



# Low-complexity even mirror fourier adaptive filter for nonlinear active noise control

Dinh Cong Le<sup>a,\*</sup>, Sheng Zhang<sup>b</sup>, Jiashu Zhang<sup>b</sup>

<sup>a</sup> Institute of Engineering and Technology, Vinh University, Viet Nam

<sup>b</sup> Sichuan Province Key Lab of Signal and Information Processing, Southwest Jiaotong University, Chengdu 610031, PR China



## ARTICLE INFO

### Article history:

Received 23 March 2022

Received in revised form 1 June 2022

Accepted 1 July 2022

### Keywords:

Active noise control

Pipelined

Even mirror fourier

Nonlinear filter

## ABSTRACT

The Even Mirror Fourier (EMF) filter is the universal approximation for nonlinear systems, which has been used in active noise control (ANC) applications, especially in scenarios where ANC contains strong nonlinearity. However, the computational burden can impede its implementation in practice. To mitigate this disadvantage, an improved pipelined EMF (IPEMF) filter has been developed and applied for the nonlinear ANC systems, in this paper. The proposed architecture uses simple small-scale EMF modules (i.e., fewer coefficients) nested in a pipelined fashion to replace the complex EMF structure, and hence has got a computational advantage. Moreover, the modules of the IPEMF are designed to update the synaptic weight vector independently, and the output of IPEMF is simplified by the sum of the local estimates of the modules. As a consequence, the nonlinear processing capability is considerably improved and the filtered-error algorithm deduced for the IPEMF controller is less complicated. Simulation results show that the proposed IPEMF controller significantly reduces the computational complexity compared to EMF, while still maintaining control performance is as equivalent as EMF and better than the pipelined EMF (PEMF), pipelined Volterra filter (PVF).

© 2022 Elsevier Ltd. All rights reserved.

## 1. Introduction

The linear ANC systems using the FIR controller or its improvements have been developed in many practical applications [1,2]. However, since these systems do not take into account the nonlinearity that exists in the noise sources or occurs in the primary or/and secondary path, their performance is degraded [3–5]. Many studies on reducing the effect of nonlinearity on the ANC system have been represented. Nonlinear controllers based on neural networks (NN) were soon applied to compensate for nonlinear distortion in the secondary path [6,7]. Extended Volterra series based-controllers have shown to be effective under the assumption that the noise source or/and the primary path contains nonlinear behavior [3–5]. In [8–10], many computationally efficient nonlinear controllers based on bilinear and spline structures have also been developed. Another structure using the functional links artificial neural network (FLANN) for the nonlinear ANC system was first proposed by Das and Panda [11]. In the recent literature, many improvements of the FLANN structure have been studied and applied to the nonlinear ANC [12–20].

As indicated by research [21], the improvements of FLANN-related structures cannot fulfill the requirements of the universal approximation. By exploiting the attractive properties of the multidimensional generalized Fourier series, an EMF linear-in-the-parameters filter was developed by Carini and Sicuranza [21]. The EMF filter can arbitrarily good approximation to a continuous nonlinear function. It has been shown to be more efficient than the Volterra series in nonlinear identification applications [21]. As can be seen in [22–25], many ANC systems using EMF filters as controllers have been developed with significant results. Like the Volterra series-based filter, however, in order to have a good enough model for systems containing strong nonlinearity, the EMF needs to expand its order and memory length. This will lead to an increase in computational complexity and training time.

In order to reduce computational complexity, many systems using the pipelined architecture have been proposed [26–30]. The idea of pipelined architecture is based on the concept of divide and conquer (i.e. a complex system can be broken down into many simple small-scale subsystems and nested in a pipelined parallel manner) [30]. Pipelined-based systems have low computational cost because they only have to deal with small-scale modules concurrently, and maintain their performance thanks to the nested nonlinearity of the modules.

\* Corresponding author.

E-mail address: [congdhv@gmail.com](mailto:congdhv@gmail.com) (D.C. Le).

The computational complexity of the controller using the EMF increases rapidly when the order and/or the memory length goes up, arising in ANC scenarios containing strong nonlinearity. To reduce this disadvantage, a computational efficiency improved pipelined EMF (IPEMF) controller for the nonlinear ANC systems is developed. The design of the proposed IPEMF architecture consists of small-scale EMF modules cascaded in a pipelined fashion. Unlike the pipelined architecture in [26,27], the EMF modules in the IPEMF have different synaptic weight vectors, and global output is the sum of local estimates of each module. As a result, the proposed IPEMF architecture can further improve the performance, and simplify the adaptive algorithm in the ANC system. Besides, to reduce the computational costs of filtering the signal through the secondary path, the filtered-error algorithm has also been adopted for the IPEMF based-ANC system. The stability conditions of the proposed system have also been analyzed and presented. Finally, we conducted many computational simulations with different scenarios and degrees of nonlinearity at the input signal, the primary and secondary path to evaluate the effectiveness of the proposed IPEMF controller.

The paper is presented as follows: Section 2 presents the pipelined EMF (PEMF) controller. In Section 3, we develop the improved structure for the PEMF controller. The stability condition and computational complexity analysis are presented in Sections 4 and 5. Section 6 deals with the simulation experiments. Finally, conclusions are drawn in Section 7.

## 2. Pipelined even mirror fourier (PEMF) filter for ANC system

### 2.1. PEMF structure

Inspired by [26], the PEMF structure is developed and illustrated in Fig. 1. It is composed of small-scale EMF modules and nested in a pipelined fashion. The input of each EMF module includes the external signal and the output of the adjacent module before it. For the  $M$ -th module, the second signal is its own delayed output. The  $i$ -th EMF module structure is illustrated in detail in Fig. 2. The module's output is then fed into a linear FIR filter to extract the information contained in the signal. The PEMF structure is designed so that the modules have the same synaptic weight vectors.

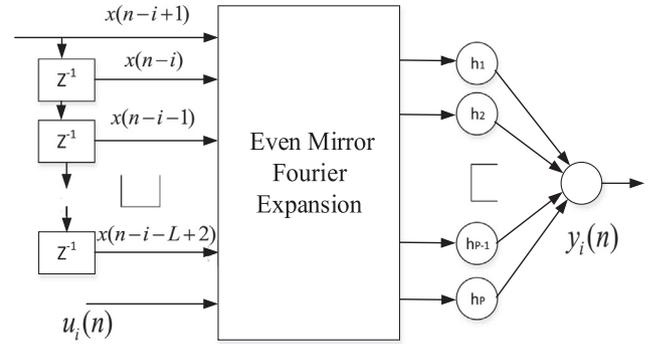


Fig. 2. The detailed structure of the  $i$ -th EMF module.

Let  $X_i(n)$  be the input vector of the  $i$ th module, we can express it as follows.

$$X_i(n) = [x(n-i+1), x(n-i), \dots, x(n-i-L+2), u_i(n)]^T, 1 \leq i \leq M \quad (1)$$

where  $[x(n-i+1), x(n-i), \dots, x(n-i-L+2)]^T$  are the external input vector at  $n$ th time,  $L$  is the input memory length, and  $u_i(n)$  is the second input signal of  $i$ th module,

$$u_i(n) = \begin{cases} y_{i+1}(n-1) & \text{if } i \neq M \\ y_M(n-1) & \text{if } i = M, \end{cases} \quad (2)$$

where  $y_{i+1}(n-1)$  is the delayed output signal of the module  $i+1$ ;  $y_M(n-1)$  is the output that is delayed one unit-time of the  $M$ -th module. And the output  $y_i(n)$  of the  $i$ th module is computed.

$$y_i(n) = H(n)^T XE_i(n) \quad (3)$$

with  $XE_i(n)$  denote the expansion vector of the external input vector  $X_i(n)$  by the third-order EMF function (For the sake of brevity, we hereafter call EMF). And  $H(n)$  is the corresponding synaptic weight vector in each EMF module. As mentioned before, the modules adopt the same synaptic weight vector (i.e.,  $H(n) = H_1(n), \dots, H_M(n)$ ).

The extended signal  $XE_i(n)$  at the  $n$ -th time can be represented as,

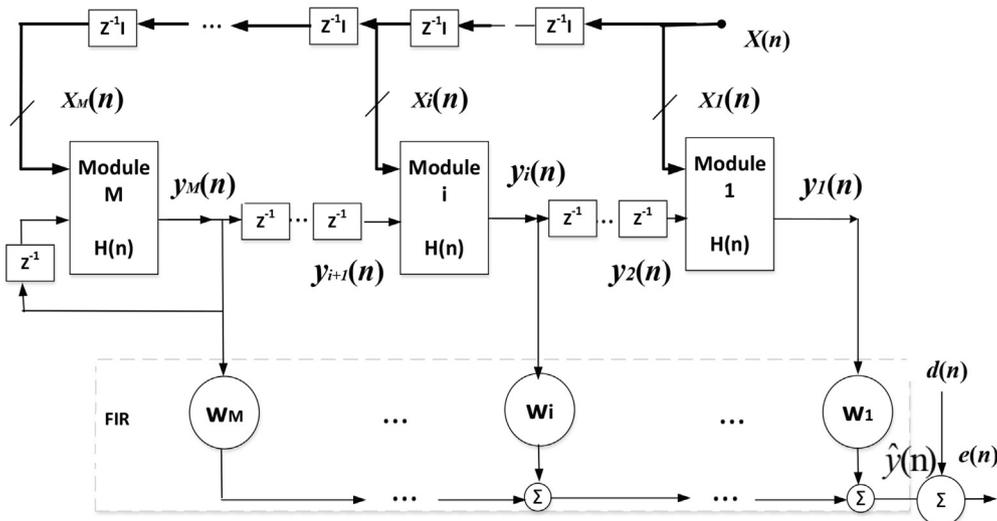


Fig. 1. PEMF structure.

$$\begin{aligned} XE_i(n) &= [XE1_i(n)^T, XE2_i(n)^T, XE3_i(n)^T]^T \\ &= [xe1_i, xe2_i, \dots, xeP_i]^T \end{aligned} \quad (4)$$

where memory length  $P$  of  $XE_i(n)$  is defined as  $P = \frac{(L+3)!}{L!3!}$ , with  $L!$  is factorial  $L$ .

$XE1_i(n)$  is extended signal of the order 0 and order 1 of the EMF, with memory length  $P_1 = L + 2$ ,

$$\begin{aligned} XE1_i(n) &= [1, \sin(0.5\pi x(n-i+1)), \dots, \sin(0.5\pi x(n-i-L+2)), \\ &\quad \sin(0.5\pi u_i(n))]^T \end{aligned} \quad (5)$$

$XE2_i(n) = [XE21_i(n)^T, XE22_i(n)^T]^T$  is the second-order extended signal, with memory length  $P_2 = L + 1 + L(L+1)/2$ ,

$$XE21_i(n) = [\cos(\pi x(n-i+1)), \dots, \cos(\pi x(n-i-L+2)), \cos(\pi u_i(n))]^T \quad (6)$$

For  $m = 0 : L-1; k = m : L-1$ ; and if  $m \neq k$

$$\begin{aligned} XE22_i(n) &= [\sin(0.5\pi x(n-i-m)) \times \sin(0.5\pi x(n-i-k)), \dots, \\ &\quad \sin(0.5\pi x(n-i-m) \times \sin(0.5\pi u_i(n))]^T \end{aligned} \quad (7)$$

$XE3_i(n) = [XE31_i(n)^T, XE321_i(n)^T, XE322_i(n)^T, XE33_i(n)^T]^T$  is the third-order extended signal, with memory length  $P_3 = L + 1 + L(L+1) + L(L-1)(L+1)/6$  and represented as.

$$\begin{aligned} XE31_i(n) &= [\sin(1.5\pi x(n-i+1)), \dots, \sin(1.5\pi x(n-i-L+2)), \\ &\quad \sin(1.5\pi u_i(n))]^T \end{aligned} \quad (8)$$

For  $m = 0 : L-1; k = m : L-1; t = k : L-1$ .

$$\begin{aligned} \text{if } m = k \quad XE321_i(n) &= [\cos(\pi x(n-i-m)) \times \sin(0.5\pi x(n-i-t)), \dots, \\ &\quad \cos(\pi x(n-i-m)) \times \sin(0.5\pi u_i(n))]^T \end{aligned} \quad (9)$$

$$\begin{aligned} \text{elseif } k = t \quad XE322_i(n) &= [\sin(0.5\pi x(n-i-m) \times \cos(\pi x(n-i-k)), \dots, \\ &\quad \sin(0.5\pi x(n-i-m)) \times \cos(\pi u_i(n))]^T \end{aligned} \quad (10)$$

else  $m \neq k \neq t$

$$\begin{aligned} XE33_i(n) &= [\sin(0.5\pi x(n-i-m)) \times \sin(0.5\pi x(n-i-k)) \\ &\quad \times \sin(0.5\pi x(n-i-t)), \dots, \sin(0.5\pi x(n-i-m)) \\ &\quad \times \sin(0.5\pi x(n-i-k)) \times \sin(0.5\pi u_i(n))]^T \end{aligned} \quad (11)$$

Thus, we can represent the weight vector  $H(n)$  consisting of 3 vectors  $H_1(n), H_2(n), H_3(n)$  corresponding to the weights of the expanded signal  $XE1_i(n), XE2_i(n), XE3_i(n)$  as follows.

$$\begin{aligned} H(n) &= [H_1(n)^T, H_2(n)^T, H_3(n)^T]^T \\ &= [h_1(n), h_2(n), \dots, h_{p-1}(n), h_p(n)]^T \end{aligned} \quad (12)$$

and the global output signal of the PEMF is the output of the FIR filter (see Fig. 1),

$$\hat{y}(n) = W(n)^T Y(n) \quad (13)$$

where  $Y(n) = [y_1(n), y_2(n), \dots, y_M(n)]^T$  and  $W(n) = [w_1(n), w_2(n), \dots, w_M(n)]^T$  is the output vector of the modules and weight vector of the linear FIR combiner, respectively.

## 2.2. Adaptive Fx-LMS algorithm for PEMF controller

The goal of the controller is to minimize residual error  $e(n)$ . Thus, we can define the cost function  $\epsilon(n)$  as an instantaneous mean squared error (MSE) as follows.

$$\epsilon(n) = e^2(n) \quad (14)$$

Residual error  $e(n) = d(n) - \hat{d}(n)$  is sensed by a microphone at the quiet location, with  $d(n)$  is the primary path noise and  $\hat{d}(n)$  is the noise-canceling signal generated through the secondary path (see Fig. 3). Similarly [31], to give a general algorithm for both linear and nonlinear secondary path cases, we use a virtual secondary path concept as,

$$\begin{aligned} \tilde{A}(n) &= [a(n, 0), a(n, 1), \dots, a(n, K_s)]^T \\ &= \left[ \frac{\partial \hat{d}(n)}{\partial \hat{y}(n)}, \frac{\partial \hat{d}(n)}{\partial \hat{y}(n-1)}, \dots, \frac{\partial \hat{d}(n)}{\partial \hat{y}(n-K_s)} \right]^T \end{aligned} \quad (15)$$

where  $a(n, 0), a(n, 1), \dots, a(n, K_s)$  are coefficients of the time-varying filter  $\tilde{A}(n)$ , and  $K_s$  denotes the memory length of the  $\hat{d}(n)$ .

To minimize the cost function  $\epsilon(n)$ , the synaptic weight vector  $H(n)$  is adapted according to the algorithm as following,

$$\begin{aligned} H(n+1) &= H(n) - \frac{1}{2} \mu \nabla_{H(n)} \epsilon(n) = H(n) - \frac{1}{2} \mu \frac{\partial \epsilon(n)}{\partial H(n)} \\ &= H(n) - \frac{1}{2} \mu \frac{\partial E(e^2(n))}{\partial H(n)} = H(n) + \mu e(n) \frac{\partial \hat{d}(n)}{\partial H(n)} \end{aligned} \quad (16)$$

Note that,

$$\frac{\partial \hat{d}(n)}{\partial H(n)} = \sum_{p=0}^{K_s} \frac{\partial \hat{d}(n)}{\partial \hat{y}(n-p)} \frac{\partial \hat{y}(n-p)}{\partial H(n)} \quad (17)$$

and assume that the adaptation process is variable slowly, which leads to.

$$\frac{\partial \hat{y}(n-p)}{\partial H(n)} \cong \frac{\partial \hat{y}(n-p)}{\partial H(n-p)} \quad (18)$$

On the other hand,

$$\begin{aligned} \frac{\partial \hat{y}(n-p)}{\partial H(n-p)} &= \frac{\partial \left( \sum_{i=1}^M w_i(n-p) y_i(n-p) \right)}{\partial H(n-p)} \\ &= \frac{\partial \left( \sum_{i=1}^M w_i(n-p) H(n-p)^T XE_i(n-p) \right)}{\partial H(n-p)} \\ &= \sum_{i=1}^M w_i(n-p) XE_i(n-p) \end{aligned} \quad (19)$$

Substituting (19) into (18) and combining (16,17), we have.

$$\begin{aligned} H(n+1) &= H(n) + \mu e(n) \sum_{p=0}^{K_s} a(n, p) \\ &\quad \times \left[ \sum_{i=1}^M w_i(n-p) XE_i(n-p) \right] \end{aligned} \quad (20)$$

Note that the term  $\sum_{p=0}^{K_s} a(n, p) \left[ \sum_{i=1}^M w_i(n-p) XE_i(n-p) \right]$  is the result of filtering the signal  $\left[ \sum_{i=1}^M w_i(n-p) XE_i(n-p) \right]$  through virtual secondary path  $\tilde{A}(n)$ . And if we set,

$$XE_{if}(n) = \left[ \sum_{i=1}^M w_i(n-p) XE_i(n-p) \right] * \tilde{A}(n) \quad (21)$$

where  $*$  is convolutional operation. Thus, the update equation of  $H(n)$  is rewritten as follows,

$$H(n+1) = H(n) + \mu e(n) XE_{if}(n) \quad (22)$$

Similarly, we obtain the update equation of  $W(n)$  as.

$$W(n+1) = W(n) + \rho e(n) Y_f(n) \quad (23)$$

where  $Y_f(n) = Y(n) * \tilde{A}(n)$  is the filtered signal of  $Y(n)$  through the virtual secondary path  $\tilde{A}(n)$ .

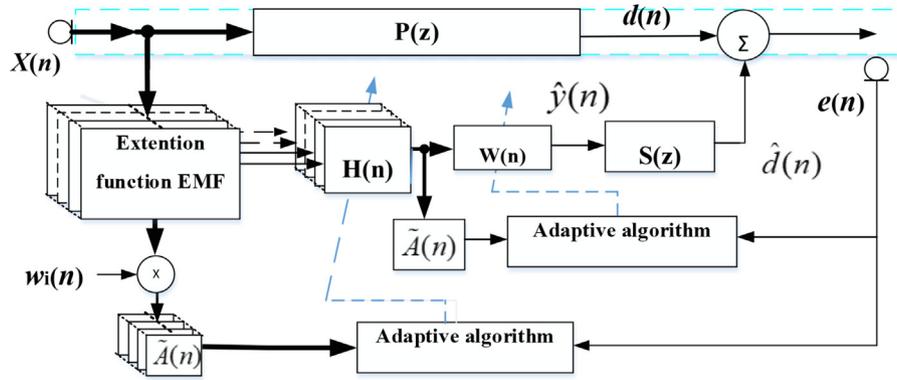


Fig. 3. The ANC system based on PEMF controller.

### 3. Proposed improved PEMF (IPEMF) filter for ANC system

#### 3.1. The IPEMF structure

In the PEMF structure, the EMF modules have the same synaptic weight vector (i.e.,  $H(n) = H_1(n) = H_2(n), \dots, = H_M(n)$ ). This can limit the performance because the input signal in each module is different (different in its delay). In addition, to extract the information containing in the input signal, the outputs of the modules need to be filtered through an adaptive linear FIR combiner. Another drawback as shown in (21), the adaptation algorithm for the ANC system needs to take into account the costs of generating signal  $[\sum_{i=1}^M w_i(n-p)XE_i(n-p)]$  and filtering it via the virtual secondary path filter. These factors can complicate the algorithm and increases the computational cost.

To overcome these disadvantages, the IPEMF structure has been proposed for the ANC system, as shown in Fig. 4. Unlike PEMF, the EMF modules of the proposed IPEMF update the synaptic weight vector independently (i.e.,  $H_1(n) \neq H_2(n), \dots \neq H_M(n)$ ) and the output of the IPEMF is the sum of the estimates of the EMF modules. This can further improve the performance because of the dynamic nature of each module. In the PEMF structure, the error  $e(n)$  needs to be backpropagated through the linear FIR combiner (see Fig. 1) to update the synaptic weight vector  $H(n)$  (i.e., the synaptic weight vector  $H(n)$  use the error which is affected by the weight  $w(n)$  of the FIR combiner to update). In other words, the PEMF uses local

errors to update the synaptic weight vector  $H(n)$ . In contrast, the proposed IPEMF is designed to update the synaptic weight vectors  $H_i(n)$  directly from the global error  $e(n)$ . We can, thus, simply the output estimate of the IPEMF by the sum of the output of the modules.

Thus, the output signal of the proposed IPEMF filter is defined as.

$$\hat{y}(n) = \sum_{i=1}^M y_i(n) = \sum_{i=1}^M H_i(n)^T XE_i(n) \quad (24)$$

#### 3.2. Filtered-error LMS algorithm for IPEMF controller

In order to reduce the computational cost of filtering the  $XE_i(n)$  signal through the  $\hat{d}(n)$ , the FE-LMS algorithm has been applied to the proposed IPEMF structure. Fig. 5 illustrates the ANC system based on the IPEMF architecture using the FE-LMS algorithm.

The coefficient vector  $H_i(n)$  will be adapted according to the algorithm as follows,

$$H_i(n+1) = H_i(n) - \frac{1}{2} \mu \nabla_{H_i(n)} \epsilon(n) \in (n) \quad (25)$$

where  $\nabla_{H_i(n)} \epsilon(n)$  is gradient of  $\epsilon(n)$  with respect to  $H_i(n)$ , and is calculated as,

$$\nabla_{H_i(n)} \epsilon(n) = \frac{\partial \epsilon(n)}{\partial H_i(n)} = \frac{\partial E(e^2(n))}{\partial H_i(n)} \cong -2e(n) \frac{\partial \hat{d}(n)}{\partial H_i(n)} \quad (26)$$

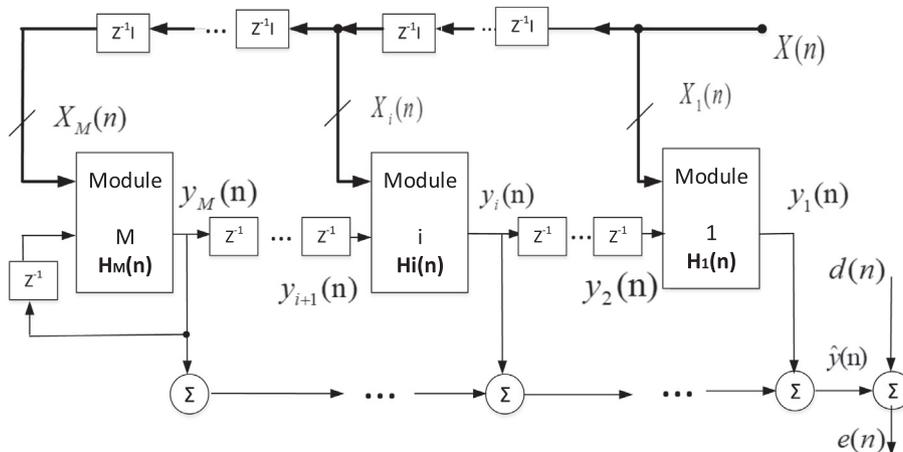


Fig. 4. The Proposed IPEMF structure.

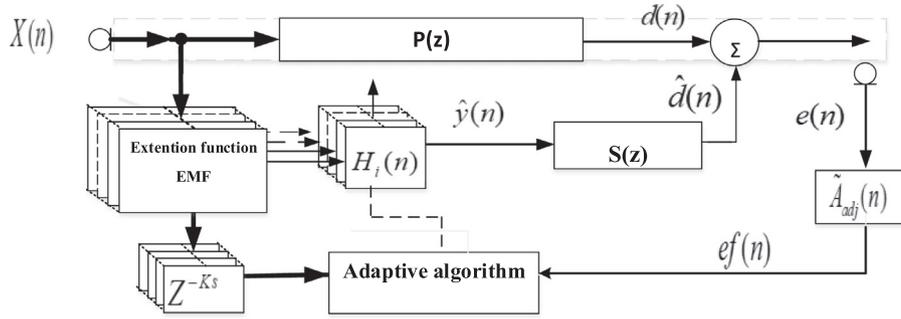


Fig. 5. The IPEMF-based ANC system using the FE-LMS algorithm.

Note that  $\frac{\partial \hat{d}(n)}{\partial H_i(n)} = \sum_{p=0}^{Ks} \frac{\partial \hat{d}(n)}{\partial \hat{y}(n-p)} \frac{\partial \hat{y}(n-p)}{\partial H_i(n)}$ , and assuming the step-size is small, the update equation of  $H_i(n)$  varies slowly, then we can write.

$$\begin{aligned} \frac{\partial \hat{y}(n-p)}{\partial H_i(n)} &\cong \frac{\partial \hat{y}(n-p)}{\partial H_i(n-p)} = \frac{\partial \left( \sum_{i=1}^M H_i(n-p)^T X E_i(n-p) \right)}{\partial H_i(n-p)} \\ &= X E_i(n-p) \end{aligned} \quad (27)$$

Substituting (26), (27) in (25), and combining with (15), we get

$$\begin{aligned} H_i(n+1) &= H_i(n) + \mu e(n) \sum_{p=0}^{Ks} \frac{\partial \hat{d}(n)}{\partial \hat{y}(n-p)} X E_i(n-p) \\ &= H_i(n) + \mu e(n) \sum_{p=0}^{Ks} a(n,p) X E_i(n-p) \end{aligned} \quad (28)$$

where  $a(n,p)$  is the  $p+1$  component of the virtual secondary path  $\tilde{A}(n)$ . Set  $v = n - p + Ks$  such that  $n = v + p - Ks$ , thus the term  $e(n) \sum_{p=0}^{Ks} a(n,p) X E_i(n-p)$  in (28) can be rewritten as.

$$\begin{aligned} e(n) \sum_{p=0}^{Ks} a(n,p) X E_i(n-p) \\ = \left[ \sum_{p=0}^{Ks} e(v+p-Ks) a(v+p-Ks,p) \right] X E_i(v-Ks) \end{aligned} \quad (29)$$

Similarly [31], we define an adjoint virtual secondary path  $\tilde{A}_{adj}(n)$  as.

$$\tilde{A}_{adj}(n) = [a(n, Ks), a(n-1, Ks-1), \dots, a(n-Ks, 0)]^T \quad (30)$$

Note that vector  $\tilde{A}_{adj}(n)$  requires an order inversion of the coefficients vector  $\tilde{A}(n)$  and delays these coefficients over time. Obviously, the term  $\left[ \sum_{p=0}^{Ks} e(v+p-Ks) a(v+p-Ks,p) \right]$  in (29) is the result of filtering the  $e(n)$  through the  $\tilde{A}_{adj}(n)$ . Thus, the filtered error  $ef(n)$  is defined as,

$$\begin{aligned} ef(n) &= \sum_{p=0}^{Ks} e(n+p-Ks) a(n+p-Ks,p) \\ &= \sum_{p=0}^{Ks} e(n-(Ks-p)) a(n-(Ks-p),p) = e(n) * \tilde{A}_{adj}(n) \end{aligned} \quad (31)$$

where  $a(n-(Ks-p),p)$  is the  $p+1$  component of the adjoint virtual secondary path  $\tilde{A}_{adj}(n)$ .

Combining the (31), (29), (28), we deduce,

$$H_i(n+1) = H_i(n) + \mu ef(n) X E_i(n-Ks) \quad i = 1 \dots M \quad (32)$$

where the term  $X E_i(n-Ks)$  is the extended input signal  $X E_i(n)$  that delayed by  $Ks$  samples (with  $Ks$  is the memory length of the  $\hat{d}(n)$ ).

#### 4. Analysis of stability condition

To ensure stability for the proposed IPEMF controller, a bound for the step-size has been analyzed in this section. We know that the residual noise  $e(n)$  at the denoising point can be represented by the Taylor series rule as.

$$e(n+1) = e(n) + \frac{\partial e(n)}{\partial H_i(n)} \Delta H_i(n) + \dots \quad (33)$$

In fact, the effect of higher-order terms on the  $e(n+1)$  is negligible, thus we only need to analyze the first and second-order terms. According to the steepest descent algorithm,  $\Delta H_i(n)$  is the adjusted amount of the synaptic weight vector at time  $n+1$  and  $n$ . From (25) we have,

$$\Delta H_i(n) = H_i(n+1) - H_i(n) = -\frac{1}{2} \mu \nabla_{H_i(n)} \epsilon(n) \quad (34)$$

Substituting the gradient of the  $\epsilon(n)$  with respect to the  $H_i(n)$  calculated in (26), we get,

$$\Delta H_i(n) = \mu e(n) \frac{\partial \hat{d}(n)}{\partial H_i(n)} \quad (35)$$

Note that with the assumption of small step-size, (35) can be represented as,

$$\begin{aligned} \Delta H_i(n) &= \mu e(n) \sum_{p=0}^{Ks} \frac{\partial \hat{d}(n)}{\partial \hat{y}(n-p)} \frac{\partial \hat{y}(n-p)}{\partial H_i(n)} \\ &\cong \mu e(n) \sum_{p=0}^{Ks} \frac{\partial \hat{d}(n)}{\partial \hat{y}(n-p)} \frac{\partial \hat{y}(n-p)}{\partial H_i(n-p)} \\ &= \mu e(n) \sum_{p=0}^{Ks} \frac{\partial \hat{d}(n)}{\partial \hat{y}(n-p)} X E_i(n-p) = \mu e(n) X E_{if}(n) \end{aligned} \quad (36)$$

where  $X E_{if}(n)$  is the filtered version of the  $X E_i(n)$  through the virtual secondary path. Similarly,  $\frac{\partial e(n)}{\partial H_i(n)}$  can be deduced as follows,

$$\begin{aligned} \frac{\partial e(n)}{\partial H_i(n)} &= -\frac{\partial \hat{d}(n)}{\partial H_i(n)} = -\sum_{p=0}^{Ks} \frac{\partial \hat{d}(n)}{\partial \hat{y}(n-p)} \frac{\partial \hat{y}(n-p)}{\partial H_i(n)} \\ &\cong -\sum_{p=0}^{Ks} \frac{\partial \hat{d}(n)}{\partial \hat{y}(n-p)} X E_i(n-p) = -X E_{if}(n) \end{aligned} \quad (37)$$

On the other hand, squaring both sides of (33) we get,

$$e^2(n+1) - e^2(n) = \left[ \frac{\partial e(n)}{\partial H_i(n)} \Delta H_i(n) \right]^2 + 2e(n) \frac{\partial e(n)}{\partial H_i(n)} \Delta H_i(n) \quad (38)$$

It should be noted that the left-hand side of (38) denotes the amount of change of the  $\epsilon(n)$  after an update process, thus we can rewrite (38) as follows,

$$\Delta\epsilon(n) = \epsilon(n+1) - \epsilon(n) = \left[ \frac{\partial e(n)}{\partial H_i(n)} \Delta H_i(n) \right]^2 + 2e(n) \frac{\partial e(n)}{\partial H_i(n)} \Delta H_i(n) \quad (39)$$

Therefore, substituting (36) and (37) into (39) we obtain,

$$\Delta\epsilon(n) = \mu \left[ e(n) XE_{if}(n)^T XE_{if}(n) \right]^2 \left\{ \mu - \frac{2}{XE_{if}(n)^T XE_{if}(n)} \right\} \quad (40)$$

According to the steepest descent algorithm,  $\Delta\epsilon(n) \leq 0$  when the algorithm reaches steady-state. Hence, we can achieve bound for  $\mu$  as follows,

$$0 < \mu \leq \frac{2}{XE_{if}(n)^T XE_{if}(n)} = \frac{2}{\lambda_{max}} \quad (41)$$

where  $\lambda_{max}$  denotes the maximum eigenvalue of the autocorrelation matrix  $\mathfrak{R} = E[XE_{if}(n)^T XE_{if}(n)]$  of the filtered input signal.

### 5. Analysis of computational complexity

To evaluate the computational efficiency, we compare the computational complexity of the ANC systems based on the IPEMF controller and the EMF, PEMF, PVF controllers. Note that PEMF and PVF are 3rd order EMF and Volterra controllers based on the pipelined architecture referenced in [26,27]. Furthermore, for a fair comparison, we assume that all these controllers use the filtered-error LMS algorithm. In this section, we only present the analysis of the computational complexity of the IPEMF-based ANC system. Note that  $K_S$  is the memory length of the secondary path  $\hat{d}(n)$ .  $L$ ,  $L_E$ ,  $L_{PE}$ , and  $L_V$  is the external signal memory length of the IPEMF, EMF, PEMF, and PVF controllers, respectively. Computational requirements of the IPEMF-based ANC system include:

1. Number of multiplication to generate the extended input signal  $XE_i(n)$  includes  $3L(L+1)/2 + L(L-1)(L+1)/6$ , where  $L(L+1)/2$  in (7);  $L(L+1)$  in (9) and (10);  $L(L-1)(L+1)/6$  in (11).
2. The controller output needs  $MP$  multiplications and  $M(P-1)$  additions, where  $M$  is the number of modules and  $P = \frac{(L+1)+3!}{(L+1)!3!}$  is the memory length of the extended signal  $XE_i(n)$ .
3. Updating the synaptic weights of the modules requires  $MP$  multiplication and  $M(P-1)$  addition
4. The number of multiplications and additions required to filter the  $e(n)$  through the  $\hat{d}(n)$  are  $K_S$  and  $K_S-1$ , respectively

Table 1 summarizes the results of the required operations of the controllers. Note that  $P_V = \frac{(L_V+1)+3!}{(L_V+1)!3!}$  is the expanded signal memory length of the PVF;  $P_E = \frac{(L_E+3)!}{L_E!3!}$  is the expanded signal memory length of the EMF;  $P_{PE} = \frac{(L_{PE}+1)+3!}{(L_{PE}+1)!3!}$  is the expanded signal memory length of the PEMF;  $M_V$  and  $M_{PE}$  denote the number of modules in PVF and PEMF structures.

**Table 1**  
The computational complexity of ANC systems.

Controllers	Multiplications	Additions
PVF	$(3 + M_V)P_V + M_V + K_S - L_V - 2$	$(1 + M_V)P_V + M_V + K_S - 4$
PEMF	$(3 + M_{PE})P_{PE} + M_{PE} + K_S - 3L_{PE} - 4$	$(1 + M_{PE})P_{PE} + M_{PE} + K_S - 4$
EMF	$3P_E - 3L_E + K_S - 1$	$2P_E + K_S - 3$
IPEMF	$(2M + 1)P - 3L + K_S - 4$	$2M(P-1) + K_S - 1$

### 6. Computer simulation

In this section, various experiments comparing the noise cancellation performance, computational complexity, processing time of the EMF, PEMF, PVF, and proposed IPEMF controllers are performed in different scenarios of the ANC system. The parameters of the controllers have been selected as follows: the proposed IPEMF (memory length  $L = 4$ , number of modules  $M = 4$ ); EMF (memory length  $L_E = 10$ ); PEMF (memory length  $L_{PE} = 4$ , number of modules  $M_{PE} = 4$ ); PVF (memory length  $L_V = 4$ , number of modules  $M_V = 4$ ). The expansion function of all controllers are chosen as the third-order. To compare the performance of the controllers, we use the mean square error (MSE),

$$MSE = 10 \log_{10} \{ E(e^2(n)) \} \quad (42)$$

where  $E(e^2(n))$  denotes the ensemble average of the  $e^2(n)$ . The learning curve is plotted after 100 independent runs.

#### 6.1. Experiment 1

In this experiment, the primary path is modeled as a strongly nonlinear behavior, and described,

$$d(n) = x(n) + 0.8x(n-1) + 0.3x(n-2) + 0.4x(n-3) - 0.8x(n)x(n-1) + 0.9x(n)x(n-2) + 0.7x(n)x(n-3) - 3.9x^2(n-1)x(n-2) - 2.6x^2(n-1)x(n-3) + 2.1x^2(n-5)x(n-6) \quad (43)$$

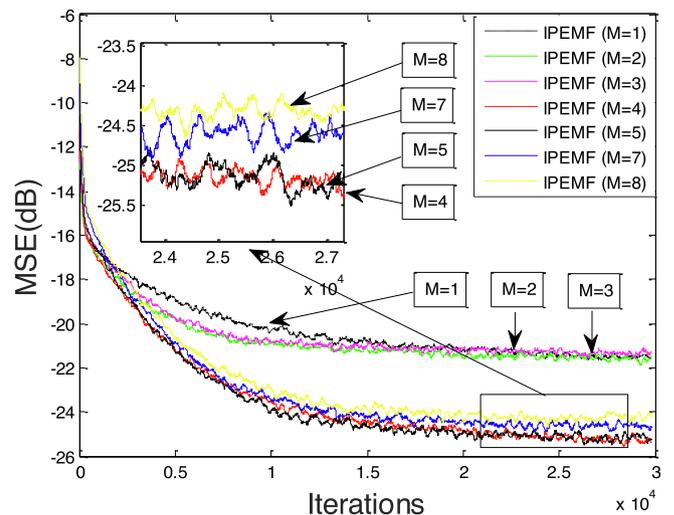
the secondary path is modeled.

$$\hat{d}(n) = \hat{y}(n) + 0.35\hat{y}(n-1) + 0.09\hat{y}(n-2) - 0.5\hat{y}(n)\hat{y}(n-1) + 0.4\hat{y}(n)\hat{y}(n-2) \quad (44)$$

The input is Gaussian noise. The performance and complexity of the proposed IPEMF controller are affected by the choice of module parameter  $M$  and external signal memory length  $L$ . Thus, we first perform two experiments for the selection of suitable parameters of  $L$  and  $M$ .

#### + Module parameter selection

To select the appropriate number of modules, at first we keep the external input signal parameter  $L = 4$  and then change the number of modules  $M$  from 1 to 8. Fig. 6 illustrates the relationship between the MSE performance of the IPEMF controller and the number of modules. Obviously, the noise cancellation performance



**Fig. 6.** The MSE performance versus the number of modules  $M$ .

reaches the best MSE value when  $M$  is equal to 4 or 5. However, to compromise between the performance and the computational complexity, we choose  $M = 4$ .

+ External input parameter selection

To select the appropriate input parameter, we keep the  $M = 4$  and then change the external input parameter  $L$  incrementally from 1 to 7. Fig. 7 shows the relationship between the MSE performance of the IPEMF controller and the external signal parameter. It is easy to see that the IPEMF controller achieves its best performance when choosing the parameter  $L$  is greater than 4. However, in order not to increase the complexity, here we choose the suitable external input parameter of  $L = 4$ .

+ Performance Compare of the EMF, PEMF, PVF and proposed IPEMF controllers

In this section, an MSE performance comparison of the PVF, PEMF, EMF controllers with the IPEMF controller has been shown in Fig. 8. The learning rate of the controllers are chosen as: PVF (linear part  $\mu_L = 0.06$ , nonlinear part (1st order  $\mu_{N1} = 0.03$ ; 2nd order  $\mu_{N2} = 0.03$ , 3rd order  $\mu_{N3} = 0.028$ )); PEMF (linear part  $\mu_L = 0.01$ , nonlinear part (1st order  $\mu_{N1} = 0.05$ ; 2nd order  $\mu_{N2} = 0.05$ , 3rd order  $\mu_{N3} = 0.045$ )); EMF (1st order  $\mu_{E1} = 0.007$ ; 2nd order  $\mu_{E2} = 0.006$ , 3rd order  $\mu_{E3} = 0.006$ ); and IPEMF (1st order  $\mu_1 = 0.0065$ ; 2nd order  $\mu_2 = 0.0045$ , 3rd order  $\mu_3 = 0.003$ );

In addition, the training time and computation complexity of controllers are also shown in Table 2. The processing time is calculated by Matlab R2014 software and the computer configuration is Intel(R) core(TM) i5-4590 CPU 3.3 GHz. The MSE performances are averaged after 20,000 iterations and the controllers are set to the same convergence rate.

From Fig. 8 and Table 2, it is clear that the performance of IPEMF is better than that of PVF, PEMF and approximately equivalent to

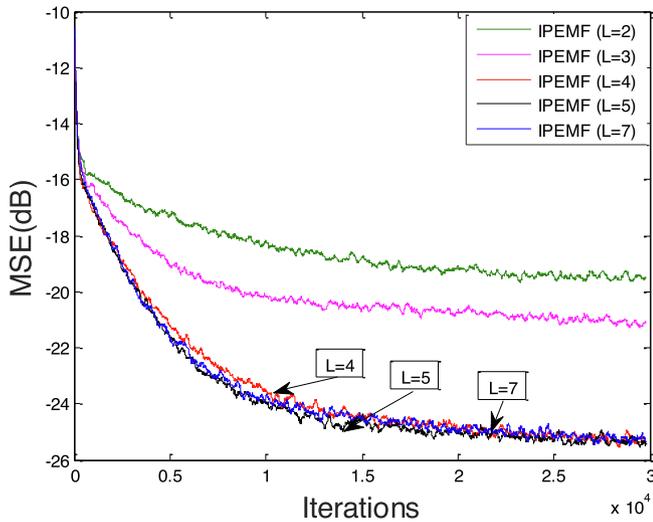


Fig. 7. MSE versus the external input signal  $L$ .

Table 2 Computational complexity, training time of controllers in experiments 1,2,3.

Controllers for ANC system	Computational Complexity		Processing time (s)		
	Mul	Add	Ex 1	Ex 2	Ex 3
PEMF	383	283	23.9168	37.8143	29.9776
PVF	393	283	22.8209	37.4210	30.5346
EMF	812	560	27.1397	47.3518	34.6973
IPEMF	491	442	23.44571	38.9515	32.2481

EMF. Moreover, the computational complexity and training time is significantly reduced in comparison with EMF. The PVF, PEMF, and IPEMF controllers use small-scale EMF and Volterra modules, resulting in less computation cost and processing time compared to EMF controllers.

6.2. Experiment 2

In this experiment, primary path noise is referenced from [25],

$$d(n) = x(n-4) + 0.3x(n-5) + 0.2x(n-6) - 0.5\sin(0.5\pi x(n-3))\sin(0.5\pi x(n-4)) \quad (45)$$

The reference signal is white noise and is normalized in [-0.8 to 0.8], the secondary path is similar to experiment 1. The parameter of the controllers is set: PVF (linear part  $\mu_L = 0.015$ , nonlinear part ( $\mu_{N1} = 0.005$ ;  $\mu_{N2} = 0.005$ ,  $\mu_{N3} = 0.0045$ )); PEMF (linear part  $\mu_L = 0.02$ , nonlinear part ( $\mu_{N1} = 0.1$ ;  $\mu_{N2} = 0.1$ ,  $\mu_{N3} = 0.12$ )); EMF ( $\mu_{E1} = 0.022$ ;  $\mu_{E2} = 0.01$ ,  $\mu_{E3} = 0.0041$ ); and IPEMF ( $\mu_1 = 0.0055$ ;  $\mu_2 = 0.0048$ ,  $\mu_3 = 0.0045$ ); Fig. 9 illustrates the smoothed MSE learning curves of the ANC systems.

Here, we see that the proposed IPEMF and EMF controllers have a better noise cancellation result than PVF and PEMF. And performance of the IPEMF (-36.1213 dB) is slightly degraded compared with that of EMF (-36.4268 dB). However, from Table 2, it is clear that the IPEMF has a significantly reduced computational cost and training time compared to EMF. In addition, it is easy to see that the PEMF controller achieves convergence faster than that of PVF. The reason may be because the EMF expansion function contains  $\sin()$ ,  $\cos()$  nonlinearity. This is an advantage of the EMF extension function over Volterra.

6.3. Experiment 3

In this experiment, the reference noise is assumed to be the logistics chaotic process [3], which is achieved based on the following recursive function,

$$x(n) = \beta x(n-1)[1 - x(n-1)] \quad (46)$$

where  $\beta = 4$ ,  $x(0) = 0.9$ , and is normalized in [-0.5 ÷ 0.5].

The secondary path is the Hammerstein model [15], is represented as follows,

$$q(n) = \tanh(\hat{y}(n)) \quad (47)$$

$$\hat{d}(n) = q(n) + 0.2q(n-1) + 0.05q(n-2)$$

Primary path is similar to the experiment 1. The parameter of controllers are set: PVF (linear part  $\mu_L = 0.06$ , nonlinear part ( $\mu_{N1} = 0.002$ ;  $\mu_{N2} = 0.0018$ ,  $\mu_{N3} = 0.0018$ )); PEMF (linear part  $\mu_L = 0.01$ , nonlinear part ( $\mu_{N1} = 0.012$ ;  $\mu_{N2} = 0.011$ ,  $\mu_{N3} = 0.011$ )); EMF ( $\mu_{E1} = 0.009$ ;  $\mu_{E2} = 0.009$ ,  $\mu_{E3} = 0.0086$ ); and IPEMF ( $\mu_1 = 0.006$ ;  $\mu_2 = 0.0058$ ,  $\mu_3 = 0.0056$ );

The MSE learning curves of the controllers are illustrated in Fig. 10. From Fig. 10 and Table 2, we find that the noise cancellation performance of the IPEMF controller is acceptable in comparison with that of the EMF controller. But, it is obviously more efficient

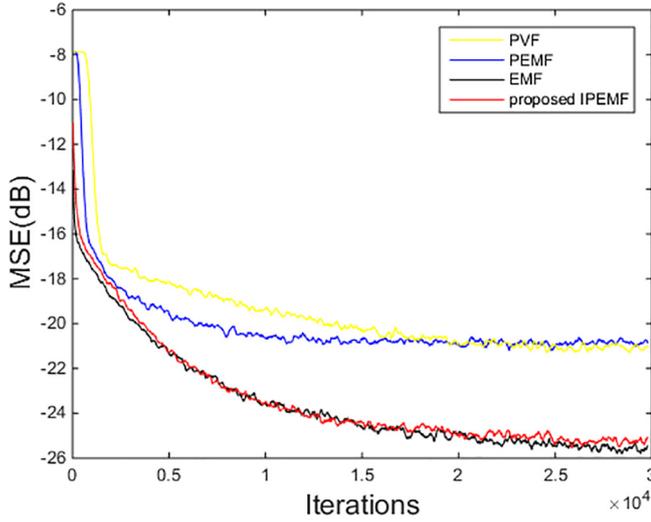


Fig. 8. The MSE performance of different controllers for the scenario in experiment 1.

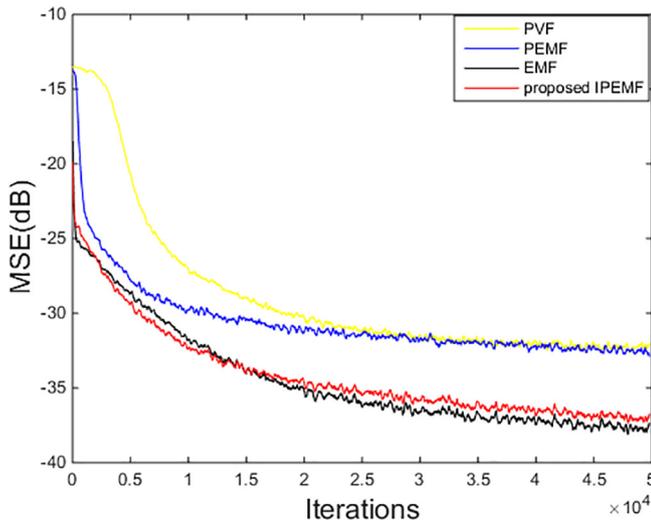


Fig. 9. The MSE performance of ANC systems for the scenario in experiment 2.

in terms of computation and processing time. The PVF and PEMF controllers require less computational complexity and training time than IPEMF. The reason is that they are designed to update the same weight vectors. In systems containing strong nonlinearities, however, this update strategy may not be effective.

#### 6.4. Experiment 4

To illustrate a real experiment, we use measured secondary and primary path which were made by Kou *et al* [1]. Here, the poles and zeros of the P(Z) and S(Z) transfer function are values measured from a real ANC system. Fig. 11a depicts the phase and amplitude response for these measured P(Z) and S(Z).

The reference input signal is three sinewaves that have normalized frequencies of 0.03, 0.06, and 0.08, and are limited to [-0.5 to 0.5]. Furthermore, this reference signal before being fed into the adaptive controller is assumed to be strongly distorted by the following nonlinear model.

$$g[x(n)] = 0.3x(n)x(n-2) + 0.7x(n) \quad (48)$$

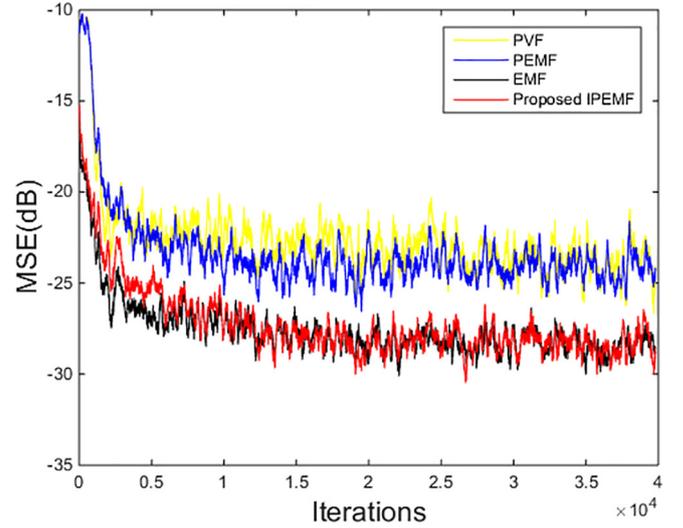


Fig. 10. The MSE performance of different controllers for the scenario in experiment 3.

The learning rate of the controllers in this experiment is chosen as: PVF (linear part  $\mu_L = 0.015$ , nonlinear part ( $\mu_{N1} = 0.0003$ ;  $\mu_{N2} = 0.00027$ ,  $\mu_{N3} = 0.00024$ )); PEMF (linear part  $\mu_L = 0.01$ , nonlinear part ( $\mu_{N1} = 0.00015$ ;  $\mu_{N2} = 0.00012$ ,  $\mu_{N3} = 0.0001$ )); EMF ( $\mu_{E1} = 0.0001$ ;  $\mu_{E2} = 0.00006$ ,  $\mu_{E3} = 0.00007$ ); and IPEMF ( $\mu_1 = 0.0007$ ;  $\mu_2 = 0.00002$ ,  $\mu_3 = 0.00002$ );

Fig. 11b shows the learning curves for the five controllers on a length of 40,000 iterations. It is easy to see that two controllers EMF, and proposed IPEMF reach the same control performance. However, the computational complexity of IPEMF is significantly less than that of EMF (Number of multiplication and addition operations saves 40% and 30% respectively).

#### 6.5. Experiment 5

In practical ANC applications (e.g., for jets turbines, mufflers), the primary noise propagating in a duct is often highly compressed [3]. Therefore, we have to take into account the nonlinearity of the air in the primary path. In this experiment, we choose the primary noise at the canceling point modeled as a third-order polynomial [3], that is.

$$d(n) = u(n-2) + 0.08u^2(n-2) - 0.04u^3(n-2) \quad (49)$$

where  $u(n) = x(n) * f(n)$ ;  $f(n)$  is the impulse response of the transfer function  $F(z) = z^{-3} - 0.3z^{-4} + 0.2z^{-5}$ ,  $x(n)$  denotes the reference noise, which is the sum of a sinusoidal wave of 500 Hz at a sampling rate of 8000 samples/s and a Gaussian noise process of 40 dB SNR.

The secondary path transfer function is chosen as a nonminimum-phase model  $S(z) = z^{-2} + 1.5z^{-2} - z^{-4}$ . The parameters of controllers are set: PVF (linear part  $\mu_L = 0.01$ , nonlinear part ( $\mu_{N1} = 0.02$ ;  $\mu_{N2} = 0.0035$ ,  $\mu_{N3} = 0.0025$ )); PEMF (linear part  $\mu_L = 0.01$ , nonlinear part ( $\mu_{N1} = 0.02$ ;  $\mu_{N2} = 0.003$ ,  $\mu_{N3} = 0.001$ )); EMF ( $\mu_{E1} = 0.002$ ;  $\mu_{E2} = 0.0005$ ,  $\mu_{E3} = 0.00015$ ); and IPEMF ( $\mu_1 = 0.004$ ;  $\mu_2 = 0.0005$ ,  $\mu_3 = 0.0002$ ). Fig. 12 shows a comparative plot obtained by the PVF, PEMF, EMF, and proposed IPEMF controllers. It is clear that the noise cancellation capacity of the proposed IPEMF is better than PEMF and PVF, while the computational load is significantly reduced compared to EMF.

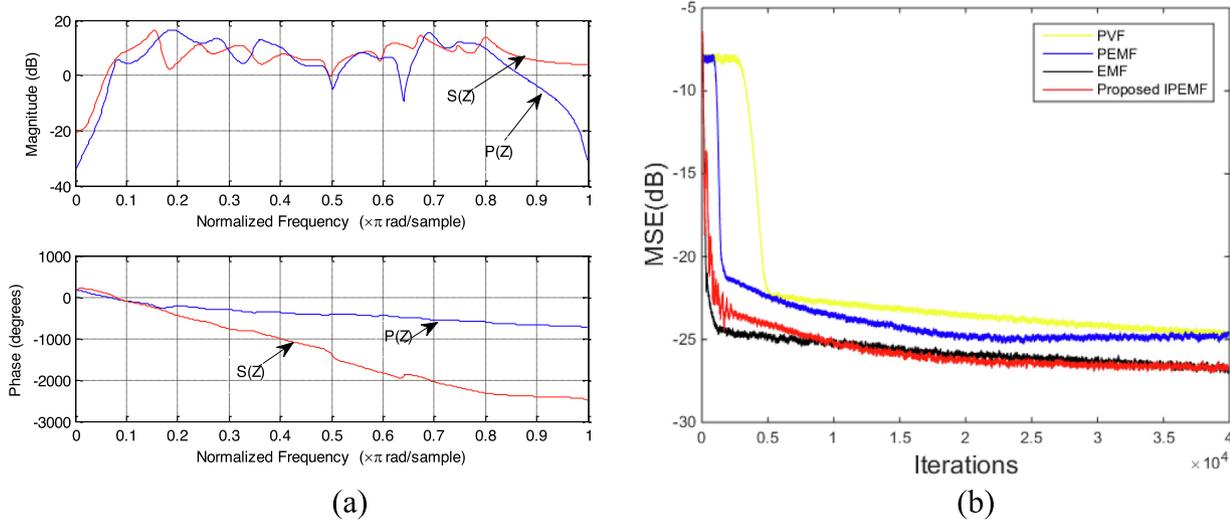


Fig. 11. a) The amplitude and phase response for P(Z) and S(Z), b) The MSE performance of different controllers for the scenario in experiment 4.

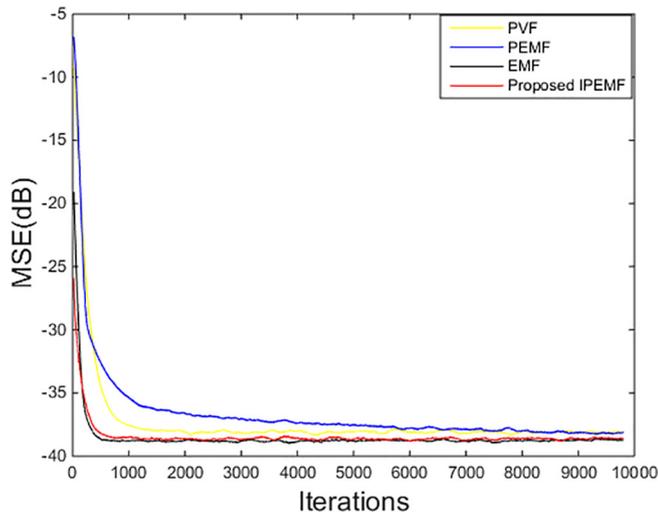


Fig. 12. Comparison of learning curves for PVF, PEMF, EMF and IPEMF controllers.

## 7. Conclusion

In this paper, a computational efficient IPEMF controller has been proposed for nonlinear ANC application. The IPEMF controller consists of small-scale EMF modules cascaded in a pipelined parallel fashion. Research points out that nonlinear processing capability based on pipelined architecture can be significantly improved when updating synaptic weights is independent. In addition, the output of the IPEMF is designed by the sum of the estimates of the modules, which makes the algorithm derived for the ANC system less complex. Experimental results have proved that the proposed IPEMF-based ANC system achieves better noise cancellation performance than conventional PVF, PEMF-based ANC systems and is equivalent to EMF-based ANC systems. Furthermore, it can greatly reduce the computational complexity and training time compared to EMF-based ANC.

## 8. Role of the funding source

This work was partially supported by Vietnam's Ministry of Education and Training under Grant No. B2021-TDV-03.

## CRediT authorship contribution statement

**Dinh Cong Le:** Conceptualization, Data curation, Writing – original draft, Software, Writing – review & editing, Validation. **Jiashu Zhang:** Conceptualization, Methodology, Supervision. **Sheng zhang:** Visualization, Data curation, Software.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

- [1] Kuo SM, Morgan DR. Active noise control systems: algorithms and DSP implementations. New York: Wiley; 1996.
- [2] Elliott SJ, Nelson PA. Active noise control. *IEEE Signal Process* 1993;10(4):12–35.
- [3] Tan L, Jiang J. Adaptive volterra filters for active control of nonlinear noise processes. *IEEE Trans Signal Process* Aug 2001;49(8):1667–76.
- [4] George NV, Panda G. Advances in active noise control: a survey, with emphasis on recent nonlinear techniques. *Signal Process* 2013;93(2):363–77.
- [5] Lu L, Yin KL, Lamare RC, Zheng Z, Yu Y, Yang X, et al. A survey on active noise control in the past decade—Part II: nonlinear systems. *Signal Process* 2021;181:1–16.
- [6] Bouchard M, Pailard B, Dinh CTL. Improved training of neural networks for nonlinear active control of sound and vibration. *IEEE Trans Neural Networks* 1999;10(2):391–401.
- [7] Zhang QZ, Gan WS, Zhou YL. Adaptive recurrent fuzzy neural networks for active noise control. *J Sound Vib* 2006;296(4):935–48.
- [8] Kuo SM, Wu HT. Nonlinear adaptive bilinear filters for active noise control systems. *IEEE Trans Circuits Syst-I: Regular Papers* 2005;52(3):617–24.
- [9] Patel V, George NV. Nonlinear active noise control using spline adaptive filters. *Appl Acoust* 2015;93:38–43.
- [10] Patel V, George NV. Multi-channel spline adaptive filters for non-linear active noise control. *Appl Acoust* 2020;161:107–42.
- [11] Das DP, Panda G. Active mitigation of nonlinear noise processes using a novel filtered-s LMS algorithm. *IEEE Trans Speech Audio Process* 2004;12(3):313–22.
- [12] Zhao HQ, Zeng XP, Zhang JS. Adaptive reduced feedback FLNN filter for active noise control of nonlinear noise processes. *Signal Process* 2010;90(3):834–47.
- [13] Sicuranza GL, Carini A. On the BIBO stability condition of adaptive recursive FLANN filters with application to nonlinear active noise control. *IEEE Trans Speech Audio Process* 2012;20(1):234–45.
- [14] George NV, Gonzalez A. Convex combination of nonlinear adaptive filters for active noise control. *Appl Acoust* 2014;76:157–61.
- [15] Sicuranza GL, Carini A. A generalized FLANN filter for nonlinear active noise control. *IEEE Trans Speech Audio Process* 2011;19(8):2412–7.
- [16] Le DC, 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.

- [17] Luo L, Sun J. A novel bilinear functional link neural network filter for nonlinear active noise control. *Appl Soft Comput* 2018;68:636–50.
- [18] Patel V, Gandhi V, Heda S, George NV. Design of adaptive exponential functional link network-based nonlinear filters. *IEEE Trans Circ Syst I Reg Pap* 2016;63(9):1434–42.
- [19] Le DC, Zhang JS, Li DF, Zhang S. A generalized exponential functional link artificial neural networks filter with channel-reduced diagonal structure for nonlinear active noise control. *Appl Acoust* 2018;139:174–81.
- [20] Zhao S, Zhang L, Lu W. A generalized collaborative functional link adaptive filter for nonlinear active noise control. *Appl Acoust* 2021;175:107799.
- [21] Carini A, Sicuranza GL. Fourier nonlinear filters. *Signal Process* 2014;94(1):183–94.
- [22] Patel V, George NV. Partial update even mirror fourier non-linear filters for active noise control. in *23rd European Signal Processing Conference (EUSIPCO)* 2015: 295-299.
- [23] Guo X, Jiang J, Tan L, Du S. Improved adaptive recursive even mirror fourier nonlinear filter for nonlinear active noise control. *Appl Acoust* 2019;12:310–9.
- [24] Guo X, Dong C, Tan L, Du S. Adaptive even mirror Fourier filtered error LMS algorithm for multichannel nonlinear active noise control. *IEEE Conf Inf Technol Electron Commun* 2016:1–6.
- [25] Guo X, Jiang J, Chen J, Du S, Tan L. Convex Combination Recursive Even Mirror Fourier Nonlinear Filter for Nonlinear Active Noise Control. In: *In 22nd International Conference on Electrical Machines and Systems (ICEMS)*. p. 1–6.
- [26] Zhao HQ, Zhang JS. A novel adaptive nonlinear filter-based pipelined feedforward second order Volterra architecture. *IEEE Trans Signal Process* 2009;57(1):237–46.
- [27] Zhao HQ, Zeng X, Zhang X, He Z, Li T, Jin W. Adaptive extended pipelined second-order Volterra filter for nonlinear active noise controller. *IEEE Trans Speech Audio Process* 2012;20(4):1394–2139.
- [28] Zhang Z, Zhang S, Zhang JS. Pipelined nonlinear spline filter for speech prediction. *Appl Acoust* 2021;179:108057.
- [29] Baltersee J, Chambers JA. Nonlinear adaptive prediction of speech with a pipelined recurrent neural network. *IEEE Trans Signal Process* 1998;46(8):2207–16.
- [30] Haykin S, Li L. Nonlinear adaptive prediction of nonstationary signals. *IEEE Trans Signal Process* 1995;43(2):526–35.
- [31] Zhou D, DeBrunner V. Efficient adaptive nonlinear filters for nonlinear active noise control. *IEEE Trans Circuits Syst-I: Regular* 2007;54(3):669–81.