MULTIPLE PARALLEL BRANCH WITH FOLDING ARCHITECTURE FOR MULTICHANNEL FILTERED-X LEAST MEAN SQUARE ALGORITHM

Dongyuan Shi, Jianjun He, Chuang Shi, Tatsuya Murao, and Woon-Seng Gan

Digital Signal Processing Lab, School of Electrical & Electronic Engineering,
Nanyang Technological University, Singapore

ABSTRACT

Multichannel active noise control (MCANC) systems are commonly used in acoustic noise or vibration control, such as large-dimension ventilation ducts, open windows and mechanical structures. However, its computational load far exceeds the capabilities of digital signal processors (DSPs) and microcontrollers. Even the field programmable gate array (FPGA) cannot straightforwardly cope with the exponential increase in the computation load of MCANC systems. A novel architecture, called the multiple parallel branch with folding, is proposed for the $J \times J \times M$ MCANC implementation with the floating-point arithmetic. This architecture addresses the tradeoff between throughput and hardware resource consumption by using parallel execution and folding. The proposed architecture is validated in an experimental setup that carries out a $4 \times 4 \times 4$ multichannel filtered-x least mean square (FxLMS) algorithm achieving the sampling rate and throughput of 24 KHz and 18.4 Mbps, respectively.

Index Terms— Filtered-x least mean square algorithm, multichannel active noise control, field programmable gate array.

1. INTRODUCTION

Long-time exposure to high decibel noise causes serious health issues, such as hearing impairment, hypertension, ischemic heart disease, annoyance, and sleep disturbance [1]. To solve this problem, passive techniques such as enclosures and shelter are deployed, but their cost is usually high and may affect the natural lighting and ventilation. Active noise control (ANC) [2, 3] is an alternative technique to tackle noise in ducts, headsets [4-6], infant incubators, and windows [7-9]. The ANC system employs an adaptive method to generate destructive interfere using loudspeakers to reduce the noise.

The digital signal processors (DSPs) equipped with fast multiplier-accumulators (MACs) play a dominating role in the implementation of single channel ANC [10, 11]. Other low-cost hardware platforms, such as microcontroller [12] and ARM, are also used, but they are limited in simple ANC systems with short tap length and low sampling rate. The exponentially increasing computation cost of multichannel ANC (MCANC) systems hinders the regular platforms to achieve a sufficiently high sampling rate that matches the distance between the reference microphone and secondary source. For this reason, the field programmable gate array (FPGA) becomes more attractive due to its flexibility, wide bandwidth, hardware resources, and parallel operation. However, programming and debugging on FPGA are complicated and tedious, since the algorithm architecture needs to be customized for the algorithm.

Due to the low complexity and good performance, the ANC system typically employs filtered-x least mean square (FxLMS) algorithm. To implement the FxLMS algorithm on FPGA, additional delay was previously brought into FxLMS to enable its pipelining feature. This practice was known as the delayed FxLMS (DFxLMS). Its derivative structure, the delayed block FxLMS (DBFxLMS) [14], made use of the parallel operation more efficiently. To reduce the usage of multiplications, a fast-FsLMS [15] was proposed for ANC application. However, these modified algorithms reduced the computation complexity, improved the throughput, but degraded the noise reduction performance. To balance of the performance between the throughput and hardware resource consumption of the FxLMS algorithm, a multiple parallel branch with folding architecture is proposed, which uses only J parallel MACs to realize a $J \times J \times M$ multichannel ANC. In this structure, each MAC works out $2(M + 2)L$ operations individually, where L denotes the tap length of the control filter and estimated secondary path. The parallel structure increases the throughput, meanwhile the folding structure minimizes the hardware resource consumption on FPGA. What's more, the proposed architecture modifies the implementation structure without affecting the performance of the FxLMS algorithm.

2. PROBLME STATMENT

As shown in Fig. 1, we consider J reference microphones, J secondary sources, and M error microphones in a $J \times J \times M$ multichannel ANC system with J control filters, where the jth reference signal is only used to update the jth control filter to generate the jth secondary source output. The primary noise can be attenuated by minimizing the sum of the square of residual noises measured by M error

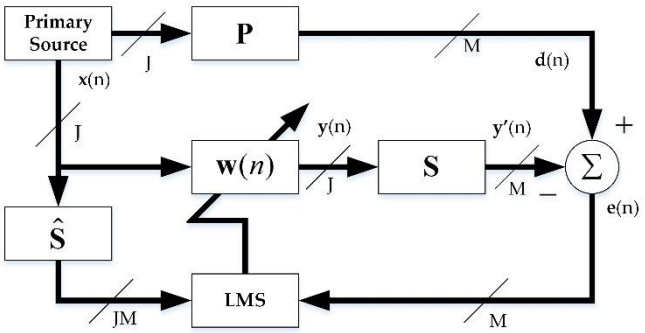

Fig. 1 Block diagram of multichannel FxLMS algorithm.

microphones. The error signal vector $\boldsymbol{e}(\mathrm{n})$ is formed by M error microphones. The matrix $\hat{\mathbf{S}}$ represents the estimates of $M \times J$ secondary paths from J secondary sources to M error microphones. The weight vector of the jth control filter is written as $\mathbf{w}_j(n) = \begin{bmatrix} w_{j,0}(n) & w_{j,1} & \cdots & w_{j,L-1} \end{bmatrix}^T$, where T denotes the transpose operation.

The J control filters can be combined into one long vector

$$\mathbf{w}(n) = [\mathbf{w}_1^T(n) \quad \mathbf{w}_2^T(n) \quad \cdots \quad \mathbf{w}_J^T(n)]^T . \tag{1}$$

The $JL \times J$ reference signal matrix is formed by

$$\mathbf{x}(n) = \begin{bmatrix} \mathbf{x}_1(n) & 0 & 0 & 0 \\ 0 & \mathbf{x}_2(n) & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \mathbf{x}_J(n) \end{bmatrix} \tag{2}$$

where the jth reference signal is

$$\mathbf{x}_j = \begin{bmatrix} x_j(n) & x_j(n-1) & \cdots & x_j(n-L+1) \end{bmatrix}^T . \tag{3}$$

The output signal vector of J secondary sources can be described as

$$\mathbf{y}(n) = \mathbf{x}(n)^T \mathbf{w}(n) . \tag{4}$$

In the multichannel FxLMS algorithm [16], the weight update equation of the jth control filter is

$$\mathbf{w}_j(n+1) = \mathbf{w}_j(n) + \mu \sum_{m=1}^{M} \mathbf{x}'_{jm}(n) e_m(n) , \tag{5}$$

where μ is the step size; $x'_{jm}(n) = \hat{s}_{mj}(n) * x_j(n)$; $\hat{s}_{jm}(n)$ is the estimate of secondary paths from the jth secondary source to the mth microphone; $x_j(n)$ is the reference signal from the jth reference microphone; and $e_m(n)$ denotes the mth error signal.

It is shown in (4) and (5) that each iteration of $J \times J \times M$ multichannel FxLMS algorithm costs $L \times J \times 2(M+1)$ multiplications and additions. For example, a $4 \times 4 \times 4$ multichannel ANC based on FxLMS algorithm with a sampling rate of 24 KHz and filter length $L = 200$ takes 192 million multiplications and additions every second. In comparison, the popular DSP, TMS320C55x, can only handle 200 million MACs per second. Therefore, FPGA is more suitable to realize multichannel ANC, since its

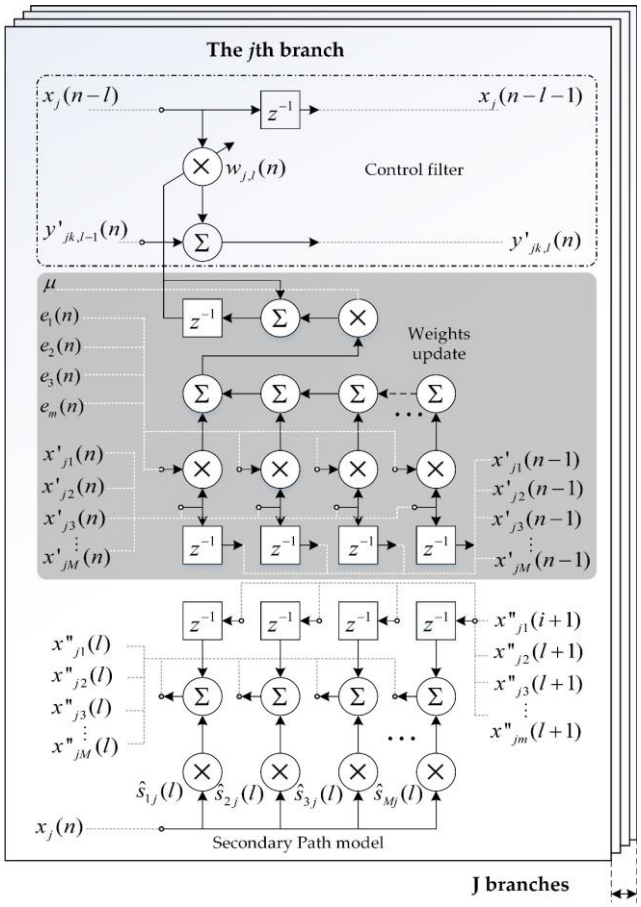

Fig. 2 Multichannel FxLMS based on J branches.

parallelism operation capability potentially allows all the arithmetic computations to be completed at the same time. However, without appropriate optimization, FPGA would consume a large number of hardware resources to finish one computation of the multichannel FxLMS algorithm, which is impractical considering that the number of adders and multipliers is still limited in the state-of-the-art FPGAs.

3. PROPOSED MULTICHANNEL FXLMS ANC ARCHITECTURE DESIGN

The challenge of implementing multichannel FxLMS in FPGA is to achieve both high throughput rate and low hardware resource consumption. Hence, we propose to split the $J \times J \times M$ multichannel FxLMS algorithm into J identical branches, where each branch includes a control filter, a weight update equation, and $J \times M$ secondary path estimates. Each branch runs simultaneously so that we can achieve faster processing speed and higher sampling rate. Inside each branch, one computation cycle is scheduled into several recursive iterations that reuse the same hardware resource. This technique, called the folding transformation, significantly reduces the hardware resource usage from multiple operations to a single functional unit. Therefore, by

integrating parallelization and folding transformation, the proposed architecture guarantees an acceptable throughput rate with minimal hardware resource consumption on FPGA.

3.1 Recursive Equation of Parallel Branch

To get the recursive equation of each parallel branch in $J \times J \times M$ multichannel FxLMS, we firstly split the branch into L recursive iterations. Hence, we rewrite the jth secondary source output from (4) using recursive formula as

$$Y_j(z) = Y'_{j,L}(z) = Y'_{j,L-1}(z) + W_{j,L-1}(z)X(z)z^{-L+1}, \quad (6)$$

where $Y'_{j,l}(z) = \sum_{i=0}^{l-1} W_{j,i}(z)X(z)z^{-i}$; and $W_{j,i}(z)$ is the ith weight of the jth control filter. The lth element in (6) can be described in z transformation form as

$$W_{j,l}(z) = \left(W_{j,l}(z) + \mu \sum_{m=1}^{M} X'_{jm}(z) z^{-l} E_m(z) \right) z^{-1}, \quad (7)$$

where $l \in [0, L-1]$. The filtered reference signal $x'_{jm}(n)$ can be rewritten as

$$\begin{aligned} X'_{jm}(z) = X''_{jm}(0) &= \sum_{l=0}^{L-1} \hat{s}_{jm}(l) X(z) z^{-l+0} \\ &= \hat{s}_{jm}(0) X(z) z^{-0} + \sum_{l=1}^{L-1} \hat{s}_{jm}(l) X(z) z^{-l+1} z^{-1} \\ &= \hat{s}_{jm}(0) X(z) z^{-0} + X''_{jm}(1) z^{-1} \end{aligned} \quad (8)$$

where $X''_{jm}(k) = \sum_{l=k}^{L-1} \hat{s}_{jm}(k) X(z) z^{-l+k}$ and $k \in [0, L-1]$. Combining (6), (7) and (8) into one recursive iteration of the jth branch leads to the proposed $J \times J \times M$ multichannel FxLMS algorithm, whose block diagram is shown in Fig. 2. In Fig. 2, there are $2(M+1)$ multipliers and adders in each branch. In one computing cycle of the whole $J \times J \times M$ multichannel FxLMS algorithm, J branches will use $J \times 2(M+1)$ multipliers and adders in total to process L recursive iterations. Compared to the original multichannel FxLMS implementation, hardware resource consumption of the proposed architecture is reduced by L times at this stage.

3.2 Folding Structure

The hardware resources in FPGA are generally categorized into interconnection, memory, logic cell, and DSP slice. Among them, DSP slice is of the most scarcity because its complexity structure costs a lot of silicon area in FPGA. Therefore, the most effective way to reduce the hardware resource consumption is to minimize the number of adders and multipliers. In the previous section, we found that the number of adders and multipliers in the new architecture of $J \times J \times M$ multichannel FxLMS is $J \times 2(M+1)$. To further reduce the resource usage, which increases dramatically with the escalation of J and the number of channels in FxLMS, we propose to use folding transformation.

After the folding transformation, the implementation of the $J \times J \times M$ multichannel FxLMS is illustrated in Fig. 3,

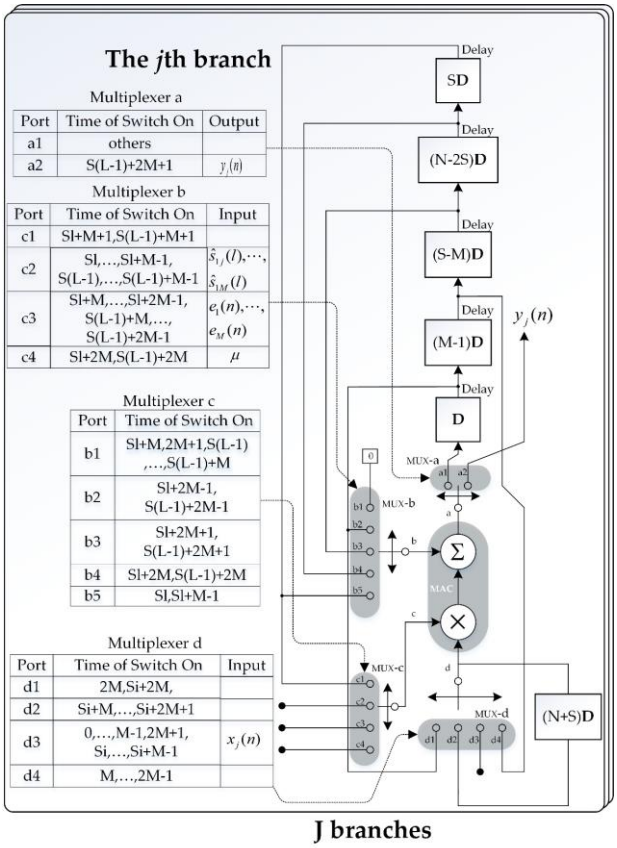

Fig. 3 Folding $J \times J \times M$ multichannel FxLMS.

where $S = 2(M+1)$; l and i range from 0 to L-1 and from 1 to L, respectively. This folding structure is composed by three main components: the arithmetic unit (one adder and one multiplier), the multiplexer, and the D flip-flop. The multiplexer selects the input and output signal for the arithmetic unit based on the time schedule described in tables embedded in Fig. 3. The unit in those tables is clock tick, which begins from zero. The D flip-flop realizes the delay and temporal storage of data. Overall, the folding structure only requires J adders and multipliers to finish the $J \times J \times M$ multichannel FxLMS. The hardware resource usage is reduced by $L \times 2(M+1)$ times. As a tradeoff, the duration to finish one computation of multichannel FxLMS is $L \times 2(M+1)$ times more than the original duration. However, due to the high running speed of FPGA, the increased duration is affordable to achieve a high throughput rate.

4. FPGA IMPLEMENTATION

To verify the proposed architecture, we realized a $4 \times 4 \times 4$ floating point multichannel FxLMS on NI 9039 Compact-RIO with NI 9220 ADC and NI 9264 DAC modules. The main processor is Xilinx Kintex-7 7K235T FPGA, which consists of 407,600 flip-flops, 203,800 LUTs, 840 DSP slice and 16,020 Kbits RAM. DSP48E is chosen to build the

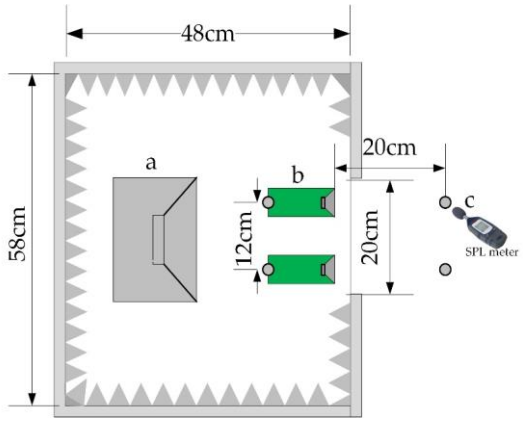

Figure 4. Top view of small chamber for $\mathbf{4 \times 4 \times 4}$ ANC

floating point arithmetic unit of the architecture, where each DSP48E has 28x15 two-complement multiplier and a 48–bit accumulator. Four DSP48Es can be combined into a single precision floating point MAC. To realize the delay and temporal storage of data, we replace D flip-flop with Bock RAM, because the magnitude of RAM on FPGA is much larger than flip-flops.

Setting the lengths of the secondary path estimates and control filters to 200 taps, the device utilization report shows that the 4-channel FxLMS algorithm costs 16 DSP48E (1.9% of total DSP48Es), 30,385 registers (7.6 % of total registers), 27,833 LUTs (13.7 % of total LUTs), 24 block RAM (5.4 % of total RAM). Since the 4-channel FxLMS algorithm takes 2000 iterations to finish one computation, the maximum achievable sampling rate is thus 25 KHz when the FPGA running clock is 50 MHz. Therefore, we set the system sampling rate at 24 KHz and the cut off frequency of anti-aliasing and reconstruction filters at 11.5 KHz.

5. EXPERIMENTS OF $4 \times 4 \times 4$ ANC

To figure out the performance of this new architecture in multichannel ANC, we conducted experiments on a small chamber and measured the noise reduction performance. The top view of this small chamber is shown in Fig. 4, where *a* denotes the primary noise loudspeaker; *b* presents the 4 ANC units, each composed of a reference microphone and a noise cancelling loudspeaker, and *c* stands for the error microphones. A sound pressure level (SPL) meter was placed near the error microphones.

In the first experiment, we used 1 KHz sine tone as the primary noise. The SPL level from the SPL meter dropped from 83.5 dB to 53.7 dB, after the ANC system was turned on. The noise reduction was 29.8 dB. Fig. 5 (a) shows the power spectrum of noise tested from one of the 4 error microphones. Meanwhile, the other 3 error microphones showed a similar pattern. We can observe that the 1 KHz tonal noise is eliminated at the error microphone positions.

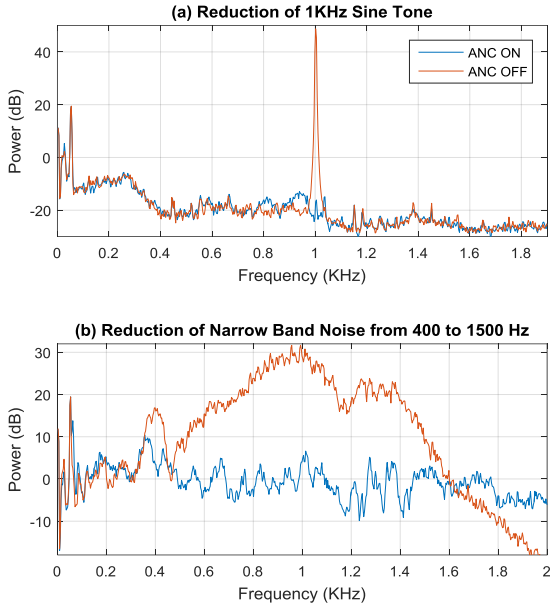

Fig. 5 Power spectrum of error signal.

In the second experiment, we used a band-limited white noise as the primary noise. The noise frequency ranged from 400Hz to 1500Hz. Readings from the SPL meter indicated that the noise level dropped from 93.3 dB to 75.6 dB with ANC being turned on. The noise reduction was 17.7dB. The power spectrum of the error microphone is shown in Fig. 5 (b), which clearly indicates the effectiveness of the ANC system. These experiment results preliminarily validate the practical performance of the proposed multiple parallel branch with folding architecture for multichannel FxLMS algorithm in ANC.

6. CONCLUSIONS

This paper proposed a multiple parallel branch with folding floating point architecture for multichannel FxLMS to implement multichannel ANC based on FPGA. For a $J \times J \times M$ multichannel FxLMS algorithm, the proposed new architecture requires only J MACs, which yields a reduction of $L \times 2(M + 1)$ times in terms of the hardware resources and achieves a high throughput. A $4 \times 4 \times 4$ multichannel FxLMS algorithm was implemented on Xilinx FPGA using the proposed architecture. The noise reduction performance at error microphones clearly validate the effectiveness of the proposed multiple parallel branch with folding architecture.

ACKNOWLEGEMENT

This material is based on research/work supported by Singapore Ministry of National Development and National Research Foundation under L2 NIC Award NO.L2NICCFP1-2013-7.

REFERENCES

[1] T. Hellmuth, T. Classen, R. Kim, and S. Kephalopoulos, "Methodological guidance for estimating the burden of disease from environmental noise," *WHO Regional Office for Europe*, 2012.

[2] S. Elliott, *Signal processing for active control*: Academic press, 2000.

[3] S. M. Kuo and D. R. Morgan, "Active noise control: a tutorial review," *Proceedings of the IEEE*, vol. 87, pp. 943-973, 1999.

[4] K. K. Parhi, *VLSI digital signal processing systems: design and implementation*: John Wiley & Sons, 2007.

[5] W. S. Gan, S. Mitra, and S. M. Kuo, "Adaptive feedback active noise control headset: implementation, evaluation and its extensions," *IEEE Transactions on Consumer Electronics*, vol. 51, pp. 975-982, 2005.

[6] S. M. Kuo, S. Mitra, and W.-S. Gan, "Active noise control system for headphone applications," *IEEE Transactions on Control Systems Technology*, vol. 14, pp. 331-335, 2006.

[7] T. Murao and M. Nishimura, "Basic study on active acoustic shielding," *Journal of Environment and Engineering*, vol. 7, pp. 76-91, 2012.

[8] T. Murao, M. Nishimura, K. Sakurama, and S. Nishida, "Basic study on active acoustic shielding (Improving noise-reducing performance in low-frequency range)," *Mechanical Engineering Journal*, vol. 1, pp. EPS0065-EPS0065, 2014.

[9] T. Murao, M. Nishimura, K. Sakurama, and S. Nishida, "Basic study on active acoustic shielding (Improving the method to enlarge the AAS window)," *Mechanical Engineering Journal*, vol. 3, pp. 15-00322-15-00322, 2016.

[10] 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.

[11] 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.

[12] C. Y. Chang and S. T. Li, "Active noise control in headsets by using a low-cost microcontroller," *IEEE Transactions on industrial electronics*, vol. 58, pp. 1936-1942, 2011.

[13] V. R. Ramachandran, I. M. 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*, 2008, pp. 1517-1520.

[14] 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*, pp. 1-27, 2016.

[15] E. P. Reddy, D. P. Das, and K. Prabhu, "Fast exact multichannel FSLMS algorithm for active noise control," *Signal Processing*, vol. 89, pp. 952-956, 2009.

[16] S. M. Kuo and D. Morgan, *Active noise control systems: algorithms and DSP implementations*: John Wiley & Sons, Inc., 1995.