ALGORITHMS FOR BLIND EQUALIZATION BASED ON RELATIVE GRADIENT AND TOEPLITZ CONSTRAINTS

Zhengwei Wu

A DISSERTATION

in

Electrical and Systems Engineering

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

2016

Supervisor of Dissertation

_____

Saleem A. Kassam, Solomon and Sylvia Charp Professor of Electrical and Systems Engineering

Graduate Group Chairperson

_____

Alejandro Ribeiro, Associate Professor of Electrical and Systems Engineering

Dissertation Committee

Santosh S. Venkatesh, Professor of Electrical and Systems Engineering

Saswati Sarkar, Professor of Electrical and Systems Engineering

Visa Koivunen, Academy Professor of Signal Processing and Acoustics, Aalto University

ALGORITHMS FOR BLIND EQUALIZATION BASED ON RELATIVE GRADIENT AND TOEPLITZ CONSTRAINTS

Dedicated to my family.

ACKNOWLEDGEMENT

My deepest gratitude goes to my advisor Prof. Saleem A. Kassam. This dissertation would not have been possible without his continual guidance, patience and support. During my studies at Penn, he spent a lot of time guiding me to think independently, to question critically, and to communicate effectively. I feel incredibly honored to be one of the students of Prof. Kassam, and extremely lucky to have an advisor who cared so much about my work and tried so hard to make me a better person. Through the invaluable academic and life lessons, he has set me an excellent example as a researcher, a mentor and a role model. I shall remain indebted to his contribution to my rewarding graduate school experiences, and carry the virtues and principles I learned throughout my life.

I would like to thank the members of my dissertation committee: Prof. Visa Koivunen, Prof. Santosh S. Venkatesh, and Prof. Saswati Sarkar. I appreciate their time and effort to review my work and provide valuable comments.

I am grateful for the opportunity to study in the Department of Electrical and Systems Engineering at Penn. I benefitted greatly from the courses offered and the seminars organized by the department. My acknowledgement also goes to the technical and departmental staff for all their help and support.

Completing this dissertation would have been more difficult without the companionship of my friends at Penn. I am lucky to have met so many excellent and supportive people, inside and outside the department, who have provided me substantial amount of help. The time spent with you will always be a part of my sweet memory.

Most importantly, I would like to thank my parents, Jing Xu and Tiemin Wu, for supporting me, sharing happiness and offering encouragement. They have been a constant source of unconditional love, strength and patience all these years, without which none of this would have been possible. I am grateful for their support in my decision to pursue education abroad, being far away from them. Last but not the least, my appreciation goes to my beloved husband, Weilu Gao, for his persistent love and compassion. He has always been there for me, through thick and thin, up and down. I would never be able to have reached this point without his support and companionship.

ABSTRACT

ALGORITHMS FOR BLIND EQUALIZATION BASED ON RELATIVE GRADIENT AND TOEPLITZ CONSTRAINTS

Zhengwei Wu

Saleem A. Kassam

Blind Equalization (BE) refers to the problem of recovering the source symbol sequence from a signal received through a channel in the presence of additive noise and channel distortion, when the channel response is unknown and a training sequence is not accessible. To achieve BE, statistical or constellation properties of the source symbols are exploited. In BE algorithms, two main concerns are convergence speed and computational complexity.

In this dissertation, we explore the application of *relative gradient* for equalizer adaptation with a *structure constraint* on the equalizer matrix, for fast convergence without excessive computational complexity. We model blind equalization with symbol-rate sampling as a blind source separation (BSS) problem and study two single-carrier transmission schemes, specifically block transmission with guard intervals and continuous transmission. Under either scheme, blind equalization can be achieved using independent component analysis (ICA) algorithms with a Toeplitz or circulant constraint on the structure of the separating matrix. We also develop *relative gradient* versions of the widely used Bussgang-type algorithms. Processing the equalizer outputs in sliding blocks, we are able to use the relative gradient for adaptation of the Toeplitz constrained equalizer matrix.

The use of relative gradient makes the Bussgang condition appear explicitly in the matrix adaptation and speeds up convergence.

For the ICA-based and Bussgang-type algorithms with relative gradient and matrix structure constraints, we simplify the matrix adaptations to obtain equivalent equalizer vector adaptations for reduced computational cost. Efficient implementations with fast Fourier transform, and approximation schemes for the cross-correlation terms used in the adaptation, are shown to further reduce computational cost.

We also consider the use of a relative gradient algorithm for channel shortening in orthogonal frequency division multiplexing (OFDM) systems. The redundancy of the cyclic prefix symbols is used to shorten a channel with a long impulse response. We show interesting preliminary results for a shortening algorithm based on relative gradient.

Table of Contents

Table of Figures

Chapter 1

Introduction

In digital communication systems, information is represented as symbols that belong to a finite, discrete constellation. The digital signals are transmitted through an analog channel between the transmitter and the receiver. Real channels are never ideal, and therefore the received signal may undergo significant distortion. A widely encountered form of distortion is caused by non-ideal linear channels, where the frequency response is not flat in magnitude or not linear in phase across the transmission bandwidth. This results in inter-symbol interference (ISI). To compensate for this distortion, a linear equalizer can be used at the receiver. An equalizer can provide significant reduction in ISI.

Traditional equalization is based on transmitted training sequences, and uses the minimum mean square error criterion for equalizer adaptation [1]. However, it is not always feasible to send training sequences, which also take up bandwidth for transmission and reduce the effective data rate. Blind equalization (BE) has the advantage of eliminating the use of pilot training sequences. Specifically, BE refers to equalization achieved without

knowledge of the channel characteristics when no training sequence is used, with the equalizer converging to a good solution based only on channel outputs during actual data transmission.

BE has been obtained by exploiting known statistical or constellation properties of the source symbols [2]–[7]. The BE technique is broadly useful in many applications beyond classical point–to-point communication channel equalization. The principles of blind channel equalization can be applied in seismic signal processing [8], in reduction of microphone-induced ISI in speech recognition [9], [10], and in massive MIMO systems with time-division duplex (TDD) where the uplink and downlink channels are reciprocal, allowing BE to be employed by a station based on uplink transmission for better information about the state of the channel [11]–[14].

There has been extensive work done on blind equalization. Existing algorithms belong mainly to two types of schemes: one based on known statistical properties of the source sequence [6], [7], [15], and the other based on the known structure of the signaling constellation [2]–[4]. The limitations of existing BE algorithms are computational complexity and/or slow convergence. Especially in practical applications where the channel is time-varying, fast convergence of channel equalization is necessary [1].

In this dissertation, we modify existing algorithms and develop new ones for BE. We utilize ideas of *relative gradient* for equalization adaptation, and *constraints on the matrix structure* of the equalizer representation, for faster convergence without excessive computational cost. First we focus on independent component analysis (ICA)-based algorithms, where the relative gradient has been used to achieve BSS. Instead of employing a fractional sampling scheme that allows, for blind equalization, the model of a standard

blind source separation (BSS) problem to be used and the application of an ICA-based algorithm, we use *symbol rate* processing. Two transmission schemes are analyzed, where the symbols are transmitted either in blocks with a guard interval, or continuously without guard intervals. With either of the schemes, the BE problem can be formulated in matrix form that has a form similar to that of the BSS problem, except with an additional constraint on the matrix structure. With a structure-constrained ICA algorithm, independent source symbols can be recovered with faster convergence. For source symbols with independent in-phase and quadrature parts, the I/Q independence constraint can be used further for phase recovery.

We then consider the widely used Bussgang-type algorithms for BE. In the standard Bussgang-type algorithms, one equalizer output is processed each time with equalizer adaptations based on standard stochastic gradient descent. In our work, a block processing scheme for the equalizer outputs is proposed, which allows the use of the relative gradient. With the relative gradient, the Bussgang condition appears in the adaptation explicitly and helps speed up convergence.

The block processing approach for both the ICA-based algorithms and the modified Bussgang-type algorithm shows the interesting connection between these two types of algorithms. Although the starting points for the two types of algorithms are different, they end up having related structures. With a matrix structure constraint, the matrix adaptations for both types of algorithms can be expressed as simpler equalizer vector adaptations. Approximation schemes simplifying the updates and the use of the fast Fourier transform allow the computational cost to be further reduced.

We also propose in a final chapter the use of relative gradient and structure constraint for channel shortening in orthogonal frequency division multiplexing (OFDM) system. Channel shortening allows a long channel impulse response to be partially corrected to a shorter one that can then be equalized based on the OFDM cyclic prefix. The redundancy due to the cyclic prefix is used in the cost function. We show through simulation the performance comparison between the proposed and existing algorithms. We also discuss briefly the potential aspects that may be considered for a more comprehensive evaluation of the channel shortening algorithms.

1.1 Organization of the Dissertation

There are five main parts in the dissertation: Chapter 2 through Chapter 6.

In Chapter 2, we summarize the fundamental concepts of BSS and BE, including the notation and the models. The connection between BSS and BE is explained. We introduce the core idea of ICA, which is a widely used approach for BSS. The "contrast" or criterion functions and algorithms for ICA methods are explained. Also included in this chapter is a brief review of BE algorithms, which lays the foundation of further development.

In Chapter 3, we describe two block transmission schemes using guard intervals, for symbol-rate processing in standard single-carrier systems. With the padded guard intervals between transmitted blocks, BE can be modeled as a standard BSS problem, which enables the use of an ICA-based algorithm for BSS. With the guard interval being zeros or a cyclic prefix, the "separating matrix" has the constraint of being Toeplitz or circulant. We present a method to enforce the structure constraint in the separating matrix adaptation. This allows

faster convergence compared to standard ICA-based algorithms. I/Q independence constraint can be combined with the structure constraint for cases where the source symbols have independent in-phase and quadrature parts. With either the Toeplitz or circulant constraint, there are repeated elements in the separating matrix. We give an equivalent computationally efficient adaptation for the vector of elements contained in the separating matrix.

In Chapter 4, we develop continuous transmission symbol-rate BE schemes related to the block transmission schemes using ICA-based algorithms. Unlike previous work where fractional sampling was needed for the use of standard ICA-based algorithms [6], [7], we show that BE can be achieved with a constrained ICA algorithm using symbol-rate sampling. We show that the matrix we aim to find for source symbol recovery is a Toeplitz matrix containing the impulse response of the equalizer. With the Toeplitz constraint during matrix adaptation, faster convergence can be achieved. The constrained adaptation leads to an equivalent form of equalizer vector adaptation. Modifications to further reduce the computational complexity using approximations of vector update equations and the fast Fourier transform (FFT) is explained in detail in this chapter. Simulation results are shown for different ICA-based algorithms, and also compared with those of other standard BE algorithms.

In Chapter 5, we introduce our approach to modifying Bussgang-type algorithms with relative gradient instead of the standard gradient, using output block processing. By looking at a block of equalizer outputs, a Toeplitz matrix containing the equalizer vector can be updated each time based on the cost function from a Bussgang-type algorithm, which exploits some known constellation property of the source symbols. The application

of relative gradient results in an explicit use of the Bussgang condition for faster convergence. The structure of matrix adaptation with Toeplitz constraint based on Bussgang-type algorithms is similar to that of the constrained ICA-based algorithms. We show that these two types of algorithms have a close connection to each other.

In Chapter 6, we investigate briefly the application of relative gradient in channel shortening algorithms for OFDM systems. In OFDM a cyclic prefix is usually used, which results in redundancy of the cyclic symbols and the corresponding data symbols in the OFDM block. When the channel impulse response is shortened to a length smaller than that of the cyclic prefix, the redundancy between the OFDM symbols and the corresponding cyclic prefix symbols should be maintained. When multiple cyclic prefix symbols are taken into consideration, the problem of exploiting redundancy can be formulated in a matrix expression. This allows the effective use of relative gradient during adaptation, with the expectation that convergence can be faster. We give preliminary simulation results and discuss possible directions of future work.

1.2 Contributions and Publications

The main contributions arising from this dissertation are listed below; there are four conference papers, and two journal papers in preparation for submission.

Zhengwei Wu, Saleem A. Kassam and Kaipeng Li, "Blind Equalization Based On Blind Separation with Toeplitz Constraint," *Proc. of 48th Asilomar Conference on Signals, Systems and Computers*, Asilomar, CA, 2014.

Zhengwei Wu and Saleem A. Kassam, "Symbol-Rate Blind Equalization Based on Constrained Blind Separation," *Proc. of 49th Annual Conference on Information Sciences and Systems (CISS)*, John Hopkins, MD, 2015.

Zhengwei Wu, Saleem A. Kassam, and Visa Koivunen, "Relative-Gradient Bussgang-Type Blind Equalization Algorithms," *Proc. of 41st IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, Shanghai, China, 2016.

Zhengwei Wu and Saleem A. Kassam, "Computationally Efficient Toeplitz-Constrained Blind Equalization Based on Independence," *Proc. of 50th Annual Conference on Information Sciences and Systems (CISS)*, Princeton, NJ, 2016.

Zhengwei Wu and Saleem A. Kassam, "ICA-Based Blind Equalization Algorithms with Toeplitz Constraint," (*In progress*).

Zhengwei Wu and Saleem A. Kassam, "Relative-Gradient Bussgang-Type Blind Equalization Algorithms," (*In progress*).

References

[1] Z. Ding and Y. Li, *Blind Equalization and Identification*. CRC press, 2001.

[2] Y. Sato, “A Method of Self-Recovering Equalization for Multilevel Amplitude-Modulation Systems,” *IEEE Trans. Commun.*, vol. 23, no. 6, pp. 679–682, 1975.

[3] D. N. Godard, “Self-Recovering Equalization and Carrier Tracking in Two-Dimensional Data Communication Systems,” *IEEE Trans. Commun.*, vol. 28, no. 11, pp. 1867–1875, 1980.

[4] J. Treichler and B. G. Agee, “A new approach to multipath correction of constant modulus signals,” *IEEE Trans. Acoust.*, vol. 31, no. 2, pp. 459–472, 1983.

[5] O. Shalvi and E. Weinstein, “New criteria for blind deconvolution of nonminimum phase systems (channels),” *IEEE Trans. Inf. Theory*, vol. 36, no. 2, pp. 312–321, 1990.

[6] H. H. Yang, “On-line blind equalization via on-line blind separation,” *Signal Processing*, vol. 68, no. 3, pp. 271–281, 1998.

[7] Y. Zhang and S. A. Kassam, “Blind separation and equalization using fractional sampling of digital communications signals,” *Signal Processing*, vol. 81, no. 12, pp. 2591–2608, 2001.

[8] Y. Li, C. Guo, and T. Fei, “Application of Improved Classical Blind Equalization Algorithm in Seismic Signal Processing,” in *Advances in Electric and Electronics*, Springer, 2012, pp. 591–598.

[9] L. Mauuary, “Blind equalization in the cepstral domain for robust telephone based speech recognition,” in *Signal Processing Conference (EUSIPCO 1998), 9th European*, 1998, pp. 1–4.

[10] D. Wang, H. Leung, K.-C. Kwak, and H. Yoon, “Enhanced Speech Recognition with Blind Equalization For Robot‘ WEVER-R2,’” in *Robot and Human interactive Communication, 2007. RO-MAN 2007. The 16th IEEE International Symposium on*, 2007, pp. 684–688.

[11] D. Neumann, M. Joham, and W. Utschick, “Channel Estimation in Massive MIMO Systems,” *arXiv Prepr. arXiv1503.08691*, 2015.

[12] J. Jose, A. Ashikhmin, P. Whiting, and S. Vishwanath, “Channel estimation and linear precoding in multiuser multiple-antenna TDD systems,” *IEEE Trans. Veh. Technol.*, vol. 60, no. 5, pp. 2102–2116, 2011.

[13] A. Farhang, A. Aminjavaheri, N. Marchetti, L. E. Doyle, and B. Farhang-Boroujeny, “Pilot Decontamination in CMT-based Massive MIMO Networks,” in *Wireless Communications Systems (ISWCS), 2014 11th International Symposium on*, 2014, pp. 589–593.

[14] H. Q. Ngo and E. G. Larsson, “Blind Estimation of Effective Downlink Channel Gains in Massive MIMO,” 2015.

[15] S. Zheng, Fu-Chun McLaughlin and B. Mulgrew, “Blind equalization of nonminimum phase channels: higher order cumulant based algorithm,” *Signal Process. IEEE Trans.*, vol. 41, no. 2, pp. 681–691, 1993.

Chapter 2

Review of Blind Source Separation and Blind Equalization

2.1 Introduction

Blind adaptive equalization has been of long-standing interest, and Bussgang-type algorithms for blind equalization (BE) based on gradient descent schemes are well-known. *Blind source separation* (BSS) and blind equalization problems have similar goals, recovery of the original signals from their observed mixtures based on limited knowledge of the sources. As a result, BE algorithms based on BSS have also been of interest.

In this dissertation, we will develop new BE algorithms based on both BSS techniques and Bussgang-type methods, incorporating *constrained matrix adaptation* and the ideas of

natural and *relative gradient*. In this background chapter, we summarize the basic concepts and approaches of BE and BSS.

In Section 2.2, the model of BSS and the well-known *Independent Component Analysis* (ICA) based algorithms are introduced first. Different criteria that may be used for BSS are given. We summarize the two useful steps in ICA-based approaches: whitening and orthogonalization. A brief discussion of gradient descent based on natural and relative gradient concepts is also given. Since the focus of the dissertation is to obtain better BE schemes, we also introduce the general model of the BE problem in Section 2.3. The relation of the BE model to the BSS problem is explained. Examples of BE algorithms that are widely used are briefly reviewed. Further details related to specific algorithms will be included as we use them in later chapters.

2.2 Review of BSS and ICA

2.2.1 Basic Model

Blind source separation (BSS) is the problem of recovering independent sources from observed mixtures when no information about the mixing process and no training sequence is available. Generally, in BSS problem, there are n independent source signals at time k, i.e. $\mathbf{s}_k = [s_k(1), s_k(2), ..., s_k(n)]^T$. These sources get mixed, and result in m linear combinations with unknown coefficients. Expressing this process in matrix form, we have

$$\mathbf{x}_k = \mathbf{A}\mathbf{s}_k, \tag{2.1}$$

where $\mathbf{x}_k = [x_k(1), x_k(2), ..., x_k(m)]^T$ is the $m \times 1$ vector of observations, and $\mathbf{A}$ is the $m \times n$ matrix of mixture coefficients. It is generally assumed that $m \geq n$, i.e. there are at least as many observations as sources.

The goal of BSS is to find an $n \times m$ separating matrix $\mathbf{B}$ such that

$$\mathbf{y}_k = \mathbf{B}\mathbf{x}_k \tag{2.2}$$

contains estimates of the source components $s_k(i)$, $i = 1, 2, ..., n$. Ideally, the separating matrix should satisfy

$$\mathbf{BA} = \mathbf{\Lambda P}, \tag{2.3}$$

where $\mathbf{\Lambda}$ is a non-singular diagonal matrix, and $\mathbf{P}$ is a permutation matrix. In other words, the sources may be recovered to within a scaling and permutation. Perfect recovery means that $\mathbf{BA} = \mathbf{I}$, i.e. $\mathbf{B}$ is the inverse of $\mathbf{A}$.

One widely applied approach to BSS is that of *independent component analysis* (ICA) [1]. The basic idea of ICA is that if no more than one source is Gaussian, the signals can be estimated with the simple constraint that recovered signals be statistically independent [1], [2]. The conditions of identifiability, separability, and uniqueness of linear ICA models are studied in [3].

2.2.2 Contrast Functions

In the ICA approach, BSS is usually obtained by defining and optimizing a real-valued *contrast function*, or cost function of the separating matrix $\mathbf{B}$. Since the contrast function measures the degree of independence of the separated components in $\mathbf{y}_k = \mathbf{B}\mathbf{x}_k$, it usually

has the form $L(\mathbf{B}) = E[G(\mathbf{y})]$. The contrast functions should be designed in a way such that when the sources are separated the optimal value of the contrast function is achieved [4]. There are several different statistical criteria that can be used to define contrast functions, and it has been shown that some of them are closely related. In this part, different examples of the criteria used in ICA methods are reviewed briefly.

Likelihood function

Assume that the sources have joint density function $f_{\mathbf{s}}(\mathbf{s}) = \prod_{i=1}^{n} f_{s(i)}(s(i))$. By virtue of (2.1), if $\mathbf{A}$ is a non-singular matrix, the joint density function of the observation can be expressed as

$$f_{\mathbf{x}}(\mathbf{x}) = |\det \mathbf{A}|^{-1} f_{\mathbf{s}}(\mathbf{A}^{-1}\mathbf{x}). \tag{2.4}$$

If the joint density function of the sources is known *a priori*, our goal is then to find a matrix $\mathbf{B} = \mathbf{A}^{-1}$ such that (2.4) is maximized. In other words, we want to get the maximum-likelihood estimate of $\mathbf{A}^{-1}$.

Using the logarithm of the likelihood function in (2.4), one can define the contrast function based on maximum likelihood as

$$L_{ML}(\mathbf{B}) = E[\sum_{i=1}^{n} \log\left(f_{s(i)}(y(i))\right)] + \log|\det \mathbf{B}|. \tag{2.5}$$

The expectation in the contrast function can be interpreted as the average computed from a finite set of observed samples.

The maximum likelihood contrast can be shown to be closely related to a group of contrast functions, such as the one based on mutual information of the separated symbols [5], and also the one based on cumulants [4]. These contrast functions are reviewed next.

Mutual information

The mutual information contrast is an information-theoretic measure of dependence between random variables. It is always nonnegative, and is zero when the variables are statistically independent of each other. As a result, the mutual information of the separated components can be used in the contrast function to achieve source separation [5], [6].

Mutual information can be interpreted using entropy or Kullback-Leibler divergence [7]. From either starting point, it can be shown that the mutual information for components in $\mathbf{y}$ can be defined as

$$I(y(1),...,y(n)) = -E[\log \frac{f(y(1))...f(y(n))}{f(y(1),...,y(n))}] = \sum_{i=1}^{n} H(y(i)) - H(\mathbf{y}), \qquad (2.6)$$

where the sub-index $s(i)$ of the density function is omitted, and $H(y(i)) = -E[\log(f(y(i)))]$ is the entropy. From (2.2) it follows that

$$H(\mathbf{y}) = H(\mathbf{x}) + \log|\det \mathbf{B}|. \qquad (2.7)$$

Since $H(\mathbf{x})$ is a constant, minimizing (2.6) leads to the contrast function

$$\begin{aligned} L_{MI}(\mathbf{B}) &= \sum_{i=1}^{n} H(y(i)) - \log|\det \mathbf{B}| \\ &= -\sum_{i=1}^{n} E[\log(f(y(i)))] - \log|\det \mathbf{B}| \end{aligned}. \qquad (2.8)$$

Comparing (2.8) and (2.5), we see that the two contrast functions differ only by a negative sign. As a result, minimization of the mutual information of the separated symbols is the same as maximization of the likelihood of the observations.

Non-Gaussianity

In addition to the criteria mentioned above, non-Gaussianity can also be used in the definition of contrast functions. According to the central limit theorem, when sources are linearly combined, the distribution of the mixed signal is closer to Gaussian than that of individual non-Gaussian sources [7]. The idea to maximize output non-Gaussianity is therefore to go in the opposite direction of mixing, i.e. to separate the signals.

From information theory we know that among random variables of equal variance, a Gaussian variable has the largest entropy. A measure that is zero for Gaussian variables and gets more positive for variables that are less Gaussian is *negentropy*. The contrast function based on negentropy can be expressed as

$$L_{NE}(\mathbf{B}) = \sum_{i=1}^{n} H(y_{Gauss}(i)) - H(y(i)), \tag{2.9}$$

where $y_{Gauss}(i)$ is a scalar Gaussian random variable that has the same mean and variance as $y(i)$, and $H(y(i))$ is the entropy of $y(i)$ [1].

When the separated symbol vector $\mathbf{y}$ is preprocessed to be uncorrelated, i.e. $E[\mathbf{y}\mathbf{y}^H] = \mathbf{I}$, we have

$$\det\left(E[\mathbf{y}\mathbf{y}^H]\right) = 1 = \det\left(\mathbf{B}E[\mathbf{x}\mathbf{x}^H]\mathbf{B}^H\right) = (\det\mathbf{B})\det\left(E[\mathbf{x}\mathbf{x}^H]\right)(\det\mathbf{B}^H). \tag{2.10}$$

This implies $\det \mathbf{B}$ should be a constant since $\det\left(E[\mathbf{x}\mathbf{x}^H]\right)$ does not depend on $\mathbf{B}$. If we look at the mutual information based contrast function (2.8), we can see that it can be written equivalently as

$$\begin{aligned} L_{MI}(\mathbf{B}) &= \sum_i H(y(i)) - \sum_i H(y_{Gauss}(i)) - \log|\det \mathbf{B}| + \sum_i H(y_{Gauss}(i)) \\ &= \text{constant} - L_{NE}(\mathbf{B}) \end{aligned} \tag{2.11}$$

As a result, when the separated symbols are constrained to be uncorrelated, minimizing mutual information of the estimated components is equivalent to maximizing the sum of their negentropies

Cumulants

The contrast functions mentioned above are based on at least an approximation of the source density function. Now we discuss an approach using high-order statistics that does not depend on the source density. For simplicity, source symbols are assumed to be real in this brief discussion..

Cumulants are high-order statistics that can be used to define contrast functions. For BSS, the most commonly used cumulants are the 2^{nd}- and 4^{th}- order ones, which can be defined as

$$Cum_{ij}(\mathbf{s}) \triangleq E[s(i)s(j)], \tag{2.12}$$

$$\begin{aligned} Cum_{ijkl}(\mathbf{s}) \triangleq\ & E[s(i)s(j)s(k)s(l)] - E[s(i)s(j)]E[s(k)s(l)] \\ & -E[s(i)s(k)]E[s(j)s(l)] - E[s(i)s(l)]E[s(j)s(k)]. \end{aligned} \tag{2.13}$$

Specially, the 2^{nd}-order cumulant terms in (2.12) compose the covariance matrix of vector $\mathbf{s}$.

When the sources in $\mathbf{s}$ are independent, the cross-correlation terms in (2.12) and (2.13) vanish, and we have that

$$Cum_{ij}(\mathbf{s}) \triangleq \sigma_i^2 \delta_{ij}, \tag{2.14}$$

$$Cum_{ijkl}(\mathbf{s}) \triangleq \kappa_i \delta_{ijkl} \ , \tag{2.15}$$

where δ_{ij} or δ_{ijkl} equals one when all the sub-indices are the same and zero otherwise; and σ_i^2 and κ_i are the variance and kurtosis of source s_i, i.e.

$$\sigma_i^2 \triangleq E[s(i)^2], \tag{2.16}$$

$$\kappa_i \triangleq E[s(i)^4] - 3E^2[s(i)^2]. \tag{2.17}$$

Cardoso pointed out in [4] that the following function can be shown to be a contrast function:

$$L_{cumu}(\mathbf{B}) = \phi_4(\mathbf{y}) = \sum_{ijkl} \left(Cum_{ijkl}(\mathbf{y}) - \kappa_i \delta_{ijkl} \right)^2 . \tag{2.18}$$

At the same time, this function can be interpreted as the quadratic mismatch of the cumulants.

The cumulants-based contrast function is closely related to the maximum likelihood function. When the $\mathbf{s}$ and $\mathbf{y}$ are symmetrically distributed with distribution that is close to normal, then using the Edgeworth expansion [8], the maximum likelihood based contrast function can be shown to be related to the following function with the 2nd- and 4th- order cumulants [4]:

$$L_{ML-Appro}(\mathbf{B}) = \frac{1}{48}\left(12\phi_2(\mathbf{y}) + \phi_4(\mathbf{y})\right) \tag{2.19}$$

where $\phi_4(\mathbf{y})$ was defined in (2.18) and

$$\phi_2(\mathbf{y}) = \sum_{ij} \left(Cum_{ij}(\mathbf{y}) - \sigma_i^2 \delta_{ij} \right)^2 . \tag{2.20}$$

Specially, when the outputs are constrained to be of zero mean and unit variance, $E[y(i)] = 0$ and $E[y(i)^2] = 1$, the term $\phi_2(\mathbf{y})$ defined in (2.20) becomes zero. Under the whitening constraint $\phi_2(\mathbf{y}) = 0$, it can be shown that $\phi_4(\mathbf{y})$ in (2.18) is equal to

$$\phi_4^{\circ}(\mathbf{y}) = -2\sum_i \kappa_i Cum_{iiii}(\mathbf{y}) = E[-2\sum_i \kappa_i (y(i)^4 - 3)] . \tag{2.21}$$

When the kurtoses of all the sources are negative, then (2.21) becomes the very simple and commonly used kurtosis-maximization contrast function used in [9], [10], [11]:

$$L_{KM}(\mathbf{B}) = E[\sum_i y(i)^4] . \tag{2.22}$$

We will see in Chapter 3 and 4 that this contrast function leads to a simple nonlinearity that can be used in ICA- based algorithms which can be interpreted as being based on non-linear decorrelation for independence.

2.2.3 Gradient and Online Algorithms

With a specific contrast function $L(\mathbf{B}) = E[G(\mathbf{y})]$ chosen for minimization, the classical approach for obtaining the minimum is *steepest descent* or *gradient descent*. In the gradient descent method, we start from an initial positon, and minimize the function iteratively by computing a gradient of the function at the current position and moving in the negative direction of the gradient by a certain amount [7]. The process is repeated until convergence.

For simplicity, we consider $\mathbf{B}$ to be a matrix with real-valued entries in this sub-section to illustrate the ideas of relative and natural gradient. More explanations will be

included in Section 5.3. The gradients for complex cases will also be derived and used in Chapter 5.

Relative Gradient and Natural Gradient

A widely-used gradient is the *standard gradient*, which is usually assumed at the mention of "gradient". For a contrast function $L(\mathbf{B})$ of matrix $\mathbf{B}$, the standard gradient of the function with respect to the variable $\mathbf{B}$ can be expressed as

$$\nabla_{\mathbf{B}} L(\mathbf{B}) = \frac{\partial L(\mathbf{B})}{\partial \mathbf{B}} \,. \tag{2.23}$$

It will be seen from detailed analysis to be given in Chapter 5 and its appendix that when the contrast function has the expression $L(\mathbf{B}) = E[G(\mathbf{y})]$, the standard gradient can be derived as

$$\nabla_{\mathbf{B}} L(\mathbf{B}) = E[\mathbf{g}(\mathbf{y})\mathbf{x}^T] \,, \tag{2.24}$$

where $\mathbf{g}$ is the component-wise derivative of function G at $\mathbf{y}$. With the standard gradient, the iterations for minimization are

$$\mathbf{B}_{k+1} = \mathbf{B}_k - \lambda \nabla_{\mathbf{B}} L(\mathbf{B}) \,, \tag{2.25}$$

where λ is the step-size.

Another gradient that can be used is the *relative gradient*. For a contrast function $L(\mathbf{B})$, a perturbation $\boldsymbol{\mathcal{E}}\mathbf{B}$ *proportional to the current value* of $\mathbf{B}$ is considered, where $\boldsymbol{\mathcal{E}}$ is a matrix with small entries [10]. Writing out the Taylor expansion for $L(\mathbf{B})$, we find

$$L(\mathbf{B} + \boldsymbol{\mathcal{E}}\mathbf{B}) = L(\mathbf{B}) + \text{trace}[(\nabla_{\mathbf{B}} L(\mathbf{B})\mathbf{B}^T)^T \boldsymbol{\mathcal{E}}] + o(\boldsymbol{\mathcal{E}}) \,. \tag{2.26}$$

The relative gradient of the function can be defined to be

$$\nabla_{\mathbf{B}}^{(R)} L(\mathbf{B}) = \nabla_{\mathbf{B}} L(\mathbf{B})\mathbf{B}^T = E[\mathbf{g}(\mathbf{y})\mathbf{y}^T]. \tag{2.27}$$

The relation between the relative gradient and the standard gradient can be seen from the expressions (2.24) and (2.27). With relative gradient, the adaptation for matrix $\mathbf{B}$ becomes

$$\mathbf{B}_{k+1} = \mathbf{B}_k - \lambda \nabla_{\mathbf{B}}^{(R)} L(\mathbf{B}_k)\mathbf{B}_k. \tag{2.28}$$

The concept of *natural gradient* was developed by Amari [6] and used for various problems including the BSS [12]. The starting points of natural gradient and the relative gradient are different. However, for the BSS problem where the variable $\mathbf{B}$ is a non-singular matrix, the two yield the same expression for gradient, and as a result the same adaptation. More details about the relative gradient and natural gradient will be given in Chapter 5.

Online Algorithms

In (2.25) and (2.28), the adaptation for matrix $\mathbf{B}$ with standard gradient and relative gradient have been given respectively. Although the contrast function includes expectation, which needs an estimate to be computed, in simple gradient descent methods the expectation is dropped and replaced by its instantaneous value. As a result, the gradient descent method is usually used as *stochastic gradient descent*.

With the relative gradient, or natural gradient, and dropping the expectation, the adaptation in (2.28) becomes

$$\mathbf{B}_{k+1} = \mathbf{B}_k - \lambda \mathbf{g}(\mathbf{y}_k)\mathbf{y}_k{}^T \mathbf{B}_k. \tag{2.29}$$

The iterative updating in (2.29) can be written as

$$\begin{aligned}\mathbf{B}_{k+1} &= \mathbf{B}_k - \lambda \mathbf{U}(\mathbf{y}_k)\mathbf{B}_k \\ &= [\mathbf{I} - \lambda \mathbf{U}(\mathbf{y}_k)]\mathbf{B}_k\end{aligned}, \tag{2.30}$$

where $\mathbf{U}(\mathbf{y}_k) = \mathbf{g}(\mathbf{y}_k)\mathbf{y}_k^T$. Online algorithms that have the form of (2.30) can be called "*serial updating*". The advantage of such an algorithm is the property of "*equivariance*". If we multiply both sides of (2.30) from the right by the mixing matrix $\mathbf{A}$, we have

$$\mathbf{C}_{k+1} = [\mathbf{I} - \lambda \mathbf{U}(\mathbf{C}_k \mathbf{s}_k)]\mathbf{C}_k, \tag{2.31}$$

where $\mathbf{C}_k \triangleq \mathbf{B}_k \mathbf{A}$.

In (2.31), the adaptation is characterized by the global system $\mathbf{C}_k$ and the source symbols. For two mixing matrices $\mathbf{A}$ and $\mathbf{A}'$ with the same sources, if we initialize the separating matrix to be $\mathbf{B}_0$ and $\mathbf{B}_0'$, as long as $\mathbf{B}_0\mathbf{A} = \mathbf{B}_0'\mathbf{A}'$, the trajectory of the global system $\mathbf{C}$ will be identical. In this case, we say that the adaptive algorithm is *equivariant*, and offers *uniform* performance. The equivariance property enables one to deal with BSS problems with an ill-conditioned mixing matrix, and consider the global system as a whole.

2.2.4 Whitening and Orthogonalization

Whitening

In BSS, to make it easier to separate the sources, the observed data is often preprocessed to have uncorrelated components. Mathematically, a zero-mean random vector $\mathbf{z}$ is said to be *white* if its elements are uncorrelated with each other and have unit variance. In other words, for $\mathbf{z}$ that is white

$$E[\mathbf{z}\mathbf{z}^H] = \mathbf{I}. \tag{2.32}$$

Whitening can be obtained through eigenvalue decomposition (EVD). Suppose we want to whiten the mixed signals $\mathbf{x}$ with a linear matrix $\mathbf{V}$, and it is transformed to be $\mathbf{z} = \mathbf{V}\mathbf{x}$. When $\mathbf{z}$ is white, we have

$$E[\mathbf{z}\mathbf{z}^H] = \mathbf{V}E[\mathbf{x}\mathbf{x}^H]\mathbf{V}^H = \mathbf{I}. \tag{2.33}$$

Let $\mathbf{C}_{\mathbf{xx}} = E[\mathbf{x}\mathbf{x}^H]$ be the covariance matrix of $\mathbf{x}$, then by EVD the matrix $\mathbf{C}_{\mathbf{xx}}$ can be written as

$$\mathbf{C}_{\mathbf{xx}} = \mathbf{E}\mathbf{D}\mathbf{E}^H, \tag{2.34}$$

where $\mathbf{E}$ is the matrix whose columns are the eigenvectors of the covariance matrix $\mathbf{C}_{\mathbf{xx}}$, and $\mathbf{D}$ is the diagonal matrix of the eigenvalues of $\mathbf{C}_{\mathbf{xx}}$. As a result, a linear operation with matrix $\mathbf{V} = \mathbf{D}^{-1/2}\mathbf{E}^H$ will make $\mathbf{z} = \mathbf{V}\mathbf{x}$ become white. In fact, whitening matrix is not unique, since we can pre-multiply $\mathbf{V}$ with an orthogonal matrix and still keep the covariance matrix of $\mathbf{z}$ identity.

In addition to algebraic methods such as EVD, whitening can also be performed with on-line algorithms. As stated in [10], the *Kullback-Leibler* divergence between two normal distributions with covariances $\mathbf{R}_z$ and $\mathbf{I}$ can be expressed as

$$K(\mathbf{z}) \triangleq \frac{1}{2}\left[\text{Trace}(\mathbf{R}_z) - \log\det(\mathbf{R}_z) - n\right] \tag{2.35}$$

where n is the size of vector $\mathbf{z}$. We see that $K(\mathbf{z}) \geq 0$ with equality if and only if $\mathbf{R}_z = \mathbf{I}$. With the cost function $L(\mathbf{V}) = K(\mathbf{z}) = K(\mathbf{V}\mathbf{x})$ and using relative gradient [7], [10], whitening can be performed iteratively with

$$\mathbf{V}_{k+1} = \mathbf{V}_k - \lambda[\mathbf{z}_k\mathbf{z}_k{}^H - \mathbf{I}]\mathbf{V}_k. \tag{2.36}$$

As has been seen in Section 2.2.2 some contrast functions are defined based on the assumption that the mixed signals are white. In such cases, it is important to pre-process the observations so that they are whitened.

Orthogonalization

With whitening as a pre-processing step, the BSS problem reduces to finding an orthogonal matrix $\mathbf{Q}$ such that $\mathbf{Qz} \simeq \mathbf{s}$, i.e. the separating matrix is $\mathbf{B} = \mathbf{QV}$. The orthogonality is based on the pre-whitening step, and is said to operate under the whiteness constraint [13].

With a particular contrast function, the gradient descent method for updating an orthogonal $\mathbf{Q}$ will not necessarily make it remain orthogonal automatically. As a result, it may be beneficial to orthogonalize the matrix at the end of each iteration.

Orthogonalization can be accomplished in many ways [7], and one method uses the following procedure:

$$\mathbf{Q} \leftarrow \mathbf{Q}(\mathbf{Q}^H\mathbf{Q})^{-1/2}. \tag{2.37}$$

It can be shown that the operation in (2.37) is the orthogonal projection of matrix $\mathbf{Q}$ onto the set of orthogonal matrices [14]. The drawback of the method of (2.37) is that a matrix inverse is involved at each iteration. A simpler approach is the following two-step iterative updating:

$$\begin{aligned} &\mathbf{Q} \leftarrow \mathbf{Q}/\|\mathbf{Q}\| \\ &\mathbf{Q} \leftarrow \frac{3}{2}\mathbf{Q} - \frac{1}{2}\mathbf{Q}\mathbf{Q}^H\mathbf{Q} \end{aligned}, \tag{2.38}$$

which is implemented until $\mathbf{QQ}^H$ is close to identity.

The orthogonality and the whitening constraint can also be combined during adaptation. In [10], Cardoso proposed the *equivariant adaptive separation via independence* (EASI) algorithm which combines the whitening constraint and preserves orthogonality in one step. In the EASI algorithm, the adaptation for the separation matrix is

$$\mathbf{B}_{k+1} = \mathbf{B}_k - \lambda\left[\mathbf{y}_k\mathbf{y}_k{}^H - \mathbf{I} + \mathbf{g}(\mathbf{y}_k)\mathbf{y}_k{}^H - \mathbf{y}_k\mathbf{g}(\mathbf{y}_k)^H\right]\mathbf{B}_k, \tag{2.39}$$

where the vector $\mathbf{g}(\mathbf{y}_k)$ is the component-wise derivative of the function $G(\mathbf{y})$ in the contrast function at $\mathbf{y}_k = \mathbf{B}_k\mathbf{x}_k$. In the updating scheme (2.39), the first two terms in brackets effect a whitening constraint. The last two terms are to make the relative change *skew symmetric* so that the orthogonality is obtained with the whitening constraint. The $G(\mathbf{y})$ function is based on criteria such as those introduced in Section 2.2.2, however it can more generally be taken to be some reasonable nonlinear function. The term $\mathbf{g}(\mathbf{y}_k)\mathbf{y}_k{}^H$ can be interpreted as forcing nonlinear decorrelation for independence of the separated symbols.

2.3 Review of Blind Equalization

2.3.1 Model of Blind Equalization

Consider a complex symbol sequence $\{s(k)\}$ transmitted through an FIR complex channel. The notation for the source symbols here is slightly different from what we set up for the BSS model, but the context makes it very clear. The general model of BE is shown in Fig. 2.1.

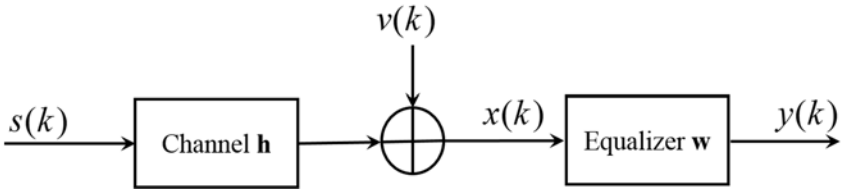

Fig. 2.1 General model of blind equalization.

For symbol-rate sampling, the output of the channel at time k can be expressed as

$$x(k) = \sum_{l=0}^{L} h(l)\, s(k-l) + v(k)\,, \tag{2.40}$$

where $\mathbf{h} = [h(0), h(1), ..., h(L)]^T$ is the channel response, and $\{v(k)\}$ is an additive white Gaussian noise sequence. The input source sequence is generally assumed to be i.i.d, but some BE algorithms work when the source symbols are correlated.

In BE, suppose an M- th order equalizer with impulse response $\mathbf{w} = [w(0), w(1), ..., w(M)]^T$ is to be designed, such that the output of the equalizer is

$$y(k) = \sum_{m=0}^{M} w(m)x(k-m) = \mathbf{w}^T \mathbf{x}_k\,, \tag{2.41}$$

where $\mathbf{x}_k = [x(k), x(k-1), ..., x(k-M+1)]^T$ is a length - $(M+1)$ vector containing the current and past M channel outputs.

Ideally, $y(k)$ is an approximation of the input symbol with some delay d and possibly a phase shift θ. In other words, when the impulse response of the channel-equalizer cascaded system is approximately

$$\mathbf{c} \triangleq \mathbf{h} * \mathbf{w} \simeq \mathbf{c}^{ideal} = e^{j\theta}(\underbrace{0, ..., 0}_{d \text{ zeros}}, 1, 0...0)\,, \tag{2.42}$$

we consider $\mathbf{w}$ to be an ideal equalizer. With an ideal equalizer, the source symbols can be recovered up to a fixed delay and phase shift.

In most adaptive BE algorithms, a single equalizer output is processed each time. To achieve BE, a cost function $J(\mathbf{w}) = E[G(y(k))]$ that is based on the fit of equalizer outputs to some known signaling constellation property of the source constellation may be defined and optimized.

Suppose the channel outputs are processed in blocks of size $P+M$ ($P > 0$) at symbol rate, by sliding along the sequence of channel outputs, with one-symbol shift each time. Let $\tilde{\mathbf{x}}_k = [x(k), x(k-1), ..., x(k-P-M+1)]^T$ be the k-th output block from the channel at time k, which is influenced by the length- $(P+M+L)$ source vector $\mathbf{s}_k = [s(k), s(k-1), ..., s(k-P-M-L+1)]^T$. The channel is assumed to be almost stationary over this observation period. The channel output block of length $P+M$ can be expressed in matrix form as

$$\tilde{\mathbf{x}}_k = \mathbf{H}\mathbf{s}_k + \mathbf{v}_k, \tag{2.43}$$

where $\mathbf{v}_k = [v(k), v(k-1), ..., v(k-P-M-L+1)]^T$ is the additive noise vector, and $\mathbf{H}$ is a $(P+M)\times(P+M+L)$ Toeplitz matrix containing the channel response:

$$\mathbf{H} = \begin{pmatrix} h(0) & h(1) & \cdots & h(L) & \cdots & \cdots & 0 \\ & \ddots & & \cdots & & \ddots & \cdots \\ \cdots & h(0) & h(1) & \cdots & h(L) & \cdots & \\ & & \ddots & & \cdots & & \ddots \\ 0 & \cdots & \cdots & h(0) & h(1) & \cdots & h(L) \end{pmatrix}. \tag{2.44}$$

When $P=1$, we can produce only one equalizer output; while when $P>1$, there are multiple equalizer output symbols that can be produced at each time. With $P>1$, the model

in (2.43) has a form similar to the BSS model of (2.1) (ignoring the noise term), and this provides another point of view for the BE problem. Note that the $\mathbf{H}$ matrix in (2.43) is an underdetermined matrix with more columns than rows. Although the dimension of $\mathbf{H}$ does not satisfy the requirement of a standard BSS problem, it will be seen in later chapters that with some constraints the BE can nonetheless be obtained with ICA-based algorithms.

2.3.2 Blind Equalization Algorithms

Many well-known algorithms for BE use steepest stochastic gradient descent methods for adaptive updates. With a defined cost function $J(\mathbf{w}) = E[G(y(k))]$ based on some source constellation property, the equalizer coefficients are updated generally according to

$$\begin{aligned} \mathbf{w}_{k+1} &= \mathbf{w}_k - \mu \nabla_{\mathbf{w}^*} J(\mathbf{w}) \\ &= \mathbf{w}_k - \mu g(y(k)) \mathbf{x}_k^*, \end{aligned} \tag{2.45}$$

where $\nabla_{\mathbf{w}^*} J(\mathbf{w})$ is the standard gradient of the cost function $J(\mathbf{w})$ with respect to the conjugate of the equalizer vector for complex case, and $g(y(k))$ is the derivative of $G(y(k))$ with respect to the conjugate of the equalizer output $y(k)$. The details of obtaining gradient when the variable is complex will be shown in the appendix in Chapter 5. In the steady state, if the coefficients of the equalizer converge, we have approximately

$$\mathbf{w}_{k+1} = \mathbf{w}_k \;\Rightarrow\; \nabla_{\mathbf{w}^*} J(\mathbf{w}) = 0. \tag{2.46}$$

Following (2.46), it can be shown that for any integer *m*, the equation

$$E[g(y(k))y(k-m)^*] = 0 \tag{2.47}$$

holds at the steady state, for any integer m . This condition (2.47) is called the *Bussgang condition*, and an algorithm that has the adaptation of the form (2.45) is called a *Bussgang-type algorithm*.

Next we will give several examples of the widely used Bussgang-type algorithms, with their cost functions and adaptive update equations. More details about these adaptive BE algorithms will be explained in Chapter 5.

Sato algorithm

The pioneer BE algorithm was the *Sato algorithm* for PAM signal [15], whose cost function is

$$J_{\text{Sato}}(\mathbf{w}) = E[(y(k) - R_{\text{Sato}}\text{sgn}(y(k)))^2],$$

where $R_{\text{Sato}} = \dfrac{E[|s(k)|^2]}{E[|s(k)|]}$. The adaptation for the Sato algorithm is

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mu\big(y(k) - R_{\text{Sato}}\text{sgn}(y(k))\big)\mathbf{x}_k .$$

Generalized Sato algorithm

A generalization of the Sato algorithm to complex signals is the *generalized Sato algorithm* (GSA) [16], with cost function

$$J_{\text{GSA}}(\mathbf{w}) = E[|y(k) - R_{\text{GSA}}\text{csgn}(y(k))|^2],$$

where $\text{csgn}(y(k)) = \text{sgn}(y_R(k)) + j\,\text{sgn}(y_I(k))$ for complex valued number $y(k) = y_R(k) + jy_I(k)$, and $R_{\text{GSA}} = \dfrac{E[|s_R(k)|^2]}{E[|s_R(k)|]} = \dfrac{E[|s_I(k)|^2]}{E[|s_I(k)|]}$. The adaptation is

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mu\big(y(k) - R_{\text{GSA}}\text{csgn}(y(k))\big)\mathbf{x}_k^* .$$

Godard algorithm/ constant modulus algorithm

The *Godard algorithm* [17] has the cost function defined as

$$J_{\text{Godard}}(\mathbf{w}) = E[(|\, y(k)\,|^p - R_p)^2],$$

where $R_p = \dfrac{E[|\, s(k)\,|^{2p}]}{E[|\, s(k)\,|^p]}$. The adaptation for the Godard algorithm is

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mu(|\, y(k)\,|^p - R_p)y(k)\,|\, y(k)\,|^{p-2}\ \mathbf{x}_k^* \ .$$

A special case of the Godard algorithm for $p = 2$ is the *constant modulus algorithm* (CMA) proposed in [18], with the cost function

$$J_{\text{CMA}}(\mathbf{w}) = E[(|\, y(k)\,|^2 - R_{\text{CMA}})^2].$$

The adaptation for the CMA is

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mu(|\, y(k)\,|^2 - R_{\text{CMA}})y(k)\mathbf{x}_k^* \ .$$

Multimodulus algorithm

The *multimodulus algorithm* (MMA) was proposed in [19] that provides more flexibility than the CMA. The MMA has cost function

$$J_{\text{MMA}}(\mathbf{w}) = E[(|\, y_R(k)\,|^p - R_{\text{MMA}}^p)^2 + (|\, y_I(k)\,|^p - R_{\text{MMA}}^p)^2],$$

where $R_{\text{MMA}}^p = \dfrac{E[|\, s_R(k)\,|^{2p}]}{E[|\, s_R(k)\,|]} = \dfrac{E[|\, s_I(k)\,|^{2p}]}{E[|\, s_I(k)\,|]}$. The adaptation for the MMA has the expression

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mu e_{\text{MMA}}(k)y(k)\mathbf{x}_k^* \ ,$$

where

$$e_{\text{MMA}}(k) = e_R(k) + je_I(k) \ ,$$

$$e_R(k) = (|\, y_R(k)\,|^p - R_{\text{MMA}}^p)y_R(k)\,|\, y_R(k)\,|^{p-2},$$

$$e_I(k) = (|\, y_I(k)\,|^p - R_{\text{MMA}}^p)y_I(k)\,|\, y_I(k)\,|^{p-2}.$$

Square contour algorithm

The square contour algorithm (SCA) in [20] combines the idea of the CMA and the Sato, with the cost function

$$J_{\mathrm{SCA}}(\mathbf{w}) = E[(|\, y_R(k) + y_I(k)\,| + |\, y_R(k) - y_I(k)\,| - R_{\mathrm{SCA}})^2],$$

where $R_{\mathrm{SCA}} = 2\max\{|\, s_R(k), s_I(k)\,|\}$. Then the adaptation for the SCA is

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mu\{(|\, y_R(k) + y_I(k)\,| + |\, y_R(k) - y_I(k)\,| - R_{\mathrm{SCA}}) \cdot \begin{pmatrix} (\mathrm{sgn}[y_R(k) + y_I(k)\,|] + \mathrm{sgn}[y_R(k) - y_I(k)\,|]) \\ + j(\mathrm{sgn}[y_R(k) + y_I(k)\,|] - \mathrm{sgn}[y_R(k) - y_I(k)\,|]) \end{pmatrix} \} \mathbf{x}_k^*.$$

ICA-based algorithm

Besides the above Bussgang-type algorithms that exploit some constellation properties of the source symbols, BE schemes have also been modeled as a BSS problem, and solved using ICA-based algorithms [21], [22], when the source symbols are independent and identically distributed (i.i.d.).

To allow exploitation of independence between symbols, *fractional sampling* can be applied so that the model satisfies the requirement of a standard BSS problem [22]. In [23], a block transmission scheme is considered, with which the BE problem can be formulated as a standard BSS model.

Blind equalization problems with respect to minimum phase and non-minimum phase systems have been discussed widely in [16], [24]–[32]. With only the second-order statistics of the source, the channel phase cannot be determined [29], [32]. The algorithms mentioned above use 1st order or high order statistics, and they have been used in both minimum phase and non-minimum phase systems, although it has been shown that some adaptive algorithms may converge to local minima. In [21], the authors show with

simulation results that the ICA-based algorithms work for non-minimum phase systems. In this dissertation, the analysis of identifiability of the channel phase information is not the main focus. In the next several chapters, our proposed algorithms will be examined in multiple channels including non-minimum phase channels. The performance of the algorithms will be analyzed as we explain more details.

In the BE problem for complex signals, it is generally assumed that complex sources are proper or circular, i.e. $E[s(k)^2]=0$ [33]. Also in the widely used BE algorithms that have been reviewed, a linear filter is usually used without the specific assumption of the circular property of the source. However, it has been shown in [34] that when the sources are improper and the channels are complex, the performance with the equalizer as a linear filter may lead to undesirable solution. The exploration of noncircular sources has motivated research both on blind equalization and blind source separation [35]–[37]. To improve performance of blind equalization for improper complex sources, a widely linear filter is used by processing the channel output $\mathbf{x}_k$ and its conjugate $\mathbf{x}_k^*$ as well [37]–[39]. In our work in the later chapters, we will develop new algorithms for general complex sources. The proposed algorithms may have a limitation in equalizing improper sources, while the idea with relative gradient and matrix structure constraint may be applied in equalization with widely linear filter for better performance.

2.4 Conclusion

In this chapter, we reviewed the basic concepts, models, and approaches for BSS and BE. Furthermore, the connection between the BSS and BE was also discussed. In Chapters 3

and 4, we will explain how the ICA-based algorithms, which have been used in BSS problems, can be applied to obtain BE under certain constraints. In Chapter 5, the idea of relative gradient will be used to improve the standard Bussgang-type BE algorithms for better performance. In Chapter 6, channel shortening for OFDM systems with the relative gradient will be shown.

References

[1] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, vol. 46. John Wiley & Sons, 2002.

[2] P. Comon, "Independent component analysis, A new concept?," *Signal Processing*, vol. 36, no. 3, pp. 287–314, 1994.

[3] J. Eriksson and V. Koivunen, "Identifiability and separability of linear ICA models," *IEEE Signal Process. Lett.*, vol. 11, no. 7, pp. 601–604, 2004.

[4] J.-F. C. Cardoso, "Blind signal separation: statistical principles," *Proc. IEEE*, vol. 86, no. 10, pp. 2009–2025, 1998.

[5] D.-T. Pham, "Blind separation of instantaneous mixture of sources via the Gaussian mutual information criterion," *Signal Processing*, vol. 81, no. 4, pp. 855–870, 2001.

[6] S. Amari, A. Cichocki, and H. Yang, "A new learning algorithm for blind signal separation," *Adv. Neural Inf. Process. Syst.*, vol. 8, pp. 757–763, 1996.

[7] A. Hyvärinen and E. Oja, "Independent component analysis: Algorithms and applications," *Neural Networks*, vol. 13, no. 4–5, pp. 411–430, 2000.

[8] J. E. Kolassa, *Series Approximation Methods in Statistics*. Springer Secient and Business Media, 2006.

[9] J. F. Cardoso, "High-order contrasts for independent component analysis," *Neural Comput.*, vol. 11, no. 1, pp. 157–192, 1999.

[10] J. F. Cardoso and B. H. Laheld, “Equivariant adaptive source separation,” *IEEE Trans. Signal Process.*, vol. 44, no. 12, pp. 3017–3030, 1996.

[11] A. Hyvärinen and O. Erkki, “A fast fixed-point algorithm for independent component analysis,” *Neural Comput.*, vol. 9, no. 7, pp. 1483–1492, 1997.

[12] S. Amari, “Natural Gradient Works Efficiently in Learning,” *Neural Comput.*, vol. 10, no. 2, pp. 251–276, 1998.

[13] P. Comon and C. Jutten, *Handbook of Blind Source Separation: Independent component analysis and applications*. Academic press, 2010.

[14] D. G. Luenberger, *Optimization by vector space methods*. John Wiley & Sons, 1997.

[15] Y. Sato, “A Method of Self-Recovering Equalization for Multilevel Amplitude-Modulation Systems,” *IEEE Trans. Commun.*, vol. 23, no. 6, pp. 679–682, 1975.

[16] A. Benveniste and M. Goursat, “Blind Equaizers,” *IEEE Trans. Commun.*, vol. 32, no. 8, pp. 871–879, 1984.

[17] D. N. Godard, “Self-Recovering Equalization and Carrier Tracking in Two-Dimensional Data Communication Systems,” *IEEE Trans. Commun.*, vol. 28, no. 11, pp. 1867–1875, 1980.

[18] J. Treichler and B. G. Agee, “A new approach to multipath correction of constant modulus signals,” *IEEE Trans. Acoust.*, vol. 31, no. 2, pp. 459–472, 1983.

[19] J. Yang, J.-J. Werner, G. A. Dumont, and Others, “The multimodulus blind equalization and its generalized algorithms,” *IEEE J. Sel. Areas Commun.*, vol. 20, pp. 997–1015, 2002.

[20] T. Thaiupathump, L. He, and S. A. Kassam, “Square contour algorithm for blind equalization of QAM signals,” *Signal Processing*, vol. 86, pp. 3357–337–, 2006.

[21] H. H. Yang, “On-line blind equalization via on-line blind separation,” *Signal Processing*, vol. 68, no. 3, pp. 271–281, 1998.

[22] Y. Zhang and S. A. Kassam, “Blind separation and equalization using fractional sampling of digital communications signals,” *Signal Processing*, vol. 81, no. 12, pp. 2591–2608, 2001.

[23] Z. Wu and S. A. Kassam, “Blind Equalization Based On Blind Separation with Toeplitz Constraint,” in *Signals, Systems and Computers, 2014 48th Asilomar Conference on*, 2014, pp. 1453–1457.

[24] S. Bellini, “Blind equalization,” *Alta Freq.*, vol. 57, no. 7, pp. 445–450, 1988.

[25] S. Bellini, “Blind equalization and deconvolution,” in *SPIE Adaptive Signal Processing*, 1991, vol. 1565, pp. 88–101.

[26] S. Bellini, “Bussgang techniques for blind equalization,” in *IEEE Global Telecommunications Conference*, 1986, pp. 1634–1640.

[27] O. Shalvi and E. Weinstein, “New criteria for blind deconvolution of nonminimum phase systems (channels),” *IEEE Trans. Inf. Theory*, vol. 36, no. 2, pp. 312–321, 1990.

[28] H.-H. Chiang and C. L. Nikias, “Adaptive deconvolution and identification of nonminimum phase FIR systems based on cumulants,” *Autom. Control. IEEE Trans.*, vol. 35, no. 1, pp. 36–47, 1990.

[29] S. Zheng, Fu-Chun McLaughlin and B. Mulgrew, “Blind equalization of nonminimum phase channels: higher order cumulant based algorithm,” *Signal Process. IEEE Trans.*, vol. 41, no. 2, pp. 681–691, 1993.

[30] L. Tong, G. Xu, and T. Kailath, “Blind identification and equalization based on second-order statistics: A time domain approach,” *Inf. Theory, IEEE Trans.*, vol. 40, no. 2, pp. 340–349, 1994.

[31] G. B. Giannakis and J. M. Mendel, “Identification of nonminimum phase systems using higher order statistics,” *Acoust. Speech Signal Process. IEEE Trans.*, vol. 37, no. 3, pp. 360–377, 1989.

[32] A. Benveniste, M. Goursat, and G. Ruget, “Robust identification of a nonminimum phase system: Blind adjustment of a linear equalizer in data communications,” *Autom. Control. IEEE Trans.*, vol. 25, no. 3, pp. 385–399, 1980.

[33] T. Adali, P. J. Schreier, and L. L. Scharf, “Complex-valued signal processing: The proper way to deal with impropriety,” *IEEE Trans. Signal Process.*, vol. 59, no. 11, pp. 5101–5125, 2011.

[34] C. B. Papadias, “On the existence of undesirable global minima of Godard equalizers,” in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, 1997, pp. 3941–3944.

[35] S. Javidi, D. P. Mandic, and A. Cichocki, “Complex blind source extraction from noisy mixtures using second-order statistics,” *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 57, no. 7, pp. 1404–1416, 2010.

[36] P. J. Schreier, T. Adali, and L. L. Scharf, "On ICA of improper and noncircular sources," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 3561–3564.

[37] T. Zhang, B. Wang, and S. Liu, "Widely linear RLS constant modulus algorithm for complex-valued noncircular signals," *J. Electron.*, vol. 31, no. 5, pp. 416–426, 2014.

[38] D. Darsena, G. Gelli, L. Paura, and F. Verde, "Widely linear equalization and blind channel identification for interference-contaminated multicarrier systems," *IEEE Trans. Signal Process.*, vol. 53, no. 3, pp. 1163–1177, 2005.

[39] A. S. Cacciapuoti, G. Gelli, L. Paura, and F. Verde, "Widely-linear fractionally-spaced blind equalization of frequency-selective channels," in *Signal Processing Conference, 14th European*, 2006.

Chapter 3

Constrained ICA for Blind Equalization with Block Transmission

3.1 Introduction

In Chapter 2 we have seen that the blind equalization (BE) problem can be formulated in matrix form, which is similar to the form of the blind source separation (BSS) problem based on independent component analysis (ICA). However, the "mixing" matrix in the general BE problem does not satisfy the dimension requirement of the standard BSS problem.

To make our problem fit a BSS model that will allow BE to be accomplished, we consider two *block transmission* schemes, with zero-padding or cyclic prefix. These single carrier modulation schemes have received interest in broadband communication, and have been compared with the orthogonal frequency division multiplexing (OFDM) technique

[1]–[4]. Our study of block-based transmission leads naturally to consideration of Toeplitz- and circulant-matrix-constrained BE algorithms based on independence. In the next chapter we will build on these ideas and consider the continuous transmission case, and show that constrained BSS schemes are also quite effective without a block-based transmission scheme.

In Section 3.2, the block transmission scheme with zero padding is analyzed. We show how to formulate the BE problem as a standard BSS problem, and how constrained ICA-based algorithms can be applied. In Section 3.3, block transmission with a cyclic prefix is considered in a parallel way to Section 3.2. For both of these schemes it is shown in Section 3.4 that the matrix adaptations of the corresponding BE algorithms can be implemented in a computationally efficient way. In Section 3.5, we explain how an I/Q independence constraint for two-dimensional constellations can be incorporated to reduce phase ambiguity for source symbols with independent I/Q parts. Simulation results and comparisons are shown in Section 3.6. We discuss briefly the relative merits of the two block transmission schemes in Section 3.7, and conclude the chapter with Section 3.8.

3.2 Block Transmission with Zero Padding

3.2.1 Formulation

We first consider a transmission scheme where the source symbols are transmitted through a channel with impulse response $\mathbf{h} = [h(0), h(1), ..., h(L)]^T$ in blocks of size Q with L

zeros padded at the beginning of each block [5]. Without loss of generality, we assume that $h(0) \neq 0$. Also it is assumed that $Q > L$.

Denote by $\mathbf{s}_k^{\text{Zblock}} = [(\mathbf{s}_k^{\text{block}})^T, 0, ..., 0]^T$ the k-th block of the transmitted symbols with zero paddings, where $\mathbf{s}_k^{\text{block}} = [s_k(Q), s_k(Q-1), ..., s_k(1)]^T$. The length-$Q$ vector $\mathbf{s}_k^{\text{block}}$ is defined this way so that its first symbol is the latest transmitted one, and the last one is the one that transmitted earliest. The length of the padded zeros is L, and is equal to the assumed maximum order of the channel. The corresponding noise vector that affects the channel outputs is $\mathbf{v}_k^{\text{Zblock}}$. The whole block with length $Q+L$ including the paddings will affect a total of Q channel outputs. According to the model of BE in matrix form in (2.44), the observation block $\mathbf{x}_k^{\text{Zblock}}$ can be written as

$$\mathbf{x}_k^{\text{Zblock}} = \mathbf{H}\mathbf{s}_k^{\text{Zblock}} + \mathbf{v}_k^{\text{Zblock}}. \tag{3.1}$$

In the above model (3.1) the last L symbols of $\mathbf{s}_k^{\text{Zblock}}$ are the padded zeros and will not contribute to the observed mixture $\mathbf{x}_k^{\text{Zblock}}$. As a result, equation (3.1) reduces equivalently to

$$\mathbf{x}_k^{\text{Zblock}} = \mathbf{H}_T\mathbf{s}_k^{\text{block}} + \mathbf{v}_k^{\text{Zblock}}, \tag{3.2}$$

where $\mathbf{H}_T$ is a $Q \times Q$ square Toeplitz matrix with lower diagonals zero, consisting of the first Q columns of matrix $\mathbf{H}$. From (3.2) it can be seen that the block of channel outputs at time k is affected only by the source symbols in the k-th block.

Since it is assumed that $h(0) \neq 0$, $\mathbf{H}_T$ is full rank. As a result, matrix $\mathbf{H}_T$ has a unique left inverse. The recovery of $\mathbf{s}_k^{\text{block}}$ from $\mathbf{x}_k^{\text{Zblock}}$ in (3.2) is a standard BSS problem, and it can be attempted using a standard BSS algorithm.

Let $\tilde{\mathbf{W}}$ be the "separating" matrix in this zero-padded block transmission BE problem, then the goal is to find a matrix $\tilde{\mathbf{W}}$ such that $\tilde{\mathbf{C}} \triangleq \tilde{\mathbf{W}}\mathbf{H}_T \simeq \mathbf{I}$. When the BSS adaptation converges, if the separating matrix $\tilde{\mathbf{W}}$ is a good approximation of $\mathbf{H}_T^{-1}$, the source symbols will be well recovered. From the structure of $\mathbf{H}$ in (2.44), we know that $\mathbf{H}_T$ is a square Toeplitz matrix with lower triangular elements zero. The Toeplitz structure is maintained under inversion [6], [7], so $\mathbf{H}_T^{-1}$ should also be a square Toeplitz matrix with lower diagonals zero.

Now let us take a slight detour to consider the elements in $\mathbf{W}_T \triangleq \mathbf{H}_T^{-1}$. Let the first row of $\mathbf{W}_T$ be $\mathbf{w}_T^T = [w_T(1), w_T(2), ..., w_T(Q)]$. Then

$$\mathbf{w}_T^T \mathbf{H}_T = \mathbf{e}_1^T, \tag{3.3}$$

where $\mathbf{e}_1$ is a length Q column vector with $e_1(1) = 1$, $e_1(j) = 0$ for $j = 2, ..., Q$. Because of the Toeplitz structure of $\mathbf{H}_T$, the vector on the right side of (3.3) via matrix multiplication can be equivalently obtained as the truncated convolution of $\mathbf{w}_T$ and $\mathbf{h}$. We denote by $(\mathbf{h} * \mathbf{w}_T)_{1:Q}$ the column vector of the first Q terms of the convolution of $\mathbf{w}_T$ and $\mathbf{h}$. We have

$$\mathbf{w}_T^T \mathbf{H}_T = \left((\mathbf{h} * \mathbf{w}_T)_{1:Q}\right)^T = \mathbf{e}_1^T. \tag{3.4}$$

Let us define the "inverse" of $\mathbf{h}$ as the column vector containing the coefficients of the inverse z-transform of $\mathbf{h}$, which can be obtained by long division. From (3.4) we see that $\mathbf{w}_T$ is the truncated version of the inverse of $\mathbf{h}$ containing only the first Q taps.

Since $\mathbf{H}_T$ has Toeplitz structure, based on (3.3), we also have

$$[\underbrace{0 \ \ldots \ 0}_{i-1 \text{ zeros}} \ w_T(1) \ w_T(2) \ \ldots \ w_T(Q-i+1)]\mathbf{H}_T = \mathbf{e}_i^T \tag{3.5}$$

where $i = 1, 2, ..., Q$, and $\mathbf{e}_i$ is a length-Q vector with the i-th element one and zero elsewhere. Denoting the transpose of the vector on the left side in (3.5) by $\mathbf{w}_T^{(i)}$ (note $\mathbf{w}_T^{(1)} = \mathbf{w}_T$), we have

$$\left[\mathbf{w}_T^{(1)} \ \ \mathbf{w}_T^{(2)} \ \ \ldots \ \ \mathbf{w}_T^{(Q)}\right]^T \mathbf{H}_T = \left[\mathbf{e}_1 \ \ \mathbf{e}_2 \ \ldots \ \mathbf{e}_Q\right]^T = \mathbf{I}_Q . \tag{3.6}$$

From the above equation we see that the inverse of $\mathbf{H}_T$, where $\mathbf{H}_T^{-1} = \left[\mathbf{w}_T^{(1)} \ \ \mathbf{w}_T^{(2)} \ \ \ldots \ \ \mathbf{w}_T^{(Q)}\right]^T$, is a $Q \times Q$ Toeplitz matrix containing the first Q taps of the inverse of $\mathbf{h}$, with lower triangular elements zero.

Let $\tilde{\mathbf{w}}^T$ be the first row of the separating matrix $\tilde{\mathbf{W}}$. If $\tilde{\mathbf{C}} = \tilde{\mathbf{W}}\mathbf{H}_T$ is close to the identity matrix, then $\tilde{\mathbf{W}}$ is approximately $\mathbf{H}_T^{-1}$, and then $\tilde{\mathbf{w}}$ is approximately $\mathbf{w}_T$, the truncated inverse of $\mathbf{h}$. In this case, all the symbols in the block $\mathbf{s}_k^{\text{block}}$ will be recovered without any arbitrary permutation. With this model, we are able to recover the source symbols, but will not obtain the impulse response of the equalizer.

3.2.2 Constrained ICA Algorithms

We have seen from the above analysis that with the zero-padding block transmission scheme, the BE problem becomes a standard BSS problem and can thus be solved with an ICA-based algorithm. Since we know that the "separating" matrix should have Toeplitz structure, this constraint can be enforced during iterations.

Forcing the Toeplitz structure on $\tilde{\mathbf{W}}$ will result in a $\tilde{\mathbf{C}}$ matrix that is also square Toeplitz with lower diagonal elements zero, but due to imperfect convergence the upper rows of the final $\tilde{\mathbf{C}}$ will generally contain small non-zero off-diagonal elements. As a result, the first several transmitted symbols, which correspond to the elements in the bottom of $\mathbf{s}_k^{\text{block}}$, will be better recovered than the last ones. The effect of imperfect convergence gets more severe when Q is large compared with L. In this case, to limit the number of nonzero elements in the top rows of $\tilde{\mathbf{W}}$, we can add a "length" constraint on $\tilde{\mathbf{w}}$ by forcing the uppermost diagonals of $\tilde{\mathbf{W}}$ to be zero. By doing so, we are adding a constraint that the coefficients of $\tilde{\mathbf{w}}$ be nonzero only up to a particular length. As a result, the number of possible non-zero elements in each row of matrix $\tilde{\mathbf{C}}$ will also be limited.

One general conclusion so far is that we can impose a Toeplitz structure constraint and length constraint on $\tilde{\mathbf{W}}$ in using a BSS adaptation scheme, with the expectation that it will allow improved performance. In fact, the performance of the constrained ICA-based algorithms depends on the characteristics of the channels. For minimum phase channel with the first tap having the largest magnitude, if we start from a good initialization it will not be difficult for the global system to converge to the identity; however, if the channel is

non-minimum phase or minimum phase with the largest tap behind the first tap, the inverse of $\mathbf{H}_T$ may have very large coefficients, which is hard for the algorithm to converge to. We will see examples based on simulations in Section 3.6.

We will next use the EASI algorithm explained in Chapter 2 as an example to show our idea of forcing Toeplitz structure and length constraint. Recall that with the EASI algorithm, the separating matrix has the adaptation as follows [8]:

$$\tilde{\mathbf{W}}_{k+1} = \tilde{\mathbf{W}}_k - \lambda\left[\mathbf{y}_k^{\text{Zblock}}(\mathbf{y}_k^{\text{Zblock}})^H - \mathbf{I} + \mathbf{g}(\mathbf{y}_k^{\text{Zblock}})(\mathbf{y}_k^{\text{Zblock}})^H - \mathbf{y}_k^{\text{Zblock}}\mathbf{g}(\mathbf{y}_k^{\text{Zblock}})^H\right]\tilde{\mathbf{W}}_k, \quad (3.7)$$

where $\mathbf{y}_k^{\text{Zblock}} = \tilde{\mathbf{W}}_k\mathbf{x}_k^{\text{Zblock}}$ contains the separated symbols in the k-th block. Denoting the relative change in the brackets as $\mathbf{U}_k$, the adaptation can be written as

$$\tilde{\mathbf{W}}_{k+1} = \tilde{\mathbf{W}}_k - \lambda\mathbf{U}_k\tilde{\mathbf{W}}_k. \quad (3.8)$$

There are two ways to enforce the Toeplitz constraint: one on the relative change matrix $\mathbf{U}_k$, and the other on the whole perturbation $\mathbf{U}_k\tilde{\mathbf{W}}_k$.

Constraint on relative change

Since the multiplication operation is closed in the space of Toeplitz matrices with lower elements zero, if we enforce Toeplitz structure on $\mathbf{U}_k$, the perturbation $\mathbf{U}_k\tilde{\mathbf{W}}_k$ as a whole will still have the Toeplitz structure. With the Toeplitz constraint on the relative change, the adaptation can be written as

$$\tilde{\mathbf{W}}_{k+1} = \tilde{\mathbf{W}}_k - \lambda\, Toeplitz\{\mathbf{U}_k\}\tilde{\mathbf{W}}_k, \quad (3.9)$$

where $Toeplitz\{\bullet\}$ means that the Toeplitz structure is enforced on $\mathbf{U}_k$ by taking averages along descending diagonals after forcing the lower left part to be zero. In fact, this Toeplitz

structure constraint can be proved to be the *orthogonal projection* of any square matrix onto the space of Toeplitz matrices with lower diagonals zero. In Chapter 4, the constraint will be enforced on non-square matrices. The proof of this orthogonal projection property will be given in the appendix of Chapter 4, and it can apply in a similar way for the square matrix case here.

From the adaptation in (3.9), we see that it has the form of "serial updating" introduced in Section 2.2.3. As a result, if the Toeplitz structure constraint is enforced on the relative change, the adaptation has the property of *equivariance*. In other words, the updating process does not depend on $\mathbf{H}_T$ as long as the global system $\tilde{\mathbf{C}} = \tilde{\mathbf{W}}\mathbf{H}_T$ is the same.

For the length constraint, it is not possible to enforce the constraint on matrix $\mathbf{U}_k$ and keep the number of non-zero elements unchanged in the first several rows of $\tilde{\mathbf{W}}$. The first row of $Toeplitz\{\mathbf{U}_k\}\tilde{\mathbf{W}}_k$ gives the truncated version of the convolution of $\tilde{\mathbf{w}}_k$ and the first row of $Toeplitz\{\mathbf{U}_k\}$. As a result, as long as $Toeplitz\{\mathbf{U}_k\}$ is not identity, $\tilde{\mathbf{w}}$ cannot keep the number of nonzero elements in its tails unchanged from the previous iteration. However, the length constraint can be added after the matrix multiplication, i.e. on $Toeplitz\{\mathbf{U}_k\}\tilde{\mathbf{W}}_k$, which leads to the T-LC-EASI algorithm. The T-LC-EASI algorithm does not have the equivariance property.

Constraint on whole perturbation

The Toeplitz constraint can also be enforced on $\mathbf{U}_k\tilde{\mathbf{W}}_k$, with the adaptation

$$\tilde{\mathbf{W}}_{k+1} = \tilde{\mathbf{W}}_k - \lambda\, Toeplitz\{\mathbf{U}_k\tilde{\mathbf{W}}_k\}. \tag{3.10}$$

However, the adaptation in (3.10) does not have the serial updating form, and thus is not equivariant. In addition to the Toeplitz constraint, the length constraint can be enforced on the term $\mathbf{U}_k\tilde{\mathbf{W}}_k$ after matrix multiplication.

3.3 Block Transmission with Cyclic Prefix

3.3.1 Formulation

Another block transmission scheme that is widely used is to pad the cyclic-prefix (CP) between transmitted blocks. This transmission scheme has similarity to the latest orthogonal frequency division multiplexing (OFDM) technique except that the block of symbols are not modulated by *multiple* subcarriers before adding the CP, and is called single-carrier (SC) modulation. The SC modulation combined with frequency domain equalization (FDE) has attracted wide interest in broadband communication systems [1]–[3]. Compared with OFDM, the SC-FDE has comparable performance with the same overall complexity, while it can overcome the drawbacks OFDM suffers: high peak to average power ratio, intolerance to amplifier nonlinearities, and sensitivity to carrier frequency offsets. In this part we will study time domain equalization for this signal carrier case with CP, for comparison with the zero padding scheme.

The source symbols without the CP are transmitted in blocks of size Q with a cyclic prefix extension, and the length of the CP is equal to the order of the channel L. As in the zero-padded block transmission case, for this block transmission scheme with the CP, it is also assumed that $Q > L$.

Denote the k-th transmission block as $\mathbf{s}_k^{\text{CPblock}} = [(\mathbf{s}_k^{\text{block}})^T, (\mathbf{s}_k^{\text{CP}})^T]^T$, where $\mathbf{s}_k^{\text{block}} = [s_k(Q), s_k(Q-1), ..., s_k(1)]^T$ are the Q transmitted symbols, and the CP is the same as the first L symbols in $\mathbf{s}_k^{\text{block}}$, i.e. $\mathbf{s}_k^{\text{CP}} = [s_k(Q), ..., s_k(Q-L+1)]^T$. The transmission scheme is illustrated in Fig. 3.1. With the corresponding noise vector $\mathbf{v}_k^{\text{CPblock}}$, the channel outputs can be expressed as

$$\mathbf{x}_k^{\text{CPblock}} = \mathbf{H}\mathbf{s}_k^{\text{CPblock}} + \mathbf{v}_k^{\text{CPblock}}. \tag{3.11}$$

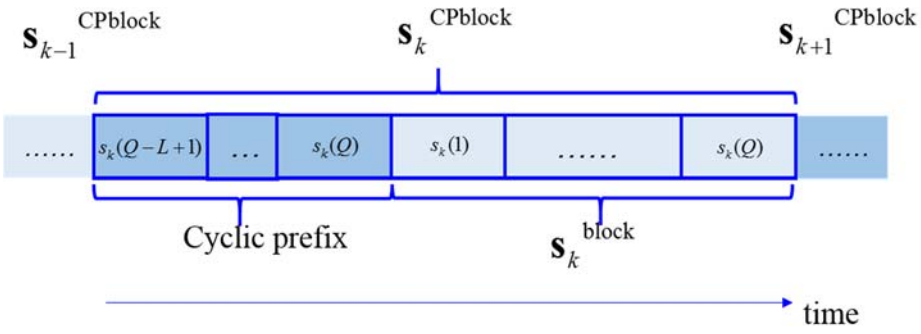

Fig. 3.1 Block transmission with cyclic prefix.

Since the last several elements of $\mathbf{s}_k^{\text{CPblock}}$ are the CP, which are copies of the elements in $\mathbf{s}_k^{\text{block}}$, equation (3.11) reduces to

$$\mathbf{x}_k^{\text{CPblock}} = \mathbf{H}_C\mathbf{s}_k^{\text{block}} + \mathbf{v}_k^{\text{CPblock}}, \tag{3.12}$$

where $\mathbf{H}_C$ is square *circulant* matrix. Matrix $\mathbf{H}_C$ is the modified version of the first Q columns of matrix $\mathbf{H}$, where the last L columns of $\mathbf{H}$ is added onto the first L columns of $\mathbf{H}$, i.e.

$$\mathbf{H}_C = \begin{pmatrix} h(0) & h(1) & \cdots & h(L) & 0 & \cdots & 0 \\ & \ddots & & \cdots & \ddots & & \\ \cdots & h(0) & h(1) & & \cdots & h(L) & \\ & & \ddots & & \cdots & \cdots & \\ h(L) & \cdots & h(0) & & \cdots & h(L-1) & \\ & \ddots & & \cdots & \ddots & \cdots & \\ h(1) & \ddots & h(L) & 0 & \cdots & 0 & h(0) \end{pmatrix}. \tag{3.13}$$

With the formulation in (3.12), our goal is to find a matrix $\tilde{\mathbf{W}}$ with size $Q \times Q$, such that $\tilde{\mathbf{W}}\mathbf{H}_C \simeq \mathbf{I}_C$. Here $\mathbf{I}_C$ is defined as a circularly shifted version of identity matrix $\mathbf{I}$. Different from the zero-padding case where the perfect global system is considered to be the identity, in this block transmission scheme with the CP, the ideal global system includes all the circularly shifted versions of the identity, i.e. $\mathbf{I}_C$; this will make the analysis more straightforward due to the properties of the circulant matrix. With the global system $\mathbf{I}_C$, the symbols in a certain block will be recovered subject to circular permutation. We need to resolve this circular permutation with other techniques; this should not be difficult and is not addressed here.

Before going to the iterative algorithm, let us first take a look at the nature of the inverse of $\mathbf{H}_C$. Since $\mathbf{H}_C$ is a circulant matrix, from the property of circulant matrices, we know that its inverse should also be circulant [9]. Let $\mathbf{h}_C$ be the first row of matrix $\mathbf{H}_C$, which is a length-Q vector that extends $\mathbf{h}$ by padding zeros at the end. Denote $\mathbf{W}_C = \mathbf{H}_C^{-1}$, and let vector $\mathbf{w}_C^T$ with length Q be the first row of $\mathbf{W}_C$, then matrix $\mathbf{W}_C$ contains circularly shifted versions of $\mathbf{w}_C^T$ in each row. Multiplication of two circulant matrices actually gives the circular convolution of the associated vectors, thus

$$\mathbf{w}_C \otimes \mathbf{h}_C = [1, 0, ..., 0]^T . \tag{3.14}$$

Denote the Q point DFT of $\mathbf{w}_C$ as $\mathbf{w}_C{}^{\mathrm{DFT}}$, and that of the channel response $\mathbf{h}_C$ as $\mathbf{h}_C{}^{\mathrm{DFT}}$, then

$$h_C{}^{\mathrm{DFT}}(k) w_C{}^{\mathrm{DFT}}(k) = 1, \; k = 1, 2, ..Q . \tag{3.15}$$

Thus the first row of $\mathbf{W}_C$ is a vector such that the element-wise multiplication of its Q-point DFT and the Q-point DFT of $\mathbf{h}_C$ gives a sequence of ones. Let $\tilde{\mathbf{w}}^T$ be the first row of matrix $\tilde{\mathbf{W}}$. If matrix $\tilde{\mathbf{W}}$ contains vector $\mathbf{w}_C{}^T$ is its first row, i.e. $\tilde{\mathbf{w}} = \mathbf{w}_C$, then $\tilde{\mathbf{W}}$ is exactly the inverse of $\mathbf{H}_C$, i.e. $\tilde{\mathbf{W}} = \mathbf{W}_C$. Since we allow $\tilde{\mathbf{W}}\mathbf{H}_C \simeq \mathbf{I}_C$, $\tilde{\mathbf{w}}$ can be any circularly shifted version of $\mathbf{w}_C$.

Now we examine the relation between $\mathbf{w}_C$ and the equalizer response. For simplicity, we only consider the case $\tilde{\mathbf{W}}\mathbf{H}_C \simeq \mathbf{I}$ with no circular shifting, i.e. $\tilde{\mathbf{w}} = \mathbf{w}_C$. Recall that the length of $\mathbf{h}$ is $L+1$, so the linear convolution of $\tilde{\mathbf{w}}$ and $\mathbf{h}$ should be of length $Q+L$, and the Q-point circular convolution of $\tilde{\mathbf{w}}$ and $\mathbf{h}_C$ is the first Q points of $\tilde{\mathbf{w}} * \mathbf{h}$ with aliasing from the last L points of $\tilde{\mathbf{w}} * \mathbf{h}$. When Q is large enough, $\tilde{\mathbf{w}}$ is likely to be the response of a good equalizer; but this is not necessary. In general, as long as $\tilde{\mathbf{w}}$ has DFT that satisfies the condition (3.15) considered above , $\tilde{\mathbf{W}}\mathbf{H}_C \simeq \mathbf{I}$ can be obtained.

3.3.2 Constrained ICA Algorithms

We know that ideally $\tilde{\mathbf{W}}$ should be a circulant matrix, and contains circularly shifted versions of the same vector in each row. For this circulant matrix case, we can enforce the

circulant structure on $\tilde{\mathbf{W}}$ in an iterative scheme by taking averages of the corresponding elements, as shown in Fig. 3.2. This can be shown to be the orthogonal projection of a square matrix onto the space of circulant matrices. The proof is similar to the one given in the appendix of Chapter 4, and will be omitted here.

$$\tilde{\mathbf{W}}_k = \begin{pmatrix} (\tilde{W}_k)_{11} & (\tilde{W}_k)_{12} & \cdots & (\tilde{W}_k)_{1Q} \\ (\tilde{W}_k)_{21} & (\tilde{W}_k)_{22} & \cdots & (\tilde{W}_k)_{2Q} \\ \cdots & \ddots & \cdots & \\ \cdots & (\tilde{W}_k)_{i,i-1} & (\tilde{W}_k)_{ii} \cdots & (\tilde{W}_k)_{iQ} \\ \cdots & \ddots & \ddots & \\ (\tilde{W}_k)_{Q-1,1} & \cdots & \cdots & (\tilde{W}_k)_{Q-1,Q} \\ (\tilde{W}_k)_{Q1} & (\tilde{W}_k)_{Q2} \cdots & (\tilde{W}_k)_{Q,Q-1} & (\tilde{W}_k)_{QQ} \end{pmatrix}$$

Fig. 3.2 Circulant structure constraint by taking averages.

As in the Toeplitz case, we will use the EASI algorithm to illustrate our idea of the structure constraint in adaptation. Since multiplication is closed in the space of circulant matrices, we can consider forcing the circulant structure either on the relative change or on the whole perturbation. However, for circulant matrices, it can be shown that if $\mathbf{Y}$ is a circulant matrix, then

$$Circulant\{\mathbf{X}\}\mathbf{Y} = Circulant\{\mathbf{XY}\}. \tag{3.16}$$

The detailed proof of equation (3.16) will be given in Appendix 3A.

For the EASI algorithm, denote the k-th output block as $\mathbf{y}_k^{\text{CPblock}} = \tilde{\mathbf{W}}_k \mathbf{x}_k^{\text{CPblock}}$ and let $\mathbf{U}_k = \mathbf{y}_k^{\text{CPblock}}(\mathbf{y}_k^{\text{CPblock}})^H - \mathbf{I} + \mathbf{g}(\mathbf{y}_k^{\text{CPblock}})(\mathbf{y}_k^{\text{CPblock}})^H - \mathbf{y}_k^{\text{CPblock}}\mathbf{g}(\mathbf{y}_k^{\text{CPblock}})^H$, then the EASI algorithm has the adaptation for matrix $\tilde{\mathbf{W}}$

$$\tilde{\mathbf{W}}_{k+1} = \tilde{\mathbf{W}}_k - \lambda \mathbf{U}_k \tilde{\mathbf{W}}_k \tag{3.17}$$

According to (3.16), forcing the circulant constraint on the relative change $\mathbf{U}_k$ and then doing matrix multiplication is the same as forcing the constraint on $\mathbf{U}_k \tilde{\mathbf{W}}_k$. As a result, the following two adaptations are equivalent:

$$\tilde{\mathbf{W}}_{k+1} = \tilde{\mathbf{W}}_k - \lambda Circulant\{\mathbf{U}_k\} \tilde{\mathbf{W}}_k \ , \tag{3.18}$$

$$\tilde{\mathbf{W}}_{k+1} = \tilde{\mathbf{W}}_k - \lambda Circulant\{\mathbf{U}_k \tilde{\mathbf{W}}_k\} \ . \tag{3.19}$$

For (3.18), the adaptation is serial updating, and thus has the equivariance property. Although (3.19) does not have the form of serial updating, it is equivalent to (3.18) and thus also has the property of equivariance. The resulting algorithm can be called as the C-EASI algorithm.

For the C-EASI algorithm, the number of non-zero elements in each row is the same. In correspondence to the T-EASI algorithm, we can still enforce a length constraint (LC) on the vector $\tilde{\mathbf{w}}$ contained in each row of $\tilde{\mathbf{W}}$, so that $\tilde{\mathbf{w}}$ can have non-zero elements up to a certain length. For the same reason as explained for the T-EASI case, the length constraint cannot be enforced on the relative change, but on the whole perturbation after matrix multiplication, i.e. $Circulant\{\mathbf{U}_k\} \tilde{\mathbf{W}}_k$. We will call the algorithm with length constraint the C-LC-EASI algorithm. With the length constraint after the matrix multiplication, the property of equivariance will not be preserved.

In fact, the length constraint on $\tilde{\mathbf{w}}$ for the C-EASI algorithm does not make as much sense as for the T-EASI algorithm. On the one hand, since circulant matrix has circulantly shifted vector in each row, all the symbols in the block can be recovered to the same degree with no bias; on the other hand, for circulant case, we allow the final global system to be a

circulant version of the identity, so it is hard to tell whether $\tilde{\mathbf{w}}$, the first row of $\tilde{\mathbf{W}}$, is exactly the vector we want to enforce length constraint on or its circularly shifted version. If we can resolve the circular shift ambiguity, and figure out the location of the vector that satisfies the DFT condition in (3.15), we will be able to enforce a more reasonable length constraint by setting the elements outside the vector to zero. We will see from later simulations that the effect of the length constraint varies from case to case.

3.4 Simplified Vector Updating and Computational Complexity

For both of the block transmission schemes, we end up with a matrix adaptation with a Toeplitz or circulant matrix constraint. Our development in the previous sections resulted in iterations for the "separating" or inverse matrix $\tilde{\mathbf{W}}$. However, since the "separating" matrix contains repeated elements in each row vector, we can seek equivalent vector iterations to make the computations more efficient. In this section, we show how the matrix adaptations can be converted to vector adaptations. For simplicity we focus on the T-EASI and C-EASI algorithm without the length constraint. In fact, with the equivalent form for vector adaptation, we can always enforce the length constraint on the vector by forcing a fixed number of elements in the last part of the vector to be zero at the end of each iteration.

3.4.1 T-EASI

For the T-EASI algorithm, there are two versions with the Toeplitz constraint, as follows:

$$\tilde{\mathbf{W}}_{k+1} = \tilde{\mathbf{W}}_k - \lambda\, Toeplitz\{\mathbf{U}_k\}\tilde{\mathbf{W}}_k\,, \tag{3.20}$$

$$\tilde{\mathbf{W}}_{k+1} = \tilde{\mathbf{W}}_k - \lambda\, Toeplitz\{\mathbf{U}_k\tilde{\mathbf{W}}_k\}\,. \tag{3.21}$$

where $\mathbf{U}_k = \mathbf{y}_k\mathbf{y}_k^{\,H} - \mathbf{I} + \mathbf{g}(\mathbf{y}_k)\mathbf{y}_k^{\,H} - \mathbf{y}_k\mathbf{g}(\mathbf{y}_k)^H$ with the superscript "Zblock" of $\mathbf{y}_k$ omitted.

In (3.20), with the matrix $\mathbf{U}_k$ forced to be Toeplitz, the multiplication of two Toeplitz matrices $Toeplitz\{\mathbf{U}_k\}$ and $\tilde{\mathbf{W}}_k$ gives a new Toeplitz matrix. Denote the first row of $Toeplitz\{\mathbf{U}_k\}$ as $\mathbf{u}_k^{\,T}$, and the first row of $\tilde{\mathbf{W}}_k$ as $\tilde{\mathbf{w}}_k^{\,T}$, then the first row of $Toeplitz\{\mathbf{U}_k\}\tilde{\mathbf{W}}_k$ will contain the first Q elements of the convolution of $\mathbf{u}_k$ and $\tilde{\mathbf{w}}_k$. As a result, it is easy to see that the matrix adaptation in (3.20) can be written equivalently as the vector adaptation

$$\tilde{\mathbf{w}}_{k+1} = \tilde{\mathbf{w}}_k - \lambda\{\tilde{\mathbf{w}}_k * \mathbf{u}_k\}_{1:Q}\,. \tag{3.22}$$

Since linear convolution can be implemented with the FFT by adding zeros to the vector, the order of computational complexity using (3.22) can be reduced from $O(Q^3)$ to $O(Q\log Q)$ with efficient implementation. In Appendix 3B, the exact number of the additions and the multiplications are listed, with details omitted. In Chapter 4, the same idea will be applied to simplify the matrix adaptation containing the equalizer vector with more detailed explanation.

For the second version of the matrix adaptation (3.21), the equivalent vector adaptation has the form

$$\tilde{\mathbf{w}}_{k+1} = \tilde{\mathbf{w}}_k - \lambda\boldsymbol{\Gamma}_k\tilde{\mathbf{w}}_k\,, \tag{3.23}$$

where $\mathbf{\Gamma}_k$ contains cross-correlation terms from matrix $\mathbf{U}_k$. However, the relation between $\mathbf{\Gamma}_k$ and $\mathbf{U}_k$ is not as straightforward as in the first case. The following is an example of how matrix $\mathbf{\Gamma}_k$ can be obtained from $\mathbf{U}_k$.

Suppose $Q=3$ and $\mathbf{U}_k$ is expressed as

$$\mathbf{U}_k = \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ U_{21} & U_{22} & U_{23} \\ U_{31} & U_{32} & U_{33} \end{bmatrix}, \tag{3.24}$$

then $\mathbf{\Gamma}_k$ can be calculated to be

$$\mathbf{\Gamma}_k = \begin{bmatrix} \frac{U_{11}+U_{22}+U_{33}}{3} & \frac{U_{32}+U_{21}}{3} & \frac{U_{31}}{3} \\ \frac{U_{12}+U_{23}}{2} & \frac{U_{11}+U_{22}}{2} & \frac{U_{21}}{2} \\ U_{13} & U_{12} & U_{11} \end{bmatrix}. \tag{3.25}$$

From extensive simulations we found that the performance of the two versions (3.20) and (3.21) of the T-EASI algorithm is almost the same. As a result, in the following analysis and simulations, the T-EASI algorithm we refer to is the first version (3.20) of the T-EASI with Toeplitz constraint on the relative change, since it has the nice property of equivariance and can be implemented computationally efficiently via FFT.

3.4.2 C-EASI

Unlike the T-EASI case, forcing circulant structure on either the relative change or the whole perturbation leads to equivalent algorithms. With the circulant structure constraint, the C-EASI algorithm has the iterative updating

$$\tilde{\mathbf{W}}_{k+1} = \tilde{\mathbf{W}}_k - \lambda Circulant\{\mathbf{U}_k\}\tilde{\mathbf{W}}_k, \tag{3.26}$$

where $\mathbf{U}_k = \mathbf{y}_k \mathbf{y}_k^H - \mathbf{I} + \mathbf{g}(\mathbf{y}_k)\mathbf{y}_k^H - \mathbf{y}_k \mathbf{g}(\mathbf{y}_k)^H$ with the superscript "CPblock" of $\mathbf{y}_k$ omitted.

In (3.26), $Circulant\{\mathbf{U}_k\}\tilde{\mathbf{W}}_k$ will give a new circulant matrix. If we denote the first row of $Circulant\{\mathbf{U}_k\}$ as $\mathbf{u}_k$, and the first row of $\tilde{\mathbf{W}}_k$ as $\tilde{\mathbf{w}}_k$, according to the definition of circular convolution, it can be seen that the first row of $Circulant\{\mathbf{U}_k\}\tilde{\mathbf{W}}_k$ will be exactly the circular convolution of $\mathbf{u}_k$ and $\tilde{\mathbf{w}}_k$, i.e. $\tilde{\mathbf{w}}_k \otimes \mathbf{u}_k$, where $\otimes$ means the circular convolution of two vectors. Here we are using the same notation $\mathbf{u}_k$ and $\tilde{\mathbf{w}}_k$ for the two different schemes, but this will be clarified where necessary to avoid confusion. Based on the property of circulant matrices, it follows that the equivalent vector adaptation for $\tilde{\mathbf{w}}_k$ can be written as

$$\tilde{\mathbf{w}}_{k+1} = \tilde{\mathbf{w}}_k - \lambda \tilde{\mathbf{w}}_k \otimes \mathbf{u}_k. \tag{3.27}$$

Circular convolution can be efficiently implemented with fast Fourier transform, thus the computational cost of implementing (3.27) can be reduced to $O(Q \log Q)$. The number of additions and multiplications is given in Appendix 3B.

3.5 I/Q Independence

When the source has independent in-phase and quadrature parts, as in the case of standard QAM signaling, Q complex source symbols can be seen as $2Q$ mutually independent real symbols. In this case, if the sources are well separated, the $2Q$ real output symbols of the

equalizer should be independent of each other, and the original symbols can be recovered with no phase ambiguity by proper I/Q association [10], [11].

In this section, we will see how the I/Q independence constraint can be incorporated to reduce phase ambiguity in our T-EASI and C-EASI schemes. Since a circulant matrix can be considered to be a special case of Toeplitz matrices, we will use the block transmission scheme with zero padding to explain the idea.

Recall that with zeros padded between transmitted blocks, the BE problem can be formulated in matrix form as

$$\mathbf{x}_k^{\text{Zblock}} = \mathbf{H}_T \mathbf{s}_k^{\text{block}} + \mathbf{v}_k^{\text{Zblock}}. \tag{3.28}$$

For simplicity of analysis, we ignore noise in (3.28) and drop the block-index k and the superscripts "Zblock" and "block". Denoting the in-phase and quadrature parts of the channel matrix $\mathbf{H}_T$ as $(\mathbf{H}_T)_R$ and $(\mathbf{H}_T)_I$, and those of the signal vector $\mathbf{s}$ as $\mathbf{s}_R$ and $\mathbf{s}_I$, the in-phase and quadrature components of the observation vector become

$$\mathbf{x}_R = (\mathbf{H}_T)_R \mathbf{s}_R - (\mathbf{H}_T)_I \mathbf{s}_I \tag{3.29}$$

$$\mathbf{x}_I = (\mathbf{H}_T)_R \mathbf{s}_I + (\mathbf{H}_T)_I \mathbf{s}_R \tag{3.30}$$

Writing (3.29) and (3.30) in a matrix, we have

$$\begin{bmatrix} \mathbf{x}_R \\ \mathbf{x}_I \end{bmatrix} = \begin{bmatrix} (\mathbf{H}_T)_R & -(\mathbf{H}_T)_I \\ (\mathbf{H}_T)_I & (\mathbf{H}_T)_R \end{bmatrix} \begin{bmatrix} \mathbf{s}_R \\ \mathbf{s}_I \end{bmatrix}. \tag{3.31}$$

The separated output $\mathbf{y}_k = \tilde{\mathbf{W}}_k \mathbf{x}_k$ can also be written in terms of their respective in-phase and quadrature components, i.e.

$$\begin{bmatrix} \mathbf{y}_R \\ \mathbf{y}_I \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{W}}_R & -\tilde{\mathbf{W}}_I \\ \tilde{\mathbf{W}}_I & \tilde{\mathbf{W}}_R \end{bmatrix} \begin{bmatrix} \mathbf{x}_R \\ \mathbf{x}_I \end{bmatrix}. \tag{3.32}$$

Defining $\overline{\mathbf{H}}_T = \begin{bmatrix} (\mathbf{H}_T)_R & -(\mathbf{H}_T)_I \\ (\mathbf{H}_T)_I & (\mathbf{H}_T)_R \end{bmatrix}$, $\overline{\tilde{\mathbf{W}}} = \begin{bmatrix} \tilde{\mathbf{W}}_R & -\tilde{\mathbf{W}}_I \\ \tilde{\mathbf{W}}_I & \tilde{\mathbf{W}}_R \end{bmatrix}$, $\overline{\mathbf{x}} = \begin{bmatrix} \mathbf{x}_R \\ \mathbf{x}_I \end{bmatrix}$ and $\overline{\mathbf{y}} = \begin{bmatrix} \mathbf{y}_R \\ \mathbf{y}_I \end{bmatrix}$, we have

$$\overline{\tilde{\mathbf{W}}}\overline{\mathbf{H}}_T = \begin{bmatrix} \tilde{\mathbf{W}}_R(\mathbf{H}_T)_R - \tilde{\mathbf{W}}_I(\mathbf{H}_T)_I & -\tilde{\mathbf{W}}_R(\mathbf{H}_T)_I - \tilde{\mathbf{W}}_I(\mathbf{H}_T)_R \\ \tilde{\mathbf{W}}_R(\mathbf{H}_T)_I + \tilde{\mathbf{W}}_I(\mathbf{H}_T)_R & \tilde{\mathbf{W}}_R(\mathbf{H}_T)_R - \tilde{\mathbf{W}}_I(\mathbf{H}_T)_I \end{bmatrix}, \tag{3.33}$$

$$\overline{\mathbf{y}} = \overline{\tilde{\mathbf{W}}}\overline{\mathbf{x}}. \tag{3.34}$$

The original problem becomes: given the observation vector $\overline{\mathbf{x}}$, find the separating matrix $\overline{\tilde{\mathbf{W}}}$ such that $\overline{\mathbf{y}}$ is a good estimate of $\begin{bmatrix} \mathbf{s}_R^T & \mathbf{s}_I^T \end{bmatrix}^T$, i.e. $\overline{\tilde{\mathbf{W}}}\overline{\mathbf{H}}_T = \mathbf{I}$.

The matrix multiplication $\overline{\tilde{\mathbf{W}}}\overline{\mathbf{H}}_T$ consists of four $Q \times Q$ blocks, of which the diagonal blocks are the same and the other two sum to zero. Thus ideally we want

$$\tilde{\mathbf{W}}_R(\mathbf{H}_T)_R - \tilde{\mathbf{W}}_I(\mathbf{H}_T)_I = \mathbf{I}_Q, \tag{3.35}$$

$$\tilde{\mathbf{W}}_R(\mathbf{H}_T)_I + \tilde{\mathbf{W}}_I(\mathbf{H}_T)_R = \mathbf{0}_Q. \tag{3.36}$$

The ideal solutions of $\tilde{\mathbf{W}}_R$ and $\tilde{\mathbf{W}}_I$ satisfying (3.35) and (3.36) are

$$\tilde{\mathbf{W}}_R^{ideal} = (\mathbf{H}_T)_I^{-1}((\mathbf{H}_T)_R(\mathbf{H}_T)_I^{-1} + (\mathbf{H}_T)_I(\mathbf{H}_T)_R^{-1})^{-1}, \tag{3.37}$$

$$\tilde{\mathbf{W}}_I^{ideal} = -(\mathbf{H}_T)_I^{-1}((\mathbf{H}_T)_R(\mathbf{H}_T)_I^{-1} + (\mathbf{H}_T)_I(\mathbf{H}_T)_R^{-1})^{-1}(\mathbf{H}_T)_I(\mathbf{H}_T)_R^{-1}. \tag{3.38}$$

Since $(\mathbf{H}_T)_R$ and $(\mathbf{H}_T)_I$ are both square Toeplitz matrices with lower diagonal elements zero, their inverses have the same structure. In addition, since the Toeplitz structure remains under matrix multiplication, $\tilde{\mathbf{W}}_R^{ideal}$ and $\tilde{\mathbf{W}}_I^{ideal}$ are also square Toeplitz matrices with lower diagonal elements zero. We see that matrix $\overline{\tilde{\mathbf{W}}}$ has a resulting block structure constraint, with identical Toeplitz diagonal blocks and off-diagonal Toeplitz blocks that are sign-inverted versions of each other.

Using the EASI algorithm with the I/Q independence constraint, we can enforce the Toeplitz constraint either on the relative change or the whole perturbation without much difference in performance. To be consistent with the T-EASI algorithm, we enforce this constraint on the relative change in the simulation parts. In addition to the Toeplitz structure, the block structure of $\bar{\tilde{\mathbf{W}}}$ is also enforced.

The original EASI algorithm, T(C)-EASI and T(C)-LC-EASI can all be combined with the I/Q independence constraint, which gives the I/Q-EASI, I/Q-T(C)-EASI, and I/Q-T(C)-LC-EASI algorithms.

3.6 Simulations

In this section, we will give examples of the (I/Q)-T-(LC)-EASI and (I/Q)-C-(LC)-EASI algorithms for the two block transmission schemes. The performance of the two different schemes (zero-padding vs. cyclic prefix) will also be compared.

First consider a *minimum phase* channel with channel impulse response shown in Fig. 3.3. The channel has order $L=4$ and the SNR is 15dB. The first tap has the largest magnitude. From the zero-pole pattern of the channel in Fig. 3.4, it can be seen that all the zeros are near the unit circle, which makes it hard to equalize. A sequence of 64-QAM source symbols is transmitted through the channel in blocks of size $Q=30$, before which there are $L=4$ padded symbols being either zero or the cyclic prefix. In the algorithms, the nonlinear function is the phase preserving cubic, i.e. $g(x)=|x|^2 x$.

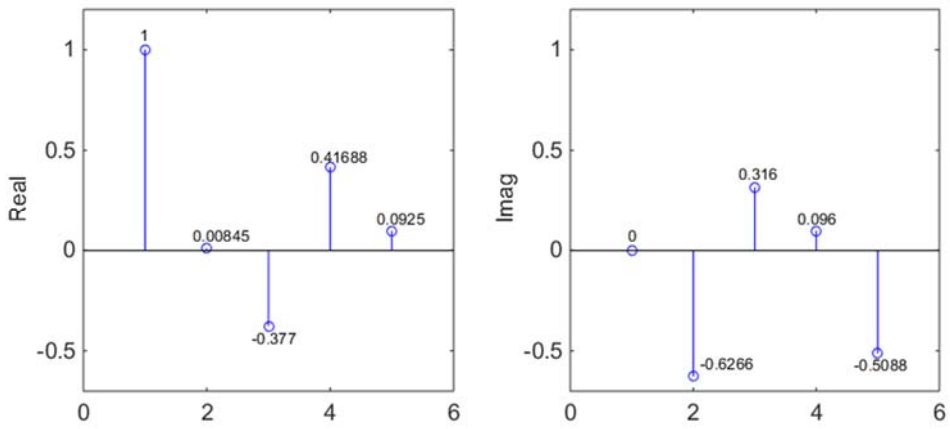

Fig. 3.3 Channel impulse response of short minimum phase channel.

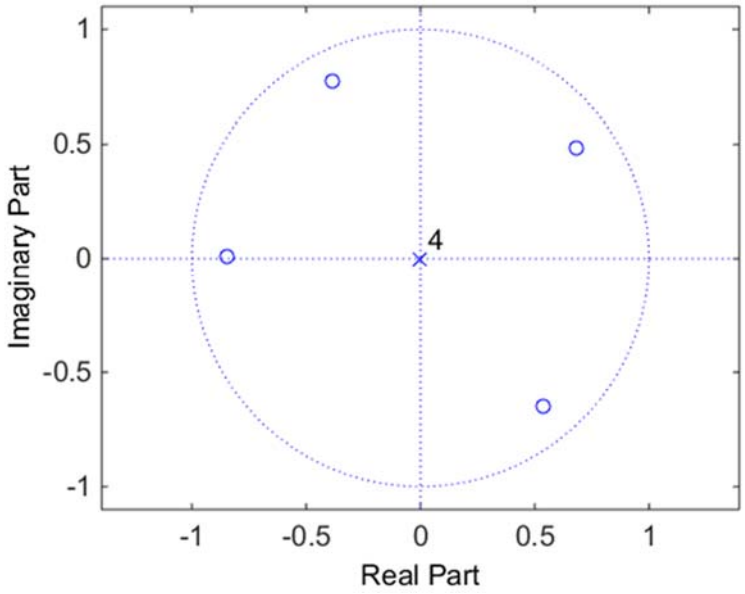

Fig. 3.4 Zero-pole pattern of the minimum phase channel.

First consider the block transmission scheme with zero padding. The performance of the EASI, T-EASI and the T-LC-EASI is compared in Fig. 3.5. For the T-LC-EASI algorithm, the length constraint is enforced so that the elements on the upper right corner of the $\tilde{\mathbf{W}}$ matrix are zero. Specifically, in the first row of $\tilde{\mathbf{W}}$ the first $M+1$ elements are

kept and the other elements are set to zero; the other rows are then shifted truncated versions of the first row. Based on our experiments, $M = 20$ is a reasonable choice.

For the case when the symbols are transmitted in blocks with the CP, the performance of the EASI, C-EASI, C-LC-EASI and their I/Q version is compared in Fig. 3.6. The parameter M is set to be the same as the last example, i.e. $M = 20$. For both of the two examples, the "separating" matrix is initialized with $\tilde{\mathbf{W}} = (1 + 0.5j)\mathbf{I}$.

In the experiment, we use the average inter-symbol interference (ISI) of the rows of the matrix $\tilde{\mathbf{W}}\mathbf{H}_T$ or $\tilde{\mathbf{W}}\mathbf{H}_C$ to measure the performance of separation, where the ISI for a vector $\mathbf{c}$ is defined as

$$\mathrm{ISI} = \sum_{i=1}^{Q} \frac{|c_i|^2}{\max_i |c_i|^2} - 1. \tag{3.39}$$

For the zero-padding case, when the T-EASI is applied, the ISI of each row is different because of the difference in the number of the off-diagonal elements of $\tilde{\mathbf{W}}\mathbf{H}_T$. As a result, the average ISI reflects an average degree to which the symbols in a block are recovered. For the CP case, the ISI of each row is the same and equal to the average ISI, thus picking any row gives the same result.

From Fig. 3.5, it can be seen that with the Toeplitz constraint, the performance is greatly improved compared to the EASI in terms of faster convergence and lower ISI at the steady state. With the length constraint, the convergence speed of the T-LC-EASI is slightly faster than the T-EASI. The algorithms with the I/Q version yield comparable performance with the ones with no I/Q independence constraint.

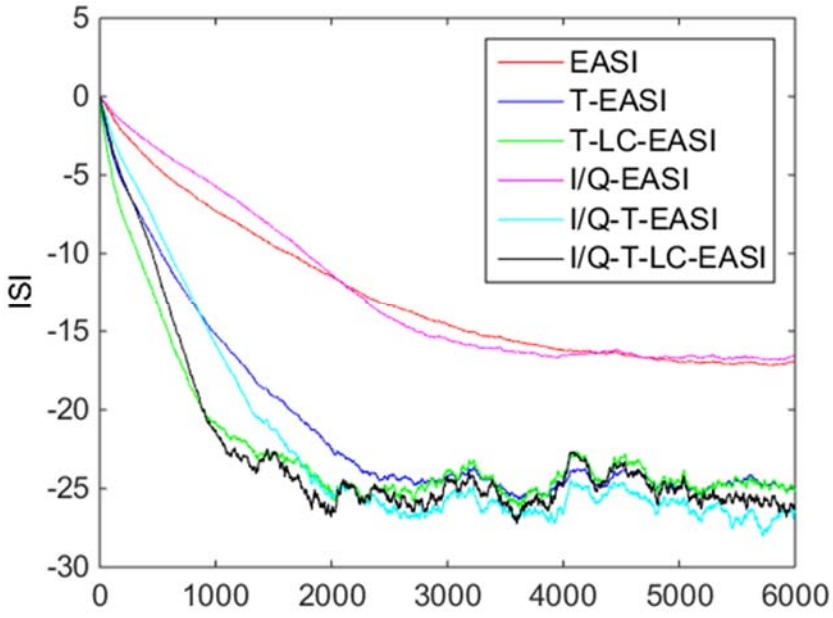

Fig. 3.5 Average ISI of each row of matrix $\tilde{\mathbf{W}}\mathbf{H}_T$ for the minimum phase system EASI, T-EASI and T-LC-EASI, initialization $\tilde{\mathbf{W}} = (1+0.5j)\mathbf{I}$.

In Fig. 3.6, the performance for the block transmission scheme with the CP is shown. The result is consistent with that of the scheme with zero padding. In this example, comparing the results of the C-EASI and C-LC-EASI, we can find that compared to the T-EASI case the length constraint helps increase convergence speed more. For the C-EASI algorithm, the I/Q independence constraint yields a little more apparent advantage compared with the T-EASI case I/Q version.

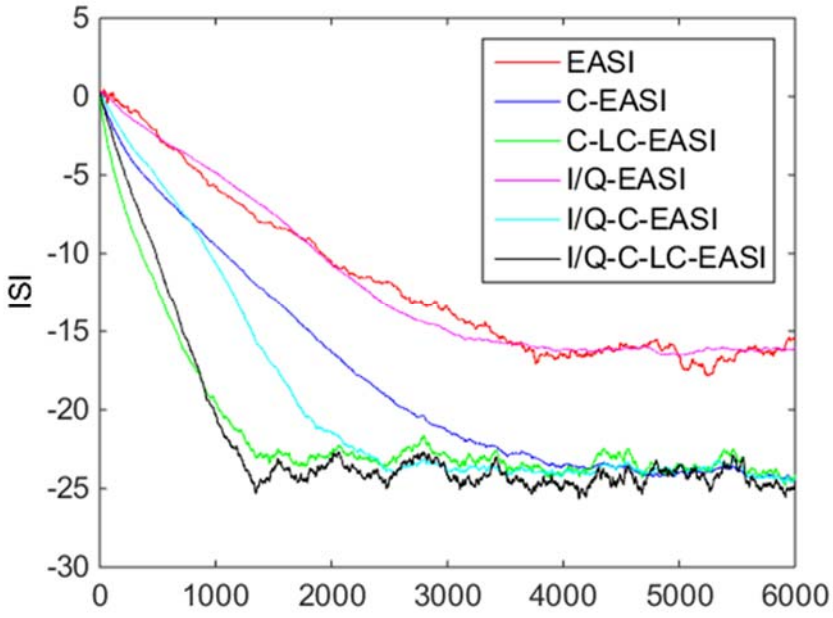

Fig. 3.6 ISI of the rows of matrix $\tilde{\mathbf{W}}\mathbf{H}_C$ for the minimum phase system with EASI, C-EASI and C-LC-EASI, initialization $\tilde{\mathbf{W}} = (1+0.5j)\mathbf{I}$.

In addition to the typical example of minimum phase channel as given above, we also examined our algorithms without the I/Q constraint on multiple channels with the Rician model. We randomly generated 10 minimum phase channels with length 4, i.e. there are four paths. Among the four discrete paths, the first one is Rician fading process with factor $K = 1$, while the others are Rayleigh fading processes. The average power gain of the four paths decreases in order. The parameters M and Q, and the initialization of $\tilde{\mathbf{W}}$ is set as above. From Fig. 3.7 and Fig. 3.8, it can be seen the average performance over multiple channel examples for different versions of the EASI algorithms is consistent with that shown in Fig. 3.5 and Fig. 3.6.

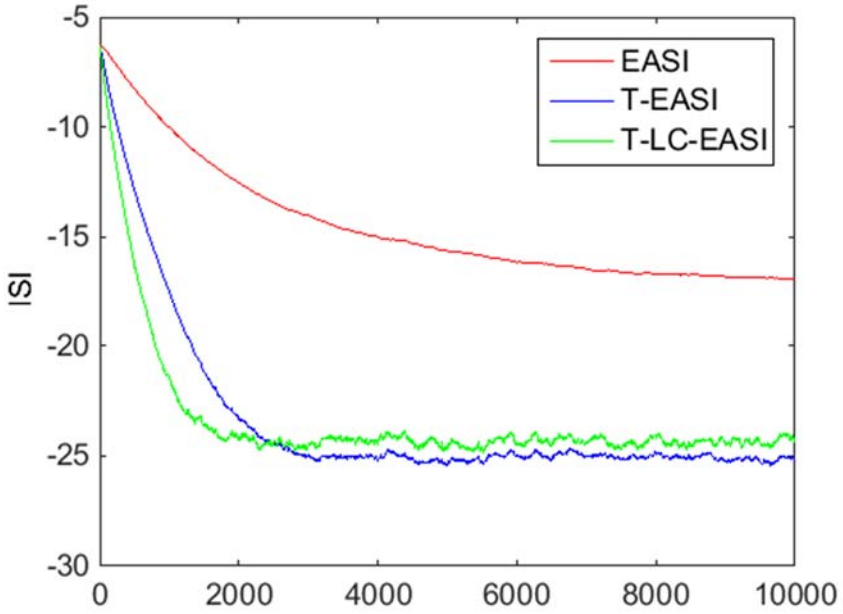

Fig. 3.7 Average ISI over multiple minimum phase channels with EASI, T-EASI and T-LC-EASI.

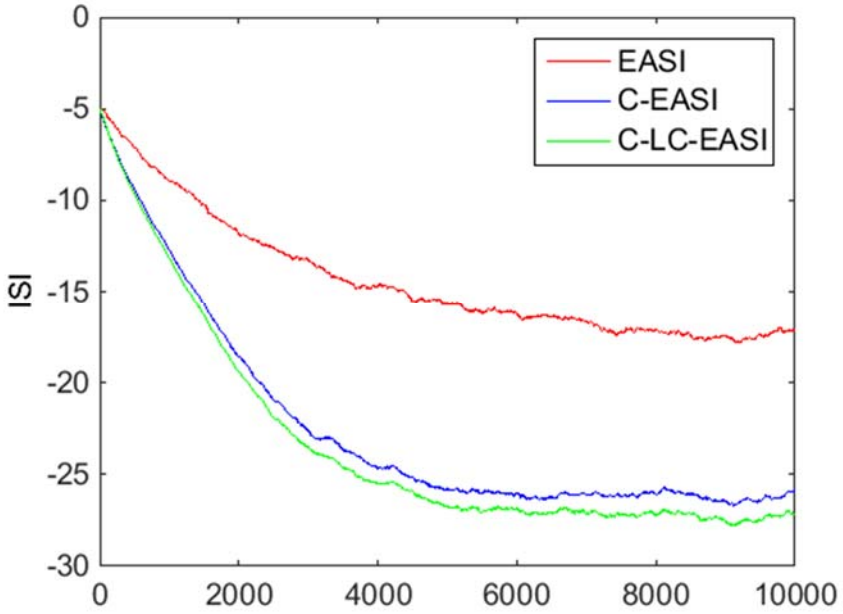

Fig. 3.8 Average ISI over multiple minimum phase channels with EASI, C-EASI and C-LC-EASI.

For the minimum phase typical example, we also tried a different initialization for matrix $\tilde{\mathbf{W}}$. Specifically, we set the center tap of the first row of $\tilde{\mathbf{W}}$ to be the only nonzero tap with $1+0.5j$. For the T-EASI case the $\tilde{\mathbf{W}}$ matrix has nonzero elements in one of the minor diagonals, and at convergence it can be expected that the largest tap will also appear in a certain minor diagonal. Since we want the matrix $\tilde{\mathbf{W}}\mathbf{H}_T$ to be a scaled identity, i.e. the first tap in the first row of $\tilde{\mathbf{W}}\mathbf{H}_T$ has the largest magnitude, the performance with this initialization cannot give good performance. For the C-EASI case, $\tilde{\mathbf{W}}$ is a circularly shifted version $\mathbf{I}_C$ of the identity, scaled by some constant. In this case, $\tilde{\mathbf{W}}\mathbf{H}_C$ may converge to a scaled version of $\mathbf{I}_C$. The ISI curve, as shown in Fig. 3.9, still gives the same result as the one as in Fig. 3.6, while the symbols in a block will be recovered with circular shift. At the same time, the LC constraint does not help convergence because it is hard to enforce length constraint due to the circular shift ambiguity. When the length constraint is enforced on the first row so that the elements are zero to a certain length, the performance may degrade.

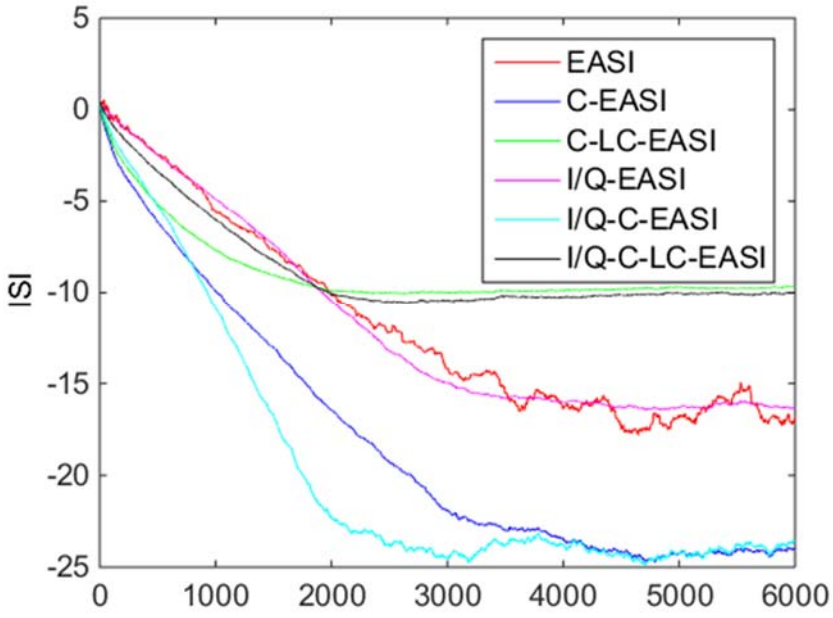

Fig. 3.9 ISI of the rows of matrix $\tilde{\mathbf{W}}\mathbf{H}_C$ for the minimum phase system with EASI, C-EASI and C-LC-EASI, initialize center tap of first row of $\tilde{\mathbf{W}}$ to value $1+0.5j$.

Next, an example for a *non-minimum* phase system with the CP scheme is shown. The impulse response and the zero-pole pattern are shown in Fig. 3.10 and Fig. 3.11. The size of the transmission block is set as $Q = 30$. When length constraint is enforced, we require $M = 20$. The performance of the C-EASI algorithm is shown in Fig. 3.12.

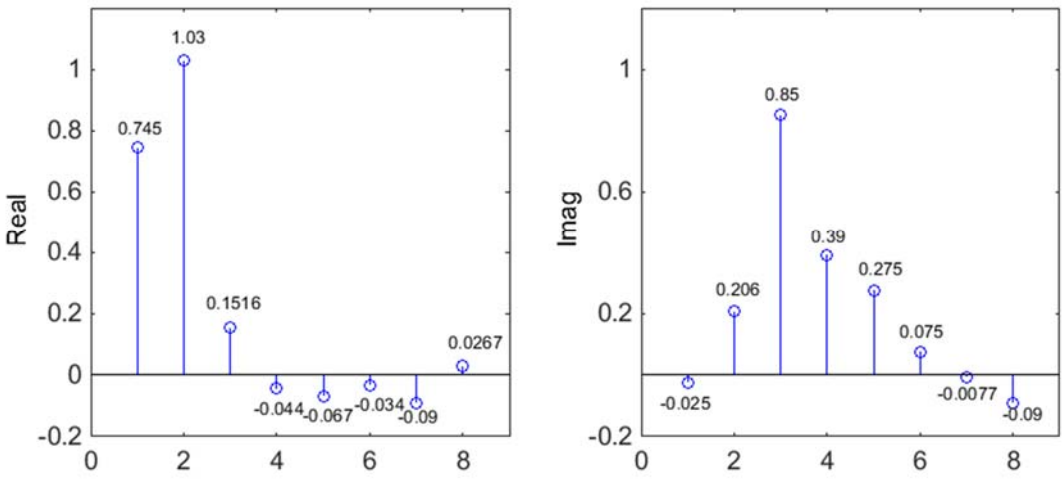

Fig. 3.10 Impulse response of the non-minimum phase channel

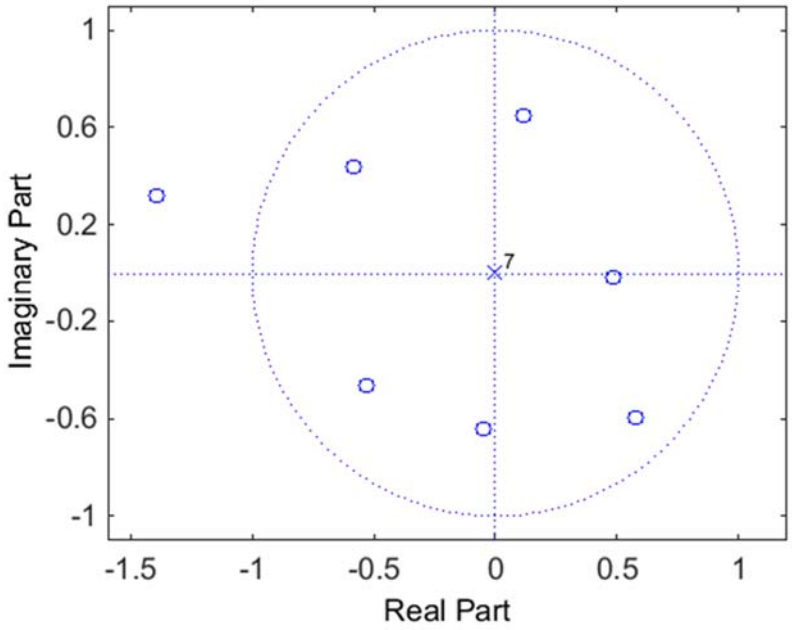

Fig. 3.11 Zero-pole pattern of the non-minimum phase channel

For the non-minimum phase system, the inverse of the $\mathbf{H}_T$ matrix has large coefficients in the upper right corner. With either EASI or different algorithms with Toeplitz constraint, we do not get convergence. However, with a reasonable block size Q,

it is not difficult for the CP scheme to converge to a separating matrix $\tilde{\mathbf{W}}$ such that $\tilde{\mathbf{W}}\mathbf{H}_C = \mathbf{I}_C$. From the figure, we can see that the C-EASI and C-LC-EASI can still yield good performance. Since there is circular shift, the LC constraint does not give better performance.

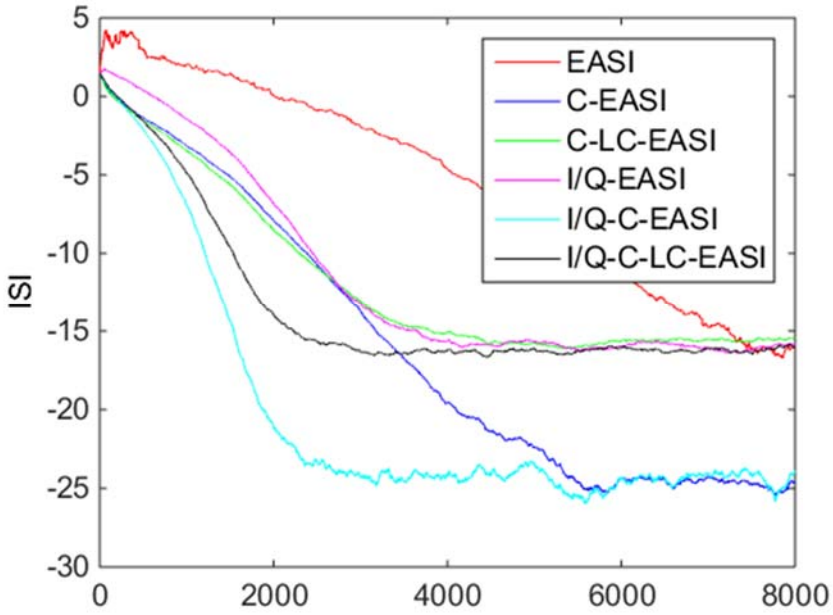

Fig. 3.12 ISI of the rows of matrix $\tilde{\mathbf{W}}\mathbf{H}_C$ for the non-minimum phase system with EASI, C-EASI and C-LC-EASI, , initialization $\tilde{\mathbf{W}} = (1+0.5j)\mathbf{I}$.

3.7 Discussion

We have compared in the simulation parts the performance for zero- and CP-padded block transmission schemes with the Toeplitz constrained and circulant constrained EASI algorithms, respectively. In this section, we will make a general comparison of the two schemes.

For either scheme, when the channel is a minimum phase system with the first tap having the largest magnitude, with a proper initialization the performance is good and comparable for the zero padding scheme with the T-EASI and for the CP scheme with the C-EASI.

According to Appendix 3B, in the implementation of the T-EASI, $(6Q-3)\log_2(2Q-1)+6Q-3$ additions and $(6Q-3)\log_2(2Q-1)+12Q-6$ multiplications are needed. To implement the C-EASI, $2Q\log_2 Q+3Q$ additions and $2Q\log_2 Q+6Q$ multiplications are needed. Comparing the computational complexity, we see that the cost of the C-EASI is slightly lower than that of the T-EASI. However the two schemes have the same order of computational complexity and when Q is large, the difference is small.

Although there is slight advantage in computational cost with the C-EASI compared to the T-EASI, the power efficiency is the opposite. In the block transmission, we need additional power to transmit the CP. Especially when the channel is long, i.e L is large, the power efficiency of the CP padded scheme can be significantly lower. In our example for the non-minimum phase channel, there is a $8/30 \approx 27\%$ power efficiency loss with the CP scheme.

From the simulations, we see that for the minimum phase system, the T-EASI is sensitive to initialization; while the C-EASI gives almost the same results for all circularly shifted versions of a particular $\tilde{\mathbf{W}}$.

Based on our many simulation experiments under different conditions, we found that for non-minimum phase channels and for minimum phase channels whose largest tap is not at the beginning, the T-EASI does not work because it is hard to get the inverse of the

"mixing" matrix $\mathbf{H}_T$; while the C-EASI is more robust since circular permutation is allowed, but the recovery of symbols is subject to circular permutation within blocks.

In general, for both of the schemes, there is a need to use an auxiliary technique to identify recovered symbols in the right order. However, the idea of using a Toeplitz or circulant structure constraint for equalization in block-transmission schemes is interesting and useful, and importantly our development in this chapter sets the stage for further development in the next chapter where the symbols are transmitted continuously.

3.8 Conclusion

In this chapter, we explored BE methods based on independent source separation for two block transmission schemes. With either padded zeros or cyclic prefix between blocks, BE can be formulated as a standard BSS problem and solved with ICA algorithms. The Toeplitz or circulant structure constraint can be enforced on the "separating" matrix with improved performance. The resulting matrix adaptations can be implemented more efficiently as equivalent vector adaptations. This work sets the stage for extension to the standard continuous transmission case in the next chapter.

References

[1] N. Al-Dhahir, M. Uysal, and H. Mheidat, "Single-carrier frequency domain equalization," *IEEE Signal Process. Mag.*, pp. 2–23, 2008.

[2] D. Falconer and B. Ariyavisitakul, Sirikiat Lek Benyamin-Seeyar, Anader Eidson, "Frequency domain equalization for single-carrier broadband wireless systems," *Commun. Mag. IEEE*, vol. 40, no. 4, pp. 58–66, 2002.

[3] N. Wang and S. D. Blostein, "Comparison of CP-based single carrier and OFDM with power allocation," *Commun. IEEE Trans.*, vol. 53, no. 3, pp. 391–394, 2005.

[4] Z. Wang, X. Ma, and G. B. Giannakis, "Optimality of single-carrier zero-padded block transmissions," in *Wireless Communications and Networking Conference*, 2002, pp. 660–664.

[5] Z. Wu and S. A. Kassam, "Blind Equalization Based On Blind Separation with Toeplitz Constraint," in *Signals, Systems and Computers, 2014 48th Asilomar Conference on*, 2014, pp. 1453–1457.

[6] W. F. Trench, "A note on Toepliz inversion formula," vol. 61, no. 1990, pp. 55–61, 1990.

[7] X. Lv and T. Z. Huang, "The inverses of block Toeplitz matrices," *Appl. Math. Comput.*, vol. 179, no. 1, pp. 243–247, 2006.

[8] J. F. Cardoso and B. H. Laheld, "Equivariant adaptive source separation," *IEEE Trans. Signal Process.*, vol. 44, no. 12, pp. 3017–3030, 1996.

[9] R. M. Gray, "Toeplitz and Circulant Matrices: A Review," *Found. Trends® Commun. Inf. Theory*, vol. 2, no. 3, pp. 155–239, 2006.

[10] L. He and S. A. Thaiupathump, Trasapong Kassam, "Blind separation of complex I/Q independent sources with phase recovery," *Signal Process. Lett. IEEE*, vol. 12, no. 5, pp. 419–422, 2005.

[11] T. Thaiupathump, "New algorithms for blind equalization and blind source separation/phase recovery," 2002.

Appendix 3A

In this part, we will prove the property (3.16) for circulant matrices:

Suppose $\mathbf{Y}$ *is a circulant matrix with size* $Q \times Q$ *, and* $\mathbf{X}$ *is a square matrix of the same size, then the following equation holds:*

$$Circulant\{\mathbf{X}\}\mathbf{Y} = Circulant\{\mathbf{XY}\} \ . \tag{3.16}$$

Before going to the proof, we define a modified version of the standard mod-Q representation of an integer, which will be used in the proof, as follows: for integer x,

$$(x)_Q = \begin{cases} Q & \text{if } \mathrm{mod}(x,Q) = 0 \\ \mathrm{mod}(x,Q) & \text{otherwise} \end{cases} \tag{3A.1}$$

Proof:

To prove equality of the matrix products in (3.16), we need to show that for any M, N with $1 \le M \le Q$, and $1 \le N \le Q$,

$$\left(Circulant\{\mathbf{X}\}\mathbf{Y}\right)_{M,N} = \left(Circulant\{\mathbf{XY}\}\right)_{M,N} \tag{3A.2}$$

where the sub-indices denote the (M, N)-th element of the matrix.

For a matrix $\mathbf{Z}$, forcing circulant structure means taking average along the diagonals circulantly, as shown in Fig. 3.1. As a result, with $Circulant\{\mathbf{Z}\}$, the (m,n)-th element becomes

$$(\mathbf{Z})_{m,n} = \frac{1}{Q}\sum_{j=1}^{Q} (\mathbf{Z})_{j,(j+n-m)_Q} \ , \ 1 \le m \le Q \text{, and } \ 1 \le n \le Q \, . \tag{3A.3}$$

Looking at the left side of (3A.2), we have

$$\begin{aligned}\left(Circulant\{\mathbf{X}\}\mathbf{Y}\right)_{M,N} &= \sum_{i=1}^{Q}\left(Circulant\{\mathbf{X}\}\right)_{M,i}(\mathbf{Y})_{i,N} \\ &= \sum_{i=1}^{Q}\left(\frac{1}{Q}\sum_{j=1}^{Q}(\mathbf{X})_{j,(j+i-M)_Q}\right)(\mathbf{Y})_{i,N} \\ &= \frac{1}{Q}\sum_{i=1}^{Q}\sum_{j=1}^{Q}(\mathbf{X})_{j,(j+i-M)_Q}(\mathbf{Y})_{i,N}.\end{aligned} \tag{3A.4}$$

Similarly,

$$\begin{aligned}\left(Circulant\{\mathbf{XY}\}\right)_{M,N} &= \frac{1}{Q}\sum_{j=1}^{Q}(\mathbf{XY})_{j,(j+N-M)_Q} \\ &= \frac{1}{Q}\sum_{j=1}^{Q}\sum_{i=1}^{Q}(\mathbf{X})_{j,i}(\mathbf{Y})_{i,(j+N-M)_Q}.\end{aligned} \tag{3A.5}$$

Since $\mathbf{Y}$ is a circulant matrix, each column includes all the elements in the matrix. As a result, for any $1 \le i \le Q$, and $1 \le j \le Q$, $(\mathbf{Y})_{i,(j+N-M)_Q}$ can be found in the N-th column of matrix $\mathbf{Y}$. Suppose in the N-th column, the m-th element equals $(\mathbf{Y})_{i,(j+N-M)_Q}$, then

$$(\mathbf{Y})_{i,(j+N-M)_Q} = (\mathbf{Y})_{m,N}\ , \tag{3A.6}$$

where $1 \le m \le Q$, then we should have

$$(N-m)_Q = ((j+N-M)_Q - i)_Q\,. \tag{3A.7}$$

This means

$$N - m + k_1 Q = (j + N - M + k_2 Q) - i + k_3 Q\,, \tag{3A.8}$$

where k_1, k_2 and k_3 are integers. The value of k_1, k_2 and k_3 are selected individually to make the corresponding terms in $(\bullet)_Q$ take integer values between 1 and Q.

We get from (3A.8) the expression of m. i.e.

$$\boxed{m = M - j + i + (k_1 - k_2 - k_3)Q + k_4 Q = (M - j + i)_Q}\ . \tag{3A.9}$$

Similarly, we find that

$$\boxed{i = (m + j - M)_Q} \quad . \tag{3A.10}$$

Substituting the element $(\mathbf{Y})_{i,(j+N-M)_Q}$ in the last line of (3A.5) with $(\mathbf{Y})_{m,N} = (\mathbf{Y})_{(M-j+i)_Q,N}$, we have

$$\begin{aligned} &\left(Circulant\{\mathbf{XY}\}\right)_{M,N} \\ &= \frac{1}{Q}\sum_{j=1}^{Q}\sum_{i=1}^{Q}(\mathbf{X})_{j,i}(\mathbf{Y})_{(M-j+i)_Q,N} \end{aligned} \quad . \tag{3A.11}$$

Let $m = (M - j + i)_Q$, then according to (3A.10) $i = (m + j - M)_Q$. Changing the variable i in (3A.11) to m, we have

$$\begin{aligned} &\left(Circulant\{\mathbf{XY}\}\right)_{M,N} \\ &= \frac{1}{Q}\sum_{j=1}^{Q}\sum_{m=1}^{Q}(\mathbf{X})_{j,(m+j-M)_Q}(\mathbf{Y})_{m,N} \end{aligned} \quad . \tag{3A.12}$$

Comparing (3A.12) with (3A.4), we see that they are exactly the same, and this completes the proof of the equation (3.16) in Section 3.3.2.

Appendix 3B

The number of additions and multiplications in the efficient implementation of the vector T-EASI and C-EASI schemes is listed in the following two tables. For fair comparison, for both algorithms that can be implemented by FFT, we start with the vector $\tilde{\mathbf{w}}_k$, $\mathbf{y}_k$, and $\mathbf{g}(\mathbf{y}_k)$ in time domain, and explore the computational complexity to get the change to $\tilde{\mathbf{w}}_k$ in time domain at each iteration.

T-EASI

To implement linear convolution with circular convolution, zeros needed to be padded. In the table, the superscript "(*p*)" means to pad $Q-1$ zeros at the end of the corresponding vector.

	Multiplication	Addition
$\tilde{\mathbf{w}}_{k+1} = \tilde{\mathbf{w}}_k - \lambda\{\tilde{\mathbf{w}}_k * \mathbf{u}_k\}_{1:Q}$ (via FFT in frequency domain)		
DFT of $\mathbf{y}_k^{(p)} / \mathbf{y}_k^{(p)*}$	$(2Q-1)\log_2(2Q-1)/2$	$(2Q-1)\log_2(2Q-1)/2$
DFT of $\mathbf{g}^{(p)}(\mathbf{y}_k) / \mathbf{g}^{(p)}(\mathbf{y}_k)^*$	$(2Q-1)\log_2(2Q-1)/2$	$(2Q-1)\log_2(2Q-1)/2$
DFT of $[flip(\mathbf{y}_k^*)^T, 0, ...0]^T$	$2Q-1$	
DFT of $[flip(\mathbf{g}(\mathbf{y}_k)^*)^T, 0, ...0]^T$	$2Q-1$	
DFT of elements in $\mathbf{U}_k$	$3(2Q-1)$	$3(2Q-1)$
elements in $\mathbf{U}_k$ with IDFT	$(2Q-1)\log_2(2Q-1)/2$	$(2Q-1)\log_2(2Q-1)/2$
$\mathbf{u}_k$ from $\mathbf{U}_k$, DFT of $\mathbf{u}_k^{(p)}$	$(2Q-1)\log_2(2Q-1)/2$	$(2Q-1)\log_2(2Q-1)/2$
DFT of $\tilde{\mathbf{w}}_k^{(p)}$	$(2Q-1)\log_2(2Q-1)/2$	$(2Q-1)\log_2(2Q-1)/2$
DFT of $\mathbf{u}_k^{(p)} * \tilde{\mathbf{w}}_k^{(p)}$	$2Q-1$	
$\{\tilde{\mathbf{w}}_k * \mathbf{u}_k\}_{1:Q}$ with an IDFT	$(2Q-1)\log_2(2Q-1)/2$	$(2Q-1)\log_2(2Q-1)/2$
Total	$(6Q-3)\log_2(2Q-1)$ $+12Q-6$	$(6Q-3)\log_2(2Q-1)$ $+6Q-3$

C-EASI

	Multiplication	Addition
$\tilde{\mathbf{w}}_{k+1} = \tilde{\mathbf{w}}_k - \lambda \tilde{\mathbf{w}}_k \otimes \mathbf{u}_k$ (via FFT in frequency domain)		
DFT of $\mathbf{y}_k$ ($\mathbf{y}_k^*$)	$(Q/2)\log_2 Q$	$(Q/2)\log_2 Q$
DFT of $\mathbf{g}(\mathbf{y}_k)$ ($\mathbf{g}(\mathbf{y}_k)^*$)	$(Q/2)\log_2 Q$	$(Q/2)\log_2 Q$
DFT of *flip*($\mathbf{y}_k^*$)	Q	
DFT of *flip*($\mathbf{g}(\mathbf{y}_k)^*$)	Q	
DFT of $\mathbf{u}_k$	$3Q$	$3Q$
DFT of $\tilde{\mathbf{w}}_k$	$(Q/2)\log_2 Q$	$(Q/2)\log_2 Q$
DFT of $\tilde{\mathbf{w}}_k \otimes \mathbf{u}_k$	Q	
$\tilde{\mathbf{w}}_k \otimes \mathbf{u}_k$ with an IDFT	$(Q/2)\log_2 Q$	$(Q/2)\log_2 Q$
Total	$2Q\log_2 Q + 6Q$	$2Q\log_2 Q + 3Q$

Chapter 4

Toeplitz Constrained ICA for Symbol-Rate Blind Equalization

4.1 Introduction

We have seen from the previous chapter that constrained independent component analysis (ICA) can be used to solve blind equalization (BE) problems with block transmission schemes. With symbols transmitted in blocks with padding, the BE problem can be modeled as a standard blind source separation (BSS) problem, where the independence of the symbols is exploited in the adaptation. In this chapter, we will show how the ICA-based algorithms with constraints can be applied to symbol-rate sampling blind equalization. Even though the associated mixing matrix may not satisfy the dimension condition of a standard BSS model, we can use an ICA-based algorithm appropriately modified to recover

the source sequence. The matrix adaptation can be simplified as equalizer vector adaptation, with efficient implementation schemes.

In Section 4.2, the symbol-rate sampling BE is formulated as an under-determined BSS problem with matrix expression. In Section 4.3, the EASI algorithm is used as an example to illustrate the proposed scheme of constrained ICA-based algorithm. The adaptation of the Toeplitz constrained EASI (T-EASI) algorithm for the matrix that contains the equalizer coefficients is given. The matrix adaptation is simplified to an equivalent efficient equalizer vector adaptation in Section 4.4. To further reduce computational complexity, in Section 4.5 we show how the T-EASI algorithm can be implemented with FFT. At the same time, instead of updating a whole vector of equalizer outputs and their nonlinear cross-correlations, two approximation schemes can be used. In Section 4.7, similar to the idea that has been introduced in the previous chapter, I/Q independence constraint is used for phase recovery when the source symbol has independent I/Q parts. In addition, phase recovery with an appropriate choice of nonlinearity in the T-EASI algorithm is proposed. In Section 4.8, we give examples to show that the Toeplitz constraint idea can also work with other ICA-based algorithms.

4.2 Symbol-Rate Blind Equalization and Blind Source Separation

In this section, we will see how the symbol-rate BE problem can be formulated as an underdetermined BSS problem. Similar to the block transmission schemes in Chapter 3, a

matrix expression will be used as the starting point to make the problem have the form of a BSS problem. With such a formulation, the independence constraint can be exploited with constrained ICA algorithms. It will be shown later that matrix adaptation can be simplified to equalizer vector adaptation.

Suppose we process channel outputs in blocks of size $P + M$ ($P > 0$) at symbol rate by sliding along the sequence of observations from the channel, with one-symbol shift each time. Note that M is the FIR equalizer order. Compared to standard BE scheme where a block of $M + 1$ channel outputs are processed for single equalizer output, a longer channel output block is processed to generate multiple equalizer outputs. Specifically, to allow the use of an ICA algorithm, we require that $P > 1$.

Let $\tilde{\mathbf{x}}_k = [x(k), x(k-1), ..., x(k-P-M+1)]^T$ be the k-th observation block with length $P+M$. Then the channel outputs can be expressed in matrix form as

$$\tilde{\mathbf{x}}_k = \mathbf{H}\mathbf{s}_k + \mathbf{v}_k, \tag{4.1}$$

where $\mathbf{s}_k = [s(k), s(k-1), ..., s(k-P-M-L+1)]^T$ is the k-th source vector with length $P+M+L$, $\mathbf{v}_k = [v(k), v(k-1), ..., v(k-P-M-L+1)]^T$ is the additive noise vector, and $\mathbf{H}$ is a $(P+M)\times(P+M+L)$ Toeplitz matrix composed of the impulse response of the channel, i.e.

$$\mathbf{H} = \begin{pmatrix} h(0) & h(1) & \cdots & h(L) & \cdots & \cdots & 0 \\ & \ddots & & \cdots & & \ddots & \cdots \\ \cdots & h(0) & h(1) & \cdots & h(L) & \cdots & \\ & & \ddots & & \cdots & & \ddots \\ 0 & \cdots & \cdots & h(0) & h(1) & \cdots & h(L) \end{pmatrix}. \tag{4.2}$$

We have seen in Chapter 2 that an ideal equalizer is defined to satisfy the requirement that the cascaded system has the response

$$\mathbf{c}^{ideal} = e^{j\theta}(\underbrace{0,...,0}_{d \text{ zeros}},1,0...0) \ . \tag{4.3}$$

Suppose $\mathbf{w}^{ideal}$ is an ideal equalizer of order M such that cascaded system response satisfies equation (4.3). Construct a $P\times(P+M)$ Toeplitz matrix $\mathbf{W}^{ideal}$ from the coefficients of $\mathbf{w}^{ideal}$ as follows:

$$\mathbf{W}^{ideal} = \begin{pmatrix} w(0) & w(1) & \dots & w(M) & \dots & \dots & 0 \\ & \ddots & & \dots & & \ddots & \dots \\ \dots & w(0) & w(1) & \dots & w(M) & \dots & \\ & & \ddots & & \dots & & \ddots \\ 0 & \dots & \dots & w(0) & w(1) & \dots & w(M) \end{pmatrix}, \tag{4.4}$$

where the elements inside the matrix are the coefficients of $\mathbf{w}^{ideal}$. The super-index "ideal" is omitted for notation simplicity. We call such a matrix an ideal *equalizer matrix*.

Ideally, the product of $\mathbf{W}$ and $\mathbf{H}$ is then the $P\times(P+M+L)$ Toeplitz matrix

$$\mathbf{C} \triangleq \mathbf{WH} \simeq \mathbf{C}^{ideal} = \mathbf{W}^{ideal}\mathbf{H} = \begin{pmatrix} \mathbf{c}^{ideal} & 0 & ... & 0 & \\ 0 & \mathbf{c}^{ideal} & 0 & ... & 0 \\ & ... & & ... & \\ 0 & & ... & 0 & \mathbf{c}^{ideal} \end{pmatrix}, \tag{4.5}$$

where $\mathbf{c}^{ideal}$ was the ideal cascaded response as in (4.3). In the matrix $\mathbf{C}$ defined in (4.5), there is only one nonzero element in each row in the ideal case. Ignoring noise, if we apply matrix $\mathbf{W}^{ideal}$ to the channel output vector $\tilde{\mathbf{x}}_k$, we are able to recover P of the $P+M+L$ source symbols. Thus in the BE problem with matrix formulation, we want to find a matrix $\mathbf{W}$ such that the outputs

$$\tilde{\mathbf{y}}_k = \mathbf{W}\tilde{\mathbf{x}}_k \tag{4.6}$$

can approximately recover the elements in $\mathbf{s}_k$. Note here $\tilde{\mathbf{y}}_k = [y(k), \tilde{y}^{(k)}(k-1), ..., \tilde{y}^{(k)}(k-P+1)]^T$, where $\tilde{y}^{(k)}(k-i) = \mathbf{w}_k^T \mathbf{x}_{k-i}$, $i = 0,1,...,P-1$, is the equalizer output using the equalizer at the k-th iteration, and $\mathbf{x}_{k-i} = [x(k-i), x(k-i-1), ..., x(k-i-M)]^T$ is length-$(M+1)$ sub-vector in $\tilde{\mathbf{x}}_k$. The tilde notation is to emphasize the difference between our block scheme and the standard BE schemes. In standard BE, only one output is considered at each iteration, and the output is from the equalizer at the corresponding iteration, i.e. $y(k-i) = \mathbf{w}_{k-i}^T \mathbf{x}_{k-i}$. Specially, the first output in $\tilde{\mathbf{y}}_k$ of (4.6) comes from the equalizer at the k -th iteration, and is the same as the one used in the standard BE schemes, i.e. $y(k) = \tilde{y}^{(k)}(k)$.

The model here is similar to that of the BSS problem introduced in Section 2.2.1, and the ideal "separating matrix" we want to find is the matrix $\mathbf{W}^{ideal}$ containing the coefficients of an ideal equalizer. In an ICA-based algorithm, the desired independence of the outputs in $\tilde{\mathbf{y}}_k$ is used to update the matrix $\mathbf{W}$ at each iteration.

4.3 Toeplitz-Constrained ICA for BE

In Section 2.2.4 we have introduced a popular ICA-based algorithm for BSS, the EASI algorithm [1]. The EASI algorithm was used in Chapter 3 for BE with block transmission [2]. In this section, we will show how constrained ICA can be used in symbol-rate sampling equalization with continuous transmission scheme. The EASI algorithm will be used to

explain our Toeplitz-constrained scheme, however such structure constraint can be used generally in ICA-based algorithms for BE, which will be shown in Section 4.8

Recall from section 4.2 that BE can be modeled as a BSS problem with the matrix expressions in (4.1) and (4.6):

$$\tilde{\mathbf{x}}_k = \mathbf{H}\mathbf{s}_k + \mathbf{v}_k,$$

$$\tilde{\mathbf{y}}_k = \mathbf{W}_k \tilde{\mathbf{x}}_k .$$

Although the "mixing matrix" $\mathbf{H}$ here does not satisfy the dimension requirement of a standard BSS problem, we will see that constrained ICA-based algorithms can still be applied to recover the source sequence.

With the standard EASI algorithm, the matrix $\mathbf{W}$ is updated as

$$\mathbf{W}_{k+1} = \mathbf{W}_k - \lambda\left[\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k^H - \mathbf{I} + \mathbf{g}(\tilde{\mathbf{y}}_k)\tilde{\mathbf{y}}_k^H - \tilde{\mathbf{y}}_k \mathbf{g}(\tilde{\mathbf{y}}_k)^H\right]\mathbf{W}_k , \qquad (4.7)$$

Where $\mathbf{g}(\tilde{\mathbf{y}}_k) = \left[g(y(k)), g(\tilde{y}^{(k)}(k-1)), ..., g(\tilde{y}^{(k)}(k-P+1))\right]^T$ is the component-wise derivative of the contrast function $G(\mathbf{y}_k)$ at $\tilde{\mathbf{y}}_k$, and λ is the adaptation step-size. Recall that the first two terms are for whitening, and the last two terms provide non-linear decorrelation for independence.

From the definition of $\mathbf{W}^{ideal}$ in (4.4) we know that the ideal equalizer matrix should be a Toeplitz matrix with repeated vector in each row. As a result, after each update, we need to project the updated matrix onto the space of Toeplitz matrices with the structure of $\mathbf{W}^{ideal}$. An intuitive guess of one projection is to impose Toeplitz structure on $\mathbf{W}_{k+1}$ by taking averages along the descending diagonals of the matrix after forcing the $P-1$ diagonals on the upper right and lower left corners of the matrix to be zero. We can show

that this structure forcing $Toeplitz\{\mathbf{W}_{k+1}\}$ turns out to be the *orthogonal projection* onto the space of Toeplitz matrices that has the same structure as $\mathbf{W}^{ideal}$.

A matrix can be considered as a long vector where the columns of the matrix are concatenated. Two vectors are orthogonal to each other if their inner product is zero. As an extension to matrix case, the *inner product* of two complex matrix $\mathbf{X}$ and $\mathbf{Y}$ is defined as $\langle \mathbf{X}, \mathbf{Y} \rangle = \text{Trace}\{\mathbf{Y}^H \mathbf{X}\}$. Matrices $\mathbf{X}$ and $\mathbf{Y}$ are defined to be orthogonal to each other when $\langle \mathbf{X}, \mathbf{Y} \rangle = 0$.

Suppose $\mathbf{W}$ is a $P \times (P+M)$ matrix at the *k*-th iteration, with sub-index *k* omitted for simplicity of notation. Let $\mathbf{W}_{Toe}$ be the resulting matrix after enforcing Toeplitz structure by taking averages along diagonals and forcing upper right and lower left parts zero. We can then show that

$$\langle \mathbf{W} - \mathbf{W}_{Toe}, \mathbf{W}_{Toe} \rangle = 0, \tag{4.8}$$

which means that $\mathbf{W}_{Toe} = Toeplitz\{\mathbf{W}\}$ is the orthogonal projection of $\mathbf{W}$ onto the space of Toeplitz matrices with the structure of $\mathbf{W}^{ideal}$. The proof is given in Appendix 4A.

With the Toeplitz structure requirement included, the EASI algorithm with Toeplitz constraint (T-EASI) can be expressed as a two-step adaptation as follows:

$$\begin{aligned} \hat{\mathbf{W}}_{k+1} &= \mathbf{W}_k - \lambda[\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k^H - \mathbf{I} + \mathbf{g}(\tilde{\mathbf{y}}_k)\tilde{\mathbf{y}}_k^H - \tilde{\mathbf{y}}_k \mathbf{g}^H(\tilde{\mathbf{y}}_k)]\mathbf{W}_k, \\ \mathbf{W}_{k+1} &= Toeplitz\{\hat{\mathbf{W}}_{k+1}\}. \end{aligned} \tag{4.9}$$

Similar to the block transmission case, if we start by initializing $\mathbf{W}$ as a Toeplitz matrix, the $\mathbf{W}$ matrix can remain in the space of Toeplitz matrices with the Toeplitz structure constraint on the perturbation, i.e. on the term $[\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k^H - \mathbf{I} + \mathbf{g}(\tilde{\mathbf{y}}_k)\tilde{\mathbf{y}}_k^H - \tilde{\mathbf{y}}_k \mathbf{g}^H(\tilde{\mathbf{y}}_k)]\mathbf{W}_k$ in the first line of (4.9). However, we write it as a two-step adaptation to make the matrix

adaptation without Toeplitz constraint more explicit. For this ICA-based algorithm we need $P \geq 2$ to allow use of independence of output symbols. The choice of P should reflect a balance between using more independence constraints and requiring more sample cross-correlations within the algorithm that may slow down convergence. From multiple experiments for different channels, we found that good performance is usually obtained with a choice of $P \approx M/2$.

4.4 Equalizer Vector Adaptation

In this section we will show how the equalizer matrix adaptation can be simplified to an equivalent equalizer vector adaptation, which can reduce computational complexity.

In (4.9) we have given the adaptation of the separating matrix for our BE problem with the T-EASI algorithm. With Toeplitz structure constraint, the separating matrix will contain the equalizer vector in each row at the end of each iteration. Although the whole matrix is updated each time, if we focus on the equalizer coefficients in each row, the computational complexity of the scheme can be reduced.

Let $\mathbf{U}_k = \tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k^H - \mathbf{I} + \mathbf{g}(\tilde{\mathbf{y}}_k)\tilde{\mathbf{y}}_k^H - \tilde{\mathbf{y}}_k \mathbf{g}(\tilde{\mathbf{y}}_k)^H$, which is a $P \times P$ matrix containing the cross-correlation terms, so that the updates for the separating matrix in (4.9) can be written as

$$\begin{aligned} \hat{\mathbf{W}}_{k+1} &= \mathbf{W}_k - \lambda \mathbf{U}_k \mathbf{W}_k, \\ \mathbf{W}_{k+1} &= Toeplitz\{\hat{\mathbf{W}}_{k+1}\}. \end{aligned} \tag{4.10}$$

Let $\hat{\mathbf{W}}_{k+1}(i,:)$ and $\mathbf{W}_k(i,:)$ be the i-th row of $\hat{\mathbf{W}}_{k+1}$ and of $\mathbf{W}_k$, respectively, and let $U_k(i,j)$ be the (i,j)-th element of matrix $\mathbf{U}_k$, then

$$\hat{\mathbf{W}}_{k+1}(i,:) = \mathbf{W}_k(i,:) - \lambda \sum_{j=1}^{P} U_k(i,j)\mathbf{W}_k(j,:). \tag{4.11}$$

Setting the lower left and upper right of the matrix $\hat{\mathbf{W}}_{k+1}$ to zero, we only update the elements within the diagonal band which contains the equalizer coefficients. Let $\hat{\mathbf{W}}_{k+1}(i, m:n)$ be the row vector containing the m-th to n-th elements of the i-th row of matrix $\hat{\mathbf{W}}_{k+1}$. In the i-th row, the i-th to $i+M$-th elements form the equalizer vector, and they are updated as

$$\begin{aligned} \hat{\mathbf{W}}_{k+1}(i, i:M+i) &= \mathbf{W}_k(i, i:M+i) - \lambda \sum_{j=1}^{P} U_k(i,j)\mathbf{W}_k(j, i:M+i) \\ &= \mathbf{W}_k(i, i:M+i) - \lambda \mathbf{U}_k(i,:)\mathbf{W}_k(:, i:M+i) \end{aligned} \tag{4.12}$$

From the expression in (4.12), we can see that each row of matrix $\mathbf{U}_k$ pre-multiplies a $P \times (M+1)$ submatrix of $\mathbf{W}_k$ to update the elements in a particular row of $\mathbf{W}_k$ within the diagonal band, as illustrated in Fig. 4.1.

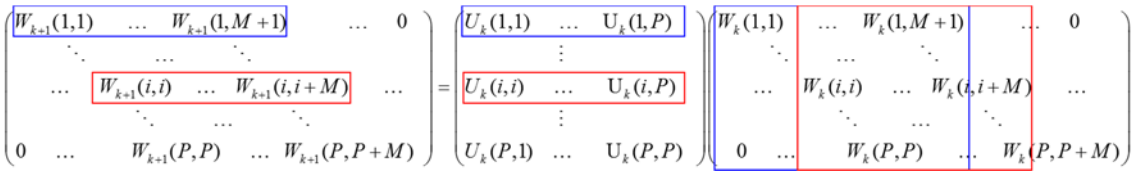

Fig. 4.1 Row updating of **W** within the band

To enforce Toeplitz structure on matrix $\hat{\mathbf{W}}$, we take averages along descending diagonals, giving

$$\frac{1}{P}\sum_{i=1}^{P} \hat{\mathbf{W}}_{k+1}(i, i:M+i) = \frac{1}{P}\sum_{i=1}^{P} \mathbf{W}_k(i, i:M+i) - \lambda \frac{1}{P}\sum_{i=1}^{P}\sum_{j=1}^{P} U_k(i,j)\mathbf{W}_k(j, i:M+i). \tag{4.13}$$

The operations (4.12) and (4.13) together are equivalent to the Toeplitz structure constraint $\mathbf{W}_{k+1} = Toeplitz\{\hat{\mathbf{W}}_{k+1}\}$ in (4.10).

With (4.12) and (4.13) at each iteration, each row of $\mathbf{W}$ then contains the updated coefficients of the equalizer as shown in Fig. 4.2.

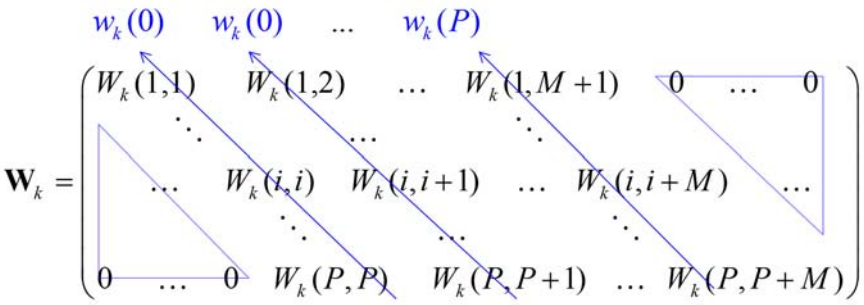

Fig. 4.2 Structure of the equalizing matrix

Denoting the equalizer vector at time k as $\mathbf{w}_k$, we have

$$\mathbf{w}_{k+1}^T = \mathbf{w}_k^T - \lambda \frac{1}{P} \sum_{i=1}^{P} \sum_{j=1}^{P} U_k(i,j) \mathbf{W}_k(j, i:M+i) \tag{4.14}$$

The l-th ($0 \le l \le M$) coefficient of the equalizer has the iterations

$$w_{k+1}(l) = w_k(l) - \lambda \frac{1}{P} \sum_{i=1}^{P} \sum_{j=1}^{P} U_k(i,j) W_k(j, i+l) \tag{4.15}$$

Letting $\tau = i - j$, with variables changing from i and j to τ and j as shown in Fig. 4.3, we see that (4.15) can be written as

$$w_{k+1}(l) = w_k(l) - \frac{\lambda}{P} \sum_{\tau=1-P}^{P-1} \sum_{j=\max\{1-\tau,1\}}^{\min\{P-\tau,P\}} U_k(\tau+j, j) W_k(j, \tau+j+l) \tag{4.16}$$

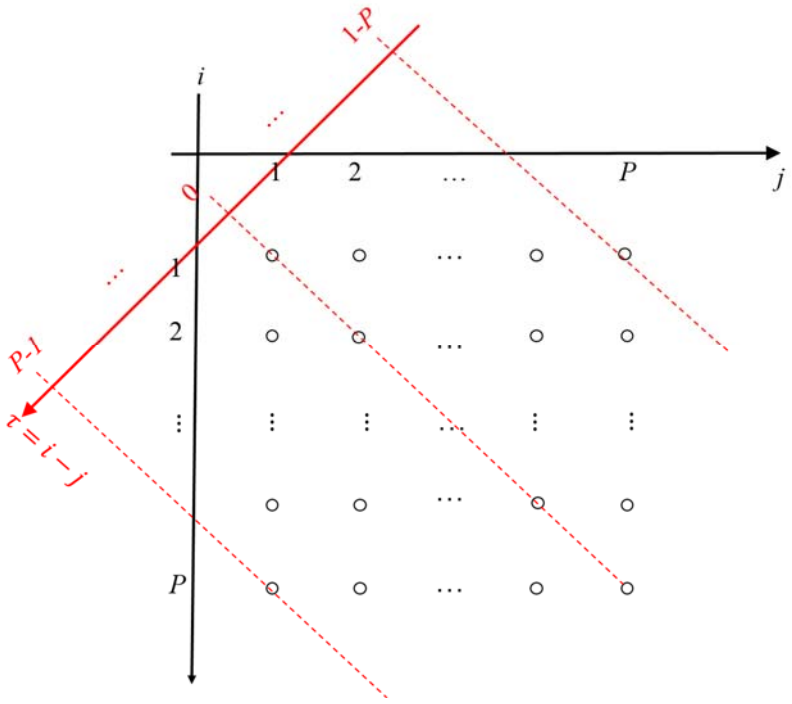

Fig. 4.3 Changing variables from i and j to $\tau = i - j$ and j

From Fig. 4.2, we can see that in the Toeplitz matrix **W**, the elements outside the diagonal band are zero, and the range is decided by the element index. Specifically, $W_k(j, \tau + j + l) = 0$ if $j > j + \tau + l$ or $j + M < j + \tau + l$. Thus the summation range over τ in (4.16) becomes $j \le j + \tau + l \le j + M$, i.e. $-l \le \tau \le M - l$. Substituting the elements in the matrix within the Toeplitz band with the coefficients of the equalizer vector, i.e. $W_k(j, \tau + j + l) = w_k(\tau + l)$, we have

$$w_{k+1}(l) = w_k(l) - \frac{\lambda}{P} \sum_{\tau=\max\{1-P,-l\}}^{\min\{P-1,M-l\}} \sum_{j=\max\{1-\tau,1\}}^{\min\{P-\tau,P\}} U_k(\tau + j, j) w_k(\tau + l) \qquad (4.17)$$

To make the expression in (4.17) simpler, let

$$r_k(\tau) = \sum_{j=\max\{1-\tau,1\}}^{\min\{P-\tau,P\}} U_k(\tau + j, j) \ , \qquad (4.18)$$

i.e.

$$r_k(\tau)=\begin{cases}\sum_{j=1}^{P-\tau}\tilde{y}^{(k)}(k-(\tau+j)+1)\tilde{y}^{(k)}(k-j+1)^*+g\left(\tilde{y}^{(k)}(k-(\tau+j)+1)\right)\tilde{y}^{(k)}(k-j+1)^* \\ \qquad -\tilde{y}^{(k)}(k-(\tau+j)+1)g\left(\tilde{y}^{(k)}(k-j+1)\right)^* & \text{when } \tau>0 \\ \sum_{j=1}^{P}\tilde{y}^{(k)}(k-j+1)\tilde{y}^{(k)}(k-j+1)^*-1+g\left(\tilde{y}^{(k)}(k-j+1)\right)\tilde{y}^{(k)}(k-j+1)^* \\ \qquad -\tilde{y}^{(k)}(k-j+1)g\left(\tilde{y}^{(k)}(k-j+1)\right)^* & \text{when } \tau=0 \\ \sum_{j=1-\tau}^{P}\tilde{y}^{(k)}(k-(\tau+j)+1)\tilde{y}^{(k)}(k-j+1)^*+g\left(\tilde{y}^{(k)}(k-(\tau+j)+1)\right)\tilde{y}^{(k)}(k-j+1)^* \\ \qquad -\tilde{y}^{(k)}(k-(\tau+j)+1)g\left(\tilde{y}^{(k)}(k-j+1)\right)^* & \text{when } \tau<0\end{cases}.$$

Then the adaptation for the l-th coefficient of vector $\mathbf{w}$ becomes

$$w_{k+1}(l)=w_k(l)-\frac{\lambda}{P}\sum_{\tau=\max\{1-P,-l\}}^{\min\{P-1,M-l\}}r_k(\tau)w_k(\tau+l) \tag{4.19}$$

Fig. 4.4 shows the relation between the matrix $\mathbf{U}_k$ and the parameters $r_k(\tau)$ defined in (4.18) for $1-P\le\tau\le P-1$. The difference in color will be used to explain a scheme for approximation of $\mathbf{U}_k$ in the next section, and can be ignored for now.

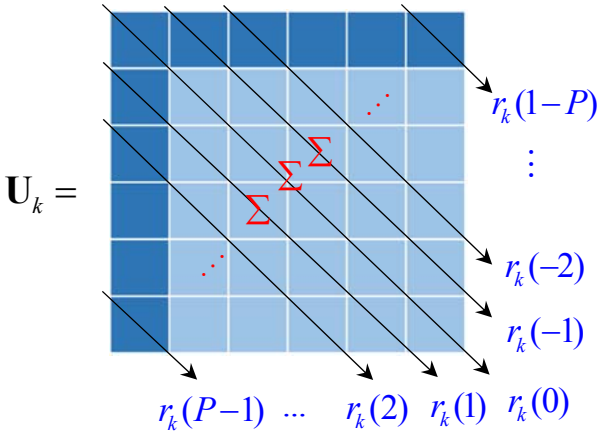

Fig. 4.4 Obtaining the $r_k(\tau)$, $1-P\le\tau\le P-1$, from matrix $\mathbf{U}_k$

Equation (4.19) is the general result showing that the adaptation in (4.10) for the separating matrix can be expressed as updates for the coefficients of the equalizer vector, in terms of nonlinear cross-correlation parameters of the outputs.

When $P \geq M+1$, $r_k(\tau)$ is well-defined in the range $-M \leq \tau \leq M$. For $P < M+1$, $r_k(\tau)$ is not defined beyond $1-P \leq \tau \leq P-1$, and we can additionally define

$$r_k(\tau) \triangleq 0 \quad \text{for} \quad -M \leq \tau < 1-P \text{ and } P-1 < \tau \leq M \tag{4.20}$$

so that $r_k(\tau)$ always has definition over the range $-M \leq \tau \leq M$. With these definitions of $r_k(\tau)$, (4.19) can be further simplified to give

$$w_{k+1}(l) = w_k(l) - \frac{\lambda}{P} \sum_{\tau=-l}^{M-l} r_k(\tau) w_k(\tau + l) \tag{4.21}$$

Writing (4.21) in vector-matrix form, we get

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \frac{\lambda}{P} \boldsymbol{\Psi}_k \mathbf{w}_k \tag{4.22}$$

where $\boldsymbol{\Psi}_k$ is a $(M+1)\times(M+1)$ Toeplitz matrix containing the cross-correlation terms $r_k(\tau)$ with the expression:

$$\boldsymbol{\Psi}_k = \begin{bmatrix} r_k(0) & r_k(1) & & r_k(M) \\ r_k(-1) & r_k(0) & & r_k(M-1) \\ \vdots & \vdots & \cdots & \vdots \\ r_k(-M) & r_k(-M+1) & & r_k(0) \end{bmatrix}. \tag{4.23}$$

The adaptation for equalizer vector in (4.22) is equivalent to the matrix adaptation in (4.10). To compare the computational complexity of the two algorithms, we start with the matrix $\mathbf{U}_k$ for both of them. In the analysis of computational complexity, we assume that $P < M+1$, which is usually true according to our experiments.

In the matrix adaptation (4.10), the matrix multiplication $\mathbf{U}_k\mathbf{W}_k$ needs $P^2(M+1)$ multiplications and $P(P-1)M$ additions, where the zeros in the lower left and upper right part of $\mathbf{W}_k$ are taken into consideration. In this matrix adaptation, starting from a Toeplitz $\mathbf{W}_k$, we can enforce the Toeplitz constraint on $\mathbf{U}_k\mathbf{W}_k$ by taking averages along the diagonals, which needs $(P-1)(M+1)$ additions; and the division by constant P can be combined with the step-size. Then the matrix subtraction can be achieved by considering a single row because of the Toeplitz structure, which needs $M+1$ additions. Thus in total[1], there are $P^2(M+1)$ multiplications and P^2M+P additions.

In the vector adaptation (4.22), getting $\mathbf{\Psi}_k$ from $\mathbf{U}_k$ needs $(P-1)^2$ additions. The matrix and vector multiplication $\mathbf{\Psi}_k\mathbf{w}_k$ needs $2PM-P^2+3P-M-1$ multiplications and $2PM-P^2+3P-2M-2$ additions considering $P<M+1$ and there are zeros at the upper right and lower left corners of $\mathbf{\Psi}_k$. Lastly, the vector subtraction needs $M+1$ additions. So there are $2PM-P^2+3P-M-1$ multiplications and $2PM+P-M-2$ additions. If we take $P\approx M/2$, the vector adaption approximately reduces the computational complexity by a factor of $\frac{2}{3}P$.

[1] The scalar multiplication of the step-size is omitted in the analysis of computational complexity.

4.5 Computationally Efficient Implementation for Equalizer Vector

Although the computational complexity of the vector adaptation has been reduced compared to matrix adaptation, it is still P times that of the popular Bussgang-type BE algorithms. In this section we will show how vector adaptation can be implemented efficiently with FFT and approximation of the cross-correlation terms, which can reduce the computational complexity further.

4.5.1 FFT Implementation of T-EASI

In T-EASI algorithm, we can consider three steps in the adaptation at each iteration: generating equalizer outputs, obtaining cross-correlation terms and the matrix $\mathbf{\Psi}_k$, implementing adaptation with matrix and vector multiplication. We will next explain in detail how each step can be implemented efficiently to reduce computational complexity.

First, a block of equalizer outputs is a truncated version of the convolution of equalizer and the channel output sequence, i.e.

$$\tilde{\mathbf{y}}_k = \mathbf{W}_k \tilde{\mathbf{x}}_k = (\tilde{\mathbf{x}}_k * flip\{\mathbf{w}_k\})_{M+1:M+P}, \tag{4.24}$$

where $flip\{\bullet\}$ inverts the order the elements in the vector. As a result, FFT can be used to reduce computational complexity. Specifically, with $P-1$ zeros padded at the end of $flip(\mathbf{w}_k)$, the length-$(P+M)$ sequence

$$\mathrm{FFT}\{\tilde{\mathbf{x}}_k\} \odot \mathrm{FFT}\{[flip\{\mathbf{w}_k\}^T, 0, \ldots, 0]^T\},$$

where $\odot$ is component-wise multiplication, would give the DFT of a vector whose last P elements compose $\mathbf{y}_k$.

Equation (4.22) gives the adaptation for the equalizer vector. Recall that the term $r_k(\tau)$ is defined in (4.18) as

$$r_k(\tau) = \sum_{j=\max\{1-\tau,1\}}^{\min\{P-\tau,P\}} U_k(\tau+j, j), \quad 1-P \le \tau \le P-1 \,,$$

where

$$\begin{aligned} U_k(\tau+j, j) = {} & \tilde{y}^{(k)}(k-\tau-j+1)\tilde{y}^{(k)}(k-j+1)^* - 1_{\tau=0} + g(\tilde{y}^{(k)}(k-\tau-j+1))\tilde{y}^{(k)}(k-j+1)^* \\ & -\tilde{y}^{(k)}(k-\tau-j+1)g(\tilde{y}^{(k)}(k-j+1))^* \end{aligned} \,,$$

and $1_{\tau=0}$ is an indicator function. As a result, for a fixed τ, $r_k(\tau)$ is the summation of the cross-correlation parameters with the same time lag τ, and thus can be computed using linear convolution.

Let $\tilde{\mathbf{r}}_k \triangleq \tilde{\mathbf{y}}_k * flip\{\tilde{\mathbf{y}}_k{}^*\} + \mathbf{g}(\tilde{\mathbf{y}}_k) * flip\{\tilde{\mathbf{y}}_k{}^*\} - \tilde{\mathbf{y}}_k * flip\{\mathbf{g}(\tilde{\mathbf{y}}_k)^*\}$. Suppose $\tilde{\mathbf{r}}_k$ has coefficient index from 1 to $2P-1$, then

$$r_k(\tau) = \begin{cases} \tilde{r}_k(\tau+P) & 1-P \le \tau \le -1 \text{ or } 1 \le \tau \le P-1 \\ \tilde{r}_k(\tau+P) - P & \tau = 0 \end{cases} \tag{4.25}$$

There are three convolutional terms in $\tilde{\mathbf{r}}_k$, and we will take $\mathbf{g}(\tilde{\mathbf{y}}_k) * flip\{\tilde{\mathbf{y}}_k{}^*\}$ as an example to show how it can be obtained via FFT. To calculate the linear convolution $\mathbf{g}(\tilde{\mathbf{y}}_k) * flip\{\tilde{\mathbf{y}}_k{}^*\}$ via FFT, $P-1$ zeros should be padded at the end of $\mathbf{g}(\tilde{\mathbf{y}}_k)$ and $flip\{\tilde{\mathbf{y}}_k{}^*\}$. Then

$$\begin{aligned} & \mathrm{FFT}\{\mathbf{g}(\tilde{\mathbf{y}}_k) * flip\{\tilde{\mathbf{y}}_k{}^*\}\} \\ & = \mathrm{FFT}\{[\mathbf{g}(\tilde{\mathbf{y}}_k)^T, 0, \ldots 0]^T\} \odot \mathrm{FFT}\{[flip\{\tilde{\mathbf{y}}_k{}^*\}^T, 0, \ldots, 0]^T\} \end{aligned} \,. \tag{4.26}$$

In fact to compute the FFT of the convolution vectors needed for $\tilde{\mathbf{r}}_k$, we only need to compute $\text{FFT}\{[\mathbf{g}(\tilde{\mathbf{y}}_k)^T, 0, ..., 0]^T\}$ and $\text{FFT}\{[\tilde{\mathbf{y}}_k{}^T, 0, ..., 0]^T\}$. We can use the property of DFT to get

$$\text{FFT}_n\{[flip\{\tilde{\mathbf{y}}_k{}^*\}^T, 0, .., 0]^T\} = \text{FFT}_n\{[\tilde{\mathbf{y}}_k{}^T, 0, ..., 0]^T\}^* e^{-j\frac{2\pi(P-1)n}{2P-1}}, \qquad (4.27)$$

where the sub-index n means the n-th element of the DFT. A final IFFT will give a length $2P-1$ vector $\tilde{\mathbf{r}}_k$, which is the summation of three convolution vectors. With $\tilde{\mathbf{r}}_k$, $r_k(\tau)$ for $1-P \le \tau \le P-1$ can be obtained from the relation in (4.25).

Finally, let $\{r_k(\tau)\}$ be the sequence of $r_k(\tau)$ values defined for $-M \le \tau \le M$. With $\mathbf{\Psi}_k$ a Toeplitz matrix containing $\{r_k(\tau)\}$, the elements in $\mathbf{\Psi}_k \mathbf{w}_k$ can also be expressed as the linear convolution of $\{r_k(\tau)\}$ and $\mathbf{w}_k$, i.e.

$$\mathbf{\Psi}_k \mathbf{w}_k = flip\{\left(\{r_k(\tau)\} * flip\{\mathbf{w}_k\}\right)_{M+1:2M+1}\}. \qquad (4.28)$$

After padding M zeros at the end of $flip\{\mathbf{w}_k\}$, doing DFT of both $\{r_k(\tau)\}$ and padded $flip\{\mathbf{w}_k\}$, we can get the DFT of a modified version of $\{r_k(\tau)\} * flip\{\mathbf{w}_k\}$. Another inverse FFT with truncation would give the vector $\mathbf{\Psi}_k \mathbf{w}_k$.

The T-EASI with FFT algorithm is defined explicitly in Table 4-1. In the table, when zeros are padded they are always at the end of a vector.

Table 4-1 T-EASI with FFT

Algorithm: T-EASI algorithm with FFT
Initialization: Initialize $\mathbf{w}_0$. **Serial Updating:** *while* there is new data $\tilde{\mathbf{x}}_k$ 1. $\tilde{\mathbf{x}}_k$, $(M+P)$-point FFT. 2. $flip\{\mathbf{w}_k\}$, pad $P-1$ zeros, $(M+P)$-point FFT.

3. Get the DFT of $\tilde{\mathbf{x}}_k * \mathbf{w}_k$.
4. $(M+P)$-point IFFT, get length-P $\tilde{\mathbf{y}}_k$, then $\mathbf{g}(\tilde{\mathbf{y}}_k)$

5. $\tilde{\mathbf{y}}_k$, pad $P-1$ zeros, $2P-1$-point FFT
6. $\mathbf{g}(\tilde{\mathbf{y}}_k)$, pad $P-1$ zeros, $2P-1$-point FFT
7. Get the DFT of $[flip\{\tilde{\mathbf{y}}_k^*\}^T, 0, .., 0]^T$ and $[flip\{\mathbf{g}(\tilde{\mathbf{y}}_k)^*\}^T, 0, .., 0]^T$
8. Get the DFT of $\tilde{\mathbf{r}}_k$
9. $(2P-1)$-point IFFT to get $\tilde{\mathbf{r}}_k$ and $\{r_k(\tau)\}$

10. $\{r_k(\tau)\}$, $(2M+1)$-point FFT
11. $flip\{\mathbf{w}_k\}$, pad M zeros, $(2M+1)$-point FFT
12. DFT of elements in $\mathbf{\Psi}_k \mathbf{w}_k$
13. IDFT, truncate to get $\mathbf{\Psi}_k \mathbf{w}_k$

end while

The complexity in each step of T-EASI implemented with FFT is listed in Table 4-2 in Appendix 4B. With FFT implementation, the complexity of T-EASI is reduced from $O(M^2)$ to $O(M \log M)$.

4.5.2 Approximation of Cross-Correlation Terms

In the standard T-EASI in (4.22) or its implementation with FFT, the vectors $\tilde{\mathbf{y}}_k$ and $\mathbf{g}(\tilde{\mathbf{y}}_k)$ and all the cross-correlation parameters in $\mathbf{U}_k$ need to be updated to get $\{r_k(\tau)\}$ at each iteration. In fact, the outputs $\tilde{\mathbf{y}}_k$ at two adjacent iterations are not greatly different, so the outputs from the previous iteration can be used as an approximation of the ones in the current iteration.

In this part, we will give two methods of getting an approximation of $\{r_k(\tau)\}$, which will be used in the adaptation for equalizer vector. The simulation results that will be given in the next section show that using approximated $\{r_k(\tau)\}$ in our methods yield virtually the

same performance as the standard T-EASI, and the computational complexity will be reduced greatly. In the two methods of approximation, instead of updating the whole $\tilde{\mathbf{y}}_k$ and $\mathbf{g}(\tilde{\mathbf{y}}_k)$ vector, only one new $y(k)$ and $g(y(k))$ is computed. In this case, FFT is not necessarily needed to get the equalizer output.

With single output, the cross-correlation terms in the first row and the first column of $\mathbf{U}_k$ (the darker blue part in Fig. 4.4) can be obtained approximately using $y(k)$, $g(y(k))$ and the first $P-1$ elements in vectors $\mathbf{y}$ and $\mathbf{g}(\mathbf{y})$ from the previous iteration, with $2(2P-1)$ multiplications. After several iterations, the vectors $\mathbf{y}$ and $\mathbf{g}(\mathbf{y})$ at the k-th iteration become $\mathbf{y}_k = [y(k), y(k-1), ..., y(k-P+1)]^T$ and $\mathbf{g}(\mathbf{y}_k)$, where the elements in $\mathbf{y}_k$ are from different equalizers at the corresponding iterations. The two alternative ways of using these to update $\mathbf{U}_k$ are as follows.

a. Partially updated cross-correlations

Instead of updating all the outputs and the corresponding cross-correlation terms in $\mathbf{U}_k$, only parts of the elements are updated. In Fig. 4.5, the updating of $\mathbf{U}_k$ in this partially updating scheme is shown. In the figure, the light blue parts are from the previous iteration. At time k, when there is one new output, the cross correlations on the first row and column can be obtained with the current output and the outputs from the previous iteration. The other elements in $\mathbf{U}_k$ can be taken directly from the first $P-1$ rows and columns in $\mathbf{U}_{k-1}$. The figure in Fig. 4.5 gives an illustration of the relation between vector $\mathbf{y}_k$ and $\mathbf{U}_k$ with

partially updated approximation scheme; however, the elements $r_k(\tau)$ in the matrix $\mathbf{\Psi}_k$ can still be obtained efficiently with FFT directly from $\mathbf{y}_k$ and $\mathbf{g}(\mathbf{y}_k)$.

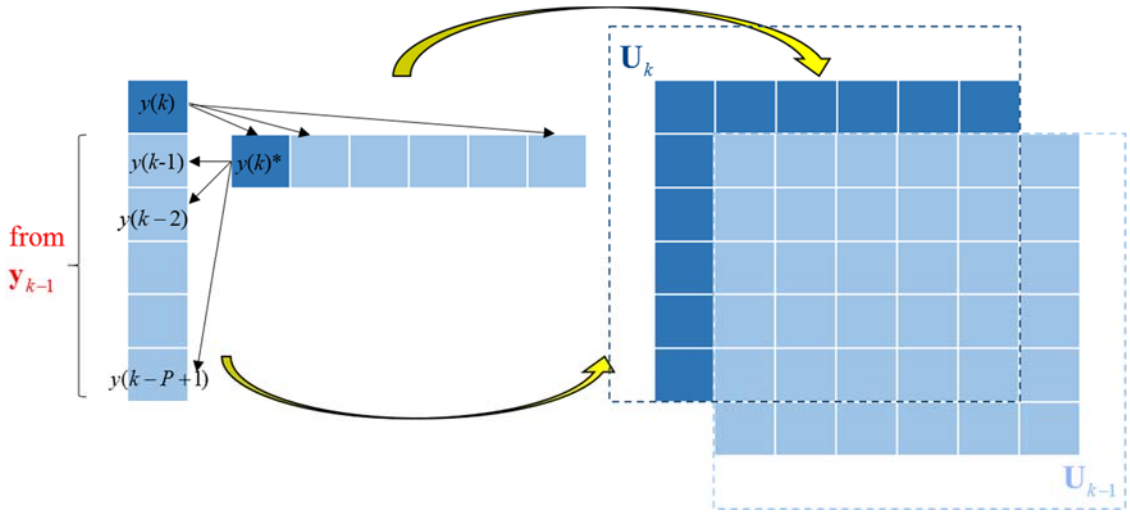

Fig. 4.5 Updating matrix $\mathbf{U}_k$ partially with current output

With the partially updated scheme, the computation needed for updating output vector is decreased; however, convolution is still needed to get the elements in $\mathbf{\Psi}_k$, which has computational cost of order $O(P\log P)$.

b. Instantaneous cross-correlations

Similar to the partially updating method, the new output $y(k)$ and the other $P-1$ outputs in $\mathbf{y}$ from the previous iteration are used to get the cross-correlation terms in the first row and first column of $\mathbf{U}_k$. These $2P-1$ include the cross-correlations with time lag from $1-P$ to $P-1$.

In the matrix $\mathbf{U}_k$ in the T-EASI algorithm, the cross-correlation terms on a certain diagonal have the same time lag. According to Fig. 4.4, the $\{r_k(\tau)\}$ terms in the matrix $\mathbf{\Psi}_k$ is the summation of the cross-correlation terms with the same time lag. For example, the terms on the main diagonal of $\mathbf{U}_k$ corresponds to the case when $\tau = 0$, there are P such terms, so $r_k(0)$ is the summation of P terms; the terms on the diagonal above the main one of $\mathbf{U}_k$ corresponds to the case when $\tau = -1$, there are $P-1$ such terms, so $r_k(-1)$ is the summation of $P-1$ terms. With the updated cross-correlation terms with time lag from $1-P$ to $P-1$, $\mathbf{U}_k$ can be considered as a Toeplitz matrix having these $2P-1$ terms in each of the descending diagonals. The elements $r_k(\tau)$ for $1-P \le \tau \le P-1$ in Toeplitz matrix $\mathbf{\Psi}_k$ can then be obtained by multiplying the instantaneous cross-correlation terms with the number of elements on the corresponding diagonal.

To get $\{r_k(\tau)\}$, the computational cost with the T-EASI in (4.22) and its FFT implementation is $O(P^2)$ and $O(P\log P)$, as can be seen from Appendix 4B. With both of the approximation schemes, the computational cost to get the equalizer outputs is $O(P)$ since single equalizer output is needed. With the first scheme, the computational complexity of getting $\{r_k(\tau)\}$ is $O(P\log P)$; while with the second one, the computational complexity of getting $\{r_k(\tau)\}$ is reduced to $O(P)$.

For the standard T-EASI and T-EASI with partially updated approximation method, instead of shifting one symbol along the channel output sequence each time, a number of symbols could be shifted at each iteration. While this may in some cases lead to reduced

computation cost for similar performance, we found from simulations that any improvement is rather limited and variable across different channels. Multiple-symbol shifting is not useful for T-EASI with instantaneous cross-correlations.

4.6 Simulations

The simulation results will be given for four cases: a long minimum phase FIR channel, a long non-minimum phase FIR channel, a short minimum phase FIR channel, and a short non-minimum phase FIR channel. The channels are selected by adjusting the positions of the zeros in a random way. For the short minimum phase channel, although the channel order is only $L = 4$, all the zeros are near the unit circle, which made it difficult to equalize the channel.

We simulated a sequence of i.i.d source symbols taken from a 64-QAM constellation transmitted continuously through the channels with AWGN and SNR=20dB. The source was normalized so that $E[\mathbf{s}\mathbf{s}^H] = \mathbf{I}$. In our simulations the nonlinearity in the T-EASI algorithm was chosen as a phase-preserving cubic function $g(y) = |y|^2 y$ unless otherwise specified.

The inter-symbol interference (ISI) of the cascaded system $\mathbf{c}$ was computed at each iteration to measure the performance, where the ISI is defined as

$$\text{ISI} = \sum_i \frac{|c_i|^2}{\max_i |c_i|^2} - 1 \ . \tag{4.29}$$

The ISI shown in the simulation results are in dB.

In the following part, we will give the simulation results for each of the four channels. For each channel, the impulse response and the zero-pole pattern of the channel are shown. The ISI performance as a function of iteration number is compared for our proposed T-EASI algorithm, the standard constant modulus algorithm (CMA) [3]–[5], multimodulus algorithm (MMA) [6], and square contour algorithm (SCA) [7]; both the result for a typical run and the average of 10 runs are shown. Our simulations also show how the choices of parameters such as the size of the equalizer output block P and the length of the equalizer M may affect the performance. For different channel cases, the simulation results are consistent, so we will explain in detail the first example, and give more explanations for the other cases when needed.

(a) Long minimum phase FIR channel

The impulse response of the channel is shown in Fig. 4.6, and the zero-pole pattern is shown in Fig. 4.7. From Fig. 4.8 and Fig. 4.9, it can be seen that the T-EASI algorithm yields faster convergence than the standard Bussgang-type algorithms. The T-EASI needs about 10^4 symbols for the algorithm to converge, while the Bussgang-type algorithms need about 4×10^4; also the ISI with T-EASI after convergence is slightly lower than the Bussgang-type algorithms. From Fig. 4.10, we see that when the value of P is reduced to $P=5$ for $M=30$, the performance of the T-EASI is worse than that of the standard CMA since not enough independence constrains are provided to update the equalizer. On the other hand for large P convergence may become slow. Based on simulations with different P, it turns out that $P=15$ gives the best performance. When P increases further, for example when $P=20$ and $P=25$, the convergence is slightly slower. There

is not much difference between $P = 20$ and $P = 25$. Experiments with other channels and different choices for P and M also indicated that $P \approx M/2$ is a good choice. In Fig. 4.11, the performance comparisons are given when the equalizer has different order M. It can be seen from the figure that when M is small, i.e. $M = 8$ or $M = 15$, the performance of T-EASI is worse than or comparable to that of the CMA. When M is large enough, i.e $M = 20$, the convergence speed of the T-EASI is faster than that of the CMA. As M increases further to $M = 30$, not only the convergence speed is faster, but also the ISI after convergence is lower than that of the CMA. Since in BE problem we tend to set the equalizer to be of a reasonable length (not too short), our T-EASI algorithm will be more likely to outperform the CMA algorithm.

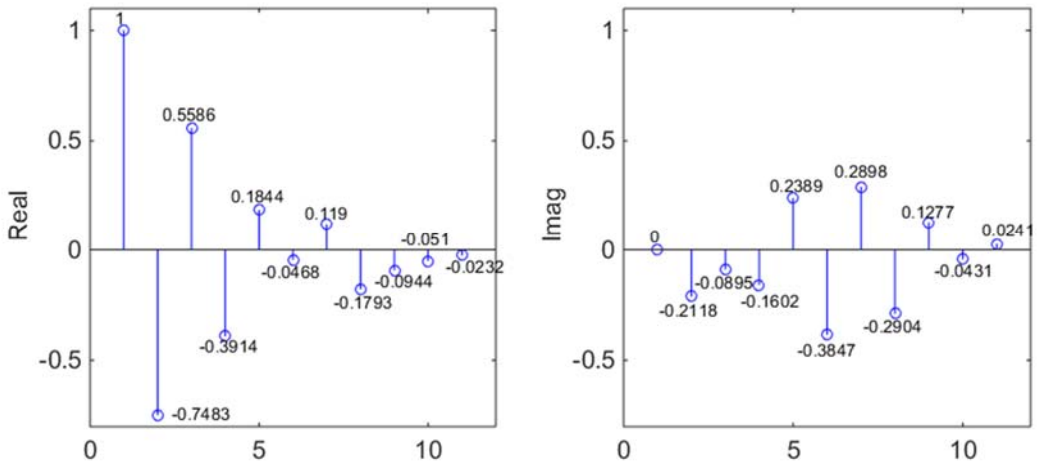

Fig. 4.6 Channel impulse response of long minimum phase channel

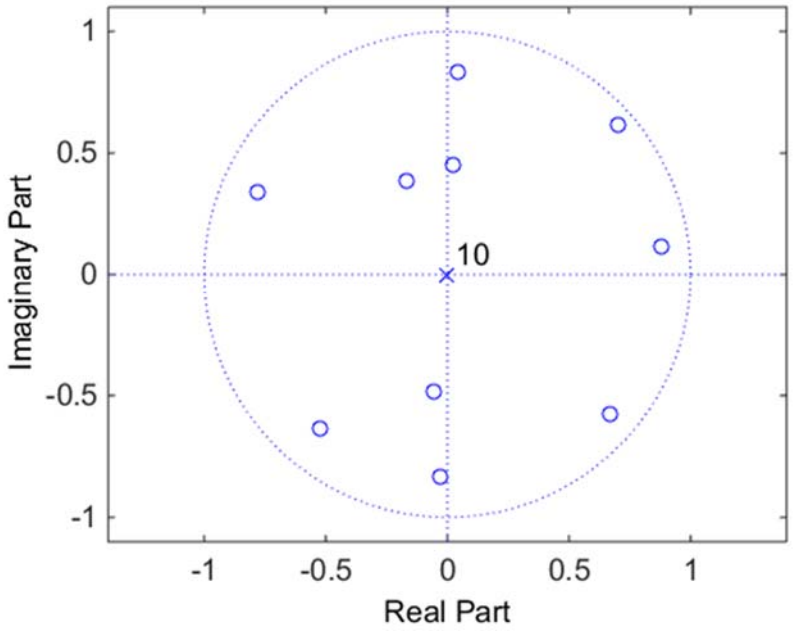

Fig. 4.7 Zero-pole pattern of long minimum phase channel

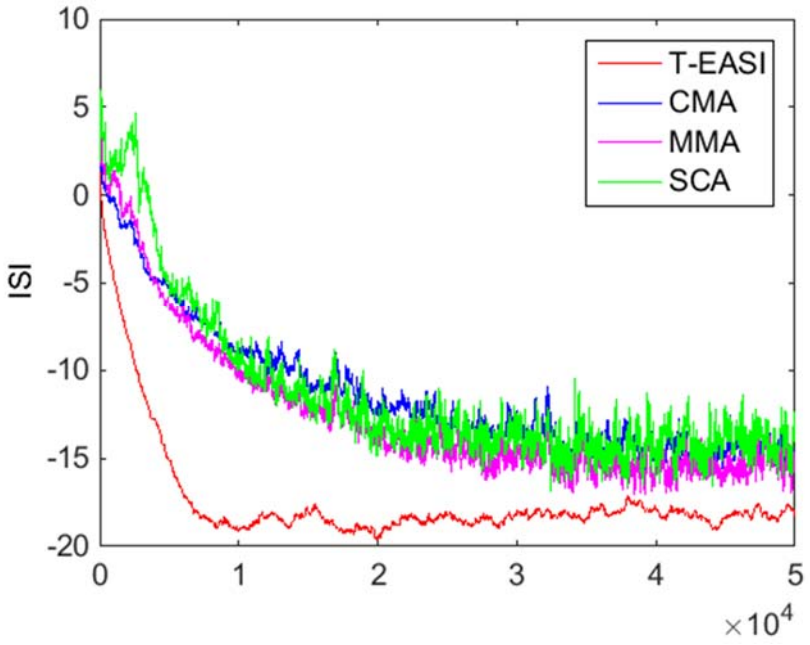

Fig. 4.8 ISI of the cascaded system for long minimum phase channel, 64-QAM, SNR=20dB. $P=15$, $M=30$.

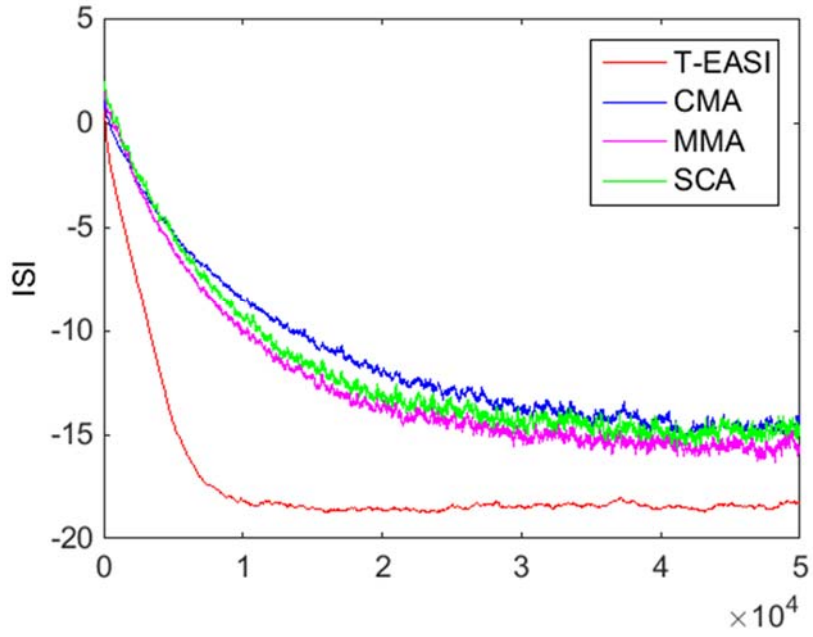

Fig. 4.9 Average over 10 runs of the ISI of the cascaded system for long minimum phase channel, 64-QAM, SNR=20dB. $P=15$, $M=30$.

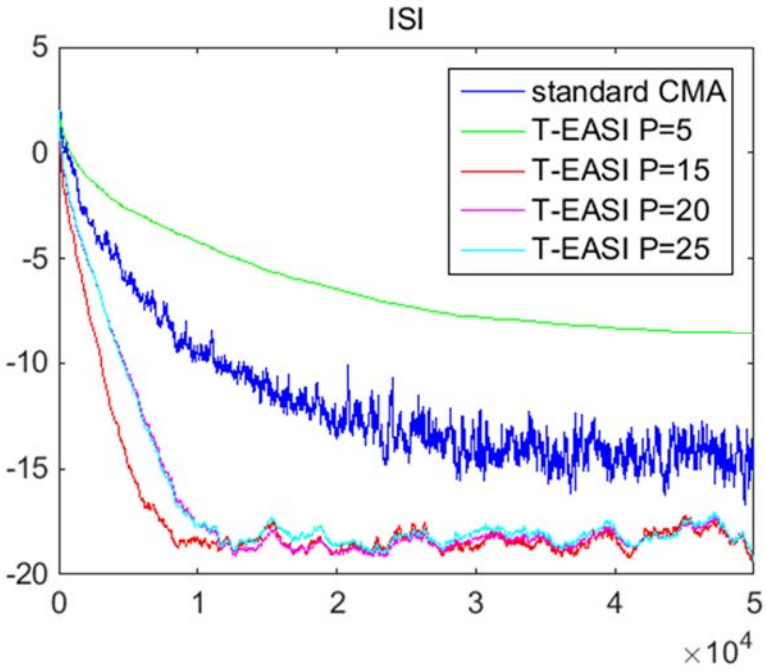

Fig. 4.10 ISI of the cascaded system for long minimum phase channel with different choices of P, 64-QAM, SNR=20dB. $M=30$.

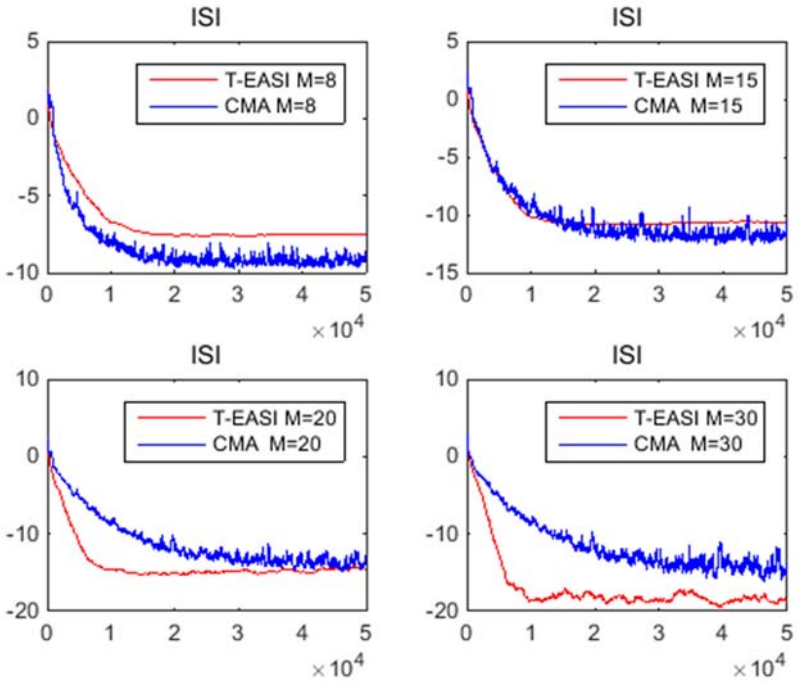

Fig. 4.11 ISI of the cascaded system for long minimum phase channel with different choices of M , $P = M/2$, 64-QAM, SNR=20dB.

(b) Long non-minimum phase FIR channel

The impulse response of the channel is shown in Fig. 4.12, and the zero-pole pattern is shown in Fig. 4.13.

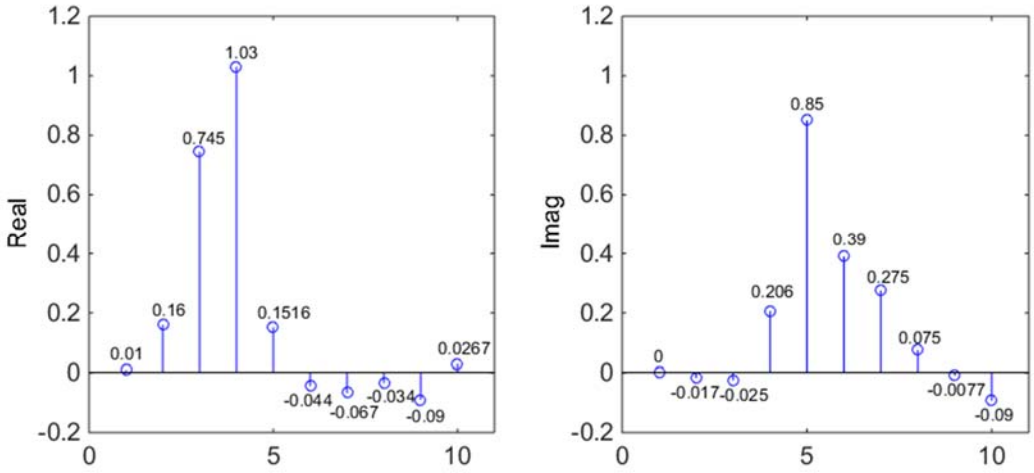

Fig. 4.12 Channel impulse response of long non-minimum phase channel

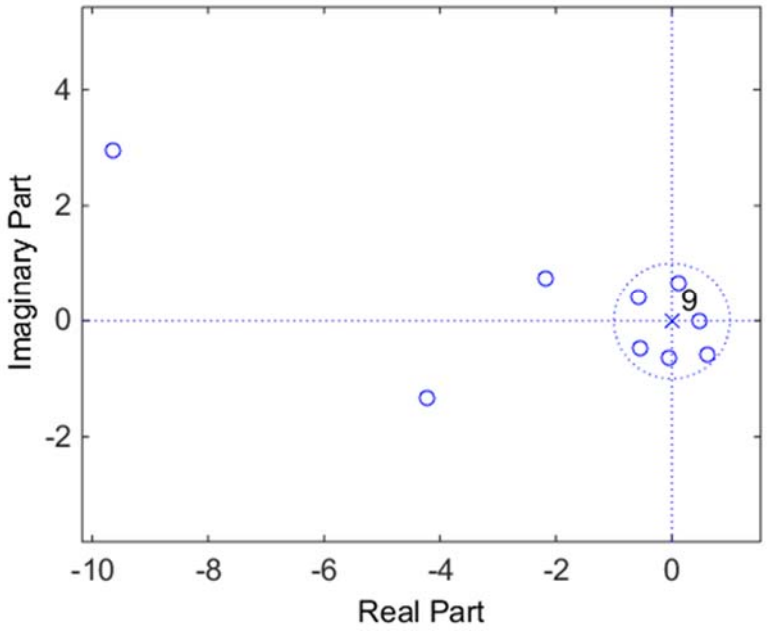

Fig. 4.13 Zero-pole pattern of long non-minimum phase channel

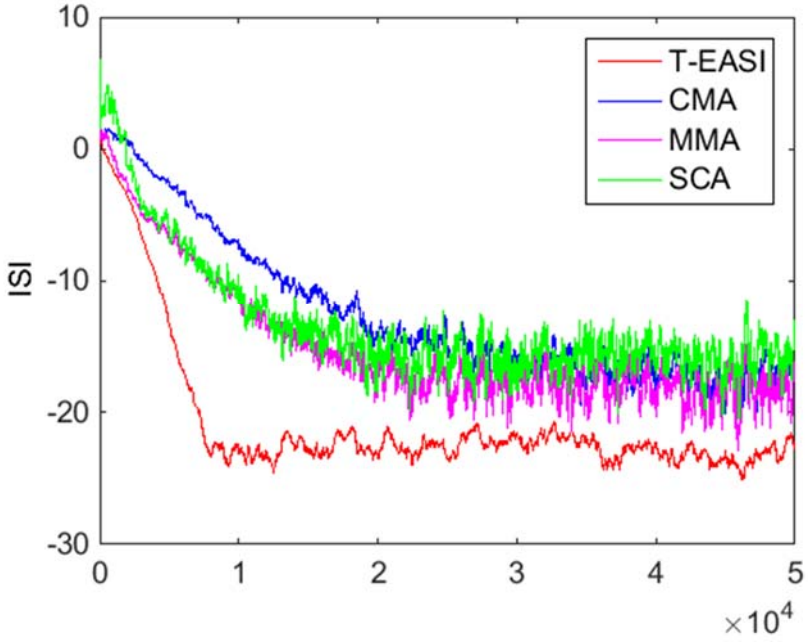

Fig. 4.14 ISI of the cascaded system for long non-minimum phase channel, 64-QAM, SNR=20dB. $P=10$, $M=20$.

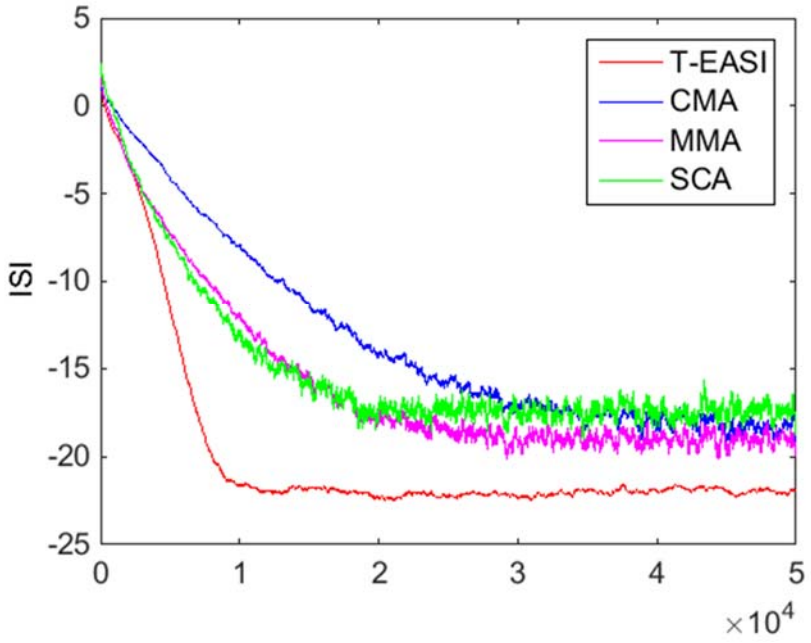

Fig. 4.15 Average over 10 runs of the ISI of the cascaded system for long non-minimum phase channel, 64-QAM, SNR=20dB. $P=10$, $M=20$.

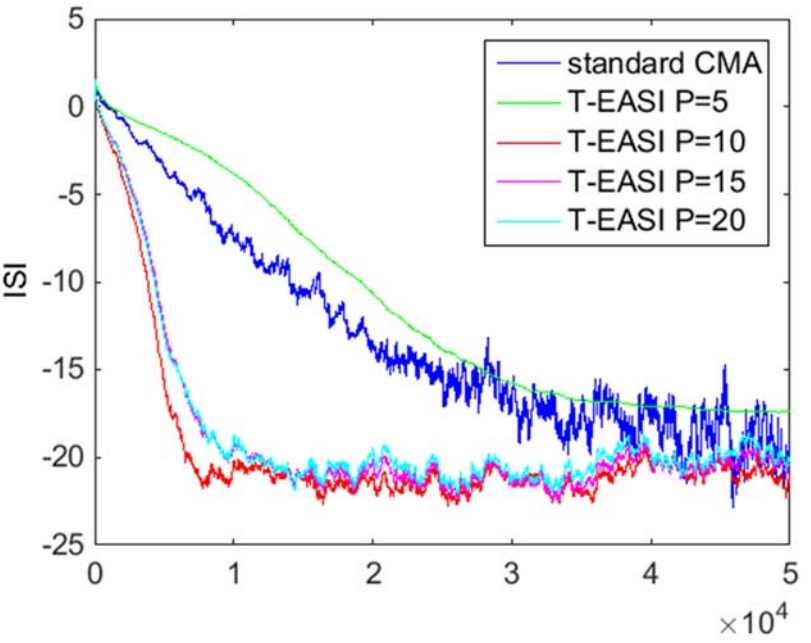

Fig. 4.16 ISI of the cascaded system for long non-minimum phase channel with different choices of P, 64-QAM, SNR=20dB. $M=20$.

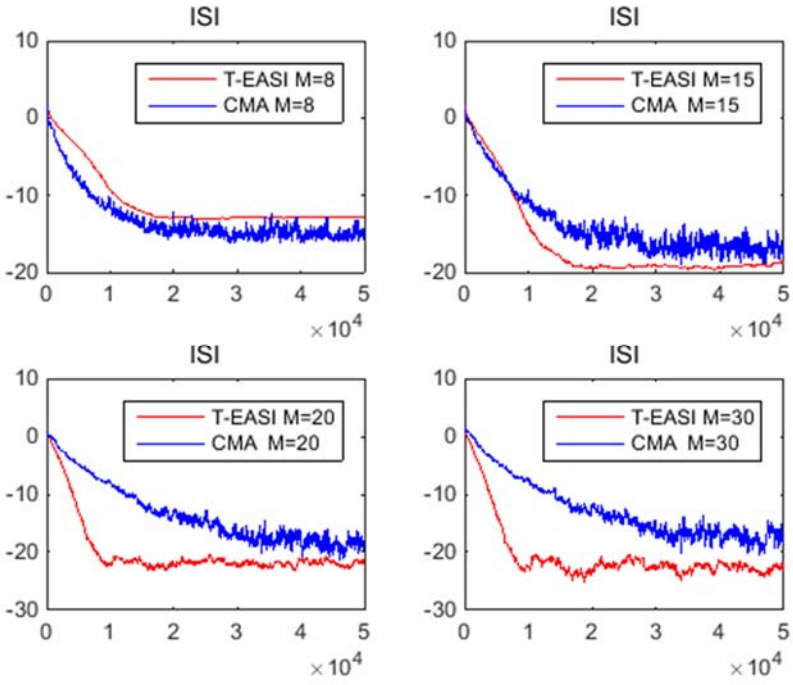

Fig. 4.17 ISI of the cascaded system for long non-minimum phase channel with different choices of M , $P = M / 2$, 64-QAM, SNR=20dB.

(c) Short minimum phase FIR channel

The impulse response of the channel is shown in Fig. 4.18, and the zero-pole pattern is shown in Fig. 4.19. From the impulse respnse, we can see that the zeros are near the unit circle, which makes it difficult to equalize.

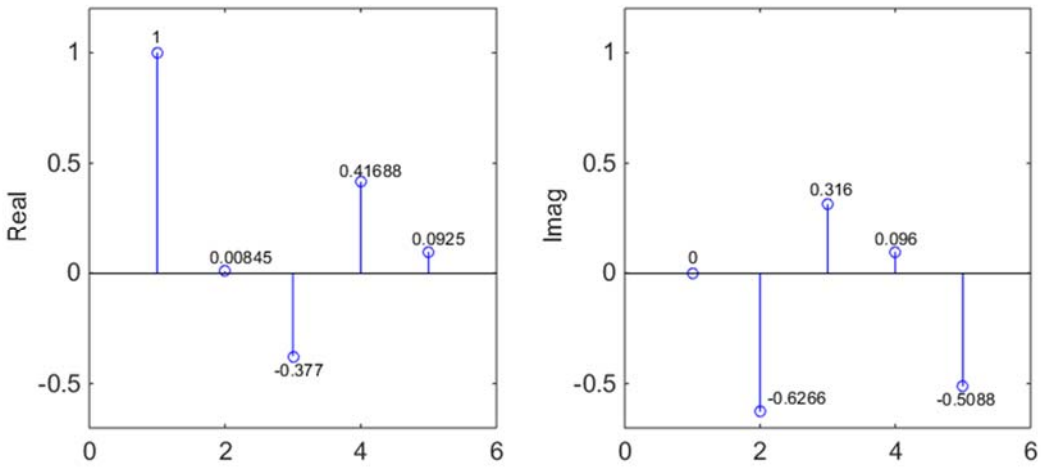

Fig. 4.18 Channel impulse response of short minimum phase channel

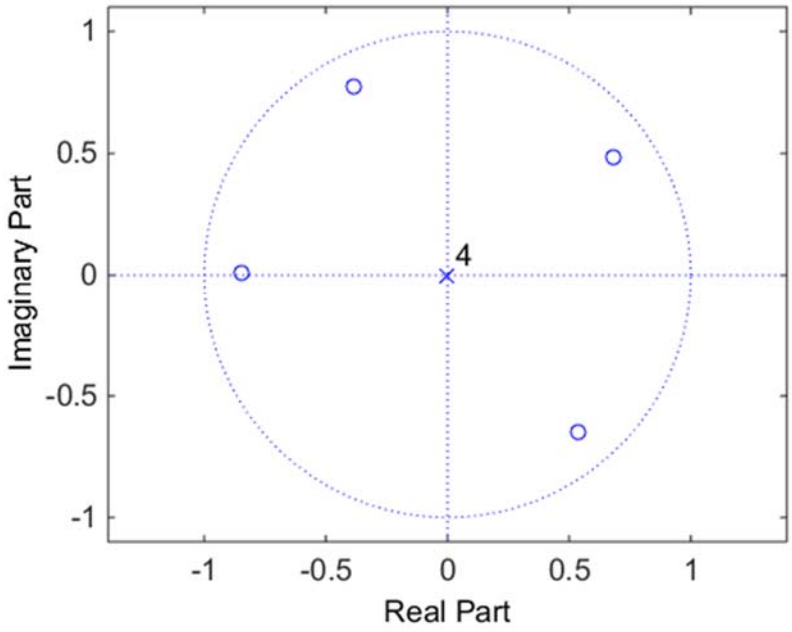

Fig. 4.19 Zero-pole pattern of short minimum phase channel

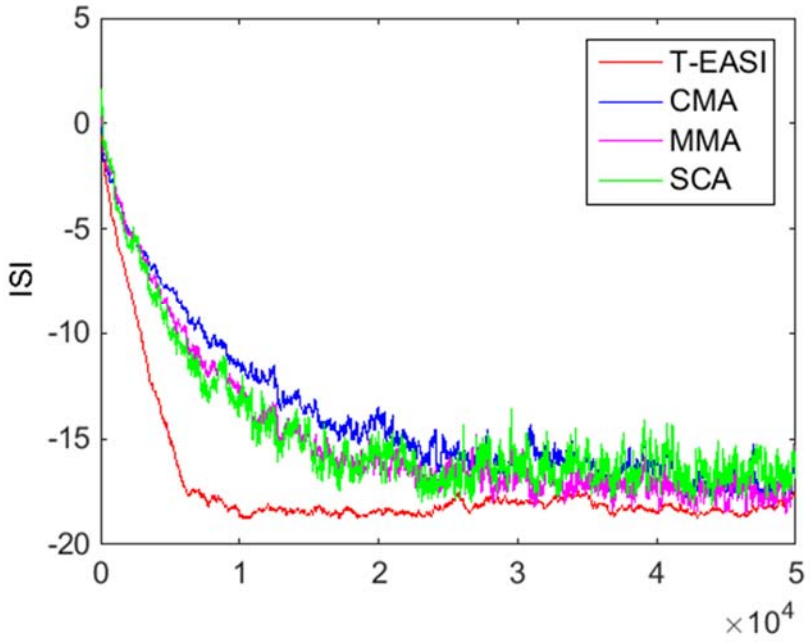

Fig. 4.20 ISI of the cascaded system for short minimum phase channel, 64-QAM, SNR=20dB. $P=10$, $M=20$.

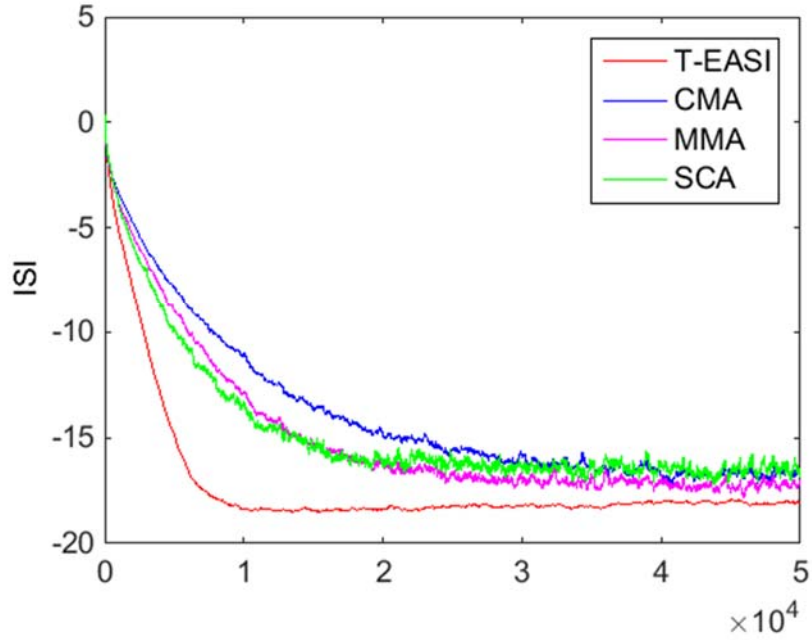

Fig. 4.21 Average over 10 runs of the ISI of the cascaded system for short minimum phase channel with different choices of *P*, 64-QAM, SNR=20dB. $M=20$.

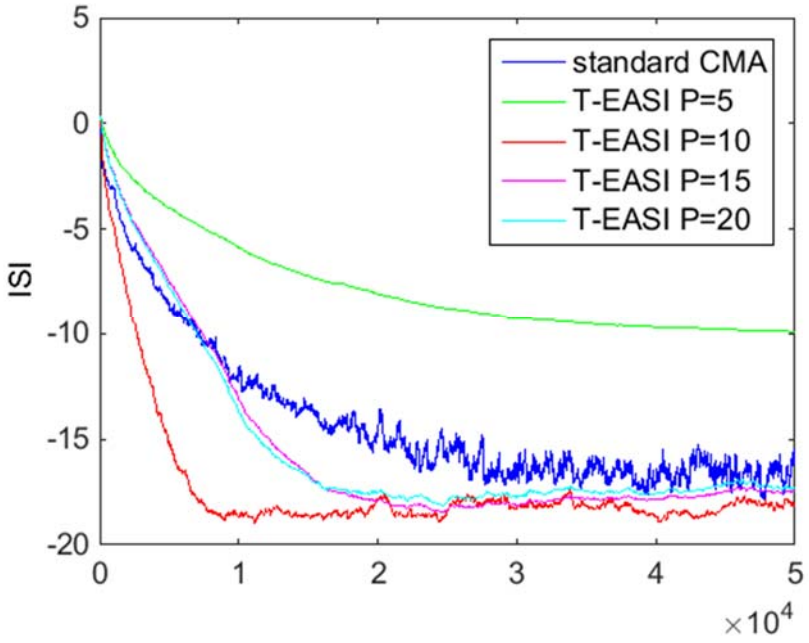

Fig. 4.22 ISI of the cascaded system for short minimum phase channel with different choices of P, 64-QAM, SNR=20dB. $M = 20$.

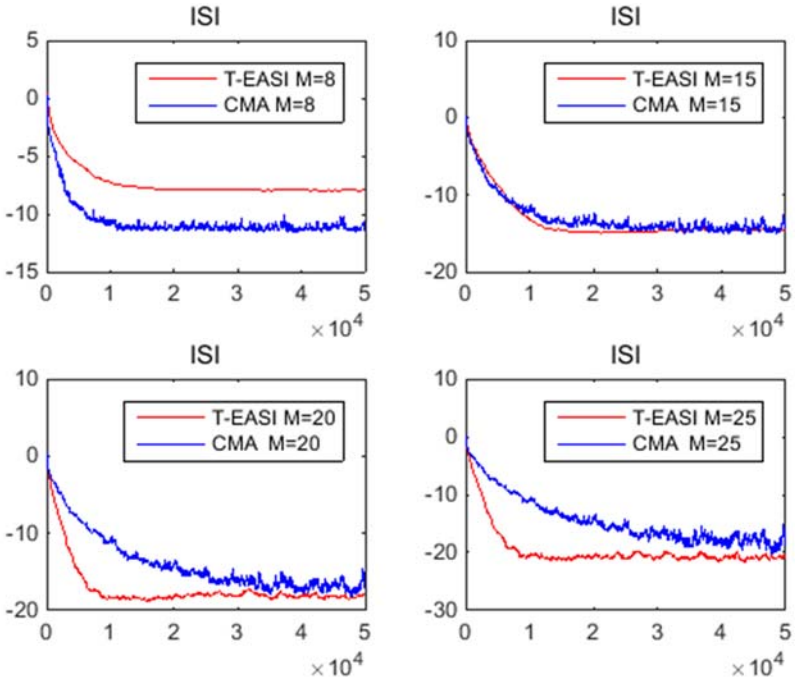

Fig. 4.23 ISI of the cascaded system for short minimum phase channel with different choices of M, $P = M/2$, 64-QAM, SNR=20dB.

(d) Short non-minimum phase FIR channel

The impulse response of the channel is shown in Fig. 4.24, and the zero-pole pattern is shown in Fig. 4.25.

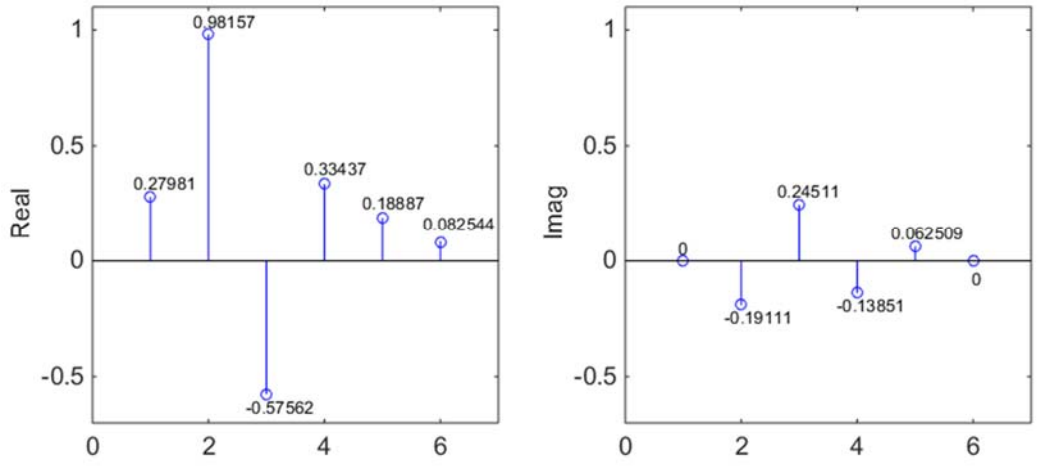

Fig. 4.24 Channel impulse response of short non-minimum phase channel

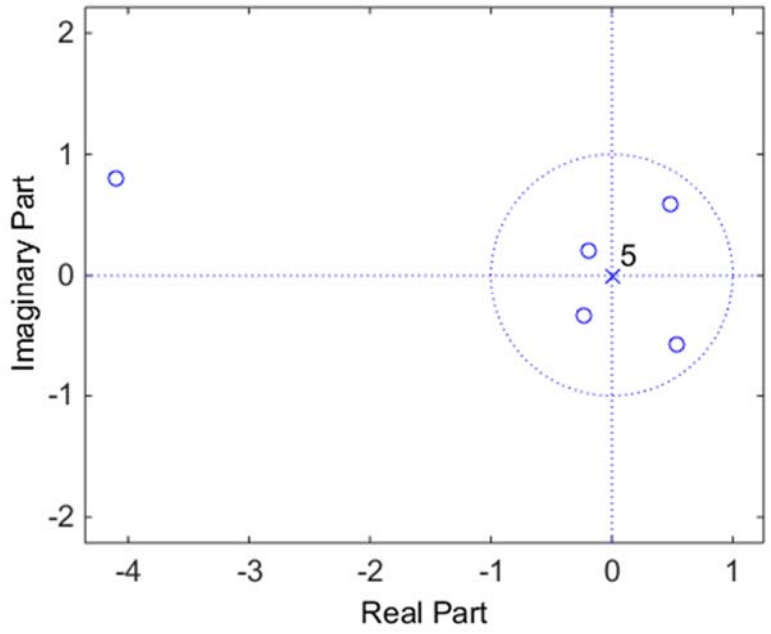

Fig. 4.25 Zero-pole pattern of short non-minimum phase channel

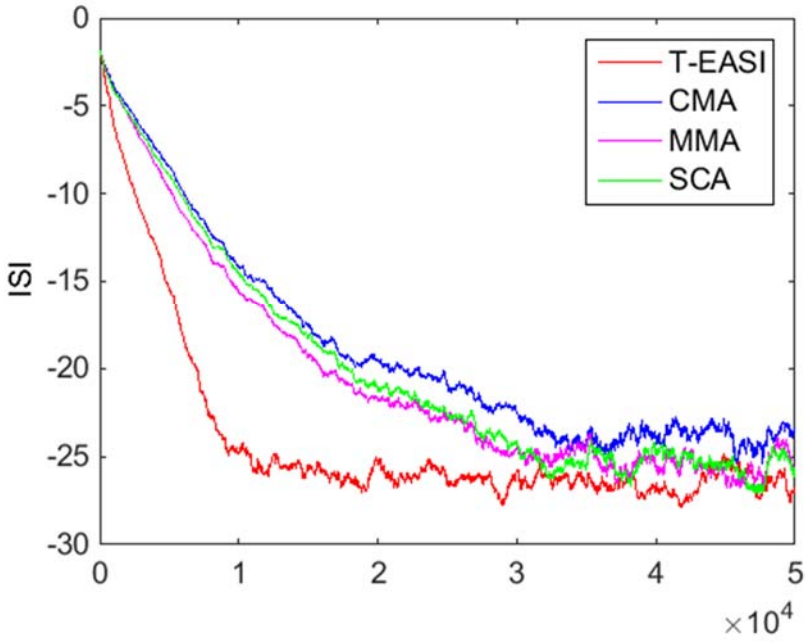

Fig. 4.26 ISI of the cascaded system for short non-minimum phase channel, 64-QAM, SNR=20dB. $P=10$, $M=20$.

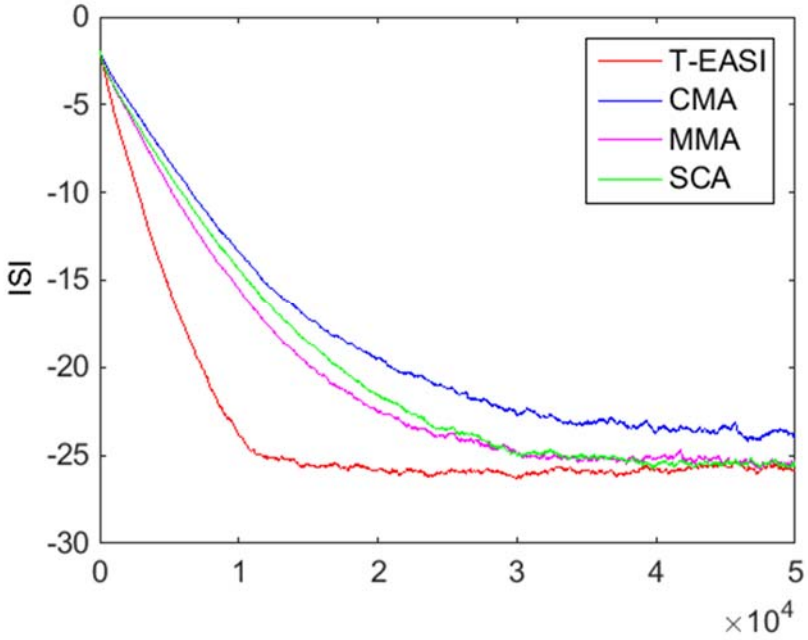

Fig. 4.27 Average over 10 runs of the ISI of the cascaded system for short non-minimum phase channel, 64-QAM, SNR=20dB. $P=10$, $M=20$.

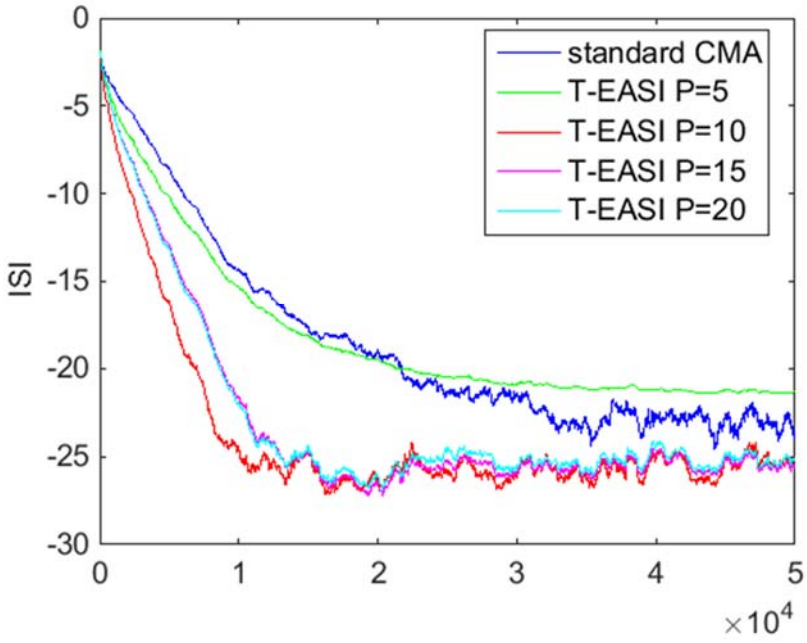

Fig. 4.28 ISI of the cascaded system for short non-minimum phase channel with different choices of P, 64-QAM, SNR=20dB. $M = 20$.

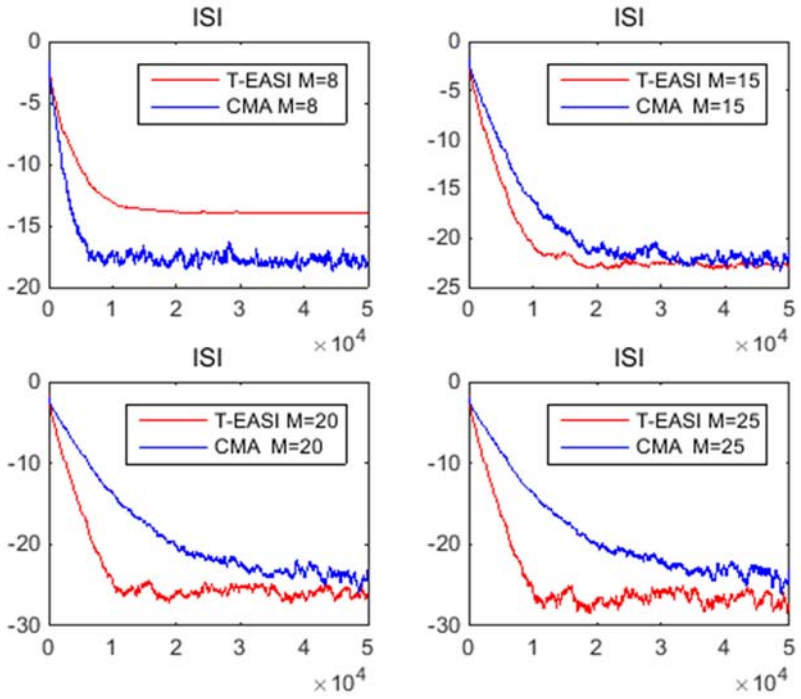

Fig. 4.29 ISI of the cascaded system for short non-minimum phase channel with different choices of M , $P = M/2$, 64-QAM, SNR=20dB.

We also performed simulations using the two schemes of approximation described in Section 4.5.2. Only the results for the short minimum phase channel will be shown in Fig. 4.30 as an example. From the simulation results, it can be seen that the adaptations with and without the approximation are virtually the same. If we magnify the curves, we can see from the small figure on the right that the three curves are not exactly overlapping, but the difference is very small. Simulation results were also done that showed a similar behavior for the other three channel cases.

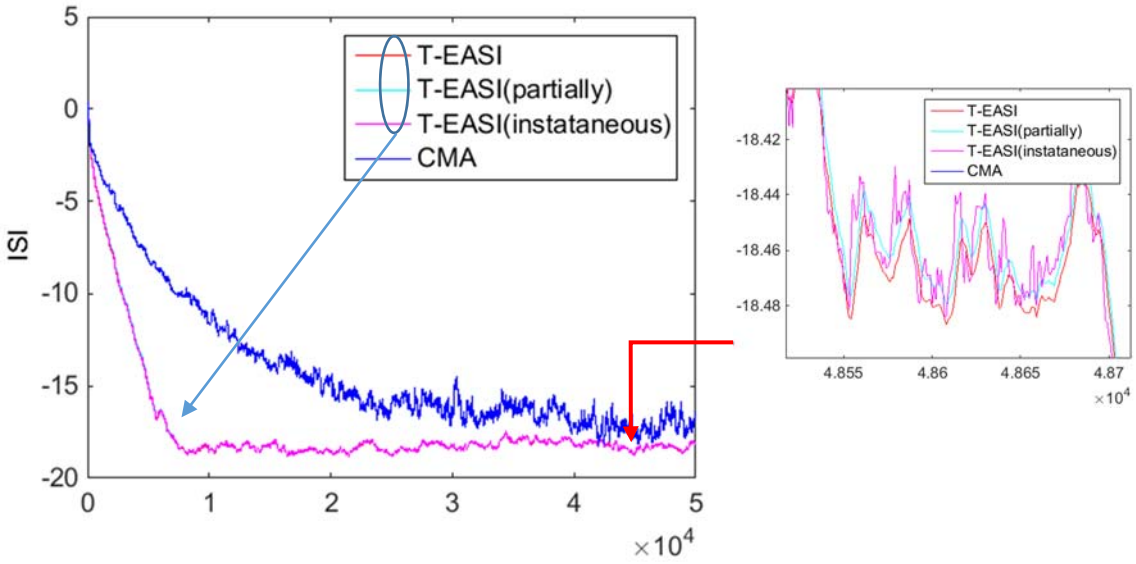

Fig. 4.30 ISI of the cascaded system for short minimum phase system with approximation, 64-QAM, SNR=20dB. $P=10$, $M=20$.

4.7 Phase Recovery

4.7.1 BE via T-EASI with I/Q Constraint

When the source has independent in-phase and quadrature parts, as in the case of standard QAM signaling, P complex source symbols can be seen as $2P$ mutually independent real symbols. In this case, if the source symbols are well recovered, the $2P$ real output symbols of the equalizer should be independent of each other. With independence of I/Q components as a constraint, phase recovery can be achieved [8], [9]. This constraint can be applied in our T-EASI scheme, as explained below and in [10].

For simplicity, we drop the sub-index *k* and the noise vector $\mathbf{v}_k$ in (4.1). Denote the in-phase and quadrature parts of the channel mixing matrix $\mathbf{H}$ by $\mathbf{H}_R$ and $\mathbf{H}_I$. The source vector and the channel output vector can also be expressed with their I/Q components, so that (4.1) can be written (without noise) as

$$\begin{bmatrix} \tilde{\mathbf{x}}_R \\ \tilde{\mathbf{x}}_I \end{bmatrix} = \begin{bmatrix} \mathbf{H}_R & -\mathbf{H}_I \\ \mathbf{H}_I & \mathbf{H}_R \end{bmatrix} \begin{bmatrix} \mathbf{s}_R \\ \mathbf{s}_I \end{bmatrix} . \tag{4.30}$$

With a similar definition for the separating matrix and the equalizer outputs, we have

$$\begin{bmatrix} \tilde{\mathbf{y}}_R \\ \tilde{\mathbf{y}}_I \end{bmatrix} = \begin{bmatrix} \mathbf{W}_R & -\mathbf{W}_I \\ \mathbf{W}_I & \mathbf{W}_R \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}_R \\ \tilde{\mathbf{x}}_I \end{bmatrix} . \tag{4.31}$$

Defining $\bar{\mathbf{H}} \triangleq \begin{bmatrix} \mathbf{H}_R & -\mathbf{H}_I \\ \mathbf{H}_I & \mathbf{H}_R \end{bmatrix}$, $\bar{\mathbf{W}} \triangleq \begin{bmatrix} \mathbf{W}_R & -\mathbf{W}_I \\ \mathbf{W}_I & \mathbf{W}_R \end{bmatrix}$, $\bar{\mathbf{x}} \triangleq \begin{bmatrix} \tilde{\mathbf{x}}_R \\ \tilde{\mathbf{x}}_I \end{bmatrix}$ and $\bar{\mathbf{y}} \triangleq \begin{bmatrix} \tilde{\mathbf{y}}_R \\ \tilde{\mathbf{y}}_I \end{bmatrix}$, we have

$$\bar{\mathbf{W}}\bar{\mathbf{H}} = \begin{bmatrix} \mathbf{W}_R\mathbf{H}_R - \mathbf{W}_I\mathbf{H}_I & -\mathbf{W}_R\mathbf{H}_I - \mathbf{W}_I\mathbf{H}_R \\ \mathbf{W}_R\mathbf{H}_I + \mathbf{W}_I\mathbf{H}_R & \mathbf{W}_R\mathbf{H}_R - \mathbf{W}_I\mathbf{H}_I \end{bmatrix} , \tag{4.32}$$

$$\bar{\mathbf{y}} = \bar{\mathbf{W}}\bar{\mathbf{x}} . \tag{4.33}$$

At convergence, if $\bar{\mathbf{W}}$ is a good "separating matrix", it is expected that

$$\mathbf{W}_R\mathbf{H}_R - \mathbf{W}_I\mathbf{H}_I = \pm\mathbf{C}^{ideal}e^{-j\theta} \tag{4.34}$$

$$\mathbf{W}_R\mathbf{H}_I + \mathbf{W}_I\mathbf{H}_R = \mathbf{0} \tag{4.35}$$

or

$$\mathbf{W}_R\mathbf{H}_R - \mathbf{W}_I\mathbf{H}_I = 0 \tag{4.36}$$

$$\mathbf{W}_R\mathbf{H}_I + \mathbf{W}_I\mathbf{H}_R = \pm\mathbf{C}^{ideal}e^{-j\theta} \tag{4.37}$$

where $\mathbf{C}^{ideal}$ is defined in (4.5) as the Toeplitz matrix containing the ideal response of the cascaded channel-equalizer system. With these equations, the constellation of the source can be recovered up to a multiple of $\pi/2$ ambiguity.

The T-EASI algorithm with this further I/Q independence constraint (I/Q-T-EASI) can be used to update matrix $\bar{\mathbf{W}}$. Specially, matrix $\bar{\mathbf{W}}$ is updated first with the EASI algorithm as in (4.7) with adaptations

$$\bar{\mathbf{W}}_{k+1} = \bar{\mathbf{W}}_k - \lambda\left[\bar{\mathbf{y}}_k\bar{\mathbf{y}}_k^H - \mathbf{I} + \mathbf{g}(\bar{\mathbf{y}}_k)\bar{\mathbf{y}}_k^H - \bar{\mathbf{y}}_k\mathbf{g}(\bar{\mathbf{y}}_k)^H\right]\bar{\mathbf{W}}_k \ . \tag{4.38}$$

Then the structure constraint that $\bar{\mathbf{W}}$ has identical diagonal blocks and off-diagonal blocks that are sign-inverted versions of each other as in (4.31) is imposed. In addition, the Toeplitz structure is forced on the sub-blocks $\mathbf{W}_R$ and $\mathbf{W}_I$.

As in the case of the T-EASI algorithm, the matrix adaptation can be simplified for the I/Q components of the equalizer vector. Dropping the sub-index *k*, and denoting the I/Q components of $\mathbf{w}$ as $\mathbf{w}_R$ and $\mathbf{w}_I$, it can be seen that the adaptation of $\mathbf{w}_R$ and $\mathbf{w}_I$ can be written as

$$\mathbf{w}_R \leftarrow \mathbf{w}_R - \frac{\lambda}{2P}\left[(\mathbf{\Psi}_{RR} + \mathbf{\Psi}_{II})\mathbf{w}_R + (\mathbf{\Psi}_{RI} - \mathbf{\Psi}_{IR})\mathbf{w}_I\right] \tag{4.39}$$

$$\mathbf{w}_I \leftarrow \mathbf{w}_I - \frac{\lambda}{2P}\left[(\mathbf{\Psi}_{IR} - \mathbf{\Psi}_{RI})\mathbf{w}_R + (\mathbf{\Psi}_{II} + \mathbf{\Psi}_{RR})\mathbf{w}_I\right] \tag{4.40}$$

where the $\mathbf{\Psi}$ matrix with sub-index contains the cross-correlation terms from the corresponding $\mathbf{U}$ matrix. Specifically, the $\mathbf{U}$ matrices have the expression as follows:

$$\mathbf{U}_{RI} = \tilde{\mathbf{y}}_R \tilde{\mathbf{y}}_I^H + \mathbf{g}(\tilde{\mathbf{y}}_R)\tilde{\mathbf{y}}_I^H - \tilde{\mathbf{y}}_I \mathbf{g}(\tilde{\mathbf{y}}_R)^H, \tag{4.41}$$

$$\mathbf{U}_{IR} = \tilde{\mathbf{y}}_I \tilde{\mathbf{y}}_R^H + \mathbf{g}(\tilde{\mathbf{y}}_I)\tilde{\mathbf{y}}_R^H - \tilde{\mathbf{y}}_R \mathbf{g}(\tilde{\mathbf{y}}_I)^H, \tag{4.42}$$

$$\mathbf{U}_{RR} = \tilde{\mathbf{y}}_R \tilde{\mathbf{y}}_R^H - \mathbf{I} + \mathbf{g}(\tilde{\mathbf{y}}_R)\tilde{\mathbf{y}}_R^H - \tilde{\mathbf{y}}_R \mathbf{g}(\tilde{\mathbf{y}}_R)^H, \tag{4.43}$$

$$\mathbf{U}_{II} = \tilde{\mathbf{y}}_I \tilde{\mathbf{y}}_I^H - \mathbf{I} + \mathbf{g}(\tilde{\mathbf{y}}_I)\tilde{\mathbf{y}}_I^H - \tilde{\mathbf{y}}_I \mathbf{g}(\tilde{\mathbf{y}}_I)^H. \tag{4.44}$$

The $\mathbf{\Psi}$ matrix can be obtained from the $\mathbf{U}$ matrix in a similar way as shown in Fig. 4.4.

4.7.2 Reducing Phase Ambiguity with Hard-limiting

Phase "recovery" can also be achieved without imposing the I/Q independence constraint if we choose an appropriate nonlinearity $g(\cdot)$ for our ICA-based algorithm. Similarly, here "recovery" does not guarantee exact phase recovery, but reduces the phase ambiguity to a multiple of $\pi/2$.

For the EASI algorithm, we are sometimes interested in reducing computational complexity by using simple, coarsely quantized versions of some "optimal" nonlinearity. One of the simplest approximations in our case would be a phase-preserving hard limiter, i.e

$$g(y) = \alpha \frac{y}{|y|}, \tag{4.45}$$

where α is a scale constant. The hard-limiting also provides additional robustness compared to nonlinearities with unbounded characteristics.

According to the stability analysis in [1], we can prove that for *second-order circular* signal satisfying $E\left[s(k)^2\right]=0$, such as QAM, α should be negative in (4.45) to ensure stability of the algorithm. A simple selection is $\alpha=-1$, and this results in

$$g(y)=-\frac{y}{|y|}. \tag{4.46}$$

Based on (4.46), we can additionally impart a phase quantization to make the nonlinearity even simpler. The resulting nonlinearity is the quad-phase version of the hard limiter, i.e.

$$g(y)=-\text{csgn}(y), \tag{4.47}$$

where $\text{csgn}(y)=\text{sign}(y_R)+j\text{sign}(y_I)$ for complex symbol $y=y_R+jy_I$. This nonlinearity quantizes the output symbol to one of the four representative points in each quadrant (with a π phase shift), and reduces the phase ambiguity to $\pi/2$, which is characteristic of the *generalized Sato algorithm* (GSA) [3] . For this non-phase-preserving nonlinearity, we cannot derive explicitly the stability conditions, but simulation results show that this amplitude and phase quantizing nonlinearity gives quite good results for QAM signal.

4.7.3 Simulations

For the different channels in our channel examples given in Section 4.6, the simulation results are consistent for the algorithms of this section. To illustrate the performance of the

algorithms for phase recovery, we therefore only show the simulation results for the long non-minimum phase channel.

In this sub-section, the SNR is 35dB, which is different from the 20dB set in Section 4.6; the final ISI gets lower so that the constellations of the recovered symbols can be better observed. From Fig. 4.31, it can be seen that with quad-phase hard-limiting in the T-EASI, the performance is comparable with that of the I/Q-T-EASI, which is slightly better than the T-EASI with cubic nonlinearity. In the figure, the red curve for T-EASI and the purple curve for T-EASI with approximation are almost overlapping, as has been shown in the previous simulations. From the constellations in Fig. 4.32 and Fig. 4.33, it can be seen that T-EASI is subject to phase ambiguity, as in the CMA. In comparison, when hard-limiting nonlinearity or I/Q independence constraint is used, phase recovery can be achieved. Since the source symbols are from 64QAM and the channel is subject to noise, there is still ISI in the recovered constellation.

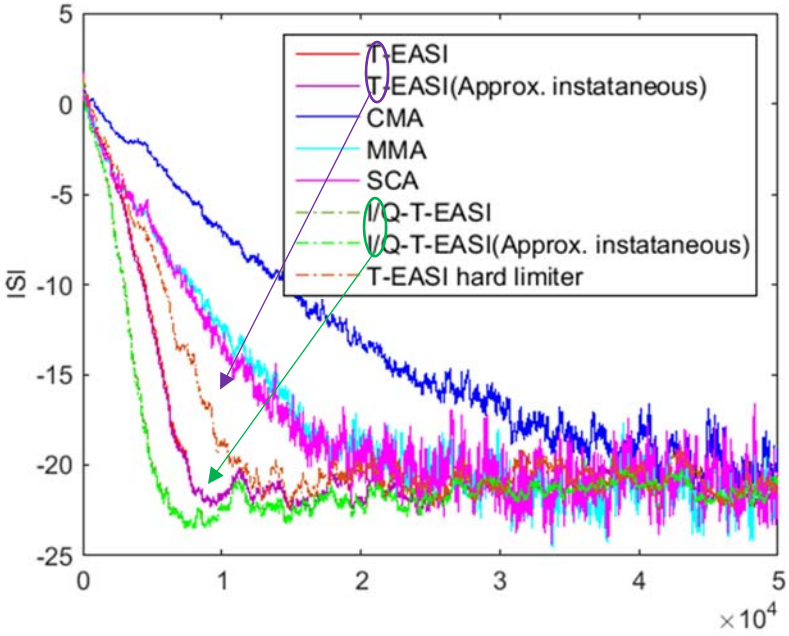

Fig. 4.31 ISI of the cascaded system for long non-minimum phase channel with phase recovery, 64-QAM, SNR=35dB. $P=10$, $M=20$.

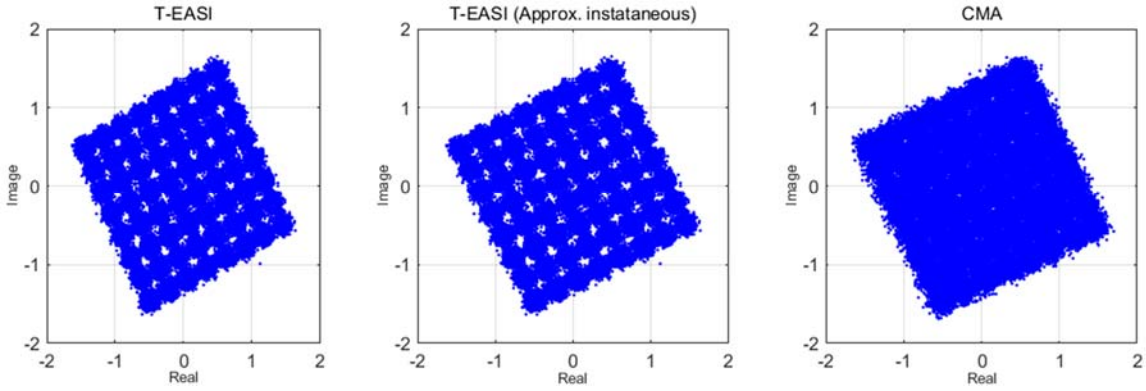

Fig. 4.32 Constellation for recovered symbols, without phase recovery, long non-minimum phase channel, 64-QAM, SNR=35dB. $P=10$, $M=20$.

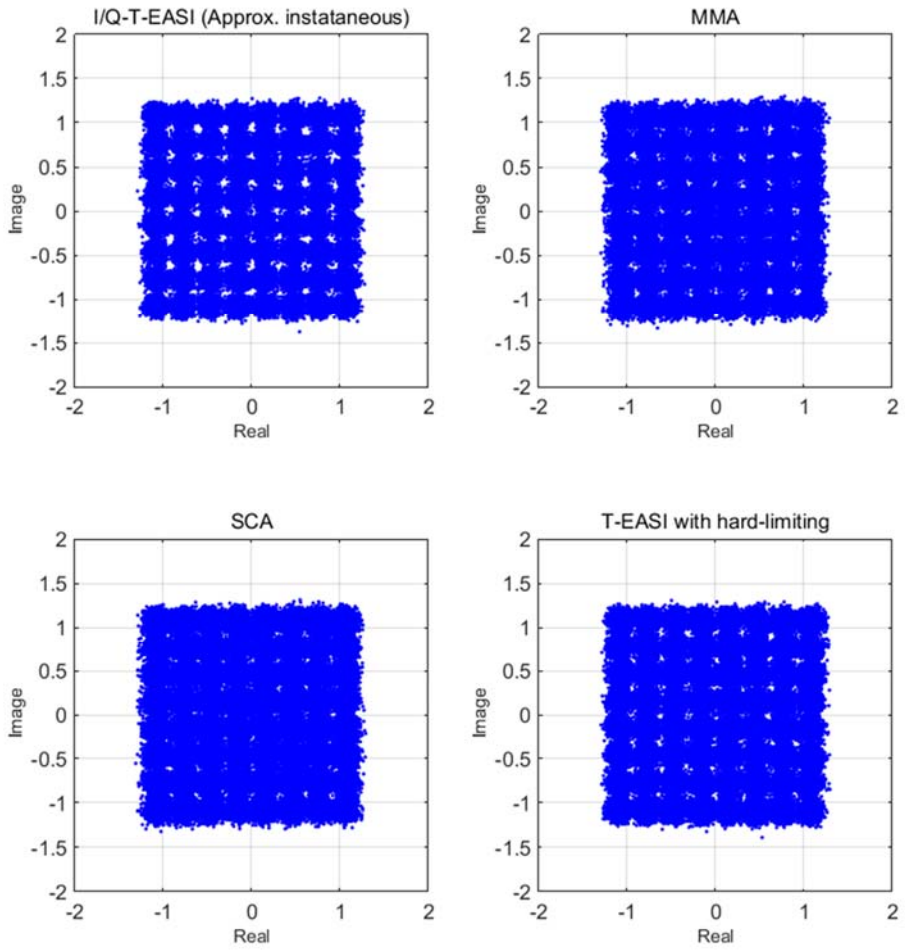

Fig. 4.33 Constellation for recovered symbols, with phase recovery, long non-minimum phase channel, 64-QAM, SNR=35dB. $P=10$, $M=20$.

4.8 Other ICA-Based Algorithms with Toeplitz Constraint

In addition to the well-known EASI algorithm, there are many adaptive ICA based algorithms for BSS. These algorithms can also be used with Toeplitz constraint for our BE

problem. A well-known algorithm that has been used widely was proposed by Amari [11]. The algorithm was based on *natural gradient*, with the adaptation

$$\mathbf{W}_{k+1} = \mathbf{W}_k - \lambda \left[\mathbf{g}(\tilde{\mathbf{y}}_k) \tilde{\mathbf{y}}_k{}^H - \mathbf{I} \right] \mathbf{W}_k \ . \tag{4.48}$$

Compared with the EASI algorithm, the algorithm in (4.48) does not have an explicit whitening part. We call the adaptation of (4.48) with Toeplitz constraint the T-Amari algorithm.

To illustrate the capability of this algorithm, and to demonstrate in general the fact that different ICA algorithms may be employed for use in our blind equalization problem, we will give simulation results for the short minimum phase channel and long non-minimum phase channel with SNR=20dB. From both the simulation results in Fig. 4.34 and Fig. 4.35, which is average of 10 runs, we can see that the Amari algorithm with Toeplitz constraint gives faster convergence speed than the standard Bussgang-type algorithms. Similar to the T-EASI case, the performance with I/Q independence constraint is slightly faster than that without the I/Q independence constraint. In addition, to reduce computational complexity, the two approximation schemes of Section 4.5.2 can be used. For T-Amari and IQ-T-Amari algorithms, approximation schemes give virtually the same results as their corresponding algorithms without approximation. The curves for the approximation schemes are not shown in Fig. 4.34 and Fig. 4.35.

In Fig. 4.36 and Fig. 4.37, the performance of the T-EASI and T-Amari is compared for the long non-minimum phase channel and short minimum phase channel, with $\mathrm{SNR} = 20\mathrm{dB}$. From the figures, it can be seen that for the channels and cubic nonlinearity used in our examples, the convergence speed of the T-EASI algorithm is about two times

faster than the T-Amari's algorithm. However, since the relative change in the T-EASI algorithm contains more terms and includes whitening, the computational cost of the T-EASI algorithm is somewhat higher. In fact, the performance of the ICA-based algorithms is impacted by the choice of nonlinearity and also the channel characteristics, and it is possible that for other channels with other choices of nonlinearity, the T-Amari could equal the T-EASI algorithm in terms of convergence speed. In general, the T-Amari algorithm is a simpler algorithm for blind equalization based on independence that yields faster convergence than the standard Bussgang-type algorithms, and is a good choice when the source symbols are statistically independent of each other.

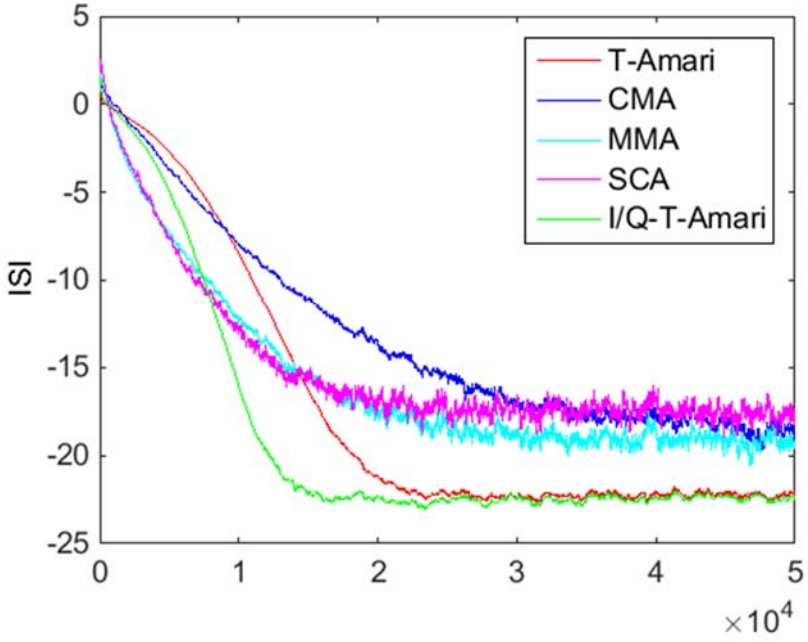

Fig. 4.34 T-Amari algorithm for long non-minimum phase channel, 64-QAM, SNR = 20dB. $P = 10$, $M = 20$.

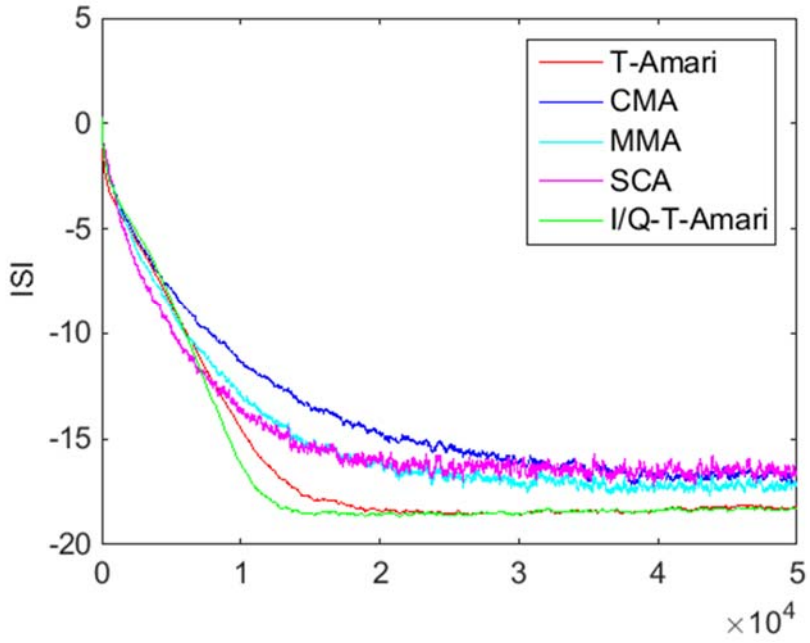

Fig. 4.35 T-Amari algorithm for short minimum phase channel, 64-QAM, $\mathrm{SNR} = 20\mathrm{dB}$. $P = 10$, $M = 20$.

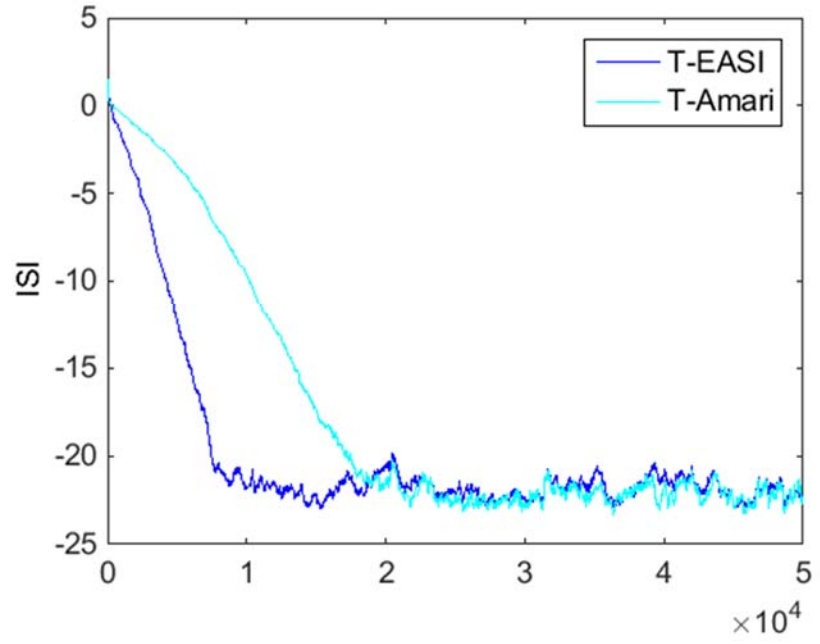

Fig. 4.36 Comparison of T-EASI and T-Amari algorithm for long non-minimum phase channel, 64-QAM, $\mathrm{SNR} = 20\mathrm{dB}$. $P = 10$, $M = 20$.

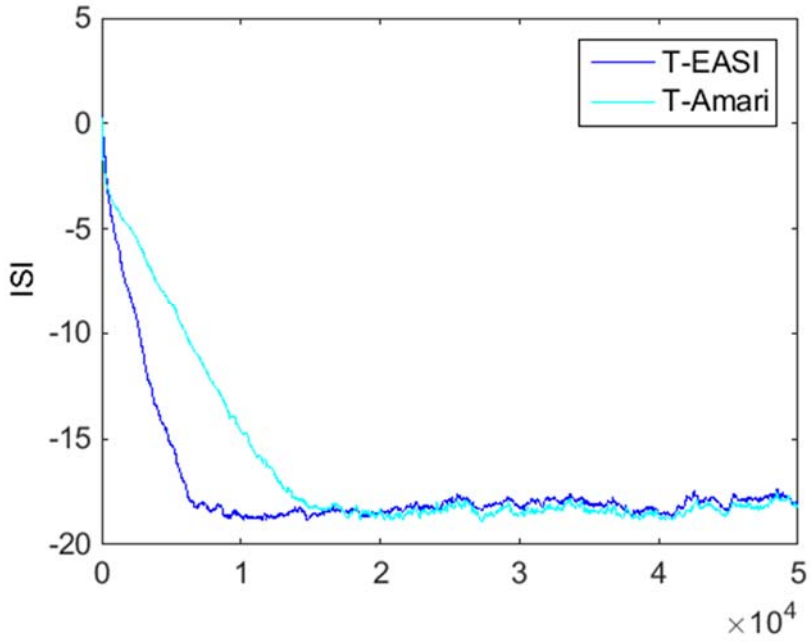

Fig. 4.37 Comparison of T-EASI and T-Amari algorithm for short minimum phase channel, 64-QAM, SNR = 20dB. $P = 10$, $M = 20$.

4.9 Conclusions

In this chapter, by extending the idea for BE with block transmission schemes of the previous chapter, we formulated the symbol-rate sampling BE as an underdetermined BSS problem. ICA-based algorithms with Toeplitz constraint were developed to exploit the independence between symbols. The algorithms can be written as an adaptation for the equalizer vector. In addition, with FFT implementation and approximation for cross-correlation terms, the computational complexity is reduced. Simulation results showed that even though the proposed algorithms have a somewhat higher computational cost than the standard BE algorithms, the performance gains obtained are significant.

References

[1] J. F. Cardoso and B. H. Laheld, "Equivariant adaptive source separation," *IEEE Trans. Signal Process.*, vol. 44, no. 12, pp. 3017–3030, 1996.

[2] Z. Wu and S. A. Kassam, "Blind Equalization Based On Blind Separation with Toeplitz Constraint," in *Signals, Systems and Computers, 2014 48th Asilomar Conference on*, 2014, pp. 1453–1457.

[3] D. N. Godard, "Self-Recovering Equalization and Carrier Tracking in Two-Dimensional Data Communication Systems," *IEEE Trans. Commun.*, vol. 28, no. 11, pp. 1867–1875, 1980.

[4] J. Treichler and B. G. Agee, "A new approach to multipath correction of constant modulus signals," *IEEE Trans. Acoust.*, vol. 31, no. 2, pp. 459–472, 1983.

[5] J. R. Treichler and M. G. Larimore, "New Processing Techniques Based on the Constant Modulus Adaptive Algorithm," *IEEE Trans. Acoust.*, vol. 33, no. 2, pp. 420–431, 1985.

[6] J. Yang, J.-J. Werner, G. A. Dumont, and Others, "The multimodulus blind equalization and its generalized algorithms," *IEEE J. Sel. Areas Commun.*, vol. 20, pp. 997–1015, 2002.

[7] T. Thaiupathump, L. He, and S. A. Kassam, "Square contour algorithm for blind equalization of QAM signals," *Signal Processing*, vol. 86, pp. 3357–337–, 2006.

[8] T. Thaiupathump, "New algorithms for blind equalization and blind source separation/phase recovery," University of Pennsylvania, 2002.

[9] L. He and S. Thaiupathump, Trasapong Kassam, "Blind separation of complex I/Q independent sources with phase recovery," *Signal Process. Lett. IEEE*, vol. 12, no. 5, pp. 419–422, 2005.

[10] Z. Wu and S. A. Kassam, "Symbol-Rate Blind Equalization based on Constrained Blind Separation," in *Information Sciences and Systems (CISS), 2015 49th Annual Conference on*, 2015, no. 4, pp. 1–6.

[11] S. Amari, A. Cichocki, and H. Yang, "A new learning algorithm for blind signal separation," *Adv. Neural Inf. Process. Syst.*, vol. 8, pp. 757–763, 1996.

Appendix 4A

Suppose $\mathbf{W}$ is a matrix of size $P \times (P+M)$, and its (i, j)-th element is denoted as $\mathbf{W}_{ij}$ in this appendix for clarity of expression[2]. It is forced to be Toeplitz by taking average along the descending diagonals and forcing the upper right and lower left corners to be zero. Let $\mathbf{W}_{Toe}$ be the resulting matrix. Here we will show that

$\mathbf{W}_{Toe}$ is the orthogonal projection of matrix $\mathbf{W}$ onto the space of Toeplitz matrix with upper right and lower left corners zero.

The *inner product* of two complex matrix $\mathbf{X}$ and $\mathbf{Y}$ can be defined as $\langle \mathbf{X}, \mathbf{Y} \rangle = \text{Trace}\{\mathbf{Y}^H \mathbf{X}\}$. To prove the above statement, we need to show that

$$\langle \mathbf{W} - \mathbf{W}_{Toe}, \mathbf{W}_{Toe} \rangle = 0$$

Proof:

$$\begin{aligned}
&\langle \mathbf{W} - \mathbf{W}_{Toe}, \mathbf{W}_{Toe} \rangle \\
&= \text{Trace}\{\mathbf{W}_{Toe}^H (\mathbf{W} - \mathbf{W}_{Toe})\} \\
&= \sum_{j=1}^{P+M} \left(\mathbf{W}_{Toe}^H (\mathbf{W} - \mathbf{W}_{Toe}) \right)_{jj} \\
&= \sum_{j=1}^{P+M} \sum_{i=1}^{P} (\mathbf{W}_{Toe}^H)_{ji} (\mathbf{W} - \mathbf{W}_{Toe})_{ij} \\
&= \sum_{j=1}^{P+M} \sum_{i=1}^{P} (\mathbf{W}_{Toe}^*)_{ij} (\mathbf{W} - \mathbf{W}_{Toe})_{ij} \\
&= \sum_{j=1}^{P+M} \sum_{i=1}^{P} (\mathbf{W}_{Toe}^*)_{ij} \left(\mathbf{W}_{ij} - (\mathbf{W}_{Toe})_{ij} \right) \\
&= \sum_{j=1}^{P+M} \sum_{i=1}^{P} \left[(\mathbf{W}_{Toe}^*)_{ij} \mathbf{W}_{ij} - |(\mathbf{W}_{Toe})_{ij}|^2 \right]
\end{aligned}$$

[2] This notation will be used exclusively in the appendix. In the previous sections, the sub-index of $\mathbf{W}$ is used to indicate iteration number.

Matrix $\mathbf{W}_{Toe}$ has Toeplitz structure with upper right and lower left elements zero, and contains shifted vector in each row, thus $(\mathbf{W}_{Toe})_{ij} = 0$ for $j - i < 0$ and $j - i > M$. Suppose $\mathbf{W}_{Toe}$ has the following form:

$$\mathbf{W}_{Toe} = \begin{bmatrix} w(0) & w(1) & \dots & w(M) & 0 & \dots & 0 \\ 0 & w(0) & w(1) & \dots & w(M) & 0 & \dots & 0 \\ & & & \dots & & & \\ 0 & \dots & 0 & w(0) & w(1) & \dots & w(M) \end{bmatrix},$$

then

$$(\mathbf{W}_{Toe})_{ij} = \begin{cases} w(j-i) & 0 \le j-i \le M \\ 0 & \text{otherwise} \end{cases}$$

Letting $\tau = j - i$, we have

$$\begin{aligned} &\langle \mathbf{W} - \mathbf{W}_{Toe}, \mathbf{W}_{Toe} \rangle \\ &= \sum_{\tau=0}^{M} \sum_{i=1}^{P} \left[(\mathbf{W}_{Toe}^{*})_{i,i+\tau} \mathbf{W}_{i,i+\tau} - | (\mathbf{W}_{Toe})_{i,i+\tau} |^2 \right] \\ &= \sum_{\tau=0}^{M} \sum_{i=1}^{P} \left[w(\tau)^{*} \mathbf{W}_{i,i+\tau} - | w(\tau)^{*} |^2 \right] \\ &= \sum_{\tau=0}^{M} \left[\sum_{i=1}^{P} w(\tau)^{*} \mathbf{W}_{i,i+\tau} - P | w(\tau)^{*} |^2 \right] \end{aligned}$$

For a fixed τ, $w(\tau)$ is equal to the average of the τ-th descending diagonal, thus $\sum_{i=1}^{P} \mathbf{W}_{i,i+\tau} = P w(\tau)$. As a result,

$$\begin{aligned} &\sum_{\tau=0}^{M} \left[\sum_{i=1}^{P} w(\tau)^{*} \mathbf{W}_{i,i+\tau} - P | w(\tau)^{*} |^2 \right] \\ &= \sum_{\tau=0}^{M} \left[P w(\tau)^{*} w(\tau) - P | w(\tau)^{*} |^2 \right] \\ &= 0 \end{aligned}$$

Thus $\langle \mathbf{W} - \mathbf{W}_{Toe}, \mathbf{W}_{Toe} \rangle = 0$ and $\mathbf{W}_{Toe}$ is the orthogonal projection of matrix $\mathbf{W}$ onto the space of Toeplitz matrix with upper right and lower left elements zero.

Appendix 4B

The computational complexity of the T-EASI implemented with FFT is listed in the table.

Table 4-2 Computational complexity of T-EASI with FFT implementation

Algorithm: T-EASI with FFT implementation	**Addition**	**Multiplication**
Initialization: Initialize $\mathbf{w}_0$. **Serial Updating:** *while* there is new data $\mathbf{x}_k$		
1. $\mathbf{x}_k$, $(M+P)$-point FFT.	$(M+P)\log(M+P)/2$	$(M+P)\log(M+P)/2$
2. $flip\{\mathbf{w}_k\}$, pad $P-1$ zeros, $(M+P)$-point FFT.	$(M+P)\log(M+P)/2$	$(M+P)\log(M+P)/2$
3. Get the DFT of $\mathbf{x}_k * \mathbf{w}_k$.		$M+P$
4. $(M+P)$-point IFFT, get length-P $\tilde{\mathbf{y}}_k$, then $\mathbf{g}(\tilde{\mathbf{y}}_k)$	$(M+P)\log(M+P)/2$	$(M+P)\log(M+P)/2$
5. Pad $P-1$ zeros at the end of $\mathbf{y}_k$, $(2P-1)$-point FFT	$(2P-1)\log(2P-1)/2$	$(2P-1)\log(2P-1)/2$
6. Pad $P-1$ zeros at the end of $\mathbf{g}(\mathbf{y}_k)$, $(2P-1)$-point FFT	$(2P-1)\log(2P-1)/2$	$(2P-1)\log(2P-1)/2$
7. Get the DFT of $[flip(\mathbf{y}_k^*);0;...;0]^T$ and $[flip(\mathbf{g}(\mathbf{y}_k)^*);0;...;0]^T$		$2(2P-1)$
8. DFT of $\tilde{\mathbf{r}}_k$	$2(2P-1)$	$3(2P-1)$
9. $(2P-1)$-point IFFT to get $\tilde{\mathbf{r}}_k$ and $r_k(\tau)$	$(2P-1)\log(2P-1)/2+1$	$(2P-1)\log(2P-1)/2$
10. $\{r_k(\tau)\}$, $(2M+1)$-point FFT	$(2M+1)\log(2M+1)/2$	$(2M+1)\log(2M+1)/2$
11. $flip\{\mathbf{w}_k\}$, pad M zeros, $(2M+1)$-point FFT	$(2M+1)\log(2M+1)/2$	$(2M+1)\log(2M+1)/2$
12. DFT of elements in $\mathbf{\Psi}_k\mathbf{w}_k$		$(2M+1)$
13. IDFT, truncate to get the change $\mathbf{\Psi}_k\mathbf{w}_k$ *End while*	$(2M+1)\log(2M+1)/2$	$(2M+1)\log(2M+1)/2$

Chapter 5

Bussgang-Type Blind Equalization Algorithms Based On Relative Gradient

5.1 Introduction

We have explained in Chapter 2 that in blind equalization (BE) statistical or structural properties of payload data can be used for finding the equalizer. A group of algorithms called Bussgang-type algorithms have been widely used to achieve BE. These algorithms define a cost function based on signaling constellation properties and use standard stochastic gradient descent method to do the equalizer adaptation. In the Bussgang-type algorithms, a single equalizer output is generated at each iteration.

In this chapter we propose to modify the Bussgang-type algorithms by using the relative gradient (RG) instead of the standard gradient (SG) formulation. A block of outputs are used each time to update the matrix that contains the coefficients of the equalizer vector. Using the RG and forcing Toeplitz structure helps speed up convergence. Unlike independent component analysis (ICA)-based BE algorithms using the RG, independence of source symbols is not required for our RG Bussgang equalizers. Our proposed algorithms yield faster convergence compared to standard Bussgang-type BE algorithms. In Section 5.2, the Bussgang-type algorithms are reviewed. The concept of the relative gradient, as well as of the natural gradient which is closely related to the RG, is explained in Section 5.3. In Section 5.4, the application of the RG in the vector adaptation for the Bussgang-type algorithms is shown. Although vector adaptation with the RG does not have good performance, the idea of the RG is useful when a block of equalizer outputs are considered. In Section 5.5, the block version of the Bussgang-type algorithm with the SG is shown. Its relation to the standard Bussgang-type algorithms is also explained. The idea of using block of equalizer outputs for adaptation leads to the block version of relative-gradient Bussgang algorithms in Section 5.6. The application of the RG in processing multiple equalizer outputs exploits the Bussgang condition at the steady state to help adaptation, and speeds up convergence. In Section 5.7 schemes to reduce computational complexity are discussed. Simulation results are shown in Section 5.8, and the connection between block RG Bussgang algorithms and the ICA-based algorithms for BE is discussed in Section 5.9.

5.2 Review of Bussgang-Type Algorithms

In this section, we review the standard stochastic-gradient algorithms for blind equalization (BE). In BE a training sequence to help the equalizer coefficients adaptation is not available. Well-known BE techniques start from a cost function that the equalizer attempts to minimize. The cost function is in general the expected value of some function of the equalizer output, which is designed based on *a priori* knowledge of the nature of the finite-alphabet signaling constellation.

Generally, we may express the cost function as $J(\mathbf{w}) = E[G(y(k))]$, which is a function of $\mathbf{w} = [w(0), w(1), ..., w(M)]^T$ by virtue of the expression

$$y(k) = \sum_{m=0}^{M} w(m)x(k-m) .$$

An adaptive equalizer may then be based on *gradient descent*, with the adaptation for the equalizer vector given as

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mu \nabla_{\mathbf{w}^*} J(\mathbf{w}_k) , \tag{5.1}$$

where μ is the step-size, $\mathbf{w}_k$ is the vector at the k-th iteration, and $\nabla_{\mathbf{w}^*} J(\mathbf{w}_k)$ is the *gradient*, or more explicitly the *standard gradient* (SG). It represents the direction of change in $\mathbf{w}_k$ for the maximum rate of change of the cost function. Note that for a complex vector, the standard gradient is the gradient of the function with respect to the conjugate of the vector. For details of this standard derivation, refer to Appendix 5A. When the algorithm (5.1) converges to the steady state, the gradient approximates zero, i.e.

$$\mathbf{w}_{k+1} \approx \mathbf{w}_k \Leftrightarrow \nabla_{\mathbf{w}^*} J(\mathbf{w}_k) \approx \mathbf{0} \tag{5.2}$$

In conventional equalization with a sequence of training symbols, one popular approach is to use the *least mean square* (LMS) equalizer. In this case, the cost function is defined as the mean of square of the difference between the actual and desired output of the equalizer, also called *mean square error* (MSE), i.e.

$$J(\mathbf{w}) = E[|\, y(k) - s(k-d)\,|^2]\,, \tag{5.3}$$

where d is the delay of the channel-equalizer system. For the equalizer iterative adaptation, the expectation is replaced by its instantaneous value in the LMS algorithm, so that the adaptation for the equalizer vector becomes

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mu(y(k) - s(k-d))\mathbf{x}_k^*\,, \tag{5.4}$$

where $\mathbf{x}_k = [x(k), x(k-1), ..., x(k-M)]^T$ is the vector of channel outputs at time k with length $M+1$. The standard LMS algorithm thus uses the *stochastic gradient* in place of the gradient.

5.2.1 Bussgang Technique and Bussgang Condition

The Bussgang technique was first proposed in [1]. Since in BE there is no explicit information about the training symbol $s(k-d)$ as required in (5.4), a memoryless estimator $\varphi(\bullet)$ based on the equalizer output $y(k)$ is used to provide an estimate of the source symbol:

$$\hat{s}(k-d) = \varphi(y(k))\,. \tag{5.5}$$

A reasonable choice of estimator is the *conditional mean* estimator [1] [2]:

$$\hat{s}(k-d) = \varphi(y(k)) = E[s(k-d)\,|\,y(k)]\,. \tag{5.6}$$

Such an estimated value can in principle be calculated at least approximately from information about the distribution of the source symbol.

With $\varphi(y(k))$ considered as an approximation of the desired symbol, the adaptation (5.4) becomes

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mu(y(k) - \varphi(y(k)))\mathbf{x}_k^* \quad . \tag{5.7}$$

From (5.7), we have for the i-th coefficient ($0 \le i \le M$) of the equalizer

$$w_{k+1}(i) = w_k(i) - \mu\big(y(k) - \varphi(y(k))\big)x(k-i)^* . \tag{5.8}$$

For the iterative updating equation described in (5.7) and (5.8), asymptotically, under algorithm convergence, the expected value of any equalizer coefficient should tend to a constant. As a result, the following condition should hold:

$$E[y(k)x(k-i)^*] = E[\varphi(y(k))x(k-i)^*] \text{ for } k \to \infty \text{ and } 0 \le i \le M . \tag{5.9}$$

As has been shown in [2], when the equalizer is doubly-infinite, i.e. $-\infty < i < +\infty$, if we multiply both sides of (5.9) with the conjugate of the equalizer coefficient $w(i-m)^*$ where m can be any integer, and sum over all i, we have

$$\begin{aligned} &E[\sum_{i=-\infty}^{+\infty} y(k)x(k-i)^* w(i-m)^*] \\ &= E[\sum_{i=-\infty}^{+\infty} y(k)x(k-i-m)^* w(i)^*] = E[y(k)y(k-m)^*] \\ &= E[\sum_{i=-\infty}^{+\infty} \varphi(y(k))x(k-i-m)^* w(i)^*] = E[\varphi(y(k))y(k-m)^*]. \end{aligned}$$

As a result, in the steady state the following equation holds

$$E[y(k)y(k-m)^*] = E[\varphi(y(k))y(k-m)^*] \ . \tag{5.10}$$

A process $\{y(k)\}$ is called a *Bussgang process* if it satisfies the condition (5.10), and the algorithm (5.7) is therefore called a *Bussgang algorithm*. From (5.10) we can see that in the steady state, upon convergence, the output is a Bussgang process, and the autocorrelation of the output sequence is equal to the cross-correlation of the output and a zero-memory nonlinearity applied to the output, with the same time lag.

Note that the Bussgang condition is derived under the assumption that the equalizer vector is doubly-infinite, which is not possible in practice. Starting from a doubly-infinite equalizer, if we keep only the central significant part containing $M+1$ taps, we get an FIR equalizer. With M large enough, this FIR equalizer can be expected to give good performance, and the Bussgang condition should be well approximated.

5.2.2 Sato Algorithm

The Sato algorithm was one of the earliest to be used for BE of multilevel PAM signals [3]. Benveniste and Goursat [4] extended the original Sato algorithm to the *generalized Sato algorithm* (GSA) to include complex signals such as QAM symbols. The GSA penalizes the deviation of the equalizer output from a representative point in the quadrant it falls in, and so it in essence uses a coarse quantization of the output as an estimate of the symbol.

The cost function of the GSA is defined as

$$J_{\mathrm{GSA}}(\mathbf{w}) = E[|\, y(k) - R_{\mathrm{GSA}}\mathrm{csgn}(y(k))\,|^2], \tag{5.11}$$

where csgn(•) is the complex sign function defined as $\text{csgn}(y(k)) = \text{sgn}(y_R(k)) + j\,\text{sgn}(y_I(k))$ for $y(k) = y_R(k) + jy_I(k)$, and $R_{\text{GSA}} = \frac{E[|s_R(k)|^2]}{E[|s_R(k)|]} = \frac{E[|s_I(k)|^2]}{E[|s_I(k)|]}$ is a constellation-dependent scaling constant.

The GSA equalizer adaptation with standard stochastic gradient is then

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mu\Big(y(k) - R_{\text{GSA}}\text{csgn}(y(k))\Big)\mathbf{x}_k^*. \tag{5.12}$$

Compared with (5.7), the estimator of the source symbol $\varphi(y(k))$ in the GSA is replaced by a slicer $R_{\text{GSA}}\text{csgn}(y(k))$.

5.2.3 Constant Modulus Algorithm

One well-known BE algorithm is the *constant modulus algorithm* (CMA) [5], [6], [7]. It was first proposed to compensate for the effect of multipath and interference by exploiting the constant modulus of the signal in signaling constellations such as binary PAM or PSK [5]. The idea of the CMA is to minimize the dispersion of the modulus of the equalizer output from a constellation-dependent scaling constant. It has been shown that the CMA can also equalize non-constant modulus signals, such as from QAM constellations.

The cost function of the CMA is expressed as

$$J_{\text{CMA}}(\mathbf{w}) = E[(|y(k)|^2 - R_{\text{CMA}})^2], \tag{5.13}$$

where the scaling constant R_{CMA} can be defined as $R_{\text{CMA}} = \frac{E[|s(k)|^4]}{E[|s(k)|^2]}$. The equalizer vector adaptation with the standard stochastic gradient descent method is

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mu(|y(k)|^2 - R_{\text{CMA}})y(k)\mathbf{x}_k^*. \tag{5.14}$$

The CMA does not use an explicit estimator to estimate the source symbol, but it can still be written in a form similar to that in (5.7). Comparing (5.14) with (5.7), we see that in the CMA the "estimator" can be considered to be $\varphi(y(k)) = y(k) - (|y(k)|^2 - R_{\text{CMA}})y(k)$.

Besides the GSA and the CMA, there are many other BE algorithms that explore various other constellation properties of the source symbols, such as the *multimodulus algorithm* (MMA) [8] and the *square contour algorithm* (SCA) [9]. There has also been other work that modifies the popular Bussgang-type algorithms to improve performance with additional terms that are based on constellation structural properties, such as by adding a constellation-matched error term [10], [11]. In fact, for all the algorithms mentioned above, the cost function $J(\mathbf{w})$ can be written as the expectation of some function of the equalizer output, i.e. $J(\mathbf{w}) = E[G(y(k))]$. For a cost function of this form, the adaptation for the standard stochastic gradient descent method can be written in general as

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mu g(y(k))\mathbf{x}_k^*, \tag{5.15}$$

where $g(y(k)) = \dfrac{\partial G(y(k))}{\partial y^*(k)}$ is the partial derivative of $G(y(k))$ with respect to the conjugate of the equalizer output $y(k)$.

Comparing (5.15) with the Bussgang algorithm updates (5.7), we see that the BE algorithms that use the standard stochastic gradient descent method have a form similar to the original Bussgang algorithm, and are thus called *Bussgang-type* algorithms [2]. For any BE algorithm whose cost function can be written as $J(\mathbf{w}) = E[G(y(k))]$, we can find a nonlinear "estimator"

$$\varphi(y(k)) = y(k) - g(y(k)), \tag{5.16}$$

such that the adaptation for the equalizer vector has the expression

$$\begin{aligned}\mathbf{w}_{k+1} &= \mathbf{w}_k - \mu(y(k) - \varphi(y(k)))\mathbf{x}_k^* \\ &= \mathbf{w}_k - \mu g(y(k))\mathbf{x}_k^*.\end{aligned}$$

In the steady-state, the Bussgang condition

$$E[y(k)y(k-m)^*] = E[\varphi(y(k))y(k-m)^*]$$

is equivalent to

$$E[g(y(k))y(k-m)^*] = 0\,. \tag{5.17}$$

We will see in Section 5.6that the Bussgang condition (5.17) helps to explain why our proposed algorithms can speed up convergence during the process of adaptation.

5.3 Natural Gradient and Relative Gradient

In the canonical standard stochastic gradient algorithm (5.1), the standard gradient (SG) is used to minimize a cost function $J(\mathbf{w})$ in the *Euclidean space* of equalizer vectors. With the SG, the vector makes a change in what we would like to be the "steepest" descent direction at each iteration. However, when the parameter space corresponds to a *Riemannian manifold*, the SG may not represent the steepest descent direction of the cost function [12], [13]. For example, the set of equalizer vectors which are nominally in $\mathbb{R}^{M+1}$ may have constraints on them which put them in such a manifold. For simplicity of explanation, in this section we will only consider functions of a real vector or matrix. We will give results for complex scalars, vectors and matrices in the other sections.

Let us first look at the iteration (5.1) when all the elements of vector $\mathbf{w}$ are real-valued, to understand why there might be a problem with the standard gradient descent algorithm [12]. For the real case we have

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mu \nabla_{\mathbf{w}} J(\mathbf{w}_k) \ . \tag{5.18}$$

Suppose we make a coordinate transformation with matrix $\mathbf{B}$ such that the vector variable $\mathbf{w}$ is transformed into $\tilde{\mathbf{w}} = \mathbf{B}\mathbf{w}$, then for the cost function $J(\mathbf{w}) = J(\mathbf{B}^{-1}\tilde{\mathbf{w}})$, the gradient with respect to $\tilde{\mathbf{w}}$ is

$$\nabla_{\tilde{\mathbf{w}}} J(\mathbf{w}) = \nabla_{\tilde{\mathbf{w}}} J(\mathbf{B}^{-1}\tilde{\mathbf{w}}) = \mathbf{B}^{-T} \nabla_{\mathbf{w}} J(\mathbf{w}), \tag{5.19}$$

where $\mathbf{B}^{-T} = (\mathbf{B}^{-1})^T$. Under the transformed coordinates, the iteration for the new variable $\tilde{\mathbf{w}}$ is

$$\tilde{\mathbf{w}}_{k+1} = \tilde{\mathbf{w}}_k - \mu \nabla_{\tilde{\mathbf{w}}} J(\mathbf{B}^{-1}\tilde{\mathbf{w}}_k), \tag{5.20}$$

which is equivalent to

$$\mathbf{B}\mathbf{w}_{k+1} = \mathbf{B}\mathbf{w}_k - \mu \mathbf{B}^{-T} \nabla_{\mathbf{w}} J(\mathbf{w}_k). \tag{5.21}$$

Comparing (5.21) and (5.18) we see that under a coordinate transformation the first term on the right of (5.18), which is the vector $\mathbf{w}$ itself, transforms with $\mathbf{B}$; while the second term, which is the small perturbation to $\mathbf{w}$, transforms with $\mathbf{B}^{-T}$. As a result, in (5.18) we are actually adding two terms that do not change in the same way under coordinate transformation. The parameter space (space of equalizer weight vectors $\mathbf{w}$) implicitly assumed here is Euclidean, while it may be more appropriate to model it as a Riemann manifold.

5.3.1 Natural Gradient

Let $\delta\mathbf{w}$ be a small change of length ε at point $\mathbf{w}$. In Euclidean space, the square of the length can be expressed as

$$\|\delta\mathbf{w}\|^2 = \sum_{i=j} \delta w_i \delta w_j = \varepsilon^2 . \tag{5.22}$$

A general form of (5.22) can be written as

$$\|\delta\mathbf{w}\|^2 = \sum_{i,j} R_{ij}(\mathbf{w}) \delta w_i \delta w_j = \delta\mathbf{w}^T \mathbf{R} \delta\mathbf{w} = \varepsilon^2 , \tag{5.23}$$

with $\mathbf{R} = \{R_{ij}(\mathbf{w})\}$ being an identity matrix for Euclidean space. In a Riemannian manifold, a positive definite $\mathbf{R} = \{R_{ij}(\mathbf{w})\}$ is defined according to the structure of the manifold of $\mathbf{w}$ vectors. Such a square matrix $\mathbf{R}$ is called the *Riemannian metric tensor*.

A Riemannian manifold is a *smooth manifold* M which is locally Euclidean. For example, a circle in $\mathbb{R}^2$ and a sphere in $\mathbb{R}^3$ are manifolds that have local Euclidean structure which changes smoothly. For each point $p \in M$, there is a *tangent space* $T_p M$. For a sphere in $\mathbb{R}^3$, the tangent space at point p is the plane that touches the sphere only at point p and is perpendicular to the radius through p. The *Riemannian metric* g_R assigns to each point in the manifold M an inner product on the corresponding tangent space $T_p M$, which changes smoothly from point to point as one traverses the manifold. A *Riemannian metric tensor* $\mathbf{R}$ arises from the Riemannian metric g_R as a function of $p \in M$.

Amari [13] showed (through a simple constrained minimization) that for manifold M, in terms of the Riemannian metric tensor $\mathbf{R}$, the steepest descent direction of $J(\mathbf{w})$ is given by

$$\nabla_{\mathbf{w}}^{(N)} J(\mathbf{w}) = \mathbf{R}^{-1} \nabla_{\mathbf{w}} J(\mathbf{w}) . \tag{5.24}$$

The modified version $\nabla_{\mathbf{w}}^{(N)} J(\mathbf{w})$ of $\nabla_{\mathbf{w}} J(\mathbf{w})$ in (5.24) is called the *natural gradient* (NG) [13]. If we use the NG in place of the SG for blind equalizer updates, we get the NG version of (5.18):

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mu \nabla_{\mathbf{w}}^{(N)} J(\mathbf{w}_k) = \mathbf{w}_k - \mu \mathbf{R}^{-1} \nabla_{\mathbf{w}} J(\mathbf{w}_k) . \tag{5.25}$$

When there is a coordinate transformation with a matrix $\mathbf{B}$, the small change $\delta \mathbf{w}$ to the equalizer vector $\mathbf{w}$ becomes $\delta \tilde{\mathbf{w}} = \mathbf{B} \delta \mathbf{w}$, i.e. $\delta \mathbf{w} = \mathbf{B}^{-1} \delta \tilde{\mathbf{w}}$. As expressed in (5.23), the square of the length of change is then

$$(\mathbf{B}^{-1} \delta \tilde{\mathbf{w}})^T \mathbf{R} (\mathbf{B}^{-1} \delta \tilde{\mathbf{w}}) = \delta \tilde{\mathbf{w}}^T (\mathbf{B}^{-T} \mathbf{R} \mathbf{B}^{-1}) \delta \tilde{\mathbf{w}} = \varepsilon^2 .$$

As a result, with coordinate transformation, the metric tensor for $\tilde{\mathbf{w}}$ is transformed to $\mathbf{B}^{-T} \mathbf{R} \mathbf{B}^{-1}$ [12]. At the same time, from (5.19) we know that the SG is transformed to $\mathbf{B}^{-T} \nabla_{\mathbf{w}} J(\mathbf{w})$, so that the NG with respect to $\tilde{\mathbf{w}}$ becomes

$$\nabla_{\tilde{\mathbf{w}}}^{(N)} J(\mathbf{w}) = \nabla_{\tilde{\mathbf{w}}}^{(N)} J(\mathbf{B}^{-1} \tilde{\mathbf{w}}) = (\mathbf{B}^{-T} \mathbf{R} \mathbf{B}^{-1})^{-1} \mathbf{B}^{-T} \nabla_{\mathbf{w}} J(\mathbf{w}) = \mathbf{B} \mathbf{R}^{-1} \nabla_{\mathbf{w}} J(\mathbf{w}) . \tag{5.26}$$

With the coordinate transformation, the adaptation with the NG for the new variable $\tilde{\mathbf{w}}$ can be written as

$$\tilde{\mathbf{w}}_{k+1} = \tilde{\mathbf{w}}_k - \mu \nabla_{\tilde{\mathbf{w}}}^{(N)} J(\mathbf{B}^{-1} \tilde{\mathbf{w}}_k) . \tag{5.27}$$

Substituting into (5.27) the expression for the NG of (5.26), we see that (5.27) is the same as

$$\mathbf{B}\mathbf{w}_{k+1} = \mathbf{B}\mathbf{w}_k - \mu\mathbf{B}\mathbf{R}^{-1}\nabla_{\mathbf{w}} J(\mathbf{w}_k). \tag{5.28}$$

Compared with (5.25), all the terms in (5.28) transform the same way with $\mathbf{B}$ when there is a coordinate transformation. This means that when there is a coordinate transformation $\mathbf{B}$ in (5.25), the variable $\mathbf{w}$ as well as the perturbation will also be transformed by $\mathbf{B}$, which makes more sense than what happens with the SG. In addition, it was shown in [13] that the NG has the advantage of being asymptotically *Fisher efficient*; in other words, the algorithm gives a result asymptotically equivalent to the batch mode approach where the available information can be reused again and again.

Besides spaces of vectors, the NG can also be defined for matrix spaces. Of particular interest in applications such as blind source separation is the space of invertible square matrices. As in the vector case, a real-valued cost function $L(\mathbf{W})$ of matrix $\mathbf{W}$ is defined to measure performance. For the matrix case, when there is a perturbation $\delta\mathbf{W}$ at $\mathbf{W}$, the squared norm of $\delta\mathbf{W}$ can be written in a form similar to (5.23):

$$\|\delta\mathbf{w}\|^2 = \sum R_{ij,kl}(\mathbf{W})\delta W_{ij}\delta W_{kl}. \tag{5.29}$$

With a specification of an appropriate Riemannian metric tensor $\mathbf{R} = \{R_{ij,kl}(\mathbf{W})\}$, the NG can be obtained and used for adaptation. In [13], the authors use a relatively simple approach to obtain the explicit form for the NG (and implicitly the metric tensor $\mathbf{R}$). For the space of invertible matrices, starting with a deviation $\delta\mathbf{W}$ at $\mathbf{W}$, we consider the deviation $\delta\mathbf{W}\cdot\mathbf{W}^{-1}$ which is the corresponding deviation at $\mathbf{I} = \mathbf{W}\cdot\mathbf{W}^{-1}$. An invariance condition for this space of matrices then requires $\|\delta\mathbf{W}\|_{\mathbf{W}}^2 = \|\delta\mathbf{W}\cdot\mathbf{W}^{-1}\|_{\mathbf{I}}^2$, where the subscripts make explicit the point in the space at which the norm is obtained. If we denote

by $\nabla_{\mathbf{W}} L(\mathbf{W}) = \dfrac{\partial L(\mathbf{W})}{\partial \mathbf{W}}$ the SG of $L(\mathbf{W})$, then according to the analysis in [13], the NG of $L(\mathbf{W})$ can be expressed explicitly as

$$\nabla_{\mathbf{W}}^{(N)} L(\mathbf{W}) = \nabla_{\mathbf{W}} L(\mathbf{W}) \mathbf{W}^T \mathbf{W}, \tag{5.30}$$

which is the SG post-multiplied by $\mathbf{W}^T \mathbf{W}$. This post-multiplication of the SG is the operation that converts it into the NG. With the NG, the adaptation for matrix $\mathbf{W}$ as expressed in the vector counterpart (5.25) becomes

$$\mathbf{W}_{k+1} = \mathbf{W}_k - \mu \nabla_{\mathbf{W}} L(\mathbf{W}_k) \mathbf{W}_k^T \mathbf{W}_k. \tag{5.31}$$

5.3.2 Relative Gradient

So far we have discussed the widely used standard stochastic gradient, and also the natural gradient when a Riemannian manifold is considered. Another gradient can also be used in stochastic gradient descent. The *relative gradient* was first proposed by Cardoso [14] in the context of blind source separation, where the mixing matrix has full column rank, as mentioned in Chapter 2. It was developed independently of [13] and from a different point view. However, the results in [13] and [14] are closely related, and this relationship has been explored by Cardoso in [15].

In general, to minimize a scalar-valued cost function $L(\mathbf{W})$, instead of searching over all possible $\delta\mathbf{W}$ of some small norm we can consider a small change $\delta\mathbf{W} = \mathcal{E}\mathbf{W}$ that is proportional to the current $\mathbf{W}$, with the matrix $\mathcal{E}$ a "small" matrix. Therefore $\mathcal{E}$ is a measure of the change $\delta\mathbf{W}$ *relative* to $\mathbf{W}$. Writing out the Taylor expansion of $L(\mathbf{W})$ when the small change in $\mathbf{W}$ is made by an amount proportional to $\mathbf{W}$, we have

$$
\begin{aligned}
L(\mathbf{W}+\mathcal{E}\mathbf{W}) &= L(\mathbf{W}) + \text{trace}[\nabla_{\mathbf{W}} L(\mathbf{W})^T \mathcal{E}\mathbf{W}] + o(\mathcal{E}) \\
&= L(\mathbf{W}) + \text{trace}[(\nabla_{\mathbf{W}} L(\mathbf{W})\mathbf{W}^T)^T \mathcal{E}] + o(\mathcal{E})
\end{aligned}
.
$$

Appendix 5B gives details of this expansion. From this we find that if we pick $\mathcal{E}$ to be in the direction opposite to $\nabla_{\mathbf{W}} L(\mathbf{W})\mathbf{W}^T$, the descent rate of $L(\mathbf{W})$ is the largest (from among all relative changes by $\mathcal{E}$ of the same length). This leads to the definition of the *relative gradient* (RG) for this matrix case as

$$
\nabla_{\mathbf{W}}^{(R)} L(\mathbf{W}) = \nabla_{\mathbf{W}} L(\mathbf{W})\mathbf{W}^T . \tag{5.32}
$$

It is significant that the result (5.32) for the RG turns out to be very closely related to the NG of (5.30) for the case of *invertible* matrices $\mathbf{W}$. Comparing (5.32) with (5.30), we see that when the matrix variable $\mathbf{W}$ is in the space of invertible square matrices, the NG is simply the RG post-multiplied by $\mathbf{W}$, i.e.

$$
\nabla_{\mathbf{W}}^{(N)} L(\mathbf{W}) = \nabla_{\mathbf{W}}^{(R)} L(\mathbf{W})\mathbf{W} . \tag{5.33}
$$

Using the RG of (5.32), the updates for matrix $\mathbf{W}$ become

$$
\begin{aligned}
\mathbf{W}_{k+1} &= \mathbf{W}_k - \mu \nabla_{\mathbf{W}}^{(R)} L(\mathbf{W}_k)\mathbf{W}_k \\
&= \mathbf{W}_k - \mu \nabla_{\mathbf{W}} L(\mathbf{W}_k)\mathbf{W}_k^T\mathbf{W}_k ,
\end{aligned} \tag{5.34}
$$

which is the same as (5.31) for the NG updates. As a result, if we start with the same initialization and use the same step-size μ in (5.31) and (5.34), the change made to the matrix $\mathbf{W}$ in the *k*-th iteration will be the same and equal to $\mu \nabla_{\mathbf{W}} L(\mathbf{W}_k)\mathbf{W}_k^T\mathbf{W}_k$.

The RG approach that we outlined above is not dependent on the concepts of Riemannian manifolds and metric tensors, and it can be understood quite intuitively and easily. We can now appreciate that the idea of using the RG is related to the concept of the NG, where the Riemannian metric on the tangent space is not fixed but depends on the

location of the variable in the manifold. In the space of invertible matrices both the NG and RG give exactly the same result for the minimization updates, which is different for the SG. With the NG and RG, the perturbation to the matrices transform in the same way as the variable when there is a coordinate transformation, unlike the case with the SG.

To use the NG approach, we need to know explicitly or implicitly the matrix tensor $\mathbf{R}$. Finding the specific metric tensor explicitly may not be straightforward in applications. Furthermore, obtaining the NG for non-square matrices, for example for over-determined or under-determined blind source separation (BSS) problems, is non-trivial [16]. The RG can be found relatively easily for such situations. Given these apparent advantages of the RG, we will now proceed to explore its use in Bussgang algorithms as an alternative to the SG versions of these algorithms.

5.4 Bussgang Algorithm with Relative Gradient

In this section, we will show how the relative gradient (RG) used in place of the standard gradient (SG) modifies the Bussgang-type algorithms for adaptive updates of the equalizer vector. The analysis is for complex vectors or matrices, and the details are in Appendix 5B and Appendix 5C.

In BE, a cost function $J(\mathbf{w})$ of the equalizer vector based on the output is defined. From the review of the Bussgang-type BE algorithms in Section 5.2, we see that the cost function $J(\mathbf{w})$ is usually defined as the expectation of some function of the equalizer output, i.e. $J(\mathbf{w}) = E[G(y(k))]$. As derived in Appendix 5C, when there is a small change

$\delta\mathbf{w}$ in $\mathbf{w}$, omitting time indices k, the Taylor expansion of function $J(\mathbf{w})$ can be written as

$$\begin{aligned} J(\mathbf{w}+\delta\mathbf{w}) &= E[G((\mathbf{w}+\delta\mathbf{w})^T\mathbf{x})] = E[G(y+\delta\mathbf{w}^T\mathbf{x})] \\ &= J(\mathbf{w}) + 2\,\mathrm{Re}\left\langle \delta\mathbf{w}, E\frac{\partial G}{\partial y^*}\mathbf{x}^* \right\rangle + o(\delta\mathbf{w}) \\ &\approx J(\mathbf{w}) + 2\,\mathrm{Re}\left\langle \delta\mathbf{w}, Eg(y)\mathbf{x}^* \right\rangle, \end{aligned} \tag{5.35}$$

Here $\langle \mathbf{m}, \mathbf{n} \rangle = \sum_i m_i n_i^*$ is the Hermitian inner product of two vectors, $g(y) = \dfrac{\partial G(y)}{\partial y^*}$ is the partial derivative with respect to the conjugate of the equalizer output, and $\mathbf{x}$ is the vector of observations from the channel. Thus the SG of $J(\mathbf{w})$ is

$$\nabla_{\mathbf{w}^*} J(\mathbf{w}) = E[g(y)\mathbf{x}^*] \ ,$$

and as given in (5.15), the standard stochastic gradient updates are

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mu g(y(k))\mathbf{x}_k^* \ . \tag{5.15}$$

Suppose we now consider a small perturbation to $\mathbf{w}$ that is proportional to the current value $\mathbf{w}$. To be comprehensive, we consider two possibilities for vector variable $\mathbf{w}$: the relative change is a small *scalar* ε such that the small perturbation is $\delta\mathbf{w} = \varepsilon\mathbf{w}$, or the relative change is a small *matrix* $\boldsymbol{\mathcal{E}}$ such that the small perturbation to $\mathbf{w}$ is $\delta\mathbf{w} = \boldsymbol{\mathcal{E}}\mathbf{w}$.

When $\delta\mathbf{w} = \varepsilon\mathbf{w}$, by replacing $\delta\mathbf{w}$ with $\varepsilon\mathbf{w}$ in equation (5.35), we have

$$\begin{aligned} J(\mathbf{w}+\varepsilon\mathbf{w}) &= E[G((\mathbf{w}+\varepsilon\mathbf{w})^T\mathbf{x})] = E[G(y+\varepsilon\mathbf{w}^T\mathbf{x})] \\ &= J(\mathbf{w}) + 2\,\mathrm{Re}\left\langle \varepsilon, Eg(y)\mathbf{w}^H\mathbf{x}^* \right\rangle + o(\varepsilon). \end{aligned}$$

To minimize $J(\mathbf{w})$, when ε is in the direction opposite to that of $Eg(y)\mathbf{w}^H\mathbf{x}^*$ in the complex plane, the change rate is maximum. Thus the RG of $J(\mathbf{w})$ with respect to $\mathbf{w}$ for scalar relative change is

$$\nabla_{\mathbf{w}}^{(R)} J(\mathbf{w}) = Eg(y)\mathbf{w}^H \mathbf{x}^* \,, \tag{5.36}$$

and the stochastic adaptation is

$$\begin{aligned} \mathbf{w}_{k+1} &= \mathbf{w}_k - \mu g(y(k))\mathbf{w}_k^H \mathbf{x}_k^* \mathbf{w}_k \\ &= \mathbf{w}_k - \mu g(y(k))\mathbf{w}_k \mathbf{w}_k^H \mathbf{x}_k^* \end{aligned} \,. \tag{5.37}$$

The second line in (5.37) comes from the fact that in the second term on the right side of equation (5.37), $\mathbf{w}_k^H \mathbf{x}_k^* = y(k)^*$ is a scalar and can either pre- or post- multiply $\mathbf{w}_k$.

Comparing (5.37) and (5.15), we see that when the RG for scalar relative change is used instead of the SG, the second term on the right side of the equation, which is the change to $\mathbf{w}_k$, is modified by a matrix $\mathbf{w}_k \mathbf{w}_k^H$. Since $\mathbf{w}_k \mathbf{w}_k^H$ is a rank-1 matrix, this is not a very useful result. This is apparent when we consider $\mathbf{w}$ to be initialized with a single non-zero coefficient. In general the perturbation to $\mathbf{w}_k$ in (5.37) is $\mu g(y(k))\mathbf{w}_k \mathbf{w}_k^H \mathbf{x}_k^* = \mu g(y(k))\mathbf{w}_k y(k)^*$, which is $\mathbf{w}_k$ multiplied with an output-dependent scalar $\mu g(y(k))y(k)^*$. As a result, at each iteration the elements of the vector $\mathbf{w}$ are changed according to the rule $\mathbf{w}_{k+1} = \left(1 - \mu g(y(k))y(k)^*\right)\mathbf{w}_k$, so the equalizer adaptation keeps it in the subspace of the initial vector $\mathbf{w}_0$.

For $\delta\mathbf{w} = \boldsymbol{\mathcal{E}}\mathbf{w}$, a similar analysis for the Taylor expansion gives

$$\begin{aligned} J(\mathbf{w} + \boldsymbol{\mathcal{E}}\mathbf{w}) &= E[G((\mathbf{w} + \boldsymbol{\mathcal{E}}\mathbf{w})^T \mathbf{x})] = E[G(y + \mathbf{w}^T \boldsymbol{\mathcal{E}}^T \mathbf{x})] \\ &= J(\mathbf{w}) + 2\,\mathrm{Re}\left\langle \boldsymbol{\mathcal{E}}, Eg(y)\mathbf{x}^* \mathbf{w}^H \right\rangle + o(\boldsymbol{\mathcal{E}}) \,, \end{aligned} \tag{5.38}$$

where $\langle \mathbf{M}, \mathbf{N} \rangle = \mathrm{Trace}\{\mathbf{N}^H \mathbf{M}\}$ is the matrix inner product. The change rate is maximum when $\boldsymbol{\mathcal{E}}$ is aligned with $E[g(y)\mathbf{x}^* \mathbf{w}^H]$, thus the RG of $J(\mathbf{w})$ for matrix relative change is

$$\nabla_{\mathbf{w}}^{(R)} J(\mathbf{w}) = E[g(y)\mathbf{x}^* \mathbf{w}^H]. \tag{5.39}$$

Based on the RG for a matrix relative change, the equalizer vector adaptation becomes

$$\begin{aligned}\mathbf{w}_{k+1} &= \mathbf{w}_k - \mu g(y(k))\mathbf{x}_k^* \mathbf{w}_k^H \mathbf{w}_k \\ &= \mathbf{w}_k - \mu g(y(k))\mathbf{x}_k^* \,\|\, \mathbf{w}_k \,\|^2 .\end{aligned} \tag{5.40}$$

Comparing (5.40) and (5.15), we see that the standard gradient descent algorithm is now scaled in the second term on the right side of (5.15) by the norm-square of the vector $\mathbf{w}_k$. Thus at each step, the effective step-size is controlled by how "large" $\mathbf{w}_k$ is. The performance with (5.40) depends on many factors such as the specific channel, the length of the equalizer, and the initialization of the equalizer vector. It cannot be concluded easily from (5.40) if the step-size scaling term $\|\, \mathbf{w}_k \,\|^2$ will help speed up convergence or not. Even though there might be some benefit, a modification that is based essentially on this simple step-size control only cannot be expected to give significant improvement.

In fact, the RG for scalar relative change is a special case of the RG for matrix relative change, because we have $\varepsilon\mathbf{w} = (\varepsilon\mathbf{I})\mathbf{w}$. When all the elements on the diagonal of the relative change matrix $\boldsymbol{\mathcal{E}}$ are identical and the off-diagonal terms are zero, the relative change becomes a scalar.

From the analysis above, we may conclude that the use of the RG (for matrix relative change) instead of the SG in defining Bussgang-type algorithms for the equalizer *vector* **w** will not give any significant benefits in blind equalizer adaptation. We note however that the original work on the RG in [14] which showed its effectiveness was for the *matrix* variable **W** (in blind source separation). We therefore now turn to a consideration of block or matrix formulations of the adaptive blind equalization problem.

5.5 Block Versions of Standard-Gradient Bussgang Algorithms

In the standard-gradient Bussgang-type algorithms, the equalizer is applied to its input vector $\mathbf{x}_k$ that contains the current and past M outputs, where M is the order of the equalizer. Suppose we apply at each time k the equalizer to a larger block of channel outputs $\tilde{\mathbf{x}}_k = [x(k), x(k-1), ..., x(k-M+1), ... x(k-P-M+1)]^T$ of size $P+M$, as in the T-EASI scheme in Chapter 4. The equalizer convolved with this block will produce P outputs at each time k. We can then use these P equalized outputs at time k and the P length-(M+1) sub-vectors of corresponding equalizer inputs in $\tilde{\mathbf{x}}_k$ to form an averaged version of the equalizer update term in (5.15); this gives

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \frac{\mu}{P} \sum_{i=0}^{P-1} g(\tilde{y}^{(k)}(k-i)) \mathbf{x}_{k-i}^* \ . \tag{5.41}$$

Recall that $\mathbf{x}_{k-i} = [x(k-i), x(k-i-1), ..., x(k-i-M)]$, $i = 0,1,..,P-1$, are the length-$(M+1)$ vectors contained in $\tilde{\mathbf{x}}_k$, and $\tilde{y}^{(k)}(k-i)$ is the equalizer output using the equalizer at the k-th iteration, i.e. $\tilde{y}^{(k)}(k-i) = \mathbf{w}_k^T \mathbf{x}_{k-i}$, $i = 0,1,..., P-1$. Note $y(k) = \tilde{y}^{(k)}(k)$, and $\tilde{y}^{(k)}(k-i)$ is different from $y(k-i) = \mathbf{w}_{k-i}^T \mathbf{x}_{k-i}$, where the equalizer used is from the $(k-i)$-th iteration in the past. Equation (5.41) can be considered to be a block version of the standard Bussgang algorithm. By using a larger block, we take more information into account in the adaptation at each iteration. Compared to the standard-gradient Bussgang algorithms with $P=1$, the convergence speed may be expected to be

faster, depending on how much the extra information from a larger block helps in the equalizer vector updates. In fact, based on simulation results that are shown later in this section, there is not much difference between the standard Bussgang algorithms with $P=1$ and their corresponding block versions with $P>1$, even when we take relatively large blocks, for example $P=50$. Nonetheless, we will find it useful to pursue this block structure further at this point, because it turns out to be quite effective in the case of relative gradient.

Define a $P\times(P+M)$ Toeplitz "equalizer matrix" containing the equalizer coefficient vector as follows:

$$\mathbf{W}=\begin{pmatrix} w(0) & w(1) & \cdots & w(M) & \cdots & \cdots & 0 \\ & \ddots & & \cdots & & \ddots & \cdots \\ \cdots & w(0) & w(1) & \cdots & w(M) & \cdots & \\ & & \ddots & & \cdots & & \ddots \\ 0 & \cdots & \cdots & w(0) & w(1) & \cdots & w(M) \end{pmatrix}. \tag{5.42}$$

Then the adaptive equalizer coefficient vector update of (5.41) can be written in matrix form with two steps at each iteration:

$$\begin{aligned} \hat{\mathbf{W}}_{k+1} &= \mathbf{W}_k - \mu\mathbf{g}(\tilde{\mathbf{y}}_k)\tilde{\mathbf{x}}_k^H, \\ \mathbf{W}_{k+1} &= Toeplitz\{\hat{\mathbf{W}}_{k+1}\}. \end{aligned} \tag{5.43}$$

Here $\tilde{\mathbf{y}}_k=[y(k),\tilde{y}^{(k)}(k-1),...,\tilde{y}^{(k)}(k-P+1)]^T$ is the block of P outputs from the current equalizer, and $\mathbf{g}(\tilde{\mathbf{y}}_k)=[g(y(k)),g(\tilde{y}^{(k)}(k-1)),...,g(\tilde{y}^{(k)}(k-P+1))]^T$; in the second step the Toeplitz structure of (5.42) is forced on $\hat{\mathbf{W}}_{k+1}$ by taking averages along the descending diagonals of $\hat{\mathbf{W}}_{k+1}$ and forcing the upper right and lower left corners of the matrix to be zero. The resulting $\mathbf{W}_{k+1}$ matrix will then contain the updated equalizer coefficients of

(5.41) in each row. Now we will show that the matrix updates of the first step of (5.43) can be obtained by the standard stochastic gradient descent method with a properly defined cost function of the equalizer matrix $\mathbf{W}$.

In the standard Bussgang-type algorithm, our goal is to minimize $J(\mathbf{w}) = E[G(y(k))]$. When the stochastic gradient descent iterations for $\mathbf{w}_k$ converge to a good equalizer vector, the equalizer outputs should be close to the actual transmitted symbols. If we apply this equalizer after convergence to a large block of observations $\tilde{\mathbf{x}}_k$, then the equalizer outputs $\tilde{y}^{(k)}(k-i)$, $i = 0,1,...,P-1,$ will be approximations of the source symbols. As a result, $E[G(\tilde{y}^{(k)}(k-i))]$ is expected to be close to its minimum for $i = 0,1,...,P-1$, and so is their average. Thus we now consider as our cost function the sum of the $E[G(\tilde{y}^{(k)}(k-i))]$. This leads to a modified optimization problem.

Let $\mathbf{W}$ be the Toeplitz matrix containing the coefficients of the equalizer $\mathbf{w}$ in each row. Define a new cost function as

$$L(\mathbf{W}) = \mathbf{1}_{1\times p} E[\mathbf{G}(\mathbf{W}\tilde{\mathbf{x}}_k)] = \mathbf{1}_{1\times p} E[\mathbf{G}(\tilde{\mathbf{y}}_k)] = \sum_{i=0}^{P-1} E[G(\tilde{y}^{(k)}(k-i))], \qquad (5.44)$$

where $\mathbf{G}(\tilde{\mathbf{y}}_k)$ is the vector with function $G(\bullet)$ applied component-wise on the $\tilde{\mathbf{y}}_k$. The function $L(\mathbf{W})$ is minimized when $\mathbf{W}$ is a good "equalizer matrix", i.e. $\mathbf{W}$ contains the coefficients of the optimal equalizer.

According to Appendix 5A, the elements of the standard gradient (SG) matrix of $L(\mathbf{W})$ can be calculated component-wise as

$$\left(\nabla_{\mathbf{W}^*} L(\mathbf{W})\right)_{mn} = \frac{\partial L(\mathbf{W})}{\partial W_{mn}^*} = E[g(\tilde{y}^{(k)}(k-m+1))\tilde{x}(k-n+1)^*] \ . \qquad (5.45)$$

As a result the SG of $L(\mathbf{W})$ is

$$\nabla_{\mathbf{W}^*} L(\mathbf{W}) = E[\mathbf{g}(\tilde{\mathbf{y}}_k)\tilde{\mathbf{x}}_k^H] \,, \tag{5.46}$$

and the standard stochastic gradient descent adaptation becomes

$$\hat{\mathbf{W}}_{k+1} = \mathbf{W}_k - \mu \mathbf{g}(\tilde{\mathbf{y}}_k)\tilde{\mathbf{x}}_k^H \,. \tag{5.47}$$

The equation (5.47) is the same as the first step iteration of (5.43), which transformed the vector form in (5.41) to the matrix form updates. Toeplitz structure is then forced on $\hat{\mathbf{W}}_{k+1}$ with $\mathbf{W}_{k+1} = Toeplitz\{\hat{\mathbf{W}}_{k+1}\}$ as in (5.43) so that the set of the equalizer coefficients are in each row of $\mathbf{W}$.

Fig. 5.1 and Fig. 5.2 give examples of the performance of the standard-gradient Bussgang algorithm and the corresponding block versions of the standard-gradient Bussgang algorithm. These figures are for the CMA and GSA, respectively. In the first example, a sequence of i.i.d symbols from a 32-PSK constellation is transmitted through a complex minimum phase channel with received $\text{SNR} = 20\text{dB}$. The channel has response $\mathbf{h} = [1, -0.349 + 0.2617j, 0.007 + 0.343j, -0.2168 - 0.019j, 0.1445 - 0.1357j]$. The order of the equalizer is set to be $M = 15$, and P is taken to be relatively large ($P = 50$). We use *inter-symbol interference* (ISI) in dB to measure the performance of the algorithm, where $\text{ISI} = \sum_i \frac{|c_i|^2}{\max_i |c_i|^2} - 1$ for equalized system $\mathbf{c} = \mathbf{h} * \mathbf{w}$. The step-sizes are chosen so that the final ISI will be approximately the same for the block and non-block versions. It can be seen from Fig. 5.1 that the performance of the standard CMA and the block version CMA in this case is almost the same. The only difference is that the curve of the block version CMA is smoother than that of the standard CMA. In the second example, a sequence of

i.i.d symbols from a 64-QAM constellation is transmitted through a non-minimum phase channel with $\mathrm{SNR} = 20\mathrm{dB}$. The channel has impulse response $\mathbf{h} = [0.01 + 0.03j, -0.024 - 0.1j, 0.85 + 0.52j, -0.22 + 0.27j, 0.05 - 0.07j, -0.016 + 0.02j]$. The equalizer has order $M = 15$, and the size of the equalizer output block is taken to be $P = 10$. The ISI curves for the standard GSA and the block version GSA are shown in Fig. 5.2. As for the GSA case, the convergence speeds for these two GSA schemes are almost the same so that the curves overlap each other in the figure.

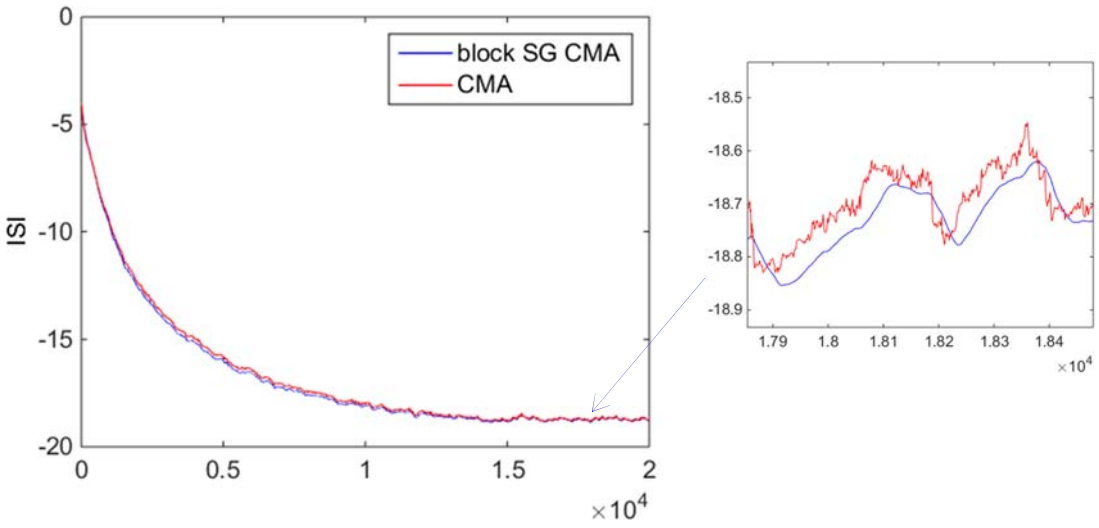

Fig. 5.1 ISI for CMA and block SG CMA. The step-sizes are $\mu_{\mathrm{CMA}} = 1.3 \times 10^{-3}$, $\mu_{\mathrm{Block\ CMA}} = 1.4 \times 10^{-3}$. $\mathrm{SNR} = 20\mathrm{dB}$, $M = 15$, $P = 50$.

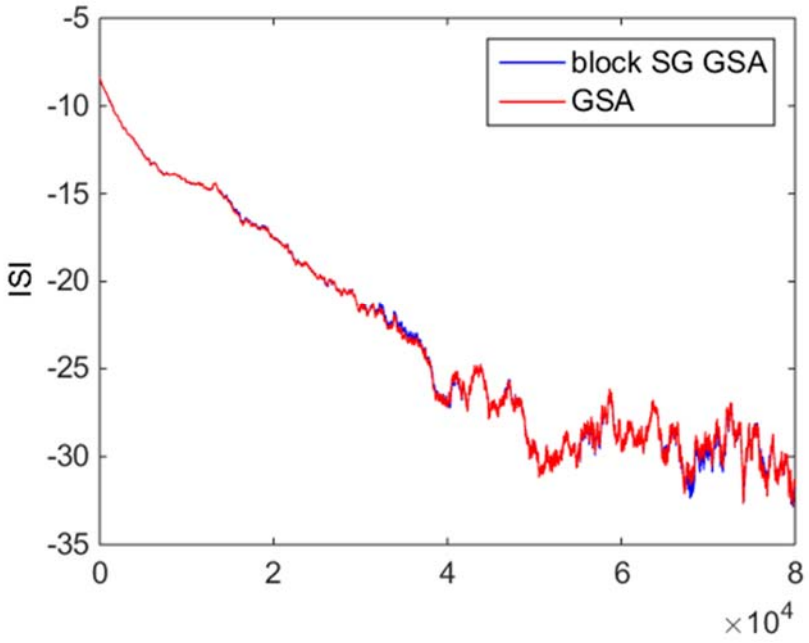

Fig. 5.2 ISI for GSA and block SG GSA. The step-sizes are $\mu_{\text{GSA}} = 8\times10^{-6}$, $\mu_{\text{Block GSA}} = 8\times10^{-6}$. SNR $= 20$dB, $M = 15$, $P = 10$.

As we have remarked earlier, even though the block versions of the standard-gradient Bussgang algorithms we have considered in this section do not show any improvement over their common vector versions, we will find the block versions to be quite effective in the case of relative gradient. Before we consider the relative gradient block versions, we digress briefly to examine an existing block version of the CMA known as the vector CMA.

Block CMA and Vector CMA

In Fig. 5.1 we considered the performance of the block version of the CMA which is a special well-known example of a Bussgang algorithm. There has been proposed in the

literature a version of the CMA called the *vector CMA* (VCMA) that is also based on processing a block of equalizer outputs at each iteration [17].

The cost function of the standard CMA is $J_{\mathrm{CMA}}(\mathbf{w}) = E[(|\, y(k)\,|^2 - R_{\mathrm{CMA}})^2]$, with $G_{\mathrm{CMA}}(y(k)) = (|\, y(k)\,|^2 - R_{\mathrm{CMA}})^2$, and as a result the $g(\bullet)$ function in (5.41) is

$$g_{\mathrm{CMA}}(\tilde{y}^{(k)}(k-i)) = (|\, \tilde{y}^{(k)}(k-i)\,|^2 - R_{CMA})\tilde{y}^{(k)}(k-i)\ . \tag{5.48}$$

Obviously $g_{\mathrm{CMA}}(\tilde{y}^{(k)}(k-i))$ depends only on $\tilde{y}^{(k)}(k-i)$. According to (5.41), the update for $\mathbf{w}$ in the block version CMA is

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \frac{\mu}{P}\sum_{i=0}^{P-1} g_{\mathrm{CMA}}(\tilde{y}^{(k)}(k-i))\mathbf{x}_{k-i}^{*}\ . \tag{5.49}$$

We know that the CMA cannot equalize Gaussian symbols, because when the source sequence is Gaussian, the output of the equalizer is always Gaussian. For zero mean Gaussian output, we have $E[|\, y(k)\,|^4] = 3E[|\, y(k)\,|^2]$, and the cost function of the CMA becomes $E[(3 - 2R_{\mathrm{CMA}})\,|\, y(k)\,|^2 + R_{\mathrm{CMA}}^2]$. In this case, the CMA cost function admits infinitely many optimal equalizer vectors.

The VCMA was first proposed in [17] for BE when the transmitted symbols come from a shaped constellation whose distribution is approximately Gaussian. It was also used for BE in OFDM systems where the source is approximately Gaussian with large number of subcarriers [18]. The cost function of the VCMA is

$$J_{\mathrm{VCMA}}(\mathbf{w}) = E[(\|\, \tilde{\mathbf{y}}_k\,\|^2 - R_{\mathrm{VCMA}})^2], \tag{5.50}$$

where the scaling constant R_{VCMA} can be defined as $R_{\text{VCMA}} = \dfrac{E[\| \tilde{\mathbf{y}}_k \|^4]}{E[\| \tilde{\mathbf{y}}_k \|^2]}$. The function (5.50) is based on the block of equalizer outputs at each iteration. It has been shown in [19] that if the source symbols are i.i.d, and $R_{\text{VCMA}} = PR_{\text{CMA}}$, (5.50) can be expressed as

$$J_{\text{VCMA}}(\mathbf{w}) = PJ_{\text{CMA}}(\mathbf{w}) + 2\sum_{i=1}^{P} E[y(k)^2 y(k-i)^2] + (1-P)R_{\text{CMA}} \ . \qquad (5.51)$$

This means that the VMCA has a composite cost function involving both the cost function of the CMA as well as higher-order cross-terms. From the analysis of stability of the channel-equalizer system, it was shown in [19] that with the higher-order cross-terms, when the source sequence is Gaussian, the VCMA admits two optimal equalizer vectors that will give the estimation of the source sequence with a possible delay.

With (5.50) as the cost function, the updates for $\mathbf{w}$ based on the stochastic SG becomes

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mu \sum_{i=0}^{P-1} g_{\text{VCMA}}(\tilde{\mathbf{y}}_k, i)\mathbf{x}_{k-i}^H \ , \qquad (5.52)$$

where $g_{\text{VCMA}}(\tilde{\mathbf{y}}_k, i) = (\| \tilde{\mathbf{y}}_k \|^2 - R_{\text{VCMA}})\tilde{y}^{(k)}(k-i)$, and μ is the step-size. Compared with (5.49), the weighting coefficient $g_{\text{VCMA}}(\tilde{\mathbf{y}}_k, i)$ of observation vector $\mathbf{x}_{k-i}$ in the updates (5.52) is a function involving multiple equalizer outputs.

The advantage of the VCMA is that it can be used for Gaussian sources. On the other hand for non-Gaussian sources the VCMA may have worse performance than the CMA. In Fig. 5.3 the ISI curves for the standard CMA, the block version of the standard CMA, and the VCMA are shown. The source sequence is composed of i.i.d 16-QAM symbols. The channel is under $\text{SNR} = 20\text{dB}$, with impulse response as

$\mathbf{h} = [0.28, 0.9816 - 0.191j, -0.5756 + 0.245j, 0.3344 - 0.1385j, 0.189 + 0.0625j, 0.0825]$.

The order of the equalizer is set to be $M = 15$, and the size of the equalizer output block is taken to be $P = 10$. We can see from Fig. 5.3 that for the source with uniform distribution over the 16-QAM constellation the VCMA performs worse than both the standard CMA and the block version of the CMA.

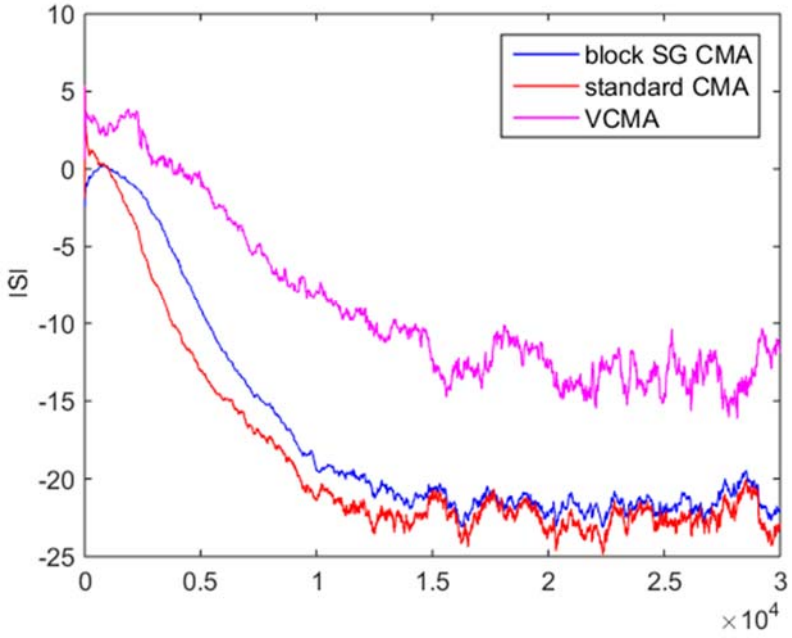

Fig. 5.3 ISI for CMA, block SG CMA and VCMA for channel with $\mathrm{SNR} = 20\mathrm{dB}$, 16QAM, $M = 15$, $P = 10$.

5.6 Block Versions of Relative-Gradient Bussgang Algorithms

We have seen in the last section that the standard BE problem can be formulated in terms of a $P \times (P + M)$ Toeplitz equalizer matrix (5.42) in which each row of the matrix is the

set of $M+1$ equalizer coefficients. The stochastic gradient descent method together with the Toeplitz constraint leads to the "block" equalizer iterations of (5.43). While we have considered the standard gradient (SG) so far in formulating the matrix or block version of the equalization problem, we now consider the use of the relative gradient (RG) in this context. We have already concluded in Section 5.3 that the RG has some appealing characteristics, and we now proceed on the expectation that use of the RG in the matrix setting for equalization will allow us to obtain useful performance gains.

5.6.1 Block RG Bussgang Equalizer Adaptation

For the matrix formulation of the block SG Bussgang equalizer algorithm, we considered the Toeplitz matrix $\mathbf{W}$ of equalizer coefficient vector $\mathbf{w}$ in each row, and defined in (5.44) a cost function $L(\mathbf{W})$ as follows:

$$L(\mathbf{W}) = \mathbf{1}_{1\times p} E[\mathbf{G}(\mathbf{W}\tilde{\mathbf{x}}_k)] = \mathbf{1}_{1\times p} E[\mathbf{G}(\tilde{\mathbf{y}}_k)] = \sum_{i=0}^{P-1} E[G(\tilde{y}^{(k)}(k-i))].$$

Here

$$\tilde{\mathbf{x}}_k = [x(k), x(k-1), ..., x(k-M+1), ...x(k-P-M+1)]^T$$

and

$$\tilde{\mathbf{y}}_k = [y(k), \tilde{y}^{(k)}(k-1), ..., \tilde{y}^{(k)}(k-P+1)]^T$$

are the input and output blocks of the equalizer at time k and $\mathbf{G}(\tilde{\mathbf{y}}_k)$ applies a scalar function G component-wise to $\tilde{\mathbf{y}}_k$.

For the cost function $L(\mathbf{W})$ of the matrix $\mathbf{W}$, derivation of the relative gradient depends on how we define the small relative change $\delta\mathbf{W}$ which is proportional to $\mathbf{W}$. We

may define $\delta\mathbf{W} = \boldsymbol{\mathcal{E}}\mathbf{W}$ with $\boldsymbol{\mathcal{E}} = \varepsilon$ a scalar. Alternatively we can have $\delta\mathbf{W} = \boldsymbol{\mathcal{E}}\mathbf{W}$ where $\boldsymbol{\mathcal{E}}$ is a $P \times P$ matrix pre-multiplying $\mathbf{W}$, or we can have $\delta\mathbf{W} = \mathbf{W}\boldsymbol{\mathcal{E}}$ where $\mathbf{W}$ is post-multiplied by a $(P+M)\times(P+M)$ matrix $\boldsymbol{\mathcal{E}}$.

From the details given in Appendix 5A, we know that the cost function $L(\mathbf{W})$ defined for a matrix has the following Taylor expansion:

$$\begin{aligned} L(\mathbf{W}+\delta\mathbf{W}) &= E[G((\mathbf{W}+\delta\mathbf{W})\tilde{\mathbf{x}})] = E[G(\tilde{\mathbf{y}}+\delta\mathbf{W}\tilde{\mathbf{x}})] \\ &= L(\mathbf{W}) + 2\,\mathrm{Re}\left\langle \delta\mathbf{W}, E\mathbf{g}(\tilde{\mathbf{y}})\tilde{\mathbf{x}}^H \right\rangle + o(\delta\mathbf{W}) . \end{aligned}$$

For the case $\delta\mathbf{W} = \boldsymbol{\mathcal{E}}\mathbf{W}$ where $\boldsymbol{\mathcal{E}} = \varepsilon$ is a scalar, Appendix 5C shows that to minimize $L(\mathbf{W})$, the change rate of $L(\mathbf{W})$ is maximum when the scalar ε is in the direction opposite to that of $\mathrm{Trace}\{E[\mathbf{g}(\tilde{\mathbf{y}})\tilde{\mathbf{y}}^H]\}$ in the complex plane. For this case of equalizer matrix $\mathbf{W}$ and scalar ε, we know from our discussion of the corresponding vector case of equalizer $\mathbf{w}$ in Section 5.4 that the scalar relative change updates the elements in the vector proportionally and does not result in a useful scheme.

When the relative change is a small pre-multiplied matrix such that the small change to $\mathbf{W}$ is $\delta\mathbf{W} = \boldsymbol{\mathcal{E}}\mathbf{W}$, the Taylor expansion is

$$\begin{aligned} L(\mathbf{W}+\boldsymbol{\mathcal{E}}\mathbf{W}) &= E[G((\mathbf{W}+\boldsymbol{\mathcal{E}}\mathbf{W})\tilde{\mathbf{x}})] \\ &= L(\mathbf{W}) + 2\,\mathrm{Re}\left\langle \boldsymbol{\mathcal{E}}, E\mathbf{g}(\tilde{\mathbf{y}})\tilde{\mathbf{y}}^H \right\rangle + o(\boldsymbol{\mathcal{E}}) . \end{aligned} \tag{5.53}$$

As a result, when $\boldsymbol{\mathcal{E}}$ is aligned with $E[\mathbf{g}(\tilde{\mathbf{y}})\tilde{\mathbf{y}}^H]$, the change rate is maximum among all relative changes $\boldsymbol{\mathcal{E}}$, and the RG for a pre-multiplied matrix change is

$$\nabla_{\mathbf{W}}^{(R)} L(\mathbf{W}) = E[\mathbf{g}(\tilde{\mathbf{y}})\tilde{\mathbf{y}}^H] = E[\mathbf{g}(\tilde{\mathbf{y}})\tilde{\mathbf{x}}^H\mathbf{W}^H]. \tag{5.54}$$

It was shown in (5.46) that the SG for function $L(\mathbf{W})$ is $\nabla_{\mathbf{W}^*} L(\mathbf{W}) = E[\mathbf{g}(\tilde{\mathbf{y}}_k)\tilde{\mathbf{x}}_k^H]$. Comparing (5.54) with the expression of the SG, we can see that the RG for a pre-multiplied matrix change is just the SG multiplied by $\mathbf{W}^H$ on the right, i.e.

$$\nabla_{\mathbf{W}}^{(R)} L(\mathbf{W}) = \nabla_{\mathbf{W}^*} L(\mathbf{W})\mathbf{W}^H \quad . \tag{5.55}$$

With the RG, the stochastic gradient descent adaptation for matrix $\mathbf{W}$ may be stated as

$$\begin{aligned} \hat{\mathbf{W}}_{k+1} &= \mathbf{W}_k - \mu \mathbf{g}(\tilde{\mathbf{y}}_k)\tilde{\mathbf{y}}_k{}^H \mathbf{W}_k, \\ \mathbf{W}_{k+1} &= Toeplitz\{\hat{\mathbf{W}}_{k+1}\}. \end{aligned} \tag{5.56}$$

The relative gradient $\mathbf{g}(\tilde{\mathbf{y}}_k)\tilde{\mathbf{y}}_k{}^H$ that pre-multiplies $\mathbf{W}_k$ in (5.56) contains cross-correlation terms of the outputs and the memoryless nonlinear function applied to the outputs. These terms are exactly the ones in the Bussgang condition $E[g(y(k))y(k-m)^*] = 0$ in (5.17), and it will be explained later that they will help make convergence faster.

When the relative change is a small post-multiplied matrix, the small change to $\mathbf{W}$ is $\delta\mathbf{W} = \mathbf{W}\boldsymbol{\mathcal{E}}$. With the small change $\mathbf{W}\boldsymbol{\mathcal{E}}$, the Taylor expansion of $L(\mathbf{W})$ is then

$$L(\mathbf{W} + \mathbf{W}\boldsymbol{\mathcal{E}}) = L(\mathbf{W}) + 2\,\mathrm{Re}\left\langle \boldsymbol{\mathcal{E}}, E\mathbf{W}^H \mathbf{g}(\tilde{\mathbf{y}})\tilde{\mathbf{x}}^H \right\rangle + o(\boldsymbol{\mathcal{E}}) \tag{5.57}$$

When $\boldsymbol{\mathcal{E}}$ is aligned with $E[\mathbf{W}^H \mathbf{g}(\tilde{\mathbf{y}})\tilde{\mathbf{x}}^H]$, the change rate is maximum, so the RG for a post-multiplied matrix change has the expression

$$\nabla_{\mathbf{W}}^{(R)} L(\mathbf{W}) = E[\mathbf{W}^H \mathbf{g}(\tilde{\mathbf{y}})\tilde{\mathbf{x}}^H]. \tag{5.58}$$

Similarly, with the matrix update and Toeplitz structure constraint, the adaptation becomes

$$\begin{aligned} \hat{\mathbf{W}}_{k+1} &= \mathbf{W}_k - \mu \mathbf{W}_k \mathbf{W}_k^H \mathbf{g}(\tilde{\mathbf{y}}_k)\tilde{\mathbf{x}}_k^H, \\ \mathbf{W}_{k+1} &= Toeplitz\{\hat{\mathbf{W}}_{k+1}\}. \end{aligned} \tag{5.59}$$

We found from our simulations for these two cases of matrix relative change (relative change $\mathcal{E}$ is a pre- or post- multiplied matrix) that the second case of post-multiplied change does not yield performance gains as good as those obtained using the pre-multiplied change. We will focus on the relative gradient algorithm of (5.56), which is based on change $\delta\mathbf{W} = \mathcal{E}\mathbf{W}$ in $\mathbf{W}$ defined by *pre-multiplication* with $\mathcal{E}$, in the rest of this discussion.

5.6.2 Expected Convergence Performance

Recall that for any Bussgang-type algorithm with cost function $J(\mathbf{w}) = E[G(y(k))]$, the iterations using the standard stochastic gradient descent method can be written as

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mu g(y(k))\mathbf{x}_k^* \ , \tag{5.15}$$

where $g(y(k))$ is a nonlinear memoryless function of the equalizer output. According to the Bussgang condition in (5.17), we have that when the algorithm reaches steady state, the following equation holds:

$$E[g(y(k))y(k-m)^*] = 0 \ .$$

In our block version of the Bussgang-type algorithms with the RG, the updates for matrix $\mathbf{W}_k$ are written as

$$\begin{aligned} \hat{\mathbf{W}}_{k+1} &= \mathbf{W}_k - \mu \mathbf{g}(\tilde{\mathbf{y}}_k)\tilde{\mathbf{y}}_k^H \mathbf{W}_k, \\ \mathbf{W}_{k+1} &= Toeplitz\{\hat{\mathbf{W}}_{k+1}\}. \end{aligned} \tag{5.56}$$

A stationary point for (5.56) is any matrix $\mathbf{W}_k$ such that $E[\mathbf{g}(\tilde{\mathbf{y}}_k)\tilde{\mathbf{y}}_k^H] = \mathbf{0}$ holds. Therefore, unlike the case of the standard stochastic gradient descent method of (5.47), the deviation

from the Bussgang condition in the steady state is taken into consideration in deciding how large the change should be relative to the current $\mathbf{W}$. When $\mathbf{g}(\tilde{\mathbf{y}}_k)\tilde{\mathbf{y}}_k^H$ is large, the adaptation is far from the steady state, and the relative change to $\mathbf{W}$ is large as a result; while when $\mathbf{g}(\tilde{\mathbf{y}}_k)\tilde{\mathbf{y}}_k^H$ is small, it is close to the steady state, and the value of $\mathbf{W}$ is then adjusted with small relative change at each iteration.

Matrix $\mathbf{g}(\tilde{\mathbf{y}}_k)\tilde{\mathbf{y}}_k^H$ is a $P \times P$ matrix that contains the cross-correlation terms with time lag up to $P-1$. On the one hand, the larger P is, the more information is used to update matrix $\mathbf{W}$ at each iteration. However, on the other hand, the larger P is, the more elements there are in the matrix $\mathbf{g}(\tilde{\mathbf{y}}_k)\tilde{\mathbf{y}}_k^H$ to adjust through updates to $\mathbf{W}$ in the adaptation, and the more difficult it is to converge to the stationary point $E[\mathbf{g}(\tilde{\mathbf{y}}_k)\tilde{\mathbf{y}}_k^H] = \mathbf{0}$. As a result, as in the selection of the size of equalizer output block in the T-EASI algorithm, the parameter P needs to be selected carefully to balance between good performance and relatively fast convergence speed. From extensive simulations we have found that a reasonable choice is $P \approx M/2$.

5.7 Equalizer Vector Adaptation and Computationally Efficient Implementation

In the block RG Bussgang algorithms, we update the "equalizer matrix" at each iteration, and force it to have Toeplitz structure, so that we get the vector of the equalizer coefficients by picking any row from the matrix. As in the case of the matrix adaptation in Chapter 4

for T-EASI algorithm, while the algorithm (5.56) is updating the whole matrix each time, if we focus only on the equalizer vector we want the computational complexity of the scheme can be reduced.

For Block RG Bussgang algorithms, let $\mathbf{U}_k = \mathbf{g}(\tilde{\mathbf{y}}_k)\tilde{\mathbf{y}}_k^H$, which is a $P \times P$ matrix containing the cross-correlation terms, so that the updates for the "equalizer matrix " in (5.56) can be written as

$$\begin{aligned} \hat{\mathbf{W}}_{k+1} &= \mathbf{W}_k - \mu \mathbf{U}_k \mathbf{W}_k, \\ \mathbf{W}_{k+1} &= Toeplitz\{\hat{\mathbf{W}}_{k+1}\}. \end{aligned} \tag{5.60}$$

The matrix adaptation in (5.60) has the same form as that in (4.10) for T-EASI algorithms, with a different expression for matrix $\mathbf{U}_k$. The step-size of the block RG Bussgang scheme in this chapter is denoted as μ. For the T-EASI algorithm in Chapter 4 the step-size was denoted by λ. Using the same technique as in Section 4.4, we can see that for the l -th coefficient of the equalizer vector the update becomes

$$w_{k+1}(l) = w_k(l) - \mu \frac{1}{P} \sum_{\tau=\max\{1-P,-l\}}^{\min\{P-1,M-l\}} \sum_{j=\max\{1-\tau,1\}}^{\min\{P-\tau,P\}} U_k(\tau+j,j) w_k(\tau+l) \tag{5.61}$$

To make the expression in (5.61) simpler, let

$$\begin{aligned} r_k(\tau) &= \sum_{j=\max\{1-\tau,1\}}^{\min\{P-\tau,P\}} U_k(\tau+j,j) \\ &= \sum_{j=\max\{1-\tau,1\}}^{\min\{P-\tau,P\}} g\left(\tilde{y}^{(k)}(k-(\tau+j)+1)\right) \tilde{y}^{(k)*}(k-j+1), \end{aligned} \tag{5.62}$$

i.e.

$$r_k(\tau)=\begin{cases}\sum_{j=1}^{P-\tau} g\left(\tilde{y}^{(k)}(k-(\tau+j)+1)\right)\tilde{y}^{(k)*}(k-j+1) & \text{when } \tau\geq 0\\ \sum_{j=1-\tau}^{P} g\left(\tilde{y}^{(k)}(k-(\tau+j)+1)\right)\tilde{y}^{(k)*}(k-j+1) & \text{when } \tau<0\end{cases}, \quad (5.63)$$

then the adaptation for the *l*-th coefficient of vector $\mathbf{w}$ becomes

$$w_{k+1}(l)=w_k(l)-\mu\frac{1}{P}\sum_{\tau=\max\{1-P,-l\}}^{\min\{P-1,M-l\}} r_k(\tau)w_k(\tau+l) \quad (5.64)$$

The adaptation in (5.64) holds for either $P\geq M+1$ or $P<M+1$. When $P\geq M+1$, $r_k(\tau)$ has definition on $-M\leq\tau\leq M$; for the case $P<M+1$ we can define

$$r_k(\tau)=0 \text{ for } -M\leq\tau<1-P \text{ and } P-1<\tau\leq M \quad (5.65)$$

so that $r_k(\tau)$ is defined over the full range of $-M\leq\tau\leq M$. With these definitions of $r_k(\tau)$, equation (5.64) can be further simplified as

$$w_{k+1}(l)=w_k(l)-\mu\frac{1}{P}\sum_{\tau=-l}^{M-l} r_k(\tau)w_k(\tau+l). \quad (5.66)$$

If we write equation (5.66) in vector-matrix form, we get

$$\mathbf{w}_{k+1}=\mathbf{w}_k-\frac{\mu}{P}\boldsymbol{\Psi}_k\mathbf{w}_k, \quad (5.67)$$

where $\boldsymbol{\Psi}_k$ is a $(M+1)\times(M+1)$ correction matrix for the equalizer vector, containing the cross-correlation terms $r_k(\tau)$ of (5.63):

$$\boldsymbol{\Psi}_k=\begin{bmatrix} r_k(0) & r_k(1) & & r_k(M)\\ r_k(-1) & r_k(0) & & r_k(M-1)\\ \vdots & \vdots & \cdots & \vdots\\ r_k(-M) & r_k(-M+1) & & r_k(0)\end{bmatrix}. \quad (5.68)$$

The terms $\{r_k(\tau)\}$ in the matrix $\mathbf{\Psi}_k$ are related to those in the matrix $\mathbf{U}_k = \mathbf{g}(\tilde{\mathbf{y}}_k)\tilde{\mathbf{y}}_k^H$, and should be approximately zero in the steady state according to the Bussgang condition. When $P-1$ is taken to be smaller than M, the lower left and upper right corners of $\mathbf{\Psi}_k$ will be zeros. As mentioned earlier, for large P the number of cross-correlation terms we use at each update is large, and convergence can be expected to be slower.

Compared with the matrix updating in (5.60), the computational complexity of vector updating in (5.67) is reduced approximately by an order of P.

As in the case of the T-EASI algorithm in Chapter 4, we can reduce computational complexity by using the FFT algorithm or by using approximations for the cross-correlation terms $\{r_k(\tau)\}$ without updating all the terms in the matrix $\mathbf{U}_k$. These schemes can reduce the computational cost from $O(M^2)$ to $O(M \log M)$. Since the matrix $\mathbf{U}_k$ in the block RG Bussgang algorithm is simpler than that in the T-EASI algorithm, it is not difficult to figure out the details of the implementation based on the explanations in Section 4.5 of Chapter 4. The details are omitted here.

5.8 Simulations for Block RG Bussgang

In this section, we will give simulation results for the block RG Bussgang algorithms and compare them with results for the standard Bussgang algorithms to illustrate the improvements with our proposed algorithms. Results for different channels, and different choices of parameters such as the order of the equalizer M and the size of the equalizer

output block P, will be shown and compared. In the experiments we will use the standard CMA and GSA and their RG block versions as examples.

First consider a sequence of i.i.d source symbols from a 64-QAM constellation transmitted through a long non-minimum phase channel with $\mathrm{SNR} = 20\mathrm{dB}$. The channel has order $L = 9$ with impulse response shown in Fig. 5.4. This channel is the same as the long non-minimum phase channel in Chapter 4. The order of the equalizer is set to be $M = 20$, and the size of the equalizer output block is taken to be $P = 10$. The equalizer is initialized with center tap $1 + 0.5j$, and zero elsewhere. This initialization of the equalizer vector is the same throughout this section.

We will first compare the performance of the CMA and the block RG CMA. ISI will be calculated at each iteration to measure the equalization performance. The step-sizes of the two algorithms are chosen as $\mu_{\mathrm{CMA}} = 2 \times 10^{-7}$, and $\mu_{\mathrm{RG\text{-}CMA}} = 6 \times 10^{-7}$ so that the ISI after convergence for either algorithm is approximately -20dB. Fig. 5.5 gives the result of a typical run, and Fig. 5.6 shows the average result over 10 runs. From both Fig. 5.5 and Fig. 5.6, we see that the CMA needs about 6.5×10^4 symbols for the adaptation to converge, while the block RG CMA needs 2.5×10^4.

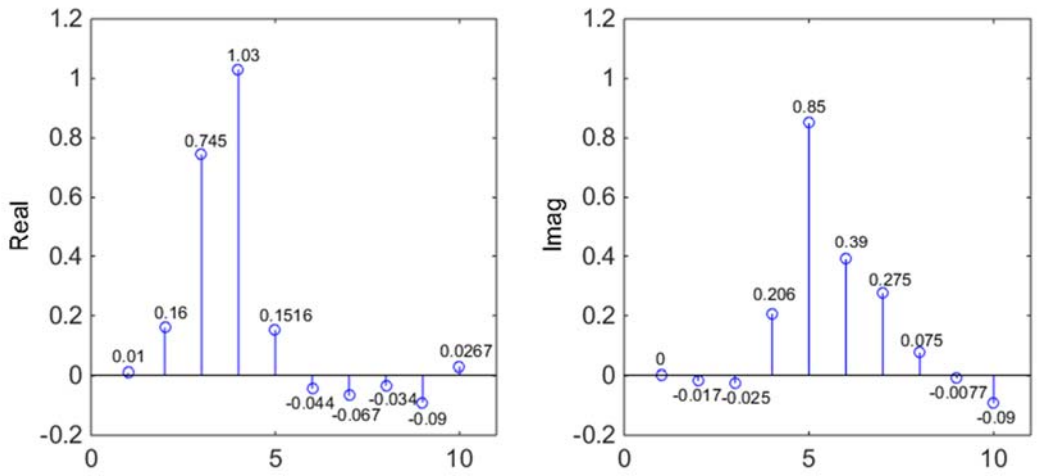

Fig. 5.4 Channel impulse response of long non-minimum phase channel.

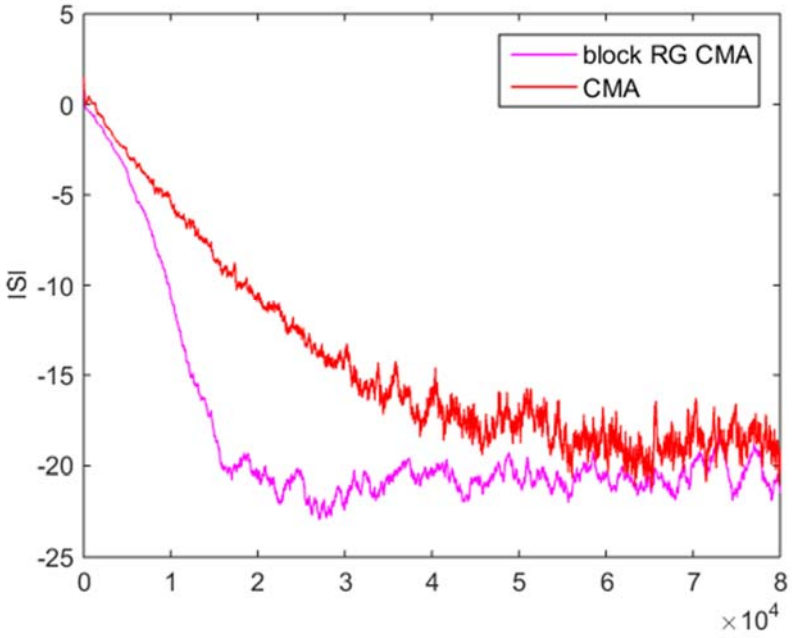

Fig. 5.5 ISI for CMA and block RG CMA, i.i.d 64-QAM source. Long non-minimum phase channel with $\mathrm{SNR} = 20\mathrm{dB}$, $\mu_{\mathrm{CMA}} = 2\times10^{-7}$, $\mu_{\mathrm{RG\text{-}CMA}} = 6\times10^{-7}$, $M = 20$, $P = 10$.

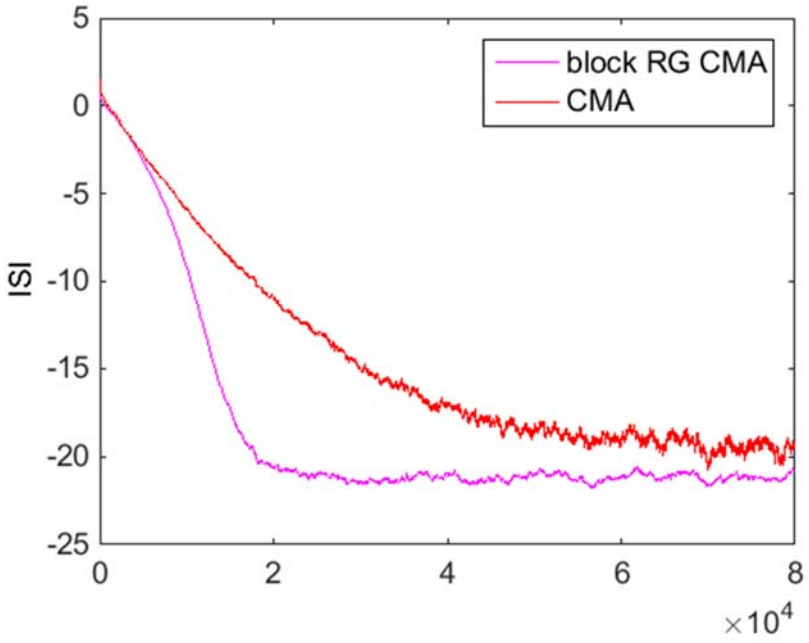

Fig. 5.6 Average over 10 runs of the ISI for CMA and block RG CMA, i.i.d 64-QAM source. Long non-minimum phase channel

Next we will see how the choice of P affects the performance. The channel, SNR, source, and the equalizer length are the same as above. We compare four choices of the block size of the equalizer outputs, i.e. $P=5$, $P=10$, $P=20$ and $P=40$. From Fig. 5.7, we can see that when P is relatively small, i.e. $P=5$, the performance of the block RG CMA is worse than that of the standard CMA. Not only the convergence speed is slower than the standard CMA, but the ISI after convergence is also higher. With $M=20$ and $P=5$, there are a total of about $\left(M-(P-1)\right)^2=256$ zero elements on the lower left and upper right corners of $\mathbf{\Psi}_k$ in (5.68). In this case the correction matrix $\mathbf{\Psi}_k$ may not provide much help in updating the coefficients of $\mathbf{w}_k$ in (5.67). As has been said, the performance improves with increasing P up to a certain point. Beyond that point, the performance may

even decrease because of the increasing dimension of the correction matrix in the adaptation. We did experiments with different choices of P, and it turned out that $P=10$ is a point when the performance is the best. When P increases further, for example when $P=20$, the performance is not much different from that when $P=10$. When $P=40$ the convergence is slower than in the cases when $P=10$ or $P=20$.

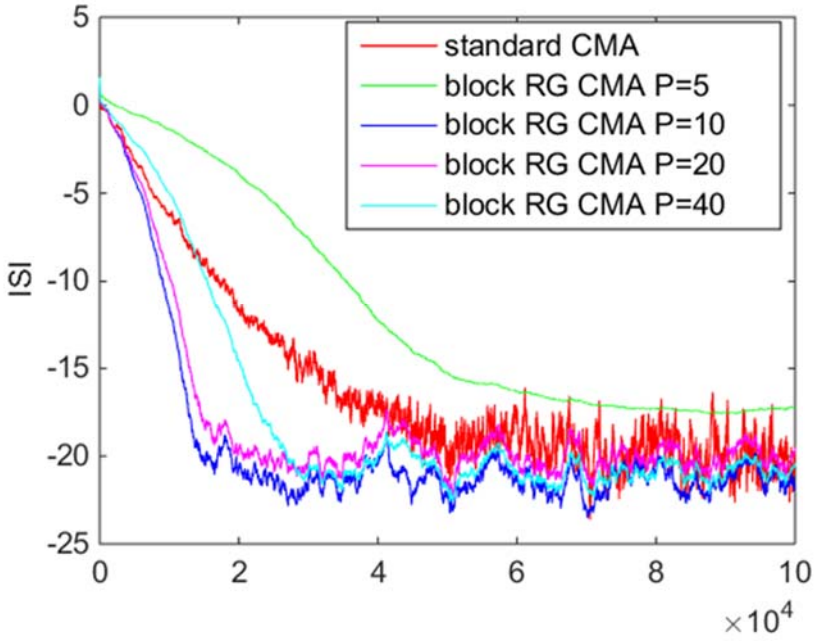

Fig. 5.7 ISI for CMA and block RG CMA with different choices of P, i.i.d 64-QAM source. Long non-minimum phase channel with $\mathrm{SNR}=20\mathrm{dB}$, $M=20$.

For the same channel in the above example, consider the case when a sequence of *correlated* symbols from a 16-QAM constellation is transmitted. For better comparison of performance, the SNR is 40dB for this example with correlated signal. The first symbol of the sequence is chosen randomly from the 16-point constellation. In the sequence, the symbol stays in the same quadrant as the previous one with probability 4/7, and jumps to one of the other three quadrants each with probability 1/7. In any of the four quadrants,

the symbol takes one of the four values in the quadrant with equal probability. From Fig. 5.8, we can see that with the standard CMA, convergence needs about 9×10^4 symbols, while block RG CMA needs 3×10^4. When the source symbols are correlated, the Bussgang algorithm can still work. However, it takes longer to converge, because the symbols are correlated, and not as much new information is provided at each iteration compared to the case of i.i.d symbols. The curves shown in Fig. 5.8 are the averages of three runs, because the curves for a single run are not smooth. When the SNR is low, the convergence speed of the block RG CMA is still faster than the CMA, but the ISI in the steady state is high.

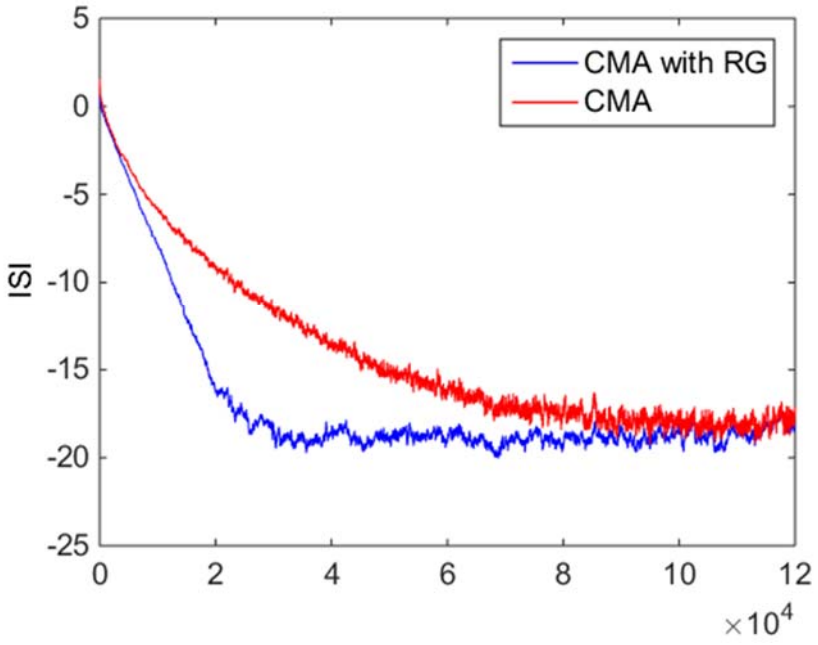

Fig. 5.8 ISI for CMA and block RG CMA, correlated 16-QAM source. Long non-minimum phase channel with $\mathrm{SNR}=40\mathrm{dB}$, $\mu_{\mathrm{CMA}}=3.5\times10^{-6}$, $\mu_{\mathrm{RG\text{-}CMA}}=13\times10^{-6}$, $M=20$, $P=10$.

For correlated source, we now compare the results for equalizer output blocks of different sizes: $P = 5$, $P = 10$, $P = 20$ and $P = 40$. From the simulation results in Fig. 5.9, we see that the worst performance is obtained when P has the relatively small value $P = 5$. The ISI after convergence is relatively high, with slow convergence. The performances of block RG CMA with $P = 10$ and $P = 20$ are similar, and the convergence speed is faster than that of the standard CMA. However, if we continue to increase P, the performance gets worse, as seen in the figure when $P = 40$.

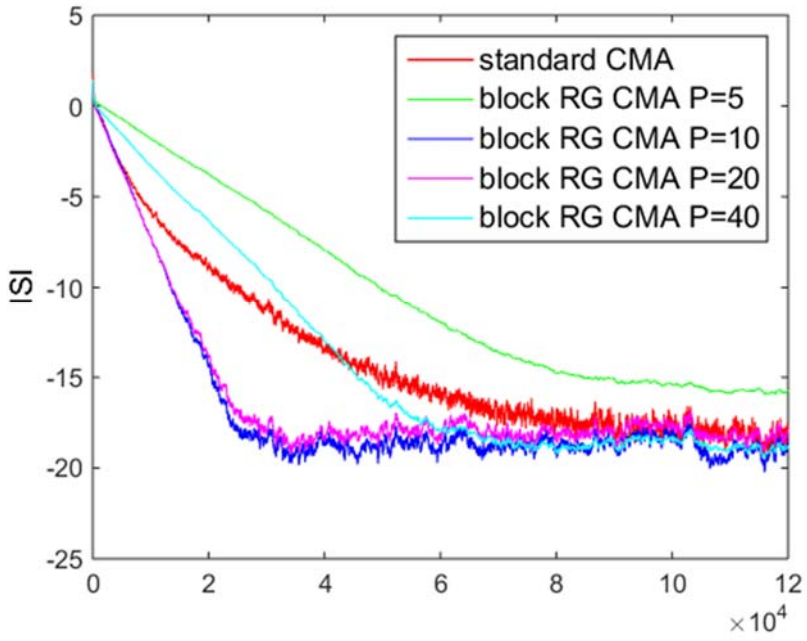

Fig. 5.9 ISI for CMA and block CMA with RG with different choices of P, correlated 16-QAM source. Long non-minimum phase channel with $\mathrm{SNR} = 40\mathrm{dB}$, $M = 20$.

For the same channel and i.i.d. 64-QAM source, we also used the standard GSA to achieve blind equalization and compared its performance with the RG block version of the GSA. The results for different values of P are shown in Fig. 5.10. The difference in convergence speeds of the GSA and the block RG GSA is similar to that of the standard

CMA and the block RG CMA. The performance with $P=10$ is best. The convergence speed with $P=20$ and $P=30$ is almost the same, and is slower than that with $P=10$. When $P=10$, for similar ISI after convergence, the standard GSA needs about 1.5×10^5 symbols to converge, while the block RG GSA only needs 6×10^4. The ISI after convergence for block RG GSA is -23dB, while that for the standard GSA is -20dB; if we adjust the step-size to make them both converge to -23dB, the standard GSA will need more symbols for convergence.

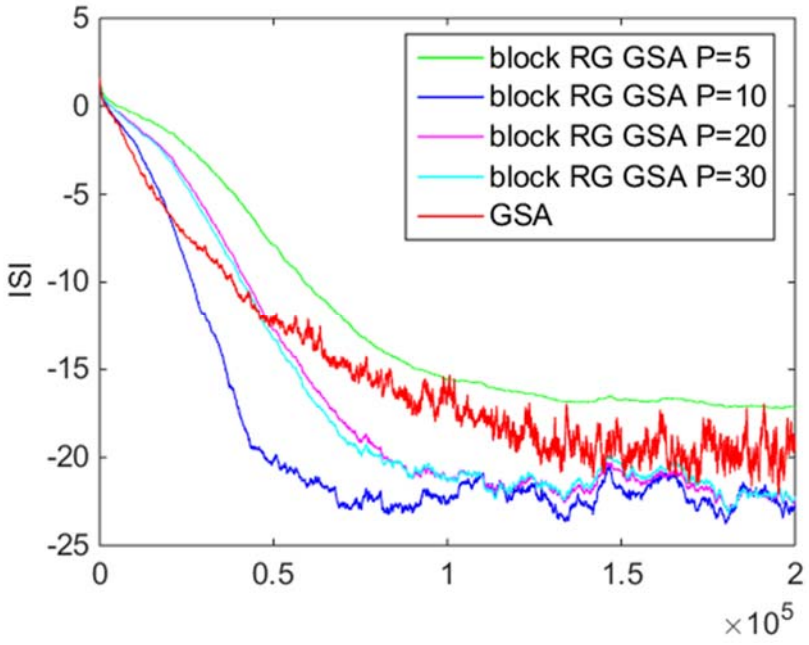

Fig. 5.10 ISI for GSA and block RG GSA with different choices of P, i.i.d 64-QAM source. Long non-minimum phase channel with $\mathrm{SNR}=20\mathrm{dB}$, $M=20$.

In the next example, the channel is a short minimum phase system that has been used in Chapter 4. Though the channel is not long with $L=4$ only, it has four zeros that are close to the unit circle, which makes it hard to equalize. The impulse response of this short

minimum phase channel is shown in Fig. 5.11. The order of the equalizer is set as $M = 20$. A sequence of i.i.d 64-QAM symbols is transmitted through the channel with $\mathrm{SNR} = 20\mathrm{dB}$. The performance of the CMA and GSA is compared with their respective RG block versions in Fig. 5.12 and Fig. 5.13. The results with different choices of P are shown. When $P = 10$, the performance of the block RG Bussgang is better than its standard Bussgang counterpart with either the CMA or GSA. When P increases further, the convergence speed becomes slower.

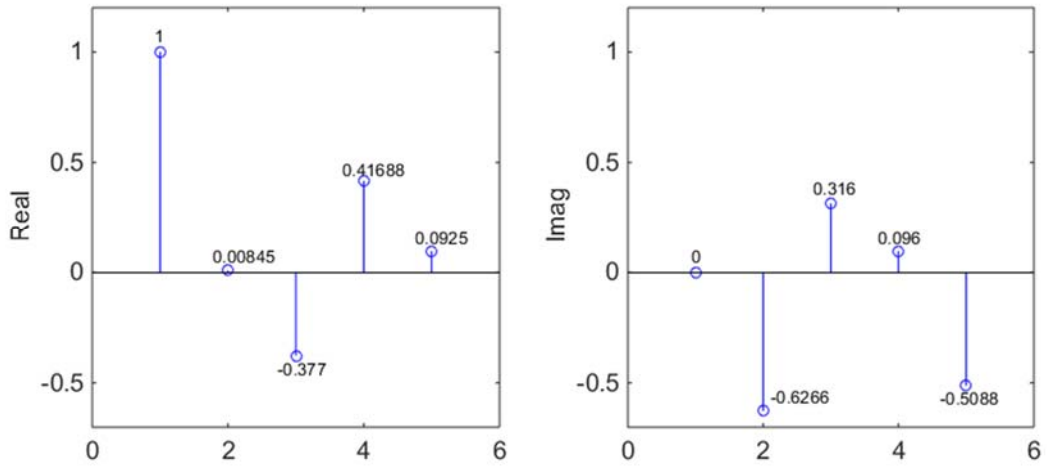

Fig. 5.11 Channel impulse response of short minimum phase channel

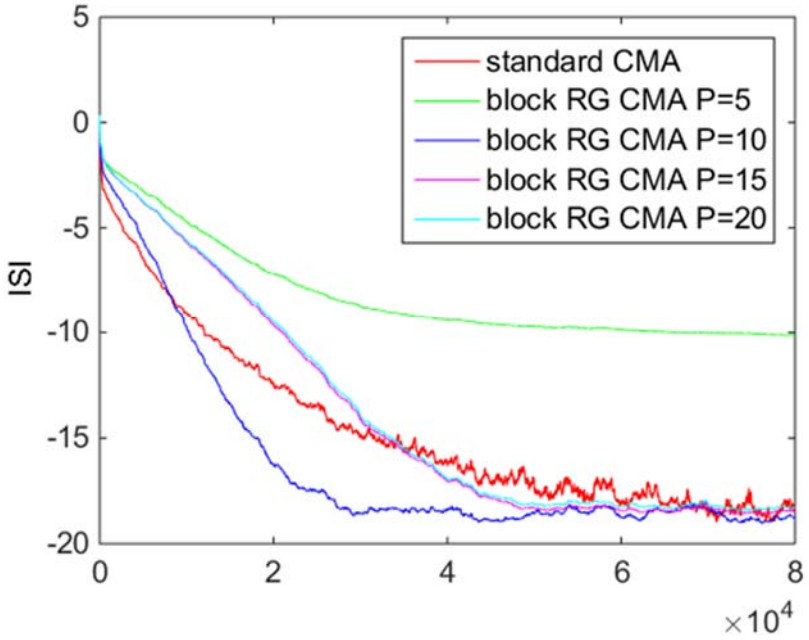

Fig. 5.12 ISI for CMA and block RG CMA with different choices of P, i.i.d 64-QAM source. Short minimum phase channel with $\mathrm{SNR} = 20\mathrm{dB}$, $M = 20$.

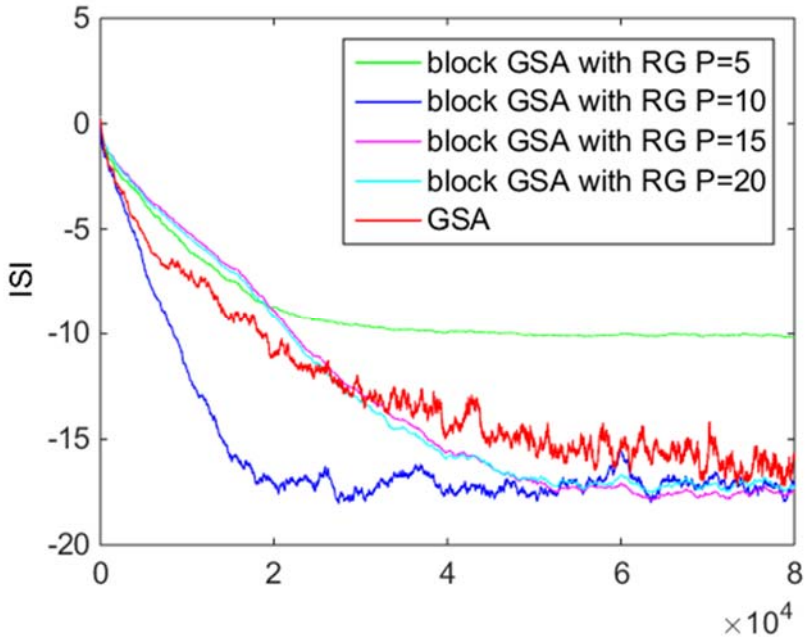

Fig. 5.13 ISI for GSA and block RG GSA with different choices of P, i.i.d 64-QAM source. Short minimum phase channel with $\mathrm{SNR} = 20\mathrm{dB}$, $M = 20$.

From the previous experiments, we see that $P = M/2$ is a good choice for the size of the equalizer output block. Next, we will see how different choices of equalizer order M may affect the performance of BE. For the non-minimum phase channel with $L = 9$, the ISI for the CMA and RG block CMA with four choices of *M* are shown in Fig. 5.14. From the figure, it can be seen that when $M = 8$ the block RG CMA performs worse than the standard CMA. There is not much difference between the standard CMA and the block RG CMA when $M = 15$; while when $M = 20$ and $M = 30$, the convergence of the block RG CMA is faster. We know that for the Bussgang condition to hold approximately, the equalizer should be long enough. When the order of the equalizer M is small, the Bussgang condition is not well approximated. In this case, the relative gradient $\mathbf{g}(\tilde{\mathbf{y}}_k)\tilde{\mathbf{y}}_k^H$ in (5.56) may not be helpful. However, when *M* is relatively large, the advantage of using the cross-correlation terms in the Bussgang condition for matrix updating becomes more significant. In practical BE problems we always set the equalizer to be of a reasonable length, so that the source symbols can be well estimated. If M is too small, the performance may not be good even for standard Bussgang algorithms. As a result, we expect that with a reasonable choice of *M*, our block RG CMA will be more likely to performance better than the standard CMA.

The performance comparisons with different choice of *M* are also shown for the minimum phase channel with zeros close to the unit circle in Fig. 5.15. The result is similar to the first case. As the equalizer length $M+1$ becomes larger, the advantage of the block RG CMA becomes more apparent. In a practical BE problem, we also need to consider the complexity of a long equalizer, thus there is a trade-off in the selection of equalizer length.

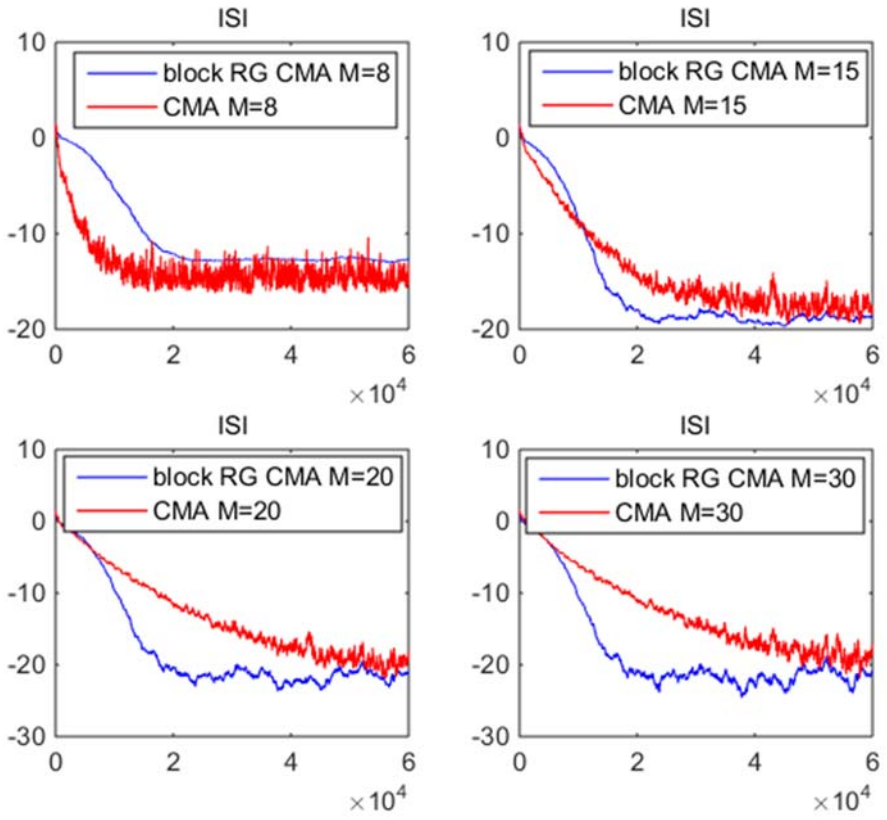

Fig. 5.14 ISI for CMA and block RG CMA with different choices of equalizer order M, for the long non-minimum phase channel. $P = M/2$.

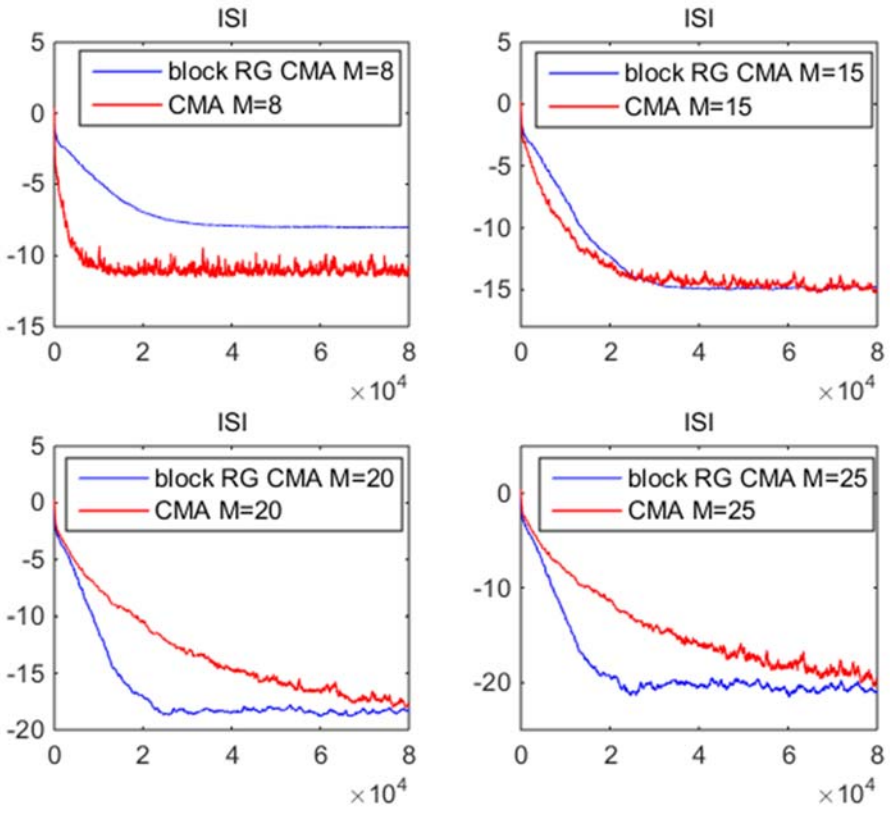

Fig. 5.15 ISI for CMA and block RG CMA with different choices of equalizer order M for the short minimum phase channel. $P = M/2$.

5.9 Comparison with ICA Based BE Algorithm

Our block RG Bussgang algorithms in this chapter have an interesting relationship to the independent component analysis (ICA)-based algorithms considered in Chapter 4. Source symbol independence is the essential condition used in the ICA approach, but this is not explicitly the basis for the Bussgang algorithms. Nonetheless these two schemes for the equalizer matrix adaptation have a structural similarity that we will examine briefly here.

In the T-EASI algorithm, the equalizer matrix is a Toeplitz matrix that contains the equalizer coefficients in each row. The equalizer matrix is updated according to

$$\begin{aligned}\hat{\mathbf{W}}_{k+1} &= \mathbf{W}_k - \lambda[\tilde{\mathbf{y}}_k\tilde{\mathbf{y}}_k{}^H - \mathbf{I} + \mathbf{g}(\tilde{\mathbf{y}}_k)\tilde{\mathbf{y}}_k^H - \tilde{\mathbf{y}}_k\mathbf{g}(\tilde{\mathbf{y}}_k)^H]\mathbf{W}_k \\ \mathbf{W}_k &= Toeplitz\{\hat{\mathbf{W}}_{k+1}\}\end{aligned}, \tag{5.69}$$

where nonlinear decorrelation is used to achieve output independence and therefore equalization. As a result, the T-EASI is not based on any specific constellation property of the source symbols. The selection of the nonlinear function g can start from a cost function that measures degree of independence of the outputs, and in general a variety of nonlinear functions can be used in the T-EASI algorithm. In the steady-state, the terms in the bracket go to zero, i.e. $\tilde{\mathbf{y}}_k\tilde{\mathbf{y}}_k{}^H - \mathbf{I} = \mathbf{0}$ and $\mathbf{g}(\tilde{\mathbf{y}}_k)\tilde{\mathbf{y}}_k^H - \tilde{\mathbf{y}}_k\mathbf{g}(\tilde{\mathbf{y}}_k)^H = \mathbf{0}$. Since the condition $\tilde{\mathbf{y}}_k\tilde{\mathbf{y}}_k{}^H - \mathbf{I} = \mathbf{0}$ requires source symbols be white or uncorrelated, the T-EASI does not work well for sources that have correlation between symbols.

For the adaptation of the block RG Bussgang with the RG of (5.56),

$$\begin{aligned}\hat{\mathbf{W}}_{k+1} &= \mathbf{W}_k - \mu\mathbf{g}(\tilde{\mathbf{y}}_k)\tilde{\mathbf{y}}_k{}^H\mathbf{W}_k, \\ \mathbf{W}_{k+1} &= Toeplitz\{\hat{\mathbf{W}}_{k+1}\}.\end{aligned} \tag{5.56}$$

one difference from (5.69) is that the correction matrix contains only cross-correlation terms $\mathbf{g}(\tilde{\mathbf{y}}_k)\tilde{\mathbf{y}}_k^H$, while in T-EASI the correction matrix is composed of two parts: a whitening part $\tilde{\mathbf{y}}_k\tilde{\mathbf{y}}_k{}^H - \mathbf{I}$ and the skew-symmetric "nonlinear decorrelation for independence" term $\mathbf{g}(\tilde{\mathbf{y}}_k)\tilde{\mathbf{y}}_k^H - \tilde{\mathbf{y}}_k\mathbf{g}(\tilde{\mathbf{y}}_k)^H$. Note that the T-EASI algorithm arises from an objective of obtaining the inverse of an unknown mixing matrix for separation of independent sources, whereas in using a Bussgang algorithm the objective is that of equalizing an unknown linear channel based directly on a source estimation error-minimizing criterion. Bussgang schemes implicitly or explicitly form a nonlinear estimate

from the equalizer output for the unknown symbol, based on which a cost function is defined. The nonlinearity *g* is the derivative of this cost function.

Although the cost functions of the block RG Bussgang algorithms do not explicitly impose the constraint of independence of source symbols, nonlinear decorrelation is included in the cross-correlation matrix condition $\mathbf{g}(\tilde{\mathbf{y}}_k)\tilde{\mathbf{y}}_k^H = 0$ for steady-state convergence. From this point of view, when the source is indeed i.i.d, the independence of the source symbols helps speed up convergence in the block RG Bussgang schemes. However, the Bussgang-type algorithms also work for correlated input symbols. It should be noted that the Bussgang condition of (5.17) contained in $\mathbf{g}(\tilde{\mathbf{y}}_k)\tilde{\mathbf{y}}_k^H = 0$ does not *necessarily* mean that the output source sequence has independent symbols.

The simulated performance of T-EASI and block RG CMA was compared for the short minimum phase channel and the long non-minimum phase channel, with i.i.d. 64-QAM source symbols, and the results are shown below. From Fig. 5.16 and Fig. 5.17, we see that the performance for the two algorithms is comparable. The convergence speed of the T-EASI is slightly faster than that of the block RG CMA.

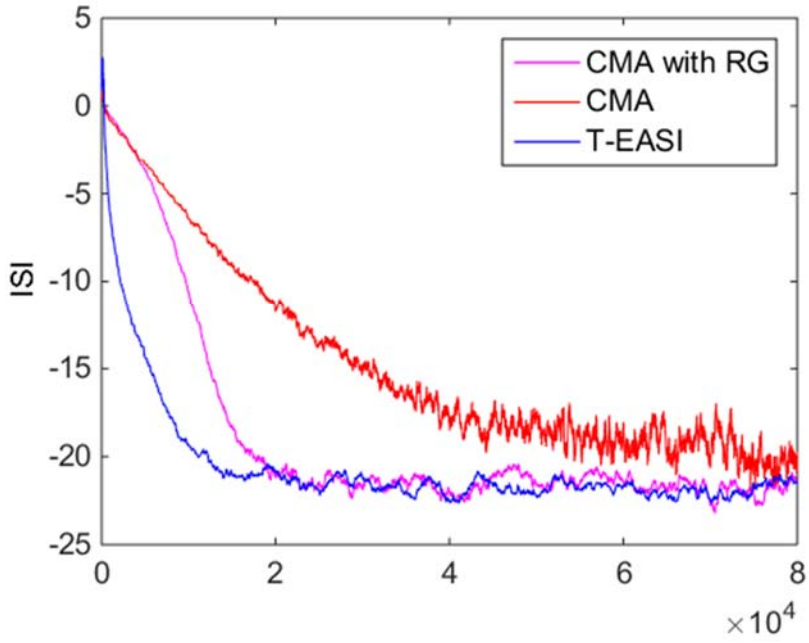

Fig. 5.16 ISI for CMA, block RG CMA and T-EASI with different choices of P, i.i.d 64-QAM source. Long non-minimum phase channel with $\mathrm{SNR} = 20\mathrm{dB}$, $M = 20$.

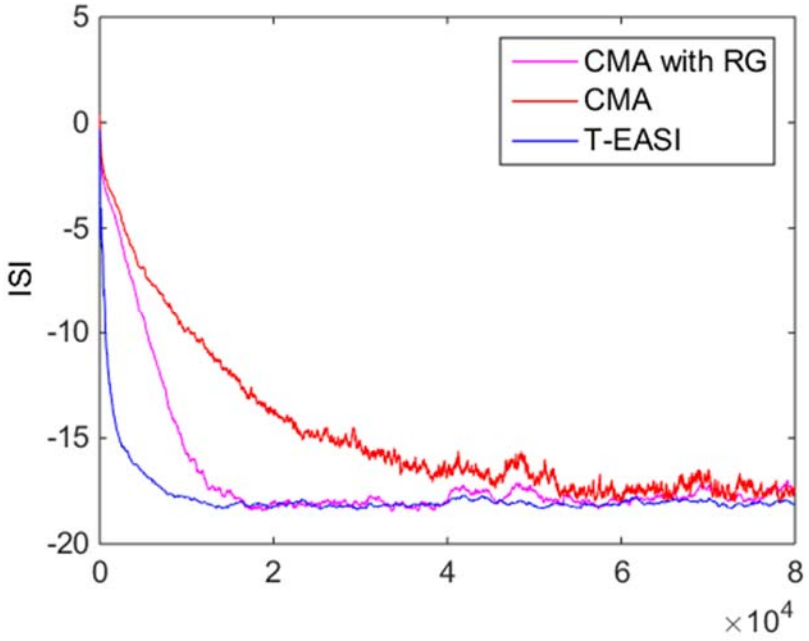

Fig. 5.17 ISI for CMA, block RG CMA and T-EASI with different choices of P, i.i.d 64-QAM source. Short minimum phase channel with $\mathrm{SNR} = 20\mathrm{dB}$, $M = 20$.

From the matrix adaptations in (5.69) and (5.56), we can see that the complexity of the T-EASI is slightly higher than that of the block RG CMA, because more terms need to be calculated in (5.69) for the equalizer matrix adaptation. This also happens when we use the computationally efficient implementation of the two algorithms. However, if the equalizer is long, and P is large as a result, the difference in computational complexity will be small, because the two algorithms have the same order of computational complexity.

In fact, the block RG CMA has very similar form to the Amari algorithm with Toeplitz constraint, which was introduced in Section 4.8. Recall that as mentioned in (4.48) that the Amari algorithm has the adaptation

$$\mathbf{W}_{k+1} = \mathbf{W}_k - \lambda\left[\mathbf{g}(\tilde{\mathbf{y}}_k)\tilde{\mathbf{y}}_k^{\,H} - \mathbf{I}\right]\mathbf{W}_k \ . \tag{5.70}$$

Comparing the first line of (5.56) and (5.70), we can see that the only difference is an identity matrix in the brackets. The Amari algorithm starts from measuring the independence of the separated symbols, but can be interpreted as using nonlinear decorrelation for separation. Generally at convergence the off-diagonal elements of $\mathbf{g}(\tilde{\mathbf{y}}_k)\tilde{\mathbf{y}}_k^H$ are zero, and the diagonal elements are normalized to one so that $\mathbf{g}(\tilde{\mathbf{y}}_k)\tilde{\mathbf{y}}_k^H = \mathbf{I}$.

Suppose now we use a nonlinearity function g for the Amari algorithm that comes from a Bussgang-type cost function based on some constellation property. It can be expected, according to the Bussgang condition, that $\mathbf{g}(\tilde{\mathbf{y}}_k)\tilde{\mathbf{y}}_k^H = \mathbf{0}$ should hold if the source symbols are recovered. Suppose after the K-th iteration $\mathbf{g}(\tilde{\mathbf{y}}_K)\tilde{\mathbf{y}}_K^H = \mathbf{0}$ is obtained, then according to (5.70), the matrix adaptation in the $(K+1)$-th iteration becomes

$$\mathbf{W}_{K+1} = (1+\lambda)\mathbf{W}_K \ , \tag{5.71}$$

and the elements in $\tilde{\mathbf{y}}_{K+1}$ becomes $(1+\lambda)\tilde{\mathbf{y}}_K$. Taking the nonlinearity of the CMA as an example, we have that vector $\mathbf{g}(\tilde{\mathbf{y}}_k)$ is a component-wise function of vector $\tilde{\mathbf{y}}_k$ with $g(y) = (|y|^2 - R_{\text{CMA}})y$. As a result, the vector $\mathbf{g}(\tilde{\mathbf{y}}_k)$ can be written as

$$\mathbf{g}(\tilde{\mathbf{y}}_k) = \mathbf{\Lambda}_k \tilde{\mathbf{y}}_k, \tag{5.72}$$

where $\mathbf{\Lambda}_k$ is a diagonal matrix with component-wise function $|y|^2 - R_{\text{CMA}}$ of vector $\tilde{\mathbf{y}}_k$ on the diagonal. After the K-th iteration, the matrix $\mathbf{g}(\tilde{\mathbf{y}}_K)\tilde{\mathbf{y}}_K^H$ becomes

$$\mathbf{g}(\tilde{\mathbf{y}}_K)\tilde{\mathbf{y}}_K^H = \mathbf{\Lambda}_K \tilde{\mathbf{y}}_K \tilde{\mathbf{y}}_K^H = \mathbf{0}. \tag{5.73}$$

Then at the $(K+2)$-th iteration, with $\mathbf{g}(\tilde{\mathbf{y}}_{K+1})\tilde{\mathbf{y}}_{K+1}^H = \mathbf{\Lambda}_{K+1}(1+\lambda)^2 \tilde{\mathbf{y}}_K \tilde{\mathbf{y}}_K^H$, the matrix adaptation becomes

$$\begin{aligned} \mathbf{W}_{K+2} &= \mathbf{W}_{K+1} - \lambda(\mathbf{g}(\tilde{\mathbf{y}}_{K+1})\tilde{\mathbf{y}}_{K+1}^H - \mathbf{I})\mathbf{W}_{K+1} \\ &= (1+\lambda)\mathbf{W}_K - \lambda(\mathbf{\Lambda}_{K+1}\tilde{\mathbf{y}}_{K+1}\tilde{\mathbf{y}}_{K+1}^H - \mathbf{I})(1+\lambda)\mathbf{W}_K \\ &= [\mathbf{I} - \lambda(1+\lambda)^2 \mathbf{\Lambda}_{K+1}\tilde{\mathbf{y}}_K \tilde{\mathbf{y}}_K^H + \lambda\mathbf{I}](1+\lambda)\mathbf{W}_K \end{aligned}. \tag{5.74}$$

The iterations keeps going as in (5.74). Intuitively, as the components of matrix $\mathbf{W}_k$ grow, so does the elements in the matrix $\mathbf{\Lambda}_k$, and this reverses the growth of $\mathbf{W}_k$ so that the tendency to converge to the steady state $\mathbf{g}(\tilde{\mathbf{y}}_k)\tilde{\mathbf{y}}_k^H = \mathbf{0}$ is maintained. As a result, even for correlated source symbols the Bussgang condition can be satisfied and we can use the nonlinearity of a Bussgang-type algorithm as the nonlinearity for the constrained Amari algorithm.

In Fig. 5.18, we compare the performance of the T-Amari algorithm and the block RG CMA for correlated input symbols. The source symbols are 16-QAM and are correlated in the way described in Section 5.8. The performance of the T-EASI algorithm is also shown for comparison. From the figure, it can be seen that for the Toeplitz constrained Amari

algorithm, when the cubic nonlinearity is used, the performance is the worst; while when the nonlinearity based on the CMA cost function is used, the performance is comparable with that of the block RG CMA algorithm. The T-EASI with cubic nonlinearity does not yield good performance. Although the algorithm works to some extent, the ISI after convergence is -14dB; when the SNR decreases or when the source symbols are from a higher-order signaling constellation, the performance can be even worse.

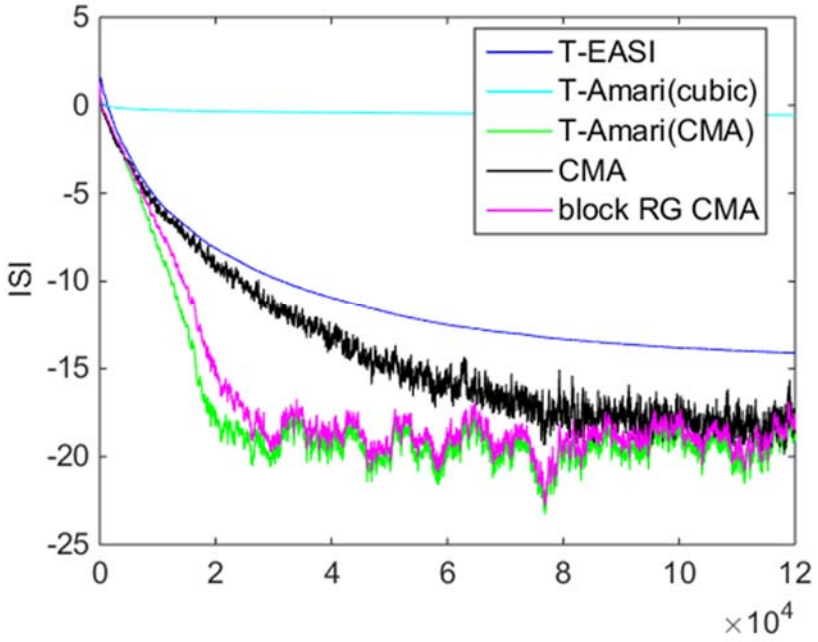

Fig. 5.18 ISI for T-EASI, T-Amari, CMA and block RG CMA, correlated 16-QAM source. Long non-minimum phase channel with $\mathrm{SNR} = 40\mathrm{dB}$, $M = 20$, $P = 10$.

The norm of the equalizer vector with the constrained Amari algorithm is also shown in Fig. 5.19. We observe from the figure that when the algorithm converges, the changes in the vector will be very small, and the curve is almost flat. This verifies our explanation for the matrix adaptation for Amari algorithm with CMA nonlinearity. When steady-state

is obtained, the algorithm tends to maintain the steady-state by preventing the components of the matrix $\mathbf{W}_k$ from large changes.

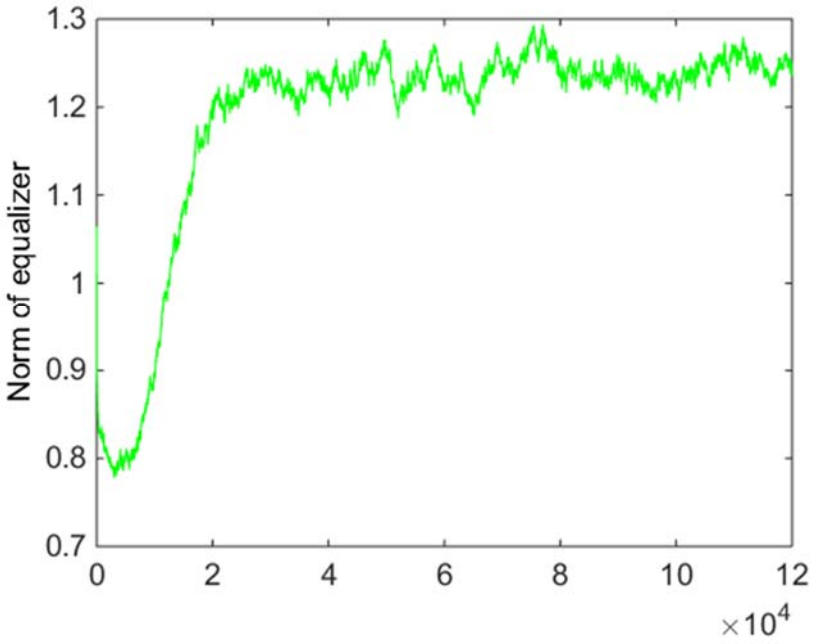

Fig. 5.19 Norm of equalizer during adaptation.

5.10 Conclusions

In this chapter we formulated an approach to process a block of equalizer outputs to modify the standard Bussgang-type algorithms, resulting in an effective relative gradient (RG) adaptation scheme for blind equalization. Our new block RG Bussgang algorithms use a block of equalizer outputs at each iteration and enforce a Toeplitz condition for faster convergence. With the RG, the Bussgang condition appears more explicitly in the equalizer adaptation steps. Simulation results suggest that the block RG Bussgang algorithms offer faster convergence compared to their standard counterparts. While the block algorithms

have a somewhat higher computational cost, our results suggest that the performance gains obtained are significant. Our work has also brought out interesting connections between different algorithms (EASI, Amari, and Bussgang).

References

[1] S. Bellini, "Bussgang techniques for blind equalization," in *IEEE Global Telecommunications Conference*, 1986, pp. 1634–1640.

[2] Z. Ding and Y. Li, *Blind Equalization and Identification*. CRC press, 2001.

[3] Y. Sato, "A Method of Self-Recovering Equalization for Multilevel Amplitude-Modulation Systems," *IEEE Trans. Commun.*, vol. 23, no. 6, pp. 679–682, 1975.

[4] A. Benveniste and M. Goursat, "Blind Equaizers.m," *IEEE Trans. Commun.*, vol. 32, no. 8, pp. 871–879, 1984.

[5] J. Treichler and B. G. Agee, "A new approach to multipath correction of constant modulus signals," *IEEE Trans. Acoust.*, vol. 31, no. 2, pp. 459–472, 1983.

[6] J. R. Treichler and M. G. Larimore, "New Processing Techniques Based on the Constant Modulus Adaptive Algorithm," *IEEE Trans. Acoust.*, vol. 33, no. 2, pp. 420–431, 1985.

[7] R. Johnson and P. Schniter, "Blind equalization using the constant modulus criterion: A review," *Proc. IEEE*, vol. 86, no. 10, pp. 1927–1950, 1998.

[8] J. Yang, J.-J. Werner, G. A. Dumont, and Others, "The multimodulus blind equalization and its generalized algorithms," *IEEE J. Sel. Areas Commun.*, vol. 20, pp. 997–1015, 2002.

[9] T. Thaiupathump, L. He, and S. A. Kassam, "Square contour algorithm for blind equalization of QAM signals," *Signal Processing*, vol. 86, pp. 3357–337–, 2006.

[10] L. He, M. G. Amin, and C. Reed Jr, "A new adaptive equalzier for QAM signals," in *IEEE Sarnoff Symposium*, 2001.

[11] L. He, M. G. Amin, C. Reed Jr, and R. C. Malkemes, "A hybrid adaptive blind equalization algorithm for QAM signals in wireless communications," *Signal Process. IEEE Trans.*, vol. 52, no. 7, pp. 2058–2069, 2004.

[12] M. Toussaint, "Lecture Notes : Some notes on gradient descent," no. 1, pp. 1–4, 2012.

[13] S. Amari, "Natural Gradient Works Efficiently in Learning," *Neural Comput.*, vol. 10, no. 2, pp. 251–276, 1998.

[14] J. F. Cardoso and B. H. Laheld, "Equivariant adaptive source separation," *IEEE Trans. Signal Process.*, vol. 44, no. 12, pp. 3017–3030, 1996.

[15] J. F. Cardoso, "Learning in manifolds: the case of source separation," in *IEEE Signal Processing Workshop on Statistical Signal and Array Processing*, 1998, pp. 136–139.

[16] L.-Q. Zhang, S. Amari, and A. Cichocki, "Natural gradient approach to blind separation of over-and undercomplete mixtures," in *In Proceeding of Independent Component Analysis and Signal Separation (ICA'99)*, 1999.

[17] V. Y. Yang and D. L. Jones, "A vector constant modulus algorithm for shaped constellation equalization," *IEEE Signal Process. Lett.*, vol. 5, no. 4, pp. 89–91, 1998.

[18] T. Abrudan and V. Koivunen, "Blind equalization in spatial multiplexing MIMO-OFDM systems based on vector CMA and decorrelation criteria," *Wirel. Pers. Commun.*, vol. 43, no. 4, pp. 1151–1172, 2007.

[19] A. Touzni, L. Tong, R. A. Casas, C. R. Johnson, and A. C. Definition, "Vector-CM Stable Equilibrium Analysis," *IEEE Signal Process. Lett.*, vol. 7, no. 2, pp. 31–33, 2000.

[20] D. H. Brandwood, "A complex gradient operator and its application in adaptive array theory," *IEE Proc. H Microwaves, Opt. Antennas*, vol. 130, no. 1, pp. 11–16, 1983.

[21] W. Wirtinger, "Zur formalen theorie der funktionen von mehr komplexen veränderlichen," *Math Ann*, vol. 97, pp. 357–375, 1927.

[22] T. Adali and S. Haykin, *Adaptive signal processing: next generation solutions*. John Wiley & Sons, 2010.

[23] H. Li and T. Adali, "Complex-Valued Adaptive Signal Processing Using Nonlinear Functions," *EURASIP J. Adv. Signal Process.*, vol. 2008, no. 1, p. 765615, 2008.

Appendix 5A

Gradient Computation in Complex Domain

One challenge of dealing with complex Blind Source Separation (BSS) or Blind Equalization (BE) problems is the derivation of gradient of the cost function in the complex domain. For a complex-valued function

$$f(z) = f_{real}(a,b) + jf_{imag}(a,b) \tag{5A.1}$$

where $z = a + jb$, the conditions of differentiability are given by the *Cauchy-Riemann equations* [20]

$$\frac{\partial f_{real}}{\partial a} = \frac{\partial f_{imag}}{\partial b}, \frac{\partial f_{imag}}{\partial a} = -\frac{\partial f_{real}}{\partial b} \tag{5A.2}$$

which require that the derivative of $f(z)$ at a point z_0 should be the same regardless of the direction of approach, in spite of the additional dimensionality in complex domain. The conditions impose a strong structure on $f_{real}(a,b)$ and $f_{imag}(a,b)$, and these are usually not satisfied for many functions of practical interest, such as the real-valued cost functions in BE and BSS problems.

One method to deal with the function in complex field is to transform the complex variable into one in real domain, i.e. $\mathbb{C}^n \mapsto \mathbb{R}^{2n}$. This transformation allows the computations in real domain, and the result can be transformed back to complex domain at the end. However, the transformation doubles the dimension of variables. Another popular method is the *Wirtinger Calculus* [21], which relaxes the Cauchy-Riemann conditions and enables the computation of derivative in complex domain in a straightforward way. Here

we will explain the Wirtinger derivative and its potential application in complex BSS and BE. We will call the gradient derived this way in Euclidean space as *standard gradient* (SG).

Scalar case

A function of complex variable z, $f(z):\mathbb{C}\mapsto\mathbb{C}$, can be considered as a function $f(a,b):\mathbb{R}\times\mathbb{R}\mapsto\mathbb{C}$ with $f(z)=f(a,b)$. It has been shown in [20] that for function $f_z(z,z^*):\mathbb{C}\times\mathbb{C}\mapsto\mathbb{C}$ such that $f_z(z,z^*)=f(a,b)$, if f_z is analytic with respect to z and z^* independently, the partial derivative could be defined as

$$\frac{\partial f_z}{\partial z}=\frac{1}{2}\left(\frac{\partial f}{\partial a}-j\frac{\partial f}{\partial b}\right) \tag{5A.3}$$

$$\frac{\partial f_z}{\partial z^*}=\frac{1}{2}\left(\frac{\partial f}{\partial a}+j\frac{\partial f}{\partial b}\right) \tag{5A.4}$$

by treating z^* and z as constant in f_z respectively. Especially, when f_z is *real-valued*, $\frac{\partial f_z}{\partial a}$ and $\frac{\partial f_z}{\partial b}$ are both real, and we have

$$\left(\frac{\partial f_z}{\partial z}\right)^*=\frac{\partial f_z}{\partial z^*}\ . \tag{5A.5}$$

When there is a small change δz to z, the Taylor expansion of function $f_z(z,z^*)$ at $(z+\delta z,z^*+\delta z^*)$ can then be written as

$$\begin{aligned} f_z(z+\delta z,z^*+\delta z^*) &= f_z(z,z^*)+\frac{\partial f_z}{\partial z}\delta z+\frac{\partial f_z}{\partial z^*}\delta z^*+o(\delta z,\delta z^*) \\ &\approx f_z(z,z^*)+2\,\mathrm{Re}\left(\frac{\partial f_z}{\partial z}\delta z\right) \end{aligned} \tag{5A.6}$$

When δz is aligned with $\left(\frac{\partial f_z}{\partial z}\right)^* = \frac{\partial f_z}{\partial z^*}$, the function will have the maximum change rate, thus the SG of real-valued function $f(z)$ is the partial derivative with respect to the conjugate z^*.

Suppose our goal is to find the optimal value of z such that the value of $f(z)$ is minimized. With the SG, the updating algorithm for the complex variable is then

$$z_{k+1} = z_k - \mu \frac{\partial f_z}{\partial z^*} \tag{5A.7}$$

where z_k is the estimate of z at the *k*-th iteration, and μ is the step-size.

Vector case

Now we will extend the scalar variable case to vector case. Let $f(\mathbf{z}): \mathbb{C}^N \mapsto \mathbb{C}$ be a function of complex vector $\mathbf{z} = [z_1, z_2, ..., z_N]^T \in \mathbb{C}^N$, and define the gradient operator with respect to $\mathbf{z}$ as $\nabla_{\mathbf{z}} = [\partial/\partial z_1, \partial/\partial z_2, ..., \partial/\partial z_N]^T$. For function $f_{\mathbf{z}}(\mathbf{z}, \mathbf{z}^*): \mathbb{C}^N \times \mathbb{C}^N \mapsto \mathbb{C}$ such that $f_{\mathbf{z}}(\mathbf{z}, \mathbf{z}^*) = f(\mathbf{z})$, if $f_{\mathbf{z}}(\mathbf{z}, \mathbf{z}^*)$ is analytic with respect to $\mathbf{z}$ and $\mathbf{z}^*$ independently, the gradient $\nabla_{\mathbf{z}}$ and $\nabla_{\mathbf{z}^*}$ of $f_{\mathbf{z}}(\mathbf{z}, \mathbf{z}^*)$ can be calculated element-wise using the definition in (5A.3) and (5A.4).

Let $\delta\mathbf{z}$ be the small change to vector $\mathbf{z}$. The first-order Taylor expansion of $f_{\mathbf{z}}(\mathbf{z}, \mathbf{z}^*)$ can be written as [22], [23]

$$f_{\mathbf{z}}(\mathbf{z} + \delta\mathbf{z}, \mathbf{z}^* + \delta\mathbf{z}^*) = f_{\mathbf{z}}(\mathbf{z}, \mathbf{z}^*) + (\nabla_{\mathbf{z}} f_{\mathbf{z}})^T \delta\mathbf{z} + (\nabla_{\mathbf{z}^*} f_{\mathbf{z}})^T \delta\mathbf{z}^* + o(\delta\mathbf{z}, \delta\mathbf{z}^*) \tag{5A.8}$$

where $\left\langle \mathbf{m},\mathbf{n} \right\rangle = \sum_{i=1}^{N} m_i n_i^* = \text{Trace}\{\mathbf{n}^H \mathbf{m}\}$ is the Hermitian inner product of two vectors. When $f_{\mathbf{z}}$ is *real-valued*, $(\nabla_{\mathbf{z}} f_{\mathbf{z}})^* = \nabla_{\mathbf{z}^*} f_{\mathbf{z}}$ and $\left\langle \delta\mathbf{z}, \nabla_{\mathbf{z}^*} f_{\mathbf{z}} \right\rangle = \left\langle \delta\mathbf{z}^*, \nabla_{\mathbf{z}} f_{\mathbf{z}} \right\rangle^*$, and it follows that

$$\begin{aligned} f_{\mathbf{z}}(\mathbf{z}+\delta\mathbf{z}, \mathbf{z}^*+\delta\mathbf{z}^*) &= f_{\mathbf{z}}(\mathbf{z},\mathbf{z}^*) + \left\langle \delta\mathbf{z}, \nabla_{\mathbf{z}^*} f_{\mathbf{z}} \right\rangle + \left\langle \delta\mathbf{z}^*, \nabla_{\mathbf{z}} f_{\mathbf{z}} \right\rangle + o(\delta\mathbf{z}, \delta\mathbf{z}^*) \\ &\approx f_{\mathbf{z}}(\mathbf{z},\mathbf{z}^*) + 2\,\text{Re}\left\langle \delta\mathbf{z}, \nabla_{\mathbf{z}^*} f_{\mathbf{z}} \right\rangle \end{aligned} \tag{5A.9}$$

From equation (5A.9) we can see that the change of function $f_{\mathbf{z}}$ is approximately $2\,\text{Re}\left\langle \delta\mathbf{z}, \nabla_{\mathbf{z}^*} f_{\mathbf{z}} \right\rangle$, and the change rate is maximum when $\delta\mathbf{z}$ is in the same or negative direction of $\nabla_{\mathbf{z}^*} f_{\mathbf{z}}$. Thus the SG of complex function $f(\mathbf{z})$ is $\nabla_{\mathbf{z}^*} f_{\mathbf{z}}$. This has wide application in standard BE problems. Suppose we want to minimize function $f(\mathbf{z})$, then the adaptation for vector $\mathbf{z}$ with the SG is

$$\mathbf{z}_{k+1} = \mathbf{z}_k - \mu \nabla_{\mathbf{z}^*} f_{\mathbf{z}} \tag{5A.10}$$

If $\mathbf{z}$ is a row vector, the gradient operator is defined accordingly as a row vector $\nabla_{\mathbf{z}} = [\partial/\partial z_1, \partial/\partial z_2, ..., \partial/\partial z_N]$. In such a case, the SG of complex function $f(\mathbf{z})$ can be derived in a similar way as above using transpose of $\mathbf{z}$ and $\nabla_{\mathbf{z}}$, and the SG still has the expression $\nabla_{\mathbf{z}^*} f_{\mathbf{z}}$. By comparing the scalar and vector case, we can see that the scalar case is a special example of the vector case where there is only one element in the vector.

Matrix case

In many applications such as BSS, the function variable is a matrix. Similar to the vector case, the gradient of the function can also be analyzed with Wirtinger Calculus.

Let $f(\mathbf{Z}):\mathbb{C}^{M\times N}\mapsto\mathbb{C}$ be a function of complex matrix $\mathbf{Z}\in\mathbb{C}^{M\times N}$. The gradient operator with respect to $\mathbf{Z}$ is defined as $\nabla_{\mathbf{Z}}$, a $M\times N$ matrix whose (i,j)-th element is the partial derivative with respect to the (i,j)-th element of $\mathbf{Z}$. For *real-valued* function $f_{\mathbf{Z}}(\mathbf{Z},\mathbf{Z}^*)=f(\mathbf{Z})$, if it is analytic with respect to $\mathbf{Z}$ and $\mathbf{Z}^*$ independently, the first-order Taylor expansion can be written as

$$\begin{aligned} f_{\mathbf{Z}}(\mathbf{Z}+\delta\mathbf{Z},\mathbf{Z}^*+\delta\mathbf{Z}^*) &= f_{\mathbf{Z}}(\mathbf{Z},\mathbf{Z}^*)+\left\langle\delta\mathbf{Z},\nabla_{\mathbf{Z}^*}f_{\mathbf{Z}}\right\rangle+\left\langle\delta\mathbf{Z}^*,\nabla_{\mathbf{Z}}f_{\mathbf{Z}}\right\rangle+o(\delta\mathbf{Z},\delta\mathbf{Z}^*) \\ &= f_{\mathbf{Z}}(\mathbf{Z},\mathbf{Z}^*)+2\,\mathrm{Re}\left\langle\delta\mathbf{Z},\nabla_{\mathbf{Z}^*}f_{\mathbf{Z}}\right\rangle+o(\delta\mathbf{Z},\delta\mathbf{Z}^*) \end{aligned} \tag{5A.11}$$

where $\left\langle\mathbf{M},\mathbf{N}\right\rangle=\mathrm{Trace}\{\mathbf{N}^H\mathbf{M}\}$, and $\nabla_{\mathbf{Z}^*}f_{\mathbf{Z}}=(\nabla_{\mathbf{Z}}f_{\mathbf{Z}})^*$. As in the vector case, the change rate is maximum when the direction of $\delta\mathbf{Z}$ aligns with that of $\nabla_{\mathbf{Z}^*}f_{\mathbf{Z}}$, which is the SG for matrix case. To minimize function $f(\mathbf{Z})$ with gradient descent the updating rule is

$$\mathbf{Z}_{k+1}=\mathbf{Z}_k-\mu\nabla_{\mathbf{Z}^*}f_{\mathbf{Z}}\,. \tag{5A.12}$$

Note that the vector case is in fact a special example of the matrix case: when $M=1,$ $\mathbf{Z}$ reduces to a row vector while with $N=1,$ $\mathbf{Z}$ reduces to a column vector; when $M=1$ and $N=1,$ $\mathbf{Z}$ becomes a scalar.

Appendix 5B

Relative Gradient in Complex Domain

The standard gradient (SG) at a particular point defines the direction and rate of steepest change at that point. However, we want to look into the case when the perturbation to a variable is proportional to its current value. Among all such perturbations, the one that gives the maximum change rate is defined as relative gradient (RG) [14].

Scalar case

First consider function $f(z):\mathbb{C}\mapsto\mathbb{C}$ of a complex scalar z, and its corresponding equivalent function with two variables $f_z(z,z^*)=f(z)$. Suppose there is a small perturbation εz proportional to z, where ε is a scalar. If f_z is *real-valued*, then according to (5A.6), the first-order expansion of f_z at (z,z^*) can be written as

$$\begin{aligned} f_z(z+\varepsilon z, z^*+\varepsilon^* z^*) &= f_z(z,z^*)+\frac{\partial f_z}{\partial z}\varepsilon z+\frac{\partial f_z}{\partial z^*}\varepsilon^* z^*+o(\varepsilon,\varepsilon^*) \\ &\approx f_z(z,z^*)+2\,\mathrm{Re}\left(\frac{\partial f_z}{\partial z}z\varepsilon\right) \end{aligned} \tag{5B.1}$$

If ε is aligned with $(\frac{\partial f_z}{\partial z}z)^*=\frac{\partial f_z}{\partial z^*}z^*$, then the change rate will be maximum. Thus the RG of the function $f(z)$ with complex scalar variable z is $\frac{\partial f_z}{\partial z^*}z^*$, which is the SG multiplied by z^*. Iterative updates of z to reach the minimum value using the RG can be given as

$$z_{k+1} = z_k - \mu \frac{\partial f_z}{\partial z^*} z_k^* z_k. \tag{5B.2}$$

Compared to (5A.7), the step-size is modified by the square of the modulus of z, i.e. $z_k^* z_k = |z_k|^2$.

Vector case

Now consider function $f(\mathbf{z}): \mathbb{C}^N \mapsto \mathbb{C}$ of a complex column vector $\mathbf{z}$, and the function $f_{\mathbf{z}}(\mathbf{z}, \mathbf{z}^*) = f(\mathbf{z})$. Suppose there is a small perturbation proportional to $\mathbf{z}$. For the vector case, there are two possibilities, one is when the perturbation is $\varepsilon\mathbf{z}$, where ε is a scalar and the other is when the perturbation is $\boldsymbol{\mathcal{E}}\mathbf{z}$, where $\boldsymbol{\mathcal{E}}$ is an $N \times N$ matrix so that $\boldsymbol{\mathcal{E}}\mathbf{z}$ has the same dimension as $\mathbf{z}$.

According to (5A.9), if $f_{\mathbf{z}}$ is *real-valued*, the first-order expansion of $f_{\mathbf{z}}$ at $(\mathbf{z}, \mathbf{z}^*)$ is

$$f_{\mathbf{z}}(\mathbf{z} + \delta\mathbf{z}, \mathbf{z}^* + \delta\mathbf{z}^*) \approx f_{\mathbf{z}}(\mathbf{z}, \mathbf{z}^*) + 2\,\mathrm{Re}\left\langle \delta\mathbf{z}, \nabla_{\mathbf{z}^*} f_{\mathbf{z}} \right\rangle .$$

If $\delta\mathbf{z} = \varepsilon\mathbf{z}$, then

$$\begin{aligned} f_{\mathbf{z}}(\mathbf{z} + \ \varepsilon\mathbf{z}, \mathbf{z}^* + \varepsilon^*\mathbf{z}^*) &\approx f_{\mathbf{z}}(\mathbf{z}, \mathbf{z}^*) + 2\,\mathrm{Re}\left\langle \varepsilon\mathbf{z}, \nabla_{\mathbf{z}^*} f_{\mathbf{z}} \right\rangle \\ &= f_{\mathbf{z}}(\mathbf{z}, \mathbf{z}^*) + 2\,\mathrm{Re}(\varepsilon \nabla_{\mathbf{z}^*} f_{\mathbf{z}}^H \mathbf{z}) \end{aligned} . \tag{5B.3}$$

When ε is aligned with $(\nabla_{\mathbf{z}^*} f_{\mathbf{z}}^H \mathbf{z})^* = (\nabla_{\mathbf{z}^*} f_{\mathbf{z}})^T \mathbf{z}^*$, the change rate is maximum. Thus the RG of $f(\mathbf{z})$ in this case is

$$\nabla^{(R)} f(\mathbf{z}) = (\nabla_{\mathbf{z}^*} f_{\mathbf{z}})^T \mathbf{z}^* = \mathbf{z}^H \nabla_{\mathbf{z}^*} f_{\mathbf{z}} \ , \tag{5B.4}$$

which is the SG pre-multiplied by $\mathbf{z}^H$. The updates (5B.2) become, for this case of scalar relative change,

$$\mathbf{z}_{k+1} = \mathbf{z}_k - \mu \mathbf{z}_k^H \nabla_{\mathbf{z}^*} f_{\mathbf{z}} \mathbf{z}_k = \mathbf{z}_k - \mu_{\mathbf{z}} \mathbf{z}_k \mathbf{z}_k^H \nabla_{\mathbf{z}^*} f \quad . \tag{5B.5}$$

Compared with (5A.10), the second term is multiplied by a square matrix of rank 1.

If $\delta \mathbf{z} = \boldsymbol{\mathcal{E}} \mathbf{z}$, then

$$\begin{aligned} f_{\mathbf{z}}(\mathbf{z} + \boldsymbol{\mathcal{E}}\mathbf{z}, \mathbf{z}^* + \boldsymbol{\mathcal{E}}^* \mathbf{z}^*) &\approx f_{\mathbf{z}}(\mathbf{z}, \mathbf{z}^*) + 2\,\mathrm{Re}\left\langle \boldsymbol{\mathcal{E}}\mathbf{z}, \nabla_{\mathbf{z}^*} f_{\mathbf{z}} \right\rangle \\ &= f_{\mathbf{z}}(\mathbf{z}, \mathbf{z}^*) + 2\,\mathrm{Re}\,\mathrm{Trace}\{\mathbf{z} \nabla_{\mathbf{z}^*} f_{\mathbf{z}}^H \boldsymbol{\mathcal{E}}\} \quad . \\ &= f_{\mathbf{z}}(\mathbf{z}, \mathbf{z}^*) + 2\,\mathrm{Re}\left\langle \boldsymbol{\mathcal{E}}, \nabla_{\mathbf{z}^*} f_{\mathbf{z}} \mathbf{z}^H \right\rangle \end{aligned} \tag{5B.6}$$

When $\boldsymbol{\mathcal{E}}$ is aligned with $\nabla_{\mathbf{z}^*} f_{\mathbf{z}} \mathbf{z}^H$ the change rate is maximum, and the RG of $f(\mathbf{z})$ as a matrix is

$$\nabla^{(R)} f(\mathbf{z}) = \nabla_{\mathbf{z}^*} f_{\mathbf{z}} \mathbf{z}^H , \tag{5B.7}$$

which is the SG post-multiplied by $\mathbf{z}^H$. The corresponding iterative minimization rule is

$$\mathbf{z}_{k+1} = \mathbf{z}_k - \mu \nabla_{\mathbf{z}^*} f_{\mathbf{z}} \mathbf{z}_k^H \mathbf{z}_k \quad . \tag{5B.8}$$

Compared with (5A.10), the second term is multiplied by a scalar $\mathbf{z}_k^H \mathbf{z}_k$, i.e. the square of the norm of the equalizer vector. As a result, at each iteration, the effective step-size is adjusted by the magnitude of the equalizer.

In fact, the first case where the RG is a scalar is a special case of the second one where the RG is a matrix. Specifically, when $\boldsymbol{\mathcal{E}}$ is a diagonal matrix, with all the elements on the diagonal identical to the scalar ε, $\boldsymbol{\mathcal{E}}\mathbf{z} = \varepsilon \mathbf{z}$.

When $\mathbf{z}$ is a row vector, we can derive the RG by considering its transpose and analyzing in a similar way as above.

Matrix case

Now consider the RG in the matrix case. For function $f(\mathbf{Z}):\mathbb{C}^{M\times N}\mapsto\mathbb{C}$, suppose $f_{\mathbf{Z}}(\mathbf{Z},\mathbf{Z}^*)=f(\mathbf{Z})$ and $f_{\mathbf{Z}}(\mathbf{Z},\mathbf{Z}^*)$ is analytic with respect to $\mathbf{Z}$ and $\mathbf{Z}^*$ independently. If the perturbation is proportional to the current $\mathbf{Z}$, there are three cases: $\mathbf{Z}$ multiplied by a scalar ε, i.e. $\varepsilon\mathbf{Z}$; $\mathbf{Z}$ pre-multiplied by an $M\times M$ matrix , i.e. $\boldsymbol{\mathcal{E}}\mathbf{Z}$; $\mathbf{Z}$ post-multiplied by an $N\times N$ matrix , i.e. $\mathbf{Z}\boldsymbol{\mathcal{E}}$.

According to (5A.11), when $f(\mathbf{Z})$ is *real-valued*, the first-order Taylor expansion is

$$f_{\mathbf{Z}}(\mathbf{Z}+\delta\mathbf{Z},\mathbf{Z}^*+\delta\mathbf{Z}^*)=f_{\mathbf{Z}}(\mathbf{Z},\mathbf{Z}^*)+2\,\mathrm{Re}\left\langle\delta\mathbf{Z},\nabla_{\mathbf{Z}^*}f_{\mathbf{Z}}\right\rangle+o(\delta\mathbf{Z},\delta\mathbf{Z}^*) \quad (5B.9)$$

When $\delta\mathbf{Z}=\varepsilon\mathbf{Z}$,

$$\begin{aligned} f_{\mathbf{Z}}(\mathbf{Z}+\varepsilon\mathbf{Z},\mathbf{Z}^*+\varepsilon^*\mathbf{Z}^*) &= f_{\mathbf{Z}}(\mathbf{Z},\mathbf{Z}^*)+2\,\mathrm{Re}\left\langle\varepsilon\mathbf{Z},\nabla_{\mathbf{Z}^*}f_{\mathbf{Z}}\right\rangle+o(\varepsilon,\varepsilon^*) \\ &\approx f_{\mathbf{Z}}(\mathbf{Z},\mathbf{Z}^*)+2\,\mathrm{Re}(\varepsilon\,\mathrm{Trace}\{\nabla_{\mathbf{Z}^*}f_{\mathbf{Z}}{}^H\mathbf{Z}\}) \end{aligned} \quad (5B.10)$$

Thus the RG of $f(\mathbf{Z})$ for scalar relative change is

$$\nabla^{(R)}f(\mathbf{Z})=(\mathrm{Trace}\{\nabla_{\mathbf{Z}^*}f_{\mathbf{Z}}{}^H\mathbf{Z}\})^H=\mathrm{Trace}\{\mathbf{Z}^H\nabla_{\mathbf{Z}^*}f_{\mathbf{Z}}\}=\mathrm{Trace}\{\nabla_{\mathbf{Z}^*}f_{\mathbf{Z}}\mathbf{Z}^H\}. \quad (5B.11)$$

When ε is aligned with the direction of $\mathrm{Trace}\{\nabla_{\mathbf{Z}^*}f_{\mathbf{Z}}\mathbf{Z}^H\}$, the change rate is maximum. The adaptation, for minimization, for matrix $\mathbf{Z}$ is

$$\mathbf{Z}_{k+1}=\mathbf{Z}_k-\mu\,\mathrm{Trace}\{\nabla_{\mathbf{Z}^*}f_{\mathbf{Z}}\mathbf{Z}_k^H\}\mathbf{Z}_k. \quad (5B.12)$$

When $\delta\mathbf{Z}=\boldsymbol{\mathcal{E}}\mathbf{Z}$,

$$\begin{aligned} f_{\mathbf{Z}}(\mathbf{Z}+\boldsymbol{\mathcal{E}}\mathbf{Z},\mathbf{Z}^*+\boldsymbol{\mathcal{E}}^*\mathbf{Z}^*) &= f_{\mathbf{Z}}(\mathbf{Z},\mathbf{Z}^*)+2\,\mathrm{Re}\left\langle\boldsymbol{\mathcal{E}}\mathbf{Z},\nabla_{\mathbf{Z}^*}f_{\mathbf{Z}}\right\rangle+o(\boldsymbol{\mathcal{E}},\boldsymbol{\mathcal{E}}^*) \\ &\approx f_{\mathbf{Z}}(\mathbf{Z},\mathbf{Z}^*)+2\,\mathrm{Re}(\mathrm{Trace}\{\nabla_{\mathbf{Z}^*}f_{\mathbf{Z}}{}^H\boldsymbol{\mathcal{E}}\mathbf{Z}\}) \\ &= f_{\mathbf{Z}}(\mathbf{Z},\mathbf{Z}^*)+2\,\mathrm{Re}(\mathrm{Trace}\{(\nabla_{\mathbf{Z}^*}f_{\mathbf{Z}}\mathbf{Z}^H)^H\boldsymbol{\mathcal{E}}\}) \\ &= f_{\mathbf{Z}}(\mathbf{Z},\mathbf{Z}^*)+2\,\mathrm{Re}\left\langle\boldsymbol{\mathcal{E}},\nabla_{\mathbf{Z}^*}f_{\mathbf{Z}}\mathbf{Z}^H\right\rangle \end{aligned} \quad (5B.13)$$

Thus when a pre-multiplying matrix relative change $\boldsymbol{\mathcal{E}}$ is aligned with the corresponding RG

$$\nabla^{(R)} f(\mathbf{Z}) = \nabla_{\mathbf{Z}^*} f_{\mathbf{Z}} \mathbf{Z}^H \ , \tag{5B.14}$$

the change rate is maximum. Compared with the SG of a function with respect to complex matrix, RG is the SG multiplied by $\mathbf{Z}^H$ from the right side. The minimization updating for matrix $\mathbf{Z}$ is

$$\mathbf{Z}_{k+1} = \mathbf{Z}_k - \mu \nabla_{\mathbf{Z}^*} f_{\mathbf{Z}} \mathbf{Z}_k^H \mathbf{Z}_k \ . \tag{5B.15}$$

When $\delta \mathbf{Z} = \mathbf{Z} \boldsymbol{\mathcal{E}}$,

$$\begin{aligned} f_{\mathbf{Z}}(\mathbf{Z} + \mathbf{Z}\boldsymbol{\mathcal{E}}, \mathbf{Z}^* + \mathbf{Z}^* \boldsymbol{\mathcal{E}}^*) &= f_{\mathbf{Z}}(\mathbf{Z}, \mathbf{Z}^*) + 2\,\mathrm{Re}\left\langle \mathbf{Z}\boldsymbol{\mathcal{E}}, \nabla_{\mathbf{Z}^*} f_{\mathbf{Z}} \right\rangle + o(\boldsymbol{\mathcal{E}}, \boldsymbol{\mathcal{E}}^*) \\ &\approx f_{\mathbf{Z}}(\mathbf{Z}, \mathbf{Z}^*) + 2\,\mathrm{Re}(\mathrm{Trace}\{\nabla_{\mathbf{Z}^*} f_{\mathbf{Z}}{}^H \mathbf{Z}\boldsymbol{\mathcal{E}}\}) \\ &= f_{\mathbf{Z}}(\mathbf{Z}, \mathbf{Z}^*) + 2\,\mathrm{Re}\left\langle \boldsymbol{\mathcal{E}}, \mathbf{Z}^H \nabla_{\mathbf{Z}^*} f_{\mathbf{Z}} \right\rangle \end{aligned} \tag{5B.16}$$

When a post-multiplying matrix relative change $\boldsymbol{\mathcal{E}}$ is aligned with the corresponding RG

$$\nabla^{(R)} f(\mathbf{Z}) = \mathbf{Z}^H \nabla_{\mathbf{Z}^*} f_{\mathbf{Z}} \ , \tag{5B.17}$$

the change rate is maximum. The RG in this case is the SG multiplied by $\mathbf{Z}^H$ from the left side. The minimization adaptation for matrix $\mathbf{Z}$ becomes

$$\mathbf{Z}_{k+1} = \mathbf{Z}_k - \mu \mathbf{Z}_k \mathbf{Z}_k^H \nabla_{\mathbf{Z}^*} f_{\mathbf{Z}} \ . \tag{5B.18}$$

The RG for scalar relative change can be seen as a special case of RG for matrix relative change. Comparing (5B.11) with (5B.14) and (5B.17), we can see that the scalar RG is the trace of the matrix RG.

For comparison, the minimization iterations based on gradient descent with the SG and RG of a function of complex variables are listed in the following table:

Table 5-1 SG and RG for complex variables

$f(\cdot)$	**SG**	**RG**		
		Scalar ε	Pre-$\boldsymbol{\mathcal{E}}$	Post-$\boldsymbol{\mathcal{E}}$
$f(z)$ $\mathbb{C} \mapsto \mathbb{C}$	$z_{k+1} = z_k - \mu \dfrac{\partial f_z}{\partial z^*}$	$z_{k+1} = z_k - \mu \dfrac{\partial f_z}{\partial z^*} z_k^* z_k$		
$f(\mathbf{z})$ $\mathbb{C}^{N\times 1} \mapsto \mathbb{C}$	$\mathbf{z}_{k+1} = \mathbf{z}_k - \mu \nabla_{\mathbf{z}^*} f_{\mathbf{z}}$	$\mathbf{z}_{k+1} = \mathbf{z}_k - \mu \mathbf{z}_k \mathbf{z}_k^H \nabla_{\mathbf{z}^*} f_{\mathbf{z}}$	$\mathbf{z}_{k+1} = \mathbf{z}_k - \mu \nabla_{\mathbf{z}^*} f_{\mathbf{z}} \mathbf{z}_k^H \mathbf{z}_k$	
$f(\mathbf{Z})$ $\mathbb{C}^{M\times N} \mapsto \mathbb{C}$	$\mathbf{Z}_{k+1} = \mathbf{Z}_k - \mu \nabla_{\mathbf{Z}^*} f_{\mathbf{Z}}$	$\mathbf{Z}_{k+1} = \mathbf{Z}_k - \mu \mathrm{Trace}\{\nabla_{\mathbf{Z}^*} f_{\mathbf{Z}} \mathbf{Z}_k^H\} \mathbf{Z}_k$	$\mathbf{Z}_{k+1} = \mathbf{Z}_k - \mu \nabla_{\mathbf{Z}^*} f_{\mathbf{Z}} \mathbf{Z}_k^H \mathbf{Z}_k$	$\mathbf{Z}_{k+1} = \mathbf{Z}_k - \mu \mathbf{Z}_k \mathbf{Z}_k^H \nabla_{\mathbf{Z}^*} f_{\mathbf{Z}}$

Appendix 5C

Standard Gradient and Relative Gradient for a Special Case

In practical applications, the function of the variable is usually defined as the expectation of some other function, i.e. $f(z) = E[G(y)]$.

Scalar case

Suppose z is a complex scalar, and function

$$f(z) = E[G(y)] = E[G(zx)], \tag{5C.1}$$

is the expectation of *real-valued* function $G(y)$, with x being a known scalar, and $y = zx$.

For function $G_y(y, y^*) = G(y)$, for a small perturbation δy at y, the Taylor expansion of G_y is

$$G_y(y + \delta y, y^* + \delta y^*) = G_y(y, y^*) + \frac{\partial G_y}{\partial y}\delta y + \frac{\partial G_y}{\partial y^*}\delta y^* + o(\delta y, \delta y^*) . \tag{5C.2}$$

Substituting (5C.2) into the function (5C.1), we have

$$\begin{aligned} f(z + \delta z) &= E[G((z + \delta z)x)] = E[G(y + \delta zx)] = E[G_y(y + \delta zx, y^* + \delta z^* x^*)] \\ &= E[G_y(y, y^*)] + E[\frac{\partial G_y}{\partial y}\delta zx] + E[\frac{\partial G_y}{\partial y^*}\delta z^* x^*] + o(\delta z, \delta z^*) \\ &\approx E[G_y(y, y^*)] + 2E[\mathrm{Re}\frac{\partial G_y}{\partial y}\delta zx] \\ &= f(z) + 2E[\mathrm{Re}\,\delta z\frac{\partial G_y}{\partial y}x] \end{aligned} \tag{5C.3}$$

Thus the SG of $f(z)$ is $(E[\frac{\partial G_y}{\partial y}x])^* = E[\frac{\partial G_y}{\partial y^*}x^*]$. If we replace δz with RG defined as εz, and derive the Taylor expansion in a similar way as the above, we will have that

$$f(z+\varepsilon z) = f(z) + 2E[\mathrm{Re}\,\varepsilon z \frac{\partial G_y}{\partial y}x] + o(\varepsilon). \tag{5C.4}$$

When ε is in alignment with $(E[z\frac{\partial G_y}{\partial y}x])^* = E[\frac{\partial G_y}{\partial y^*}x^*z^*]$, maximum change rate is achieved. This means that the RG of $f(z)$ for scalar relative change is $E[\frac{\partial G_y}{\partial y^*}x^*z^*]$.

For the scalar complex variable z, the minimization adaptation for z with the stochastic RG and SG is as follows, respectively:

$$\text{SG: } z_{k+1} = z_k - \mu\frac{\partial G_y}{\partial y^*}x_k^* \tag{5C.5}$$

$$\text{RG: } z_{k+1} = z_k - \mu\frac{\partial G_y}{\partial y^*}x_k^*z_k^*z_k \tag{5C.6}$$

where in the RG case, the step-size of the updating is adjusted by the square of the modulus of the variable, i.e. $z_k^*z_k$.

Vector case

Now consider the case when the variable is a complex column vector,

$$f(\mathbf{z}) = E[G(y)] = E[G(\mathbf{z}^T\mathbf{x})], \tag{5C.7}$$

where $G(y)$ is *real-valued* function, $\mathbf{z}$ is applied on a column vector $\mathbf{x}$, i.e. $y = \mathbf{z}^T\mathbf{x}$, and y is a scalar.

As in the complex scalar variable case, substituting (5C.2) into the function (5C.7), we have

$$\begin{aligned} f(\mathbf{z}+\delta\mathbf{z}) &= E[G((\mathbf{z}+\delta\mathbf{z})^T\mathbf{x})] = E[G(y+\delta\mathbf{z}^T\mathbf{x})] = E[G_y(y+\delta\mathbf{z}^T\mathbf{x}, y^*+\delta\mathbf{z}^H\mathbf{x}^*)] \\ &= E[G_y(y,y^*)] + E[\frac{\partial G_y}{\partial y}\delta\mathbf{z}^T\mathbf{x}] + E[\frac{\partial G_y}{\partial y^*}\delta\mathbf{z}^H\mathbf{x}^*] + o(\delta\mathbf{z},\delta\mathbf{z}^*) \\ &\approx E[G_y(y,y^*)] + 2E[\mathrm{Re}\frac{\partial G_y}{\partial y}\delta\mathbf{z}^T\mathbf{x}] \\ &= E[G_y(y,y^*)] + 2E[\mathrm{Re}\frac{\partial G_y}{\partial y}\mathbf{x}^T\delta\mathbf{z}] \\ &= E[G_y(y,y^*)] + 2\,\mathrm{Re}\left\langle \delta\mathbf{z}, E\frac{\partial G_y}{\partial y^*}\mathbf{x}^* \right\rangle \\ &= f(\mathbf{z}) + 2\,\mathrm{Re}\left\langle \delta\mathbf{z}, E\frac{\partial G_y}{\partial y^*}\mathbf{x}^* \right\rangle \end{aligned} \tag{5C.8}$$

Thus the SG of $f(\mathbf{z})$ is

$$\nabla_{\mathbf{z}^*} f(\mathbf{z}) = E[\frac{\partial G_y}{\partial y^*}\mathbf{x}^*] \ . \tag{5C.9}$$

From the analysis of RG of a general function $f(\mathbf{z})$, we know that the RG for the vector case can be defined in two ways: $\mathcal{E}\mathbf{z}$ and $\boldsymbol{\mathcal{E}}\mathbf{z}$. The first can be considered as a special case of the second one, where the matrix is a diagonal matrix with identical elements. If we consider the RG for matrix relative change, and replace $\delta\mathbf{z}$ with $\boldsymbol{\mathcal{E}}\mathbf{z}$ to derive the Taylor expansion as the above, we will have that

$$\begin{aligned} f(\mathbf{z}+\boldsymbol{\mathcal{E}}\mathbf{z}) &= f(\mathbf{z}) + 2\,\mathrm{Re}\left\langle \boldsymbol{\mathcal{E}}\mathbf{z}, E\frac{\partial G_y}{\partial y^*}\mathbf{x}^* \right\rangle + o(\boldsymbol{\mathcal{E}}) \\ &= f(\mathbf{z}) + 2\,\mathrm{Re}\left\langle \boldsymbol{\mathcal{E}}, E\frac{\partial G_y}{\partial y^*}\mathbf{x}^*\mathbf{z}^H \right\rangle + o(\boldsymbol{\mathcal{E}}) \end{aligned} \ . \tag{5C.10}$$

When $\boldsymbol{\mathcal{E}}$ is in alignment with $E[\frac{\partial G_y}{\partial y^*}\mathbf{x}^*\mathbf{z}^H]$, the change rate is maximum. This means that the RG of $f(\mathbf{z})$ for matrix relative change is

$$\nabla^{(R)} f(\mathbf{z}) = E[\frac{\partial G_y}{\partial y^*}\mathbf{x}^*\mathbf{z}^H]. \tag{5C.11}$$

When $\boldsymbol{\mathcal{E}}$ is a diagonal matrix with identical elements, i.e. $\boldsymbol{\mathcal{E}} = \varepsilon\mathbf{I}$, the change rate is maximum when the scalar ε has the value $\varepsilon = \text{Trace}\{E[\frac{\partial G_y}{\partial y^*}\mathbf{x}^*\mathbf{z}^H]\} = E[\frac{\partial G_y}{\partial y^*}\mathbf{z}^H\mathbf{x}^*]$. Thus the RG of $f(\mathbf{z})$ for scalar relative change is

$$\nabla^{(R)} f(\mathbf{z}) = E[\frac{\partial G_y}{\partial y^*}\mathbf{z}^H\mathbf{x}^*] \tag{5C.12}$$

For the vector case, the iterative updates of $\mathbf{z}$ with the stochastic RG and SG is as follows, respectively:

$$\text{SG: } \mathbf{z}_{k+1} = \mathbf{z}_k - \mu\frac{\partial G_y}{\partial y^*}\mathbf{x}_k^* \tag{5C.13}$$

$$\text{RG}(\varepsilon)\text{: } \mathbf{z}_{k+1} = \mathbf{z}_k - \mu\frac{\partial G_y}{\partial y^*}\mathbf{z}_k^H\mathbf{x}_k^*\mathbf{z}_k = \mathbf{z}_k - \mu\frac{\partial G_y}{\partial y^*}\mathbf{z}_k\mathbf{z}_k^H\mathbf{x}_k^* \tag{5C.14}$$

$$\text{RG}(\boldsymbol{\mathcal{E}})\text{: } \mathbf{z}_{k+1} = \mathbf{z}_k - \mu\frac{\partial G_y}{\partial y^*}\mathbf{x}_k^*\mathbf{z}_k^H\mathbf{z}_k \tag{5C.15}$$

Matrix case

At last we consider the matrix case. Similarly, the function of the matrix can be expressed as

$$f(\mathbf{Z}) = E[G(\mathbf{y})] = E[G(\mathbf{Z}\mathbf{x})] \tag{5C.16}$$

where $G(\mathbf{y})$ a *real-valued* function, and $\mathbf{Z}$ is applied on a known column vector $\mathbf{x}$, i.e. $\mathbf{y} = \mathbf{Z}\mathbf{x}$ is a column vector.

For function $G_{\mathbf{y}}(\mathbf{y}, \mathbf{y}^*) = G(\mathbf{y})$, when there is a small perturbation $\delta\mathbf{y}$ at $\mathbf{y}$, the Taylor expansion of $G_{\mathbf{y}}$ is

$$G_{\mathbf{y}}(\mathbf{y}+\delta\mathbf{y}, \mathbf{y}^*+\delta\mathbf{y}^*) = G_{\mathbf{y}}(\mathbf{y}, \mathbf{y}^*) + \left\langle \delta\mathbf{y}, \nabla_{\mathbf{y}^*} G_{\mathbf{y}} \right\rangle + \left\langle \delta\mathbf{y}^*, \nabla_{\mathbf{y}} G_{\mathbf{y}} \right\rangle + o(\delta\mathbf{y}, \delta\mathbf{y}^*) \quad \text{(5C.17)}$$

Substituting (5C.17) into the function (5C.16),

$$\begin{aligned} f(\mathbf{Z}+\delta\mathbf{Z}) &= E[G((\mathbf{Z}+\delta\mathbf{Z})\mathbf{x})] = E[G(\mathbf{y}+\delta\mathbf{Z}\mathbf{x})] = E[G_{\mathbf{y}}(\mathbf{y}+\delta\mathbf{Z}\mathbf{x}, \mathbf{y}^*+\delta\mathbf{Z}^*\mathbf{x}^*)] \\ &= E[G_{\mathbf{y}}(\mathbf{y}, \mathbf{y}^*)] + E\left\langle \delta\mathbf{Z}\mathbf{x}, \nabla_{\mathbf{y}^*} G_{\mathbf{y}} \right\rangle + E\left\langle \delta\mathbf{Z}^*\mathbf{x}^*, \nabla_{\mathbf{y}} G_{\mathbf{y}} \right\rangle + o(\delta\mathbf{y}) \\ &= E[G_{\mathbf{y}}(\mathbf{y}, \mathbf{y}^*)] + 2E[\mathrm{Re}\left\langle \delta\mathbf{Z}\mathbf{x}, \nabla_{\mathbf{y}^*} G_{\mathbf{y}} \right\rangle] + o(\delta\mathbf{y}) \\ &= E[G_{\mathbf{y}}(\mathbf{y}, \mathbf{y}^*)] + 2\,\mathrm{Re}\left\langle \delta\mathbf{Z}, E\nabla_{\mathbf{y}^*} G_{\mathbf{y}} \mathbf{x}^H \right\rangle + o(\delta\mathbf{y}) \\ &= f(\mathbf{Z}) + 2\,\mathrm{Re}\left\langle \delta\mathbf{Z}, E\nabla_{\mathbf{y}^*} G_{\mathbf{y}} \mathbf{x}^H \right\rangle + o(\delta\mathbf{y}) \end{aligned} \quad \text{(5C.18)}$$

Thus the SG of $f(\mathbf{Z})$ is $\nabla_{\mathbf{Z}^*} f(\mathbf{Z}) = E[\nabla_{\mathbf{y}^*} G_{\mathbf{y}} \mathbf{x}^H]$.

If we replace $\delta\mathbf{Z}$ with $\boldsymbol{\mathcal{E}}\mathbf{Z}$, and derive the Taylor expansion in a similar way as in previous parts, we will have that

$$\begin{aligned} f(\mathbf{Z}+\boldsymbol{\mathcal{E}}\mathbf{Z}) &= f(\mathbf{Z}) + 2\,\mathrm{Re}\left\langle \boldsymbol{\mathcal{E}}\mathbf{Z}, E\nabla_{\mathbf{y}^*} G_{\mathbf{y}} \mathbf{x}^H \right\rangle + o(\boldsymbol{\mathcal{E}}) \\ &= f(\mathbf{Z}) + 2\,\mathrm{Re}\left\langle \boldsymbol{\mathcal{E}}, E\nabla_{\mathbf{y}^*} G_{\mathbf{y}} \mathbf{y}^H \right\rangle + o(\boldsymbol{\mathcal{E}}) \end{aligned} \quad \text{(5C.19)}$$

When $\boldsymbol{\mathcal{E}}$ is in alignment with $E[\nabla_{\mathbf{y}^*} G_{\mathbf{y}} \mathbf{y}^H]$, maximum change rate is achieved. This means that the RG of $f(\mathbf{Z})$ with respect to $\mathbf{Z}$ for pre-multiplying matrix change is

$$\nabla^{(R)} f(\mathbf{Z}) = E[\nabla_{\mathbf{y}^*} G_{\mathbf{y}} \mathbf{y}^H] \quad \text{(5C.20)}$$

Comparing the relation between the RG and SG, we can see that

$$\nabla^{(R)} f(\mathbf{Z}) = E[\nabla_{\mathbf{y}^*} G_{\mathbf{y}} \mathbf{y}^H] = E[\nabla_{\mathbf{y}^*} G_{\mathbf{y}} \mathbf{x}^H \mathbf{Z}^H] = \nabla_{\mathbf{Z}^*} f(\mathbf{Z}) \mathbf{Z}^H \tag{5C.21}$$

If $\delta \mathbf{Z} = \mathbf{Z}\boldsymbol{\mathcal{E}}$, then

$$f(\mathbf{Z} + \mathbf{Z}\boldsymbol{\mathcal{E}}) = f(\mathbf{Z}) + 2\,\mathrm{Re}\left\langle \boldsymbol{\mathcal{E}}, E\mathbf{Z}^H \nabla_{\mathbf{y}^*} G_{\mathbf{y}} \mathbf{x}^H \right\rangle + o(\boldsymbol{\mathcal{E}}) \tag{5C.22}$$

When $\boldsymbol{\mathcal{E}}$ is in alignment with $E[\mathbf{Z}^H \nabla_{\mathbf{y}^*} G_{\mathbf{y}} \mathbf{x}^H]$, maximum change rate is achieved. This means that the RG of $f(\mathbf{Z})$ with respect to $\mathbf{Z}$ for post-multiplying matrix change is

$$\nabla^{(R)} f(\mathbf{Z}) = E[\mathbf{Z}^H \nabla_{\mathbf{y}^*} G_{\mathbf{y}} \mathbf{x}^H] \tag{5C.23}$$

As mentioned before when the variable is a matrix, RG for scalar relative change can be considered as a special case of RG for matrix relative change. When $\delta \mathbf{Z} = \varepsilon \mathbf{Z}$, when $\varepsilon = \mathrm{Trace}\{E[\nabla_{\mathbf{y}^*} G_{\mathbf{y}} \mathbf{y}^H]\} = \mathrm{Trace}\{E[\mathbf{Z}^H \nabla_{\mathbf{y}^*} G_{\mathbf{y}} \mathbf{x}^H]\}$ the maximum change rate is achieved. With scalar relative change, the elements in matrix $\mathbf{Z}$ will change proportionally at each iteration, so the performance is not expected to be good.

The iterative adaptation for $\mathbf{Z}$ with the stochastic RG and SG is as follows, respectively:

$$\text{SG: } \mathbf{Z}_{k+1} = \mathbf{Z}_k - \mu \nabla_{\mathbf{y}^*} G_{\mathbf{y}} \mathbf{x}_k^H \tag{5C.24}$$

$$\text{RG}(\varepsilon)\text{: } \mathbf{Z}_{k+1} = \mathbf{Z}_k - \mu\, \mathrm{Trace}\{E[\nabla_{\mathbf{y}^*} G_{\mathbf{y}} \mathbf{y}_k^H]\} \mathbf{Z}_k \tag{5C.25}$$

$$\text{RG(pre } \boldsymbol{\mathcal{E}}\text{): } \mathbf{Z}_{k+1} = \mathbf{Z}_k - \mu \nabla_{\mathbf{y}^*} G_{\mathbf{y}} \mathbf{y}_k^H \mathbf{Z}_k \tag{5C.26}$$

$$\text{RG(post } \boldsymbol{\mathcal{E}}\text{): } \mathbf{Z}_{k+1} = \mathbf{Z}_k - \mu \mathbf{Z}_k \mathbf{Z}_k^H \nabla_{\mathbf{y}^*} G_{\mathbf{y}} \mathbf{x}_k^H \tag{5C.27}$$

Chapter 6

Channel Shortening for OFDM with Relative Gradient

6.1 Introduction

In orthogonal frequency division multiplexing (OFDM) systems, a cyclic prefix (CP) is usually used to reduce inter-carrier interference (ICI) and inter-symbol interference (ISI). To be effective, the length of the CP should be longer than that of the channel impulse response. The transmission of a long redundant CP with each OFDM data block takes power and reduces the data rate. If the CP is not long enough for long channel impulse responses, output processing for *channel shortening* is implemented to shorten the effective channel impulse response. The CP results in redundancy or duplication in the transmitted CP symbols and the symbols in the OFDM block, which can be employed for channel shortening. In this chapter, we will study use of the relative gradient to achieve channel

shortening in OFDM systems, which provides an alternative to the standard gradient in channel shortening algorithms. Two blocks of shortener outputs that should remain redundant as a result of the redundant CP symbols and corresponding symbols in the OFDM block will be used for adaptation each time. The criteria functions used for adaptation are squared-error criteria that have been previously defined in [1], [2]. Our block processing scheme allows the use of relative gradient in the matrix adaptation containing the response of the shortener, with Toeplitz constraint. Also, we will use inter-symbol interference (ISI), together with the power ratio proposed in previous work [2] as the criteria to judge performance.

In Section 6.2, the model of the OFDM system is introduced, including the basic concepts and notation. In Section 6.3, some previous work on channel shorting that used the redundancy of the CP is reviewed. Considering multiple pairs of the CP and the corresponding OFDM symbols, we formulate the problem in matrix form and use the relative gradient for the cost function optimization. In Section 6.4, preliminary results for different algorithms are shown. We also discuss the potential directions of future research that would lead to a comprehensive evaluation of channel shortening algorithms.

6.2 Review of OFDM

In this section, we review the system model of OFDM. Some of the notation used in this chapter is the same as that used in the previous chapters for equalization problems in the single carrier systems, but since the models are different, the meaning of the notation will be clarified. Generally, an OFDM system splits the source stream into N sub-streams.

These sub-streams are used to modulate N parallel subcarriers, which employ a narrow bandwidth each and are orthogonal to each other. The symbol rate on each subcarrier is reduced by a factor of N compared to a single carrier modulation scheme that occupies the whole bandwidth.

Typically in OFDM systems guard intervals are used to combat inter-symbol interference (ISI) and inter-carrier interference (ICI). The guard interval can be zero padding or cyclic prefix (CP) as mentioned for the single carrier scheme in Chapter 3 [3]. In practice, the CP is more commonly used in OFDM because it allows the use of a single-tap equalizer at the receiver side, provided the CP length is longer than the channel impulse response.

A model of the OFDM system is shown in Fig. 6.1. The input stream $\{\tilde{S}(n)\}$ is divided into N substreams, and at time k, the symbols in the size-N block can be written as $\tilde{\mathbf{S}}_k = [\tilde{S}_k(1), ..., \tilde{S}_k(N)]^T$.

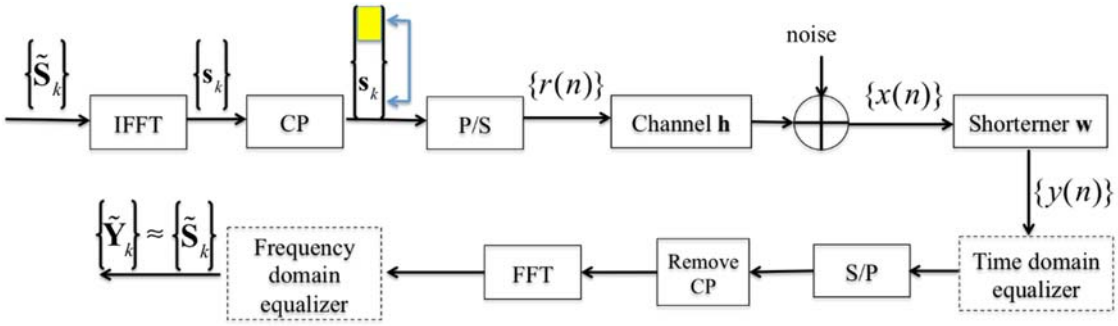

Fig. 6.1 OFDM system

OFDM symbols can be efficiently generated using inverse fast Fourier transform (IFFT). Denote $\mathbf{F}$ as the $N \times N$ normalized IFFT matrix with entries $F_{mn} = \frac{1}{\sqrt{N}} e^{j\frac{2\pi}{N}mn}$, $m, n = 0, 1, ..., N-1$, then the OFDM symbol block at time k can be expressed as

$$\mathbf{s}_k = \mathbf{F}\tilde{\mathbf{S}}_k . \tag{6.1}$$

The block $\tilde{\mathbf{S}}_k$ and $\mathbf{s}_k$ are usually called the *frequency* domain signal and the *time* domain signal respectively, due to the fact that IFFT is used to convert signal in the frequency domain to that in the time domain.

Suppose the length of the CP is ν. With the CP, the last ν symbols of an OFDM block is copied and padded at the beginning of each block. Let the OFDM block at time k be $\mathbf{s}_k = [s_k(1), ..., s_k(N)]^T$. From the figure, it can be understood that the last ν symbols in the k-th block, i.e. $s_k(N-\nu+1), s_k(N-\nu+2), ..., s_k(N)$, are copied and padded ahead of the OFDM block. The OFDM blocks with their CP are then converted into a serial sequence $\{r(n)\}$, where

$$r(k(N+\nu)+i) = \begin{cases} s_k(N-\nu+i) & i = 1, .., \nu \\ s_k(i-\nu) & i = 1+\nu, .., N+\nu \end{cases} , \tag{6.2}$$

which can be seen from Fig. 6.2.

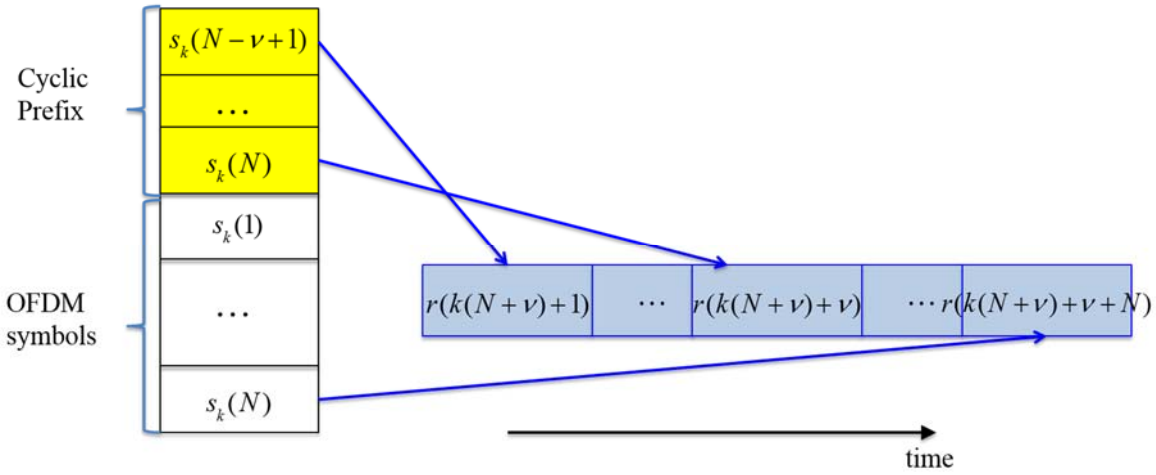

Fig. 6.2 Parallel to serial transmission

The serial CP padded OFDM sequence $\{r(n)\}$ is transmitted through the channel $\mathbf{h} = [h(0), h(1), ..., h(L)]^T$, which generates the channel output sequence $\{x(n)\}$. Omitting noise, the output of the channel can be expressed as

$$x(n) = \mathbf{h}^T \mathbf{r}_n, \tag{6.3}$$

where $\mathbf{r}_n = [r(n), ..., r(n-L)]^T$.

When the channel length is larger than the length of the CP, such as in digital subscriber loops (DSL) [4], the ICI cannot be removed even with the use of the CP. In that case, there is a need for a channel shortener so that the maximum excess delay of the shortened channel does not exceed the length of the CP. A channel shortener can be considered as a generalization of channel equalizer. Channel shortening results in a non-ideal but shorter channel, and can be achieved with a shorter linear filter compared to an equalizer. When the cascaded channel-shortener system has the ideal response with a single non-zero tap, the shortener becomes an equalizer.

In Fig. 6.1, there are two dotted blocks indicating the possible use of time domain equalizer and frequency domain equalizer. In OFDM, equalization can be realized simply with a single-tap equalizer in the frequency domain. As a result, shortener is usually combined with a frequency-domain equalizer after FFT. However, there have been many studies of equalization in the time domain [5]–[7], [8], [9]. In fact, there is a connection between the channel shortener and time domain equalizer, and this will be discussed briefly later in this section.

The impulse response of the shortener is denoted as $\mathbf{w} = [w(0), w(1), ..., w(M)]^T$. Transmitting the output of the channel $\{x(n)\}$ through the shortener gives the shortener output sequence $\{y(n)\}$, where the output symbol can be expressed as

$$y(n) = \mathbf{w}_n{}^T \mathbf{x}_n \ , \tag{6.4}$$

with $\mathbf{x}_n = [x(n), ..., x(n-M)]^T$. Specifically, the output with the shortener at the n'-th iteration in adaptive updating can be expressed as $y^{(n')}(n) = \mathbf{w}_{n'}{}^T \mathbf{x}_n$.

Let $\mathbf{c}$ be the cascaded response of the channel and the shortener, i.e. $\mathbf{c} \triangleq \mathbf{h} * \mathbf{w}$. Suppose we want to shorten the channel length to $K+1$, then $\mathbf{c}$ should have a window of length $K+1$ containing the major taps, with the taps outside the window having very small magnitude. When $K+1 \le \nu$, the ISI that affects the last $\nu - K$ symbols in the cyclic prefix will be the same as that which affects the corresponding symbols in the OFDM block. Specifically, when the channel is shortened to have length ν, i.e. $K+1 = \nu$, the last symbol in the cyclic prefix should be the same as the last symbol in the OFDM block; and when $K+1 < \nu$, the equality relation should hold for multiple pairs of cyclic prefix symbols and corresponding data symbols in the OFDM block. Let Δ be the designed delay value of the

cascaded system, which indicates the starting location of the effective length-($K+1$) window. Selection of a good delay parameter Δ was considered in [2] and [10], the goal being to concentrate the shortened channel within the window. Including the delay Δ, we can expect that with a shortener,

$$y(k(N+\nu)+\Delta+i) = y(k(N+\nu)+\Delta+i+N), \tag{6.5}$$

where index i takes values within the range $\nu' \le i \le \nu$, with ν' an integer satisfying $1 \le \nu' \le \nu$ and determined by the effective length of the shortened channel.

The redundancy of the CP has been used for channel shortening and also equalization in OFDM systems [2] – [4], [8]–[10]. In equalization algorithms, we require that every CP symbol be equal to the corresponding OFDM symbol [8], [9], [13]. In other words, equalization enforces a more strict constraint on the redundancy of the CP. In this chapter, we focus on obtaining channel shortening and consider redundancy of a subset of CP symbols, instead of every CP symbol. We will see in Section 6.4 through simulations that equalization can be achieved to some extent with our proposed shortening algorithm. In the next section, we will review the shortening algorithms and explain our proposed algorithm based on the relative gradient.

6.3 Channel Shortening Algorithms

In this section, the algorithms employing the information of the guard intervals for channel shortening will be reviewed first. Based on the ideas of considering redundancy due to the CP, we consider use of the relative gradient for adaptive blind channel shortening.

6.3.1 Review of Shortening Algorithms

We have shown in the previous section that when the length of the cascade system of the channel and the shortener is shorter than or equal to ν, the redundancy between part of the CP and the corresponding OFDM symbols is maintained. The extent of the redundancy is determined by the length of the shortened channel.

In [11], the authors proposed the *Multicarrier equalization by restoration of redundancy* (Merry) algorithm. In other words, the goal is to shorten the channel to the length of the CP. The cost function is defined to minimize the difference between the last symbol in the CP and its corresponding OFDM symbol (last symbol in OFDM block). Specifically, for the k -th OFDM block, the last symbol in the CP is a copy of the OFDM symbol $s_k(N)$. When transmitted in serial sequence, the last CP symbol is denoted as $s(k(N+\nu)+\nu)$, and it is equal to the last symbol in the OFDM block $s(k(N+\nu)+\nu+N)$. As a result, the Merry objective function has the expression

$$J_{Merry}(\mathbf{w}) = E[|\, y(k(N+\nu)+\nu+\Delta) - y(k(N+\nu)+\nu+N+\Delta)\,|^2]. \qquad (6.6)$$

The shortener is updated once with one transmitted OFDM block. With a stochastic gradient descent method, the adaptation for vector $\mathbf{w}$ is

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mu\left(y^{(k)}(k(N+\nu)+\nu+\Delta) - y^{(k)}(k(N+\nu)+\nu+N+\Delta)\right)\cdot \left(\mathbf{x}_{k(N+\nu)+\nu+\Delta} - \mathbf{x}_{k(N+\nu)+\nu+N+\Delta}\right)^*, \qquad (6.7)$$

where the shortened outputs are from the shortener at the *k*-th iteration. Since a shortener with all the elements zero will make the Merry cost function (6.6) null, a normalization may be implemented at the end of each iteration.

Since there are a total number of ν symbols in the CP, an intuitive generalization of the Merry algorithm is to use more information at each iteration. In [2], the authors proposed the *forced redundancy with optional data omission* (Frodo) algorithm by exploiting the redundancy of multiple pairs of the CP symbols and the corresponding OFDM symbols. The cost function of the Frodo algorithm is

$$J_{Frodo}(\mathbf{w}) = \sum_{i \in S_f} E[|\, y(k(N+\nu)+i+\Delta) - y(k(N+\nu)+i+N+\Delta)\,|^2], \quad (6.8)$$

where S_f is a *subset* of $\{1,2,...,\nu\}$. Note that cost function of the Frodo algorithm can be seen as a generalization of the Merry algorithm; when *i* takes single value ν, the Frodo cost function reduces to that of the Merry algorithm.

A cost function similar to (6.8) was used in [14] and [9] for equalization. In [14], it is pointed out that by requiring all the symbols in the cyclic prefix to equal the corresponding OFDM symbol, equalization can be achieved in the time domain. The equalization approach of [14] can be seen as a special case of channel shortening [2] with the goal to shorten the channel to a single tap.

In [2], it was shown that the optimization of (6.8) with a constraint for avoiding $\mathbf{w} = \mathbf{0}$ is equivalent to a group of optimization problems, some of which can be solved with an existing maximizing algorithm in [15] based on an iterative eigen-decomposition. The adaptation for shortener $\mathbf{w}$ in the Frodo algorithm avoids normalization by division at each iteration as in the Merry algorithm. Although the Frodo algorithm does not use gradient descent method directly, the idea of looking at a block of CP symbols provides an interesting way to use the redundancy information. In the next section, we will use the cost function of the Frodo algorithm, and use the relative gradient for gradient descent.

6.3.2 Shortening Algorithms with Relative Gradient

In this section, relative gradient will be used for channel shortening based on the cost function of the Frodo algorithm. Recall the cost function of Frodo (Merry is a special case when $i = \nu$) is

$$J_{Frodo}(\mathbf{w}_k) = \sum_{i \in S_f} E\left[| y\left(k(N+\nu)+i+\Delta\right) - y\left(k(N+\nu)+N+i+\Delta\right)|^2 \right].$$

At each iteration, the outputs are from the shortener from the previous iteration. With the standard gradient, the adaptation for the shortener is

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mu \sum_{i \in S_f} E[\ \left(y^{(k)}\left(k(N+\nu)+i+\Delta\right) - y^{(k)}\left(k(N+\nu)+N+i+\Delta\right)\right) \cdot \left(\mathbf{x}_{k(N+\nu)+i+\Delta} - \mathbf{x}_{k(N+\nu)+N+i+\Delta}\right)^* \ ] \quad . \quad (6.9)$$

At convergence, we expect that the coefficients for $\mathbf{w}_k$ and $\mathbf{w}_{k+1}$ would ideally be equal. Thus, at the steady state, the *m*-th element of vector $\mathbf{x}_{k(N+\nu)+i+\Delta}$ and $\mathbf{x}_{k(N+\nu)+N+i+\Delta}$, expressed as $x\left(k(N+\nu)+i+\Delta-m\right)$ and $x\left(k(N+\nu)+N+i+\Delta-m\right)$, $m = 0,1,...M$ should satisfy

$$\sum_{i \in S_f} E[\ \left(y^{(k)}\left(k(N+\nu)+i+\Delta\right) - y^{(k)}\left(k(N+\nu)+N+i+\Delta\right)\right) \left(x\left(k(N+\nu)+i+\Delta-m\right) - x\left(k(N+\nu)+N+i+\Delta-m\right)\right)^* \ ] = 0 \quad , \quad (6.10)$$

and this is equivalent to

$$\sum_{i \in S_f} E[\ \left(y^{(k)}\left(k(N+\nu)+i+\Delta\right) - y^{(k)}\left(k(N+\nu)+N+i+\Delta\right)\right) x\left(k(N+\nu)+i+\Delta-m\right)^* \ ]$$
$$= \sum_{i \in S_f} E[\ \left(y^{(k)}\left(k(N+\nu)+i+\Delta\right) - y^{(k)}\left(k(N+\nu)+N+i+\Delta\right)\right) x\left(k(N+\nu)+N+i+\Delta-m\right)^* \ ] \quad . \quad (6.11)$$

Let $\alpha = k(N+\nu)+\Delta$ and $\beta = k(N+\nu)+\Delta+N$, then equation (6.11) can be expressed in a simpler form as

$$\begin{aligned} &\sum_{i\in S_f} E[\ \left(y^{(k)}(\alpha+i)-y^{(k)}(\beta+i)\right)x(\alpha+i-m)^*\ ] \\ &=\sum_{i\in S_f} E[\ \left(y^{(k)}(\alpha+i)-y^{(k)}(\beta+i)\right)x(\beta+i-m)^*\ ] \end{aligned} . \tag{6.12}$$

Suppose the shortener **w** is approximated to be doubly-infinite in length. Following the proof for Bussgang condition in [16], we see that

$$y(\alpha+i)=\sum_{m=-\infty}^{+\infty} x(\alpha+i-m)w(m)\ , \tag{6.13}$$

and for any integer *n*,

$$\begin{aligned} y(\alpha+i-n) &= \sum_{m=-\infty}^{+\infty} x(\alpha+i-n-m)w(m) \\ &= \sum_{m=-\infty}^{+\infty} x(\alpha+i-m)w(m-n) \end{aligned} . \tag{6.14}$$

For a fixed integer *n*, multiplying both sides of equation (6.12) with $w(m-n)^*$ and summing over all integers $-\infty < m < +\infty$, we have

$$\begin{aligned} &\sum_{m=-\infty}^{+\infty}\sum_{i\in S_f} E[\ \left(y^{(k)}(\alpha+i)-y^{(k)}(\beta+i)\right)x(\alpha+i-m)^*\ ]\, w(m-n)^* \\ &=\sum_{i\in S_f} E[\ \left(y^{(k)}(\alpha+i)-y^{(k)}(\beta+i)\right)\left(\sum_{m=-\infty}^{\infty} x(\alpha+i-m)^*\, w(m-n)^*\right)\ ] \\ &=\sum_{i\in S_f} E[\ \left(y^{(k)}(\alpha+i)-y^{(k)}(\beta+i)\right)y(\alpha+i-n)^*\ ] \\ &=\sum_{m=-\infty}^{+\infty}\sum_{i\in S_f} E[\ \left(y^{(k)}(\alpha+i)-y^{(k)}(\beta+i)\right)x(\beta+i-m)^*\ ]\, w(m-n)^* \\ &=\sum_{i\in S_f} E[\ \left(y^{(k)}(\alpha+i)-y^{(k)}(\beta+i)\right)y(\beta+i-n)^*\ ]. \end{aligned} \tag{6.15}$$

The third line and the last line in (6.15) gives

$$\sum_{i\in S_f} E[\ \left(y^{(k)}\left(\alpha+i\right)-y^{(k)}\left(\beta+i\right)\right)\left(y\left(\alpha+i-n\right)-y\left(\beta+i-n\right)\right)^{*}\ ]=0\,, \tag{6.16}$$

which is:

$$\begin{aligned}\sum_{i\in S_f} E[\ &\left(y^{(k)}\left(k(N+\nu)+i+\Delta\right)-y^{(k)}\left(k(N+\nu)+N+i+\Delta\right)\right)\\ &\left(y^{(k)}\left(k(N+\nu)+i+\Delta-n\right)-y^{(k)}\left(k(N+\nu)+N+i+\Delta-n\right)\right)^{*}\ ]=0\end{aligned}\,. \tag{6.17}$$

When $S_f=\{\nu\}$, equation (6.17) corresponds to the steady state of the Merry algorithm. In fact, we can make S_f be a set containing a single index and focus on a single CP and its corresponding OFDM symbol; then at the steady state, we should have ideally

$$\begin{aligned}&\left(y^{(k)}\left(k(N+\nu)+i+\Delta\right)-y^{(k)}\left(k(N+\nu)+N+i+\Delta\right)\right)\\ &\left(y^{(k)}\left(k(N+\nu)+i+\Delta-m\right)-y^{(k)}\left(k(N+\nu)+N+i+\Delta-m\right)\right)^{*}=0\end{aligned}\,. \tag{6.18}$$

This means that the cross-correlation of the difference of the cyclic prefix and the corresponding OFDM symbol with any time lag should be zero. As in the Bussgang condition in the RG Bussgang algorithms, this steady state condition can appear explicitly in the updates.

As in the Bussgang equalization case, the shortener cannot be doubly-infinite in practice. However if we start from an ideal doubly-infinite shortener and keep only the central significant taps, we get an FIR filter, and the equation in (6.18) should still hold approximately.

Let **W** be a Toeplitz matrix of dimension $P\times(P+M)$ containing the impulse response **w** of the shortener in its rows, $0<P\le\nu$. Let the output block of length $P+M$ from the channel be

$$\tilde{\mathbf{x}}_k^{CP} = [x\big((N+\nu)k+\nu+\Delta\big), x\big((N+\nu)k+\nu+\Delta-1\big), \\ ..., x\big((N+\nu)k+\nu+\Delta-P-M+1\big)]^T \quad , \tag{6.19}$$

then $\tilde{\mathbf{y}}_k^{CP} = \mathbf{W}\tilde{\mathbf{x}}_k^{CP} = [y^{(k)}\big((N+\nu)k+\nu+\Delta\big), ..., y^{(k)}\big((N+\nu)k+\nu+\Delta-P+1\big)]^T$

corresponds to the last P symbols in the cyclic prefix. Also let

$$\tilde{\mathbf{x}}_k = [x\big((N+\nu)k+N+\nu+\Delta\big), x\big((N+\nu)k+N+\nu+\Delta-1\big), \\ ..., x\big((N+\nu)k+N+\nu+\Delta-P-M+1\big)]^T \quad , \tag{6.20}$$

then $\tilde{\mathbf{y}}_k = \mathbf{W}\tilde{\mathbf{x}}_k = [y^{(k)}\big((N+\nu)k+N+\nu+\Delta\big), ..., y^{(k)}\big((N+\nu)k+N+\nu+\Delta-P+1\big)]^T$

corresponds to the last P symbols in the OFDM block. The positive integer P indicates the number of pairs of CP and OFDM symbols used for redundancy. With these definitions, the Frodo cost function with $S_f = \{\nu - P + 1, ..., \nu\}$ can be expressed as

$$\tilde{J}_{Frodo}(\mathbf{W}_k) = E[|\,\tilde{\mathbf{y}}_k^{CP} - \tilde{\mathbf{y}}_k\,|^2] = E[|\,\mathbf{W}(\tilde{\mathbf{x}}_k^{CP} - \tilde{\mathbf{x}}_k)\,|^2].$$

With the expectation replaced by instantaneous value, the adaptation for matrix **W** with standard gradient descent is

$$\begin{aligned} \tilde{\mathbf{W}}_k &= \mathbf{W}_k - \mu(\tilde{\mathbf{y}}_k^{CP} - \tilde{\mathbf{y}}_k)(\tilde{\mathbf{x}}_k^{CP} - \tilde{\mathbf{x}}_k)^H \\ \mathbf{W}_{k+1} &= Toeplitz\{\tilde{\mathbf{W}}_k\} \end{aligned} \quad . \tag{6.21}$$

The iterative matrix updates (6.21) with Toeplitz constraint, starting from a Toeplitz matrix, will keep **W** a Toeplitz matrix with repeated rows containing **w**, and this is equivalent to the vector update in (6.9).

From the derivation in Chapter 5, we know that the following relation between the standard gradient and the relative gradient should hold:

$$\nabla_{\mathbf{W}}^{(R)}\tilde{J}(\mathbf{W}) = \nabla_{\mathbf{W}^*}\tilde{J}(\mathbf{W})\mathbf{W}^H . \tag{6.22}$$

Thus the relative gradient in the present case has the expression

$$\nabla_{\mathbf{W}}^{(R)}\tilde{J}(\mathbf{W}) = (\tilde{\mathbf{y}}_k^{CP} - \tilde{\mathbf{y}}_k)(\tilde{\mathbf{x}}_k^{CP} - \tilde{\mathbf{x}}_k)^H \mathbf{W}^H = (\tilde{\mathbf{y}}_k^{CP} - \tilde{\mathbf{y}}_k)(\tilde{\mathbf{y}}_k^{CP} - \tilde{\mathbf{y}}_k)^H . \tag{6.23}$$

With the relative gradient, the adaptation (6.21) becomes

$$\begin{aligned} \tilde{\mathbf{W}}_k &= \mathbf{W}_k - \mu(\tilde{\mathbf{y}}_k^{CP} - \tilde{\mathbf{y}}_k)(\tilde{\mathbf{y}}_k^{CP} - \tilde{\mathbf{y}}_k)^H \mathbf{W}_k \\ \mathbf{W}_{k+1} &= Toeplitz\{\tilde{\mathbf{W}}_k\} \end{aligned}, \tag{6.24}$$

which will be called the *relative gradient shortening* (RGS) algorithm. In the RGS adaptation, the relative change for matrix $\mathbf{W}_k$ is the matrix $(\tilde{\mathbf{y}}_k^{CP} - \tilde{\mathbf{y}}_k)(\tilde{\mathbf{y}}_k^{CP} - \tilde{\mathbf{y}}_k)^H$. As has been shown, at the steady state (6.18) is expected to hold. In the adaptation (6.24), the terms in (6.18) are used explicitly to update matrix **W**.

From adaptation (6.24) we can see that when $P = 1$, $(\tilde{\mathbf{y}}_k^{CP} - \tilde{\mathbf{y}}_k)(\tilde{\mathbf{y}}_k^{CP} - \tilde{\mathbf{y}}_k)^H$ is a scalar, so that the elements in $\mathbf{W}_k$ are updated proportionally at each iteration, and this will keep the shortener **w** in a certain sub-space. As a result, to use relative gradient effectively, we need to have $P > 1$.

6.4 Simulations and Discussion

In this section, we will show some preliminary results for the proposed RGS channel shortening algorithm and compare them with those of the Merry and Frodo algorithms.

The experiments were done for the DSL communication channels given in [17]. We will only give the results for one of the channels, since the results are consistent with different channel examples.

In the example, the length of the channel is 512. The size of the OFDM block without the CP is 512. In other words, the size of FFT matrix used to generate the OFDM symbols

is $N \times N = 512 \times 512$. The length of the CP is $\nu = 32$. The length of the shortener is set to be 16. The SNR of the channel is 40dB. For the proposed RGS algorithm and the Frodo algorithm, we use $P = 10$ pairs of the CP and the OFDM symbols for the cost function based on redundancy in (6.8), i.e. $S_f = \{\nu - 9, ..., \nu\}$. The delay parameter is obtained with the method in [10], with $\Delta = 21$ in the example.

During iterations, we use two criteria to measure the performance of the algorithms. One is the inter-symbol interference (ISI) of the shortened channel, defined as

$$\text{ISI} = \sum_{i=0}^{M+L} \frac{|c_i|^2}{\max_i |c_i|^2} - 1 \tag{6.25}$$

for shortened system $\mathbf{c} = \mathbf{h} * \mathbf{w}$. Another criterion used is the ratio of the power inside the shortened window to that outside the window [2], [4], [11]. The window length is the size of the CP $\nu = 32$. With this criterion, we are in effect trying to concentrate the shortened system taps with significant magnitudes within a range whose length is bounded by that of the CP. The higher the power ratio, perhaps we can expect the algorithm to be more effective in channel shortening.

The curves for the ISI and the power ratio of the Merry algorithm, the proposed RGS algorithm and the Frodo algorithm are shown in Fig. 6.3 and Fig. 6.4.

From Fig. 6.3, we can see that when the ISI is considered, our proposed RG shortening algorithm (RGS) has the lowest (best) ISI after convergence. The convergence speed of the RGS algorithm is almost the same as that of the Frodo algorithm, which is faster than that of the Merry algorithm. The convergence speed of the Merry algorithm is comparatively slow since only on CP symbol is used for redundancy at each iteration.

In Fig. 6.4, the power ratio is also compared. From the figure we see that the measure of the power ratio is not consistent with that of the ISI. After the ISI converges to a certain value, the power ratio may still change. In terms of power ratio, the Merry algorithm has the highest value after convergence. From this point of view, the Merry algorithm performs well in concentrating power. However, if we consider the whole OFDM system, we also need to check the effect of the channel shortening, i.e. whether the equalization can be solved in an efficient way after shortening.

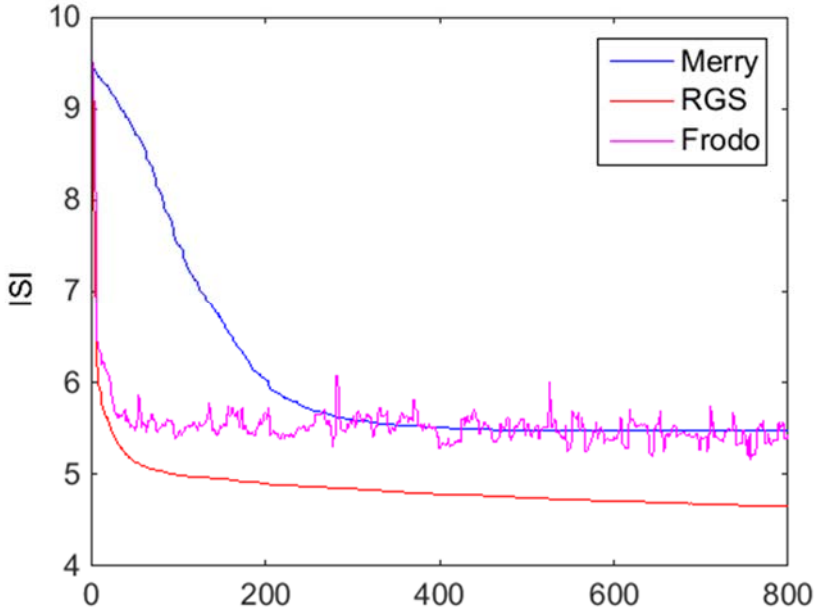

Fig. 6.3 ISI of the shortened channel.

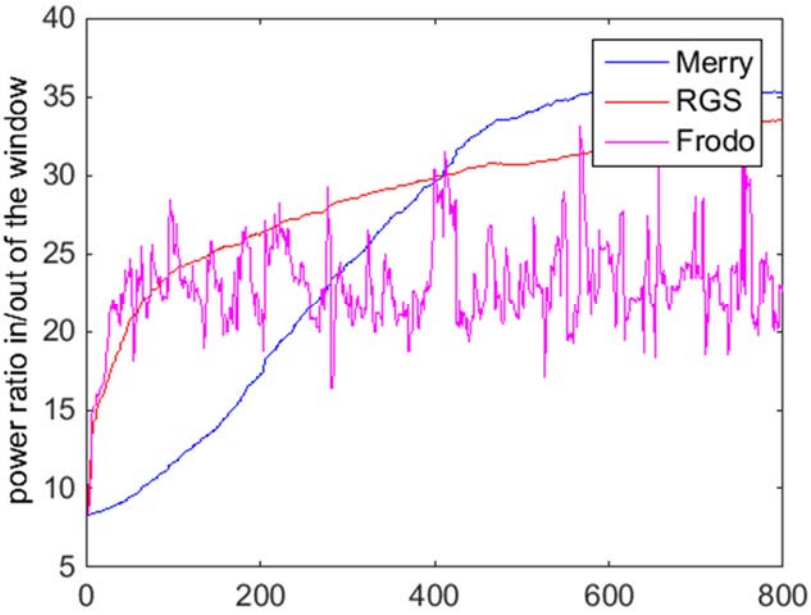

Fig. 6.4 Power ratio of taps inside to those outside the window

In Fig. 6.5, the actual impulse responses after channel shortening using the different algorithms is compared. The first 200 taps of the cascaded system **c** is shown, since there is a long tail of zeros in **c**. From the figure it can be seen with all three algorithms shortening can be achieved. The Frodo algorithm appears qualitatively to generate a shorter channel than the other algorithms. The RGS result is qualitatively also quite comparable. From the figure, it can be seen that with the Merry algorithm, the shortened channel has many nonzero taps with large magnitude in addition to the one with the largest magnitude, and this explains the high power ratio reflected in Fig. 6.4 These results indicate that a single performance criterion such as ISI or power ratio is not necessarily the best way to judge performance.

We found that the performance of the RGS and the Frodo algorithms depends on how many pairs of CP and OFDM symbols are used for redundancy. When P is selected to be small, the shortening may not be very good, and there will remain tails outside the window.

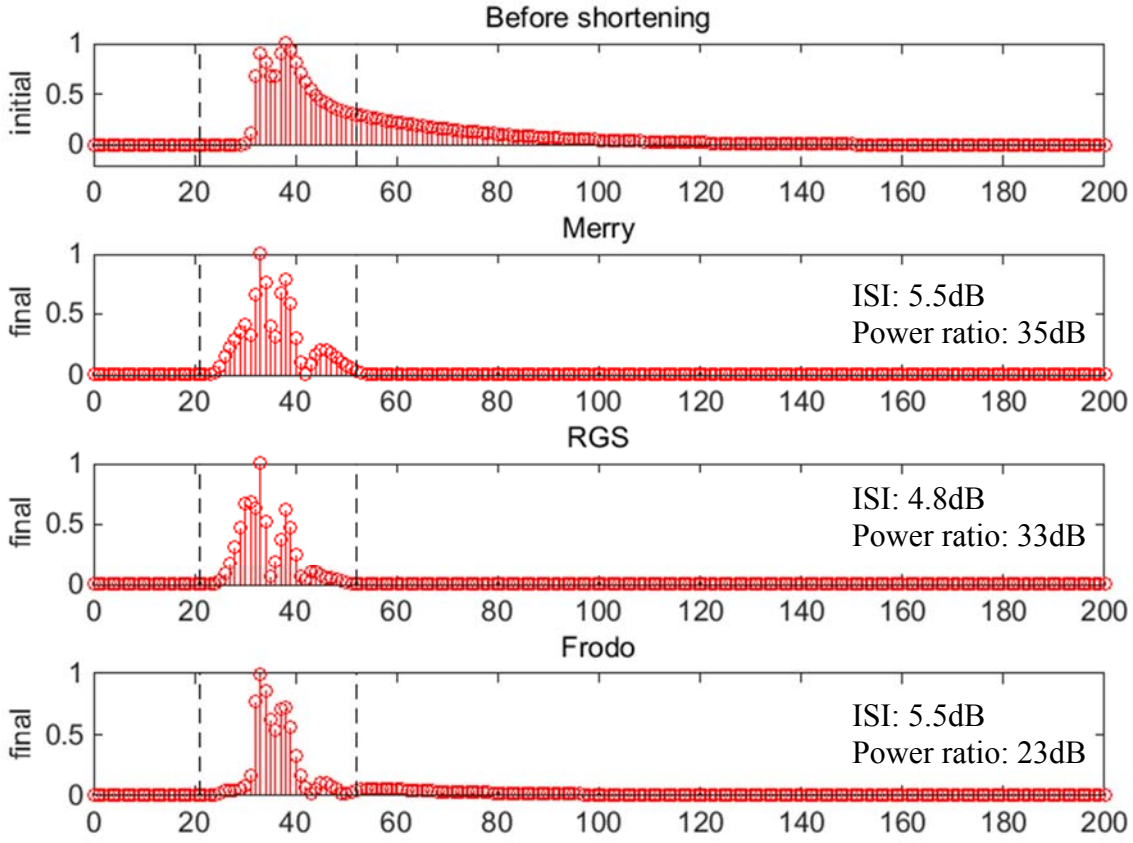

Fig. 6.5 Impulse response of shortened channel, $P = 10$.

We also did experiments for $P = \nu$ for the Frodo and our proposed RGS algorithms. The impulse responses of the shortened channels are shown in Fig. 6.6. From the figure, it can be seen that when the whole set of the CP is used for redundancy, i.e. $S_f = \{1, 2, ..., \nu\}$, qualitatively the RGS gives better performance. With the RGS algorithm the shortened channel has one dominant tap, with many of the other taps much smaller than the dominant

tap. The taps outside the window have very slightly heavier tails compared to the case when $P = 10$. As a result, equalization may be realized to a good extent, with the possibility that ISI can be further reduced with a better selection of the parameters such as the length of the shortener.

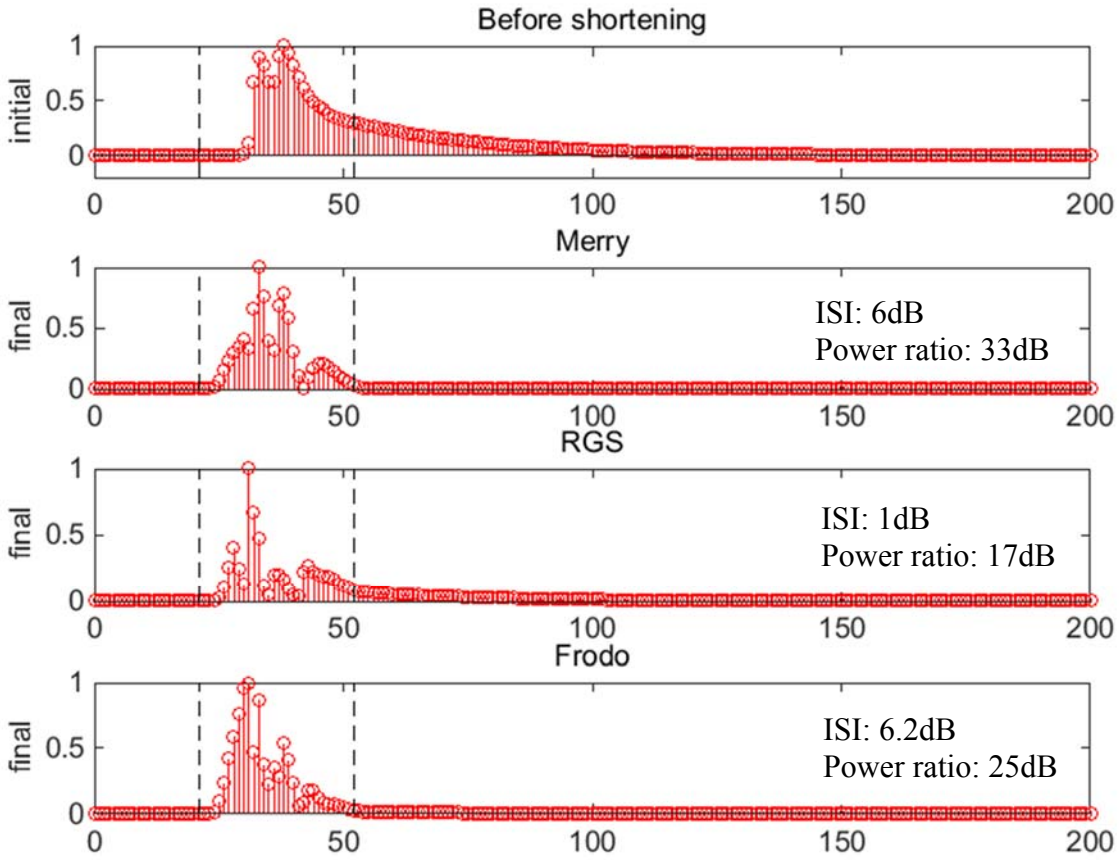

Fig. 6.6 Impulse response of shortened channel, $P = \nu$.

To make a fair comparison of the algorithms, we need to consider multiple factors that affect the performance. The purpose of channel shortening is to remove the inter-carrier interference, and to make the equalization part easier for suppressing ISI. We have seen that qualitatively the shortened channel impulse response using RGS compares very

favorably against the Merry algorithm result and is also good compared to the Frodo algorithm outcome. The ISI measure is somewhat more consistent with this observation compared to the power ratio, which has been used in previous studies [2], [4], [11]. Better criteria need to be established that not only measure the performance of the shortening part, but also the degree to which the difficulty of final equalization is reduced.

6.5 Conclusion

In this chapter, we discussed the application of the relative gradient in channel shortening for OFDM systems. The redundancy of the cyclic prefixes was used to define the cost function. By formulating the problem in a matrix form, we showed how relative gradient can be used for the adaptation of a Toeplitz matrix containing the shortener vector. Simulation results showed that the proposed RG based algorithm performs better than an existing algorithm under the ISI criterion, but appears less effective in terms of power concentration. In future work, better criteria need to be found for a fair and comprehensive evaluation of the algorithms.

References

[1] R. K. Martin, M. Ding, B. L. Evans, and C. R. Johnson, "Efficient channel shortening equalizer design," in *Conference Information Science and Systems*, 2003.

[2] R. K. Martin, J. M. Walsh, and C. R. Johnson, "Low-complexity MIMO blind, adaptive channel shortening," *IEEE Trans. Signal Process.*, vol. 53, no. 4, pp. 1324–1334, 2005.

[3] B. Muquet, Z. Wang, G. B. Giannakis, M. De Courville, and P. Duhamel, "Cyclic prefixing or zero padding for wireless multicarrier transmissions?," *Commun. IEEE Trans.*, vol. 50, no. 12, pp. 2136–2148, 2002.

[4] C. Johnson Jr, R. Martin, J. Walsh, A. Klein, C. Orlicki, and T. Lin, "Blind Channel Shorteners," in *Proc. 13th IFAC Symposium on System Identification, Rotterdam, The Netherlands*, 2003, vol. 1.

[5] T. Abrudan, M. Sirbu, and V. Koivunen, "A Block-Toeplitz VCMA Equalizer for MIMO-OFDM Systems," in *37th Asilomar Conference on Signals, Systems and Computers.*, 2003, no. 1, pp. 1037–1041.

[6] T. Abrudan and V. Koivunen, "Blind equalization in spatial multiplexing MIMO-OFDM systems based on vector CMA and decorrelation criteria," *Wirel. Pers. Commun.*, vol. 43, no. 4, pp. 1151–1172, 2007.

[7] T. Abrudan, A. Hjorungnes, and V. Koivunen, "Toeplitz method for blind equalization in MIMO OFDM systems," in *International Zurich Seminar on Communications*, 2004, pp. 212–215.

[8] F. O. Alayyan, K. Abed-Meraim, and A. M. Zoubir, "Blind equalization in ofdm systems exploiting guard interval redundancy," in *Signals, Systems and Computers, 2005. Conference Record of the Thirty-Ninth Asilomar Conference on*, 2005, pp. 697–700.

[9] D. M. Hewavithana, Thushara C Brookes, "Blind adaptive channel equalization for OFDM using the cyclic prefix data," *Glob. Telecommun. Conf. 2004. GLOBECOM'04. IEEE*, vol. 4, pp. 2376–2380, 2004.

[10] P. J. Melsa, R. C. Younce, and C. E. Rohrs, "Impulse response shortening for discrete multitone transceivers," *IEEE Trans. Commun.*, vol. 44, no. 12, pp. 1662–1672, 1996.

[11] R. K. Martin, J. Balakrishnan, W. A. Sethares, and C. R. Johnson, "A blind adaptive TEQ for multicarrier systems," *IEEE Signal Process. Lett.*, vol. 9, no. 11, pp. 341–343, 2002.

[12] R. K. Martin, M. Ding, B. L. Evans, and C. R. Johnson, "Efficient channel shortening equalizer design," *EURASIP J. Appl. Signal Processing*, vol. 2003, no. 13, pp. 1279–1290, 2003.

[13] M. de Courville, P. Duhamel, P. Madec, and J. Palicot, "Blind equalization of OFDM systems based on the minimization of a quadratic criterion," in *Communications, 1996. ICC'96, Conference Record, Converging Technologies for*

Tomorrow's Applications. 1996 IEEE International Conference on, 1996, pp. 1318–1322.

[14] D. L. Jones, "Property-restoral algorithms for blind equalization of OFDM," in *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference on*, 2003, pp. 619–622.

[15] J. Chatterjee, Chanchal Roychowdhury, Vwani P Ramos and M. D. Zoltowski, "Self-organizing algorithms for generalized eigen-decomposition," *Neural Networks, IEEE Trans.*, vol. 8, no. 6, pp. 1518–1530, 1997.

[16] Z. Ding and Y. Li, *Blind Equalization and Identification*. CRC press, 2001.

[17] R. K. Martin, "Matlab Code for Papers by R.K. Martin, http://bard.ece.cornell.edu/matlab/martin/index." .

Chapter 7

Conclusion

In this dissertation, we explored blind equalization (BE) algorithms based on the concept of relative gradient, using constrained adaptation for the equalizer parameters. We studied two types of BE algorithms: one based on the independence of source symbols, and the other exploiting signaling constellation structure. Relative gradient was used in algorithms for adaptation of the equalizer matrix containing the response of the equalizer. The Toeplitz structure constraint on the equalizer matrix was incorporated into the iterations for faster convergence. In addition, we improved the algorithms by simplifying the constrained adaptations to equivalent computationally efficient equalizer vector adaptation. Furthermore, with approximation schemes for the terms used in the adaptation and efficient implementation of the iterations, the computational cost was further reduced. Channel shortening for OFDM systems with relative gradient adaptation was also investigated in preliminary work.

In Chapter 2, we reviewed basic ideas of blind source separation (BSS) and blind equalization. We gave examples of independent component analysis (ICA) based contrast functions and online adaptive algorithms for BSS. Two key ideas of ICA, whitening and orthogonality, were discussed. We explained the connection of BE to BSS by expressing the convolution-based equalization model as a matrix model. Finally we reviewed existing algorithms for BE including the widely used Bussgang-type algorithms and ICA-based algorithms.

In Chapter 3, we started with BE for single carrier systems with block transmission schemes to allow BE to be achieved using BSS. Zeros or cyclic prefixes are padded between the transmitted blocks of source symbols as guard intervals. With the guard intervals, BE could be modeled as a standard BSS problem, with the separating matrix satisfying a Toeplitz or circulant structure constraint. For existing ICA-based algorithms for BSS that use the relative gradient, we proposed to include the structure constraint during adaptation for the "separating" matrix. The elements in the separating matrix were analyzed. With the Toeplitz or circulant structure, the matrix adaptation was simplified as an equivalent vector adaptation, which helps reduce computational cost. We also examined the channel characteristics that fit the use of the constrained ICA algorithms. In addition, for sources with independent in-phase and quadrature parts, the I/Q independence constraint was applied for phase recovery.

In Chapter 4, we extended the previous results to continuous transmission symbol-rate single-carrier schemes with no zero padding or cyclic prefix. Although in this case the model does not satisfy the requirement of a standard BSS problem, we showed how BE can still be realized with constrained ICA-based algorithms. The equalizer impulse

response is contained in a Toeplitz matrix, which is used to separate the source symbols. Similar to the block transmission scheme, the matrix adaptation was simplified to become an equalizer vector adaptation. The computational cost of the algorithm can be reduced by implementing the algorithm with fast Fourier transform. In addition, we proposed to use approximation schemes to approximate the nonlinear cross-correlation terms used in the adaptation, which can reduce the computational complexity further. This idea of constrained adaptation can be applied generally with different adaptive ICA based algorithms. Similar to the block transmission scheme, the I/Q independence constraint can also be applied for continuous transmission case.

In Chapter 5, we proposed an approach to process a block of equalizer outputs at each iteration rather than only the most recent output as in standard Bussgang algorithms. The Bussgang-type algorithms were further modified to make use of relative gradient (RG), which yields an effective BE scheme. With a block of equalizer outputs, the Toeplitz matrix constraint is enforced for faster convergence of equalizer coefficients. In the block RG version of the Bussgang type algorithms, the Bussgang condition at the steady state appears in the adaptation more explicitly, which yields faster convergence than the corresponding standard Bussgang algorithms. The block RG Bussgang-type algorithms work well for correlated sources as well. In this chapter, the relation between the block RG Bussgang-type algorithms and the ICA-based algorithms was also discussed. Although the block RG Bussgang-type algorithms and the ICA-based algorithms start from different criteria, constellation signaling property and independence, respectively, they end up with similar forms for adaptation.

In Chapter 6, we explored the application of the relative gradient in channel shortening for orthogonal frequency division multiplexing (OFDM) systems. The OFDM scheme with cyclic prefix was studied. This redundancy was used to define a cost function to minimize the difference between the OFDM data symbols and the redundant cyclic prefix symbols. Simulation results showed that the proposed RG based channel shortening algorithm performs better than an existing method, in terms of faster convergence when the inter-symbol interference from the shortened channel is measured. When the power concentration of the shortened channel was considered the algorithm with the relative gradient was slightly less effective. However, the goal of channel shortening is equalization and the ISI criterion may be more relevant. A comprehensive analysis of performance and other alternatives remains to be explored.