Low-Complexity Adaptive Filtering Implementation for Acoustic Echo Cancellation

Christian Schüldt*, Fredric Lindstrom†, Ingvar Claesson*
*Blekinge Institute of Technology, Department of Signal Processing, SE-37225, Ronneby, Sweden
†Konfie AB, Research and Development, Box 268, SE-90106, Umeå, Sweden

Abstract—Acoustic echo cancellation is generally achieved with adaptive FIR filters. Due to the often large dimensionality of the adaptive filters, required to model rooms with standard reverberation time, the adaptation process can be computationally demanding. This paper presents a block based selective updating method which reduces the complexity with nearly a half in practical situations, while showing superior convergence speed performance as compared to conventional partial update complexity reduction schemes.

I. INTRODUCTION

Acoustic echo cancellation (AEC) [1] is used in teleconferencing equipment in order to provide high quality full-duplex communication. The core of an AEC solution is generally an adaptive filter which estimates the impulse response of the loudspeaker-enclosure-microphone (LEM) system. Typical adaptive algorithms for the AEC filter update procedure are the least mean square, normalized least mean square (LMS, NLMS), affine projection (AP) and recursive least squares (RLS) algorithms [2]. Of these, the NLMS based algorithms are particularity popular in industrial implementations thanks to their low complexity and finite precision robustness.

The echo cancellation environment can vary significantly, and in order to maintain acceptable echo reduction in rooms with long reverberation time, large dimensionality of the adaptive filters is required. Thus, the signal processing task can be computationally demanding.

Several partial update methods for computational complexity reduction of various adaptive filtering algorithms have been proposed and analyzed, e.g. [3], [4], [5], [6], [7], [8], [9], [10], [11] for the LMS/NLMS algorithms and [12], [6], [11] for AP. In this paper, a low complexity scheme applicable to both NLMS and AP, is presented and compared to several other complexity reduction methods.

II. COMPLEXITY REDUCTION METHODS

Commonly, the LEM system is modelled as a finite impulse response (FIR) model [1], \[ h = [h_0, \ldots, h_{N_L-1}]^T \], where \( N_L \) is the filter order. Filtering of the input signal \( x(k) \) then produces the desired (acoustic echo) signal \( d(k) \), given by \( d(k) = h^T x_L(k) \), where \( x_L(k) = [x(k), \ldots, x(k-N_L+1)]^T \) and \( k \) is the sample index. By adding near-end noise \( n(k) \), the measured (microphone) signal \( y(k) \) is obtained, \( y(k) = d(k) + n(k) \). The NLMS algorithm is then

\[
e(k) = y(k) - \hat{d}(k) = y(k) - x(k)^T h(k)
\]

The proposed method is then

\[
\hat{h}(k+1) = \hat{h}(k) + \mu \frac{e(k) x(k)}{|x(k)|^2 + \epsilon},
\]

where \( \hat{h}(k) = [\hat{h}_0(k), \ldots, \hat{h}_{N-1}(k)]^T \) is the adaptive FIR filter of length \( N \), \( x(k) = [x(k), \ldots, x(k-N+1)]^T \) the \( N \) length regressor vector, \( d(k) \) the estimated echo, \( e(k) \) the error (echo cancelled) signal, \( \mu \) is the step-size, and \( \epsilon \) is a positive constant to avoid division by zero or near-zero [2].

Several low-complexity methods which only perform a part of the filtering update, equation (2), have been proposed. Partial NLMS is performed by dividing the \( N \) filter coefficients into \( B \) blocks and only updating \( M \) of these blocks each sample. Which blocks to update can be selected in either a sequential manner [3], randomly [8] or by updating the parts which correspond to large energy of the regressor vector [6], [7]. For \( B = N \), i.e. the block-size set to 1 sample, the later method becomes the M-Max NLMS [4], [5]. In contrast, the periodic NLMS, updates all \( N \) filter coefficients but only at periodical sample intervals. The update can also be partitioned over all samples [3]. Other low-complexity updating schemes, possible in NLMS based algorithms where the step-size is zero for a large number of updates [10], [11], have been proposed as well as those based on assumptions of the unknown plant [9].

Complexity-reductions methods based on AP are for example the Fast Affine Projection (FAP) algorithm [12], which reduces the complexity almost to a NLMS level, except for a matrix inversion. Other methods are selective partial AP [6] and M-Max AP [13] (both analog to their NLMS counterpart).

III. THE PROPOSED ALGORITHM

The method proposed in this paper is similar to the periodic NLMS, but updates with as large error as possible, instead of updating the filter with input vectors obtained at fixed time instances. This is achieved by a buffering technique, where blocks of \( L \) samples are collected and processed. As a result, a delay of \( L \) samples is introduced in the signal path, but since \( L \) is relatively small (< 10), this normally has insignificant impact on the whole system. Similarity to the block LMS algorithm [2] is also apparent, with the difference being the number of updates per block (\( L \) for the block LMS and only one for the proposed method).

The \( L \) samples of one block are filtered and the corresponding output errors are calculated according to

\[
e(l+\text{i}) = y(l+\text{i}) - x(l+\text{i})^T h(l), \quad \text{i} = 0, \ldots, L-1
\]
where the block index $l$ is related to the original sample index $k$ and block length $L$ as

$$k = lL + i, \quad i = 0, \cdots, L - 1,$$

$$l = 1, 2, \cdots$$

(4)

The decision of what output error (and corresponding input vector) should be used for the update is determined by

$$i_l = \arg \max_{i \in \{0, \cdots, L-1\}} |e(lL + i)|,$$

(5)

and an update of the filter is then performed with

$$\hat{h}(l+1) = \hat{h}(l) + \mu e(lL + i_l)x(lL + i_l) / ||x(lL + i_l)||^2 + \epsilon.$$  

(6)

Since this structure updates the filter with the input vector resulting in the largest largest output error, small errors will generally be ignored. However, the resulting impact on the filter convergence is likely to be minor, since a small error occurs due to a well converged filter or due to orthogonality between the input vector and the filter mismatch vector. In both cases, a filter update will not result in any significant convergence.

**IV. COMPLEXITY AND IMPLEMENTATION**

The whole purpose of the proposed solutions is to reduce the complexity of an NLMS-implementation without sacrificing too much convergence speed. In a real application, the solution is generally realized through a Digital Signal Processor (DSP) which is capable of performing multiply-add-and-accumulate arithmetic instructions in parallel with memory reads and/or writes.

FIR filtering with a filter length $N$ typically requires $N$ DSP-instructions. Searching for the maximum absolute valued element in equation (5) all together requires $2L$ instructions. An NLMS-update, equation (6), requires $2N$ instructions. However, since equations (5) and (6) are only calculated once for ever block of $L$ samples, the number of instructions required per sample is $2(L + N)/L$. The included scalar product $x((L + i_l))T x((L + i_l))$ can be calculated very efficient through recursion [2] and its computational complexity can be disregarded in this case.

All together, this means that the number of DSP-instructions required for the proposed solution is $N + 2(L + N)/L$ in comparison with the complexity of a standard NLMS-update which is $3N$. For acoustic echo cancellation, $N$ is generally quite large (> 1000) due to room reverberation time. In this case we typically have $N \gg L$, which means that the complexity can be approximated as $N + 2N/L$. The complexity reduction in comparison with standard NLMS-updating is then

$$\frac{1}{3} + \frac{2}{3L},$$

(7)

which for $L = 4$ gives one half of the standard NLMS complexity.

**V. SIMULATIONS**

Comparisons between the proposed method denoted Max-E NLMS, M-Max NLMS, partial NLMS, random NLMS and standard NLMS were performed through simulations. The parameters for Max-E NLMS are shown in table I and the parameters for the other methods were chosen so that the complexity of the adaptation for each method was similar. However, for the M-Max NLMS, there is also additional complexity in the sorting of the input vectors ($2 \log_2 N + 2$ comparisons and $(N - 1)/2$ memory transfers). Furthermore, $N$ additional memory locations are also required [7].

The LEM system was modelled with a FIR filter $h$ of length

**TABLE I**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td>1024</td>
</tr>
<tr>
<td>$\mu$</td>
<td>0.95</td>
</tr>
<tr>
<td>$L$</td>
<td>4</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>4</td>
</tr>
</tbody>
</table>
The convergence improvement for speech signals is not reached. It has also been shown that considerable improvement (over a normal office. The acoustic coupling between the loudspeaker and the microphone was normalized to 0dB.

To obtain the microphone signal \( y(k) \), the loudspeaker signal \( x(k) \) was filtered with the measured impulse response \( h \) and band limited flat spectrum noise was added so that the resulting echo-to-noise ratio was approximately 28dB. The sampling rate was 8kHz.

The results of the simulations for a band limited flat spectrum input signal is presented in figure 1, where the filter deviation, calculated as \( \sum_{j=0}^{N-1} (h_j - \hat{h}_j(k)) \), where \( h_j \) is the \( j \)-th element of \( h \) and \( \hat{h}_j(k) \) the \( j \)-th element of the considered adaptive filter \( \hat{h}(k) \), respectively, is presented. Not surprisingly, the full NLMS has the fastest convergence speed of the solutions with similar complexity, compared to the other. Of the four variants with reduced complexity, the M-Max NLMS has slightly reduced convergence speed compared to full NLMS, followed by the proposed Max-E solution, random and partial NLMS.

Simulations were also performed with speech as input signal. However, an issue arises with the M-Max NLMS, since the step-size stability condition is tighter for this type of signal [14], and with the current setting M-Max NLMS diverges. Due to this, \( \mu \) for M-Max NLMS is set as high as possible, but still allowing convergence, which in this case is \( \mu = 0.5 \). Step-size settings for the other methods were unchanged. Figure 2 shows the results of the simulation. Here, the proposed solution has the fastest convergence speed of the solutions with similar complexity, while full NLMS has the fastest convergence of all.

It can also be noted that the random NLMS performs slightly better than partial NLMS for speech signals (while for band limited flat spectrum signals there is no difference), which agrees with the results in [8].

VI. FAST AFFINE PROJECTION-VERSION

The previously described scheme, where \( L \) samples are collected and the sample which produces the largest error is used for the filter update, can be applied to the fast affine projection (FAP) algorithm [12] as well. The procedure is highly similar, but since FAP is recursive and the consecutive updates are dependant, the implementation is somewhat less straightforward.

For the sake of simplicity (both implementation- and notation-wise), only FAP with the projection dimension 2 (FAP-2) is considered in this paper. Higher dimensions are possible, but perhaps not directly tractable due to the necessary \( D \times D \) matrix inversion, where \( D \) is the projection dimension. For \( D = 2 \), the matrix inversion can be performed directly. Moreover, it has also been shown that considerable improvement (over NLMS) is gained for just \( D = 2 \) and that further significant convergence improvement for speech signals is not reached until \( D \) is increased up to 10 [12], [1].

The original FAP [12] is modified analogous to the NLMS version presented earlier, where filtering is performed in blocks of \( L \) samples and the output and updating is

\[
e(lL + i) = y(lL + i) - (x(lL + i)\hat{h}(l) + \mu(z(i, i_{i-1})\phi(l-1))),
\]

\[
z(i, i_{i-1}) = x(lL + i)\hat{h}(l) + \mu z(i, i_{i-1}).
\]

\[
e(l) = \left(1 - \mu\right)e(lL + i)
\]

\[
X(l) = \left(\begin{array}{c}
|x(lL + i_i)|^2 & z(i_i, i_{i-1})
\end{array}\right)
\]

\[
\xi(l) = X^{-1}(l) - e(l).
\]

\[
\phi(l) = (\begin{array}{c}
0
\end{array}) + \xi(l)
\]

and

\[
\hat{h}(l + 1) = \hat{h}(l) + \mu\phi_1(l)x(l - 1L + i_{i-1}),
\]

where \( \phi(l-1) \) is the upper-most element of \( \phi(l-1) \), \( \phi_1(l) \) is the lower-most element of \( \phi(l) \) and \( l, L, i \) and \( i_1 \) are as defined in equations (4) and (5).

The complexity of the proposed FAP-2 solution is similar to that of the NLMS-version, with the difference being the correlation matrix \( X(l) \), which can be calculated recursively, and its inverse, which then can be calculated directly.

Simulations were also performed for the FAP-2 version of the proposed algorithm. Figure 3 shows the results for a band limited flat spectrum input signal, while figure 4 shows the results for a speech signal. Similar to the NLMS results, the superior convergence performance of the proposed algorithm can be noted, while random FAP-2 performs better than the periodic FAP-2, which is also similar to the results of the NLMS simulations.
VII. CONCLUSIONS

In acoustic echo cancellation environments, the computational resources required to update all adaptive filter coefficients every sample can be too costly. This paper has presented a block based solution which updates with the input vectors which produce the largest output error. The proposed solution significantly reduces the complexity of the adaptive filter, while still performing well in terms of convergence, compared to other partial update methods. As shown, this is specially true for speech signals.

ACKNOWLEDGMENT

The authors would like to thank the Swedish Knowledge Foundation (KKS) for funding.

REFERENCES