLATION USING THE CONJUGATE GRADIENT ALGORITHM

Per Åhgren[a] *and Erik G. Larsson*[b,*]

[a]Department of Systems and Control, Information Technology, Uppsala University,
P.O. Box 27, SE-75103 Uppsala, Sweden.
`Per.Ahgren@syscon.uu.se`
[b]Dept. of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611.

ABSTRACT

In this paper an approximation to the sliding window Recursive Least Squares (RLS) algorithm with filter estimate updates using the conjugate gradient algorithm is applied to the Acoustic Echo Cancellation (AEC) problem for the mono case. The method is shown to perform much better than the Normalized Least Mean Squares (NLMS) algorithm which is one of the standard algorithms used for AEC today. While it performs somewhat worse than RLS which is the optimal (unbiased) estimator in a least squares sense, it is shown to be computationally much less demanding. In contrast to RLS we believe that it may be feasible to implement in real time and hence the method presented here should be one of the best algorithms to consider for the AEC problem.

1. INTRODUCTION

In hands-free communication [5] such as full-duplex teleconferencing the acoustic echoes of far-end speech needs to be removed. The method for doing this is called Acoustic Echo Cancellation and is usually performed using time domain FIR filters to model the acoustic echo paths and to predict the echoes. Ideally, the echoes can then be removed by removing the predicted echoes from the microphone signal. Studies show that room echoes can be very long and thus the FIR filters must be of very high orders with several thousand coefficients [2]. Since the algorithms for AEC should be feasible to implement on a standard Digital Signal Processor (DSP) it is therefore very important that the requirements of the filter estimation algorithms algorithms in terms of storage and computational complexity should not be too large. Since the echo paths are time-varying it is also very important that the algorithm rapidly adapts to changes in these. Finally it is very important that the adapted impulse responses are adapted to the true echo paths and not only to a solution that removes the echo.

A vast number of methods have been tested for the AEC problem [3] of which probably the most commonly used method, due to its low computational complexity, is the normalized least mean squares (NLMS) algorithm. It adapts, however, very slowly and what would be desired is an algorithm that adapts faster such as the recursive least squares (RLS) algorithm. The RLS[1] algorithm is, however, too computationally complex and requires too much memory to be implemented in a real-time application on a standard DSP. Instead algorithms such as the fast recursive least squares (FRLS) and the affine projection (AP) algorithm have been proposed. The FRLS algorithms suffer, however, from problems with numerical instability and it has been shown [6] that all known FRLS algorithms are unstable when using a forgetting factor less than one. The AP algorithm is inbetween the RLS and NLMS algorithms both regarding complexity and speed of adaptation and can trade increased adaptation speed against increased computational complexity.

In [6] a low complexity approximation of RLS with sliding data window was proposed which we will call Conjugate-Gradient RLS (CG-RLS). CG-RLS was shown to have a very fast speed of adaptation while requiring a relatively small amount of memory and being computationally much less demanding than the RLS algorithm. The goal of this paper is to apply the technique in [6] to the AEC problem.

2. CONJUGATE-GRADIENT RLS

The sliding window RLS algorithm tries to estimate a FIR filter $\mathbf{h}(t)$ that minimizes the sum of squared errors

$$\begin{aligned} V(t) = \sum_{j=t-M+1}^{t} |e(j)|^2 &\triangleq \sum_{j=t-M+1}^{t} |d(j) - \mathbf{h}^*(t)\mathbf{x}(t)|^2 \\ &\triangleq \|\mathbf{d}(t) - \mathbf{X}(t)\mathbf{h}(t)\|^2, \end{aligned} \tag{1}$$

[*]On leave from the Department of Systems and Control, Uppsala University, Uppsala, Sweden.

This work was partly supported by the Swedish Foundation for Strategic Research (SSF).

[1]Unless otherwise stated, we mean by RLS the (exact) recursive solution to the least squares problem [7].

where $\mathbf{h}(t)$ is the filter to estimate,

$$\mathbf{x}(t) \triangleq [x(t)\ x(t-1)\ \dots\ x(t-n+1)]^T \qquad (2)$$

is the input data vector, $d(j)$ is the known desired filter output,

$$\mathbf{d}(t) \triangleq [d(t-M+1)\ d(t-M+2)\ \dots\ d(t)]^T, \qquad (3)$$

n is the length of the filter $\mathbf{h}(t)$ and M is the length of the sliding data window. The Mxn Toeplitz matrix $\mathbf{X}(t)$ is defined as

$$\mathbf{X}(t) \triangleq \begin{bmatrix} \overline{x(t-M+1)} & \dots & \dots & \overline{x(t-M-n+2)} \\ \vdots & \ddots & & \vdots \\ \overline{x(t-n+1)} & & \ddots & \vdots \\ \vdots & \ddots & & \overline{x(t-M+1)} \\ \vdots & & \ddots & \vdots \\ \overline{x(t)} & \dots & \dots & \overline{x(t-n+1)} \end{bmatrix}, \qquad (4)$$

where $\overline{(\cdot)}$ denotes the complex conjugate. As is well-known, the filter $\mathbf{h}(t)$ minimizing (1) is given by the solution to the normal equations

$$\mathbf{X}^*(t)\mathbf{X}(t)\mathbf{h}(t) = \mathbf{X}^*(t)\mathbf{d}(t), \qquad (5)$$

which can be computed in a number of different ways. The perhaps most common approach is to form a recursion of $(\mathbf{X}^*(t)\mathbf{X}(t))^{-1}$ and $\mathbf{X}^*(t)\mathbf{d}(t)$ in time, and solve for $\mathbf{h}(t)$ by

$$\mathbf{h}(t) = (\mathbf{X}^*(t)\mathbf{X}(t))^{-1}\mathbf{X}^*(t)\mathbf{d}(t) \qquad (6)$$

as is done by RLS [6]. The approach taken in [6] is to solve (5) by using the conjugate gradient (CG) algorithm [4] which can be efficiently implemented for the least squares problem at hand using the Fast Fourier Transform (FFT) and its inverse (IFFT). This is an algorithm that iteratively solves (5) exactly from any initial estimate in at most as many steps as the number of parameters to estimate, i.e., n. If the initial estimate is good then CG will converge much faster to the solution. The method proposed in [6] is to solve (5) using CG, at time t using the solution obtained at time $t-1$ as an initial estimate. In that way only one or a few iterations per time step are required to ensure good adaptation properties of the algorithm.

The CG algorithm is presented in [4] but for completeness we present it here as well using the terminology used for our problem. Denoting

$$\mathbf{R}(t) \triangleq \mathbf{X}^*(t)\mathbf{X}(t) \qquad (7)$$

and

$$\mathbf{p}(t) \triangleq \mathbf{X}^*(t)\mathbf{d}(t) \qquad (8)$$

we describe the CG algorithm for solving (5) in Table 1. Note that in the CG algorithm in Table 1 the initial

$k = 0$
$\mathbf{r} = \mathbf{p}(t) - \mathbf{R}(t)\mathbf{h}(t-1)$
$\rho_0 = \|r\|_2^2$
While $(\sqrt{\rho_k} > \epsilon\|\mathbf{p}(t)\|_2$ and $k < k_{MAX})$
  $k = k + 1$
  If $k = 1$
    $\mathbf{q} = \mathbf{r}$
  else
    $\beta_k = \rho_{k-1}/\rho_{k-2}$
    $\mathbf{q} = \mathbf{r} + \beta_k\mathbf{q}$
  $\mathbf{w} = \mathbf{R}(t)\mathbf{q}$
  $\alpha_k = \rho_{k-1}/\mathbf{q}^T\mathbf{w}$
  $\mathbf{h}_k(t) = \mathbf{h}_{k-1}(t) + \alpha_k\mathbf{q}$
  $\mathbf{r} = \mathbf{r} - \alpha_k\mathbf{w}$
  $\rho_k = \|\mathbf{r}\|_2^2$
$\mathbf{h}(t) = \mathbf{h}_k(t)$

Table 1: The conjugate gradient algorithm.

filter estimate $\mathbf{h}(t)$ is set as the filter estimate of the previous time step as was done in [6]. Also note that the most computational demanding part in the algorithm is computing $\mathbf{R}(t)\mathbf{q}$, which for the CG algorithm to be computationally attractive must be done using as small a number of operations as possible. As we will see this matrix-vector product can be performed very efficiently using the FFT/IFFT. First note that $\mathbf{R}(t) = \mathbf{X}^*(t)\mathbf{X}(t)$ and that $\mathbf{X}(t)$ (as well as $\mathbf{X}^*(t)$) is a Toeplitz matrix. It is well known that any Toeplitz matrix $\mathbf{X}(t)$ can be extended to a circulant matrix

$$\mathbf{C} = \begin{bmatrix} \mathbf{X}(t) & \mathbf{E} \\ \mathbf{G} & \mathbf{H} \end{bmatrix}, \qquad (9)$$

where $\mathbf{E}$, $\mathbf{G}$ and $\mathbf{H}$ are suitably chosen matrices. Since circulant matrices can be decomposed as

$$\mathbf{C} = \mathbf{F}^*\mathbf{D}\mathbf{F} \qquad (10)$$

where $\mathbf{F}$ is the Fourier (or Discrete Fourier Transform) matrix and $\mathbf{D}$ is a diagonal matrix whose elements are the FFT of the first column of $\mathbf{C}$ (note that the dimension of the square matrix $\mathbf{C}$ can easily be adjusted to be a power of two so that the FFT can be applied). Thus the multiplication

$$\mathbf{X}(t)\mathbf{q} \triangleq \mathbf{a} \qquad (11)$$

can be computed as

$$\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} = \mathbf{C}\begin{bmatrix} \mathbf{q} \\ \mathbf{0} \end{bmatrix} = \mathbf{F}^*\mathbf{D}\mathbf{F}\begin{bmatrix} \mathbf{q} \\ \mathbf{0} \end{bmatrix}, \qquad (12)$$

where the value of $\mathbf{b}$ is unimportant. Since multiplying a Fourier matrix with a vector is an IFFT operation (or a FFT operation for the inverse of the Fourier matrix) the multiplication $\mathbf{X}(t)\mathbf{q}$ can be performed using the FFT/IFFT. The multiplication

$$\mathbf{X}^*(t)(\mathbf{X}(t)\mathbf{q}) \triangleq \mathbf{R}(t)\mathbf{q} \qquad (13)$$

can be computed in a similar manner using the FFT/IFFT.

We remark that the CG algorithm is often used with a *preconditioner*, leading to the so-called preconditioned CG (PCG) algorithm. This amounts to solve the transformed system

$$\widetilde{\mathbf{X}^*(t)\mathbf{X}}(t)\widetilde{\mathbf{h}(t)} = \widetilde{\mathbf{X}^*(t)\mathbf{d}}(t), \tag{14}$$

where

$$\widetilde{\mathbf{X}^*(t)\mathbf{X}}(t) \triangleq \mathbf{P}^{-1}(t)\mathbf{X}^*(t)\mathbf{X}(t)\mathbf{P}^{-1}(t), \tag{15}$$

$$\widetilde{\mathbf{X}^*(t)\mathbf{d}}(t) \triangleq \mathbf{P}^{-1}(t)\mathbf{X}^*(t)\mathbf{d}(t), \tag{16}$$

$$\widetilde{\mathbf{h}(t)} \triangleq \mathbf{P}(t)\mathbf{h}(t) \tag{17}$$

and the preconditioner $\mathbf{P}(t)$ is chosen such that $\widetilde{\mathbf{X}^*(t)\mathbf{X}}(t)$ has a low condition number. Since it appears for our problem that the CG algorithm works well also without preconditioning, and since preconditioning adds considerably to the computational complexity we do not consider it in this work.

It can be shown that the computational complexity of the CG-RLS algorithm is $\mathcal{O}((n+M)\log_2(n+M))$ (since the most computationally demanding operations can be performed using FFT) which is higher than the complexity of the NLMS algorithm that is $\mathcal{O}(n)$ but considerably lower than that for the RLS algorithm that is of complexity $\mathcal{O}(n^2)$. It can also be shown that the storage requirements for CG-RLS are of order $\mathcal{O}(n+M)$ which is comparable to those for NLMS and much lower than those for RLS ($\mathcal{O}(n^2)$). Also note that CG-RLS is highly parallellizable since it includes several FFT operations which are easily parallelized.

3. NUMERICAL EXAMPLES

To demonstrate the performance of CG-RLS for AEC we have performed numerical simulations, comparing CG-RLS to NLMS and RLS. Data was generated using a 1 second long speech sequence sampled at 16 kHz as a far-end signal and impulse responses obtained by a high-resolution acoustic impulse response prediction code. Four data sets were generated using four different time-invariant impulse responses of 600 filter taps each as the true room impulse responses. For each data set the signal to noise ratio (SNR) was adjusted to 40 dB (roughly corresponding to a hands-free telephony setup in a room with a low noise level) by adding white Gaussian noise (WGN) of suitable variance to the microphone input signal. To cancel the echo NLMS, RLS and CG-RLS were applied to all data sets to compute room impulse response estimates of lengths $n = 500$ filter taps each (shorter than the true room impulse responses since these are in reality of infinite length and thus always longer than the estimated impulse responses).

For NLMS the step size parameter was set to the standard choice of $\mu = 0.7$ to get a good adaptation speed while not being too sensitive to noise, and for the RLS algorithm the standard initialization of the P matrix, $P(t_0) = 100I$, was used. No forgetting factor was used for RLS (this makes the results for RLS somewhat better than if a forgetting factor was used which would be the case in a real scenario). To minimize the complexity of CG-RLS the length of the data window was chosen as $M = n$ which is the minimum to make $\mathbf{R}(t)$ full rank (which is a required for (1) to have a unique solution), and only one iteration of the CG algorithms was performed per sample. (Note that CG-RLS requires $M + n$ data samples to start the algorithm while NLMS and RLS only require n samples. This is easily seen in the plots for the numerical results and should be taken into account when comparing the performance of the algorithms.)

To assess the performance of the algorithms two measures were used: the Echo Return Loss Enhancement (ERLE) which measures the echo cancellation performance, and the misalignment [1] which measures the accuracy of the impulse response estimation. The ERLE is defined as

$$\begin{aligned} ERLE = -10\log_{10}\left(\frac{\sigma_e^2}{\sigma_d^2}\right) &\Rightarrow ERLE(t) \\ &\approx -10\log_{10}\left(\frac{\sum_{t-L+1}^{t} e^2(t)}{\sum_{t-L+1}^{t} d^2(t)}\right), \end{aligned} \tag{18}$$

where L is the length of a sliding window for the ERLE estimation, and the misalignment is defined as

$$M(t) = \frac{\|\mathbf{h}(t) - \tilde{\mathbf{h}}\|^2}{\|\tilde{\mathbf{h}}\|^2}, \tag{19}$$

where $\tilde{\mathbf{h}}$ is the true room impulse response truncated to be of the same length as $\mathbf{h}(t)$. For an algorithm to be well-suited for AEC the ERLE value should be as high as possible and the misalignment should be as low as possible.

The results are displayed in Figures 1-2 in terms of misalignment and ERLE achieved by the algorithms for the four different data sets. From the plot of the misalignment it is apparent that the adaptation of RLS to the room impulse response is much faster than that of CG-RLS, but also that the adaptation of CG-RLS is much faster than that of NLMS. Finally, from the plot of the echo cancellation performance (ERLE) it is clear that NLMS performs worst and that CG-RLS performs similarly to RLS.

At first sight it could seem strange that for some samples CG-RLS even outperforms RLS in terms of ERLE since both algorithms minimize (1). One should, however, keep in mind that the data window for RLS is infinite while it is finite (and relatively short) for CG-RLS and therefore the minimum of (1) achieved

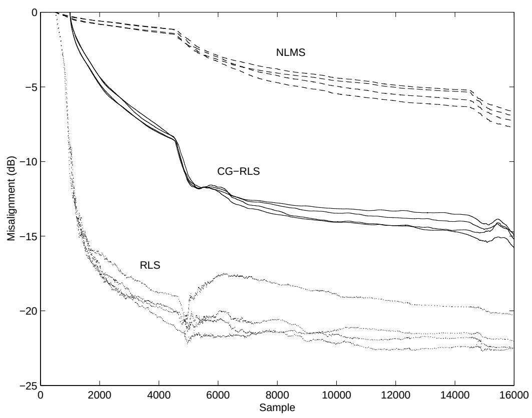

Figure 1: Misalignment for MAEC for the four different impulse responses using RLS (dotted), CG-RLS (solid) and NLMS (dashed).

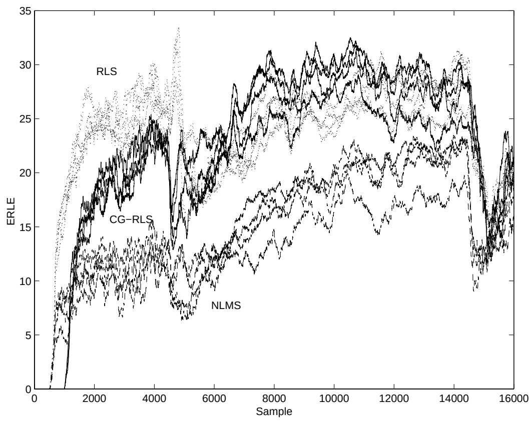

Figure 2: Echo cancellation performance for MAEC for the four different impulse responses using RLS (dotted), CG-RLS (solid) and NLMS (dashed).

by CG-RLS can be smaller than that for RLS. The filter estimates obtained by CG-RLS will, however, be much more noisy which is easily seen in the plot of the misalignment where the misalignment for RLS is much smaller than that for CG-RLS.

Another striking feature of the plots of Figure 2 is the "valleys" in the plots of the ERLE values. It might seem strange that the echo cancellation performance suddenly gets much worse at some points although the misalignment does not. This is, however, caused by the influence of the unmodeled tails of the true room impulse responses (remember that the estimated impulse responses are always shorter than the true impulse responses). When the power of the input signal (which consists of speech) abruptly decreases the influence of the unmodeled tail on the echo cancellation performance will be much larger than otherwise and there will be a reduction in the echo cancellation performance.

4. CONCLUDING REMARKS

In acoustic echo cancellation a fast adaptation of the filter estimates to the true room impulse response(s) is important. Perhaps even more important is, however, that the computational complexity and storage requirements for the acoustic echo canceller are kept low to make real-time implementation feasible. An algorithm that appears to fulfill these requirements is the CG-RLS algorithm that was introduced in [6], and applied to the MAEC and SAEC problems in this paper. The numerical results show that it performs much better than NLMS, which is one of the standard algorithms for AEC today. It performs, however, worse than RLS but as is shown the computational complexity and the storage requirements are much lower than those for RLS, which makes a real-time implementation for AEC feasible. This is hardly the case for RLS. CG-RLS is, however, much more computationally complex than NLMS but if that increase in computational complexity is affordable, CG-RLS could be the preferred choice.

5. REFERENCES

[1] Benesty J, Morgan DR, Sondhi MM. A better understanding and an improved solution to the specific problems of stereophonic acoustic echo cancellation. *IEEE Transactions on Speech and Audio Processing*, 6(2):156–165, Mar 1998.

[2] Breining C, Dreiseitel P, Hänsler E, Mader A, Nitsch B, Puder H, Schertler T, Schmidt G, Tilp J. Acoustic Echo Control - An Application of Very-High-Order Adaptive Filters. *IEEE Signal Processing Magazine*, 16(4):42–69, July 1999.

[3] Dreiseitel P, Hänsler E, Puder H. Acostic echo and noise control - a long lasting challenge. In *Proc. EUSIPCO-98, 9th European Signal Processing Conference*, Sept 1998.

[4] Golub G, Van Loan CF. *Matrix Computations*. The John Hopkins University Press, 3rd edition, 1996.

[5] Hänsler E. The hands-free problem - An annotated bibliography. *Signal Processing*, 3(27):259–271, 1992.

[6] Ng MK, Plemmons RJ. Fast Recursive Least Squares Adaptive Filtering by Fast Fourier Transform-based Conjugate Gradient Iterations. *SIAM Journal of Scientific Computing*, 17(4):920–941, July 1996.

[7] Stoica P, Åhgren P. Exact initialization of the recursive least-squares algorithm. *Preprint*, 2000.