applied sciences

MDPI

Article

Frequency-Domain Filtered-x LMS Algorithms for Active Noise Control: A Review and New Insights

Feiran Yang [1], Yin Cao [1], Ming Wu [1], Felix Albu [2] and Jun Yang [1,3,*]

[1] State Key Laboratory of Acoustics and Key Laboratory of Noise and Vibration Research, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China; feiran@mail.ioa.ac.cn (F.Y.); caoyfive@gmail.com (Y.C.); mingwu@mail.ioa.ac.cn (M.W.)

[2] Department of Electronics, Valahia University of Targoviste, Targoviste 130082, Romania; felix.albu@valahia.ro

[3] School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100049, China

* Correspondence: jyang@mail.ioa.ac.cn; Tel.: +86-10-8254-7549

Received: 23 October 2018; Accepted: 16 November 2018; Published: 20 November 2018

Abstract: This paper presents a comprehensive overview of the frequency-domain filtered-x least mean-square (FxLMS) algorithms for active noise control (ANC). The direct use of frequency-domain adaptive filters for ANC results in two kinds of delays, i.e., delay in the signal path and delay in the weight adaptation. The effects of the two kinds of delays on the convergence behavior and stability of the adaptive algorithms are analyzed in this paper. The first delay can violate the so-called causality constraint, which is a major concern for broadband ANC, and the second delay can reduce the upper bound of the step size. The modified filter-x scheme has been employed to remove the delay in the weight adaptation, and several delayless filtering approaches have been presented to remove the delay in the signal path. However, state-of-the-art frequency-domain FxLMS algorithms only remove one kind of delay, and some of these algorithms have a very high peak complexity and hence are impractical for real-time systems. This paper thus proposes a new delayless frequency-domain ANC algorithm that completely removes the two kinds of delays and has a low complexity. The performance advantages and limitations of each algorithm are discussed based on an extensive evaluation, and the complexities are evaluated in terms of both the peak and average complexities.

Keywords: Active noise control; frequency domain adaptive filter; computational complexity; delayless; causality

1. Introduction

Acoustic noise control is essential for reducing the noise level in modern society since noise seriously affects human health [1,2]. Passive noise control, which is based on using reactive devices, e.g., Helmholtz resonators and quarter wavelength resonators, and using resistive materials, e.g., acoustic linings and porous membranes, is very effective for reducing high-frequency noise, but not so effective for low-frequency noise reduction. Active noise control (ANC) based on the principle of superposition is an appealing method for low-frequency noise reduction [3–6]. In practice, passive noise control and ANC methods are usually combined to provide wideband noise reduction.

In ANC systems, the filtered-x least mean-square (FxLMS) algorithm is widely used to update the weights of the control filter, but the complexity of the FxLMS algorithm increases linearly with the filter length. In certain applications, the control filter is on the order of several thousand. Therefore, the FxLMS and other time-domain algorithms, e.g., the filtered-x affine projection (FxAP) [7–9], are too complex, which is prohibitive for real-time systems. The frequency-domain adaptive filter (FDAF) [10–17] has been successfully used in echo cancellation [18–21], acoustic feedback cancellation [22], and beamforming [23] due to its good convergence behavior and low complexity. In [24–27], the block LMS (BLMS) was extended to ANC applications and the corresponding frequency-domain implementation was provided. The experiment in [27] used both the periodic and band-limited signals as the reference signals, but the delay problem was not discussed. In [28], a more efficient implementation of BLMS was presented, where the filtering of the reference signal by the secondary path was also implemented using FFT. Meanwhile, the fast Hartley transform (FHT) was adopted to reduce the complexity. The frequency-domain implementation was extended to the multichannel case in [26,29]. The aforementioned algorithms are obtained by applying the FDAF originally used in echo cancellation to the ANC problem. However, the algorithm requirements for echo cancellation and ANC are quite different. Specifically, the direct use of the FDAF for ANC introduces two kinds of delays. First, there is at least one-block delay between the input of the reference signal and the output of the cancelling signal because the FDAF algorithm is implemented on a block-by-block basis. This delay can violate the causality constraint, which is a major concern for broadband ANC [4,30,31]. Second, the delay between the weight adaptation and the observation of the error signal is introduced because of the effect of the secondary path. The behavior of the filtered-x algorithm is similar to that of the delayed-LMS algorithm, which reduces the upper bound of the step size and slows the convergence and reconvergence rate [32]. In addition, the peak complexity of the FDAF algorithms presented in [28] is quite high because the FFT operation should be completed in one sampling interval; therefore, these algorithms are impractical for real-time implementation.

Many approaches have been proposed to reduce the two kinds of delays. The modified filtered-x structure [33,34] was used in the frequency-domain ANC in [35], which makes the adaptive algorithm behave like the standard FDAF and thus removes the delay in the weight adaptation. The partitioned-block FDAF (PBFDAF) algorithm was introduced for ANC to reduce the delay in the signal path [36–39]. This is achieved by partitioning the whole impulse response into several small sections, but this method does not completely eliminate the input-output delay. Several delayless algorithms were then proposed to totally remove the delay in the signal path. The calculation of the adaptive filter output can be implemented directly using the time-domain convolution and hence the delay in the signal path is removed [40–43]. However, the complexity of the convolution in the time domain is still high. In [44], a delayless FDAF algorithm originally used in echo cancellation [45] was extended to ANC, but the computational burden at certain sampling periods is rather high. Two computationally efficient FDAF algorithms for ANC were proposed in [46,47] using the delayless approach in [48,49], but the delay in the filter adaptation is not removed.

Although many efforts have been devoted to the frequency-domain FxLMS algorithms, these algorithms are not well compared and analyzed. To fill this gap, this paper presents a comprehensive review of the frequency-domain FxLMS algorithms. Only mono-channel algorithms are considered in this paper, since they can be straightforwardly extended to the multi-channel case. The conventional frequency-domain FxLMS algorithms are reviewed, and then the effects of the two kinds of delays on the overall performance are analyzed. Specifically, it was found that calculating the adaptive filter output and the weight vector update in different ways leads to different convergence performances, which is quite different from system identification problems. The update-first approach is then proposed to improve the stability. Several delayless frequency-domain algorithms for ANC are surveyed, but these algorithms did not remove the delay related to the secondary path. To address this problem, we present a new delayless frequency-domain FxLMS algorithm to completely remove the aforementioned two types of delays and overcome the shortcomings of the state-of-the-art frequency-domain FxLMS algorithms. The complexity of the frequency-domain approaches in terms of both peak and average multiplications per sample are evaluated. Simulations are carried out to evaluate the convergence performance and stability of the frequency-domain FxLMS algorithms.

2. FxLMS Algorithm

The diagram of the FxLMS algorithm is presented in Figure 1, where $P(z)$ is the primary path and $S(z)$ is the secondary path. We define the weight vector of the control filter at the time index n as $\mathbf{w}(n) = [w_0(n), w_1(n), \cdots, w_{N_\mathrm{w}-1}(n)]^T$ with a length of N_w. The reference signal $x(n)$ is picked up by a reference microphone and then filtered by the adaptive filter to generate the cancelling signal $y(n)$ driven by a secondary loudspeaker

$$y(n) = \mathbf{x}^T(n)\mathbf{w}(n) \tag{1}$$

where $\mathbf{x}(n) = [x(n), x(n-1), \cdots, x(n-N_\mathrm{w}+1)]^T$ is the reference signal vector. The cancelling signal $y(n)$ is then filtered by the secondary path $S(z)$ to obtain the control signal $z(n)$ at the location of the error microphone

$$z(n) = \bar{\mathbf{y}}^T(n)\mathbf{s} \tag{2}$$

where $\bar{\mathbf{y}}(n) = [y(n), y(n-1), \cdots, y(n-N_\mathrm{s}+1)]^T$, and $\mathbf{s} = [s_0, s_1, \cdots, s_{N_\mathrm{s}-1}]^T$ is the weight vector of the secondary path with a length of N_s. The residual error signal picked up by the error microphone is

$$e(n) = d(n) + z(n), \tag{3}$$

where $d(n)$ is the disturbance signal at the error microphone.

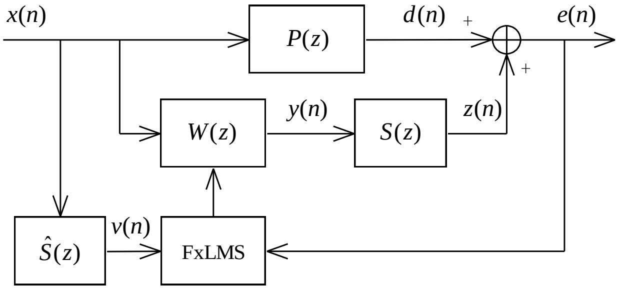

Figure 1. FxLMS Algorithm.

The FxLMS algorithm is commonly used to update the weight vector [3,4]

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu\mathbf{v}(n)e(n) \tag{4}$$

where μ is the step size, and $\mathbf{v}(n) = [v(n), v(n-1), \cdots, v(n-N_\mathrm{w}+1)]^T$ the filtered reference signal vector, where the filtered reference signal $v(n)$ is given by

$$v(n) = \mathbf{x}_\mathrm{s}^T(n)\hat{\mathbf{s}} \tag{5}$$

where $\mathbf{x}_\mathrm{s}(n) = [x(n), x(n-1), \cdots, x(n-N_\mathrm{s}+1)]^T$, and $\hat{\mathbf{s}} = [\hat{s}_0, \hat{s}_1, \cdots, \hat{s}_{N_\mathrm{s}-1}]^T$ is an estimate of the actual weight vector of the secondary path. An accurate estimate of the secondary path is required for the FxLMS to work properly, which can be obtained by an offline- or online-modeling method [4,6].

The FxLMS algorithm requires $2N_\mathrm{w} + N_\mathrm{s}$ multiplications. In certain scenarios, the length of the weight vector may reach several thousand taps. Thus, the computational complexity is a major concern for real-time systems.

3. Conventional Frequency-Domain ANC Algorithms

The conventional frequency-domain ANC algorithms in [24–28,37,39] can be understood as direct frequency-domain implementations of the FxLMS algorithm. Since the FDAF algorithm is a special case of PBFDAF, we only present the PBFDAF in this paper. The corresponding method using the conventional filtered-x scheme is referred to as the frequency-domain partitioned block filtered-x LMS (FPBFxLMS) algorithm. The method using the modified filtered-x scheme is referred to as the frequency-domain partitioned block modified filtered-x LMS (FPBMFxLMS) algorithm.

3.1. FPBFxLMS Algorithm

The diagram of the FPBFxLMS algorithm is shown in Figure 2. The linear convolution operations (1) and (5) and the cross-correlation (4) can be implemented using FFTs. The adaptive filter $\mathbf{w}(n)$ is segmented into P_w partitions as $\mathbf{w}(n) = [\mathbf{w}_0^T(n), ..., \mathbf{w}_{P_\mathrm{w}-1}^T(n)]^T$, where $\mathbf{w}_p(n) = [w_{pL}(n), ..., w_{(p+1)L-1}(n)]^T$ is the p-th subfilter with $L = N_\mathrm{w}/P_\mathrm{w}$ taps. The frequency-domain weight vector of the p-th partition is

$$\mathbf{W}_p(k) = \mathbf{F}[\mathbf{w}_p^T(k), \mathbf{0}_{1\times L}]^T \tag{6}$$

where k denotes the frame index, $\mathbf{0}_{1\times L}$ is a $1 \times L$ all-zero vector, and $\mathbf{F}$ is the Fourier transform matrix whose (p, q)-th element is $\exp(-\mathrm{j}\frac{2\pi pq}{2L})$ with $0 \le p \le 2L-1, 0 \le q \le 2L-1$ and $\mathrm{j} = \sqrt{-1}$. The output of the control filter at the k-th frame is

$$\begin{aligned} \mathbf{y}(k) &= [y(kL-L+1), ..., y(kL)]^T \\ &= \mathbf{Q}^{01}\mathbf{F}^{-1}\sum_{p=0}^{P_\mathrm{w}-1}\mathbf{X}_p(k)\mathbf{W}_p(k), \end{aligned} \tag{7}$$

where $\mathbf{Q}^{01} = [\mathbf{0}_L \quad \mathbf{I}_L]$ is the projection matrix (the notations $\mathbf{0}_L$ and $\mathbf{I}_L$ being the $L \times L$ zeros and identity matrices, respectively), and

$$\mathbf{X}_p(k) = diag\{\mathbf{F}\mathbf{x}_p(k)\} \tag{8}$$

is the frequency-domain reference matrix with $\mathbf{x}_p(k) = \{x[(k-p-2)L+1], ..., x[(k-p)L]\}^T$ ($diag\{\cdot\}$ denoting the diagonal matrix).

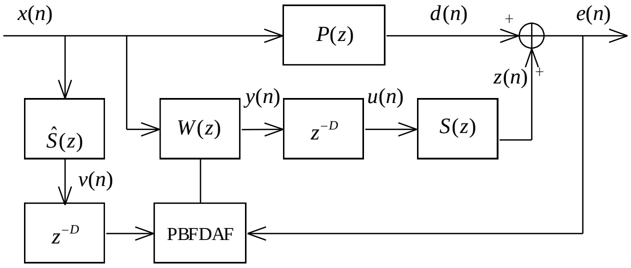

Figure 2. FPBFxLMS algorithm.

For the FxLMS algorithm, the output of the control filter is calculated on a sample-by-sample basis and hence there is only one-sample delay for the generation of the cancelling signal. For the frequency-domain implementation in (7), however, a block of the reference signals should be collected before the calculation of $\mathbf{y}(k)$. There is at least one-block delay even if (7) is completed in one sample period. Thus, the cancelling signal driven by the loudspeaker is a delayed version of $y(n)$:

$$u(n) = y(n-D) \tag{9}$$

where D denotes the delay, which includes both the buffering time and the processing time. In [28,37], the calculation of (7) is completed in one-sample period and hence the total delay $D = L$. In [39], the calculation of (7) is distributed in one block $KL \leq n \leq KL + L$, and thus the delay is $D = 2L$. The advantages and limitations of each implementation method are discussed later.

The impulse response of the secondary path is segmented into P_s partitions as $\hat{\mathbf{s}} = [\hat{\mathbf{s}}_0^T, \hat{\mathbf{s}}_1^T, ..., \hat{\mathbf{s}}_{P_\mathrm{s}-1}^T]^T$, where $\hat{\mathbf{s}}_p = [\hat{s}_{pL}, ..., \hat{s}_{(p+1)L-1}]^T$ is the p-th subfilter with $L = N_\mathrm{s}/P_\mathrm{s}$ taps. The Fourier transform of $\hat{\mathbf{s}}_p$ is

$$\hat{\mathbf{S}}_p = \mathbf{F}[\hat{\mathbf{s}}_p^T, \mathbf{0}_{1\times L}]^T. \tag{10}$$

The linear convolution in (5) can also be implemented using FFT

$$\begin{aligned} \mathbf{v}(k) &= [v(kL - L + 1), ..., v(kL)]^T \\ &= \mathbf{Q}^{01}\mathbf{F}^{-1}\sum_{p=0}^{P_\mathrm{s}-1} \mathbf{X}_p(k)\hat{\mathbf{S}}_p. \end{aligned} \tag{11}$$

The update equation of the FPBFxLMS algorithm is [10,16,37,39]

$$\mathbf{W}_p(k+1) = \mathbf{W}_p(k) - \mu\mathbf{G}^{10}\mathbf{\Lambda}^{-1}(k)\mathbf{V}_p^H(k-m)\mathbf{E}(k), \tag{12}$$

where $(\cdot)^H$ denotes the Hermitian operation,

$$\mathbf{G}^{10} = \mathbf{F}\begin{bmatrix} \mathbf{I}_L & \mathbf{0}_L \\ \mathbf{0}_L & \mathbf{0}_L \end{bmatrix}\mathbf{F}^{-1} \tag{13}$$

is the windowing matrix that forces the last L time-domain elements to zero,

$$\mathbf{E}(k) = \mathbf{F}[\mathbf{0}_{1\times L}, \mathbf{e}^T(k)]^T \tag{14}$$

is the frequency-domain error vector with $\mathbf{e}(k) = [e(kL - L + 1), ..., e(kL)]^T$,

$$\mathbf{V}_p(k) = diag\{\mathbf{F}\mathbf{v}_p(k)\} \tag{15}$$

is the filtered reference signal matrix with $\mathbf{v}_p(k) = \{v[(k-p-2)L+1], ..., v[(k-p)L]\}^T$, $\mathbf{\Lambda}(k)$ is the PSD matrix of the filtered reference signal which is used to improve the convergence, and the parameter m is related to the algorithm latency

$$m = \begin{cases} 1 & D = L \\ 2 & D = 2L \end{cases}. \tag{16}$$

The PSD matrix $\mathbf{\Lambda}(k)$ can be computed recursively [10,16]

$$\mathbf{\Lambda}(k) = \lambda\mathbf{\Lambda}(k-1) + (1-\lambda)\mathbf{V}_0^H(k)\mathbf{V}_0(k) \tag{17}$$

where λ is a smoothing factor, $0 < \lambda < 1$.

To reduce the complexity while retaining good convergence properties, the constraint in (12) can be added periodically [50,51]. Another way to reduce the complexity of (12) is to use the unconstrained algorithm [13]

$$\mathbf{W}_p(k+1) = \mathbf{W}_p(k) - \mu\mathbf{\Lambda}^{-1}(k)\mathbf{V}_p^H(k-m)\mathbf{E}(k). \tag{18}$$

The unconstrained PBFDAF exhibits a lower complexity than the constrained version, but the constrained version has a better convergence. Thus, we only consider the constrained FDAF.

The frequency-domain FxLMS algorithms with $D = L$ and $D = 2L$ are presented in Tables 1 and 2, respectively, where A denotes the number of multiplications of the $2L$-point FFT operation. The differences between the two algorithms are twofold. First, the FPBFxLMS II algorithm has a larger delay, which limits its application in broadband ANC. Second, the complexity of the FPBFxLMS II algorithm is distributed evenly in one frame, while steps (1)–(7) of the FPBFxLMS I algorithm should be completed in one-sample period. Therefore, the peak complexity of the FPBFxLMS I algorithm is rather high. This means that a strong processor should be used, and a large portion of power is wasted due to the unbalanced complexity [46,48,49]. From the perspective of real-time implementation, the FPBFxLMS I algorithm is not practical. A real-time multichannel ANC system using the FPBFxLMS II algorithm was implemented in the Graphics Processing Unit (GPU) in [39], and a comprehensive complexity evaluation was carried out.

Table 1. FPBFxLMS I [28,37].

Operation	Complexity
(1) Compute the frequency-domain reference matrix $\mathbf{X}_0(k)$ using (8)	One FFT
(2) Compute the output of the control filter using (7)	One FFT and $4LP_\mathrm{w}$ multiplications
(3) Compute the filtered reference signal vector $\mathbf{v}(k)$ using (11)	One FFT and $4LP_\mathrm{s}$ multiplications
(4) Compute the FFT of $\mathbf{v}_0(k)$ using (15)	One FFT
(5) Update $\mathbf{\Lambda}(k)$ using (17)	$4L$ multiplications
(6) Compute the FFT of the error signal using (14)	One FFT
(7) Update the weight vector using (12)	$2P_\mathrm{w}$ FFTs and $6LP_\mathrm{w}$ multiplications
(8) Compute the cancelling signal using (9): $u(n) = y(n-L)$	0

Note: Steps (1)–(7) should be completed between kL and $kL+1$; Average multiplications: $(5+2P_\mathrm{w})A/L + 10P_\mathrm{w} + 4P_\mathrm{s}$; Peak multiplications: $(5+2P_\mathrm{w})A + 10LP_\mathrm{w} + 4LP_\mathrm{s} + 4L$.

Table 2. FPBFxLMS II [39].

Operation	Complexity
(1) Compute the frequency-domain reference matrix $\mathbf{X}_0(k)$ using (8)	One FFT
(2) Compute the output of the control filter using (7)	One FFT and $4LP_\mathrm{w}$ multiplications
(3) Compute the filtered reference signal vector $\mathbf{v}(k)$ using (11)	One FFT and $4LP_\mathrm{s}$ multiplications
(4) Compute the FFT of $\mathbf{v}_0(k)$ using (15)	One FFT
(5) Update $\mathbf{\Lambda}(k)$ using (17)	$4L$ multiplications
(6) Compute the FFT of the error signal using (14)	One FFT
(7) Update the weight vector using (12)	$2P_\mathrm{w}$ FFTs and $6LP_\mathrm{w}$ multiplications
(8) Compute the cancelling signal using (9): $u(n) = y(n-2L)$	0

Note: Steps (1)–(7) should be completed between kL and $kL+L$; Average and peak multiplications: $(5+2P_\mathrm{w})A/L + 10P_\mathrm{w} + 4P_\mathrm{s}$.

3.2. Analysis

In this section, we carry out theoretical analysis on the effects of the two kinds of delays on the adaptive algorithm convergence behaviors. When the FxLMS converges, the optimal solution (which may not be realizable) in the z-domain is [4]

$$W_\mathrm{opt}(z) = \frac{P(z)}{S(z)}. \tag{19}$$

For the FPBFxLMS algorithms, a D-sample delay is introduced to the signal path, and hence the optimal solution becomes

$$W_\mathrm{opt}(z) = \frac{P(z)}{z^{-D}S(z)}. \tag{20}$$

When the FDAF algorithm is applied for echo cancellation, the algorithm delay has no effect on the algorithm convergence. However, this is not the case for broadband ANC because the causality solution of (20) may not exist and the extra delay in the signal path deteriorates the noise reduction performance [30,31]. For the broadband ANC system to work properly, the casualty condition should be satisfied [3]

$$\tau_\mathrm{buff} + \tau_\mathrm{proc} + \tau_\mathrm{sec} \le \tau_\mathrm{AD}, \tag{21}$$

where τ_{sec} includes the delay in the antialiasing filter, A/D converter, D/A converter, reconstruction filter and loudspeaker and the acoustic delay between the secondary loudspeaker and the error microphone, τ_{buff} denotes the buffering time, τ_{proc} is the processing time of the adaptive algorithm, τ_{AD} is the acoustic delay between the reference sensor and the error microphone. The algorithm latency is the sum of τ_{buff} and τ_{proc}, and we have $\tau_{\text{buff}} + \tau_{\text{proc}} = DT_s$, where T_s is the system sampling period. For the FxLMS algorithm, only one-sample delay is introduced, i.e., $D = 1$. For the aforementioned two FPBFxLMS algorithms, however, the algorithm latency is very large, i.e., $D = L$ or $D = 2L$. In the frequency-domain ANC algorithm, the distance between the reference sensor and the secondary loudspeaker should be large enough to satisfy the casualty condition [30,31]. However, this is not always achievable because of practical installation constraints. Therefore, the delay in FPBFxLMS algorithms is a major limitation for broadband ANC.

At this point, we investigate the effect of the secondary path on the weight update. Using (3), we rewrite the error vector $\mathbf{e}(k)$ as

$$\begin{aligned}\begin{bmatrix} e(kL-L+1) \\ \cdots \\ e(kL) \end{bmatrix} &= \begin{bmatrix} d(kL-L+1) \\ \cdots \\ d(kL) \end{bmatrix} + \begin{bmatrix} z(kL-L+1) \\ \cdots \\ z(kL) \end{bmatrix} \\ &= \begin{bmatrix} d(kL-L+1) \\ \cdots \\ d(kL) \end{bmatrix} + \begin{bmatrix} u(kL-L+1) * s \\ \cdots \\ u(kL) * s \end{bmatrix} \\ &= \begin{bmatrix} d(kL-L+1) \\ \cdots \\ d(kL) \end{bmatrix} + \begin{bmatrix} y(kL-L+1-D) * s \\ \cdots \\ y(kL-D) * s \end{bmatrix}\end{aligned} \tag{22}$$

where $*$ denotes the linear convolution. The last element of $\mathbf{e}(k)$ can be expressed as

$$\begin{aligned} e(kL) &= d(kL) + y(kL-D) * s \\ &= d(kL) + \sum_{p=0}^{P_{\text{s}}-1} \mathbf{y}^T(k-m-p)\tilde{\mathbf{s}}_p \\ &= d(kL) + \sum_{p=0}^{P_{\text{s}}-1} \left[\mathbf{Q}^{01}\mathbf{F}^{-1} \sum_{q=0}^{P_{\text{w}}-1} \mathbf{X}_q(k-m-p)\mathbf{W}_q(k-m-p) \right] \tilde{\mathbf{s}}_p, \end{aligned} \tag{23}$$

where $\tilde{\mathbf{s}}_p = [s_{(p+1)L-1}, ..., s_{pL}]^T$. From (23), it can be seen that the generation of $e(kL)$ requires P_{s} past weight vector $\mathbf{W}_q(k-m-p), p = 0, ..., P_{\text{s}} - 1$. Similarly, the generation of $\mathbf{e}(k)$ requires $\mathbf{W}_q(k-m-p), p = 0, ..., P_{\text{s}}$. Therefore, the following assumption is used implicitly for deriving (12)

$$\mathbf{W}_q(k) \approx \mathbf{W}_q(k-1) \approx \cdots \approx \mathbf{W}_q(k-m-P_{\text{s}}). \tag{24}$$

The relationship in (24) holds only when the step size μ is very small, and the frequency-domain algorithm used in ANC is very similar to the delayed LMS algorithm [32]. Hence, the maximum step size that can guarantee the stability of the algorithm is reduced, and the fastest convergence speed of the algorithm is lower. Therefore, the frequency-domain ANC algorithm becomes quite different from the standard FDAF algorithm in terms of convergence behavior.

The above analysis provides another interesting insight to the implementation of the frequency-domain algorithm. For the two algorithms in Tables 1 and 2, we first calculate the filtering-out using $\mathbf{W}_p(k)$ and then update $\mathbf{W}_p(k)$, which is called the filtering-first approach. We could propose an alternative method, namely, the update-first approach. That is, we first update the weight vector to obtain $\mathbf{W}_p(k+1)$ using (18), and then we use the new weight vector $\mathbf{W}_p(k+1)$ to calculate the filtering-out

$$\mathbf{y}(k) = \mathbf{Q}^{01}\mathbf{F}^{-1} \sum_{p=0}^{P_{\text{w}}-1} \mathbf{X}_p(k)\mathbf{W}_p(k+1). \tag{25}$$

By so doing, the delay between the weight update and the observation of the error signal can be reduced, and hence the convergence performance can be improved. We then obtain two algorithms in Tables 3 and 4 by applying the update-first approach to the algorithms in Tables 1 and 2, respectively. The update-first approach exhibits a better convergence behavior than the filtering-first approach, while these approaches have a similar computational complexity.

Table 3. FPBFxLMS III.

Operation	**Complexity**
(1) Compute the frequency-domain reference matrix $\mathbf{X}_0(k)$ using (8)	One FFT
(2) Compute the filtered reference signal vector $\mathbf{v}(k)$ using (11)	One FFT and $4LP_\mathrm{s}$ multiplications
(3) Compute the FFT of $\mathbf{v}_0(k)$ using (15)	One FFT
(4) Update $\mathbf{\Lambda}(k)$ using (17)	$4L$ multiplications
(5) Compute the FFT of the error signal using (14)	One FFT
(6) Update the weight vector using (12)	$2P_\mathrm{w}$ FFTs and $6LP_\mathrm{w}$ multiplications
(7) Compute the output of the control filter using (25)	One FFT and $4LP_\mathrm{w}$ multiplications
(8) Compute the cancelling signal using (9): $u(n) = y(n-L)$	0

Note: Steps (1)–(7) should be completed between kL and $kL+1$; Average multiplications: $(5+2P_\mathrm{w})A/L + 10P_\mathrm{w} + 4P_\mathrm{s}$; Peak multiplications: $(5+2P_\mathrm{w})A + 10LP_\mathrm{w} + 4LP_\mathrm{s} + 4L$.

Table 4. FPBFxLMS IV.

Operation	**Complexity**
(1) Compute the frequency-domain reference matrix $\mathbf{X}_0(k)$ using (8)	One FFT
(2) Compute the filtered reference signal vector $\mathbf{v}(k)$ using (11)	One FFT and $4LP_\mathrm{s}$ multiplications
(3) Compute the FFT of $\mathbf{v}_0(k)$ using (15)	One FFT
(4) Update $\mathbf{\Lambda}(k)$ using (17)	$4L$ multiplications
(5) Compute the FFT of the error signal using (14)	One FFT
(6) Update the weight vector using (12)	$2P_\mathrm{w}$ FFTs and $6LP_\mathrm{w}$ multiplications
(7) Compute the output of the control filter using (25)	One FFT and $4LP_\mathrm{w}$ multiplications
(8) Compute the cancelling signal using (9): $u(n) = y(n-2L)$	0

Note: Steps (1)–(7) should be completed between kL and $kL+L$; Average and peak multiplications: $(5+2P_\mathrm{w})A/L + 10P_\mathrm{w} + 4P_\mathrm{s}$.

3.3. FPBMFxLMS Algorithm

The aforementioned frequency-domain algorithms directly employ the residual error signal $e(n)$ to update the weight vector, resulting in a delay between the weight adaptation and the observation of the error signal. To reduce this type of delay, the modified filtered-x structure was adopted to update the weight vector in [35], as shown in Figure 3. The basic idea is summarized as follows. The disturbance signal $\hat{d}(n)$ at the error microphone is estimated as

$$\hat{d}(n) = e(n) - \hat{z}(n) \tag{26}$$

where

$$\hat{z}(n) = u(n) * \hat{s} \tag{27}$$

is the estimate of the control signal at the location of the error microphone. To reduce the complexity, the linear convolution in (27) is implemented using FFT. Then, $\hat{z}(n), kL - L + 1 \le n \le kL$ can be computed as

$$\begin{aligned}\hat{\mathbf{z}}(k) &= [\hat{z}(kL-L+1), ..., \hat{z}(kL)]^T \\ &= [u(kL-L+1) * \hat{s}, ..., u(kL) * \hat{s}]^T \\ &= \mathbf{Q}^{01}\mathbf{F}^{-1}\sum_{p=0}^{P_\mathrm{s}-1}\mathbf{U}_p(k)\hat{\mathbf{S}}_p,\end{aligned} \tag{28}$$

where

$$\mathbf{U}_p(k) = diag\{\mathbf{F}\mathbf{u}_p(k)\} \tag{29}$$

is the frequency-domain cancelling signal matrix with $\mathbf{u}_p(k) = \{u[(k-p-2)L+1], ..., u[(k-p)L]\}^T$. The estimated disturbance signal vector is

$$\begin{aligned}\hat{\mathbf{d}}(k) &= [\hat{d}(kL-L+1), ..., \hat{d}(kL)]^T \\ &= \mathbf{e}(k) - \hat{\mathbf{z}}(k).\end{aligned} \tag{30}$$

Recall that $\hat{\mathbf{d}}(k)$ is generated by $\mathbf{V}_p(k-2)$, and the pseudo frequency-domain error vector can be calculated as

$$\hat{\mathbf{E}}(k) = \hat{\mathbf{D}}(k) + \mathbf{G}^{01} \sum_{p=0}^{P_w-1} \mathbf{V}_p(k-2)\mathbf{W}_p(k), \tag{31}$$

where

$$\mathbf{G}^{01} = \mathbf{F} \begin{bmatrix} \mathbf{0}_L & \mathbf{0}_L \\ \mathbf{0}_L & \mathbf{I}_L \end{bmatrix} \mathbf{F}^{-1} \tag{32}$$

is the overlap-save projection matrix, and

$$\hat{\mathbf{D}}(k) = \mathbf{F}[\mathbf{0}_{1\times L}, \hat{\mathbf{d}}^T(k)]^T \tag{33}$$

is the reconstructed frequency-domain disturbance signal vector. The pseudo frequency-domain error vector $\hat{\mathbf{E}}(k)$ is only used for the weight update. The weight update equation is

$$\mathbf{W}_p(k+1) = \mathbf{W}_p(k) - \mu \mathbf{G}^{10} \mathbf{\Lambda}^{-1}(k) \mathbf{V}_p^H(k-2) \hat{\mathbf{E}}(k). \tag{34}$$

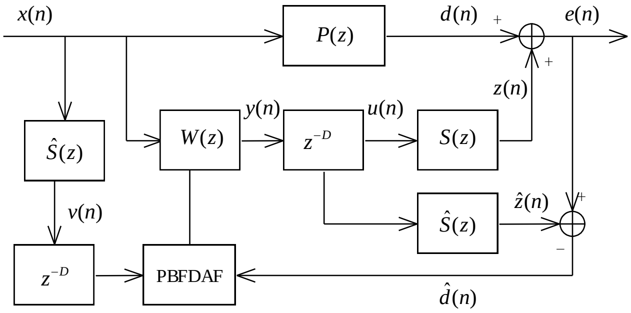

Figure 3. FPBMFxLMS algorithm.

Equations (31) and (34) describe the standard PBFDAF algorithm, which removes the effect of the secondary path. The output of the control filter can be computed using $\mathbf{W}_p(k+1)$ according to (25). However, the delay in the signal path still exists. The FPBMFxLMS algorithm is presented in Table 5.

Table 5. FPBMFxLMS algorithm [35].

Operation	Complexity
(1) Compute the frequency-domain reference matrix $\mathbf{U}_0(k)$ using (29)	One FFT
(2) Estimate the control signal at the error microphone $\hat{\mathbf{z}}(k)$ using (28)	One FFT and $4LP_\text{s}$ multiplications
(3) Estimate the disturbance signal vector $\hat{\mathbf{d}}(k)$ using (30)	0
(4) Compute the frequency-domain reference matrix $\mathbf{X}_0(k)$ using (8)	One FFT
(5) Compute the filtered reference signal vector $\mathbf{v}(k)$ using (11)	One FFT and $4LP_\text{s}$ multiplications
(6) Compute the FFT of $\mathbf{v}_0(k)$ using (15)	One FFT
(7) Commutate the pseudo frequency-domain error vector $\hat{\mathbf{E}}(k)$ using (31)	2 FFT and $4LP_\text{w}$ multiplications
(8) Update $\mathbf{\Lambda}(k)$ using (17)	$4L$ multiplications
(9) Update the weight vector using (34)	$2P_\text{w}$ FFTs and $6LP_\text{w}$ multiplications
(10) Compute the control filter out using (25)	One FFT and $4LP_\text{w}$ multiplications
(11) Compute the cancelling signal using (9) $u(n) = y(n-2L)$	0

Note: Steps (1)–(10) should be completed between kL and $kL+L$; Average and peak multiplications: $(8+2P_\text{w})A/L + 14P_\text{w} + 8P_\text{s}$.

4. Delayless Frequency-Domain ANC Algorithms

The algorithm in Table 5 removes the delay in the weight update but does not remove the delay in the signal path. It is more essential to remove the delay in the signal path because this delay has a major effect on the performance of broadband ANC systems. Several delayless frequency-domain ANC algorithms have been presented to remove delays in the signal path [44,46,47]. This section first reviews several delayless frequency-domain ANC algorithms, and then, a new delayless algorithm with the modified filtered-x scheme is proposed.

4.1. Qiu's Delayless Algorithm

The delayless frequency-domain ANC is presented in Figure 4. To avoid the delay in the signal path, the calculation of the adaptive filter output can be directly implemented using the time-domain convolution [40–43]

$$\mathbf{y}(k+1) = \begin{bmatrix} \mathbf{x}^T(kL+1)\mathbf{w}(k+1) \\ \cdots \\ \mathbf{x}^T(kL+L)\mathbf{w}(k+1) \end{bmatrix}. \tag{35}$$

If the filter length N_w is small, the complexity required in (35) may be acceptable. However, when N_w is large, the computational complexity may be too high for a resource-limited system. To resolve this problem, a hybrid fast convolution scheme is adopted to calculate the control filter output [44]

$$\begin{aligned} \mathbf{y}(k+1) &= [y(kL+1), ..., y(kL+L)]^T \\ &= \begin{bmatrix} \bar{\mathbf{x}}^T(kL+1)\mathbf{w}_0(k+1) \\ \cdots \\ \bar{\mathbf{x}}^T(kL+L)\mathbf{w}_0(k+1) \end{bmatrix} + \mathbf{\Theta}(k+1), \end{aligned} \tag{36}$$

where $\bar{\mathbf{x}}(n) = [x(n), x(n-1), \ldots, x(n-L+1)]^T$, and

$$\mathbf{\Theta}(k+1) = \mathbf{Q}^{01}\mathbf{F}^{-1}\sum_{p=1}^{P_\text{w}-1} \mathbf{X}_p(k+1)\mathbf{W}_p(k+1). \tag{37}$$

Please note that the calculation of $\mathbf{\Theta}(k+1)$ should be completed between kL and $kL+1$, and the first term of the right-hand side of (36) is calculated in the time domain on a sample-by-sample basis. The update equation is

$$\mathbf{W}_p(k+1) = \mathbf{W}_p(k) - \mu\mathbf{G}^{10}\mathbf{\Lambda}^{-1}(k)\mathbf{V}_p^H(k)\mathbf{E}(k). \tag{38}$$

The calculation of $\mathbf{w}_0(k)$ does not require extra computational cost and is expressed by

$$\mathbf{w}_0(k) = [\mathbf{I}_L \quad \mathbf{0}_L]\mathbf{F}^{-1}\mathbf{W}_0(k). \tag{39}$$

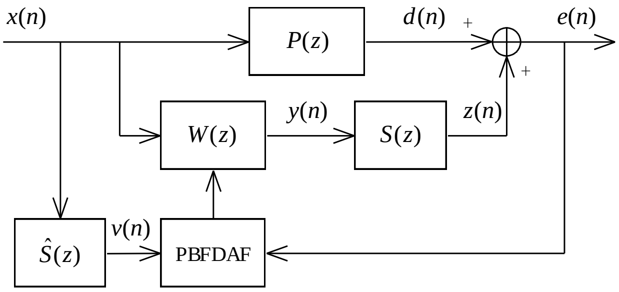

Figure 4. Qiu's delayless algorithm.

Because the calculation of $\boldsymbol{\Theta}(k+1)$ uses $\mathbf{W}_p(k+1)$, the weight update in (38) should be completed between kL and $kL+1$, which means the update-first approach must be used. By using this approach, the delay in the signal path is removed. However, this algorithm is not practical because (36) and (38) should be completed within one sampling period, which requires $(5+2P_\mathrm{w})A + 10LP_\mathrm{w} + 4LP_\mathrm{s} + L$ multiplications per sample. In addition, the algorithm requires L multiplications per sample between $kL+2$ and $kL+L$. Thus, the complexity of the algorithm is not evenly distributed over time, i.e., its peak complexity is rather high. A complexity measurement carried out in [46] indicated that the complexity of this algorithm in the first sample of each block is up to 200 times higher than the mean computational load. Even if a high-performance processor is employed to complete this task, most of this power is wasted. Furthermore, the delay in the weight update still exists, which limits the available step size for adaptation. Qiu's delayless algorithm is summarized in Table 6.

Table 6. Qiu's delayless algorithm [44].

Operation	Complexity
(1) Compute the frequency-domain reference matrix $\mathbf{X}_0(k)$ using (8)	One FFT
(2) Compute the filtered reference signal vector $\mathbf{v}(k)$ using (11)	One FFT and $4LP_\mathrm{s}$ multiplications
(3) Compute the FFT of $\mathbf{v}_0(k)$ using (15)	One FFT
(4) Update $\boldsymbol{\Lambda}(k)$ using (17)	$4L$ multiplications
(5) Compute the FFT of the error signal using (14)	One FFT
(6) Update the weight vector using (38)	$2P_\mathrm{w}$ FFTs and $6LP_\mathrm{w}$ multiplications
(7) Compute $\boldsymbol{\Theta}(k+1)$ using (37)	One FFT and $4L(P_\mathrm{w}-1)$ multiplications
(8) Compute the cancelling signal $\mathbf{y}(k+1)$ using (36)	L^2 multiplications

Note: Steps (1)–(7) should be completed between kL and $kL+1$; Average multiplications: $(5+2P_\mathrm{w})A/L + 10P_\mathrm{w} + 4P_\mathrm{s} + L$; Peak multiplications: $(5+2P_\mathrm{w})A + 10LP_\mathrm{w} + 4LP_\mathrm{s} + L$.

4.2. Fink's Delayless Algorithm

A computationally efficient delayless FDAF algorithm was presented in [46] that uses the fast filtering approach of [48]. The difference between Qiu's and Fink's delayless algorithms is twofold. First, the convolutions in the first and second partitions are calculated in the time domain for the Fink's algorithm. Second, the previous weight vector $\mathbf{W}_p(k)$ instead of $\mathbf{W}_p(k+1)$ is used to calculate the control filter output

$$\mathbf{y}(k+1) = \mathbf{y}_0(k+1) + \mathbf{y}_1(k+1) + \mathbf{b}(k+1), \tag{40}$$

where

$$\mathbf{y}_p(k+1) = \mathbf{Q}^{01}\mathbf{F}^{-1}\mathbf{X}_p(k+1)\mathbf{W}_p(k), p = 0, 1, \tag{41}$$

$$\begin{aligned}\mathbf{b}(k+1) &= [b(kL+1), ..., b(kL+L)]^T \\ &= \mathbf{Q}^{01}\mathbf{F}^{-1}\sum_{p=2}^{P_\mathrm{w}-1}\mathbf{X}_p(k+1)\mathbf{W}_p(k).\end{aligned} \tag{42}$$

It should be noted that (42) is completed in the k-th frame but not in one sampling period as in Qiu's algorithm, and hence the high peak complexity is avoided. The elements of $\mathbf{y}_0(k+1)$ and $\mathbf{y}_1(k+1)$ are computed in the time domain on a sample-by-sample basis

$$\mathbf{y}_p(k+1) = \begin{bmatrix} \bar{\mathbf{x}}^T(kL+1-pL)\mathbf{w}_p(k) \\ \cdots \\ \bar{\mathbf{x}}^T(kL+L-pL)\mathbf{w}_p(k) \end{bmatrix}. \tag{43}$$

The weight vector is updated using (38), which is completed between $kL+1$ and $kL+L$. Accordingly, the total complexity of the algorithm is distributed evenly in one frame. However, the delay in the weight adaptation is not removed. Fink's delayless algorithm is presented in Table 7.

The pipeline of the process should be considered carefully. There are two tasks, the time-domain convolution on a sample-by-sample basis and the frequency-domain procedure on a block-by-block basis. When new block data are collected, the task for the weight update in (38) and the calculation of $\mathbf{b}(k+1)$ are scheduled with low priority and should be completed in one block. When new data are available, the frequency-domain task is interrupted by the time-domain task. Then, the convolution operation is scheduled with high priority and should be completed in one sampling period. After the convolution operation has been executed, the interrupted frequency-domain task is recovered.

Table 7. Fink's delayless algorithm [46].

Operation	**Complexity**
(1) Compute the frequency-domain reference matrix $\mathbf{X}_0(k)$ using (8)	One FFT
(2) Compute the filtered reference signal vector $\mathbf{v}(k)$ using (11)	One FFT and $4LP_\mathrm{s}$ multiplications
(3) Compute the FFT of $\mathbf{v}_0(k)$ using (15)	One FFT
(4) Update $\mathbf{\Lambda}(k)$ using (17)	$4L$ multiplications
(5) Compute the FFT of the error signal using (14)	One FFT
(6) Update the weight vector using (38)	$2P_\mathrm{w}$ FFTs and $6LP_\mathrm{w}$ multiplications
(7) Compute $\mathbf{b}(k+1)$ using (42)	One FFT and $4L(P_\mathrm{w}-1)$ multiplications
(8) Compute the cancelling signal $\mathbf{y}(k+1)$ using (40)	$2L^2$ multiplications

Note: Steps (1)–(7) should be completed between kL and $kL+L$; Average and peak multiplications: $(5+2P_\mathrm{w})A/L + 10P_\mathrm{w} + 4P_\mathrm{s} + 2L$.

4.3. Proposed Delayless Algorithm

Several approaches have been proposed to remove the delay in the signal path or the weight update delay, but none of these algorithms can remove both kinds of delays. To address this problem, we propose a new delayless frequency-domain FxLMS algorithm as shown in Figure 5. First, we employ the modified filtered-x scheme to the frequency-domain ANC to remove the delay in the weight adaptation. Second, the delayless fast filtering approach in Section 4.2 is used to remove the signal path delay.

The estimate of the control signal at the location of the error microphone is

$$\begin{aligned}\hat{\mathbf{z}}(k) &= [\hat{z}(kL-L+1), ..., \hat{z}(kL)]^T \\ &= [y(kL-L+1) * \hat{s}, ..., y(kL) * \hat{s}]^T \\ &= \mathbf{Q}^{01}\mathbf{F}^{-1}\sum_{p=0}^{P_\mathrm{s}-1}\mathbf{Y}_p(k)\hat{\mathbf{S}}_p,\end{aligned} \tag{44}$$

where

$$\mathbf{Y}_p(k) = diag\{\mathbf{F}\mathbf{y}_p(k)\} \tag{45}$$

is the frequency-domain cancelling signal matrix with $\mathbf{y}_p(k) = \{y[(k-p-2)L+1], ..., y[(k-p)L]\}^T$. Then, we can use (30) to estimate the disturbance signal vector $\hat{\mathbf{d}}(k)$. The weight vector is updated as

$$\hat{\mathbf{E}}(k) = \hat{\mathbf{D}}(k) + \mathbf{G}^{01} \sum_{p=0}^{P_\mathrm{w}-1} \mathbf{V}_p(k)\mathbf{W}_p(k), \tag{46}$$

$$\mathbf{W}_p(k+1) = \mathbf{W}_p(k) - \mu \mathbf{G}^{10} \mathbf{\Lambda}^{-1}(k)\mathbf{V}_p^H(k)\hat{\mathbf{E}}(k). \tag{47}$$

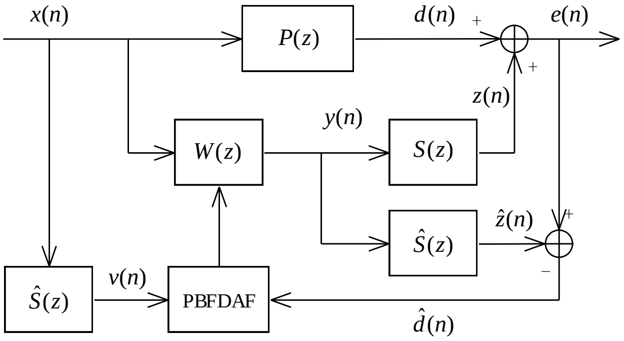

Figure 5. Proposed delayless algorithm.

Once we obtain the new weight vector $\mathbf{W}_p(k+1)$, we can use (40) to calculate the control filter output. By doing so, the delay in the signal path is removed by using the hybrid fast filtering approach, and the delay in the weight update is removed by means of the modified filtered-x structure. Furthermore, the total complexity is distributed evenly in one block, which makes the algorithm practical. The proposed delayless algorithm is presented in Table 8.

Table 8. Proposed delayless algorithm.

Operation	Complexity
(1) Compute the frequency-domain reference matrix $\mathbf{X}_0(k)$ using (8)	One FFT
(2) Compute the filtered reference signal vector $\mathbf{v}(k)$ using (11)	One FFT and $4LP_\mathrm{s}$ multiplications
(3) Compute the FFT of $\mathbf{v}_0(k)$ using (15)	One FFT
(4) Compute the frequency-domain cancelling matrix $\mathbf{Y}_0(k)$ using (45)	One FFT
(5) Estimate the control signal at the error microphone $\hat{\mathbf{z}}(k)$ using (44)	One FFT and $4LP_\mathrm{s}$ multiplications
(6) Estimate the disturbance signal vector $\hat{\mathbf{d}}(k)$ using (30)	0
(7) Compute the pseudo frequency-domain error vector $\hat{\mathbf{E}}(k)$ using (46)	Two FFTs and $4LP_\mathrm{w}$ multiplications
(8) Update $\mathbf{\Lambda}(k)$ using (17)	$4L$ multiplications
(9) Update the weight vector using (47)	$2P_\mathrm{w}$ FFTs and $6LP_\mathrm{w}$ multiplications
(10) Compute $\mathbf{b}(k+1)$ using (42)	One FFT and $4L(P_\mathrm{w}-1)$ multiplications
(11) Compute the cancelling signal $\mathbf{y}(k+1)$ using (40)	$2L^2$ multiplications

Note: Steps (1)–(10) should be completed between kL and $kL+L$; Average and peak multiplications: $(8+2P_\mathrm{w})A/L + 14P_\mathrm{w} + 8P_\mathrm{s} + 2L$.

5. Computational Complexity Analysis

The complexity of the aforementioned frequency-domain FxLMS algorithms is summarized in Table 9. We assume that one $2L$-point real FFT can be realized using one L-point complex FFT plus additional operations, requiring $2L\log_2(L) + 4L$ real multiplications [4]. The peak and average multiplications per sample required for the frequency-domain ANC algorithms involved are presented in Figure 6, where we use $L = 256$ and $P_\mathrm{s} = 2$. The average complexity of all the frequency-domain algorithms is lower than that of the FxLMS algorithm, which shows the complexity advantage of the frequency-domain adaptive algorithms. The FPBFxLMS I in [28,37] and Qiu's delayless algorithm in [44] both have high peak complexity and hence are unrealizable in real-time systems. The proposed delayless method achieves load-balanced implementation and is only slightly more complex than Fink's delayless algorithm in [46].

Table 9. Complexity of the frequency-domain ANC algorithms.

Algorithm	Average Multiplications per Sample	Peak Multiplications per Sample
FPBFxLMS I [28,37]	$(5+2P_\mathrm{w})A/L+10P_\mathrm{w}+4P_\mathrm{s}$	$(5+2P_\mathrm{w})A+10LP_\mathrm{w}+4LP_\mathrm{s}+4L$
FPBFxLMS II [39]	$(5+2P_\mathrm{w})A/L+10P_\mathrm{w}+4P_\mathrm{s}$	$(5+2P_\mathrm{w})A/L+10P_\mathrm{w}+4P_\mathrm{s}$
FPBMFxLMS [35]	$(8+2P_\mathrm{w})A/L+14P_\mathrm{w}+8P_\mathrm{s}$	$(8+2P_\mathrm{w})A/L+14P_\mathrm{w}+8P_\mathrm{s}$
Qiu [44]	$(5+2P_\mathrm{w})A/L+10P_\mathrm{w}+4P_\mathrm{s}+L$	$(5+2P_\mathrm{w})A+10LP_\mathrm{w}+4LP_\mathrm{s}+L$
Fink [46]	$(5+2P_\mathrm{w})A/L+10P_\mathrm{w}+4P_\mathrm{s}+2L$	$(5+2P_\mathrm{w})A/L+10P_\mathrm{w}+4P_\mathrm{s}+2L$
Proposed	$(8+2P_\mathrm{w})A/L+14P_\mathrm{w}+8P_\mathrm{s}+2L$	$(8+2P_\mathrm{w})A/L+14P_\mathrm{w}+8P_\mathrm{s}+2L$

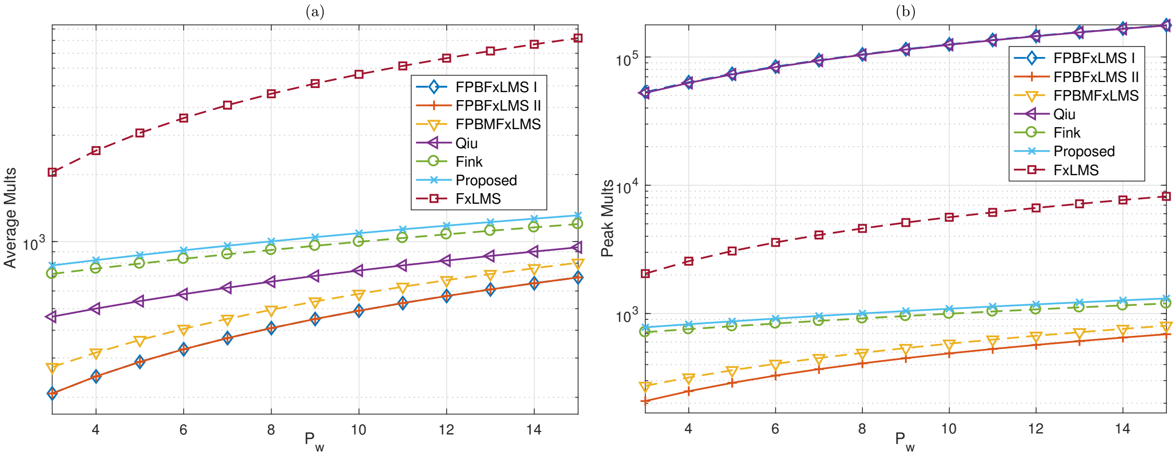

Figure 6. Peak and average complexities. (**a**) Average multiplications per sample. (**b**) Peak multiplications per sample.

6. Evaluations

In this section, we carry out computer simulations to evaluate the convergence performance of several frequency-domain algorithms in the context of ANC. The sampling rate is $f_s = 32768$ Hz. The length of the primary and secondary path are $N = 1536$ and $N_\mathrm{s} = 256$, respectively. The optimal solution of the control filter is computed using the MINT method [52] and has a length of $N_\mathrm{w} = 1536$. Their corresponding time-domain impulse responses are plotted in Figure 7. Figure 7c shows that maximum delay allowed in the signal path is $D = 256$ to guarantee the causality condition of the system. The signal-to-noise ratio (SNR) measured at the error microphone is 30 dB. Two signals, i.e., the narrow band signal between 100–800 Hz and a multi-tone signal at 40, 340 and 700 Hz, are used as the reference. The convergence performance is evaluated by the learning curve given by

$$A(k) = 10\log_{10}\left(P_e(k)/P_d(k)\right), \tag{48}$$

where $P_d(k)$ and $P_e(k)$ denote the powers of the disturbance signal $d(n)$ and the error signal $e(n)$, which are computed recursively as

$$P_d(k) = \beta P_d(k-1) + (1-\beta)\sum_{n=kL-L+1}^{kL} d^2(n), \tag{49}$$

$$P_e(k) = \beta P_e(k-1) + (1-\beta)\sum_{n=kL-L+1}^{kL} e^2(n), \tag{50}$$

where β is the smoothing factor. In the following, we use $\lambda = 0.8$ and $\beta = 0.8$.

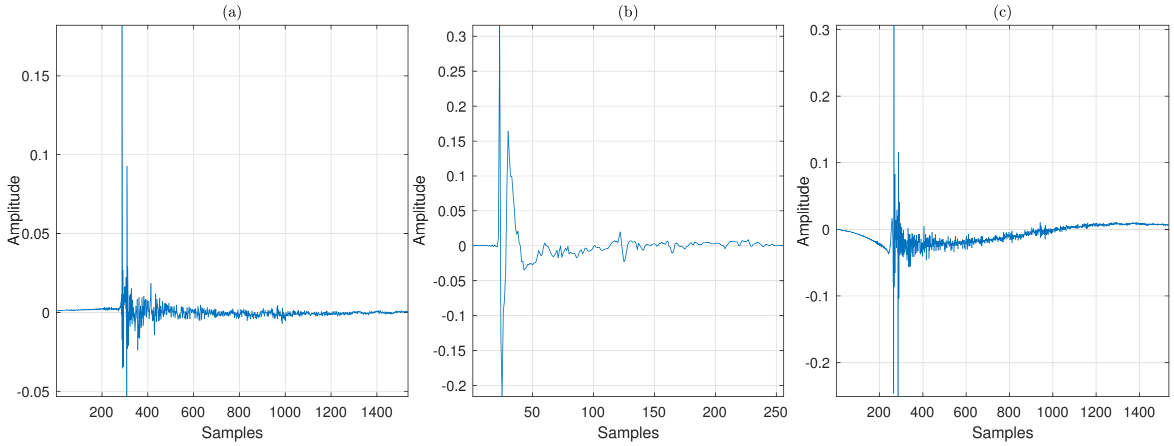

Figure 7. Primary and secondary paths. (**a**) Primary path. (**b**) Secondary path. (**c**) Least square solution of the control filer.

Figure 8 compares the convergence performance of the filtering-first and update-first approaches in Tables 1–4. The narrow band noise is used as input, and the block length is $L = 64$. The update-first approach is more stable than the filtering-first approach given the same step size, which agrees with the above analysis. In addition, the two methods have the same complexity, and hence the update-first approach is recommended for practical implementation.

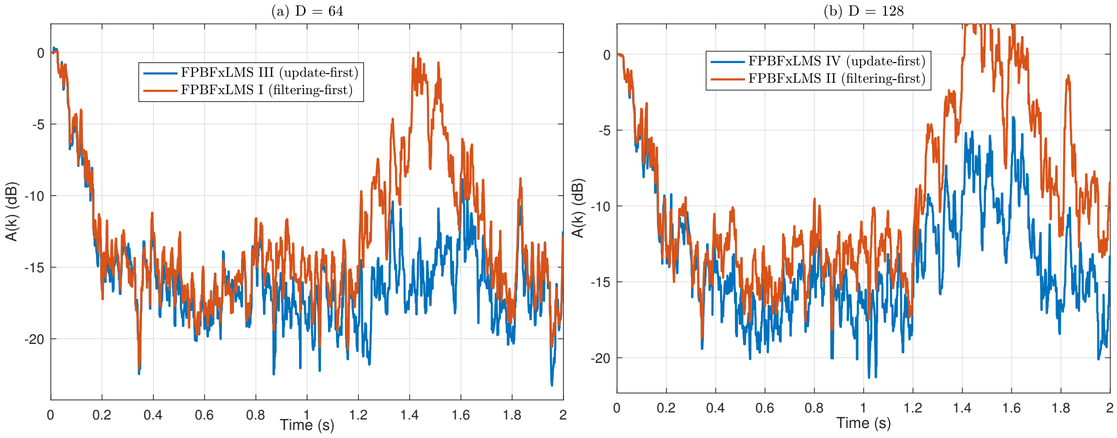

Figure 8. Convergence comparison between the update-first and filtering-first approaches. (**a**) FPBFxLMS I and FPBFxLMS III. (**b**) FPBFxLMS II and FPBFxLMS IV.

The effect of the block length on the convergence performance of several frequency-domain FxLMS algorithms is investigated in Figures 9 and 10. In Figure 9, we use the multi-tone signal as the reference, and the step size $\mu = 0.03$. The block length is set to $L = 64, 128, 256$. All the algorithms are convergent for $L = 64, 128$, while they diverge for $L = 256$. We found that all the algorithms can converge using a smaller step size $\mu = 0.001$ for the block length $L = 256$. For the multi-tone input, all the frequency-domain algorithms can converge if a sufficiently small step size is employed.

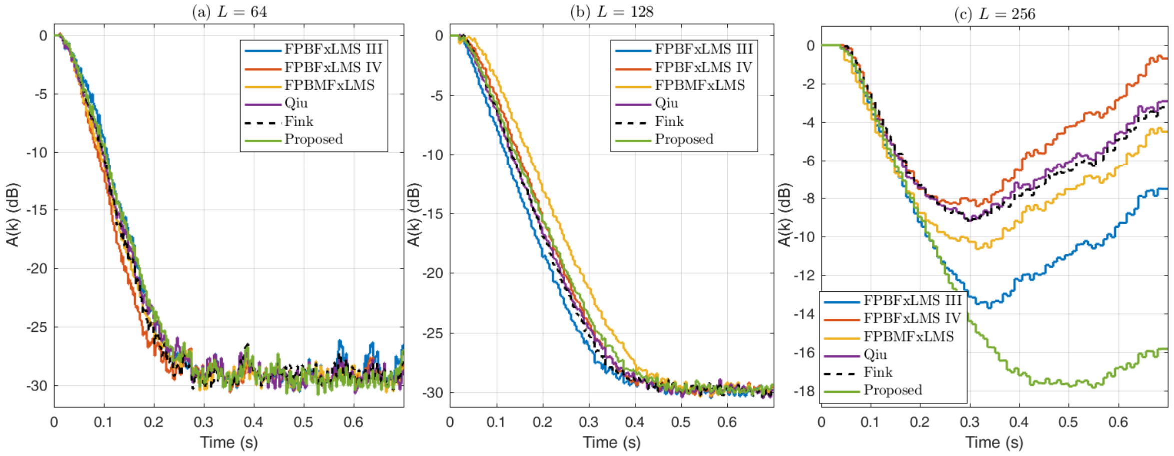

Figure 9. Learning curves of the frequency-domain ANC algorithms with multi-tone signal as input. (**a**) $L = 64$. (**b**) $L = 128$. (**c**) $L = 256$.

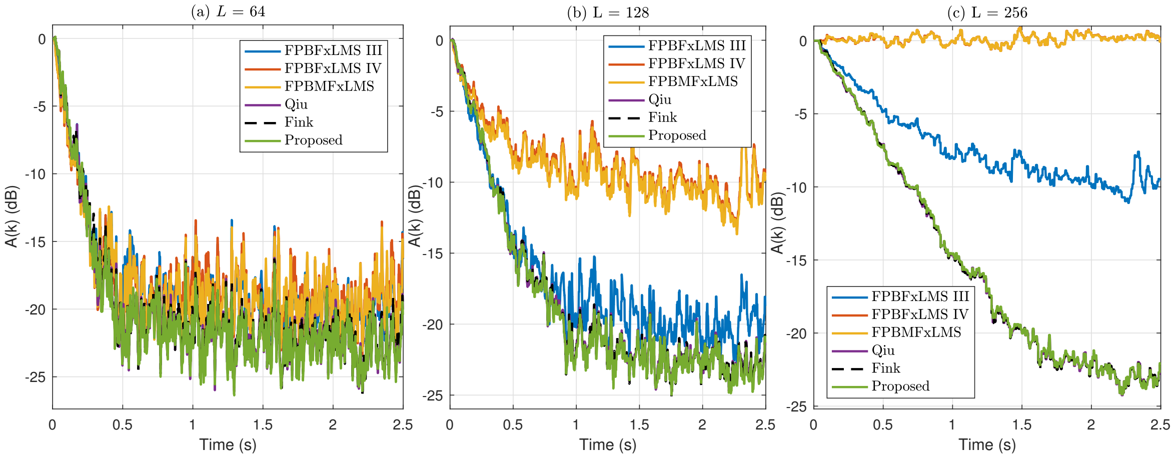

Figure 10. Learning curves of the frequency-domain ANC algorithms with narrow band noise as input. (**a**) $L = 64$. (**b**) $L = 128$. (**c**) $L = 256$.

The experiment in Figure 9 is repeated using the narrow band noise as input, and the learning curves are presented in Figure 10. The three delayless frequency-domain algorithms are stable and have a similar convergence for narrow band noise input in all the experiments. However, their convergence rate decreases as the block length. The convergence performance of the traditional frequency-domain FxLMS algorithms is dramatically affected by the block length. For $L = 64$, the causality condition is fulfilled for all the algorithm involved, and they exhibit a similar convergence. When the block length is $L = 256$, the FPBFxLMS IV and the FPBMFxLMS algorithms with a delay of $D = 512$ do not gain any noise reduction, and the FPBFxLMS III algorithm with a delay of $D = 256$ only exhibits a 10-dB noise reduction. The experiment demonstrates that the delay in the signal path has a major effect on the broadband noise reduction.

In Figure 11, we investigate the stability of the frequency-domain algorithms with and without the modified filtered-x structure. Specifically, we compare the FPBFxLMS IV, FPBMFxLMS, and three delayless algorithms. The narrow band white noise is used as the reference. We use a small step size $\mu = 0.01$ for Figure 11a and a large step size $\mu = 0.08$ for Figure 11b. All the algorithms involved exhibit a similar convergence performance when the step size is small. However, when a large step size is adopted, the frequency-domain algorithms without the modified filtered-x structure diverge, but the modified filtered-x algorithms are stable because the available step size of the modified filtered-x algorithm is larger than those of the FxLMS-type algorithms. The proposed delayless algorithm outperforms all the other methods.

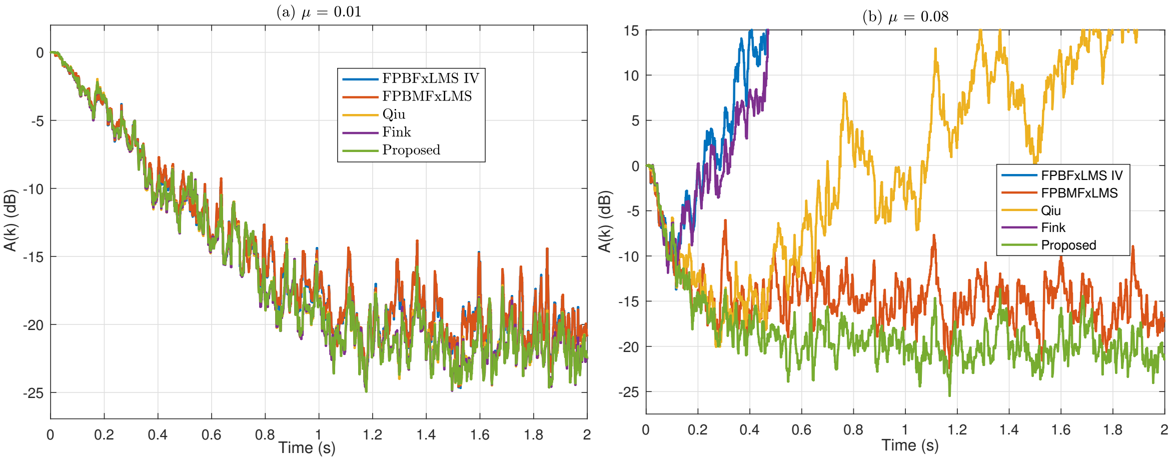

Figure 11. Algorithm stability of the traditional and modified structures. (**a**) $\mu = 0.01$. (**b**) $\mu = 0.08$.

7. Conclusions

This paper presented a comprehensive review of the frequency-domain FxLMS algorithms in terms of convergence performance, stability, and computational burden. Practical applications include active noise control of open windows [53], the ventilation system [54], and automotive cabins [55], where several thousands of coefficients are required to model the control filters. The frequency-domain algorithms used in ANC and system identification are quite different. First, broadband ANC is sensitive to algorithm latency, and a large delay may cause the system to fail. Second, direct use of an FDAF for ANC results in a delay in the weight update and the observation of the error signal, which reduces the maximum step size of the algorithm.

This paper reviewed the delayless filtering methods and the modified filtered-x structures to address the two delay problems. Specifically, the update-first approach was proposed to improve the stability of the FPBFxLMS algorithms. Several frequency-domain FxLMS algorithms have a very high peak complexity and thus are not practical for real-time systems, although these algorithms are correct in theory. A computationally efficient delayless frequency-domain algorithm was proposed that combines the hybrid fast filtering approaches and the modified filtered-x structure, which completely removes the aforementioned two types of delays. Extensive simulations were carried out to evaluate the convergence performance of the state-of-the-art frequency-domain FxLMS algorithms.

However, more work needs to be done regarding the frequency-domain FxLMS algorithms. For instance, the step size bound of the traditional frequency-domain FxLMS algorithms in Tables 1–4 should be determined. Analyzing the statistical convergence behavior of the frequency-domain FxLMS algorithms is an interesting topic [56–59].

Author Contributions: F.Y. performed the experiments and wrote the paper; Y.C., M.W. and F.A. gave valuable comments and edited the paper which improves the quality of the article; J.Y. conceived the experiments and proofread the paper.

Funding: This work of F. Yang was supported by National Natural Science Fund of China under Grants 61501449 and 11674348, Youth Innovation Promotion Association of Chinese Academy of Sciences under Grant 2018027, and IACAS Young Elite Researcher Project QNYC201722. The work of F. Albu was supported by a grant of the Romanian Ministry of Research and Innovation, CCCDI-UEFISCDI, project number PN-III-P3-3.1-PM-RO-CN-2018-0099/28BM/2018, within PNCDI III.

Acknowledgments: We thank Jianfeng Guo for preparing the impulse responses used in the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Basner, M.; Babisch, W.; Davis, A.; Brink, M.; Clark, C.; Janssen, S.; Stansfeld, S. Auditory and non-auditory effects of noise on health. *Lancet* **2014**, *383*, 1325–1332. [CrossRef]
2. Minichilli, F.; Gorini, F.; Ascari, E.; Bianchi, F.; Coi, A.; Fredianelli, L.; Licitra, G.; Manzoli, F.; Mezzasalma, L.; Cori, L. Annoyance judgment and measurements of environmental noise: A focus on Italian secondary schools. *Int. J. Environ. Res. Public Health* **2018**, *15*, 208–218. [CrossRef] [PubMed]
3. Nelson, P.; Elliott, S. *Active Control of Sound*; Academic Press: San Diego, CA, USA, 1992.
4. Kuo, S.; Morgan, D. *Active Noise Control Systems–Algorithms and DSP Implementations*; Springer: New York, NY, USA, 1996.
5. Lee, K.A.; Gan, W.S.; Kuo, S.M. *Subband Adaptive Filtering: Theory and Implementation*; Wiley: Chichester, UK, 2009.
6. Kajikawa, Y.; Gan, W.S.; Kuo, S.M. Recent advances on active noise control: Open issues and innovative applications. *APSIPA Trans. Signal Inf. Process.* **2012**, *1*, 1–21. [CrossRef]
7. Ferrer, M.; Gonzalez, A.; de Diego, M.; Piñero, G. Fast affine projection algorithms for filtered-x multichannel active noise control. *IEEE Trans. Audio Speech Lang. Process.* **2008**, *16*, 1396–1408. [CrossRef]
8. Albu, F.; Bouchard, M.; Zakharov, Y. Pseudo-affine projection algorithms for multichannel active noise control. *IEEE Trans. Audio Speech Lang. Process.* **2007**, *15*, 1044–1052. [CrossRef]
9. Yang, F.; Wu, M.; Yang, J.; Kuang, Z. A fast exact filtering approach to a family of affine projection-type algorithms. *Signal Process.* **2014**, *101*, 1–10. [CrossRef]
10. Farhang-Boroujeny, B. *Adaptive Filters: Theory and Applications*; John Wiley: Hoboken, NJ, USA, 2013.
11. Clark, G.A.; Mitra, S.K.; Parker, S.R. Block implementation of adaptive digital filters. *IEEE Trans. Acoust. Speech. Signal Process.* **1981**, *28*, 584–592. [CrossRef]
12. Ferrara, E.R. Fast implementation of LMS adaptive filters. *IEEE Trans. Acoust. Speech Signal Process.* **1980**, *28*, 474–475. [CrossRef]
13. Mansour, D.; Gray, A. Unconstrained frequency-domain adaptive filter. *IEEE Trans. Acoust. Speech Signal Process.* **1982**, *30*, 726–734. [CrossRef]
14. Shynk, J.J. Frequency-domain and multirate adaptive filtering. *IEEE Signal Process. Mag.* **1992**, *9*, 14–37. [CrossRef]
15. Sommen, P. Partitioned frequency domain adaptive filters. In Proceedings of the 23rd Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 30 October–1 November 1989; Volume 2, pp. 677–681.
16. Soo, J.S.; Pang, K.K. Multidelay block frequency-domain adaptive filter. *IEEE Trans. Acoust. Speech Signal Process.* **1990**, *38*, 373–376. [CrossRef]
17. Merched, R.; Sayed, A.H. An embedding approach to frequency domain and subband adaptive filtering. *IEEE Trans. Signal Process.* **2000**, *48*, 2607–2619. [CrossRef]
18. Enzner, G.; Vary, P. Frequency-domain adaptive Kalman filter for acoustic echo control in hands-free telephones. *Signal Process.* **2006**, *86*, 1140–1156. [CrossRef]
19. Yang, F.; Enzner, G.; Yang, J. Statistical convergence analysis for optimal control of DFT-domain adaptive echo canceler. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2017**, *25*, 1095–1106. [CrossRef]
20. Yang, F.; Enzner, G.; Yang, J. Frequency-domain adaptive Kalman filter with fast recovery of abrupt echo-path changes. *IEEE Signal Process. Lett.* **2017**, *24*, 1778–1782. [CrossRef]
21. Yang, F.; Yang, J. Optimal step-size control of the partitioned block frequency-domain adaptive filter. *IEEE Trans. Circuits Syst. II* **2018**, *65*, 814–818. [CrossRef]
22. Lu, C.; Yang, F.; Yang, J. A Frequency-domain Adaptive feedback cancellation algorithm based on convex combination. In Proceedings of the Asia-Pacific Signal and Information Processing Association, Annual Summit and Conference (APSIPA ASC), Kuala Lumpur, Malaysia, 12–15 December 2017; pp. 474–477.
23. Herbordt, W.; Kellermann, W. Computationally efficient frequency-domain robust generalized sidelobe canceller. In Proceedings of the International Workshop on Acoustic Echo and Noise Control (IWAENC), Tel-Aviv, Israel, 30 August–2 September 2010; pp. 51–54.
24. Shen, Q.; Spanias, A.S. Time and frequency domain X block LMS algorithms for single channel active noise control. In Proceedings of the 2nd Int. Congr. Recent Developments in Air- and Structure-Borne Sound Vibration, Auburn University, AL, USA, 4–6 March 1992; pp. 353–360.

25. Reichard, R.M.; Swanson, D.C. Frequency-domain implementation of the filtered-x algorithm with on-line system identification. In Proceedings of the Recent Advances in Active Control of Sound Vibration, Blacksburg, VA, USA, 28–30 April 1993; pp. 562–573.
26. Shen, Q.; Spanias, A.S. Frequency domain adaptive algorithms for multi-channel active sound control. In Proceedings of the Recent Advances in Active Control of Sound Vibration, Blacksburg, VA, USA, 28–30 April 1993; pp. 755–766.
27. Shen, Q.; Spanias, A. Time- and frequency-domain X-block least-mean-square algorithms for active noise control. *Noise Control Eng.* **1996**, *44*, 281–293. [CrossRef]
28. Das, D.P.; Panda, G.; Kuo, S.M. New block filtered-x LMS algorithms for active noise control systems. *IET Signal Process.* **2007**, *1*, 73–81. [CrossRef]
29. Das, K.K.; Satapathy, J.K. Frequency-domain block filtered-X NLMS algorithm for multichannel ANC. In Proceedings of the First International Conference on Emerging Trends in Engineering and Technology, Nagpur, Maharashtra, India, 16–18 July 2008; pp. 1293–1297.
30. Burdisso, R.A.; Vipperman, J.S.; Fuller, C.R. Causality analysis of feedforward controlled systems with broadband inputs. *J. Acoust. Soc. Am.* **1993**, *94*, 234–242. [CrossRef]
31. Kong, X.; Kuo, S.M. Study of causality constraint on feedforward active noise control systems. *IEEE Trans. Circuits Syst. II* **1999**, *46*, 183–186. [CrossRef]
32. Ahn, S.; Voltz, P.J. Convergence of the delayed normalized LMS algorithm with decreasing step size. *IEEE Trans. Signal Process.* **1996**, *44*, 3008–3016.
33. Bjarnason, E. Active noise cancellation using a modified form of the filtered-x LMS algorithm. In Proceedings of the 6th European Signal Processing Conference, Brussels, Belgium, 24–27 August 1992; pp. 1053–1056.
34. Kim, I.; Na, H.; Kim, K.; Park, Y. Constraint filtered-x and filtered-u least-mean-square algorithm for the active control of noise in ducts. *J. Acoust. Soc. Am.* **1994**, *95*, 3379–3389. [CrossRef]
35. Lorente, J.; Ferrer, M.; de Diego, M.; Belloch, J.A.; Gonzalez, A. GPU implementation of a frequency-domain modified filtered-x LMS algorithm for multichannel local active noise control. In Proceedings of the 52nd Audio Engineering Society International Conference, Guildford, UK, 2–4 September 2013.
36. Das, D.P.; Rout, N.K.; Panda, G. New partitioned block filtered-X LMS algorithm for active noise control. In Proceedings of the 1st International Conference on Emerging Trends in Signal Processing and VLSI Design, Hyderabad, India, 11–13 June 2010; pp. 112–117.
37. Rout, N.K.; Das, D.P.; Panda, G. Computationally efficient algorithm for high sampling-frequency operation of active noise control. *Mech. Syst. Signal Process.* **2015**, *56-57*, 302–319. [CrossRef]
38. Lorente, J.; Belloch, J.A.; Ferrer, M.; Gonzalez, A.; Belloch, J.A.; de Diego, M.; Piñero, G.; Vidal, A. Multichannel active noise control system using a GPU accelerator. In Proceedings of the Inter-Noise Conference, New York, NY, USA, 19–22 August 2012; pp. 7768–7780.
39. Lorente, J.; Ferrer, M.; Diego, M.D.; González, A. GPU implementation of multichannel adaptive algorithms for local active noise control. *IEEE/ACM Trans. Audio Speech, Lang. Process.* **2014**, *22*, 1624–1635. [CrossRef]
40. Kuo, S.M.; Yenduri, R.K.; Gupta, A. Frequency-domain delayless active sound quality control algorithm. *J. Sound Vib.* **2008**, *318*, 715–724. [CrossRef]
41. Morgan, D.R.; Thi, J.C. A delayless subband adaptive filter architecture. *IEEE Trans. Signal Process.* **1995**, *43*, 1819–1830. [CrossRef]
42. Park, S.J.; Yun, J.H.; Park, Y.C.; Youn, D.H. A delayless subband active noise control system for wideband noise control. *IEEE Trans. Signal Process.* **2001**, *9*, 892–897.
43. Wu, M.; Qiu, X.; Chen, G. An overlap-save frequency-domain implementation of the delayless subband ANC algorithm. *IEEE Trans. Acoust. Speech Signal Process.* **2008**, *16*, 1706–1710. [CrossRef]
44. Qiu, X.; Hansen, C.H. Multidelay adaptive filters for active noise control. In Proceedings of the 14th International Congress on Sound and Vibration (ICSV), Cairns, Australia, 9–12 July 2007; pp. 724–732.
45. Bendel, Y.; Burshtein, D.; Shalvi, O.; Weinstein, E. Delayless frequency domain acoustic echo cancellation. *IEEE Trans. Acoust. Speech Signal Process.* **2001**, *9*, 589–597. [CrossRef]
46. Fink, M.; Kraft, S.; Holters, M.; Zölzer, U. Load-balanced implementation of a delayless partitioned block frequency-domain adaptive filter. In Proceedings of the Workshop Audiosignal-und Sprachverarbeitung, Koblenz, Germany, 24–26 September 2013; pp. 2868–2879.

47. Sachau, D.; Jukkert, S. Real-time implementation of the frequency-domain FxLMS algorithm without block delay for an adaptive noise blocker. In Proceedings of the ISMA, Leuven, Belgium, 15–17 September 2014; pp. 93–105.
48. Gardner, S. Efficient convolution without input-output delay. *J. Audio Eng. Soc.* **1995**, *43*, 127–136.
49. Yang, F.; Wu, M.; Yang, J. A computationally efficient delayless frequency-domain adaptive filter algorithm. *IEEE Trans. Circuits Syst. II* **2013**, *60*, 222–226. [CrossRef]
50. McLaughlin, H.J. System and Method for an Efficiently Constrained Frequency-Domain Adaptive Filter. US5526426A, 11 June 1996.
51. Chan, K.S.; Farhang-Boroujeny, B. Analysis of the partitioned frequency-domain block LMS (PFBLMS) algorithm. *IEEE Trans. Signal Process.* **2001**, *49*, 1860–1874. [CrossRef]
52. Miyoshi, M.; Kaneda, Y. Inverse filtering of room acoustics. *IEEE Trans. Acoust. Speech Signal Process.* **1988**, *36*, 145–152. [CrossRef]
53. Lam, B.; Shi, C.; Gan, W.S. Active noise control systems for open windows: Current updates and future perspectives. In Proceedings of the 24th International Congress on Sound and Vibration, London, UK, 23–27 July 2017.
54. Larsson, M.; Johansson, S.; Claesson, I.; Håkansson, L. A module based active noise control system for ventilation systems, part I: Influence of measurement noise on the performance and convergence of the filtered-x LMS algorithm. *Int. J. Acoust. Vib.* **2009**, *14*, 188–195. [CrossRef]
55. Samarasinghe, P.N.; Zhang, W.; Abhayapala, T.D. Recent advances in active noise control inside automobile cabins: Toward quieter cars. *IEEE Signal Process. Mag.* **2016**, *33*, 61–73. [CrossRef]
56. Tobias, O.; Bermudez, J.; Bershard, N. Mean weight behavior of the filtered-x LMS algorithm. *IEEE Trans. Signal Process.* **2000**, *48*, 1061–1075. [CrossRef]
57. Carini, A.; Sicuranza, G.L. Transient and steady-state analysis of filtered-x affine projection algorithms. *IEEE Trans. Signal Process.* **2006**, *54*, 665–678. [CrossRef]
58. Ferrer, M.; Gonzalez, A.; de Diego, M.; Piñero, G. Transient analysis of the conventional filtered-x affine projection algorithm for active noise control. *IEEE Trans. Audio Speech Lang. Process.* **2011**, *19*, 652–657. [CrossRef]
59. Wang, T.; Gan, W.S. Stochastic analysis of FXLMS-based internal model control feedback active noise control systems. *Signal Process.* **2014**, *101*, 121–131. [CrossRef]