Efficient VLSI Architectures for High-Speed Ethernet Transceivers

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Jie Chen

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Professor Keshab K. Parhi, Advisor

August 2008

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my advisor, Professor Keshab K. Parhi, for his continuing encouragement, guidance and financial support (partially through the National Science Foundation under the grant number CCF-0429979) throughout all stages of my Ph.D. research work at the University of Minnesota. Without him, this dissertation would have been impossible. It certainly has been a pleasure and an honor to work with and learn from him. His enthusiasm and devotion to research will continue to inspire me in the future.

I would also like to thank Professor Gerald E. Sobelman, Professor Vladimir Cherkassky, and Professor Antonia Zhai for their support as members of my Ph.D. committee.

My thanks also go to former and current members of our research group. Particularly, Dr. Yongru Gu, Dr. Jun Tang, Dr. Yuping Zhang, Dr. Aaron E Cohen, Dr. Chao Cheng, Dr. Daesun Oh, Renfei Liu, and Xiaoming Zhu for their various help, as well as their friendship.

I would like to give special thanks to my parents, and my friends for their endless love, encouragement, and support during all these years. They have always been with me through the difficult times of my life. Without them, I would have never made it this far.

Abstract

This thesis investigates efficient VLSI architectural design aspects of a digital signal processing (DSP) transceiver in high speed multi-pair wireline communication systems, such as 10 Gigabit Ethernet over copper (10GBASE-T), with the goal to reduce the hardware complexity and power consumption of various DSP components while maintaining the speed and performance requirements. The covered topics mainly include efficient far-end crosstalk (FEXT) cancellers, novel multi-input multi-output (MIMO) equalizers combined with Tomlinson-Harashima Precoding (THP), low complexity echo and near-end crosstalk (NEXT) cancellers.

A novel feedforward delayed FEXT canceller in a THP based system is developed to remove FEXT as noise. Unlike conventional techniques on FEXT cancellation, the proposed FEXT canceller can mitigate the non-causal part of FEXT; thus it can achieve better cancellation performance. In addition, a modified design is developed by eliminating the feedback loops in the FEXT cancellers such that the resulting feedforward FEXT canceller is suitable for high speed applications.

FEXT has been found to contain information about the symbols transmitted from remote transmitters and thus MIMO equalization technique is proposed to jointly process ISI and FEXT such that the useful information in FEXT can be utilized. It is shown that the proposed architecture overcomes the limitation of the traditional equalization schemes and can achieve a better system performance and lower hardware complexity. A computationally efficient approach for calculating the optimal tap coefficients of MIMO equalizers and cancellers is also proposed to speedup the computation. Furthermore, a practical equalization scheme which combines the MIMO equalization technique and TH precoding technique is proposed for the real application of high speed Ethernet systems. Different from existing work on MIMO equalization, the proposed scheme exactly complies with the current 10GBASE-T standard and can be easily pipelined for high speed implementation. Hardware complexity reduction schemes by utilizing the increased decision point SNR (DP-SNR) are also considered.

Gigabit and multi-gigabit transceivers require very long adaptive filters for echo and

NEXT cancellation. Implementation of these filters not only occupies large silicon area but also consumes significant power. This thesis considers the problems of designing cost-efficient echo and NEXT cancellers mainly from two different aspects: one is to reduce the number of taps used in these noise cancellers; the other is to reduce the word-length used to represent data in a VLSI system. First, the sparse characteristics of the echo and NEXT channel impulse responses is exploited to reduce computational cost of adaptive echo and NEXT cancellers. Second, a novel word-length reduction scheme is proposed by replacing the original input to the echo and NEXT cancellers with a finite-level signal, which is then recoded to have shorter word-length. To further reduce the complexity of these cancellers, an improved design is proposed by exploiting the property of the compensation signal. Compared with the traditional design, the proposed echo and NEXT cancellers have exact input and do not suffer from the quantization problem, and thus they are more suitable for VLSI implementation. The design issues of adaptive noise cancellers by using the proposed word-length reduction method are also considered and modified designs of the adaptive cancellers are developed to further reduce the overall hardware cost of echo and NEXT cancellers with acceptable cancellation performance. Finally, the design approach for stable pole-zero modeling of long finite impulse response (FIR) filters is proposed.

Contents

1 Introduction **1**

1.1 Overview 1

1.2 Summary of Motivations and Contributions 3

1.2.1 Feedforward FEXT Cancellation with TH Precoding 3

1.2.2 MIMO Equalization and Cancellation for Ethernet Transmission . . 4

1.2.3 Cost-effective Echo and NEXT Cancellers 6

1.2.4 Stable IIR Approximation of Long FIR Filters 8

1.3 Outline of The Thesis 9

2 FEXT Cancellation for Ethernet Transmission **10**

2.1 Introduction 10

2.2 Conventional FEXT cancellation 13

2.3 Delayed Feedforward FEXT Canceller 16

2.4 FEXT Cancellation Combined with TH Precoding 18

2.4.1 Proposed FEXT Canceller with TH Precoding 18

2.4.2 Performance Analysis 21

2.4.3 Simulations and Discussions 28

2.5 Conclusion 36

3 MIMO Equalization and Cancellation for Ethernet Transmission **37**

3.1 Introduction 37

3.2 System Model 39

3.3 MIMO Equalization and Cancellation in Ethernet Systems 41
3.3.1 Joint MIMO-DFE Equalization and Cancellation 41
3.3.2 Separate MIMO-DFE Equalization and Cancellation 45
3.3.3 Simulation Results 47
3.4 Computation Method for MIMO equalizers and Cancellers 48
3.4.1 Background 48
3.4.2 Fast Computation Method 50
3.5 MIMO equalization incorporated with TH precoding 56
3.5.1 A Straightforward MIMO-THP Structure 58
3.5.2 Proposed MIMO Equalization Incorporated With TH Precoding . . 59
3.5.3 Simulations and Discussions 62
3.6 Hardware Complexity Reduction Scheme 67
3.7 MIMO Equalization with THP versus FEXT Cancellation 69
3.8 Conclusion 71

4 Design of Cost-Efficient Echo and NEXT Cancellers 73
4.1 Introduction 73
4.2 Adaptive Tap Management in Echo and NEXT Cancellers 77
4.2.1 Cross-Correlation Channel Estimation 80
4.2.2 Dynamic Threshold Rule 81
4.2.3 LMS With Partial Updating 84
4.2.4 Active Tap Tracking 84
4.2.5 Hardware Architecture 85
4.2.6 Simulation Results 88
4.3 Word-length Reduction in Echo and NEXT Cancellers 92
4.3.1 Traditional Echo and NEXT Cancellers with TH precoders 94
4.3.2 Proposed Word-length Reduction Scheme 96
4.3.3 An Improved Design 102
4.3.4 Complexity Analysis 111
4.3.5 Proposed Word-length Reduction Scheme in the Adaptive Filter . . 114

4.3.6 Simulations and Discussion 120
4.4 Further Cost Reduction of Adaptive Echo and NEXT Cancellers 126
4.4.1 Previous Designs of Echo and NEXT Cancellers in 10GBASE-T . . 127
4.4.2 Proposed Cost-efficient Weight Update Scheme 128
4.4.3 Computer Simulations and Hardware Comparison 135
4.5 Conclusion 138

5 Design of Stable IIR Echo and NEXT Cancellers 141
5.1 Introduction 141
5.2 The Proposed Method 143
5.2.1 Problem Formulation 143
5.2.2 Optimal MMSE Solution 145
5.2.3 Computational Complexity Comparison 149
5.3 Numerical Results and Discussions 154
5.4 Conclusion 160

6 Conclusion and Future Work 162
6.1 Conclusion 162
6.2 Future Research Direction 165

References 167

List of Figures

2.1 Impairments to data transmission over UTP 11
2.2 A straightforward design of FEXT canceller based on DFE structure 13
2.3 Insufficient FEXT cancellation due to non-causal part of FEXT 15
2.4 Delayed feedforward FEXT cancellation based on DFE structure 17
2.5 Tomlinson-Harashima precoder and its equivalent form 18
2.6 Idea of Tomlinson-Harashima precoding 19
2.7 Block diagram of the proposed FEXT cancellation with TH precoding . . . 20
2.8 Modified FEXT canceller for high speed application 22
2.9 The DP-SNR performance versus the lengths of echo and NEXT cancellers 30
2.10 The DP-SNR performance versus the lengths of the FFE and DFE filters . 31
2.11 The DP-SNR performance versus the decision delay 32
2.12 The DP-SNR performance versus the lengths of FEXT canceller and inserted bulk delay 33
2.13 Eye diagram of the system operation from startup to normal mode, Cat-6 75m, DP-SNR=36.89dB 34

3.1 Block Diagram of the MIMO Channel for 10GBASE-T 39
3.2 Joint MIMO-DFE equalization & echo and NEXT cancellation 42
3.3 Internal structure of a MIMO FFE filter 43
3.4 Separate MIMO-DFE equalization & echo and NEXT cancellation 46
3.5 Number of Multiplication Comparison 57
3.6 Complexity Reduction Comparison 57

3.7 A straightforward design of MIMO structure combined with TH precoding . 59
3.8 Internal structure of a MIMO-TH precoder for one channel 59
3.9 Block diagram of the proposed MIMO equalization with TH precoding . . . 60
3.10 Internal structure of the partial MIMO-FBE 60
3.11 Proposed MIMO equalization scheme I with TH precoding 62
3.12 Proposed MIMO equalization scheme II with TH precoding 62
3.13 The DP-SNR performance versus the lengths of echo and NEXT cancellers 64
3.14 The DP-SNR performance versus the lengths of the MIMO FFE and TH precoder 65
3.15 The DP-SNR performance versus the decision delay 65
3.16 The DP-SNR performance versus the bulk delay 66
3.17 Comparison of the learning curves for different designs during the training stage 70

4.1 Typical echo and NEXT impulse response (Cat-6 UTP) 78
4.2 Adaptive system identification model 80
4.3 Active tap positions identified using (4.11) 83
4.4 Hardware architecture of the proposed noise canceller (I) 87
4.5 Area-efficient hardware architecture of the proposed noise canceller (II) . . 87
4.6 Comparison of the learning curves between the traditional design and proposed design I/II for measured data 90
4.7 Comparison of the learning curves between the traditional design and proposed design I/II for scaled data 90
4.8 Comparison of the tap coefficients obtained from the traditional scheme and proposed scheme 91
4.9 Block diagram of a typical transceiver for one pair in 10GBASE-T 94
4.10 Input signal to echo and NEXT cancellers: traditional method 95
4.11 TH precoding: The equivalent form 96
4.12 Input signal to the echo and NEXT cancellers: proposed method 97
4.13 Encoding circuit for input signal 100

4.14 3-tap FIR filter with word-length reduction 102
4.15 Proposed echo and NEXT cancellers with word-length reduction 103
4.16 Improved design: echo and NEXT cancellers with separate inputs 105
4.17 Improved design: a 3-tap FIR filter with the input $v(n)$ 107
4.18 Circuit implementation for one tap 108
4.19 The optimized design for one tap without multipliers 108
4.20 Poles of the TH Precoder for different cases 115
4.21 Traditional Design: adaptive echo canceller without NEXT crosstalk 116
4.22 Application of the proposed method in adaptive echo canceller 117
4.23 Comparison of the convergence speed for different designs:(a) Direct design, (b) Modified design, (c) Traditional design 122
4.24 Adaptive echo canceller: traditional method 127
4.25 Adaptive echo canceller: proposed method 128
4.26 Adaptive echo canceller with modified input $d(n)$ 129
4.27 Filtered error adaptive echo canceller 130
4.28 Adaptive echo cancellers with separate inputs 132
4.29 Proposed adaptive echo canceller with cost-efficient weight updating 132
4.30 Comparison of learning curves of the proposed design with different adapting signals 134
4.31 Performances of the tradition design (Fig. 4.24) and our previous design (Fig. 4.25) 135
4.32 The proposed transceiver architecture of one pair in a 10GBASE-T Ethernet system 136

5.1 A generalized system identification model 143
5.2 Computational complexity of the proposed method and the method in [75] 153
5.3 Complexity reduction compared with the method in [75] 153
5.4 Pole-zero approximations of the first part of the feed-forward filter 155
5.5 Pole-zero approximations of the second part of the feed-forward filter 156
5.6 Pole-zero approximations of the feed-forward FIR filter 157

5.7 Pole-zero approximation of a 500-tap FIR echo canceller, $p = 80$, $q = 50$ . . 158

5.8 Pole-zero approximation of a 500-tap FIR echo canceller, $p = 50$, $q = 80$ . . 159

List of Tables

2.1 Design parameters for FEXT cancellation 32

2.2 Average decision-point (DP) SNR during the steady state, AWGN=-150dBm/Hz, Bulk delay D=40, $N_{fx} = 100$ 35

3.1 Parameter settings 47

3.2 Decision point SNR for different schemes (dB) 48

3.3 Design parameters for the proposed MIMO equalization 67

3.4 Average decision-point (DP) SNR during the steady state, AWGN=-150dBm/Hz, Bulk delay D=0, $N_x = 32$ 67

3.5 Design example for complexity reduction 68

4.1 Comparison of computational complexity 89

4.2 Comparison of hardware area 89

4.3 Decision point SNR during the steady stage for different designs 92

4.4 The number of active taps 92

4.5 Proposed data encoding table 99

4.6 Distribution of the finite-level compensation signal $v(n)$ for different cables 104

4.7 Data encoding table for the compensation signal $v(n)$ 107

4.8 Complexity comparison of four different designs 111

4.9 Implementation results of a 3-tap FIR filter 111

4.10 Complexity comparison between traditional and proposed methods in the 10GBASE-T application 112

4.11 Parameter Setup 121
4.12 Fixed point simulation setup 123
4.13 Residual error signal power at the inputs to the FFE for different designs, AWGN=-150dBm 124
4.14 Hardware Complexity Comparison of three designs 125
4.15 Residual error signal power at the inputs to the FFE for different designs, AWGN=-150dBm 137
4.16 Hardware Complexity Comparison of three designs 138

5.1 Computational complexity of the proposed solution 150
5.2 Computational complexity of the *Generalized* ARMA-Levinson algorithm, $m = max(p, q)$ 152
5.3 NNTEs of pole-zero approximations of a 40-tap FIR filter (dB) 155
5.4 NNTEs of pole-zero approximation of a 58-tap FIR filter (dB) 156
5.5 NNTEs of pole-zero approximations of a 64-tap FIR filter (dB) 157
5.6 NNTEs of pole-zero approximations of a 500-tap FIR echo canceller (dB) . 159
5.7 Comparison of required computation time simulated in MATLAB 160

Chapter 1

Introduction

1.1 Overview

The need for high data rates in local area network (LAN) applications has prompted the developments of fast Ethernet standards (such as 10 Mbps and 100 Mbps) since the mid-1990s. Gigabit Ethernet was the next generation, increasing the speed to 1000 Mbps. The initial standard for gigabit Ethernet was standardized by the IEEE in 1998. In 1999, IEEE802.3ab defined gigabit Ethernet transmission over Cat-5 unshielded twisted-pair (UTP) cables, which was known as 1000BASE-T. Faster 10 gigabit Ethernet standard over UTP cables, termed as 10GBASE-T, has become available since 2006. The target 10 Gbps throughput is achieved by duplex transmission over four pairs of UTP cables with 2.5 Gbps data rate per pair. The target bit error rate (BER) is 10^{-12} [7].

However, data transmission with high quality and capacity over such multi-pair UTP cable channels is limited by many impairments. For example, insertion loss (IL) and inter-symbol interference (ISI) are two of main impairments, and they are usually caused by the limited bandwidth and real impedance of the cable itself [30]. In addition, echo and

crosstalk interferences such as near-end crosstalk (NEXT) and far-end crosstalk (FEXT) are other major impairments and need to be suppressed to ensure reliable and high speed data transmission. Some other noises such as background noise and alien NEXT (ANEXT) interference from other cables can also reduce the received signal-noise ratio (SNR). To meet the desired throughput (*e.g.*, $10Gbps$) and target BER requirements (*e.g.*, 10^{-12}), a DSP transceiver has to perform a significant amount of digital processing operations, which include channel equalization, channel coding, and noise cancellation such as echo and crosstalk interferences including NEXT and/or FEXT.

The design and implementation of a DSP transceiver poses numerous challenges and open problems. For example, to meet the throughput requirement, all these DSP blocks must be able to operate at a high speed (*e.g.*, 800 MHz). The high-speed design and implementation of those components, especially for those containing feedback loops, are not trivial. However, the success of a DSP transceiver is largely dependent on its cost in terms of power consumption and hardware complexity, especially for the emerging 10GBASE-T application. To this end, this thesis addresses low complexity and low power design methodologies for efficient implementation of various DSP blocks including channel equalization and noise cancellation, targeting at the high speed Ethernet transmission systems, such as 10GBASE-T.

The goal of this thesis is mainly focused on reducing hardware complexity and power consumption of those DSP blocks in a 10GBASE-T transceiver while maintaining the speed and performance requirements. To achieve this goal, optimizations are considered at both algorithmic and architectural levels. The former requires extensive simulations of receiver algorithms and performance trade-off analysis; while the latter requires extensive architectural considerations with respect to feasibility of speed and power consumption. Thus, novel

equalization and noise cancellation schemes will be studied and the efficient architecture of a DSP transceiver with least area and power consumption will be sought in this thesis.

The specific motivations for the work presented in this thesis and the corresponding contributions will be summarized in the next section.

1.2 Summary of Motivations and Contributions

1.2.1 Feedforward FEXT Cancellation with TH Precoding

Motivation: In high-speed multi-pair wireline communication systems, such as 10 Gigabit Ethernet over copper (10GBASE-T), far-end crosstalk (FEXT) becomes a major impairment and needs to be suppressed to increase data rates. Conventional techniques based on crosstalk cancellation can not be easily applied due to the fact that the disturbing source of FEXT is generally unknown to the victims. One prior technique makes use of the tentative decision of the disturbing far-end transmitted signal as the input to the FEXT canceller, and both the FEXT canceller and linear equalizer are jointly adapted to combat intersymbol interference (ISI) and FEXT. One particular drawback of this technique is that the tentative decision is only a guess of the disturbing far-end transmitted signal, and incorrect tentative decisions occur in practice, which thereby increases the error rates. Another technique is to use the actual decision of the disturbing far-end transmitted signal as the input to the FEXT canceller. However, FEXT may exhibit non-causal characteristic which makes the FEXT cancellation insufficient. In addition, both techniques on FEXT cancellation did not consider error propagation problem inherent in the traditional decision feedback equalization (DFE) structure.

Contribution: We propose a novel approach to deal with FEXT interference in the application of high speed Ethernet systems. In the proposed approach, FEXT is treated as noise,

and a new feedforward FEXT canceller with TH precoding is developed by overcoming the limitations of prior FEXT cancellation applications. Compared with the existing designs, the proposed FEXT canceller can deal with the non-causal part of the FEXT, thus it can achieve better cancellation performance. Due to the use of the TH precoders, the error propagation problem is also alleviated in practice. The resulting FEXT cancellers do not contain feedback loops such that high speed VLSI implementation is possible. A modified design is also developed by using a finite signal as the input to the FEXT canceller such that the hardware complexity of the proposed FEXT canceller can be reduced. We also performed extensive simulations under practical Cat-6 UTP channel environment to verify the effectiveness of proposed designs in the 10GBASE-T application. Part of this work appeared in an IP disclosure filed at the University of Minnesota, Twin Cities [18].

1.2.2 MIMO Equalization and Cancellation for Ethernet Transmission

A. MIMO Equalization and Cancellation incorporated with TH Precoding

Motivation: Equalization technique is widely used in many wireline communication systems, such as digital subscribe line (DSL) systems and gigabit ethernet systems, to combat the intersymbol interference (ISI). The traditional equalization scheme is mainly based on single-input single output (SISO) structure, where equalization is performed individually for each channel and far end crosstalk (FEXT) is treated as noise to be cancelled at the receiver side. However, we note that, for each receiver, FEXT crosstalk inherently contains information about the symbols transmitted from the remote transmitters. Hence, it is better to exploit the far end crosstalk rather than simply treat it as noise.

Contribution: We propose to use multi-input multi output (MIMO) equalization technique to jointly process ISI and FEXT such that the useful information in FEXT can be

utilized. We demonstrate the advantage of the MIMO equalization technique by applying a general MIMO-DFE architecture to the 10GBASE-T application. To explore better system architecture, we compare two different arrangements of echo and NEXT cancellers based on the MIMO-DFE architecture. Furthermore, we propose a practical equalization scheme which combines the MIMO equalization technique and TH precoding technique for the real application of high speed Ethernet systems. Different from existing work on MIMO equalization, the proposed scheme exactly complies with the current 10GBASE-T standard and can be easily pipelined for high speed implementation. Owing to the both benefits of MIMO signal processing and TH precoding, the proposed architecture can achieve much better performance than the traditional equalization architecture. We also propose complexity reduction schemes to reduce the hardware complexity of the overall DSP transceiver by utilizing the increased decision point SNR (DP-SNR). Part of this work appeared in an IP disclosure filed at the University of Minnesota, Twin Cities [18].

B. Fast Computation of MIMO Equalizers and Cancellers with Long Taps

Motivation: Previous approaches on efficiently computing the tap coefficients of the equalizers are mostly based on the traditional finite-length MMSE-DFE structure [23, 24, 26]. Although these methods can be easily extended to the computation of the general MMSE-DFE equalizers with echo and NEXT cancellers in MIMO channels [25, 27], they may not always be computationally efficient for computing the optimal coefficients of the MIMO equalizers and cancellers in the cases where the cancellers have larger numbers of taps than the feed-forward equalizers. By using Al-Dhahir's method [25], the inversion of an embedded correlation matrix will be computationally intensive. In addition, the number of taps in echo cancellers and NEXT cancellers is not necessary to be the same in real applications. Thus, applying the efficient method in [27] is also not straightforward.

Contribution: We propose a new computationally efficient approach for calculating the optimum settings of the MIMO equalizers and cancellers for 10Gbase-T, assuming that the channel impulse response estimate and the noise characteristics are known. The proposed method provides an insight of the minimum mean-square error (MMSE) solution in a general MIMO system and shows that the mean-square error (MSE) optimization problem in a M-input N-output system can be decomposed into N independent minimization problems each with smaller size. Solving each separate problem in MMSE sense is computationally efficient, thus leading to substantial savings in overall computational complexity. Compared with a prior method, this new method is exact and much faster. This computation speedup also makes the analysis easier when alien crosstalk (*e.g.*, ANEXT) is considered in the channel model.

1.2.3 Cost-effective Echo and NEXT Cancellers

Motivation: Echo and NEXT cancellers are widely used to counter the effect of echo and NEXT noise. The typical way to implement those noise cancellers is to use finite impulse response (FIR) filters in digital domain. This straightforward approach, however, will lead to a significant hardware complexity if the number of taps in the FIR filters is large. For example, in the typical 10GBASE-T application [7], one echo canceller and three NEXT cancellers are needed for each pair of cables. Since there are four pairs of cables (four channels) in 10GBASE-T, a total of four echo cancellers and twelve NEXT cancellers are needed at the receiver end. To achieve high performance noise cancellation, each FIR based echo and NEXT canceller requires hundreds of taps [7], and the total number of taps in these cancellers is around 5600∼6800. Furthermore, all these cancellers need to be adapted to accommodate channel variation. Implementing these cancellers will consume large silicon area and power consumption. Therefore, efficient implementation of these cancellers is very

important.

Contribution: We approach the problems of designing cost-efficient echo and NEXT cancellers in the application of 10GBASE-T ethernet systems mainly from two different aspects. First, we focus on reducing the number of taps used in these noise cancellers to achieve the overall cost reduction of implementing these noise cancellers. We propose a new adaptive tap management algorithm to design echo and NEXT cancellers by exploiting sparse characteristics of the Cat-6 UTP copper channel. The proposed algorithm exhibits a significant computational complexity reduction as well as a faster initial convergence speed, compared with conventional LMS algorithm. We also develop an area efficient architecture to implement our algorithm. It is shown that, by applying the proposed method to gigabit transceiver design, we are able to achieve 50% and 29∼66.7% computational cost reduction during the initial training stage and steady state, respectively.

Second, we propose to reduce the word-length of the input signal to echo and NEXT cancellers to achieve the low complexity and low power design of these cancellers. The proposed design is derived by replacing the original input to the echo and NEXT cancellers with a finite-level signal, which is the sum of the input to the TH precoder and a finite-level compensation signal. In order to reduce the word-length of this modified input signal, an efficient data encoding and decoding scheme is developed. Moreover, we improve our design by analyzing the statistical properties of the compensation signal. It is shown that this improved design can also be optimized for a low power design by minimizing internal switching activities. Compared with the traditional design, the proposed echo and NEXT cancellers have exact input and do not suffer from the quantization problem, and thus they are more suitable for VLSI implementation. The design issues of adaptive noise cancellers by applying the proposed word-length reduction method are also considered and modified

designs of the adaptive cancellers are developed to further reduce the overall hardware cost of echo and NEXT cancellers with acceptable cancellation performance. Part of this work appeared in an IP disclosure filed at the University of Minnesota, Twin Cities [19].

1.2.4 Stable IIR Approximation of Long FIR Filters

Motivation: To meet a given magnitude-response specification, an FIR filter usually requires larger numbers of taps than an IIR filter. Thus the hardware cost of an FIR filter is more expensive than that of an IIR filter as the hardware cost in terms of chip area and power consumption is directly related to the total number of taps in the filter. If the phase response is not important, approximating a long FIR filter by an IIR filter (or pole-zero filter) with a smaller number of taps including numerator and denominator will reduce the overall hardware cost. However, since an IIR may not be stable, it is not trivial to find a stable IIR filter corresponding to a given long FIR filter.

Contribution: We propose a new efficient method for computing the optimal coefficients of the stable IIR filter, assuming that the coefficients of the target FIR filter are known. Unlike the existing approaches, we formulate an optimization problem which minimizes the mean-square error (MSE) between the target FIR filter and the approximate IIR filter. This MMSE problem can be solved by use of many methods in [23, 24, 26]. We exploit the internal structured matrices in the MMSE solution, and then derive a computationally efficient solution, which is general and applicable to cases where the number of poles and zeros of the approximate IIR filter is different. We also show that the proposed method is as accurate as the *Generalized* ARMA-Levinson algorithm but with much lower computational complexity. Finally, we apply the proposed method to generate stable pole-zero approximation of long FIR filters in the application of DFEs and echo cancellers.

1.3 Outline of The Thesis

This thesis is outlined as follows. In Chapter 2, conventional approaches on FEXT cancellation are briefly reviewed, and then the proposed approach for FEXT cancellation with TH precoding is presented in detail. Simulations are also provided to demonstrate the proposed design.

Chapter 3 presents another way to deal with FEXT crosstalk in multi-pair wireline communication systems, by applying MIMO signal processing techniques. A computationally efficient approach for calculating the optimal settings of MIMO equalizers and cancellers is also presented. A practical equalization scheme which combines the MIMO equalization technique and TH precoding technique is proposed for the real application of high speed Ethernet systems. Finally, the proposed MIMO equalization scheme is compared with the FEXT cancellation scheme proposed in Chapter 2 in terms of hardware complexity, convergence speed, and DP-SNR performance.

Chapter 4 addresses the problem of low-complexity designs of echo and NEXT cancellers in the application of 10GBASE-T Ethernet systems. Two different approaches are presented to achieve low complexity and low power designs of these cancellers: one is to reduce the number of taps used in these noise cancellers; the other is to reduce the word-length used to represent data in a VLSI system.

Chapter 5 presents the proposed approach for stable IIR filter modeling of long FIR filters, and a computational complexity analysis is also performed to compare the proposed method with the *Generalized* ARMA-Levinson algorithm.

Chapter 6 gives a brief conclusion and lists some future research tasks.

Chapter 2

FEXT Cancellation for Ethernet Transmission

2.1 Introduction

In high-speed multi-pair wireline communication systems, such as 10 Gigabit Ethernet over copper (10GBASE-T), far-end crosstalk (FEXT) becomes a major impairment and needs to be suppressed to increase data rates. Conventional techniques based on crosstalk cancellation are not suitable for FEXT due to the fact that the disturbing source of FEXT is generally unknown to the victims. This chapter reviews different FEXT cancellation schemes and then proposes an efficient FEXT cancellation scheme which can achieve much better performance than other existing noise cancellation schemes.

Fig. 2.1 illustrates a typical channel environment for gigabit and multi-gigabit Ethernet systems. It can be seen that data transmission over such multi-channel communication systems is primarily limited by channel attenuation (insertion loss), intersymbol interference (ISI), echo, crosstalk interferences such as near-end crosstalk (NEXT), far-end crosstalk

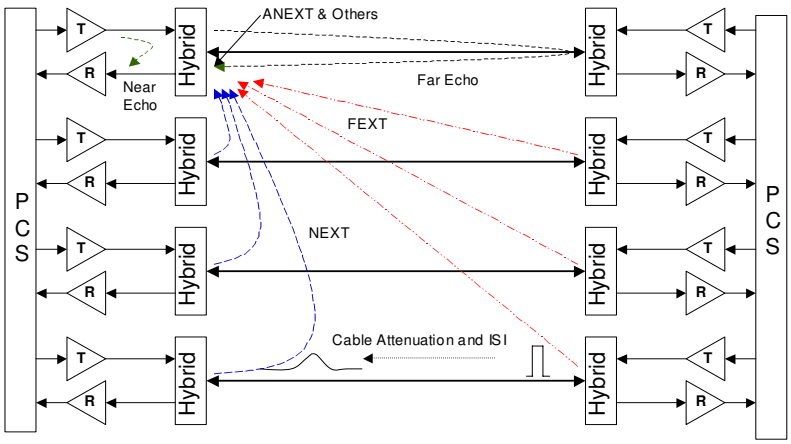


Figure 2.1: Impairments to data transmission over UTP

(FEXT), and alien crosstalk. To meet the throughput and BER requirements, the receiver needs to perform a significant amount of digital processing operations, which include channel equalization, echo and NEXT cancellation. FEXT can generally be tolerated in 1000BASE-T [10]. However, for high speed applications, such as 10GBASE-T, FEXT is a major impairment which needs to be reduced by about 25dB [8].

Conventional technology that addresses FEXT interference is mainly based on the concept of noise cancellation. The FEXT canceller is employed at the receiver side to suppress FEXT interference. Due to the fact that the disturbing source of FEXT is generally unknown to the victims, it is difficult to apply an accurate input to the FEXT canceller at the receiver side. In [12], a tentative decision of the disturbing far end transmit signal was used as the input to the FEXT canceller, and both the FEXT canceller and linear equalizer were jointly adapted to combat ISI and FEXT. To get a sufficient FEXT cancellation, the tentative decision needs to be accurate enough. However, accurate tentative decisions are hard to acquire and incorrect decisions occur in practice, which thereby increases error rates. Instead of using the tentative decisions as the input to FEXT canceller, the final

decisions of the transmitted symbols are used in [13], where a structure based on multi-input multi-output (MIMO) decision feedback was proposed to remove the FEXT crosstalk in digital subscriber line (DSL) systems. Due to the error propagation problem inherent in the decision feedback based structure, this technique is not suitable for Ethernet systems. In addition, nonlinear feedback loops in this structure limit its use for high speed applications. In [20], a joint equalization and decoding scheme was proposed for the application of 10GBASE-T. Later on, a new type of equalizer that combined the shortened filter and a decision feedback equalizer (DFE) was developed to achieve a better performance [21]. However, both designs do not address the FEXT interference adequately, and also the error propagation problem remains unsolved.

This chapter presents a novel approach to deal with FEXT interference in the application of high speed Ethernet systems. In the proposed approach, FEXT is treated as noise, and a new feedforward FEXT canceller is designed to remove FEXT interference at the receiver side. Compared with the existing designs, the proposed FEXT canceller can deal with the non-causal part of the FEXT [30], thus it can achieve better cancellation performance. Instead of using the conventional DFE structure, an equalization structure is applied with four separate TH precoders at the transmitter side and four corresponding feedforward equalizers at the receiver side. Based on this structure, the error propagation problem can be alleviated. In addition, by eliminating the feedback loops, the resulting feedforward FEXT canceller is suitable for high speed applications.

The rest of this chapter is organized as follows. In Section 2.2, conventional technique on FEXT cancellation is briefly reviewed, and a general DFE structure for FEXT cancellation is analyzed. Section 2.3 introduces the delayed feedforward FEXT cancellation scheme. Section 2.4 presents the proposed approach for FEXT cancellation with TH precoding and

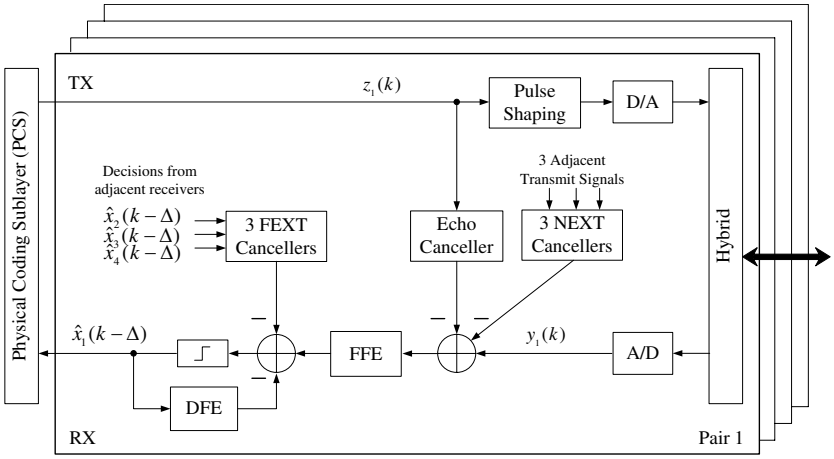


Figure 2.2: A straightforward design of FEXT canceller based on DFE structure

develops a new feedforward FEXT canceller at the receiver side. In addition, the average decision-point (DP-SNR) over four pairs is derived as a performance measure to evaluate the proposed FEXT canceller. Simulation results and discussions are presented in this section. Finally, Section 2.5 concludes the work of this chapter.

2.2 Conventional FEXT cancellation

Consider signal transmission path in Fig. 2.1, FEXT interferences are generated by disturbing signals from adjacent transmitters at the far end of the transmission link, and pass through the victim channel, *i.e.*, the first channel in Fig. 2.1. In order to suppress FEXT interferences, FEXT cancellers are introduced for the victim channel at the near end receiver. According to [12,13], the inputs to these FEXT cancellers can be obtained from the tentative decisions of the disturbing signals from the adjacent receivers. The outputs of the FEXT cancellers are then subtracted from the received signal for the victim channel, thus cancelling the FEXT interferences.

Based on this idea, Fig. 2.2 shows a straightforward design of the FEXT cancellers in

a typical Ethernet transceiver for one pair, and similar structures are applied to the other three pairs. In this figure, the local transmitted signal $z_1(k)$ goes through a shaping filter, a digital to analog (D/A) converter, hybrid circuitry, and finally is coupled to the UTP cable for transmission. On the receiving path, the received signal is first sampled to get $y_1(k)$, and then the outputs of the echo and NEXT cancellers are subtracted from $y_1(k)$ to remove the echo and NEXT interferences. After that, a DFE structure, containing a feed-forward equalizer (FFE) and a feedback equalizer (FBE), is used to perform channel equalization. Assume the decision delay is Δ, and then the estimates of the far end transmitted signals $\hat{x}_i(k-\Delta)$ are fed to the FEXT cancellers for proper FEXT cancellation. Note that all the filters in the figure are adapted to the varying channel environment and the coefficients of these filters can be obtained during an initial training period.

However the design in Fig. 2.2 leads to insufficient FEXT cancellation in practice. Notice that the inputs to the FEXT cancellers are from the decisions of the disturbing signals with a certain decision delay Δ. It is impossible to generate a FEXT cancellation signal until the decision of the disturbing signal is available. Thus, if FEXT occurs at the output of the FFE filter before the decision of the disturbing signal is made, then part of the FEXT will not be cancelled. In practice, this does happen because of two reasons. Firstly, the FFE is only designed for mitigating the pre-cursor ISI for each pair. However, the pre-cursor FEXT will not be mitigated accordingly. Secondly, FEXT exhibits non-causal characteristics, which means FEXT can arrive at the near end receiver before its disturbing source arrives [30]. To be consistent, in this chapter we define the non-causal part of the FEXT as the FEXT components that arrive at the output of the FFE before the decisions of the disturbing signals are made. For a practical channel environment, this can be illustrated in Fig. 2.3. In this figure, it shows that the decision delay is 420. The optimal coefficients

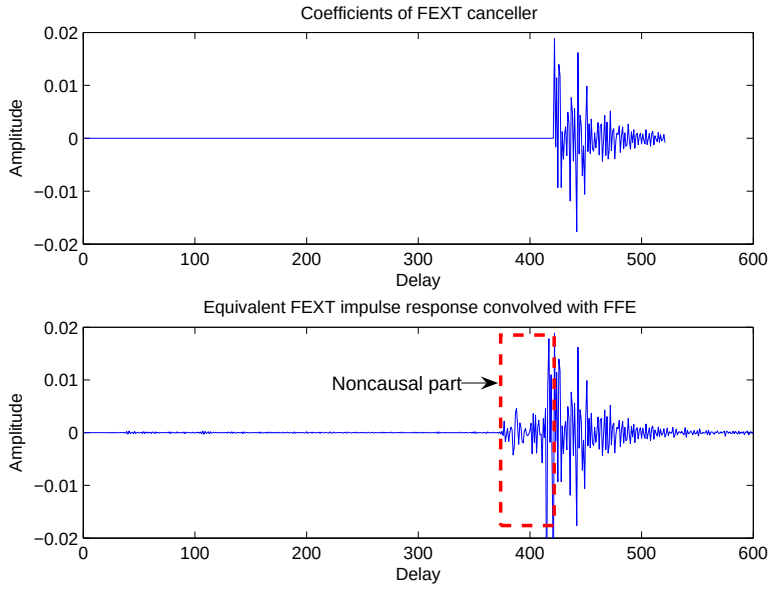


Figure 2.3: Insufficient FEXT cancellation due to non-causal part of FEXT

of the FEXT canceller are obtained based on MMSE criterion [84]. After the decision delay of 420 samples, the decision at the slicer will be available to generate a FEXT cancellation signal. However, it can be seen that part of FEXT already occurs at the output of the FFE filter before the decision delay. Thus, failure to account for this non-causal part of the FEXT will lead to insufficient FEXT cancellation, even though all previous decisions are correct.

In real applications, the performance of the design in Fig. 2.2 is even worse due to the problem of error propagation in the decision feedback structure. In addition, nonlinear feedback loops in this structure limit its use for high speed applications. Thus, it is important to develop new approaches to overcome these problems. Next, based on the conventional crosstalk cancellation techniques, a new feedforward FEXT canceller is first proposed.

2.3 Delayed Feedforward FEXT Canceller

As discussed in Section 2.2, the conventional FEXT cancellation is not sufficient because the non-causal part of FEXT is not cancelled. In Fig. 2.3, if we can delay the received signal corrupted by the FEXT interference at the victim channel till the FEXT cancellation signal from disturbing channel is available, then non-causal FEXT can be removed.

Based on this idea, Fig. 2.2 can be modified by inserting a bulk delay D at output of the FFE. The resulting design is shown in Fig. 2.4, where three FEXT cancellers are needed for the victim pair 1 since $y_1(k)$ suffers from the FEXT interferences from the other three disturbing channels. From this figure, it is seen that the FEXT cancellers do not contain any feedback loop. The inputs to these FEXT cancellers are called tentative decisions, which are described as gross estimates of the disturbing signals and can be obtained by slicing the outputs of the FFE from adjacent receivers. For example, tentative decision $\tilde{x}_1(k)$ at victim pair 1 is obtained by slicing the output at point A. Initially, the cancellers and equalizers are not well trained such that the tentative decisions are mostly wrong and can not be used as the inputs to the FEXT cancellers. Therefore, an initial training stage is needed to get the reliable tentative decisions. After convergence, the system would operate in a direct-decision (DD) mode to adapt to the varying channel environment.

It should be noted that the bulk delay D is determined by the duration of the non-causal FEXT part as shown in Fig. 2.3. By adjusting the bulk delay D, we can decide how much non-causal FEXT can be cancelled. For the general cases, applying larger bulk delay and longer FEXT cancellers would lead to better FEXT cancellation performance. However, in a similar design presented in [12], the inserted bulk delay at the output of the equalizer was defined as the delay due to the FEXT cancellers. It was shown that by applying larger bulk delay and longer FEXT cancellers, the performance was even worse. This undesired

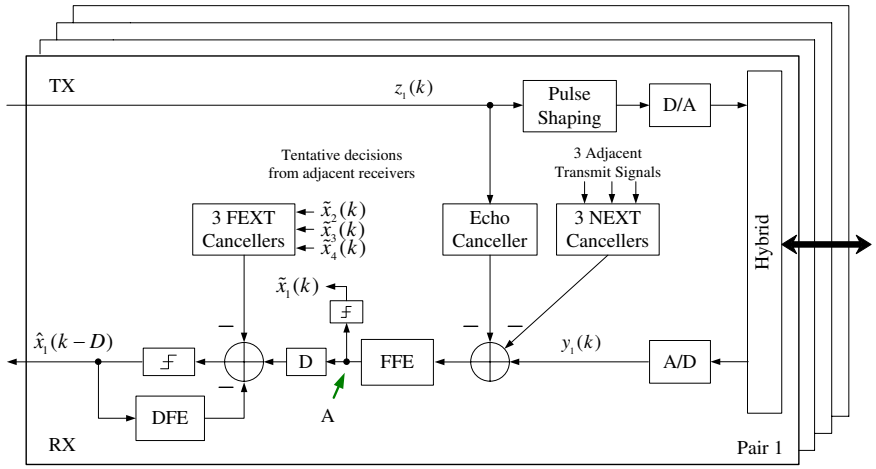


Figure 2.4: Delayed feedforward FEXT cancellation based on DFE structure

result was explained in [12] due to the drawback of the proposed blind startup algorithm for large delay cases. In this chapter, this problem is avoided by applying the ideal training process during the initial period with the known training sequences, which is supported by the 10GBASE-T standard.

Consider the point A in the design as shown in Fig. 2.4, we need to note that the signal at point A continues to suffer from FEXT crosstalk, post-cursor ISI and other background noise assuming the echo and NEXT crosstalks and pre-cursor ISI are removed perfectly. Hence, incorrect tentative decisions, *i.e.*, $\tilde{x}_i(k) \neq x_i(k)$, will occur to generate wrong FEXT cancellation signals, which may lead to wrong final decision of far end transmitted signal, *i.e.*, $\hat{x}_i(k-D)$. As we can see, the signal will be fed back to decide the next decision through a DFE structure. Therefore, the error propagation problem will result in an increase of error rates. To solve this problem, a new FEXT canceller is developed for a TH precoder based transceiver.

2.4 FEXT Cancellation Combined with TH Precoding

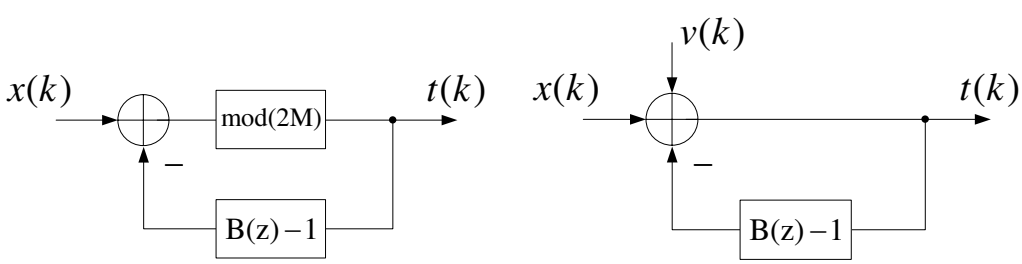
2.4.1 Proposed FEXT Canceller with TH Precoding

To describe the proposed FEXT canceller, we first briefly introduce the TH precoder.

The TH precoder was first proposed by Tomlinson and Harashima in 1971 [59–61]. It has similar structure as a traditional DFE, except that the decision device in the DFE is replaced with a modulo device in the TH precoder as shown in Fig. 2.5(a). The operation of TH precoding can be interpreted by using the equivalent form of the TH precoder as shown in Fig. 2.5(b). A unique compensation signal $v(k)$, which is a multiple of $2M$, is added to the transmitted M-PAM signal $x(k)$ such that the output of the precoder $t(k)$ lies in the interval $[-M, M)$. If the input of the TH precoder, $x(k)$, is *i.i.d*, it can be shown that $t(k)$ has uniform distribution over $[-M, M)$ [32]. The effective transmitted data sequence in Z-domain is given by

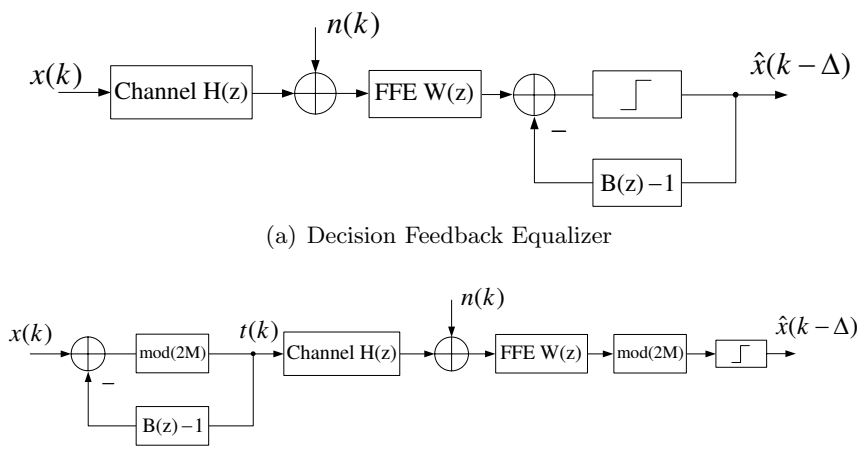
$$T(z) = \frac{X(z) + V(z)}{B(z)}, \tag{2.1}$$

where $B(z)$ is a causal FIR filter in the TH precoder feedback path. From (2.1), we see that a TH precoder can be viewed as an IIR filter with the input equal to the sum of the original TH precoder and a finite level compensation signal, *i.e.*, $x(k) + v(k)$.



(a) A TH precoder (b) Equivalent Form of a TH Precoder

Figure 2.5: Tomlinson-Harashima precoder and its equivalent form



(a) Decision Feedback Equalizer

(b) TH precoding for DFE

Figure 2.6: Idea of Tomlinson-Harashima precoding

If we assume the channel state information (CSI) is known at the transmitter side, then a general TH precoder can be applied at the transmitter to perform channel equalization. Fig. 2.6 illustrates a typical application of TH precoder, which is converted from a traditional DFE structure. As we can see from Fig. 2.6(b), DFE part is implemented at the transmitter without suffering from decision errors, thus the error propagation problem in a DFE structure can be eliminated. It is also noted that the transmitted signal $x(k)$ can be recovered from the output of the FFE by performing a modulo operation.

Applying the TH precoding technique to the design in Fig. 2.4 would be straightforward if there is no FEXT cancellation. However, we should note that the disturbing signal of the FEXT is now affected by $t_i(k)$ rather than $x_i(k)$ due to the TH precoding. In general, the output of the TH precoder $t_i(k)$ is continuous in the interval $[-M, M)$ [32]. In other words, the number of different outputs of the modulo device in the TH precoder is infinite, which makes it difficult to determine the tentative decision of the disturbing signal, $\tilde{t}_i(k)$. Thus, a design of the FEXT canceller in a TH precoding system becomes difficult.

Instead of determining $\tilde{t}_i(k)$ directly, a tentative decision of the sum signal $x_i(k) + v_i(k)$

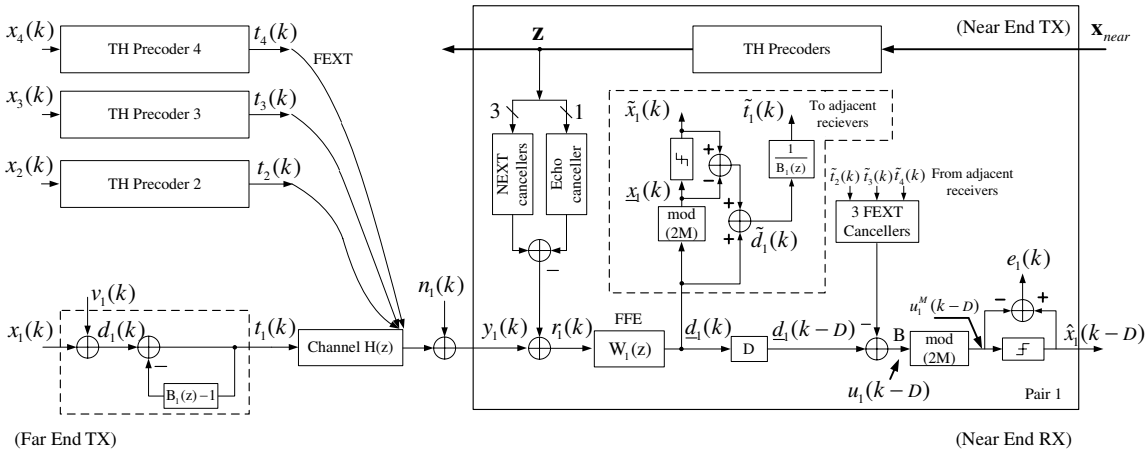


Figure 2.7: Block diagram of the proposed FEXT cancellation with TH precoding

is first made, and then $\tilde{t}_i(k)$ can be obtained using equation (2.1). Fig. 2.7 shows the block diagram of the proposed FEXT cancellation in a TH precoding system. At the far end transmitter, the transmitted signal $x_i(k)_{i=1,2,3,4}$ is first pre-equalized by four corresponding TH precoders. The output of the TH precoders $t_i(k)_{i=1,2,3,4}$ will be sent to the channel. At the near end receiver, pair 1 is taken as an example to illustrate the structure of the receiver, and the other three pairs will have similar architectures. In this figure, the received noisy signal $y_1(k)$ contains FEXT interferences, echo and NEXT interferences (which are not shown in the figure), and additive white noise $n_1(k)$. First, echo and NEXT cancellation is performed to obtain the signal $r_1(k)$. After that, signal $r_1(k)$ is fed to the FFE to remove the pre-cursor ISI. Ideally, if channel equalization is perfect, the output of the FFE, $\underline{d}_1(k)$, will only contain $x_1(k)+v_1(k)+n_1(k)$, as well as FEXT interferences. A tentative decision $\tilde{d}_1(k)$ is then obtained by slicing $\underline{d}_1(k)$. To avoid using a complex multilevel slicer, an alternate implementation is shown in Fig. 2.7. Since the FEXT interference is small, the reliable tentative decision $\tilde{d}_1(k)$ can be achieved. According to (2.1), an estimate of the effective transmitted signal, $\tilde{t}_1(k)$ can be obtained by inserting an IIR filter $\frac{1}{B_1(z)}$ after $\tilde{d}_1(k)$ as shown in Fig. 2.7. Similarly, $\tilde{t}_i(k)_{i=2,3,4}$ can be obtained from adjacent receivers, and then

can be used as the inputs to the FEXT cancellers. The final estimate of the transmitted symbol is denoted as $\hat{x}_1(k)$ with a delay D, which is inserted at the output of the FFE.

It should be noted that the coefficients of the TH precoders need to be set the same as the corresponding DFE coefficients, which are obtained during the initial training stage [36]. During the normal data transmission stage, decision error $e_i(k)$ is used to update the filters corresponding to each pair.

However, in the application of 10GBASE-T, we may note that feedback loops inside the TH precoders and the part where $\tilde{t}_1(k)$ is generated would be the bottleneck for high speed operation both in transmitter and receiver. In [29], the high speed design of the TH precoder has been addressed. To speed up the receiver in the proposed design, we can simply eliminate the IIR filter $\frac{1}{B_1(z)}$ after $\tilde{d}_1(k)$, and directly use $\tilde{d}_1(k)$ as the input to the FEXT canceller. By doing so, each of the resulting FEXT cancellers will implicitly contain an IIR filter. Thus, to achieve good FEXT cancellation performance, the length of the FEXT cancellers needs to be long enough. However, the advantage is that there are no feedback loops in the receiver structure and high speed design is possible. The resulting modified design is shown in Fig. 2.8, where the inputs to the FEXT cancellers corresponding to pair 1 are changed to $\tilde{d}_i(k)_{i=2,3,4}$ from adjacent receivers.

The modified design can also achieve a low complexity design if we note that $\tilde{d}_i(k)$ has only finite number of values. Thus, those FEXT cancellers with the input $\tilde{d}_i(k)$ can be efficiently implemented by applying techniques such as pre-computation and look-ahead [62, 85].

2.4.2 Performance Analysis

In order to evaluate the performance of the proposed FEXT canceller, we define the average decision-point SNR (DP-SNR) over four pairs as a performance measure. In this subsection,

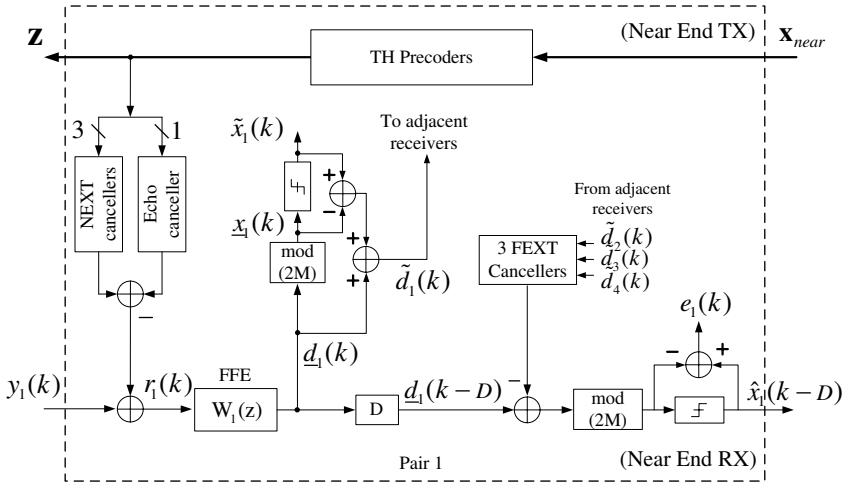


Figure 2.8: Modified FEXT canceller for high speed application

the theoretical average DP-SNR is derived based on the proposed design in Fig. 2.7, and then a simple least mean square (LMS) algorithm [84] is applied to update the equalizers and FEXT cancellers.

In Fig. 2.7, the ith effective transmitted signal $t_i(k)$ can be expressed as:

$$t_i(k) = d_i(k) - \sum_{l=1}^{N_b} b_i(l) t_i(k-l), \tag{2.2}$$

i.e.,

$$d_i(k) = t_i(k) + \sum_{l=1}^{N_b} b_i(l) t_i(k-l), \tag{2.3}$$

where $d_i(k)$ is defined as $d_i(k) = x_i(k) + v_i(k)$. $b_i(k), k = 0, \ldots, N_b$ is the coefficient of the ith TH precoder with $b_i(0) = 1$, and N_b is the order of the TH precoder. If we define the coefficient vector as $\bar{\mathbf{b}}_i \triangleq [b_i(1), \ldots, b_i(N_b)]^H$, equation (2.3) can be written as:

$$d_i(k) = t_i(k) + \bar{\mathbf{b}}_i^H \bar{\mathbf{t}}_i(k-1), \tag{2.4}$$

where the superscript H denotes Hermitian transposition, and $\bar{\mathbf{t}}_i(k)$ is an $N_b \times 1$ symbol vector, *i.e.*,

$$\bar{\mathbf{t}}_i(k-1) = [t_i(k-1), t_i(k-2), \ldots, t_i(k-N_b)]^H.$$

A typical transmission channel is shown in Fig. 2.1, and it can be modeled as two equivalent 4×4 MIMO channels. One is $\mathbf{h}_{i,j}$ which denotes the discrete-time channel impulse response from the ith transmitter (at far end) to the jth receiver (at near end) with length $N_1 + 1$, and the other is $\mathbf{g}_{m,n}$ which denotes the echo and NEXT discrete-time channel impulse response from the mth transmitter (at near end) to the nth receiver (at near end) with length $N_2 + 1$. The received signal at jth channel is given by

$$y_j(k) = \sum_{i=1}^{4} \mathbf{h}_{i,j}^H \mathbf{t}_i(k) + \sum_{m=1}^{4} \mathbf{g}_{m,j}^H \mathbf{z}_m(k) + n_j(k), \tag{2.5}$$

for $j = 1, \ldots, 4$. $n_j(k)$ denotes the additive white Gaussian noise (AWGN) at jth channel. $\mathbf{t}_i(k)$ is a $(N_1 + 1) \times 1$ far end transmitted symbol vector, and $\mathbf{z}_m(k)$ is a $(N_2 + 1) \times 1$ near end transmitted symbol vector, *i.e.*,

$$\mathbf{t}_i(k) = [t_i(k), t_i(k-1), \ldots, t_i(k-N_1)]^H,$$
$$\mathbf{z}_m(k) = [z_m(k), z_m(k-1), \ldots, z_m(k-N_2)]^H.$$

After echo and NEXT cancellation, the residual signal $r_j(k)$ can be expressed as

$$\begin{aligned} r_j(k) &= \sum_{i=1}^{4} \mathbf{h}_{i,j}^H \mathbf{t}_i(k) + \sum_{m=1}^{4} (\mathbf{g}_{m,j} - \mathbf{g}_{m,j}^c)^H \mathbf{z}_m(k) + n_j(k) \\ &= \sum_{i=1}^{4} \mathbf{h}_{i,j}^H \mathbf{t}_i(k) + \sum_{m=1}^{4} (\mathbf{g}_{m,j}^{res})^H \mathbf{z}_m(k) + n_j(k), \end{aligned} \tag{2.6}$$

where

$$\mathbf{g}^{c}_{m,j} = \begin{cases} [\mathbf{0}_{1\times\delta}, \bar{\mathbf{g}}^{ec}_{m,j}, \mathbf{0}_{1\times(N_2+1-\delta-N_{ec})}]^H, & m = j \\ [\mathbf{0}_{1\times\delta}, \bar{\mathbf{g}}^{nx}_{m,j}, \mathbf{0}_{1\times(N_2+1-\delta-N_{nx})}]^H, & m \neq j \end{cases}.$$

Here, $\bar{\mathbf{g}}^{ec}_{m,j}$ denotes the echo canceller with length N_{ec}, and $\bar{\mathbf{g}}^{nx}_{m,j}$ denotes the NEXT canceller with length N_{nc}. δ is the delay inserted in the echo and NEXT cancellers to indicate the start position of cancellation.

Further processing of the residual signal $r_j(k)$ includes the feedforward equalization and FEXT cancellation. The signal prior to the modulo device at channel 1 (*i.e.*, point B) is

$$u_1(k-D) = \mathbf{w}_1^H \mathbf{r}_1(k-D) - \sum_{m=2}^{4} \mathbf{f}_{m,1}^H \tilde{\mathbf{t}}_m(k), \tag{2.7}$$

where $\mathbf{w}_j$ denotes the feedforward equalizer with length L at jth channel, and $\mathbf{f}_{m,j}$ denotes the FEXT cancellers with length N_{fx}. $\mathbf{r}_j(k)$ is a $L\times 1$ symbol vector, and $\tilde{\mathbf{t}}_m(k)$ is a $N_{fx}\times 1$ symbol vector, *i.e.*,

$$\mathbf{r}_j(k) = [r_j(k), r_j(k-1), \ldots, r_j(k-L+1)]^H,$$
$$\tilde{\mathbf{t}}_m(k) = [\tilde{t}_m(k), \tilde{t}_m(k-1), \ldots, \tilde{t}_m(k-N_{fx}+1)]^H.$$

The decision error at the channel 1 is then given by

$$\begin{aligned} e_1(k) &= \hat{x}_1(k-D) - Mod\{u_1(k-D)\} \\ &= \hat{x}_1(k-D) - (u_1(k-D) - v_1(k-D)) \\ &= \hat{x}_1(k-D) + v_1(k-D) - u_1(k-D). \end{aligned} \tag{2.8}$$

In the following analysis, the estimate of disturbing signal supplied to the FEXT can-

celler is assumed to be correct, *i.e.*, $\tilde{t}_i(k) = t_i(k)$. In addition, the decision at the output of the decision device is assumed to be correct, *i.e.*, $\hat{x}_i(k) = x_i(k)$. It follows from (2.7) that (2.8) can be expressed as

$$e_1(k) = d_1(k-D) - \mathbf{w}_1^H \mathbf{r}_1(k-D) + \sum_{m=2}^{4} \mathbf{f}_{m,1}^H \mathbf{t}_m(k). \tag{2.9}$$

From (2.4), equation (2.9) can be written as

$$e_1(k) = t_1(k-D) - \mathbf{W}_1^H \mathbf{T}_1(k), \tag{2.10}$$

where,

$$\mathbf{W}_1 = \begin{bmatrix} \mathbf{w}_1 \\ \bar{\mathbf{b}}_1 \\ \mathbf{f}_{2,1} \\ \mathbf{f}_{3,1} \\ \mathbf{f}_{4,1} \end{bmatrix}, \mathbf{T}_1(k) = \begin{bmatrix} \mathbf{r}_1(k-D) \\ -\bar{\mathbf{t}}_1(k-D-1) \\ -\mathbf{t}_2(k) \\ -\mathbf{t}_3(k) \\ -\mathbf{t}_4(k) \end{bmatrix}$$

Applying the orthogonality principle [25, 84], *i.e.*, $E[\mathbf{T}_1(k)e_1(k)^H] = \mathbf{0}$, where $E[\cdot]$ denotes the expectation operator, we can get the optimal filter coefficients

$$\mathbf{W}_1^{opt}(D) = \mathbf{\Psi}_1^{-1}\theta_1 \tag{2.11}$$

where

$$\begin{aligned}
\mathbf{\Psi}_1 &= E[\mathbf{T}_1(k)\mathbf{T}_1(k)^H] \\
&= \begin{bmatrix} R_{\mathbf{r}_1\mathbf{r}_1} & R_{\mathbf{t}\mathbf{r}_1} \\ R_{\mathbf{r}_1\mathbf{t}} & R_{\mathbf{t}\mathbf{t}} \end{bmatrix}, \\
\theta_1 &= E[\mathbf{T}_1(k)t_1(k-D)^H] \\
&= \begin{bmatrix} E[\mathbf{r}_1 t_1] \\ \mathbf{0} \end{bmatrix}.
\end{aligned}$$

Substitute (2.11) into (2.10), the minimum mean square error (MMSE) at the receiver 1 is then given by

$$\begin{aligned}
J_1(D) &= E[e_1(k)e_1(k)^H] \\
&= E[|t_1(k-D)|^2] - \theta_1^H \mathbf{W}_1^{opt}(D) \\
&= E[|t_1(k)|^2] - \theta_1^H \mathbf{W}_1^{opt}(D). \qquad (2.12)
\end{aligned}$$

The MMSEs at the other three receivers can be derived in a similar way, thus the average DP-SNR over four channels is defined as

$$SNR_{avg}(D) = 10\log_{10}\left(\frac{E[|t_i(k)|^2]}{\frac{1}{4}\sum_{i=1}^{4} J_i(D)}\right). \qquad (2.13)$$

It can be seen that average DP-SNR is a function of D, which is related with the non-causal part of the FEXT. In Section 2.4.3, the heuristic search is performed to determine the optimal bulk delay under the certain complexity of the FEXT cancellers.

Assume the far end transmitted M-PAM signal $x_i(k)$ is $i.i.d$, then the output of the TH precoder is also $i.i.d$ and uniformly distributed over $[-M, M)$ [32]. Therefore, the average

energy per symbol of the effective transmitted signal $t_i(k)$ can be represented as

$$S_t = E[|t_i(k)|^2] = \frac{M^2}{M^2 - 1} S_x \tag{2.14}$$

where S_t and S_x are average energy per symbol of the signals $t_i(k)$ and $x_i(k)$, respectively. Especially, when M is large enough (*i.e.*, $M = 16$), $S_t \approx S_x$. This implies that the proposed design in Fig. 2.7 would have the similar theoretical average DP-SNR as a MMSE-DFE design in Fig. 2.4. Thus, the coefficients of the TH precoders can be set using coefficients of the corresponding DFE filters as mentioned previously.

In order to obtain the coefficients of these filters in practice, and also to track the slowly time-varying channel environment, the LMS adaptive algorithm is applied in the proposed design. During the initial training period, the known training sequences $x_i(k)$ over four pairs are transmitted with the THP turned off, *i.e.*, $t_i(k) = x_i(k)$. And then the update equations of the equalizers and FEXT cancellers can be summarized as

$$\begin{aligned}
\mathbf{w}_{j,j}(k+1) &= \mathbf{w}_{j,j}(k) + \mu_w \mathbf{r}_j(k)\epsilon_j(k), \\
\mathbf{b}_{j,j}(k+1) &= \mathbf{b}_{j,j}(k) - \mu_b \mathbf{t}_j(k-D)\epsilon_j(k), \\
\mathbf{f}_{i,j}(k+1) &= \mathbf{f}_{i,j}(k) - \mu_f \mathbf{t}_i(k)\epsilon_j(k),\ i \neq j,
\end{aligned} \tag{2.15}$$

where μ_w, μ_f, and μ_b are the step sizes for the corresponding adaptive filters. $\epsilon_j(k)$ is the update error signal obtained by $\epsilon_j(k) = x_j(k-D) - u_j^M(k-D)$. After convergence, the coefficient $\mathbf{b}_{j,j}$ is transmitted to the far end transmitter through a reliable channel to set up the THP.

During the data transmission mode, the coefficients of the TH precoder should be fixed. At the receiver side, the update equations of the FFE and FEXT cancellers switched to the

DD mode, *i.e.*,

$$\mathbf{w}_{j,j}(k+1) = \mathbf{w}_{j,j}(k) + \mu_w \mathbf{r}_j(k) e_j(k),$$
$$\mathbf{f}_{i,j}(k+1) = \mathbf{f}_{i,j}(k) - \mu_f \tilde{\mathbf{t}}_i(k) e_j(k),\ i \neq j, \tag{2.16}$$

where the update error signal is changed to $e_j(k)$.

2.4.3 Simulations and Discussions

In this section, simulation results are presented to demonstrate the effectiveness of the proposed designs in the 10GBASE-T application.

As mentioned in Section 2.4.2, the average DP-SNR is used as a performance measure to evaluate the proposed FEXT canceller. It is clear that the DP-SNR performance is affected by the design parameters such as the number of echo and NEXT taps, the number of FEXT canceller taps, the number of equalizer taps, the decision delay, and the inserted bulk delay D. To determine the optimal design parameters, the analytical result derived in equation (2.13) is investigated before simulation is performed.

Instead of performing exhaustive search, heuristic search is used to reduce the search space. In the analysis, Cat-6 UTP cable with length 100 meters is used as the channel model, which can be obtained from the IEEE 802.3an website [9].

Fig. 2.9 plots the DP-SNR performance versus the lengths of echo and NEXT cancellers. In this figure, other design parameters are fixed, *i.e.*, the lengths of the FFE and DFE are set as $L = 64$ and $N_b = 32$, respectively. The length of the FEXT canceller is $N_{fx} = 100$. The decision delay and the inserted bulk delay are $\Delta = 420$ and $D = 10$, respectively. The left side of the figure illustrates the DP-SNR performance versus the length of the echo canceller, and each curve corresponds to a fixed length of the NEXT canceller. It can be seen that for

each fixed length of the NEXT canceller, the DP-SNR performance continues to increase with the length of the echo canceller. Specifically, the DP-SNR increases dramatically when the length of the echo canceller increases from 300 to 400. After that, the DP-SNR increases gradually with the length of the echo canceller. For example, only less than 0.5 dB gain in the DP-SNR can be obtained when the length of the echo canceller increases from 400 to 600. If observing the curves corresponding to different lengths of the NEXT canceller, we find that the increase of the DP-SNR is very small when the length of the NEXT canceller is over 200. This can be seen clearly from the right side of the figure, where x-axis represents the length of the NEXT canceller and each curve corresponds to a fixed length of the echo canceller.

By considering the tradeoff between the DP-SNR performance and the complexity of the echo and NEXT cancellers, we choose the lengths of the echo and NEXT canceller to be $N_{ec} = 500$ and $N_{nc} = 300$, respectively.

Fig. 2.10 plots the the DP-SNR performance versus the lengths of the FFE filters. In this figure, $N_{ec} = 500$, $N_{nc} = 300$, $N_{fx} = 100$, $\Delta = 420$ and $D = 40$. The left side of the figure illustrates the DP-SNR performance versus the length of the FFE filter, and each curve corresponds to a fixed length of the DFE filter. It can be seen that for each fixed length of the DFE filter, the DP-SNR always has a steep increase when the length of the FFE filter increases from around 45 to 60. However, the increase of the DP-SNR is very small when the length of the FFE filter continues to increase. This can be seen clearly from the right side of the figure which illustrates the details of DP-SNR performance when the length of the FFE filter is from 50 to 100. In addition, it can be observed that the increase of the DP-SNR is very small when the length of the DFE filter increases. Thus, we choose the lengths of the FFE and DFE filters to be $L = 64$ and $N_b = 15$, respectively.

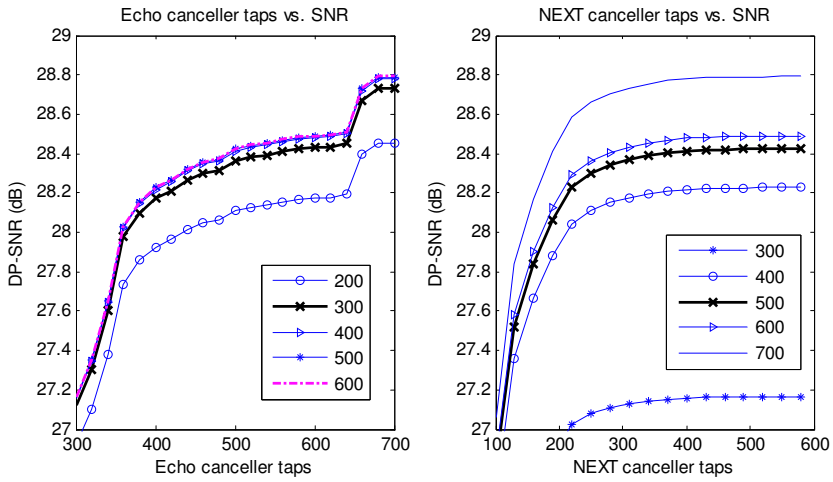


Figure 2.9: The DP-SNR performance versus the lengths of echo and NEXT cancellers

Fig. 2.11 plots the DP-SNR performance versus the decision delay, and each curve corresponds to a fixed length of the FFE filter. In this figure, $N_{ec} = 500$, $N_{nc} = 300$, $N_b = 15$, $N_{fx} = 100$ and $D = 40$. From this figure, we see that the decision delay significantly affects the DP-SNR performance. For each curve, there is an optimal decision delay. It is found that the optimal decision delay is related with the length of the FFE filter L. Specifically, larger decision delay would be expected if the length of the FFE is longer. For example, the optimal decision is 420 when $L = 64$. However, the optimal delay is 460 when $L = 100$. In our design, we choose the decision delay $\Delta = 420$, which corresponds to $L = 64$.

From the discussion in Section 2.4, it is known that the inserted bulk delay corresponds to the non-causal part of the FEXT. By increasing the bulk delay D, we can achieve more non-causal FEXT cancellation, which leads to better DP-SNR performance. In order to demonstrate the performance improvement of the proposed approach, the DP-SNR performance versus the bulk delay and the length of the FEXT canceller is plotted in Fig. 2.12. In this figure, $N_{ec} = 500$, $N_{nc} = 300$, $L = 64$, $N_b = 15$, and $\Delta = 420$. From the left side of

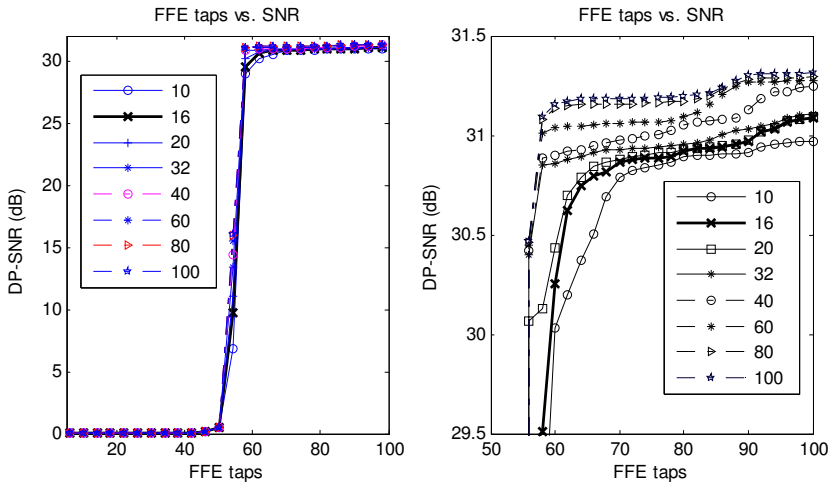


Figure 2.10: The DP-SNR performance versus the lengths of the FFE and DFE filters

this figure, it is seen that for each fixed bulk delay D, the DP-SNR performance increases with the length of the FEXT canceller. Consider the curve with $D = 0$, which corresponds to the straightforward design in Fig. 2.2, the DP-SNR curves are nearly flat when N_{fx} is over 100. This indicates that not much benefit can be obtained by increasing N_{fx} only. To further improve the DP-SNR performance, the non-causal FEXT needs to be cancelled. By increasing the bulk delay D, an obvious performance improvement can be achieved as illustrated in the figure. However, it should be noted that for a fixed length of the FEXT canceller, increasing the bulk delay can not always guarantee better DP-SNR performance. This can be illustrated in the right side of the figure, where each curve corresponds a fixed length of the FEXT canceller. Consider the curve with $N_{fx} = 32$, the DP-SNR starts at approximately 25.7 dB when $D = 0$, and continues to increase with the increase of the bulk delay till $D = 21$. After that, the DP-SNR will decrease as the bulk delay increases. Thus, for the given FEXT canceller with fixed length, we are interested in finding the optimal bulk delay D. In our design, we choose the optimal bulk delay to be $D = 40$ for $N_{fx} = 100$.

Table 2.1 summarizes the design parameters we choose according to the analysis above.

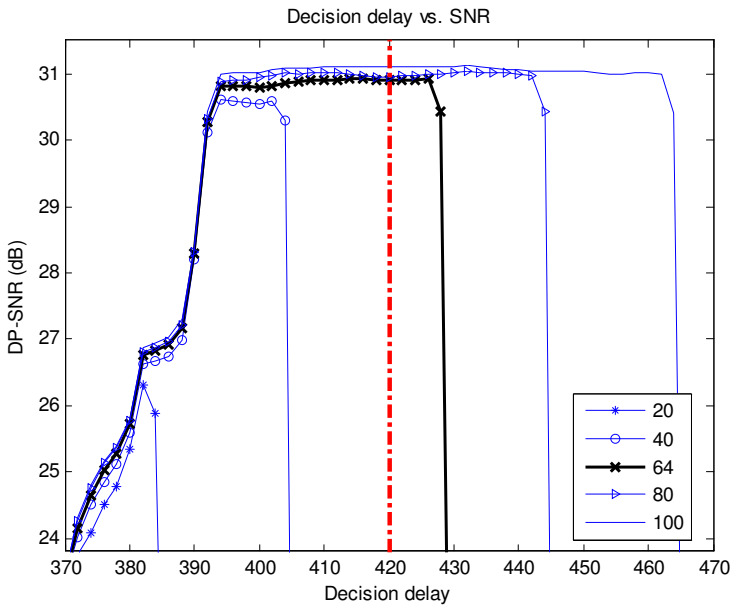


Figure 2.11: The DP-SNR performance versus the decision delay

Compared with the results in [33], the results in Table 2.1 are very similar, which demonstrates our theoretical analysis. In this table, it is noted that the TH precoder in the proposed design should have the same length as the DFE filter. Based on these parameters, the simulation can be performed under Cat-6 unshielded twisted-pair channel environment to evaluate the performance of the proposed design.

In the simulations, we consider the measured data of the Cat-6 UTP cable with three

Table 2.1: Design parameters for FEXT cancellation

Parameter	Values
Decision delay	420
Inserted bulk delay (D)	40
FFE taps	64×4
THP/DFE taps	15×4
Echo canceller taps	500×4
NEXT canceller taps	300×12
FEXT canceller taps	100×12
Total taps	7116

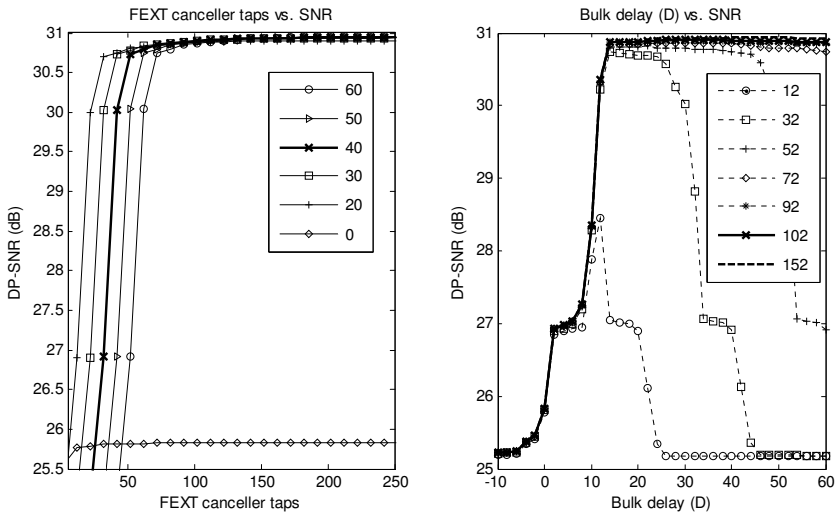


Figure 2.12: The DP-SNR performance versus the lengths of FEXT canceller and inserted bulk delay

different lengths as the channel models [9]. The transmit power is set as 5dBm, and additive white Gaussian noise (AWGN) power is set as -150dBm/Hz. In addition, a 2-PAM signal is used as the training symbol while data symbol is assumed to be 16-PAM [11]. The procedures of the simulation can be summarized as [11, 35]:

(1) At startup, the system operates in training mode. The echo and NEXT cancellers are first trained with disabling far end transmitter, near end equalizers and FEXT cancellers.

(2) Enable the equalizers and FEXT cancellers at the near end receiver when echo and NEXT cancellers are converged. Turn on the far end transmitter to send the training symbols. The equalizers and FEXT cancellers are updated using the LMS algorithm discussed in Section 2.4. Note that at the end of this stage, the coefficients of the DFE need to be sent back to the far end transmitter to set up the TH precoder [36].

(3) When the training process is done, the system switches to normal data transmission

mode, where 16-PAM data symbols are transmitted and the system is updated in DD mode. The simulated DP-SNR can be obtained after the filters have converged.

To illustrate the system operation from startup to normal data mode, a discrete eye diagram is plotted as shown in Fig. 2.13. In this figure, the discrete eye diagram is obtained by plotting the symbol values before the decision device and the simulation is performed based on Cat-6 75m channel model. Notice that the training symbols only have two possible values, *i.e.*, $\{-9, 9\}$, to keep the transmit power in the training mode the same as the transmit power in normal mode.

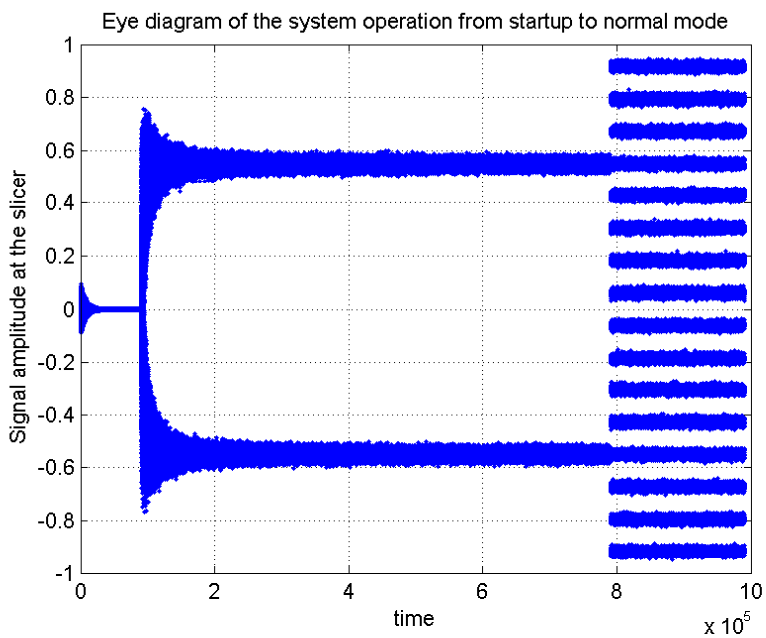


Figure 2.13: Eye diagram of the system operation from startup to normal mode, Cat-6 75m, DP-SNR=36.89dB

In order to evaluate the performance of the proposed design, we measure the DP-SNR at the normal mode. Table 2.2 gives the performance comparison for different designs. In this table, each design is evaluated based on Cat-6 measured channel models with different lengths: 100m, 75m, and 55m respectively. The analytical result (a) is also listed to be

Table 2.2: Average decision-point (DP) SNR during the steady state, AWGN=-150dBm/Hz, Bulk delay D=40, $N_{fx} = 100$

CAT-6 UTP Cable	Average DP-SNR (dB)							
	Straightforward design (Fig. 2.2)		Feedforward FEXT canceller (Fig. 2.4)		Proposed design I (Fig. 2.7)		Proposed design II (Fig. 2.8)	
	(a)	(b)	(a)	(b)	(a)	(b)	(a)	(b)
Measured 100m	25.55	19.53	30.74	21.52	30.75	30.36	30.75	28.65
Measured 75m	27.80	24.74	37.05	31.03	37.06	36.89	37.06	35.30
Measured 55m	28.87	27.51	41.37	34.34	41.38	41.23	41.38	38.78

compared with the simulation result (b) for each design.

As we can see from this table, the straightforward design in Fig. 2.2 has the worst performance in terms of analytical results and simulation results. Due to the error propagation problem, the simulation results are even worse than the analytical results. It is seen that the error propagation problem can be alleviated when DP-SNR increases at short cable channel. This is because decisions at the output of the slicer are more reliable when the decision point SNR is higher. Ideally, the decision estimate should be the same as the transmitted symbol, and the simulated DP-SNR approaches the analytical result. Similar observation can be found in the design shown in Fig. 2.4 except that this design has better performance than the straightforward design. The DP-SNR improvement is mainly because the non-causal FEXT can be properly considered in this design. However, due to the incorrect tentative decisions, the simulation results are still worse than the analytical results as we can see from Table 2.2. Specifically, in terms of the simulation results for 100m channel model, the first two designs can not meet the minimum DP-SNR requirement, 23.8dB [8]. However, in the proposed design I (Fig. 2.7) and II (Fig. 2.8), the simulation results are 30.36 and 28.65 respectively for the channel model of the same length. Thus, about 6.6dB and 4.9dB SNR margin [12] can be achieved. The significant improvement in terms of simulated DP-SNR verifies that the proposed designs can remove the non-causal FEXT effectively; in addition,

tentative decisions at the output of the FFE filter are more reliable compared with the design in Fig. 2.4. It is also found that the SNR margin is larger for shorter channel models. Compared with the proposed design I, the performance degradation in terms of simulation results in proposed design II is because an FIR filter is used to approximate an IIR filter in the FEXT canceller.

2.5 Conclusion

In this chapter, we have presented one novel approach to deal with FEXT interferences in the application of high speed Ethernet systems. The proposed approach is based on the concept of noise cancellation. By overcoming the limitations in the prior techniques on FEXT cancellation, a new feedforward FEXT canceller with TH precoding is developed. In order to speed up the receiver in the proposed design, a modified design is also developed by eliminating the feedback loops in the receiver. Compared with the traditional FEXT cancellation approaches, the proposed FEXT canceller can deal with the non-causal part of FEXT, and thus can achieve better cancellation performance. Due to the use of the TH precoders, the error propagation problem is also alleviated in practice. Detailed simulations are performed under Cat-6 UTP channel environment to verify the effectiveness of proposed designs in the 10GBASE-T application.

Chapter 3

MIMO Equalization and Cancellation for Ethernet Transmission

3.1 Introduction

In previous chapter, we have presented several approaches for FEXT cancellation in multi-pair wireline communication systems, such as 10 Gigabit Ethernet over copper (10GBASE-T). From those approaches we discussed, it is noticed that equalization is performed individually for each channel and FEXT is treated as noise to be cancelled at the receiver side. However, FEXT inherently contains information about the symbols transmitted from the remote transmitters, and thus it is better to exploit this information to facilitate signal recovery rather than simply cancel it as background noise.

In this chapter, we propose to use multi-input multi output (MIMO) equalization technique to deal with FEXT interferences. By using the MIMO technique, FEXT is treated as

signal, and a MIMO-DFE architecture is applied to jointly deal with both ISI and FEXT to improve signal to noise ration (SNR). Considering the received signal also suffers from echo and NEXT interferences, two different arrangements of echo and NEXT cancellers based on the MIMO-DFE architecture are also presented to explore better system performance. However, the MIMO-DFE structure suffers from the error propagation problem. To eliminate this problem in real applications, a new equalization scheme is proposed by combining the MIMO equalization technique with the TH precoding technique. In the proposed design, four individual TH precoders are used in the transmitter to pre-equalize each channel by removing the post-cursor ISI. At the receiver, a MIMO FFE filter is used to remove the pre-cursor ISI and also exploit the FEXT signals. In addition, a partial MIMO FFE structure rather than a partial MIMO FBE structure is used to combat the residual post-cursor FEXT. The proposed design complies with the current 10GBASE-T standard and is suitable for real applications. Compared with traditional FEXT cancellation approaches discussed in previous chapter, the proposed scheme inherits the advantage of MIMO equalization and also alleviates the error propagation, thus can achieve better performance. Moreover, unlike the general MIMO-DFE structure, the proposed architecture is easy to be pipelined for high speed applications.

The remainder of this chapter is organized as follows. Section 3.2 describes the system model of the 10GBASE-T transmission over UTP copper cables. In Section 3.3, the general MIMO equalization scheme is applied and two different arrangements of echo and NEXT cancellers are considered. Section 3.4 provides a computationally efficient method for computing the optimum settings of the MIMO equalizers and cancellers, assuming that the channel impulse response estimate and the noise characteristics are known. Section 3.5 presents the proposed equalization scheme which combines MIMO equalization technique

with TH precoding technique to deal with FEXT. In Section 3.6, methods for reducing the hardware complexity of the DSP transceiver are presented. In Section 3.7, the proposed MIMO equalization scheme is compared with the FEXT cancellation scheme proposed in previous chapter in terms of hardware complexity, convergence speed, and DP-SNR performance.

3.2 System Model

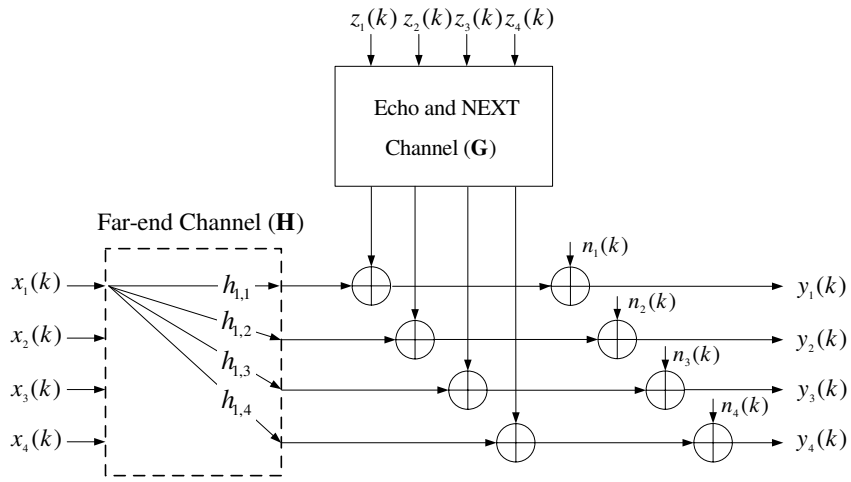


Figure 3.1: Block Diagram of the MIMO Channel for 10GBASE-T

Consider signal transmission path in Fig. 2.1, 10GBASE-T transmission system over 4 pairs of UTP can be viewed as a MIMO system, where a typical Category-6 UTP channel is modeled as two 4×4 MIMO channels as shown in Fig. 3.1. In this figure, $h_{i,j}$ denotes the MIMO channel impulse response from the ith input to the jth output with length $v+1$ and $g_{m,n}$ denotes the echo and NEXT channel impulse response from the mth input to the nth output with length $l+1$. Let x_i denote the transmitted symbol sequence from the ith far end transmitter and z_m denote the transmitted symbol sequence from the mth near end transmitter, and n_j denote background noise at the jth channel output. Then the jth

channel received symbol sequence is given by

$$y_j = \sum_{i=1}^{4} h_{i,j} \otimes x_i + \sum_{m=1}^{4} g_{m,j} \otimes z_m + n_j \tag{3.1}$$

for $j = 1, \ldots, 4$. where $\otimes$ denotes convolution.

By grouping symbols from four received channels at time k into a column vector $\mathbf{y}(k) \triangleq [y_1(k) \;\; y_2(k) \;\; y_3(k) \;\; y_4(k)]^T$, (3.1) can be expressed as

$$\mathbf{y}(k) = \sum_{\tau=0}^{v} H_\tau \mathbf{x}(k-\tau) + \sum_{p=0}^{l} G_p \mathbf{z}(k-p) + \mathbf{n}(k) \tag{3.2}$$

where H_τ and G_p represent 4×4 τth far end channel coefficient matrix and pth near end channel coefficient matrix, respectively. The signals $\mathbf{x}(k-\tau)$ and $\mathbf{z}(k-\tau)$ correspond to far end transmitted column vector and near end transmitted column vector at time index $k-\tau$, respectively. By stacking N_f successive channel output vector samples, (3.2) can be expressed as in matrix form

$$\begin{aligned} \mathbf{y}(k+N_f-1:k) = \mathbf{H} \cdot \mathbf{x}(k+N_f-1:k-v) \\ + \mathbf{G} \cdot \mathbf{z}(k+N_f-1:k-l) \\ + \mathbf{n}(k+N_f-1:k). \end{aligned} \tag{3.3}$$

where $\mathbf{y}(k+N_f-1:k)$ is a column vector with dimension $4N_f \times 1$, and it is defined as

$$\mathbf{y}(k+N_f-1:k) \triangleq [\mathbf{y}^T(k+N_f-1) \;\; \ldots \;\; \mathbf{y}^T(k)]^T. \tag{3.4}$$

Similar definitions are applied to $\mathbf{x}(k+N_f-1:k-v)$, $\mathbf{z}(k+N_f-1:k-l)$, and $\mathbf{n}(k+N_f-1:k)$. In addition, matrix $\mathbf{H}$ and matrix $\mathbf{G}$ are both block Toeplitz matrices with dimension

$N_f \times (N_f + v)$ and $N_f \times (N_f + l)$ respectively.

$$\mathbf{H} = \begin{bmatrix} H_0 & H_1 & \dots & H_v & 0 & \dots & 0 \\ 0 & H_0 & H_1 & \dots & H_v & \dots & 0 \\ \vdots & & \ddots & & & \ddots & \\ 0 & \dots & 0 & H_0 & H_1 & \dots & H_v \end{bmatrix}, \tag{3.5}$$

$$\mathbf{G} = \begin{bmatrix} G_0 & G_1 & \dots & G_l & 0 & \dots & 0 \\ 0 & G_0 & G_1 & \dots & G_l & \dots & 0 \\ \vdots & & \ddots & & & \ddots & \\ 0 & \dots & 0 & G_0 & G_1 & \dots & G_l \end{bmatrix}. \tag{3.6}$$

3.3 MIMO Equalization and Cancellation in Ethernet Systems

In this section, MIMO equalization technique is proposed to deal with FEXT interferences. Information contained in the FEXT crosstalk is exploited by the MMSE based MIMO-DFE structure to improve system performance in terms of decision point SNR. To achieve better system performance, two different arrangement of the echo and NEXT cancellers based on this MIMO equalization architecture are investigated. Simulation results are also provided to demonstrate the advantage of the MIMO equalization technique.

3.3.1 Joint MIMO-DFE Equalization and Cancellation

Fig. 3.2 shows the block diagram of the proposed joint MIMO-DFE equalization and cancellation scheme. In this figure, the received signal $y_i(k)_{i=1,2,3,4}$ first goes through a MIMO FFE which consists of 16 FFEs as shown in Fig. 3.3, to combat pre-cursor ISI and exploit

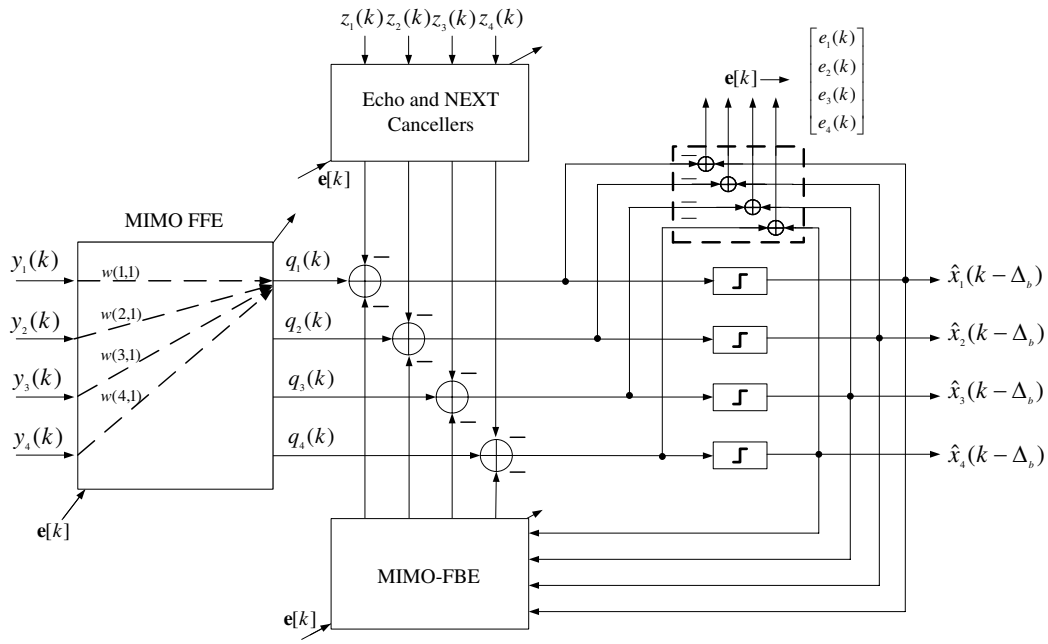


Figure 3.2: Joint MIMO-DFE equalization & echo and NEXT cancellation

far end signal transmitted from FEXT channels. After FFE, echo and NEXT cancellers are used to eliminate echo and NEXT interferences. These cancellers are implemented using adaptive FIR filters with the inputs $z_i(k)_{i=1,2,3,4}$. The architecture of the NEXT cancellers is similar to echo cancellers except that the input signals to the NEXT cancellers are from the other three transmitted signals at the near end. Instead of using FEXT cancellers, a MIMO-FBE structure, which has a similar architecture as shown in Fig. 3.3, is used to deal with post-cursor FEXT. Different from FEXT cancellers discussed in previous chapter, the length of each feedback filter in the MIMO-DFE structure is much shorter.

Let N_f, N_b, N_p be the lengths of the feedforward filter matrix $\mathbf{W}$, feedback filter matrix $\mathbf{B}$, and echo and NEXT cancellation filter matrix $\mathbf{P}$, respectively. If we further assume echo and NEXT cancellers have different numbers of taps with N_e and N_x respectively, then $N_p \triangleq max(N_e, N_x)$. The objective is to choose $\mathbf{W}$, $\mathbf{B}$, and $\mathbf{P}$ to minimize mean square error (MSE) over the four channels.

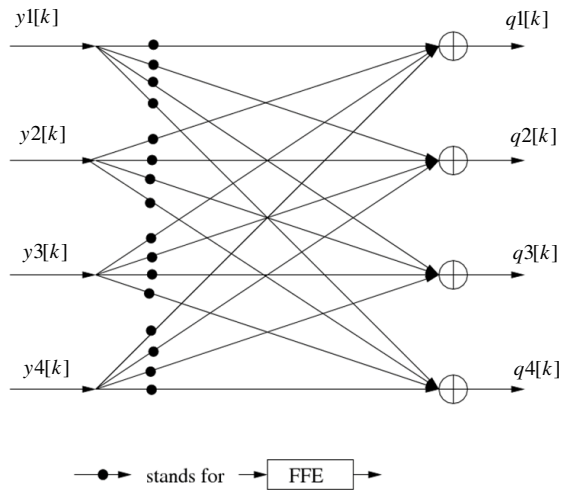


Figure 3.3: Internal structure of a MIMO FFE filter

From Fig. 3.2, the error vector at time k of the four channels can be represented by

$$\begin{aligned}\mathbf{e}(k) = &\tilde{\mathbf{B}}^H \mathbf{x}(k+N_f-1:k-v) + \tilde{\mathbf{P}}^H \mathbf{z}(k+N_f-1:k-l) \\ &- \mathbf{W}^H \mathbf{y}(k+N_f-1:k)\end{aligned} \tag{3.7}$$

where,

$$\mathbf{W}^H = \begin{bmatrix} W_0^H & W_1^H & \dots & W_{N_f-1}^H \end{bmatrix},$$

$$W_i = \begin{bmatrix} w_i^{(1,1)} & w_i^{(1,2)} & w_i^{(1,3)} & w_i^{(1,4)} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ w_i^{(4,1)} & \dots & \dots & w_i^{(4,4)} \end{bmatrix}$$

$$
\begin{aligned}
\tilde{\mathbf{B}}^H &= \begin{bmatrix} \mathbf{0}_{1\times\Delta_b} & B_0^H & B_1^H & \dots & B_{N_b}^H & \mathbf{0}_{1\times s_1} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{0}_{1\times\Delta_b} & \vec{\mathbf{B}}^H & \mathbf{0}_{1\times s_1} \end{bmatrix}, \\
\tilde{\mathbf{P}}^H &= \begin{bmatrix} \mathbf{0}_{1\times\Delta_p} & P_1^H & \dots & P_{N_p}^H & \mathbf{0}_{1\times s_2} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{0}_{1\times\Delta_p} & \vec{\mathbf{P}}^H & \mathbf{0}_{1\times s_2} \end{bmatrix},
\end{aligned}
$$

where B_i and P_i are 4×4 blocks similar to W_i, and $\mathbf{0}$ is a 4×4 zero matrix. We also define $s_1 = N_f + v - N_b - \Delta_b - 1$ and $s_2 = N_f + l - N_p - \Delta_p$ with the decision delays Δ_b and Δ_p, respectively.

Now the mean square error minimization problem can be formulated as

$$
\begin{aligned}
min. \quad & E[\|\mathbf{e}(k)\|^2] = trace(\mathbf{R}_{ee}) \\
s.t. \quad & \begin{bmatrix} \vec{\mathbf{B}}^H & \vec{\mathbf{P}}^H \end{bmatrix} \mathbf{\Phi} = \mathbf{I}
\end{aligned} \tag{3.8}
$$

where $\mathbf{\Phi^H} = \begin{bmatrix} \mathbf{I} & \mathbf{0}_{1\times(N_b+N_p)} \end{bmatrix}$, and $\mathbf{I}$ is a 4×4 indentity matrix. Solving this optimization problem, we get

$$
\begin{bmatrix} \vec{\mathbf{B}}_{opt} \\ \vec{\mathbf{P}}_{opt} \end{bmatrix} = \mathbf{R_\Delta^{-1}\Phi(\Phi^H R_\Delta^{-1}\Phi)^{-1}} \tag{3.9}
$$

$$
\mathbf{W}_{opt}^H = \begin{bmatrix} \tilde{\mathbf{B}}_{opt}^H & \tilde{\mathbf{P}}_{opt}^H \end{bmatrix} \begin{bmatrix} R_{xy} \\ R_{zy} \end{bmatrix} R_{yy}^{-1} \tag{3.10}
$$

$$\begin{aligned}\mathbf{R}_{ee,min} &= E[\mathbf{e}(k)\mathbf{e}^H(k)] \\ &= \begin{bmatrix} \vec{\mathbf{B}}_{opt}^H & \vec{\mathbf{P}}_{opt}^H \end{bmatrix} \mathbf{R}_\Delta \begin{bmatrix} \vec{\mathbf{B}}_{opt} & \vec{\mathbf{P}}_{opt} \end{bmatrix} \\ &= (\mathbf{\Phi^H R_\Delta^{-1} \Phi})^{-1} \end{aligned} \tag{3.11}$$

$$MSE_{min} = \frac{1}{4} trace(\mathbf{R}_{ee,min}) \tag{3.12}$$

where $\mathbf{R}_\Delta = \mathbf{Q}^H \mathbf{R} \mathbf{Q}$, and $\mathbf{Q}$ is a constant matrix.

$$\mathbf{R} = \begin{bmatrix} R_{xx} - R_{xy}R_{yy}^{-1}R_{yx} & -R_{xy}R_{yy}^{-1}R_{yx} \\ -R_{zy}R_{yy}^{-1}R_{yz} & R_{zz} - R_{zy}R_{yy}^{-1}R_{yz} \end{bmatrix}.$$

Since the matrix $\mathbf{R_\Delta}$ in (3.9) has dimensions $4(N_b + N_p + 1) \times 4(N_b + N_p + 1)$, Cholesky factorization would generally need $O[64(N_b+N_p+1)^3]$ operations. By using fast algorithms in [25], these computations can be performed efficiently in $O[16(N_b + N_p + 1)^2]$ operations. However, we note that the computational complexity required for fast Cholesky factorization depends on the total number of taps in feedback equalizers and echo and NEXT cancellers. Hence, for cancellers with a large number of taps, the computation of the optimal tap coefficients will be expensive. To deal with this problem, we provide an efficient method to calculate those optimal settings of the MIMO equalizer and cancellers in Section 3.4.

3.3.2 Separate MIMO-DFE Equalization and Cancellation

In Fig. 3.2, the echo and NEXT cancellers and MIMO-DFE are jointly adapted to minimize mean squared error $E(\mathbf{e}^2)$, where echo and NEXT cancellers are performed after FFE. One problem associated with this structure is that the FFE filtering will affect echo and NEXT channel characteristics, *i.e.*, channel length or amplitude, especially when FFE filters are

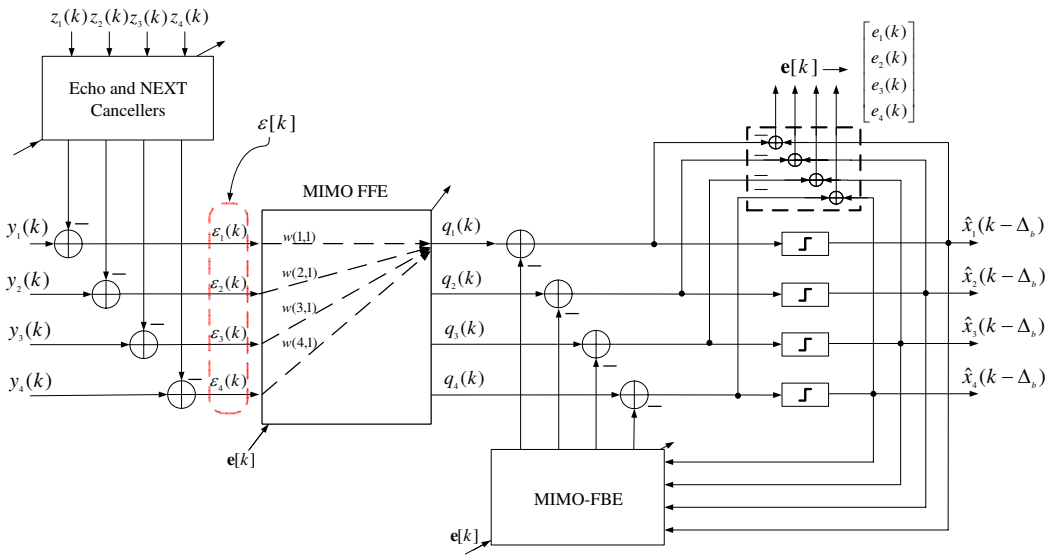


Figure 3.4: Separate MIMO-DFE equalization & echo and NEXT cancellation

very long, which will result in long echo and NEXT cancellers. To solve this problem, consider the structure in Fig. 3.4, where echo and NEXT cancellers are implemented before FFE. In this arrangement, cancellers and the MIMO-DFE are independently adapted to minimize $E(\epsilon^2)$ and $E(\mathbf{e^2})$. However, for given echo and NEXT cancellers, the optimal echo and NEXT cancellers in the sense of minimizing $E(\epsilon^2)$ may not be the one that also minimizes $E(\mathbf{e^2})$. In other words, increasing the length of the echo and NEXT cancellers in this case does not necessarily reduce $E(\mathbf{e}^2)$ as desired. Through simulations, it was found that separate minimization structure has $0.1 \sim 0.4$dB gain over jointly adapted structure with same canceller length under different channel models. Hence, it can be stated that in UTP cable channel, with the same echo and NEXT complexity, the jointly adapted structure is not necessarily superior to the separately adapted structure.

3.3.3 Simulation Results

In this section, we apply the proposed approaches to the CAT-6 UTP channels with different cable length, *i.e.*, 100m and 55m. All these channel models can be obtained from IEEE 802.3an 10GBASE-T Study Group [9].

In order to present a fair comparison, the parameters used in the simulation for traditional equalization scheme as shown in Fig. 2.2 and proposed MIMO equalization scheme are given in Table 3.1.

Table 3.1: Parameter settings

	Traditional equalization (SISO)	MIMO equalization (MIMO-DFE)
FFE taps	64	64×4
DFE taps	32	32×4
FEXT taps	200×3	0
Echo taps	500	500
NEXT taps	400×3	400×3
Total taps	2396×4=9584	2084×4=8336
Modulation	PAM-16	
AWGN	−150dBm	
TX Power	5dBm	

Table 3.2 compares the performance of the traditional equalization scheme based on FEXT cancellation and the proposed MIMO equalization based on MIMO-DFE structure. We see the proposed MIMO equalization technique has an obvious SNR gain over traditional scheme with less complexity for different channel modes. Especially for short cable, by using the proposed scheme, we can have much more SNR gain over the traditional approach. This is because the FEXT channel attenuation is proportional to the cable length. FEXT signals in short cables will be stronger than that in long cables. In this sense, MIMO technology will get more benefit from the FEXT signal.

Another interesting result is that separate adapted structure has 0.1∼0.4dB gain over

jointly adapted structure with same canceller complexity in different channel models. Hence, it can be stated that in UTP channel, with the same echo and NEXT complexity, the jointly adapted structure is not necessarily superior to the separately adapted structure.

Table 3.2: Decision point SNR for different schemes (dB)

CAT6	Traditional SISO equalization scheme	MIMO equalization with DFE	
		Joint equalization and cancellation	Separate equalization and cancellation
Measured 100m	20.9	29.5	29.9
Measured 75m	25.7	37.0	37.1
Measured 55m	28.1	41.2	41.4

3.4 Computation Method for MIMO equalizers and Cancellers

3.4.1 Background

To set the optimal tap coefficients of the equalizers and cancellers in a MIMO system, one straightforward approach is to recursively compute the tap coefficients using adaptive algorithms such as least mean square (LMS). However, the recursive LMS adaptation of these equalizers and cancellers may take millions of iterations to converge to the optimal solution on a typical Category-6 UTP channel. Much faster convergence can be achieved using RLS algorithm, but the higher computational cost is prohibitive in real applications. Another indirect approach is to compute the tap coefficients, in MMSE sense, based on the channel estimates [23]. In this approach, the knowledge of the channel impulse response, as well as the noise characteristics are required. In addition, a computationally efficient approach is also needed to compute the optimal settings of the MIMO equalizers and cancellers.

Previous studies on efficiently computing the tap coefficients of the equalizers are mostly

based on the traditional finite-length MMSE-DFE structure. In [23], Cholesky factorization is applied to carry out the involved matrix inversion efficiently. By exploiting the structured matrices, a generalized Schur algorithm for fast Cholesky matrices decomposition was provided in [24] to reduce the computational complexity in both feedback and feedforward filters computation. Recently, another efficient approach is proposed to achieve faster computation by identifying the relationship between the feed-forward equalizer computation and fast recursive least squares (RLS) adaptive algorithms, and treating the feedback equalizers computation as a convolution operation [26].

Although these methods can be easily extended to the general MMSE-DFE tap computation in MIMO channels [25, 27], they may not always be computationally efficient for computing the optimal coefficients of the MIMO equalizers and cancellers in the cases where the cancellers have a larger number of taps than the feed-forward equalizers. By using Al-Dhahir's method [25], the inversion of an embedded correlation matrix will be computationally intensive since the size of the matrix is related to the maximum number of taps in echo cancellers and NEXT cancellers, as shown in the following section. In addition, the number of taps in echo cancellers and NEXT cancellers need not be same in real applications. Thus, applying the efficient method in [27] is also not straightforward.

In the following, we present a new computationally efficient approach for computing the optimum settings of the MIMO equalizers and cancellers for 10Gbase-T, assuming that the channel impulse response estimate and the noise characteristics are known. This new method is exact and applicable to cases where echo and NEXT cancellers have different lengths. Compared with Al-Dhahir's method, it has lower computational complexity.

3.4.2 Fast Computation Method

In this section, a computationally efficient approach for computing the optimum settings of the MIMO equalizers and cancellers is presented. The basic idea is to reduce the dimension of matrix required for Cholesky factorization. This can be achieved by decomposing the original optimization problem in (3.8) into four separate/independent optimization problems, each with smaller size. Solving each optimization problem is computationally efficient. Thus, the overall computational complexity will be significantly reduced.

The minimization problem in (3.8) can be written as:

$$\begin{aligned} min. \quad & E[\|\mathbf{e}(k)\|^2] = trace(\mathbf{R}_{ee}) \\ = \quad & E[e_1(k)^2] + E[e_2(k)^2] + E[e_3(k)^2] + E[e_4(k)^2] \\ s.t. \quad & \begin{bmatrix} \vec{\mathbf{B}}^H & \vec{\mathbf{P}}^H \end{bmatrix} \mathbf{\Phi} = \mathbf{I} \end{aligned} \tag{3.13}$$

where $e_i(k),\ (i = 1, 2, 3, 4)$ is the error signal for each corresponding channel. To show that this problem is equivalent to minimizing each of the mean square error for each channel, we rewrite (3.7) as:

$$\begin{aligned} \mathbf{e}(k) &= \begin{bmatrix} \tilde{\mathbf{B}}^H & \tilde{\mathbf{P}}^H \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}(k+N_f-1:k-v) \\ \mathbf{z}(k+N_f-1:k-l) \end{bmatrix} - \mathbf{W}^H\mathbf{y}(k+N_f-1:k) \\ &= \mathbf{A}^H\mathbf{s} - \mathbf{W}^H\mathbf{y}(k+N_f-1:k) \end{aligned} \tag{3.14}$$

where $\mathbf{A}^H = \begin{bmatrix} \tilde{\mathbf{B}}^H & \tilde{\mathbf{P}}^H \end{bmatrix}$, and $\mathbf{s} = \begin{bmatrix} \mathbf{x}(k+N_f-1:k-v) \\ \mathbf{z}(k+N_f-1:k-l) \end{bmatrix}$. Each element of $\mathbf{e}(k)$ is

given by

$$e_1(k) = \mathbf{A}^{(1)H}\mathbf{s} - \mathbf{W}^{(1)H}\mathbf{y}$$

$$e_2(k) = \mathbf{A}^{(2)H}\mathbf{s} - \mathbf{W}^{(2)H}\mathbf{y}$$

$$e_3(k) = \mathbf{A}^{(3)H}\mathbf{s} - \mathbf{W}^{(3)H}\mathbf{y}$$

$$e_4(k) = \mathbf{A}^{(4)H}\mathbf{s} - \mathbf{W}^{(4)H}\mathbf{y}$$

where $\mathbf{A}^{(i)H}$ and $\mathbf{W}^{(i)H}$ represent the ith row of $\mathbf{A}^H$ and $\mathbf{W}^H$, respectively. Then, we have

$$\begin{aligned} E[e_i e_i^H] \;=\; & (\mathbf{W}^{(i)H} - \mathbf{A}^{(i)H} R_{sy} R_{yy}^{-1}) R_{yy} (\mathbf{W}^{(i)H} - \mathbf{A}^{(i)H} R_{sy} R_{yy}^{-1})^H \\ & + \mathbf{A}^{(i)H}(R_{ss} - R_{sy} R_{yy}^{-1} R_{ys})\mathbf{A}^{(i)}. \end{aligned}$$

The goal is to minimize the total MSE, $E[\|\mathbf{e}(k)\|^2]$, with respect to $\mathbf{A}^{(i)}$ and $\mathbf{W}^{(i)}, i = 1, 2, 3, 4$. Since there is no constraint on $\mathbf{W}$, we can first minimize the total MSE with respect to $\mathbf{W}^{(1)}$, assuming $\mathbf{A}^{(1)}$ is fixed. It is easy to see that $E[e_i e_i^H](i = 2, 3, 4)$ is independent of $\mathbf{W}^{(1)}$, and minimizing the total MSE with respect to $\mathbf{W}^{(1)}$ is equivalent to minimizing the $E[e_1 e_1^H]$ separately. Thus, we get the optimal coefficients of the feed-forward filters corresponding to channel 1 as follows:

$$\mathbf{W}_{opt}^{(1)H} = \mathbf{A}^{(1)H} R_{sy} R_{yy}^{-1} \tag{3.15}$$

then the minimum $E[e_1 e_1^H]$ (with respect to $\mathbf{W}^{(1)}$) turns out to be

$$E[e_1 e_1^H] = \mathbf{A}^{(1)H}(R_{ss} - R_{sy} R_{yy}^{-1} R_{ys})\mathbf{A}^{(1)} \tag{3.16}$$

similarly, we have

$$E[e_2 e_2^H] = \mathbf{A}^{(2)H}(R_{ss} - R_{sy}R_{yy}^{-1}R_{ys})\mathbf{A}^{(2)} \tag{3.17}$$

$$E[e_3 e_3^H] = \mathbf{A}^{(3)H}(R_{ss} - R_{sy}R_{yy}^{-1}R_{ys})\mathbf{A}^{(3)} \tag{3.18}$$

$$E[e_4 e_4^H] = \mathbf{A}^{(4)H}(R_{ss} - R_{sy}R_{yy}^{-1}R_{ys})\mathbf{A}^{(4)} \tag{3.19}$$

Now we minimize the total MSE with respect to $\mathbf{A}^{(i)}$. As before, minimizing the total MSE with respect to $\mathbf{A}^{(i)}$ is equivalent to minimizing each $E[e_i e_i^H]$ only. Therefore, we can solve the original problem by minimizing each of the $E[e_i e_i^H]$.

For minimizing each of the $E[e_i e_i^H]$, the problem size is greatly reduced. As an example, we solve the optimization problem corresponding to $E[e_1 e_1^H]$ minimization. We first write $e_1(k)$ as

$$\begin{aligned} e_1(k) &= x_1(n - \Delta_{b1}) - \left(\mathbf{W1}^H\mathbf{y1} - \mathbf{B1}^H\mathbf{x1} - \mathbf{P1}^H\mathbf{z1}\right) \\ &= x_1(n - \Delta_{b1}) - \left(\mathbf{W1}^H\mathbf{y1} - \mathbf{A1}^H\mathbf{s1}\right) \end{aligned} \tag{3.20}$$

where Δ_{b1} is the decision delay for channel 1, and **W1**, **B1** and **P1** are all column vectors,

$$
\begin{aligned}
\mathbf{W1}^H &\triangleq \begin{bmatrix} W^{(1,1)H} & W^{(2,1)H} & W^{(3,1)H} & W^{(4,1)H} \end{bmatrix} \\
W^{(i,j)} &\triangleq \begin{bmatrix} w_0^{(i,j)} & w_1^{(i,j)} & \dots & w_{N_f-1}^{(i,j)} \end{bmatrix}^H \\
\mathbf{B1}^H &\triangleq \begin{bmatrix} B^{(1,1)H} & B^{(2,1)H} & B^{(3,1)H} & B^{(4,1)H} \end{bmatrix} \\
B^{(i,j)} &\triangleq \begin{bmatrix} b_1^{(i,j)} & b_2^{(i,j)} & \dots & b_{N_b}^{(i,j)} \end{bmatrix}^H \\
\mathbf{P1}^H &\triangleq \begin{bmatrix} P^{(1,1)H} & P^{(2,1)H} & P^{(3,1)H} & P^{(4,1)H} \end{bmatrix} \\
P^{(i,i)} &\triangleq \begin{bmatrix} p_1^{(i,i)} & p_2^{(i,i)} & \dots & p_{N_e}^{(i,i)} \end{bmatrix}^H \\
P^{(i,j)} &\triangleq \begin{bmatrix} p_1^{(i,j)} & p_2^{(i,j)} & \dots & p_{N_x}^{(i,j)} \end{bmatrix}^H .
\end{aligned}
$$

y1, **x1** and **z1** are also column vectors,

$$
\begin{aligned}
\mathbf{y1}^H &\triangleq \begin{bmatrix} y^{(1,1)H} & y^{(2,1)H} & y^{(3,1)H} & y^{(4,1)H} \end{bmatrix} \\
\mathbf{x1}^H &\triangleq \begin{bmatrix} x^{(1,1)H} & x^{(2,1)H} & x^{(3,1)H} & x^{(4,1)H} \end{bmatrix} \\
\mathbf{z1}^H &\triangleq \begin{bmatrix} z^{(1,1)H} & z^{(2,1)H} & z^{(3,1)H} & z^{(4,1)H} \end{bmatrix} \\
x^{(i,j)} &\triangleq \begin{bmatrix} x^{(i,j)}_{n-\Delta_{b1}-1} \\ x^{(i,j)}_{n-\Delta_{b1}-2} \\ \vdots \\ x^{(i,j)}_{n-\Delta_{b1}-N_b} \end{bmatrix}, y^{(i,j)} \triangleq \begin{bmatrix} y^{(i,j)}_{n} \\ y^{(i,j)}_{n-1} \\ \vdots \\ y^{(i,j)}_{n-N_f+1} \end{bmatrix} \\
z^{(i,i)} &\triangleq \begin{bmatrix} z^{(i,i)}_{n-\Delta_{e1}-1} \\ z^{(i,i)}_{n-\Delta_{e1}-2} \\ \vdots \\ z^{(i,i)}_{n-\Delta_{e1}-N_e} \end{bmatrix}, z^{(i,j)} \triangleq \begin{bmatrix} z^{(i,j)}_{n-\Delta_{x1}-1} \\ z^{(i,j)}_{n-\Delta_{x1}-2} \\ \dots \\ z^{(i,j)}_{n-\Delta_{x1}-N_x} \end{bmatrix}
\end{aligned}
$$

To minimize $E[e_1 e_1^H]$ from (3.20), the well-known solution is given by [84],

$$
\begin{bmatrix} \mathbf{W1} \\ \mathbf{B1} \\ \mathbf{P1} \end{bmatrix} = \mathbf{\Phi^{-1}}\theta \tag{3.21}
$$

where

$$\begin{aligned}
\mathbf{\Phi} &= E\left(\begin{bmatrix} \mathbf{y1} \\ \mathbf{x1} \\ \mathbf{z1} \end{bmatrix} \begin{bmatrix} \mathbf{y1} & \mathbf{x1} & \mathbf{z1} \end{bmatrix}^H\right) \\
\theta &= E\left(\begin{bmatrix} \mathbf{y1} \\ \mathbf{x1} \\ \mathbf{z1} \end{bmatrix} x_1(k-\Delta_{b1})\right).
\end{aligned} \tag{3.22}$$

Finally, the optimum taps of the equalizers and cancellers corresponding to the channel 1 can be computed as,

$$\begin{bmatrix} \mathbf{W1} \\ \mathbf{B1} \\ \mathbf{P1} \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi_1^{-1}}\theta_1 \\ -\mathbf{R^H_{y_1x_1}W1} \\ -\mathbf{R^H_{y_1z_1}W1} \end{bmatrix} \tag{3.23}$$

where

$$\theta_1 = E\left[\mathbf{y1}x_1(k-\Delta_{b1})\right], \tag{3.24}$$

and

$$\mathbf{\Phi_1} = \mathbf{R_{y1y1}} - \mathbf{R_{y1z1}R^H_{y1z1}} - \mathbf{R_{y1x1}R^H_{y1x1}}. \tag{3.25}$$

Here, we assume the input vectors $\mathbf{x}$ and $\mathbf{z}$ are $i.i.d$ with unit power in our derivation. We see that $\mathbf{\Phi_1}$ has dimension $4N_f \times 4N_f$, which only depends on the length of the feed-forward filters. Thus, the size of the matrix required for Cholesky factorization is reduced

compared with $4(N_b+N_p+1)\times 4(N_b+N_p+1)$ in (3.9). We also note that no multiplications are needed to compute (3.24) and form the the block Toeplitz matrices $\mathbf{R_{y1z1}}$ and $\mathbf{R_{y1x1}}$ in (3.25). Therefore, computing (3.23) requires a total number of

$$2(4N_f)^2 + 4N_f(N_e + 3N_x) + 16N_f(N_b) + 16N_f^2(v-1)$$
$$+ 4N_f^2(N_e - 1) + 12N_f^2(N_x - 1) + 16N_f^2(N_b - 1)$$

multiplications. Assuming that the complexity is dominated by the multiplication operations, Fig. 3.5 shows the complexity comparison between Al-Dhahir's and the proposed method. Fig. 3.6 shows the computational reduction due to the proposed approach compared with Al-Dhahir's method. It can be observed that the proposed method can achieve substantial computational savings of 63.8% (where, N_f is the length of feed-forward MIMO equalizer; $N_b = 32$ is the length of the feedback matrix filter; $N_e = 500$ is the length of Echo cancellers; $N_x = 400$ is the length of NEXT cancellers; and direct MIMO channel & crosstalk channel have lengths $v = 1000$ & $l = 1000$, respectively).

3.5 MIMO equalization incorporated with TH precoding

In previous Sections, we have demonstrated the advantage of MIMO equalization technique in the application of 10GBASE-T. However, one problem associated with MMSE-DFE structure is catastrophic error propagation, which degrades system performance significantly when input SNR is very low. In addition, the feedback loops inside the MIMO-DFE architecture limit the high speed implementation of the receiver.

In this section, a new equalization scheme is proposed to deal with FEXT by combining MIMO equalization technique and TH precoding technique. Compared with tradi-

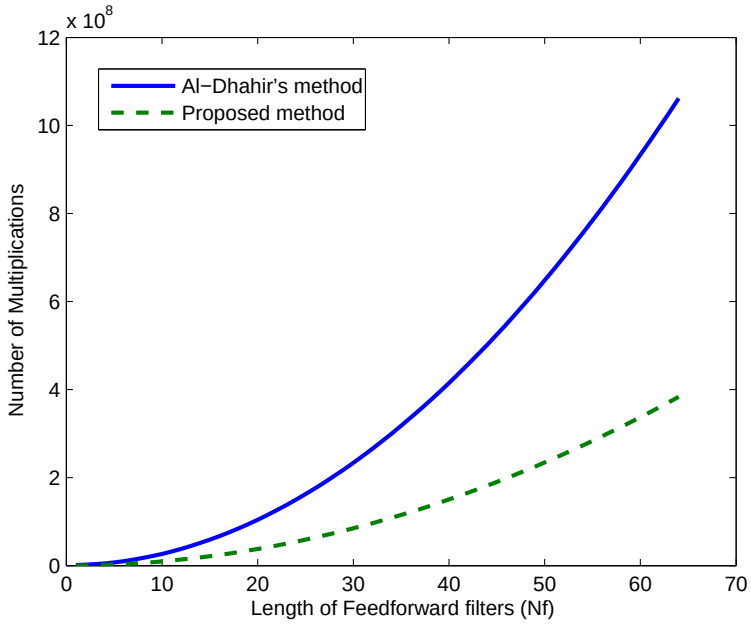


Figure 3.5: Number of Multiplication Comparison

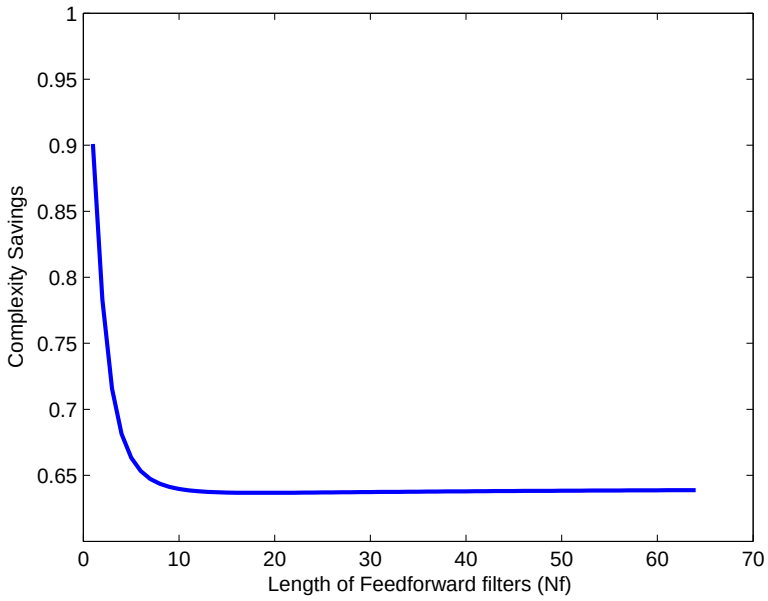


Figure 3.6: Complexity Reduction Comparison

tional FEXT cancellation approaches, the proposed scheme inherits the advantage of MIMO equalization and also alleviates the error propagation, thus can achieve better performance. However, different from traditional MIMO-DFE structures presented in Section 3.3, the proposed scheme complies with the current 10GBASE-T standard and is also suitable for high speed implementation by completely removing the feedback loops in receivers.

3.5.1 A Straightforward MIMO-THP Structure

By combining the TH precoding technique with the general MIMO equalization technique, we obtain a straightforward design as shown in Fig. 3.7, where the feedback part of the DFE structure is implemented at the transmitter [42]. In this figure, $\mathbf{x}(k)$ denotes far end transmitted vector. Due to the TH precoding, the effective transmitted vectors going through the physical channel turn to be $\mathbf{t}(k)$ and $\mathbf{z}(k)$ from far end and near end transmitters, respectively. It should be noted that $\mathbf{B}(z)$ represents a MIMO-TH precoder, which is implemented at the transmitter side to pre-equalize the transmitted symbols. The detailed MIMO-TH precoder structure for one pair is shown in Fig. 3.8. It consists of four feedback filters with the input of the effective transmitted signals $\mathbf{t}(k)$ and a nonlinear modulo device to limit the output dynamic range. At the receiver side, different from MIMO-DFE structure discussed in previous Section, MIMO-FBE is not needed any more due to the TH precoding. To recover the transmitted signals, a modulo operation is needed before the decision device.

However, this straightforward design is not compatible with the current 10GBASE-T standard [11], where only four separate TH precoders are required for each channel. Besides, the high speed design of the MIMO-TH precoder is still a challenging problem. The existing high speed design of the TH precoder in [29] is only based on single-input single-output (SISO) channel.

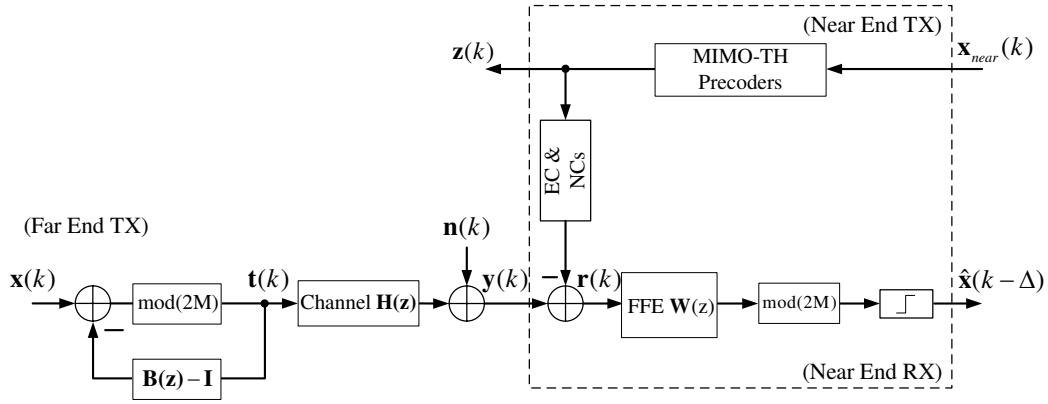


Figure 3.7: A straightforward design of MIMO structure combined with TH precoding

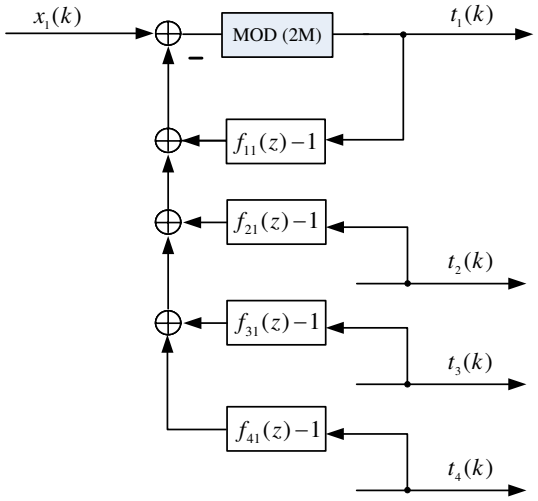


Figure 3.8: Internal structure of a MIMO-TH precoder for one channel

3.5.2 Proposed MIMO Equalization Incorporated With TH Precoding

To develop a transceiver architecture complying with the IEEE 802.3an standard while maintaining the same system performance as the design in Fig. 3.7, we propose to retain four SISO TH precoders at the transmitter side and implement the cross-channel pre-equalizers at the receiver side. The block diagram of the proposed architecture is shown in Fig. 3.9. In this figure, four individual TH-precoders are used at the transmitter to pre-equalize each channel by removing the post-cursor ISI. At the receiver, a MIMO FFE

filter is used to remove pre-cursor ISI and exploit the FEXT signals. In addition, a partial MIMO-FBE filter is used to combat the residual post-cursor FEXT. The detailed structure of the partial MIMO-FBE is shown in Fig. 3.10. It contains 12 cross feedback filters, four nonlinear modulo devices and slicers.

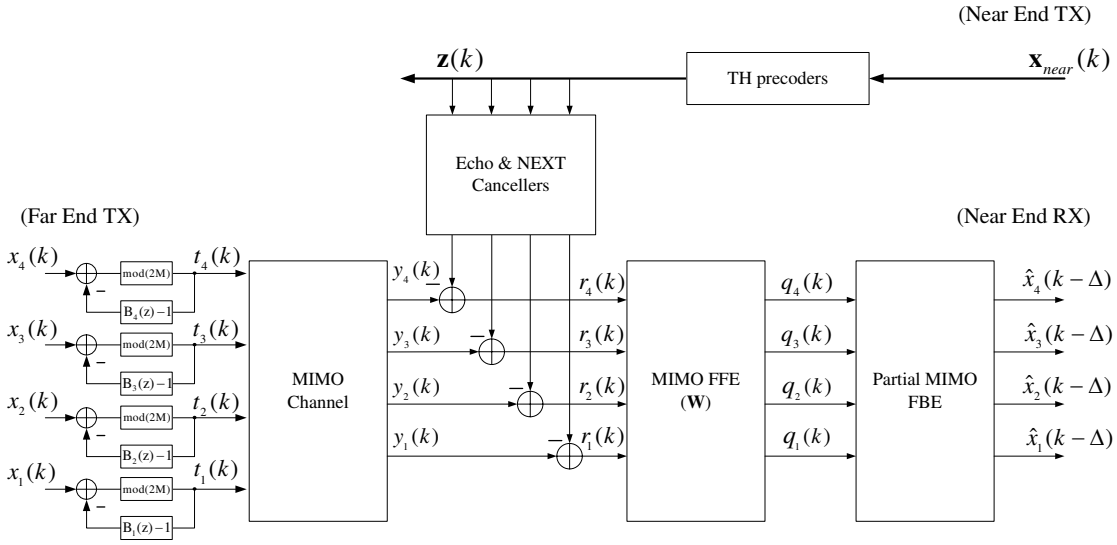


Figure 3.9: Block diagram of the proposed MIMO equalization with TH precoding

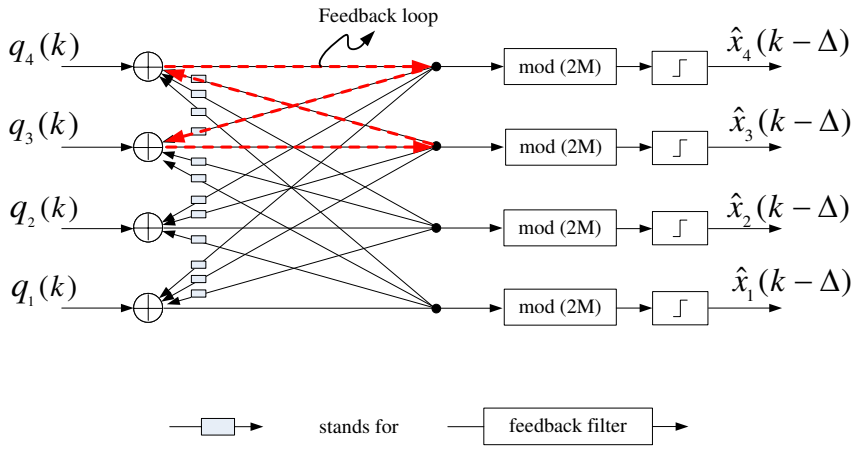


Figure 3.10: Internal structure of the partial MIMO-FBE

In Fig. 3.10, it should be noted that the inputs to these feedback filters are not easy

to obtain due to the TH precoding, which changes the transmitted signal $x_i(k)$ to $t_i(k)$. In addition, the feedback loops inside the partial MIMO-FBE filter limit the high speed implementation of these filters. To solve these problems, we develop the proposed designs as shown in Fig. 3.11 and Fig. 3.12, respectively.

Fig. 3.11 shows the detailed structure of the proposed MIMO equalization scheme for one pair in a transceiver. Compared with the design in Fig. 2.7, the proposed design here uses MIMO FFE filters $W_{i,1}(z)_{i=1,2,3,4}$ to jointly process the ISI and FEXT such that both pre-cursor ISI and non-causal FEXT can be effectively mitigated. Thus, the inserted bulk delay D is not necessary any more. In addition, if considering the output of the MIMO FFE at pair 1, we see that the tentative decision $\tilde{d}_1(k)$ based on $\underline{d}_1(k)$ will be more reliable compared with the design in Fig. 2.7. This is because non-causal FEXT has been removed by MIMO FFE filters such that the signal at the output of the MIMO FFE, $\underline{d}_1(k)$, will only contain $x_1(k) + v_1(k) + n_1(k)$ and residual post-cursor FEXT interferences. To further combat the residual post-cursor FEXT interferences, a partial MIMO feedforward filter (containing 12 cross filters) is used with the input signal $\tilde{t}_i(k)_{i=2,3,4}$.

If following the similar derivation in Section 3.3, we can define the average DP-SNR as [25]

$$SNR_{avg} = 10\log_{10}\left(\frac{\frac{1}{(L+N_1)}\text{trace}(\mathbf{R}_{tt})}{\frac{1}{4}\text{trace}(\mathbf{R}_{ee,min})}\right), \tag{3.26}$$

where $\mathbf{R}_{tt}$ is the $(L+N_1)\times(L+N_1)$ auto-correlation matrix of signal $t_i(k)$, and $\mathbf{R}_{ee,min}$ is the minimum decision error auto-correlation with dimension 4×4.

Fig. 3.12 shows a modified design of the proposed MIMO equalization scheme. It is designed for high speed applications by eliminating the IIR filter $\frac{1}{B_i(z)}$ after $\tilde{d}_i(k)$ such that all units in the receiver do not contain any feedback loops. Thus, pipelining techniques can be easily applied to speed up the operation [85].

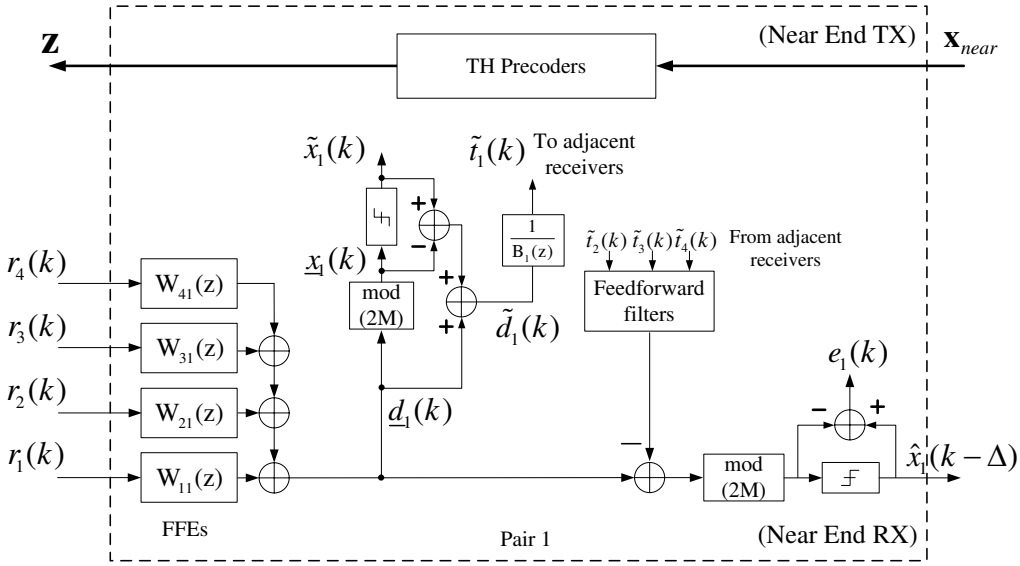


Figure 3.11: Proposed MIMO equalization scheme I with TH precoding

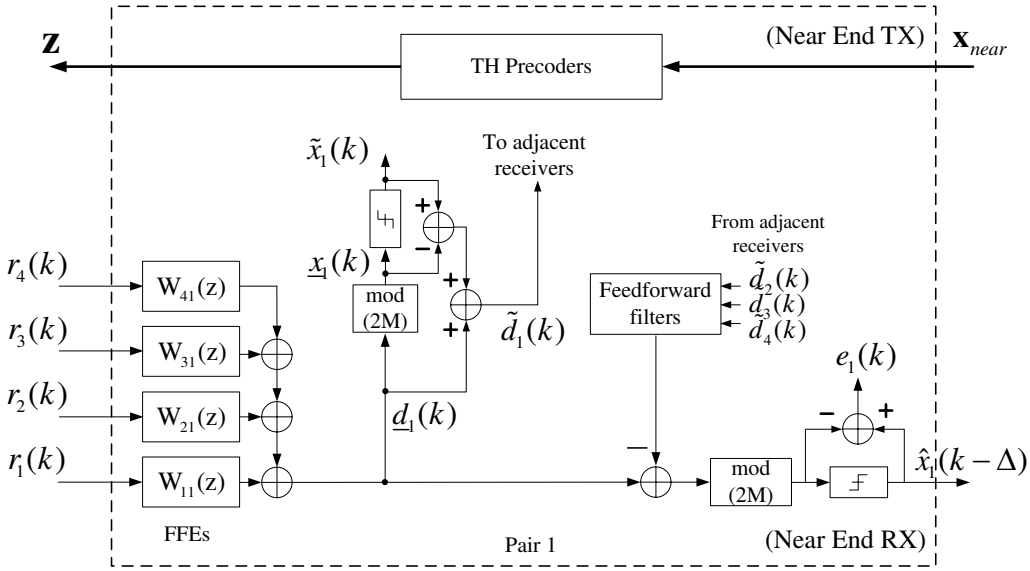


Figure 3.12: Proposed MIMO equalization scheme II with TH precoding

3.5.3 Simulations and Discussions

In this section, simulation results are presented to demonstrate the effectiveness of the proposed designs in the 10GBASE-T application. We focus on the proposed scheme on

MIMO equalization with TH precoding discussed in Section 3.5. The comparisons between the proposed design and its counterparts are also performed.

Similar to methodology used in Chapter 2, we first determine the design parameters by analyzing the theoretical result in equation (3.26). Instead of performing exhaustive search, heuristic search is used to reduce the search space. In the analysis, Cat-6 UTP cable with length 100m is used as the channel model, which can be obtained from the IEEE 802.3an website [9].

Fig. 3.13 plots the DP-SNR performance versus the lengths of echo and NEXT cancellers. In this figure, the length of the MIMO FFE is $L = 64$, the length of the TH precoder is $N_b = 32$, decision delay is $\Delta = 420$, and the bulk delay is $D = 0$. In addition, the length of each cross feedforward filter at the receiver is set as $N_x = 32$. Compared with Fig. 2.9, similar observations can be found.

Fig. 3.14 plots the the DP-SNR performance versus the lengths of the MIMO FFE and TH precoder, and each curve corresponds to a fixed length of the TH precoder. In this figure, $N_{ec} = 500$, $N_{nc} = 300$, $\Delta = 420$, $D = 0$, and $N_x = 32$. The right side of the figure illustrates the details of DP-SNR performance when the length of the MIMO FFE filter is from 50 to 100.

Fig. 3.15 plots the DP-SNR performance versus the decision delay, and each curve corresponds to a fixed length of the MIMO FFE filter. From Fig. 3.13 to 3.15, we can determine the design parameters, $N_{ec} = 500$, $N_{nc} = 300$, $L = 64$, $N_b = 15$, and $\Delta = 420$.

It is mentioned in Section 3.5, by applying the MIMO equalization technique, both precursor ISI and non-causal FEXT can be jointly mitigated by the MIMO FFE filters. Thus, the bulk delay D is not necessary any more. To demonstrate this, the DP-SNR performance versus the bulk delay is shown in Fig. 3.16. In this figure, each curve corresponds to a fixed

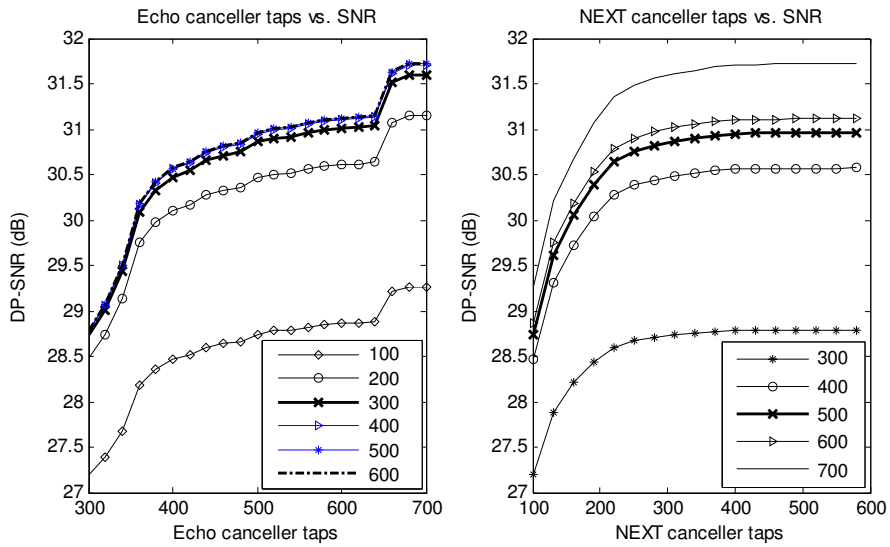


Figure 3.13: The DP-SNR performance versus the lengths of echo and NEXT cancellers

N_x. As we can see from the figure, the DP-SNR performance decreases as the bulk delay D increases. Note that for larger N_x (*i.e.*, greater than 40), the optimal D is not 0, and the DP-SNR can be improved by increasing D a little bit. However, the performance improvement is less than 0.05dB, which is very small. Thus, the bulk delay is not suggested in the proposed MIMO equalization scheme.

Table 3.3 summarizes the design parameters we choose according to the analysis above. Based on these parameters, the same simulation settings in subsection 2.4.3 can be applied to evaluate the performance of the proposed design.

Table 3.4 gives the performance comparison for different designs discussed in Section 3.5. In this table, each design is evaluated based on Cat-6 measured channel models with different lengths: 100m, 75m, and 55m respectively. The analytical result (a) is also listed to be compared with the simulation result (b) for each design.

First, compared with the results of the straightforward design listed in Table 2.2, the results of different schemes shown in Table 3.4 clearly demonstrate the advantage of using

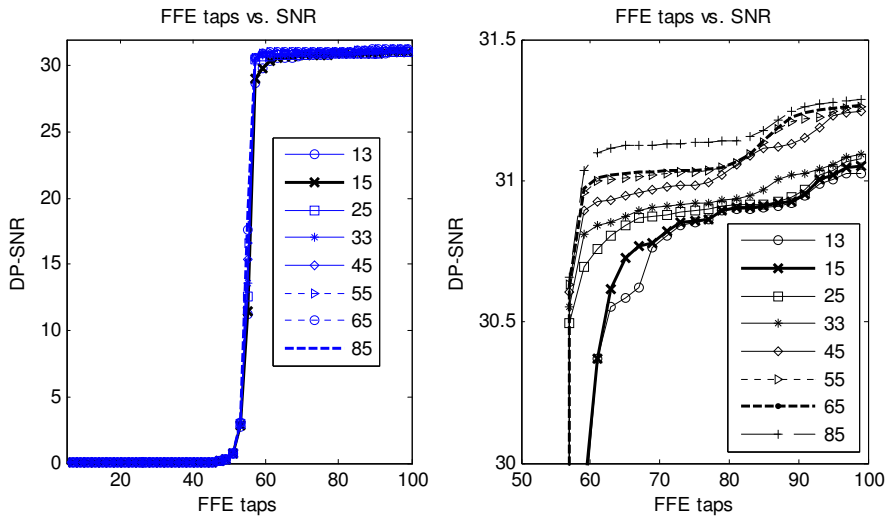


Figure 3.14: The DP-SNR performance versus the lengths of the MIMO FFE and TH precoder

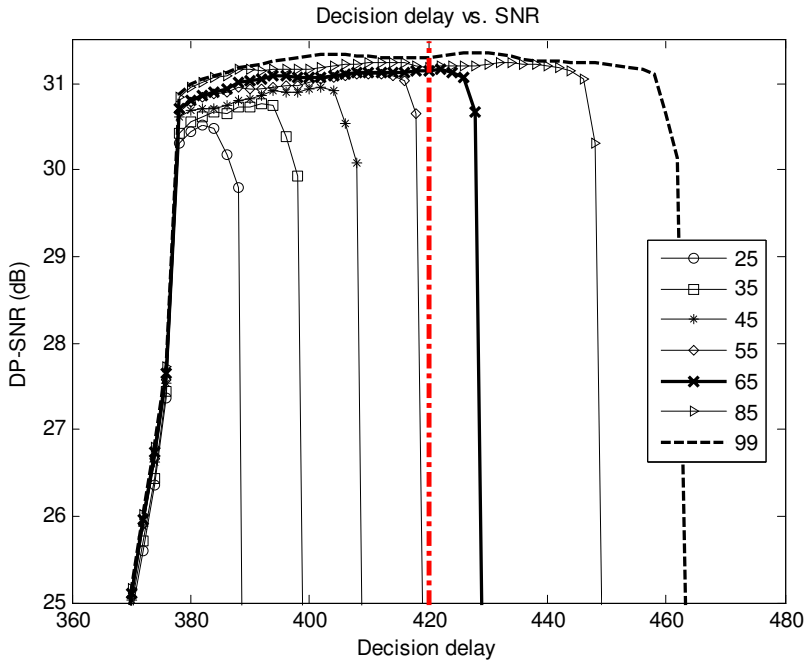


Figure 3.15: The DP-SNR performance versus the decision delay

the MIMO equalization technique. Consider the results in the traditional MIMO-DFE structure (Fig. 3.4), we see that the error propagation problem inherent in this structure

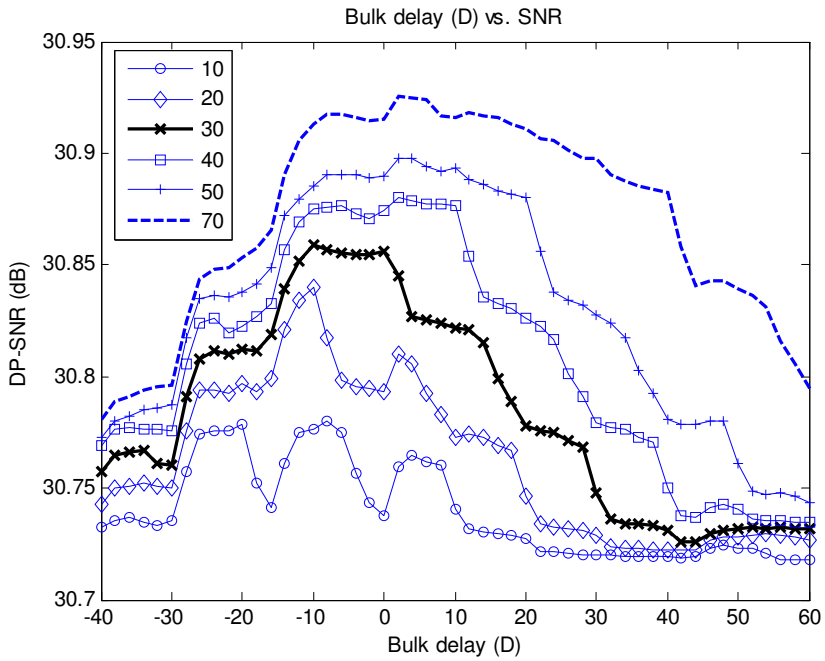


Figure 3.16: The DP-SNR performance versus the bulk delay

is alleviated. Especially, for short cable cases, the simulation results are very close to the analytical results. This indicates that the error propagation is not severe in high DP-SNR cases. For the MIMO-THP scheme shown in Fig. 3.7, the error propagation problem is eliminated by using the TH precoding. As we can see from Table 3.4, the MIMO-THP structure has better performance than the MIMO-DFE structure for 100m Cat-6 case. However, for high SNR cases (*i.e.*, 75m and 55m), the simulated DP-SNR is a little bit worse than the MIMO-DFE structure. This may be due to the effect of the estimation errors in the TH precoders [38, 40]. The proposed design III (Fig. 3.11) is obtained by combining the MIMO equalization technique and the TH precoding technique, and it can achieve similar performance as its counterparts as we expected. Compared with the proposed design III, the performance degradation in terms of simulation results in the proposed design IV (Fig. 3.12) is because an FIR filter is used to approximate an IIR filter in the cross feedforward

Table 3.3: Design parameters for the proposed MIMO equalization

Parameter	Values
Decision delay	420
Inserted bulk delay (D)	0
MIMO FFE taps	64×16
THP taps	15×4
Cross-DFE taps	32×12
Echo canceller taps	500×4
NEXT canceller taps	300×12
Total taps	7068

Table 3.4: Average decision-point (DP) SNR during the steady state, AWGN=-150dBm/Hz, Bulk delay D=0, $N_x = 32$

CAT-6 UTP Cable	Average DP-SNR (dB)							
	MIMO-DFE (Fig. 3.4)		MIMO-THP (Fig. 3.7)		Proposed design III (Fig. 3.11)		Proposed design IV (Fig. 3.12)	
	(a)	(b)	(a)	(b)	(a)	(b)	(a)	(b)
Measured 100m	30.71	29.95	30.72	30.25	30.72	30.64	30.72	29.57
Measured 75m	36.94	36.93	36.95	36.91	36.95	36.94	36.95	36.85
Measured 55m	41.12	41.12	41.13	41.10	41.13	41.10	41.13	40.95

filter at the receiver.

3.6 Hardware Complexity Reduction Scheme

When calculating the total number of taps in Table 2.1 and Table 3.3, we note that complexity of echo and NEXT cancellers accounts for about 79% in the whole transceiver. If the complexity of echo and NEXT cancellers is reduced, the overall complexity of the transceiver will be decreased.

In the following, we take the proposed MIMO equalization scheme in Fig. 3.12 as a design example to present our methods. Consider Fig. 3.13, the DP-SNR performance increases slowly when the length of the echo canceller N_{ec} increases from 400 to 600. However, when N_{ec} goes over about 650, there is about 0.5dB performance improvement. To

Table 3.5: Design example for complexity reduction

Parameter	Values in (Table 3.3)	Example I	Example II
MIMO FFE taps	64×16	64×16	70×16
THP taps	15×4	15×4	20×4
Cross-DFE taps	32×12	0×12	0×12
Echo canceller taps	500×4	670×4	420×4
NEXT canceller taps	300×12	170×12	130×12
Total taps	7068	5804	4440
Reduction rate (%)	0	18	37.2
Simulated DP-SNR (dB)	29.57	29.60	26.81

maintain the same performance in the original settings shown in Table 3.3, we can reduce the length of the NEXT canceller to counteract the 0.5dB performance gain. Table 3.5 gives the proposed design settings shown in Example I column. Since the number of NEXT cancellers is three times as many as that of echo cancellers, reducing the length of each NEXT canceller can lead to the overall complexity reduction. It is shown in this table, the total number of taps is reduced by 18% while maintaining the same performance as the original design. It is noted that the simulation results are obtained based on Cat-6 100m channel model. Compared with the minimum DP-SNR requirement 23.8dB suggested in the 10GBASE-T system, the performance gain in the proposed design can be used to further reduce the complexity of echo and NEXT cancellers. The proposed design settings are shown in Example II column. It can be seen that 37.2% overall complexity reduction can be achieved while meeting the design requirement with 3dB SNR margin. The savings will be even greater if lower SNR margin is allowed.

3.7 MIMO Equalization with THP versus FEXT Cancellation

In this subsection, the proposed MIMO equalization scheme with TH precoding is compared with the proposed FEXT cancellation scheme presented in Chapter 2 in terms of hardware complexity, convergence speed, and DP-SNR performance.

We first consider the hardware complexity of the proposed designs based on these two different approaches. In the comparison, the total number of filter taps in a transceiver is used as a measure. From Table 2.1 and Table 3.3, we can calculate the the total number of filter taps for the two different approaches as 7116 and 7068, respectively. As we can see, the proposed MIMO equalization schemes (Fig. 3.11 and 3.12) have lower hardware complexity although the complexity reduction is not much.

We also compare the convergence speed of the proposed designs based on two different approaches by applying the LMS algorithm [84]. The simulation is performed based on Cat-6 100m channel model. The learning curves of the corresponding designs during the equalizer training stage are plotted in Fig. 3.17. In this figure, each learning curve is obtained by averaging the results over 80 independent experiments. From the figure, we see that the convergence speed of both proposed approaches is similar. Since the ideal training sequence is assumed during the simulation, both can converge to the theoretical limit. For the purposes of comparison, the learning curve of the traditional design in Fig. 2.2 is also plotted in the figure.

Finally, we compare the performance of the proposed designs based on these two approaches. According to the analytical results in Tables 2.2 and Table 3.4, we see that the proposed designs I and II based on FEXT cancellation approach have slightly better performance, which may result in higher hardware complexity compared with the proposed

designs III and IV. However, in terms of the simulation results, the proposed designs III and IV exhibit better performance than the FEXT cancellation schemes because the difference between the analytical results and simulation results in both designs is smaller. Specifically, compared with the results for proposed design II in Fig. 2.8, the proposed design IV in Fig. 3.12 has better DP-SNR performance even though its analytical results are worse. This occurs because the tentative decision obtained from the proposed MIMO equalization schemes is more reliable than that obtained from the proposed FEXT cancellation schemes. In addition, we need to note that the inserted bulk delay in the proposed FEXT cancellation schemes will increase the latency of the decision output.

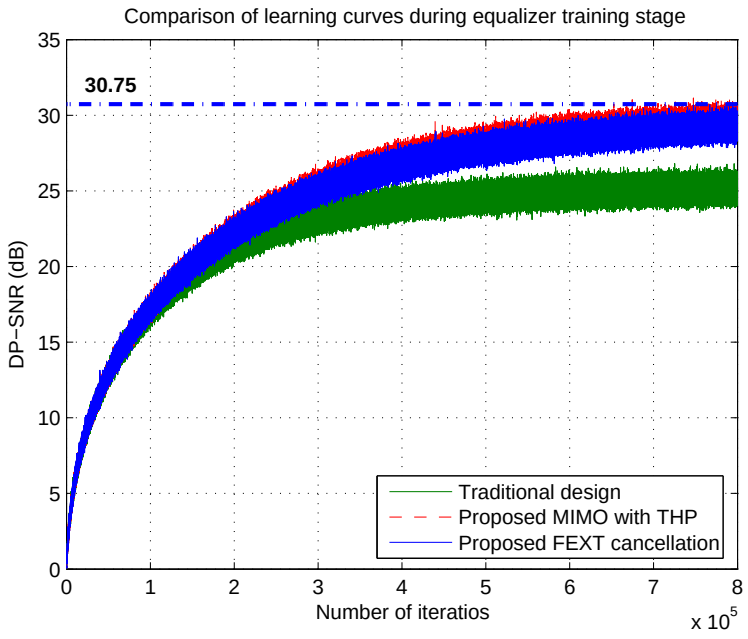


Figure 3.17: Comparison of the learning curves for different designs during the training stage

3.8 Conclusion

In this chapter, we have presented to use MIMO technique to deal with FEXT as signal rather than background noise in 10GBASE-T transceiver design. First, a classical MIMO-DFE based receiver architecture is developed to demonstrate the advantage over the traditional receiver design, where FEXT is cancelled as noise. It is shown that the proposed architecture overcomes the limitation of the traditional schemes and achieves a better SNR performance and lower receiver complexity. It is also noted that, with the same echo and NEXT complexity, the joint equalization and cancellation structure is not necessarily superior to the separate equalization and cancellation structure in the MMSE sense.

We have also presented a new method to compute the optimal coefficients of the MIMO equalizers and cancellers in 10Gbase-T channel. The proposed approach is exact and applicable to the general MIMO DFE computation as well as such cases where echo and NEXT cancellers have a large number of taps with different lengths, which usually make Al-Dhahir's method inefficient. It is shown that, by using the proposed method, we are able to achieve about 63.8% computation cost reduction in terms of multiplication operations compared with the existing methods. This computation speedup also makes the analysis easier when Alien crosstalk such as ANEXT is considered in the channel model.

Although the advantage of the MIMO equalization technique has been demonstrated, the classical MIMO-DFE based receiver architecture suffers from error propagation problem in real applications. To eliminate this problem, a MIMO-THP structure is proposed by simply moving the feedback part of the MIMO-DFE to the transmitter. However, the MIMO-THP structure is not supported in the 10GBASE-T standard, where only four separate TH precoders are required for each channel. Besides, the high speed implementation of the MIMO-TH precoder is still difficult. Thus, a new equalization scheme is proposed by

combining the MIMO equalization technique and TH precoding technique to deal with both ISI and FEXT. Different with the existing works, the proposed designs inherit the advantage of MIMO equalization and also alleviate the error propagation. In addition, they comply with the 10GBASE-T standard and are also suitable for high speed application because feedback loops in the receiver is completely removed so that pipelining techniques can be easily applied.

Chapter 4

Design of Cost-Efficient Echo and NEXT Cancellers

4.1 Introduction

In many multiple wireline communication systems, such as digital subscribe line (DSL) and gigabit Ethernet systems, the received signal not only suffers from signal attenuation and intersymbol inference (ISI), but also suffers from echo and near end crosstalk (NEXT) interferences. Especially in 10 Gigabit Ethernet over copper (10GBASE-T) system, the full duplex baseband transmission is performed over four pairs of unshielded twisted pair (UTP) copper cables, where ISI is a significant impairment against reliable high speed digital transmission, and each received signal is also corrupted by echo from its own transmitter and NEXT interferences from three adjacent transmitters. To meet the desired throughput ($10Gbps$) and target BER (10^{-12}) requirements, echo and NEXT noise cancellation are expected to be about 55dB and 40dB, respectively [6–8].

To counter the effect of echo and NEXT interferences, echo and NEXT cancellers are

widely used in many communication systems. Generally, echo and NEXT cancellers are implemented by using finite impulse response (FIR) filters in digital domain, where the replica of the echo and NEXT estimated by the FIR filters is subtracted from the received noisy signals. This traditional approach, however, will lead to a significant hardware complexity if the number of taps in the FIR filters is large. For example, in the typical 10GBASE-T application [7], one echo canceller and three NEXT cancellers are needed for each pair of cables. Since there are four pairs of cables (four channels) in 10GBASE-T, a total of four echo cancellers and twelve NEXT cancellers are needed at the receiver end. Furthermore, To achieve high performance noise cancellation, each FIR based echo and NEXT canceller requires hundreds of taps [7], and the total number of taps in these cancellers is around 5600∼6800. Straightforward implementation of these cancellers will consume large silicon area and power consumption. Thus, it is of great interest to develop techniques to reduce the hardware cost of these cancellers without degrading the performance of noise cancellation.

Reducing the complexity of the FIR based noise cancellers can be achieved in many ways. One strategy to reduce the cost of these cancellers is to reduce the computational complexity at algorithm level. It is known that Fast Fourier Transform (FFT) algorithm is efficient in filter computation. FIR filtering operation in time domain can be implemented in frequency domain. However, new issues such as block processing latency, increased memory and increased precision, associated with this technique make it unsuitable for high speed Ethernet transmission systems such as the 10GBASE-T [33]. Another effective way is to reduce the number of taps used in these noise cancellers, as the hardware cost increases with the number of taps. In [34], a shortened impulse response filter (SIRF) was used to reduce the number of taps required in echo and NEXT cancellers; hence the overall complexity can be reduced. Some work on exploiting the sparsity of the echo channel impulse response

has been addressed to find the most significant (active) taps (with large magnitude) which contribute most in the echo noise cancellation [43, 44]. Since the number of those active taps are much less compared with the original filter taps, reducing the complexity of the echo canceller is possible. In [57], adaptive IIR filter was proposed for design of echo cancellers. The advantage is that the IIR filter structure has less number of taps than the FIR filter structure, especially when the poles of the echo channels are close to the unit circle. However, the adaptive IIR filter usually suffers from problems such as instability and slow convergence, which make them not widely used. Reducing the word-length used to represent data in a VLSI system is also an effective way to reduce the overall complexity of the system, as hardware cost increases with word-length. However, due to the use of TH precoder in 10GBASE-T [7], the inputs to echo and NEXT cancellers are no longer simple PAM-M symbols but numbers uniformly distributed on $[-M, M)$. These make look-ahead and pre-computation techniques difficult to apply [62, 63]. Furthermore, the word-length of these inputs needs to be long to achieve required noise cancellation. Thus, it is more difficult to efficiently implement these cancellers in a 10GBASE-T Ethernet system [58].

This chapter approaches the problems of designing cost-efficient echo and NEXT cancellers in the application of 10GBASE-T Ethernet systems mainly from two different aspects. In the first approach, the sparse characteristics of the echo and NEXT channel impulse responses is exploited to reduce computational cost of adaptive echo and NEXT cancellers. Unlike previous work on exploiting the channel sparsity, the proposed scheme uses correlation method to estimate the long sparse channel impulse response and identifies the active taps based on a dynamic threshold. Instead of performing full size filtering operation, adaptive filtering operation in the proposed approach is only performed on a few active taps. Hence, the overall complexity of echo and NEXT cancellers can be reduced. Furthermore, to

avoid performance degradation due to the channel estimate errors in a slowly time-varying channel, an efficient tap management algorithm is developed to keep track of those active taps. An area efficient architecture is also presented to implement the proposed algorithm. Simulation results show that, by using the proposed method in the gigabit transceiver design, we are able to achieve 50% and 29∼66.7% computational cost reduction in terms of multiplication during the initial training stage and steady stage, respectively.

In the second approach, a novel method based on word-length reduction technique is proposed to reduce the hardware complexity of echo and NEXT cancellers. The proposed design is derived by replacing the original input to the echo and NEXT cancellers with a finite-level signal, which is the sum of the input to the TH precoder and a finite-level compensation signal. Then this modified input signal is recoded to have shorter word-length compared with the original input. Hence, the overall complexity can be reduced by using the proposed method. To further reduce the complexity of these cancellers, an improved design is proposed by exploiting the property of the compensation signal. Compared with the traditional design, the proposed echo and NEXT cancellers have exact input and do not suffer from the quantization problem, and thus they are more suitable for VLSI implementation. Complexity analysis shows that 22.08% hardware saving can be achieved compared with the traditional design. In practice, echo and NEXT channels are slow-varying and adaptive cancellers are needed for noise cancellation. Directly applying the proposed method to the adaptive filter design will lead to problems such as performance degradation and slow convergence. To solve these problems, a modified design of adaptive echo and NEXT cancellers is presented. It is shown that the overall hardware cost of adaptive echo and NEXT cancellers can be reduced by about 10.82% without any performance loss.

However, the hardware cost reduction is mainly due to the hardware cost reduction of

the filter part in these adaptive cancellers. To further reduce the overall hardware cost of these adaptive cancellers, a new complexity reduction scheme is proposed for the weight update part in these adaptive cancellers. The proposed scheme is general and can be applied to multiple-input-multiple-output (MIMO) systems such that hardware cost of both echo and NEXT cancellers can be reduced.

The rest of this chapter is organized as follows. In Section 4.2, the design of cost-efficient echo and NEXT cancellers based on an adaptive tap management algorithm is presented in detail, and simulation results are provided to demonstrate the applicability of the proposed design to the 10GBASE-T application. Section 4.3 presents the low complexity design of echo and NEXT cancellers based on word-length reduction techniques. In addition, the design issues of adaptive cancellers by using the proposed method are discussed and a modified design of the adaptive cancellers is presented to avoid the slow convergence during the initial training stage. Section 4.4 presents a new complexity reduction scheme for the weight update part in adaptive filters to further reduce the overall hardware cost of echo and NEXT cancellers. Conclusion is provided in Section 4.5.

4.2 Adaptive Tap Management in Echo and NEXT Cancellers

In traditional echo and NEXT cancellers, to achieve the required noise cancellation, the number of filter taps in the FIR-based cancellers is usually made as large as possible to cover the overall echo and NEXT channel impulse responses. Implementing these echo and NEXT cancellers with very large number of taps leads to a slow convergence speed as well as a heavy computational load. However, considering the feature of two typical impulse responses for 10GBASE-T echo and NEXT channels as shown in Fig. 4.1, we note that

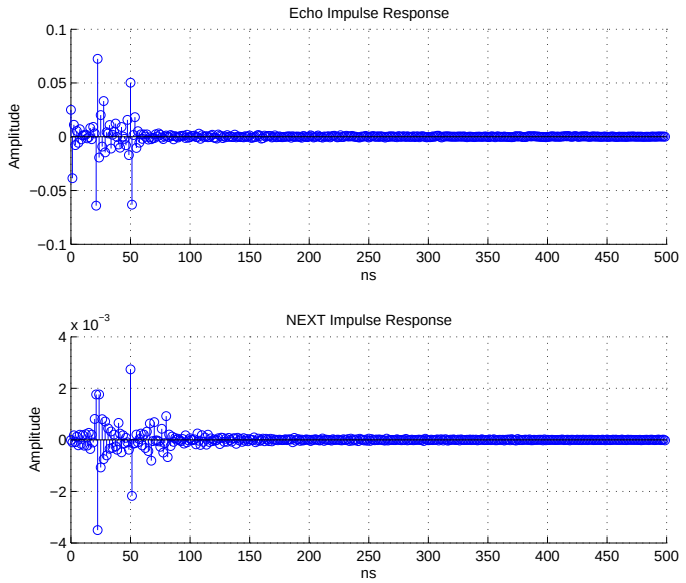


Figure 4.1: Typical echo and NEXT impulse response (Cat-6 UTP)

these channel impulse responses exhibit sparse behavior. For example, there is a short main part with large magnitude variations followed by a long tail part of very small values. To cancel the echo and NEXT interferences generated in this kind of channels, the resulting FIR filter-based cancellers will exhibit similar property as their corresponding channels. In other words, many taps are negligible with only a few significant or active taps which effectively contribute to noise cancellation. Without allocating hardware at those inactive taps, then significant complexity reduction of those echo and NEXT cancellers can be achieved. However, the challenge is how to identify those active taps during the adaptation.

The problem of identifying active taps in sparse systems has been addressed in many ways. One popular approach is known as proportionate normalized least mean square (PNLMS) [44], in which the step size at each iteration was calculated from the last estimate of the filter coefficients so that a larger coefficient received a larger weight update. Thus it gave a fast initial convergence speed under sparse channels as well as an improved stability.

In [45, 46] an improved PNLMS was proposed to make the PNLMS algorithm also suitable with non-sparse channels. However, the high complexity associated with both PNLMS and IPNLMS algorithms makes them unsuitable for hardware implementation. Queue based algorithms such as scrub taps waiting in a queue (STWQ) [47, 48] can reduce the number of computations required by considering the active taps only. The STWQ algorithm also works well with correlated input data and can cancel multiple echoes. However, STWQ often provides slower convergence than standard least mean square (LMS) [84] because its queue based technique needs long time to identify all active taps when the queue length is very large. More recently, the sparse partial (SP) update NLMS (SPNLMS) algorithm [50] incorporates the MMax tap-selection [51] for the reduction of computational complexity. Other approaches based on wavelet transform technique [52, 53] have been proposed to achieve a faster convergence performance as well as a reduced computational complexity. However, the inherent block delay introduced between the input and output makes them unacceptable for 10GBASE-T application since the number of filter taps could be several hundreds [7].

In this section, we present a new adaptive tap management algorithm to exploit the sparsity of the echo and NEXT channels and also detail the corresponding hardware architecture. First, a cross-correlation algorithm is briefly reviewed in Section 4.2.1. Section 4.2.2 describes a dynamic threshold rule based on energy cancellation level. Section 4.2.3 introduces a modified LMS algorithm with partial updating. Section 4.2.4 proposes an efficient active tap management algorithm to keep tracking active taps. Section 4.2.5 presents the hardware architecture of the proposed cancellers, as well as complexity analysis. Simulations and discussions are provided in Section 4.2.6.

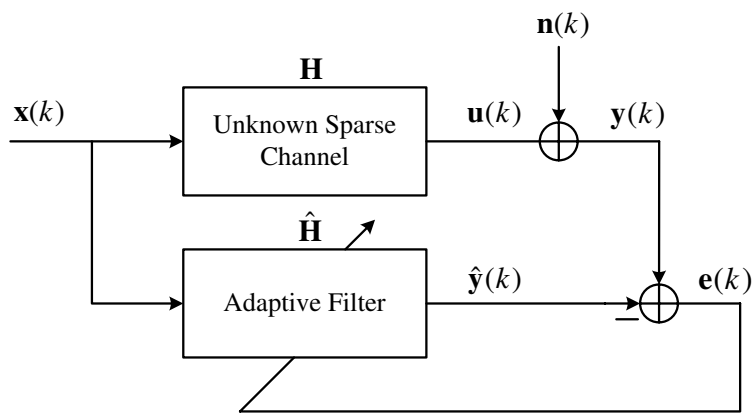


Figure 4.2: Adaptive system identification model

4.2.1 Cross-Correlation Channel Estimation

According to [54,55], we first model the channel estimation problem as the adaptive system identification problem as shown in Fig. 4.2. In this figure, $\mathbf{H}$ represents the unknown echo and NEXT channel impulse, which needs to be identified using an adaptive FIR filter $\hat{\mathbf{H}}$ with length N. Here we note, $\hat{\mathbf{H}}$ is a 4×4 matrix filter with element $(\hat{h}_{i,j})_{i=1\ldots4,j=1\ldots4}$, where $\hat{h}_{i,j}$ denotes coefficients of echo and NEXT cancellers from i-th input to the j-th output, $\mathbf{x}(k)$ is a 4×1 transmitted vector from the near end transmitter at the time index k, and $\mathbf{n}(k)$ is also a 4×1 vector, denoting background noise at the time index k. Then we have

$$\mathbf{y}(k) = \mathbf{u}(k) + \mathbf{u}(k), \tag{4.1}$$

$$\hat{\mathbf{y}}(k) = \hat{\mathbf{H}}^H\mathbf{x}(k+N-1:k), \tag{4.2}$$

$$\mathbf{e}(k) = \mathbf{y}(k) - \hat{\mathbf{y}}(k). \tag{4.3}$$

The optimal $\hat{\mathbf{H}}_{opt}$ to minimize the mean square error (MSE) is given by Wiener solution

[84],

$$\hat{\mathbf{H}}_{opt} = \mathbf{R}_{xx}^{-1}\mathbf{P}, \tag{4.4}$$

where $\mathbf{R}_{xx}$ is auto-correlation of input to the adaptive filters, and $\mathbf{P}$ is cross-correlation between input vector $\mathbf{x}(k)$ and received signal $y(k)$. Assuming the input signal $x(k)$ is uncorrelated with variance σ_x^2, we have $\mathbf{R}_{xx} = 1/\sigma_x^2$. Furthermore, according to [84], the cross-correlation $\mathbf{P}$ can be estimated using

$$\mathbf{P} = E[\mathbf{x}\mathbf{y}^H(k)] \approx \frac{1}{M}\sum_{i=0}^{M-1}\mathbf{x}\mathbf{y}^H(k), \tag{4.5}$$

where input vector $\mathbf{x}$ has block length of N, which is determined by length of the adaptive filters. M is the number of samples used to estimate the cross-correlation. The accuracy of the estimation is related to M. Substituting (4.5) into (4.4), we get

$$\hat{\mathbf{H}}_{opt} \approx \frac{1}{M\sigma_x^2}\sum_{i=0}^{M-1}\mathbf{x}\mathbf{y}^H(k), \tag{4.6}$$

which leads to the cross-correlation estimate.

4.2.2 Dynamic Threshold Rule

One straightforward method to choose active taps from estimated channel is based on their magnitude. In [54], a fixed number of active taps, B, is pre-assigned, and the most significant B taps is selected based on their absolute values. The choice of B is troublesome since the channel is unknown and time-slow varying. Small B saves computation complexity at the expense of performance; and large B maintains the system performance with the cost of computation complexity. Thus, it is more attractive to use a variable threshold based on

the channel estimate to dynamically identify those active taps.

In the following, we consider an echo impulse response with N coefficients $\{g(0),\ g(1),\ \ldots,\ g(N-1)\}$ to illustrate the proposed threshold rule. However, similar methodology can be applied to NEXT cancellers. Before cancellation is performed, the total echo energy can be defined as

$$E = \sum_{i=0}^{N-1} g^2(i). \tag{4.7}$$

Then, the residual echo energy after performing echo cancellation can be expressed as

$$E' = \sum_{i \in \Omega} g^2(i), \tag{4.8}$$

where Ω is a set of residual taps which contribute to the total echo energy. Suppose α dB echo cancellation is required, we have the relation as

$$\alpha = 10 \cdot \log(E) - 10 \cdot \log(E'). \tag{4.9}$$

From (4.9), E' can be expressed as

$$E' = E \cdot 10^{-\alpha/10} = \beta \cdot E. \tag{4.10}$$

According to the residual echo energy shown in (4.10), we see that once β is given, an energy threshold will be automatically determined from the overall echo energy E. Furthermore, we should note that this energy threshold can dynamically change with the time-varying channel such that the number of active taps is not necessary to be fixed. By using this

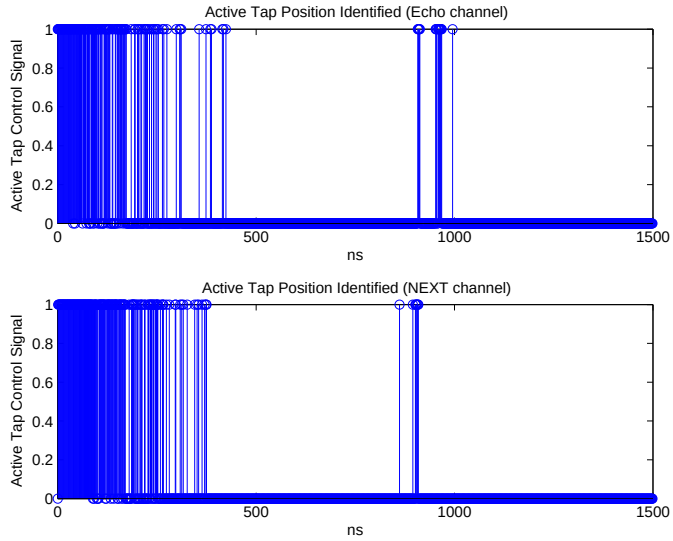


Figure 4.3: Active tap positions identified using (4.11)

threshold, the control signal for each tap can be defined as

$$c_i = \begin{cases} 1, & g^2(i) \geq E' \\ 0, & otherwise \end{cases}.$$

In practice, it is expensive to evaluate the equation (4.10) and (4.11) since square operation involved. Thus, a relaxed threshold is developed as

$$c_i = \begin{cases} 1, & |g(i)| \geq \underset{i}{mean}\, |g(i)| \sqrt{N\beta} \\ 0, & otherwise \end{cases}.$$

Fig. 4.3 shows the active taps identified by applying (4.11) to the echo and NEXT channels shown in Fig. 4.1.

4.2.3 LMS With Partial Updating

Consider the weight update equation in a LMS algorithm,

$$\hat{h}_{i,j}(k+1) = \hat{h}_{i,j}(k) + \mu \mathbf{x}_i(k) e_j(k), \tag{4.11}$$

we see that for each sample time, updating these coefficients needs lots of computational cost, especially when the length of the adaptive filter, N, is very long. However, not all filter coefficients need to be updated in the sparse channel. Once the active taps are identified, we only need to update the filter coefficients on active taps, which will save a lot of computation compared with the full-size updating on all adaptive filter taps since the number of active taps is very small. During the period of tracking, group based method can be used when channel varies slightly. For example, we can separate these coefficients into several groups, and only one or two groups need to be updated to track the slowly time-varying channel at each sample time.

4.2.4 Active Tap Tracking

Unfortunately, the cross-correlation method is not an accurate estimation unless the number of samples is very large. Especially, when the input signal to the adaptive filter is not strictly uncorrelated and channel is slow-varying, the cross-correlation method will provide more errors, which needs to perform cross-correlation again. However, this extra computation cost can be avoided by introducing an efficient active taps tracking scheme. During the process of coefficients updating, we find some coefficients in the active set will be smaller than those in inactive set which is due to either inaccurate channel estimate or slowly varying channel. Based on the idea that larger taps contribute more than smaller taps, it is reasonable to exchange the smallest coefficient in the active set and the largest coefficient in the inactive

set. By doing this, the disadvantage of the cross-correlation method can be overcome, and therefore the noise cancellation performance can be improved. In addition, the scheme can be easily implemented in hardware by simply changing their tap control signals. Considering there might be additional computation load due to the absolute comparison operation, one solution to reduce the hardware overhead is using group-based method, which can be combined with partial updating technique in practical application. The proposed algorithm can be summarized as follows:

Proposed Algorithm: Adaptive Tap Tracking Algorithm

Two Stages Process :

Step I. Initial training stage

- a. Perform the cross-correlation method using M samples ($M \sim 4N$)
- b. Identify active taps using the proposed threshold rule (4.11)
- c. Set up the adaptive filter using the coefficient values estimated in Step I.a
- d. Store the size of active taps

Step II. Tracking stage

- a. Active tap tracking
  1. At each iteration, evaluate the coefficients of all taps by groups
  2. For each updated group:
     find the $\max\{|h_i|\}$, $i \in$ inactive set
     find the $\min\{|h_j|\}$, $j \in$ active set
  3. Exchange the index i and j, and set $h_i = 0$
- b. Partial updating
  1. If there is no tap exchange, only adapt coefficients in active set
  2. Else, perform partial updating by blocks

4.2.5 Hardware Architecture

In this section, a hardware architecture is developed to implement the proposed algorithm described in the previous section and the complexity analysis of the proposed hardware architecture is also provided.

Consider equation (4.6), and we see that it consists of a matrix correlation. Generally, the matrix correlation can be implemented by 16 correlators with received data samples and near end transmitted data symbols as inputs. The scalar terms in the equation (4.6) can be

implemented using a shifter assuming that M is chosen appropriately to be the power of 2. Suppose the channel to be estimated has length N, and then only N multiplication operations are needed at each iteration, compared with $2N$ multiplication operations needed in a standard LMS algorithm. Since each correlator has N taps, implementing these correlators separately will occupy large extra silicon area if N is very large. However, we note that the adaptive tap tracking algorithm described in previous Section has two main stages, in which correlation operation and adaptive filtering operation are time-divisible. Thus, we can use the weight updating part of the adaptive filter to implement these correlators with the price of only one multiplexer and one shifter. The proposed hardware architecture is shown in Fig. 4.4, which only shows the proposed canceller architecture for one channel, cancellers for other channels can be implemented in the similar way. From this figure, we note that it is a modified N-tap standard LMS adaptive filter structure with extra control signals on each tap. The tap control signals on filtering part determine which tap is active, and control signals on updating part decide how to group the active taps needed to update. During the initial stage, the channel estimate is computed by using cross-correlation method and then stored in the registers in the weight updating part. Once the initial stage finishes, the filter coefficients are selected by active tap control signals to perform filtering operation.

Consider Fig. 4.4, and we observe that at each iteration effective taps in the filtering operation are only a few and most of them are inactive. Although those inactive taps are disabled to minimize the power consume, they still occupy the silicon area. Since there are 16 such noise cancellers with the same structure, a large amount of extra hardware area is required. To further reduce the hardware area of the proposed architecture, we develop a modified design as shown in Fig. 4.5. In this figure, we see that the number of taps required in the filtering part is reduced at the expense of two extra routing networks.

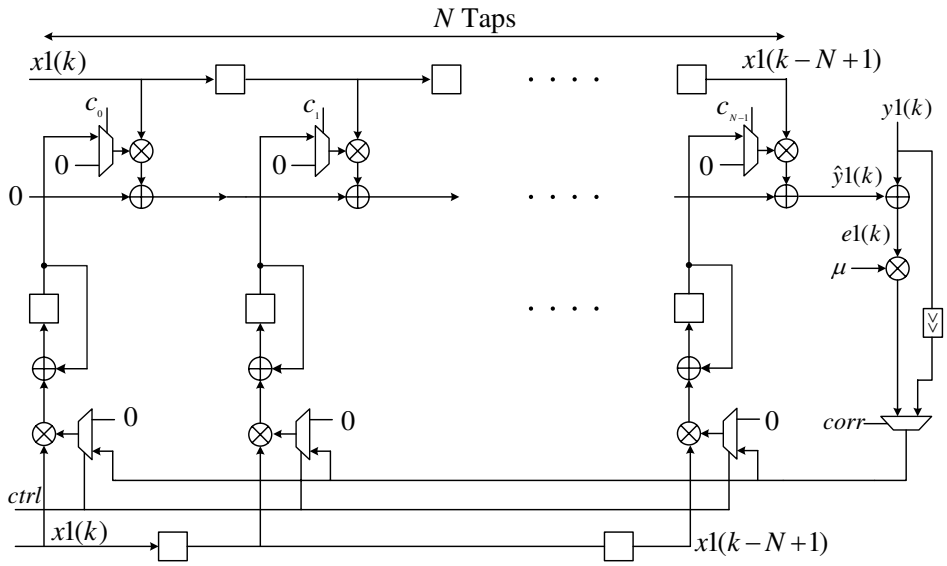


Figure 4.4: Hardware architecture of the proposed noise canceller (I)

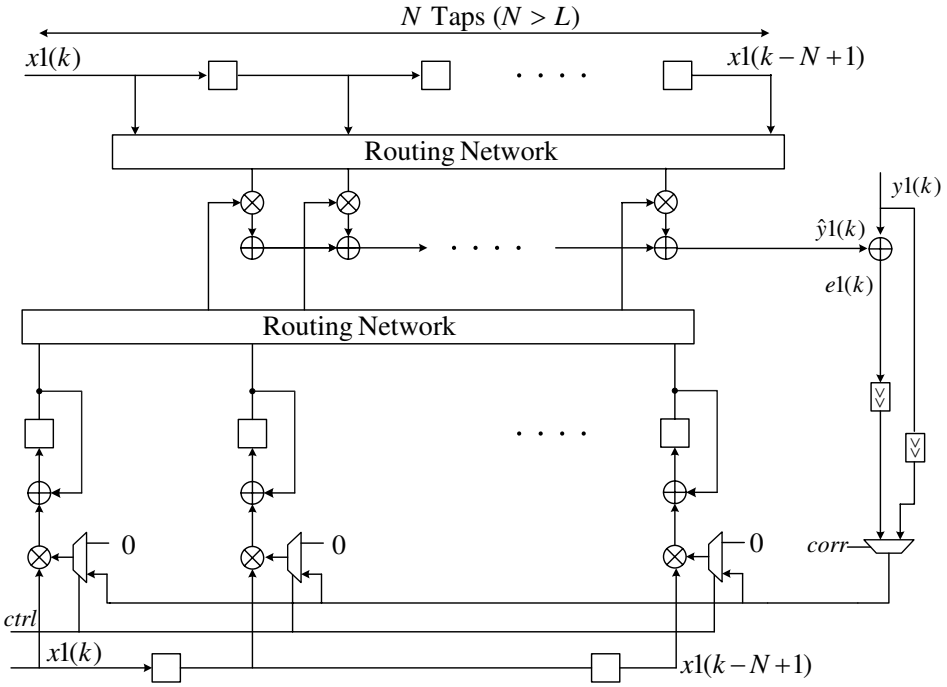


Figure 4.5: Area-efficient hardware architecture of the proposed noise canceller (II)

However, it should be noted that if designed efficiently, the hardware overhead due to the routing networks will be much less than hardware reduction due to inactive taps. Thus it will provide us an area efficient design.

Assume the computational complexity is dominated by the multiplication and addition, the proposed method will significantly reduce the computational complexity not only at the initial convergence stage, but also at the steady and tracking stage. Thus, it is important to reduce the overall power consumption of the gigabit transceiver. To demonstrate the advantage of the proposed method, Table 4.1 gives the comparison of computational complexity between the traditional LMS method and the proposed method at each sample iteration. In this table, N represents the length of the estimated channel and L denotes the size of the active tap set, and S is the number of groups when performing partial updating. For example, in Table 4.1, traditional LMS method needs $2N$ and $2N$ multiplications at the initial training stage and steady tracking stage, respectively. While the proposed method (I and II), only needs N multiplications at the initial stage and $N/S + L$ multiplications at the steady tracking stage. Thus, significant multiplication cost can be saved, especially when L is much less than N.

Table 4.2 shows the comparison of hardware complexity between the traditional LMS method and the proposed method. In this table, we see that the proposed architecture II has the smallest hardware complexity in terms of multipliers and adders, compared with traditional LMS architecture and proposed architecture I.

4.2.6 Simulation Results

In this section, we apply the proposed method to different Cat-6 UTP channels with the length of 100m. All these channel models can be obtained from IEEE 802.3an 10GBASE-T Study Group [9]. In the simulation, we set the transmit power as 5dBm, additive white

Table 4.1: Comparison of computational complexity

	Computational complexity		
	Traditional LMS	Proposed architecture I	Proposed architecture II
Multiplication	$2N + (2N)$	$N + (N/S + L)$	$N + (N/S + L)$
Addition	$2N + (2N + 1)$	$2N + (N/S + L + 1)$	$2N + (N/S + L + 1)$
Overhead	-/-	comparator multiplexer absolute evaluator	routing network multiplexer absolute evaluator

Table 4.2: Comparison of hardware area

	Hardware complexity		
	Traditional LMS	Proposed architecture I	Proposed architecture II
Multiplier	$2N$	$2N$	$N + L$
Adder	$2N + 1$	$2N + 1$	$N + L + 1$
Overhead	-/-	comparator multiplexer absolute logic	routing network multiplexer absolute logic

Gaussian noise (AWGN) power as -150dBm, and the training symbols as $\{9, -9\}$; we also assume the training process follows the current 10GBASE-T standard [11]. For simplicity, FEXT crosstalk is not considered in the simulation, because it does not affect the cancellation performance of echo and NEXT cancellers.

Fig. 4.6 shows the comparison of the learning curve between the traditional LMS method and the proposed method. In this figure, each learning curve is obtained by averaging the results over 80 independent experiments based on measured data of 100m Cat-6 cables. It can be seen that the proposed method exhibits a faster initial convergence speed, but with a slight SNR penalty. Similarly, Fig. 4.7 shows the comparison of the learning curve between the traditional LMS method and the proposed method. However, each learning curve in this figure is obtained by averaging the results over 80 independent experiments based on scaled data of 100m Cat-6 cables. From this figure, we note that the performances of different

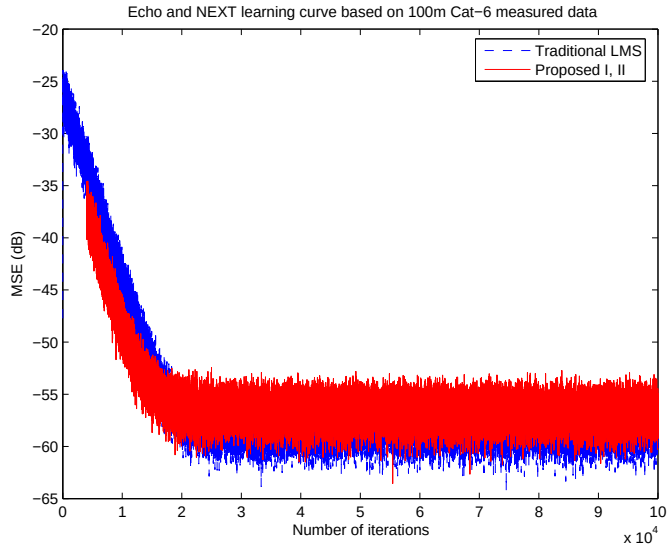


Figure 4.6: Comparison of the learning curves between the traditional design and proposed design I/II for measured data

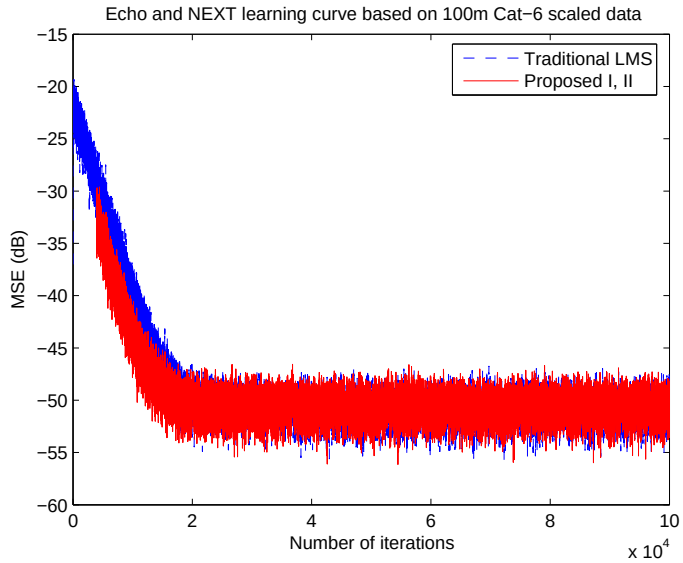


Figure 4.7: Comparison of the learning curves between the traditional design and proposed design I/II for scaled data

designs in scaled channel case are worse than those in measured channel case. This is because the scaled channel model is obtained by scaling the measured channel model to the

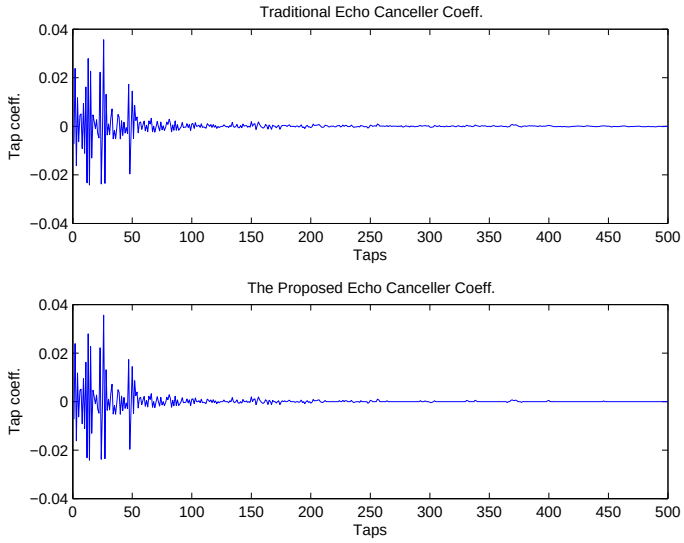


Figure 4.8: Comparison of the tap coefficients obtained from the traditional scheme and proposed scheme

theoretical limit, which is the worst case according to the cable requirement in 10GBASE-T.

To further evaluate the noise cancellation performance of the proposed method, we measure and compare the decision point signal noise ratio (DP-SNR) between the traditional LMS method and the proposed method during the steady stage. Table 4.3 gives the performance comparison for different schemes. In this table, each scheme is evaluated based on 100m Cat-6 measured and scaled data, respectively. As we can see from this table, the proposed scheme has performance loss in terms of DP-SNR under two different channel models, compared with traditional scheme. However, we should note that the performance loss is less than 0.5dB, which is acceptable in real applications.

Finally, for the purpose of illustration, Fig. 4.8 shows the tap coefficients of the echo canceller obtained by using the traditional scheme and the proposed scheme. To see the difference clearly, Table 4.4 gives the comparison of the number of active taps under two different channel conditions. It is seen that the number of active taps identified in the

proposed scheme is much less than that in the traditional scheme. This indicates that by applying the proposed scheme, the overall complexity of echo and NEXT cancellers can be reduced. Based on the analysis of the Table 4.1, we can easily see that there are 50% and 29∼66.7% computational complexity reduction in terms of multiplication and addition at the initial training stage and steady tracking stage, respectively.

Table 4.3: Decision point SNR during the steady stage for different designs

Cat-6 UTP	Traditional scheme	Proposed scheme I and II
100m measured	28.3dB	27.8dB
100m scaled	20.9dB	20.5dB

Table 4.4: The number of active taps

	Traditional scheme	Proposed scheme I and II
Echo taps (measured)	500	203
NEXT taps (measured)	500	254
Echo taps (scaled)	500	292
NEXT taps (scaled)	500	335

4.3 Word-length Reduction in Echo and NEXT Cancellers

Recently, TH precoding has been proposed to be used in 10GBASE-T because it can eliminate error propagation and allow use of capacity-achieving channel codes, such as low-density parity-check (LDPC) codes, in a natural way. However, the use of TH precoding significantly increases the hardware complexity of echo and NEXT cancellers in 10GBASE-T as the input to the echo and NEXT cancellers is no longer a simple PAM-M signal.

In this section, we propose to use word-length reduction techniques to tackle this prob-

lem. If the word-length of the input to echo and NEXT cancellers can be reduced, the overall complexity of these cancellers will be reduced. First, a TH precoder is converted to its equivalent form where the TH precoder can be viewed as an infinite impulse response (IIR) filter with an input equal to the sum of the original input to the TH precoder and a finite-level compensation signal. Instead of using the output of the TH precoder as the input to the echo and NEXT cancellers, the sum of the original input to the TH precoder and the finite-level compensation signal is proposed to be the input to these cancellers. Based on the fact that this sum signal has finite levels, it can be represented in less bits than the original input to the echo and NEXT cancellers. By using a data encoding scheme, the required word-length of the modified input signal can be reduced to 8 bits. By observing the statistical distribution of the finite-level compensation signal, an improved design is proposed to further bring down the complexity and power consumption of these cancellers. Compared with the traditional design, the proposed echo and NEXT cancellers have exact input without suffering from the quantization problem, and thus they are more suitable for VLSI implementation. In practice, echo and NEXT channels are slow-varying and adaptive cancellers are needed for noise cancellation. Directly applying the proposed method to the adaptive filter design will lead to problems such as performance degradation and slow convergence. To solve these problems, a modified design of these adaptive cancellers is presented. The modified design does not suffer from the slow convergence problem. To verify the modified design, the performance in terms of residual error signal power is investigated under Cat-6 UTP channel environment. Finally, the complexity analysis is performed to show that the proposed method can save hardware cost of the echo and NEXT cancellers in a 10GBASE-T Ethernet system.

The rest of this section is organized as follows. In Section 4.3.1, a traditional design

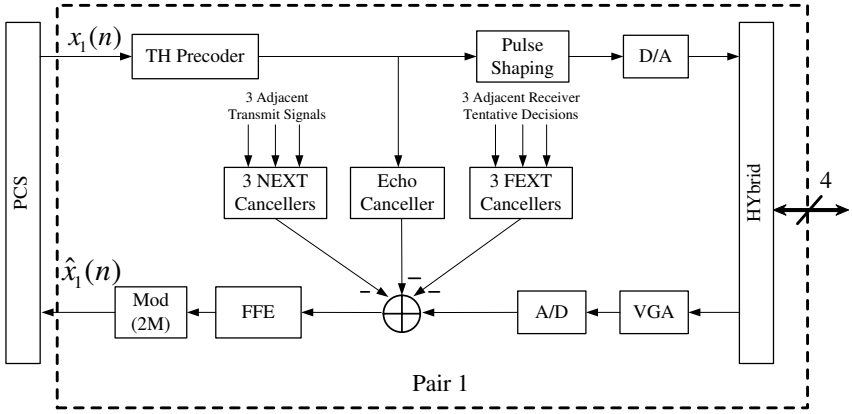


Figure 4.9: Block diagram of a typical transceiver for one pair in 10GBASE-T

of the echo and NEXT cancellers with TH precoders in the 10GBASE-T application is briefly reviewed. Section 4.3.2 describes the proposed method to reduce the word-length of the input signal to the echo and NEXT cancellers. Section 4.3.3 presents an improved design to further reduce the hardware cost of these cancellers by investigating the statistical distribution of the compensation signals. The hardware complexity analysis is provided in Section 4.3.4. In Section 4.3.5, the design issues of adaptive cancellers by using the proposed method are discussed and a modified design of the adaptive cancellers is proposed to avoid the slow convergence during the initial training stage. Simulation results and discussions are presented in Section 4.3.6.

4.3.1 Traditional Echo and NEXT Cancellers with TH precoders

Fig. 4.9 illustrates the block diagram of a typical 10GBASE-T transceiver for one pair, where one echo and three NEXT cancellers are needed for each of four channels (or four pairs). Unlike 1000BASE-T, three far end crosstalk (FEXT) cancellers are also needed to further mitigate the crosstalk interferences from the far end transmitted signals. These cancellers are usually implemented using adaptive FIR filters. In this section, we focus our

discussion on the design of low complexity echo and NEXT cancellers. In the traditional design, the input to these echo and NEXT cancellers comes from the output signal of the TH precoder associated with the same pair, *i.e.*, the signal $t(n)$ is used as the input to these cancellers, as shown in Fig. 4.10, where EC denotes echo canceller and NC denotes NEXT canceller.

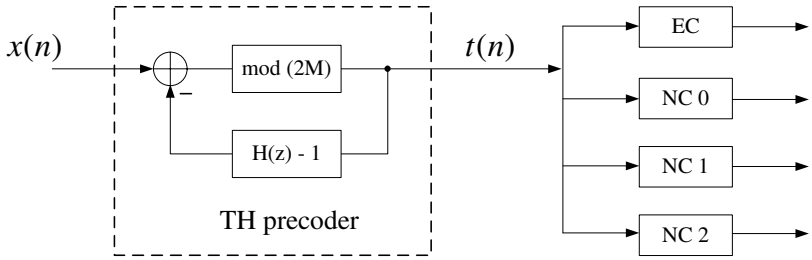


Figure 4.10: Input signal to echo and NEXT cancellers: traditional method

The TH precoder used here was first proposed by Tomlinson and Harashima in 1971 [59–61]. The operation of TH precoding can be interpreted by using the equivalent form of the TH precoder as shown in Fig. 4.11. A unique compensation signal $v(n)$, which is a multiple of $2M$, is added to the transmitted PAM-M signal $x(n)$ such that the output of the precoder $t(n)$ lies in the interval $[-M, M)$. If the input to a TH precoder, $x(n)$, is *i.i.d*, it can be shown that $t(n)$ has uniform distribution over $[-M, M)$, which means $t(n)$ is not a simple PAM-M signal any more. To reduce the echo and NEXT noise to the required level, at least 10 bits are needed to represent signal $t(n)$ [58]. However, in the design without using the TH precoder, the input to these noise cancellers is a PAM-M signal, $x(n)$, which only has M finite numbers. If assuming $M = 16$, $x(n)$ can be exactly represented with 4 bits. It can be estimated that the word-length increase from 4 bits to 10 bits would lead to 250% gate increase [58]. Since these cancellers are the largest blocks in the DSP transceiver, it is important to develop techniques to reduce the hardware cost of these echo and NEXT cancellers.

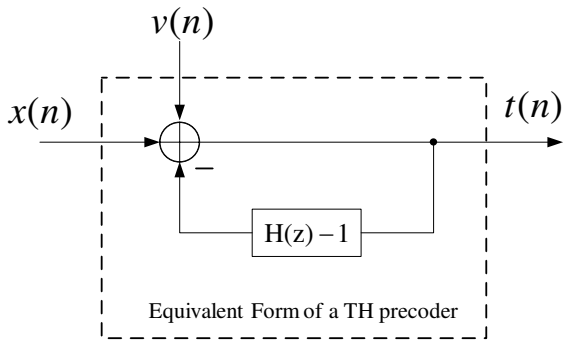


Figure 4.11: TH precoding: The equivalent form

4.3.2 Proposed Word-length Reduction Scheme

It is known that the hardware complexity and power consumption of the filter are influenced by many factors such as the number of taps used, the coefficient range of the taps, and the operating speed, *etc.*. In this section, we propose a novel method by reducing the word-length of the input signal to the echo and NEXT cancellers to achieve a low complexity design.

First, we reformulate a TH precoder into its equivalent form as shown in Fig. 4.11. The effective transmitted data sequence in Z-domain is given by

$$T(z) = \frac{X(z) + V(z)}{H(z)}, \tag{4.12}$$

where $H(z)$ is a causal FIR in the TH precoder feedback path. From (4.12), we see that a TH precoder can be viewed as an IIR filter with the input equal to the sum of the original TH precoder and a finite level compensation signal, *i.e.*, $x(n)+v(n)$. Assuming $M = 16$, the input $x(n)$ is a PAM-16 signal with symbol set $\{\pm 1,\ \pm 3,\ \ldots,\ \pm 15\}$ and can be encoded as a binary representation using 4 bits. However, the number of levels for the compensation signal, $v(n)$, is not fixed and dependent on the coefficients of the TH precoder. The range of the signal $v(n)$ can be determined by $|v(n)| \leq (1 + \sum_{i=1}^{L_H} |h_i|)M$, where L_H is length of

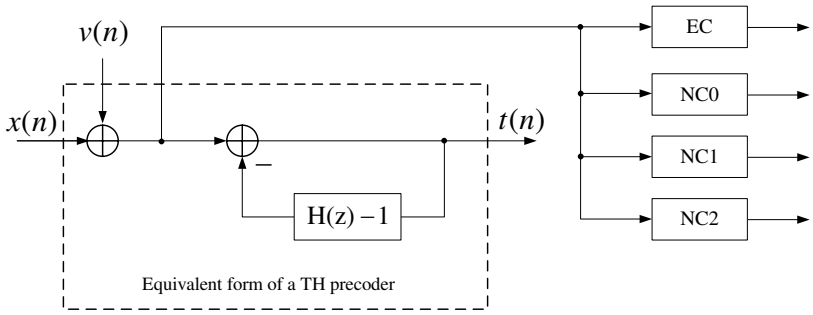


Figure 4.12: Input signal to the echo and NEXT cancellers: proposed method

the TH precoder and h_i represents the THP coefficients. According to the THP coefficients provided in [64], it is found that $v(n)$ is always in the set $\{0, \ \pm 32, \ \ldots, \ \pm 192\}$. These 13 possible numbers can be represented using 4 bits. Hence, the sum of these two signals, $x(n) + v(n)$, will have at most 16×13 possible values, which can be represented with an 8-bit binary number. Based on this key observation, the sum signal $x(n)+v(n)$ is proposed to be used as the input to the echo and NEXT cancellers instead of using the TH precoder output, $t(n)$, as shown in Fig. 4.12. The advantage is that the word-length of the input signal to these cancellers now is reduced to 8 bits, compared with 10 bits in the original input, $t(n)$. If the number of taps and the coefficient word-length of these cancellers in the traditional method and the proposed method are the same, then the overall complexity will be reduced in the proposed design. It should be noted that the modified input signal does not suffer from any quantization problem and it uses the exact values in the computation. Thus, the proposed design is more suitable for VLSI implementation.

Data Encoding Scheme

Note that the value of the sum signal, $x(n) + v(n)$, lies in the finite set with elements $\{0, \ \pm 1, \ \pm 3, \ \ldots, \ \pm 207\}$. These numbers can be converted to binary representation, for example, 2's complement representation, using only 9 bits. Thus, we can reduce the

word-length of the input signal by one bit for each tap in the echo and NEXT cancellers. However, this is not the most efficient since only 8 bits are enough to represent all the 208 numbers. Therefore, one more bit can be saved for this sum signal.

As we mentioned in Section 4.3.1, the compensation signal $v(n)$ is a multiple of $2M$, and the last five (least significant bits) LSBs are always zeros in its 2's complement representation if $M = 16$. On the other hand, 5 bits are enough to represent a PAM-16 signal. When adding these two signals, the last 5 LSBs of the sum signal are the same as those of $x(n)$, and the sum is a 9-bit number. Since all possible values of a PAM-16 signal are odd, the last bit of $x(n)$ is always one. Subtracting one from $x(n) + v(n)$ leads to a zero at its LSB. Thus, right shifting the 9-bit result leads to a unique 8-bit representation of the original sum signal. The encoding process can be formulated as:

$$q(n) = 2^{-1} \times [d(n) - 1], \tag{4.13}$$

where $d(n) = x(n) + v(n)$, and $q(n)$ represents the encoded data. Table 4.5 gives the examples of the proposed encoding scheme. Note that the encoded input signal $q(n)$ is in 2's complement representation, which means its most significant bit (MSB) is a sign bit. In fact, the encoding circuits are extremely simple and only one shifter is needed as shown in Fig. 4.13. In real implementation, this one bit right shifter does not add any hardware cost as the encoded data can be obtained by taking the first 8 MSBs of the sum signal $x(n) + v(n)$, *i.e.*, simply dropping the last LSB. The 8-bit encoded data is then applied to the echo and NEXT cancellers to compute the replica of the echo and NEXT interferences.

Table 4.5: Proposed data encoding table

$x(n)+v(n)$	Encoded bits $q(n)$	$x(n)+v(n)$	Encoded bits $q(n)$
15+192	01100111	15-192	10100111
13+192	01100110	13-192	10100110
11+192	01100101	11-192	10100101
9 +192	01100100	9 -192	10100100
7 +192	01100011	7 -192	10100011
5 +192	01100010	5 -192	10100010
3 +192	01100001	3 -192	10100001
1 +192	01100000	1 -192	10100000
-1 +192	01011111	-1-192	10011111
-3 +192	01011110	-3-192	10011110
-5 +192	01011101	-5-192	10011101
-7 +192	01011100	-7-192	10011100
-9 +192	01011011	-9-192	10011011
-11 +192	01011010	-11-192	10011010
-13 +192	01011001	-13-192	10011001
-15 +192	01011000	-15-192	10011000
$\vdots$	$\vdots$	-/-	-/-

Data Decoding Scheme

The 8-bit encoded data $q(n)$ only has the unique mapping with the sum signal $x(n)+v(n)$; however, it does not keep the actual value of the sum signal. For example, the maximum value of $x(n)+v(n)$, *i.e.*, 207, is out of range of $q(n)$, which only can represent numbers in $[-2^7, 2^7-1]$. Thus, the output of the echo and NEXT cancellers by applying the encoded input is not desirable. To get the correct result, the output of these cancellers needs to be recovered by a corresponding decoding process.

To describe the decoding scheme, we consider an N-th order FIR filter with the input $d(n) = x(n)+v(n)$. The output of this filter is given as:

$$y(n) = \sum_{k=0}^{N} g(k)d(n-k), \tag{4.14}$$

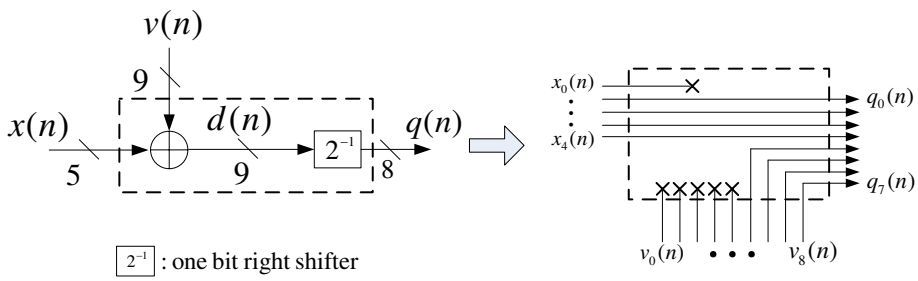


Figure 4.13: Encoding circuit for input signal

where $g(k)$ is the tap coefficient. From (4.13), we have

$$d(n) = 2 \times q(n) + 1. \tag{4.15}$$

Substituting (4.15) into (4.14), we get

$$\begin{aligned} y(n) &= \sum_{k=0}^{N} g(k)(2q(n-k)+1) \\ &= 2\sum_{k=0}^{N} g(k)q(n-k) + \sum_{k=0}^{N} g(k). \end{aligned} \tag{4.16}$$

From (4.16), we see that the output is composed of two parts: one is the weighted convolution term contributed by encoded input $q(n)$, the other is the sum of all the coefficients of the filter. If using $q(n)$ as the input to the filter, the original filter output can be easily recovered according to (4.16). As an example, Fig. 4.14 shows the overall architecture for a 3-tap FIR filter implemented in the direct form by using the proposed encoding/decoding scheme. In this figure, the input signal to each multiplier has only 8 bits, and the word-length of each delay element (register) is also reduced to 8 bits. Although there are three extra adders, the overall complexity of the proposed design is still less than the traditional design as the overall cost reduction is more than the hardware overhead of three adders. To

see this, we assume a direct form FIR structure with array multipliers and tree-structured carry ripple adders as basic functional units [65]. Then the cost of a $b1 \times b2$ bit array multiplier is approximately $b1 \times b2 \times C_{fa}$, and the cost of $b1$ bit register is $b1 \times C_{reg}$, where C_{fa} and C_{reg} are technology dependent constants. Since the word-length of the input is reduced from 10 to 8, the cost reduction for each multiplier is equal to $2 \times W_c \times C_{fa}$. Similarly, the cost reduction for each delay element is equal to $2 \times C_{reg}$. However, the overhead due to the three adders is equivalent to $3 \times W_c \times C_{fa}$. For a 3-tap FIR filter, the total cost reduction can be computed as:

$$\begin{aligned} \text{savings} &= 3(2W_cC_{fa}) + 2 \times 2C_{reg} - 2W_cC_{fa} - W_oC_{fa} \\ &= (4W_c - W_o)C_{fa} + 4C_{reg} \end{aligned} \tag{4.17}$$

where W_c is the word-length of the tap coefficients, and W_o is the word-length of the output. If the coefficients of the filter are fixed, the sum of them can be pre-computed so that only one extra adder is needed. Thus more hardware cost reduction can be expected. Generally, by using the proposed method, the complexity reduction for a N-tap FIR filter is :

$$\begin{aligned} & N(2W_cC_{fa}) + (N-1) \times 2C_{reg} - ((N-1)W_c + W_o)C_{fa} \\ = \; & (NW_c + W_c - W_o)C_{fa} + 2(N-1)C_{reg} \end{aligned} \tag{4.18}$$

From (4.18), we see that the saving will be more if the number of taps N increases.

The design in Fig. 4.14 can be generalized to the design of echo and NEXT cancellers in a 10GBASE-T ethernet system, where one echo and three NEXT cancellers are needed for each pair. Fig. 4.15 shows the proposed echo and NEXT cancellers for one pair, and the cancellers for the other three pairs will have similar architectures. Before feeding them

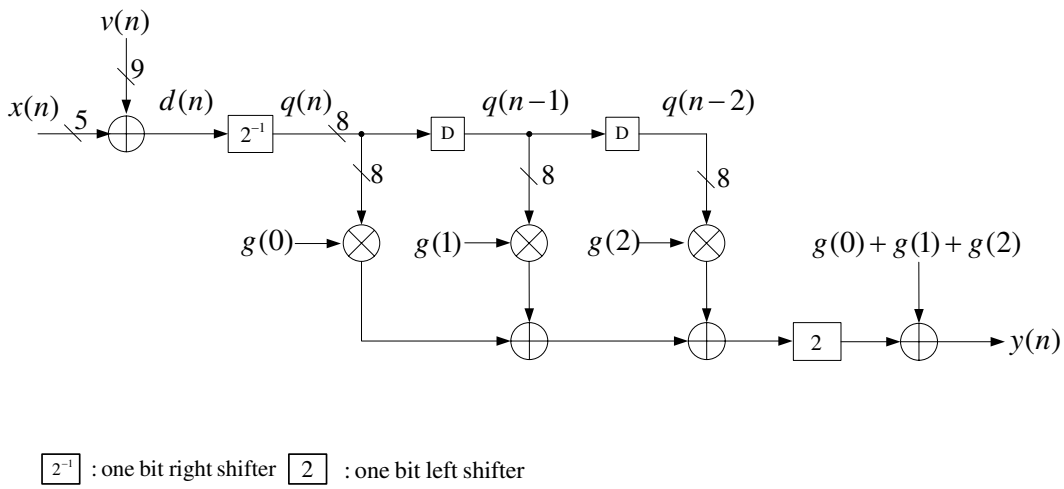


Figure 4.14: 3-tap FIR filter with word-length reduction

into the cancellers, the sum signals are first encoded into 8-bit binary numbers in their 2's complement representation. After adding the four outputs of these cancellers, the correct output is recovered by a shifter and extra adders. Since the total number of taps for echo and NEXT cancellers is very large (*i.e.*, 5600∼6800) in a 10GBASE-T system, a significant cost reduction can be obtained according to (4.18).

4.3.3 An Improved Design

In this section, we improve our design in previous Section by exploiting the statistical distribution of the finite-level compensation signal $v(n)$. It will be shown that the probability of taking each value of $v(n)$ is not equal, and some values will be taken with very high probability. A straightforward 2's complement representation of $v(n)$ in the previous design will lead to many unnecessary switching activities within the functional unit such as a multiplier [66]. To minimize these switching activities, we first investigate the property of the compensation signal, and then a sign-magnitude representation is used to represent the compensation signal instead of 2's complement representation. Finally, $v(n)$ is encoded to

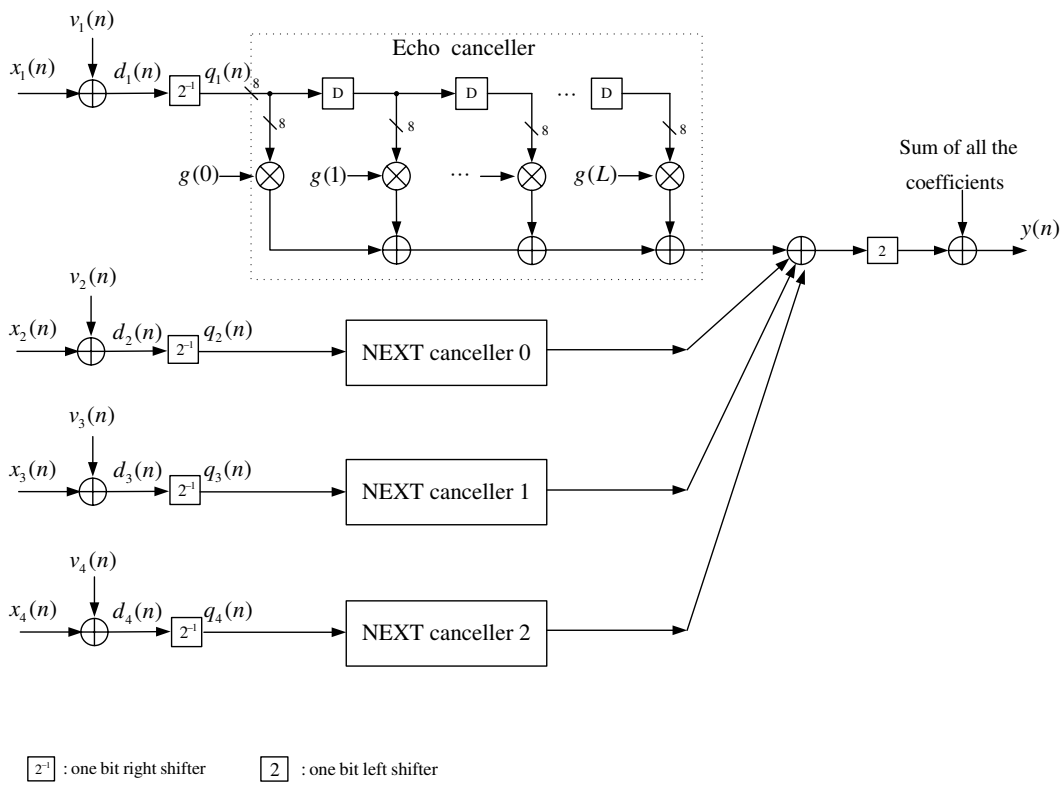


Figure 4.15: Proposed echo and NEXT cancellers with word-length reduction

further reduce the complexity of the multipliers for each tap.

From the previous discussions in Section 4.3.2, we know that the compensation signal $v(n)$ is a multiple of $2M$, and it takes values from the set $\{0, \ \pm 32, \ \ldots, \ \pm 192\}$. However, the probability of taking each value in this set is different during the operation. To illustrate this, extensive simulations are performed for different coefficients of the TH precoder corresponding to the different cable lengths. Table 4.6 lists the results for long cable, medium cable, and short cable, respectively [64]. The probabilities at each column in this table are obtained by counting the number of occurrences for each different level out of 20,000,000 random experiments. From this table, we see that $v(n)$ does not go beyond the range of $[-192, 192]$, even for the long cable case. The distribution of $v(n)$ is not uniform

Table 4.6: Distribution of the finite-level compensation signal $v(n)$ for different cables

$v(n)$	Probability		
	long (80-100m)	medium (45-80m)	short (<45m)
192	0	0	0
160	0	0	0
128	2.8001e-06	0	0
96	1.3906e-03	0	0
64	4.5535e-02	4.4435e-03	2.5000e-06
32	2.4627e-01	2.0618e-01	1.2838e-01
0	4.1307e-01	5.7869e-01	7.4320e-01
-32	2.4699e-01	2.0628e-01	1.2841e-01
-64	4.5258e-02	4.4100e-03	3.0000e-06
-96	1.4792e-03	0	0
-128	2.7401e-06	0	0
-160	0	0	0
-192	0	0	0

but symmetric, and dense around the symbol set $\{-32,\ 0,\ 32\}$. Calculating the probability of $v(n)$ lying in $\{-32,\ 0,\ 32\}$, we find that over a probability of 90%, $v(n)$ takes the values from this set for all three cases. Especially for the short cable, this probability is very close to 100%. In addition, we note that $v(n)$ is a multiple of $2M$, *i.e.*, 32 in this case and the effective values of the set $\{-32,\ 0,\ 32\}$ are $\{-1,\ 0,\ 1\}$. This implies that no multiplication is needed when multiplying any number in this set. Therefore, it is possible to further bring down the complexity and power consumption by taking advantage of these properties of $v(n)$.

To use the properties of $v(n)$, the sum signal $x(n) + v(n)$ is proposed to be separated, which leads to an architecture as shown in Fig. 4.16. In this figure, a filter with the input $x(n) + v(n)$ is implemented using two filters: one has the input $x(n)$ and the other has the input $v(n)$. The correct output can be obtained by adding the outputs of these two filters. For the filter with the input $x(n)$, we note that the input $x(n)$ is a PAM-16 signal with the symbol set $\{\pm 1,\ \pm 3,\ \ldots,\ \pm 15\}$, and at least four bits are needed to uniquely differentiate

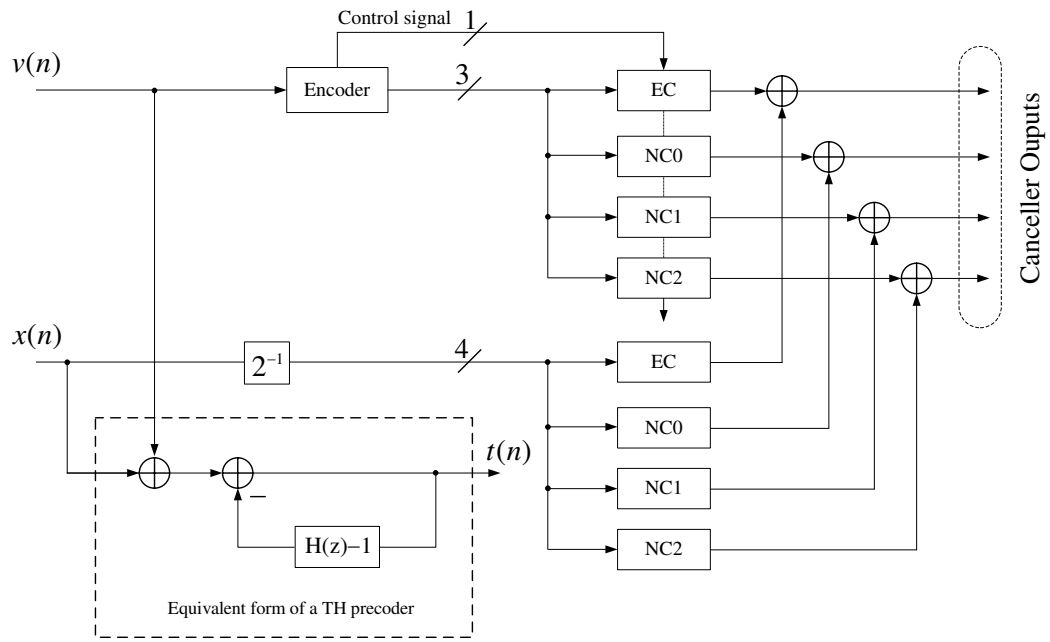


Figure 4.16: Improved design: echo and NEXT cancellers with separate inputs

different elements in this set. We also note that the probability of taking each PAM-16 symbol is equal, which infers that any representation of $x(n)$ would have the same average switching activities. Thus, the filter with the input $x(n)$ can be implemented using the idea shown in Fig. 4.14. The word-length of $x(n)$ can be finally encoded into 4 bits. For the filter with the input $v(n)$, an encoder is used to generate 3-bit encoded $v(n)$ as well as the control signal. Compared with the previous design, the advantage of this new architecture is that the $8 \times W_c$ multipliers used in the echo and NEXT cancellers are replaced by $4 \times W_c$ multipliers and $3 \times W_c$ multipliers, respectively. Thus the overall complexity can be further reduced. In addition, the power consumption due to internal switching activities can also be reduced.

As discussed above, we only consider the effective symbol set $v'(n)$, *i.e.*, $\{0, \pm 1, \ldots, \pm 6\}$, as the correct result can be easily obtained by left shifting five bits. Since the probability of taking values from this symbol set is not equal, and the probability of taking values in

$\{-1,\ 0,\ 1\}$ is much higher than values in $\{\pm 2,\ \pm 3,\ \ldots,\ \pm 6\}$, 2's complement representation of these numbers leads to higher switching activities. For example, if $v'(n)$ switches from 0 (0000_2) to -1 (1111_2), all bits will switch from 0 to 1. Even worse, this kind of transition happens a lot. Therefore, the average switching activity is more than the case where the distribution of $v'(n)$ is uniform. To reduce the average switching activity, the number of bit switches should be reduced for the symbol transitions with high probability. In our example, -1 can be represented as 1001_2 to reduce the bit switches during the transition from 0 to -1. In this representation, the MSB is the sign bit and the last three LSBs represent the magnitude of the data, which is known as sign-magnitude (S-M) representation.

Table 4.7 lists the sign-magnitude representation of $v'(n)$, as well as its 2's complement representation for comparison. It can be seen that the number of ones for those symbols with high probability is reduced, especially for negative numbers. If we note the symmetry of $v'(n)$, the word-length of the input to the multipliers can be further reduced to 3 bits while the sign bit of $v'(n)$ is used as the control signal to indicate if the result is positive or negative. Fig. 4.17 illustrates a design example for a 3-tap FIR filter with the input $v(n)$ based on the proposed idea. In this figure, $v(n)$ is first scaled down to a 4-bit number, $v'(n)$, the last 3 bits are used as the input to each multiplier. The sign bit of $v'(n)$ is used to select the correct multiplier output for each tap. It should be noted that the adders after each multiplexer are not real full adders, because the sign bit can be viewed as the carry-in bit to the accumulating adders. To illustrate this, Fig. 4.18 shows the circuits for one tap. It can be seen that the hardware overhead for one tap is one XOR gate. On the other hand, the saving obtained is equivalent to one W_c-bit full adder, as the word-length of the input to the multiplier is reduced from 4 bits to 3 bits. Therefore, the overall complexity of this design is still reduced.

Table 4.7: Data encoding table for the compensation signal $v(n)$

$v(n)$	$v'(n)$	S-M representation	2's complement
192	6	0110	0110
160	5	0101	0101
128	4	0100	0100
96	3	0011	0011
64	2	0010	0010
32	1	0001	0001
0	0	0000	0000
-32	-1	1001	1111
-64	-2	1010	1110
-96	-3	1011	1101
-128	-4	1100	1100
-160	-5	1101	1011
-192	-6	1110	1010

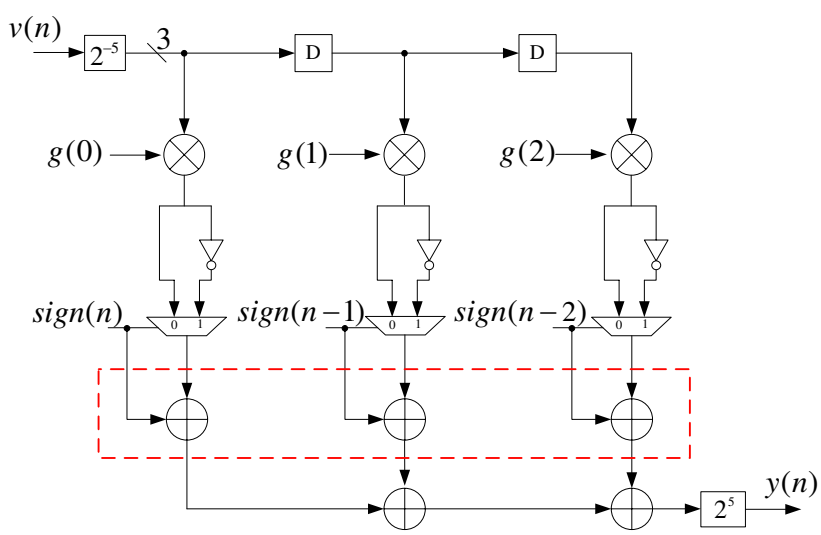


Figure 4.17: Improved design: a 3-tap FIR filter with the input $v(n)$

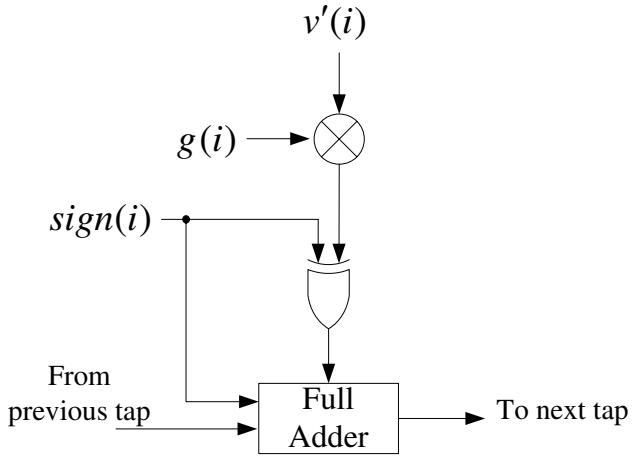
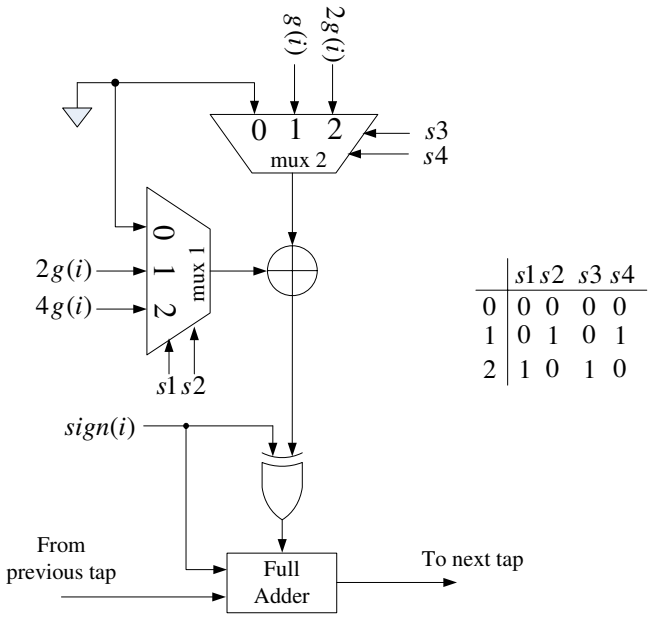


Figure 4.18: Circuit implementation for one tap



	s1	s2	s3	s4
0	0	0	0	0
1	0	1	0	1
2	1	0	1	0

Figure 4.19: The optimized design for one tap without multipliers

However, the cost of a $3 \times W_c$ multiplier is still large compared with a W_c-bit full adder. Considering the last 3 bits in the S-M representation of $v'(n)$, we find that the number of ones is at most 2. This implies only one adder is actually needed to add two partial products during the multiplication operation. Based on this fact, we propose to replace the $3 \times W_c$

multiplier with one full adder at each tap. Fig. 4.19 shows the optimized design for one tap since other taps have the same structure. In this optimized design, the multiplication of the input data and the tap coefficient is accomplished by using a full adder instead of using a multiplier. Both inputs to the full adder are from two 3-to-1 multiplexers. The inputs to these two multiplexers are 0, $g(i)$, $2g(i)$, and $4g(i)$, where $g(i)$ represents the coefficient of i-th tap. The outputs of the multiplexers are selected according to the input data. For example, if the input data is 6, $4g(i)$ from multiplexer 1 will be selected and $2g(i)$ from multiplexer 2 will be selected. Then $6g(i)$ is computed as $4g(i) + 2g(i)$ by using the full adder. It is noted that 9 combinations of multiplexer outputs are enough to cover all 6 input cases. Let A, B, C represent the last 3 bit of the encoded compensation signal $v'(n)$, then one of the possible ways to generate the control signals $s_1 \sim s_4$ is:

$$s_1 = A, \tag{4.19}$$

$$s_2 = \bar{A}B, \tag{4.20}$$

$$s_3 = AB, \tag{4.21}$$

$$s_4 = C. \tag{4.22}$$

It can be seen that the hardware overhead due to the control logic are only two AND gates. Other overhead in this optimized design are two 3-to-1 multiplexers. Compared with the cost reduction from a multiplier to a full adder at each tap, the overhead in Fig. 4.19 is still less. To explicitly see this, we take the 3-tap direct-form FIR filter as an example, and compare the hardware complexity of four different designs discussed in this section:

(1) $v(n)$ is converted to 4 bits without using S-M representation;

(2) the word-length of the input to the filter is reduced to 3 bits; however, extra XOR

gates are needed for each tap;

(3) the optimized version of design (2) requires no multipliers; however, extra multiplexers are needed compared with design (2);

(4) $v(n)$ is converted to 4 bits using S-M representation.

Based on the same assumption in Section 4.3.2, we use XOR2 (C_{xor}), 1-bit D flip-flop (C_{reg}), and basic 2-input gates (C_{gate}) such as NAND2, NOR2, *etc.*, to represent the total cost of each design. Table 4.8 lists the complexity comparison for four different designs. Note that $C_{fa} = 2 \times C_{xor} + 3 \times C_{gate}$, as one full adder is composed of two XOR gates, two AND gates and one OR gate. The hardware overhead of two 3-to-1 multiplexers used in Design (3) is obtained after synthesis using Synopsys. To evaluate the total cost of each design, we assume $C_{gate} = 32$, $C_{xor} = 56$, and $C_{reg} = 176$ based on a student cell library [67, 68] in 0.18μm CMOS technology. By setting $W_c = 15$ and $W_o = 20$, the total cost can be computed as listed in the table. It can be seen that the savings of Design (2) and Design (3) are 13.43% and 33.62%, respectively, compared with Design (1). Since the input signal has the same word-length in both Design (1) and Design (4), the hardware cost of these two designs is the same as shown in the table. It should be noted that the analysis in Table 4.8 can be easily generalized to a N-tap FIR filter case.

To verify our analysis on power reduction by using S-M representation, the 3-tap FIR filter is implemented and synthesized based on the same library mentioned above [67]. The dynamic power consumption is obtained from gate-level simulation of the synthesized netlist at 100MHz. Table 4.9 lists the area, dynamic power and timing slack for Design (1)$\sim$(4). In this table, Design (4) is compared with Design (1) by only changing the representation of the input data to observe the power consumption. It is shown that by using S-M representation in Design (4), the dynamic power consumption is reduced by about 5.7%. It is also shown

Table 4.8: Complexity comparison of four different designs

	Design (1),(4)	Design (2)	Design (3)
Multipliers ($\times C_{fa}$)	$4\times W_c\times 3$	$3\times W_c\times 3$	0
Adders ($\times C_{fa}$)	$W_o\times 2$	$W_o\times 2$	$W_o\times 2+(W_c+3)\times 3$
Registers ($\times C_{reg}$)	4×2	4×2	5×2
XOR2 gates ($\times C_{xor}$)	0	$(W_c+3)\times 3$	$(W_c+3)\times 3$
Multiplexers ($\times C_{gate}$)	0	0	72×3
Encoding logic ($\times C_{gate}$)	0	0	2
Total cost	47168	40832	31312
Saving	-/-	13.43%	33.62%

Table 4.9: Implementation results of a 3-tap FIR filter

	Design (1)	Design (2)	Design (3)	Design (4)
Area	34480	30000	25160	34480
Dynamic Power (mW)	37.97	33.61	30.46	35.82
Timing Slack (ns)	1.15	1.13	0.77	1.15

that by reducing the input word-length by 1 bit from Design (1) to Design (2), both area and power consumption are reduced. Finally, the proposed Design (3) has the minimum power consumption and area as expected. These results also demonstrate our complexity analysis in Table 4.8. It should be noted that the increased critial path in Design (3) can be easily reduced by using pipelining techniques [85].

4.3.4 Complexity Analysis

We presented the proposed method and its improved designs to reduce the complexity of the echo and NEXT cancellers. In the proposed method, the complexity reduction is due to the word-length reduction of the input signal to these cancellers. By modifying the input signal using the sum signal $x(n) + v(n)$, the input word-length is reduced to 9 bits from 10 bits without any hardware overhead. Based on the proposed encoding scheme, the word-length can be further reduced to 8 bits; however, extra adders are needed to recover the correct

Table 4.10: Complexity comparison between traditional and proposed methods in the 10GBASE-T application

	Traditional Design (a) (10-bit input) [58]	Proposed Design (b) (9-bit input)	Proposed Design (c) (8-bit input)	Improved Design (d) (optimized version)
Echo canceller	560354	508859.5	457365	320268.5
NEXT cancellers (×3)	1008644	915960.5	823277	576387.5
Hardware overhead	0	0	136493.5	325859.5
Total cost for one pair	1568998	1424820	1417135.5	1222515.5
Total cost (×4)	6275992	5810066	5668542	4890062
Saving	-/-	9.19%	9.68%	22.08%

output. In the improved designs, further complexity reduction is achieved by exploiting the properties of the compensation signal $v(n)$. In Section 4.3.3, we show that by separating the input $x(n) + v(n)$, the filters with the input $v(n)$ can be implemented without using multipliers. Thus, the overall hardware complexity can be reduced. In this section, the hardware complexity of the echo and NEXT cancellers for a typical 10GBASE-T system is analyzed by applying the traditional method and the proposed methods.

In a typical 10GBASE-T system, we assume the number of taps in each echo canceller is 500, and the number of taps in each NEXT canceller is 300 [7]. We also assume the word-length of tap coefficient in both echo and NEXT canceller is $W_c = 15$, and the word-lengths of the accumulating output in echo and NEXT cancellers are both 14. Without loss of generality, the technology dependent constants are normalized with respect to C_{gate}, *i.e.*, $C_{gate} = 1$, $C_{xor} = 1.75$, and $C_{reg} = 5.5$ in the analysis.

Table 4.10 lists the overall hardware complexity of the echo and NEXT cancellers for different designs:

(a) traditional design without modifying the input to these cancellers;

(b) proposed design by using the sum signal $x(n) + v(n)$ as the input signal to these cancellers, where the word-length of input signal is 9;

(c) proposed design by using the data encoding scheme, where the word-length of the input signal is 8;

(d) improved design with optimized version discussed in Section 4.3.3

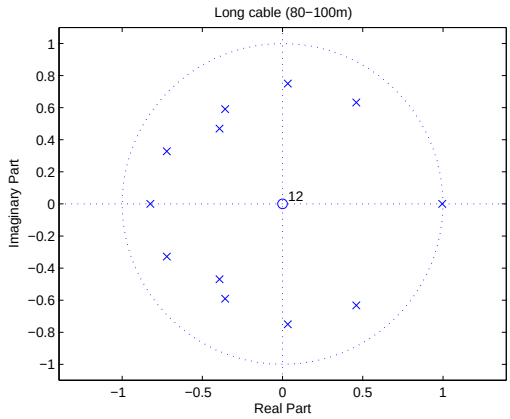
From this table, we can see that by using the proposed method, the overall complexity of the echo and NEXT cancellers can be effectively reduced. Compared with proposed design (b), proposed design (c) saves one more bit in input data, and thus the complexity is reduced.

However, the hardware saving is not obvious due to the hardware overhead of extra adders. In proposed design (d), no multipliers are needed in the cancellers with the input $v(n)$. Therefore, large savings can be achieved although extra XOR gates and multiplexers are introduced. Table 4.10 demonstrates that 22.08% hardware saving is achieved compared with the traditional design.

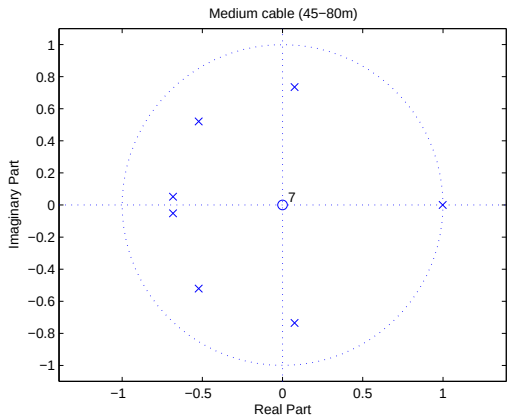
4.3.5 Proposed Word-length Reduction Scheme in the Adaptive Filter

In real applications such as 10GBASE-T Ethernet systems, echo and NEXT cancellers need to operate in adaptive mode for noise cancellation due to the slowly varying noise channels. Directly applying the proposed method to the adaptive filter design will lead to performance loss, as these adaptive cancellers implicitly contain an IIR filter with a pole close to the unit circle in Z-plane as shown in Fig. 4.20, where each figure corresponds to long cable, medium cable and short cable, respectively. If the number of taps in FIR-based cancellers is not long enough to get a good approximation, the performance of noise cancellation will be degraded. In addition, the convergence speed of the resulting adaptive cancellers will be slow, as the eigenvalues of the correlation matrix of the input signals will be widely spread after modifying the original input signal. Thus the time taken to converge will be very long [84]. To solve these problems, a modified design of these adaptive cancellers based on the proposed method is presented in this section.

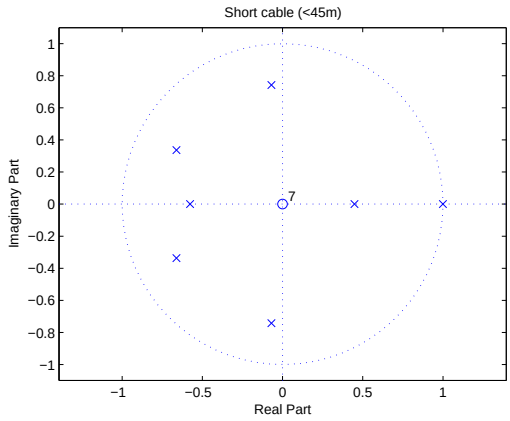
In the following, we only consider the design of an adaptive echo canceller. However, our discussion is also applied to adaptive NEXT cancellers since the architecture of NEXT cancellers is similar to an echo canceller except that the inputs are replaced by the transmitted signals from the adjacent transmitters. Fig. 4.21 shows a traditional design of an adaptive echo canceller. In this figure, the received signal y is corrupted by additive white noise w and the interference from echo channel g. Let t_{far} denote the transmitted symbol



(a) Long cable case



(b) Medium cable case



(c) Short cable case

Figure 4.20: Poles of the TH Precoder for different cases

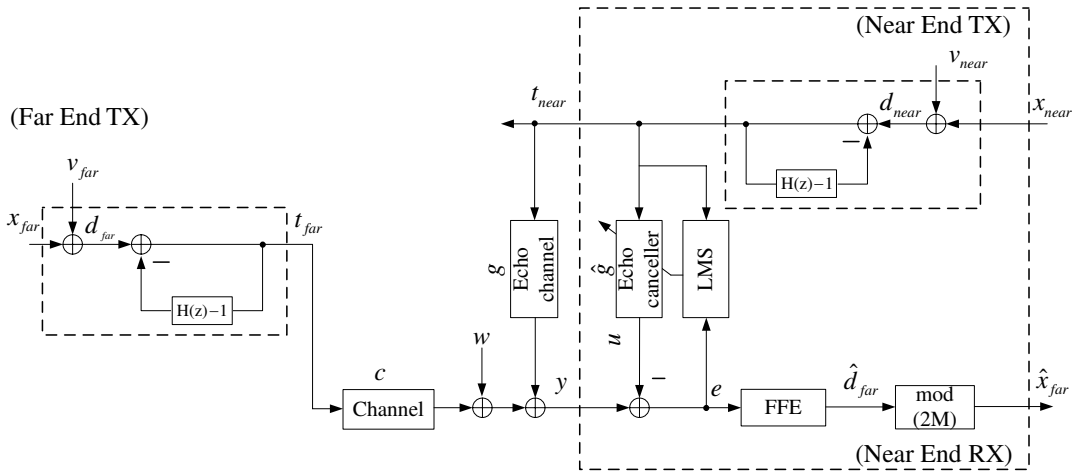


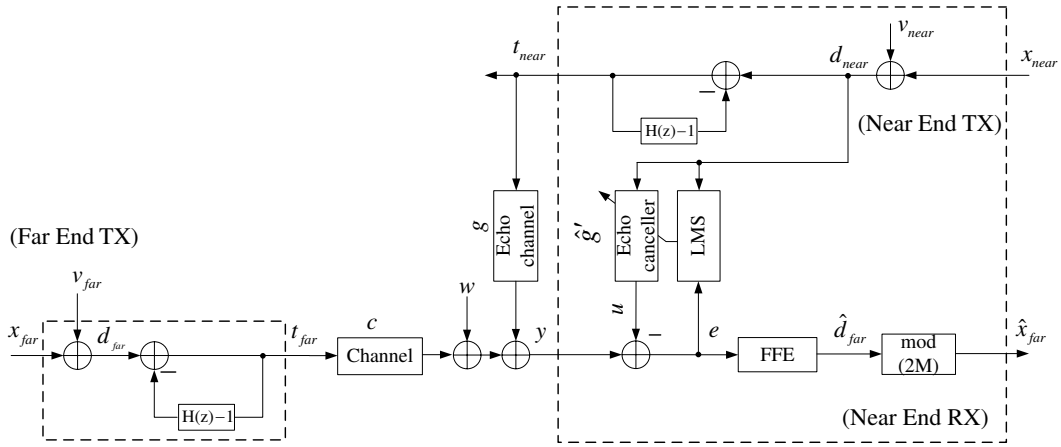
Figure 4.21: Traditional Design: adaptive echo canceller without NEXT crosstalk

sequence from the far end transmitter and t_{near} denote the transmitted symbol sequence from the near end transmitter. Then the received signal y can be represented as:

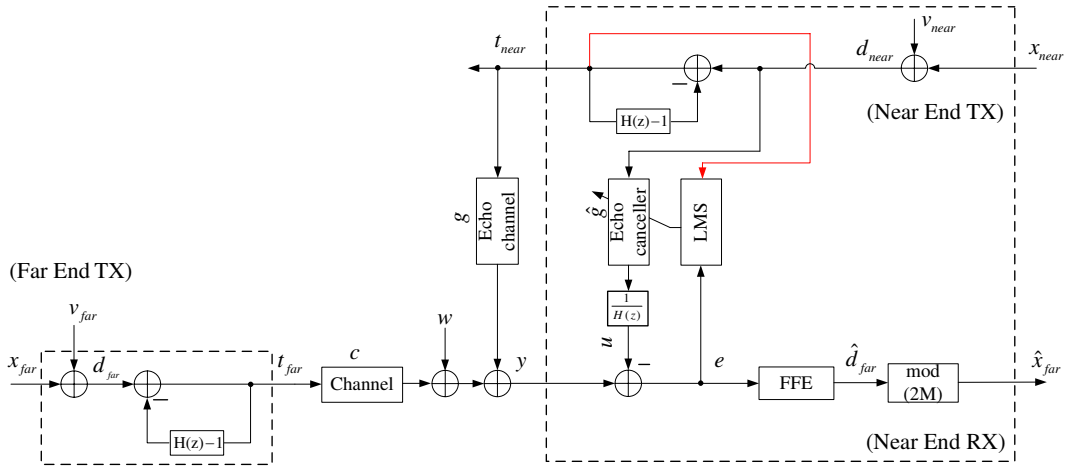
$$y = c \otimes t_{far} + g \otimes t_{near} + w, \tag{4.23}$$

where $\otimes$ represents convolution operation, and c denotes channel impulse response. To cancel the noise term $g \otimes t_{near}$, an adaptive echo canceller is used with the same input t_{near}. As we can see, an adaptive filter consists of two parts: one is the filter part and the other is the weight update part. By using a recursive algorithm (*e.g.*, least-mean square [84]), the coefficients of the echo canceller, $\hat{g}$, can be updated to approach the echo channel g, until the error signal $e = g \otimes t_{near} - \hat{g} \otimes t_{near}$ is minimized. Then the echo cancellation is achieved by subtracting the output of the echo canceller from the received noisy signal y. After echo cancellation, a feed-forward equalizer (FFE) is used to remove the ISI. Finally, the transmitted signal from far end is recovered by a corresponding modulo device [29].

If directly applying the proposed method into the design of adaptive echo canceller, we can have a design as shown in Fig. 4.22(a), where the input to the adaptive echo canceller



(a) Direct application of the proposed method in adaptive echo canceller



(b) Modified Design: adaptive echo canceller without NEXT crosstalk

Figure 4.22: Application of the proposed method in adaptive echo canceller

is changed to $d_{near} = x_{near} + v_{near}$ and the coefficients of the adaptive echo canceller are assumed to be $\hat{g}'$. The output of the echo canceller can be represented as:

$$u = \hat{g}' \otimes d_{near}. \tag{4.24}$$

Representing (4.23) and (4.24) in Z-domain, we can have

$$Y(z) = C(z)T_{far}(z) + G(z)T_{near}(z) + W(z), \tag{4.25}$$

and

$$U(z) = \hat{G}'(z)D_{near}(z). \tag{4.26}$$

Comparing these two equations, we see that the ideal echo cancellation can be achieved when

$$\hat{G}'(z)D_{near}(z) = G(z)T_{near}(z). \tag{4.27}$$

From (4.12), equation (4.27) can be represented as

$$\hat{G}'(z)D_{near}(z) = \frac{G(z)D_{near}(z)}{H(z)}. \tag{4.28}$$

Thus,

$$\hat{G}'(z) = \frac{G(z)}{H(z)}. \tag{4.29}$$

From (4.29), we can see that the canceller, $\hat{G}'(z)$, to be designed in Fig. 4.22(a) implicitly contains an IIR filter $H(z)$. To approximate this filter well using FIR-based adaptive filter, the number of taps in the FIR filter has to be very large to achieve the required noise cancellation, especially when the poles of $H(z)$ are close to unit circle in the Z-plane. If

in this case, the benefit from word-length reduction technique will be countered due to the increased taps of the FIR filter, and also the convergence speed will be slow for long adaptive FIR filter. On the other hand, system performance may degrade as the FIR filter is only an approximation of the IIR filter.

Fig. 4.22(b) shows the modified design, which does not suffer from the problems such as instability, slow convergence speed and performance degradation in noise cancellation. From Fig. 4.22(b), we see that an IIR filter, $\frac{1}{H(z)}$, with fixed coefficients is inserted into the echo cancellation path after the output of the echo canceller. In addition, t_{near} is used as the input to the weight update part in the adaptive echo canceller. In this modified design, the output of the echo canceller, $U(z)$, can be represented as:

$$U(z) = \frac{\hat{G}(z)D_{near}(z)}{H(z)} = \hat{G}(z)T_{near}(z), \tag{4.30}$$

and then the error signal can be computed as:

$$E(z) = G(z)T_{near}(z) - \hat{G}(z)T_{near}(z). \tag{4.31}$$

If using least-mean square (LMS) algorithm, the weight update equation can be represented as:

$$\hat{G}(z)_{n+1} = \hat{G}(z)_n + \mu E(z)_n T_{near}(z)_n, \tag{4.32}$$

where $\hat{G}(z)_n$ denotes the n-th iteration $\hat{G}(z)$ and similar definition for $E(z)$ and T_{near}.

From (4.30)-(4.32), we see that the modified design and the traditional design in Fig. 4.21 are mathematically equivalent. Thus the performance of the modified design is the same as the traditional design. However, unlike the traditional design, the modified design inherits the advantage of the proposed method such that it can lead to a low complexity

design. As we can see from Fig. 4.22(b), the sum signal d_{near} is used as the input to the filter part of the adaptive echo canceller instead of t_{near}. By applying the proposed method in Previous Sections 4.3.2 and 4.3.3, the hardware complexity of the filter part in the adaptive echo canceller can be reduced. Since the inserted IIR filter has less than 15 taps [64], the hardware overhead of the IIR filter is negligible in practical applications. Therefore, the overall hardware cost of the adaptive echo and NEXT cancellers will be reduced in a 10GBASE-T system.

4.3.6 Simulations and Discussion

In this section, simulations are performed to evaluate the modified adaptive echo and NEXT cancellers in a 10GBASE-T system. We compare the convergence speed and residual error signal power of three different designs discussed in Section 4.3.5. In addition, the hardware complexity analysis is also performed to show the hardware complexity reduction of the resulting modified adaptive echo and NEXT cancellers by using the proposed word-length reduction technique.

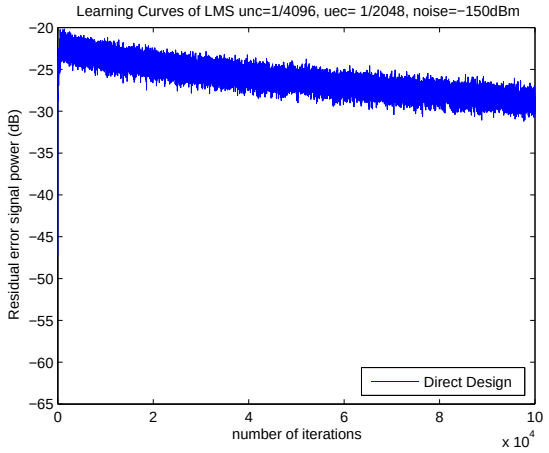
Our simulation is performed under Cat-6 unshielded twisted-pair channel environment. The channel models (*e.g.*, insertion loss, return loss, and NEXT crosstalk) with different lengths can be obtained from the IEEE 802.3an website [6]. For simplicity, FEXT crosstalk is not considered in our simulation, because it does not affect the performance of echo and NEXT cancellers. A PAM-2 signal is used as the training symbol while data symbol is assumed to be PAM-16. Other parameters used during the simulation are summarized in Table 4.11.

First, we compare the convergence speed of three designs by applying the LMS algorithm [84]. The reason to choose LMS algorithm is its low complexity and it is easy to implement. In order to give a fair comparison, fixed step sizes $\mu_{ec} = 1/2048$ and $\mu_{nc} = 1/4096$ are used

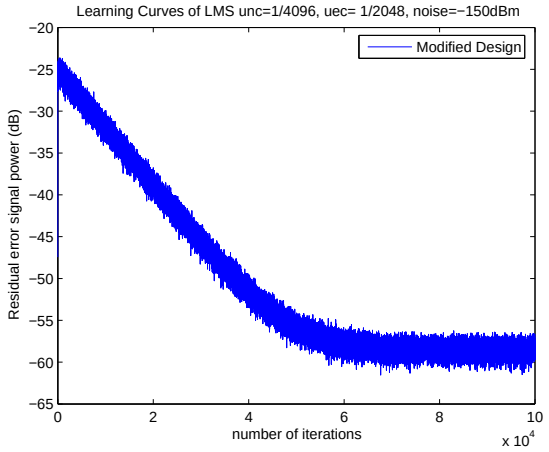
Table 4.11: Parameter Setup

Parameter	Values
FFE taps	64
THP taps	13
EC taps	500
NC taps	300
Modulation	PAM-16
AWGN	-150dBm
TX Power	5dBm
Band Width	400MHz

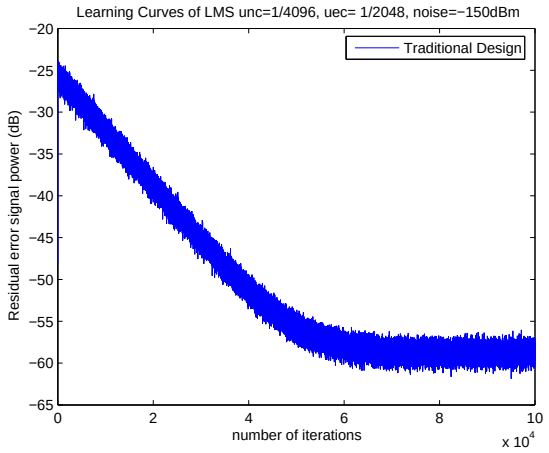
for all three designs. The simulation is performed based on Cat-6 100m channel model. The learning curves of the corresponding designs are plotted in Fig. 4.23. In these figures, each learning curve is obtained by averaging the results over 80 independent experiments. From Fig. 4.23(a), we see that by directly applying the proposed method in adaptive cancellers, the convergence speed is very slow and the performance difference is obvious compared with the traditional design. It will be shown later that the performance loss is large at the steady state although this design can save hardware cost of the resulting adaptive cancellers. However, the modified design does not suffer from the slow convergence problem. As we can see from Fig. 4.23(b), the convergence speed is greatly improved. Comparing Fig. 4.23(b) with Fig. 4.23(c), we see that the modified design has similar convergence speed as the traditional design. In addition, the performance difference between these two designs is negligible when both designs converge to the steady stage. These results in fact verify our previous analysis in Section 4.3.5.



(a)



(b)



(c)

Figure 4.23: Comparison of the convergence speed for different designs:(a) Direct design, (b) Modified design, (c) Traditional design

Table 4.12: Fixed point simulation setup

	FFE			THP			Echo canceller			NEXT canceller		
	W_i	W_c	W_o	W_i	W_c	W_o	W_i	W_c	W_o	W_i	W_c	W_o
Traditional Design (Fig. 4.21)	14	14	14	8	8	10	10	15	14	10	15	14
Direct Design (Fig. 4.22(a))	14	14	14	8	8	10	8	15	14	8	15	14
Modified Design (Fig. 4.22(b))	14	14	14	8	8	10	8/10	15	14	8/10	15	14

Next, we compare the noise cancellation performance of these designs in terms of residual error signal power after echo and NEXT noise cancellation. The residual error signal power is measured at the input to the FFE during the steady state by performing fixed point simulation. Table 4.12 lists word-length setup in the fixed-point simulation for three different designs. In this table, W_i denotes the word-length of the input data, W_c denotes the word-length of the coefficients, and W_o denotes the word-length of the output data. It should be noted that in the modified design, the input to the filter part of the echo and NEXT cancellers is changed to the sum signal d_{near} as discussed in Section 4.3.5 and thus it can be represented with 8 bits or less by using the proposed word-length reduction technique; however, the input to the weight update part of these cancellers is unchanged (*i.e.*, 10 bits). Similarly, in the direct application of the proposed method (Fig. 4.22(a)), the inputs to both filter part and weight update part in these adaptive cancellers are changed to the sum signal d_{near} and both can be represented with 8 bits or less.

Table 4.13 shows the performance comparison in terms of residual error signal power at the input to the FFE. Each different design is evaluated based on Cat-6 measured and scaled channel models with different lengths: 100m, 75m, and 55m respectively. Scaled channel models are usually obtained by scaling the measured channel models to the theoretical limits according to the cable requirement in 10GBASE-T. If the echo and NEXT noise can

be cancelled to the required level in scaled channel models, it will be certain in measured channel cases. As we can see from this table, the performances of different designs in measured channel cases are better than those in scaled channel cases. As we mentioned before, the direct design (Fig. 4.22(a)) exhibits worse performance than the traditional design and the performance degradation is obvious. We also note that it is sensitive to the cable lengths. This indicates that for short cables, the implicit IIR contained in the direct design requires longer FIR filter to approximate. If the number of taps in FIR filters is fixed, the residual echo and NEXT noise level will be large, thus the residual error signal power is higher in short cable cases. However, in the modified design, the performance is not affected by the cable lengths. As we can see from the table, it has similar residual error signal power for all three cable lengths. If comparing the modified design with the traditional design, we can find that both have very similar performance in different channel models. In addition, the modified design has the exact input signal $x(n) + v(n)$ without suffering from the quantization errors. Thus, it is shown that the modified design even has a slightly better performance than the traditional design.

Table 4.13: Residual error signal power at the inputs to the FFE for different designs, AWGN=-150dBm

CAT-6 UTP Cable Type	Residual noise level at the input to the FFE (dB)		
	Traditional Design (Fig. 4.21)	Direct application (Fig. 4.22(a))	Modified Design (Fig. 4.22(b))
Scaled 100m	-57.03	-46.89	-57.06
Scaled 75m	-57.10	-39.67	-57.11
Scaled 55m	-57.06	-33.08	-57.07
Measured 100m	-63.25	-54.63	-63.28
Measured 75m	-63.28	-49.14	-63.36
Measured 55m	-63.27	-42.78	-63.27

Finally, we compare the hardware complexity of the resulting adaptive cancellers by using these different designs, unlike in Table 4.10, where the comparison is made without

considering the weight update part in the echo and NEXT cancellers. In our analysis, each weight update unit is composed of a multiplier, an adder and a delay element. The word-lengths of the adder and the delay element should be the same as the word-length of the coefficients for each tap. We use the same assumption in Table 4.10 when calculating the hardware cost of the multipliers and adders. We also assume the same normalized technology constants, *i.e.*, $C_{gate} = 1$, $C_{xor} = 1.75$, and $C_{reg} = 5.5$ in the analysis.

Table 4.14: Hardware Complexity Comparison of three designs

	Traditional Design (Fig. 4.21)	Direct Design (Fig. 4.22(a))	Modified Design (Fig. 4.22(b))
Echo canceller (filter part)	560354	320268.5	320268.5
Echo canceller (update part)	545000	281750	545000
NEXT canceller (filter part)	1008644	576387.5	576387.5
NEXT canceller (update part)	981000	507150	981000
Hardware overhead	0	783834.5	337339.5
Total cost for one pair	3094998	2469390.5	2759995.5
Total cost (×4)	12379992	9877562	11039982
Saving	-/-	20.21%	10.82%

Table 4.14 shows the hardware complexity comparison of the resulting adaptive echo and NEXT cancellers by applying different architectures. From this table, we can see that the modified design can save the hardware cost of these cancellers about 10.82%. This indicates that the overhead of the inserted IIR filter is less than the overall hardware reduction. By direct application of the proposed word-length reduction technique (Fig. 4.22(a)), more saving can be achieved, since the hardware cost of both filter part and weight update part in the adaptive filters is reduced. However, we should note that the more hardware reduction is achieved at the expense of performance degradation. Therefore, the modified design of the adaptive echo and NEXT cancellers is preferred in practical applications.

4.4 Further Cost Reduction of Adaptive Echo and NEXT Cancellers

In previous section, we have proposed a new word-length reduction scheme which can reduce the the hardware cost of adaptive echo and NEXT cancellers by about 10.82% without any performance loss. However, this hardware cost reduction is mainly due to the hardware cost reduction in the filter part of these adaptive cancellers. The problem of reducing hardware cost of the weight update part in these adaptive cancellers remains unsolved.

This section considers the low complexity design of the weight update part in adaptive echo and NEXT cancellers. We propose a new cost-efficient scheme for weight update part in these adaptive filters. By using the proposed weight-updating scheme, the input to the weight update part is replaced by $v(n)$ which only has finite possible values. Thus, the previous word-length reduction technique in Section 4.3 can be applied to further reduce the overall hardware cost of these adaptive cancellers. It should be noted that the proposed scheme is general and can be applied to multiple-input-multiple-output (MIMO) systems such that hardware cost of both echo and NEXT cancellers can be reduced. We apply the proposed scheme to the 10GBASE-T Ethernet system, and show that the proposed architecture can further reduce hardware cost of echo and NEXT cancellers.

The rest of the section is organized as follows. In Section 4.4.1, our previous work on low complexity echo and NEXT cancellers is briefly reviewed. In Section 4.4.2, a new cost-efficient adaptive scheme for weight updating is presented. Based on the proposed adaptive scheme, a cost-efficient architecture of adaptive echo and NEXT cancellers is developed. In Section 4.4.3, we apply the proposed method to the 10GBASE-T Ethernet system. Computer simulation and hardware cost comparison are presented.

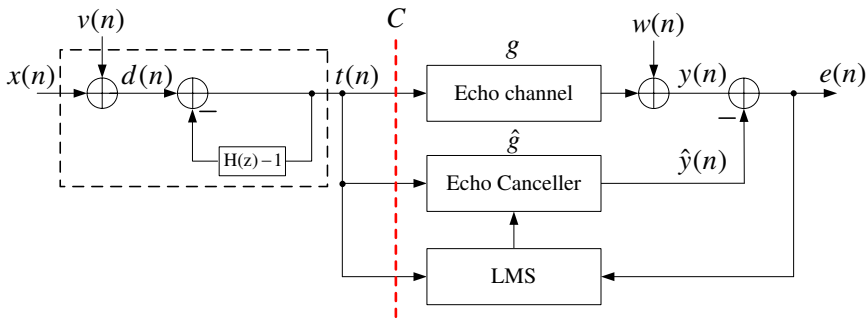


Figure 4.24: Adaptive echo canceller: traditional method

4.4.1 Previous Designs of Echo and NEXT Cancellers in 10GBASE-T

In this section, we briefly review previous design presented in Section 4.3.5. Similarly, we only consider adaptive echo cancellers to illustrate the proposed design. However, the proposed design can be applied to adaptive NEXT cancellers since the architecture of NEXT cancellers is similar to that of echo cancellers except that the inputs are replaced by the transmitted signals from the adjacent transmitters.

For the purpose of illustration, the traditional design of an adaptive echo canceller shown in Fig. 4.21 is redrew as shown in Fig. 4.24. As we can see, the adaptive echo canceller consists of two parts: one is the filter part and the other is the weight update part. By using a recursive algorithm (*e.g.*, least-mean square [84]), the coefficients of the echo canceller, $\hat{g}$, can be updated to approach the echo channel g, until the error signal $e(n)$ is minimized. Then the echo cancellation is achieved by subtracting the output of the echo canceller $\hat{y}(n)$ from the received signal $y(n)$ which is corrupted by additive white noise $w(n)$.

A direct application of the idea shown in Fig. 4.12 to the design of adaptive echo canceller will lead to an unacceptable performance degradation, as well as slow convergence speed. In previous section, a modified structure for adaptive echo canceller is developed to solve these problems. Fig. 4.25 illustrates the proposed design without showing the far

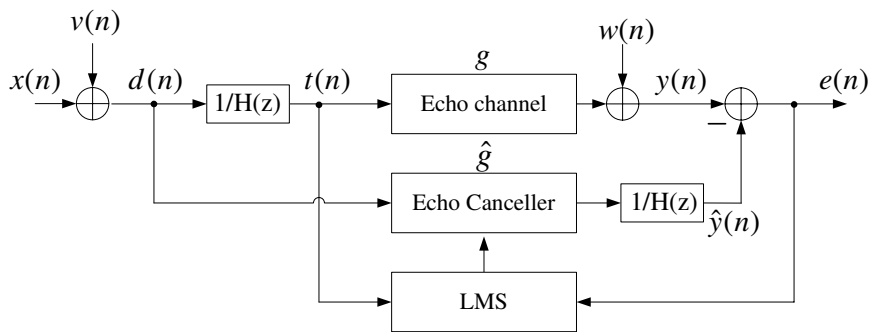


Figure 4.25: Adaptive echo canceller: proposed method

end transmitter part. In this figure, we see that an IIR filter, $\frac{1}{H(z)}$, with fixed coefficients is inserted into the echo cancellation path after the output of the echo canceller. By doing so, it can be shown that this modified design and the traditional design in Fig. 4.24 are mathematically equivalent. Thus, the performance of the modified design is the same as the traditional design. In addition, we note that the sum signal $d(n)$ is used as the input to the filter part of the adaptive echo canceller instead of $t(n)$. By applying the proposed word-length reduction scheme in previous Sections 4.3.2 and 4.3.3, the hardware complexity of the filter part in the adaptive echo canceller can be reduced. Since the inserted IIR filter has less than 15 taps [64], the hardware overhead of the IIR filter is negligible in practical applications.

4.4.2 Proposed Cost-efficient Weight Update Scheme

From previous discussions, we know that hardware cost reduction of the adaptive echo canceller in Fig. 4.25 is due to hardware cost reduction of the filter part in the adaptive filter by applying the word-length reduction technique. However, hardware cost of the weight update part in the adaptive filter does not change because the input to the weight update part is still $t(n)$, which is uniform in $[-M, M)$ such that the word-length reduction

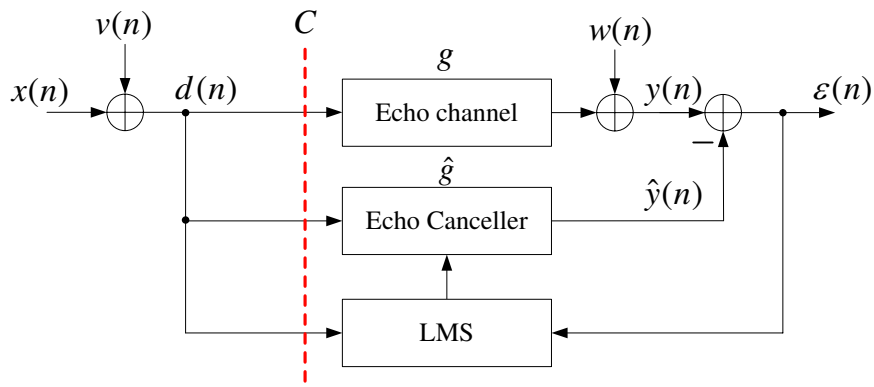


Figure 4.26: Adaptive echo canceller with modified input $d(n)$

technique can not be applied. In this section, we propose a new method to reduce the hardware complexity of the weight update part in the adaptive filter. First, a new cost-efficient weight update scheme is derived, and then experimental results are presented to illustrate the performance of the proposed design. By using the proposed design, the word-length of the input signal to the weight update part can be reduced. Thus, the overall hardware complexity of the proposed method can be reduced.

Consider the traditional architecture of adaptive echo canceller in Fig. 4.24. If we insert a short FIR filter $H(z)$ at the feed-forward cutset C denoted by dashed line in Fig. 4.24, we will obtain a design shown in Fig. 4.26. In this figure, the input to the adaptive echo canceller is changed to the sum signal $d(n)$ rather than $t(n)$, thus the word-length reduction technique can be applied to reduce the hardware complexity of the adaptive echo canceller. However, the design in Fig. 4.26 is not useful in real applications because the signal transmitted to the physical channel is also changed by inserting $H(z)$.

To solve this problem, we apply the retiming technique [85] to the design in Fig. 4.26. The resulting architecture is shown in Fig. 4.27. From this figure, we see that the physical echo channel path remains the same as the traditional design in Fig. 4.24. In addition,

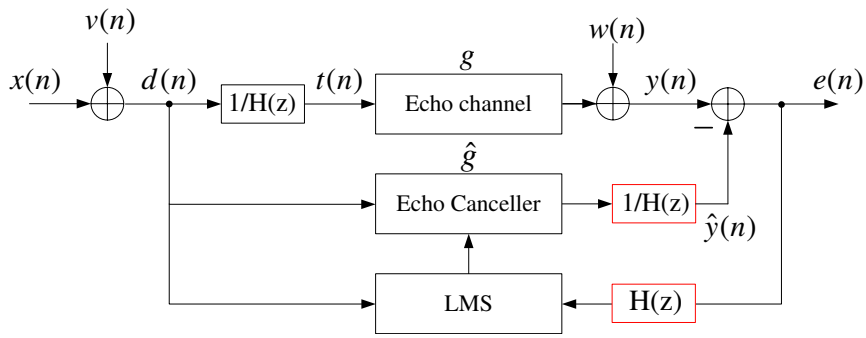


Figure 4.27: Filtered error adaptive echo canceller

the sum signal $d(n)$ is used as the input to both filter part and weight update part of the adaptive echo canceller. Compared with the design in Fig. 4.25, since the input to the weight update part of the adaptive filter is changed to $d(n)$, the hardware complexity of the weight update part of the adaptive echo canceller can be reduced by applying the word-length reduction technique.

In order to minimize the mean square error, $E[|e(n)|^2]$ and also to track the time-varying channel environment, the LMS adaptive algorithm is applied to the proposed design. Since the update error signal $e(n)$ is filtered before the weight update part, the resulting adaptive algorithm is usually referred as filtered error LMS algorithm [69], and it can be summarized as:

$$\hat{y}(n) = [\hat{\mathbf{g}}(n)^H \mathbf{d}(n)] * h^{-1}, \tag{4.33}$$

$$e(n) = y(n) - \hat{y}(n), \tag{4.34}$$

$$\hat{\mathbf{g}}(n+1) = \hat{\mathbf{g}}(n) + \mu \mathbf{d}(n) \left[e(n) * h \right], \tag{4.35}$$

where $\hat{\mathbf{g}}(n)$ denotes tap weights of the adaptive echo canceller at n-th interation, $\mathbf{d}(n)$ is the input vector at n-th iteration, h^{-1} denotes the equivalent impulse response of the IIR filter

$\frac{1}{H(z)}$, h denotes impulse response of the FIR filter $H(z)$, μ is the step size, and $*$ denotes convolution operation.

Noting that $d(n)$ is the sum of $x(n)$ and $v(n)$, we can write (4.35) as

$$\hat{y}(n) = [\hat{\mathbf{g}}(n)^H \mathbf{x}(n) + \hat{\mathbf{g}}(n)^H \mathbf{v}(n)] * h^{-1}, \tag{4.36}$$

which means the echo canceller in Fig. 4.27 can be implemented as the sum of two filters: one has the input $x(n)$ and the other has the input $v(n)$. In general, equation (4.36) can be rewritten as

$$\hat{y}(n) = [\hat{\mathbf{g}}_1(n)^H \mathbf{x}(n) + \hat{\mathbf{g}}_2(n)^H \mathbf{v}(n)] * h^{-1}, \tag{4.37}$$

where $\hat{\mathbf{g}}_1(n)$ represents the weights of the filter with the input $x(n)$, and $\hat{\mathbf{g}}_2(n)$ represents the weights of the filter with the input $v(n)$. Then the error signal $e(n)$ can be expressed as

$$e(n) = y(n) - [\hat{\mathbf{g}}_1(n)^H \mathbf{x}(n)] * h^{-1} - [\hat{\mathbf{g}}_2(n)^H \mathbf{v}(n)] * h^{-1}. \tag{4.38}$$

To minimize $E[|e(n)|^2]$, modified weight update equations can be derived as [84]

$$\hat{\mathbf{g}}_1(n+1) \;=\; \hat{\mathbf{g}}_1(n) + \mu_1 \mathbf{x}(n)\,[e(n) * h]\,, \tag{4.39}$$

and

$$\hat{\mathbf{g}}_2(n+1) \;=\; \hat{\mathbf{g}}_2(n) + \mu_2 \mathbf{v}(n)\,[e(n) * h]\,, \tag{4.40}$$

where μ_1 and μ_2 are step sizes for the corresponding adaptive filters.

From (4.39) and (4.40), we see that the sum of the two adaptive filters can be adapted

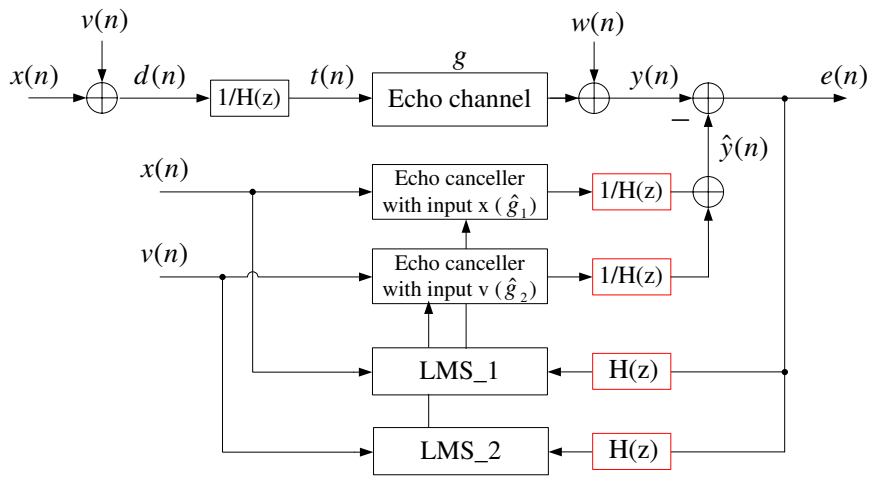


Figure 4.28: Adaptive echo cancellers with separate inputs

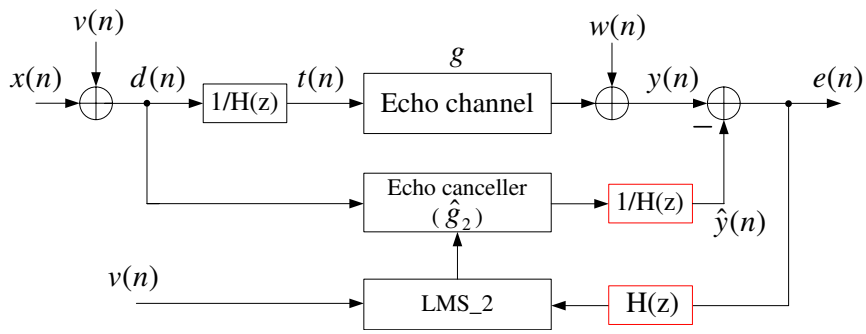


Figure 4.29: Proposed adaptive echo canceller with cost-efficient weight updating

separately with their own input signals. Based on this observation, we can obtain an architecture for the adaptive echo canceller as shown in Fig. 4.28. In this figure, we see that the adaptive echo canceller is implemented by two separate parts with the corresponding input $x(n)$ and $v(n)$, respectively. The correct output of the echo canceller can be obtained by adding outputs of these two filters. The coefficients of two filters $\hat{\mathbf{g}}_1(n)$ and $\hat{\mathbf{g}}_2(n)$ can be updated individually using (4.39) and (4.40).

However, it is not efficient to update both $\hat{\mathbf{g}}_1(n)$ and $\hat{\mathbf{g}}_2(n)$ every iteration. By considering the optimal coefficients $\hat{\mathbf{g}}_1^{opt}$ and $\hat{\mathbf{g}}_2^{opt}$ after convergence, we find that both of them will converge to the same optimal value, *i.e.*, $\hat{\mathbf{g}}_1^{opt} = \hat{\mathbf{g}}_2^{opt}$. This implies that only one of the

equations in (4.39) and (4.40) is needed for weight updating. Based on this fact, we can use either $x(n)$ or $v(n)$ as the input to the weight update part in adaptive filter, and then use the updated coefficients in both filters for filtering operation. The advantage is that the hardware complexity of the weight update part can be further reduced because $x(n)$ or $v(n)$ can be encoded to have fewer bits than the sum signal $d(n)$.

By exploiting the property of the compensation signal $v(n)$, we found that it could be encoded to have fewer bits than $x(n)$ [5]. If $v(n)$ is used as the input to the weight update part, the hardware complexity can be reduced more. On the other hand, we note that $v(n)$ dominates the eigenvalue spread of the sum signal $d(n)$. It is better to use $v(n)$ instead of $x(n)$ to avoid the performance degradation. Thus, we propose to use $v(n)$ as the input to the weight update part in the adaptive echo canceller as shown in Fig. 4.29. In this figure, every iteration the coefficients are updated according to (4.40) with the input $v(n)$, and then the updated coefficients are used in the filter part with the input $d(n)$. Compared with the design in Fig. 4.25, the hardware complexity of weight update part can be further reduced.

Next, we provide experimental results to illustrate the performance of the proposed weight-updating scheme. In Fig. 4.29, $x(n)$ is assumed to be a PAM-16 signal, and $w(n)$ is assumed to be Gaussian White noise with zero mean and variance 0.00003. For simplicity, the echo channel is assumed to be a 5-tap FIR filter with randomly generated coefficients. In addition, the coefficients of the TH precoder corresponding to the short cable are used as the coefficients of $H(z)$.

Fig. 4.30 shows different learning curves by applying the different input signals to the weight update part in the adaptive echo canceller. In this figure, each learning curve is obtained by averaging the results over 20 independent experiments. In order to give a fair

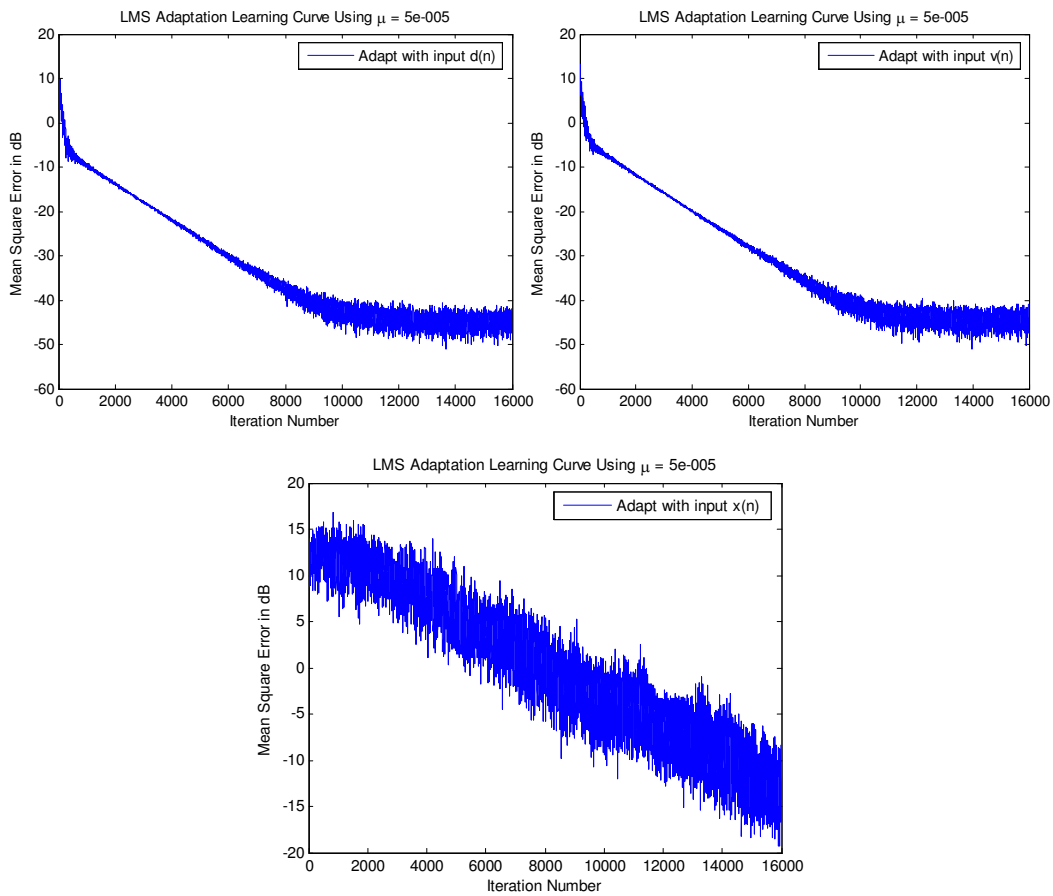


Figure 4.30: Comparison of learning curves of the proposed design with different adapting signals

comparison, a fixed step size $\mu = 5e - 5$ is used for all three cases. From the figure, we see that the convergence speed of the proposed weight update scheme with input $v(n)$ is similar as the design with input $d(n)$. By observing the mean square error (MSE) during the steady state, we find that the performance difference is negligible. However, if using $x(n)$ as the input to the weight update part in Fig. 4.29, we see that the convergence speed is slow. This is because the input statistical distribution is greatly changed by replacing $d(n)$ using $x(n)$ in the weight update part.

For comparison, the performances of the traditional design (Fig. 4.24) and our previous

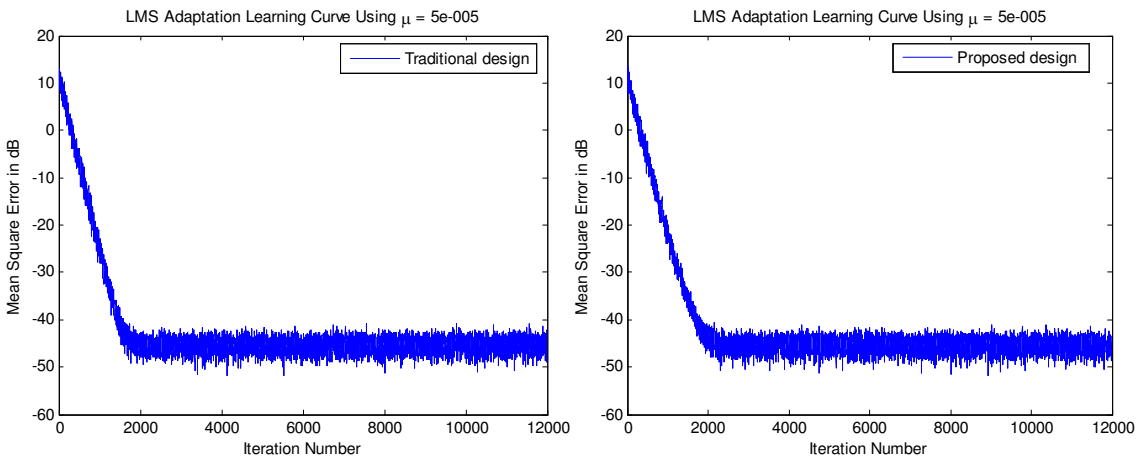


Figure 4.31: Performances of the tradition design (Fig. 4.24) and our previous design (Fig. 4.25)

design (Fig. 4.25) are also presented in Fig. 4.31. From this figure, we see that our previous design has the similar performance as the traditional design, which again demonstrates our previous work. If comparing the results shown in Fig. 4.30, we see that the proposed weight update scheme with input $v(n)$ exhibits a slower convergence speed. This is because the eigenvalues of the correlation matrix of the input signal in the proposed design is widely spread after changing the original input signal. According to [84], the time taken to converge will be longer for larger eigenvalue spread. In addition, we notice that the performance penalty of the proposed weight update scheme is negligible. It should be noted that the proposed method is general and can be applied to NEXT cancellers in the application of 10GBASE-T.

4.4.3 Computer Simulations and Hardware Comparison

In this section, we apply the proposed method to the application of 10GBASE-T. Fixed point simulations are performed to evaluate the system performance in terms of the residual error signal power by applying the proposed scheme. In addition, the hardware complexity

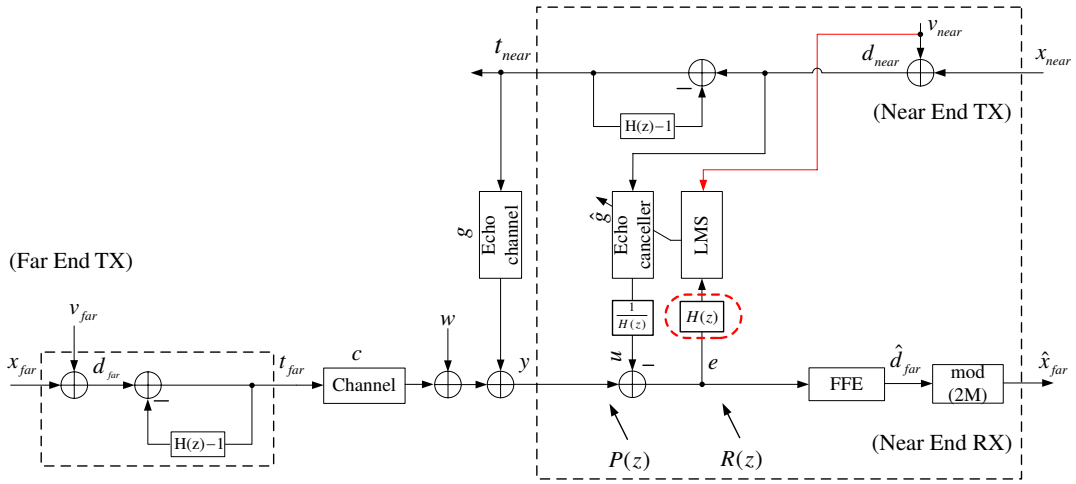


Figure 4.32: The proposed transceiver architecture of one pair in a 10GBASE-T Ethernet system

analysis is performed to show the complexity reduction of the proposed architecture for adaptive echo and NEXT cancellers by using the word-length reduction technique.

Fig. 4.32 shows the proposed architecture of the 10GBASE-T transceiver for one pair. The simulation is performed under Cat-6 unshielded twisted-pair (UTP) channel environment. The measured data of channel models with different lengths can be obtained from the IEEE 802.3an website [6]. For simplicity, FEXT crosstalk is not considered in our simulation, because it does not affect the performance of echo and NEXT cancellers. A PAM-2 signal is used as the training symbol while data symbol is assumed to be PAM-16. In addition, the transmit power is assumed to be 5dBm, AWGN power is assumed to be -150dBm. We also assume the number of taps in each echo canceller is 500, the number of taps in each NEXT canceller is 300, the number of taps in each TH precoder is 13, and the number of taps in each feed-forward equalizer (FFE) is 64.

Table 4.15 shows the performance comparison in terms of residual error signal power at the input to the FFE. In the simulation, we use the same word-length setup as shown in Table 4.12, and evaluate the proposed design over Cat-6 measured channel models with

Table 4.15: Residual error signal power at the inputs to the FFE for different designs, AWGN=-150dBm

Measured Cat-6 UTP Cable	Residual noise level at the input to the FFE (dB)		
	Traditional Design in Fig. 4.21	Modified Design in Fig. 4.22(b)	Proposed Design in Fig. 4.32
100m	−63.25	−63.28	−61.71
75m	−63.28	−63.36	−62.16
55m	−63.27	−63.27	−62.17

different lengths: 100m, 75m, and 55m respectively. As we can see from this table, the proposed design has around 1.5dB performance penalty compared with the traditional design and our previous design in previous section.

To evaluate the hardware cost of the proposed design, we use XOR2 (C_{xor}), 1-bit D flip-flop (C_{reg}), and basic 2-input gates (C_{gate}) such as NAND2, NOR2, *etc.*, to represent the total cost of the adaptive echo and NEXT cancellers, where C_{xor}, C_{reg}, and C_{gate} are technology dependent constants. For example, a one-bit full adder can be represented as $C_{fa} = 2 \times C_{xor} + 3 \times C_{gate}$. Furthermore, we assume the same normalized technology constants as in Section 4.3.2, *i.e.*, $C_{gate} = 1$, $C_{xor} = 1.75$, and $C_{reg} = 5.5$.

Table 4.16 shows the hardware complexity comparison of the resulting adaptive echo and NEXT cancellers by applying different architectures. From this table, we can see that the proposed architecture can save the hardware cost of total adaptive cancellers about 42.02%, while the previous design in [5] can only save about 10.08%. The further cost reduction is mainly due to the word-length reduction of the input signal at the weight update part. As we can see from the table, the hardware cost of the weight update part in both echo and NEXT cancellers has been greatly reduced in the proposed architecture.

Table 4.16: Hardware Complexity Comparison of three designs

	Traditional Design in Fig. 4.21	Modified Design in Fig. 4.22(b)	Proposed Design in Fig. 4.32
Echo canceller (filter part)	560354	320268.5	320268.5
Echo canceller (update part)	545000	545000	**145250**
NEXT canceller (filter part)	1008644	576387.5	576387.5
NEXT canceller (update part)	981000	981000	**261450**
Overhead	0	337339.5	491269.5
Cost for one pair	3094998	2759995.5	1794625.5
Total cost ($\times 4$)	12379992	11039982	7178502
Saving	-/-	10.82%	**42.02%**

4.5 Conclusion

In this chapter, the problem of low complexity and low power designs of echo and NEXT cancellers has been addressed in the application of 10GBASE-T ethernet systems. First, we have proposed a new adaptive tap management algorithm to design echo and NEXT cancellers by exploiting sparse characteristics of the Cat-6 UTP copper channel. The proposed algorithm exhibits a significant computational complexity reduction as well as a faster initial convergence speed, compared with conventional LMS. An area saving architecture is also presented to implement our new algorithm. Experiment results show that, by using the proposed method in gigabit transceiver design, we are able to achieve 50% and 29∼66.7% computation cost reduction in terms of multiplication during the initial training stage and steady stage respectively. The benefit from the computational complexity reduction can lead to a power efficient design in the application of 10GBASE-T.

Then, we have proposed a new word-length reduction scheme for low complexity design of echo and NEXT cancellers. Based on the equivalent form of a THP precoder, we

propose to replace the original input signal to the echo and NEXT cancellers with a finite signal which is the sum of the original signal and a finite compensation signal. In order to reduce the word-length of this modified input signal, a data encoding and decoding scheme is developed with little hardware overhead. Furthermore, the statistical properties of the compensation signal $v(n)$ is studied and an improved design is proposed to further bring down the hardware cost. It is shown that this improved design can also be optimized for a low power design by minimizing internal switching activities. Compared with the traditional design, the proposed echo and NEXT cancellers have exact input without suffering from the quantization problem, and thus they are more suitable for VLSI implementation. The proposed method can also be applied to design adaptive echo and NEXT cancellers. However, direct application leads to an unacceptable performance degradation and slow convergence speed. To solve these problems, a modified design is developed. This modified design can achieve similar performance as the traditional design. In addition, it inherits the advantage of the proposed method such that it can lead to a low complexity design. It is shown that, the resulting adaptive echo and NEXT cancellers by using the modified design can save hardware cost by about 10.82%. Detailed simulations under the Cat-6 UTP channel environment also verify the effectiveness of the modified design.

Finally, we have presented a novel weight-updating scheme to further reduce the hardware cost of echo and NEXT cancellers in the application of 10GBASE-T. Unlike our previous design, the proposed scheme can reduce the hardware cost of the weight update part in adaptive echo and NEXT cancellers, thus the overall hardware cost of these cancellers can be further reduced. It is shown that, by applying the proposed method to the gigabit transceiver design, we are able to reduce the hardware cost by about 42.02% only with about 1.5dB performance penalty, compared with the traditional design.

In a summary, our proposed methods can be applied to the design of adaptive echo and NEXT cancellers in a 10GBASE-T system. By using the proposed approaches, the overall hardware cost of these cancellers can be reduced. The hardware complexity reduction can also lead to the saving of the dynamic power consumption, as $P_{dynamic} = \alpha C_L V_{dd} f_{clk}$, where C_L is the load capacitance, V_{dd} is the supply voltage, and f_{clk} is the clock frequency [66]. Especially, during the normal data transmission mode, the power consumption of filter part in these cancellers is dominant because the weight update part operates in a much slower speed than the filter part. Thus, larger hardware cost reduction in filter part will lead to more savings of overall power consumption.

Chapter 5

Design of Stable IIR Echo and NEXT Cancellers

5.1 Introduction

In many applications, finite impulse response (FIR) filters are preferred due to their advantages over infinite impulse response (IIR) filters. For example, an FIR filter is always stable and less sensitive to the quantization effect. In addition, an FIR filter can be designed to have a strictly linear phase and the FIR filtering operation can be efficiently computed by applying the Fast Fourier Transform (FFT) algorithm [83]. However, to meet a given magnitude-response specification, an FIR filter usually requires a larger number of taps than an IIR filter. Thus the hardware cost of an FIR filter is more expensive than that of an IIR filter as the hardware cost in terms of chip area and power consumption is directly related to the total number of taps in the filter. If the phase response is not important, approximating a long FIR filter by an IIR filter (or pole-zero filter) with a smaller number of taps including numerator and denominator will reduce the overall hardware cost. Because

an IIR may not be stable, it is not trivial to find a stable IIR filter corresponding to a given long FIR filter.

Previous studies on this problem are mostly focused on the decision feedback equalization application [74, 75]. Specifically, in [74], an adaptive IIR algorithm was developed to iteratively compute the coefficients of the approximate IIR filter. To avoid the local optima and make stability easy to monitor, the approximate IIR was assumed to have only two poles or less, which limited its application in cases where a two-pole model is not adequate. However, this restriction was eliminated in [75], where a *Generalized* ARMA-Levinson algorithm was developed to compute the coefficients of the approximate IIR filter. This algorithm is computationally efficient and also applicable to the model of IIR filters with unequal numbers of poles and zeros.

Motivated by previous work, in this chapter, we propose a new efficient method for computing the optimal coefficients of the stable IIR filter, assuming that the coefficients of the target FIR filter are known. Unlike the existing approaches, we first formulate an optimization problem which minimizes the mean-square error (MSE) between the target FIR filter and the approximate IIR filter. This MMSE problem can be solved by use of many methods in [23, 24, 26]. By exploiting the structured matrices in the MMSE solution, the computational complexity can be further reduced. The proposed method is general and does not make any assumption on the number of poles and zeros of the approximate IIR filter. Numerical results show that the proposed method is as accurate as the *Generalized* ARMA-Levinson algorithm but with much lower computational complexity.

The rest of the chapter is organized as follows. In Section 5.2, the proposed approach for stable IIR filter modeling of long FIR filters is presented in detail, and a complexity analysis is performed to compare the proposed method with the *Generalized* ARMA-Levinson

algorithm. In Section 5.3, numerical results are presented and discussed. Finally, Section 5.4 concludes the work of this chapter.

5.2 The Proposed Method

In this section, the problem formulation is first introduced and then the MMSE solution is derived by exploiting the structured matrices. The computational complexity of the proposed method is also analyzed and compared with the *Generalized* ARMA-Levinson algorithm [75].

5.2.1 Problem Formulation

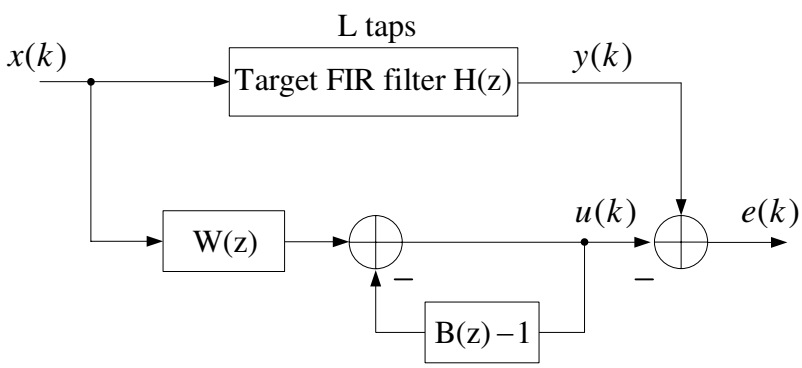


Figure 5.1: A generalized system identification model

We consider a generalized system identification model as shown in Fig. 5.1. The target long FIR filter is assumed to be $H(z)$ with L taps, and the approximate system is assumed to an IIR filter, $\frac{W(z)}{B(z)}$, with q zeros and p poles, respectively.

Let $x(k)$ denote the input signal to both filters, then the output of the target FIR filter $H(z)$ can be expressed as

$$y(k) = \sum_{i=0}^{L-1} h(i)x(k-i), \tag{5.1}$$

where $h(i), i = 0, \ldots, L-1$ represents the coefficient of the target long FIR filter.

Similarly, the output of the approximate IIR filter $\frac{W(z)}{B(z)}$ can be expressed as

$$u(k) = \sum_{i=0}^{q} w(i)x(k-i) - \sum_{j=1}^{p} b(j)u(k-j), \tag{5.2}$$

where $w(i), i = 0, \ldots, q$ represents the coefficient of the numerator part of the IIR filter, and $b(i), j = 1, \ldots, p$ represents the coefficient of the denominator part of the IIR filter. If we define the coefficient vectors $\mathbf{w} \triangleq [w(0), w(1), \ldots, w(q)]^H$, and $\mathbf{b} \triangleq [b(1), b(2), \ldots, b(p)]^H$, equation (5.2) can be written as

$$u(k) = \mathbf{w}^H \mathbf{x}(k) - \mathbf{b}^H \mathbf{u}(k-1), \tag{5.3}$$

where the superscript H denotes Hermitian transposition, $\mathbf{x}(k)$ is a $(q+1) \times 1$ input vector

$$\mathbf{x}(k) = [x(k), x(k-1), \ldots, x(k-q)]^H,$$

and $\mathbf{u}(k-1)$ is a $p \times 1$ symbol vector containing the past p symbols

$$\mathbf{u}(k-1) = [u(k-1), u(k-2), \ldots, u(k-p)]^H.$$

The error signal $e(k)$ in Fig. 5.1 is then given by

$$\begin{aligned} e(k) &= y(k) - u(k) \\ &= y(k) - \mathbf{w}^H \mathbf{x}(k) + \mathbf{b}^H \mathbf{u}(k-1). \end{aligned} \tag{5.4}$$

Our objective is to approximate the target long FIR filter $H(z)$ by the IIR filter $\frac{W(z)}{B(z)}$.

This can be achieved by adjusting the coefficients of the IIR filter, $\mathbf{w}$ and $\mathbf{b}$, such that the mean square error $E[|e(k)|^2]$ is minimized. In this paper, the well-known MMSE criterion is applied to formulate this minimization problem, *i.e.*,

$$\begin{aligned} \text{min.} \quad & E[|e(k)|^2] \\ \text{sub.} \quad & H(z) \ \text{ is known,} \end{aligned} \tag{5.5}$$

where $E[\cdot]$ denotes the expectation operator.

To solve this problem, recursive methods by applying adaptive algorithms such as least mean square (LMS) algorithm [84] can be used. However, the IIR filter structure shown in Fig. 5.1 will degrade the performance of the LMS algorithm. For example, the convergence speed will be slow and the optimal minimum may not be always obtained. In addition, the stability of the algorithm can not be always guaranteed [57, 74]. In this paper, we derive a computationally efficient solution for this problem by exploiting the structured matrix.

5.2.2 Optimal MMSE Solution

In the following analysis, we assume the input signal $x(k)$ is *i.i.d.* with unit variance, *i.e.*, $\sigma_x^2 = 1$. We also assume previous p estimates are correct, *i.e.*, $u(k-i) = y(k-i)$ for $1 \leq i \leq p$. Then (5.4) can be rewritten as

$$e(k) \ = \ y(k) - \begin{bmatrix} \mathbf{w}^H & -\mathbf{b}^H \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{y}(k-1) \end{bmatrix}, \tag{5.6}$$

where $\mathbf{y}(k-1) = [y(k-1), y(k-2), \ldots, y(k-p)]^H$. From (5.1), we can represent $\mathbf{y}(k-1)$ as

$$\mathbf{y}(k-1) \;=\; \mathbf{H}\bar{\mathbf{x}}(k-1), \tag{5.7}$$

where $\mathbf{H}$ is the $p \times (L+p)$ convolution matrix

$$\mathbf{H} = \begin{bmatrix} h_0 & h_1 & \ldots & h_{L-1} & 0 & \ldots & 0 \\ 0 & h_0 & h_1 & \ldots & h_{L-1} & \ldots & 0 \\ \vdots & & \ddots & & & \ddots & \\ 0 & \ldots & 0 & h_0 & h_1 & \ldots & h_{L-1} \end{bmatrix},$$

and $\bar{\mathbf{x}}(k-1)$ is a $(p+L) \times 1$ input vector

$$\bar{\mathbf{x}}(k-1) = [x(k-1), x(k-2), \ldots, x(k-p-L)]^H.$$

Applying the *orthogonality principle* [23, 25], we get

$$E\left\{ \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{y}(k-1) \end{bmatrix} e(k)^H \right\} = \mathbf{0}. \tag{5.8}$$

Substituting (5.6) into (5.8), we have

$$\begin{bmatrix} R_{xx} & R_{xy} \\ R_{yx} & R_{yy} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ -\mathbf{b} \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}, \tag{5.9}$$

and then the optimal coefficients of the approximate IIR filter can be obtained from

$$\begin{bmatrix} \mathbf{w} \\ -\mathbf{b} \end{bmatrix} = \begin{bmatrix} R_{xx} & R_{xy} \\ R_{yx} & R_{yy} \end{bmatrix}^{-1} \cdot \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}, \tag{5.10}$$

where

$$\begin{aligned}
R_{xx} &= E[\mathbf{x}(k)\mathbf{x}^H(k)] = I_{(q+1)}, \\
R_{yy} &= E[\mathbf{y}(k-1)\mathbf{y}^H(k-1)] = \mathbf{H}\mathbf{H}^H, \\
R_{yx} &= R_{xy}^H, \\
R_{xy} &= E[\mathbf{x}(k)\mathbf{y}^H(k-1)] \\
&= E[\mathbf{x}(k)\mathbf{x}^H(k-1)\mathbf{H}^H] \\
&= \begin{bmatrix} \mathbf{0}_{1\times q} & \mathbf{0}_{1\times(L+p-q)} \\ \mathbf{I}_{q\times q} & \mathbf{0}_{q\times(L+p-q)} \end{bmatrix} \cdot \mathbf{H}^H \\
&= \begin{bmatrix} 0 & 0 & \dots & 0 \\ h_0 & 0 & \dots & 0 \\ h_1 & h_0 & 0 & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ h_{q-1} & h_{q-2} & \dots & h_{q-p} \end{bmatrix}_{(q+1)\times p},
\end{aligned}$$

$$
\begin{aligned}
\theta_1 &= E\left[\mathbf{x}(k)y(k)^H\right] \\
&= \begin{bmatrix} \mathbf{I}_{(q+1)\times(q+1)} & \mathbf{0}_{(q+1)\times(L-q-1)} \end{bmatrix} \cdot \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_L \end{bmatrix} \\
&= \begin{bmatrix} h_0, \, h_1, \, \ldots, \, h_q \end{bmatrix}^H,
\end{aligned}
$$

and

$$
\begin{aligned}
\theta_2 &= E\left[\mathbf{y}(k-1)y(k)^H\right] \\
&= E\left[\mathbf{H}\bar{\mathbf{x}}(k-1)y(k)^H\right] \\
&= \mathbf{H} \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_L \\ \mathbf{0}_{p\times 1} \end{bmatrix}.
\end{aligned}
$$

Unlike the optimal solution derived for MMSE decision feedback equalizers in [26, 28], the MMSE solution derived above is unrelated with the decision delay. In addition, it is noted that θ_2 is not equal to zero, which limits the further simplification of (5.10) as was done in [26]. Thus the fast algorithm proposed in [26, 28] for computing the MMSE solution is difficult to apply.

On the other hand, directly performing the matrix inversion will be computationally expensive if the size of the block matrix in (5.10) is large. In order to reduce the size of matrix involved in inversion operation, we apply the matrix inversion formula [76] to (5.10),

and get

$$\begin{bmatrix} \mathbf{w} \\ -\mathbf{b} \end{bmatrix} = \begin{bmatrix} R_{xx}^{-1} & \mathbf{0}_{(q+1)\times p} \\ \mathbf{0}_{p\times(q+1)} & \mathbf{0}_{p\times p} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} + \begin{bmatrix} -R_{xy} \\ \mathbf{I}_{p\times p} \end{bmatrix} \Phi^{-1} \begin{bmatrix} -R_{yx}, & \mathbf{I}_{p\times p} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}, \tag{5.11}$$

where Φ is a $p \times p$ symmetric matrix

$$\Phi = R_{yy} - R_{yx}R_{xy}.$$

It can be seen that the size of the matrix involved in inversion operation has been reduced to $p \times p$. By applying fast algorithms on Cholesky factorization [24,25], the matrix inversion operation can be performed efficiently in $O[p^2]$ operations.

It should be noted that the MMSE solution in (5.11) has no restrictions on the number of poles p and the number of zeros q in the approximate IIR filter. In other words, the proposed solution is general for different values of p and q. Similar to the *Generalized* ARMA-Levinson algorithm, the generated IIR filter by using the proposed solution is always guaranteed to be stable. This can be shown by using the facts that the denominator $B(z)$ has all the roots strictly inside the unit circle [77].

5.2.3 Computational Complexity Comparison

Next, the computational complexity of the proposed solution is analyzed. For the purpose of comparison, the computational complexity analysis of the *Generalized* ARMA-Levinson algorithm is also presented. In the analysis, we assume the computational complexity is dominated by the multiplication operations, and then calculate the number of multiplica-

Table 5.1: Computational complexity of the proposed solution

Computation	Multiplication (for $p > q$)	Multiplication (for $p \leq q$)
R_{yy}	$(p+1)(L-\frac{p}{2})$	$(p+1)(L-\frac{p}{2})$
R_{yx},R_{xy}	0	0
θ_1	0	0
θ_2	0	0
$\Phi = R_{yy} - R_{yx}R_{xy}$	0	0
Φ^{-1}	$2p^2$	$2p^2$
$\begin{bmatrix} -R_{yx}, & \mathbf{I}_{p\times p} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$	$(q^2+q)/2$	$pq + p/2 - p^2/2$
$\Phi^{-1} \begin{bmatrix} \times \\ \times \end{bmatrix}_{p\times 1}$	p^2	p^2
$\begin{bmatrix} -R_{xy} \\ \mathbf{I}_{p\times p} \end{bmatrix} \begin{bmatrix} \times \\ \times \end{bmatrix}_{p\times 1}$	$(q^2+q)/2$	$pq + p/2 - p^2/2$
Total	$(p+1)(L-\frac{p}{2})$ $+3p^2+q^2+q$	$(p+1)(L-\frac{p}{2})$ $+2pq+2p^2+p$

tions required to find the optimal coefficients of the approximate IIR filter.

Table 5.1 summarizes the required computations of the proposed method and their complexity in terms of multiplication operations. From this table, we see that:

(1) R_{yy} only requires $(p+1)(L-\frac{p}{2})$ multiplications, which is obtained by calculating the number of multiplications required for computing the first row of R_{yy}, *i.e.*, $(L+L-1+\cdots+L-p)$. This is because R_{yy} is real symmetric, and we only need to compute the first row of R_{yy};

(2) R_{xy} and R_{yx} require no multiplications because all the elements of the matrix can be obtained from the coefficients of the target FIR, which is assumed to be known;

(3) θ_1 requires no multiplications because it can be obtained by taking the first $q+1$

coefficients of the target FIR;

(4) θ_2 requires no multiplications because it can be obtained by taking the first row of R_{yy} except the first element;

(5) Φ requires no multiplications because R_{yx} is obtained by taking the first q columns of $\mathbf{H}$, and the number of multiplications required in the computation of $R_{yx}R_{xy}$ is already computed when computing R_{yy}.

In this table, we also assume the matrix inversion of Φ requires $2p^2$ multiplications by using the fast algorithm in [24, 25]. Moreover, by noticing the zeros in R_{xy} for different p and q, we calculate the total number of multiplications for $p > q$ and $p \leq q$, respectively. It can be seen that when $p >> q$, the total computational complexity is in $O[2.5p^2]$ operations; when $p << q$, the total computational complexity is linear with q.

Table 5.2 summarizes the required computations of the *Generalized* ARMA-Levinson algorithm [75] and their complexity in terms of multiplication operations. In this table, each matrix involved in the computation has the size of 2×2. In calculating the number of multiplications for each step, we should notice the special structure of each matrix as shown in [75]. Specifically, $\mathbf{R}(i), 1 \leq i \leq m$ has a second row of all zeros, $\mathbf{\Theta}_i^j$ has a second row of all zeros, and $\mathbf{\Delta}_{j+1}^f$ also has a second row of all zeros. In addition, $\mathbf{\Sigma}_j^f$ is a diagonal matrix, and thus the inversion of $\mathbf{\Sigma}_j^f$ only requires 2 multiplications. However, $\mathbf{\Sigma}_j^b$ has no special structures and its inversion operation is assumed to be 8 multiplications. Therefore, the total number of multiplications is $(m+1)(L - \frac{m}{2}) + 16 + (5m+32)(m-1)$, where $m = max(p, q)$.

Fig. 5.2 shows the computational complexity in terms of the number of multiplications versus the number of zeros q for the proposed method and the *Generalized* ARMA-Levinson algorithm. In this figure, the number of poles is fixed, *i.e.*, $p = 50$, and the length of the

Table 5.2: Computational complexity of the *Generalized* ARMA-Levinson algorithm, $m = max(p, q)$

Computation	Multiplication
Initialization:	
$\mathbf{R}(i), 0 \leq i \leq m$	$(m+1)(L - \frac{m}{2})$
$\mathbf{\Theta}_1^1 = \mathbf{K}_1^f = -\mathbf{R}(1)\mathbf{R}^{-1}(0)$	2
$\mathbf{\Psi}_1^1 = \mathbf{K}_1^b = -\mathbf{R}(-1)\mathbf{R}^{-1}(0)$	2
$\mathbf{\Sigma}_1^f = (\mathbf{I} - \mathbf{K}_1^f\mathbf{K}_1^b)\mathbf{R}(0)$	$2+2$
$\mathbf{\Sigma}_1^b = (\mathbf{I} - \mathbf{K}_1^b\mathbf{K}_1^f)\mathbf{R}(0)$	$4+4$
For $1 \leq j \leq m-1$, repeat:	
$\mathbf{\Delta}_{j+1}^f = \mathbf{R}(j+1) + \mathbf{\Theta}_1^j\mathbf{R}(j) + \cdots + \mathbf{\Theta}_j^j\mathbf{R}(1)$	$m(m-1)$
$\mathbf{K}_{j+1}^f = -\mathbf{\Delta}_{j+1}^f(\mathbf{\Sigma}_j^b)^{-1}$	$(4+8)(m-1)$
$\mathbf{K}_{j+1}^b = -\mathbf{\Delta}_{j+1}^{*f}(\mathbf{\Sigma}_j^f)^{-1}$	$4(m-1)$
$\mathbf{\Sigma}_{j+1}^f = (\mathbf{I} - \mathbf{K}_{j+1}^f\mathbf{K}_{j+1}^b)\mathbf{\Sigma}_j^f$	$(2+2)(m-1)$
$\mathbf{\Sigma}_{j+1}^b = (\mathbf{I} - \mathbf{K}_{j+1}^b\mathbf{K}_{j+1}^f)\mathbf{\Sigma}_j^b$	$(4+8)(m-1)$
$\mathbf{\Theta}_i^{j+1} = \mathbf{\Theta}_i^j + \mathbf{K}_{j+1}^f\mathbf{\Psi}_{j-i+1}^j, 1 \leq i \leq j$	$2m(m-1)$
$\mathbf{\Theta}_{j+1}^{j+1} = \mathbf{K}_{j+1}^f$	0
$\mathbf{\Psi}_i^{j+1} = \mathbf{\Psi}_i^j + \mathbf{K}_{j+1}^b\mathbf{\Theta}_{j-i+1}^j, 1 \leq i \leq j$	$2m(m-1)$
$\mathbf{\Psi}_{j+1}^{j+1} = \mathbf{K}_{j+1}^b$	0
Total	$(m+1)(L - \frac{m}{2}) + 16$ $+(5m+32)(m-1)$

target FIR filter is set as $L = 64$. It can be observed that the proposed method has lower computational complexity than the *Generalized* ARMA-Levinson algorithm. Especially for $p \leq q$, we see that the computational complexity of the proposed method is linear with q; however, the computational complexity of the *Generalized* ARMA-Levinson algorithm is quadratic with q. Thus, the proposed method has much lower computational complexity than the *Generalized* ARMA-Levinson algorithm when the number of zeros q is larger.

Fig. 5.3 shows the computational complexity reduction of the proposed method compared with the *Generalized* ARMA-Levinson algorithm for $L = 64$ and $p = 50$. In this figure, the complexity reduction rate is defined as $(C_1 - C_2)/C_1$, where C_1 denotes the computational complexity of the *Generalized* ARMA-Levinson algorithm, and C_2 denotes

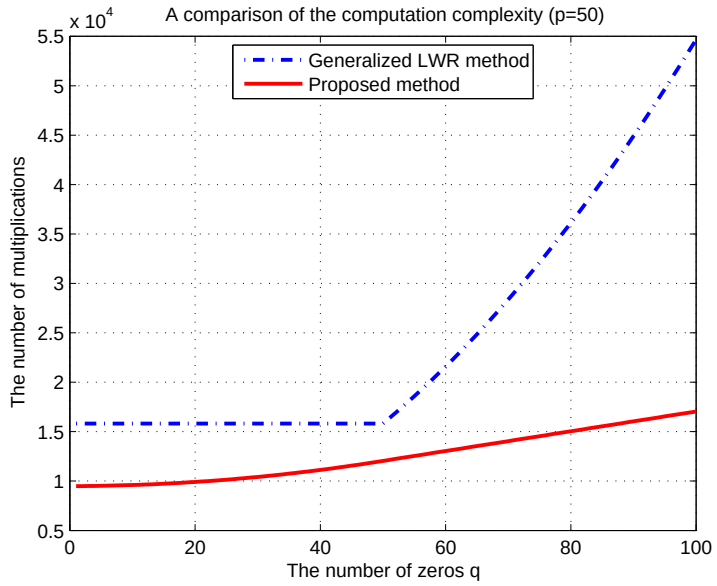


Figure 5.2: Computational complexity of the proposed method and the method in [75]

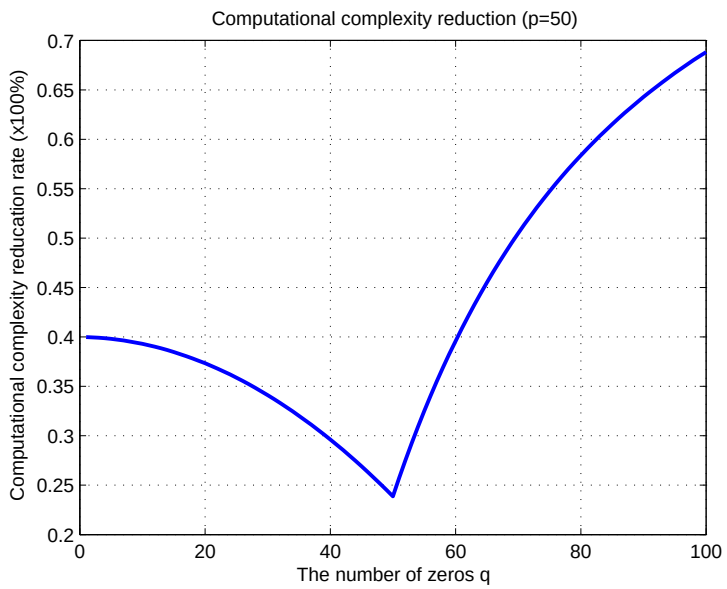


Figure 5.3: Complexity reduction compared with the method in [75]

the the computational complexity of the proposed method. From this figure, we see that the proposed method can reduce more than 23% of the overall computational complexity.

Specifically, when $p > q$, the complexity reducion rate starts at around 40% and decreases as the increase of q. When $p = q$, it reaches the minimum point at around 23%. After that, the complexity reduction rate continues to increase as the increase of q. As we can see from this figure, around 68% complexity reduction can be achieved when $q = 100$ by using the proposed method. In our further study, we find that the complexity reduction rate will approach 100% by keeping increasing q. This is reasonable because when $q >> p$, the approximate filter can be viewed as an FIR filter and thus no computations are needed.

5.3 Numerical Results and Discussions

In this section, we apply the proposed method to find the stable pole-zero approximation of long FIR filters with much fewer parameters. Numerical results are presented to validate the applicability of the proposed method. For the comparison against the *Generalized* ARMA-Levinson algorithm, we use the normalized norm tap error (NNTE) [75] as a measure to evaluate the performance of the proposed method. In addition, the computation time of both methods is simulated using MATLAB to further demonstrate the computational advantage of the proposed method.

We first consider the pole-zero approximation of long FIR feed-forward and feedback filters in the application of decision feedback equalizers (DFE). Similar to [75], we can assume the target FIR feed-forward filter has 98 taps, and the target FIR feedback filter has 64 taps. Since it is easier to model the decaying tail of the FIR filter following the peak, we decompose the FIR feed-forward filter into two parts and then approximate each part separately [75].

Fig. 5.4 shows different pole-zero approximations of the first part of the feed-forward filter by using the proposed method. In this figure, the target FIR filter has 40 taps and

Table 5.3: NNTEs of pole-zero approximations of a 40-tap FIR filter (dB)

p	q	The method in [75]	Proposed method
3	2	-37.3237	-37.3237
5	2	-44.7648	-44.7648
6	2	-48.0655	-48.0655

it is obtained by truncating the IIR filter, $\frac{47.1549-6.4777D-24.4024D^2}{1-1.0134D-.2416D^2+.3091D^3-.0193D^4+.0131D^5+.0028D^6}$. From this figure, we see that the proposed method generates fairly accurate approximations with fewer total number of taps. To further demonstrate this, the NNTE is calculated for each set of parameters as shown in Table 5.3. In this table, the NNTE obtained by using the method in [75] is also listed for comparison. It can be seen that both methods have the same accuracy for different sets of parameters. In addition, the pole-zero approximation will be more accurate if the total number of taps is increased.

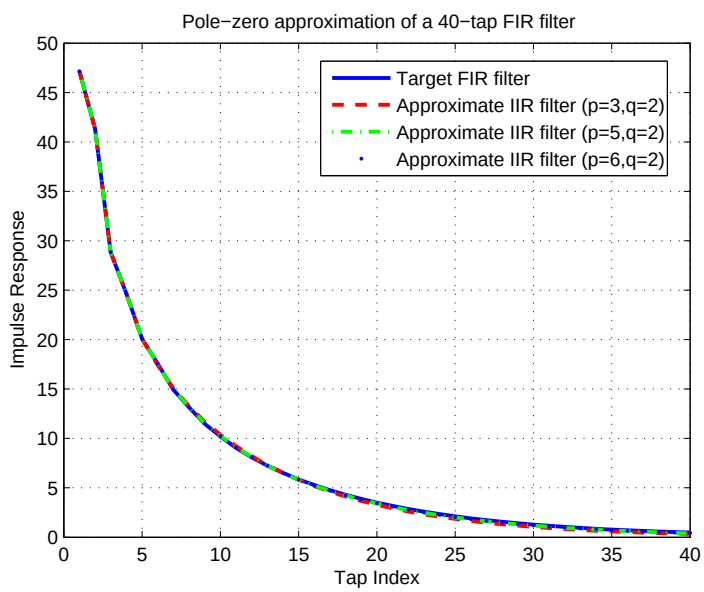


Figure 5.4: Pole-zero approximations of the first part of the feed-forward filter

Fig. 5.5 shows different pole-zero approximations of the second part of the feed-forward filter by using the proposed method. In this figure, the target FIR filter has 58 taps and it is

Table 5.4: NNTEs of pole-zero approximation of a 58-tap FIR filter (dB)

p	q	The method in [75]	Proposed method
2	1	-32.0137	-32.0137
3	3	-32.6067	-32.6067
6	6	-35.2739	-35.2739

obtained by truncating the IIR filter, $\frac{-6.0337+2.5235D+.6947D^2+.7904D^3+.7084D^4+.2405D^5-.1915D^6}{1-.9082D-.3155D^2+.0725D^3-.0224D^4+.0965D^5+.0914D^6}$. As we can see from this figure, the target FIR filter can be approximated well by using different pole-zero models. Table 5.4 summarizes the achievable NNTEs for different pole-zero approximations by using proposed method and the method in [75]. Similar observations can be found from this table.

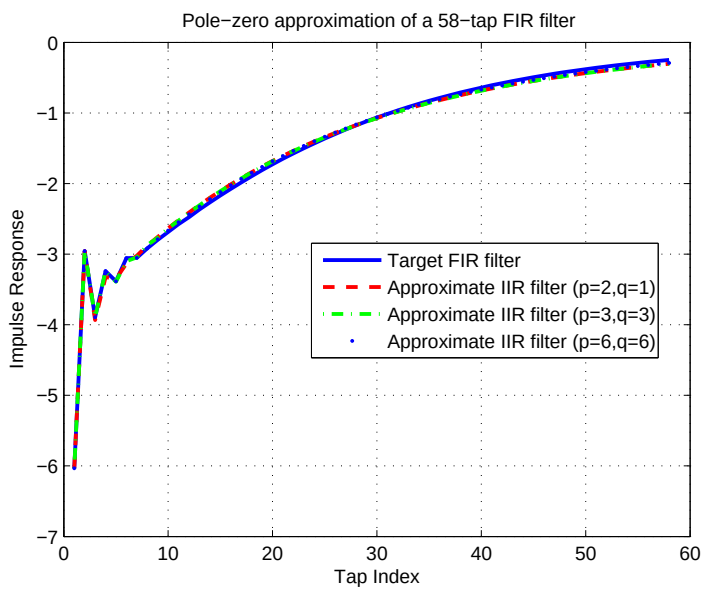


Figure 5.5: Pole-zero approximations of the second part of the feed-forward filter

Fig. 5.6 shows different pole-zero approximations of the feedback filter by using the proposed method. In this figure, the target FIR filter has 64 taps and it is obtained by truncating the IIR filter, $\frac{-1+.7325D+.572D^2-.2791D^3}{1-1.5472D+.3699D^2+.3943D^3-.2299D^4+.0636D^5-.0157D^6+.0048D^7}$. Again we can see from this figure that the generated pole-zero models by using the proposed method can

Table 5.5: NNTEs of pole-zero approximations of a 64-tap FIR filter (dB)

p	q	The method in [75]	Proposed method
3	2	−31.8049	−31.8049
4	4	−49.7145	−49.7145
6	3	−40.2330	−40.2330

approximate the target FIR filter fairly well. Table 5.5 summarizes the achievable NNTEs for different pole-zero approximations by using proposed method and the method in [75]. From this table, we note that increasing p and/or q may not lead to a better approximation, depending on the target FIR filter. This observation is consistent with the results in [75].

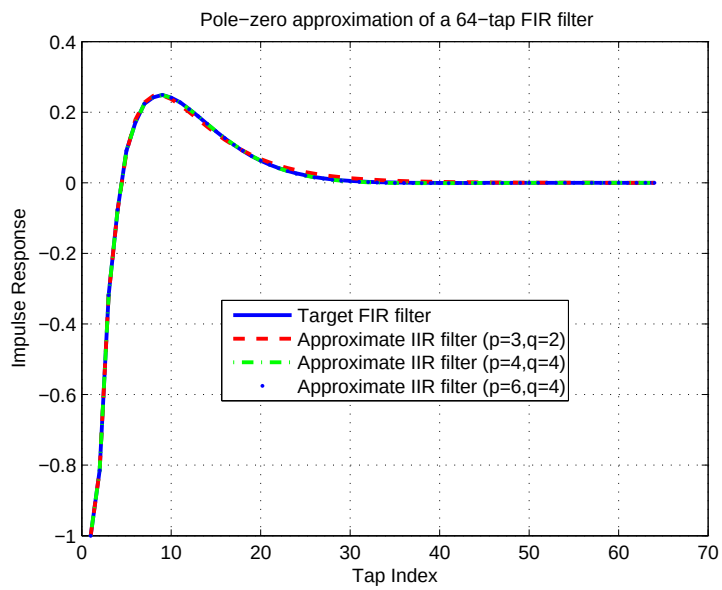


Figure 5.6: Pole-zero approximations of the feed-forward FIR filter

Next we apply the proposed method to generate stable pole-zero models for long FIR echo cancellers used in high speed Ethernet transmission system. In this paper, Cat-6 unshielded twisted-pair (UTP) measured channel model with the cable length 100 meters is used and it can be obtained from the IEEE 802.3an website [9]. In addition, the FIR based echo canceller is assumed to have 500 taps. If the long echo canceller can be approximated by

an IIR filter with fewer total number of taps while still maintaining satisfactory cancellation performance, the hardware cost will be greatly reduced.

Fig. 5.7 and 5.8 show the pole-zero approximations of a 500-tap FIR based echo canceller for two different sets of p and q. It can be seen that both approximations are fairly good. To further evaluate the performance of the approximation, the achievable NNTEs for pole-zero approximations with different sets of parameters are listed in Table 5.6. From this table, we see that by increasing the number of zeros q, the approximation will be more accurate. Similar observation can be found when increasing the number of poles p. We also note that for the same total number of poles and zeros, the approximation with more poles is better than the approximation with more zeros. This suggests that the pole-zero approximation depends on the target FIR filter, and it is important to specify the objective of optimization to achieve the best results. For example, one can pursue best accuracy of approximation under certain hardware complexity constraint.

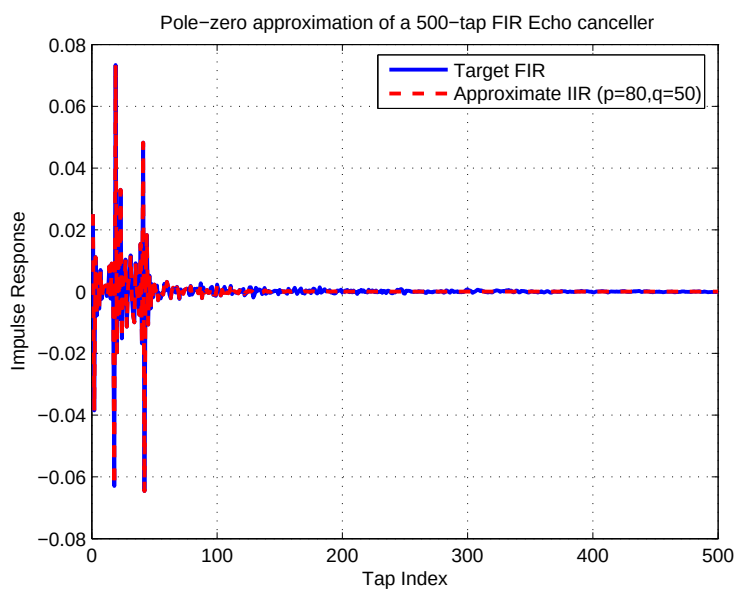


Figure 5.7: Pole-zero approximation of a 500-tap FIR echo canceller, $p = 80$, $q = 50$

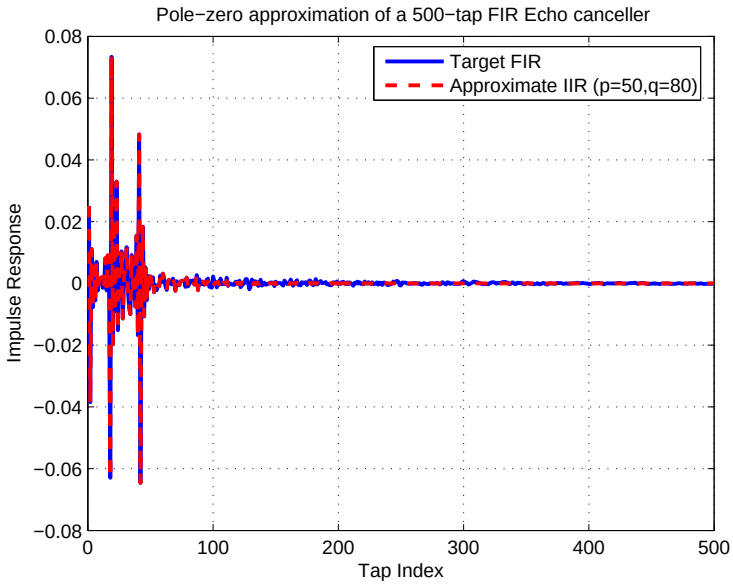


Figure 5.8: Pole-zero approximation of a 500-tap FIR echo canceller, $p = 50$, $q = 80$

Table 5.6: NNTEs of pole-zero approximations of a 500-tap FIR echo canceller (dB)

p	q	The method in [75]	Proposed method
50	70	−25.5608	−25.5608
50	80	−25.9402	−25.9402
50	100	−27.2540	−27.2540
80	50	−27.0733	−27.0733
90	50	−27.7595	−27.7595
100	50	−28.1672	−28.1672

Finally, to further demonstrate the analysis in Section 5.2.3, we compare the computation time required to generate the pole-zero models by applying the proposed method and the method in [75]. The simulations are performed based on a 2.0GHz Pentium PC running Window-XP. The required computation time can be obtained by calculating the cputime using MATLAB. Table 5.7 shows the simulation results for those two methods under different sets of parameters. It should be noted that the computation time for each case listed in this table is obtained by averaging the cputime over 100 independent experiments. From

Table 5.7: Comparison of required computation time simulated in MATLAB

p	q	The method in [75]	Proposed method	Speedup
600	100	$2.8737s$	$0.1932s$	14.9×
800	100	$4.5650s$	$0.3449s$	13.2×
800	400	$4.7689s$	$0.5163s$	9.2×
800	700	$5.0088s$	$0.7485s$	6.7×
100	600	$2.7794s$	$0.0118s$	235.7×
100	800	$4.5578s$	$0.0179s$	254.6×
100	1000	$6.9722s$	$0.0236s$	295.4×

this table, we see that for different sets of parameters, the speedup of the proposed method over the method in [75] is obvious. For example, when $p = 800$ and $q = 100$, the proposed method is computationally faster than the method in [75] by a factor about 13×. However, when increasing q, the speedup factor is reduced till q exceeds p. This observation is in fact consistent with the curve shown in Fig. 5.3. As we can see from the table, when $q > p$, for example, $p = 100$ and $q = 800$, the speed up of the proposed method is about 250×. Especially, when q is much larger than p, the speedup is even larger. This is because when $q > p$, the computational complexity of the proposed method is linear with q, however, the computational complexity of the method in [75] is quadratic with q, which further verifies the analysis in Section 5.2.3. Thus, the proposed method is computationally more efficient and faster than the *Generalized* ARMA-Levinson algorithm.

5.4 Conclusion

In this chapter, we have presented a new approach to compute the optimal settings of the pole-zero models of long FIR filters. The proposed method is derived based on the MMSE formulation of a system identification problem. Instead of solving the MMSE problem recursively, a computationally efficient solution is developed by exploiting the structured

matrices. This new approach is guaranteed to be stable and numerically accurate. It is also general and applicable to cases with unequal numbers of poles and zeros. We have demonstrated that the proposed method can generate stable pole-zero approximation of long FIR filters in the application of DFE and echo cancellers. Numerical results show that using the proposed method results in speedups of much faster than the *Generalized* ARMA-Levinson algorithm, while achieving the same accuracy.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This thesis has presented low complexity and low power design methodologies for efficient implementation of various DSP blocks including channel equalization and noise cancellation, targeting at the high speed Ethernet transmission systems, such as 10GBASE-T.

First, treating FEXT as noise, we have presented a novel feedforward FEXT canceller with TH precoding by overcoming the limitations of prior techniques on FEXT cancellation. In order to speed up the proposed FEXT canceller, a modified design is also developed by eliminating the feedback loops in the FEXT canceller. Compared with the traditional FEXT cancellation approaches, the proposed FEXT canceller can deal with the non-causal part of FEXT, and thus can achieve better cancellation performance. Due to the use of the TH precoders, the error propagation problem is also alleviated in practice. Detailed simulations have been performed under Cat-6 UTP channel environment to verify the effectiveness of proposed designs in the 10GBASE-T application.

By noting that FEXT inherently contains useful information from far-end transmitters,

we have proposed to use MIMO technique to deal with FEXT as signal rather than noise. First, a classical MIMO-DFE based receiver architecture is developed to demonstrate the advantage over the traditional receiver design, where FEXT is cancelled as noise. It is shown that the proposed architecture overcomes the limitation of the traditional schemes and achieves a better SNR performance and lower receiver complexity. It is also noted that, with the same echo and NEXT complexity, the joint equalization and cancellation structure is not necessarily superior to the separate equalization and cancellation structure in the MMSE sense. We have also presented a new method to compute the optimal coefficients of the MIMO equalizers and cancellers. The proposed approach is exact and applicable to the general MIMO DFE computation as well as such cases where echo and NEXT cancellers have a large number of taps with different lengths, which usually make Al-Dhahir's method inefficient. It is shown that, by using the proposed method, we are able to achieve about 63.8% computation cost reduction in terms of multiplication operations compared with the existing methods. This computation speedup also makes the analysis easier when Alien crosstalk such as ANEXT is considered in the channel model. Although the advantage of the MIMO equalization technique has been demonstrated, the classical MIMO-DFE based receiver architecture suffers from error propagation problem in real applications. To eliminate this problem, a new equalization scheme is proposed by combining the MIMO equalization technique and TH precoding technique to deal with both ISI and FEXT. Different with the existing works, the proposed designs inherit the advantage of MIMO equalization and also alleviate the error propagation. In addition, they comply with the 10GBASE-T standard and are also suitable for high speed application because feedback loops in the receiver is completely removed so that pipelining techniques can be easily applied. Simulation results verify that the proposed design can achieve much better

performance in terms of decision-point signal-to-noise ratio (DP-SNR) than conventional techniques. It is also shown that the hardware complexity of the transceiver can be reduced by about 37.2% by utilizing the increased DP-SNR in the proposed designs.

For low complexity and low power design of echo and NEXT cancellers, we have mainly focused on two different aspects to address this problem. First, we have proposed a new tap management algorithm to design echo and NEXT cancellers by exploiting sparse characteristics of the Cat-6 UTP copper channel. Experiment results show that, by using the proposed method in gigabit transceiver design, we are able to achieve 50% and 29∼66.7% computation cost reduction in terms of multiplication during the initial training stage and steady stage respectively. The benefit from the computational complexity reduction can lead to a power efficient design in the application of 10GBASE-T. Second, we have proposed a new word-length reduction scheme for low complexity design of echo and NEXT cancellers. In addition, the statistical properties of the compensation signal $v(n)$ is studied and an improved design is proposed to further bring down the hardware cost. It is shown that this improved design can also be optimized for a low power design by minimizing internal switching activities. Compared with the traditional design, the proposed echo and NEXT cancellers have exact input without suffering from the quantization problem, and thus they are more suitable for VLSI implementation. The proposed method can also be applied to design adaptive echo and NEXT cancellers with some modification. This modified design can achieve similar performance as the traditional design, but with lower hardware complexity. By studying the sum of two adaptive filters, we have proposed a novel weight-updating scheme to further reduce the hardware cost of echo and NEXT cancellers in the application of 10GBASE-T. Unlike our previous design, the proposed scheme can reduce the hardware cost of the weight update part in adaptive echo and NEXT cancellers,

thus the overall hardware cost of these cancellers can be further reduced. It is shown that, by applying the proposed method to the Gigabit transceiver design, we are able to reduce the hardware cost by about 42.02% only with about 1.5dB performance penalty, compared with the traditional design.

Finally, we are interested in approximating a long FIR filter by an IIR filter (or pole-zero filter) with a smaller number of taps including numerator and denominator to reduce the overall hardware cost. we have presented a new approach to compute the optimal settings of the pole-zero models of long FIR filters. The proposed method is guaranteed to be stable and numerically accurate. It is also general and applicable to cases with unequal numbers of poles and zeros. We have demonstrated that the proposed method can generate stable pole-zero approximation of long FIR filters in the application of DFE and echo cancellers. Numerical results show that using the proposed method results in speedups of much faster than the *Generalized* ARMA-Levinson algorithm, while achieving the same accuracy.

6.2 Future Research Direction

This thesis has proposed several approaches to improve the performance and reduce the cost of various DSP blocks in a high speed DSP transceiver. Further research efforts can be directed in the following.

As we mentioned in Chapter 4, the computational complexity of echo and NEXT cancellers can also be reduced at algorithmic level. For example, the linear convolution operation of these cancellers can be efficiently computed using Fast Fourier Transform (FFT) [78]. However, in practical high speed applications, such as 10GBASE-T, the frequency-domain approach seems to be unpromising due to its drawbacks such as long block processing delay, large memory and high precision requirements, especially when the length of the adaptive

filter is very long (*e.g.*, over several thousands). To address those problems, the multi-delay filter (MDF) structure has been proposed [79] which partitions the adaptive filter into M blocks each of length L, and then applies frequency-domain techniques such as FFT to each block separately. As a result, the FFT size is reduced and consequently the delay of MDF algorithm is reduced by a factor of M compared to traditional fast LMS [78]. It is also found that by using a small FFT size and updating the weights more often, the MDF adaptive filter can achieve faster convergence speed and require smaller memory. Thus, it is still promising to use frequency-based approach such as MDF to achieve the overall cost reduction. Future research effort can be focused on efficient VLSI implementations of echo and NEXT cancellers in frequency domain.

In Chapter 4, we have also proposed a low complexity and low power design of echo and NEXT cancellers by exploiting the sparsity of echo and NEXT channel impulse responses. Further optimization can be obtained by incorporating the frequency-domain technique. However, we should note that sparsity in time domain may not be retained in frequency domain. Thus, it would be interesting to investigate sparsity in frequency domain by incorporating MDF, in which the further algorithmic and architecture level optimizations of combining partial update techniques with MDF algorithm would be a challenging task.

Finally, we note that the solutions presented in this thesis is optimized for each particular problem. Further research efforts would be devoted to a complete substantial VLSI implementation by integrating all the proposed approaches for a practical application.

Bibliography

[1] J. Chen and K. K. Parhi, "New Stable Pole-Zero Modeling of Long FIR filters with Low Complexity," Submitted to *IEEE Trans. Signal Processing*, May, 2008.

[2] J. Chen, Y. Gu and K. K. Parhi, "Novel FEXT Cancellation and Equalization for High Speed Ethernet Transmission," Submitted to *IEEE Trans. Circuits and Systems-I: Regular Papers*, April 2008, Revised in June 2008, Accepted in August 2008.

[3] J. Chen and K. K. Parhi, "New Stable IIR Modeling of Long FIR filters with Low Complexity," in *the Forty-Two Asilomar Conference on Signals, Systems and Computers, 2008*, Oct. 26-29, 2008.

[4] J. Chen and K. K. Parhi, "Further Cost Reduction of Adaptive Echo and NEXT Cancellers for High-speed Ethernet Transceivers," in *Proceedings of 2008 IEEE Workshop on Signal Processing Systems (SiPS 2008)*, Oct. 8-10, 2008.

[5] J. Chen, Y. Gu and K. K. Parhi, "Low Complexity Echo and NEXT Cancellers for High Speed Ethernet Transceivers," to appear, *IEEE Trans. Circuits and Systems-I: Regular Papers*, 2008.

[6] IEEE 802.3an 10GBASE-T Study Group. [Online]. Available: http://www.ieee802.org/3/an/public/index.html

[7] IEEE 802 10GBASE-T Tutorial. Nov. 11, 2003. [Online]. Available: http://grouper.ieee.org/groups/802/3/10GBT/public/nov03/10GBASE-T_tutorial.pdf

[8] 800Mbaud PHY for 10GBase-T, *IEEE P802.3an Task Force Meeting*, Ottawa, Sept., 2004. [Online]. Available: http://www.ieee802.org/3/an/public/sep04/tellado_1_0904.pdf

[9] IEEE 802.3an 10GBASE-T Channel Model Material. [Online]. Available: http://www.ieee802.org/3/an/public/channel_models/index.html

[10] IEEE Standard 802.3ab, *Physical Layer Parameters and Specifications for 1000 Mb/s Operation Over 4-Pair of Category-5 Balanced Copper Cabling, Type 1000BASE-T,* 1999.

[11] IEEE 803.2an Standard, *Physical Layer and Management Parameters for 10 Gb/s Operation, Type 10GBASE-T*, 2006.

[12] G. Im, K. M. Kang and C. J. Park, "FEXT Cancellation for Twisted-Pair Transmission," *IEEE J. Select. Areas Commun.*, vol. 20, no. 5, pp. 959-972, June 2002.

[13] G. Ginis and J. Cioffi, "vectored transmission for digital subscriber line systems," *IEEE J. Select. Areas Commun.*, vol. 20, no. 5, pp. 1085-1104, June 2002.

[14] J. Chen, Y. Gu and K. K. Parhi, "MIMO Equalization and Cancellation for 10GBASE-T," in *Proceedings of 2006 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2006)*, vol. 4, pp. 637-640, May 2006.

[15] J. Chen and K. K. Parhi, "Fast Computation of MIMO Equalizers and Cancellers in 10GBASE-T Channels," in *Proceedings of 2007 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2007)*, vol. 3, pp. III-201-III-204, April 2007.

[16] T. Koyama, J. Peng, and P. Cohen, "A multi-dimensional equalizer for Gigabit Ethernet," in *IEEE International Conference on Communications (ICC)*, vol. 2, pp. 509-512, 2001.

[17] K. K. Parhi and Y. Gu, "System and method for MIMO equalization for DSP transceivers," *U.S. Patent Publication: US2007/0014378*, Jan. 18, 2007.

[18] J. Chen and K. K. Parhi, "FEXT Cancellation and Equalization for High-Speed Ethernet Transmission," *IP Disclosure at the University of Minnesota, Docket: Z08209*, June 2, 2008.

[19] J. Chen and K. K. Parhi, "System and Method for Low-Comlexity Adaptive Echo and NEXT Cancellers," *IP Disclosure at the University of Minnesota, Docket: Z09017*, July 23, 2008.

[20] G. Malhotra, J. H. Jeong, and M. Kavehrad, "Joint MIMO Equalization and Decoding for 10GBASE-T Transmission," in *IEEE Global Telecommunications Conference*, vol. 2, pp. 918-922, 2004.

[21] H. Wu and M. Kavehrad, "An MMSE maximal shortening equalizer for 10GBASE-T Ether Networks," in *IEEE Global Telecommunications Conference*, vol. 3, pp. 1342-1346, 2005.

[22] G. Ginis and J. M. Cioffi, "A multi-user precoding scheme achieving crosstalk cancellation withapplication to DSL systems," in *the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers*, vol. 2, pp. 1627 - 1631, 2000.

[23] N. Al-Dhahir and J. M. Cioffi, "MMSE decision-feedback equalizers: Finite-length results," *IEEE Trans. Info. Theory*, vol. 41, pp. 961-975, July 1995.

[24] N. Al-Dhahir and J. M. Cioffi, "Fast Computation of Channel-Estimate Based Equaliers in Packet Data Transmission," *IEEE Trans. Signal Processing*, vol. 43, pp. 2462-2473, Nov. 1995.

[25] N. Al-Dhahir and A. H. Sayed, "The finite-length mulit-input multi-output MMSE-DFE," *IEEE Trans. Signal Processing*, vol. 48, pp. 2921-2936, Oct. 2000.

[26] R. Merched and N. R. Yousef, "Fast Techniques for computing finite-length MMSE Decision Feedback Equalizers," in *Proceedings of 2004 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2004)*, vol. 4, pp. 1005-1008, May 2004.

[27] R. Merched and N. R. Yousef, "Fast computation of finite-length MIMO Decision Feedback Equalizers," in *Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, 2004*, vol. 2, pp. 1781-1784, Nov. 2004.

[28] R. Merched and N. R. Yousef, "Fast techniques for computing finite-length MIMO MMSE Decision Feedback Equalizers," in *IEEE Trans. on Signal Processing*, vol. 54, No. 2, pp. 701-711, Feb. 2006.

[29] Y. Gu and K. K. Parhi, "High-Speed Architecture Design of Tomlinson-Harashima Precoders," *IEEE Trans. on Circuits and Systems I*, vol. 54, pp. 1929 - 1937, Sept. 2007.

[30] G. A. Zimmerman, "Challenges for 10Gb/s implementation on UTP media," [Online]. Available:
http://www.solarflare.com/ technology/documents/Challengesfor10Gbpublishedpaper.pdf

[31] G. Long, F. Ling, and J. G. Proakis, "The LMS algorithm with delayed coefficient adaptation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 1397-1405, Sept. 1989.

[32] G. D. Forney, Jr. and M. V. Eyuboglu, "Combined equalization and coding using precoding," *IEEE Communications Magazine*, vol. 29, no. 12, pp. 25-34, Dec. 1991.

[33] S. Kasturia and J. Tellado, "Lower Complexity Architectures for Implementing 10GBT XTalk Cancellers and Equalizers FIRs," *10GBase-T Study Group Meeting*, Sept. 2003. [Online]. Available:
http://www.ieee802.org/3/10GBT/public/sep03/kasturia_1_0903.pdf

[34] M. Hsu, Y. Chen, K. Jheng, and A. Wu, "A Shortened Impulse Response Filter (SIRF) Scheme for Cost-Effective Echo Canceller Design of 10GBase-T Ethernet System," in *Proceedings of 2006 IEEE Workshop on Signal Processing Systems (SiPS 2006)*, pp. 309-312, Oct. 2006.

[35] Training block diagram, May 10, 2005. [Online]. Available:
http://www.ieee802.org/3/an/public/may05/powell_1_0505.pdf

[36] B. McClellan, "10GBASE-T Programmable THP Proposal," in *IEEE 802.3an Task Force Meetings*, May 2005. [Online]. Available:
http://www.ieee802.org/3/an/public/may05/mcclellan_2_0505.pdf

[37] M. Hatamian, et. al., "Design considerations for Gigabit Ethernet 1000Base-T twisted pair transceivers," in *Proceedings IEEE Custom Integrated Circuits Conference (CICC)*, pp. 335-342, 1998.

[38] A. P. Liavas, "Tomlinson-Harashima Precoding with partial channel knowledge," *IEEE Trans. Commun.*, vol. 53, pp. 5-9, Jan. 2005.

[39] M. Huang, X. Zhang, S. Zhou, and J. Wang, "Tomlinson-Harashima precoding in multiuser MIMO systems with imperfect channel state information," in *Proceedings of IEEE GLOBECOM 2007*, Washington, pp. 2806-2810, Nov. 2007.

[40] M. Huang, S. Zhou, and J. Wang, "Analysis of Tomlinson-Harashima Precoding in multiuser MIMO systems with imperfect channel state information," *IEEE Trans. Vehicular Technology*, Accepted for future publication.

[41] Y. Chien and H. Tsao, "A novel transmitter-side-based far-end crosstalk cancellation for 10GBASE-T," in *Proceedings of 2006 IEEE International Conference on Communication Technology (ICCT 2006)*, pp. 1-4, 2006.

[42] Y. Chien, Y. Tu, H. Tsao, and W. Mao, "Equalization and interference cancellation with MIMO THP for 10GBASE-T," in *Proceedings of 2007 IEEE Workshop on Signal Processing Systems (SiPS 2007)*, pp. 95-100, 2007.

[43] R. K. Martin, W. A. Sethares, R. C. Williamson, and C. R. J. Jr., "Exploiting Sparsity in Adaptive Filters," *IEEE Transactions on Signal Processing*, vol. 50, pp. 1883-1894, 2002.

[44] D. L. Duttweiler, "Proportionate Normalized Least Mean Squares Adaptation in Echo Cancellers," *IEEE Transactions on Speech and Audio Processing*, vol. 8, pp. 508-518, 2000.

[45] J. Benesty and S. L. Gay, "An Improved PNLMS Algorithm," in *Proceedings of ICASSP 2002*, vol. 2, pp. 1881-1884, 2002.

[46] J. Cui, P. A. Naylor, and D. T. Brown, "An improved IPNLMS algortihm for echo cancellation in packet-switched networks," in *Proceedings of ICASSP 2004*, vol. 4, pp. iv-141-iv-144, 2004.

[47] S. Kawamura and M. Hatori, "A Tap Selection Algorithm for Adaptive Filters," in *Proceedings of ICASSP 1986*, vol. 11, pp. 2979-2982, 1986.

[48] A. Sugiyama, H. Sato, A. Hirano, and S. Ikeda, "A Fast Convergence Algorithm for adaptive Fir Filters Under Computational Constraint for Adaptive Tap-Position Control," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 43, pp. 629-636, 1996.

[49] S. C. Douglas, "Adaptive filters employing partial updates," *IEEE Trans. Circuits Syst. II*, vol. 44, no. 3, pp. 209-216, Mar. 1997.

[50] H. Deng and M. Doroslovacki, "New sparse adaptive algorithms using partial update," in *Proceedings of ICASSP 2004*, vol. 2, pp. 845-848, 2004.

[51] T. Aboulnasr and K. Mayyas, "Complexity reduction of the NLMS algorithm via selective coefficient update," *IEEE Trans. Signal Processing*, vol. 47, no. 5, pp. 1421-1424, 1999.

[52] K. C. Ho and S. D. Blunt, "Adaptive Sparse System Identification Using Wavelets," *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 49, pp. 656-667, 2002.

[53] Y. Zhou, S. C. Chan and K. L. Ho, "A Wavelet Based Partial Update Fast LMS/Newton Algorithm," in *Proceedings of Int. Symposium on Intelligent Signal Processing and Communication Systems*, pp. 817-820, Dec. 2005.

[54] F. O'Regan and C. Heneghan, "A low power algorithm for sparse system identification using cross correlation," in *Proceedings of 2003 IEEE Workshop on Signal Processing System (SiPS 2003)*, pp. 18-23, 2003.

[55] Y. Lu, W. Tian and L. Thompson, "Enhancing Echo Cancellation via Estimation of Delay," *IEEE Transactions on Signal Processing*, vol. 53, pp. 4159-4168, Nov. 2005.

[56] J. Chen and K. K. Parhi, "Adaptive Tap Management in Multi-Gigabit Echo and NEXT Cancellers," in *Proceedings of 2006 IEEE Workshop on Signal Processing System (SiPS 2006)*, October 2-4, 2006.

[57] H. Fan and W. K. Jenkins, "An investigation of an adaptive IIR echo canceller: advantage and problems," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 36, no. 12, pp. 1819-1834, 1988.

[58] G. Zimmerman, "Downside of TH Precoding," *10GBase-T Study Group Meeting*, May, 2004. [Online]. Available:
http://www.ieee802.org/3/an/public/may04/zimmerman_1_0504.pdf

[59] M. Tomlinson, "New automatic equalizer employing modulo arithmetic," *Electron. Lett.*, vol. 7, pp. 138-139, Mar. 1971.

[60] M. Miyakawa and H. Harashima, "A method of code conversion for a digital communication channel with intersymbol interference," *Trans. Inst. Electron. Commun. Eng. Jap.*, vol. 52-A, pp.272-273, June 1969.

[61] H. Harashima and H. Miyakawa, "Matched-transmission technique for channels with intersymbol interference," *IEEE Trans. Commun.*, vol. 20, pp. 774-780, Aug. 1972.

[62] K. K. Parhi, "Pipelining of Parallel Multiplexer Loops and Decision Feedback Equalizers," in *Proceedings of ICASSP 2004*, vol. 5, pp. 21-24, May 2004.

[63] K. K. Parhi, "Pipelining in algorithms with quantizer loops," *IEEE Trans. on Circuits and Systems*, vol. 37 no. 7, pp. 745-754, Jul. 1991.

[64] A. Vareljian, "Fixed Set FIR Transfer Functions for 10GBASE-T THP," *10GBase-T Study Group Meeting*, Jan., 2005. [Online]. Available: http://www.ieee802.org/3/an/public/jan05/vareljian_1_0105.pdf

[65] H. Choi and W. P. Burleson, "Search-Based Wordlength Optimization For VLSI/DSP Synthesis," in *Proceedings of IEEE Workshop on VLSI Signal Processing,VII*, pp. 198-207, Oct. 1994.

[66] A. P. Chandrakasan and R. W. Brodersen, "Minimizing power consumption in digital CMOS circuits," in *Proceedings IEEE*, vol. 83, no. 4, pp. 498-523, Apr. 1995.

[67] Oklahoma State University VLSI Research, [Online]. Available: http://avatar.ecen.okstate.edu/projects/scells/download

[68] J. Grad and J. Stine, "standard cell library for student projects," in *Proceedings of Int. Conf. Microelectronic Systems Education, IEEE Computer Society*, pp. 98-99, 2003.

[69] S. Shaffer and C. S. Williams, "The Filtered Error LMS Algorithm," in *Proceedings of ICASSP 1983*, vol. 8, pp. 41-44, Apr. 1983.

[70] C. Lin, A. Wu, and F. Li, "High-Performance VLSI Architecture of Decision Feedback Equalizer for Gigabit Systems," *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 53, no. 9, pp. 911-915, Sept. 2006.

[71] E. F. Haratsch and K. Azadet, "A 1-Gb/s Joint Equalizer and Trellis Decoder for 1000BASE-T Gigabit Ethernet," *IEEE Journal of Solid-State Circuits*, vol. 36, no. 3, pp. 374-384, March 2001.

[72] K. Azadet and E. Haratsch, "DSP implementation issues in 1000BASE-T Gigabit Ethernet," in *Proceedings of Int. Symp. on VLSI Technology, Systems, and Applications*, pp. 109-112, 2001.

[73] E. F. Haratsch and K. Azadet, "A pipelined 14-tap parallel decision-feedback decoder for 1000BASE-T Gigabit Ethernet," in *Proceedings of Technical Papers, 2001 International Symposium on VLSI Technology, Systems, and Applications*, pp. 117-120, April, 2001.

[74] P. Crespo and M. Honig, "Pole-Zero Decision Feedback Equalization with a Rapidly Converging Adaptive IIR Algorithm," *IEEE J. on Select. Areas Commun.*, vol. 9, pp. 817-829, August 1991.

[75] N. Al-Dhahir, A. H. Sayed, and J. M. Cioff, "Stable Pole-Zero Modeling of Long FIR Filters with Application to the MMSE-DFE," *IEEE Trans. Commun.*, vol. 45, no. 5, pp. 508-513, May 1997.

[76] F. Lin and J. G. Proakis, "A generalized multichannel least square lattice algorithm based on sequential processing stages," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 32, no. 2, pp. 381-389, Apr. 1984.

[77] P. Stoica, and A. Nehorai, "On stability and root location of linear prediction models," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-35, no. 4, Apr. 1987.

[78] E. R. Ferrara, "Fast implementations of LMS adaptive filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 474-475, 1980.

[79] J. S. Soo and K. K. Pang, "Multidelay block frequency domain adaptive filter," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, no. 2, pp. 373-376, 1990.

[80] A. W. H. Khong, J. Benesty, and P. A. Naylor, "A low delay and fast converging improved proportionate algorithm for sparse system identification," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2007, Article ID 84376, 8 pages, 2007.

[81] X. Lin, A. W. H. Khong, M. Doroslovacki, and P. A. Naylor, "Frequency-Domain Adaptive Algorithm for Network Echo Cancellation in VoIP," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2008, Article ID 156960, 9 pages, 2008.

[82] A. W. H. Khong, X. Lin, M. Doroslovacki, and P. A. Naylor, "Frequency domain selective tap adaptive algorithms for sparse system identification," in *Proceedings of IEEE Int. Conf. Acoust. Speech Signal Process 2008 (ICASSP 2008)*, pp. 237-240, 2008.

[83] A. V. Oppenheim and R. W. Schafer, Discrete-Time Signal Processing, 2nd Edition, Prentice Hall, 1998.

[84] S. Haykin, Adaptive Filter Theory, 4th Ed, Prentice Hall, 2001.

[85] K. K. Parhi, *VLSI Digital Signal Processing System Design and Implementation*, John Wiley & Son, Inc., New York, 1999.