

1994

Linear and nonlinear adaptive digital filters and their applications.

Qiu Ping, Li
University of Windsor

Follow this and additional works at: <http://scholar.uwindsor.ca/etd>

Recommended Citation

Li, Qiu Ping, "Linear and nonlinear adaptive digital filters and their applications." (1994). *Electronic Theses and Dissertations*. Paper 2615.

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.



National Library
of Canada

Bibliothèque nationale
du Canada

Acquisitions and
Bibliographic Services Branch

Direction des acquisitions et
des services bibliographiques

395 Wellington Street
Ottawa, Ontario
K1A 0N4

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Ysa / le / Aca / r / e / s / s / e

Q / u / e / s / t / i / o / n / s

NOTICE

AVIS

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

If pages are missing, contact the university which granted the degree.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Canada

**LINEAR AND NONLINEAR ADAPTIVE DIGITAL FILTERS
AND THEIR APPLICATIONS**

by

Qiu Ping LI

A Dissertation
Submitted to the Faculty of Graduate Studies and Research
Through the Department of Electrical Engineering
in Partial Fulfilment
of the Requirements for the Degree of
Doctor of Philosophy
at the University of Windsor

Windsor, Ontario, Canada

1994



National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file / Votre référence

Our file / Notre référence

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-93285-6

Canada

©

1994 Qiu Ping LI

ABSTRACT

**LINEAR AND NONLINEAR ADAPTIVE DIGITAL FILTERS
AND THEIR APPLICATIONS**

By

Qiu Ping LI

Adaptive digital signal processing has matured to the point where it now constitutes an important part of statistical signal processing. However, high speed adaptive digital filtering and nonlinear adaptive digital filtering are yet to be developed.

In this dissertation, a family of high speed linear adaptive digital filters, which are suitable for parallel realization, are first presented. They are (1) Delayed N-path adaptive finite impulse response (DNA-FIR) digital filters; (2) Delayed N-path adaptive linear phase finite impulse response (DNALP-FIR) digital filters; (3) Delayed N-path equation-error based adaptive infinite impulse response (DNEEBA-IIR) digital filters. By using multiple digital signal processors in parallel, the processing speed of this type of filters can be increased compared with the conventional realizations. They are contrived to be useful for general applications in adaptive digital signal processing. Comparison studies have been conducted among the proposed DNA-FIR digital filter, the block implementation of adaptive finite impulse response (BIA-FIR) digital filter, and the

conventional adaptive finite impulse response (CA-FIR) digital filter. It has been shown that the processing speed of the proposed DNA-FIR digital filter is N times that of the BIA-FIR digital filter (given block length N and time domain realization).

Then a nonlinear delayed N -path adaptive finite impulse response (NDNA-FIR) digital filter is developed, the parallel structure of which lends itself to high speed implementation. This NDNA-FIR digital filter compares favorably with the CA-FIR digital filter and the nonlinear median filter when applied to broadband noise cancellation. The last part of this dissertation describes the structure and its adaptive algorithm of a nonlinear adaptive infinite impulse response (IIR) digital filter. This IIR digital filter is bounded input bounded output (BIBO) stable. Based on this structure, an individual adaptation scheme is incorporated into the adaptive algorithm to improve the convergence speed.

To my parents: Zhong-Zhen LI and Mi-Mi WANG

and

to my parents-in-law: Ying-Hua LIANG and Ying JIN

ACKNOWLEDGEMENTS

I am sincerely grateful to my advisor Dr. H. K. Kwan for valuable guidance, advice, and encouragement throughout the progress of this dissertation, and for financial support through the Natural Science and Engineering Research Council (NSERC) of Canada. Thanks are also due to my department reader, Dr. G. A. Jullien, and Dr. W. C. Miller, and my outside department reader, Dr. S. Bandyopadhyay of my doctoral committee for constructive comments. I would like to thank the external examiner of this dissertation, Dr. W. B. Mikhael of the University of Central Florida for useful comments. I would also like to thank Dr. G. R. G. Raju, the head of the department, and Dr. M. Ahmadi for encouragement in all these years of my Ph.D. program. I would like to acknowledge the NSERC of Canada for awarding me the NSERC Postgraduate Scholarship for two years, and the University of Windsor for awarding me the University of Windsor Tuition Scholarship for two years.

TABLE OF CONTENTS

ABSTRACT	iv
DEDICATION	vi
ACKNOWLEDGEMENTS	vii
LIST OF TABLES	xiii
LIST OF ILLUSTRATIONS	xv
Chapter 1 INTRODUCTION	1
1.1 Motivations and Impact of this Research	1
1.2 Earlier Work on High Speed Adaptive Digital Filtering and Nonlinear Digital Filtering for Noise Cancellation	4
1.2.1 Block Implementation of Adaptive Digital Filters	5
1.2.2 Multirate Adaptive Digital Filtering	5
1.2.3 High Sampling Rate Adaptive Filters with Lattice Structures	6
1.2.4 Nonlinear Digital Filtering for Noise Cancellation	6
1.3 Contributions of this Dissertation	7
1.3.1 Delayed N-Path Adaptive Finite Impulse Response (DNA-FIR) Digital Filter	7
1.3.2 Delayed N-Path Adaptive Linear Phase Finite Impulse Response (DNALP-FIR) Digital Filter	8
1.3.3 Delayed N-Path Equation-Error based Adaptive Infinite Impulse Response (DNEEBA-IIR) Digital Filter	8

1.3.4	Comparisons between Block Implementation Adaptive FIR (BIA-FIR) Digital Filter and Delayed N-Path Adaptive FIR (DNA-FIR) Digital Filter	8
1.3.5	Nonlinear Delayed N-Path Adaptive Finite Impulse Response (NDNA-FIR) Digital Filter	9
1.3.6	A New Nonlinear Adaptive IIR (NA-IIR) Digital Filter	10
1.4	Organization of this Dissertation	10
Chapter 2	DELAYED N-PATH ADAPTIVE DIGITAL FILTERS	13
2.1	Delayed N-Path Adaptive FIR (DNA-FIR) Digital Filter	13
2.1.1	Introduction	13
2.1.2	Structure and Adaptive Algorithm of DNA-FIR Digital Filter	14
2.1.3	Simulation Results	19
2.1.4	Conclusions	25
2.2	Delayed N-Path Adaptive Linear Phase FIR (DNALP-FIR) Digital Filter	25
2.2.1	Introduction	25
2.2.2	Structure of Delayed N-Path Linear Phase FIR Digital Filter	27
2.2.2.1	Even M Delayed N-Path Linear Phase FIR Digital Filter Structure	28
2.2.2.2	Odd M Delayed N-Path Linear Phase FIR Digital Filter Structure	30
2.2.2.3	Example	33
2.2.3	Adaptive Algorithm of DNALP-FIR Digital Filter	36
2.2.3.1	Even M DNALP-FIR Digital Filter	36
2.2.3.2	Odd M DNALP-FIR Digital Filter	37

3.2.4	Comparisons between BIA-FIR and DNA-FIR Digital Filters	99
3.2.4.1	Computation Complexity	99
3.2.4.2	System Throughput Rate	102
3.2.4.3	Convergence Speed of Adaptive Algorithm	104
3.2.4.4	System Convergence Time	105
3.2.4.5	Performance Index	106
3.2.4.6	Summary	108
3.3	Simulation Results	109
3.4	Conclusions	114
Chapter 4	NONLINEAR DELAYED N-PATH ADAPTIVE FIR DIGITAL FILTER	116
4.1	Introduction	116
4.2	Structure of Nonlinear Delayed N-Path Adaptive FIR (NDNA-FIR) Digital Filter	119
4.2.1	Structure of NDNA-FIR Digital Filter	121
4.2.2	Sigmoid Function	124
4.3	Adaptive Algorithm of Nonlinear Delayed N-Path Adaptive FIR (NDNA-FIR) Digital Filter	130
4.3.1	Updates of Weights $h_{i,k}(l)$	133
4.3.2	Updates of Parameters $\alpha_{i,k}$ and α_k	134
4.3.3	Updates of Parameters $G_{i,k}$ and G_k	135
4.3.4	Updates of Parameters $t_{i,k}$ and t_k	136
4.4	Simulation Results	137
4.5	Conclusions	159
Chapter 5	A NEW NONLINEAR ADAPTIVE IIR DIGITAL FILTER	160

5.1	Introduction	160
5.2	Structure and Adaptive Algorithm of Nonlinear Adaptive IIR Digital Filter (NAIIRDF)	163
5.2.1	Structure of NAIIRDF	163
5.2.2	Adaptive Algorithm of NAIIRDF	167
5.3	Analyses of Stability and Time Domain Performance of Nonlinear Adaptive IIR Digital Filter (NAIIRDF)	170
5.3.1	Performance Analysis of $G(k)$ and $y(k)$	171
5.3.2	Performance Analysis of $t(k)$	174
5.3.3	Performance Analysis of $a_i(k)$ and $b_j(k)$	177
5.4	Individual Parameter Adaptation Scheme for Nonlinear Adaptive IIR Digital Filter (NAIIRDF)	181
5.5	Simulation Results	185
5.6	Conclusions	206
Chapter 6	CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH	207
6.1	Conclusions of this Dissertation	207
6.2	Suggestions for Future Research	209
	BIBLIOGRAPHY	211
	VITA AUCTORIS	219

LIST OF TABLES

Table 2.1:	Convergence performance in terms of number of iterations of DNA-FIR structures without measurement noise	20
Table 2.2:	Convergence performance in terms of number of iterations of DNA-FIR structures with measurement noise	21
Table 2.3:	Coefficient configuration for the even M case and N-path realization	29
Table 2.4:	Coefficient configuration for the odd M case and N-path realization	32
Table 2.5:	Coefficient configuration for the example of 19th-order and 3-path realization	34
Table 2.6:	Convergence performance in terms of number of iterations of DNEEBA-IIR structures without measurement noise	68
Table 2.7:	Convergence performance in terms of number of iterations of DNEEBA-IIR structures with measurement noise	69
Table 2.8:	Effect of increased filter order on number of convergent iterations of DNEEBA-IIR structures	76
Table 2.9:	Convergence performance in terms of number of iterations of 22nd-order DNA-FIR and 11th-order DNEEBA-IIR structures	77
Table 3.1:	Analytical comparison results	109
Table 3.2:	Simulation comparison results	114
Table 4.1:	Simulation results of Example 1	158
Table 4.2:	Simulation results of Example 2	158
Table 5.1	Simulation results of comparison studies between NAIIRDF and LAIIRDF	199

Table 5.2	Parameters and coefficients of NAIIRDF of Example 1 after convergence	200
Table 5.3	Parameters and coefficients of NAIIRDF of Example 2 after convergence	201

LIST OF ILLUSTRATIONS

- Fig. 2.1: Block diagram of adaptive system modelling 15
- Fig. 2.2: DNA-FIR digital filter structure 17
- Fig. 2.3: Average squared output error of D2A-FIR modelling of $A_1(z^{-1})$ 22
- Fig. 2.4: Average squared output error of D5A-FIR modelling of $A_2(z^{-1})$ 23
- Fig. 2.5: Average squared output error of D8A-FIR modelling of $A_3(z^{-1})$ 24
- Fig. 2.6: Block diagram of DNALP-FIR digital filter for adaptive noise cancellation 31
- Fig. 2.7: Structure of a 19th-order delayed 3-path even symmetry linear phase FIR digital filter 35
- Fig. 2.8: Application of DNALP-FIR digital filter to adaptive noise cancellation of sinusoidal signal (11th-order, 4-path, even symmetry)
- (a) Original sinusoidal signal 41
 - (b) Corrupted input signal 42
 - (c) Recovered sinusoidal signal 43
 - (d) Squared output error 44
- Fig. 2.9: Application of DNALP-FIR digital filter to adaptive noise cancellation of sawtooth signal (11th-order, 2-path, even symmetry)
- (a) Original sawtooth signal 45
 - (b) Corrupted input signal 46
 - (c) Recovered sawtooth signal 47
 - (d) Squared output error 48
- Fig. 2.10: Block diagram of DNALP-FIR digital filter for adaptive system modelling 49
- Fig. 2.11: Output error of DNALP-FIR digital filter for adaptive system modelling (11th-order, 2-path, even symmetry) 54
- Fig. 2.12: Output error of DNALP-FIR digital filter for adaptive system modelling (24th-order, 5-path, even symmetry) 55

Fig. 2.13:	Block diagram of equation-error based adaptive IIR digital filter	58
Fig. 2.14:	DNEEBA-IIR digital filter structure	62
Fig. 2.15:	(a) Average squared equation-error of D2EEBA-IIR modelling of $H_1(z^{-1})$	70
	(b) Average squared real error of D2EEBA-IIR modelling of $H_1(z^{-1})$	71
Fig. 2.16:	(a) Average squared equation-error of D3EEBA-IIR modelling of $H_2(z^{-1})$	72
	(b) Average squared real error of D3EEBA-IIR modelling of $H_2(z^{-1})$	73
Fig. 2.17:	(a) Average squared equation-error of D4EEBA-IIR modelling of $H_3(z^{-1})$	74
	(b) Average squared real error of D3EEBA-IIR modelling of $H_3(z^{-1})$	75
Fig. 3.1:	Conventional adaptive FIR digital filter	83
Fig. 3.2:	Block implementation of adaptive FIR digital filter	86
Fig. 3.3:	Delayed N-path adaptive FIR digital filter	90
Fig. 4.1:	Signal waveform with sharp edges corrupted by random noise	118
Fig. 4.2:	Block diagram of LA-FIR digital filter	120
Fig. 4.3:	Block diagram of NDNA-FIR digital filter	122
Fig. 4.4:	Nonlinear operator of the sigmoid function	123
Fig. 4.5:	Examples of input-output characteristics of sigmoid function with different parameter values of α , G , and t	
	(a) $\alpha = 3$, $G = 2$, and $t = 0$	126
	(b) $\alpha = 10$, $G = 2$, and $t = 0$	127
	(c) $\alpha = 0.5$, $G = 4.2$, and $t = 0$	128
	(d) $\alpha = 0.1$, $G = 20$, and $t = 0$	129

- Fig. 4.6: Simulation results in Example 1
- (a) Original signal $s(k)$ 139
 - (b) Corrupted input signal $x(k)$ 140
 - (c) Converged output signal $y(k)$ of NDNA-FIR digital filter 141
 - (d) Squared output error $e^2(k)$ of NDNA-FIR digital filter 142
 - (e) Converged output signal $y(k)$ of LA-FIR digital filter of order 61 143
 - (f) Converged output signal $y(k)$ of LA-FIR digital filter of order 82 144
 - (g) Converged output signal $y(k)$ of LA-FIR digital filter of order 121 145
 - (h) Output signal of NLMED digital filter 146

- Fig. 4.7: Simulation results in Example 2
- (a) Original signal $s(k)$ 149
 - (b) Corrupted input signal $x(k)$ 150
 - (c) Converged output signal $y(k)$ of NDNA-FIR digital filter 151
 - (d) Squared output error $e^2(k)$ of NDNA-FIR digital filter 152
 - (e) Converged output signal $y(k)$ of LA-FIR digital filter of order 61 153
 - (f) Converged output signal $y(k)$ of LA-FIR digital filter of order 82 154
 - (g) Converged output signal $y(k)$ of LA-FIR digital filter of order 121 155
 - (h) Output signal of NLMED digital filter 156

Fig. 5.1: Block diagram of adaptive digital system identification 163

Fig. 5.2: Structures of two IIR digital filters

- (a) Linear IIR digital filter 165
- (b) Nonlinear IIR digital filter 165

Fig. 5.3:	Nonlinear operator of sigmoid function	166
Fig. 5.4:	Simulation results of modelling a 2nd-order linear IIR digital filter of Example 1 using NAIIRDF	
	(a) Squared output error	188
	(b) Impulse response of the unknown plant	189
	(c) Impulse response of the adaptive filter after convergence	190
Fig. 5.5:	Simulation results of modelling a 2nd-order linear IIR digital filter of Example 1 using LAIRDF in direct form	
	(a) Squared output error	191
	(b) Impulse response of the adaptive filter after convergence	192
Fig. 5.6:	Simulation results of modelling a 4th-order linear IIR digital filter of Example 2 using NAIIRDF	
	(a) Squared output error	194
	(b) Impulse response of the unknown plant	195
	(c) Impulse response of the adaptive filter after convergence	196
Fig. 5.7:	Simulation results of modelling a 4th-order linear IIR digital filter of Example 2 using LAIRDF in direct form	
	(a) Squared output error	197
	(b) Impulse response of the adaptive filter after convergence	198
Fig. 5.8:	Simulation results of modelling a 4th-order linear IIR digital filter of Example 3 using NAIIRDF	
	(a) Squared output error	203
	(b) Impulse response of the unknown plant	204
	(c) Impulse response of the adaptive filter after convergence	205

Chapter 1

INTRODUCTION

The earliest work on adaptive digital filtering may be traced back to the late 1950s [1]. At present, the subject has matured to the point where it now constitutes an important part of statistical signal processing [2]. The intelligent features of an adaptive digital filter, namely, the ability to operate satisfactorily in an unknown environment and also to track time variations of input statistics, make the adaptive filter a powerful device for signal processing applications. In fact, adaptive digital filtering has been successfully applied in such diverse fields as communications, radar, sonar, seismology, and biomedical engineering. However, more efforts are still needed on the studies of high speed adaptive digital signal processing and nonlinear adaptive digital filtering.

1.1 Motivations and Impact of this Research

The past few years have visualized two aspects which have great impacts on the area of adaptive digital signal processing:

(i) Computationally intensive problems are involved in a variety of applications which would require large amounts of data to be processed in a very short time period. For example, computer vision images of 1000x1000 data elements have to

be processed within 16.7 milliseconds. Thus processing rates of 10 to 1000 GOPS (10^9 operations per second) and input data rates approaching 1 gigabyte per second are required. Many other examples could be found in biomedical image processing, robot vision, weather prediction, and automation algorithms [3-4].

(ii) In the past decade, the unprecedented advances in very large scale integration (VLSI) technology have stimulated modern digital signal processing. While the designers of modern integrated circuitry have continually endeavored to provide faster products, the development of dedicated computing structures have provided alternative procedures for improving computing speed [5-6]. More and more digital systems are realized by parallel digital signal processors (DSPs) or digital signal processing chips to obtain high speed performance.

The above mentioned two aspects inevitably present new topics for adaptive digital signal processing. Indeed, it is the challenge of researchers in the field to define specific adaptive implementations which are efficiently matched to VLSI architectures so as to provide efficient processing at high data rates. Although there exist various adaptive filter structures and algorithms [2,7-9], high speed structures and algorithms for adaptive digital filtering have yet to be developed, especially for signals containing significant high frequency components.

Continuing advances in VLSI technology have made it attractive to interconnect many inexpensive DSPs to build a powerful processing system in a cost-effective way.

Although there exist some high speed DSPs, upper limits always exist on their throughput rates due to the limitation of manufacturing technology. For non-adaptive digital filtering, high speed processing can be achieved by using delayed two-path FIR and IIR digital filters [10], and delayed N-path FIR and IIR digital filters (for $N \geq 2$) [11-13]. Refined versions of [11-12] can be found in references 14-18. In this dissertation, a class of high speed linear adaptive digital filters has been presented, which is realized based on the delayed N-path structure and is suitable for parallel realization using multiple DSPs to improve system throughput rate. They are delayed N-path adaptive finite impulse response (DNA-FIR) digital filters, delayed N-path adaptive linear phase finite impulse response (DNALP-FIR) digital filters, and delayed N-path equation-error based adaptive infinite impulse response (DNEEBA-IIR) digital filters. Comparison studies have also been conducted among the proposed DNA-FIR digital filter, the block implementation of adaptive finite impulse response (BIA-FIR) digital filter, and the conventional adaptive finite impulse response (CA-FIR) digital filters.

The primary advantage of linear filtering is its simplicity in design and implementation. However, there are some circumstances where the performances of linear digital filtering are not satisfactory and in which nonlinear digital filtering will do a better job [19]. For this reason, a nonlinear delayed N-path adaptive finite impulse response (NDNA-FIR) digital filter is developed for broadband noise cancellation.

In developing high speed adaptive infinite impulse response (IIR) digital filters,

it was found that stability problem had always been an obstacle in the applications of direct-form IIR adaptive digital filters. This situation stimulated the author to develop a new direct-form IIR adaptive digital filter structure which is bounded input bounded output (BIBO) stable. In this regard, the structure and its adaptive algorithm of a nonlinear adaptive IIR digital filter is presented in the last part of this dissertation.

Before we go further, we would like to have a brief look at what are the approaches in the earlier work on high speed adaptive digital filtering and nonlinear digital filtering for noise cancellation.

1.2 Earlier Work on High Speed Adaptive Digital Filtering and Nonlinear Digital Filtering for Noise Cancellation

In an adaptive digital filter, two processes are present:

- (i) The adaptive or training process, which addresses the automatic adjustment of the filter coefficients;
- (ii) The filtering process, which uses the adaptive filter coefficients to produce an output by weighting the input signals.

The realization of adaptive digital filtering at a high sampling rate is usually made more difficult by the first process compared with the conventional digital signal processing, since the training process implies an inherent feedback of an error signal back to the adaptation of the coefficients [20]. For example, adaptive transversal filters have an

inherent sampling rate limitation for a given speed of hardware due to the feedback of the residual error to the adaptation of the individual stages. This feedback destroys the opportunity to pipeline the filter realization [20]. So far, very few high speed adaptive digital filters for general applications have been presented.

1.2.1 Block Implementation of Adaptive Digital Filters

One of the major approaches of high speed adaptive digital filtering is block implementation of adaptive digital filters [21-25]. There are two approaches in block adaptive digital filtering. One is to incorporate block updating strategies whereby FFT algorithms efficiently perform the filter convolution and the gradient correlation. This technique reduces the complexity because the filter output and the adaptive weights are computed only after a large block of data has been accumulated. Its formulation, however, made it difficult to study the effects of the feedback in the implementation [25-26]. The other is to realize the linear convolution in the time domain by using parallel processors. The data sampling rate of the block adaptive digital filters was increased by a factor of the block size; however, they suffered slower tracking capability as the block size increased [20-23]. In both cases, the block implementation of adaptive digital filtering introduced an end-to-end delay that could be a problem in applications such as telecommunications [26].

1.2.2 Multirate Adaptive Digital Filtering

In this approach, the adaptive processing is performed at a lower sampling rate

than that of the incoming data, thus reducing the computational complexity and obtaining a higher system throughput rate. However, this high throughput rate is obtained by changing the sampling rate of the input signal which is subject to a certain limitation on its frequency band [27-29]. This is also true in adaptive digital filtering using multirate techniques, where much effort has been directed to annihilate both aliasing and spectral gaps in the output [30-31]. Similar to the block adaptive digital filtering, an end-to-end delay would also be involved which could cause a problem in applications such as telecommunications [26].

1.2.3 High Sampling Rate Adaptive Filters with Lattice Structures

In [20], high sampling rate adaptive filters were realized based on the lattice structure. This type of filters allowed arbitrarily high sampling rates for a given speed of hardware at the expense of parallel hardware implementation. The input-output characteristics of the adaptive algorithm has not been modified, and hence the convergence and tracking capability of the adaptive filter would not be affected. However, the lattice structure employed here suggested that the overall computation complexity of the whole system was increased.

1.2.4 Nonlinear Digital Filtering for Noise Cancellation

As seen from [32-36], nonlinear digital filters have played an important role in noise cancellation and edge preservation. Nonlinear order statistic digital filters have been presented in [32] [35-36] and two types of neural network digital filters have been

presented in [33-34]. However, they are not adaptive digital filters in the sense that they could not track the input signal without knowing all the input signal samples as a priori [32] [35-36], or they need to be trained first and then fixed when performing noise cancellation on a certain type of signals as in [33-34].

1.3 Contributions of this Dissertation

1.3.1 Delayed N-Path Adaptive Finite Impulse Response (DNA-FIR) Digital Filter

In a DNA-FIR digital filter, the original transfer function is decomposed into N sub-functions of $1/N$ of the length of the original transfer function. Each sub-function can then be realized directly in the form of N parallel paths. Each of these N parallel paths can further be realized by N parallel DSP blocks. In so doing, each of the N parallel DSPs in a path is allowed to operate at a sampling rate N times slower than that of the input signal for realizing an identical set of variable coefficient sub-filters and the associated adaptive algorithm. The number of coefficients to be updated in each of the N^2 DSPs of a DNA-FIR digital filter is about $1/N$ times that of the total number of coefficients. Therefore, given N^2 DSPs of a certain maximum speed, the maximum possible throughput rate of a DNA-FIR digital filter can be N^2 times that of the same adaptive FIR digital filter realized directly using one such DSP. Simulation results obtained in applications to adaptive system modelling have shown very good performance in terms of convergence speed and output error residuals.

1.3.2 Delayed N-Path Adaptive Linear Phase Finite Impulse Response (DNALP-FIR) Digital Filter

By exploiting the symmetry property of a linear phase FIR digital filter, the above mentioned decomposition techniques can be used to obtain DNALP-FIR digital filters. The maximum possible throughput rate of a DNALP-FIR digital filter can be improved by $2N^2$ times, given N^2 DSPs of a certain maximum speed, which has an improvement of a factor of 2 compared with the DNA-FIR structures. Simulations are performed by using the DNALP-FIR digital filters for adaptive noise cancellation and adaptive system modelling.

1.3.3 Delayed N-Path Equation-Error based Adaptive Infinite Impulse Response (DNEEBA-IIR) Digital Filter

The above mentioned decomposition technique can also be applied to an equation-error based adaptive IIR digital filter to obtain the DNEEBA-IIR digital filter structure. In a DNEEBA-IIR digital filter, each of the numerator and the denominator polynomials can be realized by two DNA-FIR structures respectively. By using $2N^2$ DSPs in parallel, the maximum possible throughput rate can be improved by $2N^2$ times compared with a conventional equation-error based adaptive IIR digital filter using one DSP. Simulation results are given to show the performances of the proposed structure for system modelling.

1.3.4 Comparisons between Block Implementation of Adaptive FIR (BIA-

FIR) Digital Filter and Delayed N-Path Adaptive FIR (DNA-FIR) Digital Filter

Based on the conventional adaptive FIR digital filter first presented by Widrow [7], comparison studies have been conducted in the time domain between the block implementation of adaptive FIR digital filter and the delayed N-path adaptive FIR digital filter, in which both structures are realized by parallel digital signal processors. Both structures feature improvement in the system throughput rate compared with the conventional adaptive FIR digital filter, however, the speed gains have been degraded by a certain factor because of the slow convergence of the adaptive algorithm. Simulations have been performed to verify the conclusions obtained through theoretical analyses.

1.3.5 Nonlinear Delayed N-Path Adaptive Finite Impulse Response (NDNA-FIR) Digital Filter

The structure of a NDNA-FIR digital filter is obtained by decomposing a FIR digital filter into N delayed parallel sub-filters which facilitate high speed implementation. Adjustable bipolar sigmoid functions are placed at the output of each sub-filter as well as after the summation of all sub-filters. Applications of this type of nonlinear adaptive digital filters are found for cancelling broadband noise while preserving the sharp edges of a signal. Simulation results show that the NDNA-FIR digital filter outperforms the linear adaptive FIR digital filter and the nonlinear median filter in broadband noise cancellation in terms of convergence speed and the average squared difference between original and recovered signals.

1.3.6 A New Nonlinear Adaptive IIR Digital Filter (NAIRDF)

The structure and its adaptive algorithm of a nonlinear adaptive IIR digital filter is presented. With the use of a nonlinear sigmoid function at the output, the filter is BIBO stable. The stability analysis has been conducted together with studies on the time domain behavior of filter parameters and coefficients. To improve the convergence speed, the individual parameter adaptation scheme is developed into the adaptive algorithm to optimally adjust each parameter at every iteration. Based on the application in adaptive system identifications of linear IIR digital filters, simulation results obtained show that the new nonlinear adaptive IIR digital filters outperform those of linear adaptive IIR digital filters in terms of convergence speed and final output error residuals.

1.4 Organization of this Dissertation

In Chapter 2, a family of linear delayed N-path adaptive digital filters have been presented. There are four sections in this chapter. Section 2.1 presents the delayed N-path adaptive FIR digital filter, its structure, adaptive algorithm, and applications to adaptive system identifications. The delayed N-path linear phase adaptive FIR digital filter is given in Section 2.2. In this section, design of the structure and derivation of the adaptive algorithm have been detailed. Applications of this type of digital filters are found in adaptive noise cancellation and system modelling. Section 2.3 describes the delayed N-path equation-error based adaptive IIR digital filter. Firstly, it is illustrated that how a DNA-FIR digital filter structure developed in the first section can be adapted

for the realization of the proposed DNEEBA-IIR digital filter structure. Then the adaptive algorithm is derived with a simple variable step coefficient adjustment scheme. Applications of the DNEEBA-IIR digital filters are found in system modelling of both IIR and FIR digital filters. A short conclusion is given in the last section.

Chapter 3 presents comparison studies between block adaptive FIR digital filters and delayed N-path adaptive FIR digital filters. Four sections are included in this chapter. Following a brief introduction of Section 3.1, theoretical analyses and comparison studies are performed in Section 3.2 between block adaptive FIR digital filters and delayed N-path adaptive FIR digital filters. In this section, we first briefly describe the concepts of conventional adaptive FIR digital filters. Based on these concepts, the performances of the delayed N-path adaptive FIR digital filter are analyzed and compared with the block adaptive FIR digital filters. In Section 3.3, simulation results are provided which verify the theoretical analyses obtained. Finally, conclusions are outlined in the fourth section.

The basic advantage of linear digital filtering is its simplicity in design and implementation. In some circumstances, the performances of nonlinear digital filtering are better than those of the linear digital filtering. In Chapter 4, a nonlinear delayed N-path adaptive digital filter is presented for broadband noise cancellation. Following an introduction section, the structure is described in Section 4.2 and the adaptive algorithm is derived in Section 4.3. Simulation results in broadband noise cancellation are given

in Section 4.4, which is followed by the last section on conclusions.

In Chapter 5, a BIBO stable nonlinear adaptive IIR digital filter is presented. The motivation of this piece of research is expressed in the first section. In Section 5.2, we first present the structure of this nonlinear adaptive digital filter and then derive the adaptive algorithm for this structure. Analyses of stability and time domain performance of the coefficients and parameters are performed in Section 5.3. An individual parameter adaptation scheme for this adaptive digital filter is developed in Section 5.4. Simulation results for the adaptive system modelling and conclusions are given in the last two sections respectively.

Chapter 2

DELAYED N-PATH ADAPTIVE DIGITAL FILTERS

In this chapter, we present a family of linear delayed N-path adaptive digital filters for high speed adaptive digital signal processing. Structures and adaptive algorithms are developed for each member of the family. Simulation results are performed in applications to adaptive system identifications and adaptive noise cancellation.

2.1 Delayed N-Path Adaptive FIR (DNA-FIR) Digital Filter

2.1.1 Introduction

Adaptive FIR digital filters are fundamental to adaptive digital signal processing. Their behavior and adaptation strategies, as well as their implementations, have received considerable attention [2,7-8]. The merit of FIR digital filters lies in that they are easy to be realized and they are inherently stable which is not the case for IIR digital filters. Adaptive FIR digital filters usually provide good convergence performance. With a FIR digital filter of modest order, the misadjustment is fairly small and a reasonably fast convergence speed can be obtained. So far, adaptive FIR digital filters have found many

practical applications such as seismic exploration, signal detection, parameter estimation, echo cancellation and noise cancellation [2,7-8,37-38].

High speed adaptive FIR digital filtering can be obtained based on the structure of delayed multi-path realization of digital filters [16-18]. The resultant digital filters are called delayed N-path adaptive FIR (DNA-FIR) digital filters. There are a number of adaptive algorithms which can be used in the adaptation of an adaptive filter. Among these, the least-mean-square (LMS) algorithm [2,7-8] is popular because of its simplicity and ease of computation. The LMS algorithm is used here to illustrate the performance of the proposed DNA-FIR digital filters. Simulation results using the DNA-FIR digital filters are given following the derivation of its structure and adaptive algorithm.

2.1.2 Structure and Adaptive Algorithm of DNA-FIR Digital Filter

In many circumstances, the approach of adaptive filtering can be viewed from the point of system identification or system modelling (i.e., imitating the behaviors of a physical dynamic system which can be regarded as an unknown system, or "plant"), as shown in Fig. 2.1. In the following, we will discuss the problem based on this type of modelling structure.

Consider the transfer function of an adaptive FIR digital filter at a discrete-time k as

$$H_k(z^{-1}) = \sum_{m=0}^{M-1} h_k(m)z^{-m} \quad (2.1)$$

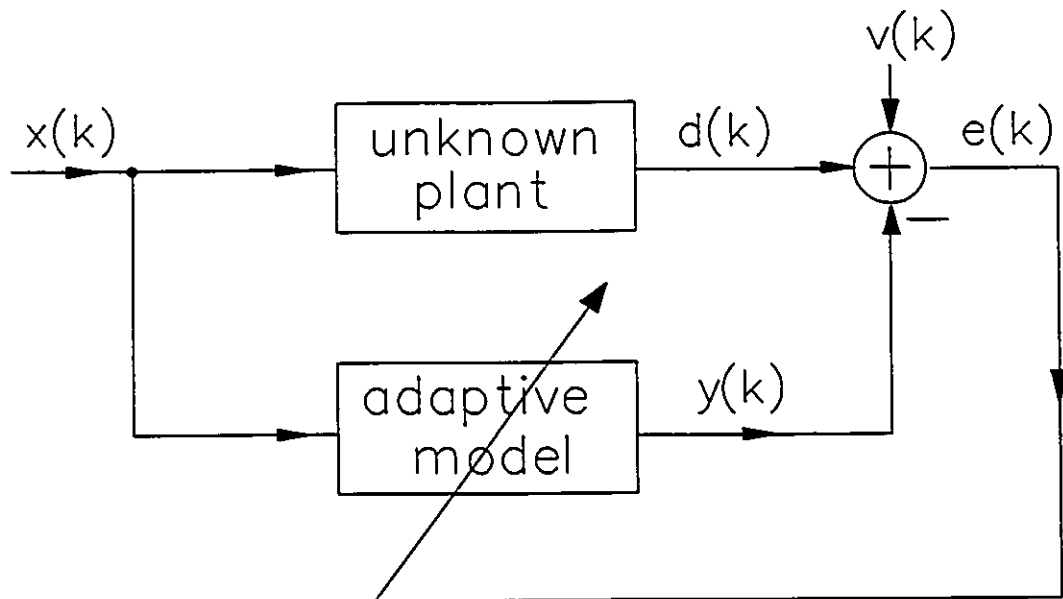


Fig. 2.1 Block diagram of adaptive system modelling

Eqn. 2.1 can be decomposed into the following format which is suitable for delayed N-path realization, i.e.,

$$H_k(z^{-1}) = \sum_{i=0}^{N-1} H_{i,k}(z^{-N})z^{-i} \quad (2.2)$$

where

$$H_{i,k}(z^{-N}) = \sum_{l=0}^{L_i} h_k(lN+i)z^{-lN} \quad (2.3)$$

and

$$L_i = [M/N]^+ \quad \text{for } i = 0, 1, \dots, J-1 \quad (2.4a)$$

$$L_i = [M/N]^+ - 1 \quad \text{for } i = J, J+1, \dots, N-1 \quad (2.4b)$$

$$J = (M)_N \quad (2.4c)$$

where in eqns. 2.4a and 2.4b, $[M/N]^+$ represents the least integer greater than or equal to M/N . $(M)_N$ in eqn. 2.4c means $M \bmod N$. The block diagram for such a delayed N -path realization is shown in Fig. 2.2.

Denoting the input signal, $x(k)$, and the output signal, $y(k)$, and their z -transforms as $X(z^{-1})$ and $Y(z^{-1})$ respectively, we shall have the following input-output relationships

$$\begin{aligned} Y(z^{-1}) &= H_k(z^{-1})X(z^{-1}) \\ &= \sum_{i=0}^{N-1} H_{i,k}(z^{-N})z^{-i}X(z^{-1}) \end{aligned} \quad (2.5)$$

$$= \sum_{i=0}^{N-1} Y_i(z^{-1})$$

where

$$Y_i(z^{-1}) = H_{i,k}(z^{-N})X(z^{-1})z^{-i} \quad (2.6)$$

is the z -transform of $y_i(k)$ which is the output of the i th path as shown in Fig. 2.2.

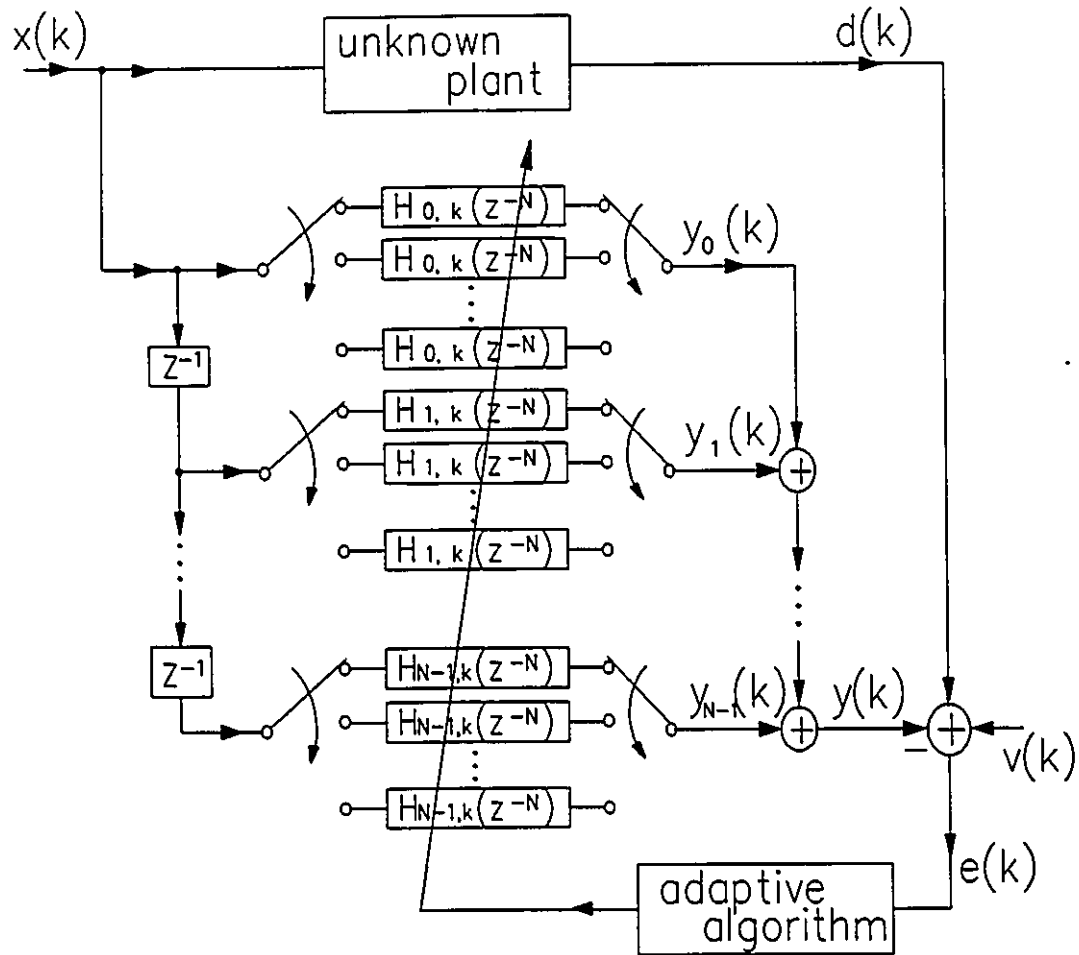


Fig. 2.2 DNA-FIR digital filter structure

Substituting eqn. 2.3 into eqn. 2.6 and taking the inverse z-transform, we obtain

$$y_i(k) = \sum_{l=0}^{L_i} h_k(lN+i)x(k-lN-i) \quad (2.7)$$

and taking the inverse z-transform of eqn. 2.5, we obtain

$$y(k) = \sum_{i=0}^{N-1} y_i(k) \quad (2.8)$$

Let $d(k)$ and $v(k)$ represent, respectively, the desired output and the measurement noise at a discrete time k , the instantaneous squared error is

$$e^2(k) = [d(k) + v(k) - y(k)]^2 \quad (2.9)$$

Define the change of a coefficient, $\Delta h_k(lN+i)$, as

$$\Delta h_k(lN+i) = -\mu \frac{\partial e^2(k)}{\partial h_k(lN+i)} \quad (2.10)$$

where μ is the convergence constant or step size parameter. From eqns. 2.9 - 2.10, we obtain

$$h_{k+N}(lN+i) = h_k(lN+i) + 2\mu e(k)\alpha(k, lN+i) \quad (2.11)$$

for $i = 0, 1, \dots, N-1; \quad l = 0, 1, \dots, L_i$

where

$$\alpha(k, lN+i) = \frac{\partial y(k)}{\partial h_k(lN+i)} \quad (2.12)$$

From eqns. 2.12, 2.8, and 2.7, we obtain

$$\alpha(k, lN+i) = x(k - lN - i) \quad (2.13)$$

As we can see from Fig. 2.2, the proposed DNA-FIR digital filter structure consists of N paths. In actual realization, each path can be realized using N DSPs such that each DSP is used to realize a variable coefficient sub-filter and its adaptive algorithm. The sampling rate of each DSP can be N times slower than that of the direct form adaptive FIR digital filter, whereas the number of coefficients to be updated in each DSP is equal to or less than $1/N$ that of the number of coefficients in the direct form adaptive FIR digital filter realized using one DSP. Using an identical sampling rate (where the maximum speed limit is imposed by the currently available DSPs) in all the N^2 DSPs of a DNA-FIR digital filter and in the only one DSP of a direct form adaptive FIR digital filter, the overall maximum throughput rate of the DNA-FIR digital filter structure can be increased up to N^2 times.

2.1.3 Simulation Results

Three plants $A_1(z^{-1})$, $A_2(z^{-1})$, and $A_3(z^{-1})$, of orders 24, 48, and 96 respectively, are to be modelled by the DNA-FIR filter structures with 1-path (equivalent to the direct form structure), 2-path, 5-path, and 8-path realizations. The values of plant coefficients are randomly chosen to be within ± 1 . In all simulations, the initial coefficient values were set to 0.0. The input signal $x(k)$ was a white random sequence with a mean value of 0 and a variance of $1/12$.

Table 2.1 summarizes the first group of simulation results without a measurement noise in terms of the number of iterations needed to converge to a specified stopping criterion. The convergence constant, μ , used in this table was 1.0×10^{-2} and the stopping criterion was set to 1.0×10^{-6} which measured the average squared output error over 24, 48, and 96 samples for $A_1(z^{-1})$, $A_2(z^{-1})$, and $A_3(z^{-1})$, respectively.

Table 2.1: Convergence performance in terms of number of iterations of DNA-FIR structures without measurement noise.

Plant	Stopping Criterion	1-path	2-path	5-path	8-path
$A_1(z^{-1})$	1.0×10^{-6}	7920	15552	38320	60240
$A_2(z^{-1})$	1.0×10^{-6}	8304	17040	42210	65328
$A_3(z^{-1})$	1.0×10^{-6}	9216	18336	46170	72288

Table 2.2 summarizes the second group of simulation results with an added measurement noise $v(k)$. $v(k)$ is a white random sequence with 0 mean value and $1/48$ variance value which is independent of $x(k)$. The convergence constant, μ , was also 1.0×10^{-2} . The stopping criteria were set to 1.5×10^{-2} , 1.5×10^{-2} , and 2.0×10^{-2} for $A_1(z^{-1})$, $A_2(z^{-1})$, and $A_3(z^{-1})$, which measured the average squared output error over 24, 48, and 96 samples, respectively.

For illustration, Figs. 2.3-2.5 show, respectively, the curves of average squared

output error of $e(k)$ versus the number of iterations k for the delayed 2-path modelling of $A_1(z^{-1})$, the delayed 5-path modelling of $A_2(z^{-1})$ and the delayed 8-path modelling of $A_3(z^{-1})$ of Table 2.2. The final output error residuals were due to the input measurement noise. Other average squared output error curves are similar and are not shown here.

Table 2.2: Convergence performance in terms of number of iterations of DNA-FIR structures with measurement noise.

Plant	Stopping Criterion	1-path	2-path	5-path	8-path
$A_1(z^{-1})$	1.5×10^{-2}	3864	7368	12780	23760
$A_2(z^{-1})$	1.5×10^{-2}	4128	9216	16965	36768
$A_3(z^{-1})$	2.0×10^{-2}	6336	9216	22705	34656

It can be seen from Tables 2.1 and 2.2 that the number of iterations needed for the DNA-FIR structure were more than those for the direct form structure (i.e., with 1-path). Under the worst situation, the number of iterations of a DNA-FIR structure could be around N times that of the direct form structure. However, as N^2 times maximum throughput rate improvement can be achieved using the DNA-FIR structure, the convergence time needed by the DNA-FIR structure can be at least N times less than that required by the direct form structure. In chapter 3, we will have detailed discussions on how the adaptive algorithm would affect the processing speed of this type of adaptive digital filters.

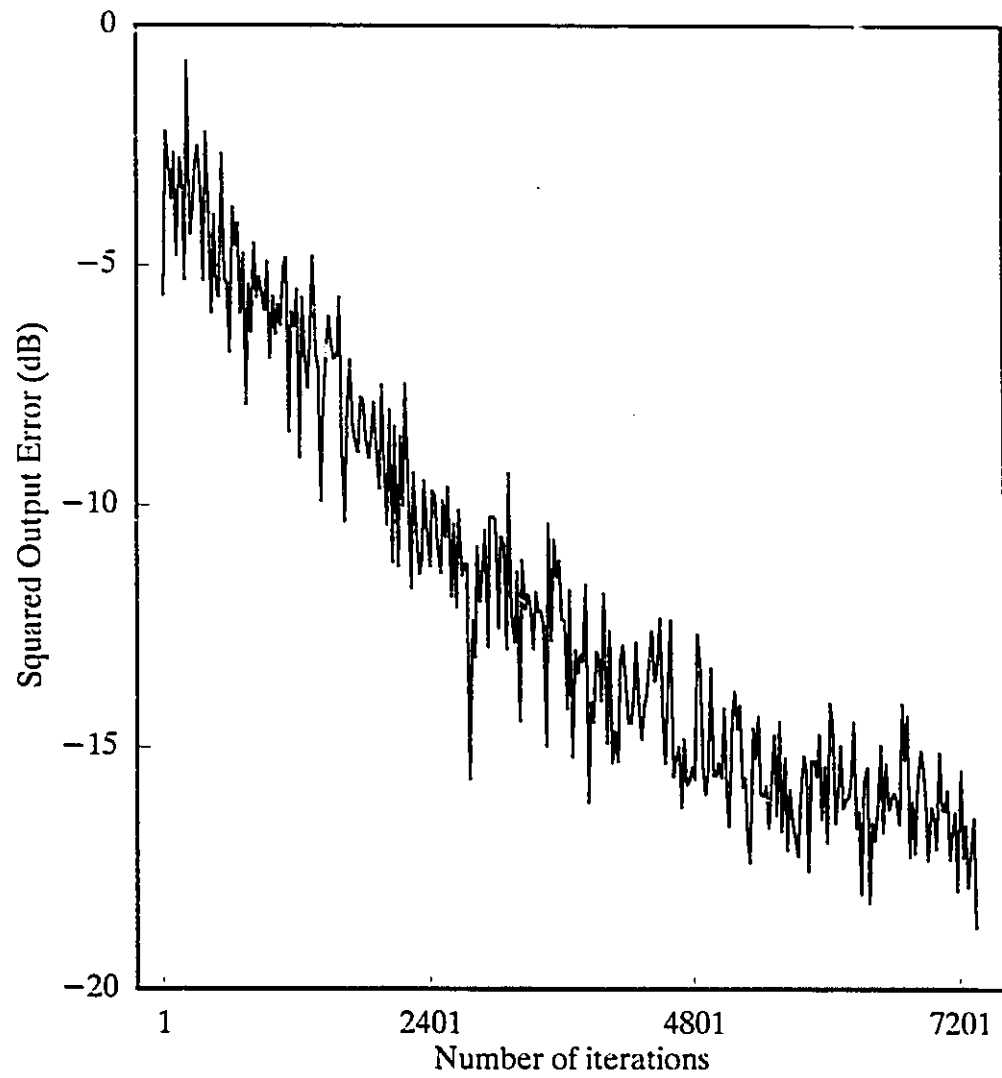


Fig. 2.3 Average squared output error of D2A-FIR modelling of $A_1(z^{-1})$

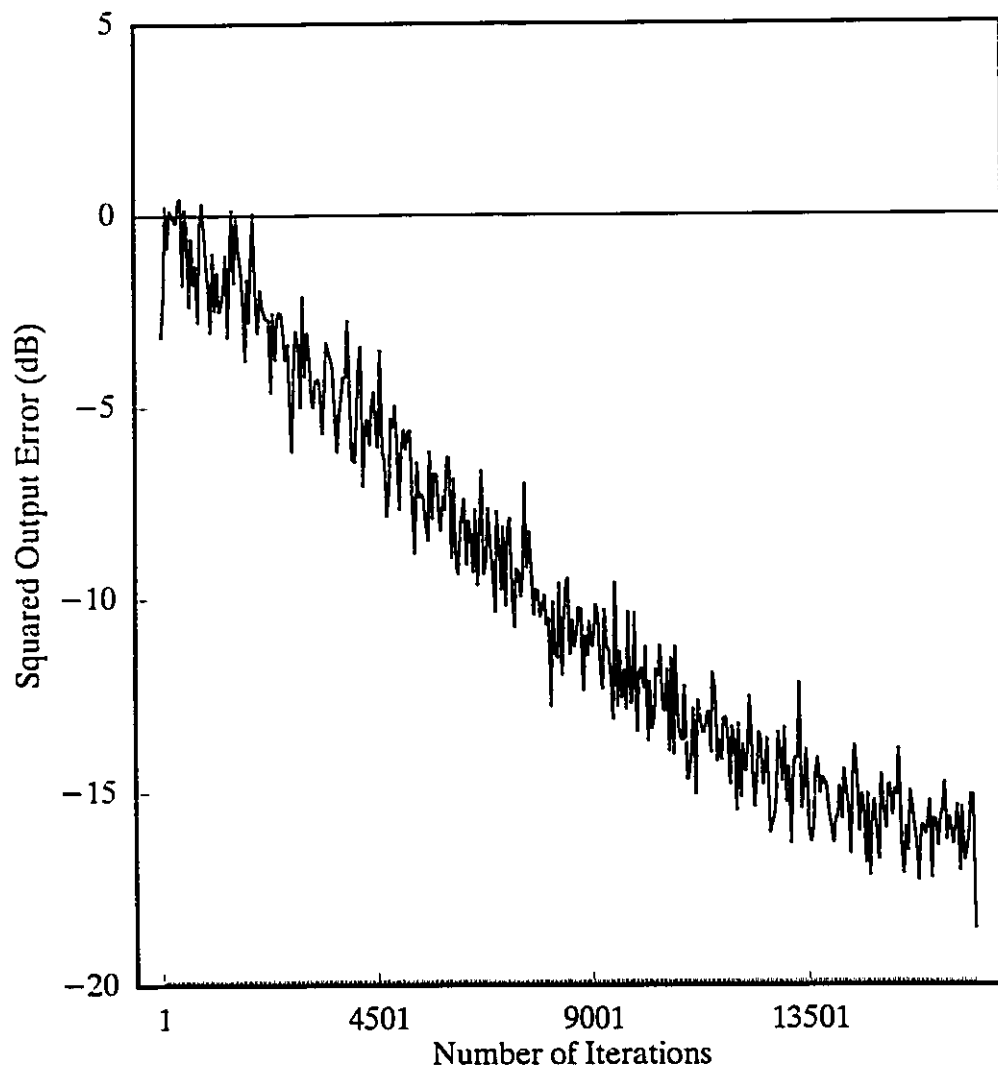


Fig. 2.4 Average squared output error of D5A-FIR modelling of $A_2(z^{-1})$

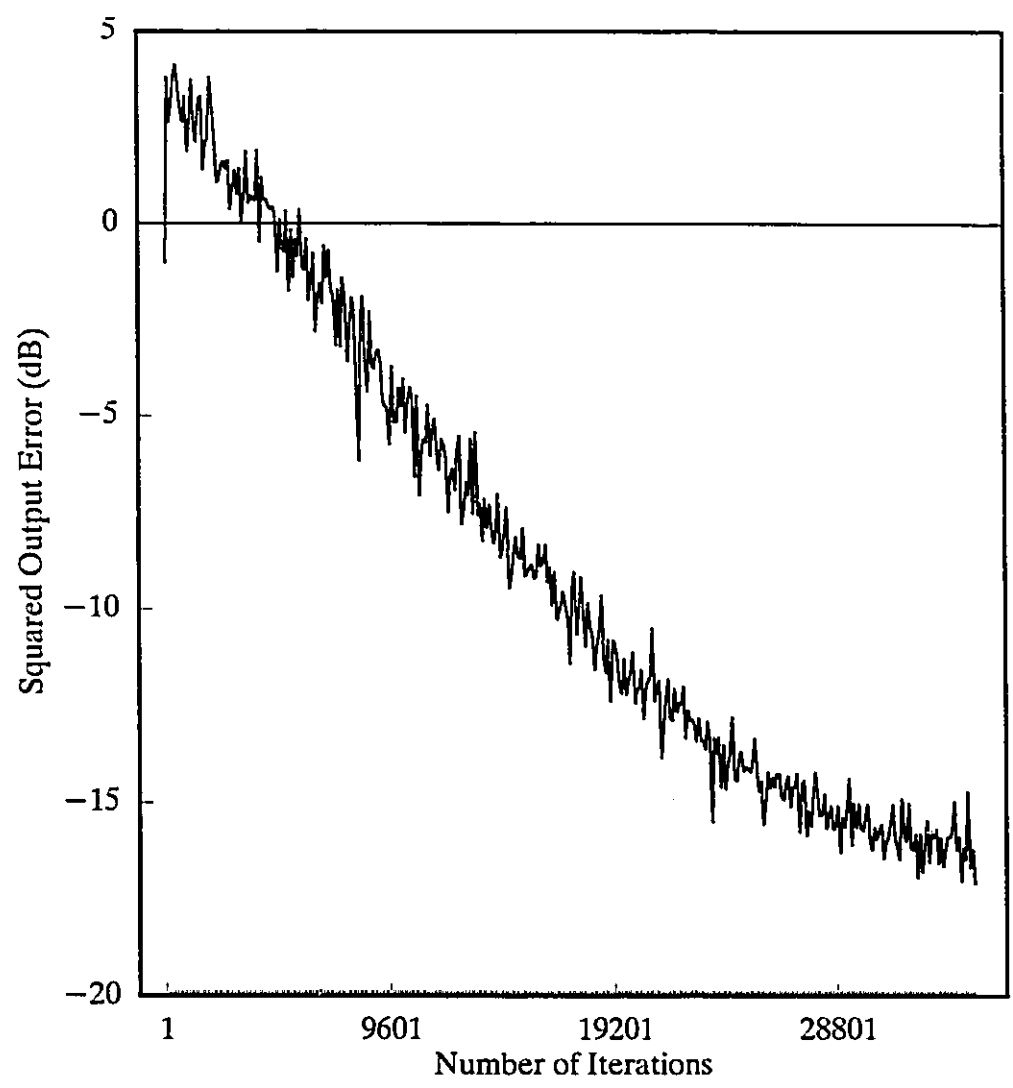


Fig. 2.5 Average squared output error of D8A-FIR modelling of $A_3(z^{-1})$

2.1.4 Conclusions

A DNA-FIR digital filter structure and its adaptive algorithm have been developed which can be realized by N^2 DSPs, and is applicable to any N-path ($N \geq 2$) realization. The maximum throughput rate can be N^2 times that of the direct form FIR adaptive digital filter realized using one DSP. Simulation results have shown that this type of adaptive digital filters performs very well in the application to adaptive system modelling.

2.2 Delayed N-Path Adaptive Linear Phase FIR (DNALP-FIR) Digital Filter

2.2.1 Introduction

While adaptive digital filtering is finding applications in such diverse fields as communications, radar, sonar, seismology, and biomedical engineering, adaptive digital filtering with linear phase characteristics constitutes an important class of filters in adaptive digital signal processing. It is because linear phase filtering causes no harmonic distortion to an input signal in circumstances where frequency dispersion due to phase nonlinearity is undesirable. For examples:

- A digital communication signal passing through an adaptive FIR equalization filter with nonlinear phase characteristic will suffer an increased intersymbol interference level and degraded detectability. With linear phase filtering, a spectral estimation technique has been developed for high resolution harmonic

analysis, when a process under measurement contains an unknown number of sinusoids of unknown frequencies and amplitudes [39].

- In spread-spectrum communications, linear phase adaptive filtering is applied for the suppression of a strong interfering signal due to jamming, to other users in the channel, or to multipath propagation, as well as for achieving low probability of intercept communications. It also makes privacy message transmission possible [40].
- The possibility of exactly linear phase processing is often useful in data transmission and speech processing applications where precise time alignment is essential [41].
- Adaptive linear phase digital filtering has also been used extensively in adaptive line enhancement, adaptive noise cancellation, linear phase modelling and system identification [42-43].
- Some other typical applications of practical interest include design of one dimensional and multidimensional frequency selective filters for speech processing, geophysical and seismic processing, image analysis, and adaptive arrays in radar systems. In some circumstances, the linear phase property is intrinsic to the problem itself as in maximum entropy spectral estimation [44]. In [45-47], a number of algorithms were introduced for efficient computation of adaptive linear phase FIR digital filters.

Based on the idea of the delayed 2-path linear phase FIR structure advanced in [18, 48-49], a high speed delayed N-path adaptive linear phase FIR (DNALP-FIR) digital filtering structure is proposed in the following. By applying the symmetry property of the filter coefficients to the realization of delayed N-path scheme, the maximum throughput rate of the present system using N^2 digital signal processors (DSPs) can be increased by a factor of $2N^2$ compared to that of a direct realization using one digital signal processor (DSP).

2.2.2 Structure of Delayed N-Path Linear Phase FIR Digital Filter

This section presents the realization of DNALP-FIR digital filters which exploits the symmetry property of the filter coefficients. Consider the transfer function of a FIR adaptive digital filter at a discrete time k as

$$H_k(z^{-1}) = \sum_{m=0}^{M-1} h_k(m)z^{-m} \quad (2.14)$$

Eqn. 2.14 is called a linear phase transfer function if

$$\begin{aligned} h_k(m) &= \pm h_k(M-1-m) \\ &\text{for } m = 0, 1, \dots, (M/2)-1 \quad \text{when } M \text{ even} \\ \text{or for } m &= 0, 1, \dots, (M-1)/2 \quad \text{when } M \text{ odd} \end{aligned} \quad (2.15)$$

In eqn. 2.15, the positive sign corresponds to the case of even symmetry and the negative sign corresponds to the case of odd symmetry in the impulse responses. In the following, the realization of the delayed N-path structure will be presented for both even M and odd M cases followed by an illustrative example.

2.2.2.1 Even M Delayed N-Path Linear Phase FIR Digital Filter Structure

By using the symmetry property, eqn. 2.14 can be decomposed into the following form which is suitable for delayed N-path realization, i.e.,

$$H_k(z^{-1}) = \sum_{i=0}^{N-1} H_{i,k}(z^{-N})_{EK} z^{-i} \quad (2.16)$$

where

$$H_{i,k}(z^{-N})_{EK} = \sum_{l=0}^{K_i-1} h_k(lN+i)_{EK} z^{-lN} [1 \pm z^{-(M-1-2(lN+i))}] \quad (2.17)$$

The subscript E represents even M and the subscript K can be E for even symmetry and O for odd symmetry. For a particular i, $i = 0, 1, \dots, N-1$, K_i is the largest integer satisfying

$$(k_i - 1)N + i \leq M/2 - 1 \quad (2.18)$$

and

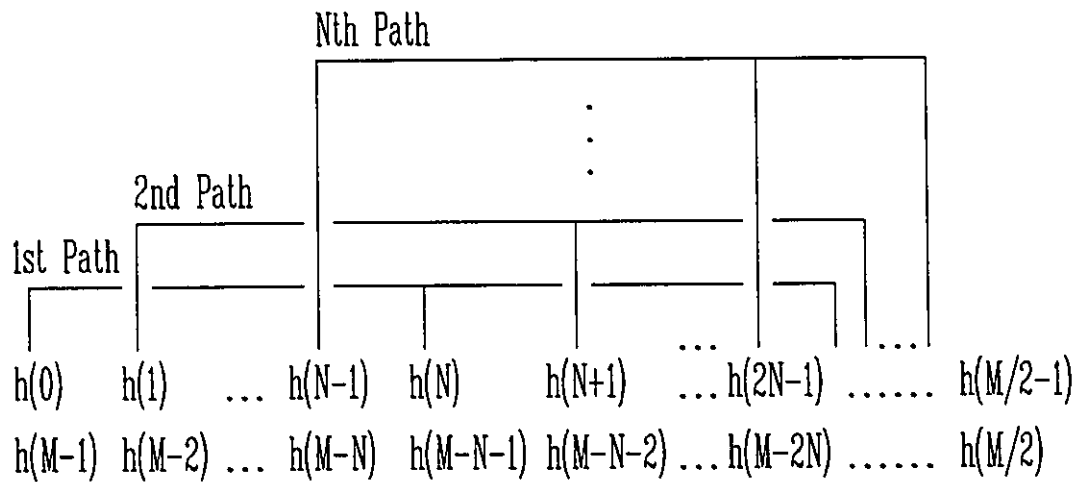
$$h_k(m)_{EK} = h_k(m) \text{ for } m = 0, 1, \dots, M/2-1 \quad (2.19)$$

Actually, for even M, the coefficients of the transfer function can be arranged as in Table 2.3 (the subscript k has been omitted for convenience).

Because of the symmetry property of the filter coefficients, every two coefficients in the same column are equal to each other for even symmetry or have the same absolute value but opposite signs for odd symmetry. Hence, only the coefficients in the first row are of interest to the realization of the delayed N-path scheme. Starting from $h(0)$, the

coefficients in every N columns are selected to form $H_{0,k}(z^N)_{EK}$ in eqn. 2.17 for the 1st path as shown in Fig. 2.6. The 2nd path, 3rd path, ..., and N th path can be realized in a similar manner by starting from $h(1)$, $h(2)$, ..., and $h(N-1)$ respectively as indicated in Table 2.3. Each path carries its own shifted delays based on eqn. 2.17 and does not necessarily have the same number of coefficients since K_i in eqn. 2.17 may vary by one for $i = 0, 1, \dots, N-1$.

Table 2.3: Coefficient configuration for the even M case and N -path realization



2.2.2.2 Odd M Delayed N-Path Linear Phase FIR Digital Filter Structure

For odd M, eqn. 2.14 can be decomposed as

$$H_k(z^{-1}) = \sum_{i=1}^{N-1} H_{i,k}(z^{-N})_{OK} z^{-i} \quad (2.20)$$

where

$$H_{i,k}(z^{-N})_{OK} = \sum_{l=0}^{K_i-1} h_k(lN+i)_{OK} z^{-iN} [1 \pm z^{-(M-1-2(lN+i))}] \quad (2.21)$$

The subscript O represents odd M and the subscript K can be E for even symmetry and O for odd symmetry. For a particular i, $i = 0, 1, \dots, N-1$, K_i is the largest integer satisfying

$$(K_i - 1)N + i \leq (M-1)/2 \quad (2.22)$$

For even symmetry, we have

$$h_k(m)_{OE} = h_k(m) \quad \text{for } m = 0, 1, \dots, (M-3)/2 \quad (2.23a)$$

and

$$h_k((M-1)/2)_{OE} = h_k((M-1)/2) \quad (2.23b)$$

and for odd symmetry we have

$$h_k(m)_{OO} = h_k(m) \quad \text{for } m = 0, 1, \dots, (M-1)/2 \quad (2.23c)$$

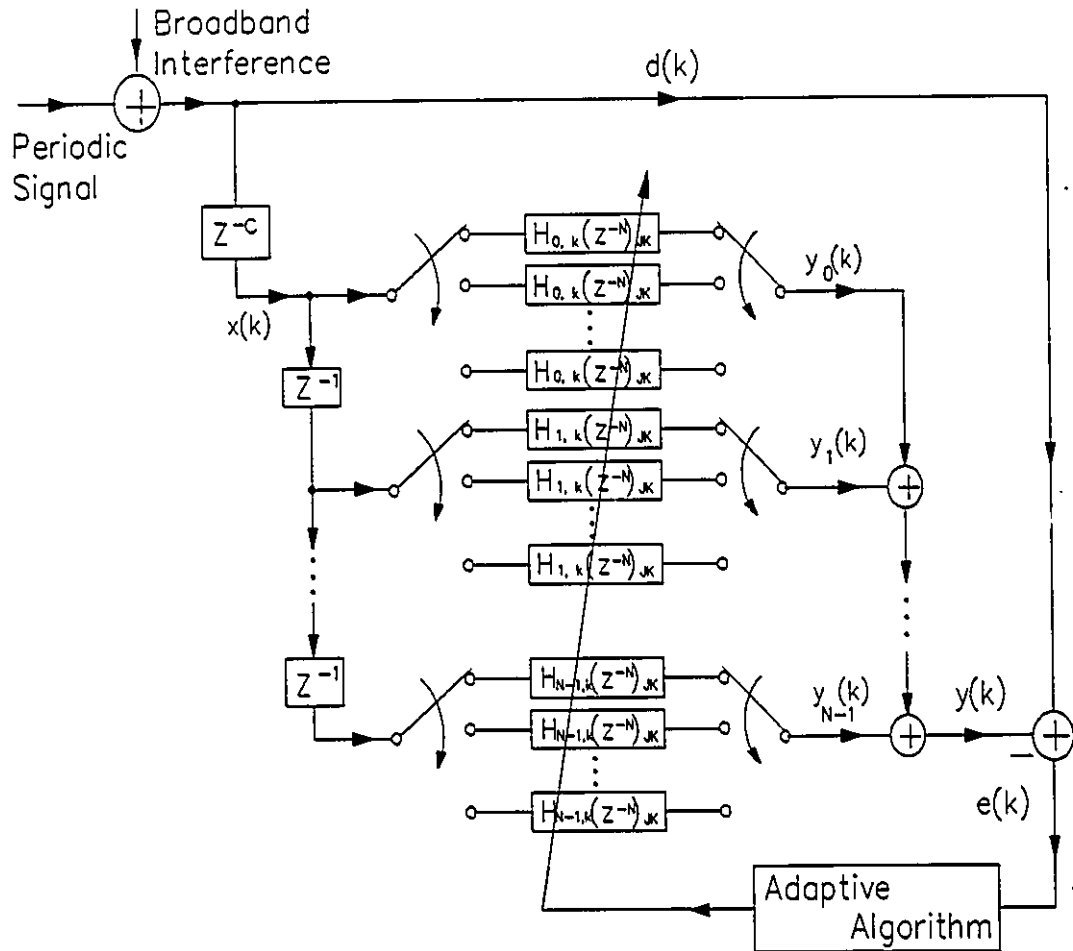
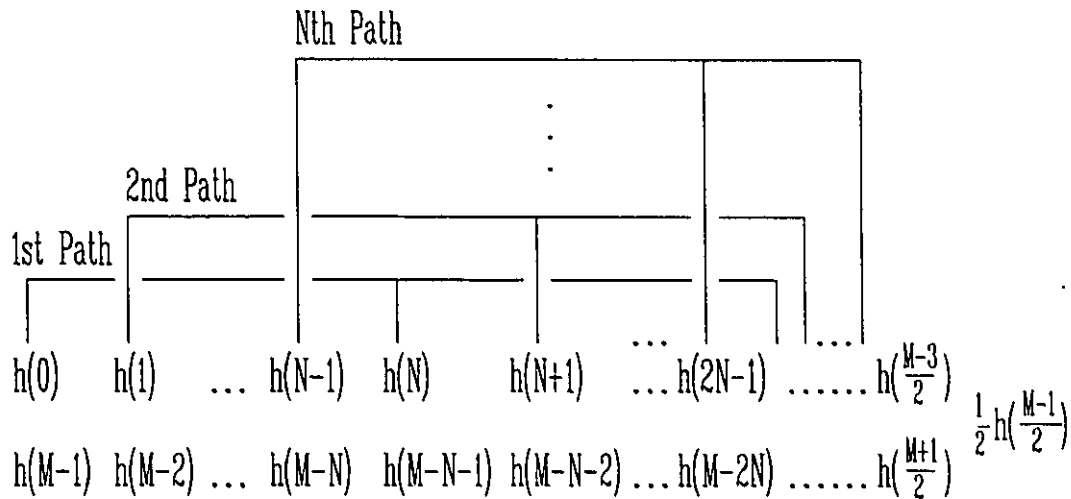


Fig. 2.6 Block diagram of DNALP-FIR digital filter for adaptive noise cancellation (where J can be E for even M and O for odd M; K can be E for even symmetry and O for odd symmetry)

The coefficients of the transfer function can be arranged as in Table 2.4 (the subscript k has been dropped off again for convenience).

Table 2.4: Coefficient configuration for the odd M case and N -path realization



Because of the symmetry property, every two coefficients in the same column are equal to each other for even symmetry or have the same absolute value but opposite signs for odd symmetry ($h((M-1)/2)=0$ can be expected for odd symmetry). Hence, the first row coefficients can be used to contrive the delayed N -path structure. In a similar manner as the even M case, $H_{i,k}(z^N)_{OK}$ in eqn. 2.21, for $i = 0, 1, \dots, N-1$, can be formed by selecting coefficients in every N columns started from $h(0)$, and then $h(1)$, ..., and $h(N-1)$ respectively. Each path has its shifted delays as expressed in eqn. 2.21. The number of coefficients of different paths may vary by one.

2.2.2.3 Example

For an example, a 19th-order ($M=20$) even symmetry linear phase FIR transfer function is to be decomposed into a 3-path structure, i.e., $N=3$. According to Table 2.3, the coefficients can be arranged as shown in Table 2.5 (the subscript k has been omitted again for convenience).

Hence, we obtain the following equations according to eqn. 2.17:

$$H_{0,k}(z^{-3})_{EE} = h_k(0)[1 + z^{-19}] + h_k(3)[1 + z^{-13}]z^{-3} \\ + h_k(6)[1 + z^{-7}]z^{-6} + h_k(9)[1 + z^{-1}]z^{-9}$$

$$H_{1,k}(z^{-3})_{EE} = h_k(1)[1 + z^{-17}] + h_k(4)[1 + z^{-11}]z^{-3} \\ + h_k(7)[1 + z^{-5}]z^{-6}$$

$$H_{2,k}(z^{-3})_{EE} = h_k(2)[1 + z^{-15}] + h_k(5)[1 + z^{-9}]z^{-3} \\ + h_k(8)[1 + z^{-3}]z^{-6}$$

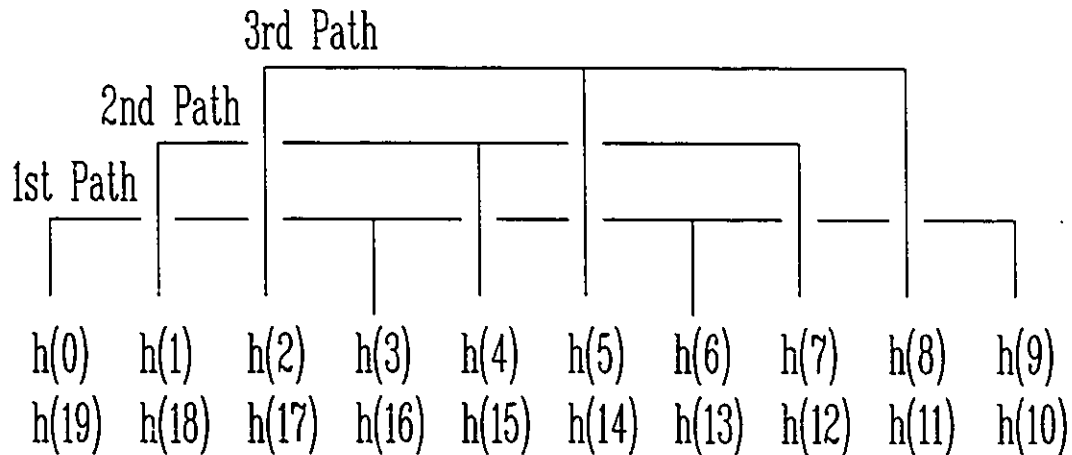
and

$$H_k(z^{-1}) = H_{0,k}(z^{-3})_{EE} + z^{-1}H_{1,k}(z^{-3})_{EE} + z^{-2}H_{2,k}(z^{-3})_{EE}$$

This resultant delayed 3-path linear phase FIR digital filter structure is shown in Fig.

2.7.

Table 2.5: Coefficient configuration for the example of 19th-order and 3-path realization



As shown in Fig. 2.6, the proposed structure consists of N paths for an N -path configuration. In actual realization, each path can be realized using N DSPs. In the N -path case, the sampling rate of the signal processing through each DSP is N times slower than that of a conventional adaptive structure. Due to the symmetry property of the coefficients, the number of coefficients each DSP processes is $1/(2N)$ times (or less than) that of the number of coefficients in a conventional adaptive filter structure realized using one DSP. Hence, the overall maximum throughput rate of the structure can be improved by $2N^2$ times, which has been further increased by a factor of 2 compared to the case without using the property of coefficient symmetry.

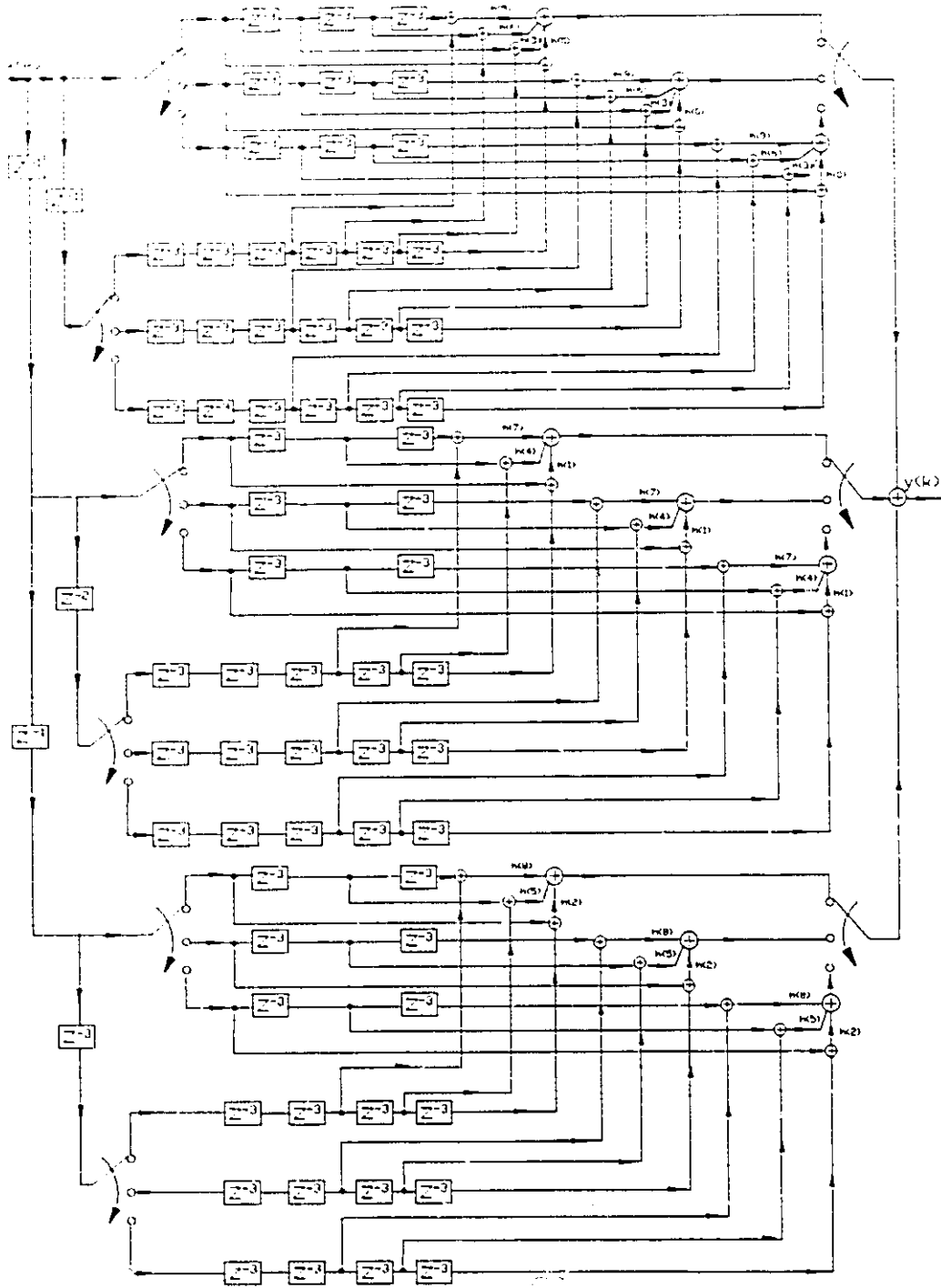


Fig. 2.7 Structure of a 19th-order delayed 3-path even symmetry linear-phase FIR digital filter

2.2.3 Adaptive Algorithm of DNALP-FIR Digital Filter

2.2.3.1 Even M DNALP-FIR Digital Filter

As shown in Fig. 2.6, denote the input signal, $x(k)$, the output signal, $y(k)$, and their z-transforms as $X(z^{-1})$ and $Y(z^{-1})$ respectively, we have the following input-output relationships

$$\begin{aligned}
 Y(z^{-1}) &= H_k(z^{-1})X(z^{-1}) \\
 &= \sum_{i=0}^{N-1} H_{i,k}(z^{-N})_{EK} z^{-i} X(z^{-1}) \\
 &= \sum_{i=0}^{N-1} Y_i(z^{-1})
 \end{aligned} \tag{2.24}$$

where

$$Y_i(z^{-1}) = H_{i,k}(z^{-N})_{EK} z^{-i} X(z^{-1}) \tag{2.25}$$

is the z-transform of $y_i(k)$ which is the output of the i th path as shown in Fig. 2.6.

Substituting eqn. 2.17 into eqn. 2.25 and taking the inverse z-transform, we obtain

$$y_i(k) = \sum_{l=0}^{K_i-1} h_k(lN+i)_{EK} [x(k-lN-i) \pm x(k-M+1+lN+i)] \tag{2.26}$$

and taking the inverse z-transform of eqn. 2.24, we have

$$y(k) = \sum_{i=0}^{N-1} y_i(k) \tag{2.27}$$

Let $d(k)$ represent the desired output at discrete time k , the instantaneous squared error

is

$$e^2(k) = [d(k) - y(k)]^2 \quad (2.28)$$

Using the LMS method, we define

$$\begin{aligned} h_{k+N}(lN+i)_{EK} &= h_k(lN+i)_{EK} + 2\mu e(k)\alpha(k, lN+i) \\ \text{for } i &= 0, 1, \dots, N-1; \quad l = 0, 1, \dots, K_i-1 \end{aligned} \quad (2.29)$$

where μ is the convergence constant, and

$$\alpha(k, lN+i) = \frac{\partial y(k)}{\partial h_k(lN+i)_{EK}} \quad (2.30)$$

Differentiating eqn. 2.26 and 2.27 with respect to $h_k(lN+i)_{EK}$, we have

$$\alpha(k, lN+i) = x(k-lN-i) \pm x(k-M+1+lN+i) \quad (2.31)$$

2.2.3.2 Odd M DNALP-FIR Digital Filter

As shown in Fig. 2.6, we obtain

$$\begin{aligned} Y(z^{-1}) &= H_k(z^{-1})X(z^{-1}) \\ &= \sum_{i=0}^{N-1} H_{i,k}(z^{-M})_{OK} z^{-i} X(z^{-1}) \\ &= \sum_{i=0}^{N-1} Y_i(z^{-1}) \end{aligned} \quad (2.32)$$

where

$$Y_i(z^{-1}) = H_{i,k}(z^{-N})_{OK} z^{-i} X(z^{-1}) \quad (2.33)$$

Substituting eqn. 2.21 into eqn. 2.33 and taking the inverse z transform, we obtain

$$y_i(k) = \sum_{l=0}^{K_i-1} h_k(lN+i)_{OK} [x(k-lN-i) \pm x(k-M+1+lN+i)] \quad (2.34)$$

Taking the inverse z-transform of eqn. 2.32, we have

$$y(k) = \sum_{i=0}^{N-1} y_i(k) \quad (2.35)$$

Define the instantaneous squared error as

$$e^2(k) = [d(k) - y(k)]^2 \quad (2.36)$$

Based on the LMS method, we have

$$\begin{aligned} h_{k+N}(lN+i)_{OK} &= h_k(lN+i)_{OK} + 2\mu e(k)\alpha(k, lN+i) \\ \text{for } i &= 0, 1, \dots, N-1; \quad l = 0, 1, \dots, K_i-1 \end{aligned} \quad (2.37)$$

where μ is the convergence constant, and

$$\alpha(k, lN+i) = \frac{\partial y(k)}{\partial h_k(lN+i)_{OK}} \quad (2.38)$$

Hence from eqn. 2.34 and 2.35, eqn. 2.38 becomes

$$\alpha(k, lN+i) = x(k-lN-i) \pm x(k-M+1+lN+i) \quad (2.39)$$

We can see from the above derivations for both the even M and odd M cases that the updates of the filter coefficients are divided into N groups which are to be performed

by N different DSPs located in the N paths. Each sub-filter in a path has $1/(2N)$ times (or less than) coefficients to be updated as compared to that of the total coefficients of the filter. Actually, each DSP is expected to realize both the variable coefficient filter and the adaptive algorithm. Hence, the maximum throughput rate of the overall adaptive digital filtering system can be increased to $2N^2$ times compared to that of a conventional direct realization using one DSP.

2.2.4 Simulation Results

2.2.4.1 Application to Adaptive Noise Cancellation

In this section, simulation results are given on the use of the proposed filter structure for adaptive noise cancellation [7]. In many situations, a periodic signal of interest is corrupted by a broadband signal or white noise, while the reference signal is not available. In this circumstance, the block diagram of Fig. 2.6 can be employed which is capable of extracting a periodic signal from broadband noise. The periodic signal and broadband interference constitute the desired signal $d(k)$ which is also the primary input to the canceller. A fixed delay c is inserted in a reference input drawn directly from the primary input, the output of which shall form the input signal $x(k)$ to the adaptive filter. The output of the adaptive filter $y(k)$ is expected to be a close replica of the input periodic signal after convergence.

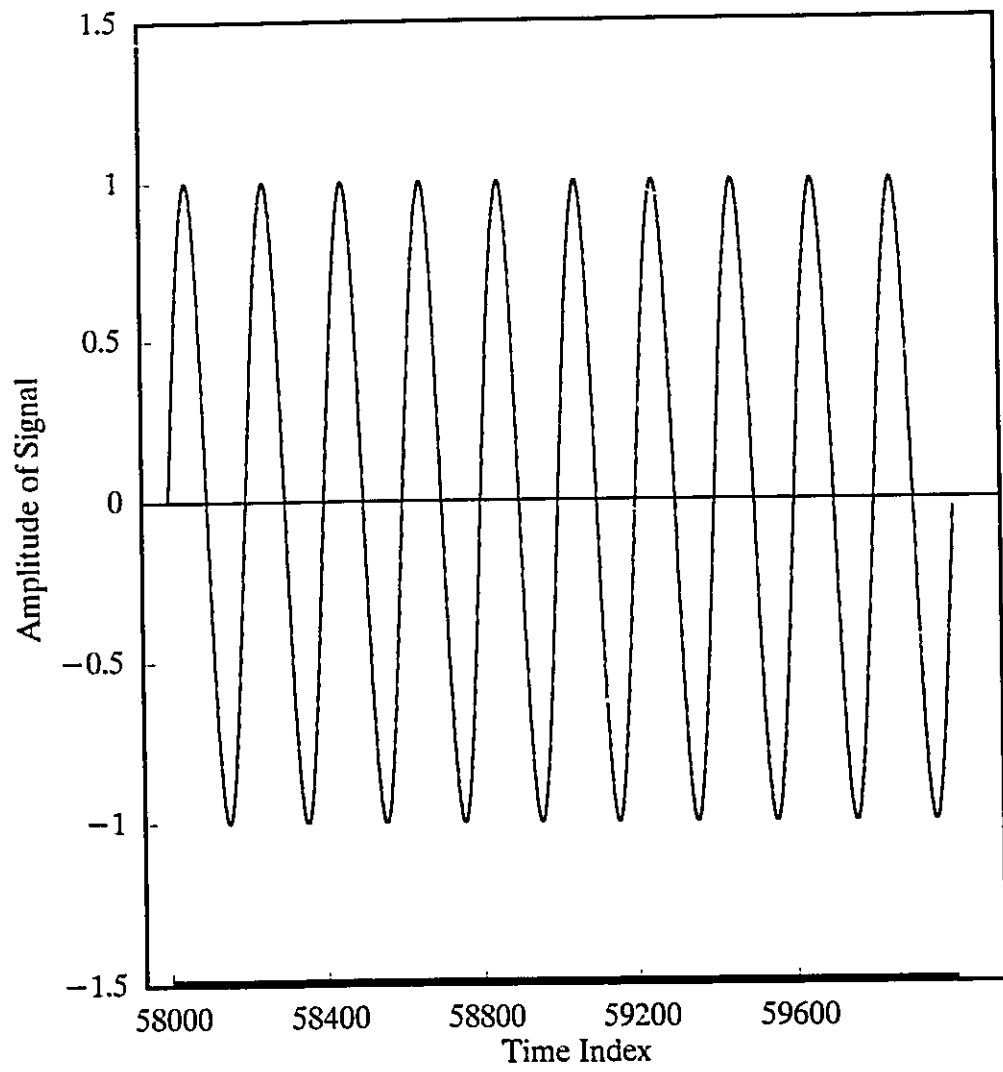
Example 1: Fig. 2.8a shows the original sinusoidal signal and Fig. 2.8b shows the corrupted input signal which is the sinusoidal signal in Fig. 2.8a contaminated by

10% white noise; Fig. 2.8c shows the filtered sinusoidal signal; and Fig. 2.8d shows the squared output error curve.

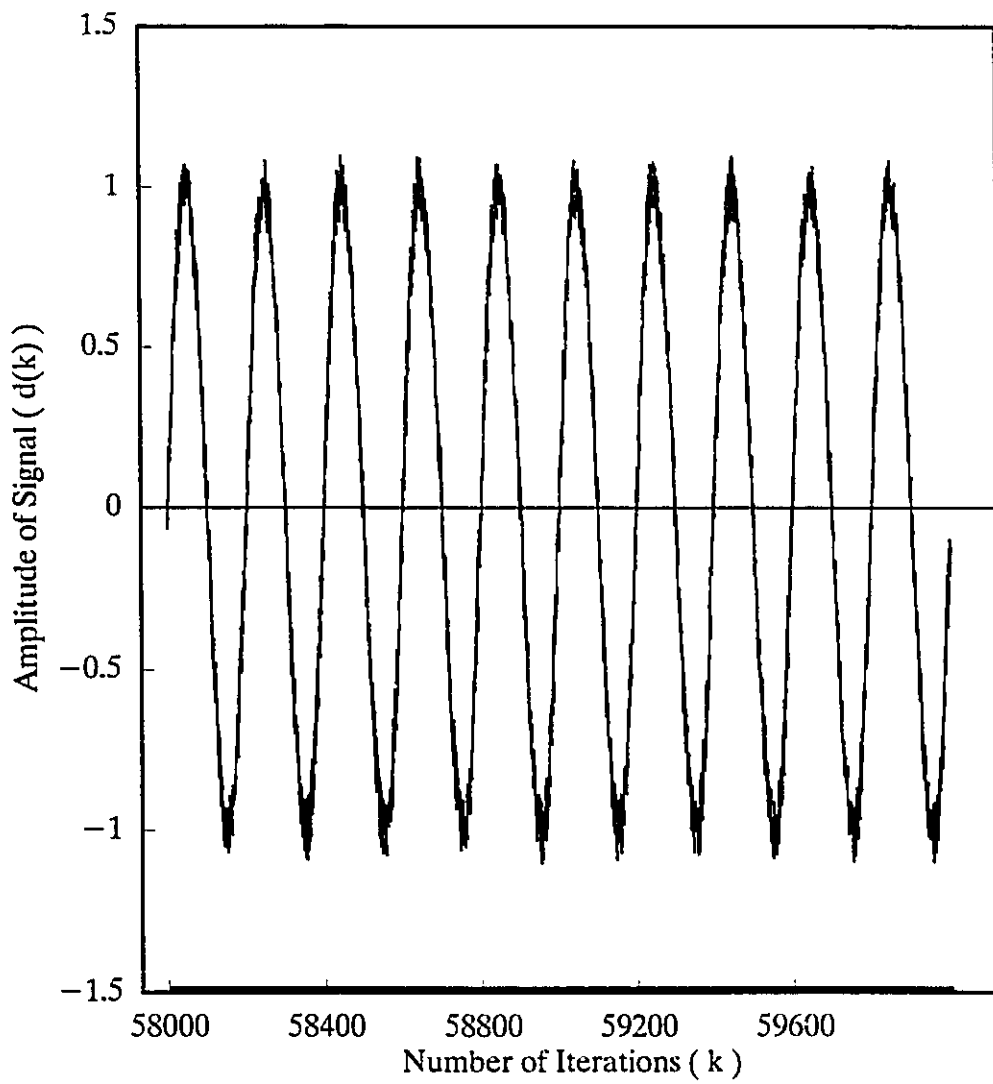
In the simulations, an 11th-order filter with a delayed 4-path even symmetry structure was used as the self-tuning adaptive noise canceller. The delay was set to 4 and the convergence constant was 1.0×10^{-5} . The initial filter coefficients were set to 0.0. After 60000 iterations, the average squared output error over 2000 iterations fell below 1.435×10^{-5} .

Example 2: Fig. 2.9 shows another example based on a sawtooth signal. Fig. 2.9a is the original sawtooth signal and Fig. 2.9b shows the input sawtooth signal corrupted by 10% white noise; Fig. 2.9c shows the recovered sawtooth signal; and Fig. 2.9d shows the squared output error curve.

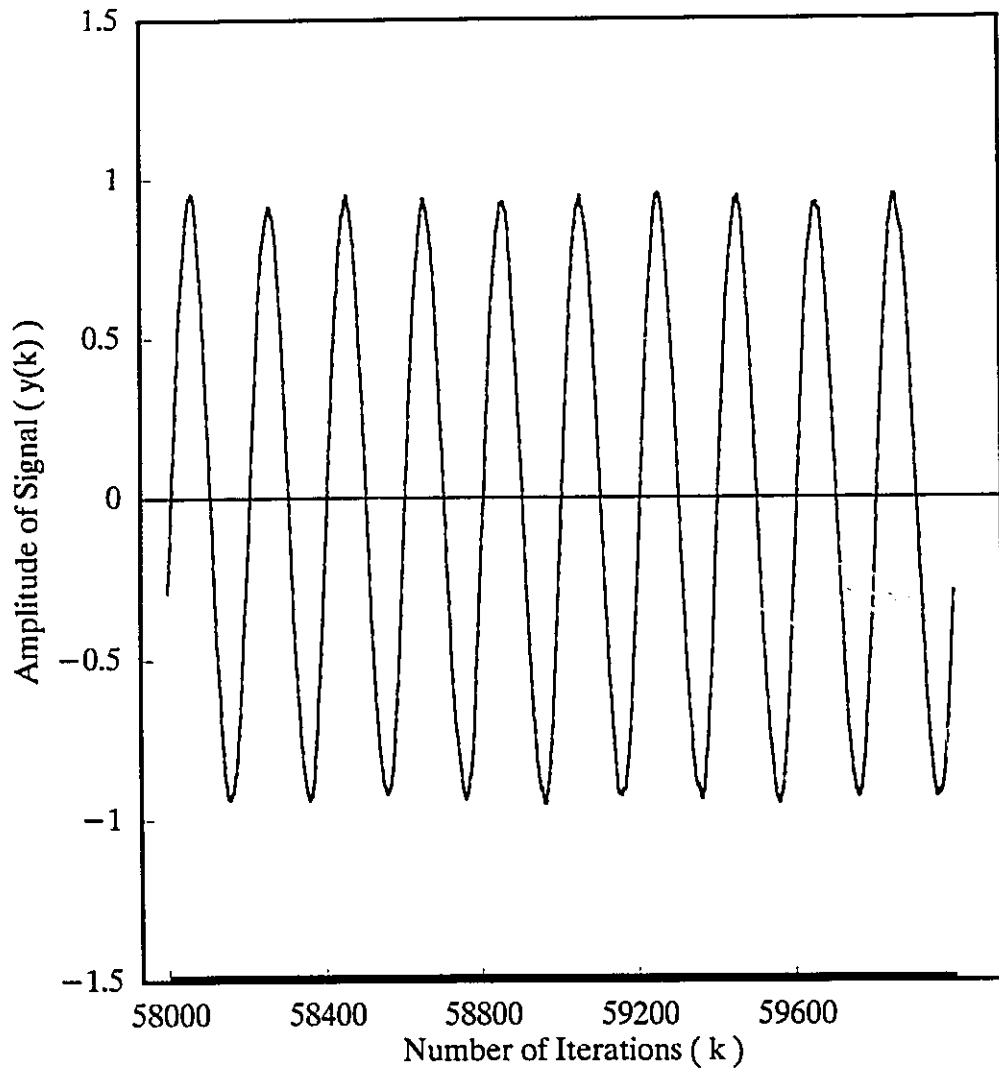
An 11th-order filter with a delayed 2-path even symmetry structure was used to perform adaptive noise cancellation. The delay was set to 4 and the convergence constant was 1.0×10^{-5} . The initial filter coefficients were set to 0.0. After 90000 iterations, the average squared output errors over 2000 samples fell below 3.08×10^{-5} .



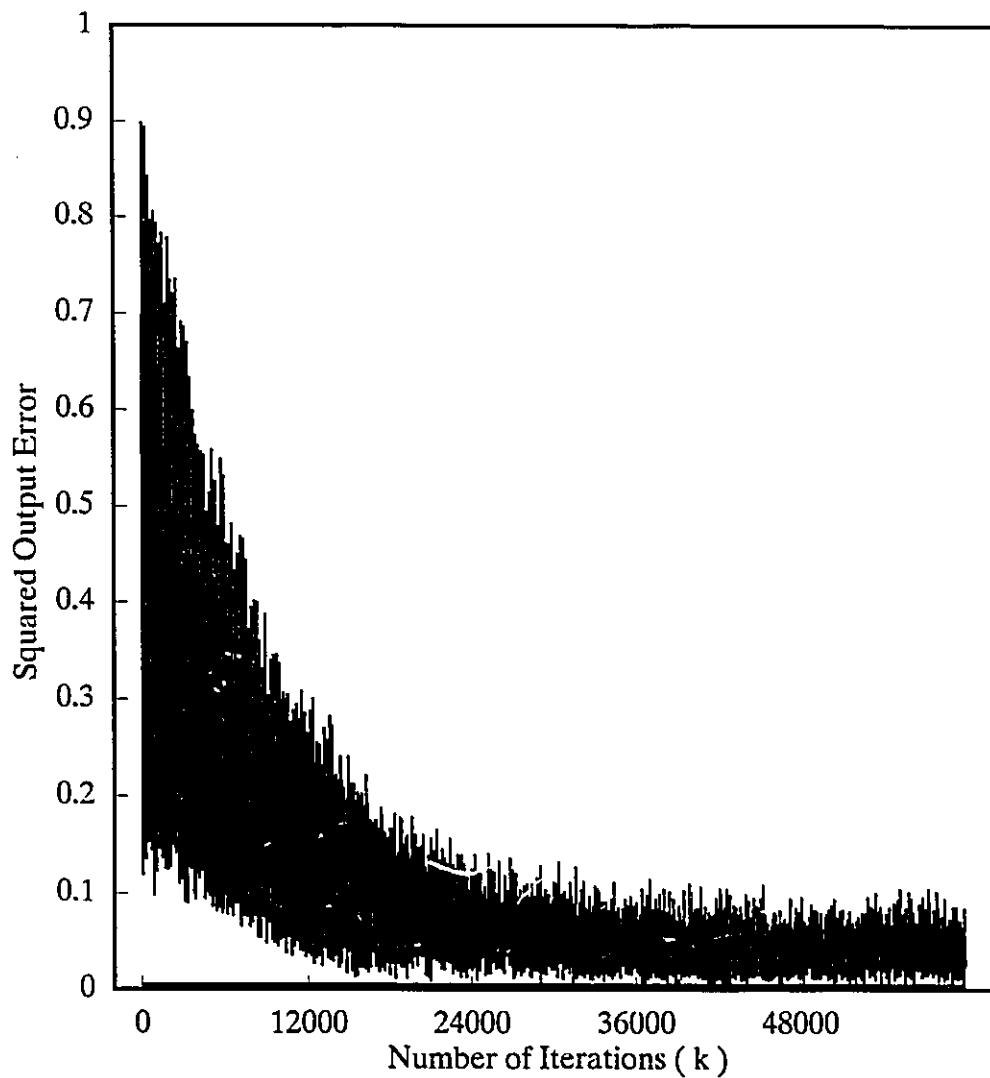
(a) Original sinusoidal signal



(b) Corrupted input signal

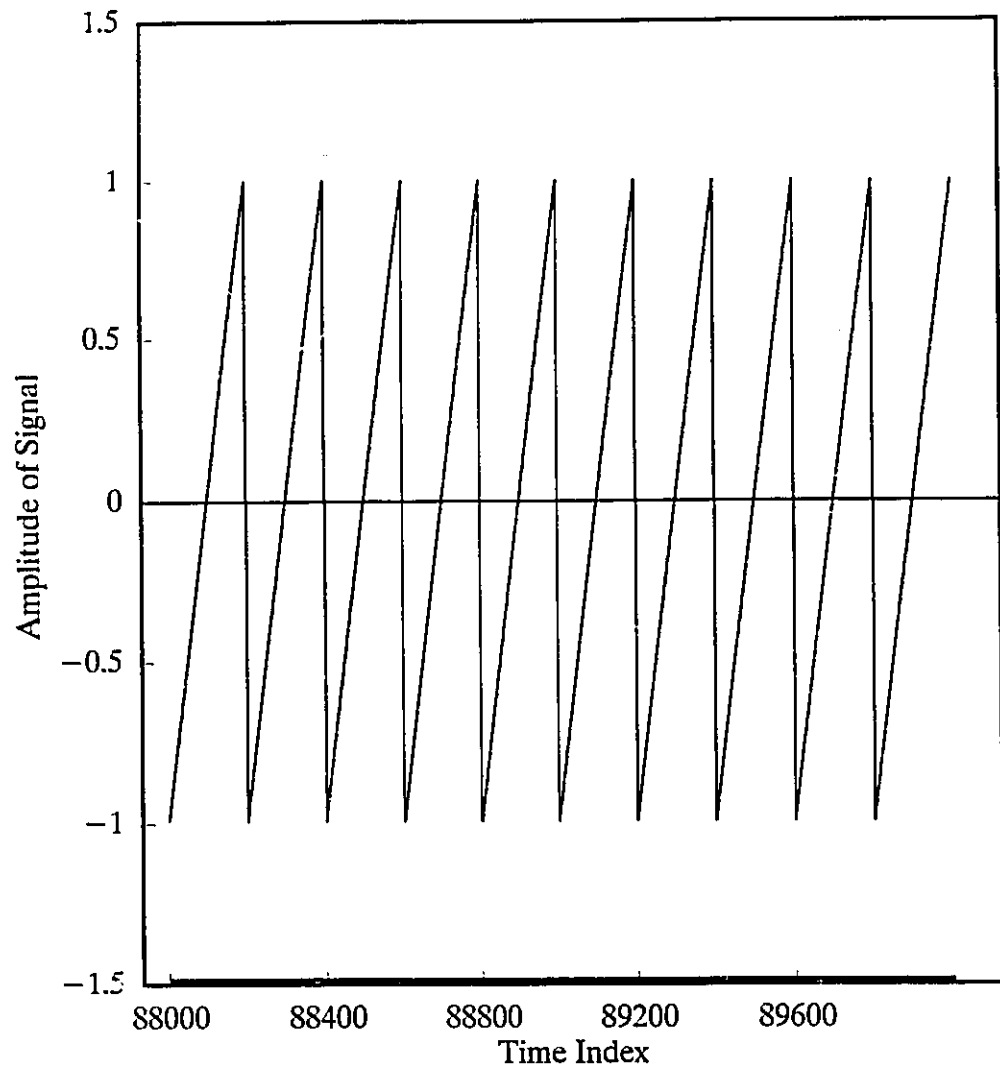


(c) Recovered sinusoidal signal

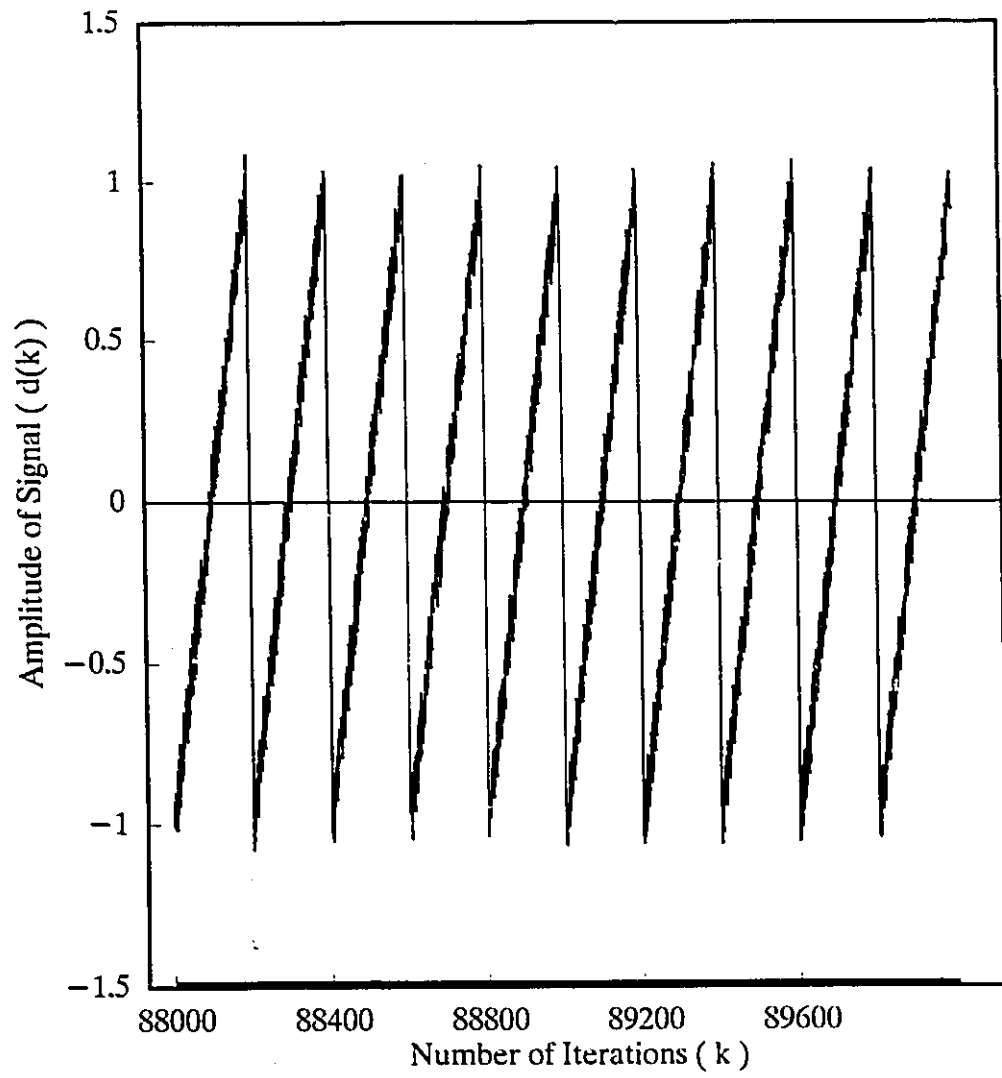


(d) Squared output error

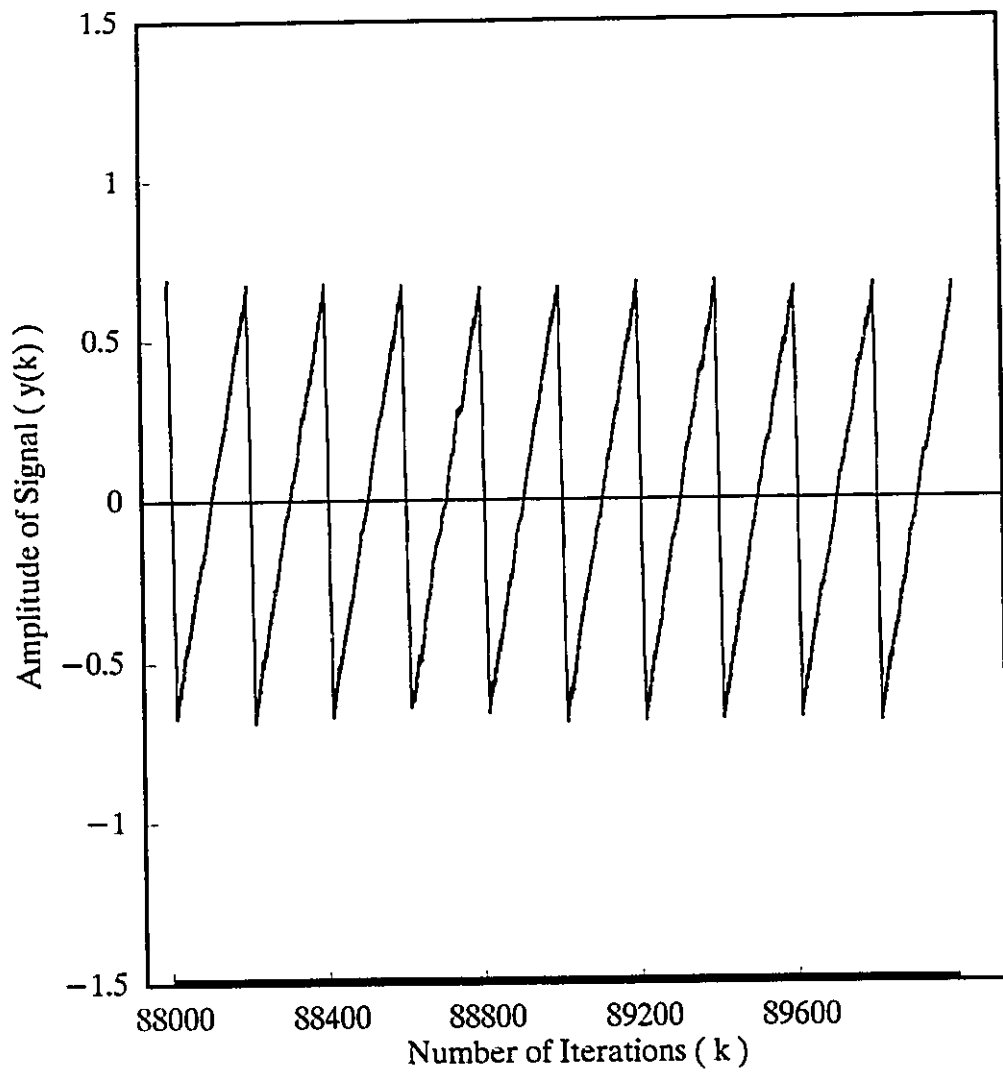
Fig. 2.8 Application of DNALP-FIR digital filter to adaptive noise cancellation of sinusoidal signal (11th-order, 4-path, even symmetry)



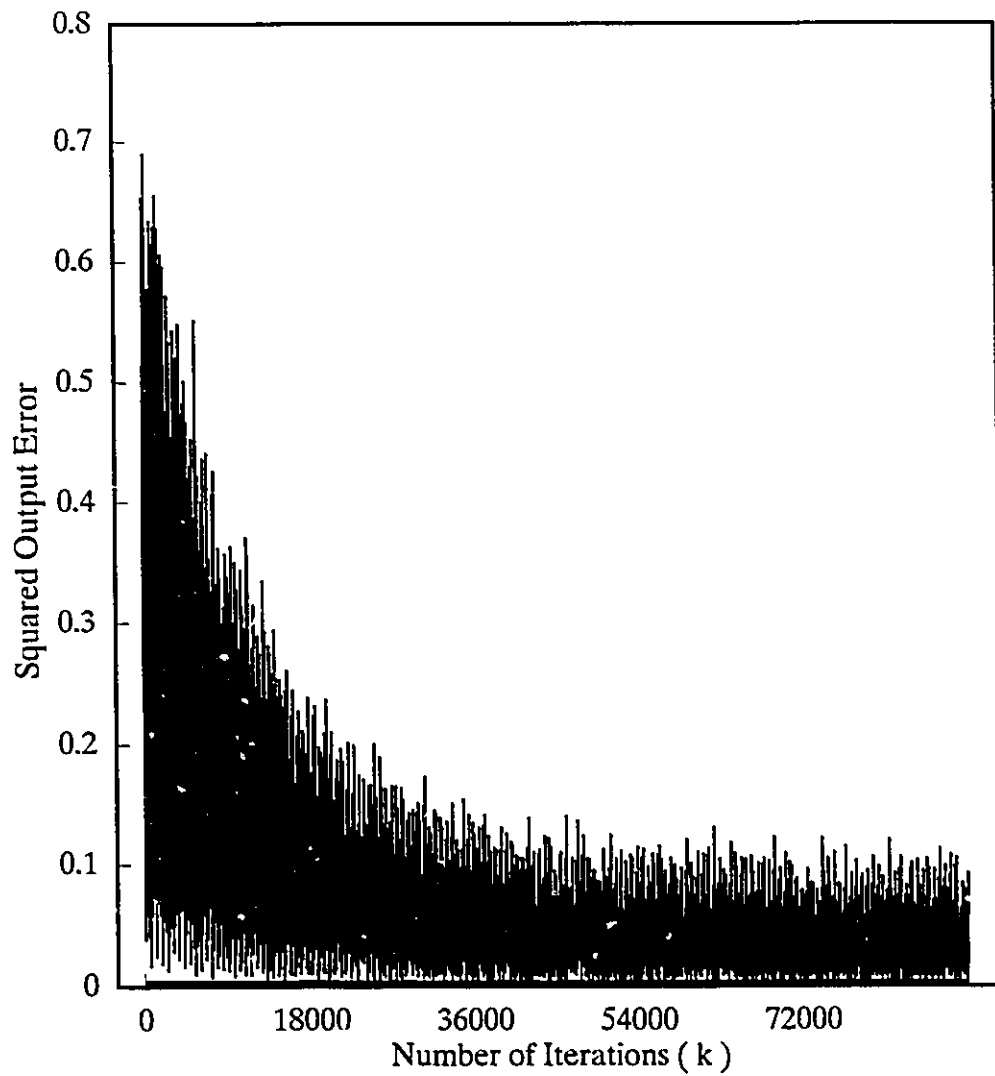
(a) Original sawtooth signal



(b) Corrupted input signal



(c) Recovered sawtooth signal



(d) Squared output error

Fig. 2.9 Application of DNALP-FIR digital filter to adaptive noise cancellation of sawtooth signal (11th-order, 2-path, even symmetry)

2.2.4.2 Application to Adaptive System Modelling

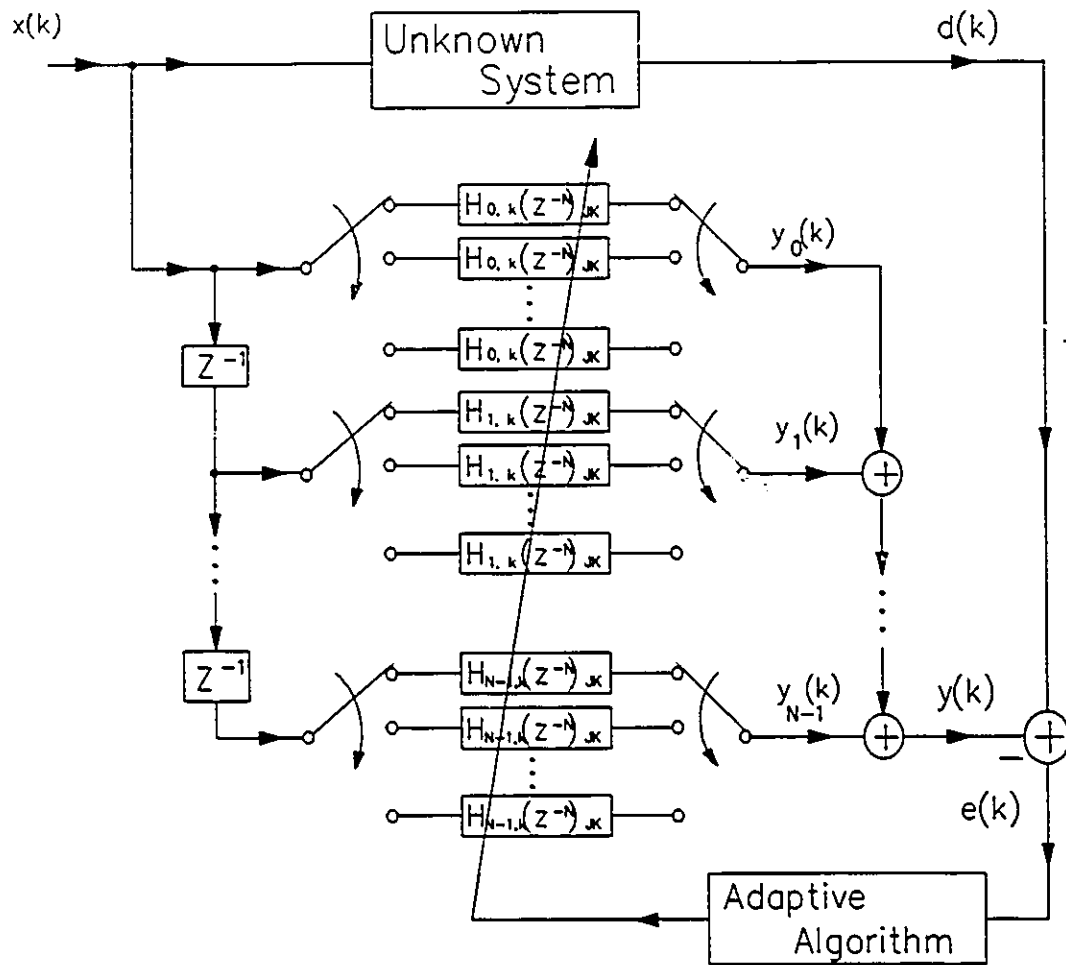


Fig. 2.10 Block diagram of DNALP-FIR digital filter for adaptive system modelling (where J can be E for even M and O for odd M ; K can be E for even symmetry and O for odd symmetry)

Adaptive system modelling, or adaptive system identification, is another important application area of adaptive filters. Fig. 2.10 shows the block diagram of DNALP-FIR digital filter for adaptive modelling of linear phase systems. In light of this application, $d(k)$ represents the output of the unknown system with input signal $x(k)$, which is also the input signal to the adaptive filter. $e(k)$ represents the difference between $d(k)$ and $y(k)$ as shown in Fig. 2.10.

Example 1: An 11th-order even symmetry linear phase FIR digital filter

$$\begin{aligned} H_1(z^{-1}) = & 0.1159 + 0.0643z^{-1} - 0.4274z^{-2} - 0.0127z^{-3} \\ & - 0.0750z^{-4} - 0.4331z^{-5} - 0.4331z^{-6} - 0.0750z^{-7} \\ & - 0.0127z^{-8} - 0.4274z^{-9} + 0.0643z^{-10} + 0.1159z^{-11} \end{aligned}$$

is to be modelled by the DNALP-FIR digital filter with $N=2$. The output error curve is shown in Fig. 2.11. In this example, the input signal $x(k)$ is a white random sequence with mean value of 0 and variance of $1/12$. All the initial weight values were set to 0 and the convergence constant was 0.1. The estimated coefficients after 716 iterations with the output error falling below 1.0×10^{-7} are

Path 1:

$$\begin{array}{ll} h(0) = 0.1159281143 & h(1) = 0.0643419902 \\ h(2) = -0.4273836326 & h(3) = -0.0127506164 \\ h(4) = -0.0750226942 & h(5) = -0.4331130216 \end{array}$$

Path 2:

$$\begin{aligned}
 h(0) &= 0.1159250814 & h(1) &= 0.0643426474 \\
 h(2) &= -0.4273885084 & h(3) &= -0.0127460453 \\
 h(4) &= -0.0750265289 & h(5) &= -0.4331121956
 \end{aligned}$$

Example 2: A 24th-order even symmetry linear phase FIR digital filter

$$\begin{aligned}
 H_2(z^{-1}) &= -0.075695 + 0.201699z^{-1} + 0.473032z^{-2} \\
 &+ 0.348487z^{-3} - 0.346237z^{-4} - 0.056987z^{-5} \\
 &+ 0.460028z^{-6} - 0.245313z^{-7} + 0.333599z^{-8} \\
 &+ 0.208192z^{-9} - 0.249571z^{-10} - 0.374330z^{-11} \\
 &+ 0.494087z^{-12} - 0.374330z^{-13} - 0.249571z^{-14} \\
 &+ 0.208192z^{-15} + 0.333599z^{-16} - 0.245313z^{-17} \\
 &+ 0.460028z^{-18} - 0.056987z^{-19} - 0.346237z^{-20} \\
 &+ 0.348487z^{-21} + 0.473032z^{-22} + 0.201699z^{-23} \\
 &- 0.075695z^{-24}
 \end{aligned}$$

is to be modelled by a delayed 5-path structure. Fig. 2.12 shows the output error curve. The input signal $x(k)$ is a white random sequence with mean value of 0 and variance of $1/12$. All the initial weight values were set to 0 and the convergence constant was 0.1. The final estimated coefficients after 4390 iterations with the output error falling below 1.0×10^{-7} are

Path 1:

$h(0) = -0.0756950123$	$h(1) = 0.2016992377$
$h(2) = 0.4730319768$	$h(3) = 0.3484869265$
$h(4) = -0.3462371581$	$h(5) = -0.0569867848$
$h(6) = 0.4600277083$	$h(7) = -0.2453127287$
$h(8) = 0.3335990980$	$h(9) = 0.2081918509$
$h(10) = -0.2495708887$	$h(11) = -0.3743295497$
$h(12) = 0.4940870675$	

Path 2:

$h(0) = -0.0756950077$	$h(1) = 0.2016993081$
$h(2) = 0.4730320020$	$h(3) = 0.3484869469$
$h(4) = -0.3462371597$	$h(5) = -0.0569867711$
$h(6) = 0.4600277428$	$h(7) = -0.2453126940$
$h(8) = 0.3335991076$	$h(9) = 0.2081918970$
$h(10) = -0.2495708672$	$h(11) = -0.3743295503$
$h(12) = 0.4940869750$	

Path 3:

$h(0) = -0.0756950178$	$h(1) = 0.2016992224$
$h(2) = 0.4730320760$	$h(3) = 0.3484869266$
$h(4) = -0.3462371568$	$h(5) = -0.0569867713$

$$\begin{aligned}h(6) &= 0.4600277785 & h(7) &= -0.2453126650 \\h(8) &= 0.3335990543 & h(9) &= 0.2081918759 \\h(10) &= -0.2495708260 & h(11) &= -0.3743296161 \\h(12) &= 0.4940869467\end{aligned}$$

Path 4:

$$\begin{aligned}h(0) &= -0.0756950755 & h(1) &= 0.2016992002 \\h(2) &= 0.4730318833 & h(3) &= 0.3484868453 \\h(4) &= -0.3462371354 & h(5) &= -0.0569868757 \\h(6) &= 0.4600276738 & h(7) &= -0.2453126198 \\h(8) &= 0.3335990719 & h(9) &= 0.2081918534 \\h(10) &= -0.2495708522 & h(11) &= -0.3743295948 \\h(12) &= 0.4940869292\end{aligned}$$

Path 5:

$$\begin{aligned}h(0) &= -0.0756950135 & h(1) &= 0.2016992290 \\h(2) &= 0.4730319805 & h(3) &= 0.3484869112 \\h(4) &= -0.3462371571 & h(5) &= -0.0569867757 \\h(6) &= 0.4600277196 & h(7) &= -0.2453126627 \\h(8) &= 0.3335990880 & h(9) &= 0.2081918276 \\h(10) &= -0.2495708969 & h(11) &= -0.3743295746 \\h(12) &= 0.4940869797\end{aligned}$$

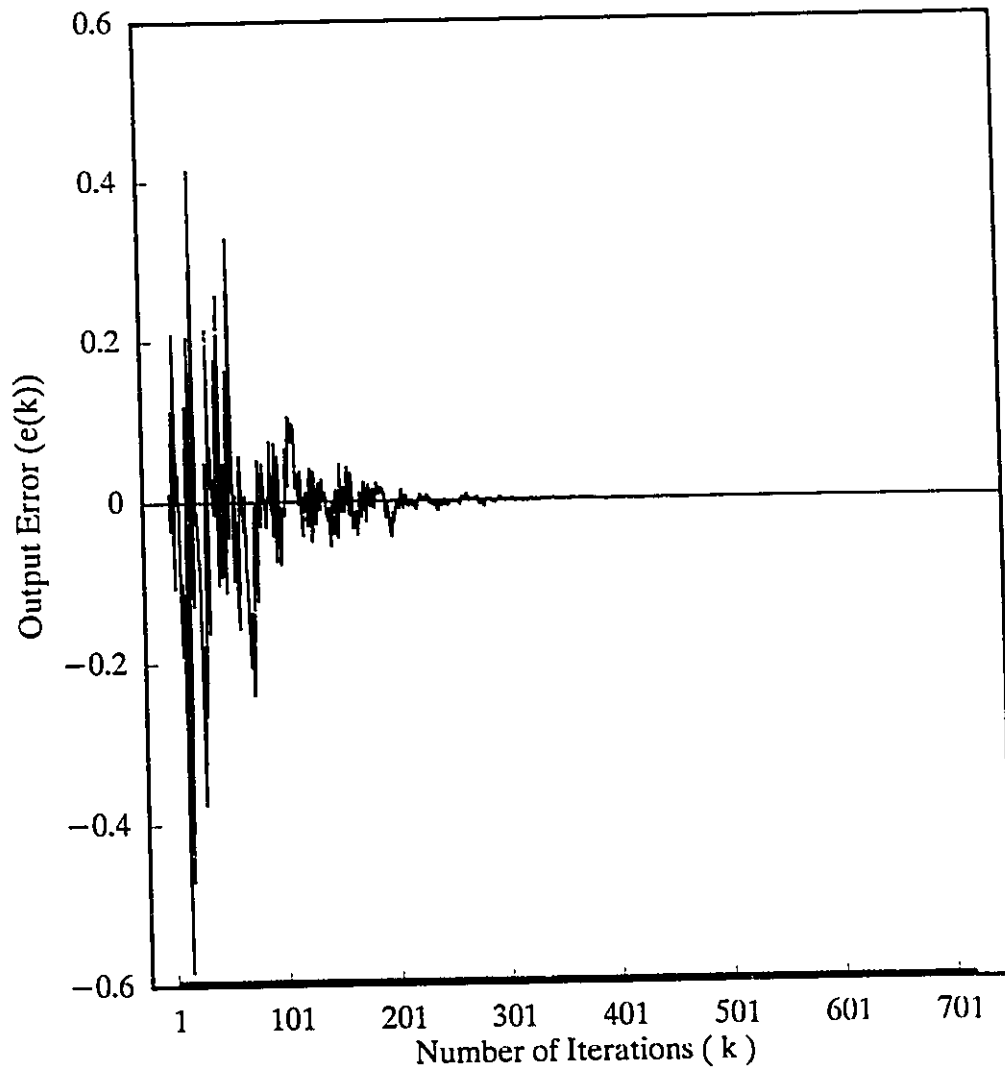


Fig. 2.11 Output error of DNALP-FIR digital filter for adaptive system modelling (11th-order, 2-path, even symmetry)

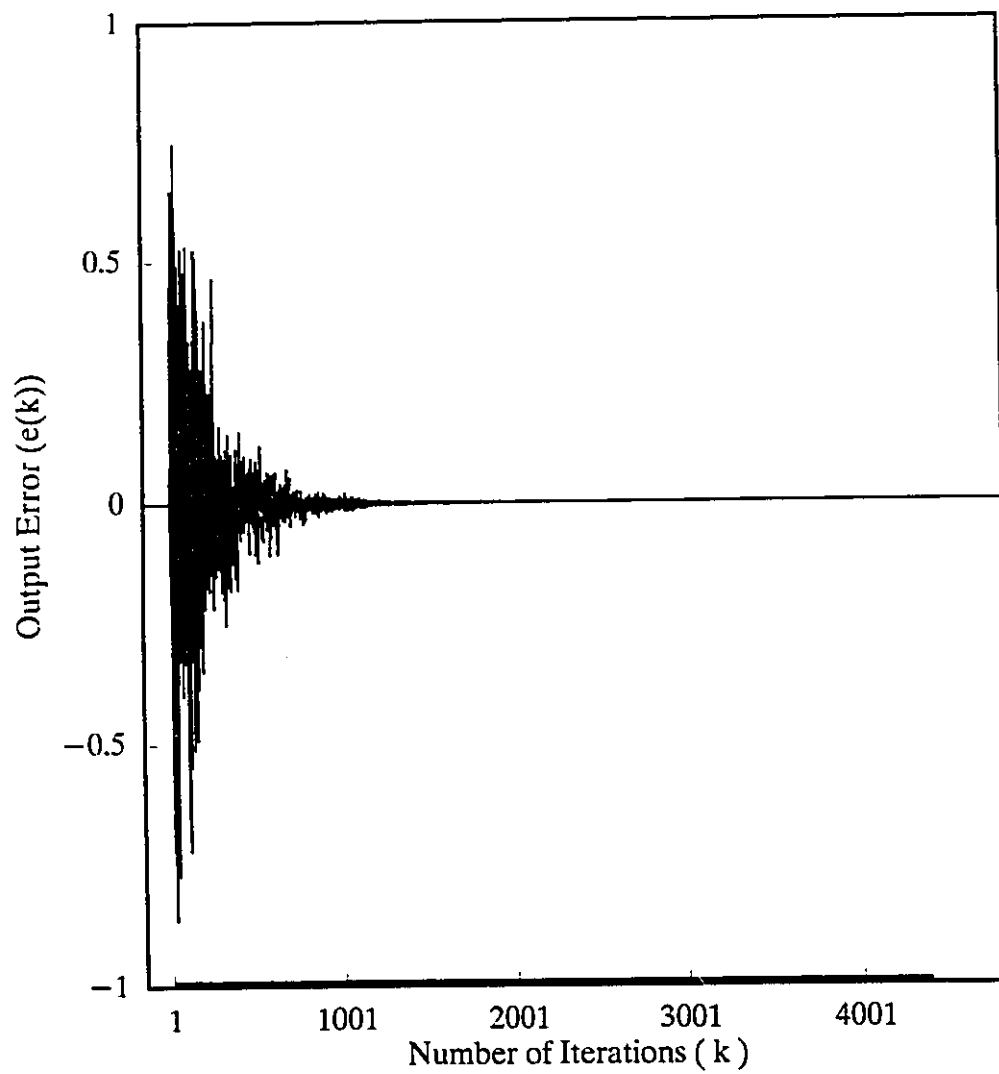


Fig. 2.12 Output error of DNALP-FIR digital filter for adaptive system modelling (24th-order, 5-path, even symmetry)

2.2.5 Conclusions

A delayed N-path adaptive linear phase FIR (DNALP-FIR) digital filter structure has been presented in this paper. The proposed filter structure is very useful for high speed adaptive linear phase digital filtering for various applications such as communications, radar, geophysical and seismic processing, and spectrum and image analysis, etc. It can be constructed using N^2 DSPs. Based on this structure, an adaptive algorithm has been developed which is applicable to any N-path realization. The maximum throughput rate of the filter can be $2N^2$ times that of a conventional adaptive FIR digital filter realized by one DSP. Simulation results on the use of the proposed structure for adaptive noise cancellation and adaptive system modelling have also been presented.

2.3 Delayed N-Path Equation-Error based Adaptive IIR (DNEEBA-IIR) Digital Filter

2.3.1 Introduction

Considerable effort has been focused on the studies of adaptive IIR digital filtering because of efficiency in terms of reduced filter order [50-52]. However, adaptive IIR digital filters have two disadvantages compared to adaptive FIR digital filters: (1) they become unstable if poles move outside the unit circle during the adaptive process; and (2) their performance surfaces are generally nonquadratic and may even have local minima. So far some methods have been suggested to solve the stability problem inherent

in adaptive IIR digital filters, for examples, [53-55].

In this section, we will present a delayed N-path equation-error based adaptive IIR (DNEEBA-IIR) digital filter. In a DNEEBA-IIR digital filter, each of the numerator and the denominator transfer functions is decomposed into N paths, and each of the N paths is realized by N DSPs. Hence, the overall maximum throughput rate can be improved by $2N^2$ times. The equation-error based IIR digital filter structure [56-57] is used to formulate the DNEEBA-IIR digital filter because the stability problem can be avoided during adaptation.

In the following, we shall describe how a DNA-FIR digital filter structure developed in Section 2.1 can be adapted for the realization of the proposed DNEEBA-IIR digital filter structure. Fig. 2.13 shows the block diagram of equation-error based adaptive IIR digital filter. We can see from Fig. 2.13 that the DNA-FIR digital filter structure can be applied to each of the numerator polynomial $A(z^{-1})$ and the denominator polynomial $B(z^{-1})$ of an equation-error based IIR digital filter to form a DNEEBA-IIR digital filter. In the actual realization, each path has N sub-filters where N DSPs are used to realize both the variable coefficient sub-filter and its adaptive algorithm. In this case, $2N^2$ DSPs are needed in a DNEEBA-IIR digital filter structure, where N^2 DSPs are used for $A(z^{-1})$ and other N^2 DSPs are used for $B(z^{-1})$.

A simple variable step algorithm is adopted here instead of the conventional LMS

algorithm where the step parameter μ is fixed. The step parameter μ controls the convergence rate of the filter coefficients and also determines the final excess mean square error as compared to the optimal Wiener solution [2, 7-8]. While convergence speed is increased when a large μ is selected, the misadjustment or error residual is also increased. By using this variable step algorithm, the step parameter μ is adjusted according to the output equation error. It is initially selected large enough which would result in fast convergence at first, and then it is reduced gradually as output error decreases. Finally a small misadjustment is reached due to a small μ at the final stage.

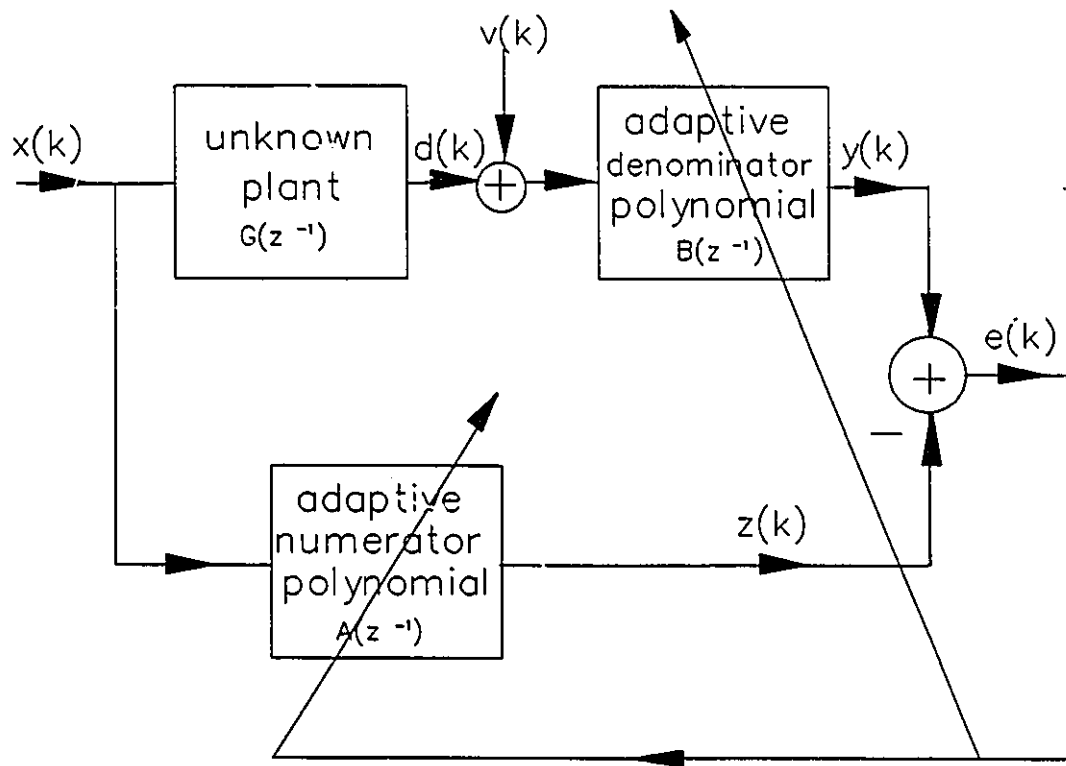


Fig. 2.13 Equation-error based adaptive IIR digital filter

2.3.2 Structure and Adaptive Algorithm of DNEEBA-IIR Digital Filter

Fig. 2.13 shows the block diagram of an equation-error based adaptive IIR digital filter. $x(k)$ is the input signal which is to be processed by an unknown plant $G(z^{-1})$. The output signal $d(k)$ of the unknown plant $G(z^{-1})$ plus the measurement noise $v(k)$, and the input signal $x(k)$ are to be processed, respectively, by the denominator adaptive polynomial, $B(z^{-1})$, and the numerator adaptive polynomials, $A(z^{-1})$, where

$$A(z^{-1}) = \sum_{m=0}^{M_A-1} a(m)z^{-m} \quad (2.40a)$$

and

$$B(z^{-1}) = 1 + \sum_{m=1}^{M_B-1} b(m)z^{-m} \quad (2.40b)$$

The squared equation error signal is defined as

$$e^2(k) = [y(k) - z(k)]^2 \quad (2.41)$$

The associated adaptive algorithm adapts the polynomials $A(z^{-1})$ and $B(z^{-1})$ or the modelling filter $A(z^{-1})/B(z^{-1})$ in such a way that $e^2(k)$ is minimized.

For high speed computation, delayed N-path FIR digital filter structures are used to realize the polynomials $A(z^{-1})$ and $B(z^{-1})$. Let us rewrite the polynomials $A(z^{-1})$ and $B(z^{-1})$ at a discrete time k as

$$A_k(z^{-1}) = \sum_{m=0}^{M_A-1} a_k(m)z^{-m} \quad (2.42a)$$

and

$$B_k(z^{-1}) = \sum_{m=0}^{M_r-1} b_k(m)z^{-m} \quad (2.42b)$$

where $b_k(0)=1$. Eqn. 2.42a and 2.42b can be decomposed into the following forms which are suitable for delayed N-path realization, i.e.,

$$A_k(z^{-1}) = \sum_{i=0}^{N-1} A_{i,k}(z^{-N})z^{-i} \quad (2.43a)$$

and

$$B_k(z^{-1}) = \sum_{i=0}^{N-1} B_{i,k}(z^{-N})z^{-i} \quad (2.43b)$$

The expressions of $A_{i,k}(z^{-N})$ and $B_{i,k}(z^{-N})$ for $i = 0, 1, \dots, N-1$ are given by

$$A_{i,k}(z^{-N}) = \sum_{l=0}^{L_{Ai}} a_k(lN+i)z^{-lN} \quad (2.44a)$$

and

$$B_{i,k}(z^{-N}) = \sum_{l=0}^{L_{Bi}} b_k(lN+i)z^{-lN} \quad (2.44b)$$

where

$$L_{Ai} = [M_A/N]^* \quad \text{for } i = 0, 1, \dots, J_A-1 \quad (2.45a)$$

$$L_{Ai} = [M_A/N]^* - 1 \quad \text{for } i = J_A, J_A+1, \dots, N-1 \quad (2.45b)$$

$$J_A = (M_A)_N \quad (2.45c)$$

and

$$L_{bi} = [M_B/N]^* \quad \text{for } i = 0, 1, \dots, J_B-1 \quad (2.46a)$$

$$L_{bi} = [M_B/N]^* - 1 \quad \text{for } i = J_B, J_B+1, \dots, N-1 \quad (2.46b)$$

$$J_B = (M_B)_N \quad (2.46c)$$

The resultant delayed N-path equation-error based adaptive IIR digital filter structure is shown in Fig. 2.14.

Denoting the z-transform of $x(k)$, $v(k)$, $d(k)$, $y(k)$, and $z(k)$ as $X(z^{-1})$, $V(z^{-1})$, $D(z^{-1})$, $Y(z^{-1})$, and $Z(z^{-1})$, respectively, we have the following relationships

$$\begin{aligned} Y(z^{-1}) &= B_k(z^{-1})(D(z^{-1}) + V(z^{-1})) \\ &= \sum_{i=0}^{N-1} B_{i,k}(z^{-M})z^{-i}(D(z^{-1}) + V(z^{-1})) \\ &= \sum_{i=0}^{N-1} Y_i(z^{-1}) \end{aligned} \quad (2.47)$$

where

$$Y_i(z^{-1}) = B_{i,k}(z^{-M})(D(z^{-1}) + V(z^{-1}))z^{-i} \quad (2.48)$$

is the z-transform of $y_i(k)$ which is the output of the i th path in $B_k(z^{-1})$ as shown in Fig. 2.14. Similarly, we have

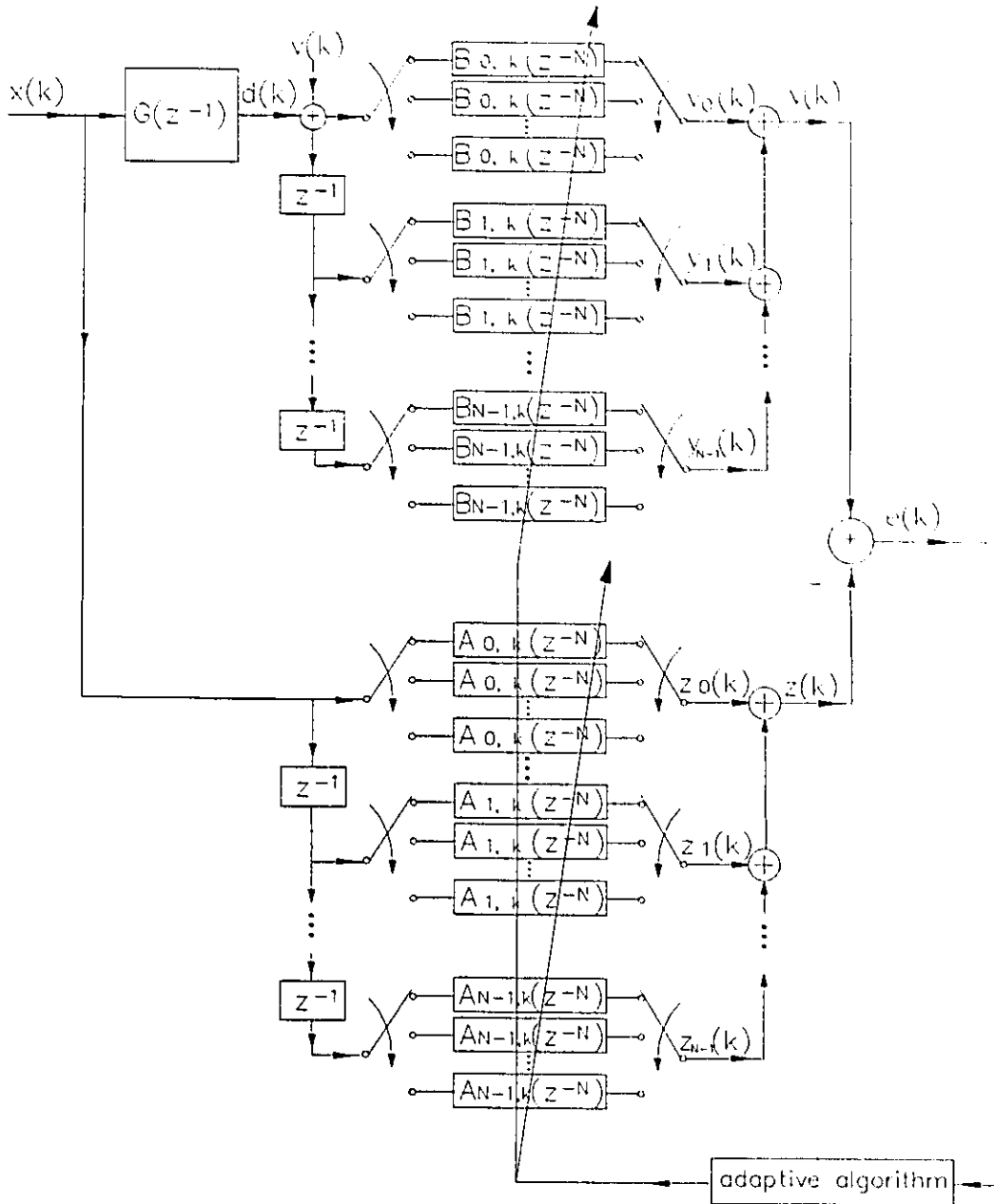


Fig. 2.14 DNEEBA-IIR digital filter structure

$$\begin{aligned}
Z(z^{-1}) &= A_k(z^{-1})X(z^{-1}) \\
&= \sum_{i=0}^{N-1} A_{i,k}(z^{-N})z^{-i}X(z^{-1}) \\
&= \sum_{i=0}^{N-1} Z_i(z^{-1})
\end{aligned} \tag{2.49}$$

where

$$Z_i(z^{-1}) = A_{i,k}(z^{-N})X(z^{-1})z^{-i} \tag{2.50}$$

is the z-transform of $z_i(k)$ which is the output of the i th path in $A_k(z^{-1})$ as shown in Fig. 2.14.

Substituting eqns. 2.44b and 2.44a into eqns. 2.48 and 2.50, respectively, and then taking their inverse z-transforms, we obtain

$$y_i(k) = \sum_{l=0}^{L_n} b_k(lN+i)(d(k-lN-i) + v(k-lN-i)) \tag{2.51}$$

and

$$z_i(k) = \sum_{l=0}^{L_n} a_k(lN+i)x(k-lN-i) \tag{2.52}$$

Taking the inverse z-transform of eqns. 2.47 and 2.49, we obtain

$$y(k) = \sum_{i=0}^{N-1} y_i(k) \tag{2.53}$$

and

$$z(k) = \sum_{i=0}^{N-1} z_i(k) \quad (2.54)$$

Substituting eqns. 2.53 and 2.54 into eqn. 2.41, we get the instantaneous equation error as

$$e^2(k) = \sum_{i=0}^{N-1} [y_i(k) - z_i(k)]^2 \quad (2.55)$$

Define the changes of the coefficients, $\Delta a_k(lN+i)$ and $\Delta b_k(lN+i)$, as

$$\Delta a_k(lN+i) = -\mu \frac{\partial e^2(k)}{\partial a_k(lN+i)} \quad (2.56a)$$

and

$$\Delta b_k(lN+i) = -\mu \frac{\partial e^2(k)}{\partial b_k(lN+i)} \quad (2.56b)$$

where μ is the convergence constant or step parameter. From eqns. 2.56a and 2.41, we obtain

$$\begin{aligned} a_{k,N}(lN+i) &= a_k(lN+i) + 2\mu e(k)\alpha(k, lN+i) \\ \text{for } i &= 0, 1, \dots, N-1; \quad l = 0, 1, \dots, L_{Ai} \end{aligned} \quad (2.57a)$$

and from eqns. 2.56b and 2.41, we obtain

$$\begin{aligned} b_{k,N}(lN+i) &= b_k(lN+i) - 2\mu e(k)\beta(k, lN+i) \\ \text{for } i &= 0, 1, \dots, N-1; \quad l = 0, 1, \dots, L_{Bi} \end{aligned} \quad (2.57b)$$

where

$$\alpha(k, lN+i) = \frac{\partial z(k)}{\partial a_k(lN+i)} \quad (2.58a)$$

and

$$\beta(k, lN+i) = \frac{\partial y(k)}{\partial b_k(lN+i)} \quad (2.58b)$$

From eqns. 2.54 and 2.52, and eqns. 2.53 and 2.51, we obtain

$$\alpha(k, lN+i) = x(k-lN-i) \quad (2.59a)$$

and

$$\beta(k, lN+i) = d(k-lN-i) + v(k-lN-i) \quad (2.59b)$$

2.3.3 Variable Step Adaptive Algorithm

The conventional LMS algorithm uses a fixed step parameter constant μ , which controls the convergence rate of the filter coefficients but also determines the final excess mean-square error as compared to the optimal Wiener solution [2, 7-8]. A trade-off is usually necessary to select a constant μ so that the convergence speed is not too slow and also the final misadjustment is not too large. However, this does not necessarily work well in all cases. In the following, we suggest a simple variable step method.

In this method, we always fix the step parameter of the numerator polynomial $A_k(z^{-1})$ but allow the step parameter of the denominator polynomial $B_k(z^{-1})$ to vary. Hence, eqns. 2.57a and 2.57b can be rewritten as follows:

$$\begin{aligned}
 a_{k,N}(lN+i) &= a_k(lN+i) - 2\mu_A e(k)\alpha(k, lN+i) \\
 \text{for } i &= 0, 1, \dots, N-1; \quad l = 0, 1, \dots, L_{Ai}
 \end{aligned} \tag{2.60a}$$

and

$$\begin{aligned}
 b_{k,N}(lN+i) &= b_k(lN+i) + 2\mu_B(k, i)e(k)\beta(k, lN+i) \\
 \text{for } i &= 0, 1, \dots, N-1; \quad l = 0, 1, \dots, L_{Bi}
 \end{aligned} \tag{2.60b}$$

where μ_A is a constant step parameter for $A_k(z^{-1})$, and $\mu_B(k, i)$ is a variable step parameter for $B_k(z^{-1})$. Initially, we select a sufficiently large parameter for the denominator $B_k(z^{-1})$ as $\mu_B(0, i) = \mu_{B0}$, for $i = 0, 1, \dots, N-1$, and then we reduce it gradually as the equation error decreases. This method can be summarized as follows:

$$\mu_B(k, i) = \frac{\mu_B(k-1, i)}{2^{T_j}} \quad \& \quad T_{j+1} = T_j + 1 \tag{2.61a}$$

$$\text{if } k = 10j+i \quad \& \quad EE_j \leq EE_{j-1}/10$$

$$\mu_B(k, i) = \mu_B(k-1, i) \quad \& \quad T_{j+1} = T_j \tag{2.61b}$$

otherwise

where

$$EE_j = \sum_{i=0}^9 e^2(10j-i) \tag{2.62}$$

and $i = 0, 1, \dots, N-1, j = 1, 2, \dots; EE_0=0$ and $T_1=1$.

2.3.4 Simulation Results

Three plants, each represented by a 5th-order IIR digital filter transfer function, are to be modelled by delayed 1-path (i.e., the direct form structure), delayed 2-path, delayed 3-path, and delayed 4-path equation error based adaptive IIR (EEBA-IIR) digital filters. In all the simulations, the input signal $x(k)$ was a white random sequence with a mean value of 0 and a variance of $1/12$, and initial coefficients were all set to zeroes.

The first plant to be identified is

$$H_1(z^{-1}) = \frac{1.0 - 0.5z^{-1} + 0.7z^{-2} + 1.1z^{-3} - 1.3z^{-4} + 0.9z^{-5}}{1.0 - z^{-1} + 0.58z^{-2} - 0.164z^{-3} + 0.0285z^{-4} - 0.0025z^{-5}}$$

The poles of the plant transfer function reside at $(0.1 \pm 0.2j)$, $(0.3 \pm 0.4j)$, and 0.2.

The second plant is represented by

$$H_2(z^{-1}) = \frac{1.0 - 0.5z^{-1} + 0.7z^{-2} + 1.1z^{-3} - 1.3z^{-4} + 0.9z^{-5}}{1.0 - 1.6z^{-1} + 1.25z^{-2} - 0.54z^{-3} + 0.1344z^{-4} - 0.01664z^{-5}}$$

The poles of the plant transfer function reside at $(0.2 \pm 0.3j)$, $(0.4 \pm 0.4j)$, and 0.4.

The third plant has the transfer function

$$H_3(z^{-1}) = \frac{1.0 - 0.5z^{-1} + 0.7z^{-2} + 1.1z^{-3} - 1.3z^{-4} + 0.9z^{-5}}{1.0 - 1.9z^{-1} + 1.84z^{-2} - 1.016z^{-3} + 0.3255z^{-4} - 0.05125z^{-5}}$$

with poles residing at $(0.3 \pm 0.4j)$, $(0.4 \pm 0.5j)$, and 0.5.

Tables 2.6 and 2.7 summarize the obtained simulation results, without and with measurement noise respectively, in terms of the number of iterations needed to converge to the specified stopping criterion. In Table 2.6, the stopping criterion was set to 1.5×10^{-3} which measured the average squared output equation error over 20 samples. The convergence constants were set to $\mu_A = 1.0 \times 10^{-1}$, $\mu_{B0} = 1.0 \times 10^{-2}$ for $H_1(z^{-1})$, $\mu_A = \mu_{B0} = 1.0 \times 10^{-2}$ for $H_2(z^{-1})$ and $H_3(z^{-1})$ respectively. In Table 2.7, due to the presence of the measurement noise, the value of the stopping criterion was increased to 1.5×10^{-2} , which measured the average squared output error over 20 samples. From Tables 2.6 and 2.7, we can observe that the number of iterations needed for a DNEEBA-IIR structure was larger than that of the conventional equation error based adaptive IIR (CEEBA-IIR) structure. This could be at most N times more than that of the CEEBA-IIR structure. As the maximum throughput rate of a DNEEBA-IIR structure (realized by $2N^2$ DSPs) can be $2N^2$ faster than that of a CEEBA-IIR structure (realized by one DSP), the convergence time needed by the DNEEBA-IIR structure can be $2N$ times less than that of the CEEBA-IIR structure.

Table 2.6: Convergence performance in terms of number of iterations of DNEEBA-IIR structures without measurement noise.

Plant	Stopping Criterion	1-path	2-path	3-path	4-path
$H_1(z^{-1})$	1.5×10^{-3}	1000	1000	1044	1220
$H_2(z^{-1})$	1.5×10^{-3}	2120	4080	5400	7560
$H_3(z^{-1})$	1.5×10^{-3}	2240	4060	5940	7560

Table 2.7: Convergence performance in terms of number of iterations of DNEEBA-IIR structures with measurement noise.

Plant	Stopping Criterion	1-path	2-path	3-path	4-path
$H_1(z^{-1})$	1.5×10^{-2}	1700	2720	2718	2720
$H_2(z^{-1})$	1.5×10^{-2}	3020	3020	8046	8640
$H_3(z^{-1})$	1.5×10^{-2}	3020	8640	8046	9280

For illustration, Figs. 2.15-2.17 show, respectively, the average squared error versus the number of iterations k for the delayed 2-path modelling of $H_1(z^{-1})$, the delayed 3-path modelling of $H_2(z^{-1})$, and the delayed 4-path modelling of $H_3(z^{-1})$ of Table 2.7. Both the equation-error curve and the real output error curve are shown for each case. Other average squared error curves are similar and are not shown here.

With a measurement noise $v(k)$ added to the output of a plant, a DNEEBA-IIR would result in a biased estimate which is also true in a conventional equation-error based IIR structure. On the other hand, increasing the adaptive filter order could reduce the number of iterations required to converge to the same specified difference between equation error and real error. Table 2.8 summarizes the comparison results where $H_1(z^{-1})$, $H_2(z^{-1})$, and $H_3(z^{-1})$ of Tables 2.6 and 2.7 were modelled respectively by delayed 2-path, delayed 3-path, and delayed 4-path EEBA-IIR digital filter structures of orders 5, 11, and 17. The stopping criteria were set to an average squared error difference between

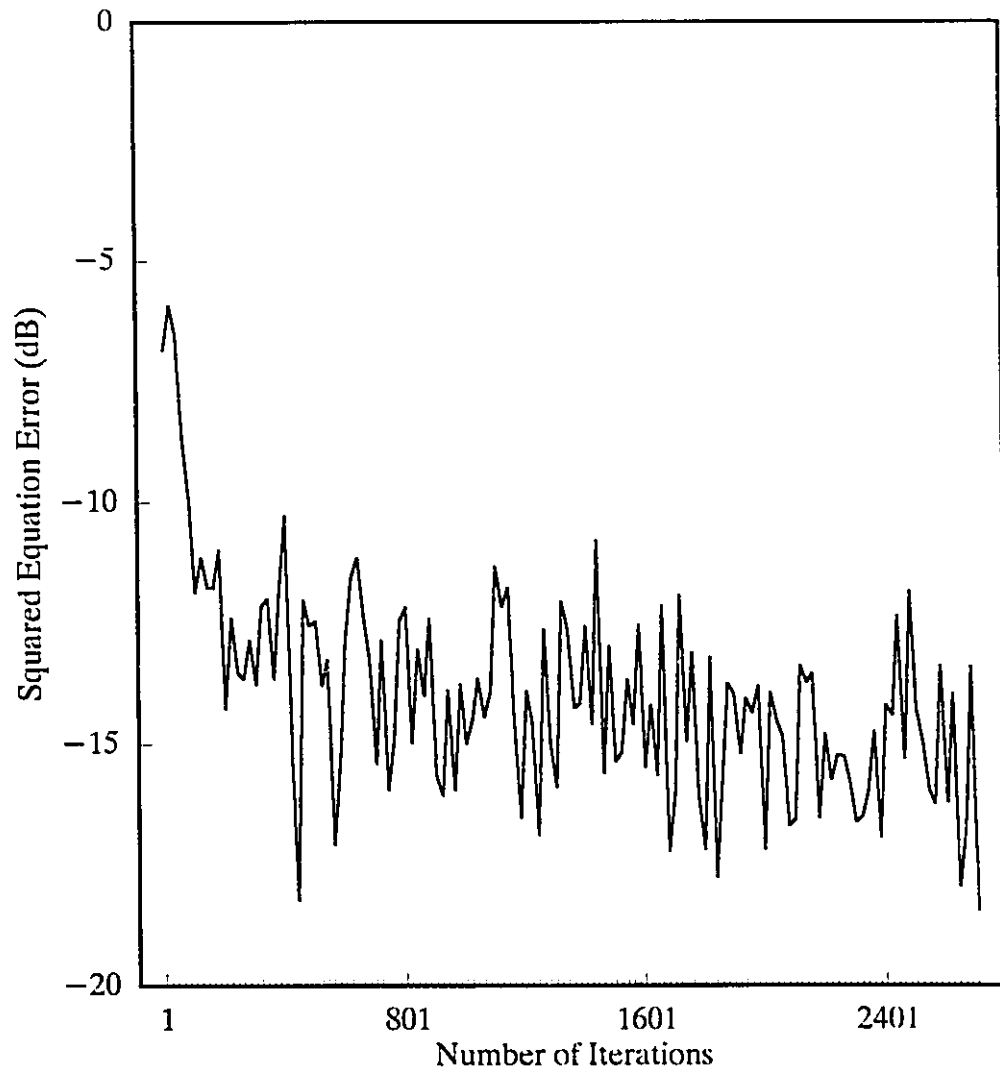


Fig. 2.15a Average squared equation-error of D2EEBA-IIR modelling of $H_1(z^{-1})$

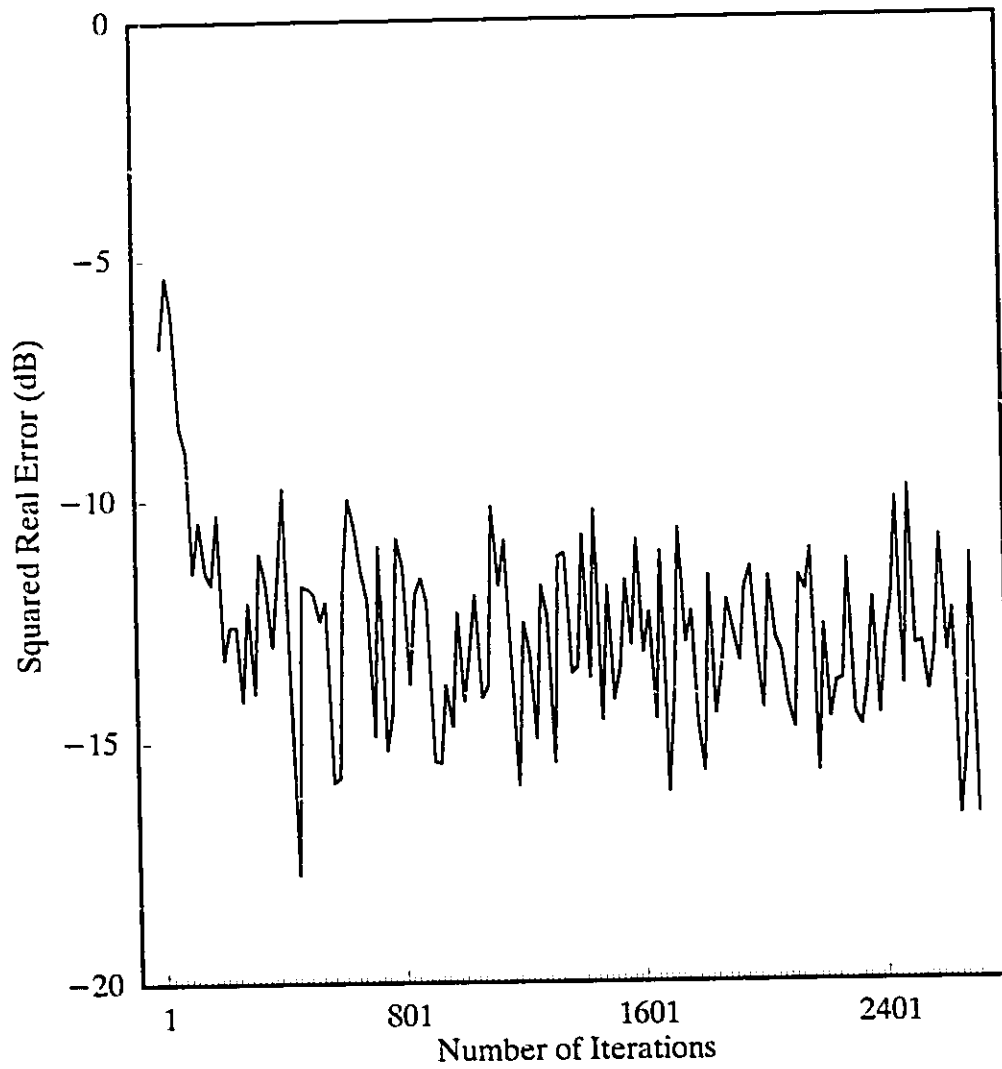


Fig. 2.15b Average squared real output error of D2EEBA-IIR modelling of $H_1(z^{-1})$

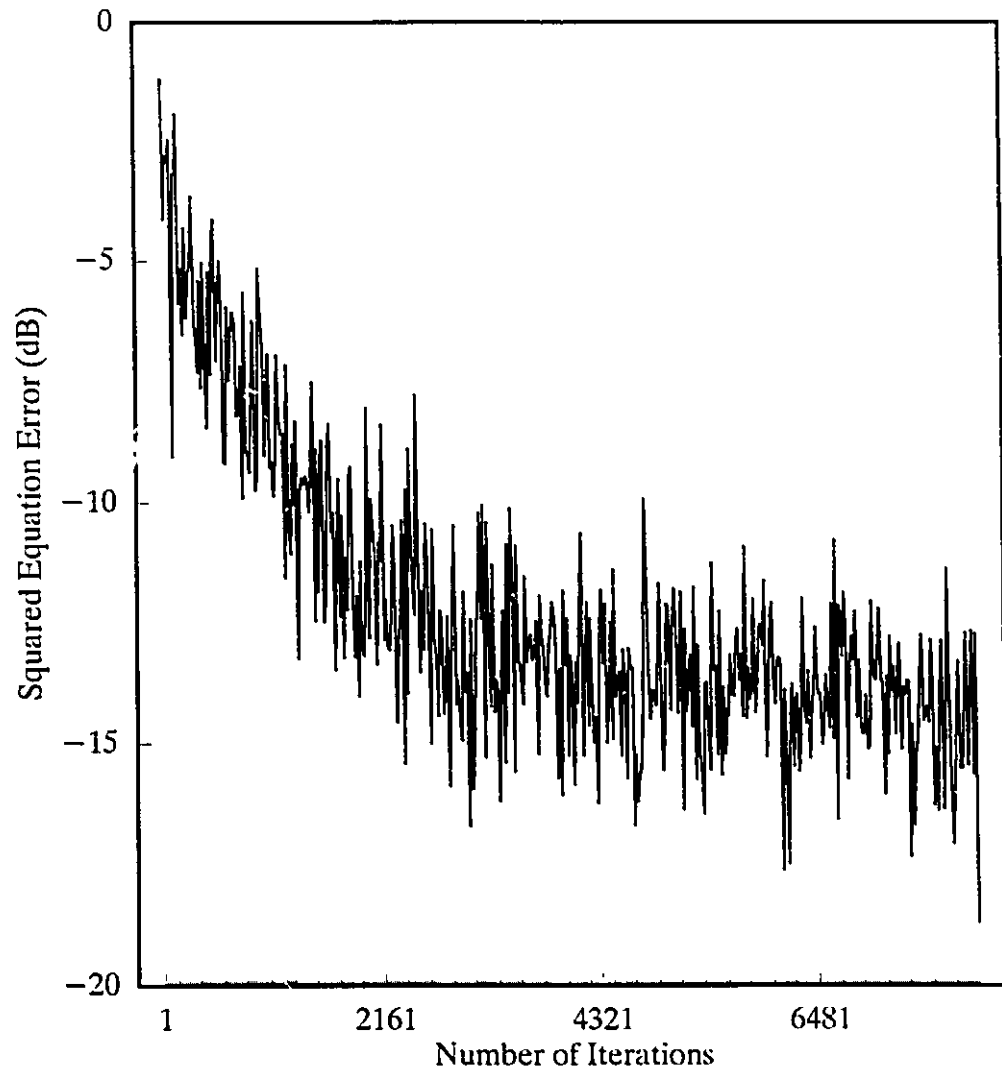


Fig. 2.16a Average squared equation-error of D3EEBA-IIR modelling of $H_2(z^{-1})$.

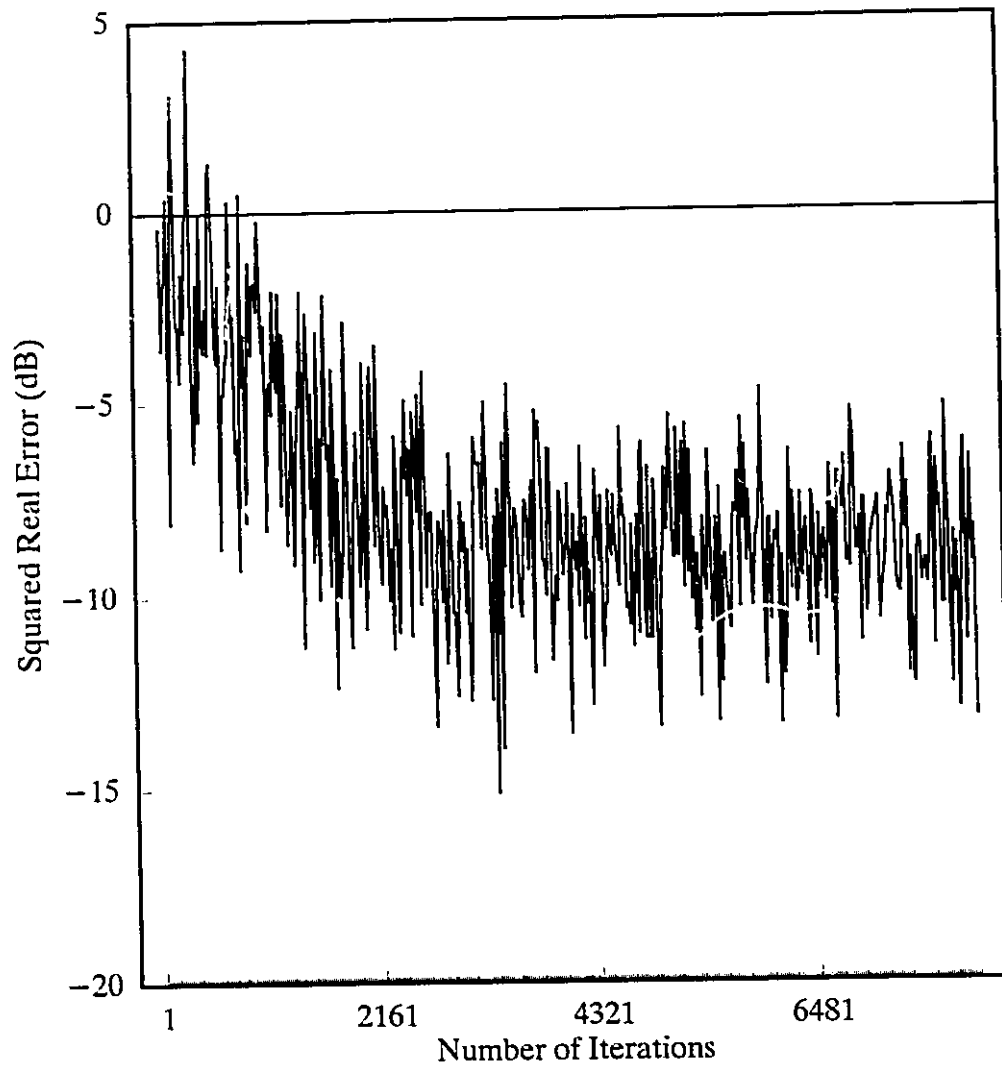


Fig. 2.16b Average squared real output error of D3EEBA-IIR modelling of $H_2(z^{-1})$

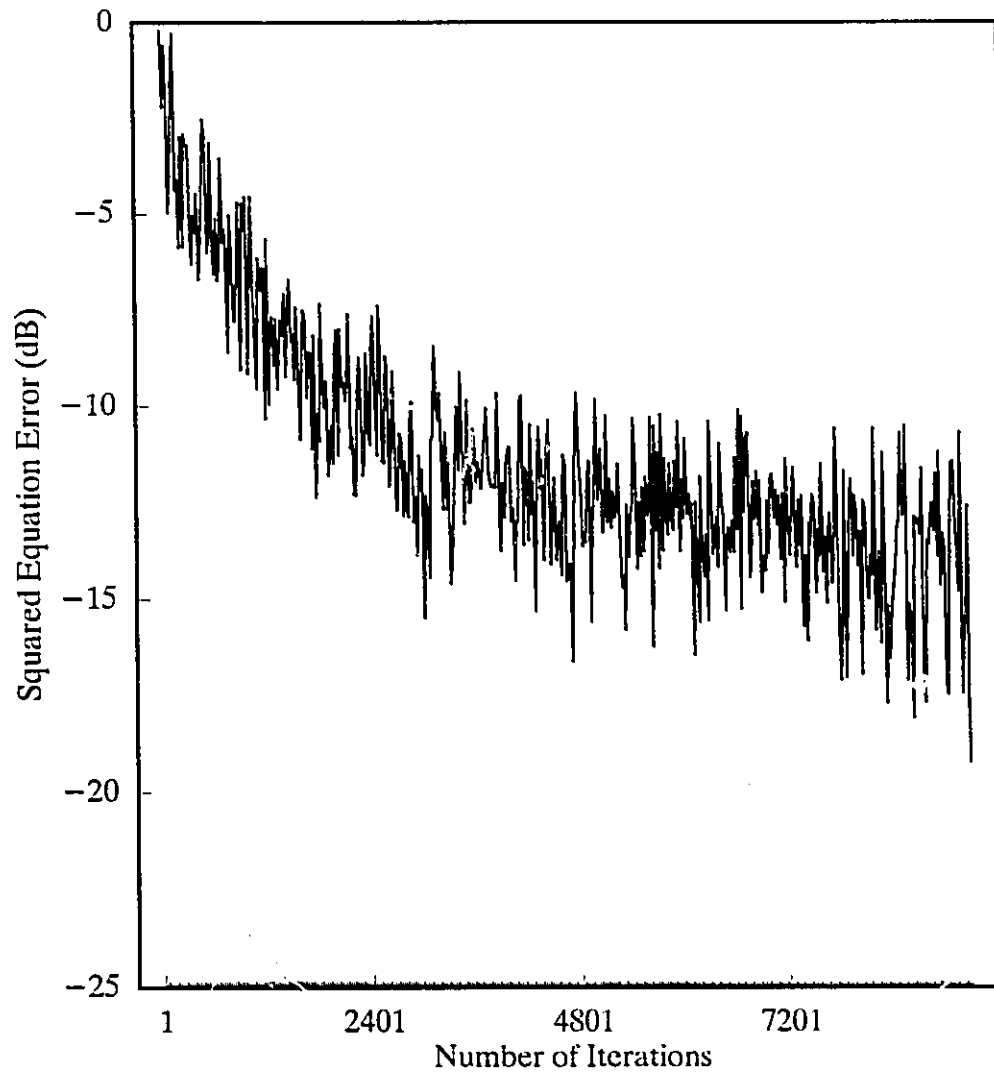


Fig. 2.17a Average squared equation-error of D4EEBA-IIR modelling of $H_3(z^{-1})$

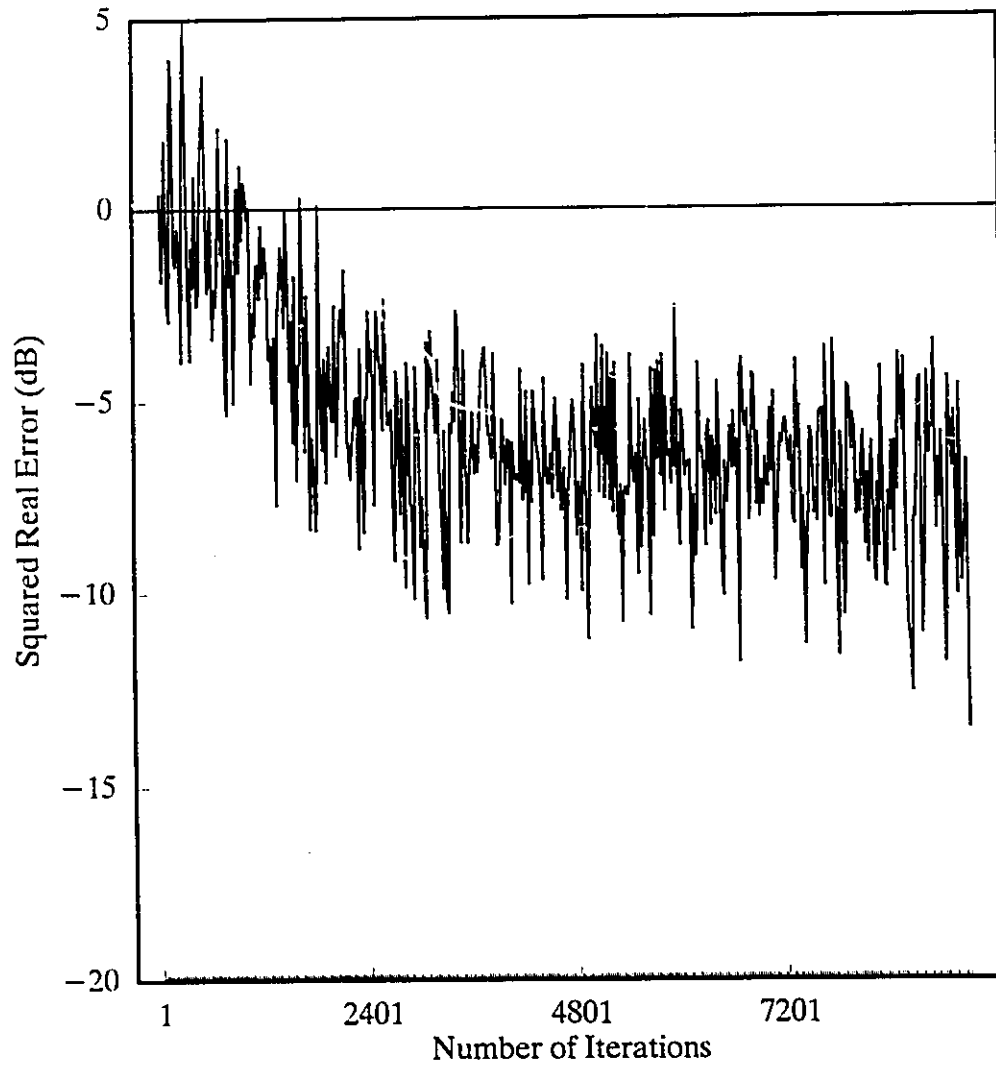


Fig. 2.17b Average squared real output error of D4EEBA-IIR modelling of $H_3(z^{-1})$

the equation error and real error of 1.0×10^{-3} , 5.0×10^{-3} , and 1.8×10^{-2} over 40 samples for $H_1(z^{-1})$, $H_2(z^{-1})$, and $H_3(z^{-1})$ respectively. The values of μ_A and μ_{B0} used were 1.0×10^{-2} and 1.0×10^{-2} for $H_1(z^{-1})$ and $H_2(z^{-1})$, and 1.0×10^{-2} and 1.0×10^{-3} for $H_3(z^{-1})$, respectively. We can observe from Table 2.8 that when a higher order DNEEBA-IIR structure was used, the difference between the equation error and the real output error could be reduced.

Table 2.8: Effect of increased filter order on number of convergent iterations of DNEEBA-IIR structures.

Plant	Stopping Criterion	No. of Paths	5th-order	11th-order	17th-order
$H_1(z^{-1})$	1.0×10^{-3}	2	35040	2000	1240
$H_2(z^{-1})$	5.0×10^{-3}	3	163449	163098	163098
$H_3(z^{-1})$	1.8×10^{-2}	4	1053680	330680	7120

A comparison of the convergence performance between the DNA-FIR and the DNEEBA-IIR structures was also carried out. In this comparison, the three plants $H_1(z^{-1})$, $H_2(z^{-1})$, and $H_3(z^{-1})$ of Tables 2.6 and 2.7 were modelled, respectively, by delayed 2-path, delayed 3-path, and delayed 4-path adaptive FIR and EEBA-IIR structures. Simulation results were summarized in Table 2.9 in terms of the number of iterations to converge to the same specified average squared error. In simulations, both DNA-FIR and DNEEBA-IIR structures had the same number of adjustable coefficients. The DNA-FIR structures used in Table 2.9 were 22th-order and the DNEEBA-IIR structures were 11th-

order. The stopping criteria for modelling $H_1(z^{-1})$, $H_2(z^{-1})$, and $H_3(z^{-1})$ were 3.0×10^{-2} , 5.0×10^{-2} , and 6.0×10^{-2} respectively, each gave a measure of the average squared error over 40 samples. Average squared real output error was used for the DNA-FIR digital filter structure. For the DNEEBA-IIR structures, average squared real output error was measured in the first column while the average squared equation error was measured in the second column. For the DNA-FIR structures, the convergence constant, μ , was set to 1.0×10^{-2} , 2.0×10^{-2} , and 5.0×10^{-3} for modelling $H_1(z^{-1})$, $H_2(z^{-1})$, and $H_3(z^{-1})$, respectively. For the DNEEBA-IIR structures, the values of μ_A and μ_{10} were set to 2.0×10^{-2} , 1.0×10^{-2} , 1.0×10^{-1} and 1.0×10^{-3} , and 1.0×10^{-2} and 1.0×10^{-3} for modelling $H_1(z^{-1})$, $H_2(z^{-1})$, and $H_3(z^{-1})$, respectively. It can be observed from Table 2.9 that the DNEEBA-IIR structures require fewer numbers of iterations than those of the corresponding DNA-FIR structures.

Table 2.9: Convergence performance in terms of number of iterations of 22nd-order DNA-FIR and 11th-order DNEEBA-IIR structures.

Plant	Stopping Criterion	No. of Paths	DNA-FIR	DNEEBA-IIR (real error)	DNEEBA-IIR (eqn. error)
$H_1(z^{-1})$	3.0×10^{-2}	2	4840	4160	1240
$H_2(z^{-1})$	5.0×10^{-2}	3	5109	4758	2106
$H_3(z^{-1})$	6.0×10^{-2}	4	27520	26280	4040

2.3.5 Conclusions

An IIR digital filter realization, called DNEEBA-IIR digital filter structure, has been formulated in which the problem of stability can be avoided. In the DNEEBA-IIR digital filter structure, a simple variable step adaptive algorithm has been developed to improve the speed of convergence. In this DNEEBA-IIR digital filter structure, $2N^2$ DSPs can be used with a maximum throughput rate of up to $2N^2$ times higher than that of the direct form IIR adaptive digital filter structure realized using one DSP. Simulation results have been given to show the performances of the proposed IIR digital filter structure for system modeling.

2.4 Conclusions

In this chapter, a family of linear delayed N-path adaptive digital filters have been presented. They are

- (i) Delayed N-path adaptive FIR (DNA-FIR) digital filter,
- (ii) Delayed N-path adaptive linear phase FIR (DNALP-FIR) digital filter,
- (ii) Delayed N-path equation-error based adaptive IIR (DNEEBA-IIR) digital filter.

By using N^2 DSPs in parallel, the maximum throughput rate of a DNA-FIR digital filter can be increased by N^2 times compared with the conventional adaptive FIR digital filter using one such DSP. Based on the DNA-FIR structure, the DNALP-FIR digital filter has been contrived in a similar manner. By exploiting the symmetry property of a

linear phase digital filter, the maximum throughput rate of a DNALP-FIR digital filter can be further improved by a factor of 2, compared with the DNA-FIR digital filter. The same technique can also be applied to an equation-error based adaptive IIR digital filter to obtain the DNEEBA-IIR digital filter structure. In a DNEEBA-IIR digital filter, each of the numerator and the denominator polynomials can be realized by two DNA-FIR digital filter structures respectively. By using $2N^2$ DSPs in parallel, the maximum throughput rate can be improved by $2N^2$ times compared with a conventional equation-error based adaptive IIR digital filter using one DSP. However, the convergence performance of this type of delayed N-path adaptive digital filters has been degraded by a factor of N, as will be discussed in detail in the following chapter. Even considering this fact, the speed gains of N, 2N, and $2N^2$ times can still be achieved for the DNA-FIR digital filter, the DNALP-FIR digital filter, and the DNEEBA-IIR digital filter, respectively.

Chapter 3

COMPARISONS BETWEEN BLOCK IMPLEMENTATION OF ADAPTIVE FIR DIGITAL FILTER AND DELAYED N-PATH ADAPTIVE FIR DIGITAL FILTER

Based on the conventional adaptive FIR digital filter first presented by Widrow [7], comparison studies have been conducted in the time domain between the block implementation of adaptive FIR digital filters and the delayed N-path adaptive FIR digital filters, in which both structures are realized by parallel digital signal processors. Both structures feature improvements in the system throughput rate compared with the conventional adaptive FIR digital filter, however, the speed gains have been degraded by a certain factor because of the slow convergence of the adaptive algorithm. Simulation results were performed to verify the conclusions obtained through theoretical analyses.

3.1 Introduction

Signal processing tasks vary considerably in nature and purpose. However, one common element is the need for performing a very large number of operations very quickly. One approach to achieve high speed digital signal processing is to increase the parallelism in the structure and algorithm and to employ many processing elements for

concurrent operations. This case is also true in the area of adaptive digital signal processing. Block implementation of adaptive FIR (BIA-FIR) digital filtering has been presented in [21] to increase the throughput rate. One direct way to realize the BIA-FIR digital filter is to use L parallel processors, where L is the input and output block length. In this structure, the throughput rate can be improved by a factor of L . The principle of the block adaptive digital filter is based on parallel processing in which each of the L parallel processors computes one output based on one block of L consecutive inputs.

In Chapter 2, a family of linear delayed N -path adaptive digital filters has been presented for high speed digital signal processing. Among them, the basic structure is the delayed N -path adaptive FIR (DNA-FIR) digital filter. In a DNA-FIR digital filter, the transfer function is decomposed into N paths and each path consists of N parallel sub-filters realized by N processors. Each of these N parallel processors in a path is allowed to operate at a sampling rate N times slower than that of the input signal; the number of coefficients to be updated in each of the N^2 processors is $1/N$ times that of the total number of the coefficients. Therefore, the maximum throughput rate of the DNA-FIR digital filter can be improved by N^2 times compared with the conventional adaptive FIR (CA-FIR) digital filter [21] using only one such processor.

In the context of adaptive digital signal processing, the convergence rate of the adaptive algorithm is another important fact which also determines the system's processing speed. The adaptive algorithms used in both the BIA-FIR and DNA-FIR

digital filters are the extended version of least-mean-square (LMS) algorithm. In this dissertation, we have shown by theoretical analyses and simulation results that the convergence rate of the BIA-FIR digital filter of block length L using parallel processors or DNA-FIR digital filter has degraded by a factor of L or N , respectively. Our comparison studies have also revealed that the performance index used in the BIA-FIR digital filter cause unnecessary computation when the input signal is stationary and the successive input vectors are independent over time.

3.2 Analyses and Comparisons between Block Implementation of Adaptive FIR (BIA-FIR) Digital Filter and Delayed N-Path Adaptive FIR (DNA-FIR) Digital Filter

In this section, comparison studies are performed analytically based on the conventional adaptive FIR (CA-FIR) digital filter. In the following, we first introduce the basic concepts on the CA-FIR digital filter developed in [21]. Then the BIA-FIR and DNA-FIR digital filters are analyzed and compared in terms of computation complexity, system throughput rate, convergence speed of adaptive algorithm, system convergence time, and the number of processors used.

3.2.1 CA-FIR Digital Filter

Shown in Fig. 3.1 is the diagram of a conventional adaptive FIR (CA-FIR) digital

filter. This CA-FIR digital filter is of order $M-1$, the output signal $y(k)$ of which at discrete time instant k is given as the convolution sum of the input $x(k)$ and the filter weights $h_k(r_i)$, i.e.,

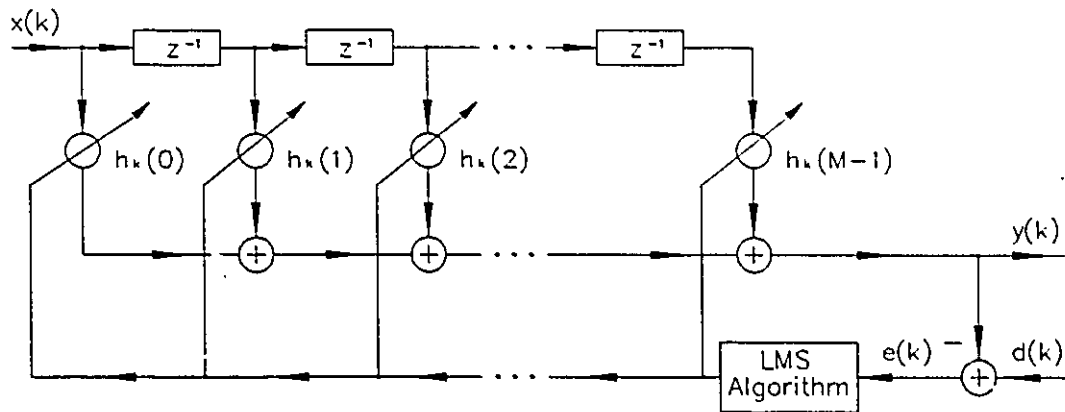


Fig. 3.1 Conventional adaptive FIR digital filter

$$y(k) = \sum_{m=0}^{M-1} h_k(m)x(k-m) \quad \text{for } k = 1, 2, 3 \dots \quad (3.1)$$

The performance index is the mean-square-error (MSE) Φ which is defined as

$$\Phi = E[e^2(k)] \quad (3.2)$$

where E is a mean value operator, and

$$e(k) = d(k) - y(k) \quad (3.3)$$

The LMS algorithm of Widrow-Hoff [7] adjusts the filter weights in accordance with

$$h_{k+1}(m) = h_k(m) + 2\mu e(k)x(k-m) \quad \text{for } m = 0, 1, \dots, M-1 \quad (3.4)$$

where μ is the convergence constant. Alternatively, (4) can be represented in matrix format:

$$H_{k+1} = H_k + 2\mu e(k)X_k \quad (3.5)$$

where H_k and X_k are the $M \times 1$ weight vector and the $M \times 1$ input vector, respectively, i.e.,

$$H_k = [h_k(0), h_k(1), \dots, h_k(M-1)]^T \quad (3.6)$$

$$X_k = [x(k), x(k-1), \dots, x(k-M+1)]^T \quad (3.7)$$

Assuming that successive input vectors are independent over time, hence H_k is independent of X_k . For the stationary input process meeting this condition, the expected value of the weight vector $E[H_k]$ after a sufficient number of iterations can be shown to converge to the Wiener optimal solution, i.e.,

$$H^* = R^{-1}P \quad (3.8)$$

where R is the input correlation matrix defined as

$$R = E[X_k X_k^T] = E \begin{bmatrix} x^2(k) & x(k)x(k-1) & \dots & x(k)x(k-M+1) \\ x(k-1)x(k) & x^2(k-1) & \dots & x(k-1)x(k-M+1) \\ \vdots & \vdots & \vdots & \vdots \\ x(k-M+1)x(k) & x(k-M+1)x(k-1) & \dots & x^2(k-M+1) \end{bmatrix} \quad (3.9)$$

and P is the vector of cross correlations between the desired response $d(k)$ and the input

components obtained as

$$P = E[d(k)X_k] = E[d(k)x(k), d(k)x(k-1), \dots, d(k)x(k-M+1)]^T \quad (3.10)$$

The time constant τ^p for p th mode of the difference equation describing the adaptation process has been shown as

$$\tau^p = \frac{1}{4\mu\sigma_p} \quad p = 1, 2, \dots, M \quad (3.11)$$

where σ_p is the p th eigenvalue of R . For the special case in which all eigenvalues of the input autocorrelation matrix R are equal, the M time constants can be lumped into one, i.e.,

$$\tau = \frac{M}{4\mu \text{tr}R} \quad (3.12)$$

where $\text{tr}R$ is the trace of R or the sum of the diagonal elements of R .

3.2.2 BIA-FIR Digital Filter

Block implementation of adaptive FIR (BIA-FIR) digital filters can be obtained by extending the conventional Widrow-Hoff LMS adaptive algorithm to the block implementation of digital filters, which allows efficient use of parallel processors and hence results in speed gains [21-22,58-60]. Shown in Fig. 3.2 is a diagram of a general configuration of BIA-FIR digital filters. S/P represents a serial-in and parallel-out register which breaks a continuous data stream into blocks before processing. P/S stands for a parallel-in and serial-out register which reassembles the resulting data blocks into a

continuous data stream after processing. The adaptive algorithm has been developed to allow a whole block of outputs to be calculated without modifying the filter parameters and adjusting filter weights once per block of data.

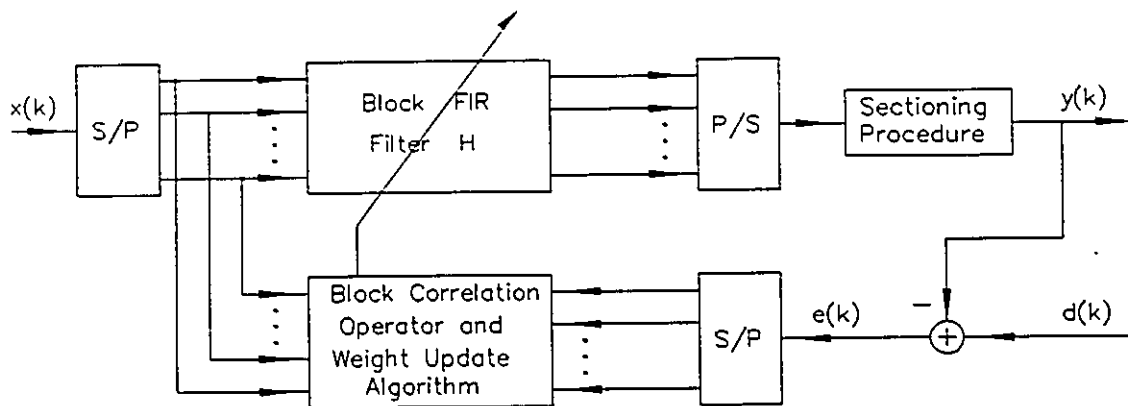


Fig. 3.2 Block implementation of adaptive FIR digital filter

As shown in Fig. 3.2, $x(k)$, $y(k)$, and $d(k)$ represent, respectively, the input, output, and desired signals. For the time-invariant case, eqn. 3.1 can be written in matrix format as

$$y(k) = H^T X_k = X_k^T H \quad (3.13)$$

where H is the same as eqn. 3.6 except for the drop of time index k . Assuming L represents block length and letting $L=3$ and $M=3$, eqn. 3.13 can be expressed in a convenient form as follows:

$$\begin{array}{c}
 U_1 \{ \\
 \\
 \\
 U_2 \{ \\
 \\
 \\
 \\
 \\
 \end{array}
 \left[\begin{array}{ccc}
 x(1) & 0 & 0 \\
 x(2) & x(1) & 0 \\
 x(3) & x(2) & x(1) \\
 \hline
 x(4) & x(3) & x(2) \\
 x(5) & x(4) & x(3) \\
 x(6) & x(5) & x(4) \\
 \hline
 x(7) & x(6) & x(5) \\
 \vdots & \vdots & \vdots
 \end{array} \right]
 \begin{array}{c}
 \\
 \\
 \\
 \left[\begin{array}{c}
 h(0) \\
 h(1) \\
 h(2)
 \end{array} \right] \\
 \\
 H \\
 \\
 \\
 \end{array}
 =
 \begin{array}{c}
 \\
 \\
 \\
 \\
 \\
 \\
 \\
 \left[\begin{array}{c}
 y(1) \\
 y(2) \\
 y(3) \\
 \hline
 y(4) \\
 y(5) \\
 y(6) \\
 \hline
 y(7) \\
 \vdots
 \end{array} \right]
 \end{array}
 \begin{array}{c}
 \\
 \\
 \\
 \} Y_1 \\
 \\
 \\
 \\
 \\
 \\
 \} Y_2 \\
 \\
 \\
 \\
 \\
 \\
 \\
 \end{array}
 \quad (3.14)$$

Actually, the format of eqn. 3.14 can be extended to the general L and M as

$$Y_{k'} = U_{k'} \cdot H \quad \text{for } k' = 1, 2, 3, \dots \quad (3.15)$$

where k' is the block index, $Y_{k'}$ and $U_{k'}$ are, respectively, the k' th output vector of length L and the LxM matrix of input vectors, i.e.,

$$Y_{k'} = [y((k'-1)L+1), y((k'-1)L+2), \dots, y(k'L)]^T \quad (3.16)$$

$$U_{k'} = [X_{(k'-1)L+1}, X_{(k'-1)L+2}, \dots, X_{k'L}]^T \quad (3.17)$$

Define the Lx1 vector of desired responses for block k' as

$$D_{k'} = [d((k'-1)L+1), d((k'-1)L+2), \dots, d(k'L)]^T \quad (3.18)$$

and the Lx1 vector of errors for block k' as

$$\varepsilon_{k'} = [e((k'-1)L+1), e((k'-1)L+2), \dots, e(k'L)]^T \quad (3.19)$$

The performance index is chosen to be as follows

$$\Phi_B = \frac{1}{L} E[\varepsilon_{k'}^T \cdot \varepsilon_{k'}] = E \left[\frac{1}{L} \sum_{k=(k'-1)L+1}^{k'L} e^2(k) \right] \quad (3.20)$$

It has been shown that the optimal set of filter weights H_B^* for the block Wiener filter is the same as for the conventional Wiener filter, i.e.,

$$H_B^* = R^{-1} P \quad (3.21)$$

During the adaptive process, the weight vector is adjusted once per data block rather than once per data sample. The algorithm then becomes

$$H_{k'+1} = H_{k'} - \mu_B \delta_{k'}^B \quad (3.22)$$

where μ_B is the convergence constant and $\delta_{k'}^B$ is the $M \times 1$ index gradient at block k' . The gradient is taken with respect to the weights as follows

$$\delta_{k'}^B = \frac{1}{L} \left. \frac{\partial E[\varepsilon_{k'}^T \cdot \varepsilon_{k'}]}{\partial H} \right|_{H=H_{k'}} \quad (3.23)$$

Because the computation of an ensemble average is difficult, $\varepsilon_{k'}^T \varepsilon_{k'}$ is used to estimate $E[\varepsilon_{k'}^T \varepsilon_{k'}]$ in eqn. 3.23. Then eqn. 3.23 becomes

$$\delta_{k'}^B \approx \frac{1}{L} \left. \frac{\partial (\varepsilon_{k'}^T \cdot \varepsilon_{k'})}{\partial H} \right|_{H=H_{k'}} \quad (3.24)$$

Use of this unbiased block gradient estimate in the weight adjustment equation eqn. 3.22

gives the block least-mean-square (BLMS) algorithm

$$\begin{aligned} H_{k+1} &= H_k + \frac{2\mu_B}{L} U_k^T \varepsilon_k \\ &= H_k + \frac{2\mu_B}{L} \sum_{k=(k'-1)L+1}^{k'L} e(k) X_k \end{aligned} \quad (3.25)$$

Adaptation speed is given in terms of time constants which indicate how fast the weight vector converges to the optimal weight vector [61]. The time constant τ_p^B for pth mode has been shown as

$$\tau_p^B = \frac{L}{4\mu_B \sigma_p} \quad p = 1, 2, \dots, M \quad (3.26)$$

and for the special case in which all eigenvalues of the input autocorrelation matrix R are equal, the N time constants are the same, i.e.,

$$\tau^B = \frac{ML}{4\mu_B \text{tr}R} \quad (3.27)$$

where $\text{tr}R$ is the trace of R or the sum of the diagonal elements of R .

3.2.3 DNA-FIR Digital Filter

3.2.3.1 Structure of DNA-FIR Digital Filter

In Section 2.1 of Chapter 2, the derivation of the structure is started from the transfer function concerned with the input-output relationship of eqn. 3.1, i.e.,

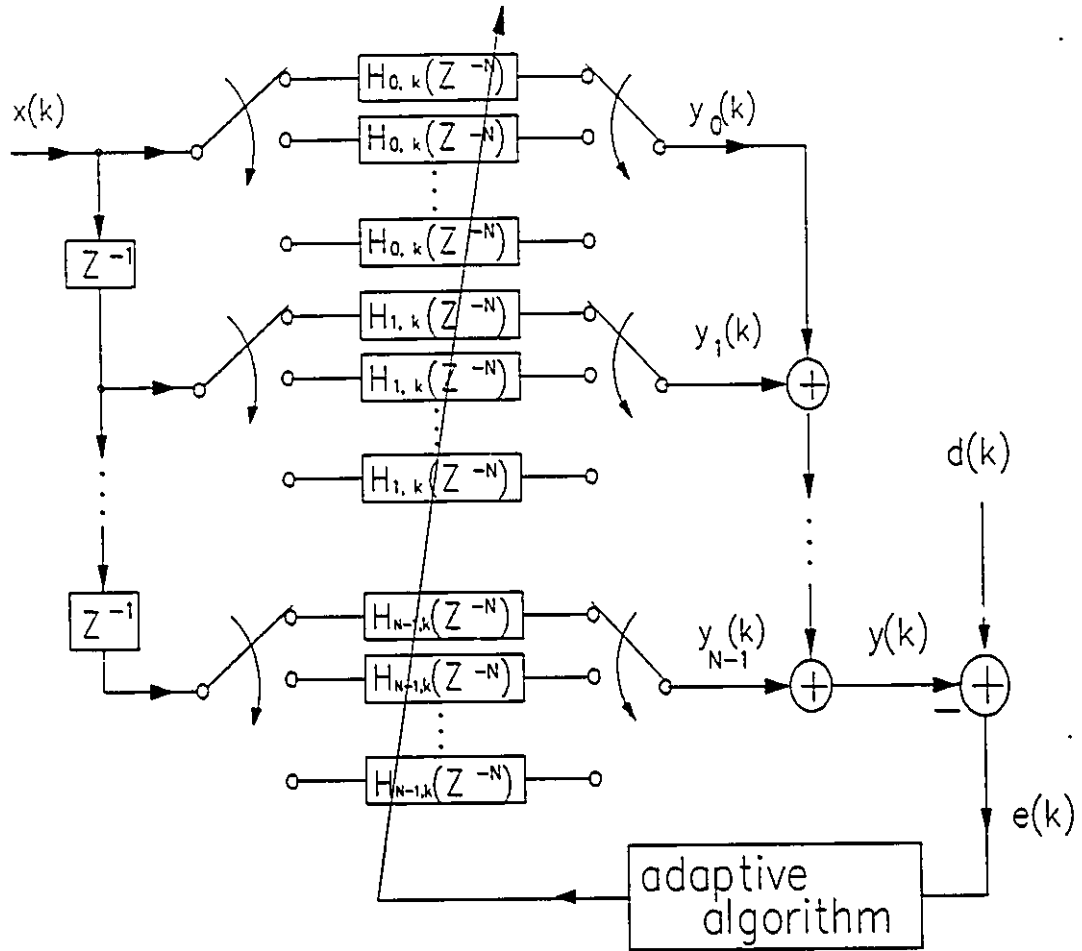


Fig. 3.3 Delayed N-path adaptive FIR digital filter

$$H_k(z^{-1}) = \sum_{m=0}^{M-1} h_k(m)z^{-m} \quad (3.28)$$

Eqn. 3.28 can be decomposed into the following format which is suitable for delayed N-path realization, i.e.,

$$H_k(z^{-1}) = \sum_{i=0}^{N-1} H_{i,k}(z^{-N})z^{-i} \quad (3.29)$$

where

$$H_{i,k}(z^{-N}) = \sum_{l=0}^{L_i} h_k(lN+i)z^{-lN} \quad (3.30)$$

and

$$L_i = [M/N]^+ \quad \text{for } i = 0, 1, \dots, J-1 \quad (3.31a)$$

$$L_i = [M/N]^+ - 1 \quad \text{for } i = J, J+1, \dots, N-1 \quad (3.31b)$$

$$J = (M)_N \quad (3.31c)$$

where $[M/N]^+$ represents the least integer greater than or equal to M/N and $(M)_N$ means M mode N . Fig. 3.3 shows the structure of N parallel paths represented by eqns. 3.29 and 3.30. Denote the input signal, $x(k)$, and the output signal, $y(k)$, respectively, we have the following input-output relationships

$$y_i(k) = \sum_{l=0}^{L_i} h_k(lN+i)x(k-lN-i) \quad (3.32)$$

$$y(k) = \sum_{i=0}^{N-1} y_i(k) \quad (3.33)$$

where $y_i(k)$ is the output of the i th path as shown in Fig. 3.3.

In Fig. 3.3, two types of switches are used to connect the input signal to the parallel sub-filters and the output of each parallel sub-filter to the overall output signal. Each path has two switches of the two different types and each switch has N different positions to connect the N sub-filters within the same path. The switches change their positions from the first sub-filter to the last sub-filter within the same path sequentially at the sampling rate of the input signal, and repeats this procedure all the time during the adaptive process. In fact, the coefficients of each sub-filter within the same path are to be changed individually at different time instants. They may have different sets of intermediate values during the adaptive process even though they finally will arrive at the same set of values after convergence of the adaptive process. In order to carry on the following analyses, we add one extra subscript j to each sub-filter transfer function $H_{i,k}(z^N)$. As can be proved later, each set of coefficients within the same path has the same theoretical optimal values.

Rewriting eqn. 3.30, the transfer function of each sub-filter can now be expressed as

$$H_{i,j,k}(z^{-N}) = \sum_{l=0}^{L_i} h_{j,k}(lN+i)z^{-lN} \quad (3.34)$$

where i and j , from 0 to $N-1$, represent the path number and the number of sub-filter in each path, respectively.

Defining

$$y(k_j) = \begin{cases} y(k) & \text{when } (k)_N = j \\ 0 & \text{otherwise} \end{cases} \quad (3.35)$$

for $j = 0, 1, \dots, N-1$, then we have

$$y(k) = \sum_{j=0}^{N-1} \delta((k)_N - j) y(k_j) \quad (3.36)$$

and

$$\begin{aligned} y(k_j) &= \sum_{i=0}^{N-1} \sum_{l=0}^{L_i} h_{j,k}(lN+i) x(k-lN-i) \\ &= \sum_{m=0}^{M-1} h_{j,k}(m) x(k-m) \end{aligned} \quad (3.37)$$

where $\delta((k)_N - j)$ in eqn. 3.36 is a unit impulse function with argument $(k)_N - j$. Substituting eqn. 3.37 into eqn. 3.36, we get

$$\begin{aligned} y(k) &= \sum_{j=0}^{N-1} \delta((k)_N - j) \sum_{m=0}^{M-1} h_{j,k}(m) x(k-m) \\ &= \sum_{j=0}^{N-1} \delta((k)_N - j) X_k^T H_{j,k} \end{aligned} \quad (3.38)$$

where X_k is defined by eqn. 3.7 and

$$H_{j,k} = [h_{j,k}(0), h_{j,k}(1), \dots, h_{j,k}(M-1)]^T \quad (3.39)$$

for $j = 0, 1, \dots, N-1$. Let $d(k)$ represent the desired output at discrete time k , the output error $e(k)$ is

$$\begin{aligned} e(k) &= d(k) - y(k) \\ &= d(k) - \sum_{j=0}^{N-1} \delta((k)_N - j) X_k^T H_{j,k} \end{aligned} \quad (3.40)$$

3.2.3.2 Adaptive Algorithm of DNA-FIR Digital Filter

The performance index is defined as

$$\Phi_D = E[e^2(k)] \quad (3.41)$$

The aim of the adaptive algorithm is to search the optimal sets of weight values of H_j , for $j = 0, 1, \dots, N-1$, to minimize the performance index. Arguing along the same lines, we use $e^2(k)$ to estimate $E[e^2(k)]$ in deriving the adaptive algorithm. Because every switch is changing its position at every sample interval, the set of weight values of each sub-filter is updated every N sample intervals as

$$H_{j,k+N} = H_{j,k} - \mu_D \delta_k^D \quad \text{when } (k)_N = j \quad (3.42)$$

where μ_D is the convergence constant and δ_k^D is the estimated $M \times 1$ gradient vector of performance index at discrete time k , i.e.,

$$\delta^D_k \approx \frac{\partial e^2(k)}{\partial H_j} \quad \text{for } j = 0, 1, \dots, N-1 \quad (3.43)$$

Here we drop the time subscript k for weight vectors, since they are assumed constant vector when performing partial derivatives. Substituting eqn. 3.40 into eqn. 3.43, we obtain

$$\begin{aligned} \delta^D_k &= 2e(k) \left[\frac{\partial e(k)}{\partial h_j(0)}, \frac{\partial e(k)}{\partial h_j(1)}, \dots, \frac{\partial e(k)}{\partial h_j(M-1)} \right] \\ &= -2e(k)X_k \end{aligned} \quad (3.44)$$

Then eqn. 3.42 becomes

$$H_{j, k+N} = H_{j, k} + 2\mu_D e(k)X_k \quad \text{when } (k)_N = j \quad (3.45)$$

for $j = 0, 1, \dots, N-1$.

3.2.3.3 Optimal Weight Vector

It can be proved in the following that each weight vector $H_{j,k}$, for $j = 0, 1, \dots, N-1$, in different sub-filters will arrive at the same optimal weight vector after convergence, which is also the same as that obtained by using a CA-FIR digital filter, provided the same input and desired signals are presented.

Squaring both sides of eqn. 3.40, we get

$$\begin{aligned}
e^2(k) &= d^2(k) + \sum_{j_1=0}^{N-1} \sum_{j_2=0}^{N-1} \delta((k)_{N-j_1}) \delta((k)_{N-j_2}) H_{j_1}^T X_k X_k^T H_{j_2} \\
&\quad - 2d(k) \sum_{j=0}^{N-1} \delta((k)_{N-j}) X_k^T H_j \\
&= d^2(k) + \sum_{j=0}^{N-1} \delta((k)_{N-j}) H_j^T X_k X_k^T H_j - 2d(k) \sum_{j=0}^{N-1} \delta((k)_{N-j}) X_k^T H_j \\
&= d^2(k) + \sum_{j=0}^{N-1} \delta((k)_{N-j}) (H_j^T X_k X_k^T H_j - 2d(k) X_k^T H_j)
\end{aligned} \tag{3.46}$$

In eqn. 3.46, we drop the subscript k of $H_{j,k}$ in eqn. 3.40 for the same reason mentioned above. The performance index Φ_D now becomes

$$\begin{aligned}
\Phi_D &= E[e^2(k)] \\
&= E[d^2(k)] + \sum_{j=0}^{N-1} \delta((k)_{N-j}) (H_j^T E[X_k X_k^T] H_j - 2E[d(k) X_k^T] H_j) \\
&= E[d^2(k)] + \sum_{j=0}^{N-1} \delta((k)_{N-j}) (H_j^T R H_j - 2P^T H_j)
\end{aligned} \tag{3.47}$$

where R and P are defined in eqn. 3.9 and eqn. 3.10. The real $M \times 1$ gradient vector δ^D_k of the performance index can be obtained by differentiating eqn. 3.47 with respect to weight vector H_j as follows

$$\delta^D_k = \frac{\partial \Phi_D}{\partial H_j} = 2R H_j - 2P \quad \text{for } j = 0, 1, \dots, N-1 \tag{3.48}$$

To obtain the minimum mean square error, the weight vector H_j , for $j = 0, 1, \dots, N-1$, is set at its optimal value H_j^* , where the gradient is zero

$$\delta_k^D = \mathbf{0} = 2RH_j^* - 2P \quad (3.49)$$

then, the optimum weight vector is

$$H_j^* = R^{-1}P \quad \text{for } j = 0, 1, \dots, N-1 \quad (3.50)$$

which is the same as the Wiener optimal solution in eqn. 3.8.

3.2.3.4 Time Constants

In the following, we will find out time constants for the DNA-FIR digital filter.

From eqn. 3.45, it follows that

$$E[H_{j, k+N}] = E[H_{j, k}] + 2\mu_D E[e(k)X_k] \quad (3.51)$$

Substituting eqn. 3.40 and eqn. 3.38 into eqn. 3.51 we have

$$\begin{aligned} E[H_{j, k+N}] &= E[H_{j, k}] + 2\mu_D (E[d(k)X_k] - E[X_k y(k)]) \\ &= E[H_{j, k}] + 2\mu_D (E[d(k)X_k] - E[X_k \sum_{n=0}^{N-1} \delta((k)_N - n) X_k^T H_{n, k}]) \\ &= E[H_{j, k}] + 2\mu_D (E[d(k)X_k] - \sum_{n=0}^{N-1} \delta((k)_N - n) E[X_k X_k^T H_{n, k}]) \end{aligned} \quad (3.52)$$

Using the foregoing assumption that X_k and $H_{j, k}$ are independent, it follows from eqn.

3.52 that

$$E[H_{j, k+N}] = E[H_{j, k}] + 2\mu_D (P - R \sum_{n=0}^{N-1} \delta((k)_N - n) E[H_{n, k}]) \quad (3.53)$$

Because the subscript k in eqn. 3.53 must satisfy the relationship

$$(k)_N = j \quad (3.54)$$

hence, eqn. 3.53 can be reduced to

$$\begin{aligned} E[H_{j, k+N}] &= E[H_{j, k}] + 2\mu_D(P - RE[H_{j, k}]) \\ &= (I - 2\mu_D R)E[H_{j, k}] + 2\mu_D RH_j^* \end{aligned} \quad (3.55)$$

Let

$$V_{j, k} = H_{j, k} - H_j^* \quad \text{for } j = 0, 1, \dots, N-1 \quad (3.56)$$

By subtracting H_j^* from both sides of eqn. 3.55, we have

$$E[V_{j, k+N}] = (I - 2\mu_D R)E[V_{j, k}] \quad (3.57)$$

By the similar derivations as in the case of CA-FIR digital filters, we arrive at

$$E[V'_{j, k+N}] = (I - 2\mu_D \Gamma)E[V'_{j, k}] \quad (3.58)$$

where

$$V'_{j, k+N} = Q^{-1}V_{j, k+N} \quad (3.59)$$

Γ is the diagonal eigenvalue matrix, in which all the elements are zero except for the main diagonal, whose elements are the set of eigenvalues of R , i.e., $\sigma_1, \sigma_2, \dots, \sigma_M$. Q is called the eigenvector matrix of R , because its columns are the eigenvectors of R . Both Γ and Q are square with dimensions $M \times M$, like R .

By reasoning inductively, the solution to eqn. 3.58 must be

$$\begin{aligned}
E[V_{j, k+N}] &= (I - 2\mu_D\Gamma)^{(1+(k-j)/N)} E[V_{j, j}] \\
&= (I - 2\mu_D\Gamma)^{k/N} (I - 2\mu_D\Gamma)^{(1-j/N)} E[V_{j, j}] \\
&\text{for } j = 0, 1, \dots, N-1
\end{aligned} \tag{3.60}$$

The individual weight relaxation time t_p , for $p = 1, 2, \dots, M$, can be obtained as follows

$$e^{-1/t_p} = (1 - 2\mu_D\sigma_p)^{1/N} = r_p^{1/N} \tag{3.61}$$

By mathematical approximation provided that t_p is large and r_p is small, as was made in [7] for the derivation of time constants of the CA-FIR digital filter, we have

$$t_p \approx \frac{N}{2\mu_D\sigma_p} \tag{3.62}$$

Therefore, the corresponding time constants τ_p^D , for $p = 1, 2, \dots, M$, are

$$\tau_p^D = \frac{t_p}{2} = \frac{N}{4\mu_D\sigma_p} \tag{3.63}$$

If all the eigenvalues σ_p for $p = 1, 2, \dots, M$ are of the same value, then we get only one time constant τ^D as

$$\tau^D = \frac{MN}{4\mu_D\sigma_p} \tag{3.64}$$

3.2.4 Comparisons between BIA-FIR and DNA-FIR Digital Filters

3.2.4.1 Computation Complexity

In the following discussions, we will define the computation complexity of

multiplications while ignoring additions for simplicity. This is reasonable because the number of additions is approximately the same as the number of multiplications and an adding operation usually takes less time than a multiplying operation.

Define the computation complexity as the number of multiplications performed per data sample by one processor, i.e.,

$$CC = \frac{\text{Multiplications}}{\text{Per data sample} \cdot \text{Per Processor}} \quad (3.65)$$

This definition can be justified in that, if CC is bigger, then the computation load borne by each processor is heavier, and the maximum throughput rate of the system will become lower. Eqn. 3.65 can actually be decomposed into two parts if each processor is assumed to finish both filtering task and adaptive coefficient adjustments, i.e.,

$$CC = CC_1 + CC_2 \quad (3.66)$$

where CC_1 and CC_2 correspond to the multiplications performed for filtering and adaptive algorithm, respectively.

The output of the BIA-FIR digital filter is computed once for every L samples in a vector format as expressed in eqn. 3.15 which represents a set of linear convolutions and can actually be implemented using parallel processors in time domain. For a block length of L, one simplest and direct way of doing this is to use L parallel processors to compute each element of the output vector Y_k . Then each processor is responsible for M multiplications at every L sample intervals, i.e.,

$$CC^{B_1} = \frac{M}{L} \quad (3.67)$$

We can see from eqn. 3.25 that the weight vector is adjusted once per data block and L multiplications are needed for each element of the weight vector. A total of $L \cdot M$ multiplications would be performed by L processors for every L input samples. Then we have

$$CC^{B_1} = \frac{L \cdot M}{L \cdot L} = \frac{M}{L} \quad (3.68)$$

It follows from eqn. 3.67 and eqn. 3.68 that

$$CC^B = CC^{B_1} + CC^{B_1} = \frac{2M}{L} \quad (3.69)$$

As for the case of DNA-FIR digital filters, each sub-filter block in Fig. 3.3 can readily be implemented by one processor for both filtering task and adaptive coefficient adjustments and a total of N^2 processors would be used in parallel for this structure. Arguing along the same lines as the case of BIA-FIR digital filters, the computation complexity for DNA-FIR digital filters can also be split into two parts as

$$CC^D = CC^{D_1} + CC^{D_2} \quad (3.70)$$

where CC^{D_1} and CC^{D_2} correspond to the multiplications performed for filtering and adaptive algorithm, respectively. Actually, eqn. 3.32 represents the filtering task borne by each processor in every N sample intervals, where L_i+1 , $i = 0, 1, \dots, N-1$,

multiplications are involved. For clear comparisons, we use M/N to approximate L_i+1 in the following calculation of CC^D , as can be justified by eqn. 3.31a and 3.31b. It follows, therefore, that

$$CC^{D_1} = \frac{M/N}{N} = \frac{M}{N^2} \quad (3.71)$$

From eqn. 3.45 we can observe that only one multiplication is necessary to update one coefficient, and hence M multiplications would be performed by N processors for every N input samples, i.e.,

$$CC^{D_2} = \frac{M}{N \cdot N} = \frac{M}{N^2} \quad (3.72)$$

Hence, we have from eqn. 3.71 and 3.72 that

$$CC^D = CC^{D_1} + CC^{D_2} = \frac{2M}{N^2} \quad (3.73)$$

Comparing eqn. 3.73 with eqn. 3.69 we can see that the computation complexity of the DNA-FIR digital filter is N times less than that of the BIA-FIR digital filter when $L=N$, which implies that the maximum throughput rate of DNA-FIR digital filters could be N times higher than that of the BIA-FIR digital filter.

3.2.4.2 System Throughput Rate

Owing to the advances of VLSI technology, more and more adaptive digital systems are realized by digital signal processors (DSPs) instead of discrete digital

elements or general purpose microprocessors. Although there exist some high speed DSPs, upper limits always exist on their throughput rates due to the limitation of manufacturing technology. On the other hand, the maximum throughput rate of a given processor is also determined by the number of instruction steps in a given signal processing program, and hence, it is directly associated with the complexity of the algorithm, or the computation complexity we discussed above. In the following, the comparison of system throughput rates (STR) between the BIA-FIR and DNA-FIR adaptive systems is carried out under the assumption that the same type of DSPs are used in both systems, and hence it is determined by adaptive algorithms and system structures.

Define the system throughput rate as

$$STR = \frac{\text{Number of input data samples}}{\text{Seconds needed to process the input data samples}} \quad (3.74)$$

Assuming t_m seconds are needed for each DSP to finish a multiplying and an adding operations. Then, for every input data sample, each DSP needs $CC \cdot t_m$ seconds to finish the multiplications. Hence, we have

$$STR = \frac{1}{CC \cdot t_m} \quad (3.75)$$

Substituting eqn. 3.69 into eqn. 3.75, we have STR^b for the BIA-FIR adaptive digital filter as

$$STR^B = \frac{L}{2M \cdot t_m} \quad (3.76)$$

and STR^D for the DNA-FIR adaptive digital filter as

$$STR^D = \frac{N^2}{2M \cdot t_m} \quad (3.77)$$

We can see from eqn. 3.76 and eqn. 3.77 that the throughput rate of the DNA-FIR adaptive system is N times higher than that of the BIA-FIR adaptive system when N and L are set to the same value.

3.2.4.3 Convergence Speed of Adaptive Algorithm

Convergence speed of adaptive algorithm is measured by time constants which indicate how fast the weight vector converges to the optimal weight vector. In other words, time constants reflect proportionally the number of data samples needed for the adaptive process to converge. For an easy reference, we rewrite eqns. 3.26, 3.63, 3.27 and 3.64 in the following

$$\tau_p^B = \frac{L}{4\mu_B\sigma_p} \quad \text{for } p = 1, 2, \dots, M \quad (3.78a)$$

$$\tau_p^D = \frac{N}{4\mu_D\sigma_p} \quad \text{for } p = 1, 2, \dots, M \quad (3.78b)$$

$$\tau^B = \frac{ML}{4\mu_B J R} \quad (3.79a)$$

$$\tau^D = \frac{MN}{4\mu_D trR} \quad (3.79b)$$

If the block length L of the BIA-FIR digital filter equals the path number N of the DNA-FIR digital filter, then τ_p^B would equal τ_p^D , when μ_B and μ_D are selected the same, as is also true for τ^B and τ^D . From this point of view, we can see that both adaptive algorithms are equivalently efficient in terms of adaptation speed.

3.2.4.4 System Convergence Time

In the previous three sub-sections, we discussed three different measurements of the adaptive systems, i.e., computation complexity, system throughput rate, and convergence speed of adaptive algorithm. For an adaptive system, it is important to preserve a high system throughput rate as well as to obtain an efficient adaptive algorithm. An efficient adaptive algorithm would need fewer input data samples for the adaptive process to converge. Hence, we introduce a more comprehensive concept to measure the adaptive system, i.e., system convergence time (SCT) which is the time needed to finish an adaptive process. It is defined as

$$SCT = \frac{\text{Number of input data samples needed to converge}}{STR} \quad (3.80)$$

where STR is the system throughput rate defined in eqn. 3.74. The numerator item in eqn. 3.80 can be approximated by $q\tau$ where τ is the time constant assuming all the eigenvalues of R are the same. q is a certain constant, the value of which would depend on different requirements on adaptation accuracy for different practical situations.

Substituting eqns 3.79a, 3.76 and eqns. 3.79b, 3.77 into eqn. 3.80 respectively, we get SCT^B and SCT^D for both systems as follows

$$SCT^B = \frac{q\tau^B}{STR^B} = \frac{qt_m M^2}{2\mu_B IrR} \quad (3.81)$$

$$SCT^D = \frac{q\tau^D}{STR^D} = \frac{qt_m M^2}{2\mu_D NtrR} \quad (3.82)$$

If $\mu_B = \mu_D$, then

$$SCT^B = N \cdot SCT^D \quad (3.83)$$

which means the DNA-FIR system is N times faster than the BIA-FIR system. It can also be observed from eqn. 3.81 and eqn. 3.82 that increasing the block length L would not affect the system convergence time SCT^B of the BIA-FIR system, while increasing the path number N does reduce the system convergence time SCT^D of the DNA-FIR system.

3.2.4.5 Performance Index

The performance index of the BIA-FIR digital filter is defined in eqn. 3.20, which is a combination of the standard MSE and the sum square error. Eqn. 3.41 gives the performance index of the DNA-FIR digital filter which is defined as the standard MSE. Despite different performance index definitions and different filter structures, however, they both arrive at the same optimal weight vector which is also the optimal solution to the conventional Wiener adaptive digital filter. In this regard, both BIA-FIR digital filtering and DNA-FIR digital filtering can be considered as two different approaches to

the conventional adaptive filtering problem which have significantly improved maximum system throughput rate. On the other hand, the different definitions on performance indexes of the BIA-FIR digital filter and the DNA-FIR digital filter did cause differences on computation complexities, which can be observed from eqn. 3.69 and eqn. 3.73. As can be proved in the following that eqn. 3.25 is equivalent to eqn. 3.45 statistically under the assumptions made above, i.e., the input signal is stationary and the successive input vectors are independent of each other.

Taking the expectation values of both sides of eqn. 3.25, we have

$$E[H_{k+1}] = E[H_k] + \frac{2\mu_B}{L} \sum_{k=(k-1)L+1}^{k'L} E[e(k)X_k] \quad (3.84)$$

Substituting eqn. 3.3 into eqn. 3.84, we get

$$\begin{aligned} E[H_{k+1}] &= E[H_k] + \frac{2\mu_B}{L} \sum_{k=(k-1)L+1}^{k'L} (E[d(k)X_k] - E[X_k X_k^T] E[H_k]) \\ &= E[H_k] + \frac{2\mu_B}{L} \sum_{k=(k-1)L+1}^{k'L} (P - RE[H_k]) \\ &= E[H_k] + 2\mu_B(P - RE[H_k]) \end{aligned} \quad (3.85)$$

for $k' = 1, 2, \dots$. Similarly, we have the following relationship by taking the expectation values of both sides of eqn. 3.45

$$E[H_{j, k+N}] = E[H_{j, k}] + 2\mu_D E[e(k)X_k] \quad (3.86)$$

Substituting eqn. 3.3 into eqn. 3.86, it follows that

$$\begin{aligned}
E[H_{j, k+N}] &= E[H_{j, k}] + 2\mu_D(E[d(k)X_k] - E[X_k X_k^T]E[H_{j, k}]) \\
&= E[H_{j, k}] + 2\mu_D(P - RE[H_{j, k}])
\end{aligned} \tag{3.87}$$

for $j = 0, 1, \dots, N-1$ and $k = 0, 1, 2, \dots$

We can observe that both eqn. 3.85 and eqn. 3.87 feature the exact same recursive relationship when $\mu_B = \mu_D$. Viewing from the statistical point under the above mentioned conditions, the summation in eqn. 3.25 introduces extra multiplications and additions, and hence increases the computation complexity and reduces system throughput rate.

3.2.4.6 Summary

It is obvious that both the BIA-FIR and DNA-FIR digital filters are reduced to the CA-FIR digital filter when L or N is set to 1. For an easy review, the above comparison results are summarized in Table 3.1, where the items of the CA-FIR digital filter are obtained by simply setting L or N to 1 in the items of the BIA-FIR or DNA-FIR digital filters.

Table 3.1 Analytical Comparison Results

Compared Items	CA-FIR	BIA-FIR	DNA-FIR
Performance Index	$E[e^2(k)]$	$E[\varepsilon_k^T \varepsilon_k]/L$	$E[e^2(k)]$
Optimal Weights	$R^{-1}P$	$R^{-1}P$	$R^{-1}P$
Time Constants	$1/(4\mu\sigma_P)$	$L/(4\mu_B\sigma_P)$	$N/(4\mu_D\sigma_P)$
Computation Complexity	$2M$	$2M/L$	$2M/N^2$
System Throughput Rate	$1/(2Mt_m)$	$L/(2Mt_m)$	$N^2/(2Mt_m)$
System Convergence Time	$qt_m M^2 / (2\mu \text{tr}R)$	$qt_m M^2 / (2\mu_B \text{tr}R)$	$qt_m M^2 / (2\mu_D N \text{tr}R)$
No. of Processors Used	1	L	N^2

3.3 Simulation Results

Two sets of simulation examples have been performed to verify the theoretical analyses obtained in Section 3.2.

Example 1: The first example problem is chosen from [21] which is very simple and features a good standard for comparisons. It is a system identification problem in which the goal is to model a plant consisting of a fixed delay of two sample periods. Its transfer function is

$$H(z^{-1}) = z^{-2}$$

Both the BIA-FIR and DNA-FIR adaptive digital filters were used to model this plant respectively. Both systems were fed with the same input signal $x(k)$ which was a software generated random numbers as an approximation to a white noise sequence with 0 mean value and 1/12 variance. The convergence constants μ_B and μ_D were chosen with the same value of 0.1 through all the simulations, and L and N were always selected with the same values for comparisons. Besides, the two systems had the same stopping criterion 1.0×10^{-10} , which was the average squared error over 100 samples. All the initial coefficients for both cases were set to 0.

Firstly, we set $L=N=1$. Then both cases were reduced to the CA-FIR digital filtering. They all converged after being fed with 700 input samples. For both cases, the finally obtained filter coefficients were

$$h(0), h_0(0) = -0.0000006943$$

$$h(1), h_0(1) = -0.0000047278$$

$$h(2), h_0(2) = 0.9999929278$$

Secondly, we chose $L=N=2$. The BIA-FIR digital filter converged after processing 1300 input data samples and the finally obtained coefficients were

$$h(0) = 0.0000070330$$

$$h(1) = -0.0000038548$$

$$h(2) = 0.9999862601$$

The DNA-FIR digital filter converged after processing 1400 input data samples and the

filter coefficients obtained after convergence were

Path 0:

$$h_0(0) = 0.0000018187$$

$$h_0(1) = -0.0000023574$$

$$h_0(2) = 0.9999956937$$

Path 1:

$$h_1(0) = 0.0000030431$$

$$h_1(1) = -0.0000021115$$

$$h_1(2) = 0.9999902541$$

Example 2: The second example is to model a 9th-order FIR digital filter of transfer function as follows

$$H(z^{-1}) = 2.3 - z^{-1} + 0.3z^{-2} + 1.1z^{-3} - z^{-4} - 0.1z^{-5} + 0.2z^{-6} + 3.1z^{-7} + 0.4z^{-8} - 1.2z^{-9}$$

The input signal $x(k)$, convergence constants μ_B and μ_D , and the stopping criterion were the same as used in the first example. All the initial coefficients for both cases were set to 0.

Firstly, we set $L=N=1$. Then both cases were reduced to the CA-FIR digital filtering. They all converged after being fed with 900 input samples. For both cases, the finally obtained filter coefficients were

$$h(0), h_0(0) = 2.3000004151$$

$$h(1), h_0(1) = -1.0000001761$$

$$h(2), h_0(2) = 0.3000000567$$

$$h(3), h_0(3) = 1.1000000708$$

$$h(4), h_0(4) = -0.9999991841$$

$$h(5), h_0(5) = -0.0999989791$$

$$h(6), h_0(6) = 0.1999998496$$

$$h(7), h_0(7) = 3.0999990559$$

$$h(8), h_0(8) = 0.3999988030$$

$$h(9), h_0(9) = -1.20000094748$$

Secondly, we chose $L=N=2$. The BIA-FIR digital filter converged after processing 1500 input data samples and the finally obtained coefficients were

$$h(0) = 2.2999906775$$

$$h(1) = -0.9999995534$$

$$h(2) = 0.3000012874$$

$$h(3) = 1.1000048088$$

$$h(4) = -0.9999926427$$

$$h(5) = -0.1000004838$$

$$h(6) = 0.1999959822$$

$$h(7) = 3.0999914706$$

$$h(8) = 0.3999966633$$

$$h(9) = -1.2000001598$$

The DNA-FIR digital filter converged after processing 1600 input data samples and the filter coefficients obtained after convergence were

Path 0:

$$h_0(0) = 2.2999959075$$

$$h_0(1) = -0.9999961041$$

Path 1:

$$h_1(0) = 2.3000114172$$

$$h_1(1) = -1.0000045553$$

$h_0(2) = 0.2999999949$	$h_1(2) = 0.2999917603$
$h_0(3) = 1.0999963920$	$h_1(3) = 1.1000025703$
$h_0(4) = -0.9999982067$	$h_1(4) = -0.9999887589$
$h_0(5) = -0.0999949760$	$h_1(5) = -0.1000048193$
$h_0(6) = 0.1999987037$	$h_1(6) = 0.1999920427$
$h_0(7) = 3.0999932433$	$h_1(7) = 3.0999986078$
$h_0(8) = 0.3999974457$	$h_1(8) = 0.3999974457$
$h_0(9) = -1.1999920597$	$h_1(9) = -1.1999920597$

We tried the cases for $L=N=5$, $L=N=10$, $L=N=20$, $L=N=50$ and $L=N=100$ for the above two examples as well. The major results are summarized in Table 3.2 where NOS stands for the number of input samples. Each entry in Table 3.2 is an average value obtained in 100 trials with different input signals generated with different seeds.

As can be observed from Table 3.2 that the BIA-FIR digital filters converged slightly faster than the corresponding DNA-FIR digital filters. This is because the input signals $x(k)$ used in the simulations were software generated pseudorandom sequences which were not strict stationary white noise processes. Hence the condition for holding eqn. 3.25 and eqn. 3.47 equivalent is not fully satisfied. The number of iterations needed for the structures of block length L or path number N were a little less than L or N times that needed by the CA-FIR digital filters. The reason for that also stemmed from the

input signals $x(k)$ and $d(k)$, which were computer generated pseudorandom sequences.

Table 3.2 Simulation Comparison Results

L or N	Example 1 (NOS)		Example 2 (NOS)	
	BIA-FIR	DNA-FIR	BIA-FIR	DNA-FIR
1	721	721	837	837
2	1358	1356	1596	1610
5	3172	3223	3797	3883
10	6228	6346	7268	7699
20	12330	12584	14215	15193
50	30621	31206	35061	37703
100	61052	62170	69819	74949

3.4. Conclusions

The BIA-FIR digital filter and the DNA-FIR digital filter have been developed to increase the processing speed of adaptive digital filtering. One direct way to realize the BIA-FIR digital filter is to use L parallel DSPs with a block length of L . For a DNA-FIR digital filter, N^2 DSPs are used to obtain concurrent operations. In this chapter, comparison studies have been conducted in time domain between those two structures. Based on the concepts developed on the CA-FIR digital filter, theoretical analyses have been given for the DNA-FIR digital filter, and the two digital filters have been compared

based on the theoretical results obtained. It has been shown that both the BIA-FIR and DNA-FIR digital filters arrive at the same optimal weight vector, which is also the same as that of the CA-FIR digital filter. By using L or N^2 DSPs in parallel, the maximum system throughput rate of the BIA-FIR digital filter or the DNA-FIR digital filter can be improved by L or N^2 times, compared with the CA-FIR digital filter realized using one such DSP. However, the convergence speed of the adaptive algorithm for the BIA-FIR digital filter or the DNA-FIR digital filter is degraded by a factor of L or N , respectively. Setting L equals to N , the convergence time needed by a DNA-FIR digital filter would be N times less than that needed by a BIA-FIR digital filter to obtain the same level of performance. The N times speed gain of a DNA-FIR digital filter stems from the N times more DSPs used in its structure.

Chapter 4

NONLINEAR DELAYED N-PATH ADAPTIVE FIR DIGITAL FILTER

The structure and the adaptive algorithm of a new nonlinear adaptive FIR digital filter, called nonlinear delayed N-path adaptive FIR (NDNA-FIR) digital filter, is presented for cancelling broadband noise from signals containing sharp edges. The filter structure is obtained by decomposing a FIR digital filter into N delayed parallel sub-filters which facilitates high speed implementation. Adjustable bipolar sigmoid functions are placed at the output of each sub-filter as well as after the summation of all sub-filter outputs. Simulation results show that the NDNA-FIR digital filter outperforms those of the conventional linear adaptive FIR digital filter and the nonlinear median filter in broadband noise cancellation in terms of convergence speed and average squared difference between original and recovered signals.

4.1 Introduction

The basic advantage of linear digital filtering is its simplicity in design and implementation, especially for adaptive FIR digital filtering which has been utilized in many practical situations [2,7-8]. However, in some circumstances, the performance of

nonlinear digital filters is better than that of linear digital filters [19]. This is the case when an image with sharp edges is contaminated by broadband noise, linear digital filtering would result in edge blurring along with noise removal. This can be a harmful quality degradation, because sharp edges are important image features and are essential for accurate motion estimation [32]. In speech signal processing and telecommunications, existing techniques of linear filtering are considered to be inadequate in applications for rejecting noise in a signal having an envelope with steep edges and that is corrupted by random noise [33]. Consider linear adaptive prediction which is used to extract a periodic signal from a broadband noise background. The order of the adaptive linear FIR digital filter used should be equal to or greater than two times that of the maximum frequency component of the predicted periodic signal [34]. When the signal to be predicted consists of high frequency components, such as a square wave corrupted by random noise as shown in Fig. 4.1, the performance of a linear adaptive FIR filter would not be satisfactory unless a sufficiently high filter order is used. A higher filter order implies a higher implementation cost. Moreover, it is always impossible to relate two signals in which their significant spectrum components do not overlap in the frequency domain by a linear filter [62]. It is inevitable, under such circumstances, that nonlinear filtering techniques are receiving more and more attention recently [19] [32-36] [62-64]. As seen from [19] and [32-35], nonlinear digital filters have played an important role in noise cancellation and edge preservation. Nonlinear order statistic digital filters have been presented in [32] and [35-36], and two types of nonlinear adaptive digital filters using neural networks have also been proposed in [33-34].

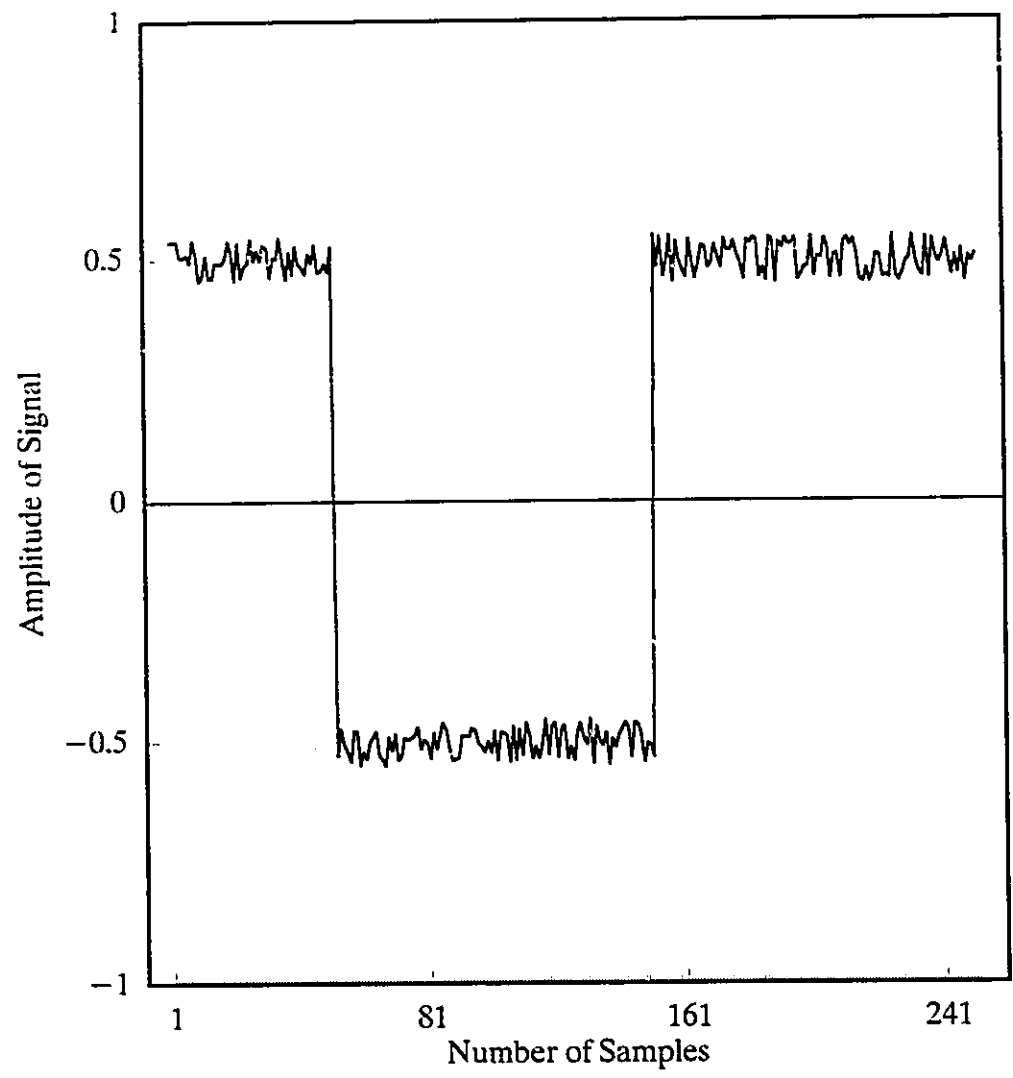


Fig. 4.1 Signal waveform with sharp edges corrupted by random noise

In this chapter, the structure and the adaptive algorithm of a nonlinear delayed N-path adaptive FIR (NDNA-FIR) digital filter is presented for cancelling broadband noise from signals containing sharp edges. The filter structure is obtained by decomposing a conventional FIR digital filter into N delayed parallel sub-filters. Adjustable bipolar sigmoid functions are placed at the output of each sub-filter as well as after the summation of all sub-filter outputs. This type of parallel structure lends itself to high speed realization and implementation by implementing the adaptive algorithm of each of the N sub-filters by one independent processor. Besides adjusting the filter coefficients, all the parameters of the two level sigmoid functions, which include gain, slope, and bias, are also adjusted in the adaptive algorithm. Due to high order nonlinearities, these two level adjustable nonlinear functions introduced play a significant role in cancelling noise from signals of sharp edges. The harmonic-rich spectral components in the sharp edges of an input signal appear to be more easily tracked by such a nonlinear adaptation. As demonstrated by simulation results, the proposed NDNA-FIR digital filter outperforms both the conventional linear adaptive FIR digital filter and the nonlinear median filter, for extracting signals with sharp edges corrupted by random noise.

4.2 Structure of Nonlinear Delayed N-Path Adaptive FIR (NDNA-FIR) Digital Filter

Fig. 4.2 shows a linear adaptive FIR (LA-FIR) digital filter, which can be used to extract a periodic input signal from a broadband noise background [7]. The input

signal $x(k)$, which is also called the primary input, consists of a mixed periodic signal $s(k)$ and a white noise $n(k)$. The output signal $y(k)$ is taken from the linear adaptive FIR digital filter, and is expected to be a close replica of the input periodic signal $s(k)$ after convergence. The delay z^{-d} is chosen to be of sufficient length such that the broadband signal components in the reference input $r(k)$ become uncorrelated with those in the primary input $x(k)$.

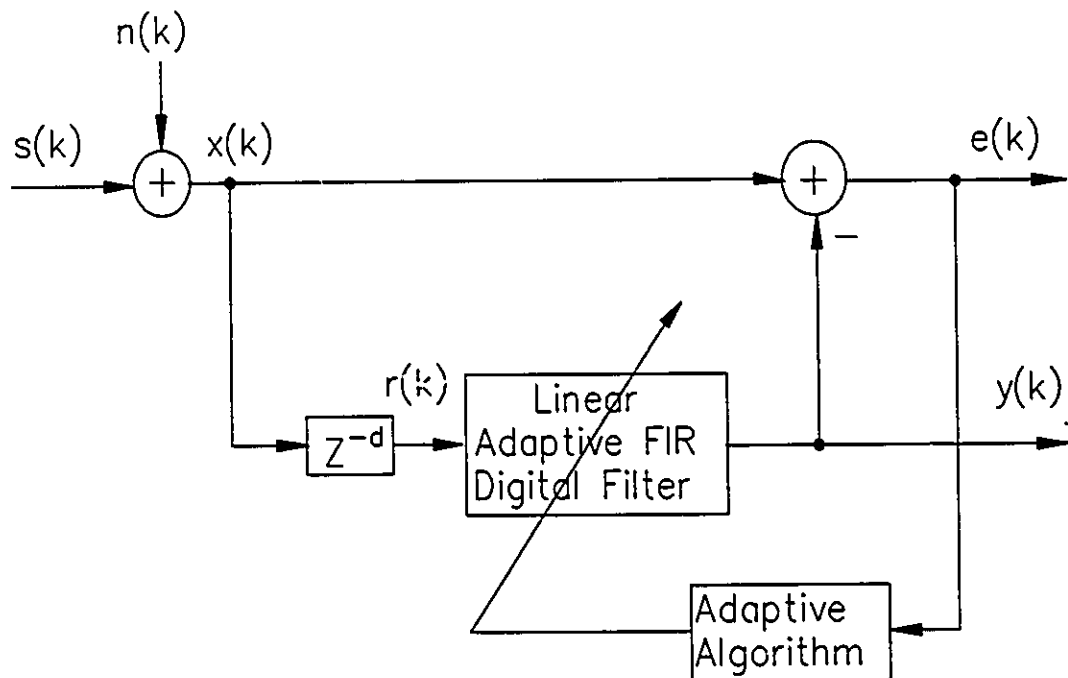


Fig. 4.2 Block diagram of LA-FIR digital filter

This system works well for a periodic input signal which does not have rich high frequency components, for example, a sinusoidal signal of a modest frequency. When an input signal $x(k)$ contains sharp edges and also flat regions like a periodic rectangular signal shown in Fig. 4.2, the performances of such a linear adaptive filter will deteriorate. By the concept of linear adaptive filtering, edge preservation would imply that the filter should have some wide band property, which is, however, in general detrimental to the non-edge areas. Under normal circumstances, the tracking property of a linear adaptive filter is poor when dealing with signals consisting of sharp edges. A large step size can lead to a fast reaction on edges and also a large residual error in non-edge areas.

4.2.1 Structure of NDNA-FIR Digital Filter

Fig. 4.3 shows the structure of the NDNA-FIR digital filter when used in broadband noise cancellation. The transfer function of its i th sub-filter, $H_{i,k}(z^{-N})$, is defined as

$$H_{i,k}(z^{-N}) = \sum_{l=0}^{M_i-1} h_{i,k}(l)z^{-lN} \quad \text{for } i = 0, 1, \dots, N-1 \quad (4.1)$$

where $k=0, 1, 2, \dots, \infty$ represents the discrete time index. The signal $x(k)$, which is a combination of the periodic signal $s(k)$ and the broadband or white noise $n(k)$, shall form the input signal to the nonlinear adaptive filter. The difference between $x(k)$ and the output signal of the nonlinear adaptive filter $y(k)$ forms the error signal $e(k)$.

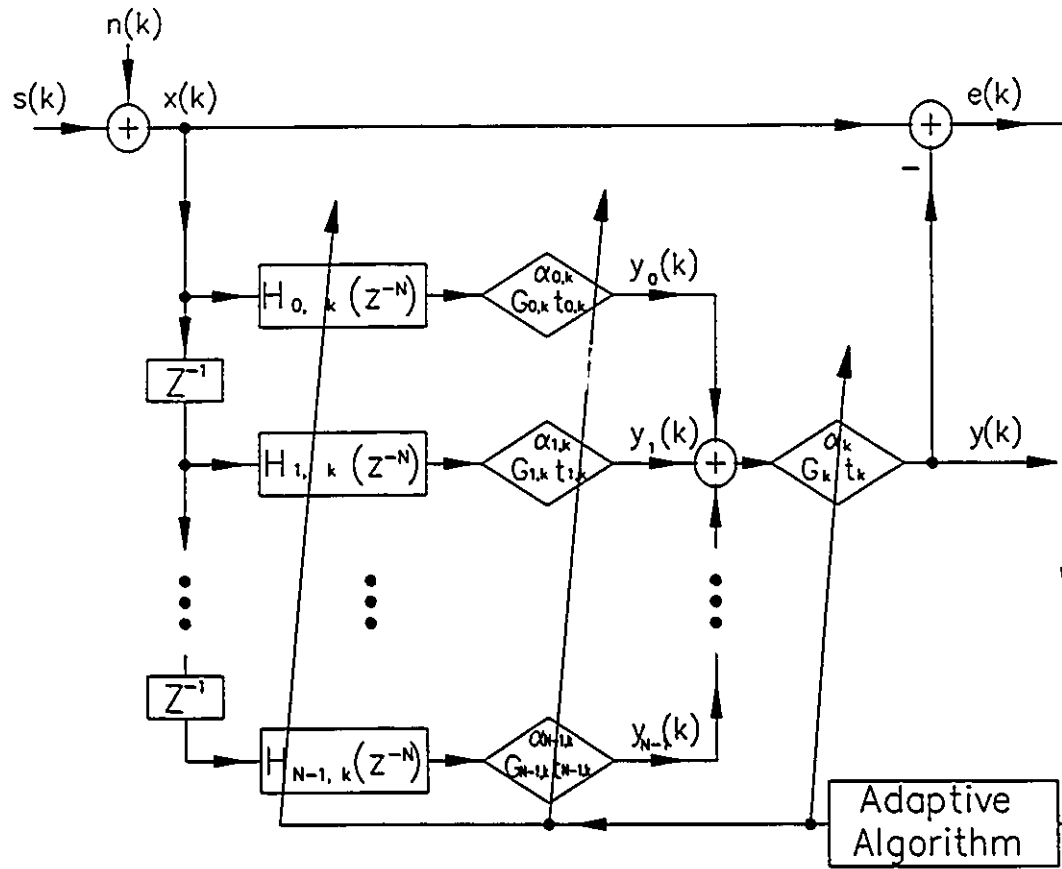


Fig. 4.3 Block diagram of NDNA-FIR digital filter

The diamond blocks shown in Fig. 4.3 stand for nonlinear operators which are defined as shown in Fig. 4.4. The nonlinear function used here is a hyperbolic tangent which is also called the bipolar sigmoid function operation. Its mathematical expressions are as follows:

$$v = F(u, \alpha, G, t) = G \frac{1 - e^{-\alpha(u+t)}}{1 + e^{-\alpha(u+t)}} \quad (4.2a)$$

or

$$v = F(z, \alpha, G) = G \frac{1 - e^{-\alpha z}}{1 + e^{-\alpha z}} \quad (4.2b)$$

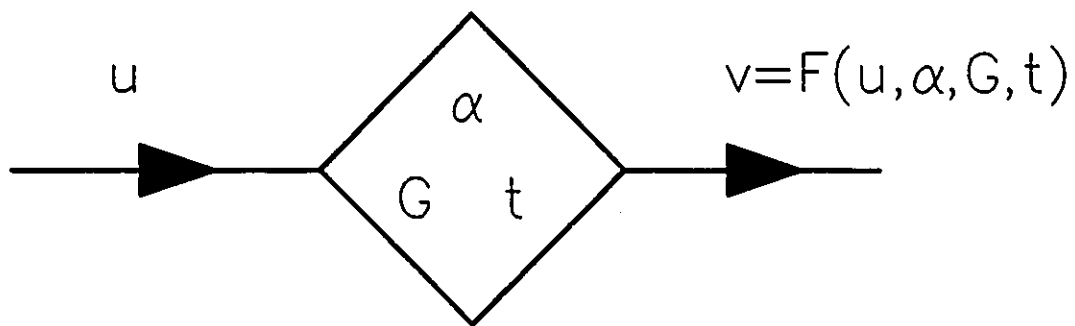


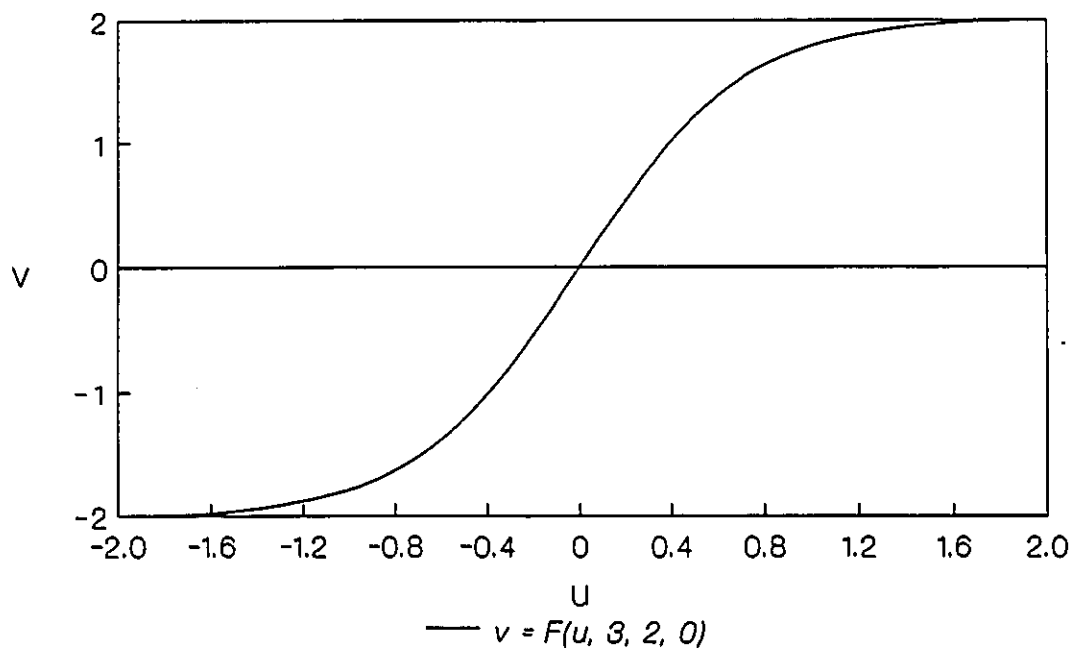
Fig. 4.4 Nonlinear operator of the sigmoid function

where u represents the input signal to the nonlinear operator and v stands for the output signal coming out from the nonlinear operator. α , G , and t are adjustable parameters. Eqn. 4.2b is an alternative expression of eqn. 4.2a where $z=u+t$, which would facilitate derivations of the adaptive algorithm. The adaptive algorithm reduces the instantaneous squared error, $e^2(k)$, gradually by adaptively adjusting weights of $h_{i,k}(l)$ and parameters of $\alpha_{i,k}$, $G_{i,k}$, $t_{i,k}$, α_k , G_k , and t_k , for $i = 0, 1, \dots, N-1$; $k = 0, 1, 2, \dots$; and $l = 0, 1, \dots, M_i-1$.

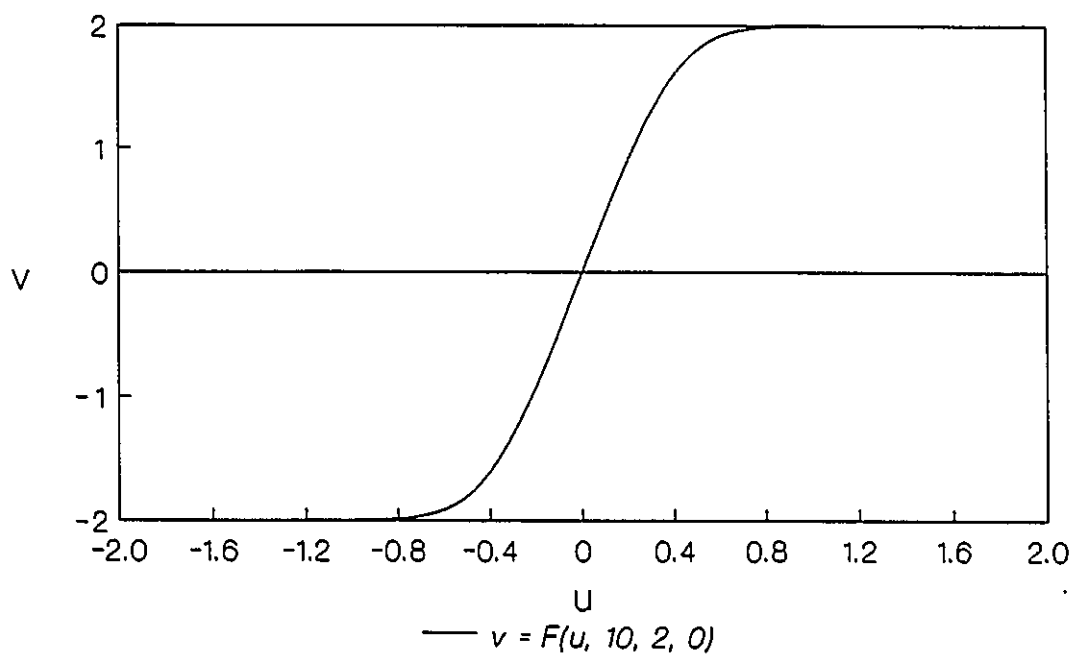
4.2.2 Sigmoid Function

An example of the nonlinear function as expressed in eqn. 4.2 is shown in Fig. 4.5a with $\alpha=3$, $G=2$, and $t=0$. It has a differentiable input-output characteristic which facilitates adaptability. Its nonsaturation and saturation regions are playing separate roles in noise cancellation in edge areas and non-edge (or flat) areas, respectively. The shape of this nonlinear function is governed by parameters α , G , and t as noted in Fig. 4.4. α and G here are assumed to be positive values. This function is anti-symmetric to the straight line $v=-t$ (where $t=0$) as shown in Fig. 4.5a. t is a shift parameter which is introduced to bias the input signal to the nonlinear operator, and hence to improve adaptability of the nonlinear function. The maximum and minimum values of this function are G and $-G$ when z or $(u+t)$ approaches positive infinite and negative infinite, respectively. The parameter α controls the sharpness of the curve within the nonsaturation region. If a bigger α is selected, the nonsaturation region would become narrower, and the function increases faster within this region. When α is chosen smaller,

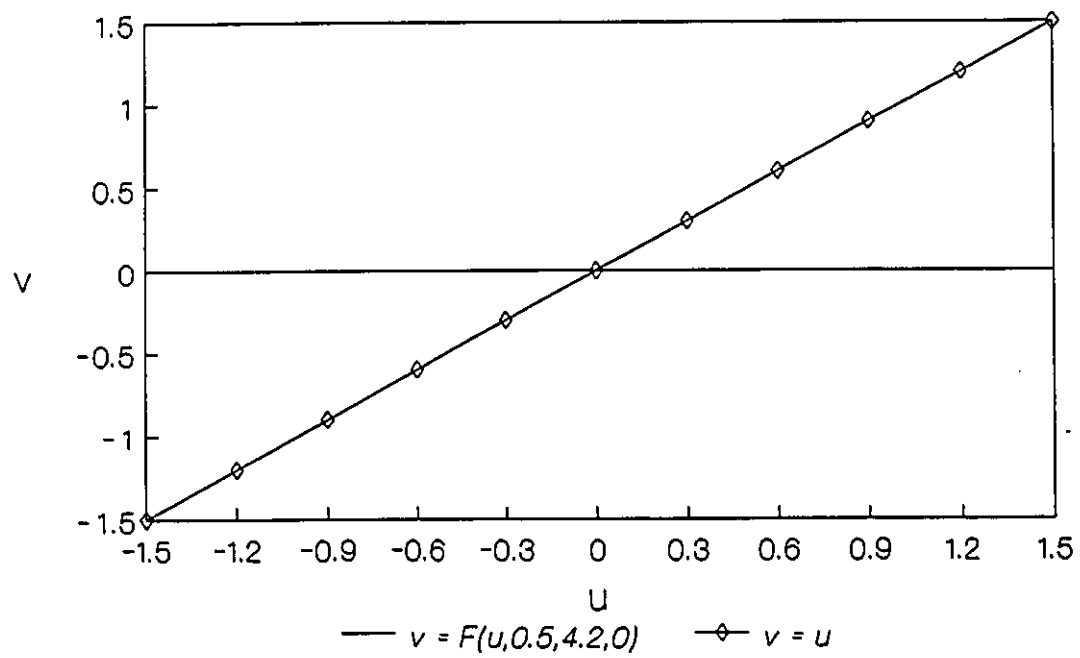
the whole curve would become flatter and the extreme case is that the nonlinear function becomes a straight line of $v=0$ when $\alpha=0$. Fig. 4.5a and Fig. 4.5b show curves of this nonlinear function with the same values of G and t , but different values of α within the same region of $[-2,2]$. Another observation is that the nonlinear function will approximate a linear function $v=u$ very closely within a given area when a proper combination of parameter values of α , G , and t is made. Shown in Fig. 4.5c are the linear function $v=u$ and the nonlinear function with $\alpha=0.5$, $G=4.2$, and $t=0$. The average squared difference between them turns out to be 3.288857×10^{-1} based on 150 samples evenly spaced over the interval of $[-1.5, 1.5]$. Another example is shown in Fig. 4.5d with $\alpha=0.1$, $G=20$, and $t=0$. Its average squared difference is 1.284112×10^{-6} based on 200 samples over the interval of $[-2, 2]$. Hence, to a certain extent, the linear adaptive digital filter shown in Fig. 4.2 can be viewed as a special case of this nonlinear adaptive digital filter when the nonlinear function in this system approaches the linear function of $v=u$. It can be expected from this point of view that with a proper adjustment of parameters $\alpha_{i,k}$, $G_{i,k}$, $t_{i,k}$, α_k , G_k , and t_k together with weights $h_{i,k}(l)$, for $i = 0, 1, \dots, N-1$ and $l = 0, 1, \dots, M_i-1$, this nonlinear system would have a better tracking property when the input signal has sharp edges and also flat areas. The nonlinear operator right before the output signal $y(k)$ would further improve the tracking property which can be observed from our simulation results.



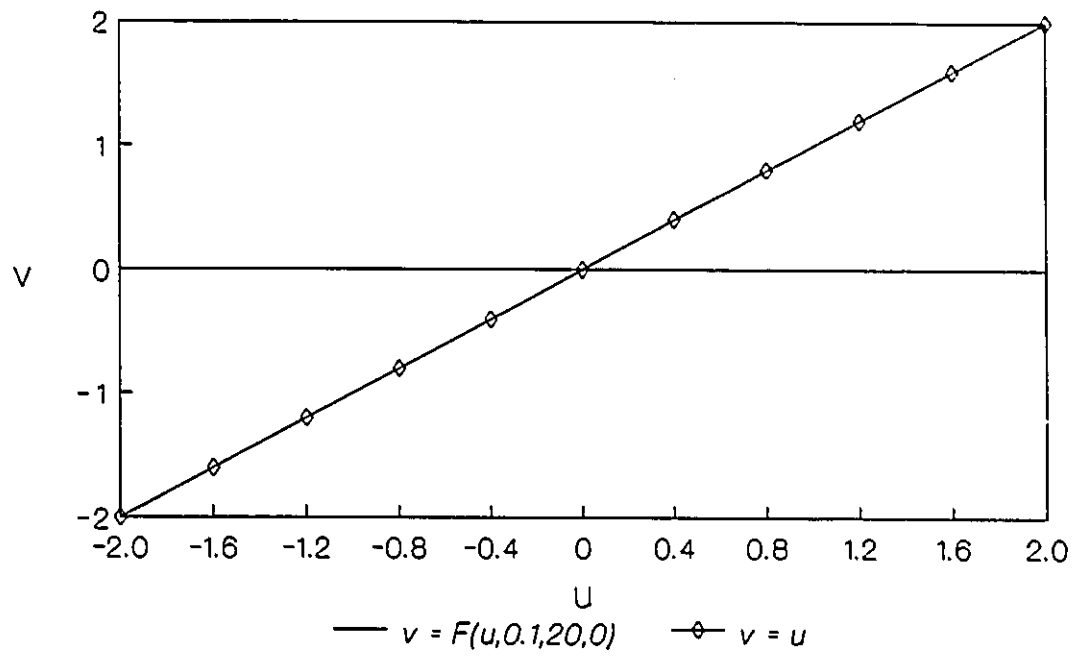
(a) $\alpha = 3$, $G = 2$, and $t = 0$



(b) $\alpha = 10$, $G=2$, and $t = 0$



(c) $\alpha = 0.5$, $G = 4.2$, and $t = 0$



(d) $\alpha = 0.1$, $G = 20$, and $t = 0$

Fig. 4.5 Examples of input-output characteristics of sigmoid function with different parameter values of α , G , and t

4.3 Adaptive Algorithm of Nonlinear Delayed N-Path Adaptive FIR (NDNA-FIR) Digital Filter

We define the error signal as shown in Fig. 4.3 as

$$e(k) = x(k) - y(k) \quad (4.3)$$

The mean square error is

$$\Phi = E[e^2(k)] \quad (4.4)$$

where E is a mean value operator. The adaptive algorithm attempts to minimize the mean square error by searching the optimum weights and parameters along the direction of negative gradients of the error surface.

In a practical situation, Φ is usually not available and gradients of $e^2(k)$ are used instead to approximate real gradients of the error surface. The error surface in this case is not quadratic like the linear FIR case because of the nonlinearity involved. It may have local minima and hence special precautions on the choices of initial parameters must be taken. In our simulations, all weight and coefficient updates are being done once only after K samples of a certain length of the signal have been presented, so that the noise with the estimated gradients can be smoothed out to a certain degree. In the following, we will derive the gradient estimates of all the weights and parameters at each iteration in the adaptive process.

As shown in Fig. 4.3, the output signal $y_i(k)$ of each parallel sub-filter is

$$y_i(k) = F(z_i(k), \alpha_{i,k}, G_{i,k}) \quad (4.5)$$

where

$$z_i(k) = \sum_{l=0}^{M_i-1} h_{i,k}(l)x(k-lN-i) + t_{i,k} \quad (4.6)$$

for $i = 0, 1, \dots, N-1$. The overall output of the nonlinear adaptive filter is

$$y(k) = F(z(k), \alpha_k, G_k) \quad (4.7)$$

where

$$z(k) = \sum_{i=0}^{N-1} y_i(k) + t_k \quad (4.8)$$

The squared output error is defined as

$$e^2(k) = (x(k) - y(k))^2 \quad (4.9)$$

The changes of weights and parameters in each iteration are defined as

$$\Delta h_{i,k}(l) = -\mu \frac{\partial e^2(k)}{\partial h_{i,k}(l)} \quad (4.10)$$

$$\Delta \alpha_{i,k} = -\mu \frac{\partial e^2(k)}{\partial \alpha_{i,k}} \quad (4.11)$$

$$\Delta G_{i,k} = -\mu \frac{\partial e^2(k)}{\partial G_{i,k}} \quad (4.12)$$

$$\Delta r_{i,k} = -\mu \frac{\partial e^2(k)}{\partial r_{i,k}} \quad (4.13)$$

$$\Delta \alpha_k = -\mu \frac{\partial e^2(k)}{\partial \alpha_k} \quad (4.14)$$

$$\Delta G_k = -\mu \frac{\partial e^2(k)}{\partial G_k} \quad (4.15)$$

and

$$\Delta r_k = -\mu \frac{\partial e^2(k)}{\partial r_k} \quad (4.16)$$

for $i = 0, 1, \dots, N-1$, $l = 0, 1, \dots, M_l-1$, $k = 0, 1, 2, \dots$, and μ is the gain constant that regulates convergence speed of adaptation.

Before we derive the updates of these weights and parameters, we define

$$\begin{aligned} \delta_k &= -\frac{\partial e^2(k)}{\partial z(k)} \\ &= -\frac{\partial e^2(k)}{\partial y(k)} \frac{\partial y(k)}{\partial z(k)} \\ &= 2e(k) \frac{\partial F(z(k), \alpha_k, G_k)}{\partial z(k)} \end{aligned} \quad (4.17)$$

and

$$\begin{aligned}
\delta_{i,k} &= -\frac{\partial e^2(k)}{\partial z_i(k)} \\
&= -\frac{\partial e^2(k)}{\partial y_i(k)} \frac{\partial y_i(k)}{\partial z_i(k)} \\
&= -\frac{\partial e^2(k)}{\partial y_i(k)} \frac{\partial F(z_i(k), \alpha_{i,k}, G_{i,k})}{\partial z_i(k)}
\end{aligned} \tag{4.18}$$

Noticing that

$$\frac{\partial e^2(k)}{\partial y_i(k)} = \frac{\partial e^2(k)}{\partial z(k)} \frac{\partial z(k)}{\partial y_i(k)} = -\delta_k \tag{4.19}$$

We have the following relation by substituting eqn. 4.19 into eqn. 4.18

$$\delta_{i,k} = \delta_k \frac{\partial F(z_i(k), \alpha_{i,k}, G_{i,k})}{\partial z_i(k)} \tag{4.20}$$

for $i = 0, 1, \dots, N-1$ and $k = 0, 1, \dots$

For simplicity in the following derivations, the subscript indexes i , l , and k will always represent integers from 0 to $N-1$, 0 to M_i-1 , and 0 to positive indefinite, respectively, unless stated otherwise.

4.3.1 Updates of Weights $h_{i,k}(l)$

From eqns. 4.10, 4.18, and 4.6, we have

$$\begin{aligned}
\Delta h_{i,k}(l) &= -\mu \frac{\partial e^2(k)}{\partial z_i(k)} \frac{\partial z_i(k)}{\partial h_{i,k}(l)} \\
&= \mu \delta_{i,k} x(k-lN-i)
\end{aligned} \tag{4.21}$$

Updating of $h_{i,k}(l)$, as well as other parameters, are not carried out at every discrete time k . Instead, they are performed at every K iterations by adding the total summation of increments over K consecutive iterations, when k divided by K gives an integer. Define a subset of time indexes of k as

$$S = \{ k \mid k = 0, 1, \dots, \text{ and } k/K = \text{integer} \} \quad (4.22)$$

then we have

$$h_{i, k+1}(l) = h_{i,k}(l) + \sum_{m=0}^{K-1} \Delta h_{i, k-m}(l) \quad (4.23)$$

4.3.2 Updates of Parameters $\alpha_{i,k}$ and α_k

In a similar manner as above, it follows from eqn. 4.11 and 4.19 that

$$\begin{aligned} \Delta \alpha_{i, k} &= -\mu \frac{\partial e^2(k)}{\partial y_i(k)} \frac{\partial y_i(k)}{\partial \alpha_{i, k}} \\ &= \mu \delta_k \frac{\partial F(z_i(k), \alpha_{i, k}, G_{i, k})}{\partial \alpha_{i, k}} \end{aligned} \quad (4.24)$$

From eqns. 4.14 and 4.9, we have

$$\begin{aligned} \Delta \alpha_k &= -\mu \frac{\partial e^2(k)}{\partial y(k)} \frac{\partial y(k)}{\partial \alpha_k} \\ &= 2 \mu e(k) \frac{\partial F(z(k), \alpha_k, G_k)}{\partial \alpha_k} \end{aligned} \quad (4.25)$$

The updates of $\alpha_{i,k}$ and α_k at every K iterations when k belongs to S defined in eqn. 4.22 are

$$\alpha_{i, k+1} = \alpha_{i, k} + \sum_{m=0}^{K-1} \Delta\alpha_{i, k-m} \quad (4.26)$$

and

$$\alpha_{k+1} = \alpha_k + \sum_{m=0}^{K-1} \Delta\alpha_{k-m} \quad (4.27)$$

while at other steps they remain unchanged.

4.3.3 Updates of Parameters $G_{i,k}$ and G_k

In the same way, we can find out from eqns. 4.12 and 4.19 that

$$\begin{aligned} \Delta G_{i, k} &= -\mu \frac{\partial e^2(k)}{\partial y_i(k)} \frac{\partial y_i(k)}{\partial G_{i, k}} \\ &= \mu \delta_k \frac{\partial F(z_i(k), \alpha_{i, k}, G_{i, k})}{\partial G_{i, k}} \end{aligned} \quad (4.28)$$

Also from eqns. 4.15 and 4.9, we obtain

$$\begin{aligned} \Delta G_k &= -\mu \frac{\partial e^2(k)}{\partial y(k)} \frac{\partial y(k)}{\partial G_k} \\ &= 2 \mu e(k) \frac{\partial F(z(k), \alpha_k, G_k)}{\partial G_k} \end{aligned} \quad (4.29)$$

$G_{i,k}$ and G_k are not changed except at the discrete times when k falls into the S set in eqn.

4.22. They are updated according to

$$G_{i, k+1} = G_{i, k} + \sum_{m=0}^{K-1} \Delta G_{i, k-m} \quad (4.30)$$

and

$$G_{k+1} = G_k + \sum_{m=0}^{K-1} \Delta G_{k-m} \quad (4.31)$$

4.3.4 Updates of Parameters $t_{i,k}$ and t_k

From eqns 4.13, 4.18, and 4.6, we arrive at

$$\Delta t_{i,k} = -\mu \frac{\partial e^2(k)}{\partial z_i(k)} \frac{\partial z_i(k)}{\partial t_{i,k}} = \mu \delta_{i,k} \quad (4.32)$$

and from eqns. 4.16, 4.17, and 4.8, we obtain

$$\Delta t_k = -\mu \frac{\partial e^2(k)}{\partial z(k)} \frac{\partial z(k)}{\partial t_k} = \mu \delta_k \quad (4.33)$$

The updates of $t_{i,k}$ and t_k are taken place at the same discrete times as above, i.e.,

$$t_{i,k+1} = t_{i,k} + \sum_{m=0}^{K-1} \Delta t_{i,k-m} \quad (4.34)$$

and

$$t_{k+1} = t_k + \sum_{m=0}^{K-1} \Delta t_{k-m} \quad (4.35)$$

while at other steps they remain unchanged. Knowing that the partial derivatives of function $F(z, \alpha, G)$ in eqn. 4.2b with respect to z , α , and G are as follows

$$\frac{\partial F(z, \alpha, G)}{\partial z} = 2G\alpha \frac{e^{-\alpha}}{(1 + e^{-\alpha})^2} \quad (4.36a)$$

$$\frac{\partial F(z, \alpha, G)}{\partial \alpha} = 2Gz \frac{e^{-\alpha}}{(1 + e^{-\alpha})^2} \quad (4.36b)$$

and

$$\frac{\partial F(z, \alpha, G)}{\partial G} = \frac{1 - e^{-\alpha z}}{1 + e^{-\alpha z}} \quad (4.36c)$$

Eqns. 4.17, 4.20, 4.24, 4.25, 4.28, and 4.29 can be evaluated respectively by choosing the corresponding formula from eqns. 4.36a-4.36c.

4.4 Simulation Results

Simulation results are presented in this section to illustrate the performance of the proposed NDNA-FIR digital filter as compared to those of the linear adaptive FIR (LA-FIR) digital filter and nonlinear median (NLMED) filter.

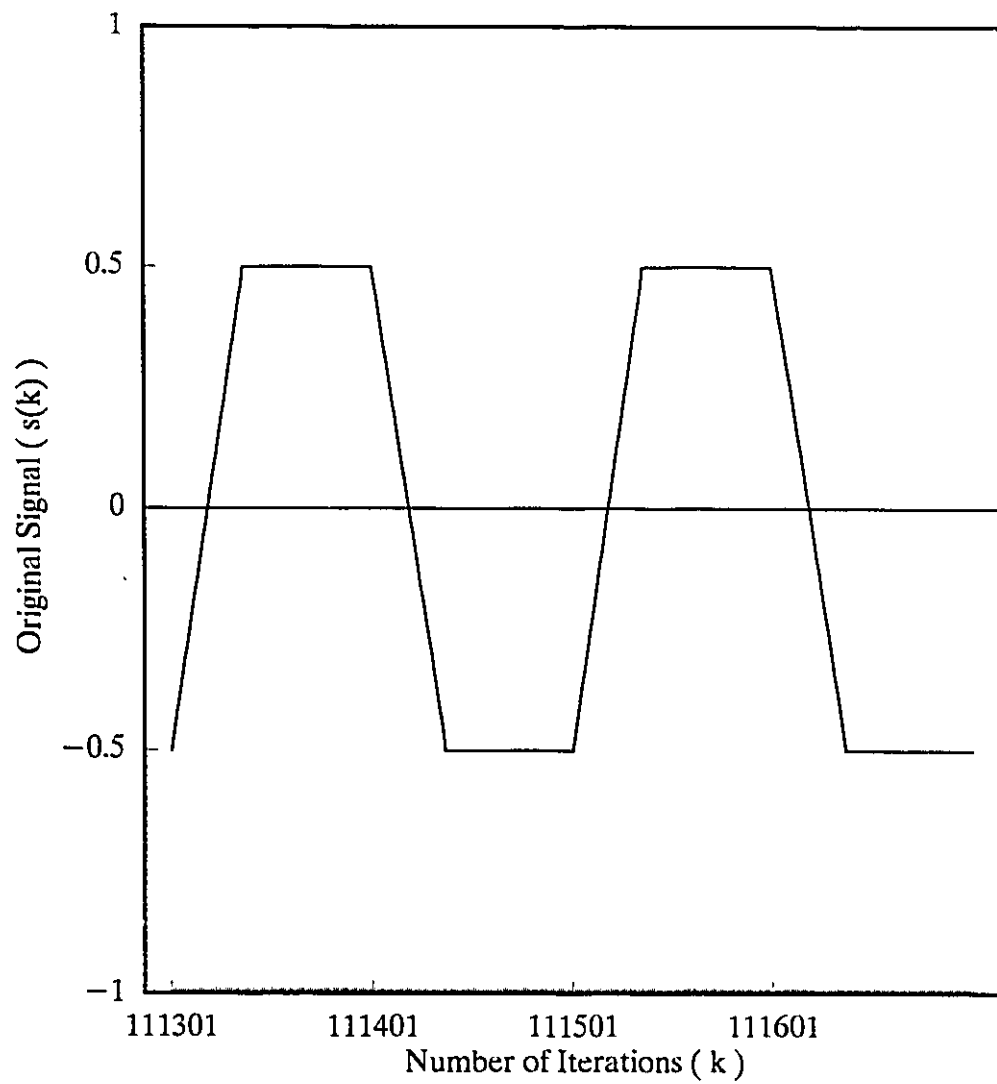
Example 1: In this example, the original signal $s(k)$ is a periodic trapezoidal signal of amplitude ± 0.5 as shown in Fig. 4.6a. $s(k)$ is corrupted by 10% white noise (i.e., noise distributed uniformly within ± 0.05) to form an input signal $x(k)$ as shown in Fig. 4.6b. The convergence criterion was set to 1.0×10^{-5} which represents the difference between the sums of $e^2(k)$ over two consecutive intervals of 100 iterations.

In the NDNA-FIR digital filter, a delayed 6-path structure was used where $M_0=11$ and $M_i=10$ for $i = 1$ to 5. The learning rate parameter, μ , was chosen as 1.0×10^{-4} and K was selected as 100. The coefficients of $h_{i,0}(l)$, for $i=0$ to 5 and $l = 0$ to M_i-1 , were initially set to zero. The initial values of the parameters were

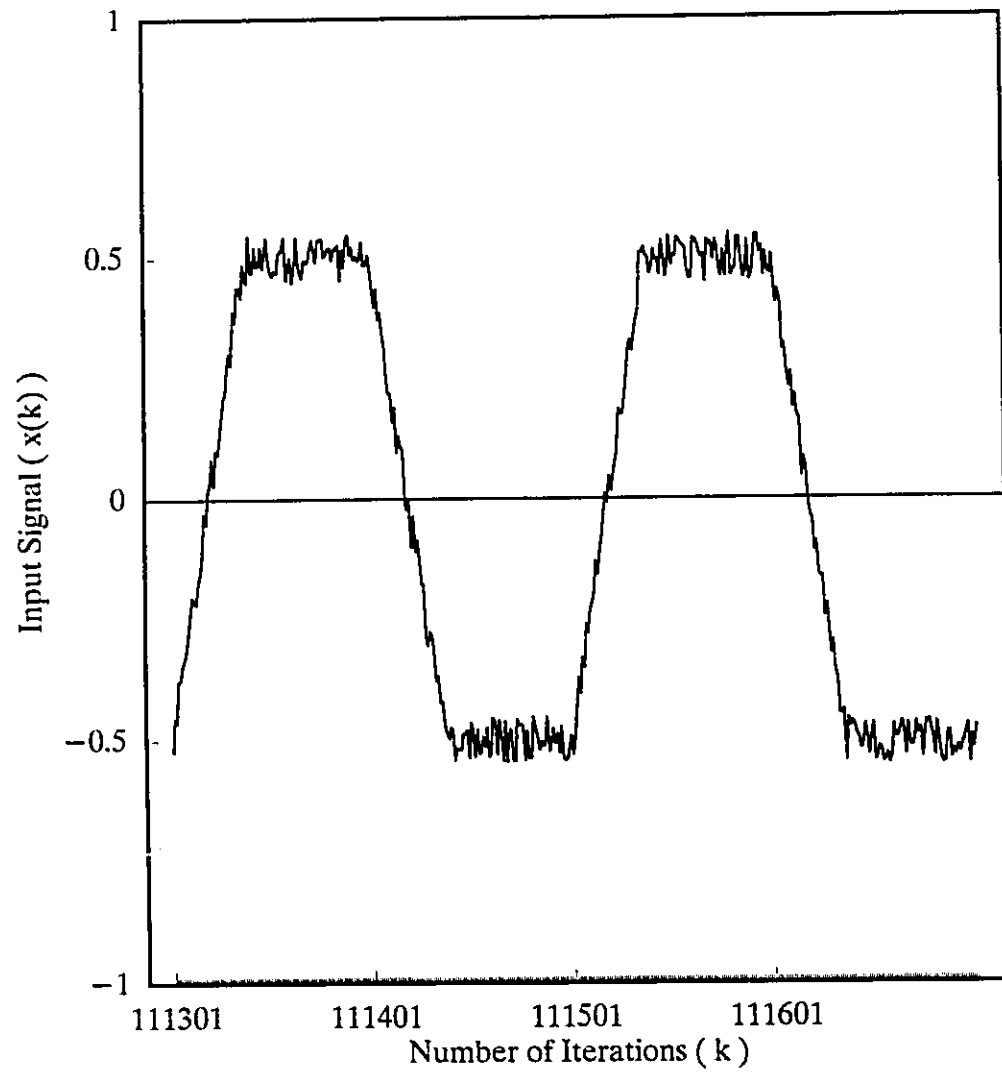
$\alpha_0 = \alpha_{i,0} = 6.0$, and $g_0 = g_{i,0} = 0.65$ for $i=0$ to 5. t_0 and $t_{i,0}$, for $i=0$ to 5, were set to random values within the range of ± 0.001 . The adaptive process converged after 111700 iterations, and the average squared output error, $e^2(k)$, over the last 100 iterations was 9.5544×10^{-4} . The average squared difference between $y(k)$ and $s(k)$ over the last 200 iterations was 4.2001×10^{-4} . Fig. 4.6c shows the converged output signal $y(k)$ and Fig. 4.6d shows the output squared error $e^2(k)$ in the adaptive process.

In the LA-FIR digital filter [7], the learning rate parameter of coefficients was set to 1.0×10^{-4} and the adaptive process was started from zero initial coefficients with delay $d=4$. Figs. 4.6e, 4.6f and 4.6g show, respectively, the converged output signals $y(k)$ of the LA-FIR digital filter with filter orders of 61, 82 and 121. These three filters converged, respectively, after 390100, 744800 and 737400 iterations, and their respective average squared output errors $e^2(k)$ over their last 100 iterations were 2.3613×10^{-3} , 1.2755×10^{-3} and 1.0389×10^{-3} . The average squared differences between $y(k)$ and $s(k)$ over the last 200 iterations were, respectively, 1.5620×10^{-3} , 5.5323×10^{-4} and 1.9502×10^{-4} .

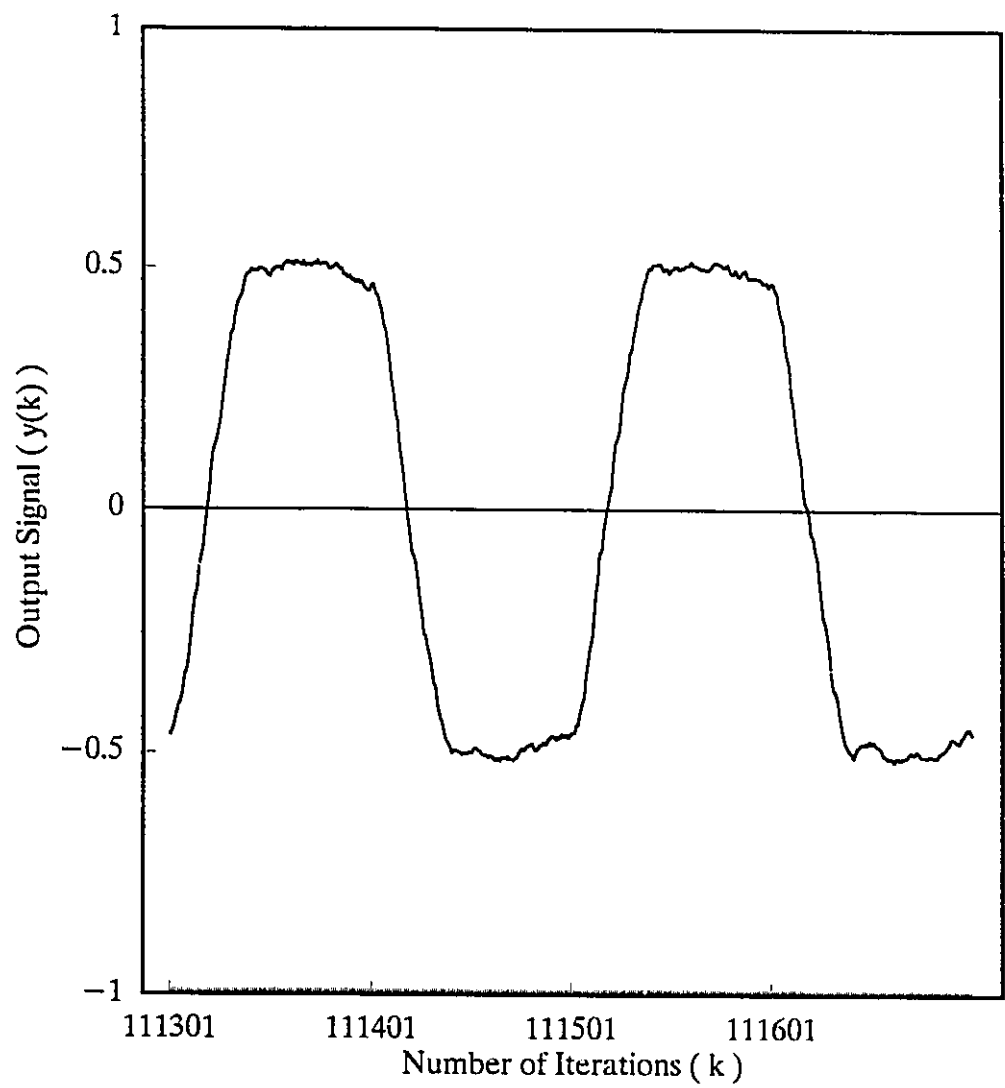
Fig. 4.6h shows the output signal of the NLMED digital filter with a window length of 3 using the input signal $x(k)$ shown in Fig. 4.6b. The average squared difference between this output signal, $y(k)$, and the original signal, $s(k)$, over the last 200 iterations was 5.9251×10^{-4} . We have also tried window lengths of all the odd numbers from 3 to 21, and the result for the window length of 3 was the best.



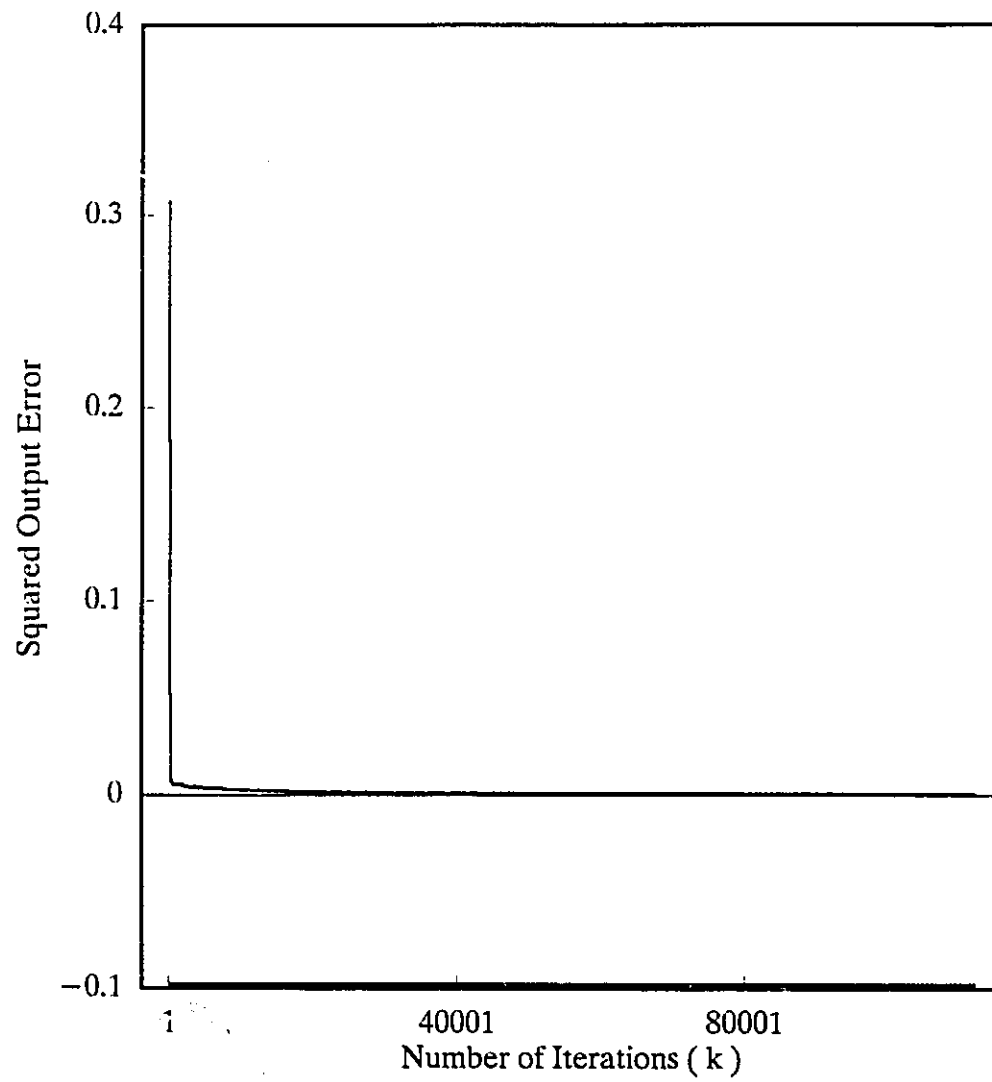
(a) Original signal $s(k)$



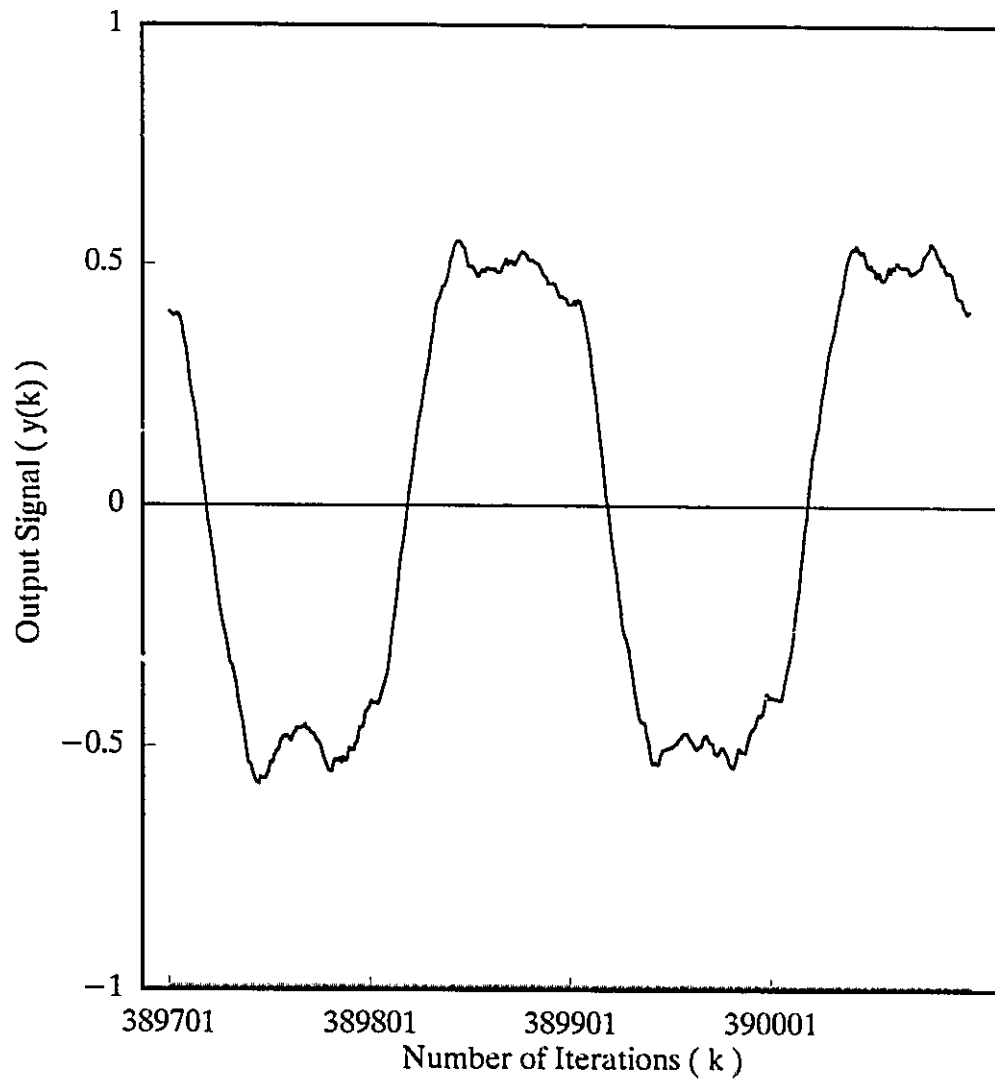
(b) Corrupted input signal $x(k)$



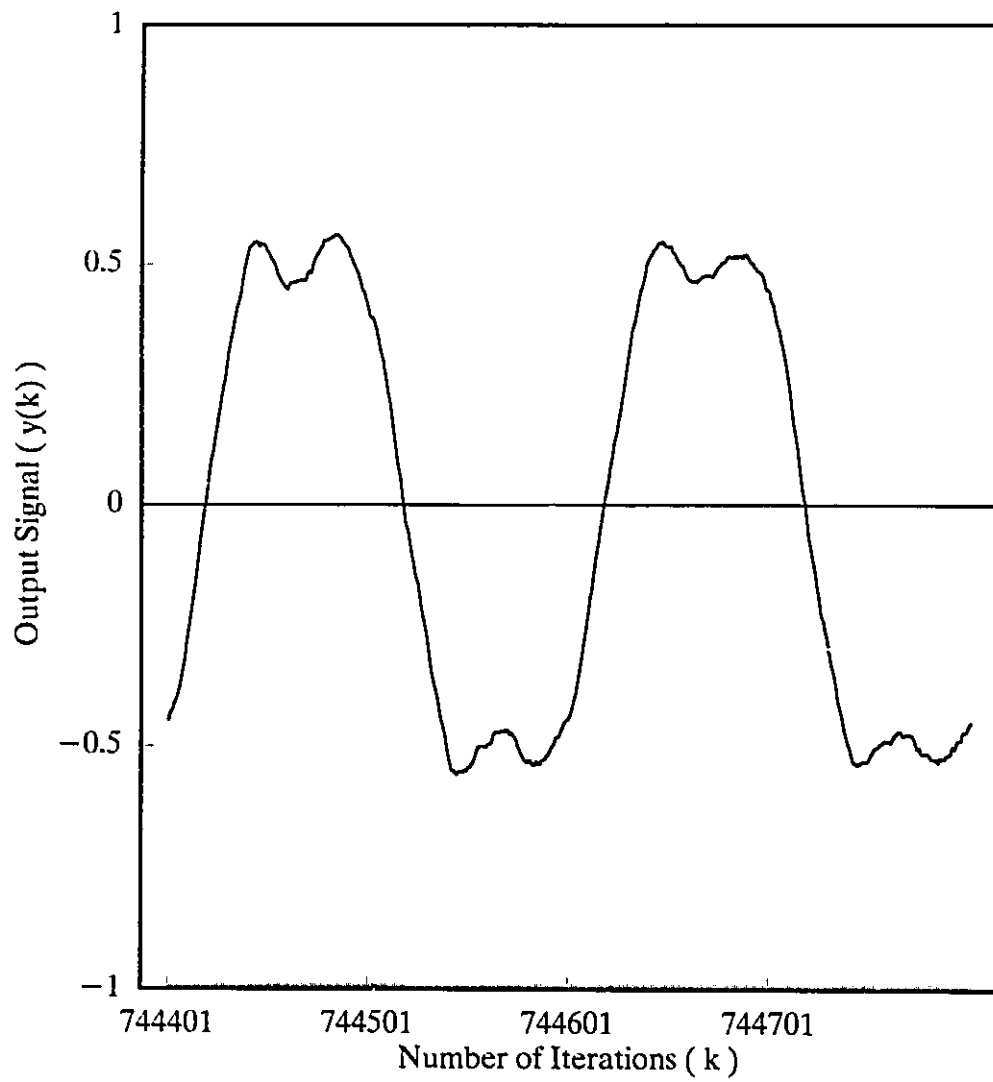
(c) Converged output signal $y(k)$ of NDNA-FIR digital filter



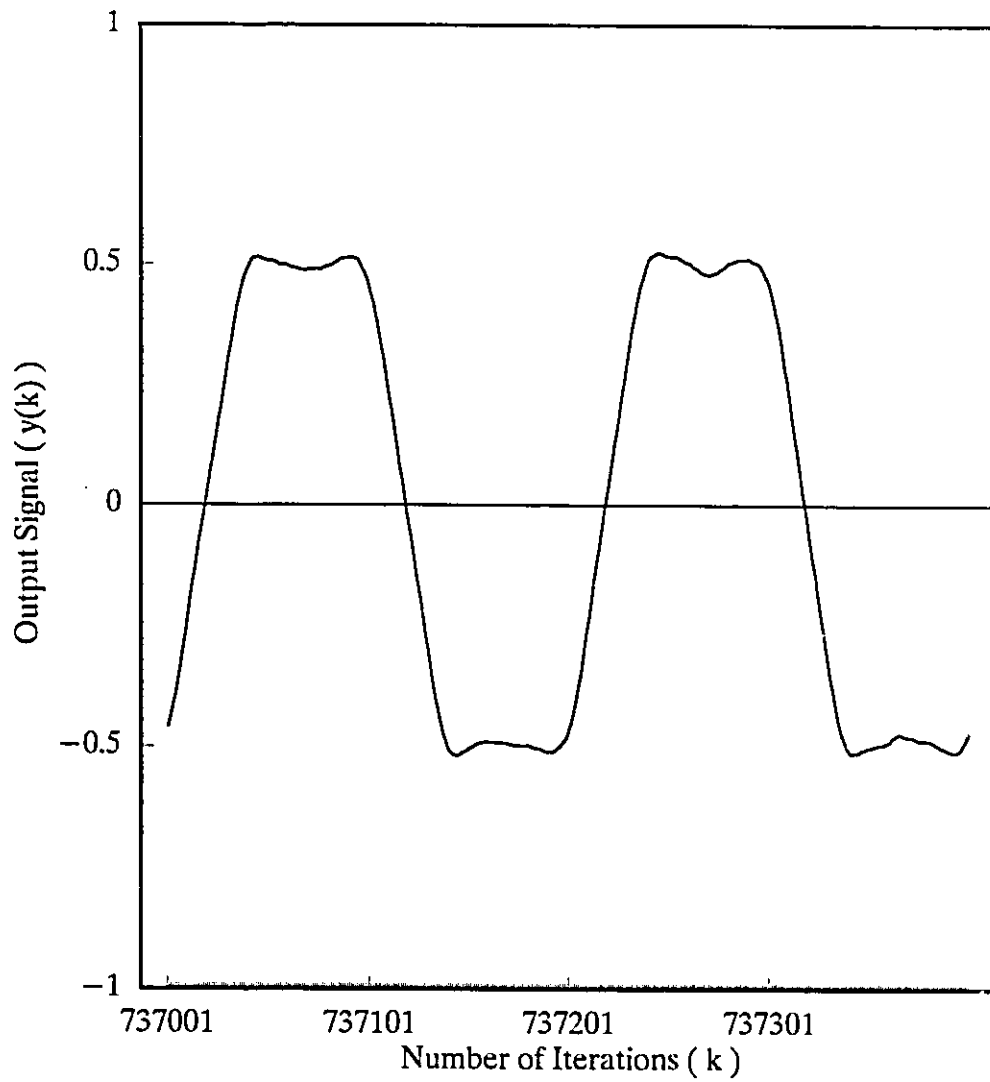
(d) Squared output error $e^2(k)$ of NDNA-FIR digital filter



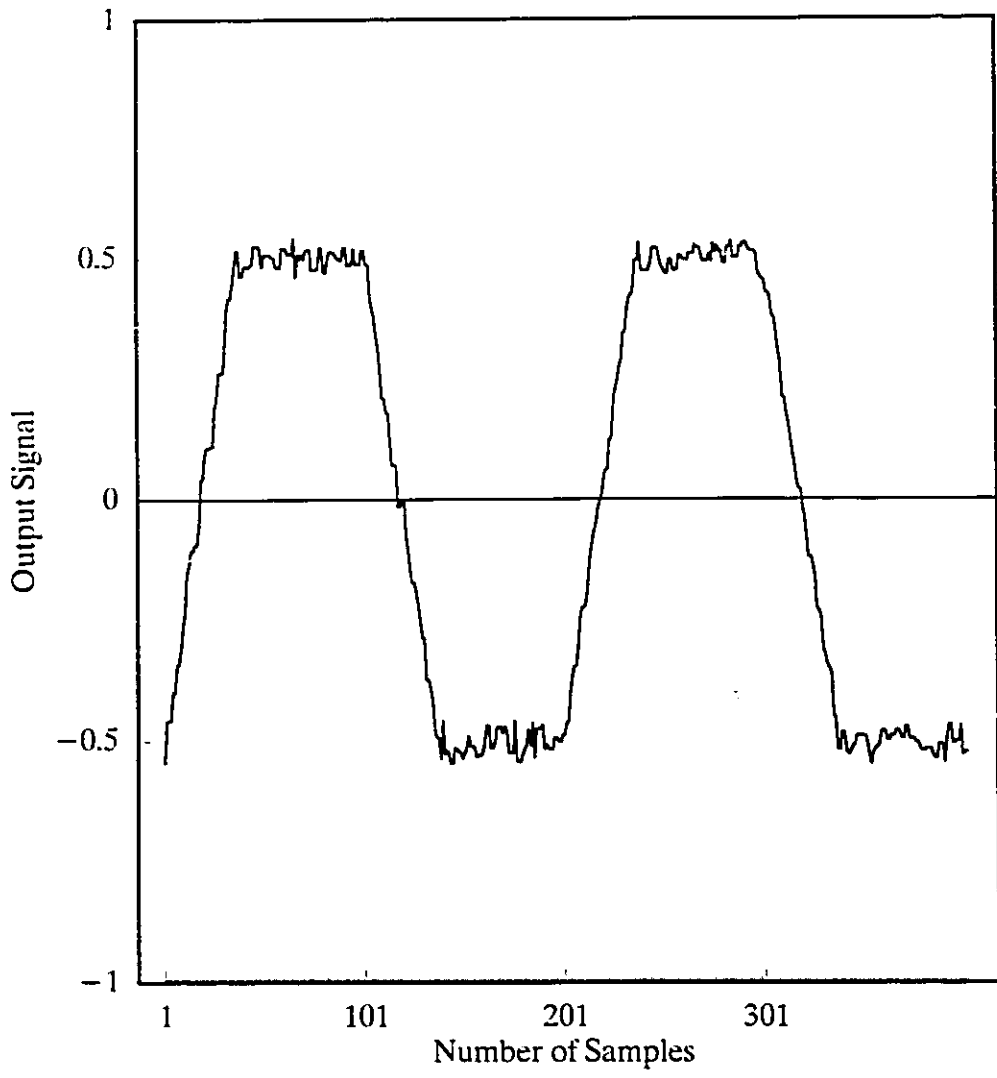
(e) Converged output signal $y(k)$ of LA-FIR digital filter of order 61



(f) Converged output signal $y(k)$ of LA-FIR digital filter of order 82



(g) Converged output signal $y(k)$ of LA-FIR digital filter of order 121



(h) Output signal of NLMED digital filter

Fig. 4.6 Simulation results in Example 1

Example 2: The original signal $s(k)$ used is a periodic rectangular signal of amplitude ± 0.5 as shown in Fig. 4.7a. The input signal $x(k)$ shown in Fig. 4.7b was obtained by combining $s(k)$ with a 10% additive white noise (i.e., noise distributed uniformly within ± 0.05). The convergence criterion was 1.0×10^{-4} , which represented the difference between the sums of $e^2(k)$ over two consecutive intervals of 1000 iterations.

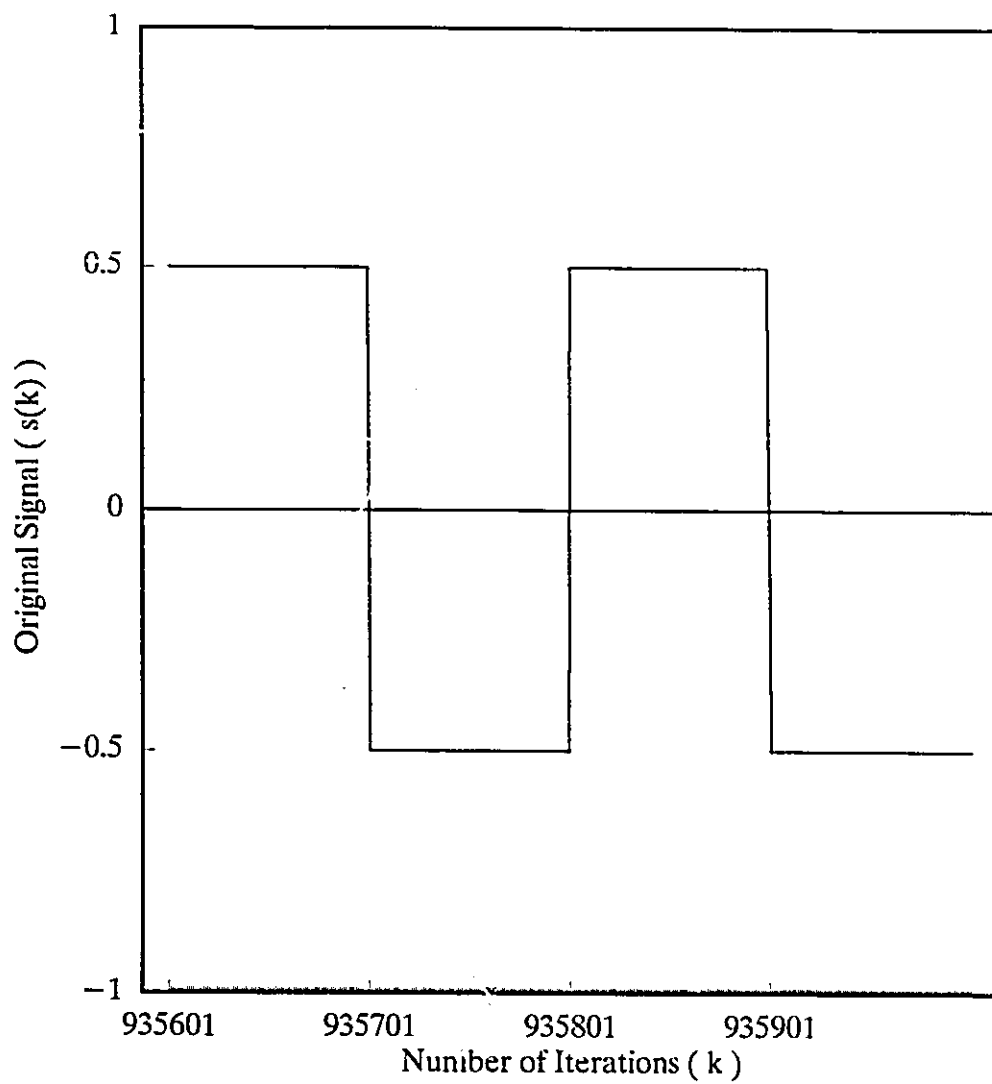
In the NDNA-FIR digital filter, a delayed 6-path structure was again used in this example. The learning rate parameter was set to 5.0×10^{-5} and K was chosen as 1000. The coefficients, $h_{i,0}(l)$ for $i=0$ to 5 and $l=0$ to M_i-1 , were all initially set to zero. The initial values of the parameters were $\alpha_0 = \alpha_{i,0} = 10.0$, and $g_0 = g_{i,0} = 0.5$, for $i=0$ to 5. t_0 and $t_{i,0}$, for $i=0$ to 5, were set to random values within ± 0.001 . The adaptive process converged after 936000 iterations, and the average squared output error $e^2(k)$ over the last 1000 iterations was 9.0601×10^{-4} . The average squared difference between $y(k)$ and $s(k)$ over the last 200 iterations was 2.0009×10^{-4} . Fig. 4.7c shows the converged output signal $y(k)$ and Fig. 4.7d shows the squared output error $e^2(k)$ in the adaptive process.

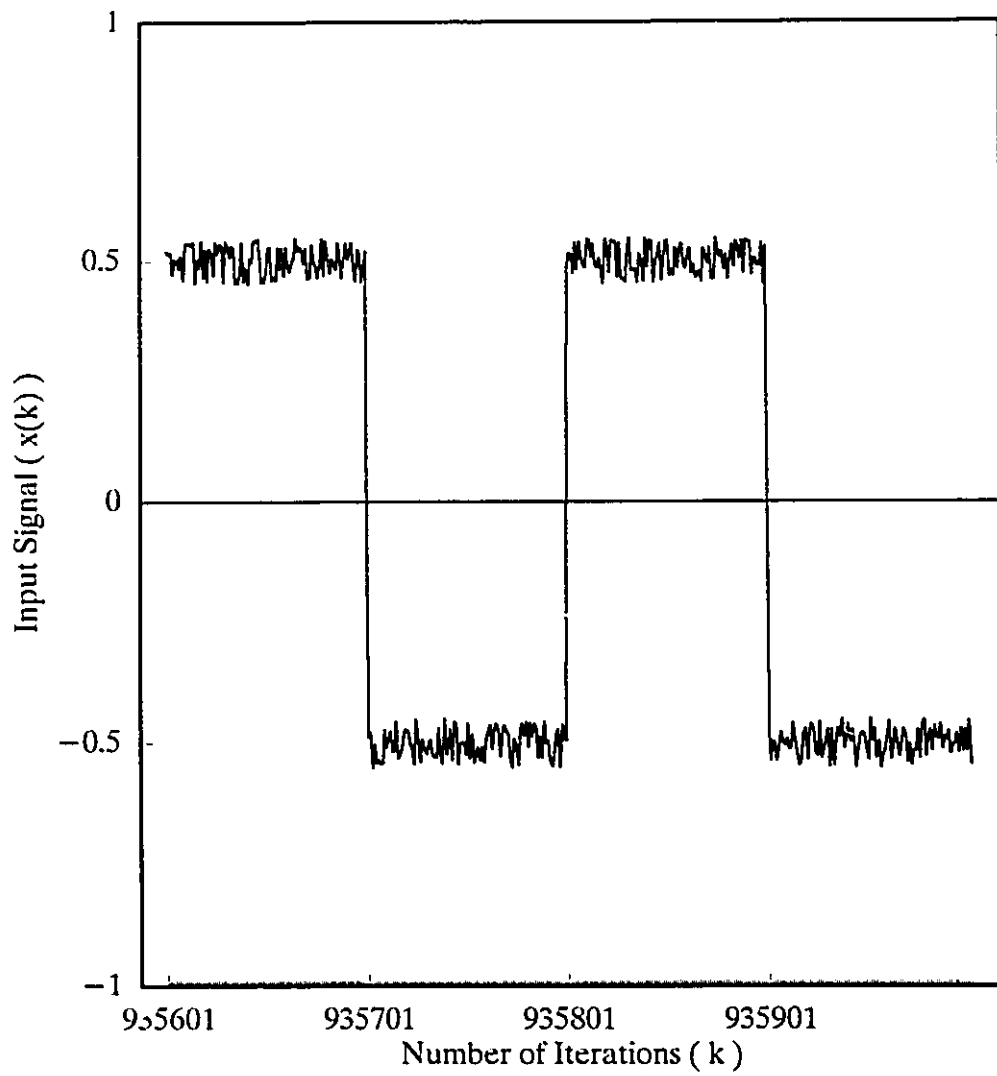
Figs. 4.7e, 4.7f and 4.7g show the converged output signals, $y(k)$, of the LA-FIR digital filter with filter orders of 61, 82, and 121 respectively. In these three cases, the learning rate parameter of coefficients was set to 1.0×10^{-5} , and the adaptive processes were started with zero initial coefficients and delay $d=4$. The adaptive processes converged, respectively, after 1368000, 3458000, and 4920000 iterations. The average squared output errors $e^2(k)$ over the last 1000 iterations for the three cases were,

respectively, 7.9461×10^{-1} , 6.7854×10^{-1} and 4.8140×10^{-2} . The corresponding average squared differences between $y(k)$ and $s(k)$ over the last 200 iterations were, respectively, 3.9135×10^{-2} , 3.3323×10^{-2} and 1.6613×10^{-3} .

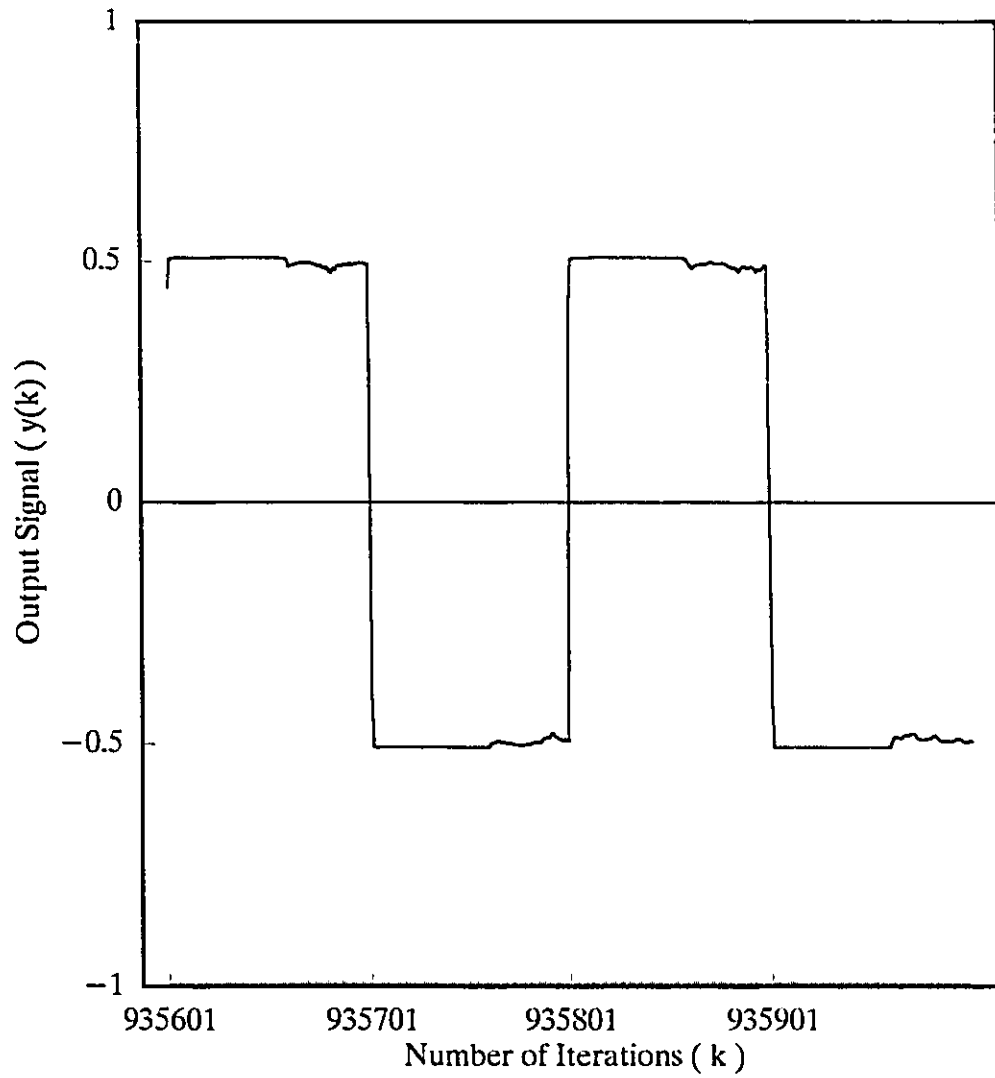
Fig. 4.7h shows the converged output signal, $y(k)$, of the NLMED digital filter with a window length of 3. The average squared difference between this output signal, $y(k)$, and the original input signal, $s(k)$, over the last 200 iterations was 4.8360×10^{-4} . We have also tried window lengths of all the odd numbers from 3 to 21, and the result for the window length of 3 was the best.

Besides the above results, three other sets of simulations of the above two examples were carried out under different initial conditions. For the cases of the NDNA-FIR digital filter, different $g_{i,0}$, g_0 , $\alpha_{i,0}$, and α_0 were used. For the cases of the LA-FIR digital filter, different initial coefficients were used. The average results of the four sets of simulations are summarized in Tables 4.1 and 4.2. In the tables, NOC and NOP represent, respectively, the number of adjustable coefficients, and the number of adjustable parameters (for NDNA-FIR digital filters only). NOI stands for the number of iterations required to reach convergence. E_{sv} and D_{sv} represent, respectively, the average squared output error $e^2(k)$ and the average squared difference between $y(k)$ and $s(k)$.

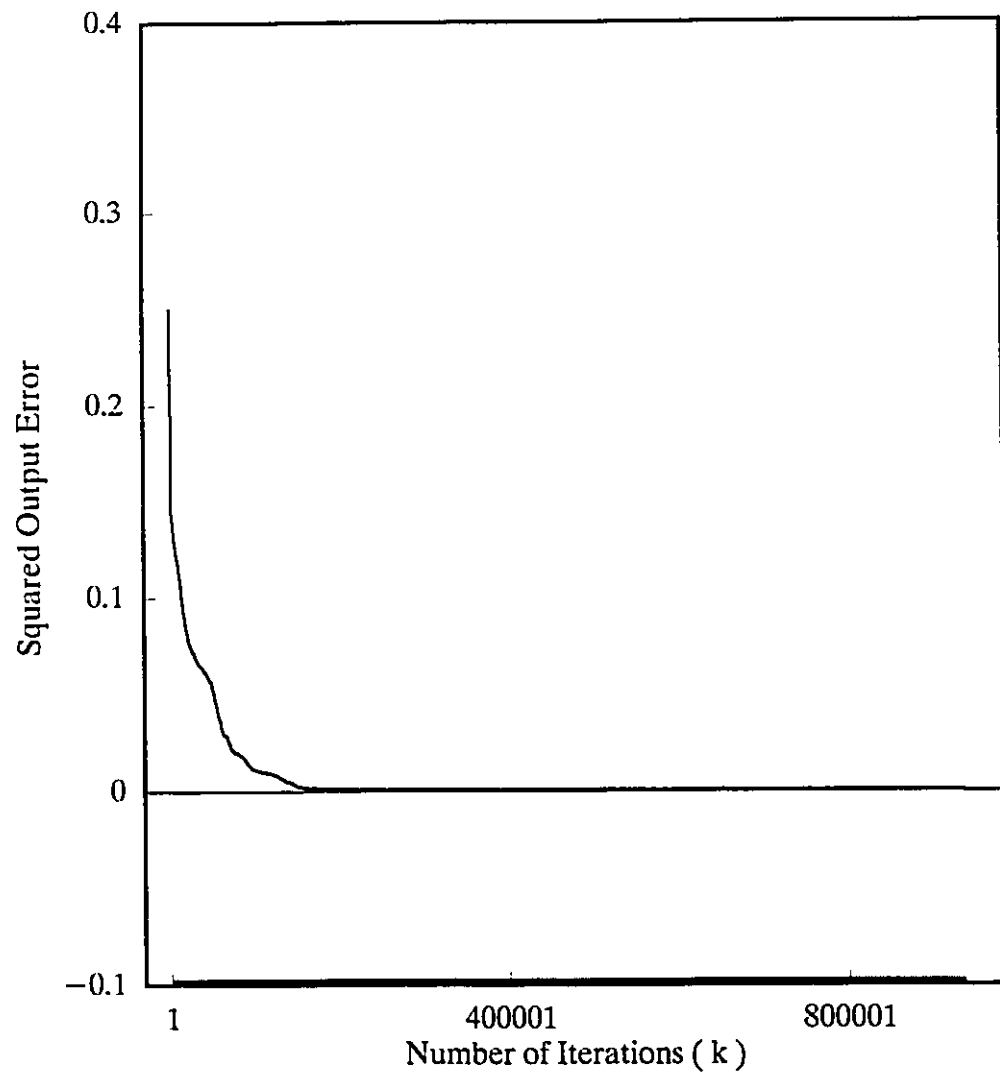
(a) Original signal $s(k)$



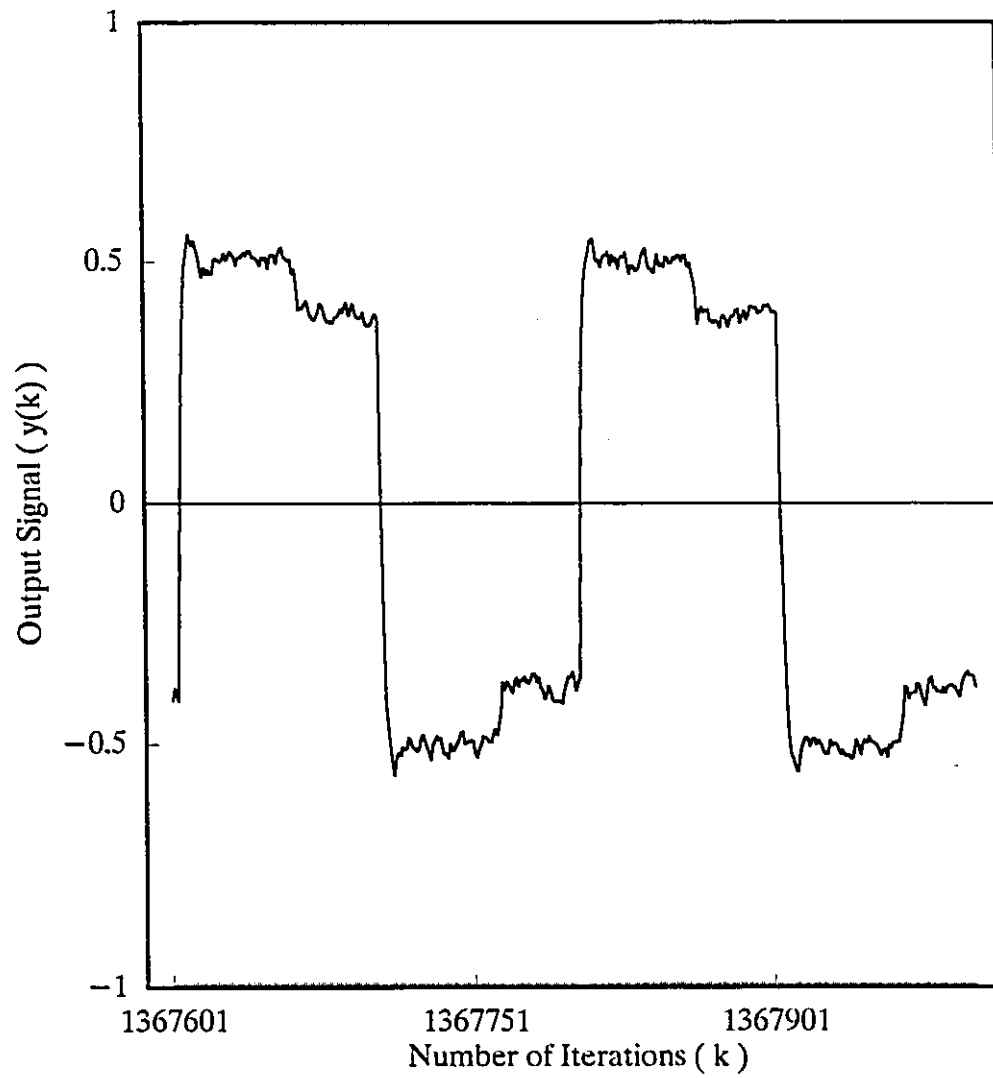
(b) Corrupted input signal $x(k)$



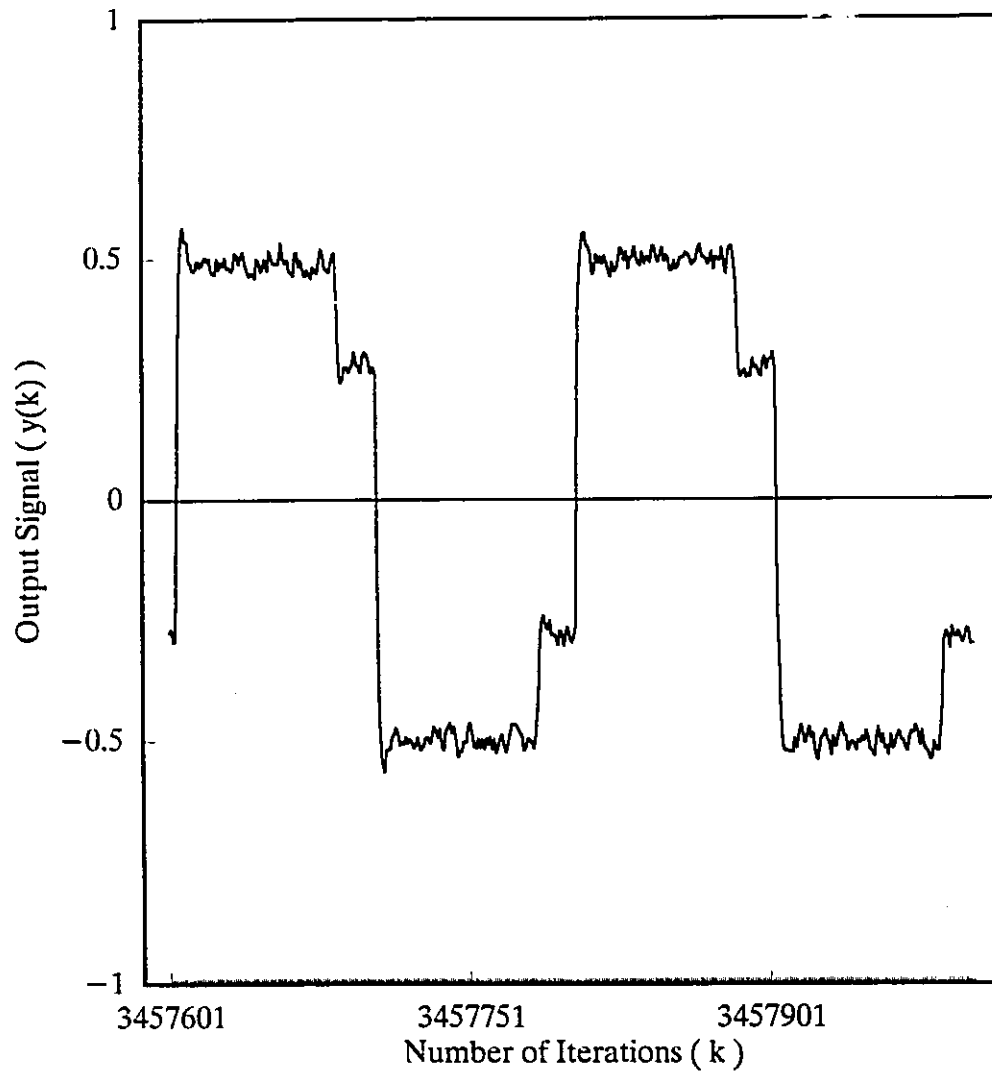
(c) Converged output signal $y(k)$ of NDNA-FIR digital filter



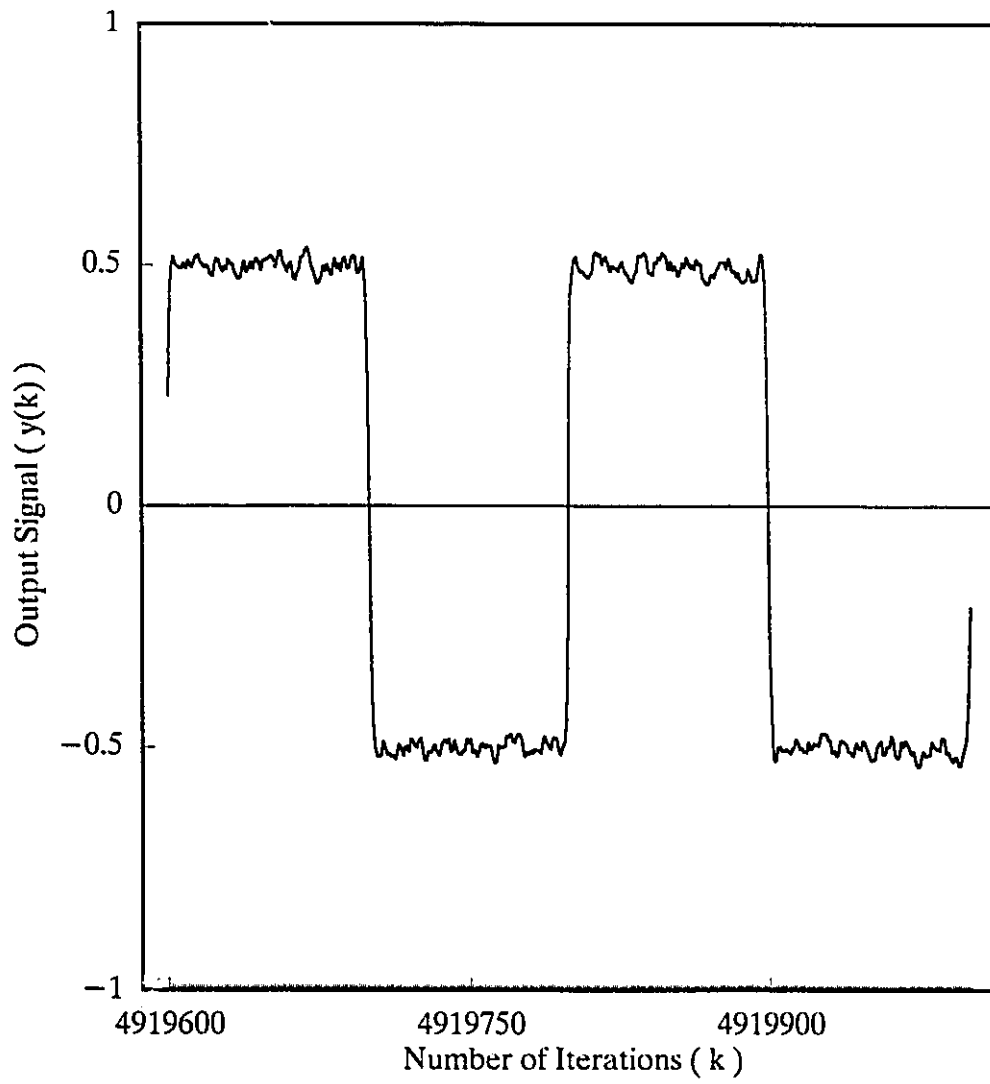
(d) Squared output error $e^2(k)$ of NDNA-FIR digital filter



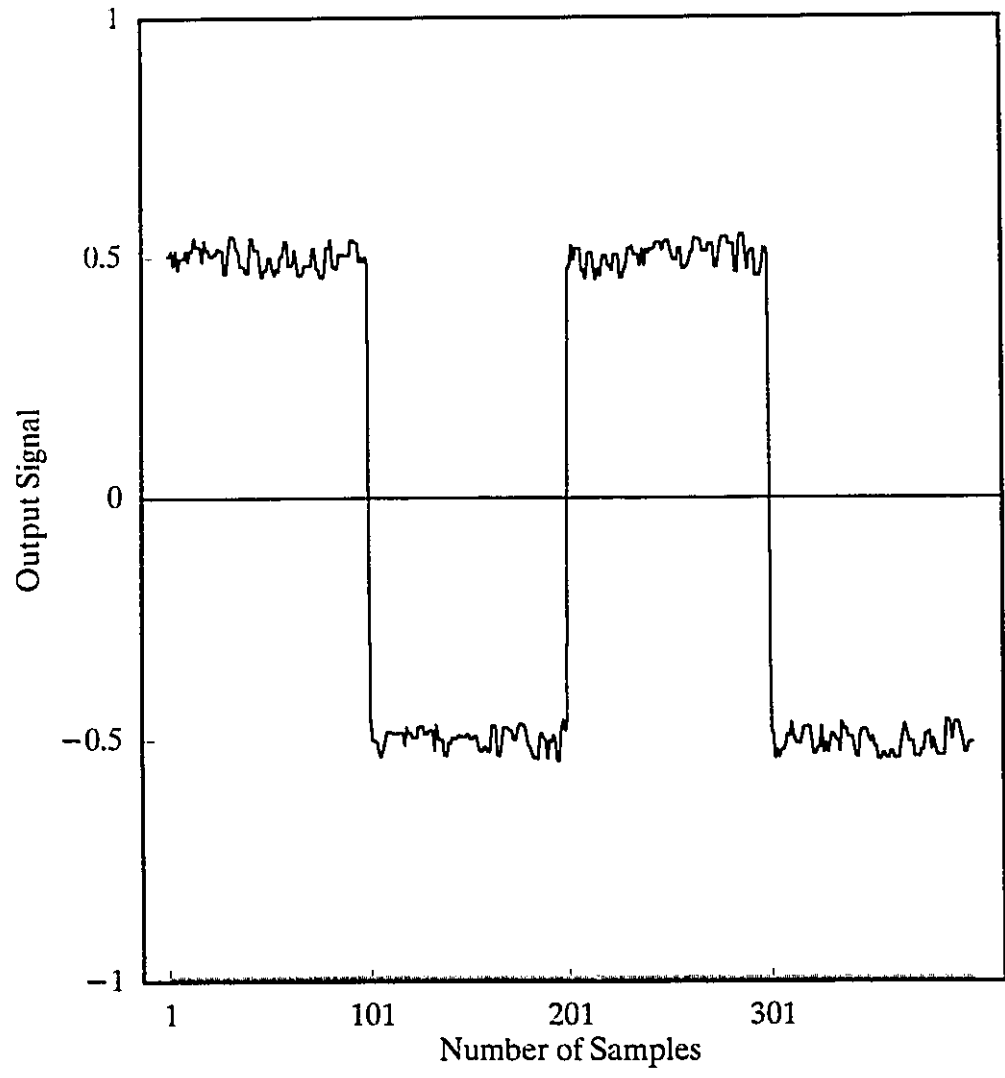
(e) Converged output signal $y(k)$ of LA-FIR digital filter of order 61



(f) Converged output signal $y(k)$ of LA-FIR digital filter of order 82



(g) Converged output signal $y(k)$ of LA-FIR digital filter of order 121



(h) Output signal of NLMED digital filter

Fig. 4.7 Simulation results in Example 2

From the Tables 4.1 and 4.2 we can see that the performances of the NDNA-FIR digital filter with 61 coefficients and 21 parameters are better than those of the LA-FIR digital filters with 61, 82, and 121 coefficients and the NLMED digital filter in terms of E_{av} and D_{av} obtained under the same convergence criterion. The numbers of iterations needed in the NDNA-FIR digital filter are less than those in the LA-FIR digital filters, which means that the NDNA-FIR digital filter is more efficient in reducing the squared output error. The performances of the NDNA-FIR digital filters are much better as compared to those of the LA-FIR digital filters and the NLMED digital filters in the second example in which the original signal has much sharper edges. For signals containing smoother edges (for example, a sinusoidal signal), no obvious differences have been observed among the three types of filtering methods. In the simulations of a NDNA-FIR digital filter, α_0 and $\alpha_{i,0}$ for $i=1$ to 5 should be chosen according to the sharpness of the input signal, and g_0 and $g_{i,0}$ for $i=1$ to 5 should not be too small to avoid limiting the output signal to a false magnitude value. As observed from simulations, the nonlinearity at the output does play a role in adjusting the output amplitude. We also observed from simulations that an added delay of z^{-d} in the NDNA-FIR digital filter would degrade its performances in NOI, E_{av} , and D_{av} . In general, the results become worse as d is getting larger. Hence, d was set to zero in the simulations. Nevertheless, a trivial convergence could hardly happen in the NDNA-FIR digital filter as its output is obtained from a sum of N outputs of N parallel nonlinear sub-filters and coefficient updating is performed only once at every K iterations. For a complete absence of a trivial convergence, the value of at least one of $h_{0,k}(l)$ for $l=1$ to M_0-1 can be set to

nonzero.

Table 4.1 Simulation results of example 1

Structure	NOC	NOP	NOI	E_{av}	D_{av}
NDNA-FIR	61	21	139425	9.5115×10^{-4}	4.2590×10^{-4}
LA-FIR	61		535200	2.6066×10^{-3}	1.6903×10^{-3}
LA-FIR	82		458025	1.7513×10^{-3}	9.5801×10^{-4}
LA-FIR	121		324725	1.3686×10^{-3}	6.0080×10^{-4}
NLMED					5.9251×10^{-4}

Table 4.2 Simulation results of example 2

Structure	NOC	NOP	NOI	E_{av}	D_{av}
NDNA-FIR	61	21	950750	9.3531×10^{-4}	1.9246×10^{-4}
LA-FIR	61		2833000	7.8163×10^{-2}	3.8111×10^{-2}
LA-FIR	82		14796000	6.6789×10^{-2}	3.3001×10^{-2}
LA-FIR	121		3692000	7.6604×10^{-3}	3.0060×10^{-3}
NLMED					4.8360×10^{-4}

4.5 Conclusions

A new nonlinear adaptive FIR digital filter, called NDNA-FIR digital filter, has been presented. The nonlinear sigmoid functions adopted play an important role in cancelling broadband noise while preserving the sharp edges of a signal. The parallel structure adopted here lends itself to high speed implementation. The associated adaptive algorithm has also been developed to minimize the squared output error by adapting the coefficients and parameters. Performance comparisons among the proposed NDNA-FIR digital filter, the LA-FIR digital filter, and the NLMED digital filter indicate that the NDNA-FIR digital filter compares favourably with the LA-FIR digital filter and the NLMED digital filter in terms of convergence speed and the average squared difference between the original signal and the recovered signal.

Chapter 5

A NEW NONLINEAR ADAPTIVE IIR DIGITAL FILTER

The new structure and its adaptive algorithm of a nonlinear adaptive IIR digital filter is presented. With the use of a nonlinear sigmoid function at the output, this IIR digital filter is bounded input bounded output (BIBO) stable. Therefore, it does not need stability monitoring in the adaptive process, which has always been a computational burden and disturbs the adaptive process. The stability analysis has been conducted together with studies on time domain behaviour of filter parameters and coefficients. To improve the convergence speed, the individual parameter adaptation scheme is developed into the adaptive algorithm to optimally adjust each parameter at every iteration. Based on the application in adaptive system identifications of linear IIR digital filters, simulation results obtained show that the new nonlinear adaptive IIR digital filters outperform those of linear adaptive IIR digital filters in terms of convergence speed and final output error residuals.

5.1 Introduction

Much research effort has been directed to the development of adaptive infinite impulse response (IIR) digital filters, because of their reduced computational complexity

compared to adaptive finite impulse response (FIR) digital filters for the same level of performance. However, there are some disadvantages in adaptive IIR digital filters: 1. They become unstable if the poles move outside the unit circle during the adaptive process; 2. Their performance surfaces are generally nonquadratic and may have local minima [7]. While the second disadvantage may vanish by adding filter weights [65], the first one becomes a serious disadvantage which has caused adaptive IIR digital filters to have very limited applications.

There have emerged some techniques to prevent instability by restricting the poles or denominator coefficients of a transfer function. For the adaptive recursive least mean square filter, stability monitoring is carried out by finding the roots of a nonlinear algebraic equation in which its solution complexity increases rapidly with filter order [66]. Also, a parallel form comprised of a number of first-order sections for adaptive IIR filtering has been presented in [67], which provides robust stability monitoring with less complexity than that of the direct form. The stability of the adaptive lattice digital filter can be guaranteed by ensuring that all filter coefficients remain bounded by unit during coefficient adaptations [68]. However, the need to provide a real-time stability check for all these techniques is a significant computational burden in many situations. Of more concern is that the adaptive process may be disturbed so that the search for the minimum value could be misled or the adaptive algorithm would be likely to lock up [69]. As is also known that the family of hyperstable filters, such as the hyperstable adaptive recursive filter (HARF) [69] and the simplified hyperstable adaptive recursive filter

(SHARF) [70], allows for monitoring the filter stability by finding a polynomial which satisfies the strictly positive real condition at all times. Not only it is difficult to find such a polynomial in the absence of specific knowledge about the source model, but also the polynomial found needs to be updated when the source model varies with time. It seems, so far, that the stability problem still remains an issue in linear adaptive IIR digital filtering and more effort is required to develop new approaches featuring implementation feasibility.

Recently nonlinear adaptive digital filtering techniques have received much attention and found many applications such as in [71-75] where problems involved feature nonlinear characteristics. However, it is also of interest to solve linear problems by means of nonlinear approaches. In this paper, a stable nonlinear adaptive IIR digital filter which has an adaptively adjustable nonlinear function at its output is presented. Unlike the linear adaptive IIR digital filters (LAIIRDF) mentioned above, this nonlinear adaptive IIR digital filter (NAIIRDF) does not need any stability monitoring and hence the adaptive process would not be interrupted by the stability problem. It has been proved that this NAIIRDF is stable in the sense that the filter output is always bounded with bounded input signals. The adaptive scheme of individual parameter adaptation [51] is developed for the case of the NAIIRDF structure. Hence, instead of using a fixed convergence constant μ for all the filter coefficients through the whole adaptive process, optimally tailored individual convergence parameters at every single iteration are used to adapt each filter parameter or coefficient. Each individual convergence parameter is

adjusted in real time to be kept optimum for a new set of input samples. The advantages of the structure and adaptive algorithm of the proposed NAIIRDF are demonstrated by applications in adaptive system identifications, where unknown linear IIR systems are to be modelled. Simulation results obtained are very satisfactory in terms of convergence speed and the output error residuals.

5.2 Structure and Adaptive Algorithm of Nonlinear Adaptive IIR Digital Filter (NAIIRDF)

5.2.1 Structure of NAIIRDF

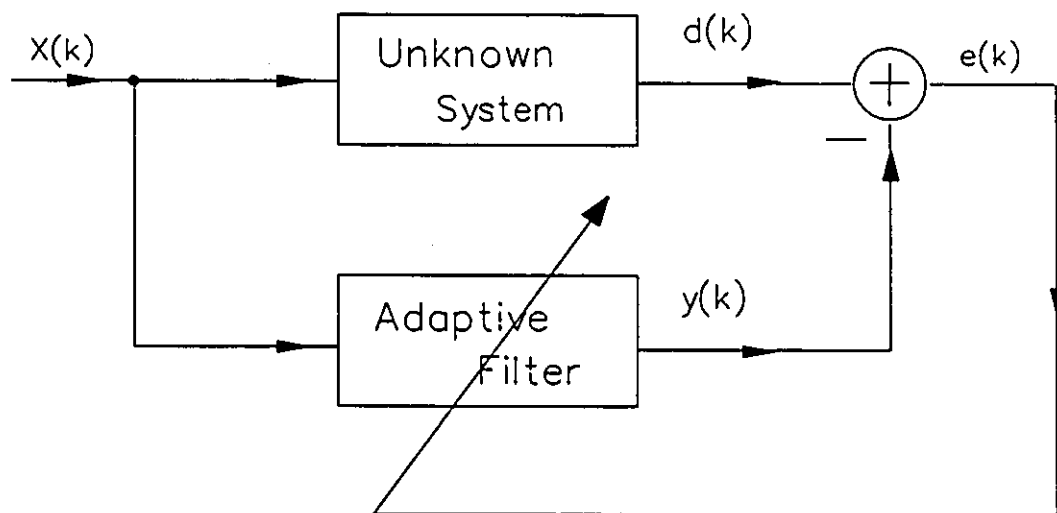


Fig. 5.1 Block diagram of adaptive digital system identification

Fig. 5.1 shows the general block diagram of adaptive system modelling. In this case, the unknown system to be modelled is a linear IIR digital filter. Instead of a linear IIR digital filter structure as shown in Fig. 5.2a, the adaptive digital filter here is constructed by the proposed nonlinear IIR digital filter structure as shown in Fig. 5.2b.

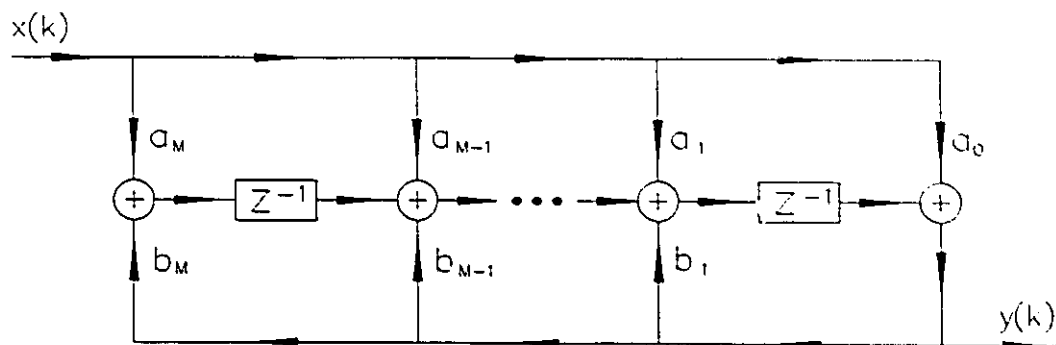
For the linear IIR digital filter structure shown in Fig. 5.2a, the system input and output relationship satisfies the following difference equation

$$y(k) = \sum_{i=0}^M a_i x(k-i) + \sum_{j=1}^M b_j y(k-j) \quad (5.1)$$

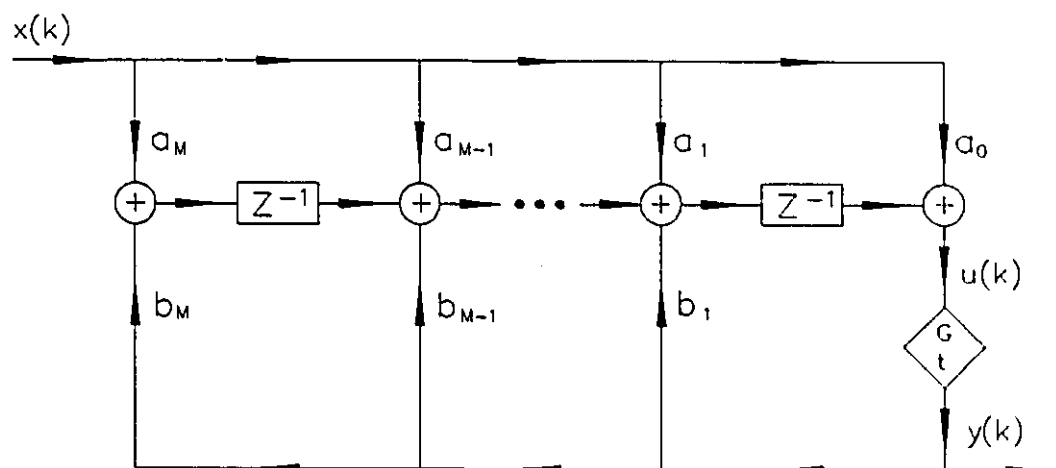
It can be seen from eqn. 5.1 that when the coefficients are not properly selected, the output signal $y(k)$ may increase without bound to infinite when positive feedback occurs. However the proposed nonlinear adaptive IIR digital filter structure in Fig. 5.2b does not have this stability problem. The diamond in Fig. 5.2b represents the nonlinear sigmoid function

$$y = F(u, G, t) = G \frac{1 - e^{-u/t}}{1 + e^{-u/t}} \quad (5.2)$$

where u is the input to the operator and y is the output from the operator as shown in Fig. 5.3. G and t are the parameters of the nonlinear sigmoid function. The corresponding difference equation of the input and output of the structure can be expressed as follows:



(a) Linear IIR digital filter



(b) Nonlinear IIR digital filter

Fig. 5.2 Structures of two IIR digital filters

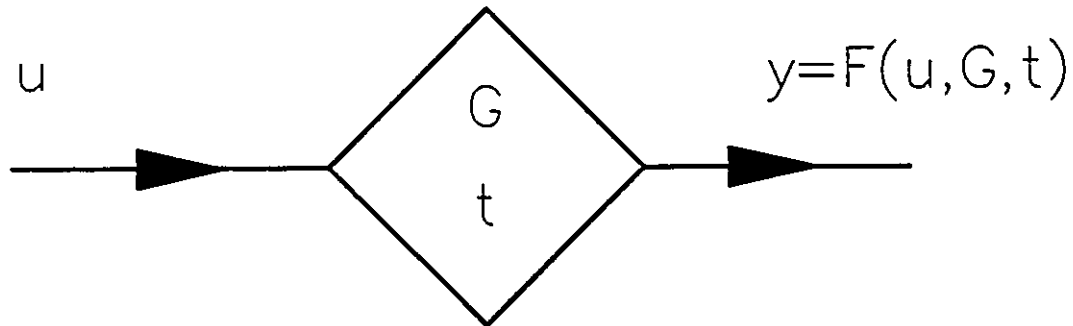


Fig. 5.3 Nonlinear operator of sigmoid function

$$u(k) = \sum_{i=0}^M a_i x(k-i) + \sum_{j=1}^M b_j y(k-j) \quad (5.3)$$

$$\begin{aligned} y(k) &= F(u(k), G, t) \\ &= G \frac{1 - e^{u(k)+t}}{1 + e^{u(k)+t}} \end{aligned} \quad (5.4)$$

Because of the saturation characteristic of the nonlinear function, the output signal will never exceed G . During the adaptive process, the parameters of the nonlinear sigmoid function G and t are updated accordingly together with the filter weights a_i and b_j , for $i = 0, 1, \dots, M$ and $j = 1, 2, \dots, M$, to reduce the output error. The whole adaptive procedure would not be disturbed in contrast to the case of the LAIRDF mentioned above where the poles or denominator coefficients are forced to be projected into a certain region in order to maintain the stability of the adaptive system.

For simplicity in the following discussions, the subscript indexes i , j , and k will always represent integers from 0 to M , 1 to M , and 0 to positive infinite, respectively, unless stated otherwise.

5.2.2 Adaptive Algorithm of NAIIRDF

The proposed nonlinear adaptive IIR digital filter structure is obtained by replacing the "Adaptive Filter" in Fig. 5.1 with the nonlinear IIR digital filter structure shown in Fig. 5.2b. Then, the output signal $y(k)$ of the adaptive digital filter with the input signal $x(k)$ in Fig. 5.1 is

$$\begin{aligned} y(k) &= F(u(k), G(k), t(k)) \\ &= G(k) \frac{1 - e^{u(k)+t(k)}}{1 + e^{u(k)+t(k)}} \end{aligned} \quad (5.5)$$

in which

$$u(k) = \sum_{i=0}^M a_i(k) x(k-i) + \sum_{j=1}^M b_j(k) y(k-j) \quad (5.6)$$

where $G(k)$ and $t(k)$ are the parameter values of the nonlinear sigmoid function at discrete time k , $a_i(k)$ and $b_j(k)$ are the feedforward and feedback coefficient values at discrete time k .

Let $d(k)$ represent the output signal of the unknown system at discrete time k , the instantaneous squared error is

$$e^2(k) = [d(k) - y(k)]^2 \quad (5.7)$$

Considering the LMS algorithm, we define the parameter and coefficient changes as follows:

$$\Delta G(k) = -\mu \frac{\partial e^2(k)}{\partial G} \quad (5.8a)$$

$$\Delta r(k) = -\mu \frac{\partial e^2(k)}{\partial r} \quad (5.8b)$$

$$\Delta a_i(k) = -\mu \frac{\partial e^2(k)}{\partial a_i} \quad (5.8c)$$

and

$$\Delta b_j(k) = -\mu \frac{\partial e^2(k)}{\partial b_j} \quad (5.8d)$$

where μ is the convergence constant. The index k of the parameters and coefficients in eqn. 5.8 after the partial derivative operators is not used because they are assumed to be fixed when partial derivatives are performed. Then we have the corresponding updating equations as

$$G(k+1) = G(k) + \Delta G(k) \quad (5.9a)$$

$$r(k+1) = r(k) + \Delta r(k) \quad (5.9b)$$

$$a_i(k+1) = a_i(k) + \Delta a_i(k) \quad (5.9c)$$

and

$$b_j(k+1) = b_j(k) + \Delta b_j(k) \quad (5.9d)$$

Substituting eqns 5.7 and 5.5 into eqns. 5.8a and 5.8b respectively, we have

$$\Delta G(k) = 2\mu e(k) \frac{\partial y(k)}{\partial G} = 2\mu e(k) F'_G(u(k), G(k), t(k)) \quad (5.10)$$

and

$$\Delta t(k) = 2\mu e(k) \frac{\partial y(k)}{\partial t} = 2\mu e(k) F'_t(u(k), G(k), t(k)) \quad (5.11)$$

Substituting eqns. 5.7, 5.5 and 5.6 into eqns. 5.8c and 5.8d respectively, we get

$$\begin{aligned} \Delta a_i(k) &= 2\mu e(k) \frac{\partial y(k)}{\partial a_i} = 2\mu e(k) \frac{\partial F(u(k), G(k), t(k))}{\partial u(k)} \frac{\partial u(k)}{\partial a_i} \\ &= 2\mu e(k) F'_u(u(k), G(k), t(k)) \frac{\partial u(k)}{\partial a_i} \end{aligned} \quad (5.12)$$

and

$$\begin{aligned} \Delta b_j(k) &= 2\mu e(k) \frac{\partial y(k)}{\partial b_j} = 2\mu e(k) \frac{\partial F(u(k), G(k), t(k))}{\partial u(k)} \frac{\partial u(k)}{\partial b_j} \\ &= 2\mu e(k) F'_u(u(k), G(k), t(k)) \frac{\partial u(k)}{\partial b_j} \end{aligned} \quad (5.13)$$

The partial derivatives F'_G , F'_t , and F'_u can be obtained from eqn. 5.5 as follows:

$$F'_G(u(k), G(k), t(k)) = \frac{1 - e^{u(k)+t(k)}}{1 + e^{u(k)+t(k)}} \quad (5.14)$$

$$F'_i(u(k), G(k), t(k)) = -2G(k) \frac{e^{u(k)+t(k)}}{(1 + e^{u(k)+t(k)})^2} \quad (5.15)$$

and

$$F'_u(u(k), G(k), t(k)) = -2G(k) \frac{e^{u(k)+t(k)}}{(1 + e^{u(k)+t(k)})^2} \quad (5.16)$$

$\partial u(k)/\partial a_i$ and $\partial u(k)/\partial b_j$ in eqns. 5.12 and 5.13 can be obtained as follows:

$$\frac{\partial u(k)}{\partial a_i} = x(k-i) + \sum_{m=1}^M b_m(k) F'_u(u(k-m), G(k-m), t(k-m)) \frac{\partial u(k-m)}{\partial a_i} \quad (5.17)$$

$$\frac{\partial u(k)}{\partial b_j} = y(k-j) + \sum_{m=1}^M b_m(k) F'_u(u(k-m), G(k-m), t(k-m)) \frac{\partial u(k-m)}{\partial b_j} \quad (5.18)$$

5.3 Analyses of Stability and Time Domain Performance of Nonlinear Adaptive IIR Digital Filter (NAIIRDF)

The stability problem has always been a concern of linear adaptive IIR digital filters, in which the output signal may go to infinite once its poles or denominator coefficients move outside the region of stability. However, the NAIIRDF presented in this paper does not have this problem. In fact, its output signal $y(k)$ would not increase without bound if $G(k)$ remains bounded during the adaptive process. In this section, stability analysis is conducted to show that the parameter $G(k)$ would not blow up based on our adaptive algorithm. While we cannot demonstrate the whole picture of this NAIIRDF's behavior during the adaptive process, we can display some interesting and

valuable aspects.

Before we go into the following discussions, we introduce two theorems on sequences and series in advanced calculus [76].

Theorem 1: Suppose $c_n \geq 0$ for all n . If

$$\lim_{n \rightarrow \infty} (c_n)^{1/n} < 1 \quad (5.19)$$

then the series $\sum_n c_n$ converges.

Theorem 2: Let $\{\Phi_k\}$ and $\{\phi_k\}$ be two sequences and related by

$$\Phi_k = \max_{1 \leq n \leq k} \{|\phi_n|\} \quad (5.20)$$

Then the following results are true:

1. Φ_k is a positive and monotone increasing sequence, and $\Phi_k \geq |\phi_k|$
2. If ϕ_k is unbounded, then $\lim_{n \rightarrow \infty} \Phi_k = \infty$
3. If Φ_k is bounded, then ϕ_k is bounded

5.3.1 Performance Analysis of $G(k)$ and $y(k)$

In this section, we will prove that the NAIIRDF is stable in the sense that the output signal $y(k)$ would always be bounded with the bounded input signals $x(k)$ and $d(k)$.

From eqns. 5.7 and 5.5, we obtain

$$\begin{aligned}
 e(k) &= d(k) - y(k) = d(k) - G(k) \frac{1 - e^{u(k)+t(k)}}{1 + e^{u(k)+t(k)}} \\
 &= d(k) - G(k)P(k)
 \end{aligned} \tag{5.21}$$

where

$$P(k) = \frac{1 - e^{u(k)+t(k)}}{1 + e^{u(k)+t(k)}} \tag{5.22}$$

has absolute value always less than 1. From eqns. 5.10 and 5.14, we get the following formula for updating parameter $G(k)$, i.e.

$$G(k+1) = G(k) + 2\mu e(k)P(k) \tag{5.23}$$

In fact, eqn. 5.23 implies a negative feedback adjustment which would prevent $G(k)$ from increasing to infinite. Let us consider the following two cases in detail: (i) $G(k)$ is a positive number with sufficiently large value, and (ii) $G(k)$ is a negative number with sufficiently large absolute value. We will prove for both cases that the absolute value of $G(k+1)$ would not exceed $G(k)$.

(i) $G(k)$ is a positive number with sufficiently large value.

In this case, we have to consider three possibilities, i.e., $P(k) < 0$, $P(k) = 0$, and $P(k) > 0$, respectively.

If $P(k) < 0$, we get

$$e(k) = d(k) - y(k) = d(k) - G(k)P(k) > 0 \tag{5.24a}$$

and

$$G(k+1) = G(k) + 2\mu e(k)P(k) < G(k) \quad (5.24b)$$

If $P(k)=0$, we have

$$G(k+1) = G(k) + 2\mu e(k)P(k) = G(k) \quad (5.25)$$

If $P(k)>0$, we obtain

$$e(k) = d(k) - y(k) = d(k) - G(k)P(k) < 0 \quad (5.26a)$$

and

$$G(k+1) = G(k) + 2\mu e(k)P(k) < G(k) \quad (5.26b)$$

(ii) $G(k)$ is a negative number with sufficiently large absolute value.

Similarly, we also have to consider three possibilities, i.e., $P(k)<0$, $P(k)=0$, and $P(k)>0$, respectively.

If $P(k)<0$, we get

$$e(k) = d(k) - y(k) = d(k) - G(k)P(k) < 0 \quad (5.27a)$$

and

$$G(k+1) = G(k) + 2\mu e(k)P(k) > G(k) \quad (5.27b)$$

If $P(k)=0$, we have

$$G(k+1) = G(k) + 2\mu e(k)P(k) = G(k) \quad (5.28)$$

If $P(k)>0$, we obtain

$$e(k) = d(k) - y(k) = d(k) - G(k)P(k) > 0 \quad (5.29a)$$

and

$$G(k+1) = G(k) + 2\mu e(k)P(k) > G(k) \quad (5.29b)$$

The above analysis states that the parameter $G(k)$ would not increase without bound if the input signal $x(k)$ and the desired signal $d(k)$ are always bounded. Therefore, it is obvious from eqn. 5.5 that the output signal $y(k)$ would stay bounded as well. Hence this NAIIRDF is always stable in the sense of bounded input - bounded output.

5.3.2 Performance Analysis of $t(k)$

In the following, we will prove that $t(k)$ would not increase to infinite independently at the rate of or higher than αk^τ when k goes to infinite, where $\alpha \neq 0$ and $\tau \geq 1$. The word 'independently' excludes the possibility of $t(k)$ increasing to infinite at the same time as one or more other filter coefficients increasing to infinite.

Substituting eqn. 5.15 into eqn. 5.11, we arrive at

$$\Delta t(k) = -4\mu e(k)G(k) \frac{e^{u(k)+t(k)}}{(1 + e^{u(k)+t(k)})^2} \quad (5.30)$$

Then we can obtain

$$\begin{aligned}
r(k+1) &= r(k) - \Delta r(k) = r(k-1) + \Delta r(k-1) - \Delta r(k) \\
&\quad \dots\dots\dots \\
&= r(0) - \sum_{n=0}^k 4\mu e(n)G(n) \frac{e^{u(n)+r(n)}}{(1 + e^{u(n)+r(n)})^2}
\end{aligned} \tag{5.31}$$

Because $G(k)$ is always bounded, then $e(k)$ would also remain bounded during the adaptive process as can be seen from eqn. 5.21. Therefore, we have

$$|e(k)| \leq E \tag{5.32a}$$

and

$$|G(k)| \leq G \tag{5.32b}$$

where E and G are certain positive real numbers. Based on eqns. 5.31 and 5.32, the following relationship can be obtained

$$\begin{aligned}
|r(k+1)| &\leq |r(0)| + \sum_{n=0}^k 4\mu |e(n)| |G(n)| \frac{e^{u(n)+r(n)}}{(1 + e^{u(n)+r(n)})^2} \\
&\leq |r(0)| + 4\mu EG \sum_{n=0}^k \frac{e^{u(n)+r(n)}}{(1 + e^{u(n)+r(n)})^2}
\end{aligned} \tag{5.33}$$

Let us assume

$$T(k) = \sum_{n=0}^k \frac{e^{u(n)+r(n)}}{(1 + e^{u(n)+r(n)})^2} = \sum_{n=0}^k c_n \tag{5.34}$$

where

$$c_n = \frac{e^{u(n)+r(n)}}{(1 + e^{u(n)+r(n)})^2} \tag{5.35}$$

In the following, the reductio ad absurdum method is used to show that $T(k)$ is bounded during the adaptive process under certain conditions. Assume $t(k)$ would increase independently without bound to infinite at the rate of αk^τ ($\alpha \neq 0$ and $\tau \geq 1$), i.e.,

$$\lim_{k \rightarrow \infty} t(k) = \lim_{k \rightarrow \infty} \alpha k^\tau = \infty \quad \text{when } \alpha \neq 0 \text{ and } \tau \geq 1 \quad (5.36)$$

On the other hand, we have from eqn. 5.35 that

$$\lim_{n \rightarrow \infty} (c_n)^{1/n} = \lim_{n \rightarrow \infty} \left[\frac{e^{u(n)+t(n)}}{(1 + e^{u(n)+t(n)})^2} \right]^{1/n} \quad (5.37)$$

When $\alpha > 0$ and $\tau \geq 1$, eqn. 5.37 becomes

$$\begin{aligned} \lim_{n \rightarrow \infty} \left[\frac{e^{u(n)+t(n)}}{(1 + e^{u(n)+t(n)})^2} \right]^{1/n} &= \lim_{n \rightarrow \infty} (e^{-t(n)})^{1/n} \\ &= \lim_{n \rightarrow \infty} (e^{-\alpha n^\tau})^{n^{-1}} \leq e^{-\alpha} < 1 \end{aligned} \quad (5.38a)$$

Otherwise, when $\alpha < 0$ and $\tau \geq 1$, eqn. 5.37 becomes

$$\begin{aligned} \lim_{n \rightarrow \infty} \left[\frac{e^{u(n)+t(n)}}{(1 + e^{u(n)+t(n)})^2} \right]^{1/n} &= \lim_{n \rightarrow \infty} (e^{t(n)})^{1/n} \\ &= \lim_{n \rightarrow \infty} (e^{\alpha n^\tau})^{n^{-1}} \leq e^\alpha < 1 \end{aligned} \quad (5.38b)$$

Based on Theorem 1, we have

$$\lim_{k \rightarrow \infty} T(k) = \lim_{k \rightarrow \infty} \sum_{n=0}^k c_n = \sum_n c_n < \infty \quad \text{when } \alpha \neq 0 \text{ and } \tau \geq 1 \quad (5.39)$$

It is then obvious from eqn. 5.33 that $t(k)$ would be bounded when $\alpha \neq 0$ and $\tau \geq 1$. This conclusion conflicts with the assumption we made previously that $t(k)$ would go to

infinite without bound at the rate of αk^τ ($\alpha \neq 0$ and $\tau \geq 1$). It can also be observed from the above proof that $t(k)$ would not increase to infinite at a rate higher than αk^τ ($\alpha \neq 0$ and $\tau \geq 1$).

5.3.3. Performance Analysis of $a_i(k)$ and $b_j(k)$

In this section, we can first obtain the following two properties:

- (i) $a_i(k)$ would not increase to infinite independently because of the blowing up of $\partial u(k)/\partial a_i$.
- (ii) $a_i(k)$ would not increase to infinite independently at the rate of or higher than αk^τ when k goes to infinite, where $\alpha \neq 0$ and $\tau \geq 1$.

Substituting eqn. 5.16 into eqn. 5.12, we can obtain

$$\Delta a_i(k) = -4\mu e(k)G(k) \frac{e^{u(k)+t(k)}}{(1 + e^{u(k)+t(k)})^2} \frac{\partial u(k)}{\partial a_i} \quad (5.40)$$

where $\partial u(k)/\partial a_i$ can be calculated from eqn. 5.17. In the following, we will show that $a_i(k)$ would not increase without bound independently because of the blowing up of $\partial u(k)/\partial a_i$. The reductio ad absurdum method is still used to serve this purpose.

Assume that $a_i(k)$ would go to infinite independently because of the blowing up of $\partial u(k)/\partial a_i$, and hence as a result, $u(k)$ would go to infinite and F'_u would go to 0. For simplicity, we use ϕ_k^i and $F'_u(k)$ to represent $\partial u(k)/\partial a_i$ and $F'_u(u(k), G(k), t(k))$ respectively, then eqn. 5.17 can be rewritten as

$$\phi_k^i = x(k-i) - \sum_{m=1}^M b_m(k) F_u^-(k-m) \phi_{k-m}^i \quad (5.41)$$

Let us construct a sequence Φ_k^i such that

$$\Phi_k^i = \max_{1 \leq n \leq k} \{ |\phi_n^i| \} = |\phi_{n_k}^i| \quad (5.42)$$

where n_k is a certain index number between and including 1 and k . Based on Theorem 2, if $\{\Phi_k^i\}$ can be proved bounded, then $\{\phi_k^i\}$ is bounded. Substituting eqn. 5.41 into eqn. 5.42, we have

$$\begin{aligned} \Phi_k^i &= |\phi_{n_k}^i| = |x(n_k-i) + \sum_{m=1}^M b_m(n_k) F_u^-(n_k-m) \phi_{n_k-m}^i| \\ &\leq |x(n_k-i)| + \sum_{m=1}^M |b_m(n_k)| |F_u^-(n_k-m)| |\phi_{n_k-m}^i| \end{aligned} \quad (5.43)$$

Based on Theorem 2.1, we can obtain

$$\Phi_k^i \leq |x(n_k-i)| + \sum_{m=1}^M |b_m(n_k)| |F_u^-(n_k-m)| \Phi_{k-m}^i \quad (5.44)$$

Because of the assumption we made and Theorem 2.2, there must exist a positive integer number N , such that when $k > N$ we have

$$\frac{|x(n_k-i)|}{\Phi_{k-1}^i} < \frac{1}{M+2} \quad (5.45a)$$

$$|b_1(n_k)| |F_u^-(n_k-1)| < \frac{1}{M+2} \quad (5.45b)$$

and

$$|b_m(n_k)| F'_u(n_k-m) \frac{\Phi_{k-1}^i}{\Phi_{k-1}^i} < \frac{1}{M+2} \quad \text{for } m = 2, \dots, M \quad (5.45c)$$

Dividing two sides of eqn. 5.44 by Φ_{k-1}^i and using eqn. 5.45, we arrive at

$$\begin{aligned} \frac{\Phi_k^i}{\Phi_{k-1}^i} &\leq \frac{|x(n_k-i)|}{\Phi_{k-1}^i} + |b_1(n_k)| F'_u(n_k-i) + \sum_{m=2}^M |b_m(n_k)| F'_u(n_k-m) \frac{\Phi_{k-1}^i}{\Phi_{k-1}^i} \quad (5.46) \\ &\leq (M+1)/(M+2) < 1 \quad \text{when } k > n \end{aligned}$$

From eqn. 5.46 we can see that $\{\Phi_k^i\}$ is a monotone decreasing sequence starting from a certain number N with low bound 0. Hence, $\{\Phi_k^i\}$ is bounded and therefore $\{\phi_k^i\}$ is bounded based on Theorem 2.3. This conclusion conflicts with the assumption that $\partial u(k)/\partial a_i$ (ϕ_k^i) would go to infinite to cause $a_i(k)$ to blow up. Then the first property exists.

If $\partial u(k)/\partial a_i$ stays bounded, then the transient performance of $a_i(k)$ would be similar to $t(k)$, as can be observed by comparing eqn. 5.40 with eqn. 5.30. Hence, the second property is true because the first property exists.

By comparing eqns. 5.13 and 5.18 with eqns. 5.12 and 5.17 respectively, we can obtain that the transient behaviour of $b_j(k)$ is similar to that of $a_i(k)$. Therefore, the following two properties hold true for $b_j(k)$ as well:

- (i) $b_j(k)$ would not increase to infinite independently because of the blowing up of $\partial u(k)/\partial b_j$.
- (ii) $b_j(k)$ would not increase to infinite independently at the rate of or higher than αk^τ when k goes to infinite, where $\alpha \neq 0$ and $\tau \geq 1$.

In summary, the above discussions can be outlined in the following statements:

- (i) The NAIIRDF is stable in the sense that the output signal $y(k)$ is always bounded with the bounded input signals $x(k)$ and $d(k)$. The reason for this is that the parameter $G(k)$ is always bounded during the adaptive process.
- (ii) The parameters $t(k)$, $a_i(k)$, and $b_j(k)$ would not increase to infinite independently at the rate of or higher than αk^τ when k goes to infinite, where $\alpha \neq 0$ and $\tau \geq 1$.
- (iii) $a_i(k)$ and $b_j(k)$ would not increase to infinite independently because of the blowing up of $\partial u(k)/\partial a_i$ and $\partial u(k)/\partial b_j$, respectively.

While only very conservative conclusions have been proved through theoretical analyses due to nonlinearity involved, simulation results have shown that this NAIIRDF works well under different circumstances, as can be seen in Section 5.5. In practical situations, it is usually possible to estimate the magnitude of the desired signal $d(k)$. Therefore, when a sufficiently large $G(0)$ is selected such that $G(0) \gg \max \{ |d(k)|, k=0,1,2,\dots\}$, the output signal $y(k)$ has little chance to be limited by the nonlinear saturation function. This is because the clipped output signal would mean a bigger error between $d(k)$ and $y(k)$ while the adaptive algorithm is aimed to reduce the difference between $d(k)$ and $y(k)$ in each step. In fact, it did not ever happen through our extensive simulation experiments that filter parameters and coefficients blew up or the filter output equalled the limiting value of the nonlinear sigmoid function.

5.4 Individual Parameter Adaptation Scheme for Nonlinear Adaptive IIR Digital Filter (NAIIRDF)

Conventional adaptation techniques are using the fixed convergence constant μ , which controls the convergence rate of the filter coefficients but also determines the final excess mean square error as compared to the optimal Wiener solution. A trade-off is usually necessary to select a constant μ so that the convergence speed is not too slow and also the final misadjustment is not too large. However, this does not necessarily work well in all the cases. [57] and [77-81] proposed an adaptation scheme with individual convergence factors which are optimally tailored to adapt individual filter coefficients for the recursive-like adaptive filters. This idea can also be applied to the NAIIRDF structure to adjust the individual convergence parameters in real time, so that their values are kept optimum for a new set of input samples to reduce the squared output error $e^2(k)$ more efficiently.

Let us rewrite eqn. 5.8 as follows

$$\Delta G(k) = -\mu_G(k) \frac{\partial e^2(k)}{\partial G} \quad (5.47a)$$

$$\Delta r(k) = -\mu_r(k) \frac{\partial e^2(k)}{\partial r} \quad (5.47b)$$

$$\Delta a_i(k) = -\mu_{a_i}(k) \frac{\partial e^2(k)}{\partial a_i} \quad (5.47c)$$

and

$$\Delta b_j(k) = -\mu_{b_j}(k) \frac{\partial e^2(k)}{\partial b_j} \quad (5.47d)$$

Based on the Taylor series expansion, the error $e(k+1)$ at the $(k+1)$ th step can be approximated in terms of the first order derivatives of $e(k)$ as

$$\begin{aligned} e(k+1) = e(k) &+ \sum_{i=0}^M \frac{\partial e(k)}{\partial a_i} \Delta a_i(k) + \sum_{j=1}^M \frac{\partial e(k)}{\partial b_j} \Delta b_j(k) \\ &+ \frac{\partial e(k)}{\partial G} \Delta G(k) + \frac{\partial e(k)}{\partial r} \Delta r(k) \end{aligned} \quad (5.48)$$

Substituting eqn. 5.47 into eqn. 5.48 and squaring, we have

$$\begin{aligned} e^2(k+1) = e^2(k) &[1 - 2 \sum_{i=0}^M \mu_{a_i}(k) (\partial e(k) / \partial a_i)^2 \\ &- 2 \sum_{j=1}^M \mu_{b_j}(k) (\partial e(k) / \partial b_j)^2 - 2 \mu_G(k) (\partial e(k) / \partial G)^2 \\ &- 2 \mu_r(k) (\partial e(k) / \partial r)^2] \end{aligned} \quad (5.49)$$

Based on the concept of steepest descent, a suitable set of parameters and coefficients are found to yield the minimum value of $e^2(k+1)$ by taking partial derivatives with respect to each convergence parameter and they are set to zero as follows:

$$\frac{\partial e^2(k+1)}{\partial \mu_G} = 0 \quad (5.50a)$$

$$\frac{\partial e^2(k+1)}{\partial \mu_r} = 0 \quad (5.50b)$$

$$\frac{\partial e^2(k+1)}{\partial \mu_{a_i}} = 0 \quad (5.50c)$$

$$\frac{\partial e^2(k+1)}{\partial \mu_{b_j}} = 0 \quad (5.50d)$$

Substituting eqn. 5.49 into eqn. 5.50 and knowing that $e(k)$, $\partial e(k)/\partial G$, $\partial e(k)/\partial t$, $\partial e(k)/\partial a_i$ and $\partial e(k)/\partial b_j$ would not equal zero at the same time (otherwise, no adaptation is necessary any more), then we have

$$\begin{aligned} & \sum_{i=0}^M \mu_{a_i}(k) (\partial e(k)/\partial a_i)^2 + \sum_{j=1}^M \mu_{b_j}(k) (\partial e(k)/\partial b_j)^2 \\ & + \mu_G(k) (\partial e(k)/\partial G)^2 + \mu_t(k) (\partial e(k)/\partial t)^2 = 0.5 \end{aligned} \quad (5.51)$$

Eqn. 5.51 is actually a constraint on the optimum values of the convergence parameters.

For an easy realization of the adaptive algorithm, we set

$$\mu_{a_i}(k) = \mu_a(k) \quad (5.52a)$$

$$\mu_{b_j}(k) = \mu_b(k) \quad (5.52b)$$

and

$$\mu_G(k) (\partial e(k)/\partial G)^2 = 0.5p_G \quad (5.53a)$$

$$\mu_t(k) (\partial e(k)/\partial t)^2 = 0.5p_t \quad (5.53b)$$

$$\sum_{i=0}^M \mu_{a_i}(k) (\partial e(k)/\partial a_i)^2 = 0.5p_a \quad (5.53c)$$

$$\sum_{j=1}^M \mu_{b_j}(k) (\partial e(k)/\partial b_j)^2 = 0.5p_b \quad (5.53d)$$

where p_G , p_t , p_a and p_b are positive numbers which satisfy

$$p_G + p_t + p_a + p_b = 1.0 \quad (5.54)$$

Then eqn. 5.51 can be split into the following equations

$$\mu_G(k) = \frac{0.5p_G}{(\partial e(k)/\partial G)^2} \quad (5.55a)$$

$$\mu_t(k) = \frac{0.5p_t}{(\partial e(k)/\partial t)^2} \quad (5.55b)$$

$$\mu_a(k) = \frac{0.5p_a}{\sum_{i=0}^M (\partial e(k)/\partial a_i)^2} \quad (5.55c)$$

$$\mu_b(k) = \frac{0.5p_b}{\sum_{j=1}^M (\partial e(k)/\partial b_j)^2} \quad (5.55d)$$

Arguing along the same lines as before, eqns. 5.55a and 5.55b can be further expressed as

$$\mu_G(k) = \frac{0.5p_G}{F_G^2(u(k), G(k), t(k))} \quad (5.56)$$

$$\mu_t(k) = \frac{0.5p_t}{F_t^2(u(k), G(k), t(k))} \quad (5.57)$$

Neglecting the recursive parts in eqns. 5.17 and 5.18 to simplify the algorithm without significant performance degradation, eqns. 5.55c and 5.55d can be represented as follows:

$$\mu_a(k) = \frac{0.5p_a}{F_u^2(u(k), G(k), t(k)) \sum_{m=0}^M x^2(k-m)} \quad (5.58)$$

$$\mu_b(k) = \frac{0.5p_b}{F_u^2(u(k), G(k), t(k)) \sum_{m=1}^M y^2(k-m)} \quad (5.59)$$

5.5 Simulation Results

In this section, simulation results are given for the application in adaptive system identifications of linear IIR digital filters. For the individual parameter adaptation scheme formulated in eqns. 5.56-5.59, the denominators of these equations may cause the overflow problem when they become very small or even zero. In order to deal with this problem, a positive bias constant σ has been introduced in our simulations which changes eqns. 5.56-5.59 into the following equations

$$\mu_G(k) = \frac{0.5p_G}{F_G^2(u(k), G(k), t(k)) + \sigma} \quad (5.60)$$

$$\mu_t(k) = \frac{0.5p_t}{F_t^2(u(k), G(k), t(k)) + \sigma} \quad (5.61)$$

$$\mu_a(k) = \frac{0.5p_a}{(F_u^2(u(k), G(k), t(k)) + \sigma) \left(\sum_{m=0}^M x^2(k-m) + \sigma \right)} \quad (5.62)$$

$$\mu_b(k) = \frac{0.5p_b}{(F_u^2(u(k), G(k), t(k)) + \sigma) \left(\sum_{m=1}^M y^2(k-m) + \sigma \right)} \quad (5.63)$$

Because of neglect of the higher order terms when performing the Taylor series expansion in eqn. 5.48 and neglect of the recursive parts in eqns. 5.58 and 5.59 as well as introduction of the bias constant σ into eqns. 5.60-5.63, the constraint of eqn. 5.54 does not actually hold, and hence p_G , p_t , p_x and p_b may be chosen without satisfying eqn. 5.54 but yet producing good simulation results.

For comparison studies, two groups of examples are first given in the following. In each group, both the NAIIRDF and the LAIRDF were used to model the same linear IIR systems. The input signal used is white noise with zero mean value and variance of 1/12.

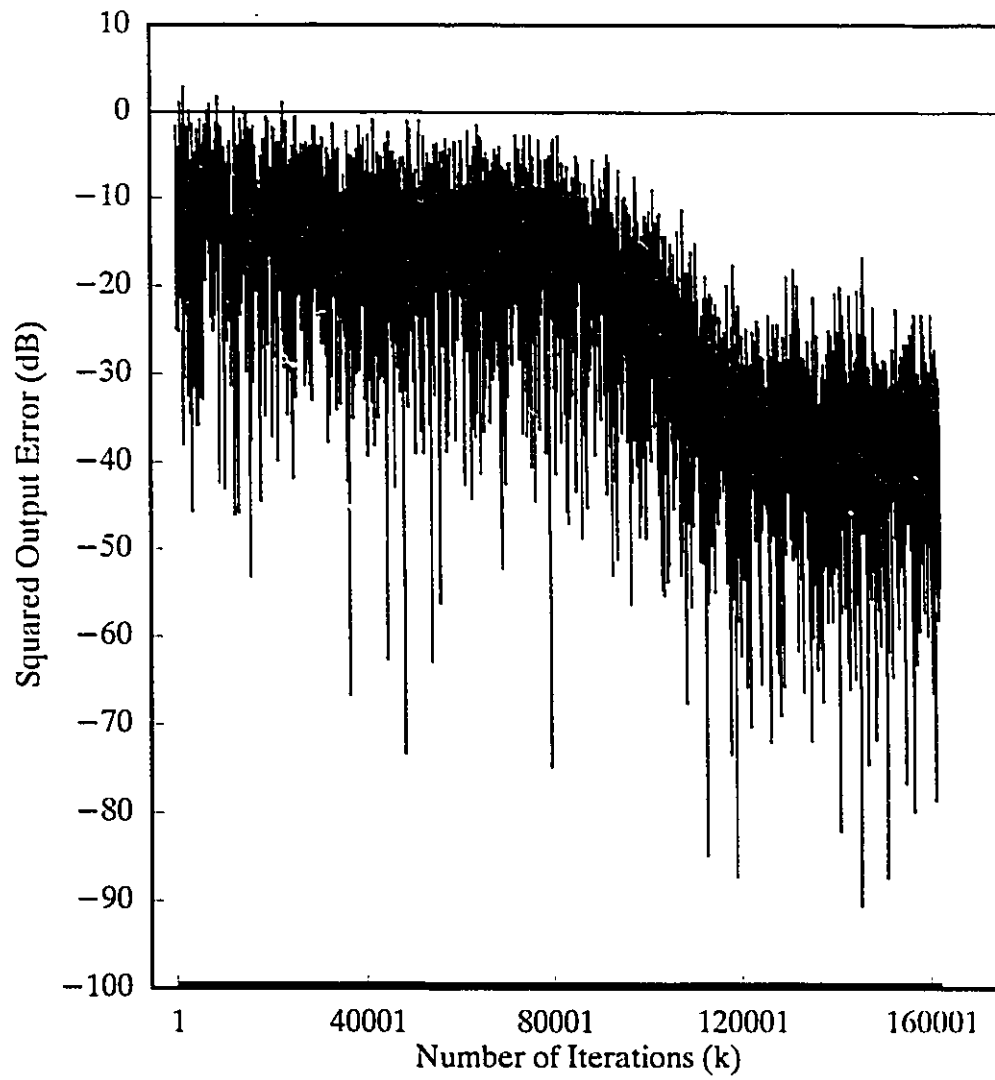
Example 1: A second order linear IIR filter

$$H(z^{-1}) = \frac{1.2 + 0.7z^{-1} - 1.0z^{-2}}{1.0 - 1.7z^{-1} + 0.89z^{-2}}$$

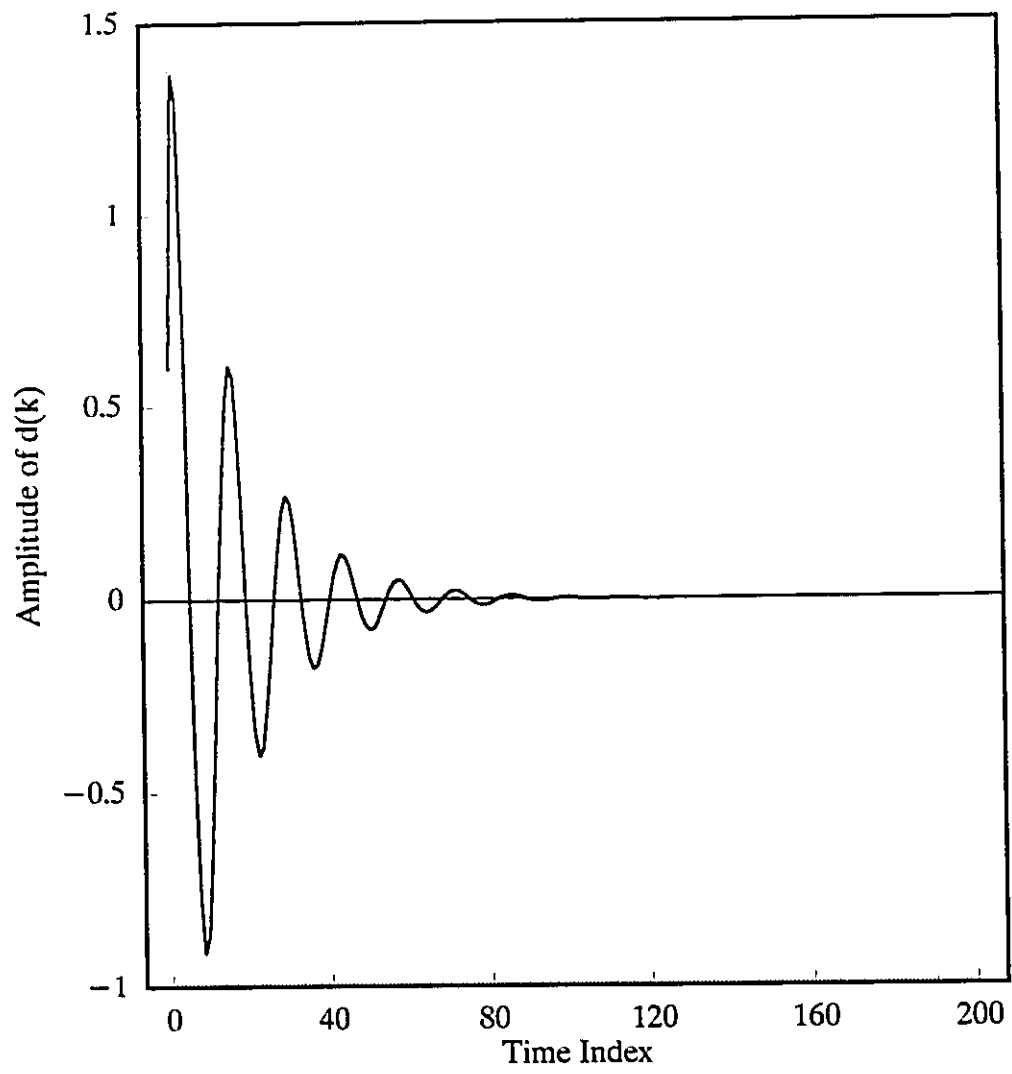
is to be modelled in this example. Its poles ($0.85 \pm 0.41j$) locate quite near to the unit circle. Simulation results obtained by using the NAIIRDF are shown in Fig. 5.4. Fig. 5.4a shows the squared output error (in dB) curve after 162160 iterations when the summation of squared output error over last 20 steps fell below 1.0×10^{-11} . The impulse responses of both the unknown plant and the adaptive filter after convergence are shown in Fig. 5.4b and Fig. 5.4c, respectively. The average squared error between the two

responses was 2.4545×10^{-4} over those 200 samples. The initial parameter $G(0)$ was set to 23.0 and $t(0)$ was set to a very small random number with an absolute value of less than 1.0×10^{-3} . The initial coefficients of $a_i(k)$ and $b_j(k)$ were assigned random numbers with their absolute values less than 0.5. The step parameters p_G , p_t , p_a and p_b were selected as 1.125×10^{-3} , 7.5×10^{-4} , 3.75×10^{-4} , and 3.75×10^{-4} respectively and σ was set to 10.0. We can see from Fig. 5.4 that the adaptive process converged faithfully. The impulse response of the adaptive filter after convergence resembles closely that of the unknown plant in shape as is stemmed from the small output error residuals finally obtained.

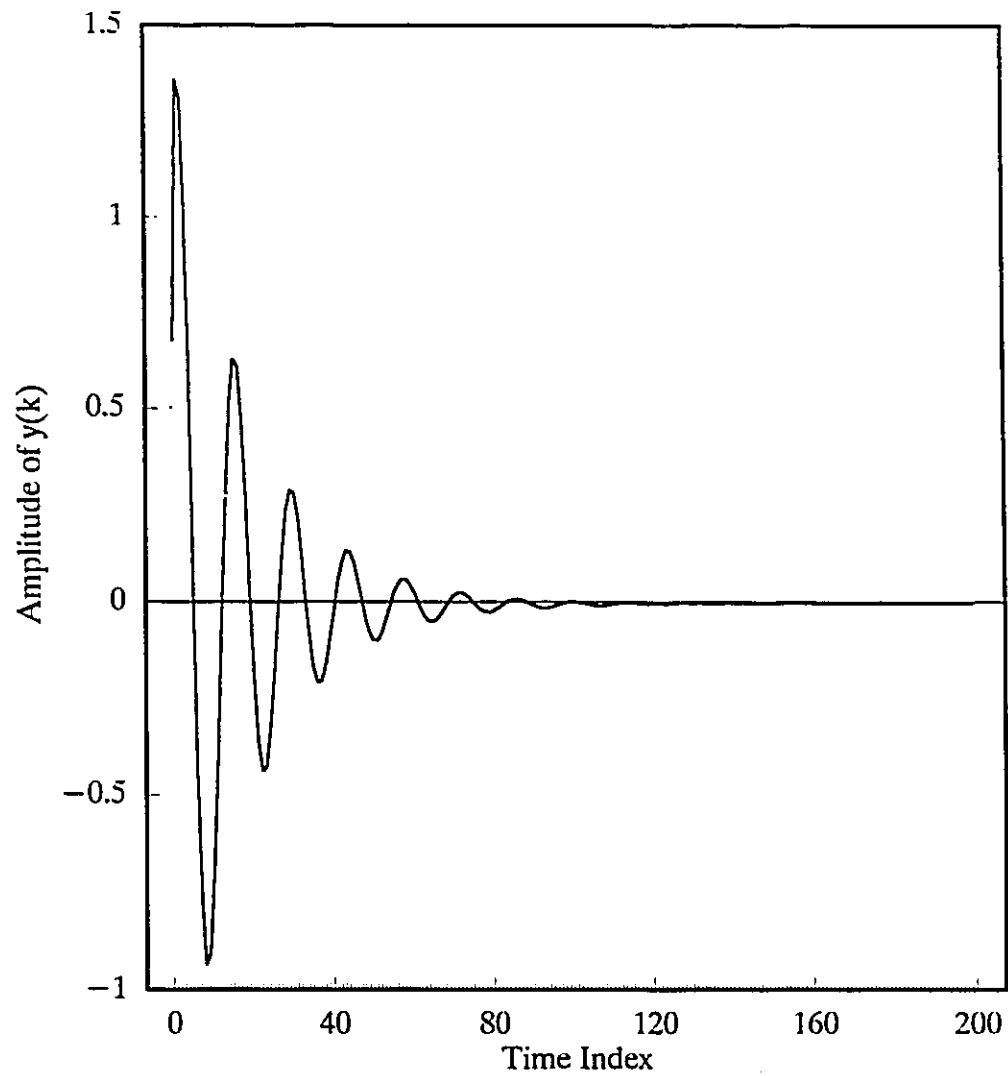
Fig. 5.5 shows the corresponding simulation results obtained by using the LAIRDF. Because the poles are quite near to the unit circle, the convergence parameter should be carefully selected so that the adaptive process could converge well. In this example, we set the convergence constant equal to 1.0×10^{-5} . Fig. 5.5a shows the squared output error (in dB) curve after 797620 iterations when the summation of squared output error of last 20 steps fell below 1.0×10^{-11} . The impulse response of the unknown plant is shown in Fig. 5.4b and that of the adaptive filter after convergence is shown in Fig. 5.5b. The average squared error between the two responses was 1.6666×10^{-4} over those 200 samples.



(a) Squared output error

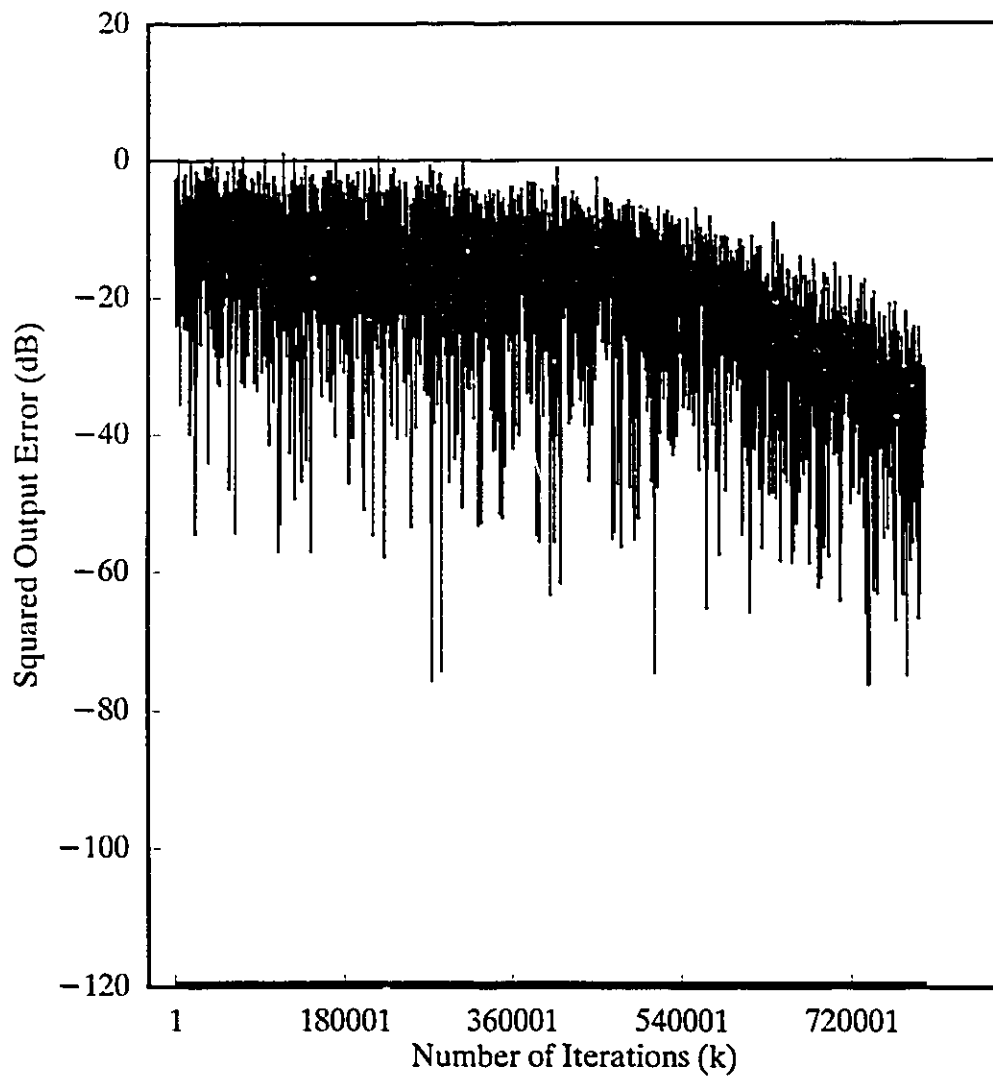


(b) Impulse response of the unknown plant

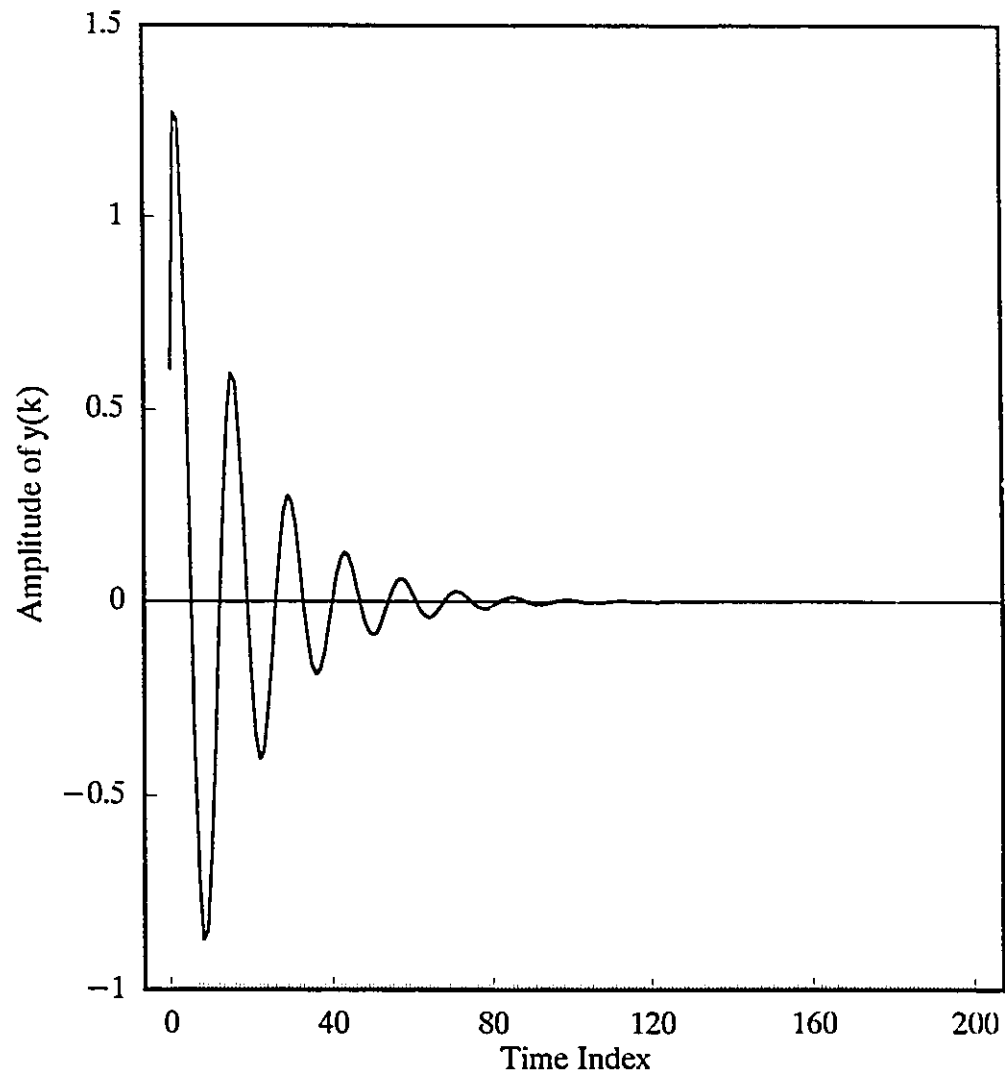


(c) Impulse response of the adaptive filter after convergence

Fig. 5.4 Simulation results of modelling a 2nd-order linear IIR digital filter of Example 1 using NAIIRDF



(a) Squared output error



(b) Impulse response of the adaptive filter after convergence

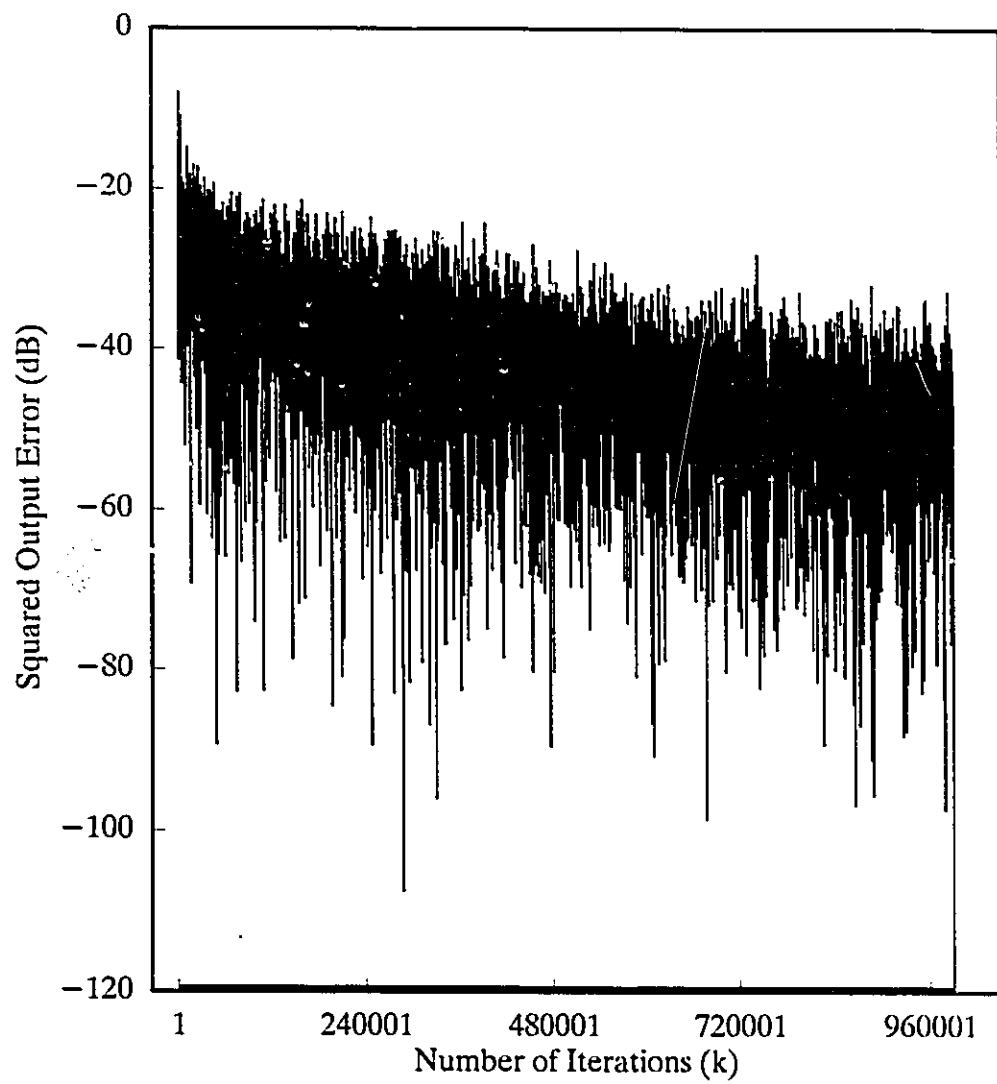
Fig. 5 .5 Simulation results of modelling a 2nd-order linear IIR digital filter of Example 1 using LAIRDF in direct form

Example 2: A fourth order linear IIR filter

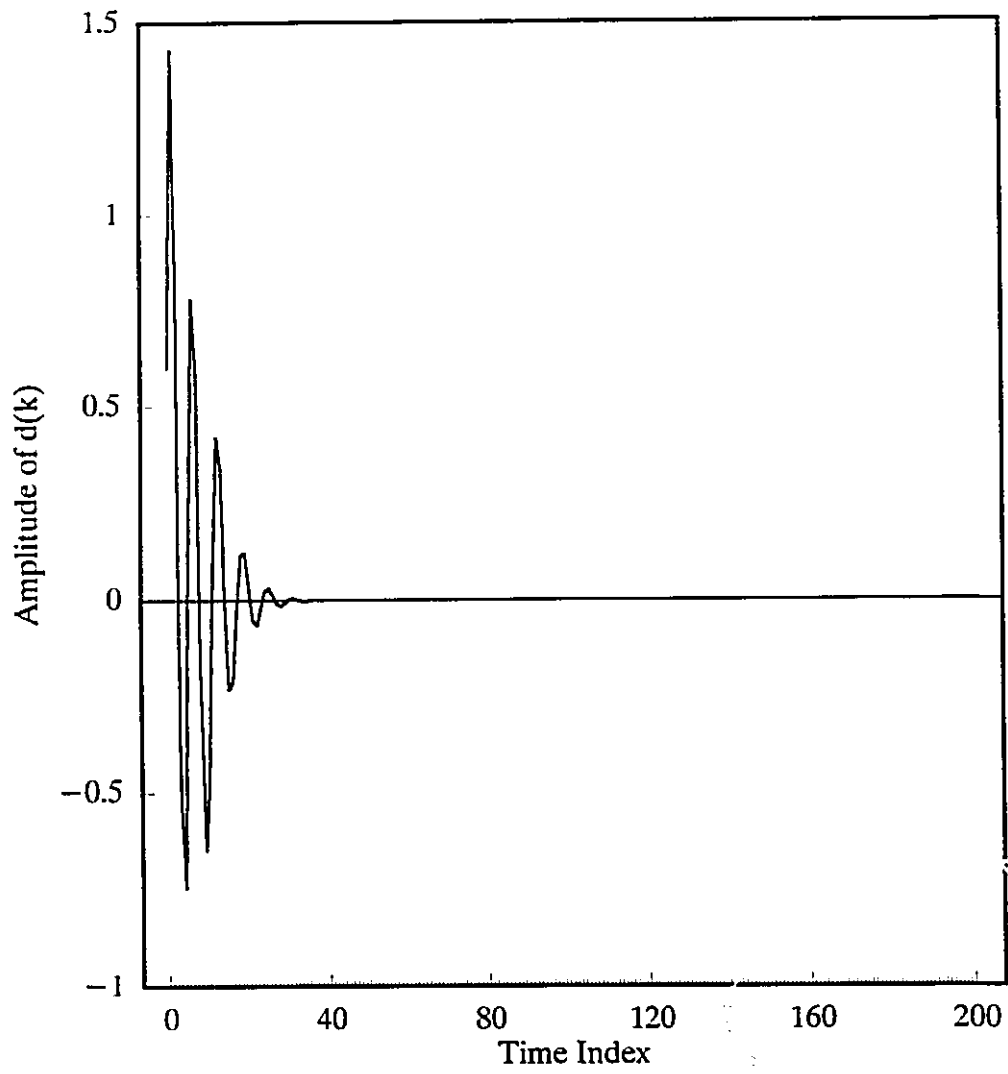
$$H(z^{-1}) = \frac{1.2 + 0.7z^{-1} - 1.0z^{-2} + 0.5z^{-3} + 1.0z^{-4}}{1.0 - 1.8z^{-1} + 2.06z^{-2} - 1.138z^{-3} + 0.3965z^{-4}}$$

is to be modelled which has poles residing at $(0.5 \pm 0.6j)$ and $(0.4 \pm 0.7j)$. Fig. 5.6 shows the simulation results obtained by using the NAIIRDF. Fig. 5.6a shows the squared output error (in dB) curve after 989600 iterations when the summation of the squared output error of the last 50 steps fell below 1.0×10^{-11} . Fig. 5.6b and Fig. 5.6c show the impulse responses of the unknown plant and the adaptive filter, respectively, and the average squared error over those 200 samples between the two responses was 2.7256×10^{-5} . The initial parameter $G(0)$ was set to 20.0 and $t(0)$ was set to a very small random number with an absolute value of less than 1.0×10^{-3} . The initial coefficients of $a_i(k)$ and $b_j(k)$ were assigned random numbers with their absolute values less than 0.5. The step parameters p_G , p_t , p_a and p_b were selected as 5.625×10^{-3} , 4.5×10^{-3} , 1.125×10^{-3} and 1.125×10^{-3} respectively.

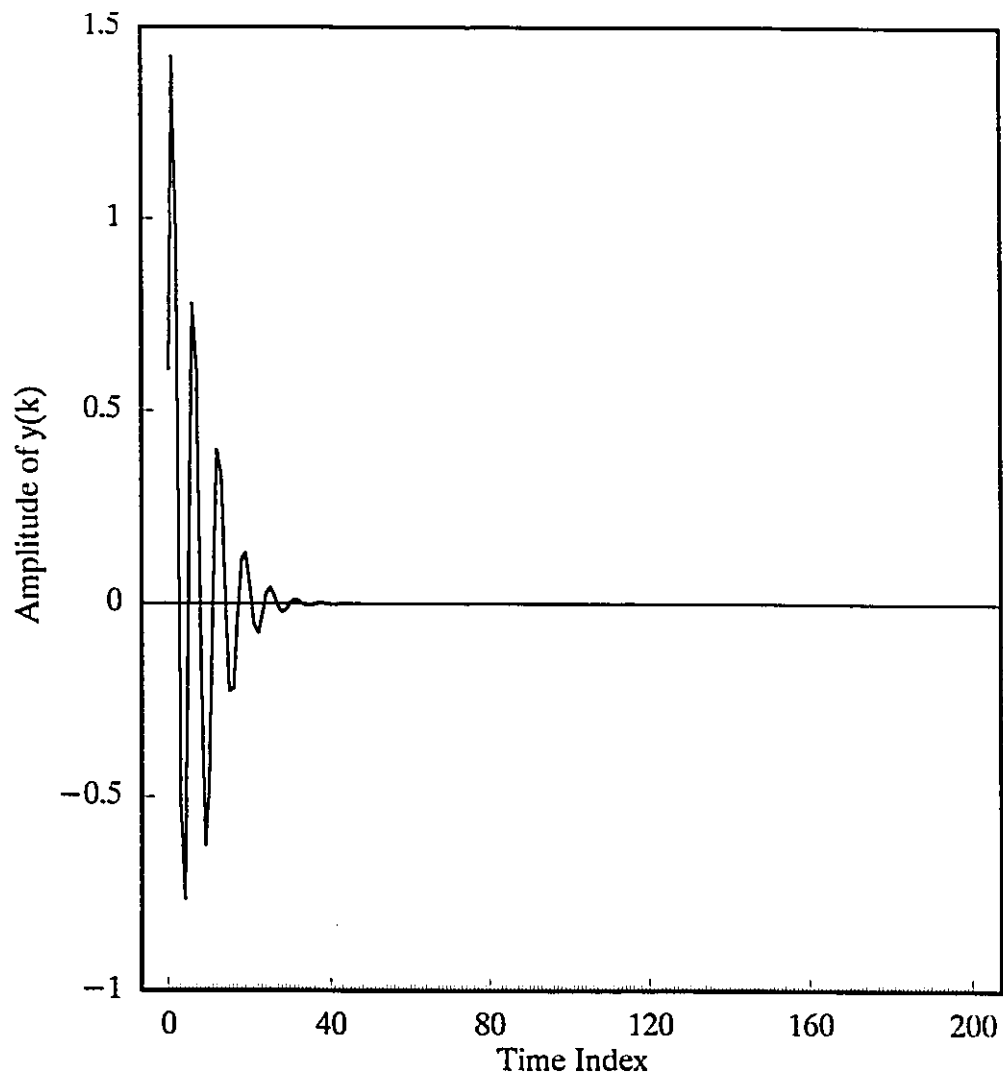
Fig. 5.7 shows the corresponding simulation results obtained by using the LAIIRDF. In this example, we set the convergence constant equal to 0.0001. Fig. 5.7a shows the squared output error (in dB) curve after 1600700 iterations when the summation of squared output error of last 50 steps fell below 1.0×10^{-11} . The impulse response of the unknown plant is shown in Fig. 5.6b and that of the adaptive filter after convergence is shown in Fig. 5.7b. The average squared error over those 200 samples between the two responses was 4.9316×10^{-5} .



(a) Squared output error

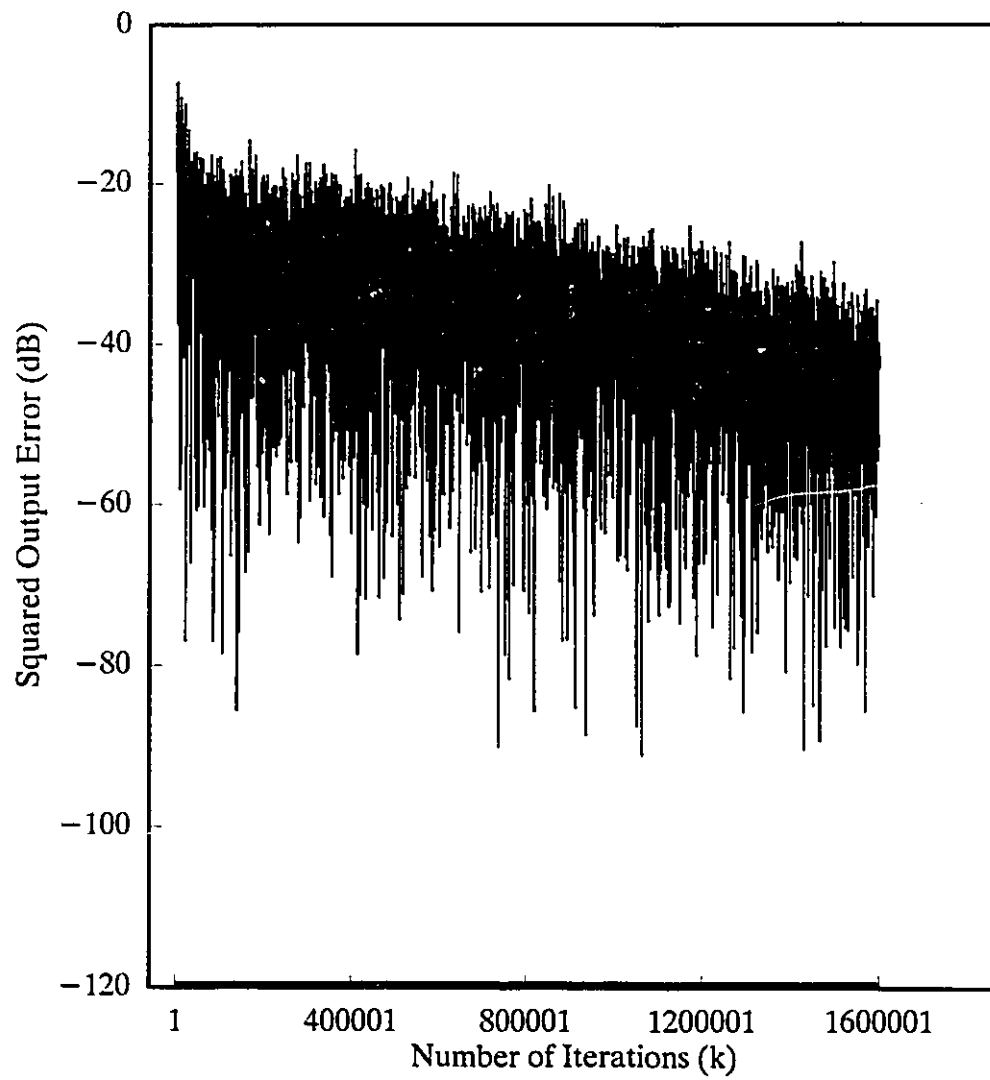


(b) Impulse response of the unknown plant

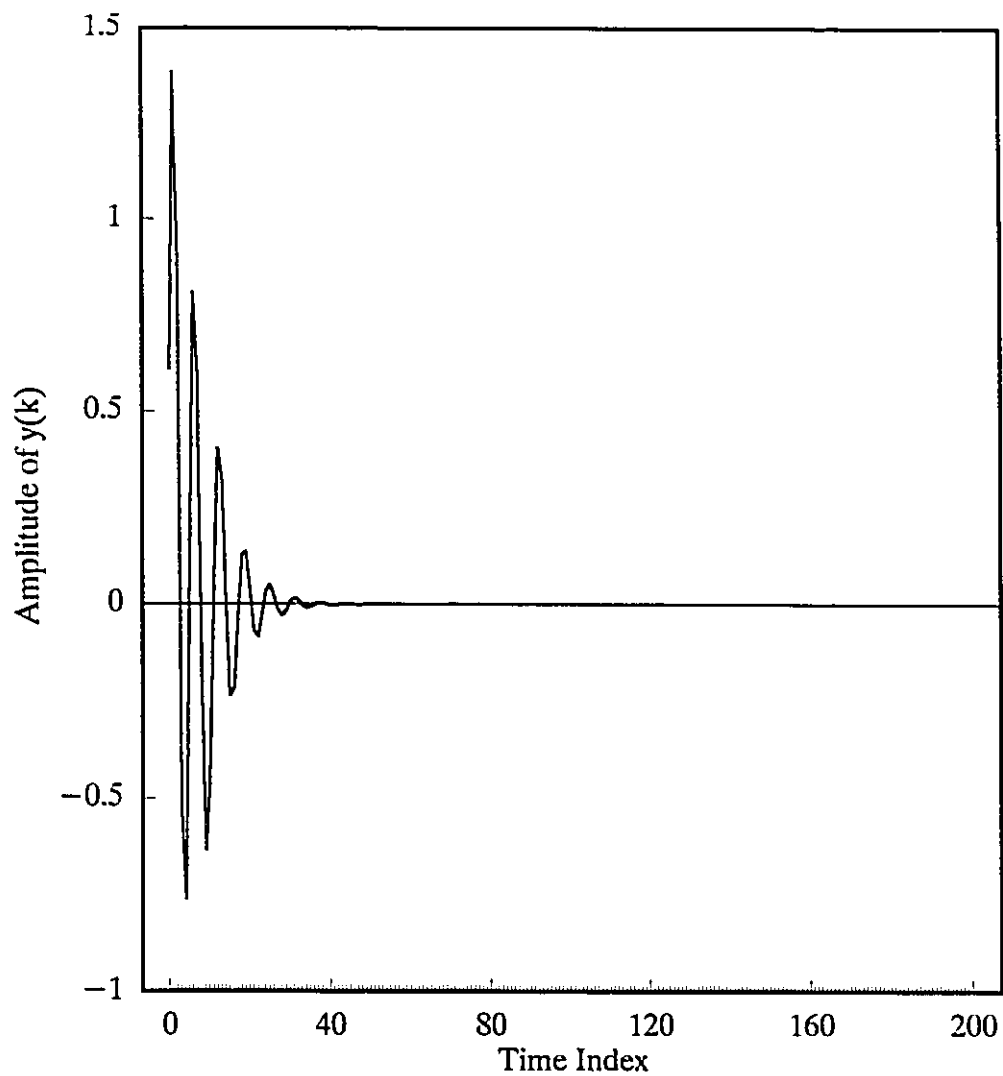


(c) Impulse response of the adaptive filter after convergence

Fig. 5.6 Simulation results of modelling a 4th-order linear IIR digital filter of Example 2 using NAIIRDF



(a) Squared output error



(b) Impulse response of the adaptive filter after convergence

Fig. 5.7 Simulation results of modelling a 4th-order linear IIR digital filter of Example 2 using LAIRDF in direct form

For a more comprehensive comparison study, we tried 10 sets of initial values for the NAIIRDF and the LAIRDF of each example respectively. The results are summarized in Table 5.1, where each number shown is actually an average value over 10 trials. In Table 5.1, NOI stands for the number of iterations and E_d represents the average squared differences over 200 samples between the responses of both the unknown plant and the adaptive filter after convergence. It can be seen from these simulation results that the proposed NAIIRDFs outperform those of LAIRDFs in terms of convergence speed and final output error residuals. The advantage of this NAIIRDF becomes more significant, when the filter order becomes larger and therefore the computational burden of stability monitoring of the LAIRDF becomes heavier. Besides the impulse responses, we also examined the differences between the responses of both the unknown plant and the adaptive filter after convergence with other types of input signals, such as unit step signal, sinusoidal signal, sawtooth signal and square wave signal. The experiment results obtained agree with that of impulse responses.

Table 5.1 Simulation results of comparison studies between NAIIRDF and LAIRDF

	Example 1		Example 2	
	NOI	E_d	NOI	E_d
NAIIRDF	420354	3.2635×10^{-4}	455560	3.0121×10^{-4}
LAIRDF	673336	4.3326×10^{-3}	1205280	3.6097×10^{-4}

Since the nonlinear sigmoid function has been involved in this structure, the adaptive algorithm would adjust the parameters of the nonlinear sigmoid function and the feedforward and feedback coefficients at the same time. The same unknown plant could be modelled by different sets of parameters and coefficients. Tables 5.2 and 5.3 list some simulation results of finally obtained parameters and coefficients of the above two examples using different initial values of $G(0)$, $t(0)$, and other coefficients' values. It is interesting to observe that even considering

$$1 - \sum_{j=0}^M b_j(k)z^{-j}$$

as a denominator polynomial like the situation in a linear IIR digital filter, those obtained coefficients of $b_j(k)$ still make the zeros of the above polynomial inside the unit circle, which can be verified by using Jury's stability criterion [82].

Table 5.2 Parameters and coefficients of NAIIRDF of Example 1 after convergence

	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	Set 7
$G(k)$	23.0190	46.0034	68.9852	91.6284	23.0120	22.9690	22.7554
$t(k)$	0.0001	0.0001	0.0000	-0.00001	0.0000	0.0001	-0.00001
$a_0(k)$	0.1180	0.0603	0.0378	0.0275	0.1179	0.1263	0.1208
$a_1(k)$	0.0352	0.0154	0.0136	0.0129	0.0362	0.0079	0.0330
$a_2(k)$	-0.0695	-0.0343	-0.0244	-0.0204	-0.0703	-0.0446	-0.0688
$b_1(k)$	0.1484	0.0740	0.0493	0.0371	0.1484	0.1489	0.1501
$b_2(k)$	-0.0779	-0.0389	-0.0259	-0.0194	-0.0778	-0.0785	-0.0788

Table 5.3 Parameters and coefficients of NAIIRDF of Example 2 after convergence

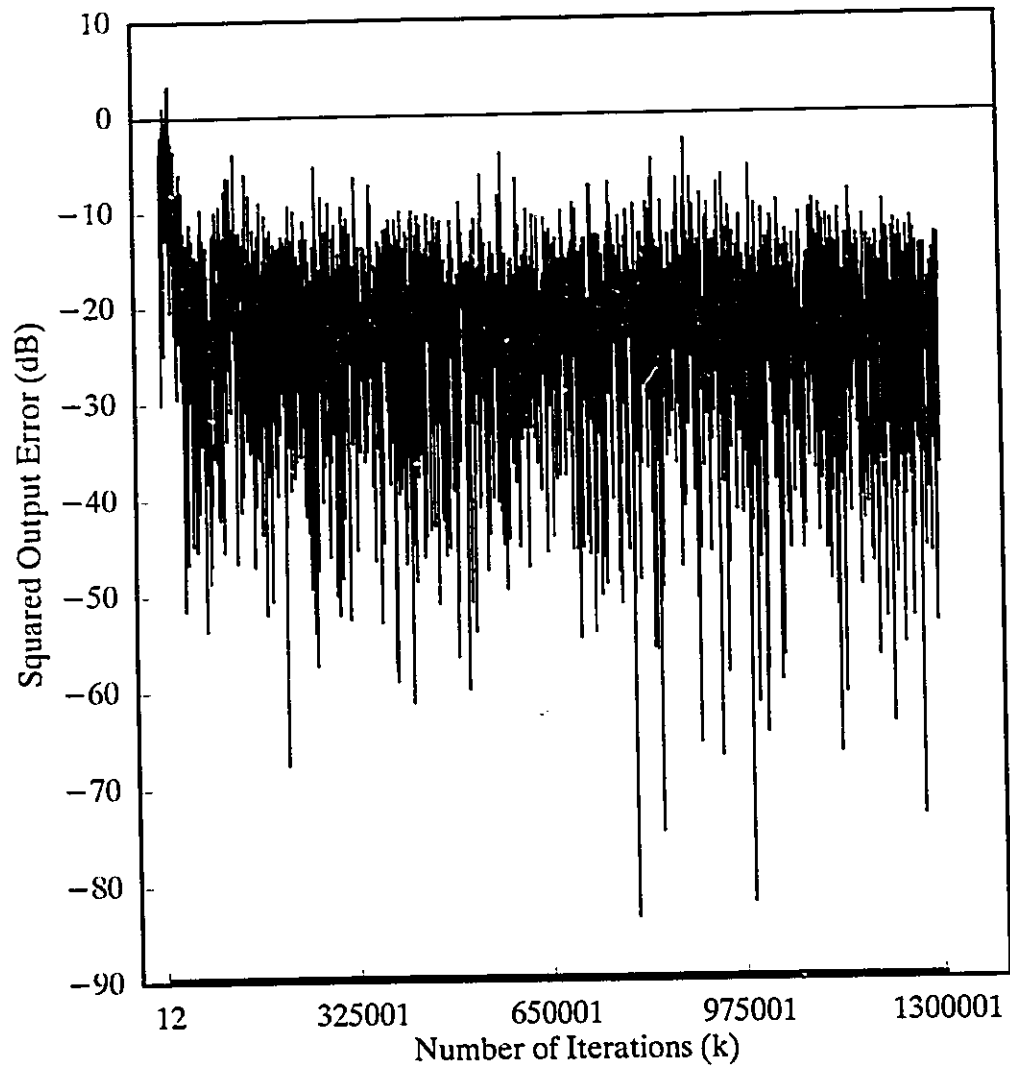
	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	Set 7
$G(k)$	20.0272	39.9869	8.8866	4.1451	19.9925	19.5361	7.3532
$t(k)$	0.0000	-0.0001	0.0003	0.0055	0.0001	0.0001	-0.0003
$a_0(k)$	0.1215	0.0609	0.2747	0.6505	0.1217	0.1215	0.3298
$a_1(k)$	0.0729	0.0375	0.2146	1.4055	0.0850	0.1217	0.2977
$a_2(k)$	-0.0832	-0.0333	-0.1415	0.5745	-0.0614	-0.0538	-0.1484
$a_3(k)$	0.0244	-0.0002	-0.0518	-0.1926	-0.0176	-0.0629	-0.1290
$a_4(k)$	0.1126	0.0630	0.2981	0.2148	0.1330	0.1481	0.5777
$b_1(k)$	0.1742	0.0840	0.3583	0.1002	0.1617	0.1427	0.4069
$b_2(k)$	-0.1958	-0.0926	-0.3786	-0.1240	-0.1737	-0.1369	-0.4063
$b_3(k)$	0.1046	0.0475	0.1785	-0.2877	0.0849	0.0500	0.1672
$b_4(k)$	-0.0356	-0.0158	-0.0530	0.0387	-0.0266	-0.0089	-0.0426

Example 3: The plant to be modelled in this example is

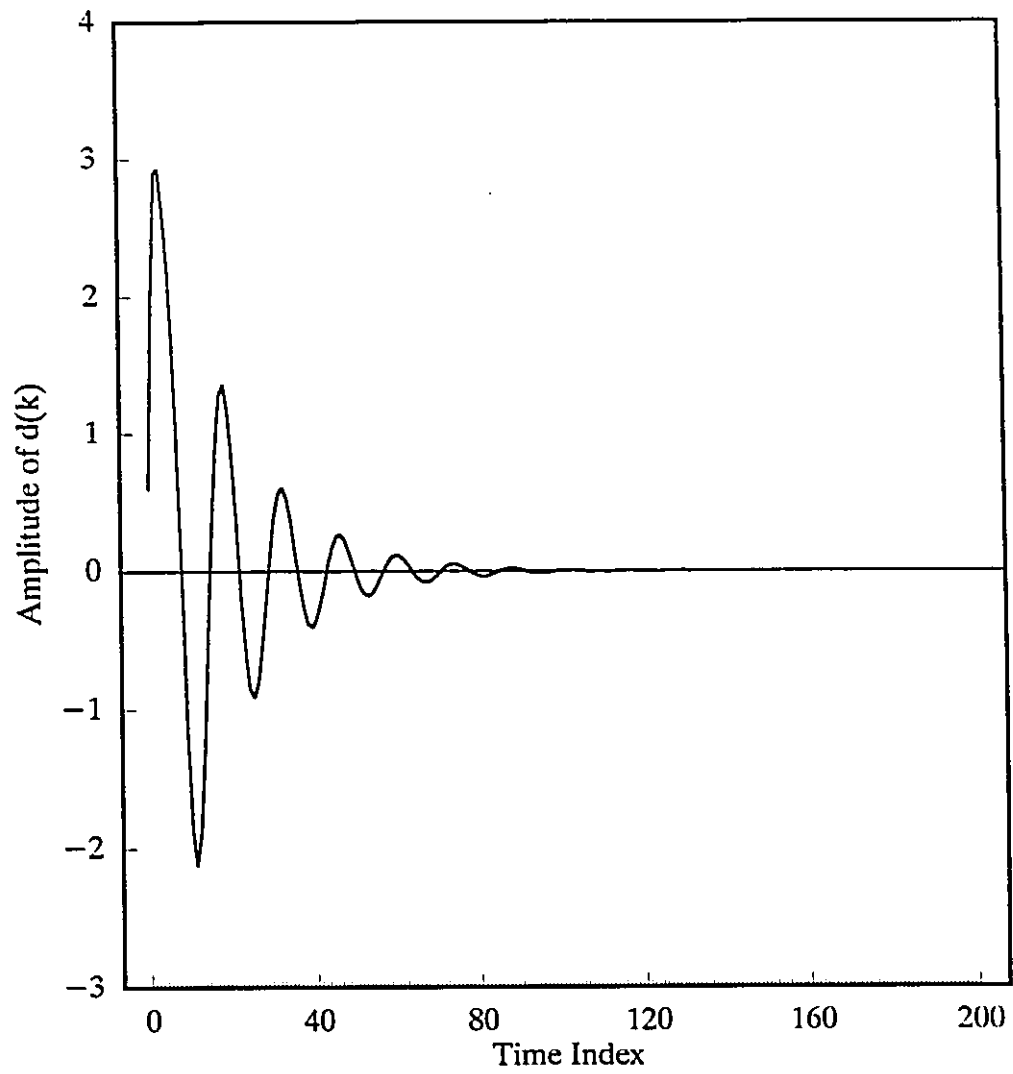
$$H(z^{-1}) = \frac{1.2 + 0.7z^{-1} - 1.0z^{-2} + 0.5z^{-3} + 1.0z^{-4}}{1.0 - 2.7z^{-1} + 3.2z^{-2} - 1.927z^{-3} + 0.5429z^{-4}}$$

It has poles residing at $(0.5 \pm 0.6j)$ and $(0.85 \pm 0.41j)$. Fig. 5.8 shows the simulation results obtained by using the NAIIRDF. Fig. 5.8a shows the squared output error (in dB) curve of the NAIIRDF after 1303650 iterations when the summation of the squared output error of the last 50 steps fell below 1.0×10^{-11} . Fig. 5.8b and Fig. 5.8c show the impulse responses of the unknown plant and the NAIIRDF, respectively, and the average squared error over those 200 samples between the two responses was 1.4383×10^{-2} . The initial parameter $G(0)$ was set to 20.0 and $t(0)$ was set to a very small random number with an absolute value of less than 0.001. The initial coefficients of $a_i(k)$ and $b_j(k)$ were

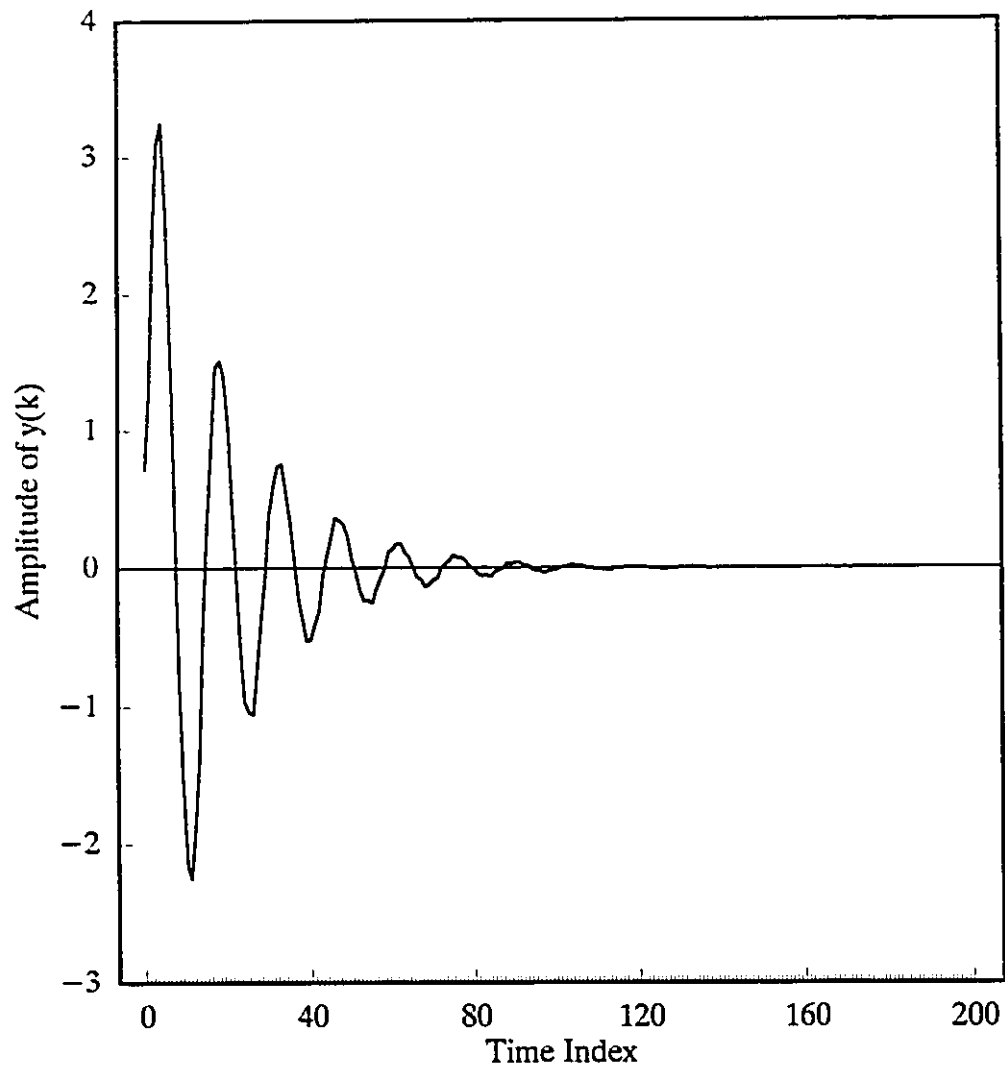
assigned random numbers with their absolute values less than 0.5. The step parameters p_G , p_t , p_a and p_b were selected as 3.125×10^{-4} , 2.5×10^{-4} , 6.25×10^{-5} and 6.25×10^{-5} respectively. Because the poles of the plant transfer function are close to the unit circle, the LAIRDF failed to identify the system. The adaptive process either went to unstable or did not converge at all.



(a) Squared output error



(b) Impulse response of the unknown plant



(c) Impulse response of the adaptive filter after convergence

Fig. 5.8 Simulation results of modelling a 4th-order linear IIR digital filter of Example 3 using NAIIRDF

5.6 Conclusions

In this chapter, we have proposed a new nonlinear adaptive IIR digital filter, namely, the NAIIRDF. Because of the nonlinear operator involved in this structure, this recursive digital filter is BIBO stable and hence does not need stability monitoring during the adaptive process. The analyses of stability and time domain performance of the NAIIRDF have been conducted, and several properties of the behavior of filter parameters and coefficients have been achieved. Based on this structure, an individual adaptation scheme to improve the convergence speed is incorporated into the adaptive algorithm, which can adjust each different parameter at every iteration so that their values are kept at an optimum for a new set of input samples. While only very conservative conclusions have been proved through theoretical analyses due to nonlinearity involved, simulation results for the applications in linear adaptive IIR system identifications have been given to demonstrate the performance of the proposed structure and algorithm. Simulation experiments performed showed that this NAIIRDF worked well with different filter orders and pole locations. However, it should be pointed out that even with the individual parameter adaptation scheme, the convergence speed of the adaptive algorithm is yet to be improved.

Chapter 6

CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH

This chapter concludes the whole dissertation presented in the previous five chapters. We also discuss some future research topics which can be considered as extensions of the research work conducted in this dissertation.

6.1 Conclusions of this Dissertation

With the rapid advances in VLSI technology and the availability of mass-produced digital signal processors (DSPs), the use of multiple processing elements to considerably increase the processing speed of a digital signal processing system beyond the limit imposed by device technology is becoming practical. The first part of this dissertation (Chapter 2 and Chapter 3) deals with high speed linear adaptive digital signal processing with parallel structures, which can be realized by multiple DSPs. Firstly, we presented a family of high speed linear adaptive digital filters, whose structures are suitable for parallel realization. They are (1) Delayed N-path adaptive finite impulse response (DNA-FIR) digital filters; (2) Delayed N-path adaptive linear phase finite impulse response (DNALP-FIR) digital filters; (3) Delayed N-path equation-error based adaptive

infinite impulse response (DNEEBA-IIR) digital filters. The processing speed of the DNA-FIR digital filter, the DNALP-FIR digital filter, and the DNEEBA-IIR digital filter has been improved by N , $2N$, and $2N$ times, respectively. The price for these speed gains is the using of multiple DSPs in each filter structure, where N^2 DSPs are used in both DNA-FIR digital filter and DNALP-FIR digital, and $2N^2$ DSPs are used in a DNEEBA-IIR digital filter. This type of digital filters is contrived to be useful for general applications in adaptive digital signal processing. Simulation examples were provided in the applications to system modelling and noise cancellation. Secondly, comparison studies were conducted among the proposed DNA-FIR digital filter (the basic structure used for the family of linear delayed N -path digital filters), the BIA-FIR digital filter, and the CA-FIR digital filter. It has been shown that the processing speed of the proposed DNA-FIR digital filter is N times that of the BIA-FIR digital filter.

As a subsequent of the above work, Chapter 4 and Chapter 5 describe research work on nonlinear adaptive digital filtering. We first presented a nonlinear delayed N -path adaptive FIR (NDNA-FIR) digital filter, the parallel structure of which lends itself to high speed implementation. This NDNA-FIR digital filter compares favorably with the CA-FIR digital filter and the nonlinear median filter in the application to broadband noise cancellation. Then studies were conducted on nonlinear adaptive IIR digital filtering. The structure and its adaptive algorithm of a new nonlinear adaptive IIR digital filter was presented in Chapter 5. This recursive digital filter is BIBO stable. Based on this structure, an individual adaptation scheme is incorporated into the adaptive algorithm to

improve the convergence speed. Simulation results for the application in linear adaptive IIR system identifications were given to demonstrate the performance of the proposed nonlinear adaptive IIR digital filter.

6.2 Suggestions for Future Research

In this final section, the author would like to mention a number of aspects worthy of future investigations.

The implementation of the proposed delayed N-path structures is suggested to be carried out using commercially available DSP chips. It can be expected that some design problems must be solved before the adaptive digital filters can be put into practical use. For examples, how the switches can be implemented efficiently and how the multiple DSPs can be connected without losing too much speed gain. In fact, the real implementation of the proposed structures could open a gateway to numerous other problems which are worthy of further studies.

In this dissertation, a new nonlinear adaptive IIR digital filter which has an adaptively adjustable nonlinear function at its output is presented. Unlike the linear adaptive IIR digital filter, this nonlinear adaptive IIR digital filter does not need any stability monitoring and hence the adaptive process will not be interrupted by the stability problem. It has been proved that the filter output is always bounded with bounded input

signals. Analysis on time domain performance has been conducted and several properties of the behaviour of filter parameters and coefficients have been achieved. However, there is still much work left to be studied. While some conclusions on the time domain performance have been achieved and simulation results have also shown good time domain behavior during the adaptive process, it is very important to conduct more theoretical analyses to obtain a whole picture of the time domain performance of this nonlinear adaptive digital filter. This kind of adaptive digital filters has performed well for linear system modelling. It is expected that some more applications can be found and subsequently their associated problems will have to be solved. At the present stage, the adaptive algorithm used for this nonlinear adaptive digital filter is based on LMS algorithm. It is of practical importance to find some fast adaptive algorithms which suit this nonlinear structure. Finally, it is suggested that this nonlinear adaptive IIR digital filter to be implemented by using VLSI technique.

BIBLIOGRAPHY

- [1] S. R. Parker and L. J. Griffiths, Editorial, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 3, June 1981, pp. 625-626.
- [2] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 2nd Edition, 1991.
- [3] N. I. Ansari, M. S. Athreya, and S. V. Gutta, "Optical computing," *IEEE Potentials*, vol. 11, No. 4, Dec. 1992, pp. 33-36.
- [4] A. S. Elmaghraby, A. Kumar, A. Desoky, and R. Sanghavi, "Simulation-based performance study of a parallel architecture," *International Journal of Mini and Microcomputers*, vol. 14, No. 1, Jan. 1992, pp. 1-8.
- [5] J. L. Aravena and W. A. Porter, "Highly parallel representations for linear maps," *IEE Proceedings, Part G on Computers and Digital Techniques*, vol. 139, No. 3, May 1992, pp. 173-178.
- [6] T. E. Dillinger, *VLSI Engineering*. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1988.
- [7] B. Widrow and S. Stearns, *Adaptive Signal Processing*, Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1985.
- [8] C. F. N. Cowan and P. M. Grant, *Adaptive Filters*, Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1985.
- [9] M. L. Honig and D. G. Messerschmitt, *Adaptive Filters: Structures, Algorithms, and Applications*, Boston, M.A.: Kluwer Academic, 1984.
- [10] K. K. Dhar and K. Hirano, "A digital filter design algorithm suitable for multi-DSP implementation," Workshop Digest of 1985 IEEE International Workshop on Digital Signal Processing, Kyoto, Japan, June 4, 1985, pp. 2g-1 to 2g-7.
- [11] H. K. Kwan, "High speed multiprocessor implementation of FIR digital filters," Internal Report, Dept. of Electrical Engineering, University of Hong Kong, August 13, 1985, 3 pages.
- [12] H. K. Kwan, "High speed multiprocessor realization of one-dimensional digital filters," Internal Report, Dept. of Electrical Engineering, University of Hong Kong, October 15, 1985, 27 pages.

- [13] K. Hayashi, K. K. Dhar, K. Sugahara, and K. Hirano, "Design of high-speed digital filters suitable for multi-DSP implementation," *IEEE Transactions on Circuits and Systems*, vol. 33, No. 2, Feb. 1986, pp. 202-217.
- [14] H. K. Kwan and M. T. Tsim, "Efficient method for the high speed realization of delayed multipath 1-D IIR digital filters," Proceedings of China Seventh Conference on Circuits and Systems, Shenzhen, China, Nov. 16-18, 1987, vol. 2, pp. 22-1 to 22-4.
- [15] H. K. Kwan and M. T. Tsim, "Systolic realization of high speed delayed multipath digital filters," Proceedings of 31st Midwest Symposium on Circuits and Systems, Missouri, U.S.A., Aug. 9-12, 1988, pp. 63-66.
- [16] H. K. Kwan and M. T. Tsim, "Efficient high speed delayed multipath one-dimensional recursive digital filter architecture," in *Signal Processing IV: Theories and Applications*, Edited by J. L. Lacoume, A. Chehikian, N. Martin, and J. Malbos, North-Holland, Amsterdam, 1988, vol. II, pp. 755-758.
- [17] H. K. Kwan and M. T. Tsim, "Efficient systolic high speed architectures for delayed multipath two-dimensional FIR and IIR digital filtering," *IEE Proceedings, Part G on Circuits, Devices and Systems*, vol. 137, No. 6, Dec. 1990, pp. 413-423.
- [18] H. K. Kwan, "Systolic realization of delayed 2-path linear phase FIR digital filters," *IEE Proceedings, Part G on Computers and Digital Techniques*, vol. 140, No. 1, Jan. 1993, pp. 75-80.
- [19] I. Pitas and A. N. Venetsanopoulos, *Nonlinear Digital Filters, Principles and Applications*. Boston: Kluwer Academic Publishers, 1990.
- [20] T. H. Meng and D. G. Messerschmitt, "Arbitrarily high sampling rate adaptive filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, No. 4, April 1987, pp. 455-470.
- [21] G. A. Clark, S. K. Mitra, and S. R. Parker, "Block implementation of adaptive digital filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, No. 3, June 1981, pp. 744-752.
- [22] G. A. Clark, "Block adaptive filtering and its application to seismic event detection," Ph.D. Dissertation, University of California, Santa Barbara, 1981.
- [23] J. M. Cioffi, "The block-processing FTF adaptive algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, No. 1, Feb. 1986, pp. 77-90.

- [24] H. Sakai, "A parallel least-squares linear prediction method based on the circular lattice filter," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, No. 3, June 1986, pp. 640-642.
- [25] G. A. Clark, S. R. Parker, and S. K. Mitra, "A unified approach to time- and frequency-domain realization of FIR adaptive digital filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 31, No. 5, Oct. 1983, pp. 1073-1083.
- [26] J. J. Shynk, "Frequency-domain and multirate adaptive filtering," *IEEE Signal Processing Magazine*, vol. 9, No. 1, Jan. 1992, pp. 15-37.
- [27] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing*, Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1983.
- [28] P. P. Vaidyanathan, "Multirate digital filters, filter banks, polyphase networks and applications: a tutorial," *Proceedings of IEEE*, vol. 78, No.1, Jan. 1990, pp. 56-93.
- [29] M. Renfors and T. Saramaki, "Recursive Nth-band digital filters - Part I: design and properties," *IEEE Transactions on Circuits and Systems*, vol. 34, No. 1, Jan. 1987, pp. 24-39.
- [30] V. S. Somayazulu, S. K. Mitra, and J. J. Shynk, "Adaptive line enhancement using multirate techniques," *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, Glasgow, Scotland, May 23-26, 1989, vol. 2, pp. 928-931.
- [31] A. Gilloire and M. Vetterli, "Adaptive filtering in sub-bands," *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, New York City, U.S.A., April 11-14, 1988, vol. III, pp. 1572-1575.
- [32] Y. Papananos and D. Anastassiou, "Analysis and VLSI architecture of a nonlinear edge-preserving noise-smoothing image filter," *IEE Proceedings, Part G on Circuits, Devices and Systems*, vol. 138, No. 4, Aug. 1991, pp. 433-440.
- [33] A. Miyazaki and M. Nakata, "A new adaptive algorithm of adaptive digital filters using neural networks," in *Proceedings of International Symposium on Circuits and Systems*, Singapore, vol. 3, June 11-14, 1991, pp. 1388-1391.
- [34] X. H. Yu and S. X. Cheng, "Nonlinear statistical optimum adaptive filtering and signal detection via BP neural nets," in *Proceedings of International Symposium on Circuits and Systems*, Singapore, vol. 3, June 11-14, 1991, pp. 1420-1423.

- [35] X. Nie and R. Unbehauen, "Edge preserving filtering by combining nonlinear mean and median filters." *IEEE Transactions on Signal Processing*, vol. 39, No. 11, Nov. 1991, pp. 2552-2554.
- [36] L. Naaman and A. C. Bovik, "Least squares order statistic filters for signal restoration," *IEEE Transactions on Circuits and Systems*, vol. 38, No. 3, March 1991, pp. 244-257.
- [37] B. Widrow, J. Glover (Jr.), J. M. McCool, J. Kaunitz, C. S. William, R. H. Hearn, J. R. Zeidler, E. Dong, and R. C. Goodlin, "Adaptive noise cancelling: principles and applications," *Proceedings of IEEE*, vol. 63, No. 12, Dec. 1975, pp. 1692-1716.
- [38] R. B. Wallace and R. A. Goubran, "Noise cancellation using parallel adaptive filters," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 39, No. 4, April 1992, pp. 239-243.
- [39] S. L. Marple (Jr.), "Fast algorithms for linear prediction and system identification filters with linear phase," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 30, No. 6, June 1982, pp. 942-953.
- [40] S. Theodoridis, N. Kalouptsidis, J. Proakis, and G. Koyas, "Interference rejection in PN spread-spectrum systems with LS linear phase FIR filters," *IEEE Transactions on Communications*, vol. 37, No. 9, Sept. 1989, pp. 991-994.
- [41] L. Rabiner and R. Schafer, *Digital Processing of Speech Signals*, Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1978.
- [42] N. Kalouptsidis and S. Theodoridis, "Fast sequential algorithm for least squares FIR filters with linear phase," *IEEE Transactions on Circuits and Systems*, vol. 35, No. 4, April 1988, pp. 425-432.
- [43] D. C. Farden and J. R. Bellegarda, "A new structure for adaptive linear-phase filtering," *IEEE Transactions on Circuits and Systems*, vol. 34, No. 7, July 1987, pp. 712-720.
- [44] N. Kalouptsidis, "Efficient computation of multichannel Wiener filters with linear phase," *IEEE Transactions on Circuits and Systems*, vol. 35, No. 4, April 1988, pp. 437-438.
- [45] N. Kalouptsidis and G. D. Koyas, "Efficient block LS design of FIR filters with linear phase," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, No. 6, June 1985, pp. 1435-1444.

- [46] N. Kalouptsidis and S. Theodoridis, "Efficient structurally symmetric algorithms for least squares FIR filters with linear phase, " *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, No. 9, Sept. 1988, pp. 1454-1465.
- [47] K. Berberidis and S. Theodoridis, "New Levinson, Schur, and lattice type algorithms for linear phase filtering, " *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, No. 11, Nov. 1990, pp. 1879-1892.
- [48] H. K. Kwan, "High Speed delayed 2-path 1-D linear phase FIR digital filtering architecture, " *Proceedings of International Symposium on Electronic Devices, Circuits and Systems, Kharagpur, India, Dec. 16-18, 1987*, pp. 876-878.
- [49] H. K. Kwan, "Systolic realization of delayed 2-path 1-D linear phase FIR digital filters, " *Proceedings of International Conference on Circuits and Systems, Nanjing, China, July 6-8, 1989*.
- [50] K. Kurosawa and S. Tsujii, "An IIR parallel-type adaptive algorithm using fast least squares method, " *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, No. 8, Aug. 1989, pp. 1226-1230.
- [51] J. J. Shynk, "Adaptive IIR filtering using parallel-form realization, " *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, No. 4, April 1989, pp. 519-533.
- [52] F. X. Y. Gao and W. M. Snelgrove, "An efficient adaptive cascade IIR filter, " *Proceedings of International Symposium on Circuits and Systems, Singapore, June 11-14, 1991*, vol. 1, pp. 444-447.
- [53] M. G. Larimore, J. R. Treichler, and C. R. Johnson, "SHARF: an algorithm for adapting IIR digital filters, " *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, No. 4, April 1980, pp. 428-440.
- [54] S. Karni and G. Zeng, "An adaptive IIR algorithm with unimodel performance surfaces," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, No. 2, Feb. 1988, pp. 286-287.
- [55] S. Dimolitsas, "An adaptive IIR filter based on the Hurwitz stability properties and a Chebychev system function approximation, " *Signal Processing*, vol. 17, No. 1, Jan. 1989
- [56] K. Steiglitz and L. E. McBride, "A technique for the identification of linear systems, " *IEEE Transactions on Automatic Control*, vol. 10, No. 10, Oct. 1965, pp. 461-464.

- [57] W. B. Mikhael, F. H. Wu, L. G. Kazovsky, L. G. Kang, and L. J. Fransen, "Adaptive filters with individual adaptation of parameters," *IEEE Transactions on Circuits and Systems*, vol. 33, No. 7, July 1986, pp. 677-686.
- [58] C. W. Barnes and S. Shinnaka, "Block-shift invariance and block implementation of discrete-time filters," *IEEE Transactions on Circuits and Systems*, vol. 27, No. 8, Aug. 1980, pp. 667-672.
- [59] C. S. Burrus, "Block realization of digital filters," *IEEE Transactions on Audio and Electroacoustics*, vol. 20, No. 4, Oct. 1972, pp. 230-235.
- [60] C. S. Burrus, "Block implementation of digital filters," *IEEE Transactions on Circuit Theory*, vol. 18, No. 6, Nov. 1971, pp. 697-701.
- [61] B. Widrow, J. McCool, M. Larimore, and C. Johnson, "Stationary and nonstationary learning characteristics of the LMS adaptive filters," *Proceedings of IEEE*, vol. 64, No. 8, Aug. 1976, pp. 1151-1161.
- [62] T. Koh and E. J. Powers, "Second-order Volterra filtering and its application to nonlinear system identification," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, No. 6, Dec. 1985, pp. 1445-1455.
- [63] K. D. Rao and D. C. Reddy, "New method of designing adaptive nonlinear filters," *IEE Proceedings, Part F on Radar and Signal Processing*, vol. 138, No. 5, Oct. 1991, pp. 513-519.
- [64] J. N. Lin and R. Unbehauen, "Adaptive nonlinear digital filter with canonical piecewise-linear structure," *IEEE Transactions on Circuits and Systems*, vol. 37, No. 3, March 1990, pp. 347-353.
- [65] S. D. Stearns, "Error surfaces of recursive adaptive filters," *IEEE Transactions on Circuits and Systems*, vol. 28, No.6, June 1981, pp. 603-606.
- [66] S. Horvath, Jr., "A new adaptive recursive LMS filter," in *Digital Signal Processing*, New York: Academic Press, 1980.
- [67] J. J. Shynk, "Adaptive IIR filtering using parallel-form realizations," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, No.4, April 1989, pp. 519-533.
- [68] B. Friedlander, "Lattice filters for adaptive processing," *Proceedings of IEEE*, vol. 70, No. 8, Aug. 1982, pp. 829-867.
- [69] C. R. Johnson, Jr., "Adaptive IIR filtering: Current results and open issues,"

IEEE Transactions on Information Theory, vol. 30, No. 2, March 1984, pp. 237-250.

- [70] 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. 28, No. 4, Aug. 1980, pp. 428-440.
- [71] J. Y. Lin and C. H. Wei, "A new adaptive equalizer for nonlinear channels," Proceedings of International Symposium on Circuits and Systems, Singapore, pp. 2814-2817, June 11-14, 1991.
- [72] Y. H. Gu and W. M. G. van Bokhoven, "Frequency bin nonlinear LMS adaptive noise canceller and its application to cochannel speech noise reduction," Proceedings of International Symposium on Circuits and Systems, Singapore, pp. 2822-2825, June 11-14, 1991.
- [73] G. Dong and X. Ling, "Combinational neural net for adaptive filtering," Proceedings of International Symposium on Circuits and Systems, Singapore, pp. 1392-1395, June 11-14, 1991.
- [74] Y. H. Gu and W. M. G. van Bokhoven, "A RLS variable-length sliding-window nonlinear filtering algorithm for system identification and adaptive noise cancellation," Proceedings of International Symposium on Circuits and Systems, Singapore, pp. 2806-2809, June 11-14, 1991.
- [75] H. K. Kwan and Q. P. Li, "Noise cancellation using nonlinear adaptive FIR digital filters," in Proceedings of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Victoria, B.C., Canada, vol. 1, pp. 358-361, May 19-21, 1993.
- [76] A. Friedman, *Advanced Calculus*, Holt, Rinehart and Winston, Inc., 1971.
- [77] W. B. Mikhael and F. H. Wu, "Gradient algorithms for FIR adaptive filtering - A tutorial," Proceedings of International Symposium on Circuits and Systems, Singapore, pp. 560-563, June 11-14, 1991.
- [78] W. B. Mikhael, F. H. Wu, G. Kang, and L. Fransen, "Optimum adaptive algorithms with applications to noise cancellation," *IEEE Transactions on Circuits and Systems*, vol. 31, No. 3, March 1984, pp. 312-315.
- [79] W. B. Mikhael and F. H. Wu "Fast algorithms for block FIR adaptive digital filtering," *IEEE Transactions on Circuits and Systems*, vol. 34, No. 10, Oct. 1987, pp. 1152-1160.

- [80] W. B. Mikhael and F. H. Wu "A fast block FIR adaptive digital filtering algorithm with individual adaptation of parameters," *IEEE Transactions on Circuits and Systems*, vol. 36, No. 1, Jan., 1989, pp. 1-10.
- [81] A. S. Spanias and W. B. Mikhael, "Implementation of the optimum block adaptive algorithm in the frequency-domain," Proceedings of International Symposium on Circuits and Systems, Philadelphia, PA, pp. 426-429, May 1987.
- [82] A. Antoniou, *Digital Filters: Analysis and Design*, McGraw-Hill, Inc., 1979.

VITA AUCTORIS

NAME: Qiu Ping (Jennifer) LI

PLACE OF BIRTH: Nanjing, Jiangsu, P. R. China

YEAR OF BIRTH: 1963

EDUCATION: University of Science and Technology of China, Hefei, Anhui,
P. R. China
1980-1985, B.A.Sc.

University of Science and Technology of China, Hefei, Anhui,
P. R. China
1985-1988, M.A.Sc.

University of Windsor, Windsor, Ontario, Canada
1989-todate, Ph.D. Candidate