Adaptive bit Modulation equalizer

Yabo Dong¹, Hong Fan², Lin Ni³, Zhaoli Shi⁴, Ziwei Zhang⁵ and Hong Fan*¹

¹School of Information Science and Technology, Donghua University, Shanghai, China
²School of Information Science and Technology, Donghua University, Shanghai, China
³School of Information Science and Technology, Donghua University, Shanghai, China
⁴School of Information Science and Technology, Donghua University, Shanghai, China
⁵School of Information Science and Technology, Donghua University, Shanghai, China

*Corresponding author e-mail: 317014905@qq.com

Abstract. Multipath effects usually occur in information dissemination, which leads to inter-symbol interference and affects the quality of communication transmission. This paper uses an adaptive equalizer to achieve channel equalization and eliminate inter symbol interference, and develops and verifies it on the FPGA using the ISE suite. Compared with the equalizer output error signal calculated on the MATLAB software, the calculation results of the two platforms are basically the same, which confirms that the design is feasible. The FPGA-designed equalizer implements hardware acceleration and is more suitable for high-speed real-time signal processing. After the data passes through the adaptive bit equalizer, the signal-to-noise ratio is improved, thereby reducing the bit error rate and verifying the relationship between the bit error rate and the signal-to-noise ratio.

1. Introduction

The equalization algorithm is an adaptive filtering algorithm that automatically adjusts the filter parameters of the current time by using the results of the filter parameters obtained at the previous moment to adapt to the statistical characteristics of the signal and noise that are unknown or change with time. Filtering. The adaptive filter does not require prior knowledge of the input signal, and the amount of calculation is small, which is especially suitable for real-time processing [1]. This paper uses the FPGA-based least mean square (LMS) algorithm to achieve adaptive equalization. Because FPGA has parallel processing capability, easy integration and high reliability, it chooses to implement adaptive filtering algorithm on FPGA [2].

2. Adaptive equalizer

Equalizers are usually implemented by filters to compensate for the frequency characteristics of the distortion. The adaptive equalizer generally includes two working modes, a training mode and a tracking mode. The transmitter first sends a fixed-length sequence for the equalizer to receive and make the correct settings [3]. The equalizer can continuously adjust the gain from the actual signal transmitted according to the LMS algorithm to adapt to the random variation of the channel.
2.1. Symbolic LMS algorithm

In this paper, the symbol LMS algorithm is adopted, which is the optimization and simplification of the LMS algorithm. This time, the method of taking symbols for X(n) is adopted. The sign function Signum(·) is used to take the sign of X(n), and the result of multiplication is e(n) or -e(n). Since the symbol method does not give a specific gradient value, but gives the direction of the gradient iteration, there is no LMS algorithm stable in performance, so the algorithm error is relatively large. However, the convergence performance and the reduction of the steady-state error are in exchange for a reduction in the amount of calculation, an increase in the operation speed, and a saving of hardware resources [4]. Let X(n) and W(n) denote the input signal and weight of the adaptive filter, respectively. X(n) and W(n) are defined as follows:

\[ X(n) = [x(n), x(n-1), x(n-2), \ldots, x(0)]^T \]  
\[ W(n) = [w(n), w(n-1), w(n-2), \ldots, w(0)]^T \]

The symbol LMS algorithm can be represented by the following set of formulas:

\[ y(n) = W^T(n)X(n) \]  
\[ e(n) = d(n) - y(n) \]  
\[ \nabla W(n) = 2u \times \text{sign}(X(n))e^*(n) \]  
\[ W(n+1) = W(n) + \nabla W(n) \]

Where \( y(n) \) is the output signal, \( d(n) \) is the desired signal, \( e(n) \) is the error signal, \( u \) is the step size used to control the stability and convergence rate, and \( u \) is the integer power of 2 one. A typical symbol LMS algorithm is shown in Figure.1.

![Figure 1. Symbol LMS algorithm block diagram](image)

From the above set of recursive formulas, the output signal \( y(n) \) is equal to the product of the filter coefficient vector \( W(n) \) and the input signal \( X(n) \). The filter coefficient vector is continuously updated according to the error signal \( e(n) \), and the error signal is the difference between the desired signal \( d(n) \) and the actual output signal.

2.2. Frequency domain overlap

The main feature of the wireless mobile channel is the multipath effect. The multipath effect is that the same symbol waveform propagates through different paths, and the symbols overlap each other to form inter-symbol interference (ISI) [5]. The signal is prone to spectral aliasing in the transmission of too long. Therefore, we simulated a three-path channel, where the input signal \( X(n) \) is a signal generated by aliasing three signals of different powers into one. The purpose is to generate multipath source data so that the resulting input signal passes through an adaptive equalizer. We are using frequency domain equalization this time. Frequency domain equalization is corrected in the frequency domain. It adjusts the equalizer to match the total spectral characteristics of the channel and the equalizer to the ideal low-pass characteristic or equivalent low-pass characteristic, thus achieving inter-codeless interference transmission [6].
3. Adaptive bit modulation

Since the characteristics of the wireless communication channel are random and time-varying, that is, the channel characteristics are unknown, the channel response is also time-varying, which requires the equalizer to be able to track the time-varying characteristics of the wireless channel in real time. Therefore, the adaptive equalizer is designed by using the LMS algorithm, which can automatically update the tap coefficients of the filter, and performs noise processing on the input multipath signals in the time domain.

The implementation process of adaptive bit modulation is as follows:

I. Set an initial value \( W_m(0) \).

II. Calculate the output \( y(n) \) of the adaptive filter from equation (1),

\[
y(n) = \sum_{m=0}^{M} W_m(n)r(n-m)
\]

III. Estimating the error \( e(n) \) at the current time \( n \),

\[
e(n) = d(n) - y(n)
\]

IV. Update the weight of the filter with the LMS algorithm,

\[
W_m(n+1) = W_m(n) + 2ue(n)x(n-m)
\]

V. Verify that the error meets the criteria. If the criteria are met, the iteration is stopped. If not, continue the iteration. From \( n \) to \( n+1 \), to the next moment, repeat the above steps until the requirements are met. After the above algorithm process is implemented on the FPGA, the resources consumed are shown in Figure. (2). It can be seen that the algorithm consumes less resources.

![Figure.2. FPGA hardware resource usage diagram](image)

4. Experimental results and analysis

We designed an adaptive equalizer with \( M=8 \), \( u=1/256 \), adaptive equalization simulation of the symbol First, the input signal \( X(n) \) and the expected signal \( d(n) \) are generated by programming in MATLAB, and 16-bit quantization is performed on it. The quantized data is called in the Test Fixture file of ISE, and after simulation by ISE, the simulation result is obtained. The timing simulation of ModelSim of an adaptive filter based on FPGA and symbol LMS algorithm is shown in Figure.3. LMS algorithm, and programming using the Verilog programming language, and then adaptively based on the symbol LMS algorithm using ISE and ModelSim software. The equalizer performs compilation, synthesis, and timing analysis. This design is based on the development of Xilinx's xc3s250e-4vq100 chip.
As can be seen from Figure (3), in ModelSim, the waveforms of CLK, X(n), Y(n) and expected signal d(n) are shown, and after normalizing the data of X(n) and Y(n), we selected a range above 0.4 and less than 0.4 in X(n) using two different colors. After the adaptive bit equalizer with an X(n) data amplitude of more than 0.4, the range of the corresponding Output Y(n) data is equalized at about 0.4, while the range of X(n) data is less than 0.4, and after the adaptive bit equalizer, the corresponding Y(n) data amplitude is equalized at about 0.4. To ensure the stability of the output signal, achieve a balanced effect. To prove this, we also used MATLAB to design adaptive equalizers of the same order and the same u. Because in ISE's Test Fixture, we output the error signal, and also output the error signal in MATLAB. The two error signals are quantized and displayed in MATLAB, and compared. As can be seen from Fig.4. It can be known that the error of our adaptive equalizer based on FPGA design and the adaptive equalizer based on MATLAB design is the same, we can know that our design is correct.

In this paper, the MATLAB simulation of the equalizer is carried out to simulate the relationship between signal-to-noise ratio and bit error rate when the channel is a three-path channel. It can be seen from the Figure.5. that the error rate after using adaptive equalization decreases as the signal-to-noise ratio increases, thereby demonstrating the advantages of the adaptive equalizer based on the symbol LMS algorithm.
Figure 5. The relationship between signal to noise ratio and bit error rate

The running time of the adaptive bit equalizer on the FPGA and MATLAB is shown in Figure 6. It can be seen that the FPGA runs faster. So we implemented this design using an FPGA.

5. Conclusion

This design uses the symbol LMS algorithm to implement an adaptive bit equalizer. Simulation and verification are performed on ISE to achieve adaptive bit-equalization of the three-path channel. The equalizer runs faster and consumes less resources on the FPGA. Based on the above advantages, it can be applied not only to resources with limited resources, but also to scenarios requiring multi-path channels and 5G communication.

References

[1] Tang Shusen, Qi Chun, Zhang Wenqiang, Zhan Jiawei. Implementation of Adaptive Filter with FPGA and Improved LMS Algorithm[J]. Microcontrollers & Embedded Systems, 2018, 18(02): 18-20.

[2] Shiva, E. Senthilkumar, J. Manikandan and V. K. Agrawal, "FPGA implementation of reconfigurable adaptive filters," 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, 2017, pp. 2544-2547.

[3] Lei Zejun. Research on Adaptive Equalization Technology in Digital Communication System[J]. Information and Computer (Theoretical Edition), 2018(22): 185-186.

[4] Qiu Chenhui, Li Feng, Xu Zuqiang. Adaptive Filter Design Based on FPGA and Symbol LMS Algorithm[J]. Electronic device, 2014, 37(05): 904-907.

[5] P. V. Dutande, S. L. Nalbalwar and S. V. Khobragade, "FPGA Implementation of Filters for Removing Muscle Artefacts from EEG Signals," 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2018, pp. 728-732.
[6] S. Shaikh and S. Pujari, "Migration from microcontroller to FPGA based SoPC design: Case study: LMS adaptive filter design on Xilinx Zynq FPGA with embedded ARM controller," 2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT), Pune, 2016, pp. 129-134.