A Meta-learning Method to Accelerate the Convergence of Adaptive Active Noise Control: Implementation of Modified Model-Agnostic Meta-learning Algorithm in MATLAB

Boxiang Wang[1]

The topic of accelerating the convergence of the adaptive filter has been a persistent concern within the field of adaptive signal processing. Although some advanced approaches have been developed to tackle this obstacle, their computational complexity continues to be a substantial barrier, thereby compromising their feasibility for application in practical situations. Hence, this article implements a Model-Agnostic Meta-Learning (MAML) method, which can seek the best initialization for the adaptive filter to increase its convergence. This study employs this strategy to expedite the convergence of the adaptive active noise control (ANC) system in attenuating aircraft noise. The experimental results demonstrate that the new method achieves significantly better convergence than the standard approach without requiring more computations for the adaptive algorithm. Furthermore, this article offers in-depth guidance on implementing the modified MAML algorithm in Matlab, with potential use in other adaptive signal processing algorithms. The code is available on GitHub and MathWork.

1 Introduction

Active Noise Control (ANC) is an advanced technique that reduces unwanted noise by using controlled anti-noise signals to cancel out the primary noise through destructive interference, applying the superposition principle in wave physics [1], [2]. The standard ANC system includes the reference microphone, the secondary source, and the error microphone, as depicted in Figure 1. The reference microphone captures the reference signal $x(n)$ and sends it to the controller for processing, which then produces the control signal $y(n)$. The control signal will prompt the secondary source to generate an anti-noise wave $y'(n)$, reducing the disturbance $d(n)$ at the error microphone's position. The error microphone will monitor the noise reduction performance and offer input on the remaining error to the controller. Compared to the traditional passive approach, the ANC system is effective in reducing low-frequency noise and usually has a compact size, which leads to its widespread use in different fields, such as headphones [3]–[8], automotive industries, and window ventilation [9]–[17], to address noise issues.

With the progression of adaptive filter theory, a multitude of adaptive algorithms have been devised to address active noise cancellation. The derivative algorithms improve the adaptive capabilities of the ANC system in response to variations in ambient noise and acoustic conditions [18]–[37]. One of the algorithms that holds considerable significance in the implementation of the real-time ANC system is the filtered reference least mean square (FxLMS) algorithm due to its remarkable computational efficiency. In the resolution of numerous engineering challenges, practical implementations of its derivative algorithms are also widespread.

However, these active control algorithms based on the least mean square (LMS) [38], [39] class still encounter an inherent issue: their sluggish convergence impacts the adaptive ANC system's ability to effectively reduce dynamic noise. This slow convergence issue not only influences the ANC

[1]BOXIANG001@e.ntu.edu.sg

system's response time but also impairs the customers' perception. Variable step-size approaches have been developed to enhance convergence speed by dynamically modifying the updating rate of the adaptive algorithm. However, this strategy lacks generalization for different applications. Although some other modified algorithms improve the response speed of the ANC system to rapidly varying noise [40]–[50], they are still slow to cope with some dynamic and nonstationary noises. The Recursive Least Squares filter (RLS) [51] and Kalman filter [52] are recognized for their excellent capability in tracking dynamic and nonstationary noise as compared to LMS-based approaches in this scenario. However, their high computational demands make them challenging to implement on a real-time processor [53], [54]. Hence, finding a practical approach to accelerate the convergence of the adaptive algorithm becomes more meaningful.

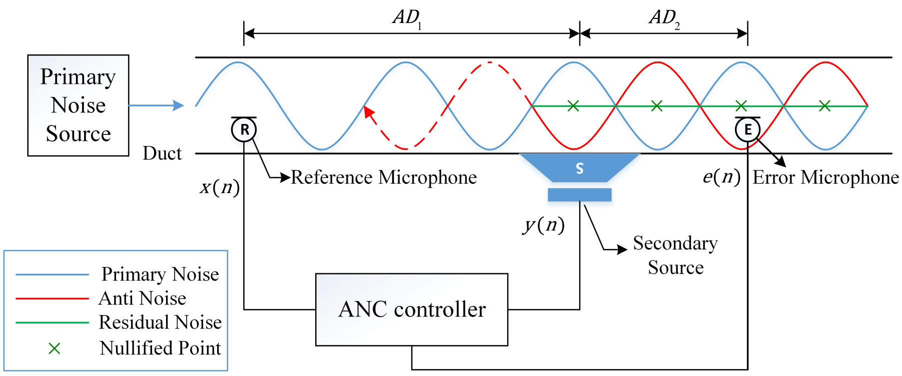

Figure 1: The feedforward active noise control (ANC) system used in the noise duct.

Meanwhile, deep-learning approaches have prevailed across industries since their excellent performance in object detection and classification, natural language processing and generation, etc. With their advanced development, these approaches were also devised for audio and speech applications, such as sound classification, event detection, and localization [55]–[58]. During these methods, meta-learning algorithms have recently been used to determine deep neural work's learning rate, structures, and initialization. Inspired by these algorithms, the previous works proposed a modified Model-Agnostic Meta-learning method dedicated to the adaptive filters [59], [60]. The modified MAML algorithm brings the forgetting factor to alleviate the zero-input issue in practical adaptive filter applications. This article offers a comprehensive overview of how to implement the modified MAML algorithm in MATLAB. This approach is used to help a standard FxLMS algorithm achieve adequate convergence behavior in reducing raw aircraft noise. The numerical simulation results demonstrate its effectiveness.

The following sections are organized as follows: Section 2.1 and 2.2 briefly introduce the theoretical foundation of the adaptive active noise control approach and the MAML algorithm. Section 4 provides a detailed explanation of the MAML code and exhibits the experimental results. Finally, Section 5 concludes this article.

2 Basic theoretical principle introduction

A brief description of the filtered reference least mean square (FxLMS) and modified model-agnostic meta-learning (MAML) algorithms is presented in this section.

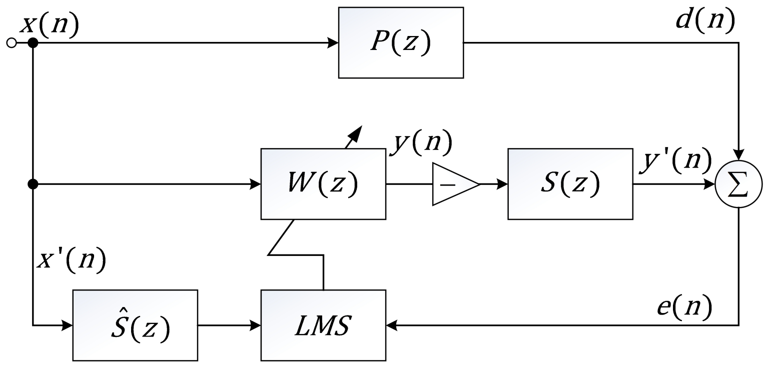

Figure 2: Block diagram of the filter reference least mean square (FxLMS) algorithm, where $P(z)$, $W(z)$, and $S(z)$ denote the z-domain expression of the primary path $\mathbf{p}(n)$, the control filter $\mathbf{w}(n)$, and the secondary path $\mathbf{s}(n)$; $S(z)$ stands for the transfer function of the secondary path estimate.

2.1 Filtered reference least mean square (FxLMS) algorithm

This letter considers a feedforward ANC system using a finite-impulse adaptive filter, as shown in Fig. 2, whose reference sensor is placed close to the primary noise source for simplicity. However, the analysis is also applicable to the case where the reference sensor is placed further from the primary noise source. The reference signal $x(n)$ propagates through the primary path $p(n)$ to form the disturbance $d(n)$, while the control filter $w(n)$ processes the reference signal to generate the control signal $y(n)$, which goes through the secondary path $s(n)$ to cancel this disturbance at the error sensor. The error signal is expressed as

$$e(n) = d(n) - \left[\mathbf{w}^{\mathrm{T}}(n)\mathbf{x}(n)\right] * s(n) \tag{1}$$

where $*$ denotes linear convolution. The control filter, with N taps, and the reference vector are given by

$$\begin{aligned} \mathbf{w}(n) &= \begin{bmatrix} w_1(n) & \cdots & w_i(n) & \cdots & w_N(n) \end{bmatrix}^{\mathrm{T}}, \\ \mathbf{x}(n) &= \begin{bmatrix} x(n) & \cdots & x(n-N+1) \end{bmatrix}^{\mathrm{T}}. \end{aligned} \tag{2}$$

To minimize the mean square of terms in (1), we can utilize the filtered-x least mean square algorithm (FxLMS) to update the new control filter as

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n)\mathbf{x}'(n), \tag{3}$$

where μ represents the step size, and the filtered reference vector is obtained from

$$\mathbf{x}'(n) = \mathbf{x}(n) * s(n). \tag{4}$$

The implementation code of the FxLMS algorithm is detailed and introduced in Section 4.2.

2.2 Modified Model-Agnostic Meta-learning method

This section presents an enhanced Model-Agnostic Meta-learning (MAML) approach for determining the initial weights of the control filter. This method improves the convergence speed of the FxLMS algorithm for handling dynamic noise.

However, most adaptive algorithms will receive null inputs during the initial iteration, inevitably impacting the algorithm's convergence performance. Hence, the conventional MAML approach is not applicable for directly initializing the adaptive filter. The modified MAML technique incorporates a forgetting factor to address the zero-input problem of the adaptive filter, thereby resolving the issue at hand. Its loss function is defined as

$$\mathbb{L}(k) = \sum_{i=0}^{N-1} \lambda^i J(k-i), \tag{5}$$

where λ and N denote the forgetting factor $\lambda \in (0,1]$ and the length of the control filter. It is important to mention that k denotes the inner index of the MAML algorithm ($k = 0, 1, \cdots, N-1$). In the equation, the cost function of ANC is expressed as

$$J(k) = e^2(k) = \left[d(k) - \mathbf{w}^{\mathrm{T}}(n)\mathbf{x}'(k)\right]^2, \tag{6}$$

where $d(k)$ and $\mathbf{x}'(k)$ are randomly sampled from the set of pre-measured disturbance tracks $\{d_1(n), d_2(n), \cdots\}$ and set of filter reference signals $\{\mathbf{x}'_1(n), \mathbf{x}'_2(n), \cdots\}$ under different configuration settings of ANC. n stands for the time index. The filtered vector $\mathbf{x}'(k-i)$ is obtained from

$$\mathbf{x}'(k-i) = \begin{bmatrix} x'(k-i) & x'(k-i-1) & \cdots & \mathbf{0}_{1\times i} \end{bmatrix}^{\mathrm{T}}. \tag{7}$$

According to the gradient descent method, the new initial value of the control filter is derived as

$$\mathbf{\Phi}(n+1) = \mathbf{\Phi}(n) - \frac{1}{2}\varepsilon\frac{\partial \mathbb{L}(k)}{\partial \mathbf{\Phi}(n)}, \tag{8}$$

in which ε represents the learning rate. Here, we assumed that the best initial value **uses one step to reach the optimal solution**:

$$\mathbf{w}(n) = \mathbf{\Phi}(n) + \mu e^{\dagger}(k)\mathbf{x}'(k). \tag{9}$$

The error signal $e^{\dagger}(k)$ is given by

$$e^{\dagger}(k) = d(k) - \mathbf{\Phi}^{\mathrm{T}}(n)\mathbf{x}'(k). \tag{10}$$

By ignoring the second derivative, the gradient in (8) can be derived as

$$\frac{\partial \mathbb{L}(k)}{\partial \mathbf{\Phi}(n)} = -2\sum_{i=0}^{N-1} \lambda^i e(k-i)\mathbf{x}'(k-i). \tag{11}$$

Therefore, the recursive formula of $\mathbf{\Phi}$ can be derived as

$$\mathbf{\Phi}(n+1) = \mathbf{\Phi}(n) + \varepsilon\sum_{i=0}^{N-1} \lambda^i e(k-i)\mathbf{x}'(k-i), \tag{12}$$

where the error signal based on the new control filter is obtained form

$$e(k-i) = d(k-i) - \mathbf{w}^{\mathrm{T}}(n)\mathbf{x}'(k-i). \tag{13}$$

The recursive formula stated in equation (12) is commonly referred to as the modified MAML algorithm, whose pseudocode is shown in Table 1. It is important to mention that n in the previous paragraph represents both the time index and the epoch index of the MAML algorithm. Each epoch consists of N iterations.

The entire progress of the modified MAML algorithm is shown in Figure 3. The one epoch of this algorithm consists of three stages.

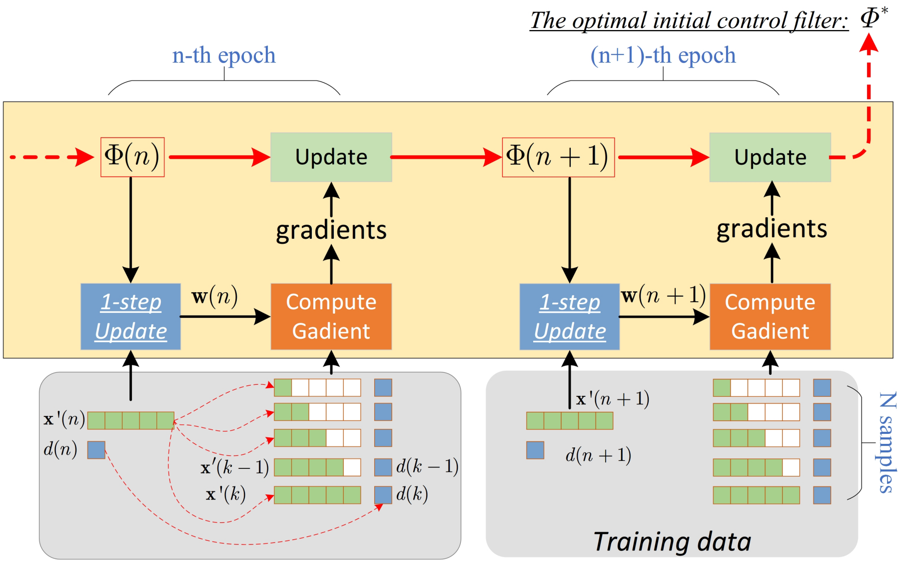

Figure 3: Block diagram of the modified MAML algorithm progress.

Table 1: Pseudocode of the proposed modified model-agnostic meta-learning

Algorithm: Modified Model-Agnostic Meta-Learning

1: Randomly initialize $\mathbf{\Phi}(0)$, and set $n = 0$
2: **for** n to M **do**
3: Randomly sample the filtered reference vector $\mathbf{x}'(k)$ and the disturbance $\mathbf{d}(k)$ from the recorded sample set: $\{(\mathbf{x}'_1(n), \mathbf{d}_1(n)), (\mathbf{x}'_2(n), \mathbf{d}_2(n)), \cdots\}$
4: Get the error signal based on the initial control filter: $e^{\dagger}(k) = d(k) - \mathbf{\Phi}^{\mathrm{T}}(n)\mathbf{x}'(k)$
5: Obtain the control filter: $\mathbf{w}(n) = \mathbf{\Phi}(n) + \mu e^{\dagger}(k)\mathbf{x}'(k)$
6: Get the error signal based on the new control filter:
for i = 0 to N-1 **do**
$e(k-i) = d(k-i) - \mathbf{w}^{\mathrm{T}}(n)\mathbf{x}'(k-i)$
end for
7: Update the initial value : $\mathbf{\Phi}(n+1) = \mathbf{\Phi}(n) + \varepsilon \sum_{i=0}^{N-1} \lambda^i e(k-i)\mathbf{x}'(k-i)$
8: **end for**

- **Stage 1**: The initial control filter vector $\mathbf{\Phi}(n)$ will take one step to update the pseudo optimal control filter $\mathbf{w}(n)$.

- **Stage 2**: The pseudo-optimal control filter $\mathbf{w}(n)$ will be used to cancel the disturbance samples with their corresponding filtered reference vector and get the gradients.

- **Stage 3**: These gradients are used to update the initial control filter vector $\mathbf{\Phi}(n)$ so that to get the new value $\mathbf{\Phi}(n+1)$.

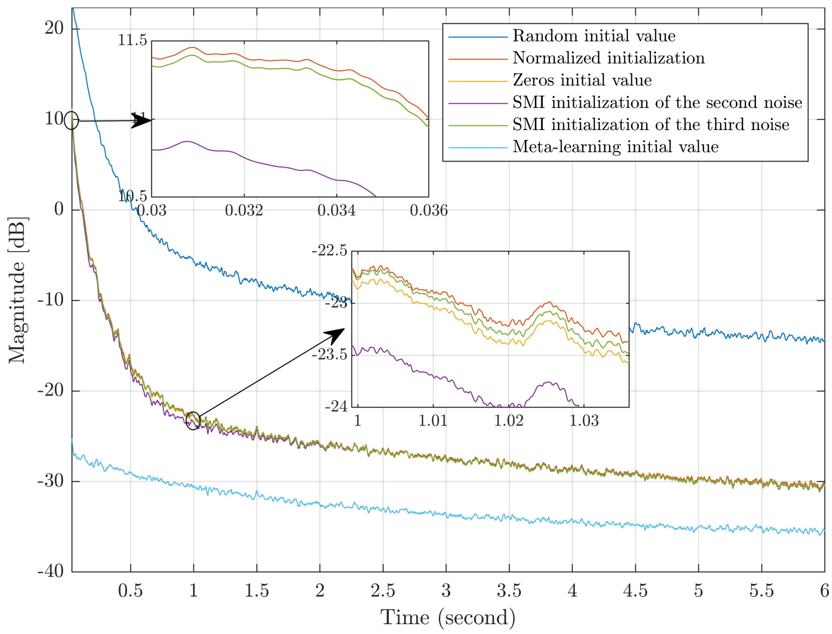

Figure 4: The mean square error of FxLMS with different initial control filters. The curves of zeros initialization and SMI initialization of the third noise overlap at the beginning.

In that way, the program continues to the next epoch $(n + 1)$ until the algorithm converges. Furthermore, the implementation code of this modified MAML algorithm is detailed and introduced in Section 4.1.

3 Experimental results

This section introduces the numerical simulation of the modified MAML algorithm to demonstrate its effectiveness. The first experiment shows the comparative studies between the MAML method and other approaches, and the second experiment exhibits the results of aircraft noise cancellation based on the MAML method.

3.1 Comparison with other initial approaches

In this simulation, the primary and secondary paths are measured from an air duct. The control filter and secondary path estimate have 512 and 256 taps, respectively. The sampling rate and step size for the FxLMS algorithm are set to 16 kHz and 0.0003, respectively. Furthermore, three distinct broadband noises are used as the primary noise. Their frequency bands range from 800 Hz to 2 kHz, 1.6 kHz to 4.4 kHz, and $4 - 6$ kHz. The sample data set consists of these noises separated into two sets: training (70%) and testing (30%).

In contrast to the meta-learning approach, which employs a forgetting factor of 0.95, we examined the following four initialization strategies: 1). an initial value of zero; 2). an initial value generated at random; 3) an initial value derived using the Sample Matrix Inversion (SMI) technique; and 4) normalized initialization.

Figure 4 shows the mean square error of the FxLMS algorithms using these specific initial control filters to reduce the first broadband noise in the testing set. The initial control determined using

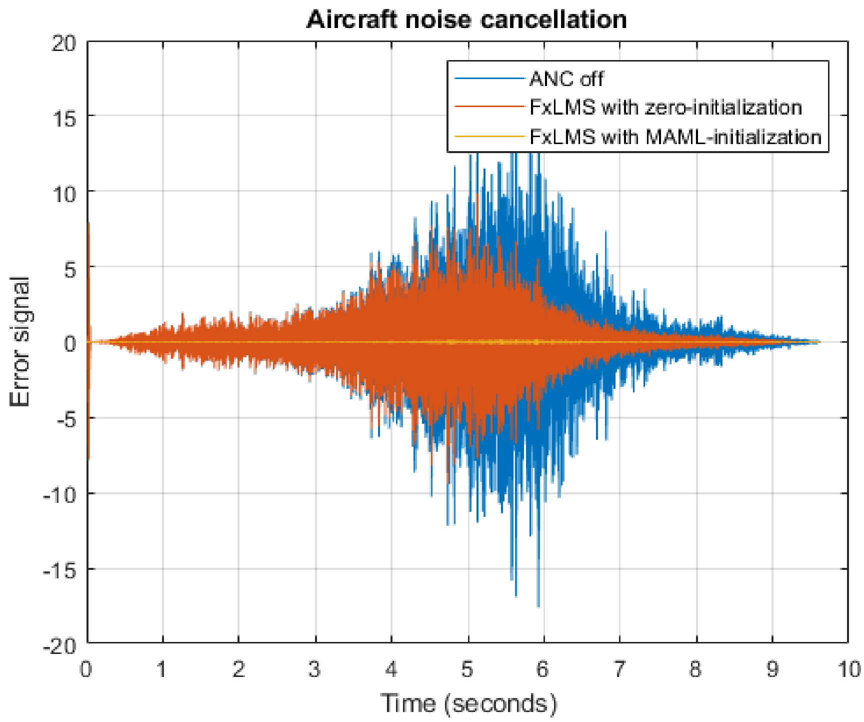

Figure 5: Aircraft noise reduction performance of the FxLMS algorithms with the zero and MAML initializations.

meta-learning demonstrates superior convergence performance compared to the other methods evaluated.

3.2 Real dynamic noise cancellation

The simulation utilized the FxLMS method with different initial values to eliminate aircraft in the measured paths. The control filter consists of 512 taps, whereas the secondary path estimate has 256 taps. The step size is set to 0.0002 and the forgetting factor is set to 0.95. The sampling rate remains set at 16 kHz.

Figure 5 shows the error signal of the FxLMS method, which utilizes the identical initial control filter to address both aircraft noise. The control filter derived from the meta-learning method can help FxLMS achieve the quickest convergence performance. Conversely, the classic FxLMS method struggles to manage amplitude-varying sounds when initialized with zeros.

It is important to note that the following paragraph elaborates on the implementation details of this experiment.

4 Code Explanation

The section provides a concise introduction to all the MATLAB codes used to validate the proposed modified MAML algorithm. The project contains three main programs.

- **MAML_Nstep_forget.m**: the code of the modified MAML algorithm.

- **FxLMS.m**: the code of the conventional single-channel FxLMS algorithm.
- **Main_tst_function.m**: the main program that validates the performance of the modified MAML algorithm.

Additionally, the primary and secondary paths, which have been measured, are used for the main numerical simulation. These paths can be found in the **Path** folder. The main source of noise in the testing program is the sound produced by aircraft, which is stored in the file named **707_Sound_for_Simulation.mat**.

4.1 Modified MAML algorithm

The program of **MAML_Nstep_forget.m** implements the proposed modified MAML algorithm, as outlined in Table 1. Table 1 shows the pseudocode of the proposed approach. It is worth noting that the random sampling method is completed during the preparation of the dataset. However, the code provided fully implements steps 4 to 7, clearly described in the code comments. The definitions of the variables in the program are illustrated in Table 2.

Algorithm 4.1: Main steps of Modified MAML algorithm

- *Building the input vectors from the randomly sampled pair:* $\{\mathbf{x}'(n), \mathbf{d}(n)\}$ *. Here,* $k = n$

$$\begin{cases} \mathbf{x}'(k-i) & = \begin{bmatrix} x'(n-i) & x'(n-i-1) & \cdots & \mathbf{0}_{1\times i} \end{bmatrix}^{\mathrm{T}} \\ d(k-i) & = d(n-i) \\ i & = 0, 1, \cdots, N-1 \end{cases} \tag{A1}$$

- *Get the error signal based on the initial control:*

$$e^{\dagger}(k) = d(k) - \mathbf{\Phi}^{\mathrm{T}}(n)\mathbf{x}'(k) \tag{A2}$$

- *Obtain the control filter:*

$$\mathbf{w}(n) = \mathbf{\Phi}(n) + \mu e^{\dagger}(k)\mathbf{x}'(k) \tag{A3}$$

- *Get the error signal based on the new control filter:*

$$e(k-i) = d(k-i) - \mathbf{w}^{\mathrm{T}}(n)\mathbf{x}'(k-i) \tag{A4}$$

- *Update the initial value :*

$$\mathbf{\Phi}(n+1) = \mathbf{\Phi}(n) + \varepsilon \sum_{i=0}^{N-1} \lambda^i e(k-i)\mathbf{x}'(k-i) \tag{A5}$$

```
%% Modified MAML algorithm
classdef MAML_Nstep_forget
    properties
        Phi % The initial control filter
    end
    methods
        function obj = MAML_Nstep_forget(len_c)
```

Parameters	Definition	Parameters	Definition
Phi	Initial control filter	len_c	Length of control filter
Fx	Filtered reference vector	Di	Disturbance vector
Grad	Gradient accumulator	Er	Error signal
mu	Stepsize of FxLMS	lamda	Forgetting factor
epslon	Learning rate of MAML		
Li	Length of control filter	e	error signal

Table 2: The definitions of the variables used in **MAML_Nstep_forget.m**.

```
            % len_c : the lenght of control filter
            obj.Phi = zeros(len_c,1);
        end
        function [obj,Er] = MAML_initial(obj,Fx,Di,mu,lamda,epslon)
            % Fx : the filtered reference vector
            % Di : the disturbance vector
            % mu : the step size
            % lamda : the forget factor
            Fx   = flipud(Fx);
            Dis  = flipud(Di);
            Grad = 0; % Temporal gradient accumulator
            Er   = 0; % Training error signal
            Li   = length(obj.Phi) ; % The length of the control filter in the FxLMS algorithm.
                         %<-4-> Get the error signal based on the initial control filter.
                         e     = Dis(1)  - obj.Phi'*Fx;
                         %<-5-> Obtain the control filter
            Wo   = obj.Phi + mu*e*Fx    ; % One-step updation for the assumed optimal control filter.
            for jj = 1:Li
                Fd   = [Fx(jj:end);zeros(jj-1,1)];
                                 %<-6-> Get the error signal based on the new control filter.
                e    = Dis(jj) - Wo'*Fd            ;
                                 % Get the gradints based on the assumed optimal control filter
                Grad = Grad    + epslon*(mu/Li)*e*Fd*(lamda^(jj-1));
                if jj == 1
                    Er =  e ;
                end
            end
                         %%<-7-> Upate the initial value
            obj.Phi = obj.Phi + Grad ;
        end
    end

end
```

4.2 Single-channel FxLMS algorithm

The program of **FxLMS.m** implements the conventional FxLMS algorithm. This algorithm can use the initial control filter to realize adaptive active noise control. The definitions of the variables in the program are illustrated in Table 3.

Parameters	Definition	Parameters	Definition
Len_Filter	Length of control filter	Wc_initial	Initial control filter
Dis	Disturbance vector	Rf	Filtered reference vector
muW	Stepsize of FxLMS	Er	Error signal

Table 3: The definitions of the variables used in **FxLMS.m**.

```
%% single-channel FxLMS algorithm
function Er = FxLMS(Len_Filter, Wc_initial, Dis, Rf, muw)
% Len_Filter : the length of the control filter
% Wc_initial : the initial control filter
% Dis        : the disturbance
% Rf         : the filtered reference vector
% muw        : the step size
N    = Len_Filter ;
Wc   = Wc_initial ;
XD   = zeros(N,1) ;
Er   = zeros(length(Rf),1);
    for tt = 1:length(Rf)
        XD    = [Rf(tt);XD(1:end-1)];
        Rf_i = XD'           ;
        Rf_i = Rf_i'         ;
        y_t  = Wc'*Rf_i      ;
        e    = Dis(tt)-y_t ;
        Er(tt) = e           ;
        Wc     = Wc + muw*e*Rf_i;
    end
end
```

4.3 Aircraft noise cancellation on the measured path

The following code snippet offers a concise overview of **Main_tst_function.m**, which serves as the main function for evaluating the effectiveness of the proposed MAML method. This code implements the experiment in Section 3.2.

For this numerical simulation, three distinct broadband sounds are utilized to train the MAML algorithm and obtain a single initial control filter. Subsequently, the aforementioned initial control is employed within an FxLMS algorithm to effectively eliminate actual aircraft noise. In comparison to zero initialization, the MAML technique can significantly enhance the convergence speed of the conventional FxLMS approach. The entire program comprises the following primary components:

- Cleaning the memory and workspace
- Configure the system simulation condition
- Build the broadband noise for training set
- Radomly sampling the noise tracks to build dataset for the MAML algorithm

- Using Modified MAML algorithm to get the best initial control filter

4.3.1 Cleaning the memory and workspace

This segment of code is utilized to clean the memory and workspace of the MATLAB software.

```
close all ;
clear     ;
clc       ;
```

4.3.2 Configure the system simulation condition

This code snippet provides the system configuration for the numerical simulation. In this program, f_s and T denote the system sampling rate and the simulation duration, respectively. Len_N represents the length of the control filter. The definitions of the variables in the program are illustrated in Table 4.

Parameters	Definition	Parameters	Definition
fs	Sampling rate	T	Simulation duration
Len_N	Length of the control filter		

Table 4: The definitions of the variables used in **Part 1**.

```
%% Configure the system simulation condition
fs  =   16000    ; % The system sampling rate.
T   =   3        ; % The duration of the simulation.
t   =   0:1/fs:T ;
N   =   length(t); % The number of the data.

Len_N = 512      ; % Seting the length of the control filter.
%<<===Progress bar===>>
f = waitbar(0,'Please␣wait...');
pause(.5)
```

4.3.3 Build the broadband noise for training set

This section of the program produces filtered references, disturbances, and primary noises. These are then used in the modified MAML algorithm to obtain the initial control filter. The main sounds consist of three distinct broadband noises, as depicted in Figure 6. The definitions of the variables in the program are illustrated in Table 5.

Parameters	Definition	Parameters	Definition
Pri_path	Primary path	Track_num	Number of the noise tracks
Pri_n	n-th primary noise	Dis_n	n-th disturbance
Rf_n	n-th filtered reference vector		

Table 5: The definitions of the variables used in **Part 2**.

```
%% Build the broad band noise for training set
%<<===Progress bar===>>
waitbar(0.25,f,'Build the broad band noise for training set');
pause(1)
%<<===Progress bar===>>
% Loading path
load('path\P1.mat')    ;
load('path\S11.mat')   ;
Pri_path = conv(P1,S11);

Track_num = 3          ; % Seting the number of the track for the trainning
    noise.
if exist('Primary_noise.mat', 'file') == 2
    disp('Primary_noise exists in the current path.\n');
    % Loading the primary noise
    load('Primary_noise.mat');
    load('Disturbance.mat')  ;
    load('Reference.mat')    ;
else
Noise     = randn(N,1);
% filter
filter_1 = fir1(512,[0.05 0.25]);
filter_2 = fir1(512,[0.20 0.55]) ;
filter_3 = fir1(512,[0.5,0.75]) ;
% Primary noise
Pri_1 = filter(filter_1,1,Noise) ;
Pri_2 = filter(filter_2,1,Noise) ;
Pri_3 = filter(filter_3,1,Noise) ;
% Drawing fiture
data = [Pri_1,Pri_2,Pri_3];
figure ;
len_fft = length(Pri_1)   ;
len_hal = round(len_fft/2);
title('Frequency spectrum of primary noises')
for ii = 1:3
    freq = 20*log(abs(fft(data(:,ii))));
    subplot(3,1,ii);
    plot(0:(fs/len_fft):(len_hal-1)*(fs/len_fft), freq(1:len_hal));
    grid on   ;
    title("The "+num2str(ii)+"th primary noise")
    xlabel('Frequency (Hz)')
end
% Save primary noise into workspace
save('Primary_noise.mat','Pri_1','Pri_2','Pri_3');
% Generating Distrubance
Dis_1 = filter(Pri_path,1,Pri_1);
Dis_2 = filter(Pri_path,1,Pri_2);
Dis_3 = filter(Pri_path,1,Pri_3);
% Save distrubancec into workspace
save('Disturbance.mat','Dis_1','Dis_2','Dis_3');
% Genrating Filtered reference signal
Rf_1 = filter(S11,1,Pri_1);
Rf_2 = filter(S11,1,Pri_2);
```

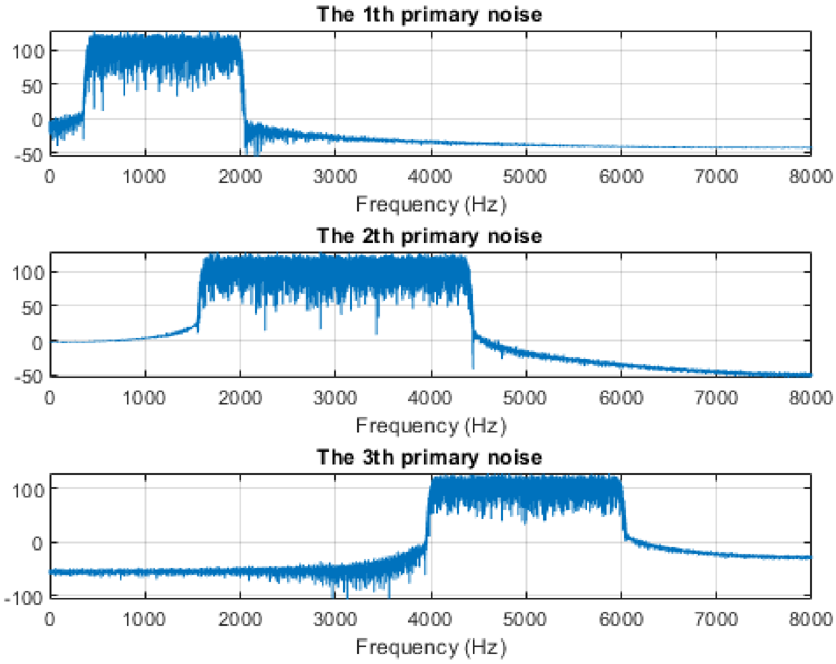

Figure 6: The frequency spectrum of the primary noises for the MAML algorithm.

```
Rf_3 = filter(S11,1,Pri_3);
% Save filter reference signal into workspace
save('Reference.mat','Rf_1','Rf_2','Rf_3');
end
```

4.3.4 Randomly sampling the noise tracks to build dataset for the MAML algorithm

This part of the program randomly samples the previously generated three types of noise tracks and their corresponding reference signals and disturbances to form the dataset. The length of the sampled vector equals that of the control filter, and these samples will be used in the proposed MAML algorithm to get the best initial control filter. The definitions of the variables in the program are illustrated in Table 6.

Parameter	**Definition**	**Parameter**	**Definition**
N_epcho	Number of training epoch	Di_data	Disturbance matrix
Fx_data	Reference signal matrix	Trac	Randomly sampling matrix

Table 6: The definitions of the variables used in **Part 3**.

```
%% Radomly sampling the noise tracks to build dataset for the MAML
    algorithm
%<<===Progress bar===>>
```

```
waitbar(0.5,f,'Radomly sampling the noise tracks to build dataset for the MAML algorithm');
pause(1)
%<<===Progress bar===>>
if exist('Sampe_data_N_set.mat', 'file') == 2
    disp('Sampe_data_N_set in the current path.\n');
    load('Sampe_data_N_set.mat');
else
N_epcho   = 4096 * 80                    ; % Setting the number of the epcho
Trac      = randi(Track_num,[N_epcho,1]); % Randomly choosing the different tracks.
Len_data = length(Dis_1)                  ;
% Seting the N steps
len    = 2*Len_N -1 ;
Fx_data = zeros(Len_N,N_epcho);
Di_data = zeros(Len_N,N_epcho);
Ref_data = [Rf_1,Rf_2,Rf_3]   ;
Dis_data = [Dis_1,Dis_2,Dis_3];
for jj = 1:N_epcho
    End = randi([len,Len_data]);
    Di_data(:,jj) = Dis_data(End-511:End,Trac(jj));
    Fx_data(:,jj) = Ref_data(End-511:End,Trac(jj));
end
save('Sampe_data_N_set.mat','Di_data','Fx_data');
end
```

4.3.5 Using Modified MAML algorithm to get the best initial control filter

This segment of the program applies the proposed modified MAML algorithm to compute the best initial control filter for the conventional FxLMS algorithm. The step size of the FxLMS algorithm, the learning rate of the MAML algorithm, and the forgetting factor are set to 0.003, 0.5, and 0.99, respectively. Figure 7 illustrates the proposed MAML algorithm's learning curve and demonstrates the proposed algorithm's convergence. The definitions of the variables in the program are illustrated in Table 7.

Parameter	Definition	Parameter	Definition
N	Number of samples in Dataset	Er	Residual error vector
mu	Step size	lamda	Forgetting factor
epslon	Learning rate of MAML	Wc	Control filter
Phi	Initial control filter		

Table 7: The definitions of the variables used in **Part 4**.

```
%<<===Progress bar===>>
waitbar(0.75,f,'Using Modified MAML algorithm to get the best initial control filter');
pause(1)
%<<===Progress bar===>>
if exist('Weigth_initiate_Nstep_forget.mat', 'file') == 2
    disp('Weigth_initiate_Nstep_forget in the current path.\n');
    load('Weigth_initiate_Nstep_forget.mat');
else
```

Parameter	Definition	Parameter	Definition
Pri_1	Primary noise	Dis_1	Disturbance
Rf_1	Reference signal	Wc_initial	Initial control filter
muw	Step size	Er	Error signal of FxLMS with zero-initialization
Wc	MAML initial value	Er_1	Error signal of FxLMS with MAML-initialization

Table 8: The definitions of the variables used in **Part 5**.

```
% Create a MAML algorithm
a  = MAML_Nstep_forget(Len_N);
N  = size(Di_data,2)          ; % The number of the sample in training set.
Er = zeros(N,1)               ; % Residual error vector
% Seting the step size for the embeded FxLMS algorithm
mu     = 0.0003             ;
% Seting the forget factor
lamda = 0.99                ;
% Seting the learning for MAML
epslon = 0.5 ;
% Runing the MAML algorithm
for jj = 1:N
    [a, Er(jj)] = a.MAML_initial(Fx_data(:,jj),Di_data(:,jj),mu,lamda,
        epslon);
end
% Drawing the residual error of the Modified MAML algorihtm
figure    ;
plot(Er)  ;
grid on   ;
title('Leanring curve of the modified MAML algorithm');
xlabel('Epoch');
ylabel('Residual error');
% Getting the optimal intial control filter
Wc = a.Phi ;
% Saving the best initial control filter into workspace
save('Weigth_initiate_Nstep_forget.mat','Wc');
end
```

4.3.6 Testing aircraft noise cancellation by using MAML initial control filter

The program employs the FxLMS algorithms, utilizing zero initialization and MAML initialization, to reduce the impact of aircraft noise. Figure 8 illustrates the error signal produced by the FxLMS algorithm during both the on and off states. It has been observed that the utilization of MAML initialization significantly enhances the convergence rate of the adaptive algorithm in comparison to the conventional zero initialization approach. The definitions of the variables in the program are illustrated in Table 8.

```
%% Testing aircraft noise cancellation by using MAML initial control filter
%<<===Progress bar===>>
waitbar(0.95,f,'Testing aircraft noise cancellation by using MAML initial 
    control filter');
pause(1)
%<<===Progress bar===>>
```

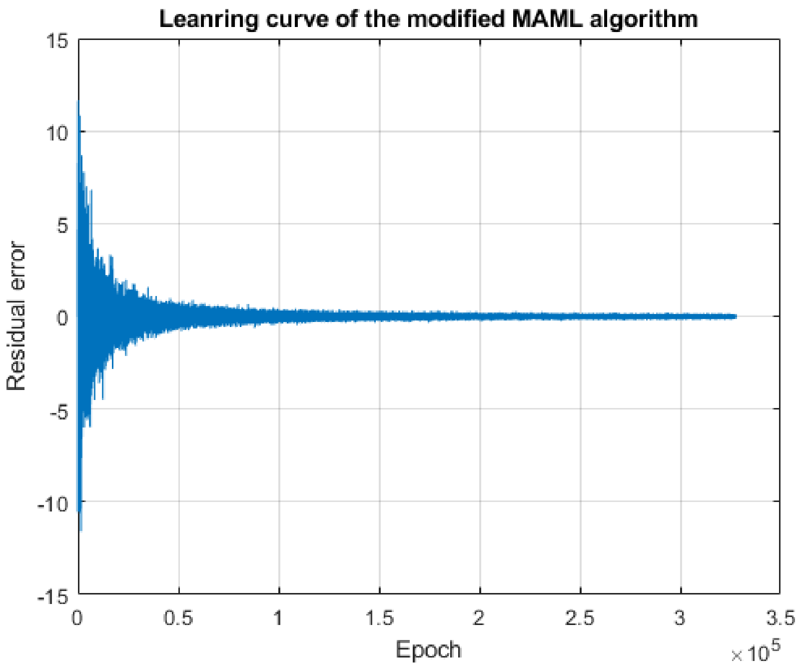

Figure 7: The learning curve of the modified MAML algorithm used to compute the initial control filter for the FxLMS algorithm.

```
% Loading aricrat noise data
aircrat = load('707_Sound_for_Simulation.mat');
% Building primary noise
Pri_1    = aircrat.PilingNoise(1:153945)       ;
% Generating the disturbacne
Dis_1    = filter(Pri_path,1,Pri_1)            ;
% Generating the filter reference
Rf_1     = filter(S11,1,Pri_1)                 ;
% Runging the FxLMS with the zero-initialization control filter
Wc_initial = zeros(Len_N,1);
muw        = 0.00001 ; % Step size of All FxLMS algorithms.
Er         = FxLMS(Len_N, Wc_initial, Dis_1, Rf_1, muw);
% Runging the FxLMS with the MAML-initialization control filter
Wc_initial = Wc;
Er1        = FxLMS(Len_N, Wc_initial, Dis_1, Rf_1, muw);
figure
% Drawing the figures of the MAML and FxLMS
plot((0:length(Dis_1)-1)*(1/fs),Dis_1,(0:length(Er)-1)*(1/fs),Er,(0:length(Er1)-1)*(1/fs),Er1);
title('Aircraft noise cancellation')
xlabel('Time (seconds)')
ylabel('Error signal')
legend({'ANC off','FxLMS with zero-initialization','FxLMS with MAML-initialization'})
grid on;
%<<===Progress bar===>>
```

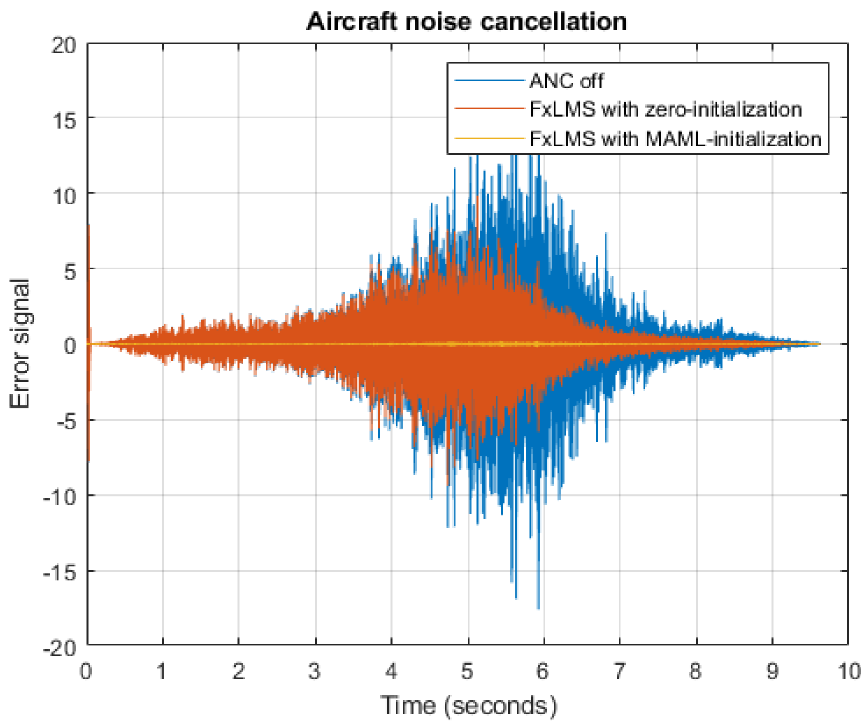

Figure 8: Aircraft noise reduction performance of the FxLMS algorithms with the zero and MAML initializations.

```
waitbar(1,f,'Finishing');
pause(1)
%<<===Progress bar===>>
%------------------------------end---------------------------------------
```

5 Conclusion

The document provides a detailed introduction to implementing a modified Model-Agnostic Meta-learning (MAML) algorithm in Matlab. This modified MAML algorithm takes the forgetting factor to alleviate the initial-zero-input issue of the adaptive filters. This work uses the proposed method to seek the best initial control filter for the adaptive active noise control system based on its noise dataset. With barely additional computations, the modified MAML algorithm can assist the filtered reference square (FxLMS) algorithm in achieving fast convergence. The numerical simulation demonstrates that the FxLMS algorithm with the MAML initialization converges much faster than that with normal zero initialization when dealing with raw aircraft noise.

References

[1] B. Lam, W.-S. Gan, D. Shi, M. Nishimura, and S. Elliott, “Ten questions concerning active noise control in the built environment,” **Building and Environment**, vol. 200, p. 107 928, 2021.

[2] D. Shi, B. Lam, W.-S. Gan, J. Cheer, and S. J. Elliott, “Active noise control in the new century: The role and prospect of signal processing,” in **INTER-NOISE and NOISE-CON Congress and Conference Proceedings**, Institute of Noise Control Engineering, vol. 268, 2023, pp. 5141–5151.

[3] X. Shen, D. Shi, S. Peksi, and W.-S. Gan, “Implementations of wireless active noise control in the headrest,” in **INTER-NOISE and NOISE-CON Congress and Conference Proceedings**, Institute of Noise Control Engineering, vol. 265, 2023, pp. 3445–3455.

[4] X. Shen, D. Shi, S. Peksi, and W.-S. Gan, “A multi-channel wireless active noise control headphone with coherence-based weight determination algorithm,” **Journal of Signal Processing Systems**, vol. 94, no. 8, pp. 811–819, 2022.

[5] X. Shen, D. Shi, and W.-S. Gan, “A hybrid approach to combine wireless and earcup microphones for anc headphones with error separation module,” in **ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**, IEEE, 2022, pp. 8702–8706.

[6] X. Shen, D. Shi, W.-S. Gan, and S. Peksi, “Adaptive-gain algorithm on the fixed filters applied for active noise control headphone,” **Mechanical Systems and Signal Processing**, vol. 169, p. 108 641, 2022.

[7] X. Shen, W.-S. Gan, and D. Shi, “Alternative switching hybrid anc,” **Applied Acoustics**, vol. 173, p. 107 712, 2021.

[8] X. Shen, D. Shi, and W.-S. Gan, “A wireless reference active noise control headphone using coherence based selection technique,” in **ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**, IEEE, 2021, pp. 7983–7987.

[9] B. Lam, C. Shi, D. Shi, and W.-S. Gan, “Active control of sound through full-sized open windows,” **Building and Environment**, vol. 141, pp. 16–27, 2018.

[10] B. Lam, D. Shi, W.-S. Gan, S. J. Elliott, and M. Nishimura, “Active control of broadband sound through the open aperture of a full-sized domestic window,” **Scientific reports**, vol. 10, no. 1, pp. 1–7, 2020.

[11] B. Lam, D. Shi, V. Belyi, S. Wen, W.-S. Gan, K. Li, and I. Lee, “Active control of low-frequency noise through a single top-hung window in a full-sized room,” **Applied Sciences**, vol. 10, no. 19, p. 6817, 2020.

[12] B. Lam, K. C. Q. Lim, K. Ooi, Z.-T. Ong, D. Shi, and W.-S. Gan, “Anti-noise window: Subjective perception of active noise reduction and effect of informational masking,” **Sustainable Cities and Society**, vol. 97, p. 104 763, 2023.

[13] C. Shi, N. Jiang, H. Li, D. Shi, and W.-S. Gan, “On algorithms and implementations of a 4-channel active noise canceling window,” in **2017 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)**, IEEE, 2017, pp. 217–221.

[14] R. Hasegawa, D. Shi, Y. Kajikawa, and W.-S. Gan, “Window active noise control system with virtual sensing technique,” in **INTER-NOISE and NOISE-CON Congress and Conference Proceedings**, Institute of Noise Control Engineering, vol. 258, 2018, pp. 6004–6012.

[15] C. Shi, H. Li, D. Shi, B. Lam, and W.-S. Gan, “Understanding multiple-input multiple-output active noise control from a perspective of sampling and reconstruction,” in **2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)**, IEEE, 2017, pp. 124–129.

[16] C. K. Lai, J. S. Tey, D. Shi, and W.-S. Gan, “Robust estimation of open aperture active control systems using virtual sensing,” in **INTER-NOISE and NOISE-CON Congress and Conference Proceedings**, Institute of Noise Control Engineering, vol. 265, 2023, pp. 3397–3407.

[17] C. Shi, T. Murao, D. Shi, B. Lam, and W.-S. Gan, “Open loop active control of noise through open windows,” in **Proceedings of Meetings on Acoustics**, AIP Publishing, vol. 29, 2016.

[18] D. Shi, C. Shi, and W.-S. Gan, “A systolic fxlms structure for implementation of feedforward active noise control on fpga,” in **2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)**, IEEE, 2016, pp. 1–6.

[19] D. Shi, W.-S. Gan, B. Lam, and C. Shi, “Two-gradient direction fxlms: An adaptive active noise control algorithm with output constraint,” **Mechanical Systems and Signal Processing**, vol. 116, pp. 651–667, 2019.

[20] S. Wen, W.-S. Gan, and D. Shi, “Convergence behavior analysis of fxlms algorithm with different leaky term,” in **INTER-NOISE and NOISE-CON Congress and Conference Proceedings**, Institute of Noise Control Engineering, vol. 261, 2020, pp. 728–739.

[21] Z. Luo, D. Shi, J. Ji, and W.-s. Gan, “Implementation of multi-channel active noise control based on back-propagation mechanism,” **arXiv preprint arXiv:2208.08086**, 2022.

[22] D. Shi, J. He, C. Shi, T. Murao, and W.-S. Gan, “Multiple parallel branch with folding architecture for multichannel filtered-x least mean square algorithm,” in **2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**, IEEE, 2017, pp. 1188–1192.

[23] D. Shi, B. Lam, W.-S. Gan, and S. Wen, “Block coordinate descent based algorithm for computational complexity reduction in multichannel active noise control system,” **Mechanical Systems and Signal Processing**, vol. 151, no. 0888-3270, p. 107 346, 2021.

[24] D. Shi, B. Lam, and W.-s. Gan, “Analysis of multichannel virtual sensing active noise control to overcome spatial correlation and causality constraints,” in **ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**, IEEE, 2019, pp. 8499–8503.

[25] D. Shi, W.-S. Gan, B. Lam, S. Wen, and X. Shen, “Active noise control based on the momentum multichannel normalized filtered-x least mean square algorithm,” in **In Inter-Noise 2020. The International Institute of Noise Control Engineering.**, 2020.

[26] D. Shi, B. Lam, S. Wen, and W.-S. Gan, “Multichannel active noise control with spatial derivative constraints to enlarge the quiet zone,” in **ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**, IEEE, 2020, pp. 8419–8423.

[27] D. Shi, B. Lam, X. Shen, and W.-S. Gan, “Multichannel two-gradient direction filtered reference least mean square algorithm for output-constrained multichannel active noise control,” **Signal Processing**, vol. 207, p. 108 938, 2023.

[28] D. Shi, B. Lam, J. Ji, X. Shen, C. K. Lai, and W.-S. Gan, "Computation-efficient solution for fully-connected active noise control window: Analysis and implementation of multichannel adjoint least mean square algorithm," **Mechanical Systems and Signal Processing**, vol. 199, p. 110 444, 2023.

[29] J. Ji, D. Shi, W.-S. Gan, X. Shen, and Z. Luo, "A computation-efficient online secondary path modeling technique for modified fxlms algorithm," **arXiv preprint arXiv:2306.11408**, 2023.

[30] D. Y. Shi, B. Lam, and W.-S. Gan, "A novel selective active noise control algorithm to overcome practical implementation issue," in **2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**, IEEE, 2018, pp. 1130–1134.

[31] C. K. Lai, D. Shi, B. Lam, and W.-S. Gan, "Mov-modified-fxlms algorithm with variable penalty factor in a practical power output constrained active control system," **IEEE Signal Processing Letters**, 2023.

[32] C. K. Lai, B. Lam, D. Shi, and W.-S. Gan, "Real-time modelling of observation filter in the remote microphone technique for an active noise control application," in **ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**, IEEE, 2023, pp. 1–5.

[33] D. Shi, W.-S. Gan, B. Lam, and X. Shen, "Comb-partitioned frequency-domain constraint adaptive algorithm for active noise control," **Signal Processing**, vol. 188, p. 108 222, 2021.

[34] D. Shi, W.-S. Gan, B. Lam, and X. Shen, "A frequency-domain output-constrained active noise control algorithm based on an intuitive circulant convolutional penalty factor," **IEEE/ACM Transactions on Audio, Speech, and Language Processing**, vol. 31, pp. 1318–1332, 2023.

[35] Z. Luo, D. Shi, W.-S. Gan, Q. Huang, and L. Zhang, "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**, Institute of Noise Control Engineering, vol. 265, 2023, pp. 1615–1622.

[36] D. Shi, B. Lam, W.-S. Gan, and S. Wen, "Optimal leak factor selection for the output-constrained leaky filtered-input least mean square algorithm," **IEEE Signal Processing Letters**, vol. 26, no. 5, pp. 670–674, 2019.

[37] D. Shi, W.-S. Gan, B. Lam, S. Wen, and X. Shen, "Optimal output-constrained active noise control based on inverse adaptive modeling leak factor estimate," **IEEE/ACM Transactions on Audio, Speech, and Language Processing**, vol. 29, pp. 1256–1269, 2021.

[38] D. ZENG, W. XIA, D. SHI, and Z. HE, "Three-term parallelizable real-valued lms identification method for digital predistorter," **JOURNAL OF INFORMATION &COMPUTATIONAL SCIENCE**, vol. 9, no. 18, pp. 5741–5748, 2012.

[39] D. Shi, Z. Liu, R. Qian, W. Xia, and Z. He, "Fpga implementation of dlms algorithm for digital repeater," **Dianshi Jishu(Video Engineering)**, vol. 36, no. 9, 2012.

[40] Z. Luo, D. Shi, W.-S. Gan, and Q. Huang, "Delayless generative fixed-filter active noise control based on deep learning and bayesian filter," **IEEE/ACM Transactions on Audio, Speech, and Language Processing**, 2023.

[41] Z. Luo, D. Shi, X. Shen, J. Ji, and W.-S. Gan, "Gfanc-kalman: Generative fixed-filter active noise control with cnn-kalman filtering," **IEEE Signal Processing Letters**, 2023.

[42] Z. Luo, D. Shi, X. Shen, J. Ji, and W.-S. Gan, "Deep generative fixed-filter active noise control," in **ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**, IEEE, 2023, pp. 1–5.

[43] J. Ji, D. Shi, Z. Luo, X. Shen, and W.-S. Gan, "A practical distributed active noise control algorithm overcoming communication restrictions," in **ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**, IEEE, 2023, pp. 1–5.

[44] Z. Luo, D. Shi, and W.-S. Gan, "A hybrid sfanc-fxnlms algorithm for active noise control based on deep learning," **IEEE Signal Processing Letters**, vol. 29, pp. 1102–1106, 2022.

[45] D. Shi, D. He, and M. Cai, **Adaptive filtering method and system based on error sub-band**, US Patent 9,419,826, Aug. 2016.

[46] D. Shi, B. Lam, K. Ooi, X. Shen, and W.-S. Gan, "Selective fixed-filter active noise control based on convolutional neural network," **Signal Processing**, vol. 190, p. 108 317, 2022.

[47] D. Shi, W.-S. Gan, B. Lam, Z. Luo, and X. Shen, "Transferable latent of cnn-based selective fixed-filter active noise control," **IEEE/ACM Transactions on Audio, Speech, and Language Processing**, 2023.

[48] X. Shen, D. Shi, Z. Luo, J. Ji, and W.-S. Gan, "A momentum two-gradient direction algorithm with variable step size applied to solve practical output constraint issue for active noise control," in **ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**, IEEE, 2023, pp. 1–5.

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

[50] Z. Luo, D. Shi, X. Shen, and W.-S. Gan, "Unsupervised learning based end-to-end delayless generative fixed-filter active noise control," **arXiv preprint arXiv:2402.09460**, 2024.

[51] L. Li, W. Xia, D. Shi, and J. Li, "Frequency estimation on power system using recursive-least-squares approach," in **Proceedings of the 2012 International Conference on Information Technology and Software Engineering: Information Technology & Computing Intelligence**, Springer Berlin Heidelberg, 2013, pp. 11–18.

[52] G. Yu, "Implementation of kalman filter approach for active noise control by using matlab: Dynamic noise cancellation," **arXiv preprint arXiv:2402.06896**, 2024.

[53] D. Y. Shi and W.-S. Gan, "Comparison of different development kits and its suitability in signal processing education," in **2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**, IEEE, 2016, pp. 6280–6284.

[54] D. Shi, W.-S. Gan, J. He, and B. Lam, "Practical implementation of multichannel filtered-x least mean square algorithm based on the multiple-parallel-branch with folding architecture for large-scale active noise control," **IEEE Transactions on Very Large Scale Integration (VLSI) Systems**, vol. 28, no. 4, pp. 940–953, 2019.

[55] J. Bai, H. Yin, M. Wang, D. Shi, W.-S. Gan, and J. Chen, "Audiolog: Llms-powered long audio logging with acoustic scenes and events joint estimation," **arXiv e-prints**, arXiv–2311, 2023.

[56] H. Yin, J. Bai, M. Wang, D. Shi, W.-S. Gan, and J. Chen, "Interactive dual attention conformer with scene-based mask for soft sound event detection," **arXiv preprint arXiv:2311.14068**, 2023.

[57] H. Yin, M. Wang, J. Bai, D. Shi, W.-S. Gan, and J. Chen, "Sub-band and full-band interactive u-net with dprnn for demixing cross-talk stereo music," **arXiv preprint arXiv:2401.08678**, 2024.

[58] J. Bai, M. Wang, H. Liu, H. Yin, Y. Jia, S. Huang, Y. Du, D. Zhang, M. D. Plumbley, D. Shi, **et al.**, "Description on ieee icme 2024 grand challenge: Semi-supervised acoustic scene classification under domain shift," **arXiv preprint arXiv:2402.02694**, 2024.

[59] D. Shi, W.-S. Gan, B. Lam, and K. Ooi, "Fast adaptive active noise control based on modified model-agnostic meta-learning algorithm," **IEEE Signal Processing Letters**, vol. 28, pp. 593–597, 2021.

[60] D. Shi, W.-s. Gan, X. Shen, Z. Luo, and J. Ji, "What is behind the meta-learning initialization of adaptive filter?—a naive method for accelerating convergence of adaptive multichannel active noise control," **Neural Networks**, p. 106 145, 2024.