An Alternative Complexity Reduction Method for Partitioned-Block Frequency-Domain Adaptive Filters

María Luis Valero, *Student Member, IEEE*, Edwin Mabande, and Emanuël A. P. Habets, *Senior Member, IEEE*

***Abstract*—Partitioned-block frequency-domain adaptive filters provide lower algorithmic complexity than their time-domain counterparts. However, for certain applications, it is desirable to reduce complexity even further. This can be achieved by reducing the complexity of the constraint, or linearization, of the circular correlations. Existing methods are either suboptimal or are not sufficiently flexible in terms of the design parameters. Therefore, an alternative complexity reduction method is proposed, which achieves near-optimal performance and which, in addition, is flexible in terms of the design parameters, as these can be arbitrarily chosen.**

***Index Terms*—Acoustic echo cancellation, frequency-domain adaptive filters, system identification, time-frequency analysis.**

I. Introduction

In hands-free communication scenarios, the far-end signal played by a loudspeaker is propagated through the near-end room and acquired by a microphone. As a consequence, the microphone signal not only contains the desired near-end speech and background noise but also acoustic echo. The most commonly used technique to cope with electro-acoustic coupling is acoustic echo cancellation (AEC) [1], [2]. Acoustic echo cancellers use adaptive filtering algorithms [3], [4] to identify the acoustic echo path, which is necessary to estimate the acoustic echo signal. The estimated echo is then subtracted from the microphone signal prior to transmission.

Frequency-domain adaptive filters (FDAFs) are algorithmically less complex than their time-domain counterparts. This is a consequence of performing convolutions in the frequency domain, which are equivalent to circular convolutions in the time domain. However, the complexity reduction comes at the cost of higher algorithmic delay. In order to reduce the delay, the partitioned-block FDAF (PB-FDAF) was introduced in [5].

For some applications, it is desirable to further reduce the complexity. This can be achieved by reducing the complexity of the constraint, or linearization, of the circular convolutions, which, given a PB-FDAF, has to be performed once per block for each frame. In the past, several methods were proposed to reduce the complexity of the constraining operation [6]–[11]. In [6], the unconstrained frequency-domain least mean squares (UFLMS) algorithm was proposed, which omits the constraining operation altogether. The UFLMS was extended to a PB-based architecture in [7], in which the alternative unconstrained gradient update method was proposed. The constraining method described in [7] applies a constraint to only one block each frame, while the alternated constrained (AC) gradient update method described in [9] also eliminates the accumulated wrap-around errors between constraints. Finally, the modified alternated constrained (MAC) gradient update method proposed in [10] uses a modified constraining window to reject part of the wrap-around errors. The raised-sine window used in [10] also modifies the linear components, which are compensated afterwards. In this letter, we propose a generalization of the AC gradient update method described in [9]. The proposed method uses a modified constraining window as in [10]. However, in contrast to [10], the proposed method not only eliminates the wrap-around errors completely, but also perfectly corrects the modified linear components. Additionally, the proposed method allows the use of an arbitrary constraining window approximation as well as an arbitrary frame overlap.

Manuscript received November 20, 2015; revised February 08, 2016; accepted March 09, 2016. Date of publication March 18, 2016; date of current version April 12, 2016. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Muhammad Zubair Ikram.

The authors are with the International Audio Laboratories Erlangen (a joint institution of the Friedrich-Alexander-University Erlangen-Nürnberg and Fraunhofer IIS), 91058 Erlangen, Germany (e-mail: maria.luis.valero@audiolabs-erlangen.de; emanuel.habets@audiolabs-erlangen.de).

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2016.2543964

II. Problem Formulation

In a loudspeaker-enclosure-microphone environment, the microphone signal at discrete time index n is defined as

$$y(n) = x(n) * h(n) + s(n) + u(n), \tag{1}$$

where $x(n)$ denotes the loudspeaker signal, $s(n)$ is the near-end speech, $u(n)$ is the background noise, and $h(n)$ is the acoustic echo path. In the following, it is assumed that $h(n)$ can be modeled by a finite impulse response (FIR) filter of length L. The adaptive filters in AEC are driven by the error signal, which is the output of the echo canceller, given by

$$e(n) = y(n) - x(n) * \hat{h}(n), \tag{2}$$

where $\hat{h}(n)$ denotes the estimated echo path. In order to obtain a PB formulation of (2), the acoustic echo path is partitioned into B blocks of length $N = \lceil L/B \rceil$. The problem is formulated using the overlap-save method as described in [3]. Thus, the bth partition of $h(n)$ and $x(n)$ are

$$\mathbf{h}_b(l) = [h_l(bN), \ldots, h_l(bN + N - 1)]^{\mathrm{T}} \tag{3}$$

$$\text{and } \mathbf{x}_b(l) = [x(lR - bN - M + 1), \ldots, x(lR - bN)]^{\mathrm{T}}, \tag{4}$$

where the frame index is denoted by l, $M \geq N$ is the input signal frame length, and R is the frame shift. In the time-frequency

domain, (3) and (4) take the form

$$\mathbf{H}_b(l) = \underline{\mathbf{F}} \begin{bmatrix} \mathbf{h}_b(l) \\ \mathbf{0}_{V\times 1} \end{bmatrix} \text{ and } \underline{\mathbf{X}}_b(l) = \text{diag}\{\underline{\mathbf{F}}\mathbf{x}_b(l)\}, \quad (5)$$

respectively. In (5), $V = M - N$ is the length of the zero padding and $\underline{\mathbf{F}}$ is the $M \times M$ discrete Fourier transform (DFT) matrix, whose elements are $\text{F}(p,q) = \exp(-j2\pi pq/M)$, with $j = \sqrt{-1}$. Moreover, the frequency index is omitted for brevity, matrices are denoted by $\underline{\mathbf{A}}$, $\mathbf{A} = \text{diag}\{\underline{\mathbf{A}}\}$ generates a column vector with the main diagonal elements of $\underline{\mathbf{A}}$, and $\text{diag}\{\mathbf{A}\}$ generates a diagonal matrix with the elements of $\mathbf{A}$ on its main diagonal. The adaptive filters complexity is reduced by performing the convolutions in the DFT domain

$$\underline{\mathbf{X}}_b(l)\mathbf{H}_b(l) \overset{\text{DFT}}{\longleftrightarrow} \mathbf{x}_b(l) \circledM \mathbf{h}_b(l), \quad (6)$$

where $\circledM$ denotes circular convolution of length M, following the notation in [12]. The circular convolution is defined by

$$\mathbf{x}_b(l) \circledM \mathbf{h}_b(l) = \sum_{n=0}^{M-1} x_b(((l-n))_M) h_b(n), \quad (7)$$

where $((\cdot))_M$ denotes the modulo-M operator. In (7), only the last $M - N + 1$ coefficients coincide with the linear convolution, while the first $N - 1$ taps are the result of the wrap-around. As described in [3] and [12], the latter have to be removed to assure the best possible performance of the adaptive algorithm. In the following, the signal partitions are defined to be of length V samples. Consequently, the error signal in the time domain, using a PB-based formulation and compact matrix notation, is

$$\begin{bmatrix} \mathbf{0}_{N\times 1} \\ \mathbf{e}(l) \end{bmatrix} = \text{diag}\{\mathbf{w}\}\underline{\mathbf{F}}^{-1} \left(\mathbf{Y}(l) - \sum_{b=0}^{B-1} \underline{\mathbf{X}}_b(l)\widehat{\mathbf{H}}_b(l) \right), \quad (8)$$

where $\mathbf{Y}(l) = \underline{\mathbf{F}}[\mathbf{0}_{1\times N}, \mathbf{y}^{\text{T}}(l)]^{\text{T}}$ is the microphone signal in the frequency domain and $\mathbf{w} = [\mathbf{0}_{1\times N}, \mathbf{1}_{1\times V}]$ is the convolution-constraining window that selects the linear components. Analogous to (3) and (4), we define

$$\mathbf{z}(l) = [z(lR - V + 1), \ldots, z(lR)]^{\text{T}}, \text{ with } \mathbf{z} \in \{\mathbf{e}, \mathbf{y}\},$$

where $V \geq R$, which provides the relation between the partition length, N, and the frame shift, R. Finally, the error signal in the frequency domain is $\mathbf{E}(l) = \underline{\mathbf{F}}[\mathbf{0}_{1\times N}, \mathbf{e}^{\text{T}}(l)]^{\text{T}}$. A generalized expression for the update of the bth filter partition estimate using a PB-FDAF is

$$\widehat{\mathbf{H}}_b(l+1) = \widehat{\mathbf{H}}_b(l) + \underline{\mathbf{G}} \underbrace{\underline{\mathbf{M}}_b(l)\underline{\mathbf{X}}_b^{\text{H}}(l)\mathbf{E}(l)}_{\Delta\widehat{\mathbf{H}}_b(l)}, \quad (9)$$

where the step-size matrix $\underline{\mathbf{M}}_b(l)$ depends on the employed adaptive algorithm and the superscript $(\cdot)^{\text{H}}$ denotes Hermitian transpose. In (9), the gradient update, $\Delta\widehat{\mathbf{H}}_b(l)$, is obtained by performing a correlation in the DFT domain and the constraining matrix for a circular correlation, denoted as $\underline{\mathbf{G}}$, is

$$\underline{\mathbf{G}} = \underline{\mathbf{F}}\,\text{diag}\{\mathbf{g}\}\underline{\mathbf{F}}^{-1}, \quad (10)$$

where $\mathbf{g} = [\mathbf{1}_{1\times N}, \mathbf{0}_{1\times V}]$ is the time-domain correlation-constraining window. It must be emphasized that if $\underline{\mathbf{G}}$ is not modified, (9) is equivalent to

$$\widehat{\mathbf{H}}_b(l+1) = \underline{\mathbf{G}} \left[\widehat{\mathbf{H}}_b(l) + \Delta\widehat{\mathbf{H}}_b(l) \right]. \quad (11)$$

It must be noted that (9) removes the wrap-around errors before adaptation, while (11) copes with them after adaptation.

In practice, an implementation with fast Fourier transforms (FFTs) and windowing in the time domain is preferred to computationally expensive $M \times M$ matrix multiplications. This implies performing two FFTs per frame and per block to obtain the filter coefficients in (9). Hereafter, it is possible to further reduce the complexity by avoiding the frequency transforms and approximating the gradient-constraining operation in the frequency domain. In the next section, some of the previously proposed complexity reduction methods are described.

III. Previously Proposed Constraining Methods

The aim of the complexity reduction methods is to avoid two FFTs per block and, consequently, reduce the total algorithmic complexity. The unconstrained (UC) gradient update, as proposed in [6], omits the constraining operation altogether

$$\widehat{\mathbf{H}}_b(l+1) = \widehat{\mathbf{H}}_b(l) + \underline{\mathbf{G}}_{\text{uc}}\Delta\widehat{\mathbf{H}}_b(l), \quad (12)$$

where $\underline{\mathbf{G}}_{\text{uc}} = \underline{\mathbf{I}}$ is the $M \times M$ identity matrix. This produces the so-called wrap-around error [13], which increases with decreasing DFT length. Therefore, the error caused by omitting the constraining operation may be non-negligible in case of the PB-based implementation. In the following, two methods are described that reduce the computational complexity of a PB-FDAF algorithm while maintaining high performance.

A. Alternated Constrained Gradient Update

In [7], it is proposed to reduce the complexity by constraining only one block per frame using (9). In contrast, the AC gradient update proposed in [9] uses (11) instead, which eliminates the accumulated wrap-around errors. In [9], the filter partitions are adapted using (12). Afterwards, every P frames, one block is corrected, i.e.,

$$\widehat{\mathbf{H}}_c(l+1) = \underline{\mathbf{G}}\widehat{\mathbf{H}}_c(l+1), \quad (13)$$

where the index $c = ((l/P))_B$ if $l/P \in \mathbb{N}^0$. Hereafter, the frame interval between constraints P can be enlarged to further reduce the complexity [9], [10]. Yet, for long intervals, the performance of the adaptive algorithm deteriorates due to accumulated wrap-around errors between corrections in the filter updates.

B. Modified Alternated Constrained Gradient Update

The MAC gradient update method proposed in [10] uses a simplified correlation-constraining matrix to reduce the wrap-around errors. It was shown in [8] for $M = 2N$ that if N is large enough, all except the first pair of off-diagonals of $\underline{\mathbf{G}}$ can be neglected. The approximation of $\underline{\mathbf{G}}$ in [10], denoted as $\underline{\mathbf{G}}_{\text{s}}$, is designed based on these findings, with the requirement that $\mathbf{g}_{\text{s}} = \text{diag}\{\underline{\mathbf{F}}^{-1}\underline{\mathbf{G}}_{\text{s}}\underline{\mathbf{F}}\}$ has to be non-negative. This results in a raised-sine window in the frequency domain, whose elements are

$$G_{\text{s}}(k,k') = \begin{cases} -j/4, & k = ((k'-1))_M \\ 1/2, & k = k' \\ j/4, & k = ((k'+1))_M \end{cases}, \quad (14)$$

where k and k' denote the discrete frequency indices. This window is able to reduce the wrap-around errors, but the linear components are also modified and have to be corrected.

Method 1. MAC gradient update method

1: **for** $b = 0 : B-1$ **do**
2: $\quad \widehat{\mathbf{H}}_b(l+1) = \widehat{\mathbf{H}}_b(l) + \underline{\mathbf{G}}_{\text{s}} \Delta\widehat{\mathbf{H}}_b(l)$
3: **if** $l/P \in \mathbb{N}^0$ then $c = ((l/P))_B$ **then**
4: $\quad \widehat{\mathbf{H}}^{\text{w}}_c(l+1) = (\underline{\mathbf{I}} - \underline{\mathbf{G}})\widehat{\mathbf{H}}_c(l+1)$
5: $\quad \widehat{\mathbf{H}}_{c+1}(l+1) = \widehat{\mathbf{H}}_{c+1}(l+1) + \underline{\mathbf{J}}\underline{\mathbf{H}}_{\text{an}}\widehat{\mathbf{H}}^{\text{w}}_c(l+1)$
6: $\quad \widehat{\mathbf{H}}_{c-1}(l+1) = \widehat{\mathbf{H}}_{c-1}(l+1) + \underline{\mathbf{J}}(\underline{\mathbf{I}} - \underline{\mathbf{H}}_{\text{an}})\widehat{\mathbf{H}}^{\text{w}}_c(l+1)$
7: $\quad \widehat{\mathbf{H}}_c(l+1) = \underline{\mathbf{G}}\widehat{\mathbf{H}}_c(l+1)$

The MAC method is summarized in Method 1, where $\underline{\mathbf{J}}$ is a diagonal translation matrix, with $J(k,k) = (-1)^k$, and $\underline{\mathbf{H}}_{\text{an}}$ is the frequency-domain Hann window, both as defined in [10]. In the correction step, first the linear elements of the previous and subsequent blocks are compensated, and finally the wrap-around error of one block is deleted. It must be noted that, for the compensation, the wrap-around error is used and the first and last blocks are not completely compensated. Due to the window selection, for both adaptation and correction, this method is only valid for $N = M/2$.

IV. Alternative Complexity Reduction Method

Inspired by the AC gradient update in [9], an alternative constraining method is proposed that uses a better conditioned window, as in [10], to enhance the block filter updates. The goal is to generalize the AC method to use an arbitrary approximation of the constraining window and a flexible frame overlap. The proposed method, denoted as the enhanced alternated constrained (EAC) gradient update, is also performed in two steps: i) adaptation and ii) correction.

If an arbitrary constraining window $\underline{\mathbf{G}}_{\text{arb}}$ is applied during adaptation, the corrected filter update (13) is defined by

$$\widehat{\mathbf{H}}_c(l+1) = \underline{\mathbf{G}}\left[\widehat{\mathbf{H}}_c(l) + \underline{\mathbf{G}}_{\text{arb}}\Delta\widehat{\mathbf{H}}_c(l)\right]. \tag{15}$$

Thus, the linear components have to be compensated prior to correction. As the correction step is applied every P frames to one block, the right hand-side of (15) is equivalent to

$$\underline{\mathbf{G}}\left[\widehat{\mathbf{H}}_c(l-PB+1) + \sum_{p=0}^{PB-1} \underline{\mathbf{G}}_{\text{arb}}\Delta\widehat{\mathbf{H}}_c(l-p)\right], \tag{16}$$

which can also be written as

$$\underline{\mathbf{G}}\left[\widehat{\mathbf{H}}_c(l-PB+1) + \sum_{p=0}^{PB-1} \Delta\widehat{\mathbf{H}}_c(l-p) - \mathbf{\Lambda}_c\right],$$

where the subscript c denotes the block to be corrected and

$$\mathbf{\Lambda}_c = \sum_{p=0}^{PB-1}\left[\Delta\widehat{\mathbf{H}}_c(l-p) - \underline{\mathbf{G}}_{\text{arb}}\Delta\widehat{\mathbf{H}}_c(l-p)\right] \tag{17}$$

is the cumulative difference between the windowed and the UC gradient updates of the cth block. Finally, the correction step of the proposed method is summarized by

$$\widehat{\mathbf{H}}_c(l+1) = \underline{\mathbf{G}}\left[\widehat{\mathbf{H}}_c(l) + \underline{\mathbf{G}}_{\text{arb}}\Delta\widehat{\mathbf{H}}_c(l) + \mathbf{\Lambda}_c\right], \tag{18}$$

which is equivalent to (13) if $\underline{\mathbf{G}}_{\text{arb}} = \underline{\mathbf{I}}$.

The proposed method is outlined in Method 2. First, a better suited constraining window in the frequency domain is used to enhance the adaptation process and the cumulative difference of every block, $\mathbf{\Lambda}_b$, is updated. Afterwards, every P frames, the unconstrained filter update of block c, $c = ((l/P))_B$, is reconstructed using $\mathbf{\Lambda}_c$, prior to eliminating the wrap-around errors. Consequently, every block is corrected perfectly, as the correction of the linear components is performed using only the cumulative difference of the block to be corrected. Further, neither the approximation of the constraining window nor the frame overlap are specified, as they can be arbitrarily chosen.

Method 2. EAC gradient update method

1: **for** $b = 0 : B-1$ **do**
2: $\quad \widehat{\mathbf{H}}_b(l+1) = \widehat{\mathbf{H}}_b(l) + \underline{\mathbf{G}}_{\text{arb}}\Delta\widehat{\mathbf{H}}_b(l)$
3: $\quad \mathbf{\Lambda}_b = \mathbf{\Lambda}_b + \Delta\widehat{\mathbf{H}}_b(l) - \underline{\mathbf{G}}_{\text{arb}}\Delta\widehat{\mathbf{H}}_b(l)$
4: **if** $l/P \in \mathbb{N}^0$ then $c = ((l/P))_B$ **then**
5: $\quad \widehat{\mathbf{H}}_c(l+1) = \underline{\mathbf{G}}\left[\widehat{\mathbf{H}}_c(l+1) + \mathbf{\Lambda}_c\right]$
6: $\quad \mathbf{\Lambda}_c = \mathbf{0}$

V. Performance Evaluation

The simulation set-up was designed as follows. A room impulse response (RIR) of length $L = 2048$ taps was generated using the image method [14], for a room of $5 \times 4 \times 3$ m^3 (width $\times$ length $\times$ height) with a reverberation time T_{60} of 350 ms at a sampling frequency of 16 kHz. The distance between loudspeaker and microphone ℓ_m was set to 1 m. A DFT length M of 512 points was used. Two partition lengths were tested, namely $N = M/2$ and $N = 3M/4$, both with a frame shift of $R = V$, which corresponds to a frame overlap of 50% and 75%, respectively. The PB frequency-domain normalized least mean squares (NLMS) algorithm [10], i.e.,

$$\widehat{\mathbf{H}}_b(l+1) = \widehat{\mathbf{H}}_b(l) + \mu\underline{\mathbf{G}}_{\text{arb}}\widehat{\underline{\mathbf{S}}}_b(l)^{-1}\underline{\mathbf{X}}_b^{\text{H}}(l)\mathbf{E}(l) \tag{19}$$

was used for the evaluation, where μ is the adaptation step size and $\hat{\underline{\mathbf{S}}}_b(l)$ is a diagonal matrix that is recursively estimated by

$$\widehat{\underline{\mathbf{S}}}_b(l) = \beta\widehat{\underline{\mathbf{S}}}_b(l-1) + (1-\beta)\underline{\mathbf{X}}_b(l)\underline{\mathbf{X}}_b^{\text{H}}(l)/M, \tag{20}$$

where β is the forgetting factor. The simulations were run with $\beta = 0.9$ and $\mu = 0.06/L$. For 50% overlap, $\underline{\mathbf{G}}_{\text{arb}} = \underline{\mathbf{G}}_{\text{s}}$ was used, while for 75% overlap, $\underline{\mathbf{G}}_{75}$ was used. The latter was designed analogously to $\underline{\mathbf{G}}_{\text{s}}$ in [10] but for $N = 3M/4$. Hence, all but the main- and two first off-diagonals of $\underline{\mathbf{G}}$ can be and, thus, are neglected. Further, $\mathbf{g}_{75} = \text{diag}\{\underline{\mathbf{F}}^{-1}\underline{\mathbf{G}}_{75}\underline{\mathbf{F}}\} \geq \mathbf{0}_{1\times M}$ was required as for $\mathbf{g}_{\text{s}}$. The performance was measured using the normalized mean-square error (NMSE), given by

$$\text{NMSE}(l) = 10\log_{10}||\mathbf{e}(l)||_2^2/||\mathbf{d}(l)||_2^2, \tag{21}$$

where $||\cdot||_2$ denotes the l_2-norm and $\mathbf{d}(l)$ is the true echo signal vector. For the performance analysis, unit-variance white Gaussian noise and speech were used as excitation signals and the obtained results were averaged over 100 trials. Moreover, white Gaussian noise was added to the microphone signal to obtain a segmental echo-to-noise ratio (segENR) of 40 dB. Thus, the lower bound for the achievable NMSE is approximately -40 dB. For the comparison of the gradient update methods, a frame interval between corrections of $P = 10$ was used for 50% overlap and $P = 20$ was used for 75% overlap.

In Fig. 1(a) and (b), a performance comparison of the gradient update methods with 50% frame overlap is depicted for

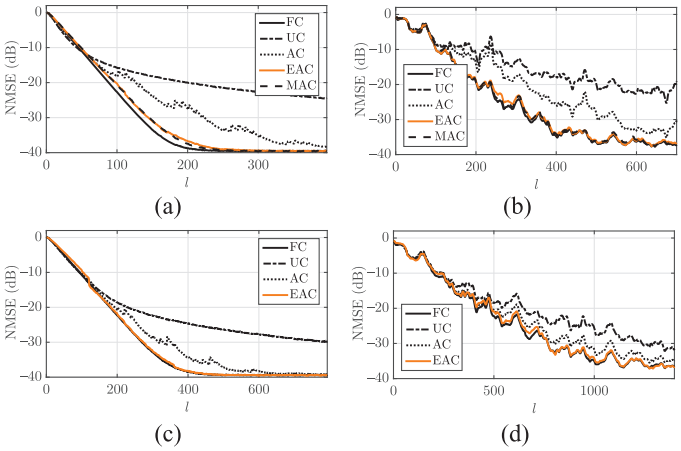

Fig. 1. Performance comparison of the gradient update methods for 50% overlap with (a) noise and (b) speech input, and for 75% overlap with (c) noise and (d) speech input.

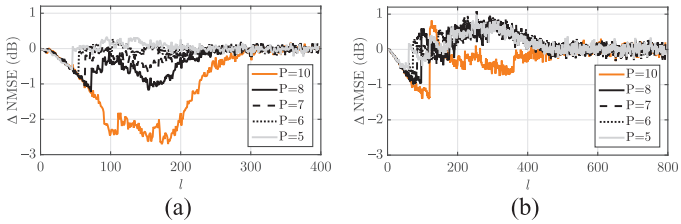

Fig. 2. $\Delta\text{NMSE}(l, P)$ for (a) 50% and (b) 75% overlap.

noise and speech as input signals. The NMSE curves corresponding to the latter were smoothed over time for better clarity. It can be observed that the slowest convergence is provided by the UC update, followed by the AC update. Moreover, if noise is used as input signal, the NMSE curve corresponding to the AC update presents a staircase-like structure. For both signals, the MAC and the EAC updates perform similarly and nearly as well as the optimal fully constrained (FC) update.

In order to demonstrate the flexibility of the proposed EAC update, a comparison with a frame overlap of 75% is depicted in Fig. 1(c) and (d). The MAC update is not included in this comparison as it is not defined for $N \neq M/2$. It can be observed that, for both signals, the EAC update outperforms the UC update and performs better than the AC update. Moreover, its performance is nearly optimal.

In Fig. 2, the difference between the NMSE curves of the optimal FC and the EAC updates, $\Delta\text{NMSE}(l) = \text{NMSE}_{\text{FC}}(l) - \text{NMSE}_{\text{EAC}}(l)$, is depicted for different values of P. It can be observed that by decreasing P, the performance improves significantly. Fig. 2(a) shows that the EAC update already achieved near-optimal performance for $P = 6$.

VI. Complexity Analysis

The aim of the proposed update method is to enhance the performance of PB-FDAF algorithms, while keeping the complexity as low as possible. For the complexity analysis the number of basic operations are used, i.e., additions and multiplications. The complexity per frame of a PB-FDAF is

$$\mathcal{C}(\text{GU}) \approx 3O(\text{FFT}) + BO(\mathbb{C}\times) + BO(\text{GU}), \qquad (22)$$

TABLE I
Complexity Per Block of the Gradient Update

Method	$O(\text{GU})$
FC	$2O(\text{FFT}) + O(\mathbb{C}\times)$
UC	$O(\mathbb{C}\times) + O(\omega_{\text{uc}})$
AC	$2/(PB)O(\text{FFT}) + O(\mathbb{C}\times) + O(\omega_{\text{uc}})$
MAC	$2/(PB)O(\text{FFT}) + O(\mathbb{C}\times) + O(\omega_{\text{s}})$
EAC	$2/(PB)O(\text{FFT}) + O(\mathbb{C}\times) + O(\omega_{\text{arb}})$

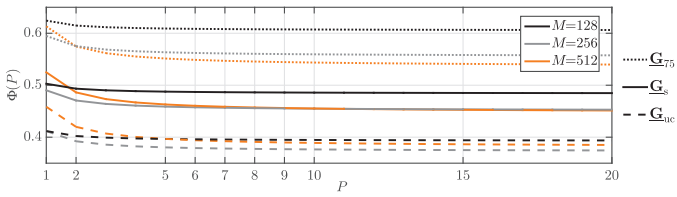

Fig. 3. Complexity reduction per frame as a function of P, for different values of M and $\underline{\mathbf{G}}_{\text{arb}}$.

where $O(\text{GU})$ denotes the complexity per block of the gradient update, $O(\text{FFT}) \approx 2M\log_2(M) - 4M$ is the complexity of a FFT, and $O(\mathbb{C}\times) \approx 6M$ is the complexity of a complex multiplication of length M (see [10] and references therein).

In Table I, the complexity per block of each of the methods is provided, where, as in [10], the complexity derived from the correction of the linear components is neglected. Further, $O(\omega_{\text{uc}})$, $O(\omega_{\text{s}})$, and $O(\omega_{\text{arb}})$ denote the algorithmic complexity of applying $\underline{\mathbf{G}}_{\text{uc}}$, $\underline{\mathbf{G}}_{\text{s}}$, and $\underline{\mathbf{G}}_{\text{arb}}$, respectively. It can be observed that the complexity of the EAC gradient update $O(\text{EAC})$ strongly depends on $\underline{\mathbf{G}}_{\text{arb}}$. So that, the lower bound of $O(\text{EAC})$ is given by the complexity of the AC gradient update, while its upper bound is given by the complexity of the FC gradient update. Moreover, if (14) is used, the EAC and the MAC gradient updates are equally complex.

The complexity reduction per frame obtained by the EAC gradient update w.r.t. the FC gradient update is given by the ratio $\Phi = \mathcal{C}(\text{EAC})/\mathcal{C}(\text{FC})$. This ratio is depicted in Fig. 3 as a function of P, for different DFT lengths and constraining matrix approximations. The complexity of applying $\underline{\mathbf{G}}_{\text{uc}}$ is $O(\omega_{\text{uc}}) = 0$, while, if an efficient implementation is used as in [10], $O(\omega_{\text{s}}) = 3M$ and $O(\omega_{75}) = 7M$. Considering Fig. 3 and the simulation set-up in Section V, one can conclude that the performance of the EAC gradient update in Fig. 1(a) and (b) was obtained at the cost of $\Phi \approx 0.46$ times the complexity of the FC gradient update, and in Fig. 1(c) and (d) of $\Phi \approx 0.54$.

VII. Conclusion

We have presented an alternative to existing complexity reduction methods for PB-FDAF algorithms. As a basis for the derivation of the proposed method, an overview of previously proposed methods was provided. The proposed method can be regarded as a generalization of the AC gradient update method, with the advantage that the constraining window can be arbitrarily approximated. It was demonstrated that for a fixed DFT length, the proposed method is flexible in terms of the block partition length, and consequently of the frame overlap. Finally, it was shown that the proposed method is able to reduce the complexity by more than 40% with respect to the FC method and still achieve near-optimal performance.

REFERENCES

[1] E. Hänsler and G. Schmidt, *Acoustic Echo and Noise Control: A practical Approach*. Hoboken, NJ, USA: Wiley, 2004.

[2] G. Schmidt, "Applications of acoustic echo control—An overview," in *Proc. Eur. Signal Process. Conf. (EUSIPCO)*, Vienna, Austria, Sep. 2004, pp. 9–16.

[3] J. J. Shynk, "Frequency-domain and multirate adaptive filtering," *IEEE Signal Process. Mag.*, vol. 9, no. 1, pp. 14–37, Jan. 1992.

[4] S. Haykin, *Adaptive Filter Theory*, 4th ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2002.

[5] P. C. W. Sommen, "Partitioned frequency-domain adaptive filters," in *Proc. Asilomar Conf. Signals Syst. Comput.*, 1989, pp. 677–681.

[6] D. Mansour and A. J. Gray, Jr, "Unconstrained frequency-domain adaptive filter," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 30, no. 5, pp. 726–734, Oct. 1982.

[7] J. S. Soo and K. K. Pang, "Multidelay block frequency domain adaptive filter," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, no. 2, pp. 373–376, Feb. 1990.

[8] J. Benesty and D. R. Morgan, "Frequency-domain adaptive filtering revisited, generalization to the multi-channel case, and application to acoustic echo cancellation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Istanbul, Turkey, Jun. 2000, vol. 2, pp. 289–292.

[9] M. Joho and G. S. Moschytz, "Connecting partitioned frequency-domain filters in parallel or in cascade," *IEEE Trans. Circuits Syst. II*, vol. 47, no. 8, pp. 685–697, Aug. 2000.

[10] R. M. M. Derkx, G. P. M. Engelmeers, and P. C. W. Sommen, "New constraining method for partitioned block frequency-domain adaptive filters," *IEEE Trans. Signal Process.*, vol. 50, no. 3, pp. 2177–2186, Sep. 2002.

[11] Y. Avargel and I. Cohen, "System identification in the short-time Fourier transform domain with crossband filtering," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 4, pp. 1305–1319, May 2007.

[12] A. Oppenheim and R. W. Schafer, *Digital Signal Processing*, 2nd ed. Englewood Cliff, NJ, USA: Prentice-Hall, 1993.

[13] L. Pelkowitz, "Frequency domain analysis of wraparound error in fast convolution algorithms," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 29, no. 3, pp. 413–422, Jun. 1981.

[14] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *J. Acoust. Soc. Amer.*, vol. 65, no. 4, pp. 943–950, Apr. 1979.