

THESE

présentée pour obtenir le titre de :

**Docteur de l'Ecole Nationale Supérieure des
Télécommunications**

Spécialité : Signal et Images

Constantinos PAPADIAS

**METHODES D'EGALISATION ET D'IDENTIFICATION
AVEUGLE POUR LES CANAUX LINEAIRES**

Soutenu le 16 Mars 1995 devant le jury composé de :

Claude GUEGUEN	Président
Anthony CONSTANTINIDES Joel LE ROUX	Rapporteurs
Pierre DUHAMEL Sylvie MARCOS Dirk SLOCK	Examineurs

PhD THESIS

presented in order to obtain the degree of :

**Docteur de l'Ecole Nationale Supérieure des
Télécommunications**

Speciality : Signals and Images

Constantinos PAPADIAS

**METHODS FOR BLIND EQUALIZATION AND
IDENTIFICATION OF LINEAR CHANNELS**

Defended on March 16, 1995, before the committee composed by :

Claude GUEGUEN	Chairman
Anthony CONSTANTINIDES Joel LE ROUX	Referees
Pierre DUHAMEL Sylvie MARCOS Dirk SLOCK	Examiners

Contents

Acknowledgements	9
Résumé	10
Summary	11
1 Introduction	12
1.1 An overview of the thesis	12
1.2 Contributions	15
1.3 Abbreviations	17
1.4 Mathematical Notation	18
PART I	20
2 Equalization and identification of linear channels	21
2.1 Introduction	21
2.2 Digital modulation	22
2.3 Impairments of communication channels	24
2.4 Symbol rate, Sampling, Inter-Symbol-Interference, Channel eye	26
2.5 ISI cancelling, equalization	30
2.6 Adaptive filtering and equalization	35
2.7 Training sequence, Decision Directed (DD) equalization	39
2.8 Blind equalization	40
2.9 Bussgang techniques for blind equalization	44
2.10 Principal aims of the thesis	52
3 A new class of algorithms for constant-modulus blind equalization	54
3.1 Introduction	54
3.2 The new class of algorithms	57
3.2.1 Derivation	57
3.2.2 Discussion	60
3.2.3 Criterion interpretation	61
3.2.4 Projection interpretation	61
3.2.5 Relation to other algorithms	63
3.3 The separation principle	64
3.4 Computational organization	66
3.4.1 The FAP algorithm	67
3.5 Performance analysis	71
3.5.1 Stability	72
3.5.2 Convergence behaviour	72
3.6 Computer simulation results	77

3.6.1 Opening the channel eye	78
3.6.2 Convergence speed	79
3.6.3 Avoiding local minima	82
3.7 Conclusions	83
3.8 Appendix 3.A	85
3.9 Appendix 3.B	85
3.10 Appendix 3.C	86
4 A modified APA algorithm for adaptive filtering and blind equalization	88
4.1 Introduction	88
4.2 A class of regularized APA-type adaptive filtering algorithms	89
4.2.1 Derivation	91
4.2.2 Asymptotic analysis	92
4.2.3 Further discussion	94
4.3 Algorithmic organization	95
4.3.1 An exact algorithmic organization	95
4.3.2 Approximative algorithmic implementations	98
4.4 NSWERCMA: a counterpart of the proposed class of algorithms for BE	98
4.5 Computer Simulations	100
4.5.1 Opening the channel eye	100
4.5.2 Convergence speed	101
4.5.3 Local minima	102
4.6 Conclusions	103
5 Decision-directed blind equalization	104
5.1 Introduction	104
5.2 Motivation	106
5.3 Analysis of the stationary points of the DDA cost function	108
5.3.1 The CM case	108
5.3.2 The non-CM case	118
5.4 Other classes of DD algorithms	120
5.5 A CMA-DD hybrid corresponding to a novel Generalized-Sato scheme	122
5.6 Computer simulations	124
5.7 Conclusions	126
5.8 Appendix 4A: On the uniqueness of the stationary points (5.13).	127
5.9 Appendix 4B: On the shape of regions separating local minima	127
6 A bilinear setup for globally convergent blind equalization	129
6.1 Introduction	129
6.2 A new blind equalization principle	131
6.2.1 The idea	131
6.2.2 The formulation	132
6.2.3 The main result	134
6.2.4 Discussion	136
6.2.5 Implementation	137
6.3 The influence of additive noise	140
6.4 Special case: the crucial role of over-parameterization	142
6.4.1 The singularity related to the over-parameterized case	142

6.4.2	A remedy: using a subspace fitting approach for the calculation of a ZF equalizer	143
6.4.3	Discussion	145
6.5	Computer simulations	146
6.5.1	The correctly parameterized case	146
6.5.2	Noiseless over-parameterized case: subspace fitting technique	147
6.5.3	FIR noisy channels	147
6.6	Conclusions and further discussion	149
PART II		150
7	New blind techniques for multichannel identification/ equalization	151
7.1	Introduction	151
7.2	Fractional spacing and stationarity	153
7.2.1	Stationarity and cyclostationarity	153
7.2.2	Vector stationarity	157
7.3	Channel identifiability from output measurements	158
7.3.1	Phase ambiguity and determinacy with SOS	158
7.3.2	Identification by vector - spectral factorization	159
7.4	Zero Forcing equalization of the SIMO channel	161
7.4.1	Fractionally-spaced equalization	161
7.4.2	FIR Zero-Forcing (ZF) Equalization	163
7.4.3	An equivalent SISO setup using upsampling and downsampling	166
7.4.4	ZF equalization and noise enhancement	175
7.5	ZFE and ID by multichannel linear prediction	178
7.5.1	Zero-forcing equalization by multichannel linear prediction	178
7.5.2	Linear prediction in the frequency-domain	183
7.6	MMSE equalization	185
7.6.1	Derivation of the MMSE equalizer	186
7.6.2	MMSE equalizers and ZF equalizers	187
7.6.3	MMSE equalization and linear prediction	190
7.6.4	The influence of the lag	191
7.6.5	Further discussion	193
7.7	Signal and Noise subspaces	193
7.7.1	Parameterization of the signal and noise subspaces	193
7.7.2	Sinusoids in noise: comparison	194
7.8	Channel Estimation by Subspace Fitting	194
7.9	Channel Estimation from Data using Conditional ML	196
7.9.1	Deterministic Maximum Likelihood	196
7.9.2	Conditional ML assuming a Gaussian prior for the symbols	204
7.10	Further discussion	206
7.11	Appendix 7A: Sylvester matrices	207
8	Applying existing blind techniques to the multichannel setup	209
8.1	Introduction	209
8.2	Formulation	210
8.2.1	CMA - like and DD algorithms	210
8.2.2	The bilinear case	212

8.3	Simulations	213
8.3.1	NSWCMA	213
8.3.2	NSWERCMA	214
8.3.3	Bilinear method	215
8.4	Conclusions and further discussion	216

APPENDICES		217
A	Sommaire détaillé en français	218
A.1	L' état de l'art en égalisation aveugle	218
A.1.1	Le principe d'égalisation linéaire	218
A.1.2	Egalisation aveugle	220
A.1.3	Algorithmes de type "Busgang" pour l'égalisation aveugle	222
A.1.4	Objectifs de la thèse	225
A.2	Egalisation et identification aveugle non sur-échantillonné	226
A.2.1	L'algorithme NSWCMA	226
A.2.2	Un principe de séparation	228
A.2.3	Une classe d'algorithmes regularisés	230
A.2.4	Egalisation Dirigé par les Décisions (DD)	233
A.2.5	Un principe bilinéaire pour l'égalisation aveugle	236
A.3	Egalisation et identification aveugle multicanal	239
A.3.1	Egalisation ZF (Zero-Forcing)	240
A.3.2	Egalisation aveugle multicanal par prédiction linéaire	244
A.3.3	Egalisation MMSE	245
A.3.4	Méthode sous-espace	246
A.3.5	Méthodes de Maximum de Vraisemblance Conditionnelle	247
A.3.6	Application des méthodes existantes à la structure multicanal	248

List of Figures

2.1	A generic communication system	22
2.2	Some typical PAM and QAM constellations	24
2.3	Linear noisy channel	25
2.4	A 2-PAM transmitted continuous-time signal	27
2.5	Received waveforms of a 2-PAM signal transmitted through a linear channel	28
2.6	A typical channel eye of a 2-PAM system	30
2.7	A linear equalization structure	32
2.8	A typical system identification setup	35
2.9	A typical adaptive equalization setup	38
2.10	The structure of a GSM time-slot	39
2.11	A Bussgang blind equalization setup	45
2.12	A Godard cost function for an AP(1) channel and a FIR(1) equalizer	51
3.1	A typical blind equalization scheme	55
3.2	Choice of the optimal projection onto the unit circle	58
3.3	A geometrical interpretation of the minimized criterion	62
3.4	A projection interpretation of the APA algorithm	63
3.5	CMA and NSWCMA (L=2) cost functions in a case of two equalizer parameters (N=2)	73
3.6	The opening of the eye achieved by the NSWCMA algorithm	79
3.7	Noiseless case : two examples	80
3.8	Noisy FIR channel: comparative simulations	82
3.9	A comparison of CMA and NSWCMA (L=2) for an AP(1) channel	83
3.10	Ill convergence: a more realistic example	84
4.1	The bad effect of a long sliding window in a blind equalization experiment	90
4.2	A comparison of the conditioning of a rectangular and an exponential covariance matrix	91
4.3	The sensitivity of the convergence speed w.r.t. to λ	95
4.4	The improvement in performance using an exponential sliding window	100
4.5	The opening of the channel-eye achieved by a member of the proposed class of algorithms	101
4.6	Comparative study of several BE algorithms in terms of convergence speed	102
4.7	Escaping from a local minimum of the cost function	103
5.1	Three standard equalization modes	105
5.2	The similarity between CMA 1-2 and DDA	107
5.3	The definition of the $bis(\cdot)$ function.	118

5.4	4-PAM: The stationary points on the line $s_1 = s_2$ for DDA and Sato, respectively	120
5.5	4-PAM: the DDA and Sato cost functions for $M = 2$	121
5.6	A CMA-DD hybrid principle: 16-QAM	123
5.7	The opening of the eye achieved by a DD algorithm	124
5.8	The opening of the eye achieved by a the CMA-DD hybrid in a non-CM case	125
6.1	The principle of a typical CM blind equalizer	131
6.2	A proposed principle for CM blind equalization: an example	132
6.3	A representation of the bilinear method concept	137
6.4	A multichannel implementation of the proposed blind equalization setup	143
6.5	Comparative simulations for the noiseless case	146
6.6	A simulation for the over-parameterized case	147
6.7	Comparative simulations for the noisy FIR case	148
6.8	A simulation of the bilinear rank-1 algorithm	149
7.1	$T/2$ fractional spacing	152
7.2	Another SIMO setup: multiple antennas	153
7.3	Time-frequency representation of the cyclostationary and stationary autocorrelation functions	157
7.4	The effect of sampling and oversampling on the stationarity of $x(t)$	158
7.5	Polyphase representation of the T/m fractionally-spaced channel and equalizer for $m = 2$.	163
7.6	The transmitting part of the SIMO setup and an equivalent representation	167
7.7	The receiving part of the SIMO setup and an equivalent representation	168
7.8	Fractionally-spaced channel and equalizer.	169
7.9	A verification of the equivalence of the setups of figures 7.6(a) and 7.6(b)	172
7.10	The overall system $G(z)$ corresponding to ZF equalization	172
7.11	Nyquist condition for the oversampled channel	173
7.12	Finding the optimal ZFE by orthogonality	177
7.13	The prediction filter	184
7.14	A verification of eq. (7.147)	189
7.15	MMSE, LP and ZF equalization	191
7.16	MMSEE performance: the influence of the lag	192
7.17	Channel id in the case of a noisy but perfectly estimated cov. matrix	195
8.1	The zeros of the two subchannels	213
8.2	Comparative simulations of the NSWCMA and the CMA for a $T/2$ fractionally-spaced channel	214
8.3	Comparative simulations of the NSWCMA, the NSWCMA and the CMA for a $T/2$ fractionally-spaced channel	215
8.4	Simulating the bilinear method for the $T/2$ fractionally spaced setup	215
A.1	Un schéma classique d'égalisation aveugle	218
A.2	Le principe d'égalisation de Bussgang	223
A.3	Comparaison de la vitesse de convergence du NSWCMA et du CMA pour un canal bruité linéaire	228
A.4	Comparaison des performances respectives du NSWCMA et du CMA en terme de "ill-convergence"	229

A.5	Le mauvais effet d'une fenêtre glissante longue	231
A.6	L'amélioration en performance due à l'utilisation d'une fenêtre exponentielle	232
A.7	L'amélioration en vitesse de convergence atteinte par le NSWERCMA	233
A.8	La similarité entre le CMA 1-2 et le DDA	234
A.9	L'ouverture de l'oeil effectué par le DDA	235
A.10	Un principe hybride CMA-DD: 16-QAM	236
A.11	L'amélioration de l'algorithme hybride par rapport au DDA	237
A.12	Le principe de l'égalisation adaptative bilinéaire	239
A.13	Simulations comparatives pour un cas idéal	240
A.14	Simulations comparatives pour le cas d'un canal RIF bruité	241
A.15	Un canal SIMO	242
A.16	L'effet du suréchantillonnage de $x(t)$ à sa stationnarité	242
A.17	Principe de l'égalisation multicanal pour le cas $m = 2$	243
A.18	Représentation équivalente de la structure d'égalisation multicanal	244
A.19	Une vérification de (A.78)	246
A.20	Simulations comparatives pour un canal sur-échantillonné avec un facteur 2	249

List of Tables

3.1	CM-type BE algorithms derived by applying the separation principle	65
3.2	The FAP algorithm	71
4.1	An exact algorithmic organization for the exponentially regularized FAP	97
4.2	An algorithmic organization for an approximative regularized FAP	99
5.1	The evolution of λ_2 for $M = \text{odd}$	117
5.2	DD-type BE algorithms	122
A.1	Certains algorithmes du type "à Module Constant" et leur correspondants pour le filtrage adaptatif classique	230

Acknowledgements

First of all I would like to thank Professor G. Carayannis of N.T.U.A. for having motivated me to pursue a PhD program in France and for having helped me in implementing this idea.

I wish to express my gratitude to Professor C. Gueguen for having accepted me as a PhD student at the E.N.S.T., for his contribution in determining the subject of my thesis, for his encouragement as well as for accepting to be the president of the jury.

The lion's share belongs undoubtedly to my thesis advisor Professor D. Slock. His willing to share with me his enthusiasm for research, his unlimited availability and his substantial help in my work are deeply appreciated.

I wish to thank especially Professor A. Constantinides of Imperial College for having accepted to be an examiner of my thesis despite his heavy schedule, for his detailed reading of the manuscript and his positive feedback, as well as for his general interest in my work and the suggestions he has given me during our exciting discussions.

I also wish to thank Professor J. LeRoux of Université de Nice for accepting to be my examiner and for having given me valuable feedback on my work.

I equally thank Professor P. Duhamel of E.N.S.T. and Mme S. Marcos of the Laboratoire des Signaux et Systèmes at Plateau de Moulon for their participation in the jury.

I also thank Professor P. Humblet for his help as the head of Eurécom's Mobile Communications department, as well as for the occasional collaboration we have had during the last two years.

Special thanks go to J. L. Dugelay and E. De Carvalho for checking the French summary and to D. Dembele for having read the whole text and for providing useful feedback.

I cannot help thinking at this point of all the people at Eurécom that have surrounded me all this time. My office mates (R. Knopp, K. Maouche, D. Samfat, M. Troulis), my colleagues (C. Blum, A. Dakroub, P. Gelin to mention but a few ...), all the scientific staff, the students (especially A. Samimi and Y. Dahmane for our collaboration during their project), and the administrative personnel : I wish to thank them all both for their professional help and the friendly international atmosphere they have helped create at Eurécom.

This paragraph would of course be incomplete if I didn't mention Tania, whose presence has helped in so many ways during the last years.

I finally want to express my deep gratitude to my parents for their constant and unconditional love and encouragement and for all they have sacrificed for the success of their children. It is to them that I dedicate this piece of work.

Résumé

Nous étudions le problème de l'égalisation et de l'identification aveugle des canaux de communication linéaires : la séquence émise (ou la réponse impulsionnelle du canal) doit être identifiée uniquement à partir de la sortie du canal.

Dans la première partie de la thèse, on considère l'égalisation non-sur-échantillonnée. Dans ce cas l'égalisation aveugle des canaux à phase non-minimale n'est pas possible en utilisant uniquement les statistiques du second ordre de la sortie du canal. Les algorithmes de Bussgang constituent une classe importante d'algorithmes qui utilisent implicitement les moments d'ordre supérieur de la sortie du canal pour l'égalisation aveugle. Quelques désavantages de ces algorithmes sont leur vitesse de convergence lente ainsi que le problème dit du "ill-convergence" : dépendants de leur initialisation, ils peuvent converger à des minima locaux qui n'ouvrent pas l'oeil du canal.

Nous proposons une nouvelle classe d'algorithmes du type "A Module-Constant (CM)" qui accélèrent la vitesse de convergence et aident à éviter le problème des minima locaux. Cette classe d'algorithmes (appelée NSWCMA) est une contre-partie de l'algorithme de filtrage adaptatif classique APA (Affine Projection Algorithm). La dérivation du NSWCMA conduit aussi à la formulation d'un principe de séparation qui lie le filtrage adaptatif classique à l'égalisation aveugle. Ensuite, nous proposons une classe d'algorithmes régularisés pour résoudre le problème d'amplification de bruit souvent présent dans l'algorithme APA. Une contre-partie pour l'égalisation aveugle est aussi dérivée et nommée NSWERCMA. Le problème de l'égalisation Dirigée par les Décisions (DD) est considéré après : nous montrons que, contrairement à ce qui semble être pensé, les algorithmes DD sont capables d'ouvrir un oeil de canal initialement fermé si le signal d'entrée est CM. Pour le cas des constellations non-CM, on propose un schéma CM-DD hybride qui correspond à un nouveau algorithme de type Sato-Généralisé.

Afin d'obtenir une solution analytique qui résout le problème des minima locaux, nous proposons un principe d'égalisation bilinéaire : nous utilisons une nouvelle paramétrisation du problème qui conduit à une fonction de coût de type CM convexe. De cette façon on résout le problème des minima locaux. Plusieurs méthodes pour obtenir l'égaliseur linéaire en présence de bruit et de sur-paramétrisation sont présentées.

Dans la deuxième partie nous considérons le problème de l'égalisation et de l'identification aveugle multicanal : le signal reçu peut être vu comme la sortie d'un canal SIMO (Single-Input-Multiple-Output). Cette structure peut être utilisée dans le cas d'un récepteur à plusieurs capteurs ou d'un égaliseur sur-échantillonné (au cas où le signal reçu serait cyclostationnaire). L'identification aveugle peut maintenant être atteinte à l'aide des statistiques du second ordre de la sortie vectorielle. Un autre avantage de la structure multicanal est qu'elle permet l'égalisation ZF (Zero-Forcing) en utilisant des égaliseurs de longueur finie. Nous montrons comment des égaliseurs ZF ainsi que MMSE (Minimum Mean Square Error) peuvent être obtenus de façon aveugle par la prédiction linéaire multivariable. Nous proposons aussi des méthodes du type sous-espace ainsi que de maximum de vraisemblance afin d'améliorer la qualité de l'estimation. Finalement, nous appliquons les algorithmes proposés dans la première partie de la thèse à la structure multicanal afin d'améliorer encore leur performance.

Summary

We consider the problem of blind equalization and identification of linear communication channels: the transmitted sequence (or the channel impulse response) has to be identified based only on the received channel output.

In the first part of the thesis we consider equalization at the symbol rate. In this case blind equalization of non-minimum-phase channels is not possible with the use of second-order statistics of the channel output. The *Bussgang* algorithms constitute an important class of algorithms that implicitly use high order moments of the channel output in order to achieve blind equalization. However, they have a number of drawbacks, e.g. they have low convergence speed and they exhibit the problem of ill-convergence: depending on their initialization they may end up at undesired solutions.

We propose a class of algorithms of the Constant Modulus (CM) type that accelerate the convergence speed and help avoiding the problem of ill-convergence. This class of algorithms (called NSWCMA) is a counterpart of the Affine Projection Algorithm (APA) for adaptive filtering. As a byproduct of the derivation of this class of algorithms we formulate a separation principle that allows us to link classical adaptive filtering to blind equalization. We then propose a regularized class of algorithms that overcome the problem of noise amplification present in some cases in the APA. Its blind equalization counterpart is also derived and given the name of NSWERCMA. The problem of Decision-Directed (DD) equalization is considered next: we show that, contrary to what seems to be believed to date, DD algorithms can open an initially closed channel eye if the input signal is CM. For the case of non-CM constellations, we propose a CM-DD hybrid that corresponds to a novel Generalized-Sato scheme.

In order to obtain an analytical solution that solves the problem of ill-convergence, we propose a bilinear blind equalization principle: using a new parameterization of the problem, we construct a convex cost function for CM equalization. Using this formulation, the problem of ill-convergence is avoided. Different methods to obtain the linear equalizer in the presence of noise and order mismatch are also presented.

In the second part of the thesis we consider the problem of blind multichannel equalization and identification: the received signal may be viewed as the output of a SIMO (Single-Input-Multiple-Output) channel. Such a setup may be applicable either in the case of reception through an antenna array or in the case of fractionally-spaced receivers (in which case the received signal is cyclostationary). Blind identification can be now achieved by using only the second order statistics of the vector output. Another important advantage of the multichannel setup is that FIR (Finite-Impulse-Response) ZF (Zero-Forcing) equalizers exist. We show how ZF as well as MMSE (Minimum Mean Square Error) equalizers may be obtained blindly by using multivariable linear prediction. Moreover, we provide subspace as well as ML (Maximum Likelihood) methods to improve upon the quality of estimation. Finally the methods developed in the first part of the thesis are applied to the multichannel setup in order to further improve their performance.

Chapter 1

Introduction

THIS thesis deals with the problem of blind channel identification and equalization in digital communication systems. In this introductory chapter we present an overview of its contents, as well as a list of the achieved contributions.

1.1 An overview of the thesis

- Part I

In the first part of the thesis we address the problem of single-channel blind identification and equalization: the transmitted continuous-time waveform is received by a single sensor and sampled at the symbol rate.

- Chapter 2

In this background chapter we give a brief overview of the domain of digital transmission: digital modulation, transmission through linear channels and reception are discussed. The problem of Inter-Symbol-Interference (ISI) is presented and the notion of equalization is defined. Then the problem of blind equalization is defined and a brief overview of existing adaptive blind equalization techniques (the emphasis being on techniques of the *Bussgang* type) is provided.

- Chapter 3

In this chapter we propose a new class of algorithms for blind equalization of the constant-modulus type. These algorithms are derived by a deterministic criterion that imposes at each iteration a number of constraints of the constant-modulus type to the equalizer. This criterion (and therefore the corresponding algorithms as well) is a counterpart of the Affine Projection Algorithm (APA) that has been recently proposed in

adaptive filtering as an intermediate algorithm that combines the features of the LMS and the RLS algorithms. An important feature of the algorithms of this class is their accelerated convergence speed with respect to the classical Constant Modulus Algorithm (CMA). Moreover, the algorithms of this class are shown to have a better behaviour in terms of the problem of ill-convergence often present in blind equalization techniques. Some efficient schemes for the improvement of the computational complexity of the algorithms are proposed, and their performance is studied both theoretically and by means of computer simulations which verify the theoretically expected behaviour.

As a side-effect of the derivation of the NSWCMA algorithms from a deterministic criterion we have been able to formulate a so-called *separation principle*. This principle allows for the acquisition of algorithms suitable for blind equalization of the constant modulus type by a simple modification of classical adaptive filtering algorithms. Apart from its practical importance, this principle gives also important intuition for the better understanding of the CMA and the decision-directed (DD) algorithm.

- Chapter 4

In this chapter we treat a problem that may appear in the APA algorithm. Namely, when the length of the rectangular window used in the APA is big (approaching the filter length), then the APA has the following erratic behaviour: it shows a significant increase in its error variance after steady-state has been achieved. This problem is due to the noise amplification caused by the algorithm. The same effect of course is reflected to the NSWCMA algorithm presented in the previous chapter. In this chapter, a remedy to the problem is proposed. Namely, we propose the use of an exponential instead of a rectangular sliding window for the windowing of the data. The advantage of using such a window is that the employed sample covariance matrix has a better conditioning compared to the case of a rectangular sliding window for window lengths close to the filter length. A deterministic criterion that corresponds exactly to the derived algorithm is given. Then the algorithm is modified so as to provide its blind equalization counterpart by applying our previously stated separation principle. The new class of algorithms shows a significant improvement in terms of noise amplification in the steady state, as well as in its initial convergence. This is true in both classical adaptive filtering and blind equalization contexts. We also propose an efficient computational organization for the proposed algorithm based on the FAP algorithm. Finally the improvement of the proposed class of algorithms is tested through computer simulations in the blind equalization context.

- Chapter 5

In this chapter we consider the problem of decision-directed blind equalization. Decision-directed equalizers are often used in the final stage of a typical blind equalization procedure in order to improve the steady-state performance. However, even though they

could be used as well for blind equalization, this does not happen in practice since they are considered to be unable of opening a closed channel eye. In this chapter we take a closer look to DD equalization by examining the special case of an input signal that has a Constant Modulus (CM). We perform a convergence analysis of the DD equalizer in this case which shows its ability to open the channel eye. Combined with the superior performance of DD equalizers in the steady state, this result motivates their use for blind equalization of constant modulus signals.

Based on the fact that the major feature that prevents DD equalizers from converging to an acceptable setting is the non-CM form of the transmitted constellation, we propose a new hybrid CM-DD scheme for blind equalization. This scheme constitutes actually an alternative to the Generalized Sato algorithm for complex signals. The success of this scheme is verified by the good performance of the corresponding algorithms when used with non-CM constellations.

- Chapter 6

In this chapter we take a closer look to the problem of ill convergence often present in blind equalizers. This problem is often observed in the algorithms of the CMA - type: depending on its initialization, such an algorithm may end up to an undesirable setting. This is primarily due to the non-convex shape of the cost function that the algorithm minimizes. We propose a new formulation of the CMA 2-2 cost function by making a *mapping* of the equalizer parameters to a new set of parameters. The proposed cost function is convex with respect to the introduced parameter set and therefore leads to a globally convergent blind equalization method. A particularity of this optimization problem is that it becomes singular when the equalizer is over-parameterized. The mechanism that produces this singularity is analyzed and a solution that overcomes this problem is given. Several methods for the implementation of the technique are also proposed.

- Part II

In this part of the thesis we consider the problem of blind multichannel identification and equalization.

- Chapter 7

The multichannel - multiequalizer setup of a communication system is presented as a general framework that may correspond to either fractionally-spaced receivers or receivers that use an array of sensors. The important issue of stationarity in this context is analyzed and the identifiability of the multichannel based on second-order statistics is analyzed. Zero Forcing (ZF) as well as Minimum-Mean-Square-Error (MMSE) equalization is considered. The channel identification based on second-order statistics is also analyzed in the time-domain by using linear prediction. The analogy of this problem

with the problem of identifying sinusoids in noise is presented, as well as subspace-fitting methods for channel identification. Finally, some Maximum Likelihood (ML) methods for blind channel identification based on the received data are presented and their performance is analyzed.

- Chapter 8

In this chapter we consider the application of the blind equalization methods presented in the first part of the thesis to the fractionally-spaced setup. The impacts of fractional spacing to the performance of the techniques are discussed and their claimed behaviour is discussed through computer simulations. These simulations show the improved performance of the techniques of the first part of the thesis to the fractionally-spaced setup.

- Appendix A

In this appendix we provide an extended summary of the work presented in this thesis in french.

1.2 Contributions

At this point we highlight a number of points which we consider to describe the major contributions of this work:

- The proposal of a new class of algorithms (NSWCMA) for blind equalization that have an accelerated convergence speed with respect to other Bussgang algorithms.
- A study of the impact of normalization to blind equalization algorithms leading to the conclusion that normalization can help avoid the problem of ill convergence.
- The establishment of a "separation principle" linking classical adaptive filtering to adaptive blind equalization. This principle allows to obtain several classes of algorithms for BE, by simply modifying the corresponding adaptive-filtering algorithms.
- The proposal of a remedy for the problem of high steady-state error observed in the APA algorithm when the size of its sliding window approaches the filter length. The solution proposed is based on the regularization of the sample covariance matrix used by the algorithm, by exponential weighting. This solution gives a better-performing alternative to both the APA and the NSWCMA algorithm. A fast algorithmic organization scheme is also provided.
- A novel study of the Decision-Directed (DD) cost function in the special case of a Constant-Modulus input signal. This study shows that under some asymptotic conditions the DD Algorithm (DDA) is globally convergent and provides important evidence

for the fact that the DDA should be able of opening a closed-channel eye under realistic conditions if the input signal is CM. This result throws new light to the understanding of the DDA and motivates its use for cases where the transmitted signal is CM.

- The proposal of a novel CMA-DD hybrid criterion for blind equalization. This is a new Generalized-Sato algorithm that is closer to Sato's philosophy for real signals.
- The proposal of a convex cost function for blind equalization of the CM type. The convexity of this (bilinear) cost function leads to globally convergent algorithms, thus circumventing the problem of ill-convergence.
- An analysis of the ZF multichannel equalization problem: this leads to an equivalent single-channel representation, to a Nyquist condition in the frequency domain and to the acquisition of the optimal ZF equalizer for a given length.
- A comparative study of MMSE and ZF equalizers in the multichannel case as well as two methods to obtain an MMSE equalizer from a given ZF equalizer. One method is based directly on second-order statistics, whereas the other one is based on multidimensional linear prediction.
- The proposal of a Conditional Maximum Likelihood method for blind multichannel identification that assumes a Gaussian distribution for the symbols. This method improves upon a previously proposed Deterministic Maximum Likelihood (DML) method. The improvement is shown by deriving and comparing the Cramer Rao bounds for the two cases.

1.3 Abbreviations

The following abbreviations will be used throughout this thesis:

AP:	All-Pole (filter)
APA:	Affine Projection Algorithm
AR:	Auto-Regressive (process)
ARMA:	Auto-Regressive Moving-Average (process)
BD:	Blind Deconvolution
BE:	Blind Equalization
CM:	Constant Modulus
CMA:	Constant Modulus Algorithm
CML:	Conditional Maximum Likelihood
CRB:	Cramer Rao Bound
CRIMNO:	Criterion with Memory Nonlinearity algorithm
DD:	Decision-Directed
DDA:	Decision-Directed Algorithm
DFE:	Decision Feedback Equalization
DFT:	Discrete Fourier Transform
DML:	Deterministic Maximum Likelihood
DOA:	Direction Of Arrival
FAP:	Fast Affine Projection (algorithm)
FIR:	Finite Impulse Response
FSE:	Fractionally Spaced Equalization
FTF:	Fast Transversal Filter
GSA:	Generalized Sato Algorithm
GSM:	Global System for Mobile communications
HOS:	Higher Order Statistics
IIR:	Infinite Impulse Response
ISI:	Inter-Symbol-Interference
LMS:	Least-Mean-Square (algorithm)
LS:	Least-Squares
LTI:	Linear Time Invariant (system)
MA:	Moving Average (process)
MIL:	Matrix Inversion Lemma
MIMO:	Multiple Input - Multiple Output (system)
MISO:	Multiple Input - Single Output (system)
ML:	Maximum Likelihood
MMSE:	Minimum Mean Square Error
MP:	Minimum Phase (system)

NCMA:	Normalized Constant Modulus Algorithm
NLMS:	Normalized Least-Mean-Square Algorithm
NSWCMA:	Normalized Sliding Window Constant Modulus Algorithm
NSWERCMA:	Normalized Sliding Window with Exponential Regularization CMA
NMP:	Non-Minimum Phase
OF:	Oversampling Factor
PSD:	Power Spectral Density
QAM:	Quadrature Amplitude Modulation
RLS:	Recursive Least Squares (algorithm)
SCD:	Spectral Correlation Density (function)
SFTF:	Stabilized Fast Transversal Filter
SGA:	Stop-and-Go Algorithm
SIMO:	Single Input - Multiple Output (system)
SISO:	Single Input - Single Output (system)
SNR:	Signal to Noise Ratio
SOS:	Second Order Statistics
SVD:	Singular Value Decomposition
SWC:	Sliding-Window Covariance
ZF:	Zero Forcing

i.i.d.	independent identically distributed (random variables)
pdf	probability density function
w.r.t.	with respect to
s.t.	subject to
iff	if and only if
i/o	input-output
id.	identification
eq.	equation

1.4 Mathematical Notation

The following mathematical notation will be used throughout the thesis:

$(\cdot)^T$:	Matrix or vector transpose
$(\cdot)^*$:	Complex conjugate of a scalar, vector or matrix
$(\cdot)^H$:	Hermitian (complex conjugate) transpose
$Re(\cdot)$:	Real part of a complex scalar
$Im(\cdot)$:	Imaginary part of a complex scalar
$ \cdot $:	Modulus of a scalar

$\ \cdot\ $:	2-norm of a vector
$\ \cdot\ _F$:	Frobenius norm of a matrix
$\text{tr}(\cdot)$:	Trace operator
$\text{rank}(\cdot)$:	Rank of a matrix
$\text{sign}(x)$:	$\frac{x}{ x }$
$E(\cdot)$:	Statistical expectation
$\mathcal{F}(\cdot)$:	Fourier transform
$\mathcal{Z}(\cdot)$:	Z- transform
\mathcal{R} :	The set of real numbers
\mathcal{C} :	The set of complex numbers
$\text{span}(X)$:	The space spanned by the columns of matrix X
$P_Y X$:	The projection of X on the column-space of Y
$P_Y^\perp X$:	The projection of X onto the orthogonal complement of the column space of Y
$[\cdot]$:	Integer part of a real scalar
$\delta(\cdot)$:	Dirac function
$\delta_{n,k}$:	Kronecker delta
$\mathbf{0}_{n \times m}$:	$n \times m$ zero matrix
$I_{m \times m}$:	$m \times m$ identity matrix
$\{1, 2, \dots\}$:	The set of positive integers
$\{x_k\}$:	A discrete stochastic process or a time series or a discrete set
$\nabla_\theta(J)$:	The gradient of function $J(\theta)$ with respect to θ
\otimes :	Kronecker product
$*$:	Convolution operator
$\langle \cdot \rangle$:	discrete-time averaging
$\langle \cdot, \cdot \rangle$:	inner product

PART I

BLIND CHANNEL IDENTIFICATION AND EQUALIZATION AT THE SYMBOL RATE

Chapter 2

Equalization and identification of linear channels

THIS chapter is an introduction to the context of the thesis. Some basic principles of digital modulation are reviewed and the problem of *equalization* is presented. A short review of the state of the art in the domain of *blind equalization* is presented in order to motivate the research results presented in the rest of the thesis.

2.1 Introduction

This thesis deals with the problem of blind equalization and channel identification in digital communication systems. The transmission of information in digital form seems to be more and more the standard way of communicating nowadays, whether if this is done through a standard telephone line, a high speed data network or a mobile cellular network. Important contributions made in various other technological fields such as microelectronics, signal processing, information theory, optical electronics, computer science e.t.c. during the last years, have drastically accelerated the development of digital communications systems. In this work we will only be concerned with aspects related to the physical layer of a digital communication system, also called transmission layer. A block diagram of a generic communication system is given in figure 2.1: the original information is first formatted in digital form in order to be represented as a series of bits. These digits are then source-encoded in order to form another sequence of bits in as a compact way as possible. This binary sequence is then channel-coded, a procedure that introduces some amount of redundancy in the transmitted message in order to detect and possibly correct errors due to the transmission. The resulting coded sequence may also be merged with sequences originating from other sources that are going to be transmitted through the same transmission medium. This merging is done with

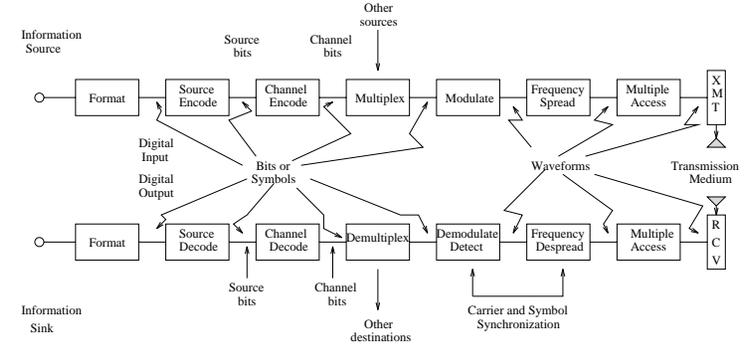


Figure 2.1: A generic communication system

the help of a multiplexer. The resulting sequence of bits is then transformed by the modulator into a continuous-time waveform proper for transmission over the physical transmission medium, also called *communication channel*. This waveform will typically occupy a certain frequency bandwidth, which might be further altered in a pre-arranged sophisticated way for reasons of robustness in the frequency spreading stage. We refer to the ensemble of processing stages considered up to this point as the “transmitter”. The resulting waveform is ready for transmission through the channel that is responsible for the delivery of the waveform to the other end. The received waveform, which according to the characteristics of the channel, may be more or less different from the transmitted one, is submitted to a series of processing stages that try to invert the operations done prior to transmission in order to finally obtain the transmitted information. It is therefore first submitted to a spectrum de-spreading stage, then to a demodulator and detector stage that tries to detect the transmitted bits, and then demultiplexed, channel decoded and source decoded in order to give the final digital output, which will be further transformed to analog form if the original information was also analog. We refer to the ensemble of the processing stages for the received signal as the “receiver”. In this work we will be primarily interested in the demodulation and detection stage of a digital communication receiver.

2.2 Digital modulation

The continuous-time waveform transmitted through a communication channel is often a carrier-modulated narrowband signal, i.e. a waveform whose bandwidth is much smaller than the carrier frequency. This is primarily due to bandwidth limitations of the channel.

Such a narrow-band signal can be expressed in the following form:

$$a(t) = \text{Re} \{g(t) e^{j2\pi f_c t}\} = \epsilon(t) \cos\{2\pi f_c t + \phi(t)\} , \quad (2.1)$$

where $g(t)$ is the so-called lowpass information bearing signal [Pro89], f_c the carrier frequency, Re denotes the real part of a complex number and $\epsilon(t)$ is the *envelope* of $a(t)$ (signals of constant envelope will be of special interest in this work). The principle of digital modulation is the following: consider the transmission of a block of K consecutive binary digits. This block may correspond to $L = 2^K$ different binary sequences, each of which is called a *symbol* s_i :

$$s_i = [b_1^{(i)} b_2^{(i)} \dots b_K^{(i)}] , \quad (2.2)$$

where $b_l^{(i)}$, $l = 1, \dots, K$, $i = 1, \dots, L$ is the bit at the l^{th} position of the i^{th} possible symbol. The set of all possible symbols is called the *alphabet* \mathcal{A} (of size L) of the transmitted symbols:

$$\mathcal{A} = \{s_1, \dots, s_L\} . \quad (2.3)$$

Then perform a one to one mapping of each of the L possible symbols to one of L different continuous-time lowpass waveforms $u_i(t)$, $i \in \{1, \dots, L\}$. When the i^{th} symbol is to be transmitted, transmit the narrowband signal (as in (2.1)) that corresponds to $u_i(t)$. The mapping of different symbols to different lowpass waveforms $u_i(t)$ can be done by attributing to the set of symbols a discrete set of amplitudes, frequencies and/or phases of $u(t)$. Such different mappings will correspond to different digital modulation techniques. A quite popular digital modulation technique is the so-called *Quadrature Amplitude Modulation* (QAM), in which the different lowpass waveforms are produced as follows:

$$u_i(t) = (R_i + jI_i) u(t) , \quad (2.4)$$

where each symbol s_i is first mapped to a complex number $A_i = R_i + jI_i$ which then multiplies a real-valued signal pulse $u(t)$ in order to produce the corresponding lowpass signal $u_i(t)$. By this one to one mapping $s_i \leftrightarrow A_i$ the symbol alphabet \mathcal{A} is represented by a set of points in the \mathcal{R}^2 complex plane. This set of points is called a *signal space diagram*, and the specific form it may take on for a particular modulation is called a *constellation*. Therefore, each point of the constellation corresponds uniquely to a symbol of the alphabet \mathcal{A} . If in (2.4) we restrict $I_i = 0$, $\forall i \in \{1, \dots, L\}$, then each symbol s_i is mapped onto a real number $A_i = R_i$ and PAM (*Pulse Amplitude Modulation*) is obtained:

$$u_i(t) = A_i u(t) , \quad (2.5)$$

where A_i may take on L different real values. Eq. (2.5) describes the so-called *double-sideband* PAM, i.e. the transmitted waveform $a_i(t) = \text{Re}\{A_i u(t) e^{j2\pi f_c t}\}$ occupies the double bandwidth of $u(t)$. If instead of the mapping in (2.5) the following one is used:

$$u_i(t) = A_i [u(t) \pm j\hat{u}(t)] , \quad (2.6)$$

where $\hat{u}(t)$ is the Hilbert transform of $u(t)$:

$$\hat{u}(t) = \lim_{\epsilon \rightarrow 0} \frac{1}{\pi} \int_{|\tau| > \epsilon} \frac{u(t - \tau)}{\tau} d\tau , \quad (2.7)$$

then $u_i(t)$ has the same spectral width as $u(t)$, and this corresponds to the *single-sideband* PAM. In the case of PAM, the signal space diagram will be a set of points on the real axis. Some typical PAM and QAM constellations are shown in figure 2.2. In the sequel we will use the

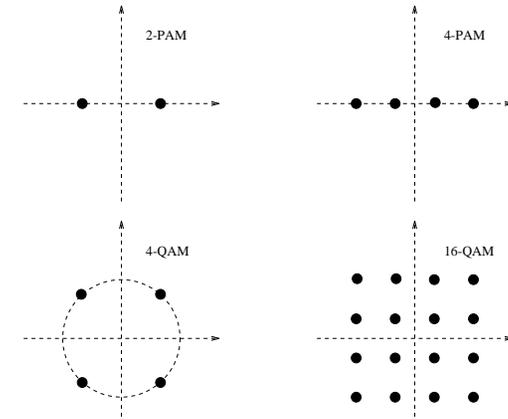


Figure 2.2: Some typical PAM and QAM constellations

term symbol to denote A_i instead of s_i , according to the mapping $s_i \leftrightarrow A_i$ described above.

2.3 Impairments of communication channels

The term communication channel corresponds to everything that separates the transmitter from the receiver: all the filters and processing stages used before transmission and after reception and the transmission medium. However, we often denote by “communication channel” the transmission medium alone. Some typical transmission media are copper wires, coaxial cables, optical fibers, the air, the water, e.t.c. As one would expect, the channel may influence in various ways the waveform that arrives to the receiver. Some serious impairments often met in communication channels are the following:

- Additive noise:

One of the most common impairments which is almost always present in any kind of transmission is the so-called additive noise, i.e. a random signal that is added to the transmitted

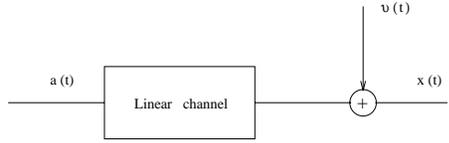


Figure 2.3: Linear noisy channel

waveform and may be due either to external (radiation of electric devices, electrical discharges, cosmic radiation, other transmitters) or to internal (thermal noise, shot noise) sources. Important quantities that characterize the additive noise $v(t)$ are its variance ($E(v(t) - E(v(t)))^2$) - E denotes statistical expectation -, its Power Spectral Density (PSD) and its bandwidth. Often the additive noise is modeled as a zero-mean white Gaussian stochastic process.

- Linear distortion:

We often consider that the communication channel is linear, i.e. it obeys the superposition principle: the output corresponding to a linear combination of input signals is the same linear combination of the outputs that would have been produced if each signal had excited the input by itself. Such a situation may arise for example when we have *multipath propagation*. A linear and time invariant (LTI) channel can be described by its impulse response $h(t)$:

$$x(t) = a(t) * h(t) , \quad (2.8)$$

where $a(t)$ is the continuous-time channel input, $x(t)$ the corresponding output and $*$ denotes convolution. In this case it makes sense to talk about the channel's frequency response, which leads to an equivalent representation of convolution in the frequency-domain:

$$X(f) = A(f) H(f) , \quad (2.9)$$

where $H(f)$ is the Fourier transform of $h(t)$:

$$H(f) = \mathcal{F}(h(t)) = \int_{-\infty}^{\infty} h(t) e^{-j2\pi ft} dt , \quad (2.10)$$

and $A(f)$, $X(f)$ are similarly defined as $\mathcal{F}(a(t))$ and $\mathcal{F}(x(t))$, respectively. Channels whose frequency response $H(f)$ is not uniformly flat at all frequencies are called *frequency selective* channels. For linear time-invariant noisy channels, the following model has been widely adopted: the noise is modeled as a random process, independent of both the input signal and the channel, that is added to the channel's *output*. In this case, the output of the noisy channel can be written as:

$$x(t) = a(t) * h(t) + v(t) = \int_{-\infty}^{+\infty} a(t - \tau) h(\tau) d\tau + v(t) . \quad (2.11)$$

Figure 2.3 shows the typical model of linear noisy channel that will be used extensively throughout the rest of the thesis.

- Fading

Due to nonstationary phenomena, the channel impulse response is often varying in time. A typical case of a time-varying impulse response is that of a mobile communication channel. The channel constantly changes as the mobile is moving and the physical medium between it and the base station changes. Channels with time-varying impulse responses are called fading channels and their frequency response is a function of two frequency variables. The problem with such channels is that at some time instants their frequency response may be close to zero at some frequencies, which might cause problems at the reception stage. A receiver designed for fading channels should therefore seriously take into account this undesired characteristic.

- Cross-talk interference

Often signals from other communication channels interfere with the signal that is supposed to use alone a channel. This kind of interference is called cross-talk interference and constitutes an important impairment of communication channels. Such a case arises for example typically in telephone lines and in mobile communications.

- Frequency offset

Often the carrier frequency of the received waveform has been altered w.r.t. its original value. This phenomenon is called frequency offset and should be given particular attention, especially in the case of synchronous receivers.

- Non-linear distortion

Sometimes non-linear distortion is also present in the received signal. In this case the channel can no longer be described by an impulse response (for example a Volterra-series modeling can be used to describe it). Non-linear channels are typical in optical transmission.

In this thesis we will only consider linear noisy channels, stationary for an adequately long time interval. As we are especially interested in the case of digital transmission, it is essential to describe the effect of linear distortion in this context.

2.4 Symbol rate, Sampling, Inter-Symbol-Interference, Channel eye

In the sequel we will use the so-called *baseband representation* of communication channels: in this representation we omit the carrier and proceed as if all the signals were transmitted and received in the baseband.

In digital communication systems the binary information stream is organized in blocks of K bits each and then for each block a different continuous-time waveform is transmitted. This transmission of waveforms is of course done at a constant rate, i.e. the time interval T

between the transmission of two consecutive waveforms is constant and is called the *symbol period* and its reciprocal is called the *symbol rate* or *Baud rate*:

$$f_B = \frac{1}{T} . \quad (2.12)$$

The transmitted signal can then be written as:

$$a(t) = \sum_{n=-\infty}^{\infty} a_n u(t - nT) , \quad (2.13)$$

where a_n is the symbol transmitted at time instant n and $u(t)$ the band-limited pulse. Figure 2.4 shows a typical transmitted waveform for a 2-PAM system. Supposing an ideal noiseless communication channel, the received continuous-time waveform $x(t)$ will equal $a(t)$ (apart from a possible delay and a scaling factor). Acquisition of the transmitted symbols is then simply done by sampling $x(t)$ at the Baud-rate, i.e. every T seconds:

$$x_k = x(t_0 + kT) , \quad k = 0, 1, \dots, \quad (2.14)$$

where t_0 accounts for the time instant that the first sample is taken. We will refer to receivers

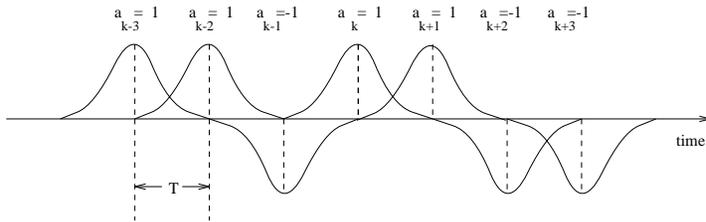


Figure 2.4: A 2-PAM transmitted continuous-time signal

that sample the received signal at the Baud-rate (or symbol-rate) as “Baud-rate receivers” or “ T -spaced receivers” in contrast to “fractionally-spaced receivers”¹, i.e. receivers that sample the received signal with a period T_s that is a *fraction* of T :

$$T_s = \frac{T}{m} , \quad m > 1 . \quad (2.15)$$

Consider now a linear-distortion channel. The frequency response of such a channel will typically occupy a specific frequency band. As a result of this the received waveforms $u_k(t)$ will spread in time and interleave in a way similar to the one shown in figure 2.5 for a 2-PAM. Mathematically, the received signal will be given by:

$$x(t) = \sum_{n=-\infty}^{\infty} a_n c(t - nT) + v(t) , \quad (2.16)$$

¹Fractionally-spaced receivers will be of particular interest in the second part of this thesis.

where $c(t)$ is defined as:

$$c(t) = \int_{-\infty}^{\infty} u(t - \tau) h(\tau) d\tau , \quad (2.17)$$

and $h(t)$ is the channel’s continuous-time impulse response. Sampling at the Baud rate will

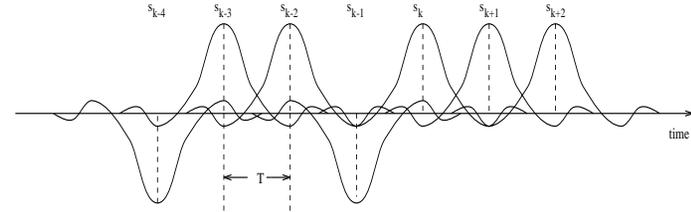


Figure 2.5: Received waveforms of a 2-PAM signal transmitted through a linear channel

then yield:

$$x_k = x(t_0 + kT) = \sum_{n=-\infty}^{\infty} a_n c(t_0 + (k - n)T) + v(t_0 + kT) , \quad (2.18)$$

which may also be written as:

$$x_k = \sum_{n=-\infty}^{\infty} a_n c_{k-n} + v_k , \quad (2.19)$$

where $c_k = c(t_0 + kT)$. Eq. (2.19) is very important to our work because it provides the baseband discrete-time model that we will assume for the received sampled process. Note that in this expression, x_k can be viewed as the output of a linear discrete-time filter with impulse response $\{c_i\}$ whose input is the discrete-time sequence of the transmitted symbols $\{a_i\}$. If we split the above sum in the following way:

$$x_k = c_0 a_k + \sum_{n \neq k} a_n c_{k-n} + v_k , \quad (2.20)$$

then it is clear that the noiseless part of the received sample at a certain time instant is the sum of the contribution of the transmitted symbol that corresponds to the same time instant ($c_0 a_k$) and an (undesired) cumulative contribution of symbols transmitted at other time instants ($\sum_{n \neq k} a_n c_{k-n}$). The presence of undesired contributions due to neighboring symbols is common in digital communications over linear channels and has been given the name of *Inter-Symbol-Interference* (ISI). This contribution represents a second kind of noise present in the received samples and may lead to high error rates in the detection of the transmitted symbols.

Intuitively, it should be clear that the symbol rate $\frac{1}{T}$ may significantly affect the resulting ISI: when T is very large, the adjacent pulses $a_n u(t - nT)$ will be fairly far from each other

and since they are practically time-limited, there will be a small amount of contributions (depending also on the channel) of adjacent pulses to the sample of a specific pulse. This intuition can be mathematically verified as follows: from (2.19), in order to have zero ISI, the discrete sequence $\{c_k\}$ should be a discrete Dirac function:

$$c_k = \delta_k = \begin{cases} 1, & k = 0 \\ 0, & k \neq 0 \end{cases} . \quad (2.21)$$

Eq. (2.20) can be written in the frequency domain as follows:

$$\frac{1}{T} \sum_{i=-\infty}^{\infty} C(f + \frac{i}{T}) = 1, \quad (2.22)$$

where $C(f)$ is the Fourier transform of $c(t)$. Eq. (2.22) may only be satisfied if the bandwidth Δf of $C(f)$ is greater than or equal to $\frac{1}{T}$. This is known as Nyquist's condition for zero ISI:

Nyquist's condition for zero ISI:

In order to achieve zero ISI for a channel with frequency response $C(f)$, (2.22) must be satisfied.

An immediate consequence of Nyquist's condition is the following: the ISI of a channel may only be zero if its bandwidth Δf is greater than or equal to the symbol rate $\frac{1}{T}$:

$$\Delta f \geq \frac{1}{T} . \quad (2.23)$$

If instead of the bandwidth Δf of $C(f)$ its baseband bandwidth Δf_{bb} is used, then (2.23) takes the following form:

$$\Delta f_{bb} \geq \frac{1}{2T} . \quad (2.24)$$

Therefore, according to Nyquist's condition, given a specific channel bandwidth and a receiver sampling the received signal at the symbol rate², there is an upper bound on the symbol rate that can be employed if one wants to make it possible to have zero ISI. This constitutes a practical limitation that must be given serious attention in high-rate communication systems.

In order to have an idea of the "amount" of ISI present at the received process $\{x_k\}$, the following *closed-eye measure* ρ of a channel is defined:

$$\rho = \frac{\sum_i |c_i| - \max_i |c_i|}{\max_i |c_i|} . \quad (2.25)$$

ρ measures the ratio of the "undesired" contributions (due to the *residual ISI*) over the desired contribution, represented by the (absolutely) largest coefficient term in the expression (2.19)

²A similar condition is given for the case of fractionally-spaced receivers in the second part of the thesis.

(the fact that the maximum and not the first term is considered to represent the "desired" contribution reflects the fact that a constant time-delay of the received sequence w.r.t. the transmitted one may be tolerated). When no ISI is present in $\{x_k\}$ $\rho = 0$. When $0 < \rho < 1$, the *channel eye* is said to be open, whereas for $\rho \geq 1$ the channel eye is said to be closed. The term "channel eye" is due to the eye-like form that is obtained in an oscilloscope if consecutive portions of a received continuous-time PAM signal of length T each are plotted on the same screen. The resulting diagram is called an *eye diagram* and is mathematically the set of points p_i defined as:

$$\{p_i\} = \{(t \bmod T, x(t)), t \in \mathcal{R}\} . \quad (2.26)$$

Assuming a noiseless channel, when the channel eye is open ($0 < \rho < 1$), the transmitted

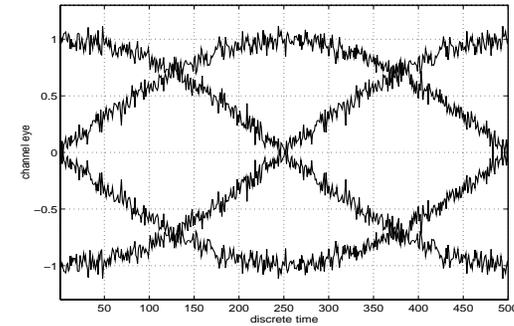


Figure 2.6: A typical channel eye of a 2-PAM system

symbol sequence can be accurately obtained at the receiver by a simple decision device with zero error probability. In other words, when the amount of ISI is small enough to keep the eye open, error-free transmission is theoretically achieved if no additive noise is present. When noise is present on the contrary, there will always be a non-zero probability of error, even for $\rho = 0$ (eye perfectly open). However the probability of error will increase as the eye opening decreases. A typical eye diagram for a 2-PAM system is shown in figure 2.6.

2.5 ISI cancelling, equalization

It is clear from the above discussion that in order to be able to obtain the transmitted data with a high reliability the receiver should try to reduce, or ideally, completely eliminate the ISI present in the received sampled process $\{x_k\}$. There exist both optimal and suboptimal receiver structures in order to combat ISI. The optimal structures optimize a Maximum Likelihood

(ML) criterion on a sequence of received samples assuming an a priori known distribution for the additive noise and the known discrete symbol alphabet, whereas suboptimal structures optimize other criteria. Assume a finite-length transmitted message of K symbols:

$$A_K = [a_0 \ a_1 \ \cdots \ a_{K-1}] . \quad (2.27)$$

If the size of the symbol alphabet is L , then there are L^K different possible transmitted sequences $A_K^{(i)}$, $i = 1, \dots, L^K$. The principle of ML receivers is to choose, among these L^K sequences the most likely one to have produced the received sampled sequence $\{x_k\}$. This is a typical detection problem. It turns out that when the additive noise is assumed to be Gaussian, the ML receiver will choose the transmitted sequence as follows [Pro89]:

$$\hat{A}_K = \arg \max_{A_K^{(i)}} J(A_K^{(i)}) = 2 \operatorname{Re} \left(\sum_n a_n^* f_n \right) - \sum_n \sum_m a_n^* a_m r_{n-m} , \quad (2.28)$$

where:

$$r_n = r(nT) = \int_{-\infty}^{\infty} c^*(t)c(t+nT)dt , \quad (2.29)$$

and:

$$f_n = f(nT) = \int_{-\infty}^{\infty} x(t)c^*(t-nT)dt . \quad (2.30)$$

The sequence $\{r_i\}$ represents the autocorrelation function of the channel $c(t)$, sampled at the rate $\frac{1}{T}$. The sequence $\{f_i\}$ on the other hand, can be derived as the output of a filter *matched*³ to $c(t)$ whose output is sampled at the rate $\frac{1}{T}$. The sequence $\{f_i\}$ forms a set of sufficient statistics.

The choice of the most likely sequence according to (2.28) can be done in an exhaustive way by calculating all the L^K metrics $J(A_K^{(i)})$, $i = 1, \dots, L^K$, but this will require a high computational complexity for big message lengths K . A substantially lower computational complexity can be achieved by employing the *Viterbi algorithm* [For73]. This is a sequence-detection technique based on dynamic programming that is well-known for its applicability and good performance. It is applicable when the channel response $c(t)$ has a finite duration, a condition that can always be made to hold with good approximation.

Often in practice, suboptimal receivers are used due to requirements such as low complexity, low cost, simplicity e.t.c. A commonly used suboptimal receiver structure for ISI cancellation is a linear discrete-time filter that operates on the sequence of received samples $\{x_i\}$. The coefficients of this filter are chosen in such a way so as to satisfy a specific criterion, the final aim being that the filter's output should be as close as possible to the transmitted symbol sequence $\{a_i\}$. Such a filter is called an *equalizer*. In a broader connotation [Pro89], the term *equalization* can be used to describe any kind of technique used by a receiver in order to compensate for the channel's ISI. The term originates from the fact that the purpose of an

³A filter matched to $c(t)$ is by definition a filter whose input $x(t)$ produces the output $\int_{-\infty}^{\infty} x(\tau)c^*(\tau-t)d\tau$.

equalizer is to produce a data stream "equal" to the transmitted symbol sequence. We will use the term *linear equalization* to describe the whole procedure of detecting the transmitted symbols with the use of a linear filter.

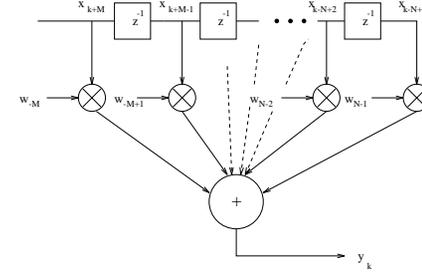


Figure 2.7: A linear equalization structure

When the received signal $x(t)$ is sampled at the symbol rate $\frac{1}{T}$, a linear equalizer W is a filter with impulse response $w(t)$ defined as follows:

$$w(t) = \sum_{i=-\infty}^{\infty} w_i \delta(t - iT) . \quad (2.31)$$

The equalizer output at time instant k will then be:

$$y_k = \sum_{i=-\infty}^{\infty} w_i x_{k-i} . \quad (2.32)$$

In practice, of course, the equalizer will be finite-length, $w(t) = \sum_{n=-N}^M w_n \delta(t - nT)$, and in this case it can be described by a vector of length $N + M + 1$ that contains the samples of its impulse response:

$$W = [w_{-M} \ \cdots \ w_{-1} \ w_0 \ w_1 \ \cdots \ w_N]^T . \quad (2.33)$$

The equalizer output can then also be written as:

$$y_k = X_k^H W , \quad (2.34)$$

where X_k is the *regression* vector defined as ⁴:

$$X_k^H = [x_{k+M} \ \cdots \ x_k \ \cdots \ x_{k-N+1}] . \quad (2.35)$$

Usually a linear equalizer is implemented with the use of a tapped-delay line (transversal filter), the delay between adjacent taps being T . In order to detect the transmitted symbols once the equalizer has been tuned to a setting that corresponds to the satisfaction of some

⁴The reasons for using complex conjugate transposition will become clear later on.

criterion, its output is then passed through a decision device that chooses at each time instant the symbol closest to it. This is not an optimal structure, but is nevertheless commonly used. Figure 2.7 shows the part of a receiver that corresponds to linear equalization:

Two among the best known methods for the tuning of a linear equalizer are:

- Zero Forcing (ZF) equalization

The equalizer is said to be zero-forcing when it minimizes the so-called *peak distortion* criterion. Consider the infinite-length equalizer described by eq. (2.31). The cascade of the equivalent discrete time channel response $\{c_i\}$ and the equalizer is a discrete-time filter whose i^{th} coefficient s_i is given by:

$$s_i = \sum_{j=-\infty}^{\infty} c_j w_{i-j} , \quad (2.36)$$

and which admits as input the symbol sequence $\{a_i\}$. Its output at time instant k will therefore be given in the absence of additive noise by:

$$y_k = s_0 a_k + \sum_{i \neq k} a_i s_{k-i} . \quad (2.37)$$

We shall call peak distortion the maximum possible value of the ISI term $\sum_{i \neq k} a_i s_{k-i}$ of the above expression:

$$D^{peak} = \max_{a_i \in \mathcal{A}} (|a_i|) \sum_{i \neq 0} |s_i| = \max_{a_i \in \mathcal{A}} (|a_i|) \sum_{i \neq 0} \left| \sum_{j=-\infty}^{\infty} c_j w_{i-j} \right| . \quad (2.38)$$

The peak distortion represents the worst-possible value of the ISI. As the equalizer is assumed to be infinite-length, it is able of completely nulling D^{peak} if its taps $\{w_i\}$ are set so as to satisfy:

$$\sum_{j=-\infty}^{\infty} c_j w_{i-j} = \delta_{i0} , \quad (2.39)$$

where $\delta_{n,k}$ stands for the Kronecker delta. The above equation takes the following form in the z-domain:

$$W(z) = \frac{1}{C(z)} . \quad (2.40)$$

Eq. (2.40) gives the ideal ZF equalizer in the absence of noise: it is the one whose z-transform is the reciprocal of the channel's discrete-time impulse response z-transform. Note that complete ISI elimination occurs only when the equalizer is of infinite length.⁵

When the equalizer is finite-length (see e.g. (2.33)), the ZF equalizer is defined as the finite-length equalizer of a specific length that minimizes D^{peak} . Note that in the absence of noise, an infinite-length ZF equalizer is also the optimal equalizer since error-free transmission is achieved. However when noise is present, the equalizer described by (2.40) is no longer optimal in the sense that it does not guarantee the minimal probability of error at the output

⁵It will be shown later on that zero-ISI ZF equalizers of finite length exist for fractionally-spaced receivers.

of the decision device. If the additive noise at the channel output is assumed to be white, of power spectral density N_0 ($\frac{N_0}{2}$ in baseband), the noise variance at the equalizer output can be shown to be equal to:

$$\sigma_b^2 = \frac{N_0 T^2}{2\pi} \int_{-\frac{\pi}{T}}^{\frac{\pi}{T}} \frac{1}{\sum_{i=-\infty}^{\infty} |C(\omega - \frac{2\pi i}{T})|^2} d\omega . \quad (2.41)$$

It is clear from (2.41) that the noise at the output of the ZF equalizer may be significantly amplified if the channel frequency response $C(f)$ is very small at some frequencies (such a phenomenon is usual for example in fading channels). In the extreme case where $C(f)$ equals zero for some frequencies⁶, the output noise variance σ_b^2 becomes infinite and will lead to a totally unreliable detected sequence at the output of the decision device! This amplification of noise variance at the entry of the decision device, heavily depending on the channel characteristics, constitutes the main drawback of ZF equalizers and is responsible for their poor performance in many practical situations.

- Minimum Mean Squared Error (MMSE) equalization

A different philosophy is behind the criterion of the so-called MMSE equalization: the equalizer now does not try to completely eliminate the ISI but to achieve a reasonable compromise between the ISI and the output noise variance σ_b^2 . The corresponding criterion is:

$$\min_W J(W) = E(|y_k - a_k|^2) = E(|X_k^H W - a_k|^2) , \quad (2.42)$$

i.e. the goal now is to minimize in a quadratic sense the deviation of the equalizer output at time instant k , y_k , w.r.t. the actually transmitted symbol at the same time instant, a_k . The solution to the above criterion is the MMSE equalizer which is given in the z-domain by:

$$W(z) = \frac{1}{C(z) + N_0} , \quad (2.43)$$

(compare (2.43) to (2.40)). $J(W)$ is minimized for the choice of $W(z)$ in (2.43) and achieves the value:

$$J_{min}(W) = \frac{N_0 T^2}{2\pi} \int_{-\frac{\pi}{T}}^{\frac{\pi}{T}} \frac{1}{\sum_{i=-\infty}^{\infty} |C(\omega - \frac{2\pi i}{T})|^2 + N_0 T} d\omega , \quad (2.44)$$

which represents the noise variance at the output of an MMSE equalizer (compare to (2.41)). Note that in contrast to the ZF equalizer, a channel frequency characteristic with deep spectral nulls will no more produce an infinite-variance noise at its output. On the other hand, it is obvious from (2.43) that in the presence of noise ($N_0 \neq 0$), the cascade channel-equalizer will not correspond to a Dirac impulse response. This means that in the presence of additive noise, the MMSE equalizer does not completely eliminate the ISI. These two remarks show the

⁶Note that in this case $C(z)$ will have zeros on the unit circle

compromising action of the MMSE equalizer between noise-amplification and ISI cancelling. Note however that in the absence of noise, both the MMSE and ZF equalizers are equal and provide error-free transmission (zero ISI, zero noise).

2.6 Adaptive filtering and equalization

A practical way to obtain the equalizer setting corresponding to a specific minimization criterion (e.g. like the one in (2.42)) is to do this adaptively, i.e. with the help of an adaptive algorithm that makes an initial guess for the equalizer setting and then updates this setting until it converges to the desired value. This procedure is called *adaptive equalization* and it can be considered as an application of *adaptive filtering*. A typical setup to which adaptive filtering applies is the system identification setup shown in the figure 2.8. In this figure, the task of the adaptive algorithm is to adapt the filter W in such a way so as to minimize the mean squared difference between its output at time instant k , y_k , and a “desired” response sample at the same time instant, d_k . The “desired” response is supposed to be the output of

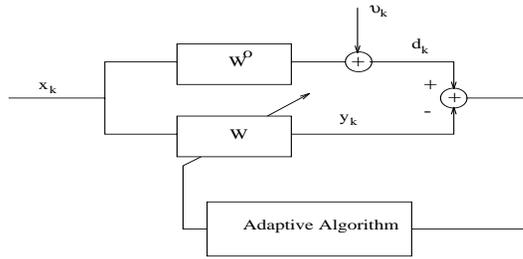


Figure 2.8: A typical system identification setup

an optimum linear FIR filter W^o with input $\{x_k\}$ plus some additive noise $\{v_k\}$:

$$d_k = X_k^H W^o + v_k, \quad (2.45)$$

X_k being a regression vector of the same length (say N) as W^o :

$$X_k^H = [x_k \cdots x_{k-N+1}]. \quad (2.46)$$

The filter W is adapted by the algorithm (usually) every time a new data sample x_k is available (i.e. at the symbol rate) and the corresponding setting is denoted by W_k . The minimization criterion used is typically quadratic of the following form:

$$\min_W J(W) = E(|d_k - y_k|^2) = E(|d_k - X_k^H W|^2). \quad (2.47)$$

The cost function $J(W)$ is quadratic and hence convex in W . It therefore admits a unique minimum point given by:

$$W = \{E(X_k X_k^H)\}^{-1} E(X_k d_k), \quad (2.48)$$

provided of course that the sample covariance matrix of the input signal, $R = E(X_k X_k^H)$ is nonsingular (this condition is always satisfied in practical cases). An adaptive filtering algorithm should therefore be able of converging after a certain number of iterations towards the solution in (2.48). Some performance measures for adaptive filtering algorithms are:

- Convergence time, i.e. the number of iterations needed to converge
- Complexity, i.e. the number of operations performed at each iteration of the algorithm
- Steady-state error, i.e. the bias of the found solution and the variance of the error after convergence
- Robustness w.r.t. the colouring of the input process $\{x_i\}$
- Robustness w.r.t. the variance and/or the colouring of the additive noise process $\{v_i\}$

Two among the best known adaptive filtering algorithms are the LMS and the RLS algorithms. The LMS is a stochastic gradient algorithm, i.e. an algorithm that performs a search along the opposite direction of the instantaneous gradient $\nabla J_k(W) = X_k(y_k - d_k)$. The resulting algorithm is:

$$\begin{cases} \epsilon_k &= d_k - X_k^H W_k \\ W_{k+1} &= W_k + \mu X_k \epsilon_k \end{cases}, \quad (2.49)$$

where μ is the so-called *stepsize* parameter, which controls the deviation of the “next” filter setting W_{k+1} w.r.t. W_k . The value of μ influences both the convergence speed and the steady-state error of the algorithm. In order to have a stable operation of the algorithm, the stepsize μ should lie in the interval:

$$0 < \mu < \frac{2}{\|X_k\|^2}, \quad (2.50)$$

and the maximum convergence speed is attained for $\mu_k = \frac{1}{\|X_k\|^2}$. The algorithm’s complexity is $2N$ multiplications/iteration where N is the equalizer length and its convergence speed heavily depends on the colouring of the input signal. The most favourable case is the one of a white input which provides the maximum convergence speed. The more strongly coloured the input signal gets, the larger number of iterations will be needed to reach the steady-state. In the extreme case where the input signal is an AR process with poles on the unit circle, the filter coefficients will practically never converge. However, despite its limitations, the LMS algorithms is one of the most popular algorithms for adaptive filtering due to its simplicity and low computational complexity.

Another popular class of algorithms for adaptive filtering is the so-called RLS algorithm. Without getting into detail in the algorithm's organization, we outline the essential computations performed at each iteration:

$$\begin{aligned} \epsilon_k &= d_k - X_k^H W_k \\ R_k^{-1} &= \lambda^{-1} R_{k-1}^{-1} - \lambda^{-1} R_{k-1}^{-1} X_k (1 + X_k^H \lambda^{-1} R_{k-1}^{-1} X_k)^{-1} X_k^H \lambda^{-1} R_{k-1}^{-1} \\ W_{k+1} &= W_k + R_k^{-1} X_k \epsilon_k \end{aligned} \quad (2.51)$$

λ is a parameter called *forgetting factor* that controls the tracking of the sample covariance matrix R_k as well as the convergence speed (typical values are between 0.9 and 1), and is of particular importance when the input signal is non-stationary. The updating equation for R_k^{-1} is the so-called *Ricatti equation*⁷. It permits to avoid the inversion of R_k , an operation that would require $O(N^3)$ multiplications/iteration. An important quantity in the algorithm is the so called *Kalman gain*, defined as $R_k^{-1} X_k$.

The RLS algorithm has in general a better performance than the LMS algorithm⁸: its convergence is faster and steady-state is typically reached at $\sim 2N$ iterations, irrespective of the colouring of $\{x_i\}$. The steady-state error variance is also often smaller than the one of LMS. This is achieved on the other hand at the cost of a higher computational complexity (of the order of $O(N^2)$ multiplications/iteration). However fast RLS versions (FTF[CK84], FAEST[CMK83], SFTF[SK91], FNTF[MT93]) that keep the complexity down to $O(N)$ multiplications/iteration have been derived based on the particular shift-invariance structure of the data matrix and the resulting low *displacement rank* of the sample covariance matrix R_k .

Several other adaptive filtering algorithms have been proposed during the last decade, offering a big variety of compromises between their performance characteristics. Some of them (namely the NLMS and the APA) will be given special attention in the sequel. For the moment we will mostly adhere to the LMS which has been widely used in channel equalization.

The LMS can be used to update an MMSE equalizer corresponding to the cost function in (2.42), which has the same form as the cost function in (2.47). It is clear that in (2.42), the role of the "desired samples" d_k is played by the actually transmitted symbols a_k . The LMS equalizer will then be described by the following equations:

$$\begin{cases} \epsilon_k = a_k - X_k^H W_k \\ W_{k+1} = W_k + \mu X_k \epsilon_k \end{cases} \quad (2.52)$$

Even though described by the same equations as (2.49), the setup corresponding to (2.52) is quite different than the typical system identification setup shown in figure 2.8. The setup corresponding to adaptive equalization is shown in figure 2.9. We now outline the main differences of the two setups shown in figures 2.8 and 2.9:

⁷The Ricatti equation is derived by applying the so-called *Matrix Inversion Lemma* (MIL) to the updating equation for R_k .

⁸This may however not be true for nonstationary environments.

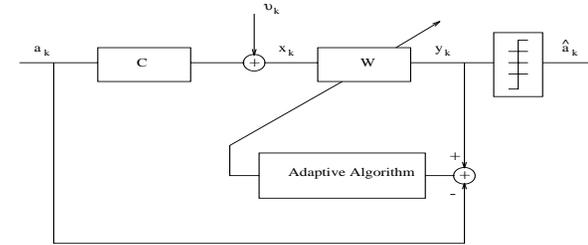


Figure 2.9: A typical adaptive equalization setup

- In fig. 2.9, the "desired" process $\{a_i\}$ is not corrupted by additive noise as in fig. 2.8.
- In fig. 2.9, the input process $\{x_i\}$ is corrupted by additive noise, which is not the case in fig. 2.8.
- In fig. 2.9, the equalizer tries to match the impulse response of the channel's "inverse", i.e. fig. 2.9 describes an inverse problem. On the other hand, fig. 2.8 describes a direct problem, in the sense that the adaptive FIR filter W tries to match the FIR filter W^o .
- Figure 2.8 depicts an "identification" or *estimation* problem, i.e. the aim is to estimate W^o , whereas in fig. 2.9 the final aim is the acquisition of the transmitted symbols $\{a_i\}$. This is a *detection* problem, and the estimation of the optimal equalizer is only an intermediate step in this direction.⁹

The differences outlined above have a number of impacts on the performance and the design of adaptive equalizers. The fact that the input process $\{x_i\}$ is corrupted by additive noise (i.e. additive noise is present at the *input* of the adaptive filter) may have a significant impact on its performance and makes the analysis of such an adaptive filter a more difficult task than in the case of fig. 2.8. The fact that we are dealing with an inverse problem, has the obvious disadvantage that a problem of *existence* of an inverse filter arises. Namely, an FIR filter of any length will never be long enough to match exactly the inverse of the channel impulse response. In the extreme case of the channel FIR filter having zeros on the unit circle, its impulse response is infinite-length and cannot be adequately approximated by any FIR filter. This is a serious limitation of the setup in fig. 2.9, which excludes a whole class of channels for equalization.

It should be also noted that there exist other equalization structures that do not obey the setup of figure 2.9. Two important structures should be mentioned:

⁹As already mentioned, the splitting of the problem in an estimation part (equalizer) and a detection part (threshold detector), as shown in fig. 2.9, is neither optimal nor unique. Other setups used are for example the *decision-feedback equalization*, or ML equalization.



Figure 2.10: The structure of a GSM time-slot

- Fractionally-spaced equalizers (FSE's)
- Decision Feedback Equalizers (DFE's)

The principle of fractionally-spaced equalization (see [Ung76]) is based on the *oversampling* of the received continuous-time signal and will be given special attention in the second part of the thesis. The principle of decision feedback equalization (see [BJ79]) is based on the feedback of a number of decisions to the adaptation algorithm in order to improve the performance. The resulting process is a nonlinear equalization process. DFE's will not be discussed throughout this thesis, which is primarily devoted to linear equalization.

2.7 Training sequence, Decision Directed (DD) equalization

It is clear from the above that for an LMS equalizer to be realizable, access to the actually transmitted sequence of symbols $\{a_i\}$ is needed. Of course this does not make perfect sense because if the actually transmitted symbols were known, there would be no reason for equalization! What usually happens in practice is that a "pre-arranged" sequence of symbols of finite length that is *a priori* known to the receiver is transmitted at the beginning of transmission. In this "start-up" period during which no useful information is transmitted, the receiver indeed has knowledge of the transmitted sequence $\{a_i\}$, which can allow the adaptive algorithm to converge towards an acceptable equalizer setting (by acceptable setting we mean one that opens the channel eye enough to lead to a correct retrieval of the transmitted symbols by the decision device with a high probability).

Once the equalizer has converged to an acceptable setting, transmission of information can begin, and, provided that the channel characteristics have not been changed, it will lead to a "correct" detection of the transmitted symbols. When the channel is time varying, a training sequence has to be sent periodically in order to allow to the receiver to adapt to the new channel setting. An example of a system that makes use of a training sequence is shown in figure 2.10. This figure shows the structure of a data frame of the European GSM system for mobile cellular communications. A different data sequence of 148 bits is transmitted every 4.6 msec. As the communication channel has probably been changed in the time interval between

the transmission of two consecutive frames, a training sequence of 26 bits is provided each time to the receiver. An obvious disadvantage of such a system is that the training sequence is responsible for a reduction of the useful information rate with respect to the total information rate of $\frac{26}{148}$ (this loss can be equivalently translated to a waste of bandwidth).

As the length of the training sequence is not always sufficiently long to allow for a good estimation of the channel inverse, the adaptation of the equalizer often continues after the end of the start-up period as well. The problem arising now of course is that there is no more access to the transmitted symbols that played the role of "desired" response. A remedy often used to this problem is to use as "desired" samples d_k the outputs of the decision device:

$$d_k = \hat{a}_k = \text{dec}(y_k) . \quad (2.53)$$

This is the so-called *Decision-Directed* (DD) principle and the idea behind it is that if the detection is of relatively good quality, the outputs of the decision device will be (often enough) identical to the actually transmitted symbols and one can hope that this will help improve the equalizer setting which will in its turn improve the detection quality and so on. The corresponding (LMS-like) Decision-Directed Algorithm (DDA) is given by:

$$\begin{cases} \epsilon_k &= \hat{a}_k - X_k^H W_k \\ W_{k+1} &= W_k + \mu X_k \epsilon_k \end{cases} . \quad (2.54)$$

In this scenario, the whole procedure is expected to converge towards a steady-state where good performance has been achieved. This is however not always the case and the dominating result in literature seems to be that DD equalization will only work if the initial channel eye is already open.¹⁰

2.8 Blind equalization

As already mentioned, the constraint of using a training sequence may be prohibitive for different reasons. Some of them are listed below:

- The useful information rate is reduced, or equivalently, the necessary bandwidth necessary for the same rate of useful information is increased
- For a fast varying channel a training sequence has to be sent very frequently
- A new training sequence has to be sent at the beginning of a new communication in order to initialize the receiver. In some cases, as for example in local area networks (LAN's), if one user loses its connection to the master server, the transmission of a training sequence to this user will also interrupt the communication towards other users

¹⁰We will see in chapter 5 that the performance of DD equalization also depends on the shape of the symbol constellation.

For all the above reasons it is desired to be able to obtain the transmitted sequence of symbols without the use of training sequence. An equalization procedure that makes no use of training sequence is called *Blind Equalization* (BE). The DD equalization discussed above has probably been the first BE procedure ever proposed, since it makes no use of training sequence. However, the poor performance of DD equalization combined with the increasing interest in the domain have resulted during the last 20 years in a rich literature for BE. In fact, BE can be considered as a special case of the so-called *Blind Deconvolution* (BD) problem. The aim of BD is to identify a system based only on measures of its output, i.e. given no access to its input. On the contrary, knowledge of the statistical properties of the input (which is usually modeled as a stochastic process) is provided. In fact it is this knowledge combined with the model used for the system that may allow identification of the system. What is actually exploited in BD is the statistics of the output of the system. The same holds of course for BE.

We now focus on a linear (say, in general IIR) channel, i.e. the z-transform of its impulse response is given by:

$$C(z) = \frac{B(z)}{D(z)}, \quad (2.55)$$

where $B(z)$ and $D(z)$ are (scalar) polynomials of z^{-1} and/or z . $C(z)$ is supposed to be stable, i.e. all roots of $D(z)$ are inside the unit circle. $C(z)$ is said to be *Minimum Phase* (MP) if the roots of its numerator $B(z)$ are also inside the unit circle, *maximum phase* if they are outside the unit circle and mixed-phase if they are both inside and outside the unit circle. The following is a fundamental identification result in BE:

Necessary condition for the identifiability of a linear channel from its output:¹¹

Identification of the channel described by (2.55) from its output second-order statistics (SOS) is only possible if $C(z)$ is minimum phase. Identification when $C(z)$ is mixed-phase is however possible if statistics of order higher than two of its output are used, provided that the input to the system is not a Gaussian process.

A way to verify the result given above is the following: first express the frequency response of the channel as:

$$C(\omega) = |C(\omega)|e^{j\Phi(\omega)}, \quad (2.56)$$

where $|C(\omega)|$ and $\Phi(\omega)$ account for the magnitude and phase, respectively of $C(\omega)$. The power spectral density (PSD) of the channel's output $y(t)$ is given by:

$$S_{yy}(\omega) = |C(\omega)|^2 S_{aa}(\omega), \quad (2.57)$$

¹¹An extension of this result to the case of a single-input multi-output system will be given in the second part of this thesis.

where $S_{aa}(\omega)$ is the PSD of the input process $a(t)$. From (2.57), provided that $S_{aa}(\omega)$ is known, the magnitude of the channel frequency response can be identified. However, it is clear from (2.57) that no information about the phase $\Phi(\omega)$ of the channel frequency response can be extracted from $S_{yy}(\omega)$. Therefore, only the magnitude of $C(z)$ can be identified by use of SOS (the PSD is the Fourier Transform of the autocorrelation function). However $|C(z)|$ will be sufficient for the identification of $C(z)$ if the channel is MP, since then there exists a unique phase $\Phi(z)$ for a given magnitude $|C(z)|$ (the same would be valid if we knew that $C(z)$ is maximum phase). Since SOS do not suffice for the identification of a NMP channel, this imposes a first major limitation on BE:

- Limitation 1: non-identifiability of a NMP channel when the input process is Gaussian:

This is due to the fact that all the statistical information of a Gaussian process is contained in its first two moments. The fact that a Gaussian input sequence is excluded in BE is not very important in the sense that the distribution of the input process is practically never Gaussian in digital communications. However, due to a technique called *constellation shaping* used in digital communications, there is sometimes a tendency of the "shaped" constellation to approach a Gaussian distribution [ZPV91]. This fact can make the BE problem a more difficult task.

At this point it is also worth mentioning two other major limitations that seem to be inherent in the BE setup:

- Limitation 2: identifiability up to a rotational factor:

Note from (2.57) that even for a MP channel, the channel frequency response is only identifiable to within a constant scalar of unit magnitude, i.e. any $C'(\omega)$ of the following form:

$$C'(\omega) = C(\omega)e^{j\theta}, \quad (2.58)$$

for any θ can be identified as well as the sought channel frequency response. In other words, an equalizer using this method cannot determine but up to a *rotated* version of the transmitted constellation. As it will be shown in the sequel, this phenomenon often arises in BE.

- Limitation 3: identifiability up to a constant time-shift

Another limitation inherent in BE is the fact that the channel can only be identified up to a *constant time-delay*. This is due to the fact that a NMP invertible channel in general will have an inverse that is *noncausal*. An immediate consequence of that is that the inverse channel (approximation) (in the z-domain) can only be defined as [BG83]:

$$S^{-1}(z^{-1})S(z^{-1}) = z^{-l}, \quad (2.59)$$

l being a constant integer. Therefore, (blind) equalization is not possible instantaneously but up to a constant time-shift.

In contrast to the above-mentioned frequency-domain approach, a time-domain approach can be followed as well. The solution to the problem when the channel is minimum-phase can be directly found by *linear prediction* (see [Mak75]) on the noise-free received sequence $\{x_i\}$. Namely, it has been shown that in this case, the input sequence $\{a_i\}$ is equal to the *innovations process* of $\{x_i\}$ [Hay91]:

$$\hat{a}_k = x_k - \hat{x}_k, \quad (2.60)$$

where \hat{x}_i is the linear MMSE prediction:

$$\hat{x}_i = E(x_k X_{k-1}) \left(E(X_{k-1} X_{k-1}^H) \right)^{-1} X_{k-1}, \quad (2.61)$$

(the prediction order equals the length of X_k). However, this is not the solution in the case of a NMP system, since the solution in (2.61) only requires autocorrelations of the received process, i.e. SOS. In the NMP case, the \hat{a}_k are related to the a_k by an all-pass filter (transforming one white noise to another one).

A sufficient condition for the identification of a NMP linear channel from its output's statistics was stated in 1980 by Benveniste *et Al.* [BGR80] and goes as follows:

Sufficient condition for the identifiability of a NMP linear channel from its output:

If the probability density functions (pdf's) of *all orders* of the equalizer output process $\{y\}$ equal the corresponding pdf's of the input process $\{a_i\}$, then the channel is identifiable from $\{y_i\}$.

In other words, if all the moments of the output process match the corresponding moments of the input process, identification is possible. However this is not a necessary condition.

A necessary and sufficient condition for the identifiability of a linear system from its output was stated by Shalvi and Weinstein in 1990 [SW90] and is as follows:

Necessary and sufficient condition for the identifiability of a NMP linear channel from its output:

A linear channel is identifiable from its output if and only if the following two conditions hold:

$$\begin{cases} E(|y|^2) = E(|a|^2) \\ |K(y)| = |K(a)| \end{cases}, \quad (2.62)$$

where $K(\cdot)$ is the *Kurtosis* of a process defined as:

$$K(z_i) = E(|z_i|^4) - 2E^2(|z_i|^2) - |E(z_i^2)|^2. \quad (2.63)$$

It is clear from the above that in order to be able to identify a mixed-phase channel one has to exploit moments of its output of order higher than two. However, if the input process $a(t)$

is a Gaussian stochastic process, the output will also be Gaussian, and all information about its statistics will be contained in its moments of order one and two. This verifies the fact that identification of a mixed-phase system that is excited by a Gaussian input is not possible based on statistics of its output.

Based on the way HOS are used in order to achieve channel identification, we can discern two different categories of methods for blind channel equalization:

- Methods that do not use "explicitly" HOS (moments and cumulants) but try to equalize by optimizing a criterion that contains "implicitly" information of higher order moments of the output via nonlinear functions. These methods belong to the so-called *Bussgang* family of blind equalizers. A simple example of such BE techniques is DD equalization, where the minimization criterion is the cost function:

$$J^{DD} = E(\text{dec}(y_k) - y_k)^2, \quad (2.64)$$

and the corresponding nonlinearity is $\text{dec}(y_k)$.

- Methods that use "explicitly" HOS of the output for the channel identification. These are typically cumulant-based methods (see for example [Nik88], [GM89], [HN91],[FVM92], [ZMM93]).

A tutorial presentation of these two big classes of BE methods can be found in [BR85]. However, in the recent literature several novel equalization techniques have been proposed that cannot be directly put in one of the above two families of techniques. Namely, there exist Maximum Likelihood BE techniques (see e.g. [GW92]) as well as BE techniques based on the exploitation of some kind of diversity (e.g. space diversity - time diversity).

In the first part of this thesis we will focus on Bussgang-like techniques for BE. In the second part we will consider diversity techniques based on an over-sampling equalization setup.

2.9 Bussgang techniques for blind equalization

As already mentioned, the Bussgang blind equalizers optimize (usually minimize) a criterion which is a nonlinear (and usually memoryless) cost function of the equalizer's output y :

$$\min_W J(y) = J(X^H W). \quad (2.65)$$

A Bussgang BE setup is depicted in figure 2.11. We will present at this point some of the best known BE algorithms of the Bussgang type:

- The Sato Algorithm

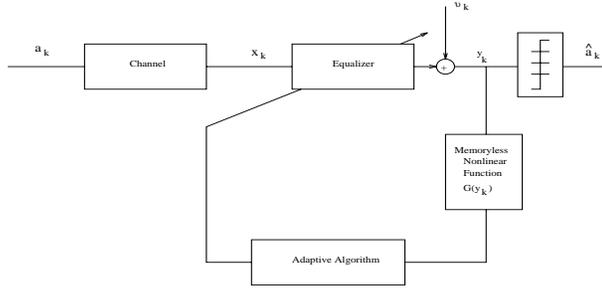


Figure 2.11: A Bussgang blind equalization setup

The Sato algorithm was proposed by Y. Sato in 1975 [Sat75] and in its original form was conceived for multi-PAM (real) constellations. Sato's motivation was that the conventional (by that time) adaptive equalization algorithm (i.e. the DD algorithm) failed to readapt to a newly connected channel in multi-level amplitude modulation systems. The cost function proposed by Sato was the following:

$$\min_W J^{\text{Sato}}(W) = E((y - \gamma \text{sign}(y))^2), \quad (2.66)$$

where sign denotes the usual sign function of a real scalar:

$$\text{sign}(r) = \frac{r}{|r|} = \begin{cases} 1, & r > 0 \\ -1, & r < 0 \end{cases}, \quad (2.67)$$

and γ is a scaling factor defined as:

$$\gamma = \frac{E a_k^2}{E |a_k|}. \quad (2.68)$$

Sato's idea behind this algorithm was to employ a binary decision device instead of a multi-level decision device, since the problem of the DD algorithm seemed to be that it couldn't cope with multi-PAM. Therefore Sato decomposed the multilevel into a polarity signal and a remaining signal and classified his decisions based only on the polarity signal, treating the remaining signal as random noise. For example, the levels of a 8-PAM signal are given by:

$$A_i = V \left(\frac{1}{2} d_1 + \frac{1}{4} d_2 + \frac{1}{8} d_3 \right), \quad (2.69)$$

where $d_j = \pm 1 (j = 1, 2, 3)$ and V is a constant factor. Sato's equalizer only used d_1 in order to classify the received samples as positive or negative, and used only this information in order to adapt his equalizer. The gradient of the cost function (2.66) is found as follows:

$$\begin{aligned} \nabla J^{\text{Sato}} &= \frac{\partial}{\partial W} \left(E(y_k - \gamma \text{sign}(y_k))^2 \right) = E \left(\frac{\partial y_k}{\partial W} \frac{\partial}{\partial y_k} (y_k - \gamma \text{sign}(y_k))^2 \right) = \\ &= 2E \left(\frac{\partial y_k}{\partial W} (y_k - \gamma \text{sign}(y_k)) \frac{\partial}{\partial y_k} (y_k - \gamma \text{sign}(y_k)) \right) = 2E (X_k (y_k - \gamma \text{sign}(y_k))). \end{aligned}$$

The corresponding stochastic gradient algorithm (Sato) will have the form

$$W_{k+1} = W_k - \mu \nabla J_k^{\text{Sato}},$$

where J_k^{Sato} accounts for the instantaneous value of ∇J^{Sato} at iteration k :

$$W_{k+1} = W_k + \mu X_k (\gamma \text{sign}(y_k) - y_k). \quad (2.70)$$

Eq. (2.70) describes the Sato algorithm for multi-PAM signals. One should note that in the 2-PAM case, the Sato coincides with the DD algorithm.

In order to analyze the algorithm's performance Sato made the following three assumptions:

- The PAM modulation admits an infinity of levels (infinite length alphabet), i.e. the transmitted symbols have a continuous uniform distribution in a symmetric interval $(-V, V)$. In this case a transmitted symbol can be written as:

$$a = V \left(\frac{1}{2} d_1 + \frac{1}{4} d_2 + \dots + \frac{1}{2^i} d_i + \dots \right), \quad (2.71)$$

$$(d_i = \pm 1, i = 1, 2, \dots).$$

- The overall combined channel-equalizer impulse response is infinite length
- The channel eye is open ($\rho < 1$) at the beginning of equalization

Under these assumptions, the cost function (2.66) can be proven [Sat75] to be equal to

$$E(y_k - \gamma \text{sign}(y_k))^2 = \frac{V^2}{3} \sum (h_i^2) - \gamma V \left(s_0 + \frac{1}{3} \sum' \frac{h_i^2}{s_0} \right) + \gamma^2, \quad (2.72)$$

where $\{s_i\}$ is the overall impulse response, s_0 is the greatest coefficient of $\{s_i\}$ at the beginning of equalization and \sum' denotes summation over all indices except the 0-lag index. From eq. (2.72) it can be shown that (under the above assumptions) the Sato cost function is a convex function¹² of the equalizer parameters. The convexity of the cost function guarantees that it has a unique minimum point. Moreover, this minimum point will provide the optimum equalizer setting, since this setting nulls the positive cost function $J^{\text{Sato}}(W)$. Therefore, under the above assumptions, the Sato algorithm will converge (if μ is chosen so as to provide stability) towards the optimal equalizer.

As is noted by Sato, the assumption of an initially open eye ($\sum' |s_i| < |s_0|$), which is a strong assumption, is sufficient for the proof of (2.72) but probably not necessary. In fact, the major advantage of the Sato algorithm, i.e. the fact that it may be able to open the channel

¹² $f(x)$ is said to be a convex function of x if for any $0 < \alpha < 1$, the following holds: $f(\alpha x_1 + (1 - \alpha)x_2) \leq \alpha f(x_1) + (1 - \alpha)f(x_2)$, for any $x_1 < x_2$ in the domain of definition of f .

eye for a multi-PAM system, even if it is initially closed, is not proven by the above analysis due to this limiting assumption!

However, in the subsequent literature, a result that removed this assumption was found by Benveniste *et Al.* [BBC⁺78] and Benveniste and Goursat [BG84]. In order to present this result we first give the following definition [BG84]:

Definition: A distribution v is said to be *sub-Gaussian* in one of the following cases:

$$\begin{cases} v \text{ is uniform over } [-d, +d] \\ v(dx) = K e^{-g(x)} dx \end{cases}, \quad (2.73)$$

where K is a constant and g is an even function, differentiable except possibly at the origin, such that $g(x)$ and $\frac{g(x)}{x}$ are strictly increasing over \mathcal{R}_+ . The result given in [BG84] is as follows:

Theorem: If the distribution v of the input sequence $\{a_i\}$ is sub-Gaussian, then the Sato cost function (2.66) admits as only local (and also global) minima the inverse channels $\pm c^{-1}$.

Therefore, according to this theorem, the Sato cost function is convex w.r.t. the (infinite in number) equalizer parameters, provided that the input sequence is sub-Gaussian. This is an asymptotic result in the sense that it assumes (as Sato did as well) a continuous input distribution, which is not a realistic hypothesis. In the realistic case of a discrete-time input distribution, the above theorem does not hold (even for an infinite number of equalizer taps) and the Sato cost function indeed has local minima that do not correspond to the optimal solutions.

It still seems however that Sato's performance is clearly better than DDA's performance in realistic multi-PAM cases with discrete input distributions. This may be justified by the different form of the local-minima regions in the two cases (as noted by Mazo, the local minima regions of DD in multi-PAM systems do not have the form of cones) whereas the corresponding Sato local minima regions are cone-shaped. These aspects will be discussed in chapter 5.

A direct extension to Sato's algorithm for QAM constellations is the so-called *Generalized Sato Algorithm (GSA)*, which has the following form:

$$W_{k+1} = W_k + \mu X_k (\gamma \text{csign}(y_k) - y_k), \quad (2.74)$$

where the "complex sign" function $\text{csign}(\cdot)$ of a complex scalar $r = r_R + jr_I$ is defined as:

$$\text{csign}(r_R + jr_I) = \text{sign}(r_R) + j \text{sign}(r_I). \quad (2.75)$$

A theorem similar to the one mentioned above has proven the convexity of the GSA as well, under the assumption that $\text{Re}(i)$ and $\text{Im}(a_i)$ are two *independent* i.i.d sequences with the same sub-Gaussian distribution v .

The Generalized Sato Algorithm classifies the received samples in one of the four quadrants of the complex plane \mathcal{R}^2 . This philosophy respects Sato's principle in the sense that the classified symbols have all the same magnitude, as in the real case. However, it does not respect it (except for 4-QAM constellations) in the sense that the classified symbol has an altered phase with respect to the received sample, which is not of course the case in the real case! We will present in chapter 5 a Sato-type algorithm that maintains the phase also for any QAM constellation.

- The Godard algorithms for BE

An important class of algorithms of the Bussgang type is the so-called Godard class of blind equalization algorithms. This class of algorithms was proposed by D. Godard in 1980 [God80]. The Godard cost function is defined as:

$$J_p^{\text{God}}(W) = \frac{1}{2p} E (|y|^p - r_p)^2, \quad p = 1, 2, \dots, \quad (2.76)$$

where r_p is a constant scalar called *dispersion constant* and defined as:

$$r_p = \frac{E |a_k|^{2p}}{E |a_k|^p}. \quad (2.77)$$

The philosophy of the Godard criterion (2.76) is that the criterion penalizes the deviations of the magnitude of the received signal with respect to a constant scalar. The constant scalar is r_p and represents the radius of a circle that "best fits" in some way the transmitted constellation. Of course the cost function (2.76) only makes perfect sense for *constant modulus* constellations, i.e. PAM or QAM constellations with the property that all the symbols of their alphabet have the same magnitude in the complex plane \mathcal{R}^2 . However, this criterion has been proven to make sense even in non-constant-modulus constellations, as we will see in the sequel.

The gradient of the Godard cost function is equal to:

$$\begin{aligned} \nabla J_p^{\text{God}} &= \frac{\partial}{\partial W} \left(\frac{1}{2p} E (|y_k| - r_p)^2 \right) = E \left(\frac{\partial y_k}{\partial W} \frac{\partial}{\partial y_k} (|y_k|^p - r_p)^2 \right) = \\ &= \frac{1}{p} E \left(\frac{\partial y_k}{\partial W} (|y_k|^p - r_p) \frac{\partial}{\partial y_k} (|y_k|^p - r_p) \right) = E \left(X_k y_k |y_k|^{p-2} (|y_k|^p - r_p) \right). \end{aligned}$$

The resulting stochastic gradient algorithm is:

$$W_{k+1} = W_k + \mu X_k y_k |y_k|^{p-2} (r_p - |y_k|^p). \quad (2.78)$$

Eq. (2.78) is the famous Godard algorithm for BE. The two best known Godard algorithms are derived by (2.78) for $p = 1$ and $p = 2$. These algorithms have been also proposed by Treichler and Agee [TA83] and Treichler and Larimore [TL85] and are commonly referred to as *Constant Modulus Algorithms*. In the first case the *CMA 1-2* algorithm is derived:

$$W_{k+1} = W_k + \mu X_k (r_1 \frac{y_k}{|y_k|} - y_k). \quad (2.79)$$

Note that in the case of a *real* constellation, the CMA 1-2 algorithm is *identical* to the Sato algorithm (note also that $\gamma = r_1$). The following table shows the coincidence of some of the above-mentioned BE algorithms for some specific input signals:

Constellation/Algorithm	DD	Sato	CMA 1-2	GSA
2-PAM	Identical	Identical	Identical	Identical
multi-PAM	-	Identical	Identical	Identical
4-QAM	Identical	-	-	Identical

In the case $p = 2$, the so-called Constant Modulus Algorithm (CMA) (or CMA 2-2) is obtained:

$$W_{k+1} = W_k + \mu X_k y_k (r_2 - |y_k|^2) . \quad (2.80)$$

The CMA (or the Godard) algorithm is considered to be among the most succesful algorithms of the Bussgang type. This is due to the following advantages (see [Jab92], [SGKC91] for some comparative studies):

- Often CMA succeeds in opening the initially closed channel eye.
- The CMA is more robust than other Bussgang algorithms to carrier phase offset. This is an important advantage of the Godard algorithm and is due to the fact that its criterion is based only on the amplitude of the received signal, i.e. it is blind to a phase rotation.
- The CMA often achieves a lower steady-state error w.r.t. other Bussgang algorithms.

The analysis of the Godard algorithm, as done in the original paper of Godard, gives the following results:

- When the channel is noiseless and the equalizer is infinite-length, the global minimum of the Godard cost function corresponds to a zero-ISI equalizer setting.
- The cost function in (2.76) is not a convex function of the equalizer taps. As a result of this, the algorithm's *initialization* is of critical importance since some initializations may lead the algorithm towards a local minimum of the cost function, which does not adequately open the channel eye.
- Two conditions are necessary for the transmitted constellation, namely, the two following equations must be satisfied:

$$E(a_i^2) = 0 , \quad (2.81)$$

and

$$E(|a_i|^4) < 2(E|a_i|^2)^2 . \quad (2.82)$$

Eq. (2.81) requires some kind of *symmetry* on the transmitted constellation, whereas eq. (2.82) some kind of *compactness* on the transmitted constellation. However, these two

conditions are often met by many constellations, especially as the transmitted samples are first *differentially encoded*¹³.

- Some practical guidelines in order to guarantee the equalizer's convergence towards the global minimum of the cost function are suggested in [God80]. However, there is no way to guarantee that a specific initialization will be correct in all cases.

The strongest result on the performance of the Godard algorithm (in T -spaced equalizers) was obtained by Shalvi and Weinstein [SW90]. By showing that the Godard criterion ($p = 2$) is a special case of their kurtosis function, they proved that the CMA 2-2 algorithm is globally convergent under the following assumptions:

- No additive noise is present
- The input distribution is sub-Gaussian

What was not explicited in [SW90] is that an extra assumption was needed for their proofs to be valid, namely:

- The equalizer has a doubly-infinite length

This remark was done by Tugnait in [Tug92], thus verifying the claims in [Din91a], [DKAJ91], [JLK92] e.t.c. about the existence of local sub-optimal minima of the Godard cost function in the case of FIR equalizers.

A way to explain the existence of local minima of the Godard cost function in the case of a finite equalizer length is the following. The extrema of the Godard cost function can be found by setting the gradient of the cost function equal to zero, i.e.:

$$E(X_k y_k |y_k|^{p-2} (|y_k|^p - r_p)) = 0 . \quad (2.83)$$

Eq. (2.83) is a system of N nonlinear equations w.r.t. N parameters (equalizer taps). Due to the nonlinear character of this system of equations, there is a proliferation of solutions to it, some of which correspond to minima. This is why local minima exist for the Godard algorithm! It must be noted at this point that the local minima in the Godard cost function exist even when the equalizer is adequately parameterized to be able to match exactly the inverse of the channel impulse response. Figure 2.12 shows the Godard cost function in the case of a linear AP(1) channel and a linear FIR(1) equalizer (perfect parameterization). Note the existence of two optimal global minima and of two local minima that do not completely open the channel eye.

- The Stop-and-Go algorithm for BE

¹³Differential coding is a technique that permits, at the loss of some dB of performance, to eliminate phase ambiguities in the received signal.

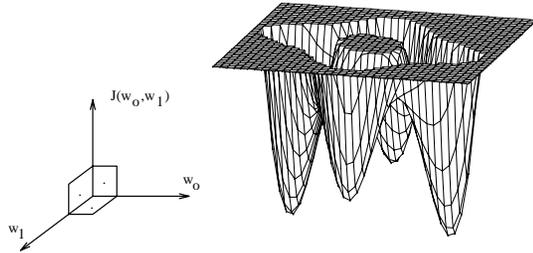


Figure 2.12: A Godard cost function for an AP(1) channel and a FIR(1) equalizer

The Stop-and-Go algorithm for BE was proposed by Picchi and Prati in 1987 [PP87] and is a hybrid scheme in the sense that it uses both the error functions of the DDA and the (generalized) Sato algorithm. We use the term error function in the same way it is used in classical system identification to denote the difference between the filter output and a “desired” response. The corresponding error functions for DDA and Generalized Sato are:

$$\begin{cases} e_k^{DD} = dec(y_k) - y_k \\ e_k^{Sato} = csign(y_k) - y_k \end{cases} \quad (2.84)$$

The principle of the Stop-and-Go algorithm is the following: the Sato algorithm has in general been proven to be able to open the channel eye more successfully than the DDA. However the DDA achieves faster convergence and a lower steady-state error once the eye is already opened. The Stop-and-Go adaptation rule is the following: if the error functions e_k^{DD} and e_k^{Sato} have the same complex *sign* at iteration k , then the DDA is considered reliable enough to do the adaptation, and therefore a DDA iteration is performed. On the other hand, if this is not the case ($csign(e_k^{DD}) \neq csign(e_k^{Sato})$), then there is no adaptation performed. In other words, the Sato error at each iteration is calculated only in order to give enough credit to the DD adaptation rule to be performed or not. The Stop-and-Go algorithm has therefore the following form:

$$\begin{aligned} W_{k+1}^R &= W_k^R - \mu_1 \left(f_k^R e_k^R X_k^R + f_k^I e_k^I X_k^I \right) \\ W_{k+1}^I &= W_k^I - \mu_2 \left(f_k^R e_k^R X_k^I + f_k^I e_k^I X_k^R \right) \end{aligned} \quad (2.85)$$

where $e_k = e_k^{DD}$, the superscripts R and I denote real and imaginary part of the corresponding quantities, respectively and f^J , $J = R$ or I is a flag function defined as:

$$f_k^J = \begin{cases} 1, & \text{if } sign(e^{DD^J}) = sign(e^{Sato^J}) \\ 0, & \text{otherwise} \end{cases} \quad (2.86)$$

The Stop-and-Go algorithm (SGA) has been reported to have a behaviour combining the advantages of both DD and Sato, i.e. it has DD’s steady-state error, has a convergence speed

faster than Sato’s and is capable of opening the channel eye in cases where DD cannot but Sato can.

Some novel Stop-and-Go-like algorithms have been introduced by Hatzinakos (see [Hat91], [Hat92]), namely the *Sign* Stop-and-Go algorithm, the *Godard* Stop-and-Go algorithm and the Stop-and-Go algorithm with 2 error functions (SG2). The philosophy of the sign stop-and-go algorithm is to use for the adaptation not the actual values of e^{DD} and e^{Sato} at each iteration but their respective sign functions, since it is only based on the information of the sign functions that the stop-and-go algorithm decides if it is going to perform an operation or not. The corresponding algorithm is therefore obtained if one replaces in (2.85) the quantities e^{DD^R} and e^{DD^I} by the quantities $sign(e^{DD^R})$ and $sign(e^{DD^I})$, respectively.

The Godard SG algorithm is the same as the SGA algorithm with the difference that now the Godard error function $e_k^{God} = y_k(|y_k|^2 - r_2)$ is used instead of e^{Sato} . Finally the SG2 algorithm is as the SGA algorithm with the difference that the flag function is now defined as:

$$f_k^J = \begin{cases} 1, & \text{if } sign(e^{DD^J}) = sign(e^{Sato^J}) = sign(e^{God^J}) \\ 0, & \text{otherwise} \end{cases} \quad (2.87)$$

All these algorithms have been shown in practice to outperform the DD algorithm for non-constant modulus constellations. However we are not aware of any complete theoretical analysis that explains this behaviour.

- Other algorithms

Several other algorithms for BE have been recently proposed. The CRIMNO algorithm [CNP92] proposes a criterion with memory nonlinearity (it imposes constraints not only on the modulus of each received sample but also on some correlation terms with previous samples). In [WK94] some other dual-mode type BE algorithms are proposed, whereas in [SW93] some HOS - based methods are proposed which do not impose restriction on the pdf of the transmitted sequence.

2.10 Principal aims of the thesis

In the remainder of the thesis we will propose several new methods for blind equalization and identification of linear channels. The motivation for the development of each method has to do with disadvantages of already existing schemes. Namely, our concern is focused on the following aspects of BE algorithms:

- Their low convergence speed
- Their steady-state error
- The problem of ill-convergence

- The impossibility of ZF equalization with FIR filters
- The need of Higher Order Statistics for Blind Equalization

The methods that will be presented propose some solutions to the above-mentioned problems.

Chapter 3

A new class of algorithms for constant-modulus blind equalization

IN this chapter we propose a new class of algorithms of the constant modulus type for blind equalization. The algorithms are derived by minimizing a deterministic criterion that imposes at each iteration a number of constraints on the equalizer. An efficient computational organization of the algorithm is proposed in order to allow for reduced computational complexity. The algorithms are analyzed both theoretically and by means of computer simulations. Two important advantages with respect to conventional CMA's is their increased convergence speed as well as their potential ability to escape from local minima of their cost function. The relation of this class of algorithms to classical adaptive filtering algorithms is investigated and as a result a so-called *separation principle* is established. This principle allows for the derivation of algorithms suitable for BE by using any classical adaptive filtering algorithm.

3.1 Introduction

We consider the blind equalization setting of figure 3.1. As we already mentioned, one of the most popular adaptive blind algorithms for equalization of linear channels in a QAM system is the Constant Modulus Algorithm, a special case of the Godard algorithms. The popularity of the algorithms of the constant modulus type is due to several advantages they offer, namely:

- their satisfactory performance in several cases
- their simplicity (low computational complexity, easy implementations)

- their ability to open blindly the channel eye not only for constant envelope but for any QAM constellation

The last advantage mentioned above, even though paradoxical (why should a constant modulus algorithm be useful in a non-constant modulus modulation system?) has been not only experimentally validated but also theoretically founded recently by Shalvi and Weinstein [SW90]. This is the main reason why we will focus on the development of algorithms of the *constant modulus* type.

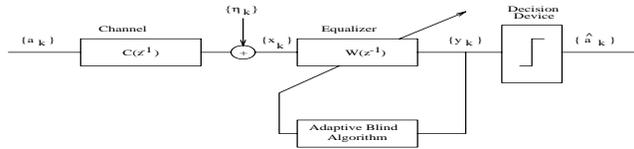


Figure 3.1: A typical blind equalization scheme

Besides the above-mentioned advantages, the CMA's exhibit also a number of drawbacks that have been extensively reported in recent literature over the last 15 years, namely:

- They have a low convergence speed, especially when the received signal is strongly coloured
- They are sensitive to their initialization, i.e. some initializations may lead to convergence around undesirable stationary points, causing the problem of ill-convergence.

The problem of low convergence speed is a typical one of stochastic gradient algorithms. It is well known for example that in the LMS algorithm (2.49) the speed of convergence for a given value of the stepsize parameter μ is sensitive to the colouring of the input signal (it decreases with the colouring of the input signal). For a given colouring of the input signal, the convergence speed increases with μ , but as in each iteration k the maximum stepsize for stable operation depends on the specific regressor X_k , one usually chooses μ small enough in order to guarantee stability at all iterations. An alternative to the LMS that overcomes this problem is the NLMS algorithm:

$$W_{k+1} = W_k + \frac{\mu}{\|X_k\|^2} X_k (\hat{a}_k - X_k^H W_k) . \quad (3.1)$$

The stability of (3.1) is guaranteed for all $0 < \bar{\mu} < 2$ and its maximum convergence speed is provided for $\bar{\mu} = 1$. As the CMA is also an algorithm of the stochastic gradient type, it is a natural reflection to investigate if its convergence speed can be increased by suitable *normalization*.

However, convergence speed is not the only performance index that dictates the normalization of constant modulus algorithms. One might expect that normalization could be also

beneficial w.r.t. to the problem of ill convergence of CMA's. Namely, the convergence of a stochastic gradient algorithm to an undesirable stationary point is the result of the interaction of three different factors:

- the existence of local undesirable minima of the corresponding cost function
- the algorithm's initialization
- the employed stepsize

Indeed, attacking each one of these three points could result in a remedy to the problem. In the first case a convex cost function should be constructed (such attempts have been recently reported in literature, e.g. [KD92] [BZA94])¹, in the second and third case practical guidelines for the algorithm's initialization (see e.g. [JL92]) or choice of stepsize should be given. In this case one does not try to influence the shape of the cost function, but rather the trajectory that the algorithm will follow on it (or close to it).

An interesting hint about the crucial role of the stepsize can be found in Mazo [Maz80], concerning the performance analysis of a decision directed equalizer in a multi-PAM system. Mazo showed that a DD algorithm in a PAM case will always converge (eventually) to its global minimum. However it is stated that "for μ small, we expect to wait a very long time for deviations of the required magnitude to occur, and our earlier assumption of getting trapped at an undesired minimum is, in this sense, justified". This implies that in DD equalization (which is a primitive form of BE), a big value for the stepsize can help the algorithm head towards its global minimum, essentially *escaping* from its local minima. We therefore expect that normalization will play a positive role in this respect as well.

Besides normalization, another feature that could be added to algorithms of the constant modulus type is *memory*. As already mentioned, all BE algorithms of the Bussgang type are memoryless, i.e. only the most recent sample of the equalizer output is used for the adaptation of the algorithm. One would expect a better performance if one takes into account a number of other past equalizer outputs as well. This can be translated to imposing a more severe *constraining* on the equalizer output than memoryless algorithms do. This constraining allows for more flexibility on the choice of the cost function² and is expected to improve also the convergence behaviour of the algorithm.

To sum up, we are interested in algorithms for BE, of the *constant modulus type*, employing an inherent *normalization* of their stepsize and imposing a more severe *constraining* on the equalizer's output. A first class of algorithms of this type is presented in the next section.

¹A construction of a convex cost function for constant-modulus BE is also proposed in chapter 6 of this thesis.

²An example is the recently proposed ([CNP92]) CRIMNO algorithm.

3.2 The new class of algorithms

3.2.1 Derivation

Consider first the following deterministic CM criterion:

$$\begin{aligned} \min_{W_{k+1}} \|W_{k+1} - W_k\|_2^2 \\ \text{subject to: } (|X_k^H W_{k+1}|^p - 1)^q = 0, \end{aligned} \quad (3.2)$$

where $\|W\|_2 = \sqrt{W^H W}$ for $W \in \mathbb{C}^N$ denotes the 2-norm of a complex vector. This is a deterministic criterion that imposes a CM constraint on the *a posteriori* equalizer output at *each* time instant, trying to keep the “next” equalizer setting W_{k+1} as close as possible to the previous one W_k in doing so. It is clear that as the number of parameters (N) to be adjusted is bigger than the number of constraints (1) set by the second equation of (3.2), this can be exactly satisfied, in which case W_{k+1} will obey the following equation:

$$|X_k^H W_{k+1}| = 1. \quad (3.3)$$

We now consider the minimum-norm criterion of the first line of (3.2). The deviation of the new equalizer setting W_{k+1} with respect to the previous one W_k can be decomposed as:

$$W_{k+1} - W_k = X_k v_k + \Omega_k. \quad (3.4)$$

The first term on the right-hand side of (3.4) represents the component of $W_{k+1} - W_k$ in the 1-dimensional subspace of \mathbb{C}^N spanned by X_k , and Ω_k the component belonging to the orthogonal complement of this subspace, of dimension $N - 1$:

$$\Omega_k^H X_k = 0. \quad (3.5)$$

Therefore, the first line of (3.2) can be written as:

$$\|X_k v_k\|_2^2 + \|\Omega_k\|_2^2 = \text{minimal}, \quad (3.6)$$

since $X_k v_k$ and Ω_k are orthogonal to each other. As Ω_k has no influence on equation (3.3) (due to the orthogonality expressed by (3.5)), the term $\|\Omega_k\|_2^2$ must be minimized independently, which gives:

$$W_{k+1} = W_k + X_k v_k. \quad (3.7)$$

Now combining (3.3) and (3.7) one gets:

$$X_k^H W_k + X_k^H X_k v_k = e_k, \quad (3.8)$$

where e_k is a complex number of modulus 1:

$$|e_k| = 1. \quad (3.9)$$

Eq. (3.8) gives when solved for v_k :

$$v_k = (X_k^H X_k)^{-1} (e_k - X_k^H W_k). \quad (3.10)$$

Reporting this value of v_k into (3.7), we get the following recursive formula for the equalizer vector W :

$$W_{k+1} = W_k + X_k (X_k^H X_k)^{-1} (e_k - X_k^H W_k). \quad (3.11)$$

Now among all the possible choices for the vector e_k , we must choose the one that results in the minimum 2-norm of $W_{k+1} - W_k$. As is shown in Appendix 3.A, this leads to the following choice for e_k :

$$e_k = \text{sign}(X_k^H W_k), \quad (3.12)$$

where the *sign* of a complex scalar is defined as:

$$\text{sign}(r e^{j\theta}) \equiv e^{j\theta}, \quad (3.13)$$

(if $r = 0$ we use in practice the convention $\text{sign}(0) \equiv 1$). A geometrical interpretation for e_k as given in (3.12) is that e_k is the projection of $X_k^H W_k$ on the unit circle in such a way so as to preserve its phase (see fig. 3.2).

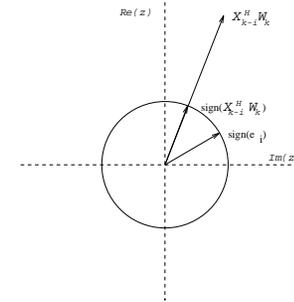


Figure 3.2: Choice of the optimal projection onto the unit circle

Substituting into (3.11), we obtain the following recursive formula that solves the minimization problem (3.2):

$$W_{k+1} = W_k + X_k (X_k^H X_k)^{-1} (\text{sign}(X_k^H W_k) - X_k^H W_k). \quad (3.14)$$

Notice that the above algorithm is derived by minimizing the criterion in (3.2) for any value of p and q , and therefore corresponds to a deterministic version of *any* CMA $p - q$ algorithm. We now consider the following alternative deterministic criterion:

$$\begin{aligned} \min_{W_{k+1}} \{ \|W_{k+1} - W_k\|_2^2 \} \\ \text{subject to: } \|\text{sign}(X_k^H W_k) - X_k^H W_{k+1}\|_2^2 = 0. \end{aligned} \quad (3.15)$$

As again we have the freedom to null exactly the scalar quantity $\text{sign}(X_k^H W_k) - X_k^H W_{k+1}$, we obtain again (3.3) and following exactly the same steps we arrive at the conclusion that the algorithm (3.14) also solves the problem (3.15). In order to allow more flexibility to this minimization criterion we shall now introduce a stepsize parameter $\bar{\mu}$ that will allow *control* of the deviation of the new equalizer setting W_{k+1} w.r.t. W_k , instead of *imposing* a minimum square-norm deviation. Using a result presented in [Slo92b] (see also Appendix 3.B), we obtain the following formulation: an exact minimization with respect to W_{k+1} of the deterministic function

$$\|\text{sign}(X_k^H W_k) - X_k^H W_{k+1}\|_{(X_k^H X_k)^{-1}}^2 + \left(\frac{1}{\bar{\mu}} - 1\right) \|W_{k+1} - W_k\|_2^2, \quad (3.16)$$

where $\|x\|_S^2 = x^H S x$, is provided at each iteration by the following algorithm:

$$W_{k+1} = W_k + \bar{\mu} X_k (X_k^H X_k)^{-1} (\text{sign}(X_k^H W_k) - X_k^H W_k). \quad (3.17)$$

In (3.16) the hard constraint of (3.15) has been replaced by a weighted minimization of the terms in (3.2). The first term expresses the information in measurements while the second term represents the prior information. The role of the parameter $\bar{\mu} \in (0, 1]$ is to control the relative weight of each term. Indeed, the problem of minimizing (3.16) w.r.t. W_{k+1} reduces (is equivalent) to the problem in (3.15) as $\bar{\mu} \rightarrow 1$.

Now consider a more severely-constrained minimization problem similar to the one described in (3.16) that imposes an ensemble of L (instead of 1) “*sign-type*” constraints on the equalizer vector W_{k+1} :

$$\|\text{sign}(\mathbf{X}_k^H W_k) - \mathbf{X}_k^H W_{k+1}\|_{P_k}^2 + \left(\frac{1}{\bar{\mu}} - 1\right) \|W_{k+1} - W_k\|_2^2, \quad (3.18)$$

where $P_k = \mathbf{X}_k^H \mathbf{X}_k$ and \mathbf{X}_k is a $N \times L$ data matrix defined as:

$$\mathbf{X}_k = [X_k \ X_{k-1} \ \dots \ X_{k-L+1}] = \begin{bmatrix} x_k^* & x_{k-1}^* & \dots & x_{k-L+1}^* \\ x_{k-1}^* & x_{k-2}^* & \dots & x_{k-L}^* \\ \vdots & \vdots & \ddots & \vdots \\ x_{k-N+1}^* & x_{k-N+2}^* & \dots & x_{k-N-L+2}^* \end{bmatrix}, \quad (3.19)$$

where superscript * denotes complex conjugate and the *sign* of a complex vector is defined as:

$$\text{sign}([x_0 \ x_1 \ \dots \ x_{N-1}]^T) = [\text{sign}(x_0) \ \text{sign}(x_1) \ \dots \ \text{sign}(x_{N-1})]^T. \quad (3.20)$$

According to the result presented in Appendix 3.B, the criterion (3.18) is minimized exactly at each iteration by the following algorithm:

$$W_{k+1} = W_k + \bar{\mu} \mathbf{X}_k P_k^{-1} (\text{sign}(\mathbf{X}_k^H W_k) - \mathbf{X}_k^H W_k). \quad (3.21)$$

Eq. (3.21) is the proposed class of algorithms that we will study in the sequel.

3.2.2 Discussion

Eq. (3.21) describes a new parametric class of algorithms for BE of CM signals. At each iteration, any algorithm of this class sets to zero the deterministic criterion (3.18). The two adjustable parameters are the data window length L and the stepsize $\bar{\mu}$. L is an integer that expresses the number of “CM constraints” imposed on the next equalizer setting at each iteration and may vary typically from 1 to N . Different values of L correspond to different members of the class of algorithms. The imposition of L constraints results in prewhitening the received data by the inverse of an $L \times L$ sample covariance matrix (P_k). As one can see from eq. (3.19) this matrix is formed by constructing first a $N \times L$ data matrix (\mathbf{X}_k), which corresponds to passing the data through a *rectangular sliding window*. $\bar{\mu}$ is a real scalar that controls the deviation of the new equalizer setting w.r.t. the previous one. Strictly speaking, only when $\bar{\mu} = 1$ do we impose a set of L constraints on W_{k+1} , whereas in the other cases we rather minimize a weighted deterministic function (3.18) that contains two additive terms. Two major features of the algorithms expressed through (3.21) with respect to the classical Godard or CMA BE algorithms are hence:

- (3.21) results from the exact minimization at each iteration of a deterministic criterion, and not by using a stochastic gradient technique in order to minimize a stochastic cost function.
- the algorithms (3.21) employ CM-type requirements that do not involve (if $L > 1$) only the most recent regression vector of data X_k but some previous regressors (of number $L - 1$) as well. In a way this can be seen as adding *memory* to the algorithm.

The deterministic aspect of the proposed algorithms offers not only important insight to the algorithm’s analysis, but provides also the very important feature of *normalization*. As it will turn out, the algorithms of this class are stable for all values $0 < \bar{\mu} < 2$ (the fastest convergence corresponding to the choice $\bar{\mu} = 1$), which provides an *a priori* known (normalized, independent of the actual regressors used at each iteration) range for stability (in contrast to stochastic-gradient algorithms where the stepsize’s region for stability is not known *a priori* and heavily depends on the input data’s colouring). The normalization not only facilitates the implementation of the algorithm in the sense that the choice of the stepsize is done very easily and independently of the received signal, but also, as it will be shown in the sequel, may have a positive impact in avoiding the problem of ill-convergence often observed in BE algorithms of the Godard type. On the other hand, the “memory” aspect of the proposed algorithms is expected to have a beneficial impact on the algorithms’ convergence speed, namely we expect that due to the extra constraining, the convergence speed will increase as L increases. Based on the two aforementioned features of normalization and constraining, we call the class of algorithms (3.21) *Normalized Sliding-Window Constant Modulus Algorithms*: NSWCMA’s.

3.2.3 Criterion interpretation

The criterion in (3.18) is a deterministic function of W_{k+1} that contains two additive quadratic terms. The first term represents the constant modulus requirement and the second one the minimum deviation (in a square-norm sense) requirement of W_{k+1} w.r.t. W_k . The relative importance of the two terms is determined by the value of the scalar parameter $\bar{\mu}$. The first term becomes less dominant in favour of the second term as $\bar{\mu}$ moves away from 1 and vice versa when $\bar{\mu}$ approaches 1. When $\bar{\mu}$ equals 1 the criterion (3.18) is equivalent to the following criterion:

$$\min_{W_{k+1}} \|W_{k+1} - W_k\|_2^2 \quad (3.22)$$

$$\text{subject to: } \|\text{sign}(\mathbf{X}_k^H W_k) - \mathbf{X}_k^H W_{k+1}\|_2^2 = 0 .$$

An interesting geometrical interpretation of this criterion is the following: consider the following vector of *a posteriori* outputs:

$$\mathcal{E}_k = \mathbf{X}_k^H W_{k+1} . \quad (3.23)$$

Substituting the expression for W_{k+1} given in (3.21) in (3.23) we obtain the following expression:

$$\mathcal{E}_k = \bar{\mu} \text{sign}(\mathbf{X}_k^H W_k) + (1 - \bar{\mu}) \mathbf{X}_k^H W_k , \quad (3.24)$$

which is a convex function for $\bar{\mu}$. As shown in fig. 3.3 each component of \mathcal{E}_k is a projection of the corresponding component of $\mathbf{X}_k^H W_k$ at some point on the straight line linking it to the origin. The actual position of the projection on this line is determined by $\bar{\mu}$. As $\bar{\mu}$ approaches 1 this position approaches the unit circle and gets on the unit circle when $\bar{\mu} = 1$. The fact that this projection is always on the same line shows that the criterion doesn't attempt to change the phase of the received signal. This is a typical characteristic of all Godard algorithms that only impose constraints on the *modulus* and not on the *phase* of the received signal.

3.2.4 Projection interpretation

As will be made clear later on, the NSWCMA may be considered to be a blind equalization-counterpart of the APA algorithm. The APA algorithm ([OU84]) was first proposed by Ozeki and Umeda in 1984. Its derivation based on a deterministic criterion was given by Slock in [Slo92b], where a block version is also derived and given the name of BUCFTF algorithm.

The following explanation of the projection mechanism that is performed in the APA will help understand better the NSWCMA as well. The APA update formula is given as

$$W_{k+1} = W_k + \bar{\mu} \mathbf{X}_k^H P_k^{-1} (D_k - \mathbf{X}_k^H W_k) , \quad (3.25)$$

where $D_k = [d_k \ d_{k-1} \ \dots \ d_{k-L+1}]^T$ contains the desired signal's elements at time instants $k, k-1, \dots, k-L+1$. The behaviour of the algorithm (3.25) may be described by the

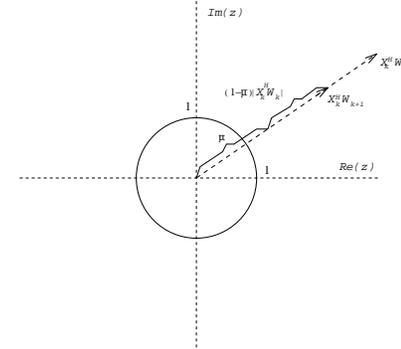


Figure 3.3: A geometrical interpretation of the minimized criterion

following projection interpretation: consider first the case $\bar{\mu} = 1$. Then W_{k+1} will be such as to null L consecutive *a posteriori* errors :

$$D_k - \mathbf{X}_k^H W_{k+1} = \mathbf{0} . \quad (3.26)$$

We denote by \mathcal{Q} the L -dimensional subspace of \mathcal{C}^N spanned by the L column vectors of the data matrix \mathbf{X}_k and by \mathcal{Q}^\perp the orthogonal complement of \mathcal{Q} , of dimension $N - L$. (3.26) is a linear system of L equations in N unknowns ($L < N$) and therefore has an infinity of solutions. We denote by W'_{k+1} the one lying on \mathcal{Q} (this is the minimum-norm solution of eq. (3.26)). Then all the other solutions will have one component on \mathcal{Q} equal to W'_{k+1} and then another component on \mathcal{Q}^\perp . The constraint of minimizing $\|W_{k+1} - W_k\|^2$ results in choosing among all these solutions the one that is closer to W_k in a square-norm sense. A schematic representation of this geometrical interpretation may be found in fig 3.4. As can be seen in the figure, W_{k+1} is obtained by an orthogonal projection of W_k to the subspace $\mathcal{Q}^\perp + W'_{k+1}$. The name "affine projection algorithm" in fact reflects the fact that this is an affine space w.r.t. \mathcal{Q}^\perp ³.

In the case $L = N$ the set of equations (3.26) has a unique solution which coincides with the classical least-square solution often sought in adaptive filtering. In this case \mathcal{Q} coincides with \mathcal{C}^N and the restriction for $\|W_{k+1} - W_k\|^2$ has no impact on the choice of W_k (we remind that $\bar{\mu} = 1$). This means that when $L = N$ and $\bar{\mu} = 1$ the algorithm will converge to the optimal equalizer only after 1 iteration (of course we have supposed here that no additive noise corrupts the data or the desired signal).

The situation becomes somewhat more complicated when $\bar{\mu} \neq 1$. We denote by J_k the

³By affine space we mean a space that has been shifted in the direction of a constant vector.

deterministic cost function at time instant k :

$$J_k = (D_k - \mathbf{X}_k^H W_{k+1})^H (\mathbf{X}_k^H \mathbf{X}_k)^{-1} (D_k - \mathbf{X}_k^H W_{k+1}) - \left(\frac{1}{\bar{\mu}} - 1\right) (W_{k+1} - W_k)^H (W_{k+1} - W_k) \quad (3.27)$$

Now J_k may be written as:

$$\begin{aligned} J_k &= \begin{bmatrix} D_k - \mathbf{X}_k^H W_{k+1} \\ W_k - W_{k+1} \end{bmatrix}^H \begin{bmatrix} (\mathbf{X}_k^H \mathbf{X}_k)^{-1} & \mathbf{0}_{L \times N} \\ \mathbf{0}_{N \times L} & \left(\frac{1}{\bar{\mu}} - 1\right) I_N \end{bmatrix} \begin{bmatrix} D_k - \mathbf{X}_k^H W_{k+1} \\ W_k - W_{k+1} \end{bmatrix} = \\ &= \begin{bmatrix} (\mathbf{X}_k^H \mathbf{X}_k)^{-\frac{1}{2}} D_k - (\mathbf{X}_k^H \mathbf{X}_k)^{-\frac{1}{2}} \mathbf{X}_k^H W_{k+1} \\ \sqrt{\frac{1}{\bar{\mu}} - 1} W_k - \sqrt{\frac{1}{\bar{\mu}} - 1} I_N W_{k+1} \end{bmatrix}^H \begin{bmatrix} (\mathbf{X}_k^H \mathbf{X}_k)^{-\frac{1}{2}} D_k - (\mathbf{X}_k^H \mathbf{X}_k)^{-\frac{1}{2}} \mathbf{X}_k^H W_{k+1} \\ \sqrt{\frac{1}{\bar{\mu}} - 1} W_k - \sqrt{\frac{1}{\bar{\mu}} - 1} I_N W_{k+1} \end{bmatrix} = \\ &= \left\| \begin{bmatrix} (\mathbf{X}_k^H \mathbf{X}_k)^{-\frac{1}{2}} D_k \\ \sqrt{\frac{1}{\bar{\mu}} - 1} W_k \end{bmatrix} - \begin{bmatrix} (\mathbf{X}_k^H \mathbf{X}_k)^{-\frac{1}{2}} \mathbf{X}_k^H \\ \sqrt{\frac{1}{\bar{\mu}} - 1} I_N \end{bmatrix} W_{k+1} \right\|^2 \quad (3.28) \end{aligned}$$

Eq. (3.28) shows the kind of projection that is performed in the case $\bar{\mu} \neq 1$. In this case the desired vector role is played by the vector $\left[(\mathbf{X}_k^H \mathbf{X}_k)^{-\frac{1}{2}} D_k \sqrt{\frac{1}{\bar{\mu}} - 1} W_k \right]^T$ of length $N + L$ and W_{k+1} is projected on the space spanned by the columns of the $(N + L) \times N$ matrix $\begin{bmatrix} (\mathbf{X}_k^H \mathbf{X}_k)^{-\frac{1}{2}} \mathbf{X}_k^H \\ \sqrt{\frac{1}{\bar{\mu}} - 1} I_N \end{bmatrix}$.

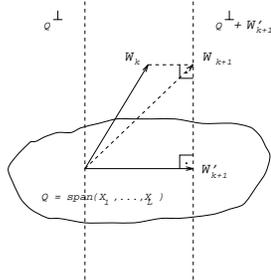


Figure 3.4: A projection interpretation of the APA algorithm

3.2.5 Relation to other algorithms

The relation of the NSWCMA algorithm to other algorithms is two-fold: when one chooses $L = 1$ which corresponds to a zero-memory criterion, the following algorithm results:

$$W_{k+1} = W_k + \bar{\mu} X_k P_k^{-1} (\text{sign}(X_k^H W_k) - X_k^H W_k) \quad (3.29)$$

The above algorithm actually corresponds to a constant-modulus counterpart (as explained in the previous subsection) of the NLMS algorithm. In fact, this algorithm has been derived in [HD92a] for the special case $\bar{\mu} = 1$ by nulling at each iteration the *a posteriori* error ϵ_k of the CMA 2-2 algorithm defined as:

$$\epsilon_k = |X_k^H W_{k+1} - 1|^2 - 1 \quad (3.30)$$

and has been given there the name NCMA (Normalized Constant Modulus Algorithm). We have derived the NCMA based on a deterministic CM criterion that incorporates also the stepsize $\bar{\mu}$ and shows its role (figure 3.3). So the NCMA is the first member ($L = 1$) of the class of NSWCMA algorithms.

On the other hand, as already mentioned, the NSWCMA is very similar to the APA algorithm for classical adaptive filtering. Namely, the NSWCMA is obtained if one chooses in the update formula (3.25) of the APA D_k as follows:

$$D_k = \text{sign}(X_k^H W_k) \quad (3.31)$$

Therefore, the NSWCMA may be seen as the BE counterpart of the APA. A further discussion on this issue is given in the next paragraph.

3.3 The separation principle

As it was shown in section 3.2, the NSWCMA was derived based on a constant modulus criterion⁴, and not by simply modifying the APA: the *sign* function appearing in the desired vector of the algorithm was not chosen empirically but was found to be the optimal choice that satisfied our CM criterion (3.2). We therefore notice the following coincidence:

Statement: An algorithm of the classical adaptive filtering form (the only difference being in the choice of the vector of desired samples) has been derived by minimizing exactly a CM criterion.

This means that if we had simply taken the NLMS algorithm and replaced the desired sample d_k by the scalar $\text{sign}(X_k^H W_k)$, the resulting algorithm would correspond exactly to the minimization of a Godard-type CM criterion! Even though this will not be always the case for any adaptive algorithm (we saw for example that for the NSWCMA ($L > 1$) the corresponding criterion is a *sign*-like instead of a CMA-like one), it provides enough motivation to use it as an easy way to obtain algorithms suitable for BE of the CM type, from any classical adaptive filtering algorithm. We formulate this idea as follows:

⁴Strictly speaking, only in the case $L = 1$ the NSWCMA is derived by a CM criterion, while for $L > 1$ it is derived by a *sign*-type criterion.

A separation principle for BE: An adaptive algorithm of the CMA-type suitable for BE can be derived by taking any classical adaptive filtering algorithm and replacing its desired signal $\{D_k\}$ by $\{\text{sign}(\mathbf{X}_k^H W_k)\}$.

We call this procedure a *separation principle* because in some cases (e.g. NLMS) its application corresponds exactly to a CM criterion, whereas in other cases it is suboptimal. The projection on the unit circle that has to be performed in order to obtain the vector of desired samples is the one shown in figure 3.2.

LMS CMA 1-2	$W_{k+1} = W_k + \mu X_k(d_k - y_k)$ $W_{k+1} = W_k + \mu X_k(\frac{y_k}{ y_k } - y_k)$
NLMS NCMA	$W_{k+1} = W_k + \frac{\mu}{\ X_k\ _2} X_k(d_k - y_k)$ $W_{k+1} = W_k + \frac{\mu}{\ X_k\ _2} X_k(\frac{y_k}{ y_k } - y_k)$
APA NSWCMA	$W_{k+1} = W_k + \bar{\mu} \mathbf{X}_k P_k^{-1} (D_k - \mathbf{X}_k^H W_k)$ $W_{k+1} = W_k + \bar{\mu} \mathbf{X}_k P_k^{-1} (\text{sign}(\mathbf{X}_k^H W_k) - \mathbf{X}_k^H W_k)$
RLS 'RLS-CMA'	$W_{k+1} = W_k + P_k^{-1} X_k(d_k - y_k)$ $W_{k+1} = W_k + P_k^{-1} X_k(\frac{y_k}{ y_k } - y_k)$

Table 3.1: CM-type BE algorithms derived by applying the separation principle

By using the separation principle it is easy to obtain many different algorithms for BE that reflect the performance characteristics of their classical adaptive filtering counterparts. An interesting algorithm (already proposed in [Age86]) is the RLS-CMA, i.e. the algorithm derived by applying the separation principle to the RLS algorithm. Of course, all fast algorithmic organizations for the RLS (e.g. (S)FTF, Fast Lattice, Fast QR) can also be used leading to fast CMA-like algorithms ((S)FTF-CMA, Fast Lattice-CMA (see also [GRS87]), Fast QR-CMA) e.t.c. The same stands for the Fast Newton algorithms [MT93] that can lead to the Fast Newton CMA algorithm.

Table 3.1 shows how some CMA-type algorithms are derived from their corresponding classical counterparts. Our separation principle has already been mentioned in [Lam94], where a blind SFTF filter was derived, based on a Newton update.

3.4 Computational organization

Taking into account the close-to-Toeplitz structure of the sample covariance matrix P_k , the calculation of the next equalizer setting W_{k+1} from the previous one W_k may be organized in a computationally efficient way as follows:

1. Compute the $L \times 1$ *a priori* error vector (NL multiplications):

$$E_k = \text{sign}(\mathbf{X}_k^H W_k) - \mathbf{X}_k^H W_k \quad . \quad (3.32)$$

2. Update a sufficient description of P_k from that of P_{k-1} (close-to-Toeplitz matrix) ($O(L)$ multiplications).

3. Solve the following linear system of equations by use of the *generalized Levinson* algorithm ($5.5L^2 + O(L)$ multiplications):

$$P_k H_k = \bar{\mu} E_k \quad . \quad (3.33)$$

4. Update the equalizer vector as follows (NL multiplications):

$$W_{k+1} = W_k + \mathbf{X}_k H_k \quad . \quad (3.34)$$

A further reduction in complexity may be achieved if the algorithm is implemented in a Block-updating form, i.e:

$$W_{k+L} = W_k + \bar{\mu} \mathbf{X}_k P_k^{-1} (\text{sign}(\mathbf{X}_k^H W_k) - \mathbf{X}_k^H W_k) \quad . \quad (3.35)$$

The algorithm (3.35) will have a complexity L times smaller than (3.21) but also a lower convergence speed as L times fewer updates are carried out. It is possible to use FFT techniques to perform the computations in (3.32) and (3.34) in $O(N \log_2 L)$ operations. Also the computations in (3.33) can be reduced to $O(L(\log_2 L)^2)$ operations by using the displacement representation of P_k^{-1} (this necessitates the use of the FTF algorithm for propagating the generators of P_k^{-1} in $O(L)$ operations). These alternative computations are interesting when L (and hence N) is large. Finally, if we introduce an approximation in NSWCMA corresponding to taking

$$E_k = [\text{sign}(X_k^H W_k) \cdots \text{sign}(X_{k-L+1}^H W_{k-L+1})]^T \quad , \quad (3.36)$$

the USWC FTF algorithm of [Slo92c] may be used to run NSWCMA in $12N + O(L)$ operations/sample.

A more interesting algorithmic organization that provides a significant reduction in computational complexity can be obtained if we apply the FAP algorithm. This algorithm has been proposed in [Gay93] as a fast alternative to the APA. It is worthwhile to describe the FAP algorithm, especially as we will introduce a modified version of it in chapter 4.

3.4.1 The FAP algorithm

The FAP implements the APA algorithm in a computationally efficient way:

$$W_{k+1} = W_k + \bar{\mu} \mathbf{X}_k l_k \quad (3.37)$$

where l_k is the $L \times 1$ vector defined as:

$$l_k = P_k^{-1} E_k = \begin{bmatrix} l_{0,k} \\ \vdots \\ l_{L-1,k} \end{bmatrix}, \quad (3.38)$$

and E_k is the $L \times 1$ *a priori* error vector at iteration k :

$$E_k = D_k - \mathbf{X}_k^H W_k. \quad (3.39)$$

The key formula that will permit for a fast algorithm is derived by developing the updating term in (3.37) as follows:

$$\begin{aligned} W_{k+1} &= W_k + \bar{\mu} \mathbf{X}_k l_k \\ &= W_1 + \bar{\mu} \sum_{i=0}^{k-1} \mathbf{X}_{k-i} l_{k-i} \\ &= W_1 + \bar{\mu} \sum_{i=0}^{k-1} [X_{k-i} \cdots X_{k-i-L+1}] \begin{bmatrix} l_{0,k-i} \\ \vdots \\ l_{L-1,k-i} \end{bmatrix} \\ &= W_1 + \bar{\mu} \sum_{i=0}^{k-1} \sum_{j=0}^{L-1} X_{k-i-j} l_{j,k-i}. \end{aligned}$$

Using the convention that $X_0 = X_{-1} = \dots = X_{-L+2} = \mathbf{0}$, the above relation gives:

$$\begin{aligned} W_{k+1} &= W_1 + \bar{\mu} \sum_{m=L}^{k-1} X_{k-m} \sum_{j=0}^{L-1} l_{j,k-m+j} + \bar{\mu} \sum_{m=0}^{L-1} X_{k-m} \sum_{j=0}^m l_{j,k-m+j} \\ &= W_1 + \bar{\mu} \sum_{m=L}^{k-1} X_{k-m} \sum_{j=0}^{L-1} l_{j,k-m+j} + \bar{\mu} [X_k \cdots X_{k-L+1}] \begin{bmatrix} l_{0,k} \\ l_{0,k-1} + l_{1,k} \\ \vdots \\ l_{0,k-L+1} + \cdots + l_{L-1,k} \end{bmatrix}. \end{aligned}$$

Defining the quantities \widehat{W}_k and F_k as:

$$\widehat{W}_k = W_1 + \bar{\mu} \sum_{m=L}^{k-1} X_{k-m} \sum_{j=0}^{L-1} l_{j,k-m+j}, \quad (3.40)$$

(a $N \times 1$ vector) and

$$F_k = \begin{bmatrix} l_{0,k} \\ l_{0,k-1} + l_{1,k} \\ \vdots \\ l_{0,k-L+1} + \cdots + l_{L-1,k} \end{bmatrix} = \begin{bmatrix} F_{0,k} \\ F_{1,k} \\ \cdots \\ F_{L-1,k} \end{bmatrix}, \quad (3.41)$$

(an $L \times 1$ vector), respectively, the above expression takes the form:

$$W_{k+1} = \widehat{W}_k + \bar{\mu} \mathbf{X}_k F_k. \quad (3.42)$$

\widehat{W}_k is a key quantity for the development of FAP and in fact the updating of W_k will be replaced by the updating of \widehat{W}_k . From the definition of \widehat{W}_k in (3.40) it results that:

$$\widehat{W}_{k+1} = \widehat{W}_k + \bar{\mu} X_{k-L+1} \sum_{j=0}^{L-1} l_{j,k-L+1+j},$$

and therefore the recurrence for \widehat{W}_k can be written as:

$$\widehat{W}_{k+1} = \widehat{W}_k + \bar{\mu} X_{k-L+1} F_{L-1,k}. \quad (3.43)$$

From (3.43) and (3.42) we find the relationship linking W_k and \widehat{W}_k at the same time instant:

$$W_{k+1} = \widehat{W}_{k+1} + \bar{\mu} \mathbf{X}_k \overline{F}_k, \quad (3.44)$$

where $\overline{\mathbf{X}}_k$ and \overline{F}_k are defined as:

$$\mathbf{X}_k = [\overline{\mathbf{X}}_k X_{k-L+1}], \quad (3.45)$$

and

$$F_k = \begin{bmatrix} \overline{F}_k \\ F_{L-1,k} \end{bmatrix}, \quad (3.46)$$

respectively. Eq. (3.43) is the most important formula for the derivation of FAP since it allows for a filter update that only needs *one* regression vector (namely, X_{k-L+1} for the updating of \widehat{W}_k), instead of the L regression vectors needed for the updating of W_k in (3.37). In this way a gain of one order of magnitude ($O(L)$) in complexity is attained. So what will be updated in each iteration of the algorithm is \widehat{W}_k and W_k will be calculated (if desired) via (3.44) only after convergence. So (3.43) is the key identity of FAP.

In order to be computationally efficient, one needs to update also F_k . Noting that:

$$F_k = \begin{bmatrix} l_{0,k} \\ l_{1,k} \\ \vdots \\ l_{L-1,k} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ l_{0,k-1} \\ \vdots \\ l_{0,k-L+1} + \cdots + l_{L-2,k-1} \end{bmatrix},$$

the updating formula for F_k is:

$$F_k = \begin{bmatrix} \mathbf{0} \\ \overline{F}_{k-1} \end{bmatrix} + l_k. \quad (3.47)$$

Now l_k will be updated using the *displacement* structure of P_k (remember that $l_k = P^{-1} E_k$). We first have to define the quantities \overline{P}_k and \hat{P}_k as the following upper left and lower right partitions of P_k , respectively:

$$P_k = \left[\begin{array}{c|c} \overline{P}_k & \begin{matrix} * \\ \vdots \\ * \end{matrix} \\ \hline * & * \end{array} \right] = \left[\begin{array}{c|c} * & * \cdots * \\ \hline * & \tilde{P}_k \end{array} \right] .$$

The corresponding partitions for the inverse matrices can be written as:

$$P_k^{-1} = \left[\begin{array}{c|c} \mathbf{0} & \mathbf{0} \cdots \mathbf{0} \\ \hline \mathbf{0} & \tilde{P}_k^{-1} \\ \vdots & \\ \mathbf{0} & \end{array} \right] + \frac{1}{E_{a,k}} a_k a_k^H = \left[\begin{array}{c|c} \overline{P}_k^{-1} & \begin{matrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \end{matrix} \\ \hline \mathbf{0} & \mathbf{0} \end{array} \right] + \frac{1}{E_{b,k}} b_k b_k^H , \quad (3.48)$$

where $a_k, b_k, E_{\alpha,k}, E_{\beta,k}$ denote the forward and backward linear prediction filters and the corresponding prediction error energies for the sample covariance matrix P_k , respectively.

If we now define the following quantities:

$$\begin{aligned} \tilde{l}_k &= \tilde{P}_k^{-1} \tilde{E}_k \\ \tilde{l}_k &= \overline{P}_k^{-1} \overline{E}_k \end{aligned} , \quad (3.49)$$

where:

$$E_k = \begin{bmatrix} E_{0,k} \\ \tilde{E}_k \end{bmatrix} = \begin{bmatrix} \overline{E}_k \\ E_{L-1,k} \end{bmatrix} , \quad (3.50)$$

then l_k can be updated efficiently via the following two identities:

$$l_k = \begin{bmatrix} \mathbf{0} \\ \tilde{l}_k \end{bmatrix} + \frac{1}{E_{a,k}} a_k a_k^H E_k , \quad (3.51)$$

and:

$$\begin{bmatrix} \tilde{l}_k \\ \mathbf{0} \end{bmatrix} = l_k - \frac{1}{E_{b,k}} b_k b_k^H E_k , \quad (3.52)$$

which are straightforwardly derived from (3.48) and (3.49). The forward and backward predictors a_k and b_k , respectively, can be updated using the prediction part of the Sliding Window Covariance Fast Transversal Filter algorithm. This will require $10L$ flops per iteration, or $11L \sim 12L$ flops for a Stabilized Fast Transversal Filter algorithm [SK91].

The updating of E_k is done efficiently by exploiting a specific relation that exists between a priori and a posteriori error vectors. Whereas the a priori error vector at iteration k is defined as:

$$E_k = D_k - \mathbf{X}_k^H W_k = \begin{bmatrix} d_k - X_k^H W_k \\ \overline{D}_{k-1} - \overline{\mathbf{X}}_{k-1}^H W_k \end{bmatrix} , \quad (3.53)$$

the a posteriori error vector at time instant $k-1$ is defined as:

$$\mathcal{E}_{k-1} = D_{k-1} - \mathbf{X}_{k-1}^H W_k = \begin{bmatrix} \overline{D}_{k-1} - \overline{\mathbf{X}}_{k-1}^H W_k \\ d_{k-L} - X_{k-L}^H W_k \end{bmatrix} = \begin{bmatrix} \overline{\mathcal{E}}_{k-1} \\ \mathcal{E}_{L-1,k-1} \end{bmatrix} . \quad (3.54)$$

From (3.53) and (3.54) it results that:

$$\tilde{E}_k = \overline{\mathcal{E}}_{k-1} . \quad (3.55)$$

Writing down the expression for the a priori error at iteration $k-1$ we find that:

$$\begin{aligned} E_{k-1} &= D_{k-1} - \mathbf{X}_{k-1}^H W_{k-1} = D_{k-1} - \mathbf{X}_{k-1}^H (W_k - \bar{\mu} \mathbf{X}_{k-1} P_{k-1}^{-1} E_{k-1}) \\ \mathcal{E}_{k-1} &= D_{k-1} - \mathbf{X}_{k-1}^H W_k = D_{k-1} - \mathbf{X}_{k-1}^H W_{k-1} - \bar{\mu} \mathbf{X}_{k-1}^H \mathbf{X}_{k-1} P_{k-1}^{-1} E_{k-1} = (1 - \bar{\mu}) E_{k-1} , \end{aligned}$$

and therefore the relationship between a priori and a posteriori errors is:

$$\mathcal{E}_k = (1 - \bar{\mu}) E_k . \quad (3.56)$$

Now combining (3.53), (3.54) and (3.56), we obtain the following equation for the updating of E_k :

$$E_k = \begin{bmatrix} E_{0,k} \\ \overline{\mathcal{E}}_{k-1} \end{bmatrix} = \begin{bmatrix} E_{0,k} \\ (1 - \bar{\mu}) \overline{E}_{k-1} \end{bmatrix} . \quad (3.57)$$

A by-product of (3.57) is that it allows for a passage from \tilde{l}_k to \tilde{l}_k . This is done by noting that:

$$\tilde{P}_{k+1} = \overline{P}_k , \quad (3.58)$$

which is due to the specific structure of P_k . Then one can easily see that:

$$\tilde{l}_{k+1} = \overline{P}_k^{-1} \tilde{E}_{k+1} = \overline{P}_k^{-1} (1 - \bar{\mu}) \overline{E}_k ,$$

and therefore the sought relation is:

$$\tilde{l}_{k+1} = (1 - \bar{\mu}) \tilde{l}_k . \quad (3.59)$$

There is also an efficient way to compute the scalar $E_{0,k}$. Using the definition of $E_{0,k}$ and (3.44) we have:

$$\begin{aligned} E_{0,k} &= d_k - X_k^H W_k = d_k - X_k^H (\widehat{W}_k + \bar{\mu} \overline{\mathbf{X}}_{k-1} \overline{F}_{k-1}) , \\ E_{0,k} &= \widehat{E}_{0,k} - \bar{\mu} X_k^H \overline{\mathbf{X}}_{k-1} \overline{F}_{k-1} , \end{aligned}$$

where $\widehat{E}_{0,k}$ is defined by:

$$\widehat{E}_{0,k} = d_k - X_k^H \widehat{W}_k . \quad (3.60)$$

Therefore the updating of $E_{0,k}$ is done through:

$$\begin{cases} \widehat{E}_{0,k} = d_k - X_k^H \widehat{W}_k \\ E_{0,k} = \widehat{E}_{0,k} - \bar{\mu} \tau_k \overline{F}_{k-1} , \end{cases} \quad (3.61)$$

where τ_k is defined as:

$$\tau_k = X_k^H \overline{\mathbf{X}}_{k-1} . \quad (3.62)$$

In order to update the quantity τ_k note that it can be written as:

$$\tau_k = x_k \tilde{\mathcal{X}}_k^H + \cdots + x_{k-N+1} \tilde{\mathcal{X}}_{k-N+1}^H , \quad (3.63)$$

where $\tilde{\mathcal{X}}_k$ is the vector consisting of the lower-most $L - 1$ elements of the following vector:

$$\mathcal{X}_k = \begin{bmatrix} x_k \\ \vdots \\ x_{k-L+1} \end{bmatrix} = \begin{bmatrix} x_k \\ \tilde{\mathcal{X}}_k \end{bmatrix}. \quad (3.64)$$

Therefore the update equation for τ_k is:

$$\tau_k = \tau_{k-1} + x_k \tilde{\mathcal{X}}_k^H - x_{k-N} \tilde{\mathcal{X}}_{k-N}^H. \quad (3.65)$$

The identities derived above that constitute the FAP algorithm are now summed up in table 3.2.

The FAP algorithm		×
Initialization: $a_0 = [\mathbf{1} \mathbf{0}_{(L-1) \times 1}]^T, b_0 = [\mathbf{0}_{(L-1) \times 1} \mathbf{1}]^T$, 0: $E_{a,0} = E_{b,0} = \delta$ (a very small number) $\tau_0 = X_0^H \bar{\mathbf{X}}_{-1}, \bar{F}_0 = \mathbf{0}, \bar{W}_0 = \mathbf{0}, \bar{l}_1 = \mathbf{0}, \bar{E}_0 = \mathbf{0}$		
1:	Use a (Stabilized) SWC FTF (prediction part) to update $a_k, b_k, E_{a,k}, E_{b,k}$	$10L \sim 12L$
2:	$\tau_k = \tau_{k-1} + x_k \tilde{\mathcal{X}}_k^H - x_{k-N} \tilde{\mathcal{X}}_{k-N}^H$	$2L$
3:	$\bar{E}_{0,k} = d_k - X_k^H \bar{W}_k$	N
4:	$E_{0,k} = \bar{E}_{0,k} - \mu \tau_k \bar{F}_{k-1}$	L
5:	$E_k = \begin{bmatrix} E_{0,k} \\ (1 - \bar{\mu}) \bar{E}_{k-1} \end{bmatrix}$	L
6:	$l_k = \begin{bmatrix} \mathbf{0} \\ \bar{l}_k \end{bmatrix} + \frac{1}{E_{a,k}} a_k a_k^H E_k$	$2L$
7:	$\begin{bmatrix} \bar{l}_k \\ \mathbf{0} \end{bmatrix} = l_k - \frac{1}{E_{b,k}} b_k b_k^H E_k$	$2L$
8:	$F_k = \begin{bmatrix} \mathbf{0} \\ \bar{F}_{k-1} \end{bmatrix} + l_k$	L
9:	$\bar{W}_{k+1} = \bar{W}_k + \mu X_{k-L+1} F_{L-1,k}$	N
10:	$\bar{l}_{k+1} = (1 - \bar{\mu}) \bar{l}_k$	L

Table 3.2: The FAP algorithm

3.5 Performance analysis

In this section we study the performance of the NSWCMAs in terms of stability, convergence and noise robustness.

3.5.1 Stability

We first consider the issue of stability of the proposed class of algorithms. Let W° denote the optimal equalizer setting and $\Delta W_k = W_k - W^\circ$ the deviation of the current equalizer setting w.r.t. the optimal one. Then by subtracting W° from both sides of (3.21) we obtain:

$$\Delta W_{k+1} = (I - \bar{\mu} \mathbf{X}_k (\mathbf{X}_k^H \mathbf{X}_k)^{-1} \mathbf{X}_k^H) \Delta W_k + \bar{\mu} \mathbf{X}_k (\mathbf{X}_k^H \mathbf{X}_k)^{-1} E_k, \quad (3.66)$$

where $E_k = \text{sign}(\mathbf{X}_k^H W_k) - \mathbf{X}_k^H W_k$. If now we define the matrix M_k as

$$M_k = I - \bar{\mu} \mathbf{X}_k (\mathbf{X}_k^H \mathbf{X}_k)^{-1} \mathbf{X}_k^H,$$

then the following recurrence holds:

$$\Delta W_{k+1} = (M_k M_{k-1} \cdots M_0) \Delta W_0 + \text{other terms}. \quad (3.67)$$

For a stable operation of the algorithm, the influence of the initial setting W_0 should asymptotically be eliminated. This implies that the eigenvalues of each one of the matrices M_k be absolutely bounded by 1 and also that the input signal is persistently exciting so that elimination of ΔW_k will occur in all subspaces. As the matrix $\mathbf{X}_k (\mathbf{X}_k^H \mathbf{X}_k)^{-1} \mathbf{X}_k^H$ is an orthogonal projection operator on the L -dimensional subspace of \mathbb{C}^N spanned by the columns of \mathbf{X}_k , it has $N - L$ eigenvalues equal to 0 and L eigenvalues equal to 1 (see also [Slo92b],[MD94]). Thus the matrix $M_k = I - \bar{\mu} \mathbf{X}_k (\mathbf{X}_k^H \mathbf{X}_k)^{-1} \mathbf{X}_k^H$ has $N - L$ eigenvalues equal to 1 and L eigenvalues equal to $1 - \bar{\mu}$. The following lemma holds:

Lemma 1: The algorithm (3.21) will be exponentially stable if the input signal is persistently exciting and the stepsize parameter $\bar{\mu}$ lies in the following region:

$$0 < \bar{\mu} < 2.$$

3.5.2 Convergence behaviour

We will examine separately two different aspects related to the convergence behaviour of NSWCMAs, i.e. we'll first examine issues related to the problem of *ill-convergence*, a problem often met in blind algorithms of the Bussgang type and then the convergence *dynamics* of the proposed class of algorithms.

Ill convergence

Even though the algorithm (3.29) has been designed from a deterministic point of view, it may however be seen as an algorithm of the stochastic gradient type that tries to minimize the following stochastic cost function:

$$J(W) = E(\|\mathbf{X}_k^H W - \text{sign}(\mathbf{X}_k^H W)\|_{p_k-1}). \quad (3.68)$$

The equilibrium points of the algorithm can be found by setting the derivative of $J(W)$ equal to zero. This will give:

$$E(\mathbf{X}_k P_k^{-1}(\text{sign}(\mathbf{X}_k^H W) - \mathbf{X}_k^H W)) = \mathbf{0} . \quad (3.69)$$

Eq. (3.69) describes a nonlinear system of N equations in N unknowns and therefore has a plenitude of solutions, some of which will be maxima and some of which minima of the cost function in (3.68). The existence of more than one solution to (3.69) is a direct consequence of the non-convex form of (3.68). In figure 3.5 one can see the cost functions of the CMA and the

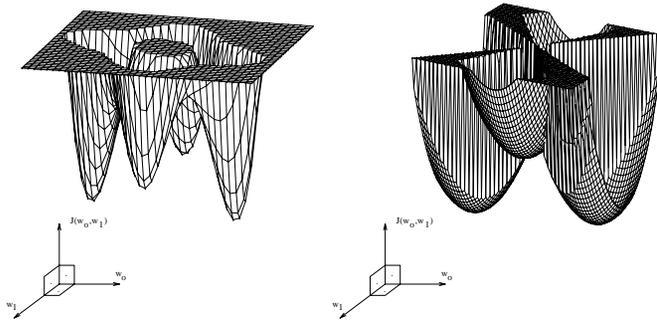


Figure 3.5: CMA and NSWCMA (L=2) cost functions in a case of two equalizer parameters (N=2)

NSWCMA (L=2) (left and right part, respectively) for the case of a two-dimensional equalizer and a FIR(1) communication channel. One can easily see two pairs of minima in each one of these cost functions. The deeper hills correspond to the global minima whereas the higher ones to undesired local minima. It is therefore clear that the problem of local minima exists also for the NSWCMA's.

However the NSWCMA's may in practice more easily override a false minimum point for the following reason: as was first shown by Mazo [Maz80] for the decision-directed LMS algorithm in a multilevel PAM context, the algorithm's stepsize is of crucial importance on what concerns its ill convergence. In fact, Mazo showed that when a constant stepsize is used the algorithm will finally escape from its local undesirable stationary points and will eventually end up to one of its global minima. However, especially when the stepsize is small, this might take too much time, justifying the claim that in this case the algorithm actually gets *trapped* by a false minimum. Such an escape however would be easier when a bigger stepsize is used. But in unnormalized algorithms one has to use a small stepsize in order to guarantee stability. This gives a potential advantage to normalized algorithms as the NSWCMA's whose

stable performance is guaranteed for all stepsize values $\bar{\mu} \in (0, 2)$. This may be theoretically justified from a deterministic point of view as follows. It is shown in Appendix 3.C that in the ideal case of a noiseless and exactly parameterized case the NSWCMA (the same holds for the NSWDDA and for all CMA's) will exactly stop if and only if it ends up at its global minimum point that corresponds to the optimal setting for the equalizer coefficients. This means that if the algorithm gets close to a local minimum it will continue on moving around it even in the absence of any kind of noise. It is exactly this movement that can be amplified by using a big stepsize which will help the algorithm escape from the local minimum until it reaches its global minimum where it will stop. Practical evidence for this behaviour will be given in the section of computer simulations.

Convergence dynamics

We will separately examine the noiseless and the noise-present cases. However before proceeding in this analysis it would be useful to discuss the role of noise in a blind-equalization setup. As mentioned in chapter 2, the modeling of noise in a blind equalization setup is somewhat different than in a classical system identification setup. In the latter case, one usually assumes the data to be noise-free and the desired signal only to be corrupted by some additive noise. So in such a case, if we denote by a_k the desired sample at time instant k and by W^o the optimal filter, then the following holds:

$$\begin{aligned} X_k^H W^o &= a_k \\ X_k^H W_k &= a_k + n_k , \end{aligned} \quad (3.70)$$

where n_k represents the additive noise corrupting the desired signal sample at time instant k . Therefore, if we denote by ΔW_k the deviation of W_k w.r.t. W^o , i.e. $\Delta W_k = W_k - W^o$, this will satisfy:

$$X_k^H \Delta W_k = n_k . \quad (3.71)$$

In the blind equalization setup on the contrary noise adds up in a different way: first, the noise contaminates the received data, and thus we have:

$$x_k = \chi_k + \eta_k , \quad (3.72)$$

where η_k represents the additive noise corrupting the noise-free data χ_k . Moreover, the desired signal d_k also deviates from the ideal a_k because of the "blind" character of the algorithm:

$$d_k = a_k + n_k . \quad (3.73)$$

In this case n_k does not represent (only) measurement noise as in (3.71) but also deviations between $\text{sign}(X_k^H W_k)$ from a_k . These deviations are not only large when the algorithm is far from convergence but are also amplified due to the already noise-contaminated data x_k . Moreover, for a non-cyclic constellation, even at convergence there will always be a residual

noise in d_k because of the distances between the constellation points and the unit circle. Finally, since in reality the communication channel is modeled as a FIR filter, a finite-length equalizer can never match exactly the infinite-length channel's inverse impulse response. This gives rise to yet another kind of noise that we call "under-parameterization" noise. So, even in the case of simple LMS decision-directed algorithm for blind equalization, the effects of noise will be quite more severe than in a classical adaptive filtering context. We now discriminate between the two cases:

- Noiseless case

In order to describe a "noiseless" case in a blind equalization setup we make the following assumptions:

- The channel is modeled as an AP(N-1) filter and the equalizer as a FIR(N-1) filter. Therefore the equalizer is capable of matching exactly the channel's inverse impulse response (exact parameterization). No additive noise corrupts the channel's output and therefore no noise amplification corrupts the *sign* signal at reception.

- The transmitted constellation is constant modulus and therefore no noise arises from the mismatch of the transmitted constellation and a constant modulus constellation patents. In this case we are close to a classical adaptive filtering noiseless case, the only difference being the replacement of the truly desired signal a_k by the quantity $\text{sign}(X_k^H W_k)$. Following the same reasoning as in section (3.5.1), the algorithm (3.21) will be performing a series of successive projections onto the subspace of \mathcal{C}^N spanned by the first L columns of the data matrix \mathbf{X}_k so as to match at each iteration the vector $\text{sign}(\mathbf{X}_k^H W_k)$. Depending on its initialization, the algorithm will proceed towards a local or a global minimum of its cost function (3.68). As explained above however, it will only stop when it reaches its global minimum point. This will happen more and more fast as the algorithm's stepsize $\bar{\mu}$ approaches unity.

Another interesting issue in the noiseless case is the algorithm's sensitivity to the correlation of the received data process. Since we consider the channel to be modeled as an AP(N-1) linear filter, all correlation information may be described by its $N \times N$ covariance matrix. When the channel's poles are close to the unit circle, this corresponds to a frequency spectral response with deep spectral nulls, resulting to an ill-conditioned covariance matrix, since:

$$\frac{\lambda_{max}}{\lambda_{min}} \simeq \frac{S_{max}}{S_{min}}, \quad (3.74)$$

where λ_{max} and λ_{min} are the largest and smallest eigenvalues of the data covariance matrix, respectively and S_{max} , S_{min} the largest and smallest values of the channel's frequency response, respectively [Hay91]. The role of P_k^{-1} in (3.21) is to perform some kind of prewhitening of the received signal. When $P_k = I$ there is no prewhitening and the algorithm's convergence

speed will be low for a strong colouring of the input signal. On the other hand, the prewhitening performed by $P_k = \mathbf{X}_k^H \mathbf{X}_k$ attenuates the influence of this colouring on the algorithm's speed (the best case being $L = N$). Therefore, in the case of a noiseless channel and an exactly parameterized equalizer the algorithm's convergence speed will be maximal for $L = N$ and $\bar{\mu} = 1$.

An interesting interpretation of the algorithm's behaviour may be obtained by considering its deterministic aspect. Consider for simplicity a 2-PAM scheme, where the transmitted symbols may equally likely take on the values $+1$ and -1 and moreover the case of a real channel. Then the vector $[\text{sign}(X_k^H W_k) \text{sign}(X_{k-1}^H W_k) \cdots \text{sign}(X_{k-L+1}^H W_k)]^T$ will contain only $+1$ and -1 elements. If by chance the L elements of these vector coincide with the corresponding symbols at time instants $k, k-1, \dots, k-L+1$, i.e.

$$[\text{sign}(X_k^H W_k) \text{sign}(X_{k-1}^H W_k) \cdots \text{sign}(X_{k-L+1}^H W_k)]^T = [a_k \ a_{k-1} \cdots a_{k-L+1}]^T, \quad (3.75)$$

then, if $L = N$ and $\bar{\mu} = 1$ the algorithm will converge to its optimal setting in one more iteration (the same will happen when $[\text{sign}(X_k^H W_k) \text{sign}(X_{k-1}^H W_k) \cdots \text{sign}(X_{k-L+1}^H W_k)]^T = -[a_k, a_{k-1} \cdots a_{k-L+1}]^T$, since then the opposite-to-the optimal setting is obtained, which is also acceptable (a phase-ambiguity may be eliminated if differential coding is used)). So, if (3.75) is valid from the first iteration, the algorithm will converge in only one iteration! This is indicative of the improvement in convergence speed due to normalization.

- Noise present

We now consider the more realistic case of a FIR channel and an additive white Gaussian noise corrupted received signal. In this case, as explained in the beginning of this section, both the data and the desired signal will be noise-corrupted, the first directly, the second indirectly. We concentrate on the following step of the algorithm:

$$P_k H_k = [\text{sign}(\mathbf{X}_k^H W_k) - \mathbf{X}_k^H W_k]^T = E_k. \quad (3.76)$$

The robustness of the solution H_k found from (3.76) w.r.t. deviations of E_k from its exact value has to do with the conditioning of the matrix P_k . When P_k has a big eigenvalue spread (and this will happen when the channel's zeros are close to the unit circle) the solution found by (3.76) will be very sensitive to small deviations of E_k w.r.t. its real value. The deviations of E_k that are due to the additive noise have to do with the location of the channel's zeros, for the following reason: when the channel's zeros are close to the unit circle this means that the channel's inverse will have spectral peaks. As a consequence of that, an equalizer that approaches the channel's inverse frequency response will amplify the additive noise at its input in a very disproportional way for different frequencies. This will result in a non-uniform corruption of E_k along different eigenvalue directions. We thus see that the channel's

“conditioning” plays a very important role in the noisy case.

Quantitatively, the noise contribution to the MMSE can be found to be directly proportional to the following term:

$$\alpha = \bar{\mu} \mathbf{X}_k P_k^{-2} \mathbf{X}_k^H . \quad (3.77)$$

Now if one uses the SVD of the data matrix $\mathbf{X}_k = U \Sigma V^H$, (3.77) becomes:

$$\alpha = \bar{\mu} U \Sigma^{-2} U^H . \quad (3.78)$$

This last equation shows indeed why when the matrix $P_k = \mathbf{X}_k^H \mathbf{X}_k$ has a big eigenvalue spread, the noise will get amplified in a large and very disproportional manner along the different eigenvector directions of this matrix, thus resulting in a big steady-state MSE. This will become more and more severe as L moves from 1 to N , which will be the worst case from this point of view. So one sees that increasing very much the dimension L towards N in the noisy case may have a catastrophic effect in the algorithm’s steady-state error in the contrary to the noiseless case where it had the beneficial effect of making it insensitive to the received signal’s colouring. This is why one should be careful in choosing the algorithm’s parameters in such a way so as to guarantee both a fast convergence and a low steady-state error.

Different remedies to the problems arising from the ill-conditioning of the sample covariance matrix P_k can be either the somewhat regularization of P_k or the reduction of the stepsize $\bar{\mu}$ or the choice of a low L or even a combination of such remedies. In the first case one deviates from satisfying exactly the deterministic criterion (3.18), in the following two cases one reduces the convergence speed. A solution based on the regularization of P_k will be given in the next chapter, where an exponential sliding window for estimating P_k will be used instead of a rectangular one. A corresponding deterministic criterion is given there also.

3.6 Computer simulation results

The behaviour discussed above of the proposed class of algorithms has been tested by computer simulations. We have considered both PAM and QAM constellations transmitted through linear channels. For the testing of noiseless cases we have simulated AP channels of the same order as the equalizer, whereas for the non-ideal case the channels were simulated as FIR non-minimum-phase filters. The following results give practical evidence of the proposed algorithm’s behaviour in terms of convergence speed, noise robustness and ill-convergence avoidability. As a measure of performance we will use the closed-eye measure of the communication system, defined in (2.25) (the only difference being now that instead of the channel impulse response $\{c_i\}$ the overall channel-equalizer response $\{s_i\}$ has to be used).

Another aspect that should be presented before the computer simulations is the following: often we will use the following modeling of a channel output: the channel output will be supposed to be an AR process whose roots may be either inside or outside the unit circle. Such a process may result by passing a white noise through the following all-pole (AP) system

$$H(z) = \frac{1}{A(z)} = \frac{1}{A^+(z^{-1}) A^-(z^{-1})} .$$

In the above representation, the polynomials $A^+(z^{-1})$ and $A^-(z^{-1})$ are maximum and minimum phase, respectively, i.e. the first contains roots only outside and the second only inside the unit circle. A problem with the implementation of such a system is of course that it cannot be used as such, since it is unstable. However the corresponding AR process can be still produced by introducing a delay. Namely, consider the maximum phase factor:

$$\frac{1}{A^+(z^{-1})} = \frac{1}{\alpha_0 + \alpha_1 z^{-1} + \dots + \alpha_p z^{-p}} .$$

This can be written in the form

$$\frac{1}{A^+(z^{-1})} = \frac{z^p}{1 + \alpha_{p-1} z + \dots + \alpha_0 z^p} ,$$

where now the denominator is minimum phase, and therefore stable. The AR process can then be implemented by first passing the white noise through this filter, and then through the minimum phase factor $\frac{1}{A^-(z^{-1})}$. This is how the problem of stability is overridden.

The fact that we use such AR processes for some of our simulations, is due to the fact that, as already explained, we want to simulate some “perfectly” parameterized cases in the sense that zero-forcing FIR equalizers exist.

3.6.1 Opening the channel eye

In a first step, we will show the ability of the proposed algorithms to open the closed eye of a linear communication channel. For this, we implemented an FIR complex channel with the following impulse response:

$$\{h\} = [-1.0397 + 0.3055i \quad 0.7846 + 0.6749i \quad -0.3433 - 2.1848i \quad 2.3927 - 0.0552i] .$$

The magnitudes of the roots of the corresponding polynomial $H(z)$ are

$$\begin{aligned} \rho_1 &= 1.8859 \\ \rho_2 &= 1.5865 \\ \rho_3 &= 0.7382 \end{aligned}$$

A 4-QAM sequence was passed through this complex channel and then additive Gaussian noise was added at the channel output, so that SNR=30 dB. We implemented the NSWCMA algorithm using the following parameters:

$\bar{\mu}$	L	N
0.01	3	21

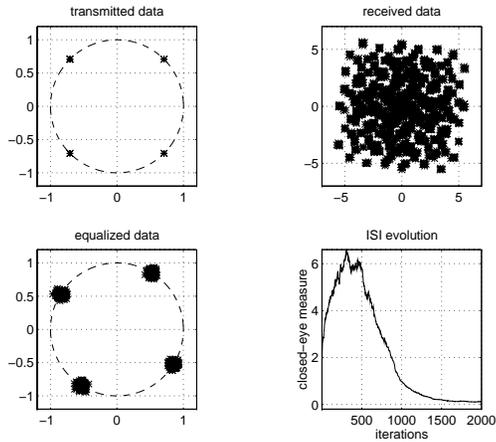


Figure 3.6: The opening of the eye achieved by the NSWCMA algorithm

The 21-tap equalizer was initialized with a single non-zero middle tap equal to 1. Figure 3.6 shows the results obtained after 2000 iterations of the algorithm. Note the closed-eye form of the received data sequence and how the transmitted 4-QAM pattern is retrieved after the algorithm's convergence. This shows the ability of the algorithm to open an initially closed channel eye. Note also the fact that the obtained equalized constellation is rotated with respect to the original one, a typical characteristic of all blind equalization methods.

3.6.2 Convergence speed

Noiseless case

As we already stated, in order to have a "noiseless" case, we need to model the channel output as an AR process of the same order as the equalizer. Moreover, the input sequence has to be CM, and of course, no additive noise has to be added. Even though such a scenario is strictly speaking not realistic, we have simulated it in order to verify the corresponding theoretical analysis. We consider two cases:

- Weak colouring of the channel output

In this case we consider an all-pole non-minimum phase channel AP(7) whose denominator $A(z)$ has the following roots

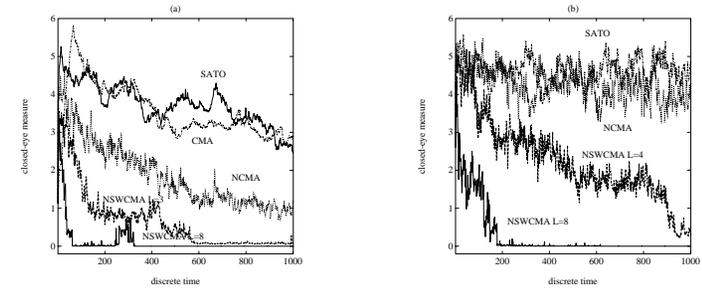


Figure 3.7: Noiseless case : two examples

$$\begin{aligned}\rho_1 &= 0.1 \\ \rho_{2,3} &= 0.2e^{\pm j\pi/4} \\ \rho_{4,5} &= 5e^{\pm j\pi/4} \\ \rho_{6,7} &= 6e^{\pm j\pi/6}\end{aligned}$$

Note that these zeros are far from the unit circle, which shows a weak colouring of the channel output and a well conditioned sample covariance matrix P_k . We consider a length-8 equalizer (whose order matches exactly the order of the inverse channel). A 2-PAM i.i.d. input sequence is transmitted through the channel, and several BE algorithms were simulated for the equalizer adaptation. Namely, we have implemented the SATO and CMA algorithms with step-sizes that have been found by trial and error to guarantee stability as well as the NSWCMA with unit stepsize ($\bar{\mu} = 1$) and three different choices for L (1,3,8). Figure 3.7(a) shows the evolution of the closed-eye measure of the communication system for the different algorithms during 1000 iterations. One can see the faster convergence provided by NCMA w.r.t. SATO and CMA and the further increase of convergence speed of NSWCMA as L grows up towards N . This verifies the theoretically expected performance: as L moves towards N , the algorithm becomes insensitive to the colouring of its input signal, and therefore converges faster.

- Stronger colouring of the channel output

In this case the poles of the AP(7) channel are

$$\begin{aligned}\rho_1 &= 0.3 \\ \rho_{2,3} &= 0.5e^{\pm j\pi/4} \\ \rho_{4,5} &= 1.5e^{\pm j\pi/4} \\ \rho_{6,7} &= 2e^{\pm j\pi/6}\end{aligned}$$

This channel has its poles closer to the unit circle than the previous one, thus resulting in a more strongly coloured received signal and a more ill-conditioned matrix P_k . Figure 3.7(b) shows again the evolution of the closed-eye measure by different algorithms in this case. We can see that all algorithms except NSWCMA $L = N$ have now a lower convergence speed because of the colouring of the received signal. However, NSWCMA $L = N$ seems to be insensitive to this colouring and to converge roughly at the same speed as for the previous channel, thus confirming its prewhitening feature.

We have also run the algorithm for channel settings whose zeros are very close on the unit circle. In these cases the NSWCMA has been shown able to converge, however, as expected it takes much more time before it reaches its steady state.

Noisy case

The improvement in convergence speed is expected to show up to the realistic case of FIR noisy channels as well: the difference now is that we cannot be allowed to give very high values to the algorithm's parameters L and μ , due to the steady-state error amplification discussed before. If this guideline is followed however, significant improvements in convergence speed with respect to the CMA can be observed.

Such an example is shown in the following simulated case: we consider an FIR complex channel with the following impulse response

$$\{h\} = [-1.0493 + 0.2305i \quad 1.4129 - 1.4497i \quad -0.2540 + 0.2021i \quad 0.5302 - 0.7732i] .$$

The magnitudes of the roots of the corresponding polynomial $H(z)$ are now

$$\begin{aligned} \rho_1 &= 1.9106 \\ \rho_2 &= 0.6421 \\ \rho_3 &= 0.7113 \end{aligned}$$

We simulated the CMA and the NSWCMA using the parameters shown in the following table

	$\bar{\mu}$	L	N
a	0.1	7	21
b	0.1	4	21
c	1	1	21
d	0.001		21

over 2000 iterations each, and in all cases used a center-spike initialization for the equalizer. The SNR at the channel output is 30 dB. Figure 3.8 shows the evolution of the closed-eye measure for the four different algorithms. Note how the channel opening is achieved much faster by the three members of the NSWCMA as compared to the CMA (it should be noted that the stepsize we used for the CMA was found by trial and error to be among the biggest ones that guarantee stability). For example, in the case of $L = 7$ the channel eye opens

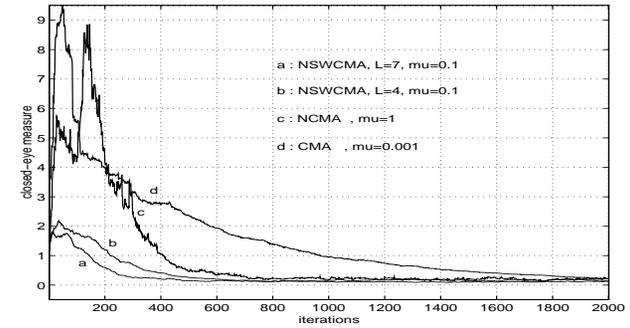


Figure 3.8: Noisy FIR channel: comparative simulations

roughly 7 times faster than the CMA! It is also observed that the convergence speed increases, as expected, with L . The price paid for the increase in convergence speed is, of course, the increased computational complexity.

3.6.3 Avoiding local minima

We now consider a simple example that shows the potential ability of NSWCMA to escape from local minima thus avoiding ill-convergence. We consider an AR(1) channel with the following z-representation:

$$C(z) = \frac{1}{1 + 0.25z^{-1}} , \quad (3.79)$$

and an FIR(1) equalizer : $W = [w_0 \ w_1]^T$. We transmit a 2-PAM sequence with no additive noise through the AR channel and then update the equalizer W by using both the CMA and the NSWCMA($L=2$) algorithm. The convergence trajectories for of the equalizer vector for 40 different initializations on a circle of radius 2 are shown in figure 3.9.

As can be seen in figure 3.9a, the CMA (with a step-size that has been found by trial and error to guarantee stability) may end up either to one of its two optimal settings $W = \pm[1 \ 0.25]^T$ or to one of two stationary points on the $w_0 = 0$ axis, corresponding to local minima of its cost function, depending on its initialization. Note that as has been already mentioned and shown in Appendix 3.C, the algorithm only stops exactly at its global minima and continuously moves around its local minima, even in the absence of noise. Figure 3.9b shows the same experiment tested on the NSWCMA ($L = 2$) algorithm when a small stepsize ($\bar{\mu} = 0.05$) is used. The same more-or less image as for the CMA holds. However one may note that the motion around the local minima is much wider now, and there are already some "escapes"

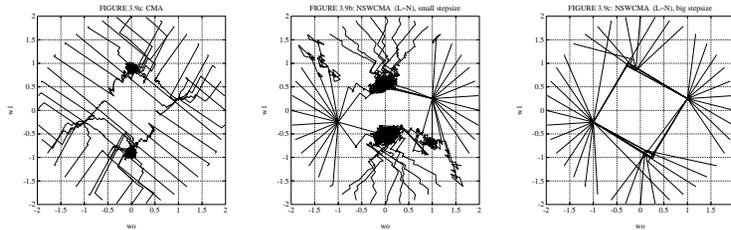


Figure 3.9: A comparison of CMA and NSWCMA ($L=2$) for an AP(1) channel

from local minima towards global minima. Finally figure 3.9c shows the of the NSWCMA ($L = 2$) with $\bar{\mu} = 1$. Here one can see that local minima are quickly abandoned and all initializations end up to the global minima. This shows the ability of the NSWCMA's to escape from local minima due to their well-defined stepsize region for stability. It should be noted here that the CMA also has been found to be able to escape from local minima for big values of its stepsize, but the same values would make the algorithm exponentially unstable in other realizations.

A similar behaviour has been found out in a noisy case as well. Figure 3.10 (left part) shows the evolution of ρ in the case of a FIR(2) channel with impulse response $[1 \ 0.6 \ 0.36]$ corrupted by additive Gaussian noise (SNR=20 dB) for two different initializations when the CMA with $\mu = 0.004$ is used. It can be seen that the algorithm ill-converges for one of these two initializations. On the contrary, when the same experiment was carried out by employing the NSWCMA ($L = 2$) with $\bar{\mu} = 0.1$, both initializations lead to the opening of the system's eye (right part of figure 3.10). Note that in this case the initialization that lead to ill-convergence when the CMA was used initially heads towards an unacceptable point but then quickly escapes and heads towards the desired solution.

3.7 Conclusions

We have proposed a new class of adaptive filtering algorithms for the blind equalization of constant modulus signals transmitted through a linear possibly non-minimum phase communication channel. The algorithms are derived by minimizing at each iteration a deterministic criterion based both on the constant modulus property and on a norm restriction of the "next" equalizer setting. Some interesting interpretation on the algorithm's derivation has indicated how any adaptive filtering algorithm may be used in a constant-modulus blind equalization

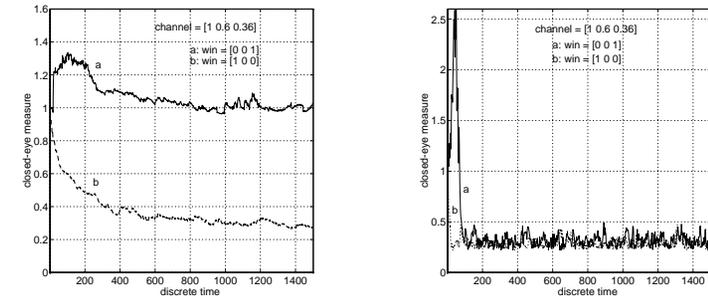


Figure 3.10: Ill convergence: a more realistic example

setup. The algorithm's behaviour has been discussed in terms of convergence speed, computational complexity, robustness to noise, ill-convergence. It has been shown that the algorithm can achieve a faster convergence rate w.r.t. conventional CMA's in the cost of its increased numerical complexity. Another important property is that its inherent normalization allows for the use of stepsize values that *a priori* guarantee stable operation. This has been also shown to have beneficial effects in terms of ill-convergence avoidance possibilities. Finally computer simulation results have been provided in support of the claimed behaviour. Some of the results presented in this chapter have been also presented in [PS93b], [PS93c], [PS94a].

3.8 Appendix 3.A

We will show at this point why the choice of e_k in (3.12) is the one that minimizes the quantity $\|W_{k+1} - W_k\|_2^2$. From (3.11) we have :

$$\|W_{k+1} - W_k\|_2^2 = \|(X_k^H X_k)^{-1}\|_2^2 \|X_k^H W_k - e_k\|^2 . \quad (3.80)$$

Therefore the following holds:

$$\min_{e_k} \{\|W_{k+1} - W_k\|_2^2\} \iff \min_{e_k} \{|X_k^H W_k - e_k|^2\} . \quad (3.81)$$

Now the right-hand side of (3.81) can be written as follows:

$$\min_{e_k} \{|X_k^H W_k - \text{sign}(l_k)|^2\} , \quad (3.82)$$

where $e_k = \text{sign}(l_k)$. As $X_k^H W_k$ and $\text{sign}(l_k)$ are complex numbers which can be considered as 2-dimensional vectors, their Euclidean distance will be minimized when they are colinear (see figure 3.2). Having defined $\text{sign}(x) = \frac{x}{|x|}$, this colinearity implies that :

$$\text{sign}(X_k^H W_k) = \text{sign}(\text{sign}(l_k)) = \text{sign}(l_k) . \quad (3.83)$$

Therefore the optimal choice for e_k is

$$e_k^{\text{opt}} = \text{sign}(X_k^H W_k) . \quad (3.84)$$

3.9 Appendix 3.B

We will now prove that the deterministic criterion

$$\min_{W_{k+M}} \|D_k - X_k^H W_{k+M}\|_{S_k}^2 + \|W_{k+M} - W_k\|_{T_k}^2 , \quad (3.85)$$

where S_k and T_k are positive definite matrices, is minimized by the following argument:

$$W_{k+M} = W_k + T_k^{-1} X_k (S_k + X_k^H T_k^{-1} X_k)^{-1} (D_k - X_k^H W_k) . \quad (3.86)$$

We denote by J_k the quantity $\|D_k - X_k^H W_{k+M}\|_{S_k}^2 + \|W_{k+M} - W_k\|_{T_k}^2$. The gradient of J_k w.r.t. W_{k+M}^H is given by:

$$\nabla_{W_{k+M}^H} (J_k) = -X_k S_k^{-1} (D_k - X_k^H W_{k+M}) S_k^{-1} X_k + T_k^{-1} (W_{k+M} - W_k) . \quad (3.87)$$

Setting this gradient equal to zero, we obtain:

$$W_{k+M} = (T_k^{-1} + X_k S_k^{-1} X_k^H)^{-1} (T_k^{-1} W_k + X_k S_k^{-1} D_k) . \quad (3.88)$$

Now we can apply the Matrix Inversion Lemma to the term $(T_k^{-1} + X_k S_k^{-1} X_k^H)^{-1}$:

$$(T_k^{-1} + X_k S_k^{-1} X_k^H)^{-1} = T_k^{-1} - T_k X_k (S_k + X_k^H T_k^{-1} X_k)^{-1} X_k^H T_k^{-1} . \quad (3.89)$$

Substituting (3.89) in (3.88) we obtain after some calculation eq. (3.86).

3.10 Appendix 3.C

In the following analysis we consider the communications channel to be modeled as an AP($N-1$) channel and the equalizer is FIR with N coefficients. Then the equalizer will be able to exactly match the channel's inverse and problems arising from the under-parameterization of the equalizer will be avoided. The transmitted symbols are assumed to belong to a cyclic PSK constellation and to be independent from one another, i.e. they constitute a white noise:

$$E(a_i a_j^*) = \alpha \delta_{ij} , \quad E a_i = 0 , \quad (3.90)$$

where δ_{ij} denotes the Kronecker delta and α a real scalar. The fact that the channel is AP($N-1$) is described by the following equation:

$$X_k^H W^o = a_k , \quad k = 1, 2, \dots , \quad (3.91)$$

where W^o is a column vector that contains the AP channel's coefficients. According to (3.91) W^o is the equalizer's optimal setting.

The same is true up to any modulus one complex scalar factor, i.e. any modulus one complex multiple of W^o is also optimal in the sense that it results in zero ISI (it completely opens the channel's eye). Of course in the latter case the transmitted constellation will be received rotated by an arbitrary angle. However this phase-shift can be eliminated if the transmitted data are differentially encoded. We will now show that these optimal stationary points are the only ones where the algorithm (in the ideal case of an AP noiseless channel) perfectly stops. Such a stationary point should satisfy:

$$\mathbf{X}_k^H W = \text{sign}(\mathbf{X}_k^H W) \quad k = 1, 2, \dots . \quad (3.92)$$

If one writes eq. (3.91) at L successive time instants, one obtains:

$$\mathbf{X}_k^H W^o = [a_k \ a_{k-1} \ \dots \ a_{k-L+1}]^T \quad k = 1, 2, \dots . \quad (3.93)$$

It is obvious from the above equation that the optimum equalizer settings $e^{j\theta} W^o$ satisfy eq. (3.92) since $\text{sign}(e^{j\theta} a_{k-i}) = e^{j\theta} a_{k-i}$. Therefore the algorithm will exactly stop if it attains one of its optimal settings. The question now is if it can exactly stop at another stationary point. Let us denote by H the overall input-output linear filter consisting of the cascade of the transmission channel and the equalizer and let $\{h_i\}$ be its impulse response. Then the output of the equalizer at time instant k is given by:

$$y_k = X_k^H W_k = \sum_{i=-\infty}^{+\infty} h_i a_{k-i} . \quad (3.94)$$

Consider now that the equalizer's setting corresponds to a stationary point that causes the algorithm to stop exactly. Then the equalizer's output will satisfy $|y_k| = 1$ and therefore,

$E|y_k|^2 = 1 \Rightarrow E \sum_{i,j} h_i h_j^* a_{k-i} a_{k-j}^* = 1$. But by definition $E a_{k-i} a_{k-j}^* = \delta_{ij}$ as stated in (3.90).

Therefore:

$$\sum_{i=-\infty}^{+\infty} |h_i|^2 = 1 . \quad (3.95)$$

Also, as $|y_k| = 1 \Rightarrow |\sum_{i=-\infty}^{+\infty} h_i a_{k-i}| = 1$ ⁵. As this should be true for all possible sequences $\{a_i\}$, it should also be valid for the particular choice $a_{k-i} = \text{sign}(h_i)$, which gives:

$$\sum_{i=-\infty}^{+\infty} |h_i| = 1 . \quad (3.96)$$

Combining now (3.94) and (3.95) one gets as only possible solution:

$$h_i = e^{j\phi} \delta_{li} \text{ for some integer } l. \quad (3.97)$$

We therefore see that the only setups where the algorithms will perfectly stop in a noiseless case are phase-shifted versions of the optimal equalizer setting W^o . This reflects also the well-known fact that blind equalizers identify the transmitted constellation up to an arbitrary phase shift.

In the case of a non-cyclic QAM constellation the same reasoning may be applied to the NSWDDA to show that it will also stop exactly only at one of its optimal settings.

⁵Strictly speaking, this is clear for a uniform continuous distribution on the unit circle. The same however can be proven for discrete CM distributions.

Chapter 4

A modified APA algorithm for adaptive filtering and blind equalization

MOTIVATED by the fact that both the APA and the NSWCMA algorithms may exhibit a high steady-state error when the length of their sliding window is close to N , we propose first a modified (*regularized*) class of algorithms of the APA type for classical adaptive filtering. By applying the separation principle introduced in chapter 3, we obtain immediately a counterpart for blind equalization. An efficient algorithmic organization is proposed and some bounds of the employed parameters for stability are theoretically derived. The expected improvement in performance is verified by computer simulations in a blind equalization context.

4.1 Introduction

The aim that motivated the development of the APA for classical adaptive filtering and of the NSWCMA (in chapter 3) for adaptive blind equalization has been the construction of algorithms that exhibit an increased convergence speed with respect to the LMS (resp. the CMA) without increasing significantly the corresponding computational complexity. The two crucial parameters of the APA (or the NSWCMA) that control the convergence speed are:

- the stepsize ($\bar{\mu}$)
- the length L of the sliding window

Namely, as already mentioned, the convergence speed increases with these two parameters. On the other hand however, due to the additive noise, the steady-state MSE of the algorithms of the APA type increases also with the stepsize and with the window length (see eq. (3.77)).

Therefore in some cases when one will want to push the parameters to high values in order to have a very fast convergence, the algorithm will have a poor steady-state performance¹. This leads us to the conclusion, in chapter 3, that there is always a compromise to be made between convergence speed and steady-state performance.

As indicated by eq. (3.78), it is mainly the *conditioning* of the sliding-window covariance matrix $\mathbf{X}_k^H \mathbf{X}_k$ that influences the steady-state performance of the APA (or the NSWCMA) at the presence of noise. In fact we concluded from several computer simulations that the conditioning of the sliding-window covariance matrix may be rather high at specific time instants, even though the algorithm has converged, if the window length L approaches the filter length N . An extreme example of such a case is shown in figure 4.1. We have simulated the NSWCMA for the channel $[1 \ 2.5 \ 1]$, taking $N = 5$, $L = 5$, $\bar{\mu} = 0.1$ and $SNR = 20dB$. Fig. 4.1 (a) shows the evolution of the condition number of the covariance matrix $\mathbf{X}_k^H \mathbf{X}_k$. Clearly the condition number becomes excessively high at some time instants, irrespective of the phase of the algorithm (initial convergence or steady-state). Due to this the channel eye corresponding at each iteration of the algorithm has the erratic behaviour shown in figure 4.1 (b): the eye measure heads quickly towards values below 1 after initial convergence, but then continues on switching between high and small values during steady-state!² In practice the filter length will be much longer than in this example, and the window length will be chosen smaller than the filter length in order to keep down the computational complexity, however the above example is indicative of the adverse influence of a long ($L \simeq N$) sliding window on the algorithm's steady-state performance in the presence of noise.

It is clearly meaningful from the above to try to improve the steady state behaviour of the APA (or the NSWCMA) by reducing the steady-state excess error due to noise amplification. Since the key feature seems to be the conditioning of the sliding window covariance matrix, it would be interesting to find ways to *regularize* this matrix (i.e. reduce its condition number), without destroying its covariance character. The next section describes a class of algorithms of the APA type that can be obtained in the above sense.

4.2 A class of regularized APA-type adaptive filtering algorithms

The above-mentioned ill conditioning of the sample covariance matrix at certain iterations is due both to the colouring of the received discrete-time process $\{x_i\}$ ³ and to the fact that it is an instantaneous covariance matrix based on a finite-length rectangular window. An obvious way of improving the conditioning of the covariance matrix is to use a window length L much

¹This phenomenon is well known from the performance of the NLMS algorithm, whose steady-state error increases with the value of its normalized stepsize.

²The switching between different high and low values corresponds also to transitions between local and global minima of the corresponding non-convex cost function.

³The lower bound of the condition number (= 1) is achieved by the true covariance matrix of a white process.

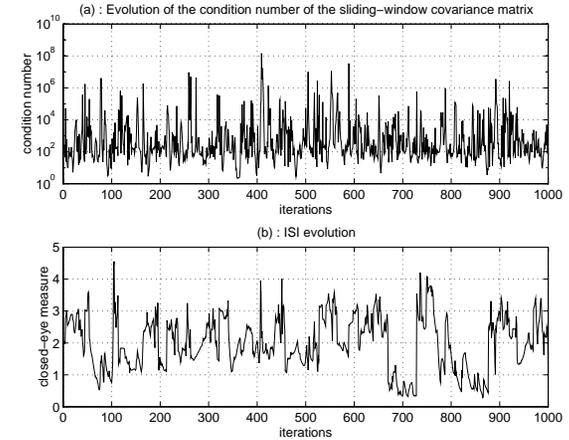


Figure 4.1: The bad effect of a long sliding window in a blind equalization experiment

bigger than the filter length N . In this case the obtained matrix will approximate the true covariance matrix and will not exhibit such a large variance from one iteration to another. However this would be impractical because of the associated increase in computational complexity. Another alternative would be to use the true covariance matrix (e.g. in the example of the previous paragraph the true covariance matrix has a condition number equal to 24.28). However this is impossible since the true covariance matrix is not known and needs a lot of data in order to be estimated. Moreover, when the received signal is not stationary, there will never be enough data to estimate the true covariance matrix corresponding to a specific time interval.

Another alternative often used in adaptive filtering (e.g. in the RLS algorithm) is the use of an *exponential* instead of a rectangular window. An exponential window offers two advantages, namely, on one hand it allows to follow the variations of the received signal, on the other hand it has a much bigger (asymptotically infinite) window length that provides a better approximation to the true covariance matrix than a rectangular window.⁴ For example, figure 4.2 shows the condition number of a sliding window covariance matrix based on an exponentially weighted data window for the example of the previous paragraph (continuous-line). The matrix at each iteration is updated as:

$$R_k = \lambda R_{k-1} + X_k X_k^H, \quad (4.1)$$

⁴The role of the covariance matrix in the APA is to perform a certain *prewhitening* of the received signal, since an adaptive algorithm converges faster when its input signal is white.

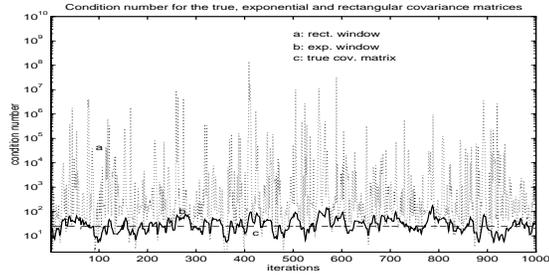


Figure 4.2: A comparison of the conditioning of a rectangular and an exponential covariance matrix

where \mathcal{X}_k is the first column of \mathbf{X}_k^H :

$$\mathcal{X}_k = \begin{bmatrix} x_k \\ x_{k-1} \\ \dots \\ x_{k-L+1} \end{bmatrix}, \quad (4.2)$$

and λ is the so-called *forgetting factor*. The value used for the example of fig. 4.2 is $\lambda = 0.9$. The dashed line represents the condition number of the true covariance matrix for this example. One clearly notices the better conditioning of the exponential covariance matrix with respect to the rectangular covariance matrix (dotted line). Therefore one would expect a better performance of the APA algorithm in the presence of noise if instead of rectangular, an exponentially-weighted data window is used for the construction of the instantaneous covariance matrix. In the next paragraph we will show how a suitable class of algorithms can be obtained by minimizing a deterministic criterion.

4.2.1 Derivation

We confine ourselves to a classical adaptive filtering context. Let d_k denote the desired response that has to be matched by the filter output at time instant k and by D_k a vector of the L most recent desired responses:

$$D_k = [d_k \ d_{k-1} \ \dots \ d_{k-L+1}]^T. \quad (4.3)$$

Now consider the following deterministic criterion:

$$\|D_k - \mathbf{X}_k^H W_{k+1}\|_{S_k}^2 + \|W_{k+1} - W_k\|^2, \quad (4.4)$$

where

$$S_k = \mu^{-1} R_k - \mathbf{X}_k^H \mathbf{X}_k, \quad (4.5)$$

and R_k is an $L \times L$ matrix updated as in (4.1). According to the result presented into Appendix 3.B, the criterion (4.4) is minimized by the update equation:

$$W_{k+1} = W_k + \mathbf{X}_k (S_k + \mathbf{X}_k^H \mathbf{X}_k)^{-1} (D_k - \mathbf{X}_k^H W_k). \quad (4.6)$$

Introducing the expression for S_k given by (4.5) in (4.6) we find that the deterministic criterion (4.4) is exactly minimized at each iteration by the algorithm:

$$\begin{aligned} R_k &= \lambda R_{k-1} + \mathcal{X}_k \mathcal{X}_k^H \\ W_{k+1} &= W_k + \mu \mathbf{X}_k R_k^{-1} (D_k - \mathbf{X}_k^H W_k). \end{aligned} \quad (4.7)$$

Eq. (4.7) describes a new parametric class of algorithms for adaptive filtering. This class of algorithms has been derived by a deterministic criterion (4.4) in a way similar to the derivation of the APA. However it does not have a projection interpretation like the one described in the previous chapter for the APA because of the exponential weighting used for the updating of the covariance matrix R_k^{-1} . On the other hand, due to the better behaviour of the conditioning of this covariance matrix, a better steady-state behaviour is expected with respect to the APA.

The algorithm has three adjustable parameters: L , λ and μ . L is the number of constraints imposed on the filter setting W_{k+1} that are expressed by the first term of (4.4) and coincides also with the size of the square matrix R_k . λ is the forgetting factor of the exponential sliding window and controls the algorithm's tracking of the input signal as well as the conditioning of the matrix R_k . Finally μ is a stepsize parameter that controls the deviation of the "next" filter W_{k+1} w.r.t. W_k . We give this class of adaptive filtering algorithms the name *Exponentially Regularized APA*: ERAPA.

In the next paragraph we provide an asymptotic analysis of the algorithm that provides some useful stability bounds.

4.2.2 Asymptotic analysis

Stability

In order to check the region of μ that guarantees the stability of (4.7), we use the following asymptotic approach: consider equations (4.5) and (4.1). Taking expectation of both sides in both equations one gets:

$$\begin{aligned} E(S_k) &= \mu^{-1} E(R_k) - E(\mathbf{X}_k^H \mathbf{X}_k) \\ E(R_k) &= \lambda E(R_k) + R \end{aligned}, \quad (4.8)$$

where R is the true $L \times L$ covariance matrix: $R = E \mathcal{X}_k \mathcal{X}_k^H$. Combining these two equations one obtains:

$$E(S_k) = R \left(-N + \frac{1}{\mu} \times \frac{1}{1-\lambda} \right) = NR \left(\frac{1}{\mu \times N(1-\lambda)} - 1 \right). \quad (4.9)$$

Since the criterion (4.4) only makes sense for S_k positive definite, S_k should also be asymptotically positive definite, and thus the following must hold according to (4.9) :

$$\mu < \frac{1}{N(1-\lambda)} . \quad (4.10)$$

Eq. (4.10) is very important for the implementation of the algorithm because it provides a bound of μ for the stable operation of the algorithm. It is obvious from (4.10) that the choice of μ must be done jointly with the choice of λ for a given filter length N .

Convergence dynamics

Denoting as in chapter 3 by $\Delta W_k = W_k - W^o$ the deviation of the current equalizer setting w.r.t. the optimal setting and subtracting W_o from both sides of the second line of (4.7) we obtain the following recurrence for ΔW_k :

$$\Delta W_{k+1} = (I_N - \mu \mathbf{X}_k R_k^{-1} \mathbf{X}_k^H) \Delta W_k + \mu \mathbf{X}_k R_k^{-1} (D_k - \mathbf{X}_k^H W^o) . \quad (4.11)$$

We conclude from (4.11) that a key quantity related to the convergence dynamics of (4.7) is the following matrix:

$$\mathcal{G}_k = I_N - \mu \mathbf{X}_k R_k^{-1} \mathbf{X}_k^H . \quad (4.12)$$

In order to get insight into the convergence dynamics of (4.7), we examine the eigenvalues of the \mathcal{G}_k . To do this it is interesting to check the influence of the pre-multiplication of \mathcal{G}_k by \mathbf{X}_k :

$$\begin{aligned} \mathbf{X}_k^H (I_N - \mu \mathbf{X}_k R_k^{-1} \mathbf{X}_k^H) &= (I_L - \mu \mathbf{X}_k^H \mathbf{X}_k R_k^{-1}) \mathbf{X}_k^H = \\ (R_k - \mu \mathbf{X}_k^H \mathbf{X}_k) R_k^{-1} \mathbf{X}_k^H &= \mu S_k R_k^{-1} \mathbf{X}_k^H . \end{aligned} \quad (4.13)$$

Eq. (4.13) shows that the eigenvalues $\neq 1$ of \mathcal{G}_k are also the eigenvalues of $\mu S_k R_k^{-1}$. The matrix $\mu S_k R_k^{-1}$ is asymptotically equal to:

$$\begin{aligned} \mu (E S_k) (E R_k)^{-1} &= \mu (1-\lambda) \left[\frac{1}{\mu(1-\lambda)} - N \right] I_L \\ &= (1 - \mu N (1-\lambda)) I_L . \end{aligned} \quad (4.14)$$

Eq. (4.14), even though asymptotical, gives important insight to the convergence behaviour of the algorithm. On one hand it shows that \mathcal{G}_k has asymptotically $N - L$ eigenvalues equal to 1 and L eigenvalues equal to $1 - \mu N (1-\lambda)$. Therefore we see that the algorithm (4.7) has indeed asymptotically a projection interpretation: at each iteration eliminations occur in the subspace of C^N that corresponds to the L non-zero eigenvalues of \mathcal{G}_k . On the other hand, eq. (4.14) reveals the fact that the algorithm's convergence speed is roughly proportional to the quantity:

$$\nu = \mu N (1-\lambda) . \quad (4.15)$$

Therefore ν plays the role of an *effective stepsize* that controls the algorithm's convergence speed. Of course one should be careful to satisfy also (4.10). This gives us enough degrees of freedom to choose the algorithm's parameters so as to provide a high convergence speed preserving the algorithm's stability.

4.2.3 Further discussion

Some additional insight in the role of the effective stepsize defined in (4.15) as well as to the convergence speed of the algorithm (4.7) can be obtained as follows: an essentially equivalent to (4.1) update equation for the sample covariance matrix R_k is the following:

$$R_k = \lambda R_{k-1} + (1-\lambda) N \mathcal{X}_k \mathcal{X}_k^H . \quad (4.16)$$

Let \bar{S}_k be now defined as:

$$\bar{S}_k = \bar{\mu}^{-1} R_k - \mathbf{X}_k^H \mathbf{X}_k , \quad (4.17)$$

where $\bar{\mu}$ is a stepsize parameter (for the moment arbitrary). Taking expectation of both sides of (4.16) we get:

$$E(R_k) = N R , \quad (4.18)$$

which gives combined with (4.17):

$$E(\bar{S}_k) = N R \left(\frac{1}{\bar{\mu}} - 1 \right) . \quad (4.19)$$

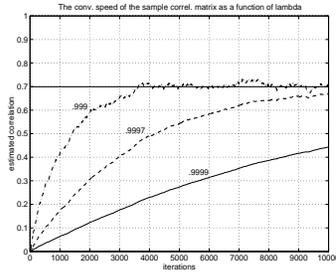
Comparing (4.19) to (4.9) we realize that equations (4.16) and (4.17) represent the same algorithm as equations (4.1) and (4.5), apart from the following *normalization* of μ :

$$\bar{\mu} = \mu N (1-\lambda) . \quad (4.20)$$

In fact, $\bar{\mu}$ is nothing else than the effective stepsize ν we derived before (eq. (4.15))! Eq. (4.19) is an alternative way to derive the region of the stepsize for stability:

$$0 < \bar{\mu} < 2 . \quad (4.21)$$

Therefore eq. (4.16) can be used as well for the analysis of the algorithm's convergence dynamics. In fact, (4.16) reveals the crucial role of the parameter λ in the algorithm's convergence speed: the closest to 1 is λ , the slowest will be the speed with which the sample correlation matrix approaches its true value. This phenomenon can be observed for example in the following figure which shows the evolution of an element of the matrix R_k for different values of λ . An interesting interpretation of this fact is the following intuitive approach for the explanation of the algorithm's initial convergence behaviour: we may expect to have a high initial convergence speed when λ is chosen in such a way so as to allow for a *slow* convergence of the sample covariance matrix to its true value, due to the fact that before this convergence, the sample covariance matrix will have a less good conditioning and provide higher "jumps" at each iteration. Indeed, such a behaviour has been observed in simulations and the above reasoning seems to be a valid explanation.

Figure 4.3: The sensitivity of the convergence speed w.r.t. to λ

4.3 Algorithmic organization

In chapter 3 we described in detail the derivation of the FAP algorithm which is a computationally efficient algorithmic organization for the APA that requires $2N + O(L)$ flops per iteration. In the following subsections we present algorithmic organizations for the algorithm (4.7) based on the philosophy of the FAP. In the first subsection we present an exact algorithmic implementation for (4.7), whereas in the second one we discuss the acquisition of faster algorithmic implementations that correspond to approximations of the algorithm (4.7).

4.3.1 An exact algorithmic organization

The basic update equation we want to implement is:

$$W_{k+1} = W_k + \mu \mathbf{X}_k R_k^{-1} E_k, \quad (4.22)$$

where R_k is defined by (4.1). If we define the quantity l_k as:

$$l_k = R_k^{-1} E_k, \quad (4.23)$$

(instead of $l_k = P_k^{-1} E_k$ defined in chapter 3), then we note that the following identities from the derivation of FAP in chapter 3 can be also used for a fast implementation of (4.7):

- Updating \widehat{W}_k :

$$\widehat{W}_{k+1} = \widehat{W}_k + \mu X_{k-L+1} F_{L-1,k}. \quad (4.24)$$

- Link between W_k and \widehat{W}_k :

$$W_{k+1} = \widehat{W}_{k+1} + \bar{\mu} \bar{\mathbf{X}}_k \bar{F}_k, \quad (4.25)$$

- Updating F_k :

$$F_k = \begin{bmatrix} \bar{F}_k \\ F_{L-1,k} \end{bmatrix}, \quad (4.26)$$

- Updating $E_{0,k}$:

$$\widehat{E}_{0,k} = d_k - X_k^H \widehat{W}_k, \quad (4.27)$$

$$E_{0,k} = \widehat{E}_{0,k} - \mu \tau_k \bar{F}_{k-1}, \quad (4.28)$$

$$\tau_k = \tau_{k-1} + x_k \tilde{X}_k^H - x_{k-N} \tilde{X}_{k-N}^H. \quad (4.29)$$

The updating of the quantities l_k , E_k and R_k^{-1} however has to be modified due to the difference in structure between R_k^{-1} and P_k^{-1} :

- Calculating l_k :

Due to the structure of R_k in (4.1), the vector l_k can no more be efficiently updated as in FAP, and therefore it must be directly computed as in (4.23).

- Updating E_k :

The relation between a priori and a posteriori errors (3.56) is no more valid for (4.7). It is still valid however that:

$$\tilde{E}_k = \bar{\mathcal{E}}_{k-1}. \quad (4.30)$$

Writing down the expression for the a posteriori error at time instant $k-1$ we get:

$$\begin{aligned} \mathcal{E}_{k-1} &= D_{k-1} - \mathbf{X}_{k-1}^H W_k = D_{k-1} - \mathbf{X}_{k-1}^H (W_{k-1} + \mu \mathbf{X}_{k-1} R_{k-1}^{-1} E_{k-1}), \\ \mathcal{E}_{k-1} &= D_{k-1} - \mathbf{X}_{k-1}^H W_{k-1} - \mu \mathbf{X}_{k-1}^H \mathbf{X}_{k-1} R_{k-1}^{-1} E_{k-1} \end{aligned}$$

and therefore (3.56) of chapter 3 is replaced by:

$$E_{k-1} = (I_L - \mu P_{k-1} R_{k-1}^{-1}) E_{k-1}. \quad (4.31)$$

Due to (4.30) we recognize that E_k can be written as:

$$E_k = \begin{bmatrix} E_{0,k} \\ \bar{\mathcal{E}}_{k-1} \end{bmatrix}. \quad (4.32)$$

Combining (4.31) and (4.32) we find the following updating formula for E_k :

$$E_k = \begin{bmatrix} E_{0,k} \\ \mathbf{0}_{(L-1 \times 1)} \end{bmatrix} + \mathcal{Z}_L (E_{k-1} - \mu P_{k-1} l_{k-1}), \quad (4.33)$$

where \mathcal{Z}_L is the following $(L \times L)$ shift operator:

$$\mathcal{Z}_L = \begin{bmatrix} 0 & \cdots & \cdots & \cdots & 0 \\ 1 & 0 & & & \vdots \\ 0 & 1 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix}. \quad (4.34)$$

The sample covariance matrix P_k can be updated by noting that:

$$P_k = \mathbf{X}_k^H \mathbf{X}_k = \mathcal{X}_k \mathcal{X}_k^H + \cdots + \mathcal{X}_{k-N+1} \mathcal{X}_{k-N+1}^H ,$$

in the following way:

$$P_k = P_{k-1} - \mathcal{X}_{k-N} \mathcal{X}_{k-N}^H + \mathcal{X}_k \mathcal{X}_k^H . \quad (4.35)$$

In order to give also an update formula for the inverse covariance matrix R_k^{-1} , we first define the following quantities:

$$C_k = R_k^{-1} \mathcal{X}_k , \quad (4.36)$$

and:

$$\gamma_k = \frac{1}{1 - \mathcal{X}_k^H C_k} . \quad (4.37)$$

C_k is the so-called (in adaptive filtering literature) "direct Kalman gain" and γ_k the "likelihood variable". Then R_k^{-1} is updated as:

$$R_k^{-1} = \frac{1}{\lambda} R_{k-1}^{-1} - \gamma_k C_k C_k^H . \quad (4.38)$$

This concludes the derivation of our algorithm, which can be summarized in table 4.1:

If a prewindowed Stabilized FTF with an exponential sliding window is used in step 1,

An Exponentially Regularized FAP		×
0: Initialization		
1: Use a (prewindowed) Stabilized FTF (prediction part) to update C_k and γ_k		$6L$
2: $\tau_k = \tau_{k-1} + x_k \hat{X}_k^H - x_{k-N} \hat{X}_{k-N}^H$		L
3: $\hat{E}_{0,k} = d_k - X_k^H \hat{W}_k$		N
4: $E_{0,k} = \hat{E}_{0,k} - \mu \tau_k \bar{F}_{k-1}$		L
5: $E_k = \begin{bmatrix} E_{0,k} \\ \mathbf{0}_{(L-1) \times 1} \end{bmatrix} + \mathcal{Z}(E_{k-1} - P_{k-1} \mu l_{k-1})$		$L^2 + L$
6: $\tilde{E}_{0,k} = \hat{E}_{0,k} - \mu \tau_k \bar{F}_{k-1}$		L
7: $R_k^{-1} = \frac{1}{\lambda} R_{k-1}^{-1} - \gamma_k C_k C_k^H$		$2L^2 + L$
8: $l_k = R_k^{-1} E_k$		L^2
9: $F_k = \begin{bmatrix} \mathbf{0} \\ \bar{F}_{k-1} \end{bmatrix} + l_k$		$\mathbf{0}$
10: $P_k = P_{k-1} - \mathcal{X}_{k-N} \mathcal{X}_{k-N}^H + \mathcal{X}_k \mathcal{X}_k^H$		L^2
11: $\hat{W}_{k+1} = \hat{W}_k + \mu X_{k-L+1} F_{L-1,k}$		N

Table 4.1: An exact algorithmic organization for the exponentially regularized FAP

this will take $6L$ multiplications. The corresponding computational complexity for the other steps in terms of multiplications is: L for each of steps 2, 4 and 6, N for each of steps 3 and 11, L^2 for each of steps 8 and 10, $L^2 + L$ for step 5 and $2L^2 + L$ for step 7. This

gives an overall complexity of $2N + 5L^2 + 11L$ for an algorithm that corresponds exactly to the criterion (4.7). The $O(L^2)$ term represents the price paid for implementing exactly an algorithm that uses a regularized covariance matrix, without any approximation. On the contrary, the FAP in [Gay93] has a complexity of $2N + 20L$ (numerically stable version) but corresponds to an algorithm with a non-regularized sample covariance matrix $R_k = \mathbf{X}_k^H \mathbf{X}_k$ and is only approximative when some sort of regularization is introduced (it is similarly possible however to achieve a lower complexity for (4.7) if approximations are introduced). However, as usually L is chosen to be significantly smaller than N (especially in acoustic echo cancellation problems), our algorithm's complexity is still comparable to that of FAP. For example, in a blind equalization case with $N = 30$ and $L = 5$, our algorithm will have a complexity of 240 multiplications/iteration whereas FAP has a complexity of 160 multiplications/iteration.

4.3.2 Approximative algorithmic implementations

Faster computational organizations than the the one given above can be achieved if one introduces some approximations. As the regularization part of the problem resides basically in the use of an exponential sliding window for the sample covariance matrix R_k , one could think of modifying FAP by changing only its FTF prediction part, namely by using an exponential instead of a rectangular sliding window. If the rest of FAP remains unchanged, then the corresponding algorithm will no longer correspond of course to the criterion (4.4),(4.5)⁵. However it will certainly have a better performance than the (rectangular window) FAP, due to the better behaviour of the sample covariance matrix in terms of eigenvalue spread. The resulting algorithm is described in table 4.2.

4.4 NSWERCMA: a counterpart of the proposed class of algorithms for BE

The class of adaptive filtering algorithms (4.7) can be modified in such a way so as to be useful for blind equalization. This can be done by direct application of the separation principle we presented in chapter 3, to the algorithm developed in the previous paragraphs. Namely, a suitable class of algorithms for blind equalization can be obtained from (4.7) if one replaces the desired response vector D_k by the projection of the a priori output vector D_k on the unit circle:

$$D_k \equiv \text{sign}(\mathbf{X}_k^H W_k) . \quad (4.39)$$

⁵Such an approach for the acquisition of a fast algorithm has been proposed in [MD94]

An approximative exponentially regularized FAP		x
0:	Initialization: $a_0 = [1 \ 0_{(L-1) \times 1}]^T$, $b_0 = [0_{(L-1) \times 1} \ 1]^T$, $E_{a,0} = E_{b,0} = \delta$ (a very small number)	
1:	Use an (exp. windowed) Stabilized FTF (prediction part) to update $a_k, b_k, E_{a,k}, E_{b,k}$	$6L$
2:	$\tau_k = \tau_{k-1} + x_k \hat{X}_k^H - x_{k-N} \hat{X}_{k-N}^H$	$2L$
3:	$\hat{E}_{0,k} = d_k - X_k^H \hat{W}_k$	N
4:	$E_{0,k} = \hat{E}_{0,k} - \mu \tau_k \bar{F}_{k-1}$	L
5:	$E_k = \begin{bmatrix} E_{0,k} \\ (1-\mu) \bar{E}_{k-1} \end{bmatrix}$	L
6:	$l_k = \begin{bmatrix} 0 \\ \bar{l}_k \end{bmatrix} + \frac{1}{E_{a,k}} a_k a_k^H E_k$	$2L$
7:	$\bar{l}_k = l_k - \frac{1}{E_{b,k}} b_k b_k^H E_k$	$2L$
8:	$F_k = \begin{bmatrix} 0 \\ \bar{F}_{k-1} \end{bmatrix} + l_k$	L
9:	$\hat{W}_{k+1} = \hat{W}_k + \mu X_{k-L+1} F_{L-1,k}$	N
10:	$\bar{l}_{k+1} = (1-\mu) \bar{l}_k$	L

Table 4.2: An algorithmic organization for an approximative regularized FAP

The resulting algorithm will then have the form:

$$\begin{aligned} R_k &= \lambda R_{k-1} + X_k X_k^H \\ W_{k+1} &= W_k + \mu X_k R_k^{-1} (\text{sign}(X_k^H W_k) - X_k^H W_k). \end{aligned} \quad (4.40)$$

In order to implement (4.40) in a computationally efficient way we can use the two algorithmic organizations presented above for (4.7). Note that these algorithms dictate a “shift invariance” structure of the desired signal. Due to the inherent exponential regularization, we give this class of algorithms the name of *Normalized Sliding Window Exponentially Regularized Constant Modulus Algorithms: NSWERCMA*.

It is expected that this class of algorithms for blind equalization, will be able to provide, with a judicious choice of its parameters a very good compromise of convergence speed, steady state error, and local minima avoidance. Before entering the computer simulations section, it would be interesting to show how the algorithm outperforms the NSWCMA in the example presented in section 4.2. Figure 4.4 shows the closed-eye measure evolution for the same example corresponding to figure 4.1. Note how, even with the “pushed” choice $L = N = 5$, the algorithm is capable of converging to an acceptable open-eye setting without deviating from it, contrarily to the NSWCMA. We remind that in this example $\lambda = 0.9$.

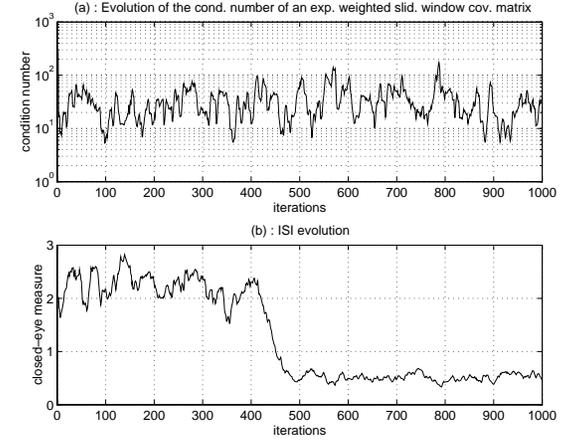


Figure 4.4: The improvement in performance using an exponential sliding window

In the next section we present some comparative computer simulations that show how the algorithms of the class (4.40) are able of outperforming other blind equalization algorithms in several cases.

4.5 Computer Simulations

The following computer simulations test the proposed class of algorithms in the blind equalization context of this thesis:

4.5.1 Opening the channel eye

We simulated the algorithm (4.40) in the following blind-equalization setup: a 16-QAM constellation is transmitted through a linear FIR channel. The received signal is sampled at the baud rate and the discrete-time channel response is $[1 \ -3 \ -3 \ 2]$. Additive noise resulting in an SNR of 30 dB is added to the received signal. Then the algorithm (4.40) is employed using $N = 5$, $L = 4$, $\mu = 0.1$, $\lambda = 0.99$ and initialized with $W_0 = [0 \ 0 \ 1 \ 0 \ 0]^T$. The evolution of the system closed-eye measure is shown in figure 4.5, where it can be seen that the channel eye opens after approximately 500 iterations. Note in this figure the perturbation of the received data due to the channel ISI and how it is reduced after the convergence of the blind equalizer. Note also the fact that the algorithm works for a 16-QAM constellation, despite the fact that its criterion is of the constant modulus type. This is a typical characteristic of algorithms of the constant-modulus type and can be explained in the same way as Shalvi

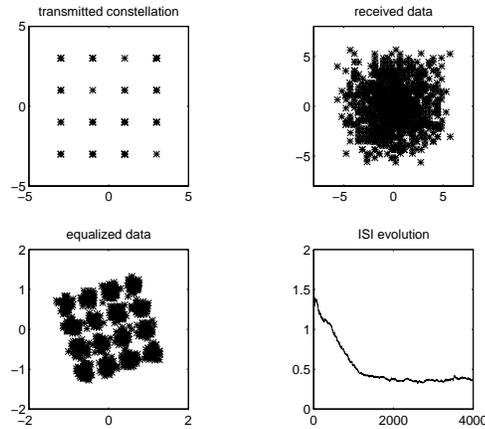


Figure 4.5: The opening of the channel-eye achieved by a member of the proposed class of algorithms

and Weinstein showed that the CMA 2-2 is able of opening the channel eye when the input is *any* sub-Gaussian signal.

4.5.2 Convergence speed

In order to test the improvement of the proposed algorithm in terms of convergence speed, we performed a comparative computer simulation experiment that implements several constant-modulus-type blind equalizers. We transmitted a 2-PAM random signal through a FIR(2) channel with impulse response $h_c = [1 \ 2 \ 0.6]$ and a white Gaussian noise η_k is added to the received signal resulting to an SNR of 20 dB. We then run several adaptive blind algorithms (according to figure 3.1) to update the equalizer. Figure 4.6 shows the evolution of the system's closed-eye measure ρ averaged over 100 Monte-Carlo simulations for four different blind equalization algorithms. The CMA 2-2 is employed with a stepsize $\mu = 0.04$, the NCMA (NSWCMA with $(L = 1)$) with $\bar{\mu} = 0.3$, the algorithm proposed in [GRS87] (that we call RLS-CMA) with a forgetting factor $\lambda = 0.94$ and our proposed algorithm ((4.7),(6.17)) with $L = 6$, $\lambda = 0.5$ and $\mu = 0.01$. The equalizer's length is equal to 6 and all algorithms are initialized with $W_0 = 10^{-4} \times [1 \ 1 \ 1 \ 1 \ 1 \ 1]^T$. It can be seen that our proposed algorithm has an increased convergence speed and opens the channel's eye faster than the other algorithms. It is also noted that it behaves well in steady-state, despite the small value for the forgetting factor it uses. This means that even such a small value regularizes adequately the received signal's sample covariance matrix. Moreover, it is this small value for λ that corresponds to

a big effective stepsize and provides a high convergence speed. The considerably fast initial convergence of our proposed algorithm provides also evidence for the above-discussed fact that the algorithm is expected to go very fast at the beginning when the sample covariance matrix has not yet reached its steady-state value.

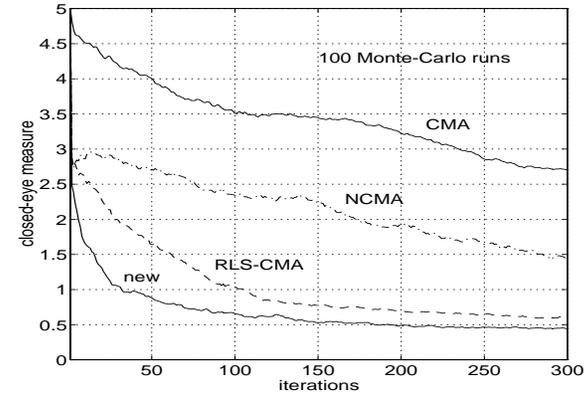


Figure 4.6: Comparative study of several BE algorithms in terms of convergence speed

4.5.3 Local minima

Figure 4.7 shows a case where the problem of false minima is encountered. The transmission channel's impulse response is now $h_c = [1 \ 0.6 \ 0.36]$ and a 3-tap equalizer is used, initialized at $[0 \ 0 \ 1]^T$. This is a typical example where the CMA's get trapped by a local minimum of their cost function, being unable to open the system's eye. The RLS-CMA is used with a forgetting factor $\lambda = 0.95$ and then our algorithm (4.40) with $L = 3$, $\lambda = 0.95$ and $\mu = 1$. It can be seen how the first algorithm is indeed trapped by a local minimum and does not open the eye, whereas the second one is able of converging to a setting that opens the eye. This fact reflects the potential advantage of the algorithms of the proposed class that permit the use of a big effective stepsize that guarantees stability on one hand but also provides a large enough motion around false minima that helps escaping from them according to the corresponding discussion and analysis that we presented in chapter 3.

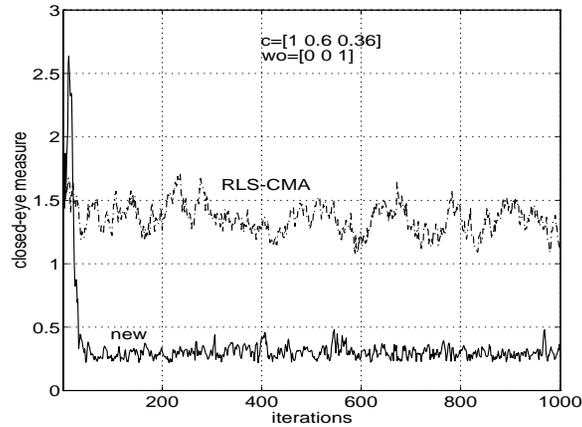


Figure 4.7: Escaping from a local minimum of the cost function

4.6 Conclusions

In this chapter we have proposed a new class of adaptive filtering algorithms for blind equalization of constant modulus signals. Our motivation has been the high steady-state error sometimes present in the APA and NSWCMA algorithms when big values for $\bar{\mu}$ and for L are used. The proposed remedy has been to use a more regularized sample-covariance matrix using exponential forgetting as shown in (4.7). The algorithms have been derived by minimizing exactly at each iteration a deterministic criterion, first in a classical adaptive-filtering and then in a blind-equalization context. An asymptotic analysis has given useful insight to the algorithm's convergence behaviour, has revealed the role of an *effective stepsize* and has provided parameter bounds that guarantee stability. Based on the algorithmic organization of the FAP given in chapter 3, we propose an algorithmic organization that corresponds exactly to our criterion and provides a complexity of $2N + 6L^2 + 10L$ multiplications/iteration. We also discuss how faster algorithms can be obtained by introducing approximations. The algorithms' behaviour has been also tested via computer simulations in the blind equalization context. These simulations show their ability of opening the channel eye, their increased convergence speed w.r.t. other constant-modulus algorithms as well as their ability to escape from false minima of their cost function. Moreover, they show the improvement achieved in the steady-state behaviour for cases when L is chosen to be close to N . A part of the work contained in this chapter has been also presented in [PS94a].

Chapter 5

Decision-directed blind equalization

DECISION -Directed (DD) equalization is probably the most primitive blind equalization method for the cancelling of Inter-Symbol-Interference in data communication systems. Even though DD equalizers are believed to be unable of opening the channel eye when it is initially closed, this does not seem to be true in the case of Constant-Modulus (CM) constellations (pure phase modulation). We investigate the shape of the DDA cost function in this case and obtain several interesting results that indicate that the DDA should be capable of opening a closed channel eye in the CM case. Based on this fact, we propose a novel hybrid CMA-DD equalization scheme that offers an appealing alternative to the Generalized Sato (GSA) algorithm for QAM constellations. Our theoretical claims about the performance of DD equalizers as well as the performance of our novel scheme are verified through computer simulations.

5.1 Introduction

Figure 5.1 shows the three different standard modes that may be used at a communication system receiver for the equalization of the received signal. Namely, the position 1 of the switch corresponds to the blind equalization mode already discussed, in which only the received signal's statistics are used in order to update the equalizer, the position 2 corresponds to Decision-Directed (DD) equalization, whereas position 3 corresponds to equalization based on a training-sequence signal (non-blind equalization). Typically, during an equalization procedure more than one of these equalization modes may alternate, corresponding to different equalization stages. For example, after using mode 3, when the equalizer has converged to an acceptable setting with the help of the training sequence, the receiver is usually then switched to the DD mode (2): since the equalizer output is supposed to approximate well the transmitted sequence, the decision device output at a specific time instant will match the transmitted

sequence with a high probability and can therefore be used as a “training sequence”. The same switching can be used in the case of a blind method: after convergence to an acceptable equalizer setting in the mode 1, there is no use any more to update the equalizer with the blind algorithm, since the DD mode can be used to keep the equalizer to a correct setting. Moreover, the DD mode has the advantage of providing a lower steady state error, due to the fact that the decision device output at most time instants is *exactly* equal to, and not an approximation of the corresponding transmitted symbols. From the above we conclude that

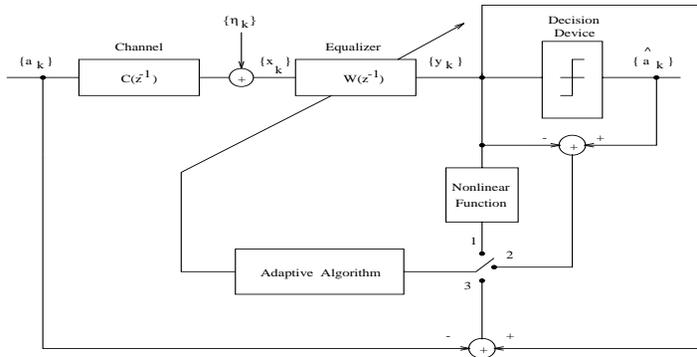


Figure 5.1: Three standard equalization modes

DD equalization is usually considered as an auxiliary equalization mode which is used in the final stage of a conventional or a blind equalization procedure. Of course, strictly speaking, DD equalization is also blind, since it does not make use of a training sequence. However it is not used as a BE procedure, since practice has shown that it is in general not successful in opening an initially-closed channel eye. Based on such remarks (see e.g. [Sat75], [God80]) as well as on theoretical studies on the performance of DD equalizers ([Maz80], [ME84]), the predominant feeling seems to be that “Decision - Directed equalizers are unable of opening an initially closed channel eye”. It is actually this feeling that has lead to attempts for hybrid algorithms that combine other BE techniques with DD equalization (see e.g. [PP87], [HD93]).

In this chapter of the thesis we will take a closer look to DD blind equalization, by restraining ourselves to the special case of a Constant Modulus (CM) transmitted signal. As will become clear in the sequel, there exist both theoretical arguments and experimental evidence that the above statement is not necessarily true in this case.

The rest of the chapter is organized as follows. In the following section we show the similarity between the CMA 1-2 and the DDA and in section 5.3 we analyze the shape of the CMA 1-2 cost function in the case of a CM input signal. The non-CM case is also considered in section 5.3. In section 5.4 we show how different classes of DD algorithms can be obtained

in the same way that several CM-BE algorithms were obtained for CM-type BE from the separation principle. In section 5.5 we present an alternative scheme for the generalization of the Sato algorithm in the complex case. The computer simulation results of this chapter that provide experimental evidence to the theoretical parts are contained in section 5.6. Finally, section 5.7 contains our conclusions and further discussion.

5.2 Motivation

Consider the well known LMS-like DD algorithm (DDA) whose update equation is given by

$$W_{k+1} = W_k + \mu X_k (dec(y_k) - y_k) , \quad (5.1)$$

where $dec(y)$ denotes the closest constellation symbol to y . Comparing (5.1) with the CMA 1-2 algorithm which can be written as we saw in the following way

$$W_{k+1} = W_k + \mu X_k (r_1 sign(y_k) - y_k) , \quad (5.2)$$

where $sign(\cdot)$ and r_1 have been defined in chapter 2, we note that these two algorithms are of the same exactly form: they look like classical LMS updates, the only difference being that, in the absence of the desired response $\{a_i\}$, each one of them uses a different “guess” for it: the DDA uses $dec(y_k)$ instead of a_k , whereas the CMA uses $r_1 sign(y_k)$. This results to different *error signals* for the updating of the equalizer as used by the two different algorithms:

$$\begin{cases} e_k^D = y_k - dec(y_k) \\ e_k^C = y_k - r_1 sign(y_k) \end{cases} . \quad (5.3)$$

The difference between these two error signals increases with the distance of the projection $r_1 sign(y_k)$ from its closest constellation symbol. For a CM constellation, the two different guesses for the desired samples have the same magnitude and differ only by their angle. Figure 5.2 shows the error signals of the two algorithms for the case of a 4-QAM constellation (fat-line segments). If one thinks of complex CM constellations with more symbols equispaced on a circle of radius r_1 , it is clear that the more the symbols, the less becomes the difference between the two error signals. Moreover, in the asymptotic case of an infinity of symbols uniformly distributed on the circle, the two error signals become identical, since each projection on the circle corresponds also to a symbol! Therefore we can state the following remark:

Remark: The CMA 1-2 and the DD algorithms are identical in the asymptotic case of an input signal of CM equal to r_1 whose angle is uniformly distributed in $[0, 2\pi)$.

The case of such an input signal of course is not realistic, however the above remark implies that in the realistic case of finite-alphabet CM constellations, the two algorithms may have comparable performance. This gives rise to the following question: since the CMA

algorithms (and particularly the CMA 1-2) are able in general of opening an initially closed channel eye, why shouldn't the DDA be able of doing the same in the case of a CM signal?

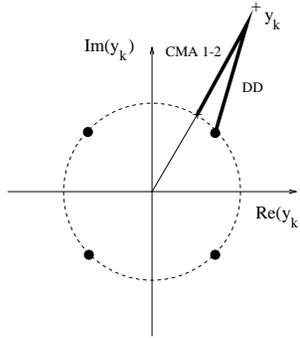


Figure 5.2: The similarity between CMA 1-2 and DDA

Now strictly speaking, as already mentioned in chapter 2, the strongest existing result regarding the performance of Godard equalizers is the one proven in [SW90] by Shalvi and Weinstein: in the case $p = 2$ the Godard cost function (2.76) is a convex function of s whose global minima are optimal settings of the form

$$s = e^{j\theta}(\cdots 0 \ 1 \ 0 \ \cdots) , \quad (5.4)$$

where s is the impulse response corresponding to the cascade of channel and equalizer

$$s_i = \sum_{l=-\infty}^{\infty} c_l w_{i-l} . \quad (5.5)$$

This result explains why the popular CMA 2-2 algorithm [TA83], which corresponds to the Godard algorithm that we repeat here for convenience:

$$W_{k+1} = W_k + \mu X_k y_k |y_k|^{p-2} (|y_k|^p - r_p) , \quad (5.6)$$

for the choice $p = 2$, has an optimal performance¹ in the case of a noiseless channel and an infinite-length equalizer, even if $\{a_i\}$ is not drawn from a CM constellation, provided that it is sub-Gaussian and symmetrical.

No similar result however seems to exist (up to our knowledge) for the CMA 1-2 algorithm (the Godard algorithm corresponding to the choice $p = 1$) in the complex case, namely there is no proof for its optimality, even under assumptions of infinite-length equalizers and specific input constellations. However, as mentioned in chapter 1, the CMA 1-2 in the real case reduces to the Sato algorithm, whose update equation is identical to (5.6) for $p = 1$, the only

¹By optimal performance here we mean that its only local minima are of the form (5.4).

difference being that all the quantities are real. As mentioned, there exists a strong result (given by Benveniste *et al* in [BG83], [BG84]) for the performance of the Sato algorithm: it is stated that in the case of a continuous sub-Gaussian input distribution, a noiseless channel and an infinite equalizer length, the Sato cost function admits as only local (and global) minima the optimal settings for s given in (5.4). This result is less strong than the one given for the CMA 2-2 in [SW90], since it is only valid for continuous (and not for discrete) input distributions. The result has also been extended to the case of the Generalized Sato algorithm, by a simple superposition principle, due to the "separability" of the real and imaginary parts of the desired signal in this case. However, this kind of analysis has not allowed to obtain similar results for cases as the one of the CMA 1-2.

As in the real case the assumption of a continuous distribution has allowed to prove the optimality of the Sato algorithm, and as the true counterpart of the Sato in the complex case is the CMA 1-2, one may hope that a similar assumption can help obtain such a result for the CMA 1-2.

Putting together the aforementioned facts and remarks, we have good chance of thinking that the DDA can be able of opening the channel eye in the case of a CM signal, at least in the asymptotic case mentioned above. This is why we will perform an analysis of the CMA 1-2 algorithm for this case.

5.3 Analysis of the stationary points of the DDA cost function

We will be interested in the stationary points of the DDA cost function, which will be parameterized in the s -domain (see (5.5)). The equalizer output at time instant k is then expressed as

$$y_k = \sum_{i=-\infty}^{\infty} s_i a_{k-i} . \quad (5.7)$$

We begin by considering the CM case whose special interest was motivated in the above section.

5.3.1 The CM case

As our analysis is based on the similarity of the CMA 1-2 to the DDA in the case of a CM signal, we begin our analysis by considering the CMA 1-2 algorithm.

Analysis of the CMA 1-2 cost function

In the sequel we make the following assumptions:

Assumption A.5.1:

- The input $\{a_i\}$ is an i.i.d. CM sequence

- The equalizer is infinite-length
- No additive noise is present

Assuming for simplicity that the constellation modulus equals 1 (and therefore $r_1 = 1$ also), the transmitted symbols take the form

$$a_k = e^{j\phi_k} , \quad (5.8)$$

and the CMA 1-2 cost function is given by

$$J(s) = E(|y_k| - 1)^2 \quad (> 0) . \quad (5.9)$$

The equilibria of the CMA 1-2 cost function are found by setting its first partial derivative w.r.t. each element of s equal to zero:

$$\frac{\partial J}{\partial s_k^*} = 2E \left(\left| \sum_k a_{n-k} s_k \right| - 1 \right) \frac{\partial \left| \sum_k a_{n-k} s_k \right|}{\partial s_k^*} = 0 . \quad (5.10)$$

Noting that

$$\left| \sum_k a_{n-k} s_k \right| = \sqrt{\left(\sum_k a_{n-k} s_k \right) \left(\sum_k a_{n-k} s_k \right)^*} ,$$

it turns out that

$$\frac{\partial \left| \sum_k a_{n-k} s_k \right|}{\partial s_k} = \frac{1}{2} \frac{a_{n-k} \sum_k a_{n-k}^* s_k^*}{\sqrt{\left(\sum_k a_{n-k} s_k \right) \left(\sum_k a_{n-k} s_k \right)^*}} ,$$

which gives when combined with (5.10)

$$\frac{\partial J}{\partial s_k^*} = E \left(\left| \sum_k a_{n-k} s_k \right| - 1 \right) \frac{a_{n-k} \sum_i a_{n-i} s_i}{\left| \sum_i a_{n-i} s_i \right|} = 0 ,$$

and therefore

$$E \left(1 - \frac{1}{\left| \sum_i a_{n-i} s_i \right|} \right) a_{n-k}^* \sum_i a_{n-i} s_i = 0, \quad \forall k. \quad (5.11)$$

which gives

$$s_k = E \left(a_{n-k}^* \operatorname{sign} \left(\sum_i a_{n-i} s_i \right) \right) . \quad (5.12)$$

As s_k appears in both parts of (5.12) and the specific structure of the equation renders difficult the extraction of an analytical expression for s_k . Things can get much easier if we restrict the settings of stationary point to be such that all the components of each stationary points have equal magnitude. This means that the stationary points we are interested in, are all settings $\{s_i\}$ that contain M non-zero elements of equal magnitude, α_M , $M = 1, 2, \dots$:

$$s_M = \alpha_M [\dots 0 e^{j\theta_1} 0 \dots 0 e^{j\theta_2} 0 \dots 0 e^{j\theta_M} 0 \dots] . \quad (5.13)$$

Taking into account the form (5.13) for the stationary points, (5.12) gives

$$\begin{aligned} \sum_{i=1}^M s_i E a_{k-1}^* a_{k-i} &= E \frac{\sum_{i=1}^M s_i a_{k-1}^* a_{k-i}}{\left| \sum_{i=1}^M s_i a_{k-i} \right|} , \\ s_l E (|a_k|^2) &= E \frac{\sum_{i=1}^M s_i a_{k-1}^* a_{k-i}}{\left| \sum_{i=1}^M s_i a_{k-i} \right|} , \\ s_l &= s_l E \frac{\sum_{i=1}^M a_{k-1}^* a_{k-i}}{\left| s_l \right| \left| \sum_{i=1}^M a_{k-i} \right|} , \end{aligned}$$

and therefore we have the following expression for the magnitudes α_M

$$\alpha_M = E \left(\frac{a_k^* \sum_{m=0}^{M-1} a_{k-m}}{\left| \sum_{m=0}^{M-1} a_{k-m} \right|} \right) . \quad (5.14)$$

This expression is identical to the one given by Godard in [God80]². A simpler expression for the amplitude α_M can be found as follows. Consider the expression of the cost function on a stationary point of the form (5.13):

$$J(\alpha_M) = M \alpha_M^2 - 2M E \left(\sum_{i \in \mathcal{I}_M} a_{n-i} \right) + 1 , \quad (5.15)$$

where \mathcal{I}_M is defined as

$$\mathcal{I}_M = 1, 2, \dots, M , \quad (5.16)$$

and we have used the fact that on a stationary point of the form (5.13)

$$E(|y(s_M)|^2) = E \left(\left| \sum a_{n-i} s_i \right|^2 \right) = M \alpha_M^2 . \quad (5.17)$$

Now since the expression $J(\alpha_M)$ given in (5.15) corresponds to a stationary point, its first derivative around it with respect to α_M should be zero:

$$\frac{\partial J}{\partial s_k} = 0 , \quad 2M \alpha_M - 2E \left| \sum_{i=1}^M a_i \right| = 0 ,$$

yielding

$$\alpha_M = \frac{\gamma_M}{M} , \quad (5.18)$$

²Even though in [God80] it is stated that (5.13), (5.14) describe all the possible sets of stationary points for the CMA 1-2, it is possible that there may be other ones as well. However these are the only ones for which we are able of deriving an analytical expression, and we will focus our attention on them. For a more complete discussion on this refer to the Appendix 5.A.

where γ_M is defined as

$$\gamma_M = E \left| \sum_{i=1}^M a_i \right| . \quad (5.19)$$

The expression (5.18) is simpler than the one in (5.14) and will be useful in the rest of our analysis. An upper bound for γ_M can be found as follows: we express (5.19) as

$$\gamma_M = E \left(\sum_{i=1}^M a_i \right) \operatorname{sign} \left(\sum_{i=1}^M a_i \right) . \quad (5.20)$$

By applying the Cauchy-Schwarz inequality to (5.20) we obtain

$$E \left(\sum_{i=1}^M a_i \right) \operatorname{sign} \left(\sum_{i=1}^M a_i \right) \leq \sqrt{E \left| \sum_{i=1}^M a_i \right|^2} \sqrt{E \mathbf{1}} = \sqrt{M} ,$$

which gives for γ_M :

$$\gamma_M \leq \sqrt{M} . \quad (5.21)$$

Combining (5.21) and (5.18) we obtain also an upper bound for α_M :

$$\alpha_M \leq \frac{1}{\sqrt{M}} . \quad (5.22)$$

We are now interested in examining the form of these stationary points, namely if they correspond to local minima, maxima or saddle points. At this point we discriminate between the following cases:

- $M = 1$

In this case $\alpha_1 = 1$ and therefore $|y_k| = 1$, for all k . Consequently, $E(|y_k| - 1)^2 = 0$ in this case. As $E(|y_k| - 1)^2$ is a nonnegative quantity, we conclude that all the stationary points for $M = 1$ correspond to global minima of the CMA 1-2 cost function. These minima are given by $s = e^{j\theta}$ where θ is an arbitrary angle and all correspond to Zero-Forcing (ZF) equalizers. The fact that they rotate the constellation by a constant but unknown factor is a problem frequently encountered in BE and can be overcome by using differential encoding.

- $M \geq 2$

The two stochastic quantities of the CMA 1-2 cost function are $E|y_k|^2$ and $E|y_k|$. Keeping (5.19) in mind and using the i.i.d. assumption, these quantities, evaluated at a stationary point of the form of (5.13) are given by

$$\begin{aligned} E|y_k|^2 &= M \alpha_M^2 \\ E|y_k| &= \alpha_M \gamma_M = M \alpha_M^2 . \end{aligned} \quad (5.23)$$

A practical way to check the form of the cost function around a stationary point is to find counterexamples that exclude the possibility of a specific form of the stationary point. This is

usually done by considering proper perturbations around the stationary points. In this scope, we consider the following perturbation of s_M

$$s_M^\epsilon = \sqrt{1 + \epsilon} \alpha_M [\dots e^{j\theta_1} \dots e^{j\theta_2} \dots e^{j\theta_M} \dots] , \quad (5.24)$$

where ϵ is a very small positive number. We now denote by y_k^+ and J^+ the output sample and the value of the cost function on the point s_M^ϵ , respectively: and define

$$\begin{aligned} y_k^+ &= \sum_{i=-\infty}^{\infty} a_{k-i} s_{M,i}^\epsilon \\ J^+ &= E(|y_k^+| - 1)^2 , \end{aligned} \quad (5.25)$$

where $s_{M,i}^\epsilon$ denotes the i^{th} element of s_M^ϵ . We also denote by J° the value of the cost function on the stationary point s_M :

$$J^\circ = E(|y_k| - 1)^2 . \quad (5.26)$$

Then we have

$$J^+ - J^\circ = M \alpha_M^2 (2 + \epsilon - 2\sqrt{1 + \epsilon}) > 0 . \quad (5.27)$$

This means that in a neighborhood around the stationary point s_M and in the direction defined by (5.24), the cost function takes always a greater value than on the stationary point itself. This means that this stationary point is not a local maximum. Since this proof is valid for any integer M , we can conclude the following lemma:

Lemma 5.1: Under the assumptions (A.5.1) the CMA 1-2 has no local maxima points of the form (5.13).

We are now interested in the existence of local minima of the cost function (5.9). We have not been successful in finding such perturbations that prove the absence of local minima for any M ³. We therefore consider the Hessian of the cost function around the stationary points. To do this we have to calculate the following second partial derivative of $J(y)$ around a stationary point of the form (5.13):

$$\frac{\partial^2 J}{\partial s_k^* \partial s_1} .$$

According to the expression for the first partial derivative obtained in (5.11) we have

³The reasons for this will become clear later on.

$$\begin{aligned} \frac{\partial^2 J}{\partial s_k^* \partial s_l} &= E \left(\frac{1}{2} \frac{a_{n-l} (\sum_i a_{n-i}^* s_i^*)}{|\sum_i a_{n-i} s_i|^3} a_{n-k}^* (\sum_i a_{n-i} s_i) + \left(1 - \frac{1}{|\sum_i a_{n-i} s_i|} \right) a_{n-k}^* a_{n-l} \right), \\ &= E \left(\frac{1}{2} \frac{a_{n-l} a_{n-k}^*}{|\sum_i a_{n-i} s_i|} + \left(1 - \frac{1}{|\sum_i a_{n-i} s_i|} \right) a_{n-k}^* a_{n-l} \right), \\ &= E \left(\left(1 - \frac{1/2}{|\sum_{i \in \mathcal{I}_M} a_{n-i} s_i|} \right) a_{n-k}^* a_{n-l} \right). \end{aligned}$$

Therefore we have the following expression for the (k, l) element of the Hessian

$$\frac{\partial^2 J}{\partial s_k^* \partial s_l} = \begin{cases} 1 - \frac{1}{2\alpha_M} E \frac{a_k^* a_l}{\sum_{i=1}^M a_i}, & k, l \in \{1, \dots, M\} \\ \delta_{kl}, & \text{else} \end{cases} \quad (5.28)$$

Now for the Hessian matrix \mathcal{H}_M we have the following properties:

- $\mathcal{H}_M = \mathcal{H}_M^H$, by definition
- $\mathcal{H}_{k,l} = \begin{cases} c_1 & \text{for } k \neq l \\ c_2 & \text{for } k = l \end{cases}$,

where c_1, c_2 represent two different scalar constants. From the two above properties we conclude that \mathcal{H}_M is real, and therefore

$$\mathcal{H}_M = \mathcal{H}_M^T. \quad (5.29)$$

\mathcal{H}_M can then be expressed in the following form

$$\mathcal{H}_M = \alpha I_M + \beta \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}^T, \quad (5.30)$$

where α and β are constant scalars. According to (5.30), the eigenvalues of \mathcal{H}_M are

$$\begin{cases} \lambda_1 = \alpha + M\beta \\ \lambda_2 = \dots = \lambda_M = \alpha \end{cases}$$

From (5.28) we have for the diagonal elements of $\mathcal{H}_{k,k}$:

$$\mathcal{H}_{k,k} = 1 - \frac{M}{2} \frac{1}{\gamma_M} E \frac{1}{\sum_{i=1}^M a_i} = \alpha + \beta. \quad (5.31)$$

and

$$\beta = \frac{M}{2} \frac{1}{\gamma_M} E \frac{a_1^* a_2}{|\sum_{i=1}^M a_i|}. \quad (5.32)$$

On the other hand we have

$$\begin{aligned} [1 \dots 1] \mathcal{H}_M \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} &= M(\alpha + M\beta) = M\lambda_1 \\ &= M - \frac{1}{2\alpha_M} E \left(\frac{\sum_{k=1}^M \sum_{l=1}^M a_k^* a_l}{|\sum_{i=1}^M a_i|} \right) = M - \frac{1}{2\alpha_M} \gamma_M = \frac{M}{2}, \end{aligned}$$

which gives

$$\lambda_1 = \alpha + M\beta = \frac{1}{2}. \quad (5.33)$$

Combining (5.33) and (5.32) we get

$$\alpha = \frac{M - \frac{1}{2}}{M - 1} - \frac{M^2}{2(M - 1)} \frac{1}{E |\sum_{i=1}^M a_i|} E \frac{1}{|\sum_{i=1}^M a_i|}. \quad (5.34)$$

To summarize, the eigenvalues of \mathcal{H}_M are

$$\begin{cases} \lambda_1 = \frac{1}{2} \\ \lambda_2 = \frac{M - \frac{1}{2}}{M - 1} - \frac{M^2}{2(M - 1)} \frac{1}{E |\sum_{i=1}^M a_i|} E \frac{1}{|\sum_{i=1}^M a_i|} \end{cases}. \quad (5.35)$$

A bound for λ_2 can be obtained if we assume for a moment a continuous uniform distribution of the angle ϕ_k of a transmitted symbol in $[0, 2\pi)$. Then we have the following inequalities

$$\begin{aligned} E \left(\frac{1}{|e^{j\phi_1} + \dots + e^{j\phi_M}|} \right) &= \\ \int_0^{2\pi} \dots \int_0^{2\pi} \frac{1}{|e^{j\phi_1} + \dots + e^{j\phi_M}|} \left(\frac{1}{2\pi} \right)^M d\phi_1 \dots d\phi_M &> \left(\frac{1}{2\pi} \right)^M \frac{(2\pi)^M}{M} = \frac{1}{M}, \\ \frac{1}{E |e^{j\phi_1} + \dots + e^{j\phi_M}|} &= \frac{1}{\int_0^{2\pi} \dots \int_0^{2\pi} |e^{j\phi_1} + \dots + e^{j\phi_M}| \left(\frac{1}{2\pi} \right)^M d\phi_1 \dots d\phi_M} > \frac{1}{M}, \end{aligned}$$

which give when combined with (5.35)

$$\lambda_2 < \frac{M - \frac{1}{2}}{M - 1} - \frac{M^2}{2(M - 1)} \frac{1}{M^2},$$

and therefore

$$\lambda_2 < 1 . \quad (5.36)$$

Let us summarize. We have up to now for the eigenvalues of \mathcal{H}_M

$$\begin{cases} \lambda_1 = \frac{1}{2} \\ \lambda_2 < 1 . \end{cases} \quad (5.37)$$

In a first place, the fact that the eigenvalues are not all simultaneously negative (guaranteed by the positiveness of λ_1) means that a stationary point cannot be a local maximum, thus verifying the lemma 5.1 stated above. On the other hand, the information given in (5.37) is not enough to further characterize the shape of the stationary points (5.13). This would need a certainty about the sign of λ_2 : if λ_2 was guaranteed to be positive, for example, this would imply that all the stationary points are local minima, whereas if it was guaranteed to be negative that would imply that all the stationary points were saddle points. In order to take a closer look to the sign of λ_2 we have to examine separately the following two cases:

- M is even

This case has the following particularity: consider the following among the terms appearing in the expression for λ_2 given in (5.35):

$$E \left(\frac{1}{|\sum_{i=1}^M a_i|} \right) .$$

For any discrete CM constellation whose alphabet size is L , the above expectation can be computed as a sum of terms corresponding to all different (L^M) possibilities for the M transmitted symbols:

$$E \left(\frac{1}{|\sum_{i=1}^M a_i|} \right) = \frac{1}{L^M} \sum_{j=1}^{L^M} \frac{1}{|\sum_{i=1}^M a_i^{(j)}|} ,$$

where superscript (j) accounts for the j^{th} among the L^M scenarios. For any symmetrical CM constellation (meaning that for each constellation symbol its opposite is also a constellation symbol), there will be some among the L^M terms appearing in the above expression whose denominator equals 0, due to the fact that M is even. These terms correspond to singularities, (they have contributions of the form $\frac{1}{0}$ in the expectation). This means that in this case,

$$\lambda_2 = -\infty . \quad (5.38)$$

As λ_1 is positive, it turns out that the stationary points of the form (5.13) for M are saddle points. We have therefore the following lemma:

Lemma 5.2: All stationary points of the form (5.13) for M even are saddle points of the cost function (5.9).

The above given proof of Lemma 5.2 verifies the following fact: for the case $M = \text{even}$ we have been able to find proper perturbations around the stationary points such that the cost function takes a smaller value on them than on the stationary points. Such a perturbation for example is

$$s_M^{\pm\epsilon} = \alpha_M [\sqrt{1+\epsilon} [\dots e^{j\theta_1} \dots e^{j\theta_{M/2}}] \sqrt{1-\epsilon} [\dots e^{j\theta_{M/2+1}} \dots e^{j\theta_M}]]$$

Combined with (5.27), this also proves the fact that for M even the stationary points are saddle points.

- M is odd

A similar image as for the case $M = \text{even}$ is not unfortunately obtained for the case $M = \text{odd}$. Evaluating the expression of λ_2 for CM constellations, it turns out that for some of them there exist values of M (especially $M = 3$) for which $\lambda_2 > 0$. For example, consider the case $M = 3$ and a 4-QAM input constellation. It can be easily found that in this case

$$\lambda_2 = 0.1862 .$$

Such counterexamples constitute enough evidence for the fact that for the case $M = \text{odd}$, there exist stationary points that are indeed local minima (since λ_1 is also positive). Therefore we can state the following lemma :

Lemma 5.3: Some of the stationary points of the form (5.13) for M odd are local minima of the cost function (5.9).

The two Lemmas 5.2 and 5.3 contain the main conclusions of this study with respect to the performance of the CMA 1-2 algorithm and show its inferior performance as compared to that of the CMA 2-2: whereas the CMA 2-2 has been proven to have no local minima in the case of an infinite equalizer length (for any constellation), the same does not hold for the CMA 1-2, which we have proven to have some local minima in the case of a discrete CM constellation.

However, the following remark is of interest. As the number of constellation points on the unit circle increases, the eigenvalues λ_2 head towards negative values, and therefore the corresponding settings are saddle points. This can be observed for example in table 1, which shows the evolution of λ_2 for the case $M = 3$. It should be noted that the case $M = 3$ is the more difficult one, since it corresponds to the smallest possible number of even terms. In all the other cases ($M > 3$), λ_2 takes on the negative values much quicker, if not already for 4-QAM.

By increasing even more the number of constellation points equispaced on the circle we arrive to asymptotic case of a continuous input sequence whose angle is uniformly distributed in

L- QAM ; M = 3	
L	λ_2
4	0.1862
8	0.0417
14	0.0144
26	-0.0053
34	-0.0095
52	-0.017

Table 5.1: The evolution of λ_2 for $M = \text{odd}$

$[0, 2\pi)$. In this case λ_2 will be evidently negative for any M and therefore we may state the following result.

Theorem 5.1: When the input signal is CM and its angle is continuously-uniformly distributed in $[0, 2\pi)$, then all equilibria of the CMA 1-2 cost function of the form (5.13) are saddle points, except for $M = 1$.

This result parallels the one stated by Benveniste *et al* in [BG84]: they have proven that the Sato algorithm has no local minima in the case of several *continuous* input sequences. Here we prove the same thing for the CMA 1-2 and the special case of a continuous CM input sequence. This result even though never stated before, should be expected, since the CMA 1-2 is the natural extension of the Sato in the complex case (they share exactly the same update formula).

Application to the DDA cost function

As already stated, in the case of the continuous input distribution considered, the CMA 1-2 and the DDA algorithms are identical. The same result in Theorem 5.1 applies therefore also for the DDA:

Theorem 5.2: When the input signal is CM and its angle is continuously-uniformly distributed in $[0, 2\pi)$, then all equilibria of the DDA cost function of the form (5.13) are saddle points, except for $M = 1$.

This is our main result with regard to the performance analysis of the DDA: based on two assumptions similar to the ones that have been used to analyze the performance of classical BE algorithms, we arrive at a similar result: for an infinite-length equalizer and a continu-

ous CM input distribution⁴, the DDA has no local minima in its cost function. From this perspective, it seems not to be a less preferable BE method than the other CMA's, since in any case under realistic assumptions, none of them is optimal, whereas under the unrealistic assumptions stated above they are all more or less equivalent (the CMA 2-2 being of course always better since it only needs the infinite-length assumption). Moreover, the DDA has the obvious advantage of a lower steady-state error.

We will now try to analyze the DDA in the non-CM case, in order to throw some light to its observed misbehaviour in this case.

5.3.2 The non-CM case

In the case of non-CM constellations DD equalization usually does not have a satisfactory performance, especially when the channel eye is initially closed. We consider for simplicity the PAM case. The DD cost function in this case is

$$J^D(s) = E(y_k - dec(y_k))^2. \quad (5.39)$$

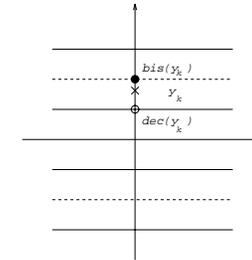
For the partial first derivative we have

$$\begin{aligned} \frac{\partial J^D}{\partial s_k} &= 2E(y_k - dec(y_k)) \left(\frac{\partial}{\partial s_k} \sum_i a_{n-i} s_i - \frac{\partial}{\partial s_k} dec \left(\sum_i a_{n-i} s_i \right) \right), \\ &= 2E(y_k - dec(y_k)) a_{n-k} \left(1 - \frac{\partial dec(y_k)}{\partial y_k} \right). \end{aligned}$$

Now we have that

$$\frac{\partial dec(y_k)}{\partial y_k} = \delta(y_k - bis(y_k)),$$

where $bis(y)$ is the closest point to y that belongs on a bisector between the different PAM levels (as shown in figure 5.3 for a multi-PAM case), and $\delta(\cdot)$ is the Dirac function.

Figure 5.3: The definition of the $bis(\cdot)$ function.

⁴Note that this is a sub-Gaussian distribution.

From the two above relations we have

$$\frac{\partial J^D}{\partial s_k} = 2E \left(\sum_i a_{n-i} s_i - \text{dec} \left(\sum_i a_{n-i} s_i \right) \right) a_{n-k}^* \left(1 - \delta \left(\sum_i a_{n-i} s_i - \text{bis} \left(\sum_i a_{n-i} s_i \right) \right) \right), \quad (5.40)$$

and therefore the stationary points are given by the equation

$$E \left(\sum_i a_{n-i} s_i - \text{dec} \left(\sum_i a_{n-i} s_i \right) \right) a_{n-k}^* \left(1 - \delta \left(\sum_i a_{n-i} s_i - \text{bis} \left(\sum_i a_{n-i} s_i \right) \right) \right) = 0, \quad (5.41)$$

If we neglect the singularities introduced by the δ function the above equation simplifies to

$$E \left(a_{n-k}^* \sum_i a_{n-i} s_i \right) = E \text{dec} \left(a_{n-k}^* \text{dec} \left(\sum_i a_{n-i} s_i \right) \right). \quad (5.42)$$

Assuming that s has M non-zero taps ($s = (\dots 0 s_{k_1} 0 \dots 0 s_{k_M} 0 \dots)$), the non-zero elements of s should satisfy

$$s_{k_i} = E \left(a_{n-k_M}^* \text{dec} \left(\sum_{j=1}^M a_{n-k_j} s_{k_j} \right) \right). \quad (5.43)$$

Notice the similarity between (5.43) and the corresponding equation (5.12) for the Sato algorithm: in (5.43) the $\text{dec}(\cdot)$ function replaces the $\text{sign}(\cdot)$ function of (5.12). This seems natural since it corresponds to the difference between the update equations of the two algorithms: the Sato projects on a CM, whereas the DDA on the closest symbol. However, it should be reminded that in order to arrive to (5.43) we have had to neglect some points corresponding to singularities, and this reflects the fact that the DDA is by one degree of differentiability more complicated than the Sato algorithm.

In the case of the Sato (or the CMA 1-2), we have been able to derive an analytical expression for the solution of (5.12) by restraining the taps of each solution to have equal magnitudes. If now we impose the same constraint on the solution for eq. (5.43) however, we are still not able of obtaining an analytical solution for the stationary points, due to the strong nonlinearity of the $\text{dec}(\cdot)$ function. This also reflects the fact that the solution of (5.43) is no longer unique (as for the Sato or the CMA 1-2) for each M . This non-uniqueness of solution for each set of stationary points that satisfy (5.43) has to do of course with the number and the shape of the local minima of the cost function (5.39).

In order to have an idea of the solutions of (5.43) we have proceeded as follows. We have numerically evaluated for some examples the two terms appearing in (5.43) in order to find when they coincide, corresponding to a stationary point. For example, in a 4-PAM case with $M = 2$ there exist 5 different stationary points for the DDA cost function, as compared to a unique stationary point for the Sato cost function.

This can be seen in figure 5.4, where we have plotted the difference of the two terms in (5.43) and the corresponding equation for Sato, respectively. Moreover, it can be proven that 2 of these solutions are local maxima (compared to the absence of local maxima for the Sato

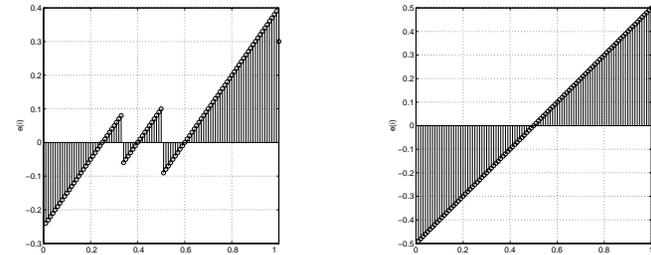


Figure 5.4: 4-PAM: The stationary points on the line $s_1 = s_2$ for DDA and Sato, respectively

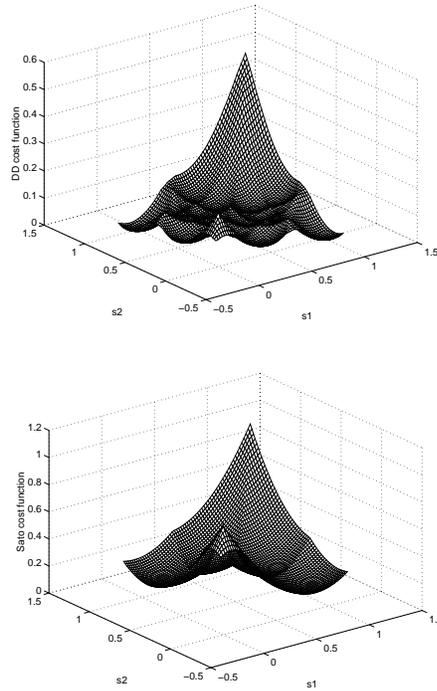
or the CMA 1-2). This multitude of stationary points that can be either local minima, saddle points, or local maxima is responsible for the very different shape of the DDA cost function in a multi-level PAM system, as compared to the Sato (or CMA 1-2) cost function. This can be seen in figure 5.5, where the two corresponding cost functions have been plotted for a 4-PAM system and $M = 2$. Note in this figure the fact that in the Sato cost function, the regions separating local minima are open cone-shaped regions, whereas the corresponding regions of the DDA cost function are closed, and various local maxima and saddle points separate the local maxima. This fact has also been discussed in [Maz80] (see also Appendix 5.B for more details). This is why escaping from a local minimum is much more difficult in the DD case (for non-CM constellations).

5.4 Other classes of DD algorithms

The LMS-like DD algorithm (5.1) is up to our knowledge the only adaptive algorithm that has been proposed for DD equalization. This is probably due to the fact that it was always used during the steady-state, after the initial convergence of either the blind or the training algorithm. As according to the above analysis we have now good reason to believe that with a DD equalizer the channel eye can open, there is interest in proposing other (non LMS-like) algorithms with better performance in terms of convergence speed or steady-state error.

In this perspective, and since the DDA has the same form as the CMA 1-2, we propose the following simple rule for the acquisition of algorithms suitable for DD-BE:

An algorithm suitable for DD blind equalization can be obtained if one considers any classical adaptive filtering algorithm and replaces the desired sample d_k at each iteration by the decision $\text{dec}(y_k)$.

Figure 5.5: 4-PAM: the DDA and Sato cost functions for $M = 2$

This rule parallels the separation principle that was presented in chapter 3⁵. Table 5.2 shows some of the algorithms that can be obtained by applying it to the NLMS, APA, Regularized APA and RLS, respectively

In this table, the quantity $dec([v_1 \cdots v_N])$ is the vector $[dec(v_1) \cdots dec(v_N)]$.

The various characteristics of the above adaptive filtering algorithms are expected to be reflected to the corresponding DD algorithms. Moreover, the DD algorithms are expected to have a better steady-state behaviour w.r.t. the corresponding CMA's, for the reason explained in section 5.1.

⁵Normalized DD algorithms can be derived by minimizing exactly a deterministic criterion in a similar way as the one followed for the derivation of the NSWCMA in chapter 3.

LMS	$W_{k+1} = W_k + \mu X_k(d_k - y_k)$
DDA	$W_{k+1} = W_k + \mu X_k(dec(y_k) - y_k)$
NLMS	$W_{k+1} = W_k + \frac{\mu}{\ X_k\ _2^2} X_k(d_k - y_k)$
NDDA	$W_{k+1} = W_k + \frac{\mu}{\ X_k\ _2^2} X_k(dec(y_k) - y_k)$
APA	$W_{k+1} = W_k + \mu X_k P_k^{-1} (D_k - X_k^H W_k)$
NSWDDA	$W_{k+1} = W_k + \mu X_k P_k^{-1} (dec(X_k^H W_k) - X_k^H W_k)$
Reg. APA	$W_{k+1} = W_k + \mu X_k R_k^{-1} (D_k - X_k^H W_k)$
Reg. NSWDDA	$W_{k+1} = W_k + \mu X_k R_k^{-1} (dec(X_k^H W_k) - X_k^H W_k)$
RLS	$W_{k+1} = W_k + R_k^{-1} X_k(d_k - y_k)$
'RLS-DDA'	$W_{k+1} = W_k + R_k^{-1} X_k(dec(y_k) - y_k)$

Table 5.2: DD-type BE algorithms

5.5 A CMA-DD hybrid corresponding to a novel Generalized-Sato scheme

The results of the previous section suggest that the principal factor that causes the misbehaviour of the DDA is the multiple constellation amplitudes, whereas in the case of a single constellation amplitude its performance should be similar to the one of the CMA 1-2 algorithm. Indeed, as can be observed in [Sat75], this has been the original motivation for Sato's work. Sato proposed his cost function empirically by applying a somewhat "reduced constellation" philosophy: for any multi-level PAM constellation he formed his decisions by finding the closest symbol of a corresponding 2-PAM constellation. This corresponds to projecting the multi-PAM constellation (which is real) on the only real CM constellation: 2-PAM.

When the Sato algorithm was generalized to the GSA for QAM constellations, a similar (even though not quite) philosophy was applied: the reduced constellation used was a 4-QAM. However this principle deviates from Sato's idea in the following: in the real-case Sato, all the projected (2-PAM) symbols have the same angle as their initial multi-PAM counterparts.

But in GSA, the use of a 4-QAM reduced constellation changes the angles of the projected symbols.

Trying to keep unchanged the symbols angle in the complex case, has lead us to propose the following CM-DD hybrid method for equalization:

- Construct from the non-CM constellation a reduced CM constellation by associating each symbol a_i with the complex number $r_1 \text{sign}(a_i)$
- Use the CM constellation to take the decisions about the received samples.

This method can be used to derive LMS-like or other adaptive BE algorithm in the same way as described in the previous section, the only difference being that now instead of using the function $\text{dec}(\cdot)$ (DD) or $\text{sign}(\cdot)$ we use the following function $\text{cdec}(\cdot)$:

$$\text{cdec}(y_k) = \text{the closest symbol of } \{\text{sign}(a_i)\} \text{ to } y_k. \quad (5.44)$$

An example of this principle can be seen in figure 5.6, where the non-CM constellation is 16-QAM ('+' denotes a constellation symbol, '*' a projected symbol). Our hybrid principle

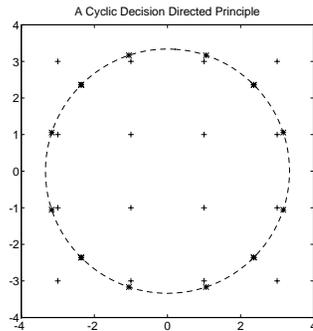


Figure 5.6: A CMA-DD hybrid principle: 16-QAM

preserves the constant modulus character of the constellation, but not in the 4-QAM reduced constellation sense of the GSA. It is still however in some sense a DDA since it takes decisions, but on the reduced constellation: in our opinion it is the kind of DDA that should be used in non-CM constellations and its performance should verify the fact that DD equalization is good if all the decisions taken have the same amplitude. Moreover, the fact that the number of symbols in the projected constellation is in general higher than in the GSA is expected to lead to a reduced steady-state error.

A short discussion about this principle can also be found in [PS94a] where different algorithms for its implementation are given.

5.6 Computer simulations

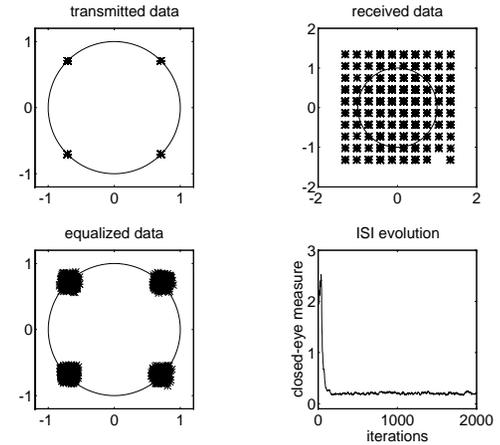


Figure 5.7: The opening of the eye achieved by a DD algorithm

In this section we will provide some computer simulations that verify and support our theoretically obtained results. These simulations are by no means exhaustive, but just indicative of the arguments of this chapter.

DD-BE simulation:

We consider a noisy FIR communication channel. The channel impulse response has four non-zero taps: $c = [1 - 3 3 2]$ and the SNR at the channel output has been fixed at $30dB$. The equalizer was taken to have 7 taps and was always initialized with a unique non-zero middle tap equal to 1. The following three cases were tested:

- CM input signal, DD algorithm

In this case, the input signal is taken to be 4-QAM. We then use the NSWDDA (see table 5.2) with parameters $\bar{\mu} = 0.1$ and $L = 2$. The results obtained can be shown in figure 5.7. Note the form of the received data, indicative of the fact that the channel eye is closed (the closed-eye measure is around 2 at the beginning of the algorithm), and how the 4-QAM pattern is retrieved after the algorithm has converged.

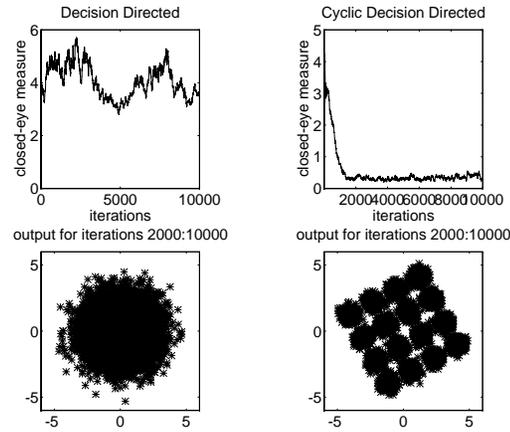


Figure 5.8: The opening of the eye achieved by a the CMA-DD hybrid in a non-CM case

This result shows that a DD algorithm is indeed capable of opening a closed channel eye, and this in accordance with our theoretical claims, since the transmitted constellation is CM.

- Non-CM input signal, DD algorithm

In this case, the input signal is taken to be 16-QAM and we use again the same NSWDDA as before, with parameters $\bar{\mu} = 0.1$, $L = 2$. Figure 5.8 shows the evolution of the closed-eye measure (as well as the corresponding equalizer output) for this case, for 10000 iterations of the algorithm. Notice that the channel eye hasn't been opened in the case. This is due, as expected, to the fact that the transmitted constellation is not CM. It should be noted here that there have been observed some cases where an eye opening has been achieved even for a non-CM signal. This has been very rare and can be explained by the fact that despite the existence of many undesirable stationary points, in some cases the algorithm has by chance converged to stationary points that correspond to open-eye settings.

- Non-CM input signal, CMA-DD hybrid algorithm

The same 16-QAM constellation as before is used, but now we use one of the CMA-DD hybrid algorithms proposed in section 5.5, namely a NSW one with parameters $\bar{\mu} = .05$, $L = 2$ (same as before). Note how now the channel eye gets gradually opened and how the 16-QAM pattern is retrieved⁶. This shows the success of our hybrid CMA-DD principle and verifies

⁶A better opening of the eye can be achieved of course if a longer equalizer is used.

our initial intuition that the problem in DD equalization comes from the existence of multiple amplitudes in the transmitted constellation.

5.7 Conclusions

We have taken a close look at the problem of DD BE in the particular case of CM signals. An analysis of the stationary points of the CMA 1-2 cost function has shown the absence of local minima for the DDA cost function in the asymptotic case of a CM distribution whose angle is uniformly distributed in $[0, 2\pi)$ ⁷. This result parallels the one presented in [BG84], where it was proven that the Sato algorithm has no local minima in the case of a continuous sub-Gaussian input distribution and justifies the following statement: DD equalizers are capable of opening an initially closed channel eye when the input signal is CM. Combined with the fact that DD equalizers have a reduced steady-state error as compared to other blind equalizers, this result indicates (in contrast to what is widely believed to date) that DD equalization is a valid BE method for CM constellations.

Our stationary-point analysis also provides useful insight into the performance of DD equalizers in the non-CM case: clearly, it is the existence of multiple constellation amplitudes that is responsible for the bad shape of the cost function in this case. Based on this remark, we propose a CMA-DD hybrid scheme for BE which constitutes a novel Generalized Sato-like scheme for QAM constellations that respects Sato's philosophy more closely than the already existing GSA algorithm: a *projected* constellation on a circle of radius r_1 is used (instead of a *reduced* constellation) in order to form the decisions about the transmitted symbols. The performance of algorithms based on this scheme as well as the ability of DD equalizers to open an initially closed channel eye are verified by our computer simulations. A part of the work presented in this chapter has been also presented in [PS94b].

⁷In the case of discrete symmetrical CM distributions, the difference between settings with an odd or an even number of nonzero taps has been revealed.

5.8 Appendix 4A: On the uniqueness of the stationary points (5.13).

Consider eq. (5.12). Combining it with (5.8) we get

$$\begin{aligned} s_1 &= E \left(e^{-j\phi_1} \text{sign}(e^{j\phi_1}s_1 + \dots + e^{j\phi_M}s_M) \right), \\ &= E \left(e^{-j\phi_1} \text{sign} \left([e^{j\phi_1} \dots e^{j\phi_M}] \begin{bmatrix} s_1 \\ \vdots \\ s_M \end{bmatrix} \right) \right), \end{aligned}$$

which can be written in matrix form

$$\mathbf{s} = E \left((\mathbf{a} \text{sign}(\mathbf{a}^H \mathbf{s})) \right), \quad (5.45)$$

where we have defined

$$\mathbf{s} = \begin{bmatrix} s_1 \\ \vdots \\ s_M \end{bmatrix}, \quad \mathbf{a} = \begin{bmatrix} e^{-j\phi_1} \\ \vdots \\ e^{-j\phi_M} \end{bmatrix}.$$

Eq. (5.45) can be written in the form

$$\mathbf{s} = E \left(\frac{\mathbf{a}\mathbf{a}^H}{|\mathbf{a}^H \mathbf{s}|} \mathbf{s} \right). \quad (5.46)$$

The solution to (5.46) is

$$\mathbf{s} \in \mathcal{N} \left(I_M - E \left(\frac{\mathbf{a}\mathbf{a}^H}{|\mathbf{a}^H \mathbf{s}|} \mathbf{s} \right) \right), \quad (5.47)$$

where $\mathcal{N}(\cdot)$ denotes the nullspace of a matrix.

Due to the stationarity of the input sequence, according to (5.46), if \mathbf{s} is a solution of it, then any reordering of its entries is also a solution of (5.46). Now if the solution of (5.46) was guaranteed to be unique, then all elements of \mathbf{s} should be equal between them. Or, if the solution was at least guaranteed to have solutions with a unique magnitude, then all elements of \mathbf{s} would have the same magnitude, and in this case s would be of the form (5.13), which is the solution given by Godard in [God80]. However if this uniqueness of solution of (5.46) cannot be guaranteed, then there may exist also other solutions for \mathbf{s} whose elements do not share the same magnitude.

5.9 Appendix 4B: On the shape of regions separating local minima

We adhere to Mazo's article [Maz80]. As explained in this article, the regions around local minima of the DDA algorithm are defined by the equation

$$s_l(y_k) = \text{constant}, \quad (5.48)$$

where $i \in \{1, 2\}$, $i = 1, 2$ corresponding to the 2-PAM and 4-PAM case, respectively, and we use the definitions

$$s_{l_1}(y_k) = \begin{cases} 1 & y_k > 0 \\ -1 & y_k < 0 \end{cases}, \quad (5.49)$$

and

$$s_{l_2}(y_k) = \begin{cases} 1 & y_k < 2 \\ 3 & y_k > 2 \end{cases}, \quad s_{l_2}(-x) = -s_{l_2}(x). \quad (5.50)$$

The regions corresponding to $i = 1$ are cone-shaped because they are formed by intersecting half planes. This is not however the case for $i = 2$, since due to the form of $s_{l_2}(\cdot)$ the resulting regions are more complicated N -dimensional surfaces.

This fact is reflected also in figure 5.5. It is also worth noting that in the case of our hybrid CMA-DD scheme, the local minima regions are also open cones, and this explains the fact that the corresponding algorithms are able of opening an initially closed channel eye.

Chapter 6

A bilinear setup for globally convergent blind equalization

IN this chapter we address the problem of ill convergence of constant modulus algorithms. In order to face this problem we propose a different parameterization of the problem that leads to the construction of a *convex* cost function with respect to an introduced parameter set. The minimization of this cost function can be performed either by batch or by adaptive techniques, by use of any multichannel classical adaptive filtering algorithm, which will converge to the same minimum point regardless of the algorithm's *initialization*. It is shown then how the unique minimum of the cost function is linked to the ideal equalizer setting for zero forcing equalization. The influence of various factors as additive noise and over-parameterization of the problem is also discussed and computer simulations are provided in order to support the claimed behaviour.

6.1 Introduction

In this chapter we are primarily interested in the problem of ill-convergence of Godard-type algorithms for blind equalization. As previously mentioned, this is one of the major problems related to the performance of these algorithms and has to do with the shape of the corresponding cost functions. Consider for example the Godard minimization problem which we repeat here for convenience:

$$\min_W J_p(W) = \frac{1}{2^p} E(|y|^p - r_p)^2 \quad p \in \{1, 2, \dots\}, \quad (6.1)$$

where E denotes statistical expectation and $r_p = \frac{E|a_k|^{2p}}{E|n_k|^{2p}}$. In the sequel we assume for simplicity that $r_p = 1$. The Godard cost function appearing above is not convex with respect to the

equalizer parameters, due to the nonlinearity involved. As was already explained in previous chapters this is the case for all Godard algorithms, at least when the equalizer is finite-length. The direct implication of the non-convex shape of the cost function is the fact that there exists a hole class of local minima some of which are undesired in the sense that they are different than the global minima of the cost function. The equation that gives the stationary points is, as already mentioned before, obtained by setting to zero the first derivative of the cost function $J_p(W)$ with respect to the equalizer parameters W . This gives for example in the case $p = 2$ (CMA 2-2 cost function):

$$E((|y_k|^2 - 1)y_k X_k) = 0. \quad (6.2)$$

A stochastic gradient search that attempts to minimize the cost function will inevitably be likely to end up to a non-desired local minimum. This is the heart of the problem of ill-convergence of BE algorithms, a problem that has been discussed extensively in the literature (see e.g. [Din91a],[DKAJ91],[DKAJ93],[JL92],[LDKJ94],[JLK92],[JDS88],[CS90]). Several attempts to overcome this problem have also been realised ([CNP92],[BZA94],[VAK91],[KD92]). In general there are three different factors that influence the ill convergence of such a cost function:

- The shape of the cost function
- The algorithm's initialization
- The algorithm's stepsize

The first among these factors is of course the most important, since if one is able of obtaining a convex cost function, this means that the algorithm will converge to its optimal setting independently of its initialization or its stepsize (provided that the latter is within the stability region). On the other hand, the other two factors are important when the cost function is not convex, since they might help the algorithm converge to an acceptable (or even to its optimal) setting. For example, a center-spike initialization has often been recommended as a practical guideline for the use of BE algorithms (even though it has been shown that it might not always be successful). On the other hand, the stepsize has an important role to play in the sense that it can help escaping from local minima. We have discussed this issue in chapter 3, where it was shown that normalized CMA algorithms have an improved performance with respect to their non-normalized counterparts in terms of ill-convergence.

In this chapter we are interested in the first of the above factors, namely, we will propose the construction of a convex cost function for blind equalization of the CM type.

6.2 A new blind equalization principle

6.2.1 The idea

Figure 6.1 shows the basic principle used for BE by the CMA algorithm. Our emphasis in this figure is the traditional receiver scheme: the received data sequence is first passed through a linear equalizer. Then some kind of nonlinearity is applied to the equalizer output. This nonlinearity has the role to provide the higher order statistics needed for the implicit identification of the channel impulse response. The error computed with the help of this nonlinear function is then used by an adaptive algorithm in order to update the current equalizer setting. A direct consequence of this scheme is the fact that, as the nonlinearity

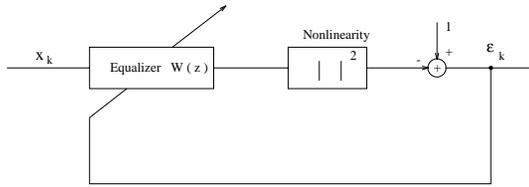


Figure 6.1: The principle of a typical CM blind equalizer

is applied to the equalizer output (which is already a linear combination of the equalizer parameters), the error at each time instant will be a nonlinear function of the equalizer parameters. It is exactly this fact that differentiates the BE adaptive algorithms from their classical adaptive filtering counterparts: in the latter case, the update error is a linear function of the filter parameters (corresponding to a convex (quadratic) cost function), whereas in the former it is nonlinear, and this is why the problem of local minima and ill convergence appears.

In order to keep the algorithm's error a linear function of the parameters the following alternative equalization scheme could be proposed: instead of applying the nonlinearity to the equalizer output, which will result in a nonlinear error for the equalizer parameters, one might try instead to apply first the nonlinearity to the received data and then pass the resulting process through a linear equalizer. In this case the update error will be a linear combination of parameters and we will be confronted with a case similar to classical adaptive filtering. Of course it is necessary to find a proper parameterization for the equalizer, as well as a proper way to apply the nonlinearity to the input process in such a way so that the overall system corresponds to an equivalent BE technique. In the following subsection we present a way to do that for the CMA 2-2 algorithm.

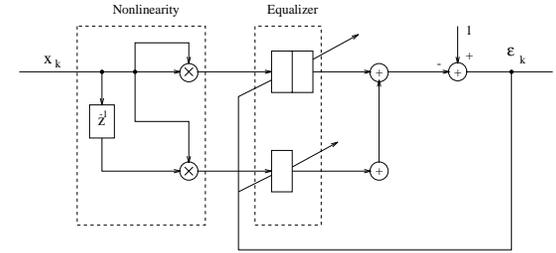


Figure 6.2: A proposed principle for CM blind equalization: an example

6.2.2 The formulation

Before giving our general formulation, we will first present a simple example in order to explain better our idea. Consider a length-2 equalizer and the CMA 2-2 algorithm. Assume also for the moment that all quantities are real. The nonlinearity $|y_k|^2$ used by the algorithm can be then expressed as

$$|y_k|^2 = y_k^2 = (x_k w_0 + x_{k-1} w_1)^2 = x_k^2 w_0^2 + x_{k-1}^2 w_1^2 + 2x_k x_{k-1} w_0 w_1,$$

which can be put in matrix form as

$$y_k^2 = [x_k^2 \ x_k x_{k-1} \ x_{k-1}^2] \begin{bmatrix} w_0^2 \\ 2w_0 w_1 \\ w_1^2 \end{bmatrix}. \quad (6.3)$$

In the above equation we have expressed the nonlinear quantity $|y_k|^2$ as a linear combination of the entries of the appearing column vector. The price paid for having such a linear relation is that both the input data as well as the equalizer parameters are now appearing through their corresponding *bilinear* terms ($x_k^2, x_k x_{k-1}$, e.t.c.). However this expansion indicates to us a way to implement a BE scheme that interchanges the position of the equalizer and the nonlinearity. (6.3) can also be written in the form

$$y_k^2 = [x_k^2 \ x_{k-1}^2] \begin{bmatrix} w_0^2 \\ w_1^2 \end{bmatrix} + [x_k x_{k-1}] [2w_0 w_1] = \mathcal{X}_1^T \mathcal{W}_1 + \mathcal{X}_2^T \mathcal{W}_2 \quad (6.4)$$

Note that in (6.4) the output squared modulus is now a sum of two regressor-equalizer products, where now the entries of each regressor are of the same type: \mathcal{X}_1 contains the quadratic whereas \mathcal{X}_2 the cross bilinear terms. This can be seen in figure 6.2: the process $\{x_k^2\}$ is passed through a length-2 filter that corresponds to the quadratic equalizer terms, whereas the process $\{x_k x_{k-1}\}$ through a length-1 filter corresponding to the cross term $2w_0 w_1$. Note

that despite the interchanged order of the two stages (nonlinearity-equalization), the output is the same as the one of figure 6.1: it can therefore be subtracted from 1 and provide the same error at each iteration as the one provided in figure 6.1. This error can then be used in order to update the two equalizers as shown in figure 6.2. Before we enter into the details of this implementation we first give the formulation of this approach for the general complex case.

We now consider all the quantities (transmitted symbols, channel impulse response, equalizer) to be complex, and assume (as usually) that the corresponding linear equalizer has N coefficients:

$$y_k = [x_k \cdots x_{k-N+1}] \begin{bmatrix} w_0 \\ \vdots \\ w_{N-1} \end{bmatrix} = X_k^H W_k .$$

The equalizer output squared modulus can be expanded in the following form

$$|y_k|^2 = y_k y_k^* = (w_0 w_0^* x_k x_k^* + \cdots + w_0 w_{N-1}^* x_k x_{k-N+1}^* + \cdots + (w_{N-1} w_0^* x_{k-N+1} x_k^* + \cdots + w_{N-1} w_{N-1}^* x_{k-N+1} x_{k-N+1}^*)) . \quad (6.5)$$

We may now introduce a $N^2 \times 1$ bilinear regression vector

$$\mathcal{X}_k = [x_k x_k^* \cdots x_k x_{k-N+1}^* \cdots x_{k-1} x_k^* \cdots x_{k-1} x_{k-N+1}^* \cdots x_{k-N+1} x_k^* \cdots x_{k-N+1} x_{k-N+1}^*]^H , \quad (6.6)$$

and a $N^2 \times 1$ parameter vector

$$\theta_k = [w_0 w_0^* \cdots w_0 w_{N-1}^* \cdots w_1 w_0^* \cdots w_1 w_{N-1}^* \cdots w_{N-1} w_0^* \cdots w_{N-1} w_{N-1}^*]^T , \quad (6.7)$$

that contain all the bilinear terms $x_{k-i} x_{k-j}^*$ and $w_i w_j^*$ of the expansion in (6.5), respectively. Then the squared modulus of the output y_k can be written as:

$$|y_k|^2 = z_k = \mathcal{X}_k^H \theta_k . \quad (6.8)$$

Since inner products correspond to convolutions, z_k may be viewed as the output of a linear “filter” with impulse response θ which is excited by the “bilinear” regression vector \mathcal{X} , at time instant k . Expressing the equalizer output squared modulus as in (6.8), the CMA 2-2 cost function can be written as

$$J_2(W) = \frac{1}{4} E(|y|^2 - 1)^2 = \frac{1}{4} E(z - 1)^2 .$$

Now keeping for z the parameterization (6.8), we can modify the minimization problem (6.1) by considering it as a problem for the determination of the θ parameters. The resulting minimization problem will then take the form

$$\min_{\theta} J^{bil}(\theta) = \min_{\theta} E(z - 1)^2 = \min_{\theta} E(\mathcal{X}^H \theta - 1)^2 . \quad (6.9)$$

In (6.9), the same exactly quantity as in (6.1) ($p = 2$) is minimized. From this point of view, the two minimization criteria are equivalent. However, the fact that the involved parameters

are different, changes the shape of the cost function with respect to its parameters. Indeed, the cost function $J^{bil}(\theta)$ as it appears in (6.9) is a quadratic function of the θ parameters. This has been exactly the role of the parameter transformation in (6.7): it is a mapping of the N -dimensional W -parameter space to the N^2 -dimensional θ -parameter space, such that when the CMA 2-2 cost function is expressed in the θ -parameter space, it is a convex function. Of course the question of crucial importance is about the existence and uniqueness of the inverse parameter mapping.

6.2.3 The main result

We now elaborate more upon the cost function (6.9) as a function of the θ parameters. The gradient of $J^{bil}(\theta)$ w.r.t. θ is given by:

$$\nabla_{\theta} J^{bil}(\theta) = 2E(\mathcal{X}(\mathcal{X}^H \theta - 1)) , \quad (6.10)$$

and therefore any solution θ to (6.9) should satisfy the following equation:

$$\mathbf{R} \theta = \mathbf{d} , \quad (6.11)$$

where

$$\begin{aligned} \mathbf{R} &= E(\mathcal{X} \mathcal{X}^H) \\ \mathbf{d} &= E(\mathcal{X}) . \end{aligned} \quad (6.12)$$

Now if the matrix \mathbf{R} is invertible, the problem (6.9) has the following unique solution:

$$\theta = \mathbf{R}^{-1} \mathbf{d} . \quad (6.13)$$

So if the non-singularity of \mathbf{R} can be guaranteed, we have achieved our principle aim: the minimization criterion (6.9) has a unique solution. What is important now is to see how channel identification (or equalization) is possible based on the solution (6.13). In a first step, we assume the following ideal characteristics for the channel:

Assumption 6A: The channel is assumed to be a noiseless, all-pole filter of order $N - 1$ (and therefore its output is an AR($N - 1$) process).

$$a_k = \sum_{i=0}^{N-1} c_i x_{k-i} = X_k^H C . \quad (6.14)$$

In this case the corresponding equalizer can be found from θ as follows: the optimal equalizer of length N in the absence of noise is the ZF equalizer whose entries are the coefficients of the impulse response of the inverse channel:

$$W^o = [c_0 \cdots c_{N-1}] = C^T . \quad (6.15)$$

The corresponding parameter vector will be given by:

$$\theta^{opt} = [c_0 c_0^* \cdots c_0 c_{N-1}^* \cdots c_1 c_0^* \cdots c_1 c_{N-1}^* \cdots c_{N-1} c_0^* \cdots c_{N-1} c_{N-1}^*]^T . \quad (6.16)$$

It is clear that θ^{opt} solves (6.9) since $J^{bil}(\theta^{opt}) = 0$. Moreover, under the assumption (6A), there exists no other solution that nulls exactly $J^{bil}(\theta)$. Therefore the unique solution (6.13) of (6.9), under the assumption (6A) is the one given in (6.16). This fact constitutes the main result of this chapter and is expressed through the following theorem:

Theorem 6.1: When the channel satisfies the assumption (6A), the blind equalization criterion (6.9) is a convex function in the θ parameter space of dimension N^2 . The unique solution θ^{opt} (given by (6.13) then contains the inverse channel coefficients as expressed in (6.16).

What is left now in order to identify the channel inverse impulse response, is to determine a way to acquire its coefficients from the obtained solution (6.16). This will show how the inverse parameter mapping can be achieved, namely how one may go from the redundant N^2 dimensional parameter space θ back to the N – dimensional parameter space of W . Consider a $N \times N$ matrix Θ that is created by filling in the N^2 entries of the vector θ according to the following rule: the matrix is getting filled column by column, beginning from the first column and from the top to the bottom:

$$\Theta = \begin{bmatrix} \theta^{(0)} & \theta^{(N)} & \dots & \theta^{(N^2-N)} \\ \theta^{(1)} & \theta^{(N+1)} & \dots & \theta^{(N^2-N+1)} \\ \vdots & \vdots & \dots & \vdots \\ \theta^{(N-1)} & \theta^{(2N-1)} & \dots & \theta^{(N^2-1)} \end{bmatrix} = \mathcal{G}(\theta) . \quad (6.17)$$

In the case of the optimal setting θ^{opt} , the corresponding matrix will be

$$\Theta^{opt} = \begin{bmatrix} |c_0|^2 & c_1 c_0^* & \dots & c_{N-1} c_0^* \\ c_0 c_1^* & |c_1|^2 & \vdots & c_{N-1} c_1^* \\ \vdots & \vdots & \vdots & \vdots \\ c_0 c_{N-1}^* & \dots & \dots & |c_{N-1}|^2 \end{bmatrix} . \quad (6.18)$$

Then it follows that $\Theta^{opt} = C C^H$, which means that in the ideal case, Θ^{opt} is a rank-1 matrix. A direct method to find back C from Θ is therefore to perform an eigendecomposition of Θ . Ideally, there will be a unique non-zero eigenvalue and the corresponding vector will be proportional to C :

$$C = \sqrt{\lambda} V(\lambda) , \quad (6.19)$$

where λ and $V(\lambda)$ are the non-zero eigenvalue and the corresponding eigenvector of Θ , respectively. Note that for a given Θ , any multiple $e^{j\phi} C$ still satisfies (6.18). This represents the fact that in blind equalization we can only identify the transmission channel apart from a rotational factor. We have therefore by now completed the presentation of a blind method

that allows the identification of the inverse channel coefficients for a channel obeying the assumption (6A). The method can be summarized as follows:

- Formulate a bilinear regressor (6.6) and a bilinear parameter vector (6.7).
- Minimize the cost function (6.9)
- Calculate the coefficients of the channel inverse from (6.19)

It is important to note that this method allows for perfect ZF equalization of the CMA 2-2 type, with no ambiguity, in contrast to the CMA 2-2 algorithm itself that has the problem of local minima and ill convergence (even for channels obeying the assumption (6A)). The problem of ill convergence has been therefore solved for this case.

In practice the matrix Θ will not be rank-1. In this case an easy way to approximate from it the channel inverse impulse response is to still use an eigenvalue decomposition:

$$W^{opt} \sim \sqrt{\lambda_{max}} V(\lambda_{max}) , \quad (6.20)$$

where now λ_{max} and $V(\lambda_{max})$ simply denote the maximum eigenvalue and the corresponding eigenvector of Θ . Note that there exist other also methods to obtain approximations of C . For example, one might just consider the first column of Θ and try to approximate C by dividing all its elements by the square root of its first element. However (6.20) is a more proper approximation since it takes into account all the elements of the matrix (the superior performance of (6.20) has been also verified by computer simulations).

6.2.4 Discussion

Apart from offering a method that overcomes the problem of ill-convergence, our bilinear formulation has allowed us to obtain an interesting alternative way to express the CMA 2-2 cost function. Namely, we have proven the following equivalence

$$\min_W E(|y|^2 - 1)^2 \equiv \begin{cases} \min_{\theta} E(\mathcal{X}^H \theta - 1)^2 \\ \text{subject to: } \text{rank}(\Theta) = \text{rank}(\mathcal{G}(\theta)) = 1 \end{cases} \quad (6.21)$$

Under the assumption (6A), the two problems have the same global solution, however the second one has the advantage of having no local minima. In section 6.4 we will examine the case in which the above condition $\text{rank}(\Theta) = 1$ does not hold. This will give valuable insight to the bilinear cost function and an interesting interpretation of local minima. This is the second important contribution of the bilinear formulation.

The fact that we have been able of identifying the (generally non-minimum phase) transmission channel by minimizing the cost function $J^{bil}(\theta)$ w.r.t. θ should not be an astonishing result, since the matrix \mathbf{R} contains 4th order moments of the received signal (it is known that identification of a non-minimum phase channel is not possible at the baud rate by use of

only second order statistics). A different point of view for the obtained identification result is the following: by making our parameter transformation, we introduce redundancy to our problem. The number of parameters introduced is greater than the number of parameters that need to be estimated. This redundancy may be viewed as some form of *diversity*, that allows the identification of the channel.

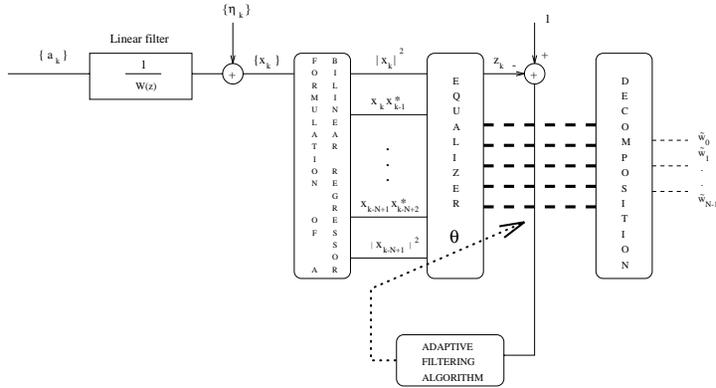


Figure 6.3: A representation of the bilinear method concept

6.2.5 Implementation

The inverse channel identification procedure we have formulated above can be used to obtain various blind equalization algorithms according to the different ways it can be implemented. Corresponding to two different criterion minimization philosophies, we can classify them as either batch or adaptive methods.

Batch methods

In batch methods, the general procedure followed is the following:

- Estimate from data the two stochastic quantities appearing in (6.11): \mathbf{R} and \mathbf{d}
- Use (6.13) to calculate θ

This is a typical off-line procedure, in which we suppose that enough data have been collected so as to provide accurate enough estimates for \mathbf{R} and \mathbf{d} . The quality of the method used will

depend not only on the amount of data available but also on the particular estimation method used (corresponding e.g. to different windows for the data et.c.) which will determine the variance and bias of the estimated quantities. Batch methods are usually employed in stationary environments.

Adaptive methods

In this case the minimization of the cost function in (6.9) is done adaptively, given an initial guess θ_0 . The convexity of the cost function (when \mathbf{R} is non-singular) guarantees the uniqueness of the solution and the global convergence of the adaptive algorithm employed, independently of the initialization. The advantages of adaptive methods are that they can be used on-line, they require less storage capabilities, often have a reduced computational complexity and that they may be used as well in non-stationary environments.

The general philosophy of “bilinear blind adaptive equalization” is depicted in figure 6.3. In contrast to batch techniques which compute their final equalizer setting at once (after all data have been collected and processed), with the setting of figure 6.3 the equalizer W_k can be computed at each iteration (decomposition stage) and be used in order to open gradually the channel eye. However this is not necessary and can be done at any frequency, depending on the computational requirements. We first present some classical adaptive techniques that can be adapted to the needs of this setup.

- Classical adaptive filtering

LMS:

The simplest adaptive algorithm that can be used is the LMS algorithm, which corresponds to a stochastic gradient minimization of the criterion in (6.9). The adaptation of the parameter vector θ is then given by

$$\begin{cases} \epsilon_k &= 1 - \mathcal{X}_k^H \theta_k \\ \theta_{k+1} &= \theta_k + \mu \mathcal{X}_k \epsilon_k, \end{cases} \quad (6.22)$$

where ϵ_k is the a priori error at time instant k and μ the stepsize parameter that controls both the convergence speed and steady-state error of the algorithm. The only difference of (6.22) and the classical way in which the LMS is used in system identification resides in the particular form of the regression vector \mathcal{X} : it is not stationary in the sense that all its entries do not correspond to the same process, as is usually the case. This does not influence however the performance characteristics of the algorithm. Of course the longer length (N^2) of the regressor with respect to the order of the underlying FIR filtering ($N - 1$) will increase the algorithm’s complexity, which will be now $2N^2$ multiplications/iteration. An NLMS analog can be as well implemented with roughly the same complexity.

RLS:

Similarly, an RLS-like algorithm can be used that will allow for faster convergence (in approximately $2N^2$ iterations) but at the expense of a higher computational complexity ($O(N^4)$ multiplications/iteration):

$$\begin{cases} \epsilon_k &= \mathbf{1} - \mathcal{X}_k^H \theta_k \\ R_k^{-1} &= \lambda^{-1} R_{k-1}^{-1} - \lambda^{-1} R_{k-1}^{-1} \mathcal{X}_k (1 + \mathcal{X}_k^H \lambda^{-1} R_{k-1}^{-1} \mathcal{X}_k)^{-1} \mathcal{X}_k^H \lambda^{-1} R_{k-1}^{-1} \\ \theta_{k+1} &= \theta_k + R_k^{-1} \mathcal{X}_k \epsilon_k, \end{cases} \quad (6.23)$$

where R_k is $N^2 \times N^2$ and λ is the forgetting factor.

In the case of the RLS algorithm, a reduction in computational complexity can be achieved if a *multichannel* structure is employed. Such a structure can be obtained by grouping the entries of the regressor \mathcal{X} so as to create a number of stationary regressors. Due to its simplicity, we will present this structure for the real case, the extension to the complex case being straightforward. When the received samples x_k are real (which corresponds to a PAM modulation and a real channel), a reduction in the required number of parameters for the bilinear method is possible. Namely, in this case the expansion in (6.5) will have only $\frac{N \times (N+1)}{2}$ terms, and therefore the regression and parameter vectors will be defined as :

$$\mathcal{X}_k = [x_k^2 \ 2x_k x_{k-1} \ \cdots \ x_{k-1}^2 \ 2x_{k-1} x_{k-2} \ \cdots \ x_{k-N+1}^2]^T, \quad (6.24)$$

and

$$\theta = [\theta^{(0)} \ \theta^{(1)} \ \cdots \ \theta^{(\frac{N \times (N+1)}{2} - 1)}]^T, \quad (6.25)$$

respectively. All the above-mentioned equations are still valid in this case, the only difference being the dimensions of θ , \mathcal{X} and the matrix \mathbf{R} . All Hermitian transposes can also be replaced by simple transposes in this case. Then the multichannel implementation of the equalizer consists in splitting the entries of \mathcal{X} into N *stationary* regressors as follows

$$\begin{aligned} X_1^T(k) &= [x_k^2 \ \cdots \ x_{k-N+1}^2] \\ X_2^T(k) &= [2x_k x_{k-1} \ \cdots \ 2x_{k-N+2} x_{k-N+1}] \\ &\vdots \\ X_N^T(k) &= [2x_k x_{k-N+1}] \end{aligned}$$

of lengths $N, N-1, \dots, 1$, respectively. The same holds for the corresponding equalizers. Figure 6.4 shows the setup for bilinear equalization in the real case. Note that all the equalizers share the same error function.

If a multichannel Fast Transversal Filter algorithm [Slo89] is used for this setup, the complexity is reduced to $O(N^3)$ multiplications/iteration.

This multichannel structure is equally applicable to any classical adaptive filtering algorithm and offers implementation advantages.

- A bilinear rank-1 approach

A different philosophy that takes into account the specific form of the bilinear regression vector can be used in order to obtain adaptive algorithms of a different type. For this we define first a matrix \mathbf{W}_k as follows

$$\mathbf{W}_k = W_k W_k^H. \quad (6.26)$$

The minimization problem (6.10) can then be written as

$$\begin{aligned} \min_{\mathbf{W}} J^{bil}(\mathbf{W}) \\ \text{subject to: } \text{rank}(\mathbf{W}) = 1 \end{aligned} \quad (6.27)$$

A bilinear rank-1 approach to solve adaptively this problem is

$$\begin{cases} \epsilon_k = \mathbf{1} - (W_{k-1}^H X_k)^2 \\ W_{k-1} W_{k-1}^H + \mu_k \epsilon_k X_k X_k^H = \lambda_1 V_1 V_1^H + \lambda_2 V_2 V_2^H \\ W_k = \sqrt{\lambda_1} V_1 \text{sign}(V_1^H W_{k-1}). \end{cases} \quad (6.28)$$

In (6.28), λ_1, V_1 can be found from W_{k-1}, X_k in $O(N)$ operations. (6.28) is an adaptive procedure of the stochastic gradient type (similar to the LMS) but applied to a parameter matrix instead of a parameter vector. Even though the global convergence property cannot in general be guaranteed for (6.28), some simple computer experiments have shown its successful performance in correctly parameterized cases. Such an example will be given in the computer simulations section.

6.3 The influence of additive noise

When additive noise is present in the received signal, it will corrupt both the quantities \mathbf{R} and \mathbf{d} and therefore the solution (6.13) will be biased and no longer correspond to a ZF equalizer (even in the absence of order mismatch). We will study the influence of additive white zero-mean Gaussian noise n_k in the real case (a_k, x_k, y_k are all real):

$$x_k = x'_k + n_k, \quad (6.29)$$

where x'_k is the noiseless channel output. The fourth order moments involved in \mathbf{R} are:

$$\begin{aligned} E(x_{k-i}^4) &= m_{4,x'}(\mathbf{0}, \mathbf{0}, \mathbf{0}) + 3\sigma_n^4 + 6m_{2,x'}(\mathbf{0})\sigma_n^2 \\ E(x_{k-i}^3 x_{k-j}) &= m_{4,x'}(\mathbf{0}, \mathbf{0}, |i-j|) + 3m_{2,x'}(|i-j|)\sigma_n^2 \\ E(x_{k-i}^2 x_{k-j}^2) &= m_{4,x'}(\mathbf{0}, |i-j|, |i-j|) + 2m_{2,x'}(\mathbf{0})\sigma_n^2 + \sigma_n^4 \\ E(x_{k-i}^2 x_{k-j} x_{k-l}) &= m_{4,x'}(\mathbf{0}, |i-j|, |i-l|) + m_{2,x'}(|i-j|)\sigma_n^2 \\ E(x_{k-i} x_{k-j} x_{k-l} x_{k-m}) &= m_{4,x'}(|i-j|, |i-l|, |i-m|), \end{aligned} \quad (6.30)$$

If the noise variance σ_n^2 is known, then the noise-free fourth order moments can be calculated as follows:

$$\begin{aligned} E((x_{k-i}^2 - 3\sigma_n^2)^2) - 6\sigma_n^4 &= m_{4,x'}(\mathbf{0}, \mathbf{0}, \mathbf{0}) \\ E((x_{k-i}^2 - 3\sigma_n^2)x_{k-i}x_{k-j}) &= m_{4,x'}(\mathbf{0}, \mathbf{0}, |i-j|) \\ E((x_{k-i}^2 - \sigma_n^2)(x_{k-j}^2 - \sigma_n^2)) &= m_{4,x'}(\mathbf{0}, |i-j|, |i-j|) \\ E((x_{k-i}^2 - \sigma_n^2)x_{k-j}x_{k-l}) &= m_{4,x'}(\mathbf{0}, |i-j|, |i-l|) \\ E(x_{k-i}x_{k-j}x_{k-l}x_{k-m}) &= m_{4,x'}(|i-j|, |i-l|, |i-m|) , \end{aligned} \quad (6.31)$$

Proof:

- $E(x_{k-i}^2 - \sigma_n^2)(x_{k-j}^2 - \sigma_n^2) = E x_{k-i}^2 x_{k-j}^2 - 2\sigma_n^2 E x_{k-i}^2 + \sigma_n^4$
 $= E(x_{k-i}^2 + v_{k-i}^2 + 2x'_{k-i}v_{k-i})(x_{k-j}^2 + v_{k-j}^2 + 2x'_{k-j}v_{k-j}) - 2\sigma_n^2 E(x'^2 + v^2 + 2x'v) + \sigma_n^4$
 $= E x_{k-i}^2 x_{k-j}^2 .$
- $E(x_{k-i}^2 - \sigma_n^2)x_{k-j}x_{k-l} = E x_{k-i}^2 x'_{k-j} x'_{k-l} - \sigma_n^2 E x'_{k-j} x'_{k-l}$
 $= E(x_{k-i}^2 + v_{k-i}^2 + 2x'_{k-i}v_{k-i})(x'_{k-j} + v_{k-j})(x'_{k-l} + v_{k-l}) - \sigma_n^2 E(x'_{k-j} + v_{k-j})(x'_{k-l} + v_{k-l})$
 $= E(x_{k-i}^2 + v_{k-i}^2 + 2x'_{k-i}v_{k-i})(x'_{k-j}x'_{k-l} + v_{k-j}x'_{k-l} + v_{k-i}x'_{k-j} + v_{k-j}v_{k-l}) - \sigma_n^2 E x'_{k-j} x'_{k-l}$
 $= E x_{k-i}^2 x'_{k-j} x'_{k-l} .$
- $E(x_{k-i}^2 - 3\sigma_n^2)x_{k-i}x_{k-j} = E x_{k-i}^3 x'_{k-j} - 3\sigma_n^2 E x'_{k-i} x'_{k-j}$
 $= E(x_{k-i}^3 + 3x_{k-i}^2 v_{k-i} + 3x'_{k-i} v_{k-i}^2 + v_{k-i}^3)(x'_{k-j} + v_{k-j}) - 3\sigma_n^2 E x'_{k-i} x'_{k-j} = E x_{k-i}^3 x'_{k-j}$
- $E(x_{k-i}^2 - 3\sigma_n^2)^2 - 6\sigma_n^4 = E(x_{k-i}^4 + 9\sigma_n^4 - 6\sigma_n^2 x_{k-i}^2) - 6\sigma_n^4$
 $= E(x_{k-i}^4 + 4x_{k-i}^3 v_{k-i} + 6x_{k-i}^2 v_{k-i}^2 + 4x'_{k-i} v_{k-i}^3 + v_{k-i}^4) + 9\sigma_n^4 - 6\sigma_n^2 E x'^2 - 6\sigma_n^4 = E x'^4$
- $E(x_{k-i}x_{k-j}x_{k-l}x_{k-m}) = E(x'_{k-i} + v_{k-i})(x'_{k-j} + v_{k-j})(x'_{k-l} + v_{k-l})(x'_{k-m} + v_{k-m})$
 $= E(x'_{k-i}x'_{k-j}x'_{k-l}x'_{k-m})$

□

The second order noise-free moments can also be obtained as:

$$\begin{aligned} E(x_{k-i}^2) - \sigma_n^2 &= m_{2,x'}(\mathbf{0}) \\ E(x_{k-i}x_{k-j}) &= m_{2,x'}(|i-j|) . \end{aligned} \quad (6.32)$$

In the above expressions

$$\begin{aligned} m_{4,x}(i, j, l) &= E(x(k)x(k+i)x(k+j)x(k+l)) \\ m_{2,x}(i) &= E(x(k)x(k+i)) . \end{aligned}$$

Therefore, the influence of additive noise can be theoretically completely eliminated, given knowledge of its variance. Similar results are also obtained in the complex case.

6.4 Special case: the crucial role of over-parameterization

6.4.1 The singularity related to the over-parameterized case

In this section we consider the case where the true channel is all-pole of order $N-1$ (AP($N-1$)) and the equalizer FIR of order $M-1$ (FIR($M-1$)), $M > N$. This case does not seem to be of practical interest, since in practice the transmission channel is a FIR filter and therefore an equalizer of any length will never exceed the order of the channel's inverse impulse response. However, it merits special attention because it gives insight in the behaviour of the bilinear method. Suppose that the equalizer is FIR(N) ($M = N+1$). Now consider the following two equalizer settings

$$W^1 = \begin{bmatrix} C \\ \mathbf{0} \end{bmatrix} , \quad W^2 = \begin{bmatrix} \mathbf{0} \\ C \end{bmatrix} .$$

Obviously, both W^1 and W^2 are zero forcing, and therefore if we denote by θ^1 and θ^2 the corresponding bilinear parameter vectors, respectively, they will both satisfy (6.11):

$$\begin{cases} \mathbf{R}\theta^1 = \mathbf{d} \\ \mathbf{R}\theta^2 = \mathbf{d} . \end{cases}$$

Moreover, any vector of the form $\alpha\theta^1 + \beta\theta^2$, where α and β are scalars such that $\alpha + \beta = 1$ will also satisfy (6.11):

$$\begin{cases} \mathbf{R}(\alpha\theta^1 + \beta\theta^2) = \mathbf{d} \\ \alpha + \beta = 1 . \end{cases} \quad (6.33)$$

In the general case $M = N + L$, the eq. (6.11) will be satisfied by any vector of the form:

$$\theta = \sum_{i=1}^L \alpha_i \theta^i , \quad \text{with} \quad \sum_{i=1}^L \alpha_i = 1 , \quad (6.34)$$

where θ^i is the bilinear parameter vector corresponding to

$$W^i = \begin{bmatrix} \mathbf{0}_{1 \times (i-1)} \\ C \\ \mathbf{0}_{1 \times (L-i+1)} \end{bmatrix}$$

This means that when the equalizer is *over-parameterized* w.r.t. the inverse of the channel impulse response, the matrix \mathbf{R} is singular, and as a consequence, the problem (6.9) has an infinite number of solutions. Running an adaptive algorithm like the one in (6.22) or in (6.23) will converge to one of these solutions. The same will happen if one uses a batch technique to estimate \mathbf{R} and \mathbf{d} and a pseudo-inverse for the matrix inversion needed in (6.13). In all cases, the solution obtained will no longer correspond to a rank-1 matrix Θ : (6.17) and (6.20) will no longer yield the optimal ZF equalizer. Therefore the problem of ill-convergence appears (under a different form) also in the bilinear method. In the case of a FIR channel (which is a realistic one), one might think that the problem should not arise since the impulse response

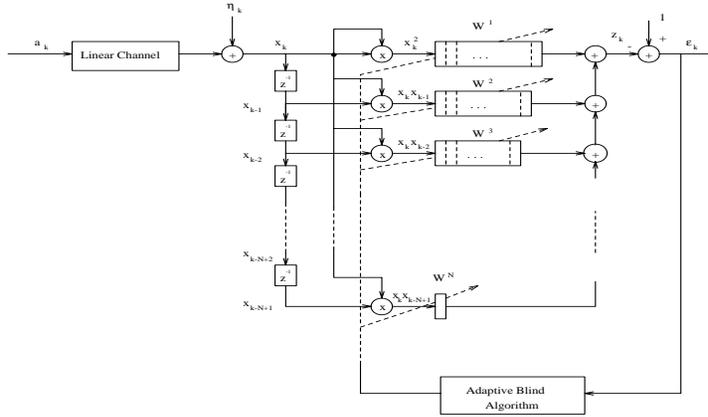


Figure 6.4: A multichannel implementation of the proposed blind equalization setup of the channel is theoretically of infinite length and thus an equalizer long enough should be able to approximate fairly well the ZF equalizer. However, as in practice the channel's inverse impulse response will be very close to zero out of a specific interval, if the equalizer's length is bigger than the number of samples in this interval, the same over-parameterization problem will exist and the matrix \mathbf{R} will be very ill-conditioned. This makes the choice of the equalizer length N in this case a problem of critical importance, as, unlike in conventional equalization, the longest possible length will not necessarily yield the best possible equalizer! In the next section we present a method to calculate the ZF equalizer from any solution of the form (6.34) when L is given.

6.4.2 A remedy: using a subspace fitting approach for the calculation of a ZF equalizer

Our task is to try to extract a ZF equalizer from the matrix Θ that corresponds to a vector θ of the form in (6.34). As will be shown, this will be possible due to the known specific structure of Θ . The channel inverse is assumed to be of length N and the equalizer of length $M = N + L$. Consider the eigenvalue decomposition of the matrix Θ :

$$\Theta = \sum_{i=1}^M \lambda_i V_i V_i^H, \quad (6.35)$$

where the (real) eigenvalues λ_i are in descending order of magnitude and V_i is the eigenvector corresponding to λ_i . When $M = N$ ($L = 0$), we saw that the ZF equalizer can be found based

on a rank-1 decomposition of Θ . When $L > 0$ we will try to determine the ZF equalizer based on a rank-($L+1$) decomposition of Θ by the following subspace fitting approach: we first construct an extended equalizer vector of $M + L$ entries:

$$W^e = [w_{-L} \cdots w_{-1} \ w_0 \cdots w_{M-1}]^T, \quad (6.36)$$

where $w_{-L} \cdots w_{-1}$ and $w_N \cdots w_{M-1}$ stand for the additional coefficients. If the length of the channel inverse is indeed N , then ideally

$$W^e = [0 \cdots 0 \ w_0 \cdots w_{N-1} \ 0 \cdots 0]^T.$$

We now create a Toeplitz matrix \mathcal{W} as follows:

$$\mathcal{W} = \begin{bmatrix} w_0 & w_{-1} & \cdots & w_{-L} \\ w_1 & w_0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & w_0 \\ \vdots & \vdots & \ddots & \vdots \\ w_{M-1} & w_{M-2} & \cdots & w_{M-1-L} \end{bmatrix}. \quad (6.37)$$

We will try to fit the matrix \mathcal{W} to a subspace of the space \mathcal{C}^M created by the first L eigenvectors of Θ . This fitting may be accomplished by minimizing the following criterion:

$$\min_{Q, W^e} \|\mathcal{W} - \mathcal{V}Q\|_F^2, \quad (6.38)$$

where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix ($\|A\|_F^2 = \text{tr}(A^H A)$), $Q \in \mathcal{C}^{(L+1) \times (L+1)}$ and \mathcal{V} is a matrix containing the $L + 1$ first eigenvectors of Θ :

$$\mathcal{V} = [V_1 \cdots V_{L+1}]. \quad (6.39)$$

Minimization w.r.t. Q only, yields:

$$Q = (\mathcal{V}^H \mathcal{V})^{-1} \mathcal{V}^H \mathcal{W}. \quad (6.40)$$

The problem (6.38) now becomes:

$$\begin{cases} \min_{W^e} \|P_{\mathcal{V}}^{\perp} \mathcal{W}\|_F^2 \\ \text{s.t. } \|W^e\|_2^2 = 1 \end{cases} = \begin{cases} \min_{W^e} \text{tr}\{\mathcal{W}^H P_{\mathcal{V}}^{\perp} \mathcal{W}\} \\ \text{s.t. } \|W^e\|_2^2 = 1 \end{cases}. \quad (6.41)$$

Noting that

$$\text{tr}\{\mathcal{W}^H P_{\mathcal{V}}^{\perp} \mathcal{W}\} = \text{tr}(\mathcal{W}^H \mathcal{W}) - \text{tr}\{\mathcal{W}^H P_{\mathcal{V}} \mathcal{W}\},$$

the problem (6.41) may be written as:

$$\begin{aligned} & \max_{W^e} \text{tr}\{\mathcal{W}^H \mathcal{V}(\mathcal{V}^H \mathcal{V})^{-1} \mathcal{V}^H \mathcal{W}\} \\ & = \max_{W^e} F(\mathcal{W}, \mathcal{V}) = \text{tr}\{\mathcal{W}^H \mathcal{V} \mathcal{V}^H \mathcal{W}\}. \end{aligned} \quad (6.42)$$

The quantity $F(\mathcal{W}, \mathcal{V})$ can be written as:

$$\begin{aligned} F(\mathcal{W}, \mathcal{V}) = & W^e H \begin{bmatrix} \mathbf{0}_{L \times (L+1)} \\ \mathcal{V} \end{bmatrix} \begin{bmatrix} \mathbf{0}_{L \times (L+1)} \\ \mathcal{V} \end{bmatrix}^H W^e + \\ & + W^e H \begin{bmatrix} \mathbf{0}_{L-1 \times (L+1)} \\ \mathcal{V} \end{bmatrix} \begin{bmatrix} \mathbf{0}_{L-1 \times (L+1)} \\ \mathcal{V} \end{bmatrix}^H W^e + \dots + \\ & + W^e H \begin{bmatrix} \mathbf{0}_{1 \times (L+1)} \\ \mathcal{V} \\ \mathbf{0}_{L \times (L+1)} \end{bmatrix} \begin{bmatrix} \mathbf{0}_{1 \times (L+1)} \\ \mathcal{V} \\ \mathbf{0}_{L \times (L+1)} \end{bmatrix}^H W^e, \end{aligned} \quad (6.43)$$

which gives the following expression for (6.42):

$$\max_{W^e} W^e H \left(\sum_{i=1}^{L+1} \Omega_i \Omega_i^H \right) W^e, \quad (6.44)$$

where

$$\Omega_i = \begin{bmatrix} \mathbf{0}_{(L-i+1) \times (L+1)} \\ \mathcal{V} \\ \mathbf{0}_{(i-1) \times (L+1)} \end{bmatrix}$$

The solution to (6.44) is:

$$W^e = \text{max eigenvector of } \Lambda = \sum_{i=1}^{L+1} \Omega_i \Omega_i^H. \quad (6.45)$$

When L is known, (6.45) will give the optimal ZF equalizer (in the absence of additive noise).

6.4.3 Discussion

The analysis presented in this section has thrown light to the mechanism that provokes the singularity of the matrix \mathbf{R} : clearly, it is the existence of ZF equalizers that are shifted versions of the same FIR equalizer that is responsible for the singularity of the matrix and therefore for the existence of undesired solutions. The fact that we have been able to develop a method that allows for the determination of the correct solution when the order mismatch is known, by taking into account different shifted versions of eigenvectors is also indicative of this phenomenon. Not astonishingly, the misbehaviour of the CMA 2-2 when used with finite length equalizers has also to do with the positioning of the strongest equalizer tap¹. Our work has revealed the same phenomenon by its reflection to the singularity of the matrix \mathbf{R} .

The fact that the order mismatch L needs to be known for the above proposed subspace-fitting method may give rise the following question: if L was to be known, then why not

¹In [LD94] it is proven that within a cone that preserves the position of the strongest equalizer tap, the CMA 2-2 is convex. In [BZA94] and in [VAK93], globally convergent methods have been proposed by constraining the position of the strongest tap to remain fixed.

use from the beginning the correct equalizer length? The answer to this is that in practical implementations the equalizer length is usually fixed. We can then proceed as follows: after the convergence of θ , one may apply (in a batch mode) the subspace-fitting method given above, for different values of L , and choose by trial and error the one that gives the best performance.

6.5 Computer simulations

The above claimed theoretical performance of the proposed methods has been verified through computer simulations. The following cases have been considered.

6.5.1 The correctly parameterized case

In the case of all-pole noiseless channels of the same order as the equalizer, as expected, the optimal (ZF) equalizer was obtained with the help of (6.20) in all cases (irrespective of initialization) by using either an adaptive (6.22), (6.23) or a batch technique (6.13), whereas the CMA converged as well to other equalizer settings for some initializations (ill-convergence).

Figure 6.5 shows a comparative simulation that verifies this fact. In the left part, the starting and ending points of the equalizer setting are shown for the bilinear method implemented with the LMS algorithm, whereas in the right part the corresponding figure is shown for the standard CMA algorithm. One may note how the bilinear method converges to its unique optimal setting whereas the CMA (as already seen in chapter 2) may end up (depending on its initialization) on two undesired local minima. The channel used is an all-pole filter with coefficients $[c_0 \ c_1] = [1 \ 0.25]$ and 40 different initializations on a circle of radius 2 are used.

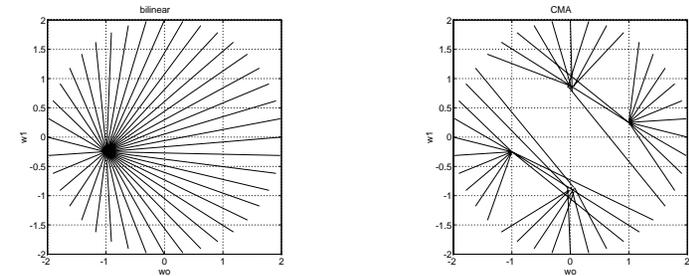


Figure 6.5: Comparative simulations for the noiseless case

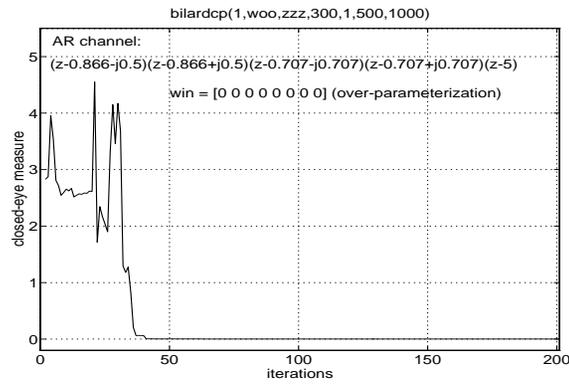


Figure 6.6: A simulation for the over-parameterized case

6.5.2 Noiseless over-parameterized case: subspace fitting technique

In the case of noiseless $AP(N-1)$ channels and an over-parameterized equalizer $FIR(M-1)$, $M = N + L > N$, the optimal (ZF) equalizer was obtained with the help of (6.45) instead of (6.20). This of course implies knowledge of L . An example of this performance is shown in figure 6.6, where the equalizer input process is AR of order 4, whereas the equalizer order is 6 (the roots of the corresponding polynomial are shown on the figure). We use an LMS algorithm and the eigendecomposition method mentioned in section 4, considering the order mismatch ($L = 2$) to be known. The fact that the algorithm converges to its optimal setting verifies the success of our subspace-fitting method.

6.5.3 FIR noisy channels

The standard technique

In the case of FIR channels, the method is sensible to the choice of the equalizer length, as already explained. However, for a given length, one can still use (6.45) for different values of L ranging from 0 to $M-2$, and then choose the best among the $M-1$ derived equalizers by evaluating for each the constant modulus criterion and choosing the one that best satisfies it. In our simulations, there was always one among the equalizers that sufficiently opens the system's eye, provided of course that the equalizer's length is long enough to be able to approximate well the channel's inverse. It was also observed that the influence of additive white Gaussian noise resulting in an SNR up to 20dB did not in general cause serious damage to the obtained solution. A realistic simulation is shown in the upper and lower part of figure 6.7 where one may see the evolution of the closed-eye measure of a linear noisy (SNR=30

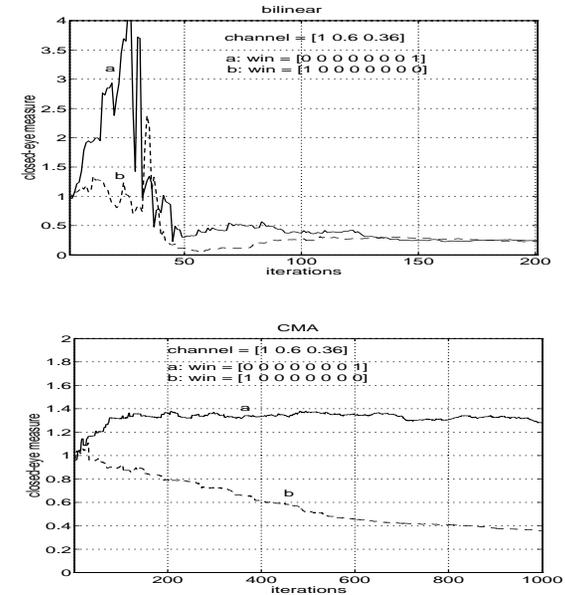


Figure 6.7: Comparative simulations for the noisy FIR case

dB) communications system using the algorithm in (6.23) ($\lambda = 1$) and CMA, respectively. The channel's impulse response is $[1 \ 0.6 \ 0.36]$ and an equalizer W of 8 taps is used (28 taps for θ). One may see how the opening of the system's eye may be achieved for 2 different initializations by using a bilinear algorithm, while CMA gets trapped by a local minimum for one of these initializations. The successful performance of the bilinear technique has of course to do with the fact that the number of taps used matches well the length of the impulse response of the channel inverse.

The bilinear rank-1 technique

We have also tested the bilinear rank-1 approach (6.28). We considered a $FIR(1)$ channel with impulse response $[c_0 \ c_1] = [1 \ 0.5]$ and an equalizer of length 2. Additive white Gaussian noise (SNR=30dB) is added to the channel output. Figure 6.8 shows the convergence trajectories for 30 different initializations on a circle of radius 1. The algorithm was run over 3000 iterations for each initialization, and we have plotted the equalizer setting every 30 iterations. The stepsize used is $\mu = 0.01$. Note that the algorithm converges only to one of the two

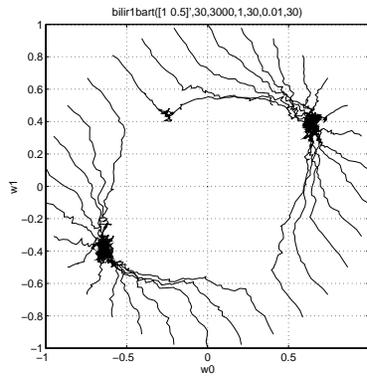


Figure 6.8: A simulation of the bilinear rank-1 algorithm symmetrical settings, despite the fact that it is not proven to be globally convergent and despite the presence of the additive noise.

6.6 Conclusions and further discussion

In this chapter we have presented a new, namely, a bilinear formulation for the problem of equalization of the CMA 2-2 type. This formulation has allowed us to express in a different way the CMA 2-2 cost function and offers the advantage of leading to globally convergent BE techniques.

The same formulation has been independently presented in literature during the last two years. In [JWC92], the bilinear formulation has been introduced (under the name of “Kronecker product” Godard equalizer), however the problem of over-parameterization and methods to overcome it have not been discussed there. The formulation also appears in [VP94], where it is applied to a different context, namely to the setup of constant modulus arrays. An interesting method based on eigenvalue decomposition is also presented there. Finally, the concept appears also in [Kennedy94]. Interesting aspects related to the rank of the matrix \mathbf{R} appear also there, as well as a test to determine the singularity of the matrix. This can be used to determine the order mismatch and then be combined with our subspace fitting approach. In chapter 8 of the thesis we will also discuss the application of the bilinear formulation to a multichannel context. The work contained in this chapter has partially appeared in [PS93a] and [PS94c].

PART II

BLIND MULTICHANNEL IDENTIFICATION AND EQUALIZATION

Chapter 7

New blind techniques for multichannel identification/equalization

IN this chapter we analyze a multichannel setup for blind equalization. This setup is applicable to the case of fractionally-spaced equalization and to the case of reception through an antenna array. A review of some recent results in this domain is given. The important issue of cyclostationarity in this context is presented and the identifiability of the multichannel setup based only on Second Order Statistics (SOS) is discussed. We are interested in both linear and Maximum Likelihood (ML) equalization. In the first category, we show how MMSE and ZF equalizers can be obtained by multichannel linear prediction and discuss several related aspects. In the second, we propose ML techniques and analyze their performance by calculating their corresponding Cramer-Rao bounds.

7.1 Introduction

In the first part of this thesis we have been interested in linear channels whose discrete-time baseband equivalent corresponds to a single-input-single-output (SISO) system. Even though this is a standard structure for a digital communication system, there exist other setups also that correspond to either single-input-multiple-output (SIMO) or even multiple-input-multiple-output (MIMO) systems. For example, consider first the typical i/o relation for a linear channel with continuous-time impulse response $h(t)$

$$x(t) = \sum_{i=-\infty}^{\infty} a(i)h(t - iT) + v(t) . \quad (7.1)$$

This equation (if one ignores the additive noise $v(t)$) corresponds to a SISO continuous-time signal $x(t)$. As we saw in the introduction of the first part of the thesis, the sampling of the received continuous-time signal at the symbol rate $\frac{1}{T}$ results in the following discrete-time equivalent of the above i/o relation 7.1)¹:

$$x(k) = \sum_{i=-\infty}^{\infty} a(i)h(k - i) + v(i) . \quad (7.2)$$

An alternative way to sample the received continuous-time signal $x(t)$ is to use a sampling

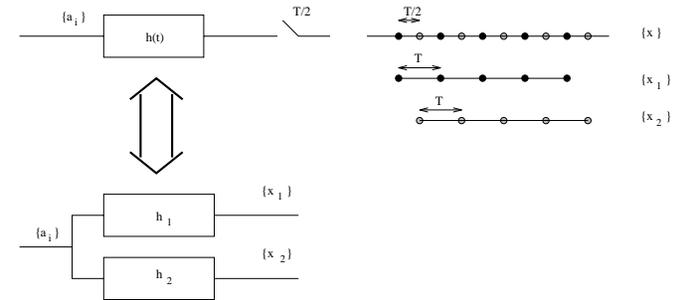


Figure 7.1: $T/2$ fractional spacing

rate greater than the symbol rate $\frac{1}{T}$. This is the principle of fractionally-spaced equalizers which are often used in practice. Consider for example a sampling rate of $\frac{2}{T}$. In this case one may separate the even from the odd samples of $x(k)$ and write down the i/o relationship corresponding to each one of these two discrete signals:

$$\begin{aligned} x_1(k) &= \sum_{i=-\infty}^{\infty} a(i)h_1(k - i) \\ x_2(k) &= \sum_{i=-\infty}^{\infty} a(i)h_2(k - i) , \end{aligned} \quad (7.3)$$

where $\{x_1\}$, $\{x_2\}$ and $\{h_1\}$, $\{h_2\}$ account for the even and odd discrete time received sequences and impulse responses, respectively. A graphic way to represent (7.3) can be shown in figure 7.1. As can be seen in the figure, an equivalent way to obtain the two discrete sequences $\{x_1\}$ and $\{x_2\}$ is to implement a “parallel” configuration in which the input sequence $\{a_i\}$ is passed through the two discrete channel impulse responses that consist of the even and odd samples of the discrete-time impulse response $\{h_i\}$. Therefore the fractional spacing results to a SIMO structure, or in other words in a *multichannel* setup.

¹In this chapter indices will be denoted with parentheses instead of subscripts. Subscripts will be reserved to denote the phase of a channel or an equalizer.

A similar multichannel setup may arise when we consider reception through multiple antennas. In this case, the different antennas receive different continuous-time waveforms, due to the different physical channels that separate them from the transmitter. However after sampling (e.g. at the symbol rate) the corresponding discrete sequence of each receiving antenna can be modeled as the output of a discrete time impulse response. An example for the case of two receiving antennas can be shown in figure 7.2. The same SIMO structure that models fractional spacing is therefore also suitable for the modeling of symbol-rate transmission through several antennas. Both operations mentioned above, i.e. fractional

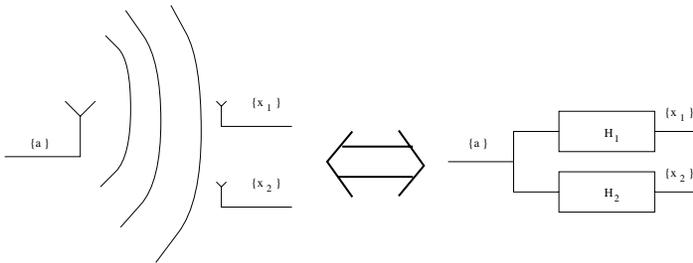


Figure 7.2: Another SIMO setup: multiple antennas

spacing and reception through an antenna array can be viewed as different ways to introduce some *diversity* when receiving the continuous-time waveform. In the case of fractional spacing the diversity is due to the fact that we sample at a higher rate than the one strictly needed for the retrieval of the transmitted information, whereas in the case of antenna array reception, the diversity is due to the fact that we receive multiple waveforms. The introduction of diversity at the receiver of a communication system has been shown to have several advantages, for example fractional-spaced receivers allow for better equalization for a fixed equalizer length, antenna array reception allows for the estimation of the directions of arrival (DOA) of the received waveforms, etc. In this part of the thesis we will be interested in the influence of such diversity on the issues of *blind* identification and equalization of a communication channel. As will be shown in the sequel, a key element of the multichannel setup is the issue of *stationarity* of the received signal, which is treated in the next section.

7.2 Fractional spacing and stationarity

7.2.1 Stationarity and cyclostationarity

We will confine ourselves to fractionally-spaced receivers (the extension of the results to the case of antenna arrays being straightforward). We consider uniform sampling at a rate $\frac{m}{T}$,

where $m > 1$ is an integer that we call *Oversampling Factor* (OF). This corresponds to m received *phases* (discrete sequences), each one of which is described by

$$x_n(k) = \sum_{i=-\infty}^{\infty} a(i)h_n(k-i) + v_n(k) \quad , \quad n = 1, \dots, m \quad , \quad (7.4)$$

where $\{h_n(i)\}$ corresponds to the n^{th} phase of the channel impulse response $h(t)$. The degenerate case of symbol-rate sampling corresponds to the choice $m = 1$ in (7.4). We are interested in the deconvolution problem of finding the transmitted sequence $\{a_i\}$ from the samples of the m received phases $\{x_n\}$, $n = 1 \dots m$. A particularity of the deconvolution problem as appearing in (7.1) is that $x(t)$ is a cyclostationary signal. This is due to the kind of linear modulation we have assumed ($h(\cdot)$ is a narrow-band pulse) and to the stationarity of both the discrete input process $\{a_i\}$ and additive noise $v(t)$.

At this point we give some definitions needed for the characterization of cyclostationary signals as given by Gardner [Gar94], [Gar91].

Cyclic correlation function: The cyclic correlation function $R_x^\alpha(\tau)$ of a deterministic signal $x(t)$ is defined as

$$R_x^\alpha(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t + \frac{\tau}{2}) x^*(t - \frac{\tau}{2}) e^{-j2\pi\alpha t} dt \quad . \quad (7.5)$$

In (7.5) α denotes the frequency and τ the time lag at which $R_x^\alpha(\tau)$ is evaluated. In the case of stochastic processes we use the following modified definition for $R_x^\alpha(\tau)$:

$$R_x^\alpha(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} E(x(t + \frac{\tau}{2}) x^*(t - \frac{\tau}{2})) e^{-j2\pi\alpha t} dt \quad . \quad (7.6)$$

The cyclic correlation function is a measure of cyclostationarity of a signal according to the following definition:

Second order cyclostationarity: $x(t)$ is said to be second-order cyclostationary iff its cyclic correlation function $R_x^\alpha(\tau)$ is not identically zero for all $\alpha \neq 0$. All nonzero values of α for which $R_x^\alpha(\tau) \neq 0$ are called *cycle frequencies*. The set of all cycle frequencies is called *cycle spectrum*.

In the sequel we will use the term “cyclostationary” instead of “second-order cyclostationary”. Qualitatively, a cyclostationary signal is a random process whose statistical parameters vary in time with single or multiple periodicities. A signal $x(t)$ for which $R_x^\alpha(\tau)$ is identically zero for all $\alpha \neq 0$ will be called “purely stationary”. We are interested in the stationarity of the continuous-time received signal of a linear communication channel as given by (7.1). We assume that $\{a_i\}$ is a white discrete-time stochastic process with variance σ_a^2 (and therefore wide-sense stationary). The additive noise $v(t)$ is also assumed to be wide-sense stationary

(not necessarily white) with variance $R_v(0) = \sigma_v^2$. Under the above assumptions we can prove the following lemma:

Lemma 7.1 The continuous-time stochastic process $x(t)$ defined in (7.1) is a cyclostationary signal.

Proof: (see also [Din94])

From (7.6), it is clear that $R_x^\alpha(\tau)$ is the Fourier coefficient at frequency α of the autocorrelation function $R_x(t_1, t_2)$ evaluated at $t_1 = \frac{\tau}{2}$, $t_2 = -\frac{\tau}{2}$. For $x(t)$ as in (7.1) we have

$$\begin{aligned} R_x\left(\frac{\tau}{2}, -\frac{\tau}{2}\right) &= E \left(\left(\sum_{i=-\infty}^{\infty} a(i)h(t - iT + \frac{\tau}{2}) + v(t + \frac{\tau}{2}) \right) \left(\sum_{k=-\infty}^{\infty} a^*(k)h^*(t - kT - \frac{\tau}{2}) + v(t - \frac{\tau}{2}) \right) \right) \\ &= \sum_{k=-\infty}^{\infty} \sigma_a^2 h(t - kT + \frac{\tau}{2})h^*(t - kT - \frac{\tau}{2}) + R_v\left(\frac{\tau}{2}, -\frac{\tau}{2}\right), \end{aligned}$$

and therefore

$$R_x\left(\frac{\tau}{2}, -\frac{\tau}{2}\right) = R_x\left(\frac{\tau}{2} + T, -\frac{\tau}{2} + T\right). \quad (7.7)$$

Hence $R_x(\frac{\tau}{2}, -\frac{\tau}{2})$ is periodic with fundamental period T . This periodicity implies that $R_x(\frac{\tau}{2}, -\frac{\tau}{2})$ has non-zero frequency content at frequencies $\frac{m}{T}$, m being an integer. Therefore, the Fourier coefficient of $R_x(\frac{\tau}{2}, -\frac{\tau}{2})$ (which is $R_x^\alpha(\tau)$) is nonzero at all frequencies $\alpha = \frac{m}{T}$. This means that $x(t)$ in (7.1) is cyclostationary with period T . Its cycle spectrum is the discrete set $\{\alpha_m = \frac{m}{T}, m = 1, 2, \dots\}$. \square

The cyclostationarity of $x(t)$ in (7.1) is of great importance in what concerns the deconvolution problem we cited above, as will be clear in the sequel. Before that, it is interesting to see how the cyclostationarity of $x(t)$ reflects in the stationary characteristics of sampled versions of $x(t)$. We will prove the following two lemmas:

Lemma 7.2 The discrete sequence $\{x_i\}$ obtained by sampling $x(t)$ at the symbol rate $\frac{1}{T}$ is purely (wide-sense) stationary.

Lemma 7.3 The discrete sequence $\{x_i\}$ obtained by sampling $x(t)$ at the rate $\frac{m}{T}$, $m > 1$ is cyclostationary.

Proof:

We consider sampling of $x(t)$ with sampling period T_s , where T_s obeys

$$T_s = \frac{T}{m}. \quad (7.8)$$

This results in the discrete sequence $\{x(nT_s)\}$. In the case of discrete-time stochastic processes the cyclic correlation function is defined as [Gar94]

$$R_x^\alpha(kT_s) = \langle r_x(kT_s) e^{-j2\pi\alpha nT_s} \rangle e^{-j\pi\alpha kT_s}, \quad (7.9)$$

where the discrete autocorrelation function $r_x(\cdot)$ is defined as

$$r_x(kT_s) = E(x(nT_s + kT_s)x^*(nT_s)), \quad (7.10)$$

and $\langle \cdot \rangle$ denotes discrete-time averaging. In order to determine whether $\{x(nT_s)\}$ is cyclostationary or purely stationary we have to find if there exist nonzero frequencies $\alpha \neq 0$ for which $R_x^\alpha(kT_s)$ is not identically zero. With the employed notation, (7.1) takes the form (in the absence of noise)

$$x(nT_s) = \sum_{i=-\infty}^{\infty} a(iT)h(nT_s - iT). \quad (7.11)$$

The autocorrelation $r_x(kT_s)$ can then be found to be given by

$$r_x(kT_s) = \sum_{i=-\infty}^{\infty} \sigma_a^2 h((n+k)T_s - iT)h^*(nT_s - iT). \quad (7.12)$$

Therefore $\{r_x(kT_s)\}$ is periodic with period T and contains m samples into each period. Now consider the quantity $\langle r_x(kT_s) e^{-j2\pi\alpha nT_s} \rangle$ appearing in (7.9). As $\{r_x(kT_s)\}$ contains m (due to (7.8)) samples during a period of length T , this quantity equals

$$\langle r_x(kT_s) e^{-j2\pi\alpha nT_s} \rangle = \frac{1}{m} \sum_{k=0}^{m-1} r_x(kT_s) e^{-j2\pi\alpha nT_s}. \quad (7.13)$$

If one expresses the frequency α as

$$\alpha = \frac{k}{T}, \quad (7.14)$$

one recognizes that

$$\langle r_x(kT_s) e^{-j2\pi\alpha nT_s} \rangle = \{\mathcal{F}(r_x)\}\left(\frac{n}{T}\right), \quad (7.15)$$

where $\mathcal{F}(r_x)$ denotes the DFT of $r_x(kT_s)$. In other words, the quantity $\langle r_x(kT_s) e^{-j2\pi\alpha nT_s} \rangle$ equals (up to a scalar factor) the coefficient of the DFT of $r_x(kT_s)$ corresponding to the frequency $\frac{n}{T}$. When we have sampling at the symbol rate, $m = 1, T = T_s$ and there is only one sample of the DFT in one period, corresponding to the zero frequency. In this case the quantity $\langle r_x(kT_s) e^{-j2\pi\alpha nT_s} \rangle$ is nonzero only for $\alpha = 0$ and this will hold also for $R_x^\alpha(kT_s)$. Therefore the discrete process $\{x(nT)\}$ is purely (second-order) stationary. This concludes the proof of lemma 7.2. \square

On the other hand, when $m > 1$, there exist apart from $\alpha = 0$ another $m - 1$ samples of the DFT in one period ($\alpha = \frac{k}{T}$, $k = 1, \dots, m - 1$), which correspond to nonzero values for $R_x^\alpha(kT_s)$. Therefore when $m > 1$ the discrete process $\{x(nT_s)\}$ is cyclostationary. This concludes the proof of lemma 7.3 \square

A graphic representation of the above interpretation can be found in figure 7.3.

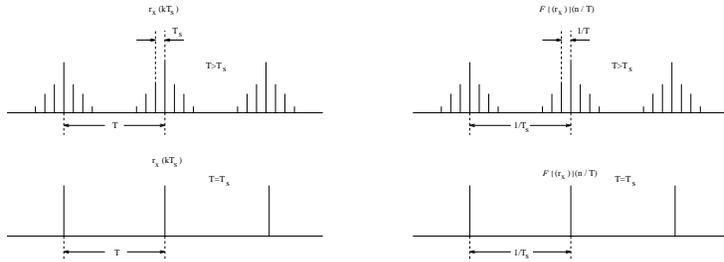


Figure 7.3: Time-frequency representation of the cyclostationary and stationary autocorrelation functions

According to lemmas 7.2 and 7.3 it is clear that the value of the OF is of critical importance in what concerns the stationarity of the oversampled signal: this will be purely stationary when $OF = 1$ but it will be cyclostationary when $OF > 1$. This fact is a key-element for the identification of the channel based on its output statistics, as will be made clear in the sequel.

7.2.2 Vector stationarity

A different way to represent the output of the fractionally-spaced channel is the following: instead of considering the output to be scalar, we may group together in a vector of m entries the m consecutive samples that belong to the same symbol period (these are the m discrete-time phases). We therefore define the vector output \mathbf{x}_k as

$$\mathbf{x}_k = [x_1(k) \ x_2(k) \ \dots \ x_m(k)]^T . \quad (7.16)$$

We now give the following definition:

Vector-stationarity: We call a vector process (wide-sense) stationary (in the vector sense), when the scalar process corresponding to each of its entries is a (wide-sense) stationary process.

In the sequel we will use the term “vector-stationary” instead of “wide-sense vector stationary”. According to the following definition and due to the stationarity of each of the phases of (7.4) we have the following lemma:

Lemma 7.4: The vector output of the communication channel \mathbf{x}_k defined in (7.16) is stationary in the vector sense.

The lemmas stated in this section are summarized in figure 7.4. As it will be clear in the sequel, these stationarity properties have to do with the identifiability of the over-sampled channel from output measurements.

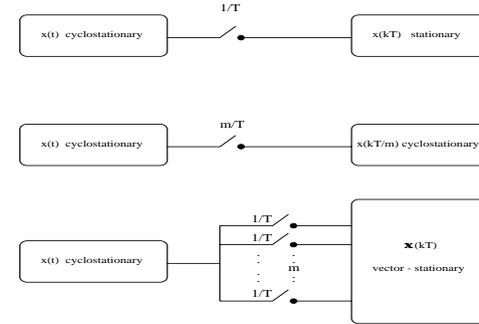


Figure 7.4: The effect of sampling and oversampling on the stationarity of $x(t)$

7.3 Channel identifiability from output measurements

7.3.1 Phase ambiguity and determinacy with SOS

We saw in chapter 2 that a necessary condition for the identifiability of a linear channel sampled at the symbol rate from its output SOS is that the channel should be minimum phase. This conclusion has been drawn by the fact that the PSD of the output contains no phase information. For example, if the channel and equalizer transfer functions are denoted by $H(f)$ and $F(f)$ respectively, then if one tries to equalize the PSD's of the input and the output of the overall system channel-equalizer, one gets

$$|H(f)| |F(f)| = 1 , \quad (7.17)$$

due to the i/o relation for the overall system $G(f) (G(f) = H(f)F(f))$:

$$S_{yy}(f) = |G(f)|^2 S_{aa}(f) . \quad (7.18)$$

According to equations (7.17),(7.18), only the magnitude of the transfer function of the ideal equalizer can be identified from the output PSD and this reflects the fact that the PSD itself carries no information about the channel phase. The phase can be uniquely identified only in the special case that the channel is known to be MP. The fact that the PSD of a stationary signal contains all information about its second-order statistics (SOS) has led to the conclusion that equalization (or identification) of a NMP system is not possible when only the SOS of the

output are given. This is true for stationary signals; however in the case of cyclostationary signals, all second-order statistical information is not contained in the PSD. In this case, important information is also contained in the SCD of the cyclostationary signal. The i/o relation of the system $G(f)$ in terms of SCD is (see [Gar94])

$$S_{yy}^\alpha(f) = G(f + \frac{\alpha}{2}) G^*(f - \frac{\alpha}{2}) S_{aa}^\alpha(f) , \quad (7.19)$$

where the SCD S_{xx}^α of a signal $x(t)$ is defined as

$$S_{xx}^\alpha(f) = S_x^\alpha(f) = \sum_{\tau=-\infty}^{\infty} R_x^\alpha(\tau) e^{-j2\pi f\tau} . \quad (7.20)$$

The important feature of eq. (7.19) is that, in contrast to (7.18) which shows the absence of phase information in the PSD, (7.19) shows the existence of phase information in the SCD of y , provided that S_{yy}^α is nonzero for at least one frequency $\alpha \neq 0$. But this is the condition for a cyclostationary signal!² This implies that identification (or equalization) of a NMP system is not a priori impossible based exclusively on SOS, provided that the received signal is cyclostationary. Such a case may arise as we saw in the previous section when the received signal is oversampled. We therefore have the following lemma:

Lemma 7.5: When the received signal of a linear communication channel is oversampled, information about the channel phase is contained in the SCD of the sampled process and therefore in its SOS.

A theoretical way to identify completely the channel transfer function can be found for example in [Din94]. This proves the identifiability of the oversampled channel based on SOS.

7.3.2 Identification by vector - spectral factorization

Another point of view is the following (see also [Lou94]). Consider the vector output \mathbf{x}_k defined in (7.16). As already noted, this is a stationary vector-process. For a stationary $m \times 1$ vector (\mathbf{x}), its correlation matrix is defined as

$$\mathbf{R}_{\mathbf{xx}} = E(\mathbf{xx}^H) , \quad (7.21)$$

and the corresponding power spectral density is also a $m \times m$ matrix each element of which is the Fourier transform of the corresponding element of $\mathbf{R}_{\mathbf{xx}}$:

$$\mathbf{S}_{\mathbf{xx}}(f) = \mathcal{F}(\mathbf{R}_{\mathbf{xx}}) . \quad (7.22)$$

The stationary output vector \mathbf{x}_k is a discrete-time process and therefore we use the z instead of the Fourier transform. The i/o relationship of the SIMO system shown in figure 7.4 in

²This condition also explains why no phase information can be extracted from SOS of a purely stationary signal, even if its SCD is used, since for $\alpha = 0$ (7.19) takes the same form as (7.18).

terms of the PSD of the oversampled received signal is given as

$$\mathbf{S}_{\mathbf{xx}}(z) = \mathbf{H}(z)\mathbf{H}^H(z^{-*}) S_{aa}(z) , \quad (7.23)$$

where $\mathbf{H}(z)$ is the z -transform of the multichannel $\mathbf{h}(k)$:

$$\mathbf{h}(k) = \begin{bmatrix} h_1(k) \\ \vdots \\ h_m(k) \end{bmatrix} , \quad (7.24)$$

($\mathbf{H}(z) = \sum_{k=0}^{N-1} \mathbf{h}(k)z^{-k}$: FIR assumption). In (7.23) the input power spectral density S_{aa} is scalar since the input is scalar too. According to (7.23) $\mathbf{S}_{\mathbf{xx}}$ is an $m \times m$ polynomial spectral density of rank 1. Before giving an important theorem that has to do with the identifiability of the multichannel from its PSD, we need to give the following definition:

Minimum phase property in the vector sense A $m \times 1$ rational transfer function $\mathbf{K}(z)$ is said to be minimum phase iff there exists no z , $|z| > 1$ for which all the m entries of $\mathbf{K}(z)$ become simultaneously zero:

$$\mathbf{K}(z) \neq \mathbf{0}_{m \times 1} \quad \forall z \in \{z : |z| > 1\} . \quad (7.25)$$

The following spectral factorization theorem has been stated for the general case of a rational $m \times m$ spectral density:

Theorem 7.1: Consider an $m \times m$ rational spectral density matrix $\mathbf{S}_{\mathbf{xx}}$ of rank 1. Then there exists a rational $m \times 1$ transfer matrix $\mathbf{K}(z)$ that is causal, stable, minimum-phase, *unique* up to a unitary constant, of minimal McMillan degree³ $deg(\mathbf{K}) = \frac{1}{2}deg(\mathbf{S}_{\mathbf{xx}})$ such that it factorizes $\mathbf{S}_{\mathbf{xx}}$ as

$$\mathbf{S}_{\mathbf{xx}} = \mathbf{K}(z)\mathbf{K}^H(z^{-*}) . \quad (7.26)$$

Applying the above theorem to the PSD $\mathbf{S}_{\mathbf{xx}}$ of the multichannel $\mathbf{h}(k)$ (which has a polynomial transfer function in the FIR case) we obtain the following theorem:

Theorem 7.2: If the multichannel $\mathbf{h}(k)$ is minimum phase (in the above sense), the factorization of the output spectral density $\mathbf{S}_{\mathbf{xx}}$ (according to (7.23) and theorem 7.2) gives the channel transfer function $\mathbf{H}(z)$ up to a complex constant:

$$\mathbf{K}(z) = \sigma_\alpha e^{j\phi} \mathbf{H}(z) . \quad (7.27)$$

The importance of this theorem is that it proves explicitly the identifiability of the SIMO system described above based on second order statistics of the output: the channel is identified as

³See [Lou94] for more details.

the spectral factor of the output multichannel PSD. This is therefore a method to identify the multichannel transfer function in the frequency domain. Eq. (7.26) is a counterpart of (7.19) in the following sense: (7.19) shows the existence of phase information in the PSD of the cyclostationary scalar output $x(kT_s)$, whereas (7.26) shows the existence of channel information in the PSD of the stationary vector $\mathbf{x}(kT)$. These are two different ways to prove the identifiability of the multichannel from SOS of its output. One is based on scalar cyclostationarity, the other on vector stationarity. In fact, theorem 7.2 has also the following interesting interpretation: in the case of a scalar stationary process (e.g. the channel output sampled at the symbol rate), the channel can be identified from SOS up to its minimum-phase equivalent, i.e. SOS-based identification is only achieved if it is minimum phase. According to theorem 7.2, this property is immediately generalized in the case of a stationary vector output: the multichannel can be identified from its output PSD only if it is minimum-phase in the sense of the above definition. One might think that due to that, nothing has been gained from the oversampling of the channel output in the sense that the MP requirement is still there. However one must realize that the MP requirement is much less strong in the vector than in the scalar case. For example, if the different channels $h_i(k)$, $i = 1, \dots, m$ have no zeros in common (i.e. there exists no z such that all entries of $\mathbf{H}(z)$ are simultaneously 0), then it is guaranteed that the multichannel is MP! This is another important feature provided by theorem 7.2: it provides a necessary condition for multichannel ID based on output SOS.

The above identification results have been given in the frequency domain. In the sequel we will see how multichannel identification can be achieved in the time domain.

7.4 Zero Forcing equalization of the SIMO channel

7.4.1 Fractionally-spaced equalization

At this point we introduce the notation and the basic assumptions that will be used through the rest of this chapter.

The continuous-time channel $h(t)$ is assumed to be FIR with duration of approximately NT . The oversampling factor (OF) is assumed to be m and the sampling instants for the received signal $x(t)$ in (7.1) are $t_0 + T(k + \frac{j}{m})$ for integer k and $j = 0, 1, \dots, m-1$. t_0 represents the initial sampling time instant. In principle, it suffices to introduce a restricted $t_0 \in [0, T)$ to be fully general. However, we shall take $t_0 = t'_0 + dT$ where $t'_0 \in [0, T)$ in order to incorporate also an inherent delay due to transmission. d is chosen as the smallest integer such that

$$\left[h(t'_0 + dT) \cdots h(t'_0 + (d + \frac{m-1}{m})T) \right] \neq \mathbf{0} . \quad (7.28)$$

The channel being causal implies that d will be nonnegative.

We introduce now the *polyphase description* of the received signal, the channel discrete-time

impulse response and the additive noise, respectively, as follows

$$\begin{aligned} x_j(k) &= x(t_0 + T(k + \frac{j}{m})) \\ h_j(k) &= h(t_0 + T(k + \frac{j}{m})) \quad j = 0, 1, \dots, m-1, \\ v_j(k) &= v(t_0 + T(k + \frac{j}{m})) \end{aligned} \quad (7.29)$$

where k is an integer. The oversampled received signal can now be represented in vector form at the symbol rate as

$$\mathbf{x}(k) = \sum_{i=0}^{N-1} \mathbf{h}(i) a_{k-i} + \mathbf{v}(k) = \mathbf{H}_N A_N(k) + \mathbf{v}(k), \quad (7.30)$$

where $\mathbf{x}(k)$, $\mathbf{h}(k)$, $\mathbf{v}(k)$ are defined as

$$\mathbf{x}(k) = \begin{bmatrix} x_1(k) \\ \vdots \\ x_m(k) \end{bmatrix}, \mathbf{v}(k) = \begin{bmatrix} v_1(k) \\ \vdots \\ v_m(k) \end{bmatrix}, \mathbf{h}(k) = \begin{bmatrix} h_1(k) \\ \vdots \\ h_m(k) \end{bmatrix}. \quad (7.31)$$

The sub-channels are defined as

$$H_i = [h_i(0) \cdots h_i(N-1)], \quad (7.32)$$

and the channel matrix \mathbf{H}_N is a $m \times N$ matrix defined as

$$\mathbf{H}_N = \begin{bmatrix} h_1(0) & \cdots & h_1(N-1) \\ \vdots & \cdots & \vdots \\ h_m(0) & \cdots & h_m(N-1) \end{bmatrix} = [\mathbf{h}(0) \cdots \mathbf{h}(N-1)] = \begin{bmatrix} H_1 \\ \vdots \\ H_m \end{bmatrix}. \quad (7.33)$$

Finally, we denote by $A_N(k)$ the $N \times 1$ symbol regressor:

$$A_N(k) = [a_k^H \cdots a_{k-N+1}^H]^H. \quad (7.34)$$

We formalize the finite duration NT assumption of the channel as follows

FIR assumption: (single user case)⁴ $\mathbf{h}(0) \neq \mathbf{0}$, $\mathbf{h}(N-1) \neq \mathbf{0}$ and $\mathbf{h}(i) = \mathbf{0}$ for $i < 0$ or $i \geq N$.

In order to equalize the fractionally-spaced channel, we will use a fractionally-spaced equalizer so that an m/T -rate equalizer corresponds to each sub-channel. The channel-equalizer structure in the case of an oversampling factor $m = 2$ will then look as in figure 7.5.

In the frequency domain, the z -transform of the channel response at the sampling rate $\frac{m}{T}$ is given as

$$H(z) = \sum_{j=1}^m z^{-(j-1)} H_j(z^m). \quad (7.35)$$

⁴A generalized FIR assumption for the case of multiple users (MIMO system) can be found in [Slo94b].

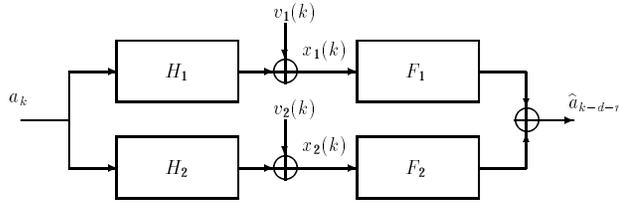


Figure 7.5: Polyphase representation of the T/m fractionally-spaced channel and equalizer for $m = 2$.

Similarly, the z -transform of the fractionally-spaced ($\frac{T}{m}$) equalizer can also be decomposed into its polyphase components:

$$F(z) = \sum_{j=1}^m z^{(j-1)} F_j(z^m) . \quad (7.36)$$

Although the equalizer defined by (7.36) is slightly noncausal, this does not cause a problem because the discrete-time filter is not a sampled version of an underlying continuous-time function. In fact, a particular equalizer phase $z^{(j-1)} F_j(z^m)$ follows in cascade the corresponding channel phase $z^{-(j-1)} H_j(z^m)$ so that the cascade $F_j(z^m) H_j(z^m)$ is causal. We assume the equalizer phases to be causal and FIR of length L :

$$F_j(z) = \sum_{k=0}^{L-1} f_j(k) z^{-k}, \quad j = 1, \dots, m. \quad (7.37)$$

So far we have obtained multiple received signals by unraveling the multiple phases of the oversampled continuous-time received signal. As already mentioned, an alternative way to arrive at the same picture as in figure 7.5 is to have several antennas. Each of the antenna signals can then be oversampled or not. Hence, the total number of received signals is the product of the number of antennas times the oversampling factor. Some of the frequency-domain interpretations we have given above and that will follow below only apply to the case of an oversampled signal coming from one antenna only.

The SIMO deconvolution problem now reduces to the calculation of the optimal equalizer coefficients $f_j(k)$, $j = 1, \dots, m$, $k = 0, \dots, N-1$. In the next subsection we are interested in the noiseless case. The optimal equalizer then is the one that completely eliminates intersymbol interference i.e. a zero forcing equalizer.

7.4.2 FIR Zero-Forcing (ZF) Equalization

We introduce first the following notation: $\mathbf{f}(k)$ is a $1 \times m$ vector that contains the k^{th} sample of each one of the m equalizer phases and \mathbf{F}_L a $1 \times Lm$ vector that contains the L consecutive

vectors $\mathbf{f}(k)$, $k = 0, \dots, L-1$:

$$\begin{aligned} \mathbf{f}(k) &= [f_1(k) \cdots f_m(k)] \\ \mathbf{F}_L &= [\mathbf{f}(0) \cdots \mathbf{f}(L-1)] . \end{aligned} \quad (7.38)$$

We also introduce the following multichannel z -transforms of the channel and the equalizer:

$$\begin{aligned} \mathbf{H}(z) &= \sum_{k=0}^{N-1} \mathbf{h}(k) z^{-k} \\ \mathbf{F}(z) &= \sum_{k=0}^{N-1} \mathbf{f}(k) z^{-k} . \end{aligned} \quad (7.39)$$

Then the z -transform $\hat{A}(z)$ of the equalizer output $\{\hat{a}_k\}$ is given by

$$\hat{A}(z) = \mathbf{F}(z) \mathbf{H}(z) A(z) . \quad (7.40)$$

In order to achieve zero-forcing equalization in the absence of noise, we should have $\hat{A}(z) = A(z)$, or if we allow for a constant discrete delay n , $\hat{A}(z) = A(z) z^{-n}$ which gives the following ZF condition for the equalizer parameters:

$$\mathbf{F}(z) \mathbf{H}(z) = z^{-n}, \quad n = 0, 1, \dots, N+L-2 . \quad (7.41)$$

Eq. (7.41) is the ZF condition in the z -domain. The counterpart of (7.41) in the time domain is

$$\mathbf{f} * \mathbf{h} = \delta(k-n), \quad (7.42)$$

where $*$ denotes convolution. By expressing this convolution as a matrix-vector product, (7.42) takes the form

$$[\mathbf{f}(0) \cdots \mathbf{f}(L-1)] \begin{bmatrix} \mathbf{h}(0) & \cdots & \mathbf{h}(N-1) & \mathbf{0}_{m \times 1} & \cdots & \mathbf{0}_{m \times 1} \\ \mathbf{0}_{m \times 1} & \mathbf{h}(0) & \cdots & \mathbf{h}(N-1) & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \mathbf{0}_{m \times 1} \\ \mathbf{0}_{m \times 1} & \cdots & \mathbf{0}_{m \times 1} & \mathbf{h}(0) & \cdots & \mathbf{h}(N-1) \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{1} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}^T, \quad (7.43)$$

or equivalently,

$$\mathbf{F}_L \mathcal{T}_L(\mathbf{H}_N) = [\mathbf{0} \cdots \mathbf{0} \ \mathbf{1} \ \mathbf{0} \cdots \mathbf{0}] \quad (7.44)$$

where the $\mathbf{1}$ is in the $n+1$ st position and we define $\mathcal{T}_M(\mathbf{x})$ as a (block) Toeplitz matrix with M (block) rows and $[\mathbf{x} \ \mathbf{0}_{p \times (M-1)}]$ as first (block) row (p is the number of rows in \mathbf{x}).

(7.44) is a linear system of $L+N-1$ equations in the Lm unknowns $\mathbf{f}(0), \dots, \mathbf{f}(L-1)$. For the existence of a solution the system needs to be exactly or underdetermined, i.e. the number

of equations needs to be equal to or greater than the number of unknowns in (7.44). This imposes the following constraint to the length L of each equalizer phase:

$$L \geq \underline{L} = \left\lceil \frac{N-1}{m-1} \right\rceil . \quad (7.45)$$

The matrix $\mathcal{T}_L(\mathbf{H}_N)$ is a generalized Sylvester matrix (see Appendix 7.A) It can be shown that for $L \geq \underline{L}$ it has full column rank if $\mathbf{H}(z) \neq 0, \forall z$ or in other words if the $H_j(z)$ have no zeros in common (note that in this case the multichannel is MP and therefore it can be identified from SOS according to theorem 7.2)). The same condition was given (in a different form) by Tong *et al.* in [TXK94]

To find a ZF equalizer (corresponding to some delay n), it suffices to take an equalizer length equal to \underline{L} . Then there exists at least one solution that satisfies exactly the system of equations (7.44). Every such solution therefore corresponds to a FIR ZF equalizer F_L that completely eliminates ISI! Therefore we have the following result:

Theorem 7.3: Under the FIR assumption, a ZF FIR equalizer can be found from (7.44) provided that the equalizer length L satisfies (7.45) and that the channel phases $H_j(z), j = 1, \dots, m$ have no zeros in common (in which case the multichannel $\mathbf{H}(z)$ will be MP in the vector sense).

The fact that there exist equalizers with a finite number of parameters that are ZF (completely eliminate ISI) seems to be an astonishing result because in the classical case of a SISO discrete channel, it is well known that even in the absence of noise, a ZF equalizer needs to be infinite-length, and is therefore not implementable in practice. This is due to the fact that for an FIR SISO channel with transfer function $H(z)$, the condition for ZF equalization in the z -domain is

$$H(z) F(z) = z^{-n} , \quad (7.46)$$

as compared to (7.46), which is the corresponding condition for the fractionally spaced channel and equalizer setup. Two things are worth noticing in (7.46): the first is that the ZF equalizer transfer function is the inverse of a polynomial, and therefore has an infinite number of terms. Each one of these terms corresponds to a tap for the ZF equalizer. This is why in this case there exists no ZF equalizer of finite length! The second is that the inversion of $H(z)$ in (7.46) is of course impossible if $H(z)$ has zeros on the unit circle (this corresponds to a singularity for the channel, or in other words to spectral nulls in the channel frequency response). In this case ZF equalization is impossible, even if the equalizer length is infinite.

On the other hand, the “inversion problem” appearing in (7.46) is somewhat different: by decomposing the channel and equalizer in their polyphase components, eq. (7.46) can be written as

$$\sum_{i=1}^m H_i(z) F_i(z) = z^{-n} . \quad (7.47)$$

This expression shows why the ZF condition can be exactly satisfied in the multichannel case with finite length equalizers. There is no need any more to invert a polynomial: in (7.47) the different phases of channel and equalizer are coupled in such a way so that the overall sum can be a constant each phase being a polynomial with a finite number of terms. (7.47) has more degrees of freedom than (7.46) in the sense that there are more than one polynomials to determine, but on the other hand it requires less (7.46) in the sense that it does not require the inversion of a polynomial. So the redundancy that we introduced by oversampling the received signal has resulted in the forming of a multi-equalization problem which allows for perfect equalization with a finite number of parameters.

Eq. (7.47) is known as the “Bezout” equation, and it is known that the condition for it to be satisfied is that the different channel phases $\{H_i\}, i = 1, \dots, m$ have no zeros in common⁵. Therefore we find again the same identifiability condition that was obtained in the time-domain. This condition now takes the place of the “no zeros on the unit circle” condition that has to be satisfied by the channel in the SISO case. This shows yet another astonishing result for the oversampled case: even if one of the channel phases is such that it has zeros on the unit circle, if these do not coincide with those of any other channel phase, ZF FIR equalization is still possible!

7.4.3 An equivalent SISO setup using upsampling and downsampling

In this section we will be interested in an alternative representation of the SIMO setup of the fractionally spaced channel and equalizer. Our point of departure is the following: does there exist a single-input single-output setup equivalent to the SIMO setup described in the previous sections such that the transmission line consists of one only phase instead of being split in m channel-equalizer phases? And in this case, how does the transfer function of this setup compare with the m phases of the SIMO setup?

The answer to this question resides in the area of *multirate filter banks*. We therefore present at this point some key facts of multirate filter banks that will be applicable to our SIMO setup. We first define the functions of upsampling and downsampling as follows:

The upsampled by a factor L version of a discrete signal $x(kT)$ is defined as

$$x^u(k T_L^u) = x^u(k \frac{T}{L}) = \begin{cases} x(\frac{k}{L}T) & \text{if } k \bmod L = 0 \\ 0 & \text{else} \end{cases} . \quad (7.48)$$

The downsampled by a factor M version of a discrete signal $x(kT)$ is defined as

$$x^d(k T_M^d) = x^d(k (TM)) = x(kM T) . \quad (7.49)$$

⁵The Bezout equation can be seen as a counterpart for polynomials of the following equation analyzed by the Greek mathematician Diophantos for real numbers: the equation $\sum_{i=1}^m \alpha_i x_i = \gamma$ ($\gamma \neq 0$) has a solution even if some of the α_i are 0, provided that they are not all simultaneously 0. On the contrary, the equation $\alpha x = \gamma$ ($\gamma \neq 0$) is of course impossible if $\alpha = 0$!

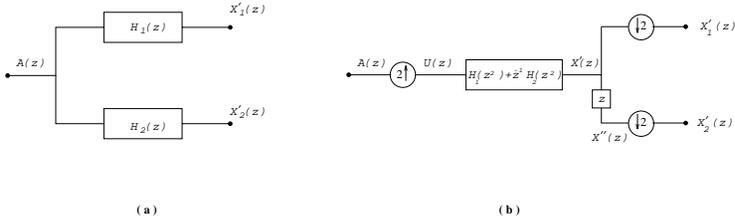


Figure 7.6: The transmitting part of the SIMO setup and an equivalent representation

The corresponding relations in the z -domain are

$$X^u(z) = \mathcal{Z}(x^u(kT_L^u)) = X(z^L), \quad (7.50)$$

and

$$X^d(z) = \mathcal{Z}(x^d(kT_M^d)) = \frac{1}{M} \sum_{i=0}^{M-1} X(w^i z^{\frac{1}{M}}), \quad (7.51)$$

respectively, where

$$w = e^{-j\frac{2\pi}{M}}.$$

In the sequel, we assume for simplicity that $m = 2$ (however the results that will be presented are valid for any integer $m > 2$). Consider first the figure 7.6. Figure 7.6(a) shows the transmitting part of the SIMO system, namely the splitting of the input into the two channel phases. Figure 7.6(b) shows a similar SIMO structure which uses upsampling and downsampling elements and in which instead of the two channel phases, one only channel (having the z -transform of the channel at the sampling rate $2/T$) is used.⁶ We will now prove the following lemma:

Lemma 7.6: The setup of figure 7.6(a) is equivalent to the setup of figure 7.6(b).

Proof:

For the setup of figure 7.6(b) we have (by making use of (7.50) and (7.51)) the following i/o relationships

⁶The circled arrows in the figures of this section denote upsampling or downsampling devices, depending on whether the arrow points upwards or downwards, respectively. The up(down)-sampling factor is indicated next to the arrow.

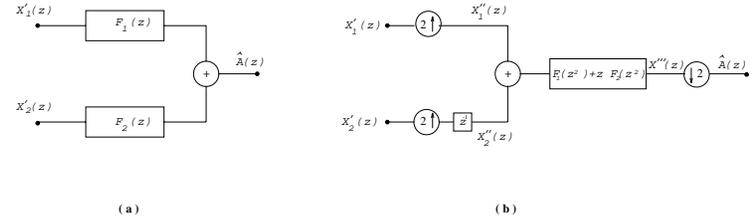


Figure 7.7: The receiving part of the SIMO setup and an equivalent representation

$$U(z) = A(z^2)$$

$$X'(z) = (H_1(z^2) + z^{-1}H_2(z^2)) U(z)$$

$$X_1'(z) = \frac{1}{2} (X'(z^{1/2}) + X'(-z^{1/2}))$$

$$X_2'(z) = \frac{1}{2} (X''(z^{1/2}) + X''(-z^{1/2})) = \frac{1}{2} (z^{1/2}X'(z^{1/2}) - z^{1/2}X'(-z^{1/2})),$$

which gives for the two outputs $X_1'(z)$ and $X_2'(z)$:

$$X_1'(z) = \frac{1}{2} (A(z)(H_1(z) + z^{-1/2}H_2(z)) + A(z)(H_1(z) - z^{-1/2}H_2(z)))$$

$$X_2'(z) = \frac{1}{2} z^{1/2} (A(z)(H_1(z) + z^{-1/2}H_2(z)) - A(z)(H_1(z) - z^{-1/2}H_2(z))),$$

and therefore

$$X_1'(z) = A(z) H_1(z) \quad (7.52)$$

$$X_2'(z) = A(z) H_2(z).$$

Now the i/o relations of (7.52) are exactly the same for figure 7.6(a). Therefore the two setups are equivalent. This concludes the proof. \square

An equivalent representation can also be found also for the receiving part that contains the equalizer phases as depicted in figure 7.7(a). In a similar way as for the transmitting part, the following lemma holds:

Theorem 7.7: The setup of figure 7.7(a) is equivalent to the setup of figure 7.7(b).

Proof:

For the setup of figure 7.7(b) we have the following i/o relationships

$$X_1''(z) = X_1'(z^2)$$

$$X_2''(z) = z^{-1}X_2''(z^2)$$

$$X'''(z) = (F_1(z^2) + zF_2(z^2)) (X_1''(z) + X_2''(z))$$

$$\hat{A}(z) = \frac{1}{2} (X'''(z^{1/2}) + X'''(-z^{1/2}))$$

which gives for $X'''(z)$:

$$X'''(z) = X'_1(z^2)F_1(z^2) + X'_2(z^2)F_2(z^2) + z^{-1}X'_2(z^2)F_1(z^2) + zX'_1(z^2)F_2(z^2) , \quad (7.53)$$

and therefore for the output $\hat{A}(z)$:

$$\hat{A}(z) = X'_1(z)F_1(z) + X'_2(z)F_2(z) . \quad (7.54)$$

The i/o relationship (7.54) is the same as the one corresponding to figure 7.7(a). Therefore the two setups are equivalent. This concludes the proof. \square

As the setups in figures 7.6(a) and 7.7(a) correspond to the transmission and reception part of the SIMO setup depicted in figure 7.5, it turns out that the series of the setups in figures 7.6(b) and 7.7(b) is also equivalent to the setup of figure 7.5. It can be easily seen that the rightmost part of the setup of figure 7.6(b) (the one that contains the downsampling and shift elements), connected in series to the corresponding leftmost part of the setup of figure 7.7(b), is simply equivalent to a channel with an ideal Dirac transfer function. Therefore it can be eliminated from the series of the two setups. Figure 7.8 depicts the corresponding setting (in our case $m = 2$). We therefore have the following theorem:

Theorem 7.4: The setup of figure 7.5 (for $m = 2$) is equivalent (in the absence of noise) to the SISO setup of figure 7.8.

It is straightforward to verify that the same result holds for any integer oversampling factor $m > 2$, leading to the following theorem:

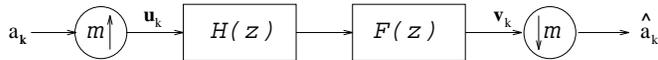


Figure 7.8: Fractionally-spaced channel and equalizer.

Theorem 7.5: The fractionally-spaced channel and equalizer corresponding to an integer oversampling factor of $OF = m$ can be represented as in figure 7.8, where $H(z)$ and $F(z)$ are defined as

$$\begin{aligned} H(z) &= \sum_{i=1}^n z^{-(i-1)} H_i(z^m) \\ F(z) &= \sum_{i=1}^n z^{i-1} F_i(z^m) \end{aligned} \quad (7.55)$$

The SISO setup of figure 7.8 is an appealing alternative to the polyphase representation presented in the previous sections. It provides a scheme closer to the classical SISO symbol-rate setup in that it employs a single transmission line. The oversampling aspect is now present in the upsampling and downsampling devices, as well as in the construction of the single phase channel and equalizer. In the degenerate case of no oversampling ($m = 1$), the symbol-rate setup is immediately provided. Moreover, as will be shown in the sequel, this setup allows for an interesting frequency-domain analysis. Before that, we examine first the zero forcing equalization aspect in the light of the new setup.

Zero-forcing equalization

We require ZF equalization at the output of the setup of figure 7.8. This gives

$$\hat{A}(z) = A(z) z^{-n} , \quad (7.56)$$

if we allow also for a delay n as in (7.41). Using again the identities (7.50) and (7.51) we get for the case $m = 2$

$$\begin{aligned} \hat{A}(z) &= \frac{1}{2}(V(z^{1/2}) + V(-z^{1/2})) \\ V(z) &= (F_1(z^2) + zF_2(z^2))(H_1(z^2) + z^{-1}H_2(z^2)) U(z) \\ U(z) &= A(z^2) \end{aligned}$$

which gives combined with (7.51)

$$\frac{1}{2}\{[(F_1(z) + z^{1/2}F_2(z))(H_1(z) + z^{-1/2}H_2(z))] + [(F_1(z) - z^{1/2}F_2(z))(H_1(z) - z^{-1/2}H_2(z))]\} = z^{-n} ,$$

resulting to

$$H_1(z)F_1(z) + H_2(z)F_2(z) = z^{-n} . \quad (7.57)$$

Eq. (7.57) is, as expected, identical to the one derived for the setup of figure 7.5, since the two setups have been shown to be identical. The above demonstration just shows an alternative way to derive the Bezout equation (7.47) using multirate filter banks instead of a polyphase structure. It can be shown that the Bezout equation is also derived from the setup of figure 7.8 for any integer oversampling factor $m > 2$. In the sequel we assume for simplicity a zero delay ($n = 0$).

An interesting interpretation of the ZF condition in the light of the setup of figure 7.8 is the following. We denote by $G(z)$ the overall filter corresponding to the cascade of oversampled channel and equalizer:

$$G(z) = F(z)H(z) . \quad (7.58)$$

The equalizer output can be written in the z -domain as ($m = 2$)

$$\hat{A}(z) = \frac{1}{2}[V(z^{1/2}) + V(-z^{1/2})] = [G(z^{1/2})A(z) + G(-z^{1/2})A(z)] ,$$

which gives the following i/o relationship in terms of $G(z)$.

$$\hat{A}(z) = \frac{G(z^{1/2}) + G(-z^{1/2})}{2} A(z) . \quad (7.59)$$

In the case of $m > 2$ the above relation generalizes to

$$\hat{A}(z) = \frac{1}{m} \sum_{i=1}^m G(w^{i-1} z^{1/m}) A(z) . \quad (7.60)$$

If one now focuses on (7.60) it is clear that the transfer function $\sum_{i=1}^m G(w^{i-1} z^{1/m})$ represents just a downsampled by a factor m version of $G(z)$ (compare to (7.51)). Now let's consider the m phases of $g(\cdot)$ in the time domain:

$$g_i(k) = g(T(mk + i - 1)) , \quad i = 1, \dots, m . \quad (7.61)$$

The ZF requirement then takes the following form in the time domain

$$g_i(k) = \delta(k) , i \text{ is one of } \{1, \dots, m\} . \quad (7.62)$$

Eq. (7.62) provides interesting insight to the ZF equalization aspect in the multichannel setup: in order to be ZF, *one only* among the m different phases of the channel-equalizer cascade as defined in (7.58) needs to be a Dirac function! This is a different point of view that explains why ZF equalization is possible with finite-length equalizers in the case of a multichannel setup: if all the phases of $G(z)$ were required to be Dirac functions, this would not be achievable with finite-length equalizers; however, the fact that only one of the phases needs to be a Dirac, and the rest can be whatever, can be achieved due to the many degrees of freedom present in the m phases. At this point we present an example that demonstrates the above mentioned characteristics.

Example:

We consider the case of two channel phases with transfer functions given as

$$\begin{aligned} H_1(z) &= 1 + z^{-1} \\ H_2(z) &= 1 + \frac{1}{2}z^{-1} , \end{aligned}$$

(notice that the first channel has a zero on the unit circle). Figure 7.9 refers to the equivalence of the setups of figures 7.6(a) and 7.6(b): on top one can see the samples of the upsampled input binary signal and in the middle the result of the convolution of this signal with the SISO channel $H(z) = H_1(z^2) + z^{-1}H_2(z^2)$. The bottom graph shows the sequence obtained by embedding the two outputs of the SIMO structure of figure 7.6(a). As expected, this coincides with the output of $H(z)$, thus verifying the equivalence between the two representations. Now we are interested in the ZF aspect: we choose an equalizer length of $L = 2$ for each equalizer phase (this exceeds the minimum required value given by (7.45), $\underline{L} = 1$). By solving the system of equations (7.43), we find easily that a pair of ZF equalizers is

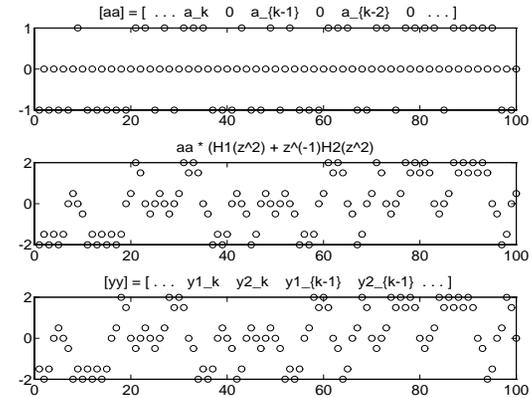


Figure 7.9: A verification of the equivalence of the setups of figures 7.6(a) and 7.6(b)

$$\begin{aligned} F_1(z) &= \frac{1}{2} + \frac{3}{4}z^{-1} \\ F_2(z) &= \frac{1}{2} - \frac{3}{2}z^{-1} . \end{aligned}$$

This pair of equalizers will completely eliminate ISI in the absence of noise. Now let's calculate the cascade of $H(z)$ and $F(z)$ as given by (7.55). It turns out that

$$G(z) = H(z)F(z) = \frac{1}{2}z + 1z^0 - \frac{1}{2}z^{-1} + 0z^{-2} - \frac{1}{2}z^{-3} + 0z^{-4} + \frac{3}{8}z^{-5} .$$

Figure 7.10 shows the corresponding impulse response $\{g(k)\}$. Circles correspond to samples of the even phase of g , and crosses to samples of the odd phase. Note that only the even phase corresponds to a Dirac function. This shows that forcing one only phase to be a Dirac function suffices for ZF equalization as shown by eq. (7.62).

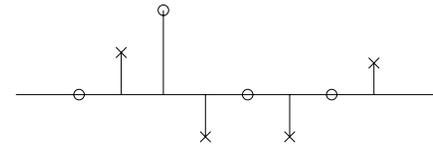


Figure 7.10: The overall system $G(z)$ corresponding to ZF equalization

In the next section we will examine the setup of figure 7.8 in the frequency domain.

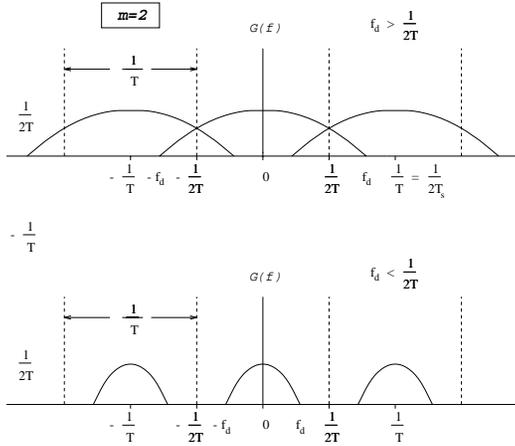


Figure 7.11: Nyquist condition for the oversampled channel

Frequency domain analysis

According to eq. (7.60), the transfer function of the channel-equalizer cascade of the setup of figure 7.8 is given in the z -domain as

$$G(z) = \frac{1}{m} \sum_{i=1}^m G(w^{i-1} z^{1/m}) . \quad (7.63)$$

So the ZF condition in the z -domain is

$$\frac{1}{m} \sum_{i=1}^m G(w^{i-1} z^{1/m}) = 1 . \quad (7.64)$$

By substituting $e^{j2\pi f}$ for z we find the corresponding expression in the frequency domain:

$$\frac{1}{m} \sum_{i=1}^m G(e^{j2\pi(\frac{f+k}{m})}) = 1 ,$$

which can be written as follows if now we denote by $G(f)$ the frequency response instead of the z -transform of $g(\cdot)$

$$\frac{1}{m} \sum_{i=1}^m G\left(\frac{f+k}{m}\right) = 1 \quad -\frac{1}{2} < f < \frac{1}{2} . \quad (7.65)$$

$G(f)$ is a periodic function with period $1/T$ where T is the symbol period. Now the following interpretation can be drawn from eq. (7.65): in order to have (7.65) satisfied, there needs to be some aliasing between adjacent frequency characteristics $G(\frac{f+k}{m})$ (otherwise, if there are

some frequency regions with no aliasing, it will be impossible to have a non-zero sum within these regions). Let's suppose now that $G(f)$ is band-limited with a bandwidth B

$$B = f_d . \quad (7.66)$$

Since the distance between adjacent frequency-pulses in (7.65) is $1/T$ and each pulse occupies a frequency range of width $2f_d$, it turns out that the condition for aliasing is

$$f_d > \frac{1}{2T} . \quad (7.67)$$

A graphical representation of the condition (7.67) can be found in figure 7.11, in which we assume $m = 2$. One can see the two different situations that arrive in this case when (7.67) is satisfied or not, respectively. Therefore we have established the following necessary condition for ZF equalization in the frequency domain:

Theorem 7.6: A necessary condition in order to achieve zero ISI in the multichannel setup is that the bandwidth of the overall system $G(f)$ defined in (7.63) satisfy (7.67).

Since now $G(f) = H(f)F(f)$, in order to have $G(f)$ satisfy (7.67) it is necessary that $H(f)$ satisfies it too. Therefore we have the following theorem

Lemma 7.7: Let f_d^h denote the bandwidth of the channel transfer function $H(f)$ that corresponds to $H(z)$ as defined in (7.55). Then a necessary condition in order to achieve zero ISI in the multichannel setup is that

$$f_d^h > \frac{1}{2T} . \quad (7.68)$$

where T is the symbol rate.

This theorem is actually the counterpart of the Nyquist condition presented in chapter 2, for fractionally-spaced equalization. This shows another advantage of using the representation of figure 7.8: it has allowed to obtain the Nyquist condition for the multichannel setup as a simple extension of the classical case: the condition has exactly the well-known classical form, the only difference being that the bandwidth f_d refers now to the channel $H(z)$ as defined in (7.55).

Moreover, (7.67) is a counterpart of the identifiability condition mentioned before in the frequency domain. Namely, when the channel bandwidth f_d^h does not satisfy (7.68), this means that the m channel phases will have zeros in common into the frequency regions that correspond to non-overlapping. But this is exactly the necessary condition that was derived in our time-domain analysis. So our frequency-domain analysis of the setup of figure 7.8 has allowed for the acquisition of the same identifiability condition from a different point of view.

Finally, theorem 7.6 gives us some insight on whether band limitations influence or not the channel estimation problem. If the channel is bandlimited with bandwidth $f_d^h \in (\frac{1}{2T}, \frac{m}{2T})$,

this poses no particular problem for the determination of a ZF equalizer (assuming infinite length). If $f_d^h < \frac{1}{2T}$ however, then the $H_j(f)$, $j = 1, \dots, m$ are zero simultaneously for some f rendering ZF equalization impossible. This is the infinite length equivalent of the condition of no zeros in common in the FIR case.

7.4.4 ZF equalization and noise enhancement

In this section we will remain in the frequency domain in order to analyze the problem of noise enhancement of ZF equalizers for the multichannel system. As mentioned in chapter 1, the noise enhancement produced by ZF equalizers is of great importance in the single channel case. We are therefore interested in studying the same phenomenon in the case of our multichannel setup.

We begin with the following remark: consider for simplicity the case $m = 2$ and suppose that the setting

$$\begin{cases} F_1^0 \\ F_2^0 \end{cases},$$

corresponds to a ZF equalizer and therefore satisfies

$$H_1 F_1^0 + H_2 F_2^0 = 1 .$$

Now consider another setting, namely

$$\begin{aligned} F_1' &= F_1^0 - G H_2 \\ F_2' &= F_2^0 - G H_1 \end{aligned} \quad (7.69)$$

where G is any stable filter of finite or infinite length⁷. It can be easily verified that

$$H_1 F_1' + H_2 F_2' = 1 ,$$

which means that *any* equalizer of the form (7.69) is also zero forcing! The variety of filters G that can be used represents a lot of degrees of freedom to determine different ZF equalizers for a given equalizer length. These will be all equivalent in the absence of noise, however one will be the optimal in the presence of noise in terms of noise enhancement. Now the optimal equalizer for a given length is only a special case of an equalizer of greater length, which can be still optimized (due to the degrees of freedom introduced by increasing the length) resulting to a better performing ZFE. We can sum up this discussion as follows:

- There is an optimal FIR equalizer in terms of noise enhancement for a given equalizer length.
- The length of a ZFE can be increased so as to reduce noise enhancement.

⁷In the equations of this section we assume all the filters to be extended with 0's so that they have compatible lengths, when needed.

This second remark shows that in practice (where of course noise will be present), even though ZF equalizers of finite length exist (as mentioned before), it will still be useful to use greater equalizer lengths in order to reduce noise enhancement. The difference with the single channel case resides in the fact that there increasing the equalizer length not only reduces the noise enhancement, but also improves the matching of the channel inverse, whereas in the multichannel case it only reduces the noise enhancement⁸. In the following subsection we derive the overall optimal ZF equalizer in terms of noise enhancement, namely the optimal infinite-length ZF equalizer.

Optimal infinite-length ZF equalizer

Considering the noise variance at the input of each equalizer F_i to be of variance σ_v^2 , the corresponding variance at the equalizer output will be given as

$$\sigma_{v,o}^2 = \sigma_v^2 \sum_{i=1}^m \sum_{j=1}^L |f_i(j)|^2 = \sigma_v^2 \sum_{i=1}^m |F_i(j)|^2 , \quad (7.70)$$

where we recall that $f_i(j)$ denotes the j^{th} coefficient of the i^{th} equalizer phase. In the case of infinite-length equalizer, using Pisarenko we get the following criterion that must be satisfied by the optimal infinite-length ZFE

$$\begin{cases} \min_{F_i(f)} \int_{-\frac{1}{2}}^{\frac{1}{2}} \sum_{i=1}^m |f_i(f)|^2 df \\ \text{subject to } \sum_{i=1}^m H_i(f) F_i(f) = 1 \end{cases} , \quad (7.71)$$

As the integrated quantity in (7.71) is positive, the criterion reduces to

$$\begin{cases} \min_{F_i(f)} \sum_{i=1}^m |f_i(f)|^2 \\ \text{subject to } \sum_{i=1}^m H_i(f) F_i(f) = 1 \end{cases} , \quad (7.72)$$

which can be written in matrix form as

$$\begin{cases} \min_{\mathbf{F}(f)} \|\mathbf{F}(f)\|^2 \\ \text{subject to } \langle \mathbf{H}^*(f), \mathbf{F}(f) \rangle = 1 \end{cases} , \quad (7.73)$$

where $\mathbf{F}(f)$, $\mathbf{H}(f)$ are defined as

$$\mathbf{F}(f) = \begin{bmatrix} F_1(f) \\ \vdots \\ F_m(f) \end{bmatrix} , \quad \mathbf{H}(f) = \begin{bmatrix} H_1(f) \\ \vdots \\ H_m(f) \end{bmatrix} . \quad (7.74)$$

As can be seen in figure 7.12, the solution \mathbf{F}^0 to the minimization problem (7.73) is

⁸An exception to this fact exists in the particular case when the channels have zeros in common: in this case increasing the equalizer length results in a better equalizer, even in the absence of noise.

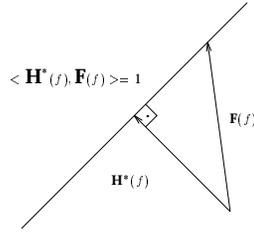


Figure 7.12: Finding the optimal ZFE by orthogonality

$$\mathbf{F}^o = \frac{1}{\sum_{i=1}^m |H_i(f)|^2} \mathbf{H}^*(f) \quad (7.75)$$

Eq. (7.75) gives the optimum infinite-length ZF equalizer in terms of noise enhancement in the frequency domain. According to (7.70), the corresponding minimal noise variance at the equalizer output will be given by

$$\sigma_{min}^2 = \sigma_v^2 \int_{-\frac{1}{2}}^{\frac{1}{2}} \frac{df}{\sum_{i=1}^m |H_i(f)|^2} . \quad (7.76)$$

This is therefore the lower bound of output noise variance that can be achieved by a ZF equalizer. In order to give some insight to the expression (7.76) we examine the case $m = 2$. Then (7.76) can be written as

$$\sigma_{min}^2 = \sigma_v^2 \int_{-\frac{1}{2}}^{\frac{1}{2}} \frac{df}{|H_1(f)|^2} - \sigma_v^2 \frac{\int_{-\frac{1}{2}}^{\frac{1}{2}} \frac{H_2^*(f)}{H_1^*(f)} df}{\int_{-\frac{1}{2}}^{\frac{1}{2}} \left(\sum_{i=1}^2 |H_i(f)|^2 \right) df} \quad (7.77)$$

The first term in the right-hand side of (7.77) represents the optimal output noise variance in the case of symbol-rate sampling ($m = 1$). On the other hand, the second term represents the modification due to the fact that we have 2 channels. In fact, as this second term is positive, it reduces the noise variance of the first term, which means that the degrees of freedom introduced by oversampling give the potential to achieve better ZFE performance. We therefore arrive at the following conclusion.

- Increasing the oversampling factor m (or the number of receiving antennas) can help improve the zero-forcing equalization performance.

Moreover, this fact doesn't seem to be influenced by bandwidth limitations of the channel, provided of course that (7.68) is satisfied in order to be able to derive ZF equalizers. This is

another point of view (independent of blind aspects) that proves the advantages related with the introduction of extra degrees of freedom by introducing diversity.

7.5 ZFE and ID by multichannel linear prediction

In the case of symbol-rate sampling, as we saw in section 7.3, the channel can be identified by spectral factorization if it is minimum-phase. The counterpart of spectral factorization in the time-domain is linear prediction: in the absence of noise, the input sequence equals the innovation process as described in eq. (2.61). In this section we shall see how a multichannel linear prediction problem can be formulated for the case of a MP SIMO system and how it allows for the acquisition of the transmitted symbols in the absence of noise.

7.5.1 Zero-forcing equalization by multichannel linear prediction

1-step ahead forward linear prediction

- Forward linear prediction and second-order statistics

We consider the noiseless case:

$$v(t) \equiv 0.$$

We are interested in identifying the channel coefficients of the SIMO setup based on linear prediction. Intuitively this should be possible, since linear prediction is based on second-order statistics of the received data and we have already seen (Theorem 7.2) that the identifiability of the SIMO channel can be achieved based on SOS.

The input-output relation of the SIMO channel can be written in the absence of noise as

$$\mathbf{X}_L(k) = \mathcal{T}_L(\mathbf{H}_N) A_{L+N-1}(k) , \quad (7.78)$$

where

$$\mathbf{X}_L(k) = [\mathbf{x}^H(k) \cdots \mathbf{x}^H(k-L+1)]^H . \quad (7.79)$$

Therefore, the covariance matrix $\mathbf{R}_L^{\mathbf{X}}$ of the received signal $\mathbf{x}(k)$ has the following structure

$$\mathbf{R}_L^{\mathbf{X}} = \mathbf{E} \mathbf{X}_L(k) \mathbf{X}_L^H(k) = \mathcal{T}_L(\mathbf{H}_N) \mathbf{R}_{L+N-1}^A \mathcal{T}_L^H(\mathbf{H}_N) , \quad (7.80)$$

where

$$\mathbf{R}_L^A = \mathbf{E} A_L(k) A_L^H(k) . \quad (7.81)$$

The covariance matrix $\mathbf{R}_L^{\mathbf{X}}$ is of dimension $Lm \times Lm$, and its rank is $L + N - 1$ (the number of columns in $\mathcal{T}_L(\mathbf{H}_N)$). Therefore, in the absence of noise we have

$$\mathbf{R}_L^{\mathbf{X}} = \begin{cases} \text{full-rank,} & Lm \leq L + N - 1 \\ \text{singular,} & Lm > L + N - 1 \end{cases} \quad (7.82)$$

When $\mathbf{R}_L^{\mathbf{X}}$ is singular, each further increase of L by 1 results in an increase of $\text{rank}(\mathbf{R}_L^{\mathbf{X}})$ by 1 and an increase of the dimension of its nullspace by $m-1$.

We consider the following linear prediction problem:

Predict $\mathbf{x}(k)$ as a linear combination of the components of $\mathbf{X}_L(k-1)$.

This problem corresponds to vector-forward linear prediction. The predicted signal is a linear combination of the L most recent received vector-samples:

$$\tilde{\mathbf{x}}(k) = \mathbf{p}_1 \mathbf{x}(k-1) + \cdots + \mathbf{p}_L \mathbf{x}(k-L), \quad (7.83)$$

where $\{\mathbf{p}_i\}$ are $m \times m$ matrices and represent the LP coefficients. The prediction error can then be written as

$$\begin{aligned} \tilde{\mathbf{x}}(k)|_{\mathbf{X}_L(k-1)} &= \mathbf{x}(k) - \tilde{\mathbf{x}}(k)|_{\mathbf{X}_L(k-1)} \\ &= [I_m \quad -\mathbf{P}_L] \mathbf{X}_{L+1}(k), \end{aligned} \quad (7.84)$$

where

$$\mathbf{P}_L = [\mathbf{p}_1 \cdots \mathbf{p}_L].$$

The prediction error variance is by definition

$$\begin{aligned} \sigma_{\tilde{\mathbf{x}}_L}^2 &= E \tilde{\mathbf{x}}(k) \tilde{\mathbf{x}}^H(k) = \\ [I_m \quad -\mathbf{P}_L] \mathbf{R}_{L+1}^{\mathbf{X}} [I_m \quad -\mathbf{P}_L]^H. \end{aligned} \quad (7.85)$$

The minimization of the prediction error variance leads therefore to the following optimization problem

$$\min_{\mathbf{P}_L} [I_m \quad -\mathbf{P}_L] \mathbf{R}_{L+1}^{\mathbf{X}} [I_m \quad -\mathbf{P}_L]^H = \sigma_{\tilde{\mathbf{x}}_L}^2. \quad (7.86)$$

This gives, in a way similar to the case of scalar forward linear prediction:

$$[I_m \quad -\mathbf{P}_L] \mathbf{R}_{L+1}^{\mathbf{X}} = \begin{bmatrix} \sigma_{\tilde{\mathbf{x}}_L}^2 & \mathbf{0} \cdots \mathbf{0} \end{bmatrix}. \quad (7.87)$$

By partitioning the covariance matrix $\mathbf{R}_{L+1}^{\mathbf{X}}$, the above equation can be written as

$$\begin{bmatrix} I_m & | & -\mathbf{P}_L \end{bmatrix} \begin{bmatrix} \mathbf{r}_0 & | & \mathbf{r} \\ \hline \mathbf{r}^H & | & \mathbf{R}_L^{\mathbf{X}} \end{bmatrix} = \begin{bmatrix} \sigma_{\tilde{\mathbf{x}}_L}^2 & | & \mathbf{0} \cdots \mathbf{0} \end{bmatrix}. \quad (7.88)$$

This gives

$$\begin{cases} \sigma_{\tilde{\mathbf{x}}_L}^2 = \mathbf{r}_0 - \mathbf{r}(\mathbf{R}_L^{\mathbf{X}})^{-1} \mathbf{r}^H \\ \mathbf{P}_L = \mathbf{r}(\mathbf{R}_L^{\mathbf{X}})^{-1}. \end{cases} \quad (7.89)$$

Equation (7.89) shows how both the prediction error variance and the prediction coefficients can be computed from the SOS of the cyclostationary received signal. The importance of this equation lies in the fact that those two quantities may allow the identification of the multichannel, as will be shown in the sequel.

- Forward linear prediction and multichannel identification/equalization

When $L_m > L+N-1$, $\mathcal{T}_L(\mathbf{H}_N)$ has full column rank. Therefore, due to (7.78) (applied at time instant $k-1$), we have

$$\text{span}(\mathbf{X}_L(k-1)) = \text{span}(A_{L+N-1}(k-1)). \quad (7.90)$$

Due to (7.90), the prediction error can be expressed as

$$\tilde{\mathbf{x}}(k)|_{\mathbf{X}_L(k-1)} = \tilde{\mathbf{x}}(k)|_{A_{L+N-1}(k-1)}. \quad (7.91)$$

By definition, the prediction error $\tilde{\mathbf{x}}(k)$ is orthogonal to the set of samples $\mathbf{X}_L(k-1)$:

$$\tilde{\mathbf{x}}(k) \perp \mathbf{X}_L(k-1). \quad (7.92)$$

Using (7.90) and (7.84), (7.92) reads

$$[I_m \quad -\mathbf{P}_L] \mathcal{T}_{L+1}(\mathbf{H}_N) A_{L+N}(k) \perp A_{L+N-1}(k-1), \quad (7.93)$$

or

$$[I_m \quad -\mathbf{P}_L] \mathcal{T}_{L+1}(\mathbf{H}_N) E(A_{L+N}(k) A_{L+N-1}^H(k-1)) = \mathbf{0}_{m \times (L+N-1)}. \quad (7.94)$$

Now we have that

$$\begin{aligned} E(A_{L+N}(k) A_{L+N-1}^H(k-1)) &= \begin{bmatrix} E a_k a_{k-1}^* & \cdots & E a_k a_{k-(L+N-1)}^* \\ \hline & R_{L+N-1}^a & \\ \hline \end{bmatrix} = \\ &= \begin{bmatrix} * & | & E a_k a_{k-1}^* & \cdots & E a_k a_{k-(L+N-1)}^* \\ * & | & & & \\ \vdots & | & & & \\ * & | & & & \end{bmatrix} \times \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} \\ \hline I_{L+N-1} \end{bmatrix}, \end{aligned} \quad (7.95)$$

where $*$ may be any scalar. If instead of arbitrary values for this column we use the following elements:

$$[* \ * \ \cdots \ *] = [E|a_k|^2 \ E a_k^* a_{k-1} \ \cdots \ E a_k^* a_{k-(L+N-1)}],$$

then (7.95) takes the form

$$E(A_{L+N}(k) A_{L+N-1}^H(k-1)) = R_{L+N}^a \begin{bmatrix} \mathbf{0} \cdots \mathbf{0} \\ \hline I_{L+N-1} \end{bmatrix}, \quad (7.96)$$

which gives when combined with (7.94)

$$[I_m \quad -\mathbf{P}_L] \mathcal{T}_{L+1}(\mathbf{H}_N) \mathbf{R}_{L+N}^a \begin{bmatrix} \mathbf{0} \cdots \mathbf{0} \\ \hline I_{L+N-1} \end{bmatrix} = \mathbf{0}_{m \times (L+N-1)}. \quad (7.97)$$

Now let us consider the prediction problem for the transmitted symbols. We get similarly

$$\hat{a}(k)_{A_M(k-1)} = \mathbf{Q}_M A_M(k-1), \quad (7.98)$$

$$[\mathbf{1} - \mathbf{Q}_M] \mathbf{R}_{M+1}^a = \begin{bmatrix} \sigma_{a,M}^2 & \mathbf{0} \cdots \mathbf{0} \end{bmatrix}, \quad (7.99)$$

where now the elements of \mathbf{Q}_M are scalars. For $M = L + N - 1$, equation (7.99) becomes

$$[\mathbf{1} - \mathbf{Q}_{L+N-1}] \mathbf{R}_{L+N}^a = \begin{bmatrix} \sigma_{a,L+N-1}^2 & \mathbf{0} \cdots \mathbf{0} \end{bmatrix} = \sigma_{a,L+N-1}^2 [\mathbf{1} \ \mathbf{0} \cdots \mathbf{0}]. \quad (7.100)$$

Now note that (7.97) can be written in the form

$$[I_m - \mathbf{P}_L] \mathcal{T}_{L+1}(\mathbf{H}_N) \mathbf{R}_{L+N}^a = \begin{bmatrix} * \\ \vdots \\ \mathbf{0}_{m \times (L+N-1)} \\ * \end{bmatrix} = \begin{bmatrix} * \\ \vdots \\ * \end{bmatrix} [\mathbf{1} \ \mathbf{0} \cdots \mathbf{0}], \quad (7.101)$$

where * can be again any scalar. Combining (7.101) with (7.100), we get

$$[I_m - \mathbf{P}_L] \mathcal{T}_{L+1}(\mathbf{H}_N) = \frac{\mathbf{1}}{\sigma_{a,L+N-1}^2} \begin{bmatrix} * \\ \vdots \\ * \end{bmatrix} [\mathbf{1} - \mathbf{Q}_{L+N-1}]. \quad (7.102)$$

From (7.97) we note also that

$$[I_m - \mathbf{P}_L] \mathcal{T}_{L+1}(\mathbf{H}_N) = \begin{bmatrix} \mathbf{h}(0) \\ * \end{bmatrix}. \quad (7.103)$$

Combining (7.102) and (7.103) we get

$$[I_m - \mathbf{P}_L] \mathcal{T}_{L+1}(\mathbf{H}_N) = \mathbf{h}(0) [\mathbf{1} - \mathbf{Q}_{L+N-1}]. \quad (7.104)$$

The minimum prediction error variance is therefore:

$$\begin{aligned} \sigma_{\mathbf{x},L}^2 &= \mathbf{h}(0) [\mathbf{1} - \mathbf{Q}_{L+N-1}] \mathbf{R}_{L+N}^a [\mathbf{1} - \mathbf{Q}_{L+N-1}]^H \mathbf{h}^H(0) \\ &= \mathbf{h}(0) \begin{bmatrix} \sigma_{a,L+N-1}^2 & \mathbf{0} \cdots \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{1} \\ -\mathbf{Q}_{L+N-1}^H \end{bmatrix} \mathbf{h}^H(0). \end{aligned}$$

We therefore have

$$\sigma_{\mathbf{x},L}^2 = \sigma_{a,L+N-1}^2 \mathbf{h}(0) \mathbf{h}^H(0). \quad (7.105)$$

Therefore, in the case $Lm > L + N - 1$ the prediction error variance $\sigma_{\mathbf{x},L}^2$ is rank-1. Moreover, (7.105) allows us to find $\mathbf{h}(0)$ up to a scalar multiple.

At this point we discriminate between two cases

- Uncorrelated input sequence

In this case,

$$R_{L+N}^a = \sigma_a^2 I_{L+N}. \quad (7.106)$$

Combining (7.106) with (7.100) we find

$$\mathbf{Q}_{L+N-1} = \mathbf{0}_{1 \times (L+N-1)}. \quad (7.107)$$

In this case instead of (7.104) we will have

$$[I_m - \mathbf{P}_L] \mathcal{T}_{L+1}(\mathbf{H}_N) = \mathbf{h}(0) [\mathbf{1} \ \mathbf{0}_{L+N-1}],$$

and therefore

$$\left(\frac{\mathbf{1}}{\mathbf{h}^H(0)\mathbf{h}(0)} \mathbf{h}^H(0) [I_m - \mathbf{P}_L] \right) \mathcal{T}_{L+1}(\mathbf{H}_N) = [\mathbf{1} \ \mathbf{0}_{L+N-1}]. \quad (7.108)$$

We therefore have the following result:

- When the transmitted data are uncorrelated, the channel is noiseless and $L > \underline{L}$, then a ZF equalizer can be found by linear prediction and equals

$$F_{ZF} = \frac{\mathbf{1}}{\mathbf{h}^H(0)\mathbf{h}(0)} \mathbf{h}^H(0) [I_m - \mathbf{P}_L].$$

Using (7.44), we could also determine the channel \mathbf{H}_N up to a scalar multiple.

In this case, the prediction problem allows us also (in theory) to check whether the H_j have zeros in common. Indeed, the common factor colors the transmitted symbols (MA process) and hence once $\sigma_{\mathbf{x},L}^2$ becomes of rank 1, its one nonzero eigenvalue $\sigma_{a,L+N-1}^2 \mathbf{h}^H(0)\mathbf{h}(0)$ continues to decrease as a function of L since for a MA process, $\sigma_{a,L}^2$ is a decreasing function of L .

- Correlated input sequence

If the transmitted symbols are correlated, we proceed as follows (Pisarenko-style [Sch91, page 500]). Linear prediction corresponds to the LDU factorization $\mathbf{L}\mathbf{R}\mathbf{L}^H = \mathbf{D}$. The prediction filters are rows of \mathbf{L} while the prediction variances are the diagonal elements of \mathbf{D} . Let's take l prediction filters corresponding to singularities in \mathbf{D} and assume the longest one has block length L . So we obtain \mathbf{F}_L^b of size $l \times Lm$. We introduce a block-componentwise transposition operator t , viz.

$$\begin{aligned} \mathbf{H}_N^t &= [\mathbf{h}(0) \cdots \mathbf{h}(N-1)]^t = [\mathbf{h}^T(0) \cdots \mathbf{h}^T(N-1)] \\ \mathbf{F}_N^t &= [\mathbf{f}(0) \cdots \mathbf{f}(N-1)]^t = [\mathbf{f}^T(0) \cdots \mathbf{f}^T(N-1)], \end{aligned} \quad (7.109)$$

where t is the usual transposition operator. Due to the singularities, we have

$$\mathbf{F}_L^b \mathcal{T}_L(\mathbf{H}_N) = \mathbf{0} \iff \mathbf{H}_N^t \mathcal{T}_N(\mathbf{F}_L^t) = \mathbf{0}. \quad (7.110)$$

Since $\mathbf{F}_L^b \mathbf{X}_L(k) = \mathbf{0}$, we call \mathbf{F}_L^b a *blocking* equalizer. We find: if $l(L+N-1) \geq mN-1$, then

$$\dim(\text{Range}^\perp\{\mathcal{T}_N(\mathbf{F}_L^t)\}) = 1. \quad (7.111)$$

In that case, we can identify the channel \mathbf{H}_N^{iH} (up to scalar multiple) as the last right singular vector of $\mathcal{T}_N(\mathbf{F}_L^{iH})$. In particular, let \mathbf{h}^\perp be $m \times (m-1)$ of rank $m-1$ such that $\mathbf{h}^{\perp H} \mathbf{h}(0) = 0$, then with $L = \underline{L}+1$ and $l = m-1$, we can take

$$\mathbf{F}_{\underline{L}+1}^i = \mathbf{h}^{\perp H} [I_m \quad -\mathbf{P}_L] . \quad (7.112)$$

From (7.104), one can furthermore identify \mathbf{Q}_{L+N-1} and via (7.99), this leads to the identification of the (Toeplitz) symbol covariance matrix \mathbf{R}_{L+N}^i up to the multiplicative scalar σ_a^2 (which may be known).

n-step ahead prediction

In order to avoid the dependence on $\mathbf{h}(0)$, we propose the following more flexible LP scheme. We consider an $(n+1)$ -step ahead prediction of $\mathbf{x}(k)$ of order L :

$$\hat{\mathbf{x}}(k) \text{ is predicted from } \mathbf{x}(k-n-1), \mathbf{x}(k-n-2), \dots, \mathbf{x}(k-n-L) . \quad (7.113)$$

Then in the absence of noise the prediction error will be equal to:

$$\tilde{\mathbf{x}}(k) = \mathbf{x}(k) - \hat{\mathbf{x}}(k) = \sum_{i=0}^n \mathbf{h}(i) a_{k-i} = P(z) \mathbf{x}(k) . \quad (7.114)$$

Now if one performs a backward prediction on the *prediction error* in (7.114), the resulting prediction error will equal $\mathbf{h}(n) a_{k-n}$:

$$\mathbf{h}(n) a_{k-n} = Q(z) \tilde{\mathbf{x}}(k) , \quad (7.115)$$

($P(z)$ and $Q(z)$ are $m \times m$ matrices). From (7.114) and (7.113), we deduce that a ZFE can be determined as:

$$F_{ZF}^L = \frac{\mathbf{h}^H(n)}{\mathbf{h}^H(n) \mathbf{h}(n)} Q(z) P(z) , \quad (7.116)$$

and corresponds to an overall impulse response of the form:

$$[0_{1 \times n} \quad 1 \quad 0 \quad \dots \quad 0] . \quad (7.117)$$

In this way, the dependence on $\mathbf{h}(0)$ is eliminated, and moreover, the kind of ZF equalizer we want can be a priori determined.

7.5.2 Linear prediction in the frequency-domain

We consider the z -transform of the linear prediction filter:

$$P_L(z) = [I_m \quad -\mathbf{P}_L] \begin{bmatrix} I_m \\ I_m z^{-1} \\ \vdots \\ I_m z^{-L} \end{bmatrix} . \quad (7.118)$$

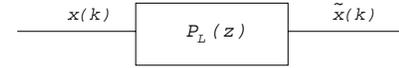


Figure 7.13: The prediction filter

The $m \times m$ matrix $P_L(z)$ is the prediction filter in the frequency (z) domain. As can be deduced from figure 7.13, we have the following i/o relationship for the prediction error spectral density:

$$P_L(z) S_{\mathbf{xx}}(z) P_L^H(z^{-*}) = S_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}}(z) . \quad (7.119)$$

When the prediction order L is infinite, then the prediction error $\tilde{\mathbf{x}}$ is white, and therefore its spectral density is flat and equal to the prediction error variance:

$$S_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}_\infty}(z) = \sigma_{\tilde{\mathbf{x}}_\infty}^2 , \quad (7.120)$$

and therefore in this case (7.119) will take the form

$$P_\infty(z) S_{\mathbf{xx}}(z) P_\infty^H(z^{-*}) = \sigma_{\tilde{\mathbf{x}}_\infty}^2 , \quad (7.121)$$

which gives when solved for $S_{\mathbf{xx}}(z)$:

$$S_{\mathbf{xx}}(z) = P_\infty^{-1}(z) \sigma_{\tilde{\mathbf{x}}_\infty}^2 P_\infty^{-H}(z^{-*}) \quad (7.122)$$

We now consider separately two cases.

- Noiseless case

In this case, we have for the PSD of $\mathbf{x}(k)$:

$$S_{\mathbf{xx}}(z) = \sigma_a^2 \mathbf{H}(z) \mathbf{H}^H(z^{-*}) , \quad (7.123)$$

where we have assumed the input data to be i.i.d. Since we have no noise, (7.105) holds, which gives when combined with (7.122)

$$S_{\mathbf{xx}}(z) = \sigma_a^2 P_\infty^{-1}(z) \mathbf{h}(0) \mathbf{h}^H(0) P_\infty^{-H}(z^{-*}) . \quad (7.124)$$

Now combining (7.123) and (7.124) we get

$$\mathbf{H}(z) = P_\infty^{-1}(z) \mathbf{h}(0) . \quad (7.125)$$

According to (7.125), in the noiseless case, the SIMO channel can be identified in the frequency domain from the inverse linear prediction filter of infinite order. This is an identifiability result that shows the role of the linear prediction filter in the frequency domain.

- Noisy case

We assume Gaussian i.i.d. additive noise, in which case $S_{\mathbf{xx}}(z)$ will be given as

$$S_{\mathbf{xx}}(z) = \sigma_a^2 \mathbf{H}(z) \mathbf{H}^H(z^{-*}) + \sigma_v^2 I_m . \quad (7.126)$$

Now the rank of $S_{\mathbf{xx}}(z)$ will be m (instead of 1 in the noiseless case). The PSD $S_{\mathbf{xx}}(z)$ can now be factored as

$$S_{\mathbf{xx}}(z) = S_{\mathbf{xx}}^+(z) S_{\mathbf{xx}}^{+H}(z^{-*}) . \quad (7.127)$$

$S_{\mathbf{xx}}^+(z)$ will be in general of size $r \times m$, where r is the rank of $S_{\mathbf{xx}}(z)$, in our case it will be therefore $m \times m$. We also factor the prediction error variance as

$$\sigma_{x,\infty}^2 = \tilde{\sigma}_{x,\infty} \sigma_{x,\infty}^H . \quad (7.128)$$

Combining (7.128) with (7.126) and (7.127) we find for the channel:

$$\mathbf{H}(z) = \sigma_a P_\infty^{-1}(z) \tilde{\sigma}_{x,\infty} . \quad (7.129)$$

This is the analog of (7.125) in the noisy case.

7.6 MMSE equalization

Minimum-mean-square-error equalizers (MMSEE's) are known to perform better in general than ZFE's in the presence of noise. This fact has already been discussed in chapter 2, where it was shown that when the channel has very deep spectral nulls, then the noise enhancement introduced by a ZF equalizer is very high (in the extreme case of channel zeros on the unit circle, the noise enhancement introduced by the ZFE is infinite). On the other hand, MMSEE's avoid this problem by trying to combat jointly both kinds of interference: ISI and noise. In this section we are interested in MMSE equalization in the context of the multichannel setup.

Before proceeding in the derivation and analysis of the MMSEE, we first derive the expression of the output SNR for the multichannel setup that will be used in the sequel as a measure of equalizer performance.

The SNR at the output of the multichannel setup

If we denote by $\{s(i)\}$ the oversampled impulse response of the channel-equalizer cascade then the equalizer output can be written as

$$\hat{a}(k) = s(\mathbf{1})a(k) + \sum_{i \neq \mathbf{1}} s(i) a(k-i) + b(k) , \quad (7.130)$$

where $b(k)$ represents the noise at the equalizer *output* and where we have assumed zero delay in transmission. The output noise $\{b(k)\}$ is the input noise filtered by the equalizer bank, it is

therefore given as

$$b(k) = \sum_{i=1}^m \sum_{j=0}^{L-1} v_i(k-j) f_i(j) , \quad (7.131)$$

where subscript i denotes, as usual, the equalizer or noise phase. Therefore the energy at the equalizer output is

$$E |\hat{a}(k)|^2 = E |s(\mathbf{1})a_k + \sum_{i \neq \mathbf{1}} s(i)a(k-i) + b(k)|^2 .$$

Assuming that all channels have i.i.d. additive noise of the same variance σ_v^2 , independent of the input sequence and that the input sequence is i.i.d. of variance σ_a^2 , the above expression becomes

$$E |\hat{a}(k)|^2 = \sigma_a^2 \left(|s(\mathbf{1})|^2 + \sum_{i \neq \mathbf{1}} |s(i)|^2 \right) + \sigma_v^2 \sum_{i=1}^m \sum_{j=0}^{L-1} |f_i(j)|^2 . \quad (7.132)$$

Therefore the SNR at the equalizer output will be equal to

$$SNR_o = \frac{\sigma_a^2 |s(\mathbf{1})|^2}{\sigma_a^2 \sum_{i \neq \mathbf{1}} |s(i)|^2 + \sigma_v^2 \sum_{i=1}^m \sum_{j=0}^{L-1} |f_i(j)|^2} . \quad (7.133)$$

7.6.1 Derivation of the MMSE equalizer

In order to obtain a formulation similar to the one used in the classical symbol-rate case, we define the regression vector $\mathbf{X}_L(k)$ and the equalizer vector $\mathbf{F}_L(k)$ as follows

$$\begin{aligned} \mathbf{X}_L^H(k) &= [\mathbf{x}^T(k) \cdots \mathbf{x}^T(k-L+1)] = [x_1(k) \cdots x_m(k) \cdots x_1(k-L+1) \cdots x_m(k-L+1)] \\ \mathbf{F}_L^T(k) &= [\mathbf{f}^T(\mathbf{0}) \cdots \mathbf{f}^T(L-1)] = [f_1(\mathbf{0}) \cdots f_m(\mathbf{0}) \cdots f_1(L-1) \cdots f_m(L-1)] . \end{aligned} \quad (7.134)$$

Then the equalizer output at time instant k can be written as

$$\hat{a}_k = \mathbf{X}_L^H(k) \mathbf{F}_L(k) . \quad (7.135)$$

The MMSE criterion will then have the typical form

$$\min_F E (|\hat{a}_k - a_{k-i}|^2) , \quad (7.136)$$

where F is the sought equalizer setting and i accounts for an introduced delay⁹. The solution to the criterion (7.136) is the MMSE equalizer corresponding to delay i and has the following form (taking into account (7.135))

$$F_{MMSE,i} = E \left(\mathbf{X}_L(k) \mathbf{X}_L^H(k) \right)^{-1} E (\mathbf{X}_L(k) a_{k-i}) = (R_L^{\mathbf{x}})^{-1} \mathbf{d}_i . \quad (7.137)$$

Notice that the form of the equalizer given in (7.137) is the same as the one of the classical MMSE equalizer, the only difference being the different composition of the regression and

⁹We will see in a next section that the choice of this delay may influence considerably the equalizer performance

equalizer vectors. All that is needed in (7.137) for the determination of the quantities R_L^x and \mathbf{d}_i is information about the channel and the second-order statistics of the input and the channel noise. In the presence of noise, the channel i/o relationship can be written as

$$\mathbf{X}_L(k) = \mathcal{T}_L(\mathbf{H}_N) A_{L+N-1} + V_L(k) , \quad (7.138)$$

and therefore the matrix R_L^x is equal to ¹⁰

$$R_L^x = \mathcal{T}_L(\mathbf{H}_N) R_{L+N-1}^o \mathcal{T}_L^H(\mathbf{H}_N) + R_L^v . \quad (7.139)$$

Let's see now how the channel information reflects in the vector \mathbf{d}_i . For example, when $i = 0$ (no delay) and the input sequence is white with variance σ_a^2 , we will have (using also the FIR assumption)

$$d_0^T = \sigma_a^2 [h_1^*(0) \cdots h_m^*(0) \mathbf{0} \cdots \mathbf{0}] \quad (7.140)$$

(the case of non-zero delays will be discussed in a subsequent section). According to (7.139) and (7.140), the MMSE equalizer can be easily determined if the channel has already been identified. Of course, the MMSE can be also calculated iteratively by using an adaptive algorithm like LMS or RLS if a training sequence is available.

7.6.2 MMSE equalizers and ZF equalizers

In this section we examine jointly MMSE and ZF equalizers. In the first part we compare them in terms of noise enhancement (by deriving in the frequency domain similar conditions as the ones existing for the continuous-time case). Then we show how an MMSE equalizer can be derived from a ZF equalizer. This will help obtain the MMSEE blindly.

Comparing MMSEE's and ZFE's in terms of noise enhancement

In the frequency domain, the MMSEE minimizes the following quantity (assuming infinite length equalizers, a white input sequence of variance σ_a^2 and white additive noise of variance σ_v^2 at each channel)

$$\sigma^2 = \min_{F_i(f)} \int_{-\frac{1}{2}}^{\frac{1}{2}} \left\{ \sigma_a^2 \left| \sum_{i=1}^m H_i(f) F_i(f) - 1 \right|^2 + \sigma_v^2 \left(\sum_{i=1}^m |F_i(f)|^2 \right) \right\} df , \quad (7.141)$$

¹⁰It is worth showing the structure of this matrix in the special case of a white input sequence of variance σ_a^2 :

$$R_L^x = \begin{bmatrix} \mathbf{H}_N \mathbf{H}_N^H & \mathbf{H}_N J_1^- \mathbf{H}_N^H & \cdots & \mathbf{H}_N J_{L-1}^- \mathbf{H}_N^H \\ \mathbf{H}_N J_1^+ \mathbf{H}_N^H & \mathbf{H}_N \mathbf{H}_N^H & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{H}_N J_{L-1}^+ \mathbf{H}_N^H & \cdots & \cdots & \mathbf{H}_N \mathbf{H}_N^H \end{bmatrix} + R_L^v ,$$

where J_1^+ and J_1^- are shift matrices that have as only non-zero elements their i^{th} upper or lower anti-diagonals, respectively, which are full of 1's. Note the block Toeplitz form of the above matrix and how it depends on the channel matrix \mathbf{H}_N .

which represents the SNR at the equalizer output. In the right hand side of the expression (7.141), the first term represents the ISI and the second term the noise contribution. It is actually due to this second term that the MMSEE differs from the ZFE (compare to (7.70)). Now the solution to the problem (7.141) is

$$\mathbf{F}(f) = \frac{1}{\sum_{i=1}^m |H_i(f)|^2 + \frac{\sigma_v^2}{\sigma_a^2}} \mathbf{H}^*(f) , \quad (7.142)$$

which gives the optimal infinite-length MMSE equalizer. Eq. (7.142) should be compared to (7.75). The similarity to the corresponding relation presented in chapter 2 is obvious. The additive term in the denominator of the expression appearing in (7.142) is the one that makes the compromise between reducing ISI and noise enhancement. It is also this term that protects against the infinite noise enhancement that can be produced by a ZFE since the denominator in (7.142) is always strictly positive. It is also worth noting that in contrast to the symbol-rate case in which the problem of infinite noise amplification of the ZFE appears when the channel has zeros on the unit circle, according to equation (7.75) and (7.142) in the multichannel case this will happen when the channels have zeros in common. This is one more point that shows that the counterpart of "zeros on the unit circle" in the multichannel case is "zeros in common".

In the noiseless case, according to (7.142) the optimal (infinite-length) MMSE and ZF equalizers coincide. However in the noisy case, the MMSE equalizer has a superior performance. This can be proven as follows: the output SNR in the case of the optimal MMSE is

$$\sigma_{MMSE}^2 = \sigma_v^2 \int_{-\frac{1}{2}}^{\frac{1}{2}} \frac{df}{\sum_{i=1}^m |H_i(f)|^2 + \frac{\sigma_v^2}{\sigma_a^2}} . \quad (7.143)$$

Due to the fact that $\frac{\sigma_v^2}{\sigma_a^2} > 0$, comparing (7.143) with (7.76) gives

$$\sigma_{MMSE}^2 < \sigma_{ZFE}^2 . \quad (7.144)$$

Therefore, as in the classical case, the optimal MMSE equalizer will always be superior to the corresponding ZF equalizer.

The relation between MMSEE's and ZFE's

In this section we will show the fact that there exists a direct relation linking FIR ZFE's to MMSEE's of equal length in the time-domain. The MMSEE can be written in the case of white additive Gaussian noise of variance σ_v^2 as

$$F_M = (R_L^x + \sigma_v^2 I_{Lm})^{-1} \mathbf{d}_i \quad (7.145)$$

(note that the term \mathbf{d}_i remains unaffected by the additive noise). The corresponding ZFE is given by

$$F_z = (R_L^x)^{-1} \mathbf{d}_i \quad (7.146)$$

Now by applying the matrix inversion lemma (MIL) to the inverse matrix appearing in (7.145) we get

$$(R_L^{\mathbf{x}_s} + \sigma_v^2 I_{L_m})^{-1} = (R_L^{\mathbf{x}_s})^{-1} - \sigma_v^2 (R_L^{\mathbf{x}_s})^{-1} [I_{L_m} + \sigma_v^2 (R_L^{\mathbf{x}_s})^{-1}]^{-1} (R_L^{\mathbf{x}_s})^{-1} ,$$

which gives combined with (7.146)

$$\begin{aligned} F_M &= F_Z - \sigma_v^2 (R_L^{\mathbf{x}_s})^{-1} [I_{L_m} + \sigma_v^2 (R_L^{\mathbf{x}_s})^{-1}]^{-1} F_Z , \\ F_M &= \left\{ I_{L_m} - \sigma_v^2 (R_L^{\mathbf{x}_s})^{-1} [I_{L_m} + \sigma_v^2 (R_L^{\mathbf{x}_s})^{-1}]^{-1} \right\} F_Z , \\ F_M &= \left\{ I_{L_m} - \sigma_v^2 (R_L^{\mathbf{x}_s})^{-1} [I_{L_m} - \sigma_v^2 (R_L^{\mathbf{x}_s})^{-1}] \right\} F_Z . \end{aligned}$$

The multiplying matrix appearing above can be simplified by noting that

$$\begin{aligned} R_L^{\mathbf{x}} &= R_L^{\mathbf{x}_s} + \sigma_v^2 I_{L_m} , \\ (R_L^{\mathbf{x}_s})^{-1} R_L^{\mathbf{x}} &= I_{L_m} + \sigma_v^2 (R_L^{\mathbf{x}_s})^{-1} , \\ (R_L^{\mathbf{x}_s})^{-1} R_L^{\mathbf{x}} - \sigma_v^2 (R_L^{\mathbf{x}_s})^{-1} &= I_{L_m} , \\ (R_L^{\mathbf{x}_s})^{-1} - \sigma_v^2 (R_L^{\mathbf{x}_s})^{-1} (R_L^{\mathbf{x}})^{-1} &= (R_L^{\mathbf{x}})^{-1} , \\ I_{L_m} - \sigma_v^2 (R_L^{\mathbf{x}_s})^{-1} + \sigma_v^4 (R_L^{\mathbf{x}_s})^{-1} (R_L^{\mathbf{x}})^{-1} &= I_{L_m} - \sigma_v^2 (R_L^{\mathbf{x}})^{-1} , \\ I_{L_m} - \sigma_v^2 (R_L^{\mathbf{x}_s})^{-1} [I_{L_m} - \sigma_v^2 (R_L^{\mathbf{x}_s})^{-1}] &= I_{L_m} - \sigma_v^2 (R_L^{\mathbf{x}})^{-1} . \end{aligned}$$

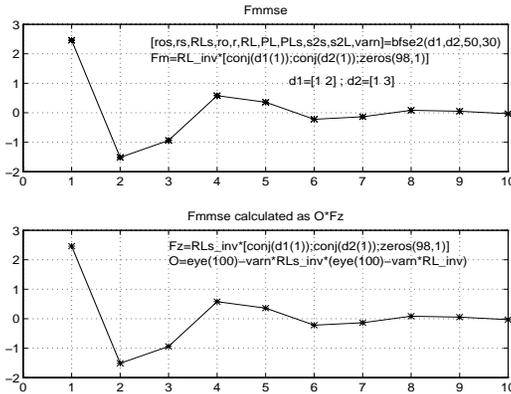


Figure 7.14: A verification of eq. (7.147)

Therefore the MMSEE and ZFE are related through

$$F_M = \Omega F_Z , \quad (7.147)$$

where Ω is the matrix defined as

$$\Omega = I_{L_m} - \sigma_v^2 (R_L^{\mathbf{x}})^{-1} \quad (7.148)$$

The usefulness of (7.148) is that it shows that there exists a simple linear relation that allows to obtain a MMSEE from the corresponding ZFE. A verification of (7.147) can be found in figure 7.14. We have simulated the simple case of two ($m = 2$) MA(1) channels with impulse responses $\mathbf{h}_1 = [1 \ 2]$, $\mathbf{h}_2 = [1 \ 3]$. The input sequency is binary with variance $\sigma_a^2 = 1$ and the noise variance $\sigma_v^2 = 0.005$ (which corresponds to an SNR of 30 dB at the output of the first channel). The equalizer length is $L = 50$. In the upper graph the MMSE equalizer (corresponding to a zero delay) obtained by (7.145) is plotted, whereas the lower graph shows the setting obtained by (7.147). As can be seen, the two settings are identical.

Moreover, the MMSEE according to (7.147) can be obtained blindly (it has been already shown in the previous section how the ZF equalizer can be obtained blindly). Eq. (7.147) however requires the inversion of the $L_m \times L_m$ matrix Ω , which can be costly in terms of computational complexity. In the next session we will show that a method with a reduced computational complexity can be derived based on linear prediction.

7.6.3 MMSE equalization and linear prediction

In the case of zero-delay in the criterion (7.147) ($i = 0$), the corresponding MMSE can be easily found by LP as follows: we have from section 7.5 that

$$[I_m - \mathbf{P}_L]^H = R_L^{\mathbf{x}}^{-1} \begin{bmatrix} I_{L_m} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \sigma_{\mathbf{x}}^2 \quad (7.149)$$

In the case of zero-delay, according to (7.137) and (7.139), the MMSE equalizer takes the form

$$F_{MMSE_0} = (R_L^{\mathbf{x}})^{-1} \begin{bmatrix} I_{L_m} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \sigma_a^2 \mathbf{h}_0^* \sigma_a^2 . \quad (7.150)$$

From (7.149) and (7.150) we deduce that

$$F_{MMSE_0} = \begin{bmatrix} I_m \\ -\mathbf{P}_{L-1}^H \end{bmatrix} \sigma_{\mathbf{x}}^{-2} \mathbf{h}_0^* \sigma_a^2 , \quad (7.151)$$

where $\sigma_{\mathbf{x}}^2$ is the prediction error matrix. Eq. (7.151) offers an alternative to (7.147) in that it gives a different way to obtain the MMSE equalizer. Now the MMSEE is obtained by performing first linear prediction in the received vector sequence. All this can be done blindly as discussed in the previous section. Moreover, (7.151) has the advantage that only the inversion of the $m \times m$ prediction-error matrix that is used instead of the $L_m \times L_m$ covariance matrix of the received signal needed in (7.147). This leads to a substantial reduction in computational complexity.

Figure 7.15 shows an example that demonstrates the superiority of MMSE with respect to ZF equalizers as well as the performance of the method described by (7.151). We use the typical multipath radio channel given in [SGKC91] which has been oversampled at a double symbol rate ($m = 2$). The SNR at the equalizer input is 30 dB . The solid and dashed horizontal lines represent the output SNR (see (7.133)) produced by the corresponding MMSEE and ZFE equalizers (both for 0 delay), respectively. One may note the superior performance of the MMSEE (as predicted by (7.144)). The other dashed line shows the performance attained by the equalizer derived from (7.151) for different amounts of data. Note that asymptotically (as the number of available grows towards infinity) the curve attains the performance of the MMSEE thus verifying (7.151). In this figure it is also worth noting that all the performances

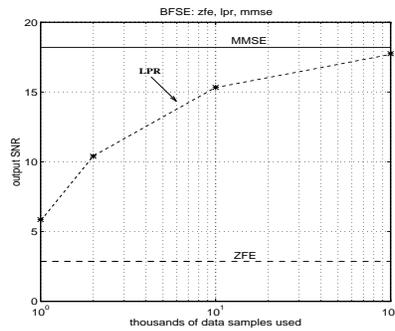


Figure 7.15: MMSE, LP and ZF equalization

are very poor. This is due to the fact that we have only considered 0 delay. In the next subsection we discuss this problem.

7.6.4 The influence of the lag

It is well-known that the delay parameter i in the MMSE criterion is of crucial importance to the performance of the equalizer in the symbol-rate case. No exact method to choose this parameter optimally exists up to our knowledge, a practical guideline being that it should be such that all the channel coefficients appear in the cross-correlation \mathbf{d}_i vector that post-multiplies the covariance matrix in the expression of the MMSE equalizer. In the case of

fractionally spaced equalization, when this happens, the MMSE equalizer will look like

$$\tilde{F}_{MMSE,i} = E \left(\mathbf{X}_L(k) \mathbf{X}_L^H(k) \right)^{-1} E \left(\mathbf{X}_L(k) a_{k-i} \right) = \left(R_L^{\mathbf{x}} \right)^{-1} \begin{bmatrix} \mathbf{0} \\ \mathbf{h}^*(L-1) \\ \vdots \\ \mathbf{h}^*(0) \\ \mathbf{0} \end{bmatrix}. \quad (7.152)$$

On the other hand, when this does not happen, there will be some channel coefficients missing either from the top or the bottom of \mathbf{d}_i . In these cases the performance is expected to degrade exactly for the same reasons that this happens in the symbol-rate equalizers.

In order to provide an example where this phenomenon shows up we will compare different MMSEE's corresponding to different delays by plotting the corresponding output SNR's. Figure 7.16 shows how the output SNR is influenced by the delay parameter i for the channel that was simulated in the previous example. Note how much the performance is getting improved as the delay increases and how it decreases again when it gets too big. Therefore the same kind of behaviour as in classical cases is observed.

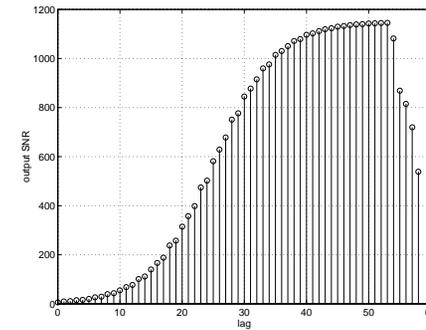


Figure 7.16: MMSEE performance: the influence of the lag

Discussion:

At this point it is worth noting two more things. First, the relation (7.147) is valid for any delay i , which means that if a ZFE corresponding to any delay is already known,¹¹ it is straightforward to obtain the MMSEE corresponding to the same delay. Another thing is that in the absence of noise, the ZFE (and therefore the MMSEE also) corresponding to a specific

¹¹For example by using the n -step ahead prediction described in section 7.5

delay can be adjusted to a different delay by a simple shift. However this is obviously not the case in the presence of noise.

7.6.5 Further discussion

In this section we have been interested in the problem of MMSE equalization in the multi-channel context. Seen either from the point of view of fractionally-spaced equalization or from that of reception through multiple antennas, this problem seems to be receiving quite some attention nowadays. In [Cre94], an interesting geometrical interpretation of the MMSE fractionally-spaced equalizer is given as well as an analysis that shows the advantages of the oversampling case w.r.t. the symbol-rate one. In [BD94] a performance analysis is given for a decision feedback equalizer in the case of reception through an antenna array. In this study it is also shown also that the performance improves as the diversity (in this case the number of receiving antennas) increases. It is therefore possible that using diversity in one way or another, linear equalizer methods can be used in such a way so as to achieve good performance for several systems.

7.7 Signal and Noise subspaces

7.7.1 Parameterization of the signal and noise subspaces

Suppose now that we have additive white noise $v(t)$ with zero mean and unknown variance σ_v^2 (in the complex case, real and imaginary parts are assumed to be uncorrelated, colored noise could equally well be handled). Then since

$$\mathbf{R}_L^{\mathbf{X}} = \mathcal{T}_L(\mathbf{H}_N) \mathbf{R}_{L+N-1}^i \mathcal{T}_L^H(\mathbf{H}_N) + \sigma_v^2 \mathbf{I}_{Lm}, \quad (7.153)$$

for $L \geq \underline{L}$, σ_v^2 can be identified as the smallest eigenvalue of $\mathbf{R}_L^{\mathbf{X}}$. Replacing $\mathbf{R}_L^{\mathbf{X}}$ by $\mathbf{R}_L^{\mathbf{X}} - \sigma_v^2 \mathbf{I}_{Lm}$, all results of the prediction approach in the noiseless case still hold. Given the structure of $\mathbf{R}_L^{\mathbf{X}}$ in (7.153), the column space of $\mathcal{T}_L(\mathbf{H}_N)$ is called the signal subspace and its orthogonal complement the noise subspace. In [Slo94a], a linear parameterization of the noise subspace is given in terms of a blocking equalizer for $m = 2$. For $m > 2$, a linear minimal parameterization can be found by using linear prediction [Slo94c].

Consider now the eigendecomposition of $\mathbf{R}_L^{\mathbf{X}}$ of which the real nonnegative eigenvalues are ordered in descending order:

$$\begin{aligned} \mathbf{R}_L^{\mathbf{X}} &= \sum_{i=1}^{L+N-1} \lambda_i V_i V_i^H + \sum_{i=L+N}^{Lm} \lambda_i V_i V_i^H \\ &= V_S \Lambda_S V_S^H + V_N \Lambda_N V_N^H, \end{aligned} \quad (7.154)$$

where $\Lambda_N = \sigma_v^2 \mathbf{I}_{(m-1)L-N+1}$ (see (7.153)). Assuming $\mathcal{T}_L(\mathbf{H}_N)$ and \mathbf{R}_{L+N-1}^i to have full rank, the sets of eigenvectors V_S and V_N are orthogonal: $V_S^H V_N = \mathbf{0}$, and $\lambda_i > \sigma_v^2$, $i = 1, \dots, L+N-1$.

We then have the following equivalent descriptions of the signal and noise subspaces

$$\text{Range}\{V_S\} = \text{Range}\{\mathcal{T}_L(\mathbf{H}_N)\}, \quad V_N^H \mathcal{T}_L(\mathbf{H}_N) = \mathbf{0}. \quad (7.155)$$

7.7.2 Sinusoids in noise: comparison

The form of the covariance matrix $\mathbf{R}_L^{\mathbf{X}}$, as it appears in (7.153), is similar to the form of the covariance matrix that appears in the classical problem of sinusoids in noise. Indeed, in the latter case the covariance matrix has the form

$$R = V D V^H + \sigma_v^2 \mathbf{I}, \quad (7.156)$$

where V is a Vandermonde matrix. Now the signal subspace is the column space of V , and the noise subspace is its orthogonal complement. The noise subspace in this case is the column space of a Toeplitz matrix that contains the prediction filters for each sinusoid (a sinusoid can be perfectly predicted). On the other hand, in the case of (7.153), the signal subspace is the column space of $\mathcal{T}_L(\mathbf{H}_N)$ which is a block Toeplitz matrix. The noise subspace is the column space of a matrix that contains $(m-1)L-N+1$ blocking equalizers (see [Slo94c]). It has the form of a banded block-Toeplitz matrix (apart from some missing lines in the first block). The following table shows a comparison between of the signal and noise subspaces in the two cases.

	Sin. in noise	SIMO channel
Signal Subspace	Vandermonde matrix	Toeplitz matrix
Noise Subspace	Toeplitz matrix	Block-Toeplitz matrix

7.8 Channel Estimation by Subspace Fitting

When the covariance matrix is estimated from data, it will no longer satisfy exactly the properties we have elaborated upon. We assume that the detection problem of the signal subspace dimension $L+N-1$ has been solved correctly. The signal subspace will now be defined as the space spanned by the eigenvectors corresponding to the $L+N-1$ largest eigenvalues, while the noise subspace is its orthogonal complement. Consider now the following subspace fitting problem

$$\min_{\mathbf{H}_N, T} \|\mathcal{T}_L(\mathbf{H}_N) - V_S T\|_F, \quad (7.157)$$

where the Frobenius norm of a matrix Z can be defined in terms of the trace operator: $\|Z\|_F^2 = \text{tr}\{Z^H Z\}$. The problem considered in (7.157) is quadratic in both \mathbf{H}_N and T . If V_S contains the signal subspace eigenvectors of the actual covariance matrix $\mathbf{R}_L^{\mathbf{X}}$, then the minimal value of the cost function in (7.157) is zero. If $\mathbf{R}_L^{\mathbf{X}}$ is estimated from a finite amount of data however, then its eigenvectors (and eigenvalues) are perturbed w.r.t. their theoretical values. Therefore, in general there will be no value for \mathbf{H}_N for which the column space of

$\mathcal{T}_L(\mathbf{H}_N)$ coincides with the signal subspace $\text{Range}\{V_S\}$. But it is clearly meaningful to try to estimate \mathbf{H}_N by taking that $\mathcal{T}_L(\mathbf{H}_N)$ into which V_S can be transformed with minimal cost. This leads to the subspace fitting problem in (7.157). The optimization problem in (7.157) is separable. With \mathbf{H}_N fixed, the optimal matrix T can be found to be (assuming $V_S^H V_S = I$)

$$T = V_S^H \mathcal{T}_L(\mathbf{H}_N). \quad (7.158)$$

Using (7.158) and the commutativity of the convolution operator as in (7.110), one can show that (7.157) is equivalent to

$$\begin{aligned} \min_{\mathbf{H}_N} \mathbf{H}_N^t \left(\sum_{i=(m-1)L-N+1}^{Lm} \mathcal{T}_L(V_i^{Ht}) \mathcal{T}_L^H(V_i^{Ht}) \right) \mathbf{H}_N^{tH} \\ = \min_{\mathbf{H}_N} \left[L \|\mathbf{H}_N^t\|_2^2 - \mathbf{H}_N^t \left(\sum_{i=1}^{L+N-1} \mathcal{T}_L(V_i^{Ht}) \mathcal{T}_L^H(V_i^{Ht}) \right) \mathbf{H}_N^{tH} \right], \end{aligned} \quad (7.159)$$

where V_i^{Ht} (like \mathbf{F}_L) is considered a block vector with L blocks of size $1 \times m$. These optimization problems have to be augmented with a nontriviality constraint on \mathbf{H}_N^t . In case we choose the quadratic constraint $\|\mathbf{H}_N^t\|_2 = 1$, then the last term in (7.159) leads equivalently to

$$\max_{\|\mathbf{H}_N^t\|_2=1} \mathbf{H}_N^t \left(\sum_{i=1}^{L+N-1} \mathcal{T}_L(V_i^{Ht}) \mathcal{T}_L^H(V_i^{Ht}) \right) \mathbf{H}_N^{tH} \quad (7.160)$$

the solution of which is the eigenvector corresponding to the maximum eigenvalue of the matrix appearing between the brackets.

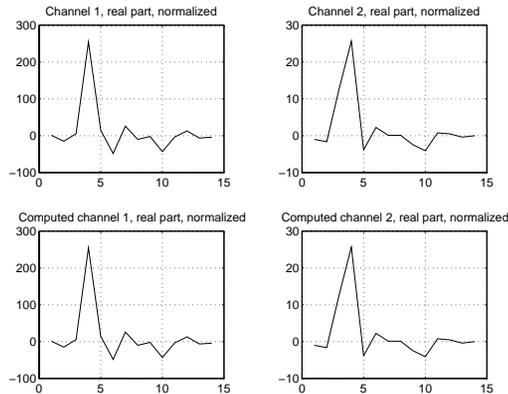


Figure 7.17: Channel id in the case of a noisy but perfectly estimated cov. matrix

7.9 Channel Estimation from Data using Conditional ML

Maximum Likelihood estimation is by definition the most efficient (in terms of estimation accuracy) technique when one has to deal with a finite amount of data. In order to be able to implement ML techniques, one needs to know the way that the data set depends on the parameters, as well as the statistical description of the parameters and the noise that usually corrupts the data. This results in a likelihood function, which is the criterion that has to be optimized.

In some cases however, the actual pdf of the parameters is not used (either because it is not available or because using it would result to a more computationally demanding method). Then we usually attribute to the parameters an arbitrary (not corresponding to the true) statistical distribution. The resulting problem is then called a *Conditional Maximum Likelihood* (CML) technique. This problem is sub-optimal in the sense that it will always contain some estimation error due to its modeling, but may nevertheless combine a good estimation quality with a low computational complexity. In the sequel of this section we will propose two different CML techniques for the joint problem of estimating the transmitted data and channel parameters from the available channel output data set. The reason why we propose such techniques is that in the case of given data (samples of $y(\cdot)$), the subspace fitting approach of the previous section involves the data through the sample covariance matrix, which leads to computationally tractable optimization problems, but not to very efficient estimates.

7.9.1 Deterministic Maximum Likelihood

As its name indicates, the principle of DML is to consider some of the stochastic parameters as deterministic quantities. The formulation of the problem in the multichannel context is given below.

Problem formulation

We make the following assumption for the transmitted data parameters

- The transmitted data $\{a(k)\}$ are assumed to be deterministic quantities.

This means that not only we do not assume the actual marginal pdf for the transmitted data, but that moreover, we consider them to be deterministic quantities. The channel parameters are of course also assumed to be deterministic quantities, and therefore, the only stochastic part is assumed to come from the additive noise at the channel output. We now make the following assumption for the statistical description of the additive noise.

- The additive noise samples $\{v(k)\}$ are assumed to be i.i.d. complex random variables, that are Gaussian with zero mean and unknown variance σ_v^2 . We also assume, as usually, that the real and imaginary parts of the noise are independent between them.

Therefore, the additive noise variance is another deterministic parameter that can be estimated.

We finally assume a snapshot of M vector channel outputs to be available.

$$\mathbf{X}_M(k) = \begin{bmatrix} \mathbf{x}(k) \\ \vdots \\ \mathbf{x}(k - M + 1) \end{bmatrix}$$

As the only stochastic part comes from the noise, the likelihood function is the pdf of the noise, and we write it in such a way so as to make appear the involved parameters in it. The noise samples are expressed in terms of the channel and symbol parameters as

$$\mathbf{V}_M(k) = \mathbf{X}_M(k) - \mathcal{T}_M(\mathbf{H}_N) A_{M+N-1}(k) , \quad (7.161)$$

and therefore the likelihood function is

$$p(\mathbf{X}_M(k); \theta(k)) = \frac{1}{(2\pi\sigma_v^2)^{mM/2}} e^{-\frac{1}{2\sigma_v^2} \|\mathbf{X}_M(k) - \mathcal{T}_M(\mathbf{H}_N) A_{M+N-1}(k)\|^2} , \quad (7.162)$$

where $\theta(k)$ represents the parameter vector defined as

$$\theta(k) = [A_{M+N-1}^T(k) H_1 \cdots H_m]^T . \quad (7.163)$$

The goal is to determine the parameter vector $\theta(k)$ that maximizes the likelihood that the received data snapshot has been produced in the presence of additive white Gaussian noise:

$$\max_{\theta(k)} p(\mathbf{X}_M(k); \theta(k)) . \quad (7.164)$$

The maximization of the likelihood function therefore boils down to the following least-squares problem

$$\min_{\mathbf{H}_N, A_{M+N-1}(k)} \|\mathbf{X}_M(k) - \mathcal{T}_M(\mathbf{H}_N) A_{M+N-1}(k)\|_2^2 . \quad (7.165)$$

The optimization problem in (7.165) is separable. We therefore proceed in two steps. Assuming \mathbf{H}_N to be a constant, the problem (7.165) is a classical least-squares problem for the parameters $A_{M+N-1}(k)$ whose solution is

$$A_{M+N-1}(k) = (\mathcal{T}_M^H(\mathbf{H}_N) \mathcal{T}_M(\mathbf{H}_N))^{-1} \mathcal{T}_M^H(\mathbf{H}_N) \mathbf{X}_M(k) . \quad (7.166)$$

Replacing the expression (7.166) in (7.165) we get the following equivalent expression for (7.165)

$$\min_{\mathbf{H}_N, A_{M+N-1}(k)} \|\mathbf{X}_M(k) (I_{mM} - \mathcal{T}_M(\mathbf{H}_N) (\mathcal{T}_M^H(\mathbf{H}_N) \mathcal{T}_M(\mathbf{H}_N))^{-1} \mathcal{T}_M^H(\mathbf{H}_N))\|_2^2 , \quad (7.167)$$

which can be written as

$$\begin{cases} \min_{\mathbf{H}_N} \left\| P_{\mathcal{T}_M(\mathbf{H}_N)}^\perp \mathbf{X}_M(k) \right\|_2^2 & \text{or} \\ \max_{\mathbf{H}_N} \left\| P_{\mathcal{T}_M(\mathbf{H}_N)} \mathbf{X}_M(k) \right\|_2^2 , \end{cases} \quad (7.168)$$

since the projection operators are defined as

$$\begin{cases} P_{\mathcal{T}_M(\mathbf{H}_N)} = \mathcal{T}_M(\mathbf{H}_N) (\mathcal{T}_M^H(\mathbf{H}_N) \mathcal{T}_M(\mathbf{H}_N))^{-1} \mathcal{T}_M^H(\mathbf{H}_N) \\ P_{\mathcal{T}_M(\mathbf{H}_N)}^\perp = I_{mM} - P_{\mathcal{T}_M(\mathbf{H}_N)} . \end{cases} \quad (7.169)$$

Since $\mathcal{T}_M^H(\mathbf{H}_N) \mathbf{X}_M(k) = \mathcal{T}_N^T(\mathbf{X}_M^T(k)) \mathbf{H}_N^H$, we can rewrite the second approach in (7.168) as

$$\max_{\mathbf{H}_N} \mathbf{H}_N^H \mathcal{T}_N^s(\mathbf{X}_M^T(k)) \left(\mathcal{T}_M^H(\mathbf{H}_N) \mathcal{T}_M(\mathbf{H}_N) \right)^{-1} \mathcal{T}_N^T(\mathbf{X}_M^T(k)) \mathbf{H}_N^H . \quad (7.170)$$

Algorithmic organization

The optimization problem (7.170) is highly nonlinear w.r.t. the channel parameters. Moreover, it has to be augmented with a non-triviality constraint such as for example a norm restriction on the channel. In this case, one might think of solving it iteratively following a 2-step procedure:

- Use the previous estimate of \mathbf{H}_N to calculate the inverse matrix appearing in the middle of the expression (7.170).
- Then maximize the expression (7.170) considering as only parameters the channel matrices appearing as outer factors in (7.170).

The second step mentioned above can be easily solved since it has a classical quadratic form and has analytical solutions depending on the non-triviality constraints imposed. Of course a good initialization for this method is needed, which may be provided for example by the subspace-fitting approach discussed above.

However, the fact that the problem (7.170) is a maximization and not a minimization one poses a problem in what concerns the success of such an iterative procedure.¹² We are therefore interested in replacing the problem (7.170) by a similar minimization problem. We now discern between two cases:

- $m = 2$

In (7.170), the problem is parameterized in terms of the signal subspace parameters (i.e. the channel coefficients). A minimization problem would appear if one were able of finding a minimal parameterization of the noise subspace (by minimal parameterization we mean one in which the degrees of freedom used equals the number of parameters to be determined,

¹²This has to do with the fact that, contrary to what happens in the corresponding minimization problem, in the maximization one, the value of the middle matrix influences the global solution even in the absence of noise. On the other hand, in the minimization problem, one will always find the correct solution in the absence of noise, for any matrix in the middle (the role of the middle matrix in this case is only to "weight" the expression and has an optimal setting in the noisy case). This is why an iterative procedure will be successful for the minimization problem.

in our case, the channel coefficients). In the case $m = 2$, as we already saw, a minimal parameterization of the noise subspace is achieved if we parameterize everything in terms of the blocking equalizer. Assuming that $\mathcal{T}_M(\mathbf{H}_N)$ has full rank, as already mentioned, $\mathcal{T}_M(\mathbf{H}_N)$ spans a space orthogonal to the one spanned by $\mathcal{T}_{M-N+1}^H(\mathbf{F}_N^b)$. Therefore the orthogonal projections w.r.t. $\mathcal{T}_M(\mathbf{H}_N)$ correspond to orthogonal projection complements to $\mathcal{T}_{M-N+1}^H(\mathbf{F}_N^b)$ and vice versa. Therefore the problem (7.168) can be written as

$$\begin{aligned} \min_{\mathbf{H}_N} \left\| P_{\mathcal{T}_M(\mathbf{H}_N)}^\perp \mathbf{X}_M(k) \right\|_2^2 &= \min_{\mathbf{F}_N^b} \left\| P_{\mathcal{T}_{M-N+1}^H(\mathbf{F}_N^b)} \mathbf{X}_M(k) \right\|_2^2 = \\ &= \min_{\mathbf{F}_N^b} \mathbf{X}_M^H(k) \mathcal{T}_{M-N+1}^H(\mathbf{F}_N^b) \left(\mathcal{T}_{M-N+1}(\mathbf{F}_N^b) \mathcal{T}_{M-N+1}^H(\mathbf{F}_N^b) \right)^{-1} \mathcal{T}_{M-N+1}(\mathbf{F}_N^b) \mathbf{X}_M(k). \end{aligned}$$

Now because of the commutativity of convolution we can write

$$\mathcal{T}_{M-N+1}(\mathbf{F}_N^b) \mathbf{X}_M(k) = \mathcal{H}_{M-N+1}(\mathbf{X}_M^b(k)) \mathbf{F}_N^{bT}, \quad (7.171)$$

where

$$\mathbf{X}_M^b(k) = \begin{bmatrix} \mathbf{x}^T(k) \\ \vdots \\ \mathbf{x}^T(k-M+1) \end{bmatrix} \quad (7.172)$$

and $\mathcal{H}_L(\mathbf{u})$ is a block Hankel matrix with L block rows, obtained by taking the block entries from the block vector \mathbf{u} and filling up a Hankel matrix starting from the top left corner:

$$\mathbf{u} = \begin{bmatrix} \mathbf{u}(0) \\ \vdots \\ \mathbf{u}(M-1) \end{bmatrix}, \quad \mathcal{H}_L(\mathbf{u}) = \begin{bmatrix} \mathbf{u}(0) & \mathbf{u}(1) & \cdots & \mathbf{u}(M-L) \\ \mathbf{u}(1) & \ddots & \ddots & \mathbf{u}(M-L+1) \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{u}(L-1) & \cdots & \cdots & \mathbf{u}(M-1) \end{bmatrix}. \quad (7.173)$$

Taking into account (7.171), the minimization problem takes now the form

$$\min_{\mathbf{F}_N^b} \mathbf{F}_N^{b*} \mathcal{H}_{M-N+1}^H(\mathbf{X}_M^b(k)) \left(\mathcal{T}_{M-N+1}(\mathbf{F}_N^b) \mathcal{T}_{M-N+1}^H(\mathbf{F}_N^b) \right)^{-1} \mathcal{H}_{M-N+1}(\mathbf{X}_M^b(k)) \mathbf{F}_N^b. \quad (7.174)$$

The minimization problem (7.174) can now be easily solved in an iterative way as described before. As the cost function is nonlinear, it is important to have a good initial guess (e.g. use the outcome of the subspace fitting approach mentioned before). It can be proven that if the initial estimate is consistent, then one iteration of the iterative solution suffices to obtain an asymptotically best consistent (ABC) estimate.¹³

- $m > 2$

¹³The problem (7.174) has a form identical to the one presented in [BM86], even though the latter corresponds to a different (DOA) context. Moreover, the same iterative procedure is proposed and simulated as well.

In order to have a minimization problem as in (7.174) we need again a minimal parameterization of the noise subspace. The signal subspace (which is the column space of $\mathcal{T}_M(\mathbf{H}_N)$) has $mN - 1$ degrees of freedom (the channel coefficients apart from a multiplicative factor). The noise subspace can also be determined as the column space of (a slight modification of) some block Toeplitz matrix $\mathcal{G}(G)$ which is determined by G . As shown previously, G can be taken to be (an appropriate parameterization of) the first set of $m - 1$ prediction filters that give zero prediction error (in the noiseless case with uncorrelated transmitted symbols). Each such filter corresponds to a blocking equalizer. The number of degrees of freedom in G is again $mN - 1$. \mathbf{H}_N (apart from a scaling factor) and G can be uniquely determined from one another. The optimization problem now becomes

$$\min_G \left\| P_{\mathcal{G}(G)} \mathbf{X}_M(k) \right\|_2^2. \quad (7.175)$$

Due to the commutativity of convolution, we can write

$$\mathcal{G}(G) \mathbf{X}_M(k) = \mathcal{X}(\mathbf{X}_M(k)) [\mathbf{1} \ G^T]^T.$$

Therefore (7.175) can be written as

$$\min_G \begin{bmatrix} \mathbf{1} \\ G \end{bmatrix}^H \mathcal{X}^H(\mathbf{X}_M(k)) \left(\mathcal{G}^H(G) \mathcal{G}(G) \right)^{-1} \mathcal{X}(\mathbf{X}_M(k)) \begin{bmatrix} \mathbf{1} \\ G \end{bmatrix} \quad (7.176)$$

The optimization problem (7.176) now has the same form as (7.174) and can be iteratively minimized as before. This is therefore the method that should be used for the general case of $m > 2$.

Performance analysis: Cramer-Rao bounds

In this paragraph we will pursue a performance analysis of the DML method above by determining the corresponding Cramer-Rao bounds.

We consider first the case $m = 2$. In order to discriminate between the channel and the transmitted symbol parameters, we partition the parameter vector $\theta(k)$ defined in (7.163) as

$$\theta(k) = [b^T \ a^T]^T, \quad (7.177)$$

where $b^T = A_{M+N-1}^T(k)$ and $a^T = [H_1 \ H_2]$. Now our problem has the same form as the linear separable problem appearing in [MS93]:

$$\mathbf{X}_M^b = \mathcal{T}_M b \quad ; \quad \mathcal{T}_M = \mathcal{T}_M(a), \quad (7.178)$$

where \mathbf{X}_M^b denotes the noiseless data vector and where we have dropped all indices indicating time. The Fisher information matrix $J(\theta)$ is then given by

$$J(\theta) = \frac{1}{\sigma_v^2} \begin{pmatrix} \frac{\partial x^T}{\partial \theta} \\ \frac{\partial x^T}{\partial \theta} \end{pmatrix}^T. \quad (7.179)$$

Taking into account the structure of the linear separable model, the Fisher matrix take the form

$$J(b, a) = \frac{1}{\sigma_v^2} \begin{bmatrix} H^H H & H^H K \\ K^H H & K^H K \end{bmatrix}, \quad (7.180)$$

where the elements of the matrices H and K are defined as

$$\begin{aligned} H &= [h_i] \quad h_i = \frac{\partial \mathbf{X}_M}{\partial b_i} \\ K &= [k_i] \quad k_i = \frac{\partial \mathbf{X}_M}{\partial a_i} = \frac{\partial H}{\partial a_i} b. \end{aligned} \quad (7.181)$$

By performing the necessary derivations, we find that the matrix K is given as

$$K = \begin{bmatrix} a_0 & a_1 & \cdots & a_{N-1} & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & a_0 & a_1 & \cdots & a_{N-1} \\ a_1 & a_2 & \cdots & a_N & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & a_1 & a_2 & \cdots & a_N \\ \vdots & \vdots \\ a_{M-1} & a_M & \cdots & a_{M+N-2} & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & a_{M-1} & a_M & \cdots & a_{M+N-2} \end{bmatrix}. \quad (7.182)$$

The CR bounds are found from the inverse of the Fisher matrix which has the form

$$J^{-1}(a, b) = \sigma_v^2 \begin{bmatrix} [H^H(I - P_K)H]^{-1} & F \\ F^H & [K^H(I - P_H)K]^{-1} \end{bmatrix}. \quad (7.183)$$

We are primarily interested in the CR bound for the channel, which is therefore given as

$$C(a) \geq \sigma_v^2 (K^H Z K)^{-1}, \quad (7.184)$$

where Z is

$$Z = I_{2M} - \mathcal{T}_M (\mathcal{T}_M^H \mathcal{T}_M)^{-1} \mathcal{T}_M^H. \quad (7.185)$$

The generalization to the case $m > 2$ is straightforward. By noting that the quantity $\mathcal{T}_M(\mathbf{H}_N) A_{M+N-1}(k)$ can be expressed as

$$\mathcal{T}_M(\mathbf{H}_N) A_{M+N-1}(k) = \mathcal{A}_{M,N}(k) \mathbf{H}_N^T,$$

where $\mathcal{A}_{M,N}(k) = A_{M,N}(k) \otimes I_m$ and

$$\mathcal{A}_{M,N}(k) = \begin{bmatrix} a(k) & \cdots & a(k-N+1) \\ \vdots & \ddots & \vdots \\ a(k-M+1) & \cdots & a(k-M-N+2) \end{bmatrix}, \quad (7.186)$$

(Hankel matrix), the CR bound for the case $m > 2$ takes the form (if we consider the estimation of the channel modulo the problem of determining the proper scale factor)

$$C(\hat{\mathbf{H}}_N^T) \geq \sigma_v^2 \left[\mathcal{A}_{M,N}^H(k) P_{\mathcal{T}_M(\mathbf{H}_N)}^\perp \mathcal{A}_{M,N}(k) \right]^+, \quad (7.187)$$

where $^+$ denotes pseudo-inverse. Note that this leads to a singularity in the joint information matrix for $A_{M+N-1}(k)$ and \mathbf{H}_N^T , which translates into a singularity for the information matrix for \mathbf{H}_N^T separately (we can only determine \mathbf{H}_N up to a scalar multiple).

In order to get an idea of the quality of this estimation, we will compare it to the one that would be obtained if we had at our disposal a training sequence, i.e. if the data were known.

Suppose that we have a set of L symbols :

$$A_L = \{a_0 \ a_1 \ \cdots \ a_{L-1}\}.$$

Then the noiseless output would be given as

$$\mathbf{X}' = H_L A_L, \quad (7.188)$$

where H_L is defined as

$$H_L = \begin{bmatrix} H_1(0) & \cdots & H_1(N-1) & 0 & \cdots & 0 \\ 0 & H_1(0) & \cdots & H_1(N-1) & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & H_1(0) & \cdots & H_1(N-1) \end{bmatrix}. \quad (7.189)$$

Now the corresponding matrix K' will be

$$K' = \begin{bmatrix} a_0 & \cdots & a_{N-1} \\ a_1 & \cdots & a_N \\ \vdots & \vdots & \vdots \\ a_{L-N} & \cdots & a_{L-1} \end{bmatrix}. \quad (7.190)$$

Therefore the corresponding CR bound (per channel) is

$$C(H_1) \geq \sigma_v^2 (K'^H K')^{-1}. \quad (7.191)$$

Now note that the following inequality holds

$$\left[\mathcal{A}_{M,N}^H(k) P_{\mathcal{T}_M(\mathbf{H}_N)}^\perp \mathcal{A}_{M,N}(k) \right]^+ \geq \left[\mathcal{A}_{M,N}^H(k) \mathcal{A}_{M,N}(k) \right]^{-1}. \quad (7.192)$$

The last expression is according to (7.191) the CR bound if the data $A_{M+N-1}(k)$ were known (training sequence). For small m (e.g. 2), we find that the quality of the channel estimate may be relatively bad if the channel impulse response tapers off near the ends (channel length detection problem!). For large m however, the CR bound approaches the value corresponding to known data (which is independent of the channel)! These remarks can be seen in the following computer simulation example:

Examples

We will present some computer simulation examples in which we compare the CR bounds obtained by the DML method to the ones obtained by the corresponding ML method that uses a training sequence.

- First example: $m = 2$, the two channels have impulse responses given as

$$\begin{aligned} h_1 &= [0.6912 \quad - .5364 \quad - 0.0434 \quad 0 \quad 0 \quad 0] \\ h_2 &= [0.6771 \quad 0.3404 \quad 0 \quad 0 \quad 0 \quad 0] \end{aligned}$$

($N = 6$). We assume $M = 500$ data. The average CR bound/channel then turns to be

$$CR_1 = \sigma_v^2 \times 10.1382 .$$

Now suppose we make the same estimation based on 26 only training data¹⁴. Then the corresponding CR bound is

$$CR_{t,1} = \sigma_v^2 \times 0.4341 .$$

This shows that the estimation by the DML method, based even on as much as 500 data samples is clearly worse than the one of a training sequence of only 26 samples. This has to do primarily with two factors: the fact that the channel have 0 coefficients at their ends and the fact that the oversampling factor is small. We now present a counterexample.

- Second example: $m = 8$, the eight channels have impulse responses given as

$$\begin{aligned} h_1 &= [0.8259 \quad 1.0397 \quad - 0.6167 \quad - 0.0823 \quad 0.2224 \quad 0.1834] \\ h_2 &= [-1.3291 \quad 0.0389 \quad 0.4373 \quad 0.0228 \quad - 0.0445 \quad - 0.0144] \\ h_3 &= [-0.0365 \quad 0.2892 \quad 0.0972 \quad - 0.0179 \quad 0.0036 \quad 0.0454] \\ h_4 &= [0.4128 \quad - 0.4918 \quad 0.0664 \quad - 0.0193 \quad 0.0179 \quad - 0.0273] \\ h_5 &= [0.5506 \quad 0.3144 \quad - 0.0803 \quad - 0.0187 \quad 0.0535 \quad 0.0426] \\ h_6 &= [-0.8631 \quad 0.1622 \quad 0.0394 \quad - 0.0286 \quad - 0.0403 \quad 0.0155] \\ h_7 &= [-0.5345 \quad 0.2110 \quad - 0.0068 \quad - 0.0549 \quad 0.0255 \quad - 0.0409] \\ h_8 &= [-0.2299 \quad 0.5997 \quad - 0.0253 \quad - 0.0055 \quad 0.0310 \quad 0.0577] \end{aligned}$$

The corresponding CR bound based on 148 data samples with the DML method is

$$CR_2 = \sigma_v^2 \times 0.0921 .$$

whereas the corresponding CR bound based on 26 training data is

$$CR_{t,2} = \sigma_v^2 \times 0.4221 .$$

In this case the blind method performs clearly better. This reflects the severe oversampling employed as well as the absence of 0 coefficients in the channels.

¹⁴This is the case for the training sequence of a GSM time-slot.

7.9.2 Conditional ML assuming a Gaussian prior for the symbols

A problem with the optimization criterion (7.165) is that the joint information Fisher matrix for the channel and the data is singular, thus necessitating the use of a pseudo-inverse operation for the acquisition of the Cramer-Rao bound for the channel estimation. Moreover, computer simulations have shown that for small m , the channel estimation can be relatively bad if the channel impulse response tapers off near the ends.

We now propose a modified DML method that seems to avoid these problems: we consider the transmitted symbols to be no longer deterministic quantities but random variables that obey to a zero-mean Gaussian distribution of variance σ_a^2 . This may be an unrealistic assumption, it helps however to regulate the ML problem and provide a better estimation quality. The conditional likelihood function will now be

$$f(X, A|H) = f(X|A, H) f(A) . \quad (7.193)$$

$f(X|A, H)$ represents the likelihood function conditioned on both the channel and transmitted data :

$$f(X|A, H) = \frac{1}{(2\pi\sigma_v^2)^{mM/2}} e^{-\frac{1}{2\sigma_v^2} \|\mathbf{X}_M(k) - \mathcal{T}_M(\mathbf{H}_N)A_{M+N-1}(k)\|^2} , \quad (7.194)$$

and corresponds to the ML function of the previous DML method. $f(A)$ now represents the likelihood function for the assumed Gaussian transmitted data:

$$f(A) = \frac{1}{(2\pi\sigma_a^2)^{mM/2}} e^{-\frac{1}{2\sigma_a^2} \|A_{M+N-1}(k)\|^2} \quad (7.195)$$

Combining (7.194) and (7.195) we obtain the ML function for our problem:

$$f(X|A, H) = \frac{1}{(2\pi\sigma_v^2)^{mM/2}} e^{-\frac{1}{2\sigma_v^2} \|\mathbf{X}_M(k) - \mathcal{T}_M(\mathbf{H}_N)A_{M+N-1}(k)\|^2} + \frac{-1}{2\sigma_a^2} \|A_{M+N-1}(k)\|^2 . \quad (7.196)$$

The maximization of the ML function in (7.196) leads to the following problem:

$$\mathbf{H}_{N \times A_{M+N-1}(k)} \min \left\| \begin{bmatrix} \mathbf{X}_M(k) \\ \mathbf{0}_{N+M-1} \end{bmatrix} - \begin{bmatrix} \mathcal{T}_M(\mathbf{H}_N) \\ \frac{\sigma_v}{\sigma_a} I_{N+M-1} \end{bmatrix} A_{M+N-1}(k) \right\|_2^2 . \quad (7.197)$$

Under this formulation, the problem (7.197) has the same linear-separable form as the DML problem presented before. The Fisher information matrix has the form

$$\tilde{J}(b, a) = \frac{1}{\sigma_v^2} \begin{bmatrix} \tilde{H}^H \tilde{H} & \tilde{H}^H \tilde{K} \\ \tilde{K}^H \tilde{H} & \tilde{K}^H \tilde{K} \end{bmatrix} , \quad (7.198)$$

where the elements of the matrices \tilde{H} and \tilde{K} are now defined as

$$\begin{aligned} \tilde{H} &= [\tilde{h}_i] & \tilde{h}_i &= \frac{\partial \tilde{\mathbf{X}}_M}{\partial b_i} \\ \tilde{K} &= [\tilde{k}_i] & \tilde{k}_i &= \frac{\partial \tilde{\mathbf{X}}_M}{\partial a_i} = \frac{\partial \tilde{H}}{\partial a_i} b \end{aligned} \quad (7.199)$$

and

$$\widetilde{\mathbf{X}}'_M = \begin{bmatrix} \mathbf{X}'_M \\ \mathbf{0}_{N+M-1} \end{bmatrix}, \quad (7.200)$$

(\mathbf{X}'_M is defined in (7.178)). It turns out that

$$\widetilde{K} = \begin{bmatrix} K \\ \mathbf{0}_{N+M-1} \end{bmatrix}, \quad (7.201)$$

(K is defined in (7.182)).

Therefore the Cramer-Rao bound $CRB_{\widetilde{\mathbf{H}}_N}$ for the channel is given as

$$C(\widehat{\mathbf{H}}_N) = \sigma_v^2 \left[\widetilde{K}_{M,N}^H(k) P_{\widetilde{T}_M(\mathbf{H}_N)}^\perp \widetilde{K}_{M,N}(k) \right]^{-1}, \quad (7.202)$$

where

$$\widetilde{T}_M(\mathbf{H}_N) = \begin{bmatrix} T_M(\mathbf{H}_N) \\ \frac{\sigma_v}{\sigma_a} I_{N+M-1} \end{bmatrix}. \quad (7.203)$$

Due to the modification w.r.t. the criterion (7.165), there is no longer a singularity problem related to the Cramer-Rao bound. Moreover, as computer simulations have shown, the quality of the estimation is good even for channels whose impulse response tapers off near the ends, conversely to the criterion (7.165).

Example

The following simulation result shows the increased quality in estimation of the criterion (7.197). We consider two channels ($m=2$) of 6 coefficients each:

$$h_1 = [0.3651 \ 0.5983 \ -0.0825 \ 0 \ 0 \ 0] \\ h_2 = [0.4076 \ -0.0625 \ 0 \ 0 \ 0] .$$

The following table shows the mean CR bound for the estimation of a channel coefficient, using either the criterion (7.197), the criterion (7.165) based on 140 data samples or a training sequence (non-blind estimation) of 26 data samples (a typical situation for the European GSM cellular phone) . The SNR is 10 dB.

DML	GDML	TRS
99.9062	0.0149	0.0198

Notice the very good quality in estimation of the criterion (7.197), especially as compared to the very bad quality provided by (7.165). This shows clearly the improvement in estimation quality achieved with the GDML method.

7.10 Further discussion

Some of the results presented in this chapter have been also presented in [SP94a], [SP94b], [SP95]. It should be noted that after the appearance of the article of Tong, Xu, and Kailath [TXK94] the area of blind fractionally spaced equalization has been given a lot of attention during the last two years, and remains always a hot topic. Similar results to some of the ones that were presented in this chapter have been independently obtained as well by other researchers, namely: Moulines *et al.* have presented in [MDCM93] the same essentially i/o modeling as in (7.43) and a subspace fitting approach similar to the one presented in section 7.8 (see also [MLM94]). A linear prediction approach similar to the one presented in section 7.5 was presented in [MDC⁺94]. Hua has proposed in [Hua94] a DML approach that results to the same minimization problem as the one presented in section 7.9.1 (but a different algorithm to optimize the criterion is proposed by Hua).

Some comparative studies that show the improvement of the modeling (7.43) w.r.t. the one proposed in [TXK94] may be found in [dMB94], [QH94].

7.11 Appendix 7A: Sylvester matrices

Sylvester matrices are associated with the coefficients of two polynomials. Consider the two polynomials

$$\begin{aligned} A(z) &= a_0 z^{n_a} + a_1 z^{n_a-1} + \cdots + a_{n_a} \\ B(z) &= b_0 z^{n_b} + b_1 z^{n_b-1} + \cdots + b_{n_b}, \end{aligned} \quad (7.204)$$

Then the corresponding Sylvester matrix is a $(\bar{n}_a + \bar{n}_b) \times (\bar{n}_a + \bar{n}_b)$ defined as

$$S(A, B) = \begin{bmatrix} a_0 & \cdots & a_{n_a} & 0 & \cdots & 0 \\ 0 & a_0 & \cdots & a_{n_a} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_0 & \cdots & a_{n_a} \\ b_0 & \cdots & b_{n_b} & 0 & \cdots & 0 \\ 0 & b_0 & \cdots & b_{n_b} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & b_0 & \cdots & b_{n_b} \end{bmatrix}. \quad (7.205)$$

The upper partition of $S(A, B)$ has \bar{n}_b rows and the lower one \bar{n}_a rows. \bar{n}_a, \bar{n}_b must satisfy

$$\min(\bar{n}_a - n_a, \bar{n}_b - n_b) = 0. \quad (7.206)$$

Then the following holds:

Lemma 7A.1: Consider the Sylvester matrix $S(A, B)$. Assume that $A(z)$ and $B(z)$ as defined in (7.204) have exactly k common zeros. Then

$$\text{rank}(S(A, B)) = \bar{n}_a + \bar{n}_b - k. \quad (7.207)$$

Corollary 7A.1: $A(z)$ and $B(z)$ are coprime if and only if $S(A, B)$ is nonsingular.

These results extend also to the case of $A(z)$ being a vector polynomial as follows. Consider the polynomial $A(z)$ and $B(z)$ to be defined now as

$$\begin{aligned} A(z) &= A_0 z^{n_a} + A_1 z^{n_a-1} + \cdots + A_{n_a} \\ B(z) &= b_0 z^{n_b} + b_1 z^{n_b-1} + \cdots + b_{n_b}, \end{aligned} \quad (7.208)$$

where the coefficients A_i are now $1 \times m$ vectors. The generalized Sylvester matrix $\bar{S}(A, B)$ is a $(m\bar{n}_a + \bar{n}_b) \times (m\bar{n}_a + \bar{n}_b)$ matrix defined as

$$\bar{S}(A, B) = \begin{bmatrix} A_0 & \cdots & A_{n_a} & 0 & \cdots & 0 \\ 0 & A_0 & \cdots & A_{n_a} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & A_0 & \cdots & A_{n_a} \\ b_0 I_m & \cdots & b_{n_b} I_m & 0 & \cdots & 0 \\ 0 & b_0 I_m & \cdots & b_{n_b} I_m & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & b_0 I_m & \cdots & b_{n_b} I_m \end{bmatrix}. \quad (7.209)$$

The upper partition of $\bar{S}(A, B)$ has \bar{n}_b and the lower $m\bar{n}_a$ rows. In the case $m = 1$ the matrix $S(A, B)$ is obtained. When $m > 1$ the generalized Sylvester matrix is rectangular with more columns than rows. The following holds for $\bar{S}(A, B)$:

Lemma 7A.2: Consider the generalized Sylvester matrix $\bar{S}(A, B)$. Assume that $A(z)$ and $B(z)$ as defined in (7.208) have exactly k common zeros¹⁵. Then

$$\text{rank}(\bar{S}(A, B)) = m\bar{n}_a + \bar{n}_b - k. \quad (7.210)$$

Corollary 7A.2: $A(z)$ and $B(z)$ are coprime if and only if $\bar{S}(A, B)$ has full rank equal to $m\bar{n}_a + \bar{n}_b$.

Other generalizations that lead to rectangular Sylvester matrices exist, the common property of all being that their rank decreases with the number of zeros in common for the corresponding polynomials. Such a generalized Sylvester matrix is the matrix $\mathcal{T}_L(\mathbf{H}_N)$ defined in section 7.4.

¹⁵A complex number \bar{z} is said to be a zero of the vector polynomial $A(z)$ as defined in (7.208) if $A(\bar{z}) = 0$

Chapter 8

Applying existing blind techniques to the multichannel setup

THE focus of this chapter is on equalization rather than channel identification. We apply the blind equalization techniques that were presented in the first part of the thesis to the fractionally-spaced setup presented in chapter 7. The general methodology that allows to apply these techniques to the fractionally spaced case is presented and the performance of the algorithms in the fractionally-spaced case is demonstrated through computer simulations.

8.1 Introduction

In the first part of this thesis we presented several BE methods (mostly of the CMA type) that improved the performance of the classical CMA in several aspects. Even though in the presentation of these algorithms we used a Baud-rate formulation, these algorithms can be equally well adapted so as to be applied to the case of fractionally-spaced receivers. The advantages of fractional spacing that were discussed in the previous chapter are expected to reflect to the performance of these algorithms. Moreover, it has been recently stated [LD94], [May94], [LFHJ95], that the classical CMA 2-2 itself has an essentially increased performance when combined with fractional spacing: under certain conditions it is globally convergent. In this chapter we will discuss the adaptation of the algorithm of the first part of the thesis to the fractionally-spaced setup and will show some computer simulation results that correspond to multipath transmission.

8.2 Formulation

8.2.1 CMA - like and DD algorithms

It is straightforward to apply the algorithms of this kind to the setup of a fractionally-spaced receiver. Namely, if one defines the regressor at each time instant X_k and the corresponding equalizer as

$$X_k = \begin{bmatrix} x_k^{(1)} \\ \dots \\ x_k^{(m)} \\ \dots \\ x_{k-N+1}^{(1)} \\ \dots \\ x_{k-N+1}^{(m)} \end{bmatrix}^* ; \quad W_k = \begin{bmatrix} w_0^{(1)}(k) \\ \vdots \\ w_0^{(m)}(k) \\ \vdots \\ w_0^{(1)}(k-N+1) \\ \vdots \\ w_0^{(m)}(k-N+1) \end{bmatrix}, \quad (8.1)$$

then the corresponding equalizer output at time instant k will equal

$$y_k = X_k^H W_k.$$

The corresponding fractionally-spaced BE algorithms will have the following general form

$$W_{k+1} = W_k + \mu T_k P_k^{-1} e_k. \quad (8.2)$$

The essential difference with respect to the Baud-rate case is that in (8.2) the regression vector is not stationary in the scalar but in the vector sense. This implies that it should be shifted by m (instead of 1) positions at each iteration. The following table shows how different choices for the quantities in (8.2) lead to different algorithms.

- CMA p-q: $T_k = X_k$, $P_k = \mathbf{1}$, $e_k = y_k |y_k|^{p-2} (r_p - |y_k|^p)$
- NCMA: $T_k = X_k$, $P_k = (X_k^H X_k)$, $e_k = \text{sign}(y_k) - y_k$
- NSWCMA: $T_k = \mathbf{X}_k$, $P_k = \mathbf{X}_k^H \mathbf{X}_k$, $e_k = \text{sign}(\mathbf{X}_k^H W_k) - \mathbf{X}_k^H W_k$, $\mathbf{X}_k = [X_k \dots X_{k-L+1}]$
- NSWERCMA: $\begin{cases} T_k = \mathbf{X}_k, P_k = \lambda P_{k-1} + X_k X_k^H, e_k = \text{sign}(\mathbf{X}_k^H W_k) - \mathbf{X}_k^H W_k, \\ \mathbf{X}_k = [X_k \dots X_{k-L+1}], X_k = [x_k \dots x_{k-L+1}]^T \end{cases}$
- RLSCMA: $T_k = X_k$, $P_k = \lambda P_{k-1} + X_k X_k^H$, $e_k = \text{sign}(y_k) - y_k$
- DDA: $T_k = X_k$, $P_k = \mathbf{1}$, $e_k = \text{dec}(y_k) - y_k$
- NSWDDA: $T_k = \mathbf{X}_k$, $P_k = \mathbf{X}_k^H \mathbf{X}_k$, $e_k = \text{dec}(\mathbf{X}_k^H W_k) - \mathbf{X}_k^H W_k$, $\mathbf{X}_k = [X_k \dots X_{k-L+1}]$
- NSWERDDA: $\begin{cases} T_k = \mathbf{X}_k, P_k = \lambda P_{k-1} + X_k X_k^H, e_k = \text{dec}(\mathbf{X}_k^H W_k) - \mathbf{X}_k^H W_k, \\ \mathbf{X}_k = [X_k \dots X_{k-L+1}], X_k = [x_k \dots x_{k-L+1}]^T \end{cases}$
- RLSDDA: $T_k = X_k$, $P_k = \lambda P_{k-1} + X_k X_k^H$, $e_k = \text{dec}(y_k) - y_k$

- CMDDHA: $T_k = X_k$, $P_k = 1$, $e_k = cdec(y_k) - y_k$
- NSWCMDDHA: $T_k = \mathbf{X}_k$, $P_k = \mathbf{X}_k^H \mathbf{X}_k$, $e_k = cdec(\mathbf{X}_k^H W_k) - \mathbf{X}_k^H W_k$, $\mathbf{X}_k = [X_k \cdots X_{k-L+1}]$
- NSWERCMDHA: $\begin{cases} T_k = \mathbf{X}_k, P_k = \lambda P_{k-1} + X_k X_k^H, e_k = cdec(\mathbf{X}_k^H W_k) - \mathbf{X}_k^H W_k, \\ \mathbf{X}_k = [X_k \cdots X_{k-L+1}], X_k = [x_k \cdots x_{k-L+1}]^T \end{cases}$
- RLSCMDHA: $T_k = X_k$, $P_k = \lambda P_{k-1} + X_k X_k^H$, $e_k = cdec(y_k) - y_k$

These algorithms are expected to combine two things: the performance characteristics of their Baud-rate counterparts and the features of linear fractionally-spaced equalizers. Namely, the ability of opening a closed channel eye is expected to be preserved: all the above classes of algorithms (except for the DDA, NSWDDA, NEWDDA, RLSDDA) should open a closed channel eye for any QAM constellation, whereas the decision-directed ones should open a closed channel eye for CM constellations. The convergence speed is also expected to be increasing with the number of constraints L imposed by the criterion of each algorithm at each iteration. Moreover, these algorithms will benefit from the structure of the fractionally-spaced setup, namely:

- They will be able of providing FIR ZF equalizers in the absence of noise.
- They will be able of dealing with channels that have zeros on the unit circle -as long as these zeros are not shared by all the sub-channels-.
- They will be able of providing acceptable equalizer settings even for channels that have zeros in common - as long as these zeros are not on the unit circle-. In this case they will not exist of course FIR ZF equalizers, however the performance will increase with the equalizer length.

The fact that these techniques will work even when the no-zeros-in-common condition is violated is due to the fact that, as they are of the Busgang-type (they make implicit use of HOS), they do not depend only on second order statistics for their implicit channel identification.

Moreover, it has been stated recently [LD94], [May94], [LFHJ95], that the CMA 2-2 algorithm is globally convergent in the fractionally-spaced setup under the following conditions:

- The FIR assumption (mentioned in chapter 7) should be satisfied
- The sub-channels must have no zeros in common
- No additive noise should be present

In the presence of noise, the FS-CMA 2-2 will still have a unique solution into each cone (described in [LD94]), however all these solutions will not be equivalent between them. However, the result stated above promises an improved performance of the CMA in the fractionally-spaced case, and we expect this to be reflected to the other algorithms presented above as well.

It should be noted at this point, that applying a reasoning similar to the one in [LD94], the characteristics of the DD algorithm that we analyzed in chapter 5, can be reflected to the FS case: when the input signal is CM, there are very few sets of local minima, which vanish as the number of points on the circle increases. This fact promises an “almost-sure” global convergence of the DDA (and of its normalized counterparts) in the fractionally-spaced case when the transmitted signal is CM (and if there is no ambiguity about the transmitted frequency). Combined with the fact that the DDA will provide in general a better steady-state error than the CMA, these remarks predict a very promising behaviour of the FS-DDA when used with CM constellations.

8.2.2 The bilinear case

We have also adapted the bilinear approach of chapter 6 to the framework of fractionally-spaced equalization (the application is straightforward if one takes into account cyclostationarity as in (8.1)). The motivation for this was the following: in contrast to the Baud-rate case, in the fractionally-space one, FIR ZF equalizers exist. This means that if the channel length is given, then the minimal length of the FS equalizer in order to be ZF is known. Therefore knowledge of the channel length may help avoiding over-parameterization in the FS case, whereas knowledge of the channel *inverse* length is needed to deal with over-parameterization in the Baud-rate case. This gives a potential advantage to the FS setup, since the length of a channel can be more easily known or estimated than the length of the channel inverse impulse response.

It turns out that in the FS setup, there are two mechanisms that may result to a singularity in the bilinear method:

- The excessive equalizer length (as in the Baud-rate case)
- The existence of several ZF equalizers for a given equalizer length that correspond to different position of the non-zero entry of the overall channel-equalizer impulse response

As a result of these two factors, the covariance matrix \mathbf{R} appearing in the solution of the bilinear cost function will be singular (in the absence of noise), even when the equalizer length $N = \underline{N}^1$ (in which case it is -strictly speaking- perfectly parameterized). However, the problem is nonsingular for all equalizer lengths $N < \underline{N}$, the best choice being of course $N = \underline{N} - 1$. This will regulate the problem and will provide a unique solution at the cost of an introduced sub-optimality. However, as it will be shown in computer simulations, this

¹Note that, contrarily to chapter 7, in this chapter N (instead of L) is the equalizer length. This was done in order to avoid confusion with L : the number of constraints imposed by the algorithms.

suboptimality is often not very important, especially in cases where the channel coefficients decay towards the tails.

8.3 Simulations

At this point we provide some computer simulation results that support the above claims. These results (which are by no means exhaustive) are indicative of the behaviour of some of the above algorithms and add to the value of the methods proposed in the first part of the thesis, since they show their very good performance when combined with fractional spacing.

8.3.1 NSWCMA

We have simulated the NSWCMA for the case of the mobile multipath channel given in [SGKC91]. The channel output has been sampled at the rate $2/T$, has 27 coefficients and is considered to be a typical mobile propagation channel. Figure 8.1 shows the zeros of the two subchannels (note that there exist indeed some pairs of zeros in common!). We have simulated

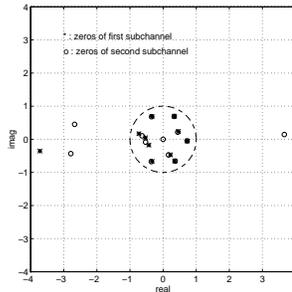


Figure 8.1: The zeros of the two subchannels

the CMA and the NSWCMA algorithms for the transmission of a 4-QAM sequence, and an output SNR of 30 dB. The equalizer used had 33 coefficients and was initialized with a unique non-zero center tap equal to 1. Figure 8.2 shows the evolution of the closed-eye measure for the CMA and two members of the NSWCMA algorithms, used with the following parameters:

	$\bar{\mu}$	L	N
a	0.3	4	33
b	1	1	33
c	$\mu=0.05$	-	33

Note how all the algorithms open fast the channel eye and how the convergence speed increases drastically with the number of CM constraints L . This verifies the good performance

of CMA-like algorithms in the FS setup, and moreover shows the considerable increase in convergence speed that can be achieved by the NSWCMA. Note that the good behaviour is observed despite the fact that the two sub-channels have zeros in common.

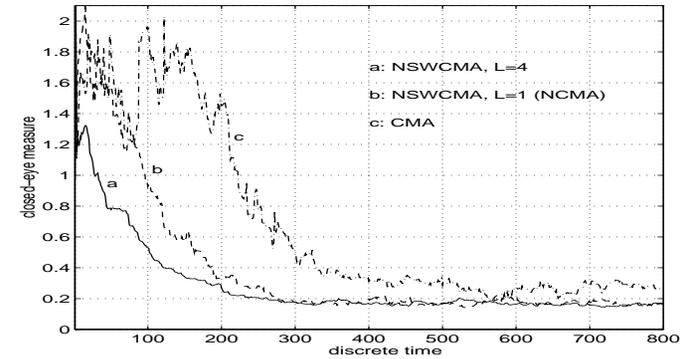


Figure 8.2: Comparative simulations of the NSWCMA and the CMA for a $T/2$ fractionally-spaced channel

8.3.2 NSWERCMA

A further increase in convergence speed (by keeping also down the steady-state error) can be provided by the NSWERCMA algorithm. This can be shown in figure 8.3, which shows the evolution of the closed-eye measure for the NSWERCMA, the NSWCMA and the CMA algorithm. The transmitted constellation is again 4-QAM, and we use the same channel and SNR as before. The parameters used are shown in the next table.

	$\bar{\mu}$	L	λ
a	0.2	15	0.99
b	0.5	5	-
c	1	1	-
d	$\mu=0.02$	-	-

Notice the very fast convergence of the NSWERCMA algorithm as compared to its other counterparts. This is due to the very bog value (15) given to L .

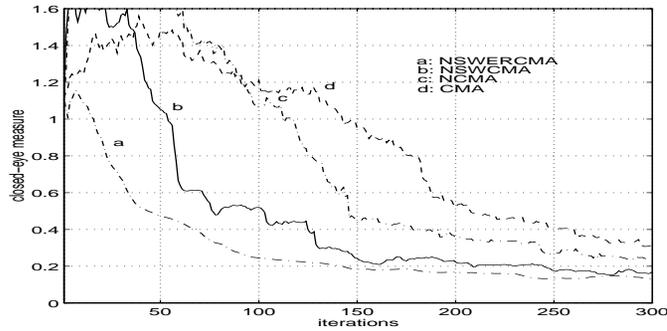


Figure 8.3: Comparative simulations of the NSWERCMA, the NSWCMA and the CMA for a $T/2$ fractionally-spaced channel

8.3.3 Bilinear method

We tested the bilinear method of chapter 6 applied to the fractionally spaced setup through the following simulation: we consider $T/2$ sampling and two channels with impulse responses

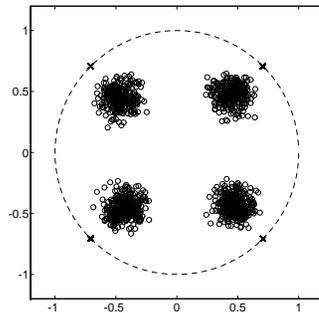


Figure 8.4: Simulating the bilinear method for the $T/2$ fractionally spaced setup

$$h_1 = [0.311 \ 0.637 \ 0.965 \ 0.069 \ 0.383 \ 0.367]$$

$$h_2 = [0.657 \ 0.151 \ 0.755 \ 0.160 \ 0.9680 \ 0.239]$$

We considered a fractionally-spaced equalizer of length 9 (one less than what strictly needed for FIR ZF equalization in the absence of noise), which was initialized at an all-zero setting.

The transmitted constellation is 4-QAM. Figure 8.4 shows the constellation after 1000 iterations of the algorithm. The dark points correspond to zero-noise, whereas the circles to an SNR of 20 dB

8.4 Conclusions and further discussion

Similar results have been obtained for a number of other tests (including DD-like algorithms in the context of a GSM channel). In all cases, the expected behaviour was verified: the combination of the techniques developed in the first part of this thesis with the advantages of fractionally-spaced equalization provides some very well performing methods for blind equalization.

APPENDICES

Appendix A

Sommaire détaillé en français

CET appendice contient un sommaire détaillé en français du travail effectué dans cette thèse.

A.1 L' état de l'art en égalisation aveugle

A.1.1 Le principe d'égalisation linéaire

L'égalisation est une procédure qui est utilisée par les récepteurs des systèmes de communications numériques afin de réduire l'effet d'Interférence Entre Symboles (IES), due à la propagation du signal modulé à travers un canal linéaire. La figure A.1 montre le schéma classique d'égalisation linéaire adaptative:

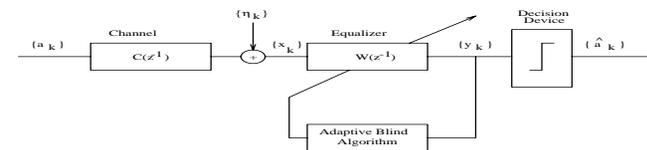


Figure A.1: Un schéma classique d'égalisation aveugle

Dans cette figure, $\{a_k\}$ représente la séquence discrète des données émises, $\{n_k\}$ les échantillons du bruit additif à la sortie du canal, $\{x_k\}$ les échantillons bruités à la sortie du canal, $\{y_k\}$ la séquence discrète à la sortie de l'égaliseur et $\{\hat{a}_k\}$ la séquence des données estimées à la sortie de l'élément de décision. Le canal $C(z)$ est supposé être linéaire, il peut donc être paramétrisé par sa réponse impulsionnelle, que l'on suppose être finie (ayant une Réponse

Impulsionnelle Finie: RIF) de longueur L :

$$C = [c_0 \cdots c_{L-1}]^T . \quad (\text{A.1})$$

La sortie discrète du canal (qui inclut aussi le bruit additif) peut être écrite sous la forme

$$x_k = A_k^H C + \eta_k , \quad (\text{A.2})$$

où A_k est un vecteur de N échantillons émis consécutivement:

$$A_k^H = [a_k \cdots a_{k-L+1}] . \quad (\text{A.3})$$

L'égaliseur est aussi un filtre linéaire à RIF qui a N coefficients. La sortie de l'égaliseur peut donc être écrite sous la forme

$$y_k = X_k^H W_k , \quad (\text{A.4})$$

où

$$X_k^H = [x_k \cdots x_{k-N+1}] , \quad (\text{A.5})$$

et

$$W_k = [w_0(k) \cdots w_{N-1}(k)]^T . \quad (\text{A.6})$$

X_k est le vecteur dit *régresseur* et W_k est l'égaliseur à l'instant k . Le principe de l'égalisation linéaire adaptative est le suivant: la sortie de l'égaliseur à chaque instant k , y_k , est utilisée par un algorithme récursif qui adapte les coefficients de l'égaliseur selon un critère inhérent d'optimisation. Après plusieurs itérations, l'algorithme doit converger vers un régime stationnaire où les échantillons à la sortie de l'égaliseur s'approchent de la séquence de données émises. Dans ce cas, l'oeil du canal est dit "ouvert" et l'élément de décision fournit à sa sortie les symboles émis avec une petite probabilité d'erreur.

Typiquement, une séquence dite "d'apprentissage" est utilisée au début de la transmission des données afin de permettre à l'égaliseur de converger vers une solution qui ouvre l'oeil du canal. Cette séquence est connue d'avance par le récepteur, elle lui permet alors de créer un signal d'erreur

$$\epsilon_k = a_k - y_k ,$$

qui pourra ensuite être utilisé par un algorithme adaptatif standard afin d'aider l'égaliseur à converger. Dans le cas de l'algorithme LMS, l'équation récursive de l'adaptation de l'égaliseur sera donnée par

$$W_{k+1} = W_k + \mu X_k (a_k - X_k^H W_k) , \quad (\text{A.7})$$

où μ représente le pas d'adaptation. Cet algorithme est un algorithme à gradient stochastique pour le critère de minimisation suivant

$$\min_W J(W) = E(|y_k - a_k|^2) = E(|X_k^H W - a_k|^2) . \quad (\text{A.8})$$

Ce critère correspond à une fonction de coût convexe qui a un seul minimum. Si le pas d'adaptation μ est choisi de façon à ce que l'algorithme soit stable, il va alors converger vers le minimum unique de sa fonction de coût (A.8). Donc si la séquence d'apprentissage comprend un nombre d'échantillons suffisant, l'algorithme va converger vers une solution qui correspond à un oeil de canal ouvert.

Comme, après l'ouverture de l'oeil, les décisions $\{\hat{a}_k\}$ approximent assez bien la séquence émise $\{a_k\}$, ils peuvent être utilisés à leur place pour l'étape finale de l'algorithme. L'adaptation de l'égaliseur dans ce cas suit l'équation suivante

$$\begin{cases} \epsilon_k &= \hat{a}_k - X_k^H W_k \\ W_{k+1} &= W_k + \mu X_k \epsilon_k \end{cases} . \quad (\text{A.9})$$

Cet algorithme s'appelle "L'algorithme Dirigé par les Décisions (DD)" et est utilisé en pratique pour améliorer l'ouverture de l'oeil pendant la phase finale de la convergence de l'égaliseur.

A.1.2 Egalisation aveugle

Dans certains cas, cette procédure standard d'égalisation à l'aide d'une séquence d'apprentissage n'est pas convenable pour des raisons qui sont liées soit à la réduction du débit d'information à cause de cette séquence, soit à l'applicabilité de la méthode (par exemple, dans les réseaux locaux la déconnection d'un utilisateur nécessiterait la retransmission de la séquence d'apprentissage à plusieurs utilisateurs). Il serait alors souhaitable d'essayer d'égaliser la séquence d'information transmise en ayant seulement accès à la séquence des données reçues $\{x_k\}$, et non pas aux $\{a_k\}$. Ceci est le principe d'*égalisation aveugle* (dite aussi "*égalisation autodidacte*"), qui se base seulement sur les données reçues (et sur quelques informations *a priori* sur les statistiques de $\{a_k\}$) pour égaliser le canal. Dans certains cas où on souhaite l'identification des coefficients du canal au lieu de l'égalisation, on parle d'*identification de canaux aveugle*.

Résultats fondamentaux

Avant de présenter les classes d'algorithmes d'égalisation aveugle qui nous intéressent, on va citer quelques résultats fondamentaux sur l'égalisation et identification de canal aveugles.

Condition nécessaire pour l'identification d'un canal linéaire à partir de sa sortie:

L'identification d'un canal linéaire à partir de sa sortie en utilisant les statistiques du second ordre est seulement possible pour un canal à phase minimale. Dans les autres cas, l'exploitation des moments d'ordre supérieur à 2 de la sortie du canal est nécessaire pour son identification.

Ce fait peut s'expliquer de la façon suivante: si $S_{aa}(\omega)$ et $S_{yy}(\omega)$ sont respectivement les densités spectrales de l'entrée et de la sortie du canal et $C(\omega)$, la réponse fréquentielle du canal, nous avons la relation d'entrée-sortie suivante

$$S_{yy}(\omega) = |C(\omega)|^2 S_{aa}(\omega) , \quad (\text{A.10})$$

ce qui veut dire qu'en mesurant la densité spectrale à la sortie du canal (et en sachant la densité spectrale de son entrée) on peut seulement identifier le module de la réponse fréquentielle du canal: l'information sur la phase est perdue! Seulement dans le cas où il est connu d'avance que le canal est à minimum de phase (où à maximum de phase) il est possible de trouver aussi la phase correspondante à chaque fréquence. Ce fait explique pourquoi l'identification aveugle à l'aide de la densité spectrale (ou des moments de deuxième ordre de la sortie) est impossible, ce qui nécessite l'utilisation des statistiques d'ordre supérieur pour l'identification ou l'égalisation aveugle. Une conséquence immédiate de cette condition est la limitation suivante:

Limitation 1: impossibilité d'identification aveugle quand le signal d'entrée est Gaussien:

Un canal linéaire ne peut pas être identifié à partir de sa sortie si son entrée est un processus Gaussien.

Cette exclusion des entrées Gaussiennes est due au fait que pour les processus Gaussiens toute leur information statistique se concentre dans leurs moments d'ordre 1 et 2: il n'y a donc pas d'information sur les statistiques d'ordre supérieur qui puisse être exploitée pour l'identification du canal dans ce cas. Deux autres limitations majeures pour l'identification de canal aveugle sont:

Limitation 2: identification aveugle à une constante près:

La réponse fréquentielle (ou impulsionnelle) d'un canal ne peut être identifiée qu'à un scalaire complexe unitaire près.

Ceci dit que si la réponse fréquentielle du canal est $C(\omega)$, la réponse identifiée sera de la forme

$$C'(\omega) = C(\omega)e^{j\theta} , \quad (\text{A.11})$$

où θ sera une phase indéterminée. Ce phénomène arrive toujours en égalisation aveugle et a souvent comme résultat la *rotation* de la constellation émise. Cet effet peut être enlevé à l'aide du *codage différentiel* appliqué sur le signal émis (au détriment de la réduction des performances du système de quelques dB).

Limitation 3: identification à un décalage temporel près:

La séquence émise peut être identifiée seulement à un décalage temporel près.

Ceci est dû au fait qu'un canal à Non Minimum de Phase (NMP) aura un inverse qui sera non-causal. Dans ce cas, l'égalisation a pour but d'identifier le canal "inverse" qui satisfait:

$$C^{-1}(z^{-1})C(z^{-1}) = z^{-l} , \quad (\text{A.12})$$

où l est un entier constant.

À part ces limitations, quelques conditions sur l'identification aveugle d'un système sont les suivantes:

Condition suffisante pour l'identification d'un canal NMP par sa sortie:

Un canal linéaire sera parfaitement égalisé si les fonctions de densité de probabilité de tous les ordres de la séquence reçue sont égales aux fonctions correspondantes de la séquence émise.

L'importance de ce résultat (paru dans [BGR80]) se situe dans le fait que les fonctions de densité de probabilité *conjointes* de l'entrée et de la sortie ne sont pas nécessaires. Néanmoins, cela ne permet pas la construction d'une méthode simple d'égalisation aveugle, car cela nécessiterait l'égalisation de tous les moments de la sortie avec les moments correspondants de l'entrée. Un résultat beaucoup plus fort qui est apparu dans [SW90] est le suivant:

Condition nécessaire et suffisante pour l'identification d'un canal NMP par sa sortie:

Un canal linéaire sera parfaitement égalisé si et seulement si les deux conditions suivantes sont satisfaites:

$$\begin{cases} E(|y|^2) = E(|a|^2) \\ |K(y)| = |K(a)| \end{cases} , \quad (\text{A.13})$$

où $K(\cdot)$ est la *Kurtosis* d'un processus défini comme:

$$K(z_i) = E(|z_i|^4) - 2E^2(|z_i|^2) - |E(z_i^2)|^2 . \quad (\text{A.14})$$

L'importance de cette condition réside dans le fait que deux quantités statistiques seulement sont nécessaires pour l'égalisation parfaite du canal.

A.1.3 Algorithmes de type "Bussgang" pour l'égalisation aveugle

Le schéma général des algorithmes Bussgang pour l'égalisation aveugle est montré dans la figure A.2. Le principe des algorithmes de Bussgang est le suivant: une fonction non linéaire

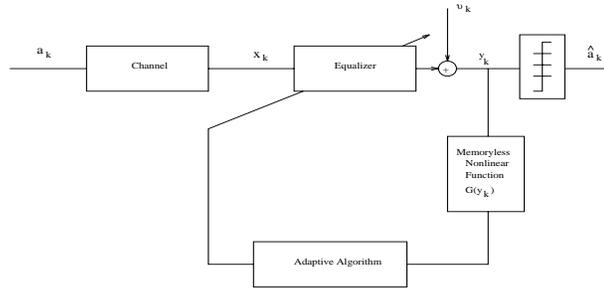


Figure A.2: Le principe d'égalisation de Bussgang

G est appliquée à la sortie du canal afin de fournir la fonction d'erreur qui sera ensuite utilisée par l'algorithme aveugle pour adapter l'égaliseur. Cette fonction non linéaire est utilisée pour fournir l'information statistique d'ordre supérieur qui permettra l'égalisation (ou l'identification implicite) du canal. Quelques algorithmes classiques de Bussgang pour l'égalisation aveugle sont les suivants:

- L'algorithme de Sato

Cet algorithme a été proposé par Y. Sato en 1975 [Sat75] et a été conçu dans sa forme originale pour des constellations MDP (Modulation De Phase). La fonction de coût à minimiser est la suivante:

$$\min_W J^{Sato}(W) = E((y - \gamma \text{sign}(y))^2) , \quad (\text{A.15})$$

où la fonction $\text{sign}(\cdot)$ est défini comme ci-dessous pour un argument réel:

$$\text{sign}(r) = \frac{r}{|r|} = \begin{cases} 1, & r > 0 \\ -1, & r < 0 \end{cases} , \quad (\text{A.16})$$

et γ est un facteur défini comme

$$\gamma = \frac{E a_k^2}{E |a_k|} . \quad (\text{A.17})$$

L'idée de Sato était la suivante: ayant remarqué que l'algorithme dirigé par les décisions ne convergait pas vers une solution acceptable quand le signal modulé avait plus de deux niveaux, il a utilisé une constellation *réduite* à deux niveaux pour prendre ces décisions: l'échantillon reçu est seulement classé comme positif ou négatif. Cette intuition a en effet mené à un algorithme qui réussissait en pratique à ouvrir un oeil de canal initialement fermé. L'algorithme de Sato (qui est un algorithme de gradient stochastique pour le critère (A.15)) est donné par

$$W_{k+1} = W_k + \mu X_k (\gamma \text{sign}(y_k) - y_k) . \quad (\text{A.18})$$

Cet algorithme est applicable seulement au cas des signaux réels. Son extension au cas complexe est l'algorithme de Sato généralisé:

- L'algorithme de Sato Généralisé (GSA)

Cet algorithme est décrit par l'équation suivante

$$W_{k+1} = W_k + \mu X_k (\gamma \text{csign}(y_k) - y_k) , \quad (\text{A.19})$$

où la fonction $\text{csign}(\cdot)$ d'un scalaire complexe $r = r_R + jr_I$ se définit comme

$$\text{csign}(r_R + jr_I) = \text{sign}(r_R) + j \text{sign}(r_I) . \quad (\text{A.20})$$

Cet algorithme est une extension de l'algorithme de Sato dans le cas complexe. Nous verrons par la suite qu'une meilleure extension est possible.

- Les algorithmes de Godard

Ces algorithmes ont été proposés par D. Godard en 1980 [God80]. Le critère minimisé par ces algorithmes est le suivant:

$$\min_W J_p^{God}(W) = \frac{1}{2p} E (|y|^p - r_p)^2 , \quad p = 1, 2, \dots , \quad (\text{A.21})$$

où r_p est un scalaire appelé *constante de dispersion* et défini comme suit

$$r_p = \frac{E |a_k|^{2p}}{E |a_k|^p} . \quad (\text{A.22})$$

Le principe de ce critère est qu'il pénalise les déviations du *module* du signal reçu par rapport à une constante. Les algorithmes résultants sont de la forme

$$W_{k+1} = W_k + \mu X_k y_k |y_k|^{p-2} (r_p - |y_k|^p) . \quad (\text{A.23})$$

Les deux premiers membres de cette classe d'algorithmes sont aussi appelés "les algorithmes à module constant" (CMA), et ont été aussi proposés par J. Treichler et B. Agee en 1983 [TA83]. L'algorithme de Godard correspondant à $p = 1$ est le CMA 1-2:

$$W_{k+1} = W_k + \mu X_k (r_1 \frac{y_k}{|y_k|} - y_k) , \quad (\text{A.24})$$

tandis que le choix $p = 2$ correspond au CMA 2-2:

$$W_{k+1} = W_k + \mu X_k y_k (r_2 - |y_k|^2) . \quad (\text{A.25})$$

Ces algorithmes sont bien connus pour leur bonne performance: ils arrivent en général à ouvrir l'oeil du canal, ils sont insensibles au décalage fréquentiel de la porteuse et ont un meilleur comportement que d'autres algorithmes en régime permanent. La performance de ces algorithmes a été analysée par plusieurs chercheurs, le résultat dominant étant présenté par Shalvi et Weinstein en 1990 [SW90]. Ce résultat est le suivant:

Supposons que les hypothèses suivantes soient satisfaites:

- Il n'y a pas de bruit additif
- Le signal émis suit une distribution *sous-Gaussienne*¹
- L'égaliseur est de longueur infini

Dans ce cas l'algorithme CMA 2-2 a un seul minimum dans sa fonction de coût, qui est le minimum optimal. Il en résulte que l'algorithme sera toujours capable d'ouvrir parfaitement l'oeil du canal quand ces conditions sont satisfaites. Ce résultat est très fort et montre pourquoi l'algorithme CMA 2-2 est capable de converger même quand le signal émis n'est pas à module constant (il suffit qu'il soit sous-Gaussien).

Sous des conditions réelles quand même, tous les algorithmes de Godard ont le problème suivant: leur fonction de coût n'est pas convexe. Il n'y a donc pas uniquement un minimum global mais également plusieurs *minima locaux* sous optimaux. Ces points d'attraction peuvent être atteints par l'algorithme; dans ce cas on parle du problème de "ill-convergence": l'algorithme converge vers une solution qui n'ouvre pas suffisamment l'oeil du canal. Les points stationnaires de l'algorithme sont trouvés en mettant à zero la dérivée de la fonction de coût, ce qui donne pour l'algorithme de Godard

$$E \left(X_k y_k |y_k|^{p-2} (|y_k|^p - r_p) \right) = 0 \quad . \quad (\text{A.26})$$

Cette équation représente un système non linéaire de N équations à N paramètres, elle a donc plusieurs solutions, parmi lesquelles quelques-unes sont des minima locaux non-désirables de la fonction de coût.

A.1.4 Objectifs de la thèse

Les objectifs de cette thèse sont liés aux problèmes des méthodes de Bussgang pour l'identification aveugle. Une liste de ces problèmes est la suivante:

- La vitesse de convergence des algorithmes de Bussgang
- Le problème des minima locaux
- La dépendence de leur pas d'adaptation par la couleur du signal reçu
- L'impossibilité d'égalisation quand le canal a des zéros sur le cercle unité
- L'impossibilité d'égalisation parfaite avec des égaliseurs à RIF
- L'impossibilité d'égalisation aveugle quand le signal émis est Gaussien
- L'impossibilité de l'algorithme DD d'ouvrir un oeil initialement fermé

¹Un processus $\{y\}$ est dit sous-Gaussien si et seulement si sa kurtosis est négative: $K(y) < 0$.

- Le besoin d'utiliser des statistiques d'ordre supérieur pour l'identification ou l'égalisation de canal aveugle

Les méthodes qui seront présentées par la suite ont été motivées par les problèmes indiqués ci-dessus.

A.2 Egalisation et identification aveugle non sur-échantillonné

A.2.1 L'algorithme NSWCMA

Une solution qui pourrait améliorer la performance des algorithmes aveugles de type Bussgang est la *normalisation*. Par normalisation on entend la modification d'un algorithme adaptatif de façon à ce que son pas d'adaptation puisse varier dans un intervalle connu d'avance tout en garantissant l'opération stable de l'algorithme. Un exemple typique d'un algorithme normalisé est l'algorithme NLMS (Normalized LMS), qui est donné par

$$W_{k+1} = W_k + \frac{\mu}{\|X_k\|^2} X_k (a_k - X_k^H W_k) \quad . \quad (\text{A.27})$$

Cet algorithme peut être vu comme un algorithme LMS à pas d'adaptation variable:

$$\mu = \frac{\bar{\mu}}{\|X_k\|^2} \quad .$$

Le résultat de cette normalisation est que l'algorithme (A.27) est stable pour toutes les valeurs $\bar{\mu}$ dans la région suivante

$$0 < \bar{\mu} < 2 \quad . \quad (\text{A.28})$$

Il est à noter que cet intervalle est indépendant du signal d'entrée, contrairement au cas du LMS. Cette normalisation améliore la vitesse de convergence de l'algorithme (la plus grande est atteinte pour $\bar{\mu} = 1$). De plus, la normalisation engendre un autre effet positif dans le cas d'égalisation aveugle: il a été déjà remarqué par Mazo en 1980 [Maz80] que l'algorithme DDA arrive à s'échapper plus facilement des minima locaux de sa fonction de coût quand le pas d'adaptation utilisé est grand. Les algorithmes normalisés ont l'avantage que la borne maximale du pas d'adaptation qui garantit la stabilité soit connue: on peut donc utiliser un pas qui est grand pour aider à éviter minima locaux sans provoquer la divergence de l'algorithme.

Outre la normalisation, on veut aussi ajouter de la mémoire dans les algorithmes: une façon d'améliorer encore plus la vitesse de convergence est d'imposer à l'égaliseur plusieurs contraintes qui n'impliquent pas seulement le dernier vecteur de regression, mais également quelques autres vecteurs du passé.

On peut récapituler: on cherche un algorithme normalisé du type Bussgang pour l'égalisation aveugle, qui introduit aussi une mémoire pour améliorer la vitesse de convergence. En choisissant l'algorithme de type "à module constant", on propose le critère suivant:

$$\min_{W_{k+1}} \left\{ \text{sign}(\mathbf{X}_k^H W_k) - \mathbf{X}_k^H W_{k+1} \right\}_{p-1}^2 + \left(\frac{1}{\bar{\mu}} - 1 \right) \|W_{k+1} - W_k\|^2 \quad , \quad (\text{A.29})$$

où $P_k = \mathbf{X}_k^H \mathbf{X}_k$, \mathbf{X}_k est une matrice $N \times L$ définie comme:

$$\mathbf{X}_k = [X_k \ X_{k-1} \ \dots \ X_{k-L+1}] = \begin{bmatrix} x_k^* & x_{k-1}^* & \dots & x_{k-L+1}^* \\ x_{k-1}^* & x_{k-2}^* & \dots & x_{k-L}^* \\ \vdots & \vdots & \ddots & \vdots \\ x_{k-N+1}^* & x_{k-N+2}^* & \dots & x_{k-N-L+2}^* \end{bmatrix}, \quad (\text{A.30})$$

et on définit la fonction $\text{sign}(V)$ où V est un vecteur ($V = [v_0 \ \dots \ v_{K-1}]^T$), comme ci-dessous

$$\text{sign}([v_0 \ \dots \ v_{K-1}]^T) = \left[\frac{v_0}{|v_0|} \ \dots \ \frac{v_{K-1}}{|v_{K-1}|} \right]^T. \quad (\text{A.31})$$

Le critère (A.29) est un critère déterministe qui impose un ensemble de L contraintes de type "module constant" à l'égaliseur W_{k+1} . Le rôle de la matrice de covariance instantanée P_k est d'effectuer un "pré-blanchissement" du signal d'entrée, ce qui doit amener à une convergence plus rapide de l'algorithme résultant. Le rôle du pas d'adaptation $\bar{\mu}$ est de contrôler la déviation de W_{k+1} par rapport à W_k . Dans le cas où $\bar{\mu} = 1$ le critère (A.29) est équivalent à minimiser le premier terme additif sous contrainte de minimisation de la quantité $\|W_{k+1} - W_k\|^2$.

Le critère ci-dessus (A.29) est minimisé exactement par l'algorithme suivant

$$W_{k+1} = W_k + \bar{\mu} \mathbf{X}_k P_k^{-1} (\text{sign}(\mathbf{X}_k^H W_k) - \mathbf{X}_k^H W_k). \quad (\text{A.32})$$

L'équation (A.32) décrit l'algorithme NSWCMA (Normalized Sliding Window Constant Modulus Algorithm). Il s'agit en effet d'une classe d'algorithmes à deux paramètres:

- Le nombre de regresseurs L impliqués à chaque itération
- Le pas d'adaptation $\bar{\mu}$

Il peut être démontré que ces algorithmes sont stables pour toute valeur de $\bar{\mu}$ dans la région $(0, 2)$ (comme en (A.28)). Ces algorithmes de type "à module constant", sont normalisés, et utilisent une fenêtre glissante pour les données. Leur complexité peut être amenée à $2N + 20L$ multiplications/itération (voir le chapitre 3 de la thèse).

L'analyse théorique des algorithmes NSWCMA mène à trois conclusions principales:

- Leur vitesse augmente avec L
- Des grandes valeurs de $\bar{\mu}$ peuvent aider à s'échapper des minima locaux
- Leur performance en présence de bruit peut décroître quand L s'approche de la longueur N de l'égaliseur

Cette dernière remarque indique aussi un conseil pratique pour l'utilisation de l'algorithme: L doit être choisi clairement plus petit que N .

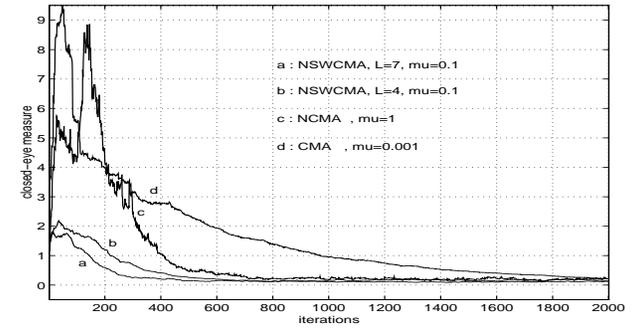


Figure A.3: Comparaison de la vitesse de convergence du NSWCMA et du CMA pour un canal bruité linéaire

Ce comportement a été vérifié à l'aide des simulations par ordinateur, comme il peut être remarqué dans les deux figures suivantes: la figure A.3 montre l'évolution de l'ouverture de l'oeil d'un canal de communication linéaire bruité à l'aide du CMA et du NSWCMA. Dans cet exemple $N = 21$. Il est clair que l'augmentation de L peut accélérer considérablement la vitesse de convergence (à remarquer que pour $L = 7$ l'oeil du canal s'ouvre 7 fois plus vite que pour le CMA). Notons également que dans cet exemple l'oeil est initialement fermé (le coefficient d'ouverture d'oeil est plus grand que 1).

La figure A.4 montre un exemple où le problème des minima locaux est adressé: à gauche, on peut voir l'évolution du CMA pour deux initialisations différentes: il est clair qu'une des deux converge vers une solution inacceptable. À droite, on peut voir la même expérience pour le NSWCMA. On peut observer que malgré le fait que la mauvaise initialisation le NSWCMA avance vers un minimum inacceptable, il arrive rapidement à s'échapper et arrive au même point que la bonne initialisation. Ceci montre l'amélioration de performance du NSWCMA en ce qui concerne les minima locaux.

A.2.2 Un principe de séparation

L'algorithme NSWCMA présenté au paragraphe précédent peut être vu comme une modification de l'algorithme APA pour être appliqué à l'égalisation aveugle. Le APA est un algorithme de filtrage adaptatif classique qui minimise à chaque itération le critère déterministe suivant:

$$\min_{W_{k+1}} \left\{ \|D_k - \mathbf{X}_k^H W_{k+1}\|_{P_k^{-1}}^2 + \left(\frac{1}{\bar{\mu}} - 1\right) \|W_{k+1} - W_k\|^2 \right\}, \quad (\text{A.33})$$

où D_k est un vecteur $L \times 1$ qui contient L échantillons du signal désiré:

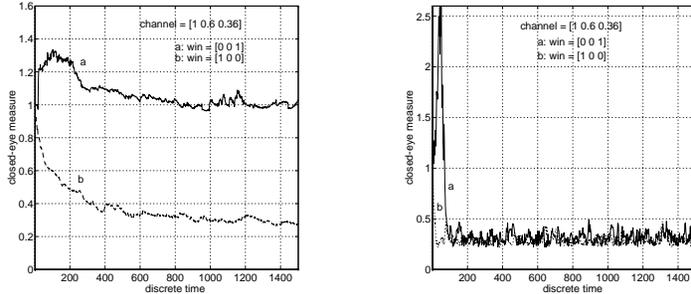


Figure A.4: Comparaison des performances respectives du NSWCMA et du CMA en terme de "ill-convergence"

$$D_k = [d_k \cdots d_{k-L+1}]^T .$$

L'algorithme résultant est le APA (Affine Projection Algorithm) et est donné par

$$W_{k+1} = W_k + \bar{\mu} \mathbf{X}_k^H P_k^{-1} (D_k - \mathbf{X}_k^H W_k) . \quad (\text{A.34})$$

En comparant (A.34) à (A.32), on peut remarquer que le NSWCMA aurait pu être dérivé par le APA en remplaçant le vecteur D_k par le vecteur $\text{sign}(\mathbf{X}_k^H W_k)$:

$$D_k \equiv \text{sign}(\mathbf{X}_k^H W_k) .$$

D'ailleurs, le NSWCMA n'a pas été dérivé de cette façon-ci, mais en minimisant un critère de type "module constant". Le fait que l'algorithme résultant ressemble à une modification du APA nous motive à formuler le *principe de séparation* suivant:

Un principe de séparation pour l'égalisation aveugle: Un algorithme adaptatif de type "Module-Constant" pour l'égalisation aveugle peut être dérivé par un algorithme de filtrage adaptatif classique en remplaçant le signal désiré $\{D_k\}$ par $\{\text{sign}(\mathbf{X}_k^H W_k)\}$.

Ce principe nous permet d'obtenir plusieurs classes d'algorithmes pour l'égalisation aveugle sans passer par une dérivation exacte, en étant motivé par le fait que dans le cas du NSWCMA, cela donne le même algorithme. Le tableau A.1 montre quelques algorithmes dérivés en appliquant ce principe.

La performance de certains algorithmes dérivés avec ce principe sera montrée par la suite.

LMS	$W_{k+1} = W_k + \mu X_k(d_k - y_k)$
CMA 1-2	$W_{k+1} = W_k + \mu X_k \left(\frac{y_k}{ y_k } - y_k \right)$
NLMS	$W_{k+1} = W_k + \frac{\bar{\mu}}{\ X_k\ _2^2} X_k(d_k - y_k)$
NCMA	$W_{k+1} = W_k + \frac{\bar{\mu}}{\ X_k\ _2^2} X_k \left(\frac{y_k}{ y_k } - y_k \right)$
APA	$W_{k+1} = W_k + \bar{\mu} \mathbf{X}_k P_k^{-1} (D_k - \mathbf{X}_k^H W_k)$
NSWCMA	$W_{k+1} = W_k + \bar{\mu} \mathbf{X}_k P_k^{-1} (\text{sign}(\mathbf{X}_k^H W_k) - \mathbf{X}_k^H W_k)$
RLS	$W_{k+1} = W_k + P_k^{-1} X_k(d_k - y_k)$
'RLS-CMA'	$W_{k+1} = W_k + P_k^{-1} X_k \left(\frac{y_k}{ y_k } - y_k \right)$

Table A.1: Certains algorithmes du type "à Module Constant" et leur correspondants pour le filtrage adaptatif classique

A.2.3 Une classe d'algorithmes régularisés

Comme indiqué dans le paragraphe A.2.1, le NSWCMA pose le problème suivant de APA: quand le nombre L de contraintes imposées devient grand (comparable à la longueur N de l'égaliseur), l'algorithme amplifie beaucoup le bruit additif de son signal d'entrée, ce qui peut amener à un mauvais comportement en régime stationnaire. Un exemple où ce problème se manifeste est présenté dans la figure A.5. Dans cette figure on a choisi $L = N$ pour une expérience d'égalisation aveugle. L'amplification du bruit est due au mauvais conditionnement de la matrice de covariance $P_k = \mathbf{X}_k^H \mathbf{X}_k$. La figure A.5 montre en haut l'évolution du conditionnement de cette matrice pendant la convergence de l'algorithme et en bas l'évolution de l'ouverture de l'oeil: il est clair que malgré la convergence initiale vers une bonne solution, l'algorithme est incapable de rester dans une région de bonne performance. Comme ce problème existe aussi pour le APA, on va d'abord proposer une solution pour celui-ci.

Le problème étant dû au mauvais conditionnement de la matrice de covariance, on va essayer de *régulariser* cette matrice afin de garantir un meilleur conditionnement. Une méthode permettant d'effectuer une telle régularisation est de remplacer P_k par une matrice de

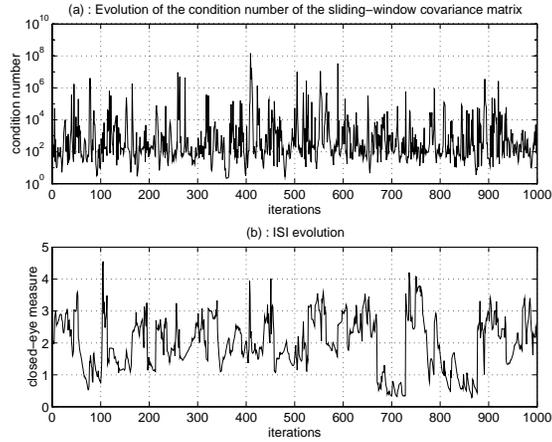


Figure A.5: Le mauvais effet d'une fenêtre glissante longue

covariance qui est pondérée de façon *exponentielle*:

$$R_k = \lambda R_{k-1} + \mathcal{X}_k \mathcal{X}_k^H, \quad (\text{A.35})$$

où \mathcal{X}_k est la première colonne de \mathbf{X}_k^H :

$$\mathcal{X}_k = \begin{bmatrix} x_k \\ x_{k-1} \\ \dots \\ x_{k-L+1} \end{bmatrix}, \quad (\text{A.36})$$

et λ est un *facteur d'oubli*. Maintenant on considère le critère déterministe suivant:

$$\min_{W_{k+1}} \left\{ \|D_k - \mathbf{X}_k^H W_{k+1}\|_{S_k^{-1}}^2 + \|W_{k+1} - W_k\|^2 \right\}, \quad (\text{A.37})$$

avec

$$S_k = \mu^{-1} R_k - \mathbf{X}_k^H \mathbf{X}_k. \quad (\text{A.38})$$

Ce critère est minimisé exactement à chaque itération par l'algorithme suivant:

$$\begin{aligned} R_k &= \lambda R_{k-1} + \mathcal{X}_k \mathcal{X}_k^H \\ W_{k+1} &= W_k + \mu \mathbf{X}_k R_k^{-1} (D_k - \mathbf{X}_k^H W_k). \end{aligned} \quad (\text{A.39})$$

L'équation (A.39) est une nouvelle classe d'algorithmes pour le filtrage adaptatif classique. Grâce à la fenêtre exponentielle appliquée aux données, on attend des algorithmes de cette classe d'éviter le problème de grande amplification de bruit présent dans APA quand L est

proche de N . Une organisation du calcul proposée dans le chapitre 4 conduit à une complexité de $2N + 6L^2 + 10L$ multiplications/itération.

En appliquant le principe de séparation à l'algorithme (A.39) on obtient une contrepartie pour l'égalisation aveugle:

$$\begin{aligned} R_k &= \lambda R_{k-1} + \mathcal{X}_k \mathcal{X}_k^H \\ W_{k+1} &= W_k + \mu \mathbf{X}_k R_k^{-1} (\text{sign}(\mathbf{X}_k^H W_k) - \mathbf{X}_k^H W_k). \end{aligned} \quad (\text{A.40})$$

On appelle cette classe d'algorithmes (A.40) NSWERCMA (Normalized Sliding Window with Exponential Regularization Constant Modulus Algorithms). L'amélioration du comportement de l'algorithme quand L est choisi près de N peut être observé dans la figure A.6: Dans cette figure on montre la performance du NSWERCMA pour le même canal que celui

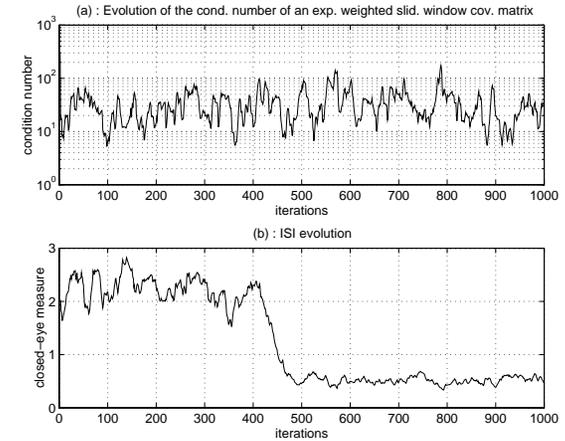


Figure A.6: L'amélioration en performance due à l'utilisation d'une fenêtre exponentielle

utilisé pour la figure A.5. Notons (figure A.6(b)) que le NSWERCMA arrive à ouvrir l'oeil et rester dans un régime stationnaire acceptable. Cela reflète le meilleur conditionnement de sa matrice de covariance (figure A.6(a)).

Un autre exemple de simulation comparative dans le cadre d'égalisation aveugle est montré dans la figure A.7. Il est intéressant de remarquer comment le nouvel algorithme arrive à ouvrir l'oeil du canal considérablement plus rapidement que les autres algorithmes (il est même plus rapide que le RLS-CMA).

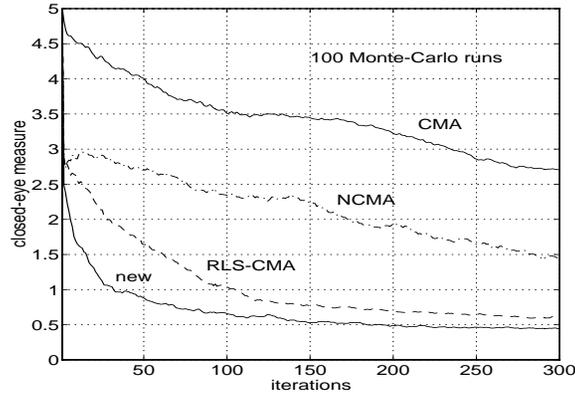


Figure A.7: L'amélioration en vitesse de convergence atteinte par le NSWERCMA

A.2.4 Egalisation Dirigé par les Décisions (DD)

L'Algorithme Dirigé par les Décisions (DDA) est donné par l'équation suivante:

$$W_{k+1} = W_k + \mu X_k (dec(y_k) - y_k) , \quad (\text{A.41})$$

où $dec(y)$ représente le symbole de la constellation émise qui est le plus proche de y . Comme déjà indiqué, cet algorithme a été proposé pour la phase finale d'une procédure d'égalisation: quand l'oeil du canal est déjà ouvert, cet algorithme peut être utilisé pour continuer l'adaptation et même améliorer la performance en régime stationnaire. Cela est dû au fait que quand l'oeil du canal est suffisamment ouvert les décisions prises seront très souvent les bonnes, et donc la sortie de l'élément de décision ressemble à une séquence d'apprentissage.

D'ailleurs, le DDA peut être vu comme un algorithme d'égalisation du type de Busgang qui utilise la non linéarité $dec(y_k)$. Néanmoins, le DDA n'est pas utilisé en pratique pour ouvrir l'oeil du canal car il est connu pour "ne pas être capable d'ouvrir un oeil initialement fermé". Cette conclusion a été faite sur la base des remarques sur la performance du DDA ([Sat75], [God80]) ainsi que des études théoriques sur le sujet ([Maz80], [ME84]). D'ailleurs c'est à cause de cette idée que les algorithmes hybrides qui modifient le DDA ont été proposés (voir [PP87], [HD93]).

Dans ce travail on effectue une étude de l'algorithme DDA pour le cas particulier où la constellation émise a un module constant (ce qui correspond à une modulation de phase). La motivation pour cette étude était basée sur l'intuition suivante: le DDA ressemble beaucoup

à l'algorithme CMA 1-2:

$$W_{k+1} = W_k + \mu X_k (r_1 \text{sign}(y_k) - y_k) . \quad (\text{A.42})$$

En effet, les deux algorithmes diffèrent seulement par le choix de leurs signaux d'erreur, qui sont donnés par

$$\begin{cases} e_k^D = y_k - dec(y_k) \\ e_k^C = y_k - r_1 \text{sign}(y_k) , \end{cases} \quad (\text{A.43})$$

pour le DDA et le CMA 1-2, respectivement. La figure A.8 montre ces deux signaux d'erreur dans le cas d'une constellation MAQ-4.

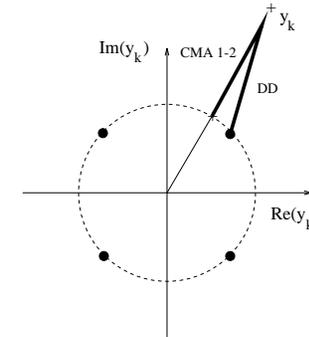


Figure A.8: La similarité entre le CMA 1-2 et le DDA

Il est évident que pour une modulation de phase avec plusieurs symboles équi-répartis sur le cercle, les deux signaux d'erreur se rapprochent de plus en plus. En effet, dans le cas asymptotique où la constellation a une infinité de points sur le cercle, les deux algorithmes sont identiques! Cela fait penser que si le CMA 1-2 arrive à ouvrir l'oeil fermé d'un canal, le DDA devrait être aussi capable d'avoir une performance similaire.

Pour examiner cette intuition, on a d'abord effectué une étude de l'algorithme en supposant que:

- Le signal émis $\{a_i\}$ est un signal i.i.d. à module constant
- L'égaliseur a une longueur infinie
- Il n'y a pas de bruit additif

De cette analyse, on trouve que dans le cas asymptotique d'une infinité de symboles sur le cercle, le CMA 1-2 (qui est équivalent dans ce cas au DDA) n'a pas de minima locaux. Ce résultat est parallèle à celui présenté par Benveniste *et al* [BG84] pour l'algorithme de Sato:

le Sato n'a pas de minima locaux sous les conditions d'égaliseur infini, d'absence de bruit, et des constellations continues sous-Gaussiennes. En plus, il montre que le DDA devrait être capable d'ouvrir l'oeil d'un canal, même dans des cas pratiques, si le signal émis a un module constant. Combiné avec le fait que le DDA a en général un meilleur comportement dans son régime stationnaire comparé aux CMA, notre étude motive l'utilisation de l'algorithme DDA avec des constellations CM.

Un exemple qui vérifie nos arguments théoriques est montré dans la figure A.9. Il est à

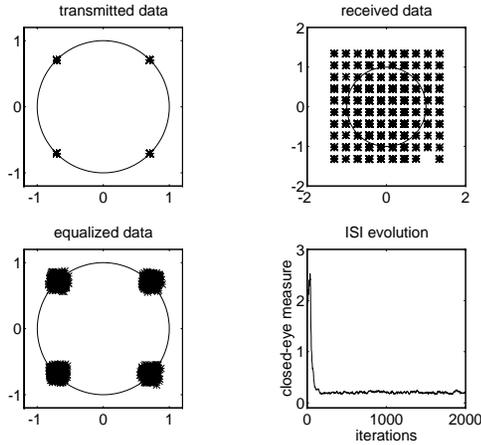


Figure A.9: L'ouverture de l'oeil effectué par le DDA

noter dans cette figure que l'algorithme DDA arrive à récupérer la constellation MAQ-4 émise malgré le fait que l'oeil du canal était initialement fermé.

Un schéma hybride CM-DD pour les constellations non-CM

Les résultats de nos recherches indiquant que la raison principale qui empêche le DDA de se comporter correctement est la présence de plusieurs amplitudes dans l'alphabet des symboles émis, on propose le schéma suivant pour éviter ce problème:

- Construire à partir de la constellation émise une autre constellation réduite en remplaçant chaque symbole a_i par sa projection sur le cercle: $r_1 \text{sign}(a_i)$.
- Utiliser cette constellation réduite à module constant pour prendre les décisions pour les échantillons reçus.

Dans ce cas, on remplace la fonction $\text{dec}(\cdot)$ par la fonction $\text{cdec}(\cdot)$ définie comme

$$\text{cdec}(y_k) = \text{le symbole de } \{\text{sign}(a_i)\} \text{ le plus proche de } y_k. \quad (\text{A.44})$$

Cet algorithme est en effet un alternatif à l'algorithme GSA, qui respecte mieux l'idée originale de Sato: supprimer les multiples niveaux. La différence avec le GSA est que la constellation réduite qui est utilisée par le GSA a seulement 4 symboles, tandis que dans notre schéma il y en a plusieurs. Un exemple de ce principe, pour une constellation MAQ-16, est montré dans la figure A.10. Dans cette figure la constellation originale est indiquée avec '+' et la

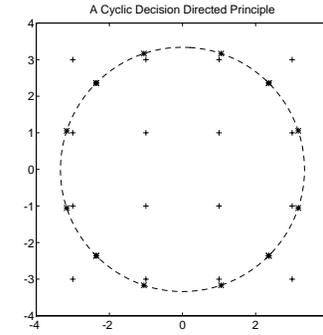


Figure A.10: Un principe hybride CMA-DD: 16-QAM

constellation projetée avec des '*'. Un exemple qui montre la performance de ce schéma est montré dans la figure A.11. Il est à noter que l'algorithme hybride arrive à converger vers une bonne solution, tandis que le DDA n'arrive pas à ouvrir l'oeil du canal.

A.2.5 Un principe bilinéaire pour l'égalisation aveugle

Comme indiqué auparavant, un problème essentiel des algorithmes de type Bussgang est le problème de "ill-convergence". On a déjà vu que ce problème peut être surmonté à l'aide des algorithmes normalisés: ces algorithmes ne correspondent pas à des fonctions de coût convexes, mais arrivent souvent à s'échapper des minima locaux. Ici on propose une autre approche pour ce problème: on propose une fonction de coût *convexe* pour garantir la convergence vers un optimum global unique.

Le concept qu'on propose est le suivant: considérons le développement suivant du module au carré du signal reçu:

$$\begin{aligned} |y_k|^2 = y_k y_k^* &= (w_0 w_0^* x_k x_k^* + \dots + w_0 w_{N-1}^* x_k x_{k-N+1}^* + \dots \\ &+ (w_{N-1} w_0^* x_{k-N+1} x_k^* + \dots + w_{N-1} w_{N-1}^* x_{k-N+1} x_{k-N+1}^*) \end{aligned} \quad (\text{A.45})$$

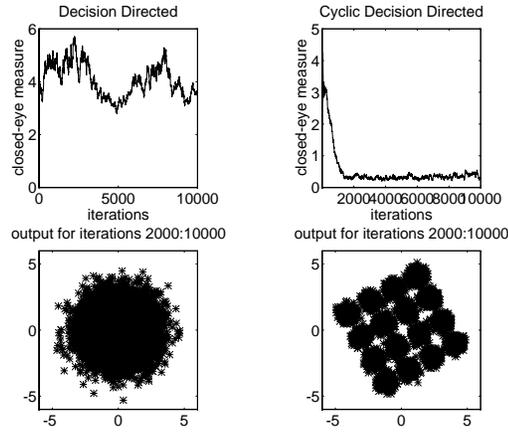


Figure A.11: L'amélioration de l'algorithme hybride par rapport au DDA

Dans ce cas le module au carré $|y_k|^2$ peut s'écrire comme

$$|y_k|^2 = z_k = \mathcal{X}_k^H \theta_k, \quad (\text{A.46})$$

où θ_k est un vecteur de paramètres défini comme

$$\theta_k = [w_0 w_0^* \cdots w_0 w_{N-1}^* \quad w_1 w_0^* \cdots w_1 w_{N-1}^* \cdots w_{N-1} w_0^* \cdots w_{N-1} w_{N-1}^*]^T, \quad (\text{A.47})$$

et \mathcal{X}_k un vecteur de régression défini comme

$$\mathcal{X}_k = [x_k x_k^* \cdots x_k x_{k-N+1}^* \quad x_{k-1} x_k^* \cdots x_{k-1} x_{k-N+1}^* \cdots x_{k-N+1} x_k^* \cdots x_{k-N+1} x_{k-N+1}^*]^H. \quad (\text{A.48})$$

A noter que ces deux vecteurs contiennent des termes *bilinéaires* des coefficients de l'égaliseur et des échantillons reçus, respectivement. Avec cette notation, la fonction de coût de l'algorithme CMA 2-2 peut être écrite sous la forme

$$J_2(W) = \frac{1}{4} E(|y|^2 - 1)^2 = \frac{1}{4} E(z - 1)^2,$$

ce qui peut mener au problème d'optimisation suivant

$$\min_{\theta} J^{bil}(\theta) = \min_{\theta} E(z - 1)^2 = \min_{\theta} E(\mathcal{X}^H \theta - 1)^2. \quad (\text{A.49})$$

En ayant donc effectué une *transformation des paramètres* on a pu construire une fonction de coût quadratique, et donc *convexe* par rapport aux paramètres θ . En définissant les quantités

\mathbf{R} et \mathbf{d} comme

$$\begin{aligned} \mathbf{R} &= E(\mathcal{X}\mathcal{X}^H) \\ \mathbf{d} &= E(\mathcal{X}) \end{aligned} \quad (\text{A.50})$$

on arrive à la conclusion que si \mathbf{R} est non-singulière, le problème (A.49) a une solution unique, donnée par

$$\theta = \mathbf{R}^{-1} \mathbf{d}. \quad (\text{A.51})$$

A partir de cette solution l'égaliseur W correspondant peut être calculé de la manière suivante:

$$W^{opt} = \sqrt{\lambda} V(\lambda), \quad (\text{A.52})$$

où λ et $V(\lambda)$ sont respectivement l'unique valeur propre non-nulle et le vecteur propre correspondant, de la matrice suivante:

$$\Theta = \begin{bmatrix} \theta^{(0)} & \theta^{(N)} & \cdots & \theta^{(N^2-N)} \\ \theta^{(1)} & \theta^{(N+1)} & \cdots & \theta^{(N^2-N+1)} \\ \vdots & \vdots & \ddots & \vdots \\ \theta^{(N-1)} & \theta^{(2N-1)} & \cdots & \theta^{(N^2-1)} \end{bmatrix} = \mathcal{G}(\theta). \quad (\text{A.53})$$

Le problème de "ill-convergence" et des minima locaux est donc surmonté: en utilisant cette méthode on peut trouver sans ambiguïté l'égaliseur optimal.

Plusieurs manières d'implanter ce principe sont données dans le chapitre 6. Le principe général d'égalisation aveugle bilinéaire est affiché dans la figure A.12. Remarquez que la décomposition (A.52) ne doit pas nécessairement être effectuée à chaque itération de l'algorithme adaptatif.

Une analyse plus profonde de la méthode bilinéaire est aussi présentée dans le chapitre 6, notamment l'influence du bruit additif et le problème de sur-paramétrisation sont analysés. Une méthode pour éliminer l'influence du bruit additif est donné pour le cas Gaussien. Une méthode pour surmonter le problème de sur-paramétrisation est aussi présentée. Les figures A.13 et A.14 montrent deux exemples où la "ill-convergence" de l'algorithme CMA est surmontée en utilisant la méthode bilinéaire: la figure A.13 montre un cas idéal (sans bruit, parfaitement paramétrisé). L'égaliseur a deux coefficients. A gauche, on peut voir les points initiaux et finaux de la méthode bilinéaire pour 40 initialisations différentes sur un cercle. Notez qu'il n'y a qu'un seul point de convergence (qui est aussi optimal). A droite, on a la même expérience pour le CMA, où on peut voir qu'il y a une paire des points de convergence non optimaux.

La figure A.14 montre un résultat similaire pour le cas plus réaliste d'un canal RIF bruité. On peut à nouveau constater la convergence globale de la méthode bilinéaire, contrairement au CMA.

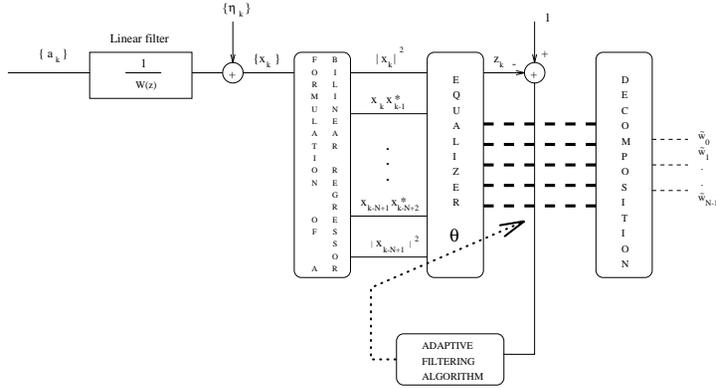


Figure A.12: Le principe de l'égalisation adaptative bilinéaire

A.3 Egalisation et identification aveugle multicanal

A la suite des méthodes de type Bussgang présentées précédemment on a regardé le problème d'égalisation multicanal. On s'intéresse à la structure d'un canal SIMO (Single Input - Multiple Output). Une telle structure apparaît dans la figure A.15. Les m canaux ont la même entrée discrète $\{a_i\}$, et leurs sorties discrètes sont de la forme

$$x_j(k) = \sum_{i=-\infty}^{\infty} a(i)h_j(k-i) \quad , j = \{1, \dots, m\} \quad . \quad (\text{A.54})$$

Une telle structure peut être appliquée pour décrire soit un système d'égalisation à sur-échantillonnage, ou un système d'égalisation multi-capteurs. Dans le premier cas les m signaux $\{x_j(k)\}$, $j = 1, \dots, m$ représentent les m phases différentes du signal reçu sur-échantillonné à la cadence T/m . Dans le deuxième cas ils représentent les sorties des m capteurs.

La structure d'un canal SIMO a des conséquences importantes et ce à plusieurs niveaux ne concernant pas seulement les propriétés de la sortie multicanal mais aussi son identification à partir de la sortie. Dans la suite on va se concentrer sur le cas d'une réception suréchantillonné par un seul capteur. On définit le facteur de suréchantillonnage OF (Oversampling Factor) comme

$$\text{OF} = \frac{T}{T_s} \quad , \quad (\text{A.55})$$

où T et T_s sont respectivement les périodes de Baud et d'échantillonnage à la sortie du canal.

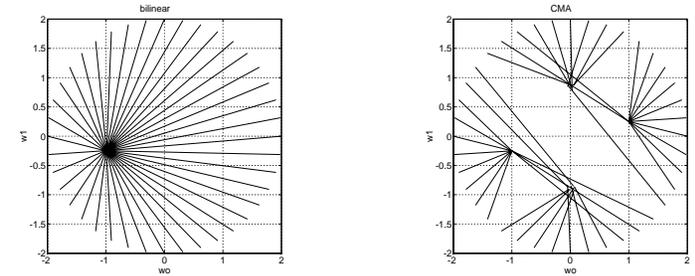


Figure A.13: Simulations comparatives pour un cas idéal

Le résultat suivant est fondamental:

- Le signal discret $\{x_i\}$ obtenu par suréchantillonnage de la sortie d'un canal linéaire avec un facteur OF > 1 est cyclostationnaire.

D'autre part, si on considère la sortie multicanal comme un vecteur qui contient les m phases consécutives de la sortie de la façon suivante

$$\mathbf{x}_k = [x_1(k) \ x_2(k) \ \dots \ x_m(k)]^T \quad , \quad (\text{A.56})$$

on a le résultat suivant:

- Le signal vectoriel $\{\mathbf{x}_k\}$ à la sortie du système SIMO est stationnaire au sens vectoriel: chacune de ses entrées est un signal scalaire stationnaire.

Cet impact du suréchantillonnage sur la stationnarité du signal continu reçu $x(t)$ est affiché dans la figure A.16. Une conséquence importante de la propriété de cyclostationnarité de la sortie du canal suréchantillonné est la suivante: le canal SIMO peut être identifié en utilisant les statistiques de second ordre de sa sortie (sauf si les m canaux partagent des zéros en commun) (ce résultat a été récemment donné par plusieurs chercheurs). Avant de s'intéresser à l'aspect aveugle, nous étudions l'égalisation ZF (Zero-Forcing) du canal SIMO.

A.3.1 Egalisation ZF (Zero-Forcing)

Le schéma d'égalisation linéaire multicanal pour le cas $m = 2$ est montré dans la figure A.17. Dans le cas général (m canaux, m égaliseurs) la sortie vectorielle $\mathbf{x}(k)$ définie en (A.56) peut être mise sous la forme

$$\mathbf{x}(k) = \sum_{i=0}^{N-1} \mathbf{h}(i)a_{k-i} + \mathbf{v}(k) = \mathbf{H}_N \mathbf{A}_N(k) + \mathbf{v}(k) \quad , \quad (\text{A.57})$$

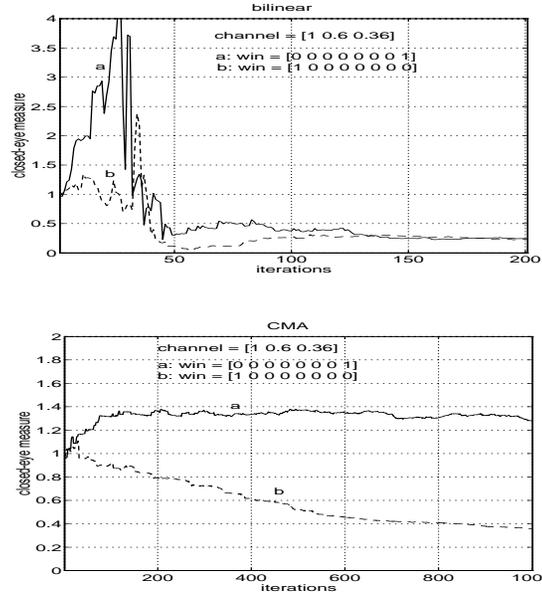


Figure A.14: Simulations comparatives pour le cas d'un canal RIF bruité

où $\mathbf{h}(k)$, $\mathbf{v}(k)$ sont définis comme

$$\mathbf{v}(k) = \begin{bmatrix} v_1(k) \\ \vdots \\ v_m(k) \end{bmatrix}, \mathbf{h}(k) = \begin{bmatrix} h_1(k) \\ \vdots \\ h_m(k) \end{bmatrix}, \quad (\text{A.58})$$

et représentent le bruit et le canal discret vectoriel, respectivement. Si on définit les sous-canaux H_i comme

$$H_i = [h_i(0) \cdots h_i(N-1)], \quad (\text{A.59})$$

la matrice du canal \mathbf{H}_N de taille $m \times N$ est définie comme

$$\mathbf{H}_N = \begin{bmatrix} h_1(0) & \cdots & h_1(N-1) \\ \vdots & \cdots & \vdots \\ h_m(0) & \cdots & h_m(N-1) \end{bmatrix} = [\mathbf{h}(0) \cdots \mathbf{h}(N-1)] = \begin{bmatrix} H_1 \\ \vdots \\ H_m \end{bmatrix}. \quad (\text{A.60})$$

Finalement, $A_N(k)$ représente un vecteur $N \times 1$ des symboles émis, défini comme

$$A_N(k) = [a_k^H \cdots a_{k-N+1}^H]^H. \quad (\text{A.61})$$

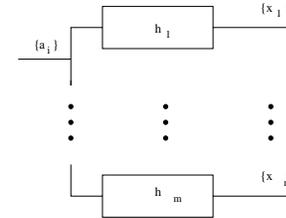
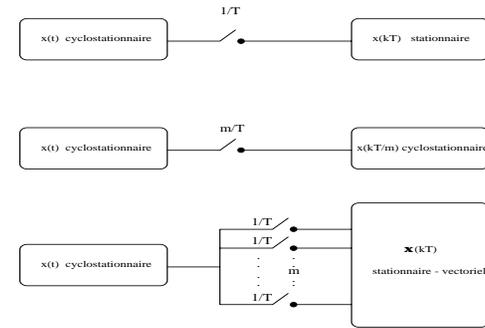


Figure A.15: Un canal SIMO

Figure A.16: L'effet du suréchantillonnage de $x(t)$ à sa stationnarité

Dans le domaine des z on définit les transformées en z du canal et de l'égaliseur suréchantillonné comme

$$H(z) = \sum_{j=1}^m z^{-(j-1)} H_j(z^m), \quad (\text{A.62})$$

et

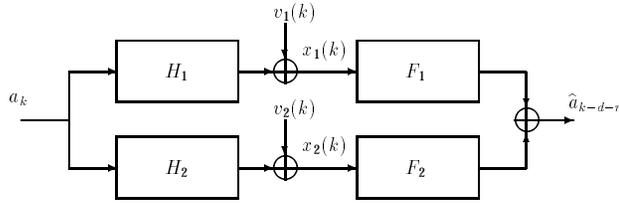
$$F(z) = \sum_{j=1}^m z^{(j-1)} F_j(z^m), \quad (\text{A.63})$$

respectivement. En définissant aussi les transformées en z vectoriels du canal et d'égaliseur comme

$$\begin{aligned} \mathbf{H}(z) &= \sum_{k=0}^{N-1} \mathbf{h}(k) z^{-k} \\ \mathbf{F}(z) &= \sum_{k=0}^{N-1} \mathbf{f}(k) z^{-k}. \end{aligned} \quad (\text{A.64})$$

Dans ce cas la condition d'égalisation ZF prend la forme

$$\mathbf{F}(z)\mathbf{H}(z) = z^{-n}, \quad n = 0, 1, \dots, N+L-2. \quad (\text{A.65})$$

Figure A.17: Principe de l'égalisation multicanal pour le cas $m = 2$

La contrepartie de (A.65) dans le domaine temporel est l'équation suivante

$$[\mathbf{f}(0) \ \cdots \ \mathbf{f}(L-1)] \begin{bmatrix} \mathbf{h}(0) & \cdots & \mathbf{h}(N-1) & \mathbf{0}_{m \times 1} & \cdots & \mathbf{0}_{m \times 1} \\ \mathbf{0}_{m \times 1} & \mathbf{h}(0) & \cdots & \mathbf{h}(N-1) & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \mathbf{0}_{m \times 1} \\ \mathbf{0}_{m \times 1} & \cdots & \mathbf{0}_{m \times 1} & \mathbf{h}(0) & \cdots & \mathbf{h}(N-1) \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}^T, \quad (\text{A.66})$$

ou de façon équivalente,

$$\mathbf{F}_L \mathcal{T}_L(\mathbf{H}_N) = [0 \cdots 0 \ 1 \ 0 \cdots 0]. \quad (\text{A.67})$$

L'équation (A.67) étant un système linéaire de $L + N - 1$ équations à Lm inconnues, elle aura donc au moins une solution si la longueur L de chaque égaliseur satisfait la condition suivante

$$L \geq L = \left\lceil \frac{N-1}{m-1} \right\rceil, \quad (\text{A.68})$$

et si la matrice $\mathcal{T}_L(\mathbf{H}_N)$ est de rang plein (ce qui est vrai si les canaux $H_j(z)$, $j = 1, \dots, m$ n'ont pas de zéro en commun). On arrive donc à la conclusion suivante:

- Des égaliseurs ZF de longueur finie qui égalisent parfaitement en l'absence de bruit existent pour le canal SIMO, si les m sous-canaux n'ont pas de zéro en commun et si la longueur L de chaque égaliseur satisfait (A.68).

Une interprétation intéressante de l'égalisation multicanal est la suivante: on a démontré que la structure canal SIMO - égaliseur MISO est équivalente à la structure SISO affichée dans la figure A.18, où $H(z)$ et $F(z)$ sont définies respectivement en (A.62) et (A.63). Désignons maintenant par $G(z)$ la cascade de $H(z)$ et de $F(z)$:

$$G(z) = F(z)H(z). \quad (\text{A.69})$$

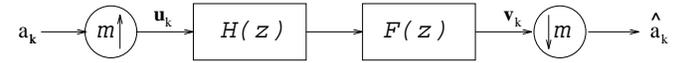


Figure A.18: Représentation équivalente de la structure d'égalisation multicanal

La condition pour l'égalisation ZF prend maintenant la forme

$$\frac{1}{m} \sum_{i=1}^m G(\omega^{i-1}z^{1/m}) = 1. \quad (\text{A.70})$$

La contrepartie de cette équation dans le domaine fréquentiel est

$$g_i(k) = \delta(k), \text{ } i \text{ est un parmi } \{1, \dots, m\}. \quad (\text{A.71})$$

L'équation (A.71) permet l'interprétation suivante de l'égalisation ZF multicanal: une seule des m phases de la cascade $\{g_i\}$ doit être un Dirac pour atteindre une égalisation ZF parfaite. Cela indique pourquoi il est possible d'égaliser avec un nombre fini de paramètres: il n'est pas nécessaire d'inverser un polynôme (voir eq. (A.70)) comme dans le cas de l'égalisation non-suréchantillonnée. Un problème existera seulement si tous les canaux ont des zéros en commun, car dans ce cas il faudra inverser leur facteur commun qui sortira de la somme dans (A.70).

Selon les résultats de ce paragraphe, trois problèmes importants de l'égalisation non-suréchantillonnée peuvent être surmontés avec l'égalisation suréchantillonnée: d'une part il n'est plus nécessaire d'avoir des égaliseurs de longueur infinie pour atteindre une égalisation parfaite en l'absence de bruit. D'autre part, même si les différents sous-canaux ont des zéros en commun (pourvu qu'ils ne soient pas les mêmes), l'égalisation ZF peut être tout de même atteinte. En plus, l'égalisation est possible même si quelques canaux ont des zéros sur le cercle unité (pourvu que ces zéros ne soient pas partagés par *tous* les canaux).

On démontre aussi dans le chapitre 7 comment se traduit la condition d'égalisation ZF multicanal dans le domaine des fréquences, ce qui se traduit par une condition fréquentielle de Nyquist pour le multicanal. L'effet de la longueur de l'égaliseur dans le cas bruité est aussi analysé: clairement, en présence du bruit l'augmentation de la longueur améliore la performance. L'égaliseur optimal pour une longueur donnée est aussi dérivé.

A.3.2 Egalisation aveugle multicanal par prédiction linéaire

La faisabilité de l'identification multicanal à l'aide des statistiques du second ordre de sa sortie a été démontrée dans le domaine fréquentiel. Il s'avère que l'égalisation aveugle du

multicanal peut être effectuée dans le domaine temporel aussi à l'aide de la prédiction linéaire multivariable. Le problème de prédiction dans le cas multicanal prend la forme

$$\min_{\mathbf{P}_L} \sigma_{\tilde{\mathbf{x}}_L}^2, \quad (\text{A.72})$$

où $\sigma_{\tilde{\mathbf{x}}_L}^2$ est une matrice $m \times m$ qui représente la variance de l'erreur de prédiction multivariable d'ordre L :

$$\sigma_{\tilde{\mathbf{x}}_L}^2 = E \tilde{\mathbf{x}}(k) \tilde{\mathbf{x}}^H(k), \quad (\text{A.73})$$

où $\tilde{\mathbf{x}}(k)$ est l'erreur de prédiction:

$$\begin{aligned} \tilde{\mathbf{x}}(k) | \mathbf{X}_{L(k-1)} &= \mathbf{x}(k) - \hat{\mathbf{x}}(k) | \mathbf{X}_{L(k-1)} \\ &= [I_m \quad -\mathbf{P}_L] \mathbf{X}_{L+1}(k), \end{aligned} \quad (\text{A.74})$$

et \mathbf{P}_L le filtre de prédiction de dimension $m \times Lm$:

$$\mathbf{P}_L = [\mathbf{p}_1 \cdots \mathbf{p}_L]. \quad (\text{A.75})$$

Il en résulte que la solution à (A.72) \mathbf{P}_L donne un égaliseur ZF à travers l'équation suivante:

$$F_{ZF} = \frac{1}{\mathbf{h}^H(0)\mathbf{h}(0)} \mathbf{h}^H(0) [I_m \quad -\mathbf{P}_L]. \quad (\text{A.76})$$

Le coefficient $\mathbf{h}(0)$ d'ailleurs peut s'identifier à partir de la relation

$$\sigma_{\tilde{\mathbf{x}}_L}^2 = \sigma_{\tilde{\mathbf{x}}_{L+N-1}}^2 \mathbf{h}(0) \mathbf{h}^H(0), \quad (\text{A.77})$$

où $\sigma_{\tilde{\mathbf{x}}_{L+N-1}}^2$ est la variance d'erreur de prédiction des symboles. A l'aide des équations (A.76) et (A.77) on peut donc obtenir un égaliseur ZF par la prédiction linéaire multivariable, de façon aveugle! Cela enlève encore un problème présent dans les méthodes d'égalisation aveugle standards: l'utilisation des statistiques d'ordre supérieur (implicites ou explicites) ne sont pas nécessaires! Une autre conséquence importante de ce résultat est que l'égalisation aveugle est maintenant possible même si le signal émis est Gaussien!

A.3.3 Egalisation MMSE

Dans le cas non-suréchantillonné il est bien connu qu'en présence du bruit, les égaliseurs MMSE (Minimum-Mean-Square-Error) (minimisant l'erreur quadratique moyenne) ont une meilleure performance que les égaliseurs ZF. On constate que ce phénomène se répète aussi dans le cas des égaliseurs suréchantillonnés. De plus, on trouve une relation qui lie un égaliseur MMSE à un égaliseur ZF. Cette relation est la suivante

$$F_M = \Omega F_{ZF}, \quad (\text{A.78})$$

où Ω est une matrice définie comme

$$\Omega = I_{Lm} - \sigma_v^2 (R_L^*)^{-1} \quad (\text{A.79})$$

et F_M est l'égaliseur MMSE. Remarquez que des statistiques de second ordre (matrice de covariance) sont seulement nécessaires pour calculer Ω . De cette façon, en utilisant les statistiques de second ordre on peut calculer un égaliseur MMSE de façon aveugle (en passant par l'égaliseur ZF d'abord). La validité de cette méthode se confirme dans la figure A.19. En haut, on peut voir l'égaliseur MMSE dans le cas où $m = 2$. En bas, figure l'égaliseur calculé à l'aide de (A.79).

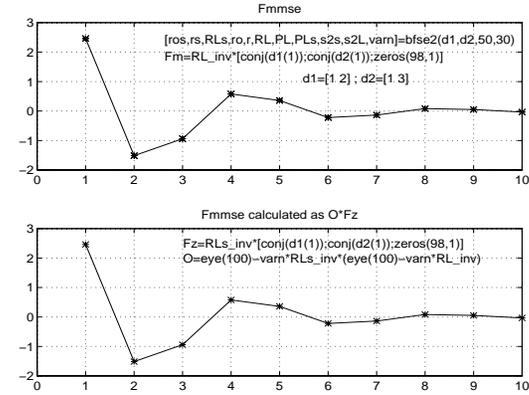


Figure A.19: Une vérification de (A.78)

Il se trouve qu'une méthode de complexité réduite peut être obtenue en utilisant la prédiction linéaire. L'égaliseur MMSE peut être obtenu par la relation suivante

$$F_{MMSE_0} = \begin{bmatrix} I_m \\ -\mathbf{P}_{L-1}^H \end{bmatrix} \sigma_{\tilde{\mathbf{x}}}^{-2} h_0^* \sigma_a^2, \quad (\text{A.80})$$

où σ_a^2 est la variance des symboles émis (considérés indépendants). L'équation (A.80) permet l'acquisition d'un égaliseur MMSE avec une complexité réduite (seule l'inversion de la matrice $m \times m$ $\sigma_{\tilde{\mathbf{x}}}^2$ est nécessaire pour (A.80)).

A.3.4 Méthode sous-espace

La matrice de covariance du signal reçu suréchantillonné a la forme suivante

$$\mathbf{R}_L^{\mathbf{x}} = \mathcal{T}_L(\mathbf{H}_N) \mathbf{R}_{L+N-1}^{\mathbf{s}} \mathcal{T}_L^H(\mathbf{H}_N) + \sigma_v^2 I_{Lm}, \quad (\text{A.81})$$

en présence de bruit additif blanc de variance σ_v^2 . Etant donné cette structure de matrice, on appelle l'espace colonne de la matrice $\mathcal{T}_L(\mathbf{H}_N)$ l'"espace signal" et son complément

orthogonal l' "espace bruit". Ces espaces sont caractérisés par les ensembles de vecteurs propres V_S et V_N définis par

$$\begin{aligned} \mathbf{R}_L^{\mathbf{X}} &= \sum_{i=1}^{L+N-1} \lambda_i V_i V_i^H + \sum_{i=L+N}^{L_M} \lambda_i V_i V_i^H \\ &= V_S \Lambda_S V_S^H + V_N \Lambda_N V_N^H, \end{aligned} \quad (\text{A.82})$$

respectivement. Si la matrice de Toeplitz $\mathcal{T}_L(\mathbf{H}_N)$ a un rang plein, alors elle sera générée par son espace signal:

$$\mathcal{T}_L(\mathbf{H}_N) = V_S T. \quad (\text{A.83})$$

Par contre, si la matrice de covariance est estimée par les données, la relation (A.83) n'est plus valable. Une façon d'estimer le canal dans ce cas est de résoudre le problème d'optimisation suivant

$$\min_{\mathbf{H}_N, T} \|\mathcal{T}_L(\mathbf{H}_N) - V_S T\|_F, \quad (\text{A.84})$$

où $\|Z\|_F^2 = \text{tr}\{Z^H Z\}$. (A.84) est un problème de sous-espace: on essaie de trouver le canal de façon que ça soit le plus proche possible du sous-espace signal de la matrice estimée.

Il se trouve que la solution du problème (A.84) (si on incorpore aussi la contrainte de non-trivialité $\|\mathbf{H}_N\|_2 = 1$) est le vecteur propre qui correspond à la valeur propre maximale de la matrice suivante

$$\sum_{i=1}^{L+N-1} \mathcal{T}_L(V_i^H) \mathcal{T}_L^H(V_i^H).$$

On obtient ainsi une méthode aveugle pour l'estimation du multicanal basée sur une matrice de covariance estimée.

A.3.5 Méthodes de Maximum de Vraisemblance Conditionnelle

Pour améliorer la qualité d'estimation, on peut aussi envisager des méthodes de maximum de vraisemblance (Maximum Likelihood). On va se concentrer sur des méthodes de maximum de vraisemblance conditionnelles: on attribue une certaine distribution statistique aux données émises qui ne correspond pas forcément à leur vraie distribution (ceci est fait pour avoir une complexité réduite). On propose deux méthodes différentes:

Méthode de Maximum de Vraisemblance Déterministe

Dans ce cas on suppose que les données sont des quantités *déterministes*, qui font partie des paramètres à estimer. Le problème d'optimisation correspondant peut s'écrire sous la forme

$$\min_{\mathbf{H}_N, A_{M+N-1}(k)} \|\mathbf{X}_M(k) - \mathcal{T}_M(\mathbf{H}_N) A_{M+N-1}(k)\|_2^2. \quad (\text{A.85})$$

Ce problème peut être minimisé de façon adaptative comme indiqué dans le paragraphe 7.9.1. Pour analyser la qualité de l'estimation atteinte par cette méthode on calcule la borne de Cramér-Rao correspondante. La borne pour le canal est donnée par

$$C(\hat{\mathbf{H}}_N^{\lambda T}) \geq \sigma_v^2 \left[\mathbf{A}_{M,N}^H(k) P_{\mathcal{T}_M(\mathbf{H}_N)}^\perp \mathbf{A}_{M,N}(k) \right]^+, \quad (\text{A.86})$$

où $^+$ est le pseudo-inverse d'une matrice.

Maximum de Vraisemblance Conditionnelle Gaussienne

Dans ce cas on suppose que les données émises ont une distribution Gaussienne. Cela ne correspond pas à la réalité, mais cette hypothèse peut aider à améliorer la qualité d'estimation. Le problème d'optimisation correspondant est le suivant

$$\min_{\mathbf{H}_N, A_{M+N-1}(k)} \left\| \begin{bmatrix} \mathbf{X}_M(k) \\ \mathbf{0}_{N+M-1} \end{bmatrix} - \begin{bmatrix} \mathcal{T}_M(\mathbf{H}_N) \\ \frac{\sigma_v}{\sigma_a} I_{N+M-1} \end{bmatrix} A_{M+N-1}(k) \right\|_2^2. \quad (\text{A.87})$$

La borne de Cramér-Rao correspondante à ce problème est la suivante

$$C(\hat{\mathbf{H}}_N^{\lambda T}) = \sigma_v^2 \left[\tilde{\mathbf{K}}_{M,N}^H(k) P_{\mathcal{T}_M(\mathbf{H}_N)(\mathbf{H}_N)}^\perp \tilde{\mathbf{K}}_{M,N}(k) \right]^{-1}. \quad (\text{A.88})$$

Cette méthode améliore considérablement dans certains cas la qualité d'estimation, grâce à la régularisation implicite due à l'hypothèse de Gaussiannité. Le tableau suivant montre une comparaison des performances des ces deux méthodes (aveugles) de maximum de vraisemblance dans le cas du système GSM d'une part et avec la méthode qui utilise la séquence d'apprentissage (TRS) d'autre part. On suppose deux sous-canaux de réponses impulsives

$$\begin{aligned} h_1 &= [0.3651 \ 0.5983 \ -0.0825 \ 0 \ 0 \ 0] \\ h_2 &= [0.4076 \ -0.0625 \ 0 \ 0 \ 0]. \end{aligned}$$

DML	GDML	TRS
99.9062	0.0149	0.0198

Notons l'amélioration atteinte par la méthode Gaussienne, qui est même plus performante que la méthode non-aveugle.

A.3.6 Application des méthodes existantes à la structure multicanal

Motivés par les avantages de la structure multicanal, nous avons appliqué les méthodes proposées dans la première partie de la thèse à l'égalisation sur-échantillonnée. Comme attendu (voir chapitre 8), l'utilisation de suréchantillonnage améliore la performance de tous les algorithmes. La figure A.20 montre une simulation comparative des performances de

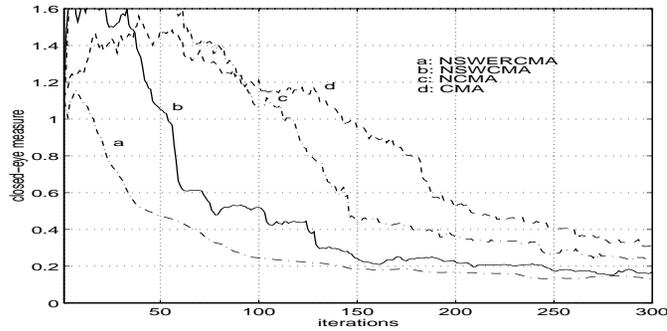


Figure A.20: Simulations comparatives pour un canal sur-échantillonné avec un facteur 2

divers algorithmes dans le cas d'un canal typique des communication mobiles. Le facteur de sur-échantillonnage dans ce cas est égal à 2. Notons l'amélioration en performance atteinte par les nouveaux algorithmes NSWCMA et NSWERCMA.

Bibliography

- [Age86] B. Agee. "The Least-Squares CMA: A New Technique for Rapid Correction of Constant Modulus Signals". In *Proc. ICASSP 86 Conference*, pages 953–956, Tokyo, Japan, 1986.
- [BBG⁺78] A. Benveniste, M. Bonnet, M. Goursat, C. Macchi, and G. Ruget. "Identification d'un système à non minimum de phase par approximation stochastique". *Research report*, N° 325, INRIA, 1978.
- [BD94] S. Buljore and J. F. Diouris. "Theoretical Study of a Multisensor Equalizer Using the MSE for the Radiomobile Channel(GSM Model)". In *Proc. 28th Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, CA, Oct. 31 - Nov. 2 1994.
- [BG83] A. Benveniste and M. Goursat. "Blind Equalizers". *Research report*, N° 219, INRIA, 1983.
- [BG84] A. Benveniste and M. Goursat. "Blind Equalizers". *IEEE Trans. on Communications*, COM-32(8):871–883, Aug. 1984.
- [BGR80] A. Benveniste, M. Goursat, and G. Ruget. "Robust Identification of a Nonminimum Phase System: Blind Adjustment of a Linear Equalizer in Data Communications". *IEEE Trans. on Automatic Control*, AC-25(3):1740–1748, June 1980.
- [BJ79] C. A. Belfiore and J. H. Park, Jr. "Decision Feedback Equalization". *Proceedings of the IEEE*, 67(8):1143–1156, Aug. 1979.
- [BM86] Y. Bresler and A. Macovski. "Exact Maximum Likelihood Parameter Estimation of Superimposed Signals in Noise". *IEEE Trans. on Acoustics, Speech, and Signal Processing*, ASSP-34(5):1081–1089, Oct. 1986.
- [BP86] E. Biglieri and G. Prati, editors. *Digital Communications*. Elsevier Science Publishers B. V., North Holland, 1986.
- [BR85] S. Bellini and F. Rocca. Blind deconvolution: Polyspectra or bussgang techniques? In E. Biglieri and G. Prati, editors, *Proc. Second Tirrenia International Workshop on Digital Communications*, pages 251–263, North-Holland, Sept. 2-6 1985. Elsevier Science Publishers B. V.

- [BZA94] J. M. Páez Borallo, S. Zazo, and I. A. Pérez Alvarez. "A fully convergent joint blind equalization/phase recovery scheme for data communication systems". In *Proc. Vehicular Technology Conf.*, Stockholm, Sweden, June 1994.
- [CK84] J.M. Cioffi and T. Kailath. "Fast, recursive least squares transversal filters for adaptive filtering". *IEEE Trans. on ASSP*, ASSP-32(2):304–337, April 1984.
- [CMK83] G. Carayannis, D. G. Manolakis, and N. Kalouptsidis. "A Fast Sequential Algorithm for Least-Squares Filtering and Prediction". *IEEE Trans. on Acoustics, Speech, and Signal Processing*, ASSP-31:1394–1402, Dec. 1983.
- [CN93] Y. Chen and C. L. Nikias. "Identifiability of a Band-Limited System from Its Cyclostationary Output Autocorrelation". *IEEE Trans. on Signal Processing*, 42:483–485, Feb. 1993.
- [CNP92] Y. Chen, C. L. Nikias, and J. G. Proakis. "Blind equalization with criterion with memory nonlinearity". *Optical Engineering*, 31(6):1200–1210, June 1992.
- [Cre94] P. M. Crespo. "On the MMSE Behavior of Finite Length Transversal Equalizers". In D. Docampo and A.R. Figueiras, editors, *Proc. Final COST 229 Workshop on Adaptive Systems, Intelligent Approaches, Massively Parallel Computing and Emergent Techniques in Signal Processing and Communications*, pages 103–107, Bayona (Vigo), Spain, Oct. 1994.
- [CS90] C. K. Chan and J. J. Shynk. "Stationary Points of the Constant Modulus Algorithm for Real Gaussian Signals". *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 38(12):2176–2181, Dec. 1990.
- [Din91a] Z. Ding. "Existing Gap between Theory and Application of Blind Equalization". In *Proc. SPIE 1991*, volume 1565, pages 154–165, 1991.
- [Din91b] Z. Ding. "Joint Blind Equalization and Carrier Recovery of QAM Systems". In *Proc. 25th Conf. on Info. Sci. and Systems*, pages 786–797, Baltimore, MD, 1991.
- [Din94] Z. Ding. "Blind Channel Identification and Equalization Using Spectral Correlation Measurements, Part I: Frequency-Domain Analysis". In *Cyclostationarity in Communications and Signal Processing*, editor W. A. Gardner, pages 417–436, New Jersey, USA, 1994.
- [DK94] K. Dogançay and R. A. Kennedy. "A Globally Admissible Off-Line Modulus Restoral Algorithm for Low-Order Adaptive Channel Equalisers". In *Proc. ICASSP 94 Conference*, pages III–61–III–64, Adelaide, Australia, 1994.

- [DKAJ91] Z. Ding, R. A. Kennedy, B. D. O. Anderson, and C. R. Johnson, Jr. "Ill-Convergence of Godard Blind Equalizers in Data Communication Systems". *IEEE Trans. on Communications*, 39(9):1313–1327, Sept. 1991.
- [DKAJ93] Z. Ding, R. A. Kennedy, B. D. O. Anderson, and C. R. Johnson, Jr. "Local convergence of the Sato Blind Equalizer and Generalizations Under Practical Constraints". *IEEE Trans. on Information Theory*, 39(1), Jan. 1993.
- [dMB94] J. B. G. del Moral and E. Biglieri. "Algorithms for Blind Identification of Terrestrial Microwave Channels". In *Proc. COST 229 final conference, tutorial presentation*, Vigo, Spain, Oct. 1994.
- [FB90] M. R. Frater and R. R. Bitmead. "Escape form Local Equilibria in Adaptive Equalizers". In *Proc. ISSPA 90*, Gold Coast, Australia, 1990.
- [FL93] J. Fijalkow and P. Loubaton. "Identification of rank 1 rational spectral densities from noisy observations: A stochastic realization approach". *submitted to System and Control Letters*, 1993.
- [For73] G. D. Forney, Jr. "The Viterbi Algorithm". *Proceedings of the IEEE*, 61(3):268–278, March 1973.
- [Fos85] G. J. Foschini. "Equalizing Without Altering or Detecting Data". *ATT Technical Journal*, 64(8):1885–1911, Oct. 1985.
- [FVM92] J. A. R. Fonollosa, J. Vidal, and A. Moreno. "FIR System Identification using a Linear Combination of Cumulants". In *Proc. ICASSP 92 Conference*, pages V–473–V–476, San Fransisco, CA, March 1992.
- [Gar91] W. A. Gardner. "Exploitation of Spectral Redundancy in Cyclostationary Signals". *IEEE Signal Processing Magazine*, 8(2):14–36, April 1991.
- [Gar94] W. A. Gardner, editor. *Cyclostationarity in Communications and Signal Processing*. IEEE press, New Jersey, USA, 1994.
- [Gay93] S. L. Gay. "A Fast Conergering, Low Complexity Adaptive Filtering Algorithm". In *Proc. Third International Workshop on Acoustic Echo Control*, Plestin les Grèves, France, Sept 7-8 1993.
- [GL89] G.H. Golub and C.F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 1989. 2nd edition.
- [GM89] G. B. Giannakis and J. M. Mendel. "Identification of nonminimum phase systems using higher order statistics". *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 37(3):360–377, March 1989.

- [God80] D. N. Godard. "Self-Recovering Equalization and Carrier Tracking in Two-Dimensional Data Communication Systems". *IEEE Trans. on Communications*, COM-28(11):1867–1875, Nov. 1980.
- [GRS87] R. Gooch, M. Ready, and J. Svoboda. "A Lattice-Based Constant Modulus Adaptive Filter". In *Proc. Asilomar Conference on Signals, Systems, and Computers*, pages 282–286, Pacific Grove, California, 1987.
- [GW92] M. Ghosh and C. L. Weber. "Maximum-likelihood blind equalization". *Optical Engineering*, 31(6):1224–1228, June 1992.
- [Hat91] D. Hatzinakos. Stop-and-go sign algorithms for blind equalization. In *Proc. SPIE*, volume 1565 Adaptive Signal Processing, pages 118–129, 1991.
- [Hat92] D. Hatzinakos. "Blind equalization using stop-and-go adaptation rules". *Optical Engineering*, 31(6):1181–1188, June 1992.
- [Hay91] S. Haykin. *Adaptive Filter Theory*. Prentice-Hall, Englewood Cliffs, NJ, 1991, second edition.
- [HD92a] K. Hilal and P. Duhamel. "A Convergence Study of the Constant Modulus Algorithm Leading to a Normalized-CMA and a BLock-Normalized-CMA". In *Proc. EUSIPCO 92 Conference*, pages 135–138, 1992.
- [HD92b] K. Hilal and P. Duhamel. "A Convergence Study of the Constant Modulus Algorithm Leading to a Normalized-CMA and a BLock-Normalized-CMA". *submitted to IEEE Trans. on Signal Processing*, 1992.
- [HD93] K. Hilal and P. Duhamel. "A Blind Equalizer allowing soft transition between the Constant Modulus and the Decision-Directed Algorithm for PSK Modulated signals". In *Proc. ICC 93 Conference*, 1993.
- [HN91] D. Hatzinakos and C. L. Nikias. "Blind Equalization Using a Tricpestrum-Based Algorithm". *IEEE Trans. on Communications*, 39(5):669–682, May 1991.
- [HPK94] B. Hassibi, A. Paulraj, and T. Kailath. "On a Closed Form Solution to the Constant Modulus Factorization Problem". In *Proc. 28th Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, CA, Oct. 31 - Nov. 2 1994.
- [Hua94] Y. Hua. "Fast Maximum Likelihood for Blind Identification of Multiple FIR Channels". *submitted to IEEE Trans. on Signal Processing*, May 1994.
- [Jab92] N. K. Jablon. "Joint Blind Equalization, Carrier Recovery and Timing Recovery for High-Order QAM Constellations". *IEEE Trans. on Signal Processing*, 40(6):1383–1398, June 1992.

- [JDS88] C. R. Johnson, Jr., S. Dasupta, and W. A. Sethares. "Averaging Analysis of Local Stability of a Real Constant Modulus Algorithm Adaptive Filter". *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 36(6):900–910, June 1988.
- [JL92] C. R. Johnson, Jr. and J. P. LeBlanc. "Toward Operational Guidelines for Memoryless-Error-Function-Style Blind Equalizers". In *Proc. 2nd COST 229 workshop, tutorial presentation*, Bordeaux, France, Oct. 1992.
- [JLK92] C. R. Johnson, Jr., J. P. LeBlanc, and V. Krishnamurthy. "Godard Blind Equalizers Misbehavior with Correlated Sources: Two Examples". In *Moroccan Journal of Control, Computer Science, and Signal Processing*, 1992.
- [JWC92] H. Jamali, S. L. Wood, and R. Cristi. "Experimental Validation of the Kronecker Product Godard Blind Adaptive Algorithms". In *Proc. Asilomar 26th Conference on Signals, Systems, and Computers*, Pacific Grove, California, 1992.
- [KD92] R. A. Kennedy and Z. Ding. "Blind adaptive equalizers for quadrature amplitude modulated communication systems based on convex cost functions". *Optical Engineering*, 31(6):1189–1199, June 1992.
- [Lam94] R. H. Lambert. "BLINDSFTF- Stable Fast Transversal Filters Algorithm Applied to Blind Equalization". In *Proc. 28th Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, CA, Oct. 31 - Nov. 2 1994.
- [LD94] Y. Li and Z. Ding. "Global Convergence of Fractionally Spaced Godard Adaptive Equalizers". In *Proc. 28th Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, CA, Oct. 31 - Nov. 2 1994.
- [LDKJ94] J. P. LeBlanc, K. Dogancay, R. A. Kennedy, and C. R. Johnson, Jr. "Effects of Input Data Correlation on the Convergence of Blind Adaptive Equalizers". In *Proc. ICASSP 94 Conference*, pages III–313–III–316, Adelaide, Australia, 1994.
- [LFHJ95] J. LeBlanc, I. Fijalkow, B. Huber, and R. Johnson, Jr. "Fractionally Spaced CMA Equalizers Under Periodic and Correlated Inputs". In *Proc. ICASSP 95 Conference*, Detroit, Michigan, U.S.A., 1995.
- [Lou94] Ph. Loubaton. "Egalisation autodidacte multi-capteurs et systèmes multivariables". GDR 134 (Signal Processing) working document, February 1994, France, 1994.
- [LT86] M. G. Larimore and J. R. Treichler. "Data Equalization Based on the Constant Modulus Adaptive Filter". In *Proc. ICASSP 86 Conference*, pages 949–952, Tokyo, Japan, 1986.

- [LXT93] H. Liu, G. Xu, and L. Tong. "A Deterministic Approach to Blind Equalization". In *Proc. Asilomar 25th Conference on Signals, Systems, and Computers*, Pacific Grove, California, Nov. 1993.
- [Mak75] J. Makhoul. "Linear Prediction: A Tutorial Review". *Proceedings of the IEEE*, 63(4):561–580, April 1975.
- [May94] S. Mayrargue. "A Blind Spatio-Temporal Equalizer for a Radio-Mobile Channel Using the Constant Modulus Algorithm (CMA)". In *Proc. ICASSP 94 Conference*, Adelaide, Australia, 1994.
- [Maz80] J. E. Mazo. "Analysis of Decision-Directed Equalizer Convergence". *The Bell System Technical Journal*, 59(10):1857–1876, Dec. 1980.
- [MD94] M. Montazeri and P. Duhamel. "A Set of Algorithms Linking NLMS and Block RLS Algorithms". *submitted to IEEE Trans. on Signal Processing*, 1994.
- [MDCM93] E. Moulines, P. Duhamel, J. F. Cardoso, and S. Mayrargue. "Subspace methods for the blind identification of multichannel FIR filters". *submitted to IEEE Trans. on Signal Processing*, 1993.
- [MDCM94] E. Moulines, P. Duhamel, J. F. Cardoso, and S. Mayrargue. "Subspace methods for the blind identification of multichannel FIR filters". In *Proc. ICASSP 94 Conference*, pages IV–573–IV–576, Adelaide, Australia, 1994.
- [MDG+ 94] K. Abed Meraim, P. Duhamel, D. Gesbert, P. Loubaton, S. Mayrargue, E. Moulines, and D. Slock. "Prediction error methods for time-domain blind identification of multichannel FIR filters". *submitted to IEEE Letters on Signal Processing*, 1994.
- [ME84] O. Macchi and E. Eweda. "Convergence Analysis of Self-Adaptive Equalizers". *IEEE Trans. on Information Theory*, IT-30(2):161–176, March 1984.
- [MLM94] K. Abed Meraim, P. Loubaton, and E. Moulines. "A subspace algorithm for certain blind identification problems". *submitted to IEEE Trans. on Information Theory*, July 1994.
- [MS93] T. McWorther and L. L. Scharf. "Cramer-Rao Bounds for Deterministic Modal Analysis". *IEEE Trans. on Signal Processing*, 41:1847–1866, May 1993.
- [MT93] G. V. Moustakides and S. Theodoridis. "Fast Newton Transversal Filters-A New Class of Adaptive Estimation Algorithms". *IEEE Trans. on Signal Processing*, 39:2184–2193, Oct. 1993.

- [Nik88] C. L. Nikias. "Arma bispectrum approach to nonminimum phase system identification". *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 36(4):513–524, April 1988.
- [OU84] K. Ozeki and T. Umeda. "An Adaptive Algorithm Using Orthogonal Projection to an Affine Subspace and its Properties". *Trans. IECE Japan*, J67-A:126–132, 1984.
- [PP87] G. Picchi and G. Prati. "Blind Equalization and Carrier Recovery Using a 'Stop-and-Go' Decision-Directed Algorithm". *IEEE Trans. on Communications*, COM-35(9):877–887, Sept. 1987.
- [Pro89] J.G. Proakis. *Digital Communications*. McGraw-Hill, New York, 1989.
- [PS93a] C.B. Papadias and D.T.M. Slock. "A Bilinear Approach to Constant Modulus Blind Equalization". In *Proc. ATHOS (ESPRIT Basic Research Working Group no 6620) Workshop on "System Identification and High Order Statistics"*, Sophia Antipolis, France, Sept. 20-21 1993.
- [PS93b] C.B. Papadias and D.T.M. Slock. "Normalized Sliding Window Constant Modulus Algorithms for Blind Equalization". In *Proc. Quatorzième Colloque sur le Traitement du Signal et des Images*, pages 507–510, Juan-les-Pins, France, Sept. 13-16 1993.
- [PS93c] C.B. Papadias and D.T.M. Slock. "On the Convergence of Normalized Constant Modulus Algorithms for Blind Equalization". In *Proc. Int'l Conf. on Digital Signal Processing*, pages 245–250, Nicosia, Cyprus, July 14-16 1993.
- [PS94a] C.B. Papadias and D.T.M. Slock. "New Adaptive Blind Equalization Algorithms for Constant Modulus Constellations". In *Proc. Int'l Conf. on Acoustics, Speech and Signal Processing*, Adelaide, South Australia, April 19-22 1994.
- [PS94b] C.B. Papadias and D.T.M. Slock. "On the Decision-Directed Equalization of Constant Modulus Signals". In *Proc. 28th Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, CA, Oct. 31 - Nov. 2 1994.
- [PS94c] C.B. Papadias and D.T.M. Slock. "Towards Globally Convergent Blind Equalization of Constant Modulus Signals: a Bilinear Approach". In *Proc. VII European Signal Processing Conference*, Edinburgh, Scotland, Sept. 13-16 1994.
- [PS95] C.B. Papadias and D.T.M. Slock. "Normalized Sliding Window Constant Modulus Algorithms for Blind Equalization: a Link Between Blind Equalization and Classical Adaptive Filtering". *submitted to IEEE Trans. on Signal Processing*, 1995.

- [QH94] W. Qui and Y. Hua. "Performance Analysis of a Blind Channel Identification Method Based on Second Order Statistics". In *Proc. 28th Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, CA, Oct. 31 - Nov. 2 1994.
- [Sat75] Y. Sato. "A Method of Self-Recovering Equalization for Multilevel Amplitude-Modulation Systems". *IEEE Trans. on Communications*, COM-23:679-682, June 1975.
- [Sch91] L.L. Scharf. *Statistical Signal Processing*. Addison-Wesley, Reading, MA, 1991.
- [SGKC91] J. J. Shynk, R. P. Gooch, G. Krishnamurthy, and C. K. Chan. "A Comparative Performance Study of Several Blind Equalization Algorithms". In *Proc. SPIE 1991*, volume 1565, pages 102-117, 1991.
- [SK91] D.T.M. Slock and T. Kailath. "Numerically Stable Fast Transversal Filters for Recursive Least-Squares Adaptive Filtering". *IEEE Trans. Signal Proc.*, ASSP-39(1):92-114, Jan. 1991.
- [Slo89] D.T.M. Slock. *Fast algorithms for Fixed-Order Recursive Least-Squares Parameter Estimation*. PhD thesis, Stanford University, Stanford, CA, USA, September 1989.
- [Slo92a] D.T.M. Slock. "On the Normalization of the Step Size in Nonsymmetric Stochastic Gradient Algorithms". In *Proc. 26th Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, CA, Oct. 26-28 1992.
- [Slo92b] D.T.M. Slock. "The Block Underdetermined Covariance (BUC) Fast Transversal Filter (FTF) Algorithm for Adaptive Filtering". In *Proc. 26th Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, CA, Oct. 26-28 1992.
- [Slo92c] D.T.M. Slock. "Underdetermined Growing and Sliding Window Covariance Fast Transversal Filter RLS Algorithms". In *Proc. EUSIPCO 92, Vith European Signal Processing Conference*, pages 1169-1172, Brussels, Belgium, Aug. 24-27 1992.
- [Slo93] D.T.M. Slock. "On the Convergence Behavior of the LMS and the Normalized LMS Algorithms". *IEEE Trans. on Signal Processing*, 41(9):2811-2825, Sept. 1993.
- [Slo94a] D.T.M. Slock. "Blind Fractionally-Spaced Equalization, Perfect-Reconstruction Filter Banks and Multichannel Linear Prediction". In *Proc. ICASSP 94 Conf.*, Adelaide, Australia, April 1994.
- [Slo94b] D.T.M. Slock. "Blind Joint Equalization of Multiple Synchronous Mobile Users Using Oversampling and/or Multiple Antennas". In *Proc. 28th Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, CA, Oct. 31 - Nov. 2 1994.

- [Slo94c] D.T.M. Slock. "Subspace Techniques in Blind Mobile Radio Channel Identification and Equalization using Fractional Spacing and/or Multiple Antennas". In *Proc. 3rd International Workshop on SVD and Signal Processing*, Leuven, Belgium, Aug. 22-25 1994.
- [SP94a] D.T.M. Slock and C.B. Papadias. "Blind Fractionally-Spaced Equalization Based on Cyclostationarity". In *Proc. Vehicular Technology Conf.*, Stockholm, Sweden, June 1994.
- [SP94b] D.T.M. Slock and C.B. Papadias. "Blind Radio Channel Identification and Equalization based on Oversampling and/or Antenna Arrays". In D. Docampo and A.R. Figueiras, editors, *Proc. Final COST 229 Workshop on Adaptive Systems, Intelligent Approaches, Massively Parallel Computing and Emergent Techniques in Signal Processing and Communications*, pages 103-107, Bayona (Vigo), Spain, Oct. 1994.
- [SP95] D.T.M. Slock and C.B. Papadias. "Further Results on Blind Identification and Equalization of Multiple FIR Channels". In *Proc. ICASSP 95 Conf.*, Detroit, Michigan, May 1995.
- [SW90] O. Shalvi and E. Weinstein. "New Criteria for Blind Deconvolution of Non-Minimum Phase Systems". *IEEE Trans. on Information Theory*, 36:312-321, March 1990.
- [SW93] O. Shalvi and E. Weinstein. "Super-Exponential Methods for Blind Deconvolution". *IEEE Trans. on Information Theory*, 39:504-519, March 1993.
- [TA83] J. R. Treichler and B. G. Agee. "A New Approach to Multipath Correction of Constant Modulus Signals". *IEEE Trans. on Acoustics, Speech, and Signal Processing*, ASSP-31(2):459-472, April 1983.
- [TL85] J. R. Treichler and M. G. Larimore. "New Processing Techniques Based on the Constant Modulus Adaptive Algorithm". *IEEE Trans. on Acoustics, Speech, and Signal Processing*, ASSP-33(2):420-431, April 1985.
- [TP94] S. Talwar and A. Paulraj. "Performance Analysis of Blind Digital Signal Copy Algorithms". In *Proc. 28th Asilomar Conf. on Signals, Systems and Computers*, 1994.
- [Tug92] J. K. Tugnait. "Comments on 'New Criteria for Blind Deconvolution of Nonminimum Phase Systems Phase Systems (Channels)'" ". *IEEE Trans. on Information Theory*, 38(38):210-213, Jan. 1992.
- [TVP94] S. Talwar, M. Viberg, and A. Paulraj. "Blind Estimation of Multiple Co-Channel Digital Signals Using an Antenna Array". *IEEE Signal Processing Letters*, SPL-1(2):29-31, Feb. 1994.

- [TXK94] L. Tong, G. Xhu, and T. Kailath. "Blind Identification and Equalization of Multipath Channels: A Time Domain Approach". *IEEE Trans. on Information Theory*, 40(2):340–349, March 1994.
- [Ung76] G. Ungerboeck. "Fractional Tap-Spacing Equalizer and Consequences for Clock Recovery in Data Modems". *IEEE Trans. on Communications*, COM-24(8):856–864, Aug. 1976.
- [VAK91] S. Verdú, B. D. O. Anderson, and R. A. Kennedy. "Anchored Blind Equalization". In *Proc. 25th Conf. on Info. Sci. and Systems*, pages 774–779, Baltimore, MD, 1991.
- [VAK93] S. Verdú, B. D. O. Anderson, and R. A. Kennedy. "Blind Equalization Without Gain Identification". *IEEE Trans. on Information Theory*, 39(1), Jan. 1993.
- [VP94] A. Van der veen and A. Paulraj. "A constant modulus factorization technique for smart antenna application in mobile communications". In *Proc. SPIE 1994*, volume 2296, 1994.
- [WK94] V. Weerackody and S. Kassam. "Dual-Mode Type Algorithms for Blind Equalization". *IEEE Trans. on Communications*, 42(1):22–28, Jan. 1994.
- [YP93] D. Yellin and B. Porat. "Blind Identification of FIR Systems Excited by Discrete-Alphabet Inputs". *IEEE Trans. on Signal Processing*, 41:1331–1339, March 1993.
- [ZC92] R. A. Ziegler and J. M. Cioffi. "Estimation of time-varying Digital Radio Channels". *IEEE Trans. on Vehicular Technology*, 41(2):134–151, May 1992.
- [ZMM93] F. Zheng, S. McLaughlin, and B. Mulgrew. "Cumulant-based deconvolution and identification: Several new families of linear equations". *Signal Processing*, 30:199–219, 1993.
- [ZPV91] E. Zervas, J. Proakis, and Eyuboglu V. "Effects of Constellation Shaping on Blind Equalization". In *Proc. SPIE 1991*, volume 1565, 1991.