Expert Systems With Applications 309 (2026) 131247

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Hybrid deep learning–based active noise control for encapsulated structures with openings

Alkahf Aboutiman [a,*], Khaled Said Ahmed Maamoun [b], Hamid Reza Karimi [a], Francesco Ripamonti [a]

[a] Department of Mechanical Engineering, Politecnico di Milano, Via La Masa 1, 20156, Milan, Italy
[b] Department of Measurements and Control Systems, Silesian University of Technology, Akademicka 2A, 44–100, Gliwice, Poland

ARTICLE INFO

Keywords:
Active noise control
Learning-based weighting
Encapsulated structure with opening
Loudspeaker nonlinearity

ABSTRACT

Traditional active noise control (ANC) systems rely on adaptive filtering techniques that assume a linear relationship between signals. However, in practical implementations, nonlinear distortions, particularly those introduced by loudspeakers, can degrade control performance, limiting the effectiveness of conventional ANC approaches. This paper presents a hybrid ANC algorithm, GFANC-THFxNLMS, designed to enhance adaptability and robustness in complex acoustic environments, such as encapsulated structures with openings. The proposed method leverages deep learning to construct a dataset of pre-trained sub-control filters and employs a 1D convolutional neural network (CNN) to dynamically generate the most suitable filter for the incoming noise. An adaptive filtering stage then fine-tunes the control filter in real time, ensuring optimal noise attenuation. The performance of GFANC-THFxNLMS is evaluated against conventional ANC algorithms, namely GFANC and THFxNLMS, under both time-invariant and time-varying conditions. The results indicate that the proposed method achieves enhanced noise reduction, with 6.74 dB using THFxNLMS, 8.01 dB using GFANC, and 9.49 dB using GFANC-THFxNLMS, highlighting its effectiveness in noise mitigation.

1. Introduction

Exposure to excessive noise is recognized as a serious threat to human well-being, with significant impacts on comfort, health, and safety (World Health Organization, 2018). The advancement of industrial machinery and domestic devices has contributed to elevated sound levels, potentially compromising both user comfort and operational safety. Encapsulated structures are commonly used as noise barriers to isolate sound sources from their surroundings (Isaac et al., 2022; Wrona et al., 2022). However, incorporating openings in these structures-necessary for airflow, heat dissipation, and device monitoring-compromises their ability to reduce noise transmission (Maamoun et al., 2024, 2025). This trade-off is essential in many engineering and industrial applications. It has increased the demand for advanced noise reduction techniques, such as ANC. The principle of superposition serves as the foundation for ANC, a technique that reduces unwanted noise by generating an anti-noise signal of the same amplitude but opposite phase, thus canceling the disturbance (Goodwin et al., 2010; Jeong & Park, 2023). Unlike passive noise control, which relies on sound-absorbing materials, ANC actively adapts in real time by manipulating acoustic signals (Aboutiman et al., 2025a; Lam et al., 2020; Tan & Wang, 2019). This real-time adaptability makes ANC a valuable solution in fields such as domestic appliances, construction, and manufacturing, where passive methods are limited by cost and space constraints (Hartmann, 2004).

In enclosed environments, passive noise control remains the predominant approach, primarily relying on materials engineered for sound absorption. However, such methods become less effective at low frequencies due to the substantial mass required for adequate performance (Aboutiman et al., 2025b; Fuller et al., 1991; Misol et al., 2012). This limitation is particularly significant for many domestic appliances and industrial machines, where the dominant noise components typically lie below 500 Hz. One implementation method for ANC in enclosures with openings, known as boundary control (Wang et al., 2017), involves placing secondary actuators along the perimeter of the opening. This configuration improves airflow since the speakers do not obstruct the aperture. However, the effectiveness of this method may be limited by physical constraints imposed by the boundary length. It also requires a boundary thick enough to mount the actuators securely. In contrast, planar control arranges the secondary actuators in a grid pattern across the plane of the opening (Murao et al., 2016; Tao et al., 2016). Although

* Corresponding author.
E-mail addresses: alkahfahmat.aboutiman@polimi.it (A. Aboutiman), khaled.maamoun@polsl.pl (K.S.A. Maamoun), hamidreza.karimi@polimi.it (H.R. Karimi), francesco.ripamonti@polimi.it (F. Ripamonti).

https://doi.org/10.1016/j.eswa.2026.131247
Received 27 June 2025; Received in revised form 14 January 2026; Accepted 14 January 2026
Available online 16 January 2026

this approach eliminates the need for thick boundaries, it reduces access to the noise source because the actuator grid partially blocks the opening. However, none of these approaches discuss the effects of nonlinearities in such systems.

Nonlinearities in ANC systems present significant challenges, as they can degrade the performance of adaptive filtering techniques. Traditional ANC relies on adaptive filters, such as the widely used Filtered-x Least Mean Square (FxLMS) algorithm, which estimates the secondary path and minimizes an error signal to optimize noise cancellation (Elliott et al., 1987; Manolakis et al., 2005). However, real-world ANC applications introduce nonlinear distortions due to hardware imperfections in amplifiers and loudspeakers, which result in inaccurate secondary path estimation and reduced noise attenuation (Kuo & Morgan, 1996). Several approaches have been proposed to address these issues. The Volterra expansion has proven effective in modeling weak nonlinearities, and second-order Volterra-based FxLMS algorithms have been developed to improve feedforward ANC performance (Guo et al., 2018; Lashkari, 2006). Additionally, nonlinear models such as the Nonlinear AutoRegressive model with eXogenous variables (NARX) have been employed to enhance ANC controller efficiency (Napoli & Piroddi, 2009). To mitigate saturation-type nonlinearities, variants like the Tangential Hyperbolic Function FxLMS (THF-FxLMS) and bilinear FxLMS have been proposed (Ghasemi et al., 2016; Kuo & Wu, 2005). More recent methods, including FXlogLMS, FXatanLMS, and FXsigLMS, use nonlinear transformations such as logarithmic, inverse tangent, and sigmoid functions. These techniques help avoid threshold estimation and improve convergence (Jun et al., 2012; Wu et al., 2010; Zhou et al., 2015). While such advancements improve control accuracy, they often increase computational cost. To balance efficiency and robustness, hybrid algorithms like MGFxRLS-NSSFxLMS and GMACFxLMS have been introduced (Meng & Chen, 2020). In parallel, feedback ANC systems, which rely solely on error microphones to measure residual noise, have been explored as alternatives to feedforward ANC in reverberant environments where reference signals are distorted (Lam et al., 2021; Zhang & Wang, 2021). However, traditional feedback systems struggle with broadband noise attenuation and require precise controller design to maintain stability. Recent advances in robust control, including fuzzy-based and fault-tolerant strategies, show potential for nonlinear stochastic systems but remain underexplored in ANC applications (Liang et al., 2022; Sun et al., 2020). Despite significant progress, efficiently mitigating nonlinear distortions in ANC remains an active area of research.

To address these limitations, neural networks have been explored for nonlinear ANC due to their ability to model complex relationships. Early studies introduced multilayer perceptron (MLP) networks for active vibration control, where adaptive filtered-x backpropagation was used to update network weights (Snyder & Tanaka, 1995). Further advancements led to improved training algorithms that enhanced convergence speed while reducing computational cost (Bouchard et al., 1999; Chang & Luoh, 2007). Alternative architectures, such as functional link neural networks (Das & Panda, 2004; Krukowicz, 2010), radial basis function networks (Tokhi & Wood, 1997), fuzzy neural networks (Zhang et al., 2006), and recurrent neural networks (Bambang, 2008), have also been employed to mitigate nonlinear distortions. These approaches demonstrated the potential of artificial intelligence in ANC by using online adaptation to continuously adjust control parameters. With the advent of deep learning, new opportunities have emerged for tackling nonlinear ANC problems. Deep neural networks (DNNs) can model highly nonlinear and time-varying systems, allowing them to adapt to complex acoustic environments. Unlike traditional machine learning models that require manual feature extraction, deep learning methods-such as convolutional neural networks (CNNs) and long short-term memory (LSTM) networks-can automatically learn spatial and temporal features relevant to noise control. CNN-based ANC models have shown superior performance by leveraging hierarchical feature extraction, which improves noise reduction across various environments (Cha et al., 2018). Meanwhile, LSTM networks excel at capturing long-term temporal dependencies, making them particularly effective for predicting and canceling nonstationary noise patterns (Duan et al., 2016; Fu et al., 2016). Recent studies have demonstrated that deep learning-based ANC approaches can outperform conventional adaptive filtering methods. For instance, convolutional recurrent networks combining CNN and LSTM architectures have been proposed as alternatives to Volterra-based FxLMS controllers and MLP-based ANC systems. These deep models show improved generalization across noise types and environments, overcoming the limitations of single-filter adaptive methods (Cha et al., 2023; Guo et al., 2018; Zhang & Wang, 2021). In parallel, hybrid learning-based architectures have been explored in real-time embedded applications under strict resource constraints, such as threat detection in IoT networks. These studies show that combining lightweight neural networks for feature reduction with classical adaptive stages enables efficient and robust real-time processing, which is well aligned with the requirements of embedded ANC systems (Elzaghmouri et al., 2024; Sivasakthi et al., 2024).

This paper introduces a hybrid ANC algorithm, GFANC-THFxNLMS, which leverages deep learning to address the nonlinearities introduced by loudspeakers in ANC applications for encapsulated structures with openings. The proposed method builds upon the GFANC framework described in (Luo et al., 2025, 2023), which enhances traditional ANC algorithms such as SFANC by decomposing a pre-trained broadband control filter into multiple sub-filters (Vetterli & Le Gall, 1989). A lightweight 1D CNN running on a co-processor then determines the optimal combination weights based on the characteristics of the primary noise.This design minimizes computational load and simplifies the ANC setup, as it relies on a single pre-trained filter rather than multiple independently trained ones. In addition, the GFANC-THFxNLMS hybrid improves performance by adaptively updating the control filter coefficients, following the adaptive strategy introduced in (Luo et al., 2022). The main contributions of this paper are threefold: (i) developing an ANC algorithm that uses deep learning to overcome the limitations caused by nonlinearities; (ii) implementing and validating this algorithm in a complex acoustic environment, specifically an encapsulated structure with an opening; and (iii) proposing a 1D CNN architecture that achieves high accuracy while keeping the number of parameters minimal.

The rest of this paper is structured as follows: The paper first presents the experimental setup used to compute the impulse response of the control environment. Secondly, it introduces the ANC algorithm along with its neural network architecture and training process. The results obtained with the proposed algorithm are then presented and compared with traditional algorithms. Finally, the paper concludes the study and highlights its contributions.

2. System identification

The identification experiments were conducted in a laboratory room measuring 5.8 m in length and 3.5 m in width. While the walls are covered with sound-absorbing materials, the presence of various equipment and objects results in acoustic conditions that more closely resemble a typical operational environment than those of an anechoic chamber.
The experimental setup consists of a rectangular steel plate mounted on a concrete enclosure that houses the primary noise source, as shown in Fig. 1. A Behringer EUROLIVE B208D speaker is employed as the primary excitation source, generating band-limited random noise with frequencies up to 1500 Hz with a maximum sound pressure level (SPL) of 113 dB. The secondary source is a loudspeaker (JBL Stage2 424) located inside the enclosure, positioned centrally behind the opening. The Analog Device SC598 with SHARC+ core is used along with EVSOMCRR-EZKIT development board (Analog Devices, 2024). The plate is made of steel, characterized by a density of $\rho = 7850\,\mathrm{kg/m^3}$ and a Young's modulus of $E = 210 \times 10^9$ Pa. Its dimensions are $a = 0.42$ m (length), $b = 0.39$ m (width), and $h = 0.001$ m (thickness). This setup, with the given parameters, is used to identify the primary and secondary paths using Mic 1, as it represents a wide range of engineering applications involving encapsulated structures with openings.

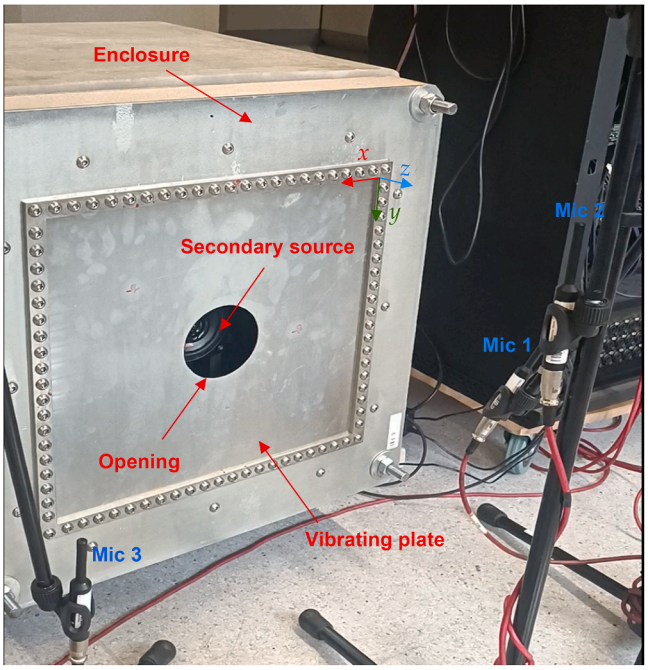


Fig. 1. Overview of the experimental rig showing a vibrating plate with a central circular opening of radius 0.05 m. The steel plate spans 0.42 m in x, 0.39 m in y, and has a thickness of 0.001 m in $-z$. Microphones (Mic 1, Mic 2, and Mic 3) are positioned at coordinates (-0.18 m, 0.12 m, 0.25 m), (0.21 m, -0.1 m, 0.4 m), and (0.53 m, 0.24 m, 0.25 m), respectively. Mic 1 is used as the error microphone, while Mic 2 and Mic 3 are intended for observation, if needed. The primary noise source is located at (0.3 m, 0.1 m, 0.1 m) and the loudspeaker, used as a secondary source, is centered behind the opening.

For path identification, the Normalized Least Mean Square (NLMS) algorithm is employed. The original signals are recorded at a sampling rate of 48 kHz and subsequently downsampled to 3 kHz to reduce computational complexity, given that the frequency content of interest lies below the Nyquist limit. A filter length of 300 coefficients is used for both the primary and secondary paths. The corresponding transfer functions, obtained from the measured impulse responses, are presented in Fig. 2. The primary path exhibits prominent resonant behavior, particularly below 350 Hz, due to the complex noise barrier characteristics influenced by both the radiated sound from the vibrating plate and the sound leakage through the opening (Maamoun et al., 2024). In contrast, the secondary path, which originates from a secondary loudspeaker positioned near an opening, is less affected by structural vibrations. Its response is more attenuated and smoother, particularly at mid-to-high frequencies, indicating a more direct acoustic path with minimal structural coupling. The phase responses of both paths exhibit smooth behavior which indicates physically consistent acoustic paths' modeling , as in Fig. 2.

3. The GFANC-THFxNLMS algorithm

This section presents the GFANC-THFxNLMS based on deep learning and the configuration used to make it effective in the environment presented in the previous section. The block diagram of the algorithm is presented in Fig. 3.

3.1. The tangent hyperbolic filtered-x normalized least mean square algorithm description

As presented in (Ghasemi et al., 2016), the Tangent Hyperbolic Filtered-x Normalized Least Mean Square (THFxNLMS) algorithm is an adaptive algorithm designed to enhance the robustness of ANC systems, particularly in the presence of nonlinearities. This section presents the THFxNLMS algorithm, which extends the standard FxNLMS method by incorporating a hyperbolic tangent transformation to improve stability and performance. The block diagram of the THFxNLMS algorithm is the one presented in Fig. 3 without the co-processor, where $P(z)$ represents the transfer function of the primary path, $W(z)$ denotes the control filter, $S(z)$ is the transfer function of the secondary path, and its estimate is denoted as $\hat{S}(z)$.

The reference signal vector $x(n)$ is defined as $x(n) = [x(n), x(n-1), \ldots, x(n-L+1)]^T$. Let T denote the transpose operation and L denotes the length of the control filter. The resulting output of the control filter is therefore $y(n)$, as defined in Eq. (1).

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

Where $w(n)$ represents the coefficient vector of the control filter. The residual error signal can be expressed as follows:

$$e(n) = d(n) - s(n) * \left(w^T(n)x(n)\right), \tag{2}$$

where $d(n)$ and $s(n)$ denote the primary disturbance and the impulse response of the secondary path, respectively, and $*$ is the linear convolution. A key difference introduced by the THFxNLMS algorithm is the use of a nonlinear transformation for the residual error signal $e(n)$. The transformation function is defined as:

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{3}$$

This function is odd-symmetric, continuous, monotonic, and bounded within the range $(-1, 1)$. By applying this transformation, the variance of the error signal remains finite, improving the stability of the learning process and making the algorithm more resilient to nonlinear distortions. The loss function of the THFxNLMS algorithm is given by:

$$J(n) = \frac{f^2(b(n)e(n))}{2 \cdot \|x'(n)\|^2 + \epsilon}, \tag{4}$$

where b is a compression factor that adjusts the degree of compression of the residual error. ϵ is a small positive constant that overcomes the possible numerical instability when updating the weights. In our case, it is equal to 2.2204×10^{-16}. Using the steepest descent method, the gradient of the loss function with respect to the filter coefficients $w(n)$ is computed as:

$$\nabla J(n) = \frac{-2b\tanh(b(n)e(n))[1 - \tanh^2(b(n)e(n))] \cdot x'(n)}{2 \cdot \|x'(n)\|^2 + \epsilon}, \tag{5}$$

where $x'(n)$ represents the filtered reference signal. The update rule for the THFxNLMS algorithm is therefore given by:

$$w(n+1) = w(n) - \mu\psi(e(n))x'(n), \tag{6}$$

where $\psi(e(n)) = \frac{\tanh(b(n)e(n))[1-\tanh^2(b(n)e(n))]}{2\cdot\|x'(n)\|^2+\epsilon}$ acts as a robust transformation function. The error signal is expressed as:

$$e(n) = d(n) - s(n) * w^T(n)x(n), \tag{7}$$

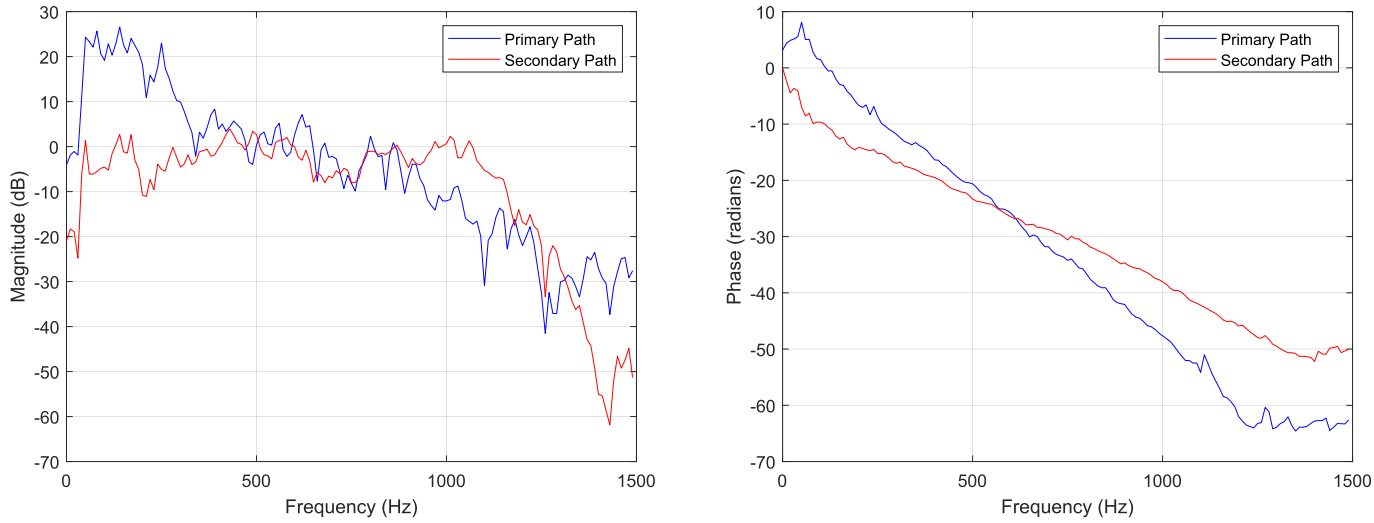
where $d(n)$ and $s(n)$ denote the primary disturbance and the impulse response of the secondary path, respectively, and $*$ represents the linear convolution. The stability of the THFxNLMS algorithm depends on the choice of the step size μ. Considering the weight update rule presented in Eq. 6. a sufficient condition to ensure stability is:

$$0 < \mu < \frac{1}{\lambda_{\max} D_s},$$

where:

- D_s is the group delay of the secondary path, which can be estimated from its impulse response $s(n)$;
- $\lambda_{\max}$ is the maximum eigenvalue of the autocorrelation matrix of the filtered reference signal $x'(n)$:

$$R_{x'} = E\left[x'(n)x'^T(n)\right], \quad \lambda_{\max} = \max \text{ eig}(R_{x'}).$$



(a) Magnitude response (b) Phase response

Fig. 2. Frequency responses of the primary and secondary paths identified using the NLMS algorithm.

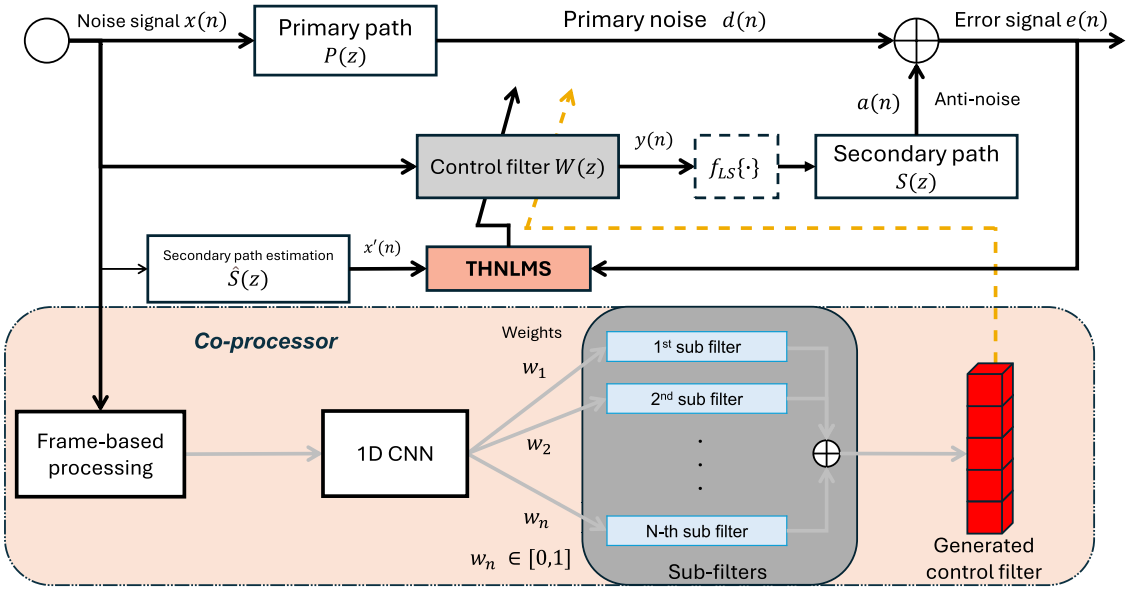


Fig. 3. Block diagram of the hybrid GFANC-THFxNLMS algorithm. f_{LS} represents the nonlinearity function associated with the loudspeaker. The noise signal is processed by the co-processor frame by frame, and for each frame, the 1D CNN provides the most suitable weight to design the control filter. This control filter is then fine-tuned by the THFxNLMS algorithm.

This inequality provides a verifiable upper bound for μ, ensuring that the adaptive update remains stable under the nonlinear transformation $\psi(e(n))$. It is important to note that $\lambda_{\max}$ depends on the reference signal statistics and D_s is determined by the secondary path. Therefore, in practical applications, μ is chosen conservatively based on the expected signal and path characteristics to ensure robust performance. The inclusion of the hyperbolic tangent function in the THFxNLMS algorithm enhances its robustness against nonlinearities, making it more suitable for ANC applications in complex acoustic environments where standard adaptive algorithms may struggle to achieve optimal performance. In addition, as indicated in (Li et al., 2023) the choice of the parameter b significantly influences the behavior of the robust transformation function. A higher b-value results in a steeper transformation curve, enhancing the algorithm's ability to suppress outliers. However, this also leads to a slower convergence rate. Conversely, a lower b-value produces a smoother transformation curve, accelerating the convergence speed but weakening the suppression of outliers. Additionally, a small b-value ensures that the step size remains stable during the adaptive process, preventing abrupt changes in the update rule.

In practical applications, selecting an appropriate b-value is crucial and should be based on the characteristics of the noise environment. To address the challenge of choosing b, an adaptive adjustment mechanism using a sigmoid function is introduced:

$$b(n) = \frac{\rho}{1 + \exp(-P_e(n))}, \tag{8}$$

$$P_e(n) = \lambda P_e(n-1) + (1-\lambda)e^2(n), \tag{9}$$

where ρ is a scaling factor within the range $[0, 1]$, $P_e(n)$ represents the energy estimation of the error signal, and λ is a forgetting factor, typically set within the range $0.9 < \lambda < 1$. This adaptive adjustment ensures that the transformation function dynamically adapts to

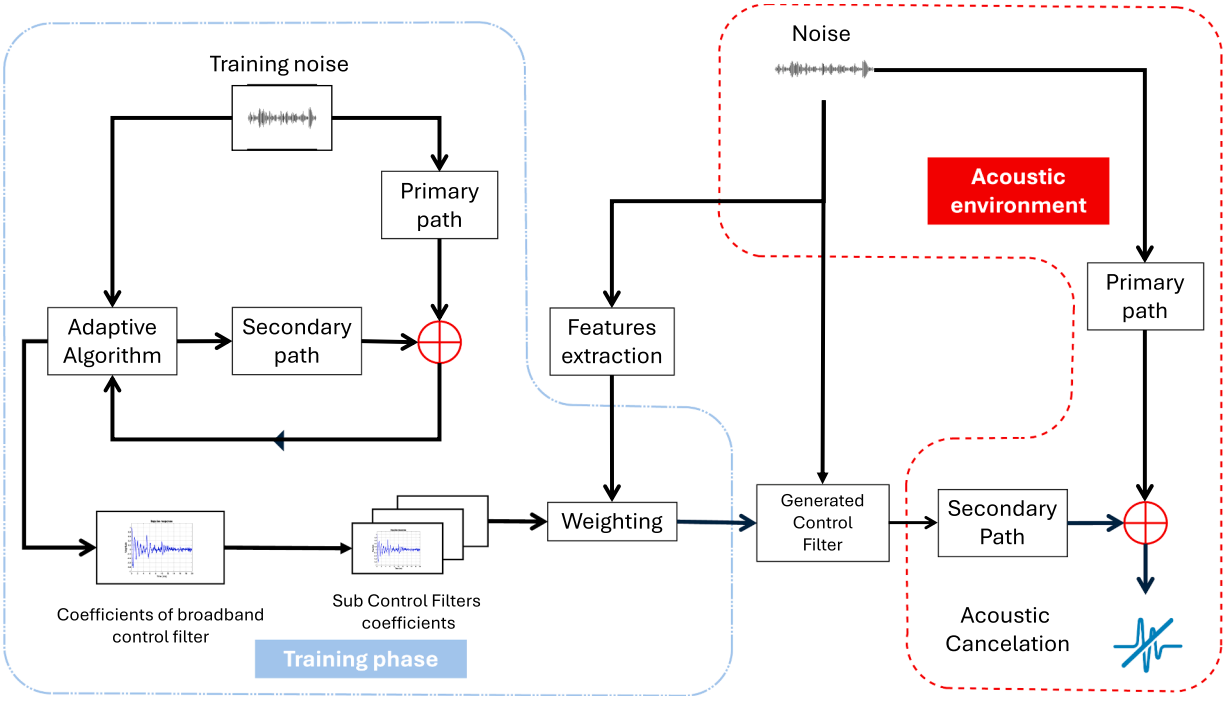


Fig. 4. Block diagram of the generative fixed-filter active noise control method, inspired by (Shi et al., 2020).

varying noise conditions, improving the robustness and stability of the THFxNLMS algorithm. A full derivation of THFxNLMS is proposed in Appendix B.

3.2. Convolutional neural network-Based generative fixed-Filter active noise control

3.2.1. GFANC Method

The Generative Fixed-Filter Active Noise Control (GFANC) method introduces a noise reduction approach by pre-training a set of broadband control filters. Based on this initial broadband control filter, a database of sub-control filters is created, allowing a weighting mechanism to dynamically generate an optimal control filter during operation (Luo et al., 2023). The method operates in two distinct stages: (1) offline filter pre-training and subdivision, and (2) real-time noise control, as illustrated in Fig. 4.

During the first phase, the target ANC system is used to cancel a broadband primary noise $d(n)$ containing the relevant frequency components. The optimal control filter, determined using the THFxNLMS algorithm, serves as the pre-trained broadband control filter. Once obtained, this filter is decomposed into sub-control filters using a practical filter decomposition technique based on the theory of perfect reconstruction filters (Dam et al., 2004). The decomposition of the broadband control filter relies on a perfect-reconstruction filter bank. Given a pre-trained control filter c with L coefficients, its frequency-domain representation is first obtained using the discrete Fourier transform (DFT):

$$G = F_L\, c$$

Where F_L is the DFT matrix. Exploiting the conjugate symmetry property of real signals in the frequency domain, the control filter is then decomposed into C sub-filters G_m, each covering a specific frequency range:

$$G = \sum_{m=1}^{C} G_m$$

Each sub-filter is defined such that its frequency components are extracted according to a perfect-reconstruction criterion, ensuring no loss of information. The time-domain representation of each sub-filter is then obtained through the inverse DFT:

$$c_m = F_L^{-1} G_m$$

Finally, the control signal can be reconstructed by summing the contributions of all sub-filters:

$$y(n) = \sum_{m=1}^{C} x^T(n) c_m$$

This decomposition method ensures that the original broadband filter is perfectly reconstructed in the time domain while allowing the generation of multiple fixed sub-filters. This process enhances flexibility in adaptive control strategies while preserving the optimal characteristics of the original filter (Luo et al., 2023). Resulting broadband filter and subfilters are presented in Fig. 5. They were computed with a sampling rate of 3 kHz and a control filter length set to 256 taps.

The 1D CNN does not model the noise directly, but learns the relationship between the spectral content of the reference signal and the optimal gain vectors for the sub-control filters. Based on these features, it generates continuous weights between 0 and 1 to combine the sub-filters and form the most suitable control filter for the input signal. This frequency-to-gain mapping clarifies that the CNN predicts optimal gains rather than the noise itself, justifying the use of synthetic training data while enabling generalization to real-world noises.

3.2.2. 1D Convolutional neural network architecture and training

Motivated by (Shi et al., 2020), the proposed 1D Convolutional Neural Network (CNN) is designed to process time-domain signals of 1-second duration, extracting relevant features to adaptively design the control filter. The input consists of a one-dimensional waveform, which is passed through a series of convolutional and fully connected layers to generate the optimal filter weights. The first convolutional layer applies 10 filters with a kernel size of 5, capturing local temporal dependencies in the signal. This is followed by a second convolutional layer with 20 filters and a larger kernel size of 32, allowing for deeper and more abstract feature extraction. A Max-Pooling operation with a kernel

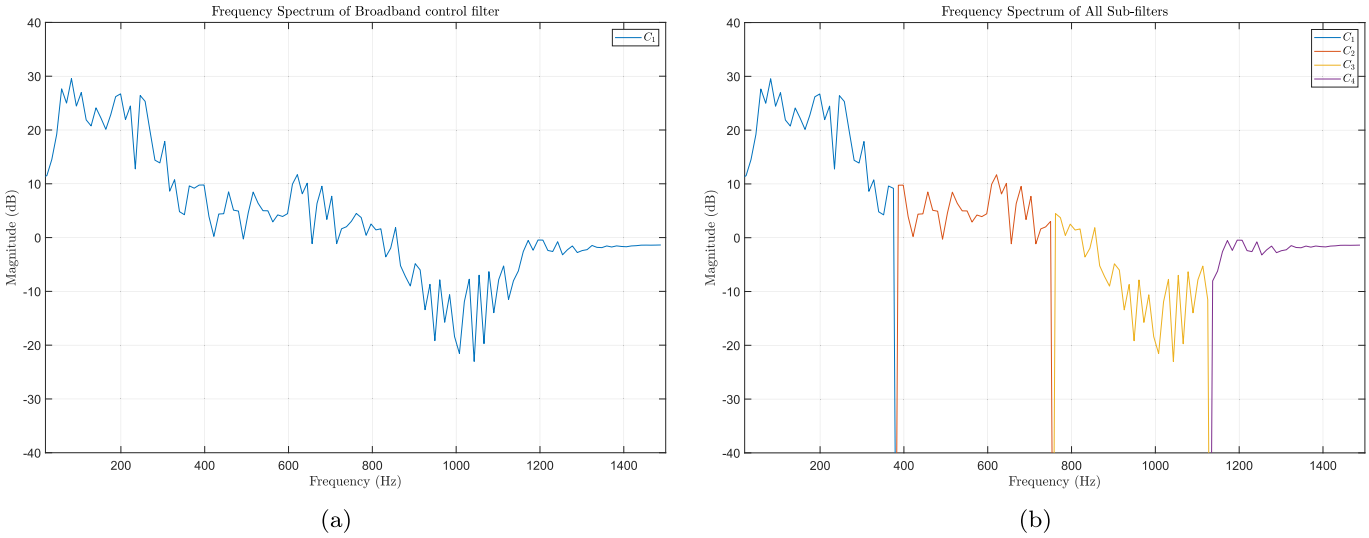


Fig. 5. Frequency responses of the broadband control filter (a) and the sub control filters (b).

Table 1
Architecture of the 1D CNN Model.

Layer	Type	Parameters	Output Shape
Conv1	Conv1D	In: 1, Out: 10, Kernel: 5, Stride: 1	(10, 15996)
Conv2	Conv1D	In: 10, Out: 20, Kernel: 32, Stride: 1	(20, 15965)
MaxPool	MaxPool1D	Kernel: 1024, Stride: 1024	(20, 15)
Flatten	Flatten	-	(40)
BatchNorm	BatchNorm1D	Num features: 40	(40)
Linear1	Fully Connected	In: 40, Out: 4	(4)
Activation	Tanh	-	(4)
Linear2	Fully Connected	In: 4, Out: 4	(4)

and stride size of 1024 is then applied to significantly reduce the temporal resolution while emphasizing the most informative components. The resulting feature map is flattened into a 40-dimensional vector and normalized using batch normalization to stabilize and accelerate training. The network then processes the extracted features through two fully connected layers: the first maps the features to a 4-dimensional space, followed by a Tanh activation function introducing non-linearity, and the second refines the output into the final set of 4 continuous values. These values correspond to the adaptive weights used to update the sub-control filters in real time. By leveraging hierarchical feature extraction and dimensionality reduction, this architecture enables the CNN to generate an effective and responsive noise control strategy, tailored to the spectral characteristics of the input signal. The detailed architecture is presented in Table 1.

The training process of the 1D CNN model is designed for a regression task, where the goal is to predict adaptive weights associated with input signals. The dataset consists of 80,000 training samples, each representing a 1-second temporal signal filtered within a randomly selected frequency range between 20 Hz and 1480 Hz. The corresponding target values are 4 continuous-valued coefficients ranging between 0 and 1, one for each subfilter. The validation set comprises 2000 samples following the same preprocessing pipeline. The model is trained using a Mean Squared Error (MSE) loss function, given the regression nature of the task. The optimizer employed is Adam, initialized with a learning rate of 0.0002 and a weight decay of 1e-4, ensuring both efficient convergence and regularization. A learning rate scheduler is also applied, reducing the learning rate by a factor of 0.5 every 5 epochs to facilitate stable optimization. The model is trained for 30 epochs using a batch size of 500, with training and validation losses monitored at each epoch. During training, the network undergoes weight initialization using Xavier initialization (Glorot & Bengio, 2010) for convolutional layers. The training loop consists of forward propagation, loss computation, backpropagation, and parameter updates, while validation is performed without gradient computation to assess generalization. In Fig. 6 the loss curves are presented.

Table 2
Computational complexity of the proposed 1D CNN model.

Component	Parameters	MACs per frame
1D CNN	6.74 K	19.33 M

To assess the real-time feasibility of the proposed framework, we analyze the computational complexity of the 1D CNN used for filter generation. The model contains a total of 6.74K trainable parameters, making it lightweight compared to typical deep ANC architectures. Despite its compact size, the convolutional layers account for most of the computational load. Using the `ptflops` library, the total number of Multiply-Accumulate operations (MACs) required for a single forward pass of a 1-second frame sampled at 3 kHz is measured as 19.33 million MACs. Table 2 summarizes the parameter count and MAC cost of the network.

Beyond raw computational complexity, memory access and buffer management are also critical factors for real-time deployment on hybrid embedded platforms such as the ADSP-SC598. Due to its small parameter footprint, the entire CNN model can be stored in on-chip memory, minimizing external memory access and associated latency. Moreover, the frame-based processing strategy allows the input buffer to be reused across successive inferences, while the generated filter coefficients are transferred only once to the real-time controller. This design limits memory bandwidth requirements and avoids frequent buffer switching, which is particularly important in heterogeneous CPU-DSP systems.

These results confirm that the proposed CNN architecture is suitable for embedded deployment.

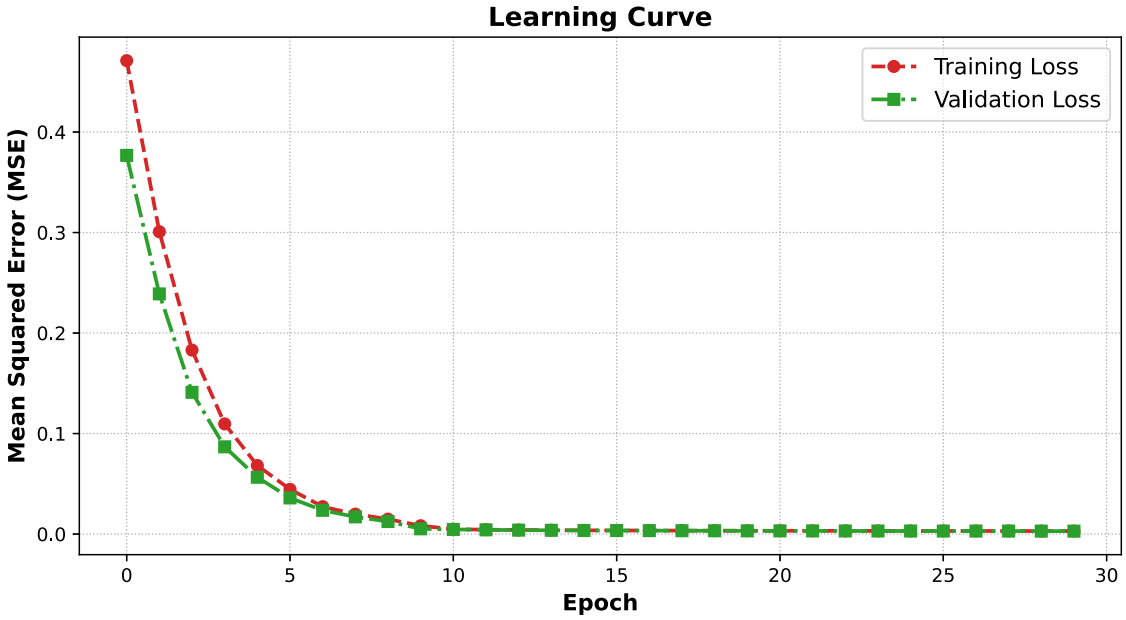


Fig. 6. Loss function curve of the 1D CNN training.

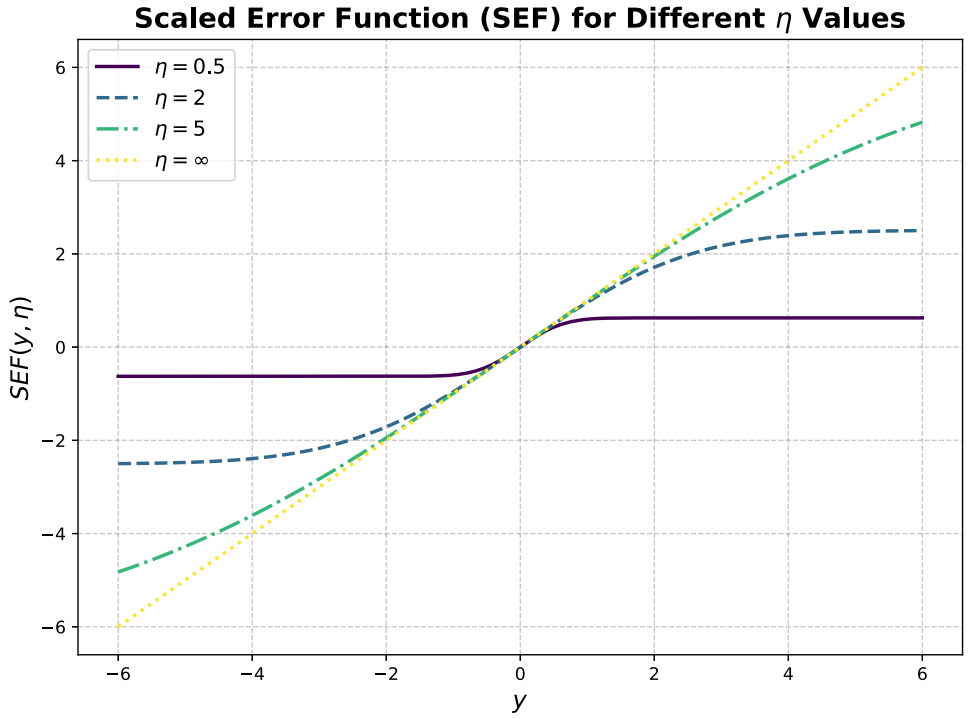


Fig. 7. Scaled error function (SEF) for different values of η.

3.2.3. Data construction

Concerning the data used to train the neural network, the dataset consisted, as previously mentioned, of 80,000 samples filtered with noise in a random frequency range for training and 2000 samples each for validation and testing. All noise signals were generated with a sampling rate of 3 kHz and a duration of 1 second. Specifically, the training samples consisted of narrow-band and broad-band noise signals with passbands randomly selected across the frequency range of interest. In the proposed approach, the labels used for supervised training correspond to adaptive gain vectors $\mathbf{w}_n^{(i)} \in \mathbb{R}^C$, which are estimated dynamically at each time step n for each sample i in a batch of size N, where C is the number of pre-trained control filters. In our case $N = 1000$ and $C = 4$.

The adaptation process follows a normalized least mean squares (NLMS) algorithm. Each control filter is fixed and stored in the matrix $\mathbf{H} \in \mathbb{R}^{C\times L}$, where L is the filter length. The input signal $x_n^{(i)}$ at time n is inserted into a rolling buffer $\mathbf{X}_d \in \mathbb{R}^{N\times L}$ that retains the past L samples for each signal in the batch. The output of the control filters is then computed as:

$$\mathbf{Y}_n = \mathbf{X}_d \cdot \mathbf{H}^\top \tag{10}$$

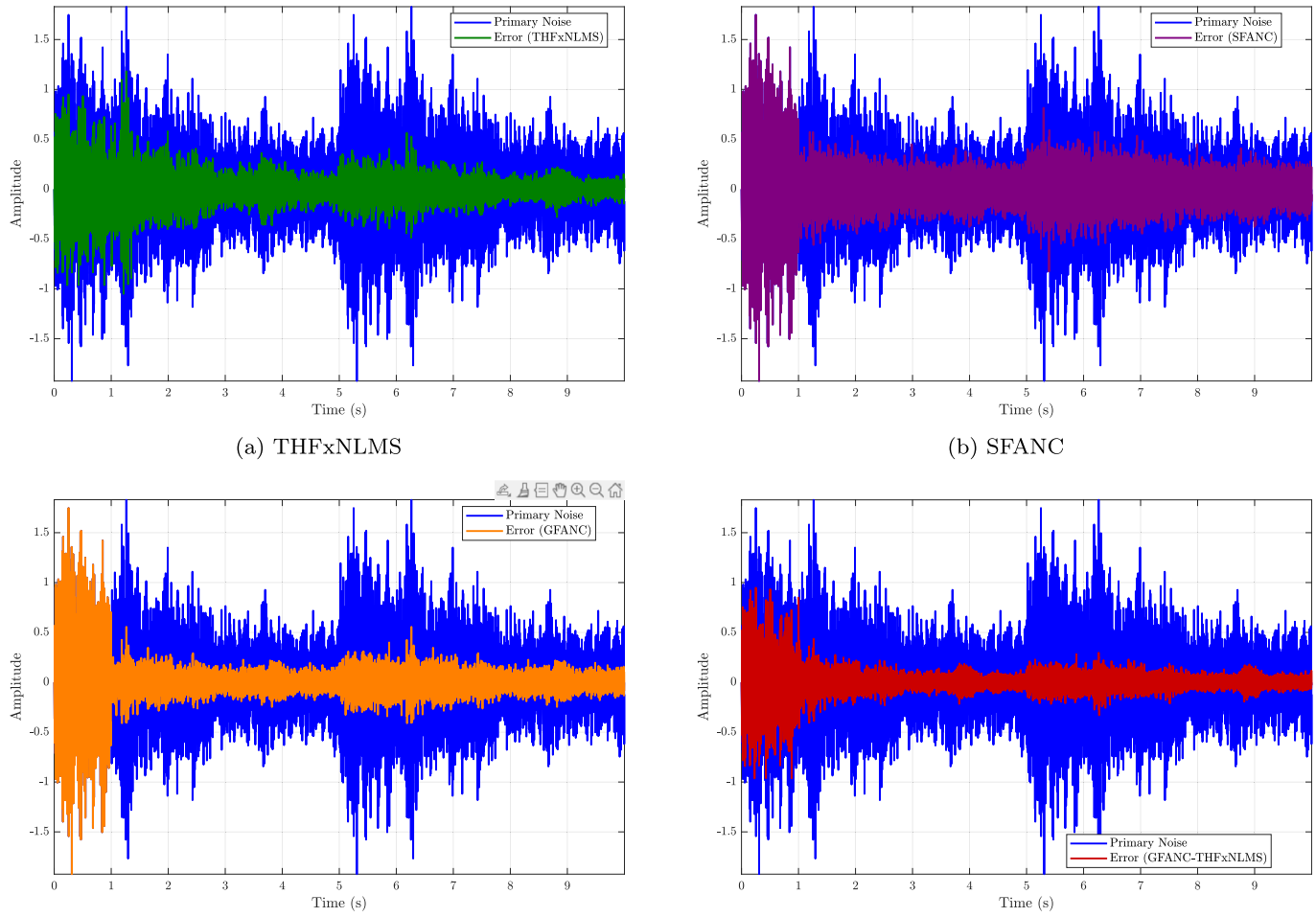


Fig. 8. Graphics of the primary noise and error signals obtained using different ANC algorithms. The control is applied to compressor noise with $\eta = 5$.

where each row $\mathbf{y}_n^{(i)} \in \mathbb{R}^C$ of $\mathbf{Y}_n$ corresponds to the filtered signal for sample i. The anti-noise signal is formed by a weighted sum of the control filter outputs:

$$\hat{d}_n^{(i)} = \mathbf{w}_n^{(i)} \cdot \mathbf{y}_n^{(i)} = \sum_{c=1}^{C} w_n^{(i,c)} \cdot y_n^{(i,c)} \tag{11}$$

and the residual error is computed as:

$$e_n^{(i)} = d_n^{(i)} - \hat{d}_n^{(i)} \tag{12}$$

where $d_n^{(i)}$ is the target disturbance. The gain vector $\mathbf{w}_n^{(i)}$ is updated using the NLMS rule:

$$\mathbf{w}_{n+1}^{(i)} = \mathbf{w}_n^{(i)} + \mu \cdot \frac{e_n^{(i)}}{\|\mathbf{y}_n^{(i)}\|^2} \cdot \mathbf{y}_n^{(i)} \tag{13}$$

with μ denoting the adaptation step size, and $\|\mathbf{y}_n^{(i)}\|^2 = \sum_{c=1}^{C} \left(y_n^{(i,c)}\right)^2$ representing the instantaneous output energy of the control filters. In our case $\mu = 0.001$.

At each iteration, the updated gain vector $\mathbf{w}_n^{(i)}$ represents the optimal linear combination (in the mean-square sense) of the control filter outputs. These vectors are recorded as training labels and paired with the corresponding system state $\mathbf{z}_n^{(i)}$, forming a dataset $\left\{\left(\mathbf{z}_n^{(i)}, \mathbf{w}_n^{(i)}\right)\right\}_{n,i}$ that is later used to train a predictive model capable of estimating $\mathbf{w}_n^{(i)}$ directly from the observed system state.

The objective of the 1D CNN is not to directly model real-world noise sources, but to learn how to optimally combine a set of pre-designed subfilters according to the spectral content of the reference signal. The network is therefore trained using synthetic band-pass noises with randomly generated passbands, exposing it to a wide range of controlled spectral profiles. For each training sample, the NLMS algorithm provides the optimal gain vector, which serves as the supervision signal and encodes the relationship between spectral content and subfilter weighting. As a result, the CNN learns a frequency-to-gain mapping that generalizes beyond the synthetic training data. Although real-world noises such as washing machines, engines, compressors, or airflow are more complex and non-stationary, their spectral characteristics remain within the range covered during training, allowing the CNN to generate appropriate subfilter weights for unseen real-world disturbances.

3.3. GFANC-THFxNLMS approach

The system processes incoming noise by sampling it at 3 kHz, capturing one second of audio per frame. As illustrated in Fig. 3, a dedicated coprocessor analyzes each frame using the GFANC framework. In this configuration, a one-dimensional convolutional neural network (1D CNN) is employed to generate a set of weights. These CNN-predicted weights are used to construct an initial optimal control filter that is tailored to the spectral characteristics of the current noise frame. Once the control filter is generated, it is transferred to the real-time controller, where the THFxNLMS algorithm takes over. Importantly, is to fine-tune the coefficients of the generated control filter during continuous operation. THFxNLMS is specifically designed to handle system nonlinearities through its nonlinear error formulation and adaptive

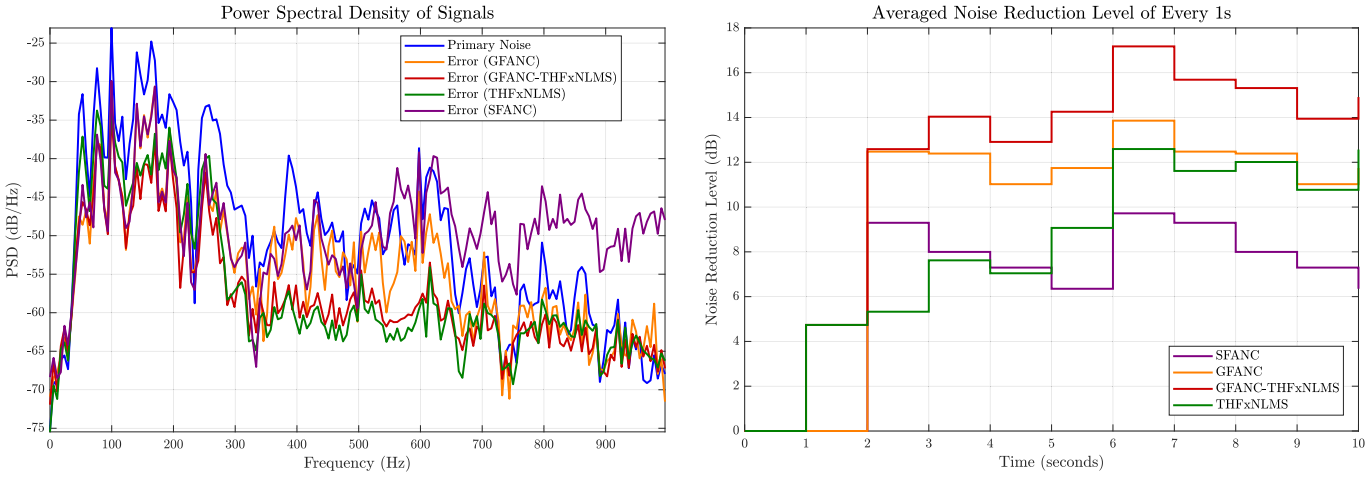


Fig. 9. Comparison of (a) PSD and (b) noise reduction per second obtained with THFxNLMS (green), GFANC (orange), SFANC (violet) and GFANC-THFxNLMS (red) with a compressor noise as a disturbance and $\eta = 5$.

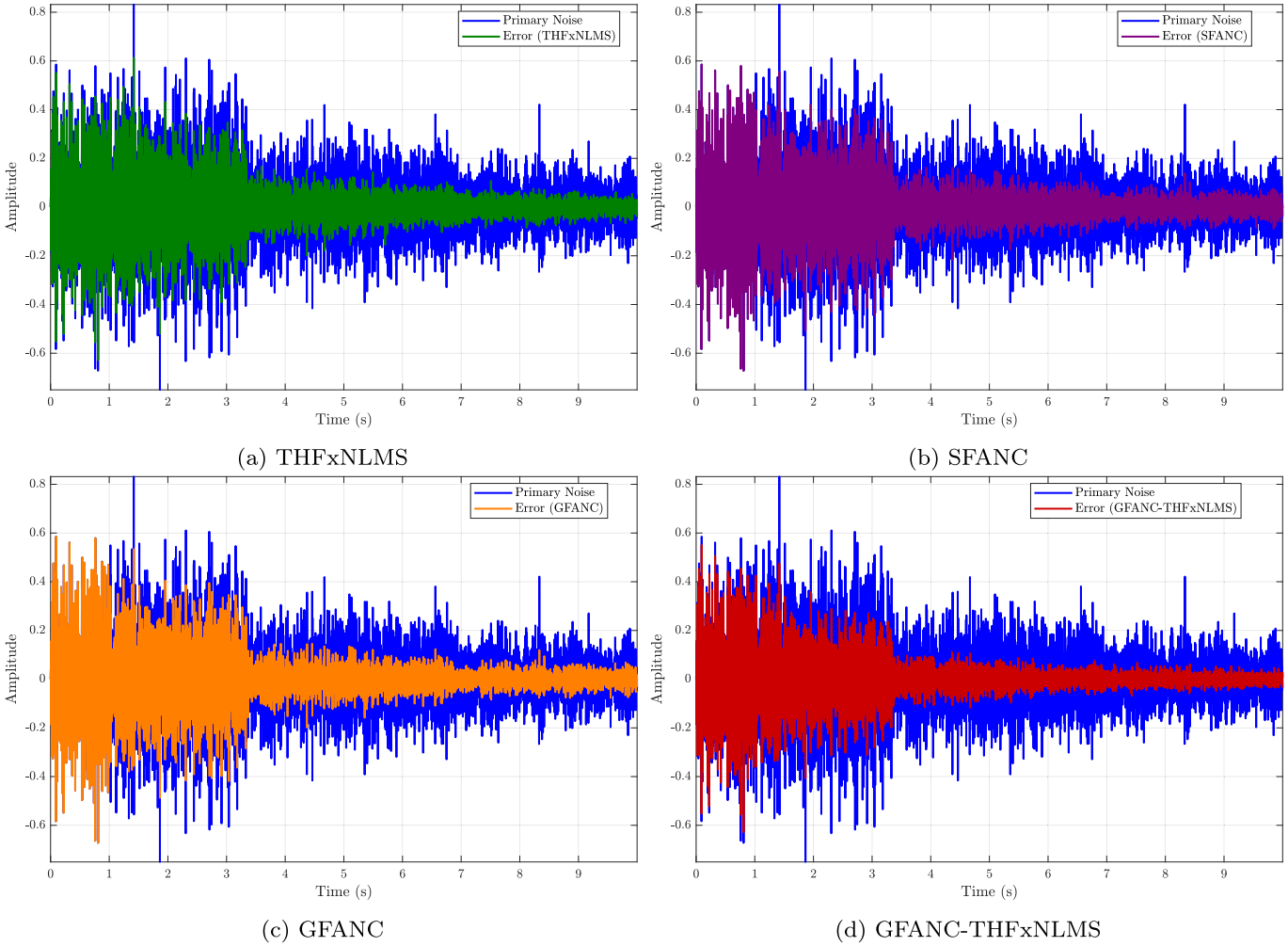


Fig. 10. Graphs of the primary noise and error signals obtained using different ANC algorithms. The control is applied to hood fan noise with $\eta = 5$, while the SNR varies over time.

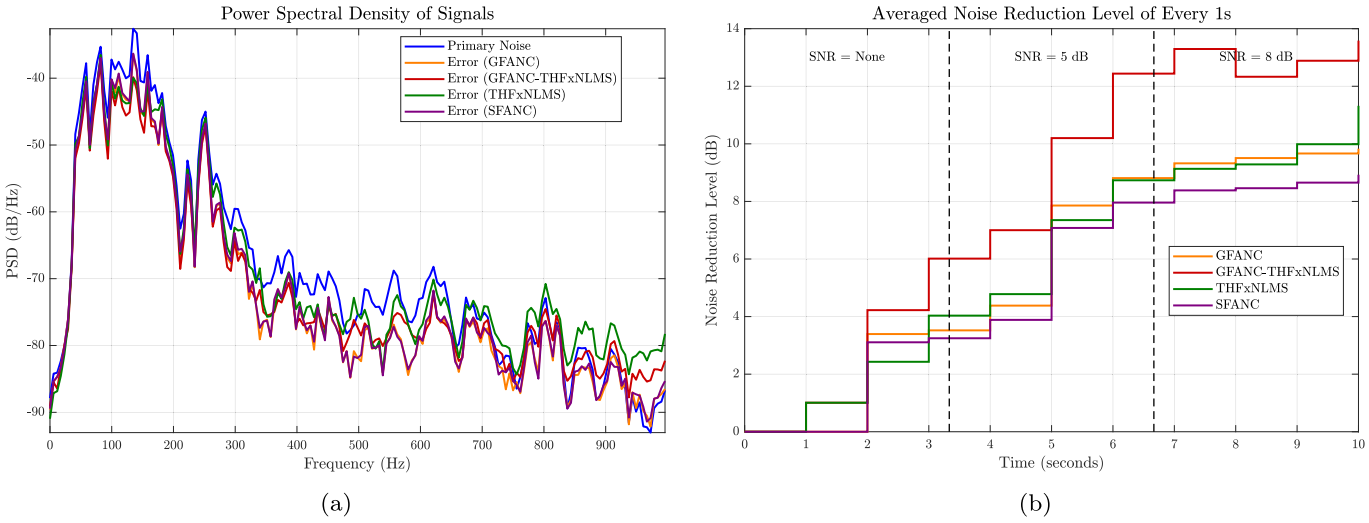


Fig. 11. Comparison of (a) PSD and (b) noise reduction per second obtained with THFxNLMS (green), GFANC (orange), SFANC (violet) and GFANC-THFxNLMS (red) with a hood fan noise as a disturbance and $\eta = 5$, while the SNR varies over time.

Table 3
Step-by-step description of the GFANC-THFxNLMS algorithm with corresponding equations.

Step Description	Corresponding Equation
1D CNN predicts weights	$\hat{\mathbf{w}}_{\text{CNN}} = \text{CNN}(x(n))$
Generated control filter	$\mathbf{w} = \hat{\mathbf{w}}_{\text{CNN}} \times \mathbf{H}$
Filtered reference signal	$\mathbf{x}'(n) = \hat{s}(n) * x(n)$
Compute controller output	$y(n) = \mathbf{w}^T(n)\mathbf{x}(n)$
Compute error signal	$e(n) = d(n) - s(n) * f_{SEF}(y(n))$
Update error power estimate	$P_e(n) = \lambda P_e(n-1) + (1-\lambda)e^2(n)$
Compute gradient scaling factor	$b(n) = \frac{\rho}{1+e^{-P_e(n)}}$
Compute weighted loss	$\text{loss}(n) = \frac{\tanh^2\left(b(n)e(n)\right)}{2 \cdot \|\mathbf{x}'(n)\|^2 + \varepsilon}$
Update filter weights	$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu\psi(e(n))\mathbf{x}'(n)$
	where $\psi(e) = \frac{\tanh(b(n)e(n))[1-\tanh^2(b(n)e(n))]}{2\|\mathbf{x}'(n)\|^2 + \varepsilon}$

gradient scaling, allowing the controller to compensate for secondary-path distortions and nonlinear loudspeaker-acoustic effects that cannot be captured at the frame level. This separation of roles is central to the proposed architecture: the CNN operates at the frame level to provide a robust and noise dependent initialization of the control filter, while THFxNLMS performs sample level adaptation to explicitly address nonlinearities and time-varying dynamics in the ANC system. By combining the generative fixed filter approach with the THFxNLMS algorithm, the system ensures dynamic adaptation, progressively improving the noise reduction performance in real time. Unlike the algorithm proposed in (Luo et al., 2022), once the initial control filter is generated, THFxNLMS continues to adapt its coefficients throughout the remainder of the signal without requiring frame-wise CNN updates. Table 3 summarizes the step-by-step procedure of the proposed algorithm, where $\mathbf{H}$ denotes the subfilters dataset and $\hat{\mathbf{w}}_{\text{CNN}}$ represents the weights predicted by the 1D CNN.

4. Results

This section presents simulations using a single-channel ANC system. The setup includes a 3 kHz sampling rate and a control filter with 256 taps. The impulse responses previously introduced define the primary and secondary paths. Additionally, 4 pre-trained sub-control filters are employed. The THFxNLMS algorithm is adopted to derive the optimal control filters with a stepsize of 0.1. Loudspeaker saturation is a primary source of nonlinearity in ANC systems (Ghasemi et al., 2016). Several studies have examined this phenomenon (Cha et al., 2023; Zhang & Wang, 2021), often modeling it using the scaled error function (SEF) (Tobias & Seara, 2006):

$$f_{\text{SEF}}(y) = \int_0^y e^{-\frac{2z}{\eta^2}}\, dz, \tag{14}$$

where y is the loudspeaker input, and η determines the nonlinearity severity. This function represents a commonly observed saturation effect, such as sound level limitation due to loudspeaker size constraints. As $\eta \to \infty$, the SEF behaves linearly, while it acts as a hard limiter when $\eta \to 0$. To assess the robustness of the proposed approach against nonlinear distortions, three loudspeaker functions are considered during training:

- $\eta = 0.5$ (severe nonlinearity),
- $\eta = 5$ (moderate nonlinearity),
- $\eta = \infty$ (linear case).

Fig. 7 highlights the trade-off introduced by the nonlinearity parameter η. Large values of η correspond to quasi-linear loudspeaker behavior, where noise cancellation is mainly limited by the adaptive filter performance. For moderate values (e.g., $\eta = 5$), saturation effects are present but remain smooth, representing a realistic operating range in which the proposed method achieves a favorable balance between robustness and control accuracy. In contrast, very small values of η (e.g., $\eta = 0.5$) introduce severe saturation, where nonlinearity becomes a dominant factor and significantly challenges conventional ANC algorithms. Fig. 7 visualizes the SEF for these different η values. The evaluation phase includes testing with both trained and untrained loudspeaker functions. It is important to note that the pre-trained sub-control filters were obtained using training noise signals that were passed through nonlinear loudspeaker functions. This ensures that each sub-filter already incorporates robustness to saturation-type nonlinearities. The 1D CNN then efficiently combines these nonlinear-aware sub-filters, while the online THFxNLMS stage further fine-tunes the generated control filter coefficients.

To evaluate the efficiency of the algorithm, non-stationary noise was generated from real-life recordings, and control was computed under different degrees of nonlinearity. Additionally, a study was conducted to assess control performance under time-varying primary paths, examining the algorithm's robustness with moderate nonlinearities ($\eta = 5$). The results obtained with GFANC-THFxNLMS are compared to those

Table 4
Average Noise Reduction (in dB) for GFANC-THFxNLMS, GFANC, THFxNLMS, and SFANC with respect to different noises and nonlinear distortions.

Noise type	Engine				Compressor				Washing Machine				Air Flow			
η	∞	5	2	0.5	∞	5	2	0.5	∞	5	2	0.5	∞	5	2	0.5
THFxNLMS	8.85	8.82	8.68	5.96	7.76	7.74	7.66	6.03	9.60	8.86	8.12	5.16	6.74	6.74	6.73	6.52
SFANC	7.22	7.10	7.02	5.22	5.06	5.90	5.20	4.78	7.66	7.98	5.30	4.96	6.74	6.74	6.77	7.02
GFANC	7.41	7.40	7.32	5.33	6.14	6.14	6.11	5.11	7.95	7.60	7.50	5.35	8.01	8.01	8.00	7.79
GFANC-THFxNLMS	**11.44**	**11.42**	**11.30**	**7.07**	**10.27**	**10.25**	**10.16**	**7.60**	**10.64**	**10.18**	**9.12**	**6.41**	**9.74**	**9.73**	**9.73**	**9.49**

Table 5
Complexity analysis of the THFxNLMS algorithm.

Operation	Dimensions	$*$ Multiplications	$+ / -$ Additions/Subtractions
$x_f(n) = \hat{s}(n) * x(n)$	$\mathbb{R}^{L\times 1}$	L	1
$y(n) = \mathbf{w}^T(n)\mathbf{x}(n)$	$\mathbb{R}^{L\times 1}$	L	$L-1$
$e(n) = d(n) - s(n) * f_{SEF}(y(n))$	$\mathbb{R}$	1	1
$P_e(n) = \lambda P_e(n-1) + (1-\lambda)e^2(n)$	Scalar	2	1
$b(n) = \frac{\rho}{1+e^{-P_e(n)}}$	Scalar	2	1
$\text{loss}(n) = \frac{\tanh^2\left(b(n)e(n)\right)}{2\cdot\|\mathbf{x}'(n)\|^2+\varepsilon}$	Scalar	6	3
$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu\psi(e(n))\mathbf{x}'(n), \quad \psi(e) = \frac{\tanh(b(n)e(n))\left[1-\tanh^2(b(n)e(n))\right]}{2\|\mathbf{x}'(n)\|^2+\varepsilon}$	$\mathbb{R}^{L\times 1}$	$4L+2$	$3L+1$
Total		$8L+11$	$4L+7$

of the THFxNLMS, Selective Fixed Filter Active Noise Control (SFANC) presented in (Shi et al., 2022) and GFANC algorithms.

4.1. ANC performance

To evaluate the algorithm's efficiency in noise control within the noise box, it was first tested using non-stationary noise sources, including sounds from a washing machine, engine, compressor, and airflow. Fig. 8 shows the results for washing machine noise. Fig. 8 presents signals in time domain and Fig. 9 illustrates the Power Spectral Density (PSD) of the primary noise and the controlled signals for all algorithms, as well as the average noise reduction achieved per second by each algorithm.

Fig. 8 illustrates the performance of various algorithms, highlighting that GFANC outperforms THFxNLMS for this noise type. This advantage significantly influences the effectiveness of GFANC-THFxNLMS. Notably, GFANC-THFxNLMS achieves lower error amplitudes due to its adaptive behavior. Additionally, GFANC introduces control after a delay of 1 second, allowing the CNN to analyze primary noise characteristics and determine the most appropriate controller. In contrast, GFANC-THFxNLMS applies control immediately, leveraging its adaptive properties.

Fig. 9 displays the Power Spectral Density (PSD) of each controlled signal up to 1000 Hz. The GFANC-THFxNLMS-controlled signal (red curve) exhibits the lowest levels. To evaluate its performance, it can be compared to the PSD of the GFANC controlled signal (yellow curve). In the 50–300 Hz range, the red curve remains either below or at the same level as the yellow one. Overall, GFANC-THFxNLMS demonstrates superior performance across most frequencies up to 1000 Hz, particularly in handling resonances and anti-resonances.

Fig. 9 illustrates the noise reduction achieved over time, measured at each second. The results show that GFANC and GFANC-THFxNLMS reach optimal noise reduction after 2 seconds. After 10 seconds of computation, THFxNLMS achieves a noise reduction similar to that of GFANC. This figure highlights two key aspects: first, the faster convergence time provided by GFANC, and second, the potential for improved noise reduction when effectively combining the characteristics of both algorithms. SFANC present an interesting convergence speed but obtain a lower noise reduction than GFANC. Table 4 presents the results obtained with different algorithms, noise types, and non-linearity conditions.

Table 4 reports the average noise reduction achieved by the three algorithms in different noise types and levels of nonlinear distortion. In all cases, the hybrid approach GFANC-THFxNLMS consistently outperforms both GFANC, SFANC and THFxNLMS individually, demonstrating the benefit of combining both methods. It is worth noting that GFANC occasionally yields lower noise reduction than THFxNLMS, which can be attributed to the initial one-second delay required by GFANC before it starts applying control, resulting in temporarily reduced effectiveness at the beginning of the signal.

4.2. ANC robustness

To assess the algorithm's robustness, a 10-second hood fan signal was used, with the SNR of the primary path varying over time. The SNR remained constant during the first third of the signal, was set to 5 dB for the second third, and then increased to 8 dB in the final third. Fig. 10 presents the results in the time domain, while Figs. 11a and 11b display the PSD of the controlled signals for each algorithm and the noise reduction achieved per second, respectively.

Regarding time-domain signals, the control performance appears similar across the three algorithms, but the lowest error amplitude is observed with GFANC-THFxNLMS. This observation is further supported by Fig. 11, which shows that GFANC-THFxNLMS achieves the best attenuation. Similarly, Fig. 11b clearly demonstrates that GFANC-THFxNLMS provides the most effective noise reduction. In the frequency domain, it is evident that the control achieved with GFANC-THFxNLMS is superior at low frequencies, particularly between 100 and 350 Hz. Regarding noise reduction per second, both in time-invariant and time-varying cases, it can be observed that, over time, the control obtained with THFxNLMS converges to that of GFANC. With longer-duration signals, THFxNLMS could potentially surpass GFANC. This highlights the benefit of merging GFANC and THFxNLMS, leveraging the strengths of both algorithms in terms of convergence speed and adaptability. However, a limitation in time-varying systems is that if the SNR variation becomes more significant, the control performance may degrade. The average noise reduction achieved by the algorithms is as follows: 3.63 dB for the THFxNLMS algorithm, 3.35 dB for GFANC, 3.12 dB for SFANC and 4.83 dB for GFANC-THFxNLMS.

Table 6
Complexity analysis of the SFANC algorithm. The 1D CNN outputs a scalar index identifying the most appropriate control filter. C_{CNN} denotes the computational cost of the CNN, and L is the filter length.

Operation	Dimensions	$*$ Multiplications	$+$ / $-$ Additions/Subtractions
1D CNN (scalar index prediction)	Input: $\mathbb{R}^{1\times1\times f_s}$	C_{CNN}	C_{CNN}
Filter selection: $\mathbf{w} = \mathbf{W}_c[i]$	$i \in \mathbb{R}$ (scalar index)	0	0
$x_f(n) = \hat{s}(n) * x(n)$	$\mathbb{R}^{L\times1}$	L	1
$y(n) = \mathbf{w}^T(n)\mathbf{x}(n)$	$\mathbb{R}^{L\times1}$	L	$L-1$
$e(n) = d(n) - s(n) * f_{SEF}(y(n))$	$\mathbb{R}$	1	1
Total		$C_{\text{CNN}} + 2L + 2$	$C_{\text{CNN}} + L + 2$

Table 7
Complexity analysis of the GFANC algorithm. $\hat{\mathbf{w}}_{\text{CNN}}$ denotes the weights predicted by the 1D CNN, C_{CNN} is its computational cost, and $\mathbf{H}$ represents the subfilters dataset. C represents the number of subfilters, and L represents the length of each filter.

Operation	Dimensions	$*$ Multiplications	$+$ / $-$ Additions/Subtractions
1D CNN	Input: $\mathbb{R}^{1\times1\times f_s}$	C_{CNN}	C_{CNN}
$\mathbf{w} = \hat{\mathbf{w}}_{\text{CNN}} \times \mathbf{H}$	$\mathbb{R}^{1\times C} \times \mathbb{R}^{C\times L}$	$C \times L$	$C \times (L-1)$
$x_f(n) = \hat{s}(n) * x(n)$	$\mathbb{R}^{L\times1}$	L	1
$y(n) = \mathbf{w}^T(n)\mathbf{x}(n)$	$\mathbb{R}^{L\times1}$	L	$L-1$
$e(n) = d(n) - s(n) * f_{SEF}(y(n))$	$\mathbb{R}$	1	1
Total		$C_{\text{CNN}} + CL + 2L + 1$	$C_{\text{CNN}} + C(L-1) + L + 1$

Table 8
Complexity analysis of the GFANC-THFxNLMS algorithm. $\hat{\mathbf{w}}_{\text{CNN}}$ denotes the weights predicted by the 1D CNN, C_{CNN} is its computational cost, and $\mathbf{H}$ represents the subfilters dataset. C represents the number of subfilters, and L represents the length of each filter.

Operation	Dimensions	$*$ Multiplications	$+$ / $-$ Additions/Subtractions
1D CNN	Input: $\mathbb{R}^{1\times1\times f_s}$	C_{CNN}	C_{CNN}
$\mathbf{w} = \hat{\mathbf{w}}_{\text{CNN}} \times \mathbf{H}$	$\mathbb{R}^{1\times C} \times \mathbb{R}^{C\times L}$	$C \times L$	$C \times (L-1)$
$x_f(n) = \hat{s}(n) * x(n)$	$\mathbb{R}^{L\times1}$	L	1
$y(n) = \mathbf{w}^T(n)\mathbf{x}(n)$	$\mathbb{R}^{L\times1}$	L	$L-1$
$e(n) = d(n) - s(n) * f_{SEF}(y(n))$	$\mathbb{R}$	1	1
$P_e(n) = \lambda P_e(n-1) + (1-\lambda)e^2(n)$	Scalar	2	1
$b(n) = \frac{\rho}{1+e^{-P_e(n)}}$	Scalar	2	1
$\text{loss}(n) = \frac{\tanh^2\left(b(n)e(n)\right)}{2 \cdot \|\mathbf{x}'(n)\|^2 + \varepsilon}$	Scalar	6	3
$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu\psi(e(n))\mathbf{x}'(n), \quad \psi(e) = \frac{\tanh(b(n)e(n))\left[1-\tanh^2(b(n)e(n))\right]}{2\|\mathbf{x}'(n)\|^2 + \varepsilon}$	$\mathbb{R}^{L\times1}$	$4L + 2$	$3L + 1$
Total		$C_{\text{CNN}} + CL + 8L + 11$	$C_{\text{CNN}} + CL + 4L + 8$

4.3. Computational complexity

In this section, we present a comparative analysis of the computational complexity of THFxNLMS, SFANC, GFANC, and GFANC-THFxNLMS. The computational complexity of the four algorithms is detailed in Tables 5–8. The THFxNLMS algorithm exhibits a relatively moderate complexity, with $8L + 11$ multiplications and $4L + 7$ additions/subtractions per iteration. The SFANC algorithm also integrates a pre-trained 1D CNN, its output is a scalar index to select the most appropriate pre-trained control filter. The total complexity per sample of SFANC amounts to $C_{\text{CNN}} + 2L + 2$ multiplications and $C_{\text{CNN}} + L + 2$ additions/subtractions. The GFANC algorithm integrates a pre-trained 1D CNN to predict weights to generate the control filter. Although the network is not updated during operation, the inference step still introduces a non-negligible computational cost called C_{CNN}. Combined with the matrix multiplication $\hat{\mathbf{w}}_{\text{CNN}} \times \mathbf{H}$, the total complexity amounts to $C_{\text{CNN}} + CL + 2L + 1$ multiplications and $C_{\text{CNN}} + C(L-1) + L + 1$ additions/subtractions. The GFANC-THFxNLMS algorithm combines the CNN-based subfilter generation with the adaptive refinement of the THFxNLMS update. As a result, its complexity is higher, with $C_{\text{CNN}} + CL + 8L + 11$ multiplications and $C_{\text{CNN}} + CL + 4L + 8$ additions/subtractions. This hybrid structure increases the computational load but leads to improved noise reduction performance compared to each component used individually. Although the number of MACs for the CNN inference is relatively low (19.33 M per frame), memory transfers for the CNN weights ($\hat{\mathbf{w}}_{\text{CNN}}$) and buffer management for frame-based processing are critical in real hardware. Efficient on-chip storage and reuse of input buffers help minimize memory latency and bandwidth issues in embedded implementations.

5. Conclusion

This study applies active noise control in an encapsulated structure with an opening using the GFANC-THFxNLMS algorithm. The GFANC-THFxNLMS algorithm follows a deep learning-based approach designed to enhance active noise control. The core strategy involves creating a dataset of multiple pre-trained sub-control filters derived from a broadband control filter. A 1D CNN dynamically generates the most suitable controller for the target noise, while an adaptive algorithm fine-tunes the coefficients of the chosen filter in real time. The primary goal of this algorithm is twofold: to improve efficiency compared to traditional adaptive algorithms and to address the limitations of GFANC in handling nonlinearities, particularly those introduced by the loudspeaker. These nonlinearities, which become more pronounced at higher amplitudes, can degrade control performance, making a more flexible and adaptive approach necessary. The results demonstrate that in an environment where such distortions are present, GFANC-THFxNLMS outperforms both GFANC and THFxNLMS. Specifically, Table 4 shows that GFANC-THFxNLMS achieves an average noise reduction of up to 11.44 dB for Engine noise, representing an improvement of approximately 2.59 dB over THFxNLMS, 4.22 dB over SFANC and 4.03 dB over GFANC. For

other noise types, similar numerical improvements are observed, e.g., a gain of 3.13 dB over THFxNLMS for compressor noise and 2.69 dB for washing machine noise. This hybrid approach enhances adaptability and convergence, ensuring effective noise reduction even under non-ideal conditions. However, the algorithm has certain limitations: when variations in the primary path coincide with significant SNR changes, control effectiveness diminishes, highlighting areas for further improvement. Moreover, when nonlinearities become too strong, the achievable noise reduction may be significantly limited, regardless of the control strategy used.

Looking ahead, deep learning could evolve from a supporting tool into the primary driver of control, directly generating the anti-noise signal. This shift would be particularly valuable for managing nonlinearities related to the environment and electroacoustic phenomena associated with the materials used in noise control systems. However, several challenges remain for deploying such systems in real-time applications. One major limitation concerns the large number of parameters in deep architectures, which can conflict with the very low-latency requirements of active noise control. In addition, the adaptive nature of ANC must be preserved, meaning that the algorithm should be able to adjust dynamically to new and unseen acoustic situations that were not encountered during training. These aspects highlight the need for lightweight and adaptive architectures capable of maintaining stability and real-time performance while dealing with the inherent variability of acoustic environments.

CRediT authorship contribution statement

Alkahf Aboutiman: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Khaled Said Ahmed Maamoun:** Conceptualization, Validation, Formal analysis, Resources, Data curation, Writing – review & editing; **Hamid Reza Karimi:** Conceptualization, Validation, Writing – original draft, Writing – review & editing, Supervision; **Francesco Ripamonti:** Conceptualization, Validation, Writing – original draft, Writing – review & editing, Supervision.

Data availability

Data will be made available on request.

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.

Acknowledgments

The work was supported by the project IN-NOVA: Active reduction of noise transmitted into and from enclosures through encapsulated structures, which has received funding from the European Union's Horizon Europe programme under the Marie Sklodowska-Curie grant agreement no. 101073037.

Appendix A. Notation Table

Table A.1
Notation used throughout the manuscript.

Symbol	Description
L	Length of the control filter (number of taps)
C	Number of sub-filters in the GFANC or GFANC-THFxNLMS algorithms
G	Frequency-domain representation of the pre-trained control filter
$x(n)$	Reference signal vector at discrete time n
$x'(n)$	Filtered reference signal vector at discrete time n
$y(n)$	Output of the control filter at discrete time n
$e(n)$	Residual error signal at discrete time n
$d(n)$	Primary disturbance signal at discrete time n
$s(n)$	Impulse response of the secondary path at discrete time n
$\hat{s}(n)$	Estimate of the secondary path impulse response at discrete time n
$w(n)$	Coefficient vector of the control filter at discrete time n
$w(n+1)$	Updated control filter coefficient vector at discrete time n
$P(z)$	Transfer function of the primary path
$W(z)$	Transfer function of the control filter
$S(z)$	Transfer function of the secondary path
$f(x)$	Nonlinear transformation function (hyperbolic tangent) used in THFxNLMS
$f_{\text{SEF}}(y)$	Scaled Error Function modeling loudspeaker saturation: $f_{\text{SEF}}(y) = \int_0^y e^{-2z/\eta^2}\, dz$
η	Parameter determining the severity of loudspeaker nonlinearity in SEF
b	Compression factor for the robust transformation function
$b(n)$	Adaptive compression factor at discrete time n
$\psi(e(n))$	Robust transformation function used in the THFxNLMS weight update
$J(n)$	Loss function at discrete time n
$P_e(n)$	Error power estimate at discrete time n
λ	Forgetting factor used in the error power estimation
ρ	Scaling factor for the adaptive compression parameter $b(n)$
μ	Step size of the adaptive algorithm
ϵ	Small constant to ensure numerical stability during weight update
D_s	Group delay of the secondary path
$\lambda_{\max}$	Maximum eigenvalue of the autocorrelation matrix of $x'(n)$
C_{CNN}	Computational cost of the 1D CNN
$\mathbf{H} \in \mathbb{R}^{C\times L}$	Dataset of pre-trained sub-filters
$\mathbf{w}_n^{(i)} \in \mathbb{R}^C$	Gain vector for sample i at time n, used to weight sub-filters
$\mathbf{X}_d \in \mathbb{R}^{N\times L}$	Rolling buffer storing past L samples for each of N signals in a batch
$\mathbf{Y}_n \in \mathbb{R}^{N\times C}$	Outputs of all control filters at time n for all N samples
$\mathbf{y}_n^{(i)} \in \mathbb{R}^C$	Row vector of $\mathbf{Y}_n$ for sample i, filtered sub-filter outputs
$\hat{d}_n^{(i)}$	Anti-noise signal formed by weighted sum of sub-filter outputs for sample i at time n
$e_n^{(i)}$	Residual error for sample i at time n
N	Batch size for training the CNN model
$\mathbf{z}_n^{(i)}$	Observed system state for sample i at time n, used as input to predictive model

Appendix B. Full Derivation of the THFxNLMS Algorithm

The THFxNLMS algorithm defines the loss function as:

$$J(n) = \frac{f^2(b(n)e(n))}{2\|x'(n)\|^2 + \epsilon}, \tag{B.1}$$

where

$$f(x) = \tanh(x),$$

b is the compression factor, $x'(n)$ is the filtered reference signal, and ϵ is a small positive constant to prevent division by zero. The residual error is:

$$e(n) = d(n) - s(n) * w^T(n)x(n), \tag{B.2}$$

where $d(n)$ is the primary disturbance and $s(n)$ is the secondary path impulse response.

The gradient of the loss with respect to the filter coefficients $w(n)$ is computed using the chain rule. Let

$$g(e(n)) = f(b(n)e(n)) = \tanh(b(n)e(n)).$$

Then the derivative of $g^2(e(n))$ with respect to $e(n)$ is:

$$\frac{\partial g^2(e(n))}{\partial e(n)} = 2\,g(e(n))g'(e(n)) = 2\tanh(b(n)e(n)) \cdot b(n)\left[1 - \tanh^2(b(n)e(n))\right].$$

Applying the chain rule to the full loss function $J(n)$:

$$\begin{aligned}\nabla J(n) &= \frac{\partial J(n)}{\partial w(n)} = \frac{\partial J(n)}{\partial e(n)} \cdot \frac{\partial e(n)}{\partial w(n)} \\ &= \frac{2b(n)\tanh(b(n)e(n))\left[1 - \tanh^2(b(n)e(n))\right] \cdot (-x'(n))}{2\|x'(n)\|^2 + \epsilon}.\end{aligned} \tag{B.3}$$

We define the robust transformation function:

$$\psi(e(n)) = \frac{\tanh(b(n)e(n))\left[1 - \tanh^2(b(n)e(n))\right]}{\|x'(n)\|^2 + \epsilon}, \tag{B.4}$$

so that the weight update rule can be written as:

$$w(n+1) = w(n) - \mu\psi(e(n))x'(n). \tag{B.5}$$

To ensure stability of the adaptive update, the step size μ must satisfy:

$$0 < \mu < \frac{1}{\lambda_{\max} D_s}, \tag{B.6}$$

where

$$R_{x'} = E\left[x'(n)x'^T(n)\right], \quad \lambda_{\max} = \max \operatorname{eig}(R_{x'}), \tag{B.7}$$

is the maximum eigenvalue of the autocorrelation matrix of the filtered reference signal $x'(n)$, and D_s is the secondary path group delay. This condition guarantees that the nonlinear transformation $\psi(e(n))$ does not destabilize the weight update.

To improve robustness against nonlinearities and varying noise conditions, the compression factor b is adapted using:

$$b(n) = \frac{\rho}{1 + \exp(-P_e(n))}, \tag{B.8}$$

$$P_e(n) = \lambda P_e(n-1) + (1-\lambda)e^2(n), \tag{B.9}$$

where ρ is a scaling factor in $[0, 1]$ and λ is a forgetting factor. This ensures that $\psi(e(n))$ dynamically adjusts to the current error level, enhancing stability and robustness in varying noise conditions.

References

Aboutiman, A., Rachman, Z., Oberman, T., Aletta, F., Kang, J., Karimi, H. R., & Ripamonti, F. (2025a). Subjective perception analysis of active noise control algorithms in an encapsulated structure: An experimental study. *Applied Acoustics*, *239*, 110823. https://doi.org/10.1016/j.apacoust.2025.110823

Aboutiman, A., Shams, R., Karimi, H. R., Ripamonti, F., & Pawełczyk, M. (2025b). Active noise control in encapsulated structures with non-minimum phase characteristics using a kalman filter approach. *Journal of Sound and Vibration*, *615*, 119187. https://doi.org/10.1016/j.jsv.2025.119187

Analog Devices (2024). ADSP-SC598 Product Page. Accessed: 2024-12-29 https://www.analog.com/en/products/adsp-sc598.HTML.

Bambang, R. T. (2008). Adjoint EKF learning in recurrent neural networks for nonlinear active noise control. *Applied Soft Computing*, *8*(4), 1498–1504. https://doi.org/10.1016/j.asoc.2007.10.017

Bouchard, M., Paillard, B., & Le Dinh, C. T. (1999). Improved training of neural networks for the nonlinear active control of sound and vibration. *IEEE Transactions on Neural Networks*, *10*(2), 391–401. https://doi.org/10.1109/72.750568

Cha, Y.-J., Choi, W., Suh, G., Mahmoudkhani, S., & Büyüköztürk, O. (2018). Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. *Computer-Aided Civil and Infrastructure Engineering*, *33*(9), 731–747. https://doi.org/10.1111/mice.12334

Cha, Y.-J., Mostafavi, A., & Benipal, S. S. (2023). Dnoisenet: Deep learning-based feedback active noise control in various noisy environments. *Engineering Applications of Artificial Intelligence*, *121*, 105971. https://doi.org/10.1016/j.engappai.2023.105971

Chang, C.-Y., & Luoh, F.-B. (2007). Enhancement of active noise control using neural-based filtered-x algorithm. *Journal of Sound and Vibration*, *305*(1–2), 348–356. https://doi.org/10.1016/j.jsv.2007.04.007

Dam, H. H., Nordholm, S., Cantoni, A., & de Haan, J. M. (2004). Iterative method for the design of DFT filter bank. *IEEE Transactions on Circuits and Systems II: Express Briefs*, *51*(11), 581–586. https://doi.org/10.1109/TCSII.2004.836041

Das, D. P., & Panda, G. (2004). Active mitigation of nonlinear noise processes using a novel filtered-s LMS algorithm. *IEEE Transactions on Speech and Audio Processing*, *12*(3), 313–322. https://doi.org/10.1109/TSA.2003.822741

Duan, Y., Yisheng, L. V., & Wang, F.-Y. (2016). Travel time prediction with LSTM neural network. In *2016 IEEE 19Th international conference on intelligent transportation systems (ITSC)* (pp. 1053–1058). IEEE. https://doi.org/10.1109/ITSC.2016.7795686

Elliott, S., Stothers, I., & Nelson, P. (1987). A multiple error LMS algorithm and its application to the active control of sound and vibration. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *35*(10), 1423–1434. https://doi.org/10.1109/TASSP.1987.1165044

Elzaghmouri, B. M., Jbara, Y. H. F., Elaiwat, S., Innab, N., Osman, A. A. F., Ataelfadiel, M. A. M., Zawaideh, F. H., Alawneh, M. F., Al-Khateeb, A., & Abu-Zanona, M. (2024). A novel hybrid architecture for superior iot threat detection through real iot environments. *Computers, Materials & Continua*, *81*(2).

Fu, R., Zhang, Z., & Li, L. (2016). Using LSTM and GRU neural network methods for traffic flow prediction. In *2016 31St youth academic annual conference of chinese association of automation (YAC)* (pp. 324–328). IEEE. https://doi.org/10.1109/YAC.2016.7804912

Fuller, C. R., Hansen, C. H., & Snyder, S. D. (1991). Active control of sound radiation from a vibrating rectangular panel by sound sources and vibration inputs: An experimental comparison. *Journal of Sound and Vibration*, *145*(2), 195–215. https://doi.org/10.1016/0022-460X(91)90587-A

Ghasemi, S., Kamil, R., & Marhaban, M. H. (2016). Nonlinear thf-fxlms algorithm for active noise control with loudspeaker nonlinearity. *Asian Journal of Control*, *18*(2), 502–513. https://doi.org/10.1002/asjc.1140

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 249–256). JMLR Workshop and Conference Proceedings.

Goodwin, G. C., Silva, E. I., & Quevedo, D. E. (2010). Analysis and design of networked control systems using the additive noise model methodology. *Asian Journal of Control*, *12*(4), 443–459. https://doi.org/10.1002/asjc.201

Guo, X., Li, Y., Jiang, J., Dong, C., Du, S., & Tan, L. (2018). Sparse modeling of nonlinear secondary path for nonlinear active noise control. *IEEE Transactions on Instrumentation and Measurement*, *67*(3), 482–496. https://doi.org/10.1109/TIM.2017.2781992

Hartmann, W. M. (2004). Signals, sound, and sensation. Springer Science & Business Media.

Isaac, C. W., Wrona, S., Pawelczyk, M. et al. (2022). Modelling vibro–acoustic response of lightweight square aluminium panel influenced by sound source locations for active control. *Scientific Reports*, *12*(1), 10727. https://doi.org/10.1038/s41598-022-14951-y

Jeong, I., & Park, Y. (2023). Suboptimal controller design of global active noise control system for various acoustic environments. *Scientific Reports*, *13*(1), 5453. https://doi.org/10.1038/s41598-023-32261-9

Jun, S., Ya-li, Z., & Qi-zhi, Z. (2012). Arc-tangent transformation algorithm for active impulsive noise control. *Noise and Vibration Control*, *32*(2), 27. https://doi.org/10.3969/j.issn.1006-1355-2012.02.006

Krukowicz, T. (2010). Active noise control algorithm based on a neural network and nonlinear input-output system identification model. *Archives of Acoustics*, *35*(2), 191–202.

Kuo, S. M., & Morgan, D. R. (1996). Active noise control systems (vol. 4). Wiley, New York.

Kuo, S. M., & Wu, H.-T. (2005). Nonlinear adaptive bilinear filters for active noise control systems. *IEEE Transactions on Circuits and Systems I: Regular Papers*, *52*(3), 617–624. https://doi.org/10.1109/TCSI.2004.842429

Lam, B., Gan, W.-S., Shi, D., Nishimura, M., & Elliott, S. (2021). Ten questions concerning active noise control in the built environment. *Building and Environment*, *200*, 107928. https://doi.org/10.1016/j.buildenv.2021.107928

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), 10021. https://doi.org/10.1038/s41598-020-66563-z

Lashkari, K. (2006). A novel volterra-wiener model for equalization of loudspeaker distortions. In *2006 IEEE International conference on acoustics speech and signal processing proceedings* (pp. 1–5). IEEE (*vol. 5*). https://doi.org/10.1109/ICASSP.2006.1661226

Li, C., Jin, G., Liu, H., & Li, J. (2023). Active impulsive noise control algorithm based on adjustable hyperbolic tangent function. *Circuits, Systems, and Signal Processing*, *42*(9), 5559–5578. https://doi.org/10.1007/s00034-023-02374-7

Liang, H., Chen, L., Pan, Y., & Lam, H.-K. (2022). Fuzzy-based robust precision consensus tracking for uncertain networked systems with cooperative–antagonistic interactions. *IEEE Transactions on Fuzzy Systems*, *31*(4), 1362–1376. https://doi.org/10.1109/TFUZZ.2022.3200730

Luo, Z., Ji, J., Wang, B., Shi, D., Ma, H., & Gan, W.-S. (2025). Deep learning-based generative fixed-filter active noise control: Transferability and implementation. *Mechanical Systems and Signal Processing*, *238*, 113207.

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. https://doi.org/10.1109/LSP.2022.3169428

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 (icassp)* (pp. 1–5). IEEE. https://doi.org/10.1109/ICASSP49357.2023.10095205

Maamoun, K. S. A., Wrona, S., Pawelczyk, M., & Karimi, H. R. (2024). Shaping of the frequency response of vibrating plates with openings for vibro-acoustic systems. *Mechanical Systems and Signal Processing*, *218*, 111539. https://doi.org/10.1016/j.ymssp.2024.111539

Maamoun, K. S. A., Wrona, S., Pawelczyk, M., & Karimi, H. R. (2025). Optimizing design of openings in vibrating plates for enhanced vibro-acoustic performance using a genetic algorithm approach. *Journal of Sound and Vibration*, *598*, 118847. https://doi.org/10.1016/j.jsv.2024.118847

Manolakis, D. G., Ingle, V. K., & Kogon, S. M. (2005). Spectral estimation, signal modeling, adaptive filtering, and array processing. *ARTECH HOUSE: Boston, MA, USA*, .

Meng, H., & Chen, S. (2020). A modified adaptive weight-constrained fxLMS algorithm for feedforward active noise control systems. *Applied Acoustics*, *164*, 107227. https://doi.org/10.1016/j.apacoust.2020.107227

Misol, M., Algermissen, S., & Monner, H. P. (2012). Experimental investigation of different active noise control concepts applied to a passenger car equipped with an active windshield. *Journal of Sound and Vibration*, *331*(10), 2209–2219. https://doi.org/10.1016/j.jsv.2012.01.001

Murao, T., Nishimura, M., Sakurama, K., & Nishida, S.-i. (2016). Basic study on active acoustic shielding (improving the method to enlarge the AAS window). *Mechanical Engineering Journal*, *3*(1), 15–00322. https://doi.org/10.1299/mej.15-00322

Napoli, R., & Piroddi, L. (2009). Nonlinear active noise control with NARX models. *IEEE Transactions on Audio, Speech, and Language Processing*, *18*(2), 286–295. https://doi.org/10.1109/TASL.2009.2025798

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. https://doi.org/10.1109/TASLP.2020.2989582

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*, 108317.

Sivasakthi, D. A., Sathiyaraj, A., & Devendiran, R. (2024). Hybridrobustnet: Enhancing detection of hybrid attacks in iot networks through advanced learning approach. *Cluster Computing*, *27*(4), 5005–5019.

Snyder, S. D., & Tanaka, N. (1995). Active control of vibration using a neural network. *IEEE Transactions on Neural Networks*, *6*(4), 819–828. https://doi.org/10.1109/72.392246

Sun, J., Zhang, H., Wang, Y., & Sun, S. (2020). Fault-tolerant control for stochastic switched IT2 fuzzy uncertain time-delayed nonlinear systems. *IEEE Transactions on Cybernetics*, *52*(2), 1335–1346. https://doi.org/10.1109/TCYB.2020.2997348

Tan, K., & Wang, D. (2019). Learning complex spectral mapping with gated convolutional recurrent networks for monaural speech enhancement. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *28*, 380–390. https://doi.org/10.1109/TASLP.2019.2955276

Tao, J., Wang, S., Qiu, X., & Pan, J. (2016). Performance of an independent planar virtual sound barrier at the opening of a rectangular enclosure. *Applied Acoustics*, *105*, 215–223. https://doi.org/10.1016/j.apacoust.2015.12.019

Tobias, O. J., & Seara, R. (2006). On the LMS algorithm with constant and variable leakage factor in a nonlinear environment. *IEEE Transactions on Signal Processing*, *54*(9), 3448–3458. https://doi.org/10.1109/TSP.2006.879274

Tokhi, M. O., & Wood, R. (1997). Active noise control using radial basis function networks. *Control Engineering Practice*, *5*(9), 1311–1322. https://doi.org/10.1016/S0967-0661(97)84370-5

Vetterli, M., & Le Gall, D. (1989). Perfect reconstruction FIR filter banks: Some properties and factorizations. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *37*(7), 1057–1071. https://doi.org/10.1109/29.32283

Wang, S., Yu, J., Qiu, X., Pawelczyk, M., Shaid, A., & Wang, L. (2017). Active sound radiation control with secondary sources at the edge of the opening. *Applied Acoustics*, *117*, 173–179. https://doi.org/10.1038/s41598-017-13546-2

World Health Organization (2018). Environmental Noise Guidelines for the European Region: Executive Summary. World Health Organization, Regional Office for Europe.

Wrona, S., Pawelczyk, M., & Cheng, L. (2022). Sound transmission through a panel with a hybrid active and semi-active control system. *Journal of Sound and Vibration*, *536*, 117172. https://doi.org/10.1016/j.jsv.2022.117172

Wu, L., He, H., & Qiu, X. (2010). An active impulsive noise control algorithm with logarithmic transformation. *IEEE Transactions on Audio, Speech, and Language Processing*, *19*(4), 1041–1044. https://doi.org/10.1109/TASL.2010.2061227

Zhang, H., & Wang, D. (2021). Deep ANC: A deep learning approach to active noise control. *Neural Networks*, *141*, 1–10. https://doi.org/10.1016/j.neunet.2021.03.037

Zhang, Q.-Z., Gan, W.-S., & Zhou, Y.-l. (2006). Adaptive recurrent fuzzy neural networks for active noise control. *Journal of Sound and Vibration*, *296*(4–5), 935–948. https://doi.org/10.1016/j.jsv.2006.03.020

Zhou, Y., Zhang, Q., & Yin, Y. (2015). Active control of impulsive noise with symmetric α-stable distribution based on an improved step-size normalized adaptive algorithm. *Mechanical Systems and Signal Processing*, *56*, 320–339. https://doi.org/10.1016/j.ymssp.2014.10.002