

RESEARCH

Open Access



The generalized frequency-domain adaptive filtering algorithm as an approximation of the block recursive least-squares algorithm

Martin Schneider* and Walter Kellermann

Abstract

Acoustic echo cancellation (AEC) is a well-known application of adaptive filters in communication acoustics. To implement AEC for multichannel reproduction systems, powerful adaptation algorithms like the generalized frequency-domain adaptive filtering (GFDAF) algorithm are required for satisfactory convergence behavior. In this paper, the GFDAF algorithm is rigorously derived as an approximation of the block recursive least-squares (RLS) algorithm. Thereby, the original formulation of the GFDAF algorithm is generalized while avoiding an error that has been in the original derivation. The presented algorithm formulation is applied to pruned transform-domain loudspeaker-enclosure-microphone models in a mathematically consistent manner. Such pruned models have recently been proposed to cope with the tremendous computational demands of massive multichannel AEC. Beyond its generalization, a regularization of the GFDAF is shown to have a close relation to the well-known block least-mean-squares algorithm.

Keywords: Acoustic echo cancellation, System identification, Generalized frequency-domain adaptive filtering

1 Introduction

Acoustic echo cancellation (AEC) is generally necessary in full-duplex communication scenarios where loudspeaker echoes should be removed from a microphone signal. This is, e.g., necessary for teleconferences where the microphone signal is sent to far-end communication partners who would be disturbed when hearing their own voices. Another application scenario is an acoustic human-machine interface where the automatic speech recognition would be impaired by the loudspeaker feedback in microphone signal.

AEC uses an adaptive filter identifying the echo path to obtain an echo replica that then is subtracted from the microphone signal [1–7]. Ideally, this is achieved without distorting the signals of the local acoustic scene in the microphone signal. This distinguishes AEC from acoustic

echo suppression, where the microphone signal is filtered in a way such that a distortion of the local acoustic scene cannot be avoided [8, 9]. However, acoustic echo suppression is often also used as a post-filtering method after AEC [10–15].

The principle of AEC has originally been applied to cancel the echoes in telephone hybrids [16]. The necessary adaptation algorithms were typically based on the well-known least-mean-squares (LMS) algorithm [17, 18]. The very popular normalized least-mean-squares (NLMS) algorithm is closely related to the class of affine projection algorithm [19, 20] of which efficient implementations are available [21]. The sparse nature of impulse responses describing telephone hybrid echoes motivated the formulation of the proportionate normalized least-mean-squares algorithm (PNLMS) and the improved PNLMS (IPNLMS) algorithms [22, 23], respectively.

When hands-free telephone sets were introduced, acoustic echoes became another significant problem in many telecommunication scenarios. Unlike the echo paths of telephone hybrids, acoustic echo paths are

*Correspondence: schneider@int.de
Multimedia Communication and Signal Processing, University
Erlangen-Nuremberg, Cauerstr. 7, 91058 Erlangen, Germany

described by significantly longer typically non-sparse impulse responses, as described by Hänslér in [24, 25]. This increased complexity fueled the search for efficient frequency-subband [26, 27] and discrete Fourier transform (DFT)-domain algorithms [28–30], where multiple shorter adaptive filters or individual DFT bins, respectively, are adapted independently and lead to faster convergence and increased computational efficiency. As the block processing of computationally efficient DFT-domain algorithms implies large algorithmic delays, the generalized multidelay adaptive filter was developed which reduces the block size by partitioning the impulse responses [31, 32]. Note that it is also possible to reduce this delay on cost of computational efficiency by choosing an appropriate block-overlap for single-partition processing [4].

On another track of research, a state-space model of the acoustic impulse responses was used to apply the concept of the Kalman filter to AEC [14, 33, 34]. These approaches feature an inherent step-size control, which renders a double-talk detection [3, 35] unnecessary. When the Kalman filter approach is formulated in the frequency domain, be it single [14] or partitioned block [36], the framework interestingly delivers an integrated frequency-domain filter structure and, hence, unites important concepts from adaptive filtering and adaptation control.

Recently emerging multichannel reproduction systems allow for improving the user experience in many kinds of telepresence systems but also human-machine interfaces, such as multi-party teleconferencing and immersive interactive gaming environments whenever the latter comprise an acoustic human-machine interface. Such scenarios imply the use of a multichannel AEC system where the typically strong correlation between the various loudspeaker channels hampers the convergence of adaptation algorithms [37, 38]. The generalized frequency-domain adaptive filtering (GFDAF) algorithm has been shown to largely overcome this problem while retaining computational efficiency. The GFDAF algorithm was first presented in [39], being inspired by [30, 40] and incorporating concepts of [31, 41, 42]. Note that the Kalman filter-based approaches have also been generalized for the efficient identification of multiple-input/multiple-output systems [43].

However, for massive multichannel systems with dozens of loudspeaker channels, AEC still involves tremendous computational demands and algorithmic challenges. Wave-domain adaptive filtering (WDAF) has been shown to overcome these problems by using a physically motivated loudspeaker-enclosure-microphone (LEM) model [44, 45], which allows to approximate the LEM system by a drastically reduced number of loudspeaker-to-microphone couplings described in the wave-domain. The resulting models will be referred to as pruned models in

the following. Due to its desirable properties, the GFDAF algorithm was also the algorithm of choice for most WDAF implementations.

This paper presents a comprehensive derivation of the GFDAF algorithm as an approximation of the well-known block recursive least-squares (RLS) algorithm with exponential windowing. The presented derivation clearly identifies all approximations that were implied in the original derivation such that an additional variant of this algorithm can be formulated. Moreover, a notation is used that was optimized for conciseness to alleviate further development of the algorithm. As a first step towards further development, the GFDAF algorithm is generalized to use pruned LEM models, for the first time in a mathematically consistent manner.

The paper is organized as follows: In Section 2, we formulate the system identification problem and its relation to AEC. As the basis for the following main parts of the paper, the RLS algorithm is briefly reviewed in Section 3 where it is shown that errors induced in the filter coefficients decay exponentially during adaptation using this algorithm. Additionally, a link between the LMS algorithm and a Tikhonov regularization of the RLS algorithm is shown. In Section 4, the GFDAF algorithm is rigorously derived as an approximation of the block RLS algorithm with exponential windowing and then generalized to pruned LEM models in Section 5. The derived algorithms are briefly evaluated in Section 6 to show the effect of the individual approximations used in the derivation. In Section 7, implications of the presented derivation for real-world implementations are discussed before conclusions are presented in Section 8.

2 System identification and acoustic echo cancellation

In this section, the system identification problem is related to AEC, and the signal model is introduced along with Fig. 1.

In the following, the L loudspeaker signals are described by the matrix $\mathbf{X}(k)$ which captures the individual samples $x_l(k)$ of loudspeaker channel l ($l = 0, 1, \dots, L - 1$) at discrete-time instant k . The structure of this matrix will

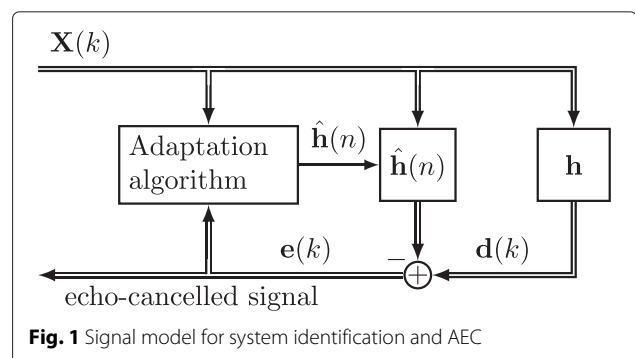


Fig. 1 Signal model for system identification and AEC

be explained and motivated later. These signals are fed to the LEM system, which is represented by the vector \mathbf{h} that captures the impulse responses of all microphone-to-loudspeaker paths. The LEM system is assumed to be time-invariant for the subsequent analysis. The individual samples of the impulse response from loudspeaker l to microphone m ($m = 0, 1, \dots, M - 1$) are denoted by $h_{m,l}(k)$ such that microphone signal m is given by

$$d_m(k) = \sum_{l=0}^{L-1} \sum_{\kappa=0}^{K-1} x_l(k - \kappa) h_{m,l}(\kappa). \quad (1)$$

This implies that the LEM system impulse responses are considered to be of length K and that additive noise in the microphone signals is neglected. Although these conditions will not be fulfilled for real-world systems, this does not limit the applicability of the following derivation.

The discrete-time microphone signals are captured by the vector

$$\mathbf{d}(k) = (\mathbf{d}_0(k), \mathbf{d}_1(k), \dots, \mathbf{d}_{M-1}(k))^T, \quad (2)$$

$$\mathbf{d}_m(k) = (d_m(k - P + 1), \dots, d_m(k))^T, \quad (3)$$

where signal segments of length P are considered and \cdot^T denotes the transposition. The choice of P will be discussed later. For system identification, an adaptation algorithm provides an estimate $\hat{\mathbf{h}}(n)$ of \mathbf{h} . Typically, this estimate is determined implicitly by minimizing a given norm of the error signal $\mathbf{e}(k)$. In the echo cancellation context, the error signal $\mathbf{e}(k)$ contains also the signals of the local sources in the LEM system, which should then be further processed and/or transmitted after the acoustic echoes are removed.

The column vector $\hat{\mathbf{h}}(n)$ of length KLM has the same structure as \mathbf{h} but is dependent on the block-time index $n = \lfloor k/N \rfloor$, where N denotes the frame shift of the adaptation algorithms as defined later and $\lfloor \cdot \rfloor$ denotes the floor operator. The resulting structure of $\hat{\mathbf{h}}(n)$ is then given by

$$\hat{\mathbf{h}}(n) = \left(\hat{\mathbf{h}}_{0,0}^T(n), \hat{\mathbf{h}}_{0,1}^T(n), \dots, \hat{\mathbf{h}}_{0,L-1}^T(n), \right. \\ \left. \hat{\mathbf{h}}_{1,0}^T(n), \dots, \hat{\mathbf{h}}_{M-1,L-1}^T(n) \right)^T \quad (4)$$

$$\hat{\mathbf{h}}_{m,l}(n) = \left(\hat{h}_{m,l}(0, n), \dots, \hat{h}_{m,l}(K - 1, n) \right)^T, \quad (5)$$

where $\hat{h}_{m,l}(k, n)$ are estimates of $h_{m,l}(k)$.

In order to implement (1) by the multiplication

$$\mathbf{d}(k) = \mathbf{X}(k)\mathbf{h}, \quad (6)$$

the $MP \times LMK$ matrix $\mathbf{X}(k)$ has to be defined as follows. The loudspeaker signals are first represented by

$$\mathbf{x}_l(k) = (x_l(k - P + 1), \dots, x_l(k))^T, \quad (7)$$

which is then used to form

$$\mathbf{X}_l(k) = (\mathbf{x}_l(k), \dots, \mathbf{x}_l(k - K + 1)), \quad (8)$$

such that

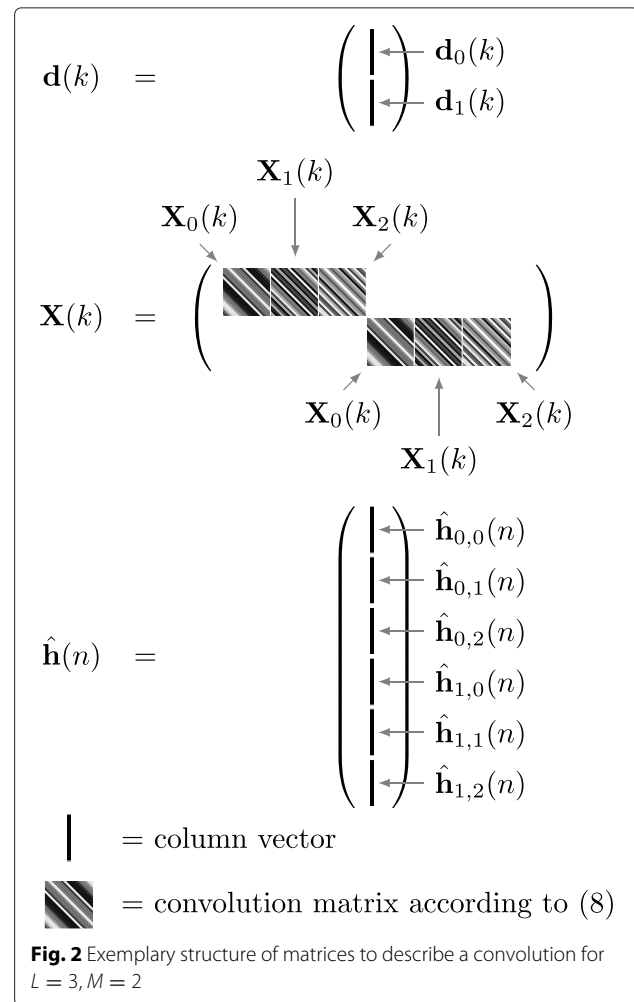
$$\mathbf{X}(k) = \mathbf{I}_M \otimes (\mathbf{X}_0(k), \mathbf{X}_1(k), \dots, \mathbf{X}_{L-1}(k)), \quad (9)$$

can be defined. Here, \otimes denotes the Kronecker product, and \mathbf{I}_M is an $M \times M$ identity matrix. The redundant representation of $\mathbf{X}(k)$ in (9), as illustrated by Fig. 2, allows for describing the microphone signals $\mathbf{d}(k)$ as a column vector. This representation will be exploited later in Section 5.

The adaptation error signal is then defined by

$$\mathbf{e}(k) = \mathbf{d}(k) - \mathbf{X}(k)\hat{\mathbf{h}}(n). \quad (10)$$

Note that k may assume any integer number, while only the time instants $k = nN$ will be relevant for the derivation of the adaptation algorithms presented later. In real-world implementations, $\mathbf{e}(k)$ will be used for further processing, which suggests to set the microphone signal segment length P equal to the frame shift N to obtain a gap-less signal in $\mathbf{e}(k)$. However, for generality, P is not determined by N in the presented derivation.



When minimizing the mean-square error (MSE), i. e., solving

$$\underset{\hat{\mathbf{h}}(n)}{\operatorname{argmin}} \left\{ \mathcal{E} \left\{ \mathbf{e}^H(k) \mathbf{e}(k) \right\} \right\} \quad (11)$$

with H denoting the Hermitian transpose, the following normal equation results:

$$\mathbf{R} \hat{\mathbf{h}}(n) = \mathbf{r}, \quad (12)$$

with

$$\mathbf{R} = \mathcal{E} \left\{ \mathbf{X}^H(k) \mathbf{X}(k) \right\}, \quad (13)$$

$$\mathbf{r} = \mathcal{E} \left\{ \mathbf{X}^H(k) \mathbf{d}(k) \right\} \quad (14)$$

being scaled versions of the loudspeaker signal autocorrelation matrix and the cross-correlation vector of loudspeaker and microphone signals, respectively. The scaling results from the P rows of $\mathbf{X}_l(k)$, which are time-shifted versions of each other. Since stationarity implies shift invariance under the expectation operator, \mathbf{R} represents the sum of P identical loudspeaker signal autocorrelation matrices. The same holds for \mathbf{r} with respect to the cross-correlation vector. If $P = 1$ is chosen, \mathbf{R} and \mathbf{r} describe the loudspeaker signal autocorrelation matrix and the cross-correlation vector of loudspeaker and microphone signals, respectively, as they are commonly used in the literature [17].

3 Recursive least-squares (RLS) and least-mean-square (LMS) algorithms

In the first part of this section, the well-known RLS algorithm is briefly reviewed to form the basis for the subsequent derivations. In Section 3.1, the effect of filter coefficient errors on the further convergence of the RLS algorithm is treated. The LMS algorithm is derived in Section 3.2 in order to establish a link between the LMS algorithm and a Tikhonov regularization of the RLS algorithm.

The RLS algorithm as considered here minimizes the cost function

$$J_{\text{RLS}}(n) = \sum_{v=0}^n \lambda^{n-v} \mathbf{e}^H(vN) \mathbf{e}(vN), \quad (15)$$

using the exponential window defined by “forgetting factor” λ [17]. The weighted least-squares criterion given by (15) approximates $\mathcal{E} \left\{ \mathbf{e}^H(k) \mathbf{e}(k) \right\}$ in (11) and can be used when the second order moments of the loudspeaker and microphone signals are unknown. When choosing $N = P = K$, (15) is identical to the cost function used in [39] up to a scaling factor.

Plugging (10) into (15) and setting the Wirtinger gradient [17, 46] of the result to zero leads to an $\hat{\mathbf{h}}(n)$ that minimizes (15). This gradient is given by

$$\frac{\partial J_{\text{RLS}}(n)}{\partial \hat{\mathbf{h}}^H(n)} = \sum_{v=0}^n \lambda^{n-v} \mathbf{X}^H(vN) \mathbf{X}(vN) \hat{\mathbf{h}}(n) - \mathbf{X}^H(vN) \mathbf{d}(vN), \quad (16)$$

where $\frac{\partial}{\partial \hat{\mathbf{h}}^H(n)} J_{\text{RLS}}(n) = 0$ leads to

$$\hat{\mathbf{R}}(n) \hat{\mathbf{h}}(n) = \hat{\mathbf{r}}(n) \quad (17)$$

with

$$\hat{\mathbf{R}}(n) = \sum_{v=0}^n \lambda^{n-v} \mathbf{X}^H(vN) \mathbf{X}(vN) \quad (18)$$

$$= \lambda \hat{\mathbf{R}}(n-1) + \mathbf{X}^H(nN) \mathbf{X}(nN), \quad (19)$$

$$\hat{\mathbf{r}}(n) = \sum_{v=0}^n \lambda^{n-v} \mathbf{X}^H(vN) \mathbf{d}(vN) \quad (20)$$

$$= \lambda \hat{\mathbf{r}}(n-1) + \mathbf{X}^H(nN) \mathbf{d}(nN). \quad (21)$$

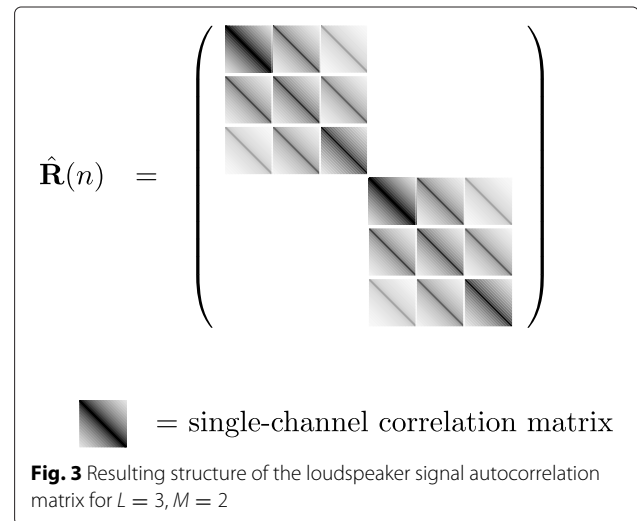
Note that $\hat{\mathbf{R}}(n)$ and $\hat{\mathbf{r}}(n)$ can be seen as recursive estimates of \mathbf{R} and \mathbf{r} , respectively, such that (17) approximates the solution defined by (12). Due to the similarity between (13) and (18), $\hat{\mathbf{R}}(n)$ shares the structure of \mathbf{R} , which is illustrated in Fig. 3.

In the following, a recursive algorithm determining $\hat{\mathbf{h}}(n)$ is derived. Multiplying (19) from the right-hand side by the previously computed filter coefficients $\hat{\mathbf{h}}(n-1)$ and subtracting the result from (21) leads to

$$\begin{aligned} \hat{\mathbf{r}}(n) - \hat{\mathbf{R}}(n) \hat{\mathbf{h}}(n-1) &= \lambda \left(\hat{\mathbf{r}}(n-1) - \hat{\mathbf{R}}(n-1) \hat{\mathbf{h}}(n-1) \right) \\ &\quad + \mathbf{X}^H(nN) \left(\mathbf{d}(nN) - \mathbf{X}(nN) \hat{\mathbf{h}}(n-1) \right). \end{aligned} \quad (22)$$

Substituting $\hat{\mathbf{r}}(n)$ using (17) and defining the a priori estimation error

$$\mathbf{e}'(k) = \mathbf{d}(k) - \mathbf{X}(k) \hat{\mathbf{h}}(n-1) \quad (23)$$



leads to

$$\hat{\mathbf{R}}(n)\hat{\mathbf{h}}(n) = \hat{\mathbf{R}}(n)\hat{\mathbf{h}}(n-1) + \mathbf{X}^H(nN)\mathbf{e}'(nN) + \lambda \left(\hat{\mathbf{r}}(n-1) - \hat{\mathbf{R}}(n-1)\hat{\mathbf{h}}(n-1) \right) \quad (24)$$

and finally to the explicit formulation of the adaptation algorithm, assuming that $\hat{\mathbf{R}}(n)$ is invertible and $\hat{\mathbf{h}}(n-1)$ fulfills (17) for $n-1$:

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \hat{\mathbf{R}}^{-1}(n)\mathbf{X}^H(nN)\mathbf{e}'(nN). \quad (25)$$

Note that the a priori estimation error $\mathbf{e}'(k)$ must be clearly distinguished from the a posteriori estimation error $\mathbf{e}(k)$, which depends on $\hat{\mathbf{h}}(n)$ instead of $\hat{\mathbf{h}}(n-1)$. Unfortunately, these errors are not correctly distinguished in [39].

Since (17) describes the solution of a least-squares problem, $\hat{\mathbf{R}}^{-1}(n)$ can be replaced by the Moore-Penrose pseudoinverse, if $\hat{\mathbf{R}}(n)$ is not invertible [17]. However, this is merely of theoretical interest since the Moore-Penrose pseudoinverse is expensive to compute, and real-world implementations will most likely rely on regularization as described later by (36). Moreover, the inverse $\hat{\mathbf{R}}^{-1}(n)$ can also be computed using the well-known matrix inversion lemma [17]. However, this approach leads only to an increased efficiency, if (19) describes a rank-deficient update of $\hat{\mathbf{R}}(n)$, where a higher rank of the update implies a lower gain in efficiency. Hence, this approach is less attractive with growing P . Additionally, the update of the matrix inverted in (36) is generally full-rank. Thus, this approach is not discussed in the paper.

3.1 Effect of regularization and approximation errors

For any approximation or regularized version of the RLS algorithm, (17) will not be fulfilled exactly by $\hat{\mathbf{h}}(n-1)$. In that case, $\lambda \left(\hat{\mathbf{r}}(n-1) - \hat{\mathbf{R}}(n-1)\hat{\mathbf{h}}(n-1) \right)$ does not vanish, and $\hat{\mathbf{h}}(n-1)$ can be described by

$$\hat{\mathbf{h}}(n-1) = \hat{\mathbf{h}}_{\text{opt}}(n-1) + \Delta\hat{\mathbf{h}}(n-1), \quad (26)$$

where the optimal component $\hat{\mathbf{h}}_{\text{opt}}(n-1)$ of the filter coefficients fulfills (17) at block time instant $n-1$, while the error component $\Delta\hat{\mathbf{h}}(n)$ does not. Then, multiplying (24) with $\hat{\mathbf{R}}^{-1}(n)$ from the left-hand side leads to

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \hat{\mathbf{R}}^{-1}(n)\mathbf{X}^H(nN)\mathbf{e}'(nN) + \lambda\hat{\mathbf{R}}^{-1}(n) \left(\hat{\mathbf{r}}(n-1) - \hat{\mathbf{R}}(n-1)\hat{\mathbf{h}}(n-1) \right) \quad (27)$$

as the adaptation rule to obtain optimal filter coefficients from previous suboptimal coefficients. When comparing (25) to (27) it can be seen that suboptimal filter coefficients $\hat{\mathbf{h}}(n-1)$ require an additional correction term in order to obtain optimal coefficients in $\hat{\mathbf{h}}(n)$. Since this term is not considered in (25), any perturbation of $\hat{\mathbf{h}}(n-1)$

will lead to suboptimal coefficients in $\hat{\mathbf{h}}(n)$. The resulting error can be determined by subtracting (25) from (27) and plugging (26) into the result. Then, $\hat{\mathbf{r}}(n-1) - \hat{\mathbf{R}}(n-1)\hat{\mathbf{h}}_{\text{opt}}(n-1)$ vanishes, which leads to

$$\Delta\hat{\mathbf{h}}(n) = -\lambda\hat{\mathbf{R}}^{-1}(n)\hat{\mathbf{R}}(n-1)\Delta\hat{\mathbf{h}}(n-1). \quad (28)$$

This gives rise to the question how this error propagates in the following iterations. Fortunately, recursive application of (28) leads to

$$\Delta\hat{\mathbf{h}}(n+1) = (-\lambda)^2\hat{\mathbf{R}}^{-1}(n+1)\hat{\mathbf{R}}(n-1)\Delta\hat{\mathbf{h}}(n-1), \quad (29)$$

$$\Delta\hat{\mathbf{h}}(n+2) = (-\lambda)^3\hat{\mathbf{R}}^{-1}(n+2)\hat{\mathbf{R}}(n-1)\Delta\hat{\mathbf{h}}(n-1), \quad (30)$$

which shows that any error introduced in $\hat{\mathbf{h}}(n)$ decays exponentially, while the reconvergence speed is determined by the parameter λ .

3.2 Link between the LMS algorithm and the Tikhonov regularized RLS algorithm

To establish the link between the LMS algorithm and the regularized RLS algorithm, the LMS algorithm is briefly derived in the following. To this end, solving (11) using the gradient descent method can be viewed as a first step [17]. In that approach, the filter coefficients $\hat{\mathbf{h}}(n)$ are determined computing the gradient of $\mathcal{E} \{ \mathbf{e}^H(nN)\mathbf{e}(nN) \}$ for $\hat{\mathbf{h}}(n-1)$,

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu \left(\mathbf{r} - \hat{\mathbf{R}}\hat{\mathbf{h}}(n-1) \right), \quad (31)$$

where μ is a parameter to control the step size that could also be set adaptively [35]. For simplicity, the LMS algorithm uses the instantaneous estimates of (13) and (14) given by

$$\mathbf{R} \approx \mathbf{X}^H(k)\mathbf{X}(k), \quad (32)$$

$$\mathbf{r} \approx \mathbf{X}^H(k)\mathbf{d}(k). \quad (33)$$

This leads to a representation of (31) by

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu\mathbf{X}^H(nN) \cdot \left(\mathbf{d}(nN) - \mathbf{X}(nN)\hat{\mathbf{h}}(n-1) \right), \quad (34)$$

$$= \hat{\mathbf{h}}(n-1) + \mu\mathbf{X}^H(nN)\mathbf{e}'(nN), \quad (35)$$

where (23) was used to obtain (34) from (35). While choosing $P = N = 1$, (35) describes the LMS algorithm in its most common form, the formulation presented here allows for block-wise processing of the data.

When comparing (35) to (25), structural similarities can be exploited to obtain the following equation:

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \left((1-\alpha)\hat{\mathbf{R}}(n) + \frac{\alpha}{\mu}\mathbf{I}_{KLM} \right)^{-1} \mathbf{X}^H(nN)\mathbf{e}'(nN), \quad (36)$$

where α is a parameter of choice with $0 \leq \alpha \leq 1$. For $\alpha = 0$, (36) describes the RLS algorithm (25), for $\alpha = 1$, the LMS algorithm (35) is described. By choosing α between 0 and 1, the adaptation steps can be continuously varied in between both algorithms, although the relation is not linear. Since $\hat{\mathbf{R}}(n)$ is positive semi-definite, the inverse exists for any $\alpha > 0$. Moreover, when computing the inverse, choosing a larger α can reduce the condition number of the matrix to be inverted.

A comparable consideration can be found in [47]. However, the block RLS algorithm used there does not allow for an exponential time-windowing. Furthermore, the NLMS algorithm described there is identical to the algorithm that is most commonly referred to as affine projection algorithm. In [47], it is not possible to continuously vary the relative weight of the adaptation steps provided by both algorithms.

4 Generalized frequency-domain adaptive filtering algorithm

The derivation of the GFDAF algorithm presented in this section differs from [39] in the following points:

- The derivation is based on replacing the convolution matrices captured in (25) by DFT-domain multiplication instead of defining an equivalent to (15) in the DFT domain. This allows to show the relation of the GFDAF algorithm to the RLS algorithm more clearly. A similar approach is known for the Kalman filter-based AEC [14, 36].
- An erroneous equality used in the original derivation is clearly identified as an approximation.
- The frame shift N and the lengths of the adaptive filters K can be chosen independently of the microphone signal segment length P , as it was already described for the single-channel frequency-domain adaptive filtering algorithm [4] and for the Kalman filter implementations in the DFT-domain [14, 36, 43].
- The multichannel microphone signals are represented by a vector $\mathbf{d}(k)$ instead of a matrix, which allows for considering simplified models, as described later.
- A different regularization approach is proposed that is closely linked to the well-known LMS algorithm.

In [39], a DFT-domain equivalent to (15) was used to derive the GFDAF algorithm. Since the block RLS algorithm derived in (3), which minimizes (15) involves no approximations, (25) can be used for further derivations without restrictions. As a first step, (25) will be rewritten in the DFT domain such that this representation can be approximated to formulate the GFDAF algorithm. It is well-known that a time-domain convolution can

be facilitated by a DFT-domain multiplication using the overlap-save method [14, 36]. This approach is described in the following to ensure compatibility with the notation used in this paper, aiming at a length- Q DFT-domain representation of the loudspeaker signals captured in $\mathbf{X}(k)$. First, the individual loudspeaker signals $\mathbf{X}_l(k)$ are considered, where

$$\mathbf{X}_l(k) = (\mathbf{0}_{P \times (Q-P)}, \mathbf{I}_P) \hat{\mathbf{X}}_l(k) \begin{pmatrix} \mathbf{I}_K \\ \mathbf{0}_{(Q-K) \times K} \end{pmatrix} \quad (37)$$

with

$$\hat{\mathbf{X}}_l(k) = \begin{pmatrix} \mathbf{X}_l^{(A)}(k) & \mathbf{X}_l^{(B)}(k) \\ \mathbf{X}_l^{(C)}(k) & \mathbf{X}_l^{(D)}(k) \end{pmatrix} \quad (38)$$

holds for any matrices $\mathbf{X}_l^{(A)}(k)$, $\mathbf{X}_l^{(B)}(k)$, and $\mathbf{X}_l^{(C)}(k)$ of compatible dimensions. Since $\mathbf{X}_l(k)$ is a Toeplitz matrix, $\mathbf{X}_l^{(A)}(k)$, $\mathbf{X}_l^{(B)}(k)$, and $\mathbf{X}_l^{(C)}(k)$ can be chosen such that $\hat{\mathbf{X}}_l(k)$ is a circulant matrix that can be diagonalized by the DFT matrix. The entries of the $Q \times Q$ DFT matrix \mathbf{F}_Q in row p column q are given by

$$[\mathbf{F}_Q]_{p,q} = \frac{1}{\sqrt{Q}} e^{-j(p-1)(q-1)\frac{2\pi}{Q}}, \quad (39)$$

where j is used as the imaginary unit. The symmetric definition of the DFT using the scaling factor $1/\sqrt{Q}$ in (39) is crucial for the following considerations, although a different scaling factor was used in [39]. This leads to

$$\underline{\mathbf{X}}_l(k) = \mathbf{F}_Q \hat{\mathbf{X}}_l(k) \mathbf{F}_Q^H \quad (40)$$

$$= \sqrt{Q} \text{Diag} \{ \mathbf{F}_Q \mathbf{x}'_l(k) \} \quad (41)$$

where $\mathbf{x}'_l(k)$ is defined like $\mathbf{x}_l(k)$ but capturing signal segments of length Q instead of P . To describe the filtering through a multiple-input/multiple-output system, the individual matrices $\underline{\mathbf{X}}_l(k)$ are captured by

$$\underline{\mathbf{X}}(k) = \mathbf{I}_M \otimes (\underline{\mathbf{X}}_0(k), \underline{\mathbf{X}}_1(k), \dots, \underline{\mathbf{X}}_{L-1}(k)), \quad (42)$$

as it is also described by [39, 43]. The structure of $\underline{\mathbf{X}}(k)$ is illustrated in Fig. 4.

Using $\underline{\mathbf{X}}(k)$, it is possible to write

$$\mathbf{X}(k) = \mathbf{W}_{01} \underline{\mathbf{X}}(k) \mathbf{W}_{10}, \quad (43)$$

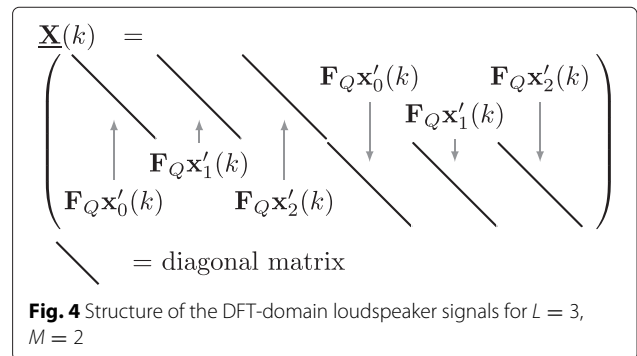


Fig. 4 Structure of the DFT-domain loudspeaker signals for $L = 3$, $M = 2$

where the matrices

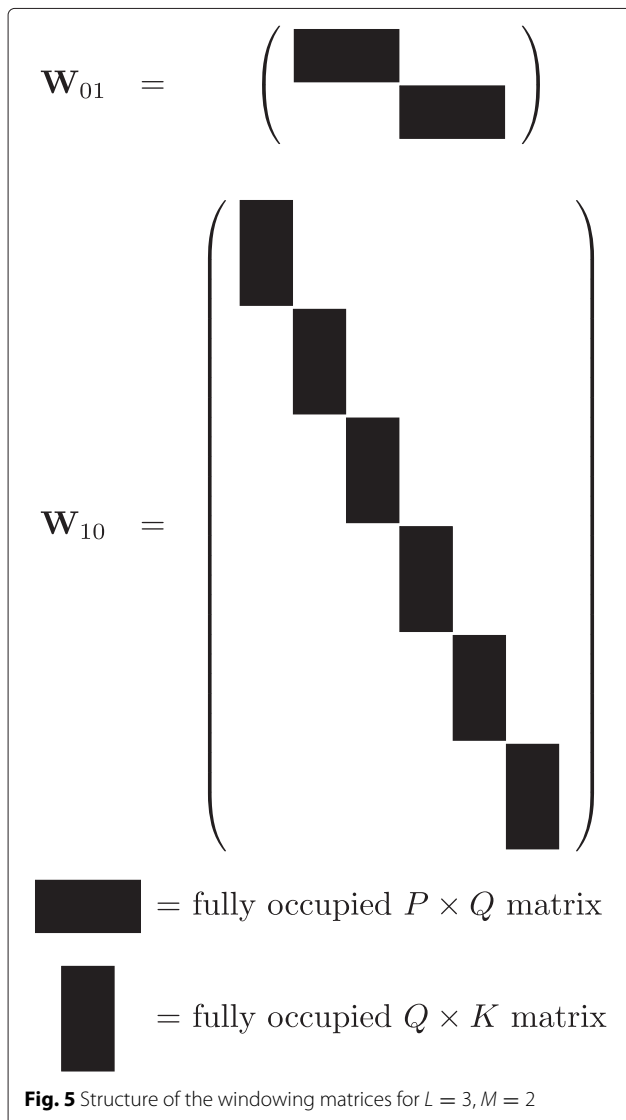
$$\mathbf{W}_{01} = \mathbf{I}_M \otimes \left((\mathbf{0}_{P \times (Q-P)}, \mathbf{I}_P) \mathbf{F}_Q^H \right), \quad (44)$$

$$\mathbf{W}_{10} = \mathbf{I}_{LM} \otimes \left(\mathbf{F}_Q \begin{pmatrix} \mathbf{I}_K \\ \mathbf{0}_{(Q-K) \times K} \end{pmatrix} \right) \quad (45)$$

are used to transform signal vectors from and to the DFT domain as well as for discrete-time truncation and zero-padding operations. An example for the structure of the matrices described by (44) and (45) is shown in Fig. 5. Note that $Q = P + K - 1$ is not necessary following, but $Q \geq P + K - 1$ is assumed. This is different to [39], where only the case $Q = 2P = 2K$ was covered.

For the further derivations,

$$\hat{\underline{\mathbf{S}}}(n) = \lambda \hat{\underline{\mathbf{S}}}(n-1) + \underline{\mathbf{X}}^H(nN) \mathbf{W}_{01}^H \mathbf{W}_{01} \underline{\mathbf{X}}(nN) \quad (46)$$



is defined, where (43) and (19) can be used to verify that

$$\hat{\mathbf{R}}(n) = \mathbf{W}_{10}^H \hat{\underline{\mathbf{S}}}(n) \mathbf{W}_{10} \quad (47)$$

holds. The matrix $\hat{\underline{\mathbf{S}}}(n)$ can be interpreted as an estimate of the DFT-domain power spectral density of the loudspeaker signals. Considering (25) and replacing $\hat{\mathbf{R}}(n)$ by (47) and $\underline{\mathbf{X}}^H(nN)$ by (43) results in

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \left(\mathbf{W}_{10}^H \hat{\underline{\mathbf{S}}}(n) \mathbf{W}_{10} \right)^{-1} \cdot \mathbf{W}_{10}^H \underline{\mathbf{X}}^H(nN) \mathbf{W}_{01}^H \mathbf{e}'(nN), \quad (48)$$

which represents the same time-domain update equation as (25) but is based on DFT-domain representations of the involved signals $\underline{\mathbf{X}}(k)$ and $\mathbf{e}'(k)$.

In (48), the size of the generally fully occupied matrix $\mathbf{W}_{10}^H \hat{\underline{\mathbf{S}}}(n) \mathbf{W}_{10}$ and its inverse preclude a real-world implementation of this algorithm for larger filter lengths or a larger number of loudspeaker channels. To overcome this obstacle, it was proposed in [39] to invert a sparse approximation of $\hat{\underline{\mathbf{S}}}(n)$ rather than inverting $\mathbf{W}_{10}^H \hat{\underline{\mathbf{S}}}(n) \mathbf{W}_{10}$ in (48).

As $\underline{\mathbf{X}}(k)$ is sparse, the lack of sparsity in $\hat{\underline{\mathbf{S}}}(n)$ can be attributed to the term $\mathbf{W}_{01}^H \mathbf{W}_{01}$ which represents windowing with a rectangular window in the time domain. Considering the definition of the DFT matrix given by (39), evaluating $\mathbf{W}_{01}^H \mathbf{W}_{01}$ leads to

$$\mathbf{W}_{01}^H \mathbf{W}_{01} = \mathbf{I}_M \otimes \left(\mathbf{F}_Q \text{Diag} \{ \mathbf{w} \} \mathbf{F}_Q^H \right), \quad (49)$$

with

$$\left[\mathbf{F}_Q \text{Diag} \{ \mathbf{w} \} \mathbf{F}_Q^H \right]_{\zeta, \eta} = \frac{1}{Q} \sum_{k=0}^{Q-1} w(k) e^{jk(\eta-\zeta) \frac{2\pi}{Q}}, \quad (50)$$

where $w(k)$ describes an appropriate rectangular window function with the vector representation

$$\mathbf{w} = (w(0), w(1), \dots, w(Q-1))^T. \quad (51)$$

For $w(k) = 1$, (50) would describe an identity matrix, while the definition of

$$w(k) = \begin{cases} 1 & \text{for } Q-P \leq k < Q, \\ 0 & \text{otherwise} \end{cases} \quad (52)$$

describes the time-domain windowing according to (44). As described in [39], (50) can be identified as a finite geometric series, which allows to write (52) as

$$\begin{aligned} & \left[\mathbf{F}_Q \text{Diag} \{ \mathbf{w} \} \mathbf{F}_Q^H \right]_{\zeta, \eta} \\ &= \begin{cases} \frac{P}{Q} & \text{for } \zeta = \eta \\ \frac{-1}{Q} \cdot \frac{1 - e^{j(Q-P)(\eta-\zeta) \frac{2\pi}{Q}}}{1 - e^{j(\eta-\zeta) \frac{2\pi}{Q}}} & \text{otherwise.} \end{cases} \end{aligned} \quad (53)$$

It can be shown that the resulting circulant matrix captures an infinite series of sinc functions, multiplied by an exponential phase term to represent the time-domain

shift or asymmetry of the window in each row. The maximum of this function is located on the main diagonal, which suggests an approximation of $\mathbf{W}_{01}^H \mathbf{W}_{01}$ by an identity matrix. The approximation tends to be more accurate, the narrower the main lobe of the sinc function is or, equivalently, the larger the time-domain window is. Hence, $\mathbf{W}_{01}^H \mathbf{W}_{01}$ can be better approximated by a scaled identity matrix the larger P is, where

$$\mathbf{W}_{01}^H \mathbf{W}_{01} \approx \mathbf{I}_{MQ} \frac{P}{Q}. \tag{54}$$

This is an important generalization relative to [39], since P is a parameter of choice in contrast to [39], where only the special case $Q = 2P$ was considered. Note that the same result was already obtained for the Kalman filter implementations in the DFT-domain [14, 36, 43].

Using (54), (46) can be approximated by

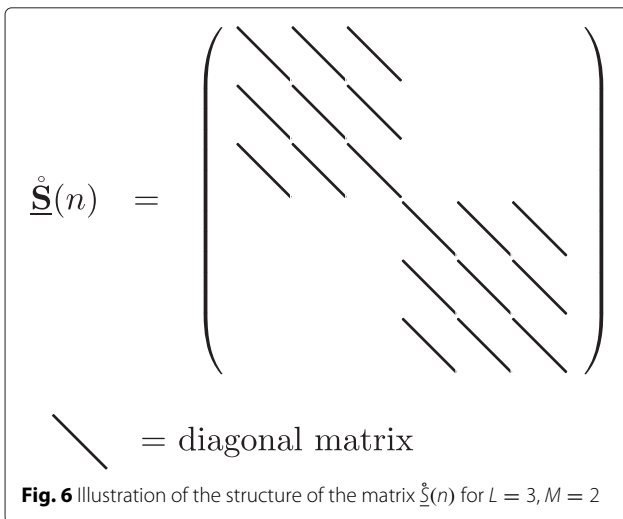
$$\underline{\hat{\mathbf{S}}}(n) = \lambda \underline{\hat{\mathbf{S}}}(n-1) + \frac{P}{Q} \underline{\mathbf{X}}^H(nN) \underline{\mathbf{X}}(nN) \tag{55}$$

where the structure of $\underline{\hat{\mathbf{S}}}(n)$ is illustrated in Fig. 6. Replacing $\underline{\hat{\mathbf{S}}}(n)$ by $\underline{\hat{\mathbf{S}}}(n)$ in (48) does not lead to an obvious advantage, yet. Therefore, another approximation is used:

$$\left(\mathbf{W}_{10}^H \underline{\hat{\mathbf{S}}}(n) \mathbf{W}_{10} \right)^{-1} \approx \mathbf{W}_{10}^H \underline{\hat{\mathbf{S}}}^{-1}(n) \mathbf{W}_{10}, \tag{56}$$

where $\underline{\hat{\mathbf{S}}}^{-1}(n)$ is now the inverse of a sparse matrix that is inexpensive to compute when exploiting the matrix structure accordingly. Erroneously, (56) was not identified as an approximation in [39] but as an equality. This is discussed in Appendix A, where using (56) is also justified.

Eventually, (56) can be used to approximate $\hat{\mathbf{R}}^{-1}(n)$ in the DFT-domain, which distinguishes the GFDAF algorithm from the RLS algorithm. Not only that this leads to tremendous computational savings, it also decouples the adaptation of the individual DFT bins [39]. As a result,



there are Q -independent inverses of $L \times L$ matrices to be determined, instead of the inverse of one $LK \times LK$ matrix. This explains the robustness of this algorithm, which is described by:

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \underbrace{\mu \mathbf{W}_{10}^H \left(\underline{\hat{\mathbf{S}}}(n) + \mathbf{D}(n) \right)^{-1} \mathbf{W}_{10}}_{\approx \hat{\mathbf{R}}^{-1}(n)} \cdot \underbrace{\mathbf{W}_{10}^H \underline{\mathbf{X}}^H(nN) \mathbf{W}_{01}^H \mathbf{e}'(nN)}_{=\mathbf{X}^H(nN) \mathbf{e}'(nN)}, \tag{57}$$

where the step-size parameter μ can be viewed as accounting for the inaccuracy of the approximation. This allows for using a step size close to or even larger than $\mu = 1$, according to the needs of the considered application scenario. Furthermore, the matrix

$$\mathbf{D}(n) = \Delta(n) \mathbf{I}_{KLM} \tag{58}$$

is used for regularization which is generally necessary for real-world implementations since $\underline{\hat{\mathbf{S}}}(n)$ will typically exhibit a large condition number or even become singular for signals $\mathbf{X}(k)$ with small spectral flatness or when the loudspeaker signals are strongly correlated.

A straightforward approach is to use a regularization weight function that is proportional to the loudspeaker signal power as defined by

$$\Delta(n) = \lambda \Delta(n-1) + \frac{\delta P}{LQ} \sum_{l=1}^L \mathbf{x}_l^H(nN) \mathbf{x}_l(nN), \tag{59}$$

where the non-negative parameter δ can be chosen to control the regularization and the multiplication by P is used to ensure the balance with the weight of the diagonal of $\underline{\hat{\mathbf{S}}}(n)$. Introducing $\mathbf{D}(n)$ into (57) describes a simple Tikhonov regularization where typical choices for δ are values close to zero. In Section 3.2, it was shown that such a Tikhonov regularization of the RLS algorithm is closely related to the LMS algorithm. The same holds for the GFDAF algorithm, such that a large δ would force the GFDAF algorithm to approach the adaptation steps of the LMS algorithm. Since the LMS algorithm is a well-understood algorithm, this regularization can easily be justified. Still, any $\delta > 0$ would lead to suboptimal filter coefficients. In Section 3.1, it was shown that filter coefficient errors will decay exponentially for the RLS algorithm. Since the GFDAF algorithm approximates the RLS algorithm, it can be expected to inherit this property such that suboptimal filter coefficients are not a major issue. Note that due to disregarding the approximation (56) (see also Appendix A), the algorithm variant described by (57) could not be presented in [39].

For efficient implementations, it is common to implement the time-domain convolution in the DFT domain. Accordingly, the a priori error signal can be expressed by

$$\mathbf{e}'(k) = \mathbf{d}(k) - \mathbf{W}_{01} \underline{\mathbf{X}}(k) \mathbf{W}_{10} \hat{\mathbf{h}}(n-1). \quad (60)$$

Any multiplication with \mathbf{W}_{10} implies LM DFTs, while a multiplication with \mathbf{W}_{01} implies M DFTs. To decrease the computational demands, four multiplications by \mathbf{W}_{10} can be eliminated by approximations. First, $\mathbf{W}_{10} \mathbf{W}_{10}^H$ can be approximated in the same way as $\mathbf{W}_{01}^H \mathbf{W}_{01}$ by

$$\mathbf{W}_{10} \mathbf{W}_{10}^H \approx \mathbf{I}_{LMQ} \frac{K}{Q}, \quad (61)$$

where K has the same role as P in (54). In (57), this leads to

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu \frac{K}{Q} \mathbf{W}_{10}^H (\hat{\mathbf{S}}(n) + \mathbf{D}(n))^{-1} \cdot \underline{\mathbf{X}}^H(nN) \mathbf{W}_{01}^H \mathbf{e}'(nN), \quad (62)$$

where the time-domain windowing by $\mathbf{W}_{10} \mathbf{W}_{10}^H$ is omitted. Furthermore, when considering

$$\hat{\underline{\mathbf{h}}}(n) = \mathbf{W}_{10} \hat{\mathbf{h}}(n), \quad (63)$$

the matrix \mathbf{W}_{10}^H can also be neglected which leads to the so-called unconstrained variant [39, 41] of this algorithm given by

$$\hat{\underline{\mathbf{h}}}(n) = \hat{\underline{\mathbf{h}}}(n-1) + \mu \frac{K}{Q} (\hat{\mathbf{S}}(n) + \mathbf{D}(n))^{-1} \underline{\mathbf{X}}^H(nN) \mathbf{W}_{01}^H \mathbf{e}'(nN). \quad (64)$$

In that case, (60) is also simplified to

$$\mathbf{e}'(k) = \mathbf{d}(k) - \mathbf{W}_{01} \underline{\mathbf{X}}(k) \hat{\underline{\mathbf{h}}}(n-1). \quad (65)$$

The time-domain windowing operations applied to the error signal cannot be neglected for the definition of the algorithm. Otherwise, the adaptive filter would converge to a solution for cyclic convolution and not to a solution for linear convolution with the time-domain filter coefficients.

5 Pruned loudspeaker-enclosure-microphone models

In this section, the adaptation algorithms described above are generalized to allow for system identification or AEC using pruned LEM models. Considering the structure of $\hat{\mathbf{h}}(n)$ described in (4), it is possible to define a matrix \mathbf{V} that implies certain components in $\hat{\mathbf{h}}(n)$ to be zero. This is done by requiring

$$\mathbf{V}^T \mathbf{V} \hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n) \quad (66)$$

within this section, which implies that certain coefficients in $\hat{\mathbf{h}}(n)$ are zero. Note that this is the only definition necessary to generalize the considered adaptation algorithms to pruned LEM models.

The values of \mathbf{V} can be defined by

$$\mathbf{V}^T \mathbf{V} = \text{Diag} \{ \mathbf{v} \}, \quad (67)$$

where component ζ of the vector \mathbf{v} is given by

$$[\mathbf{v}]_{\zeta} = \begin{cases} 1 & \text{if } [\mathbf{h}]_{\zeta} \text{ is modeled,} \\ 0 & \text{otherwise.} \end{cases} \quad (68)$$

At the same time, a multiplication by \mathbf{V} from the left-hand side would prune the zero-valued coefficients from $\hat{\mathbf{h}}(n)$. Exemplary structures of \mathbf{V} and $\mathbf{V}^T \mathbf{V}$ are shown in Fig. 7. Note that the following derivation of the RLS algorithm allows to choose freely whether any of the coefficients of \mathbf{h} is modeled or not. This comprises the simplified model proposed in [45], but it would also allow for choosing individual impulse response lengths for each modeled loudspeaker-to-microphone path. However, the derivation of the GFDAF algorithm in Sec. 4 is based on DFTs of the same lengths, which precludes choosing \mathbf{v} arbitrarily. Hence, for the GFDAF, it is only possible to model either all or none of the coefficients describing a certain loudspeaker-to-microphone path.

To derive an adaptation algorithm for pruned models, the error signal (10) must be modified according to

$$\mathbf{e}_s(k) = \mathbf{d}(k) - \mathbf{X}(k) \mathbf{V}^T \mathbf{V} \hat{\mathbf{h}}(n). \quad (69)$$

For the RLS algorithm, this results in the cost function

$$J_{\text{RLS}}^{\text{app}}(n) = \sum_{\nu=0}^n \lambda^{n-\nu} \mathbf{e}_s^H(\nu N) \mathbf{e}_s(\nu N), \quad (70)$$

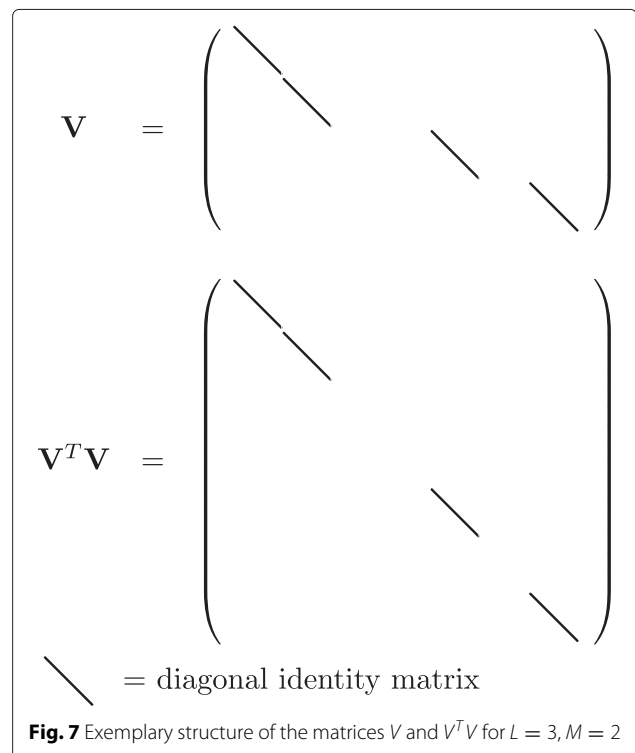


Fig. 7 Exemplary structure of the matrices \mathbf{V} and $\mathbf{V}^T \mathbf{V}$ for $L = 3, M = 2$

where the derivation of the adaptation algorithm uses exactly the same steps as shown above. Consequently, considering the derivation above, while replacing $\hat{\mathbf{h}}(n)$ by $\mathbf{V}\hat{\mathbf{h}}(n)$ and $\mathbf{X}(k)$ by $\mathbf{X}(k)\mathbf{V}^T$ results in the desired algorithm. The latter replacement implies a further replacement of $\hat{\mathbf{R}}(n)$ by $\mathbf{V}\hat{\mathbf{R}}(n)\mathbf{V}^T$.

Then, assuming $\mathbf{V}\hat{\mathbf{R}}(n)\mathbf{V}^T$ to be invertible results in

$$\mathbf{V}\hat{\mathbf{h}}(n) = \mathbf{V}\hat{\mathbf{h}}(n-1) + \left(\mathbf{V}\hat{\mathbf{R}}(n)\mathbf{V}^T\right)^{-1} \mathbf{V}\mathbf{X}^H(nN)\mathbf{e}'(nN). \quad (71)$$

At this point, a definition of the a priori estimation error for pruned LEM models might be expected, which would then be used instead of $\mathbf{e}'(nN)$. However, this is not necessary since (71) already implies that all unmodeled coefficients are set to zero in $\hat{\mathbf{h}}(n)$. It can be seen from (71) that the dimensions of the involved inverse can be reduced when using pruned models.

Multiplying (71) by \mathbf{V}^T from the left-hand side and requiring

$$\left(\mathbf{I}_{KLM} - \mathbf{V}^T\mathbf{V}\right)\hat{\mathbf{h}}(n) = \mathbf{0}_{KLM \times 1}, \quad (72)$$

which is implied by (66), leads to an explicit formulation of the algorithm given by

$$\hat{\mathbf{h}}(n) = \mathbf{V}^T\mathbf{V}\hat{\mathbf{h}}(n-1) + \mathbf{V}^T\left(\mathbf{V}\hat{\mathbf{R}}\mathbf{V}^T\right)^{-1} \mathbf{V}\mathbf{X}^H(nN)\mathbf{e}'(nN). \quad (73)$$

For pruned models, the gradient descent approach described in (31) is given by

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu\mathbf{V}^T\mathbf{V}\left(\mathbf{R} - \mathbf{r}\mathbf{V}^T\mathbf{V}\hat{\mathbf{h}}(n-1)\right). \quad (74)$$

Plugging (32) and (33) into (74) results in the LMS update

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu\mathbf{V}^T\mathbf{V}\mathbf{X}^H(nN) \cdot \left(\mathbf{d}(nN) - \mathbf{X}(nN)\mathbf{V}^T\mathbf{V}\hat{\mathbf{h}}(n-1)\right), \quad (75)$$

where (23) and (66) can be used to formulate the LMS algorithm for pruned models:

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu\mathbf{V}^T\mathbf{V}\mathbf{X}^H(nN)\mathbf{e}'(nN). \quad (76)$$

As the GFDAF algorithm approximates the RLS algorithm, the formulation of the RLS algorithm for simplified models can be straightforwardly translated to obtain

$$\hat{\mathbf{h}}(n) = \mathbf{V}^T\mathbf{V}\hat{\mathbf{h}}(n-1) + \mu\mathbf{W}_{10}^H\mathbf{V}^T\left(\mathbf{V}\left(\hat{\mathbf{S}}(n) + \mathbf{D}(n)\right)\mathbf{V}^T\right)^{-1} \cdot \mathbf{V}\mathbf{W}_{10}\mathbf{W}_{10}^H\mathbf{X}^H(nN)\mathbf{W}_{01}^H\mathbf{e}'(nN) \quad (77)$$

by comparing (57) and (73). Note that the term $\mathbf{V}^T\mathbf{V}$ can be omitted as long as $\hat{\mathbf{h}}(n-1)$ fulfills (66). The formulations of (62) and (64) can be obtained in the same manner, where (64) requires a redefinition of \mathbf{V} to consider the doubled number of coefficients in $\hat{\mathbf{h}}(n)$, compared to $\hat{\mathbf{h}}(n)$.

When comparing (73), (77), and (75) to (36) and (57), it can be seen that the relation of the regularized GFDAF algorithm to the LMS algorithm can also be established for simplified LEM models.

6 Evaluation results

In this section, a brief experimental evaluation of the treated adaptation algorithms is presented. This evaluation is focused on the effects of the approximations used in the derivation of the GFDAF algorithm. Hence, the RLS algorithm given by (25) is compared to the three presented variants of the GFDAF algorithm, given by (57), (62), and (64). To this end, the following AEC scenario is considered: two loudspeaker signals ($L = 2$) carry a stereo recording of a speech signal superimposed by mutually uncorrelated white Gaussian noise signals such that a signal-to-noise ratio (SNR) of 20 dB results on average. This rather low SNR was chosen to avoid using a regularization in the first two experiments that would otherwise obscure insights into the influence of the approximations used for the GFDAF algorithm. From the loudspeaker signals, two microphone signals ($M = 2$) have been obtained through convolution with four impulse responses, measured in a room with a reverberation time T_{60} of approximately 0.36 s.

Using a sampling frequency of 8 kHz, the impulse responses were truncated to 128 samples such that the adaptive filters could, in theory, perfectly model the impulse responses with the chosen $K = 128$. This choice was imposed by the large computational demands of the RLS algorithm. To simulate microphone noise, mutually uncorrelated white Gaussian noise signals were added to the microphones such that a signal-to-noise ratio of 40 dB results on average.

The three experiments last for a simulated time span of 60 s, where no adaptation is performed during the first 3 s in order to obtain sufficiently well-conditioned matrices $\hat{\mathbf{R}}(n)$ and $\hat{\mathbf{S}}(n)$ prior to their inversion. Note that no regularization was used ($\delta = 0$), unless stated otherwise, and both matrices, $\hat{\mathbf{R}}(n)$ and $\hat{\mathbf{S}}(n)$, were initialized with zero values. In the course of the experiment, two events are simulated to challenge the robustness of the algorithms. First, the impulse responses used to determine the microphone signals are exchanged at $t = 23$ s to investigate the robustness against sudden changes in the room impulse response. Second, a sample of snare drum hit is added to the microphone signals at $t = 43$ s as an example of a strong impulsive local source, where the maximum amplitude of the snare drum sample was chosen to be twice the maximum amplitude of the microphone signal. For the assessment, two measures have been considered, the echo return loss enhancement (ERLE) and the normalized system misalignment (NMA). The ERLE measures the AEC performance and is given by

$$\text{ERLE}(n) = 20 \log_{10} \left(\frac{\|\mathbf{d}(nN)\|_2}{\|\mathbf{e}(nN)\|_2} \right), \quad (78)$$

where $\|\cdot\|_2$ denotes the Euclidean norm. Note that the actual microphone signal was only used for the adaptation of the filters, while the noise-free microphone signal was used to determine the ERLE. This measure was termed “true ERLE” in [48] and allows to assess the echo cancellation performance also during perturbation. On the other hand, the NMA is defined by

$$\text{NMA}(n) = 20 \log_{10} \left(\frac{\|\hat{\mathbf{h}}(n) - \mathbf{h}\|_F}{\|\mathbf{h}\|_F} \right), \quad (79)$$

where $\|\cdot\|_F$ denotes the Frobenius norm and measures the system identification accuracy.

The results of this experiment can be seen in Fig. 8, where $\lambda = 0.99$, $\mu = 1$, $K = 128$, $P = 128$, $Q = 256$, and $N = 64$ have been chosen for all algorithms (if applicable). It can be seen that the RLS algorithm shows the best performance in terms of ERLE and NMA. This is an expected result since it uses no approximations. It can be furthermore seen that the approximation introduced with the algorithm described by (57) leads to a slower convergence. The approximations used for (62) leads to a further reduction in convergence speed, while the algorithm described by (64) shows nearly an identical performance compared to using (62). It can be seen that the reconvergence behavior after the impulse response change and the perturbation of the microphone signal is very similar to the initial convergence. However, it can be seen that the algorithm described by (57) shows a low

robustness at some time instants. Note that the breakdown in ERLE and NMA that exceed the scale of the plot reach up to -100 and 100 dB, respectively. After that, a stable reconvergence can be seen. An explanation for this sudden breakdown can be found when considering (55) in conjunction with (62), which describes a GFDAF variant that does not exhibit this property. It can be shown that

$$\hat{\mathbf{S}}^{-1}(n) \mathbf{X}^H(nN) \quad (80)$$

yields a matrix with entries that are weighted inversely proportional to the weights of the corresponding entries in $\mathbf{X}^H(nN)$. This is because $\mathbf{X}^H(nN)$ is also considered in $\hat{\mathbf{S}}(n)$, while all DFT bins are decoupled. However, when considering

$$\hat{\mathbf{S}}^{-1}(n) \mathbf{W}_{10} \mathbf{W}_{10}^H \mathbf{X}^H(nN), \quad (81)$$

as implied by (57), the DFT bins are no longer decoupled because of the time-domain windowing by $\mathbf{W}_{10} \mathbf{W}_{10}^H$. Thus, it is possible that a sharp spectral peak in $\mathbf{X}(nN)$ leaks into the neighboring bins in the product $\mathbf{W}_{10} \mathbf{W}_{10}^H \mathbf{X}^H(nN)$. On the other hand, $\hat{\mathbf{S}}^{-1}(n)$ does not describe a time-domain windowing, which implies that a sharp spectral peak in $\mathbf{X}(nN)$ will not be spread in $\hat{\mathbf{S}}(n)$. Due to this mismatch, the entries in the matrix resulting from (81) can exhibit a relatively strong weight, which implies larger adaptation steps and can lead to problems in some cases.

While comparing the considered algorithm using identical parameters for all of them allows to investigate the properties of the individual algorithms, it is not a fair performance comparison. Hence, an optimal step size μ for

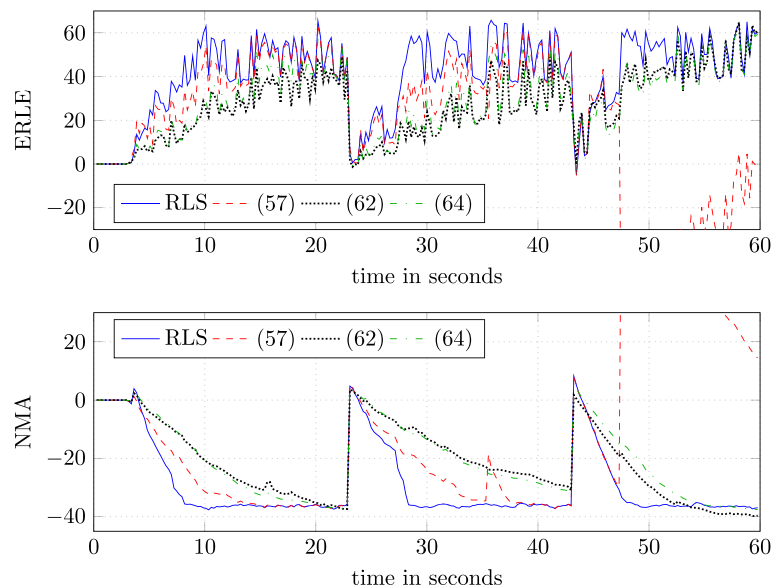


Fig. 8 ERLE and NMA for an AEC experiment with identical algorithm parameters

the variants of the GFDAF algorithm was determined by subsequent increasing μ until no further improvement in ERLE was noticeable. The optimal step sizes determined for (57), (62) and, (64) were $\mu = 1.2$, $\mu = 3.1$, and $\mu = 2.4$, respectively. This is actually a surprising result since approximations typically call for a more conservative step size. For performance evaluation, the experiment described above was repeated using these step sizes, while all other parameters were kept. The results presented in Fig. 9 show that all variants of the GFDAF algorithm are able to approach the performance of the RLS algorithm, which is shown for comparison. As expected, the robustness of the algorithm described by (57) is obviously even more reduced when μ is increased.

In Fig. 10, the experiment described for Fig. 8 was repeated, where the variants of the GFDAF algorithm have been regularized, while the RLS algorithm was not regularized to allow for a comparison. As explained above, larger values of δ will result in adaptation steps of the GFDAF algorithm that are closer to the adaptation steps the LMS algorithm would provide. Since the GFDAF algorithm is superior to the LMS algorithm in terms of convergence speed, δ should be chosen as low as possible. As the algorithms described by (62) and (64) do not depend strongly on a regularization in the considered scenario, $\delta = 0.03$ was chosen. This choice represents a compromise between the optimum regularization of the algorithm described by (57) and not hampering the convergence of the other algorithms.

As expected, the convergence speed of the all regularized algorithms is slightly reduced, while the impact of the impulse in the microphone signal is also slightly reduced

by the regularization. The most interesting results are those for the algorithm described by (57). While the regularization mitigates the robustness problem, it is not able to prevent the divergence completely. At the same time, the NMA achieved by this algorithm during normal convergence of this algorithm is improved, such that it achieves a better system identification than the RLS algorithm. Note that the RLS is optimal with respect to increasing ERLE, but not necessarily optimal to decrease the NMA, as it would be the case for the Kalman filter-based approaches.

Results of an evaluation with a varied microphone-signal segment size are not presented as the effect of varying the P was only marginal in the considered experimental scenario.

7 Real-world implementations of the GFDAF algorithm

In this section, some notes on the implementation of the GFDAF algorithm are given. The most attractive variants for implementation of this algorithm are given by (62) and (64), as also proposed in [39]. In both cases, the term $(\hat{\mathbf{S}}(n) + \mathbf{D}(n))^{-1} \mathbf{X}^H(nN)$ needs to be computed. Considering the dimensions of the involved matrices, it becomes clear that a real-world implementation must exploit sparsity in order to be feasible. The structures of the relevant matrices are illustrated in Figs. 4 and 6 and can be considered straightforwardly. For computing $(\hat{\mathbf{S}}(n) + \mathbf{D}(n))^{-1} \mathbf{X}^H(nN)$, all DFT-bins can be treated independently, which allows for a straightforward parallelization of the algorithm. This can be beneficial for the

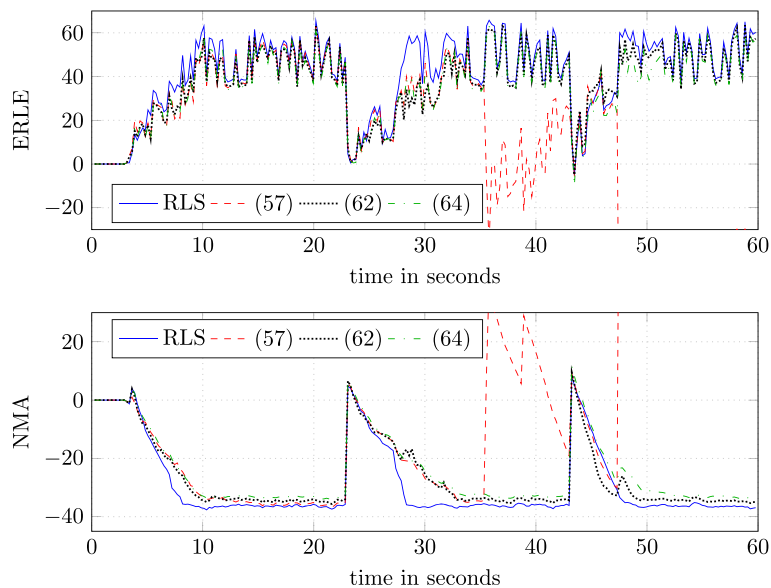


Fig. 9 ERLE and NMA for an AEC experiment with adjusted algorithm step sizes

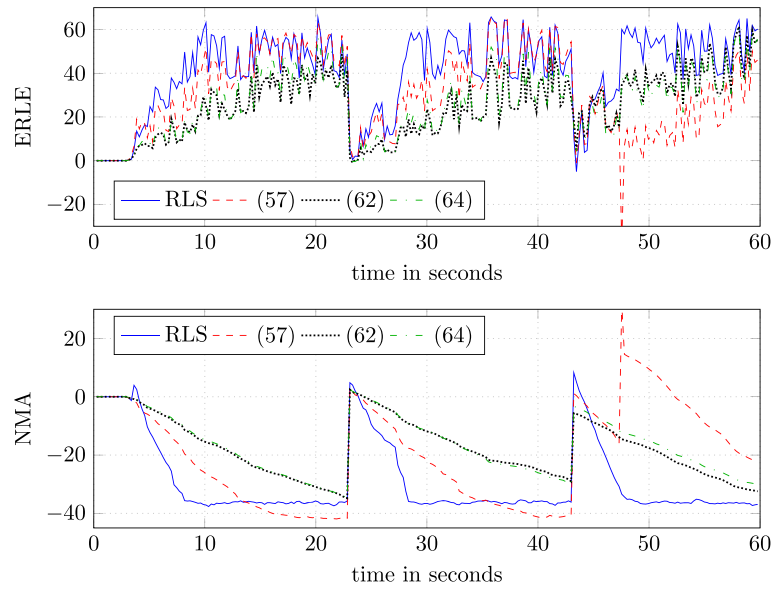


Fig. 10 ERLE and NMA for an AEC experiment with regularized algorithms

implementation of the algorithm on multi-core processors [49]. Since $\left(\hat{\mathbf{S}}(n) + \mathbf{D}(n)\right)$ is positive semi-definite, it is possible to use the Cholesky decomposition for an efficient computation.

Still, for $M > 1$, the definition (42) implies redundancy in $\underline{\mathbf{X}}(k)$ that is propagated into $\hat{\mathbf{S}}(n)$ such that it appears as if the derivation given above would not describe an efficient implementation. Equation 55 in conjunction with (42) and the well-known identity

$$\mathbf{AC} \otimes \mathbf{BD} = (\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) \quad (82)$$

can be used to show that $\hat{\mathbf{S}}(n)$ can also be obtained by

$$\hat{\mathbf{S}}(n) = \mathbf{I}_M \otimes \hat{\mathbf{S}}'(n), \quad (83)$$

where $\hat{\mathbf{S}}'(n)$ is equal to $\hat{\mathbf{S}}(n)$ for $M = 1$. Additionally, the Kronecker product has the property

$$(\mathbf{A} \otimes \mathbf{B})^{-1} = \mathbf{A}^{-1} \otimes \mathbf{B}^{-1}, \quad (84)$$

which implies that this redundancy does not increase the computational effort for inverting $\hat{\mathbf{S}}(n)$. Finally, the cost of computing $\left(\hat{\mathbf{S}}(n) + \mathbf{D}(n)\right)^{-1} \underline{\mathbf{X}}^H(nN)$ dominates the overall effort and is proportional to QL^3 . When accepting some restrictions on the regularization, this value may be reduced to QL^2 [39]. While this constitutes a considerable effort, it has to be considered that the RLS algorithm (25) would imply a computational effort proportional to $(KL)^2$, noting that typically $K, Q \gg L$.

When considering the term $\left(\mathbf{V} \left(\hat{\mathbf{S}}(n) + \mathbf{D}(n)\right) \mathbf{V}^T\right)$ in (77), it can be seen that (83) and (84) are not generally applicable to determine its inverse. Thus, care

must be taken that \mathbf{V} reduces the matrix dimensions sufficiently such that a computational advantage is achieved compared to general models. It has been shown that WDAF allows for sufficiently simplified models to decrease the computational demands [45]. When coupling W wave-domain loudspeaker signals to each wave-domain microphone signal, the cost of computing $\left(\mathbf{V} \left(\hat{\mathbf{S}}(n) + \mathbf{D}(n)\right) \mathbf{V}^T\right) \mathbf{V} \underline{\mathbf{X}}^H(nN)$ is proportional to QW^3M , which implies computational savings whenever $W^3M < L^3$. A value of $W = 3$ is already sufficient for many application scenarios [45].

8 Conclusions

The GFDAF algorithm was presented as an approximation of the block RLS algorithm with exponential windowing, such that the microphone signal block length can be chosen independently from the modeled impulse response length, as it is also possible for other adaptive filtering approaches. An error in the original derivation of the GFDAF algorithm was identified, and it was shown that the erroneous equality can be used as a reasonable approximation. Furthermore, it was shown that a Tikhonov regularization of the GFDAF algorithm has a close relation to the well-known LMS algorithm. The notation of the presented derivation was optimized for conciseness to allow further development of this algorithm. This was exploited to formulate the GFDAF algorithm for simplified LEM models, which constitutes an original contribution of this paper. Moreover, a newly found variant of the GFDAF algorithm, which omits an approximation inherent to the original derivation, potentially shows an increased

convergence speed, while some robustness issues still have to be solved. This can be an avenue for future research.

Appendix A: Approximating the inverse of a power spectral density matrix

For the derivation of the GFDAF algorithm, the following approximation is crucial:

$$\mathbf{W}_{10}^H \hat{\mathbf{S}}^{-1}(n) \approx \left(\mathbf{W}_{10}^H \hat{\mathbf{S}}(n) \mathbf{W}_{10} \right)^{-1} \mathbf{W}_{10}^H. \quad (85)$$

Unfortunately, (85) was mistaken for an equivalence in [39], and it was claimed that multiplying $\hat{\mathbf{S}}(n) \mathbf{W}_{10}$ from the right-hand side would prove this. However, $\hat{\mathbf{S}}(n) \mathbf{W}_{10}$ is a singular matrix which invalidates this proof. In the following, (85) is analyzed for the case $L = M = 1, Q = 2K$, which is chosen for the sake of brevity and can be straightforwardly extended to scenarios with different L, M, Q , and K .

Since the dimensions of $\hat{\mathbf{S}}(n)$ are larger than those of $\hat{\mathbf{R}}(n)$, a further matrix has been defined to represent $\hat{\mathbf{S}}(n)$ in the time domain:

$$\hat{\mathbf{R}}_2(n) = \mathbf{F}_Q^H \hat{\mathbf{S}}(n) \mathbf{F}_Q. \quad (86)$$

The definitions of $\hat{\mathbf{R}}(n)$ and $\hat{\mathbf{S}}(n)$ in (18) and (46), respectively, (40) and (44) can be used to obtain

$$\hat{\mathbf{R}}_2(n) = \sum_{v=0}^n \lambda^{n-v} \left(\mathbf{X}_l(vN) \mathbf{X}_l^{(C)}(vN) \right)^H \cdot \left(\mathbf{X}_l(vN) \mathbf{X}_l^{(C)}(vN) \right) \quad (87)$$

$$= \begin{pmatrix} \hat{\mathbf{R}}(n) & \hat{\mathbf{R}}_C(n) \\ \hat{\mathbf{R}}_C^H(n) & \hat{\mathbf{R}}_{CC}(n) \end{pmatrix}. \quad (88)$$

To determine the inverse of $\hat{\mathbf{R}}_2(n)$, the block-matrix inversion can be used. It is given by

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{A}' & \mathbf{B}' \\ \mathbf{C}' & \mathbf{D}' \end{pmatrix}, \quad (89)$$

with

$$\mathbf{A}' = (\mathbf{A} - \mathbf{B} \mathbf{D}^{-1} \mathbf{C})^{-1}, \quad (90)$$

$$\mathbf{B}' = -\mathbf{A}^{-1} \mathbf{B} (\mathbf{D} - \mathbf{C} \mathbf{A}^{-1} \mathbf{B})^{-1}, \quad (91)$$

$$\mathbf{C}' = -\mathbf{D}^{-1} \mathbf{C} (\mathbf{A} - \mathbf{B} \mathbf{D}^{-1} \mathbf{C})^{-1}, \quad (92)$$

$$\mathbf{D}' = (\mathbf{D} - \mathbf{C} \mathbf{A}^{-1} \mathbf{B})^{-1}, \quad (93)$$

where $\mathbf{A}, \mathbf{B}, \mathbf{C}$, and \mathbf{D} are arbitrary matrices of compatible dimensions. Considering $\mathbf{W}_{10} \mathbf{W}_{10}^H$ in (85), it is clear that only \mathbf{A}' and \mathbf{B}' are relevant in our case, which are given by

$$\mathbf{A}' = \left(\hat{\mathbf{R}}(n) - \hat{\mathbf{R}}_C(n) \hat{\mathbf{R}}_{CC}^{-1}(n) \hat{\mathbf{R}}_C^H(n) \right)^{-1}, \quad (94)$$

$$\mathbf{B}' = -\hat{\mathbf{R}}^{-1}(n) \hat{\mathbf{R}}_C(n) \cdot \left(\hat{\mathbf{R}}_{CC}(n) - \hat{\mathbf{R}}_C^H(n) \hat{\mathbf{R}}^{-1}(n) \hat{\mathbf{R}}_C(n) \right)^{-1}, \quad (95)$$

The matrices $\hat{\mathbf{R}}(n)$ and $\hat{\mathbf{R}}_{CC}(n)$ estimate the autocorrelation matrices of $\mathbf{X}_l(vN)$ and $\mathbf{X}_l^{(C)}(vN)$, while $\hat{\mathbf{R}}_C(n)$ describes the cross-correlation between both. Assuming that $\hat{\mathbf{R}}(n)$ and $\hat{\mathbf{R}}_{CC}(n)$ are well-conditioned, while their entries exhibit significantly stronger weights than those in $\hat{\mathbf{R}}_C(n)$, the terms $\hat{\mathbf{R}}_C(n) \hat{\mathbf{R}}_{CC}^{-1}(n) \hat{\mathbf{R}}_C^H(n)$ and $\hat{\mathbf{R}}_C^H(n) \hat{\mathbf{R}}^{-1}(n) \hat{\mathbf{R}}_C(n)$ are of no importance. Hence, \mathbf{A}' approximates $\hat{\mathbf{R}}^{-1}(n)$ while the influence of \mathbf{B}' is small, which justifies the use of (85) as an approximation.

Abbreviations

AEC: acoustic echo cancellation; DFT: discrete Fourier transform; ERLE: echo return loss enhancement; GFDAF: generalized frequency-domain adaptive filtering; IPNLMS: improved proportionate normalized least-mean squares algorithm; LEM: loudspeaker-enclosure-microphone; LMS: least mean square; MSE: mean square error; NLMS: normalized least-mean-square; NMA: normalized system misalignment; PNLMs: proportionate normalized least-mean-squares algorithm; RLS: recursive least-squares; SNR: signal-to-noise ratio; WDAF: wave-domain adaptive filtering.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

Martin Schneider is currently with the Fraunhofer Institute for Integrated Circuits IIS, Germany.

Received: 30 June 2015 Accepted: 30 December 2015

Published online: 15 January 2016

References

1. E Hänsler, The hands-free telephone problem—an annotated bibliography. *Signal Process.* **27**(3), 259–271 (1992)
2. J Benesty, T Gänslér, DR Morgan, MM Sondhi, SL Gay, *Advances in network and acoustic echo cancellation*. (Springer, Berlin, Germany, 2001)
3. E Hänsler, G Schmidt, *Acoustic echo and noise control: a practical approach*. (Wiley, Hoboken (NJ), USA, 2004)
4. P Vary, R Martin, *Digital speech transmission: enhancement, coding and error concealment*. (Wiley, Hoboken (NJ), USA, 2006)
5. E Hänsler, G Schmidt, *Topics in acoustic echo and noise control: selected methods for the cancellation of acoustical echoes, the reduction of background noise, and speech processing*. (Springer, Berlin, Germany, 2006)
6. MM Sondhi, in *Springer Handbook of Speech Processing*. Adaptive echo cancellation for voice signals (Springer, Berlin, Germany, 2008), pp. 903–928
7. G Enzner, H Buchner, A Favrot, F Kuech, in *Academic press library in signal processing: image, video processing and analysis, hardware, audio, acoustic and speech Processing*, ed. by S Theodoridis, R Chellappa. Acoustic echo control, vol. 4 (Academic Press, Waltham (MA), USA, 2014)
8. SF Boll, Suppression of acoustic noise in speech using spectral subtraction. *IEEE Trans. Acoustics, Speech and Signal Processing.* **27**(2), 113–120 (1979)
9. C Faller, J Chen, Suppressing acoustic echo in a spectral envelope space. *IEEE Trans. Speech and Audio Processing.* **13**(5), 1048–1062 (2005)
10. S Gustafsson, R Martin, P Vary, Combined acoustic echo control and noise reduction for hands-free telephony. *Signal Process.* **64**(1), 21–32 (1998)
11. E Hänsler, GU Schmidt, Hands-free telephones—joint control of echo cancellation and postfiltering. *Signal process.* **80**(11), 2295–2305 (2000)
12. S Gustafsson, R Martin, P Jax, P Vary, A psychoacoustic approach to combined acoustic echo cancellation and noise reduction. *IEEE Trans. Speech and Audio Processing.* **10**(5), 245–256 (2002)
13. G Enzner, P Vary, in *Proc. European Signal Processing Conf. (EUSIPCO)*. New insights into the statistical signal model and the performance bounds of acoustic echo control (IEEE, Antalya, Turkey, 2005), pp. 1–4
14. G Enzner, P Vary, Frequency-domain adaptive Kalman filter for acoustic echo control in hands-free telephones. *Signal Process.* **86**(6), 1140–1156 (2006)
15. J Wung, TS Wada, B-H Juang, B Lee, T Kalker, RW Schafer, in *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*. A system

- approach to residual echo suppression in robust hands-free teleconferencing (IEEE, Prague, Czech Republic, 2011), pp. 445–448
16. MM Sondhi, AJ Presti, A self-adaptive echo canceller. *Bell Syst. Tech. J.* **45**(10), 1851–1854 (1966)
 17. S Haykin, *Adaptive filter theory*. (Prentice Hall, Englewood Cliffs (NJ), USA, 2001)
 18. E Hänsler, *Statistische Signale: Grundlagen und Anwendungen*. (Springer, Berlin, Germany, 2001)
 19. K Ozeki, T Umeda, An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties. *Electron. Commun. in Japan (Part I: Communications)*. **67**(5), 19–27 (1984)
 20. S Werner, JA Apolinário Jr, PSR Diniz, Set-membership proportionate affine projection algorithms. *EURASIP J. Audio, Speech, and Music Processing*. **2007**(1), 10–10 (2007)
 21. SL Gay, S Tavathia, in *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*. The fast affine projection algorithm, vol. 5 (IEEE, Detroit (MI), USA, 1995), pp. 3023–3026
 22. DL Duttweiler, Proportionate normalized least-mean-squares adaptation in echo cancelers. *IEEE Trans. Speech and Audio Processing*. **8**(5), 508–518 (2000)
 23. J Benesty, SL Gay, in *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*. An improved PNLMS algorithm, vol. 2 (IEEE, Orlando (FL), USA, 2002), pp. 1881–1884
 24. E Hänsler, in *IEEE International Symposium on Circuits and Systems*. Adaptive echo compensation applied to the hands-free telephone problem, vol. 1 (IEEE, New Orleans (LA), USA, 1990), pp. 279–282
 25. C Breining, P Dreiseitel, E Hänsler, A Mader, B Nitsch, H Puder, T Schertler, G Schmidt, J Tilp, Acoustic echo control. an application of very-high-order adaptive filters. *IEEE Signal Proc. Mag.* **16**(4), 42–69 (1999)
 26. W Kellermann, Kompensation akustischer Echos in Frequenzteilbändern. *Frequenz*. **39**(7–8), 209–215 (1985)
 27. W Kellermann, in *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*. Analysis and design of multirate systems for cancellation of acoustical echoes, vol. 5 (IEEE, New York (NY), USA, 1988), pp. 2570–2573
 28. JJ Shynk, Frequency-domain and multirate adaptive filtering. *IEEE Signal Process. Mag.* **9**(1), 14–37 (1992)
 29. P Sommen, P van Gerwen, H Kotmans, A Janssen, Convergence analysis of a frequency-domain adaptive filter with exponential power averaging and generalized window function. *IEEE Trans. Circuits Syst.* **34**(7), 788–798 (1987)
 30. E Ferrara, Fast implementations of LMS adaptive filters. *IEEE Trans. Acoustics, Speech, and Signal Processing*. **28**(4), 474–475 (1980)
 31. E Moulines, O Ait Amrane, Y Grenier, The generalized multidelay adaptive filter: structure and convergence analysis. *IEEE Trans. Signal Processing*. **43**(1), 14–28 (1995)
 32. J-S Soo, KK Pang, Multidelay block frequency domain adaptive filter. *IEEE Trans. Acoustics, Speech and Signal Processing*. **38**(2), 373–376 (1990)
 33. G Enzner, in *Proc. European Signal Processing Conf. (EUSIPCO)*. Bayesian inference model for applications of time-varying acoustic system identification (IEEE, Aalborg, Denmark, 2010), pp. 2126–2130
 34. Cn Paleologu, J Benesty, S Ciochina, Study of the general Kalman filter for echo cancellation. *IEEE Trans. Audio, Speech, and Language Processing*. **21**(8), 1539–1549 (2013)
 35. A Mader, H Puder, GU Schmidt, Step-size control for acoustic echo cancellation filters—an overview. *Signal Process.* **80**(9), 1697–1719 (2000)
 36. F Kuech, E Mabande, G Enzner, in *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*. State-space architecture of the partitioned-block-based acoustic echo controller, (2014), pp. 1295–1299
 37. J Benesty, F Amand, A Gilloire, Y Grenier, in *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*. Adaptive filtering algorithms for stereophonic acoustic echo cancellation, vol. 5 (IEEE, Detroit (MI), USA, 1995), pp. 3099–3102
 38. J Benesty, DR Morgan, MM Sondhi, A better understanding and an improved solution to the specific problems of stereophonic acoustic echo cancellation. *IEEE Trans. Speech and Audio Process.* **6**(2), 156–165 (1998)
 39. H Buchner, J Benesty, W Kellermann, ed. by J Benesty, Y Huang. *Adaptive Signal Processing: Application to Real-World Problems* (Springer, Berlin, Germany, 2003)
 40. M Dentino, J McCool, B Widrow, Adaptive filtering in the frequency domain. *Proc. IEEE*. **66**(12), 1658–1659 (1978)
 41. D Mansour, A Gray Jr., Unconstrained frequency-domain adaptive filter. *IEEE Trans. Acoustics, Speech, and Signal Processing*. **30**(5), 726–734 (1982)
 42. J Benesty, P Duhamel, A fast exact least mean square adaptive algorithm. *IEEE Trans. Signal Process.* **40**(12), 2904–2920 (1992)
 43. S Malik, G Enzner, Recursive Bayesian control of multichannel acoustic echo cancellation. *IEEE Signal Process. Letters*. **18**(11), 619–622 (2011)
 44. H Buchner, S Spors, W Kellermann, in *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*. Wave-domain adaptive filtering: acoustic echo cancellation for full-duplex systems based on wave-field synthesis, vol. 4 (IEEE, Montreal, Canada, 2004), pp. 117–120
 45. M Schneider, W Kellermann, in *Proc. Joint Workshop on Hands-free Speech Communication and Microphone Arrays (HSCMA)*. A wave-domain model for acoustic MIMO systems with reduced complexity (IEEE, Edinburgh, UK, 2011), pp. 133–138
 46. RFH Fischer, *Precoding and signal shaping for digital transmission*. (Wiley, Hoboken (NJ), USA, 2002)
 47. M Montazeri, P Duhamel, A set of algorithms linking NLMS and block RLS algorithms. *IEEE Trans. Signal Process.* **43**(2), 444–453 (1995)
 48. P Thune, G Enzner, in *Proc. Intl. Symposium on Image and Signal Processing and Analysis (ISPA)*. Trends in adaptive MISO system identification for multichannel audio reproduction and speech communication (IEEE, Berlin, Germany, 2013), pp. 767–772
 49. M Schneider, F Schuh, W Kellermann, in *ITG-Fachbericht Sprachkommunikation*. The generalized frequency-domain adaptive filtering algorithm implemented on a GPU for large-scale multichannel acoustic echo cancellation (VDE, Braunschweig, Germany, 2012), pp. 39–42

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com