Katholieke Universiteit Leuven

Departement Elektrotechniek

ESAT-SISTA/TR 2000-83

Iterated Partitioned Block Frequency–Domain Adaptive Filtering for Acoustic Echo Cancellation [1]

Koen Eneman, Marc Moonen [2]

Published in the IEEE Transactions on Speech and Audio Processing, vol. 11, no.2, pp. 143-158, March 2003

[1]This report is available by anonymous ftp from *ftp.esat.kuleuven.ac.be* in the directory *pub/sista/eneman/reports/00-83.ps.gz*

[2]ESAT (SCD) - Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, 3001 Leuven (Heverlee), Belgium, Tel. +32/16/321809, Fax +32/16/321970, WWW: *http://www.esat.kuleuven.ac.be/sista*. E-mail: *koen.eneman@esat.kuleuven.ac.be*. This research work was carried out at the ESAT laboratory of the Katholieke Universiteit Leuven, in the frame of the Belgian State Interuniversity Poles of Attraction Programme – IUAP P4–02 (1997–2001) : Modeling, Identification, Simulation and Control of Complex Systems, Concerted Research Action GOA–MEFISTO–666 of the Flemish Government, Research Project F.W.O. nr. G.0295.97 ('Design and implementation of adaptive digital signal processing algorithms for broadband applications') and IWT project AUT/970517: MUSETTE ('Multimicrophone Signal Enhancement Techniques for handsfree telephony and voice controlled systems'). The scientific responsibility is assumed by its authors.

Iterated Partitioned Block Frequency–Domain Adaptive Filtering for Acoustic Echo Cancellation

Koen Eneman* Marc Moonen
ESAT – Katholieke Universiteit Leuven
Kasteelpark Arenberg 10, B–3001 Leuven – Belgium
phone : +32/16321809 +32/16321060
fax : +32/16321970
email : koen.eneman@esat.kuleuven.ac.be
marc.moonen@esat.kuleuven.ac.be

Abstract

For high quality acoustic echo cancellation long echoes have to be suppressed. Classical LMS–based adaptive filters are not attractive as they are suboptimal from a computational point of view. Multirate adaptive filters such as the partitioned block frequency–domain adaptive filter (PBFDAF) are good alternatives and are widely used in commercial echo cancellers nowadays.
In this paper the PBFDRAP is analyzed, which combines frequency–domain adaptive filtering with so–called "row action projection". Fast versions of the algorithm will be derived and it will be shown that the PBFDRAP outperforms the PBFDAF in a realistic echo cancellation setup.

EDICS : 1–ENHA

I. Introduction

For high quality acoustic echo cancellation long echoes have to be suppressed. Acoustic echo paths are characterized by FIR filters with lengths up to 250 ms. Filters clocked at a rate of, say, 10 kHz then require several thousands of filter taps to be identified. A general acoustic echo cancellation setup is shown in figure 1, where d is a filtered version of x with an unknown acoustic path $w[k]$. In most applications a so–called local near–end talker signal s is added to $w \star x$ such that $d = s + w \star x$. The ultimate goal is to suppress $w \star x$ at the output e and to retain a non–distorted version of s.

Of all existing adaptive filtering algorithms the Least Mean Squares algorithm may be best known [1]. LMS–based algorithms have a complexity that is linear in the filter length, but they suffer from a rather slow convergence for signals with a colored spectrum such as speech. More efficient implementations of the LMS algorithm are available, often relying on frequency–domain techniques [2]. The so–called Partitioned Block Frequency–Domain Adaptive Filter (PBFDAF) [3] [4] [5] is used in commercial echo cancellers nowadays. In this paper the PBFDRAP will be discussed, which is a combination of the PBFDAF algorithm and so–called "row action projection" (RAP). Row Action Projection was initially proposed as an improvement to the LMS algorithm. A good reference on Row Action Projection is [6]. Fast versions of the PBFDRAP algorithm will be derived and it will be shown that the PBFDRAP can outperform the PBFDAF in a realistic echo cancellation setup.

The paper is organized as follows. In section II the PBFDAF adaptive filter is discussed. The PBFDRAP algorithm is presented in section III. In section IV the asymptotic properties of the PBFDRAP adaptive filter are investigated for different algorithmic settings. Fast implementation schemes are derived in section V. In section VI the different fast algorithmic variants are compared with the standard implementation of the PBFDRAP algorithm from a computational complexity point of view. Finally, some simulation examples are given in section VII.

Notation : upper bold face letters are used for matrices, lower bold face letters for column vectors; $(\cdot)^*$, $(\cdot)^T$ and $(\cdot)^H$ denote complex conjugation, transpose and Hermitian transpose, respectively; $\mathbf{I}_N$ and $\mathbf{0}_N$ denote the $N \times N$ identity and all–zero matrix; diag$\{\mathbf{v}\}$ is a diagonal matrix with vector $\mathbf{v}$ on its diagonal and $\mathbf{F}$ is the $M \times M$ DFT matrix with element (p, q) given by $e^{-j\frac{2\pi pq}{M}}$.

II. Partitioned Block Frequency–Domain Adaptive Filter

As a cheaper alternative to the LMS algorithm, the frequency–domain adaptive filter (FDAF) was introduced, which is a direct translation of Block–LMS to the frequency domain [2]. In-

stead of linear convolutions and correlations circular operations are performed in the frequency domain. This requires some 'restore' operations which are based on overlap–save or overlap–add techniques. The performance and convergence properties of the FDAF algorithm are comparable to those of the LMS algorithm. It appears that the FDAF algorithm is computationally attractive only if the block length has the same order of magnitude as the filter length. In practice however, this leads to unacceptable input/output delays : for a realistic echo cancellation setup with an adaptive filter $\hat{w}$, modelling 1000 taps in the time domain and sampled at 8 kHz for instance, the delay is twice as long as the acoustic impulse response, i.e. 250 ms.

By splitting the adaptive filter $\hat{w}$ (see figure 1) in equal parts and transforming to the frequency domain, a mixed time/frequency–domain adaptive filter can be obtained, called the Partitioned Block Frequency–Domain Adaptive Filter (PBFDAF) [3] [4] [5] : the N–taps fullband adaptive filter $\hat{w}^{(n)}[k]$ is partitioned in $\frac{N}{P}$ parts[1] $\hat{\boldsymbol{w}}_p^{(n)}$ of length P each, which are transformed to the frequency domain, as shown in Eqs. 1 and 2.

$$\hat{\boldsymbol{w}}_p^{(n)} \stackrel{\forall p}{=} \begin{bmatrix} \hat{w}^{(n)}[pP] \\ \vdots \\ \hat{w}^{(n)}[(p+1)P-1] \end{bmatrix}, \quad p = 0 : \frac{N}{P} - 1 \tag{1}$$

$$\mathbf{w}_p^{(n)} \stackrel{\forall p}{=} \mathbf{F} \begin{bmatrix} \hat{\boldsymbol{w}}_p^{(n)} \\ \mathbf{0} \end{bmatrix} \begin{matrix} \updownarrow P \\ \updownarrow M-P \end{matrix} \tag{2}$$

where $\mathbf{F}$ is the $M \times M$ DFT matrix. The equations defining the overlap–save PBFDAF are :

$$\mathbf{X}_p^{(n)} \stackrel{\forall p}{=} \text{diag} \left\{ \mathbf{F} \begin{bmatrix} x[(n+1)L - pP - M + 1] \\ \vdots \\ x[(n+1)L - pP] \end{bmatrix} \right\} \Bigg\updownarrow M, \quad p = 0 : \frac{N}{P} - 1 \tag{3}$$

$$\mathbf{y}^{(n)} = \begin{bmatrix} \mathbf{0}_{M-L} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{N}{P}-1} \mathbf{X}_p^{(n)} \mathbf{w}_p^{(n)} \tag{4}$$

$$\mathbf{d}^{(n)} = \begin{bmatrix} \mathbf{0} \\ \mathbf{d}_n \end{bmatrix} \begin{matrix} \updownarrow M-L \\ \updownarrow L \end{matrix}, \qquad \mathbf{d}_n = \begin{bmatrix} d[nL+1] \\ \vdots \\ d[(n+1)L] \end{bmatrix} \Bigg\updownarrow L \tag{5}$$

$$\mathbf{e}^{(n)} = \mathbf{d}^{(n)} - \mathbf{y}^{(n)} \tag{6}$$

$$\mathbf{w}_p^{(n+1)} \stackrel{\forall p}{=} \mathbf{w}_p^{(n)} + \mathbf{G}\boldsymbol{\Delta}\mathbf{X}_p^{(n)*}\mathbf{F}\mathbf{e}^{(n)}, \quad p = 0 : \frac{N}{P} - 1 \tag{7}$$

Each time an iteration of the PBFDAF is performed, as described by Eqs. 3–7, L new x–samples are taken in, and L new filter output samples e are produced. Vector $\mathbf{d}_n$ contains the L most

[1] It is assumed that $\frac{N}{P}$ is integer.

recent values of signal d. Parameter n is the block time index. L is called the block length, and hence the corresponding input/output delay of the algorithm is $2L-1$. It is required that $M \geqslant P+L-1$ to ensure proper operation. Finally, matrix $\mathbf{\Delta} = \text{diag}\{\boldsymbol{\mu}_s^{(n)}\}$ contains subband dependent stepsizes, controlling the adaptation speed.

If P is divisible by L, as is typically the case in practice, $\mathbf{X}_p^{(n)} = \mathbf{X}_0^{(n-pP/L)}$ and hence Eq. 3 requires only 1 DFT operation, namely the one that corresponds to $p=0$. The other $\mathbf{X}_p^{(n)}$ can be recovered from previous iterations. In most practical applications $P=L$ and M is a power of 2. The PBFDAF algorithm reduces to the FDAF if $N=P$.

There exist two variants of the algorithm, called the constrained and the unconstrained PBFDAF [2] [4] [5]. For the constrained PBFDAF algorithm matrix $\mathbf{G}$ in Eq. 7 is defined as

$$\mathbf{G} = \mathbf{F} \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{M-P} \end{bmatrix} \mathbf{F}^{-1} = \mathbf{F}\,\mathbf{\Pi}\,\mathbf{F}^{-1} \tag{8}$$

and represents the gradient correction operation. In the case of the unconstrained PBFDAF $\mathbf{G} = \mathbf{I}_M$, hence the expensive gradient correction is left out, leading to a cheaper, approximate algorithm. The unconstrained algorithm requires 3 DFTs per iteration, provided P is divisible by L, whereas the constrained PBFDAF has an extra $\frac{2N}{P}$ DFTs to compute. Although both updating schemes converge to the same Wiener solution the constrained variant has better convergence properties.

The block length L of the PBFDAF can be adjusted, resulting in a cheap adaptive filter with acceptable processing delay. Consider for example a realistic acoustic echo cancellation setup in which an unknown system w is modelled using the PBFDAF with an adaptive filter $\hat{w}$ having 1000 equivalent filter taps in the time domain. If the signals are sampled at 8 kHz and a block length $L=128$ is applied, the delay is 32 ms, which is acceptable, compared to the delay of 250 ms that would be introduced by the FDAF algorithm.

An ambiguity can occur with the unconstrained PBFDAF algorithm if $M > P+L-1$, as it is not guaranteed that the filter weights maintain the initial format set in Eq. 2. A random value ϵ can appear in Eq. 2 as an extra $(P+1)$–th element of $\hat{\boldsymbol{w}}_p^{(n)}$ for instance, as long as it is compensated for at the first element of $\hat{\boldsymbol{w}}_{p+1}^{(n)}$ [7]. This ambiguity can easily be compensated for by slightly changing Eqs. 4 and 5, i.e. computing $\mathbf{e}^{(n)}$ based on $L+\sigma$ instead of L signal samples, with $M = P+L-1+\sigma$:

$$\mathbf{d}^{(n)} = \begin{bmatrix} \mathbf{0} \\ \mathbf{d}_n \end{bmatrix} \begin{matrix} \updownarrow P-1 \\ \updownarrow L+\sigma \end{matrix}, \qquad \mathbf{d}_n = \begin{bmatrix} d[nL-\sigma+1] \\ \vdots \\ d[(n+1)L] \end{bmatrix} \Big\updownarrow L+\sigma \tag{9}$$

$$\mathbf{y}^{(n)} = \begin{bmatrix} \mathbf{0}_{P-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{L+\sigma} \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{N}{P}-1} \mathbf{X}_p^{(n)} \mathbf{w}_p^{(n)}. \tag{10}$$

The other equations remain unchanged and the additional algorithmic cost is almost negligible. The equations defining the PBFDAF are summarized in table I for $R = 1$. In fact table I defines the PBFDRAP algorithm, which —as will be shown in section III— reduces to the PBFDAF if parameter R is set to 1. The equivalent number of real operations is computed for each algorithmic step, assuming that a real multiplication and a real addition have the same complexity. A complex multiplication amounts to 6 equivalent real operations and a complex addition to 2 equivalent real operations. It is assumed that both x and d are real–valued and that M is a power of 2. As a consequence only $\frac{M}{2}+1$ out of M subbands need to be processed as the subbands are pairwise complex conjugated. Furthermore, efficient FFT routines can be used. The cost of an M–point real–input FFT is assumed to be $2M\log_2 M - 4M + 6$. For an M–point real–output IFFT $2M\log_2 M$ operations are counted. The cost or execution time associated with other operations such as memory accesses or data copying is not taken into account.

III. Partitioned Block Frequency–Domain RAP

Stepping several times through the filter and weight updating part of the PBFDAF algorithm (Eqs. 4, 6 and 7) while the block index n is kept constant, leads to an improved estimate of the error output and the filter weights :

$$\mathbf{X}_p^{(n)} \overset{\forall p}{=} \operatorname{diag}\left\{ \mathbf{F} \begin{bmatrix} x[(n+1)L - pP - M + 1] \\ \vdots \\ x[(n+1)L - pP] \end{bmatrix} \right\} \Bigg\uparrow M, \qquad p = 0 : \frac{N}{P} - 1 \tag{11}$$

$$\mathbf{d}^{(n)} = \begin{bmatrix} \mathbf{0} \\ \mathbf{d}_n \end{bmatrix} \begin{matrix} \updownarrow M-L \\ \updownarrow L \end{matrix}, \qquad \mathbf{d}_n = \begin{bmatrix} d[nL+1] \\ \vdots \\ d[(n+1)L] \end{bmatrix} \Bigg\updownarrow L \tag{12}$$

`for` $r = 1$ `to` R `do` $\{$

$$\mathbf{y}^{(n,r)} = \begin{bmatrix} \mathbf{0}_{M-L} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{N}{P}-1} \mathbf{X}_p^{(n)} \mathbf{w}_p^{(n,r)} \tag{13}$$

$$\mathbf{e}^{(n,r)} = \mathbf{d}^{(n)} - \mathbf{y}^{(n,r)} \tag{14}$$

$$\mathbf{w}_p^{(n,r+1)} \overset{\forall p}{=} \mathbf{w}_p^{(n,r)} + \mathbf{G}\boldsymbol{\Delta}\mathbf{X}_p^{(n)*}\mathbf{F}\mathbf{e}^{(n,r)}, \qquad p = 0 : \frac{N}{P} - 1 \tag{15}$$

$\}$

$$\mathbf{w}_p^{(n+1,1)} \stackrel{\forall p}{=} \mathbf{w}_p^{(n,R+1)}, \quad p = 0 : \frac{N}{P} - 1 \tag{16}$$

in which $\mathbf{w}_p^{(n,r)}$, $\mathbf{y}^{(n,r)}$, $\mathbf{e}^{(n,r)}$ are, respectively, the adaptive filter weights, the reconstructed system output and the error output at iteration step r. This extension of the PBFDAF algorithm which iterates R times on the same block of data $(\mathbf{X}_p^{(n)}, \mathbf{d}_n)$ will be called the partitioned block frequency–domain RAP (PBFDRAP) adaptive filter. RAP stands for Row Action Projection and was initially proposed as an improvement to the LMS algorithm. A good reference on Row Action Projection is [6]. The formulas defining the PBFDRAP adaptive filter and the complexity of the different algorithmic steps are summarized in table I. Remark that the PBFDRAP reduces to the PBFDAF if $R = 1$.

The PBFDRAP algorithm iterates R times on the same data $(\mathbf{X}_p^{(n)}, \mathbf{d}_n)$. Hence, an improved weight update $\mathbf{w}_p^{(n,R+1)}$ and a smaller a–posteriori error output $\mathbf{e}^{(n,R)}$ are obtained. If the number of iteration steps R is increased the algorithm further reduces the a–posteriori error. This is partly done by exploiting the signal characteristics, i.e. adapting the filter coefficients towards a solution that minimizes $\mathbf{e}^{(n,R)}$ for data block $(\mathbf{X}_p^{(n)}, \mathbf{d}_n)$, rather than trying to improve the model of the unknown system w as such. Therefore, a small a–posteriori error $\mathbf{e}^{(n,R)}$ does not necessarily imply a good echo path modelling. Hence, the quality of the model approximation should be evaluated by plotting the norm between the unknown system w and the model $\hat{w}$ or, if w is not known, by looking at the time evolution of the so–called a–priori error $\mathbf{e}^{(n,1)}$.

At first sight, iterating several times on the same block of data seems rather expensive (see table I). However, in section V it is shown how the complexity can be reduced, leading to fast versions of the PBFDRAP algorithm. Furthermore, nowadays echo cancellation is just one of the many signal processing building blocks that are combined to build a signal enhancement system, which includes apart from the echo cancellation e.g. also noise suppression and dereverberation. The signal enhancement unit may be used as a preprocessor to a more complex building block such as a speech recognizer for instance. Depending on the mode in which each of these units operates, the overall complexity of the system can vary from time to time such that at some time instances more processing power may be available for the preprocessing and the echo cancellation in particular. The number of iteration steps R can therefore be adapted based on the available processing power, leading to better echo suppression if R can be chosen large.

IV. On Iterating the PBFDRAP

The following equations

$$y^{(k,r)} = \mathbf{x}_k^T \mathbf{w}^{(k,r)} \tag{17}$$

$$e^{(k,r)} = d[k] - y^{(k,r)} \tag{18}$$

$$\mathbf{w}^{(k,r+1)} = \mathbf{w}^{(k,r)} + \mu\, \mathbf{x}_k e^{(k,r)} \tag{19}$$

define the iterated LMS algorithm with $r = 1, 2, ...$ being the iteration index and $\mathbf{x}_k$ and $\mathbf{w}^{(k,1)}$ column vectors of length N representing the input data and the initial adaptive filter at time k, respectively. It was shown [8] that reiterating the LMS algorithm leads to the normalized LMS algorithm. In other words, based on Eqs. 17–19, it can be shown [9] that

$$\lim_{R\to\infty} \mathbf{w}^{(k,R)} = \mathbf{w}^{(k,1)} + \mathbf{x}_k \frac{1}{\mathbf{x}_k^T \mathbf{x}_k}(d[k] - \mathbf{x}_k^T \mathbf{w}^{(k,1)}), \tag{20}$$

which is the weight update equation for the normalized LMS algorithm with stepsize $\mu = 1$. The same sort of derivation can be done for the PBFDRAP algorithm in order to check what happens if the number of iteration steps R goes to infinity. For the analysis different algorithmic settings are considered, making a distinction between unconstrained and constrained updating on the one hand and subband normalization and unnormalized implementation of the PBFDRAP on the other hand. In this section the results of these computations are presented and discussed. The detailed derivations can be found in [10].

A. Unnormalized and globally normalized unconstrained PBFDRAP

For the unnormalized unconstrained PBFDRAP, $\mathbf{G} = \mathbf{I}_M$ and $\mathbf{\Delta}$ is defined as $\mu\mathbf{I}_M$, with μ a constant. More generally, $\mathbf{\Delta} = \mu_n \mathbf{I}_M$, with μ_n a block dependent stepsize. For instance, if[2]

$$\mu_n = \frac{\mu}{\frac{1}{M}\sum_{m=0}^{M-1}\sum_{p=0}^{\frac{N}{P}-1} \mathbf{X}_p^{(n)}(m)\mathbf{X}_p^{(n)*}(m)} \tag{21}$$

$$= \frac{\mu}{\sum_{m=0}^{M-1}\sum_{p=0}^{\frac{N}{P}-1} x[(n+1)L - pP - m]^2}, \tag{22}$$

the algorithm performs a global normalization, leading to better convergence if signals with a high dynamic range (such as speech) are involved.

In [10] it is proven that if the number of iteration steps R goes to infinity, both the unnormalized and globally normalized unconstrained PBFDRAP approach a normalized version of the

[2] $\mathbf{X}_p^{(n)}(m)$ is the m–th diagonal element of diagonal matrix $\mathbf{X}_p^{(n)}$.

unconstrained PBFDAF that is based on projected subband energies and adapted with optimal stepsize $\mu_n = 1$, i.e.

$$\lim_{R\to\infty} \mathbf{w}_p^{(n,R)} \stackrel{\forall p}{=} \mathbf{w}_p^{(n,1)} + \mathbf{X}_p^{(n)*} \left(\boldsymbol{\mathcal{P}} \sum_{p=0}^{\frac{N}{P}-1} \mathbf{X}_p^{(n)} \mathbf{X}_p^{(n)*} \right)^{-1} \mathbf{F}\mathbf{e}^{(n,1)}, \tag{23}$$

in which $\boldsymbol{\mathcal{P}}$ is a projection matrix

$$\boldsymbol{\mathcal{P}} = \mathbf{F} \begin{bmatrix} \boldsymbol{\mathcal{G}} & -\boldsymbol{\mathcal{G}}\boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{D}}^{-1} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1}, \tag{24}$$

with $\boldsymbol{\mathcal{G}} = \left((\boldsymbol{\mathcal{A}} - \boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{D}}^{-1}\boldsymbol{\mathcal{C}})\boldsymbol{\mathcal{K}}\right)^{-1}$,

$$\begin{bmatrix} \boldsymbol{\mathcal{A}} & \boldsymbol{\mathcal{B}} \\ \boldsymbol{\mathcal{C}} & \boldsymbol{\mathcal{D}} \end{bmatrix} = \mathbf{F}^{-1} \sum_{p=0}^{\frac{N}{P}-1} \mathbf{X}_p^{(n)} \mathbf{X}_p^{(n)*} \mathbf{F}, \tag{25}$$

$\boldsymbol{\mathcal{D}}$ an $L \times L$ matrix and $\boldsymbol{\mathcal{K}}$ a random $(M-L) \times (M-L)$ matrix.

In figure 2 it is illustrated how the convergence speed of the unnormalized unconstrained PBFDRAP algorithm improves as R increases. A random FIR filter w of order 31 was generated and fed with a white noise signal x (see figure 1). The unnormalized unconstrained PBFDRAP ($P = L = 16$, $M = 32$ and $N = 32$) was used to identify w. The ambiguity which typically occurs with unconstrained updating was compensated for by applying Eqs. 9 and 10. In figure 2 the normalized short–time energy of the a–priori error (ERLE) is plotted as a function of time. It appears that if R is increased the bottom curve ($R = \infty$) is approached, which corresponds to Eq. 23.

B. Subband–normalized unconstrained PBFDRAP

The PBFDRAP may be considered as a subband adaptive system [7]. Hence, the convergence properties can be optimized by applying different stepsizes to each of the subband adaptive filters. Therefore, matrix $\boldsymbol{\Delta}$ can be chosen as follows :

$$\boldsymbol{\Delta} = \mu \left(\sum_{p=0}^{\frac{N}{P}-1} \mathbf{X}_p^{(n)} \mathbf{X}_p^{(n)*} \right)^{-1}. \tag{26}$$

In this way in each subband the adaptation stepsize μ is divided by the subband energy. It is proven (see [10]) that the subband–normalized unconstrained PBFDRAP converges to the subband–normalized unconstrained PBFDAF with $\mu = 1$: increasing the number of iteration steps R has the same effect as applying a larger stepsize μ.

C. Unnormalized and globally normalized constrained PBFDRAP

In [9] the relation between two adaptive filtering techniques was examined : iterated Block–LMS and the Partial Rank Algorithm (PRA). The PRA is a block version of the Affine Projection Algorithm (APA) [6] and can be interpreted as a block–normalized version of the Block–LMS algorithm [9]. For each block of L samples the PRA performs the following steps :

$$\mathbf{X}_n = \begin{bmatrix} x[nL+1] & \dots & x[(n+1)L] \\ \vdots & \ddots & \vdots \\ x[nL-N+2] & \dots & x[(n+1)L-N+1] \end{bmatrix} \tag{27}$$

$$\mathbf{e}_n = \mathbf{d}_n - \mathbf{X}_n^T \hat{\mathbf{w}}_n \tag{28}$$

$$\hat{\mathbf{w}}_{n+1} = \hat{\mathbf{w}}_n + \mu_n \mathbf{X}_n \left(\mathbf{X}_n^T \mathbf{X}_n\right)^{-1} \mathbf{e}_n. \tag{29}$$

$\mathbf{X}_n$ is a (real–valued) $N \times L$ Toeplitz matrix containing the x–samples, $\mathbf{d}_n$ is defined in Eq. 5. The $N \times 1$ vector $\hat{\mathbf{w}}_n$ contains the adaptive filter weights and models the unknown system w. Further, μ_n is a block dependent stepsize. If $L = 1$ the PRA reduces to the normalized LMS algorithm. Hence, the PRA is scalable and lies in between the normalized LMS algorithm ($L = 1$) and a block adaptive least squares estimate of the unknown system ($L = N$).

It was shown that iterated Block–LMS approaches PRA with stepsize $\mu = 1$, if the number of iteration steps goes to infinity [9]. Stepsize $\mu = 1$ guarantees optimal convergence. As the unnormalized constrained PBFDRAP is an exact representation of iterated Block–LMS in the frequency domain, it will also approach the PRA with stepsize $\mu = 1$, if R goes to infinity. In [10] it was verified more formally that the unnormalized and also the globally normalized constrained PBFDRAP algorithms approach the PRA. By series expansion the expensive matrix inverse operation in Eq. 29 is approximated resulting in a cheaper algorithm with reduced performance. An artificial acoustic environment was simulated using the image method presented in [11] : an FIR filter w of 1024 taps was generated and fed with colored noise x (see figure 1) that was obtained by filtering a white noise sequence with a first order lowpass IIR filter having a pole at 0.4. The globally normalized constrained PBFDRAP ($P = L = 128$, $M = 256$ and $N = 1024$) was applied to identify w. The number of iterations R was increased from 1 to 10. For each curve the adaptation constant μ (see Eq. 22) was optimized to obtain maximum convergence speed. In figure 3 the short–time energy of the a–priori error is plotted as a function of time. It appears that if R is increased the bottom curve ($R = \infty$) is approached, which corresponds to the PRA ($L = 128, N = 1024$) with stepsize $\mu = 1$.

D. Subband–normalized constrained PBFDRAP

Also for the subband–normalized constrained PBFDRAP a similar derivation can be made to check what happens if R goes to infinity. Stability problems arise however in this case as the algorithm is found to sometimes diverge for large values of R. Starting from the weight update equation (Eq. 15) the limit for $R \to \infty$ can be computed, but for some data sets it appears that the algorithm will converge to an unstable filter [10]. However, this does not automatically imply divergence of the algorithm for finite R, but indicates that for some data sets iterating this algorithm is expected to invoke stability problems. It appears that also for this algorithmic variant increasing the number of iteration steps R has the same effect as applying a larger stepsize μ (cf section IV-B).

A white noise signal x was fed into a randomly generated FIR filter w of order 15. Filter w was adaptively estimated using the subband–normalized constrained PBFDRAP ($N = 16$, $M = 16$, $P = L = 8$ and $\mu = 0.2$). The results are shown in figure 4. If R is increased from 1 to 10 additional error suppression is obtained. If R is further increased to 100 the filter becomes unstable.

V. Fast PBFDRAP

At first sight iterating R times on the same block of data is rather expensive, the implementation cost being almost R times higher. It will however be shown in this section that the complexity can be reduced, leading to a set of fast algorithmic variants.

At the end of iteration step R the error output $\mathbf{e}^{(n,R)}$ (see table I) can be written as

$$\mathbf{e}^{(n,R)} = (\mathbf{I}_M - \mathbf{N}_n)^{R-1}\mathbf{e}^{(n,1)} \tag{30}$$

with

$$\mathbf{N}_n = \begin{bmatrix} \mathbf{0}_{M-L} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{N}{P}-1} \mathbf{X}_p^{(n)} \mathbf{G} \boldsymbol{\Delta} \mathbf{X}_p^{(n)^*} \mathbf{F} \tag{31}$$

and the adaptive filter weights are given by

$$\mathbf{w}_p^{(n,R+1)} \stackrel{\forall p}{=} \mathbf{w}_p^{(n,1)} + \mathbf{G} \boldsymbol{\Delta} \mathbf{X}_p^{(n)^*} \mathbf{F} \sum_{r=1}^{R} \mathbf{e}^{(n,r)}, \quad p = 0 : \frac{N}{P} - 1. \tag{32}$$

Proof :

Eqs. 30 and 32 are trivially fulfilled for $R = 1$ as can be easily verified from table I. Now assuming that Eqs. 30 and 32 hold for iteration step $R - 1$, i.e.

$$\mathbf{e}^{(n,R-1)} = (\mathbf{I}_M - \mathbf{N}_n)^{R-2}\mathbf{e}^{(n,1)} \tag{33}$$

and

$$\mathbf{w}_p^{(n,R)} \overset{\forall p}{=} \mathbf{w}_p^{(n,1)} + \mathbf{G}\boldsymbol{\Delta}\mathbf{X}_p^{(n)^*}\mathbf{F}\sum_{r=1}^{R-1}\mathbf{e}^{(n,r)}, \tag{34}$$

it needs to be proven that Eqs. 30 and 32 also hold for iteration step R.
After the $(R-1)$–th iteration (see table I)

$$\mathbf{w}_p^{(n,R)} \overset{\forall p}{=} \mathbf{w}_p^{(n,R-1)} + \mathbf{G}\boldsymbol{\Delta}\mathbf{X}_p^{(n)^*}\mathbf{F}\mathbf{e}^{(n,R-1)}, \tag{35}$$

such that after iteration step R (using Eqs. 13, 14, 31, 33 and 35)

$$\begin{aligned}\mathbf{e}^{(n,R)} &= \mathbf{d}^{(n)} - \begin{bmatrix}\mathbf{0}_{M-L} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}_L\end{bmatrix}\mathbf{F}^{-1}\sum_{p=0}^{\frac{N}{P}-1}\mathbf{X}_p^{(n)}\mathbf{w}_p^{(n,R)}\\ &= \left(\mathbf{I}_M - \mathbf{N}_n\right)\mathbf{e}^{(n,R-1)}\\ &= \left(\mathbf{I}_M - \mathbf{N}_n\right)^{R-1}\mathbf{e}^{(n,1)}\end{aligned} \tag{36}$$

and from Eqs. 15 and 34

$$\begin{aligned}\mathbf{w}_p^{(n,R+1)} &\overset{\forall p}{=} \mathbf{w}_p^{(n,R)} + \mathbf{G}\boldsymbol{\Delta}\mathbf{X}_p^{(n)^*}\mathbf{F}\mathbf{e}^{(n,R)}\\ &\overset{\forall p}{=} \mathbf{w}_p^{(n,1)} + \mathbf{G}\boldsymbol{\Delta}\mathbf{X}_p^{(n)^*}\mathbf{F}\sum_{r=1}^{R}\mathbf{e}^{(n,r)},\end{aligned} \tag{37}$$

which completes the proof.

□

A. Fast PBFDRAP, version 1

Using relations 30 and 32 the equations defining the PBFDRAP can be reformulated. If matrix

$$\mathbf{B}_n = \mathbf{I}_M - \begin{bmatrix}\mathbf{0}_{M-L} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}_L\end{bmatrix}\mathbf{F}^{-1}\sum_{p=0}^{\frac{N}{P}-1}\mathbf{X}_p^{(n)}\mathbf{G}\boldsymbol{\Delta}\mathbf{X}_p^{(n)^*}\mathbf{F} \tag{38}$$

is defined and inserted into Eq. 36, it appears that $\mathbf{e}^{(n,r)}$ can be written as

$$\mathbf{e}^{(n,r)} = \mathbf{B}_n\mathbf{e}^{(n,r-1)} \tag{39}$$

for $r > 1$, such that only $\mathbf{e}^{(n,1)}$ needs to be computed explicitly following Eqs. 12–14. $\mathbf{B}_n$ is computed once for each data block and $\mathbf{e}^{(n,r)}$, $r > 1$ follow from Eq. 39. In order to compute $\mathbf{w}_p^{(n,R+1)}$, Eq. 32 can be used. The intermediate error outputs $\mathbf{e}^{(n,r)}$ are accumulated and stored in vector $\mathbf{c}^{(n,r)}$:

$$\begin{aligned}\mathbf{c}^{(n,1)} &= \mathbf{e}^{(n,1)},\\ \mathbf{c}^{(n,r)} &= \mathbf{c}^{(n,r-1)} + \mathbf{e}^{(n,r)}, \quad r > 1\end{aligned} \tag{40}$$

such that Eq. 32 becomes

$$\mathbf{w}_p^{(n,R+1)} = \mathbf{w}_p^{(n,1)} + \mathbf{G}\boldsymbol{\Delta}\mathbf{X}_p^{(n)^*}\mathbf{F}\mathbf{c}^{(n,R)} \tag{41}$$

The different algorithmic steps of this first version of the fast PBFDRAP are summarized in table II.

B. Fast PBFDRAP, version 2

The explicit matrix–vector multiplication $\mathbf{B}_n\mathbf{e}^{(n,r-1)}$ in Eq. 39, however, is quite expensive. The algorithm can be simplified by replacing the matrix–vector multiplications by FFT operations. Matrix $\mathbf{D}_n$ is introduced and is defined as

$$\mathbf{D}_n = \sum_{p=0}^{\frac{N}{P}-1} \mathbf{X}_p^{(n)}\mathbf{G}\boldsymbol{\Delta}\mathbf{X}_p^{(n)^*}, \tag{42}$$

such that Eq. 39 can be written as

$$\mathbf{e}^{(n,r)} = \mathbf{e}^{(n,r-1)} - \begin{bmatrix} \mathbf{0}_{M-L} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1}\mathbf{D}_n\mathbf{F}\mathbf{e}^{(n,r-1)}. \tag{43}$$

For the unconstrained PBFDRAP for which $\mathbf{G} = \mathbf{I}_M$, $\mathbf{D}_n$ is a diagonal matrix. The matrix–vector product $\mathbf{D}_n\mathbf{F}\mathbf{e}^{(n,r-1)}$ can therefore be replaced by an element–wise vector–vector multiplication which leads to a reduced cost. The algorithm is summarized in table III.

C. Fast PBFDRAP, version 3

If the order of the unknown FIR system w (see figure 1) is high with respect to the filter partition length P, the algorithm can be further simplified. Note that $\boldsymbol{\Delta}$ and $\mathbf{X}_p^{(n)^*}$ in Eq. 42 are diagonal matrices and hence commute. If $P = L$, Eq. 42 can be replaced by the following update equation :

$$\bar{\mathbf{D}}_n = \bar{\mathbf{D}}_{n-1} + \mathbf{X}_0^{(n)}\mathbf{G}\mathbf{X}_0^{(n)^*} - \mathbf{X}_{\frac{N}{P}}^{(n)}\mathbf{G}\mathbf{X}_{\frac{N}{P}}^{(n)^*} \tag{44}$$

$$\mathbf{D}_n = \bar{\mathbf{D}}_n\boldsymbol{\Delta}. \tag{45}$$

In this way extra savings can be made. The algorithm is presented in table IV.

D. Fast constrained PBFDRAP

The algorithmic schemes presented in tables III and IV are less attractive if constrained updating is used as the matrix–vector product $\mathbf{D}_n\mathbf{F}\mathbf{e}^{(n,r-1)}$ cannot be replaced by an element–wise vector–vector multiplication in this case (see Eq. 8). Starting from the algorithm presented in

table II another fast algorithm can be derived. From Eqs. 12, 13 and 14 it can be verified that $\mathbf{e}^{(n,r)}$ can be written as

$$\mathbf{e}^{(n,r)} = \begin{bmatrix} \mathbf{0}_{M-L\times 1} \\ \mathbf{e}_n^{(r)} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_L \end{bmatrix} \mathbf{e}_n^{(r)}, \tag{46}$$

if constrained updating is applied, with $\mathbf{e}_n^{(r)}$ an $L \times 1$ vector. Hence, Eq. 39 can be rewritten as

$$\mathbf{e}_n^{(r)} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{B}_n \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_L \end{bmatrix} \mathbf{e}_n^{(r-1)}. \tag{47}$$

Now,

$$\hat{\mathbf{B}}_n = \begin{bmatrix} \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{B}_n \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_L \end{bmatrix} \tag{48}$$

can be updated instead of $\mathbf{B}_n$. If constrained updating is considered and $\mathbf{\Delta} = \mu_n \mathbf{I}_M$, and Eqs. 8 and 38 are inserted into Eq. 48,

$$\begin{aligned} \hat{\mathbf{B}}_n &= \mathbf{I}_L - \mu_n \begin{bmatrix} \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{N}{P}-1} \mathbf{X}_p^{(n)} \mathbf{\Pi}\, \mathbf{F}^{-1} \mathbf{X}_p^{(n)*} \mathbf{F} \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_L \end{bmatrix} \\ &= \mathbf{I}_L - \mu_n \sum_{p=0}^{\frac{N}{P}-1} \begin{bmatrix} \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \mathbf{X}_p^{(n)} \mathbf{\Pi}\, \mathbf{\Pi}\, \mathbf{X}_p^{(n)*} \mathbf{F} \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_L \end{bmatrix} \end{aligned} \tag{49}$$

is obtained. As $\mathbf{X}_p^{(n)}$ is a diagonal matrix, defined by Eq. 11, $\bar{\mathbf{X}}_{n,p}^T = \mathbf{F}^{-1} \mathbf{X}_p^{(n)} \mathbf{F}$ is a right–circulant $M \times M$ matrix. Hence, the discrete Fourier transform of the first row of $\bar{\mathbf{X}}_{n,p}$ are the diagonal elements of $\mathbf{X}_p^{(n)}$. Taking in account Eq. 11,

$$\bar{\mathbf{X}}_{n,p} = (\mathbf{F}^{-1} \mathbf{X}_p^{(n)} \mathbf{F})^T = \begin{bmatrix} x[(n+1)L - pP - M + 1] & \dots & x[(n+1)L - pP] \\ x[(n+1)L - pP] & \dots & x[(n+1)L - pP - 1] \\ \vdots & \ddots & \vdots \\ x[(n+1)L - pP - M + 2] & \dots & x[(n+1)L - pP - M + 1] \end{bmatrix} \tag{50}$$

is found. Now, $\bar{\mathbf{X}}_{n,p}^* = (\mathbf{F}^{-1} \mathbf{X}_p^{(n)} \mathbf{F})^H = \mathbf{F}^{-1} \mathbf{X}_p^{(n)*} \mathbf{F}$ and as x is real–valued $\bar{\mathbf{X}}_{n,p}^* = \bar{\mathbf{X}}_{n,p}$, such that

$$\begin{aligned} \hat{\mathbf{B}}_n &= \mathbf{I}_L - \mu_n \sum_{p=0}^{\frac{N}{P}-1} \left(\mathbf{\Pi}\, \bar{\mathbf{X}}_{n,p} \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_L \end{bmatrix} \right)^T \left(\mathbf{\Pi}\, \bar{\mathbf{X}}_{n,p} \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_L \end{bmatrix} \right) \\ &= \mathbf{I}_L - \mu_n \sum_{p=0}^{\frac{N}{P}-1} \mathbf{X}_{n,p}^T \mathbf{X}_{n,p} \end{aligned} \tag{51}$$

with $\mathbf{X}_{n,p}$ the following $P \times L$ Toeplitz matrix :

$$\mathbf{X}_{n,p} = \begin{bmatrix} x[nL-pP+1] & \dots & x[(n+1)L-pP] \\ \vdots & \ddots & \vdots \\ x[nL-pP-P+2] & \dots & x[(n+1)L-pP-P+1] \end{bmatrix}. \tag{52}$$

If $P = L$, Eq. 51 can be replaced by the following update equation

$$\bar{\mathbf{B}}_n = \bar{\mathbf{B}}_{n-1} + \mathbf{X}_{n,0}^T \mathbf{X}_{n,0} - \mathbf{X}_{n,\frac{N}{P}}^T \mathbf{X}_{n,\frac{N}{P}} \tag{53}$$

$$\hat{\mathbf{B}}_n = \mathbf{I}_L - \mu_n \bar{\mathbf{B}}_n. \tag{54}$$

In this way extra savings can be made. The algorithm is presented in table V.

VI. Computational Cost

In this section the PBFDRAP algorithm is analyzed from a computational complexity point of view. Which of the five algorithmic schemes, presented in tables I, II, III, IV or V, will lead to the cheapest implementation, depends on the actual values of the parameters (M, N, L, P and R) and the type of updating ((un)normalized/(un)constrained) that is used. For a typical application for which the PBFDRAP is used, i.e. for the identification of high–order FIR systems, the scheme presented in table IV will in general be more appropriate if unconstrained updating is preferred. The intermediate schemes presented in tables II and III are less interesting as they are typically more expensive than the algorithm of table IV if high–order FIR systems are identified. For the constrained PBFDRAP the first (non–fast) and the last scheme (table V) seem more attractive. In this section the "standard" (non–fast) implementation (table I) is compared with the most promising fast versions of the algorithm, being the algorithm of table IV for unconstrained updating and that of table V for constrained updating.

A. Unconstrained PBFDRAP

The equivalent number of real operations needed to process a block of L samples with the "standard" (non–fast) unnormalized unconstrained PBFDRAP (table I) is

$$2M \log_2 M(1+2R) - 4M(R+1) + 6(1+R) + RL + 16\frac{RN}{P} + 8\frac{RMN}{P}, \tag{55}$$

where it is assumed that P is divisible by L, that M is a power of 2 and that x and d are real–valued. The cost for processing a block of L samples with the fast unnormalized unconstrained PBFDRAP for which $P = L$ (table IV) is

$$2M \log_2 M(1+2R) - M(3R+\frac{1}{2}) + (13+8R) + 2RL - L + 16\frac{N}{L} + 8\frac{MN}{L}. \tag{56}$$

Cost estimates for the standard PBFDRAP and the fast PBFDRAP algorithm of table IV are presented in figure 5. For a realistic comparison the following parameters were chosen : $M = 256$, $P = L = 128$, $N = 1024$. It is further assumed that the sampling frequency is $f_s = 8000$ Hz. It appears that the fast PBFDRAP algorithm of table IV outperforms the standard PBFDRAP, except for $R = 1$ for which it is slightly more expensive. Iterating twice with the standard approach is as expensive as iterating 4 times with the fast algorithm. It is furthermore observed that the PBFDRAP algorithm is feasible in terms of real–time operation.

B. Constrained PBFDRAP

For the constrained PBFDRAP the standard implementation presented in table I and the fast algorithm of table V seem more appropriate and typically lead to the lowest implementation cost. The equivalent number of real operations needed to process a block of L samples with the unnormalized constrained PBFDRAP using the "standard" (non–fast) algorithm of table I is

$$2M(1 + 2R + 2\frac{RN}{P}) \log_2 M - 4M(R + 1) + 6(1 + R) + RL + 22\frac{RN}{P} + 4\frac{RMN}{P}. \qquad (57)$$

Also here it is assumed that P is divisible by L, that M is a power of 2 and that x and d are real–valued. The equivalent formula for the fast algorithm of table V is ($P = L$)

$$2M(3 + 2\frac{N}{L}) \log_2 M - 9M + 10 + L^3 + 2RL^2 + 4L + 22\frac{N}{L} + 4\frac{MN}{L}. \qquad (58)$$

The standard and the fast version of the unnormalized constrained PBFDRAP algorithm (formula 57 and 58) are compared in figure 6. The following realistic parameters are chosen : $M = 64$, $P = L = 32$, $N = 1024$ and the sampling frequency $f_s = 8000$ Hz. It appears that the fast PBFDRAP outperforms the standard PBFDRAP in terms of computational complexity, except for $R = 1$, and that algorithm is feasible in terms of real–time operation.

In figure 7 the number of iteration steps R is plotted for which the standard implementation of the unnormalized constrained PBFDRAP (table I) is as expensive as the fast implementation of table V. Each curve corresponds to a different block length L. For a parameter set (N, R) above the curve, the fast algorithm leads to the lowest cost. Below the curve the standard algorithm of table I is more appropriate. It appears that Eq. 58 is more or less insensitive to changes in R, whereas Eq. 57 strongly depends on R. Observe that for small values of L, and N and R sufficiently large, the algorithm of table V leads to the lowest cost.

Infinitely iterating the unnormalized constrained PBFDRAP leads to the PRA algorithm (see also section IV-C). For the weight updating of the PRA the inverse of a regularized symmetric matrix $\mathbf{M}_n + \delta\mathbf{I}_L = \mathbf{X}_n^T\mathbf{X}_n + \delta\mathbf{I}_L$ has to be computed (see Eq. 29). As the data is shifted block

by block efficient QR–based updating of $\mathbf{X}_n^T\mathbf{X}_n$ is not possible for $L > 1$. However, if N is a multiple of L,

$$\mathbf{X}_n^l = \begin{bmatrix} x[nL - lL + 1] & \dots & x[nL + L - lL] \\ \vdots & \ddots & \vdots \\ x[nL - L + 2 - lL] & \dots & x[nL - lL + 1] \end{bmatrix}, \quad l = 0 : \frac{N}{L} \tag{59}$$

can be defined and $\mathbf{X}_n^T\mathbf{X}_n$ can be updated as

$$\begin{aligned} \mathbf{M}_n = \mathbf{X}_n^T\mathbf{X}_n &= \begin{bmatrix} \left(\mathbf{X}_n^0\right)^T & \left(\mathbf{X}_n^1\right)^T & \dots & \left(\mathbf{X}_n^{N/L-1}\right)^T \end{bmatrix} \begin{bmatrix} \mathbf{X}_n^0 \\ \mathbf{X}_n^1 \\ \vdots \\ \mathbf{X}_n^{N/L-1} \end{bmatrix} \\ &= \mathbf{M}_{n-1} + \left(\mathbf{X}_n^0\right)^T\mathbf{X}_n^0 - \left(\mathbf{X}_n^{N/L}\right)^T\mathbf{X}_n^{N/L}. \end{aligned} \tag{60}$$

This reduces the cost substantially if N/L is large. As $\mathbf{M}_n$ is symmetric and $\left(\mathbf{X}_n^{N/L}\right)^T\mathbf{X}_n^{N/L}$ can be recovered from a previous iteration, the cost for updating $\mathbf{M}_n$ amounts to $L^3 + \frac{3}{2}L^2 + \frac{L}{2}$. Furthermore, $\mathbf{p}_n = (\mathbf{M}_n + \delta\mathbf{I}_L)^{-1}\mathbf{e}_n$ can be computed efficiently by solving the linear set of equations $(\mathbf{M}_n + \delta\mathbf{I}_L)\mathbf{p}_n = \mathbf{e}_n$ using Gauss elimination (cost $\frac{2L^3}{3}$) and back substitution (cost L^2). The equivalent number of real operations needed to process a block of L samples with the PRA (see table VI) is then equal to

$$4NL + \frac{5L^3}{3} + \frac{5L^2}{2} + \frac{5L}{2}. \tag{61}$$

R_{crit}, the value for R for which the unnormalized constrained PBFDRAP is as expensive as PRA, can now be computed. By combining Eqs. 57 and 61 (PRA vs. non–fast PBFDRAP), it is found that

$$R_{crit1} = \frac{4NL + \frac{5}{3}L^3 + \frac{5}{2}L^2 + \frac{5}{2}L - 2M\log_2 M + 4M - 6}{4M\log_2 M(1 + \frac{N}{P}) - 4M + 6 + L + 22\frac{N}{P} + 4\frac{MN}{P}}. \tag{62}$$

In most practical applications, $M = 2L = 2P$. Hence,

$$R_{crit1} = \frac{4NL + \frac{5}{3}L^3 + \frac{5}{2}L^2 + \frac{5}{2}L - 4L\log_2 L + 4L - 6}{8L\log_2 L(1 + \frac{N}{L}) + 6 + L + 22\frac{N}{L} + 16N}. \tag{63}$$

By combining Eqs. 58 and 61 (PRA vs. fast PBFDRAP), one obtains

$$R_{crit2} = \frac{4NL + \frac{2}{3}L^3 + \frac{5}{2}L^2 - \frac{3}{2}L - 2M(3 + 2\frac{N}{L})\log_2 M + 9M - 10 - 22\frac{N}{L} - 4\frac{MN}{L}}{2L^2}. \tag{64}$$

In figure 8, $R_{crit} = \max(R_{crit1}, R_{crit2})$ is plotted for $\frac{N}{L}$=1, 2, 5, 10 and 20. Block length L varies between 2 and 1024 and $M = 2L = 2P$. It appears that for large block lengths, the iterated PBFDRAP is a cheaper alternative for the PRA.

VII. Simulation Examples

The different algorithms presented in this text were applied to an acoustic echo cancellation setup (see figure 1) for validation and comparison. A lowpass noise signal having the spectral characteristics of speech was fed into a loudspeaker and recorded with a microphone in a controlled stationary laboratory environment. No local near–end talker signal s was added on top.

In a first experiment the unknown acoustic path was estimated using the globally normalized PBFDRAP (cf. Eq. 22), varying the type of updating (unconstrained, constrained) and the number of iterations steps R. The other parameters were kept constant and were chosen as follows : $L = P = 128$, $M = 256$, $N = 1024$ (see section II for more explanation). For the unconstrained updating Eqs. 9 and 10 were employed to compensate for the ambiguity which can occur in that case (see section II). For each of the algorithmic settings the stepsize μ was tuned to maximize the convergence speed. This optimal stepsize was divided by 10 and applied to the adaptive filter in order to compare the algorithms under realistic conditions. If the stepsize is kept significantly smaller than the theoretical optimum large deviations of the filter coefficients can be avoided if, for instance, the algorithm would be used in the presence of double–talk[3]. For each of the adaptive filters the short–time energy of the a–priori error output e is shown as a function of time in figure 9.

In a second experiment the unknown acoustic path was estimated using the subband–normalized PBFDRAP (cf. Eq. 26), varying the type of updating (unconstrained, constrained) and the number of iterations steps R. Also in this experiment the ambiguity was compensated for in the case of unconstrained updating and $L = P = 128$, $M = 256$, $N = 1024$. Again, the adaptation stepsizes were tuned to maximize the convergence speed and were then divided by 10. For each of the adaptive filters the short–time energy of the a–priori error output is shown in figure 10 as a function of time.

Both in figure 9 and 10 the error output of the PRA algorithm (table VI, $L = 128$, $N = 1024$, adaptation stepsize an order of magnitude below the optimum) is shown for comparison. Furthermore, for each of the adaptive filters the algorithmic complexity was computed, assuming a sampling frequency of 8000 Hz. The cost estimates are based on Eq. 61 and on Eqs. 55–58, which assume unnormalized updating. For the normalization Eqs. 21 and 26 are computed using updating and downdating techniques, amounting to an extra $5M + 23$ operations per block for

[3]During double–talk the local signal source s is active (see figure 1) and hence the adaptation stepsize μ must be set immediately to zero in order to freeze the coefficients of the adaptive filter [4]. If the adaptation is switched off too late the filter coefficients easily drift away from their Wiener solution whenever the stepsize is too large, leading to a bad model and a larger error output e.

the globally normalized PBFDRAP and $8M + 16$ operations per block in the case of subband normalization. The results of the cost computation are presented in table VII.

It is clear that by increasing the number of iteration steps R the error suppression can be improved. As the input signal is colored, the subband–normalized PBFDRAP converges faster than the globally normalized PBFDRAP. Although stability is not guaranteed for $R \to \infty$, the subband–normalized constrained PBFDRAP offers the best performance for realistic values of R. It appears furthermore from table VII that the PBFDRAP algorithm can be considered as a low–cost alternative for the PRA for large block lengths L. Finally, if the a–posteriori errors are plotted instead of the a–priori errors extra echo enhancement can be obtained. However, during double–talk a–priori errors should be passed to the output to avoid near–end signal cancellation.

VIII. Conclusion

In this paper the PBFDRAP was discussed, an adaptive algorithm that combines row action projection and partitioned block frequency–domain adaptive filtering. For some parameter settings the PBFDRAP algorithm approaches well–known adaptive filtering algorithms such as the PRA. Fast versions of the algorithm were derived, leading to a reduced algorithmic complexity. Based on simulation results it was shown that in a realistic signal enhancement setup such as acoustic echo cancellation, the PBFDRAP can be employed to obtain improved system estimates outperforming the standard partitioned block frequency–domain adaptive filter.

Acknowledgment

This research work was carried out at the ESAT laboratory of the Katholieke Universiteit Leuven, in the frame of the Belgian State Interuniversity Poles of Attraction Programme – IUAP P4–02 (1997–2001) : Modeling, Identification, Simulation and Control of Complex Systems, Concerted Research Action GOA–MEFISTO–666 (Mathematical Engineering for Information and Communication Systems Technology) of the Flemish Government, Research Project FWO no. G.0295.97 (‘Design and implementation of adaptive digital signal processing algorithms for broadband applications’) and IWT project AUT/970517: MUSETTE (’Multimicrophone Signal Enhancement Techniques for handsfree telephony and voice controlled systems’). The scientific responsibility is assumed by its authors.

References

[1] B. Widrow and S. Stearns. *Adaptive Signal Processing.* Prentice Hall, Englewood Cliffs, New Jersey, 1985.

[2] J. Shynk. Frequency–Domain and Multirate Adaptive Filtering. *IEEE Signal Processing Magazine*, 9(1):15–37, January 1992.

[3] G. Egelmeers, P. Sommen, and J. de Boer. Realization of an Acoustic Echo Canceller on a Single DSP. In *Proceedings of the European Signal Processing Conference (EUSIPCO96)*, pages 33–36, Trieste, Italy, September 1996.

[4] J. Páez Borrallo and M. García Otero. On the implementation of a partitioned block frequency domain adaptive filter (PBFDAF) for long acoustic echo cancellation. *Signal Processing*, 27:301–315, June 1992.

[5] J.-S. Soo and K. Pang. Multidelay Block Frequency Domain Adaptive Filter. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38(2):373–376, February 1990.

[6] S. Gay. *Fast Projection Algorithms with Application to Voice Echo Cancellation*. PhD thesis, Rutgers, The State University of New Jersey, New Brunswick, New Jersey, October 1994.

[7] K. Eneman and M. Moonen. Hybrid Subband/Frequency–Domain Adaptive Systems. *Signal Processing*, 81(1):117–136, January 2001.

[8] D. Morgan and S. Kratzer. On a Class of Computationally Efficient, Rapidly Converging, Generalized NLMS Algorithms. *IEEE Signal Processing Letters*, 3(8):245–247, August 1996.

[9] K. Eneman, S. Doclo, and M. Moonen. A Comparison between Iterative block–LMS and Partial Rank Algorithm. Technical Report 99–52, ESAT–SISTA, Katholieke Universiteit Leuven, Belgium, March 1999. *http://www.esat.kuleuven.ac.be/~eneman/abstracts/99-52.html*

[10] K. Eneman and M. Moonen. On Iterating the Partitioned Block Frequency–Domain Adaptive Filter. Technical Report 00–127, ESAT–SISTA, Katholieke Universiteit Leuven, Belgium, December 2000. *http://www.esat.kuleuven.ac.be/~eneman/abstracts/00-127.html*

[11] J. Allen and D. Berkley. Image method for efficiently simulating small–room acoustics. *Journal of the Acoustical Society of America*, 65(4):943–950, April 1979.

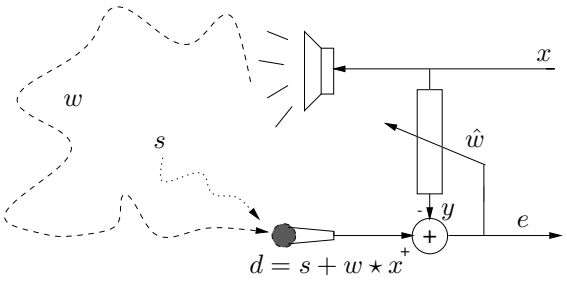


Fig. 1. Acoustic echo cancellation setup : d is a filtered version of x with an unknown acoustic path $w[k]$. In most applications a so–called local near–end talker signal s is added to $w \star x$ such that $d = s + w \star x$. The ultimate goal is to suppress $w \star x$ at the output e and to retain a non–distorted version of s.

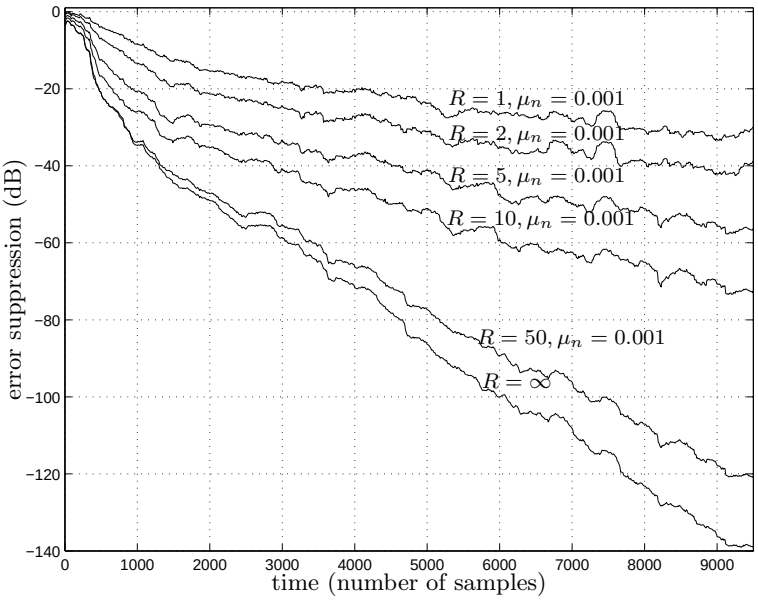


Fig. 2. A random FIR filter w of order 31 was generated and fed with a white noise signal. The unnormalized unconstrained PBFDRAP was used to identify w. The short–time energy of the a–priori error output is plotted as a function of time. It appears that if R is increased the bottom curve ($R = \infty$) is approached, which corresponds to a normalized version of the PBFDAF that is based on projected subband energies and adapted with stepsize $\mu_n = 1$ (Eq. 23).

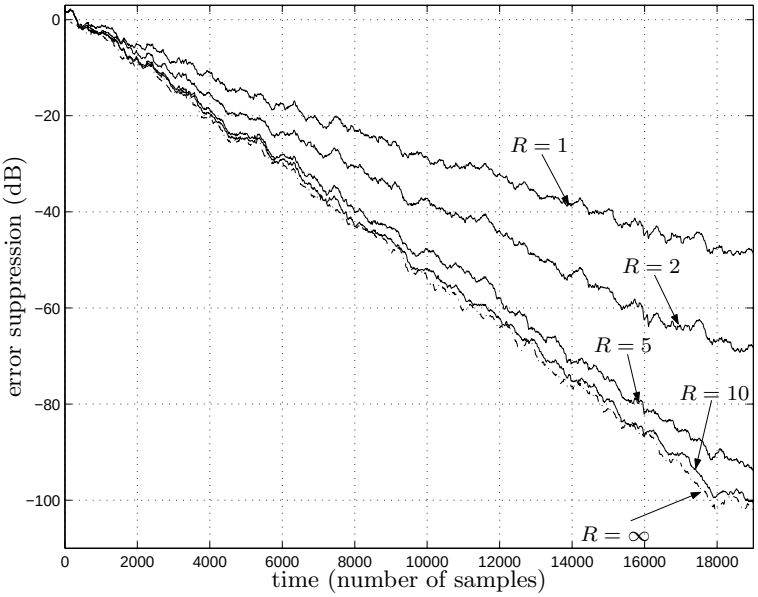


Fig. 3. An artificial acoustic path w was generated and fed with colored noise x. The acoustic path was identified with the globally normalized constrained PBFDRAP. The short–time energy of the a–priori error output is plotted as a function of time. It appears that if R is increased the bottom curve ($R = \infty$) is approached, which corresponds to the PRA with stepsize $\mu = 1$.

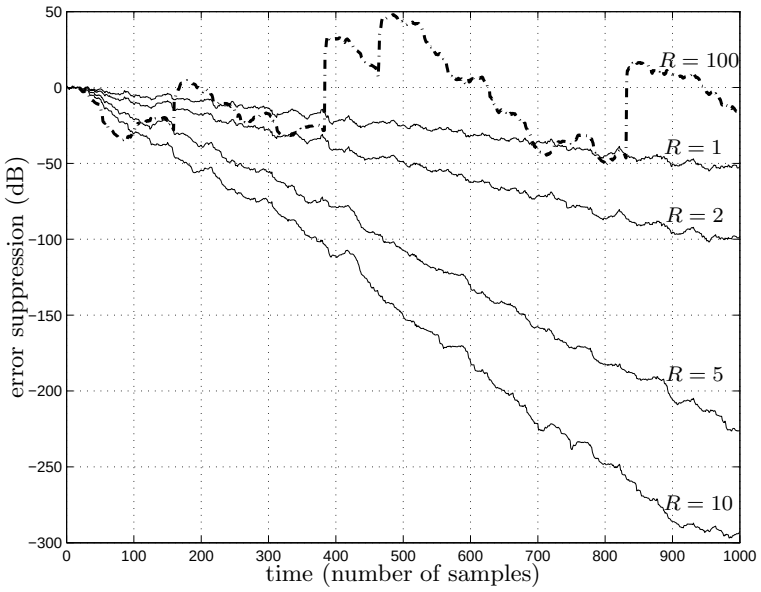


Fig. 4. Subband–normalized constrained PBFDRAP : the short–time energy of the a–priori error output is plotted as a function of time. It appears that if R is increased the subband–normalized constrained PBFDRAP becomes unstable.

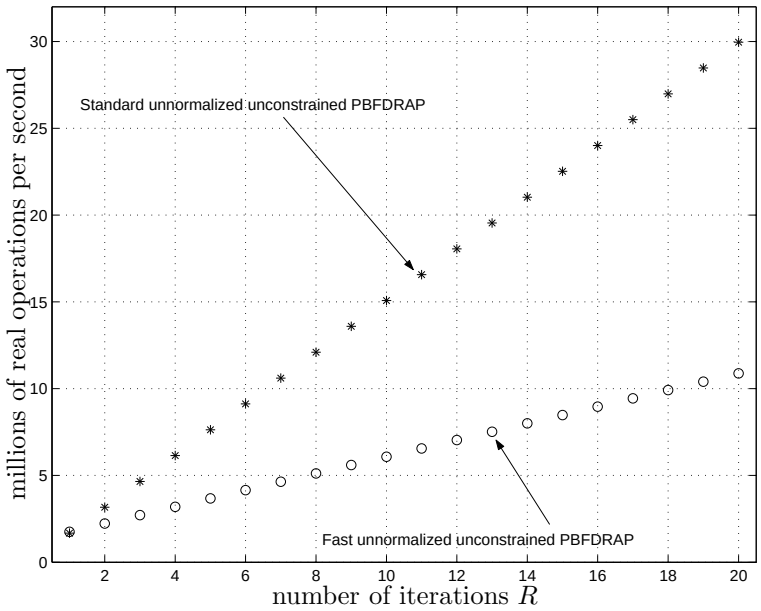


Fig. 5. Cost standard unnormalized unconstrained PBFDRAP (table I) vs. fast unnormalized unconstrained PBFDRAP (table IV) for a realistic parameter setting ($M = 256$, $P = L = 128$, $N = 1024$, $f_s = 8000$ Hz). It appears that the fast PBFDRAP outperforms the standard PBFDRAP, except for $R = 1$ for which it is slightly more expensive. Iterating twice with the standard approach is as expensive as iterating 4 times with the fast algorithm.

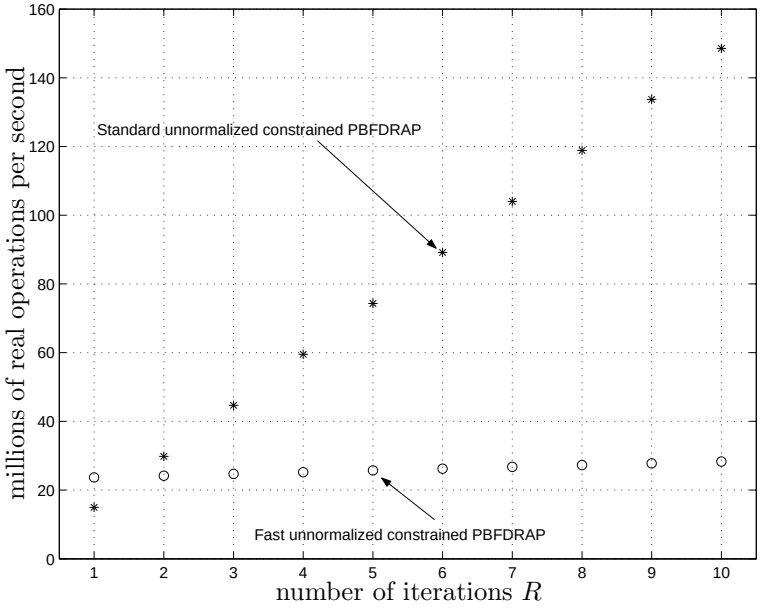


Fig. 6. Cost standard unnormalized constrained PBFDRAP (table I) vs. fast unnormalized constrained PBFDRAP (table V) for a realistic parameter setting ($M = 64$, $P = L = 32$, $N = 1024$, $f_s = 8000$ Hz). It appears that the fast PBFDRAP outperforms the standard PBFDRAP, except for $R = 1$.

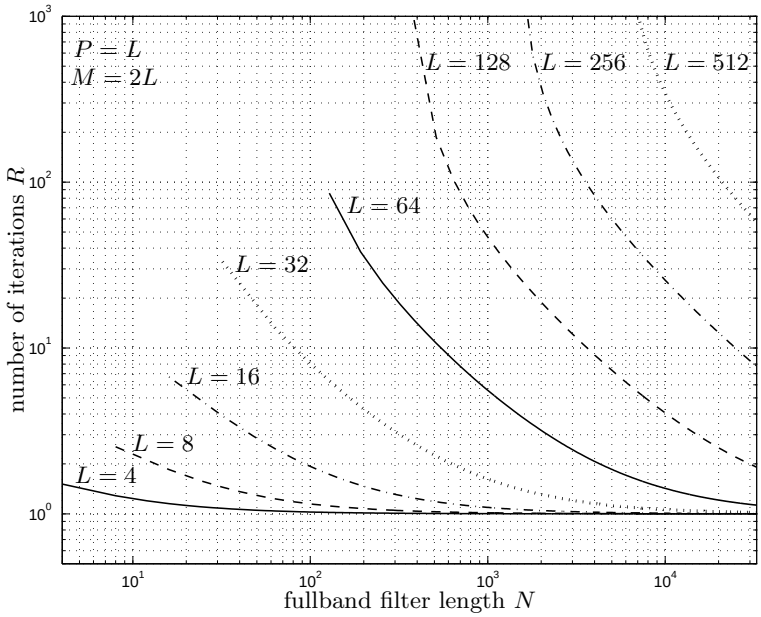


Fig. 7. Number of iteration steps R for which the standard implementation of the unnormalized constrained PBFDRAP (table I) is as expensive as the fast implementation of table V. For a parameter set (N, R) above the curve, the fast algorithm leads to the lowest cost. Observe that for small values of L, and N and R sufficiently large, the algorithm of table V leads to the lowest cost.

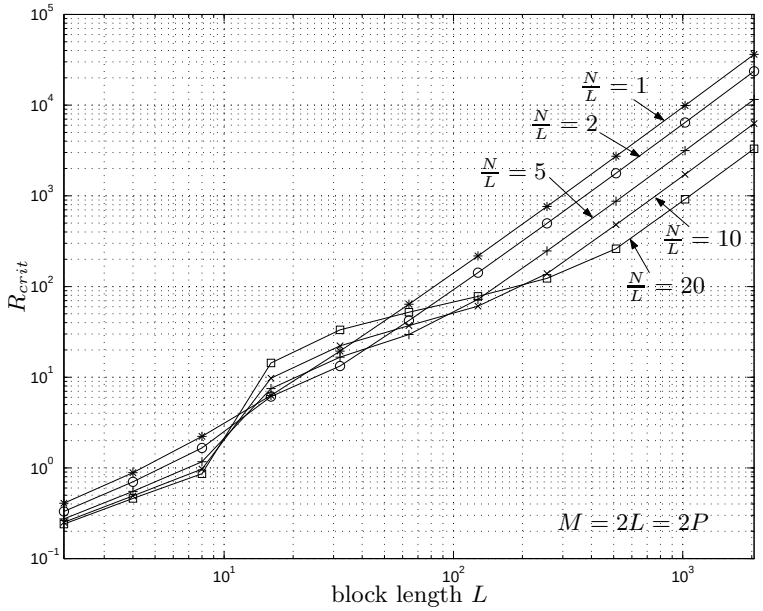


Fig. 8. The number of iteration steps R_{crit} for which the unnormalized constrained PBFDRAP is as expensive as the PRA, is plotted for different parameters settings. For large block lengths, the iterated PBFDRAP is a cheaper alternative for the PRA.

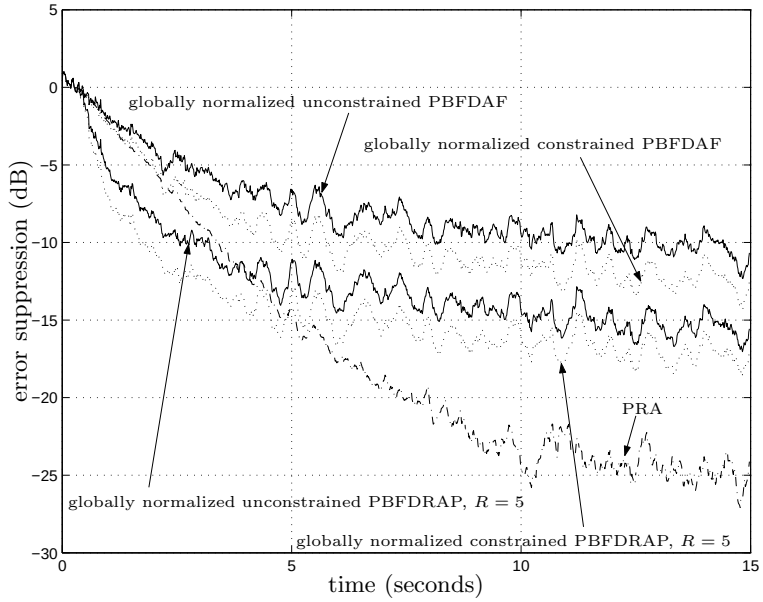


Fig. 9. An unknown stationary acoustic path was identified using the PRA algorithm and the globally normalized PBFDRAP, varying the type of updating (unconstrained, constrained) and the number of iterations steps R. The following parameters were applied: $L = P = 128$, $M = 256$, $N = 1024$. For each of the adaptive filters the short–time energy of the a–priori error output e is shown as a function of time. It is clear that by increasing the number of iteration steps R the error suppression can be improved.

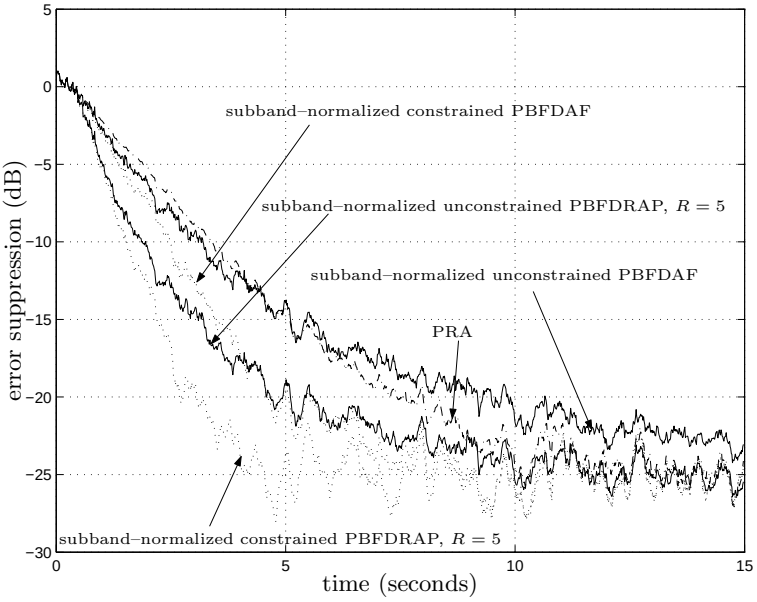


Fig. 10. An unknown stationary acoustic path was identified using the PRA algorithm and the subband–normalized PBFDRAP, varying the type of updating (unconstrained, constrained) and the number of iterations steps R. The following parameters were applied: $L = P = 128$, $M = 256$, $N = 1024$. For each of the adaptive filters the short–time energy of the a–priori error output e is shown as a function of time. The performance of the PBFDRAP can be improved by increasing the number of iteration steps R. As the input signal is colored, the subband–normalized PBFDRAP converges faster than the globally normalized PBFDRAP (cf. figure 9). Although stability is not guaranteed for $R \to \infty$, the subband–normalized constrained PBFDRAP offers the best performance for realistic values of R.

for each block of L samples do $\{$

$$\mathbf{X}_p^{(n)} \overset{\forall p}{=} \operatorname{diag}\left\{ \mathbf{F} \begin{bmatrix} x((n+1)L - pP - M + 1) \\ \vdots \\ x((n+1)L - pP) \end{bmatrix} \right\} \qquad 2M\log_2 M - 4M + 6 \text{ (assuming } \mathbf{X}_p^{(n)} = \mathbf{X}_0^{(n-pP/L)})$$

$$\mathbf{d}^{(n)} = \begin{bmatrix} \mathbf{0} \\ \mathbf{d}_n \end{bmatrix} \begin{matrix} \updownarrow M-L \\ \updownarrow L \end{matrix}, \qquad \mathbf{d}_n = \begin{bmatrix} d[nL+1] \\ \vdots \\ d[(n+1)L] \end{bmatrix} \Big\uparrow \Big\downarrow L$$

for $r = 1$ to R do $\{$

$$\mathbf{y}^{(n,r)} = \begin{bmatrix} \mathbf{0}_{M-L} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{N}{P}-1} \mathbf{X}_p^{(n)} \mathbf{w}_p^{(n,r)} \qquad R(2M\log_2 M + 4\tfrac{MN}{P} + 8\tfrac{N}{P} - M - 2)$$

$$\mathbf{e}^{(n,r)} = \mathbf{d}^{(n)} - \mathbf{y}^{(n,r)} \qquad RL$$

$$\mathbf{w}_p^{(n,r+1)} \overset{\forall p}{=} \mathbf{w}_p^{(n,r)} + \mathbf{G}\boldsymbol{\Delta}\mathbf{X}_p^{(n)*}\mathbf{F}\mathbf{e}^{(n,r)} \qquad R(2M\log_2 M - 3M + 8 + 4\tfrac{MN}{P} + 8\tfrac{N}{P})$$
(if $\mathbf{G} = \mathbf{I}_M$)

$$R((2M + 4\tfrac{MN}{P})\log_2 M - 3M + 8 + 14\tfrac{N}{P})$$
(if $\mathbf{G} = \mathbf{F}\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\mathbf{F}^{-1}$)

$\}$

$$\mathbf{w}_p^{(n+1,1)} \overset{\forall p}{=} \mathbf{w}_p^{(n,R+1)}$$

$\}$

$$p = 0 : \tfrac{N}{P} - 1 \qquad M \geqslant P + L - 1$$
$$\mathbf{F}(p,q) = e^{-\frac{j2\pi pq}{M}} \qquad \boldsymbol{\Delta} = \operatorname{diag}\{\boldsymbol{\mu}_s^{(n)}\}$$

unconstrained		constrained			PBFDAF	PBFDRAP
$\mathbf{G} = \mathbf{I}_M$	$\mathbf{G} = \mathbf{F}$	$\begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{M-P} \end{bmatrix}$		$\mathbf{F}^{-1}$	$R = 1$	$R > 1$

TABLE I

Standard implementation of the PBFDRAP

for each block of L samples do $\{$

$$\mathbf{X}_p^{(n)} \overset{\forall p}{=} \operatorname{diag}\left\{ \mathbf{F} \begin{bmatrix} x((n+1)L - pP - M + 1) \\ \vdots \\ x((n+1)L - pP) \end{bmatrix} \right\} \qquad \begin{array}{l} 2M\log_2 M - 4M + 6 \\ (\text{assuming } \mathbf{X}_p^{(n)} = \mathbf{X}_0^{(n-pP/L)}) \end{array}$$

$$\mathbf{B}_n = \mathbf{I}_M - \begin{bmatrix} \mathbf{0}_{M-L} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{N}{P}-1} \mathbf{X}_p^{(n)} \mathbf{G}\boldsymbol{\Delta} \mathbf{X}_p^{(n)*} \mathbf{F} \qquad \begin{array}{l} 2M\log_2 M + 2\frac{MN}{P} + 4\frac{N}{P} + 1 \\ (\text{if } \mathbf{G} = \mathbf{I}_M,\ \boldsymbol{\Delta} = \mu \mathbf{I}_M) \end{array}$$

$$\mathbf{y}^{(n,1)} = \begin{bmatrix} \mathbf{0}_{M-L} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{N}{P}-1} \mathbf{X}_p^{(n)} \mathbf{w}_p^{(n)} \qquad 2M\log_2 M + 4\frac{MN}{P} + 8\frac{N}{P} - M - 2$$

$$\mathbf{d}^{(n)} = \begin{bmatrix} \mathbf{0} \\ \mathbf{d}_n \end{bmatrix} \begin{matrix} \updownarrow M-L \\ \updownarrow L \end{matrix}, \qquad \mathbf{d}_n = \begin{bmatrix} d[nL+1] \\ \vdots \\ d[(n+1)L] \end{bmatrix} \begin{matrix} \uparrow \\ \\ \downarrow \end{matrix} L$$

$$\mathbf{e}^{(n,1)} = \mathbf{d}^{(n)} - \mathbf{y}^{(n,1)} \qquad L$$

$$\mathbf{c}^{(n,1)} = \mathbf{e}^{(n,1)}$$

for $r = 2$ to R do $\{$

$$\mathbf{e}^{(n,r)} = \mathbf{B}_n \mathbf{e}^{(n,r-1)} \qquad (R-1)(2L^2 - L)$$

$$\mathbf{c}^{(n,r)} = \mathbf{c}^{(n,r-1)} + \mathbf{e}^{(n,r)} \qquad (R-1)L$$

$\}$

$$\mathbf{w}_p^{(n+1)} \overset{\forall p}{=} \mathbf{w}_p^{(n)} + \mathbf{G}\boldsymbol{\Delta} \mathbf{X}_p^{(n)*} \mathbf{F} \mathbf{c}^{(n,R)} \qquad \begin{array}{l} 2M\log_2 M - 3M + 8 + 4\frac{MN}{P} + 8\frac{N}{P} \\ (\text{if } \mathbf{G} = \mathbf{I}_M) \end{array}$$

$\}$

$$\begin{array}{rclcrcl} p & = & 0 : \frac{N}{P} - 1 & & M & \geqslant & P + L - 1 \\ \mathbf{F}(p,q) & = & e^{-\frac{j2\pi pq}{M}} & & \boldsymbol{\Delta} & = & \operatorname{diag}\{\boldsymbol{\mu}_s^{(n)}\} \end{array}$$

unconstrained		constrained			PBFDAF	PBFDRAP
$\mathbf{G} = \mathbf{I}_M$	$\mathbf{G} = \mathbf{F}$	$\begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{M-P} \end{bmatrix}$		$\mathbf{F}^{-1}$	$R = 1$	$R > 1$

TABLE II

FAST PBFDRAP, VERSION 1

`for each block of` L `samples do` $\{$

$$\mathbf{X}_p^{(n)} \stackrel{\forall p}{=} \operatorname{diag}\left\{\mathbf{F}\begin{bmatrix} x((n+1)L-pP-M+1) \\ \vdots \\ x((n+1)L-pP) \end{bmatrix}\right\} \qquad 2M\log_2 M - 4M + 6$$
$$(\text{assuming } \mathbf{X}_p^{(n)} = \mathbf{X}_0^{(n-pP/L)})$$

$$\mathbf{D}_n = \sum_{p=0}^{\frac{N}{P}-1} \mathbf{X}_p^{(n)}\mathbf{G}\boldsymbol{\Delta}\mathbf{X}_p^{(n)*} \qquad 2\tfrac{MN}{P} + 4\tfrac{N}{P}$$
$$(\text{if } \mathbf{G} = \mathbf{I}_M,\ \boldsymbol{\Delta} = \mu\mathbf{I}_M)$$

$$\mathbf{y}^{(n,1)} = \begin{bmatrix} \mathbf{0}_{M-L} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix}\mathbf{F}^{-1}\sum_{p=0}^{\frac{N}{P}-1}\mathbf{X}_p^{(n)}\mathbf{w}_p^{(n)} \qquad 2M\log_2 M + 4\tfrac{MN}{P} + 8\tfrac{N}{P} - M - 2$$

$$\mathbf{d}^{(n)} = \begin{bmatrix} \mathbf{0} \\ \mathbf{d}_n \end{bmatrix}\begin{matrix}\updownarrow M-L \\ \updownarrow L\end{matrix}, \qquad \mathbf{d}_n = \begin{bmatrix} d[nL+1] \\ \vdots \\ d[(n+1)L] \end{bmatrix}\Big\uparrow\Big\downarrow L$$

$$\mathbf{e}^{(n,1)} = \mathbf{d}^{(n)} - \mathbf{y}^{(n,1)} \qquad L$$

$$\mathbf{c}^{(n,1)} = \mathbf{e}^{(n,1)}$$

`for` $r = 2$ `to` R `do` $\{$

$$\underline{\mathbf{e}}^{(n,r-1)} = \mathbf{F}\mathbf{e}^{(n,r-1)} \qquad (R-1)(2M\log_2 M - 4M + 6)$$

$$\mathbf{a}^{(n,r)} = \begin{bmatrix} \mathbf{0}_{M-L} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix}\mathbf{F}^{-1}\mathbf{D}_n\underline{\mathbf{e}}^{(n,r-1)} \qquad (R-1)(2M\log_2 M + M + 2)$$
$$(\text{if } \mathbf{G} = \mathbf{I}_M)$$

$$\mathbf{e}^{(n,r)} = \mathbf{e}^{(n,r-1)} - \mathbf{a}^{(n,r)} \qquad (R-1)L$$

$$\mathbf{c}^{(n,r)} = \mathbf{c}^{(n,r-1)} + \mathbf{e}^{(n,r)} \qquad (R-1)L$$

$\}$

$$\mathbf{w}_p^{(n+1)} \stackrel{\forall p}{=} \mathbf{w}_p^{(n)} + \mathbf{G}\boldsymbol{\Delta}\mathbf{X}_p^{(n)*}\mathbf{F}\mathbf{c}^{(n,R)} \qquad 2M\log_2 M - 3M + 8 + 4\tfrac{MN}{P} + 8\tfrac{N}{P}$$
$$(\text{if } \mathbf{G} = \mathbf{I}_M)$$

$\}$

$$p = 0 : \tfrac{N}{P} - 1 \qquad M \geqslant P + L - 1$$
$$\mathbf{F}(p,q) = e^{-\frac{j2\pi pq}{M}} \qquad \boldsymbol{\Delta} = \operatorname{diag}\{\boldsymbol{\mu}_s^{(n)}\}$$

unconstrained		constrained		PBFDAF	PBFDRAP
$\mathbf{G} = \mathbf{I}_M$	$\mathbf{G} = \mathbf{F}$	$\begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{M-P} \end{bmatrix}$	$\mathbf{F}^{-1}$	$R = 1$	$R > 1$

TABLE III

FAST PBFDRAP, VERSION 2

for each block of L samples do $\{$

$$\mathbf{X}_p^{(n)} \overset{\forall p}{=} \mathrm{diag}\left\{\mathbf{F}\begin{bmatrix} x((n+1)L-pP-M+1) \\ \vdots \\ x((n+1)L-pP) \end{bmatrix}\right\} \qquad \begin{array}{l} 2M\log_2 M - 4M + 6 \\ (\text{assuming } \mathbf{X}_p^{(n)} = \mathbf{X}_0^{(n-p)}) \end{array}$$

$$\bar{\mathbf{D}}_n = \bar{\mathbf{D}}_{n-1} + \mathbf{X}_0^{(n)}\mathbf{G}\mathbf{X}_0^{(n)^*} - \mathbf{X}_{\frac{N}{P}}^{(n)}\mathbf{G}\mathbf{X}_{\frac{N}{P}}^{(n)^*} \qquad \tfrac{9}{2}M + 9$$

$$\mathbf{D}_n = \bar{\mathbf{D}}_n\boldsymbol{\Delta} \qquad (\text{if } \mathbf{G} = \mathbf{I}_M \text{ and } \boldsymbol{\Delta} = \mu\mathbf{I}_M)$$

$$\mathbf{y}^{(n,1)} = \begin{bmatrix} \mathbf{0}_{M-L} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix}\mathbf{F}^{-1}\sum_{p=0}^{\frac{N}{P}-1}\mathbf{X}_p^{(n)}\mathbf{w}_p^{(n)} \qquad 2M\log_2 M + 4\tfrac{MN}{P} + 8\tfrac{N}{P} - M - 2$$

$$\mathbf{d}^{(n)} = \begin{bmatrix} \mathbf{0} \\ \mathbf{d}_n \end{bmatrix}\begin{matrix} \updownarrow M-L \\ \updownarrow L \end{matrix}, \qquad \mathbf{d}_n = \begin{bmatrix} d[nL+1] \\ \vdots \\ d[(n+1)L] \end{bmatrix}\Big\uparrow\Big\downarrow L$$

$$\mathbf{e}^{(n,1)} = \mathbf{d}^{(n)} - \mathbf{y}^{(n,1)} \qquad L$$

$$\mathbf{c}^{(n,1)} = \mathbf{e}^{(n,1)}$$

for $r = 2$ to R do $\{$

$$\underline{\mathbf{e}}^{(n,r-1)} = \mathbf{F}\mathbf{e}^{(n,r-1)} \qquad (R-1)(2M\log_2 M - 4M + 6)$$

$$\mathbf{a}^{(n,r)} = \begin{bmatrix} \mathbf{0}_{M-L} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix}\mathbf{F}^{-1}\mathbf{D}_n\underline{\mathbf{e}}^{(n,r-1)} \qquad \begin{array}{l}(R-1)(2M\log_2 M + M + 2) \\ (\text{if } \mathbf{G} = \mathbf{I}_M)\end{array}$$

$$\mathbf{e}^{(n,r)} = \mathbf{e}^{(n,r-1)} - \mathbf{a}^{(n,r)} \qquad (R-1)L$$

$$\mathbf{c}^{(n,r)} = \mathbf{c}^{(n,r-1)} + \mathbf{e}^{(n,r)} \qquad (R-1)L$$

$\}$

$$\mathbf{w}_p^{(n+1)} \overset{\forall p}{=} \mathbf{w}_p^{(n)} + \mathbf{G}\boldsymbol{\Delta}\mathbf{X}_p^{(n)^*}\mathbf{F}\mathbf{c}^{(n,R)} \qquad \begin{array}{l} 2M\log_2 M - 3M + 8 + 4\tfrac{MN}{P} + 8\tfrac{N}{P} \\ (\text{if } \mathbf{G} = \mathbf{I}_M) \end{array}$$

$\}$

$$\begin{array}{rclcrcl} p & = & 0 : \frac{N}{P} - 1 & & M & \geqslant & P + L - 1 \\ \mathbf{F}(p,q) & = & e^{-\frac{j2\pi pq}{M}} & & \boldsymbol{\Delta} & = & \mathrm{diag}\{\boldsymbol{\mu}_s^{(n)}\} \\ P & = & L & & & & \end{array}$$

unconstrained		constrained			PBFDAF	PBFDRAP
$\mathbf{G} = \mathbf{I}_M$	$\mathbf{G} = \mathbf{F}$	$\begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{M-P} \end{bmatrix}$		$\mathbf{F}^{-1}$	$R = 1$	$R > 1$

TABLE IV

FAST PBFDRAP, VERSION 3 : $P = L$

`for each block of` L `samples do` $\{$	
$\mathbf{X}_p^{(n)} \stackrel{\forall p}{=} \mathrm{diag}\left\{ \mathbf{F} \begin{bmatrix} x((n+1)L-pP-M+1) \\ \vdots \\ x((n+1)L-pP) \end{bmatrix} \right\}$	$2M\log_2 M - 4M + 6$ (assuming $\mathbf{X}_p^{(n)} = \mathbf{X}_0^{(n-pP/L)}$)
$\mathbf{X}_{n,p} \stackrel{\forall p}{=} \begin{bmatrix} x[nL-pP+1] & \dots & x[(n+1)L-pP] \\ \vdots & \ddots & \vdots \\ x[nL-pP-P+2] & \dots & x[(n+1)L-pP-P+1] \end{bmatrix}$	
$\bar{\mathbf{B}}_n = \bar{\mathbf{B}}_{n-1} + \mathbf{X}_{n,0}^T\mathbf{X}_{n,0} - \mathbf{X}_{n,\frac{N}{P}}^T\mathbf{X}_{n,\frac{N}{P}}$	$L^3 + \frac{3}{2}L^2 + \frac{L}{2}$ (assuming $P = L$)
$\hat{\mathbf{B}}_n = \mathbf{I}_L - \mu_n\bar{\mathbf{B}}_n$	$\frac{1}{2}L^2 + \frac{3}{2}L$
$\mathbf{y}^{(n,1)} = \begin{bmatrix} \mathbf{0}_{M-L} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \sum_{p=0}^{\frac{N}{P}-1} \mathbf{X}_p^{(n)}\mathbf{w}_p^{(n)}$	$2M\log_2 M + 4\frac{MN}{P} + 8\frac{N}{P} - M - 2$
$\mathbf{d}^{(n)} = \begin{bmatrix} \mathbf{0} \\ \mathbf{d}_n \end{bmatrix} \begin{matrix} \updownarrow M-L \\ \updownarrow L \end{matrix}$, $\mathbf{d}_n = \begin{bmatrix} d[nL+1] \\ \vdots \\ d[(n+1)L] \end{bmatrix} \Big\updownarrow L$	
$\mathbf{e}^{(n,1)} = \mathbf{d}^{(n)} - \mathbf{y}^{(n,1)}$	L
$\mathbf{c}^{(n,1)} = \mathbf{e}^{(n,1)}$	
$\mathbf{e}_n^{(1)} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_L \end{bmatrix} \mathbf{e}^{(n,1)}$	
`for` $r = 2$ `to` R `do` $\{$	
$\mathbf{e}_n^{(r)} = \hat{\mathbf{B}}_n\mathbf{e}_n^{(r-1)}$	$(R-1)(2L^2 - L)$
$\mathbf{c}^{(n,r)} = \mathbf{c}^{(n,r-1)} + \begin{bmatrix} \mathbf{0} \\ \mathbf{e}_n^{(r)} \end{bmatrix}$	$(R-1)L$
$\}$	
$\mathbf{w}_p^{(n+1)} \stackrel{\forall p}{=} \mathbf{w}_p^{(n)} + \mu_n\mathbf{F} \begin{bmatrix} \mathbf{I}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{M-P} \end{bmatrix} \mathbf{F}^{-1}\mathbf{X}_p^{(n)*}\mathbf{F}\mathbf{c}^{(n,R)}$	$(2M + 4\frac{MN}{P})\log_2 M - 4M + 6 + L + 14\frac{N}{P}$
$\}$	

$$\begin{array}{rclcrcl} p & = & 0 : \frac{N}{P} - 1 & & M & \geqslant & P + L - 1 \\ \mathbf{F}(p,q) & = & e^{-\frac{j2\pi pq}{M}} & & P & = & L \end{array}$$

TABLE V

FAST UNNORMALIZED CONSTRAINED PBFDRAP : $P = L$

for each block of L samples do $\{$

$$\mathbf{X}_n^l \overset{l=0:\frac{N}{L}}{=} \begin{bmatrix} x[nL-lL+1] & \dots & x[nL+L-lL] \\ \vdots & \ddots & \vdots \\ x[nL-L+2-lL] & \dots & x[nL-lL+1] \end{bmatrix}$$

$$\mathbf{X}_n = \begin{bmatrix} \left(\mathbf{X}_n^0\right)^T & \left(\mathbf{X}_n^1\right)^T & \dots & \left(\mathbf{X}_n^{N/L-1}\right)^T \end{bmatrix}^T$$

$$\mathbf{d}_n = \begin{bmatrix} d[nL+1] \\ \vdots \\ d[(n+1)L] \end{bmatrix}$$

$\mathbf{y}_n = \mathbf{X}_n^T \hat{\mathbf{w}}_n$	$2LN - L$
$\mathbf{e}_n = \mathbf{d}_n - \mathbf{y}_n$	L
$\mathbf{M}_n = \mathbf{M}_{n-1} + \left(\mathbf{X}_n^0\right)^T \mathbf{X}_n^0 - \left(\mathbf{X}_n^{N/L}\right)^T \mathbf{X}_n^{N/L}$	$L^3 + \frac{3}{2}L^2 + \frac{L}{2}$
$\mathbf{p}_n = \left(\mathbf{M}_n + \delta \mathbf{I}_L\right)^{-1} \mathbf{e}_n$	$\frac{2}{3}L^3 + L^2 + L$
$\hat{\mathbf{w}}_{n+1} = \hat{\mathbf{w}}_n + \mu_n \mathbf{X}_n \mathbf{p}_n$	$2LN + L$

$\}$

TABLE VI

PARTIAL RANK ALGORITHM (PRA)

globally normalized unconstrained PBFDAF	1.76 MOPS
globally normalized constrained PBFDAF	5.35 MOPS
globally normalized unconstrained PBFDRAP ($R = 5$)	3.76 MOPS
globally normalized constrained PBFDRAP ($R = 5$)	25.65 MOPS
subband–normalized unconstrained PBFDAF	1.81 MOPS
subband–normalized constrained PBFDAF	5.40 MOPS
subband–normalized unconstrained PBFDRAP ($R = 5$)	3.80 MOPS
subband–normalized constrained PBFDRAP ($R = 5$)	25.70 MOPS
PRA	253.80 MOPS

TABLE VII

THE ALGORITHMIC COMPLEXITY IN MILLIONS OF REAL OPERATIONS PER SECOND FOR EACH OF THE ADAPTIVE FILTERS SHOWN IN FIGURES 9 AND 10 ($M = 256$, $P = L = 128$, $N = 1024$, SAMPLING RATE $f_s = 8000$ Hz). IT APPEARS THAT THE PBFDRAP ALGORITHM CAN BE CONSIDERED AS A LOW–COST ALTERNATIVE FOR THE PRA FOR LARGE BLOCK LENGTHS L.