**Proceedings of the Institute of Acoustics**

# SPARSELY UPDATED FILTERS FOR ADAPTIVE DIGITAL PROCESSING OF AUDIO SIGNALS

Felipe Orduña-Bustamante (1), Philip A. Nelson (1) and Hareo Hamada (2)

(1) Institute of Sound and Vibration Research, The University, Southampton, England, SO9 5NH.
(2) Department of Electrical Communication Engineering, Tokyo Denki University, Tokyo 101, Japan.

## ABSTRACT

A sparse update strategy is presented that allows the implementation of adaptive filters at high sampling rates using existing DSP technology. This technique has the important property that the processing time per sampling period spent in filter update operations is independent of the filter length. The paper presents code for both the LMS and the filtered-$x$ LMS algorithms and demonstrates their practical use for loudspeaker equalization.

## 1. INTRODUCTION

The single-channel system identification problem is shown in Figure (1a). A vector containing the $L$ most recent samples of the input signal is defined as

$$\mathbf{x}_n = \begin{bmatrix} x_n & x_{n-1} & \cdots & x_{n-L+1} \end{bmatrix}^T;$$

similarly, a vector containing the coefficients of a (non-stationary) non-recursive digital filter is defined as

$$\mathbf{g}_n = \begin{bmatrix} g_0(n) & g_1(n) & \cdots & g_{L-1}(n) \end{bmatrix}^T.$$

The dot product of the input vector with the coefficient vector produces the output

$$y_n = \mathbf{g}_n^T \mathbf{x}_n.$$

This is required to approximate the desired signal $d_n$ (the output from the system under identification), in a way that minimizes the variance of the error signal defined by

$$e_n = d_n - y_n.$$

SPARSELY UPDATED FILTERS FOR ADAPTIVE DIGITAL AUDIO

The LMS algorithm [1] performs a stochastic gradient search in $L$-dimensions according to the formula

$$\mathbf{g}_{n+1} = \mathbf{g}_n + \mu e_n \mathbf{x}_n$$

which can be shown to converge on the mean to the exact least mean squares solution of the problem provided the adaptation rate is chosen to satisfy the following condition

$$\mu < 1 / L\sigma_x^2$$

The implementation of this algorithm in a microprocessor requires $4L$ arithmetic operations ($2L$ to perform the filtering and $2L$ to update the filter). At high sampling rates this can prove very taxing and heavy restrictions must be imposed on the number of coefficients.

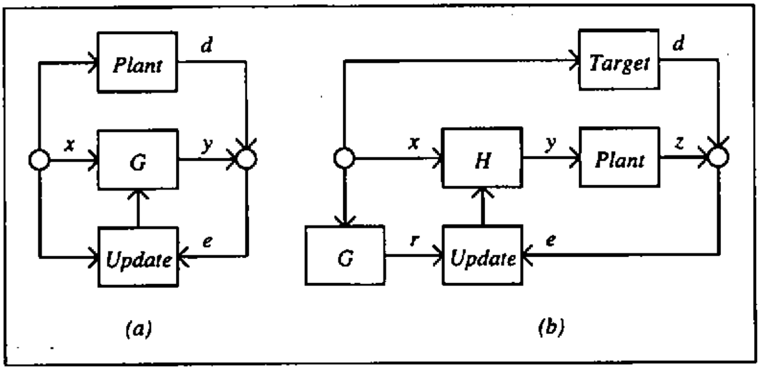


Figure 1. (a) Block diagram for the adaptive system identification problem. The filter $G$ must be adapted so that its output approximates the output from the plant. (b) Block diagram for the adaptive system equalization problem. The filter $H$ must be adapted so that the output from the plant approximates the output of a given target system which, in general, includes a modelling delay that ensures the existence of a causal optimal filter.

The system equalization problem is shown in Figure (1b). In this case, a new filter with coefficients

$$\mathbf{h}_n = \left[h_0(n) \quad h_1(n) \quad \dots \quad h_{L-1}(n)\right]^T$$

SPARSELY UPDATED FILTERS FOR ADAPTIVE DIGITAL AUDIO

acts on the input signal to produce the signal

$$y_n = \mathbf{h}_n^T \mathbf{x}_n$$

which after transmission through the system produces an output $z_n$ that minimizes the variance of a new error signal defined as

$$e_n = d_n - z_n.$$

The coefficients of the optimal filter can be searched using the filtered-$x$ LMS algorithm [1]

$$\mathbf{h}_{n+1} = \mathbf{h}_n + \mu e_n \mathbf{r}_n;$$

where the vector $\mathbf{r}_n$, containing recent samples of the reference signal

$$r_n = \mathbf{g}^T \mathbf{x}_n,$$

is used instead of the vector of input signals as in the LMS algorithm. This increases the arithmetic work by another $2L$ operations (assuming that both filters $\mathbf{g}$ and $\mathbf{h}$ have the same length). The arithmetic work thus increases to $6L$ operations.

## 2. SPARSE UPDATE IMPLEMENTATION OF THE LMS ALGORITHM

One way of reducing the operation count required to update the filter coefficients using the LMS algorithm is to devise a criterion to select the absolute minimum number of operations that are still necessary to maintain the convergence properties of the algorithm. The minimum work that can be done is to update only one filter coefficient per sampling period. This can be implemented by performing the following operations at every processing cycle $n$

$$g_k(n+1) = g_k(n) + \mu e_n x_{n-k}, \quad \text{(update current filter tap)}$$
$$k = (k+1) \bmod L; \quad \text{(increment tap counter)}$$

where $k$ is a counter set initially to $k$=0 that runs circularly along the vector of filter coefficients. Note that because $n$ and $k$ are always incremented by 1 every cycle (except when $k$ wraps around to zero), the input sample that is used for the update $x_{k-n}$ is exactly the same during the $L$ cycles that it takes to perform one pass along the whole filter. This observation leads to the following alternative, but exactly equivalent, version of the sparse update version of the LMS algorithm

SPARSELY UPDATED FILTERS FOR ADAPTIVE DIGITAL AUDIO

| | |
|---|---|
| if $k = 0$ then $\alpha = \mu x_n$ | (store, and pre-multiply, input sample) |
| $g_k(n+1) = g_k(n) + \alpha e_n$ | (update current filter tap) |
| $k = (k+1) \bmod L$ | (increment tap counter) |

By following any of these procedures, it takes $L$ processing cycles to update the whole filter, but the operation count is reduced to $2L$ (basically those involved in the actual filtering).

## 3. SPARSE UPDATE IMPLEMENTATION OF THE FILTERED-X LMS ALGORITHM

A sparse update implementation of the filtered-$x$ LMS algorithm presents the additional challenge of having to calculate the filtered reference signal in a way which is compatible with the sparse update of the filter coefficients. Interestingly enough the calculation of the filtered reference signal can be performed also on a single-tap basis as follows

| | |
|---|---|
| if $k = 0$ then $r = 0$ | (clear accumulator for filtered-$x$) |
| $r = r + g_{L-k-1} x_n$ | (accumulate current product) |
| $k = (k+1) \bmod L$ | (increment tap counter) |
| if $k = 0$ then $\alpha = \mu r$ | (store and pre-multiply filtered-$x$) |
| $h_k(n+1) = h_k(n) + \alpha e_n$ | (update current filter tap) |

Note that the calculation of the next sample of the filtered reference signal starts $L$ cycles in advance. To this end, the coefficients of the reference filter **g** are accessed in reverse, as shown, and the filtering makes use of the most recent input sample at every cycle. Also, performing the tap counter increment in between the filtered-$x$ accumulation and the filter update ensures that both processes are correctly synchronized. The operation count reduces again to just over $2L$ as in the sparse update implementation of the LMS algorithm.

## 4. LOUDSPEAKER EQUALIZATION USING SPARSE UPDATE ADAPTIVE FILTERS

Figure (2) shows the frequency response function of a loudspeaker in an anechoic chamber equalized to obtain a flat magnitude response and a linear phase response. The processing was performed in floating point arithmetic using a Texas Instruments TMS320C30 processor. The sampling frequency was set to $f$=32 kHz and the filter length to $L$=48. The impulse response function of the system was first identified using the sparse update version of the LMS algorithm. The system was later equalized using the sparse update implementation of the filtered-$x$ LMS algorithm. (A full update could only be possible at $f$=16kHz or $L$=24.)

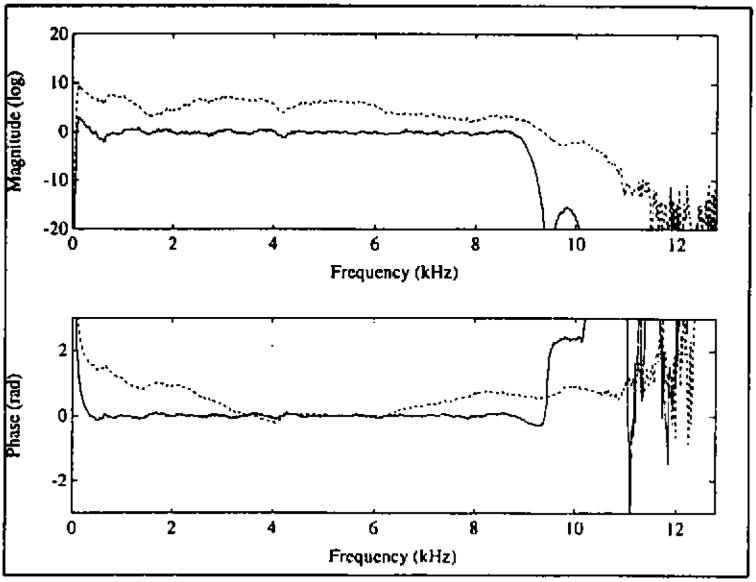


Figure 2. Frequency response function of a loudspeaker in an anechoic chamber: unequalized (dashed line) and equalized using the sparse update adaptive algorithms (solid line).

## 5. CONCLUSIONS

The sparse update implementation reduces the operation count by a factor of 2 in the case of the LMS algorithm and by a factor of 3 in the case of the filtered-$x$ LMS algorithm. More importantly, the bulk of the computational work that remains to be done is mainly related to the actual filtering operation. These calculations can be performed by a dedicated filtering unit external to the main processor and then most of the processing time can be dedicated to update the filter. This technique can be easily extended to complex systems based on a number of adaptive filters having arbitrary lengths. This is an ability which is highly desirable in multi-channel applications [2][3].

## 6. REFERENCES

[1]. B. WIDROW and S. D. STEARNS, Adaptive Signal Processing, Prentice-Hall, Englewood Cliffs, New Jersey, USA, (1985).

[2]. S. J. ELLIOTT, I. M. STOTHERS and P. A. NELSON, "A multiple error LMS algorithm

and its application to the active control of sound and vibration", *IEEE Trans. Acoust., Speech Signal Processing*, **35** pp. 1423-1434 (1987).

[3]. P. A. NELSON, H. HAMADA and S. J. ELLIOTT, "Adaptive inverse filters for stereophonic sound reproduction", *IEEE Transaction on Signal Processing*, **40**(7) (1992).

ISBN 1 873082 54 1