applied sciences

MDPI

Article

A Low-Complexity Volterra Filtered-Error LMS Algorithm with a Kronecker Product Decomposition

Jinhui Zhang [1,2], Chengshi Zheng [1,2,*], Fangjie Zhang [1,2] and Xiaodong Li [1,2]

[1] Key Laboratory of Noise and Vibration Research, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China; zhangjinhui@mail.ioa.ac.cn (J.Z.); zfj2012@mail.ioa.ac.cn (F.Z.); lxd@mail.ioa.ac.cn (X.L.)

[2] University of Chinese Academy of Sciences, Beijing 100049, China

[*] Correspondence: cszheng@mail.ioa.ac.cn

Abstract: Nonlinear active control is very important in many practical applications. Many well-known nonlinear active noise control algorithms may suffer from high computational complexity and low convergence speed, especially in the nonlinear secondary path case. Thus, it is still an actively researched topic for reducing complexity and improving the convergence rate. This paper presents a low-complexity Volterra filtered-error least mean square algorithm when taking a decomposable Volterra model into account for active control of nonlinear noise processes, which is referred as DVMFELMS. The computational complexity analysis shows that the proposed DVMFELMS algorithm can significantly reduce the nonlinear active noise control system's complexity. The simulation results further show that the proposed algorithm can achieve promising performance compared with the Volterra-based FELMS algorithm and other state-of-the-art nonlinear filters, while the decomposable error of the Volterra kernel may be introduced inevitably. Moreover, the proposed DVMFELMS algorithm shows a better convergence rate in the broadband primary noise case due to fewer parameters used in each sub-filter.

Keywords: low-complexity; active noise control; nonlinear; adaptive Volterra filter; Kronecker product decomposition

check for updates

Citation: Zhang, J.; Zheng, C.; Zhang, F.; Li, X. A Low-Complexity Volterra Filtered-Error LMS Algorithm with a Kronecker Product Decomposition. *Appl. Sci.* **2021**, *11*, 9637. https://doi.org/10.3390/app11209637

Academic Editor: Jun Yang

Received: 26 September 2021
Accepted: 12 October 2021
Published: 15 October 2021

1. Introduction

Active noise control has developed rapidly in the last few decades [1,2], and it has been successfully applied in many fields, such as heating, ventilating and air-conditioning systems [3], motor systems [4], and active headrests [5,6]. An active noise control system can generate a sound wave with the same amplitude but inverse phase at the canceling point by utilizing the secondary loudspeaker. Compared with passive methods, active noise control has shown its potential in canceling low-frequency environmental noises. A huge amount of research has been conducted in this field that focuses on the performance limitation of the active noise control system [7,8], algorithms [9], and stability [10].

The most commonly used algorithm for a single-channel feedforward active noise control system is the filtered-reference least mean square (FXLMS) algorithm modeled by the finite impulse response (FIR) filter due to its stability and low computational complexity. Although the FXLMS algorithm has already been successfully applied in active noise control systems, its performance may degrade in controlling nonlinear noise processes [11]. It is well-known that nonlinear phenomena may occur in many circumstances. First, when the primary noise has a high sound pressure level, the primary path may suffer nonlinear processes. Second, when the secondary actuator works at a high sound pressure level or the size of the secondary loudspeaker is small but works at low frequencies, nonlinear distortion may occur at the secondary actuator [12]. Third, the primary noise may come from a dynamic system, such as a chaotic noise process. The transfer function of the secondary path may have a non-minimum phase, so that the causality constraint will be violated [11]. In all of these situations, the traditional linear structure cannot model the nonlinear part of the system accurately enough. To overcome these challenges, many researchers have proposed numerous ways when taking the nonlinearity of the system into account, which can be roughly divided into two categories [13,14]. The former one uses artificial neural networks to model the nonlinearity of the system [15,16]. However, most artificial neural network-based algorithms are too computationally consuming for real applications of active noise control systems, which is out of the scope of this paper. The latter one employs the modeling capability of adaptive nonlinear filters including adaptive Volterra filters [11,17], the functional link network (FLNN) [18] and its generalized form (GFLNN) [19], the exponential functional link network (EFLNN) [20], the even mirror Fourier nonlinear (EMFN) filter [21], the Legendre neural network (LeNN) [22], Chebyshev filters (CF) [23,24], and so on. Furthermore, due to the different properties of different algorithms, convex combinations of the above-mentioned nonlinear filters have also been studied to achieve better performance [25,26].

For active noise control systems, the influence of the secondary path must be considered for most algorithms. The NANC algorithm with a linear secondary path (LSP) and with a nonlinear secondary path (NSP) was unified in [27] by introducing a virtual secondary path into NANC with the NSP case. For most nonlinear filters, NANC with NSP case is more computationally consuming compared to NANC with the LSP case. The NANC problem with a nonlinear secondary path (NSP), however, is more important in real applications [12]. When the memory length becomes large, the algorithms mentioned above often require high computational complexity that cannot be neglected. There are two ways to reduce the computational complexity of the NANC algorithm. One way is to use the recursive structure, which can be seen as an expansion of the infinite impulse response (IIR) filter. This kind of NANC algorithm includes the bilinear filter algorithm [28,29], recursive second-order Volterra filter [30], recursive even mirror Fourier nonlinear filter [31,32], adaptive function expansion bilinear filter [33,34], and so on. However, it is worth noting that the stability of all recursive algorithms mentioned above should be carefully considered, and more efforts must be made to make the recursive algorithm stable [35]. The other way is to use the simplified structure firstly introduced in [36], which uses a diagonal channel structure of nonlinear filters and channel reduced strategy that only uses a few channels near the main one. Note that the choice of the number of the reduced channels may significantly influence the steady-state performance.

This paper focuses on a group of adaptive Volterra filter-based algorithms, which are widely used in practice and perform well in a low-order nonlinear case. The major challenge of these traditional Volterra filter-based algorithms is that the computational complexity increases rapidly with the increase of the length of the memory and the order of the Volterra kernel. Recently, the Kronecker product decomposition has been introduced to linear system identification [37] and acoustic echo cancellation [38]. When using the Kronecker product decomposition, the computational complexity can be reduced dramatically in theory. In addition, it is demonstrated that the decomposable Volterra model outperforms the simplified structure in [36]. Motivated by these latest works, this paper proposes an adaptive Volterra-filtered error least mean square algorithm (DVMFELMS) that takes a decomposable Volterra model and filtered-error structure into account for the nonlinear active control problem. By comparing the computational complexity and the convergence performance of the proposed algorithm with the linear FELMS algorithm, VFELMS algorithm [11], FEGFLNN algorithm [19], and FEEMFNL algorithm [39], this paper shows that the proposed algorithm not only reduces the computational complexity, but also improves the convergence behavior slightly in the broadband noise cases.

The remainder of this paper is organized as follows. Section 2 formulates the problem of nonlinear active noise control. Section 3 derives the proposed DVMFELMS algorithm, and Section 4 compares its computational complexity with some state-of-the-art nonlinear filter algorithms. Simulation and conclusions are presented in Section 5 and Section 6, respectively.

2. Problem Formulation

2.1. Linear Feedforward ANC System

Before describing the nonlinear active noise control problem, the standard linear feedforward active noise control structure is first depicted in Figure 1 [1]. $u(n)$ is the reference signal at the time index n recorded by the reference microphone, and $y(n)$ is the output of the secondary actuator, which can be expressed as

$$y(n) = w(n) * u(n), \tag{1}$$

where $w(n)$ is the coefficient of the active noise control filter, and $*$ denotes the linear convolution. $e(n)$ is the residual error sensed by the error microphone, which can be expressed as

$$e(n) = d(n) + y(n) * \boldsymbol{s}(n), \tag{2}$$

where $S(z)$ and $\hat{S}(z)$ are the secondary path transfer function and its estimation, respectively. $P(z)$ is the primary path transfer function, and $d(n)$ is the primary noise signal at the error microphone.

In a standard linear feedforward active noise control structure, our goal is to find a suitable output signal $y(n)$ to minimize the mean square of the error signal $e(n)$. All transfer functions and control filters are assumed to be linear, which may be violated in many circumstances, as mentioned above.

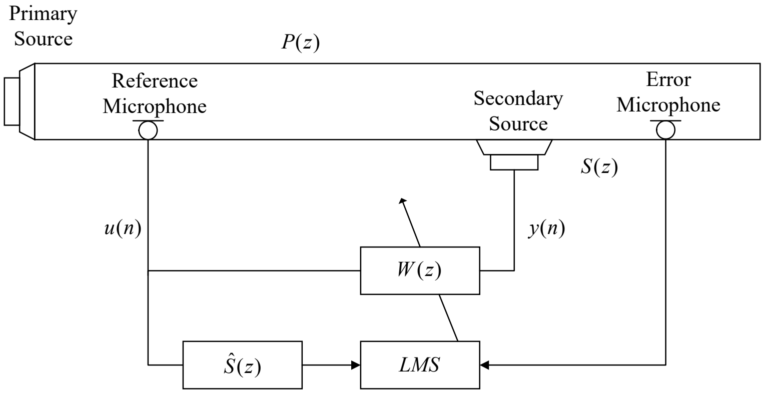

Figure 1. Block diagram of linear feedforward ANC system.

2.2. Nonlinear Feedforward ANC System

This paper expresses the nonlinear process using the well-known Volterra structure [11], where the Volterra series representation of $y(n)$ can be given by

$$y(n) = \sum_{p=0}^{\infty} y_p(n), \tag{3}$$

where

$$y_p(n) = \sum_{0 \le l_1, \cdots, l_p \le L} h_p(l_1, \cdots, l_p) \times u(n - l_1) \times \cdots \times u(n - l_p), \tag{4}$$

$h_p(l_1, \cdots, l_p)$ is the p-th-order Volterra kernel, and $L + 1$ is the length of the memory. Equation (4) can be further expressed in the vector form as below:

$$y_p(n) = \boldsymbol{U}_p^T(n)\boldsymbol{H}_p(n), \tag{5}$$

where T denotes transportation operation, and $\boldsymbol{U}_p(n)$ is the p-th-order reference vector with the length $(L+1)^p$, which is:

$$\boldsymbol{U}_p(n) = [u(n) \times \cdots \times u(n), \cdots, u(n - l_1) \times \cdots \times u(n - l_p), \cdots, u(n - L) \times \cdots \times u(n - L)]^T, \tag{6}$$

and similarly, $\boldsymbol{H}_p(n)$ is the p-th-order Volterra kernel vector with length $(L+1)^p$ given by

$$\boldsymbol{H}_p(n) = [h_p(0,\cdots,0),\ \cdots, h_p(l_1,\cdots,l_p),\ \cdots,\ h_p(L,\cdots,L)]^T. \tag{7}$$

This Volterra expression can be regarded as a Taylor expansion of the generally nonlinear function. Generally, only the first P order Volterra kernels are used to model the system's nonlinearity, although the truncated error may be introduced inevitably.

3. Proposed Algorithm

After considering a decomposable Volterra model, a low-complexity adaptive Volterra algorithm based on a filtered-error structure is proposed in this section for both nonlinear active noise control (NANC) with a linear secondary path (LSP) and that with a nonlinear secondary path (NSP). By using the Kronecker product decomposition, the system's computational complexity can be reduced dramatically, which is promising for active noise control systems.

3.1. DVMFXLMS

To derive the DVMFELMS algorithm, the adaptive Volterra algorithm based on a filtered-reference structure (DVMFXLMS) is first derived in this part. To model the system's nonlinearity, a P-order Volterra model with the length of the memory $L+1$ is utilized. By using the Kronecker product decomposition and the characteristic of the Kronecker product [38], Equation (5) can be rewritten as

$$\begin{aligned} y_p(n) &= (\boldsymbol{u}(n)\otimes\cdots\otimes\boldsymbol{u}(n))^T(\boldsymbol{w}_{p,1}(n)\otimes\cdots\otimes\boldsymbol{w}_{p,p}(n)),\\ &= (\boldsymbol{u}^T(n)\boldsymbol{w}_{p,1}(n))\otimes\cdots\otimes(\boldsymbol{u}^T(n)\boldsymbol{w}_{p,p}(n)),\\ &= (\boldsymbol{u}^T(n)\boldsymbol{w}_{p,1}(n))\times\cdots\times(\boldsymbol{u}^T(n)\boldsymbol{w}_{p,p}(n)), \end{aligned} \tag{8}$$

where $\otimes$ denotes the Kronecker product, and $\boldsymbol{u}(n)$ is the primary noise vector given by

$$\boldsymbol{u}(n) = [u(n),\ u(n-1),\ \cdots\ u(n-L)]^T, \tag{9}$$

where $\boldsymbol{w}_{p,1}(n)\otimes\cdots\otimes\boldsymbol{w}_{p,p}(n)$ is the nearest Kronecker product decomposition of $\boldsymbol{H}_p(n)$, which can be expressed as

$$\underset{w_{p,q}\ (0\le p\le P,\ 0\le q\le p)}{\arg\min} \quad ||\boldsymbol{H}_p(n)-\boldsymbol{w}_{p,1}(n)\otimes\cdots\otimes\boldsymbol{w}_{p,p}(n)||^2, \tag{10}$$

where $||\ .\ ||^2$ is the l_2-norm of a vector, and $\boldsymbol{w}_{p,q}(n)$ is the q-th filter of the p-th-order of the decomposed Volterra structure with memory length $L+1$, which is defined as

$$\boldsymbol{w}_{p,q}(n) = [w_{p,q}(n,0),\ w_{p,q}(n,1),\ \cdots\ w_{p,q}(n,L)]^T, with\ \ 0\le p\le P,\ 0\le q\le p. \tag{11}$$

Minimizing the mean square error can be given by

$$\begin{aligned} J &= E\Big\{e(n)^2\Big\},\\ &= E\Big\{\Big(d(n)+\hat{d}(n)\Big)^2\Big\}. \end{aligned} \tag{12}$$

According to the steepest descent algorithm, the proposed FXLMS can update the sub-filter $\boldsymbol{w}_{p,q}(n)$ as follows:

$$\boldsymbol{w}_{p,q}(n+1) = \boldsymbol{w}_{p,q}(n) - \mu_p\frac{\partial J}{\partial \boldsymbol{w}_{p,q}}\Big|_n, \tag{13}$$

where $\frac{\partial J}{\partial \boldsymbol{w}_{p,q}}\Big|_n$ is the single sample estimation of the gradient of J, and μ_p is the step size. Defining $y_{p,q}(n) = \boldsymbol{u}^T(n)\boldsymbol{w}_{p,q}(n)$, $y(n)$ can be rewritten as

$$y(n) = \sum_{p=1}^{P}\left(\prod_{q=1}^{p} y_{p,q}(n)\right). \tag{14}$$

We further assume that $\boldsymbol{w}_{p,q}(n)$ only changes slowly, and we can derive the gradient in (13) by substituting (12) and (14) into (15):

$$\frac{\partial J}{\partial \boldsymbol{w}_{p,q}}\Big|_n = \left(\left(\prod_{s=1}^{q-1} y_{p,s}(n) \prod_{s=q+1}^{p} y_{p,s}(n)\boldsymbol{u}(n)\right) * \frac{\partial \hat{d}(n)}{\partial \boldsymbol{y}(n)}\right)e(n), \tag{15}$$

defining the virtual secondary path filter $\boldsymbol{s}(n)$ with memory length $M+1$:

$$\boldsymbol{s}(n) = \left[\frac{\partial \hat{d}(n)}{\partial y(n)}, \frac{\partial \hat{d}(n)}{\partial y(n-1)}, \cdots \frac{\partial \hat{d}(n)}{\partial y(n-M)}\right]^T, \tag{16}$$

and, for notation convenience, defining an auxiliary sub-vector:

$$\overline{\boldsymbol{u}}_{p,q}(n) = \overline{y}_{p,q}(n)\boldsymbol{u}(n), \tag{17}$$

where $\overline{y}_{p,q}(n) = \left(\prod_{s=1}^{q-1} y_{p,s}(n) \prod_{s=q+1}^{p} y_{p,s}(n)\right)$. Substituting (15)–(17) into (13), we can derive the updating equation of $\boldsymbol{w}_{p,q}(n)$ for all p and q as

$$\boldsymbol{w}_{p,q}(n+1) = \boldsymbol{w}_{p,q}(n) - \mu_p(\overline{\boldsymbol{u}}_{p,q}(n) * \boldsymbol{s}(n))e(n). \tag{18}$$

It is worth noting that for NANC with the LSP case, the virtual secondary path filter $\boldsymbol{s}(n)$ can be modeled offline using a linear FIR filter. For NANC with the NSP case, the virtual secondary path filter will change with the iteration, even though the nonlinear secondary path is time-invariant. Thus, the nonlinear secondary path must be modeled firstly based on one of the nonlinear filters mentioned above, such as the Volterra filter. Then, the derivation in (16) is calculated using the modeled nonlinear filter. A sparse modeling method of the nonlinear secondary path using a second-order Volterra filter can be used here to reduce the computational complexity [39].

To improve the stability of the proposed algorithm, a normalized method is considered, and a regularization parameter λ is also introduced, which is similar to [38]. The updating equation of each sub-filter is finally given by

$$\boldsymbol{w}_{p,q}(n+1) = \boldsymbol{w}_{p,q}(n) - \widetilde{\mu}_p(\overline{\boldsymbol{u}}_{p,q}(n) * \boldsymbol{s}(n))e(n), \tag{19}$$

where

$$\widetilde{\mu}_p = \frac{\mu_p}{||\overline{y}_{p,q}(n)||^2 + \lambda_p}. \tag{20}$$

3.2. DVMFELMS

It should be noted that the auxiliary vector $\overline{\boldsymbol{u}}_{p,q}(n)$ defined in (17) did not hold the time-shifting property due to the presence of $\overline{y}_{p,q}(n)$; thus, each element of $\overline{\boldsymbol{u}}_{p,q}(n)$ must be filtered by the virtual secondary path filter even in the LSP case, which is a computationally consuming process. It was found that the filtered-error structure-based ANC system can offer comparable performance with less computational and structural complexity compared to a filtered-reference structure-based ANC [40]. Zhou and Debrunner introduced the filtered-error structure to nonlinear active noise to reduce the computational complexity in the NSP case [27]. The filtered-error structure's main advantage is that only the error signal needs to be filtered by the virtual secondary path filter, which benefits the proposed nonlinear active noise control algorithm based on the decomposable Volterra structure.

The diagram of the proposed DVMFELMS algorithm is depicted in Figure 2. To derive the DVMFELMS algorithm, the final updating equation of the DVMFXLMS algorithm is first recalled in the following:

$$\boldsymbol{w}_{p,q}(n+1) = \boldsymbol{w}_{p,q}(n) - \tilde{\mu}_p(n)\left(\overline{\boldsymbol{u}}_{p,q}(n) * \boldsymbol{s}(n)\right)e(n). \tag{21}$$

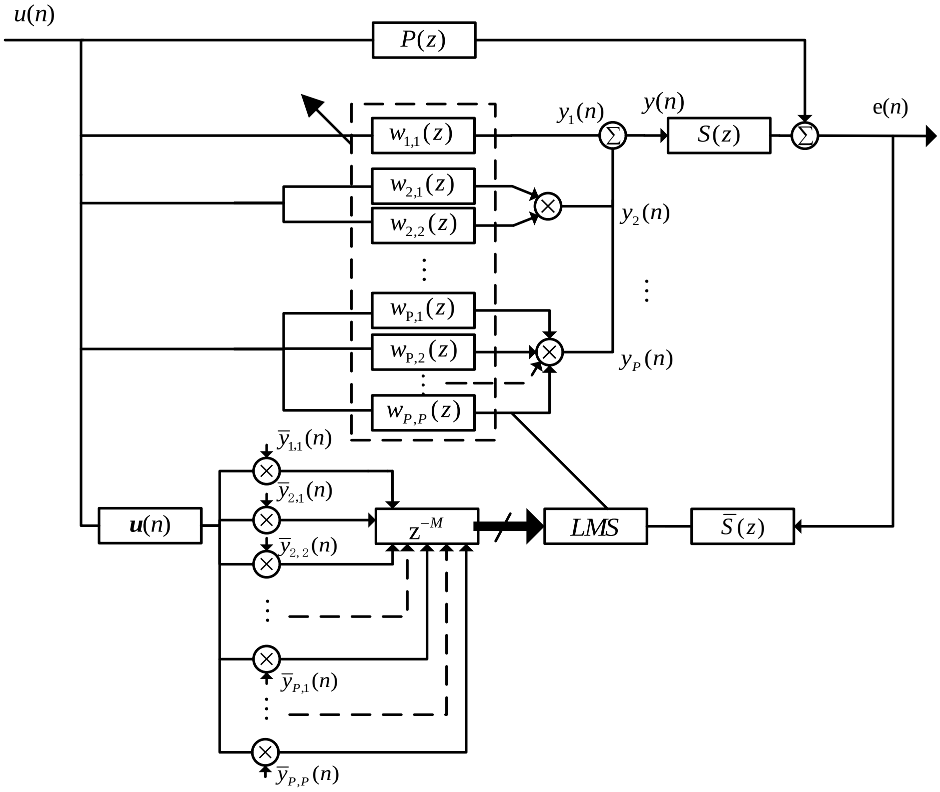

Figure 2. Block diagram of the proposed DVMFELMS algorithm.

Rewriting the update part of (21) as

$$\Delta\boldsymbol{w}_{p,q}(n) = -\left(\sum_{m=0}^{M} \overline{\boldsymbol{u}}_{p,q}(n-m)\boldsymbol{s}(n,m)\right)e(n), \tag{22}$$

and let $k = n - m + M$; then, $n = k + m - M$. Equation (22) can be written as

$$\Delta\boldsymbol{w}_{p,q}(n) = -\left(\sum_{m=0}^{M} e(k+m-M)\boldsymbol{s}(k+m-M,m)\right)\overline{\boldsymbol{u}}_{p,q}(k-M). \tag{23}$$

By substituting the time variable, only the error signal is filtered by the adjoint virtual secondary path filter $\overline{\boldsymbol{s}}(n)$, which is defined in the following instead of the whole new auxiliary sub-vector $\overline{\boldsymbol{u}}_{p,q}(n)$.

$$\overline{\boldsymbol{s}}(n) = [s(n,M),\ s(n-1,M-1),\ \cdots\ s(n-M,0)]^T, \tag{24}$$

It should be noted that in the LSP case, the adjoint virtual secondary path filter is just the flip of the linear FIR filter in (16), while for the NSP case, due to the time-variant property of the virtual secondary path filter, the last M values of the virtual secondary path filter must be stored to reconstruct the adjoint virtual secondary path filter. The final update equation of the proposed DVMFELMS algorithm is given by

$$\boldsymbol{w}_{p,q}(n+1) = \boldsymbol{w}_{p,q}(n) - \widetilde{\mu}_p \overline{\boldsymbol{u}}_{p,q}(n-M)e_f(n), \tag{25}$$

where $e_f(n)$ is the filtered-error signal calculated by filtering the error signal by the adjoint virtual secondary path filter defined in (24). From (25), one can see that the auxiliary vector $\overline{\boldsymbol{u}}_{p,q}(n)$ is just delayed by the M time scale without being filtered by the virtual secondary path filter, while only the error signal needs to be filtered by the adjoint virtual secondary path filter. Thus, the computational complexity of the proposed algorithm can be further reduced, especially when the memory of the secondary path becomes large. The detailed implementation of the proposed DVMFELMS algorithm is summarized in Algorithm 1.

Algorithm 1 Pseudo code of the proposed DVMFELMS algorithm.

```
Input: $L$,$M$,$P$,$\mu$,reference signals $u(n)$, desired signals $d(n)$, secondary path $\boldsymbol{s}(n)$, adjoint
    virtual secondary path $\overline{\boldsymbol{s}}(n)$
Initialization:
for $p = 1; p < P+1; p++$ do
    for $q = 1; q < p; q++$ do
        $\boldsymbol{w}_{p,q}(0) = [2^{1-q},\ 0,\ \cdots\ 0]^T$
    end for
    $\boldsymbol{w}_{p,p}(0) = [0,\ 0,\ \cdots\ 0]^T$
end for
Iteration:
for $n = 0$ to $END$ do
    for $p = 1; p < P+1; p++$ do
        $y_p(n) = 1$
        for $q = 1; q < p+1; q++$ do
            $y_p(n) = y_p(n)(\boldsymbol{u}^T(n)\boldsymbol{w}_{p,q}(n))$
        end for
    end for
    calculate the output signal $y(n) = \sum_{p=0}^{P} y_p(n)$
    calculate the error signal $e(n) = d(n) + y(n) * \boldsymbol{s}(n)$
    calculate the filtered-error signal $e_f(n) = e(n) * \overline{\boldsymbol{s}}(n)$
    for $p = 1; p < P+1; p++$ do
        for $q = 1; q < p+1; q++$ do
            calculate the auxiliary sub-vector $\overline{\boldsymbol{u}}_{p,q}(n) = \overline{y}_{p,q}(n)\boldsymbol{u}(n)$ and gain $\overline{\boldsymbol{u}}_{p,q}(n-
    M)$
            update $\boldsymbol{w}_{p,q}(n+1) = \boldsymbol{w}_{p,q}(n) - \widetilde{\mu}_p \overline{\boldsymbol{u}}_{p,q}(n-M)e_f(n)$
        end for
    end for
end for
Output: error signals $e(n)$, output signals $y(n)$, and filter coefficients $\boldsymbol{w}_{p,q}(n)$
```

4. Computational Complexity

The proposed algorithm's computational complexity is analyzed and compared with state-of-the-art nonlinear filters in NANC, including the Volterra filter [11], GFLNN [19], and second-order EMFNL (2OEMFNL) [39]. The computational complexity based on the filtered-reference and the filtered-error structure is considered. The computational load per iteration can be calculated according to L, M, and P. Note that only the multiplications required for all algorithms are considered in this section.

The number of multiplications of the proposed DVMFXLMS algorithm is first calculated in the following. The number of multiplications for generating the pth-order output $y_p(n)$ is $Lp + 1$. The number of multiplications for the pth-order filtered-reference signal vector is $Lp(M+1) + 3(p-2)$, and for the updating weights, the number of multiplications is Lp. In all, the total number of multiplications of the proposed DVMFXLMS algorithm is

$$\begin{aligned} T_{\text{DVMFXLMS}} &= \sum_{p=1}^{P} (Lp(M+3) + 3p - 5), \\ &= \frac{(P+1)P(LM+3L+3)}{2} - 5P, \end{aligned} \tag{26}$$

The computational complexity required for VFXLMS, FXGFLNN, 2OFXEMFNL, and the proposed DVMFXLMS algorithm for NANC with NSP is summarized in Table 1. It can be seen that the VFXLMS algorithm has the highest computational complexity. The 2OFXEMFNL algorithm and the FXGFLNN algorithm can achieve comparable performance with less computational complexity in higher-order nonlinearity. The proposed DVMFXLMS algorithm has the lowest computational complexity due to the Kronecker product decomposition and a similar property against the VFXLMS algorithm; however, due to the decomposition error of (10), its performance may have some degradations in some real applications, which will be analyzed in next section. From Table 1, it should be noted that the number of multiplications of the DVMFXLMS algorithm for generating the filtered signal are computationally consuming. The computational complexity required for the DVMFELMS algorithm is then calculated. From (25), the difference between the filtered-reference and filtered-error structure is the generation of the filtered signal. In the DVMFELMS algorithm, only the error vector needed to be filtered by the adjoint virtual secondary path filter; thus, only M multiplications are needed for all the sub-filter updates. While the generation of all p-th order auxiliary vector $\overline{\boldsymbol{u}}_{p,q}(n)$ needs $Lp + 3(p-2)$ multiplications, the total complexity of the DVMFELMS algorithm is calculated as

$$\begin{aligned} T_{\text{DVMFELMS}} &= \sum_{p=1}^{P} (3Lp + 3p - 1) + M, \\ &= \frac{(P+1)P(3L+3)}{2} - P + M, \end{aligned} \tag{27}$$

The computational complexity required for the VFELMS algorithm, the FEGFLNN algorithm, the 2OFEEMFNL algorithm, and the proposed DVMFELMS algorithm for NANC with an NSP case is summarized in Table 2. In order to make the computational complexity required for all algorithms visible, the number of multiplications required for all algorithms versus memory length is shown in Figure 3, while the length of the secondary path M is 10 for all the cases. It is worth noting that only the NANC with the NSP case is considered in this section because the NANC with the LSP can be seen as a special case for NANC with NSP. In addition, for structures that have a time-shifting property, such as the Volterra filter, GFLNN, and EMFNL, the computational complexity required for all algorithms in NANC with the LSP case based on the filtered-reference structure is almost the same as in NANC with the NSP case based on the filtered-error structure, while for the proposed algorithm, the computational complexity required for all algorithms in NANC with the LSP case based on the filtered-reference structure is the same as in NANC with the NSP case regardless of the computational complexity of the generation of the virtual secondary path filter. Figure 3 shows that the filtered-error structure can reduce the computational complexity in the NANC with the NSP case for all algorithms. With the growth of memory length, the computational complexity of the VFELMS algorithm can be intolerable for ANC systems. The computational complexity of the GFLNN and EMFNL structure is smaller than the VFELMS algorithm, while it is still large for a real-time system. However, by utilizing a decomposable Volterra structure and filtered-error structure, the proposed DVMFELMS algorithm can significantly reduce the computational complexity of the algorithm, which is promising in NANC systems, especially for the NSP case.

Table 1. Computational complexity of different algorithms for NANC with the NSP using filtered-reference structure.

Algorithm	VFXLMS	FXGFLNN	2OFXEMFNL	DVMFXLMS
Inputs signal	$\frac{(P+L-1)!}{L!(P-1)!}-1$	$2LP$	$L-1$	1
Output signal	$\frac{(P+L)!}{L!P!}-1$	$PL^2-(P-3)L$	$\frac{L^2+5L}{2}$	$\frac{LP(P+1)}{2}+P$
Filtered signal	$\frac{M(P+L)!}{L!P!}$	$(PL^2-(P-3)L)M$	$\frac{M(L^2+5L)}{2}$	$\frac{(L(M+1)+3)P(P+1)}{2}-5P$
Weight update	$2\left(\frac{(P+L)!}{L!P!}-1\right)$	$2PL^2-2(P-3)L$	L^2+5L	$\frac{LP(P+1)}{2}$
Total	$\frac{(P+L-1)!(MP+ML+4P+3L)}{P!L!}-4$	$(M+3)(PL^2-2(P-3)L)+2LP$	$\frac{(M+3)(L^2+5L)}{2}+L-1$	$\frac{(L(M+3)+3)P(P+1)}{2}-4P$

Table 2. Computational complexity of different algorithms for NANC with the NSP using filtered-error structure.

Algorithm	VFELMS	FEGFLNN	2OFEEMFNL	DVMFELMS
Inputs signal	$\frac{(P+L-1)!}{L!(P-1)!}-1$	$2LP$	$L-1$	1
Output signal	$\frac{(P+L)!}{L!P!}-1$	$PL^2-(P-3)L$	$\frac{L^2+5L}{2}$	$\frac{LP(P+1)}{2}+P$
Filtered signal	M	M	M	M
Weight update	$2\left(\frac{(P+L)!}{L!P!}-1\right)$	$2PL^2-2(P-3)L$	L^2+5L	$\frac{LP(P+1)}{2}$
Total	$\frac{(P+L-1)!(4P+3L)}{P!L!}-4$	$3(PL^2-2(P-3)L)+2LP+M$	$\frac{3(L^2+5L)}{2}+M+L-1$	$\frac{(P+1)P(3L+3)}{2}-P+M$

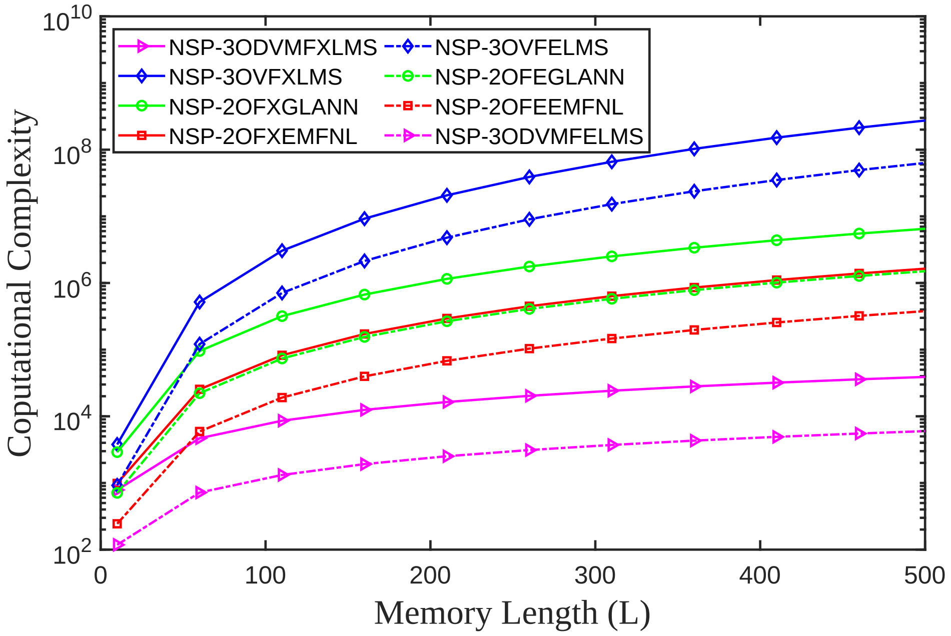

Figure 3. Number of multiplications versus memory length.

5. Performance Evaluation

This section evaluates the proposed DVMFELMS algorithm's performance and compares it with the FELMS algorithm, the VFELMS algorithm, the FEGFLNN algorithm, and the FEEMFNL algorithm. We evaluate the algorithms in three different nonlinear situations; the primary path has a nonlinear process in the first case, the primary and secondary paths have nonlinearity in the second case, and there is a real measured path with an intensely saturated signal in the last case. The normalized mean square error (NMSE) [11] is used to evaluate the performance of the algorithm, which is defined as

$$NMSE = 10\log_{10}\left(\frac{E\{e^2(n)\}}{\sigma_d^2}\right), \tag{28}$$

where σ_d^2 is the mean square energy of the primary noise. All results presented in this section are based on an ensemble average of 100 independent runs for each algorithm. The primary path $P(z)$ is set to be $-P(z)$ in the middle of the iteration to study all algorithms' stability and tracking capability. Furthermore, the step-sizes are all selected to achieve the maximum convergence rate with the minimum achievable NMSE.

5.1. Simulation with Nonlinear Primary Path

In the first experiment, the situation where the primary path has a nonlinear process is studied. The primary noise recorded by the error microphone is generated by passing the reference signal through a third-order Volterra model with cross terms, which is given by

$$\begin{aligned} d(n) = {} & u(n-5) + 0.8u(n-6) + 0.3u(n-7) + 0.4u(n-8) \\ & + 0.2u(n-5)u(n-6) - 0.3u(n-5)u(n-7) + 0.4u(n-5)u(n-8) \\ & + 0.04u(n-5)u(n-6)u(n-7) - 0.02u(n-5)u(n-6)u(n-8), \end{aligned} \tag{29}$$

where the transfer function of the secondary path is $S(z) = z^{-2} + 0.3z^{-3}$. The reference noise signal $u(n)$ is considered to be two cases; a sinusoidal wave of 500 Hz with a sampling rate of 8000 Hz is utilized in the first case, and a band-limited white Gaussian noise in the second case. The sinusoidal reference signal is given by

$$u(n) = \sqrt{2}\sin\left(\frac{2\pi \times 500 \times n}{8000}\right) + v(n), \tag{30}$$

where $v(n)$ is a white Gaussian noise with zero mean and unit variance, and the signal-to-noise ratio (SNR) is set to be 30 dB. The memory length for all algorithms is chosen to be 10. The order of the Volterra kernel is chosen to be 3, and the order of the functional link neural network and even mirror Fourier nonlinear filter is chosen to be 2. The step size for the FELMS algorithm is chosen to be $\mu_1 = 2 \times 10^{-3}$, step sizes for the third-order Volterra filter are set to $\mu_1 = 2 \times 10^{-3}, \mu_2 = 1 \times 10^{-3}, \mu_3 = 6 \times 10^{-4}$, step sizes for the 2OFEGFLNN algorithm and 2OFEEMFNL algorithm are set to $\mu_1 = 2 \times 10^{-3}, \mu_2 = 1 \times 10^{-3}$, and step sizes for the 3ODVMFELMS algorithm are set to $\mu_1 = 2 \times 10^{-3}, \mu_2 = 2 \times 10^{-2}$, $\mu_3 = 1 \times 10^{-4}, \lambda_2 = 0.6, \lambda_3 = 0.04$.

Figure 4 shows the NMSE achieved by all five algorithms versus the number of iterations in the sinusoidal noise case. It can be seen from Figure 4 that the reduction of primary noise achieved by the FELMS algorithm, the 3OVFELMS algorithm, the 2OFEGFLNN algorithm, the 2OFEEMFNL algorithm, and the 3ODVMFELMS algorithm is about 22.5 dB, 25.3 dB, 25.9 dB, 26.2 dB, and 26.3 dB, respectively. All nonlinear algorithms achieve better performance against the FELMS algorithm. The proposed DVMFELMS algorithm performs slightly better in the steady-state performance and the convergence rate in this case. It should be noted that the proposed DVMFELMS algorithm can even outperform the VFELMS algorithm in NMSE, which might be explained by the Volterra model used in the VFELMS algorithm being over-fitting for this simulation case.

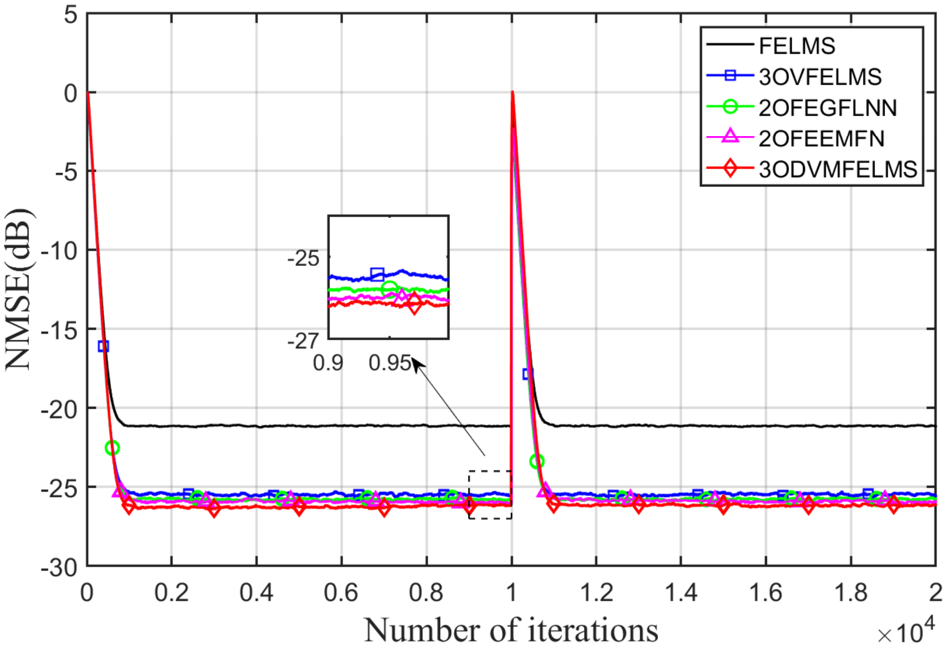

Figure 4. Performance comparison using the nonlinear primary path with sinusoidal reference signal.

For results of the broadband noise, the frequency range of the band-limited noise is chosen to be from 50 Hz to 1000 Hz, among which the ANC system usually works well. The signal is normalized to unity, and a white Gaussian noise with zero mean and unit variance with an SNR of 30 dB is also added. All the transfer functions and nonlinearities used are exactly the same as in the first case. The step size for the FELMS algorithm is chosen to be $\mu_1 = 6 \times 10^{-3}$, step sizes for the third-order Volterra filter are set to $\mu_1 = 6 \times 10^{-3}, \mu_2 = 4 \times 10^{-3}, \mu_3 = 6 \times 10^{-4}$, step sizes for the 2OFEGFLNN algorithm and 2OEMFNL algorithm are set to $\mu_1 = 6 \times 10^{-3}, \mu_2 = 1 \times 10^{-3}$, and step sizes for the 3ODVMFELMS algorithm are set to $\mu_1 = 6 \times 10^{-3}, \mu_2 = 4 \times 10^{-3}, \mu_3 = 1 \times 10^{-3}$, $\lambda_2 = 0.02, \lambda_3 = 0.02$. Figure 5 shows that the reduction of primary noise achieved by the FELMS algorithm, the 3OVFELMS algorithm, the FEGFLNN algorithm, the 2OEMFNL algorithm, and the 3ODVMFELMS algorithm in the broadband noise case is about 19.8 dB, 23.8 dB, 23.2 dB, 23 dB, and 23.6 dB, respectively. It is worth noting that in the broadband case, the VFELMS algorithm outperforms the 2OFEGFLNN algorithm and the 2OEMFNL algorithm in NMSE, which coincides with the results in [19]. Nevertheless, the 2OEMFNL algorithm has a fast convergence rate due to its inherent property, as shown in [27]. The proposed DVMFELMS algorithm can achieve comparable steady-state performance against the VFELMS algorithm, while with a slightly faster convergence rate. This may be explained by the fewer parameters used in the DVMFELMS algorithm.

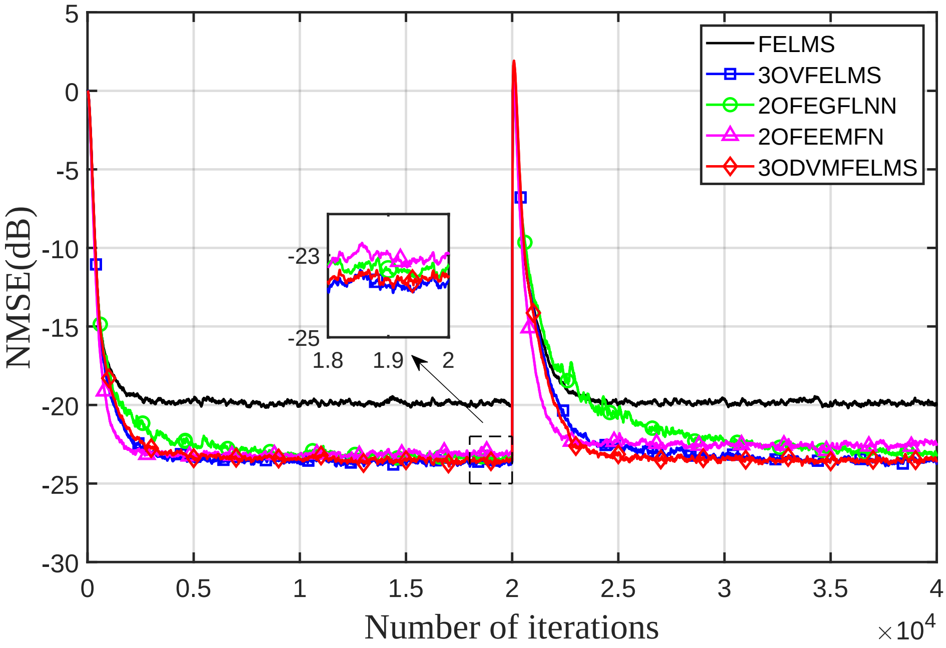

Figure 5. Performance comparison using the nonlinear primary path with band-limited white Gaussian reference signal.

5.2. Simulation with Nonlinear Primary Path and Nonlinear Secondary Path

To further demonstrate the effectiveness of the proposed DVMFELMS algorithms in the NANC/NSP case, both primary and secondary paths are considered to have nonlinearity in this experiment. The primary noise recorded by the error microphone is generated by passing the reference signal through a second-order Volterra model given by

$$\begin{aligned} d(n) &= u(n-5) + 0.8u(n-6) + 0.3u(n-7) + 0.4u(n-8) \\ &\quad + 0.2u(n-5)u(n-6) - 0.3u(n-5)u(n-7) + 0.4u(n-5)u(n-8), \end{aligned} \tag{31}$$

while the secondary path is considered to be an NLN model used in [27], which is expressed as

$$\begin{aligned} l_1 &= 1 - 0.6z^{-1} + 0.05z^{-2}, \\ N(z_1(n)) &= 3.3tanh(0.3z_1(n)), \\ l_2 &= 1 + 0.2z^{-1} + 0.05z^{-2}. \end{aligned} \tag{32}$$

Figure 6 depicts the NMSE obtained by all five active noise control algorithms versus the number of iterations by using the same sinusoidal signal as the reference signal in (30). The memory length for all algorithms is chosen to be 10, and the order of the Volterra kernel is chosen to be 2. The step size for the FELMS algorithm is chosen to be $\mu_1 = 2 \times 10^{-1}$, step sizes for the second-order Volterra filter are set to $\mu_1 = 2 \times 10^{-1}, \mu_2 = 2 \times 10^{-1}$, step sizes for the FEGFLNN algorithm are chosen to be $\mu_1 = 2 \times 10^{-1}, \mu_2 = 4 \times 10^{-3}$, step sizes for the 2OEMFNL algorithm are set to $\mu_1 = 2 \times 10^{-1}, \mu_2 = 1 \times 10^{-2}$, and step sizes for the 2ODVMFELMS algorithm are set to $\mu_1 = 2 \times 10^{-1}, \mu_1 = 2 \times 10^{-2}, \lambda_2 = 0.02$. Figure 6 shows that the reductions of primary noise achieved by the FELMS algorithm, the 2OVFELMS algorithm, the 2OFEGFLNN algorithm, the 2OFEEMFNL algorithm, and the proposed 2ODVMFELMS algorithm are about 20.8 dB, 29.7 dB, 29.3 dB, 29.1 dB, and 29.7 dB, respectively. It should be noted that the FEEMFNL algorithm shows bad convergence speed in the NSP case. The proposed DVMFELMS algorithm can still achieve comparable performance with the lowest computational complexity in the NSP case. However, the convergence rate of the proposed 2ODVMFELMS algorithm is worse in these cases because of the sinusoidal signal and NSP used.

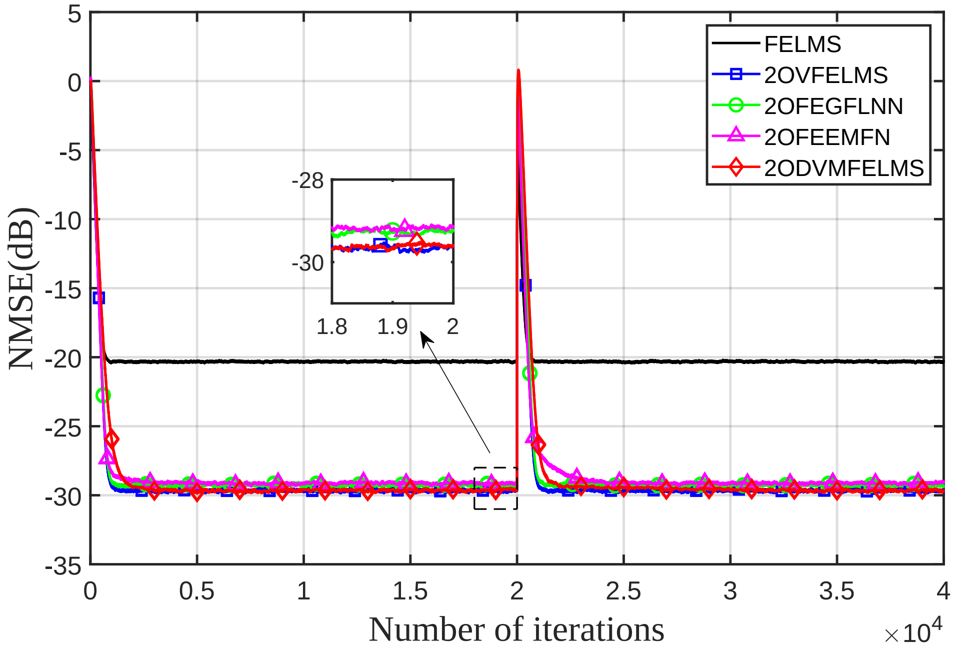

Figure 6. Performance comparison using the nonlinear primary path and secondary path with sinusoidal reference signal.

For broadband results, the band-limited white Gaussian noise used here is exactly the same as in experiment 2. The SNR is also set to be 30 dB. All the transfer functions are exactly the same as in the first case. Figure 7 shows the NMSE achieved by all five algorithms versus the number of iterations. The step size for the FELMS algorithm is chosen to be $\mu_1 = 2 \times 10^{-1}$, step sizes for the second-order Volterra filter are set to $\mu_1 = 2 \times 10^{-1}, \mu_2 = 2 \times 10^{-1}$, step sizes for the 2OFEGFLNN algorithm are chosen to be $\mu_1 = 2 \times 10^{-1}, \mu_2 = 4 \times 10^{-3}$, step sizes for the 2OEMFNL algorithm are set to $\mu_1 = 2 \times 10^{-1}$, $\mu_2 = 1 \times 10^{-2}$, and step sizes for the 2ODVMFELMS algorithm are set to $\mu_1 = 2 \times 10^{-1}$, $\mu_1 = 2 \times 10^{-2}, \lambda_2 = 0.02$. One can see from Figure 7 that the reductions of primary noise achieved by the FELMS algorithm, the 2OVFELMS algorithm, the 2OFEGFLNN algorithm, the 2OEMFNL algorithm, and the 2ODVMFELMS algorithm are about 17.3 dB, 24.1 dB, 23.7 dB, 23.5 dB, and 23.8 dB, respectively. It can be seen that the FEFLNN algorithm and the FEEMFNL algorithm show worse performance in steady-state NMSE and convergence speed than the Volterra filter-based algorithm in the broadband reference signal and NSP case. The proposed DVMFELMS algorithm has performance comparable with the VFELMS algorithm, with a dramatic reduction of computational complexity.

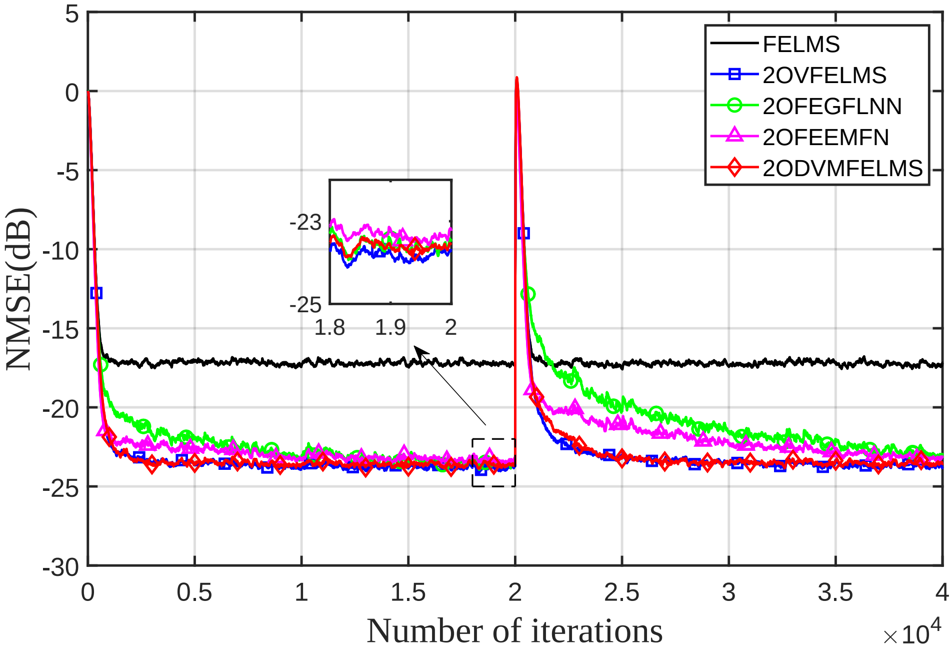

Figure 7. Performance comparison using the nonlinear primary path and secondary path with band-limited white Gaussian reference signal.

5.3. Simulation with Measured Primary Path and Secondary Path

In this experiment, a real measured primary and secondary path is used to justify the proposed DVMFELMS algorithm's effectiveness in a real application. The amplitude and phase response of both primary and secondary paths is shown in Figure 8. The reference signal is chosen to be three sinusoidal waves at normalized frequencies of 0.01, 0.02, and 0.08, which is normalized to unity power [29]. A white Gaussian noise with zero mean and unit variance with a signal-to-noise ratio of 30 dB is also added. Furthermore, the reference signal is assumed to have strong saturation generated by clipping the reference at 50% of the maximum signal value. The memory length of the primary and secondary path is chosen to be 80. Figure 9 shows the NMSE achieved by all five algorithms versus the number of iterations. The step size for the FELMS algorithm is chosen to be $\mu_1 = 1 \times 10^{-4}$, step sizes for the second-order Volterra filter are set to $\mu_1 = 1 \times 10^{-4}, \mu_2 = 1 \times 10^{-4}$, step sizes for the 2OFEGFLNN algorithm are chosen to be $\mu_1 = 1 \times 10^{-4}, \mu_2 = 5 \times 10^{-6}$, step sizes for the 2OFEEMFNL algorithm are set to $\mu_1 = 1 \times 10^{-4}, \mu_2 = 4 \times 10^{-5}$, and step sizes for the 2ODVMFELMS algorithm are set to $\mu_1 = 1 \times 10^{-4}, \mu_1 = 1 \times 10^{-3}, \lambda_2 = 0.02$. Figure 9 shows that the reduction of primary noise achieved by the FELMS algorithm, the 2OVFELMS algorithm, the FEGFLNN algorithm, the 2OEMFNL algorithm, and the 2ODVMFELMS algorithm in the real measured primary and secondary path cases is about 27.5 dB, 29.7 dB, 29.6 dB, 29.9 dB, and 29.6 dB, respectively. From Figure 9, it can be seen that the 2OEMFNL algorithm outperforms all algorithms in NMSE, while having the lowest convergence speed. The proposed DVMFELMS algorithm can achieve almost the same NMSE compared with the VFELMS algorithm in this case. The number of multiplications in this experiment for all four nonlinear algorithms are 278960, 39210, 10289, and 958, respectively, which shows the potential in the real application of the proposed DVMFELMS algorithm.

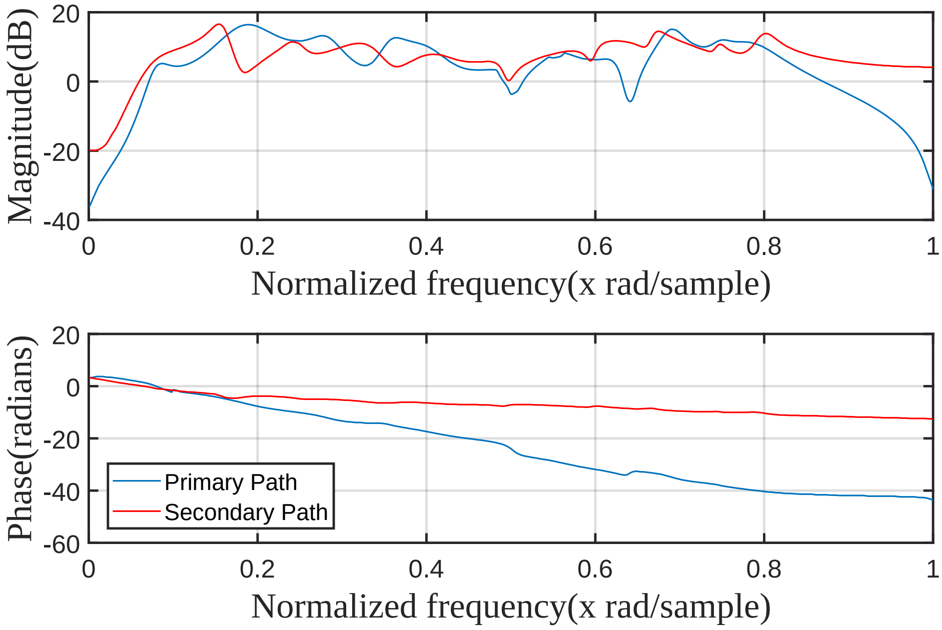

Figure 8. Amplitude and phase response of measured primary and secondary path.

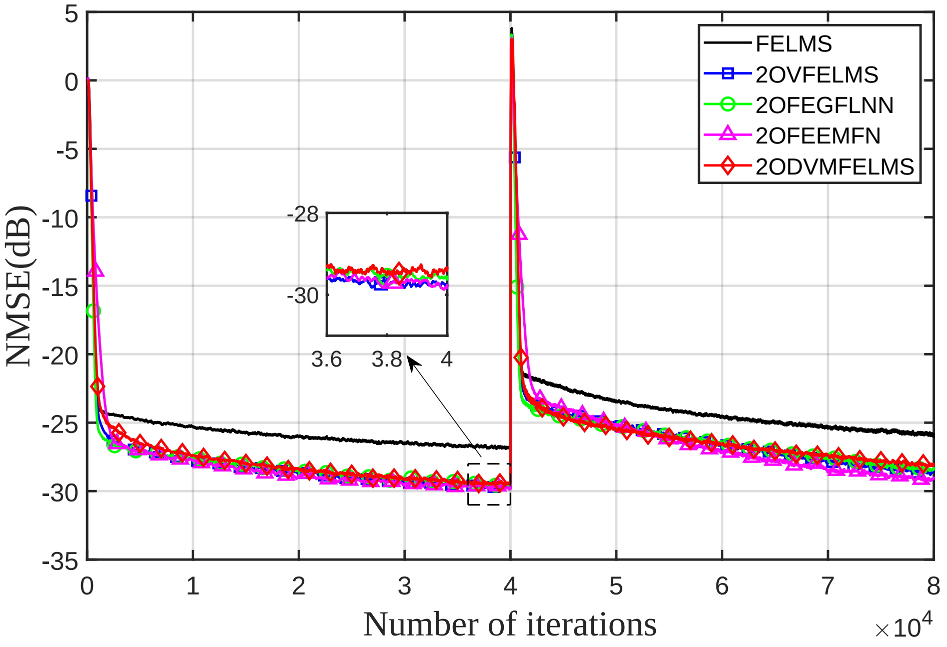

Figure 9. Performance comparison using measured primary and secondary path with saturated reference signal.

6. Conclusions

This paper proposes a very low-complexity adaptive Volterra filter-error least mean square algorithm for the active control of nonlinear noise processes after considering a decomposable Volterra model and filtered-error structure. The computational complexity analysis shows that the proposed DVMFELMS algorithm can significantly reduce the complexity, especially for the NANC system with the NSP case. The simulation result shows that the proposed algorithm can achieve comparable performance with the VFELMS algorithm and other state-of-the-art nonlinear filters in simulation conditions, while the decomposable error of the Volterra kernel may inevitably be introduced. In addition, the proposed DVMFELMS algorithm shows potential in a better convergence property due to the fewer parameters used in each sub-filter. Further work should concentrate on studying the step control to improve the stability of the DVMFELMS algorithm for practical applications.

Author Contributions: All authors have contributed to the work presented in this article. J.Z. derived the algorithm and conducted the analysis and simulation; C.Z. proposed the scheme, directed the work, analyzed the obtained results, and revised the manuscript. F.Z. conducted the simulation and interpreted the obtained results; X.L. analyzed the the obtained results and revised the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

Abbreviations

The following abbreviations are used in this manuscript:

FXLMS	Filtered-Reference Least Mean Square
FELMS	Filtered-Error Least Mean Square
FIR	Finite Impulse Response
IIR	Infinite Impulse Response
NANC	Nonlinear Active Noise Control
LSP	Linear Secondary Path
NSP	Nonlinear Secondary Path
FLNN	Functional Link Neural Network
GFLNN	Generalized Functional Link Neural Network
EFLNN	Exponential Functional Link Neural Network
EMFN	Even Mirror Fourier Nonlinear
LeNN	Legendre Neural Network
VFXLMS	Volterra Filtered-Reference Least Mean Square
VFELMS	Volterra Filtered-Error Least Mean Square
DVMFXLMS	Volterra Filtered-Reference Least Mean Square Based on Decomposable Volterra Model
DVMFELMS	Volterra Filtered-Error Least Mean Square Based on Decomposable Volterra Model
FXGFLNN	Filtered-Reference Functional Link Neural Network
FEGFLNN	Filtered-Error Functional Link Neural Network
FXEMFNL	Filtered-Reference Even Mirror Fourier Nonlinear Filter with a Linear Part
FEEMFNL	Filtered-Error Even Mirror Fourier Nonlinear Filter with a Linear Part
NMSE	Normalized Mean Square Error
SNR	Signal-to-Noise Ratio

References

1. Elliott, S.J. *Signal Processing for Active Control*; Academic: New York, NY, USA, 2001.
2. Hansen, C.; Snyder, S.; Qiu, X.; Brooks, L.; Moreau, D. *Active Control of Noise and Vibration*; CRC Press: Boca Raton, FL, USA, 2012.
3. Kuo, S.M.; Morgan, D.R. Active noise control: A tutorial review. *Proc. IEEE* **1999**, *87*, 943–973. [CrossRef]
4. Cheer, J.; Elliott, S.J. Multichannel control systems for the attenuation of interior road noise in vehicles. *Mech. Syst. Signal Process.* **2015**, *60–61*, 753–769. [CrossRef]
5. 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**, *e3*, 753–769. [CrossRef]
6. Han, R.; Wu, M.; Gong, C.; Jia, S.; Han, T.; Sun, H.; Yang, J. Combination of robust algorithm and head-tracking for a feedforward active headrest. *Appl. Sci.* **2019**, *9*, 1760. [CrossRef]

7. Rafaely, B. Zones of quiet in a broadband diffuse sound field. *J. Acoust. Soc. Am.* **2001**, *110*, 296–302. [CrossRef]
8. Elliott, S.J.; Cheer, J. Modeling local active sound control with remote sensors in spatially random pressure fields. *J. Acoust. Soc. Am.* **2015**, *137*, 1936–1946. [CrossRef] [PubMed]
9. Rafaely, B.; Elliott, S.J. A computationally efficient frequency-domain LMS algorithm with constraints on the adaptive filter. *IEEE Trans. Signal Process.* **2000**, *48*, 1649–1655. [CrossRef]
10. Zhang, G.; Tao, J.; Qiu, X.; Burnett, I. Decentralized two-channel active noise control for single frequency by shaping matrix eigenvalues. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2019**, *27*, 44–52. [CrossRef]
11. Tan, L.; Jiang, J. Adaptive volterra filters for active control of nonlinear noise processes. *IEEE Trans. Signal Process.* **2001**, *49*, 1667–1676. [CrossRef]
12. Zhu, L.; Yang, T.; Pan, J. Design of nonlinear active noise control earmuffs for excessively high noise level. *J. Acoust. Soc. Am.* **2019**, *146*, 1547–1555. [CrossRef] [PubMed]
13. George, N.V.; Panda, G. Advances in active noise control: A survey, with emphasis on recent nonlinear techniques. *Signal Process.* **2013**, *93*, 363–377. [CrossRef]
14. Lu, L.; Yin, K.L.; de Lamare, R.C.; Zheng, Z.; Yu, Y.; Yang, X.; Chen, B. A survey on active noise control in the past decade–part II: Nonlinear systems. *Signal Process.* **2020**, *181*, 10729.
15. Snyder, S.D.; Tanaka, N. Active control of vibration using a neural network. *IEEE Trans. Neural Netw.* **1995**, *6*, 819–828. [CrossRef]
16. Zhou, Y.L.; Zhang, Q.Z.; Li, X.D.; Gan, W.S. Analysis and DSP implementation of an ANC system using a filtered-error neural network. *J. Sound Vib.* **2005**, *285*, 1–25. [CrossRef]
17. Sicuranza, G.L.; Carini, A. Filtered-x affine projection algorithm for multichannel active noise control using second-order volterra filters. *IEEE Signal Process. Lett.* **2004**, *11*, 853–857. [CrossRef]
18. Das, D.P.; Panda, G. Active mitigation of nonlinear noise processes using a novel filtered-s LMS algorithm. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2004**, *12*, 313–322. [CrossRef]
19. Sicuranza, G.; Carini, A. A generalized FLANN filter for nonlinear active noise control. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2011**, *19*, 2412–2417. [CrossRef]
20. Patel, V.; Gandhi, V.; Heda, S.; George, N.V. Design of adaptive exponential functional link network-based nonlinear filters. *IEEE Trans. Circuits Syst. I Reg. Pap.* **2016**, *63*, 1434–1442. [CrossRef]
21. Carini, A.; Sicuranza, G.L. Even mirror fourier nonlinear filters. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; pp. 5608–5612.
22. Carini, A.; Cecchi, S.; Romoli, L.; Sicuranza, G.L. Legendre nonlinear filters. *Signal Process.* **2015**, *109*, 84–94. [CrossRef]
23. Carini, A.; Sicuranza, G.L. A study about chebyshev nonlinear filters. *Signal Process.* **2016**, *122*, 24–32. [CrossRef]
24. Chen, B.; Yu, S.; Yu, Y.; Guo, R. Nonlinear active noise control system based on correlated EMD and chebyshev filter. *Mech. Syst. Signal Process.* **2019**, *130*, 74–86. [CrossRef]
25. George, N.V.; Gonzalez, A. Convex combination of nonlinear adaptive filters for active noise control. *Appl. Acoust.* **2014**, *76*, 157–161. [CrossRef]
26. Zhao, H.; Zeng, X.; He, Z.; Yu, S.; Chen, B. Improved functional link artificial neural network via convex combination for nonlinear active noise control. *Appl. Soft Comput.* **2016**, *42*, 351–359. [CrossRef]
27. Zhou, D.; Debrunner, V. Efficient adaptive nonlinear filters for nonlinear active noise control. *IEEE Trans. Circuits Syst. I Reg. Pap.* **2007**, *54*, 669–681. [CrossRef]
28. Kuo, S.M.; Wu, H.T. Nonlinear adaptive bilinear filters for active noise control systems. *IEEE Trans. Circuits Syst. I Reg. Pap.* **2005**, *52*, 617–624. [CrossRef]
29. Tan, L.; Dong, C.; Du, S. On implementation of adaptive bilinear filters for nonlinear active noise control. *Appl. Acoust.* **2016**, *106*, 122–128. [CrossRef]
30. Zhao, H.; Zeng, X.; He, Z.; Li, T. Adaptive RSOV filter using the felms algorithm for nonlinear active noise control systems. *Mech. Syst. Signal Process.* **2013**, *34*, 378–392. [CrossRef]
31. Carini, A.; Sicuranza, G.L. Recursive even mirror fourier nonlinear filters and simplified structures. *IEEE Trans. Signal Process.* **2014**, *62*, 6534–6544. [CrossRef]
32. Guo, X.; Jiang, J.; Tan, L.; Du, S. Improved adaptive recursive even mirror fourier nonlinear filter for nonlinear active noise control. *Appl. Acoust.* **2018**, *146*, 310–319. [CrossRef]
33. Le, D.C.; Zhang, J.; Pang, Y. A bilinear functional link artificial neural network filter for nonlinear active noise control and its stability condition. *Appl. Acoust.* **2018**, *132*, 19–25. [CrossRef]
34. Guo, X.; Jiang, J.; Chen, J.; Du, S.; Tan, L. Bibo-stable implementation of adaptive function expansion bilinear filter for nonlinear active noise control. *Appl. Acoust.* **2020**, *168*, 107407. [CrossRef]
35. Carini, A.; Sicuranza, G.L. Bibo-stable recursive functional link polynomial filters. *IEEE Trans. Signal Process.* **2017**, *65*, 1595–1606. [CrossRef]
36. Fermo, A.; Carini, A.; Sicuranza, G.L. Low-complexity nonlinear adaptive filters for acoustic echo cancellation in GSM handset receivers. *Eur. Trans. Telecommun.* **2003**, *14*, 161–169. [CrossRef]
37. Paleologu, C.; Benesty, J.; Ciochină, S. Linear system identification based on a kronecker product decomposition. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2018**, *26*, 1793–1808. [CrossRef]

38. Pinheiro, F.C.; Lopes, C.G. A low-complexity nonlinear least mean squares filter based on a decomposable volterra model. *IEEE Trans. Signal Process.* **2019**, *67*, 5463–5478. [CrossRef]
39. Guo, X.; Li, Y.; Jiang, J.; Dong, C.; Du, S.; Tan, L. Sparse modeling of nonlinear secondary path for nonlinear active noise control. *IEEE Trans. Instrum. Meas.* **2018**, *67*, 482–496. [CrossRef]
40. Miyagi, S.; Sakai, H. Performance comparison between the filtered-error LMS and the filtered-x LMS algorithms ANC. In Proceedings of the 2001 IEEE International Symposium on Circuits and Systems, Sydney, NSW, Australia, 6–9 May 2001; pp. 661–664.