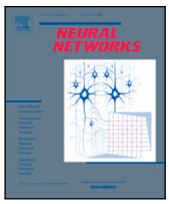
Neural Networks 180 (2024) 106687

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet



Full Length Article

GFANC-RL: Reinforcement Learning-based Generative Fixed-filter Active Noise Control

Zhengding Luo [a,1], Haozhe Ma [b,1], Dongyuan Shi [a,*], Woon-Seng Gan [a]

[a] Digital Signal Processing Lab, School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
[b] School of Computing, National University of Singapore, Singapore

ARTICLE INFO

Keywords:
Active Noise Control
Generative Fixed-filter ANC
Reinforcement Learning
Convolutional Neural Network

ABSTRACT

The recent Generative Fixed-filter Active Noise Control (GFANC) method achieves a good trade-off between noise reduction performance and system stability. However, labelling noise data for training the Convolutional Neural Network (CNN) in GFANC is typically resource-consuming. Even worse, labelling errors will degrade the CNN's filter-generation accuracy. Therefore, this paper proposes a novel Reinforcement Learning-based GFANC (GFANC-RL) approach that omits the labelling process by leveraging the exploring property of Reinforcement Learning (RL). The CNN's parameters are automatically updated through the interaction between the RL agent and the environment. Moreover, the RL algorithm solves the non-differentiability issue caused by using binary combination weights in GFANC. Simulation results demonstrate the effectiveness and transferability of the GFANC-RL method in handling real-recorded noises across different acoustic paths.[2]

1. Introduction

The proliferation of industrial equipment has significantly exacerbated acoustic noise issues (George & Panda, 2013; Kajikawa, Gan, & Kuo, 2012; Kuo, Mitra, & Gan, 2006; Pan, Chen, & Benesty, 2022; Zhang, Elliott, & Cheer, 2021). Due to the compact size and effectiveness of controlling low-frequency noises, Active Noise Control (ANC) has gained intensive development in various commercial products, including windows, headphones, headrests, vehicles, etc (Elliott & Nelson, 1993; Fuller & von Flotow, 1995; Hansen, 2002; Jung, Elliott, & Cheer, 2019; Yang et al., 2022). The ANC technology utilizes a secondary source to generate an anti-noise wave with the same amplitude but opposite phase to suppress the disturbance (Han & Qiu, 2007; Shi, Du, & Wu, 2022; Yang, Guo, & Yang, 2020). To cope with variations in the acoustic environment and primary noise, adaptive ANC algorithms, such as the Filtered-reference Least Mean Square (FxLMS) algorithm, are widely deployed in ANC systems. The FxLMS algorithm can adaptively adjust the control filter coefficients to minimize the squared error signal (Benois, Roden, Blau, & Doclo, 2022; Gupta et al., 2022; Lam, Shi, Gan, Elliott, & Nishimura, 2020; Schumacher, Krüger, Jeub, Vary, & Beaugeant, 2011).

However, adaptive ANC algorithms have inherent limitations including slow convergence speeds, inadequate tracking capabilities, and a potential for divergence (Chang, Chuang, Kuo, & Lin, 2022; Pawełczyk, 2002; Sun, Zhang, Abhayapala, & Samarasinghe, 2022; Zhang et al., 2020). To overcome these limitations, many commercial ANC products have adopted fixed-filter methods, whose control filters are pre-trained rather than adaptively updated throughout noise control (Iotov, Nørholm, Belyi, Dyrholm, & Christensen, 2022; Luo, Shi, Ji, & Gan, 2022; Xie et al., 2024). Although fixed-filter ANC methods offer fast response speeds and good robustness, they are only suitable for particular types of noise, resulting in mediocre performance for other types (Shi, Gan, Lam, & Wen, 2020; Sun, Jin, Abhayapala, & Samarasinghe, 2024; Xiao & Doclo, 2024). Therefore, we intend to leverage the powerful learning capabilities of neural networks to improve traditional fixed-filter methods in handling various types of noise (Luo, Shi, Gan, Huang, & Zhang, 2023).

Some recent ANC methods leveraged neural networks to replace the control filter to predict the control signal (Mostafavi & Cha, 2023). Park, Patterson, and Baum (2019) investigated the application of Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) architectures to predict the control signal. Zhang et al. (Zhang & Wang, 2021, 2023) employed a Convolutional Recurrent Network (CRN) to estimate the real and imaginary spectrograms of the control signal. However, due to the high computational complexity of neural networks, directly using neural networks as the control filter may introduce huge processing latency (Shi, Gan, Shen, Luo, & Ji, 2024; Toyooka

* Corresponding author.
E-mail address: dongyuan.shi@ntu.edu.sg (D. Shi).
[1] Contributed equally to this work.
[2] The code will be accessible at https://github.com/Luo-Zhengding/GFANC-RL.

https://doi.org/10.1016/j.neunet.2024.106687
Received 11 May 2024; Received in revised form 14 August 2024; Accepted 30 August 2024
Available online 5 September 2024

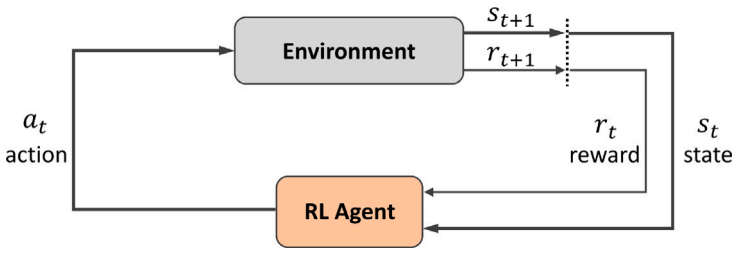


Fig. 1. The interaction between the RL agent and the environment.

& Kajikawa, 2023; Zhang, Pandey, & Wang, 2022). This latency might violate the causality requirements of ANC systems and result in less effective noise control. To address the processing latency, Shi, Lam, Ooi, Shen, and Gan (2022) proposed a Selective Fixed-filter Active Noise Control (SFANC) method. This SFANC method selects suitable pre-trained control filters for different noises via a CNN implemented in a co-processor, which ensures that the real-time noise control will not be affected by the CNN's processing delay.

As an enhancement of the SFANC method (Shi, Lam, et al., 2022), a Generative Fixed-filter Active Noise Control (GFANC) method (Luo, Shi, Shen, Ji, & Gan, 2023) has been proposed recently to generate appropriate control filters for different noises. Unlike SFANC, which selects from a limited number of pre-trained control filters, GFANC employs a CNN to generate various control filters by the weighted combination of sub control filters. Nevertheless, this approach depends on supervised learning to train the CNN, known as the GFANC-Supervised method. Hence, labelling noise data is the prerequisite of this method and is typically resource-intensive. Even worse, labelling errors may degrade the CNN's filter-generation accuracy and impair the noise reduction performance (Luo, Shi, Gan & Huang, 2024; Luo, Shi, Shen, Ji, & Gan, 2024).

To improve the flexibility of training the CNN with unlabelled noise data in the GFANC method, we can consider the use of Reinforcement Learning (RL) techniques (Arulkumaran, Deisenroth, Brundage, & Bharath, 2017; Ma, Sima, Vo, Fu, & Leong, 2024; Sutton & Barto, 2018). RL is a sub-field of machine learning that focuses on instructing an RL agent how to make decisions by interacting with the environment (Ma, Vo, & Leong, 2024; Mnih et al., 2015). The RL agent aims to maximize its cumulative rewards over time. The interaction process is illustrated in Fig. 1, where the RL agent receives reward signals from the environment to evaluate its actions. Such interactions provide continuous feedback, supporting the RL agent's learning and enhancing its decision-making abilities (Hessel et al., 2018; Ma, Vo, & Leong, 2023).

Therefore, this paper provides a novel perspective to analyse GFANC and leverage RL technology to improve its flexibility and practicality. The contributions of the paper are listed as

1. A novel GFANC-RL method that leverages an RL algorithm to train the CNN model is proposed, which aims to omit the complicated labelling process and improve the CNN's exploration abilities.
2. This paper initially formulates the GFANC method as a Markov decision process (MDP) from a decision-making standpoint, which provides the theoretical foundations to employ RL algorithms.
3. The RL algorithm not only uses unlabelled noise data to train the CNN, but also solves the non-differentiable problem caused by the use of binary combination weights in GFANC.
4. Simulation results demonstrate the effectiveness and good transferability of the GFANC-RL method in dealing with different real noises and acoustic paths.

The rest of this paper is organized as follows: Section 2 outlines the existing challenges associated with the GFANC-Supervised approach. To tackle the challenges, the proposed GFANC-RL method is introduced in Section 3. Section 4 presents numerical simulations to evaluate the effectiveness of the GFANC-RL method. Finally, the conclusion is provided in Section 5.

2. Problem statement

This section briefly introduces the ANC system and the GFANC method. The limitations of the GFANC-Supervised approach are also discussed.

2.1. Active Noise Control (ANC)

Fig. 2 illustrates the block diagram of a feedforward ANC system, where the control filter $\mathbf{w}(n)$ processes the reference signal $\mathbf{x}(n)$ sampled from the primary noise and generates the control signal $y(n)$. The control signal can be computed as

$$y(n) = \mathbf{x}^{\mathrm{T}}(n)\mathbf{w}(n), \tag{1}$$

where n denotes the sample index, and T represents the transpose operation. Subsequently, the control signal passes through the secondary path to cancel the disturbance $d(n)$. The obtained error signal $e(n)$ can be expressed as

$$\begin{aligned} e(n) &= d(n) - y(n) * s(n), \\ &= d(n) - \mathbf{x}^{\mathrm{T}}(n)\mathbf{w}(n) * s(n), \end{aligned} \tag{2}$$

where $*$ and $s(n)$ denote the convolution operation and the impulse response of the secondary path, respectively.

The optimal control filter of the ANC system is usually defined as the filter that can minimize the mean square of the error signal:

$$\mathbf{w}^o = \arg\min_{\mathbf{w}} \mathbb{E}\left[e^2(n)\right], \tag{3}$$

where $\mathbb{E}[\cdot]$ stands for the expectation of the argument. Adaptive ANC algorithms, such as the FxLMS algorithm (Kuo & Morgan, 1999), have been developed to obtain this optimal solution. Their high computational complexity and poor stability significantly impair their deployments in practical ANC systems. In contrast, the fixed-filter ANC method has been applied in many ANC products due to its low computational complexity and high robustness (Luo, Shi, Ji, & Gan, 2022). However, it exhibits poor noise reduction performance when dealing with noise that differs much from the training noise of the control filter. A recent method named Generative Fixed-filter ANC (GFANC) (Luo, Shi, Shen, et al., 2023) has been proposed to tackle this problem. It generates suitable control filters for different primary noises by a CNN and achieves a good noise reduction performance.

2.2. The GFANC method

As illustrated in Fig. 3, the control filter in the GFANC method (Luo, Shi, Shen, et al., 2023) is the weighted sum of a set of sub control filters as

$$\mathbf{w} = \mathbf{g} \cdot \mathbf{c} = \sum_{i=1}^{M} g_i \mathbf{c}_i, \tag{4}$$

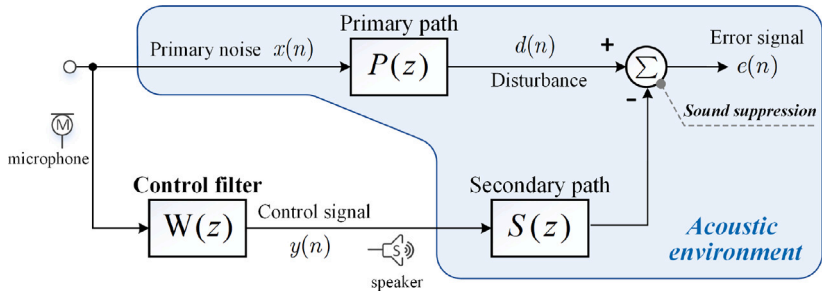


Fig. 2. Block diagram of a feedforward ANC system.

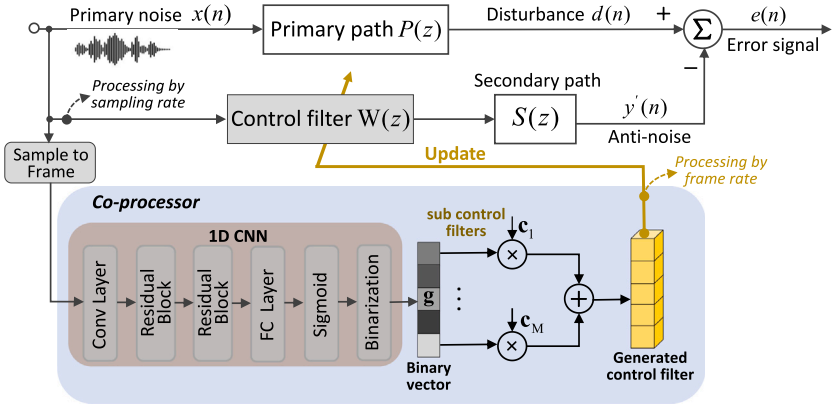
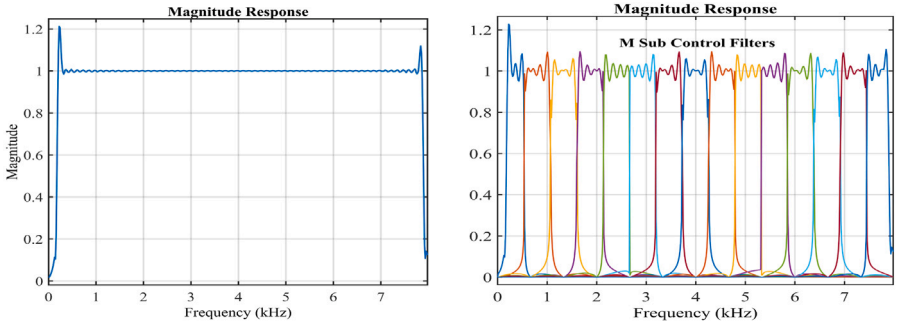


Fig. 3. Block diagram of the GFANC method (Luo, Shi, Shen, et al., 2023). Different control filters are generated for different primary noises through the weighted combination of sub control filters.



(a) Pre-trained broadband control filter (b) Sub control filters

Fig. 4. The frequency bandwidths of the pre-trained broadband control filter and its sub control filters in the GFANC method.

where sub control filters matrix is given by

$$\mathbf{c} = \left[\mathbf{c}_1 \dots \mathbf{c}_m \dots \mathbf{c}_M\right]^{\mathrm{T}}. \tag{5}$$

$\mathbf{c}_m$ represents the impulse response of the mth sub control filter, which is orthogonal to the others.

To obtain sub control filters, the optimal control filter for a broadband noise, whose frequency band contains our interested components shown in Fig. 4(a), is firstly derived using the FxLMS algorithm. Subsequently, it is decomposed into M orthogonal sub bands as the desired sub control filters (Luo, Shi, Shen, et al., 2023), as shown in Fig. 4 (b). The binary weight vector $\mathbf{g} = \left[g_1 \dots g_m \dots g_M\right]$ for combining these sub control filters is the output of a one-dimensional (1D) CNN, whose input is the reference signal vector.

During real-time noise control, the co-processor runs the 1D CNN at the frame rate while the real-time controller performs noise control at the sampling rate in parallel. The efficient coordination between the co-processor and real-time controller can achieve delayless noise control. Additionally, different from adaptive algorithms, the GFANC method does not require the error signal to update the control filter during noise control, which minimizes the risk of divergence and increases its stability (Luo, Shi, Ji, Shen, & Gan, 2024).

2.3. Limitations of the GFANC-supervised method

Fig. 5 exhibits the training process of the 1D CNN in the previous GFANC method (Luo, Shi, Shen, et al., 2023). Since the method is based on supervised learning, it is denoted as the GFANC-Supervised method in this paper. The training dataset has N samples and is denoted as $\{\mathbf{x}_t, \mathbf{g}'_t\}_{t=1}^{N}$, where the binary vector $\mathbf{g}'_t = \left[g'_{t,1} \dots g'_{t,m} \dots g'_{t,M}\right]$ ($g'_{t,m} \in \{0, 1\}$) represents the label of the noise $\mathbf{x}_t$. The output of the 1D CNN is denoted as $\mathbf{g}_t = \left[g_{t,1} \dots g_{t,m} \dots g_{t,M}\right]$ given the noise $\mathbf{x}_t$. To minimize the difference between $\mathbf{g}'_t$ and $\mathbf{g}_t$, Binary Cross Entropy (BCE) loss function (Read, Pfahringer, Holmes, & Frank, 2011) is utilized in the

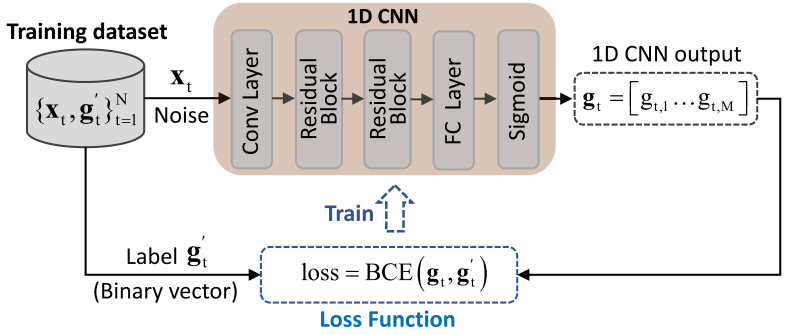


Fig. 5. The training process of the 1D CNN in the GFANC-Supervised method (Luo, Shi, Shen, et al., 2023).

training process as

$$\begin{aligned} \mathcal{L}oss &= \mathbf{BCE}\left(\mathbf{g}_t, \mathbf{g}_t'\right), \\ &= -\frac{1}{N}\frac{1}{M}\sum_{t=1}^{N}\sum_{m=1}^{M}\left[g_{t,m}' \cdot \log g_{t,m} + (1-g_{t,m}') \cdot \log\left(1-g_{t,m}\right)\right], \end{aligned} \tag{6}$$

where $\log(\cdot)$ stands for the logarithm function.

2.3.1. Labelling problem

In the GFANC-Supervised approach, training the 1D CNN is supervised and heavily relies on the accuracy of noise labels. These labels can be assigned manually based on frequency band matching (Shi et al., 2020) or generated automatically through an LMS-based adaptive labelling mechanism (Luo, Shi, Shen, et al., 2023). However, both methods depend on manually selected hyperparameters, which can introduce biases and errors if not properly set. As a result, the labelling process is not only resource-consuming but also prone to erroneous labels that can negatively impact the training results.

2.3.2. Non-differentiability issue

Furthermore, due to the use of binary combination weights, if the error signal is directly used as the training loss without using noise labels, the derivatives cannot be backpropagated to update the CNN's parameters. Therefore, we aim to leverage RL techniques to omit the labelling process and address the non-differentiability issue. Additionally, the CNN's parameters can be updated through interactions between the RL agent and the environment, enhancing the CNN's exploration capabilities.

3. The proposed GFANC-RL method

Fig. 6 presents the training process of the 1D CNN in the proposed GFANC-RL method. In this approach, the noise cancellation part is considered as the environment, while the 1D CNN and its parameter updates are considered as the RL agent. The 1D CNN's parameters within this agent are updated by an RL algorithm to maximize the reward, which is the noise reduction level (NR), using unlabelled noise data. Moreover, the RL algorithm addresses the non-differentiability problem arising from the use of binary combination weights. Once the RL algorithm converges, the 1D CNN, loaded with its optimal parameters, will be employed for real-time noise control in the GFANC system. This section offers a detailed overview of how the RL algorithm is employed to train the 1D CNN.

3.1. Formulating GFANC with Markov Decision Process

The GFANC approach can be regarded as a Markov Decision Process (MDP), whose four critical components $\langle S, A, T, R\rangle$ are defined as follows: The state space S is the set of all possible states in the environment, each state in GFANC refers to a noise frame $\mathbf{x}_t$ at time t. The action space A is the set of all possible actions that the agent can take, where each action refers to a binary combination weight vector $\mathbf{g}_t = \left[g_{t,1} \cdots g_{t,m} \cdots g_{t,M}\right]$ output by the 1D CNN. Since the next noise frame is independent of the current action in GFANC, the transition probabilities of transitioning from one state to another can be described as $T(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{g}_t) = T(\mathbf{x}_{t+1}|\mathbf{x}_t)$. Furthermore, $R(\mathbf{x}_t, \mathbf{g}_t)$ is the reward function that maps a state–action pair to a real number, representing the immediate reward received by the agent after taking a specific action in a specific state. In this paper, the noise reduction level (NR) is defined as the reward value r_t to assess the noise reduction performance:

$$r_t = R(\mathbf{x}_t, \mathbf{g}_t) = 10\log_{10}\frac{\sum_{n=1}^{L} d_t^2(n)}{\sum_{n=1}^{L} e_t^2(n)}, \tag{7}$$

where $d_t(n)$ and $e_t(n)$ represent the disturbance and error signal corresponding to the noise frame $\mathbf{x}_t$, and L denotes the length of the signal.

As introduced in Section 2.2, the control filter is generated from the inner product between the weight vector $\mathbf{g}_t$ and the sub control filter bank $\mathbf{c}$ as

$$\mathbf{w}_t = \mathbf{g}_t \cdot \mathbf{c} = \sum_{m=1}^{M} g_{t,m}\mathbf{c}_m. \tag{8}$$

According to (2), the error signal given the noise vector $\mathbf{x}_t$ and the control filter $\mathbf{w}_t$ is obtained from

$$\begin{aligned} e_t(n) &= d_t(n) - \mathbf{x}_t^{\mathrm{T}}(n) \cdot \mathbf{w}_t * s(n), \\ &= d_t(n) - \mathbf{x}_t^{\mathrm{T}}(n) \cdot \mathbf{g}_t \cdot \mathbf{c} * s(n). \end{aligned} \tag{9}$$

By substituting (9) into (7), the reward value can be rewritten as

$$r_t = R(\mathbf{x}_t, \mathbf{g}_t) = 10\log_{10}\frac{\sum_{n=1}^{L} d_t^2(n)}{\sum_{n=1}^{L}\left[d_t(n) - \mathbf{x}_t^{\mathrm{T}}(n) \cdot \mathbf{g}_t \cdot \mathbf{c} * s(n)\right]^2}, \tag{10}$$

which represents that given each state (the noise frame $\mathbf{x}_t$) and each action (the weight vector $\mathbf{g}_t$), there is a corresponding reward value r_t.

Based on the MDP model $\langle S, A, T, R\rangle$, a policy $\pi : S \rightarrow A$ is a mapping function from state space to action space. A stochastic policy approach is employed in this paper, where the policy π is a probability distribution, and the weight vector $\mathbf{g}_t \sim \pi(\cdot|\mathbf{x}_t)$. Specifically, the policy π can represent the 1D CNN in the GFANC system. The agent's goal is to learn a policy π that maximizes the expected return, expressed as $\mathbb{E}_{\mathbf{g}_t\sim\pi(\cdot|\mathbf{x}_t)}(G_t)$. The return G_t is defined as the discounted cumulative reward over time:

$$G_t = R(\mathbf{x}_t, \mathbf{g}_t) + \gamma R(\mathbf{x}_{t+1}, \mathbf{g}_{t+1}) + \cdots = R(\mathbf{x}_t, \mathbf{g}_t) + \sum_{\tau=t+1}^{\infty}\gamma^{\tau-t}R(\mathbf{x}_\tau, \mathbf{g}_\tau), \tag{11}$$

where $\gamma \in [0, 1]$ is a discount factor to balance the importance of future rewards. It is worth noting that since the GFANC method pays more attention to the current reward, the γ can be set to 0. Hence, G_t can be simplified as the one-step immediate reward:

$$G_t = R(\mathbf{x}_t, \mathbf{g}_t), \tag{12}$$

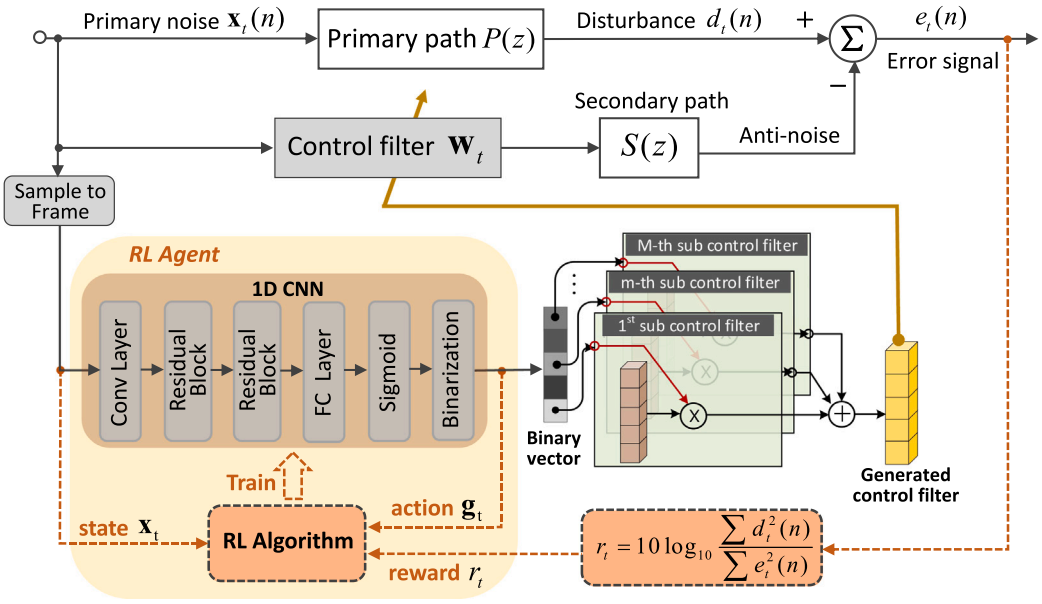


Fig. 6. The training process of the 1D CNN in the GFANC-RL method leverages the exploration ability of RL. This allows the 1D CNN's parameters to be updated as the RL agent interacts with the environment.

and the optimal policy π^* can be derived as

$$\pi^* = \arg\max_{\pi} \mathop{\mathbb{E}}_{\mathbf{g}_t\sim\pi(\cdot|\mathbf{x}_t)}(G_t) = \arg\max_{\pi} \mathop{\mathbb{E}}_{\mathbf{g}_t\sim\pi(\cdot|\mathbf{x}_t)}[R(\mathbf{x}_t,\mathbf{g}_t)]. \tag{13}$$

To solve the MDP-modeled GFANC problem, RL algorithms can be used to learn a policy for making optimal decisions within the environment. This learning process involves interactions between the agent and the environment to determine the optimal parameters for the 1D CNN. Unlike supervised learning, RL allows for training the 1D CNN without the need for noise labels, greatly simplifying the training process (Haarnoja, Tang, Abbeel, & Levine, 2017). Additionally, since RL algorithms are not constrained by differentiability requirements (Mnih et al., 2016), they can be easily employed in GFANC systems that use binary combination weights.

3.2. The RL algorithm in GFANC-RL

Since the acoustic environment in real ANC applications is difficult to model accurately, we use a model-free RL algorithm based on the soft actor-critic (SAC) algorithm (Haarnoja, Zhou, Abbeel & Levine, Sergey, 2018) to solve the MDP-modeled GFANC problem. Moreover, the excellent exploration ability of the SAC algorithm also can improve the performance of the GFANC approach. The SAC algorithm typically has two components: the actor represents the policy and is responsible for taking actions in the environment, and the critic evaluates the actions taken by the actor based on the expected rewards (Fujimoto, Hoof, & Meger, 2018; Silver et al., 2014). The optimal policy (the 1D CNN in GFANC) can be learned by alternating between two steps: policy evaluation and policy improvement (Haarnoja, Zhou, Hartikainen, et al., 2018; Ma, Luo, Vo, Sima, & Leong, 2024).

3.2.1. Framework of the RL algorithm

In the policy evaluation step, we assess the learned policy using the evaluation function (critic). The critic estimates the expected return of taking action $\mathbf{g}_t$ at state $\mathbf{x}_t$ following the policy π, defined as a Q-function:

$$Q^{\pi}(\mathbf{x}_t,\mathbf{g}_t) = \mathop{\mathbb{E}}_{\mathbf{g}_t\sim\pi(\cdot|\mathbf{x}_t)}(G_t) = \mathop{\mathbb{E}}_{\mathbf{g}_t\sim\pi(\cdot|\mathbf{x}_t)}[R(\mathbf{x}_t,\mathbf{g}_t)]. \tag{14}$$

In the policy improvement step, we update the policy π to maximize the objective function, which is defined as

$$J(\pi) = Q^{\pi}(\mathbf{x}_t,\mathbf{g}_t) + \alpha\mathcal{H}(\pi(\cdot|\mathbf{x}_t)), \tag{15}$$

where $\mathcal{H}(\cdot)$ is the entropy of the policy given by

$$\mathcal{H}(\pi(\cdot|\mathbf{x}_t)) = \mathop{\mathbb{E}}_{\mathbf{g}_t\sim\pi(\cdot|\mathbf{x}_t)}\left[-\log\pi(\mathbf{g}_t|\mathbf{x}_t)\right]. \tag{16}$$

The higher entropy indicates more significant uncertainty regarding optimal actions, encouraging the agent to explore a broader range of actions (Schulman, Levine, Abbeel, Jordan, & Moritz, 2015; Wu, Mansimov, Grosse, Liao, & Ba, 2017). The temperature parameter α plays a crucial role in balancing the impact of the entropy term relative to the critic value. It is automatically maintained at a reasonable level following the methodology proposed by Haarnoja, Zhou, Hartikainen, et al. (2018).

Substituting (14) and (16) into (15), the objective function of the policy is derived as

$$J(\pi) = \mathop{\mathbb{E}}_{\mathbf{g}_t\sim\pi(\cdot|\mathbf{x}_t)}\left[R(\mathbf{x}_t,\mathbf{g}_t) - \alpha\log\pi(\mathbf{g}_t|\mathbf{x}_t)\right]. \tag{17}$$

This function aims to maximize the noise reduction levels by adjusting the parameters of the 1D CNN while promoting extensive explorations. Based on the objective function, the optimal policy is given by

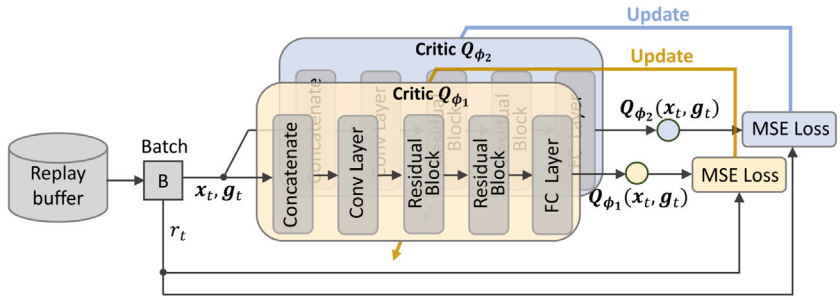
$$\begin{aligned}\pi^* &= \arg\max_{\pi} J(\pi),\\ &= \arg\max_{\pi} \mathop{\mathbb{E}}_{\mathbf{g}_t\sim\pi(\cdot|\mathbf{x}_t)}\left[R(\mathbf{x}_t,\mathbf{g}_t) - \alpha\log\pi(\mathbf{g}_t|\mathbf{x}_t)\right].\end{aligned} \tag{18}$$

3.2.2. Optimization of the RL algorithm

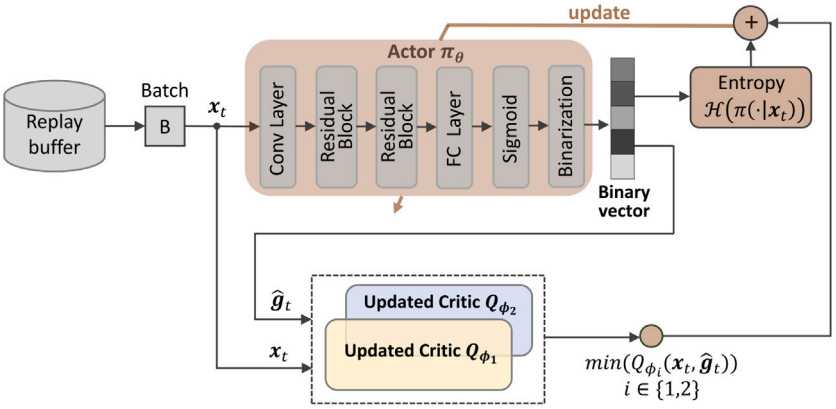
In the SAC-based RL algorithm, the critic and actor are parameterized by $Q_\phi(\mathbf{x}_t,\mathbf{g}_t)$ and $\pi_\theta(\mathbf{x}_t)$, respectively. The parameters of the critic and actor are denoted as ϕ and θ. In the optimization of the RL algorithm, we adopted an off-policy learning strategy, where the agent stores past experiences in a replay buffer and samples from this buffer during the learning process. After each interaction step between the agent and the environment, the collected transition $(\mathbf{x}_t,\mathbf{g}_t,r_t)$ is stored in the replay buffer $\mathcal{D}$.

The schematic diagram of the parameter updates for the actor and critic is illustrated in Fig. 7. Inspired by Fujimoto et al. (2018), we used two critic functions to alleviate over-estimation and speed up training. The two critic functions are parameterized with parameters $\phi_i, i\in\{1,2\}$, which are updated independently as shown in Fig. 7(a). The parameters of the critic are updated by minimizing the expectation of the mean squared errors (MSE) between the predicted Q-values and the received rewards in the replay buffer:

$$\mathcal{L}(\phi_i) = \mathop{\mathbb{E}}_{(\mathbf{x}_t,\mathbf{g}_t,r_t)\sim\mathcal{D}}\left[\frac{1}{2}\left(Q_{\phi_i}(\mathbf{x}_t,\mathbf{g}_t) - r_t\right)^2\right], i\in\{1,2\}, \tag{19}$$



(a) Update of critic parameters



(b) Update of actor parameters

Fig. 7. The parameter update mechanisms for the critic and actor in the RL algorithm.

where $(\mathbf{x}_t, \mathbf{g}_t, r_t) \sim D$ indicates that the transitions are drawn from the replay buffer D. Thus, the critic parameters ϕ_i can be updated via the gradient descent method with a learning rate λ_ϕ:

$$\phi_i \leftarrow \phi_i - \lambda_\phi \nabla_{\phi_i} \mathcal{L}(\phi_i), i \in \{1,2\}. \tag{20}$$

Additionally, as shown in Fig. 7(b), the policy parameters θ can be updated based on the objective function:

$$J(\pi_\theta) = \mathop{\mathbb{E}}_{\mathbf{x}_t \sim D,\ \hat{\mathbf{g}}_t \sim \pi_\theta(\cdot|\mathbf{x}_t)} \left[Q_\phi(\mathbf{x}_t, \hat{\mathbf{g}}_t) - \alpha \log \pi_\theta(\hat{\mathbf{g}}_t|\mathbf{x}_t) \right], \tag{21}$$

where $\mathbf{x}_t$ is drawn from the replay buffer D and $\hat{\mathbf{g}}_t$ is sampled from $\pi_\theta(\cdot|\mathbf{x}_t)$. $Q_\phi(\mathbf{x}_t, \hat{\mathbf{g}}_t)$ is the minimum of the two critic function values:

$$Q_\phi(\mathbf{x}_t, \hat{\mathbf{g}}_t) = \min_{i=1,2} Q_{\phi_i}(\mathbf{x}_t, \hat{\mathbf{g}}_t). \tag{22}$$

Therefore, the policy parameters θ can be updated by the gradient ascent method with a learning rate λ_θ:

$$\theta \leftarrow \theta + \lambda_\theta \nabla_\theta J(\pi_\theta). \tag{23}$$

The RL algorithm alternates between collecting experience with the current policy and updating the policy using batches sampled from the replay buffer. Finally, the optimized policy parameters θ^* are used as the parameters of the 1D CNN in the GFANC-RL method for real-time noise control.

3.3. Real-time noise control

The training of the 1D CNN is carried out offline using the RL algorithm. Once trained, the 1D CNN equipped with its optimized parameters θ^* is utilized for online noise control, as outlined in Section 2.2. During real-time noise control, the weight vector $\mathbf{g}$ for each noise frame $\mathbf{x}$ is calculated as

$$\mathbf{g} = CNN(\mathbf{x}; \theta^*) = \left[g_1 \cdots g_m \cdots g_M \right], \tag{24}$$

where $CNN(\cdot)$ denotes the operations performed by the 1D CNN. Subsequently, this weight vector $\mathbf{g}$ is employed to combine sub control filters to generate a control filter within the co-processor.

Overall, based on the RL algorithm, the 1D CNN's optimal parameters are obtained through the interaction between the agent and the environment during training. The training process does not need noise labels, and there is no requirement for GFANC's differentiability. Furthermore, the reward function $R(\mathbf{x}_t, \mathbf{g}_t)$ can be flexibly adjusted according to specific tasks. After training, the efficient coordination between the co-processor and real-time controller can achieve delayless noise control in practical scenarios. The method allows for using a batch-processing co-processor, such as a laptop, to run the trained 1D CNN, while the real-time controller handles immediate processing to maintain low latency.

4. Numerical simulations

In this section, the proposed GFANC-RL approach's effectiveness is evaluated through some numerical simulations. In terms of experimental setup, the number of sub control filters M is 15. The sampling rate and control filter's length are set to 16 kHz and 1,024 taps, respectively. In Sections 4.1 to 4.3, the synthetic ANC system utilizes bandpass filters for the primary path and secondary path, with a frequency range of 20 to 7980 Hz. Section 4.4 employs acoustic paths measured from realistic ANC systems.

A synthetic noise dataset[3] containing 80,000 noise instances is used for the training the 1D CNN. Noise instances are generated by applying bandpass filters to white noise, where the filters' frequency bands are randomly selected within the range of 20 to 7980 Hz. Each noise

[3] The link of the synthetic noise dataset: https://drive.google.com/file/d/1hs7_eHITxL16HeugjQoqYFTs-Cm7J-Tq/view?pli=1.

Table 1
NR values (dB) for three real noises obtained by different ANC algorithms.

ANC algorithm	Traffic noise	Aircraft noise	Drill noise
GFANC-RL	14.4	13.3	9.2
GFANC-Supervised (Luo, Shi, Shen, et al., 2023)	12.6	12.4	9.1
SFANC (Shi, Lam, et al., 2022)	11.1	10.4	8.0
FxLMS (Kuo & Morgan, 1999)	7.1	4.2	3.9

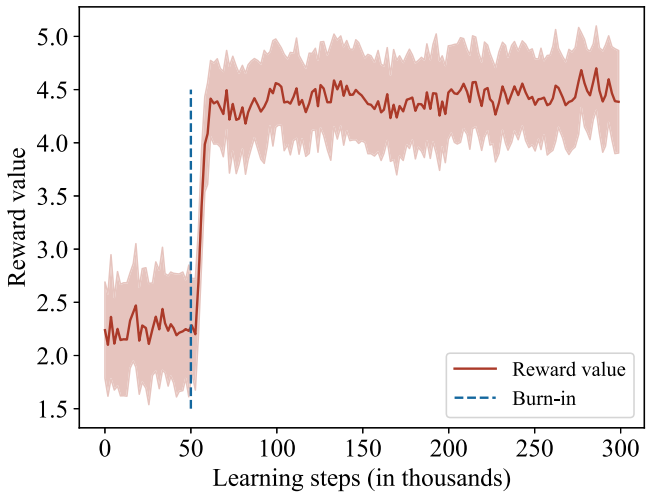


Fig. 8. The reward curve of the RL algorithm.

instance with a duration of 1 second is considered as a reference signal. To enhance robustness, additional white noise is added to the reference signal to achieve a signal-to-noise ratio (SNR) of 5 dB during training. It is worth noting that no data labels are used in the training phase.

4.1. Effectiveness of the RL algorithm

Fig. 8 depicts the reward curve of the RL algorithm during training the 1D CNN. The figure presents the average rewards per 500 steps and their standard deviation. The blue dashed line marks the end of the burn-in phase, during which the agent randomly generates binary vectors $\mathbf{g}_t$ for $\mathbf{x}_t$ and calculates the corresponding r_t. All random samples $(\mathbf{x}_t, \mathbf{g}_t, r_t)$ are stored in the replay buffer $\mathcal{D}$. Beyond the burn-in phase, the agent begins actual exploration learning, using binary vectors generated from the policy.

As shown in Fig. 8, following the burn-in phase, rewards are sharply increased, primarily due to the off-policy learning approach's high sampling efficiency. This approach allows quick optimization by reusing past experiences stored in the replay buffer. As training progresses, the agent gradually converges towards the optimal policy, with rewards stabilizing at a high level. After all learning steps, the final policy is used as the trained 1D CNN in the GFANC-RL method. Additionally, a detailed discussion about the hyperparameters used in the RL algorithm is introduced in Appendix A.

4.2. Noise control for real noises

After training via the RL algorithm, the GFANC-RL method is used for real-time noise control. The noise control performance is assessed using three real noises: traffic noise, aircraft noise, and drill noise. These real noises are dynamic noises, which are quite different from noise instances in the training dataset. When dealing with these real noises, the GFANC-RL method is compared to the GFANC-Supervised method (Luo, Shi, Shen, et al., 2023), the SFANC method (Shi, Lam, et al., 2022), and the FxLMS algorithm (Kuo & Morgan, 1999) with a step size of 0.0001.

Noise reduction level (NR) is a common metric for evaluating ANC performance. It is the ratio of disturbance signal power to error signal power, as introduced in Eq. (7). The NR values for three real noises obtained by different ANC algorithms are summarized in Table 1. The GFANC-RL method consistently outperforms the compared methods in terms of NR for the three real noises. Specifically, compared to the FxLMS algorithm, it achieves approximately 7 dB, 9 dB, and 5 dB improvements in NR for the three noises, respectively. Furthermore, the NR values of the GFANC-RL method show a moderate increase compared to the GFANC-Supervised method and the SFANC method. This performance enhancement demonstrates the effectiveness of the GFANC-RL method for different real noises.

4.2.1. Comparison with the GFANC-supervised method

The detailed noise reduction performances on the three real noises are illustrated in Figs. 9, 10, 11 and 12. It is observed that GFANC-RL and GFANC-Supervised (Luo, Shi, Shen, et al., 2023) exhibit comparable performance. As shown in Figs. 12, in several seconds of the traffic noise, GFANC-RL outperforms GFANC-Supervised, potentially due to its capacity to mitigate certain biases associated with labelling procedures. For the aircraft noise, GFANC-RL achieves higher NR values than GFANC-Supervised during 1-2 second and 9-10 second, but the situation is reversed in 2-3 second. In addition, for the drill noise, the NR value of GFANC-RL at 8-9 second is slightly improved than GFANC-Supervised, but the performance in other seconds is the same. Overall, GFANC-RL achieves comparable noise reduction performance to GFANC-Supervised while eliminating the costs and errors associated with the noise labelling process.

4.2.2. Comparison with the SFANC method

As introduced in Section 1, GFANC-RL can generate more appropriate control filters by combining sub control filters compared to SFANC (Shi, Lam, et al., 2022). The simulation results in Figs. 9 to 12 verify the superiority of GFANC-RL over SFANC. Both GFANC-RL and SFANC respond rapidly and effectively attenuate the real noises, but GFANC-RL outperforms SFANC by approximately 5 dB in terms of averaged NR at certain seconds. Particularly, for 7 seconds out of the 10-second traffic noise, the averaged NR of GFANC-RL is higher than that of SFANC, while the performance of the remaining seconds is similar. It may be attributed to the fact that GFANC-RL can flexibly generate various control filters, while SFANC can only select from a limited set of control filters.

4.2.3. Comparison with the FxLMS algorithm

Compared to the FxLMS algorithm, the GFANC-RL method responds faster to real noises, as illustrated in Figs. 9 to 12. The GFANC-RL method consistently surpasses the FxLMS algorithm in terms of averaged NR per second, suggesting that the FxLMS algorithm has not reached steady states on these real noises. Specifically, for aircraft noise as shown in Fig. 12(b), the averaged NR of GFANC-RL is about 10 dB higher than that of FxLMS for most seconds. These findings indicate that the FxLMS algorithm is limited by slow convergence speed and mediocre tracking ability for dynamic noises. Additionally, unlike the FxLMS algorithm, the absence of the feedback error signal during noise control in GFANC-RL minimizes the risk of divergence.

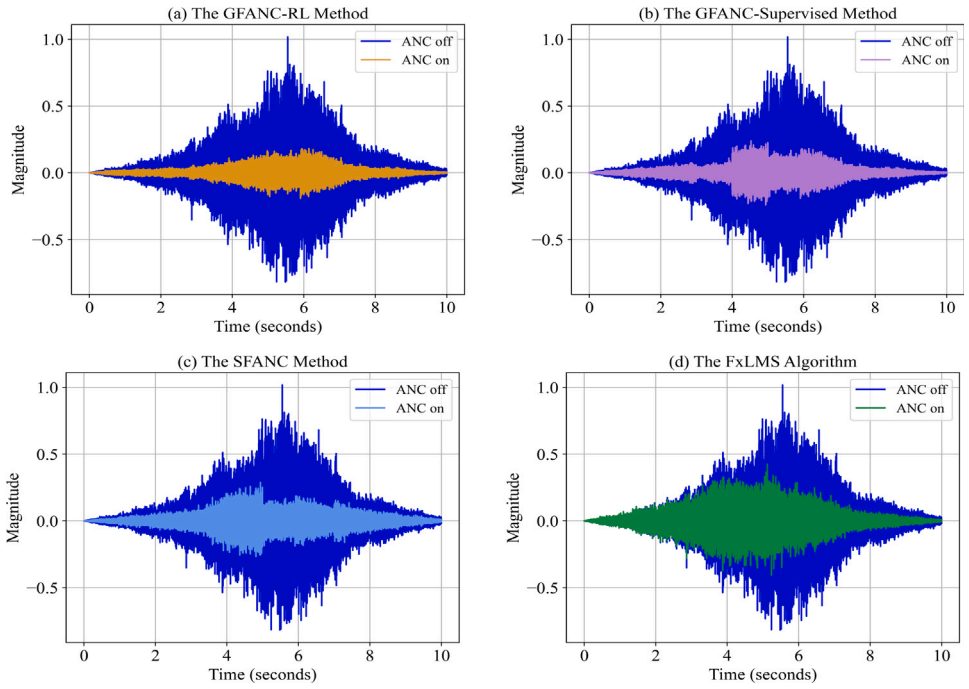


Fig. 9. Noise reduction performances of different ANC algorithms on the traffic noise.

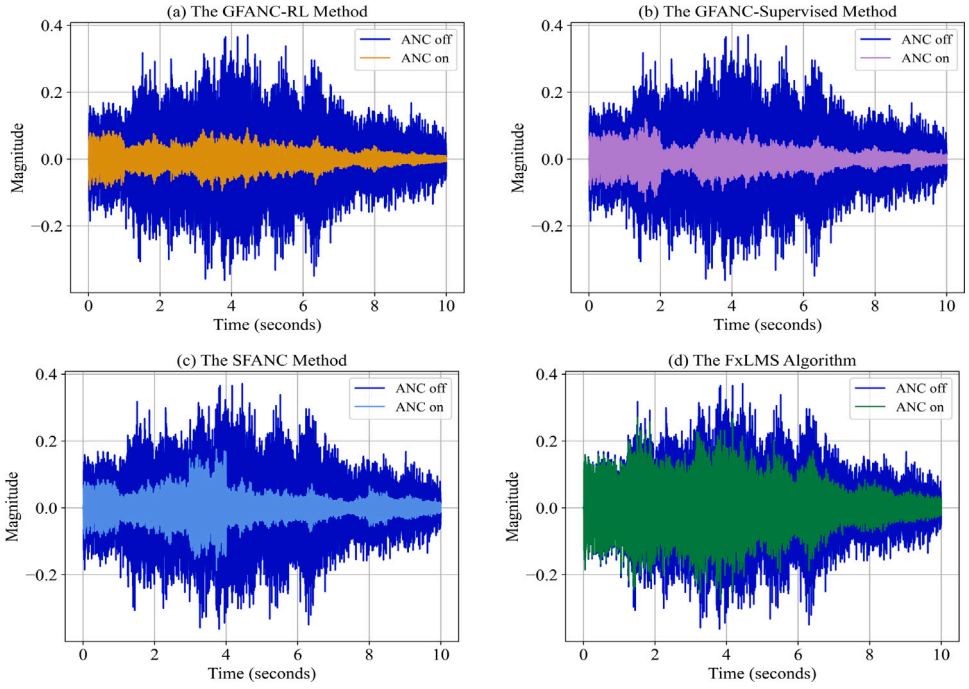


Fig. 10. Noise reduction performance of different ANC algorithms on the aircraft noise.

Additionally, the noise reduction performances of the GFANC-RL method and the FxLMS algorithm are compared by analysing the power spectral density (PSD) in Fig. 13. Noticeably, the main frequency components of these real noises are effectively attenuated by the GFANC-RL method. In comparison, the FxLMS algorithm is less effective at removing the frequency components of these real noises, which aligns with the results in Table 1 and Figs. 9 to 12. Therefore, the simulation results demonstrate that the GFANC-RL method is superior to the FxLMS algorithm in attenuating real dynamic noises.

Table 2
NR values (dB) for the traffic noise obtained by different ANC algorithms on primary paths with different SNR levels.

ANC algorithm	No noise	SNR = 50 dB	SNR = 40 dB	SNR = 30 dB
GFANC-RL	14.4	13.2	11.7	9.3
GFANC-Supervised	12.6	11.9	10.8	8.8
SFANC	11.1	10.0	8.9	8.1
FxLMS	7.1	7.1	7.0	6.6

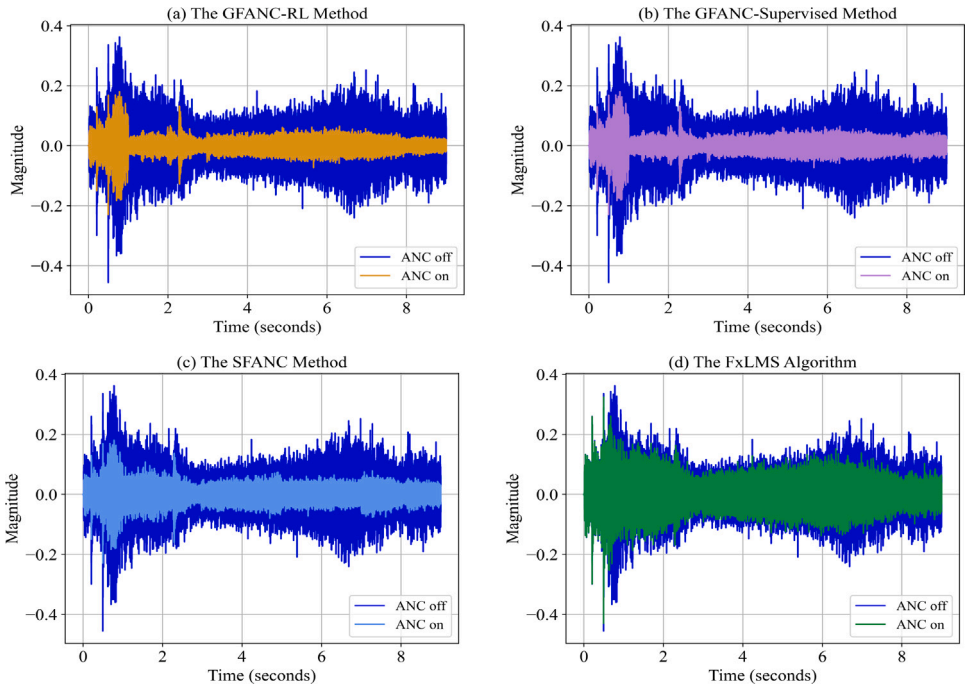


Fig. 11. Noise reduction performance of different ANC algorithms on the drill noise.

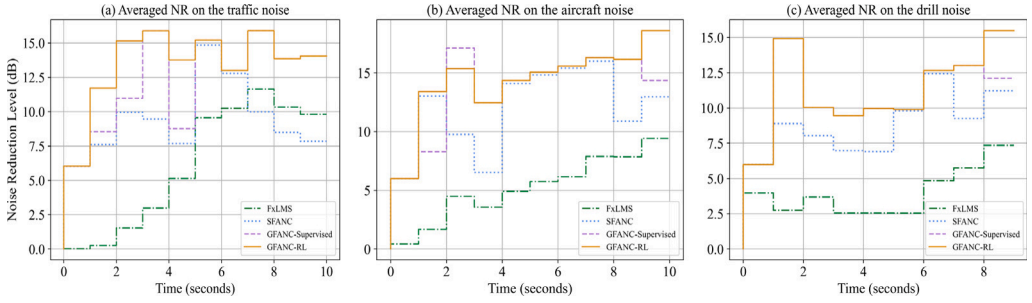


Fig. 12. The averaged NR values in each second obtained by different ANC algorithms on the traffic noise (a), aircraft noise (b), and drill noise (c).

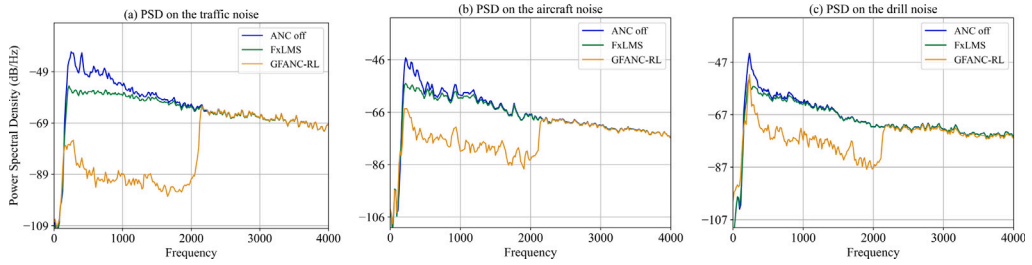


Fig. 13. The power spectral density (PSD) of the traffic noise (a), aircraft noise (b), and drill noise (c) using the GFANC-RL method and the FxLMS algorithm.

4.3. GFANC-RL performance on minor path variations

Section 4.2 has demonstrated the effectiveness of the GFANC-RL method in handling different real noises. This section will assess the performance of the GFANC-RL method against minor variations in acoustic paths. To mimic these variations, white noise was added to the synthetic primary path, and its signal-to-noise ratio (SNR) was adjusted to 50 dB, 40 dB, and 30 dB, respectively. The frequency responses of slightly varied primary paths are depicted in Fig. 14. The NR values achieved by different ANC algorithms for traffic noise on these primary paths are summarized in Table 2.

Table 2 shows that as the SNR of the primary path decreases, the NR value achieved by the GFANC-RL method also diminishes, likely due to the inherent robustness limitations of RL. Despite this, the GFANC-RL method consistently outperforms all other algorithms tested on varied primary paths. Its NR values significantly exceed those of the FxLMS algorithm. For instance, even when the SNR is reduced to 40 dB, GFANC-RL achieves an NR value approximately 5 dB higher than that of FxLMS. However, the FxLMS algorithm experiences a smaller decline in NR as the primary path's SNR decreases, attributed to its adaptive update capability (Luo, Shi, & Gan, 2022). Overall, despite some reduction in performance, the GFANC-RL method continues to work effectively on slightly varied primary paths.

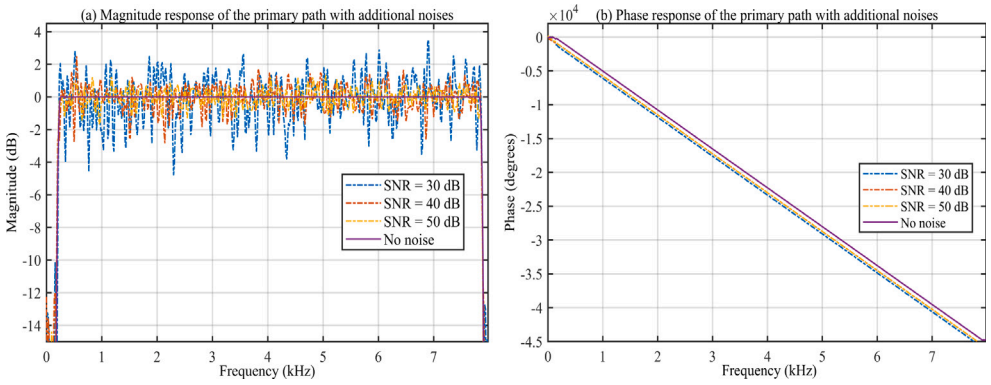


Fig. 14. The magnitude responses (a), and phase responses (b) of the primary path with additional noises.

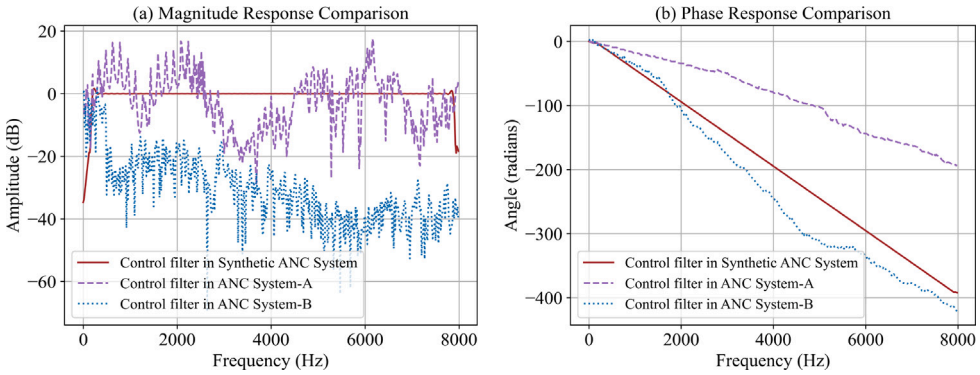


Fig. 15. Frequency spectrum comparison of the pre-trained broadband control filters in different ANC systems.

4.4. GFANC-RL performance in different ANC systems

The GFANC-RL approach can be used in many applications such as ducts, windows, headphones, and vehicles, where rapid responses to varying noises are required. Different applications require specific sub control filters based on their acoustic paths. However, the outputs of the 1D CNN only depend on the input noises. This setup allows the GFANC-RL method to be effectively used in different scenarios with system-specific sub control filters, while keeping the trained 1D CNN consistent across all applications.

This section focuses on assessing the transferability of the GFANC-RL method across various ANC systems. We define two new ANC systems, System-A and System-B, using acoustic paths (illustrated in Appendix B) respectively measured from the vent of a noise chamber and an ANC window. For the two ANC systems, corresponding broadband control filters are pre-trained, with their frequency responses shown in Fig. 15. Sub control filters specific to each system are then obtained, as detailed in Section 2.2.

Detailed PSD results can be found in Figs. B.20 and B.21. Fig. 16 compares the NR values achieved by the GFANC-RL method in different ANC systems. It shows that the GFANC-RL method effectively attenuates the real noises in both System-A and System-B. Notably, for the traffic noise and aircraft noise, the NR values of System-A are lower than those of System-B. This variation can be attributed to differences in the acoustic paths and system-specific sub control filters. Overall, the GFANC-RL method exhibits strong transferability across different ANC systems. Transferring the GFANC-RL method to new systems involves only updating the sub control filters, with the trained 1D CNN remaining unchanged, thus simplifying implementation across various scenarios.

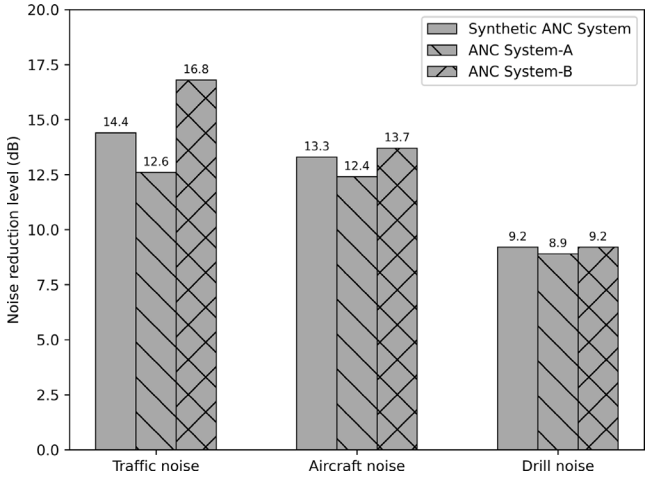


Fig. 16. NR values for real noises obtained by GFANC-RL in different ANC systems.

5. Conclusion

To omit the complicated labelling process and solve the non-differentiable issue in the GFANC-Supervised method, we propose a novel GFANC-RL method in this paper. This paper initially models GFANC as a Markov decision process, providing the theoretical foundations for leveraging RL algorithms. In the GFANC-RL method, the SAC-based RL algorithm is utilized to train the CNN offline, eliminating the labelling process of noise data and enhancing the exploration ability of the CNN. After training the CNN, the GFANC-RL method can achieve delayless real-time noise control.

Table A.3
The hyperparameters of the RL algorithm.

Hyperparameters	Values
total learning steps	300,000
burn-in steps	5,000
replay buffer size	300,000
batch size	128
actor module learning rate	3×10^{-4}
critic module learning rate	1×10^{-3}

Simulations demonstrate the GFANC-RL method's efficacy in handling various real noises, achieving noise reduction performance comparable to the GFANC-Supervised method and surpassing the SFANC method and the FxLMS algorithm. The GFANC-RL method also shows good robustness on slightly varied primary paths. Moreover, transferring the GFANC-RL method to new ANC systems involves only updating sub control filters, while the trained CNN remains unchanged, making it easy to implement across different scenarios.

CRediT authorship contribution statement

Zhengding Luo: Writing – original draft, Validation, Methodology, Formal analysis, Data curation, Conceptualization. **Haozhe Ma:** Writing – original draft, Validation, Methodology, Formal analysis. **Dongyuan Shi:** Writing – review & editing, Validation, Supervision, Methodology. **Woon-Seng Gan:** Writing – review & editing, Supervision.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Appendix A. Hyperparameters of the RL algorithm

Table A.3 summarizes the set of hyperparameters used in the RL algorithm. Furthermore, in this appendix, we investigated the effect of replay buffer size and batch size on the performance of the RL algorithm.

Fig. A.17(a) presents the reward curves for different replay buffer sizes. The results indicate that the algorithm's performance improves with increased replay buffer size. There is a significant drop in the rewards with a replay buffer size of 100k. This drop is due to buffer capacity constraints, causing early experiences to be squeezed out. Additionally, a small buffer causes more frequent changes in the distribution of internal samples, thus leading to instability. In comparison,

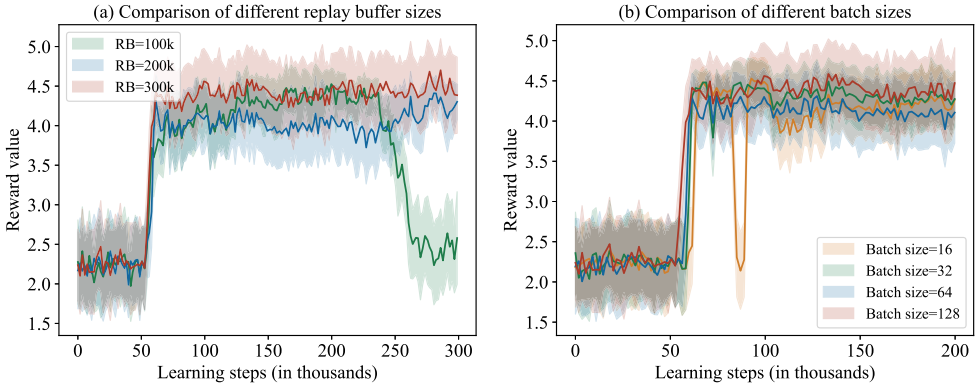


Fig. A.17. Performance comparison of the RL algorithm using different replay buffer sizes and batch sizes.

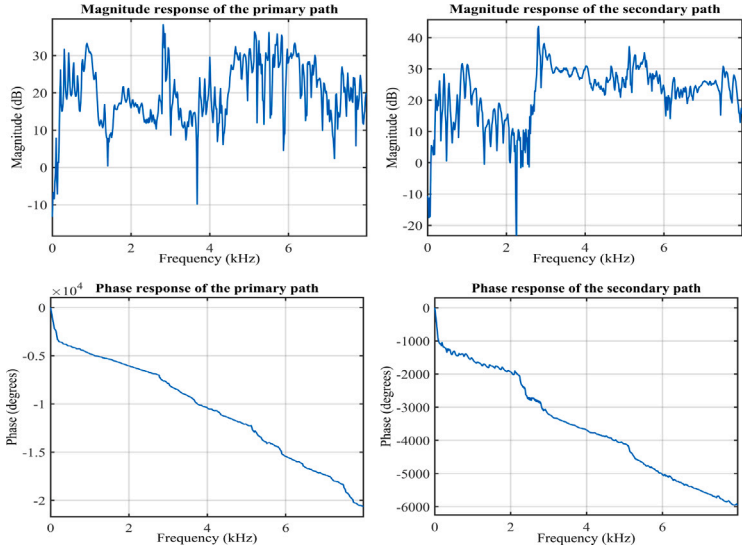


Fig. B.18. The magnitude responses and phase responses of acoustic paths in ANC System-A.

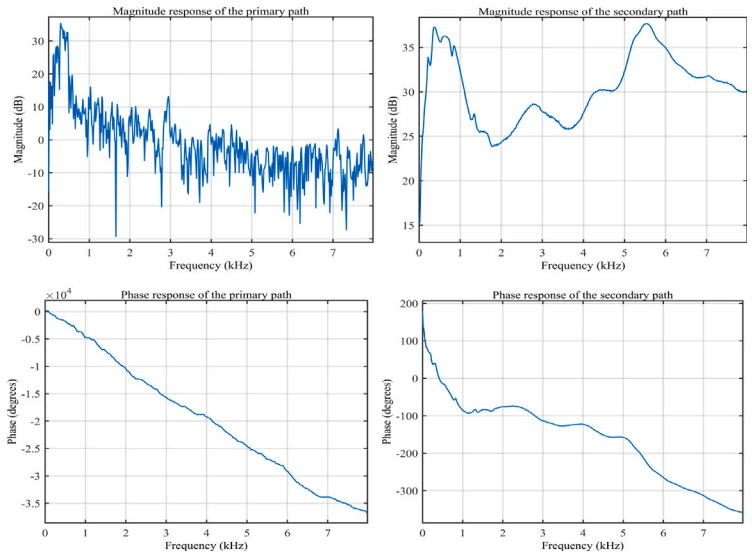


Fig. B.19. The magnitude responses and phase responses of acoustic paths in ANC System-B.

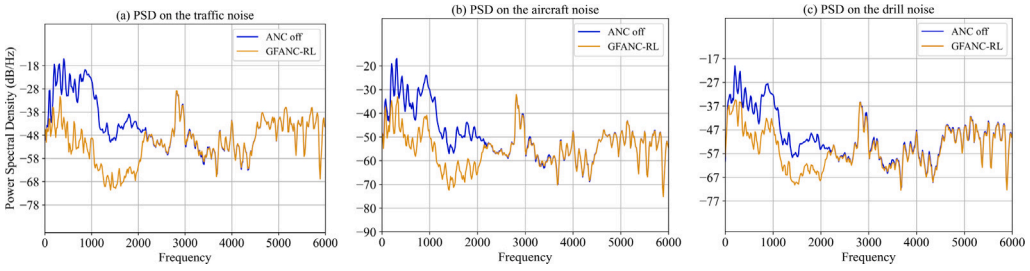


Fig. B.20. The PSD of the traffic noise (a), aircraft noise (b), and drill noise (c) using GFANC-RL in ANC System-A.

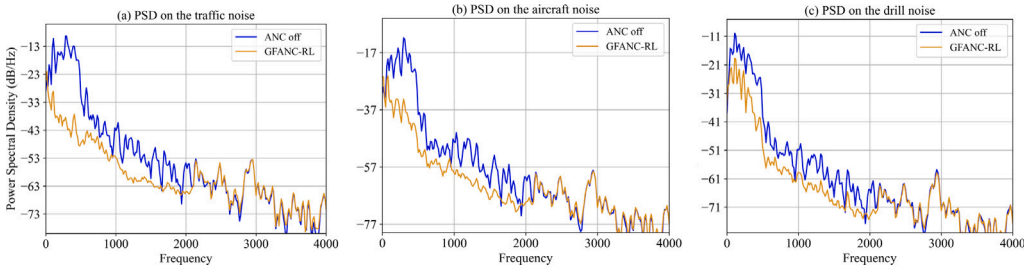


Fig. B.21. The PSD of the traffic noise (a), aircraft noise (b), and drill noise (c) using GFANC-RL in ANC System-B.

buffer sizes of 200k and 300k both achieve stable training outcomes. Specifically, a replay buffer size of 300k is sufficient to store all experiences encountered during training. Therefore, it is essential that the replay buffer maintains a broad range of experiences to facilitate effective learning.

Additionally, the performance comparison of the RL algorithm with different batch sizes is shown in Fig. A.17(b). The results indicate that the convergence speed slightly increases with larger batch sizes. However, the improvement becomes less significant when the batch size exceeds 32. Moreover, we find that a batch size of 16 exhibits significant instability in the learning process, characterized by a huge drop. This is due to the smaller batch size leading to less accurate estimations of the loss. The experimental findings suggest that the convergence of the algorithm is relatively insensitive to the batch size. Since larger batch sizes can enhance stability, we chose a batch size of 128 in this paper.

Appendix B. ANC system-A and system-B

The magnitude and phase responses of the acoustic paths in ANC System-A and System-B are illustrated in Figs. B.18 and B.19. When the GFANC-RL method is used in ANC System-A and System-B to attenuate three real noises, the PSD results are shown in Figs. B.20 and B.21.

References

Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. (2017). Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, *34*(6), 26–38.

Benois, P. R., Roden, R., Blau, M., & Doclo, S. (2022). Optimization of a fixed virtual sensing feedback ANC controller for in-ear headphones with multiple loudspeakers. In *ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing* (pp. 8717–8721). IEEE.

Chang, C.-Y., Chuang, C.-T., Kuo, S. M., & Lin, C.-H. (2022). Multi-functional active noise control system on headrest of airplane seat. *Mechanical Systems and Signal Processing*, *167*, Article 108552.

Elliott, S. J., & Nelson, P. A. (1993). Active noise control. *IEEE Signal Processing Magazine*, *10*(4), 12–35.

Fujimoto, S., Hoof, H., & Meger, D. (2018). Addressing function approximation error in actor-critic methods. In *International conference on machine learning* (pp. 1587–1596). PMLR.

Fuller, C. R., & von Flotow, A. H. (1995). Active control of sound and vibration. *IEEE Control Systems Magazine*, *15*(6), 9–19.

George, N. V., & Panda, G. (2013). Advances in active noise control: A survey, with emphasis on recent nonlinear techniques. *Signal Processing*, *93*(2), 363–377.

Gupta, R., He, J., Ranjan, R., Gan, W.-S., Klein, F., Schneiderwind, C., et al. (2022). Augmented/mixed reality audio for hearables: Sensing, control, and rendering. *IEEE Signal Processing Magazine*, *39*(3), 63–89.

Haarnoja, T., Tang, H., Abbeel, P., & Levine, S. (2017). Reinforcement learning with deep energy-based policies. In *International conference on machine learning* (pp. 1352–1361). PMLR.

Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning* (pp. 1861–1870). PMLR.

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., et al. (2018). Soft actor-critic algorithms and applications. arXiv preprint arXiv:1812.05905.

Han, N., & Qiu, X. (2007). A study of sound intensity control for active noise barriers. *Applied Acoustics*, *68*(10), 1297–1306.

Hansen, C. N. (2002). *Understanding active noise cancellation*. CRC Press.

Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., et al. (2018). Rainbow: Combining improvements in deep reinforcement learning. *32*, In *Proceedings of the AAAI conference on artificial intelligence*. (1).

Iotov, Y., Nørholm, S. M., Belyi, V., Dyrholm, M., & Christensen, M. G. (2022). Computationally efficient fixed-filter anc for speech based on long-term prediction for headphone applications. In *ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing* (pp. 761–765). IEEE.

Jung, W., Elliott, S. J., & Cheer, J. (2019). Local active control of road noise inside a vehicle. *Mechanical Systems and Signal Processing*, *121*, 144–157.

Kajikawa, Y., Gan, W.-S., & Kuo, S. M. (2012). Recent advances on active noise control: open issues and innovative applications. *APSIPA Transactions on Signal and Information Processing*, *1*, Article e3.

Kuo, S. M., Mitra, S., & Gan, W.-S. (2006). Active noise control system for headphone applications. *IEEE Transactions on Control Systems Technology*, *14*(2), 331–335.

Kuo, S. M., & Morgan, D. R. (1999). Active noise control: a tutorial review. *Proceedings of the IEEE*, *87*(6), 943–973.

Lam, B., Shi, D., Gan, W.-S., Elliott, S. J., & Nishimura, M. (2020). Active control of broadband sound through the open aperture of a full-sized domestic window. *Scientific Reports*, *10*(1), 1–7.

Luo, Z., Shi, D., & Gan, W.-S. (2022). A hybrid SFANC-FxNLMS algorithm for active noise control based on deep learning. *IEEE Signal Processing Letters*, *29*, 1102–1106.

Luo, Z., Shi, D., Gan, W.-S., & Huang, Q. (2024). Delayless generative fixed-filter active noise control based on deep learning and Bayesian filter. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *32*, 1048–1060.

Luo, Z., Shi, D., Gan, W.-S., Huang, Q., & Zhang, L. (2023). Performance evaluation of selective fixed-filter active noise control based on different convolutional neural networks. In *INTER-nOISE and NOISE-CON congress and conference proceedings* (pp. 1615–1622).

Luo, Z., Shi, D., Ji, J., & Gan, W.-s. (2022). Implementation of multi-channel active noise control based on back-propagation mechanism. arXiv preprint arXiv:2208.08086.

Luo, Z., Shi, D., Ji, J., Shen, X., & Gan, W.-S. (2024). Real-time implementation and explainable AI analysis of delayless CNN-based selective fixed-filter active noise control. *Mechanical Systems and Signal Processing*, *214*, Article 111364.

Luo, Z., Shi, D., Shen, X., Ji, J., & Gan, W.-S. (2023). Deep generative fixed-filter active noise control. In *ICASSP 2023-2023 IEEE international conference on acoustics, speech and signal processing* (pp. 1–5). IEEE.

Luo, Z., Shi, D., Shen, X., Ji, J., & Gan, W.-S. (2024). GFANC-Kalman: Generative fixed-filter active noise control with CNN-Kalman filtering. *IEEE Signal Processing Letters*, *31*, 276–280.

Ma, H., Luo, Z., Vo, T. V., Sima, K., & Leong, T.-Y. (2024). Highly efficient self-adaptive reward shaping for reinforcement learning. arXiv preprint arXiv:2408.03029.

Ma, H., Sima, K., Vo, T. V., Fu, D., & Leong, T.-Y. (2024). Reward shaping for reinforcement learning with an assistant reward agent. In *Forty-first international conference on machine learning*. PMLR.

Ma, H., Vo, T. V., & Leong, T.-Y. (2023). Hierarchical reinforcement learning with human-AI collaborative sub-goals optimization. In *Proceedings of the 2023 international conference on autonomous agents and multiagent systems* (pp. 2310–2312).

Ma, H., Vo, T. V., & Leong, T.-Y. (2024). Mixed-initiative Bayesian sub-goal optimization in hierarchical reinforcement learning. In *Proceedings of the 23rd international conference on autonomous agents and multiagent systems* (pp. 1328–1336).

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., et al. (2016). Asynchronous methods for deep reinforcement learning. In *International conference on machine learning* (pp. 1928–1937). PMLR.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533.

Mostafavi, A., & Cha, Y.-J. (2023). Deep learning-based active noise control on construction sites. *Automation in Construction*, *151*, Article 104885.

Pan, N., Chen, J., & Benesty, J. (2022). DNN based multiframe single-channel noise reduction filters. In *ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing* (pp. 8782–8786). IEEE.

Park, S., Patterson, E., & Baum, C. (2019). Long short-term memory and convolutional neural networks for active noise control. In *2019 5th international conference on frontiers of signal processing* (pp. 121–125). IEEE.

Pawełczyk, M. (2002). Analogue active noise control. *Applied Acoustics*, *63*(11), 1193–1213.

Read, J., Pfahringer, B., Holmes, G., & Frank, E. (2011). Classifier chains for multi-label classification. *Machine Learning*, *85*(3), 333–359.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2015). Trust region policy optimization. In *International conference on machine learning* (pp. 1889–1897). PMLR.

Schumacher, T., Krüger, H., Jeub, M., Vary, P., & Beaugeant, C. (2011). Active noise control in headsets: A new approach for broadband feedback ANC. In *2011 IEEE international conference on acoustics, speech and signal processing* (pp. 417–420). IEEE.

Shi, C., Du, F., & Wu, Q. (2022). A digital twin architecture for wireless networked adaptive active noise control. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *30*, 2768–2777.

Shi, D., Gan, W.-S., Lam, B., & Wen, S. (2020). Feedforward selective fixed-filter active noise control: Algorithm and implementation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *28*, 1479–1492.

Shi, D., Gan, W.-s., Shen, X., Luo, Z., & Ji, J. (2024). What is behind the meta-learning initialization of adaptive filter? — A naive method for accelerating convergence of adaptive multichannel active noise control. *Neural Networks*, *172*, Article 106145.

Shi, D., Lam, B., Ooi, K., Shen, X., & Gan, W.-S. (2022). Selective fixed-filter active noise control based on convolutional neural network. *Signal Processing*, *190*, Article 108317.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., & Riedmiller, M. (2014). Deterministic policy gradient algorithms. In *International conference on machine learning* (pp. 387–395). Pmlr.

Sun, H., Jin, C. T., Abhayapala, T., & Samarasinghe, P. (2024). Active noise control over 3D space with a dynamic noise source. In *ICASSP 2024-2024 IEEE international conference on acoustics, speech and signal processing* (pp. 1236–1240). IEEE.

Sun, H., Zhang, J., Abhayapala, T., & Samarasinghe, P. (2022). Spatial active noise control with the remote microphone technique: An approach with a moving higher order microphone. In *ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing* (pp. 8707–8711). IEEE.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT Press.

Toyooka, S., & Kajikawa, Y. (2023). Hybrid active noise control with auxiliary filter-based virtual sensing. In *INTER-nOISE and NOISE-CON congress and conference proceedings*: *vol. 268*, (2), (pp. 6488–6495). Institute of Noise Control Engineering.

Wu, Y., Mansimov, E., Grosse, R. B., Liao, S., & Ba, J. (2017). Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. *Advances in Neural Information Processing Systems*, *30*.

Xiao, T., & Doclo, S. (2024). Effect of target signals and delays on spatially selective active noise control for open-fitting hearables. arXiv preprint arXiv:2401.07681.

Xie, R., Tu, A., Shi, C., Elliott, S., Li, H., & Zhang, L. (2024). Cognitive virtual sensing technique for feedforward active noise control. In *ICASSP 2024-2024 IEEE international conference on acoustics, speech and signal processing* (pp. 981–985). IEEE.

Yang, F., Guo, J., & Yang, J. (2020). Stochastic analysis of the filtered-x LMS algorithm for active noise control. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *28*, 2252–2266.

Yang, Z., Wang, Y., Zhong, X., Tao, J., Shi, L., Zou, H., et al. (2022). A database for active control of road noise in automobile cabins. In *The 24th international congress on acoustics*.

Zhang, J., Elliott, S. J., & Cheer, J. (2021). Robust performance of virtual sensing methods for active noise control. *Mechanical Systems and Signal Processing*, *152*, Article 107453.

Zhang, J. A., Murata, N., Maeno, Y., Samarasinghe, P. N., Abhayapala, T. D., & Mitsufuji, Y. (2020). Coherence-based performance analysis on noise reduction in multichannel active noise control systems. *Journal of the Acoustical Society of America*, *148*(3), 1519–1528.

Zhang, H., Pandey, A., & Wang, D. (2022). Attentive recurrent network for low-latency active noise control. In *INTERSPEECH* (pp. 956–960).

Zhang, H., & Wang, D. (2021). Deep ANC: A deep learning approach to active noise control. *Neural Networks*, *141*, 1–10.

Zhang, H., & Wang, D. (2023). Deep MCANC: A deep learning approach to multi-channel active noise control. *Neural Networks*, *158*, 318–327.