On Algorithms, Structures, and Implementations of Adaptive IIR Filters

by

Sergio Lima Netto
Elec. Eng., Federal University of Rio de Janeiro, 1991
M.Sc., COPPE/Federal University of Rio de Janeiro, 1992

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Electrical and Computer Engineering

We accept this dissertation as conforming
to the required standard

Dr. Panajotis Agathoklis, Supervisor (Dept. of Elect. and Comp. Eng.)

Dr. Andreas Antoniou, Departmental Member (Dept. of Elect. and Comp. Eng.)

Dr. Vijay K. Bhargava, Departmental Member (Dept. of Elect. and Comp. Eng.)

Dr. Reinhard Illner, Outside Member (Dept. of Mathematics)

Dr. Stergios Stergiopoulos, External Examiner (Defence Research Establishment Atlantlic)

Supervisor: Dr. Panajotis Agathoklis

ABSTRACT

In this dissertation, a thorough study of the area of adaptive IIR filtering is performed. A general framework to analyze adaptive systems is included along with a short presentation of the equation error (EE) and output error (OE) adaptive algorithms. Based on these algorithms, the so called composite squared error (CSE) algorithm is introduced in attempt to combine the good qualities associated to the EE and OE basic schemes. Stationary and transient analyses of the CSE convergence behavior are performed demonstrating the interesting properties associated to this algorithm. Techniques to implement the CSE algorithm using time-varying convergence factors and composite parameter are also considered. In addition, a new way to efficiently realize any adaptive IIR algorithm, including the CSE algorithm, is proposed based on the two-multiplier and the normalized lattice structures, thus allowing a necessary pole-monitoring procedure to be performed on line. The implementation of adaptive techniques to process real-time signals is considered with emphasis on the digital-signal-processor method as this approach results in a better cost/performance ratio when compared to alternative methods. The contributions of this dissertation aim to convert adaptive IIR techniques into a reliable alternative to the well-known adaptive FIR methods for all practical purposes.

Examiners:

Dr. Panajotis Agathoklis, Supervisor (Dept. of Elect. and Comp. Eng.)

Dr. Andreas Antoniou, Departmental Member (Dept. of Elect. and Comp. Eng.)

Dr. Vijay K. Bhargava, Departmental Member (Dept. of Elect. and Comp. Eng.)

Dr. Reinhard Illner, Outside Member (Dept. of Mathematics)

Dr. Stergios Stergiopoulos, External Examiner (Defence Research Establishment Atlantic)

Contents

Abstract **ii**

Contents **iii**

List of Tables **vi**

List of Figures **vii**

List of Abbreviations **x**

Acknowledgment **xi**

Dedication **xiii**

1 Introduction to Adaptive IIR Filtering **1**

1.1 Introduction 1

1.2 Adaptive Signal Processing 3

1.2.1 Basic Concepts 3

1.2.2 Technical Background 5

1.2.3 System Identification with the Direct-Form IIR Realization 6

1.2.4 Introduction to Adaptive Filter Algorithms 9

1.3 Adaptive IIR Filter Algorithms 12

1.3.1 The Equation Error Algorithm 13

1.3.2 The Output Error Algorithm 14

1.3.3 Other Adaptive IIR Filter Algorithms 18

1.4 Thesis Organization and Contributions 19

1.5 Conclusion 21

2 The Composite Squared Error Algorithm 22

2.1 Introduction 22

2.2 The Composite Squared Error Algorithm 23

2.3 Steady-State Analysis of the CSE Algorithm 25

2.4 Transient Analysis of the CSE Algorithm 32

2.4.1 The Local Linearization Approach 33

2.4.2 The Ordinary Difference Equation Approach 35

2.5 Conclusion 39

3 Time-Varying Adaptation Parameters for the CSE Algorithm 43

3.1 Introduction 43

3.2 Variable Convergence Factor for the CSE Algorithm 44

3.2.1 The Steepest-Descent CSE Case 45

3.2.2 The Quasi-Newton CSE Case 47

3.3 Variable Composite Factor for the CSE algorithm 50

3.4 Conclusion 54

4 Lattice-Based Adaptive IIR Filters 55

4.1 Introduction 55

4.2 Direct-Form Adaptive IIR Filter Algorithms 57

4.3 The Two-Multiplier Tapped Lattice IIR Realization 58

4.4 Efficient Lattice-Based Adaptive IIR Filter Algorithms 60

4.4.1 Normalized-Lattice Adaptive IIR Filter Algorithms 63

4.5 Simulation Results 64

4.6 Conclusion 69

5 On the DSP Implementation of Adaptive Filters 71

5.1 Introduction 71

5.2 Adaptive Noise Cancellation 72

5.2.1 The FIR Adaptive Noise Canceller 74

5.3 The DSP-Based Adaptive Noise Canceller 75

5.3.1 Hardware Description 75

5.3.2 Software Description 77

5.4 Practical Experiments 78

5.4.1 Description of Experiments 78

5.4.2 Experimental Results: The FIR Case 79

5.4.3 Experimental Results: The IIR Case 81

5.5 Comments and Discussions 83

5.6 Conclusion 85

6 Final Comments and Further Considerations 87

6.1 Final Comments 87

6.2 Further Considerations 89

References 91

A.1 Introduction 98

A.2 The Difference Polynomial Operator 100

A.3 The Time-Varying Difference Polynomial Operator 103

A.4 Conclusion 107

Appendix A 98

List of Tables

4.1 Summary of the efficient two-multiplier lattice CSE adaptation algorithm . 70

5.1 Number of DSP cycles per iteration for the off-line LMS algorithm. 77

List of Figures

1.1 General scheme of adaptive noise cancellation of power line interference in an ECG recording system. 2

1.2 Block diagram of a general adaptive system. 3

1.3 Block diagram of an adaptive system identifier. 7

2.1 Example 2.7 - MCSE performance surfaces: (A) $\gamma = 1.0$; (B) $\gamma = 0.6$; (C) $\gamma = 0.2$; (D) $\gamma = 0.0$. 32

2.2 Example 2.10 - CSE parameter error trajectories: Predicted (A) and actual (B) for $\gamma = 1.0$; predicted (C) and actual (D) for $\gamma = 0.8$; predicted (E) and actual (F) for $\gamma = 0.0$. 40

2.3 Example 2.13 - Predicted CSE parameter trajectories using the ODE method: (A) $\gamma = 1.0$; (B) $\gamma = 0.6$; (C) $\gamma = 0.2$; (D) $\gamma = 0.0$. 41

2.4 Example 2.13 - Actual CSE parameter trajectories: (A) $\gamma = 1.0$; (B) $\gamma = 0.6$; (C) $\gamma = 0.2$; (D) $\gamma = 0.0$. 42

3.1 Example 3.5 - Number of iterations required for the CSE signal averaged over 50 experiments to reach -200dB as a function of μ in the fixed convergence factor case ('o') and α in the variable convergence factor case ('x'), for the SD-CSE algorithm. 47

3.2 Example 3.8 - Number of iterations required for the CSE signal averaged over 50 experiments to reach -200dB as a function of μ in the fixed convergence factor case ('o') and α in the variable convergence factor case ('x'), for the QN-CSE algorithm. 49

3.3 Example 3.9 - Convergence in time of adaptive filter coefficients using the CSE algorithm with time-varying composite parameter. 51

3.4 Example 3.9 - Convergence in time of the time-varying composite parameter: (A) $\hat{\boldsymbol{\theta}}(0) = [0.8\ 0.3]^T$; (B) $\hat{\boldsymbol{\theta}}(0) = [0.0\ -0.3]^T$. 52

4.1 Example 4.5 - Lattice adaptive convergence of the CSE algorithm with $\gamma = 0.0$: (A) Two-multiplier denominator coefficients; (B) Two-multiplier numerator coefficients; (C) Normalized denominator coefficients; (D) Normalized numerator coefficients. 66

4.2 Example 4.5 - Lattice adaptive convergence of the CSE algorithm with $\gamma = 1.0$ - With perturbation noise: (A) Two-multiplier denominator coefficients; (B) Two-multiplier numerator coefficients; (C) Normalized denominator coefficients; (D) Normalized numerator coefficients. 67

4.3 Example 4.5 - Lattice adaptive convergence of the CSE algorithm with $\gamma = 1.0$ - Without perturbation noise: (A) Two-multiplier denominator coefficients; (B) Two-multiplier numerator coefficients; (C) Normalized denominator coefficients; (D) Normalized numerator coefficients. 68

5.1 Block diagram of an adaptive noise canceller. 73

5.2 Number of cycles per iteration × filter order for the off-line LMS algorithm using standard C, optimized C, and assembly languages. 77

5.3 Noise cancellation example using the LMS algorithm with $\mu = 0.003$ and $N = 300$: (A) Original reference signal; (B) Resulting error signal. 80

5.4 Maximum filter order × sampling frequency for the real-time implementation of the LMS algorithm using assembly language. 80

5.5 Steady-state mean squared errors of the LMS algorithm for maximum convergence speed ('o') and for convergence after 20000 iterations ('x') as functions of the adaptive filter order N. 81

5.6 Maximum filter order × sampling frequency for the real-time implementation of the CSE algorithm using assembly language. 82

5.7 Steady-state mean squared error of the CSE algorithm for convergence after 20000 iterations with $\gamma = 1.0$ ('o') and $\gamma = 0.0$ ('x') as function of the filter order N. 83

5.8 Steady-state mean squared error of the CSE algorithm for convergence after 20000 iterations with $N = 15$ as function of the composite factor value γ. . 84

A.1 Example A.18 - Time-varying coefficients $c_0(n)$, $c_1(n)$, and $d_1(n)$, with $s = 0.2$. 106

A.2 Example A.18 - Signals $y_1(n)$, $y_2(n)$, and $(y_1(n) - y_2(n))$ for $0 \leq n \leq 200$ and $90 \leq n \leq 120$. 106

A.3 Example A.18 - Average maximum value of the difference $(y_1(n) - y_2(n))$ as a function of the time scale s. 107

List of Abbreviations

A/D	:	Analog to Digital
CSE	:	Composite Squared Error
D/A	:	Digital to Analog
DM	:	Daughter Module
DPO	:	Delay Polynomial Operator
DSP	:	Digital Signal Processor
ECG	:	Electrocardiographic
EE	:	Equation Error
FIR	:	Finite-duration Impulse Response
GNM	:	Global Nonconvexity Method
HCM	:	Homotopy Continuation Method
IIR	:	Infinite-duration Impulse Response
ISV	:	Instantaneous Squared Value
LMS	:	Least Mean Squares
LS	:	Least Squares
LSI	:	Loughborough Sound Images
MSCE	:	Mean Squared Composite Error
MSE	:	Mean Squared Error
MSEE	:	Mean Squared Equation Error
MSOE	:	Mean Squared Output Error
ODE	:	Ordinary Difference Equation
OE	:	Output Error
RLS	:	Recursive Least Squares
SHARF	:	Simplified Hyperstable Adaptive Recursive Filtering
SM	:	Steiglitz-McBride
TDL	:	Tapped Delay Line
TI	:	Texas Instruments
TVDPO	:	Time Varying Delay Polynomial Operator

Acknowledgment

The writing of this dissertation could not have been made possible without technical and fraternal aid from many people that, willing or not, have become part of this work, probably without even knowing.

First, I shall thank my parents Maria Christina and Sérgio for giving me the opportunity and inspiration to follow my dreams and make them turn into reality. I must also readily thank my sisters Paula, Lúcia, and Patrícia and my brother Jair, always present in their own special way. Overall, I do acknowledge the importance of my entire family for their friendship and love throughout all these years.

In addition, I would like to mention some very important friends that in many ways have become an important part of my life. These friends include, using the Brazilian colloquial way of solely referring to their given names, true life companions as Marcello Luiz and Luciana, Bruno, Leandro, Hélio, Ronald and Rebecca, Luiz Felipe Baginski (allow me in this case to include the complete name), Sérgio Ricardo, Márcio Magalhães, Leonardo, Márcio Guidorizzi, Juan, Alexandre, André, Paulo, Marcelo, Vera, Peixoto and Sean, Lane, José Luiz and Adriana, Miriam, Maher, Nelson, Srikanth, Dipankar, Ali, and many others that I am sure I will later regret not having them included in this section.

I must also deeply thank CAPES/Ministry of Education - Brazil for the financial support of my doctorate program.

In the technical aspect, and others too, I would like to mention my previous supervisor, Dr. Paulo Sérgio Ramirez Diniz, for the continuous guidance and friendship through all my academic life. In fact, I take the opportunity here to deeply thank all my idols and teachers, in the noblest and most general forms of these words, for paving and illuminating the way for a formidable voyage through knowledge and life.

Last, but by no means least, I thank my supervisor, Prof. Panajotis Agathoklis, for giving me the opportunity to perform this work, and for his patience, guidance, and the always present support during the doctorate program at the University of Victoria.

For all of you, and many others whose names are not included here but surely deserve my respect and admiration, I humbly offer my deepest and sincere thanks.

à Vida,

é claro

to Life,

naturally

Chapter 1

Introduction to Adaptive IIR Filtering

1.1 Introduction

The term adaptive filter usually refers to a system the characteristics of which are modified by adjusting the coefficients of its transfer function in an automatic fashion in order to satisfy a particular specification or criterion.

The number of different applications in which adaptive techniques are being successfully used increased enormously recently. These applications include, for instance, system identification [22], equalization of dispersive channels for faster data transmission [25], echo cancellation in telephone lines [94], beamforming using linear arrays [99], and noise cancellation [99].

An example of a real application where adaptive filters are successfully employed is the measurement of electrocardiographic (ECG) signals which, in practice, present very small amplitudes and most frequency components in the range below 100Hz [94], [99]. Due to these characteristics, recording ECG signals tends to be very sensitive to power-line interference. Fixed-in-time solutions for this problem might not be able to accurately eliminate the perturbation noise if the power line fluctuates making the interference characteristics varying in time. A very efficient and robust way to solve this problem is illustrated in Figure 1.1 where the combination (ECG + interference) signal is processed by an adaptive system which has also access to some version of the interference signal. By properly adjusting the amplitude, the frequency, and the phase of its sinusoidal output signal, the adaptive filter is then able to successfully cancel out the interference from the original ECG signal. Due to its own adjustable nature, the adaptive filter can accomplish the noise reduction in a continuous fashion by being able to track fluctuations on the perturbation characteristics. Adaptive filters are advantageous in this situation due to the particular nature of heartbeat signals which are very distinctive for each individual person and, in addition, continuously change in time. This example demonstrates the general philosophy behind the usage of adaptive filters. The employment of adaptive techniques becomes particularly useful in cases where the characteristics of the surrounding environment are not completely known or even varying in time, as under these circumstances a fixed system would not be able to achieve a desirable performance level. Others forms of interference can also be eliminated using adaptive systems as it will be discussed later.

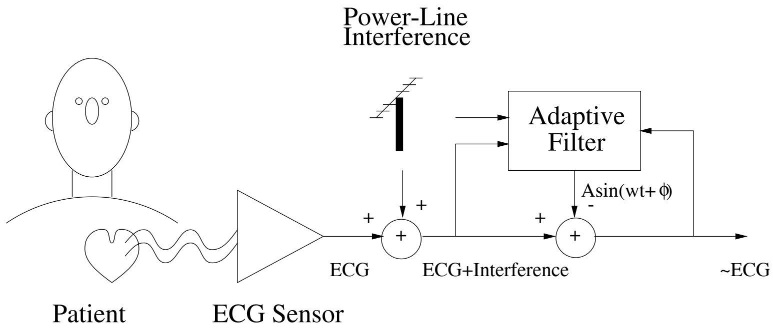


Figure 1.1: General scheme of adaptive noise cancellation of power line interference in an ECG recording system.

Early forms of adaptive filters [99] were based on structures with finite-duration impulse response (FIR) that presented very good adaptation properties such as existence of a single solution, good convergence speed, filter stability, etc. However, due to steadily increasing demand for more efficient adaptive filters to be applied in a wide variety of applications, substantial research effort has been invested to turn adaptive filters with infinite-duration impulse response (IIR)[1] into reliable alternatives to traditional adaptive FIR filters. The main advantages of IIR filters are their better suitability for modelling physical systems due to the pole-zero structure and their requirement of fewer parameters to achieve the

[1] The acronyms FIR and IIR are commonly used in time-invariant digital filter theory to indicate, respectively, the finite or infinite duration of the impulse response of these devices. The same terminology also applies to the theory of adaptive filters.

same performance as FIR filters. Unfortunately, these good characteristics come along with possible drawbacks such as algorithm instability, filter instability, convergence to biased or local minima, and slow convergence. The objective of this dissertation is to present reliable techniques for adaptive IIR filters in an attempt to make such systems an efficient and robust practical alternative to adaptive FIR filters.

1.2 Adaptive Signal Processing

1.2.1 Basic Concepts

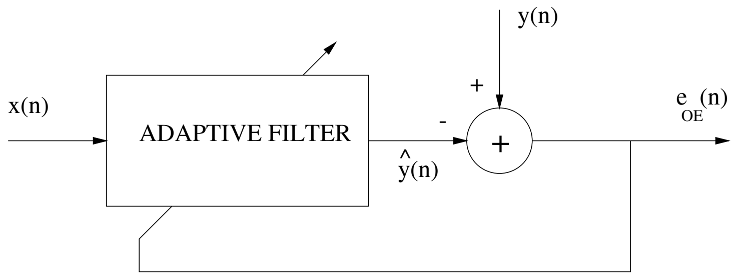


Figure 1.2: Block diagram of a general adaptive system.

The general configuration of a basic adaptive system is depicted in Figure 1.2. In this type of process, an input signal $x(n)$ is filtered by a time-varying system generating at each time interval n the output $\hat{y}(n)$. This signal is then compared to a reference $y(n)$, also called the desired output signal, leading to the error signal $e_{OE}(n)$. This error signal is then used by an algorithm to adjust the adaptive filter coefficients in order to minimize a given performance criterion. The specification of an adaptive system, as shown in Figure 1.2, consists of three items: Application, filter structure, and algorithm.

The type of application defines the input and reference signals acquired from the surrounding environment. For the interested reader, good sources of information on adaptive filtering applications can be found in [25], [94], [99]. In this dissertation, the system identification problem is briefly introduced in the next section and it serves as basic environment throughout the remainder of this dissertation, with exception of Chapter 5, where the noise cancellation problem is considered.

The choice of the adaptive filter structure defines the mathematical model to be used and the parameters to be adapted. Basically, there are two classes of adaptive digital filter realizations[2] : FIR and IIR structures. The structure being used influences the computational complexity (amount of arithmetic operations per iteration) of the adaptation process and also the necessary number of iterations to achieve a desired performance level.

The most widely used adaptive filter structure is the transversal FIR filter, also referred to as the tapped delay line (TDL), that implements a nonrecursive transfer function in a canonic form without feedback. For this realization, the output signal $\hat{y}(n)$ is a linear combination of the filter coefficients, which yields a quadratic mean squared error function $E[e^2_{OE}(n)]$ with respect to the adaptable coefficients with a unique optimum point [99]. Other alternative adaptive FIR realizations are also used in order to obtain improvements as compared to the transversal filter structure in terms of computational complexity [9], [19], speed of convergence [45], [49], [50], and finite wordlength properties [25]. In this dissertation, although most of the work is concentrated in adaptive IIR filters, the basic TDL is used in Chapter 5 with the standard least mean squares (LMS) algorithm [99] for the sake of performance comparison with some of the IIR adaptation techniques.

The earliest attempt to implement an adaptive IIR filter reported in the literature was made by White [98] in 1975 using the standard IIR direct form. Since then, a large number of papers have been published in this area. Initially, most of the work on adaptive IIR filters was based on the canonic IIR direct-form realization due to its simple implementation and analysis. However, due to some inherent problems of recursive adaptive filters, such as continuous pole-monitoring requirement and slow speed of convergence, different realizations were contemplated in an attempt to overcome the limitations of the direct-form structure. Among these alternative structures, the cascade [8], parallel [73], and lattice [65] realizations were considered due to their unique feature of allowing simple pole monitoring for stability testing. The cascade and parallel structures, however, lead to manifolds on the respective error performance surfaces which cause the adaptive convergence process to be dramatically slowed down [51]. For the lattice realization, the large computational complexity to calculate its gradient vector has prevented its widespread use in practical systems. New efficient lattice-type realizations, however, are proposed in Chapter 4 of this dissertation, turning

[2]Despite the fact that some adaptive filters can also be implemented with continuous-time techniques, general results have shown that this type of realization still faces many practical implementation problems [31], [46], [95]. Because of that, this work will focus on discrete-time implementations of adaptive systems.

this structure into a reliable and efficient tool for the realization of adaptive IIR filters.

The algorithm is the procedure used to adjust the adaptive filter coefficients in order to minimize a prescribed criterion. The algorithm is determined by defining the search method or minimization algorithm, the objective function and the error signal nature. The choice of the algorithm determines several crucial aspects of the adaptive process such as existence of sub-optimal or biased solutions, the convergence speed, and the overall computational complexity. A general discussion of some of the most important adaptation algorithms is presented later in this chapter.

1.2.2 Technical Background

In this section, some definitions to be used later are presented. One of the most heavily used concepts in this dissertation is the difference polynomial operator used in the time-domain description of adaptive systems. A thorough presentation including definition, properties, and extensions of this concept can be found in Appendix A. Due to its importance, the reader who is unfamiliar with its properties and associated notation is referred to Appendix A, where much valuable information can be gathered on that subject.

Definition 1.1: A signal $u(n)$ is said to be persistently exciting [75] of order $\tilde{n}$ if the limit

$$\lim_{N\to\infty} \frac{1}{N}\sum_{n=1}^{N} u(n)u(n+\tau) = r_u(\tau) \tag{1.1}$$

exists and the following $\tilde{n} \times \tilde{n}$ matrix is positive definite

$$\begin{bmatrix} r_u(0) & \cdots & r_u(\tilde{n}-1) \\ \vdots & \ddots & \vdots \\ r_u(\tilde{n}-1) & \cdots & r_u(0) \end{bmatrix} \tag{1.2}$$

□

An important result on persistently exciting sequences states that [75] if $u(n)$ is persistently exciting of order $\tilde{n}$, then there is no filter described by $H(q) = h_0 + h_1 q^{-1} + \ldots + h_{\tilde{n}} q^{-\tilde{n}}$ with $h_{\tilde{n}} \neq 0$, such that $H(q)\{u(n)\} \equiv 0$. In addition, if $v(n) = H(q)\{u(n)\}$, then $v(n)$ is also persistently exciting of order $\tilde{n}$. In general terms, the persistence of excitation concept indicates when an input signal carries enough information in the time or frequency domain to allow a given plant to be completely identified.

Definition 1.2: For a given stochastic process $u(n)$, the ith-order moment is defined as [64]

$$m_i = E\left[u^i(n)\right] \tag{1.3}$$

□

Definition 1.3: Conditional probability of two events U and V is defined as [64]

$$P(U/V) = \frac{P(UV)}{P(V)} \tag{1.4}$$

where $P(UV)$ is the joint probability of both events U and V and $P(V)$ is different from zero by assumption.

□

Definition 1.4: The pair of signals $u(n)$ and $v(n)$ is said to be ϕ-mixing if [14]

$$\sum_{n=0}^{\infty} \phi^{1/2}(n) < \infty \tag{1.5}$$

with

$$\phi(n) = sup \left\{ |P(U/V) - P(U)| \, ; \begin{array}{l} U \in \sigma\left\{[u(k) \;\, v(k)]; k \leq 0\right\} \\ V \in \sigma\left\{[u(l+n) \;\, v(l+n)]; l \geq 0\right\} \end{array} \right\} \tag{1.6}$$

where $sup\{.\}$ denotes the supreme or minimum-upper-bound operator and $\sigma\{.\}$ represents the algebraic field generated by the corresponding elements.

□

Essentially, the ϕ-mixing condition implies that $u(n)$ and $v(n)$ are uncorrelated to each other for large time separations as indicated in [14].

1.2.3 System Identification with the Direct-Form IIR Realization

In order to present a simple framework for the remainder of this dissertation, most of the analyses shown in this work will be based on the system identification application and on the direct-form IIR structure. Most results discussed, however, can be extended to other applications and realizations, following the works of Johnson [32], [33] and Nayeri and Jenkins [51]. In fact, in [32], [33], Johnson discusses the relationships between system identification, adaptive filtering, and control problems. Meanwhile, in [51], Nayeri and Jenkins show the relationships existing between the direct form and other alternative IIR realizations as the cascade, parallel, and lattice forms.

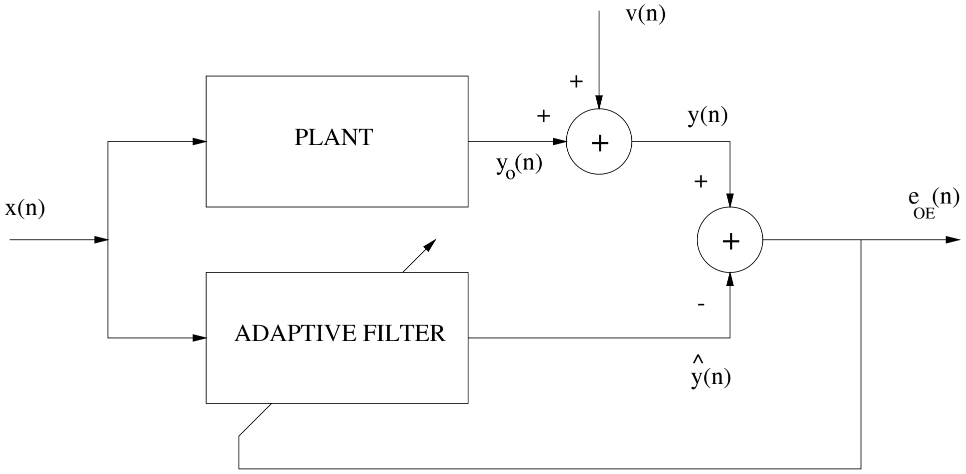


Figure 1.3: Block diagram of an adaptive system identifier.

The general block diagram of an adaptive system identifier is shown in Figure 1.3. In this configuration, an adaptation algorithm adjusts the adaptive filter coefficients such that the filter input-output relationship better matches that of an unknown plant. The plant, or unknown system, is commonly assumed time-invariant and stable, being described by

$$y_o(n) = \left[\frac{B(q)}{A(q)}\right]\{x(n)\} \tag{1.7a}$$

$$y(n) = y_o(n) + v(n) \tag{1.7b}$$

or equivalently

$$y(n) = \left[\frac{B(q)}{A(q)}\right]\{x(n)\} + v(n) \tag{1.8}$$

where $x(n)$ is the input signal, $v(n)$ represents some form of perturbation noise, and $A(q) = 1 + \sum_{i=1}^{n_a} a_i q^{-i}$ and $B(q) = \sum_{j=0}^{n_b} b_j q^{-j}$ are coprime polynomials of the unit delay operator[3] defined by $q^{-1}\{x(n)\} = x(n-1)$. Using the direct-form structure, the adaptive filter is described by

$$\hat{y}(n) = \left[\frac{\hat{B}(q,n)}{\hat{A}(q,n)}\right]\{x(n)\} \tag{1.9}$$

with $\hat{A}(q,n) = 1 + \sum_{i=1}^{n_{\hat{a}}} \hat{a}_i(n) q^{-i}$ and $\hat{B}(q,n) = \sum_{j=0}^{n_{\hat{b}}} \hat{b}_j(n) q^{-j}$.

An equivalent way to represent the adaptive identification process depicted in Figure 1.3

[3]For a complete discussion of the unit delay operator and of some of its extensions, the reader may refer to Appendix A.

is obtained by defining the variables

$$\boldsymbol{\theta} = [a_1 \ldots a_{n_a} \; b_0 \ldots b_{n_b}]^T \tag{1.10a}$$

$$\boldsymbol{\phi}(n) = [-y_o(n-1) \ldots - y_o(n-n_a) \; x(n) \ldots x(n-n_b)]^T \tag{1.10b}$$

$$\hat{\boldsymbol{\theta}}(n) = \left[\hat{a}_1(n) \ldots \hat{a}_{n_{\hat{a}}}(n) \; \hat{b}_0(n) \ldots \hat{b}_{n_{\hat{b}}}(n)\right]^T \tag{1.10c}$$

$$\hat{\boldsymbol{\phi}}_{MOE}(n) = \left[-\hat{y}(n-1) \ldots - \hat{y}(n-n_{\hat{a}}) \; x(n) \ldots x(n-n_{\hat{b}})\right]^T \tag{1.10d}$$

where $\boldsymbol{\theta}$ is the plant parameter vector, $\boldsymbol{\phi}(n)$ is the plant information vector, $\hat{\boldsymbol{\theta}}(n)$ is the adaptive filter parameter vector, and $\hat{\boldsymbol{\phi}}_{MOE}(n)$ is the adaptive filter information vector. With the above definitions, equations (1.8) and (1.9) become

$$y(n) = \boldsymbol{\theta}^T \boldsymbol{\phi}(n) + v(n) \tag{1.11}$$

$$\hat{y}(n) = \hat{\boldsymbol{\theta}}^T(n) \hat{\boldsymbol{\phi}}_{MOE}(n) \tag{1.12}$$

respectively. Comparing the difference polynomial operator notation seen in (1.8) and (1.9) with the vector notation given in (1.11) and (1.12), it can be seen that the physical meaning of a signal is more clear when the delay operator is used, while the vector notation greatly simplifies the adaptation algorithm representation, as it will become clear later in this chapter.

In order to present an overview of some adaptive IIR filtering algorithms in a structured form, it is useful to classify the IIR identification problem using three distinct criteria given below.

Classification with respect to the adaptive IIR filter orders: Let $n^* = min[(n_a - n_{\hat{a}}); (n_b - n_{\hat{b}})]$, thus the following classes are defined

- Case (a): Insufficient order case, where $n^* < 0$;
- Case (b): Strictly sufficient order case, where $n^* = 0$;
- Case (c): More than sufficient order case, where $n^* > 0$.

In many cases, cases (b) and (c) are grouped in one single class, called the sufficient order case, where $n^* \geq 0$.

Classification with respect to the input signal properties:

- Case (d): Problem with persistent exciting input signal;
- Case (e): Problem with nonpersistent exciting input signal.

Basically, the persistence of excitation concept [1], [75] is associated with the amount of information carried by the external signal $x(n)$ to the adaptive process. Processes belonging to class (e) may lead to situations where it is not possible to identify the system parameters and therefore they are not commonly considered in the literature.

Classification with respect to the disturbance signal properties:

- Case (f): Without perturbation;
- Case (g): With perturbation correlated to the input signal;
- Case (h): With perturbation uncorrelated to the input signal.

The presence or not of perturbation signal greatly influences the overall performance level of a given adaptive system as will later be demonstrated. Also, it must be mentioned that case (g) can be considered a special case of (a), and therefore it is often disregarded in the literature.

Following this framework, all the above described cases are widely studied in the literature and will be addressed at the appropriate time in this dissertation, with the exception of cases (e) and (g), due to the reasons mentioned above.

1.2.4 Introduction to Adaptive Filter Algorithms

The basic objective of the adaptive filter in a system identification problem is to find the parameter vector $\hat{\boldsymbol{\theta}}(n)$ that equivalently represents the input-output relationship of the unknown system, i.e., the mapping of $x(n)$ into $y(n)$. Usually, system equivalence [1] is determined by the objective functional $\mathcal{W}$ of the input $x(n)$, the available plant output $y(n)$, and the adaptive filter output $\hat{y}(n)$ signals. In that sense, two systems, S1 and S2, are considered equivalent if, for the same external signals $x(n)$ and $y(n)$, the objective function assumes the same value for these systems, i.e.

$$\mathcal{W}\left[x(n), y(n), \hat{y}_1(n)\right] = \mathcal{W}\left[x(n), y(n), \hat{y}_2(n)\right] \tag{1.13}$$

In an adaptive system identification process, this concept implies that [42], [43] the adaptation algorithm attempts to minimize the functional $\mathcal{W}$ in such a way that $\hat{y}(n)$ approximates $y(n)$ and, as a consequence, $\hat{\boldsymbol{\theta}}(n)$ converges to $\boldsymbol{\theta}$ or to a best possible approximation of this vector.

It is important to notice, however, that in order to have a meaningful definition of the objective function, this function must satisfy the nonnegativity and optimality properties described respectively by

$$\mathcal{W}[x(n), y(n), \hat{y}(n)] \geq 0; \forall \hat{y}(n) \tag{1.14a}$$

$$\mathcal{W}[x(n), y(n), y(n)] = 0 \tag{1.14b}$$

Another way to interpret the objective function is to consider it as a direct function of a generic error signal $e(n)$, which in turn is a function of the signals $x(n)$, $y(n)$, and $\hat{y}(n)$. That function $\mathcal{W}[e(n)]$ defines a surface in the domain of the adaptive filter coefficients that is referred to as the performance surface [83]. Based on this interpretation, the adaptive filter algorithm can be seen as the numeric procedure used to adjust the adaptive filter coefficients in order to search for the global minimum of that geometric performance surface. In practice, the adaptation algorithm is characterized by the definition of three basic aspects: The minimization algorithm, the form of the objective function, and the error signal. These items are discussed in the remainder of this section.

The minimization algorithm for the functional $\mathcal{W}$ is the subject of optimization theory and it essentially affects the speed of convergence and the computational complexity of the adaptation process. The most commonly used optimization methods in the adaptive signal processing field include the Newton method, the quasi-Newton methods, and the steepest-descent method.

The Newton method seeks the minimum of a second-order approximation of the objective function $\mathcal{W}[e(n)]$ using an iterative updating formula for the parameter vector of the form

$$\hat{\boldsymbol{\theta}}(n+1) = \hat{\boldsymbol{\theta}}(n) - \mu' \mathcal{H}^{-1}_{\hat{\boldsymbol{\theta}}}\{\mathcal{W}[e(n)]\} \boldsymbol{\nabla}_{\hat{\boldsymbol{\theta}}}\{\mathcal{W}[e(n)]\} \tag{1.15}$$

where μ' is a factor that controls the step size of the algorithm, $\mathcal{H}_{\hat{\boldsymbol{\theta}}}\{\mathcal{W}[e(n)]\}$ is the Hessian matrix of the objective function [44], and $\boldsymbol{\nabla}_{\hat{\boldsymbol{\theta}}}\{\mathcal{W}[e(n)]\}$ is the gradient of the objective function with respect to the adaptive filter coefficients.

Quasi-Newton methods are simplified versions of the standard Newton method. They attempt to minimize the objective function $\mathcal{W}[e(n)]$ using a recursively calculated estimate of the inverse of the Hessian matrix and they are described by

$$\hat{\boldsymbol{\theta}}(n+1) = \hat{\boldsymbol{\theta}}(n) - \mu' \mathbf{T}(n+1) \boldsymbol{\nabla}_{\hat{\boldsymbol{\theta}}}\{\mathcal{W}[e(n)]\} \tag{1.16}$$

where $\mathbf{T}(n)$ is an estimate of the inverse of the Hessian matrix $\mathcal{H}^{-1}_{\hat{\boldsymbol{\theta}}}\{\mathcal{W}[e(n)]\}$, such that $\lim_{n\to\infty} \mathbf{T}(n) = \mathcal{H}^{-1}_{\hat{\boldsymbol{\theta}}}\{\mathcal{W}[e(n)]\}$. A common form to implement this approximation for the inverse of the Hessian matrix is obtained using the matrix inversion lemma [44], which yields the recursion

$$\mathbf{T}(n+1) = \frac{1}{\lambda}\mathbf{T}(n) - \frac{\mu'}{\lambda}\left[\frac{\mathbf{T}(n)\hat{\boldsymbol{\phi}}(n)\hat{\boldsymbol{\phi}}^T(n)\mathbf{T}(n)}{\lambda + \mu'\hat{\boldsymbol{\phi}}^T(n)\mathbf{T}(n)\hat{\boldsymbol{\phi}}(n)}\right] \tag{1.17}$$

where λ is the so-called forgetting factor usually defined as $\lambda = 1 - \mu'$.

The steepest-descent method searches for the minimum of the objective function following the opposite direction of the gradient vector of this function. The updating equation for this type of algorithm assumes the form

$$\hat{\boldsymbol{\theta}}(n+1) = \hat{\boldsymbol{\theta}}(n) - \mu'\boldsymbol{\nabla}_{\hat{\boldsymbol{\theta}}}\{\mathcal{W}[e(n)]\} \tag{1.18}$$

In general, the steepest-descent method is the easiest one to be implemented but, on the other hand, the Newton method usually requires a smaller number of iterations to converge. In many cases, quasi-Newton methods can be considered a good compromise between the computational efficiency of the gradient method and the fast convergence of the Newton method. A detailed study of the most widely used minimization algorithms is out of the scope of this dissertation and can be found in the seminal work of Luenberger [44].

The definition of the form of the objective function $\mathcal{W}[e(n)]$ directly affects the complexity of the gradient vector and Hessian matrix calculations. There are many ways to define an objective function that satisfies the nonnegativity and optimality properties seen in (1.14). The following forms for the objective function are the most commonly used in the derivation of adaptation algorithms. The mean squared error (MSE) function is given by

$$\mathcal{W}[e(n)] = E\left[e^2(n)\right] \tag{1.19}$$

The least squares (LS) function is defined as

$$\mathcal{W}[e(n)] = \frac{1}{N+1}\sum_{i=0}^{N} e^2(n-i) \tag{1.20}$$

where N is a particular number of samples specified by the algorithm designer. Finally, the instantaneous squared value (ISV) function is characterized by

$$\mathcal{W}[e(n)] = e^2(n) \tag{1.21}$$

In a strict sense, the MSE method is of theoretical value since it requires an infinite amount of information to be processed. In practice, this ideal objective function is approximated by the other two methods listed. The LS and ISV schemes differ in the implementation complexity and in the convergence behavior characteristics. In general, the ISV method is easier to be implemented but it tends to present noisier convergence properties as it represents a greatly simplified estimate of the MSE objective function.

The definition of the error signal $e(n)$ is the last, but by no means the least important, factor to completely characterize a given adaptation algorithm. Its choice is crucial for the algorithm definition since it affects characteristics of the overall algorithm such as computational complexity, speed of convergence, robustness, and, most importantly, the existence of biased or multiple solutions. Due to its importance, however, a deeper analysis introducing several examples of error signal definitions is left to be presented in detail in the following section.

As it was described here, the minimization algorithm, the objective function, and the error signal give us a structured and simple way to interpret and understand the intrinsic structure of an adaptation algorithm. In fact, all currently known adaptation algorithms can in some way or another be analyzed following this framework. In the next section, we present a detailed review of two well known adaptation algorithms used in adaptive IIR filtering, emphasizing their similarities, distinctions, and their advantages and disadvantages when compared to each other. That discussion will serve as a motivating point for the introduction in the subsequent chapters of newly proposed efficient techniques for adaptive IIR filtering.

1.3 Adaptive IIR Filter Algorithms

The discussion in the previous section indicates that the minimization algorithm and the form of the objective function affect mainly the convergence speed and computational complexity of the adaptation process. On the other hand, the most important aspect in the definition of an adaptation algorithm consists of the choice of the error signal, since the convergence properties of the adaptation algorithm are greatly influenced by this signal. Therefore, in order to concentrate the analysis on the influence of the error signal, the minimization algorithm and the objective function will be fixed. In this section, the minimization algorithm will be the SD method and the objective function will the ISV of the error signal. Based on this framework, consider the following adaptive IIR algorithms.

1.3.1 The Equation Error Algorithm

The simplest way to model an unknown system is to use the input-output relationship described by a linear difference equation of the form

$$\begin{aligned} y(n) = {} & \hat{b}_0(n)x(n) + \ldots + \hat{b}_{n_{\hat{b}}}(n)x(n-n_{\hat{b}}) \\ & -\hat{a}_1(n)y(n-1) - \ldots - \hat{a}_{n_{\hat{a}}}(n)y(n-n_{\hat{a}}) + e_{EE}(n) \end{aligned} \tag{1.22}$$

where $\hat{a}_i(n)$ and $\hat{b}_j(n)$ are the parameters for the direct structure and $e_{EE}(n)$ is a residual error, referred to as the equation error signal. Equation (1.22) can be rewritten using the delay polynomial operator form as

$$y(n) = \left[\frac{\hat{B}(q,n)}{\hat{A}(q,n)}\right]\{x(n)\} + \left[\frac{1}{\hat{A}(q,n)}\right]\{e_{EE}(n)\} \tag{1.23}$$

or in the vector form as

$$y(n) = \hat{\boldsymbol{\theta}}^T(n)\hat{\boldsymbol{\phi}}_{EE}(n) + e_{EE}(n) \tag{1.24}$$

with

$$\hat{\boldsymbol{\phi}}_{EE}(n) = \left[-y(n-1)\ldots -y(n-n_{\hat{a}})\ x(n)\ldots x(n-n_{\hat{b}})\right]^T \tag{1.25}$$

From the previous equations, it is easy to verify that the adaptation algorithm that attempts to minimize the ISV of the equation error, $e_{EE}^2(n)$, using a gradient-type search method, assumes the form

$$\hat{\boldsymbol{\theta}}(n+1) = \hat{\boldsymbol{\theta}}(n) + \mu e_{EE}(n)\hat{\boldsymbol{\phi}}_{EE}(n) \tag{1.26}$$

This equation identifies the so-called equation error (EE) adaptive IIR algorithm which is characterized by the following property [22].

Property 1.5: The Euclidean norm of the error parameter vector defined by $s(n) = \parallel \hat{\boldsymbol{\theta}}(n) - \boldsymbol{\theta} \parallel^2$ and the EE signal, $e_{EE}(n)$, are convergent sequences if $n^* \geq 0$, $v(n) \equiv 0$, and μ satisfies

$$0 < \mu < \frac{2}{\parallel \hat{\boldsymbol{\phi}}_{EE}(n) \parallel^2} \tag{1.27}$$

□

This first property establishes the interval of μ that guarantees convergence and stability for the EE algorithm. However, although this property asserts that $s(n)$ and $e_{EE}(n)$ are convergent sequences, it is not clear to what value these sequences tend to. Although the derivation of the EE algorithm was based on the ISV function of the EE signal, the properties of the EE solution are usually referred to the mean squared equation error (MSEE) function $E[e_{EE}^2(n)]$ which is characterized below [22].

Property 1.6: If $n^* \geq 0$, $v(n) \equiv 0$, and the input signal is persistently exciting of the order $max[(n_{\hat{a}} + n_b);(n_{\hat{b}} + n_a)]$, then the MSEE function $E[e_{EE}^2(n)]$ has only global minimum solutions of the form

$$\hat{A}^*(q) = A(q)L(q) \tag{1.28a}$$

$$\hat{B}^*(q) = B(q)L(q) \tag{1.28b}$$

where $L(q) = 1 + l_1 q^{-1} + \ldots + l_{n_l} q^{-n_l}$ represents the common factor between $\hat{A}^*(q)$ and $\hat{B}^*(q)$.

On the other hand, if $n^* < 0$ or if perturbation noise is present, the optimal solution is unique but it does not minimize the mean squared output error. The bias of this solution depends on the plant transfer function and on the perturbation signal characteristics.

□

This property indicates that in cases of sufficient order identification, given the above two conditions, any EE solution is a global minimum of the MSEE function that includes the polynomials describing the plant and a common factor $L(q)$ present in the numerator and denominator polynomials of the adaptive filter transfer function.

In short, the main characteristic of the EE algorithm is the fact that the MSEE function is quadratic with respect to the adaptive filter coefficients what results in a unique solution, given that the input signal is persistently exciting of sufficient order. This property, however, comes along with the serious problem of generating a biased solution in the presence of any form of disturbance signal $v(n)$.

1.3.2 The Output Error Algorithm

The output error (OE) algorithm attempts to minimize the mean squared value of the output error signal defined as the difference between the plant and the adaptive filter output signals, i.e.

$$
\begin{aligned}
e_{OE}(n) &= \left[\frac{B(q)}{A(q)} - \frac{\hat{B}(q,n)}{\hat{A}(q,n)}\right]\{x(n)\} + v(n) \\
&= \boldsymbol{\theta}^T \boldsymbol{\phi}(n) - \hat{\boldsymbol{\theta}}^T(n)\hat{\boldsymbol{\phi}}_{MOE}(n) + v(n)
\end{aligned}
\tag{1.29}
$$

with $\boldsymbol{\phi}(n)$ and $\hat{\boldsymbol{\phi}}_{MOE}(n)$ as defined in equation (1.10). Finding the gradient of $\mathcal{W}[e_{OE}(n)] = e_{OE}^2(n)$ with respect to the adaptive filter coefficients, one obtains

$$
\begin{aligned}
\boldsymbol{\nabla}_{\hat{\boldsymbol{\theta}}}[e_{OE}^2(n)] &= 2e_{OE}(n)\boldsymbol{\nabla}_{\hat{\boldsymbol{\theta}}}[e_{OE}(n)] \\
&= -2e_{OE}(n)\boldsymbol{\nabla}_{\hat{\boldsymbol{\theta}}}[\hat{y}(n)]
\end{aligned}
\tag{1.30}
$$

which, using the so-called small step approximation[4] [5], [26], [28], yields that

$$
\boldsymbol{\nabla}_{\hat{\boldsymbol{\theta}}}[e_{OE}^2(n)] \approx -2e_{OE}(n)\hat{\boldsymbol{\phi}}_{OE}(n) \tag{1.31}
$$

with

$$
\hat{\boldsymbol{\phi}}_{OE}(n) = \left[-\hat{y}^f(n-1)\ldots-\hat{y}^f(n-n_{\hat{a}})\; x^f(n)\ldots x^f(n-n_{\hat{b}})\right]^T \tag{1.32}
$$

where the superscript ^{f} indicates that the corresponding signal is being preprocessed by the allpole filter $\frac{1}{\hat{A}(q,n)}$. From the previous equations, the steepest-descent form of the OE algorithm is then written as

$$
\hat{\boldsymbol{\theta}}(n+1) = \hat{\boldsymbol{\theta}}(n) + \mu e_{OE}(n)\hat{\boldsymbol{\phi}}_{OE}(n) \tag{1.33}
$$

The stationary convergence properties of the OE algorithm are usually characterized with respect to the mean squared output error (MSOE) performance surface defined by $\mathcal{W}[e(n)] = E[e_{OE}^2(n)]$. This function, assuming the input and perturbation signals to be statistically independent with zero means, is characterized by the following properties.

Property 1.7 [2], [75]: The stationary points of the MSOE performance surface are given by

$$
E\left\{\left\{\left[\frac{\hat{A}(q,n)B(q) - A(q)\hat{B}(q,n)}{A(q)\hat{A}(q,n)}\right]\{x(n)\}\right\}\cdot\left\{\left[\frac{\hat{B}(q,n)}{\hat{A}^2(q,n)}\right]\{x(n-i)\}\right\}\right\} = 0 \tag{1.34a}
$$

$$
E\left\{\left\{\left[\frac{\hat{A}(q,n)B(q) - A(q)\hat{B}(q,n)}{A(q)\hat{A}(q,n)}\right]\{x(n)\}\right\}\cdot\left\{\left[\frac{-1}{\hat{A}(q,n)}\right]\{x(n-j)\}\right\}\right\} = 0 \tag{1.34b}
$$

[4] See Appendix A.

for $i = 1, \ldots, n_{\hat{a}}$ and $j = 0, \ldots, n_{\hat{b}}$. In practice, only the stationary points that result in a stable adaptive filter are of interest. These points are commonly referred to as equilibrium points and they are classified as [53]:

Degenerated points: These are the equilibrium points where

$$\hat{B}(q,n) \equiv 0; \text{ if } n_{\hat{b}} < n_{\hat{a}} \tag{1.35a}$$

$$\hat{B}(q,n) \equiv \hat{A}(q,n)L(q), \text{ if } n_{\hat{b}} \geq n_{\hat{a}} \tag{1.35b}$$

with $L(q) = 1 + l_1 q^{-1} + \ldots + l_{n_l} q^{-n_l}$.

Non-degenerated points: All the equilibrium points that are not degenerated.

□

The following properties define how the equilibrium points characterize the MSOE performance surface associated to the OE adaptation algorithm.

Property 1.8 [2], [75]: If $n^* \geq 0$, all global minima of the MSOE performance surface are of the form given in equation (1.28). This means that in all cases of sufficient order identification, the global minimum solutions of the OE algorithm will include the polynomials of the unknown system plus a common factor $L(q)$ present in the numerator and denominator polynomials of the adaptive filter.

□

Property 1.9 [77]: If $n^* \geq 0$ and the input signal $x(n)$ is persistently exciting of order $max[(n_{\hat{a}}+n_b);(n_{\hat{b}}+n_a)]$, then all equilibrium points that satisfy the strictly-positive-realness condition

$$Re\left[\frac{\hat{A}^*(z)}{A(z)}\right] > 0; \quad \forall\, |z| = 1 \tag{1.36}$$

are global minima of the form given in equation (1.28).

□

This theorem is essential to the characterization of the stationary properties of the composite squared error algorithm to be presented later and it will be discussed in detail in Chapter 2.

Property 1.10 [77]: Let the input signal $x(n)$ be generated as

$$x(n) = \left[\frac{F(q)}{G(q)}\right]\{w(n)\} \tag{1.37}$$

where $F(q) = \sum_{k=0}^{n_f} f_k q^{-k}$ and $G(q) = 1 + \sum_{k=1}^{n_g} g_k q^{-k}$ are coprime polynomials and $w(n)$ is a white noise signal. Then, if

$$n^* \geq n_f \tag{1.38a}$$

$$n_{\hat{b}} - n_a + 1 \geq n_g \tag{1.38b}$$

all equilibrium points of the OE algorithm are global minima of the form given in equation (1.28).

□

This latter property is indeed the most general result about the unimodality of the MSOE performance surface in cases of sufficient order identification and it has a very important special case.

Corollary 1.11 [77]: If $x(n)$ is a white noise, the orders of the adaptive filter are strictly sufficient, such that $n_{\hat{a}} = n_a$ and $n_{\hat{b}} = n_b$, and if $n_{\hat{b}} - n_a + 1 \geq 0$, then the MSOE function has one single equilibrium point, which is the global minimum of the form

$$\hat{A}^*(q) = A(q) \tag{1.39a}$$

$$\hat{B}^*(q) = B(q) \tag{1.39b}$$

□

The case analyzed by this last statement was further investigated by Nayeri in [52], who obtained a less restrictive sufficient condition to guarantee unimodality of the OE algorithm when the input signal is a white noise and the orders of the adaptive filter exactly match the unknown system. This result is given by the property below.

Property 1.12 [52]: If $x(n)$ is a white noise, the orders of the adaptive filter are strictly sufficient, such that $n_{\hat{a}} = n_a$ and $n_{\hat{b}} = n_b$, and if $n_{\hat{b}} - n_a + 2 \geq 0$, then there is only one equilibrium point, which is the global minimum of the form given in equation (1.39).

□

Fan and Nayeri [15] showed that this last condition is the least restrictive sufficient condition of this form that assures unimodality of the adaptive process for the corresponding adaptive system identification case. In fact, a numerical counterexample was presented in [15] for the case $n_{\hat{b}} - n_a + 3 \geq 0$. Another important result associated to the OE algorithm is given below.

Property 1.13 [53]: All degenerated equilibrium points are saddle points and their existence implies multimodality of the performance surface if either $n_{\hat{a}} > n_{\hat{b}} = 0$ or $n_{\hat{a}} = 1$.

□

Notice that this last property is independent of the value of n^* and, as a consequence, is also valid for insufficient order cases.

Another interesting statement related to the OE algorithm was made by Stearns [83] in 1981. In fact, it was conjectured that *if* $n^* \geq 0$ *and* $x(n)$ *is a white noise input signal, then the performance surface defined by the MSOE objective function is unimodal.* This conjecture was supported by several numerical examples and it was considered valid until 1989, when Fan and Nayeri published a numerical counterexample for it in [15].

Basically, the most important characteristics of the OE algorithm are the possible existence of multiple local minima and the assured existence of an unbiased global minimum solution, even in presence of perturbation noise in the unknown system output signal. Other important aspect of the OE algorithm is the stability checking requirement during the adaptive process. A practical alternative to avoid extensive computations related to the filter stability check is the use of an alternative filter realization other than the direct-form structure, as it will be discussed later in Chapter 4.

1.3.3 Other Adaptive IIR Filter Algorithms

Besides the EE and OE algorithms, other IIR algorithms that are definitely worth mentioning are the following.

The Steiglitz-McBride (SM) method was introduced in [84] and later an on-line version of it was introduced in [12] by Fan and Jenkins. The central idea behind the SM approach is the intent of combining all the good properties of the EE and OE algorithms. For that purpose, a new error signal was introduced as

$$e_{SM}(n) = \left[\frac{\hat{A}(q,n)}{\hat{A}(q,n-1)}\right]\{e_{OE}(n)\} \tag{1.40}$$

Due to the linear relationship of this signal with respect to the adaptive filter coefficients at time n, the SM scheme was expected to yield a single solution independent of the initial values of the adaptive filter coefficients. In addition, as the SM error signal tends to resemble the OE signal as the adaptive process converges, it can be inferred from the definition of $e_{SM}(n)$ that the SM solution will be also expected to be identical to the MSOE global solution. These assumptions were confirmed for the cases of sufficient order identification when the perturbation noise was a white signal [86]. However, in cases of insufficient order identification, it was verified in [16], [17], [78], [86] that the characterization of the SM convergence process is not easy and it is possible that biased or multiple solutions exist. More recently [7], [58], some attempts have been made to associate the SM approach to a time-varying performance surface, thus giving a better physical meaning to the convergence process of this algorithm. Preliminary analyses using these methods, however, are somewhat inconclusive and further research is yet to be performed for any practical result be achieved.

Another significant algorithm is the so-called simplified hyperstable adaptive recursive filtering (SHARF) algorithm presented in [39] as an extension of Landau's work in [38]. The SHARF algorithm is based on the signal

$$e_{SHARF}(n) = D(q)\,\{e_{OE}(n)\} \tag{1.41}$$

with $D(q) = 1 + d_1(n)q^{-1} + \ldots + d_{n_d}(n)q^{-n_d}$ being chosen by the system designer. The SHARF algorithm is known to be unimodal and unbiased with respect to the MSOE global minimum in cases of sufficient-order identification, when a specific positive-realness condition is satisfied by the adaptive filter polynomials [39]. The main problem for the implementation of the SHARF algorithm, however, is the lack of a robust practical procedure to determine the additional FIR processing $D(q)$ in order to satisfy the aforementioned convergence condition.

Finally, one must mention the composite regressor algorithm presented by Kenney and Rohrs in [36]. This algorithm makes use of an explicit composition of two other individual algorithms in order to combine their respective properties in a single approach. The same method is also utilized in the next chapter to generate a new algorithm combining the EE and OE algorithms.

For the sake of brevity, a more complete study of adaptive IIR algorithms was not included in this dissertation and the interested reader is referred to [60] which includes all the above listed schemes and also the modified output error algorithm [18], the bias-remedy equation error algorithm [40], and the composite error algorithm [56].

1.4 Thesis Organization and Contributions

In this chapter, the basic material necessary for the understanding of the remaining chapters was introduced. In fact, here a structured presentation of the field of adaptive filtering was given, dividing the area into the topics of applications, filter realizations, and algorithms. Also, the system identification application and the direct-form realization were briefly introduced as they will constitute the basic environment for most of all subsequent analyses. Finally, a discussion of some concepts associated to adaptation algorithms was included leading to the presentation of two important adaptive IIR algorithms, namely the equation error (EE) and the output error (OE) algorithms. The organization of the rest of this dissertation is as follows.

In Chapter 2, a new adaptation algorithm, the so-called composite squared error (CSE) algorithm, is introduced. This algorithm is based on the idea of combining the good properties of both the EE and OE algorithms. The relationship of the CSE algorithm with the EE and OE algorithms facilitates the steady-state and transient behavior analyses of the CSE algorithm. Based on these analyses, convergence properties of the CSE algorithm, such as the existence of suboptimal or biased solutions, and its stability properties are verified.

In Chapter 3, the implementation of the CSE algorithm using time-varying adaptable parameters in order to obtain more efficient and more robust convergence is discussed. Adaptation parameters for the CSE algorithm include the convergence factor that controls the stability and speed of convergence of the algorithm and the composition factor that mainly dictates the steady-state characteristics of the final solution of the algorithm.

In Chapter 4, alternatives to the direct-form realization for IIR filters are presented. One of the disadvantages of this realization is the complexity of the required real-time stability test. A new lattice implementation for adaptive IIR filters is proposed which can be used with the CSE algorithm as well as with any other adaptive IIR algorithm. The proposed lattice implementation is shown to be equivalent to the direct form such that both approaches present similar transient processes and equivalent sets of stationary points in the sense of realizing identical transfer functions. The lattice structure, however, possess the additional feature of allowing real-time pole monitoring in a simple manner throughout the adaptation process.

In Chapter 5, the implementation of adaptive filtering algorithms for real-time applications is discussed. A noise canceller is implemented on a DSP chip using both FIR and IIR adaptation techniques and their performances are compared. The main goal of this chapter is to illustrate the issues related to the implementation of practical adaptive IIR filters comparing the performance of these systems with well-known standard FIR counterparts.

In Chapter 6, the conclusions are presented, some open issues in the vast area of adaptive IIR filtering are discussed, and possible ways for the extension and continuation of this work are proposed.

1.5 Conclusion

The purpose of this chapter has been to outline some of the issues related to the project of an adaptive IIR filtering system. This has been attempted with the presentation of a structured introduction to the field of adaptive signal processing. Also, two well known adaptive IIR algorithms were introduced following a previously defined unifying framework. In the remaining of this dissertation, using the content of this chapter as a starting point, several adaptive IIR filtering techniques are introduced in attempt to turn this class of systems into a reliable alternative to the adaptive FIR filters.

Chapter 2

The Composite Squared Error Algorithm

2.1 Introduction

In the previous chapter, two commonly used adaptive IIR filter algorithms, namely the equation error (EE) and output error (OE) algorithms, were introduced. One of the best features of these two schemes is the fact that each one can be associated to a mean squared error performance surface [83]. This results in a better understanding of the general convergence characteristics of the respective adaptive technique.

The EE algorithm is a simple adaptation algorithm that generally presents a unimodal mean square equation error (MSEE) performance surface and good stability characteristics. However, the EE algorithm yields a biased solution in the presence of measurement or modelling noise in the desired output signal. On the other hand, the mean square output error (MSOE) performance surface associated to the OE algorithm has an unbiased global minimum when the noise is independent of the input signal. This surface, however, may have suboptimal local minima in cases of insufficient order modelling or when the unimodality condition of Söderström [77], seen in Property 1.10, is not satisfied in cases of sufficient order identification. In cases of strictly sufficient order modelling, the unimodality of the MSOE performance surface is guaranteed if the sufficient condition of Nayeri [55] given in Property 1.12 is satisfied. Comparing the characteristics of the EE and OE algorithms, one concludes that the EE scheme tends to present strong positive qualities during the transient part of the convergence process, while the OE scheme is characterized by the important steady-state fact of possessing an unbiased global minimum.

In this chapter, the composite squared error (CSE) algorithm is introduced. This algorithm attempts to combine the good characteristics of the EE algorithm in the transient part of the convergence process with the good properties of the OE algorithm at steady state. In order to allow a better control of the overall properties of the CSE algorithm the combination of the EE and OE algorithms is made in an explicit form using a composition parameter, as will be seen later. This approach is similar to the one used by Kenney and Rohrs in [36].

In [56], [57] an adaptive IIR filter algorithm was developed using the same motivation as the CSE algorithm. For that algorithm, an auxiliary error signal was defined as a linear combination of the individual EE and OE signals. For the CSE algorithm, however, the corresponding error signal is defined based on the squared values of the EE and OE signals, resulting in a much easier way to determine the resultant performance surface through the direct composition of the MSEE and MSOE functions. This results into simpler analysis of the final convergence properties for the CSE algorithm, as will be verified later in this chapter.

2.2 The Composite Squared Error Algorithm

Consider the adaptive filtering configuration as described in Section 1.2. The basic form of a general adaptation algorithm is given by

$$\hat{\boldsymbol{\theta}}(n+1) = \hat{\boldsymbol{\theta}}(n) + \mu(n)e(n)\hat{\boldsymbol{\phi}}(n) \tag{2.1}$$

where $\hat{\boldsymbol{\theta}}(n)$ is the parameter vector to be updated, $\mu(n)$ is a gain factor that can be a matrix or a scalar, $e(n)$ is an estimation error, and $\hat{\boldsymbol{\phi}}(n)$ is the regressor or information vector associated to the respective adaptation algorithm. As seen in Section 1.3, following this approach the EE algorithm is characterized by

$$e_{EE}(n) = \hat{A}(q,n)\{y(n)\} - \hat{B}(q,n)\{x(n)\} \tag{2.2a}$$

$$\hat{\boldsymbol{\phi}}_{EE}(n) = \left[-y(n-1) \ldots -y(n-n_{\hat{a}}) \;\; x(n) \ldots x(n-n_{\hat{b}})\right]^T \tag{2.2b}$$

and for the OE algorithm, one has

$$e_{OE}(n) = y(n) - \hat{y}(n) \tag{2.3a}$$

$$\hat{\boldsymbol{\phi}}_{OE}(n) = \left[-\hat{y}^f(n-1) \ldots -\hat{y}^f(n-n_{\hat{a}}) \; x^f(n) \ldots x^f(n-n_{\hat{b}})\right]^T \tag{2.3b}$$

Consider a new signal, hereby referred to as the composite squared error (CSE) signal, that explicitly combines the EE and OE quadratic signals in the form

$$e^2_{CSE}(n) = \gamma e^2_{EE}(n) + (1-\gamma)e^2_{OE}(n) + K \tag{2.4}$$

where γ is the composite parameter of the two basic schemes and $K \geq 0$ is a constant that guarantees the right-hand side of the above equation to be nonnegative for a general range of the values of γ. Notice that if the composite parameter is constrained to the interval $\gamma \in [0,1]$, then K can be set to zero.

From (2.4), the mean composite square error (MCSE) performance surface associated to the CSE signal can be directly calculated as

$$E\left[e^2_{CSE}(n)\right] = \gamma E\left[e^2_{EE}(n)\right] + (1-\gamma)E\left[e^2_{OE}(n)\right] + K \tag{2.5}$$

i.e., the MCSE performance surface is obtained as a weighted combination of the MSEE and MSOE surfaces added to a factor $K \geq 0$ that assures the nonnegativity of the MCSE function. Obtaining the updating equation of the adaptation algorithm that is based on the CSE signal using a steepest descent minimization scheme of the form (1.18), one gets

$$\hat{\boldsymbol{\theta}}(n+1) = \hat{\boldsymbol{\theta}}(n) + \mu\left[\gamma e_{EE}(n)\hat{\boldsymbol{\phi}}_{EE}(n) + (1-\gamma)e_{OE}(n)\hat{\boldsymbol{\phi}}_{OE}(n)\right] \tag{2.6}$$

This equation shows that the instantaneous gradient vector is in this case a weighted combination of the instantaneous gradient vectors of the EE and OE adaptation algorithms, as expected due to the definition used for the CSE signal.

From equations (1.16) and (1.17), the quasi-Newton version of the CSE algorithm is obtained as

$$\hat{\boldsymbol{\theta}}(n+1) = \hat{\boldsymbol{\theta}}(n) + \mu\left[\gamma \mathbf{T}_{EE}(n+1)e_{EE}(n)\hat{\boldsymbol{\phi}}_{EE}(n) + (1-\gamma)\mathbf{T}_{OE}(n+1)e_{OE}(n)\hat{\boldsymbol{\phi}}_{OE}(n)\right] \tag{2.7}$$

where

$$\mathbf{T}_{EE}(n+1) = \frac{1}{\lambda}\mathbf{T}_{EE}(n) - \frac{\mu'}{\lambda}\left[\frac{\mathbf{T}_{EE}(n)\hat{\boldsymbol{\phi}}_{EE}(n)\hat{\boldsymbol{\phi}}_{EE}^T(n)\mathbf{T}_{EE}(n)}{\lambda + \mu'\hat{\boldsymbol{\phi}}_{EE}^T(n)\mathbf{T}_{EE}(n)\hat{\boldsymbol{\phi}}_{EE}(n)}\right] \tag{2.8a}$$

$$\mathbf{T}_{OE}(n+1) = \frac{1}{\lambda}\mathbf{T}_{OE}(n) - \frac{\mu'}{\lambda}\left[\frac{\mathbf{T}_{OE}(n)\hat{\boldsymbol{\phi}}_{OE}(n)\hat{\boldsymbol{\phi}}_{OE}^T(n)\mathbf{T}_{OE}(n)}{\lambda + \mu'\hat{\boldsymbol{\phi}}_{OE}^T(n)\mathbf{T}_{OE}(n)\hat{\boldsymbol{\phi}}_{OE}(n)}\right] \tag{2.8b}$$

with $\lambda = 1 - \mu'$ and $\mu' = \mu/2$.

2.3 Steady-State Analysis of the CSE Algorithm

The steady-state performance of the CSE algorithm is completely characterized by the stationary points of the MCSE performance surface described in equation (2.5) in the previous section. These points are the solution of

$$E\left[\gamma e_{EE}(n)\hat{\boldsymbol{\phi}}_{EE}(n) + (1-\gamma)e_{OE}(n)\hat{\boldsymbol{\phi}}_{OE}(n)\right] = \mathbf{0} \tag{2.9}$$

In order to analyze the MCSE stationary points, however, consider first the stationary solutions of the basic EE and OE schemes for which the following results, briefly stated in Chapter 1 and discussed here in more detail, apply.

Property 2.1 [77]: For the OE scheme, if $n^* \geq 0$, the stationary points of the MSOE performance surface such that

$$Re\left[\frac{\hat{A}^*(z)}{A(z)}\right] > 0;\ \forall\ |z| = 1 \tag{2.10}$$

are global minima of the form

$$\hat{A}^*(q) = A(q)L(q) \tag{2.11a}$$

$$\hat{B}^*(q) = B(q)L(q) \tag{2.11b}$$

with $L(q) = 1 + l_1q^{-1} + \ldots + l_{n_l}q^{-n_l}$, assuming that $x(n)$ and $v(n)$ are zero mean statistically independent sequences and that $x(n)$ is persistently exciting of order $\tilde{n} = max[(n_{\hat{a}} + n_b); (n_{\hat{b}} + n_a)]$.

Proof: The stationary points for the OE algorithm are given by

$$E\left[e_{OE}(n)\hat{\boldsymbol{\phi}}_{OE}(n)\right] = \mathbf{0} \tag{2.12}$$

Define the delay polynomial operators $\bar{A}(q)$, $\bar{B}(q)$, and $L(q)$ based on the following relationships

$$\hat{A}(q) = \bar{A}(q)L(q) \tag{2.13a}$$

$$\hat{B}(q) = \bar{B}(q)L(q) \tag{2.13b}$$

with $L(q)$ as above, and $\bar{A}(q)$ and $\bar{B}(q)$ being two relatively prime polynomials with orders respectively equal $n_{\bar{a}} = n_{\hat{a}} - n_l$ and $n_{\bar{b}} = n_{\hat{b}} - n_l$. Developing the terms in the left-hand side of equation (2.12), one can write from (2.3) and (2.13) that

$$\hat{\boldsymbol{\phi}}_{OE}(n) = \left[\left(\frac{-\bar{B}(q)}{\hat{A}(q)\bar{A}(q)}\right)\{[x(n-1)\ldots x(n-n_{\hat{a}})]\}\ \left(\frac{\bar{A}(q)}{\hat{A}(q)\bar{A}(q)}\right)\{[x(n)\ldots x(n-n_{\hat{b}})]\}\right]^T$$

$$= \left(\frac{1}{\hat{A}(q)\bar{A}(q)}\right)\left\{\begin{bmatrix} -\bar{b}_0 x(n-1) - \ldots - \bar{b}_{n_{\bar{b}}} x(n-n_{\bar{b}}-1) \\ \vdots \\ -\bar{b}_0 x(n-n_{\hat{a}}) - \ldots - \bar{b}_{n_{\bar{b}}} x(n-n_{\bar{b}}-n_{\hat{a}}) \\ x(n) + \ldots + \bar{a}_{n_{\bar{a}}} x(n-n_{\bar{a}}) \\ \vdots \\ x(n-n_{\hat{b}}) + \ldots + \bar{a}_{n_{\bar{a}}} x(n-n_{\hat{b}}-n_{\bar{a}}) \end{bmatrix}\right\}$$

$$= \left(\frac{1}{\hat{A}(q)\bar{A}(q)}\right)\left\{\begin{bmatrix} 0 & -\bar{b}_0 & \ldots & -\bar{b}_{n_{\bar{b}}} & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & -\bar{b}_0 & \ldots & \ldots & -\bar{b}_{n_{\bar{b}}} \\ 1 & \bar{a}_1 & \ldots & \bar{a}_{n_{\bar{a}}} & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & 1 & \bar{a}_1 & \ldots & \bar{a}_{n_{\bar{a}}} \end{bmatrix}\begin{bmatrix} x(n) \\ \vdots \\ x(n-n_{\hat{a}}-n_{\hat{b}}+n_l) \end{bmatrix}\right\} \quad (2.14)$$

and also

$$e_{OE}(n) = \left(\frac{1}{A(q)\bar{A}(q)}\right)\{H(q)\{x(n)\}\} + v(n)$$
$$= \left(\frac{1}{A(q)\bar{A}(q)}\right)\left\{[x(n)\ldots x(n-n_h)]\begin{bmatrix} h_0 \\ \vdots \\ h_{n_h} \end{bmatrix}\right\} + v(n) \quad (2.15)$$

where $H(q) = \bar{A}(q)B(q) - A(q)\bar{B}(q) = h_0 + h_1 q^{-1} + \ldots + h_{n_h} q^{-n_h}$, with $n_h = max[(n_{\bar{a}} + n_b); (n_a + n_{\bar{b}})]$.

Thus, defining the $(n_{\hat{a}} + n_{\hat{b}} - n_l + 1) \times (n_h + 1)$ matrix

$$\mathbf{P}_{OE} = E\left[\left(\frac{1}{\hat{A}(q)\bar{A}(q)}\right)\left\{\begin{bmatrix} x(n) \\ \vdots \\ x(n-n_{\hat{a}}-n_{\hat{b}}+n_l) \end{bmatrix}\right\}\left(\frac{1}{A(q)\bar{A}(q)}\right)\{[x(n)\ldots x(n-n_h)]\}\right] \quad (2.16)$$

and using (2.14) and (2.15) in equation (2.12), results in

$$\mathbf{R}^*_{OE}\mathbf{P}^*_{OE}\mathbf{h}^* = \mathbf{0} \quad (2.17)$$

where $\mathbf{R}_{OE}$ is the $(n_{\hat{a}} + n_{\hat{b}} + 1) \times (n_{\hat{a}} + n_{\hat{b}} - n_l + 1)$ coefficient matrix in (2.14), $\mathbf{h}$ is the $(n_h + 1)$-dimensional vector of h coefficients in (2.15), and the superscript asterisk symbol indicates the given point to be a solution of equation (2.12). In equation (2.17), the fact that $x(n)$ and $v(n)$ are considered to be zero mean statistically independent sequences is used.

From the fact that $\bar{A}(q)$ and $\bar{B}(q)$ are two relatively prime polynomials, $\mathbf{R}_{OE}$ is then a Sylvester matrix [35] with linearly independent columns and, consequently, rank equal $(n_{\hat{a}} + n_{\hat{b}} - n_l + 1)$. Hence, the square matrix $\mathbf{R}_{OE}^T \mathbf{R}_{OE}$ is invertible [91] and (2.17) simplifies to

$$\mathbf{P}_{OE}^* \mathbf{h}^* = \mathbf{0} \tag{2.18}$$

which can be rewritten as

$$E\left[\left(\frac{q^{-i}}{\hat{A}^*(q)\bar{A}^*(q)}\right)\{x(n)\}\left(\frac{H^*(q)}{A(q)\bar{A}^*(q)}\right)\{x(n)\}\right] = 0;\ \ 0 \leq i \leq (n_{\hat{a}} + n_{\hat{b}} - n_l) \tag{2.19}$$

For sufficient order identification cases, one has that $n_h \leq (n_{\hat{a}} + n_{\hat{b}} - n_l)$ and then it follows that

$$\begin{aligned} E\left[\left(\frac{H^*(q)}{\hat{A}^*(q)\bar{A}^*(q)}\right)\{x(n)\}\left(\frac{H^*(q)}{A(q)\bar{A}^*(q)}\right)\{x(n)\}\right] &= E\left[r(n)\left(\frac{\hat{A}^*(q)}{A(q)}\right)\{r(n)\}\right] \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} Re\left[\frac{\hat{A}^*(e^{jw})}{A(e^{jw})}\right]\Phi_r(w)\, dw \\ &= 0 \end{aligned} \tag{2.20}$$

with

$$r(n) = \left(\frac{H^*(q)}{\hat{A}^*(q)\bar{A}^*(q)}\right)\{x(n)\} \tag{2.21}$$

and $\Phi_r(w)$ being the spectral density of $r(n)$.

Using the strictly-positive-realness assumption (2.10), one must have that $\Phi_r(w) \equiv 0$ and then $r(n) \equiv 0$. Finally, as the input signal is assumed to be persistently exciting of order $\tilde{n}$, $r(n) \equiv 0$ yields that $H^*(q) = 0$ or equivalently

$$\hat{A}^*(q)B(q) = A(q)\hat{B}^*(q) \tag{2.22}$$

which implies the statement of Property 2.1.

□

Although this proof for Property 2.1 had already been presented by Söderström in [77], its inclusion here was considered essential to obtain a simpler presentation of the following results.

Property 2.2: For the EE scheme, if $n^* \geq 0$, the stationary points of the MSEE performance surface are global minima of the form given in equation (2.11), if the perturbation signal $v(n)$ is inexistent and the input signal is persistently exciting of the order $\tilde{n} = max[(n_{\hat{a}} + n_b); (n_{\hat{b}} + n_a)]$.

Proof: The proof in this case follows similar approach to the one given for Property 2.1. In fact, the stationary points of the EE algorithm are obtained by solving

$$E\left[e_{EE}(n)\hat{\boldsymbol{\phi}}_{EE}(n)\right] = \mathbf{0} \tag{2.23}$$

Using the same delay operator polynomials $\bar{A}(q)$, $\bar{B}(q)$, and $L(q)$ as previously defined for Property 2.1, the terms in this equation can be rewritten from (2.2) and (2.13) as

$$\begin{aligned}
\hat{\boldsymbol{\phi}}_{EE}(n) =& \left[\left(\frac{-B(q)}{A(q)}\right)\{[x(n-1)\ldots x(n-n_{\hat{a}})]\}\ \left(\frac{A(q)}{A(q)}\right)\{[x(n)\ldots x(n-n_{\hat{b}})]\}\right]^T \\
&-[v(n-1)\ldots v(n-n_{\hat{a}})0\ldots 0]^T \\
&= \left(\frac{1}{A(q)}\right)\left\{\left[\begin{array}{c}
-b_0x(n-1)-\ldots-b_{n_b}x(n-n_b-1) \\
\vdots \\
-b_0x(n-n_{\hat{a}})-\ldots-b_{n_b}x(n-n_b-n_{\hat{a}}) \\
x(n)+\ldots+a_{n_a}x(n-n_a) \\
\vdots \\
x(n-n_{\hat{b}})+\ldots+a_{n_a}x(n-n_{\hat{b}}-n_a)
\end{array}\right]\right\} \\
&-[v(n-1)\ldots v(n-n_{\hat{a}})0\ldots 0]^T \\
&= \left(\frac{1}{A(q)}\right)\left\{\left[\begin{array}{cccccccc}
0 & -b_0 & \ldots & -b_{n_b} & 0 & \ldots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
0 & \ldots & 0 & \ldots & -b_0 & \ldots & -b_{n_b} \\
1 & a_1 & \ldots & a_{n_a} & 0 & \ldots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
0 & \ldots & 0 & 1 & a_1 & \ldots & a_{n_a}
\end{array}\right]\left[\begin{array}{c}
x(n) \\
\vdots \\
x(n-n_h-n_l)
\end{array}\right]\right\} \\
&-[v(n-1)\ldots v(n-n_{\hat{a}})0\ldots 0]^T
\end{aligned} \tag{2.24}$$

and also

$$e_{EE}(n) = \left(\frac{L(q)}{A(q)}\right)\{H(q)\{x(n)\}\} + \hat{A}(q)\{v(n)\}$$

$$= \left(\frac{L(q)}{A(q)}\right)\left\{[x(n)\ldots x(n-n_h)]\begin{bmatrix} h_0 \\ \vdots \\ h_{n_h} \end{bmatrix}\right\} + \hat{A}(q)\{v(n)\} \quad (2.25)$$

where $H(q)$ and n_h are given as before for the previous property.

Defining, the $(n_h + n_l + 1)\times(n_h + 1)$ matrix

$$\mathbf{P}_{EE} = E\left[\left(\frac{1}{A(q)}\right)\left\{\begin{bmatrix} x(n) \\ \vdots \\ x(n-n_h-n_l) \end{bmatrix}\right\}\left(\frac{L(q)}{A(q)}\right)\{[x(n)\ldots x(n-n_h)]\}\right] \quad (2.26)$$

and using equations (2.24) and (2.25) in (2.23), results in

$$\mathbf{R}^*_{EE}\mathbf{P}^*_{EE}\mathbf{h}^* - E\left[\mathbf{v}(n)\hat{A}^*(q)\{v(n)\}\right] = \mathbf{0} \quad (2.27)$$

where the superscript asterisk symbol indicates that the given point is a solution of equation (2.23), $\mathbf{R}_{EE}$ is the $(n_{\hat{a}} + n_{\hat{b}} + 1) \times (n_h + n_l + 1)$ coefficient matrix in (2.24), $\mathbf{h}$ is the $(n_h + 1)$-dimensional vector of h coefficients in (2.25), and $\mathbf{v}(n)$ is the vector in equation (2.24) of samples of the noise signal $v(n)$ and $(n_{\hat{b}} + 1)$ null entries. It should be noticed that the sequences $x(n)$ and $v(n)$ are once again here assumed to be statistically independent with $E[v(n)] = 0$.

As $A(q)$ and $B(q)$ are two relatively prime polynomials, the matrix $\mathbf{R}_{EE}$ has linearly independent columns [35] and, consequently, $(\mathbf{R}^T_{EE}\mathbf{R}_{EE})^{-1}$ exists what simplifies (2.27) to

$$\mathbf{P}^*_{EE}\mathbf{h}^* = \mathbf{0} \quad (2.28)$$

which can then be rewritten as

$$E\left[\left(\frac{q^{-i}}{A(q)}\right)\{x(n)\}\left(\frac{L^*(q)H^*(q)}{A(q)}\right)\{x(n)\}\right] = 0;\ 0 \le i \le (n_h + n_l) \quad (2.29)$$

or equivalently

$$E\left[\left(\frac{L^*(q)H^*(q)}{A(q)}\right)\{x(n)\}\left(\frac{L^*(q)H^*(q)}{A(q)}\right)\{x(n)\}\right] = E\left[\left[\left(\frac{L^*(q)H^*(q)}{A(q)}\right)\{x(n)\}\right]^2\right]$$

$$= \frac{1}{2\pi}\int_{-\pi}^{\pi}\Phi_s(w)\, dw$$

$$= 0 \quad (2.30)$$

with

$$s(n) = \left(\frac{L^{*}(q)H^{*}(q)}{A(q)} \right) \{x(n)\} \tag{2.31}$$

Hence, it is valid that $\Phi_s(w) \equiv 0$, implying that $s(n) \equiv 0$ and then $L^*(q)H^*(q) = 0$, or equivalently

$$\hat{A}^{*}(q)B(q) = A(q)\hat{B}^{*}(q) \tag{2.32}$$

leading to the result stated in Property 2.2.

□

Based on the previously included results and on the definition of the CSE scheme, the subsequent result follows.

Property 2.3: The stationary points of the CSE algorithm are given by

$$[\gamma \mathbf{R}_{EE}^{*}\mathbf{P}_{EE}^{*} + (1-\gamma)\mathbf{R}_{OE}^{*}\mathbf{P}_{OE}^{*}]\,\mathbf{h}^{*} - \gamma E\left[\mathbf{v}(n)\hat{A}^{*}(q)\{v(n)\}\right] = \mathbf{0} \tag{2.33}$$

where all terms in this equation are as defined before for the EE and OE algorithms.

Proof: Equation (2.33) follows directly from Properties 2.1 and 2.2, which can be seen as special cases of Property 2.3 when $\gamma = 0$ and $\gamma = 1$, respectively.

□

Corollary 2.4: In a sufficient order identification case with $v(n) \equiv 0$, the solutions of the form (2.11) are stationary points of the CSE algorithm.

Proof: This result immediately follows from the fact that for solutions of the form (2.11), $H^*(q) = \bar{A}^*(q)B(q) - A(q)\bar{B}^*(q)$ becomes identically null and, since $v(n) \equiv 0$, equation (2.33) holds.

□

It must be pointed out that equation (2.33) can not be further simplified in general due to the fact that $\mathbf{R}_{EE}$ and $\mathbf{R}_{OE}$ are two different matrices. However, some conclusions can still be drawn for the MCSE performance surface based on the MSEE and MSOE functions.

Property 2.5: If for a given single point $\hat{\boldsymbol{\theta}}^*$, it is valid that

$$E\left[e_{EE}(n)\hat{\boldsymbol{\phi}}_{EE}(n)\right] = E\left[e_{OE}(n)\hat{\boldsymbol{\phi}}_{OE}(n)\right] = \mathbf{0} \tag{2.34}$$

then this particular point is also the single stationary point of the MCSE algorithm. The converse statement is not always true.

□

Property 2.6: The composition of the MSEE and MSOE functions as described in equation (2.5) for the CSE algorithm can be regarded as a trade-off between obtaining a unique or unbiased solution when the composite parameter varies within the interval $\gamma \in [0, 1]$.

□

This property is clearly illustrated by the system identification case shown below.

Example 2.7: Consider the system identification application depicted in Figure 1.3 with a plant given by

$$H(q) = \frac{0.05 - 0.4q^{-1}}{1 - 0.0003q^{-1} - 0.68915q^{-2}} \tag{2.35}$$

and the adaptive filter defined by $n_{\hat{a}} = 1$ and $n_{\hat{b}} = 0$. Assume also a zero mean, unitary variance, Gaussian noise as input signal $x(n)$ and no perturbation noise being present in the desired output signal. Figure 2.1 shows the MCSE performance surfaces for several values of the composite factor γ. Notice that as γ approaches one, the MCSE function becomes more quadratic and well behaved. On the other hand, when γ is close to zero, the MCSE performance surface becomes multimodal. In this example, the MCSE becomes multimodal for values of γ in the interval $0 \leq \gamma \leq \approx 0.28$ and when $\gamma = 0$ the error function presents a local minimum at $\hat{\boldsymbol{\theta}}^* = [-0.85 \ \ -0.15]^T$ and a global minimum at $\hat{\boldsymbol{\theta}}^* = [0.89 \ 0.21]^T$.

□

Property 2.8: There is always an open interval for the composite factor of the form $\gamma \in (\tilde{\gamma}, 1]$ for which any case of multiple solutions for the OE algorithm is transformed into a unimodal MCSE problem.

□

There is no general method to precisely calculate the value of $\tilde{\gamma}$ that guarantees unimodality of the MCSE function. However, it can be heuristically concluded that this threshold value is a direct function of the relative positions of the MSEE and MSOE minimum points. In fact, the closer the optimal point of the EE algorithm is to the global solution of the OE algorithm, the smaller the value of $\tilde{\gamma}$ must be. Additionally, the deeper the valley of the EE minimum point is with respect to the valleys for the OE algorithm, the closer to zero the value of $\tilde{\gamma}$ should also be. This results from the fact that less weighting of the EE algorithm is necessary to compensate for the multimodality of the MSOE performance surface in cases when the minimum value of the MSEE function is much deeper than the values of the MSOE function at its local minima.

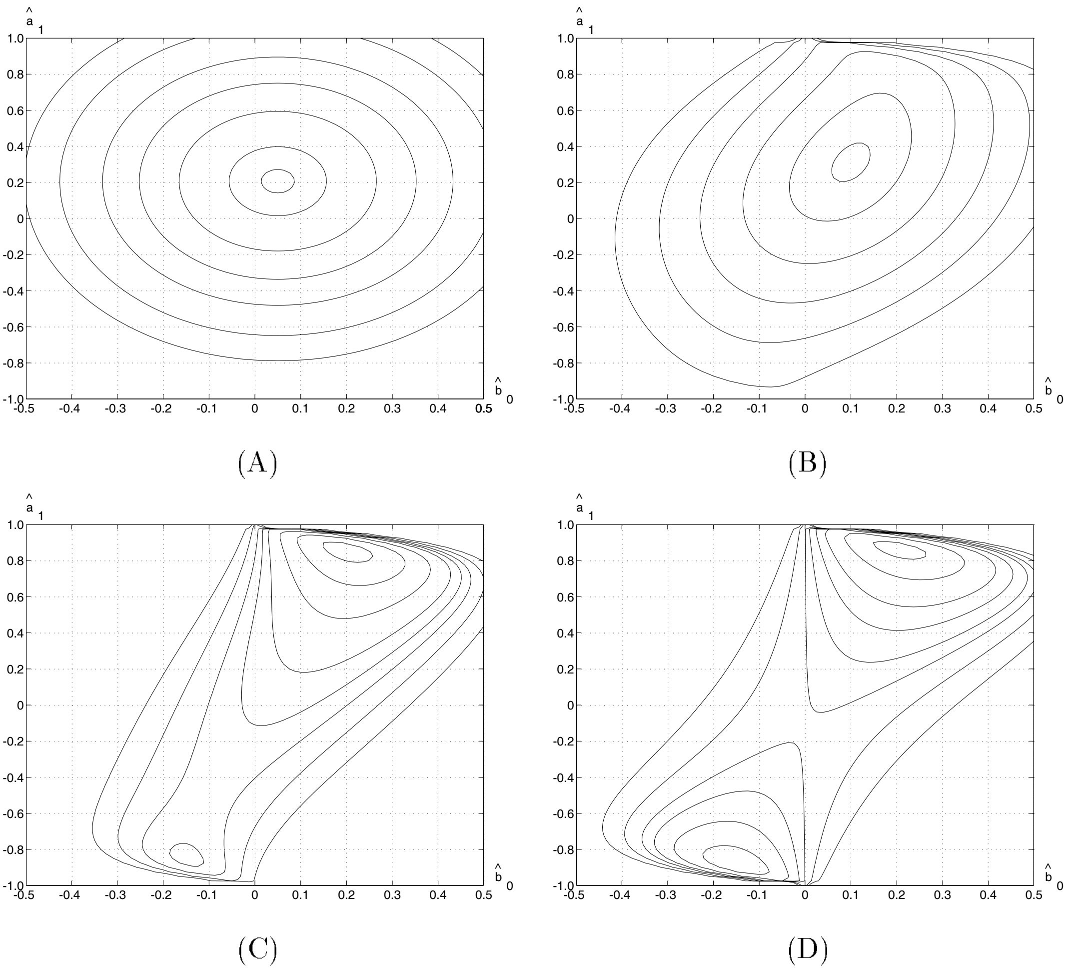


Figure 2.1: Example 2.7 - MCSE performance surfaces: (A) $\gamma = 1.0$; (B) $\gamma = 0.6$; (C) $\gamma = 0.2$; (D) $\gamma = 0.0$.

2.4 Transient Analysis of the CSE Algorithm

In the previous section, the stationary points associated to the CSE adaptive IIR filter algorithm were characterized and general properties of the MCSE performance surface were described. In this section, issues related to the convergence of the CSE algorithm are addressed using two different approaches, namely, the local linearization method and the ordinary difference equation technique.

2.4.1 The Local Linearization Approach

An adaptation algorithm when driven by a nondeterministic input $x(n)$ generates a stochastic trajectory in the parameter space, but its ensemble mean $E[\hat{\boldsymbol{\theta}}(n)]$ over all possible input sequences is a deterministic locus. For the CSE algorithm, one has that

$$E\left[\hat{\boldsymbol{\theta}}(n+1) - \hat{\boldsymbol{\theta}}(n)\right] = \mathbf{f}(\hat{\boldsymbol{\theta}}(n)) \tag{2.36}$$

where

$$\mathbf{f}(\hat{\boldsymbol{\theta}}(n)) = \mu E\left[\gamma e_{EE}(n)\hat{\boldsymbol{\phi}}_{EE}(n) + (1-\gamma)e_{OE}(n)\hat{\boldsymbol{\phi}}_{OE}(n)\right] \tag{2.37}$$

This function $\mathbf{f}(\hat{\boldsymbol{\theta}}(n))$ can be approximated around a stationary point $\hat{\boldsymbol{\theta}}^*$, using the Taylor series for vector functions, by a linear expansion of the form

$$\mathbf{f}(\hat{\boldsymbol{\theta}}(n)) \simeq \frac{\mathrm{d}\mathbf{f}}{\mathrm{d}\hat{\boldsymbol{\theta}}}\Big|_{\hat{\boldsymbol{\theta}}=\hat{\boldsymbol{\theta}}^*} E\left[\hat{\boldsymbol{\theta}}(n) - \hat{\boldsymbol{\theta}}^*\right] \tag{2.38}$$

Using (2.38) in (2.36), results in

$$E\left[\hat{\boldsymbol{\theta}}(n+1) - \hat{\boldsymbol{\theta}}(n)\right] \simeq \mathbf{Q}E\left[\hat{\boldsymbol{\theta}}(n) - \hat{\boldsymbol{\theta}}^*\right] \tag{2.39}$$

where $\mathbf{Q}$ is the "sensitivity" matrix given by

$$\mathbf{Q} = \frac{\mathrm{d}\mathbf{f}}{\mathrm{d}\hat{\boldsymbol{\theta}}}\Big|_{\hat{\boldsymbol{\theta}}=\hat{\boldsymbol{\theta}}^*} = -\frac{\mu}{2}\frac{\mathrm{d}^2 E\left[e^2_{CSE}(n)\right]}{\mathrm{d}\hat{\boldsymbol{\theta}}^2}\Big|_{\hat{\boldsymbol{\theta}}=\hat{\boldsymbol{\theta}}^*} \tag{2.40}$$

Defining the coefficient error vector as $\tilde{\boldsymbol{\theta}}(n) = \hat{\boldsymbol{\theta}}(n) - \hat{\boldsymbol{\theta}}^*$, equation (2.39) can be rewritten as

$$E\left[\tilde{\boldsymbol{\theta}}(n+1)\right] \simeq (\mathbf{I}+\mathbf{Q})\, E\left[\tilde{\boldsymbol{\theta}}(n)\right] \tag{2.41}$$

or equivalently

$$E\left[\tilde{\boldsymbol{\theta}}(n)\right] \simeq (\mathbf{I}+\mathbf{Q})^n\, \tilde{\boldsymbol{\theta}}(0) \tag{2.42}$$

That is, given initial conditions for $\hat{\boldsymbol{\theta}}(n)$ within a small neighborhood of a stationary point $\hat{\boldsymbol{\theta}}^*$, the expected value of the coefficient vector evolves deterministically according to (2.42). Convergence of this equation, however, does require that the matrix $(\mathbf{I}+\mathbf{Q})$ possesses all eigenvalues inside the unit circle. Moreover, it is necessary for the adaptive filter to remain stable throughout the convergence process otherwise the expected values involved in the above procedure are not guaranteed to exist. Now, since the matrix $\mathbf{Q}$ defined as above is symmetric, the eigenvalues of $(\mathbf{I}+\mathbf{Q})$ are all real valued. Thus, one finds that these eigenvalues are smaller than 1, if $\mathbf{Q}$ is negative definite, which implies that the stationary point $\hat{\boldsymbol{\theta}}^*$ is a local minimum of $E[e^2_{CSE}(n)]$. Also, the eigenvalues are all larger than -1, if the matrix $(2\mathbf{I}+\mathbf{Q})$ is positive definite.

Note that such linearization process could be performed at any point in the space of $\hat{\boldsymbol{\theta}}$, but only a linearization around a stationary point yields an autonomous system with no forcing function, as given by equation (2.42). This results from the fact that only at a stationary point the gradient function $\mathbf{f}(\hat{\boldsymbol{\theta}}(n))$ becomes a null vector.

The local linearization method allows one to analyze the convergence of the CSE algorithm close to a limit $\hat{\boldsymbol{\theta}}^*$ in a formal fashion. In fact, using this approach the following local convergence result applies.

Property 2.9: Let $\hat{\boldsymbol{\theta}}^*$ be a local minimum of the MCSE performance surface and assume that the adaptive filter remains stable throughout the adaptation process. Convergence of the CSE algorithm to $\hat{\boldsymbol{\theta}}^*$ is then guaranteed for a sufficiently small value of μ, if the adaptive filter is initialized in a sufficiently small neighborhood of this point.

Proof: Using the linearization approach, its was verified that the adaptation process of the CSE in a small neighborhood of a MCSE stationary point $\hat{\boldsymbol{\theta}}^*$ can be modelled by a process of the form (2.42). Assuming that $\hat{\boldsymbol{\theta}}^*$ is a local minimum of the MCSE function, one has that the symmetric matrix $\mathbf{Q}$ defined in (2.40) is negative definite and then $(\mathbf{I}+\mathbf{Q})$ has all eigenvalues smaller than one. In addition, using the continuity concept, it can be verified that there is always an open interval for $\mu \in [0,\ \tilde{\mu})$ that guarantees the matrix $(2\mathbf{I}+\mathbf{Q})$ to be positive definite. In that case, it is valid that $-1 < \lambda[(\mathbf{I}+\mathbf{Q})] < 1$, where $\lambda[.]$ represents the set of eigenvalues of a particular matrix, and then equation (2.42) converges to zero, implying that $\hat{\boldsymbol{\theta}}(n)$ converges to $\hat{\boldsymbol{\theta}}^*$.

□

To examine the adaptation process analyzed by the local linearization approach, consider the following example.

Example 2.10: Consider an identification case where $n_a = n_{\hat{a}} = 1$, $n_b = n_{\hat{b}} = 0$, and the perturbation signal $v(n)$ is identically null. Then

$$y(n) = b_0 x(n) - a_1 y(n-1) \tag{2.43a}$$

$$\hat{y}(n) = \hat{b}_0(n) x(n) - \hat{a}_1(n) \hat{y}(n-1) \tag{2.43b}$$

and the mean update vector for the CSE adaptation algorithm is given by

$$E\left[\hat{\boldsymbol{\theta}}(n+1)-\hat{\boldsymbol{\theta}}(n)\right]=\mu E\left[\begin{array}{c}-\gamma e_{EE}(n)y(n-1)-(1-\gamma)e_{OE}(n)\hat{y}^f(n-1)\\ \gamma e_{EE}(n)x(n)+(1-\gamma)e_{OE}(n)x^f(n)\end{array}\right] \quad (2.44)$$

Assuming the input signal to be a Gaussian noise with zero mean and unitary variance and μ to be small enough to validate the small step approximation[1], equation (2.44) becomes

$$E\left[\hat{\boldsymbol{\theta}}(n+1)-\hat{\boldsymbol{\theta}}(n)\right]=\mu E\left[\begin{array}{c}\gamma\frac{(a_1-\hat{a}_1(n))b_0^2}{(1-a_1^2)}+(1-\gamma)\left[\frac{a_1b_0\hat{b}_0(n)}{(1-a_1\hat{a}_1(n))^2}-\frac{\hat{a}_1(n)\hat{b}_0^2(n)}{(1-\hat{a}_1^2(n))^2}\right]\\ \gamma(b_0-\hat{b}_0(n))+(1-\gamma)\left[\frac{b_0}{(1-a_1\hat{a}_1(n))}-\frac{\hat{b}_0(n)}{(1-\hat{a}_1^2(n))}\right]\end{array}\right] \quad (2.45)$$

Linearization of this function using the Taylor series requires its partial differentiation with respect to $\hat{a}_1(n)$ and $\hat{b}_0(n)$ and subsequent evaluation of the result at $\hat{\boldsymbol{\theta}}^*=[a_1\ b_0]^T$, yielding expressions similar to (2.41) and (2.42) with

$$\mathbf{Q}=\mu\left[\begin{array}{cc}\frac{(3a_1^2\gamma-a_1^4\gamma-a_1^2-1)b_0^2}{(1-a_1^2)^3} & -(1-\gamma)\frac{a_1b_0}{(1-a_1^2)^2}\\ -(1-\gamma)\frac{a_1b_0}{(1-a_1^2)^2} & \frac{(a_1^2\gamma-1)}{(1-a_1^2)}\end{array}\right] \quad (2.46)$$

To illustrate the validity of these results, several parameter error trajectories were calculated for $a_1=0.7$, $b_0=0.5$, $\mu=0.002$, three different values of the composite parameter γ, and several initial conditions for the adaptive filter. The predicted trajectories using equation (2.42) are presented on the left-hand side of Figure 2.2 and the actual trajectories when the CSE algorithm is used are shown in the right-hand side of this figure. The crosses on each trajectory indicate the corresponding progress at every 200 of the first 1000 iterations. From this figure, it is easy to verify that the proposed local analysis complies well with the obtained actual results when the initial values of $\hat{\boldsymbol{\theta}}(n)$ are chosen sufficiently close to the particular stationary point.

□

2.4.2 The Ordinary Difference Equation Approach

In this section a broader transient analysis tool is presented for the CSE algorithm in the sense that it is valid for the entire coefficient space, as opposed to only a neighborhood of a given stationary point. The central idea is the association of the stochastic CSE adaptation algorithm to a deterministic ordinary difference equation (ODE) the convergence of which is easier to be analyzed. The contents of this section follow the approach of Fan in [13], [14]

[1] See Appendix A

which complements the work of Benveniste extending the work of Ljung in [41]. The main result included here is expressed as follows.

Property 2.11: Let $x(n)$ and $v(n)$ be stationary processes with finite first, second, and fourth moments. Assume that the denominator polynomial $\hat{A}(q,n)$ describes a stable filter for all n and that $x(n)$ and $v(n)$ are ϕ-mixing, as defined in Chapter 1. Then, the behavior of the CSE algorithm with constant μ and γ converges to the solution of the ODE

$$\frac{\mathrm{d}\hat{\mathbf{\Theta}}(t)}{\mathrm{d}t} = V\left[\hat{\mathbf{\Theta}}(t)\right];\ \hat{\mathbf{\Theta}}(0) = \hat{\mathbf{\Theta}}_0 \tag{2.47}$$

where

$$V\left[\hat{\mathbf{\Theta}}(t)\right] = E\left[\gamma e_{EE}(n)\hat{\boldsymbol{\phi}}_{EE}(n) + (1-\gamma)e_{OE}(n)\hat{\boldsymbol{\phi}}_{OE}(n)\right]\Big|_{\hat{\boldsymbol{\theta}}=\hat{\mathbf{\Theta}}(t)} \tag{2.48}$$

in probability, such that

$$P\left\{\sup_{0\le n\tau\le S}\|\hat{\boldsymbol{\theta}}(n) - \hat{\mathbf{\Theta}}^*(n\tau)\| > C\varepsilon(\tau)\right\} < C'\varepsilon(\tau) \tag{2.49}$$

where S, C, and C' are positive constants and $\varepsilon(\tau)$ is a positive function going to zero as τ decreases.

Proof: The proof of this result follows directly from the work of Fan in [13], [14] that shows that the EE and OE algorithms can each be associated to a corresponding ODE. In that manner, the association of the CSE algorithm to the ODE in (2.47) and (2.48) follows from a direct linear combination of the individual ODEs for the EE and OE schemes.

□

Notice that several common restrictions to the adaptive problem are not imposed in this theorem as, for instance, adaptive filter order constraints and statistical independence of the $x(n)$ and $v(n)$ sequences, thus giving a quite general character to this result. Such assumptions, however, do become necessary when considering the convergence properties of the resulting ODE.

The ϕ-mixing condition of the theorem implies that $x(n)$ and $v(n)$ are uncorrelated to each other for large time separations as indicated before. Moreover, as one of the restrictions in this theorem is the requirement that the adaptive filter has a stable denominator polynomial at each and every instant of time, a stabilization procedure must be incorporated to the algorithm. This is normally accomplished by using alternative filter realizations as opposed to the standard direct-form structure to implement the adaptive IIR filter algorithm. The use of lattice-type structures to implement the CSE algorithm, resulting in the required stability monitoring procedure being efficiently implemented in real time, is discussed in Chapter 4 of this dissertation.

In equation (2.49), $\sup_{0\le n\tau\le S} \|\hat{\boldsymbol{\theta}}(n)-\hat{\boldsymbol{\Theta}}^*(n\tau)\|$ gives a measure of how bad the approximation of $\hat{\boldsymbol{\theta}}(n)$ by $\hat{\boldsymbol{\Theta}}^*(n\tau)$ is. The greater the quality of the approximation, the smaller the value of $\sup_{0\le n\tau\le S} \|\hat{\boldsymbol{\theta}}(n)-\hat{\boldsymbol{\Theta}}^*(n\tau)\|$ should be and vice-versa. Equation (2.49) then implies that the probability of $\sup_{0\le n\tau\le S} \|\hat{\boldsymbol{\theta}}(n)-\hat{\boldsymbol{\Theta}}^*(n\tau)\|$ being greater than a certain value $C\varepsilon(\tau)$ approximates zero as τ converges to zero, implying that a good approximation has been obtained. Note, however, that nothing is said about how small τ must be for the probability given in (2.49) to be satisfactory. In fact, although an explicit form of $\varepsilon(\tau)$ can be obtained with further assumptions, no explicit expressions are given for this function, C, and C'. In general, $\varepsilon(\tau)$, C, and C' are application dependent, such that τ is chosen by trading off the convergence speed with variance of the converged coefficients.

The association of the on-line adaptation algorithm to an ODE allows one to analyze the convergence characteristics of such algorithm in a deterministic manner, greatly simplifying much of the algebraic manipulation involved in such analytic process. To emphasize this distinction, $\hat{\boldsymbol{\Theta}}$ is used here to denote the deterministic counterpart of the stochastic variable $\hat{\boldsymbol{\theta}}$. The following theorem makes use of the ODE approach to analyze the convergence of the CSE algorithm.

Property 2.12: The CSE algorithm always converges to a local minimum of the function $W[\hat{\boldsymbol{\Theta}}(t)] = E[e^2_{CSE}(n)] \mid_{\hat{\boldsymbol{\theta}}=\hat{\boldsymbol{\Theta}}(t)}$, the stationary points of which were studied in Section 2.3.

Proof: Obtaining the time derivative of the Lyapunov function $W[\hat{\boldsymbol{\Theta}}(t)]$ above defined, one obtains

$$\begin{aligned}
\frac{\mathrm{d}W\left[\hat{\boldsymbol{\Theta}}(t)\right]}{\mathrm{d}t} &= -2E\left[\gamma e_{EE}(n)\hat{\boldsymbol{\phi}}_{EE}(n)+(1-\gamma)e_{OE}(n)\hat{\boldsymbol{\phi}}_{OE}(n)\right]^T \mid_{\hat{\boldsymbol{\theta}}=\hat{\boldsymbol{\Theta}}(t)} \frac{\mathrm{d}\hat{\boldsymbol{\Theta}}(t)}{\mathrm{d}t} \\
&= -2E\left[\gamma e_{EE}(n)\hat{\boldsymbol{\phi}}_{EE}(n)+(1-\gamma)e_{OE}(n)\hat{\boldsymbol{\phi}}_{OE}(n)\right]^T \mid_{\hat{\boldsymbol{\theta}}=\hat{\boldsymbol{\Theta}}(t)} \cdot \\
&\quad\ E\left[\gamma e_{EE}(n)\hat{\boldsymbol{\phi}}_{EE}(n)+(1-\gamma)e_{OE}(n)\hat{\boldsymbol{\phi}}_{OE}(n)\right] \mid_{\hat{\boldsymbol{\theta}}=\hat{\boldsymbol{\Theta}}(t)} \\
&\le\ 0
\end{aligned} \tag{2.50}$$

with the equality occurring at the local minimum. Hence, Property 2.12 immediately follows.

□

This theorem is the first result that formally demonstrates the descending nature of the CSE algorithm on a given performance surface as defined above. That result is only valid in the deterministic domain of the variable $\hat{\boldsymbol{\Theta}}$ following the result expressed in Property 2.11. To illustrate the use of the ODE to predict and analyze the convergence behavior of the CSE algorithm, consider the following example.

Example 2.13: Let the plant and the adaptive filter be as defined in Example 2.7. It is straightforward but tedious to show that in this case the CSE algorithm is associated to the first-order ODE given by

$$\frac{\mathrm{d}V\left[\hat{\boldsymbol{\Theta}}(t)\right]}{\mathrm{d}t} = \begin{bmatrix} \gamma f_1\left(\hat{\boldsymbol{\Theta}}(t)\right) + (1-\gamma) f_2\left(\hat{\boldsymbol{\Theta}}(t)\right) \\ g_1\left(\hat{\boldsymbol{\Theta}}(t)\right) \end{bmatrix} \tag{2.51}$$

where

$$f_1\left(\hat{\boldsymbol{\Theta}}(t)\right) = \left[\frac{\left(b_0^2+b_1^2\right)\left(a_1 - \hat{a}_1(t) - a_2\hat{a}_1(t)\right) + b_0 b_1\left(2a_1\hat{a}_1(t) - 1 - a_1^2 + a_2^2\right)}{\left(1-a_2\right)\left(1+2a_2-a_1^2-a_2^2\right)}\right] \tag{2.52a}$$

$$f_2\left(\hat{\boldsymbol{\Theta}}(t)\right) = \left[\frac{\hat{b}_0(t)\left(a_1 b_0 - b_1 - 2a_2 b_0 \hat{a}_1(t) + a_2 b_1 \hat{a}_1^2(t)\right)}{\left(1 - a_1\hat{a}_1(t) + a_2\hat{a}_1^2(t)\right)^2} - \frac{\hat{a}_1(t)\hat{b}_0^2(t)}{\left(1-\hat{a}_1^2(t)\right)^2}\right] \tag{2.52b}$$

$$g_1\left(\hat{\boldsymbol{\Theta}}(t)\right) = \left[\frac{\gamma\hat{a}_1(t)\left(b_1 - a_1 b_0 + a_2 b_0\hat{a}_1(t)\right) + \left(b_0 - b_1\hat{a}_1(t)\right)}{\left(1 - a_1\hat{a}_1(t) + a_2\hat{a}_1^2(t)\right)} - \frac{\hat{b}_0(t)\left(1-\gamma\hat{a}_1^2(t)\right)}{\left(1-\hat{a}_1^2(t)\right)}\right] \tag{2.52c}$$

with $\hat{\boldsymbol{\Theta}}(t) = \left[\hat{a}_1(t)\ \hat{b}_0(t)\right]^T$.

Figures 2.3 and 2.4 respectively show the predicted trajectories using (2.51) and (2.52) and the actual adaptable coefficient trajectories for distinct values of γ and different initial conditions of the adaptive filter indicated by the circles in these figures. For the actual execution of the CSE algorithm, the convergence parameter was set to $\mu = 0.0005$ and the crosses in each trajectory indicate the corresponding progress after every 500 of the first 2500 iterations. The trajectories in these figures are consistent with Figure 2.1 in Example 2.7. Notice, however, that no time comparison between the CSE algorithm trajectories and the ODE trajectories can be made, as opposed to the convergence analysis based on the local linearization approach given in the previous section. From Figures 2.3 and 2.4, it is immediate to verify that the proposed association of a deterministic ODE to the stochastic CSE algorithm complies well with the obtained results, thus resulting in an easier analysis of the CSE convergence process based on the ODE deterministic setting.

□

2.5 Conclusion

An alternative algorithm for adaptive IIR filtering based on the composition of the two well known EE and OE schemes was introduced. The new algorithm, referred to as the CSE algorithm, utilizes an explicit combination factor that allows better control of its convergence characteristics. Steady-state analysis for the new algorithm was included relating its stationary points to the ones of the EE and OE algorithms. Two methods were also discussed to analyze the transient behavior of the CSE algorithm. The first method consisted of a local linearization of the algorithm near a given stationary point. The second approach was based on the association of an ordinary difference equation to the transient process of the CSE algorithm, thus allowing a more general analysis than the previous method. Based on these techniques it was possible to verify that the CSE algorithm is an interesting adaptation tool that manages to take advantage of the positive qualities of the EE and OE schemes allowing a trade off between the unimodality of the performance surface and good stability properties of its corresponding convergence process.

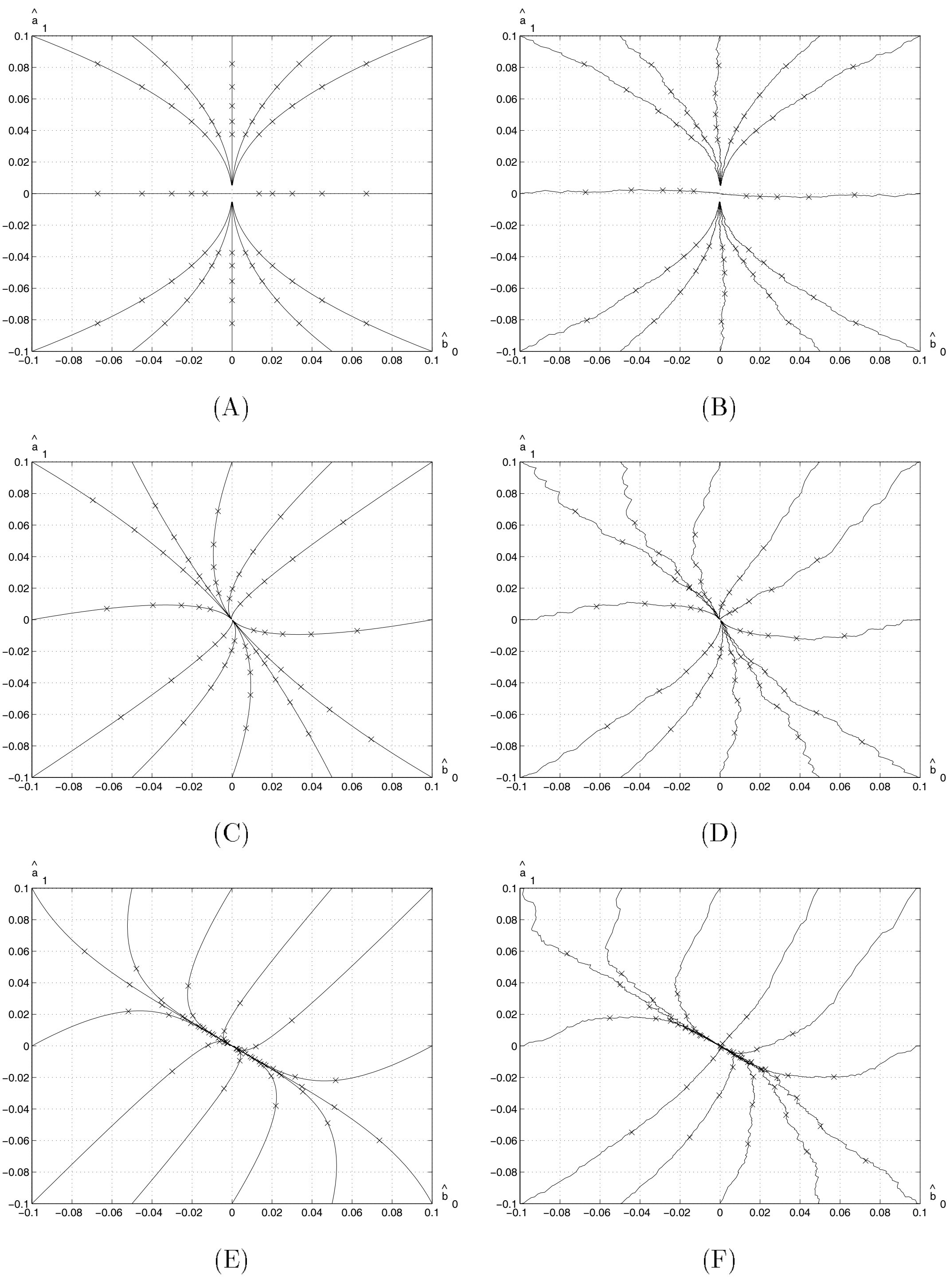


Figure 2.2: Example 2.10 - CSE parameter error trajectories: Predicted (A) and actual (B) for $\gamma = 1.0$; predicted (C) and actual (D) for $\gamma = 0.8$; predicted (E) and actual (F) for $\gamma = 0.0$.

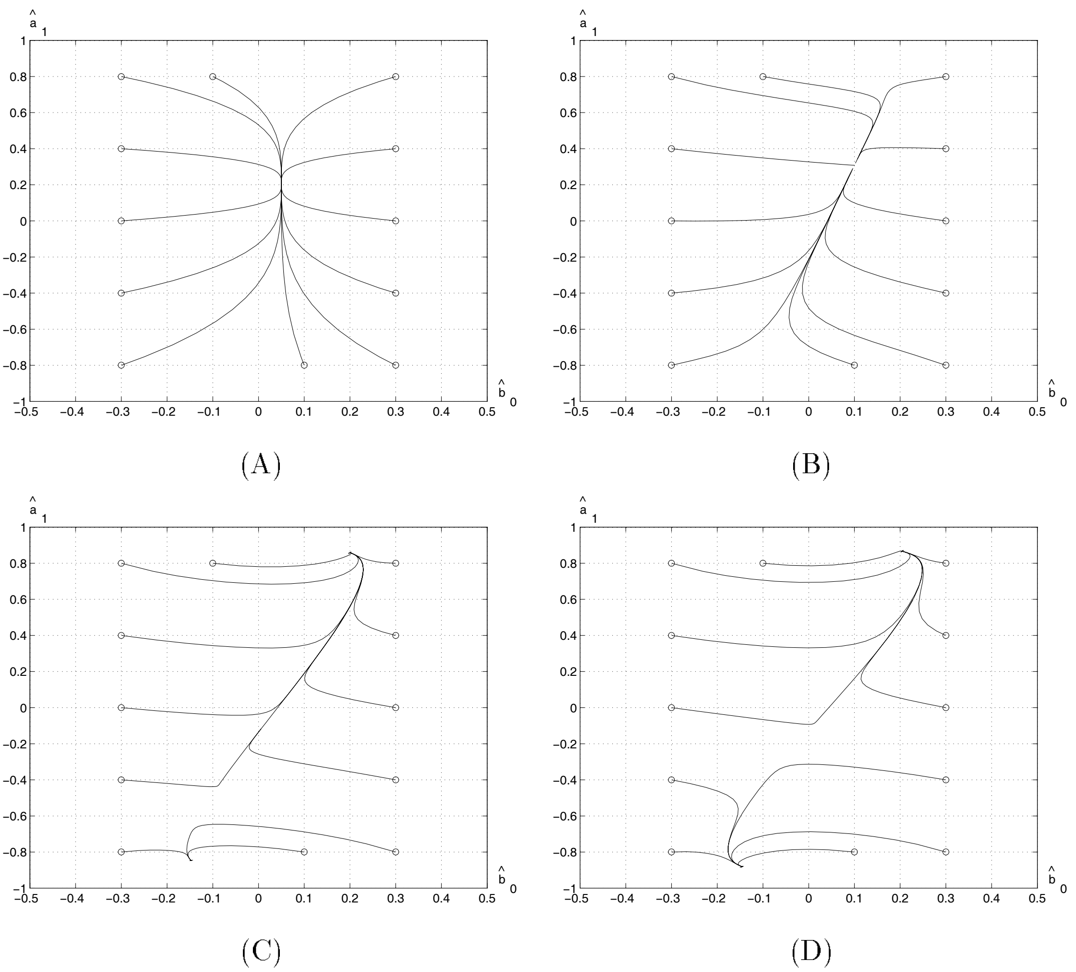


Figure 2.3: Example 2.13 - Predicted CSE parameter trajectories using the ODE method: (A) $\gamma = 1.0$; (B) $\gamma = 0.6$; (C) $\gamma = 0.2$; (D) $\gamma = 0.0$.

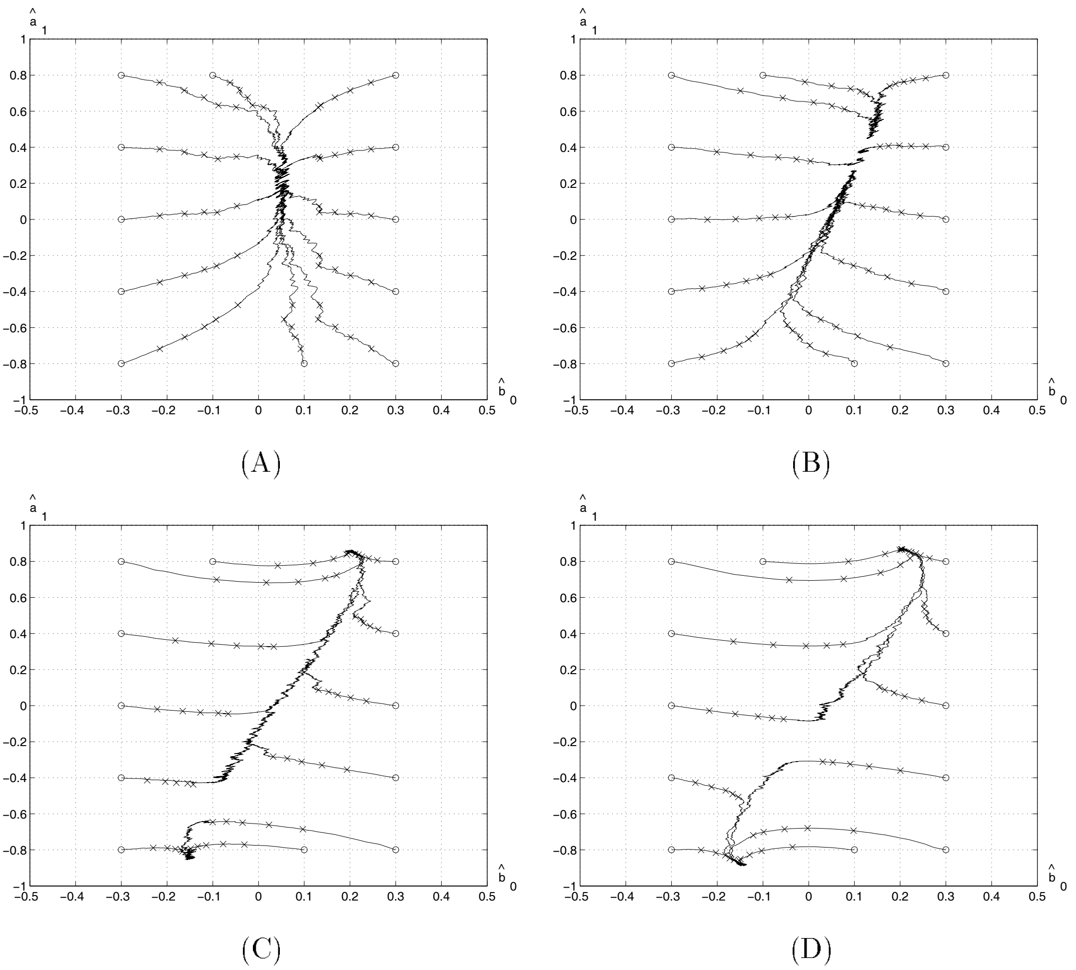


Figure 2.4: Example 2.13 - Actual CSE parameter trajectories: (A) $\gamma = 1.0$; (B) $\gamma = 0.6$; (C) $\gamma = 0.2$; (D) $\gamma = 0.0$.

Chapter 3

Time-Varying Adaptation Parameters for the CSE Algorithm

3.1 Introduction

Practical utilization of the composite squared error (CSE) algorithm introduced in the previous chapter requires the specification of both the convergence parameter μ and the composite factor γ. Choosing the values for such parameters is usually left as an open problem to be addressed by the system designer depending on the application, based on heuristics or other more subjective insights. In this chapter, some guidelines to facilitate the CSE implementation using time-varying adaptation parameters able to adjust their values in order to achieve prescribed objectives are given. The use of variable convergence factors simplifies the algorithm design greatly reducing the amount of heuristics required to achieve a satisfactory performance level with respect to convergence speed. In addition, the use of time-varying composite factor for the CSE algorithm is considered in an attempt to obtain convergence to the global minimum, despite possible existence of local minima for the equation error (EE) algorithm or biased solution for the output error (OE) algorithm. It is then verified that these techniques can be efficiently used to achieve a more robust convergence with the CSE algorithm.

3.2 Variable Convergence Factor for the CSE Algorithm

It is clear from the presentation given in this dissertation that the CSE algorithm is basically composed by two distinct algorithms. Hence, it is natural to conclude that one can utilize a specific convergence factor for each individual algorithm resulting in a modified CSE algorithm of the form

$$\hat{\boldsymbol{\theta}}(n+1) = \hat{\boldsymbol{\theta}}(n) + \left[\mu_{EE}\gamma e_{EE}(n)\hat{\boldsymbol{\phi}}_{EE}(n) + \mu_{OE}(1-\gamma)e_{OE}(n)\hat{\boldsymbol{\phi}}_{OE}(n)\right] \tag{3.1}$$

as opposed to the form given in equation (2.6) with a unique convergence factor. An equivalent modification for the quasi-Newton CSE algorithm given in equation (2.7) yields

$$\begin{aligned}\hat{\boldsymbol{\theta}}(n+1) = \hat{\boldsymbol{\theta}}(n) &+ \mu_{EE}\gamma\mathbf{T}_{EE}(n+1)e_{EE}(n)\hat{\boldsymbol{\phi}}_{EE}(n) \\ &+ \mu_{OE}(1-\gamma)\mathbf{T}_{OE}(n+1)e_{OE}(n)\hat{\boldsymbol{\phi}}_{OE}(n)\end{aligned} \tag{3.2}$$

with

$$\mathbf{T}_{EE}(n+1) = \frac{1}{\lambda_{EE}}\mathbf{T}_{EE}(n) - \frac{\mu'_{EE}}{\lambda_{EE}}\left[\frac{\mathbf{T}_{EE}(n)\hat{\boldsymbol{\phi}}_{EE}(n)\hat{\boldsymbol{\phi}}_{EE}^{T}(n)\mathbf{T}_{EE}(n)}{\lambda_{EE} + \mu'_{EE}\hat{\boldsymbol{\phi}}_{EE}^{T}(n)\mathbf{T}_{EE}(n)\hat{\boldsymbol{\phi}}_{EE}(n)}\right] \tag{3.3a}$$

$$\mathbf{T}_{OE}(n+1) = \frac{1}{\lambda_{OE}}\mathbf{T}_{OE}(n) - \frac{\mu'_{OE}}{\lambda_{OE}}\left[\frac{\mathbf{T}_{OE}(n)\hat{\boldsymbol{\phi}}_{OE}(n)\hat{\boldsymbol{\phi}}_{OE}^{T}(n)\mathbf{T}_{OE}(n)}{\lambda_{OE} + \mu'_{OE}\hat{\boldsymbol{\phi}}_{OE}^{T}(n)\mathbf{T}_{OE}(n)\hat{\boldsymbol{\phi}}_{OE}(n)}\right] \tag{3.3b}$$

with $\lambda_{EE} = 1 - \mu'_{EE}$, $\mu'_{EE} = \mu_{EE}/2$, $\lambda_{OE} = 1 - \mu'_{OE}$, and, $\mu'_{OE} = \mu_{OE}/2$. One can then utilize time-varying convergence factors $\mu_{EE} \equiv \mu_{EE}(n)$ and $\mu_{OE} \equiv \mu_{OE}(n)$ to normalize the CSE adaptation process by forcing these variable parameters to minimize specific criteria associated to each distinct respective algorithm. This approach is discussed in the following sections for both the steepest-descent and quasi-Newton versions of the CSE algorithm. First, however, consider the definitions below.

Definition 3.1: Consider the EE signal given as seen in (1.24) by

$$e_{EE}(n) = y(n) - \hat{\boldsymbol{\theta}}^{T}(n)\hat{\boldsymbol{\phi}}_{EE}(n) \tag{3.4}$$

The so-called a-posteriori EE signal is defined as

$$\begin{aligned}\varepsilon_{EE}(n) &= y(n) - \hat{\boldsymbol{\theta}}^{T}(n+1)\hat{\boldsymbol{\phi}}_{EE}(n) \\ &= e_{EE}(n) - \boldsymbol{\Delta}\hat{\boldsymbol{\theta}}^{T}(n+1)\hat{\boldsymbol{\phi}}_{EE}(n)\end{aligned} \tag{3.5}$$

where $\boldsymbol{\Delta}\hat{\boldsymbol{\theta}}(n+1) = \hat{\boldsymbol{\theta}}(n+1) - \hat{\boldsymbol{\theta}}(n)$.

□

Definition 3.2: Consider the OE signal as in (1.29)

$$e_{OE}(n) = y(n) - \hat{\boldsymbol{\theta}}^T(n)\hat{\boldsymbol{\phi}}_{MOE}(n) \tag{3.6}$$

with $\hat{\boldsymbol{\phi}}_{MOE}(n)$ defined in (1.10d). The so-called a-posteriori OE signal is defined analogously as above by

$$\begin{aligned}\varepsilon_{OE}(n) &= y(n) - \hat{\boldsymbol{\theta}}^T(n+1)\hat{\boldsymbol{\phi}}_{MOE}(n) \\ &= e_{OE}(n) - \boldsymbol{\Delta}\hat{\boldsymbol{\theta}}^T(n+1)\hat{\boldsymbol{\phi}}_{MOE}(n)\end{aligned} \tag{3.7}$$

with $\boldsymbol{\Delta}\hat{\boldsymbol{\theta}}(n+1)$ as before.

□

3.2.1 The Steepest-Descent CSE Case

In this section time-varying convergence factors are given for the steepest-descent (SD) version of the CSE algorithm.

Property 3.3: For the SD-EE algorithm, the time-varying convergence factor that minimizes the a-posteriori squared EE signal is given by

$$\mu^*_{EE}(n) = \frac{1}{\tau_{SD-EE}(n)} \tag{3.8}$$

with $\tau_{SD-EE}(n) = \hat{\boldsymbol{\phi}}^T_{EE}(n)\hat{\boldsymbol{\phi}}_{EE}(n)$.

Proof: From the updating equation for the SD-EE algorithm, (3.1) with $\gamma = 1$, one gets

$$\boldsymbol{\Delta}\hat{\boldsymbol{\theta}}(n+1) = \mu_{EE}(n)e_{EE}(n)\hat{\boldsymbol{\phi}}_{EE}(n) \tag{3.9}$$

and then, from (3.5)

$$\varepsilon^2_{EE}(n) = e^2_{EE}(n)\left[1 - \mu_{EE}(n)\hat{\boldsymbol{\phi}}^T_{EE}(n)\hat{\boldsymbol{\phi}}_{EE}(n)\right]^2 \tag{3.10}$$

Therefore, it is straightforward to show that the value of $\mu_{EE}(n)$ that minimizes the a-posteriori squared EE is given by (3.8).

□

Property 3.4: For the SD-OE algorithm, (3.1) with $\gamma = 0$, the time-varying convergence factor that minimizes the a-posteriori squared OE signal is given by

$$\mu^*_{OE}(n) = \frac{1}{\tau_{SD-OE}(n)} \tag{3.11}$$

with $\tau_{SD-OE}(n) = \hat{\boldsymbol{\phi}}^T_{MOE}(n)\hat{\boldsymbol{\phi}}_{OE}(n)$.

Proof: The proof is entirely analogous to the one given for Property 3.3.

□

Both Properties 3.3 and 3.4 are obtained following a simplified approach to the one seen in [101]. In general, in order to deal with the uncertainties of the generally stochastic adaptive problems, it is common to introduce a safety normalizing parameter $0 \leq \alpha \leq 1$ that scales the aforementioned time-varying convergence factors $\mu^*_{EE}(n)$ and $\mu^*_{OE}(n)$ to avoid instability of the adaptation process. Although this procedure introduces a new parameter to be specified, it has been verified in [11] that the choice for α is not nearly as critical as the choice of the value for a constant-in-time convergence factor μ and hence no essential heuristics is necessary to determine a good value for α. This property is illustrated in the example below.

Example 3.5: Consider the exact-order identification case for the plant used in Example 2.7, i.e.

$$H(q) = \frac{0.05 - 0.4q^{-1}}{1 - 0.0003q^{-1} - 0.68915q^{-2}} \tag{3.12}$$

and

$$\hat{H}(q,n) = \frac{\hat{b}_0(n) + \hat{b}_1(n)q^{-1}}{1 + \hat{a}_1(n)q^{-1} + \hat{a}_2(n)q^{-2}} \tag{3.13}$$

Assume, once again, a zero mean, unitary variance, Gaussian noise as input signal and no perturbation noise being present in the desired output signal. Let also $\gamma = 0.5$ to assure that both EE and OE algorithms are in effect. Consider then the implementation of the SD-CSE algorithm with fixed convergence factor as in (2.6) and with time-varying convergence factors using equation (3.1), where

$$\mu_{EE} \equiv \mu_{EE}(n) = \frac{\alpha}{\tau_{SD-EE}(n)} \tag{3.14a}$$

$$\mu_{OE} \equiv \mu_{OE}(n) = \frac{\alpha}{\tau_{SD-OE}(n)} \tag{3.14b}$$

with $\tau_{SD-EE}(n)$ and $\tau_{SD-OE}(n)$ as defined above. Figure 3.1 depicts the required number of iterations for the CSE signal averaged over 50 experiments to reach a -200dB level for both implementations. From this figure, it can be clearly inferred that a good interval for the value of the fixed convergence factor was found to be $\mu \in [0.06\ 0.15]$ within which the -200dB level for the CSE signal was reached in less than 800 iterations with optimal value $\mu = 0.105$. For the time-varying case, a good interval for the safety parameter was approximately $\alpha \in [0.18\ 0.51]$ with optimal value given by $\alpha = 0.25$. This illustrates that the performance of the SD-CSE algorithm with time-varying convergence factors is less sensitive to the selection of the stepsize than the case with constant factor.

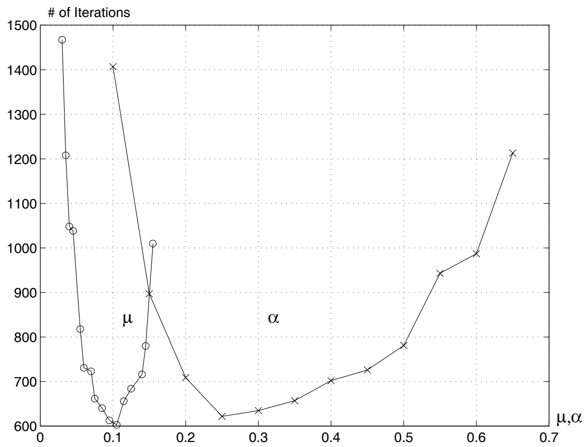


Figure 3.1: Example 3.5 - Number of iterations required for the CSE signal averaged over 50 experiments to reach -200dB as a function of μ in the fixed convergence factor case ('o') and α in the variable convergence factor case ('x'), for the SD-CSE algorithm.

□

It should be pointed out that the whole mechanism inherent to the variable convergence factors can be understood as a normalization of the internal adaptive signals such that the confidence interval for the convergence factor becomes almost independent to the respective adaptive problem. In that way, while the interval for the values of μ that result in good convergence speed is entirely dependent on the particular adaptive problem, the corresponding interval for α is always approximately the same.

3.2.2 The Quasi-Newton CSE Case

In this section time-varying convergence factors are given for the quasi-Newton (QN) version of the CSE algorithm.

Property 3.6: For the QN-EE algorithm, the time-varying convergence factor that minimizes the a-posteriori squared EE signal is given by

$$\mu^*_{EE}(n) = \frac{2}{1 + \tau_{QN-EE}(n)} \tag{3.15}$$

where $\tau_{QN-EE}(n) = \hat{\boldsymbol{\phi}}^T_{EE}(n)\mathbf{T}_{EE}(n)\hat{\boldsymbol{\phi}}_{EE}(n)$.

Proof: From the updating equation for the QN-EE algorithm, (3.2) with $\gamma = 1$, one has that

$$\mathbf{\Delta}\hat{\boldsymbol{\theta}}(n+1) = \mu_{EE}(n) e_{EE}(n) \mathbf{T}_{EE}(n+1) \hat{\boldsymbol{\phi}}_{EE}(n) \tag{3.16}$$

and then, from (3.5)

$$\varepsilon_{EE}^2(n) = e_{EE}^2(n) \left[1 - \mu_{EE}(n) \hat{\boldsymbol{\phi}}_{EE}^T(n) \mathbf{T}_{EE}(n+1) \hat{\boldsymbol{\phi}}_{EE}(n)\right]^2 \tag{3.17}$$

From the recursion (3.3a), it is valid that

$$\hat{\boldsymbol{\phi}}_{EE}^T(n) \mathbf{T}_{EE}(n+1) \hat{\boldsymbol{\phi}}_{EE}(n) = \frac{\tau_{QN-EE}(n)}{1 + \mu'_{EE}(n)\left(\tau_{QN-EE}(n) - 1\right)} \tag{3.18}$$

with $\mu'_{EE}(n) = \mu_{EE}(n)/2$ and $\tau_{QN-EE}(n)$ as defined above. Hence, it is easy to verify that the value of $\mu_{EE}(n)$ that minimizes (3.17) is given by (3.15).

□

Property 3.7 [6]: For the QN-OE algorithm, (3.2) with $\gamma = 1$, the time-varying convergence factor that minimizes the a-posteriori squared OE signal is given by

$$\mu_{OE}^*(n) = \frac{2}{1 + 2\tau'_{QN-OE}(n) - \tau_{QN-OE}(n)} \tag{3.19}$$

with $\tau'_{QN-OE}(n) = \hat{\boldsymbol{\phi}}_{MOE}^T(n) \mathbf{T}_{OE}(n) \hat{\boldsymbol{\phi}}_{OE}(n)$ and $\tau_{QN-OE}(n) = \hat{\boldsymbol{\phi}}_{OE}^T(n) \mathbf{T}_{OE}(n) \hat{\boldsymbol{\phi}}_{OE}(n)$.

Proof: The proof for Property 3.7 can be found in [6] and it is entirely analogous to the one given for Property 3.6.

□

Both of these properties are obtained following a simplified version of the approach in [10] that extends the work of Yassa [101] for the QN case. A safety scaling parameter $0 \leq \alpha \leq 1$ is introduced as in the previous steepest-descent case. This takes the stochastic nature of the adaptive problem into consideration thus avoiding adaptation instability.

Example 3.8: Consider the use of the QN-CSE algorithm for the same identification problem described in Example 3.5. Figure 3.2 depicts the required number of iterations for the CSE signal averaged over 50 experiments to reach a -200dB level using the QN-CSE algorithm with fixed or time-varying convergence factors. Implementation of the QN-CSE algorithm using a fixed convergence factor followed the form given in (2.7) and using time-varying convergence factors was based equation (3.2), where

$$\mu_{EE} \equiv \mu_{EE}(n) = \frac{2\alpha}{1 + \tau_{QN-EE}(n)} \tag{3.20a}$$

$$\mu_{OE} \equiv \mu_{OE}(n) = \frac{2\alpha}{1 + 2\tau'_{QN-OE}(n) - \tau_{QN-OE}(n)} \tag{3.20b}$$

with $\tau_{QN-EE}(n)$, $\tau'_{QN-OE}(n)$, and $\tau_{SD-OE}(n)$ as defined before. It was verified that a relatively good interval for the value of the fixed convergence factor would be in this case $\mu \in [0.06\ 0.32]$ within which the -200dB level for the CSE signal was reached in less than 400 iterations. For the time-varying case, a good interval for the safety parameter was approximately $\alpha \in [0.13\ 0.73]$.

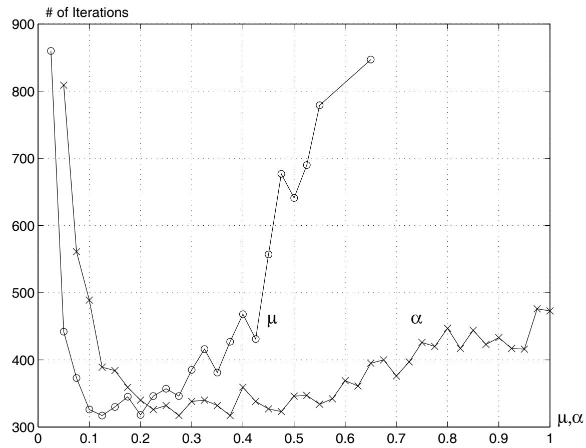


Figure 3.2: Example 3.8 - Number of iterations required for the CSE signal averaged over 50 experiments to reach -200dB as a function of μ in the fixed convergence factor case ('o') and α in the variable convergence factor case ('x'), for the QN-CSE algorithm.

□

Examples 3.5 and 3.8 illustrate how the aforementioned time-varying parameters influence the adaptation process in such a way that satisfactory convergence speeds are achieved with a much lower level of heuristics required to specify the algorithm parameters. Similar results applied to different algorithms and realizations are found, for example, in [11]. In the literature, the use of variable convergence factor in adaptive filtering is commonly associated with a faster adaptation process. This, however, is not generally true as it can be verified from the simulation examples included in this section where the optimal convergence speeds using fixed or variable parameters were found to be quite comparable. Nevertheless, the fact that a more judicious choice of a fixed μ is necessary to achieve a good level of performance (refer, for instance, to Figures 3.1 and 3.2) can be seen as a strong indication that in practice usage of time-varying convergence factors does result in better adaptation speed in general.

3.3 Variable Composite Factor for the CSE algorithm

As mentioned earlier, the EE algorithm has some interesting convergence properties such as overall stability and unique solution. Unfortunately, however, the final solution for this algorithm tends to be biased in the presence of a perturbation signal. On the other hand, the OE algorithm is characterized by possibly unstable adaptation and convergence to suboptimal solutions. However, the global optimum solution of the OE algorithm is proven unbiased even in the presence of any kind of perturbation signal statistically independent of the input signal. Consequently, one may conclude that an ideal kind of adaptation algorithm would be the one that combines the good initial features of the EE algorithm, as good stability properties and unique solution, with the good final property of the OE of unbiased global optimum solution. This can be achieved by using the proposed CSE algorithm with a time-varying composite parameter $\gamma \equiv \gamma(n)$ with value initially set to one and approximating zero as the adaptation process progresses.

Following the above argument, one simple form to utilize an effective time-varying composite factor is based on a recursive updating equation of the form

$$\begin{aligned} \gamma(n+1) &= \gamma(n) - \mu_\gamma \nabla_\gamma \left[e^2_{CSE}(n) \right] \\ &= \gamma(n) - \mu_\gamma \left[e^2_{EE}(n) - e^2_{OE}(n) \right] \end{aligned} \tag{3.21}$$

Notice, however, that as the above scheme is based on the minimization of the mean composite square error, $\gamma(n)$ will tend to one if $E[e^2_{EE}(n)]^* < E[e^2_{OE}(n)]^*$, or zero in case of $E[e^2_{EE}(n)]^* > E[e^2_{OE}(n)]^*$, where $E[e^2_{EE}(n)]^*$ and $E[e^2_{OE}(n)]^*$ are respectively the minimum MSEE and MSOE values. In order to force the composite parameter to tend to zero in all cases, equation (3.21) should be modified to

$$\gamma(n+1) = \begin{cases} \gamma(n) - \mu_\gamma |e^2_{EE}(n) - e^2_{OE}(n)|, \text{ if } \gamma(n+1) > 0 \\ 0, \text{ otherwise} \end{cases} \tag{3.22}$$

This approach has been used in a series of identification problems such as the example below with very positive results.

Example 3.9: Consider the same insufficient order identification problem described in Example 2.12 where the plant is given by

$$H(q) = \frac{0.05 - 0.4q^{-1}}{1 - 0.0003q^{-1} - 0.68915q^{-2}} \tag{3.23}$$

and the adaptive filter is characterized by $n_{\hat{a}} = 1$ and $n_{\hat{b}} = 0$. Assume once again a zero mean, unitary variance Gaussian noise as input signal $x(n)$ and no perturbation noise being present in the reference signal. Applying the CSE algorithm with time-varying composite parameter as described in (3.22) with $\gamma(0) = 1$ and fixed $\mu = \mu_\gamma = 0.002$, results in the coefficient trajectories depicted in Figure 3.3, where the small circles indicate the starting points. From this figure, one must observe that, in this example, convergence to the MSOE global minimum $\hat{\boldsymbol{\theta}}^* = [0.89\ 0.21]^T$ is reached independently of the initial value of the adaptive filter coefficient vector. Compare these trajectories with the ones given in Example 2.13, Figure 2.3(d), where the OE algorithm may converge to a local solution as expected, depending on its starting point. In addition, the fact that in all cases the composite parameter was forced to converge to zero, as indicated in Figure 3.4, assures the CSE final solution to be the exact MSOE global minimum. This is an advantage of the CSE algorithm over others, including the Steiglitz-McBride algorithm, which does not necessarily lead to the MSOE global minimum in the insufficient order cases [17], [58], [78].

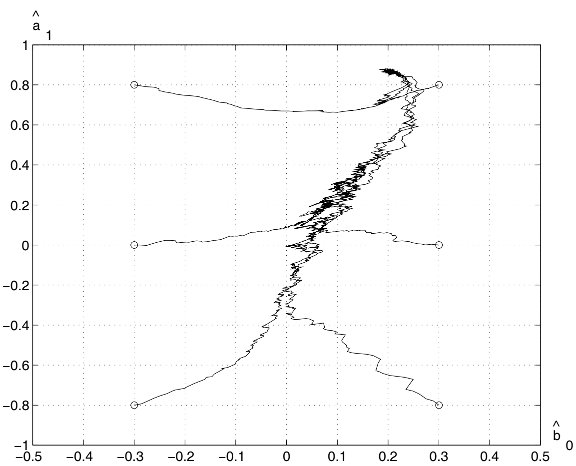


Figure 3.3: Example 3.9 - Convergence in time of adaptive filter coefficients using the CSE algorithm with time-varying composite parameter.

□

In the rest of this section, a geometric interpretation for the CSE algorithm with time-varying composite parameter $\gamma(n)$ and relationship of this algorithm with some off-line nonlinear optimization methods are given.

As seen before, adaptation algorithms are often analyzed in the literature as being iterative procedures that attempt the minimization of a deterministic objective function in a stochastic approximate form [83]. This perspective has proven itself to be quite useful in

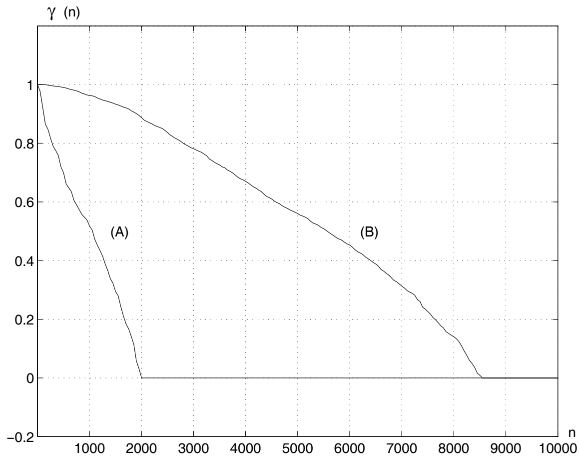


Figure 3.4: Example 3.9 - Convergence in time of the time-varying composite parameter: (A) $\hat{\boldsymbol{\theta}}(0) = [0.8\ 0.3]^T$; (B) $\hat{\boldsymbol{\theta}}(0) = [0.0\ \ -0.3]^T$.

the understanding of many convergence characteristics of a given adaptation algorithm, as observed, for instance, in Chapter 2 of the present dissertation. An alternative point of view, however, interprets an adaptation algorithm as consisting of an exact minimization scheme of a given stochastic time-varying performance surface. This approach is used, for example, in the derivation of an adaptation algorithm, when the instantaneous squared value of a given error signal is used as an objective function, the gradient of which is determined in order to obtain the corresponding iterative equation. This latter perspective, brilliantly presented by Stonick in [90], naturally leads to a simple geometric interpretation of the CSE algorithm with variable composite factor. In fact, as one may have already inferred, using a varying parameter γ with the CSE algorithm can be geometrically visualized as performing the minimization of a time-varying objective function given by the instantaneous value of the CSE signal as defined in (2.4) with $\gamma \equiv \gamma(n)$.

Based on this interpretation, to verify convergence of the CSE algorithm to the MSOE global solution, one must consider the relative position of both the EE single minimum and the OE global minimum. In fact, in cases where both of these minima are located in the same valley of the MCSE objective function, convergence to the MSOE global minimum using the above scheme is assured if the adaptation progresses in such a way that the adaptive filter is directed to the EE solution and it is thence redirected to the OE solution as γ varies from one to zero. On the other hand, for the cases where the EE valley does not coincide with the valley of the MSOE global minimum, more complex strategies could be attempted in order to achieve convergence to the global solution.

An interesting relationship exist between the CSE algorithm implemented with variable composite factor and some off-line nonlinear optimization algorithms belonging to such groups as of the homotopy continuation method (HCM) and the global nonconvexity method (GNM). As described in [89], HCMs are numerical methods that follow paths originating at known minima of a selected simple problem and terminate at the minima of the original nonlinear problem of interest. These paths are defined by the homotopy function that combines the original problem and the simple system of equations by making use of a homotopy parameter which varies between zero and one. A commonly used homotopy function $\mathbf{h}(\hat{\boldsymbol{\theta}}, \tau)$ uses the linear composition of the form

$$\mathbf{h}(\hat{\boldsymbol{\theta}}, \tau) = (1 - \tau)\mathbf{g}(\hat{\boldsymbol{\theta}}) + \tau\mathbf{f}(\hat{\boldsymbol{\theta}}) \tag{3.24}$$

where τ is the homotopy parameter, $\mathbf{g}(\hat{\boldsymbol{\theta}}) = \mathbf{0}$ has known solutions, and $\mathbf{f}(\hat{\boldsymbol{\theta}}) = \mathbf{0}$ is the system of equations to be solved. In the context of adaptive IIR filtering, $\mathbf{f}(\hat{\boldsymbol{\theta}}) = \mathbf{0}$ is defined by setting the MSOE gradient vector equal to zero.

The GNMs are numerical methods that can be seen as special cases of HCMs by defining $\mathbf{g}(\hat{\boldsymbol{\theta}})$ as a convex function [88]. Although, neither HCMs nor GNMs require the deformation from $\mathbf{g}(\hat{\boldsymbol{\theta}})$ to $\mathbf{f}(\hat{\boldsymbol{\theta}})$ to be linear with respect to the composite parameter, the linear form above presented is the one most widely used in practice. In the context of adaptive IIR filtering, the MSEE objective function is quadratic and therefore it can be viewed as a convex approximation of the MSOE function. Hence, one can interpret the proposed CSE adaptation algorithm as an on-line version of a linear GNM such that the convex EE problem is linearly transformed into the OE problem of interest. This relationship between the CSE algorithm and the two nonlinear approaches is interesting as one may attempt to apply some of the convergence results for the HCMs and the GNMs known in the respective literature to the convergence process associated to the CSE algorithm.

In fact, for HCMs it is known that if the system of equations $\mathbf{g}(\hat{\boldsymbol{\theta}}) = \mathbf{0}$ presents the same finite number of solutions as the system $\mathbf{f}(\hat{\boldsymbol{\theta}}) = \mathbf{0}$, there is a theoretical guarantee that the global optimal solution can be found [89]. This result, however, requires the utilization of multiple identifiers working in parallel what can be excessively cumbersome in a computational point of view. For algorithms of the GNM type, global optimality of the final solution is guaranteed when the convex approximation and the intermediate performance surfaces are always below the objective function of interest [88]. Although this is not always valid in our adaptive IIR filtering context, where it is not possible to guarantee the MCSE to be underneath the MSOE for $\gamma \in (0\ 1]$, this result may still provide some additional insight for the task of achieving global optimality.

It must be emphasized, however, that the aforementioned theorems shall only apply to the CSE algorithm convergence process if the variation of the composite factor is performed in such a slow enough fashion that allows for the adaptable coefficient vector to track the intermediary solutions defined by the composite system of equations associated to each value of the parameter γ.

3.4 Conclusion

The use of variable convergence factors and composite parameter was investigated for the CSE algorithm. It was verified that time-varying adaptation parameters can be considered very important to the task of simplifying the design of both the steepest-descent and the quasi-Newton versions of the CSE algorithm with a minimal amount of additional computational burden. Moreover, the use of a time-varying composite factor for the CSE algorithm was considered in order to obtain global convergence despite presence of bias in the EE solution or existence of local minima for the OE algorithm. Overall it was emphatically verified that the use of these techniques result in an efficient way to implement the CSE algorithm greatly simplifying its design.

Chapter 4

Lattice-Based Adaptive IIR Filters

4.1 Introduction

Adaptive IIR filters constitute a potential alternative to adaptive FIR filters as they are suitable for modelling real systems with sharp resonances using significantly fewer coefficients. Standard algorithms for adaptive IIR filters are commonly presented in the literature based on the direct-form realization to obtain a simpler understanding of the nature of the respective algorithm as well as of its convergence properties. The direct-form realization, however, is not suitable for many practical implementations of adaptive filters because it does not allow an efficient on-line pole monitoring as required by several adaptation algorithms to avoid instability of the adaptive IIR filter during the convergence process[1]. Consequently, several alternative structures have been considered for the implementation of adaptive IIR filter algorithms.

The lattice realization [23], [24], [30] is an example of a filter structure the stability

[1]It is a well known fact in the literature [87], [100] that the location of the roots of the adaptive-filter denominator polynomial inside the unit circle is not a necessary neither a sufficient stability condition for time-varying systems. In fact, even if these roots remain inside the unit circle at every instant of time n, it is possible for the system to become unstable. Nevertheless, for most practical cases, it can be shown [20] that if the system is slowly varying in time, a common assumption for adaptive filters known as the small-step approximation discussed in Appendix A, the asymptotic stability of the system is indeed associated to the location of the roots of the denominator polynomial at the interior of the unit circle. Based on these facts, it is a common practice in the area of adaptive filtering to relate the stability of the overall time-varying adaptive filter to the stability of each and every time-invariant linear system defined at each instant of time n.

of which can be assured in real time, making this structure well suited for adaptive IIR filtering. The relationships between the solutions of adaptation algorithms based on the lattice and direct-form realizations were first studied by Nayeri in [54]. In this paper, it was proven that the convergence points of an adaptation algorithm using lattice structure present an one-to-one correspondence with the solutions of the direct-form version of the same algorithm. This property further motivated the idea of using the lattice structure as an efficient and equivalent alternative realization for adaptive IIR filter algorithms.

Initial attempts of applying the lattice structure to adaptive IIR filtering [27], [65] have led to computationally complex adaptation algorithms. Subsequent attempts to simplify [3], [72] and accelerate [68]–[72] lattice-based adaptive IIR filter algorithms presented problems like parameter drift during the convergence process or further increase in the computational complexity of the algorithm.

In [70], a simplified adaptive IIR lattice filter was presented based on recursive-in-order equations for the state variables of the tapped lattice realization. This method was consistent in the sense that it converged to a set of parameters that realized the same transfer function as the corresponding direct-form algorithm. In [67], Regalia introduced computationally efficient lattice versions for the Steiglitz-McBride (SM) and output error (OE) algorithms. These simplified algorithms were obtained by exploiting the modularity of an extended state-space representation for the normalized lattice structure, which enabled Regalia to dramatically simplify the calculation of the information vectors for the SM and OE algorithms. This approach, however, cannot be extended in a straightforward manner to other adaptation algorithms and it is somewhat specific for the normalized lattice structure. More recently, Miao *et al.* [47] introduced new algorithms based on a lattice IIR structure formed by a tapped-delay-line FIR filter in serial connection with a feedback lattice denominator. In this work, the authors developed recursive-in-order equations for that particular structure resulting in the implementation of the adaptation algorithms in a very efficient form.

In this chapter, the results presented in [47], [67], and [70] are complemented. The work here concentrates on the tapped lattice structure used in [70], taking advantage of the inherent characteristics of that realization [23], [24]. Moreover, we obtain recursive-in-order equations for the numerator and denominator polynomials of the adaptive filter transfer function, as opposed to the state-space approach used in [47]. This results into simpler recursion equations and it also allows the implementation of adaptation algorithms from the equation error (EE) family of algorithms as, for example, the standard EE [99], the bias-remedy EE [40], and the composite algorithms of [56] and of Chapter 2 [59]. These facts possibly constitute the major contribution of this chapter. The efficient adaptation algorithms are then obtained using simplifications to the gradient vector following a similar approach to the one proposed in [67].

In here, the two-multiplier and normalized tapped lattice versions of the CSE algorithm are presented. However, the proposed methods are quite general and, to the author's best knowledge, they can be used to implement a consistent and efficient version of any currently known adaptive IIR filter algorithm as it will be later indicated.

This chapter is organized as follows. In the next section, the direct-form EE, OE, and CSE algorithms are given following the general framework for the description of adaptation algorithms. Later, the two-multiplier lattice structure is introduced along with a new technique to implement a given direct-form transfer function. In Section 4.4, using that approach, we present an efficient implementation of the CSE algorithm based on the two-multiplier lattice realization. Additionally, directions are given to extend the method to other adaptation algorithms and to the normalized lattice realization. Computer simulations are included to demonstrate the validity and usefulness of the proposed techniques.

4.2 Direct-Form Adaptive IIR Filter Algorithms

As seen before, a direct-form adaptive filter is generally described by

$$\hat{y}(n) = \left[\frac{\hat{B}(q,n)}{\hat{A}(q,n)}\right]\{x(n)\} \tag{4.1}$$

with $\hat{A}(q,n) = 1 + \sum_{i=1}^{n_{\hat{a}}} \hat{a}_i(n)q^{-i}$ and $\hat{B}(q,n) = \sum_{j=0}^{n_{\hat{b}}} \hat{b}_j(n)q^{-j}$. Also, the basic form of a general adaptive filtering algorithm can be written as

$$\hat{\boldsymbol{\theta}}(n+1) = \hat{\boldsymbol{\theta}}(n) + \mu(n)e(n)\hat{\boldsymbol{\phi}}(n) \tag{4.2}$$

where $\hat{\boldsymbol{\theta}}(n)$ is the adaptive filter coefficient vector which for the direct structure characterized above is defined as[2]

$$\hat{\boldsymbol{\theta}}_d(n) = \left[\hat{a}_1(n) \ldots \hat{a}_{n_{\hat{a}}}(n)\ \hat{b}_0(n) \ldots \hat{b}_{n_{\hat{b}}}(n)\right]^T \tag{4.3}$$

For instance, the EE algorithm is defined by

$$e_{EE}(n) = \hat{A}(q,n)\{y(n)\} - \hat{B}(q,n)\{x(n)\} \tag{4.4a}$$

$$\hat{\boldsymbol{\phi}}_{EE}(n) = \left[-y(n-1) \ldots -y(n-n_{\hat{a}})\ x(n) \ldots x(n-n_{\hat{b}})\right]^T \tag{4.4b}$$

and for the OE algorithm, one has

$$e_{OE}(n) = y(n) - \hat{y}(n) \tag{4.5a}$$

$$\hat{\boldsymbol{\phi}}_{OE}(n) = \left[-\hat{y}^f(n-1) \ldots -\hat{y}^f(n-n_{\hat{a}})\ x^f(n) \ldots x^f(n-n_{\hat{b}})\right]^T \tag{4.5b}$$

As before, the CSE algorithm is readily derived from the EE and OE schemes and it is given by

$$\hat{\boldsymbol{\theta}}(n+1) = \hat{\boldsymbol{\theta}}(n) + \mu(n)\left[\gamma e_{EE}(n)\hat{\boldsymbol{\phi}}_{EE}(n) + (1-\gamma)e_{OE}(n)\hat{\boldsymbol{\phi}}_{OE}(n)\right] \tag{4.6}$$

Later, a different way to implement such algorithms using lattice structures that allow real-time pole monitoring throughout the entire adaptive convergence process will be described.

4.3 The Two-Multiplier Tapped Lattice IIR Realization

As described in [23], a rational transfer function of the form

$$H(z) = \frac{B_N(z)}{A_N(z)} = \frac{b_0 + b_1 z^{-1} + \ldots + b_N z^{-N}}{1 + a_1 z^{-1} + \ldots + a_N z^{-N}} \tag{4.7}$$

can be implemented using an alternative set of parameters $\boldsymbol{\theta}_{2\ell} = [k_1 \ldots k_N\ h_0 \ldots h_N]^T$ obtained from the following set of equations

$$A_{m-1}(z) = \left[A_m(z) - k_m A_m(z^{-1})z^{-m}\right]/(1-k_m^2) \tag{4.8a}$$

$$B_{m-1}(z) = B_m(z) - A_m(z^{-1})z^{-m}h_m;\ m = N, \ldots, 1 \tag{4.8b}$$

[2] The subscripts d and ℓ will be used throughout this chapter to associate a given variable respectively to the direct-form or lattice realizations. More specifically, the subscripts 2ℓ and 4ℓ will refer to the two-multiplier and normalized lattice structures, respectively.

where the polynomials $A_m(z)$ and $B_m(z)$ are defined as

$$A_m(z) = a_{m,0} + a_{m,1}z^{-1} + \ldots + a_{m,m}z^{-m} \tag{4.9a}$$

$$B_m(z) = b_{m,0} + b_{m,1}z^{-1} + \ldots + b_{m,m}z^{-m}; \;\; m = N, \ldots, 1 \tag{4.9b}$$

with $a_{m,0} = 1$. The $\boldsymbol{\theta}_{2\ell}$ coefficients are then obtained as $k_m = a_{m,m}$ and $h_m = b_{m,m}$, for $m = N, \ldots, 1$, and also $h_0 = b_{0,0}$. As it is stressed in [23], it is important to notice that the divisions in (4.8) are always possible, as the k_m parameters will always be of magnitude less than one for a stable transfer function $H(z)$. Using the $\boldsymbol{\theta}_{2\ell}$ coefficients then, the output signal $y(n)$ of the corresponding two-multiplier lattice filter to an input signal $x(n)$ can be calculated by the following set of equations:

$$F_i(n) = F_{i+1}(n) - k_{i+1}G_i(n-1); \;\; i = N-1, \ldots, 0 \tag{4.10a}$$

$$G_j(n) = G_{j-1}(n-1) + k_jF_{j-1}(n); \;\; j = 1, \ldots, N \tag{4.10b}$$

$$y(n) = \sum_{j=0}^{N} h_jG_j(n) \tag{4.10c}$$

with $F_N(n) = x(n)$ and $G_0(n) = F_0(n)$.

An alternative approach to implement the $H(z)$ transfer function still using the $\boldsymbol{\theta}_{2\ell}$ set of coefficients is obtained based on the relationships given in the following lemma.

Lemma 4.1: Consider the $A_m(z)$ and $B_m(z)$ polynomials for $m = N, \ldots, 1$ as given in (4.9) and the parameter vector $\boldsymbol{\theta}_{2\ell}$. Then, the following recursive-in-order equations hold

$$A_m(z) = A_{m-1}(z) + \frac{k_m}{k_{m-1}}\left[A_{m-1}(z) - (1-k_{m-1}^2)A_{m-2}(z)\right]z^{-1} \tag{4.11a}$$

$$B_m(z) = B_{m-1}(z) + \frac{h_m}{k_m}\left[A_m(z) - (1-k_m^2)A_{m-1}(z)\right]; \;\; m = 2, \ldots, N \tag{4.11b}$$

with $A_0(z) = 1$, $A_1(z) = 1 + k_1z^{-1}$, and $B_1(z) = (h_0 + h_1k_1) + h_1z^{-1}$.

□

The proof of this result is obtained from straightforward algebraic manipulation of (4.8) for the filter described by (4.7), whereas the initial conditions for $m = 0, 1$ are obtained by simple algebraic calculations. It is interesting to note that equation (4.11) includes solely causal polynomials while equation (4.8) includes also noncausal terms, implying that the former is better suited for real-time implementations of lattice-based filters. In addition, equation (4.11a) leads to

$$A'_m(z) = A'_{m-1}(z) + \frac{k_m}{k_{m-1}}\left[A_{m-1}(z) - (1-k_{m-1}^2)A_{m-2}(z)\right]z^{-1} \tag{4.12}$$

for the same initial conditions for $A_m(z)$ and values of m as before, where the auxiliary polynomial $A'_m(z)$ is defined as

$$A'_m(z) = A_m(z) - 1 = a_1 z^{-1} + \ldots + a_m z^{-m} \tag{4.13}$$

An important consequence of recursions (4.11b) and (4.12) is that the output $y(n)$ of an IIR filter with transfer function $H(z)$ to an input signal $x(n)$ can then be obtained by rewriting the input-output relationship in the time domain as

$$y(n) = B_N(q)\{x(n)\} - A'_N(q)\{y(n)\} \tag{4.14}$$

To implement this equation with Lemma 4.1, $B_N(q)\,\{x(n)\}$ is obtained recursively using equation (4.11b) with $m = 2, \ldots, N$ and, analogously, $A'_N(q)\,\{y(n)\}$ using equation (4.12). It can be easily seen that the computation effort required to compute the output signal $y(n)$ via the above equation is higher that the one required by the standard form seen in (4.10). In the next section, however, it will be indicated how the recursions given in Lemma 4.1 can be used in the efficient implementation of lattice-based adaptive IIR filter algorithms.

4.4 Efficient Lattice-Based Adaptive IIR Filter Algorithms

In Section 4.2, it was observed that the implementation of an adaptive IIR filter algorithm basically requires the calculation of a residual error signal and an information vector, as depicted in (4.2). These procedures fundamentally consist of processing present or past samples of $x(n)$, $y(n)$, $\hat{y}(n)$ or any other auxiliary signal with the available numerator and denominator polynomials of the adaptive filter. In some cases, as for the algorithms belonging to the EE family, the adaptive filter denominator polynomial may be even required to operate as an all-zero filter to compose the equation error signal, as seen in equation (4.4). In order to derive lattice IIR algorithms then, all that is necessary is to find a possible way to perform any of this additional processing based solely on the coefficients $\hat{\boldsymbol{\theta}}_{2\ell}$. To accomplish that, one must first generalize recursions (4.11b) and (4.12) to the case of time-varying coefficients. This is easily accomplished due to the fact that although these equations are given in Lemma 4.1 for a constant-in-time lattice digital filter, their extension to adaptive filters is valid and natural assuming that the filter coefficients are slowly variant in time. This process of small step approximation is a common assumption for adaptation algorithms and it has been shown to maintain all transient and steady-state properties of an adaptive IIR filter. Using the versions of equations (4.11)-(4.12) with time-varying coefficients, one can then implement an adaptive IIR filter algorithm with the two-multiplier lattice realization in a very general way as it is now demonstrated.

To obtain efficient lattice-based algorithms, it is suggested the implementation of the same updating equations relative to the direct-form version of the algorithm, but in this case using the adaptive lattice set of coefficients $\hat{\boldsymbol{\theta}}_{2\ell}$ through the time-varying extensions of the recursions given in Lemma 4.1. In this way, it is guaranteed that $e_\ell(n) = e_d(n)$ and $\hat{\boldsymbol{\phi}}_\ell(n) = \hat{\boldsymbol{\phi}}_d(n)$, and then an entirely equivalent updating process is obtained with the additional feature of enabling pole-monitoring during the convergence process to avoid instability of the adaptive filter. This fact is formally stated as follows.

Property 4.2: The standard direct-form and the above proposed lattice implementations of any given adaptive IIR filter algorithm are equivalent processes in the sense that these two methods result in sets of stationary points corresponding to the same input-output descriptions of the adaptive filter obtained as solutions of the adaptation process.

Proof: This result was first discussed for a particular algorithm in [67], making use of the fact that the stationary points of an adaptation algorithm are the solutions of the equation:

$$E\left[e(n)\hat{\boldsymbol{\phi}}(n)\right] = \mathbf{0} \tag{4.15}$$

Then, for equivalent direct-form and two-multiplier lattice realizations, the residual errors are automatically equal. Moreover, with the proposed simplification on the information vector, the corresponding regression vectors also become identical, and consequently the following applies

$$E\left[e_d(n)\hat{\boldsymbol{\phi}}_d(n)\right] = \mathbf{0} \Longleftrightarrow E\left[e_{2\ell}(n)\hat{\boldsymbol{\phi}}_{2\ell}(n)\right] = \mathbf{0} \tag{4.16}$$

□

Property 4.2 above indicates the steady-state equivalence of the given lattice version with the direct-form implementation of an adaptation algorithm. The corresponding result concerning the transient portion of the adaptation process is described in the following.

Property 4.3: The standard direct-form and the above proposed lattice implementations of any given adaptive IIR filter algorithm possess similar transient processes in the sense of convergence speed and stability characteristics.

Proof: This result immediately follows from the fact that the two methods can be associated to similar ordinary difference equations as observed in [70] following the approach described in [14] and seen in Section 2.4.2.

Based on Properties 4.2 and 4.3, one can associate in a straightforward manner the well-known steady-state and transient characteristics of the direct structure implementation of any adaptation algorithm to the proposed lattice version of the same algorithm. This applies also, of course, to the CSE algorithm whose direct-form convergence characteristics were contemplated in Chapter 2.

Using the proposed simplification for the regressor vector such that $\hat{\boldsymbol{\phi}}_\ell(n) = \hat{\boldsymbol{\phi}}_d(n)$ and utilizing the recursive equations (4.11) and (4.12), lattice-based algorithms are efficiently implemented, requiring $O(N)$ multiplication and division operations as opposed to the $O(N^2)$ operations required by earlier lattice adaptation algorithms [27], [65]. Thus, the algorithms proposed here and in [61]-[63], along with the ones presented in [47], [67], and [70], present similar computational complexity to their equivalent direct-form counterparts with the additional property of allowing pole-monitoring to be implemented in real time.

Table 4.1 includes an easy-to-follow routine describing the implementation of the efficient two-multiplier lattice-based version of the CSE algorithm, to which the following comments apply:

- The auxiliary vectors defined in step 0 can be filled with zeros for a simple initialization procedure;
- The number of multiplication and division operations at each iteration for the CSE algorithm add to $15N$;
- The update of the filter coefficients in step 4 can also be performed by using a quasi-Newton type algorithm analogous to the one described in Chapter 1;
- The stability monitoring routine in step 5 must check if any coefficient $\hat{k}_m(n)$ of the two-multiplier lattice structure becomes greater or equal to unity in absolute value. If this happens, the coefficient must be stabilized by forcing its value to be inside the open interval $\hat{k}_m(n) \in (-1, 1)$ [23]. This can be accomplished by cancelling the most recent update of the respective coefficient or by making it equal to a given predefined stable value;
- The optional use of time-varying scalars μ and γ to achieve faster or more robust convergence can be done by following similar strategies to the ones previously mentioned in Chapter 3 for the direct-form realization.

Due to the shortage of space, this chapter focuses solely on lattice realizations of the CSE algorithm. However, the extension of the proposed methods to other adaptive IIR filter algorithms is easily performed. In fact, for any other given algorithm, one must simply obtain the corresponding error signal and regression vector based on their original direct-form definitions adequately applying the recursion-in-order equations (4.11)-(4.12), as illustrated here for the CSE method. For general collections of adaptive IIR filter algorithms the reader may refer to [34], [60], [74].

4.4.1 Normalized-Lattice Adaptive IIR Filter Algorithms

The extension of adaptive IIR filter algorithms from the two-multiplier lattice to the normalized lattice realization follows naturally from the relationships existing between the coefficients of these two structures. Indeed, the normalized lattice structure has the set of coefficients given by

$$\boldsymbol{\theta}_{4\ell} = \left[\phi_1 \dots \phi_N \; h_0^{'} \dots h_N^{'}\right]^T \tag{4.17}$$

that are related to the entries of $\boldsymbol{\theta}_{2\ell}$ by the following equations [24]

$$\sin\phi_i = k_i; \;\; i = 1, \dots, N \tag{4.18a}$$

$$h_j^{'} = \frac{h_j}{\pi_j}; \;\; j = 0, \dots, N \tag{4.18b}$$

where the parameters π_j are given by:

$$\pi_{j-1} = \pi_j \cos\phi_{j-1}; \;\; j = N-1, \dots, 1 \tag{4.19}$$

with $\pi_N = 1$. Applying these relationships to the equations given in Lemma 4.1, one obtains the following equivalent recursive-in-order equations for the normalized lattice realization.

Lemma 4.4: Consider the $A_m(z)$ and $B_m(z)$ polynomials with auxiliary order m as defined in (4.9). For the normalized lattice realization, these polynomials may be recursively computed as

$$A_m(z) = A_{m-1}(z) + \frac{\sin\phi_m}{\sin\phi_{m-1}} \left[A_{m-1}(z) - \cos^2\phi_{m-1} A_{m-2}(z)\right] z^{-1} \tag{4.20a}$$

$$B_m(z) = B_{m-1}(z)\cos\phi_m + \frac{h_m^{'}}{\sin\phi_m} \left[A_m(z) - \cos^2\phi_m A_{m-1}(z)\right]; \;\; m = 2, \dots, N \tag{4.20b}$$

with $A_0(z) = 1$, $A_1(z) = 1 + \sin\phi_1 z^{-1}$, and $B_1(z) = (h_0^{'}\cos\phi_1 + h_1^{'}\sin\phi_1) + h_1^{'} z^{-1}$.

□

These recursions for the normalized lattice form can be obtained by applying the coefficient transformations given in (4.18) to the recursion equations (4.11) and (4.12) for the two-multiplier lattice realization. As a consequence of equation (4.20a), one has that

$$A'_m(z) = A'_{m-1}(z) + \frac{\sin\phi_m}{\sin\phi_{m-1}}\left[A_{m-1}(z) - \cos^2\phi_{m-1}A_{m-2}(z)\right]z^{-1} \tag{4.21}$$

with the auxiliary polynomial $A'_m(z)$ as defined in (4.13).

Applying these recursions to any adaptive IIR filter algorithm in the same fashion as demonstrated for the CSE algorithm with the two-multiplier lattice structure, one can implement the normalized-lattice version of the respective algorithm in a very efficient and consistent way. As a reminder, the stability of the normalized lattice realization is guaranteed even for time-varying filters as long as the reflection coefficients satisfy $\mid \phi_i \mid < \pi/2$ or equivalently $\mid \cos\phi_i \mid < 1$ [23], [67].

4.5 Simulation Results

Below, we present computer simulations of the given efficient lattice-based CSE adaptation algorithm in an attempt to demonstrate its usefulness in practical applications.

Example 4.5: Consider the system identification example described in [65] where the plant is defined as

$$H(q) = \frac{0.0154 + 0.0462q^{-1} + 0.0462q^{-2} + 0.0154q^{-3}}{1 - 1.99q^{-1} + 1.572q^{-2} - 0.4583q^{-3}} \tag{4.22}$$

what yields the direct-form, two-multiplier and normalized lattice coefficient vectors being respectively given by

$$\boldsymbol{\theta}_d = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} -1.9900 \\ 1.5720 \\ -0.4583 \\ 0.0154 \\ 0.0462 \\ 0.0462 \\ 0.0154 \end{bmatrix} ;\ \boldsymbol{\theta}_{2\ell} = \begin{bmatrix} k_1 \\ k_2 \\ k_3 \\ h_0 \\ h_1 \\ h_2 \\ h_3 \end{bmatrix} = \begin{bmatrix} -0.87559 \\ 0.83546 \\ -0.45830 \\ 0.08565 \\ 0.14549 \\ 0.07685 \\ 0.01540 \end{bmatrix} ;\ \boldsymbol{\theta}_{4\ell} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ h'_0 \\ h'_1 \\ h'_2 \\ h'_3 \end{bmatrix} = \begin{bmatrix} -1.06665 \\ 0.98897 \\ -0.47608 \\ 0.36299 \\ 0.29787 \\ 0.08646 \\ 0.01540 \end{bmatrix} \tag{4.23}$$

Consider an adaptive filter with $N = 3$ corresponding to a strictly sufficient order identification case and let the input signal be a Gaussian noise with zero mean and unitary variance.

Assume also a perturbation signal statistically independent to the input signal consisting of Gaussian noise with zero mean and variance $\sigma_v^2 = 0.007$.

Figure 4.1 depicts the convergence of the adaptive filter coefficient vector using the CSE algorithm with $\gamma = 0$ realized by the two-multiplier and normalized lattice structures. It can be noted that as the perturbation signal is independent of the input signal, both versions of the CSE algorithm converged to their respective global optimal solutions given above and indicated by dotted lines in the figure. In these simulations, the values of the convergence parameter was optimized for a faster stable convergence via trial and error and it was equal $\mu_\ell = 0.01$.

For the same example, the direct-form OE algorithm and the lattice-form algorithm described in [65] were also utilized to identify the given plant. The direct-form version presented convergence problems due to instability of the adaptive filter during the adaptation process. To eliminate this problem the convergence parameter had to be significantly reduced largely increasing the number of iterations to achieve a satisfactory steady-state. The algorithm in [65] presented similar adaptation to the given lattice-based algorithms with the additional burden of requiring $O(N^2)$ multiplications per iteration.

Figure 4.2 shows the convergence trajectories followed by the adaptable coefficients of the two-multiplier and normalized versions of the lattice-based CSE algorithm with $\gamma = 1.0$, respectively. In both cases, due to the better stability characteristics inherent to the EE scheme, a higher value to the convergence parameter was utilized, $\mu_\ell = 0.06$, resulting in faster general convergence of the adaptive process. It can be observed that due to the presence of a perturbation signal, the solutions achieved by these methods were biased with respect to the optimal one represented in this figure by the dotted lines. It was noticed, however, that both final solutions were equivalent to each other and also to the solution of the direct-form EE algorithm for the same problem, verifying once again the validity of the result expressed in Property 4.2.

A similar experiment without the presence of the perturbation signal was then executed for the lattice-based CSE algorithm with $\gamma = 1.0$. As expected, in this case both lattice CSE versions converged to their respective global solutions given above, as seen in Figure 4.3.

□

As a summary of what is known, the advantages of the (normalized) lattice realization when applied for adaptive IIR filtering include:

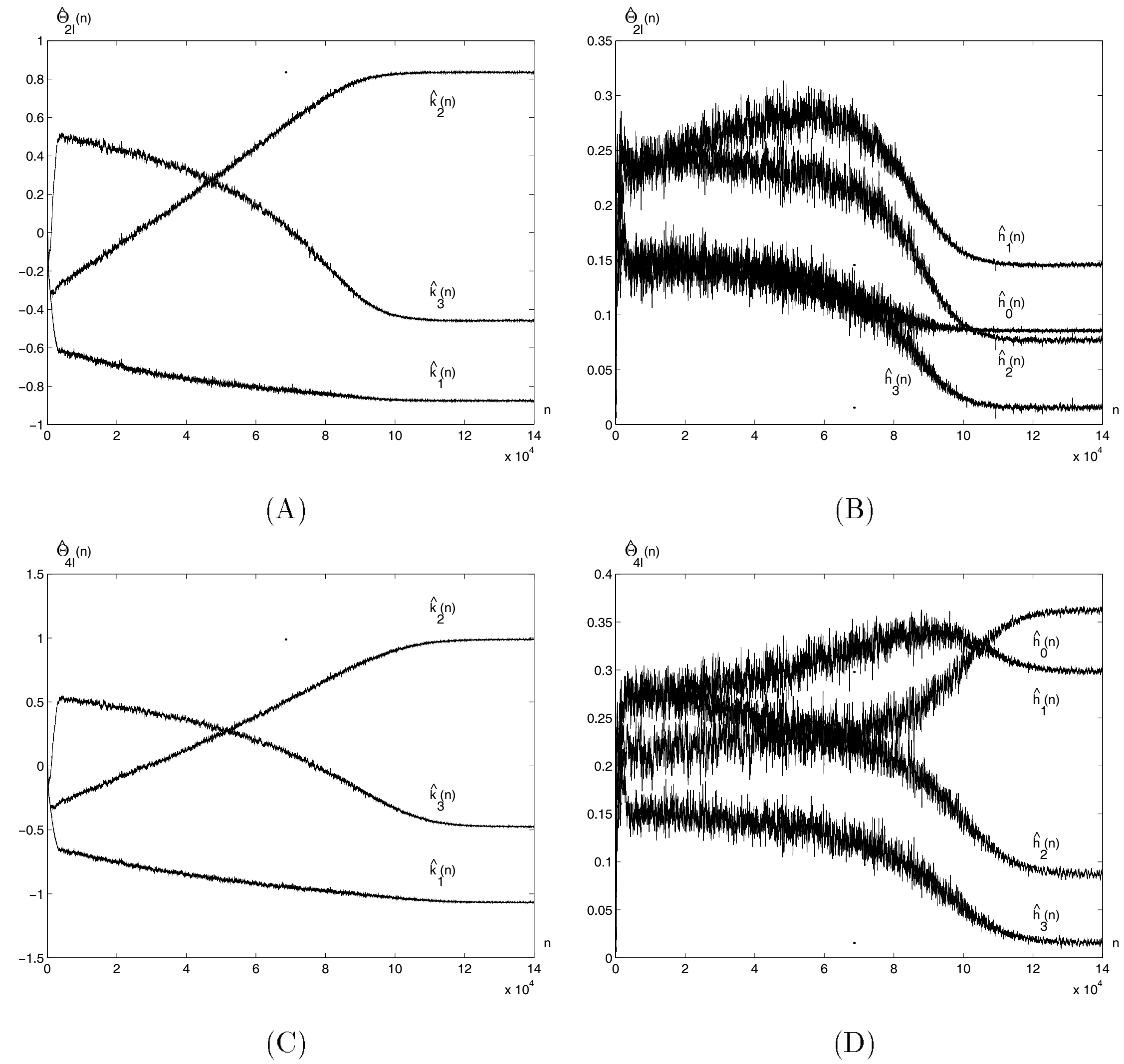


Figure 4.1: Example 4.5 - Lattice adaptive convergence of the CSE algorithm with $\gamma = 0.0$: (A) Two-multiplier denominator coefficients; (B) Two-multiplier numerator coefficients; (C) Normalized denominator coefficients; (D) Normalized numerator coefficients.

- Simple pole monitoring during the adaptation process;
- Stability monitoring refers to one coefficient at a time what can eliminate update stalemate in cases of convergence close to the stability boundaries;
- The coefficient mapping from the direct form to a lattice realization is a one-to-one transformation. As a result, no additional stationary points are created when implementing an adaptation algorithm with the lattice realization [54]. Also, no reduced-order manifolds [54] are introduced as opposed to what occurs when the

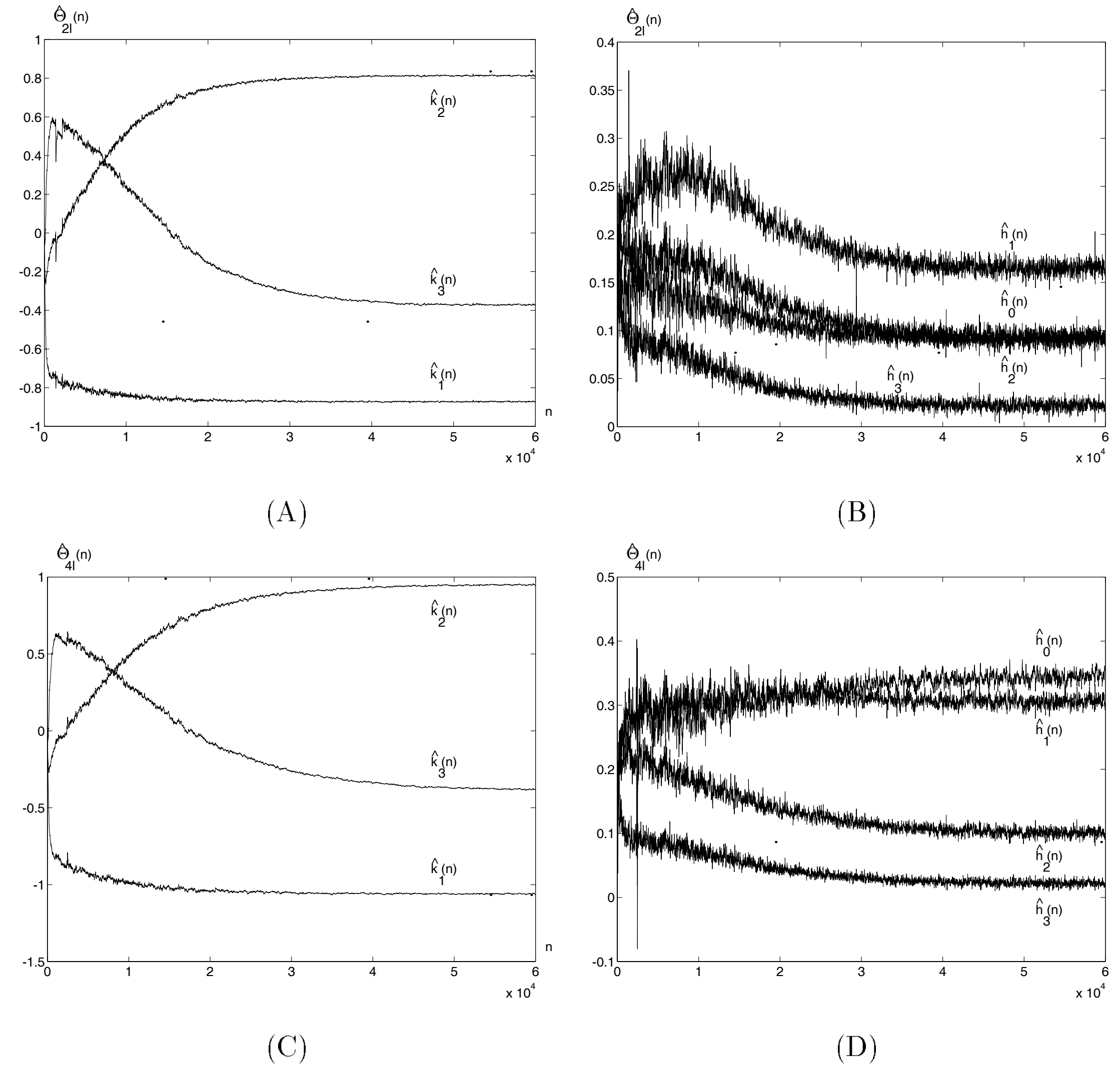
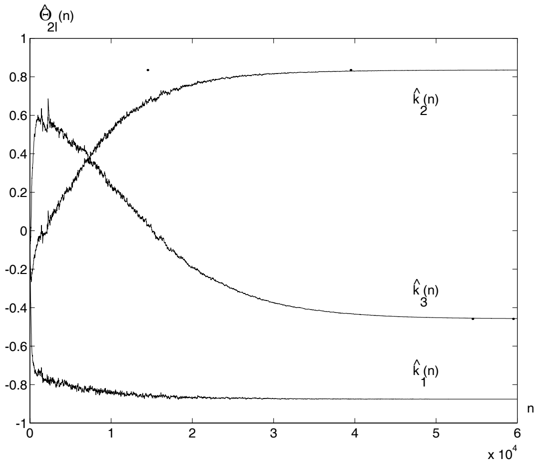


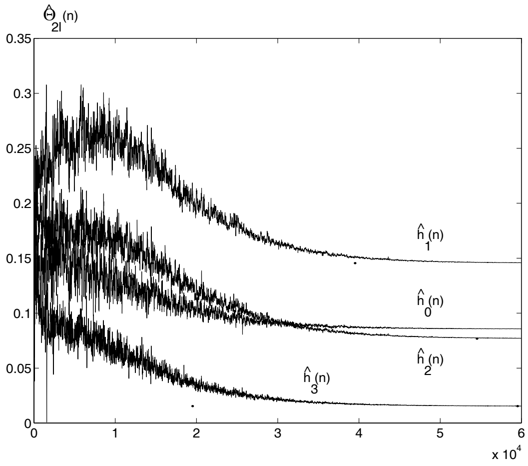
Figure 4.2: Example 4.5 - Lattice adaptive convergence of the CSE algorithm with $\gamma = 1.0$ - With perturbation noise: (A) Two-multiplier denominator coefficients; (B) Two-multiplier numerator coefficients; (C) Normalized denominator coefficients; (D) Normalized numerator coefficients.

structures of parallel or cascade of blocks are used [51]. Thus lattice-based adaptive IIR filter algorithms tend to present convergence speed similar to their respective direct-form counterparts.

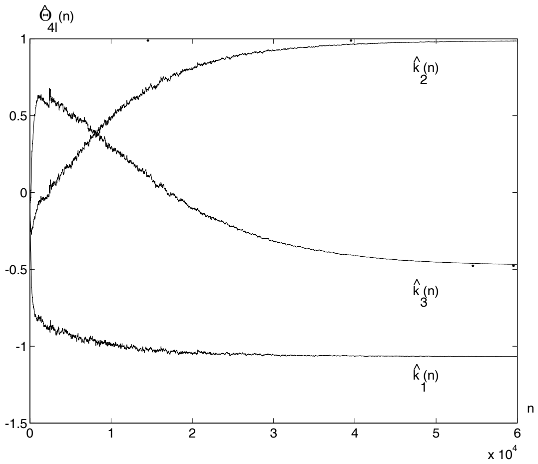
- The normalized lattice structure has proved BIBO stable properties even in cases of time-varying coefficients [24], [67]. This fact eliminates the slow adaptation requirement to guarantee BIBO stability of the adaptive filter during the convergence process;



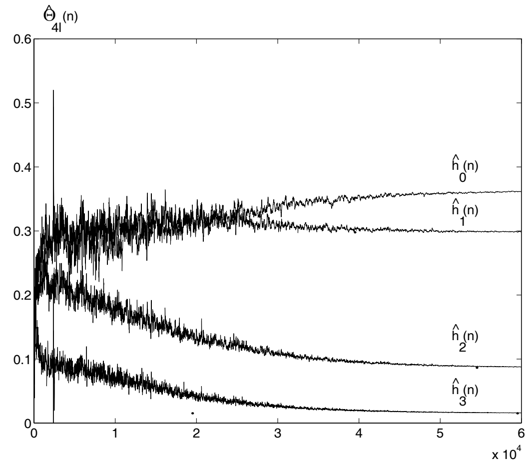
(A)



(B)



(C)



(D)

Figure 4.3: Example 4.5 - Lattice adaptive convergence of the CSE algorithm with $\gamma = 1.0$ - Without perturbation noise: (A) Two-multiplier denominator coefficients; (B) Two-multiplier numerator coefficients; (C) Normalized denominator coefficients; (D) Normalized numerator coefficients.

- The normalized lattice realization possesses extremely good sensitivity properties [24] with respect to variations of the coefficient values. This results into greater independence of convergence process with regard to the coefficient vector initial value and also into smaller misadjustment of the lattice coefficients after a steady-state is reached;

- A common problem associated to the two-multiplier lattice realization is the dynamic range of the signals in the internal nodes of the filter. This may require some sort of a signal normalization during the updating routine and also may cause implementation problems using fixed-point arithmetic. The energy of the signals at the internal nodes of the normalized lattice realization is unity [24] as the name of the structure indicates. Consequently, for this realization no normalization routine is required and additional computational effort is saved in the updating procedure of the filter coefficients;

- The major disadvantage associated to the normalized lattice structure is the utilization of the sine and cosine functions. The number of these operations, however, can be dramatically reduced at each iteration by saving their results in auxiliary variables during the adaptive processing. In real-time systems, those trigonometric functions can be implemented via the use of tables or CORDIC algorithms [97]. CORDIC techniques can compute the sine and cosine functions with p bits of accuracy in p iterations, where each iteration requires only a small number of shifts and fixed point additions.

4.6 Conclusion

A new efficient lattice-based realization was proposed for adaptive IIR filter algorithms allowing the adaptive filter poles to be monitored in real-time. Relationships of the transfer function polynomials of the direct-form and two lattice realizations were introduced. It was shown that these equations lead to the implementation of efficient lattice-based adaptive IIR filter algorithms requiring $O(N)$ multiplications per iteration, with N being the order of the adaptive filter. Additionally, consistency was achieved in the sense that the proposed lattice-based approach led to a set of parameters that realizes identical transfer functions to the ones obtained by corresponding direct-form algorithms. Computer simulations of the simplified CSE algorithm were included to demonstrate the application of the proposed methods. The results presented here and the ones previously found in the literature indicate that lattice structures constitute an efficient tool for the realization of real-time adaptive IIR filters due to the simple stability testing and good convergence speed.

Table 4.1: Summary of the efficient two-multiplier lattice CSE adaptation algorithm

step 0 - Initialize:

The scalars n, μ, and γ

The adaptive filter coefficient vector $\hat{\boldsymbol{\theta}}_{2\ell}(n) = \left[\hat{k}_1(n)\ldots\hat{k}_N(n)\ \hat{h}_0(n)\ldots\hat{h}_N(n)\right]^T$

The auxiliary vectors:

$$\begin{bmatrix} G_0(n-1) \\ \vdots \\ G_{N-1}(n-1) \end{bmatrix}, \begin{bmatrix} \hat{A}_0(q,n)\{\hat{y}^f(n-1)\} \\ \vdots \\ \hat{A}_{N-1}(q,n)\{\hat{y}^f(n-1)\} \end{bmatrix}, \begin{bmatrix} \hat{A}_0(q,n)\{x^f(n-1)\} \\ \vdots \\ \hat{A}_{N-1}(q,n)\{x^f(n-1)\} \end{bmatrix}, \begin{bmatrix} \hat{A}_0(q,n)\{e_{OE}(n-1)\} \\ \vdots \\ \hat{A}_{N-1}(q,n)\{e_{OE}(n-1)\} \end{bmatrix}$$

step 1 - Compute:

$F_m(n) = F_{m+1}(n) - \hat{k}_{m+1}(n)G_m(n-1)$; for $m = N-1,\ldots,0$; with $F_N(n) = x(n)$

$G_m(n) = G_{m-1}(n-1) + \hat{k}_m(n)F_{m-1}(n)$; for $m = 1,\ldots,N$; with $G_0(n) = F_0(n)$

$$\hat{A}'_m(q,n)\{\hat{y}^f(n)\} = \hat{A}'_{m-1}(q,n)\{\hat{y}^f(n)\} + \frac{\hat{k}_m(n)}{\hat{k}_{m-1}(n)} \cdot \left[\hat{A}_{m-1}(q,n)\{\hat{y}^f(n-1)\} - (1-\hat{k}^2_{m-1}(n))\hat{A}_{m-2}(q,n)\{\hat{y}^f(n-1)\}\right];$$

for $m = 2,3,\ldots,N$; with $\hat{A}'_1(q,n)\{\hat{y}^f(n)\} = \hat{k}_1(n)\hat{y}^f(n-1)$

$$\hat{A}'_m(q,n)\{x^f(n)\} = \hat{A}'_{m-1}(q,n)\{x^f(n)\} + \frac{\hat{k}_m(n)}{\hat{k}_{m-1}(n)} \cdot \left[\hat{A}_{m-1}(q,n)\{x^f(n-1)\} - (1-\hat{k}^2_{m-1}(n))\hat{A}_{m-2}(q,n)\{x^f(n-1)\}\right];$$

for $m = 2,3,\ldots,N$; with $\hat{A}'_1(q,n)\{x^f(n)\} = \hat{k}_1(n)x^f(n-1)$

$$\hat{A}'_m(q,n)\{e_{OE}(n)\} = \hat{A}'_{m-1}(q,n)\{e_{OE}(n)\} + \frac{\hat{k}_m(n)}{\hat{k}_{m-1}(n)} \cdot \left[\hat{A}_{m-1}(q,n)\{e_{OE}(n-1)\} - (1-\hat{k}^2_{m-1}(n))\hat{A}_{m-2}(q,n)\{e_{OE}(n-1)\}\right];$$

for $m = 2,3,\ldots,N$; with $\hat{A}'_1(q,n)\{e_{OE}(n)\} = \hat{k}_1(n)e_{OE}(n-1)$

step 2 - Obtain: $\hat{y}(n) = \sum_{j=0}^{N} \hat{h}_j(n)G_j(n)$

$e_{OE}(n) = y(n) - \hat{y}(n)$

$e_{EE}(n) = e_{OE}(n) + \hat{A}'_N(q,n)\{e_{OE}(n)\}$

$\hat{y}^f(n) = \hat{y}(n) - \hat{A}'_N(q,n)\{\hat{y}^f(n)\}$

$x^f(n) = x(n) - \hat{A}'_N(q,n)\{x^f(n)\}$

step 3 - Form: $\hat{\boldsymbol{\phi}}_{OE_{2\ell}}(n) = \left[-\hat{y}^f(n-1)\ldots-\hat{y}^f(n-N)\ x^f(n)\ldots x^f(n-N)\right]^T$

$\hat{\boldsymbol{\phi}}_{EE_{2\ell}}(n) = \left[-y(n-1)\ldots-y(n-N)\ x(n)\ldots x(n-N)\right]^T$

step 4 - Update: $\hat{\boldsymbol{\theta}}_{2\ell}(n+1) = \hat{\boldsymbol{\theta}}_{2\ell}(n) + \mu\left[\gamma e_{EE}(n)\hat{\boldsymbol{\phi}}_{EE_{2\ell}}(n) + (1-\gamma)e_{OE}(n)\hat{\boldsymbol{\phi}}_{OE_{2\ell}}(n)\right]$

step 5 - Perform the stability monitoring routine for the adaptive filter (see main text)

step 6 - Update the scalars n, μ (optional), and γ (optional) and auxiliary vectors in **step 0**

Return to **step 1**

Chapter 5

On the DSP Implementation of Adaptive Filters

5.1 Introduction

Computer simulations are a common tool for testing general convergence properties of adaptive filters. In fact, based on simulation results, one can obtain important insights on, for instance, convergence speed, stability properties, and global or biased convergence of any given adaptive algorithm or structure. However, some other significant issues associated with practical adaptive filters cannot be analyzed based solely on computer simulations. These issues include, for example, the maximum processing speed, device cost, maximum filter order, and quantization effects. For this reason, in this chapter, we describe, perform, and analyze the implementation of adaptive filters using real-time systems.

The implementation of adaptive filters can be performed in the basic forms of high-level software programming, general-purpose hardware, and application-specific or dedicated hardware. The software form, performed in general-purpose computers, is the easiest and most convenient way, as it can be performed using all kinds and levels of programming languages. However, software-based systems are relatively slow in comparison to the other two schemes, with the result of being unable to process most types of signals in real time. Consequently, the software approach is essentially of academic importance being used mainly in off-line simulations. The use of dedicated hardware requires the design and manufacturing of an integrated circuit built to perform a specific adaptive filtering task as, for instance, described in [4]. The hardware designed for this type of implementation restricts several aspects of the adaptive system as the order and the structure of the adaptive filter, the adaptation algorithm, and so on. Such an implementation allows the adaptive system to be extremely fast, reaching processing speeds far higher that the ones achieved by the other two schemes. However, the cost involved in the stages of design and manufacturing of the dedicated hardware tends to be extremely high, making this scheme improper for many practical applications. The implementation of adaptive filters using a digital signal processor (DSP) is more flexible, as this type of hardware can be programmed to perform any form of signal processing task, and it can be seen as a good compromise of processing speed, cost, and system flexibility when compared to the two other schemes. Today, this is the choice for the implementation of several practical commercial systems utilizing adaptive filters.

In this chapter, to demonstrate the practical usage of adaptive techniques, the implementation of an adaptive noise canceller on a DSP system is considered. In the following section, the configuration of an adaptive noise canceller is presented along with several considerations associated with this type of system. Next, the hardware and software used for the DSP chip implementation presented in this chapter are described. The results of this implementation are presented, analyzed, and compared to expected results from theory and to other implementations found in the literature.

5.2 Adaptive Noise Cancellation

Figure 5.1 depicts the adaptive filter in a noise cancellation configuration. In this figure, the signal $s(n)$ is being corrupted by an additive noise $v(n)$ and a distorted but correlated version of this noise, $v'(n)$, is also available. The basic idea in this case is to generate an output $\hat{y}(n)$ that closely resembles the corrupting noise $v(n)$ so that the output error signal would approximate $s(n)$ after appropriate subtraction. In this noise cancellation application, the signal $s(n)$ is also called the primary or original signal and the source of the noise signals $v(n)$ and $v'(n)$ is also referred to as the secondary input.

Initially, one may think that cancelling the noise from the reference signal $y(n)$ via subtraction to be a dangerous procedure which, if improperly done, could result in even an amplification of the noise level. However, adaptive noise cancellation is efficiently ac-

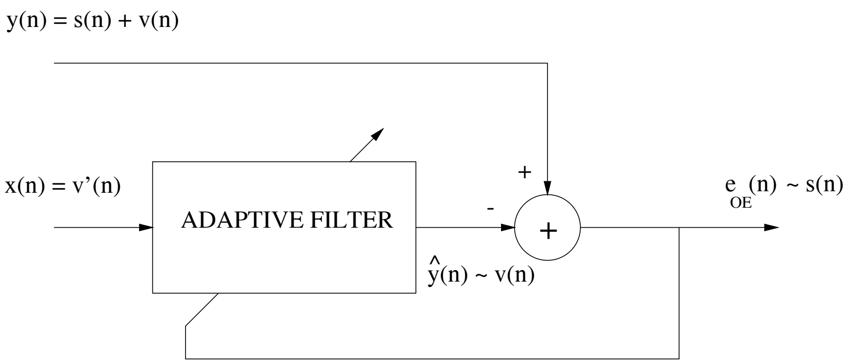


Figure 5.1: Block diagram of an adaptive noise canceller.

complished in practice with little risk to the original signal $s(n)$ by taking advantage of an estimate of $v(n)$ obtained from a position in the noise field where the original signal is negligible. In fact, in cases where adaptive techniques can be used for cancelling noise, a much better degree of noise rejection is often achieved based on adaptive filtering than using other filtering techniques. This is particularly true if the characteristics of the channel over which the noise is transmitted to the primary sensor are unknown or of a nonfixed nature.

Mathematically, obtaining the mean-squared value of the output error as defined in Figure 5.1, one gets

$$E\left[e_{OE}^2(n)\right] = E\left[s^2(n)\right] + E\left[(v(n) - \hat{y}(n))^2\right] \tag{5.1}$$

assuming that $s(n)$ is uncorrelated with both $v(n)$ and $v'(n)$, and all signals in Figure 5.1 are statistically stationary with zero means. From this equation, it is easy to verify that when the adaptive filter is adjusted to minimize $E[e_{OE}^2(n)]$, then $E[(v(n) - \hat{y}(n))^2]$ is also minimized, as the signal power $E[s^2(n)]$ is unaffected throughout the adaptation process. Consequently, by minimizing the output error power, one maximizes the output signal-to-noise ratio, as it would be expected from a noise cancellation system.

In a practical implementation of an adaptive noise canceller, some considerations must be taken into account due to nonideal situation. These considerations include, for example, effects from the presence of uncorrelated noise in either the input $x(n)$ or reference signal $y(n)$, effects from the presence of components of the original signal $s(n)$ in the input signal $x(n)$, and consequences from causality and finite-length constraints on the filter realization.

With respect to the problem of noise presence in the real environment, it can be shown that all input noise components which are correlated with reference noise components are cancelled, whereas other noncorrelated noise components are not cancelled and will be present in the error signal, negatively affecting the overall performance of the noise canceller.

The presence of the original signal $s(n)$ in the input $x(n)$ can be shown to cause the error signal of the adaptive canceller to be a distorted version of $s(n)$. Furthermore, if the input signal $x(n)$ contained signal $s(n)$ components and no noise, the signal would be perfectly cancelled and it would not be part of the error signal $e_{OE}(n)$ of the adaptive system. In a well implemented system, however, where the reference input is properly obtained, this situation is not realistic and it should not constitute a major problem. A desirable low level of signal distortion is obtained if the signal-to-noise ratio at the input $x(n)$ is low enough and the signal-to-noise ratio at the reference input $y(n)$ is reasonably high.

The analytical solution of the off-line noise cancellation problem can result, as described in [99], in an ideal optimal configuration that is both noncausal and of infinite order. Both these aspects, however, can not be complied in an on-line practical adaptive implementation. Nevertheless, the consequences of a causal and finite-length implementation of an adaptive noise canceller can be respectively counterbalanced by the inclusion of an appropriate delay in the primary input path and by the use of an adaptive filter with enough number of parameters, thus ensuring an appropriate level of performance for the overall system [99], as it will be seen in our experiments.

Although most of this dissertation is focused on adaptive IIR filtering, a quick note on the description of the adaptive FIR noise canceller is now given, as this configuration will serve as the basis to which its IIR counterpart is going to be compared to.

5.2.1 The FIR Adaptive Noise Canceller

Amongst the several forms of adaptive FIR filters known in the literature, the most common configuration is based on the tapped-delay-line (TDL) structure the output signal $\hat{y}(n)$ of which is obtained as

$$\hat{y}(n) = \mathbf{w}^T(n)\mathbf{x}(n) \tag{5.2}$$

where $\mathbf{w}(n) = [w_0(n)\ w_1(n) \ldots\ w_N(n)]^T$ is the coefficient vector and $\mathbf{x}(n) = [x(n)\ x(n-1) \ldots\ x(n-N)]^T$ is the so-called input signal vector, with N representing the filter order.

A popular way to adapt the TDL's coefficients makes use of the least mean squares (LMS) algorithm described by [99]

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e_{OE}(n)\mathbf{x}(n) \tag{5.3}$$

with $e_{OE}(n) = \hat{y}(n) - y(n)$ as before.

It has been shown that the LMS algorithm can be associated to a quadratic objective function, due to the linear relationship seen in (5.2), the minimization of which is performed using the steepest descent method [99]. The existence of a unique solution, due to the quadratic performance surface, and its extremely simple implementation justify the characterization of the LMS algorithm as the standard method to which all others adaptive techniques presented in the literature are compared. A deep study on the properties of the LMS convergence process are naturally beyond the scope of this dissertation and, for this, the interested reader is referred to [99].

5.3 The DSP-Based Adaptive Noise Canceller

In this section, a complete description of the hardware and the software used in the presented implementation is given in order to understand the characteristics of the resulting system.

5.3.1 Hardware Description

The adaptive noise canceller implemented for this work was based on the TMS320C30 digital signal processor by Texas Instruments Inc. (TI) on the SDSP/C30D board developed by Loughborough Sound Images Inc. (LSI) with added analog-to-digital (A/D) and digital-to-analog (D/A) capabilities.

The TMS320C30 [93] is a 32-bit device with an architecture optimized for computationally intense signal processing and mathematical operations using floating-point arithmetic. The 33.3MHz version used in this work has a 60ns cycle time and operates at a maximum rate of 16.7MIPS (millions of instructions per second) and 33.3MFLOPS (millions of floating point operations per second) due to possible performance of parallel multiplication and addition operations in a single cycle. Along with its optimized integer and floating-point arithmetic logic units, the DSP chip integrates 4K$\times$32-bit ROM, 2K$\times$32-bit on-chip RAM, two timers, parallel and serial interfaces, and a control unit. Moreover, the TMS320C30 uses a 24-bit addressing system capable of addressing up to 16.7 millions of 32-bit words. As it will be later verified from our experiments, the 24-bit mantissa and 8-bit exponent of the TMS320C30's floating-point format were good enough to keep any finite-wordlength effect on the overall performance of the adaptive system to a quite imperceptible level.

The SDSP/C30D SBus board [79], [81] (also referred to as the DSP motherboard, as opposed to the daughter module which includes the A/D and D/A converters) is a single-width SBus board, 83.8mm×146.7mm, connected to an SBus slot, and designed for use with SPARC/SunOS 4.1.1 (or greater) compliant systems. The DSP motherboard has two banks of zero wait-state 64K×32-bit (expandable to 2×256K×32-bit) of static RAM (SRAM) memory. Bank 0 is used to set up the reset and interrupt routines plus the code for the LSI's auxiliary library, the debugger, and the TMS320C30 executable program, while Bank 1 is generally used for data storage. In addition, there is a block of 2K×32-bit of dual-port RAM (DPRAM) to which the host computer has also full access via the SBus interface. If the host access the SRAM, the DSP chip must be held off the whole of its external SRAM until the host access is complete. In contrast, both DSP chip and host can access the DPRAM simultaneously. If access to the same location is requested, arbitration occurs, but otherwise DPRAM accesses cause no processing disruption to the host or the DSP chip, what is of particular importance when the DSP chip is performing a real-time task such as data acquisition. Last, for greater modularity, the DSP motherboard also allows communication to other boards or devices through very efficient serial and parallel interfaces.

The daughter module (DM) [80] presents two 16-bit dual input/output (I/O) channels with a maximum input sampling rate of 200KHz and maximum output reconstruction rate of 500KHz. Both I/O channels are provided with low-pass filters consisting of two cascaded, second-order, conventional Sallen-Key configurations with unity *dc* gain. Identical resistor values give a maximally-flat Butterworth response with −24dB/octave roll-off in the stopband. The cut-off frequency F_c can be adjusted by interchangeable resistor packs following the relationship $F_c = 60.5/R$, where R is the chosen resistor value in KΩ and the resultant F_c is given in KHz. The use of the A/D and D/A capabilities of the DM is simply a matter of setting up a number of registers, defining an interrupt routine, enabling DSP interrupts, and reading or writing the signal samples. Other characteristics of the DM include interchannel isolation of 85dB, input impedance of 20KΩ, and output impedance of 0.5Ω, being it capable of driving a load of down to 600Ω to a voltage range of $\mp$3V.

5.3.2 Software Description

An important conclusion drawn during early testing stage was the necessity to disregard the use of the C language as basic approach for programming the TMS320C30. This resulted from prohibitively cumbersome executable code being generated by the C compiler. In fact, for the sake of comparison, consider the off-line version of the LMS algorithm based on the TI's C and assembly languages for the respective DSP chip. Table 5.1 shows the number of cycles required by each program to implement a single iteration of that particular adaptation algorithm as a function of the adaptive filter order and the same results are graphically represented in Figure 5.2. Based on these results, the assembly language was considered the most efficient form and the only practical alternative to implement real-time adaptation algorithms for the TMS320C30.

Table 5.1: Number of DSP cycles per iteration for the off-line LMS algorithm.

Program Version	# of Cycles per Iteration
Standard C	$81 + 33N$
Optimized C	$84 + 7N$
Assembly	$28 + 3N$

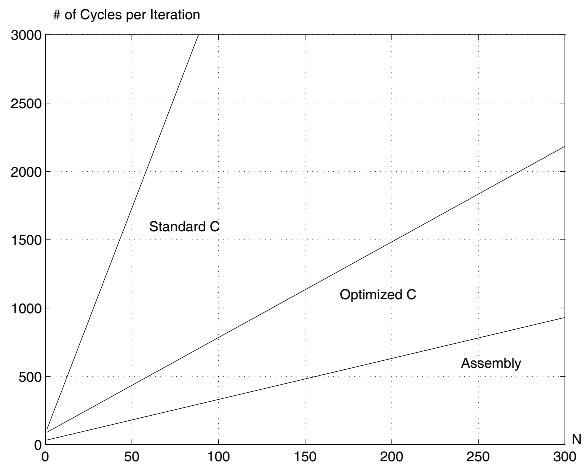


Figure 5.2: Number of cycles per iteration $\times$ filter order for the off-line LMS algorithm using standard C, optimized C, and assembly languages.

Having opted for the assembly language to program the TMS320C30, two versions of each of the LMS and CSE algorithms were developed. The first version consisted of implementing the adaptation algorithms with the TMS320C30 receiving real prerecorded data from the Sun station instead of directly from the DM. This allowed quantitative analysis of the convergence performance of each adaptation algorithm before its real-time implementation. Further, a rough estimate of the algorithm's computational complexity was also obtained since the main code in this case was essentially the same as the one of the real-time implementation. Testing the off-line version was also useful for a tune-up of the adaptation algorithms with respect to filter order, convergence factor, combination parameter, etc. A second on-line version of each algorithm was implemented on the TMS320C30 using the DM for input and output. These algorithms differed from the off-line versions by the addition of a set of instructions necessary to achieve data-format compatibility between the DM module and the TMS320C30 chip. These additional operations consisted of a proper signal scaling procedure followed by *dc* removal and subsequent integer to floating-point transformation of the input signals, while for the output signals, the appropriate inverse operations were included in the reverse order. With these on-line versions, the maximum filter orders that could be implemented using the algorithms in real time were determined.

5.4 Practical Experiments

5.4.1 Description of Experiments

The configuration depicted in Figure 5.1 was used to compose the input and reference signals for the noise cancellation experiments performed in this section. Based on that scheme, three different experiments were devised. The results of these experiments, using both adaptive FIR and IIR filters, are presented and discussed in the following sections.

Experiment 5.1: A sample of a speech signal containing the message "My name is Sergio" (with a touch of Brazilian accent) was generated as original signal $s(n)$ and additional noise, created by a second person shouting in the background, was sensed by two different sources as $v(n)$ and $v'(n)$. The obtained signals $y(n) = (s(n) + v(n))$ and $x(n) = v'(n)$, sampled at a 16KHz rate for 2 seconds (with a total of 32000 samples), were then saved to the respective files corresponding to the input and reference signals in order to be utilized by the off-line versions of the adaptation algorithms. This experiment leads mainly to qualitative results on the overall performance of the adaptive noise canceller due to the nonstationary characteristics of the speech signal.

□

Experiment 5.2: This experiment deals with the maximum order of an adaptive filter that could be implemented with the available hardware. For each distinct FIR and IIR noise canceller implemented, the on-line versions of both LMS and CSE algorithms were used to perform real-time noise cancellation to a speech signal being captured as $s(n)$ with no apparent background noise other than that generated by the microphone system. The error signal resembling $s(n)$ could be monitored with an oscilloscope after the appropriate D/A conversion performed by the DM board. For each sampling frequency, the maximum order of each filter configuration could be determined from the signal on the oscilloscope. For filter orders above the limit no output signal would be generated by the DM.

□

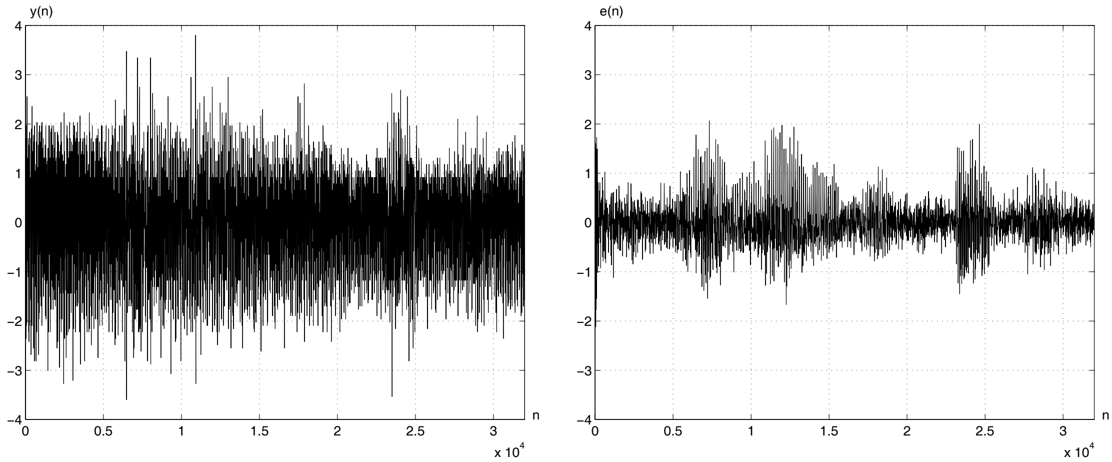
Experiment 5.3: In order to obtain quantitative results for the convergence performance of both FIR and IIR types of adaptive systems, in this experiment, some form of a humming sound with approximately constant amplitude and tone was generated as original signal $s(n)$. That signal can be approximately considered as a stationary signal. For the perturbation signal, the recording noise was used as $v(n)$ and $v'(n)$. The files corresponding to the input and reference signals were recorded using a sampling frequency of 16KHz for a total of 32000 samples to be utilized by the off-line versions of the adaptation algorithms. In order to verify the overall performance of the noise cancellation system, estimates of the final MSOE were obtained for different values of the filter order using the LMS and CSE algorithms.

□

5.4.2 Experimental Results: The FIR Case

The results of Experiment 5.1 using the off-line assembly LMS algorithm with $\mu = 0.003$ and $N = 300$ are depicted in Figure 5.3. Part (A) shows the reference signal while part (B) depicts the resulting error signal. In this case, it was verified that the mean squared value of the error signal was reduced from $\sigma_y^2 = 1.3062$ to $\sigma_e^2 = 0.1811$ during the given time interval, with the original message becoming quite audible after the adaptive noise cancelling process.

In Experiment 5.2, the maximum adaptive filter order was determined for different values of the sampling frequency using the real-time LMS algorithm. For this version, the expected



(A) (B)

Figure 5.3: Noise cancellation example using the LMS algorithm with $\mu = 0.003$ and $N = 300$: (A) Original reference signal; (B) Resulting error signal.

number of cycles per iteration, after incorporating the data-conversion instructions, was equal $48 + 3N$. Compare this value to the one for the off-line algorithm shown in Table 5.1. In Figure 5.4, the maximum filter order that could be implemented at a particular sampling frequency is presented. For the sampling frequency $F_s = 16$KHz, the maximum filter order was $N = 95$, which showed to be satisfactory for the other experiments.

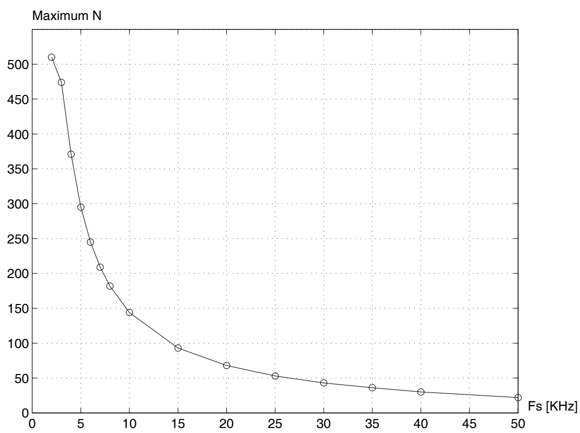


Figure 5.4: Maximum filter order × sampling frequency for the real-time implementation of the LMS algorithm using assembly language.

For Experiment 5.3, the off-line LMS algorithm was used to obtain quantitative measurements of the overall system performance, as the on-line version does not support I/O data recording. Two different estimates of the final MSOE were used in this case. A first measurement consisted of the final value of the MSOE when the adaptation algorithm was forced to converge as fast as possible without instability. That value was determined via trial-and-error experiments obtained by increasing the value of the convergence factor μ up to the point where the adaptive process presented an unstable behavior. A second measurement of MSOE was obtained when the adaptive convergence process was forced to reach a steady state after a fixed number of iterations, in this case equal 20000, and the next 12000 samples were used to estimate the MSOE. The obtained MSOE estimates for both cases are plotted in Figure 5.5. Notice the improvement of about $10dB$ achieved by increasing the order of the adaptive canceller up to $N = 128$. For higher orders, the rate of improvement becomes very small.

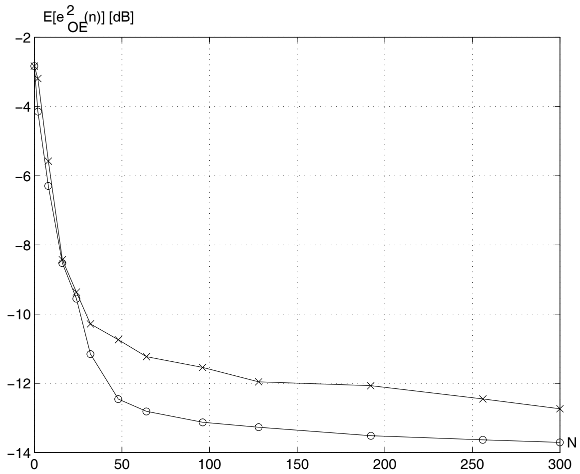


Figure 5.5: Steady-state mean squared errors of the LMS algorithm for maximum convergence speed ('o') and for convergence after 20000 iterations ('x') as functions of the adaptive filter order N.

In practice, based on these results, a good design would make use of a reasonable high-order filter, to achieve the best possible performance level at the particular sampling frequency, and a sufficiently high adaptation parameter, in order to obtain the fastest possible convergence, without being unstable.

5.4.3 Experimental Results: The IIR Case

The direct-form CSE adaptive IIR filter algorithm introduced in this dissertation was also used to implement the adaptive noise canceller.

For the real-time version of the CSE algorithm, the number of cycles necessary for each iteration was $131 + 7n_{\hat{a}} + 4n_{\hat{b}}$, where the off-line assembly version required $94 + 7n_{\hat{a}} + 4n_{\hat{b}}$ DSP cycles at each iteration. These relationships present the first difficulty when comparing the results obtained with adaptive IIR techniques with the results achieved by adaptive FIR filters. As IIR filters present numerator and denominator polynomials, their inherent implementation complexity is a function of two variables, namely $n_{\hat{a}}$ and $n_{\hat{b}}$, as opposed to the single order variable of an FIR filter. To simplify our analyses, consider $n_{\hat{a}} = n_{\hat{b}} = N$, where N is then the given order of the adaptive IIR filter.

For Experiment 5.1, similar qualitative results to the ones obtained for the LMS algorithm were obtained using the CSE algorithm. As an example, when the adaptive filter order was set to $N = 40$ the resulting MSOE was reduced from $\sigma_y^2 = 1.3062$ to $\sigma_e^2 = 0.2014$ during the given time interval, when $\mu = 0.0004$ and $\gamma = 1.0$.

The maximum value of N is depicted in Figure 5.6 for each sampling frequency for the real-time adaptive noise canceller implemented with the CSE algorithm using the TMS320C30 assembly language following the scheme described as Experiment 5.2.

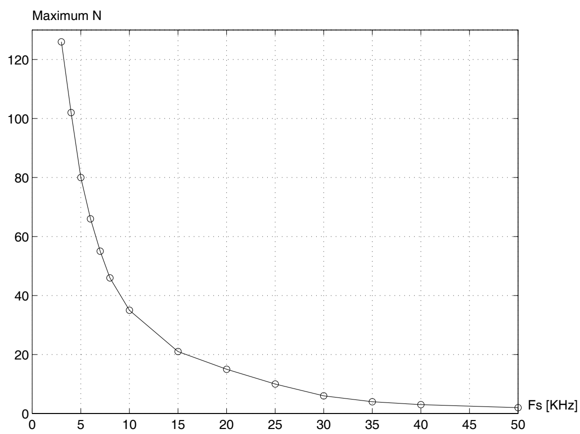


Figure 5.6: Maximum filter order × sampling frequency for the real-time implementation of the CSE algorithm using assembly language.

For Experiment 5.3, estimates for the MSOE were obtained when convergence was forced at approximately 20000 iterations and they are depicted in Figure 5.7 as a function of N for the cases when $\gamma = 1.0$ (EE algorithm) and $\gamma = 0.0$ (OE algorithm). From this figure, it can be easily concluded that in both cases the behavior of the CSE algorithm was essentially the same with a small edge in performance for the OE algorithm.

To better understand the influence of the composite parameter, the MSOE was measured for different values of γ when the filter order was fixed to $N = 15$. The results are

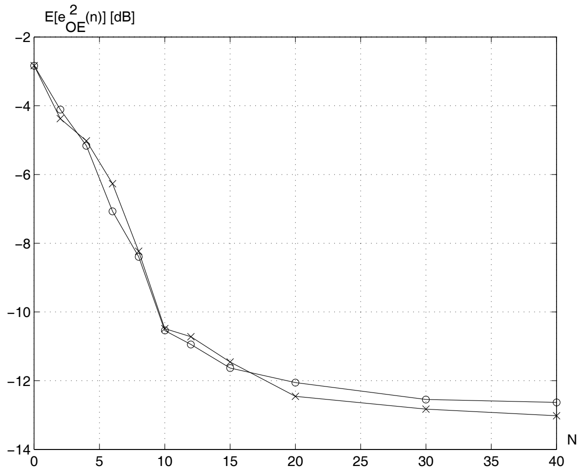


Figure 5.7: Steady-state mean squared error of the CSE algorithm for convergence after 20000 iterations with $\gamma = 1.0$ ('o') and $\gamma = 0.0$ ('x') as function of the filter order N.

shown in Figure 5.8 from which it can be seen that a proper choice of the composite factor can improve the overall performance of the algorithm. In this case, however, due to the low selectivity of the poles associated to the noise cancellation process, both EE and OE algorithms presented similar overall efficiency. Therefore, variations of γ had only limited influence on the convergence performance of the CSE algorithm. Indeed, as previously mentioned, adaptive IIR filters and also the CSE algorithm become highly effective in cases when the adaptive problem is associated to high selectivity poles. In these cases, the order of the adaptive FIR filter becomes extremely large and the EE and OE algorithms for adaptive IIR filters tend to present faulty behaviors, either converging to a suboptimal solution or presenting serious stability problems. Furthermore, the experiments included here for the CSE algorithm assume that $n_{\hat{a}} = n_{\hat{b}}$, what is not necessary in general. In fact, for most practical cases, the number of denominator coefficients can be greatly reduced in comparison to the number of numerator coefficients, thus considerably decreasing the amount of computation required by the CSE algorithm.

5.5 Comments and Discussions

The results presented in this section indicate that both FIR and IIR types of the DSP-based adaptive noise cancellers presented quite satisfactory performances. The fact that adaptive FIR systems were able to efficiently cancel the disturbance signal is an indication that the channel through which $v'(n)$ is being generated from $v(n)$ is essentially a channel

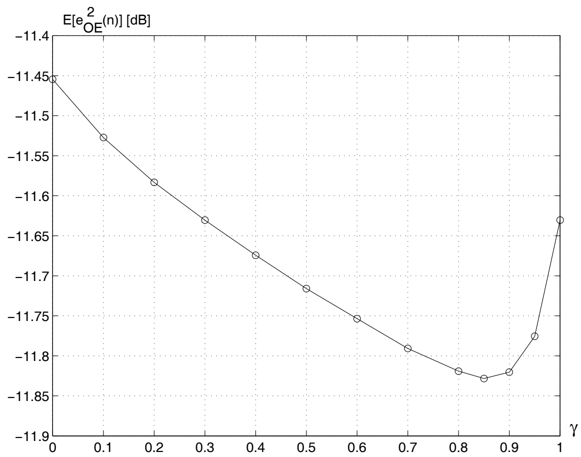


Figure 5.8: Steady-state mean squared error of the CSE algorithm for convergence after 20000 iterations with $N = 15$ as function of the composite factor value γ.

with finite-duration impulse response. This should also explain why the EE and the OE algorithms presented similar performances, as indicated in Figure 5.7.

The good behavior of both EE and OE algorithms is a positive aspect for the adaptive IIR filtering option, even for this case of a system of low-selectivity poles. It should be kept in mind that in cases where adaptive IIR filters are bound to outperform their FIR counterparts, the adaptive filter is expected to present long-duration impulse responses indicating the presence of high-selectivity poles. Such problems include, for instance, cancellation of acoustic interferences generated by multipath propagation in closed environments [85] or by propagation through channels characterized by long-duration impulse responses [99] . For these cases, it is clear that adaptation algorithms and structures others than the OE and EE algorithms and the direct-form realization are a must to achieve desirable performance levels. In such cases, the CSE algorithm and the lattice realization are of particular importance, as indicated in the previous chapters of this dissertation.

In [37] and [92], other successful applications of adaptive IIR filters are presented. In fact, in [37] the authors compare the performance of adaptive IIR and FIR techniques in an active noise control system. Due to the peculiar recursive nature of the problem, it was concluded that in that case, adaptive IIR filters actually outperformed adaptive FIR filters by achieving better noise control levels. In [92], adaptive IIR filter algorithms are implemented in an acoustic echo canceller configuration using the ADSP21020 chip.

The comparison of computational complexity of different adaptation algorithms depends on the implementation itself, as well as many other factors. These factors include, for instance, the type of DSP chip being used and the programmer's ability to write the best possible code for the given algorithm. For example, in software implementations, a division operation usually takes as much time to be performed as a multiplication operation. This is not generally true for DSP-based implementations which perform multiplications as fast as additions or subtractions and many times faster than division operations, that usually require several CPU cycles to be performed. In addition, when using DSP chips, one must take into account the time spent to perform tasks like loading data to registers, storing data in memory, writing and reading data to and from stack, and so on. Due to these issues, the analyses included in this chapter are not intended to be absolute in their content, but to represent basic evidence for the sake of a better comparison of essentially distinct approaches. In practice, one must know the basic properties associated to each and every form of implementation to better be able to choose between the several practical alternatives.

Commercial DSP chips suited for adaptive filtering include the Motorola DSP56200 chip [4] that can implement the LMS algorithm with 256 taps on a sampling rate up to 19KHz. As this is already an old design introduced in 1988, one can expect more recent devices to achieve much higher performance levels.

In [66], a dedicated architecture implementing the recursive least squares (RLS) [25] algorithm for adaptive FIR filters is performed using pipelining techniques achieving a processing speed of up to 100MHz. The design, however, is specific to a third-order filter using a mix of 8-bit and 12-bit fixed-point representations and it does not represent a commercial product.

5.6 Conclusion

The usage of adaptive techniques to process real-time practical signals was considered. The implementation of an adaptive noise canceller based on the TMS320C30 digital signal processor of Texas Instruments Inc. was carried out. Both C and assembly programming languages were considered to encode the adaptation algorithms, but final results largely favored the assembly language as much faster executable code was generated by this approach. That ultimately resulted in a larger filter order being possible for the implementation of the real-time adaptation algorithm. Also, both adaptive FIR and IIR techniques were considered and emphasis was given to the comparison of their efficiency with respect to computational complexity and overall system performance.

In the particular experiments included here, it was verified that adaptive IIR filters presented similar performance to their FIR counterparts, even in cases of short-duration impulse responses. The practical importance of adaptive IIR filters, however, is mostly associated with cases when the adaptive filter must present long-duration impulse responses, as for such cases FIR realizations tend to become excessively cumbersome. It must be made clear, however, that adaptive IIR filters are not expected to completely substitute adaptive FIR systems, since in many applications that are not associated with long impulse responses, adaptive FIR filters are expected to present satisfactory performances. The bottom line then is that, due to recent advancements in the topic of adaptive IIR filtering, as some of the ones included in this dissertation, an adaptive system designer should be familiar with both FIR and IIR forms of adaptive technology to judiciously decide on which one to use, based on the characteristics of the practical application of interest.

Chapter 6

Final Comments and Further Considerations

6.1 Final Comments

The object of this dissertation has been to present new robust and efficient techniques applicable to adaptive IIR filtering.

In Chapter 1 some issues associated to the design of adaptive IIR filters were outlined by presenting a structured introduction to the area of adaptive signal processing. It was verified that a basic adaptive system is generally characterized by the application for which it has been designed, the adaptive filter structure and the adaptation algorithm. Examples were given of adaptive filter applications and structures. In addition, the so-called equation error (EE) and output error (OE) adaptive IIR filter algorithms were introduced following a predefined unifying framework.

In Chapter 2, one of the main contributions of this dissertation, the composite squared error (CSE) adaptive IIR filter algorithm, was presented. Development of the CSE algorithm is based on the composition of the EE and OE schemes previously introduced and it makes use of an explicit composition factor that allows better control of the CSE convergence characteristics. Steady-state analyses for the CSE algorithm were included relating its stationary points to the ones of the EE and OE algorithms. Two methods were discussed for the transient analysis of the CSE algorithm. The first method consisted of a local linearization of the algorithm near a given stationary point. The second approach, based on the association of an ordinary difference equation to the CSE algorithm, allowed a more general analysis than the linearization method covering the complete adaptive parameter space. Based on these techniques it was verified that the CSE algorithm is an interesting tool that is able to take advantage of the positive qualities of both the EE and OE schemes.

In Chapter 3, the use of variable convergence factor and composite parameter for the CSE algorithm was investigated. It was demonstrated that time-varying convergence factors for both steepest-descent and quasi-Newton versions of the CSE algorithm can be very useful in simplifying the convergence parameter choice with a minimal amount of additional computational burden. Additionally, the use of a time-varying composite factor for the CSE algorithm was considered in order to obtain global convergence despite existence of a biased EE solution or of local minimum solutions for the OE algorithm. It was then verified that these techniques can in fact constitute an efficient form to obtain faster and global convergence with the CSE algorithm.

In Chapter 4, a new efficient lattice-based realization was proposed for any adaptive IIR filter algorithm allowing the adaptive filter poles to be monitored in real time. Relationships of the transfer function polynomials of the direct-form and two lattice realizations were introduced. It was shown that these equations lead to the implementation of efficient lattice-based adaptive IIR filter algorithms requiring $O(N)$ multiplications/divisions per iteration, with N being the order of the adaptive filter. Additionally, consistency was achieved in the sense that the proposed lattice-based approach led to a set of parameters realizing identical transfer functions to the ones obtained by corresponding direct-form algorithms. The implementation of the CSE algorithm using the proposed realization was performed and examples were given demonstrating its usefulness. The results presented here and the ones previously found in the literature indicate that lattice structures constitute an efficient tool for the realization of real-time adaptive IIR filters due to the simple stability testing, efficient implementation, and good convergence speed.

In Chapter 5, the application of adaptive techniques to process real-time practical signals was investigated. The implementation of an adaptive noise canceller on the TMS320C30 digital signal processor of Texas Instruments Inc. was carried out. Both C and assembly programming languages were considered to encode the adaptation algorithms, with final results largely favoring the assembly language as much faster executable code was generated with this approach. That ultimately resulted in a larger filter order being used when the implementation of the real-time adaptation algorithm. Also, both adaptive FIR and IIR techniques were considered and emphasis was given to the comparison of their efficiency with respect to computational complexity and overall system performance. It was demonstrated that adaptive IIR filters are also associated to positive qualitative and quantitative performances when a proper implementation is considered.

6.2 Further Considerations

The field of adaptive IIR filtering has become a very active research area in the past recent years. Development of better techniques, as the ones proposed in this dissertation, can only motivate even more research to be performed in this area. Also, improvements on implementation devices allow the application of adaptive IIR filters in a broader range of practical situations, resulting into even more importance being associated to this topic. In this section, some possible ways to naturally extend the work presented in this dissertation are given in an attempt to motivate the interested reader to continue and further advance the respective subject.

A very important issue related to adaptive filters is the analysis of the implementation of the adaptation algorithm based on finite-precision fixed-point arithmetic. This kind of numeric representation yields faster implementation of the adaptive processing techniques than the one achieved with floating-point representation due to its intrinsic simplified format. For adaptive FIR filters, it has been verified that the finite-precision effects include degeneration of the overall performance of the adaptation algorithm, causing an excessive squared-error misadjustment on the steady-state performance, and possible instability. The study of the effects resulting from the implementation of adaptive IIR filters in a fixed-point format becomes an even more important issue due to the recursive nature of these devices that make them more susceptible to serious stability problems.

Other very interesting aspect is the study of time-varying convergence parameters for the CSE algorithm extending the work seen here in Chapter 3. One possible approach could be the use of individual parameters for each adaptable coefficient following the method seen in [48]. Also, other methods to update the composite parameter of the CSE algorithm may also be attempted in order to obtain global convergence with this algorithm.

For some applications such as synthetic aperture [96], a common procedure to simplify the design of the overall adaptive system is to use partially adaptive signal processing techniques. These methods are based on the use of a time-invariant system in combination with a standard adaptive filter to process the incoming signals. It has been verified in the corresponding literature [96] that this methodology lead to simpler reduced-order sub-optimal adaptive filters that are easier to be implemented in practice. A good research line in this area, following the material included in this dissertation, would be to consider the use of adaptive IIR techniques for partially adaptive processing with the intent of reducing even further the number of adaptable coefficients.

As mentioned before, improvements on the speed and cost of the hardware devices suitable to implement adaptive techniques result in an even greater variety of applications to which adaptive IIR filters can be used in. Naturally, this class of systems must present sufficiently robust and reliable convergence properties, as obtained with the proposed techniques given in this dissertation. This larger number of practical situations opens a huge opportunity to experiment with adaptive IIR filters extending their usage to applications other than the ones seen here and known so far. This represents an endless research topic that will tend to become more and more important the more efficient and the more reliable adaptive IIR filters become.

In Chapter 5, the implementation of DSP-based adaptive filters was considered to perform real-time noise cancellation of some acoustic interference. A very important research topic in that matter would be to analyze and develop specific architectures suitable to the implementation of adaptive IIR filters. These architectures should take advantage of benefits like parallel and pipeline processing making them more efficient to deal with real-time applications.

References

[1] K. J. Åström and P. Eykhoff, "System identification — a survey," *Automatica*, vol. 7, no. 2, pp. 123–162, Mar. 1971.

[2] K. J. Åström and T. Söderström, "Uniqueness of the maximum likelihood estimates of the parameters of an ARMA model," *IEEE Transactions on Automatic Control*, vol. AC-19, no. 6, pp. 769–773, Dec. 1974.

[3] I. L. Ayala, "On a new adaptive lattice algorithm for recursive filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-30, no. 2, pp. 316–319, Apr. 1982.

[4] D. E. Borth, I. A. Gerson, J. R. Haug, and C. D. Thompson, "A flexible adaptive FIR filter VLSI IC," *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 3, pp. 494–502, Apr. 1988.

[5] J. A. Cadzow, "Recursive digital filter synthesis via gradient based algorithms," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-24, no. 5, pp. 349–355, Oct. 1976.

[6] J. E. Cousseau and P. S. R. Diniz, "On optimal convergence factor for IIR adaptive filters," *Proceedings of IEEE International Symposium on Circuits and Systems*, London, England, pp. 2.137–2.140, May/June 1994.

[7] J. E. Cousseau and P. S. R. Diniz, "On the behavior of the Steiglitz-McBride method for insufficient-order identification," *Proceedings of 37th Midwest Symposium on Circuits and Systems*, Rio de Janeiro, Brazil, Aug. 1995.

[8] R. A. David, "A modified cascade structure for IIR adaptive algorithms," *Proceedings of 15th Asilomar Conference on Circuits, Systems & Computers*, Pacific Grove, USA, pp. 175–179, May 1981.

[9] M. Dentino, J. M. McCool, and B. Widrow, "Adaptive filtering in the frequency domain," *Proceedings IEEE*, vol. 66, no. 12, pp. 1658–1659, Dec. 1978.

[10] P. S. R. Diniz and L. W. P. Biscainho, "Optimal variable step size for the LMS/Newton algorithm with application to subband adaptive filtering," *IEEE Transactions on Signal Processing*, vol. 40, no. 11, pp. 2825–2829, Nov. 1992.

[11] P. S. R. Diniz, J. E. Cousseau, and A. Antoniou, "Improved parallel realisation of IIR adaptive filters," *IEE Proceedings-G*, vol. 140, no. 5, pp. 322–328, Oct. 1993.

[12] H. Fan and W. K. Jenkins, "A new adaptive IIR filter," *IEEE Transactions on Circuits and Systems*, vol. CAS-33, no. 10, pp. 939–947, Oct. 1986.

[13] H. Fan, "A unified analysis of direct form adaptive IIR filters using an ODE method," *Proceedings of 21st Asilomar Conference on Circuits, Systems & Computers*, Pacific Grove, USA, pp. 139–143, Nov. 1987.

[14] H. Fan, "Application of Benveniste's convergence results in a study of adaptive IIR filtering algorithms," *IEEE Transactions on Information Theory*, vol. 34, no. 4, pp. 692–709, July 1988.

[15] H. Fan and M. Nayeri, "On error surfaces of sufficient order adaptive IIR filters: Proofs and counterexamples to a unimodality conjecture," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 9, pp. 1436–1442, Sept. 1989.

[16] H. Fan and M. Nayeri, "On reduced order identification; revisiting 'On some system identification techniques for adaptive filtering'," *IEEE Transactions on Circuits and Systems*, vol. 37, no. 9, pp. 1144–1151, Sept. 1990.

[17] H. Fan and M. Dorolosvački, "On 'global convergence' of Steiglitz-McBride adaptive algorithm," *IEEE Transactions on Signal Processing*, vol. 40, no. 2, pp. 73–87, Feb. 1993.

[18] P. L. Feintuch, "An adaptive recursive LMS filter," *Proceedings IEEE*, vol. 64, no. 11, pp. 1622–1624, Nov. 1976.

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

[20] J. J. Fuchs, "On the good use of the spectral radius of a matrix," *IEEE Transactions on Automatic Control*, vol. AC-27, no. 5, pp. 1134–1135, Oct. 1982.

[21] R. A. Gabel and R. A. Roberts, *Signals and Linear Systems*, John Wiley & Sons, New York, USA, 3rd Edition, 1987.

[22] G. C. Goodwin and K. S. Sin, *Adaptive Filtering Prediction and Control*, Prentice-Hall, Englewood Cliffs, USA, 1984.

[23] A. H. Gray, Jr., and J. D. Markel, "Digital lattice and ladder filter synthesis," *IEEE Transactions on Audio and Electroacoustics*, vol. AU-21, no. 6, pp. 492–500, Dec. 1973.

[24] A. H. Gray, Jr., and J. D. Markel, "A normalized digital filter structure," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-23, no. 3, pp. 268–277, June 1975.

[25] S. S. Haykin, *Adaptive Filter Theory*, Prentice-Hall, Englewood Cliffs, USA, 2nd Edition, 1991.

[26] S. Horvath, Jr., "A new adaptive recursive LMS filter," *Digital Signal Processing*, V. Cappellini and A. G. Constantinides, Academic Press, pp. 21–26, New York, USA, 1980.

[27] S. Horvath, Jr., "Lattice form adaptive recursive digital filters: Algorithms and applications," *Proceedings of IEEE International Symposium on Circuits and Systems*, Houston, USA, pp. 128–133, Apr. 1980.

[28] T. C. Hsia, "A simplified adaptive recursive filter design," *Proceedings IEEE*, vol. 69, no. 9, pp. 1153–1155, Sept. 1981.

[29] V. Hutson and J. S. Pym, *Applications of Functional Analysis and Operator Theory*, Academic Press, London, UK, 1980.

[30] F. Itakura and S. Saito, "Digital filtering techniques for speech analysis and synthesis," *Proceedings of 7th. International Congress on Acoustics*, Budapest, Hungary, pp. 261–264, 1971.

[31] D. A. Johns, W. M. Snelgrove, and A. S. Sedra, "Nonideal effects in continuous-time adaptive recursive filters," *Proceedings of 32nd. Midwest Symposium on Circuits and Systems*, Champaign, USA, pp. 594–587, Aug. 1990.

[32] C. R. Johnson, Jr., "An output error identification interpretation of model reference adaptive control," *Automatica*, vol. 16, pp. 419–421, July 1980.

[33] C. R. Johnson, Jr., "The common parameter estimation basis of adaptive filtering, identification, and control," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-30, no. 4, pp. 587–595, Aug. 1982.

[34] C. R. Johnson, Jr., "Adaptive IIR filtering: Current results and open issues," *IEEE Transactions on Information Theory*, vol. IT-30, no. 2, pp. 237–250, Mar. 1984.

[35] T. Kailath, *Linear Systems*, Prentice-Hall, Englewood Cliffs, USA, 1980.

[36] J. B. Kenney and C. E. Rohrs, "The composite regressor algorithm for IIR adaptive systems," *IEEE Transactions on Signal Processing*, vol. 41, no. 2, pp. 617–628, Feb. 1993.

[37] S. M. Kuo and J. Luan, "On-line modelling and feedback compensation for multiple-channel active noise control systems," *Applied Signal Processing*, vol. 1, no. 2, pp. 64–75, 1994.

[38] I. D. Landau, "Unbiased recursive identification using model reference adaptive techniques," *IEEE Transactions on Automatic Control*, vol. AC-21, no. 2, pp. 194–202, Apr. 1976.

[39] M. G. Larimore, J. R. Treichler, and C. R. Johnson, Jr., "SHARF: an algorithm for adapting IIR digital filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-28, no. 4, pp. 428–440, Aug. 1980.

[40] J.-N. Lin and R. Unbehauen, "Bias-remedy least mean square equation error algorithm for IIR parameter recursive estimation," *IEEE Transactions on Signal Processing*, vol. 40, no. 1, pp. 62–69, Jan. 1992.

[41] L. Ljung, "Analysis of recursive stochastic algorithms," *IEEE Transactions on Automatic Control*, vol. AC-22, no. 4, pp. 551–575, Aug. 1977.

[42] L. Ljung and Söderström, *Theory and Practice of Recursive Identification*, The MIT Press, Cambridge, USA, 1983.

[43] L. Ljung, *System Identification: Theory for the User*, Prentice-Hall, Englewood Cliffs, USA, 1987.

[44] D. G. Luenberger, *Introduction to Linear and Nonlinear Programming*, Addison-Wesley, Reading, USA, 1973.

[45] D. F. Marshall, W. K. Jenkins, and J. J. Murphy, "The use of orthogonal transforms for improving performance of adaptive filters," *IEEE Transactions on Circuits and Systems*, vol. 36, no. 4, pp. 474–484, Apr. 1989.

[46] U. Menzi and G. S. Moschytz, "Adaptive switched capacitor filters based on the LMS algorithm," *Proceedings of IEEE International Symposium on Circuits and Systems*, New Orleans, USA, pp. 2220–2223, May 1990.

[47] K. X. Miao, H. Fan, and M. Doroslovački, "Cascade lattice IIR adaptive filters," *IEEE Transactions on Signal Processing*, vol. 42, no. 4, pp. 721–742, Apr. 1994.

[48] W. B. Mikhael, F. H. Wu, L. G. Kazovsky, G. S. Kang, and L. J. Fransen, "Adaptive filters with individual adaptation of parameters," *IEEE Transactions on Circuits and Systems*, vol. CAS-33, no. 7, pp. 677–686, July 1986.

[49] S. S. Narayan and A. M. Peterson, "Frequency domain least-mean-square algorithm," *Proceedings IEEE*, vol. 69, no. 1, pp. 124–126, Jan. 1981.

[50] S. S. Narayan, A. M. Peterson, and N. J. Narasimha, "Transform domain LMS algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing*. vol. ASSP-31, no. 3, pp. 609–615, June 1983.

[51] M. Nayeri and W. K. Jenkins, "Alternate realizations to adaptive IIR filters and properties of their performance surfaces," *IEEE Transactions on Circuits and Systems*, vol. 36, no. 4, pp. 485–496, Apr. 1989.

[52] M. Nayeri, "Uniqueness of MSOE estimates in IIR adaptive filtering; a search for necessary conditions," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Glasgow, Scotland, pp. 1047–1050, May 1989.

[53] M. Nayeri, H. Fan, and W. K. Jenkins, "Some characteristics of error surfaces for insufficient order adaptive IIR filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 7, pp. 1222–1227, July 1990.

[54] M. Nayeri, "Lattice form IIR adaptive filters: A structural issue in filtering," *Proceedings of 32nd. Midwest Symposium on Circuits and Systems*, Champaign, USA, pp. 1147–1149, Aug. 1990.

[55] M. Nayeri, "A mildly weaker sufficient condition in IIR adaptive filtering," *IEEE Transactions on Signal Processing*, vol. 42, no. 1, pp. 203–206, Jan. 1994.

[56] S. L. Netto and P. S. R. Diniz, "Composite algorithms for adaptive IIR filtering," *IEE Electronics Letters*, vol. 28, no. 9, pp. 886–888, Apr. 1992.

[57] S. L. Netto and P. S. R. Diniz, "New algorithms for adaptive IIR filtering and their properties," *Proceedings of Canadian Conference on Electrical and Computer Engineering*, Toronto, Canada, pp. TM4.13.1–TM4.13.4, Sept. 1992.

[58] S. L. Netto and P. Agathoklis, "A homotopy continuation mapping for the Steiglitz-McBride IIR adaptive algorithm," *Proceedings of IEEE International Symposium on Circuits and Systems*, London, England, pp. 2.57–2.60, May/June 1994.

[59] S. L. Netto and P. Agathoklis, "A new composite adaptive IIR algorithm," *Proceedings of 28th Asilomar Conference on Signals, Systems & Computers*, Pacific Grove, USA, pp. 1506–1510, Oct./Nov. 1994.

[60] S. L. Netto, P. S. R. Diniz, and P. Agathoklis, "Adaptive IIR filtering algorithms for system identification: A general framework," *IEEE Transactions on Education*, vol. 26, no. 1, pp. 54–66, Feb. 1995.

[61] S. L. Netto and P. Agathoklis, "Composite algorithms for adaptive IIR filtering using lattice realization," *Proceedings of IEEE Pacific Rim Conference on Communications, Computers, and Signal Processing*, Victoria, Canada, pp. 189–192, May 1995.

[62] S. L. Netto and P. Agathoklis, "Extension of the equation error scheme for alternative realizations of adaptive IIR filters," *Proceedings of 37th Midwest Symposium on Circuits and Systems*, Rio de Janeiro, Brazil, Aug. 1995.

[63] S. L. Netto and P. Agathoklis, "Efficient lattice realizations of adaptive IIR algorithms," *Submitted to IEEE Transactions on Signal Processing*, July 1995.

[64] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, New York, USA, 3rd Edition, 1991.

[65] D. Parikh, N. Ahmed, and S. D. Stearns, "An adaptive lattice algorithm for recursive filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-28, no. 1, pp. 110–112, Feb. 1980.

[66] K. J. Ragunath and K. K. Parhi, "A 100 MHz pipelined RLS adaptive filter," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Detroit, USA, pp. 3187–3190, May 1995.

[67] P. A. Regalia, "Stable and efficient lattice algorithms for adaptive IIR filtering," *IEEE Transactions on Signal Processing*, vol. 40, no. 2, pp. 375–388, Feb. 1992.

[68] J. A. Rodríguez-Fonollosa and E. Masgrau, "Improved convergence of gradient algorithms for adaptive IIR filters," *Proceedings of IEEE International Symposium on Circuits and Systems*, Portland, USA, pp. 1732–1735, May 1989.

[69] J. A. Rodríguez-Fonollosa and E. Masgrau, "Spectral sensitivity and convergence rate in adaptive IIR filtering," *Proceedings of IEEE International Symposium on Circuits and Systems*, New Orleans, USA, pp. 1959–1962, May 1990.

[70] J. A. Rodríguez-Fonollosa and E. Masgrau, "Simplified gradient calculation in adaptive IIR lattice filters," *IEEE Transactions on Signal Processing*, vol. 39, no. 7, pp. 1702–1705, July 1991.

[71] J. J. Shynk, "Performance of alternative adaptive IIR filter realizations," *Proceedings of 21st Asilomar Conference on Circuits, Systems & Computers*, Pacific Grove, USA, pp. 144–150, Nov. 1987.

[72] J. J. Shynk, "On lattice-form algorithms for adaptive IIR filtering," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, New York, USA, pp. 1554-1557, Apr. 1988.

[73] J. J. Shynk, "Adaptive IIR filtering using parallel-form realizations," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 4, pp. 519–533, Apr. 1989.

[74] J. J. Shynk, "Adaptive IIR filtering," *IEEE Acoustics, Speech, and Signal Processing Magazine*, vol. 6, no. 2, pp. 4–21, Apr. 1989.

[75] T. Söderström, "On the uniqueness of maximum likelihood identification," *Automatica*, vol. 11, no. 2, pp. 193–197, Mar. 1975.

[76] T. Söderström and P. Stoica, "On the stability of dynamic models obtained by least-squares identification," *IEEE Transactions on Automatic Control*, vol. AC-26, no. 2, pp. 575–577, Apr. 1981.

[77] T. Söderström and P. Stoica, "Some properties of the output error method," *Automatica*, vol. 18, no. 1, pp. 93–99, Jan. 1982.

[78] T. Söderström and P. Stoica, "On some system identification techniques for adaptive filtering," *IEEE Transactions on Circuits and Systems*, vol. 35, no. 4, pp. 457–461, Apr. 1988.

[79] Spectrum Signal Processing Inc., *SDSP/C30D Sbus Board User's Guide*, Document no. 500-00049, Version 1.10, Mar. 1993.

[80] Spectrum Signal Processing Inc., *Burr-Brown Analog Daughter Module User's Manual*, Document no. 500-00060, Version 1.03, Nov. 1993.

[81] Spectrum Signal Processing Inc., *SDSP/C30D Sbus Board Technical Reference Manual*, Document no. 500-00081, Version 1.02, Feb. 1994.

[82] S. D. Stearns, G. R. Elliott, and N. Ahmed, "On adaptive recursive filtering," *Proceedings of 10th Asilomar Conference on Circuits, Systems & Computers*, Pacific Grove, USA, pp. 5–11, Nov. 1976.

[83] S. D. Stearns, "Error surfaces of recursive adaptive filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-29, no. 3, pp. 763–766, June 1981.

[84] K. Steiglitz and L. E. McBride, "A technique for the identification of linear systems," *IEEE Transactions on Automatic Control*, vol. AC-10, no. 4, pp. 461–464, Oct. 1965.

[85] R. W. Stewart, R. Duncan, and S. Weiss, "Multi-channel active noise cancellation using the DSP56001," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Minneapolis, USA, pp. I-185–I-188, Apr. 1993.

[86] P. Stoica and T. Söderström, "The Steiglitz-McBride identification algorithm revisited — convergence analysis and accuracy aspects," *IEEE Transactions on Automatic Control*, vol. AC-26, no. 3, pp. 712–717, June 1981.

[87] P. Stoica and T. Söderström, "On the convergence properties of a time-varying recursion," *IEEE Signal Processing Letters*, vol. 2, no. 5, pp. 95–96, May 1995.

[88] V. L. Stonick and S. T. Alexander, "Numerical methods for globally optimal adaptive IIR filtering," *Proceedings of 24th Asilomar Conference on Circuits, Systems & Computers*, Pacific Grove, USA, pp. 745–749, Nov. 1990.

[89] V. L. Stonick and S. T. Alexander, "Globally optimal rational approximation using homotopy continuation methods," *IEEE Transactions on Signal Processing*, vol. 40, no. 9, pp. 2358–2361, Sept. 1992.

[90] V. L. Stonick, "Time-varying performance surfaces for adaptive IIR filters: Geometric properties and implications for filter stability," *IEEE Transactions on Signal Processing*, vol. 43, no. 1, pp. 29–42, Jan. 1995.

[91] G. Strang, *Linear Algebra and its Applications*, Harcourt Brace Jovanovich, San Diego, USA, 3rd Edition, 1988.

[92] M. Tahernezhadi and L. Liu, "Real-time implementation of an IIR acoustic echo canceller on ADSP21020," *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, Detroit, USA, pp. 2723–2726, May 1995.

[93] Texas Instruments Inc., *TMS320C3x User's Guide*, 1992.

[94] J. R. Treichler, C. R. Johnson, Jr., and M. G. Larimore, *Theory and Design of Adaptive Filters*, Wiley, New York, USA, 1987.

[95] C.-P. J. Tzeng, "An adaptive offset cancellation technique for adaptive filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 5, pp. 799-803, May 1990.

[96] B. D. Van Veen and K. M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE Acoustics, Speech, and Signal Processing Magazine*, vol. 5, no. 2, pp. 4–24, Apr. 1988.

[97] J. E. Volder, "The CORDIC trigonometric computing technique," *IRE Transactions on Electronic Computers*, vol. EC-8, no. 3, pp. 330–334, Sept. 1959.

[98] S. A. White, "An adaptive recursive digital filter," *Proceedings of 9th Asilomar Conference on Circuits, Systems & Computers*, Pacific Grove, USA, pp. 21–25, Nov. 1975.

[99] B. Widrow and S. Stearns, *Adaptive Signal Processing*, Prentice-Hall, Englewood Cliffs, USA, 1985.

[100] M.-Y. Wu, "A note on stability of time-varying systems," *IEEE Transactions on Automatic Control*, vol. AC-19, p. 162, Apr. 1974.

[101] F. F. Yassa, "Optimality in the choice of the convergence factor for gradient-based adaptive algorithms," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-35, no. 1, pp. 48–59, Jan. 1987.

Appendix A

A.1 Introduction

The difference polynomial operator (DPO) notation is commonly used in the field of adaptive filters [25], [99] to simplify most of the somewhat intricate equations that are so often encountered in that area. However, probably due to its extremely simple characterization, the DPO notation can sometimes confuse even the most careful reader leading to dubious or incorrect interpretation of its true meaning. The situation can get even more complicate when polynomials with time-varying coefficients are used. In this appendix, in order to formalize the correct use and interpretation of this important mathematical tool, the DPO notation is presented and some of its interesting properties are thoroughly discussed.

In order to study the DPO notation, let us first introduce some basic concepts and assumptions necessary to simplify the following discussion. Throughout this text, let ℓ be the complex infinite-dimensional vector space of sequences [29], and let $x(n)$ and $y(n)$ be two elements of ℓ with n integer, i.e., $n \in \mathbb{Z}$. Let also $i, j \in \mathbb{Z}$ and c, d be complex scalars in $\mathbb{C}$. The strictly real case can be easily derived from the following discussion by forcing ℓ to be the vector space of real sequences and restricting c, d to the set of real numbers $\mathbb{R}$.

Definition A.1: The time-shift operator[1] q^{-i} is an operator that maps ℓ into ℓ and is described by

$$\begin{aligned} &q^{-i} : \ell \to \ell \\ &q^{-i}\{x(n)\} = x(n-i) \qquad \text{(A.1)} \end{aligned}$$

As the time-shift operator can be applied to any element in ℓ, this set is said to be the domain

[1] Although there is no strict distinction between the terms operator and function, it is usual to reserve the term operator for the case when the domain and image sets are infinite dimensional and to use the term function otherwise.

of this operator and this is represented by $\mathcal{D}[q^{-i}] = \ell$. Equivalently, as all sequences in ℓ can be obtained from the application of the time-shift operator on an element (not necessarily distinct) of ℓ, ℓ is also said to be the image of the operator, this being indicated by $\mathcal{I}[q^{-i}] = \ell$.

□

The quantity which an operator acts on is called an operand and, in this text, the curled brackets $\{.\}$ will be consistently used to designate the time-shift operation being applied to the given respective operand. Also, the negative sign on the definition of the time-shift operator is standardly used to make the relationship depicted in equation (A.1) appear causal for positive i. Notice, however, that as the parameter i belongs to the set of integer numbers $\mathbb{Z}$, the above definition is not restricted to causal time shifts. A special case of the above definition results when $i = 1$ which defines the unit delay operator characterized by $q^{-1}\{x(n)\} = x(n-1)$. Let us now introduce some important operations associated to the time-shift operator above presented.

Property A.2: The time-shift operator is a linear operator and then [29]

$$q^{-i}\{cx(n) + dy(n)\} = cx(n-i) + dy(n-i) \tag{A.2}$$

□

Property A.3: A more general result for the time-shift operator results from its own definition and can be written as

$$q^{-i}\{f(x(n), y(n))\} = f(x(n-i), y(n-i)) \tag{A.3}$$

where $f(.,.)$ is any given function defined in the discrete-time domain.

□

A very important fact associated to time-shift operators is that the set of these operators is itself a vector space, hereby denoted by $\mathcal{L}(\ell)$. This results from the fact that basic algebraic operations can be used with the operator, as seen below, instead of with the operand, as seen in the previous properties.

Property A.4:

The addition of time-shift operators is performed as follows

$$(q^{-i} + q^{-j})\{x(n)\} = q^{-i}\{x(n)\} + q^{-j}\{x(n)\} = x(n-i) + x(n-j) \tag{A.4a}$$

The multiplication of the time-shift operator by a scalar is given by

$$(cq^{-i})\{x(n)\} = c(q^{-i}\{x(n-i)\}) = cx(n-i) \tag{A.4b}$$

Equations (4a) and (4b) are often combined in just one equation as

$$(cq^{-i} + dq^{-j})\{x(n)\} = cx(n-i) + dx(n-j) \tag{A.5}$$

□

Property A.5: The concatenation of time-shift operators is performed as follows

$$q^{-j}\{q^{-i}\{x(n)\}\} = q^{-j}\{x(n-i)\} = x(n-i-j) = q^{-(i+j)}\{x(n)\} \tag{A.6}$$

□

Notice the similarity of this operation with the general rule for the multiplication of exponentials with identical bases. In fact, this property justifies the option of representing the time-shift operator by the q^{-i} notation. A similar relationship exists for the division of time-shift operators.

Property A.6: The division of time-shift operators follows the rule

$$\left(\frac{q^{-i}}{q^{-j}}\right)\{x(n)\} = q^{(-i-(-j))}\{x(n)\} = x(n-i+j) \tag{A.7}$$

□

A different but equivalent way to understand the division operation of two time-shift factors is illustrated by the following equation

$$\begin{aligned} y(n) = \left(\frac{q^{-i}}{q^{-j}}\right)\{x(n)\} &\Longleftrightarrow q^{-j}\{y(n)\} = q^{-i}\{x(n)\} \\ &\Longleftrightarrow y(n-j) = x(n-i) \end{aligned} \tag{A.8}$$

As it can be noticed from this equation, a rational operator of the form $\frac{q^{-i}}{q^{-j}}$ acting on a signal $x(n)$ can be seen as q^{-i} is operating on $x(n)$, while the denominator operator q^{-j} acts on the signal on the other side of the equality sign. This interpretation shows to be particularly useful in the next section when the difference polynomial operator notation is introduced.

Since $\mathcal{L}(\ell)$ is a vector space, there exists an identity operator, always denoted by I, which is defined by the relationship $I\{x(n)\} = x(n)$. For the time-shift operator space $\mathcal{L}(\ell)$, the identity operator is given by $I \equiv q^0$.

A.2 The Difference Polynomial Operator

An important extension of the time-shift operator is the difference polynomial operator (DPO) concept defined below.

Definition A.7: The extrapolation of equation (A.5) for several terms results in

$$\begin{aligned}\tilde{C}(q)\{x(n)\} &= (c_{n_c} + \ldots + c_1 q^{n_c-1} + c_0 q^{n_c})\{x(n)\} \\ &= c_{n_c} x(n) + \ldots + c_1 x(n + n_c - 1) + c_0 x(n + n_c) \qquad \text{(A.9)}\end{aligned}$$

where $c_0, c_1, \ldots, c_{n_c}$ are said to be the coefficients of the DPO $\tilde{C}(q)$ and n_c the order of $\tilde{C}(q)$.

□

In the context of adaptive filters, it is more common to use the causal form of the DPO notation described by

$$C(q) = q^{-n_c}\tilde{C}(q) = c_0 + c_1 q^{-1} + \ldots + c_{n_c} q^{-n_c} \qquad \text{(A.10)}$$

Although $C(q)$ is not genuinely a polynomial in the variable q, the form (A.10) is the one used throughout this dissertation and it will be referred to as a DPO. The reason for using the notation $C(q)$ for the causal DPO shown above instead of $C(q^{-1})$ is due to the close relationship existing between the DPO notation and the transfer function of a nonrecursive system. As a matter of fact, the DPO can be seen as a time-domain description of a discrete-time system equivalent to the transfer function concept in the frequency domain. This equivalence is clearly inferred from the time-shift property of the Z-transform described by [21]

$$Z\{q^{-i}\{x(n)\}\} = z^{-i}X(z) \qquad \text{(A.11)}$$

where $Z\{.\}$ denotes the Z-transform of a given discrete-time signal defined as [21]

$$Z\{x(n)\} = X(z) = \sum_{k=-\infty}^{+\infty} z^{-k}x(k) \qquad \text{(A.12)}$$

Another form to understand the dual relationship between the DPO and the transfer function associated to a nonrecursive system is given below.

Property A.8: Extending equation (A.12) to the concept of transfer function, it follows that

$$y(n) = H(q)\{x(n)\} \Longleftrightarrow Y(z) = H(z)X(z) \qquad \text{(A.13)}$$

where $Y(z)$ and $X(z)$ are respectively the Z-transforms of the $y(n)$ and $x(n)$ sequences, $H(z)$ is the transfer function between $X(z)$ and $Y(z)$, and $H(q) = H(z)\mid_{z=q}$ is the DPO acting of $x(n)$.

□

At this point, consider the following classification of operators which is useful when analyzing the operator equation $C(q)\{x(n)\} = y(n)$.

Definition A.9: An operator Q that maps the space vector X into the space vector Y is said to be injective if and only if for each element $y \in \mathcal{I}[Q]$ there is exactly one element $x \in \mathcal{D}[Q]$ such that $Q\{x\} = y$. The operator Q is called surjective if and only if its image set satisfies $\mathcal{I}[Q] = Y$ and then we say that Q maps X onto Y. Finally, Q is known as a bijective operator if and only if it is both injective and surjective.

□

The DPO definition implies that this class of operators is associated to straightforward extensions of Properties A.4 and A.5 and, therefore, these similar properties are not repeated here. Some other important properties of the DPO, however, are listed below, where $C(q) = c_0 + \ldots + c_{n_c}q^{-n_c}$ and $D(q) = d_0 + \ldots + d_{n_d}q^{-n_d}$ are assumed to be two generally distinct DPOs.

Property A.10: It is easy to verify that a given nonidentically null DPO, i.e., one with at least one nonzero coefficient, represents a bijective operator in the subspace of one-sided sequences $x(n)$, such that $x(n) = 0, \forall n < 0$.

□

Although the above property does not apply to the complete space of two-sided sequences ℓ, the above form does not represent any loss of generality when dealing with adaptive filters, as this class of systems is solely used in the processing of causal signals, i.e., one-sided sequences. A crucial consequence of the result stated in Property 10 is described below.

Property A.11: The inverse DPO exists and it is defined by

$$\left(\frac{1}{C(q)}\right)\{x(n)\} = C^{-1}(q)\{x(n)\} \tag{A.14}$$

in such a way that

$$C^{-1}(q)\{C(q)\{x(n)\}\} = (C^{-1}(q)C(q))\{x(n)\} = x(n) \tag{A.15}$$

□

Property A.12: The concatenation of direct and inverse DPOs is a commutative operation, i.e.

$$\left(\frac{1}{C(q)}\right)\left\{\frac{1}{D(q)}\{x(n)\}\right\}=\left(\frac{1}{D(q)}\right)\left\{\frac{1}{C(q)}\{x(n)\}\right\} \tag{A.16a}$$

$$C(q)\{D(q)\{x(n)\}\}=D(q)\{C(q)\{x(n)\}\} \tag{A.16b}$$

$$\begin{aligned}\left(\frac{C(q)}{D(q)}\right)\{x(n)\}&=\left(\frac{1}{D(q)}\right)\{C(q)\{x(n)\}\}\\ &=C(q)\left\{\frac{1}{D(q)}\{x(n)\}\right\}\end{aligned} \tag{A.16c}$$

□

The previous properties suggest that we can perform common algebraic operations directly on the DPOs and finally apply the resulting DPO to the signal sequence. As a matter of fact, this is one of the most important features of the DPO along with its extremely simple notation. In the next section, however, an extension of the DPO notation that does not follow these interchangeability properties is introduced.

A.3 The Time-Varying Difference Polynomial Operator

The time-varying difference polynomial operator is a type of DPO where the coefficients may vary in time, as seen in the following definition.

Definition A.13: The time-varying difference polynomial operator (TVDPO) is defined as

$$\begin{aligned}C(q,n)\{x(n)\}&=(c_0(n)+c_1(n)+\ldots+c_{n_c}(n)q^{-n_c})\{x(n)\}\\ &=c_0(n)x(n)+c_1(n)x(n-1)+\ldots+c_{n_c}(n)x(n-n_c)\end{aligned} \tag{A.17}$$

The TVDPO notation is largely used in the description, convergence analysis, and stability analysis of adaptive filter algorithms [25], [99]. Although the definition of a TVDPO is similar to the definition of the basic DPO, due to the time-varying characteristic of its coefficients the TVDPO does not possess the same properties of the DPO. This fact is stressed in Property A.14 below.

Property A.14: The concatenation of a TVDPO with either a DPO or a TVDPO is not a commutative operation.

□

Example A.15: In order to illustrate this last property, consider the two first-order TVDPOs $C(q,n) = 1 + c_1(n)q^{-1}$ and $D(q,n) = 1 + d_1(n)q^{-1}$, with $C(q,n) \neq D(q,n)$. Defining

$$e_1(n) = C(q,n)\{D(q,n)\{x(n)\}\} \tag{A.18a}$$

$$e_2(n) = D(q,n)\{C(q,n)\{x(n)\}\} \tag{A.18b}$$

it is easy to verify that

$$e_1(n) - e_2(n) = [c_1(n)d_1(n-1) - c_1(n-1)d_1(n)]\, x(n-2) \tag{A.19}$$

is generally different from zero, implying that $e_1(n)$ and $e_2(n)$ are two distinct sequences.

□

An important consequence of Property A.14 is that some regularly used notations for DPOs may lead to dubious interpretations when applied to a TVDPO. Some examples of it are given in Property A.16.

Property A.16:

$$\begin{aligned}(C(q,n))^2\{x(n)\} &= C(q,n)\{C(q,n)\{x(n)\}\}\\ &\neq C^2(q,n)\{x(n)\}\end{aligned} \tag{A.20a}$$

$$\begin{aligned}\left(\frac{C(q,n)}{D(q,n)}\right)\{x(n)\} &= \left(\frac{1}{D(q,n)}\right)\{C(q,n)\{x(n)\}\}\\ &\neq C(q,n)\left\{\left(\frac{1}{D(q,n)}\right)\{x(n)\}\right\}\end{aligned} \tag{A.20b}$$

□

Example A.17: To illustrate equation (A.20b), consider $C(q,n)$ and $D(q,n)$ as given in Example A.15. Assume that all sequences in this example are null for $n < 0$ and let

$$e_3(n) = C(q)\left\{\left(\frac{1}{D(q,n)}\right)\{x(n)\}\right\} \tag{A.21a}$$

$$e_4(n) = \left(\frac{1}{D(q,n)}\right)\{C(q)\{x(n)\}\} \tag{A.21b}$$

It can be readily verified that

$$e_3(2) - e_4(2) = [c_1(2)d_1(1) - c_1(1)d_1(2)]\, x(0) \tag{A.22}$$

From this equation, it is easy to see that the coefficients of $C(q,n)$ and $D(q,n)$ being dependent on time, results in $e_3(n)$ and $e_4(n)$ being two distinct sequences in general. Similar results are obtained for $[e_3(n) - e_4(n)]$ with different values of $n \geq 2$.

$\square$

When dealing with adaptive systems, the TVDPO is often used to represent the input/output relationship of adaptive filters. An approach to ensure that the differences between the left and right hand sides of the inequalities in Property A.16 approximate zero is to assume that the adaptable coefficients of the TVDPO are essentially constant, during at least a certain small number of iterations (usually at the order of the number of coefficients of the adaptive filter). This approach is called the small step approximation [5], [26], [28], as if an adaptation algorithm with a very small convergence step was used resulting in equivalently small variations in the values of the filter coefficients at each iteration.

Example A.18: To illustrate the mechanism of the small-step approximation, consider the sequence $y_1(n)$ given by

$$y_1(n) = q^{-1}\left\{\left(\frac{c_0(n) + c_1(n)q^{-1}}{1 + d_1(n)q^{-1}}\right)\{x(n)\}\right\} \tag{A.23}$$

and assume that the coefficients $(c_0(n); c_1(n); d_1(n))$ vary in time according to

$$c_0(n) = 2\ \tan^{-1}(ns-20);\ c_1(n) = -\frac{4}{3}\ \tan^{-1}(ns-16);\ d_1(n) = -\frac{10}{19}\ \tan^{-1}(ns-12) \tag{A.24}$$

with $n = 0, 1, \ldots, (40/s)$, where the parameter s is used as a time scale to control the speed of variation of those three coefficients. The higher the value of s, the faster the coefficients change in value, and vice-versa. Notice from (A.24) that the coefficients change in time following a respective sigmoid path characteristic of the arctangent function. The sigmoid function is almost constant in the beginning and at the end and it changes fast in its middle portion, leading to time-variant coefficients as the ones depicted in Figure A.1 for the case $s = 0.2$.

The small-step approximation for $y_1(n)$ is given by

$$y_2(n) = \left(\frac{c_0(n)q^{-1} + c_1(n)q^{-2}}{1 + d_1(n)q^{-1}}\right)\{x(n)\} \tag{A.25}$$

The main advantage of the small step approximation is the fact that only the current values of the adaptable coefficients are required to calculate the present value of a particular sequence.

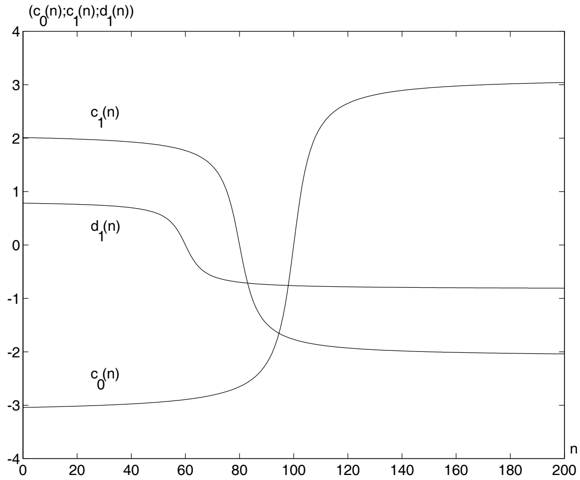


Figure A.1: Example A.18 - Time-varying coefficients $c_0(n)$, $c_1(n)$, and $d_1(n)$, with $s = 0.2$.

Consider input sequences $x(n)$ which are stochastic and uniformly distributed between zero and one for all the experiments in this example. Figure A.2 shows $y_1(n)$, $y_2(n)$, and the difference between these two signals. It can be observed that $(y_1(n) - y_2(n))$ is relatively small compared to the values of the signals, and that this difference assumes its largest value in the period when the coefficients are varying the most (compare with Figure A.1).

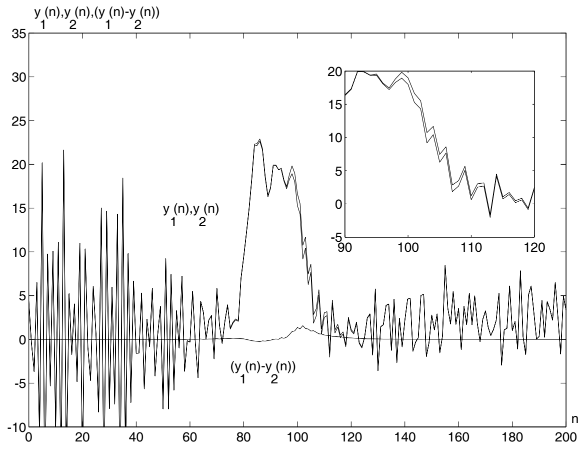


Figure A.2: Example A.18 - Signals $y_1(n)$, $y_2(n)$, and $(y_1(n) - y_2(n))$ for $0 \leq n \leq 200$ and $90 \leq n \leq 120$.

The validity of the small step approximation can be inferred from Figure A.3. In this figure, the average maximum value (over a set of 100 experiments) of the difference between the signal $y_1(n)$ and its approximation $y_2(n)$ was plotted as a function of the step s used to update the set of coefficients $(c_0(n); c_1(n); d_1(n))$ in time. Observe from this figure that as the step approximates zero (an extremal case for the small step approximation) the maximum value of the difference $(y_1(n) - y_2(n))$ becomes almost zero, as expected from the small step basic assumption.

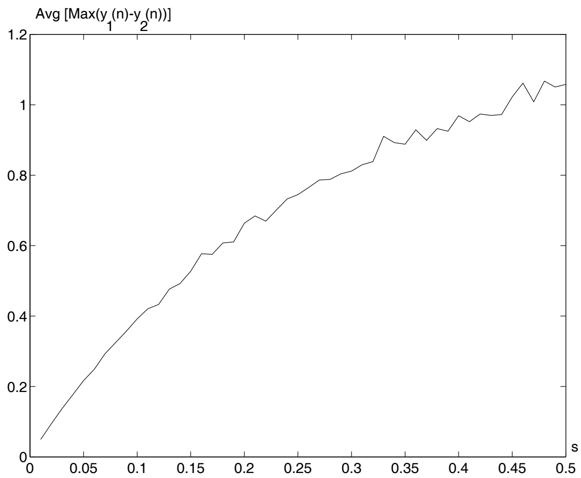


Figure A.3: Example A.18 - Average maximum value of the difference $(y_1(n) - y_2(n))$ as a function of the time scale s.

□

Another case where a TVDPO is considered as a DPO, i.e. without time-variant coefficients, occurs when the stationary properties of an adaptation algorithm are analyzed. In such cases, the adaptation process is supposed to have reached steady state and then the adaptive filter coefficients are assumed to be constant. Such analysis is very important in the characterization of the performance surface and the stationary points of adaptive filter algorithms such as the output error and equation error algorithms.

A.4 Conclusion

The time-shift operator and two important extensions of it, the difference polynomial operator (DPO) and the time-varying difference polynomial operator (TVDPO), were introduced. They significantly simplify the notations used to describe and analyze adaptive systems. It was shown that not all properties of the DPO can be extended to the TVDPO. However, some of these properties can still be used with TVDPOs when the so-called small step approximation is used or when steady-state analyses are performed.

VITA

Surname: Lima Netto — Given Name: Sergio

Place of Birth: Rio de Janeiro, RJ, Brazil

Educational Institutions Attended:

COPPE - Federal University of Rio de Janeiro — 1991 to 1992

Federal University of Rio de Janeiro — 1985 to 1991

Degrees Awarded:

M.Sc. — 1992

Elec. Eng. (*cum laude*) — 1991

Publications:

S. L. Netto and P. Agathoklis, "The composite squared error algorithm for adaptive IIR filtering," *Submitted to IEEE Transactions on Circuits and Systems–II: Analog and Digital Signal Processing*, Feb. 1996.

S. L. Netto and P. Agathoklis, "Efficient lattice realizations of adaptive IIR algorithms," *Submitted to IEEE Transactions on Signal Processing*, July 1995.

S. L. Netto and P. Agathoklis, "Extension of the equation error scheme for alternative realizations of adaptive IIR filters," *Proceedings of 37th Midwest Symposium on Circuits and Systems*, Rio de Janeiro, Brazil, Aug. 1995.

S. L. Netto, M. L. R. de Campos, A. Antoniou, and P. Agathoklis, "A stable RLC parameterization of the direct structure for adaptive IIR filters," *Proceedings of 37th Midwest Symposium on Circuits and Systems*, Rio de Janeiro, Brazil, Aug. 1995.

S. L. Netto and P. Agathoklis, "Composite algorithms for adaptive IIR filtering using lattice realization," *Proceedings of IEEE Pacific Rim Conference on Communications, Computers, and Signal Processing*, Victoria, Canada, pp. 189–192, May 1995.

S. L. Netto, P. S. R. Diniz, and P. Agathoklis, "Adaptive IIR filtering algorithms for system identification: A general framework," *IEEE Transactions on Education*, vol. 26, no. 1, pp. 54–66, Feb. 1995.

S. L. Netto and P. Agathoklis, "A new composite adaptive IIR algorithm," *Proceedings of 28th Asilomar Conference on Circuits, Systems & Computers*, Pacific Grove, USA, pp. 1506–1510, Oct./Nov. 1994.

S. L. Netto and P. Agathoklis, "A homotopy continuation mapping for the Steiglitz-McBride IIR adaptive algorithm," *Proceedings of IEEE International Symposium on Circuits and Systems*, London, England, pp. 2.57–2.60, May/June 1994.

S. L. Netto and P. S. R. Diniz, "New algorithms for adaptive IIR filtering and their properties," *Proceedings of Canadian Conference on Electrical and Computer Engineering*, Toronto, Canada, pp. TM4.13.1–TM4.13.4, Sept. 1992.

S. L. Netto and P. S. R. Diniz, "Composite algorithms for adaptive IIR filtering," *IEE Electronics Letters*, vol. 28, no. 9, pp. 886–888, Apr. 1992.

S. L. Netto, M. L. R. de Campos, and P. S. R. Diniz, "PACPAS: um pacote de apoio para processamento adaptativo de sinais," (in Portuguese) *Anais do 8o. Congresso Brasileiro de Automática*, Belem, Brazil, pp. 159–163, Sept. 1990.

Title of Dissertation:

On Algorithms, Structures, and Implementations of Adaptive IIR Filters

Author ______________________

Sergio Lima Netto

February 2, 1996