A Neural Network Assisted FuLMS Algorithm for Active Noise Control System

Liang Jiang[1(✉)], Hongqing Liu[2], Liming Shi[2], and Yi Zhou[2]

[1] School of Communications and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China
S210131072@stu.cqupt.edu.cn

[2] Intelligent Speech and Audio Research Lab, Chongqing University of Posts and Telecommunications, Chongqing, China
{shilm,zhouy}@cqupt.edu.cn

Abstract. In active noise control (ANC) systems, the Filtered-u Least Mean Square (FuLMS) algorithm has better control performance and faster rate of convergence than Filtered-x Least Mean Square (FxLMS) algorithm. However, due to the instability of adaptive infinite impulse response (IIR) filters, the application of FuLMS algorithm is not as extensive as that of FxLMS algorithm using adaptive finite impulse response (FIR) filters. The Equation Error (EE) method for adaptive IIR filtering can solve stability issues caused by poles, so in this paper we use the EE-based adaptive IIR filter to improve the FuLMS algorithm. Moreover, in this work, we introduce a neural network assisted method for designing adaptive IIR filters coefficients for the use in ANC systems. The results, compared with the FuLMS algorithm and neural network only approach, demonstrate the effectiveness of this scheme.

Keywords: Active Noise Control · Filtered-u Least Mean Squares Algorithm · Equation Error Method · Neural Network · Infinite Impulse Response Filter

1 Introduction

Noise in the real world usually has a significant power in the low frequency range. However, the traditional passive methods are not effective to suppress the low frequency noise [1]. Active noise control (ANC) generates equal amplitude and opposite phase anti-noise to suppress unwanted noise. The use of ANC systems as a complement of passive methods for canceling unwanted low frequency noise signals have widely been investigated in various applications [1–5].

Conventional ANC systems employ adaptive algorithms to attenuate noise and two structures are commonly used to implement ANC systems: adaptive finite impulse response (FIR) filter and adaptive infinite impulse response (IIR) filter [6]. Using an adaptive IIR filter requires fewer parameters for modeling than

F. Gao et al. (Eds.): Chinacom 2023, LNICST 590, pp. 412–427, 2024.
https://doi.org/10.1007/978-3-031-67162-3_26

an adaptive FIR filter and also leading to a less computational complexity [7–9]. Employing an adaptive IIR filter with least mean square (LMS) algorithm [10,11] in ANC is called FuLMS algorithm [4,12]. Moreover, adaptive IIR filters can be further divided into output-error (OE) [12–14] models and equation-error (EE) models [14–16]. The OE-based IIR filter has both zeros and poles, subjecting to an unstable system and local minimum. The EE filter has only zeros and no poles, and therefore, the EE based methods for adaptive IIR filtering can achieve the global minimum as long as the appropriate step size is selected [14]. For that reason, this study adopts the EE-based IIR filter based ANC system ensuring that the FuLMS algorithm is not affected by unstable poles.

In the EE IIR filter-based ANC system, there are still problems such as slow convergence speed and poor noise reduction performance. In [17], authors provide a method for training filter parameters with deep neural networks. The parameters of the filter can be trained using the back propagation of a deep neural network to minimize the cost function and since the EE IIR filter has no recursive filter, the neural network can directly train the coefficients of the filter.

Inspired by this, we propose a neural network assisted FuLMS algorithm in ANC, where a neural network is designed to learn the initial weights of adaptive IIR filter. After that, the learned weights are utilized to initialize the IIR filter. Compared with the neural network only approach, this concept still maintains the adaptation of the filter because neural network only approach cannot adaptively update its weights when deployed. Compared with the traditional algorithm such as FuLMS, the convergence of the algorithm is boosted significantly. To the best of our knowledge, this is the first study that exploits the ability of a neural network in IIR filter design for ANC system.

2 Methods

2.1 Filtered-u Algorithm Based Output-Error Adaptive IIR Filter

A simple block diagram of OE-based Filtered-u algorithm is shown in Figure 1, where the reference input $x(n)$ represents the undesired noise and the output of the adaptive filter $y(n)$ drives a loudspeaker. The IIR filter structure is composed of two transverse FIR filters $A(z)$ and $B(z)$, which are feedforward filter and feedback filter, respectively [8]. The feedforward filter length is of L_1 taps and the feedback filter length is of L_2 taps, and time-domain input-output relationship of the IIR filter is

$$y(n) = \sum_{k=0}^{L_1} a_k(n)x(n-k) + \sum_{k=1}^{L_2} b_k(n)y(n-k). \tag{1}$$

For the IIR filter, the reference signal vector and the output vector are denoted by

$$\begin{aligned}\mathbf{X}(n) &= [x(n), x(n-1), \ldots, x(n-L_1+1)]^T,\\ \mathbf{Y}(n) &= [y(n-1), y(n-2), \ldots, y(n-L_2)]^T.\end{aligned} \tag{2}$$

We define the weight vector $\mathbf{W}(n)$ and the data vector $\mathbf{U}(n)$

$$\begin{aligned}\mathbf{W}(n) &= [a_0(n), a_1(n), \ldots, a_{L_1}(n), b_1(n), b_2(n), \ldots, b_{L_2}(n)]^T,\\ \mathbf{U}(n) &= [x(n), \ldots, x(n-L_1+1), y(n-1), \ldots, y(n-L_2)]^T.\end{aligned} \tag{3}$$

Based on the adaptive filter, the output $y(n)$ is given by

$$y(n) = \mathbf{W}^T(n)\mathbf{U}(n). \tag{4}$$

In Fig. 1, $\mathbf{H}_p(n)$ is the z-transform of the impulse response $\mathbf{h}_p(n)$, which denotes the transfer function of the duct from the reference microphone to the location of the loudspeaker, and the reference noise $x(n)$ passes through the primary path and becomes the desired noise $d(n)$, given by

$$d(n) = x(n) * h_s(n), \tag{5}$$

where $*$ denotes the convolution operation. The acoustical transfer function $\mathbf{H}_s(z)$, its impulse response is $h_s(n)$, represents the secondary path from the output loudspeaker to the error microphone. Based on those definitions, the error signal is

$$e(n) = d(n) - \mathbf{W}^T(n)[\mathbf{U}(n) * h_s(n)]. \tag{6}$$

The weight vector updates equation of two transversal filter weights recursively and it is

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \mathbf{u}e(n)\mathbf{r}(n), \tag{7}$$

where $\mathbf{u}$ is a $(L_1+L_2)\times(L_1+L_2)$ diagonal matrix diagonal matrix [18] denoting the step size and it has step size u_1 for feedforward filter $A(z)$, step size u_2 for feedback filter $B(z)$ with a form of

$$\mathbf{u} = diag[\overbrace{(u_1, \ldots, u_1)}^{L_1}, \overbrace{(u_2, \ldots, u_2)}^{L_2}]. \tag{8}$$

In (7), $\mathbf{r}(n)$ is the filtered reference signal generated by passing data signal $\mathbf{U}(n)$ through the estimated secondary path $\hat{h}_s(n)$, given by

$$\mathbf{r}(n) = \mathbf{U}(n) * \hat{h}_s(n). \tag{9}$$

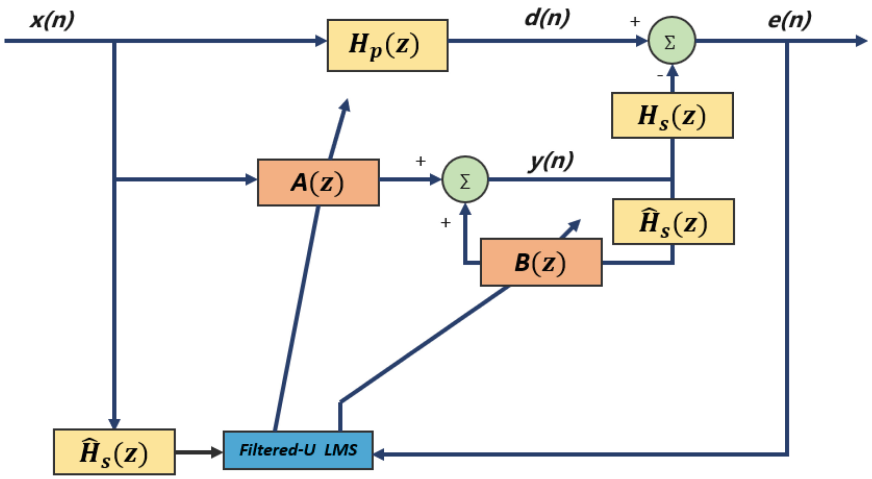


Fig. 1. A simplified ANC block diagram of OE-model Filtered-U LMS algorithm.

2.2 Filtered-u Algorithm Based Equation-Error Adaptive IIR Filter

According to above algorithm based output-error adaptive IIR filter In OE model, the adaptive IIR filter uses a recursive form to obtain the input-output relationship as in (1) and the transfer function of the filter is given by

$$H_{OE}(z) = (A(z))/(1 - B(z)). \tag{10}$$

According to (10), recursive filter poles are subject to stability issues since the poles of the IIR filter can be outside the unit circle if not designed well. It may also make the adaptive filter parameters fail to converge to the global minimum, because the OE model is highly nonlinear. The EE-model IIR filter has a unique global minimum (stationary) point associated with the classical least squares solution, so if can avoid that issue and its block diagram is shown in Fig. 2.

In EE model, the output signal can be expressed as

$$y(n) = \sum_{k=0}^{L_1} a_k(n)x(n-k) + \sum_{k=1}^{L_2} b_k(n)d(n-k). \tag{11}$$

The desired noise signal $d(n)$ is obtained by (5), and in this method, $\mathbf{U}(n)$ in FuLMS algorithm is different from (3), given by

$$\mathbf{U}(n) = [x(n), \ldots, x(n-L_1+1), d(n-1), \ldots, d(n-L_2)]^T. \tag{12}$$

The error signal $e(n)$ is the same as in (6). Moreover, the input $\mathbf{X}(n) = [x(n), \cdots, x(n-L_1+1)]$ passing through the estimated secondary path $\hat{h}_s(n)$ produces

$$\mathbf{X}^{'}(n) = \mathbf{X}(n) * \hat{h}_s(n), \tag{13}$$

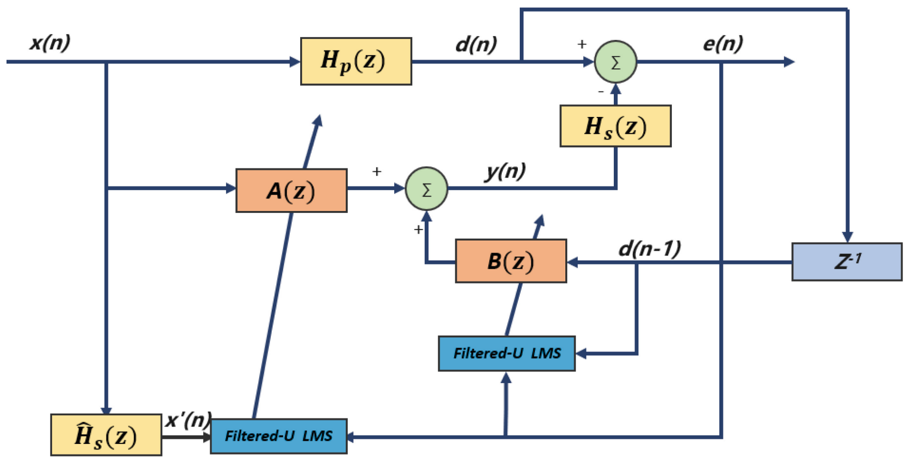


Fig. 2. A simplified ANC block diagram of EE-model Filtered-U LMS algorithm.

where $\mathbf{X}^{'}(n) = [x^{'}(n), \cdots, x^{'}(n-L_1+1)]$, and for the adaptation of the EE-based ANC system, we define $\mathbf{U}^{'}(n) = [x^{'}(n), \cdots, x^{'}(n-L_1+1), d(n-1), \cdots, d(n-L_2)]^T$, and the weight update of the EE-based ANC system can be derived

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \mathbf{u}e(n)\mathbf{U}^{'}(n), \tag{14}$$

where $\mathbf{W}(n)$ is the weight vector and the step-size diagonal matrix $\mathbf{u}$ can be obtained by (8).

Using the EE IIR filter model, the adaptive IIR filter system transfer function now is

$$H_{EE}(z) = A(z) + H_p(z)B(z). \tag{15}$$

Because the output signal does not contain a recursive term and the EE-based IIR filter as a combination of two transversely filters and the EE-based IIR filter is like that of an adaptive FIR filter. They have similar adaptive algorithms and similar convergence properties. The convergence rate and stability of coefficient updating are usually determined by the eigenvalues of the Hessian matrix [14]. Therefore instability can be avoided compared with OE method. Since neither the input signal nor the expected signal is a function of the adaptive filter parameters, it is believed that the only approximate relationship in gradient calculation is generated by the expected value estimation required in reality. Mean-Squared value of the EE (MSEE) is a quadratic function of the parameters, so as long as the signal is uniformly excited, there is only one global minimum point. The adaptive procedure can also reach the global minimum with a suitable step size [14,15,19]. However, EE based methods still suffer from slow convergence speed and unsatisfactory noise reduction performance.

2.3 Neural Network for Initialization

Supervised learning is used for this task, and the propose of the neural network is to obtain the filter coefficients for the initialization [17,20] and the neural network is trained by backpropagation to obtain an optimal coefficient as an output that minimizes losses in the frequency domain. Each set of noise requires different training, which means that the network does not need to generalize, but instead performs optimization by fitting its weights, unlike common deep learning classification and regression tasks.

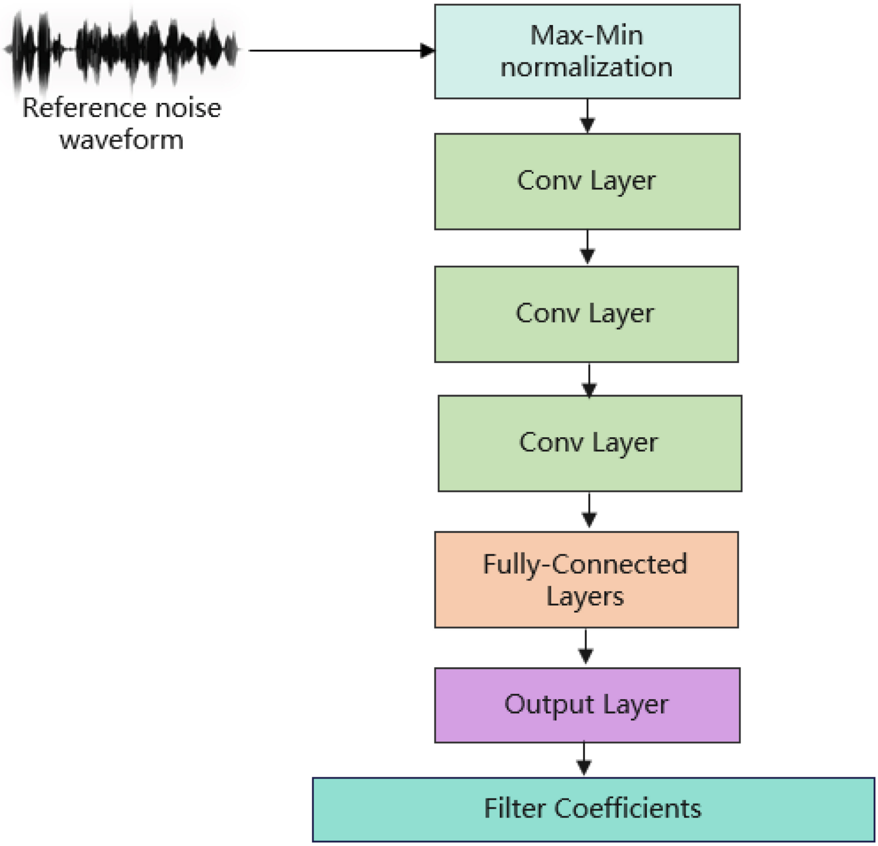


Fig. 3. The structure of Convolutional neural network.

The adaptive IIR filter based on EE model is similar consisting of two adaptive FIR filters, without recursive filter. Therefore, the EE method can avoid the pole problem of the recursive filter, and the neural network can directly obtain the parameters of the IIR filter by calculating parameters of two FIR filters. To that aim, we design a neural network as follows. We adopt the 1D Convolutional neural networks architecture and provide a reference signal $x(n)$ and an ideal noise $d(n)$ to the network for training [17,21]. The 1D Convolutional neural networks consist of series 1D convolutional layers and fully connected layers [22]. The convolutional layer reduces the size of the input data while providing features to the fully connected layer. The architecture of the network is illustrated in Fig. 3. The input reference signal is a one-dimensional vector of size $1 \times L$, which is the input of neural network after regularization. The final fully connected layer of the neural network outputs the coefficients of the filter. The feedforward filter length is L_1 taps and the feedback filter length is L_2. In that sense, the first L_1 terms of the neural network output are the coefficients of the feedforward filter and the remaining terms are the coefficients of the feedback filter. An issue related to the proposed filter design technique is the computational cost, as the design process requires complete training of the network. The experiment needs to set a large number of iteration coefficients, but the loss function decays very fast. We can set the required error threshold in advance, and we will stop training when the threshold is reached.

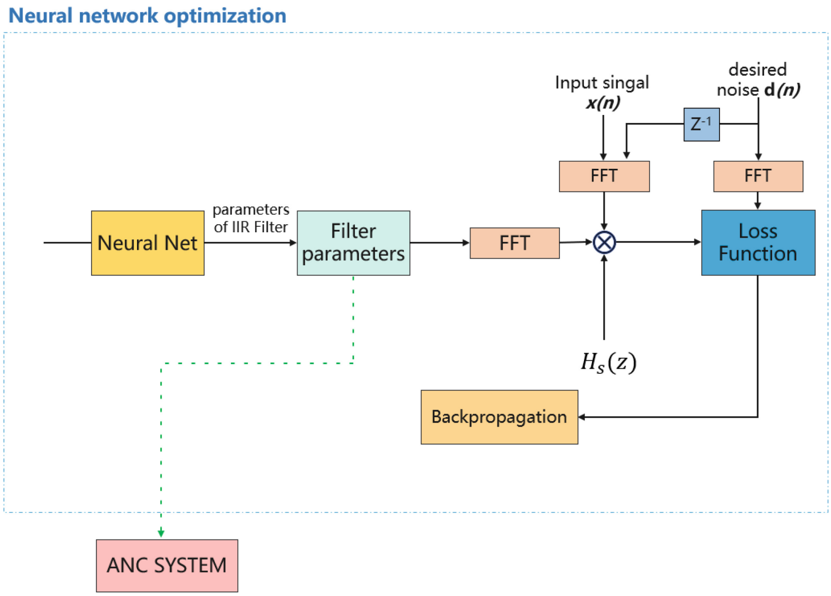


Fig. 4. Block diagram of the neural network initializes the filters of the FuLMS algorithm. Once the neural network computes the optimal filter parameters, these parameters are used to initialize the feedforward and feedback filters, respectively.

Assume the generated IIR filters include feedforward parametric filter G_a and feedback parametric filter G_b, the input signal $x(n)$ of length L and desired noise $d(n)$ are passed through the generated IIR filter and secondary path to generate anti-noise $\hat{y}$, given by

$$\hat{y} = h_s(n) * (G_a * x(n) + G_b * d(n-1)). \tag{16}$$

To train the network and reduce the computational complexity of each iteration, the Euclidean distance loss between the ideal anti-noise and the given anti-noise in frequency domain is used, given by

$$loss = \left\| \hat{Y}(\omega) - D(\omega) \right\|_2, \tag{17}$$

where $\hat{Y}(\omega)$ and $D(\omega)$ are the Fourier transforms of $\hat{y}$ and $d(n)$, respectively.

Finally, the filter coefficients obtained by the neural network are used as the initial values of the feedforward filter and feedback filter, and FuLMS algorithm uses those initials to adaptively update the weights. A brief framework diagram of the proposed method is shown in Fig. 4.

3 Experiments

In the experiments, we model the physical structure of the ANC system. We simulate a rectangular enclosure of size 1.23 m × 3.14 m × 2 m and the reference microphone is located at position of (0.8, 1, 1) m, the cancelling loudspeaker is located at position of (0.8, 2, 1) m, the error microphone is located at the position (1, 2.4, 1) m. We use the image method [23] to generate impulse responses $h_p(n)$ of primary path and $h_s(n)$ of secondary path. The impulse responses with reverberation time (T60) 0.3 s and the length of the $h_p(n)$ is set to 512, the length of the $h_s(n)$ is set to 256.

For the neural network, we adopt 1D convolutional layers to extract features with max-min normalized reference noise waveforms [24]. In the preliminary pre-experiment, we generated 10 single frequency noise signals with frequencies ranging from 200-2000 Hz. Each track has a duration of 1 s. The values of neural network hyperparameters and iteration coefficients are determined by preliminary experiments. In the pre-experiment, we observed that a high enough number of iterations can make the network converge to a very low error, but choosing an appropriate number of iterations can reduce the complexity and training time of the network training.

Table 1. Comparisons of CNNs with different configurations

(kernel size, channels, stride)	fully connected layers	Trainable Parameters
[(128, 1, 4), (64, 1, 2), (16, 1, 1)]	(100, 256)	0.23M
[(128, 1, 4), (64, 1, 2), (16, 1, 1)]	(1024)	2.04M
[(128, 1, 4), (64, 1, 1)]	(100, 256)	0.42M
[(128, 1, 4), (64, 1, 2), (16, 1, 1)]	(512, 100)	1.08M

Finding a suitable network architecture through pre-training can reduce the number of parameters [17]. The filter trained by the neural network structure listed in Table 1 can reduce the noise by about 20 dB when used in the ANC system, but the number of trainable parameters is very different. We adopt the neural network consisting of three convolutional layers and two fully connected layers. The input layer inputs a one-dimensional time-domain waveform. The size of three convolution layers are (128, 1, 4), (64, 1, 2), (16, 1, 1), respectively, and the configuration of convolution layer is form of $(kernelsize, channels, stride)$. The dimension of the kernel for the all convolutional layer is 1×1. The number of neurons in the two fully connected layers is 100 and 256, respectively. The output layer outputs the coefficients of the filter. We use exponential linear units (ELU) [25] in all convolutional and fully connected layers except the output layer. In the output layer, the activation function is sine [26] because the output weights will be used for initializing the adaptive filter. In the experiments, model is trained using the Adam optimizer [27] with a learning rate of 0.001 and the number of iterations of was set to 10000. The weights of the neural network were initialized with a uniform distribution.

3.1 Noise Cancellation Results

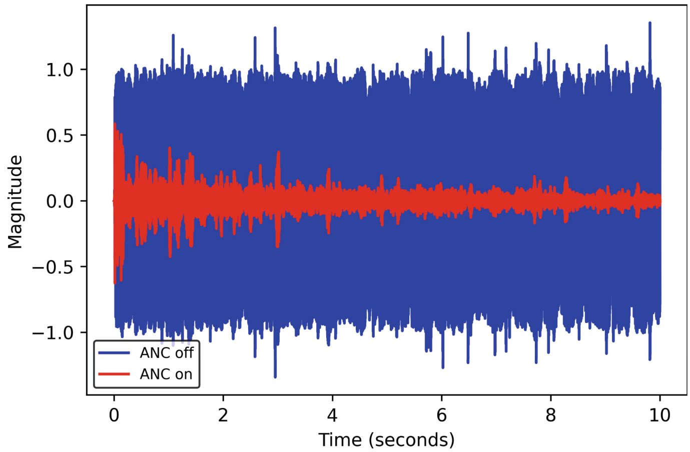
We initialize the IIR filters of the FuLMS algorithm with the filter parameters obtained from the neural network and we refer to this approach as Initialized Filters FuLMS (IFFuLMS), where the step size of the FuLMS algorithm is set to 0.04. Moreover, the length of both feedforward and feedback filters are 512 taps since the dimension of the neural network output is 1024.

Single Frequency Noise Cancellation. In this simulation, we created a noise with a frequency of 200 Hz. The noise reduction results using different methods on the single frequency noise are shown in Fig. 5. In this work, noise reduction level (NRL) is defined by

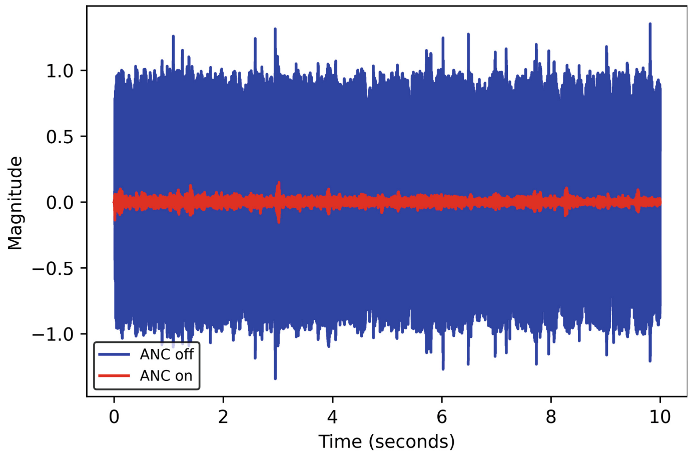
$$NRL(dB) = 10 \log_{10} \frac{P_d}{P_e}, \tag{18}$$

where P_d and P_e are the powers of ideal noise $d(n)$ and error signal $e(n)$, respectively. The experimental results show, for single-frequency noise signal, that the proposed method significantly improves the convergence speed and noise reduction performance of FuLMS algorithm. In Fig. 5(c), we also include the results from neural network, and it is seen that the noise reduction performance of the filter trained by the neural network is not as good as the traditional FuLMS algorithm. However, FuLMS algorithm is improved by using it as the initial values. During first 0–2 s, the NRL of the proposed method is on average 12.5

dB higher than that of the FuLMS algorithm. After reaching the convergence steady state, the NRL of the proposed method is 4.3 dB higher than that of the FuLMS algorithm. This result verifies our claim that neural network can assist the traditional algorithms in actively reducing the noise, while maintaining the adaptation ability.

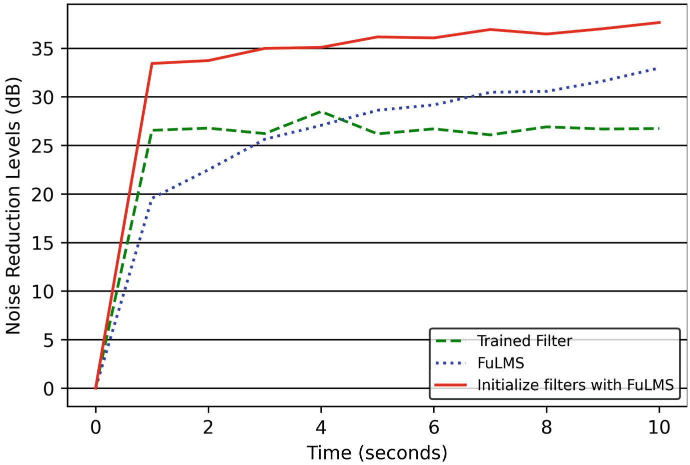


(a) The FuLMS algorithm



(b) The IFFuLMS algorithm

Fig. 5. (a): FuLMS algorithm; (b): IFFuLMS algorithm; (c): NRL of different algorithms on single frequency noise.



(c) NRL of different algorithms

Fig. 5. (*continued*)

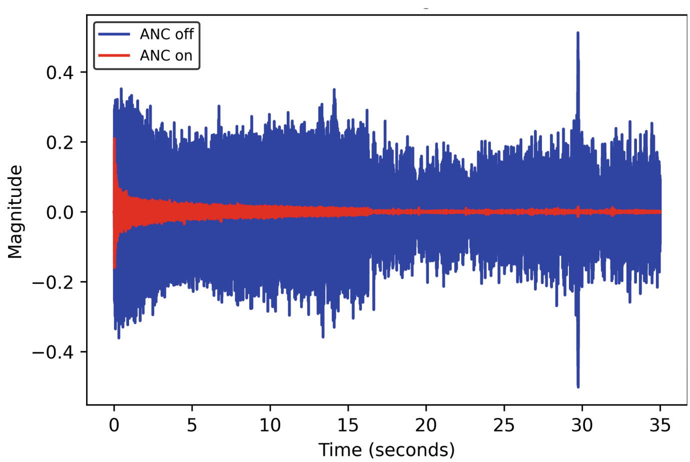
Real World Noise Cancellation. In this simulation, the noise comes from a real-world factory with a frequency range of 35–4000 Hz. The frequency band and amplitude of this noise change over time. Figure 6 depicts the performance of different algorithms. From the results, for real-world noise, the proposed method can still effectively attenuate the noise. The filter obtained by neural network is not adaptive as the traditional method, but the proposed method has the adaptive ability of FuLMS algorithm in addition to the advantage of a fast convergence. During 5–10 s, the NRL of the proposed method is on average 10 dB higher than that of the FuLMS algorithm. In addition, the NRL of the proposed method after the convergence is also 5.2 dB higher than that of the FuLMS algorithm.

3.2 Length of Filters Effect

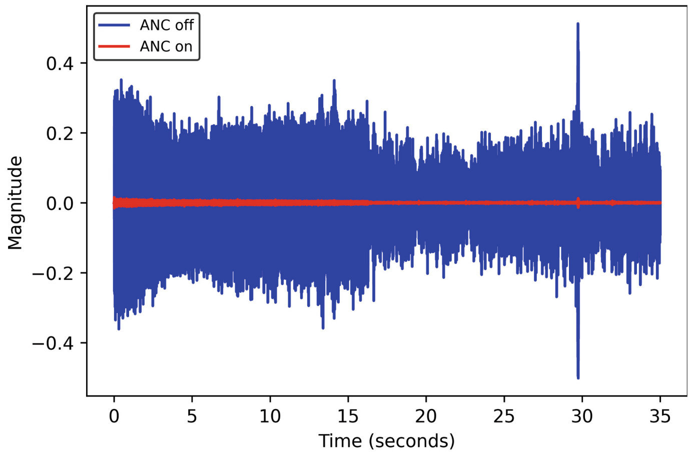
Another advantage of using this concept is filter length used can be reduced because of a good initialization provided by the neural network. The same or even better performance can be achieved using filters with a shorter length of taps.

In this section, the step size of the FuLMS algorithm is set to 0.04 and the noise comes from voice in a noisy environment. We compare the noise reduction level of the FuLMS algorithm and the proposed method with different filter lengths. The FuLMS algorithm with the feedforward filter length of 512 taps and the feedback filter length of 256 taps is studied. The proposed method with feedforward filter length of 256 taps and feedback filter length of 128 taps is used. The noise reduction results are shown in Fig. 7. To fit the new filter length, the number of neurons in the two fully connected layers is 512 and 384 and the network was retrained.

From the results, when the length of the feedforward filter is 256 taps and the length of the feedback filter is 128 taps, the noise reduction performance and convergence speed of the FuLMS algorithm are poor. The noise reduction performance of the proposed method with shorter length is better the FuLMS algorithm with the longer length, which really shows the potential of using neural network in boosting the performance of traditional methods.

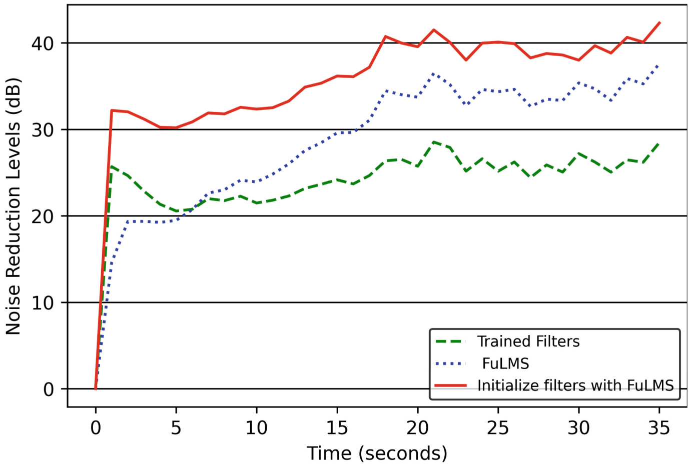


(a) The FuLMS algorithm



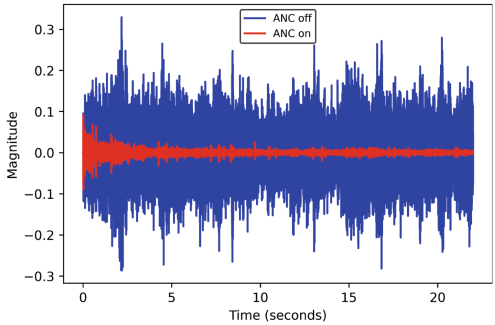
(b) The IFFuLMS algorithm

Fig. 6. (a): FuLMS algorithm; (b): IFFuLMS algorithm; (c): NRL of different algorithms on real-world noise.

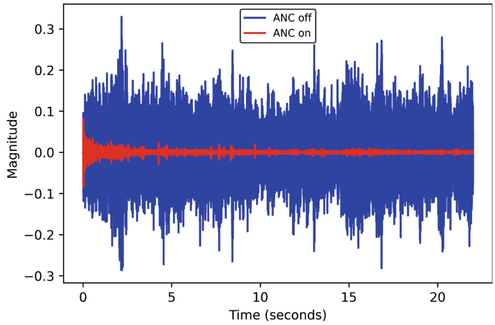


(c) NRL of different algorithms

Fig. 6. (*continued*)

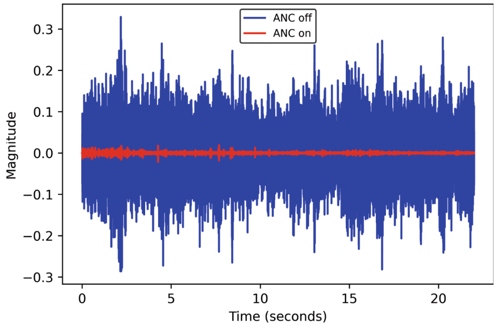


(a) The FuLMS algorithm with 256 and 128 taps

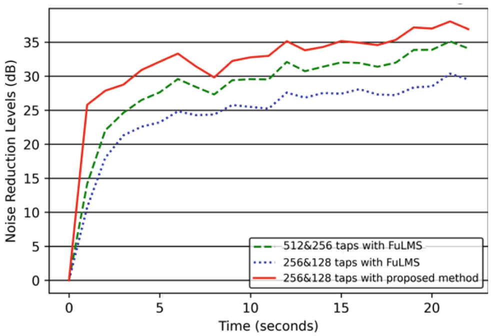


(b) The FuLMS algorithm with 512 and 256 taps

Fig. 7. (a): FuLMS algorithm with feedforward filter length 256 taps and feedback filter length 128 taps; (b): FuLMS algorithm with feedforward filter length 512 taps and feedback filter length 256 taps; (c): The proposed method with feedforward filter length 256 taps and feedback filter length 128 taps; (d) NRL of different methods.



(c) The IFFuLMS algorithm with 256 and 128 taps



(d) NRL of filters with different lengths

Fig. 7. (*continued*)

4 Conclusion

In this work, we leverage the power of the neural network to produce EE-based IIR filter coefficients. Although the computational cost of the training phase is relatively high, the convergence rate is quite fast, allowing us to set the required error threshold to stop the iteration immediately when the goal is reached to reduce the number of iterations. At the same time, pre-training can be used to find the appropriate network structure parameters and reduce the training parameters. The FuLMS algorithm is used to adaptively update the filter coefficients with the initialization provided by the neural network parameters. Simulation results show that the proposed method presents a good noise reduction performance and robustness in ANC system. This method can reduce the length of the adaptive filter required, which means reducing the computational cost in the ANC system. In future work, more lightweight neural networks can be further investigated to demonstrate the performance.

References

1. Hansen, C.H.: Understanding Active Noise Cancellation. CRC Press, Boca Raton (2002)
2. Elliot, S., Nelson, P.: Active noise control. Noise News Int. **2**, 75–98 (1994)
3. Kuo, S.M., Morgan, D.R.: Active noise control: a tutorial review. Proc. IEEE **87**(6), 943–973 (1999)
4. Kim, K.K.I., Na, H., Park, Y.: Constraint filtered-x and filtered-u least-mean-square algorithms for the active control of noise in ducts. J. Acoust. Soc. Amer. **95**, 3379–3389 (1994)
5. Widrow, B., Stems, S.: Adaptive Signal Processing. Prentice Hall, Englewood Cliffs (1985)
6. Haykin, S.: Adaptive Filter Theory. PrenticeHall, Englewood Cliffs (1991)
7. Regalia, P.: Adaptive IIR Filtering in Signal Processing and Control. Marcel Dekker (1994)
8. Shynk, J.: Adaptive IIR filtering. IEEE ASSP Mag. **6**, 4–21 (1989)
9. Carini, A., Mathews, V.J., Sicuranza, G.L.: Sufficient stability bounds for slowly varying direct-form recursive linear filters and their applications in adaptive IIR filters. IEEE Trans. Signal Process. **47**(9), 2561–2567 (1999)
10. Johnson, C.R., Larimore, M.: Comments on and additions to "an adaptive recursive LMS filter". Proc. IEEE **65**(9), 1399–1402 (1977)
11. Fujii, K., Ohga, J.: Individually normalized LMS algorithm suitable for fixed point processing and its application to active noise control. IEICE Trans. **178–A**, 669–677 (1995)
12. Eriksson, L.J.: Development of the filtered-u algorithm for active noise control. J. Acoust. Soc. Amer **89**, 257–265 (1991)
13. Bouvet, M., Schwartz, S.C.: Comparison of adaptive and robust receivers for signal detection in ambient underwater noise. IEEE Trans. Acoust. Speech Signal Process. **37**(5), 621–626 (1989)
14. Feng, D.-Z., Zheng, W.X.: Fast RLS-type algorithm for unbiased equation-error adaptive IIR filtering based on approximate inverse-power iteration. IEEE Trans. Signal Process. **53**, 4169–4185 (2005)
15. Regalia, P.A.: Comparison of Adaptive and Robust Receivers for Signal Detection in Ambient Underwater Noise. Marcel Dekker, New York (1995)
16. Yuan, J., Li, J., Zhang, A., Zhang, X., Ran, J.: Active noise control system based on the improved equation error model. In: Acoustics, pp. 354–363 (2021)
17. Pepe, G., Gabrielli, L., Squartini, S., Cattani, L.: Designing audio equalization filters by deep neural networks. Science **10**, 2483 (2020)
18. Chang, C., Shyu, K.-K.: A self-tuning fuzzy filtered-u algorithm for the application of active noise cancellation. IEEE Trans. Circuits Syst. I: Fundam. Theory Appl. **49**(9), 1325–1333 (2002)
19. Ho, C.-Y., Shyu, K.-K., Chang, C.-Y., Kuo, S.M.: Equation-error model based active noise cancellation systems. In: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8474–8478 (2019)
20. Villarrubia, G., De Paz, J.F., Chamoso, P., De la Prieta, F.: Artificial neural networks used in optimization problems. Neurocomputing **272**, 10–16 (2018)
21. Pepe, G., Gabrielli, L., Squartini, S., Tripodi, C., Strozzi, N.: Deep optimization of parametric IIR filters for audio equalization. IEEE/ACM Trans. Audio Speech Lang. Process. **30**, 1136–1149 (2022)

22. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning, vol. 1, pp. 421–425. MIT Press, Cambridge (2016)
23. Allen, J.B., Berkley, D.A.: Image method for efficiently simulating small-room acoustics. Acoust. Soc. Am. **65**(4), 943–950 (1979)
24. Dai, W., Dai, C., Qu, S., Li, J., Das, S.: Very deep convolutional neural networks for raw waveforms. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 421–425 (2017)
25. Clevert, D. A., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (ELUs). arXiv preprint arXiv:1511.07289 (2015)
26. Sitzmann, V., Martel, J., Bergman, A., Lindell, D., Wetzstein, G.: Implicit neural representations with periodic activation functions. Adv. Neural. Inf. Process. Syst. **33**, 7462–7473 (2020)
27. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations, San Diego, CA, USA, pp. 1–15 (2015)