GENERALISED SUBBAND DECOMPOSITION WITH SELECTIVE PARTIAL UPDATES AND ITS APPLICATION TO ACOUSTIC ECHO CANCELLATION

*Kutluyıl Doğançay**

School of Electrical and Information Eng.
University of South Australia
Mawson Lakes, SA 5095, Australia
E-mail: Kutluyil.Dogancay@unisa.edu.au

Oğuz Tanrıkulu

Tellabs Media Processing Group
175 Middlesex Turnpike
Bedford, MA 01730, USA
E-mail: Oguz.Tanrikulu@tellabs.com

ABSTRACT

The paper presents a reduced-complexity implementation for the generalised subband decomposition LMS (GSD-LMS) algorithm based on selective partial updating. The major advantage of the GSD-LMS over the transform-domain LMS (TD-LMS) algorithm is the reduced transform size and the smaller number of divisions for power normalisation. The proposed algorithm capitalizes on this by further reducing the number of multiplications in the update term. The only overhead introduced is comparison operations for sorting which can be implemented cheaply. The superior performance of the selective-partial-update GSD-LMS compared with sequential block updating is demonstrated via simulation examples.

1. INTRODUCTION

Generalised subband decomposition (GSD) is a structural subband decomposition of FIR filters [1]. The resulting structure implements an adaptive FIR filter of length N as a parallel connection of L branches where each branch is composed of a fixed interpolator followed by a sparse adaptive FIR filter with at least N/L nonzero coefficients. The fixed interpolators can be implemented by transforms such as the discrete Fourier transform (DFT), discrete cosine transform (DCT) and discrete Hartley transform (DHT), to name but a few.

The LMS algorithm can be used to adapt the GSD coefficients, which results in a variant of the transform-domain LMS (TD-LMS) algorithm. For coloured input signals, the TD-LMS algorithm is known to have a significantly faster convergence speed than the normalised LMS (NLMS) algorithm. This improvement is achieved by decorrelating (whitening) the input regressor vector to the adaptive filter. Decorrelation is accompanied by power normalisation to ensure uniform convergence from any initialisation with the same distance from the solution vector. In acoustic echo cancellation applications, the adaptive filter usually requires a large number of coefficients ranging from 256 to 2048. The number of adaptive filter coefficients is affected to a large extent by the acoustic environment where the echo canceller is to employed. Inside a car, for instance, an adaptive FIR filter with approximately 200 coefficients is deemed sufficient. In a videoconference room, the required number of coefficients (the filter length) is usually larger. The computational complexity of the TD-LMS algorithm is proportional to the adaptive filter length. The root cause of large computational complexity is the number of multiplications that need to be performed to update the adaptive filter coefficients for every received data sample. The cost associated with the orthogonal transform and power normalisation also plays a significant role.

In this paper, we introduce a selective-partial-update GSD algorithm for acoustic echo cancellation. The proposed algorithm permits a tradeoff between performance and affordable computational complexity. The performance of the new reduced-complexity algorithm is illustrated with computer simulations for synthetic as well as real speech signals.

2. GENERALISED SUBBAND DECOMPOSITION

It is desirable to reduce the computational complexity of the TD-LMS algorithm while maintaining a better convergence performance than the ordinary NLMS. One approach to complexity reduction is to use a smaller size orthogonal transform than what would be required by the TD-LMS, and apply the transform outputs to adaptive FIR filters of appropriate length after power normalisation. The outputs of adaptive FIR filters are summed to obtain the adaptive filter output. This approach is called GSD. GSD reduces the complexity associated with orthogonal transform and the number of divisions required for power normalisation.

For an adaptive filter of length N with regressor vector $\boldsymbol{x}(k) = [x(k), x(k-1), \cdots, x(k-N+1)]^T$ and coefficient vector $\boldsymbol{w}(k) = [w_0(k), w_1(k), \cdots, w_{N-1}(k)]^T$,

* This work was partially supported by a DITEE Research Grant of the University of South Australia.

the TD-LMS algorithm is defined by the recursion [2]

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \mu\, e(k)\boldsymbol{\Lambda}^{-2}\boldsymbol{v}^*(k) \qquad (1)$$

where $\boldsymbol{v}(k) = \boldsymbol{T}\boldsymbol{x}(k)$ is the transformed regressor vector and

$$\boldsymbol{\Lambda}^2 = \begin{bmatrix} \sigma_0^2 & & & \\ & \sigma_1^2 & & \\ & & \ddots & \\ & & & \sigma_{N-1}^2 \end{bmatrix}$$

is the power matrix which is used to normalise the powers of approximately uncorrelated transform outputs $\boldsymbol{v}(k) = [v_0(k), \ldots, v_{N-1}(k)]^T$. The power matrix can be estimated online using a sliding exponential window:

$$\sigma_i^2(k) = \alpha\sigma_i^2(k-1) + (1-\alpha)|v_i(k)|^2, \qquad i = 0, \ldots, N-1$$

where $0.95 < \alpha < 1$ is the forgetting factor for the exponential window.

The transform $\boldsymbol{T}$ is a fixed $N \times N$ orthogonal matrix and can be obtained, e.g., from DFT, DCT or DHT. The computational complexity of the TD-LMS algorithm is N multiplications for calculation of $\boldsymbol{T}$ by bank of IIR filters, $3N$ multiplications and N divisions for power normalisation, N multiplications for calculation of $e(k)$, and N multiplications for calculation of the update term.

GSD uses an $M \times M$ orthogonal transform $\boldsymbol{T}$ where $M < N$. The transform outputs are applied to sparse FIR filters with K nonzero coefficients spaced by L:

$$W_n(z^L) = \sum_{i=0}^{K-1} w_{n,i}(k) z^{-iL}, \quad n = 0, 1, \ldots, M-1.$$

For $L \leq M$, the relationship between filter parameters is given by $N = (K-1)L + M$. The output of the adaptive filter is

$$y(k) = \sum_{i=0}^{K-1} \boldsymbol{v}^T(k - iL)\boldsymbol{w}_i(k)$$

where $\boldsymbol{w}_i(k)$ is the vector of ith coefficients of the filters $W_n(z^L)$:

$$\boldsymbol{w}_i(k) = [w_{0,i}(k), w_{1,i}(k), \ldots, w_{M-1,i}(k)]^T.$$

The error signal is

$$e(k) = d(k) - y(k)$$

where $d(k)$ is the desired filter response. The GSD-LMS algorithm is given by

$$\boldsymbol{w}_i(k+1) = \boldsymbol{w}_i(k) + \mu\, e(k)\boldsymbol{\Lambda}^{-2}\boldsymbol{v}^*(k - iL) \qquad i = 0, \ldots, K-1. \quad (2)$$

The complexity of the GSD-LMS is M multiplications for calculation of $\boldsymbol{T}$ by bank of IIR filters, $3M$ multiplications and M divisions for power normalisation, KM multiplications for calculation of $e(k)$, and KM multiplications for calculation of the update term.

Setting $M = N$, $K = 1$ and $L = 1$ reduces the GSD-LMS to the TD-LMS algorithm.

3. USING SELECTIVE PARTIAL UPDATES

The computational cost of GSD can be further reduced by employing selective partial updating [3]. First collect the regressor vectors and coefficient vectors together and then partition the new vectors to P blocks of length Q:

$$\boldsymbol{\nu}(k) = \begin{bmatrix} \boldsymbol{\nu}_1(k) \\ \boldsymbol{\nu}_2(k) \\ \vdots \\ \boldsymbol{\nu}_P(k) \end{bmatrix}_{PQ \times 1} = \begin{bmatrix} \boldsymbol{v}(k) \\ \boldsymbol{v}(k-L) \\ \vdots \\ \boldsymbol{v}(k-(K-1)L) \end{bmatrix}_{KM \times 1} \qquad (3)$$

$$\boldsymbol{\omega}(k) = \begin{bmatrix} \boldsymbol{\omega}_1(k) \\ \boldsymbol{\omega}_2(k) \\ \vdots \\ \boldsymbol{\omega}_P(k) \end{bmatrix}_{PQ \times 1} = \begin{bmatrix} \boldsymbol{w}_0(k) \\ \boldsymbol{w}_1(k) \\ \vdots \\ \boldsymbol{w}_{K-1}(k) \end{bmatrix}_{KM \times 1} . \qquad (4)$$

Partition the augmented power matrix in a similar way:

$$\boldsymbol{G}^2 = \begin{bmatrix} \boldsymbol{\Lambda}^2 & & \\ & \ddots & \\ & & \boldsymbol{\Lambda}^2 \end{bmatrix}_{KM \times KM} = \begin{bmatrix} \boldsymbol{G}_1^2 & & \\ & \ddots & \\ & & \boldsymbol{G}_P^2 \end{bmatrix}.$$

To achieve the desired complexity reduction, only some blocks of $\boldsymbol{\omega}(k)$ is updated. To select the blocks to be updated, we solve the following constrained optimisation problem:

$$\min_{1 \leq i \leq P} \min_{\boldsymbol{\omega}_i(k+1)} \|\boldsymbol{G}_i(\boldsymbol{\omega}_i(k+1) - \boldsymbol{\omega}_i(k))\|_2^2$$
$$\text{subject to} \quad \boldsymbol{\nu}^T(k)\boldsymbol{\omega}(k+1) = d(k).$$

This constrained optimisation problem is reminiscent of the one used in the derivation of the NLMS algorithm, but it has two main differences: (i) minimisation over coefficient vector blocks rather than the entire coefficient vector, and (ii) the weighting of coefficient update by the power matrix. The solution to this optimisation problem is obtained by the method of Lagrange multipliers. The cost function to be minimised is given by

$$J_i(k) = \|\boldsymbol{G}_i(\boldsymbol{\omega}_i(k+1) - \boldsymbol{\omega}_i(k))\|_2^2 + \lambda(d(k) - \boldsymbol{\nu}^T(k)\boldsymbol{\omega}(k+1))$$

where λ is a Lagrange multiplier. Solving

$$\frac{\partial J_i(k)}{\partial \boldsymbol{\omega}_i(k)} = \mathbf{0} \quad \text{and} \quad \frac{\partial J_i(k)}{\partial \lambda} = 0,$$

and introducing a small stepsize μ to control speed of convergence yields the recursion

$$\boldsymbol{\omega}_i(k+1) = \boldsymbol{\omega}_i(k) + \mu e(k) \frac{\boldsymbol{G}_i^{-2} \boldsymbol{\nu}_i^*(k)}{\boldsymbol{\nu}_i^H(k) \boldsymbol{G}_i^{-2} \boldsymbol{\nu}_i(k)}. \qquad (5)$$

The block to be updated is the one with the smallest squared Euclidean norm update weighted by the power matrix partition:

$$\begin{aligned} i &= \arg\min_{1 \le j \le P} \left\| e(k) \frac{\boldsymbol{G}_j^{-1} \boldsymbol{\nu}_j^*(k)}{\boldsymbol{\nu}_j^H(k) \boldsymbol{G}_j^{-2} \boldsymbol{\nu}_j(k)} \right\|_2^2 \\ &= \arg\max_{1 \le j \le P} \boldsymbol{\nu}_j^H(k) \boldsymbol{G}_j^{-2} \boldsymbol{\nu}_j(k). \end{aligned} \qquad (6)$$

In (5), the normalisation factor is redundant since the recursion is over approximately white input signals. Thus, (5) can be further simplified by dropping the normalisation factor. Incorporating the selection criterion in (6) into the simplified recursion yields

$$\begin{aligned} \boldsymbol{\omega}_i(k+1) &= \boldsymbol{\omega}_i(k) + \mu e(k) \boldsymbol{G}_i^{-2} \boldsymbol{\nu}_i^*(k), \\ & i = \arg\max_{1 \le j \le P} \boldsymbol{\nu}_j^H(k) \boldsymbol{G}_j^{-2} \boldsymbol{\nu}_j(k). \end{aligned} \qquad (7)$$

The selective-partial-update algorithm in (7) can be generalised to multiple-block selection [3]:

$$\boldsymbol{\omega}_{\mathcal{I}_B}(k+1) = \boldsymbol{\omega}_{\mathcal{I}_B}(k) + \mu e(k) \boldsymbol{G}_{\mathcal{I}_B}^{-2} \boldsymbol{\nu}_{\mathcal{I}_B}^*(k) \qquad (8)$$

where $\mathcal{I}_B = \{i : \boldsymbol{\nu}_i^H(k) \boldsymbol{G}_i^{-2} \boldsymbol{\nu}_i(k)$ is one of the B largest in $\boldsymbol{\nu}_1^H(k) \boldsymbol{G}_1^{-2} \boldsymbol{\nu}_1(k), \cdots, \boldsymbol{\nu}_P^H(k) \boldsymbol{G}_P^{-2} \boldsymbol{\nu}_P(k)\}$ is the set of B blocks to be selected. We will refer to (8) as the *selective-partial-update generalised subband decomposition LMS (SPU-GSD-LMS) algorithm.* The subscript $\mathcal{I}_B$ in, e.g., $\boldsymbol{\omega}_{\mathcal{I}_B}$ denotes an augmented vector composed of blocks with indices in $\mathcal{I}_B$, i.e.,

$$\boldsymbol{\omega}_{\mathcal{I}_B}(k) = [\boldsymbol{\omega}_{i_1}^T(k), \boldsymbol{\omega}_{i_2}^T(k), \cdots, \boldsymbol{\omega}_{i_B}^T(k)]^T$$

where $\mathcal{I}_B = \{i_1, i_2, \cdots, i_B\}$. Note that setting $P = KM$ and $B = P$ corresponds to the full-update GSD-LMS algorithm in (2).

4. SEQUENTIAL BLOCK UPDATES

For comparison purposes, we will consider the sequential-block GSD-LMS (SB-GSD-LMS) algorithm. While the SPU-GSD-LMS allows for significant reduction in the number of multiplications required for calculation of the update term, it also introduces some overhead for sorting of the P blocks. A zero overhead alternative to the SPU-GSD-LMS is the SB-GSD-LMS algorithm which uses the concept of sequential block updating [4].

Unlike the SPU-GSD-LMS, the SB-GSD-LMS does not select the blocks to be updated in an "intelligent" way. It simply updates one block in a sequential manner. The regressor and coefficient vectors are partitioned into P blocks as in (3) and (4). The SB-GSD-LMS uses the same recursion in (5). The block index i is changed with the recursion index k in a circular fashion:

$$i = (k \bmod P) + 1.$$

Thus the selection of the block to be updated requires no sorting. As we will see in the computer simulations, the convergence speed of the SB-GSD-LMS is $1/P$th that of the GSD-LMS. This reduction in convergence speed is often not tolerable, in particular for large P. Although the SB-GSD-LMS has zero overhead for block selection, its convergence speed is inferior to the SPU-GSD-LMS.

If $B \neq 1$ for the SPU-GSD-LMS, the equivalent SB-GSD-LMS will use P/B blocks of length BQ (assuming that the division results in an integer).

5. COMPUTATIONAL COMPLEXITY

The SPU-GSD-LMS algorithm requires M multiplications for the transform, M divisions and $3M$ multiplications for power normalisation, KM multiplications for $e(k)$, and $B(KM)/P$ multiplications for the update term where $B/P \le 1$. The SPU concept reduces the complexity for the update term by a factor of P/B. The overhead required for the selection of B blocks is $O(2M \log_2(KM) + 2M)$ comparisons for $P = KM$ (i.e., $Q = 1$). Here the Sortline algorithm [5] is assumed to be used for sorting the blocks using a heap of KM numbers. Every time a new sample comes in, M elements of the heap are replaced by new transformed and power normalised M numbers. This block-shift property of the heap can be exploited by the Sortline algorithm.

6. SIMULATIONS

We demonstrate the SPU-GSD-LMS in an acoustic echo cancellation application. The loudspeaker signal $x(k)$ is a stationary USASI signal with a speech-like spectrum, and the acoustic echo path used is a measured car echo impulse response of length 256 (see Fig. 1). The output of the echo path is corrupted by additive white Gaussian noise with zero mean. The approximate signal-to-noise ratio of the echo signal is 30dB.

In the simulations, the GSD parameters were $M = 10$, $K = 25$ and $L = 9$. The equivalent FIR filter length is $N = (K-1)L + M = 226$ [1]. If $L < M$, the equivalent length N of the GSD structure is smaller than the number of parameters MK. However, the performance of the GSD improves significantly if L is chosen less than M. The DCT was used for transforming the

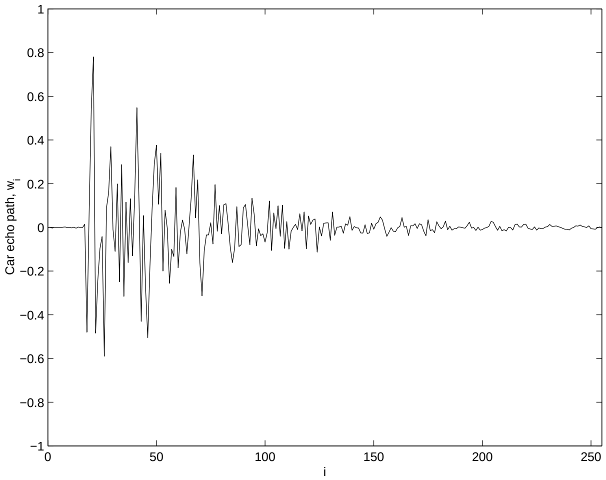

Figure 1: Car echo path.

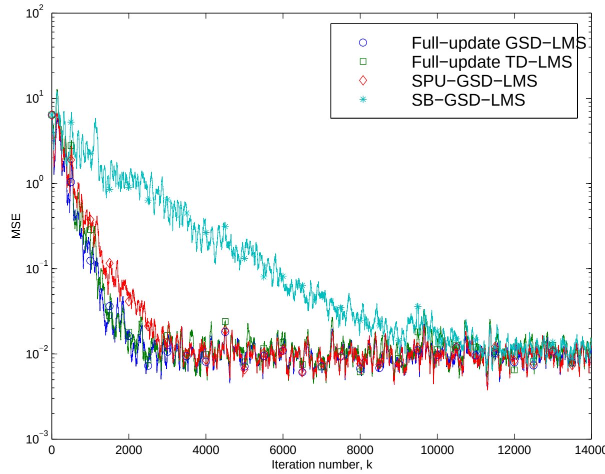

Figure 2: Convergence curves for the TD-LMS, GSD-LMS, SPU-GSD-LMS ($P/B = 5$) and SB-GSD-LMS.

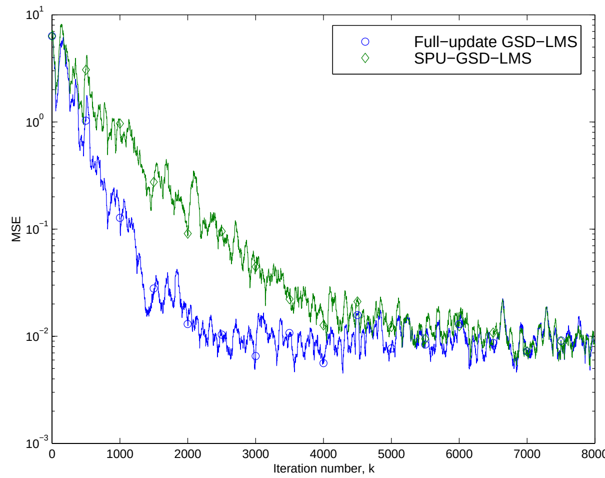

Figure 3: Convergence curves for the GSD-LMS and SPU-GSD-LMS ($P/B = 10$).

time-domain loudspeaker signal. Selective partial updating parameters were set to $P = MK = 250$ (i.e., $Q = 1$) and $B = 50$, i.e., 1/5th of the filter coefficients are updated per iteration. Fig. 2 shows the convergence curves for the TD-LMS ($N = 226$), the full-update GSD-LMS, the SPU-GSD-LMS and the SB-GSD-LMS. The stepsizes of the algorithms were set equal to 0.002. The SB-GSD-LMS has $P/B = 5$ blocks of length $BQ = 50$. As is evident from Fig. 2, the convergence speed of the SB-GSD-LMS is 1/5th that of the GSD-LMS. Note that the TD-LMS and GSD-LMS have almost identical convergence speeds and that the SPU-GSD-LMS has a comparable convergence to the GSD-LMS. Fig. 3 shows the convergence curves for the full-update GSD-LMS and the SPU-GSD-LMS with $P = 250$ and $B = 25$, i.e., 1/10th of the GSD coefficients are updated at every iteration.

7. CONCLUSION

We have developed a reduced-complexity implementation for the GSD-LMS algorithm. Unlike sequential block updating, selective partial updating does not severely penalize the convergence performance of the GSD-LMS. However, some overhead is introduced by selective partial updating. This overhead can be significant for DSP implementations unless an efficient algorithm such as the sortline algorithm is used. For VLSI implementations, the overhead for sorting is not significant.

8. REFERENCES

[1] M. R. Petraglia and S. K. Mitra, "Adaptive FIR filter structure based on the generalized subband decomposition of FIR filters," *IEEE Trans. on Circuits and Systems–II*, vol. 40, no. 6, pp. 354–362, June 1993.

[2] W. K. Jenkins, A. W. Hull, J. C. Strait, B. A. Schnaufer, and X. Li, *Advanced Concepts in Adaptive Signal Processing*, Kluwer, Boston, 1996.

[3] K. Doğançay and O. Tanrıkulu, "Selective-partial-update NLMS and affine projection algorithms for acoustic echo cancellation," in *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, ICASSP 2000*, Istanbul, Turkey, June 2000, vol. 1, pp. 448–451.

[4] S. M. Kuo and J. Chen, "Multiple microphone acoustic echo cancellation system with the partial adaptive process," *Digital Signal Processing*, vol. 3, pp. 54–63, 1993.

[5] I. Pitas, "Fast algorithms for running ordering and max/min calculation," *IEEE Trans. on Circuits and Systems*, vol. 36, no. 6, pp. 795–804, June 1989.