Design of Active Noise Control System Based on FPGA

Jun Yuan, Jun Li a,*, Qiang Zhao, Xiangsheng Meng
Department of Optoelectronic Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China
a1093878959@qq.com

Abstract

In order to improve the convergence speed of the active noise control (ANC) system in practice. This paper presents a design method for ANC system implementation for online auxiliary path modeling. The proposed momentum LMS algorithm (MLMS) is used to adjust the modeling filter and control the filter. The design includes three convolution modules and two weight update modules to reduce noise. When it is necessary to better reduce the noise of the ANC system, this innovative modular implementation of the ANC system can achieve rapid design and rapid convergence, and has the ability to expand four adaptive filters and data buses. The system showed convincing results with accuracy, convergence speed, and the functions of the designed and implemented system. At the same time, using FPGA's low power consumption and high-speed parallel processing functions to design the system, the final modeling error and noise reduction performance are -25dB and 15dB, respectively.

Keywords
Active noise control; FPGA; MLMS.

1. Introduction

The traditional approach to attenuate the transportation noise is to use passive techniques such as barriers. These passive silencers are valued for their high attenuation over a broad frequency range; however, they are relatively large, costly, and ineffective at low frequencies [1]. Meanwhile, the Soundproof window cannot be closed for silencers to attenuate the undesired noise acoustic noise control uses passive techniques such as enclosures, barriers, and silencers to attenuate the undesired noise [1], [2]. Mechanical vibration is another related type of noise that commonly creates problems in all areas of transportation and manufacturing, as well as with many household appliances. In the hot summer or in the tropical hot area, cannot be closed for a long time for ventilation, which affects the indoor air quality and cannot reduce the indoor temperature, resulting in the reduction of indoor living comfort. Therefore, it is an urgent need to study the effective application of active noise control (ANC) in high-rise buildings under sufficient ventilation conditions [4].

Acoustic noise pollution has gradually become an urgent problem in large and medium-sized cities, especially near the vital traffic road. Long-term exposure to such noise environment will cause serious harm to people's physiology and psychology, while general noise interference will affect people's normal work and life [1-3]. Research by Mathias Basner and Wolfgang Babisch shows that reducing the exposure time of daily life in a noisy environment will ultimately reduce people's distress and keep children away from the noisy environment, which will improve students' learning efficiency. Reducing noise exposure time can improve sleep, thereby reducing the prevalence of cardiovascular disease. The hospital found that long-term exposure to the noise environment is more severe, and the patient's hospital stay is usually longer [4].
Active noise control system [5] are frequently employed to reduce undesired noise sound in headsets, industrial ducts, automobiles, etc. Compared with other passive methods, ANC offers more flexibility in controlling low-frequency noises. One of the most widely used adaptive control algorithm for ANC systems is the filtered-x LMS (FxLMS) algorithm [6-8]. An important issue in FxLMS algorithm is the modeling of the secondary path. The well-known structure in [9] with auxiliary noise injection is commonly employed, where the secondary path is identified online during the operation of the controller.

In the hardware design of active noise control using DSP, due to the need to execute the inherent software program in the early stage, the operating efficiency of the system will be reduced [10]. On the contrary, if the active noise control system is designed with FPGA [11-13], it will be more efficient than the DSP method because the design is directly hardware oriented. This paper introduces an implementation method of FPGA for online secondary path modeling of active noise control system.

The organization of this paper is as follows. Section 2 gives a brief overview of the existing methods and proposes new method for ANC systems with online secondary path modeling. Section 3 details the proposed modular ANC system design method. Section 4 presents results of computer simulations, and Section 5 concludes the paper.

2. ANC Systems with Online Secondary Path Modeling

2.1. ANC System Model

The ANC system is a method of cancelling noise in the main channel by generating signals of equal amplitude and opposite phase according to the superposition principle of sound waves. As shown in Fig.1, the filtered-x LMS (FxLMS) algorithm is used the ANC system [6-8]. The $P(z)$ is the transfer function of the primary path between the reference microphone and the error microphone, and $S(z)$ is the transfer function of the secondary path between the loudspeaker and the microphone. $\hat{S}(z)$ is the estimate of $S(z)$. In practical applications, the transfer function of $S(z)$ is usually time-varying. The $x(n)$ is the reference noise signal at reference microphone. $e(n)$ is error signal obtained by $e(n) = d(n) - y'(n)$, where $d(n) = p(n) * x(n)$; $y'(n) = s(n) * y(n) + y(n) = w(n) * x(n)$. Weights, $w(n)$, of adaptive filter are updated by using

$$w(n+1) = w(n) + u_w e(n)x'(n)$$

Where the step size of the $w(z)$ filter is $\mu$;

$$x'(n) = \hat{s}(n) * x(n)$$

![Fig.1 Block diagram of FxLMS algorithm in ANC system](image)

In order to solve the problem of secondary path time-varying, Eriksson first proposed the online secondary path estimation model (OSPM) [9]. The additive random noise technique is used in this model. To improve the performance of Eriksson’s method, the methods proposed in [14] add another third adaptive filter. This third adaptive filter works on the principle of adaptive noise cancelation (ADNC). In order to reduce the computational complexity, Akhtar
proposed a method of secondary path modeling based on variable step size LMS algorithm [15]. Akhtar’s method reduces the computational complexity and improves the convergence of the secondary path (Fig.2).

However, the variable step size algorithm requires a large number of floating-point operations in the actual FPGA implementation, which is not easy to implement in hardware. In this article, we modified Akhtar’s method to reduce hardware consumption. This article proposes a momentum LMS (MLMS) algorithm (Figure 3) to make the hardware easier to implement.

Consider Akhtar’s method shown in Fig.2. The residual error signal $e(n)$ of this algorithm is expressed as:

$$e(n) = d(n) - y'(n) + v'(n)$$  \(\text{(3)}\)

Where $v(n)$ is an interally generated white Gaussian noise, which is injected at the output of the control filter $W(z)$. In this figure $\hat{S}(z)$ is the modelling FIR filter with length M that generates $\hat{v}'(n)$ expressed below:

$$\hat{v}'(n) = \hat{s}^T(n)v_M(n)$$  \(\text{(4)}\)

As the figure shows, $\hat{v}'(n)$ generates the error signal for both the modelling filter $\hat{S}(z)$ and the control filter $W(z)$ subtracting from $e(n)$:

$$f(n) = [d(n) - y'(n) + v'(n)] - \hat{v}'(n)$$  \(\text{(5)}\)

Coefficients of the modelling filter $\hat{S}(z)$ are updated as follows:

$$\hat{s}(n+1) = \hat{s}(n) + \mu_s(n)f(n)v(n)$$  \(\text{(6)}\)

Where $\mu_s(n)$ is the step-size parameter of the VSS-LMS algorithm which will be explained later. Finally coefficients of the control filter $W(z)$ are updated as below:

$$w(n+1) = w(n) + \mu_w(n)f(n)\hat{x}'(n)$$  \(\text{(7)}\)

The input to the LMS algorithm is derived by filtering the reference signal through $\hat{S}(z)$:

$$\hat{x}'(n) = \hat{s}^T(n)x_M(n)$$  \(\text{(8)}\)

Where $x_M(n) = [x(n), x(n-1), ..., x(n-M + 1)]^T$ is an M sample reference signal.

The VSS-LMS algorithm is used to update modelling filter $\hat{S}(z)$ coefficients. For more detail on theory of this algorithm reader may refer to [4]. As we mentioned before, the step-size
parameter ($\mu_s(n)$) of VSS-LMS algorithm and this parameter is calculated using the following three steps [4]:

Initially, the power of error signals $e(n)$ and $f(n)$ are computed:

$$P_e(n) = \lambda P_e(n-1) + (1 - \lambda)e^2(n)$$
$$P_f(n) = \lambda P_f(n-1) + (1 - \lambda)f^2(n)$$  \hspace{1cm} (9)

Then, the ratio of the estimated powers is obtained:

$$\rho(n) = \frac{P_f(n)}{P_e(n)}$$
$$\rho(0) \approx 1, \lim_{n \to \infty} \rho(n) \to 0$$  \hspace{1cm} (10)

Finally, the step size is calculated as follows:

$$\mu_s(n) = \rho(n)\mu_{s_{\text{min}}} + (1 - \rho(n))\mu_{s_{\text{max}}}$$  \hspace{1cm} (11)

Where $\mu_{s_{\text{min}}}$, $\mu_{s_{\text{max}}}$ and $\lambda$ are experimentally determined. Using VSS-LMS algorithm increases the modelling accuracy and correspondingly improves system performance. Indeed, Akhtar’s method completely provided these features.

### 2.2. Simulation Results

Here we suggest modification to Akhtar’s method (Fig.3.), so that the amount of computation in the hardware implementation process can be reduced. This method uses the momentum LMS algorithm to adjust the modeling filter $T(z)$ and the control filter $W(z)$. This modified-Akhtar method is shown in Fig.3.

The momentum LMS algorithm adds a momentum term introduced by the weight coefficient correlation to the basic LMS algorithm.

$$W(n+1) = W(n) - 2\mu e(n)x'(n) + \alpha[W(n) - W(n-1)]$$  \hspace{1cm} (12)

Where: $\alpha$ is momentum factor, $|\alpha| < 1$.

This method, hereafter called MLMS based method. Compared with the VSS-LMS algorithm, the computational complexity is reduced under the premise of accelerating the convergence speed. In order to verify the convergence rate and modeling accuracy of the algorithm for the ANC system, we use the relative modeling error as defined below.

$$\Delta S(dB) = 10\log_{10}\left(\frac{\sum_{i=0}^{M-1}|s_i(n) - \hat{s}_i(n)|^2}{\sum_{i=0}^{M-1}|s_i(n)|^2}\right)$$  \hspace{1cm} (13)

In order to verify the performance of the system noise reduction, the definition of Equation 11 is used:

$$R = -10\log_{10}\left(\frac{\sum e^2(n)}{\sum d^2(n)}\right)$$  \hspace{1cm} (14)

The larger the positive value of $R$ indicates the noise cancellation effect of ANC system is better.
Fig. 7. Performance comparison between proposed method and the other existing methods

3. A Summary of Modular Design of an ANC System

3.1. FPGA Implementation of the proposed momentum LMS based ANC system

As presented in section 2, the general architecture of an ANC system which is shown in Fig. 4 is implemented using the C and W modules which are shown in Fig. 5 and Fig. 6. The entire ANC system contains a total of three inputs and one output. The \( x(n) \), is the reference noise signal received by reference microphone, and \( e(n) \) is the error noise signal received by error microphone. The objective of ANC system is to generate anti-noise signals \( y(n) \) of the same phase as the amplitude of the noise signal.

The modular design of the ANC system consists of 5 modules, 3 convolution modules, and 2 MLMS algorithm tap-weights updating modules. As shown in figure 4, the convolution module C1 outputs \( \hat{s}(n) \ast v(n) = \hat{v}'(n) \). The error signals \( e(n) \) and \( \hat{v}'(n) \) are subtracted to obtain \( f(n) \). The module W1 outputs the weight coefficient of the modeling filter \( \hat{S}(z) \). The convolution module C2 outputs \( \hat{s}(n) \ast x(n) = \tilde{x}'(n) \). The module W2 outputs the weight coefficient of the control filter \( W(z) \). The convolution module C3 outputs \( x(n) \ast w(n) = y(n) \). The secondary signals \( y(n) \) and \( v(n) \) are subtracted to obtain Anti-noise signal \( y(n) - v(n) \).

There are two input signals \( \hat{s}(n) \) and \( v(n) \) in the C1 module. The output signal of C1 module is to convolve the two input signals, as indicated in Eq.7.

\[
v_n(n) = \sum_{j=0}^{N-1} S_n(j)v_n(N - j)
\]  

There are two input signals \( v(n) \) and \( f(n) \) in the W1 module. The tap-weights of the secondary path modeling filter \( \hat{S}(n) \), as indicated in Eq.8.

\[
S_n(n + 1) = S_n(n) + u_s f_n(n)v_n(n) + \alpha(S_n(n) - S_{n-1}(n))
\]  

There are two input signals \( x(n) \) and \( S(n) \) in the C2 module. The output signal of C2 module is to convolve the two input signals, as indicated in Eq.9.
\[ x'_n(n) = \sum_{j=0}^{N-1} S_n(j)x_n(N-j) \]  \hspace{1cm} (17)

There are two input signals \( x'(n) \) and \( f(n) \) in the W1 module. The tap-weights of the secondary path modeling filter \( w(n) \), as indicated in Eq.10.

\[
w_n(n+1) = w_n(n) + u_wf_n(n)x'_n(n) + \alpha(w_n(n) - w_{n-1}(n))
\]  \hspace{1cm} (18)

There are two input signals \( x(n) \) and \( w(n) \) in the C3 module. The output signal of C3 module is to convolve the two input signals, as indicated in Eq.11.

\[
y_n(n) = \sum_{j=0}^{N-1} w_n(j)x_n(N-j)
\]  \hspace{1cm} (19)

3.2. Implementation of module C

The design of weight update module is shown as figure 5 below. \( i_n(N) \) represents the input signal, subscript N represents the same time, N represents the length of the input signal, \( e_n \) represents the input error signal, \( w_n(L) \) represents the weight coefficient, and L represents the filter order. Error signal \( e_n \), \( u_e n(n) \) is obtained by a shift (n) and a parallel multiplication \( \Delta w_n(n) = u_e n(n)i_n(n) \), and then through a parallel addition operation to get the updated weight coefficient.

\[
w_n(n+1) = w_n(n) + \Delta w_n(n)
\]  \hspace{1cm} (20)

\[
w_n(n+1) = w_n(n) + u_we_n(n)i_n(n) + \alpha(w_n(n) - w_{n-1}(n))
\]  \hspace{1cm} (21)

3.3. Block diagram of the module W

As shown in Fig.7, the convolution module contains a shift register, a divider, N multiplies, and an accumulator.
\[ y_n(n) = \sum_{j=0}^{N-1} w_n(j) i_n(N - j) \] 

(22)

The digital \( W(z) \) contains \( N \) tap-weights, \( w_n(j) \), for \( j = 0, \ldots, N - 1 \). \( w_n(j) \) is the value of the tap-weight at time \( n \). The output of this unit at time \( n \) is the convolution of the input vector, \( i_n(n) \) and the tap-weight vector of the filter, \( h(n) \), as shown in Eq.15.

4. Simulation Results

This section presents the simulation experiments performed to verify the effectiveness of the proposed method. This section is divided into three parts, first verifying the effectiveness of the MLMS algorithm based on the Akhtar’s method. Secondly, verify the effectiveness of the proposed modular ANC system design.

Table 1 presents a computational complexity (multiplications per iteration) comparison of the various methods discussed in this paper.

Table 1 computational complexity comparison of the proposed method with existing method

<table>
<thead>
<tr>
<th>Method</th>
<th>Number of multiplications per iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eriksson’s method</td>
<td>( 2L + 3M + 2 )</td>
</tr>
<tr>
<td>Akhtar’s method</td>
<td>( 2L + 3M + 10 )</td>
</tr>
<tr>
<td>Proposed method</td>
<td>( 2L + 3M + 4 )</td>
</tr>
</tbody>
</table>

For confidence of true performance of the implemented ANC system, a comparison between MATLAB simulation and Verilog HDL implementation is done. The corresponding curves of error signals are shown in Fig.9.

![Fig.8 Comparison between the Simulation and Implementation](image)

A summary of implementation results is shown in table 2.

Table 2 The number of elements used on Alinx FPGA, \( \mu_w = 2^{-11}, \mu_s = 2^{-6} \)

<table>
<thead>
<tr>
<th>Logical utilization</th>
<th>Use</th>
<th>Available</th>
<th>Utilization rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>On-chip register</td>
<td>2558</td>
<td>408000</td>
<td>0%</td>
</tr>
<tr>
<td>LUT</td>
<td>22404</td>
<td>204000</td>
<td>10%</td>
</tr>
<tr>
<td>LUT-FF</td>
<td>161</td>
<td>24801</td>
<td>13%</td>
</tr>
<tr>
<td>IOBs</td>
<td>82</td>
<td>600</td>
<td>13%</td>
</tr>
<tr>
<td>BUFG/BUFGCTRLS</td>
<td>1</td>
<td>32</td>
<td>3%</td>
</tr>
</tbody>
</table>
5. Conclusion

This paper has modified the basic Akhtar’s method on the basis of concept of MLMS algorithm. The computation complexity of the proposed method is lower than the Akhtar’s method, and it achieves a higher convergence rate. The ANC system is designed with 3 convolution modules and two weight update modules. The ANC system is simulated using the MATLAB tool. Simulation results demonstrated that the proposed method achieves a higher convergence rate, a more accurate modelling accuracy, and a better noise reduction performance compared with the existing approach.

References