

Article

A Novel Adaptive State Detector-Based Post-Filtering Active Control Algorithm for Gaussian Noise Environment with Impulsive Interference

Wenzhao Zhu ¹, Shengguo Shi ², Lei Luo ^{1,*} and Jinwei Sun ¹

¹ School of Electrical Engineering and Automation, Harbin Institute of Technology, Harbin 150001, China; wzzhu@hit.edu.cn (W.Z.); jwsun@hit.edu.cn (J.S.)

² School of Underwater Acoustic Engineering, Harbin Engineering University, Harbin 150001, China; shishengguo@hrbeu.edu.cn

* Correspondence: llei@hit.edu.cn

Received: 25 February 2019; Accepted: 17 March 2019; Published: 20 March 2019



Featured Application: This work is proposed for air-conditioner systems in a metro line train cabin. More exactly, the noise produced by rotating machines is the main Gaussian noise. While the train is moving, the friction between wheels and rails provide impulsive noise. The impulsive interference will also influence the active noise control ANC system in air-conditioner systems.

Abstract: For Gaussian noise with random or periodic impulsive interference, the conventional active noise control (ANC) methods with finite second-order moments may fail to converge. Furthermore, the intensity of impulsive noise typically varies over time in the actual application, which also decreases the performance of conventional active impulsive noise control methods. To address these problems, a novel adaptive state detector based post-filtering active control algorithm is proposed. In this work, information entropy with adaptive kernel size is first introduced into the cost function of a post-filtering algorithm to improve its tracking. To enhance the robust performance of adaptive filters when impulsive interference happens, a recursive optimal threshold selecting method is also developed and analyzed by statistical theories. Simulations show that the new method has fast tracking ability in non-impulsive noise environment and keeps robust when impulsive interference happens. It also works well for the impulsive noise of different degrees. Experiment results confirm the effectiveness of the proposed algorithm.

Keywords: active impulsive noise control; post-adaptive filtering; information entropy; state detector

1. Introduction

Active noise control (ANC) is a method of attenuating unwanted noise. Since the development of digital signal processing (DSP) technology and adaptive filter theory, various ANC technologies have been widely used in the industrial field, and the filtered-x least mean square (FxLMS) is the most representative one [1]. However, this algorithm is based on the minimization theory of the mean square error, which may become unstable when exposed to the impulsive interference. Impulsive interference is usually manifested by power line communication noise, atmospheric noise and mechanical noise [2,3]. As impulsive noise is non-Gaussian, the signals with outliers that are produced can be described as symmetric α stable ($S\alpha S$) distributions [4]. More importantly, the process of burst interference cannot be described by limited second order moments.

To solve these problems, many algorithms have been proposed, where the algorithms using off-line α -estimation are applied earliest [5–7]. However, the parameter α is variable in the natural process, which may lead these methods to fail to converge. Therefore, M-estimate functions are

developed for active impulsive noise control in [8–11]. These functions are non-linear transformation methods with different thresholds that are calculated by using prior information. In this circumstance, the methods based on the combination of normalized FxLMS (FxnLMS) algorithm and parameter selection using prior pieces of knowledge have also been proposed [12,13]. Unfortunately, their constant thresholds and parameters may limit the convergence rate of adaptive filters and degrade the performance of the algorithms. For example, in [8], if the absolute value of the n -th iteration of the error signal is less than a certain constant, the M-estimate function restrains the updating rate of the FxLogLMS algorithm in the non-impulsive noise environment. Consequently, the FxlogLMS could only works well in strong impulsive noise environment. To enhance the effectiveness of active impulse noise control using M-estimate techniques, some algorithms based on on-line detection without certain thresholds are proposed for the variable- α impulsive noise environment. In [14], Marco Bergamasco et al. proposed an on-line estimation method for ANC systems to get better attenuation level in variable- α impulsive noise environments. Nevertheless, the replacement of outliers remains the principle strategy of it for overcoming the interference of impulsive noise, which prevents the tracking process of adaptive filters. A recent method proposed by Lu and Zhao using maximum correntropy with adaptive kernel size is demonstrated to perform better than conventional methods on noise attenuation [15]. For this algorithm, the information entropy is used to overcome the influence of impulsive noise. In practical applications, the computational complexity of it is the main barrier to achieve real time noise attenuation.

In fact, these mentioned methods are applied in an absolutely impulsive noise environment, which neglects the fact that most ANC systems are working in Gaussian noise with variable- α impulsive interference environment. To solve these mentioned problems, in this paper, a novel adaptive state detector based post-filtering active control algorithm is proposed. The proposed algorithm uses information entropy method to change the updating of the weight vector, which enhances the tracking abilities compared with conventional method in [13]. By applying the improved version of the optimal detection method of image processing in [16,17], the proposed method is also able to select parameters to achieve robust or fast-tracking automatically for impulsive interference and Gaussian noise environment. Furthermore, the computational load of the detection method is easy to control because its complexities are determined independently and not related to the length of the weight vector.

The rest of this paper is organized as follows. In Section 2, a standard model of active impulsive noise control system is briefly reviewed, and the proposed post-filtering algorithm with M-estimate function is also discussed. Moreover, the new state detector is also introduced and analyzed by statistical analysis in this section. In Sections 3 and 4, simulation and experiments are conducted to evaluate the performance of the proposed method, respectively. Finally, conclusions and discussions are presented in Section 5.

2. Proposed Algorithms

2.1. Preliminary

The best-known single channel or one-dimensional ANC system uses a secondary signal to attenuate the primary noise in a duct. In this paper, this is the fundamental model to test the performance of the proposed method. The diagram of the standard ANC system is shown in Figure 1, where $P(z)$ denotes the primary path from the reference signal $x(n)$ to the error sensor $e(n)$. $S(z)$ denotes the secondary path from the secondary source to the error sensor. $\hat{S}(z)$ is the estimate one of $S(z)$, which can be estimated using either on-line or off-line techniques [1]. $\Theta(n)$ is the weight vector of the adaptive filter and $y(n)$ is the output of adaptive filter.

Based on this structure, several methods of M-estimate have been introduced into active impulsive noise control [6–13]. However, the main problem of applications of these methods is that it cannot works well in the variable- α impulsive noise environment, because these methods use a nonlinear

transform function with a certain threshold. Especially for the post normalized filtered-x least mean square (PFxNLMS) algorithm in [13], it must select a proper value of ρ ($-1 < \rho < 1$) for a certain environment.

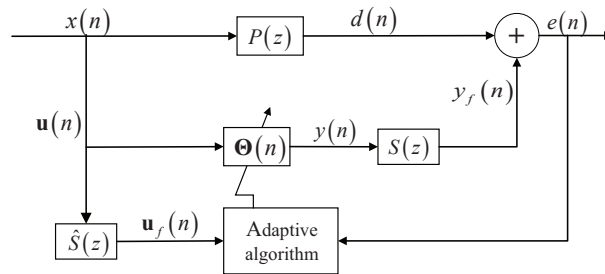


Figure 1. Block diagram of the feed-forward ANC systems in a duct.

2.2. PFxNLMS with M-Estimate Function

For automation selection of an appropriate value of ρ to enhance the tracking of conventional method in [13], PFxNLMS with M-estimate function (MPFxNLMS) algorithm is proposed in this section. A Gaussian kernel of information entropy criterion is used in PFxNLMS to act as the new M-function. Referring to [15], the adaptive Gaussian kernel of information entropy criterion is defined as

$$\kappa(n) = \exp\left(-\frac{e^2(n)}{2\sigma^2(n)}\right). \tag{1}$$

Apparently, the Gaussian kernel of information entropy criterion is an M-estimate function with adaptive adjustment threshold.

To calculate $\sigma(n)$ recursively, the on-line power is estimated as

$$\sigma^2(n+1) = \sigma^2(n) + \frac{1}{N_q}[e(n)]^2 - \frac{1}{N_q}[e(n-N_q+1)]^2, \tag{2}$$

where N_q is used to represent the length of sliding window for $\sigma(n)$ estimation.

Thus, the cost function of the proposed method could be written as

$$\min_{\Theta(n+1)} \sum_{k=1}^{\infty} (\varepsilon(n))^{k-1} \|\Theta(n+1) - \Theta(n+1-k)\|^2, \tag{3}$$

subject to $d(n) + \mathbf{u}_f^T \Theta(n+1) = 0$

where $\varepsilon(n)$ is the forgetting coefficient defined as

$$\varepsilon(n) = \kappa(n) - 1. \tag{4}$$

Because the forgetting coefficient is variable rather than constant, the convergence rate may change in different iterations.

By applying Lagrange multiplier which is denoted as λ , the minimization criterion in Equation (3) can be rewritten as follows

$$J = \sum_{k=1}^{\infty} (\varepsilon(n))^{k-1} \|\Theta(n+1) - \Theta(n+1-k)\|^2 + \lambda \left(d(n) + \mathbf{u}_f^T \Theta(n+1) \right). \tag{5}$$

Using the least mean square (LMS) algorithm, it follows from Equation (5) that the weight vector $\Theta(n)$ is calculated recursively as

$$\Theta(n+1) = \frac{1}{\sum_{k=1}^{\infty} (\varepsilon(n))^{k-1}} \sum_{k=1}^{\infty} (\varepsilon(n))^{k-1} \Theta(n+1-k) - \frac{1}{\sum_{k=1}^{\infty} (\varepsilon(n))^{k-1}} \lambda \mathbf{u}_f(n). \tag{6}$$

According to the theorem of progression in mathematics that $1/\sum_{k=1}^{\infty} (\varepsilon(n))^{k-1} = 1 - \varepsilon(n)$, Equation (6) is rewritten as

$$\Theta(n+1) = (1 - \varepsilon(n)) \left[\sum_{k=1}^{\infty} (\varepsilon(n))^{k-1} \Theta(n+1-k) \right] - (1 - \varepsilon(n)) \lambda \mathbf{u}_f(n). \tag{7}$$

Considering the limit condition in Equation (3), the λ could be calculated as

$$\lambda = \frac{d(n) + (1 - \varepsilon(n)) \mathbf{u}_f^T \sum_{k=1}^{\infty} (\varepsilon(n))^{k-1} \Theta(n+1-k)}{(1 - \varepsilon(n)) \mathbf{u}_f^T(n) \mathbf{u}_f(n)}. \tag{8}$$

To simplify, we define

$$\tilde{\Theta}(n) = (1 - \varepsilon(n)) \sum_{k=1}^{\infty} (\varepsilon(n))^{k-1} \Theta(n+1-k). \tag{9}$$

Substituting Equations (8) and (9) into Equation (6) to yields

$$\Theta(n+1) = \tilde{\Theta}(n) - \frac{\mathbf{u}_f(n)}{\mathbf{u}_f^T(n) \mathbf{u}_f(n)} e(n), \tag{10}$$

where the error signal is given by

$$e(n) = d(n) + \mathbf{u}_f^T(n) \tilde{\Theta}(n). \tag{11}$$

Using the small step-size assumption $\varepsilon(n) \approx \varepsilon(n-1)$, Equation (9) can be calculated recursively, expressed as

$$\tilde{\Theta}(n) = (1 - \varepsilon(n)) \Theta(n) + \varepsilon(n) (1 - \varepsilon(n-1)) \sum_{k=1}^{\infty} (\varepsilon(n-1))^{k-1} \Theta(n-k). \tag{12}$$

Thus, the $\tilde{\Theta}(n)$ can be calculated recursively as

$$\tilde{\Theta}(n) = (1 - \varepsilon(n)) \Theta(n) + \varepsilon(n) \tilde{\Theta}(n-1). \tag{13}$$

The MPF_xNLMS algorithm discussed in this section adjusts the convergence rate more quickly than the other conventional normalized least mean square (NLMS)-based algorithms, therefore, it is defined as fast tracking mode. In order to better adapt to the noise environment, the adaptive switching between fast tracking mode and robust mode can be achieved by using the recursive detection method in next section.

2.3. Novel Recursive State Detector

In the MPF_xNLMS algorithm, the detection factor $\varepsilon(n)$ of is bounded by $-1 < \varepsilon(n) \leq 0$. For a constant detection factor, the mismatch may have arisen in the output signals of the adaptive filter if the input signal changed drastically. Therefore, the forgetting factor should be automatically adjusted

to ensure that the adaptive algorithm works in robust modes under severe noise conditions [16]. To solve this problem, an on-line detection method is combined with the MPF_xNLMS algorithm to enhance the robust performance, called the detector based M-estimate function post normalized filtered-x least mean square (DMPF_xNLMS) algorithm, where “D” represents the new state detector. The block diagram of the DMPF_xNLMS algorithm is shown in Figure 2.

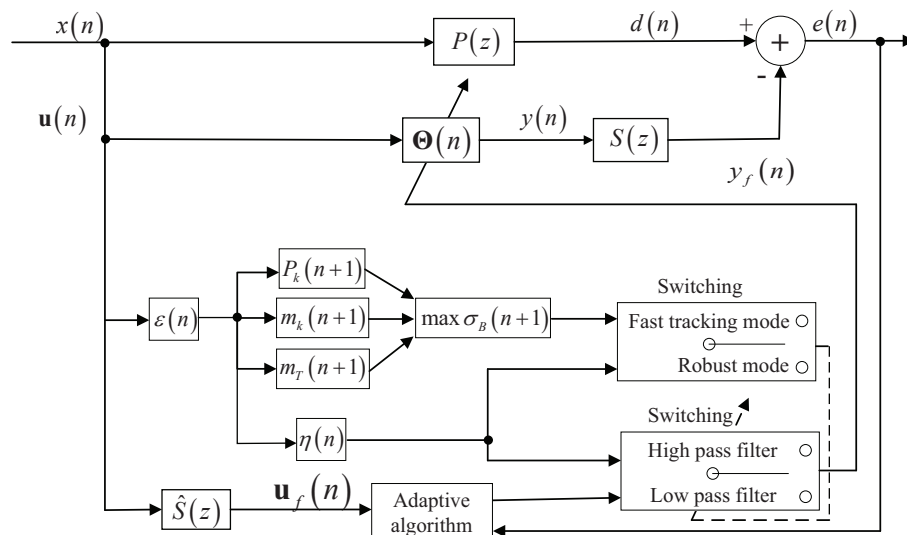


Figure 2. Block diagram of the proposed DMPF_xNLMS algorithm.

In the DMPF_xNLMS, the forgetting coefficient in Equation (3) is redefined as

$$\varepsilon(n) = \begin{cases} \kappa(n) - 1, & \text{if fast tracking mode} \\ 1 - \kappa(n), & \text{if robust mode} \end{cases}, \quad (14)$$

where $\varepsilon(n) < 0$ is defined as the **fast tracking mode** and $0 \leq \varepsilon(n) < 1$ is defined as the **robust mode**. Inspired by [17] combining the statistical theories and the two-class division method to select an interested target for two-dimensional digital graphics, we developed a novel recursive impulsive noise detection method for the ANC system. The detection factor is calculated as follows

$$\eta(n) = \frac{k[1 - \kappa(n)]}{L}, \text{ if } \frac{k}{L} \leq \eta(n) < \frac{k+1}{L} \quad (k = 0, 1, \dots, L - 1), \quad (15)$$

where $\eta(n)$ is divided into L levels $[1, 2, \dots, L]$. A sliding window of length N_w is used to restore the quantization value of $\eta(n)$. To illustrate the adaptive detection process more briefly, define

$$\Phi_1 = \eta(n) \quad (16)$$

$$\Phi_2 = \eta(n - N_w + 1) \quad (17)$$

where Φ_1 represents the input value of detection factor, and Φ_2 represents the output value of detection factor. Based on [17], the optimal threshold is calculated as follows. Assuming that the values of L levels of $\eta(n)$ can be divided into two groups as non-impulsive and impulsive noise, the number of amplitudes at level i is denoted by n_i , and the total number of amplitudes is denoted by $N_w = n_1 + n_2 + \dots + n_L$. The probability at the i -th level is normalized and expressed by

$$p_i = \frac{n_i}{N_w} \quad (18)$$

and

$$p_i \geq 0, \sum_{i=1}^L p_i = 1. \tag{19}$$

On the basis of the different impulsiveness of amplitudes, the data is split into two classes C_0 and C_1 with a threshold at level k . C_0 represents amplitudes with levels $[1, \dots, k]$, which denotes the non-impulsiveness class. The parameter C_1 represents amplitudes with levels $[k + 1, \dots, L]$, which denotes the impulsiveness class. Then the probabilities of occurrence of a class and its mean level are given by Equations (20)–(23):

$$P_{0,k} = \Pr(C_0) = \sum_{i=1}^k p_i = P_k \tag{20}$$

$$P_{1,k} = \Pr(C_1) = \sum_{i=k+1}^L p_i = 1 - P_k \tag{21}$$

and

$$m_{0,k} = \sum_{i=1}^k i \Pr(i | C_0) = \sum_{i=1}^k \frac{ip_i}{P_{0,k}} = \frac{m_k}{P_k} \tag{22}$$

$$m_{1,k} = \sum_{i=k+1}^L i \Pr(i | C_1) = \sum_{i=k+1}^L \frac{ip_i}{P_{1,k}} = \frac{m_T - m_k}{1 - P_k}, \tag{23}$$

where

$$P_k = \sum_{i=1}^k p_i \tag{24}$$

and

$$m_k = \sum_{i=1}^k ip_i \tag{25}$$

are the cumulative probability and the expectation of the histogram up to the k -th level, respectively. Define

$$m_T = m_{k=L} = \sum_{i=1}^L ip_i \tag{26}$$

as the total mean value of the full data in the sliding window.

Equations (24)–(26) can also be expressed recursively in the following process. The input updating expressions are derived using Equations (27) and (28):

$$P_{ki} = \begin{cases} P_k + \frac{1}{N_w}, & \text{if } \left(\frac{k}{L} - \Phi_1\right) \leq 0 \\ P_k, & \text{if } \left(\frac{k}{L} - \Phi_1\right) > 0 \end{cases} \tag{27}$$

$$m_{ki} = \begin{cases} m_k + \frac{\Phi_1}{N_w}, & \text{if } \left(\frac{k}{L} - \Phi_1\right) \geq 0 \\ m_k, & \text{if } \left(\frac{k}{L} - \Phi_1\right) < 0 \end{cases}. \tag{28}$$

The output updating equations are given by Equations (29)–(31):

$$P_k = \begin{cases} P_{ki} - \frac{1}{N_w}, & \text{if } \left(\frac{k}{L} - \Phi_2\right) \leq 0 \\ P_{ki}, & \text{if } \left(\frac{k}{L} - \Phi_2\right) > 0 \end{cases} \tag{29}$$

$$m_k = \begin{cases} m_{ki} - \frac{\Phi_2}{N_w}, & \text{if } \left(\frac{k}{L} - \Phi_2\right) \geq 0 \\ m_{ki}, & \text{if } \left(\frac{k}{L} - \Phi_2\right) < 0 \end{cases}. \tag{30}$$

The optimal threshold of the k -th level is selected through a sequential search, using

$$\sigma_{B,k}^2 = \frac{(m_T m_k - P_k)^2}{m_k (1 - m_k)} \tag{31}$$

and the optimal threshold is derived as

$$\sigma_{B,*}^2 = \max_{1 \leq k < L} \sigma_{B,k}^2. \tag{32}$$

To change the states of adaptive filters between fast tracking mode and robust mode, in addition to the initiate impulsive threshold $\sigma_{B,*}^2$, another threshold γ_s should also be introduced, given by

$$\gamma_s = \tan(\theta(n)) \sigma_\eta(n) \tag{33}$$

where the average angle is updating as

$$\theta(n+1) = \zeta \theta(n) + (1 - \zeta) \theta_e(n) \tag{34}$$

where $\zeta = 0.999$, and $\theta_e(n)$ is the feedback angle error for average threshold calculation, expressed as

$$\theta_e(n) = \begin{cases} \arctan\left(\sqrt{\frac{\eta^2(n)}{\sigma_\eta^2(n)}}\right) & \text{if } \Phi_1 \geq \sigma_{B,*}^2 \\ 0 & \text{else} \end{cases} \tag{35}$$

According to [15], the average power of input signal σ_η^2 is updating as

$$\sigma_\eta^2(n+1) = \sigma_\eta^2(n) + \frac{1}{N_w} [\eta(n)]^2 - \frac{1}{N_w} [\eta(n - N_w + 1)]^2 \tag{36}$$

From the analysis above, the fast tracking mode and robust mode are selected as

$$\begin{cases} \text{if } \Phi_1 \geq \max[\sigma_{B,*}^2, \gamma_s] & : \text{robust mode, } \varepsilon(n) = 1 - \kappa(n) \\ \text{else} & : \text{fast tracking mode, } \varepsilon(n) = \kappa(n) - 1 \end{cases} \tag{37}$$

In fast tracking mode ($\varepsilon(n) < 0$), the tracking performance of the proposed method could be enhanced, because the new weight vector works as a high-pass filter which enlarges the changing of weight vector of NLMS. In robust mode ($0 \leq \varepsilon(n) < 1$), the value of the input signal rises beyond the threshold in Equation (37) and the new weight vector acts like a low-pass filter, limiting heavy fluctuation of coefficients of weight vector. It means that if the burst extents of noise become unacceptable, the robust mode can restrain the updating process of the adaptive filter, stabilizing the performance.

2.4. Statistical Analysis of the Proposed Recursive Detection Method

To give further explanation of the recursive state detector in Section 2.3, the probability problem analysis is shown in this section. To explain the detection problem briefly, we use the binary detection mode to describe it. Assuming that the impulsive signal $\tan(\theta) s(n)$ follows the Gaussian distribution with mean $a_1 = E[\tan(\theta) s(n) + \omega(n)]$ and variance σ_η^2 . For the smaller amplitude Gaussian noise $\omega(n)$, its mean value is $a_0 = E[\omega(n)]$ and variance is also σ_η^2 . The hypotheses H_0 means that is no impulsive noise exists, and H_1 for impulsive noise to occur. The data length is N . The binary detection problem is expressed as

$$\left. \begin{aligned} H_0 : x(n) &= \omega(n) \\ H_1 : x(n) &= \tan(\theta) s(n) + \omega(n) \end{aligned} \right\} \forall n \in [1, 2, \dots, N] \tag{38}$$

Define P_{FA} is false alarm probability as

$$P_{FA} = P(T(x) > \gamma_s; H_0) \tag{39}$$

and define P_D is the detection probability as

$$P_D = P(T(x) > \gamma_s; H_1). \tag{40}$$

To minimize P_{FA} and maximize P_D , using the method in [16], the statistic threshold for the impulsive noise detection is set as

$$T(x) = \frac{1}{N_w} \sqrt{\sum_{n=1}^{N_w} [x(n)]^2 - \sum_{n=1}^{N_w} [x(n) - \tan(\theta) s(n)]^2}. \tag{41}$$

But to minimize the computational load, the threshold in Equation (41) is calculated recursively as mentioned in [15]. It means the threshold $\sigma_{B,*}^2$ is the same one as $T(x)$ in theory.

For the second threshold γ_s , it is calculated using the maximum the posteriori estimation of the Bayes method. Based on the assumption that $\tan(\theta) s(n)$ and $\omega(n)$ follows the Gaussian distribution, the probability distribution function (PDF) is shown as follows

$$p(x|H_k) = (2\pi\sigma_{\eta}^2)^{-N_w/2} \exp\left(-\sum_{i=1}^{N_w} \frac{(x_i - a_k)^2}{2\sigma_{\eta}^2}\right) \quad (k = 0, 1). \tag{42}$$

Therefore, the likelihood ratio function is written as

$$\Lambda(x) = \frac{p(x|H_1)}{p(x|H_0)} = \exp\left[\frac{(a_1 - a_0)}{\sigma_{\eta}^2} \sum_{i=1}^{N_w} x_i - \frac{N(a_1^2 - a_0^2)}{2\sigma_{\eta}^2}\right] \tag{43}$$

and the posterior probability threshold is given by

$$\gamma_s = \frac{(a_1 + a_0)}{2} + \frac{\sigma^2 \ln \Lambda_0}{N_w (a_1 - a_0)}, \tag{44}$$

where the Λ_0 is the optimal likelihood ratio threshold.

It is clear that if the data length N is long enough, then $\gamma_s \rightarrow (a_1 + a_0) / 2$, which means that the P_{FA} is decreasing as the increasing of length N_w . To calculate the γ_s recursively, Equations (33)–(36) are used in the proposed DMPF×NLMS algorithm.

3. Simulations

Using MATLAB, the performance of DMPF×NLMS algorithm was compared with that of the other standard ANC methods. The sampling frequency was set to 8000 Hz. The primary path $P(z)$ and the secondary path $S(z)$ were modelled as FIR filters with lengths of 256 and 128, respectively. To avoid the influence of a secondary path, the estimated secondary path $\hat{S}(z)$ was assumed to be identified as $S(z)$ in simulation work. The frequency response of the primary path and secondary path of a real duct are shown in Figure 3. The weight vector $\Theta(n)$ was modelled as a FIR filter with 256 taps. Referring to [13], the averaged noise reduction (ANR) was used to compare the performance, defined as

$$ANR(n) = 20 \log\left(\frac{A_e(n)}{A_d(n)}\right), \tag{45}$$

where

$$\begin{aligned} A_e(n) &= \zeta A_e(n-1) + (1-\zeta) |e(n)| \\ A_d(n) &= \zeta A_d(n-1) + (1-\zeta) |d(n)| \end{aligned} \tag{46}$$

The brief operation of DMPF_xNLMS is shown in Table 1, its computational complexity for one iteration is summarized in Table 2. In Table 2, FxLMS, FxlogLMS [8], PF_xNLMS [13] and FxRMC [15] are used as the references to evaluate the computational efficient of DMPF_xNLMS. The parameter setting of the mentioned algorithms are shown in Table 3.

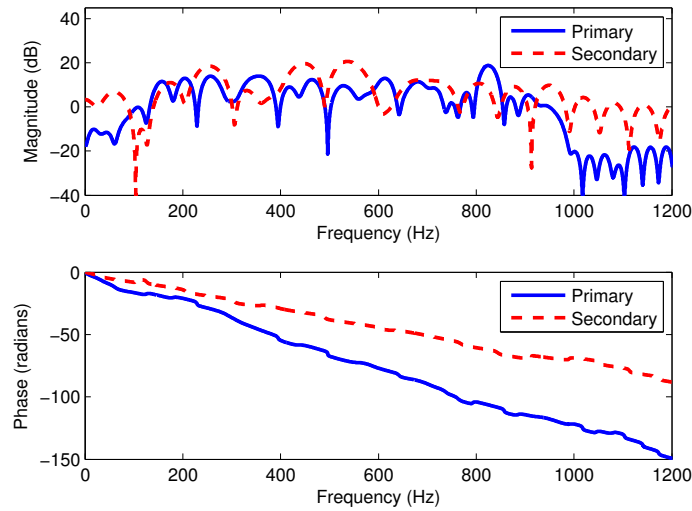


Figure 3. Frequency responses of the primary path and secondary path for a real duct ANC system.

Table 1. Operations of the proposed DMPF_xNLMS method.

Initialization and parameter selection	
Length of primary path $P(z)$, length of secondary path $S(z)$.	
Step size of proposed method: μ	
Length of weightvector M	Initiate angle θ
Length of sliding windows: N_q, N_w	
Length of quantity levels: L	
Brief operations	
While $\{\mathbf{u}(n), e(n)\}$ available do	
1. $y_f(n) = [\tilde{\Theta}^T(n) \mathbf{u}(n)] * s(n)$	
2. $\mathbf{u}_f(n) = \mathbf{u}(n) * s(n)$	
3. $e(n) = d(n) - y_f(n)$	
4. $\kappa(n) = \exp\left(-\frac{e^2(n)}{2\sigma^2}\right)$	
5. $\sigma^2(n+1) = \sigma^2(n) + \frac{1}{N_q}[e(n)]^2 - \frac{1}{N_q}[e(n-N_q+1)]^2$	
6. $\eta(n), \Phi_1 = \eta(n), \Phi_2 = \eta(n-N_w+1)$, compare Φ_1 with $\max[\sigma_{B*}^2(n+1), \gamma_s(n+1)]$	
7. $\varepsilon(n) = \begin{cases} \kappa(n) - 1, & \text{if Fast tracking} \\ 1 - \kappa(n), & \text{if Robust} \end{cases}$	
8. $\tilde{\Theta}(n) = (1 - \varepsilon(n)) \Theta(n) + \varepsilon(n) \tilde{\Theta}(n-1)$	
9. $\Theta(n+1) = \tilde{\Theta}(n) - \frac{\mathbf{u}_f(n)}{\mathbf{u}_f^T(n)\mathbf{u}_f(n)} e(n)$	
10. Updating $P_k(n), m_k(n), P_k(n+1), m_k(n+1)$	
11. Updating $\sigma_{B*}^2(n+1), \gamma_s(n+1)$	
end While	

Table 2. Computational complexity of the mentioned algorithms in one iteration. ($M = 256, L = 10$).

Algorithms	Additions/Subtractions	Multiplications	Divisions	Exponential Operations
FxLMS	$2M(512)$	$2M + 1(513)$	No	No
FxlogLMS	$2M(512)$	$2M + 2(514)$	1	1
PF _x NLMS	$4M(1024)$	$5M + 1(1281)$	1	No
FxRMC	$3M^2(196,608)$	$4M^2 + 4M + 7(263,175)$	5	1
DMPF _x NLMS	$4M + 6L + 14(1098)$	$5M + 2L + 13(1313)$	$2L + 5(25)$	1

To provide a fair comparison of these algorithms, the step size was selected as $\mu = 0.01$ for NLMS based algorithms, so that they have a similar ANR results in Gaussian noise environment. According to [13], the parameter ($-1 < \rho < 1$) of PFxNLMS needed to be set manually to fit different environments, and its optimal value was selected by simulations as shown in Figure 4. According to Figure 4b, the optimal value of PFxNLMS was $\rho = -0.5$. Thus, PFxNLMS(Opt) with $\rho = -0.5$ was used to represent the PFxNLMS in the following simulations.

Table 3. Parameters for different methods in this paper.

Algorithms	Parameters
FxLMS	$\mu = 2.5 \times 10^{-4}$
FxlogLMS	$\mu = 2.5 \times 10^{-4}, \Delta = 1 \times 10^{-10}$
PFxNLMS	$\mu = 0.01, \rho = -0.5$
FxRMC	$\lambda = 0.999, N_w = 2, \delta = 0.02$
DMPF _x NLMS	$\mu = 0.01, N_q = 2, N_w = 100, L = 10, \text{initial } \theta = 0.4\pi$

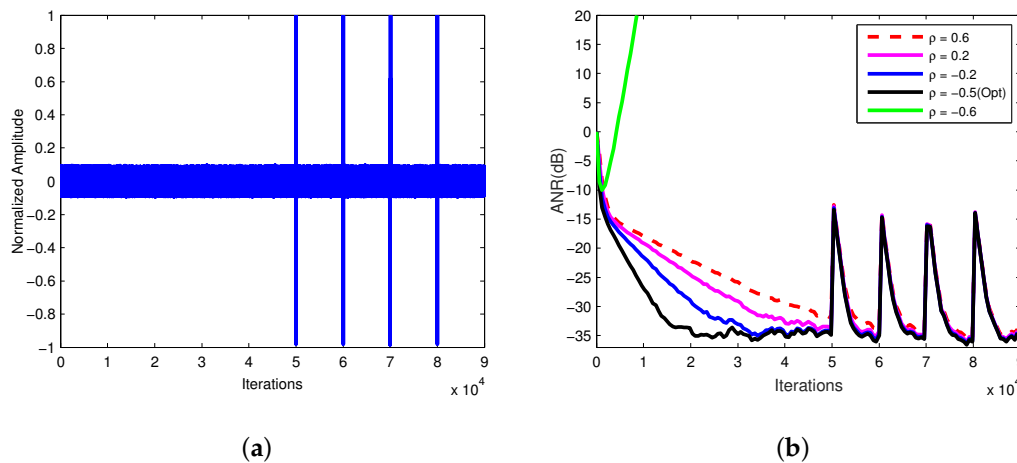


Figure 4. Selection of optimal ρ for PFxNLMS. (a) Reference signal in time domain; (b) ANR performance.

3.1. Random Impulsive Interference

The stable noise environment was simulated by the mixture of sinusoidal signal of 150 Hz, 300 Hz, 450 Hz and additional Gaussian noise. In this simulation, using [18,19], the impulsive interference process was produced by a random noise function of a symmetric α -stable ($S\alpha S$) distribution. The probability function of the random noise producer is given by

$$\varphi(t) = \exp(-|t|^\alpha). \tag{47}$$

The α changed with different noise environments. To make comments of the proposed DMPF_xNLMS completely, the Gaussian noise ($\alpha = 2$) field and the variable- α environment should also be considered. According to this idea, as shown in Figure 5, in the noise interference environment where $\alpha = 1.2$ and $\alpha = 1.7$, the M-estimation method based FxlogLMS, the PFxNLMS(Opt) and FxRMC were used to compare with DMPF_xNLMS. The impulsive interferences began at the 25,000-th iteration and end with the 31,000-iteration. The changing process of $\varepsilon(n)$ is also shown to give further information to understand DMPF_xNLMS.

In Figure 5a, PFxNLMS(Opt) diverges at about 30,000 iterations. The FxRMC has the best initiate tracking ability, when impulsive interference is strong enough ($\alpha = 1.2$), it diverges at about 25,000 iterations. The FxlogLMS performs worse on noise reduction level, and it cannot keep steady when interference occurs. The proposed DMPF_xNLMS had a lower computational load than FxRMC, and it had similar initial tracking performance and better robust performance when strong interference

happened. In Figure 5b, it is noted that in the iterations between 30,000 and 31,000, DMPF_xNLMS works in robust state, which shows the reason of that DMPF_xNLMS dose not diverge at the 30,000 iterations like the PF_xNLMS(Opt).

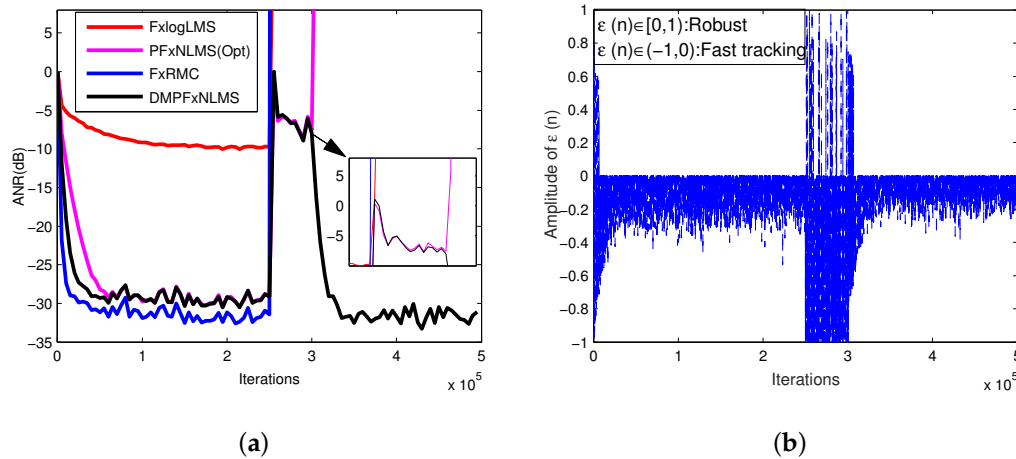


Figure 5. Performance comparison when impulsive interference $\alpha = 1.2$ (a) ANR curves; (b) changing process of $\epsilon(n)$.

In a lower degree impulsiveness noise environment as shown in Figure 6a, the DMPF_xNLMS had better tracking performance than that of the PF_xNLMS(Opt). Although the FxRMC has superior performance than other methods, it has bigger computational load. The Figure 6b reveals that DMPF_xNLMS works most of the time in fast tracking mode, which proves the conclusion again that DMPF_xNLMS could detect the impulsive degrees and find optimal parameters for adaptive filters.

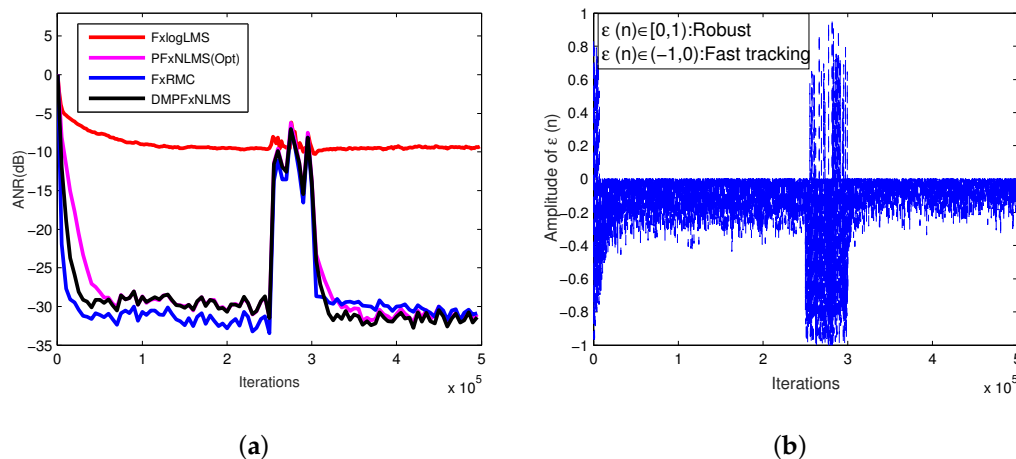


Figure 6. Performance comparison when impulsive interference $\alpha = 1.7$ (a) ANR curves; (b) changing process of $\epsilon(n)$.

Based on theory analysis, it is noted that the PF_xNLMS used the certain ρ to fit a certain environment. Considering Figures 5a and 6a, the impulsive process always has a variable α , which may decrease the performance of PF_xNLMS. The FxlogLMS was able to deal with burst interference, but it had lower noise reduction level in Gaussian environment, which was also demonstrated in [9]. For FxRMC, the RLS based algorithm had better tracking capability than DMPF_xNLMS, but it cannot

be used when severe impulsive interference occurs. Simulations in this section show that the proposed DMPF_xNLMS works well in both burst and less-burst noise field.

3.2. Periodic Impulsive Interference

Periodic impulsive noise is also common in acoustic signals [20], to test the tracking and robust performance of DMPF_xNLMS in this environment, several simulations are conducted. The 300 Hz sinusoidal wave is added periodically to represent the periodic impulsive noise. The interferences of 300 Hz noise are amplified as 15 dB and 5 dB compared with the reference signal in Figure 4a. The comparisons of ANR performance for the mentioned algorithms are shown in Figure 7a,b.

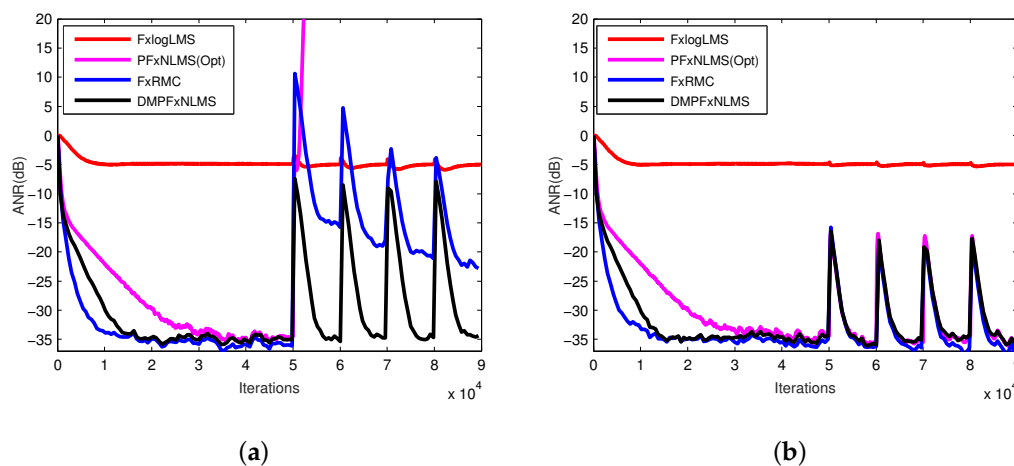


Figure 7. ANR performance (a) interference amplifying 15 dB; (b) interference amplifying 5 dB.

In Figure 7a, the PFXNLMS(Opt) diverged at about 50,000 iterations. The Fxlms maintains the steady state all the times, but it had bad ANR performance. The FxRMC had the best initial tracking, but its fluctuation became bigger when the impulsive occurred. In fact, the FxRMC had nearly perfect tracking due to the RLS core. But the bigger fluctuation was also led by its perfect tracking. When burst noise occurred, the weight vectors of FxRMC were changing fast. But the impulsive interferences often disappeared quickly, the weight vector needed to be changed again. It changed too fast such that the FxRMC had bigger fluctuations in Figure 7a. However, under the same conditions, the DMPF_xNLMS had a good initial tracking rate and remained stable when exposed to periodic burst signals.

As shown in Figure 7b, the periodic impulsive interference was smaller than that in Figure 7a, although the mentioned methods had different tracking rates and ANR performance when interference happens, they all kept steady.

In Sections 3.1 and 3.2, the simulation results show that the PFXNLMS(Opt) with a certain ρ cannot fit different S_aS environment. The Fxlms had a bad ANR performance. As for the FxRMC, it really had a superior tracking performance due to its RLS core, but it cannot fit strong impulsive noise environment. Compared with these mentioned methods, the DMPF_xNLMS had comprehensive using. To get further details of robust performance of DMPF_xNLMS, the robustness analysis is discussed deeply in the next section.

3.3. Robustness Analysis

In this section, simulations are conducted to investigate the significance of introducing a detection method into the adaptive filtering process. The state detector was mainly used to enhance the robust performance of DMPF_xNLMS, therefore, we only considered the performance in completely impulsive noise environments in this section. Considering the robust performance in the Sections 3.1 and 3.2, only the PFXNLMS, Fxlms, and DMPF_xNLMS were chosen here for sake of clarity. These methods

have been demonstrated to perform robustly above. The signal with $\alpha = 1.1$ described by the SaS distribution was introduced to simulate the completely impulsive noise environment.

In Figure 8a, the PFxNLMS(Opt) and FxlogLMS had similar ANR performance, but the fluctuation ranges of ANR of them were bigger than the proposed DMPF_xNLMS. The Figure 8b is used to show that DMPF_xNLMS worked in its robust mode when sever noise happens. More exactly, most values fall in the interval (0, 1), which proves that DMPF_xNLMS worked in the robust state. The results demonstrate that DMPF_xNLMS was capable of automatic parameter's setting, which really improves the robustness when burst noise happens.

Figure 8c shows the Euclidean norm of the different methods, which is also used to explain the robust performance of adaptive filters. The FxlogLMS has smaller Euclidean norm values, but its fluctuation range is bigger than DMPF_xNLMS. As for the PFxNLMS(Opt), both its Euclidean norm value and fluctuation range were bigger than DMPF_xNLMS. It is also confirmed that the burst amplitude signals have little influence on the convergence rate of DMPF_xNLMS, suggesting that it is more robust under these conditions.

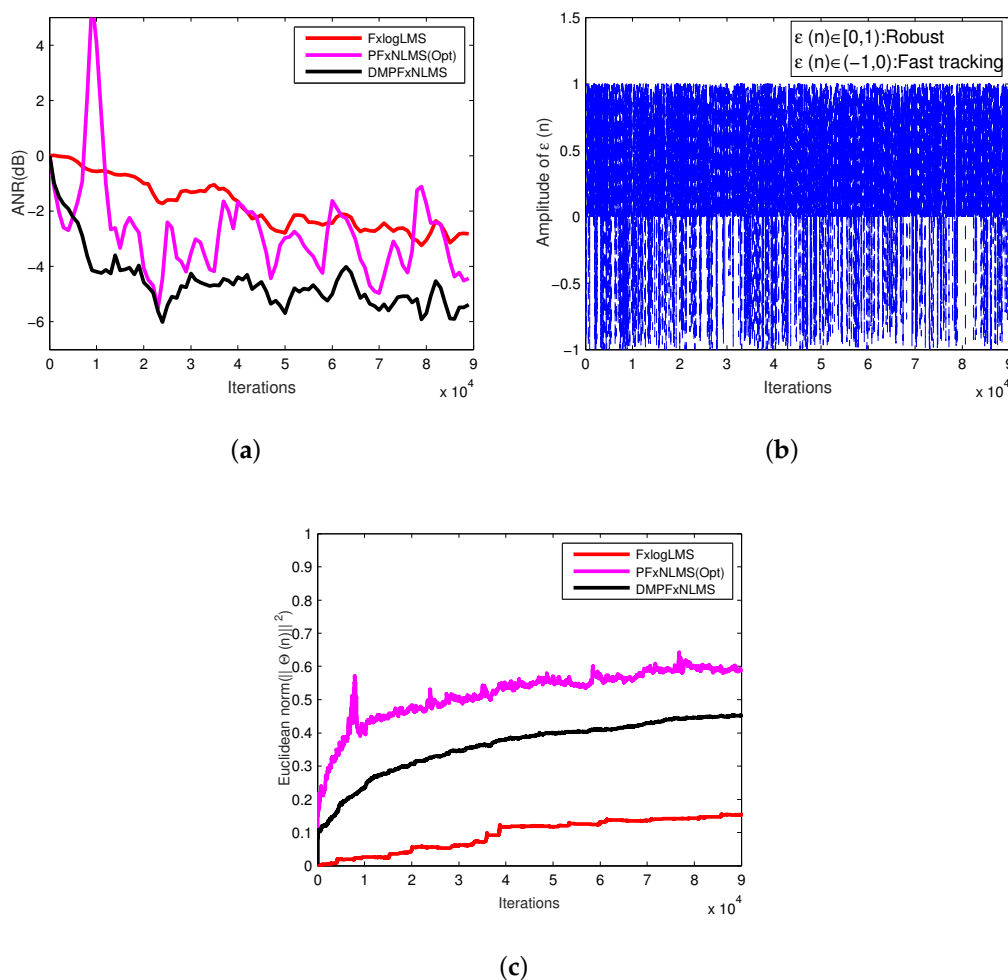


Figure 8. Robustness analysis (a) ANR curves; (b) changing process of $\epsilon(n)$; (c) Euclidean norm of weight vector $\Theta(n)$.

4. Experiments

To investigate the performance of the proposed method in real applications, experiments of real-time noise control were conducted. In this experiment, the real sound of car horns was used as

the reference signal. Recordings were taken in Songbei Area of Harbin, Heilongjiang province, China. An NI 4472 data acquisition card was used for recording, and the horn is that of a Volkswagen car.

In real-time experiments, the FxLMS, FxlogLMS, PFxNLMS, and DMPF_xNLMS were applied to IntervalZero RTX 64 real-time system, respectively. The standard FxLMS was introduced to evaluate the performance of the proposed DMPF_xNLMS. The FxRMC was deprived in the real-time experiments due to its heavy computational load [21–24]. The estimated secondary path using off-line LMS algorithm was modelled as a FIR filter with 16 orders [25]. The sampling frequency was 8000 Hz. The length of weight vector was 32 taps. The diagram of the real-time experiment is shown in Figure 9. Figure 10a,b show the analysis of the record sound in the time domain and frequency domain, respectively.

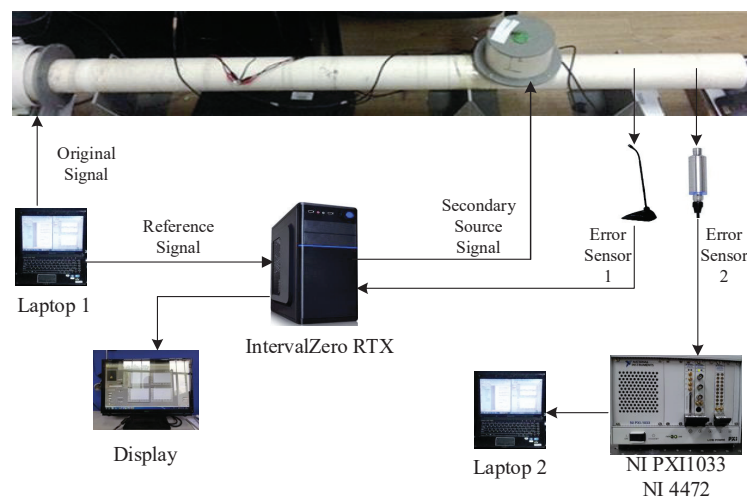


Figure 9. Diagram of the real-time experiment.

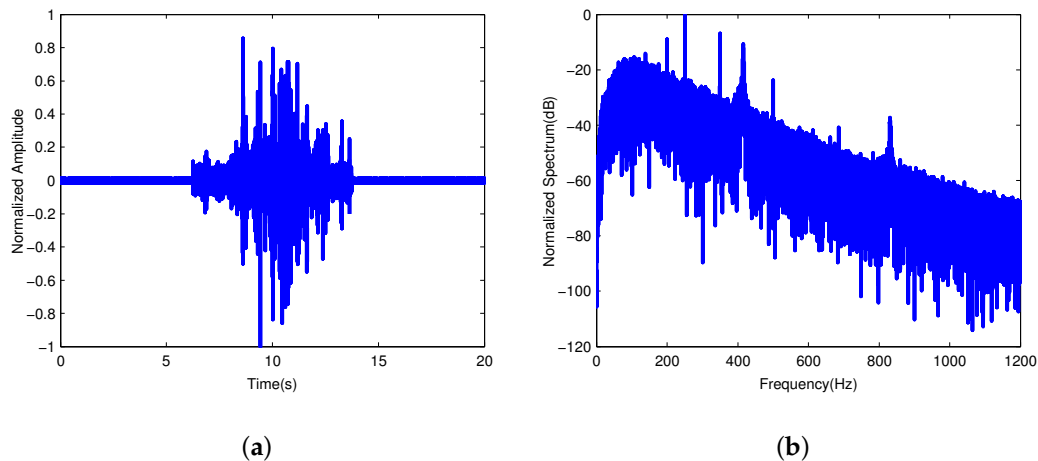


Figure 10. Reference signal of real-time experiment (a) time domain; (b) frequency domain.

The error learning curves are shown in Figure 11. The FxLMS nearly diverged at 11 s, thus it was stopped manually after that time. It shows that DMPF_xNLMS had better attenuation performance than FxlogLMS and PFxNLMS. To show clearly, the ANR performance was considered to give further results in Figure 12.

In Figure 12, the attenuation of FxLMS was inferior to that of other methods applied in this experiment when impulsive interference occurred, it diverged at about 11 s, which also confirmed the conclusion in [8–15] theoretical analysis that the FxLMS cannot cancel impulsive

noise. Although the FxlogLMS could maintain the steady state after the impulsive interference, the average attenuation of it was inferior to that of PFxNLMS(Opt) and DMPFxFNLMS. Comparing to PFxNLMS(Opt), DMPFxFNLMS had a better initial convergence rate. Between 6 s and 15 s, it performed more stably in this experiment. When the impulsive interference disappeared, it had better average attenuation than the PFxNLMS. These proved that the proposed DMPFxFNLMS had better performance in the Gaussian noise environment with accidental impulsive interferences.

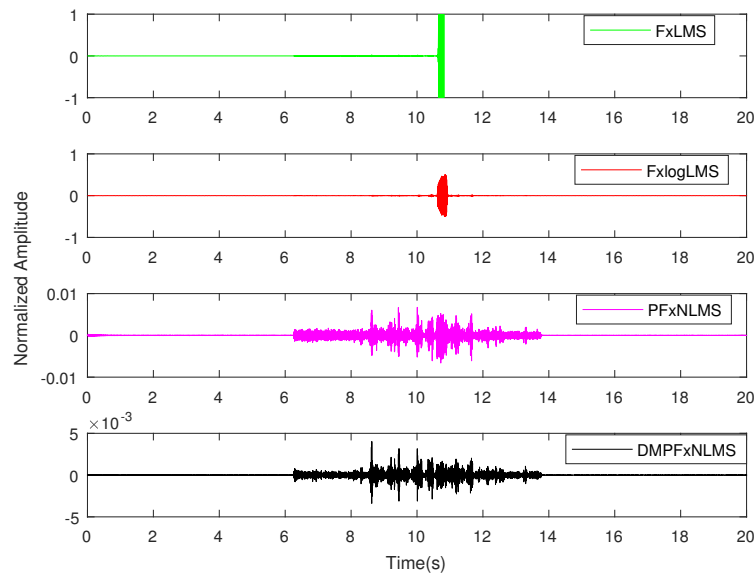


Figure 11. Performance comparison of noise in error learning curves.

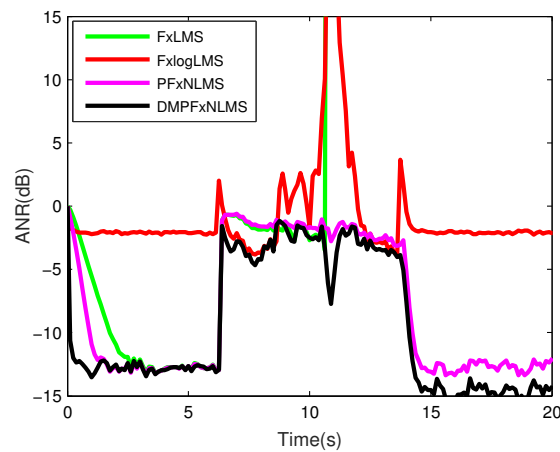


Figure 12. ANR performance comparison of the real-time experiment.

5. Conclusions

In this paper, a novel adaptive state detector-based post-filtering active control (DMPFxFNLMS) algorithm is proposed to achieve ANC in both non-impulsive and impulsive noise environment. An adaptive Gaussian core is introduced that allows the convergence rate of the weight vector to be raised and lowered more quickly than conventional methods when the value of the feedback signal $e(n)$ increases and decreases in fast tracking mode. A novel state detector is also developed to keep robust for the adaptive filter when the divergence of $e(n)$ is barely acceptable, and its feasibility is proved by theoretical analysis. Simulations and experimental results show that the proposed DMPFxFNLMS algorithm can provide faster tracking than the PFxNLMS algorithm using optimal

parameter in non-impulsive interference environment and have better robust performance than FxRMC and FxlogMLS algorithms. Moreover, the DMPF \times NLMS algorithm also has less computational load than FxRMC algorithm. Therefore, the proposed DMPF \times NLMS algorithm is able to provide better ANC performance in Gaussian environment with impulsive interference.

Author Contributions: Conceptualization, W.Z.; methodology, W.Z. and L.L.; software, W.Z.; validation, W.Z. and L.L.; formal analysis, W.Z. and L.L.; investigation, W.Z.; resources, S.S. and J.S.; data curation, W.Z. and L.L.; writing—original draft preparation, W.Z.; writing—review and editing, W.Z. and L.L.; visualization, W.Z. and L.L.; supervision, W.Z. and L.L.; project administration, L.L.; funding acquisition, S.S. and J.S.

Funding: This research was funded by National Science Foundation of China, grant number 61171183, 61471140. This research is also funded by Aerospace Support Fund in Harbin Institute of Technology, grant number 01320214.

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

References

1. Kuo, S.M.; Morgan, D.R. *Active Noise Control Systems: Algorithms and DSP Implementations*; Wiley: New York, NY, USA, 1996.
2. Zimmermann, M.; Dostert, K. Analysis and modelling of impulsive noise in broad-band powerline communications. *IEEE Trans. Electromagn. Compat.* **2002**, *44*, 249–258. [[CrossRef](#)]
3. Lui, L.; Gujjula, S.; Thanigai, P.; Kuo, S.M. Still in Womb: Intrauterine acoustic embedded active noise control for infant incubators. *Adv. Acoust. Vib.* **2008**, *10*. [[CrossRef](#)]
4. Georgiou, P.G.; Tsakalides, P.; Kyriakakis, C. Alpha-stable modeling of noise and robust time-delay estimation in the presence of impulsive noise. *IEEE Trans. Multimed.* **1999**, *10*, 291–301. [[CrossRef](#)]
5. Leahy, R.; Zhou, Z.; Hsu, Y.C. Adaptive filtering of stable processes for active attenuation of impulsive noise. In Proceedings of the 1995 International Conference on Acoustics, Speech, and Signal Processing, Detroit, MI, USA, 9–12 May 1995; pp. 2983–2986.
6. Sun, X.; Kuo, S.M.; Meng, G.A. Adaptive algorithm for active control of impulsive noise. *J. Sound Vib.* **2006**, *291*, 516–522. [[CrossRef](#)]
7. Akhtar, M.T.; Mitsuhashi, W. Improving performance of FxLMS algorithm for active noise control of impulsive noise. *J. Sound Vib.* **2009**, *327*, 647–656. [[CrossRef](#)]
8. Wu, L.F.; He, H.S.; Qiu, X. An active impulsive noise control algorithm with logarithmic transformation. *IEEE Trans. Audio Speech Lang.* **2011**, *19*, 1041–1044. [[CrossRef](#)]
9. Li, P.; Yu, X. Active noise cancellation algorithms for impulsive noise. *Mech. Syst. Signal Proc.* **2013**, *36*, 630–635. [[CrossRef](#)] [[PubMed](#)]
10. Tan, L.; Jiang, J. Active control of impulsive noise using a nonlinear companding function. *Mech. Syst. Signal Proc.* **2015**, *58*, 29–40. [[CrossRef](#)]
11. Al-Sayed, S.; Zoubir, A.M.; Sayed, A.H. Robust adaptation in impulsive noise. *IEEE Trans. Signal Process.* **2016**, *65*, 2851–2865. [[CrossRef](#)]
12. Akhtar, M.T.; Mitsuhashi, W. Improving robustness of filtered-x least mean p-power algorithm for active attenuation of standard symmetric- α -stable impulsive noise. *Appl. Acoust.* **2011**, *72*, 688–694, doi:10.1016/j.apacoust.2011.02.009. [[CrossRef](#)]
13. Wu, L.F.; Qiu, X. Active impulsive noise control algorithm with post adaptive filter coefficient filtering. *IET Signal Process.* **2013**, *7*, 515–521. [[CrossRef](#)]
14. Bergamasco, M.; Della Rossa, F.; Piroddi, L. Active noise control with on-line estimation of non-Gaussian noise characteristics. *J. Sound Vib.* **2012**, *331*, 27–40. [[CrossRef](#)]
15. Lu, L.; Zhao, H. Active noise control using maximum correntropy with adaptive kernel size. *Mech. Syst. Signal Proc.* **2017**, *87*, 180–191. [[CrossRef](#)]
16. Mahmood, A.; Chitre, M. Optimal and Near-Optimal Detection in Bursty Impulsive Noise. *IEEE J. Ocean. Eng.* **2017**, *42*, 639–653. [[CrossRef](#)]
17. Otsu, N. A threshold selection method from gray-level histograms. *IEEE Syst. Man Cybern. B* **1979**, *9*, 62–66. [[CrossRef](#)]
18. Nikiyas, C.L.; Shao, M. *Signal Processing with Alpha-Stable Distributions and Applications*; Wiley-Interscience: New York, NY, USA, 1995.

19. Liu, W.; Pokharel, P.P.; Principe, J.C. Correntropy: Properties and applications in non-gaussian signal processing. *IEEE Trans. Signal Process.* **2007**, *55*, 5286–5298. [[CrossRef](#)]
20. Zhou, Y.; Yin, Y.; Zhang, Q. An optimal repetitive control algorithm for periodic impulsive noise attenuation in a non-minimum phase ANC system. *Appl. Acoust.* **2013**, *74*, 1175–1181. [[CrossRef](#)]
21. Sun, G.; Li, M.; Lim, T.C. A family of threshold based robust adaptive algorithms for active impulsive noise control. *Appl. Acoust.* **2015**, *97*, 30–36. [[CrossRef](#)]
22. Mirza, A.; Zeb, A.; Sheikh, S.A. Robust adaptive algorithm for active control of impulsive noise. *EURASIP J. Adv. Signal Process.* **2016**, *44*. [[CrossRef](#)]
23. Kurian, N.C.; Patel, K.; George, N.V. Robust active noise control: An information theoretic learning approach. *Appl. Acoust.* **2017**, *117*, 180–184. [[CrossRef](#)]
24. Luo, L.; Sun, J. A novel bilinear functional link neural network filter for nonlinear active noise control. *Appl. Soft. Comput.* **2018**, *68*, 636–650. [[CrossRef](#)]
25. He, J.; Lam, B.; Shi, D.; Gan, W.S. Exploiting the Underdetermined System in Multichannel Active Noise Control for Open Windows. *Appl. Sci.* **2019**, *9*, 390. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).