Tampereen teknillinen korkeakoulu
Julkaisuja 284

Tampere University of Technology
Publications 284

Corneliu Rusu

Several Approaches to Cost Function Adaptation and Phase Approximation

Tampere 2000

Tampereen teknillinen korkeakoulu
Julkaisuja 284

Tampere University of Technology
Publications 284

Corneliu Rusu

Several Approaches to Cost Function Adaptation and Phase Approximation

Thesis for the degree of Doctor of Technology to be presented with due permission for public examination and criticism in Auditorium HB116, Hermitec at Tampere University of Technology, on the 4th of February 2000, at 12 o'clock noon.

Tampere 2000

ISBN 952-15-0341-6 (printed)
ISBN 952-15-1827-8 (PDF)
ISSN 0356-4940

TTKK- PAINO, Tampere 2000

Abstract

In the particular application of data echo cancellation we have a signal environment which violates basic assumptions inherent in adaptive estimation theory. First, the training signal is noise dominated (by the far end signal), and second, the noise statistics are non-Gaussian. The second fact tends to suggest that least squares optimisation is not optimal for this case. Previous work has shown that considerable performance enhancement is possible by the use of high order metrics (of order greater than 3). However, the exact metric to be applied depends on the channel dispersion. Therefore the logical solution is to use an adaptive error metric which changes dependent on the actual error statistics. As in the case of variable step-size algorithms, the optimum cost function adaptation algorithm requires complete knowledge of the signal and system statistics and thus it cannot be implemented in most practical conditions. In the first part of this thesis, the solution for this kind of situation is examined. The main goal is to derive a stochastic gradient algorithm for the case of data echo cancellation in which the error exponent is adjusted using the value of the error.

Two main classes of cost function adaptation approaches are derived: nonrecursive and recursive. Nonrecursive adaptation of cost function can be distinguished by the fact that the error exponent is updated every iteration using only direct relationships between an approximation of the error and the power of the cost function. Alternatively, in recursive cost function adaptation, we do not need anymore to estimate the actual error of the system, the updated error exponent is computed recursively.

The idea behind the first approach relies on adjusting the error exponent parameter during the adaptation by enforcing same value of gradients for two consecutive error exponents. Two different type of algorithms are proposed: the staircase and the smooth CFA (Cost Function Adaptation) algorithms, and it is shown that they behave with comparable results.

Then the gradient approach is derived by enforcing the same direction of the gradient as in the case of non-quadratic algorithms. As a special case the linear adaptation of the power of the cost function results.

In the recursive cost function adaptation case, the new error exponent is computed from the previous one using a customary LMS (Least Mean Square) recursive equation. This method improves the sensitivity of the power of the cost function with respect to the noisy error, while the other benefits of the CFA algorithms in terms of the convergence speed and residual error remain.

The second part of the thesis deals with a special form of Hilbert transform, known as Bode gain-phase relationships. Here the interest is to develop formulae for phase approximations by gain samples, equally separated in logarithmic domain. Our approach consists of the following steps:

First we discuss the most important theoretical results related to our specific subject. Sampling in logarithmic domain is pointed out as a more general rule, not necessarily only for the presented case.

Then we establish new relationship for computing the phase of the minimum-phase functions from the gain derivatives, as a first advance to approach phase by gain samples. As a beginning, we show that for a given frequency the phase could be obtained from the odd derivatives of the neperian gain, evaluated for this frequency. Then we select a finite number of terms of the main formula and we derive an approximation of phase. We show that the approximations derived can be improved by taking into account the Gibbs phenomenon and the Feher kernel.

Finally we derive a completely novel relationship for approximating the phase values from the gain samples, in nepers, equally spaced in the logarithmic frequency domain. A general approximation formula is proved, then two quadrature formulae are obtained using Newton-Cotes and Simpson rules. Numerical examples are also provided to emphasize the achievements of the method.

Preface

This thesis was carried out during the period August 1997 to September 1999, at the Signal Processing Laboratory, Tampere University of Technology, Finland. The first part of the dissertation was initiated during my visit at Loughborough University of Technology in Spring 1996.

I express my gratitude to my thesis supervisors Prof. Colin F. N. Cowan and Prof. Pauli Kuosmanen for the guidance I received during this work and for their active participation in making this thesis better. I also thank them for their generous help in several important moments.

I would like also to thank for the entire support to Prof. Serban Lungu, the advisor of my Ph.D. thesis in Electronics.

I am very grateful to Prof. Jaakko Astola and the staff of the Tampere International Center for Signal Processing for the opportunity to attend their excellent seminars and workshops, and to have interesting discussions about topics related to this dissertation with Prof. Edward J. Delp, Prof. Edward R. Dougherty, Prof. Robert A. Gonsalves and Prof. Corneliu Popeea.

I wish to thank the people of Signal Processing Laboratory for this special atmosphere where it is a pleasure to work. Especially I would like to mention Prof. Tapio Saramäki for the very interesting scientific talks, and the last ICASSP team for traveling together.

Special thanks are due to my co-authors: Prof. Ioan Gavrea, Adrian Burian, Dr. Tommy K. Bysted, Dr. Mika Helsingius, Dr. Heikki Huttunen and Marius Tico. I would like to thank also Dr. Sangarapillai Lambotharan and Dr. Vladimir Melnyk for presenting my work at IEEE DSP'98 and FINSIG'99 workshops.

While I was working in these topics, I received valuable information from Prof. Brian D. O. Anderson, Prof. Jonathon A. Chambers, Prof. Monson H. Hayes, Dr. Shin'ichi Koike, Dr. Thomas Strohmer, Dr. Oguz Tanrikulu, and Dr. Azzedine Zerguine.

I am gratefully acknowledged the financial support received before 1997 from the organizers of the following DSP events: COST#229 Workshops in 1994 from Ljubljana (Slovenia) and Bayona (Spain), IWAENC95 Røros (Norway), DSP95 Limassol (Cyprus),

and EUSIPCO'96 Trieste (Italy).

I thank the Romanian community in Tampere, especially those who generously offered me help at the beginning: Maria and Ion Crişan, Doina Petrescu, Bogdan Cramariuc, Octavian Sarca and Ioan Tăbuş.

I wish to thank my parents Cornelia and Gheorghe, and my brother Veronel for everything they have done for me.

My warmest thanks go to my dear wife Lucia for her never ending support, patience and love.

This thesis is dedicated to my daughter Ioana-Elena, and those who saved her life in Summer 1994.

Tampere, September 1999

Corneliu Rusu

Contents

List of Figures

List of Tables

Abbreviations

CFA Cost Function Adaptation, 2

CVSS Convex Variable Step-Size, 97

EEAS Error Exponent Adaptation Subsystem, 71

EPIS Echo-Path Identification System, 69

FIR Finite Impulse Response, 15

GAS Gradient Adaptive Step Algorithm, 9

LCFA Linear Cost Function Adaptation, 14

LMFSF The Threshold Technique with LMS and LMF, 106

LMF Least Mean Fourth, 2

LMMN Least-Mean Mixed Norm, 11

LMP Least Mean P-Power Algorithm, 8

LMS+F LMS+F Adaptive Algorithm, 11

LMS-LMF Mixed Controlled Norm, 11

LMS/F Combined LMS/F Algorithm, 11

LMSEN Least Mean Switched Error Norm, 109

LMS Least Mean Square, 1

NQSG Non-Quadratic Stochastic Gradient Algorithm, 9

RCFA Recursive Cost Function Adaptation, 2

RLS Recursive Least Squares, 8

VSS Variable Step-Size Algorithm, 9

VS Variable Step Algorithm, 9

Notations

List of symbols

Part I

r	constant error exponent
$r(k)$	error exponent sequence
J_r	cost function
$e(k)$	error signal
$\bar{e}(k)$	finite-time average of the error signal
α	constant used in CFA definition
F	function used to define CFA
$x(k)$	input signal
$y(k)$	echo-path output signal
$\hat{y}(k)$	synthetic echo signal
$\boldsymbol{x}(k)$	the transpose of the input observations vector
$\boldsymbol{y}(k)$	the transpose vector of echo-path output signal
$\hat{\boldsymbol{y}}(k)$	the transpose vector of synthetic echo signal
$f(k)$	attenuated far-end signal
$\hat{\boldsymbol{h}}(k)$	transpose of the vector of echo-path filter coefficients
$\boldsymbol{h}(k)$	transpose of the vector of estimated filter coefficients
N	number of filter coefficients
μ	constant step-size
$\mu(k)$	variable step-size
μ_{max}	maximum step-size for LMS
μ_{min}	minimum step-size for variable step-size algorithms
RE_{dB}	arbitrary constant used in first CFA algorithm [21]
$H(z)$	transfer function

p	pol of single pol single zero digital filter
A	attenuation at the N-th sample
$R_i, i = \overline{1, \mathrm{P}}$	error exponent constants used in staircase CFA algorithm
$E_i, i = \overline{1, \mathrm{P}}$	error modulus constants used in staircase CFA algorithm
E_0	initial error modulus constant used in smooth CFA algorithm
E_∞	final error modulus constant used in smooth CFA algorithm
R_0	initial error exponent constant used in smooth CFA algorithm
R_∞	final error exponent constant used in smooth CFA algorithm
E_M	constant used in simulations
R	error exponent constant used in simulations
K	instant of time used in Example 3
p_k	normalised tap-error vector sequence
a	attenuation of the far-end signal
σ	variance of Gaussian interference
L	length of the time-average window in smooth CFA algorithm
	or time-constant of digital filter in gradient approach
W	sampling period of the error exponent in smooth CFA algorithm
$\mathcal{K}$	kernel of gradient CFA
β	CFA constant used in gradient approach
$S^{r(k+1)}_{\lvert e(k)\rvert_{dB}}$	sensitivity of $r(k+1)$ with respect to $\lvert e(k)\rvert_{dB}$
$\hat{e}(k)$	output of the recursive digital filter in gradient CFA
r_{min}	minimum error exponent
r_{max}	maximum error exponent
$\varphi(k)$	attenuation in EEAS subsystem
$\epsilon(k)$	error signal in EEAS subsystem
ρ	step-size of EEAS subsystem
ϱ	step-size in analytic approach
r_-	minorant of error exponent sequence
r_+	majorant of error exponent sequence
ϕ_{xx}	autocorrelation matrix of the input sequence
λ	maximum value of ϕ_{xx}
$\mathbf{\Delta}$, $\mathbf{\Gamma}$, $\boldsymbol{A}$	auxiliary matrixes used in convergence analysis
δ_i	leading diagonal elements of $\mathbf{\Delta}$
δ	maximum of δ_i
T_i	time constants in convergence analysis
$\mathcal{M}$	misadjustment

Part II

$r(\tau)$	autocorrelation function
$\alpha(\omega)$	gain
$\beta(\omega)$	phase
$A(u)$	log-log gain
$B(u)$	log-log phase
ω_c	normalised frequency
Δ	sampling period in logarithmic sampling

Mathematical Notations

$\mathbb{N}$	the set of positive integer numbers
$\mathbb{Z}$	the set of integer numbers
$\mathbb{R}$	the set of real numbers
$\mathbb{R}_+$	the set of real positive numbers
$\mathbb{C}$	the set of complex numbers
sgn	signum function
ln	natural logarithm
log	logarithm function
E	expectation operator
tr	trace of a matrix
$\mathbf{\Delta}$	identity matrix
lim sup	limit superior
B_n	Bernoulli numbers of order n
$S_j^{(k)}$	Stirling numbers of the first kind
$\overline{\Delta}^i$	finite difference of order i
$\|.\|$	norm
L^1	norm
L^2	norm
L^∞	norm

Chapter 1

Introduction

1.1 Overview of the Thesis

Telecommunications is a growing and changing industry which has proved to be fertile ground for the application of signal processing [28]. The rapid growth in data communications has created a need for adaptive filtering to overcome impairments inherent in existing telephony networks. Although other structures and algorithms have been investigated for telecommunications applications, they are of less importance compared to transversal filters adjusted using stochastic gradient algorithms and its variants [20]. In the particular application of data echo cancellation a Hilbert transformer is very often used in implementation in order to generate the complex error signal [82]. The present dissertation deals with aspects concerning these two theoretic problems: stochastic gradient algorithms and Hilbert transform, and for each of these topics we assign a part of the thesis.

The first part of this work is dedicated to the introduction and development of the cost function adaptation for data echo cancellation. In this framework the adaptive Least Mean Square (LMS) algorithm has received a great deal of attention during the last decades, and it has been used in many applications due to its simplicity and relatively well-behaved performance. However, the convergence speed to optimal filter coefficients is relatively slow. This can be a drawback in the case of the digital echo cancellation, where one of the goals is to reduce the adaptation time, during which transmission of useful data is not possible.

Chapter 2 gives background on main adaptive filtering techniques used in data echo cancellation. First quadratic and nonquadratic algorithms are discussed, then the most important variable step-size algorithms are recalled. A short description of combined

LMS and Least Mean Fourth (LMF) methods follows, and the important Cost Function Adaptation (CFA) issue is pointed out in details. Before describing the already proposed CFA algorithms, an *a priori* comparative study between variable step-size methods and cost function adaptation algorithms is developed. Modelling of the echo canceller and of the signals ends this Chapter.

Chapter 3 considers the important family of nonrecursive cost function adaptation algorithms. The idea behind the first approach (Section 3.1) relies on adjusting the error exponent parameter during the adaptation by enforcing same value of gradients for two consecutive error exponents. Two different types of algorithms are proposed: the staircase and the smooth CFA algorithms, and it is shown that they behave with comparable results. The second main method (Section 3.2) is called the gradient approach. It was derived by enforcing the same direction of the gradient as in the case of non-quadratic algorithms. As a special case the linear adaptation of power of the cost function results.

Chapter 4 deals with recursive cost function approaches. The derivation of this type of algorithm does not use an estimator of the instantaneous error as the previous CFA algorithms did. In the RCFA (Recursive Cost Function Adaptation) case, the new error exponent is computing from the previous one using an usual LMS recursive equation. The proposed method improves the sensitivity of the power of the cost function with respect to the noisy error, while the other benefits of the CFA algorithms in terms of the convergence speed and residual error remain.

Chapter 5 analyses the convergence and steady-state properties of the proposed algorithms, and closed-form expressions for the step-size bounds and misadjustment are obtained.

Chapter 6 ends the first part of the thesis by pointing out two close related techniques to cost function adaptation family. First the convex variable step-size algorithm is introduced, for which the convexity of the resulting cost function is guaranteed. Then a threshold technique using quadratic algorithms is proposed. It is based on a comparison of an error estimate with some selected thresholds, and after that to decide whether LMS or LMF should be applied.

The second part of the thesis approaches phase approximation using logarithmic gain. This a special case of a more general topic. A large number of scientific problems deals with Hilbert transform and their sampled derivations. In certain applications the domain is restricted, or other specific conditions are imposed. Nevertheless, some particular problems are encountered almost in every situation. A critical issue is related to the singularities involved in the Hilbert transform computation, since we are confronted with an improper integral. If the integral cannot be evaluated in closed form, as it is the case

with discrete input data, numerical integration is in general complicated. Another inconvenience is related to the properties of algorithms when there is noise present. As it is known, the Hilbert transform can behave as an unbounded operator. Our approach to develop phase approximations by gain samples consists of the following three steps.

Chapter 7 discusses the most important theoretical results related to our specific subject. The phase retrieval problem is recalled, then the old fashion Bode gain-phase relations are reconsidered for our purposes. Sampling in logarithmic domain is pointed out as a more general rule, not necessarily only for the presented case. The issue addressed is described, and a systolic implementation for this problem is proposed.

Chapter 8 establishes new relationship for computing the phase of the minimum-phase functions from the gain derivatives, as a first step to approach phase by gain samples. As a beginning, we show that for a given frequency the phase could be obtained from the odd derivatives of the neperian gain, evaluated for this frequency. Then we select a finite number of terms of the main formula and we derive an approximation of phase. We compute this approximation for first-order and second-order systems, and we emphasize the issue of higher-oder derivatives majorants. We show that the approximations derived can be improved by taking into account the Gibbs phenomenon and the Feher kernel. Finally we use finite differences in order to substitute the higher derivatives involved in the proposed approaches.

Chapter 9 derives completely new relationships for approximating the phase values from the gain samples, in nepers, equally spaced in the logarithmic frequency domain. First a general approximation formula is proved, then two quadrature formulae are derived using Newton-Cotes and Simpson rules. Finally some numerical examples are provided.

Some concluding remarks and several considerations for the future developments are also given.

1.2 Author's Contribution

The author's contribution to the existing theory is mainly in Chapters 3-6, 8, 9. To the author's knowledge no work has been done before in cost function adaptation, at least in the signal processing area and for the case of adaptive filters. In addition to that, various novel approaches to phase approximations are developed. The structure of the thesis is constructed to follow easily these results.

The main contribution of this thesis is in the following points:

1. Derivation of stationary cost function adaptation approaches: the staircase and

smooth CFA algorithms; topics within Section 3.1 have been published at *Fourth IMA Int. Conf. Math. for Sign. Proc.* [21] and presented at *ICASSP'98* [22].

2. The gradient approach; this part has been submitted for publication in *IEE.Proc.-Vis.Image Signal Process.* [60], and the linear cost function adaptation has been published in *IEEE DSP'98* [59].

3. The recursive cost function adaptation, presented at *ICASSP'99* [61].

4. Several closed formulae for CFA convergence and steady-state analysis, accepted at *ISCAS 2000* [62].

5. Derivation of the convex variable step-size method.

6. The threshold LMS and LMF technique, presented at *Cost#254 Workshop* [65] and *ECCTD'99* [57].

7. A gain derivatives series formula for phase, published partially at *ICECS'96* [63] and in details at *ICECS'98* [64].

8. The Feher kernel effect in phase approximation, published at *ICECS'98* [64].

9. A phase approximation from gain samples, published at *ISCAS'99* [66].

The author has done the basic derivation, experimental and writing work in all these publications. Except [63, 64] where Prof. Ioan Gavrea also contributed, the author fulfilled the publications task with the supervisors. Other results related with one part or another of the thesis deal with median algorithms [12, 13, 14, 37, 58], non-minimum 1-D phase retrieval [11, 57], and curves fitting [80].

Part I

Cost Function Adaptation for Data Echo Cancellation

Chapter 2

Adaptive Algorithms for Data Echo Cancellation

For a digital echo canceller it is desirable to reduce the adaptation time, during which the transmission of useful data is not possible. Least Mean Square is a non-optimal algorithm in this case as the signals involved are statistically non-Gaussian. Walach and Widrow [81] investigated the use of an error power of 4, while other research established algorithms with arbitrary integer [52] or non-quadratic error power [71]. This dissertation suggests that continuous and automatic, adaptation of the error exponent gives a more satisfactory result. The family of CFA stochastic gradient algorithm proposed allows an increase in convergence rate and an improvement of residual error.

2.1 Data Echo Cancellation

In full-duplex transmission over telephone lines, hybrid couplers are typically used to divide the transmissions in different directions. Due to impedance mismatch at the hybrids, the resulting echoes interfere with the information-bearing signals [85]. An echo canceller is employed to generate an accurate replica of the echoes and subtract them from the received signal. Although basically voice and data echo cancellers are similar, there are some important differences imposed by the type of the signals involved. Unlike voice echo cancellers, the data echo canceller is usually located at each end of the circuit. In addition due to its placing at the data equipment, in the case of data transmission the filter is split into two adjustable transversal filters separated by a bulk delay. The third difference is in the properties of the signals used, speech and digital data. The statistical properties of the speech signals are complicated and difficult to quantify, whereas data signals have

much simpler statistical properties. Finally, in the case of human speech a small amount of echo or an echo with a small delay can be tolerated, but digital data transmission is a different case where even a small amount of echo with a very short delay is intolerable. In this last case an important goal is a short adaptation time, during which the transmission of useful data is not possible.

Two techniques dominate this area, namely, the LMS algorithm and the recursive least squares (RLS) algorithm [1]. The RLS algorithm converges faster than LMS, but it is computationally complex compared to the LMS algorithm especially in the presence of multiple echoes and long bulk delays. In the case of data modems, the LMS algorithm is known to track remarkably well the variations of a slowly time-varying model. However, the conventional LMS algorithm is slow-converging during initialization. One solution to this problem is to use different adaptation techniques during initialization and the steady state.

2.2 Adaptive Algorithms

2.2.1 Quadratic and nonquadratic cost functions

Among various stochastic gradient adaptation algorithms based on different cost functions $J_r = E[|e(k)|^r]$, the least mean square ($r = 2$) algorithm is the most popular and widely used because of its simplicity and robustness to numerical error accumulation. Since its introduction [83], the LMS algorithm has been the focus of many studies, leading to its implementation in many applications. However, LMS has two main disadvantages: a sensitivity to impulse interference and a relatively slow convergence. As a result, researchers have looked for alternative means to improve its performance. Order-statistics operators [6] improve in a significant way the LMS behaviour in noise environments [12, 13, 14, 33, 58, 69, 86]. Many other adaptive algorithms based upon non-mean-square error cost functions can also be chosen to increase the speed of convergence. Walach and Widrow have studied the error minimization in the mean fourth, mean sixth etc. sense ($r = 2p, p \in \mathbb{N}$), as other alternative cost functions. As a special case the LMF algorithm results [81]. More recently, adaptive filtering algorithms that are based on positive integer error exponent LMP (Least Mean P-Power Algorithm) are proposed and their convergence properties have been investigated [52][1]. Unfortunately,

[1]We note that the LMP algorithm with $P < 2$ was used before in the case of α-stable processes [72]. The least squares cost function cannot be defined for such signals because the variance of the error is not finite [5].

these higher-order algorithms have stability problems and they are relatively sensitive to noise due to the very large gradient terms which result for higher-order representation of errors. Other research studies established algorithms with arbitrary non-quadratic error power r (NQSGr) [71], and their results indicated that the error exponent r ($2 < r < 3$) improves stability. In the case of NQSGr the simulations indicate also that a reduction in convergence time can be achieved compared with the LMS algorithm. The powers of the cost function which are in general greater than 3 do not significantly increase the convergence rates. There is a range of exponent r for which convergence rates are less sensitive to the variations of r.

Based on the cost function $J_r = E[|e(k)|^r]$, the general form of the stochastic gradient algorithm with exponent r can be computed using the simple recursive relation below:

$$\hat{\boldsymbol{h}}(k+1) = \hat{\boldsymbol{h}}(k) + \mu r|e(k)|^{r-2}e(k)\boldsymbol{x}(k). \tag{2.1}$$

A review of Equation (2.1) for general case ($e(k) \in \mathbb{C}$, and $r \in \mathbb{R}$) is included in Appendix 2.4.

2.2.2 Variable step-size algorithms

The convergence speed and misadjustment of the LMS algorithm are both dependent on to the step-size μ, so a trade-off between these characteristics exists. The variable step-size algorithms, children of LMS family, use a step-size $\mu(k)$ which is larger at the beginning of adaptation for fast convergence and is smaller at the end of adaptation for smaller misadjustment.

There is a large literature on variable step-size methods. A number of references deal with algorithms that vary the step-size as $1/k$ slowly decreasing to zero over time, but such a procedure is unfortunately not suitable for a transmission system [45]. The popular gear-shifting approach [85] is based on using large step-size values when the algorithm is far from the optimal solution, and small step-sizes values near the optimum. Other variable step (VS) adaptive filtering algorithm controls the step-size by examining the polarity of successive samples of the estimation errors [44]. If there is a given number of consecutive sign changes, the algorithm decreases the step-size by an appropriate amount, whereas if there is another certain number of consecutive sign changes, the algorithm increases the step-size. Based on the fluctuation of the prediction squared error, another proposed alternative is the variable step-size algorithm (VSS) [42]. A different technique [46, 74], usually called gradient adaptive step (GAS) employs gradient adaptation for the step-sizes as well as for the coefficients.

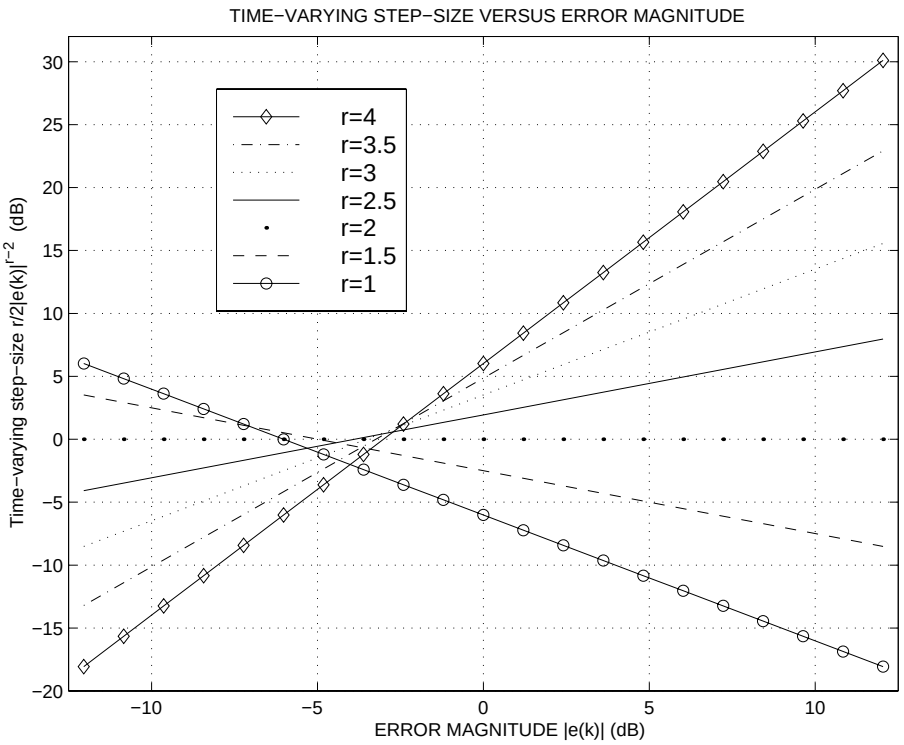


Figure 2.1: The time-varying step-size and the magnitude of the error

Except this last stochastic algorithm, none of the above mentioned variable step-size algorithms assures that the step-size is optimum at any time instant along the convergence process [40]. Moreover, they are based on some linguistic rules on step-size adjustment translated into numerical formulae. An overall weakness of the variable step-size algorithms is that they require the user to select additional constants and initial step-size to control the adaptive behaviour of the step-size sequence. In addition, these methods need steps to be taken to anticipate the step-sizes from exceeding their maximum and minimum limits [29].

We may also consider the higher-order algorithms as the LMS algorithm with time-varying step-size [52]:

$$\mu(k) = \frac{\mu r |e(k)|^{r-2}}{2}. \tag{2.2}$$

Figure 2.1 illustrates the relation between the normalised step-size $\mu(k)/\mu$ and the error magnitude. It is easy to see that the time-varying step-size decays as error decreases for $r < 2$, which suggests that these error exponents should not be applied. Furthermore, caution should be taken when we use higher powers for large errors. All the plots cross

the LMS line almost in the same point, around -5 dB:

$$|e|_{dB}(r) = \frac{6.02 - r_{dB}}{r - 2}.$$

2.2.3 Combined LMS and LMF methods

From the very beginning, it was shown that the LMF algorithm, under some circumstances will have a substantially lower weight noise than the LMS algorithm, if both are set to have the same time constants for the weight relaxation process [81]. Though LMF can outperform LMS in certain situations, it is restrained by the difficulty of setting a stable step-size parameter. As a consequence, recent researches were concentrated in differently composed LMS and LMF techniques. Most of them used a time-varying combination of the LMF and LMS cost functions. A mixed-norm LMMN (Least-Mean Mixed Norm) cost function of this type was first proposed in [15], where a constant mixing parameter was used. The resulting cost function is convex, perhaps the only one for which this property is obvious to prove. In [51] the mixing-parameter was adaptively adjusted and the authors claimed an adding flexibility of their LMS+F algorithm. A switched error norm was proposed in [73], and the mixed controlled norm LMS-LMF algorithm [89] was suggested for long cancellers[2]. In [44], the authors used the LMS with large step-size and the LMF with small step-size, together with a positive threshold in the gradient expression, resulting the Combined LMS/F algorithm. Recently [57, 65] a joined LMS and LMF threshold technique for data echo cancellation has been proposed (Section 6.2).

As a general characteristic all these novel algorithms improve the performances of the quadratic algorithms in their appropriate application, but more or less difficulties appear when we skip to another situation. The issue of parameter selection encountered in the case of variable step-size methods is still challenging. To the author's knowledge, except for LMMN [43, 78] no detailed convergence and steady-analysis has been performed.

2.2.4 The CFA issue

In the particular application of data echo cancellation we have a signal environment which violates basic assumptions inherent in adaptive estimation theory. First, the training signal is noise dominated (by the far-end signal) and second, the noise statistics are non-Gaussian. The second fact tends to suggest that least squares optimisation is not optimal for this case. Previous work [70, 71] has shown that considerable performance enhancement is possible by the use of high order metrics (of order greater than 3). However,

[2]In this particular application, two other techniques were reported [88, 90].

the exact metric to be applied depends on the channel dispersion. Therefore the logical solution is to use an adaptive error metric which changes dependent on the actual error statistics [18]. However, as in the case of variable step-size algorithms [25], the optimum CFA algorithm requires complete knowledge of the signal and system statistics and thus it cannot be implemented in most practical conditions. In this thesis, the solution for this kind of situation is examined. The main goal of our task is to derive a stochastic gradient algorithm for the case of data echo cancellation in which the error exponent r is adjusted using the value of the error $e(k)$ [19], i.e. to find a relation of the form

$$r(k+1) = r(k) + \alpha F[r(k), e(k)], \tag{2.3}$$

where α is a constant and F is a function to be found. The weights would be updated following formula (2.1) with $r = r(k)$. It is supposed also that the algorithm will converge on a steady-state cost function $J_r = E[|e(k)|^{r(k)}]$.

For the sake of clarity we shall consider in this dissertation two different types of error exponent updates. First we discuss adjustments without memory. In this case we have nonrecursive cost function adaptation (Chapter 3). Thus relationship (2.3) becomes $r(k+1) \equiv F[e(k)]$. Alternatively, the recursive cost function adaptation case is detailed in Chapter 4, where the adaptation of the cost function is with memory.

It will be shown that the adaptation of cost function is a promising alternative, since continuous and automatic adjustment of the error exponent gives a more satisfactory result compared to the LMS and the LMF algorithms. The new family of stochastic gradient algorithms allows an increase in convergence rate and, at the same time, an improvement of the residual error [21, 22, 59, 61].

2.2.5 Cost function adaptation versus variable step-size

Before proceeding further we have to point out that dealing with variable error exponents seems more difficult than using variable step-sizes. A first reason is given by the nonlinearity of the exponential operator, compared with step-size linearity involvement. Moreover, the multiplications by a constant of the step-size or of the error exponent have completely different result and significance. Also we have to note the specific point $|e(k)| = 1$. In this case the cost functions $J_r = E[|e(k)|^r]$ are always one, despite of the type and magnitude of the other signals involved in adaptation process, and this value does not depend on the error exponent[3]. All these suggests the difficulties we can expect when this point is reached during our adaptation. In addition, the criteria controlling the

[3]Further reflections on this issue can be found in [15, 71].

error exponent update $r(k+1) \equiv F[|e(k)|]$ directly obtained from instantaneous error, contain also the measurement noise, so in this case the performances of the algorithms will be quite sensitive to the noise disturbance.

The goal is to obtain as much as possible benefits in the desired framework by combining different cost functions. However we should not expect more than their best performances. For instance, LMF outperforms LMS by larger initial gradients and lower residual error, but these are generally associated with decrease in the degree of stability and convergence speed.

An important aspect derives from the comparison of algorithms. We need to specify clearly the criteria we shall use when we compare them. It is also important the type of step-size we are dealing with during error exponent adaptation: we can consider algorithms where the step-size is always constant, or the step-size is that optimal one for the corresponding error exponent and the given criteria.

In this thesis we are mainly interested in convergence speed, where the convergence level is achieved 20 dB below the far-end signal, and we consider only constant step-size CFA algorithms, except Section 6.1.

Thus, in almost all situations, during error exponent adaptation, it is desirable to start with a large power $r(0)$ and to stop with a small power $r(\infty)$. In order to exploit both the tracking capabilities and noisy stability of the LMS algorithm, and the initial faster convergence of LMF, a good choice[4] is $r(0) = 4$, and $r(\infty) = 2$. However, in some applications $r(\infty) = 1$ is desired, which gives better behaviour in impulsive environments.

2.2.6 Cost function adaptation algorithms

The first cost function adaptation algorithm was introduced in [21]. The derivation of this CFA stochastic gradient algorithm follows the principle of minimum disturbance [35, 84]. The result is in fact a piecewise non-quadratic algorithm and the power of the cost function is updated using the relationship:

$$r(k+1) \equiv F[|e(k)|] = \frac{RE_{dB}}{|e(k)|_{dB}},$$

where RE_{dB} is an arbitrary constant and $|e(k)|_{dB}$ is the error modulus, measured in dB. The weights were computed using the simple recursive relation (2.1) as in the case of

[4]There is another important reason for the CFA algorithm to have an error exponent adaptation time short, and also for its steady-state to follow LMS [39]: The bounds of the step-size provided for the LMF and NQSG algorithms in [71, 81] compared with that one for the LMS [85] indicate that the algorithm based on cost functions with $r \neq 2$ can be more sensitive to attenuated far-end signals than the LMS.

non-quadratic algorithms, where $r = r(k)$:

$$\hat{\boldsymbol{h}}(k+1) = \hat{\boldsymbol{h}}(k) + \mu r(k)\boldsymbol{x}(k)|e(k)|^{r(k)-1}\text{sgn}[e(k)], \tag{2.4}$$

as we took into account only real signals. However the error exponent had to be updated in terms of a well-behaved estimator of the instantaneous error, otherwise instability can occur. At the beginning two types of error mappings have been tried: the running average of the modulus of the instantaneous error and the log running average of the squared instantaneous error. The first CFA algorithms implemented were the decreasing staircase power-error algorithm (with only integer or square roots powers) and the decreasing smooth power-error algorithm. We found that there is no big differences between these approaches, including the divergence appearances [21].

The next step was to reduce this effect, and we tried to distinguish the causes of instabilities. Beside bounding the error exponents like the variable step-size methods for step-size, we were also looking for an error estimate as smooth as possible, in order to reduce the error exponent update sensitivity to the noisy error. We found that the normalised tap-error vector norm behaves quite well [22]. The obtained error assessment is smooth, convenient to compute, but sometimes it is difficult to find in practical situations. Nevertheless, we established that both higher order error exponents, and noisy error estimate contribute to missconvergence. Another new feeling was that we need more degrees of freedom in choosing the error exponent update, as we noticed that best results were achieved when the error exponent update and adaptation periods are closed. The cancellation of the posterior error output [22] or the stationarity of gradients (Section 3.1) do not provide enough parameters sometimes.

A more general case was pointed out in [59], where the power updating rule:

$$r(k+1) \equiv F[|e(k)|] = \frac{r(0)}{|e(k)|_{dB}{}^{\beta-1}},$$

was derived by enforcing the same direction of the instantaneous gradient as in the case of non-quadratic algorithms. If $\beta = 1$, then $r(k+1) = r(0)$, and we retrieve LMS, LMF and NQSGr. A detailed analysis of this algorithm is provided in [60].

The LCFA (Linear Cost Function Adaptation) algorithm is a special case of this family. The error exponent is adjusted in such manner that it is linearly decreasing during the time of adaptation. A new error mapping was implemented [59]. This was done using the technique of the peak detector in classical amplitude modulation. We pass the logarithmic modulus of the instantaneous error through a first order recursive digital filter, the equivalent of the low-pass RC filter. If the noisy error is processed as mentioned

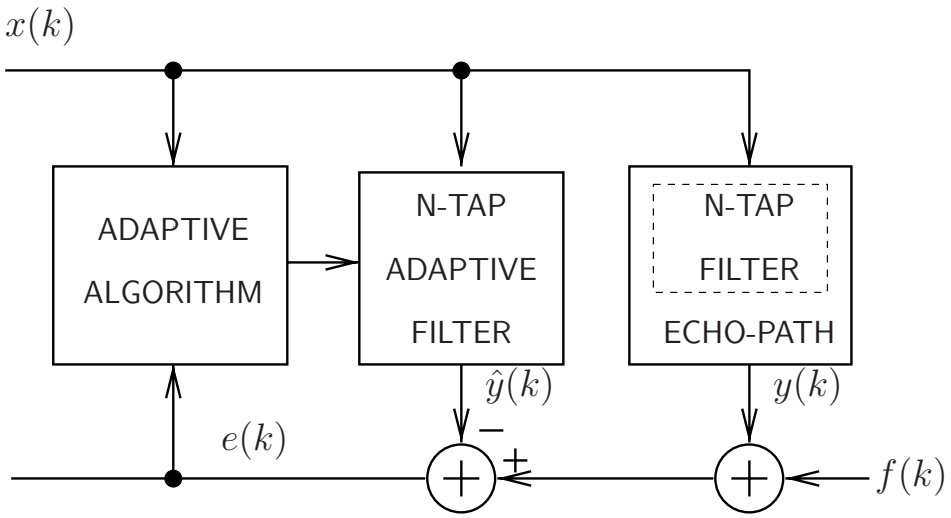


Figure 2.2: Simplified block diagram of simulation setup.

above, and under some circumstances, the logarithmic output is linear decreasing, and also the error exponent is linear decreasing.

Other efforts were concentrated in adaptation of cost function in a recursive way [61]. The derivation of the new algorithm does not use an estimator of the instantaneous error as the previous cost function adaptation algorithms did. In the RCFA case, the new error exponent is computed from the previous one using an usual LMS recursive equation. We found that this method improves the sensitivity of the error exponent with respect to the noisy error, while the other benefits of the CFA algorithms in terms of the convergence speed and residual error remain.

2.3 Modelling

The setup for the data echo canceller is shown in Figure 2.2. The adaptive FIR filter is trying to make a copy $\hat{y}(k)$, of the echo-path output $y(k)$, using the signal $x(k)$ as an input, based upon a measurement of the signal that remains after subtracting $\hat{y}(k)$ from the received signal $y(k) + f(k)$ [17]. Thus our framework is described by the following signals and equations:

- $x(k)$ is the input sequence (near-end signal);
- $y(k)$ is actual echo path output sequence;

- $\hat{y}(k)$ is the synthetic echo signal;
- $f(k)$ is the far-end signal;
- $e(k)$ is the resulting error:

$$e(k) = y(k) + f(k) - \hat{y}(k). \tag{2.5}$$

At the kth time sample, the adaptive and the echo-path filter have impulse responses given by

$$\hat{\boldsymbol{h}}(k) = [\hat{h}(0), \hat{h}(1), \ldots, \hat{h}(N-1)]^t, \ \boldsymbol{h}(k) = [h(0), h(1), \ldots, h(N-1)]^t,$$

where N is the number of filter coefficients. We denote by $\boldsymbol{x}(k)$ the transpose of the input observations vector

$$\boldsymbol{x}(k) = [x(k), x(k-1), \ldots, x(k-N+1)]^t,$$

and by $\Delta\boldsymbol{h}(k) = \hat{\boldsymbol{h}}(k) - \boldsymbol{h}(k)$ the tap-error vector. The outputs of the two filters can be written as

$$y(k) = \sum_{n=0}^{N-1} h(n) \cdot x(k-n) = \boldsymbol{h}^t(k)\boldsymbol{x}(k), \qquad \hat{y}(k) = \sum_{n=0}^{N-1} \hat{h}(n) \cdot x(k-n) = \hat{\boldsymbol{h}}^t(k)\boldsymbol{x}(k). \tag{2.6}$$

Therefore

$$e(k) = f(k) - \Delta\boldsymbol{h}^t(k)\boldsymbol{x}(k). \tag{2.7}$$

Two types of echo path have been used. The first one consists of channels that are single pole single zero digital filters, with the impulse response series truncated [71]. The transfer function of the echo path is of the form:

$$H(z) = \sum_{j=0}^{N-1} p^j z^{-j},$$

where $p \in (0, 1)$. The feedback coefficient p of the echo path filter is chosen in such a way that the lower level of the impulse response will be attenuated by A dB at the N-th sample. The second type of echo path model is numerically generated as in [70], by sampling a diagram of an actual telephone network connection. The impulse responses for a first type channel ($p = 0.80025, \ A = -60, \ N = 32$) and for the second echo-path (real hybrid) are shown in Figure 2.3.

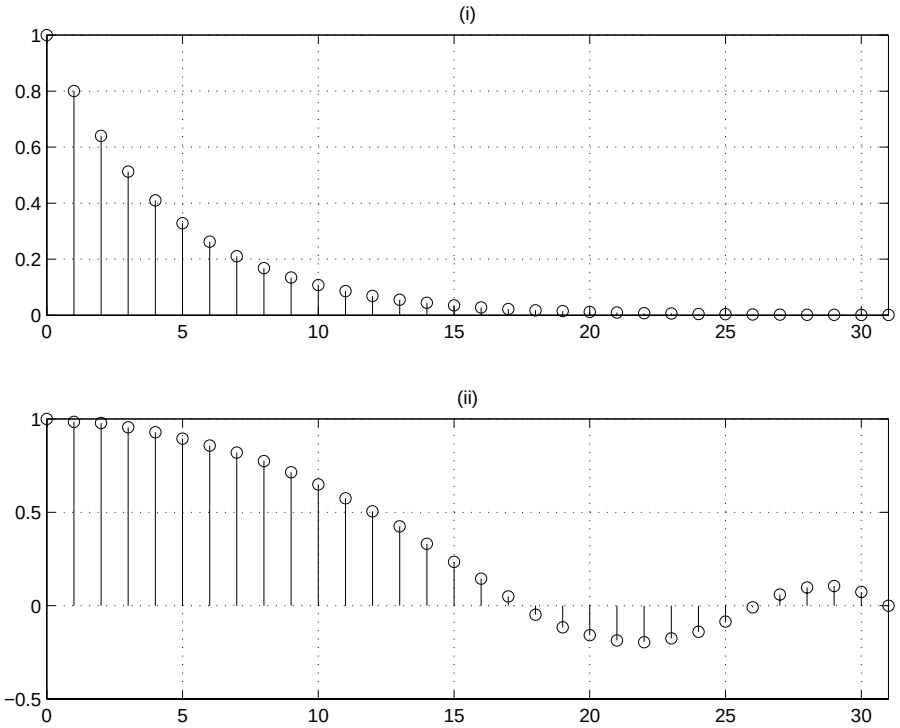


Figure 2.3: Impulse responses for the two sample echo path of channel models:
(i) the single pole single zero digital filter ($p = 0.80025,\ A = -60,\ N = 32$);
(ii) the real hybrid ($N = 32$) .

In our simulations the near-end sequence $\{x(k)\}$ is modeled by a non-Gaussian random bipolar sequence from the set $\{1, -1\}$, and the far-end signal is generated by an independent random bipolar sequence from the set $\{a, -a\}$, where a is the attenuation of the far-end signal. The presence of Gaussian interference is done by adding to the far-end signal of a zero-mean Gaussian density with variance σ. We performed some experiments with added dispersion to the attenuation. In this case the far-end signal is modelled by passing the signal through a transversal digital filter.

As we are searching for optimized cost functions with different error exponents during adaptation, the performance measure selected should not include directly a certain power of the instantaneous error. Taking into account the framework addressed, we choose the normalised form of the tap-error vector [70, 71]:

$$p(k) = \frac{\|\hat{\boldsymbol{h}}(k) - \boldsymbol{h}(k)\|}{\|\boldsymbol{h}(k)\|}. \tag{2.8}$$

2.4 Appendix

In the following we shall derive relationship (2.1) for computing the gradient of the general form of the stochastic gradient algorithm if $e(k) \in \mathbb{C}$, and $r \in \mathbb{R}$. Let us first consider the function $f : \mathbb{C} \to \mathbb{R}_+$, defined by $f(z) = |z|^r$, and $z = x + jy$, with $x, y \in \mathbb{R}$. We have

$$\frac{\partial f}{\partial x} = rx(x^2 + y^2)^{\frac{r}{2}-1}, \quad \frac{\partial f}{\partial y} = ry(x^2 + y^2)^{\frac{r}{2}-1},$$

and where they exist, it results that

$$f'(z) = \frac{\partial f}{\partial x} + j\frac{\partial f}{\partial y} = r|z|^{r-2} \cdot z. \tag{2.9}$$

This is true for every $r \in \mathbb{R}$, unless $x = y = 0$, where we distinguish three cases:

- $r < 1$. It is easy to show that
$$\lim_{x=y\to 0} \frac{\partial f}{\partial x} = \lim_{x=y\to 0} \frac{\partial f}{\partial y} = \infty;$$
- $r = 1$. For every $m \in \mathbb{R}$ we obtain
$$\lim_{x=my\to 0} \frac{\partial f}{\partial x} = r(m^2 + 1);$$
- $r > 1$. Finally we have
$$\lim_{x,y\to 0} \frac{\partial f}{\partial x} = \lim_{x=y\to 0} \frac{\partial f}{\partial y} = 0.$$

It follows that $f'(z)$ exists for all $z \in \mathbb{C}$ iff $r > 1$.

Now, in order to derive an adaptive algorithm to adjust filter parameters based on the least mean r-power error metric, we choose the steepest descent algorithm

$$\hat{\boldsymbol{h}}(k+1) = \hat{\boldsymbol{h}}(k) - \mu\frac{\partial J_r}{\partial \hat{\boldsymbol{h}}} = \hat{\boldsymbol{h}}(k) - \mu\frac{\partial\{E[|e(k)|^r]\}}{\partial \hat{\boldsymbol{h}}}.$$

From Equation (2.9) and taking account of

$$\frac{\partial e(k)}{\partial \hat{\boldsymbol{h}}} = \frac{\partial}{\partial \hat{\boldsymbol{h}}}(f(k) - \Delta\boldsymbol{h}^t(k)\boldsymbol{x}(k)) = -\boldsymbol{x}(k),$$

we obtain

$$\frac{\partial J_r}{\partial \hat{\boldsymbol{h}}} = E[-r|e(k)|^{r-2}e(k)\boldsymbol{x}(k)]. \tag{2.10}$$

We replaced the ensemble averaged value by the instantaneous value and in a corresponding fashion we use the instantaneous estimate in place of the gradient of the ensemble averaged [35]. Thus formula (2.1) holds for every $r > 1$.

Chapter 3

Nonrecursive Cost Function Adaptation

A family of stochastic gradient algorithms and their behaviour in the data echo cancellation work platform are presented in this Chapter. The cost function adaptation algorithms use an error exponent update strategy based on an absolute error mapping, which is updated at every iteration. The quadratic and nonquadratic cost functions are special cases of the new family. Several possible realizations are introduced using these approaches.

The idea behind the first approach (Section 3.1) relies on adjusting the error exponent parameter during the adaptation by enforcing the same value of gradient for two consecutive error exponents. The update expression is identically with that derived in [22], obtained using the stationary gradient approach.

Its extension is the second main method (Section 3.2). In this special situation we have the gradient approach. It is derived by enforcing the same direction of the gradient as in the case of non-quadratic algorithms. As a special case the linear adaptation of power of the cost function results [59].

For all these methods first the derivation of algorithms is presented, then the influence of different parameters is discussed. The noisy error problem is pointed out and several solutions are proposed. Finally the simulation outcomes confirm the effectiveness of the proposed family of algorithms. The results obtained for different values of parameters, types of signals and echo-path models have demonstrated the reliability of the proposed method, their benefits over the quadratic algorithms and the comparable performances to the variable step-size methods.

The convergence and steady-state analysis of the proposed algorithms has been carried out and closed-form expressions for the step-size bounds and misadjustment will be

provided in Chapter 5.

3.1 The Stationary Approaches

3.1.1 Method

In addition to the weights update equation, the CFA algorithms need an extra update equation for the error exponent. As we are searching for $r(k)$ depending only on the system error, a problem is whether we should update the error exponent before or after the estimated filter coefficients. This dilemma can be solved by enforcing the same value of the gradient for the given error and the two consecutive error exponents:

$$r(k)|e(k)|^{r(k)-2} = r(k+1)|e(k)|^{r(k+1)-2}. \tag{3.1}$$

In practical situations the system identification error cannot be evaluated exactly, as the instantaneous error contains the measurement noise. Although Equation (3.1) can be implemented directly (Examples 1 and 2), special average techniques can be useful in order to obtain a satisfactory CFA adaptation (Section 3.1.2). Thus, regardless of the implementation detail, the proposed CFA algorithm is basically driven by two well-known principles as follows.

- The updated filter estimate should be disturbed in a minimal fashion (The principle of minimal disturbance [35, 84]).
- The adjustment of $r(k)$ must be done using finite-time average characteristics $\bar{e}(k)$ rather than the instantaneous values of the error $e(k)$ [85].

The proposed CFA algorithms try to satisfy these two principles as simply and directly as possible. We start the kth iteration keeping the error exponent constant. We then apply NQSG$r(k)$, the non-quadratic algorithm with power $r(k)$. The error $e(k)$ results and the estimated filter coefficient vector is computed using (2.1) with $r = r(k)$. At the same time we are searching for the new power $r(k+1)$ of the cost function, keeping the gradient unchanged. This finishes the kth iteration. The resulting algorithm is a piecewise non-quadratic algorithm, direct and simple to understand.

3.1.2 The staircase CFA algorithm

Previous works [70, 71, 73] consider algorithms with gradient switched during convergence, consisting of applying the fixed error exponent algorithm ($r > 2$) and switching to the

LMS when the absolute value of the error $e(k)$ reaches or crosses the limits of ± 1. These methods were developed in order to handle the instability which arises, by reducing for a while the gradient. Combining these achievements with the previous comments, we consider the staircase CFA algorithm defined by the following:

1) The weights are computed with Equation (2.1);

2) The power $r = r(k)$ is updated by:

$$r(k+1) \equiv F[e(k)] = \begin{cases} R_1, & \text{if} \quad E_1 \leq |e(k)| \leq E_M, \\ R_2, & \text{if} \quad E_2 \leq |e(k)| \leq E_1, \\ \vdots & \\ R_P, & \text{if} \quad E_P \leq |e(k)| \leq E_{P-1}, \\ 2, & \text{otherwise.} \end{cases} \tag{3.2}$$

where R_i, E_M and E_i $(i = 1, 2, \ldots, P)$ are positive constants for error exponents, respectively for error magnitudes. To be noticed that we can consider only integer or square root powers in order to reduce the computational effort [21], though other values are also possible. To find E_i, we naturally derive the gradients for every constant and the neighbouring error exponents, then set them equal:

$$\begin{aligned} R_1 {E_1}^{R_1-1} &= R_2 E_1^{R_2-1}; \\ R_2 E_2^{R_2-1} &= R_3 {E_2}^{R_3-1}; \\ &\ldots \\ 2E_P &= R_P E_P^{R_P-1}. \end{aligned} \tag{3.3}$$

It results:

$$\ln E_i = -\frac{\ln R_{i+1} - \ln R_i}{R_{i+1} - R_i}, \ i = 1, 2, \ldots, P, \tag{3.4}$$

where $R_{P+1} = 2$. Note that the E_i constants do not depend on far-end signals levels or parameters of the echo-path. From Equations (3.2) and (3.4), we have

$$\begin{aligned} E_1 \geq E_2 \geq \ldots \geq E_P, \\ R_1 \geq R_2 \geq \ldots \geq R_P. \end{aligned} \tag{3.5}$$

In the following an experiment will justify the benefits of a smooth estimate of the error's modulus.

Example 1 .

In this simulation example the proposed algorithm is applied to data echo cancellation of the first echo-path channel ($p = 0.80025, \ A = -60, \ N = 32$), where the input signals are binary (Section 2.3). Every learning curve was obtained doing 20 averages.

First an implementation of the LMS algorithm was simulated. For each of the following levels of attenuated far-end signal: $a_{dB} = -15, -20, -25, -30$, an optimum step-size was found. This value gives the minimum number of iterations in such a way that convergence level is 20 dB below the far-end signal.

Then the staircase CFA algorithm ($P = 3$, $R_1 = 4$, $R_2 = 3$, $R_3 = 2.5$, $E_M = 1$, $E_1 = 0.75$, $E_2 = 0.6944$, $E_3 = 0.64$) as described in Equation (3.2), i.e. using the instantaneous error for updating the error exponent, was completed with the same optimal procedure as for the LMS.

Finally another three finite-time averages of the error modulus (Section 2.3), assessed every $W = 128$ iterations, with same $L = 16$ samples length have been replaced the instantaneous error in the right-hand side of the Equation (3.2). These are the mean, the mean-square and the median. The initializations of the averages are so that $r(k) = R_1$, $k = 1, 2, \ldots, W - 1$.

For all these examples the results show that the adaptation time reduces when the far-end signal level decreases. Also the CFA algorithm outperforms systematically LMS, but in different manner. Therefore, in all cases, we normalise the convergence speed of CFA and LMS algorithms for the corresponding conditions, and these ratios are shown in Figure 3.1. The top curve suggests that using directly Equation (3.2) we gain 1-8 % reduction in adaptation time, but if we average the right-hand side of Equation (3.2), the results are more promising (9-22%), with emphasis at high noise levels.

Thus we show that the smoothness and error tracking capabilities of the term in the right-hand side of Equation (3.2) clearly affects the convergence speed of the staircase CFA algorithm. This issue is important in the case of the smooth CFA algorithm, where the error term interferes both in the endpoints of intervals and error exponent update expression.

3.1.3 Derivation of the smooth CFA algorithm

For the beginning we reconsider from Equation (3.1) the equality of the two gradients, assessed for the same instantaneous error, but for two different consecutive error exponents. We distinguish two cases:

- $|e(k)| \geq 1$; in this case we have $r(k+1) = r(k)$ (Appendix 3.3).
- $|e(k)| < 1$; the resulted Equation in $r(k+1)$ unknown is transcendental.

In order to solve this Equation, we use the assumption of a smoothed error exponent:

$$|r(k+1) - r(k)| \ll r(k), \quad r(k+1). \tag{3.6}$$

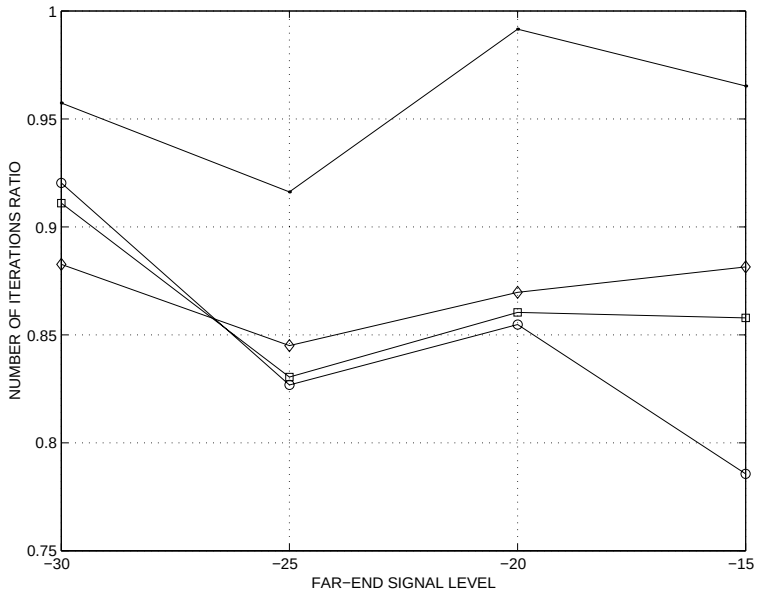


Figure 3.1: Ratio between the minimum numbers of iterations of the staircase CFA algorithm and LMS, when in right-hand side of Equation (3.2) we have modulus of the instantaneous error ($\cdot-$), its mean ($\circ-$), mean-square ($\square-$), and median ($\diamondsuit-$) finite-time averages.

Using [7]:

$$\ln(1+x) \approx x, \text{for all real } |x| \ll 1, \tag{3.7}$$

we conclude that:

$$\ln|e(k)| = \frac{\ln r(k) - \ln r(k+1)}{r(k+1) - r(k)} = \frac{\ln\left[1 + \dfrac{r(k) - r(k+1)}{r(k+1)}\right]}{r(k+1) - r(k)} \approx -\frac{1}{r(k+1)}, \tag{3.8}$$

so a solution of the new error exponent might be

$$r(k+1) \approx -\frac{1}{\ln|e(k)|} = -\frac{8.68}{|e(k)|_{dB}}, \tag{3.9}$$

where $|e(k)|_{dB}$ is the error modulus measured in dB.

Remark 1. *The above proof is valid only at points where condition (3.6) holds.*

Combining this with Equation (3.9), we obtain that

$$||e(k+1)|_{dB} - |e(k)|_{dB}| \ll ||e(k)|_{dB}|, \quad ||e(k+1)|_{dB}|. \tag{3.10}$$

This shows us clearly that the term in the right-hand side of Equation (3.9) should be also smooth, therefore a possibility is to average the modulus of the instantaneous error.

Remark 2. *The staircase CFA relation between error exponents and E_i constants (3.4) is the left-hand part of (3.8).*

From this equality, we obtain $r(k) \approx r(k+1)$, if $|e(k)| \approx 1$. This recommends no error exponent adaptation should occur, for $|e(k)| \leq E_0 < 1$, where R_0 and E_0 are positive constants such that $R_0 > 2$ and $\ln E_0 \approx -1/R_0$.

Remark 3. *We can substitute in our approach the exponent $r(k)$ by $Cr(k)$, where C is an arbitrary positive constant.*

This will change the cost function expression, the right-hand side of the weights (2.1) and error exponent update (3.9) relationships by the corresponding constant. Finally we retrieve the result from [22], but using a different approach[1].

It follows that the smooth CFA algorithm is defined by the following:

1) The weights are computed with Equation (2.1);

2) The power $r = r(k)$ of the cost function is updated by:

$$r(k+1) \equiv F[e(k)] = \begin{cases} R_0, & \text{if} \quad E_0 \leq |e(k)| \leq E_M, \\ \dfrac{R_0 |E_0|_{dB}}{|e(k)|_{dB}}, & \text{if} \quad E_\infty \leq |e(k)| < E_0, \\ R_\infty, & \text{if} \quad |e(k)| < E_\infty, \\ 2, & \text{otherwise}, \end{cases} \tag{3.11}$$

where R_∞ ($R_\infty < R_0$) and E_∞ are positive constants.

Remark 4. $E_M \geq 1$

Now we recall Equations (3.2) and (3.11). Instead of switching to the LMS when the absolute value of the error reaches and crosses the limit 1 as in [70, 71, 73], this can be done when the error modulus meets another constant. For instance this constant E_M can be obtained directly comparing the LMS maximum step-size μ_{max} with the maximum value of the variable step-size of the CFA algorithm given by (2.2). From Equations (3.5) and (3.11) we have

$$E_M = \left(\frac{2\mu_{max}}{\mu R} \right)^{\frac{1}{R-2}}, \tag{3.12}$$

[1] In [22] we derived the stationary approach by nulling the gradient of the cost function $J_r = E[|e(k)|^r]$, where the power r is a function only of the instantaneous error.

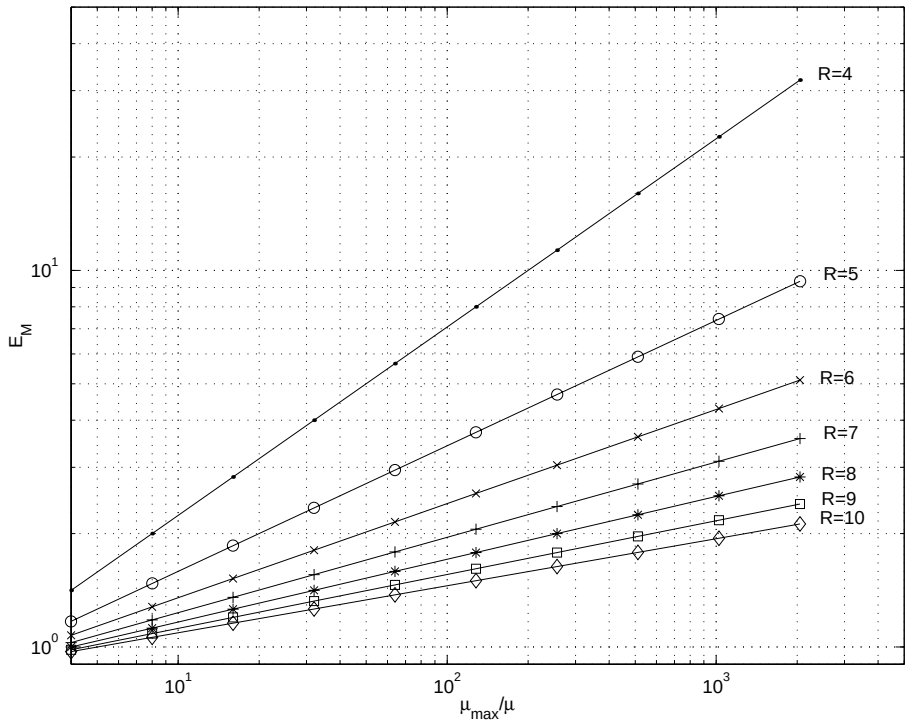


Figure 3.2: Constant E_M (dB) versus step-size ratio $\dfrac{\mu_{max}}{\mu}$ for different error exponents.

where $R = R_1$ for the staircase CFA algorithm and $R = R_0$ for the smooth CFA algorithm. The variation of the constant E_M in dB, as a function of the step-size ratio μ_{max}/μ, for different values of error exponent R is presented in Figure 3.2. It is apparent that for $R = 4$ not many switchings to maximum step-size of the LMS are needed, however this amount increases toward $R = 10$. With this consideration in mind, we can select the constant E_M given by Equation (3.12) in the following simulations.

Note that the proposed CFA algorithms can use a step-size bigger than the LMF. Actually the step-size can be increased close to the maximum step-size of the LMS algorithm for small lengths of the time-average of the error, since we skip to the power two, when we cross the $\pm E_M$ limits of the error. But if we increase the length of the time-average, the maximum-step size will decrease correspondingly as we have to consider the introduced delay, and the occurrence of possible instabilities. This situation warns us that if we use in such cases the constant E_M in (3.2) and (3.11) it is better to decrease the value given by (3.12).

3.1.4 Discussion

This section shows some different aspects of the implementation of the smooth CFA algorithm. The first one considers the necessity of the R_∞ and E_∞ ($E_\infty > 0$) final parameters in Equation (3.11), then the influence of initial constants in algorithm behaviour is discussed, providing that a smooth estimator of the error is available. In this context we provide a short comparison between CFA and quadratic algorithms.

For the beginning, as an illustration of the problems generated by the simultaneous presence of small errors and exponents, we consider the following example.

Example 2. *The trade-off between instantaneous error and error exponent*

Let the channel be the same as in Example 1, with one difference. We consider for simplification only one level $a_{dB} = -15$ of attenuated far-end signal. A direct implementation of (3.11) was performed, where $r_0 = 4$, $E_M = 1$, $E_0 = 0.75$, $E_\infty = 0$ and using the step-size found as the optimal for LMS. The normalised tap-error vector norm for one simulation experiment is shown in Figure 3.3, and it is apparent that the algorithm behaves similarly as though it is perturbed by an impulsive noise. Indeed, this might be the case. To have a look deep insight of the function, we use a narrow window on the same plot, in addition to those of the instantaneous error and error exponent update (Figure 3.4). We can see in Figure 3.4(a) a triplet of very small error, close to zero, and they lead to the next small error exponents, less than 0.5, shown in Figure 3.4(c). Thus the next gradient becomes suddenly large, and acts as an impulse as we can see in Figure 3.4(b). This always happens when a couple of very small errors appear, but the effect is more obvious if the number of small errors is bigger. However, this number cannot be too large, since after an impulse in gradient, the system goes far from the minimum. Afterwards this do not affect the system too much, since for large errors $|e(k)| \geq 1$, the gradient is reduced by forcing $r(k) = 2$. It still reappears for successive small errors.

We can avoid this situation. One possibility is to low bound the error modulus and the error exponent, and the necessity of the E_∞ and R_∞ extra parameters results. Another possible way to improve the performances is to use time-averaging of the error modulus when we update the error exponent. Thus the error exponent will also be smoothed and the reduction in convergence speed is also present (Section 3.1.3).

Proceeding further, from Equation (3.8) and Remark 3 we get that during the error exponent adaptation the product $r(k+1)|e(k)|_{dB}$ is constant [22]. This suggests to study the behaviour of the proposed smooth CFA algorithm, with different initial constants, provided that a smooth estimate of the error modulus is available. Indeed, we consider

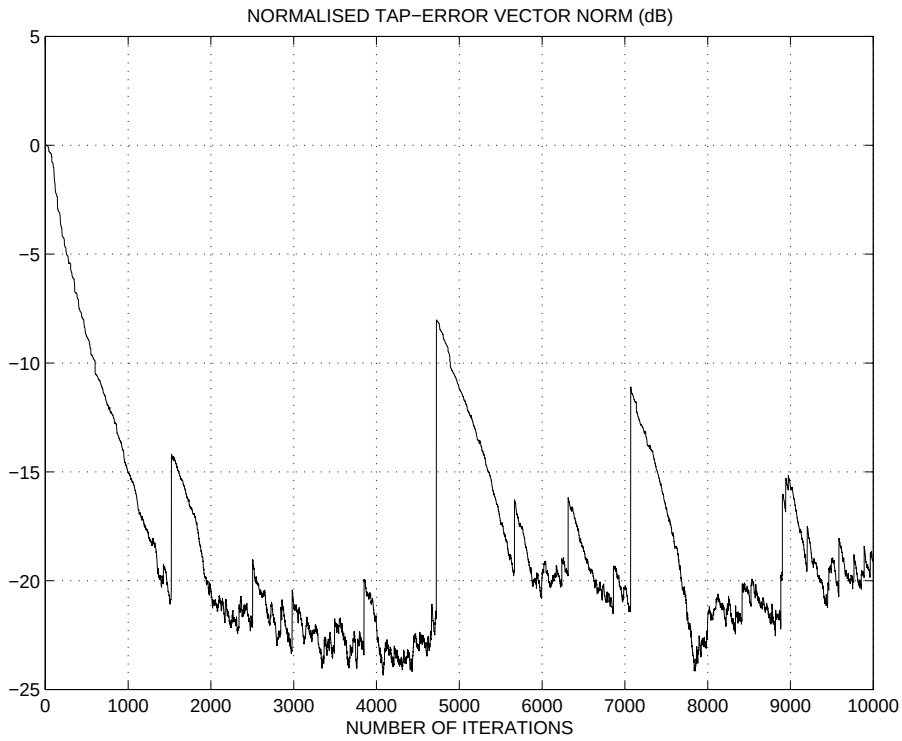


Figure 3.3: Learning curve of the smooth CFA algorithm given by Equation (3.11), where no time-average was used.

the normalised form of tap-error [21] and it is twice used in this simulation example, as a performance measure and in order to compute the error estimate in the error exponent update.

Example 3. *The initial conditions settlement*

The CFA's starting power was $r(0) = 4$ and it is kept unchanged till $k = K$ corresponding to $p(k)$ less than a given value $p(K)$. For $k > K$ the error exponent adaptation rule is done by:

$$r(k+1) = r(0) \cdot \frac{\log[a^2 + \|\hat{\boldsymbol{h}}(K)\|^2 p^2(K)]}{\log[a^2 + \|\hat{\boldsymbol{h}}(k)\|^2 p^2(k)]}. \tag{3.13}$$

Derivation of this formula is given in the Appendix 3.3. We select for this experiment the same data echo cancellation simulation framework of the first channel with the parameters $p = 0.80025,\ A = -60,\ N = 32$ (Section 2.3). Figure 3.5 shows the relation (3.13) for $p(K)_{dB}$, from 0 to -20 dB, where "CFAxx" is the corresponding plot for $p(K)_{dB} = -xx$ dB. Once again, it is clear that an earlier start of error exponent update can affect the

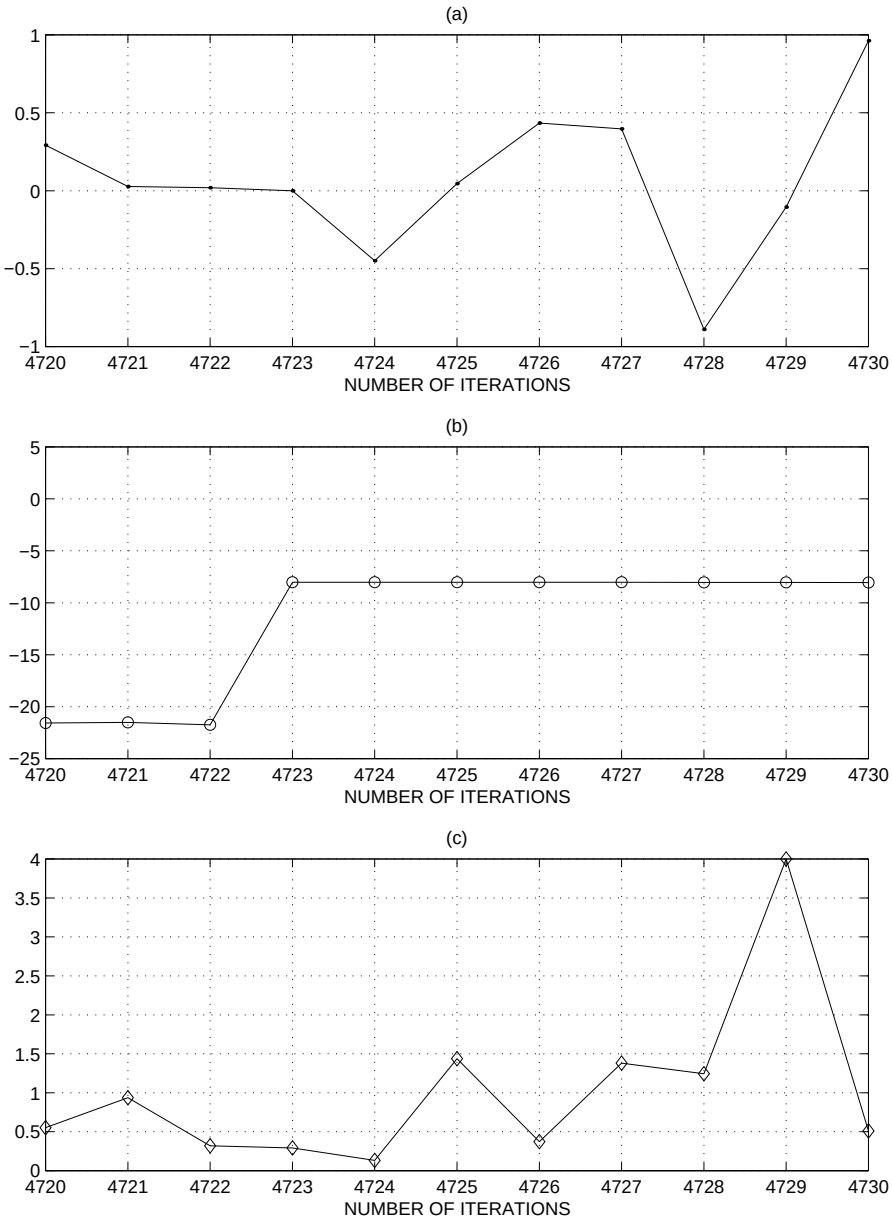


Figure 3.4: A narrow window on the plots of the instantaneous error (a), normalised tap-error vector norm in dB (b) and the error exponent update (c) for the same example as in Figure 3.3.

convergence. For instance, following the curve CFA0, the decreasing of the error exponent will increase the normalised tap-error vector norm. This is not surprisingly since $|\bar{e}(K)|_{dB}$ is positive (Section 3.1.3). Moreover, condition (5.1) is not verified. Also $r(\infty)$ could be

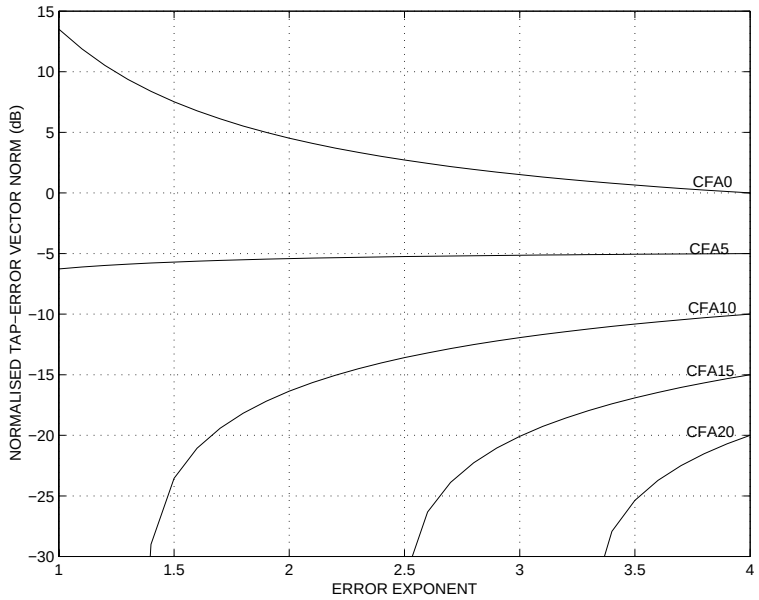


Figure 3.5: Error exponent update and the normalised tap-error norm (Example 3).

less than one (CFA5). As we have already pointed out, this situation forces small errors back to bigger errors, increasing the residual error, thus worsening the steady-state of the algorithm. For CFA10, CFA15 and CFA20 this is not anymore valid, since we have

$$r(\infty) = \lim_{p(k)\to 0} r(0) \cdot \frac{\log[a^2 + \|\hat{\boldsymbol{h}}(K)\|^2 p^2(K)]}{\log[a^2 + \|\hat{\boldsymbol{h}}(k)\|^2 p^2(k)]} = r(0) \cdot \frac{\log[a^2 + \|\hat{\boldsymbol{h}}(K)\|^2 p^2(K)]}{2\log a} > 1.$$

A set of simulations was performed with the same step size [52] for all the algorithms ($\mu = 825 \cdot 10^{-6}$). The input signals are generated as in Section 2.3 and the level of the attenuated far-end signal power is -15 dB. The convergence performances of LMS, LMF, CFA5, CFA10, CFA15 and CFA20 algorithms are illustrated in Fig. 3.6. The learning curves obtained are the average of 20 runs. Now we can see how CFA algorithms behave qualitatively in comparison with quadratic algorithms. A late start reduces the variation of the exponent (CFA20) and in this case we expect performance close to LMF. Alternatively, an earlier start affects the convergence and residual error. It seems that CFA11.5 which exploits all the range of exponents between 4 and 2, has the error exponent update period equal with the adaptation time and steady-state like LMS, behaves the best, as we have 23% improvement in convergence speed compared with LMS. From this result, we can state that the best performances of the staircase CFA with noisy finite-time averaging (Figure 3.1) and smooth CFA when a smooth estimate of the error modulus is

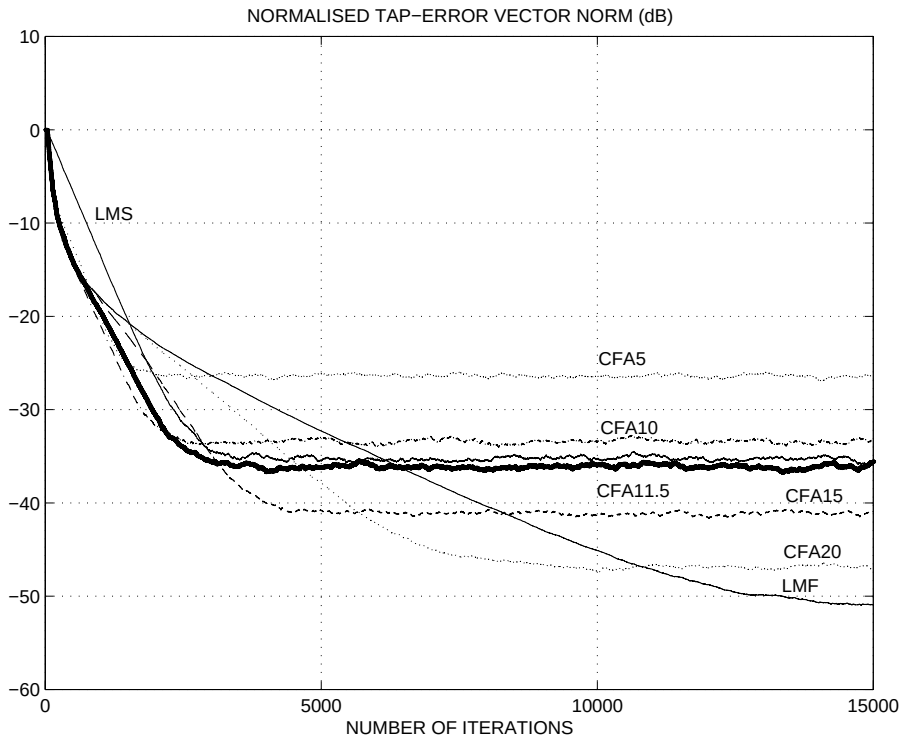


Figure 3.6: Learning curves of the quadratic and CFA algorithms (same step size).

available, are quite similar.

We can conclude that from the initial conditions we can foresee the behaviour of the CFA algorithm, including the performances such as the adaptation time and of the steady-state error. Moreover, the error exponent must be constant for a while during convergence, thus the effective error exponent update is less than the adaptation period.

Example 4. *A short comparison between CFA and quadratic algorithms*

Performances of the proposed CFAxx approaches were evaluated through a series of experiments and compared with quadratic algorithms. A few types of graphical outcomes are presented in this section. Now the echo canceller is modelled by the first echo path channel, with $p = 0.75,\ A = -122,\ N = 32$ (Section 2.3). The level of the attenuated far-end signal is kept -15 dB.

The first set of simulations uses the same step size for all the algorithms [52], in our case $\mu = 75 \cdot 10^{-5}$. This ensures the convergence of all the algorithms: CFAxx, LMS, LMF. Figure 3.7 is the 3D representation of the trajectories for a sample of LMS, LMF

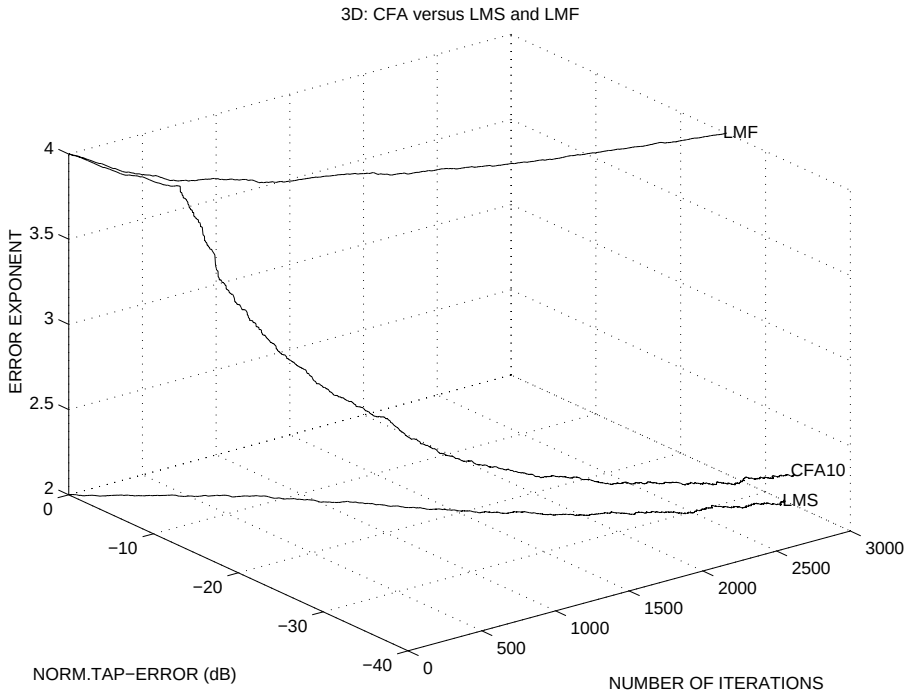


Figure 3.7: 3D representation of learning curves for a sample of LMS, LMF and CFA.

and CFA10. Now it is clear that during adaptation the CFA algorithm changes the error exponent, after a period while it is constant.

The second procedure for distinguishing the performances of proposed algorithms is to compare convergence rates for a given steady-state cancellation level [71]. The curves for LMS and CFA10 algorithms are shown in Figure 3.8, and they were obtained doing 100 averages. The results presented show us that CFA10 reaches the steady state convergence level before LMS does.

Finally Figure 3.9. compares the proposed CFA algorithm and the quadratic algorithms from the point of view of steady-state cancellation levels for a given maximum number of iterations. The results were obtained by averaging 25 times the learning curves for step-sizes in the range $(3 \cdot 10^{-5}, 10^{-3})$. For bigger step-sizes, we established that the LMF algorithm diverges. These plots display the performances over the step-size, for a number of iterations which is equal with 100, 500, 1000, 5000, and respectively 10000. It follows that the proposed time-varying error exponent algorithms permits a trade-off between convergence speed and cancellation level in the steady-state, at the cost of increased computational complexity.

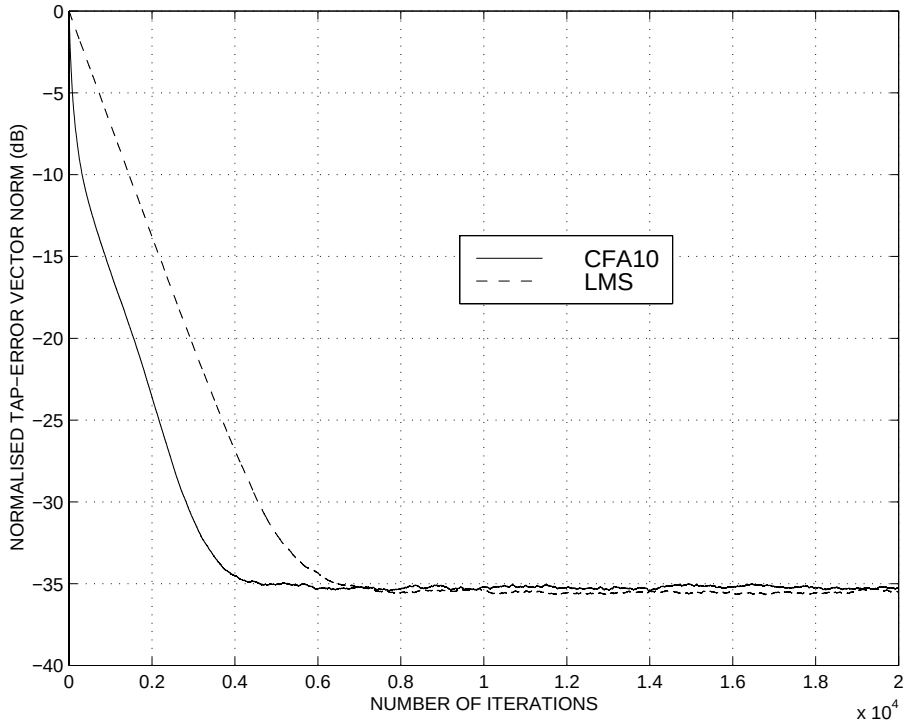


Figure 3.8: Learning curves of the LMS and CFA algorithms (optimum step-size).

3.1.5 Simulations

Now our goal is to investigate several performances of the proposed algorithms via computer simulations. First we compare the two offsprings of the CFA algorithm: the staircase and the smooth variant. The reference of the comparison is the LMS algorithm. Then the effect of algorithms parameters (finite-time averages and error exponents) is considered. Finally different parts of the signal and system model are changed, and their influences are discussed. In this Section the initializations of the smooth CFA algorithm are R_0, R_∞, and we always select $E_0 = \exp(-1/R_0)$, and $E_\infty = \exp(-1/R_\infty)$ (Section 3.1.3).

The following three finite-time averages have been used in the right-hand side of Equations (3.2) and (3.11): the mean, the mean square and the median of the modulus of instantaneous error. The length of the rectangular window is L. The guess of window length L needs special attention to statistical averages, and to follow statistical variations. Providing that we avoid the extremal cases, for large windows, there is a delay in following the error modifications and the computational effort is bigger. For very small windows the noise effect is also important (Section 3.1.4). In both cases instability may occur. More-

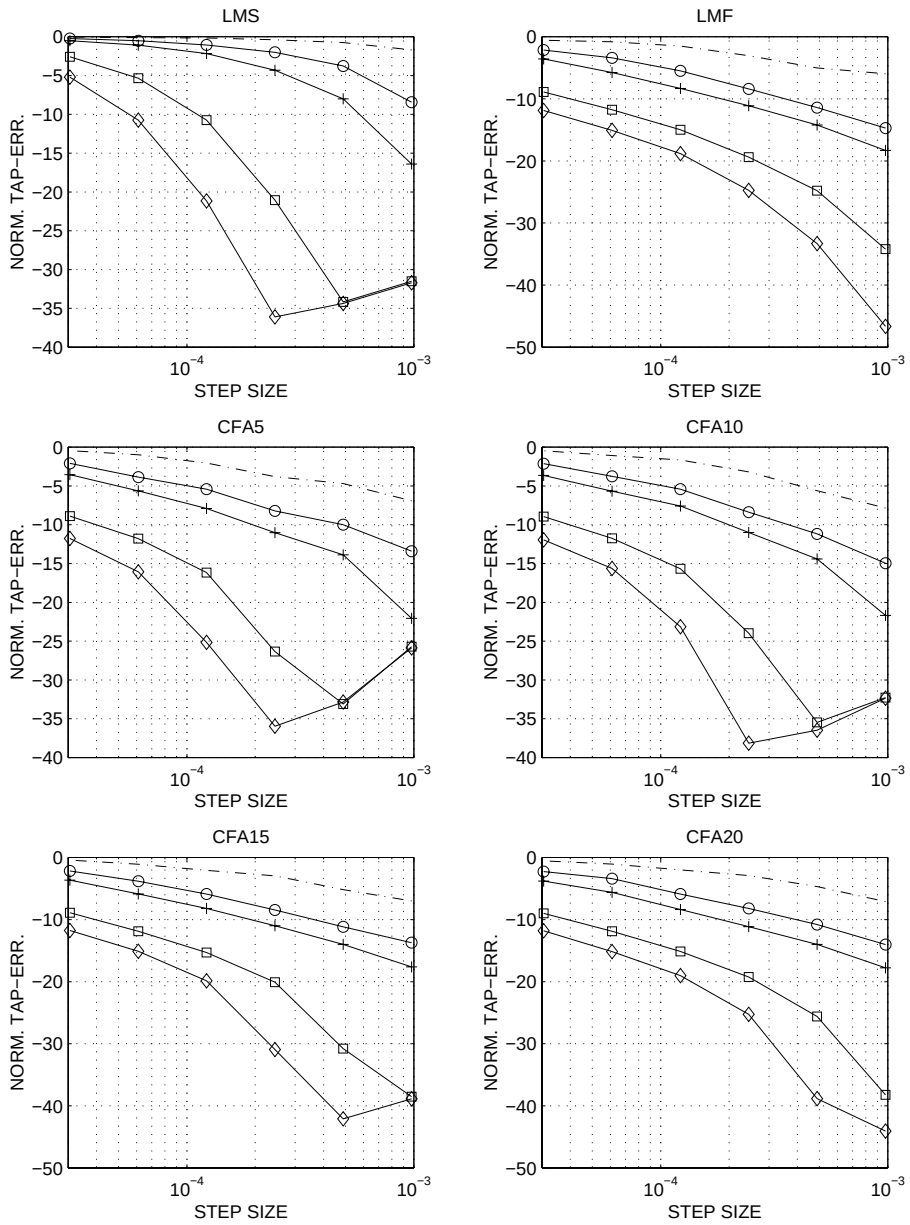
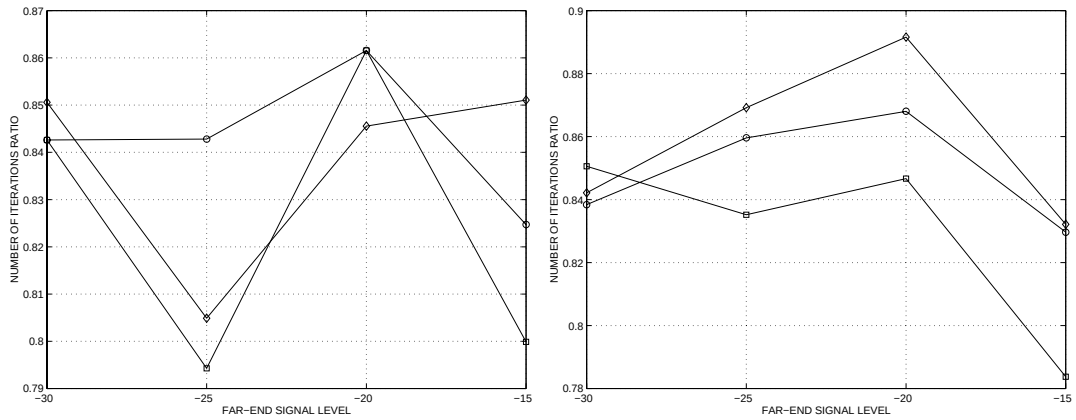


Figure 3.9: Performances of the algorithms versus step-size for different number of iterations: 100 ($-\cdot$), 500 ($-\circ$), 1000 ($-+$), 5000 ($-\square$), and respectively 10000 ($-\diamond$).

over, in implementation we have to consider the initializations of the finite-time averages. Apart from [21], these finite-time averages are not assessed in every iteration, only once every W iterations[2]. In this way we reduce the computational time, the approximation of

[2]For this reason in the following we call the parameter W as the sampling period of the time-average.



(a) (b)

Figure 3.10: Example 1 and Figure 3.1 revisited for staircase (a), and smooth CFA (b).

the error is smooth, and the error exponent is piecewise. We can see that this formulation of the algorithm includes both smooth ($W = 1$) and staircase ($W > 1$) CFA algorithms, if the parameter W is properly selected. Actually we need the logarithm of the time-average in order to update the error exponent, and in some cases the succession of average and logarithm operators can produce different results.

The first set of simulations reconsiders Example 1 ($E_M = 3$, $\bar{e}_0 = 1$), and we use the same procedure and performances. For the staircase CFA algorithm ($L = 128$, $W = 128$) with the same parameters as in Example 1, results shown in Figure 3.1 and Figure 3.10(a) indicate that for low levels of noise the increase of the window length from $L = 16$ to $L = 128$ improves the convergence speed, but this effect is not present for high levels of noise. However, for high levels of noise and large windows the mean-square time-average seems to work better than the mean. This result is retrieved in the case of smooth CFA algorithm, where Figure 3.10(b) illustrates the same experiment ($L = 128$). We remark that in almost all cases the mean-square time-average behaves the best, though there is no very important difference when we change the type of time-average.

We can conclude that there is no significant change between staircase and smooth CFA algorithms from the convergence speed point of view, and they always surpass the LMS algorithm performance from this point of view. The steady-state behaviour is similar for staircase and smooth CFA algorithms, providing that we have the same final error

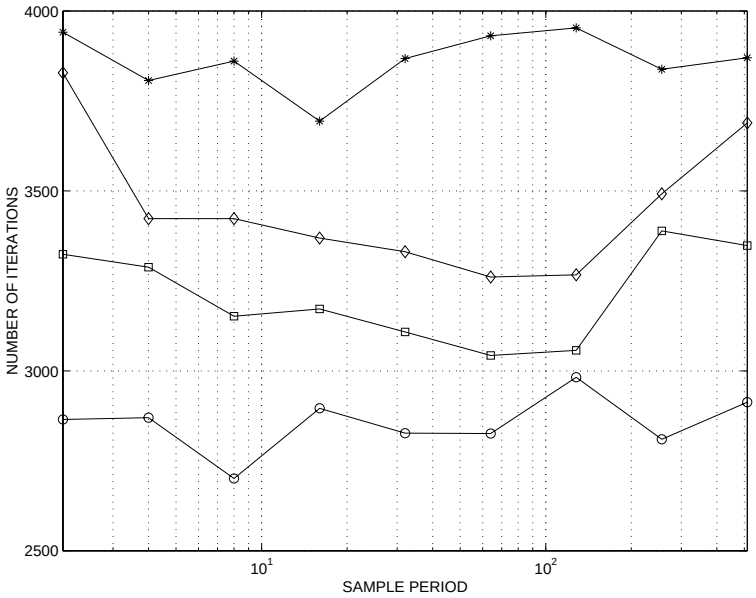


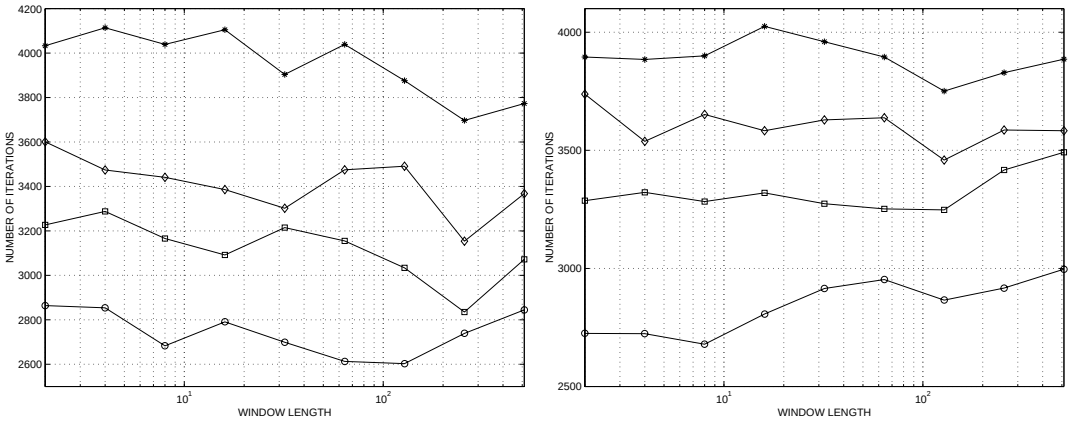
Figure 3.11: Sample period of the staircase CFA and convergence speed for -15 dB ($\circ-$), -20 dB ($\square-$), -25 dB ($\diamondsuit-$) and -30 dB ($*-$) noise level.

exponent $r(\infty)$.

Example 5. *Sample period and window length of the finite-time averages*

The first parameter modification we discuss is the sample period of the finite-time average for the staircase CFA algorithm, since for the smooth CFA this is not necesary as we change the error exponent every iteration. Our experiment considers the mean-square as average ($L = 32$), the first channel as echo-path model, and four levels of attenuated far-end signal: $a_{dB} = -15, -20, -25, -30$. The staircase CFA algorithm has the following parameters: $P = 3$, $R_1 = 4$, $R_2 = 3$, $R_3 = 2.5$, $E_M = 3$, $E_1 = 0.75$, $E_2 = 0.6944$, $E_3 = 0.64$ and $\mu = 825 \cdot 10^{-6}$. The averages of 20 runs were used to find the convergence rate needed to achieve 20 dB below the far-end signal level. The results are shown in Figure 3.11, and they suggest the sample period of few dozens of iterations as a good trade-off between computation load and convergence speed.

Next we fix the sample period to $W = 32$ and modify the window length. The rest of the parameters remain the same and the achievements are illustrated in Figure 3.12(a). Then the same investigation was carried on for the smooth CFA algorithm (Figure 3.12(b)), where we have the same parameters, plus $R_0 = 4$, $R_\infty = 2$. We found



(a) (b)

Figure 3.12: Window length of the staircase (a) and smooth (b) CFA, and convergence speed for -15 dB ($\circ-$), -20 dB ($\square-$), -25 dB ($\diamondsuit-$) and -30 dB ($*-$) noise level.

in our experiments that the results are not very sensitive to window length variation. In almost all situations both CFA algorithms behave rather similarly. However, we recommend to choose a window length close to the echo path filter length, if this information is available.

Example 6. *Initial and final error exponents influence*

The next experiment addresses the influence of the initial, and final error exponent. We consider the smooth CFA algorithm with the same parameters as in Section 5, but here we have $E_M = 1$, and $L = 32$. From Figure 3.13 for $a_{dB} = -15$, and Figure 3.14 for $a_{dB} = -20$, it is suggesting that the final error exponent modifies considerably the residual error of the CFA algorithm. Despite this, initial error exponent does not change dramatically the convergence speed. This can be explain by the fact that higher order error exponent ($r(k) > 4.5$) produce higher gradients which are very sensitive to stability issues. We can avoid this by reducing the step-size, but the final result does not justify this, as we obtain poorer results than by fourth power.

Example 7. *Experiments with added dispersion*

We fulfilled some experiments consisting in simulations performed with added dispersion to the attenuation. We select the first channel model for echo-path and the attenuation

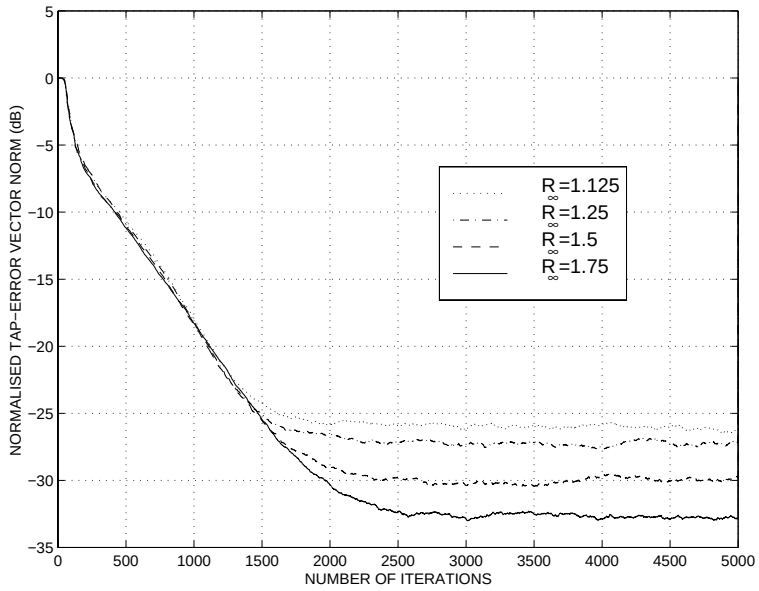


Figure 3.13: Smooth CFA with different final error exponent ($a_{dB} = -15$).

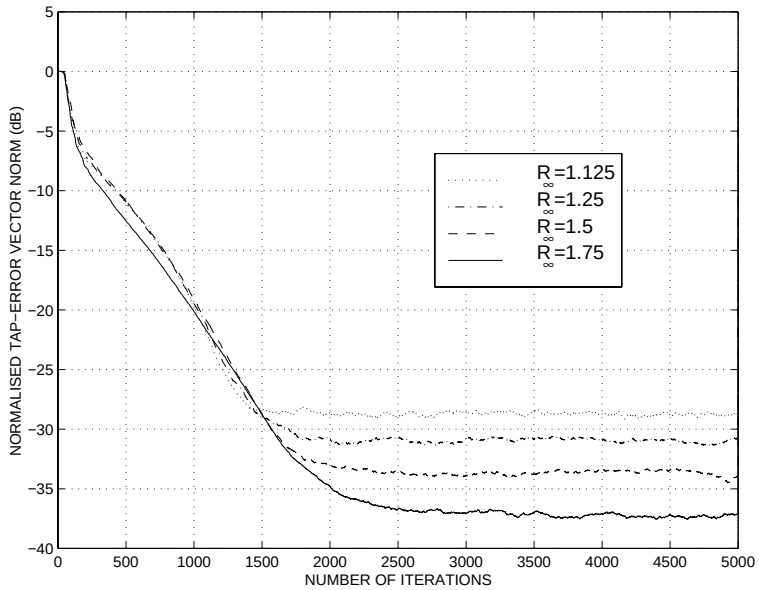


Figure 3.14: Smooth CFA with different final error exponent ($a_{dB} = -20$).

filter, and the following parameters $L = 8$, $\mu = 8 \cdot 10^{-4}$, $W = 128$, $a_{dB} = -15$. The adaptation process is the same as before, however the steady-state behaviour seems dif-

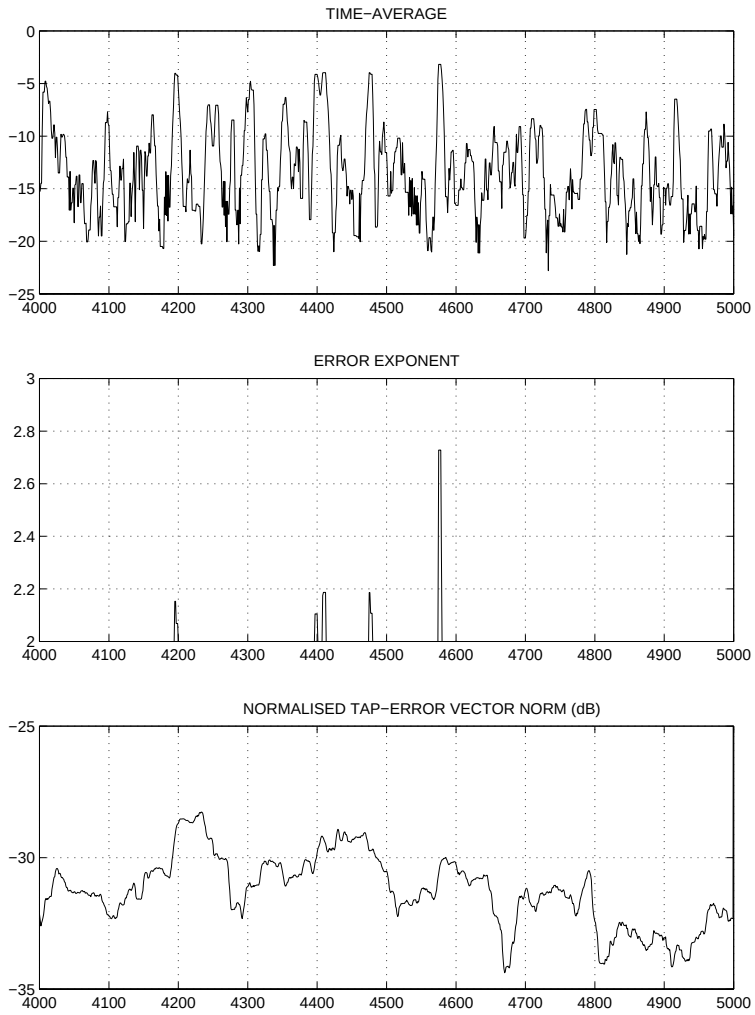


Figure 3.15: Smooth CFA with attenuated dispersion.

ferent. Thus in the following we are preocupied only for this. Figure 3.15 shows a zoom ($k = 4000, 4001, \ldots, 5000$) after we reach the convergence level. We consider remarkable to present the time-average, error exponent and learning curve of the smooth CFA algorithm. The error exponent can change after adaptation, but this happens only for few instants. Actually if we increase the window length, the error exponent remains always constant and we get in steady-state the LMS. This is the case for the staircase CFA algorithm, where the error exponent equals two in steady-state.

Example 8. *Real hybrid system identification*

The smooth CFA algorithm ($E_M = 1$, $\bar{e}_0 = E_\infty$, $L = 256$) is applied here to the

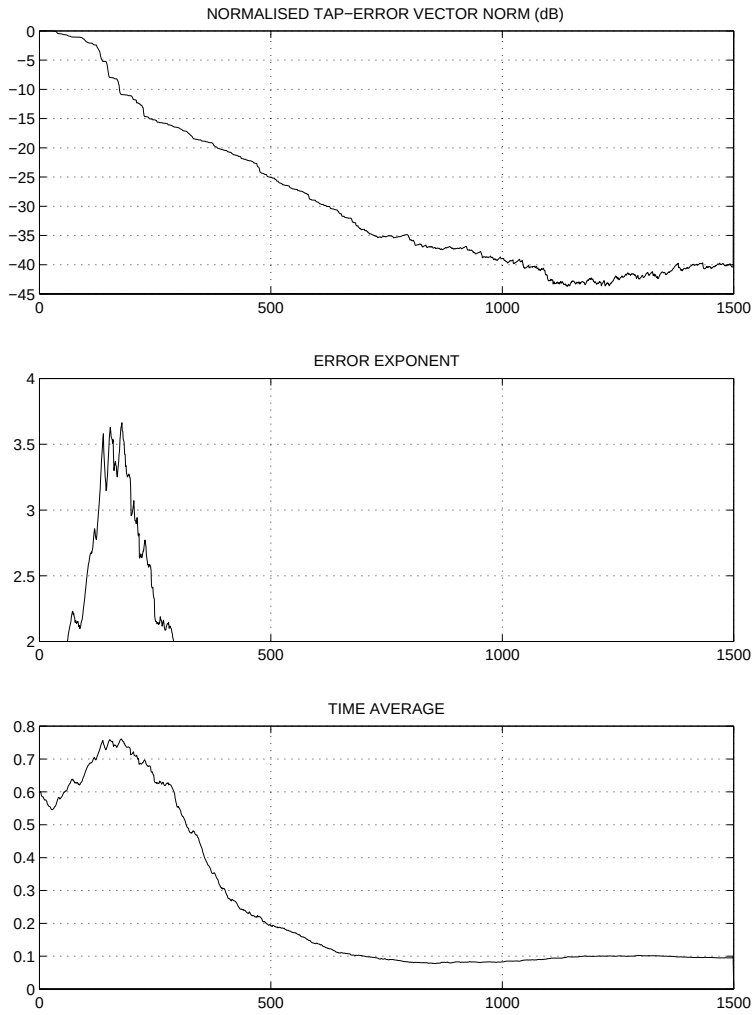


Figure 3.16: Performances versus number of iterations for the smooth CFA algorithm in the case of a real hybrid.

cancellation of the echo produced by a real hybrid of length $N = 32$ [70]. First the evolution of the normalised tap-error norm, the error exponent update and the finite-time average modification are illustrated in Figure 3.16. Then Figure 3.17 shows that for a binary input as before, the smooth CFA algorithm behaves better than the LMS algorithm. For the same step-size ($\mu = 0.0025$) which gives the same residual error as we have $R_\infty = 2$ in both cases, the convergence speed is faster for Smooth CFA. Both algorithms work in same conditions. It is also presented the effect on residual error, if the attenuated far-end signal contains a Gaussian additive noise with variance $\sigma = 0.1$.

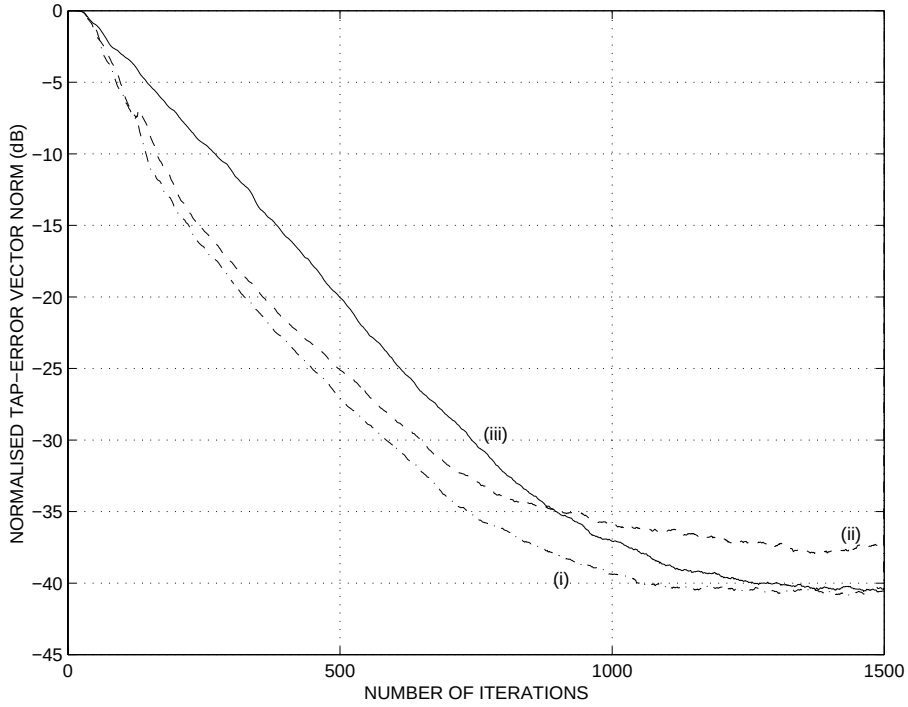


Figure 3.17: Smooth CFA with binary far-end signal (i), smooth CFA with binary far-end signal and Gaussian additive noise (ii), LMS with binary far-end signal (iii).

Example 9. *Abrupt changes in system*

The last experiment consists in tracking the first channel, where the noise is the subject of an attenuation of -15 dB. At the 5001 iteration, the echo-path has a sign change. In this framework we present a comparison between the proposed CFA and VS [32] algorithms. The staircase CFA has the following parameters: $L = W = 8$, $\mu = 8.25 \cdot 10^{-4}$, $E_M = \sqrt{3}$ and the rest of thresholds are the same as in Example 1. For the VS algorithm we choose $\mu_{min} = 8.25 \cdot 10^{-4}$, $\mu_{max} = 8.25 \cdot 10^{-3}$, $m_0 = m_1 = 3$, and $\alpha = 1.01$.

Figure 3.18 shows two different aspects of their behaviour. In the first part of the diagram it is suggested that the VS algorithm performs better in the stationary case, but the second half clearly proves that the introduced CFA algorithm manages in non-stationary case better than the VS algorithm. The distinct starting gradients differentiate the VS and CFA algorithms in stationary case, but CFA takes advantage of higher gradients after the abrupt change in the system.

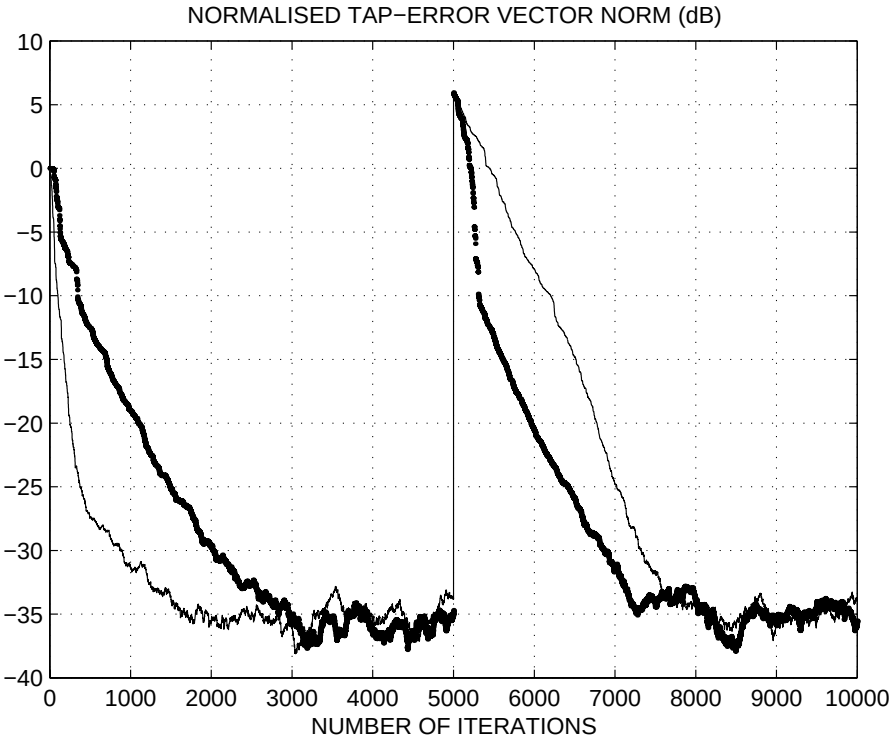


Figure 3.18: Non-stationary behaviour of the staircase CFA algorithm and variable step-size algorithm: VS (——), and CFA (▬▬).

3.2 The Gradient Approach

3.2.1 Method

In this Section our method is based on the analysis of the gradient of the cost function $J_r = E[|e(k)|^r]$, where the power is a function only of the instantaneous error[3]: $r = F[e(k)]$. In order to derive the CFA algorithm, we first compute

$$\frac{\partial J_r}{\partial \hat{\boldsymbol{h}}} = \frac{\partial \{E[|e(k)|^r]\}}{\partial \hat{\boldsymbol{h}}} \tag{3.14}$$

with respect to each element of the general impulse response. Differentiating (3.14) we obtain

$$\frac{\partial J_r}{\partial \hat{\boldsymbol{h}}} = E[\frac{\partial}{\partial \hat{\boldsymbol{h}}}(|e(k)|^r)] = E[\frac{\partial}{\partial e(k)}(|e(k)|^r) \cdot \frac{\partial e(k)}{\partial \hat{\boldsymbol{h}}}].$$

[3]Our theoretical approach makes the exponent r a function of the instantaneous error. For practical reasons, as in Section 3.1, the employed relation will be of the form $r(k+1) \equiv F[\bar{e}(k)]$, where $\bar{e}(k)$ is a time-average of the instantaneous error.

From Appendix 3.3 we have

$$\frac{\partial}{\partial e(k)}(|e(k)|^r) = |e(k)|^{r-2}\left\{e(k)\cdot r + |e(k)|^2 \ln|e(k)|\frac{dr}{d|e(k)|}\right\}, \tag{3.15}$$

and using

$$\frac{\partial e(k)}{\partial \hat{\boldsymbol{h}}} = \frac{\partial}{\partial \hat{\boldsymbol{h}}}(f(k) - \Delta \boldsymbol{h}^t(k)\boldsymbol{x}(k)) = -\boldsymbol{x}(k), \tag{3.16}$$

we obtain

$$\frac{\partial J_r}{\partial \hat{\boldsymbol{h}}} = E[-|e(k)|^{r-2}\{e(k)\cdot r + |e(k)|^2 \ln|e(k)|\frac{dr}{d[e(k)]}\}\cdot \boldsymbol{x}(k)]. \tag{3.17}$$

The major change in the gradient expression, compared to the LMS or the LMF case is the appearance of the CFA kernel

$$\mathcal{K}(r,e,k) = e(k)\cdot r + |e(k)|^2 \ln|e(k)|\frac{dr}{d[e(k)]}.$$

With this notation, Equation (3.17) can also be written as

$$\frac{\partial J_r}{\partial \hat{\boldsymbol{h}}} = -E[\mathcal{K}(r,e,k)\cdot |e(k)|^{r-2}\cdot \boldsymbol{x}(k)], \tag{3.18}$$

or in matrix form

$$\frac{\partial J_r}{\partial \hat{\boldsymbol{h}}} = -E\begin{pmatrix} \mathcal{K}(r,e,k)\cdot |e(k)|^{r-2}\cdot x(k) \\ \mathcal{K}(r,e,k)\cdot |e(k)|^{r-2}\cdot x(k-1) \\ \vdots \\ \mathcal{K}(r,e,k)\cdot |e(k)|^{r-2}\cdot x(k-N+1) \end{pmatrix}.$$

As we do not know the signal statistics required by the previous equation, we shall use a training sequence, which in the case of a data echo canceller can be the data sequence itself. The coefficients of the adaptive filter are computed via the method of steepest descent [35] of the new cost function $J_{r(k)} = E[|e(k)|^{r(k)}]$. The gradient given by Equation (3.18) at kth time instant becomes the noisy or the stochastic gradient, after removing the statistical expectation. At each step we change the current vector by an amount proportional to the negative of the gradient vector:

$$\hat{\boldsymbol{h}}(k+1) = \hat{\boldsymbol{h}}(k) - \mu \times \frac{\partial J_r}{\partial \hat{\boldsymbol{h}}}. \tag{3.19}$$

Thus the proposed gradient update:

$$\hat{\boldsymbol{h}}(k+1) = \hat{\boldsymbol{h}}(k) + \mu\cdot\mathcal{K}(r,e,k)\cdot |e(k)|^{r-2}\cdot \boldsymbol{x}(k) \tag{3.20}$$

holds for every $r > 1$. As we do not know yet the shape of the desired cost function, we shall use at every kth iteration gradients that preserves collinearity and direction of the corresponding constant metric algorithms NQSG$r(k)$, where $r(k) > 1$. For that reason we reconsider the CFA kernel, and we will force the equality of $\mathcal{K}(r, e, k)$ by a positive constant[4] times the product instantaneous error - exponent error:

$$e(k)r + |e(k)|^2 \ln |e(k)| \frac{dr}{d[e(k)]} = \beta e(k) r, \tag{3.21}$$

where β is called the CFA constant and it has been introduced mainly for two reasons: to control the error exponent adaptation process and to assure that the resulting gradient direction points towards the inside of the bowl of the corresponding constant metric algorithms.

We distinguish the following cases:

- $\beta = 0$; in this case there is no adaptation.
- $0 < \beta < 1$; we expect smaller gradients than the constant metric algorithms.
- $\beta = 1$; we retrieve LMS ($r(k) = 2$), LMF ($r(k) = 4$), NQSGr ($2 < r < 3$), LMP ($r(k) = \text{constant} = r \in \mathbb{N}$).
- $\beta > 1$; the error surface has steeper gradients than the constant metric algorithms.

For us the most interesting case is the last one. The solution of Equation (3.21) is obtained in Appendix 3.3 and the proposed cost function adaptation algorithm might follow the mapping

$$r(k+1) \equiv F[e(k)] = r(0) \Big| \frac{|e(0)|_{dB}}{|e(k)|_{dB}} \Big|^{\beta - 1}, \tag{3.22}$$

where the gradient update should be computed with[5]:

$$\hat{\boldsymbol{h}}(k+1) = \hat{\boldsymbol{h}}(k) + \mu \beta r(k) |e(k)|^{r(k)-2} e(k) \boldsymbol{x}(k). \tag{3.23}$$

We recall also from [85] the following:

Remark 5. *The adjustment of $r(k)$ must be done using finite-time average characteristics $\bar{e}(k)$ rather than the instantaneous values of the error $e(k)$.*

[4] For a negative constant the resulted gradient direction is not pointing towards the inside of the bowl.

[5] Apparently the effect of the CFA constant is only the multiplication of the error surfaces gradients using an error norm of $|e(k)|^r$. Actually the parameter β has a more important interference in Equation (3.23) through the error exponent r.

This will be more clear than in the case of stationary approaches, if we focus once again on Equation (3.22). The sensitivity [16] of the error exponent $r(k+1)$ with respect to the magnitude of error $|e(k)|_{dB}$ is given by

$$S^{r(k+1)}_{|e(k)|_{dB}} = \frac{|e(k)|_{dB}}{r(k+1)} \cdot \frac{\partial r(k+1)}{\partial |e(k)|_{dB}} = \beta - 1. \tag{3.24}$$

Note that if β is equal to one, and this is the case for LMS, LMF and NQSGr, it is obvious that there is no sensitivity of the error exponent with respect to the error. If $\beta \approx 1$ the influence is smaller. It results that the CFA will iterate in small steps, that is, for a long time. If β increases, the CFA effect is more important.

Therefore (3.24) suggests that every decade change in the magnitude of the error will change the error exponent correspondingly by $\beta - 1$ units. Thus a direct implementation of the relationship (3.22) could lead to divergence and suggests the introduction of a smooth estimate. However,

Remark 6. *The guess of $\bar{e}_k$ needs a special attention to statistical averages and to follow statistical variations.*

This will be recalled in Section 3.2.2.

As a consequence of these aspects, the error exponent might be updated by:

$$r(k+1) \equiv F[e(k)] = \begin{cases} r(0), & \text{if} \quad k \leq K, \\ r(K)\left|\dfrac{|\bar{e}(K)|_{dB}}{|\bar{e}(k)|_{dB}}\right|^{\beta-1}, & \text{if} \quad k > K, \end{cases} \tag{3.25}$$

where K is a positive integer required by the delay of computing the finite-time average $\bar{\mathrm{e}}(k)$. From the relationship (3.25), it follows that $r(0) = r(K)$, and for $k > K$ the values $|\bar{\mathrm{e}}(k)|_{dB}$ and $|\bar{\mathrm{e}}(K)|_{dB}$, should have the same sign. It follows that we have to keep the error exponent constant until we obtain an integer K for which $|\bar{\mathrm{e}}(K)|_{dB} < 0$. Moreover, unless the term on the right-hand side of (3.25) can be assessed, the range of the error exponent should be bounded to avoid instability. In order to obtain the convergence of the algorithm, a condition is the convergence of the sequence $\{r(k)\}$. One possibility is to design a monotonic time-average during the adaptation time[6]. For this reason enforcing the monotonicity of the sequence can be used, though other less restrictive criterion can guarantee the convergence.

[6]If the relation between $r(k)$ and k during the convergence period is linear [59], the LCFA algorithm results (Section 3.2.5).

3.2.2 Error and exponent update

We now reconsider the CFA algorithm given by Equations (3.23) and (3.25), and we shall concentrate on the mechanism of computing the error exponent $r(k)$ taking into account the noisy characteristics of the error. The problem we address now is how to obtain an error estimate, which is as noise free as possible, and monotonically decreasing during adaptation, to be used in the implementation of the CFA algorithm. In [21] the error estimate was computed in several different ways. First the running average of the modulus of the instantaneous error and the log running average of the squared instantaneous error were used. A small window means a noisy estimate, a large window introduces delay. Better performances from the stability point of view have been obtained if this estimate is computed only once at every W iterations (Section 3.1). Another estimate [22] was obtained via the normalised tap-error vector norm.

We propose a different solution for this problem [59], which will be used mainly in the simulations of this Section. The idea is to employ the technique of the envelope detector in classical amplitude modulation. We shall pass the modulus of the instantaneous error through a first order recursive digital filter, the equivalent of the low-pass RC filter. Let L ($L \gg 1$) be its time constant[7], and $\hat{e}(0)$ the initial error estimate, normalised to the maximum of the driving signals. From the transfer function in s of the analogue filter [31], we obtain the linear difference equation of the recursive digital filter:

$$\begin{gathered} 0 < \hat{e}(0) < 1, \\ \hat{e}(k) = \tfrac{L}{L+1} \cdot \hat{e}(k-1) + \tfrac{1}{L+1} \cdot |e(k)|,\ k \geq 1, \\ \bar{e}_{dB}(k) = 20 \cdot \log_{10}(\hat{e}(k)),\ k \geq 1. \end{gathered}$$

Clearly we have $0 < \hat{e}(k) < 1$ for any $k \in \mathbb{N}$, if $e(k)$ is also normalised, and we can start the CFA algorithm from the beginning. Note that if $|e(k)| \geq a$ for all $k \in \mathbb{N}$, then so is $|\hat{e}(k)|$. Moreover, if $\lim_{n\to\infty} |e(k)|$ exists, then $\lim_{n\to\infty} \hat{e}(k)$ exists and they are equal. It follows that we have

$$\frac{r(\infty)}{r(0)} = \Big| \frac{|\hat{e}(0)|_{dB}}{|\hat{e}(\infty)|_{dB}} \Big|^{\beta-1}, \tag{3.26}$$

which suggests the design process of the error mapping[8].

[7] In the case of envelope detector, the time constant is bounded by the carrier period and respectively the period correspondingly to message bandwidth [34]. In our case the time constant of the recursive digital filter should be between the quasi-period of zero crossings and the quasi-period of the envelope of the instantaneous error.

[8] Note that if $\beta \gg 1$, then $|\hat{e}(0)|_{dB} \approx |\hat{e}(\infty)|_{dB}$. Also if $\beta \approx 1$, then $||\hat{e}(0)|_{dB}| \ll ||\hat{e}(\infty)|_{dB}|$.

3.2.3 Parameter selection

Next we define the parameters of the adaptation algorithm. The recursive digital filter used in our simulations is of special interest.

The initial error estimator $\hat{e}(0)$ will be selected equal to the attenuation a, because we expect that in the steady-state the error is given mainly by the far-end signal. Another assumption this choice was based on is that the new initialisation will be started after the previous one attains its steady state.

The appropriate time-constant L of the recursive filter can be guessed using the fact that the period of adaptation should be almost equal with the decay period of the error estimator.

Thus we proceed in the following manner:

- First we take the error records of the LMS and LMF simulations, and compute the output of the recursive filter $\hat{e}(k)$ with $\hat{e}(0) = a$ for different L.

Figure 3.19 shows the graphically result of such an experiment for the LMS algorithm ($\mu = 0.009$), and the first echo path channel ($N = 32$, $A = -60$, $a_{dB} = -15$) and the time-constant $L = 100$. The normalised tap-error norm and its 20 times average are presented in Figure 3.19(a), and Figure 3.19(b) respectively. It is clear now that the error estimator starts from value a, attains a maximum value $\hat{e}_{max}$, then decreases to a again. This is illustrated in Figure 3.19(c). In this case, as the period of adaptation is longer than the sum of the increasing and decreasing time of the output of the recursive filter, we will need to increase the time-constant of the filter in order to obtain a faster adaptation.

Another important issue is the relation between the filter time-constant L and the maximum output $\hat{e}_{max}$. Taking 20 averages of the output error estimator curve, for different time-constant L and noise level a, the result is similar to that from Figure 3.19(d). Thus we can obtain the average of the maximum value of the error estimator $\hat{e}_{max}^{LMS}$ for LMS (Table 3.1), and $\hat{e}_{max}^{LMF}$ for LMF (Table 3.2). We require this value for our CFA algorithm, and we suggest

- to take $\hat{e}_{max}$ as the geometric mean of $\hat{e}_{max}^{LMS}$ and $\hat{e}_{max}^{LMF}$.

This will help us

- to guess the parameter β

from relationship (3.26) between the minimum r_{min} and the maximum r_{max} error exponents, rewritten as following

$$\frac{r_{min}}{r_{max}} = \left| \frac{|\hat{e}_{max}|_{dB}}{|\hat{e}_{min}|_{dB}} \right|^{\beta-1}, \tag{3.27}$$

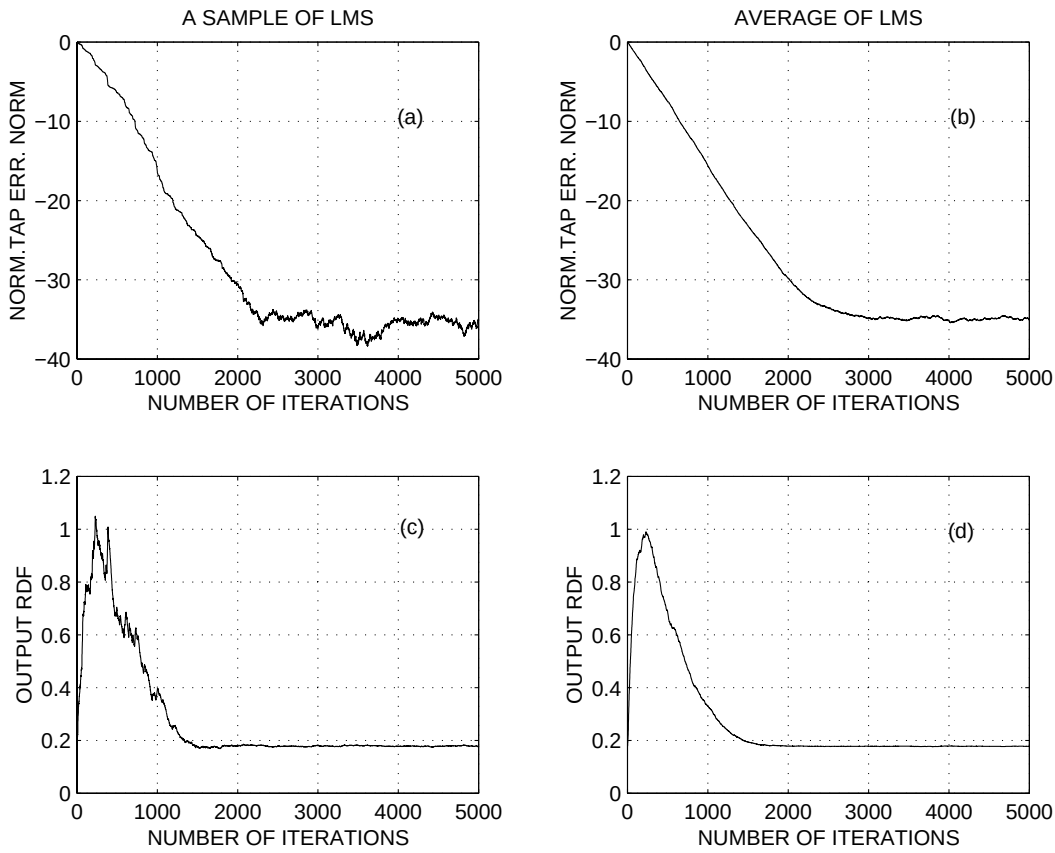


Figure 3.19: The normalised tap error norm for one LMS run (a) and 20 times average LMS runs (b), the error estimator (c) and its 20 times average (d).

where, usually $r_{min} = 2$, $r_{max} = 4$, and $\hat{e}_{min} = a$.

3.2.4 Simulation results

In this Section we present a wide range of experimental results to confirm the theoretical achievements of previous Sections. Simulations demonstrate the performance of the proposed CFA algorithm in comparison with the standard LMS and LMF algorithms, and the variable step-sizes algorithms proposed in [32, 46]. The error exponent update was implemented as follows:

$$\hat{e}(0) = a, \quad r(0) = r(1) = 2;$$

$$\text{for all } k \geq 1: \quad \hat{e}(k) = (1 - \tfrac{1}{L+1}) \cdot \hat{e}(k-1) + \tfrac{1}{L+1} \cdot |e(k)|, \quad r(k+1) = r(0) \left[\frac{\log_{10} |\hat{e}(0)|}{\log_{10} |\hat{e}(k)|} \right]^{\beta - 1}.$$

L	$a = -15$ dB	$a = -20$ dB	$a = -25$ dB	$a = -30$ dB
100	1.01444	0.97740	0.93051	0.97059
200	0.81536	0.78735	0.77736	0.77596
300	0.72905	0.69481	0.67314	0.67779
400	0.65906	0.63000	0.61487	0.60922
500	0.60399	0.57175	0.55808	0.54122
600	0.56890	0.53035	0.50823	0.50059
700	0.53525	0.49918	0.47375	0.45939
800	0.51028	0.46336	0.44299	0.43097
900	0.48424	0.44122	0.41911	0.40318
1000	0.46483	0.41758	0.39546	0.38467
1100	0.44652	0.39990	0.37016	0.36320
1200	0.43220	0.38677	0.36053	0.34456
1300	0.41861	0.36461	0.34338	0.32853
1400	0.40388	0.35596	0.33058	0.31732
1500	0.39500	0.34231	0.31894	0.30121
1600	0.38778	0.33606	0.31025	0.29088
1700	0.37562	0.32454	0.29618	0.28031
1800	0.36758	0.31467	0.28476	0.27419
1900	0.36252	0.30493	0.28093	0.26404
2000	0.35406	0.29945	0.26999	0.25425

Table 3.1: The maximum value of the error estimator $\hat{e}_{max}$ for the first channel and $\mu = 0.9$ with LMS simulation

If $L + 1$ is an integer power of 2, the implementation of the recursive digital filter is simplified since the divisions are reduced to shifts. Also the error exponent update is simple, if $\beta - 1$ is integer. Note that in all the following experiments we did not bound the power of the cost function.

The first example presented is the behaviour of the proposed CFA algorithm when the echo-path has a sign change. Then a set of simulations will show how the proposed algorithm works when different parameters are modified. We always change only one parameter, and the resulting parameters are not anymore exactly those derived from Equation (3.27), Table 3.1, and Table 3.2. The learning curves obtained for normalised tap-error norm are the average of 20 independent runs. We also present for every type of experiment one sample of the error exponent update. A short comparison with LMS, LMF and variable step-sizes algorithms is also included. In all cases the echo path channel

L	$a = -15$ dB	$a = -20$ dB	$a = -25$ dB	$a = -30$ dB
100	0.57036	0.54255	0.51159	0.52769
200	0.46009	0.42537	0.40676	0.38127
300	0.40847	0.36982	0.35117	0.31896
400	0.35913	0.31944	0.29961	0.29818
500	0.33267	0.29061	0.27430	0.26680
600	0.32737	0.28424	0.25453	0.24781
700	0.30331	0.25862	0.24558	0.23723
800	0.29607	0.24971	0.22361	0.21490
900	0.28928	0.24044	0.21733	0.21108
1000	0.28575	0.23721	0.21003	0.20377
1100	0.27605	0.22714	0.20233	0.19648
1200	0.26803	0.21848	0.19525	0.18429
1300	0.26590	0.21446	0.18781	0.17831
1400	0.25851	0.20874	0.18447	0.17626
1500	0.25522	0.20372	0.17931	0.16730
1600	0.25309	0.20048	0.17449	0.16312
1700	0.25125	0.19714	0.17250	0.15901
1800	0.25011	0.19636	0.16837	0.15511
1900	0.24184	0.18858	0.16508	0.15127
2000	0.24095	0.18687	0.15848	0.14889

Table 3.2: The maximum value of the error estimator $\hat{e}_{max}$ for the first channel and $\mu = 0.9$ with LMF simulation

is single pole single zero digital filter.

Example 10. *The gradient approach of CFA algorithm in a non-stationary situation*

First we are interested in the behaviour of the gradient CFA algorithm in a non-stationary case. The experiment consists in tracking the channel used before in parameter selection (the attenuated level is -60 dB after the 32nd sample), where the noise is subject of an attenuation of -15 dB. At the 5001 iteration, the echo-path has a sign change. The parameters of the CFA algorithm and of the recursive filter filter are $L = 1023$ and $\beta = 2$, and this choice reduces the exponent error update to a simple implementation, as we have $r(k) = -1.5/\log_{10}|\hat{e}(k)|$, for all $k \geq 1$. The step-size of the CFA algorithm was kept at $\mu = 0.009$.

Figure 3.20 shows clearly that the proposed CFA algorithm is applicable to non-

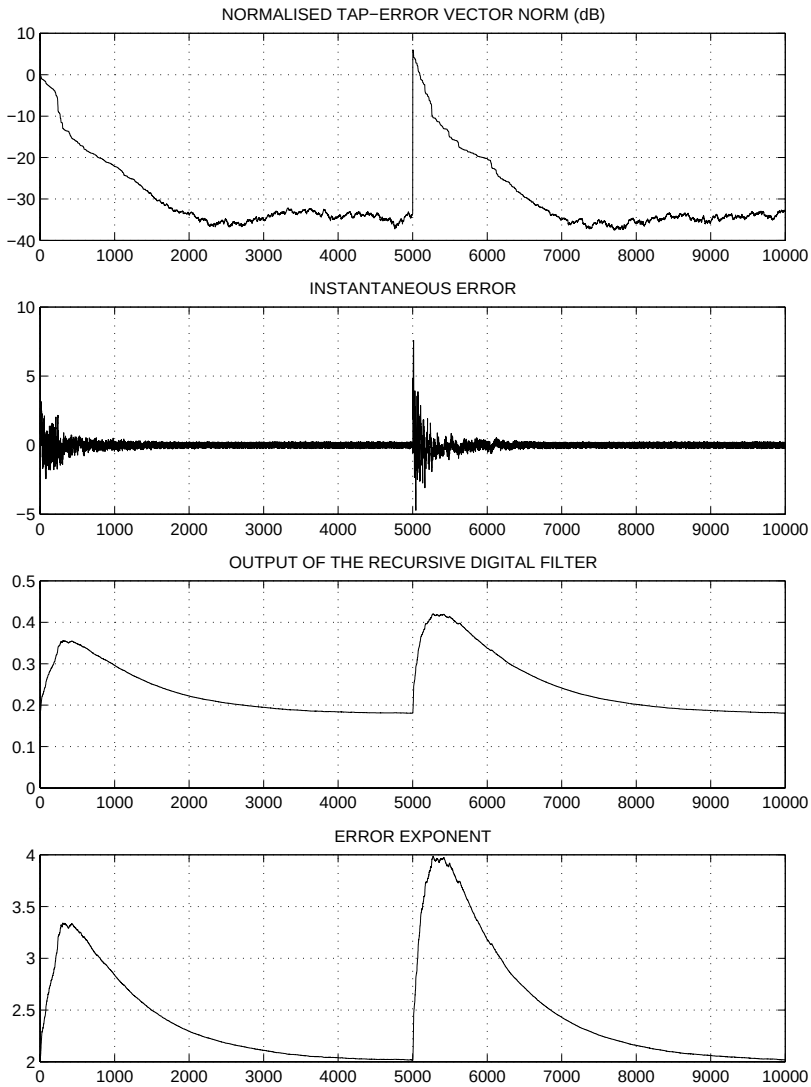


Figure 3.20: Several characteristics of the CFA algorithm versus number of iterations. The framework was for the first channel, in case of a far-end signal subject of an attenuation of -15 dB, for $L = 1023$ and $\beta = 2$.

stationary situations. Moreover, we can compare the period of the error exponent update with the adaptation time, and the duration of the recursive filter output. Also the first maximum of the recursive filter output is closer to the geometric mean of corresponding values from Table 3.1 and Table 3.2, for $L = 1000$ and $L = 1100$.

Example 11. *The effect of the algorithm and channel parameters*

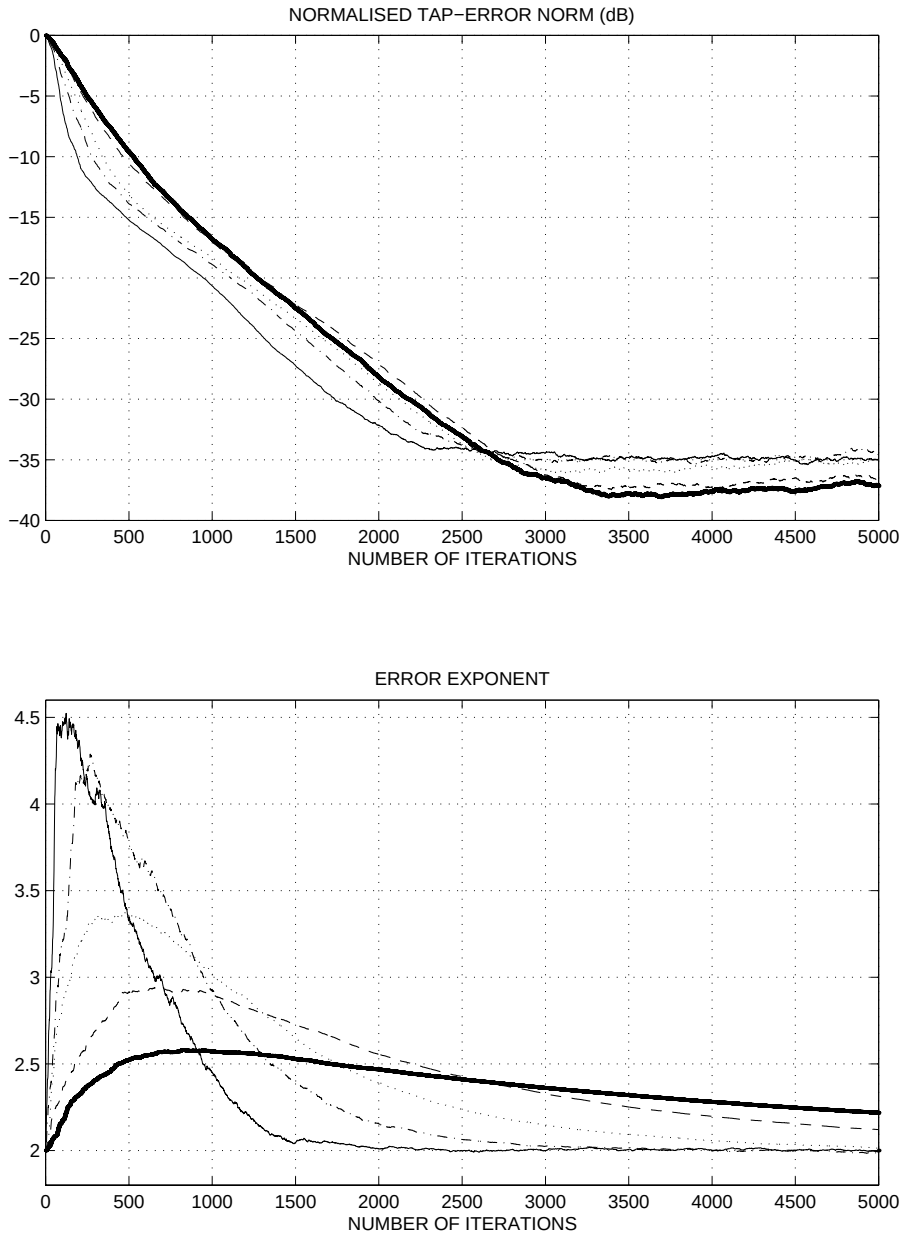


Figure 3.21: The learning curves and the error exponent update for different time-constants of the recursive filter: $L = 255$ (——), $L = 511$ ($\cdot - \cdot - \cdot$), $L = 1023$ ($\cdot \cdot \cdot \cdot \cdot$), $L = 2047$ ($- - - -$), $L = 4095$ (▬▬).

The first adjustment we tried was to vary the recursive filter time-constant such that $L + 1 = 2^m, m = 8, 9, 10, 11, 12$. The results of this experiment are shown in Figure 3.21. A reduction of the parameter L increases the peak and decreases the period of the error

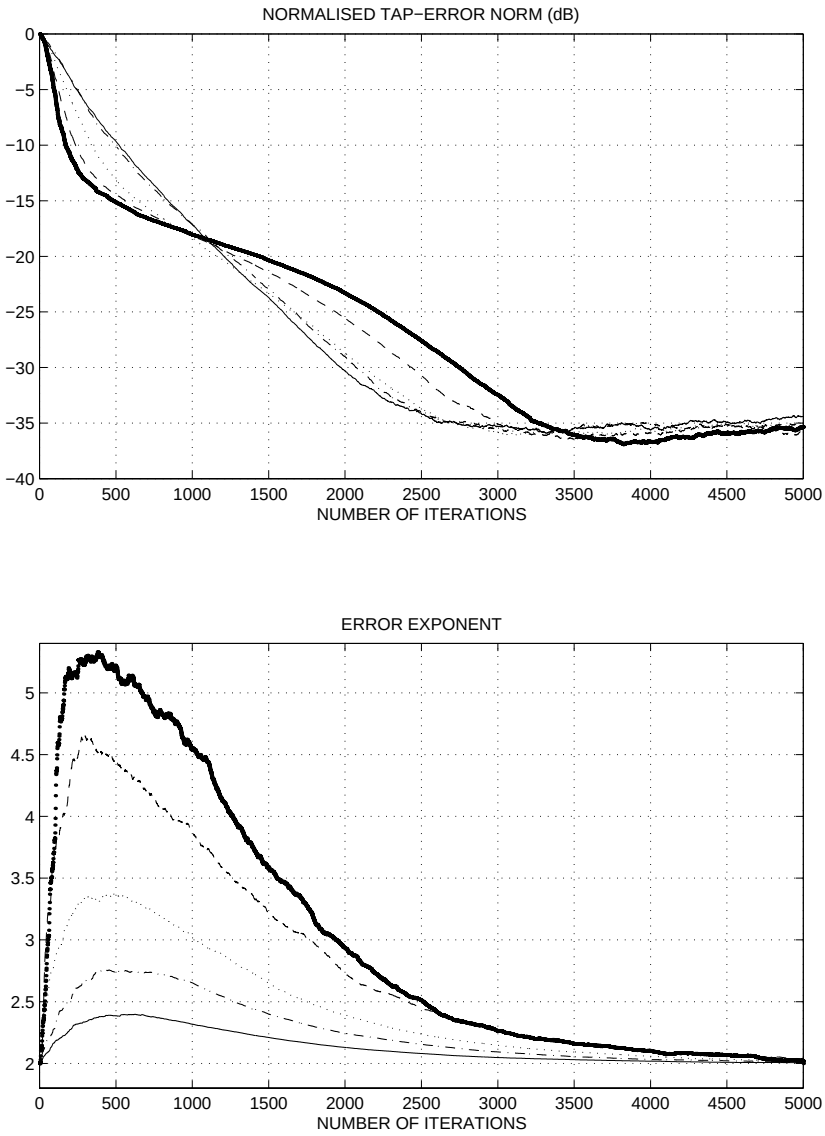


Figure 3.22: The learning curves and the error exponent update for different CFA constants: $\beta = 1.25$ (——), $\beta = 1.5$ ($\cdot - \cdot - \cdot$), $\beta = 2$ ($\cdot\cdot\cdot\cdot\cdot\cdot$), $\beta = 3$ ($- - - -$), $\beta = 4$ (▬▬).

exponent update. As a consequence the CFA algorithm converges faster. If we decrease the time-constant of the recursive filter further, the CFA algorithm could diverge. Alternatively, if we increase the L parameter, the output of the recursive filter will be smoothed and the behaviour of the CFA algorithm will be close to that of LMS.

The CFA constant β has greater affect on the quality and the speed of the adaptation

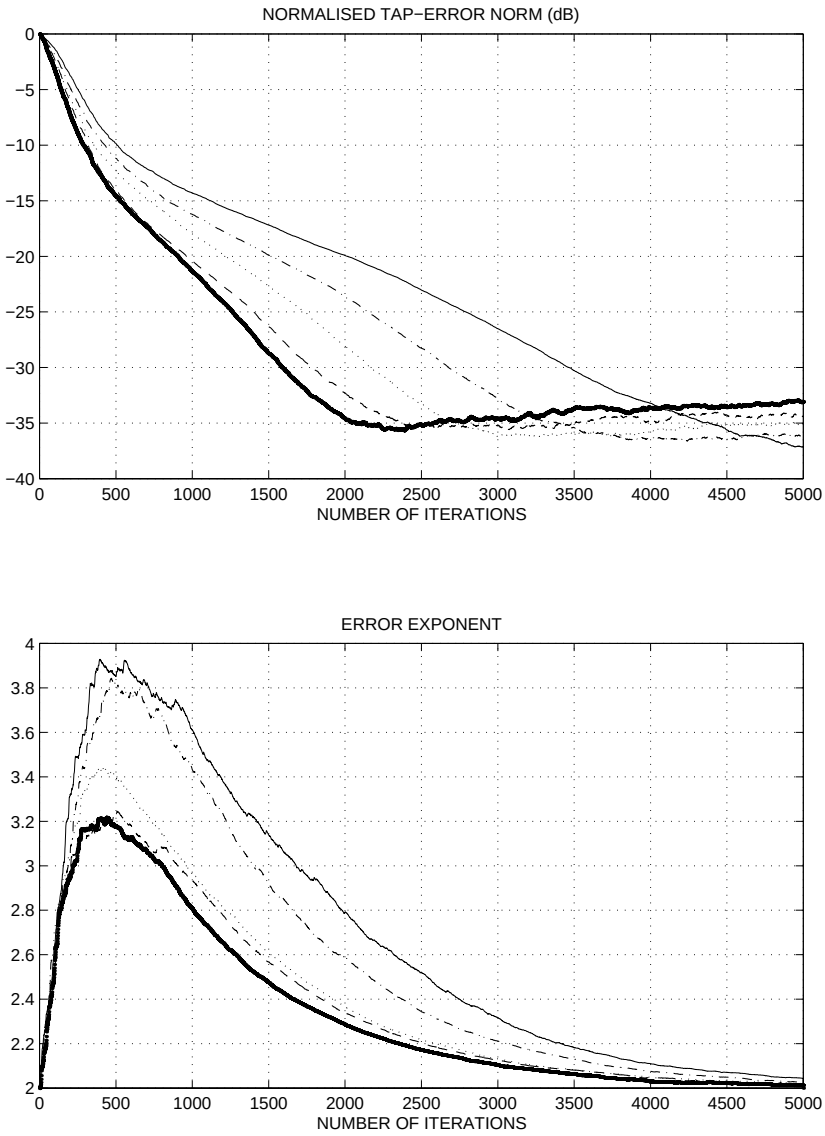


Figure 3.23: The performances for different step-sizes of the CFA algorithm: $\mu = 0.0005$ (——), $\mu = 0.0007$ ($\cdot - \cdot - \cdot$), $\mu = 0.0009$ ($\cdot\cdot\cdot\cdot\cdot\cdot$), $\mu = 0.0011$ ($- - - -$), $\mu = 0.0013$ (▬▬).

as can be seen in Figure 3.22. For our purposes we select the following error exponent update constants: $\beta - 1 = \frac{1}{4}, \frac{1}{2}, 1, 2, 3$ (which gives the simplest implementations, as we have integer or square roots in the error exponent update). Now it is clear that increasing the constant β of the CFA algorithm, we increase the peak of the error exponent update. If the time-constant of the recursive filter is kept constant, the period of adaptation will

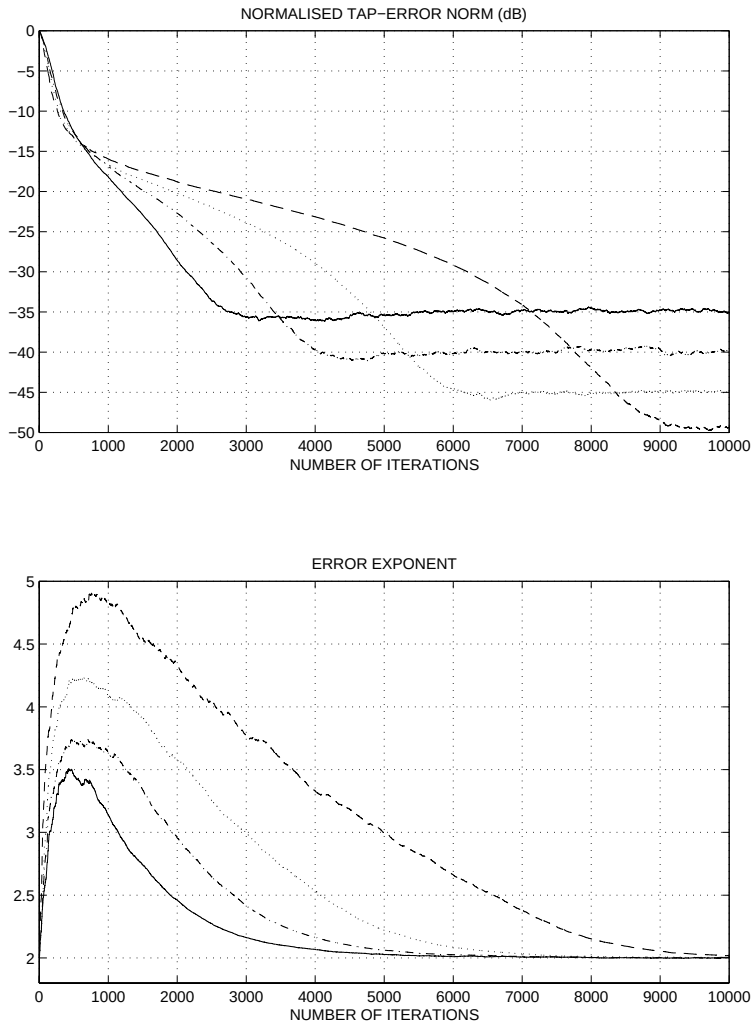
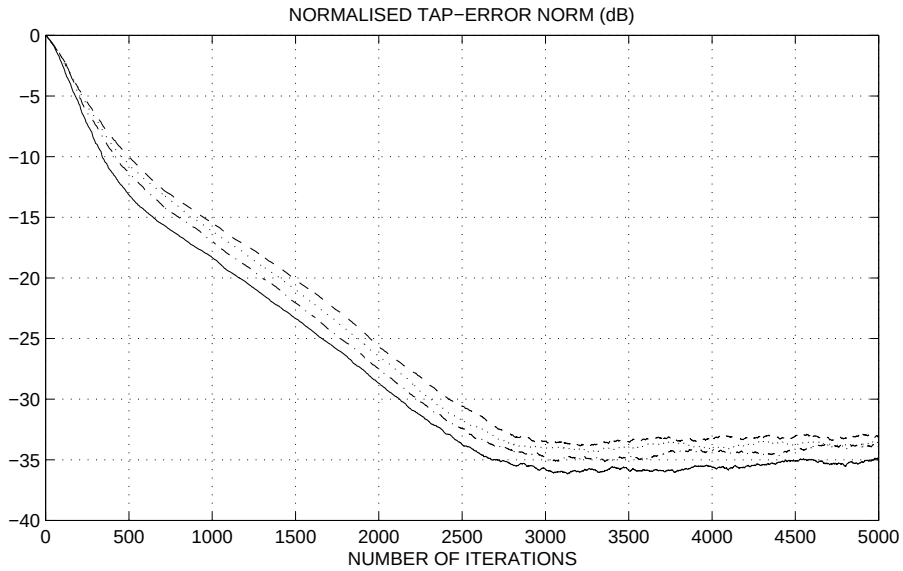


Figure 3.24: The learning curves and the error exponent update for different noise levels: $a_{dB} = -15$ (——), $a_{dB} = -20$ ($\cdot - \cdot - \cdot$), $a_{dB} = -25$ ($\cdot \cdot \cdot \cdot \cdot$), $a_{dB} = -30$ ($- - - -$).

increase. However, another combination $\beta \leftrightarrow L$ could lead to better results. We can say that decreasing L, and increasing β, the results are qualitatively almost the same.

The step-size of the algorithm also modifies the evolution of adaptation (Figure 3.23). An interesting fact is that in some cases the learning curve attains a lower level during adaptation than in the steady state (e.g. $\mu = 0.0013$, where the performance is -35 dB around 2000-2500 iterations, and the steady-state is about -33 dB). This can be explained by the variation of the error exponent during adaptation and steady-state, and as a result their correspondent residual errors differ.



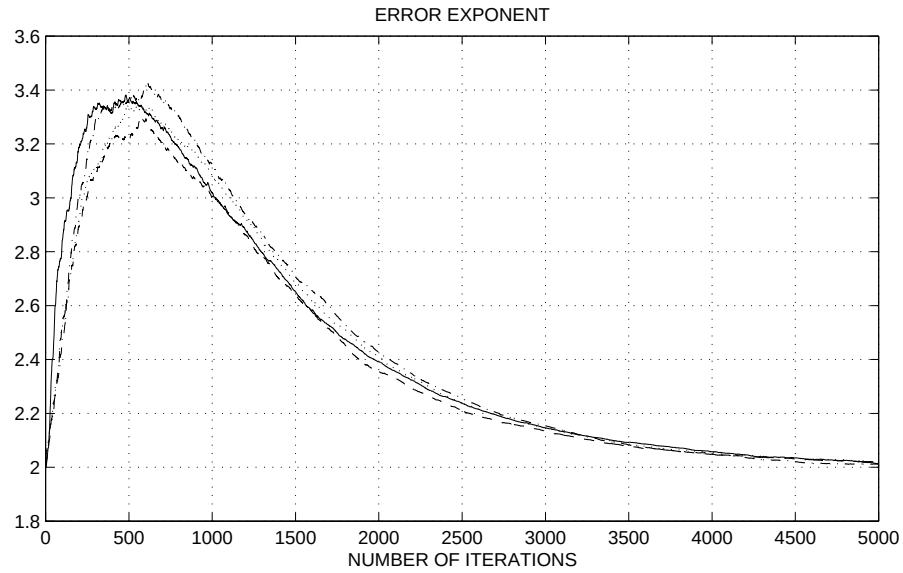


Figure 3.25: The learning curves and the error exponent update for different levels of attenuations at the 32nd sample: $A = -60$ (——), $A = -80$ ($\cdot - \cdot - \cdot$), $A = -100$ ($\cdot\cdot\cdot\cdot\cdot$), $A = -120$ ($- - - -$).

Figure 3.24 illustrates the convergence performance of the CFA algorithm when the level of the noise in the far-end signal is changed. The shape of the error exponent update is similar, however the results of adaptation differ greatly. Now it is clear that the

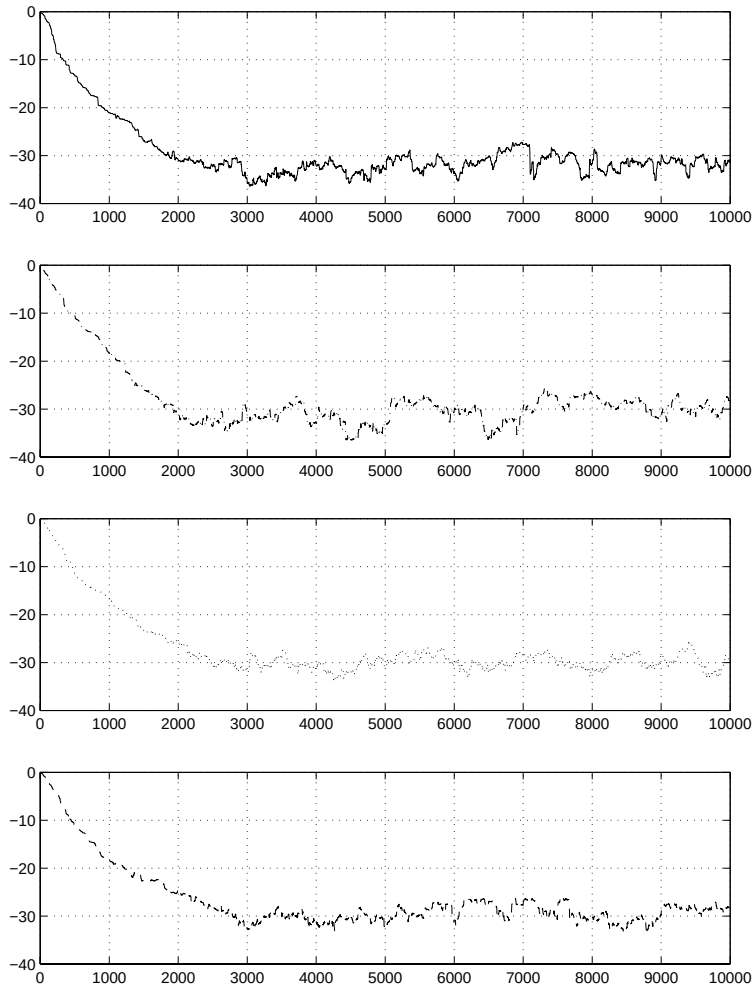


Figure 3.26: Normalised tap-error norm (dB) versus number of iterations of the CFA algorithm for different dispersion filters: $A = -60$ (——), $A = -80$ ($\cdot - \cdot - \cdot$), $A = -100$ ($\cdot \cdot \cdot \cdot \cdot \cdot$), $A = -120$ ($- - - -$).

parameter selection is related to the far-end signal characteristics. These plots show that the important fact which provides a faster convergence is the optimal length of the error exponent update, though its peak could also have a contribution.

From Figure 3.25 it appears that the algorithm is not very sensitive to the alteration of the single pole parameter of the channel. For our simulations we select four echo path single pole single zero filters corresponding to an attenuated level of -60, -80, -100 and -120 dB, at the 32nd sample. For all of these, the speed of convergence and residual error are almost the same.

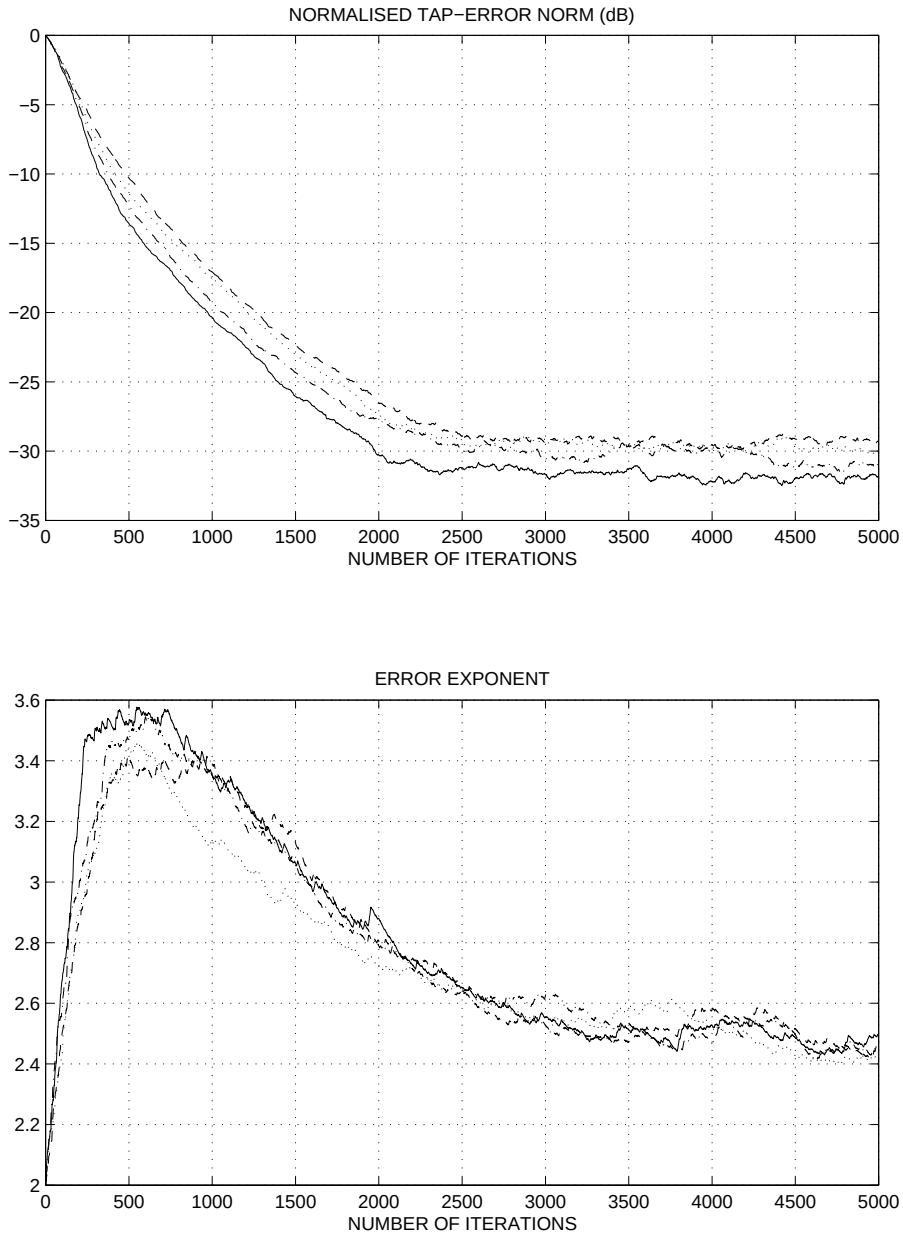


Figure 3.27: The learning curves and the error exponent update for different dispersion filters: $A = -60$ (——), $A = -80$ ($\cdot - \cdot - \cdot$), $A = -100$ ($\cdot\cdot\cdot\cdot\cdot$), $A = -120$ ($- - - -$).

Finally, simulations with added dispersion are shown (Figure 3.27). The far-end dispersion filters were selected identical with the above mentioned echo-path channels. The noise in the error exponent and residual error (Figure 3.26) is larger now. The CFA algorithm is able to track the channel and far-end signal changes.

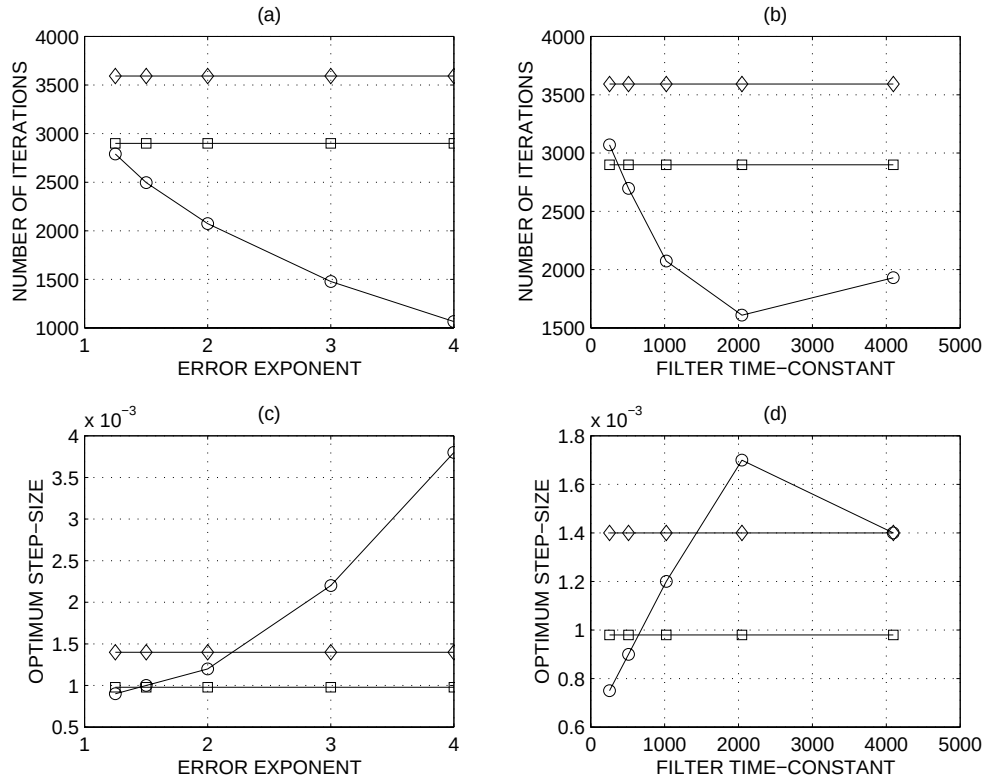


Figure 3.28: Simulation results for the case when the far-end signal level is at -15 dB as compared with that of the near-end signal, for LMS (- □), LMF (- ◇) and CFA (- ○): the fastest convergence for different values β (a) and L (b), respectively their optimum step-sizes: (c) and (d).

Example 12. *A comparison of CFA and quadratic algorithms*

The first procedure for distinguishing the performances of the proposed algorithms is to compare convergence rates for a given convergence level. The results presented in this dissertation are for the case where the steady-state were achieved 20 dB below the far-end signal level, which is at -15 dB. For some algorithms we cannot achieve this convergence level, as they become unstable. However, for their maximum step-size we get a lower convergence level, and the convergence time is measured when learning curves cross the -35 dB level. Each of the learning curves were obtained using an ensemble of 20 averages, and the results (number of iterations needed for -35 dB convergence level, and respectively their optimum step-sizes) are shown in Figure 3.28, where for our purposes we select the values of the CFA and recursive filter constants the same as in previous experiments.

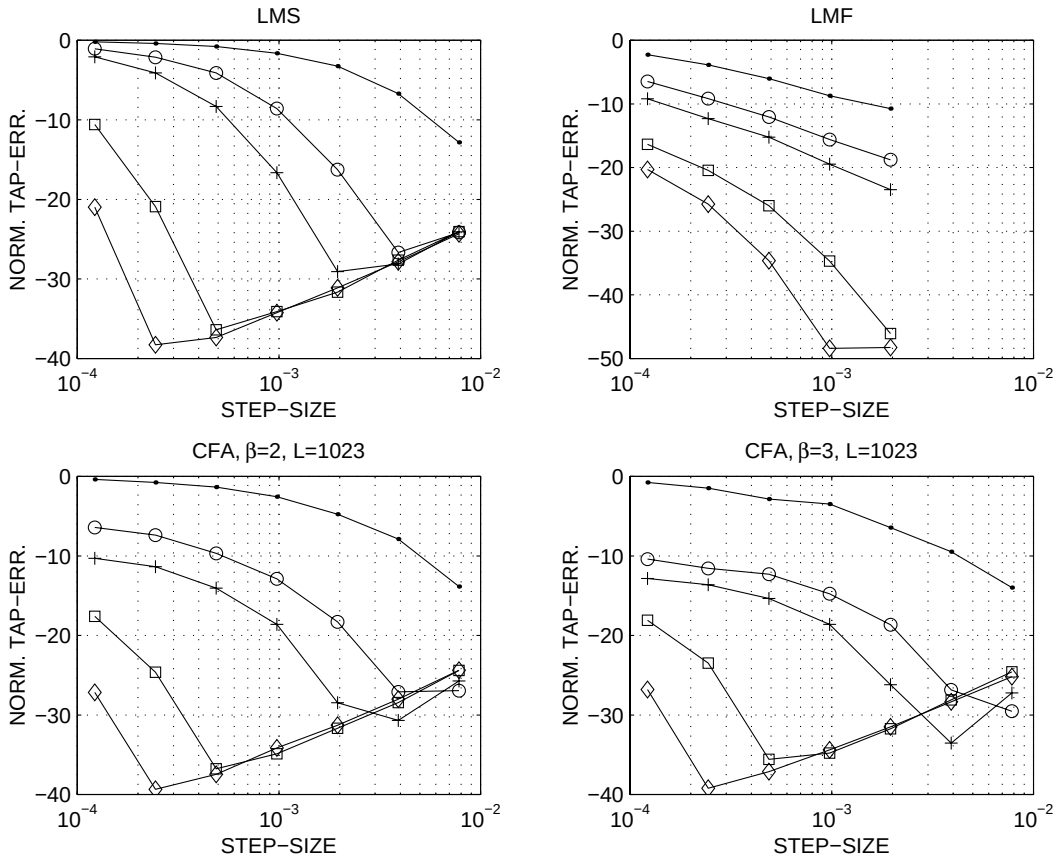


Figure 3.29: Performances of the LMS, LMF and CFA algorithms versus step-size for a given number of iterations: 100 (- ·), 500 (- ∘), 1000 (- +), 5000 (- □), and 10000 (- ⋄).

Then Figure 3.29 compares the proposed CFA and the quadratic algorithms from the point of view of the residual error for a given number of iterations, where the performances over 20 trials are presented. Note that the CFA algorithm is not sensitive to step-size increase unlike the LMF or other higher-order algorithms. It can also be seen that in some cases we can achieve a better performance with less iterations, and this can be explained in terms of the previous discussion on the influence of the step-size.

As a conclusion, the gradient approach of CFA algorithm offers an attractive implementation compared to the LMS and LMF, due to the superiority in almost all of the cases of the speed of convergence and residual error. The CFA drawback is increased computational complexity as we use sometimes non-integer gradients and additional operations for error exponent update.

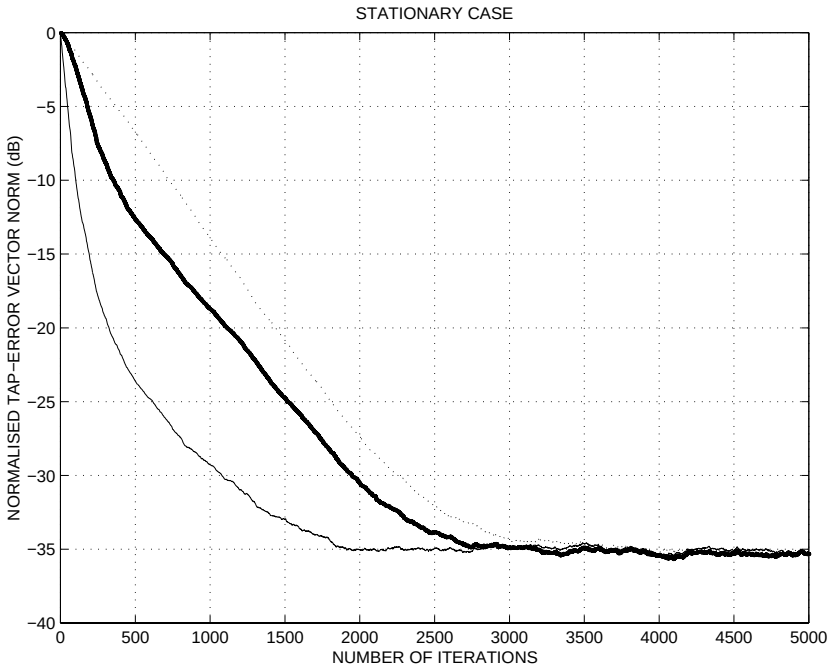


Figure 3.30: The variable step-size VS (——), GAS ($\cdots\cdots$), and CFA (▬▬) algorithms in the stationary case.

Example 13. *Cost function adaptation versus variable step-size*

We shall focus now on the comparison of the proposed CFA algorithm with the following standard variable step-sizes algorithms: VS [32] and GAS [46]. Parameters of these algorithms are selected to produce a comparable level of convergence, to assure stability, and if there is a choice, we follow the recommended values in their corresponding publication. For the VS algorithm we choose $\mu_{min} = 8.25 \cdot 10^{-4}$, $\mu_{max} = 8.25 \cdot 10^{-3}$, $m_0 = m_1 = 3$, and $\alpha_{VS} = 1.01$. For the GAS algorithm, we choose $\rho = 2 \cdot 10^{-8}$, and $\mu(0) = 8.25 \cdot 10^{-4}$ to obtain a comparable misadjustment value with the other algorithms. The CFA algorithm has the same step-size parameter $\mu = 8.25 \cdot 10^{-4}$, and we select $\beta = 1.5$, $L = 300$.

The framework used for comparison is the same as in Section 5.1, and we provide the results of two tests. The first one is from the point of view of the convergence speed where the learning curves are averaged over 20 trials, and the second one gives the algorithms' evolution for an abrupt sign change in the unknown channel transfer function. It can be seen from Figure 3.30 that the different starting gradients differentiate the VS, GAS and CFA algorithms in the stationary case[9]. Alternatively, Figure 3.31 shows that the

[9]Actually the initial gradients for VS algorithm are several times bigger than of the GAS and CFA.

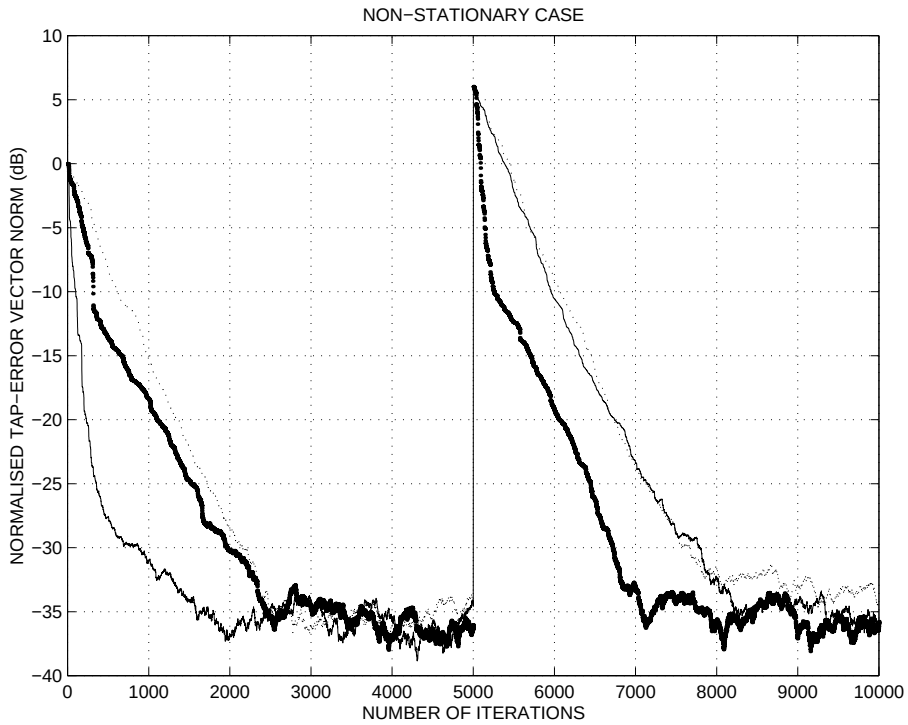


Figure 3.31: The non-stationary behaviour of the CFA and variable step-size algorithms: VS (——), GAS (· · · · · ·), and CFA (▬▬).

gradient approach of CFA behaves better in the non-stationary case than these variable step-size algorithms.

3.2.5 The linear CFA algorithm

The input process used to adapt is white (successive samples are uncorrelated) in the case of echo cancellation of data signals [36]. All eigenvalues of the autocorrelation matrix are therefore equal, and the eigenvectors are the columns of the identity matrix. In this case the output mean-square error exhibits exponential convergence to its asymptotic value, which is twice the minimum value. If the instantaneous error is processed as the input of a suitable low-pass filter, the output, in our case the estimate of the error, can be near of the form:

$$\hat{e}(k) = E_0 e^{-\frac{k}{\tau_0}}.$$

We did not bound the error exponent for the CFA algorithm, though such strategy is possible and can give bigger gradients as a result.

In this way we have a linear decreasing logaritmic error estimator:

$$\hat{|e|}_{dB}(k) = |E_0|_{dB} - \frac{8.68}{\tau_0}k.$$

Now we can write Equation (3.22) as:

$$r(k+1) = RE_{dB}{}^{1-\beta} \cdot \left(|E_0|_{dB} - \frac{8.68}{\tau_0}k\right)^{\beta-1},$$

where the parameters $r(0)$ and $|\hat{e}(0)|_{dB}$ were absorbed into the constant RE_{dB}. Assuming that $|E_0|_{dB}\tau_0 \gg 8.68$, this can be done easily since $\tau \gg 100$, and using the approximation [16]:

$$(1+b)^c \cong 1 + b \cdot c,\ b \ll 1,$$

then the following error exponent update results:

$$r(k+1) = RE_{dB}{}^{1-\beta}|E_0|_{dB} - \frac{8.68RE_{dB}{}^{1-\beta}(1-\beta)}{\tau_0}k.$$

This means that for a suitable choice of the parameters τ_0, $|E_0|_{dB}$ and β, the relation between $r(k)$ and k during the adaptation time is linear [59]. The LCFA algorithm results:

1) The weights are computed with Equation (2.1);

2) The power $r = r(k)$ is updated by:

$$r(k+1) = \max(r_{min}, r(0) - \overline{\Delta}r \cdot k),$$

where

$$r(0) = RE_{dB}{}^{1-\beta}|E_0|_{dB},$$
$$\overline{\Delta}r = \frac{8.68RE_{dB}{}^{1-\beta}(1-\beta)}{\tau_0}.$$

We tested the LCFA algorithm and we present the results for two different echo path channels, both single pole single zero digital filter. The first one is a $N = 72$ tap linear time-varying FIR adaptive filter whose coefficients are updated regularly by the adaptation algorithm. The channel considered has one pole at 0.8232 ($A = -120$). The attenuation of the far-end signal sequence corresponds to -20 dB. The parameter of the recursive digital filter was $L = 2500$. The step size μ was chosen as $5 \cdot 10^{-4}$ for all the LMS, LMF, LCFA algorithms, in order to assure convergence of the LMF algorithm. The convergence performance of all three algorithms is illustrated in Figure 3.32, where we have four plots. The top-left-side plot shows the behaviour of the LMS versus the LCFA algorithm, whereas the bottom-left-side plots makes the same comparison for the LMF

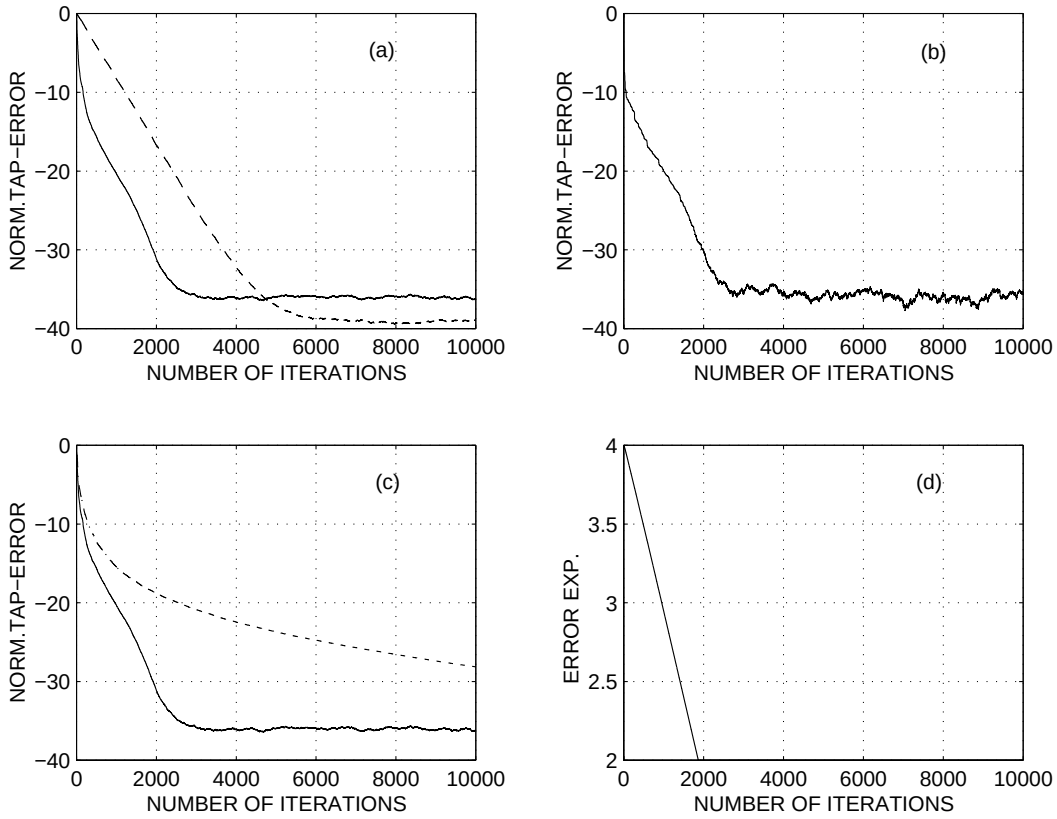


Figure 3.32: Learning curves (a) of the LCFA(-) and LMS (- -), a sample of this type of LCFA learning curve (b), learning curves (c) of the LCFA(-) and LMF (-.), and error exponent update (d) for the sample shown at (b).

algorithm. All these plots were obtained with 20 runs. The right-hand side of the figures presents the plots of a sample of the LCFA algorithm: on the top the normalised tap-error vector norm evolution and beneath, the evolution of the power of the cost function during adaptation. Figure 3.33 presents how the error estimator works, by showing all the signals involved, from the instantaneous error to the error exponent update.

The second set of results considered a echo path channel with the pole at 0.8. The number of filter coefficients was $N = 40$. The level of the attenuated far-end signal was -20 dB. Figure 3.34 shows the 3D representation of the LCFA, LMS and LMF algorithms. Clearly, if we choose properly the time constant of the low-pass filter, then the results are better than LMS and LMF. For instance, if $L = 2000$, the LCFA ($\beta = 2, \mu = 0.0005$) is faster than LMS1 ($\mu = 0.0005$), even the LMS2 ($\mu = 0.001$) where the step-size is the double of LCFA. LCFA is also faster than LMF ($\mu = 0.0005$). The learning curves

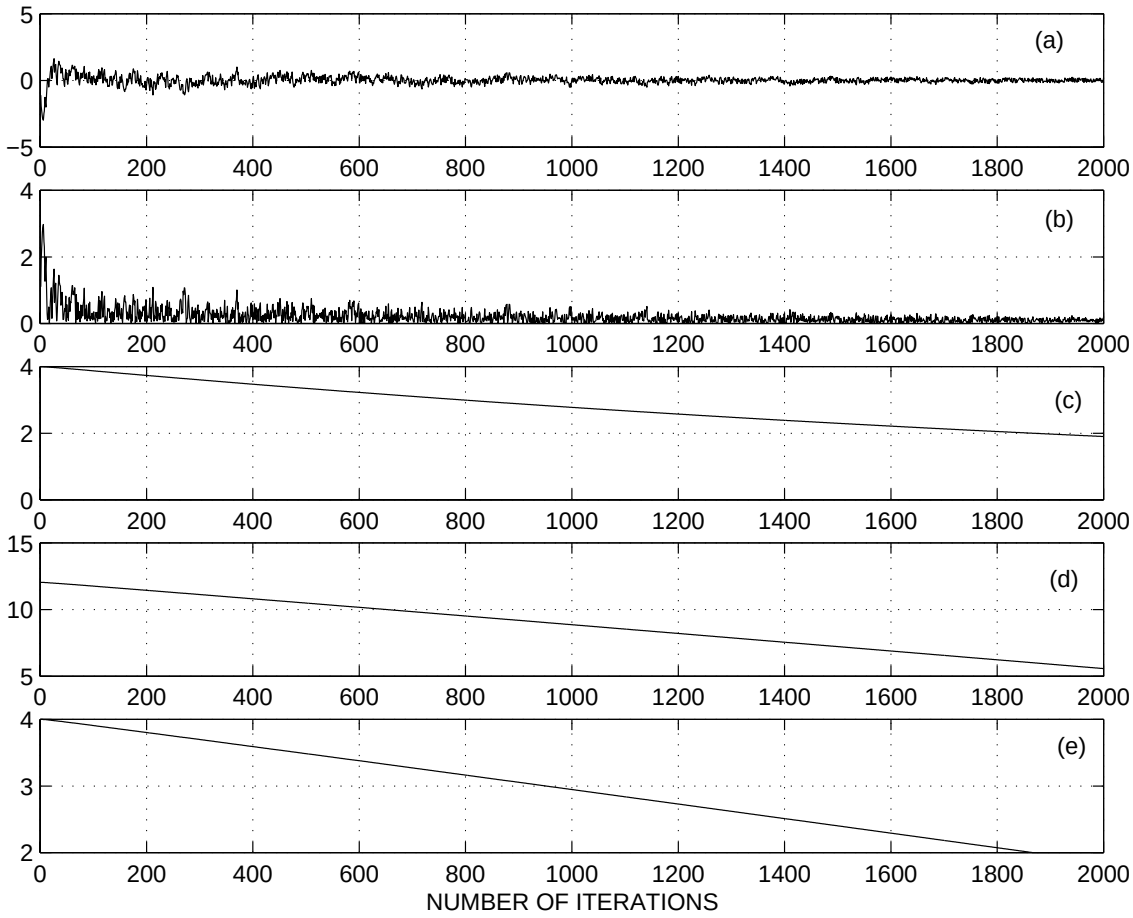


Figure 3.33: The LCFA algorithm: the error $e(k)$ (a), its absolute value $|e(k)|$ (b), error estimate $|\hat{e}(k)|$ (c), its dB's value $|\hat{e}(k)|_{dB}$ (d), and linear power update $r(k)$ (e).

(Figure 3.35) obtained are the average of 50 runs.

3.3 Appendixes

First case derivation of smooth CFA algorithm

Consider the function $g : (0, +\infty) \to (0, +\infty)$, $g(x) = xd^{x-2}$.

For $d > 1$, g is strictly increasing, for it is the product of two positive strictly increasing functions: $x \to x$ and $x \to d^{x-2}$.

For $d = 1$, $g(x) = x$ is also strictly increasing.

Thus function g is injection and from $g(x) = g(y)$, follows that $x = y$.

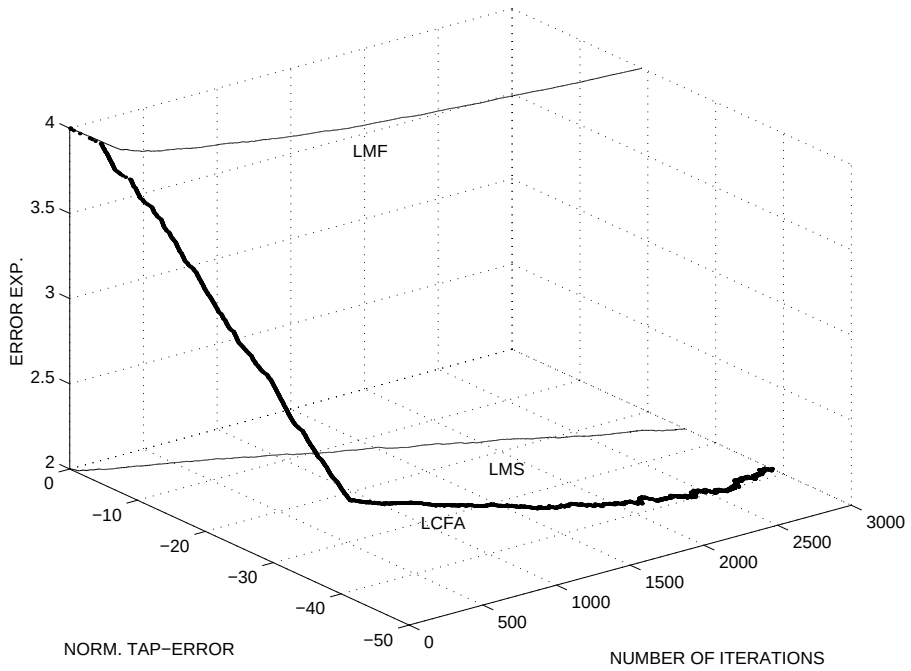


Figure 3.34: The 3D representation of the LCFA, LMS and LMF algorithms.

Proof of Equation (3.13)

Let $\{f(k)\}$ and $\{x(k)\}$ be two independent bipolar sequences, from the sets $\{a, -a\}$ and $\{1, -1\}$, respectively ($f^2(k) = a^2, x^2(k) = 1$), then the error satisfies

$$e^2(k) = (f(k) + \sum_{n=0}^{N-1} (\hat{h}(n) - h(n)) \cdot x(k-n))^2$$
$$= f^2(k) + \sum_{i=0,j=0}^{N-1} \Delta h(i)\Delta h(j)x(k-i)x(k-j) + 2f(k)\sum_{n=0}^{N-1} \Delta h(n)x(k-n).$$

If $i \neq j$, then $E[x(k-i), x(k-j)] = E[f(k), x(k-n)] = 0$, hence

$$E[e^2(k)] = a^2 + E[\sum_{n=0}^{N-1} (\hat{h}(n) - h(n))^2],$$

therefore an estimate is

$$|\hat{e}(k)|_{dB} = 10\log(E[e^2(k)]) = 10\log(a^2 + E[(\hat{\boldsymbol{h}}(k) - \boldsymbol{h}(k))^2]),$$

and we choose

$$\bar{\mathrm{e}}(k) = 10\log(a^2 + \|\hat{\boldsymbol{h}}(k)\|^2 p^2(k)).$$

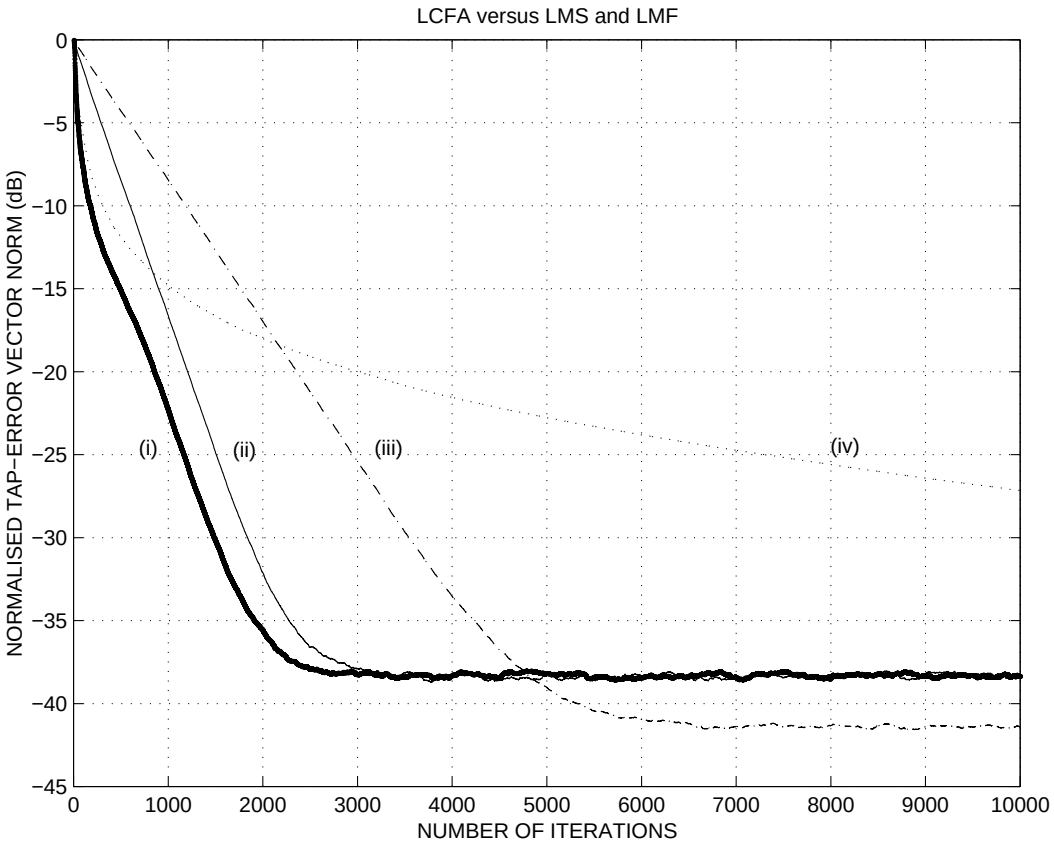


Figure 3.35: Learning curves for LCFA (i), LMS2 (ii), LMS1 (iii) and LMF (iv).

Proof of Equation (3.15)

Consider the function $f : \mathbb{C} \to \mathbb{R}_+$, defined by $f(z) = |z|^{F(z)}$, where $F : \mathbb{C} \to \mathbb{R}$, and suppose the function F has derivative in the given domain of definition. Then the function $g : \mathbb{C} \setminus \{0\} \to \mathbb{R}$, $g(z) = \ln f(z)$ also has derivative in this domain, and

$$g'(z) = [F(z) \ln |z|]' = F'(z) \ln |z| + F(z)[\ln |z|]' = F'(z) \ln |z| + F(z) \cdot \frac{z}{|z|^2};$$

Hence the function f also has derivative in the indicated domain, and

$$f'(z) = f(z)g'(z) = |z|^{F(z)-2}[zF(z) + |z|^2 \ln |z| F'(z)].$$

For $z = 0$ we can follow Appendix 2.4 and it results, that $f'(z) = 0$, iff $F(z) > 1$, for all $z \in \mathbb{C}$.

Solution of Equation (3.21)

Let $z = e(k)$ be, and from Equation (3.21) we obtain

$$|z|^2 \ln |z| \frac{dr}{dz} = (\beta - 1) z r.$$

For $z \neq 0$, and $|z| \neq 1$, we have two cases:

(i) $\beta = 1$, then $r =$ constant;

(ii) $\beta \neq 1$. In this case

$$\frac{r'}{r} = \frac{(\beta - 1) z}{|z|^2 \ln |z|},$$

thus

$$r = [\theta \ln |z|]^{(\beta-1)\mathrm{sgn}(\ln |z|)}.$$

For $z = 0$, and $|z| = 1$, there should be a trade off between continuity of r, and its finite, non-zero values[10].

[10]Perhaps it is interesting to note that we can develop another approach for which these discontinuities disappear, by selecting in the right-hand side of Equation (3.21) of other expressions, for instance $(1 + r(k) \ln |e(k)|)|e(k)|$, suggested by the gradient from Section 4.2. The resulting approach we got has a mathematical beauty rather than a physical interpretation. Moreover, the resulting gradient expression is more complicated than in the case of stationary and gradient approaches. It seems that such an approach is not attractive in data echo cancellation. However it might be useful in certain applications.

Chapter 4

Recursive Cost Function Adaptation

The goal of this Chapter is to introduce an approach for recursive cost function adaptation. The derivation of the new algorithm does not use an estimator of the instantaneous error as the previous CFA algorithms did. In the RCFA case, the new error exponent is computed from the previous one using usual LMS recursive equation. The proposed method improves the sensitivity of the power of the cost function with respect to the noisy error, while the other benefits of the CFA algorithms in terms of the convergence speed and residual error remain. In Section 4.1 the heuristic approach is developed. The properties of the new algorithm will be compared, using computer simulations, to standard LMS, LMF. The effect of the parameters involved in the design are also discussed following a short analysis of the error exponent adaptive subsystem. Finally the idea of an analytic approach, similar with the variable gradient step-size technique [46] is considered.

4.1 The Heuristic Approach

The simplified block diagram of the main adaptive system EPIS (Echo-Path Identification System) is shown in Figure 2.2, where the adaptive FIR filter is trying to make a copy $\hat{y}(k)$, of the echo-path output $y(k)$, using the signal $x(k)$ as an input, based on a measurement of the signal that remains after subtracting $\hat{y}(k)$ from the received signal $y(k)+f(k)$. For the beginning we consider that in this framework the LMS algorithm is acting:

$$\hat{\boldsymbol{h}}(k+1) = \hat{\boldsymbol{h}}(k) + 2\mu e(k)\boldsymbol{x}(k),$$

where μ is the step-size of the echo-path identification system. From the Equation (2.7) we have

$$\hat{\boldsymbol{h}}(k+1) = \hat{\boldsymbol{h}}(k) + 2\mu[f(k) - \Delta\boldsymbol{h}^t(k)\boldsymbol{x}(k)]\boldsymbol{x}(k). \tag{4.1}$$

If the channel is slowly varying, which is a very common assumption in adaptive filtering, then we can subtract the echo-path filter coefficients vector from both sides of the Equation (4.1). It follows

$$\Delta \boldsymbol{h}(k+1) \approx \Delta \boldsymbol{h}(k) + 2\mu[f(k) - \Delta \boldsymbol{h}^t(k)\boldsymbol{x}(k)]\boldsymbol{x}(k). \tag{4.2}$$

Now we return to CFA framework, and in the following the Equation (2.1) will be used in order to update the EPIS coefficients. Our interest is to determine the formula between the instantaneous error $e(k)$ of the adaptive filter and the new error exponent $r(k+1)$ which is to be used to update the adaptive filter coefficients with the Equation (2.4). Unlike Chapter 3 where a direct relationship was obtained, now the goal is to find something similar to (4.2). Our approach is based on the idea that there should be a desired error exponent function r_k at every instant k, which gives the best performance for the actual error of the system. As we do not know r_k, at least we can try to estimate it, based on the available measurements and certain information.

4.1.1 Concept of the heuristic approach

Suppose that we do not know the "unknown desired" error exponent r_{k+1}. On this assumption we can use the LMS algorithm and compute an "estimate of the new error exponent" $r(k+1)$. For this reason we need the "near-end signal", and this will be the instantaneous error $e(k)$, because $r(k)$ is expected to be a function of $e(k)$ (Section 2.2.4). We need also the "attenuated far-end signal". It must have similar statistical properties as the "near-end signal". From the available signals we select the input observation sample $x(k)$, which is subject to an attenuation $\varphi(k)$. The attenuation might be constant or not, whether we use some appropriate averages of the attenuated far-end signal $f(k)$, or simply $f(k)$. A graphical representation of these relationships is shown in Figure 4.1.

Remark 7. *The attenuation $\varphi(k)$ is necessary.*

The difference between $r(k)$ and r_k should be very small ($|r(k) - r_k| \ll 1$), at least after adaptation, but the data signal $x(k)$ is not so small ($|x(k)| \simeq 1$).

Thus we have the useful equations:

$$\begin{aligned} \Delta r(k) &= r(k) - r_k, \\ \epsilon(k) = r(k) + \varphi(k)x(k) - r_k &= \varphi(k)x(k) - \Delta r(k)e(k), \\ r(k+1) &= r(k) + 2\rho e(k)\epsilon(k), \end{aligned} \tag{4.3}$$

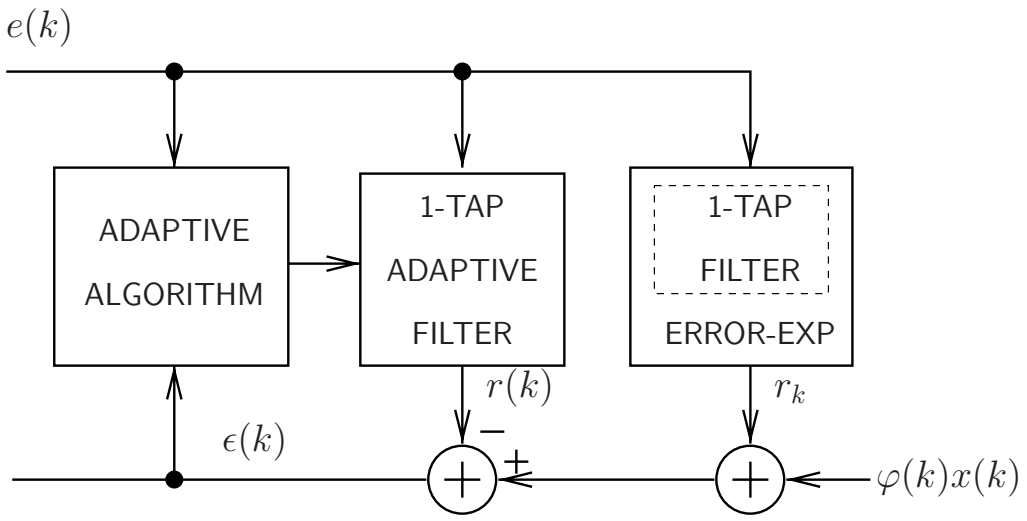


Figure 4.1: Error exponent identification setup.

where $\epsilon(k)$ is the error signal and ρ is the step-size of the EEAS (Error Exponent Adaptation Subsystem) subsystem.

Now the equations (4.3) can give us the estimated error exponent, and in this way $r(k)$ can be used to update the weights with the relationship (2.1) with $r = r(k)$. It follows that the heuristic approach of the RCFA algorithm is defined by the following:

1) The weights are computed with Equation (2.1);

2) The power $r = r(k)$ is updated by Equations (4.3).

The proposed algorithm introduces a new parameter ρ and sequence r_k.

The EEAS step-size contributes in a different manner to the stability of the subsystem as we are used in other adaptive filtering tasks. Actually the increase of ρ does not necessarily provide divergence, as it could give a faster decrease of the estimated error exponent. Indeed, we emphasize that

Property 1 . *One of the cost function adaptation merits is that we deal with higher-order exponents only for short periods.*

The higher-error exponents could easily lead to instabilities, and this happens always after a certain number of iterations. If we can deal with these type of algorithms only for a while, we can avoid this type of problem, and take advantage of the higher gradients. However, in some cases it is clear that the step-size of EEAS subsystem must be bounded (Section 4.1.2).

Clearly, the choice of "unknown desired" error exponent sequence is more disputable.

First we have to mention the necessity of r_k, otherwise the EEAS system could diverge. If no targeted error exponent is present, the EEAS system can easily reach different high or low error exponents according to the input data. This differentiate RCFA from GAS variable step-size technique. There the modification of the step-size by a factor of two is not so dramatic as for the error exponent in the cost function adaptation case.

A second issue is that this sequence is what we would like to know. Actually it is for EEAS subsystem the same as Wiener filter for mean-square optimisation problem. As we usually do not have knowledge of this information, we suggest to try the following:

1. A constant r_k.
2. A rough approximation of r_k.
3. A nonrecursive cost function adaptation approach (Chapter 3).

In the first case, the choice of the constant error exponent can be justified by some additional information about the type of involved signals [30]. If this is not available, we suggest to select $r_k = 2$, and the LMS will act in steady-state. Another modality is to implement rough approximation of r_k, as it will be used later in Example 17, when we present the non-stationary behaviour of RCFA algorithm. Perhaps a better solution is an assessment of r_k given by a nonrecursive CFA, but this alternative increases the computational complexity of the overall algorithm.

4.1.2 A short analysis of the EEAS subsystem

Consider once again that the "unknown desired" error exponent is slowly varying $r_{k+1} \simeq r_k$, and subtract them from both sides of the last of the (4.3) relationships. We have

$$r(k+1) - r_{k+1} = r(k) - r_k + 2\rho e(k)\epsilon(k).$$

From the first two of the (4.3) relationships we obtain

$$\begin{aligned}\Delta r(k+1) = \Delta r(k) + 2\rho e(k)\epsilon(k) &= \Delta r(k) + 2\rho e(k)[\varphi(k)x(k) - \Delta r(k)e(k)] \\ &= 2\rho\varphi(k)x(k)e(k) + \Delta r(k)[1 - 2\rho e^2(k)].\end{aligned}$$

Applying previous recursion P times with respect to k, we conclude that

$$\begin{aligned}\Delta r(P+1) = \alpha(P) + \beta(P)\alpha(P-1) + \beta(P)\beta(P-1)\alpha(P-2) + \cdots \\ +\beta(P)\beta(P-1)\cdots\beta(1)\alpha(0) + \beta(P)\beta(P-1)\cdots\beta(1)\beta(0)\Delta r(0),\end{aligned}$$

where

$$\alpha(k) = 2\rho e(k)\varphi(k)x(k), \quad \beta(k) = 1 - 2\rho e^2(k).$$

For the error exponent adaptation subsystem to converge it is necessary that the product

$$\prod_{k=0}^{\infty}[1 - 2\rho e^2(k)]$$

also converges. This is equivalent [45] with the convergence of the series

$$\sum_{k=0}^{\infty} e^2(k),$$

if $\lim_{k\to\infty} e(k) = 0$. If this last condition does not happen, then the following had to be true:

$$\limsup_{k\to\infty} |e(k)| < \frac{1}{\sqrt{\rho}}. \tag{4.4}$$

In this case the estimated error exponent is convergent and thus bounded[1], and for the stability of the overall adaptive system we can follow the general treatment developed in Chapter 5.

As a rule, we have noticed in our simulations that the RCFA algorithm has a better stability than the LMF and other CFA algorithms. We believe that the initial fast decrease of the estimated error exponent is one of the contributing factors (Section 4.1.3). We did not find remarkable differences between recursive and nonrecursive approaches from the convergence speed point of view.

4.1.3 Experimental results

Now we proceed with several examples. In order to test the proposed algorithm, we first consider a channel of the first type, with one zero at the origin and one pole at 0.8. The number of filter coefficients is $N = 40$. The level a of the attenuated far-end signal is given by $a_{dB} = -20$. The step-size of the main adaptive filter is chosen $\mu = 5 \cdot 10^{-4}$. The performance measure is the normalised form of the tap-error vector norm given by (2.8), and the learning curves obtained are the average of 20 runs. For the beginning the unknown error exponent r_k is a constant function, and the initial estimated error exponent is $r(0) = 4$. The attenuation $\varphi(k)$ is constant and equal with the level of attenuated far-end signal ($\varphi(k) = a$).

[1]The error exponent can be roughly bounded, in the way that the variable step-size techniques are doing for the step-size, but this was not our option when we developed the CFA approaches.

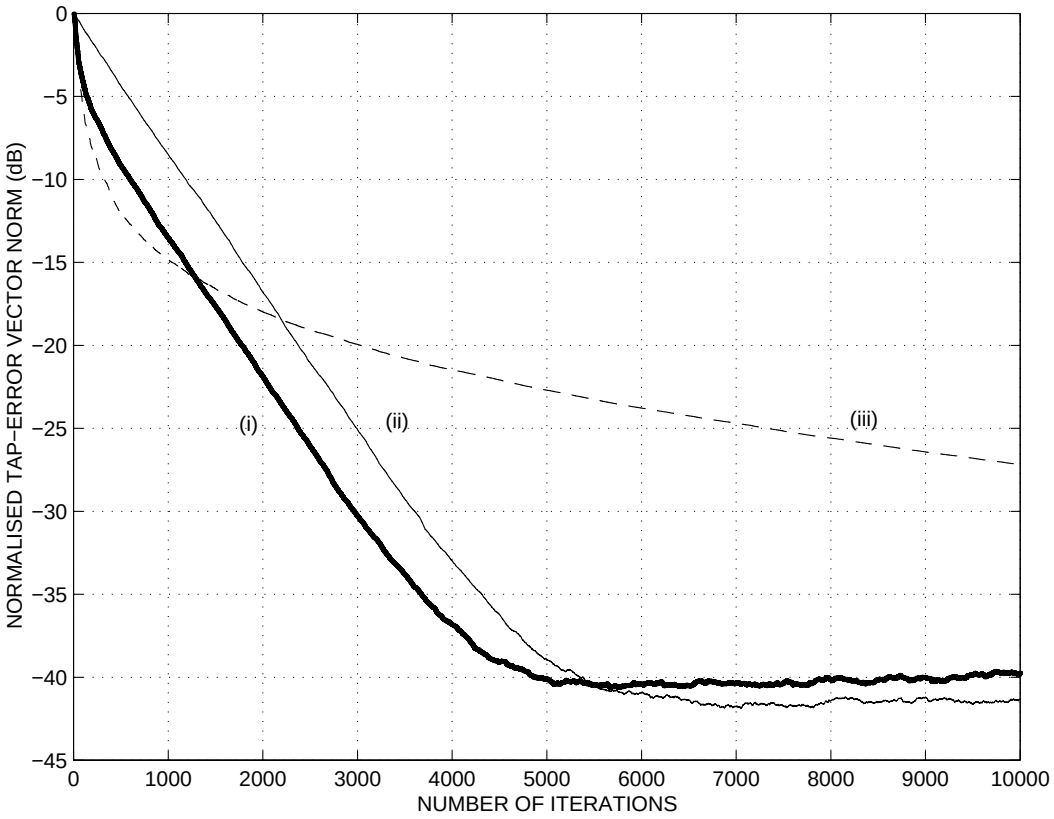


Figure 4.2: The learning curves of the RCFA (i), LMS (ii) and LMF (iii) algorithms.

Example 14. *A short comparison of RCFA with LMS and LMF*

Figure 4.2 shows a comparison between LMS, LMF and RCFA ($\rho = 0.001, r_k = 1$) algorithms, from the convergence speed and steady-state point of view, where all the algorithms have the same step-size [52]. If $r(\infty) = 2$, then it can be noticed that RCFA has a faster convergence than both algorithms, and the steady-state properties are the same as for the LMS algorithm.

In this experiment the error exponent reaches 2, then it slowly decreases to 1. We can stop this evolution by low bounding the error exponent, and in this way we obtain a better residual error. Instead of using this strategy, we prefer to continue the cost function adaptation and do not intervene in the error exponent adaptation process.

Perhaps it is better to remind here the special CFA effect of changing the levels in steady-state, encountered also in Example 11:

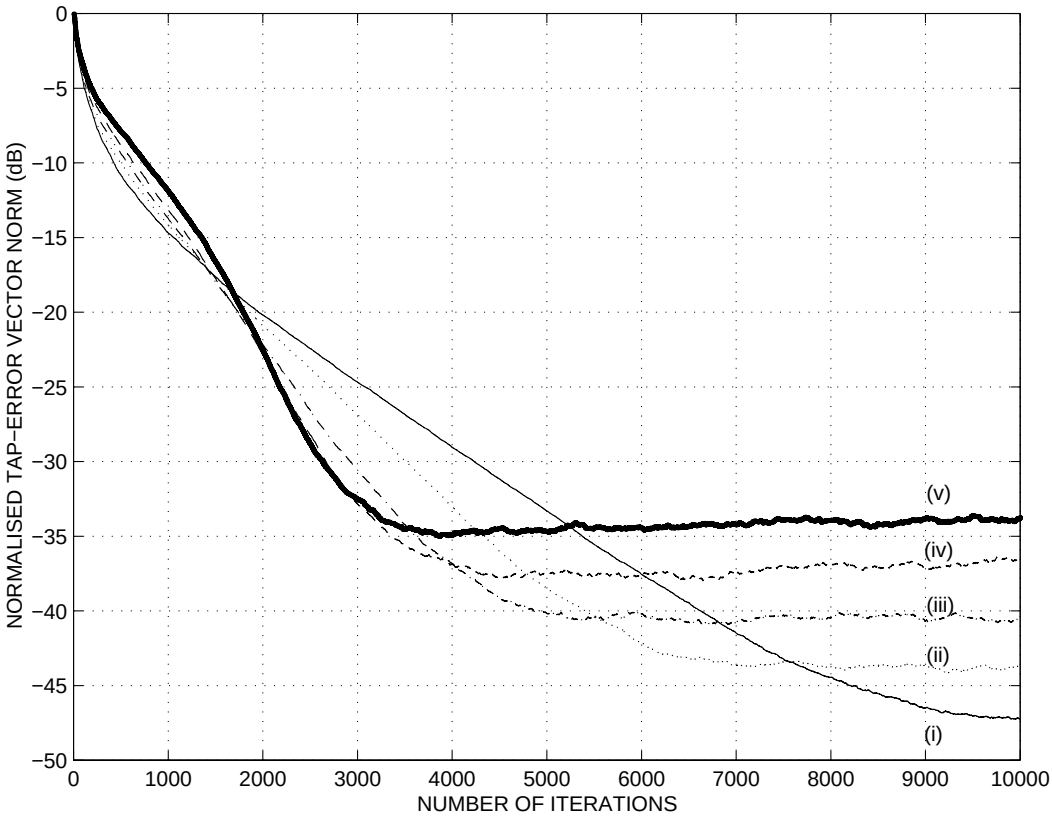


Figure 4.3: RCFA learning curves and EEAS step-sizes: $\rho = 6 \cdot 10^{-4}$ (i), $\rho = 8 \cdot 10^{-4}$ (ii), $\rho = 10 \cdot 10^{-4}$ (iii), $\rho = 12 \cdot 10^{-4}$ (iv), $\rho = 14 \cdot 10^{-4}$ (v).

Property 2. *The CFA algorithm reaches first a lower (apparent) steady-state level corresponding to a certain error exponent, but because the error exponent sequence is still slightly decreasing, the (apparent) steady-state level increases.*

Example 15. *EEAS parameters influence in RCFA performances*

Figure 4.3 (for learning curves) and Figure 4.4 (for estimated error exponent $r(k)$) illustrate the performances of the RCFA algorithms, if the EEAS step-size changes. For a small step-size ($\rho = 0.0006$), the estimated error exponent decreases slowly, and as a consequence the respective RCFA algorithm behaves closer to LMF at the beginning, and after that with NQSGr ($r \approx 2.5$). For $\rho = 0.0014$, the corresponding RCFA has a faster convergence, but the steady-state is worse than for the LMS algorithm as we have $r(\infty) = 1.4$.

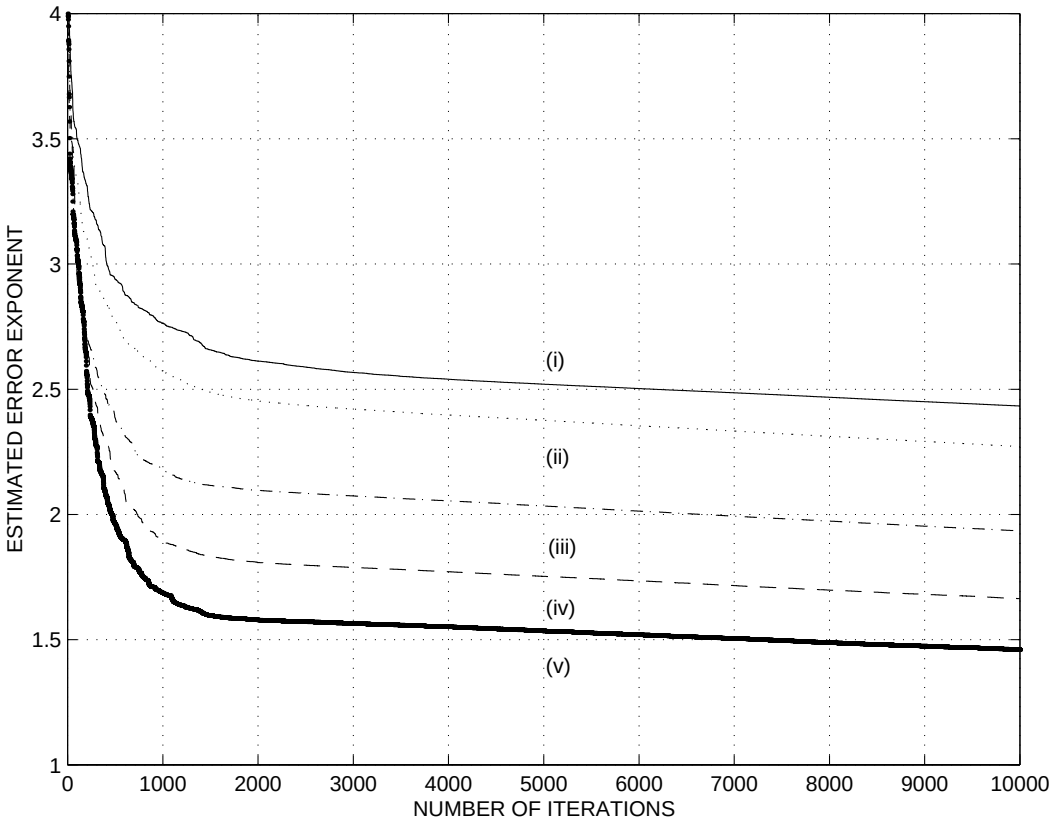


Figure 4.4: Estimated error exponent and EEAS step-sizes: $\rho = 6 \cdot 10^{-4}$ (i), $\rho = 8 \cdot 10^{-4}$ (ii), $\rho = 10 \cdot 10^{-4}$ (iii), $\rho = 12 \cdot 10^{-4}$ (iv), $\rho = 14 \cdot 10^{-4}$ (v).

The same type of comparison was done from the unknown error exponent r_k point of view. The choice of the constant function r_k affects both the convergence rate and the steady-state (Figure 4.5). Also the estimated error exponent $r(k)$ is changed (Figure 4.6).

By modifying the initial error exponent $r(0)$, the performances of the system are quite affected as we can see in Figure 4.7. The time constant of the EEAS system is almost the same, but its steady-state is reached at different instants of time. As a consequence the behaviour of the algorithm differs greatly.

Example 16. *EPIS parameters and RCFA properties*

In the first experiment we modify the noise levels, keeping the rest of parameters fixed ($r_k = 1.9$, $\mu = 75 \cdot 10^{-5}$, $\rho = 0.0025$, $N = 40$, $p = 0.8$, $r(0) = 4$). Figure 4.8 shows that the

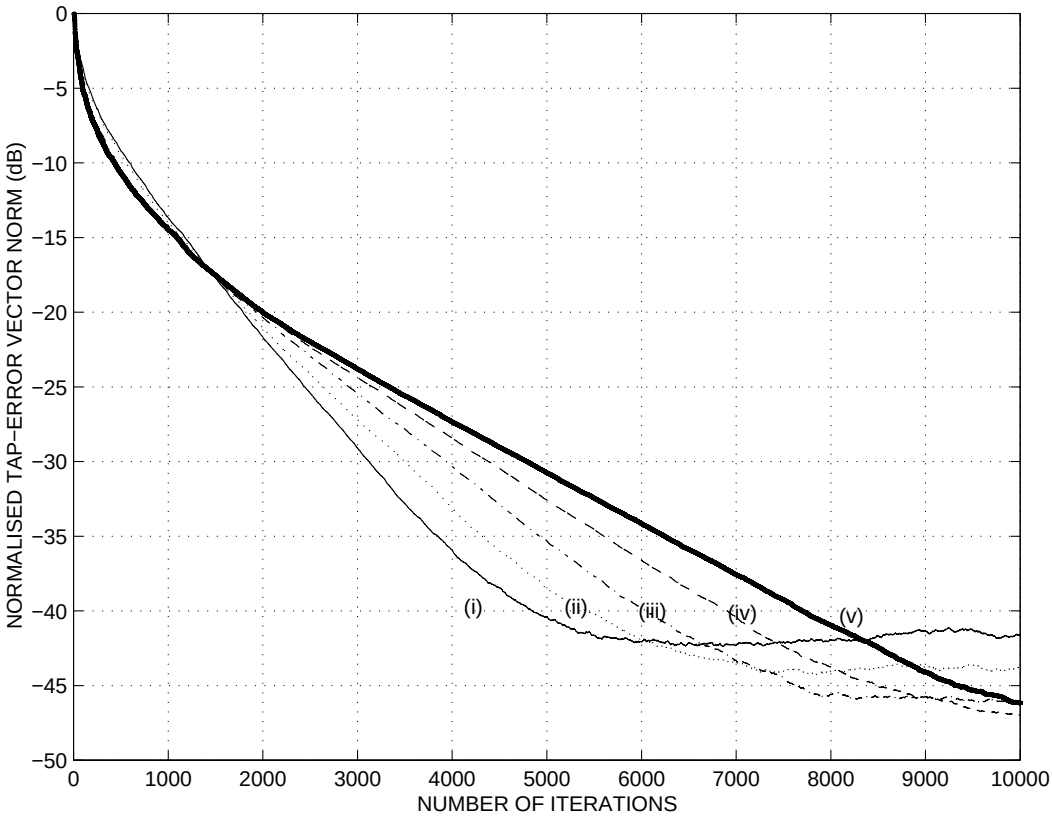


Figure 4.5: Learning curves of RCFA for different desired error exponents: $r_k = 1.2$ (i), $r_k = 1.4$ (ii), $r_k = 1.6$ (iii), $r_k = 1.8$ (iv), $r_k = 2$ (v).

steady-state performances are completely different for the four levels selected, and this was what we actually expected. However for a long part of adaptation time the adaptive filter performs quite the same. Figure 4.9 presents the behaviour of the estimated error exponent when the level of attenuated far-end signal is changed. When the instantaneous error is noisy, the EEAS subsystem converges slower.

Property 2 is illustrated in Figure 4.10, where the effect of step-size of the EPIS system is illustrated. We note that the estimated error exponent perceives the modification of the step-size through the instantaneous error signal.

These experiments clearly show that a trade-off should be done between the parameters involved in the design of this complex adaptive filter, i.e. ρ, r_k, and respectively μ. The plots presenting the estimated error exponent suggest also that it is not very sensitive to noisy error during adaptation and steady-state, in the sense that the decrease of the estimated error exponent is smooth and almost monotonic.

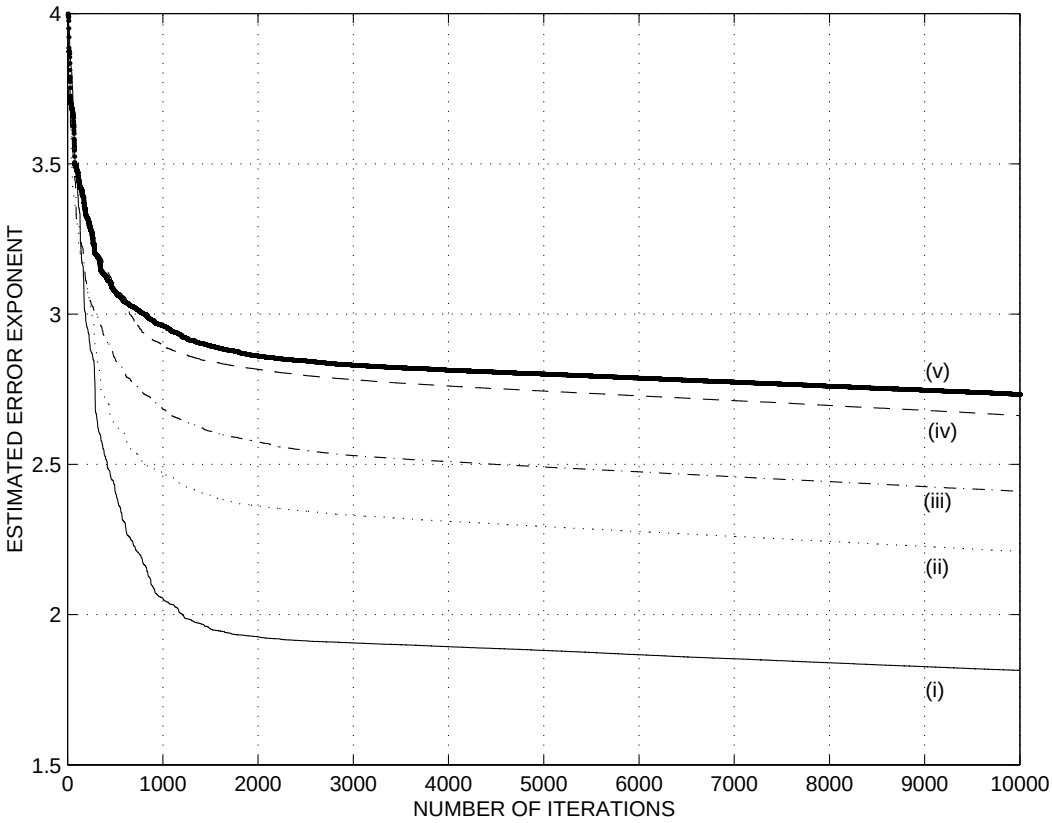


Figure 4.6: Estimated error exponent for different desired error exponents: $r_k = 1.2$ (i), $r_k = 1.4$ (ii), $r_k = 1.6$ (iii), $r_k = 1.8$ (iv), $r_k = 2$ (v).

Example 17. *Abrupt change in the system*

The last experiment of this Section is to track the second channel which is the real hybrid, where at 2501 iteration the echo-path has a sign change. The level of noise was -20 dB. The step-size of EEAS subsystem is $\rho = 0.5$, and for the "unknown desired" error exponent sequence r_k we implement a rough approximation. It consists in selecting $r_k = 2$ close to the end of adaptation period, and $r_k = 2.5$ far from optimum. The decision whether we are in one of these two cases is taken by counting the number of recent iterations where the instantaneous error is less than a given value. In our simulations we considered that we reach the steady state if for 8 consecutive iterations the modulus of the error is less than the maximum peak-to-peak of noise signal.

The outcome is shown in Figure 4.11, and it shows that RCFA behaves well in data echo-cancellation of a real hybrid and also in non-stationary situations.

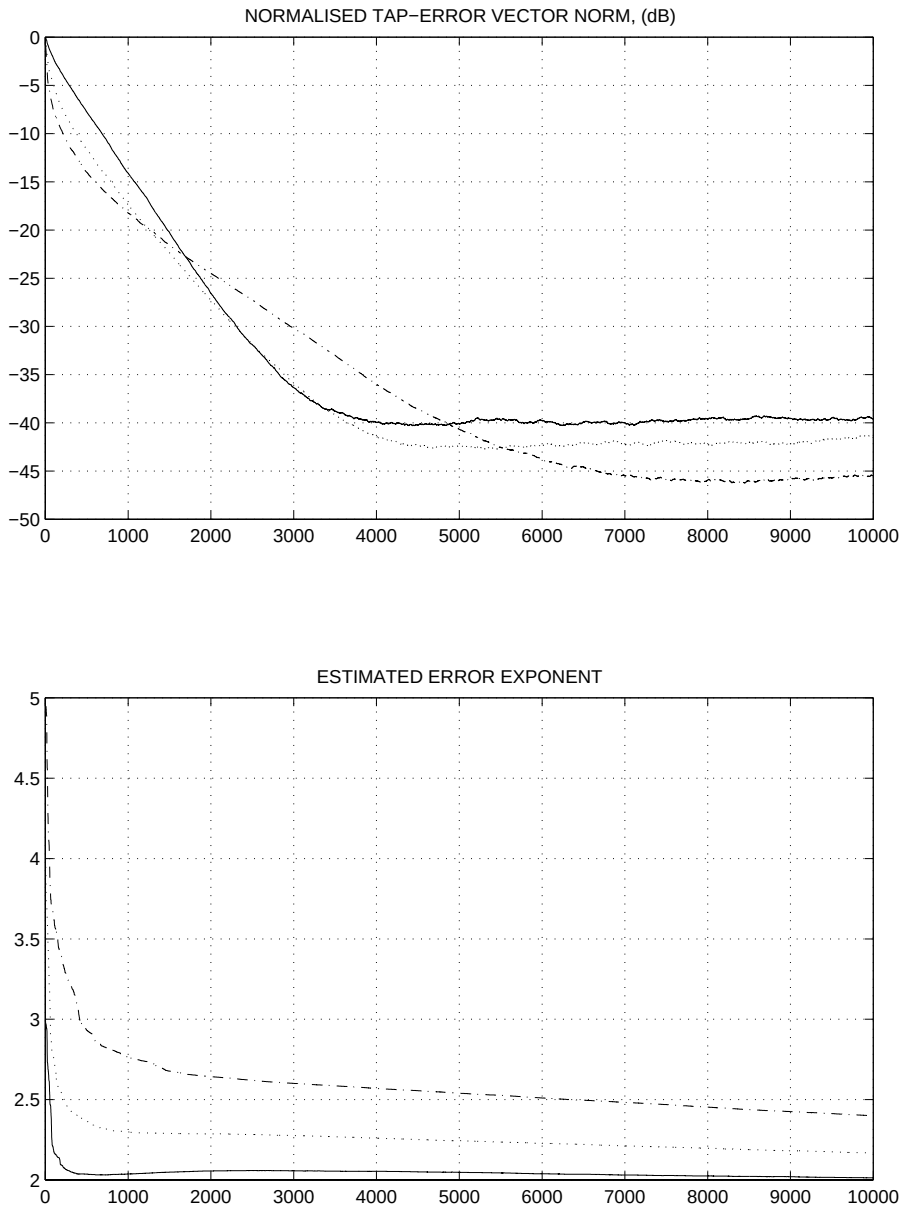


Figure 4.7: Performances of the RCFA algorithm for different initial estimated error exponents: $r(0) = 3$ (——), $r(0) = 4$ ($\cdot\cdot\cdot\cdot\cdot$), $r(0) = 5$ ($\cdot - \cdot - \cdot$).

4.2 A Note on The Analytic Approach

A stochastic gradient adaptive filter with gradient adaptive step-size was proposed in [46, 74]. Despite other variable step-size methods, rather heuristic techniques, the GAS

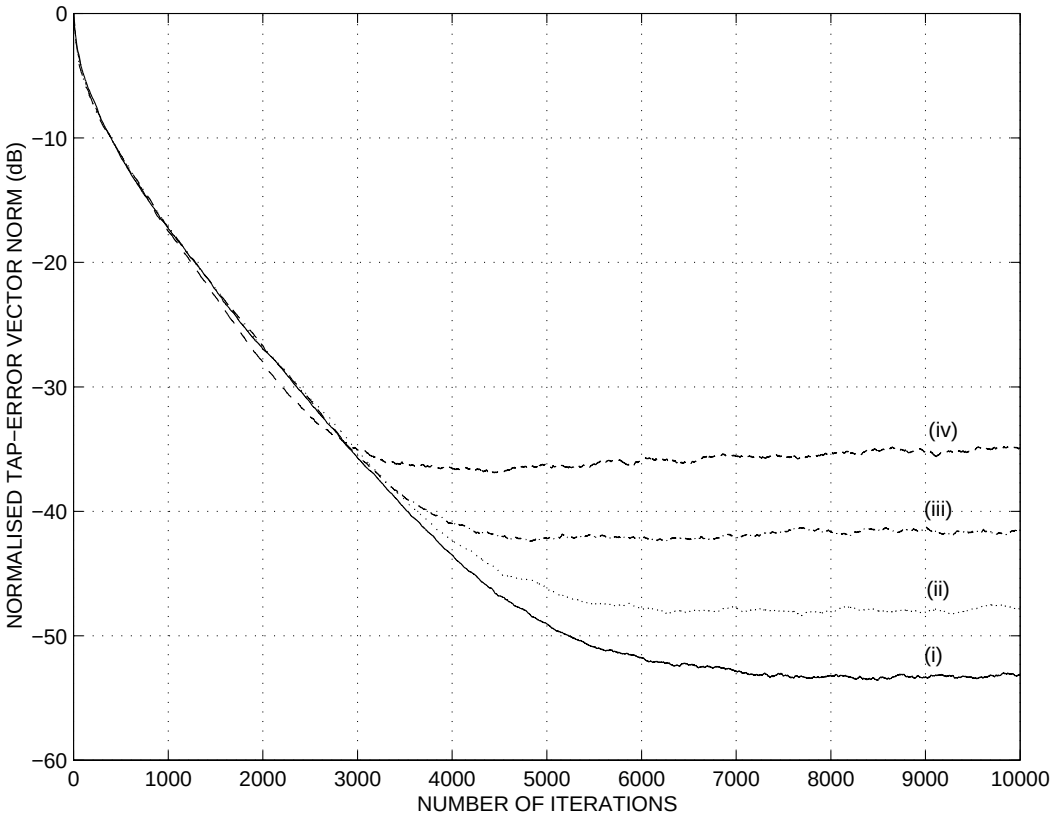


Figure 4.8: Learning curves for different noise levels: $a_{dB} = -30$ (i), $a_{dB} = -25$ (ii), $a_{dB} = -20$ (iii), $a_{dB} = -15$ (iv).

algorithm changes the time-varying step-size in such a way that the change is proportional to the negative of the gradient of the squared estimation error with respect to the convergence parameter.

In the following we would like to see if we can follow this idea and to derive another recursive cost function adaptation technique based on the same optimality principle. The proposed method should try to adapt the error exponent sequence using a gradient descendent algorithm so as to reduce the squared estimation error at each instant of time.

We start by computing the recursion formula as in [46]:

$$r(k+1) = r(k) - \varrho \frac{\partial e^2(k+1)}{\partial r(k)} = r(k) - \varrho \frac{\partial e^2(k+1)}{\partial \hat{\boldsymbol{h}}(k)} \cdot \frac{\partial \hat{\boldsymbol{h}}(k)}{\partial r(k)}, \tag{4.5}$$

where ϱ is a small positive constant that controls the adaptive behaviour of the error exponent sequence.

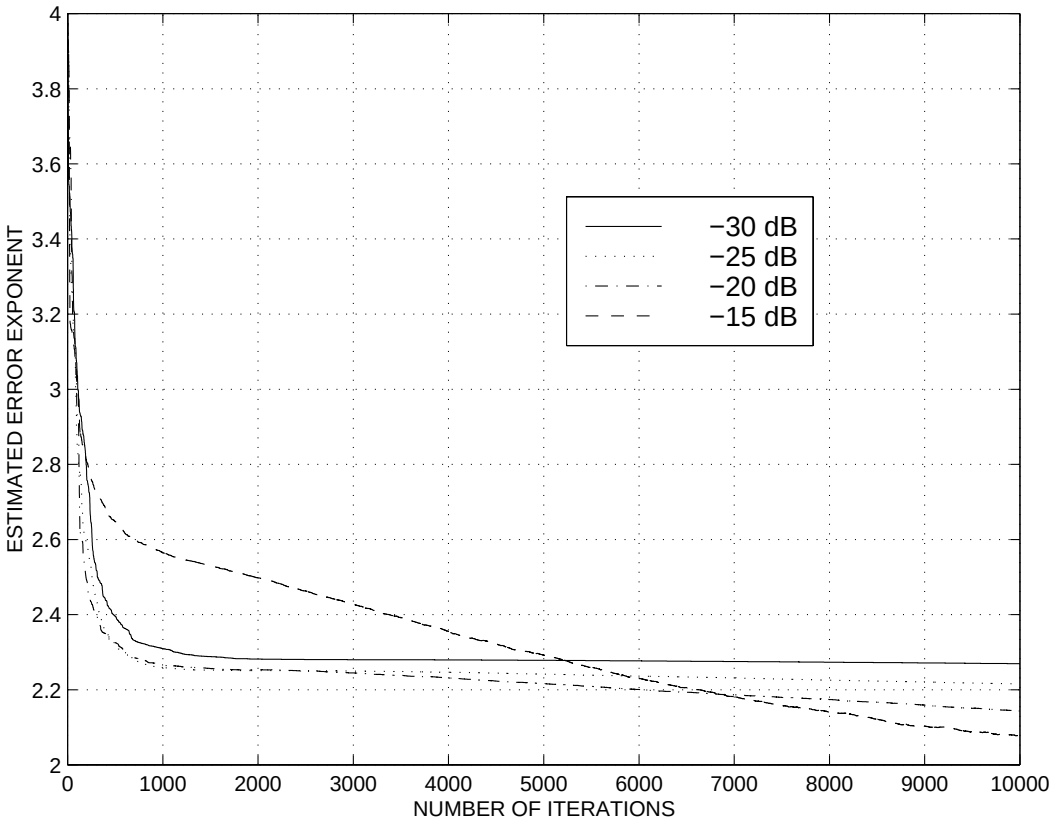


Figure 4.9: Estimated error exponent for different noise levels.

By using (3.16), and taking into account that from (2.1) we have

$$\frac{\partial \hat{\boldsymbol{h}}(k)}{\partial r(k)} = \mu[1 + r(k) \ln |e(k)|]|e(k)|^{r(k)-2} e(k)\boldsymbol{x}(k), \tag{4.6}$$

it results

$$r(k+1) = r(k) + 2\varrho\mu e(k+1)[1 + r(k) \ln |e(k)|]|e(k)|^{r(k)-2} e(k)\boldsymbol{x}^t(k)\boldsymbol{x}(k+1), \tag{4.7}$$

which is a similar relationship as for gradient step-size in GAS technique [46].

Equation (4.7) might provide a formula for an analytic approach. It seems that the update equation is more complicated in comparison with the above mentioned approaches. The presence of $\ln |e(k)|$ factor can create some problems when error is closed or equal to zero, unless the error exponent is greater than two. Also the error exponent can not be adapted, if $\boldsymbol{x}^t(k)\boldsymbol{x}(k+1) = 0$, as in the case of GAS algorithm [46].

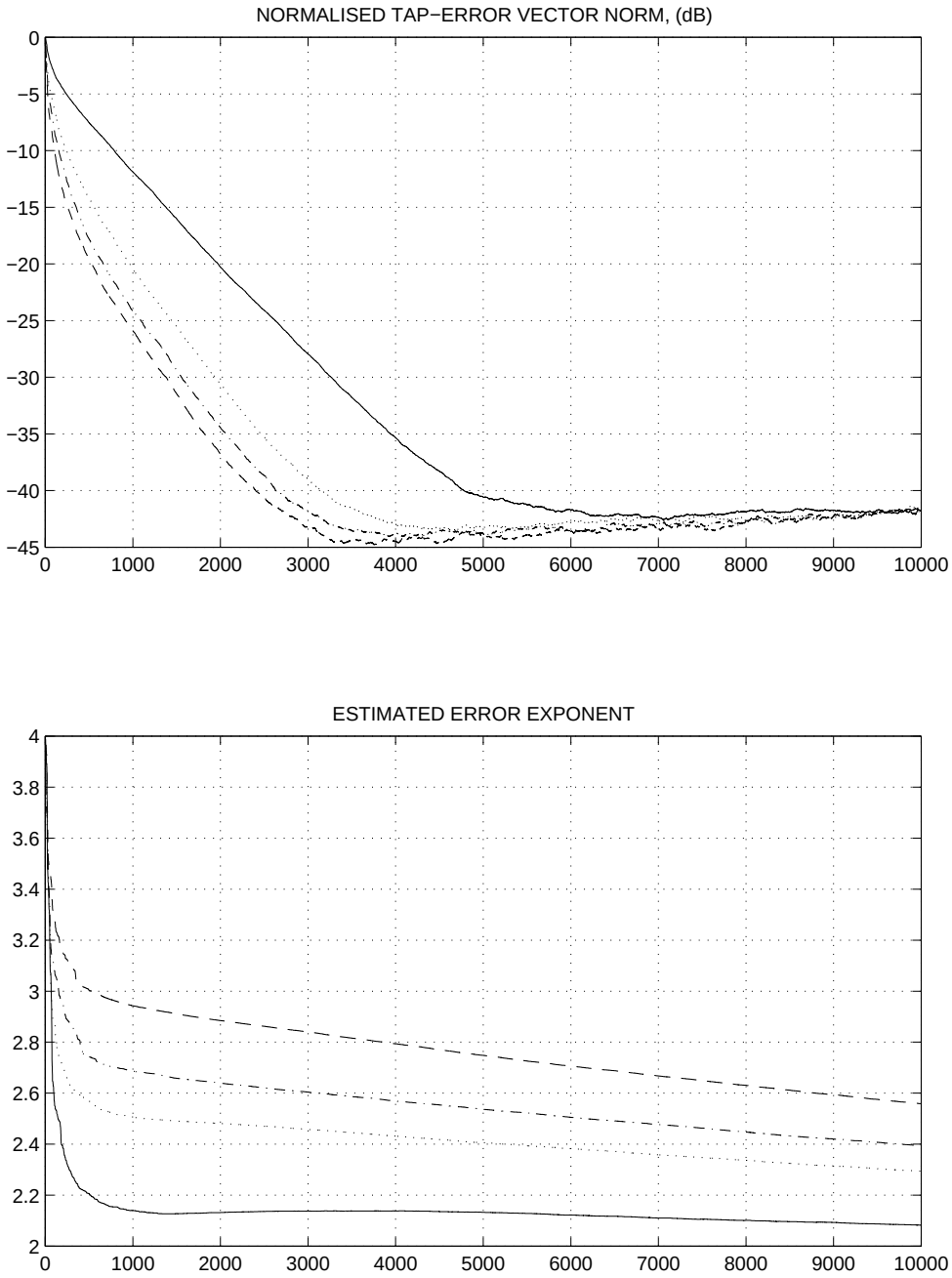


Figure 4.10: Performances of the RCFA algorithm for different step-sizes of the EPIS system $\mu = 5 \cdot 10^{-4}$ (——), $\mu = 10 \cdot 10^{-4}$ ($\cdots\cdots$), $\mu = 15 \cdot 10^{-4}$ ($\cdot - \cdot - \cdot$), $\mu = 20 \cdot 10^{-4}$ ($- - - -$).

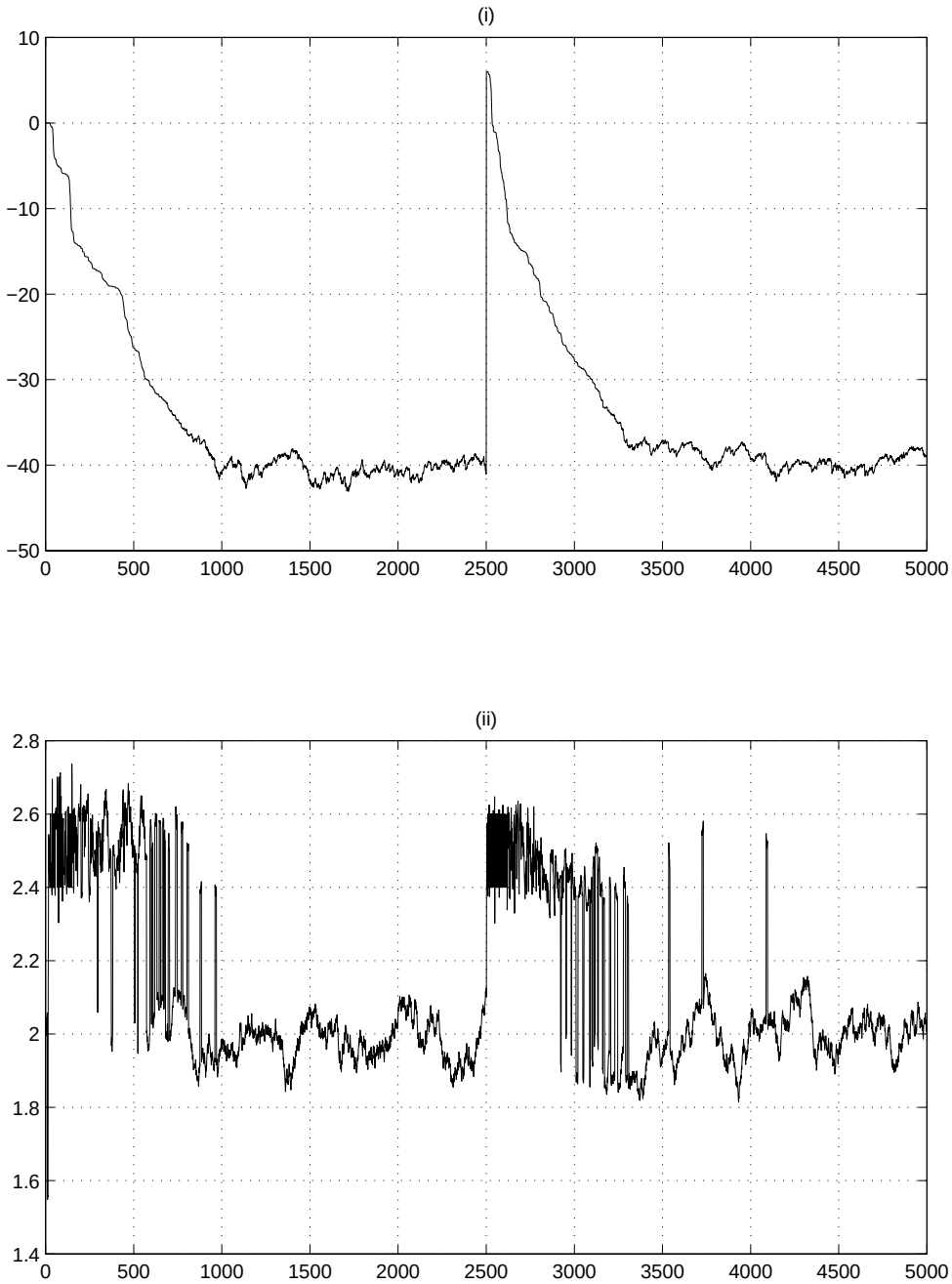


Figure 4.11: The normalised tap-error vector norm (i), and the estimated error exponent (ii) for the real hybrid echo-path for an abrupt change in the system.

All these difficulties suggest that the analytic approach as it was pointed out in this Section is not very suitable for data echo cancellation applications, though it might provide interest in other fields.

Chapter 5

Cost Function Adaptation Convergence

A general convergence and steady-state analysis of the proposed cost function adaptation algorithms is carried out in this Chapter, and closed-form expressions for the step-size bounds, time-constants and misadjustment are obtained. The theoretical achievements are also compared with computer simulations results.

5.1 Convergence and Steady-State Analysis

Both the stationary and gradient CFA approaches have been obtained by using an update of the form $r(k+1) \equiv F[|e(k)|]$. The results can be improved (Sections 3.1.2 and 3.1.4) by replacing the instantaneous values with finite-time averages. Also in the heuristic approach of RCFA, the EEAS subsystem can be designed to converge (Section 4.1.2). In all these cases, the error exponent sequence can be bounded:

$$0 < r_- \leq r(k) \leq r_+.$$

In these specific conditions we will develop the next convergence analysis. In the following we assume that:

- $\{x(k)\}$, $\{f(k)\}$, and $\{\Delta h(k)\}$ are independent real bipolar sequences;
- the autocorrelation matrix ϕ_{xx} of the input sequence $x(k)$ is positive definite;
- the echo path is stationary.

If we consider only real signals, the gradient update is given by Equation (2.4), providing that in the case of the gradient approach the constant β is absorbed into the constant μ.

5.1.1 Convergence in the mean

First we are interested in finding the conditions for the convergence of the mean of the tap-error vector. We will show that $E[\Delta \boldsymbol{h}(k)] \to 0$, in the case of small deviations of $\hat{\boldsymbol{h}}(k)$ from $\boldsymbol{h}(k)$:

$$|\boldsymbol{x}^t(k)\Delta \boldsymbol{h}(k)| \ll |f(k)| \tag{5.1}$$

From Equation (2.4) we get

$$\Delta \boldsymbol{h}(k+1) = \Delta \boldsymbol{h}(k) + \mu r(k)\boldsymbol{x}(k)|e(k)|^{r(k)-1}\mathrm{sgn}[e(k)].$$

and using (2.7), we have

$$\Delta \boldsymbol{h}(k+1) = \Delta \boldsymbol{h}(k) + \mu r(k)\boldsymbol{x}(k)|f(k) - \boldsymbol{x}^t(k)\Delta \boldsymbol{h}(k)|^{r(k)-1}\mathrm{sgn}[f(k) - \boldsymbol{x}^t(k)\Delta \boldsymbol{h}(k)].$$

According to our previous assumptions:

$$\Delta \boldsymbol{h}(k+1) \cong \Delta \boldsymbol{h}(k) + \mu r(k)\boldsymbol{x}(k)|f(k) - \boldsymbol{x}^t(k)\Delta \boldsymbol{h}(k)|^{r(k)-1}\mathrm{sgn}[f(k)]. \tag{5.2}$$

A little manipulation of Equation (5.2) gives:

$$\Delta \boldsymbol{h}(k+1) \cong \Delta \boldsymbol{h}(k) + \mu r(k)\boldsymbol{x}(k)|f(k)|^{r(k)-1}\left(1 - \frac{\boldsymbol{x}^t(k)\Delta \boldsymbol{h}(k)}{f(k)}\right)^{r(k)-1}\mathrm{sgn}[f(k)].$$

Applying the binomial theorem [7], and then ignoring the higher order terms:

$$\begin{aligned}\Delta \boldsymbol{h}(k+1) &\cong \Delta \boldsymbol{h}(k) + \mu r(k)\boldsymbol{x}(k)|f(k)|^{r(k)-1}\left[1 - (r(k)-1)\tfrac{\boldsymbol{x}^t(k)\Delta \boldsymbol{h}(k)}{f(k)}\right]\mathrm{sgn}[f(k)] \\ &\cong \Delta \boldsymbol{h}(k) + \mu r(k)\boldsymbol{x}(k)\mathrm{sgn}[f(k)]|f(k)|^{r(k)-1} - \mu r(k)(r(k)-1)|f(k)|^{r(k)-2}\phi_{xx}\Delta \boldsymbol{h}(k).\end{aligned}$$

After taking mathematical expectations on both sides, as $E[\boldsymbol{x}(k)] = 0$, the middle term on the right hand side of previous relationship will vanish, and we conclude that:

$$E[\Delta \boldsymbol{h}(k+1)] = (\boldsymbol{I} - \mu\phi_{xx}E[r(k)(r(k)-1)|f(k)|^{r(k)-2}])E[\Delta \boldsymbol{h}(k)]. \tag{5.3}$$

We denote by

$$\boldsymbol{\Delta} = \boldsymbol{I} - \mu\phi_{xx}E[r(k)(r(k)-1)|f(k)|^{r(k)-2}],$$

or in canonical form as,

$$\boldsymbol{\Delta} = \boldsymbol{A} \begin{pmatrix} \delta_1 & & 0 \\ & \ddots & \\ 0 & & \delta_N \end{pmatrix} \boldsymbol{A}^t, \tag{5.4}$$

where $\boldsymbol{A}\boldsymbol{A}^t = \boldsymbol{I}$, then (5.3) will become

$$E[\Delta \boldsymbol{h}(k+1)] = \boldsymbol{\Delta} E[\Delta \boldsymbol{h}(k)].$$

Let λ be the maximum eigenvalue of ϕ_{xx}. We can choose μ in such a way that all the leading diagonal elements of $\boldsymbol{\Delta}$ $(\delta_i, i = 1, \ldots, N)$, will have absolute values less than 1:

$$\delta \equiv sup|\delta_i| < 1, i = 1, \ldots, N,$$

We get:

$$0 < \mu < \frac{1}{E[r(k)(r(k)-1)|f(k)|^{r(k)-2}]\lambda}. \tag{5.5}$$

In this case, for the L^2 norm of $E[\Delta \boldsymbol{h}(k)]$ we have the following

$$\begin{aligned} E[\Delta \boldsymbol{h}^t(k+1)]E[\Delta \boldsymbol{h}(k+1)] &= E[\Delta \boldsymbol{h}^t(k)]\boldsymbol{A} \begin{pmatrix} \delta_1^2 & & 0 \\ & \ddots & \\ 0 & & \delta_N^2 \end{pmatrix} \boldsymbol{A}^t E[\Delta \boldsymbol{h}(k)] \\ &\leq \delta^2 E[\Delta \boldsymbol{h}^t(k)]E[\Delta \boldsymbol{h}(k)], \end{aligned}$$

and the adaptation process will lead to the convergence.

The condition set in Equation (5.5) might be difficult to check in practice. We can, however, bound the maximal eigenvalue of a positive definite matrix by its trace [20]:

$$\mathrm{tr}(\phi_{xx}) = NE[x_k^2],$$

and so we can obtain an easily applied sufficient condition of the mean of the tap-error vector of the cost function adaptation algorithm:

$$0 < \mu < \frac{1}{NE[r(k)(r(k)-1)|f(k)|^{r(k)-2}]E[x_k^2]}. \tag{5.6}$$

In the following we will calculate the upper bound on the step-size μ using Equation (5.5). In addition to previous assumptions, we consider that the error exponent is uniform linearly decreasing during adaptation, and as a consequence its probability density function is uniform between $[r_{min}, r_{max}]$. In this case, taking into account that the far-end signal is considered as a bipolar sequence, and $|f(k)| = a, \forall k \in \mathbb{N}$, then we have:

$$E[r(k)(r(k)-1)|f(k)|^{r(k)-2}] = \frac{1}{r_{max}-r_{min}} \int_{r_{min}}^{r_{max}} x(x-1)a^{x-2}dx$$

$$= \frac{1}{r_{max}-r_{min}} \left[r_{max}(r_{max}-1)\frac{a^{r_{max}-2}}{\ln a} - (2r_{max}-1)\frac{a^{r_{max}-2}}{(\ln a)^2} + 2\frac{a^{r_{max}-2}}{(\ln a)^3} \right. \tag{5.7}$$

$$\left. -r_{min}(r_{min}-1)\frac{a^{r_{min}-2}}{\ln a} + (2r_{min}-1)\frac{a^{r_{min}-2}}{(\ln a)^2} - 2\frac{a^{r_{min}-2}}{(\ln a)^3} \right].$$

The calculated value of the upper bound on μ given by Equations (5.6) and (5.7) will be compared in Section 5.2 with those obtained by experiment.

Remark 8. *The measurement noise effect*

It is of interest to remark that the upper bound of the step-size from the right-hand side of Equation (5.5) is not independent of the measurement noise, and this is reminiscent of the LMF and NQSG algorithms too [71, 81]. However, by cost function adaptation we can reduce this effect by reaching $r(k) = 2$ as soon as possible.

Remark 9. *Sub-unitary error exponents and CFA stability*

From Equation (5.5) one concludes that if $r(k) < 1$ outside a finite interval, then the CFA algorithm could divergence for any value of the variable step-size. Indeed, since the term within the expectation brackets is negative, the right-hand side of (5.5) can be negative, therefore we have no permitted values for the step-size. According to Section 3.1.4, this is not possible for the proposed stationary CFA algorithms, because the error exponent equals two for large errors. Also, in the implementation of the rest of the proposed CFA algorithms, we avoid sub-unitary error exponents.

5.1.2 Convergence in the mean square

For the convergence of the variance of $\Delta\boldsymbol{h}(k)$, we will derive a range based on the same assumption of small deviations from the solution [81], i.e., $\Delta\boldsymbol{h}(k) \approx \mathbf{0}$. We shall use the same strategy in the vicinity of the solution as used before for the convergence of the mean, neglecting all terms which depend on $\Delta\boldsymbol{h}(k)$ to the power higher than two. From (2.4) we have

$$\Delta\boldsymbol{h}(k+1) = \Delta\boldsymbol{h}(k) + \mu r(k)|e(k)|^{r(k)-1}\text{sgn}[e(k)]\boldsymbol{x}(k)$$

$$\Delta\boldsymbol{h}^t(k+1) = \Delta\boldsymbol{h}^t(k) + \mu r(k)|e(k)|^{r(k)-1}\text{sgn}[e(k)]\boldsymbol{x}^t(k),$$

and

$$\Delta\boldsymbol{h}^t(k+1)\Delta\boldsymbol{h}(k+1) = \Delta\boldsymbol{h}^t(k)\Delta\boldsymbol{h}(k) + \mu r(k)|e(k)|^{r(k)-1}\text{sgn}[e(k)][\Delta\boldsymbol{h}^t(k)\boldsymbol{x}(k)+ \boldsymbol{x}^t(k)\Delta\boldsymbol{h}(k)] + \mu^2 r^2(k)|e(k)|^{2r(k)-2}\boldsymbol{x}^t(k)\boldsymbol{x}(k). \tag{5.8}$$

Using the relation $\Delta\boldsymbol{h}^t(k)\boldsymbol{x}(k) = \boldsymbol{x}^t(k)\Delta\boldsymbol{h}(k)$, we have

$$\Delta\boldsymbol{h}^t(k+1)\Delta\boldsymbol{h}(k+1) = \Delta\boldsymbol{h}^t(k)\Delta\boldsymbol{h}(k) + 2\mu r(k)|e(k)|^{r(k)-1}\text{sgn}[e(k)]\Delta\boldsymbol{h}^t(k)\boldsymbol{x}(k) + \mu^2 r^2(k)|e(k)|^{2r(k)-2}\boldsymbol{x}^t(k)\boldsymbol{x}(k).$$

In the following we shall use the above mentioned condition $|\boldsymbol{x}^t(k)\Delta\boldsymbol{h}(k)| \ll |f(k)|$, and the formula[1]:

$$(1+\xi)^n \cong 1 + n\cdot\xi + \frac{n(n-1)}{2}\cdot\xi^2,\ |\xi| \ll 1. \tag{5.9}$$

The expression

$$|e(k)|^Q = |f(k) - \Delta\boldsymbol{h}^t(k)\boldsymbol{x}(k)|^Q = |f(k)|^Q\left[1 - \frac{\Delta\boldsymbol{h}^t(k)\boldsymbol{x}(k)}{f(k)}\right]^Q,$$

is approximated as

$$|e(k)|^Q \cong |f(k)|^Q\left[1 - Q\cdot\frac{\Delta\boldsymbol{h}^t(k)\boldsymbol{x}(k)}{f(k)} + \frac{Q(Q-1)}{2}\cdot\left(\frac{\Delta\boldsymbol{h}^t(k)\boldsymbol{x}(k)}{f(k)}\right)^2\right]$$
$$\cong |f(k)|^Q - Q\cdot\Delta\boldsymbol{h}^t(k)\boldsymbol{x}(k)|f(k)|^{Q-1}\text{sgn}[f(k)] + \frac{Q(Q-1)}{2}\cdot\left(\Delta\boldsymbol{h}^t(k)\boldsymbol{x}(k)\right)^2|f(k)|^{Q-2}, \tag{5.10}$$

then after inserting (5.10) in (5.8), and using $\text{sgn}[e(k)] \cong \text{sgn}[f(k)]$, we obtain

$$\begin{aligned}\Delta\boldsymbol{h}^t(k+1)\Delta\boldsymbol{h}(k+1) = \Delta\boldsymbol{h}^t(k)\Delta\boldsymbol{h}(k) &+ 2\mu r(k)|f(k)|^{r(k)-1}\text{sgn}[f(k)]\Delta\boldsymbol{h}^t(k)\boldsymbol{x}(k)\\ &-2\mu r(k)[r(k)-1]|f(k)|^{r(k)-2}[\Delta\boldsymbol{h}^t(k)\boldsymbol{x}(k)]^2\\ &+\mu r(k)[r(k)-1][r(k)-2]|f(k)|^{r(k)-3}\text{sgn}[f(k)]\left[\Delta\boldsymbol{h}^t(k)\boldsymbol{x}(k)\right]^3\\ &+\mu^2 r^2(k)|f(k)|^{2r(k)-2}\boldsymbol{x}^t(k)\boldsymbol{x}(k)\\ &-2\mu^2 r^2(k)[r(k)-1]|f(k)|^{2r(k)-3}\text{sgn}[f(k)][\Delta\boldsymbol{h}^t(k)\boldsymbol{x}(k)]\boldsymbol{x}^t(k)\boldsymbol{x}(k)\\ &+\mu^2 r^2(k)[r(k)-1][2r(k)-3]|f(k)|^{2r(k)-4}\left[\Delta\boldsymbol{h}^t(k)\boldsymbol{x}(k)\right]^2\boldsymbol{x}^t(k)\boldsymbol{x}(k).\end{aligned} \tag{5.11}$$

[1]By Taylor's formula [7], we have

$$(1+\xi)^n - 1 - n\xi - \frac{n(n-1)}{2}\xi^2 = \frac{n(n-1)(n-2)}{6}\xi^3(1+\theta\xi)^{n-3},\quad 0<\theta<1.$$

By our assumptions $|\xi| \ll 1$ and $1+\theta\xi \cong 1$, so that Equation (5.9) is true.

Now we assume that the $\Delta\boldsymbol{h}(k)$ is given. We take also the expectations of both sides of Equation (5.11) over all possible $\Delta\boldsymbol{h}(k)$. The terms on the right hand side which include odd powers of $\Delta\boldsymbol{h}^t(k)\boldsymbol{x}(k)$ will vanish under expectation due to the term $\mathrm{sgn}[f(k)]$. Hence

$$E[\Delta\boldsymbol{h}^t(k+1)\Delta\boldsymbol{h}(k+1)] = \Delta\boldsymbol{h}^t(k)\Delta\boldsymbol{h}(k)$$
$$-2\mu E[r(k)(r(k)-1)|f(k)|^{r(k)-2}]\Delta\boldsymbol{h}^t(k)E[\boldsymbol{x}(k)\boldsymbol{x}^t(k)]\Delta\boldsymbol{h}(k)$$
$$+\mu^2 E[r^2(k)|f(k)|^{2r(k)-2}]E[\boldsymbol{x}^t(k)\boldsymbol{x}(k)]$$
$$+\mu^2 E[r^2(k)(r(k)-1)(2r(k)-3)|f(k)|^{2r(k)-4}]\Delta\boldsymbol{h}^t(k)E[\boldsymbol{x}^t(k)\boldsymbol{x}(k)\boldsymbol{x}(k)\boldsymbol{x}^t(k)]\Delta\boldsymbol{h}(k),$$

or

$$E[\Delta\boldsymbol{h}^t(k+1)\Delta\boldsymbol{h}(k+1)] = \Delta\boldsymbol{h}^t(k)\boldsymbol{\Gamma}\Delta\boldsymbol{h}(k) + \mu^2 E[r^2(k)|f(k)|^{2r(k)-2}]E[\boldsymbol{x}^t(k)\boldsymbol{x}(k)]$$

where

$$\boldsymbol{\Gamma} = \boldsymbol{I} - 2\mu E[r(k)(r(k)-1)|f(k)|^{r(k)-2}]E[\boldsymbol{x}(k)\boldsymbol{x}^t(k)]$$
$$+\mu^2 E[r^2(k)(r(k)-1)(2r(k)-3)|f(k)|^{2r(k)-4}]E[\boldsymbol{x}^t(k)\boldsymbol{x}(k)\boldsymbol{x}(k)\boldsymbol{x}^t(k)].$$

It is now clear that the convergence properties depend entirely on the nature of the matrix $\boldsymbol{\Gamma}$. In fact, the algorithm will converge if and only if the magnitudes of its eigenvalues are less than one. Now we will need an additional approximation ($N \gg 1$):

$$\boldsymbol{x}^t(k)\boldsymbol{x}(k) \cong NE[x^2(k)].$$

We have

$$\boldsymbol{\Gamma} = \boldsymbol{I} - \mu\Big\{2E[r(k)(r(k)-1)|f(k)|^{r(k)-2}]$$
$$-\mu\, E[r^2(k)(r(k)-1)(2r(k)-3)|f(k)|^{2r(k)-4}]NE[x^2(k)]\Big\}\phi_{xx}.$$

Since the autocorrelation matrix ϕ_{xx} is positive definite, all eigenvalues of $\boldsymbol{\Gamma}$ will have absolute values smaller than one if and only if

$$0 < \mu < \frac{2E[r(k)(r(k)-1)|f(k)|^{r(k)-2}]}{E[r^2(k)(r(k)-1)(2r(k)-3)|f(k)|^{2r(k)-4}]NE[x^2(k)]} \tag{5.12}$$

and

$$1 - \lambda\mu\Big\{2E[r(k)(r(k)-1)|f(k)|^{r(k)-2}]$$
$$-\mu\, E[r^2(k)(r(k)-1)(2r(k)-3)|f(k)|^{2r(k)-4}]NE[x^2(k)]\Big\} > -1.$$

These conditions will be satisfied, if we properly choose μ.

Remark 10.

By using the same assumptions as when we computed Equation (5.7), we get the following expression for the expectation from the denominator of the Equation (5.12):

$$\begin{aligned} & E[r^2(k)(r(k)-1)(2r(k)-3)|f(k)|^{2r(k)-4}] \\ &= \frac{1}{r_{max}-r_{min}} \int_{r_{min}}^{r_{max}} x^2(x-1)(2x-3)a^{2x-4}dx \\ &= \frac{1}{r_{max}-r_{min}} \sum_{k=0}^{4} P^{(k)}(x) \frac{a^{2x-4}}{(2\ln a)^{k+1}} \Big|_{r_{min}}^{r_{max}}, \end{aligned} \tag{5.13}$$

where $P(x) = x^2(x-1)(2x-3)$. With this result we can compute another upper bound on the step-size μ.

5.1.3 Time constants

Now we proceed with the assessment of every time constant, i.e. the time taken by the corresponding mode of convergence to reach 36.8% of its initial value. Again we assume that the current estimate of the adaptive filter coefficients $\hat{\boldsymbol{h}}(k)$ is in the vicinity of the target filter taps $\boldsymbol{h}(k)$, so that the approximation of Equation (5.3) holds. For the above we also assume that the vectors $\Delta\boldsymbol{h}(k)$ and $\boldsymbol{x}(k)$ are independent of each other. From Equations (5.3) and (5.4) we deduce that generally there will be N different[2] relaxation time constants of the filter taps

$$T_i = \frac{1}{\mu E[r(k)(r(k)-1)|f(k)|^{r(k)-2}]\lambda_i}, \tag{5.14}$$

where λ_i are the eigenvalues of the autocorrelation matrix ϕ_{xx} of the input signal.

5.1.4 Steady-state analysis

The last step in our analysis of the adaptive process will be the evaluation of the misadjustment. Since the misadjustment is defined only for the adaptive processes in steady state (after adaptive transients have died out), we can assume that the error vector $\Delta\boldsymbol{h}(k)$ is small, close to zero. Therefore we can use once more the basic expression given in Section 5.1.2, where the terms multiplied by $\text{sgn}[f(k)]$ will vanish when we take expectations

[2]They might be very closed each other (Section 3.2.5).

of both sides. Hence:

$$E[\Delta\boldsymbol{h}(k+1)\Delta\boldsymbol{h}^t(k+1)] = E[\Delta\boldsymbol{h}(k)\Delta\boldsymbol{h}^t(k)]$$
$$-\mu E[r(k)(r(k)-1)|f(k)|^{r(k)-2}(\boldsymbol{x}(k)\boldsymbol{x}^t(k)\Delta\boldsymbol{h}(k)\Delta\boldsymbol{h}^t(k) + \Delta\boldsymbol{h}(k)\Delta\boldsymbol{h}^t(k)\boldsymbol{x}(k)\boldsymbol{x}^t(k))]$$
$$+\mu^2 E[r^2(k)|f(k)|^{2r(k)-2}\boldsymbol{x}(k)\boldsymbol{x}^t(k)]$$
$$+\mu^2 E[r^2(k)(r(k)-1)(2r(k)-3)|f(k)|^{2r(k)-4}\boldsymbol{x}(k)\boldsymbol{x}^t(k)\Delta\boldsymbol{h}(k)\Delta\boldsymbol{h}^t(k)\boldsymbol{x}(k)\boldsymbol{x}^t(k)]. \tag{5.15}$$

Now we shall neglect the fourth term on the right hand side of Equation (5.15) since for any small μ it will be small relative to the second term. Moreover, assuming that the algorithm has converged and is in steady state, we have

$$E[\Delta\boldsymbol{h}(k+1)\Delta\boldsymbol{h}^t(k+1)] = E[\Delta\boldsymbol{h}(k)\Delta\boldsymbol{h}^t(k)]$$

then we get:

$$-\mu E[r(k)(r(k)-1)|f(k)|^{r(k)-2}(\boldsymbol{x}(k)\boldsymbol{x}^t(k)\Delta\boldsymbol{h}(k)\Delta\boldsymbol{h}^t(k) + \Delta\boldsymbol{h}(k)\Delta\boldsymbol{h}^t(k)\boldsymbol{x}(k)\boldsymbol{x}^t(k))]$$
$$+\mu^2 E[r^2(k)|f(k)|^{2r(k)-2}\boldsymbol{x}(k)\boldsymbol{x}^t(k)] = 0, \tag{5.16}$$

and using the assumption of the independence of the far-end sequence, near-end sequence and tap-error sequence, Equation (5.16) has a unique solution [4]:

$$E[\Delta\boldsymbol{h}(k)\Delta\boldsymbol{h}^t(k)] = \frac{\mu E[r^2(k)|f(k)|^{2r(k)-2}]}{2E[r(k)(r(k)-1)|f(k)|^{r(k)-2}]}\boldsymbol{I}.$$

The power of the additional noise at the system output due to noise weights can be approximated by:

$$E[\big(\Delta\boldsymbol{h}^t(k)\boldsymbol{x}(k)\big)^2] = \frac{\mu N E[r^2(k)|f(k)|^{2r(k)-2}]E[x^2(k)]}{2E[r(k)(r(k)-1)|f(k)|^{r(k)-2}]},$$

where we have neglected all the cross-terms given by the near-end and tap-error sequences. We shall now use the definition of the misadjustment as a ratio between the power of the error sequence due to the weight noise and the Wiener error sequence power. We obtain the final expression for the misadjustment:

$$\mathcal{M} = \frac{\mu N E[r^2(k)|f(k)|^{2r(k)-2}]E[x^2(k)]}{2E[f^2(k)]E[r(k)(r(k)-1)|f(k)|^{r(k)-2}]}$$
$$= \frac{E[r^2(k)|f(k)|^{2r(k)-2}]}{2E[|f(k)|^2]E[r(k)(r(k)-1)|f(k)|^{r(k)-2}]^2}\sum_{i=1}^{N}\frac{1}{T_i}$$

CFA-xx	r_{max}	r_{min}
CFA-05	4	1.2037
CFA-10	4	1.8126
CFA-15	4	2.6604
CFA-20	4	3.3426

Table 5.1: Initial (maximum) and final (minimum) error exponents.

Suppose now that the error exponent sequence is convergent, i.e. there exists a value of $r(\infty)$ so that $\lim_{k\to\infty} r(k) = r(\infty),\ r_- \leq r(\infty) \leq r_+$. When $k \to \infty$, the steady-state is identical with NQSG$r(\infty)$ and the final misadjustment is given by

$$\mathcal{M} = \frac{E\left[|f(k)|^{2r(\infty)-2}\right]}{2(r(\infty)-1)^2 E\left[|f(k)|^2\right] E\left[|f(k)|^{r(\infty)-2}\right]^2} \cdot \sum_{i=1}^{N} \frac{1}{T_i}.$$

If $|f(k)|$ is constant, as it is the case of Example 3, then we have

$$\mathcal{M} = \frac{1}{2[r(\infty)-1]^2} \cdot \sum_{i=1}^{N} \frac{1}{T_i}. \tag{5.17}$$

5.2 Simulation Results

There is a general difficulty if we want to compare the values obtained in this analysis with those given by the simulations. Our assumption was that the system is near the optimum, where the error exponent should be also close to $r(\infty)$. But for almost all the algorithms we have developed and the experiments we have done, the error exponent during adaptation is at a considerable distance from $r(\infty)$. The following example will emphasize these aspects.

Example 18. *Example 3 revisited*

Several CFAxx approaches have been introduced in Example 3. In the same framework, now we shall present other algorithms performances. First in Figure 5.1 we illustrate the error exponent update, as it is obtained from simulations. Thus we can see that except CFA05, where the assumption of a smoothed error exponent (Section 3.1.3) might not be respected, the final error exponent is very close to the values suggested by Figure 3.5. These measured maximum and minimum error exponents are presented in Table 5.1.

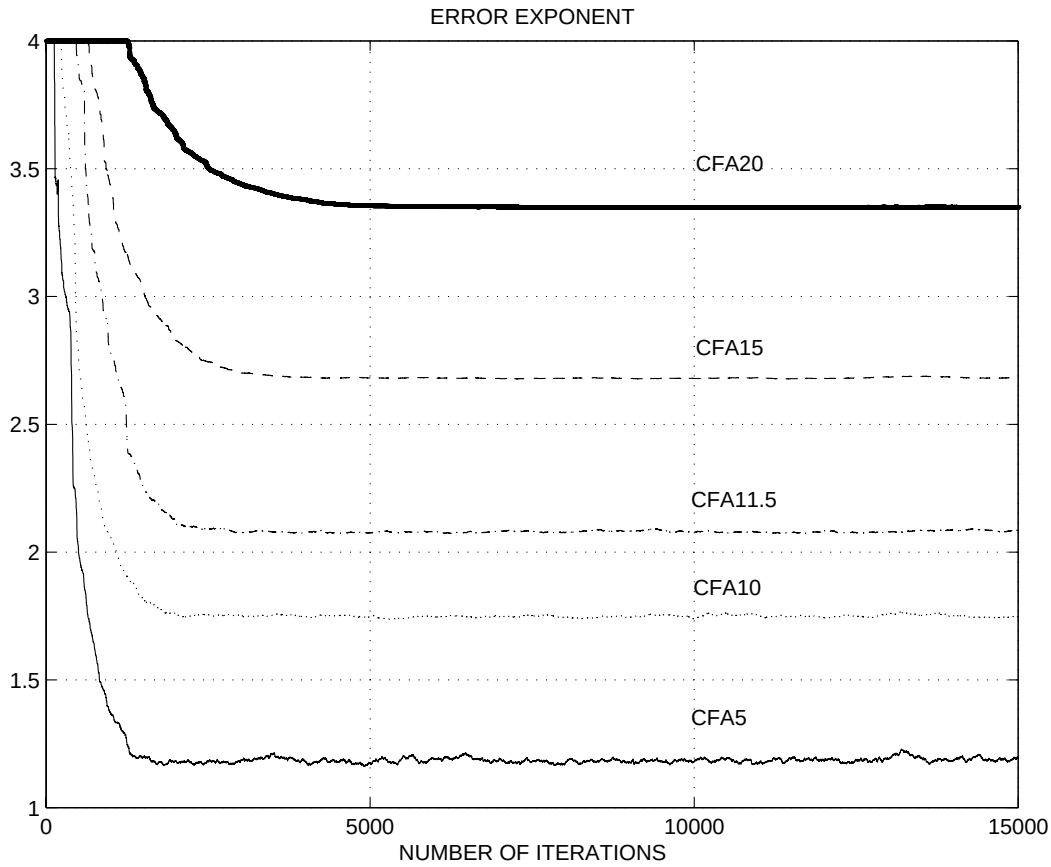


Figure 5.1: Error exponent update (Example 3) for different CFAxx initializations constants.

Figure 5.2 shows the behaviour of the mean-square error for the same experiment, obtained by averaging 50 runs. The time-constants differ from an approach to other, but the difference is quite small. However, we can realize that CFA05 approach (curve (d)) has the lowest convergence speed, and this is an effect of the short error exponent update period, and also of the small final error exponent.

The computed and measured values of the time constants are shown in Table 5.2, and they are obtained using Equation (5.14), and respectively the results from Figure 5.2. It is easy to see that the differences between measured and computed values depend on the distance between $r(0)$ and $r(\infty)$.

We compute the misadjusment by inserting the previous computed and measured time constants, and final error exponents in Equation (5.17). The measured misadjustment was obtained directly from definition [35] and by using the results shown in Figure 5.2.

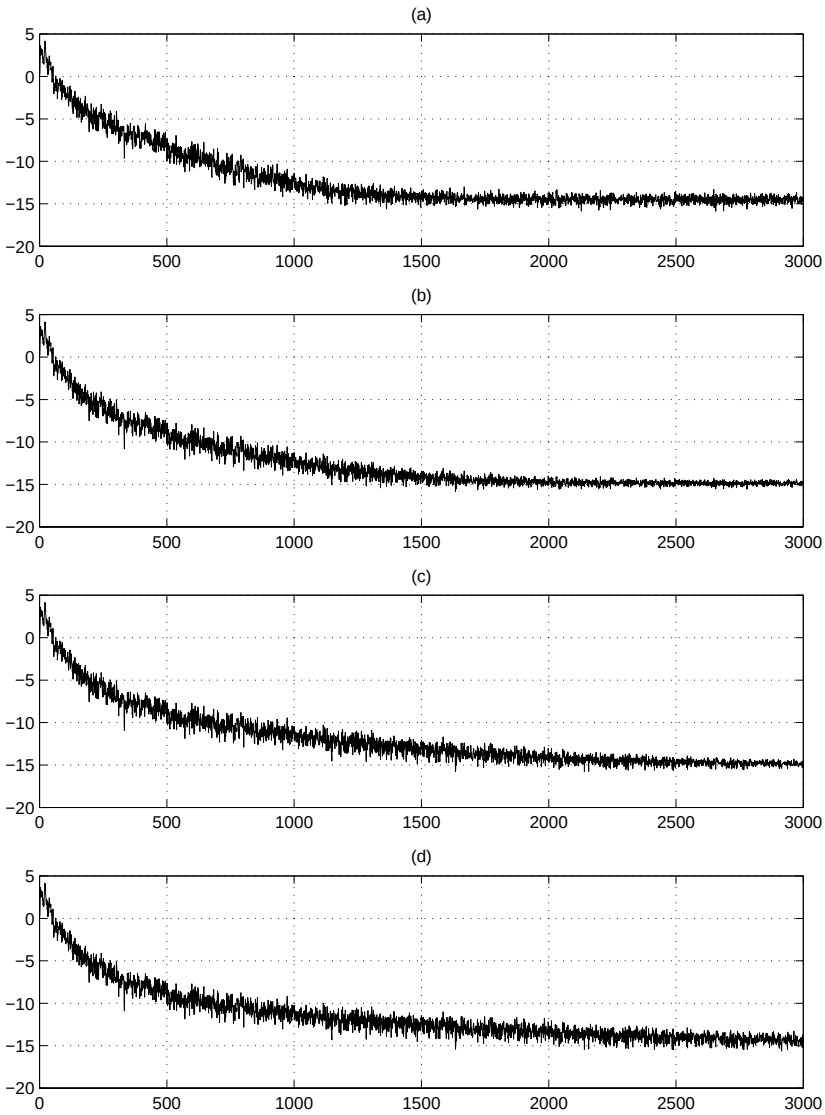


Figure 5.2: Mean-square error for CFA20 (a), CFA15 (b), CFA10 (c), and CFA05 (d) approaches.

The obtained values are provided in Table 5.3, and except for CFA05 the computed and measured misadjustment are quite close.

Finally, the maximum computed and measured step-sizes are presented in Table 5.4. First we use Equations (5.6) and (5.12) to calculate the maximum step-size. Then the measured step-size was obtained by averaging 20 times the learning curves for every increasing μ until divergence occurs.

CFA-xx	T_i computed	T_i measured
CFA-05	838	333
CFA-10	713	311
CFA-15	639	311
CFA-20	462	311

Table 5.2: Computed and measured time-constants.

CFA-xx	$\mathcal{M}$ measured	$\mathcal{M}$ computed with T_i computed	$\mathcal{M}$ computed with T_i measured
CFA-05	0.4596	0.1205	1.1580
CFA-10	0.0340	0.0303	0.0779
CFA-15	0.0091	0.0054	0.0187
CFA-20	0.0063	0.0014	0.0094

Table 5.3: Computed and measured misadjustment.

The big difference between μ_{max} computed and μ_{max} measured could be the result of several factors:

1. The assumption that we are near the optimum.
2. The approximations made in the analysis while obtaining the expression for the calculation of μ_{max} computed.
3. The insufficient number of averagings and the selected convergence level to detect the divergence which have to be increased according to hump effect [58].

We conclude that from qualitative point of view the convergence and steady-state analysis provided in this Chapter emphasized the general assertions used in previous Chapters.

CFA-xx	μ_{max} from Equation (5.6)	μ_{max} from Equation (5.12)	μ_{max} measured
CFA-05	0.0216	0.0485	0.0016
CFA-10	0.0184	0.0352	0.0015
CFA-15	0.0165	0.0445	0.0011
CFA-20	0.0119	0.0668	0.0007

Table 5.4: Computed and measured maximum step-sizes.

Chapter 6

Related Techniques

The first part of the chapter introduces the Convex Variable Step-Size (CVSS) algorithm. The convexity of the resulting cost function is guaranteed. Simulations presented show that with the proposed algorithm we obtain similar results as with VSS [42] algorithm. Then a threshold technique with quadratic algorithms is discussed. The idea is to decide whether for the next step we apply LMS or LMF, based on a comparison of an error estimate with some selected thresholds.

6.1 Convex Variable Step-Size

The convexity of the employed cost function was an important goal in adaptive filtering problems, as in this case every minimum is a global minimum [35]. Thus any optimum weight vector the adaptive algorithm converges to is globally optimal [52]. Most of the combined LMS and LMF methods used a time-varying combination of the LMF and LMS cost functions. The resulting cost function is not anymore convex, or at least this property is not obvious. Perhaps the one for which this quality is easy to prove is the mixed-norm LMMN (Least-Mean Mixed Norm) cost function [15], where a constant mixing parameter was used. This is also true for the variable step-size algorithms, where heuristic methods have been found useful in practice, even they cannot claim any sort of optimality in their performance [26]. An often encountered example is the variable step-size (VSS) adaptive filtering algorithm, which is based on the fluctuation of the prediction squared error [42]. An overall weakness of the variable step-size algorithms is that the convexity of the cost function cannot be anymore guaranteed. For instance, we may consider the least mean P-algorithms as the LMS algorithm with time-varying step-size [52] given by (2.2), but for $0 < P < 1$ the corresponding cost function $J_P = E[|e(k)|^P]$ is not anymore convex. Thus

we have doubts that the variable step-size techniques provide unavoidably convex cost functions, and also seems very difficult to show this at least in the case of heuristic variable step-size methods. In this Section our goal is to derive such a variable step-size method for which the convexity of cost function is proved. First we recall some theoretical aspects of convex functions. Then the algorithm is derived, and finally experimental results are shown. For the beginning we recall a known result [47] (pp.18):

Theorem 1. *Let be a function $F : \mathbb{R} \to \mathbb{R}$, which is differentiable and its derivative is a nondecreasing function on $\mathbb{R}$. In this case, F is a convex function.*

Example 19.

$$F : \mathbb{R} \to \mathbb{R},\ F(x) = \begin{cases} Ax^2 - \dfrac{A^2 - C^2}{4B}, & \text{if } |x| > E_1, \\ Bx^4 + \dfrac{C^2}{4B}, & \text{if } E_0 \le |x| \le E_1, \\ Cx^2, & \text{if } |x| < E_0, \end{cases} \tag{6.1}$$

where A, B, C are positive constants, $A > C$, and the constants E_0, and E_1 are such that

$$E_0 = \sqrt{\frac{C}{2B}},\ E_1 = \sqrt{\frac{A}{2B}}. \tag{6.2}$$

F is continuous and differentiable, and its derivative is given by

$$\frac{\partial F}{\partial x} = \begin{cases} 2Ax, & \text{if } |x| > E_1, \\ 4Bx^3, & \text{if } E_0 \le |x| \le E_1, \\ 2Cx, & \text{if } |x| < E_0. \end{cases}$$

This derivative is nondecreasing and continuous (Figure 6.1), thus F is convex.

Now we are going back to the general case when F is convex, and in the following we study the cost function $J = E\{F[e(k)]\}$. Following [52], for every $\Delta\boldsymbol{h}_1$, $\Delta\boldsymbol{h}_2$, and $\lambda \in [0, 1]$, we have

$$F[f - \boldsymbol{x}^t(\lambda\Delta\boldsymbol{h}_1 + (1-\lambda)\Delta\boldsymbol{h}_2)] \le \lambda F(f - \boldsymbol{x}^t\Delta\boldsymbol{h}_1) + (1-\lambda)F(f - \boldsymbol{x}^t\Delta\boldsymbol{h}_2).$$

Multiply both side of previous equation by the joint probability density function of f and $\boldsymbol{x}$, integrate with respect to them, and after we take the expected values of both sides, we obtain that

$$J\{\lambda\Delta\boldsymbol{h}_1 + (1-\lambda)\Delta\boldsymbol{h}_2)\} \le \lambda J\{\Delta\boldsymbol{h}_1\} + (1-\lambda)J\{\Delta\boldsymbol{h}_2\},$$

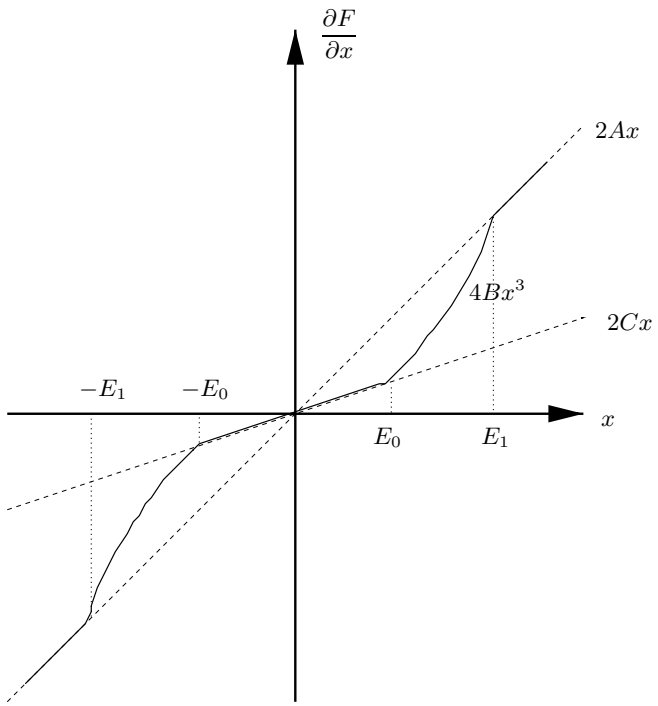


Figure 6.1: Derivative of the function given by (6.1).

i.e. the cost function $J = E\{F[e(k)]\}$ is a convex function over $\mathbb{R}^N$.

Proceeding further, we consider the following even-symmetric error criterion in instantaneous form:

$$J_{CVSS}[e(k)] = F[e(k)] = \begin{cases} Ae^2(k) - \dfrac{A^2 - C^2}{4B}, & \text{if} \quad |e(k)| > E_1, \\ Be^4(k) + \dfrac{C^2}{4B}, & \text{if} \quad E_0 \leq |e(k)| \leq E_1, \\ Ce^2(k), & \text{if} \quad |e(k| < E_0, \end{cases} \tag{6.3}$$

where the constants are as before.

First we note that this type of cost function belongs to the class examined in [27], where general expressions for the mean and the mean-square convergence of the filter coefficients have been derived. Then, from above results the proposed cost function $J_{CVSS}[e(k)]$ is convex. The convex variable step-size algorithm results (μ is the fixed step-size of the

algorithm):

$$\begin{aligned}\hat{\boldsymbol{h}}(k+1) &= \hat{\boldsymbol{h}}(k) - \mu\nabla J_{CVSS} \\ &= \hat{\boldsymbol{h}}(k) - 2\mu\frac{\partial}{\partial e(k)}\left\{J_{CVSS}[e(k)]\right\}\frac{\partial e(k)}{\partial \hat{\boldsymbol{h}}(k)} \\ &= \hat{\boldsymbol{h}}(k) + 2\mu(k)e(k)\boldsymbol{x}(k),\end{aligned}$$

where:

$$\mu(k) = \begin{cases} A\mu, & \text{if} \quad |e(k)| > E_1, \\ 2B\mu e^2(k), & \text{if} \quad E_0 \leq |e(k)| \leq E_1, \\ C\mu, & \text{if} \quad |e(k| < E_0. \end{cases}$$

As in other variable step-size methods we have three regions: in two of them LMS acts with different gradients, and the middle one is where the LMF is applied. This last region, and the selection of the constants E_0 and E_1 will differentiate our algorithm from other variable step-size methods.

The proposed variable step-size algorithm has several benefits compared to the previous ones:

1. Convex cost function;

2. A simple implementation:

   - The selection of additional constants and initial step-size to control adaptive behaviour of the step-size sequence is clear and easy to handle according to the application;
   - The amount of computation load is reduced or is identical.
   - We do not need steps to be taken to anticipate the step-sizes from exceeding their maximum and minimum limits; the same comparison for error seems effortless.

3. The algorithm analysis is a subject already approached as it belongs to the general class discussed in [27], and does not cause major difficulties.

As a drawback, the CVSS algorithm can perform worse than some variable step-size algorithms, if the period when LMF acts, i.e. middle term of (6.3), is very long. This is the case of a big constant E_0, or a big ratio A/C. However, in this case, in the middle region instead of LMF another non-quadratic algorithm [71] could improve the performances. As we shall see in the following example, this artifice is not so often necessarily.

Example 20. *Constant E_0 influence in CVSS algorithm's behaviour*

We took into consideration for this example three channels: first two of them are echo path of the first model (single pole single zero digital filter), and the third is the real hybrid. Also the levels of attenuated far-end signal are $a = -15$ dB, $a = -20$ dB, $a = -25$ dB, and $a = -30$ dB. For every one of this we found the step-sizes (Table 6.1) μ_{min} (for which the convergence level was 20 dB below far-end signal level was reached) and μ_{max}. It seems that the obtained values are not very sensitive to noise levels. Then we compute the ratio $A/C = \mu_{max}/\mu_{min}$. Given E_0, from (6.2), and taking $C = 1$, the other constants results. We performed 20 averages of learning curves for these parameters, and we measure the number of iterations needed to reach the convergence level in every case. The results are shown in Tables 6.2, 6.3, and 6.4.

We remark a sort of dispersion of results, which can be explained by the fact, that here the variable step-size is not fixed, and this affects the residual error. In spite of this, it comes out that except for the extremal values, for a wide range of values E_0, the performances are quite the same.

Example 21. *A short comparison between CVSS and VSS algorithms*

First we tested the LMS algorithm in the designated framework, in our case the real hybrid. For $a_{dB} = -15$ we found that the maximum step-size for which instability we did not detect was $\mu_{max} = 0.01$. The required convergence level of the echo canceller is at -20 dB below the far-end signal, and we reached this level when the step-size was $\mu_{min} = 0.00275$.

The VSS implementation follows from recommendations in [42], with $\alpha = 0.97$, $\gamma = 48 \cdot 10^{-5}$.

For CVSS algorithm we took $\mu = \mu_{min}$, and we get the ratio $A/C = \mu_{max}/\mu_{min} = 3.64$. If $C = 1$, then $A = 3.64$. If we take $E_0 = 0.5$, then from (6.2) we obtain $B = C/(2E_0^2) = 2$, and $E_1 = E_0 \cdot \sqrt{A/C} = 0.95$.

μ	Table 6.2	Table 6.3	Table 6.4
maximum	0.01	0.005	0.01
minimum	0.000825	0.0005	0.00275

Table 6.1: Step-sizes used in simulations of Example 20.

E_0	$a = -15$ dB	$a = -20$ dB	$a = -25$ dB	$a = -30$ dB
0.30	1305	2068	2626	2988
0.31	1777	2082	2909	2710
0.32	1258	2215	2974	2891
0.33	1817	2104	2655	3293
0.34	1432	2417	2758	3044
0.35	2017	2136	2894	3193
0.36	1966	2610	3142	3209
0.37	1738	2440	2755	3192
0.38	1675	2165	2809	2975
0.39	1878	2776	3030	3030
0.40	1669	2396	3012	3128
0.41	1922	2123	3091	3191
0.42	1673	2669	2607	3337
0.43	1963	2278	2599	3385
0.44	2024	2029	3024	3348
0.45	1796	2353	2878	3099
0.46	2391	2489	2720	3304
0.47	1744	2310	2697	3005
0.48	2047	2278	3190	3604
0.49	2244	2600	2936	3071
0.50	1951	2716	2986	3485
0.51	2040	2740	2875	2977
0.52	2329	2232	2598	3011
0.53	1846	2500	2855	3108
0.54	2190	2511	2853	3237
0.55	1834	2349	3075	3057
0.56	2303	2548	2786	3086
0.57	2195	2515	2938	2944
0.58	2097	2781	2953	3483
0.59	2067	2810	2979	3330
0.60	2562	2789	3152	3344
0.61	2283	2664	2887	3491
0.62	2134	2797	2934	3647
0.63	2135	2851	3110	3416
0.64	2063	3003	3184	3569
0.65	2166	2759	3200	3370

Table 6.2: Number of iterations needed to reach the convergence level of CVSS for first echo path with $p = 0.80025$, $A = -60$, $N = 32$, for different constants E_0.

E_0	$a = -15$ dB	$a = -20$ dB	$a = -25$ dB	$a = -30$ dB
0.30	2726	3763	4339	5178
0.31	2745	4129	4133	4991
0.32	2276	3631	4788	5025
0.33	2459	3414	4390	5392
0.34	2722	3796	4050	5250
0.35	2744	3691	4141	4992
0.36	2896	3947	4562	5683
0.37	2392	4125	4959	5884
0.38	2910	4158	5223	5174
0.39	2807	3787	4790	5093
0.40	2770	3832	4909	5741
0.41	2959	3952	4450	5562
0.42	3092	4238	4886	5144
0.43	3007	3849	4881	5127
0.44	3407	4037	4907	5223
0.45	3086	3775	4874	6159
0.46	3106	4009	4714	5651
0.47	4287	4253	5051	5278
0.48	3431	4530	4617	5626
0.49	3664	4576	5151	5746
0.50	3168	5075	5404	5386
0.51	3651	4461	5507	5660
0.52	3524	4352	5108	5636
0.53	3329	4150	4912	5823
0.54	3537	4161	4906	5560
0.55	3474	4316	5042	5891
0.56	3857	4448	4847	5436
0.57	3668	4377	4917	5806
0.58	3744	4248	6172	5884
0.59	4054	5006	5093	6068
0.60	3823	4136	4892	5698
0.61	3919	4268	4866	5441
0.62	3604	4234	4634	5785
0.63	3925	4241	5234	5736
0.64	3601	4638	5292	6539
0.65	4199	4734	4800	5837

Table 6.3: Number of iterations needed to reach the convergence level of CVSS for first echo path with $p = 0.64$, $A = -120$, $N = 32$, for different constants E_0.

E_0	$a = -15$ dB	$a = -20$ dB	$a = -25$ dB	$a = -30$ dB
0.30	554	779	856	996
0.31	710	714	957	1007
0.32	500	799	1082	1089
0.33	523	760	998	1042
0.34	589	735	919	960
0.35	589	775	1097	1106
0.36	567	882	943	1225
0.37	619	746	937	1018
0.38	886	929	962	1121
0.39	737	902	1009	1182
0.40	625	824	1061	1065
0.41	640	749	1053	1146
0.42	667	853	943	1044
0.43	637	890	1090	1113
0.44	636	823	1226	1241
0.45	801	750	1024	1487
0.46	767	976	969	1040
0.47	836	881	1126	1182
0.48	718	808	1002	1243
0.49	826	998	881	1000
0.50	690	855	970	1147
0.51	741	1004	1129	1164
0.52	688	877	988	1050
0.53	748	938	972	1079
0.54	697	937	1054	1102
0.55	938	1188	1003	1181
0.56	933	798	881	1021
0.57	785	883	1049	1108
0.58	716	985	974	1270
0.59	806	901	1038	1128
0.60	716	757	1008	1142
0.61	1037	867	971	1172
0.62	852	890	1162	1357
0.63	709	1007	1115	1268
0.64	1043	946	986	1101
0.65	739	929	1106	1209

Table 6.4: Number of iterations needed to reach the convergence level of CVSS for real hybrid, for different constants E_0.

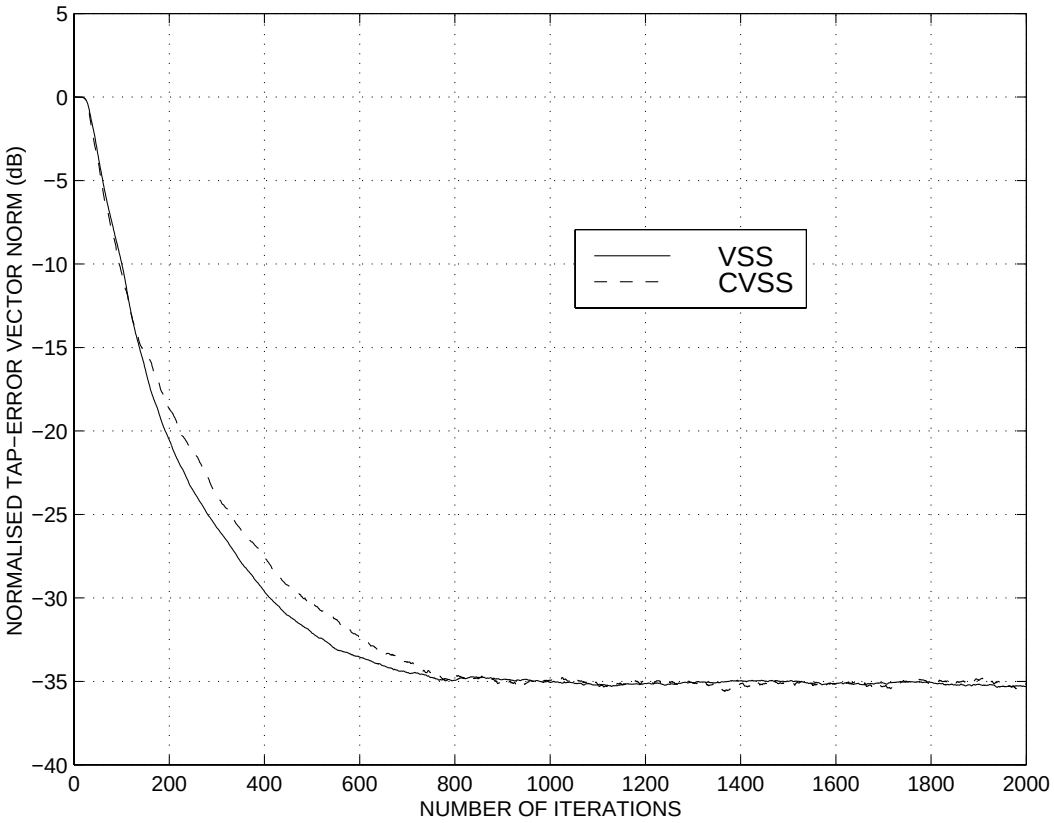


Figure 6.2: Learning curves of VSS and CVSS algorithms.

The first set of simulations we present is the stationary case, where the algorithms were running 100 times and averaged. The learning curves are presented in Figure 6.2. The second example (Figure 6.3) shows the behaviour of the CVSS and VSS algorithms when the unknown system coefficients experience a sign change at iteration $k = 2000$.

These results show that the algorithms under comparison achieve quite similar performances both in stationary and non-stationary cases.

6.2 Threshold Technique

A new kind of solution [57, 65] is proposed in this Section to address the mentioned problem (Section 2.2.4). The main idea is to apply the LMF algorithm at the beginning and at the end of the adaptation time, where its performances are superior to the LMS in the case of data echo cancellation. An important issue was how to decide the instants

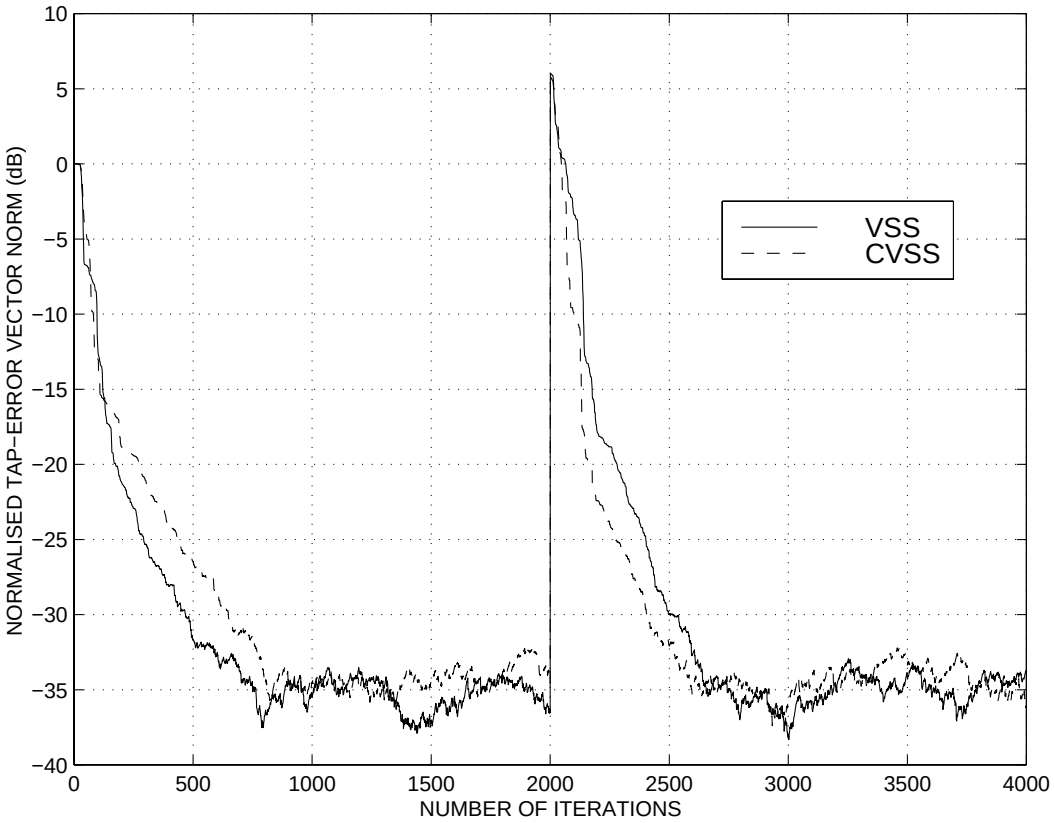


Figure 6.3: Performance of VSS and CVSS algorithms for a sign change in coefficients.

when we apply one of these two algorithms. One of our assumptions was that it is better to apply for a long time a certain algorithm. We considered that skipping from one to another could cause instability issues or just loose the benefits of the convexity of the error surfaces.

From the implementation point of view, the error mapping proposed in Section 3.2.2 is used again. The output of the filter is used for comparison with the selected thresholds to decide whether on the next step we apply LMS or LMF. Also we need to avoid possible oscillations: LMS to LMF and back. Our suggestion was to use two different hysteresis loops. As a consequence the LMFSF algorithm results:

Let be $V_4 > V_3 > V_2 > V_1 > 0$ the selected thresholds. Then for LMFSF:

1) The weights are computed with Equation (2.1);

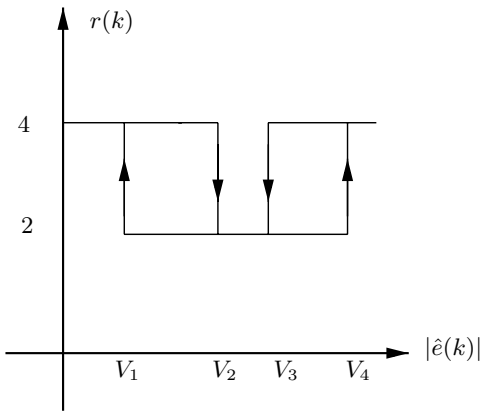


Figure 6.4: The error exponent update as a function of the magnitude of the error.

2) The power $r = r(k)$ is updated by:

$$r(k) = \begin{cases} 4, & \text{if: } \begin{cases} k < 3 \quad \text{or} \\ |\hat{e}(k-1)| < V_1 \quad \text{or} \\ |\hat{e}(k-2)| < |\hat{e}(k-1)| < V_2 \text{ or} \\ |\hat{e}(k-1)| > V_4 \quad \text{or} \\ |\hat{e}(k-2)| > |\hat{e}(k-1)| > V_3 \end{cases} \\ 2, & \text{otherwise.} \end{cases} \tag{6.4}$$

A graphical representation of the LMFSF algorithm is shown in Figure 6.4.

Thus we try to take advantage of the benefits given by the large gradients and good residual weight error from the LMF algorithm, and of the better performances of LMS during tracking and all adaptation period.

Example 22. *The threshold technique and quadratic algorithms*

The channel considered to test the LMFSF algorithm has one zero at the origin and one pole at 0.8, and the length of FIR adaptive filter is $N = 40$. The level of the attenuated far-end signal is $a_{dB} = -20$, and consequently the step-size of the adaptive algorithm is $\mu = 5 \cdot 10^{-4}$ to assure the convergence of the LMF.

The initial estimated error is $\hat{e}(0) = 4 \cdot a$. The constant of the recursive filter is $L = 500$. The thresholds are $V_1 = 0.1005$, $V_2 = 0.1125$, $V_3 = 0.375$, $V_4 = 0.4$. The learning curves obtained are the average of 20 runs.

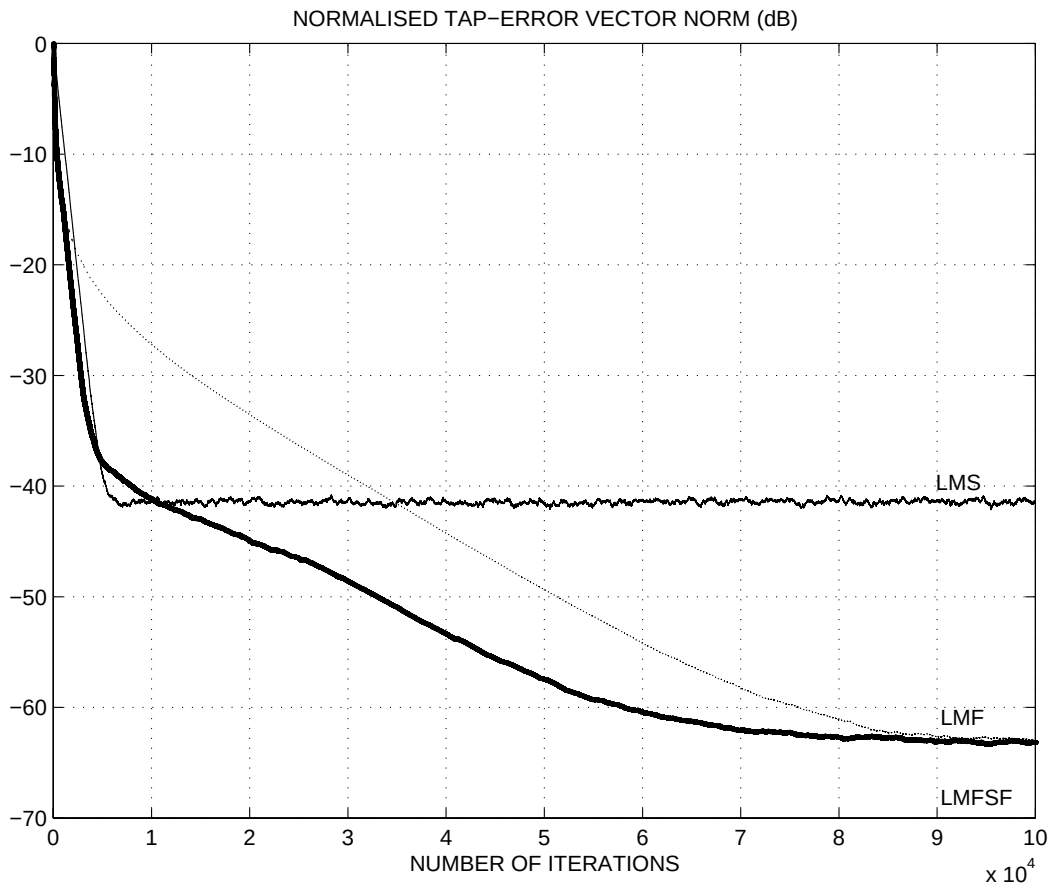


Figure 6.5: The learning curves of the LMS, LMF and LMFSF algorithms.

Figure 6.5 delivers a comparison between LMS, LMF and LMFSF algorithms, from both the convergence speed and steady-state point of view. It is clear now that the LMFSF algorithm has a faster convergence than both LMS and LMF algorithms, and this is a result of applying the LMF algorithm at the beginning of adaptation.

The steady-state also finds LMF acting, and as a consequence the residual error properties are similarly with the LMF algorithm. Thus the steady-state performances are better than of the LMS algorithm.

For more details, Figure 6.6 illustrates the performances of the above mentioned algorithms during the adaptation time, and now we can estimate the gain in adaptation speed in the case of the LMFSF algorithm. This is almost the same as other CFA techniques.

Example 23. *The threshold technique and switched error algorithms*

The last comparison addresses the family of switched error norms algorithms. As an

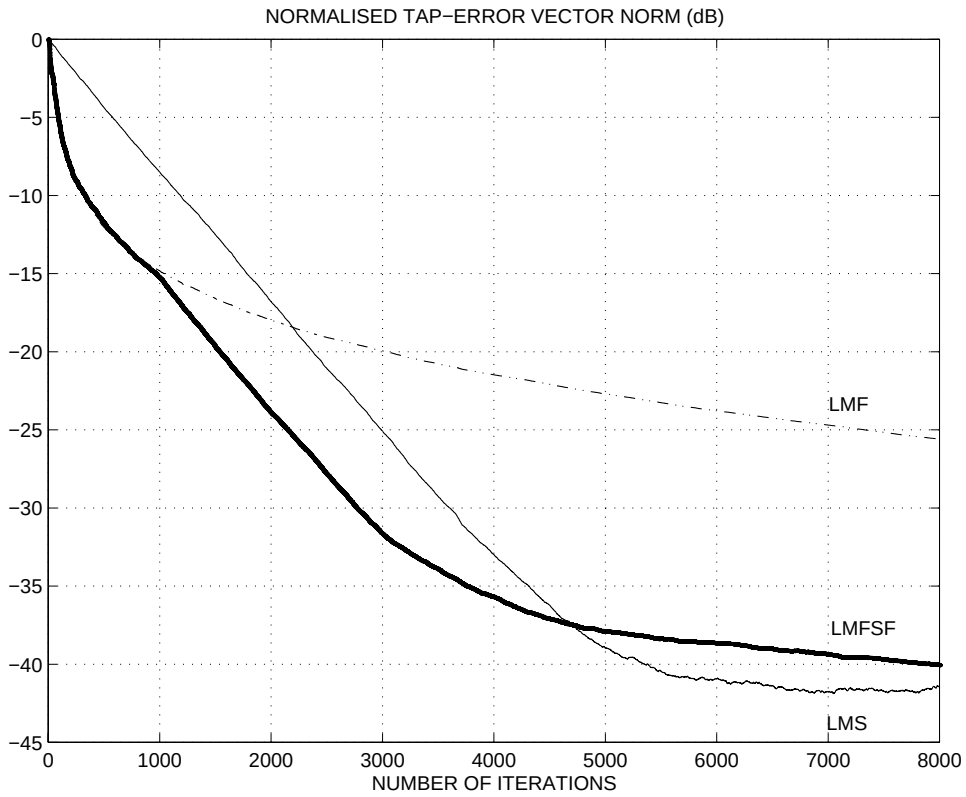


Figure 6.6: Adaptation period of the LMS, LMF and LMFSF algorithms.

example, we consider the LMSEN (Least Mean Switched Error Norm) algorithm, which consists of applying the LMF algorithm, and switching to the LMS algorithm when the absolute value of error is greater than one.

Figure 6.7 shows the performances of the LMFSF algorithm in comparison with the LMSEN algorithm [73], using the same parameters as before. It is clear now that the LMFSF algorithm behaves better than LMSEN algorithm, with little extra computation cost due to the implementation of the error estimator filter.

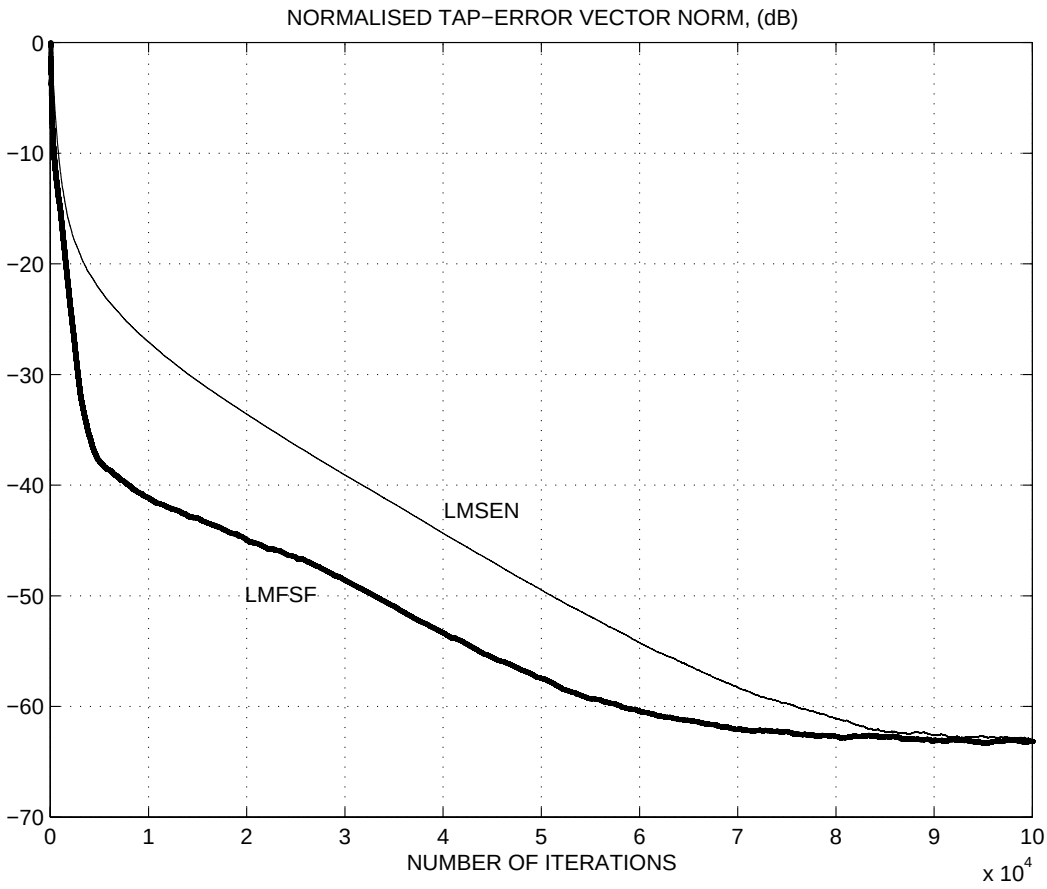


Figure 6.7: The learning curves of the LMFSF and LMSEN algorithms.

Part II

Logarithmic Gain and Phase Approximation

Chapter 7

Motivation and Problem Statement

A large number of scientific problems deals with Hilbert transform and their sampled derivations. In certain applications the domain is restricted, or other specific conditions are imposed. Nevertheless, some particular problems are encountered almost in every situation.

A critical issue is related to the singularities involved in the Hilbert transform computation, since we are confronted with an improper integral. If the integral cannot be evaluated in a closed form, as it is the case with discrete input data, numerical integration is in general complicated.

Another inconvenience is related to the properties of algorithms when noise is present. As it is known, the Hilbert transform can behave as an unbounded operator [2]. However, the situation is more promising for rational and stable matrix functions where an L^∞ continuity result was established [3]. There the bounds obtained were affine in n, where the complex matrix rational function has a McMillan degree[1] less than or equal to n. The affine-with-n nature is the best that can be expected.

In the following, the second part of the thesis discusses some aspects of the Hilbert transform for causal signals in the logarithmic domain, where several novel numerical approaches to Bode gain-phase relationships are the desired outcome. Implementation considerations and results of numerical experiments are also presented.

In the derivation of proposed methods we shall concentrate on minimum-phase functions, since the achievements can be very easily applied to non minimum-phase functions [11, 67]. In addition we restrict our discussion on functions defined only on real axis, therefore we consider only 1-D phase approximation case. All improper integrals will be

[1]Briefly, McMillan degree measures the degree of complexity of a circuit, for instance the number of delay cells needed for implementation. For a general definition, see [38].

interpreted as Cauchy principal values. This will apply also for singularities at finite point [49].

7.1 The Phase Retrieval Problem

The 1-D phase retrieval problem is to reconstruct a signal given the modulus of its Fourier transform. This problem is associated with various applications including antenna design, filter design, image reconstruction, wavefront sensing, electronic microscopy, and the characterization of astronomical objects [79]. The phase retrieval problem arises also in applications of electromagnetic theory in which wave phase is apparently lost or impractical to measure and only intensity data are available.

Solutions of the phase retrieval problem are of two types: solutions depending on analytic properties and solutions depending on numerical procedures. The analytical solutions are usually related with the logarithmic Hilbert transform. Customary solution techniques approximate the solution by discretizing the continuous problem.

We note also that 1D and 2D phase retrieval are completely different approaches. In the case of 1D phase retrieval we have to reconstruct the phase given the modulus of its Fourier transform. This is equivalent to synthesize the signal using inverse Fourier transform. There are some ambiguities in this problem. Clearly, if $x(n)$ is a solution, then its symmetric conjugate, scaling and time-shifting are also. Even excluding them, we can still have more solutions.

The difficulty of the problem consists in the fact that in 1D we cannot find the phase from the magnitude of the Fourier transform. More information about the signal is needed, for instance that the signal is of minimum phase. In 2D this is not anymore a problem, the phase retrieval problem has a unique solution, except the trivial ambiguities.

The phase retrieval is equivalent with autocorrelation retrieval. For instance, in 1D the solution can be found by solving the corresponding integral equation:

$$r(\tau) = \int x^*(t)x(t+\tau)dt, \tag{7.1}$$

where the autocorrelation function is obtained by inverse Fourier transform of squared modulus. Note that for the autocorrelation function the zeros occur in reciprocal conjugate pairs, and we need some supplementary information to derive a unique solution. Non-negativity is not enough, but knowledge of a single point can be helpful. The additional information can be also some samples of the signal, or imposing a specific condition on the support of minimum and non-minimum phase solutions [11, 67].

Other methods for solving phase-retrieval problem employ iterative algorithms applied to a discrete approximation, but their utilization is limited by the unpredictability of their convergence [76]. Recently, also some wavelet basis functions were employed [8]. Our approaches in the problem of the approximation of the phase from the gain will start from the very early results of this field.

7.2 The Bode Gain-Phase Relations

The Bode relations are known as one of the most useful techniques available in network theory, communications, and signal processing [48]. The method is based on the fact that the transform $H(j\omega) = R(\omega) + jI(\omega)$ of a causal function $h(t)$ is uniquely determined in terms of $R(\omega)$ or $I(\omega)$. Proofs based on Cauchy's residue theorem [10, 87] or convolution [49] established that we have

$$R(\omega) = R(\infty) - \frac{1}{\pi}\int_{-\infty}^{\infty}\frac{I(y)}{y-\omega}dy = R(\infty) - \frac{2}{\pi}\int_{0}^{\infty}\frac{yI(y)-\omega I(\omega)}{y^2-\omega^2}dy, \tag{7.2}$$

and

$$I(\omega) = \frac{1}{\pi}\int_{-\infty}^{\infty}\frac{R(y)}{y-\omega}dy = \frac{2\omega}{\pi}\int_{0}^{\infty}\frac{R(y)-R(\omega)}{y^2-\omega^2}dy. \tag{7.3}$$

From the mathematical point of view, the gain-phase relations can be obtained from the Equations (7.2) and (7.3) of Hilbert transform directly by taking logarithms [49]. However, the stable and minimum phase condition ask for supplementary conditions in order to satisfy the right half plane analyticity requirements of the Hilbert transform.

If we assume that $H(s)$ is not only analytic, but has no zeros for $\mathrm{Re}(s) \geq 0$, then $\ln(H(j\omega)) = \alpha(\omega) + j\beta(\omega)$ will also be analytic in the right-hand plane, and the phase $\beta(\omega)$ will be uniquely determined from the gain (in nepers) $\alpha(\omega)$:

$$\beta(\omega) = \frac{2\omega}{\pi}\int_{0}^{\infty}\frac{\alpha(y)-\alpha(\omega)}{y^2-\omega^2}dy. \tag{7.4}$$

A change of variable $u = \ln(y/\omega_c)$ where ω_c is a normalizing frequency, is usually introduced. The results are [10]:

$$\begin{aligned}\beta(\omega_c) &= \frac{2}{\pi}\int_{-\infty}^{\infty}\frac{\alpha(\omega_c e^u)-\alpha(\omega_c)}{e^u-e^{-u}}du = \frac{2}{\pi}\int_{0}^{\infty}\frac{\alpha(\omega_c e^u)-\alpha(\omega_c e^{-u})}{e^u-e^{-u}}du \\ &= \frac{1}{\pi}\int_{-\infty}^{\infty}\left(\frac{d}{du}\alpha(\omega_c e^u)\right)\ln(\coth\frac{|u|}{2})du.\end{aligned} \tag{7.5}$$

Equation (7.5) shows mostly that the phase characteristic is proportional to the derivative of the gain characteristic on a logarithmic frequency scale, weighted by an even function of frequency.

This result is the basis of a method used for a long time in analogue electronics in order to draw the phase characteristics as concatenation of straight lines. Bode straight-line approximations are an extremely useful tool also in the study of system frequency response [23]. These approximations give good insight into the frequency variation of the amplitude and the phase of a system response without the use of computer simulation or complex calculations. Several variants of this technique have been proposed, which differ by the way to approximate the derivative. Three Bode straight-line approximations to the phase of second-order, underdamped systems were subjected to quantitative error analysis in [68], and the results of the analysis indicate a small superiority of the little-known decade-fraction phase approximation technique. From this result it seems that the guess of sampling points and ratio is important for an appropriate straight-line phase approximation.

Providing that the frequency ω_c is given, and after having logarithmized both x and y-axis, we can define the next functions: the log-log gain $A : \mathbb{R} \to \mathbb{R}_+$, and the log-log phase $B : \mathbb{R} \to \mathbb{R}$, by

$$A(u) = \alpha(\omega_c e^u);\ B(u) = \beta(\omega_c e^u), \tag{7.6}$$

and they will be intensively used in the following. Taking into account the new functions, Equation (7.5) becomes:

$$B(0) = \beta(\omega_c) = \frac{2}{\pi}\int_0^\infty \frac{\alpha(\omega_c e^u) - \alpha(\omega_c e^{-u})}{e^u - e^{-u}} du = \frac{2}{\pi}\int_0^\infty \frac{A(u) - A(-u)}{e^u - e^{-u}} du. \tag{7.7}$$

Perhaps it is interesting to note that if the normalizing frequency is changed, then the log-log functions are shifted correspondingly. It follows that their derivatives are also shifted in the same manner. Thus the identities based on linear combinations of log-log functions and their derivatives are still valid even the normalizing frequency is modified, if both axes are shifted correspondingly.

The evaluation of Equations (7.4) and (7.5) is in general complicated. By a change of variable $\omega = -\tan\frac{\delta}{2}$, a set of equations, known as Wiener-Lee transforms can be derived [49]. Thus the resulting attenuation function can be related to the matching phase by a Fourier series expansion:

$$\overline{\alpha} = d_0 + d_1 \cos\delta + \cdots + d_n \cos n\delta + \cdots,$$
$$\overline{\beta} = e_1 \sin\delta + \cdots + d_n \sin n\delta + \cdots,$$

where the coefficients of the corresponding cosine and sine series are opposed each other $d_n = -e_n$.

The numerical assessment is still difficult for this method, it needs a nonlinear variable change, identification of Fourier coefficients, development of a Fourier series. Furthermore,

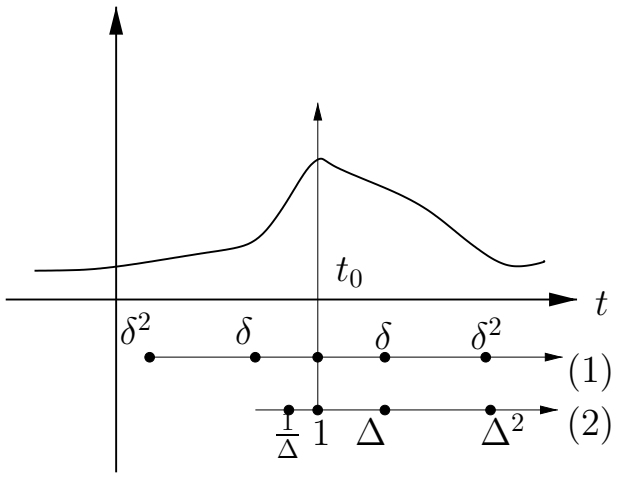


Figure 7.1: The sampling points for wavelet transform (1) and logarithmic sampling (2).

the computation of phase only from a certain number of gain samples might be also problematic.

7.3 Sampling in Logarithmic Domain

Sampling in logarithmic manner was not a very often used way in the transform domain[2], the usual modality was to sample in the uniform fashion. Nevertheless, the recent spread of the wavelet transform and time-scale analyses [56] reconsiders more frequently other types of sampling beside the traditional uniform way.

The fundamental result, known as the sampling theorem, expresses a band-limited function in terms of its sample values at a sequence of equidistant points. From the worth of the weights, it is suggested that the neighbour samples contribute most to the actual value of the signal. It follows that perhaps it is more efficient to concentrate on the samples near the point of interest, and skip the furthest samples. This was an important reason for developing a lot of non-uniform sampling methods [77].

As we are usually more interested in transient details, another type of sampling might be the one which consists of a large number of samples at the beginning of signal burst, and to dedicate less and more less samples as the time passes. Actually something like this is done by wavelet transform, where the main characteristics are localized first by

[2]Recent interests for this logarithmic sampling technique were in pictorial recognition of objects employing affine invariance in the frequency domain [9].

a shift operation, then details are recovered by scaling. However, for the wavelet transform the left-side and right-side samples are equally spaced relative to the scale origin (Figure 7.1). But if the signal has non-symmetric wave shape, and this is the most often case in electronic circuits, perhaps it is better to sample in an asymmetric manner as the logarithmic sampling does. Certainly, this is the case of almost real ordinary signals in electronics and communications, where, except the step-unit changes, they exhibit (sums of) exponential evolution.

In addition, after applying the logarithm, the sampling ratio is easy to evaluate as in any usual uniform sampling. Clearly, the importance of shifting operation is the same for both wavelet transform and logarithmic sampling.

Note that in the case of the Bode gain-phase relation the sampling in log domain is natural for electronic engineers, since for a long time the charts were represented in logarithmic coordinates. Moreover, from practical point of view, a sequence of frequencies in geometrical progression seems easier to generate than in arithmetic progression.

7.4 The Issue Addressed

In the following we are interested to establish relations in order to approximate the phase characteristic from the gain samples, given at equally spaced points on the logarithmic frequency domain. For this reason we consider now the following set of sampling points: $\{x_j | x_j = \omega_c \Delta^j,\ j \in \mathbb{Z}\}$, where $\Delta > 1$.

The following type of formula is suggested by the symmetry of the second term from Equation (7.5):

$$\beta(\omega_c) \approx \sum_{n \in \mathbb{N}} \Gamma_n \left[\alpha(\omega_c \Delta^n) - \alpha(\omega_c \Delta^{-n}) \right]. \tag{7.8}$$

In this dissertation two different types of approaches are presented:

1. In Chapter 8 we show first that the phase is a series of the odd derivatives of the neperian gain. Then by approximating the derivatives by gain differences the first approach is obtained.

2. In Chapter 9 the phase is obtained directly from gain samples, by discretizing in a certain way of the Equation (7.5).

Let denote the gain samples by $A(n) = \alpha(\omega_c \Delta^n)$ and, respectively the phase samples by $B(n) = \beta(\omega_c \Delta^n)$, where $n \in \mathbb{Z}$. It is easy to check that this definition is consistent

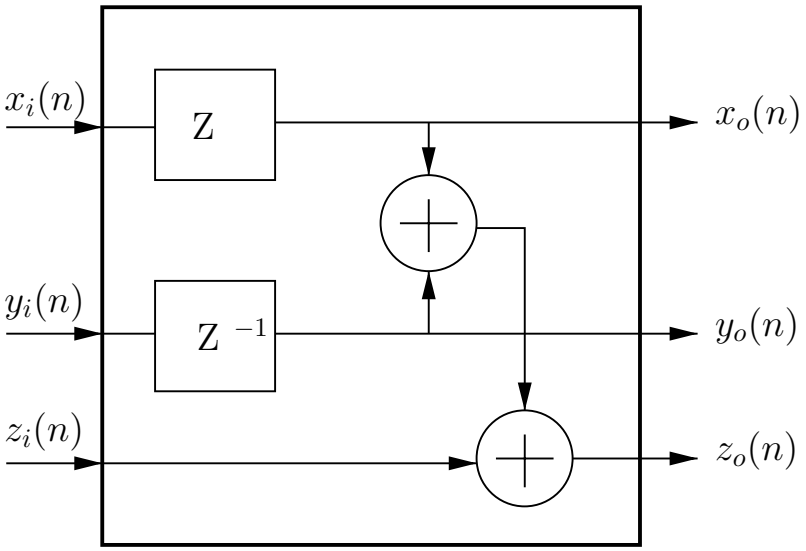


Figure 7.2: The basic cell $x_o(n) = x_i(n+1)$, $y_o(n) = y_i(n-1)$, and $z_o(n) = z_i(n) + C_p[x_i(n) - y_i(n)]$.

with Equation (7.7). Using these notations, the previous formula (7.8) can be written in the following form:

$$B(p) = \sum_{n \in N} C_n[A(p+n) - A(p-n)],$$

where p is given by $\omega_c = \Delta^p$.

Perhaps it is interesting to note that if $p \in \mathbb{Z}$, then the last relationship leads directly to a linear phase FIR filter, where the impulse response has an odd symmetry. For implementation a systolic realization is suggested below, similar with those proposed in [41].

The basic cell is shown in Figure 7.2, where $x_i(n)$, $y_i(n)$, $z_i(n)$ and $x_o(n)$, $y_o(n)$, $z_o(n)$ are the input data, and the output data of the systolic array respectively. The minimum cycle time of the basic cell can be taken as the time required by one real multiplication, and two additions. The proposed array is presented in Figure 7.3 and it consists of P basic cells. The values of their coefficients C_p are from the left to the right equal to $C_1, C_2, \cdots, C_P$.

By passing the input data $A(n)$, which is in fact the gain samples equally spaced in the logarithmic frequency domain, once for each operation cycle into the top left-hand corner of the array, the first output data appears at the bottom right-hand corner of the array after a delay of P cycles. Successive output data $B(n)$ are then obtained once for

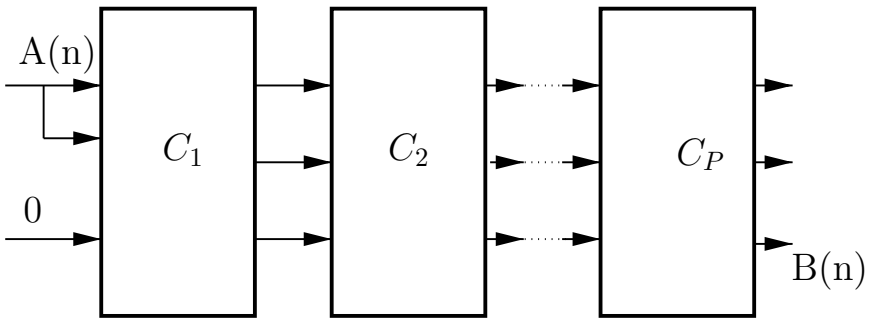


Figure 7.3: The systolic implementation of the approximated phase.

each subsequent operation cycle.

However, only off-line processing can be accomplished due to the existence of the advance operator in the basic cell. This is not a major impediment, without the knowledge of the future gain samples, the phase cannot be approximated for a minimum-phase transfer functions.

Chapter 8

Phase Approximation by Gain Derivatives

The goal of this Chapter is to establish a new relationship for computing the phase of the minimum-phase functions from the gain derivatives, as a first step to approach phase by gain samples. As a beginning, we show that for a given frequency the phase could be obtained from the odd derivatives of the neperian gain, evaluated for this frequency. Then we select a finite number of terms of the main formula and we derive an approximation of phase. We compute this approximations for first-order and second-order systems, and we emphasize the issue of higher-oder derivatives majorants. We show that the approximations derived can be improved by taking into account the Gibbs phenomenon and the Feher kernel. Finally we utilize finite differences in order to substitute the higher derivatives involved in the proposed approaches.

8.1 Main Result

In the following we shall prove the following main result[1]:

Theorem 2. *We have*

$$\int_0^{\infty} \frac{A(u)-A(-u)}{e^u-e^{-u}}du = \sum_{n=0}^{\infty} \frac{(2^{n+2}-1)\pi^{2n+2}|B_{2n+2}|}{(2n+2)!}A^{(2n+1)}(0), \tag{8.1}$$

under the following specified conditions: Assumptions 1,2, where B_n are Bernoulli numbers of order n.

[1]It is easy to see that it is related to the term from the right-hand side of Equation (7.7).

Proof:

Suppose that

Assumption 1. *A has derivatives of all orders in an interval around origin.*

In this case, by expanding the numerator of the integrand into a series, we get:

$$A(u) = \sum_{n=0}^{\infty} \frac{A^{(n)}(0)}{n!} u^n. \tag{8.2}$$

The following series development can be obtained for $u > 0$:

$$\frac{1}{e^u - e^{-u}} = \frac{1}{e^u(1 - e^{-2u})} = \sum_{k=0}^{\infty} e^{-(2k+1)u}.$$

It follows that we have:

$$\int_0^{\infty} \frac{A(u) - A(-u)}{e^u - e^{-u}} du = \int_0^{\infty} \left(\sum_{n=0}^{\infty} \sum_{k=0}^{\infty} \frac{2A^{(2n+1)}(0)}{(2n+1)!} e^{-(2k+1)u} u^{2n+1} \right) du.$$

Furthermore, we consider that

Assumption 2. *The log-log gain $A(u)$ does not grow faster than an exponential ($A(u) \leq e^{\Psi u}, \Psi \leq 1$).*

After changing the order of summation and integration, in the right-hand side of previous formula we obtain the following expression:

$$2 \sum_{n=0}^{\infty} A^{(2n+1)}(0) \sum_{k=0}^{\infty} \frac{1}{(2n+1)!} \int_0^{\infty} e^{-(2k+1)u} u^{2n+1} du. \tag{8.3}$$

The last integral is the Laplace transform of function u^{2n+1}, and from [24] we deduce

$$\int_0^{\infty} e^{-(2k+1)u} u^{2n+1} = \frac{(2n+1)!}{(2k+1)^{2n+2}}.$$

Thus we get the relationship:

$$\int_0^{\infty} \frac{A(u) - A(-u)}{e^u - e^{-u}} du = 2 \sum_{n=0}^{\infty} A^{(2n+1)}(0) \sum_{k=0}^{\infty} \frac{1}{(2k+1)^{2n+2}}.$$

But it is known that [53]

$$\sum_{k=0}^{\infty} \frac{1}{(2k+1)^{2n+2}} = \frac{(2^{2n+2} - 1)\pi^{2n+2}}{2(2n+2)!} |B_{2n+2}|. \tag{8.4}$$

Thus we obtain finally the result claimed in Theorem 2.

Remark 11 .

The first ten terms on the right hand side of Equation (8.4) are descending slowly to 1:

$$\begin{array}{lll} A_2 = 1.23370055; & A_3 = 1.05179974; & A_4 = 1.01467803; \\ A_5 = 1.00452376; & A_6 = 1.00144708; & A_7 = 1.00047155; \\ A_8 = 1.00015518; & A_9 = 1.00005135; & A_{10} = 1.00001704. \end{array}$$

As a consequence sometimes the series from right-hand side of Equation (8.1) can also converge slowly.

Remark 12 . *We can truncate the log-log gain series (Equation 8.2), and obtain an approximation of the phase.*

This can be done only if the next order derivative is bounded over $\mathbb{R}_+$. The accuracy of approximation depends on the corresponding bound and the number of terms of the approximation (Remark 15).

8.2 A Series for Phase

We are going now forward and present the method for computing, without integral relations, of the phase by the gain derivatives. By combining Equation (7.7) with Equation (8.1) we obtain:

$$\begin{aligned} \beta(\omega_c) &= \frac{2}{\pi}\int_{-\infty}^{\infty}\frac{\alpha(\omega_c e^{u}) - \alpha(\omega_c e^{-u})}{e^{u}-e^{-u}}du = \frac{2}{\pi}\int_{-\infty}^{\infty}\frac{A(u)-A(-u)}{e^{u}-e^{-u}}du \\ &= \frac{2}{\pi}\sum_{n=0}^{\infty}\frac{(2^{n+2}-1)\pi^{2n+2}|B_{2n+2}|}{(2n+2)!}A^{(2n+1)}(0) \\ &= \sum_{n=0}^{\infty}\frac{2(2^{n+2}-1)\pi^{2n+1}|B_{2n+2}|}{(2n+2)!}\cdot\left[\frac{d^{(2n+1)}}{du^{2n+1}}\left(\alpha\left(\omega_c e^{u}\right)\right)\right]|_{u=0}. \end{aligned}$$

Thus the phase of a minimum-phase function can be computed with

$$\beta(\omega_c) = \sum_{n=0}^{\infty}\frac{2(2^{n+2}-1)\pi^{2n+1}|B_{2n+2}|}{(2n+2)!}\cdot\left[\frac{d^{(2n+1)}}{du^{2n+1}}\left(\alpha\left(\omega_c e^{u}\right)\right)\right]|_{u=0}. \tag{8.5}$$

In this way we have shown that the phase at a given frequency is the series of the odd derivatives of the neperian gain evaluated at this frequency.

Remark 13 . *Equation (8.5) is valid for every ω_c.*

This results from the fact that during the proof the normalizing frequency was chosen arbitrarily.

Remark 14. *A change in exponent base affects the right-hand side of Equation (8.5).*

Indeed, if $\Delta > 1$ we have the following equalities:

$$\begin{aligned}
\beta(\omega_c) &= \sum_{n=0}^{\infty} \frac{2(2^{n+2}-1)\pi^{2n+1}|B_{2n+2}|}{(2n+2)!} \cdot \left[\frac{d^{(2n+1)}}{du^{2n+1}}\left(\alpha\left(\omega_c e^u\right)\right)\right]|_{u=0} \\
&= \sum_{n=0}^{\infty} \frac{2(2^{n+2}-1)\pi^{2n+1}|B_{2n+2}|}{(2n+2)!} \cdot \left[\frac{d^{(2n+1)}}{du^{2n+1}}\left(\alpha\left(\omega_c \Delta^{u(\ln\Delta)^{-1}}\right)\right)\right]|_{u=0} \\
&= \sum_{n=0}^{\infty} \frac{2(2^{n+2}-1)\pi^{2n+1}|B_{2n+2}|}{(2n+2)!} \cdot \left[\frac{d^{(2n+1)}}{d[(\ln\Delta)^{-1}u]^{2n+1}}\left(\alpha\left(\omega_c \Delta^u\right)\right)\right]|_{u=0} \\
&= \sum_{n=0}^{\infty} \frac{2(2^{n+2}-1)\pi^{2n+1}(\ln\Delta)^{2n+1}|B_{2n+2}|}{(2n+2)!} \cdot \left[\frac{d^{(2n+1)}}{du^{2n+1}}\left(\alpha\left(\omega_c \Delta^u\right)\right)\right]|_{u=0}.
\end{aligned}$$

As a consequence, let denote by α_{dB} the gain expressed in decibels, and consider that the frequencies are measured in decades [10]. Then the phase can be computed with the relationship:

$$\begin{aligned}
\beta(\omega_c) &= \ln 10 \sum_{n=0}^{\infty} \frac{2(2^{n+2}-1)\pi^{2n+1}(\ln 10)^{2n}|B_{2n+2}|}{(2n+2)!} \cdot \left[\frac{d^{(2n+1)}}{du^{2n+1}}\left(\alpha\left(\omega_c 10^u\right)\right)\right]|_{u=0} \\
&= \ln 10 \sum_{n=0}^{\infty} \frac{2(2^{n+2}-1)\pi^{2n+1}(\ln 10)^{2n}|B_{2n+2}|}{(2n+2)!} \cdot \left[\frac{d^{(2n+1)}}{du^{2n+1}}\left(\alpha\left(\omega_c 10^u\right)\right)\right]|_{u=0} \\
&= \frac{\ln 10}{20} \sum_{n=0}^{\infty} \frac{2(2^{n+2}-1)\pi^{2n+1}(\ln 10)^{2n}|B_{2n+2}|}{(2n+2)!} \cdot \left[\frac{d^{(2n+1)}}{du^{2n+1}}\left(\alpha_{dB}\left(\omega_c 10^u\right)\right)\right]|_{u=0} \\
&= \frac{1}{8.68} \sum_{n=0}^{\infty} \frac{2(2^{n+2}-1)\pi^{2n+1}(\ln 10)^{2n}|B_{2n+2}|}{(2n+2)!} \cdot \left[\frac{d^{(2n+1)}}{du^{2n+1}}\left(\alpha_{dB}\left(\omega_c 10^u\right)\right)\right]|_{u=0}.
\end{aligned}$$

Remark 15. *We can use Equation (8.5) to obtain approximations of the phase.*

Taking into account Remark 12, this is possible if the log-log gain $A(u)$ has bounded higher-order derivatives. As an example we can consider the first four terms derived from the main formula:

$$B(0) \equiv \frac{\pi}{2}A'(0) + \frac{\pi^3}{24}A^{(3)}(0) + \frac{\pi^5}{240}A^{(5)}(0) + \frac{17\pi^7}{40320}A^{(7)}(0). \tag{8.6}$$

In this case the residual in the Equation (8.2) is given by

$$A(u) - \sum_{n=0}^{7} \frac{A^{(n)}(0)}{n!}u^n = \frac{A^{(8)}(\xi)}{8!}u^n,\ \xi \in (0,u).$$

An evaluation of the degree this difference affects at the end the phase computation is done by:

$$2\left|A^{(8)}(\xi)\right|\sum_{k=0}^{\infty}\frac{1}{(2k+1)^8},\ \xi\in(0,u).$$

The value is around double of 8-th order derivative (Remark 11). From this evaluation it is suggested that caution must be taken when it is intended to compute the phase with the previous formula. A reason (bounded derivative) has been already pointed out. Another motivation is related with the Remark 11, i.e. the relatively slow decrease of series. In view of these restrictions, a modality to approximate the phase is presented in the following (Section 8.3.4).

8.3 Case Studies

Before applying the main result in the discrete case, first we study the level of approximations when the derivatives are computed analytically. In this way we can distinguish between the sources of the errors in the proposed approach. Because real transfer functions can be factored into real first-order and real second-order transfer functions, these types are probably the most important systems available. Most designs are based upon them [16]. For this motivation, we shall first discuss the proposed approach in these two cases, and only after that we proceed with phase approximations examples.

8.3.1 First-order system

We consider as an example the system:

$$H(s)=\frac{1}{s+\omega_m},$$

with magnitude given by

$$|H(j\omega)|=\frac{1}{\sqrt{\omega^2+\omega_m^2}}.$$

The neperian gain is

$$\alpha(\omega)=-\frac{1}{2}\ln(\omega^2+\omega_m^2),$$

and it becomes the log-log gain after we choose the normalised frequency ω_c:

$$A(u)=\alpha\left(\omega_c e^u\right)=-\frac{1}{2}\ln(\omega_c^2e^{2u}+\omega_m^2)=-\frac{1}{2}\ln(\omega_c^2)-\frac{1}{2}\ln(e^{2u}+\frac{\omega_m^2}{\omega_c^2}).$$

If we introduce the parameter $\gamma = \frac{\omega_m^2}{\omega_c^2}$, then the first derivative of the log-log gain is given by

$$A'(u) = -\frac{1}{2}[\ln(e^{2u} + \gamma)]' = -\frac{e^{2u}}{e^{2u} + \gamma},$$

or equivalently

$$A'(u)(e^{2u} + \gamma) = -e^{2u}.$$

If we apply the Leibnitz rule [7] to the last relationship in order to compute the higher-order derivatives of the product of two functions, and if $n \geq 1$, we obtain:

$$\sum_{k=0}^{n} \binom{n}{k} (e^{2u} + \gamma)^{(k)} A^{(n-k+1)}(u) = -(e^{2u})^{(n)},$$

or

$$(e^{2u} + \gamma)A^{(n+1)}(u) + \sum_{k=1}^{n} \binom{n}{k} 2^k e^{2u} A^{(n-k+1)}(u) = -2^n e^{2u},$$

which gives us the $(n+1)$-derivative of the log-log gain:

$$A^{(n+1)}(u) = -\frac{2^n e^{2u} + \sum_{k=1}^{n} \binom{n}{k} 2^k e^{2u} A^{(n-k+1)}(u)}{e^{2u} + \gamma}.$$

As we want to compute the phase function for a given frequency, we need only the values of odd derivatives in origin. They are:

$$A^{(n+1)}(0) = -\frac{2^n + \sum_{k=1}^{n} \binom{n}{k} 2^k A^{(n-k+1)}(0)}{1 + \gamma} = -\frac{2^n + \sum_{k=1}^{n} \binom{n}{k} 2^k A^{(n-k+1)}(0)}{1 + \frac{\omega_m^2}{\omega_c^2}}. \tag{8.7}$$

Example 24. *Higher-order derivatives majorants*

Perhaps it is better to recognize that the applicability of the Theorem 2 to evaluate an approximation of phase is reduced in those points where the higher-order derivatives of log-log gain have large majorants. This can happen even for simple first-order systems. As an example we consider the case of $H(s) = s+1$, and we compute their log-log derivatives. The results are presented in Figure 8.1. It can be seen that if the order of derivatives is higher than 5, the magnitude of derivative around $\omega_c = 1$ is increasing rapidly. This suggests us to use derivatives until this order, for phase approximations of this type of systems.

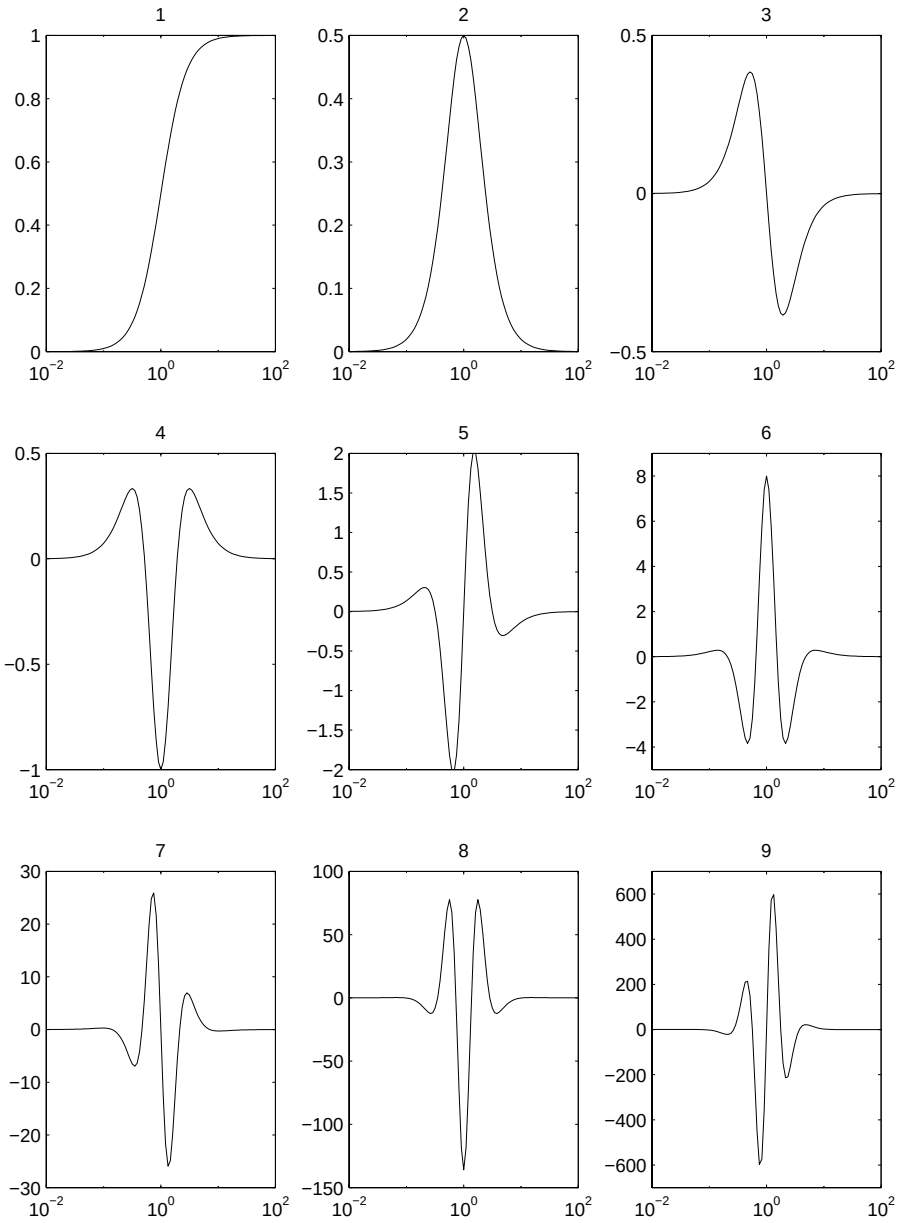


Figure 8.1: Higher-order derivatives for the first-order system presented in Example 24.

8.3.2 Second-order system

The second example considered is the second-order system:

$$H(s) = \frac{1}{s^2 + 2\xi\omega_n s + \omega_n^2}, \ (\xi \in \mathbb{R}).$$

The magnitude is given by

$$|H(j\omega)| = \frac{1}{\sqrt{(\omega^2 - \omega_n^2)^2 + 4\xi^2\omega^2\omega_n^2}},$$

and the neperian gain equals:

$$\alpha(\omega) = -\frac{1}{2}\ln[(\omega^2 - \omega_n^2)^2 + 4\xi^2\omega^2\omega_n^2] = -\frac{1}{2}\ln[\omega^4 + 2(2\xi^2 - 1)\omega^2\omega_n^2 + \omega_n^4].$$

After choosing the normalised frequency ω_c, the log-log gain becomes:

$$A(u) = \alpha\left(\omega_c e^u\right) = -\frac{1}{2}\ln[\omega_c^4 e^{4u} + 2(2\xi^2 - 1)\omega_c^2\omega_n^2 e^{2u} + \omega_n^4] =$$

$$= -\ln\omega_c^2 - \frac{1}{2}\ln[e^{4u} + 2(2\xi^2 - 1)\frac{\omega_n^2}{\omega_c^2}e^{2u} + \frac{\omega_n^4}{\omega_c^4}].$$

If we introduce the parameters $\delta = 2(2\xi^2 - 1)\dfrac{\omega_n^2}{\omega_c^2}$ and $\sigma = \dfrac{\omega_n^4}{\omega_c^4}$, then the first derivative of the log-log gain is given by

$$A'(u) = [\ln(e^{4u} + \delta e^{2u} + \sigma)]' = -\frac{4e^{4u} + 2\delta e^{2u}}{e^{4u} + \delta e^{2u} + \sigma},$$

or equivalently

$$A'(u)(e^{4u} + \delta e^{2u} + \sigma) = -(4e^{4u} + 2\delta e^{2u}).$$

If we apply again the Leibnitz rule to last relationship by n times ($n \geq 1$), then we get the following

$$\sum_{k=0}^{n}\binom{n}{k}(e^{4u} + \delta e^{2u} + \sigma)^{(k)}A^{(n-k+1)}(u) = -(4e^{4u} + 2\delta e^{2u})^{(n)},$$

or

$$(e^{4u} + \delta e^{2u} + \sigma)A^{(n+1)}(u) + \sum_{k=1}^{n}\binom{n}{k}(4^k e^{4u} + \delta 2^k e^{2u})A^{(n-k+1)}(u) = -(4^{n+1}e^{4u} + 2^{n+1}\delta e^{2u}),$$

which gives us the $(n+1)$-derivative of the log-log gain:

$$A^{(n+1)}(u) = -\frac{4^n e^{4u} + 2^{n+1}\delta e^{2u} + \sum_{k=1}^{n}\binom{n}{k}(4^k e^{4u} + \delta 2^k e^{2u})A^{(n-k+1)}(u)}{e^{4u} + \delta e^{2u} + \sigma}.$$

In order to compute the phase function for a given frequency, we need only the values of odd derivatives in origin. They are:

$$\begin{aligned} A^{(n+1)}(0) &= -\frac{4^{n+1} + 2^{n+1}\delta + \sum_{k=1}^{n} \binom{n}{k} (4^k + \delta 2^k) A^{(n-k+1)}(u)}{1 + \delta + \sigma} \\ &= -\frac{4^{n+1} + 2^{n+2} 2(2\xi^2 - 1)\frac{\omega_n^2}{\omega_c^2} + \sum_{k=1}^{n} \binom{n}{k} [4^k + 2^{k+1}(2\xi^2 - 1)\frac{\omega_n^2}{\omega_c^2}] A^{(n-k+1)}(u)}{1 + 2(2\xi^2 - 1)\frac{\omega_n^2}{\omega_c^2} + \frac{\omega_n^4}{\omega_c^4}}. \end{aligned} \tag{8.8}$$

8.3.3 An example

We consider as an example the system:

$$H(s) = \frac{s + \omega_m}{s + \omega_n},$$

with the gain given by:

$$\alpha(\omega) = \frac{1}{2} \ln(\omega^2 + \omega_m^2) - \frac{1}{2} \ln(\omega^2 + \omega_n^2).$$

After introducing the normalised frequency ω_c, the log-log gain is:

$$A(u) = \alpha(\omega_c e^u) = \frac{1}{2} \ln(\omega_c^2 e^{2u} + \omega_m^2) - \frac{1}{2} \ln(\omega_c^2 e^{2u} + \omega_n^2).$$

After applying the achievements from Section 8.3.1, the first, the second and the third derivative of the function $A(u)$ for $u = 0$ are given by:

$$A'(0) = \frac{1}{1 + \left(\frac{\omega_m}{\omega_c}\right)^2} - \frac{1}{1 + \left(\frac{\omega_n}{\omega_c}\right)^2};$$

$$A''(0) = \frac{2 - 2\frac{1}{1 + \left(\frac{\omega_m}{\omega_c}\right)^2}}{1 + \left(\frac{\omega_m}{\omega_c}\right)^2} - \frac{2 - 2\frac{1}{1 + \left(\frac{\omega_n}{\omega_c}\right)^2}}{1 + \left(\frac{\omega_n}{\omega_c}\right)^2} = \frac{2\left(\frac{\omega_m}{\omega_c}\right)^2}{\left[1 + \left(\frac{\omega_m}{\omega_c}\right)^2\right]^2} - \frac{2\left(\frac{\omega_n}{\omega_c}\right)^2}{\left[1 + \left(\frac{\omega_n}{\omega_c}\right)^2\right]^2};$$

$$A^{(3)}(0) = \frac{2^2 - 2^2 \cdot \dfrac{2\left(\dfrac{\omega_m}{\omega_c}\right)^2}{\left[1+\left(\dfrac{\omega_m}{\omega_c}\right)^2\right]^2} - 2^2 \cdot \dfrac{1}{1+\left(\dfrac{\omega_m}{\omega_c}\right)^2}}{1+\dfrac{\omega_m^2}{\omega_c^2}}$$

$$- \frac{2^2 - 2^2 \cdot \dfrac{2\left(\dfrac{\omega_m}{\omega_c}\right)^2}{\left[1+\left(\dfrac{\omega_m}{\omega_c}\right)^2\right]^2} - 2^2 \cdot \dfrac{1}{1+\left(\dfrac{\omega_m}{\omega_c}\right)^2}}{1+\dfrac{\omega_m^2}{\omega_c^2}} =$$

$$= \frac{4\left(\dfrac{\omega_m}{\omega_c}\right)^2\left[1-\left(\dfrac{\omega_m}{\omega_c}\right)^2\right]}{\left[1+\left(\dfrac{\omega_m}{\omega_c}\right)^2\right]^3} - \frac{4\left(\dfrac{\omega_n}{\omega_c}\right)^2\left[1-\left(\dfrac{\omega_n}{\omega_c}\right)^2\right]}{\left[1+\left(\dfrac{\omega_n}{\omega_c}\right)^2\right]^3}.$$

Now we can compute the first and the second phase approximations derived from the main formula. They are:

$$\beta_1(\omega_c) = B_1(u) \approx \frac{\pi}{2}\left[\frac{1}{1+\left(\dfrac{\omega_m}{\omega_c}\right)^2} - \frac{1}{1+\left(\dfrac{\omega_n}{\omega_c}\right)^2}\right],$$

and

$$\beta_2(\omega_c) = B_2(u) \approx \frac{\pi}{2}\left[\frac{1}{1+\left(\dfrac{\omega_m}{\omega_c}\right)^2} - \frac{1}{1+\left(\dfrac{\omega_n}{\omega_c}\right)^2}\right] +$$

$$\frac{\pi^3}{24}\left\{\frac{4\left(\dfrac{\omega_m}{\omega_c}\right)^2\left[1-\left(\dfrac{\omega_m}{\omega_c}\right)^2\right]}{\left[1+\left(\dfrac{\omega_m}{\omega_c}\right)^2\right]^3} - \frac{4\left(\dfrac{\omega_n}{\omega_c}\right)^2\left[1-\left(\dfrac{\omega_n}{\omega_c}\right)^2\right]}{\left[1+\left(\dfrac{\omega_n}{\omega_c}\right)^2\right]^3}\right\},$$

respectively.

A plot of $B(u)$, $B_1(u)$, $B_2(u)$ for $\omega_m = 5$; $\omega_n = 1$ is shown in Figure 8.2. It seems that the first approximation is better than the second one. Indeed we have the following

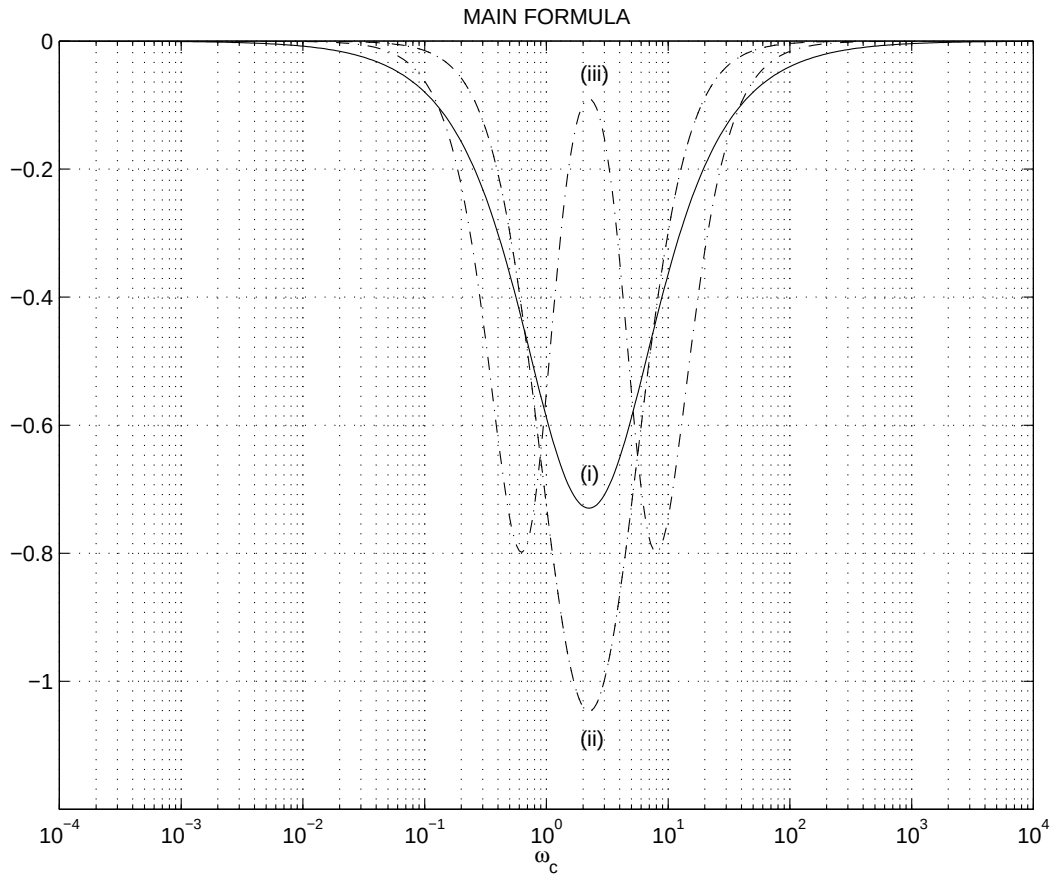


Figure 8.2: Phase (i) versus first (ii) and second (iii) approximations with main formula.

inequalities which describe the behaviour in L^1, L^2, and L^∞ spaces.

$$\max_{|u|\leq 6}(|B_1(u) - B(u)|) < \max_{|u|\leq 6}(|B_2(u) - B(u)|);$$

$$\int_{-6}^{6} |B_1(u) - B(u)|du < \int_{-6}^{6} |B_2(u) - B(u)|du;$$

$$\int_{-6}^{6} |B_1(u) - B(u)|^2 du < \int_{-6}^{6} |B_2(u) - B(u)|^2 du.$$

It is clear now that the magnitude of the third derivative of log-log gain affects the quality of phase approximation. As we shall see in the following, this is not the only one responsible, actually its influence is quite low.

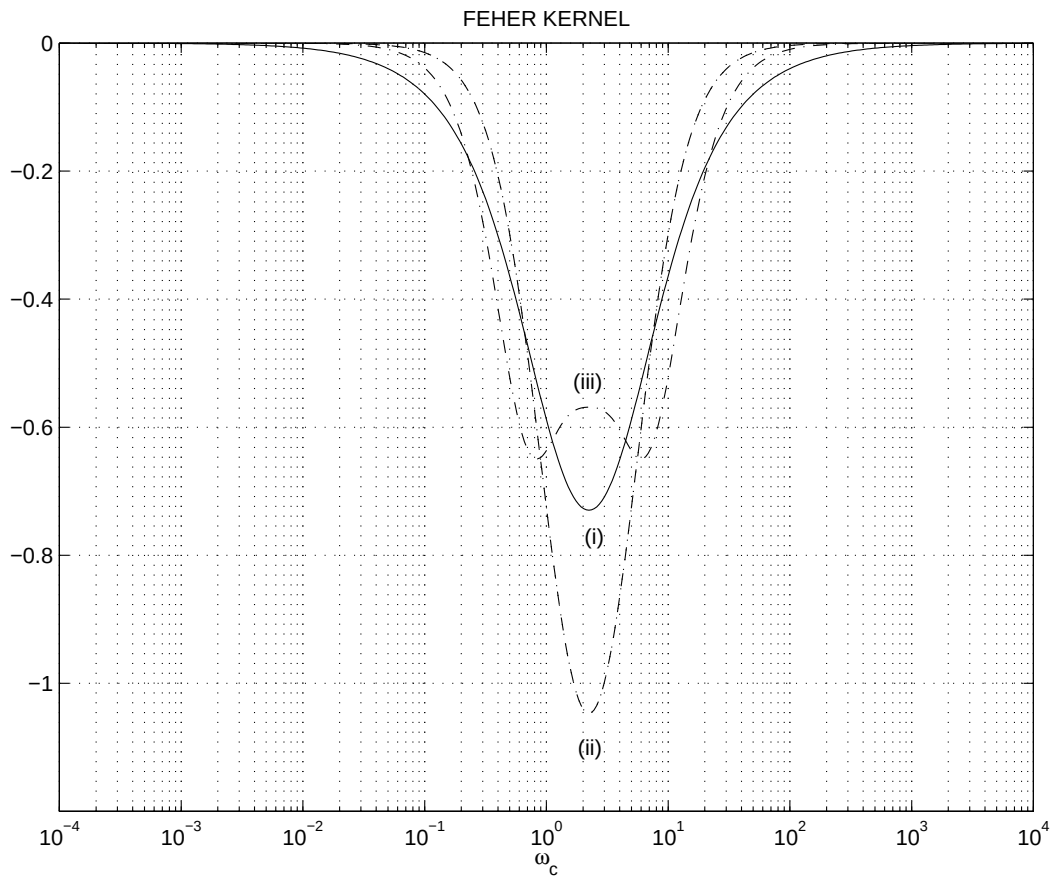


Figure 8.3: Phase (i) versus first (ii) and second (iii) approximations with Feher kernel.

8.3.4 The Feher kernel

If we reconsider the relations (8.2) to (8.4), it can be easily noticed that the above approximations are obtained after the correspondent Taylor series is truncated. Then the error of truncation is propagated through the frequency domain and the Gibbs phenomenon appears [6]. In order to avoid it, we shall use the Feher kernel, i.e., we shall pass the derivatives weights through a triangular window. It gives the following second approximation $\beta_{2F}(\omega_c)$:

$$\beta_{2F}(\omega_c) = B_{2F}(u) \approx \frac{\pi}{2}\left[\frac{1}{1+\left(\frac{\omega_m}{\omega_c}\right)^2} - \frac{1}{1+\left(\frac{\omega_n}{\omega_c}\right)^2}\right] +$$

$$+\frac{1}{2}\cdot\frac{\pi^3}{24}\left\{\frac{4\left(\frac{\omega_m}{\omega_c}\right)^2\left[1-\left(\frac{\omega_m}{\omega_c}\right)^2\right]}{\left[1+\left(\frac{\omega_m}{\omega_c}\right)^2\right]^3}-\frac{4\left(\frac{\omega_n}{\omega_c}\right)^2\left[1-\left(\frac{\omega_n}{\omega_c}\right)^2\right]}{\left[1+\left(\frac{\omega_n}{\omega_c}\right)^2\right]^3}\right\}.$$

A plot of $B(u)$, $B_1(u)$, $B_{2F}(u)$ is shown in Figure 8.3. It is clear now that $B_{2F}(u)$ is a better approximation of $B(u)$ than $B_1(u)$. In fact we have

$$\int_{-6}^{6}|B_{2F}(u)-B(u)|du<\int_{-6}^{6}|B_2(u)-B(u)|du;$$

$$\int_{-6}^{6}|B_{2F}(u)-B(u)|^2du<\int_{-6}^{6}|B_2(u)-B(u)|^2du;$$

$$\max_{|u|\leq 6}(|B_{2F}(u)-B(u)|)<\max_{|u|\leq 6}(|B_1(u)-B(u)|).$$

Finally, we can state that if we need an appropriate approximation in four terms through a triangular window, then Equation (8.6) must be rewritten as

$$B_{4F}(u)\approx\frac{\pi}{2}A'(u)+\frac{3}{4}\cdot\frac{\pi^3}{24}A'''(u)+\frac{1}{2}\cdot\frac{\pi^5}{240}A^{(v)}(u)+\frac{1}{4}\cdot\frac{17\pi^7}{40320}A^{(vii)}(u).$$

However some other smooth windows with exponential decrease could provide better results than the triangular one.

8.3.5 Phase approximation by gain differences

The practical problem we address (Section 7.4) gives us only the gain's samples and thus it is necessary to approximate the higher derivatives with differences. There are several possibilities to do this. Following [54] we use the Stirling numbers of the first kind[2], and in this situation we have:

$$\frac{d^k}{dx^k}y(a_0)=\frac{k!}{h^k}\left(\sum_{j=k}^{n}\frac{S_j^{(k)}}{j!}\overline{\Delta}^j f_0\right),$$

where $\overline{\Delta}^i$ are the finite difference of order i and n is the order of approximation.

Taking into account previous results, we develop only the first three approaches derived from the Feher kernel formula. An one term approximation is given by:

$$B_1(0)=\frac{\pi}{2}A'(0)\approx\frac{\pi}{2h}\overline{\Delta}^1 f_0,$$

[2]The Stirling numbers of the first kind are defined by $S_{n+1}^{(k)}=S_n^{(k-1)}-nS_n^{(k)}$, $S_k^{(k)}=1$, and $S_j^{(0)}=0$ for $j=0$.

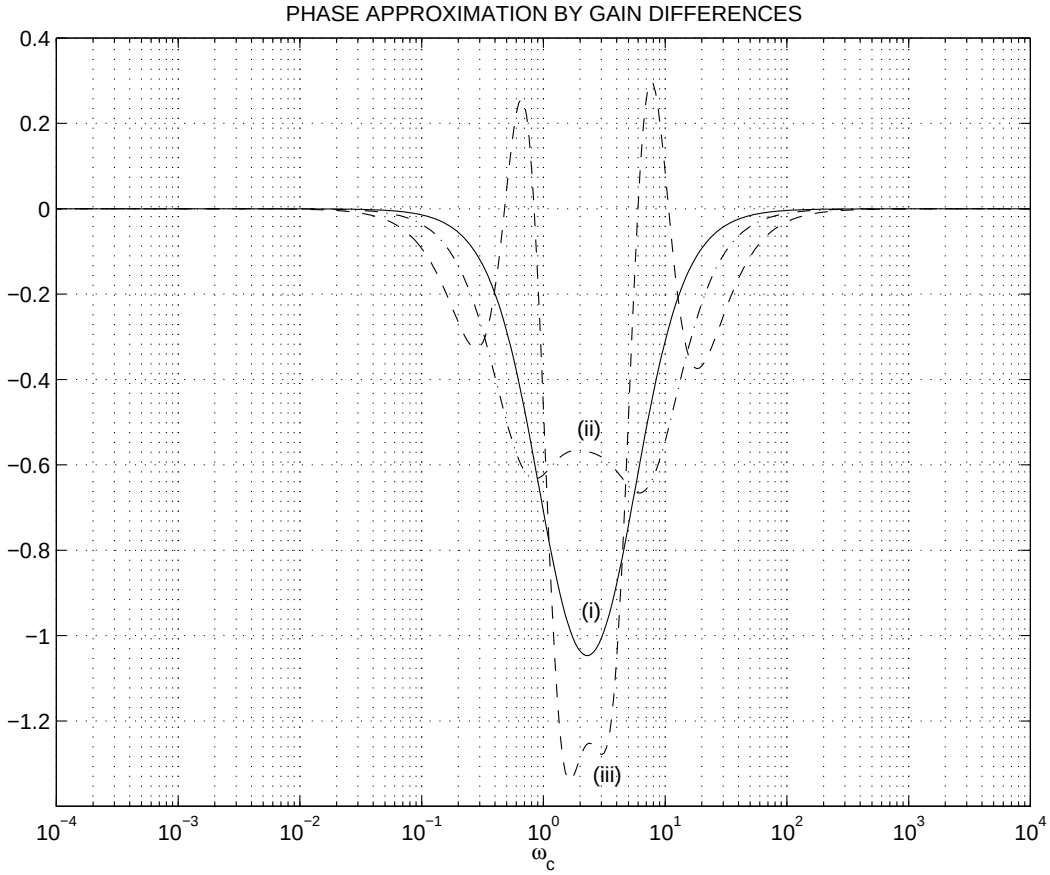


Figure 8.4: First (i), second (ii) and third (iii) approximation of phase for $\Delta = 1.05$.

and this is the well-used old approximation in Bode charts. If we add the third finite differences we get the second approach:

$$B_2(0) = \frac{\pi}{2}A'(0) + \frac{1}{2}\cdot\frac{\pi^3}{24}A'''(0) \approx \frac{\pi}{2h}(\overline{\Delta}^1 f_0 - \frac{1}{2}\overline{\Delta}^2 f_0 + \frac{1}{3}\overline{\Delta}^3 f_0) + \frac{\pi^3}{48h^3}\overline{\Delta}^3 f_0$$

$$= \frac{\pi}{2h}\overline{\Delta}^1 f_0 - \frac{\pi}{4h}\overline{\Delta}^2 f_0 + (\frac{\pi}{6h} + \frac{\pi^3}{48h^3})\overline{\Delta}^3 f_0;$$

Finally we obtain the third approximation:

$$B_3(0) = \frac{\pi}{2}A'(0) + \frac{2}{3}\cdot\frac{\pi^3}{24}A'''(0) + \frac{1}{3}\cdot\frac{\pi^5}{240}A^{(v)}(0)$$

$$\approx \frac{\pi}{2h}(\overline{\Delta}^1 f_0 - \frac{1}{2}\overline{\Delta}^2 f_0 + \frac{1}{3}\overline{\Delta}^3 f_0 - \frac{1}{4}\overline{\Delta}^4 f_0 + \frac{1}{5}\overline{\Delta}^5 f_0)$$

$$+\frac{\pi^3}{36h^3}(\overline{\Delta}^3 f_0 - \frac{3}{2}\overline{\Delta}^4 f_0 + \frac{7}{4}\overline{\Delta}^5 f_0) + \frac{\pi^5}{720h^5}\overline{\Delta}^5 f_0$$

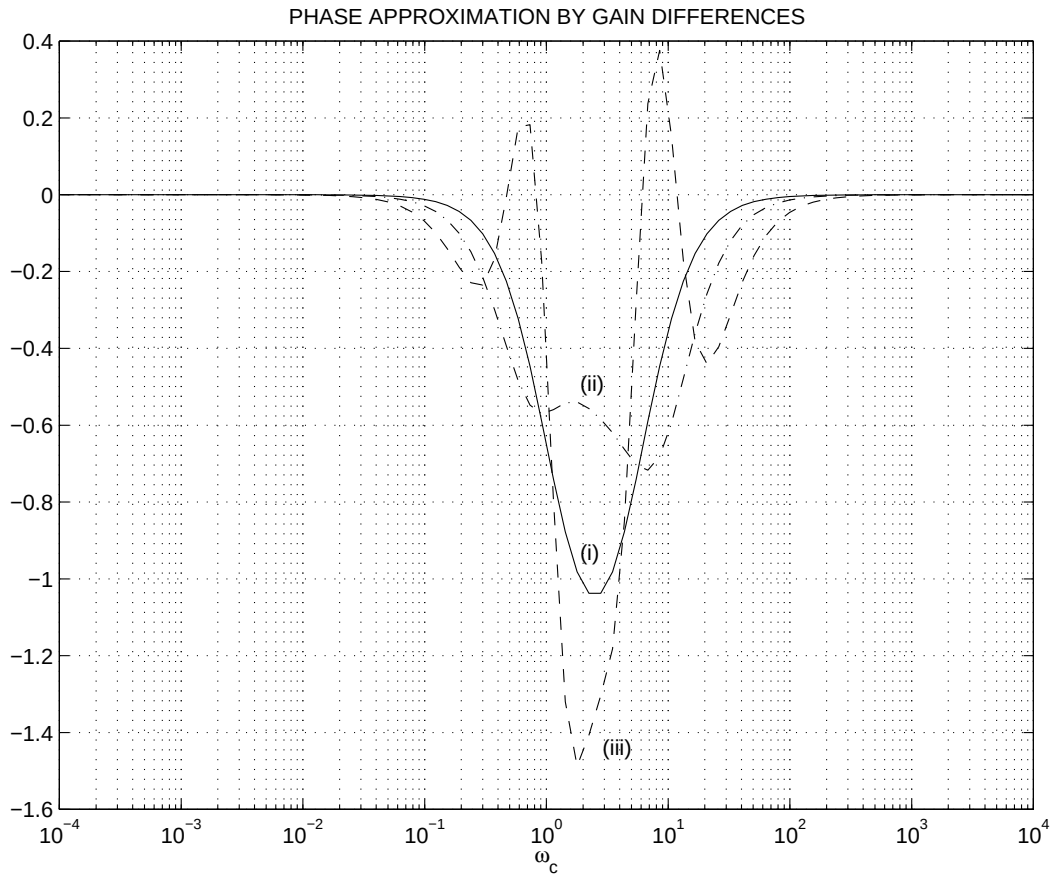


Figure 8.5: First (i), second (ii) and third (iii) approximation of phase for $\Delta = 1.25$.

$$\approx \frac{\pi}{2h}\overline{\Delta}^1 f_0 - \frac{\pi}{4h}\overline{\Delta}^2 f_0 + (\frac{\pi}{6h} + \frac{\pi^3}{36h^3})\overline{\Delta}^3 f_0 - (\frac{\pi}{8h} + \frac{\pi^3}{24h^3})\overline{\Delta}^4 f_0$$
$$+(\frac{\pi}{10h} + \frac{7\pi^3}{144h^3} + \frac{\pi^5}{720h^5})\overline{\Delta}^5 f_0.$$

In order to illustrate the performances of the proposed phase approximations by gain differences, Figures 8.4 and 8.5 present the behaviour of B_1, B_2 and B_3 approaches for $\Delta = 1.05$, and respectively $\Delta = 1.25$. We consider once again the system discussed in Sections 8.3.3 and 8.3.4. Some remarks can be given:

1. The behaviour of these approaches follows the performed analysis outcomes in the previous section.

   Indeed, the third approximations behaves the worst, and this can be explained by the fact that the sixth derivative of log-log gain has a large value near the natural

frequencies of the system.

2. The choice of the abscissas can affect the precision of results.

   It is easy to see that for $\Delta = 1.05$ we obtain better results than when we have $\Delta = 1.25$.

3. A non-symmetry on the waveforms of approximations can be distinguished.

   This is a reminisced of the skew in derivatives values, an additional result of the asymmetry of finite differences.

Chapter 9

Phase Approximation by Gain Samples

In this Chapter we establish new relationships for approximating the phase values from the gain samples, in nepers, equally spaced in the logarithmic frequency domain. First a general approximation formula is proved, then two quadrature formulae are derived using Newton-Cotes and Simpson rules. Finally some numerical examples are shown.

9.1 The Main Formula

In the following we shall prove the following approximation result:

Theorem 3.

$$\beta(\omega) \approx \frac{1}{\pi}[\alpha(\omega\Delta) - \alpha(\omega/\Delta)] + \frac{2\ln\Delta}{\pi}\int_1^k \frac{\alpha(\omega\Delta^z) - \alpha(\omega\Delta^{-z})}{\Delta^z - \Delta^{-z}}dz, \tag{9.1}$$

where $\Delta > 1$ *and* $k \in \mathbb{N}, k \geq 1$ *satisfy certain conditions (Assumptions 3,4,7).*

First we recall three well-known results:

1. **A logarithm inequality**:

   Proposition 1. *For any* $x > 0$*, we have* $\ln(1+x) \leq x$*.*

   If we define the function $f : \mathbb{R}_+ \to \mathbb{R}$, by $f(x) = \ln(1+x) - x$, then the inequality results from $f(0) = 0$ and

   $$f'(x) = \frac{x}{1+x} \leq 0,$$

   for any $x > 0$.

2. **Taylor Theorem** [55]:

   Theorem 4. *Let f be defined on [a,b], let n be a positive integer, suppose $f^{(n-1)}$ is continuous on [a,b], and suppose $f^{(n)}$ exists at every point of (a,b). If R is defined by:*

$$f(b) = f(a) + \sum_{k=1}^{n-1} \frac{(b-a)^k}{k!} f^{(k)}(a) + R,$$

   there is a point $\xi \in (a,b)$ such that

$$R = \frac{(b-a)^n}{n!} f^{(n)}(\xi).$$

   .

3. **The Mean-Value Theorem** [50]:

   Theorem 5. *If F is continuous and p is positive, then*

$$\int_a^b F(x)p(x)dx = F(\xi) \int_a^b p(x)dx, \quad a \leq \xi \leq b.$$

   If F is also positive, it follows that

$$\int_a^b F(x)p(x)dx \leq \sup_{\xi \in [a,b]} F(\xi) \int_a^b p(x)dx.$$

Now we shall give the proof of Theorem 3.

Proof:

We shall also use the decomposition:

$$\begin{aligned}\beta(\omega) = \frac{2\omega}{\pi} \int_0^\infty \frac{\alpha(y)-\alpha(\omega)}{y^2-\omega^2} dy = \frac{2\omega}{\pi} \int_0^{\omega/\Delta^k} \frac{\alpha(y)-\alpha(\omega)}{y^2-\omega^2} dy \\ + \frac{2\omega}{\pi} \int_{\omega/\Delta^k}^{\omega/\Delta} \frac{\alpha(y)-\alpha(\omega)}{y^2-\omega^2} dy + \frac{2\omega}{\pi} \int_{\omega/\Delta}^{\omega\Delta} \frac{\alpha(y)-\alpha(\omega)}{y^2-\omega^2} dy \\ + \frac{2\omega}{\pi} \int_{\omega\Delta}^{\omega\Delta^k} \frac{\alpha(y)-\alpha(\omega)}{y^2-\omega^2} dy + \frac{2\omega}{\pi} \int_{\omega\Delta^k}^{\infty} \frac{\alpha(y)-\alpha(\omega)}{y^2-\omega^2} dy.\end{aligned} \tag{9.2}$$

We first consider the last integral:

$$\int_{\omega\Delta^k}^{\infty} \frac{\alpha(y)-\alpha(\omega)}{y^2-\omega^2} dy.$$

We suppose that:

Assumption 3 . *The variation of the gain is bounded over any interval of the type* $\mathcal{I}_j = [\omega\Delta^j, \omega\Delta^{j+1}]$ *of constant length in the logarithmic scale.*

That is, for any $x, y \in \mathcal{I}_j$ and for every $j > 1$, there exists $M_1 > 0$ such that $|\alpha(x) - \alpha(y)| < M_1$. We suppose also that:

Assumption 4 . M_1 *is independent of* j*.*

This condition is usually respected, since the gain plots are drawn as asymptotes in Bode charts. We further remark that if $\Delta > 1$, $k > 1$ and $j \geq k$, then

$$\left| \int_{\mathcal{I}_j} \frac{\alpha(y) - \alpha(\omega)}{y^2 - \omega^2} dy \right| \leq \frac{(j+1)M_1\Delta^{1-j}}{\omega}. \tag{9.3}$$

This follows from

$$\begin{aligned} \left| \int_{\mathcal{I}_j} \frac{\alpha(y) - \alpha(\omega)}{y^2 - \omega^2} dy \right| &< \sup_{y \in \mathcal{I}_j} \frac{|\alpha(y) - \alpha(\omega)|}{|y - \omega|} \int_{\mathcal{I}_j} \frac{1}{y + \omega} dy \\ &\leq (j+1) \frac{\sup_{x,y \in \mathcal{I}_j} |\alpha(y) - \alpha(x)|}{\inf_{y \in \mathcal{I}_j} |y - \omega|} \int_{\mathcal{I}_j} \frac{1}{y + \omega} dy, \end{aligned}$$

with

$$|y - \omega| \geq \omega(\Delta^j - 1), \ \forall y \in [\Delta^j, \Delta^{j+1}],$$

and the fact that

$$\frac{1}{\Delta^j - 1} \ln(y + \omega) \Big|_{\omega\Delta^j}^{\omega\Delta^{j+1}} = \frac{1}{\Delta^j - 1} \ln\left(\frac{1 + \Delta^{j+1}}{1 + \Delta^j} \right) = \frac{\ln\left(1 + \dfrac{\Delta^{j+1} - \Delta^j}{1 + \Delta^j}\right)}{\Delta^j - 1} \leq \Delta^{1-j},$$

where the inequality follows from $\ln(1 + x) \leq x$ for any $x > 0$ (Proposition 1). From (9.3) we obtain (Appendix 9.4)

$$\left| \int_{\omega\Delta^k}^{\infty} \frac{\alpha(y) - \alpha(\omega)}{y^2 - \omega^2} dy \right| \leq \frac{M_1}{\omega} \sum_{j=k}^{\infty} (j+1)\Delta^{1-j} = \frac{M_1\Delta^{-k+1}}{\omega(1 - \Delta^{-1})^2} [k(1 - \Delta^{-1} + 1)].$$

The following assumption is that:

Assumption 5 . *The variation of the gain is bounded over any interval of the type* $\mathcal{I}_{-j} = [\omega\Delta^{-(j+1)}, \omega\Delta^{-j}]$ *of constant length in the logarithmic scale.*

That is, for any $x, y \in \mathcal{I}_j$ and for every $j > 1$, there exists $M_2 > 0$ such that $|\alpha(x) - \alpha(y)| < M_2$. We suppose also that:

Assumption 6. *M_2 is independent of j.*

In this case, a similar bound can be established for the first integral of (9.2).

We shall evaluate now the sum of the integrals

$$\int_{\omega/\Delta^k}^{\omega/\Delta} \frac{\alpha(y)-\alpha(\omega)}{y^2-\omega^2}dy + \int_{\omega\Delta}^{\omega\Delta^k} \frac{\alpha(y)-\alpha(\omega)}{y^2-\omega^2}dy.$$

Using the substitution $y=\omega\Delta^z$, it can clearly be written as

$$\frac{\ln\Delta}{\omega}\left(\int_{-k}^{-1} \frac{\alpha(\omega\Delta^z)-\alpha(\omega)}{\Delta^z-\Delta^{-z}}dz + \int_{1}^{k} \frac{\alpha(\omega\Delta^z)-\alpha(\omega)}{\Delta^z-\Delta^{-z}}dz\right).$$

Thus the sum of the two integrals is given by

$$\frac{\ln\Delta}{\omega}\left(\int_{1}^{k} \frac{\alpha(\omega\Delta^z)-\alpha(\omega\Delta^{-z})}{\Delta^z-\Delta^{-z}}dz\right).$$

We next evaluate the integral from the middle of (9.2) and we have

$$\begin{aligned}\int_{\omega/\Delta}^{\omega\Delta} \frac{\alpha(y)-\alpha(\omega)}{y^2-\omega^2}dy &= \frac{1}{2\omega}\int_{\omega/\Delta}^{\omega\Delta}\left(\frac{\alpha(y)-\alpha(\omega)}{y-\omega}-\alpha'(y)\right)dy\\ &-\frac{1}{2\omega}\int_{\omega/\Delta}^{\omega\Delta}\frac{\alpha(y)-\alpha(\omega)}{y+\omega}dy+\frac{1}{2\omega}\int_{\omega/\Delta}^{\omega\Delta}\alpha'(y)dy.\end{aligned}$$

Assume that the following assumption holds.

Assumption 7. *The gain second derivative is continuous in $[\omega/\delta,\omega\delta]$, there exists $M_3>0$ and $M_4>0$ such that $\alpha''(x)<M_3$, $\alpha'(x)<M_4$, $\forall x\in[\omega/\delta,\omega\delta]$.*

In this case using the Taylor formula (Theorem 4) and the Mean-Value Theorem (Theorem 5) we have:

$$\begin{aligned}\int_{\omega/\Delta}^{\omega\Delta}\left(\frac{\alpha(y)-\alpha(\omega)}{y-\omega}-\alpha'(y)\right)dy &= \int_{\omega/\Delta}^{\omega\Delta}\left(\frac{\alpha(y)-\alpha(\omega)-\alpha'(y)(y-\omega)}{y-\omega}\right)dy\\ &\le \frac{M_3}{2}\int_{\omega/\Delta}^{\omega\Delta}|\omega-y|dy = \frac{M_3\omega^2(\Delta-1)^2(1+\Delta^{-2})}{4\Delta^2},\end{aligned} \tag{9.4}$$

and

$$\begin{aligned}\left|\int_{\omega/\Delta}^{\omega\Delta}\frac{\alpha(y)-\alpha(\omega)}{y+\omega}\right|dy &= \left|\int_{\omega/\Delta}^{\omega\Delta}\frac{\alpha(y)-\alpha(\omega)}{y-\omega}\cdot\frac{y-\omega}{y+\omega}\right|dy \le M_4\left|\int_{\omega/\Delta}^{\omega\Delta}\frac{y-\omega}{y+\omega}dy\right|\\ &= M_4\omega\left[\frac{(\Delta-1)^2}{\Delta}+\ln\frac{4\Delta}{(1+\Delta)^2}\right].\end{aligned} \tag{9.5}$$

Hence for suitable Δ we obtain:

$$\int_{\omega/\Delta}^{\omega\Delta} \frac{\alpha(y)-\alpha(\omega)}{y^2-\omega^2}dy \approx \frac{1}{2\omega}[\alpha(\omega\Delta)-\alpha(\omega/\Delta)].$$

Now, for any $M_i, i=\overline{1,4}$, given any $\epsilon>0$, there exists:

1. A frequency ratio $\Delta=\Delta(\omega,\epsilon)$ such that

$$\frac{M_3\omega^2(\Delta-1)^2(1+\Delta^{-2})}{4\Delta^2}<\frac{\epsilon}{4},\ M_4\omega\left[\frac{(\Delta-1)^2}{\Delta}+\ln\frac{4\Delta}{(1+\Delta)^2}\right]<\frac{\epsilon}{4};$$

2. An integer $k=k(\Delta,\omega,\epsilon)$ which verifies

$$\frac{M_1\Delta^{-k+1}}{\omega(1-\Delta^{-1})^2}[k(1-\Delta^{-1}+1)]<\frac{\epsilon}{4},\quad \frac{M_2\Delta^{-k+1}}{\omega(1-\Delta^{-1})^2}[k(1-\Delta^{-1}+1)]<\frac{\epsilon}{4}.$$

From these conditions we conclude that

$$\left|\beta(\omega)-\left\{\frac{1}{\pi}[\alpha(\omega\Delta)-\alpha(\omega/\Delta)]+\frac{2\ln\Delta}{\pi}\int_1^k\frac{\alpha(\omega\Delta^z)-\alpha(\omega\Delta^{-z})}{\Delta^z-\Delta^{-z}}dz\right\}\right|<\epsilon,$$

which is the main formula (9.1).

9.2 Quadrature Approximations

For numerical computations support, it is of interest to develop a quadrature formula where the phase function will be determined by the gain samples. Such a formula was derived in [63, 64] and viewed in Section 8.3.4, where it was shown that the phase is the series of the odd derivatives of the neperian gain. Unfortunately the convergence of the series is slower. Also the numerical evaluation of the higher derivatives are likely to have sizeable errors [54].

The condition of equally spaced abscissas leads to one of the Newton-Cotes or Simpson's quadrature formulae [54]. At the beginning we selected the trapezoidal formula and we obtained the first approximation $\beta_T(\omega)$ of the phase $\beta(\omega)$:

$$\begin{aligned}\beta_T(\omega)=\frac{1}{\pi}[\alpha(\omega\Delta)-\alpha(\omega/\Delta)]+\frac{\ln\Delta}{\pi}[\frac{\alpha(\omega\Delta)-\alpha(\omega/\Delta)}{\Delta-\Delta^{-1}}+\\ +2\sum_{p=2}^{k-1}\frac{\alpha(\omega\Delta^p)-\alpha(\omega\Delta^{-p})}{\Delta^p-\Delta^{-p}}+\frac{\alpha(\omega\Delta^k)-\alpha(\omega\Delta^{-k})}{\Delta^k-\Delta^{-k}}],\end{aligned}\tag{9.6}$$

or

$$\beta_T(\omega) = \sum_{p\in\mathbb{Z}} T_p \alpha(\omega\Delta^p),$$

$$T_p = T_{-p} = \begin{cases} \dfrac{1}{\pi}\left(1 + \dfrac{\ln\Delta}{\Delta - 1/\Delta}\right), & p = 1; \\ \dfrac{2\ln\Delta}{\pi\left(\Delta^p - \Delta^{-p}\right)}, & p = 2, 3, \ldots, k-1; \\ \dfrac{\ln\Delta}{\pi\left(\Delta^p - \Delta^{-p}\right)}, & p = k; \\ 0, & \text{otherwise.} \end{cases}$$

The parabolic rule, for $k = 2m + 1$ gives the second proposed quadrature approach $\beta_S(\omega)$, and we have:

$$\begin{aligned} \beta_S(\omega) = \frac{1}{\pi}[\alpha(\omega\Delta) - \alpha(\omega/\Delta)] + \frac{2\ln\Delta}{3\pi}[\frac{\alpha(\omega\Delta) - \alpha(\omega\Delta^{-1})}{\Delta - \Delta^{-1}} + \\ +4\frac{\alpha(\omega\Delta^2) - \alpha(\omega\Delta^{-2})}{\Delta^2 - \Delta^{-2}} + 2\frac{\alpha(\omega\Delta^3) - \alpha(\omega\Delta^{-3})}{\Delta^3 - \Delta^{-3}} + \\ +4\frac{\alpha(\omega\Delta^4) - \alpha(\omega\Delta^{-4})}{\Delta^4 - \Delta^{-4}} + \ldots + 4\frac{\alpha(\omega\Delta^{k-1}) - \alpha(\omega\Delta^{1-k})}{\Delta^{k-1} - \Delta^{1-k}} \\ +\frac{\alpha(\omega\Delta^k) - \alpha(\omega\Delta^{-k})}{\Delta^k - \Delta^{-k}}], \end{aligned} \tag{9.7}$$

or

$$\beta_S(\omega) = \sum_{p\in\mathbb{Z}} S_p \alpha(\omega\Delta^p),$$

$$S_p = S_{-p} = \begin{cases} \dfrac{1}{\pi}\left(1 + \dfrac{2/3\ln\Delta}{\Delta - 1/\Delta}\right), & p = 1; \\ \dfrac{8\ln\Delta}{3\pi\left(\Delta^p - \Delta^{-p}\right)}, & p = \pm 2, \pm 4, \ldots, \pm 2m; \\ \dfrac{4\ln\Delta}{3\pi\left(\Delta^p - \Delta^{-p}\right)}, & p = \pm 3, \ldots, \pm(2m-1); \\ \dfrac{2\ln\Delta}{3\pi\left(\Delta^p - \Delta^{-p}\right)}, & p = \pm(2m+1); \\ 0, & \text{otherwise.} \end{cases}$$

In these two cases, in addition to the error of the main result (9.1), we have the error of approximation of quadrature formula, which is given by $E_T = -((k-1)\alpha''(\eta))/12$ for the Newton-Cotes rule, and respectively $E_S = -((k-1)\alpha^{iv}(\xi))/180$ for Simpson formula, where $\eta, \xi \in (1, k)$ [54].

Some remarks are necessary:

1. First, it seems from the previous estimations that the two quadrature formulae proposed are comparable in performance according to the number of samples. However, the multiplying constants and the level of derivatives differ by much. It will

be shown in the next section that there is not a prevalent choice; sometimes the parabolic rule outperforms the trapezoidal formula, sometimes not.

2. Secondly, the formulae developed above are both easy to implement. Also, their coefficients are anti-symmetric which leads straightaway to the linear phase FIR filters structures (Section 7.4).

3. Finally, we remark that we did not assume special conditions for the frequency ω, and consequently the phase can be approximated for every frequency within the conditions mentioned in Section 2.

9.3 Numerical Examples

To illustrate the performance of the approaches described here, we used the transfer functions of the gain-phase plots presented in Bode's book [10]:

$$H(s) = \frac{1|}{|s} + \frac{1|}{|sK^2} + \frac{1|}{|s/H} + \frac{1|}{|1}$$

Their phase is almost constant for $\omega < 0.01$ and $\omega > 10$, consequently the interval of interest in our experiments was $\omega \in [0.01, 10]$.

9.3.1 Phase versus approximated phase

The quality of the approximated phase for both Newton-Cotes and Simpson approaches is presented first. In Figure 9.1 for the Newton-Cotes approach and respectively in Figure 9.2 for the Simpson approach, the three different phase approximations are plotted together with the actual phase for the case of $\Delta = \sqrt{2}$, $k = 5, 9, 17$ and $K = 1/2$, $H = 1/2$. Note that the phase increases from $-\pi/4$ to $\pi/3$ and then descends almost to $-\pi/2$ during four consecutive gain samples, which makes it difficult to approximate. We can conclude also that there are no major differences between the results of the different proposed methods.

9.3.2 Sampling parameters influence

In the following we are interesting in the effect of distance between the gain samples and of the number of samples. For comparison we used the well known three norms L^1, L^2, L^∞, where for simulations purposes the integral was replaced by numerical integration with a trapezoidal rule in the case of L^1, L^2, and with a min-max criterion over the available results for L^∞. Nine frequency ratio step-sizes were selected: $\Delta = 2^{m(q)}$, where $m(q) =$

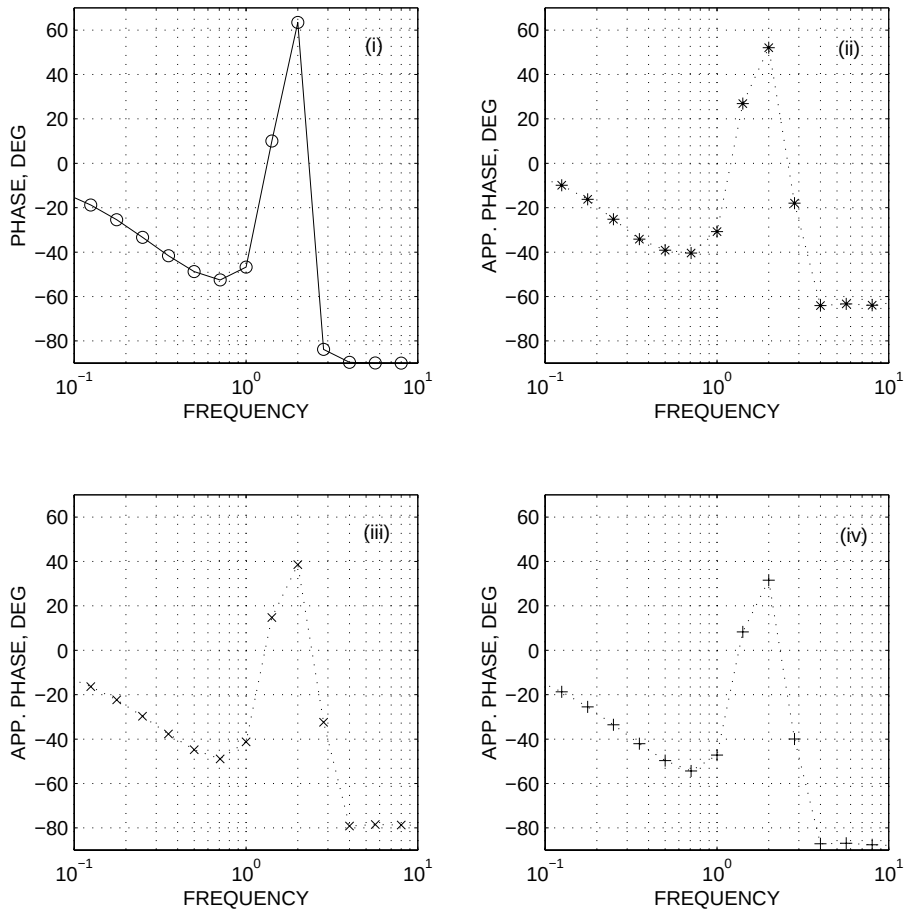


Figure 9.1: The phase of $H(s) = \frac{1|}{|s} + \frac{1|}{|s/4} + \frac{1|}{|2s} + \frac{1|}{|1}$ (i), and the approximated phase by Newton-Cotes approach, for $k = 5$ (ii), $k = 9$ (iii) and $k = 17$ (iv).

2^{2-q}, and $q = 1, 2, \ldots, 9$. The number of samples was chosen to increase exponential $k = 2^p + 1, p = 2, 3, \ldots 8$. In this way we can compare the two approaches, as in the case of the Simpson approach k is odd. The number of samples, the frequency ratio step-sizes, and their corresponding performances for trapezoidal and parabolic approaches, are shown in Tables 9.1, 9.2, 9.3, and respectively Tables 9.4, 9.5, 9.6, for the transfer functions parameters equal to $K = 2$, $H = 1/2$ and L^1, L^2, L^∞ norms.

Similar results were also obtained for the other transfer functions, at all various parameters K and H given in Bode's book, i.e., $K = 1/2$, $1/\sqrt{2}$, 1, $\sqrt{2}$, 2, and $H = 1/2$, $1/\sqrt{2}$, 1, $\sqrt{2}$, 2. The behavior of all simulations is the same:

1. For some values of $\Delta > 1$, one of the Simpson or the Newton-Cotes approaches

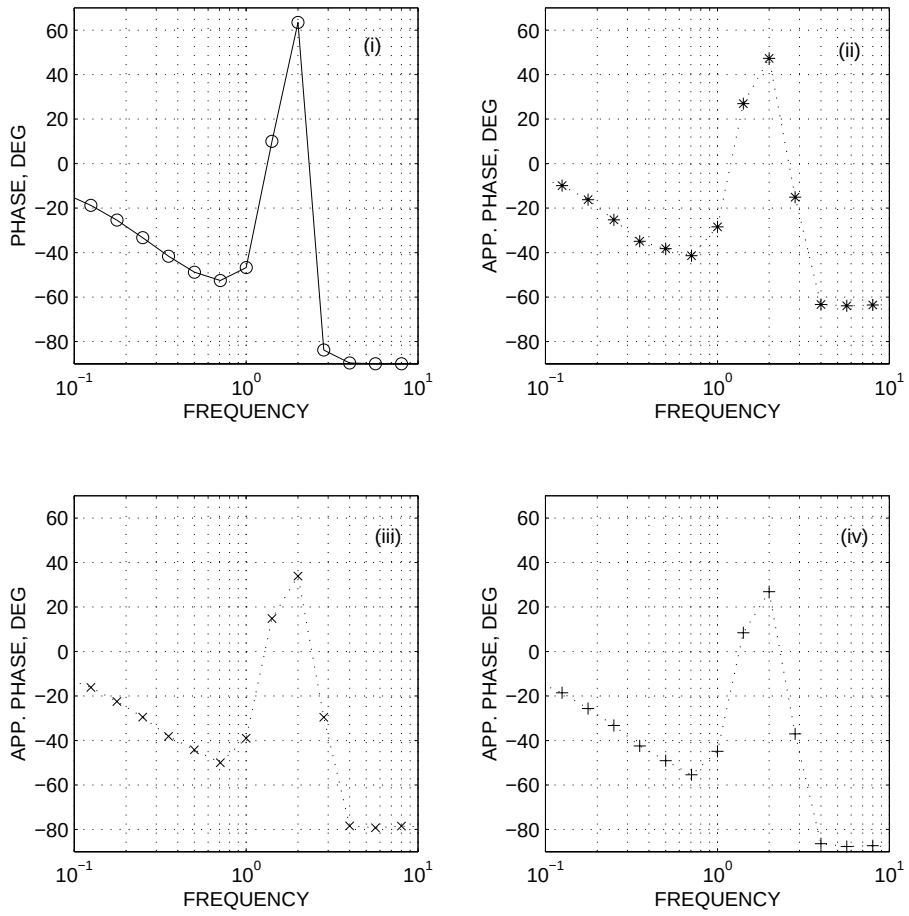


Figure 9.2: The phase of $H(s) = \frac{1|}{|s} + \frac{1|}{|s/4} + \frac{1|}{|2s} + \frac{1|}{|1}$ (i), and the approximated phase by Simpson approach, for $k = 5$ (ii), $k = 9$ (iii) and $k = 17$ (iv).

outperforms the other in L^1, L^2, or in L^∞, but there is not a universal alternative.

2. As $\Delta \to 1$, both approaches behave quite similarly.

3. In both cases, if the number of samples increases, all types of the error decreases. After they attain a certain level, they become stationary.

This can be explained by the fact that if the number of samples passes a given value, the error of quadrature formulas (9.6) and (9.7) is much greater than the corresponding remaining error in (9.1).

If we increase more and more the number of the samples, the approaches could blow-up. This fact can be also justified. Once since k increases, we have to compute the phase

q	k=5	9	17	33	65	129	257
1	0.365	0.368	0.368	0.368	0.368	0.368	0.368
2	0.165	0.103	0.106	0.106	0.106	0.106	0.106
3	0.340	0.145	0.046	0.037	0.037	0.037	0.037
4	0.572	0.368	0.136	0.018	0.006	0.006	0.006
5	0.746	0.603	0.389	0.145	0.016	0.001	0.001
6	0.858	0.767	0.619	0.401	0.149	0.016	0.001
7	0.923	0.870	0.778	0.628	0.407	0.152	0.016
8	0.957	0.929	0.877	0.783	0.632	0.409	0.153
9	0.975	0.961	0.933	0.879	0.786	0.634	0.411

Table 9.1: Newton-Cotes Approach L^1

q	k=5	9	17	33	65	129	257
1	0.4743	0.477	0.477	0.477	0.477	0.477	0.477
2	0.228	0.222	0.224	0.224	0.224	0.224	0.224
3	0.412	0.172	0.111	0.110	0.110	0.110	0.110
4	0.714	0.450	0.156	0.025	0.021	0.021	0.021
5	0.926	0.752	0.478	0.166	0.016	0.005	0.005
6	1.050	0.949	0.773	0.492	0.172	0.016	0.001
7	1.119	1.064	0.962	0.783	0.499	0.175	0.017
8	1.155	1.126	1.070	0.968	0.789	0.503	0.176
9	1.173	1.159	1.129	1.073	0.971	0.791	0.505

Table 9.2: Newton-Cotes Approach L^2

for larger frequencies; if $\omega \gg 1$, then it is required for Δ to tend to one more rapidly, otherwise the error of (9.4) becomes very important. If we keep Δ constant, for some k this will give increasing errors.

We can conclude that the two approaches we develop in this Section can find an approximation of the phase assuming that the unknown minimum-phase system has no finite zeros on the imaginary axis. This could be a satisfactory result when we have only finite gain samples available and no further information.

q	k=5	9	17	33	65	129	257
1	0.921	0.926	0.926	0.926	0.926	0.926	0.926
2	0.603	0.672	0.680	0.680	0.680	0.680	0.680
3	0.619	0.392	0.390	0.401	0.401	0.401	0.401
4	1.041	0.692	0.261	0.115	0.102	0.102	0.102
5	1.298	1.090	0.730	0.279	0.042	0.032	0.032
6	1.433	1.325	1.115	0.751	0.289	0.024	0.005
7	1.502	1.447	1.338	1.128	0.761	0.293	0.024
8	1.536	1.509	1.453	1.345	1.134	0.766	0.296
9	1.554	1.540	1.512	1.458	1.349	1.138	0.769

Table 9.3: Newton-Cotes Approach L^∞

q	k=5	9	17	33	65	129	257
1	0.330	0.333	0.333	0.333	0.333	0.333	0.333
2	0.170	0.113	0.114	0.114	0.114	0.114	0.114
3	0.343	0.148	0.150	0.050	0.042	0.042	0.042
4	0.572	0.369	0.137	0.020	0.009	0.009	0.009
5	0.746	0.603	0.390	0.145	0.016	0.002	0.002
6	0.859	0.767	0.620	0.401	0.150	0.016	0.001
7	0.923	0.870	0.778	0.628	0.407	0.152	0.016
8	0.957	0.929	0.877	0.783	0.632	0.409	0.153
9	0.975	0.961	0.933	0.880	0.786	0.634	0.411

Table 9.4: Simpson Approach L^1

9.4 Appendix

If $\Delta > 1$ and $k > 0$, then we have

$$\sum_{j=k}^{\infty}(j+1)\Delta^{1-j} = \frac{\Delta^{-k+1}}{(1-\Delta^{-1})^2}\left[k(1-\Delta^{-1})+1\right].$$

Indeed,

q	k=5	9	17	33	65	129	257
1	0.454	0.456	0.456	0.456	0.456	0.456	0.456
2	0.235	0.229	0.231	0.231	0.231	0.231	0.231
3	0.413	0.178	0.121	0.120	0.120	0.120	0.120
4	0.713	0.450	0.157	0.030	0.026	0.026	0.026
5	0.926	0.752	0.478	0.166	0.017	0.006	0.006
6	1.050	0.950	0.772	0.492	0.172	0.016	0.001
7	1.119	1.063	0.962	0.783	0.499	0.175	0.017
8	1.155	1.126	1.070	0.968	0.789	0.503	0.177
9	1.173	1.158	1.129	1.074	0.971	0.791	0.505

Table 9.5: Simpson Approach L^2

q	k=5	9	17	33	65	129	257
1	0.930	0.927	0.927	0.927	0.927	0.927	0.927
2	0.625	0.692	0.700	0.700	0.700	0.700	0.700
3	0.618	0.431	0.417	0.428	0.428	0.428	0.428
4	1.041	0.692	0.261	0.123	0.113	0.113	0.113
5	1.298	1.090	0.730	0.279	0.049	0.041	0.041
6	1.433	1.325	1.115	0.751	0.289	0.024	0.007
7	1.502	1.447	1.339	1.128	0.761	0.293	0.025
8	1.537	1.509	1.454	1.345	1.134	0.766	0.296
9	1.554	1.540	1.512	1.457	1.348	1.138	0.768

Table 9.6: Simpson Approach L^∞

$$\sum_{j=k}^{\infty}(j+1)\Delta^{1-j} = \Delta^2\left[(k+1)\Delta^{-(k+1)} + (k+2)\Delta^{-(k+2)} + (k+3)\Delta^{-(k+3)} + (k+4)\Delta^{-(k+4)} + \cdots\right]$$

$$= \Delta^2[k\Delta^{-(k+1)} + k\Delta^{-(k+2)} + k\Delta^{-(k+3)} + k\Delta^{-(k+4)} + \cdots$$

$$+\Delta^{-(k+1)} + 2\Delta^{-(k+2)} + 3\Delta^{-(k+3)} + 4\Delta^{-(k+4)} + \cdots]$$

$$= \Delta^2\left[\frac{k\Delta^{-(k+1)}}{1-\Delta^{-1}} + \frac{\Delta^{-(k+1)}}{1-\Delta^{-1}} + \frac{\Delta^{-(k+2)}}{1-\Delta^{-1}} + \frac{\Delta^{-(k+3)}}{1-\Delta^{-1}} + \cdots\right] = \frac{\Delta^{-k+1}}{(1-\Delta^{-1})^2}\left[k(1-\Delta^{-1})+1\right].$$

Chapter 10

Conclusions

This thesis has introduced several different approaches to cost function adaptation for data echo cancellation and phase approximation by logarithmic gain, offering a broad picture of their performances and benefits. The contributions of the dissertation have both theoretical and practical nature.

The goal of this thesis has been twofold. On the hand, we have aimed at creating new tools[1] for adaptive filtering applications in data communications. On the other hand, the aim has been to recall and further develop the old fashioned and challenging field of phase approximation by gain, which in our opinion is still attractive for several types of applications in communications and industry.

In the newly introduced concept of cost function adaptation, we have derived two main classes of approaches: nonrecursive and recursive algorithms. Nonrecursive adaptation of cost function can be distinguished by the fact that the error exponent is updated every iteration using only direct relationships between an approximation of the error and the power of the cost function. Alternatively, in recursive cost function adaptation, we do not need anymore to estimate the actual error of the system, the updated error exponent is computed recursively.

As in other cases of signal processing, the nonrecursive and recursive families have common and different features. We have found that they behaves similarly in the sense that the best performances from speed of convergence and steady-state point of view are closed. Also the computation requirements do not differ dramatically from one to another. However, the analytical analysis and parameter selection seems for us more

[1] As Prof. Simon Haykin states in [35]: "There is no unique solution to adaptive filtering problem. Rather, we have a "kit of tools" represented by a variety of recursive algorithms, each of which offers desirable features of its own".

straightforward and easier to design and to foresee for the nonrecursive case. On the other hand the necessary memory for a given performance remain a potential important benefit of recursive techniques.

The convergence and steady-state analysis carried out suggests that an important contribution to the stability of cost function adaptation algorithms is given by the length of the period when the higher-order algorithms act. Perhaps a more detailed analysis, similar with [78] can reveal more insights of these aspects.

We are optimistic that cost function adaptation concept (or variable error exponent technique) can be applied in more situations and applications. Actually we feel that whenever there is a different behaviour for different powers of certain error cost functions, a similar strategy can extract their benefits in a new and efficient tool.

In the second part of the work we have been concentrated in obtaining new analytical and numerical formulae for phase. The novelty of these approaches is that we took into account as inputs the gain samples, equally distanced in logarithmic domain. We did not discuss extensively the logarithmic sampling, but we are expecting that this sampling method can be a more fruitful domain that it has been considered until now.

Phase obtained by a series of gain derivatives provided us a very interesting analytical avenue toward phase approximation. In certain situations we have improved previous results, but unfortunately after a rank the resulting numerical formulae are sensitive to the higher derivatives or higher-order differences. A future direction of study will be to reduce their effect, and some nonlinear a priori filtering techniques like those described in [6, 75] might be helpful.

Certainly the phase approximation by gain samples as it is described in Chapter 9 is a better method to approximate the phase. First of all, the bounds of the approximation are clear. For a given level of accuracy we can find the sampling period and after that we can search the number of terms for the desired approximation. With this formula we can get good approximations, even when the frequencies are not so closed each other.

As a final conclusion we can state that this thesis succeeded to get ready several solutions to relative difficult problems, and it can provide also enough possible topics for future research.

Bibliography

[1] S. Ahmed and J. Cruz, "A new system identification method for fast echo canceller initialization," *IEEE Trans. Commun.*, vol. 44, pp. 137–142, Feb. 1996.

[2] B. D. O. Anderson, "About Hilbert transform." Personal e-mail, Australian National University, 26 August 1998.

[3] B. D. O. Anderson and M. Green, "Hilbert transform and gain/phase error bounds for rational functions," *IEEE Transactions on Circuits and Systems*, vol. 35, pp. 528–535, May 1988.

[4] B. D. O. Anderson and J. B. Moore, *Optimal filtering.* Englewood Cliffs: Prentice Hall, 1979.

[5] O. Arikan, A. E. Çetin, and E. Erzin, "Adaptive filtering for non-gaussian stable processes," *IEEE Signal Processing Letters*, vol. 1, pp. 163–165, Nov. 1994.

[6] J. Astola and P. Kuosmanen, *Fundamentals of Nonlinear Digital Filtering.* Boca Raton, Florida: CRC Press, 1997.

[7] A. Bajpai, I. Calus, and J. Fairley, *Mathematics for Engineers and Scientists*, vol. I. London: Wiley, 1973.

[8] A. E. Bell, "Phase retrieval using the wavelet transform," in *Proc. ICASSP'96*, (Atlanta), pp. 1695–1698, May 1996.

[9] J. Ben-Arie and Z. Wang, "Pictorial recognition of objects employing affine invariance in the frequency domain," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 604–618, June 1996.

[10] H. W. Bode, *Network analysis and feedback amplifier design.* Princeton, NJ: D. Van Nostrand, 1945.

[11] A. Burian, P. Kuosmanen, and C. Rusu, "1-D non-minimum phase retrieval by gain differences," in *Proc. ICECS'99*, vol. 2, (Paves, Cyprus), pp. 1001–1004, Sept. 1999.

[12] T. K. Bysted and C. Rusu, "Improving the stability of the LMF adaptive algorithm using the median filter," in *Proc. NORSIG'98*, (Vigsø, Denmark), pp. 157–160, June 1998.

[13] T. K. Bysted and C. Rusu, "A theoretical analysis of the median LMF adaptive algorithm," in *Proc. ICECS'99*, vol. 2, (Paves, Cyprus), pp. 671–674, Sept. 1999.

[14] T. K. Bysted and C. Rusu, "Theoretical aspects of the median LMF adaptive algorithm," in *Proc. FINSIG'99*, (Oulu, Finland), pp. 108–111, May 1999.

[15] J. Chambers, O. Tanrinkulu, and A. Constantinides, "Least mean mixed-norm adaptive filtering," *Electronics Letters*, vol. 30, pp. 1574–1575, Sept. 1994.

[16] W. K. Chen, ed., *The Circuits and Filters Handbook*. Boca Raton, Florida: CRC Press, 1995.

[17] T. A. C. M. Claasen and W. F. G. Mecklenbräuker, "Comparison of the convergence of two algorithms for adaptive FIR digital filters," *IEEE Trans. on ASSP*, vol. ASSP-29, pp. 670–678, June 1981.

[18] C. F. N. Cowan, "Adaptive cost functions for echo cancellers." Current projects, http://www.ee.qub.ac.uk/dsp/research/csp/index.html, The Queen's University of Belfast.

[19] C. F. N. Cowan, "The cost function adaptation issue." Personal communication, Loughborough University, UK, 19 March 1996.

[20] C. F. N. Cowan and P. M. Grant, *Adaptive Filters*. Englewood Cliffs: Prentice-Hall, 1985.

[21] C. F. N. Cowan and C. Rusu, "Adaptive echo cancellation using cost function adaptation," in *Conference Digest of Fourth IMA International Conference on Mathematics in Signal Processing*, (Warwick, UK), Dec. 1996.

[22] C. F. N. Cowan and C. Rusu, "Novel cost function adaptation algorithm for echo cancellation," in *Proc. ICASSP'98*, (Seattle, Washington), pp. 1501–1504, May 1998.

[23] J. J. D'Azzo and C. H. Houpis, *Linear control systems analysis and design: conventional and modern*. New-York: McGraw-Hill, 1995.

[24] G. Doetsch, *Introduction to the Theory and Application of the Laplace Transformation*. Berlin Heidelberg New-York: Springer-Verlag, 1970.

[25] S. C. Douglas, "Generalized gradient adaptive step-sizes for stochastic gradient adaptive filters," in *Proc. ICASSP'95*, (Detroit, MI), pp. 1396–1399, May 1995.

[26] S. C. Douglas and V. J. Mathews, "Stochastic gradient adaptive step size algorithms for adaptive filtering," in *Proc. DSP95*, (Limassol, Cyprus), pp. 142–147, June 1995.

[27] S. C. Douglas and T. H.-Y. Meng, "Stochastic gradient adaptation under general error criteria," *IEEE Transactions on Signal Processing*, vol. 42, pp. 1335–1351, June 1994.

[28] A. R. Figueiras-Vidal, ed., *Digital Signal Processing in Telecommunications*. London: Springer-Verlag, 1996.

[29] W.-S. Gan, "Designing a fuzzy step size LMS algorithm," *IEE.Proc.-Vis.Image Signal Process.*, vol. 144, no. 5, pp. 261–266, 1997.

[30] R. Gonin and A. Money, *Nonlinear L_p-Norm Estimation*. New York and Basel: Marcel Dekker, Inc., 1989.

[31] P. M. Grant, C. F. N. Cowan, B. Mulgrew, and J. H. Dripps, *Analogue and digital signal processing and coding*. Lund, SW: Chartwell-Bratt, 1989.

[32] R. W. Harris, D. Chabries, and F. Bishop, "A variable step (VS) adaptive filter algorithm," *IEEE Trans. on ASSP*, vol. ASSP-34, pp. 309–316, Apr. 1986.

[33] T. I. Haweel and P. M. Clarkson, "A class of order statistic LMS algorithms," *IEEE Transactions on Signal Processing*, vol. 40, pp. 44–53, Jan. 1992.

[34] S. Haykin, *Communications Systems*. London: Wiley, 1994.

[35] S. Haykin, *Adaptive Filter Theory*. Upper Saddle River, NJ: Prentice-Hall, 3rd ed., 1996.

[36] M. L. Honig and D. G. Messerschmitt, *Adaptive Filters: Structures, Algorithms and Applications*. Boston, MA: Kluwer Academic Publishers, 1984.

[37] H. Huttunen, M. Tico, C. Rusu, and P. Kuosmanen, "Ordering methods for multivariate RCRS filters," in *Proc. NSIP'99*, vol. II, (Antalya, Turkey), pp. 506–510, June 1999.

[38] T. Kailath, *Linear Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1980.

[39] S. Koike, "About stability of LMS, LMF and CFA." Personal communication, ICASSP'99, Phoenix, Arizona, Mar. 1999.

[40] S. Koike, "A novel adaptive step size control algorithm for adaptive filters," in *Proc. ICASSP'99*, (Phoenix, Arizona), pp. 1845–1848, Mar. 1999.

[41] H. K. Kwan, "Systolic realization of linear phase FIR digital filters," *IEEE Transactions on Circuits and Systems*, vol. CAS-34, pp. 1604–1605, Dec. 1987.

[42] R. H. Kwong and E. W. Johnston, "A variable step size LMS algorithm," *IEEE Transactions Signal Processing*, vol. 40, pp. 1633–1642, July 1992.

[43] S. Lambotharan, J. A. Chambers, and O. Tanrikulu, "Mixing and switching of adaptive filter algorithms," in *Proc. DSP95*, (Limassol, Cyprus), pp. 130–135, June 1995.

[44] S.-J. Lim and J. G. Harris, "Combined LMS/F algorithm," *Electron. Lett.*, vol. 33, pp. 467–468, Mar. 1997.

[45] O. Macchi, *Adaptive Processing. The Least Mean Squares Approach with Applications in Transmission*. Baffins Lane, Chicester, UK: John Wiley & Sons, 1995.

[46] V. J. Mathews and Z. Xie, "A stochastic gradient adaptive filter with gradient adaptive step size," *IEEE Trans. Sign. Proc.*, vol. 41, pp. 2075–2087, June 1993.

[47] D. S. Mitrinović, *Analytic Inequalities*. Berlin, Germany: Springer-Verlag, 1970.

[48] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[49] A. Papoulis, *The Fourier integral and its applications*. McGraw-Hill, 1962.

[50] A. Papoulis, *Signal analysis*. McGraw-Hill, 1977.

[51] D. I. Pazaitis and A. G. Constantinides, "LMS+F algorithm," *Electron. Lett.*, vol. 31, pp. 1423–1424, Aug. 1995.

[52] S.-C. Pei and C.-C. Tseng, "Least mean P-power error criterion for adaptive FIR filter," *IEEE Journal on Selected Areas in Communications*, vol. 12, pp. 1540–1547, Dec. 1994.

[53] A. P. Prudnikov, Y. A. Brychkov, and O. I. Marichev, *Integrals and series*. New York: Gordon & Brench Science Publishers, 1992.

[54] A. Ralston, *A first course in numerical analysis*. McGraw-Hill, 1965.

[55] W. Rudin, *Principles of Mathematical Analysis*. New-York: McGraw-Hill, 1953.

[56] C. Rusu, "A short way to compute WT from STFT," in *Proc. EUSIPCO'96*, vol. 1, (Trieste, Italy), pp. 710–713, Sept. 1996.

[57] C. Rusu, A. Burian, and P. Kuosmanen, "An LMS and LMF threshold technique," in *Proc. ECCTD'99*, vol. 2, (Stresa, Italy), pp. 1387–1390, Sept. 1999.

[58] C. Rusu, T. K. Bysted, and M. Helsingius, "On the behaviour of the median quadratic algorithms," in *Proc. COST254*, (Neuchatel, Switzerland), May 1999. accepted.

[59] C. Rusu and C. F. N. Cowan, "Linear cost function adaptation echo cancellation," in *1998 IEEE DSP Workshop Proceedings CD-ROM*, (Bryce Canyon, Utah), Aug. 1998. #027.

[60] C. Rusu and C. F. N. Cowan, "Cost function adaptation (CFA): a stochastic gradient algorithm for data echo cancellation," *IEE.Proc.-Vis.Image Signal Process.*, Mar. 1999. submitted.

[61] C. Rusu and C. F. N. Cowan, "Recursive cost function adaptation for echo cancellation," in *Proc. ICASSP'99*, vol. III, (Phoenix, Arizona), pp. 1265–1268, Mar. 1999.

[62] C. Rusu and C. F. N. Cowan, "Convergence and steady-state analysis of the cost function adaptation algorithms for data echo cancellation," in *Proc. ISCAS 2000*, (Geneva, Switzerland), May 2000. accepted.

[63] C. Rusu and I. Gavrea, "A new approach of transfer function angle," in *Proc. ICECS'96*, vol. 1, (Rodos, Greece), pp. 112–115, Oct. 1996.

[64] C. Rusu, I. Gavrea, and P. Kuosmanen, "On phase approximation by gain differences," in *Proc. ICECS'98*, vol. 2, (Lisbon, Portugal), pp. 497–500, Sept. 1998.

[65] C. Rusu, M. Helsingius, and P. Kuosmanen, "Joined LMS and LMF threshold technique for data echo cancellation," in *Proc. Int. Work. Intelligent Communications and Multimedia terminals COST #254*, (Ljubljana, Slovenia), pp. 127–130, Nov. 1998.

[66] C. Rusu and P. Kuosmanen, "Logarithmic sampling of gain and phase approximation," in *Proc. ISCAS'99*, vol. IV, (Orlando, Florida), pp. 41–44, May 1999.

[67] C. Rusu, P. Kuosmanen, and A. Burian, "1-D non-minimum phase retrieval by gain sampling," in *Proc. ECCTD'99*, vol. 2, (Stresa, Italy), pp. 755–758, Sept. 1999.

[68] T. F. Schubert, "A quantitative comparison of three Bode straight-line phase approximations for second-order, underdamped systems," *IEEE Transactions on Education*, vol. 40, pp. 135–138, May 1997.

[69] W. A. Sethares and J. A. Buclew, "Local stability of the median LMS filter," *IEEE Transactions on Signal Processing*, vol. 42, pp. 2901–2905, Nov. 1994.

[70] S. Shah, *Adaptation algorithms for data echo cancellation using nonquadratic cost functions*. PhD thesis, Loughborough University of Technology, 1995.

[71] S. Shah and C. F. N. Cowan, "Modified stochastic gradient algorithm using non-quadratic cost functions for data echo cancellation," *IEE.Proc.-Vis.Image Signal Process.*, vol. 142, pp. 187–191, June 1995.

[72] M. Shao and C. L. Nikias, "Signal processing with fractional lower order moments: Stable processes and their applications," *Proceedings IEEE*, vol. 81, pp. 986–1010, July 1993.

[73] S. Shima, C. F. N. Cowan, and M. Holt, "Adaptive channel equalisation using least mean switched error norm algorithm," in *Proc. of the IEE Sixteenth Saraga Colloquium on Digital and Analoque Filters and Filtering Systems*, vol. 8, (London, UK), pp. 1–6, 1996.

[74] Y. K. Shin and J. G. Lee, "A study of the fast convergence algorithm for the LMS adaptive filter design," *Proc. KIEE*, vol. 19, pp. 12–19, Oct. 1985.

[75] G. L. Sicuranza, "Theory and approximation of polynomial filters," in *IEEE Circuits and Systems Tutorials* (C. Toumazou, ed.), pp. 50–58, London: IEEE, 1994.

[76] H. Song and G. Gu, "Phase approximation via Remez algorithm," in *Proceedings of the Twenty-Seventh Southeastern Symposiun on System Theory*, pp. 441–444, Mar. 1995.

[77] T. Strohmer, "On the estimation of the banwidth of nonuniformly sampled signals," in *Proc. ICASSP'99*, (Phoenix, Arizona), pp. 2047–2050, Mar. 1999.

[78] O. Tanrikulu and J. A. Chambers, "Convergence and steady-state properties of the least-mean mixed-norm (LMMN) adaptive algorithm," *Vision, Image and Signal Processing, IEE Proc.*, vol. 143, pp. 137–142, June 1996.

[79] L. S. Taylor, "The phase retrieval problem," *IEEE Transactions on Antennas and Propagation*, vol. AP-29, no. 2, pp. 386–391, 1981.

[80] M. Tico, C. Rusu, and P. Kuosmanen, "A geometric invariant representation for the identification of corresponding points," in *Proc. ICIP'99*, (Kobe, Japan), Oct. 1999. accepted.

[81] E. Walach and B. Widrow, "The least mean fourth (LMF) adaptive algorithm," *IEEE Trans. Inform. Theory*, vol. 30, pp. 275–283, Mar. 1984.

[82] S. B. Weinstein, "Echo cancellation in the telephone network," *IEEE Communications Society Magazine*, vol. 15, pp. 9–15, Jan. 1977.

[83] B. Widrow and M. E. Hoff, "Adaptive switching circuits," *1960 IRE EWSCON Conv. Rec.*, vol. pt.4, pp. 96–140, 1960.

[84] B. Widrow and M. Lehr, "30 years of adaptive neural networks: perceptron, madaline, and backpropagation," *Proc. IEEE*, vol. 78, pp. 1415–1441, Sept. 1990.

[85] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.

[86] G. A. Williamson, P. M. Clarkson, and W. A. Sethares, "Performance characteristics of the median LMS adaptive filter," *IEEE Transactions on Signal Processing*, vol. 41, pp. 667–680, Feb. 1993.

[87] L. A. Zadeh and C. A. Desoer, *Linear system theory*. New York: Mc Graw Hill, 1969.

[88] A. Zerguine, M. Bettayeb, and C. F. N. Cowan, "A hybrid LMS-LMF scheme for echo cancellation," in *Proc. ICASSP'97*, (Munich, Germany), pp. 2313–2316, Apr. 1997.

[89] A. Zerguine, C. F. N. Cowan, and M. Bettayeb, "LMS-LMF adaptive scheme for echo cancellation," *Electron. Lett.*, vol. 32, pp. 1776–1778, Sept. 1996.

[90] A. Zerguine, C. F. N. Cowan, and M. Bettayeb, "Adaptive echo cancellation using least mean mixed-norm algorithm," *IEEE Trans. on Signal Processing*, vol. 45, pp. 1340–1342, May 1997.

Tampereen teknillinen korkeakoulu
PL 527
33101 Tampere

Tampere University of Technology
P. O. B. 527
FIN-33101 Tampere Finland