Implementation of the Multichannel Filtered Reference Least Mean Square (McFxLMS) Algorithm with an Arbitrary Number of Channels by Using MATLAB

Wang Boxiang[1]

Multichannel filtered reference least mean square (McFxLMS) algorithms are widely utilized in adaptive multichannel active noise control (MCANC) applications. As a critical and high-computationally efficient adaptive critical algorithm, it also typically works as a benchmark for comparative studies of the new algorithms proposed by peers and researchers. However, up to now, there are few open-source codes for the FxLMS algorithm, especially for large-count channels. Therefore, this work provides a MATLAB code for the McFxLMS algorithm, which can be used for the arbitrary number of channels system. The code is available on GitHub and MathWork.

1 Introduction

Active Noise Control (ANC) is an advanced technique used to reduce unwanted sound by introducing controlled anti-noise that effectively cancels the original noise. This process, known as destructive interference, is a sophisticated application of the principle of superposition in wave physics. Due to its efficiency in reducing low-frequency noise and compact size, ANC is widely utilized in many applications sensitive to noise interferences [1], [2], such as headphones [3]–[7], automobiles, and windows [8]–[15]. Compared to the single-channel structure [16]–[18], the Multichannel Active Noise Control (MCANC) system owns more freedom of controllable degree and, hence, is more effective in dealing with complicated acoustic environments [19]. Furthermore, to assist MCANC in coping with the dynamic noise and varying acoustic environments, adaptive algorithms, such as the multichannel filtered reference least mean square (McFxLMS) algorithm, have been developed to use in noise cancellation. Despite the development of numerous sophisticated algorithms [20]–[29] and methods [30]–[35] to improve the efficiency of ANC systems, McFxLMS plays a crucial role in active control and serves as the benchmark for these new algorithms. In addition, its derivative algorithms [36]–[44] are flourishing to address the practical obstacles of actual ANC applications. Therefore, this study employs the widely-used simulation tool, MATLAB, to realize the McFxLMS algorithm and make its code accessible to the public. This code can be readily employed to implement the co-located and fully connected structures.

The document provides a detailed explanation of a MATLAB program that simulates feedforward multichannel active noise control (ANC) using the filtered-x least mean square (FxLMS) algorithm. Specifically, the user is free to choose the number of reference microphones, the number of secondary sources, and the number of error microphones. In addition, the program offers the option of using the GPU to accelerate the computation.

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

2 Brief Theoretical Introduction on Multichannel Filtered Reference Least Mean Square Algorithm

Active noise control (ANC) is a mechanism used to address low-frequency noise issues based on the principle of acoustic wave superposition [45]. The ANC system artificially generates an anti-noise wave that has the same amplitude but the reverse phase of the noise wave, which interferes with the disturbance destructively.

In general, the ANC system can be classified as feedforward structure and feedback structure. The feedforward structure implements the reference microphone and the error microphone to generate anti-noise that can dynamically match with the variation of the primary noise, which allows it to deal with many noise types. Moreover, the ANC system also can be referred to as single-channel ANC [16] or multichannel ANC based on the number of secondary sources used. Compared to single-channel ANC, multichannel ANC is implemented to gain a larger quiet zone through multiple secondary sources and error microphones.

The FxLMS is among the most practical adaptive algorithms proposed to compensate for the influence of the secondary path in an ANC system. Figure 1 shows the block diagram of the multichannel FxLMS algorithm, which has J reference microphones, K secondary sources, and M error microphones. The control filter matrix is given by

$$\mathbf{w}(n) = \begin{bmatrix} \mathbf{w}_{11}^T(n) & \mathbf{w}_{12}^T(n) & \cdots & \mathbf{w}_{1J}^T(n) \\ \mathbf{w}_{21}^T(n) & \mathbf{w}_{22}^T(n) & \cdots & \mathbf{w}_{2J}^T(n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{w}_{K1}^T(n) & \mathbf{w}_{K2}^T(n) & \cdots & \mathbf{w}_{KJ}^T(n) \end{bmatrix} \in \mathbb{R}^{K\times JN}, \tag{1}$$

where $\mathbf{w}_{kj}(n) = [w_{kj,0}(n), w_{kj,1}(n), \ldots, w_{kj,N-1}(n)]^T \in \mathbb{R}^{JN\times 1}$ is the control filter from the jth input to the kth output, and N denotes the length of the control filter. The reference vector is given by

$$\mathbf{x}(n) = \left[\mathbf{x}_1^T(n), \mathbf{x}_2^T(n), \ldots, \mathbf{x}_J^T(n)\right]^T \in \mathbb{R}^{JN\times 1}, \tag{2}$$

where $\mathbf{x}_j = [x_j(n), x_j(n-1), \ldots, x_j(n-N+1)]^T \in \mathbb{R}^{N\times 1}$. The output of the control filter is given by

$$\mathbf{y}(n) = \mathbf{w}(n)\mathbf{x}(n) \in \mathbb{R}^{K\times 1}. \tag{3}$$

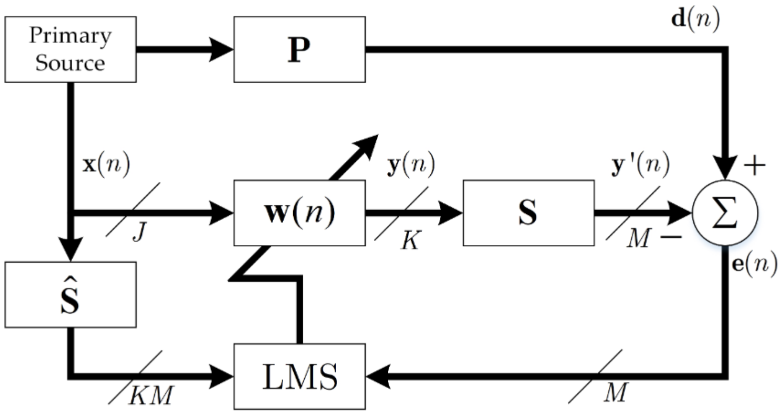

Figure 1: Block diagram of the multichannel FxLMS algorithm.

The anti-noise vector at the error microphones, as illustrated in Figure 1, is given by

$$\mathbf{y}'(n) = \mathbf{s} * \mathbf{y}(n) \in \mathbb{R}^{M\times 1}, \tag{4}$$

where $\mathbf{s}$ represents the impulse response matrix of secondary paths

$$\mathbf{s} = \begin{bmatrix} \mathbf{s}_{11}(n) & \mathbf{s}_{12}(n) & \cdots & \mathbf{s}_{1K}(n) \\ \mathbf{s}_{21}(n) & \mathbf{s}_{22}(n) & \cdots & \mathbf{s}_{2K}(n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{s}_{M1}(n) & \mathbf{s}_{M2}(n) & \cdots & \mathbf{s}_{MK}(n) \end{bmatrix} \in \mathbb{R}^{M\times K}. \tag{5}$$

The error signal vector $\mathbf{e}(n)$ measured by M error microphones can be expressed as

$$\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{y}'(n) \in \mathbb{R}^{M\times 1}, \tag{6}$$

where $\mathbf{d}(n) = [d_1(n), d_2(n), \ldots, d_M(n)]^T \in \mathbb{R}^{M\times 1}$ stands for the disturbance vector, with $d_m(n)$ denoting the disturbance at the mth $(m = 1, 2, , M)$ error microphone. The cost function of the adaptive filter is defined as

$$J(n) = \sum_{m=1}^{M} e_m^2(n). \tag{7}$$

We replace the impulse response $s_{mk}(n)$ with its estimate $\hat{\mathbf{s}}_{mk}(n)$ to give

$$x'_{jkm}(n) = \hat{s}_{mk}(n) * x_j(n). \tag{8}$$

The control filter weight vector $\mathbf{w}_{kj}(n)$ is updated in the direction of the negative gradient with step size μ based on the minimization of the estimated cost function $J(n)$:

$$\mathbf{w}_{kj}(n+1) = \mathbf{w}_{kj}(n) + \mu \sum_{m=1}^{M} e_m(n)\mathbf{x}'_{jkm}(n), \tag{9}$$

where the filtered reference vector is

$$\mathbf{x}'_{jkm}(n) = \left[x'_{jkm}(n), x'_{jkm}(n-1), \ldots, x'_{jkm}(n-N+1)\right]^T. \tag{10}$$

3 Code Explanation

This MATLAB program implements the simulation of a feedforward multichannel FxLMS ANC system, with the number of channels that can be specified arbitrarily by the user. The program contains three MATLAB files: *Multichannel_FxLMS.m* and *Control_filter.m* constitute the program of the McFxLMS algorithm, while *Four_MCANC.m* is the main program to test the McFxLMS algorithm.

3.1 Control_filter

The control filter is an essential element of the ANC system, responsible for processing the reference signal to generate the control signal.

Therefore, *Control_filter.m* defines a class called **Control_filter**, which specifies the properties and functions of the control filter for a multichannel ANC system and is an important component that will be used in the *Multichannel_FxLMS.m* class.

3.1.1 Properties of Control_filter

The properties of **Control_filter** are shown below.

```
properties
    Sec_Matrix %--The coefficients of the secondary path (Ls x E_num).
    umW        %--The step size of the algorithm.
    Len        %--The length of the control filter.
    Ls         %--The length of the secondary path.
    Wc         %--The control filter coefficients.
    Fd         %--The buffer of the control filter (Len x E_num).
    E_num      %--The number of the error microphones.
    Xd         %--The delay line of the reference.
    Xf         %--The delay line of the filtered reference.
    Yd         %--The delay line of the control filter output.
end
```

In this code snippet, Sec_Matrix is used to store the coefficients of the secondary path and has a dimension of Ls by E_num; Ls and E_num denote the length of the secondary path impulse response and the number of the error microphones, respectively. Len represents the length of the control filter; Wc is the vector of the control filter. Xd, Xf, and Yd stands for the delay line of the input data.

3.1.2 Function 1: Control_filter

This function is used to initialize the class instance, which sets the properties of the control filter. In particular, it initializes some properties based on whether *Sec_Matrix* is a gpuArray to support GPU-accelerated computation.

```
function obj=Control_filter(Sec_Matrix, Len, umW)
    obj.Len        = Len        ;
    obj.umW        = umW        ;
    csize          = size(Sec_Matrix);
    obj.Ls         = csize(1)   ;
    obj.E_num      = csize(2)   ;
    obj.Sec_Matrix = Sec_Matrix ;
    if isa(Sec_Matrix,'gpuArray')
        obj.Wc = gpuArray(zeros(Len,1));
        obj.Xd = gpuArray(zeros(Len,1));
        obj.Xf = gpuArray(zeros(obj.Ls,1));
        obj.Yd = gpuArray(zeros(obj.Ls,1));
        obj.Fd = gpuArray(zeros(Len,obj.E_num));
    else
        obj.Wc = zeros(Len,1);
        obj.Xd = zeros(Len,1);
        obj.Xf = zeros(obj.Ls,1);
        obj.Yd = zeros(obj.Ls,1);
        obj.Fd = zeros(Len,obj.E_num);
    end
end
```

3.1.3 Function 2: Generator_antinoise

This function is the key function of *Control_filter* that generates the anti-noise signal. It receives the input signal xin and the error signal Er, then updates the filter coefficients Wc according to the FxLMS algorithm to generate the anti-noise signal y_anti. This function also updates various delay lines and the buffer of the control filter.

```
function [y_anti, obj] = Generator_antinoise(obj, xin, Er)
    umW1= obj.umW ;
    Xd1 = obj.Xd  ;
    Xf1 = obj.Xf  ;
    Wc1 = obj.Wc  ;
    Fd1 = obj.Fd  ;
    Yd1 = obj.Yd  ;
    Sec_Matrix1 = obj.Sec_Matrix;
    %---------------------------->>>----------------------<<<
    Xd1   = [xin;Xd1(1:end-1)];
    Xf1   = [xin;Xf1(1:end-1)];
    Wc1   = Wc1 - umW1 * Fd1 * Er;
    y_o   = Wc1'*Xd1 ;
    Yd1   = [y_o;Yd1(1:end-1)];
    y_anti = Sec_Matrix1' * Yd1;
    fd   = Sec_Matrix1' * Xf1 ;
    Fd1   = [fd'; Fd1(1:end-1,:)];
    %---------------------------->>>----------------------<<<
    obj.Xd = Xd1;
    obj.Xf = Xf1;
    obj.Wc = Wc1;
    obj.Fd = Fd1;
    obj.Yd = Yd1;
end
```

3.2 Multichannel_FxLMS

Multichannel_FxLMS.m defines a class called *Multichannel_FxLMS*, which specifies the properties and functions of the multichannel ANC system.

3.2.1 Properties of Multichannel_FxLMS

The properties of *Multichannel_FxLMS* are shown below.

```
properties
    controller %--The controller of the multichannel system.
    Cunit      %--The number of the control units.
    R_num      %--The number of reference microphones.
    E_num      %--The number of error microphones.
    S_num      %--The number of secondary sources.
    lenc       %--The length of the control filter.
end
```

3.2.2 Function 1: Multichannel_FxLMS

This function is used to initialize the class instance, which sets the properties of the multichannel ANC system. Specifically, this function supports two types of multichannel ANC systems: fully-connection ANC and collocated ANC based on the number of control units *Cunit*. In addition, information such as system structure (fully-connection or collocated) and hardware usage (CPU or GPU) is printed out.

```
function obj = Multichannel_FxLMS(Wc,Sec_estimate,umW)
    w_size = size(Wc);
    len    = w_size(1);
    Cunit = w_size(2);
    if length(w_size)== 2
        R_num = 1;
    else
        R_num = w_size(3);
    end
    obj.lenc  = len  ;
    obj.Cunit = Cunit;
    obj.R_num = R_num;

    s_size = size(Sec_estimate);
    E_num  = s_size(2);
    if length(s_size)==2
        S_num  = 1;
    else
        S_num  = s_size(3);
    end
    obj.E_num = E_num ;
    obj.S_num = S_num ;
    obj.controller = repmat(Control_filter(squeeze(Sec_estimate(:,:,1)),len
        , umW),Cunit,R_num);
    if Cunit ~= 1  % Full structure
        for jj = 1:R_num
            for kk = 1:Cunit
                obj.controller(kk,jj) = Control_filter(squeeze(Sec_estimate
                    (:,:,kk)),len, umW);
            end
        end
        disx3 = sprintf('Structure:␣Fully-connection␣feedforward␣ANC.');
    else            % Collocated structure
        for jj = 1:R_num
            obj.controller(1,jj) = Control_filter(squeeze(Sec_estimate(:,:,
                jj)) ,len, umW);
        end
        disx3 = sprintf('Structure:␣Collocated␣feedforward␣ANC.');
    end
    fprintf('<<--------------------------------------------------->>\n');
    fprintf('The␣multichannel␣FxLMS␣has␣been␣sucessfuly␣created.\n');
    disp(disx3);
    if isa(Wc,'gpuArray')
        fprintf('Hardware␣usage:␣GPU.\n');
    else
        fprintf('Hardware␣usage:␣CPU.\n');
    end
    fprintf('Dimension:␣%d␣x␣%d␣x␣%d␣\n', R_num, S_num, E_num);
    fprintf('<<--------------------------------------------------->>\n');
end
```

3.2.3 Function 2: FxLMS_canceller

This function is the key function of *Multichannel_FxLMS* that implements the FxLMS algorithms to achieve multichannel ANC. It receives the reference signal *Re* and the disturbance signal *Disturbance* as inputs then calls the *Generator_antinoise* function of *control_filter* moment by moment to generate the anti-noise signal. Finally, the anti-noise signal is added to the disturbance signal *Disturbance* to obtain the error signal *Er*.

```
function [Er, obj] = FxLMS_canceller(obj, Re, Disturbance)
    % Re: the reference matrix [microphone x signal length]
    % Disturabnce: the disturbance matrix [microphone x signal length]
    data_size = size(Re)     ;
    len       = data_size(2);
    if isa(Re,'gpuoArray')
        E = gpuArray(zeros(obj.E_num,len+1));
    else
        E = zeros(obj.E_num,len+1);
    end
    %% Filtering processing
    tic
    fprintf('<<--------------------START------------------------------\n');
    for nn = 1: len
        if obj.Cunit ~=1
            for jj = 1:obj.R_num
                for kk = 1:obj.Cunit
                    [Y,obj.controller(kk,jj)] = Generator_antinoise(obj.
                        controller(kk,jj),
                    Re(jj,nn),E(:,nn));
                    E(:,nn+1) = E(:,nn+1) + Y ;
                end
            end
        else
            for jj = 1:obj.R_num
                [Y,obj.controller(1,jj)] = Generator_antinoise(obj.
                    controller(1,jj),
                Re(jj,nn),E(:,nn));
                 E(:,nn+1) = E(:,nn+1) + Y ;
            end
        end
        E(:,nn+1) = Disturbance(:,nn) + E(:,nn+1);
    end
    Er = gather(E(:,2:end));
    toc
    fprintf('---------------------END------------------------------>>\n');
end
```

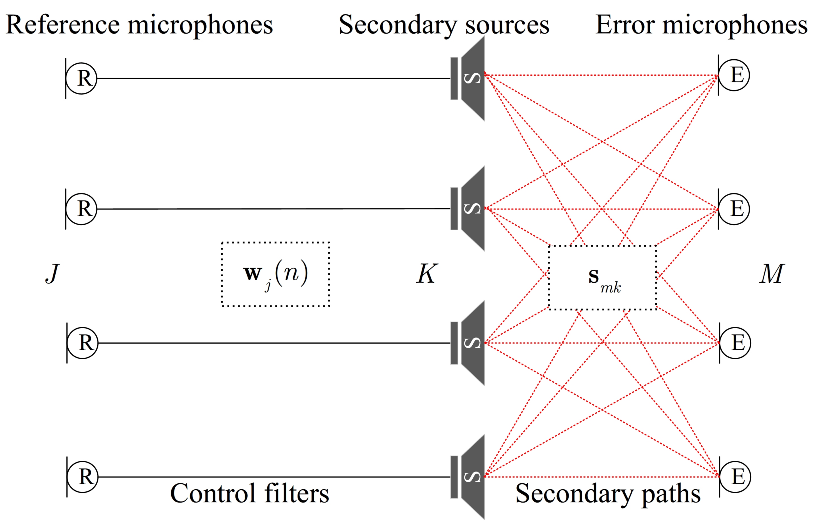

Figure 2: Block diagram of the 4-channel collocated active noise control system.

3.2.4 Function 3: Get_coefficients

This function is used to obtain the coefficients of the control filters in the multichannel ANC system.

```
function Wc = Get_coefficients(obj)
    if obj.Cunit ~=1
        Wc = zeros(obj.lenc,obj.Cunit,obj.R_num);
        for jj = 1:obj.R_num
            for kk = 1:obj.Cunit
                contorler = obj.controller(kk,jj);
                Wc(:,kk,jj)=contorler.Wc;
            end
        end
    else
        Wc = zeros(obj.lenc,obj.R_num);
        for jj = 1:obj.R_num
            contorler = obj.controller(1,jj);
            Wc(:,jj)=contorler.Wc;
       end
   end
end
```

4 Testing code: Four channel active noise cancellation

The *Four_MCANC.m* carries out a simulation on a collocated 4x4x4 ANC system, as shown in Figure 2. In the simulation, *Multichannel_FxLMS.m* and *Four_MCANC.m* are used to reduce the disturbance.

4.1 Loading primary and secondary paths

The primary path is the path the noise takes from the noise source to the error microphone and the secondary path is the path the anti-noise takes from the secondary source to the error microphone. Here, the primary paths and the secondary paths are loaded from files and the length of both the primary paths and the secondary paths are 256.

```
Pri = zeros(256,4); % Primary path
for nn=1:4
    a = sprintf('path\\P%d.mat',nn);
    b = load(a);
    c = sprintf('b.P%d',nn);
    d = eval(c)    ;
    Pri(:,nn) = d ;
end
Sec = zeros(256,4,4); % Secondary path
for ss = 1:4 % Number of secondary sources
    for mm = 1:4 % Number of error microphones
        a = sprintf('path\\S%d%d.mat',ss,mm);
        b = load(a);
        c = sprintf('b.S%d%d',ss,mm);
        d = eval(c)    ;
        Sec(:,mm,ss)=d;
    end
end
```

4.2 Generate reference and disturbance signals

The following commands generate a broadband noise of 400-800 Hz as reference signals by filtering white noise through a bandpass filter, and generate disturbance signals by filtering the reference signal through the primary path. The sampling frequency of the system is set to 16 kHz.

```
fs =  16000; % Sampling frequency
t  =  40; % Sampling time
T  = 0:1/fs:t ;
len= length(T); % Signal length
Re = randn(len,1); % Generate white noise
bf = fir1(512,[0.05 0.1]); % Bandpass filter with passband 400-800Hz
Re = filter(bf,1,Re); % Filtering the white noise
Re1 = [Re';Re';Re';Re']; % Reference [reference microphone x signal length]
Dir = zeros(4,len); % Disturbance [reference microphone x signal length]
for jj = 1:4
    Dir(jj,:) = (filter(Pri(:,jj),1,Re1(jj,:)))';
end
Red = Re1;
```

4.2.1 Noise cancellation based on the McFxLMS algorithm

The following commands simulate a collocated 4x4x4 ANC system with the length of control filters is 512 and the stepsize of the FxLMS algorithm is 0.00001.

```
Wc   = zeros(512,1,4); % Implement a 4-by-4-by-4 channel FxLMS.
muW = 0.00001; % Stepsize of the FxLMS algorithm
a = Multichannel_FxLMS(Wc,Sec,muW);
[E,a]= a.FxLMS_cannceller(Red,Dir);
```

4.3 Drawing the figure of error signals

Drawing the error signals of the four error microphones.

```
subplot(2,2,1)
plot(E(1,:));
xlim([0 650000])
xlabel('Time')
ylabel('Amplitude')
title('Error␣Microphone␣1')
grid on ;
subplot(2,2,2)
plot(E(2,:));
xlim([0 650000])
xlabel('Time')
ylabel('Amplitude')
title('Error␣Microphone␣2')
grid on ;
subplot(2,2,3)
plot(E(3,:));
xlim([0 650000])
xlabel('Time')
ylabel('Amplitude')
title('Error␣Microphone␣3')
grid on ;
subplot(2,2,4)
plot(E(4,:));
xlim([0 650000])
xlabel('Time')
ylabel('Amplitude')
title('Error␣Microphone␣4')
grid on ;
```

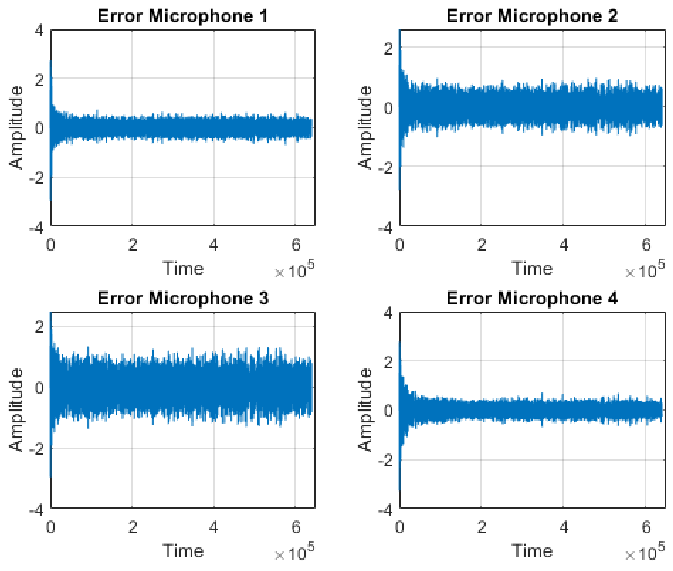

Figure 3: Simulated error signals of the MCANC system using the FxLMS algorithm.

4.4 Drawing the figure of control filter coefficients

Drawing the control filter coefficients of the four control filters.

```matlab
W1=a.Get_coefficients();
subplot(2,2,1)
plot(W1(:,1));
xlabel('Index')
ylabel('Amplitude')
title('Control Filter 1')
grid on ;
subplot(2,2,2)
plot(W1(:,2));
xlabel('Index')
ylabel('Amplitude')
title('Control Filter 2')
grid on ;
subplot(2,2,3)
plot(W1(:,3));
xlabel('Index')
ylabel('Amplitude')
title('Control Filter 3')
grid on ;
subplot(2,2,4)
plot(W1(:,4));
xlabel('Index')
ylabel('Amplitude')
title('Control Filter 4')
grid on ;
```

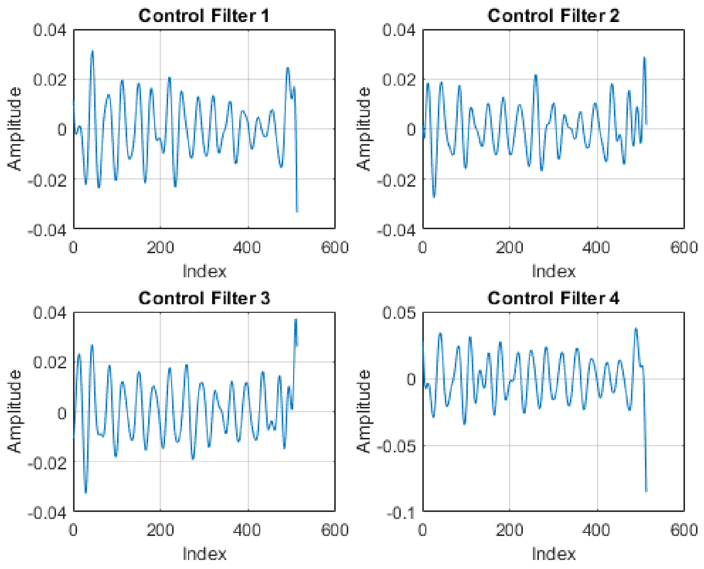

Figure 4: Simulated control filters coefficients of the MCANC system using the FxLMS algorithm.

5 Conclusion

The article offers an elaborate elucidation of a MATLAB program that emulates feedforward multichannel active noise control (ANC) by utilizing the filtered-x least mean square (FxLMS) technique. More precisely, the user has the freedom to select the quantity of reference microphones, secondary sources, and error microphones. Furthermore, the program provides the choice to utilize the GPU to enhance the computational efficacy.

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] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

[45] S. Dongyuan, "Algorithms and implementations to overcome practical issues in active noise control systems," Ph.D. dissertation, Nanyang Technological University, 2020.