INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

HAD-ANC: A Hybrid System Comprising an Adaptive Filter and Deep Neural Networks for Active Noise Control

JungPhil Park, Jeong-Hwan Choi*, Yungyeo Kim, and Joon-Hyuk Chang*

Department of Electronic Engineering, Hanyang University, Seoul, Republic of Korea
{wndvlf96, brent1104, kimyungyeo, jchang}@hanyang.ac.kr

Abstract

Our study proposes a novel hybrid active noise control (ANC) system, called HAD-ANC, that combines an adaptive filter with deep neural networks. HAD-ANC employs a cascade design comprising the frequency-domain block least mean square algorithm and two gated convolutional recurrent networks (GCRNs). The first GCRN follows the adaptive filter to handle nonlinear distortion by reducing the residual error of linear filtering and models the reverse of both loudspeaker and secondary path. The second GCRN models the loudspeaker and secondary path to force the adaptive filter to estimate the primary path. Additionally, we utilize a delay-compensated reference signal to consider the causal constraints of frequency-domain ANC system. Experimental results based on NOISEX-92 dataset show that the proposed system outperforms recent ANC methods, enables wideband noise reduction, and indicates robustness to path changes.

Index Terms: active noise control, adaptive filter, deep learning, hybrid system, nonlinear distortion

1. Introduction

Active noise control (ANC) is a technology that reduces noise levels in a target location by generating an anti-noise signal with the same magnitude but opposite phase via controller to cancel out the noise signal received by an error microphone [1]. Typically, feedforward ANC designs involve a controller considering both primary and secondary paths. The traditional method has applied adaptive filters to the controller, where filter weights are updated by an optimization algorithm at every time step. The filtered-x least mean square (FxLMS) [2], which represents adaptive filter-based ANCs, is most commonly used as the controller for noise reduction, owing to its simple implementation and low computation load. Despite these advantages, this method uses linear filtering, which cannot address the nonlinear distortion caused by the limited quality of the loudspeaker [3,4]. The structural solution [5–9] have been proposed to address nonlinearites.

Recently, researches have been conducted to introduce deep neural networks (DNN), suitable for modeling nonlinear relationships, into ANC systems [10–12]. Deep ANC [10] employed a frequency-domain convolutional recurrent network (CRN) [13] as a controller, which was the first end-to-end deep learning approach for ANC. However, the deep ANC involves more computations than adaptive filters and operates in the frequency-domain, resulting in high algorithmic delay. Zhang *et al.* [11] introduced a low-latency deep ANC that utilizes a revised overlap-add algorithm during signal resynthesis to avoid latency while maintaining noise reduction performance. On the other hand, secondary path decoupled ANC (SPD-ANC) [12] adopted time-domain CRNs, known as SPD modules, to estimate the forward and reverse impulse response (IR) of loudspeaker and secondary path. The SPD modules compute the SPD error signal by processing the error and control signals, which are received at the error microphone and generated at the adaptive filter. The adaptive filter of SPD-ANC is subsequently updated to minimize the SPD error signal.

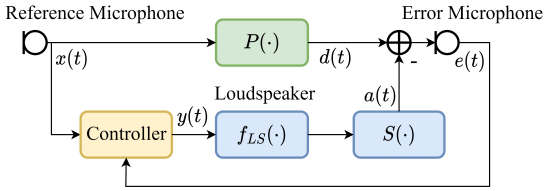

Figure 1: *Block diagram of typical feedforward ANC system.*

This paper proposes a *Hybrid system of an Adaptive filter and DNNs for the ANC* (HAD-ANC) system for effective noise attenuation. We employ a cascade structure of an adaptive filter and DNN as a controller, where two algorithms estimate the IR of the primary path and inverse IR of the loudspeaker and secondary path, respectively. The DNN also minimizes residual error caused by underestimating the primary path through linear filtering. The cascaded structure complements each other, addressing weaknesses caused by insufficient modeling capabilities and fixed parameters. Another DNN models the forward IR of the loudspeaker and secondary path to estimate the anti-noise effect. The two DNNs enable the adaptive filter to estimate the primary path exclusively. The detailed implementation of the HAD-ANC employs a normalized frequency-domain block least mean square (NFBLMS) [14, 15] algorithm and two gated CRNs (GCRNs) [16] for efficient operation via block processing. Additionally, we utilize a delay-compensated reference signal to consider the causal constraints of ANC system [17].

2. Overview of typical ANC system

Figure 1 outlines the typical framework of the feedforward ANC system, which includes the reference and error microphones and loudspeaker. The primary and secondary paths connect these components. The reference noise signal, $x(t)$, becomes the desired signal, $d(t)$, to be deleted through the IR of the primary path, $P(\cdot)$. The control signal, $y(t)$, becomes anti-noise, $a(t)$, owing to the IR of the loudspeaker, $f_{LS}(\cdot)$, and secondary path, $S(\cdot)$, which cancels out the desired signal. Finally, the error signal, $e(t)$, is expressed as follows:

$$\begin{aligned} e(t) &= d(t) - a(t) \\ &= P(x(t)) - S(f_{LS}(y(t))), \end{aligned} \tag{1}$$

*Equal contributions.

10.21437/Interspeech.2023-1795

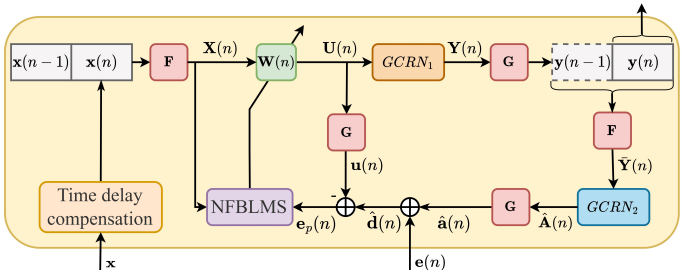

Figure 2: *Block diagram of the proposed HAD-ANC controller.*

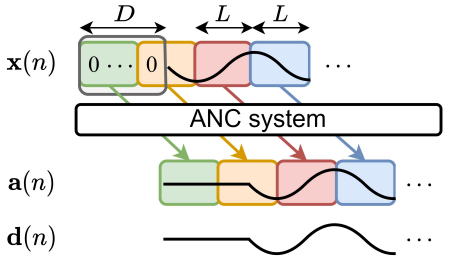

Figure 3: *Diagram of the delay compensation method.*

where t is the time index. The goal of the feedforward ANC system is to build the controller to model $P(\cdot)/S(f_{LS}(\cdot))$ considering real-time. In the FxLMS series algorithm, $x(t)$ is filtered into a pre-trained linear filter to reduce the effect of $S(\cdot)$. Subsequently, it is converted into $y(t)$ with an adaptive filter. On the other hand, the controller of the deep ANC generates $y(t)$ using the end-to-end deep learning model. The controller of the SPD-ANC contains an adaptive filter and deep learning-based SPD module that alleviates the effect of $S(\cdot)$, similar to FxLMS. Note that $y(t)$ is generated with a convolution of $x(t)$ and a single adaptive filter in SPD-ANC.

3. The proposed HAD-ANC system

Prior research has employed either an adaptive filter or a DNN to produce control signals $y(t)$. However, the adaptive filter-based ANC systems have limited noise attenuation at high frequencies because of convergence and latency, as well known in these fields [2, 18]. It is likewise recognized that end-to-end deep learning approaches exhibit incapability in coping with situations where the primary path suddenly changes [12]. Thus, our motivation is to tackle both difficulties by deploying a hybrid ANC system that concatenates an adaptive filter and DNN in a cascading configuration.

Figure 2 shows the details of the proposed HAD-ANC controller, which is capable of block processing in an overlap-save manner, comprising a NFBLMS and two frequency-domain GCRNs. We employ frequency-domain algorithms allowing for potential delays because the computational complexity is significantly less than the time-domain algorithm [10, 14]. Figure 3 illustrates a delay compensation method due to the primary path in the proposed HAD-ANC. We employ delay-compensated L-length signal vectors $\mathbf{x}(n) = [x(nL - D), x(nL + 1 - D), \cdots, x(nL + L - 1 - D)]^T$ and $\mathbf{d}(n) = [d(nL), d(nL + 1), \cdots, d(nL + L - 1)]^T$ where n, T, and D denote the block index, transpose operation, and delay index between the reference and error microphone, respectively. Consequently, the frequency-domain reference signal vector, $\mathbf{X}(n) = \text{diag}\left[\mathbf{F}[\mathbf{x}^T(n-1), \mathbf{x}^T(n)]^T\right]$ is obtained with a $2L$-point discrete Fourier transform (DFT), where $\mathbf{F}$ is a $2L \times 2L$ DFT matrix and diag$[\boldsymbol{\sigma}]$ is a diagonal operation that make a $2L \times 2L$-size diagonal matrix with the components of the L-length column vector $\boldsymbol{\sigma}$; $\mathbf{w}(n) = [w_0(n), w_1(n), \cdots, w_{L-1}(n)]^T$ is the L-tap filter; $\mathbf{W}(n) = \mathbf{F}[\mathbf{w}^T(n), \mathbf{0}_{L\times1}^T]^T$ is the weight in the frequency-domain; and $\mathbf{0}_{L\times1}$ is the L-length vector with zero components. The result of linear filtering $\mathbf{U}(n)$ is fed into the first neural network, $GCRN_1$ to generate the control signal vector $\mathbf{y}(n)$ as follows:

$$\begin{aligned}\mathbf{Y}(n) &= GCRN_1(\mathbf{X}(n)\mathbf{W}(n)) \\ &= GCRN_1(\mathbf{U}(n)),\end{aligned} \tag{2}$$

$$\mathbf{y}(n) = \mathbf{G}\mathbf{Y}(n). \tag{3}$$

We define $\mathbf{G} = [\mathbf{0}_{L\times L}; \mathbf{I}_{L\times L}]^T\mathbf{F}^{-1}$ as a time-domain drop operation that removes L samples from the end of the block; $\mathbf{F}^{-1}$ denotes $2L \times 2L$ inverse DFT matrix. Subsequently, an anti-noise vector, $\mathbf{a}(n)$, is obtained by feeding $\mathbf{y}(n)$ into $f_{LS}(\cdot)$ and $S(\cdot)$. The error signal vector $\mathbf{e}(n)$ is obtained by subtracting $\mathbf{a}(n)$ from the desired signal vector, $\mathbf{d}(n)$.

The HAD-ANC controller uses the error between $\mathbf{d}(n)$ and $\mathbf{u}(n)$ as an adaptive filter input to estimate $P(\cdot)$, where $\mathbf{u}(n)$ is the temporal representation of $\mathbf{U}(n)$. The error is defined as the primary error signal vector $\mathbf{e}_p(n)$. However, $\mathbf{a}(n)$ and $\mathbf{d}(n)$ are unknown because the error microphone only treats $\mathbf{e}(n)$. We estimate $\mathbf{a}(n)$ and $\mathbf{d}(n)$ as $\hat{\mathbf{a}}(n)$ and $\hat{\mathbf{d}}(n)$, respectively, by modeling $S(f_{LS}(\cdot))$ as the second neural network, $GCRN_2$. Then $\mathbf{e}_p(n)$ is formulated as follows:

$$\begin{aligned}\mathbf{e}_p(n) &= \hat{\mathbf{d}}(n) - \mathbf{u}(n) = \mathbf{e}(n) + \hat{\mathbf{a}}(n) - \mathbf{u}(n) \\ &= \mathbf{e}(n) + \mathbf{G}(GCRN_2(\bar{\mathbf{Y}}(n)) - \mathbf{U}(n)),\end{aligned} \tag{4}$$

$$\bar{\mathbf{Y}}(n) = \mathbf{F}\left[[\mathbf{G}\mathbf{Y}(n-1)]^T, [\mathbf{G}\mathbf{Y}(n)]^T\right]^T, \tag{5}$$

where $\bar{\mathbf{Y}}(n)$ is the input of the loudspeaker in the frequency-domain. Then, $\mathbf{e}_p(n)$ is converted into a frequency-domain primary error vector, $\mathbf{E}_p(n)$, by padding with the zero vector $\mathbf{0}_{L\times1}$:

$$\mathbf{E}_p(n) = \mathbf{F}[\mathbf{0}_{L\times1}^T, \mathbf{e}_p^T(n)]^T. \tag{6}$$

The updated equation for the NFBLMS using the obtained $\mathbf{X}(k)$ and $\mathbf{E}_p(k)$ is as follows:

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \mu\mathbf{Q}\boldsymbol{\Lambda}^{-1}\mathbf{X}^H(n)\mathbf{E}_p(n), \tag{7}$$

$$\mathbf{Q} = \mathbf{F}\begin{bmatrix}\mathbf{I}_{L\times L} & \mathbf{0}_{L\times L} \\ \mathbf{0}_{L\times L} & \mathbf{0}_{L\times L}\end{bmatrix}\mathbf{F}^{-1}, \tag{8}$$

where μ and H represent a constant step-size, Hermitian operation. $\boldsymbol{\Lambda} = E\left[\mathbf{X}(n)^H\mathbf{X}(n)\right]$ is a diagonal matrix comprising the reference signal powers at each frequency bin, where E is a expectation operation. $\boldsymbol{\Lambda}$, along with the μ, is known as the normalized step-size and enables a increase in convergence [15].

In summary, the NFBLMS and two GCRNs have different roles. The NFBLMS and $GCRN_2$ estimate the primary path and the path where the control signal changes to anti-noise, respectively. The $GCRN_1$ is designed to minimize the residual error caused by the limitations of modeling owing to the linearity of the adaptive filter while simultaneously estimating the nonlinearity of the loudspeaker and secondary path. High-level wideband noise attenuation is expected in the proposed HAD-ANC system because the $GCRN_1$ can reduce the noise additionally, which is attenuated through linear filtering.

3.1. Training steps for DNNs

We propose two training steps in which the GCRNs are trained separately according to their purpose. First, the $GCRN_1$ is

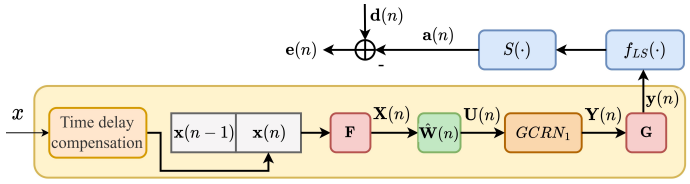

Figure 4: *Training step of* $GCRN_1$.

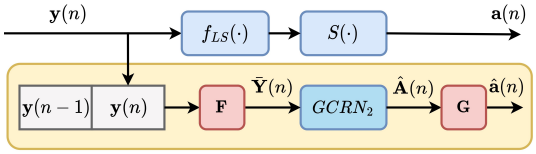

Figure 5: *Training step of* $GCRN_2$.

trained to reduce the noise signal that is fed after the adaptive filter, as illustrated in Figure 4. Because the primary path error used as input to the adaptive filter is affected by the $GCRN_1$ output, combining the two algorithms during training is challenging. Inspired form teacher forcing [19], we store the fixed weights, $\hat{\mathbf{W}}(n)$, by estimating the primary path using the NFBLMS algorithm in advance and subsequently used them for the $GCRN_1$ training. Considering that the adaptive filter can not converge sufficiently, the weights of the NFBLMS algorithms are extracted at various times to obtain the robustness of the $GCRN_1$ against the estimated primary path error. We utilize normalized mean square error (NMSE) as the loss function, which is calculated with error signal and desired signal as:

$$L_{GCRN_1} = 10\log_{10}\left[\Sigma_n \mathbf{e}^2(n)/\Sigma_n \mathbf{d}^2(n)\right]. \quad (9)$$

Figure 5 shows the second training step that trains the $GCRN_2$ to simultaneously estimate the IR of the loudspeaker and secondary path. The loss function is calculated with the output of the $GCRN_2$ and anti-noise as follows:

$$L_{GCRN_2} = 10\log_{10}\left[\Sigma_n |\mathbf{a}(n) - \hat{\mathbf{a}}(n)|^2/\Sigma_n \mathbf{a}^2(n)\right]. \quad (10)$$

4. Experimental setup

4.1. Data preparation

The development set consisted of 26 h noise signals from the DEMAND and MS-SNSD datasets excluding the babble signals. These noise signals were split into six-second audio clips and normalized. A validation set consisting of 1,563 signals was chosen for every tenth clip in the development set and the remaining clips, totaling 14,076 signals, composed a training set. To introduce scale variation in the training process, we randomly multiplied the training set by a value between 0.3 and 1.0 for every epoch. The engine, factory, and babble noises of the NOISEX-92 dataset [20] were utilized for testing. These signals did not overlap with the development set.

We simulated a rectangular cuboid space of 3 m × 4 m × 2 m, following a typical scenario [10, 11, 21] where the primary noise source was far from the wall. The positions of the reference microphone, error microphone, and loudspeaker were (1.5, 1, 1) m, (1.5, 3, 1) m, and (1.5, 2.5, 1) m, respectively. Note that, the delay D is set to 90 samples due to the fixed room condition. IRs of the primary and secondary paths were set to 160-length, and those were generated with ten reverberation times (RT60s) of 0.15, 0.175, 0.225, 0.25 s were used for training, and 0.1, 0.2, 0.3, 0.4, 0.5, 0.6 s were used for testing. Following [5, 10–12], the scaled error function [22] was utilized to represent the loudspeaker saturation effect of the major nonlinearity of the ANC

Table 1: *Performance comparison of ANC systems for RT60 of 0.2 s in terms of the NMSE.*

Noise type	**Engine**			**Factory**			**Babble**		
η^2	1.0	0.5	0.1	1.0	0.5	0.1	1.0	0.5	0.1
FxLMS	-4.02	-4.02	-2.69	-2.02	-2.02	-1.99	-2.77	-2.77	-2.56
THF-FxLMS	-5.69	-5.69	-5.69	-2.34	-2.34	-2.34	-3.13	-3.13	-3.13
SPD-ANC	-7.09	-7.09	-7.08	-4.62	-4.62	-4.62	-7.30	-7.30	-7.30
Deep ANC	-10.49	-10.49	-10.48	-8.60	-8.60	-8.59	-10.49	-10.49	-10.48
HAD-ANC	**-10.72**	**-10.72**	**-10.71**	**-8.61**	**-8.61**	**-8.61**	**-10.74**	**-10.74**	**-10.74**

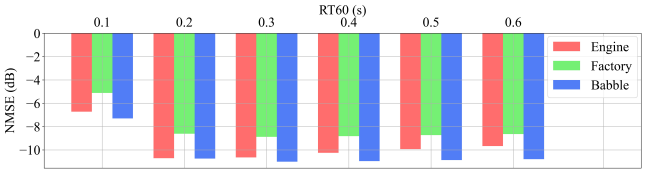

Figure 6: *HAD-ANC performance on untrained paths generated for different RT60s with* $\eta^2 = 0.1$ *in terms of the NMSE.*

as follows:

$$f_{LS}(y) = \int_0^y e^{-\frac{n^2}{2\eta^2}}\, dn, \quad (11)$$

where y is the input to the loudspeaker, and η^2 indicates the degree of nonlinearity of the loudspeaker. When η^2 tends to infinity, $f_{LS}(\cdot)$ becomes a linear function, while the nonlinearity of the function increases when η^2 approaches zero. To train the GCRNs, we randomly set a real number between 0.1 and 1 for η^2, and 0.1, 0.5, and 1.0 were chosen for testing.

4.2. Model configuration

Two GCRNs in the HAD-ANC had the same structure and consisted of 1.84 million parameters each. The GCRN consisted of an encoder with five gated convolutional blocks, two long short-term memory (LSTM) [23] layers, and a decoder with five gated deconvolutional blocks. Each gated convolutional block employed two convolutional layers with a stride of two and kernel size of three for the gated mechanism [24]. The frequency and channel dimension of the feature map was halved and doubled, respectively, each time it passed through a block. We set the output channel size at 16 for the first blocks. Batch normalization [25] and exponential linear unit activation function [26] were applied successively between the blocks. Two LSTM layers consisted of 128 nodes each, and the gated deconvolution blocks were symmetrical to the gated convolution blocks. Further details regarding the model configuration are similar to those in [16]. The length of the reference signal vector, L, was set to 64, which is smaller than the delay between reference microphone and loudspeaker. The step-size, μ, of the NFBLMS algorithm was set to 0.1 and the filter length was set to 64, the same as the length of the L.

Two GCRNs were trained using the Adam optimizer [27] for 30 epochs with a learning rate of 0.001 and the batch size was set to 32. The $GCRN_1$ and $GCRN_2$ were trained for ten hours each, on a system consisting of an Intel i9-11900K CPU @ 3.50GHz using 16 threads and an NVIDIA GeForce RTX 2080 Ti GPU, using PyTorch version 1.10.2[1]. To test the proposed HAD-ANC system, we used a single thread of the CPU, which achieved an average processing time of 2.92 ms for a 4 ms block, resulting in a real-time factor of 0.705. Note that, the block size of 4 ms was considered acceptable for conventional ANC methods [10].

[1]Our PyTorch implementation including subscription of data preparation at https://github.com/wndvlf96/HAD-ANC.

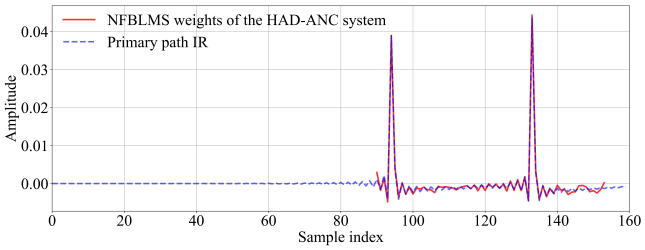

Figure 7: *Simulated primary path and estimated filter weights of HAD-ANC on the engine noise for RT60 of 0.2 s with* $\eta^2 = 0.1$.

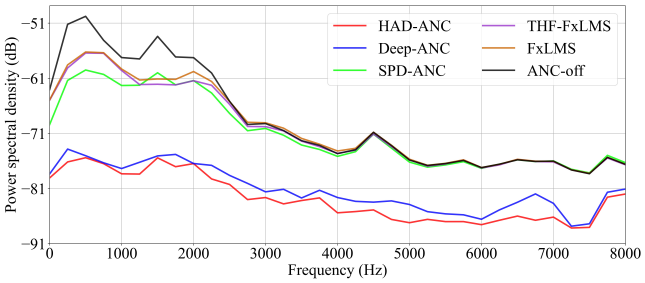

Figure 8: *Power spectral density of ANC systems on the engine noise for RT60 of 0.2 s with* $\eta^2 = 0.1$.

5. Experimental results

We evaluated the performance of the ANC systems in terms of NMSE on untrained paths. The performance of the proposed HAD-ANC was compared with those of the FxLMS [2], tangential hyperbolic function-based FxLMS (THF-FxLMS) [5], deep ANC [10], and SPD-ANC [12]. For the adaptive filter based ANC systems, the filter lengths were set to 64, the same as that of HAD-ANC. We report best step-size to yield high-level noise attenuation in validation set. We set μ to 0.5, 0.06, 0.1 with respect to engine noise, factory noise, babble noise in FxLMS and THF-FxLMS. μ for all types of noises in SPD-ANC was set to 0.01, 0.003, 0.001. In particular, the ANC systems of the FxLMS series estimated various secondary paths using the training set in advance, and the performance evaluation for each test set was averaged after measuring the NMSE using all secondary paths.

Table 1 and Figure 6 summarize the performance of the proposed models on the untrained IRs with different RT60s and compare them with other models for RT60 of 0.2 s. As shown in Figure 6, the HAD-ANC showed a low NMSE in a reverberation environment where RT60 was greater than 0.4 s. Notably, performance degradation occurred with RT60 of 0.1 s, and we confirmed that a mismatch occurred in estimating the IRs of the loudspeaker followed by the secondary path using the GCRN. When η^2 varied from 1.0 to 0.1, a performance degradation of the FxLMS was observed for all noises, as specified in Table 1. The SPD-ANC and THF-FxLMS performed better than the FxLMS because these could model nonlinearity. The SPD-ANC achieved a better noise attenuation than the THF-FxLMS on the untrained path. The deep ANC outperformed the SPD-ANC and THF-FxLMS in terms of the NMSE. The HAD-ANC slightly outperformed the deep ANC in all types of noises whenever η^2 changed and as η^2 decreased from 1.0 to 0.1. These results show that the HAD-ANC performed robustly on various reverberation environments, degrees of nonlinearity, and noise types. Figure 7 illustrates the differences between the filter weights of the HAD-ANC and primary path estimated from the experimental results in Table 1. We confirmed that the NFBLMS of the HAD-ANC sufficiently tracks the primary path,

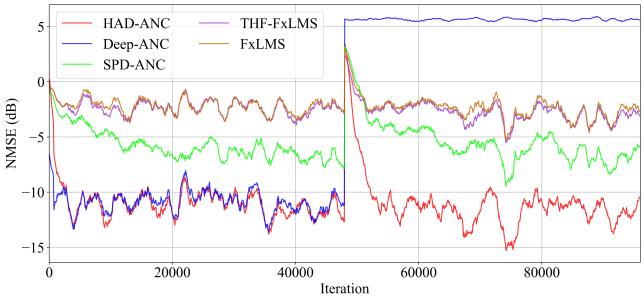

Figure 9: *NMSE of ANC systems where the primary path is inverted during simulation on the babble noise with* $\eta^2 = 0.1$.

indicating that the primary path error estimated through two training steps proposed in Subsection 3.1 successfully estimated the actual value. Figure 8 shows the differences in noise attenuation between the adaptive filter- and DNN-based ANC systems in the frequency-domain. In the adaptive filter-based ANC systems, such as the FxLMS, and THF-FxLMS, noise attenuation occurs only in the low-frequency band. In contrast, the HAD-ANC and DNN-based ANC systems performed outstandingly in wideband noise reduction. In the case of the HAD-ANC, we assume that the combined structure of the GCRN and NFBLMS successfully overcomes the limitations of adaptive filtering.

5.1. Simulation of abrupt path change

Figure 9 shows the NMSE when the abrupt path change occurs. We employed babble noise in this experiment, and η^2 was set to 0.1. We initialized the IRs of the primary and secondary paths with RT60 of 0.2 s. Following [12], we changed the primary path by multiplying -1 at the beginning of the 48,000th iteration. In the deep ANC, which attenuates noise with fixed parameters, the algorithm diverges when the primary path changes. Other ANC systems reduced the noise by converging the adaptive filter with a changed IR. The HAD-ANC generated the control signal with the GCRN of fixed parameters, similar to the deep ANC, but outperformed the others because the NFBLMS could track the paths change and deep learning model was used to estimate the nonlinearity of the loudspeaker and secondary path.

6. Conclusion

In this study, we proposed the HAD-ANC comprising the NFBLMS and GCRNs that two GCRNs forced the NFBLMS to estimate the primary path. $GCRN_1$ estimated the IRs of the loudspeaker and secondary path, whereas $GCRN_2$ estimated the inverse of the former while minimizing the residual error. In experimental results, the HAD-ANC achieved wideband noise attenuation and involved robustness for the primary path changing similar to the DNN- and adaptive filter-based ANC, respectively. However, our study used a fixed room condition and fixed time delay compensation strategy, which may not be applicable in more general scenarios. Future work could address this limitation by developing a time-alignment calculator such as the generalized cross-correlation phase transform.

7. Acknowledgment

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. 2021-0-00456, Development of Ultra-high Speech Quality Technology for Remote Multi-speaker Conference System)

8. References

[1] G. C. Goodwin, E. I. Silva, and D. E. Quevedo, "Analysis and design of networked control systems using the additive noise model methodology," *Asian J. Control*, vol. 12, pp. 443–459, 2010.

[2] S. M. Kuo and D. R. Morgan, "Active noise control: a tutorial review," *Proc. IEEE*, vol. 87, no. 6, pp. 943–973, 1999.

[3] W. Klippel, "Dynamic measurement and interpretation of the nonlinear parameters of electrodynamic loudspeakers," *AES: J. Audio Eng. Soc.*, vol. 38, no. 12, pp. 944–955, 1990.

[4] M. H. Costa, J. C. M. Bermudez, and N. J. Bershad, "Stochastic analysis of the filtered-X LMS algorithm in systems with nonlinear secondary paths," *IEEE Trans. Signal Process.*, vol. 50, no. 6, pp. 1327–1342, 2002.

[5] S. Ghasemi, R. Kamil, and M. H. Marhaban, "Nonlinear THF-FxLMS algorithm for active noise control with loudspeaker nonlinearity," *Asian J. Control*, vol. 18, pp. 502–513, 2016.

[6] S. M. Kuo and H.-T. Wu, "Nonlinear adaptive bilinear filters for active noise control systems," *IEEE Trans. Circuits Syst. I: Regul. Pap.*, vol. 52, pp. 617–624, 2005.

[7] J.-M Song and P. Park, "An optimal variable step-size affine projection algorithm for the modified filtered-x active noise control," *Signal Process.*, vol. 114, pp. 100–111, 2015.

[8] L. Luo, J. Sun, B. Huang, and D. D. Quoc, "Efficient combination of feedforward and feedback structures for nonlinear narrowband active noise control," *Signal Process.*, vol. 128, pp. 494–503, 2016.

[9] X. Tian, J. Huang, X. Feng, and Y. Shen, "An intermittent FxLMS algorithm for active noise control systems with saturation nonlinearity," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 30, pp. 2347–2356, 2022.

[10] H. Zhang and D.L. Wang, "Deep ANC: a deep learning approach to active noise control," *Neural Netw.*, vol. 141, pp. 1–10, 2021.

[11] H. Zhang, A. Pandey, and D.L. Wang, "Attentive recurrent network for low-latency active noise control," in *Proc. Interspeech*, 2022, pp. 956–960.

[12] D. Chen, L. Cheng, D. Yao, J. Li, and Y. Yan, "A secondary path-decoupled active noise control algorithm based on deep learning," *IEEE Signal Proc. Lett.*, vol. 29, pp. 234–238, 2022.

[13] K. Tan and D.L. Wang, "Complex spectral mapping with a convolutional recurrent network for monaural speech enhancement," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 6865–6869.

[14] E. Ferrara, "Fast implementations of LMS adaptive filters," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 28, no. 4, pp. 474–475, 1980.

[15] J. J. Shynk, "Frequency-domain and multirate adaptive filtering," *IEEE Signal Proces. Mag.*, vol. 9, no. 1, pp. 14–37, 1992.

[16] K. Tan and D.L. Wang, "Learning complex spectral mapping with gated convolutional recurrent networks for monaural speech enhancement," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 380–390, 2020.

[17] X. Kong and S. M. Kuo, "Study of causality constraint on feedforward active noise control systems," *IEEE Trans. on Circuits and Syst.*, vol. 46, no. 2, pp. 183–186, 1999.

[18] P. N. Samarasinghe, W. Zhang, and T. D. Abhayapala, "Recent advances in active noise control inside automobile cabins: toward quieter cars," *IEEE Signal Process. Mag.*, vol. 33, no. 6, pp. 61–73, 2016.

[19] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Comput.*, vol. 1, no. 2, pp. 270–280, 1989.

[20] A. Varga and H. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: a database and an experiment to study the effect of additive noise on speech recognition systems," *Speech commun.*, vol. 12, no. 3, pp. 247–251, 1993.

[21] S.-K. Lau and S.-K. Tang, "Sound fields in a slightly damped rectangular enclosure under active control," *J. Sound Vib.*, vol. 238, pp. 637–660, 2000.

[22] W. Klippel, "Tutorial: loudspeaker nonlinearities - causes, parameters, symptoms," *AES: J. Audio Eng. Soc.*, vol. 54, pp. 907–939, 2006.

[23] J. Chen and D.L. Wang, "Long short-term memory for speaker generalization in supervised speech separation," in *Proc. Interspeech*, 2016, pp. 3314–3318.

[24] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, vol. 70, pp. 933–941.

[25] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[26] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," in *Proc. Int. Conf. Learn. Represent.*, 2016.

[27] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," in *Proc. Int. Conf. Learn. Representations*, 2018.