

## The Circuit Design of Audio Adaptive Filter via Model-Based Design

Hua Ge<sup>1</sup>, Yang Nie<sup>1,2</sup>, Lili Jing<sup>1</sup> and Pengyu Zhao<sup>1</sup>

<sup>1</sup>*Department of Physics, Jining Normal University, Inner Mongolia, China*

<sup>2</sup>*Digital Engineering Center, Communication University of China, Beijing, China  
nieyangwork@163.com*

### Abstract

*In this paper, model-based design is used to complete the design of an adaptive filter by Least Mean Square (LMS) algorithm, which implements the recovery process of audio signal disturbed by noise. We can quickly build a system simulation model by model-based design approach, and accomplish efficiently the system test, simulation and implementation. Theoretical analysis and experimental results show that the method of model-based design is not only valuable to the design and implementation of DSP system, but also can significantly improve the design efficiency of the DSP system.*

**Keywords:** *Model-Based Design; LMS; Noise-cancel; DSP system*

### 1. Introduction

Adaptive filter is the signal processing methods and techniques of the filter [1]. An adaptive filter is required when either the fixed specifications are unknown or the specifications cannot be satisfied by time-invariant filters. Strictly speaking an adaptive filter is a nonlinear filter since its characteristics are dependent on the input signal and consequently the homogeneity and additive conditions are not satisfied. Some of the classical applications of adaptive filtering are system identification, channel equalization, signal enhancement, and prediction. The least-mean-square (LMS) is a search algorithm in which a simplification of the gradient vector computation is made possible by appropriately modifying the objective function [2]. The LMS algorithm, as well as others related to it, is widely used in various applications of adaptive filtering due to its computational simplicity. Although LMS algorithm is relatively simple and suitable for hardware implementation, but its structure is still much more complicated compared to the FIR filter.

Currently, DSP algorithms are implemented in hardware by ASIC, DSP and FPGA. FPGA used widely in high-speed, parallel to achieve aspects of digital signal processing algorithms. With the continuous improvement of semiconductor process technology, FPGA is widely used to implement high-performance DSP system. Due to the lack of efficient design methods, many are familiar with C language programming signal processing engineers, which cannot effectively use FPGA. Therefore, an efficient design method is needed to help them complete the hardware design.

Model-Based Design (MBD) is innovating the way for engineers and scientists working. In Model-Based Design, a system model is right on the center of the development process, from requirements development, through design, implementation, and testing [3]. Model-based design gives us a complete product from idea to generate the code development process. In MBD, a system model is right in the center around the development process, from requirement's development, through design, implementation, and testing [4-5]. This workflow is elaborated to create hardware and software partitioning, automatically create hardware and software implementation code, and verify the hardware and software implementations in the context of the complete system. It is

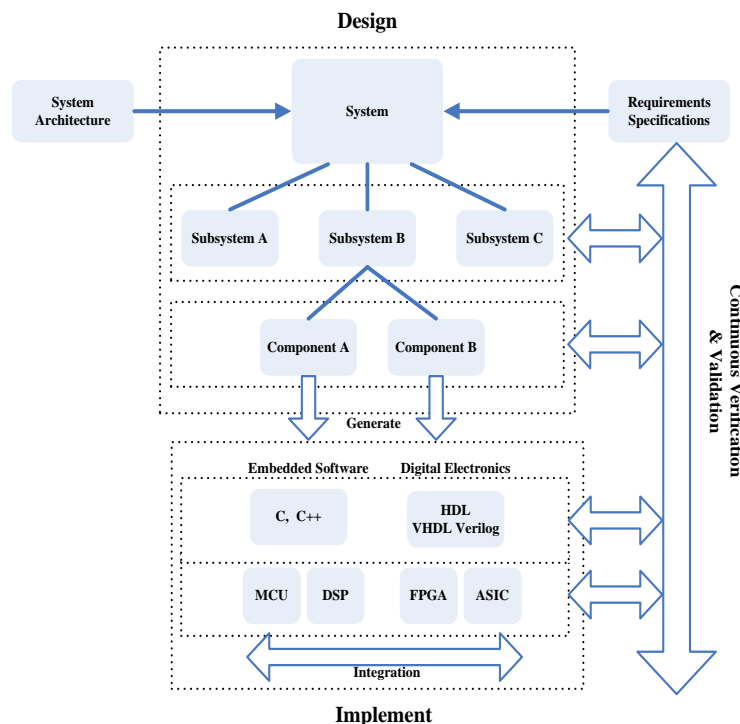
not only overcoming the defects of low efficiency and difficulty of meeting the requirements in traditional methods of progress, but also to meet time-to-market and cost objectives.

The paper is structured as follows: In Section II, we introduce MBD workflow. Section III discusses adaptive algorithm and the LMS algorithm, which are most trustworthy and efficient for adaptive filter hardware implementation. The main contribution of this work is presented in Section IV, where noise-cancel of audio adaptive filter is designed and verified using MBD. Finally, Section V summarizes the main conclusions of this work.

## 2. The Workflow of Model-Based Design

With the continuous improvement of safety and reliability of electronic products, embedded system development presents explosive growth trend. However, the development of the workflow has long remained in the form of initial artificial hand written code, which is increasingly unable to adapt to the design of complex systems. for the hardware development, MBD not only can effectively improve system security and reliability, but also reduce the development time and cost compared with the traditional design method

The system model is the key to the development process in the MBD workflow [6]. MBD workflow is illustrated in Figure 1. Through the establishment of floating point model, the fixed-point model and the system-level model for presenting a complete system design requirements, different professional engineers work effectively, and can communicate at different stages of the development flow. It enforces continuous testing and verification throughout the design process. Using MBD related tools, it can ensure that the design of the various stages of continuous testing, thereby ensuring the correctness of the design



**Figure 1. The Workflow of MBD**

With a graphical environment based on MATLAB/Simulink and a pre-designed block set of DSP cores, the system generator can be used to model a DSP system,

which includes core DSP algorithms and synchronous interfaces to external peripherals and system control unit. System Generator is a MBD tool for the Xilinx FPGA design. Engineers are caught in the DSP friendly Simulink modeling environment using a Xilinx specific block set [7]. The System Generator design can then be imported into a Vivado IDE project using the IP catalog. The tool will automatically generate the hardware description language (HDL) code, which is mapped to the FPGA device and the workflow is shown in Figure 2. Therefore, the designer can define a model representation of the system level design. It is easy to convert the model design into a gate level representation. System Generator provides accelerated simulation through hardware co-simulation. System Generator will automatically create a hardware simulation token for a design caught in the Xilinx DSP block set that will run on supported hardware platforms.

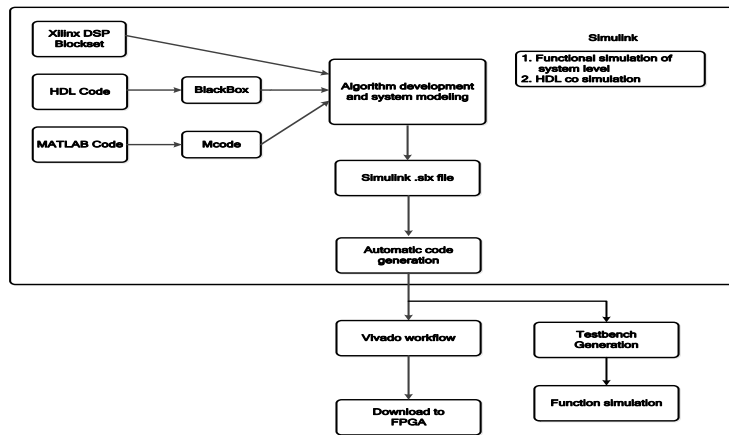


Figure 2. The Workflow of MBD Design Using System Generator

### 3. The Introduction of LMS Algorithm

#### 3.1. The Principle of Adaptive Filter

An adaptive filter is made up of two parts: a digital filter with adjustable parameters and adaptive algorithm. An adaptive digital filter is often represented by a signal flow graph with adaptive nature of weights shown in Figure 3, [8-9]. The aim is to adaptive digital filter such that the input signal  $x(k)$  is filtered to produce  $y(k)$  which when subtracted from desired signal  $d(k)$ , will minimize the power of the error signal  $e(k)$ , where  $w$  is adaptive weights.

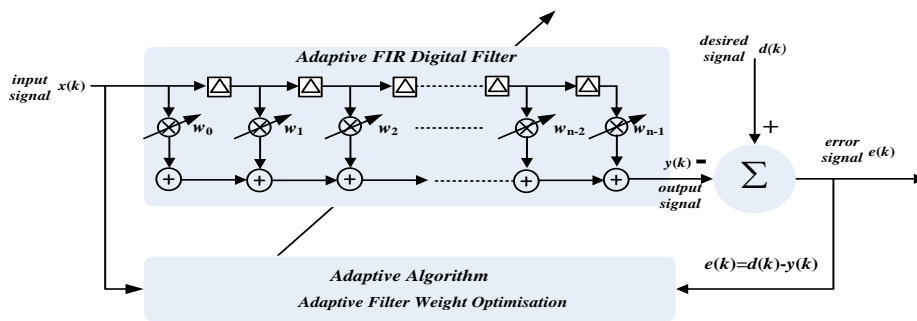


Figure 3. The Basic Structure of Adaptive Filter

When deriving adaptive algorithms it is useful to use a vector notation for the input vector and the weight vector. The output of the filter  $y(k)$ , is the convolution of the weight vector and the input vector [10]:

$$y(k) = w^T x(k) = \sum_{n=0}^{N-1} x(k-n)w_n \quad (1)$$

where  $w = [w_0 \ w_1 \ w_2 \ \dots \ w_{N-2} \ w_{N-1}]^T$

and  $x(k) = [x(k) \ x(k-1) \ x(k-2) \ \dots \ x(k-N+2) \ x(k-N+1)]^T$ .

The aim of the adaptive algorithm is to minimise the error signal power over a period of time. Consider the squared error and take expected (or mean) values

$$\begin{aligned} e^2(k) &= (d(k) - y(k))^2 \\ &= d^2(k) + w^T [x(k)x^T(k)]w - 2d(k)w^T x(k) \end{aligned} \quad (2)$$

$$E[e^2(k)] = E(d^2) + w^T E[xx^T]w - 2w^T E[dx] \quad (3)$$

Writing in terms of the correlation matrix,  $R$  and the cross correlation vector,  $p$ , gives:

$$E[e^2(k)] = E[d^2(k)] + w^T R w - 2w^T p \quad (4)$$

This equation is quadratic in the vector  $w$ . Hence there is only one minimum value of  $\zeta = E[d^2(k)]$ , denoted MMSE (minimum mean square error) and  $w_{opt}$  is the weight vector which minimizes the error signal in the mean squared sense. The MMSE is found from setting the (partial derivative) gradient vector  $\nabla$ , to zero:

$$\begin{aligned} \nabla = \frac{\partial \zeta}{\partial w} &= 2Rw - 2p = 0 \\ \Rightarrow w_{opt} &= R^{-1}p \end{aligned} \quad (5)$$

### 3.2. The LMS Algorithm

In 3.1, we derived the optimal solution for the parameters of the adaptive filter implemented through a linear combiner, which corresponds to the case of multiple input signals [11-12]. This solution leads to the minimum mean-square error in estimating the reference signal  $d(k)$ .

If good estimates of matrix  $R$ , denoted by  $R(k)$ , and of vector  $p$ , denoted by  $P(k)$ , are available, a steepest descent based algorithm can be used to search the Wiener solution of (5) as follows:

$$\begin{aligned} w(k+1) &= w(k) - \mu g_w(k) \\ &= w(k) + 2\mu(P(k) - R(k)w(k)) \end{aligned} \quad (6)$$

for  $k=0, 1, 2, \dots$ , where  $g_w(k)$  represents an estimate of the gradient vector of the objective function with respect to the filter coefficients.

One possible solution is to estimate the gradient vector by employing instantaneous estimates for  $R$  and  $p$  as follows:

$$\begin{aligned} R(k) &= x(k)x^T(k) \\ P(k) &= d(k)x(k) \end{aligned} \quad (7)$$

The resulting gradient estimate is given by

$$\begin{aligned}
 g_w(k) &= -2d(k)x(k) + 2x(k)x^T(k)w(k) \\
 &= 2x(k)(-d(k) + x^T(k)w(k)) \\
 &= -2e(k)x(k)
 \end{aligned} \tag{8}$$

Note that if the objective function is replaced by the instantaneous square error  $e^2(k)$ , instead of the MSE, the above gradient estimate represents the true gradient vector since

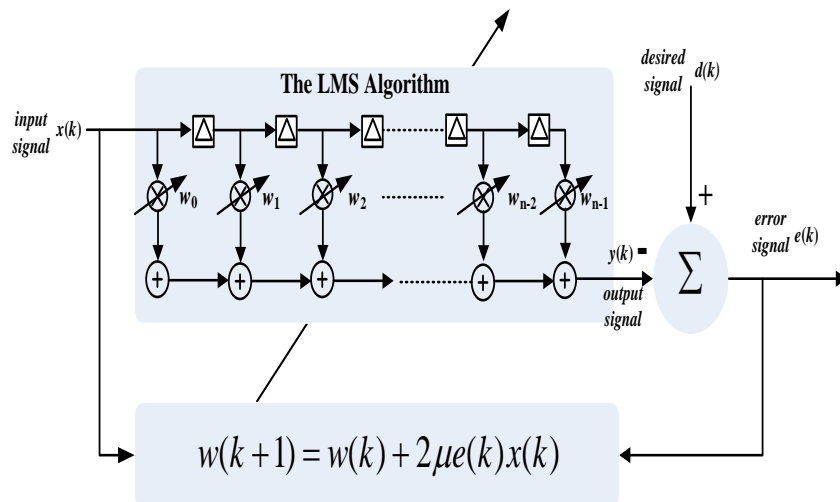
$$\begin{aligned}
 \frac{\partial e^2(k)}{\partial w} &= \left[ 2e(k) \frac{\partial e(k)}{\partial w_0(k)} \quad 2e(k) \frac{\partial e(k)}{\partial w_1(k)} \quad \dots \quad 2e(k) \frac{\partial e(k)}{\partial w_N(k)} \right]^T \\
 &= -2e(k)x(k) \\
 &= g_w(k)
 \end{aligned} \tag{9}$$

The resulting gradient based algorithm is known as the least mean square (LMS) algorithm, whose updating equation is

$$w(k+1) = w(k) + 2\mu e(k)x(k) \tag{10}$$

where the convergence factor  $\mu$  should be chosen in a range to guarantee convergence.

LMS is a search algorithm which a simplification of the gradient vector computation is made possible by appropriately modifying the objective function. Figure 4 depicts the realization of the LMS algorithm for a delay line input  $x(k)$ .

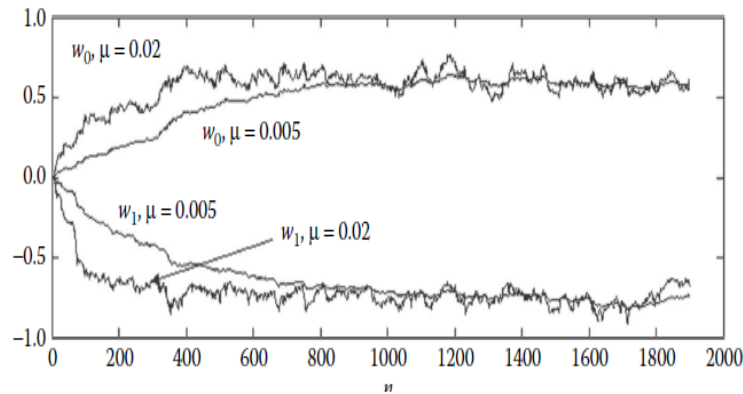


**Figure 4. The LMS Adaptive FIR Filter**

The stability of the LMS is dependent on the convergence factor,  $\mu$ . For stability it can be shown that the step size should be:

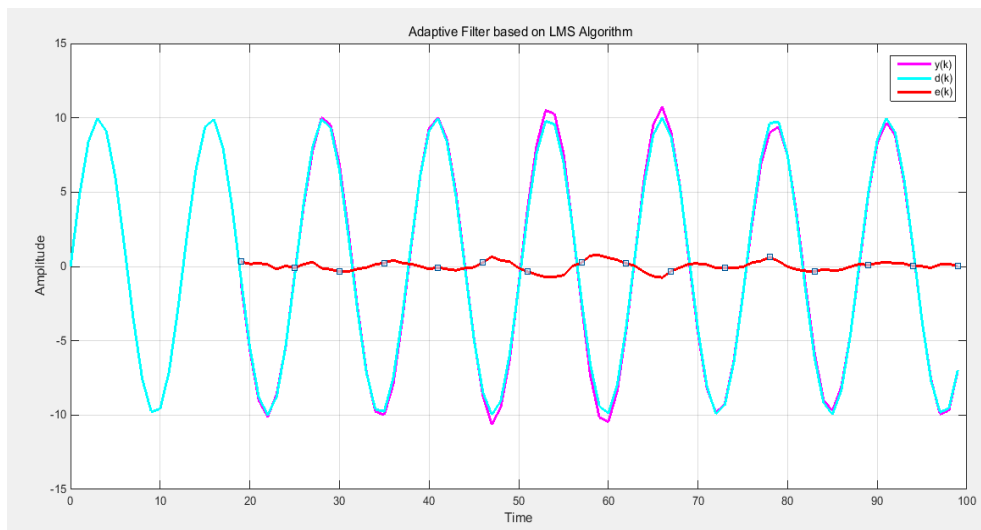
$$0 < \mu < \frac{1}{NE[x^2(k)]} \tag{11}$$

where  $E[x^2(k)]$  is effectively a measure of the power in the input signal. Figure 5, shows the trajectory of  $w_0$  and  $w_1$  versus the number of iterations for two different values of the convergence factor ( $\mu = 0.02$  and  $\mu = 0.005$ ).



**Figure 5. The Iterative Result of Different Convergence Factor**

Outwith these bounds it is likely that the LMS algorithm may go unstable, and the therefore fail to adapt to minimise the error. Figure 6, is an example of the LMS function.

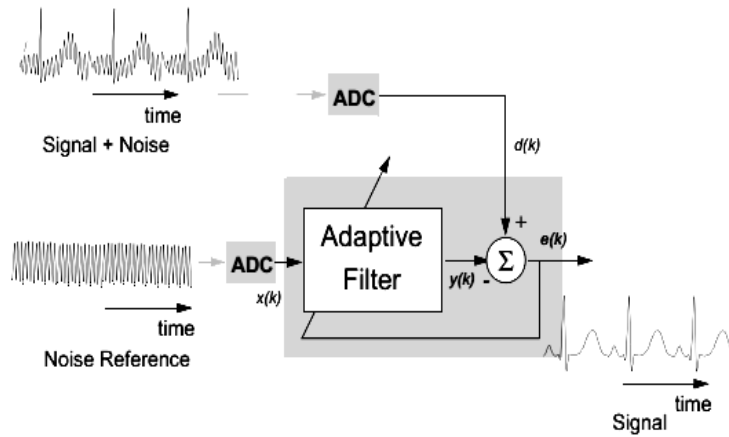


**Figure 6. An Instance of the LMS Adaptive FIR Filter**

#### 4. The Design of Noise-Cancel of Audio Adaptive Filter

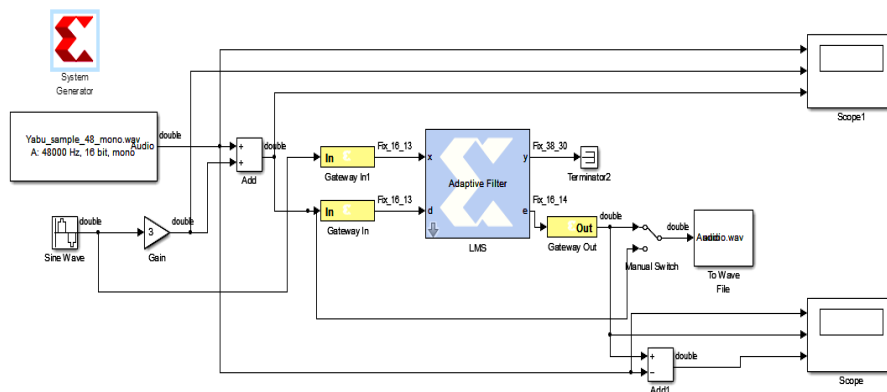
In the past decades, the application of speech recognition and speech communication has attracted considerable attention, in which the noise reduction of the audio signal is a key issue in speech signal processing. The most widely used method is LMS adaptive filtering algorithm, which achieves clean audio estimate by passing the noise observation through an optimal linear filter or transformation. However, Although LMS algorithm is relatively simple, but its structure is still much more complicated compared to the FIR filter for hardware implementation. This section will use the MBD workflow to complete the system modeling, simulation, HDL code automatically generated.

In the noise cancellation case, input signal  $d(k)$  includes speech signal and noise, and  $x(k)$  is a reference noise. If  $d(k)$  is used as an input to the adaptive filter with the signal corrupted by noise playing the role of the desired signal, after convergence the output error will be an enhanced version of the signal. Figure 7, illustrates a typical speech signal noise cancellation [13-14].

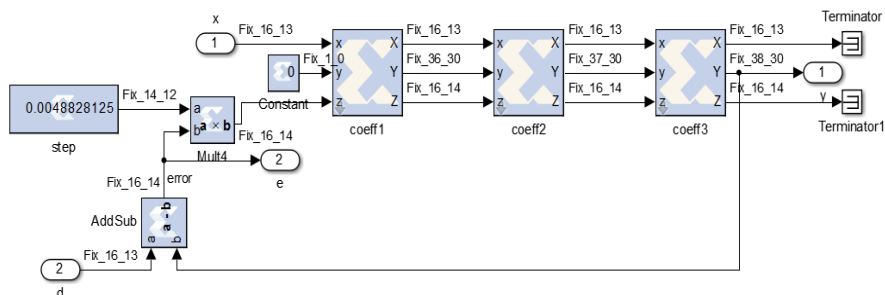


**Figure 7. Noise Cancellation via the LMS Adaptive FIR Filter**

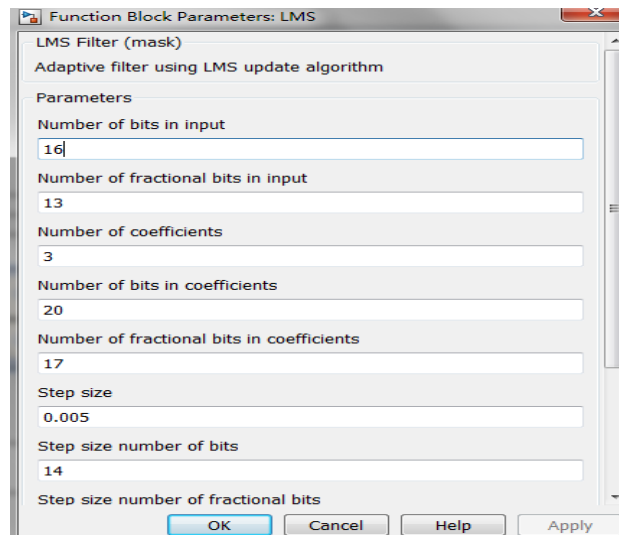
An adaptive LMS block is used to noise-cancel a tone present in an audio input signal. The design process of the whole system is completed with MBD workflow. The system model is shown in Figure 8. The frequency of the speech signal is 48kHz and the noise signal is 500Hz sine signal. Adaptive filtering algorithm is LMS algorithm. The body of the LMS module is presented in Figure 9. The parameter configuration of the LMS module is shown in Figure 10.



**Figure 8. Noise Cancellation Based on System Generator**

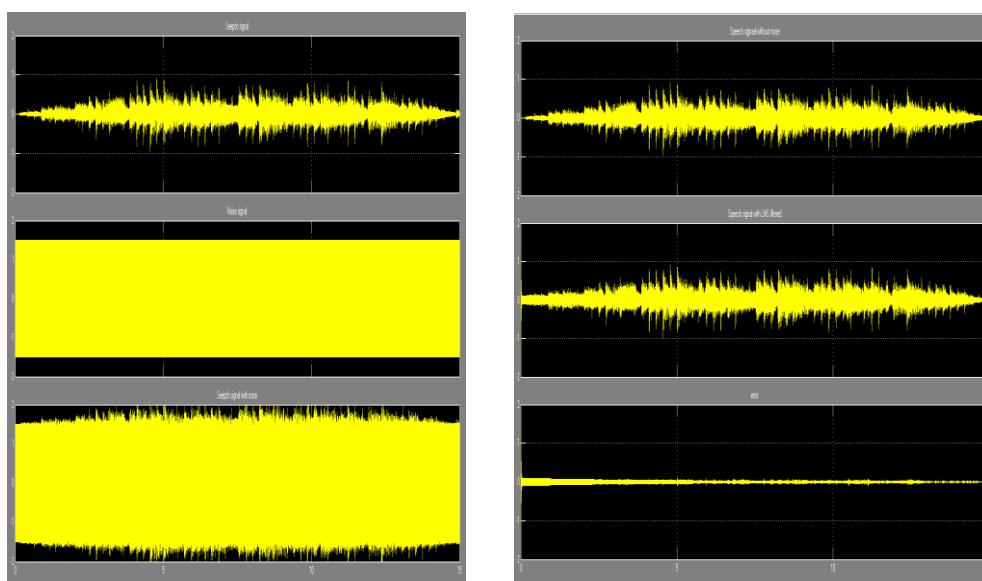


**Figure 9. The LMS Adaptive Filter Based on System Generator**



**Figure 10. The Module of Parameter Configuration in System Generator**

The FIR filter structure is used to achieve LMS adaptive filter. In order to facilitate the FPGA implementation, the LMS adaptive filter is implemented by a fixed-point algorithm. Fixed length bits are 16, and fractional bits are 13. The step size of the algorithm is 0.005, which is represented by 14bit. Experimental simulation waveforms are illustrated in Figure 11. Figure 11 (a), contains the input speech signal, the noise signal and the speech signal disturbed with the noise. Figure 11 (b), shows the filtered speech signal and error. We can see from Figure 11 (b), that the error is expected to be very small, and the noise signal has been eliminated.



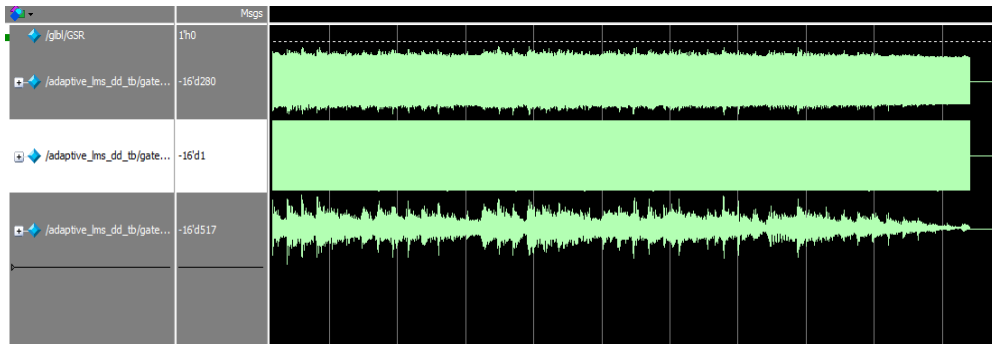
(a) Speech signal and Noise signal

(b) Filtered signal and Error

**Figure 11. The Simulation of Speech Noise Cancellation with LMS**

With the MBD workflow, we not only can quickly complete the system model and simulation, but also can automatically generate HDL codes, which are easy to achieve FPGA. Modelsim SE 10.2c is used as a function simulation tool for the automatic generation of HDL codes, and the simulation results are shown in Figure 12.





**Figure 12. The Simulation Result of Speech Noise Cancellation with Modelsim**

In the ISE environment, we use Xilinx Spartan-6 FPGA (xc6slx16-3csg324) to complete the LMS adaptive filter for hardware implementation. The hardware resource consumption of the LMS algorithm is shown in Figure 13.

adaptive_lms_dd_cw Project Status (05/11/2016 - 08:46:17)			
<b>Project File:</b>	adaptive_lms_dd_cw.xise	<b>Parser Errors:</b>	No Errors
<b>Module Name:</b>	adaptive_lms_dd_cw	<b>Implementation State:</b>	Synthesized
<b>Target Device:</b>	xc6slx16-3csg324	<b>Errors:</b>	No Errors
<b>Product Version:</b>	ISE 14.7	<b>Warnings:</b>	<a href="#">231 Warnings (231 new)</a>
<b>Design Goal:</b>	Balanced	<b>Routing Results:</b>	
<b>Design Strategy:</b>	<a href="#">Xilinx Default (unlocked)</a>	<b>Timing Constraints:</b>	
<b>Environment:</b>	<a href="#">System Settings</a>	<b>Final Timing Score:</b>	

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	114	18224	0%
Number of Slice LUTs	275	9112	3%
Number of fully used LUT-FF pairs	76	313	24%
Number of bonded IOBs	49	232	21%
Number of BUFG/BUFGCTRLs	1	16	6%

**Figure 13. The Resource Consumption of Speech Noise Cancellation with LMS**

## 5. Conclusion

In this paper, based on the MBD workflow is used to complete the noise cancellation of the speech signal based on LMS adaptive filter. The whole design workflow starts from algorithm to realization can be done without written any hardware and software code. Through the MBD process, we can quickly establish the system's modeling, simulation and automatic code generation. Compared with the traditional design process, based on the MBD design process, not only can quickly and efficiently build the system model, but also to perform the simulation in each stage and the automatic generation of code for the hardware implementation. This innovative design ideas and development process is not only the development of the new trend of complex system design process, but also improve the efficiency and security of the code generation. The proposed solution is particularly suitable for hardware development for the complex embedded system. It is believed that MBD will become the new trend of development in the FPGA development. Therefore, the implementation of DSP algorithms using MBD workflow will be widely used in the future.

## Acknowledgments

We would like to thank the members of the project team for the great efforts of scientific research projects. Moreover, we greatly appreciate the reviewers' comments that lead to an improved presentation of the results. This work was supported by Research Program of science and technology at Universities of Inner Mongolia Autonomous Region (NJZZ14288).

## References

- [1] B. Widrow, "Adaptive noise cancelling: Principles and applications", Proc. of IEEE., vol. 63, no. 2, (1975), pp. 1692-1716.
- [2] B. Widrow and E. Walach, "On the statistical efficiency of the LMS algorithm with nonstationary inputs", IEEE Trans. Inf Theory, vol. 30, no. 2, (1984), pp. 211-221.
- [3] MathWorks Inc, "HDL Coder Reference", (2014).
- [4] J. Liu, "Model based design workflow for FPGA compliance with DO-254 standard", Proceedings of international symposium on information technology in medicine and education, (2012), pp. 1026-1030.
- [5] S. SanchezSolano, M. BroxJimenez, E. delToro, P. BroxJimenez and I. Baturone, "Model-based design methodology for rapid development of fuzzy controllers on FPGAs", IEEE Trans. Ind. Informat, vol. 9, no. 3, (2013), pp. 1361-1370.
- [6] V. Soccì, "Implementing a model-based design and test workflow", Proceedings of 2015 IEEE International Symposium on Systems Engineering (ISSE), (2015), pp. 28 -30.
- [7] Xilinx Inc, "Vivado Design Suite User Guide: Model-Based DSP Design using System Generator", (2016).
- [8] E. Ferrara and B. Widrow, "The time-sequenced adaptive filter", IEEE Trans. Circ. Syst., (1981), pp. 519-523.
- [9] V. J. Mathews, "An efficient FIR adaptive filter using DPCM and the sign algorithm", IEEE Trans. ASSP, vol. 37, no. 1, (1989), pp. 128-133.
- [10] B. Widrow, P. E. Mantey, L. J. Griffiths and B. B. Goode, "Adaptive antenna systems", Proc. IEEE, vol. 55, (1967), pp. 2143-2159.
- [11] B. Widrow, J. McCool and M. Ball, "The Complex LMS Algorithm", IEEE Proceedings Letters, vol. 63, no. 4, (1975), pp. 719-720.
- [12] P. L. Feintuch, "An Adaptive Recursive LMS Filter", Proc. IEEE, vol. 64, no. 11, (1976), pp. 1622-1624.
- [13] J. E. Paul, "Adaptive Digital Techniques for Audio-Noise Cancellation", IEEE Circuits & Systems Magazine, vol. 1, no. 4, (1979), pp. 2-7.
- [14] H. Ding and C. Yu, "Adaptive lattice noise canceller antioptimal step size", ICASSP IEEE 86, vol. 11, (1986), pp. 2939-2942.

## Authors



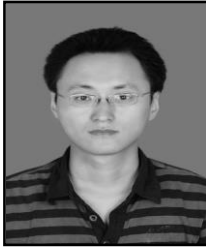
**Hua Ge**, He is also a Professor of Jining Normal University, China. His research interest includes Nonlinear Adaptive Signal Processing, High-performance DSP Algorithms and VLSI Architectures.



**Yang Nie**, He received the MS degree from Wuhan Research Institute of Posts & Telecommunications in 2009. He is currently working toward the PhD degree in the Digital Engineering Center of Communication University of China. He is also a lecturer of Jining Normal University, China. His research interest includes Compressive Sensing, High-performance DSP Algorithms and VLSI Architectures.



**Lili Jing**, She is currently a lecturer of Jining Normal University, China. Her research interest includes sensor detection technology, High-performance DSP Algorithms and VLSI Architectures.



**Pengyu Zhao**, He is currently a lecturer of Jining Normal University, China. His research interest includes PLC system design, and program design using C++.

