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

Shi, Dongyuan; Gan, Woon-Seng; He, Jianjun; Lam, Bhan

2019

Shi, D., Gan, W.-S., He, J., & Lam, B. (2020). 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*, 28(4), 940-953. doi:10.1109/tvlsi.2019.2956524

<https://hdl.handle.net/10356/142541>

<https://doi.org/10.1109/TVLSI.2019.2956524>

© 2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at:
<https://doi.org/10.1109/TVLSI.2019.2956524>

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

Dongyuan Shi, *Student Member, IEEE*, Woon-Seng Gan, *Senior Member, IEEE*, Jianjun He, *Member, IEEE*, and Bhan Lam

Abstract—Multichannel active noise control (MCANC) is widely recognized as an effective and efficient solution for acoustic noise and vibration cancellation, such as in large-dimension ventilation ducts, open windows, and mechanical structures. The feedforward multichannel filtered-x least mean square (FFMCFxLMS) algorithm is commonly used to dynamically adjust the transfer function of the multichannel controllers for different noise environments. The computational load incurred by the FFMCFxLMS algorithm, however, increases exponentially with increasing channel count and thus requiring high-end field programmable gate array (FPGA) processors. Nevertheless, such processors still need specific configurations to cope with soaring computing loads as the channel count increases. To achieve a high-efficiency implementation of the FFMCFxLMS algorithm with floating-point arithmetic, a novel architecture based on multiple-parallel-branch with folding (MPBF) technique, is proposed. This architecture parallelizes the branches and reuses the multiplier and adder in each folded branch so that the trade-off between throughput and the usage of the hardware resources is balanced. The proposed architecture is validated in an experimental setup that implements the FFMCFxLMS algorithm for the MCANC system with 24 reference sensors, 24 secondary sources, and 24 error sensors, at a sampling and throughput rates of 25 kHz and 260 Mbps, respectively.

Index Terms—Multichannel Active Noise Control, FxLMS, Field Programmable Gate Array (FPGA), Parallel and Folded Structure.

I. INTRODUCTION

ACTIVE noise control (ANC), which generates an equal-amplitude but anti-phase ‘anti-noise’ wave to attenuate the unwanted sound wave, has gradually become an efficient solution to deal with noise pollution, especially for low-frequency noise disturbances [1]. The effectiveness and limitations of the ANC technique have been widely reported [2]–[10]. Recently, with the availability of high-speed, low-cost digital signal processors (DSP), the single-channel implementation of ANC has been applied in noise ducts [11], [12], for functional magnetic resonance imaging (fMRI) [13]–[15], in headsets [16], [17], for power transformers [18], in the aircraft [19], and in incubators [20].

DongYuan Shi, Woon-Seng Gan and Bhan Lam are with the Digital Signal Processing Lab, School of Electrical and Electronic Engineering at the Nanyang Technological University, Singapore. (e-mail: DSHI003@e.ntu.edu.sg, ewsgan@ntu.edu.sg, and bhanlam@ntu.edu.sg).

Jianjun He is with the Maxim Integrated, USA. (Jianjun.He@maximintegrated.com).

TABLE I
SINGLE-CHANNEL ANC IMPLEMENTATIONS ON PROCESSORS [33]

Chip	L_w	L_s	Structure	Data width	F_s
TMS320C25 [34]	64	64	Feedback	16 bit	8
TMS320C50 [35]	16	256	Feedback	32 bit	19.2
TMS320C6711 [36]	-	-	Feedback	32 bit	8
STM32F407 [17]	32	32	Feedback	32 bit	4
PIC24H [37]	20	20	Feedback	16 bit	10

L_w : the length of the control filter; F_s : Sampling rate (kHz);
 L_s : the length of the secondary path estimate.

To ensure causality [21], [22], real-time ANC systems have a stringent requirement on total electrical latency or group delay, i.e., the amount of time taken to complete a single point of data processing. General-purpose processors, such as ARM or DSP, are suitable for embedded systems and signal processing applications based on batch (data block) processing [23]–[25]. Since they lack parallel processing capability and incur high latency on data communications between their memory components and periphery devices, most of these general-purpose processors are only suitable for single-channel feedback ANC with a low sampling frequency, short filter length and data width, as shown in Table I. The aforementioned feedback ANC systems avoid the causality constraint at the expense of limited noise reduction performance for broadband noise [26].

In the free-field, the diameter of ‘quiet zone’ generated by single-channel ANC systems is approximately one-tenth the acoustic wavelength [27], which is inadequate for larger control regions, such as in the control of noise through open windows for dwellings [28]–[30]. Hence, global control is usually achieved through a multichannel active noise control (MCANC) implementation with the feedforward multichannel FxLMS (FFMCFxLMS) algorithm. As the number of channels increases, the vast computational complexity largely exceeds the computing capability of the most high-speed DSPs. For this reason, the field programmable gate array (FPGA) with its abundant reconfigurable hardware resources and efficient parallel processing capability appears well suited to realize complicated adaptive algorithm [31], [32], such as the MCANC.

There are existing computationally efficient single-channel ANC algorithms implemented on FPGA [38]–[42], as shown

TABLE II
SINGLE-CHANNEL ANC IMPLEMENTATIONS ON FPGA

Device	L_w	L_s	Algorithm	Data width	F_s
VLSI [45]	24	64	FxLMS	16 Bit	96
NI 7388R [13]	-	1024	NFxLMS	32 Bit	40
EP2S180F [46]	64	-	LMS	16 Bit	-
XC7Z010 [33]	40	40	Systolic FxLMS	32 Bit	-
cRIO [42]	6	-	Notch filter	32 Bit	20

in Table II. For instance, to realize the single-channel FxLMS algorithm, additional delays are inserted to enable its pipelining feature. This architecture is known as the delay FxLMS (DFxLMS) [43] algorithm and has a similar structure to the DLMS algorithm [31], [44]. A variant structure, known as the delayed block FxLMS (DBFxLMS) algorithm [43] has an improved parallel performance. Other improved algorithms have also been proposed, such as the systolic FxLMS algorithm that utilizes the pipelining and systolic structure on FPGA to gain a higher throughput [33]. However, these applications mainly focus on signal-channel noise control. Most of these algorithms sacrifice the convergence performance to achieve computational efficient or high throughput. Besides, without any optimizations on the computation amount, these architectures are not suitable for scaling up to multichannel applications.

Other types of modified algorithms [47], [48], such as the adjoint FxLMS algorithm [49], the mixed error algorithm [29], and the fast multichannel FxLMS algorithm [50] reduce the computational load by minimizing the number of secondary path models used to filter the reference signals or by removing the delays of the secondary paths within the coefficient updates. The mini-max algorithm [51], [52], which is based on the partial-updating method, reduces the computation load for coefficient updates in the control filters. In realizing a J (reference signals) $\times J$ (secondary sources) $\times M$ (error signals) MCANC system, as exhibited in Table III, these algorithms have a reduced computational load compared to the conventional FFMCFxLMS algorithm, but at the expense of slowing down the algorithm convergence speed and degrading the noise reduction performance.

However, there are few dedicated architectures or intellectual property (IP) of the FFMCFxLMS algorithm for FPGA implementation. To fill this gap, we propose a multiple-parallel-branch with folding (MPBF) architecture for the implementation of the FFMCFxLMS algorithm. This architecture balances the trade-off between throughput and hardware resource consumption, resulting in the requirement of only $2J$ parallel multiplier-accumulators (MAC) to realize a $J \times J \times M$ MCANC on FPGA. The proposed architecture consists of a parallel structure to increase its throughput, while the folded structure minimizes the hardware resource consumption on the FPGA. Furthermore, this proposed architecture only modifies the implementation structure without affecting the performance of the FFMCFxLMS algorithm.

This paper is organized as follows. Section II revisits the FFMCFxLMS algorithm and highlights its implementation issues. Section III describes the proposed multiple-parallel-branch with folding architecture of the FFMCFxLMS. Section

TABLE III
COMPUTATIONAL COMPLEXITY OF $J \times J \times M$ MCANC ALGORITHMS

Algorithm	Multiply-accumulate operations
FFMCFxLMS Algorithm [53]	$(2M + 1)J^2L + M$
Adjoint MCFxLMS Algorithm [49]	$2J^2L + MJL$
Mixed-error Algorithm [29]	$3J^2L + 1$
Minimax Algorithm [54]	$(2 + M)J^2L + M$
P-U Minimax Algorithm [51]	$3J^2L + 1$
Fast MCFxLMS Algorithm [50]	$2J^2L + (3 + M)JL + M - 3J$
Slowing Adaptation [47]	$(M + 2)J^2L + 1$
Scan Error Algorithm [48]	$3J^2L + 1$

* L denotes the length of the control filter and secondary path estimate.

IV describes the implementation of the proposed architecture on FPGA in detail and outlines the key results. Section V presents the experimental result of a real-time $24 \times 24 \times 24$ multichannel ANC applied on an open window. The last section concludes this paper. Table IV summaries the symbols used in the following sections.

II. PROBLEM STATEMENT

Figure 1 shows the block diagram of the MCANC, which consists of J reference sensors, J secondary sources, and M error sensors. By using the stacked coefficient vector of the J control filters $\mathbf{w}(n) = [\mathbf{w}_1^T(n), \mathbf{w}_2^T(n), \dots, \mathbf{w}_J^T(n)]^T$, where $\mathbf{w}_j(n) = [w_{j,1}(n), w_{j,2}(n), \dots, w_{j,L}(n)]^T$ is the coefficient vector of the j th control filter, the MCANC creates J control signals $\mathbf{y}(n) = [y_1(n), y_2(n), \dots, y_J(n)]^T$. These control signals propagate through acoustic secondary paths to cancel the noise disturbance $\mathbf{d}(n) = [d_1(n), d_2(n), \dots, d_M(n)]^T$ at the error sensor locations. The transfer function $\mathbf{P}$ denotes the primary paths from the primary source to the error sensors, and the transfer function $\mathbf{S}$ represents the physical secondary paths from the secondary sources to the error sensors, and its estimate is denoted as $\hat{\mathbf{S}}$. The matrix of reference signals picked up by J reference sensors is written as

$$\mathbf{X}(n) = \text{diag}\{\mathbf{x}_1(n), \mathbf{x}_2(n), \dots, \mathbf{x}_J(n)\} \quad (1)$$

where $\text{diag}\{\cdot\}$ denotes the diagonal elements of a matrix, and the j th reference signal vector is stated as $\mathbf{x}_j(n) = [x_j(n), x_j(n-1), \dots, x_j(n-L+1)]^T$. Hence, the output signal vector of J secondary sources is given by

$$\mathbf{y}(n) = \mathbf{X}^T(n)\mathbf{w}(n). \quad (2)$$

In the FFMCFxLMS algorithm, the weight updating equation of the j th control filter is given by [26]

$$\mathbf{w}_j(n+1) = \mathbf{w}_j(n) + \mu \sum_{m=1}^M \mathbf{x}'_{jm}(n)e_m(n) \quad (3)$$

where μ denotes the step size, and $e_m(n)$ represents the error signal picked up by the m th error sensor. The filtered reference signal is expressed by

$$x'_{jm}(n) = \hat{\mathbf{s}}_{mj}^T \mathbf{x}_j(n) \quad (4)$$

where $\hat{\mathbf{s}}_{mj} = [\hat{s}_{mj,1}, \hat{s}_{mj,2}, \dots, \hat{s}_{mj,L}]^T$ denotes the estimate of the secondary path from the j th secondary source to the m th error sensor.

TABLE IV
SUMMARY OF VARIABLES

Symbol	: Definition of the variable.
J	: Number of the reference microphones.
j	: Index of the reference microphone and secondary source $j \in [1, J]$.
M	: Number of the error microphones.
m	: Index of the error microphone $m \in [1, M]$.
L	: Length of the control filter and secondary path estimate.
l	: Index of the filter tape $l \in [1, L]$.
$x_j(n)$	: Reference signal picked up by the j th microphone.
$\mathbf{x}_j(n)$	: Reference signal vector $[x_j(n), x_j(n-1), \dots, x_j(n-L+1)]^T$.
$X_j(z)$	: The z -transform of the reference signal $x_j(n)$.
$x'_{jm}(n)$	: Filtered reference signal $\hat{\mathbf{s}}_{mj}^T \mathbf{x}_j(n)$.
$\mathbf{x}'_{jm}(n)$	: Filtered reference vector $[x'_{jm}(n), x'_{jm}(n-1), \dots, x'_{jm}(n-L+1)]^T$.
$X''_{jm}(l)$	: The l th iteration output in the secondary path model.
$x'_{jm}(l)$	: The inverse z -transform of $X''_{jm}(l)$.
$\hat{\mathbf{s}}_{mj}$	: Secondary path estimate from the j th secondary source to the m th error microphone.
$\hat{s}_{mj,l}$	: The l th element of the vector $\hat{\mathbf{s}}_{mj}$.
$y_j(n)$	: Control signal of the j th control filter.
$\mathbf{y}(n)$	: Control signal vector $[y_1(n), y_2(n), \dots, y_J(n)]^T$.
$Y_j(z)$	: The z -transform of the control signal $y_j(n)$.
$Y'_{j,l}(z)$	: The l th iteration output of the j th control filter.
$y'_{j,l}(n)$	: The inverse z -transform of $Y'_{j,l}(z)$.
$e_m(n)$	: Error signal picked up by the m th error microphone.
$\mathbf{w}_j(n)$	: Coefficient vector of the j th control filter $[w_{j1}(n), w_{j2}(n), \dots, w_{jL}(n)]^T$.
$\mathbf{w}(n)$	: Stacked coefficient vector of all the control filters $[\mathbf{w}_1^T(n), \mathbf{w}_2^T(n), \dots, \mathbf{w}_J^T(n)]^T$.
$W_{j1}(z)$	: The z -transform of the weight element $w_{j1}(n)$.

Fig. 1. A $J \times J \times M$ MCANC using the FxLMS algorithm [26].

By observing (2) and (3), a $J \times J \times M$ FFMCFxLMS

Fig. 2. Block diagram of the j th control filter.

algorithm requires $J \times L \times 2(M+1)$ additions and multiplications in each iteration. For instance, a $24 \times 24 \times 24$ MCANC based on the FFMCFxLMS algorithm with a sampling rate of 25 kHz and filter lengths of $L = 200$ (the control filter and secondary path estimate) amounts to 6 billion MAC operations per second. In comparison, the high-end Texas Instruments TMS320C674X DSP can only handle up to 2.7 billion MAC operations per second [55]. Hence, the FPGA is better equipped to realize MCANC with its parallel operations that potentially allows all the arithmetic computations to be completed at the same time. However, without appropriate optimization, FPGA would consume a tremendous amount of hardware resources to implement the FFMCFxLMS algorithm, and it is thus impractical to scale.

III. PROPOSED ARCHITECTURE FOR FFMCFxLMS ALGORITHM

The challenge of implementing the FFMCFxLMS algorithm on FPGA is to achieve both high throughput rate and low hardware resource consumption. To address this issue, we propose a new architecture for the FFMCFxLMS algorithm, which integrates both the parallel and folded structures that improve throughput and reduce hardware resources, respectively.

A. Parallel Branch

To parallelize the FFMCFxLMS architecture [56], we propose to split the $J \times J \times M$ multichannel FxLMS algorithm into J identical branches, which are equivalent to J parallel processors. Each branch includes a control filter, a weight updating block, and $J \times M$ secondary path model blocks. These branches run in parallel simultaneously, which yields J times faster processing speed and higher throughput than sequential execution by one single processor.

B. Recursive formula derivation

In this architecture, we let the branch execute L (the length of control filter is L) iterations to complete one computing cycle¹, which reduces the hardware resource usage of the algorithm on FPGA. The recursive formula of the FFMCFxLMS algorithm was derived in the z -transform domain.

The block diagram of the control filter in the j th branch is illustrated in Fig. 2. $X_j(z)$ denotes the z -transform of the j th reference signal, and $w_{j,l}$ represents the l th weight in the j th

¹One cycle: the duration that the filter processes one input data. Here, 1 cycle = L iterations, and 1 iteration = $M + 1$ clock periods.

Fig. 3. Block diagram of the reference signal filtering.

control filter. Hence, the recursive formula of the j th control filter's output can be expressed as

$$Y_j(z) = Y'_{j,L}(z) = Y'_{j,L-1}(z) + w_{j,L}X_j(z)z^{-L+1} \quad (5)$$

where $Y'_{j,l}(z)$ is the output of the l th iteration and given by

$$Y'_{j,l}(z) = Y'_{j,l-1}(z) + w_{j,l}X_j(z)z^{-l+1} \quad (6)$$

and the first iteration output is stated as

$$Y'_{j,1}(z) = w_{j,1}X_j(z). \quad (7)$$

Here, $j \in [1, J]$ and $l \in [1, L]$. By using (6) and (7), we can transform one computing cycle of the j th control filter into L iterations, which is indicated by dashed lines in Fig. 2.

The weight updating equation of the j th control filter is given by (3), the l th element of which can be rewritten in the z -transform as

$$W_{j,l}(z) = \left[W_{j,l}(z) + \mu \sum_{m=1}^M X'_{jm}(z)z^{-l+1}e_m \right] z^{-1} \quad (8)$$

where e_m and $X'_{jm}(z)$ denote the m th error signal and the filtered reference signal, respectively.

The transposed filter is utilized to realize the filtering of the reference signal, which increases the throughput compared to the conventional transversal filter [25], [56]. Each branch possesses M secondary path models. The block diagram of the m th secondary path mode in the j th branch is illustrated in Fig. 3, in which $\hat{s}_{mj,l}$ represents the l th coefficients of the secondary path estimate from the j th secondary source to the m th error microphone. Hence, the recursive formula of the filtered reference signal is given by

$$X'_{jm}(z) = X''_{jm}(1) = \hat{s}_{mj,1}X_j(z) + X''_{jm}(2)z^{-1} \quad (9)$$

where

$$X''_{jm}(2) = \hat{s}_{mj,2}X_j(z) + X''_{jm}(3)z^{-1}; \quad (10)$$

analogously, the output of the l th iteration can be expressed as

$$X''_{jm}(l) = \hat{s}_{mj,l}X_j(z) + X''_{jm}(l+1)z^{-1} \quad (11)$$

and the final iteration result is stated as

$$X''_{jm}(L) = \hat{s}_{mj,L}X_j(z). \quad (12)$$

By combining (6), (8) and (11), we outlined the recursive iteration for one computing cycle of the $J \times J \times M$ FFMCFxLMS algorithm, whose block diagram is shown in Fig. 4. In the figure, the multichannel algorithm is divided into J parallel branches, and each branch consists of $2(M+1)$ multipliers and adders.

Fig. 4. Block diagram of the FFMCFxLMS based on J branches [57].

In summary, the J branches require L iterations to complete one computing cycle of the algorithm. Hence, compared to the original implementation structure of the FFMCFxLMS algorithm, the proposed architecture utilizes L times less hardware resource at this stage.

C. Folded structure derivation

In Fig. 4, each branch of the FFMCFxLMS algorithm consists of many identical operators (adders and multipliers). Hence, we can reuse some of them to complete all the computations in each branch, which further reduces the hardware resource usage of the FFMCFxLMS algorithm on FPGA. To derive the folded structure of each branch, we adopted a modified folding method as described in the following procedure:

1. Assigning the reused unit. The redrawn data-flow graph (DFG) of the j th branch is shown in Fig. 5. In the DFG, there are $2(M+1)$ additions and multiplications, which can be executed by two MAC units, S_0 and S_1 , according to a particular running sequence or folding order. S_0 is assigned to complete $M+1$ additions and multiplications of the weight updating in the branch; S_0 is used to complete the other $M+1$ additions and multiplications coming from the control filter and reference signal filtering tasks, as shown in Fig. 5.

Fig. 5. Data-flow graph of the j th branch.

2. Scheduling folding order. The folding order refers to the executing sequence of the MAC. In the paper, the folding order corresponds to the data flow direction in the DFG, which can avoid the negative delay in the traditional folding method. The operation label $(S_0 | m)$ in Fig. 5 denotes that this operation will be executed by S_0 in the m th period.

3. Labeling the ports of nodes. We notated all the adding and multiplying operations as $2(M+1)$ nodes in arbitrary sequence, as shown in Fig. 5. Next, the ports of the node are labeled as A, B, C, D, E, and H, which indicate the connections between different nodes.

4. Calculating the new delay time. To derive the new delay time in the folded structure of Fig. 5, we utilized the folding equation [56] as follows

$$D_F(U \rightarrow V) = (M+1)w(e) + v - u, \quad (13)$$

where $w(e)$ is the delay from node U to node V , whose folding order are u and v , respectively. The derived delays are listed in Table V. It is worth noting that the delay of all the data paths equate to one sampling period, as a result of the folding order of the DFG being set along the direction of data flow, as illustrated in Fig. 5.

TABLE V
THE NEW DELAY ON EACH DATA PATH.

Path $D_F(U \rightarrow V)$	Folding equation	Delay time
$M+1 \rightarrow M+2$	$(M+1) \times 0 + M+2 - M-1$	1
$M+2 \rightarrow M+3$	$(M+1) \times 0 + M+3 - M-2$	1
$M+3 \rightarrow M+4$	$(M+1) \times 0 + M+4 - M-3$	1
$\dots$	$\dots$	1
$2M \rightarrow 2M+1$	$(M+1) \times 0 + 2M+1 - 2M$	1

*Note: One delay time is equivalent to one sampling period.

Fig. 6. The folded structure of the j th branch.

5. Constructing the folded structure. According to the new delay time table, the folded structure of the j th branch can be constructed, as illustrated in Fig. 6. The label $\{m\}$ on the data route indicates that the switch connects to this route in the m th clock period, and l denotes the l th iteration. These labels construct a specific execution schedule for the switches. Accordingly, the switches in the folded structure select different routes based on its schedule so that the MAC can sequentially process the data to complete one iteration. These switches can be readily implemented by the multiplexers and de-multiplexers in the digital circuit. Between the adjacent iterations, the data communication is realized by the dual-port block random-access memories (BRAM), which will be explained in detail in the next section.

Therefore, taking advantage of recursive formula and folded structure, the architecture saves the hardware consumption of additions and multiplications by $L(M+2)$ times.

IV. IMPLEMENTATION OF THE PROPOSED ARCHITECTURE ON FPGA.

By integrating J parallel branches with the folded structures as described above, a new architecture for the $J \times J \times M$ FFMCFxLMS algorithm is obtained, as illustrated in Fig. 7. This architecture is composed of J identical branches. At the n th computational cycle, the j th branch in the architecture picks up the j th reference signal $x_j(n)$ and all error signals

Fig. 7. Proposed architecture of the FFMCFxLMS algorithm.

from $e_1(n)$ to $e_M(n)$. Subsequently, it spends L iterations to generate the j th output signal $y_j(n)$. During each iteration, the branch firstly feeds input data to two MACs through the input ports (A, B, C, and E). After $M + 1$ system clock (CLK_S) periods, it obtains the iteration results from the output ports D and H. Meanwhile, the other parallel branches execute the same procedure simultaneously.

The synchronous design is used to implement the proposed architecture on FPGA, which makes the design more reliable and simplifies the timing issue. In the implementation, two critical issues have to be first addressed: realizing the control of the switches in the folded structure, and implementing the data communication and storage between iterations.

A. Finite-state machine to control the switches

With the assistance of the switches in the branch, two MACs are reused $M + 1$ times to complete one iteration according to the folding order in Fig. 6. In the proposed architecture, a Mealy finite-state machine (FSM) is adopted to control these switches, which are in charge of the data communication between MAC and ports. The FSM comprises $M + 1$ states and outputs the control signals through the seven control lines (ctr-a to ctr-h), as shown in Fig. 7. These states circularly transfer

(a) Statechart diagram of the Mealy finite state machine

Action 0	Action 1	Action m*	Action M
If iteration = 0 ctr-h= x'_{jM} else ctr-h= x''_{j1} End ctr-a= x'_{jM} ctr-b= e_M ctr-c=0 ctr-e=IN ctr-d=E1 ctr-f= $\hat{s}_{1j}$ ctr-g= x''_{j1}	If iteration = 0 ctr-h= $x'_{j(M-1)}$ else ctr-h= x''_{j2} End ctr-a= $x'_{j(M-1)}$ ctr-b= e_{M-1} ctr-c=D1 ctr-e=IN ctr-d=E1 ctr-f= $\hat{s}_{2j}$ ctr-g= x''_{j2}	If iteration = 0 ctr-h= $x'_{j(M-m)}$ else ctr-h= $x''_{j(m+1)}$ End ctr-a= $x'_{j(M-m)}$ ctr-b= e_{M-m} ctr-c=D1 ctr-e=IN ctr-d=E1 ctr-f= $\hat{s}_{(m+1)j}$ ctr-g= $x''_{j(m+1)}$	If iteration = L-1 ctr-h= y_j else ctr-h=E2 End ctr-a=E1 ctr-b= μ ctr-c= w_j ctr-e= x_j ctr-d= w_j ctr-f= w_j ctr-g=D2

*Note: $m \in [1, M - 1]$

(b) The detailed actions in each state of FSM

Fig. 8. Mealy finite state machine used to control the multiplexers in the proposed architecture.

to each other at the pace of the system clock, as illustrated in Fig. 8 (a). Each state has its corresponding action to output control signals, which is listed out in Fig. 8 (b). These control signals decide the inter-connections of the multiplexers and de-multiplexers to realize the function of the switches in Fig. 6. The output of the multiplexers and de-multiplexers during each clock period is exhibited in Fig. 9. In the FSM, the 'iteration' signal indicates the index of the iteration. Since the branch requires L iterations to complete one computational cycle, an L mode counter is used to create the 'iteration' signal. The rst is the reset signal. When it is activated, the current state will return to the state 0, and all the registers are cleared to prevent unpredictable state. Furthermore, to avoid the risk of competition [58] and obtain enough setup time and hold on time for the input data, the FSM takes CLK/S as the clock signal, which is generated by shifting the system clock CLK_S by 180°, as illustrated in Fig. 9.

B. Data communication between iterations

After every iteration, the branch needs to store the computed result and fetch new data for the next iteration. Again, we

Fig. 9. Read and write timing of the MAC ports.

(a) The address generator of BRAM.

(b) Read and write timing of BRAM.

Fig. 10. The schematic and timing diagrams of the BRAM address generator.

can utilize the folding technique to schedule the data communication sequence between iterations. The folded structure used between iterations, however, requires a significant amount of registers whose quantity is limited by the FPGA [59]. The workaround still requires a complicated control scheduler and a large number of registers, even when combining with retiming and lifetime analysis techniques [56], [60]. Therefore, in this implementation, we replaced the registers with the block random access memory (BRAM), which are abundant and under-utilized in the FPGA. Although BRAM runs slower than the register, the data communication between iterations

Fig. 11. The schematic diagram of the divider.

TABLE VI
THE PERIODS AND INITIAL PHASE OF CLOCK SIGNALS USED IN THE PROPOSED ARCHITECTURE

Clock name	Period	Initial phase	Function
CLK*	1	0°	Hardware clock
CLK_S	1	0°	system clock
CLK/S	1	180°	Inverse system clock
CLK_R	M+1	0°	Reading BRAM
CLK_W	L(M+1)	0°	Writing BRAM

* denotes the hardware clock of the device.

happens at the interval of $M + 1$ system clocks, which coincides with the reading frequency of BRAM. In addition, the dual-port BRAM also buffers the data between different clock domains.

In the architecture, each branch utilizes $3M + 2$ dual-port BRAMs to realize data communication and temporary storage between iterations. Among them, the BRAM x_j , x'_{jm} , and x''_{jm} implement the delay lines in the DFG of the j th branch. Meanwhile, the BRAM w_j and $\hat{s}_{mj}$ store the coefficients of the j th control filter and the secondary path model from the j th secondary source to the m th error sensor. The address signals (A1, B1, and C1) of these BRAMs are generated by a L mode counter whose block diagram is illustrated in Fig. 10 (a). The L mode counter consists of three adders, three multiplexers, and two latches. It is driven by the reading clock, CLK_R, and generates the address signals, as shown in Fig. 10 (b). The rst signal is used to clean its memory and reset its initial value. It is worth noting that A1 is always greater than B1 by 1, and C1 remains no change in the L th period, which helps the BRAMs x_j and x'_{jm} to realize the delay lines by the circular shifting operation and thus, reduces the computing consumption using for the data shifting process.

Furthermore, there are two clocks CLK_R and CLK_W driving these BRAMs in the architecture. The CLK_R, having $M + 1$ periods of the system clock CLK_S, drives both reading and writing operations of the BRAM w_j , $\hat{s}_{mj}$, x'_{jm} , and the reading operation of the BRAM x_j and x'_{jm} , as shown in Fig. 9. The clock CLK_W, whose period is L times that of CLK_R, is used to trigger the writing operation of the BRAMs x_j and x'_{jm} . The timing diagram of these clock signals is exhibited in Fig.10 (b).

C. Clock divider

As mentioned in the previous subsection, there are three derived clock signals, CLK/S, CLK_R, and CLK_W, of system clock CLK_S in the architecture. They need to be synchronized but running at different frequencies. In FPGA, all these

Fig. 12. The equivalent diagram of the Two-MAC.

clock signals are assigned to the global tree to reduce skew and duty cycle distortion and improve the jitter tolerance. For synthesizing these derived clocks, we implemented a novel divider, as shown in Fig. 11. It comprises of a 32-bit accumulator and two control ports: an input clock port and a frequency control port. During operation, the input clock signal drives the accumulator to increase gradually at the step size of a frequency tuning word (FTW), which is loaded from the frequency control port. Once the value of the accumulator exceeds 2^{31} , the most significant bit (MSB) of the accumulator will toggle its original value to create the desired output clock with a 50% duty cycle.

This divider works like the direct digital synthesizer (DDS) [61], in which the frequency f_{out} of the output clock is decided by the FTW. The relationship between f_{out} , FTW, and the frequency of the system clock f_{CLK_S} is given by

$$f_{out} = \left\lceil \frac{FTW}{2^{32}} \right\rceil \times f_{CLK\_S} \quad (14)$$

where FTW is a 32-bit integer from 0 to $2^{31} - 1$. Therefore, the FTW for CLK_R and CLK_W can be derived to $\text{round}\left[\frac{2^{32}}{2M+2}\right]$ and $\text{round}\left[\frac{2^{32}}{L(2M+2)}\right]$ from (14), where the $\text{round}[\cdot]$ function rounds the argument to the nearest integer. Equipped with this divider, all the clock signals used in the proposed architecture can be synthesized as shown in Table VI.

D. The floating-point multiplier-accumulator

As shown in Fig. 7, each branch has two identical multiplier-accumulators. The two-MAC has seven ports (A, B, C, D, E, F, G, and H), as shown in Fig. 12, which are 32-bit wide. To reduce the finite word effect and improve the representation precision and stability in dealing with large dynamic range signal [42], [62], this two-MAC adopts floating-point arithmetic operation.

Altogether, the folded structure, BRAMs, and the clock divider of the proposed architecture implemented on FPGA requires $L(M+1)$ clock periods to accomplish one computing cycle of the $J \times J \times M$ FFMCFxLMS algorithm. However, the reduction of MACs by $L(M+1)$ times and owing to the J parallel branches, a faster processing speed is still achieved over the sequential structure.

TABLE VII
DEVICE UTILIZATION OF THE MULTICHANNEL FxLMS ALGORITHM ON
KINTEX-7 7K325T.

Item	Ratio of the device utilization
Total slices	97.1% (49494 out of 50950)
Slice registers	44.8% (182665 out of 407600)
Slice LUTs	63.9% (130214 out of 203800)
Block RAMs	36.2% (161 out of 445)
DSP48s	57.1% (480 out of 840)

V. IMPLEMENTATION RESULTS

To verify the proposed architecture, we implemented a $24 \times 24 \times 24$ floating-point FFMCFxLMS algorithm, which comprised of 24 control filters and 576 secondary path estimates. The algorithm was programmed on the National Instruments (NI) 9039 Compact RIO (with Xilinx Kintex-7 7K325T FPGA as the main processor) equipped with NI 9220 ADC and NI 9264 DAC modules. In this architecture, we used 24 parallel branches and set the length L of both the control filters and the secondary path estimates to 200 taps.

A summary of the device utilization after compilation on Vivado v2015.4 is shown in Table VII. It is evident that the proposed architecture balances the usage of MAC, registers, and BRAMs. In comparison, the conventional FFMCFxLMS requires $2JL(M+1) = 28800$ MACs, whereas the proposed architecture demonstrated a significant savings of $2JL(M+1) - 2J = 28752$ MACs.

Since the architecture requires $L(M+1) = 5000$ clock periods to complete one computing cycle, the maximum achievable sample rate f_s was 25 kHz when the FPGA runs with a system clock of 125 MHz. The maximum floating-point operations per second (FLOPS) of the architecture can be computed as

$$\text{FLOPS} = 2 \times J \times L \times 2(M+1) \times f_s = 12 \times 10^9. \quad (15)$$

Due to each branch receives 1 reference (16 bits) signal, 24 error (16 bits) signals, and 1 step size (32 bits) inputs, the throughput rate of the architecture is given by

$$R_{th} = 24 \times (25 \times 16 + 32) \times f_s = 260 \times 10^6 \text{ (bit/s)}. \quad (16)$$

VI. REAL-TIME ACTIVE CONTROL OF NOISE THROUGH THE OPEN WINDOW

To evaluate the performance of the $24 \times 24 \times 24$ MCANC system, active control units were distributed across the open aperture of a $1 \text{ m} \times 0.5 \text{ m}$ sliding window mounted on the wall of a $2.1 \text{ m} \times 2.1 \text{ m} \times 2.1 \text{ m}$ cuboid room, as shown in Fig. 13. The 24 identical ANC units were placed to a security grille. Each unit consisted of a 4.5 cm diameter loudspeaker facing into the room interior at one end and a reference electric microphone facing the noise source at the other end, 12 cm apart, as illustrated in Fig. 14. The interior of the ANC unit was filled with the dampening material to reduce acoustic feedback from the loudspeaker to the reference microphone. The error microphone array was placed in the room, 20 cm away from the ANC units. The primary noise to be attenuated was generated by a loudspeaker (GENELEC 8341a) mounted 2 m away from the window.

(a) The front view of the ANC window

(b) The schematic diagram of the ANC window

Fig. 13. The 24-channel ANC window.

Fig. 14. The front and cross-sectional views of the ANC unit.

To evaluate the noise reduction performance, the energy-average sound pressure level (SPL) was calculated from five observation microphones (G.R.A.S. 40PH) distributed in the room interior according to ISO 16283 [63], as depicted in Fig. 15.

The connections between the 24-channel ANC units and the FPGA system is described in Fig. 16. The reference and error signals picked up by the microphones were conditioned by their pre-amplifiers and anti-aliasing filters, and were subsequently converted to digital signals with a data-width of 16 bit by the NI-9220 ADC modules. The ADC modules transmitted the data to the FPGA (Kintex-7 7K325T) via the PXIe bus. After computation, the FPGA transmitted the output

Fig. 15. The positions of the observation microphones.

Fig. 16. The implementation diagram of the 24-channel ANC system.

data to the NI-9264 DAC modules that have a maximum data update rate of 25 kHz. Finally, with the aid of the DACs, reconstruction filters, and audio amplifiers, the output data were converted back to analog signals that drive the secondary sources to generate the anti-noise waves.

A. Tonal noise cancellation

In the first experiment, tonal noise from 100 Hz to 2500 Hz at intervals of 100 Hz were the primary disturbances to be controlled. The time-averaged SPL of the monitoring microphone A, located 50 cm away from the window, with the MCANC was turned on and off, is illustrated in Fig. 17.

Fig. 17. The time-averaged SPL of the monitoring microphone A as a function of frequency in the active control of tonal disturbances.

Fig. 18. The SPL of monitoring microphone B, C, D, E for sine tones.

The maximum noise reduction of the 24-channel ANC was 35.1 dB at 1800 Hz; the minimum noise reduction was 3.6 dB at 2400 Hz.

The SPL at monitoring microphone B, positioned in the center of the room, is shown in Fig. 18 (a) for the same range of tonal signals. A maximum noise reduction was recorded at 32.2 dB for 1600 Hz, and the minimum noise reduction was 0.7 dB at 2300 Hz. Fig. 18 (b) depicts the SPL of monitoring phone C, where its maximum noise reduction was 19.9 dB at 2100 Hz. However, its SPL increased by 12.6 dB at 2300 Hz when MCANC was turn on. The SPL of the monitoring microphone D is shown in Fig. 18 (c). Its maximum noise reduction was 12.9 dB at 1900 Hz. Its SPL increased at higher frequencies. Finally, the SPL of monitoring microphone E, sheltered by the glass panel of the sliding window, is shown in Fig. 18 (d). When the MCANC system was turned on, the noise level reduced slightly and even increased at higher frequencies. The poor performance at higher frequencies can be attributed to the fundamental limits of control of the MCANC system in an open aperture [64], [65].

Figure 19 shows the energy-average noise reduction level

Fig. 19. The energy-average noise reduction level of microphone A, B, C, D, and E.

TABLE VIII
BROADBAND NOISE REDUCTION LEVEL AT THE LOCATIONS OF THE MONITORING MICROPHONES.

Position	A	B	C
Reduction	10.1 dB	9.5 dB	7.2 dB
Position	D	E	Energy-average
Reduction	3.6 dB	3.5 dB	8.4 dB

(NR) of microphone A, B, C, D, and E for tonal noises at different frequencies, which is calculated by [63]

$$NR_{\text{average}} = 10 \log \left[\frac{\sum_{i=A}^{A,B,C,D,E} p_{\text{OFF},i}^2}{\sum_{i=A}^{A,B,C,D,E} p_{\text{ON},i}^2} \right] \quad (17)$$

where $p_{\text{OFF},i}$ and $p_{\text{ON},i}$ denote the SPL at the location of the microphone i when ANC is turned off and on, respectively. The MCANC system resulted in a the maximum energy-average noise reduction level of 17.5 dB at 600 Hz but negative noise reduction level at 1300 Hz, which may be caused by the acoustic mode of the room (one of oblique mode in the room is 1304 Hz). As these results indicate, the ANC window can achieve better global noise reduction of the room at lower frequencies.

B. Broadband noise cancellation

In the second experiment, the primary disturbance was a broadband noise from 500 Hz to 2000 Hz. Fig. 20 shows the power spectrum of the broadband noise at the monitoring microphone A when MCANC was turned off and on. It is evident from the result that the noise in the band was reduced, and the noise reduction level was 10.1 dB. Furthermore, the noise at other locations was also mitigated by the ANC window, as shown in Fig. 21 and Table VIII. Although the noise reduction level degraded as the distance between the monitoring microphone and the window gradually increased, the 24-channel ANC window still achieved global control for the broadband noise.

C. Metro noise cancellation

To figure out the performance of the ANC window in response to real noise, we measured the reduction level of the

Fig. 20. The power spectrum of the broadband noise (500-2000 Hz) at the monitoring microphone A.

Fig. 21. The power spectrum of the broadband noise (500-2000 Hz) at the monitoring microphone B, C, D, and E.

metro noise whose frequency ranges from 400 to 1200 Hz. In the experiment, the acoustic camera (TYPE 9712-W-FEN of *B&K*) was used to capture the holography of the metro noise propagating through the window. It was placed at the location of the monitoring microphone B.

In the experiment, the metro noise was generated by the primary speaker and directly propagated through the window. Hence, the noise source located at the primary speaker and was indicated by the dark yellow circle in Fig. 22 (a). Note that the dark color represents the high sound pressure level in the holography. In the figure, the average and central sound pressure level of the window were 64 dBA. Once the ANC was switched on, the average and central sound pressure level of the window decreased to 53.6 and 53.3 dBA, respectively. The noise source also shifted from the primary speaker to the left corner of the open window, which is labeled by the shallow green circle in Fig. 22 (b). At this moment, the secondary sources suppressed a large part of the metro noise played by the primary speaker. Only a few parts leaked through

(a) Holography of the metro noise propagating through the window.

(b) Holography of the metro noise propagating through the window when ANC is on.

Fig. 22. The holography of the ANC window when it cancels the real metro noise.

the window corner where there was no secondary source. Therefore, the opening ANC window worked as a virtual barrier, successfully blocking the propagation of the ambient noise without influencing the ventilation.

In summary, the results from experiments validated that the 24-channel FFMCFxLMS based on the proposed architecture was successfully implemented in FPGA, and it functioned well in a complex acoustic scenario of the open window. Furthermore, this MCANC achieved global control for broadband and tonal noise in a wide frequency range.

VII. CONCLUSION

The bottleneck for the implementation of the large-scale MCANC systems lies in the considerable amount of computation load required by the FFMCFxLMS algorithms, which largely exceeds the computing capability of today's digital signal processors. Even on a high-end FPGA, it is hard to realize the MCANC efficiently unless some dedicated architectures optimize it.

Therefore, this paper proposed a multiple-parallel-branch with folding floating-point architecture for the implementation of the FFMCFxLMS algorithm on FPGA. In realizing a $J \times J \times M$ FFMCFxLMS algorithm with the filter length of L taps, the proposed architecture requires only $2J$ MACs, which equates to a hardware resource reduction by $L(M+1)$ times. The parallel branches in the architecture enable the FPGA to execute J adaptive filters simultaneously, which increases the processing speed or the throughput rate by J times. So, with the aid of the folded and parallel structures, the proposed

architecture achieves a reasonable trade off between the lower hardware resource utilization and higher throughput of the system. Furthermore, the proposed architecture does not alter, nor reduce the operations of the FFMCFxLMS algorithm, and thus, will not compromise on its performance.

To verify the proposed architecture, we implemented a $24 \times 24 \times 24$ FFMCFxLMS algorithm based on the proposed architecture on an FPGA for real-time control of noise through a $1 \text{ m} \times 0.5 \text{ m}$ window opening into the room interior. The experiment results showed that a maximum tonal noise reduction of 35.1 dB and broadband noise reduction of 10.1 dB from 500 Hz to 2000 Hz was achieved directly in front of the window. The energy-average noise reduction levels in the room for the tonal and broadband noise were 17.6 dB and 8.4 dB, respectively. Furthermore, the proposed architecture achieved a computation performance of 12 billion floating-point operations per second (FLOPS) on the FPGA of Xilinx Kintex-7.

ACKNOWLEDGMENT

This material is based on research/work supported by the Singapore Ministry of National Development and National Research Foundation under the L2 NIC Award No.: L2NICCFP1-2013-7. This paper was also made possible by efforts from Dr. Rishabh Ranjan, Prof. Shi Chuang, Prof. Tatsuya Murao, and Nguyen Duy Hai for the window construction and data acquisition. This work was also inspired by input from Prof. Stephen J. Elliott and Prof. Masaharu Nishimura.

REFERENCES

- [1] Y. Kajikawa, W. S. Gan, and S. M. Kuo, "Recent advances on active noise control: open issues and innovative applications," *APSIPA Transactions on Signal and Information Processing*, vol. 1, 2012.
- [2] I. Tabatabaei Ardekani and W. H. Abdulla, "Effects of imperfect secondary path modeling on adaptive active noise control systems," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 5, pp. 1252–1262, Sep. 2012.
- [3] S. M. Kuo and M. T. and, "Convergence analysis of narrow-band active noise control system," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 46, no. 2, pp. 220–223, Feb 1999.
- [4] S. Wrona, K. Mazur, and M. Pawelczyk, "Internal model control of a washing machine casing for active noise reduction," in *Proceedings of the 24th International Congress on Sound and Vibration, London, UK, 2017*, pp. 23–27.
- [5] I. T. Ardekani and W. H. Abdulla, "Active noise control in three dimensions," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 6, pp. 2150–2159, Nov 2014.
- [6] A. Goel, A. Vetteth, K. R. Rao, and V. Sridhar, "Active cancellation of acoustic noise using a self-tuned filter," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 51, no. 11, pp. 2148–2156, Nov 2004.
- [7] S. M. Kuo, X. Kong, and W. S. Gan, "Applications of adaptive feedback active noise control system," *IEEE transactions on control systems technology*, vol. 11, no. 2, pp. 216–220, 2003.
- [8] M. Pawelczyk, "Analog active control of acoustic noise at a virtual location," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 2, pp. 465–472, March 2009.
- [9] P. Loiseau, P. Chevrel, M. Yagoubi, and J. Duffal, "Investigating achievable performances for robust broadband active noise control in an enclosure," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 1, pp. 426–433, Jan 2019.
- [10] K. Mazur, S. Wrona, and M. Pawelczyk, "Design and implementation of multichannel global active structural acoustic control for a device casing," *Mechanical Systems and Signal Processing*, vol. 98, pp. 877–889, 2018.
- [11] J. Hu and J.-F. Lin, "Feedforward active noise controller design in ducts without independent noise source measurements," *IEEE Transactions on Control Systems Technology*, vol. 8, no. 3, pp. 443–455, 2000.
- [12] I. D. Landau, R. Meléndez, L. Dugard, and G. Buche, "Robust and adaptive feedback noise attenuation in ducts," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 2, pp. 872–879, March 2019.
- [13] V. R. Ramachandran, I. M. S. Panahi, and E. Perez, "Active reduction of high-level acoustic noise on a fMRI test-bed using labview and FPGA platforms," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, March 2008, pp. 1517–1520.
- [14] M. Kumamoto, M. Kida, R. Hirayama, Y. Kajikawa, T. Tani, and Y. Kurumi, "Active noise control system for reducing MR noise," *IEICE transactions on fundamentals of electronics, communications and computer sciences*, vol. 94, no. 7, pp. 1479–1486, 2011.
- [15] R. M. Reddy, I. M. Panahi, and R. Briggs, "Hybrid FxRLS-FxNLMS adaptive algorithm for active noise control in fmri application," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 2, pp. 474–480, 2011.
- [16] S. M. Kuo, S. Mitra, and W. S. Gan, "Active noise control system for headphone applications," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 2, pp. 331–335, March 2006.
- [17] C. Chang and S. Li, "Active noise control in headsets by using a low-cost microcontroller," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 5, pp. 1936–1942, May 2011.
- [18] T. Zhao, J. Liang, L. Zou, and L. Zhang, "A new FXLMS algorithm with offline and online secondary-path modeling scheme for active noise control of power transformers," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 8, pp. 6432–6442, Aug 2017.
- [19] R. Cabell, D. Palumbo, and J. Viperman, "A principal component feedforward algorithm for active noise control: flight test results," *IEEE Transactions on Control Systems Technology*, vol. 9, no. 1, pp. 76–83, Jan 2001.
- [20] S. M. Kuo, L. Liu, and S. Gujjula, "Development and application of audio-integrated ANC system for infant incubators," *Noise Control Engineering Journal*, vol. 58, no. 2, pp. 163–175, 2010.
- [21] X. Kong and S. M. Kuo, "Study of causality constraint on feedforward active noise control systems," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 46, no. 2, pp. 183–186, 1999.
- [22] L. Zhang and X. Qiu, "Causality study on a feedforward active noise control headset with different noise coming directions in free field," *Applied Acoustics*, vol. 80, pp. 36 – 44, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0003682X1400005X>
- [23] D. Y. Shi and W. 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)*, March 2016, pp. 6280–6284.
- [24] J. Lorente, M. Ferrer, M. de Diego, and A. González, "Gpu implementation of multichannel adaptive algorithms for local active noise control," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 11, pp. 1624–1635, Nov 2014.
- [25] S. M. Kuo and W. S. Gan, *Digital signal processors: architectures, implementations, and applications*. Prentice Hall, 2005.
- [26] S. M. Kuo and D. Morgan, *Active noise control systems: algorithms and DSP implementations*. John Wiley & Sons, Inc., 1995.
- [27] S. J. Elliott and J. Cheer, "Modeling local active sound control with remote sensors in spatially random pressure fields," *The Journal of the acoustical Society of america*, vol. 137, no. 4, pp. 1936–1946, 2015.
- [28] B. Lam, C. Shi, D. Shi, and W. S. Gan, "Active control of sound through full-sized open windows," *Building and Environment*, 2018.
- [29] T. Murao, C. Shi, W. S. Gan, and M. Nishimura, "Mixed-error approach for multi-channel active noise control of open windows," *Applied Acoustics*, vol. 127, pp. 305–315, 2017.
- [30] X. Qiu, "Recent advances on active control of sound transmission through ventilation windows," in *Proceedings of the 24th International Congress on Sound and Vibration*, 2017.
- [31] Lok-Kee Ting, R. Woods, and C. F. N. Cowan, "Virtex FPGA implementation of a pipelined adaptive LMS predictor for electronic support measures receivers," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 13, no. 1, pp. 86–95, Jan 2005.
- [32] S. Mula, V. C. Gogineni, and A. S. Dhar, "Algorithm and vlsi architecture design of proportionate-type lms adaptive filters for sparse system identification," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 9, pp. 1750–1762, Sep. 2018.
- [33] D. Shi, C. Shi, and W. 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)*, Dec 2016, pp. 1–6.
- [34] S. M. Kuo, I. Panahi, K. M. Chung, T. Horner, M. Nadeski, and J. Chyan, "Design of active noise control systems with the TMS320 family," *Texas Instruments*, 1996.
- [35] S. Boucher, M. Bouchard, A. L'esperance, and B. Paillard, "Implementing a single channel active adaptive noise canceller with the TMS320C50 DSP starter kit," *Application Report: SPRA285*, 1997.
- [36] J. Lin, S. Tang, W. Han, C. Chuang, P. Liu, and S. Young, "Evaluation of speech intelligibility for feedback adaptive active noise cancellation headset," in *2006 International Conference on Biomedical and Pharmaceutical Engineering*, Dec 2006, pp. 24–29.
- [37] K. Shyu, C. Ho, and C. Chang, "A study on using microcontroller to design active noise control systems," in *2014 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, Nov 2014, pp. 443–446.
- [38] K. Chen, H. Vu, K. Weng, J. Huang, Y. Tsai, Y. Liu, and W. Wang, "Design of an efficient active noise cancellation circuit for in-ear headphones," in *2014 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, Nov 2014, pp. 599–602.
- [39] H. Vu, K. Chen, S. Sun, T. Fong, C. Hsu, and L. Wang, "A power-efficient circuit design of feed-forward FxLMS active noise cancellation for in-ear headphones," in *VLSI Design, Automation and Test (VLSI-DAT)*, April 2015, pp. 1–4.
- [40] H. Vu, K. Chen, and T. Fong, "Active noise control for in-ear headphones: Implementation and evaluation," in *2015 IEEE International Conference on Consumer Electronics - Taiwan*, June 2015, pp. 264–265.
- [41] M. S. Aslam, P. Shi, and C. Lim, "Variable threshold-based selective updating algorithms in feed-forward active noise control systems," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 2, pp. 782–795, Feb 2019.
- [42] A. Leva and L. Piroddi, "FPGA-based implementation of an active vibration controller," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 5077 – 5082, 2008, 17th IFAC World Congress. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1474667016397488>
- [43] B. K. Mohanty, G. Singh, and G. Panda, "Hardware design for VLSI implementation of FxLMS and FsLMS-based active noise controllers," *Circuits, Systems, and Signal Processing*, vol. 36, no. 2, pp. 447–473, 2017.
- [44] G. Long, F. Ling, and J. G. Proakis, "Corrections to 'the LMS algorithm with delayed coefficient adaptation'," *IEEE Transactions on Signal Processing*, vol. 40, no. 1, pp. 230–232, Jan 1992.
- [45] H. Vu and K. Chen, "A low-power broad-bandwidth noise cancellation vlsi circuit design for in-ear headphones," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 6, pp. 2013–2025, June 2016.
- [46] A. Jalali, S. G. Boroujeny, and M. Eshghi, "Design and implementation of a fast active noise control system on fpga," in *2007 Mediterranean Conference on Control & Automation*. IEEE, 2007, pp. 1–4.
- [47] S. D. Snyder and N. Tanaka, "Algorithm adaptation rate in active control: is faster necessarily better?" *IEEE Transactions on Speech and Audio Processing*, vol. 5, no. 4, pp. 378–381, July 1997.
- [48] H. Hamada, "Signal processing for active control," in *Proc. 1991 Acoust. Soc. Jpn. Int. Symp. Active Control of Sound and Vibration*, 1991, pp. 33–44.
- [49] E. A. Wan, "Adjoint LMS: an efficient alternative to the filtered-x LMS and multiple error LMS algorithms," in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol. 3, May 1996, pp. 1842–1845 vol. 3.
- [50] S. C. Douglas, "Fast implementations of the filtered-x LMS and LMS algorithms for multichannel active noise control," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 4, pp. 454–465, July 1999.
- [51] C. Shi and Y. Kajikawa, "A partial-update minimax algorithm for practical implementation of multi-channel feedforward active noise control," in *IWAENC*, 2018.
- [52] 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 *Intelligent Signal Processing and Communication Systems (ISPACS), 2017 International Symposium on*. IEEE, 2017, pp. 217–221.
- [53] S. Elliott, I. Stothers, and P. Nelson, "A multiple error LMS algorithm and its application to the active control of sound and vibration," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 10, pp. 1423–1434, October 1987.
- [54] A. Gonzalez, A. Albiol, and S. J. Elliott, "Minimization of the maximum error signal in active control," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 3, pp. 268–281, May 1998.
- [55] T. Instruments, "C6000 power-optimized DSP : Energy efficient single and multicore dsps for embedded processing applications," <http://www.ti.com/processors/dsp/c6000-dsp/overview.html/>, 2018, copyright 1995–2018 Texas Instruments Incorporated.
- [56] K. K. Parhi, *VLSI digital signal processing systems: design and implementation*. John Wiley & Sons, 2007.
- [57] D. Shi, J. He, C. Shi, T. Murao, and W. 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)*, March 2017, pp. 1188–1192.
- [58] J. F. Wakerly, *Digital design*. Pearson Australia Pty Limited, 2016.
- [59] C. Maxfield, *FPGAs: world class designs*. Newnes, 2009.
- [60] Y. Yi, R. Woods, L. Ting, and C. Cowan, "High speed FPGA-Based Implementations of Delayed-LMS filters," *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 39, no. 1, pp. 113–131, Jan 2005. [Online]. Available: <https://doi.org/10.1023/B:VLSI.0000047275.54691.be>
- [61] ADI, "3.5 GSPS direct digital synthesizer with 12-bit DAC: AD9914," *Analog Devices*, 2018.
- [62] M. Maggio, A. Leva, and L. Piroddi, "Finite-precision implementation issues in narrowband active control," in *2007 46th IEEE Conference on Decision and Control*, Dec 2007, pp. 1076–1081.
- [63] B. ISO, "16283-3: 2016 acoustics— field measurement of sound insulation in buildings and of building elements," *Part 3. Facade sound insulation*, 2018.
- [64] B. Lam, S. Elliott, J. Cheer, and W. S. Gan, "Physical limits on the performance of active noise control through open windows," *Applied Acoustics*, vol. 137, pp. 9–17, 2018.
- [65] S. J. Elliott, J. Cheer, L. Bhan, C. Shi, and W. S. Gan, "A wavenumber approach to analysing the active control of plane waves with arrays of secondary sources," *Journal of Sound and Vibration*, vol. 419, pp. 405–419, 2018.

VIII. THE LIST OF DIFFERENCES FROM THE PREVIOUS WORK

This paper is the extension of the article: "Multiple parallel branch with folding architecture for multichannel filtered-x least mean square algorithm [57]", which is published in 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

Compared to the previous work, this article has the following differences:

- 1) The paper optimizes the Multiple-Parallel-Branch with Folding Architecture by using the multiple MAC.
- 2) The architecture in the paper uses a more practical clock divider.
- 3) The paper shows the detail derivation of the recursive iteration of the algorithm.
- 4) The paper shows the details of the design of the folded structure and the implementation of the proposed architecture.
- 5) The paper realizes a $24 \times 24 \times 24$ multichannel FxLMS algorithm based on the proposed architecture.
- 6) The MCANC has been validated in a 24 channel open window.

Dongyuan Shi received his B.ENG and M.S. degrees in the University of Electronic Science and Technology of China (UESTC), in 2010 and 2013, respectively. He worked as a research fellow of the Applied Micro-fluidicology Laboratory of Nanyang Technological University to design the cell CMOS image capture system and the framework of the image processing algorithm from 2013-2014. He is currently pursuing a Ph.D. degree in electrical and electronic engineering at Nanyang Technological University (NTU), Singapore. His research interests

include advanced digital signal processing, adaptive array processing, high-speed real-time digital system implementation, and active noise control.

Woon-Seng Gan received his BEng (1st Class Hons) and PhD degrees, both in Electrical and Electronic Engineering from the University of Strathclyde, UK in 1989 and 1993 respectively. He is currently a Professor of Audio Engineering in the School of Electrical and Electronic Engineering in Nanyang Technological University. He also served as the Head of the Information Engineering Division in the School of Electrical and Electronic Engineering in Nanyang Technological University from 2011-2014. He is currently the Director, Centre for Infocomm Technology @ NTU, which was formed in 2011 and brings together

expert in the area of Information and Communication Engineering.

His research has been concerned with the connections between the physical world, signal processing and sound control, which resulted in the practical demonstration and licensing of spatial audio algorithms, directional sound beam, and active noise control for headphones. He has published more than 330 international refereed journal and conference papers, and has translated his research into 6 granted US/Singapore patents. He had co-authored three books on Subband Adaptive Filtering: Theory and Implementation (John Wiley, 2009); Embedded Signal Processing with the Micro Signal Architecture, (Wiley-IEEE, 2007); and Digital Signal Processors: Architectures, Implementations, and Applications (Prentice Hall, 2005). In 2017, he won the APSIPA Sadaoki Furui Prize Paper Award. He is a Fellow of the Audio Engineering Society(AES), a Fellow of the Institute of Engineering and Technology(IET), and a Senior Member of the IEEE. He also served as an Associate Editor of the IEEE/ACM Transaction on Audio, Speech, and Language Processing (TASLP; 2012-15); IEEE Signal Processing Letters (SPL; 2016-2019); and Associate Technical Editor of the Journal of Audio Engineering Society (JAES; 2013-).

Jianjun He received his B.ENG. degree in automation from Nanjing University of Posts and Telecommunications, China in 2011 and his Ph.D. degree in electrical and electronic engineering at Nanyang Technological University (NTU), Singapore in 2016. In 2011, he was working in Nanjing International Center of Entrepreneurs (NICE), building platforms for start-ups from overseas Chinese scholars. From 2015 to 2016, he was a research fellow with School of Electrical and Electronic Engineering in NTU, Singapore, leading the research projects in 3D audio

and active noise control. Since July 2016, he has been a senior member of technical staff, software with Maxim Integrated, developing disruptive audio solutions for consumer audio devices. His research work has resulted in more than 40 international refereed publications, including 1 book in Springer, several top-tier journals in IEEE Signal Processing Magazine, IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP), etc. He was invited as a tutorial speaker at APSIPA 2015 and ICASSP 2018. He has been an active reviewer for various journals and conferences on acoustics, audio, and signal processing. Aiming at improving humans' listening experience, his research interests include audio and acoustic signal processing, 3D/spatial audio, psychoacoustics, active noise control, and emerging audio and speech applications (wearables, virtual/augmented reality). Currently, He is a member of the IEEE, IEEE Signal Processing Society (SPS), APSIPA, AES. He served at the IEEE SPS Singapore Chapter technical committee, APSIPA Speech Language Audio technical committee, as well as being an affiliate member of IEEE SPS audio and acoustic technical committee.

Lam Bhan received his B.Eng. (Hons) and Ph.D. degrees in electrical and electronic engineering from Nanyang Technological Univ., Singapore, in 2013 and 2019, respectively. He is currently a research fellow at the School of Electrical and Electronic Engineering, Nanyang Technological Univ., Singapore. His current research includes the acoustics of active noise control, soundscape, and signal processing for active control