FREQUENCY-DOMAIN ADAPTIVE FILTERING: FROM REAL TO HYPERCOMPLEX SIGNAL PROCESSING

Danilo Comminiello, Michele Scarpiniti, Raffaele Parisi, and Aurelio Uncini

Dept. Information Engineering, Electronics and Telecommunications
Sapienza University of Rome
Via Eudossiana, 18 - 00184 Rome, Italy

ABSTRACT

Frequency-domain adaptive filters (FDAFs) have been widely used over the years, but they are still matter of research due to their powerful capabilities that differentiate them from the whole family of time-domain adaptive filters. This paper aims at providing an overview on FDAFs through a unifying framework that can be used for the derivation of the most popular algorithms of the FDAF family and enables the processing of a wide variety of signals, from real-valued ones to complex- and hypercomplex-valued signals. In particular, we focus on a recent class of FDAFs in the quaternion domain and we show how to derive it from the described framework. Moreover, we evaluate the application of the derived quaternion FDAF to the processing of 3D audio signals. Experimental results show the effectiveness of the proposed adaptive filter in estimating the inverse of a multidimensional acoustic impulse response.

Index Terms— Frequency-domain adaptive filter, Quaternion adaptive filtering, 3D audio, Adaptive signal processing, Hypercomplex signal processing, DSP education.

1. INTRODUCTION

Adaptive filtering has always been an important research topic in the last half the century [1–6]. However, it has also a great impact on digital signal processing (DSP) education, since it requires advanced interdisciplinary knowledge [7]. Indeed, it often characterizes both classical and advanced signal processing faculty courses.

Online adaptive filters are characterized by a parameter updating that is performed at each time instant n, when a new signal sample is fed as input. The most popular adaptive filters are the time-domain ones (e.g., least mean square (LMS), recursive least square (RLS) and affine projection algorithm (APA), among others), in which the filter impulse response, denoted as $\mathbf{w}_n \in \mathbb{R}^{M\times 1}$, being M the filter length, is time variant and the convolution algorithm is implemented directly in the time domain [1–6]. Let $\mathbf{x}_n \in \mathbb{R}^{M\times 1}$ be the filter input, the filter output is determined by a simple scalar product: $y[n] = \mathbf{w}_{n-1}^T \mathbf{x}_n$. One of the main drawbacks of these filters is that the computational complexity, proportional to the filter length, can become prohibitive for significantly long filters, especially for real-time applications.

A reduction of the computational cost is given by the *block* (or also said *mini-batch*) adaptive filters, like the block LMS, which are characterized by a periodic update rule [8, 9]. The filter coefficients, indeed, are updated only every L samples. Denoting with k the block index, the filter output is returned in blocks with length L, as the convolution sum is implemented as $\mathbf{y}_k = \mathbf{X}_k \mathbf{w}_k$, where $\mathbf{y}_k \in \mathbb{R}^{L\times 1}$ is the signal output, $\mathbf{w}_k$ represents a static filter for all the rows of the signal matrix $\mathbf{X}_k \in \mathbb{R}^{L\times M}$.

However, when dealing with real-time application problems, the best solution is provided by using frequency-domain adaptive filters (FDAFs) [9–13], which are capable of reducing significantly the computational complexity, while keeping comparable performance with time-domain filters. Rather, FDAFs may even achieve convergence performance improvement by simply involving a frequency-bin power normalization procedure. The main drawback of such filters is the introduction of a systematic delay between the input and output signals, due to the intrinsic block approach, which is the same problem affecting also time-domain mini-batch filters.

Early works on FDAFs were mainly focused to solve problems using reduced computational complexity with respect to classic time-domain algorithms [14–18]. Hence, the main goal was to save resources rather than improving performance. However, nowadays, despite the significantly larger availability of computational resources with respect to some decades ago, the research and development on frequency-domain algorithms have never slowed down. Indeed, we can find recent literature on FDAF-based models aiming at improving modeling performance [19–22], addressing emerging and complex signal processing problems [22, 23], dealing with high-resolution data, multichannel and multidimensional signals [24–27]. Thus, we have now FDAFs with higher efficiency and effectiveness.

Despite their powerful capabilities that make them suitable for several real-time application fields, FDAFs are not always easy to be clearly explained in adaptive filtering courses due to the additional theoretical concepts involved with respect to time-domain adaptive filters (e.g., buffer composition and signal transformations, above all) and also due to the more complex notation.

The aim of this paper is twofold. Firstly, we provide a unifying framework for the class of FDAF algorithms that can be adopted in DSP education to introduce the concepts of block filtering, signal buffering and transformation of signals into a different domain. This may help to approach frequency-domain filtering even to those students who may not have all the theoretical knowledge necessary to cover every aspect introduced in this topic. Secondly, starting from the proposed framework that holds for real-, complex- and hypercomplex-valued domains, we show how to derive even new frequency-domain algorithms. In particular, we derive and propose a novel FDAF in the quaternion domain and we assess its effectiveness in a 3D audio processing problem.

The paper is organized as follows. In Section 2, we introduce a unifying approach for the frequency-domain block filtering, starting from which we describe how to derive a classic overlap-save FDAF in Section 3. The, in Section 4, we show how to exploit the framework and simply extend the OS-FDAF to derive a quaternion-domain FDAF. Experimental results on 3D audio processing are shown in Section 5, and our conclusion is finally drawn in Section 6.

2. A UNIFYING FRAMEWORK FOR THE CLASS OF FREQUENCY-DOMAIN ADAPTIVE FILTERS

Frequency-domain algorithms are defined starting from the same theoretical assumptions of time-domain ones. However, very often they are not derived as a simple redefinition "in frequency" of the time-domain algorithms. Indeed, frequency-domain algorithms have their own structures and properties that may be even rather different compared to the related time-domain algorithms. However, a similarity can be found for the FDAFs with the time-domain block algorithms, since all the frequency-domain algorithms need a block filtering approach to process the input signal.

The block filtering approach requires an appropriate filling mechanism of *memory buffers*, hereinafter simply *buffers*, which contain the input/output signal blocks to be processed. In addition, the transformation operator, here indicated as *transformation matrix* $\mathbf{F} \in \mathbb{C}^{N\times N}$, requires the variables redefinition in the new domain. These aspects, along with others discussed below and related to the filtering algorithms, involve the proliferation of indices, symbols and new variables that can sometimes burden the formalism. To this end, we introduce a unifying framework, depicted in Fig. 1, which includes the concepts of buffering and transformation and enables the derivation of the whole family of FDAFs.

FDAFs, like mini-batch algorithms, operate on a L-sample signal block, but the (possible) domain transformation can be made by considering a buffer of greater length. In general terms, the transformation can be performed on a signal segment (or *running window*) composed by L new samples (i.e., the new input block) and possibly by a block of M past samples. Therefore, as shown in Fig. 1(a), the input buffer of length N includes the new block of L samples, M samples belonging to the previous block and the presence of a *buffer composition mechanism* implemented by a simple blockwise shift and up-date operation.

The framework in Fig. 1(b) involves the quantities:

- The linear transform operator $\mathbf{F} \in \mathbb{C}^{N\times N}$ (e.g., discrete-Fourier transform (DFT) matrix) such that $\mathbf{F}\cdot\mathbf{F}^H = \mathbf{I}$;
- A *windowing constraint* $\mathbf{G}$ that may be used for output and error signals and also for the weights;
- Sequence block vectors $\mathbf{x}_k$, $\mathbf{y}_k \in \mathbb{R}^{L\times 1}$ and $\mathbf{w}_k \in \mathbb{R}^{M\times 1}$, respectively, for input, output and filter weights;
- Time-domain input data matrix $\mathbf{X}_k \in \mathbb{R}^{N\times M}$;
- Frequency-domain corresponding structures $\boldsymbol{X}_k, \boldsymbol{Y}_k, \boldsymbol{W}_k \in \mathbb{R}^{N\times 1}$

Said M_F the filter length, for the FDAF algorithms, the block length is generally chosen as $L = M_F$ and the FDAF buffer composition is usually chosen such that $L = M \equiv M_F$. To operate a correct domain transformation, for example with a DFT/FFT, and in particular for the filter output calculation, it is necessary to choose a number of FFT points $N \geq M + L - 1$. A usual choice for FDAF class, is $N = M + L$. From the described framework, several FDAF-based algorithms can be easily derived, as discussed below.

Let us consider the case of very long filters (with thousands of coefficients), which is rather usual in real applications, the block length turns out to be necessarily $L \ll M_F$, thus it is necessary to perform an impulse response partition for the FDAF implementation. The resulting *partitioned block FDAF* (PBFDAF) algorithm thus enables a block latency reduction. A usual choice is to consider: a number of P partitions of length M, such that the filter length is equal to the product $M_F = P\cdot M$, and a block length such that $M = p\cdot L$, with $p = 1, 2, \ldots, P$ [12,17,18,28].

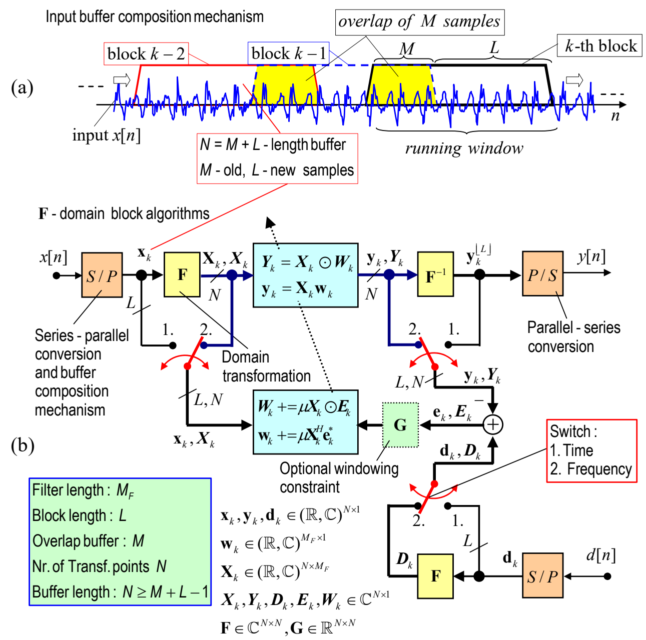

Fig. 1. Block adaptive filtering algorithms. (a) Input signal buffer composition mechanism. (b) Unifying framework for *block algorithms* in time and in **F**-transform domain. If the operator $\mathbf{F} = \mathbf{I}$, the algorithm is entirely in the time domain and the switches 1. or 2. position, is indifferent. For $\mathbf{F} \neq \mathbf{I}$, the weights adaptation can be done in the time-domain (switch position 1.) or in the transformed domain (switch position 2.). (Modified from [6]).

In the extreme case of $L = 1$ (i.e., a block of 1-sample length), the FDAF algorithm is defined as *transform domain adaptive filter* (TDAF) [29,30]. The input window (also said *sliding window* here), is simply defined by the filter delay-line length. In this case, the operator $\mathbf{F}$ performs a linear transformation just to orthogonalize the input signal so as to facilitate the uniform convergence of the adaptive algorithm. The change of domain can be of various nature. Although, in theory, the operator $\mathbf{F}$ can be any orthonormal transformation, it is usual to choose transformations that imply, in addition to the input signal orthogonalization, a computational complexity reduction. Usual choices are the DFT (implemented as FFT), the DCT or any other transformations tending to input signal orthogonalization. It is worth noting that for $L = 1$, the transformation $\mathbf{F}$ can be replaced by a suitably designed parallel filter bank, uniformly or not non-uniformly spaced [30]. In addition, in order to obtain a computational cost reduction, it is possible to perform a signal decimation/interpolation. The resulting filter class in this case is called *subband adaptive filter* (SAF) (see for example [2–6]).

3. DERIVATION OF THE OS-FDAF

We derive now a well-known FDAF algorithm starting from the framework introduced in the previous section.

3.1. The Frequency-Domain Adaptive Filter

The FDAF has a recursive formulation similar to block LMS (BLMS), also known in the literature as *fast LMS* [8, 9]. The

Algorithm 1 Implementation procedure of the OS-FDAF.

Initialization $\boldsymbol{W}_0 = \mathbf{0}$, $P_0(m) = \delta_m \; \forall m$
for $k = 0, 1, \ldots$ *for each block of L samples* **do**
  $\mathbf{x}_k \leftarrow [\mathbf{x}_{old}^{(M)} \; \mathbf{x}_{new}^{(L)}]$ *buffer composition rule*
  $\boldsymbol{X}_k = \mathrm{FFT}\,[\mathbf{x}_k]$ *fast Fourier transform*
  $\mathbf{y}_k = [\mathrm{IFFT}(\boldsymbol{X}_k \boldsymbol{W}_k)]^{\lfloor L \rfloor}$ *convolution*
  $\boldsymbol{E}_k = \mathrm{FFT}\,([\mathbf{0} \;\; \mathbf{d}_k - \mathbf{y}_k])$ *frq. domain error*
  $P_k(m) = \lambda P_{k-1}(m) + (1-\lambda)\left|X_k(m)\right|^2, \; \forall m$
  $\boldsymbol{\mu}_k = \mu \mathrm{diag}\left\{\left[P_k^{-1}(0) \; P_k^{-1}(1) \; \cdots \; P_k^{-1}(N-1)\right]\right\}$
  $\nabla J = \boldsymbol{\mu}_k \boldsymbol{X}_k^H \boldsymbol{E}_k$ *stochastic gradient*
  $\nabla J = \mathrm{FFT}\left(\begin{bmatrix} \mathrm{IFFT}\,[\nabla J]^{\lceil M \rceil} \\ \mathbf{0}_L \end{bmatrix}\right)$ *grad. constraint*
  $\boldsymbol{W}_{k+1} = \boldsymbol{W}_k + \nabla J$ *frq. domain up-date rule*
end for

learning rule of the time-domain BLMS can be written as $\nabla J_k = \sum_{i=0}^{L-1} e^\star[kL+i]\mathbf{x}_{kL+i}$, i.e., the gradient estimate is determined by the cross-correlation between the data vector $\mathbf{x}_k$ and the error $\mathbf{e}_k$. Thus the weight update equation can be formulated as $\mathbf{w}_{k+1} = \mathbf{w}_k + \frac{\mu_B}{L}\mathbf{X}_k^H \mathbf{e}_k^\star$. Transforming this rule in the frequency domain (see for example [13]) and using a compact and general notation [6], we obtain

$$\boldsymbol{W}_{k+1} = \boldsymbol{W}_k + \mathbf{G}\left[\boldsymbol{\mu}_k \boldsymbol{X}_k^H \boldsymbol{E}_k^\star\right] \tag{1}$$

where $[\cdot]^H$ is the Hermitian operator, $\boldsymbol{\mu}_k$ is a diagonal matrix $\boldsymbol{\mu}_k = \mathrm{diag}\left\{[\mu_k(0), \mu_k(1), \ldots, \mu_k(N-1)]\right\}$ containing the learning rates (or step sizes) that can assume different values for each frequency bin. The matrix $\mathbf{G}$ represents the windowing or *gradient constraint*, which is necessary to impose the linearity of the correlation in the gradient calculation. It can be interpreted as a particular signal pre-windowing in the time domain and it is inserted in learning rule only to generalize the FDAF formalism.

In the class of the FDAF algorithms the error calculation can be performed directly in the time or frequency domain. In the case where the error is calculated in frequency domain, the gradient constraint can be chosen unitary $\mathbf{G} = \mathbf{I}$ and the FDAF is said *unconstrained frequency domain adaptive filter* (UFDAF) [15].

3.2. Frequency-Bin Step-Size Normalization

One of the main advantages of the frequency approach is that the adaptation equations (1) are decoupled, i.e., in the frequency domain, the convergence of each filter coefficient is not dependent on the other ones. It follows that to speed-up the convergence rate, such that we can obtain a uniform convergence, it is possible to define a simple *power normalization rule*. Indicating with $P_k(m)$ the estimated power of the m-th frequency bin and, let μ a suitable predetermined scalar parameter, the step size can be chosen independently for each frequency bin m, proportional to the inverse of its power, i.e., $\mu_k(m) = \mu/[\alpha + P_k(m)]$, $m = 0, \ldots, N-1$, where the parameter $0 < \alpha \ll 1$ avoids divisions by zero. So, the power normalization rule allows to accelerate the slower convergence modes. Obviously, in the case of white and stationary input processes, the powers are identical for all frequencies bin and we have $\boldsymbol{\mu}_k = \mu\mathbf{I}$. Moreover, to avoid significant step-size discontinuity that could destabilize the adaptation, as suggested in [14], it is appropriate to estimate m-th power frequency bin with a one-pole low-pass smoothing filter usually implemented by the following finite difference equation

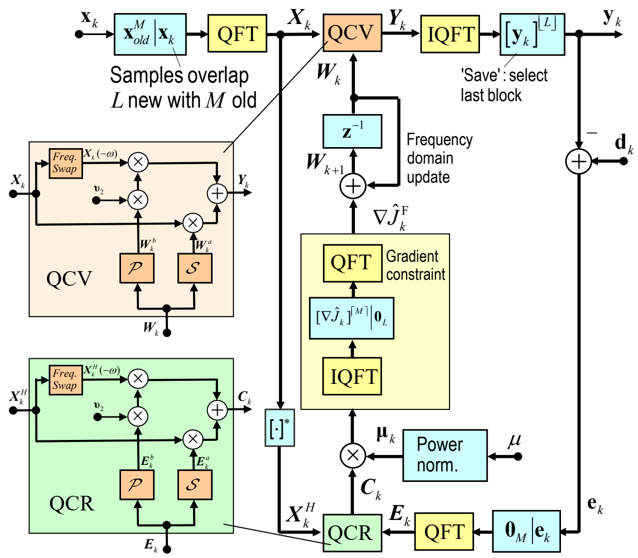

Fig. 2. Overlap-Save QFDAF block diagram. With $\mathcal{S}$ and $\mathcal{P}$ we denote the operators that extract the *simplex* and *perplex* parts of a quaternion, respectively [25].

$$P_k(m) = \lambda P_{k-1}(m) + (1-\lambda)|X_k(m)|^2 \tag{2}$$

where λ represents a forgetting parameter and $|X_k(m)|^2$ the m-th measured energy bin.

3.3. Overlap-Save FDAF Algorithm

The *overlap-save FDAF* (OS-FDAF) algorithm is the frequency-domain equivalent version of the BLMS, since it has the same convergence characteristics in terms of speed, stability, misalignment. It converges, in the mean, to the optimum Wiener solution [13]. The possibility of choosing different learning rates for each frequency bin, as for the power normalization (2), allows a convergence speed improvement without, however, further improving the minimum mean-square error (MSE). Compared to the BLMS, the OS-FDAF shows the dual advantage of reduced complexity and higher convergence speed (due to the step-size normalization). However, as the FFT is calculated for each signal block, the main drawback is that the algorithm introduces a systematic delay between the input and output of the filter of (at least) L samples.

In the implementation, the constraint matrix $\mathbf{G}$ does not appear explicitly . The FFT is used instead. An explicit determination of $\mathbf{G}$ would lead to loose the computational cost reduction inherent to the FFT calculation.

The algorithm implementation is described in Algorithm 1, where $\mathbf{e}_k, \mathbf{d}_k \in \mathbb{R}^{L\times 1}$ are the time domain error and desired output of the k-th sample block.

4. QUATERNION FDAF

In recent years, a large interest has arisen in the implementation of quaternion adaptive filtering, which led to the extension of several time-domain algorithms into the quaternion domain [31, 32].

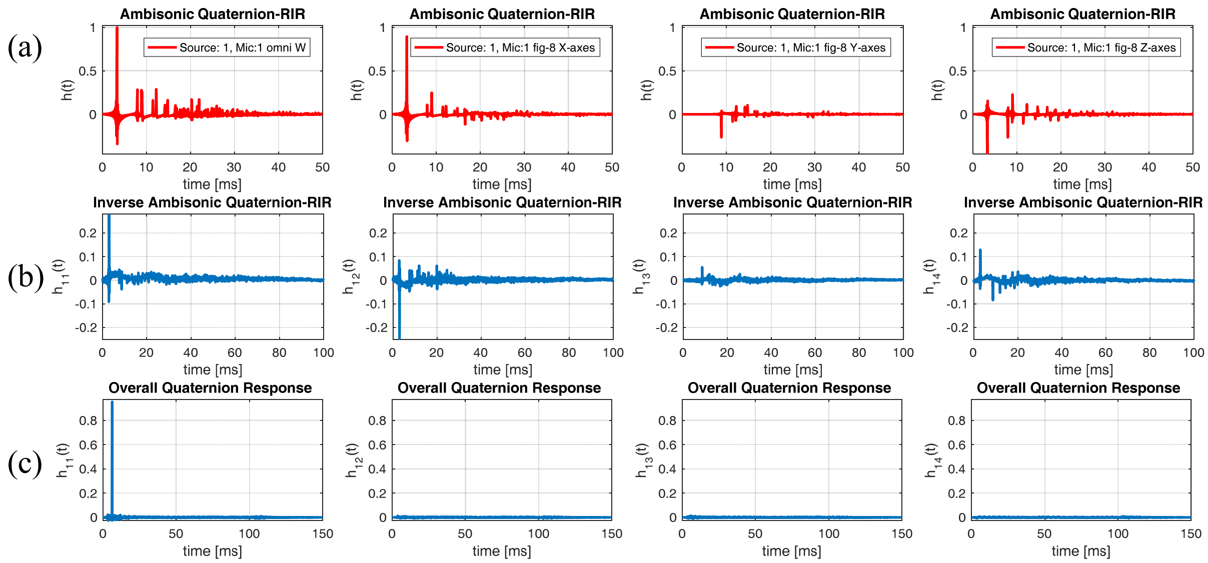

Fig. 3. Quaternion-Ambisonic room impulse response (RIR). (a) B-Format RIR. (b) Estimated inverse of RIR (Inv-RIR) by overlap-save QFDAF. (c) Quaternion convolution between RIR and Inv-RIR.

Introduced in 1843 by Sir William Rowan Hamilton, quaternion algebra is based on the fundamental formula with the symbols $\hat{\imath}, \hat{\jmath}, \hat{\kappa}$; namely, $\hat{\imath}^2 = \hat{\jmath}^2 = \hat{\kappa}^2 = \hat{\imath}\hat{\jmath}\hat{\kappa} = -1$. Quaternions, extend the complex numbers with four components, one scalar and three imaginary parts, defined as $q \in \mathbb{H} = q_0 + q_1\hat{\imath} + q_2\hat{\jmath} + q_3\hat{\kappa}$. The main feature of quaternions is that multiplication of two quaternions is noncommutative $\hat{\imath}\hat{\jmath} \neq \hat{\jmath}\hat{\imath}$. So, the quaternions form non-commutative algebra that allows the definition of an associative skew-field, i.e., satisfying all the usual properties of fields over real- or complex-valued numbers, except the commutative properties of the product.

Due to this particular property, the convolution/correlation over time is not equivalent to the frequency domain product. Thus, the extension of the FDAF methods to the quaternion domain is not a trivial operation and can be computed using the following equation (see [25] for further details)

$$\begin{aligned} \boldsymbol{Y}_k &= \boldsymbol{W}_k^a \boldsymbol{X}_k + \boldsymbol{W}_k^b \boldsymbol{\nu}_2 \boldsymbol{X}_{-k} \\ \boldsymbol{C}_k &= \boldsymbol{E}_k^a \boldsymbol{X}_k^H + \boldsymbol{E}_k^b \boldsymbol{\nu}_2 \boldsymbol{X}_{-k}^H \end{aligned} \tag{3}$$

where $\boldsymbol{\nu}_2$ is a versor that describes the spatial direction of the imaginary part of a quaternion. In fact, differently from the real/complex-valued OS-FDAF, in the quaternion domain some modifications are required due to the fact that the convolution theorem is not valid in the standard formulation as the product of the DFT sequences, but it is slightly more complicated. However, exploiting the framework and the derivation of the OS-FDAF introduced in the previous section, it is easy to derive the scheme of the *overlap-save quaternion FDAF* (OS-QFDAF), reported in Fig. 2.

5. EXAMPLE OF APPLICATION TO 3D AUDIO

In this Section, we propose a quaternion adaptive 3D audio processing application, considering the representation of the acoustic field with a coincident microphones array (CMA). In particular, we consider the *Ambisonics* microphone recording technique, which is based on local-space sampling of the acoustic field with an CMA. Each microphone has a polar diagram equal to the Fourier spherical harmonics so that the acoustic pressure field, due to external sources, can be decomposed into a set of orthogonal functions. In particular, the *B-format* Ambisonic is composed by four microphones, one omnidirectional, indicated with W, and three figure-of-eight capsules, indicated with X, Y, Z. Let $w[n]$, be the signal corresponds to the omnidirectional microphone, whereas $x[n]$, $y[n]$, $z[n]$, are the components that would be picked up by figure-of-eight capsules oriented along the three spatial axes; these signals are representative of the same acoustic sources and therefore strongly correlated with each other. The idea is to consider the B-format 3D audio captured as a quaternion signal [33,34], that is $s[n] = w[n] + x[n]\hat{\imath} + y[n]\hat{\jmath} + z[n]\hat{\kappa}$.

As an application example, we consider the 3D components W, X, Y, Z of a room impulse response (RIR) captured with a B-format soundfield microphone, depicted in Fig. 3(a). In order to derive the estimated inverse impulse response (e.g., for room equalization purposes) we process the quaternion signals by the OS-FDAF described in Section 4. Inversion results are depicted in Fig. 3(b). Finally, in order to demonstrate the consistence of the proposed algorithm, in Fig. 3(c) we show the result of the quaternion convolution between the RIR and its inverse that, from a simple visual inspection, it approximates perfectly the unitary real impulse.

6. CONCLUSIONS

Adaptive filtering represents a central theme in several university courses oriented to signal analysis. Frequency-domain algorithms play an important role in adaptive filtering due to their efficient capabilities, but they may result difficult to introduce due to heavy notation and variety of algorithm versions. In this paper, we have introduce FDAFs in a simplified, extended and unitary way, from which the most popular algorithms belonging to the family of such filters can be easily derived. The proposed framework also enables the derivation of complex and hypercomplex FDAFs. In particular, we show how to derive a quaternion OS-FDAF, whose effectiveness has been proved in a 3D audio signal processing problem.

7. REFERENCES

[1] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, Prentice Hall, Inc., Upper Saddle River, NJ, USA, 1985.

[2] A. H. Sayed, *Adaptive Filters*, John Wiley & Sons, 2008.

[3] B. Farhang-Boroujeny, *Adaptive Filters: Theory and Applications*, John Wiley & Sons, Chichester, UK, 2nd edition, 2013.

[4] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*, Springer, New York, NY, 2013.

[5] S. O. Haykin, *Adaptive Filter Theory*, Pearson, 2014.

[6] A. Uncini, *Fundamentals of Adaptive Signal Processing*, Signal and Communication Technology. Springer International Publishing AG, Cham, Switzerland, 2015.

[7] M. G. Morrow, C. H. G. Wright, and T. B. Welch, "Real-time DSP for adaptive filters: A teaching opportunity," in *IEEE Int. Conf. on Acoust., Speech, and Signal Process. (ICASSP)*, Vancouver, Canada, May 2013, pp. 4335–4338.

[8] E. R. Ferrara, "Fast implementations of LMS adaptive filters," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 28, no. 4, pp. 474–475, Aug. 1980.

[9] G. A. Clark, S. R. Parker, and S. K. Mitra, "A unified approach to time- and frequency-domain realization of FIR adaptive digital filters," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 31, no. 5, pp. 1073–1083, Oct. 1983.

[10] M. Dentino, J. McCool, and B. Widrow, "Adaptive filtering in the frequency domain," *Proc. IEEE*, vol. 66, no. 12, pp. 1658–1660, Dec. 1978.

[11] N. J. Bershad and P. L. Feintuch, "Analysis of the frequency domain adaptive filter," *Proc. IEEE*, vol. 67, no. 12, pp. 1658–1659, Dec. 1979.

[12] 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.

[13] J. J. Shynk, "Frequency-domain and multirate adaptive filtering," *IEEE Signal Process. Mag.*, vol. 9, pp. 14–37, 1992.

[14] S. S. Narayan and A. M. Peterson, "Frequency-domain least-mean-square algorithm," *Proc. IEEE*, vol. 69, no. 1, pp. 124–126, Jan. 1981.

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

[16] W. B. Mikhael, A. S. Spanias, and F. H. Wu, "Fast frequency domain implementation of a block IIR filter with application," in *IEEE Int. Symp. on Circuits and Syst. (ISCAS)*, Espoo, Finland, June 1988, pp. 285–288.

[17] P. C. W. Sommen, "Partitioned frequency domain adaptive filters," in *Twenty-Third Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, CA, Nov. 1989, pp. 677–681.

[18] B. Farhang-Boroujeny and S Gazor, "Generalized sliding FFT and its application to implementation of block LMS adaptive filters," *IEEE Trans. Signal Process.*, vol. 42, no. 3, pp. 532–538, Mar. 1994.

[19] M. Schneider and W. Kellermann, "The generalized frequency-domain adaptive filtering algorithm as an approximation of the block recursive least-squares algorithms," *EURASIP J. on Advances in Signal Process.*, vol. 6, pp. 1–15, Jan. 2016.

[20] F. Yang, G. Enzner, and J. Yang, "Frequency-domain adaptive Kalman filter with fast recovery of abrupt echo-path changes," *IEEE Signal Process. Lett.*, vol. 24, pp. 1778–1782, 2017.

[21] D. Shi, B. Lam, and W.-S. Gan, "A novel selective active noise control algorithm to overcome practical implementation issue," in *IEEE Int. Conf. on Acoust., Speech, and Signal Process. (ICASSP)*, Calgary, Canada, Apr. 2018, pp. 1130–1134.

[22] H. Schepker, L. T. T. Tran, S. Nordholm, and S. Doclo, "Improving adaptive feedback cancellation in hearing aids using an affine combination of filters," in *IEEE Int. Conf. on Acoust., Speech, and Signal Process. (ICASSP)*, 2016, pp. 231–235.

[23] J. Taghia and R. Martin, "A frequency-domain adaptive line enhancer with step-size control based on mutual information for harmonic noise reduction," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 24, no. 6, pp. 1140–1154, June 2016.

[24] T. Dietzen, A. Spriet, W. Tirry, S. Doclo, M. Moonen, and T. van Waterschoot, "Partitioned block frequency domain Kalman filter for multi-channel linear prediction based blind speech dereverberation," in *IEEE Int. Workshop on Acoust. Signal Enhancement (IWAENC)*, Xi'an, China, 2016, pp. 1–5.

[25] F. Ortolani, D. Comminiello, M. Scarpiniti, and A. Uncini, "Frequency domain quaternion adaptive filters: Algorithms and convergence performance," *Signal Process.*, vol. 136, pp. 69–80, July 2017.

[26] H. He, J. Chen, J. Benesty, and T. Yang, "Noise robust frequency-domain adaptive blind multichannel identification with ℓ_p-norm constraint," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 26, no. 9, pp. 1608–1619, Sept. 2018.

[27] J. Franzen and T. Fingscheidt, "An efficient residual echo suppression for multi-channel acoustic echo cancellation based on the frequency-domain adaptive Kalman filter," in *IEEE Int. Conf. on Acoust., Speech, and Signal Process. (ICASSP)*, Calgary, Canada, Apr. 2018, pp. 226–230.

[28] M. Asharif, T. Takebayashi, T. Chujo, and K. Murano, "Frequency domain noise canceller: Frequency bin adaptive filtering (FBAF)," in *IEEE Int. Conf. on Acoust., Speech, and Signal Process. (ICASSP)*, Tokyo, Japan, 1986, pp. 2219–2222.

[29] S. S. Narayan, A. M. Peterson, and M. J. Narashima, "Transform domain LMS algorithm," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 31, no. 3, pp. 609–615, June 1983.

[30] F. Beaufays, "Transform-domain adaptive filters: An analytical approach," *IEEE Trans. Signal Process.*, vol. 43, no. 2, pp. 422–431, Feb. 1995.

[31] C. C. Took and D. P. Mandic, "The quaternion LMS algorithm for adaptive filtering of hypercomplex processes," *IEEE Trans. Signal Process.*, vol. 57, no. 4, pp. 1316–1327, Apr. 2009.

[32] M. Xiang, S. Kanna, and D. P. Mandic, "Performance analysis of quaternion-valued adaptive filters in nonstationary environments," *IEEE Trans. Signal Process.*, vol. 66, no. 6, pp. 1566–1579, Mar. 2018.

[33] F. Ortolani, D. Comminiello, M. Scarpiniti, and A. Uncini, "Advances in hypercomplex adaptive filtering for 3D audio processing," in *IEEE First Ukraine Conf. on Elect. and Comput. Eng. (UKRCON)*, Kiev, Ukraine, 2017, pp. 1125–1130.

[34] F. Ortolani, D. Comminiello, and A. Uncini, "The widely linear block quaternion least mean square algorithm for fast computation in 3D audio systems," in *26th IEEE Workshop on Machine Learning for Signal Process. (MLSP)*, Vietri sul Mare, Italy, Sept. 2016, pp. 1–6.