ADAPTIVE BEAMFORMING WITH PARTITIONED FREQUENCY-DOMAIN FILTERS

Marcel Joho, George S. Moschytz

Signal and Information Processing Laboratory,
Swiss Federal Institute of Technology Zürich, Switzerland

joho@isi.ee.ethz.ch, tel.:+41 (0) 1 632 27 71, moschytz@isi.ee.ethz.ch, tel.:+41 (0) 1 632 27 63

ABSTRACT

In this paper an adaptive broadband beamformer is presented which is based on a partitioned frequency-domain least-mean-square algorithm (PFDLMS). This block algorithm is known for its efficient computation and fast convergence even when the input signals are correlated. In applications where long filters are required but only a small processing delay is allowed, a frequency domain adaptive beamformer without partitioning demands a large FFT length despite the small block size. The FFT length can be shortened significantly by filter partitioning, without increasing the number of FFT operations. The weaker requirement on the FFT size makes the algorithm attractive for acoustical applications.

1 INTRODUCTION

Adaptive broadband beamforming is becoming increasingly important in acoustical applications for spatial filtering [1]. It can be implemented in the time-domain when short filters meet the desired performance. However, in acoustical applications related to hearing aids or echo canceling, where a few hundred filter coefficients may be required to achieve the desired performance, the computational complexity, expressed in real multiplications, grows quadratically with the filter length. A second drawback of a time-domain realization is the poor adaptation speed and tracking ability due to the large eigenvalue disparity typically arising in such a system.

These two problems can be circumvented by using a frequency-domain LMS (FDLMS) algorithm where both filtering and adaptation are carried out in the frequency-domain [2]. The linear convolution (filtering) in the time-domain is derived from a cyclic convolution using the *overlap-save* method for fast filtering. This can be implemented efficiently in the frequency-domain using fast Fourier techniques [3].

As is known, the DFT generates signals that are approximately uncorrelated (orthogonal). As a result, the update of the filter weights can be decoupled in the frequency-domain by using a normalized LMS algorithm (NLMS) in every frequency bin, each with only one coefficient. This leads to a more uniform convergence rate of the adaptive filter [4].

The greatest disadvantage of a frequency-domain implementation is the long processing delay caused by the block-wise execution of the algorithm. In fact, this delay can be lowered by increasing the overlap between two successive input blocks, but this decreases the effectiveness of the overlap-save method. For applications where this is unacceptable, *partitioned* frequency-domain adaptive filters provide a way out of the problem [5].

2 ADAPTIVE BEAMFORMING

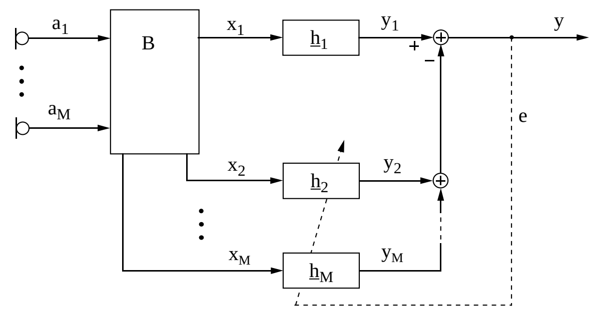

Figure 1: Griffiths-Jim beamformer

2.1 Griffiths-Jim Beamformer

The system studied here is based on the adaptive beamformer described by Griffiths and Jim [6]. As seen in Fig. 1 the array consists of M sensors. Its input signals a_m are transformed with a matrix B into a *main channel* x_1 and $M-1$ *auxiliary channels* $x_2 .. x_M$. B is an $M \times M$ matrix organized such that $x_m(t) = \sum_{i=1}^{M} b_{m,i} \cdot a_i(t)$ or, in vector notation, $\underline{x}(t) = B\,\underline{a}(t)$, where $\underline{a}(t) = [a_1(t), \ldots, a_M(t)]^T$ and $\underline{x}(t) = [x_1(t), \ldots, x_M(t)]^T$.

The matrix B is set up such that it prevents a *target* (in form of a plane wave impinging perpendicularly on the array), from passing through to the auxiliary channels, while letting it pass unimpeded through to the main channel ($x_m(t) = 0$ for $m = 2..M$ if $a_i(t) = a(t)$ for $i = 1..M$). One possible realization of B is (as described in [6] for $M = 3$):

$$B = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix} . \tag{1}$$

2.2 Time-Domain Filtering and Adaptation

Each channel x_m of the beamformer contains an FIR-filter $\underline{h}_m$. The filter in the main channel, $\underline{h}_1$, is assumed to be time-invariant and designed to shape the target spectrum. $\underline{h}_2 .. \underline{h}_M$ are adaptive filters updated after every time sample such as to minimize the power of the beamformer output:

$$y(t) = y_1(t) - \sum_{m=2}^{M} y_m(t) . \tag{2}$$

If perfect adaptation occurs, only the target signal filtered with $\underline{h}_1(t)$ remains at the beamformer output $y(t)$, whereas signal components from other spatial directions (*jammers*) vanish.

For the adaptation it is assumed that target and jammers are uncorrelated, therefore the beamformer output is taken as the adaptation error as well: $y(t) = e(t)$. The filter update equations for real-valued input signals are:

$$\underline{h}_m(t+1) = \underline{h}_m(t) + \mu_0 \underline{x}_m(t) e(t) \quad (3)$$
$$\underline{x}_m(t) = [x_m(t), \ldots, x_m(t-N+1)]^T , \quad (4)$$

where μ_0 is the step-size controlling the rate of convergence and stability of the adaptation.

2.3 Frequency-Domain Filtering

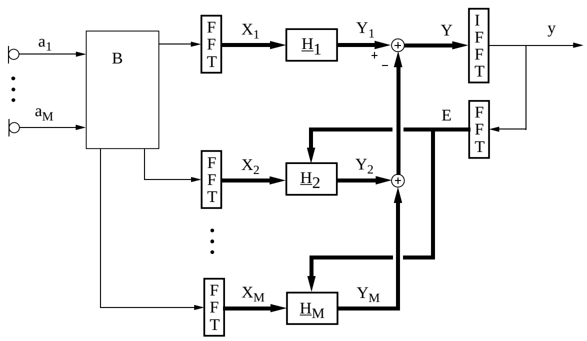

Figure 2: Frequency-domain realization.

The time-domain adaptive beamformer structure shown in Fig. 1 can, by analogy, be implemented in the frequency-domain (FDAB), as shown in Fig. 2. A major difference is that the processing is now done block-wise and not sample by sample. The new block input vector of channel m is defined as

$$\underline{x}_m[k] = [x_m(kL), \ldots, x_m(kL+L-1)]^T , \quad (5)$$

where (.) and [.] denote the sample and block index, respectively ($[.] \triangleq (.\cdot L)$). L is the block length and N is the filter length (number of taps). The overlap-save method requires that the FFT length C fulfills the condition

$$C \geq N + L - 1 \quad (6)$$

so that L new output samples per block can be derived which coincide with those of the appropriate linear convolution in the time-domain. For the sake of simplicity it is assumed, throughout this paper, that the filter length is a multiple of the block length ($N = SL$) and the FFT length C is chosen to be $N+L$. Therefore, to obtain the C-point DFT $\underline{X}_m$ the last $S+1$ block input vectors are needed:

$$\underline{X}_m[k] = \mathbf{F} \begin{pmatrix} \underline{x}_m[k-S] \\ \vdots \\ \underline{x}_m[k] \end{pmatrix} . \quad (7)$$

$\mathbf{F}$ is the Fourier matrix ($(\mathbf{F})_{a,b} = \exp(-j\frac{2\pi}{C}ab)$ for $a,b = 0..C-1$) and $j = \sqrt{-1}$. The filter output of channel m is

$$\underline{Y}_m[k] = \underline{H}_m[k] \odot \underline{X}_m[k] , \quad (8)$$

where $\odot$ denotes the element-wise multiplication of two vectors. $\underline{H}_m[k]$ is the frequency-domain weight vector. Finally, the beamformer output in the frequency-domain is, by analogy with (2), the main channel output minus the auxiliary channel outputs:

$$\underline{Y}[k] = \underline{Y}_1[k] - \sum_{m=2}^{M} \underline{Y}_m[k] . \quad (9)$$

The time-domain block output vector $\underline{y}[k] = [y(kL), \ldots, y(kL+L-1)]^T$ is obtained with the overlap-save method by taking the last L elements of the inverse DFT of $\underline{Y}[k]$

$$\underline{y}[k] = \mathbf{P}_{\underline{y}} \cdot \mathbf{F}^{-1} \cdot \underline{Y}[k] , \quad (10)$$

where

$$\mathbf{P}_{\underline{y}} = \begin{bmatrix} \mathbf{0}^{L\times(C-L)} & \mathbf{I}^L \end{bmatrix}^{L\times C} \quad (11)$$

is the output projection matrix. $\mathbf{I}$ stands for the unity matrix and $\mathbf{0}$ is a matrix whose elements are all zero. The superscript indicates the matrix dimension. The output samples calculated with (10) correspond to those of a linear time-domain convolution if the filter coefficients are held constant over L samples. As can be seen from (7) and (10), $M+1$ FFT operations[1] are required for the filtering part.

2.4 Frequency-Domain Adaptation

Just as the filtering is carried out in the frequency-domain, the same is true of the adaptation. For this the error signal $\underline{e}[k] = \underline{y}[k]$ has to be transformed into the frequency-domain:

$$\underline{E}[k] = \mathbf{F} \begin{pmatrix} \underline{0}^{(C-L)} \\ \underline{e}[k] \end{pmatrix} . \quad (12)$$

Note that this additional FFT is required to obtain an error signal which, as opposed to $\underline{Y}[k]$, is free of a cyclic convolution contribution. Similar to (3) the same error signal is used for the update of the filter weights, therefore only one additional FFT is needed, regardless of the number of sensors M.

Note also that only the filter coefficients in the auxiliary channels are adapted, so the following update equations are valid only for $m = 2..M$. The incremental updates of the frequency-domain weight vectors are:

$$\triangle \underline{H}_m[k] = \underline{\mu}_m[k] \odot \underline{X}_m^*[k] \odot \underline{E}[k] , \quad (13)$$

where * denotes the complex conjugate. $\underline{X}_m^*[k] \odot \underline{E}[k]$ corresponds to a cyclic correlation between the input and the error sequence [2]. The step-size of the adaptation can be controlled in each frequency bin by $\underline{\mu}_m[k]$. The update equation for the adaptive coefficients is

$$\underline{H}_m[k+1] = \underline{H}_m[k] + \mathbf{P}_{\underline{H}} \cdot \triangle \underline{H}_m[k] , \quad (14)$$

where

$$\mathbf{P}_{\underline{H}} = \mathbf{F} \begin{bmatrix} \mathbf{I}^N & \mathbf{0} \\ \mathbf{0} & \mathbf{0}^{C-N} \end{bmatrix} \mathbf{F}^{-1} \quad (15)$$

sets the last $C-N$ elements of $\triangle \underline{h}_m[k] = \mathbf{F}^{-1} \triangle \underline{H}_m[k]$ to zero (*gradient constraint*). The overlap-save method requires this to ensure that each output vector $\underline{y}[k]$ is the counterpart of the corresponding linear convolution.

[1] FFT or IFFT

2.5 Power-normalized step-sizes

To achieve fast convergence, even for a large eigenvalue spread of the TD input autocorrelation matrix, the adaptation step-sizes are adjusted independently in each frequency bin. This works because of the convenient decorrelation property of the DFT [4], and is done by a bin-wise power-normalization of the step-sizes:

$$\underline{\mu}_m[k] = \mu_0 \cdot \underline{P}_{\underline{X}_m}^{((-1))}[k] \,, \tag{16}$$

where $\underline{P}_{\underline{X}_m}[k]$ is a vector whose elements contain a power estimate of the appropriate frequency bin. $(.)^{((-1))}$ denotes the element-wise inversion of a vector. μ_0 is a constant that determines the rate of convergence and the stability of the adaptive process. For the bin-wise power estimation an exponential forgetting is used:

$$\underline{P}_{\underline{X}_m}[k] = \lambda \cdot \underline{P}_{\underline{X}_m}[k-1] + (1-\lambda) \cdot (\underline{X}_m^*[k] \odot \underline{X}_m[k]) \,, \tag{17}$$

where λ is the forgetting factor ($0 < \lambda < 1$).

If the matrix B is chosen such that each auxiliary channel receives the same amount of input power, e.g. (1), then the power estimation (17) and step-size computation (16) have to be done only in one of the auxiliary channels. This leads to a great saving in the number of multiplications and inversions.

Note that if the input signals of the sensor array are not white, e.g. speech, then care has to be taken about the FFT size C. For a too small FFT size the decorrelation property of the FFT declines, which results in a poorer adaptation performance. As a rule of thumb the FFT size should not be smaller than the length of the correlation of the appearing signals.

3 PARTITIONED FILTERS

As seen from (7) and (10), for every block, M FFTs and one IFFT have to be computed for the filtering alone, which is quite a demanding task. When only a small processing delay is allowed (e.g. hearing aid), but large filters are needed ($N \gg L$), then from each block-operation only a small number of L new output samples are obtained compared to the large FFT length $C \gg L$. This discrepancy comes from (6).

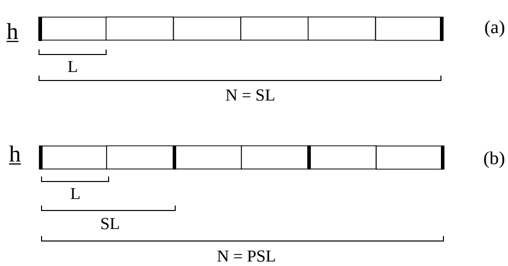

Figure 3: Partitioning of the filter impulse response: (a) no partitioning: P=1, S=6 (b) with partitioning: P=3, S=2.

The overlap is defined as the number of common input elements of two succeeding FFTs divided by the total number of input elements $N/(N+L)$. For $N \gg L$ the overlap is near unity, leading to a poor computational yield of the overlap-save method.

To circumvent such a large overlap, the filters are partitioned into P subfilters of length N/P [5]. The underlying idea is to bring down the length of the FFT. Because the filter length of each partition is N/P, the new constraint on the lower bound of the FFT size is now:

$$C \geq N/P + L - 1 \tag{18}$$

which is considerably smaller than (6) for $N \gg L$. Furthermore it is assumed in this paper that each partition can be subdivided into S segments of length L such that the total length of each filter is $N = PSL$. An example of filter partitioning is shown in Fig. 3.

3.1 Partitioned Filtering

The block diagram of the partitioned frequency-domain adaptive beamformer (PFDAB) is presented in Fig. 4. Both the filtering and the adaptation is carried out in the frequency-domain. Similarly to the FDAB, the input vectors for the first partitions ($p = 0$) are transformed with (7) into the frequency-domain, i.e. $\underline{X}_{m,0}[k] = \underline{X}_m[k]$, bearing in mind that the FFT length C and the number of segments S are now smaller. The input vectors of the other partitions ($p = 1..\,P-1$) are delayed versions of the first partitions of every channel:

$$\underline{X}_{m,p}[k] = \underline{X}_m\left[k - pS\right] . \tag{19}$$

There is a total of $P \cdot M$ partitions, each contributing

$$\underline{Y}_{m,p}[k] = \underline{H}_{m,p}[k] \odot \underline{X}_{m,p}[k] \tag{20}$$

to the beamformer output, which is, similarly to (9), the output of the main channel minus the auxiliary channels:

$$\underline{Y}[k] = \sum_{p=0}^{P-1} \left(\underline{Y}_{1,p}[k] - \sum_{m=2}^{M} \underline{Y}_{m,p}[k] \right) . \tag{21}$$

As for the FDAB, the time-domain output vector of the beamformer $\underline{y}[k]$ is obtained with (10) and (11). Although the FFT length is smaller with filter partitioning, the number of FFT operations remains the same.

3.2 Partitioned Adaptation

As in section 2.4, the filter in the main path (m=1) is assumed to be time-invariant, therefore the following update equations are only valid for $m = 2..\,M$ and $p = 0..\,P-1$. The adaptation error needed for the incremental update of the filter coefficients is again given by (12). Each filter partition is updated with the algorithm

$$\triangle \underline{H}_{m,p}[k] = \underline{\mu}_{m,p}[k] \odot \underline{X}_{m,p}^*[k] \odot \underline{E}[k] \tag{22}$$

$$\underline{H}_{m,p}[k+1] = \underline{H}_{m,p}[k] + \mathbf{P}_{\underline{H}} \cdot \triangle \underline{H}_{m,p}[k] \tag{23}$$

$$\mathbf{P}_{\underline{H}} = \mathbf{F} \begin{bmatrix} \mathbf{I}^{SL} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}^{C-SL} \end{bmatrix} \mathbf{F}^{-1} \tag{24}$$

which is the appropriate counterpart to the FDLMS (13), (14) and (15). Similar to (19) the step-size vector is conveyed from one partition to the next: $\underline{\mu}_{m,p}[k] = \underline{\mu}_m\left[k - pS\right]$ where $\underline{\mu}_m[k]$ is the same as in (16). Note that as long as the FFT length is not smaller than the correlation length of the input signals, the decorrelation property of the FFT is preserved and the adaptation behavior is unaffected.

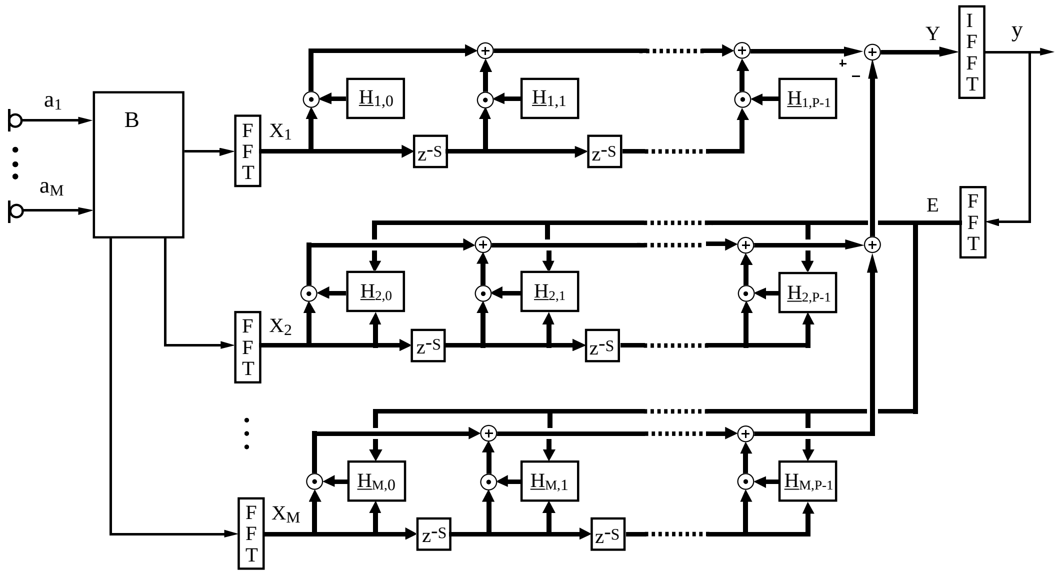

Figure 4: Block diagram of the partitioned frequency-domain adaptive beamformer. z^{-S} denotes a delay of S blocks.

As seen from (23), the gradient constraint (24) requires 2 FFT operations for each partition. This makes a total of $2P(M-1)+1$ FFT operations only for the adaptation, if (12) is also taken into account. Even for small partitioning this number becomes the dominant contribution in terms of FFT operations.

This can be drastically reduced with a slight modification of the coefficient update (23):

$$\underline{H}_{m,p}[k+1] = \begin{cases} \mathbf{P}_{\underline{H}} \cdot (\underline{H}_{m,p}[k] + \triangle\underline{H}_{m,p}[k]) \\ \qquad\qquad \text{if } p(m-2) = <k>_{P(M-1)} \\ \underline{H}_{m,p}[k] + \triangle\underline{H}_{m,p}[k] \qquad \text{else} \end{cases} \quad (25)$$

where $<a>_b = a - \left\lfloor \frac{a}{b} \right\rfloor \cdot b$ is the remainder when a is divided by the non-zero integer b.

In every block, one of the $P(M-1)$ adaptive filter partitions is picked out and constrained such that the last $C-SL$ elements of $\mathbf{F}^{-1}\underline{H}_{m,p}[k]$ are set to zero (*alternated weight constraint*). Therefore only 3 FFT operations remain for the adaptation part, independently of M and P, which is a substantial reduction. Different from (23), not only the incremental update $\triangle\underline{H}_{m,p}[k]$ is premultiplied with $\mathbf{P}_{\underline{H}}$, but the whole weight vector $\underline{H}_{m,p}[k]$.

Here it is assumed that the incremental update $\triangle\underline{H}_{m,p}[k]$ is much smaller than $\underline{H}_{m,p}[k]$ itself, and that if the last $C-SL$ elements of $\mathbf{F}^{-1}\underline{H}_{m,p}[k]$ are set to zero only every $P(M-1)$ blocks, they will not deviate much. Because the overlap-save constraint is violated between two such 'clearing operations', the filter output is disturbed by wrap-around effects of the cyclic convolution. This degradation can be controlled by chosing a smaller step-size μ_0.

4 SUMMARY

In this paper an efficiently implemented and fast converging adaptive beamformer is proposed which uses a partitioned frequency-domain LMS algorithm. Whenever long filters are needed but only a small processing delay is allowed, filter partitioning can lead to a substantial reduction of the minimum required FFT length, without increasing the number of FFT operations.

Furthermore an alternative filter update rule is presented based on an alternated weight constraint which achieves almost the same performance, but requires only two instead of $P(M-1)$ FFTs[2]. Finally, it is shown that the minimum total number of FFT operations amounts to $M+4$ for the filtering and adaptation. These properties make the algorithm very suitable for broadband spatial filtering in acoustical applications.

REFERENCES

1. J.M. Kates, "A comparison of hearing-aid array-processing techniques," J. Acoust. Soc. Amer., Vol. 99, 1996, pp. 3138–48.
2. J.J. Shynk, " Frequency-domain and multirate adaptive filtering," *IEEE SP Magazine*, Jan. 1992, pp. 14–37.
3. A.V. Oppenheim, R.W. Schafer *Discrete-time signal processing*, Prentice-Hall, Englewood Clifs, NJ, 1989.
4. D. Mansour, A.H. Gray, "Unconstrained frequency-domain adaptive filter," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, Oct. 1982, pp. 726–734.
5. P.C.W. Sommen, "Partitioned frequency-domain adaptive filters," *23rd Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, 1989, pp. 677–81.
6. L.J. Griffiths and C.W. Jim, "An alternative approach to linearly constrained adaptive beamforming," *IEEE Trans. Ant. Prop.*, AP-30, Jan. 1982, pp. 27–34.

[2] M and P are the number of input sensors and filter partitions, respectively.