Adaptation Algorithms for Data Echo Cancellation Using Nonquadratic Cost Functions

by

Syed Amjad Hussain Shah, B. Sc., M. Sc.

A Doctoral Thesis
Submitted in partial fulfilment of the requirements
for the award of Doctor of Philosophy
of the Loughborough University of Technology

July 1995

Supervisor: Prof. C. F. N. Cowan
Department of Electronic and Electrical Engineering,

Adaptation Algorithms for Data Echo Cancellation Using Nonquadratic Cost Functions

by

Syed Amjad Hussain Shah, B. Sc., M. Sc.

A Doctoral Thesis
Submitted in partial fulfilment of the requirements
for the award of Doctor of Philosophy
of the Loughborough University of Technology

July 1995

Supervisor: Prof. C. F. N. Cowan
Department of Electronic and Electrical Engineering,

Contents

Certificate of originality v

Synopsis vi

Acknowledgements vii

Principal symbols viii

Abbreviations x

1 Introduction **1**

2 Adaptive echo Cancellation **5**

2.1 Introduction 5

2.2 Adaptive Filters 6

2.3 Adaptive filter operation 6

2.3.1 Direct system modelling 8

2.3.2 Inverse system modelling 8

2.3.3 Prediction 10

2.3.4 Interference cancelling 10

2.4 Programmable filter designs, IIR and FIR 12

2.4.1 Recursive IIR filters 12

2.4.2 Nonrecursive FIR filters 14

2.5 Cost functions 16
2.6 Significance of echo 17
2.7 Sources and mechanism of echo generation 18
2.8 Methods for cancelling echoes 22
2.9 Echo cancellation 23
2.10 Echo cancellation for full duplex data transmission 25

3 Proposed adaptive algorithm **28**
3.1 Introduction 28
3.2 Proposed adaptive algorithm 29
3.3 Derivation of the adaptation algorithm 32
3.4 Conditions for the convergence of the mean of the tap error vector 35
3.5 Time constants 38

4 Modelling **39**
4.1 Introduction 39
4.2 Modelling the data streams 39
4.3 Echo path models 40
4.3.1 First model 40
4.3.2 Second model 43
4.3.3 Third model 45
4.4 Simulation setup 47
4.5 Performance measure 49
4.6 Simulations 50

5 Simulations with binary data **54**
5.1 Introduction 54

5.2 Simulations with binary data 55
5.3 Convergence level and some aspects of calculated and measured step size of the convergence process. 75
5.4 Dispersion of far-end signal in addition to attenuation 82
5.5 Achievement of extended tolerance 85
5.6 Stretching tolerance by switching to lower gradient during convergence 88
5.7 Switching behaviour 98
5.8 Conclusions 104

6 Simulations with quaternary data **105**
6.1 Introduction 105
6.2 Simulations with quaternary data 106
6.3 Stretching tolerance by switching to lower gradient during convergence 118
6.4 Switching behaviour 128
6.5 Conclusions 134

7 Conclusions **135**
7.1 Summary of achievements 135
7.2 Suggestions for future research 138

A Numerical values **152**
A.1 A calculation of the numerical value of the far-end signal attenuation f 152
A.2 Numerical values of filter coefficients 153

A.2.1 Numerical values of the filter coefficients for the first echo path model 154
A.2.2 Numerical values of the filter coefficients for the second echo path model 154
A.2.3 Numerical values of the filter coefficients used for the dispersion of the far-end signal 155

B Computer programmes **156**

B.1 Fortran source code used for binary data sequences without switching of gradient during convergence 156
B.2 Fortran source code used for binary data sequences with switching of gradient during convergence 160
B.3 Fortran source code used for quaternary data sequences without switching of gradient during convergence 164
B.4 Fortran source code used for quaternary data sequences with switching of gradient during convergence 168
B.5 Fortran source code used for binary data sequences with dispersion added to far-end signal 172

C Publication **176**

Synopsis

A new stochastic gradient adaptation algorithm based on the cost function $E[|e_k|^\tau]$, where $\tau \geq 2.0$, and is a rational number, is proposed. Conditions for the convergence in the mean of the adaptive algorithm are derived along with the stability bounds on the step size μ. Merits of the new adaptation algorithm as compared with that of the least mean square (LMS) algorithm are demonstrated by means of simulations. Computer simulations were performed with non-Gaussian binary and quaternary sequences of data. Simulations are performed in the presence of far-end signal sequences of various attenuation levels in data echo cancellers for full duplex digital data transmission over telephone lines. Three different echo path models were used in these simulations along with four attenuation levels for the far-end data sequences. Convergence goals were set 20 dB below the attenuation level of the far-end signals in each case. In a given set of simulations, τ was increased starting from 2.0 in steps of 0.1 for each successive simulation as long as the algorithm remains convergent. It is observed that convergence time decreases with the increase in τ initially and then levels off before increasing once again. These simulations indicate that a substantial reduction in convergence time can be achieved relative to the mean square algorithm. The amount of reduction in initial convergence time depends upon various parameters such as transfer function characteristics of the echo path, attenuation level of the far-end signal and type of data. A set of simulations was also performed after introducing dispersion in the far-end signal in addition to the attenuation. Results of which show the same trend of reduction in convergence time with the increase in τ, as for the case of attenuated only far-end signal. Although the superiority of the proposed algorithm is demonstrated for digital data echo cancellation only, it could be applied to various other areas of adaptive signal processing where data are non-Guassian.

Acknowledgements

My special thanks go to my supervisor Professor Colin Cowan, for his continuous inspiration and guidance over a period of almost two and a half years.

I also thank my wife Taqleed and children Shamama and Rajeel for their understanding and constant support in keeping the morale high and for bearing with my extended working hours during the research and preparation of this thesis. I would also like to thank my parents whose encouragements and prayers were always with me during this separation of more than three years

I also acknowledge the financial assistance provided by the Ministry of Science and Technology, Government of Pakistan, without which this research would never have been possible.

Principal symbols

$\mathbf{0}$	Null matrix.
a	Feedback coefficient of the transfer function of the echo path.
$\mathbf{A}$	A matrix.
d_k	Dispersed and attenuated far-end signal.
e_k	Error signal at time sample k.
E	Mathematical expectation operator.
f	Far-end signal attenuation.
f_k	Attenuated far-end signal at time sample k.
h_n	nth filter coefficient.
$\mathbf{h}$	Filter coefficients vector, or the impulse response.
$\mathbf{h}_k$	Filter coefficients vector at time sample k.
$\hat{\mathbf{h}}_k$	Estimated filter coefficients vector at time sample k.
$H(z)$	Transfer function.
$\mathbf{I}$	Identity matrix.
k	The number of time sampling instant.
n	Variable counter.
N	Total number of filter coefficients.
$p_{(k)}$	Performance measure at time sample k.
T_i	Time constant of the ith convergence mode.
tr.	Trace of a matrix.
$\mathbf{v}_k$	Tap error vector, $\hat{\mathbf{h}}_k - \mathbf{h}_k$.
v_k	Norm of $E[\mathbf{v}_k]$.

x_k	Input signal at time sample k.
$\mathbf{x}_k$	Input data vector with present and $N-1$ previous values.
$\mathbf{x}_k^t$	Matrix transposition of $\mathbf{x}_k$.
y_k	Output of the echo path at time sample k.
$\hat{y}_k$	Output of the estimated echo path at time sample k.
z_k	Unattenuated far-end signal at time sample k.
z^{-1}	Unit delay.
τ	Power metric of the cost function.
ξ_k	Cost function at time sample k.
λ_i	ith eigenvalue of the input autocorrelation matrix ϕ_{xx}.
λ_{max}	Maximal eigenvalue of the input autocorrelation matrix ϕ_{xx}.
μ	Step size, or gain of the adaptation process.
ϕ_{xx}	Autocorrelation matrix of the input data sequence.
$\underline{\nabla}$	Gradient vector.

Abbreviations

CCITT	Consultative Committee on International Telegraph and Telephone.
dB	Decibel.
e.g.	*exempli gratia*; for example.
etc.	*et cetera*; and the rest.
FIR	Finite impulse response.
IBM	International Business Machines.
i.e.	*id est*; that is.
IEE	Institution of Electrical Engineers.
IEEE	Institution of Electronic and Electrical Engineers.
IIR	Infinite impulse response.
IRE	Institution of Radio Engineers.
km	Kilometer.
LMS	Least mean square.
max	Maximum.
ms	Milliseconds.
MHz	Mega Hertz.
no.	Number.
RLS	Recursive least squares.
Rx	Receiver.
Tx	Transmitter.
vol.	Volume.

Chapter 1

Introduction

This thesis attempts to suggest ways to increase the initial convergence rate of the adaptive algorithms in adaptive filtering applications. Applications of the modified stochastic gradient algorithms in digital data echo cancellation in digital data transmission over telephone lines is exhibited by means of computer simulations. For a digital data echo canceller, adaptation is a part of the initialisation period during which transmission of useful data is not possible. Thus it is desirable to decrease the adaptation time i.e. to increase the convergence rate of the adaptation process. Present techniques for minimising the initial adaptation time involve protocols in which the far-end signal is switched off during initialisation. The intention here is to develop algorithms with higher convergence rates in the presence of far-end signal, so that these kinds of protocols are no more required.

A new cost function $E[|e_k|^\tau]$ [1] where $\tau \geq 2.0$ [1] and is a rational number is chosen for investigation using stochastic gradient methods. A well known

[1] e_k is the error between the estimated and the desired values at kth time sample.

stochastic gradient least mean square (LMS) algorithm with cost function $E[|e_k|^2]$ is widely in use in adaptive signal processing including echo cancellation [2]. The LMS algorithm, or the mean square error criterion in general, is optimum for Gaussian data. That means there is no room for further improvements in a Gaussian environment.

However, with the emergence of computer networks and an ever increasing number of computer users, the demand for transmission of digital data over telephone lines is increasing continuously. Digital data streams, e.g. +1 & -1 in binary form, are non-Gaussian. Thus the LMS algorithm becomes nonoptimal for digital data. We therefore, investigate error criteria other than the mean square one, for convergence rate improvements. Error criteria like $|e_k|^4$, $|e_k|^6$ and so on have already been suggested in the literature [3]. They all have integer powers only, and tend to suffer from instability because of large gradients. We, on the other hand, are suggesting a general error criterion which is no more restricted to integer powers. In principle, the suggested error power is any rational number.

Some other kind of algorithms have been investigated in the past for use in digital data communication applications including echo cancellation. It is the LMS algorithm which is mostly in use due to its numerical robustness and simplicity in implementation using transversal filters [2]. Recursive least squares (RLS) algorithms have much higher convergence rates even with the mean square function but they tend to be numerically unstable. We therefore, have confined this thesis to stochastic gradient methods only.

In this thesis we have developed stochastic gradient algorithms based on the general error criteria $|e_k|^\tau$, where τ is a rational number greater than 2.0. The algorithm is developed, analysed and then implemented for digital data echo

cancellers. Three different echo path models are used for computer simulations. Two echo path models comprise of decaying impulse responses with two different time constants, whereas the third echo path model is a ringing sequence of filter coefficients. Echo of the near-end signal is assumed to be mixed with the incoming far-end signal. The far-end signal is assumed to be attenuated between 15 and 30 dBs compared to the near end signal or echo of the near-end signal. For computer simulations we have used four attenuation levels of the far-end signal, which are -15, -20, -25 and -30 dB. During simulations convergence was always achieved 20 dB below the far-end signal level. The effects of dispersion in addition to the attenuation in the far-end signal are also observed with computer simulations. For the new adaptation algorithms, computer simulations were performed on data transmission systems with two levels as well as four levels. Simulation results indicate large improvements in initial convergence time.

The thesis structure is straightforward and easy to follow. The second chapter provides the necessary background material. It is mainly divided into two parts, adaptive filtering and echo cancellation. The adaptive filtering sections help in understanding adaptive filters, their structures, implementations, etc. The echo cancellation sections help in understanding the echo process, echo canceller structures and implementation of adaptive filtering in echo cancellation.

The third chapter starts with the actual development and derivation of the stochastic gradient algorithms. It then verifies the proposed algorithms analytically as well as performing some other mathematical analysis including determination of the theoretical boundaries of the convergence coefficient or the step size. Chapter four describes in detail all kinds of modelling for computer simulations. This includes, modelling of the binary and quaternary data sequences and of various echo paths. Setup for simulations, performance measure and

simulation procedures are discussed at the end of the chapter.

The fifth and sixth chapters provide the results of the computer simulations by applying the newly developed stochastic gradient adaptation algorithms in digital data echo cancellers. Chapter five deals with the binary data whereas in chapter six, four level data results are presented. These chapters start with comparing the results of various echo path models for particular far-end signal levels and then comparing the results of various far-end signal levels for particular echo path models. They also include the results of switched gradient simulations and the switching behaviour for a particular case. In short these two chapters summarise experimental results and prove the validity of the proposed adaptation algorithms. Chapter five also incorporates and analyses the calculated and measured step sizes of the convergence process It also includes some results where dispersion is added to the far-end signal in addition to the attenuation.

Chapter seven summarises the conclusions and suggests areas of further investigation. After the references, some numerical values, computer codes used for simulations and published material from this work, are added in appendices.

Chapter 2

Adaptive echo Cancellation

2.1 Introduction

This chapter provides the necessary background material for the work presented in this thesis. Adaptive filters are described briefly at the start. A few uses of adaptive filters, and adaptive filter design are discussed. After introducing the concept of echo, its sources and mechanisms of generation in the telephone system are discussed. The need for cancelling the echoes and various methods used for this purpose are also discussed. A relatively new and widely used technique is echo cancellation. Echo cancellation is used in both voice and data echo cancellers.

2.2 Adaptive Filters

Adaptive filters [4]–[11] are so common in modern signal processing that the term seldom needs a detailed definition here in this thesis. In short, filters are referred to as devices consisting of a piece of hardware or software capable of extracting information of interest from corrupted data. Adaptation is a self adjusting process which converges towards minimising the error between the output of the filter and the required output, according to a certain criterion.

An adaptive filter consists of two key components. A programmable filter and an adaptation algorithm. Figure 2.1 shows basic construction of an adaptive filter. The adaptation process automatically adjusts the parameters of the programmable filter to maximise the performance at the output of the filter compared with some conditioning input, continuously. The output can be taken from various points of the adaptive filter loop, depending upon its particular application.

Adaptive filtering in control and signal processing has been researched for almost the past four decades. Hundreds of research papers have been published and a number of books [12]–[20] have now appeared.

2.3 Adaptive filter operation

An adaptive filter operates in many different ways to perform modelling, prediction, interference cancelling, etc. Modelling or system identification can further be divided into two types. Direct system modelling and inverse system modelling. These operations are further elaborated next.

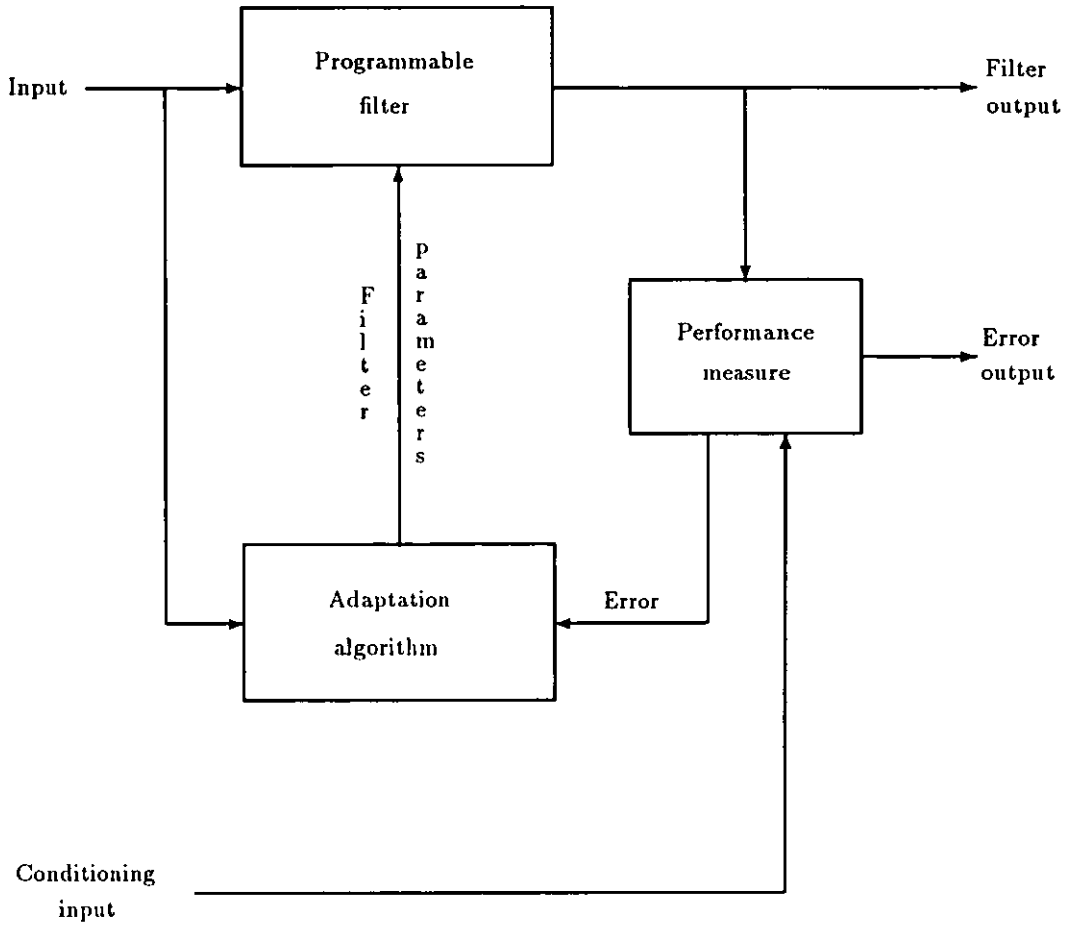


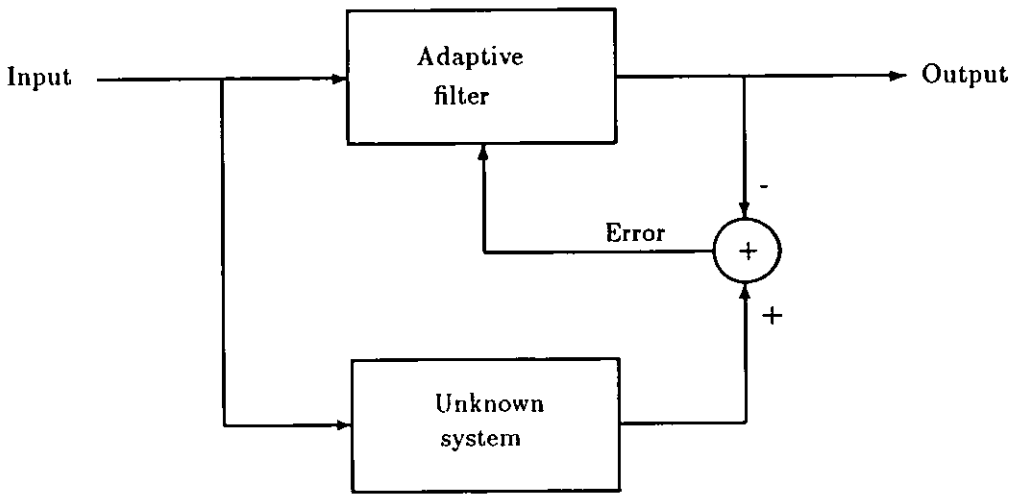
Figure 2.1: Block diagram of the essential components of an adaptive filter.

2.3.1 Direct system modelling

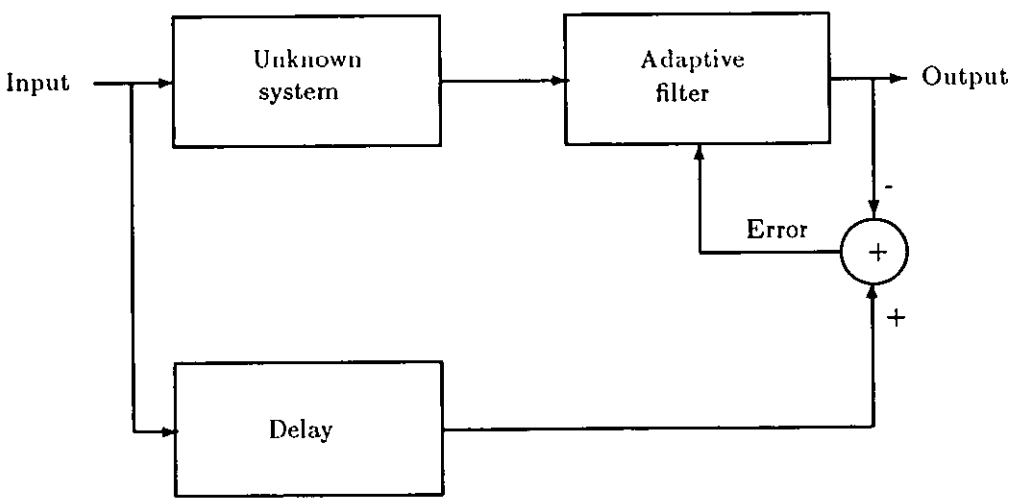
Figure 2.2 (a) shows the direct system modelling approach. The system input is supplied to the adaptive filter as well as to the unknown system with the transfer function $H(z)$. The output of both (adaptive filter as well as the unknown system) is subtracted and the resulting error is supplied to the adaptive filter. The adaptive filter parameters are then adjusted with the help of this error signal, in order to minimise a certain cost function. Adaptation of the filter leads towards convergence in such a way that the adaptive filter coefficients are matched in best possible way to the coefficients of the unknown system filter. Hence modelling of the unknown system with the adaptive filter is achieved. An example is system identification [21, 22]. Another practical example which illustrates the direct system modelling use of the adaptive filter is in echo cancellation in telephone circuits [23]–[27]. Direct system modelling is used in this thesis to model various echo paths.

2.3.2 Inverse system modelling

Figure 2.2 (b) shows the inverse system modelling approach. The signal input is supplied to the unknown system whose output in turn is supplied to the adaptive filter. The output of the adaptive filter is then compared with the delayed signal input. The resulting error is then applied to the adaptive filter. The adaptive filter then adjusts its parameters with the help of this error signal in order to minimise a certain cost function. Adaptation in this way leads towards convergence such that the inverse of the adaptive filter coefficients match the filter coefficients of the unknown system.



(a)



(b)

Figure 2.2: Modelling of the unknown system using adaptive filters: (a) direct system modelling; (b) inverse system modelling.

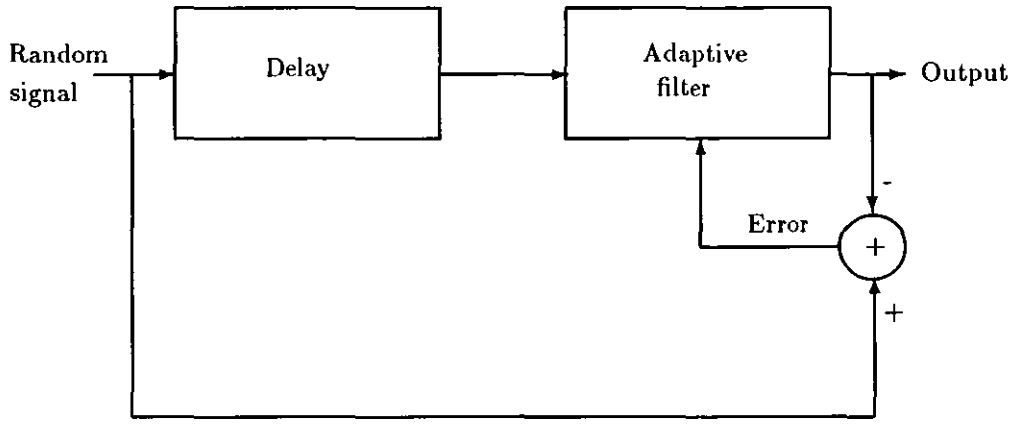
A practical example of such a system is equalisation [28]–[33] in telephone lines where the signal characteristics at the output are changed due to distortions in the communication channel.

2.3.3 Prediction

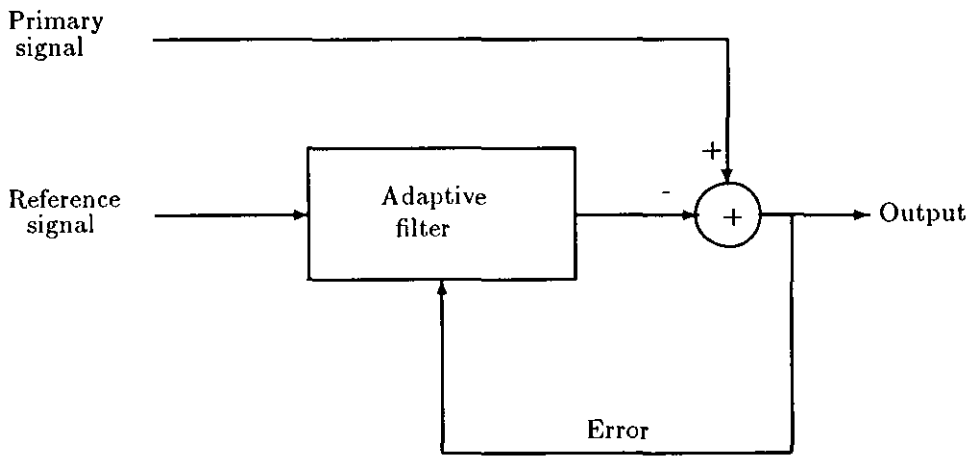
Figure 2.3 (a) shows a predictor. A delayed signal is supplied to the adaptive filter. The output of the adaptive filter, which uses previous signal values for prediction, is compared to the current value of the input signal. The resulting error is then applied to the adaptive filter. The adaptive filter adjusts its parameters to minimise a certain cost function, in order to predict the present value of the random signal in the best possible way. A practical example is the efficient coding of speech signals [34]–[37]

2.3.4 Interference cancelling

Figure 2.3 (b) shows the application of an adaptive filter in interference cancelling. Here the adaptive filter is used to cancel an unknown interference from an information bearing signal. A reference signal derived from a sensor close to the source of interference is supplied to the adaptive filter. The output of the adaptive filter is subtracted from the information bearing corrupted signal (the primary signal). The resulting error signal is used by the adaptive filter to minimise a certain cost function. After convergence, the error signal is the recovered information bearing signal. Practical examples of this configuration include adaptive noise cancelling [38]–[40], mains interference cancelling for medical equipment [38] and echo cancellation in telephone lines [23]–[27]. This configuration is used for echo cancellation in the present thesis as well.



(a)



(b)

Figure 2.3: Adaptive filter configurations for various uses: (a) prediction; (b) interference cancelling.

2.4 Programmable filter designs, IIR and FIR

Programmable filters [41] are of many types that can be used in the design of adaptive filters. Two basic designs of programmable filters namely, recursive or IIR (infinite impulse response) filter and nonrecursive or FIR (finite impulse response) filters are discussed here.

2.4.1 Recursive IIR filters

Design of a recursive filter [42, 43] that is also the most generalised digital filter structure [13] is shown in Figure 2.4. This comprises both feedforward and feedback coefficients or multipliers. The response of this n-stage filter is governed by the nth order difference equation which shows that the value of the present filter output is given by a linear combination of the weighted present and past input values as well as the past output values. This structure results in a pole-zero filter design. The number of poles and zeros, or order of the filter, is given by the number of delay stages. Recursive integrated filters compatible with digital telephony systems are commercially available [44].

This recursive filter is referred to as an infinite impulse response (IIR) filter, as it has, theoretically an infinite memory. It could well be unstable if restrictions are not placed on the feedback coefficients. Adaptive IIR filters are applied in equalisation and in reduction of multipath interference in high frequency (3 to 30 MHz) digital communication channels, where their high speed of convergence is of primary importance [13].

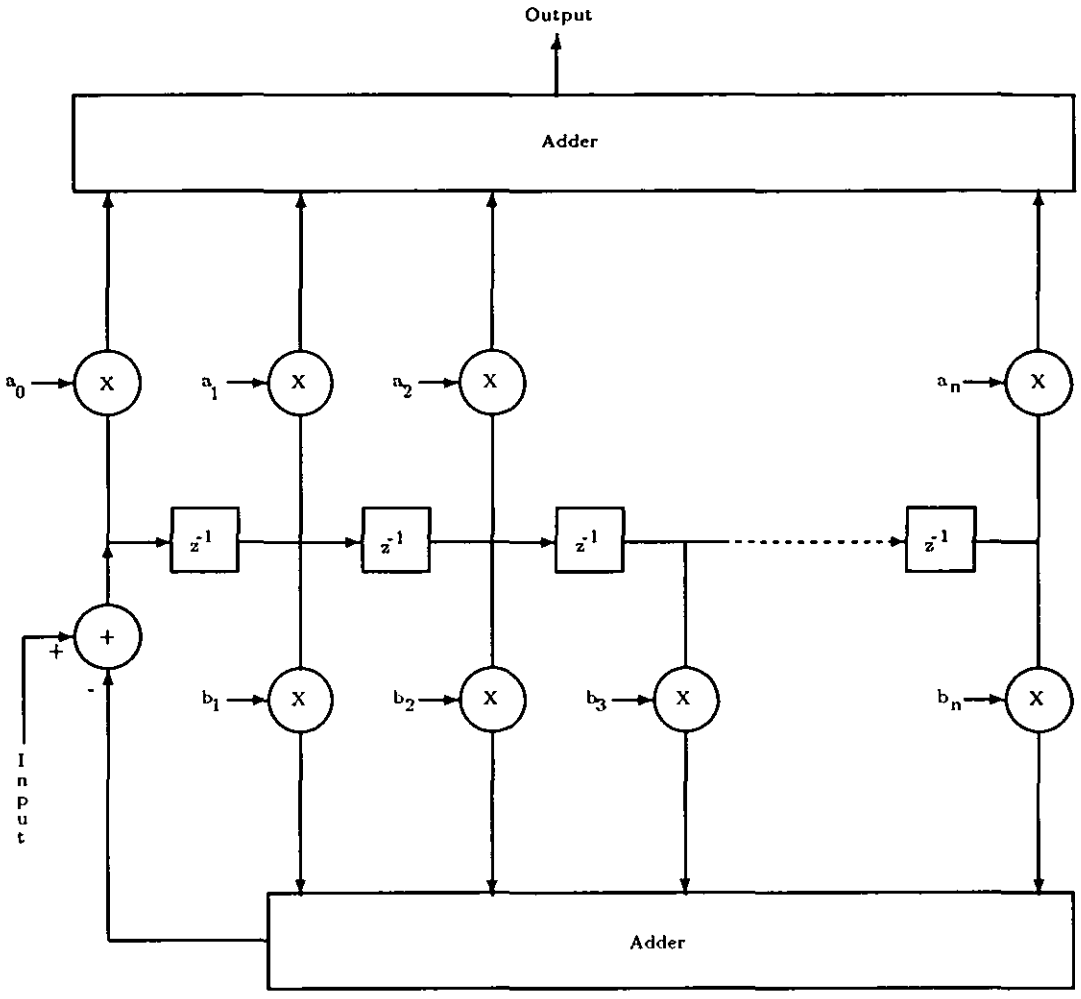


Figure 2.4: Structure of infinite impulse response (IIR) recursive filter. $\{a_n\}$ are feedforward and $\{b_n\}$ are feedback coefficients. z^{-1} are unit delays.

2.4.2 Nonrecursive FIR filters

The potential instability in the IIR design comes from the feedback coefficients. One way to overcome this drawback is to design the filter with feedforward coefficients only, as an all zero filter, as shown in Figure 2.5. This has only a limited memory which is controlled by the number of delay stages and is called a finite impulse response (FIR) or the transversal filter design [45]–[50]. The input signal is delayed by a number of delay elements. Output of the delay elements is multiplied with the feedforward multipliers or filter coefficients or stored weights or impulse response values. These products are then summed to form the output signal. The filter is always stable as there are no feedback elements. FIR filters are relatively simple to design and construct. They are most widely applied in telecommunications applications of adaptive filters such as equalisation [28] and echo cancellation [49].

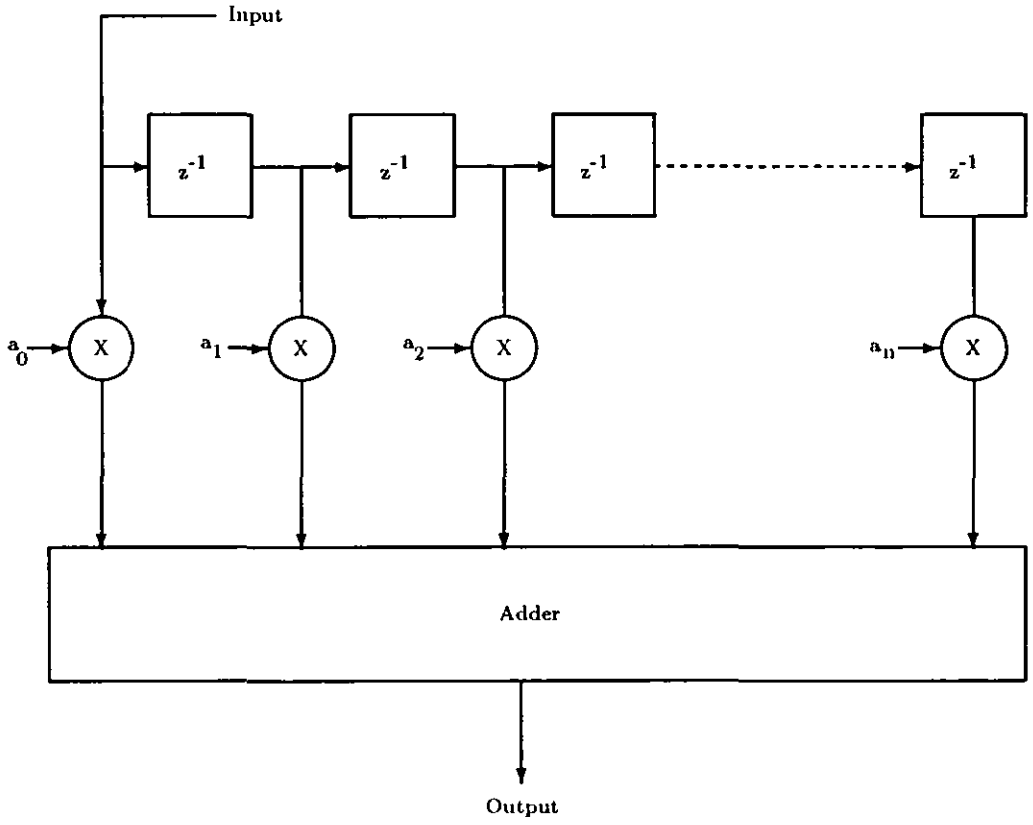


Figure 2.5: Structure of finite impulse response (FIR) nonrecursive filter. $\{a_n\}$ are feedforward coefficients. z^{-1} are unit delays.

2.5 Cost functions

A cost function provides a quantitative measure for assessing quality of performance. The concept of the cost function is basic to adaptive filtering. A cost function defines a transformation from a vector space spanned by the elements of the coefficient vector into the space of a real scalar [16]. A highly popular cost function is the mean square error criterion. It is defined as the mean square value of an estimation error e_k.

$$\xi_k = E[|e_k|^2] \tag{2.1}$$

The basis of famous Wiener filters is to minimise this mean square error criterion. Some other error criteria have also been suggested in literature [51]–[59]. Error criteria other than the mean square one are generally ignored by contemporary researchers as they are difficult to analyse mathematically. A few more cost functions are the absolute error criterion

$$\xi_k = E[|e_k|] \tag{2.2}$$

and the nonlinear threshold error criterion.

$$\xi_k = \begin{pmatrix} 0 & \text{if} & |e_k| < l \\ m & \text{if} & |e_k| \geq l \end{pmatrix} \tag{2.3}$$

where l and m are arbitrary numbers. In this thesis we suggest a cost function of the form

$$\xi_k = E[|e_k|^\tau] \tag{2.4}$$

where $\tau \geq 2.0$.

Trajectories of $|e_k|^\tau$ against $|e_k|$, for selected values of τ are shown in Figure 3.1. The slopes of $|e_k|^\tau$ increase sharply for higher values of $|e_k|$ and higher

values of τ. Whereas, slopes of $|e_k|^\tau$ decrease slowly for lower values of $|e_k|$ and higher values of τ. This provides a clue that convergence may be faster with high values of τ.

2.6 Significance of echo

Normally all conversations take place in the presence of echoes. We hear echoes of our own speech when the signal is reflected from floor, walls and other objects present in the surroundings. If the time delay between the speech and echo is short, the echo is not noticeable. If, however, the time delay exceeds a few tens of milliseconds, the echo becomes distinct and noticeable. Distinct echoes are invariably annoying and under certain conditions can completely disrupt a conversation.

In the domain of communications, echo can be defined as the reflected portion of the transmitted signal. Echoes may be generated in telephone circuits as well. When a signal at any point in the circuit during its transmission, encounters an impedance mismatch, a portion of the signal is reflected back as an echo. The longer is the delay, the greater is the requirement to attenuate the echo before it can be tolerated. Typically echo which is 11 dB below the original signal or higher requires special treatment, if the round trip delay exceeds 40 msec [25].

Commercial communication satellites came into operation in 1965. A geostationary or synchronous satellite must be placed in an orbit about 36,000 km from the surface of the Earth. One way travel of signal between two earth stations communicating with the satellite is more than about 240 ms or of the order of a quarter of a second. The round trip delay becomes as large as 600

ms if we take into account terrestrial delays as well. Even some long distance calls on Earth exceeds the threshold and produce noticeable echoes.

2.7 Sources and mechanism of echo generation

The echo is an unwanted signal defined as the portion of the transmitted signal which reflects back from a distant point of impedance mismatch. The main source of echoes in the telephone network is the hybrid [2] (also known as hybrid transformer or differential transformer). A typical long haul communication circuit is shown in Figure 2.6. The essential components of a hybrid circuit are as shown in Figure 2.7

Figure 2.6 is the diagram of a full duplex transmission system where box A can transmit to box B and can receive from B simultaneously. In the telephone system, every subscriber is connected to a central office (usually known as telephone exchange) via a single pair of wires. When a local telephone call is made, subscriber A and B are connected through their single pair of wires via central office. During their conversation, the signal travels essentially on a single pair of wires in both directions. For the circuits of the order of 50–60 km or more, a separate pair of wire becomes necessary for transmission in each direction. Firstly, the long distance circuits require repeater amplifiers along their length, which are one-way devices. Secondly, most long distance calls are multiplexed for economic reasons. This means a number of calls use portions of a wide band channel. Multiplexing requires that the signals in the two opposite directions be sent in different slots. A device that connects a two wire (one pair) circuit to

a four wire (two pairs) circuit and vice-versa is called a hybrid. Nearly all the significant echoes in the telephone network are generated at these hybrids. It is possible to observe multiple echoes on a poorly terminated circuit but mostly the significant echo heard by a talker is the first reflection from a distant hybrid.

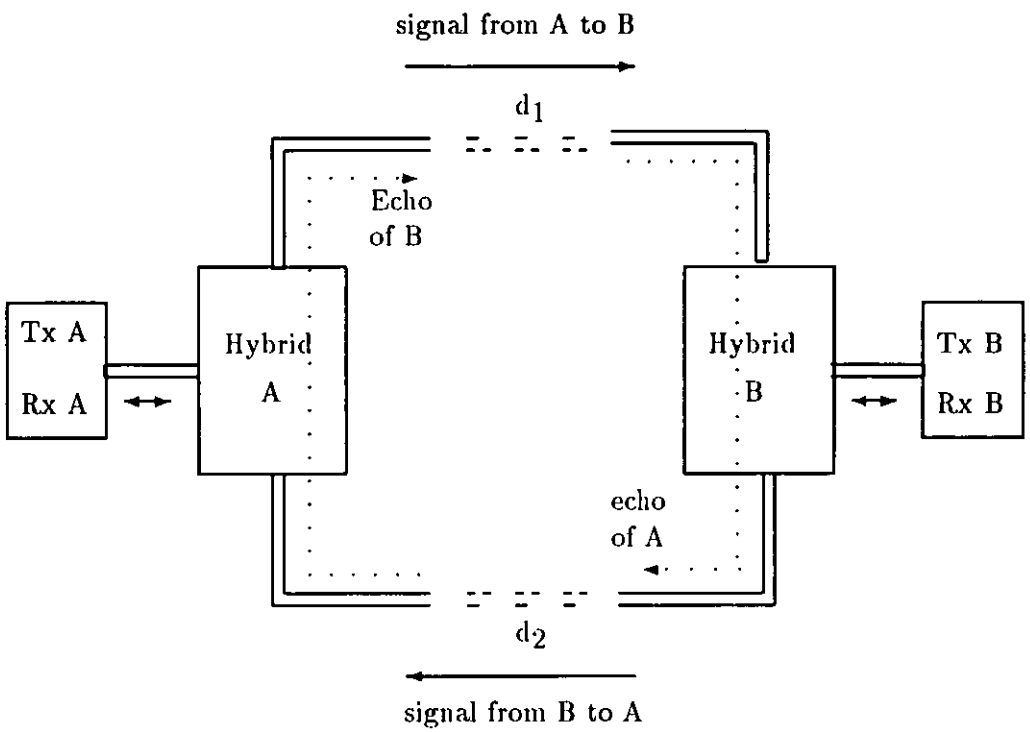


Figure 2.6: Simplified representation of a typical long haul communication link. Each pair of solid lines represent a pair of wires.

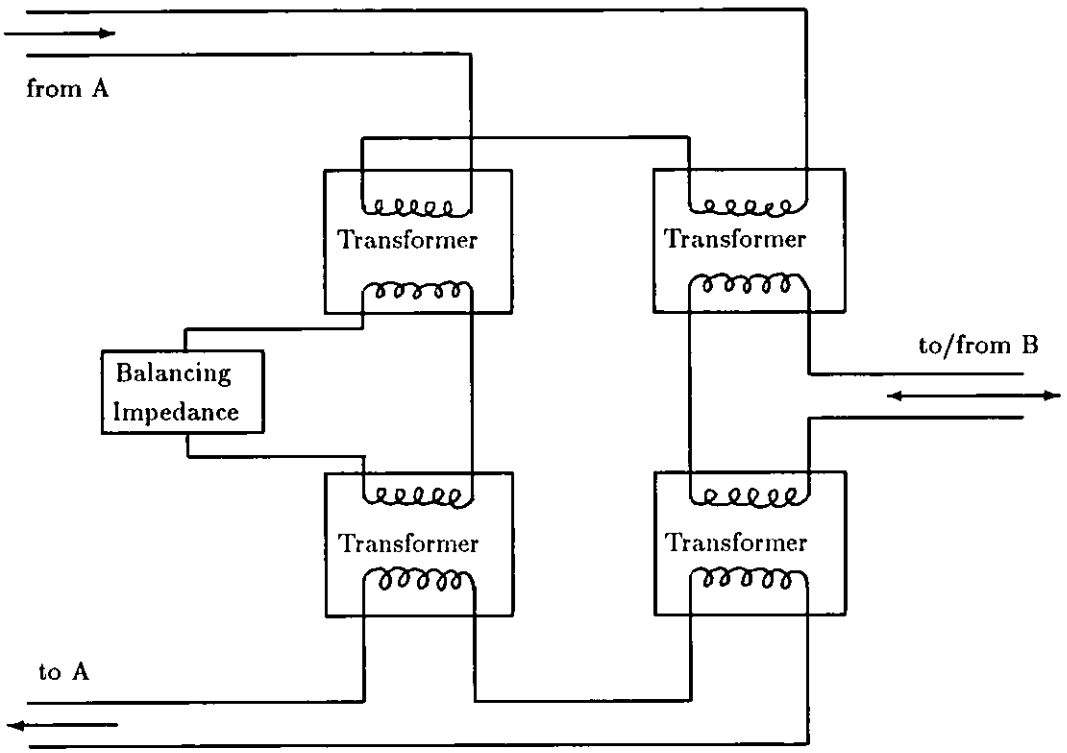


Figure 2.7: Essential components of hybrid circuit located at position B with reference to Figure 2.6. These kind of hybrids are used to convert a two wire circuit into a four wire one and vice versa.

2.8 Methods for cancelling echoes

Different methods are used for reducing the effects of echoes e.g. via net loss, echo suppression and echo cancellation. In the method, via net loss, an equal amount of attenuation is introduced in both lines (one line is one pair of wires). If the attenuation is say G dB, the signal will be attenuated only by G dB but the echo loss will be 2G dB. Hence signal to echo ratio is improved. But at the same time signal level is reduced. If the length of a circuit of this kind exceeds 3000 km or so, the method becomes very inefficient as the received signal levels are very low [60].

A second method is known as echo suppression [60, 61] which has been widely used in telephone networks until the last decade. This method takes advantage of the dynamics of the human speech. Both A and B do not speak simultaneously during conversations most of the time. So when A is speaking the switch in the line from A to B is closed and a one way connection is established for the listening of B. At the same time the switch in the line from B to A is opened because B is not speaking. In this way there is no return path for the signal which reflects from the hybrid near B. As A stops speaking and B starts speaking the switch in the line B to A is closed and a connection is established from B to A and the switch in the line A to B is opened as A is not speaking. A problem occurs when both speakers start speaking at the same time or one interrupts the other. There is a 20 percent probability for one speaker to interrupt the other [62]. A comparator makes the decision to close a switch and to open the other, primarily on the basis of who was speaking for most of the time during interruption. These type of decisions critically affect the performance of the echo suppressor. Some amount of chopping of

the initial portions of the interrupter's speech is still unavoidable even in the best of echo suppressors. Also echo is not eliminated during interruptions. In spite of all this echo suppressors have been used successfully for over 50 years on terrestrial circuits with round trip delays of up to 100 ms [26]. The performance of echo suppressors degrades when they are used over very long distances such as satellite links with round trip delays of the order of 600 ms, because the interruption rate increases [61], as does the sensitivity to improper operation of echo control devices [26].

The latest method being used to control echoes is called echo cancellation [63]–[71]. This is discussed in detail in the next section.

2.9 Echo cancellation

An echo canceller [72]–[83] is an adaptive device which can synthesise a filter to match the transfer characteristics of the echo path. The first full exposition of the principles of adaptive echo cancellation appeared in [83]. A block diagram of an echo canceller is shown in Figure 2.8.

We assume that the channel is linear and is completely specified by its transfer function or the impulse response. A digital transversal filter which consists of a tapped delay line with the number of taps proportional to the round trip delay from the canceller to the hybrid is used for synthesising echo. If we apply an impulse at point a in Figure 2.8, its response at point b will give the impulse response of the echo path, which can then be used to set the tap values of the filter.

The echo path changes continuously during the course of a conversation in a digital telephone network or during the course of data transfer in the case

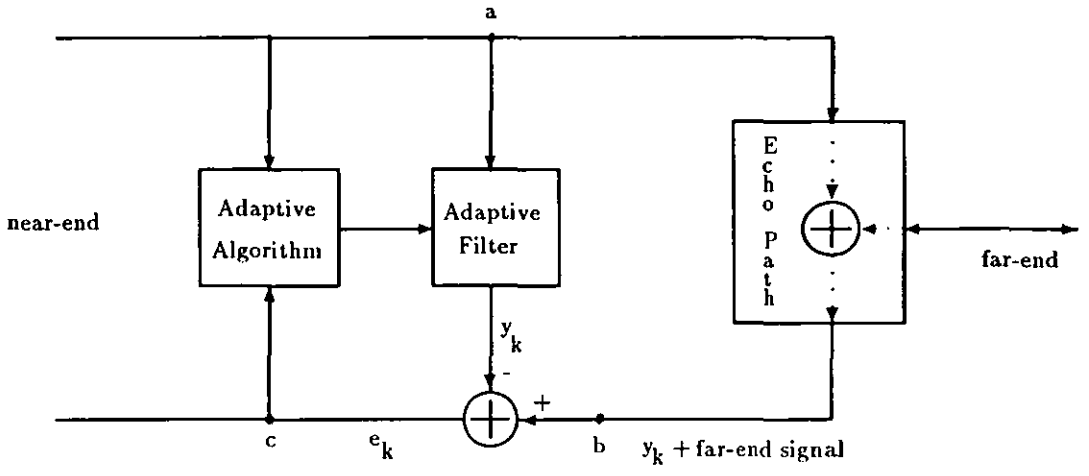


Figure 2.8: Block diagram of an adaptive echo canceller.

of a full duplex digital data transmission system. It is then required to measure the impulse response to update the filter co-efficients. This type of echo canceller can be termed as adaptive echo canceller.This open loop procedure requires frequent measurements of the impulse response for adjustments. The continuous transmission of test pulses is undesirable. The speech or data signal itself can be used in place of test signals. An estimate or replica of echo is generated by synthesising a linear approximation to the echo path through the filter. Which is then subtracted from the returned signal. As a result an error signal is generated, which is used by an adaptive algorithm to modify and update the tap-weights of the adaptive filter (or the filter co-efficients). This type of echo canceller can be termed a self-adaptive echo canceller [27], because it automatically tracks any variations in the echo path. In the present thesis we are concerned only with digital data echo cancellers.

A digital transversal filter of the type used in echo cancellers as an adaptive

filter with adjustable tap weights is as shown in Figure 2.5.

2.10 Echo cancellation for full duplex data transmission

Echoes are generated in the telephone network at the hybrids irrespective of the nature of the signals being transmitted. i.e. whether the signal is human voice or digital data. In case of human speech a small amount of echo or an echo with a small delay can be tolerated. Even a certain amount of echo known as sidetone is necessary to avoid the dullness in conversation over a telephone network. But digital data transmission is a different case. No sidetones are required and even small amount of echoes with a very short delay are also intolerable. Full duplex digital data can be transmitted on a four wire private line, one pair for transmission in each direction. Connections for full duplex transmission of digital data can also be established by dialing through the switches of the telephone network over a single pair of cables. A local hybrid is required at the subscriber's premises to convert the four wire circuit of the data transmitter/receiver to the two wire circuit of the telephone network and vice versa. Echoes may also be generated at this local hybrid.

Although basically the canceller in this case is similar to the one required for human speech (i.e., a transversal adaptive filter with tap weights updated regularly along the gradient of some cost function of the error), there are some important differences worth noting [26]. The first difference is the placement of the echo canceller in the circuit. Voice echo cancellers are part of the telephone network and are not required to be placed at the subscriber's premises. Where

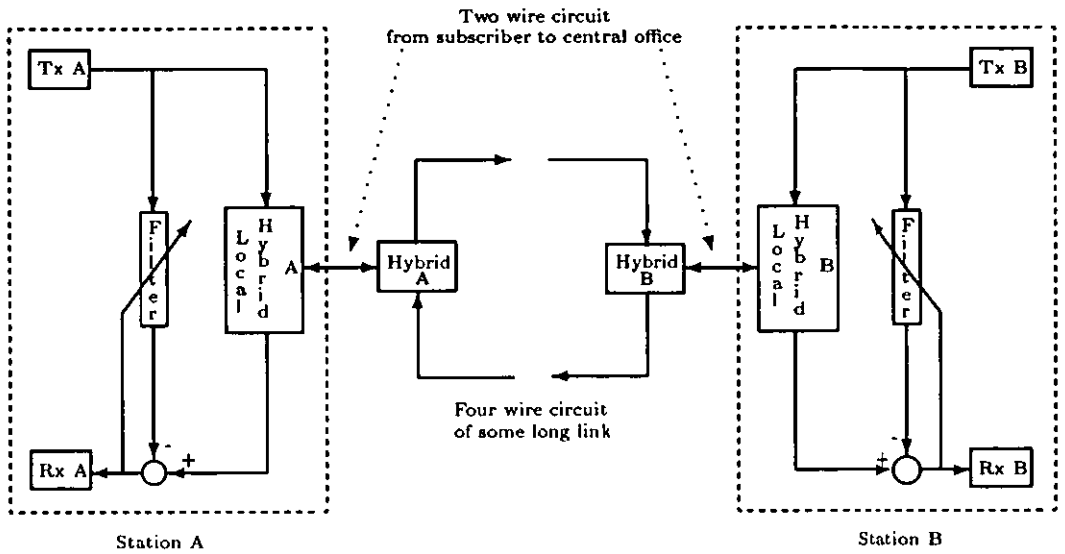


Figure 2.9: Placement of data echo canceller in case of full duplex digital data transmission over single pair of cables in the telephone network

as the data echo canceller would be placed at the subscriber's premises as part of the data (transmit/receive) equipment. Placement of the echo canceller for data transmission is as shown in Figure 2.9. With reference to Figure 2.9 when A transmits data, the first echo received is that from its local hybrid, followed by the ringing in the two wire telephone circuit between A's local hybrid and the hybrid at A's central office. This early echo is quite harmless during voice communications. But for data transmission this is as harmful as the delayed echo. This early echo will remain there even if the four wire circuit (for long haul communications) is absent. To counter this problem, data echo canceller must be placed at the data equipment. Another difference arises from the necessity of placing the echo canceller at the data equipment. In addition to the early echoes there is delayed echo as well. This problem can be solved by splitting the filter

into two adjustable transversal filters separated by a bulk delay [26]. The third difference is in the properties of the signals used for speech and digital data. The statistical properties of the speech signals are complicated and difficult to quantify. They can be described in broad terms only. e.g., bandwidth, the range of fundamental frequencies that might be encountered, etc., whereas data signals have much simpler statistical properties. The sequences of data symbols x_k (selected from some alphabet, e.g., ± 1 for binary sequences) may be assumed to be sequences of independent and identically distributed variables.

The stochastic gradient or least mean square (LMS) algorithm is the most popular practical adaptation algorithm in current use. For a speech echo canceller where the canceller is not dedicated to a particular subscriber, adaptation must occur for each new call. Speech quality may be significantly degraded during the adaptation period. In the data echo canceller case, adaptation is part of the initialisation period, during which transmission of useful data is not possible. Thus it is desirable to decrease the adaptation period, i.e. to increase the speed of convergence of the adaptation algorithm. Also adaptation is very slow in the presence of far-end signaling. Recent research [84, 85] shows that the use of an adaptation algorithm based on the nonquadratic cost function $|e|^\tau$ in general could be advantageous. This work investigates this area further to find out the range of power metric τ for the highest convergence rates possible for various levels of convergence and far-end signals power.

Chapter 3

Proposed adaptive algorithm

3.1 Introduction

The newly proposed adaptation algorithm (Section 3.2) is presented and mathematically derived in this chapter. Mathematical analysis of the newly proposed adaptive algorithm is carried out. Conditions for the convergence of the mean of the tap error vector are derived along with the derivation of the stability limits of the step size μ. Conditions for the convergence of the variance of the tap error vector have also been derived along with the stability limits for the step size μ. The proposed algorithm is very similar to the famous least mean square (LMS) [86]–[88] algorithm. The LMS algorithm uses a power metric of 2. Whereas, in the proposed algorithm, non quadratic power metrics such as 2.1, 2.2, ...etc. are used.

3.2 Proposed adaptive algorithm

Various adaptation algorithms are in use in adaptive filtering applications. Stochastic gradient [1, 3, 25, 54, 89, 90] algorithms remain popular because of their numerical robustness and simplicity in calculation, even though they are slow in convergence. To increase the efficiency of the digital data transmission over telephone lines, it is important to increase the convergence rate of the adaptation process in digital data echo cancellers. The mean square function is optimal for Gaussian signals. However we are dealing with the non-Gaussian signals where mean square function is non-optimal. Thus we propose a cost function ξ_k to be minimised as:

$$\xi_k = E[|e_k|^\tau] \quad \text{where} \quad \tau \geq 2.0$$

In principle, τ could be any rational number greater than or equal to 2.0 as long as adaptation algorithm remains convergent. For simulation purposes we took the values of τ with an interval of 0.1 so that, $\tau = 2.0, 2.1, 2.2, \ldots$. Figure 3.1 gives a visual impression of the gradients of $|e_k|^\tau$ for some selected values of τ.

The proposed adaptation algorithm can be obtained by replacing the mean square function in the LMS algorithm with the above mentioned non-Euclidean cost function. The new algorithm can be written as:

$$\hat{\mathbf{h}}_{k+1} = \hat{\mathbf{h}}_k + \mu\tau\mathbf{x}_k|e_k|^{\tau-1}\text{sign}(e_k)$$

The variables and parameters are defined in Section 3.3. The above is essentially a general form of the LMS algorithm. We can easily obtain the LMS algorithm by replacing τ with 2 in the newly proposed algorithm. Hence the LMS algorithm becomes a special case of the proposed algorithm.

The adaptation process will lead to quicker convergence as τ is increased beyond 2 as long as the convergence factor μ remains within the stable limits. The proposed adaptation algorithm is mathematically derived in the next section.

Mathematical analysis with respect to stability, convergence, time constants, etc. are carried out in this chapter. To practically show the advantages of the new algorithm, computer simulations were performed. Three different echo path models, in the simulations with binary as well as quaternary data sequences, were used. Details of the experiments and their results are discussed in later chapters.

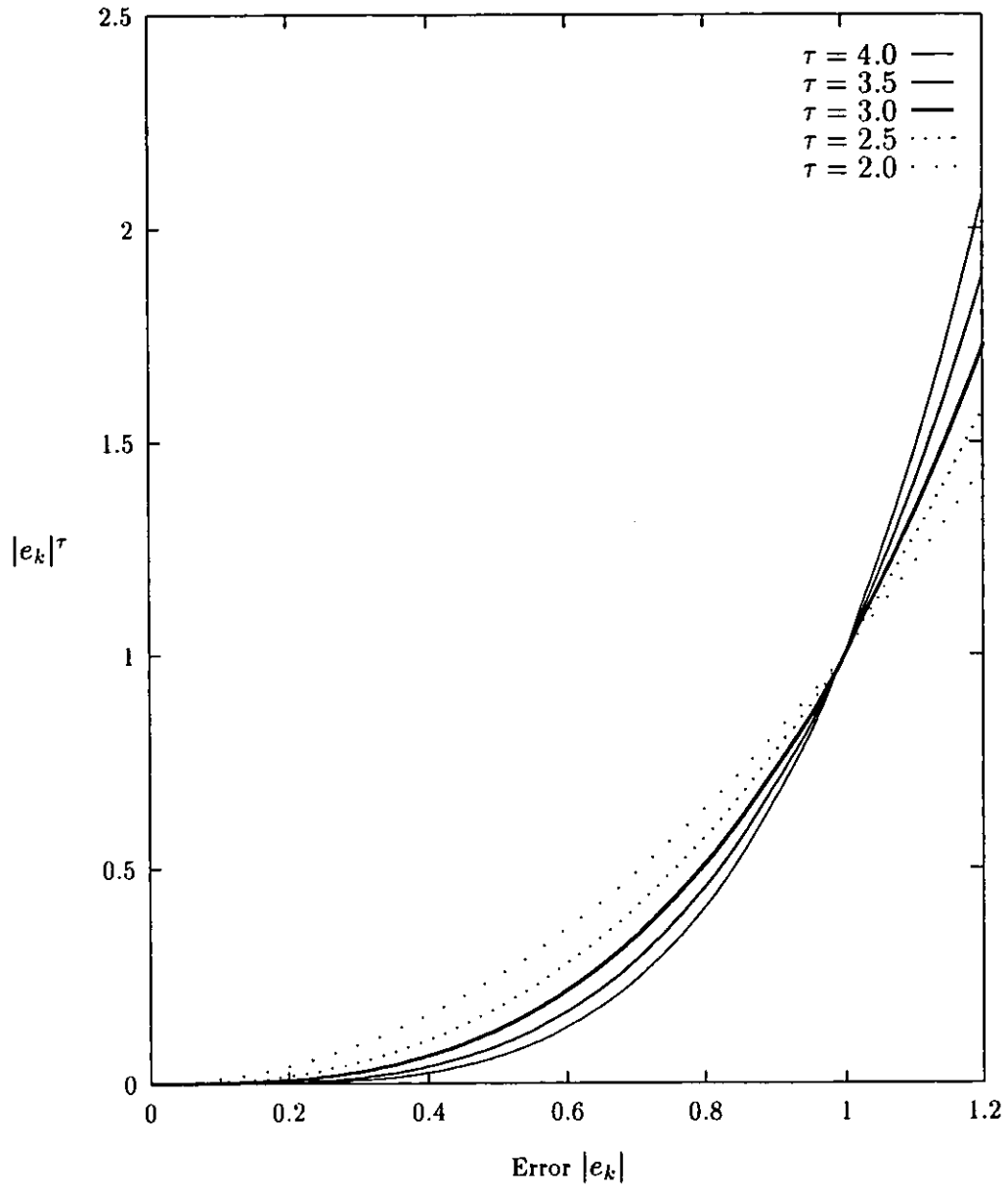


Figure 3.1: $|e_k|^\tau$ is plotted versus $|e_k|$ for various values of τ. Different slopes of curves at both sides of the crossover point $|e_k| = 1$ are evident.

3.3 Derivation of the adaptation algorithm

With reference to Figure 4.4, the error signal e_k in terms of the echo path output y_k, far-end signal f_k and synthetic echo signal $\hat{y}_k$ is

$$e_k = y_k + f_k - \hat{y}_k \tag{3.1}$$

The output of the estimated echo path $\hat{y}_k$ can be written as:

$$\hat{y}_k = \sum_{n=0}^{N-1} \hat{h}_n x_{k-n} = \hat{\mathbf{h}}_k^t \mathbf{x}_k = \mathbf{x}_k^t \hat{\mathbf{h}}_k \tag{3.2}$$

Where N is the number of filter co-efficients, $\hat{h}_n$ is the value of the nth estimated co-efficient of the filter at kth time sample and x_{k-n} is the stored value of the input sequence (near-end signal) n symbols before.

$$\hat{\mathbf{h}}_k = [\hat{h}_0 \hat{h}_1 \hat{h}_2 \ldots \hat{h}_{N-1}]^t \quad \text{at } k\text{th time sample}$$

$$\text{and} \quad \mathbf{x}_k = [x_k x_{k-1} x_{k-2} \ldots x_{k-N+1}]^t$$

The cost function ξ_k to be minimised is:

$$\xi_k = E[|e_k|^\tau] \quad \text{where } \tau \geq 2.0 \tag{3.3}$$

The minimum value of the cost function can be obtained by differentiating (3.3) with respect to each element of the impulse response $\mathbf{h}$ of the adaptive filter and then setting all the partial differentials equal to zero.

$$\frac{\delta \xi}{\delta \mathbf{h}} = E\left[\frac{\delta}{\delta \mathbf{h}}(|e_k|^\tau)\right] = \mathbf{0} \tag{3.4}$$

Equation (3.4) can be differentiated by substitution.

$$\text{Let } u = |e_k| \text{ and } y = u^\tau$$

$$\text{Thus } \frac{dy}{du} = \tau u^{\tau-1}$$

$$\text{and } \quad \frac{du}{de_k} = \begin{pmatrix} 1 \text{ if } e_k > 0 \\ -1 \text{ if } e_k < 0 \end{pmatrix}$$

It is obvious from the above equation that du/de_k can also be written as $\text{sign}(e_k)$. Applying the chain rule of differentiation we can obtain:

$$\frac{dy}{de_k} = \frac{dy}{du}\frac{du}{de_k} = \tau u^{\tau-1}\text{sign}(e_k)$$

Substituting $u = |e_k|$ we get,

$$\frac{dy}{de_k} = \tau |e_k|^{\tau-1}\text{sign}(e_k)$$

Thus,

$$\frac{\delta\xi}{\delta \mathbf{h}} = E\left[\tau |e_k|^{\tau-1}\text{sign}(e_k)\frac{\delta}{\delta \mathbf{h}}e_k\right]$$

$$= E\left[\tau |e_k|^{\tau-1}\text{sign}(e_k)\frac{\delta}{\delta \mathbf{h}}(y_k + f_k - \mathbf{h}^t\mathbf{x}_k)\right]$$

$$\frac{\delta\xi}{\delta \mathbf{h}} = E\left[-\tau |e_k|^{\tau-1}\text{sign}(e_k)\mathbf{x}_k\right] = \mathbf{0} \tag{3.5}$$

$\delta\xi/\delta\mathbf{h}$ is the set of N differential terms. All these terms can be collected to form a vector known as the gradient vector $\underline{\nabla}$. Equation 3.5 can also be written in matrix form as below.

$$\underline{\nabla} = \begin{pmatrix} \delta\xi/\delta h_0 \\ \delta\xi/\delta h_1 \\ \vdots \\ \delta\xi/\delta h_{N-1} \end{pmatrix} = -\tau E \begin{pmatrix} x_k|e_k|^{\tau-1}\text{sign}(e_k) \\ x_{k-1}|e_k|^{\tau-1}\text{sign}(e_k) \\ \vdots \\ x_{k-N+1}|e_k|^{\tau-1}\text{sign}(e_k) \end{pmatrix} \tag{3.6}$$

$$\underline{\nabla} = -\tau E\left[\begin{pmatrix} x_k \\ x_{k-1} \\ \vdots \\ x_{k-N+1} \end{pmatrix} \times |e_k|^{\tau-1} sign(e_k)\right]$$

$$\underline{\nabla} = -\tau E\left[\mathbf{x}_k|e_k|^{\tau-1}\text{sign}(e_k)\right] \tag{3.7}$$

The calculation of gradient $\underline{\nabla}$ in Equation (3.7) is not a simple task. This is because it involves the application of the statistical expectation operator E. The knowledge of the signal statistics required in (3.7) can be replaced by a training sequence. The training sequence is available in the case of the echo canceller as the data sequence itself. The co-efficients of the adaptive filter are then calculated and updated by an adaptive algorithm. We use here the method of steepest descent [15]. An initial guess $\hat{\mathbf{h}}_k$ at time sample k is made and the gradient vector $\underline{\nabla}_k$ of the $E[|e_k|^\tau]$ surface is calculated. The gradient $\underline{\nabla}$ given by (3.7) at time sample k becomes the noisy or stochastic gradient $\underline{\hat{\nabla}}_k$, by removing the statistical expectation

$$\underline{\hat{\nabla}}_k = \left[\frac{\delta\xi}{\delta\hat{h}_0}, \frac{\delta\xi}{\delta\hat{h}_1}, \frac{\delta\xi}{\delta\hat{h}_2}, \ldots, \frac{\delta\xi}{\delta\hat{h}_{N-1}}\right]^t_{\hat{\mathbf{h}}=\hat{\mathbf{h}}_k} \tag{3.8}$$

$$\text{Or} \quad \underline{\hat{\nabla}}_k = -\tau\mathbf{x}_k|e_k|^{\tau-1}\text{sign}(e_k) \tag{3.9}$$

To obtain the new estimate $\hat{\mathbf{h}}_{k+1}$ of the adaptive filter co-efficients, a scaled version of the gradient is subtracted from the previous estimate of the adaptive filter co-efficients as follows,

$$\hat{\mathbf{h}}_{k+1} = \hat{\mathbf{h}}_k - \mu \times \underline{\hat{\nabla}}_k \tag{3.10}$$

The small positive scaling constant μ is known as the step-size or gain of the echo canceller. The value of μ plays an important role in determining the convergence speed and stability [71]. By substituting (3.9) into (3.10) we obtain the final result,

$$\hat{\mathbf{h}}_{k+1} = \hat{\mathbf{h}}_k + \mu\tau\mathbf{x}_k|e_k|^{\tau-1}\text{sign}(e_k) \tag{3.11}$$

Equation (3.11) represents the general form of the stochastic gradient algorithm with nonquadratic exponent τ. If we set $\tau = 2$, then (3.11) will represent the well known LMS algorithm.

3.4 Conditions for the convergence of the mean of the tap error vector

We now find the conditions for the convergence of the mean of the tap error vector. We will show that $E[\mathbf{v}_k] \rightarrow \mathbf{0}$ as k tends to infinity, where $\mathbf{v}_k = \hat{\mathbf{h}}_k - \mathbf{h}$. We take a relatively simple case of smaller deviations of $\hat{\mathbf{h}}_k$ from $\mathbf{h}$. We assume that $\mathbf{x}_k$ and f_k are zero mean non-Gaussian signals independent to each other, the autocorrelation matrix ϕ_{xx} of the input signal $\mathbf{x}_k$ is positive definite, and the echo path is stationary. By subtracting $\mathbf{h}$ from both sides of (3.11), we get:

$$\mathbf{v}_{k+1} = \mathbf{v}_k + \mu\tau\mathbf{x}_k \text{sign}(e_k)|e_k|^{\tau-1} \tag{3.12}$$

From (3.1) and (3.2) we have

$$e_k = f_k - \hat{y}_k + y_k = f_k - \mathbf{x}_k^t\hat{\mathbf{h}}_k + \mathbf{x}_k^t\mathbf{h} = f_k - \mathbf{x}_k^t\mathbf{v}_k \tag{3.13}$$

Substituting (3.13) into (3.12)

$$\mathbf{v}_{k+1} = \mathbf{v}_k + \mu\tau\mathbf{x}_k \text{sign}(f_k - \mathbf{x}_k^t\mathbf{v}_k)|f_k - \mathbf{x}_k^t\mathbf{v}_k|^{\tau-1} \tag{3.14}$$

A little manipulation with (3.14) will give,

$$\mathbf{v}_{k+1} \cong \mathbf{v}_k + \mu\tau\mathbf{x}_k \text{sign}(f_k)|f_k|^{\tau-1}\left(1 - \frac{\mathbf{x}_k^t\mathbf{v}_k}{f_k}\right)^{\tau-1} \tag{3.15}$$

$$\text{Since} \quad 1 - \frac{\mathbf{x}_k^t\mathbf{v}_k}{f_k} > 0 \quad \text{because} \quad \left|\frac{\mathbf{x}_k^t\mathbf{v}_k}{f_k}\right| < 1$$

as $\mathbf{v}_k$ is assumed very small. Applying the binomial theorem [91] to (3.15) and ignoring higher order terms, we get

$$\mathbf{v}_{k+1} \cong \mathbf{v}_k + \mu\tau\mathbf{x}_k \text{sign}(f_k)|f_k|^{\tau-1}\left(1-(\tau-1)\frac{\mathbf{x}_k^t\mathbf{v}_k}{f_k}\right) \tag{3.16}$$

$$\mathbf{v}_{k+1} \cong \mathbf{v}_k + \mu\tau\mathbf{x}_k \text{sign}(f_k)|f_k|^{\tau-1} - \mu\tau(\tau-1)|f_k|^{\tau-2}\mathbf{x}_k\mathbf{x}_k^t\mathbf{v}_k \tag{3.17}$$

The autocorrelation matrix ϕ_{xx}, of the input signal is denoted by

$$\phi_{xx} = E\left[\mathbf{x}_k\mathbf{x}_k^t\right] \tag{3.18}$$

The middle term on the right hand side of (3.17) will vanish after taking mathematical expectations of both sides, as $E[\mathbf{x}_k] = \mathbf{0}$. It is assumed that f_k, $\mathbf{x}_k$, and $\mathbf{v}_k$ are independent to each other.

$$E[\mathbf{v}_{k+1}] \cong E[\mathbf{v}_k] - \mu\tau(\tau-1)E\left[|f_k|^{\tau-2}\right]\phi_{xx}E[\mathbf{v}_k] \tag{3.19}$$

$$E[\mathbf{v}_{k+1}] \cong \left(\mathbf{I} - \mu\tau(\tau-1)E\left[|f_k|^{\tau-2}\right]\phi_{xx}\right)E[\mathbf{v}_k] \tag{3.20}$$

Let

$$\boldsymbol{\Delta} \triangleq \mathbf{I} - \mu\tau(\tau-1)E\left[|f_k|^{\tau-2}\right]\phi_{xx} \tag{3.21}$$

Equation (3.21) can be written in canonical form as,

$$\boldsymbol{\Delta} = \mathbf{A}\begin{pmatrix} \delta_1 & & 0 \\ & \ddots & \\ 0 & & \delta_n \end{pmatrix}\mathbf{A}^t$$

$$\delta \triangleq sup|\delta_i| < 1, i = 1,\ldots,n; \quad \mathbf{A}\mathbf{A}^t = \mathbf{I} \tag{3.22}$$

Then (3.20) will become

$$E[\mathbf{v}_{k+1}] \cong \boldsymbol{\Delta}E[\mathbf{v}_k] \tag{3.23}$$

We can now choose μ in such a way, that all the leading diagonal elements of $\mathbf{\Delta}$ in (3.21) will have absolute values less than 1.

$$0 < \mu < \frac{2}{\tau(\tau-1)E\left[|f_k|^{\tau-2}\right]\lambda_i}, \quad i = 1, \ldots, n \tag{3.24}$$

Where λ_i are the eigenvalues of ϕ_{xx}. Equation (3.24) will give n values. The lowest of these values, or the upper bound on μ can be obtained using the maximal eigenvalue λ_{max} of ϕ_{xx}

$$0 < \mu < \frac{2}{\tau(\tau-1)E\left[|f_k|^{\tau-2}\right]\lambda_{max}} \tag{3.25}$$

Let L2 norm of $E[\mathbf{v}_{k+1}]$ be denoted by v_{k+1}, then

$$v_{k+1} \triangleq E[\mathbf{v}_{k+1}^t]E[\mathbf{v}_{k+1}] \tag{3.26}$$

From (3.22), (3.23) and (3.26) we obtain,

$$v_{k+1} = E[\mathbf{v}_k^t]\mathbf{A}\begin{pmatrix} \delta_1^2 & & 0 \\ & \ddots & \\ 0 & & \delta_n^2 \end{pmatrix}\mathbf{A}^t E[\mathbf{v}_k] \leq \delta^2 v_k \tag{3.27}$$

$$\text{Or} \quad v_{k+1} \leq \delta^2 v_k \tag{3.28}$$

Inequality (3.28) suggests that the adaptation process will lead to the convergence $E[\mathbf{v}_k] \rightarrow \mathbf{0}$, when k tends to infinity. This of course is a local convergence and cannot be extrapolated other than the bounds on μ stated in (3.25). The condition set in (3.25) might be difficult to check in practice. We can, however, bound the maximal eigenvalue of a positive definite matrix by its trace, $\text{tr}(\phi_{xx}) = nE[x_k^2]$. So we obtain an easily applied sufficient condition for the convergence of the mean of the tap error vector of the proposed adaptive algorithm:

$$0 < \mu < \frac{2}{n\tau(\tau-1)E\left[|f_k|^{\tau-2}\right]E\left[x_k^2\right]} \tag{3.29}$$

3.5 Time constants

We now proceed to the evaluation of the time constants of the adaptive process. The time constant of a mode of convergence is the time taken by that mode to converge to 36.8% of its initial value. Again we assume that the current estimate of the adaptive filter co-efficients $\hat{\mathbf{h}}_k$ is in the vicinity of the target filter taps $\mathbf{h}$, so that approximation (3.20) holds. For the above we also assume that the vectors $\mathbf{v}_k$ and $\mathbf{x}_k$ are independent of each other. From Equations (3.18) and (3.20) we deduce that generally there will be n different modes of convergence corresponding to the n different values of ϕ_{xx}. Accordingly there will be n different relaxation time constants of the filter taps.

$$T_i = \frac{1}{\mu\tau(\tau-1)E\left[|f_k|^{\tau-2}\right]\lambda i}, \quad i = 1, 2, \ldots, n. \tag{3.30}$$

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

Chapter 4

Modelling

4.1 Introduction

This chapter discusses the simulation setup and procedures for computer simulations. Modelling of binary and quaternary data is discussed in the beginning, followed by the three echo path models used in the simulations of digital data echo canceller. Simulation setup is illustrated with the help of the block diagrams. Performance measure and simulation procedure is described next. Specific computer codes written for these simulations are added in Appendix B.

4.2 Modelling the data streams

The near-end signal sequence x_k is modelled by a non-Gaussian pseudo random bipolar sequence from the set {1,-1}. The far-end signal sequence f_k is also modelled by an independent random bipolar sequence from the set {1,-1}, which is subject to an attenuation f modelling the transmission loss. The far-end signal power level is kept below the near-end signal power level by a certain

amount which dictates the value of f (Appendix A.1).

The four level near-end signal sequence x_k is modelled by a non-Gaussian pseudo random four level sequence from the set {3,1,-1,-3}. Far-end signal sequence f_k is also modelled by an independent pseudo random four level sequence from the set {3,1,-1,-3}, which is subject to an attenuation f modelling the transmission loss.

4.3 Echo path models

The echo path was modelled in two different ways. A single pole single zero model and a numerically generated model. The single pole single zero model was further split into two with different decaying sequences. So we end up with three echo path models as described below.

4.3.1 First model

The echo path is modelled by a single pole and single zero digital filter for simplicity. The zero is placed at $z = 0$ and pole at $z = a$. Thus the transfer function of the echo path is

$$H(z) = \frac{z}{z-a} \quad \text{where} \quad 0 < a < 1 \tag{4.1}$$

i.e. a is positive as well as inside the unit circle in the z-plane.

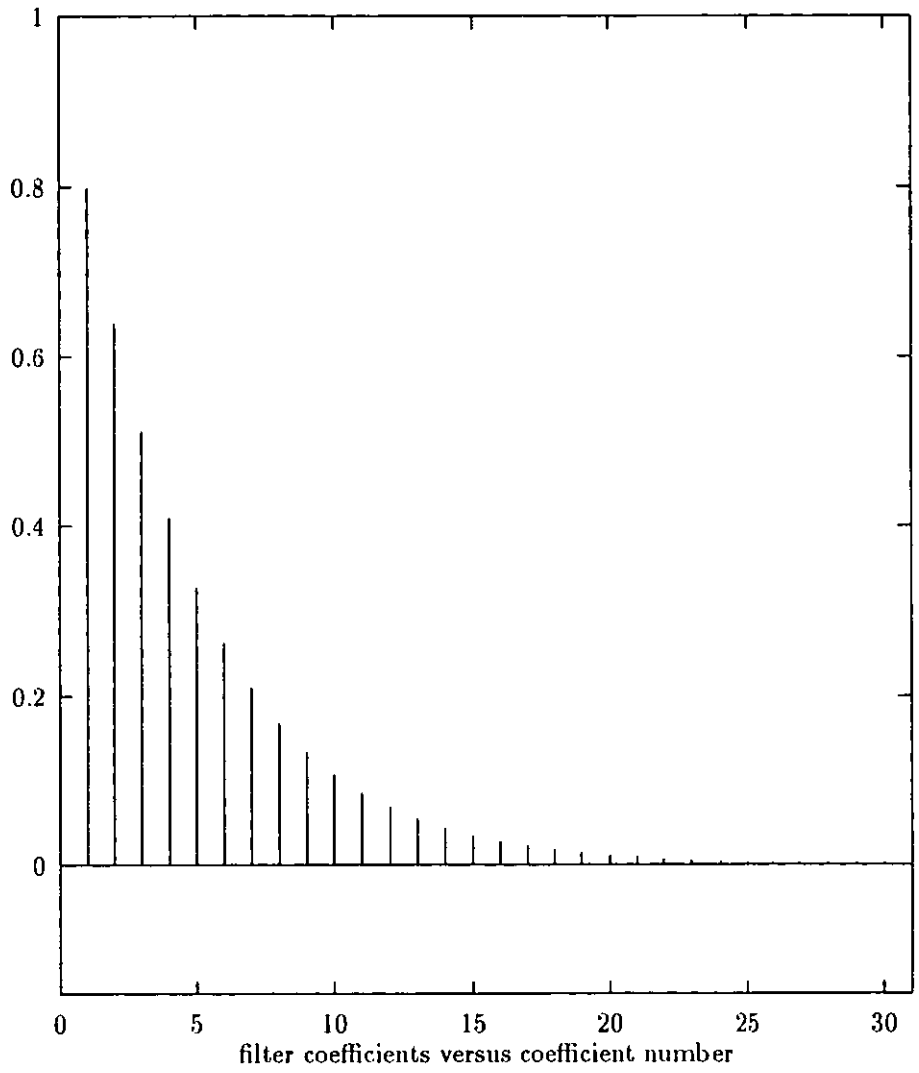


Figure 4.1: Impulse response sequence of the echo path filter for the first echo path model.

The value of a has been chosen in this way to fulfil the conditions of stability. The transfer function of Equation 4.1 can be re-arranged as

$$H_{(z)} = \frac{1}{1 - az^{-1}} = (1 - az^{-1})^{-1} = 1 + az^{-1} + a^2z^{-2} + \cdots \qquad (4.2)$$

The impulse response of the filter is $\{a^k\}$ which is a decaying sequence since $|a| < 1$. The feedback co-efficient of the echo path filter is chosen in such a way that the power level of near-end transmitted signal will be attenuated by 60 dB in 32 samples. The 60 dB power level attenuation is chosen arbitrarily. Since we are using a 32 stage filter, the series is truncated after 32 samples. Figure 4.1 shows the impulse response sequence of the echo path filter for the first model. A calculation of the feedback co-efficient a is given below.

$$20 \log_{10} a^{31} = -60 \text{ dB}$$

$$\text{or} \quad 20 \times 31 \log_{10} a = -60$$

$$\text{or} \quad a = 10^{\frac{-60}{20 \times 31}}$$

$$\text{or} \quad a = 0.800250$$

The transfer function of the echo path becomes

$$H(z) = (1 - 0.800250z^{-1})^{-1}$$

$$= 1 + 0.800250z^{-1} + 0.640400z^{-2} + \cdots + 0.001000z^{-31}$$

The co-efficients of the successive powers of z in the first 32 terms of the series are the co-efficients (tap values) of the echo path FIR filter. These are taken as the target tap co-efficients for the data echo canceller, as shown below.

$$h(0) = 1.00000$$

$$h(1) = 0.800250$$

$$h(2) = 0.640400$$

$$\vdots$$

$$h(31) = 0.00100$$

4.3.2 Second model

The second echo path model used in simulations is also a single pole single zero model as described in the first model description with the transfer function given in Equation 4.1 and Equation 4.2. The only difference is in the choice of the feedback co-efficient a. For the second echo path model, a is chosen such that the power level of the near-end transmitted signal will be attenuated by 120 dB in 32 samples. The filter coefficients thus obtained are as given below.

$$h(0) = 1.00000$$

$$h(1) = 0.640400$$

$$h(2) = 0.410113$$

$$\vdots$$

$$h(31) = 1.00000E - 06$$

The series is truncated after 32 samples here as well. Figure 4.2 shows the impulse response sequence of the echo path filter for the second model.

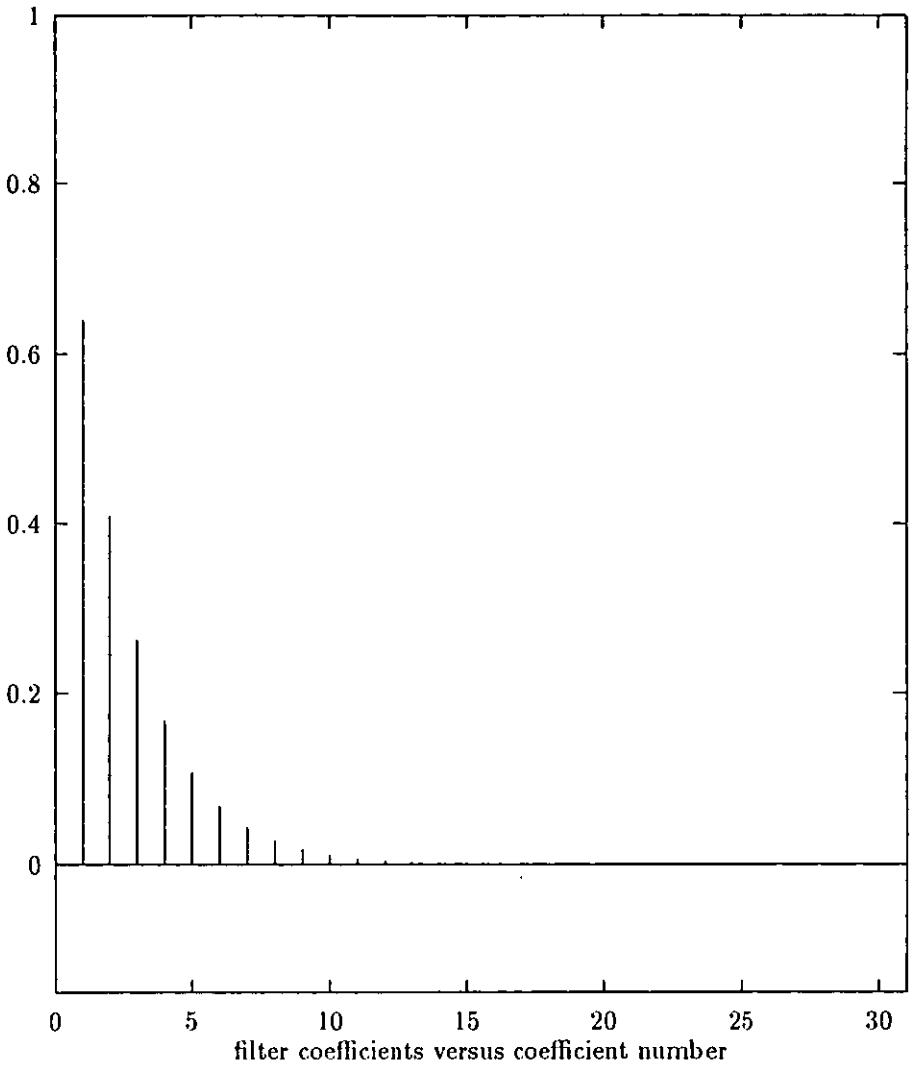


Figure 4.2: Impulse response sequence of the echo path filter for the second echo path model.

4.3.3 Third model

The third echo path model is numerically generated. It is a simplified form of the actual echo impulse responses of various telephone network connections from a British international gateway [2]. First a diagram with ringing near the end, was drawn on a piece of paper. It was then sampled, filter coefficients were measured and recorded for future use. The coefficients thus obtained are as given below.

$$\{h\} = \{1, .985, .978, .955, .929, .895, .858, .820, .774, .715, .650, .575, .505,$$

$$.425, .332, .235, .145, .050, -.048, -.116, -.158, -.186, -.195,$$

$$-.175, -.140, -.085, -.009, .060, .098, .105, .073, 0.0\}$$

The impulse response of the filter used in the third model of echo path is shown in Figure 4.3.

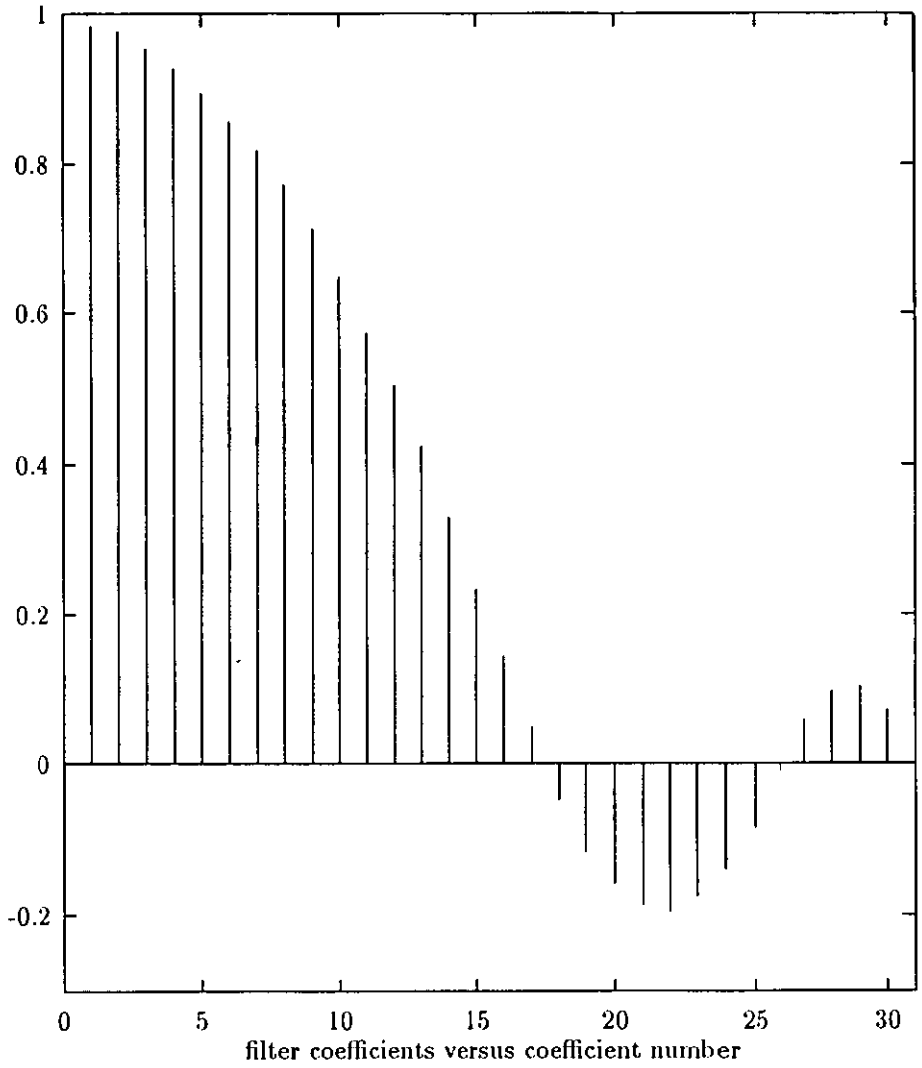


Figure 4.3: Impulse response of the filter used in the third echo path model.

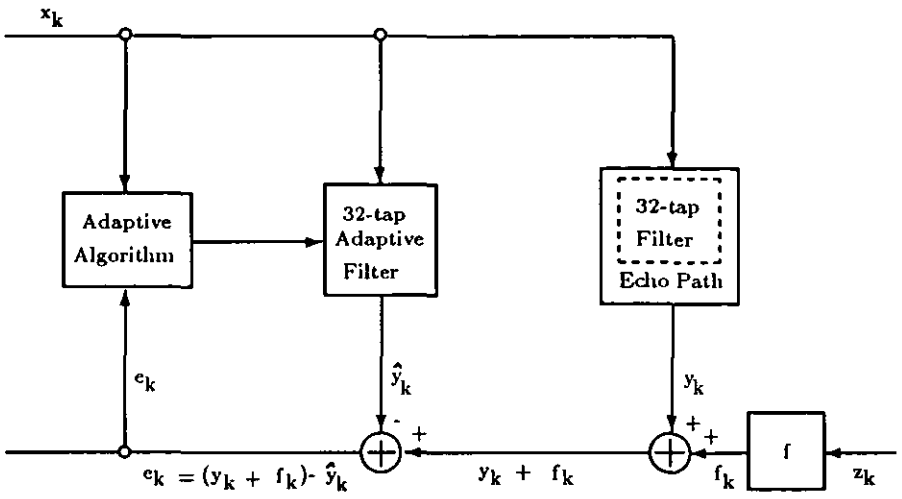


Figure 4.4: Simplified block diagram of the simulation setup used for most of the computer simulations. x_k is the input signal sequence at the near-end, z_k is the far-end signal sequence and f_k is the attenuated far-end signal sequence with attenuation f. y_k is the output of the actual echo path, $\hat{y}_k$ is the output of the estimated echo path and e_k is the error signal.

4.4 Simulation setup

The simplified block diagram of simulation setup is shown in Figure 4.4. The simulator is a direct system modelling type where the echo canceller is trying to model the echo path. The data echo canceller is modelled as a 32 tap linear time varying FIR adaptive filter whose co-efficients are updated regularly by the adaptation algorithm. The output of the adaptive filter which is an estimate of the echo path is then subtracted from the signal which is sum of echo signal and far-end signal. The difference of these two is called the error signal.

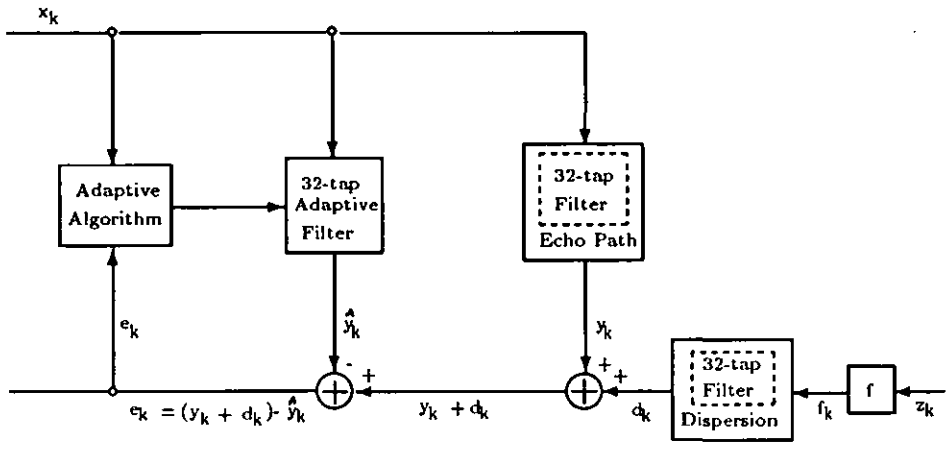


Figure 4.5: Simplified block diagram of the simulation setup used for computer simulations when dispersion was also added to the far-end signal sequence. x_k is the input signal sequence at the near-end, z_k is the far-end signal sequence, f_k is the attenuated far-end signal sequence with attenuation f and d_k is the dispersed as well as attenuated far-end signal sequence. y_k is the output of the actual echo path, $\hat{y}_k$ is the output of the estimated echo path and e_k is the error signal.

> error signal = (far-end signal + echo signal) - (estimate of echo signal)

The error signal is then used by the adaptation algorithm to update the coefficients of the adaptive filter in order to get the best estimate of echo path, so that the error signal could be minimised.

Dispersion can be added to the far-end signal in addition to the attenuation. Some simulations are also performed with added dispersion. The simulation setup used for these kind of computer simulations is given in Figure 4.5.

4.5 Performance measure

We have used the average of normalised tap-error vector to demonstrate the behaviour of algorithm, below the far-end signal power level. The performance measure $p_{(k)}$ used in the present data echo canceller algorithm is as given below.

$$p_{(k)} = E\left[\frac{\sum(\mathbf{h} - \hat{\mathbf{h}}_\mathbf{k})^2}{\sum(\mathbf{h})^2}\right] \tag{4.3}$$

where,

$\mathbf{h}$ =cooefficient vector of the echo path filter,

$\hat{\mathbf{h}}$ =co-efficient vector of the filter estimating the echo path, and

E is the statistical expectation operator for ensemble averaging.

This kind of averaging helps in smoothing out any noise present in the convergence curves. The ideal situation is that the averaging should be over an infinite ensemble, which is impractical. A more practical way of approximating statistical expectation is to average over a suitable finite number of examples. The number chosen here is 20, which is large enough to provide reasonably smooth results. Equation 4.3 with averaging 20 times becomes

$$p_{(k)} = \frac{1}{20}\sum_{i=1}^{20}\left(\frac{\sum_{n=0}^{N-1}\left(h(n) - \hat{h}(n)\right)^2}{\sum_{n=0}^{N-1}\left(h(n)\right)^2}\right) \tag{4.4}$$

Where N is the number of taps and is equal to 32 in our case. Four different convergence levels were chosen as performance goals for their respective situations, which are, -35, -40, -45 and -50 dB.

4.6 Simulations

Computer simulations were carried out on SUN SPARC stations using the FORTRAN computer language. Computer codes are given in the Appendix. All the results presented in this report are obtained using "rand(0)", a random number generator from the Sun FORTRAN library.

The new adaptation algorithm as discussed earlier (Equation 3.11) was used. The cost function power τ was varied starting always from τ=2.0, then incrementing in steps of 0.1 for each simulation. For each set of simulations, far-end signal power level along with the convergence level were fixed with reference to the near-end signal level. An optimum step size value μ was used for each individual simulation depending upon the value of cost function power τ, far-end signal power level and convergence level for that particular simulation. A slightly larger μ will result in a higher level of convergence and a slightly lower than the optimum μ will result in a lower than the required convergence level, if all the other parameters are kept constant. The value of μ which is called optimum here will give the required convergence level. e.g. for $\tau = 2.2$, far-end signal level at -15 dB, third echo path model, and binary signals, the value of μ is 0.004 to achieve a convergence level of -35 dB. If all the other parameters are kept constant, a value of μ higher than 0.004 will result in a convergence level higher than -35 dB. On the other hand if the value of μ is lowered than the 0.004 keeping all the other parameters constant, the resultant convergence level will fall down accordingly. In this case the value of μ which gives the required convergence level of -35 dB (20 dB lower than the far-end signal level of -15 dB), or $\mu = 0.004$ is designated as the optimum value of μ

The optimum value of step size for each simulation was determined prac-

tically by running the program several times. Each time raising or lowering the value of μ to achieve the required convergence level. Once the required convergence level is achieved, the value of μ used to get that was labelled as optimum value of μ and was then used in ensemble averaging. Before recording the convergence time, each simulation was ensemble averaged over twenty times to reduce noise in the convergence curves. A typical convergence curve is shown in Figure 4.6. In Figure 4.6, the sum of squares of tap errors normalised by the sum of squares of target tap values is plotted in dBs against the number of iterations. The same curve is shown with averaging by twenty times in Figure 4.7. The same curve was ensemble averaged by 50 and 100 times. Ideally the averaging should be over infinite number of examples or a very very large number, which is impractical for these simulations. The curve with 100 times averaging was clearest of all but it also takes more time as well. There is relatively more noise in the plot shown in Figure 4.7 but 20 times averaging is much quicker to achieve than that of 100 times. A compromise has to be made between the time taken and the noise left behind, depending upon the requirement of the job. Finally it was decided that 20 times averaging is suitable for our purpose. Practically the 20 times and 100 times averaging makes no difference in our results.

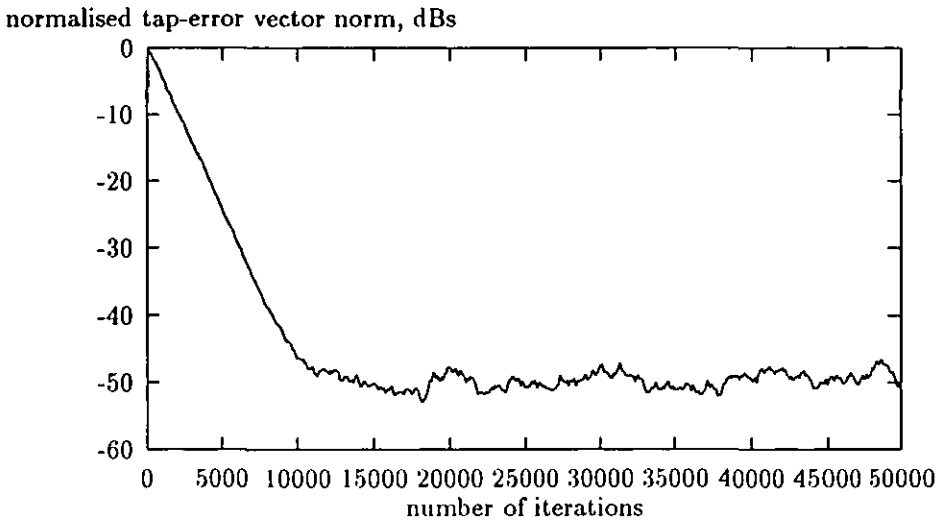


Figure 4.6: A typical convergence curve.

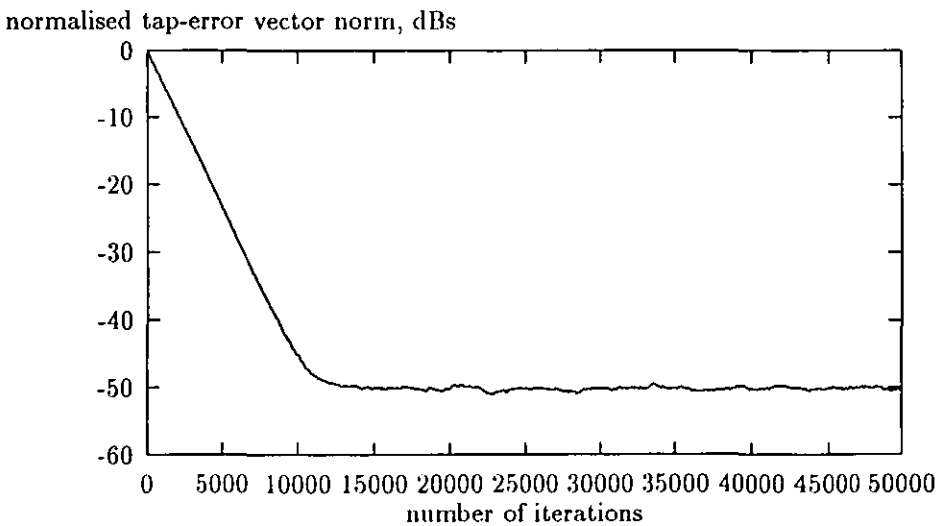


Figure 4.7: Convergence curve of Figure 4.6 with 20 times averaging.

The aim of the experiment was to find out the increase in the rate of initial convergence by using the non quadratic cost function power τ. Also to find out the region of low sensitivity to the variations in τ if there is any.

The simulations were carried out in such a way that convergence level is always a few decades below the level of the far-end signal. This is necessary in order to extract the useful information (i.e. far-end signal), from the incoming signal that has been corrupted, by an echo signal of power level higher than the far-end signal one. Typically, a received signal-to-uncancelled echo ratio of better than 20 dB is required [2]. For the simulations presented in this thesis, convergence was achieved at 20 dB or two decades below the far-end signal power level. A range of far-end signal power attenuation levels was used as described in the next chapter. Convergence time, in terms of number of iterations, was observed and recorded for each individual simulation and then plotted against cost function power τ for each set of simulations.

Chapter 5

Simulations with binary data

5.1 Introduction

This chapter describes various simulation methods and results. Computer simulations were performed with pseudo random non-Gaussian binary data with the two levels of +1 and -1, as described in Section 4.6. Four cases, with various far-end signal levels added to the returned signal, were studied in order to illustrate a range of possible operating conditions. The far-end signal levels used were -15, -20, -25 and -30 dB. In all these cases, convergence was achieved 20 dB below the far-end signal level. Three different echo path models as described in Section 4.3, were used. Calculated and measured step sizes of the convergence process are tabulated. Later simulations are also performed with the higher value of τ switched back to 2.0 and vice versa during the course of convergence. Switching behaviour of gradient is investigated. Simulations are also performed with dispersion added to the far-end signal in addition to the attenuation.

5.2 Simulations with binary data

A pseudo random sequence of values +1 and -1 is used as non-Gaussian binary signal at the input (transmitter) of the near-end of our echo canceller model. Part of this input or transmitted signal is reflected back from a point of impedance mismatch (echo path) as shown in Figure 4.4. This reflected or returned signal which is now distorted by the echo path transfer function is received at the receiver of the near-end of our echo canceller model, along with the attenuated far-end signal. The far-end signal sequence is modelled by a similar but independent pseudo-random sequence of values +1 and -1 with a certain attenuation factor. Transfer function of the echo path is modelled as a digital FIR filter of length 32 using three different echo path models, as described in Section 4.3. The transfer function of the echo path is estimated by a digital adaptive filter of length 32. The output of the adaptive filter is then subtracted from the returned signal to obtain an error signal. The error signal is then used by the newly developed adaptation algorithm (3.11) to calculate and update the co-efficients of the adaptive filter, which in turn calculates a new and better estimate of the echo path in the next cycle. In this way the process leads towards convergence. The new algorithm (3.11) was developed with the intention of speeding up the process of convergence in terms of the number of iterations. Computer simulations are performed to look at this decrease in convergence time.

Coefficients of the adaptive filter were reset at the start of each convergence process, or simulation. At each time step or the sampling instant, the difference of the coefficients of the echo path filter and the adaptive filter was taken, squared and normalised by the square of the echo path filter coefficients. These

near-end signal level dBs	far-end signal level dBs	conver -gence level dBs	(far-end signal level) -(convergence level) dBs	max. τ for the fastest convergence		
				1st model	2nd model	3rd model
0	-15	-35	20	3.1	3.6	2.6
0	-20	-40	20	3.0	3.1	2.5
0	-25	-45	20	2.8	3.0	2.5
0	-30	-50	20	2.7	2.8	2.4

Table 5.1: Various combinations of power levels and the maximum values of cost function power τ attained for the fastest convergence at or below a particular convergence level. Data streams were binary.

values, obtained with the appropriate convergence factor or step size μ and cost function power τ, were then plotted for a sufficient number of iterations. The curves obtained in this way were similar to those of Figures 4.6 and 4.7. Desired convergence levels were achieved after a number of tries by modifying step size μ while keeping the rest of the parameters constant. Convergence times were observed and recorded from these curves. An appreciable decrease in convergence time with increasing cost function power τ can be observed from the above mentioned record of convergence times.

Various levels of far-end signals and convergence were used in simulations. Four levels of far-end signal (-15, -20, -25 and -30 dB) were selected for presentation in this thesis. Convergence was always achieved 20 dB below the far-end signal levels. Respective power levels in decibels along with some related infor-

mation are summarised in Table 5.1.

The decrease in convergence time with the increase in cost function power τ is presented in graphical form in Figures 5.1–5.4. In its most basic form, convergence time in terms of the number of iterations is plotted versus cost function power τ. Change in convergence time with respect to the change in cost function power τ can be readily observed from such a curve. Four figures are presented here, one for each far-end signal level. Each of these figures contain simulation results of all three echo path models used.

Figures 5.1–5.4 show the results of simulations for the far-end signal levels of -15, -20, -25 and -30 dB respectively. It is obvious from these figures that the shape of all these curves is essentially similar. The convergence time (in terms of the number of iterations) reduces as we go on increasing cost function power τ. In the initial portions of the curves, convergence time is more sensitive to the changes in cost function power τ. As we go on increasing τ, this sensitivity gradually decreases. At a certain value of τ, the algorithm stops converging exactly at a particular target convergence level, however, it may converge below that level. We may continue increasing τ to achieve convergence below the target level and record the convergence time of the crossing point of the curve and the target convergence level. Soon after, the convergence process starts slowing down with further increments in the value of τ. The end result is that we obtain a bathtub like feature by plotting convergence time versus τ. A certain value of τ or a range of the values of τ achieve the minimum convergence time for a particular combination of echo path model and far-end signal. These maximum values of τ for the minimum possible convergence time are summarised in Table 5.1.

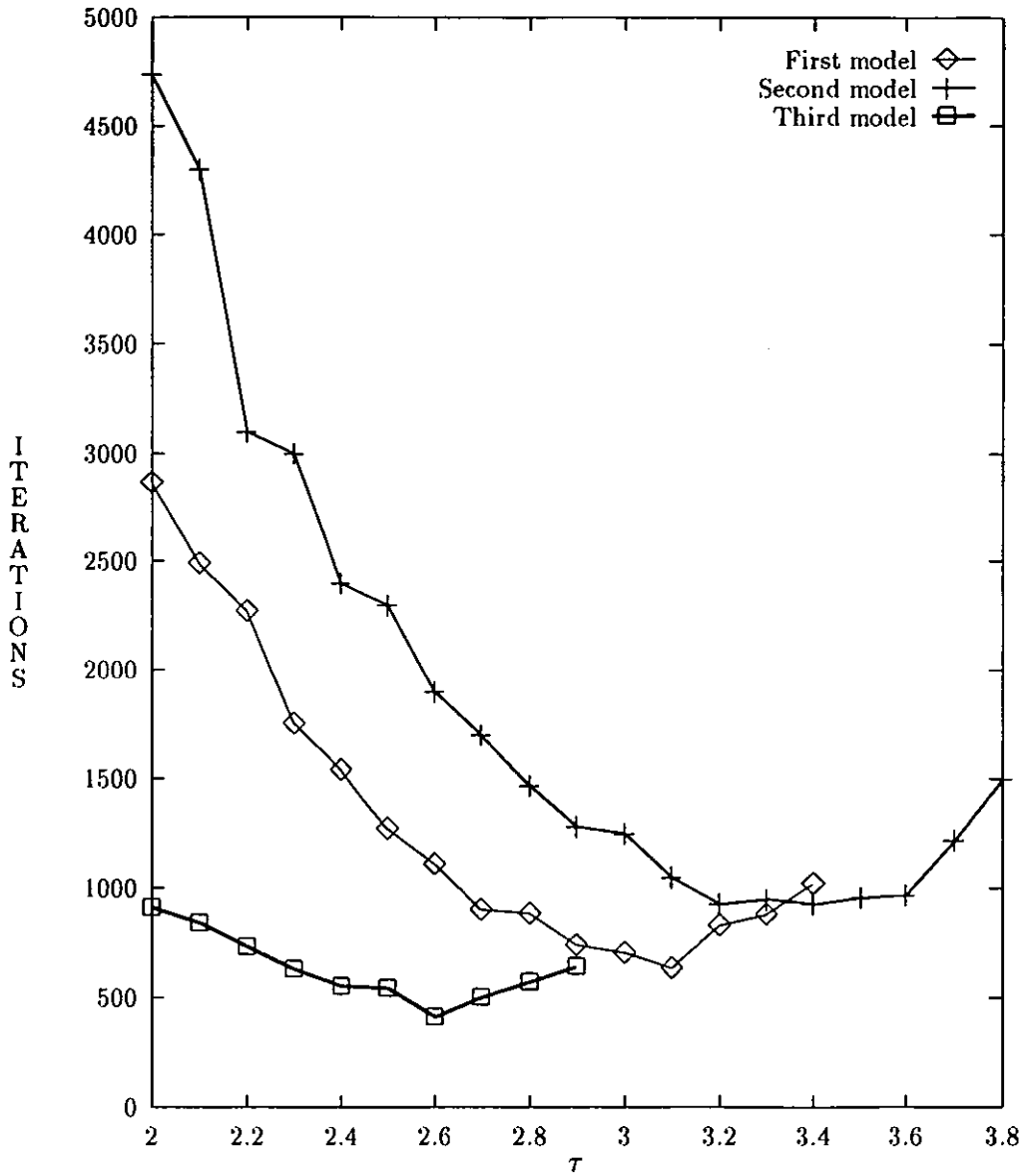


Figure 5.1: Convergence time versus cost function power τ is plotted for all three echo path models. Far-end signal is at -15 dB and convergence was achieved at -35 dB as compared with that of the near-end signal.

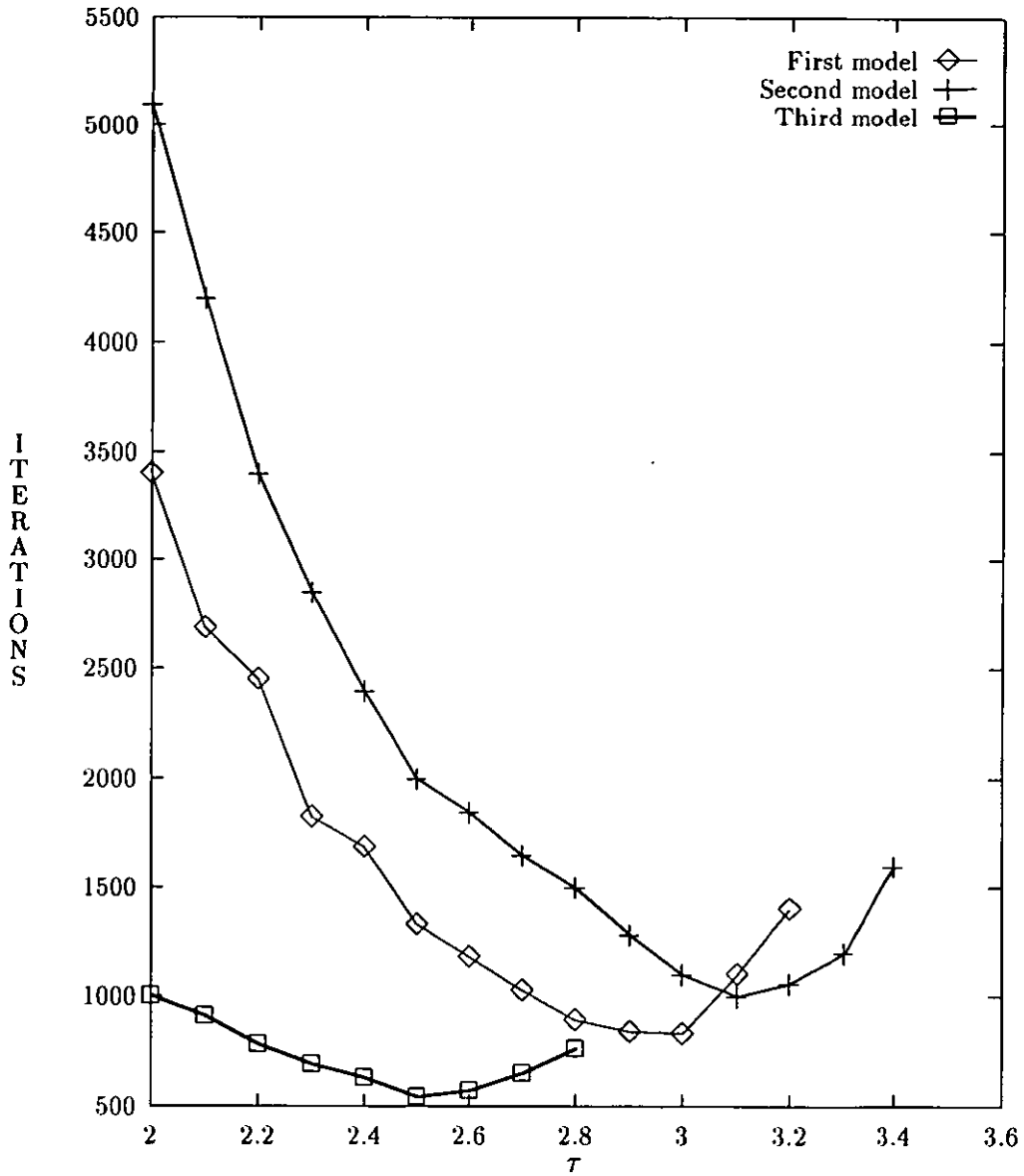


Figure 5.2: Convergence time versus cost function power τ is plotted for all three echo path models. Far-end signal is at -20 dB and convergence was achieved at -40 dB as compared with that of the near-end signal.

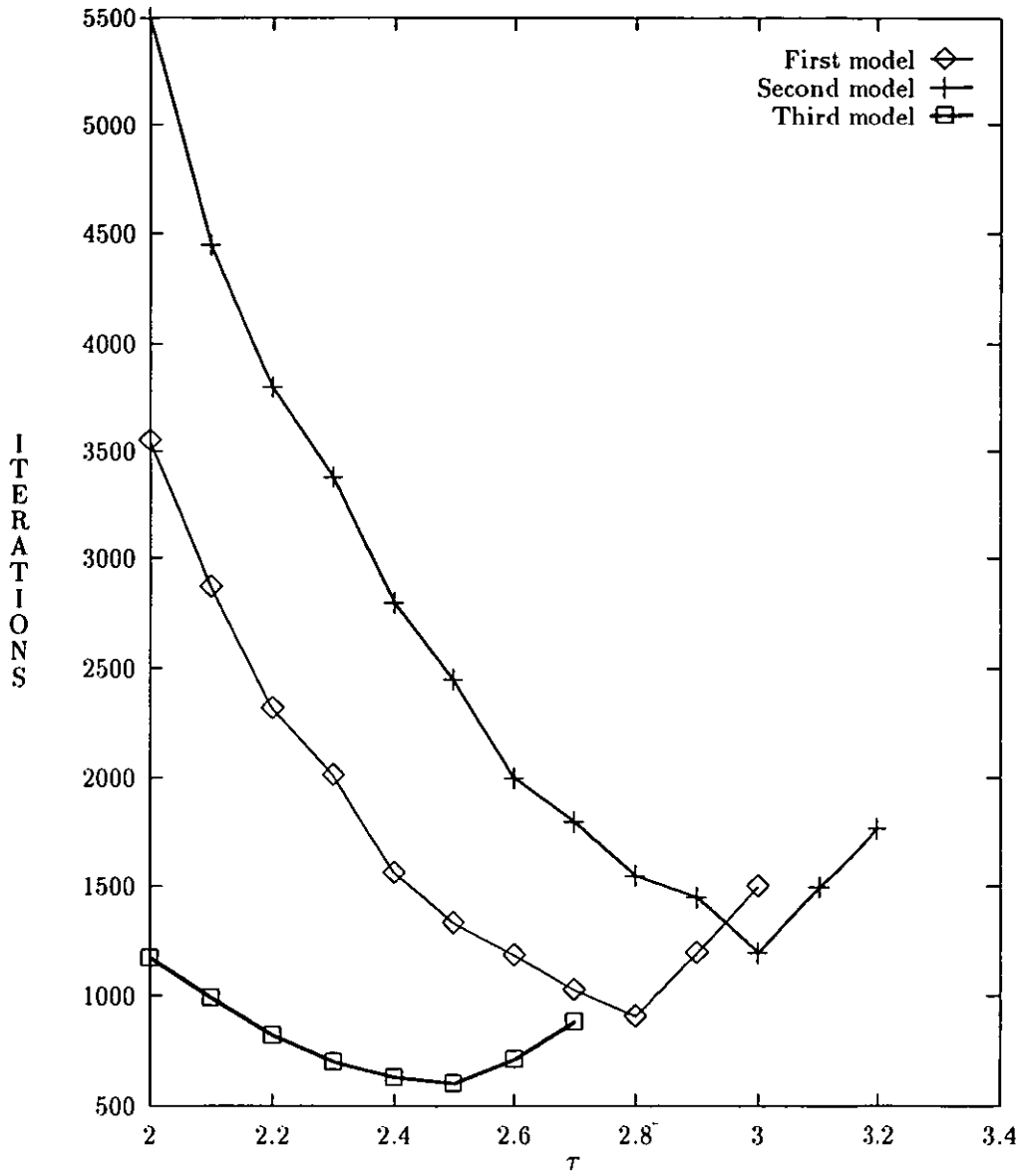


Figure 5.3: Convergence time versus cost function power τ is plotted for all three echo path models. Far-end signal is at -25 dB and convergence was achieved at -45 dB as compared with that of the near-end signal.

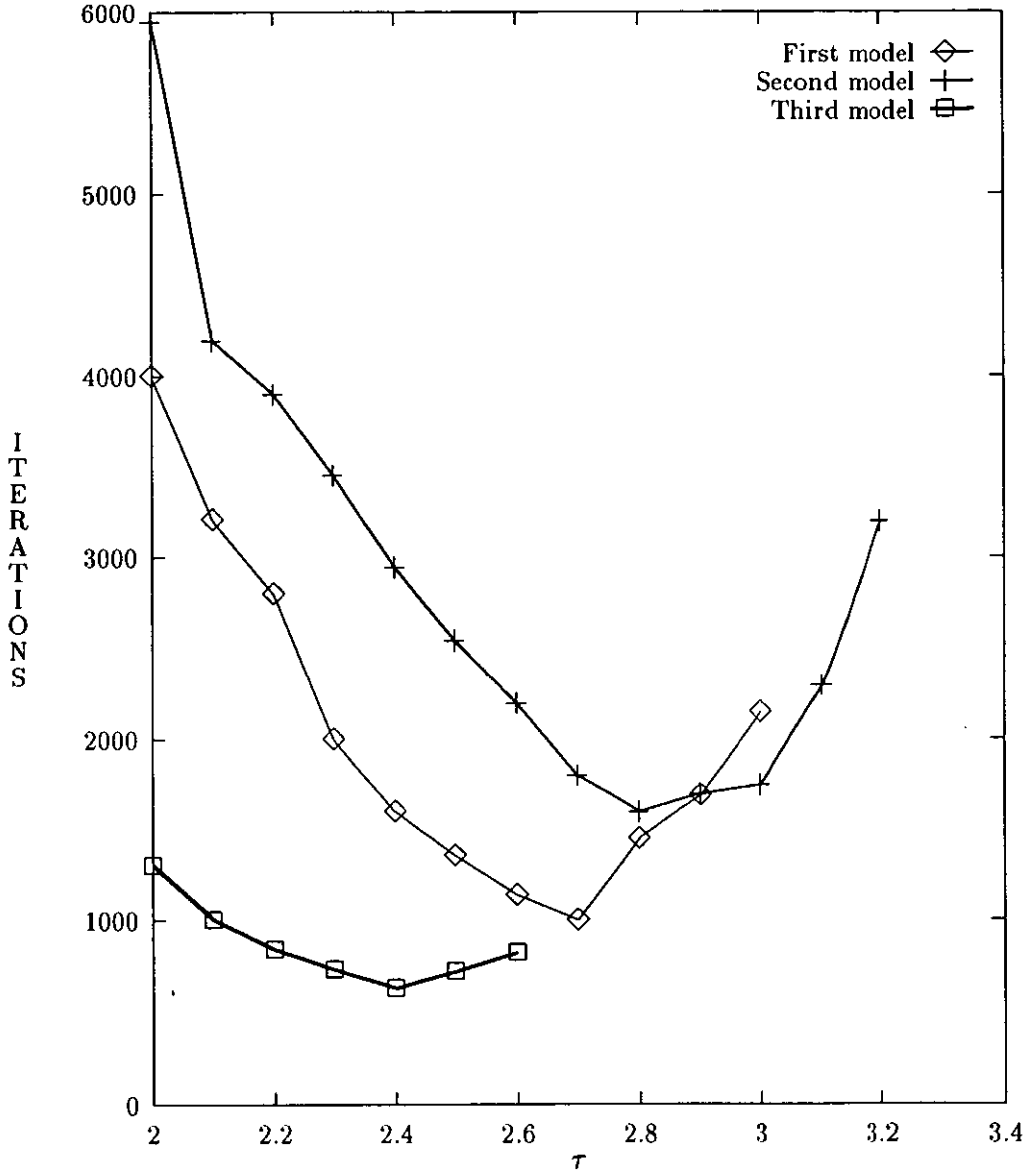


Figure 5.4: Convergence time versus cost function power τ is plotted for all three echo path models. Far-end signal is at -30 dB and convergence was achieved at -50 dB as compared with that of the near-end signal.

Figures 5.1–5.4 also incorporate the effects of the echo path models for the respective far-end signal levels. Characteristics of the echo path filter play an important role. Area under the transfer function curve of the echo path relates to the signal power by the Wiener-Khinchine theorem at a shift of zero [92]. According to Wiener-Khinchine relations, the zero-frequency value of the spectral density of a stationary random process equals the total area under the graph of the autocorrelation function, i.e.

$$S_{xx}(0) = \int_{-\infty}^{\infty} \phi_{xx}(t)dt \tag{5.1}$$

The mean square value of the output of a linear filter is given by,

$$E[Y_k^2] = \int_{-\infty}^{\infty} |H(z)|^2 S_{xx}(z)dz \tag{5.2}$$

Where $|H(z)|$ is the magnitude of the transfer function. As we observe from the three different models used here that the area under the curve of the transfer function of the echo path model (see Figures 4.1–4.3) affect these curves and determine their relative length, percentage reduction in convergence time, etc. e.g. in Figure 5.1, convergence time reduces by 79% for the second echo path model, whereas it undergoes a reduction of 55% for the third echo path model. Reduction in convergence time is computed by comparing the minimum convergence time with that of the mean square convergence time. Mean square convergence time for the third echo path model is already better than the first and the second models. Therefore, comparatively less reduction in convergence time may be achieved for an echo path model, which already has comparatively better convergence time for the mean square function, and vice versa.

The relative improvement in convergence time is more for models with relatively small transfer function areas (Equations (5.1) and (5.2)) (e.g. the second

echo path model). The models with larger transfer function areas (e.g. the third echo path model) can only achieve a relatively small decrease in convergence time with increase in τ. A smaller transfer function area means a higher value of the maximum τ for the minimum possible convergence time, more tolerance and vice versa.

The same curves provide different information if presented in a different way. Another way of looking at the above mentioned results is to examine all four cases of different far-end signals for a particular echo path model in a single diagram. Figures 5.5–5.7 show all four cases of the same model in one figure respectively. By inspecting these figures we observe that the curves mostly overlap each other. Variations in far-end signal level affect the length etc. of the curves. Relative reductions in convergence time remain nearly the same for all four far-end signal power attenuation levels in a particular echo path model. This observation leads to the conclusion that the reduction in convergence time is independent of the power attenuation levels of the far-end signal. On the other hand, it is evident from figures 5.1–5.4, that the reduction in convergence time does depend upon the type of the echo path model. Figures 5.5–5.7 also indicate that the maximum value of τ, for the lowest convergence rate, is higher, for the less attenuated signals, and vice versa. It is evident from Figures 5.1–5.4 that the above mentioned parameters also depend upon the type of echo path model.

While the curves start at different points they rapidly converge. All three echo path models follow the same pattern with regard to the starting points in these curves. The starting points i.e. the convergence time at $\tau = 2$ is higher for more attenuated far-end signal and vice versa.

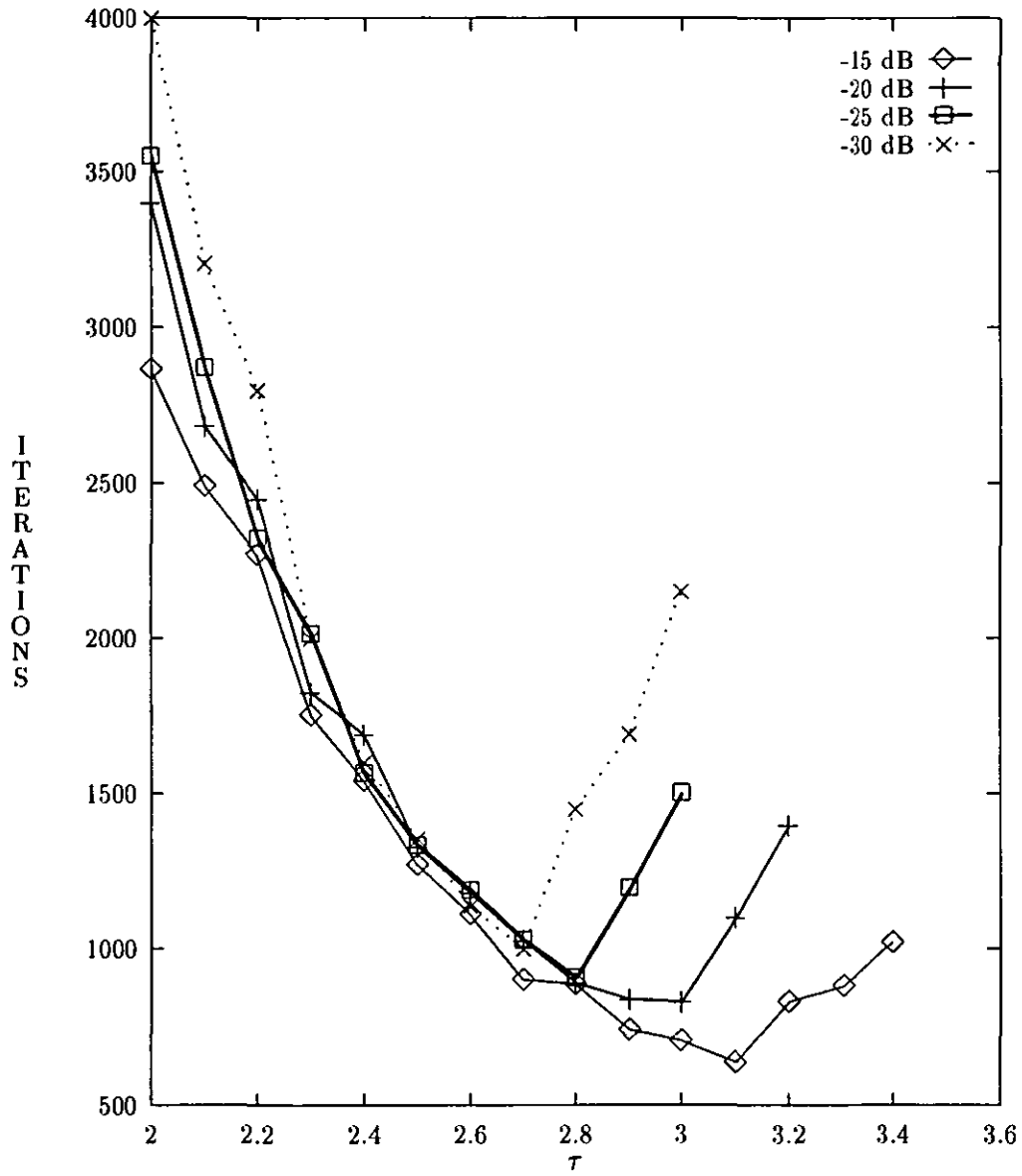


Figure 5.5: Convergence time versus cost function power τ is plotted for the first echo path model and all four far-end signal levels of -15, -20, -25 and -30 dB.

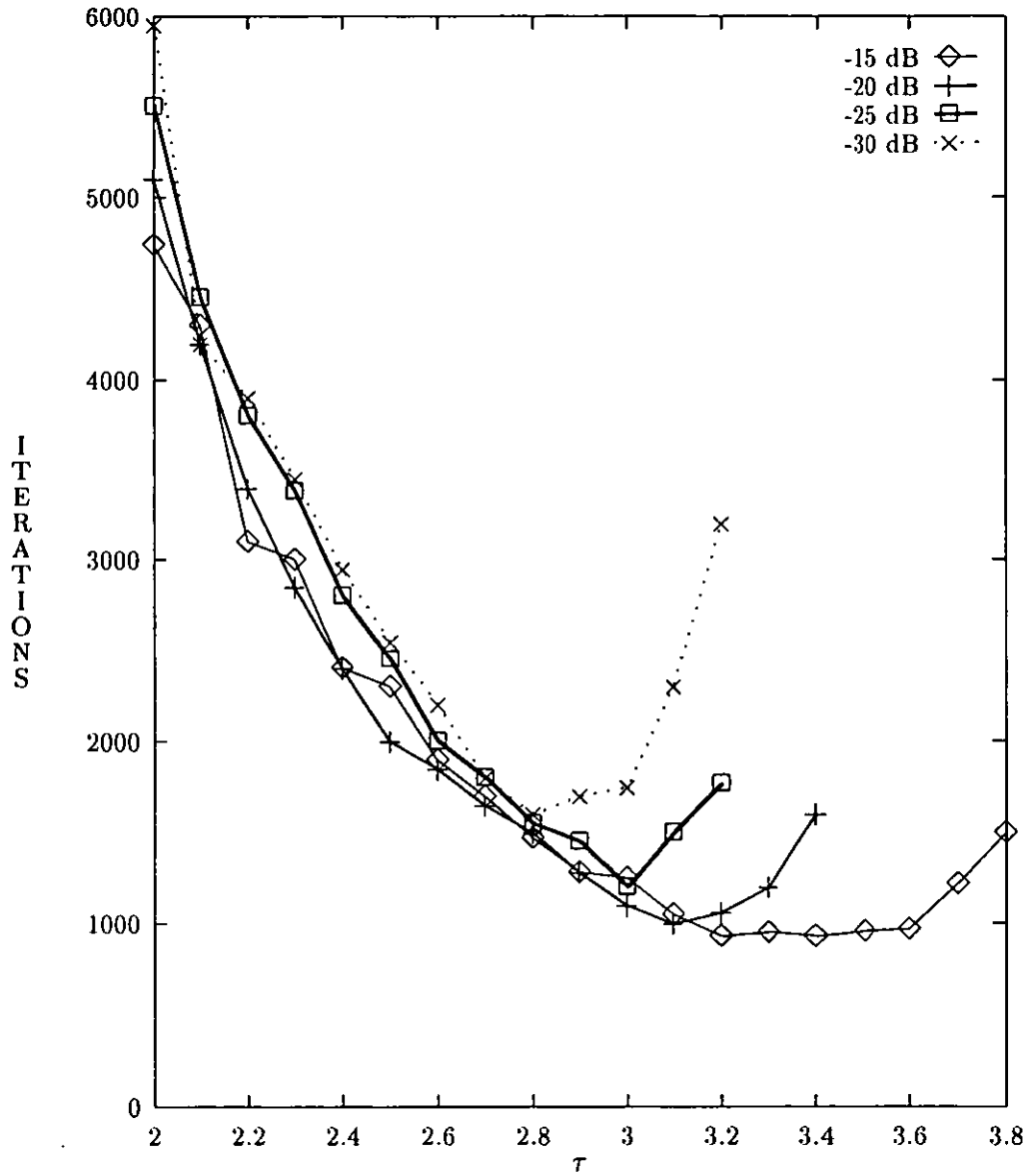


Figure 5.6: Convergence time versus cost function power τ is plotted for the second echo path model and all four far-end signal levels of -15, -20, -25 and -30 dB.

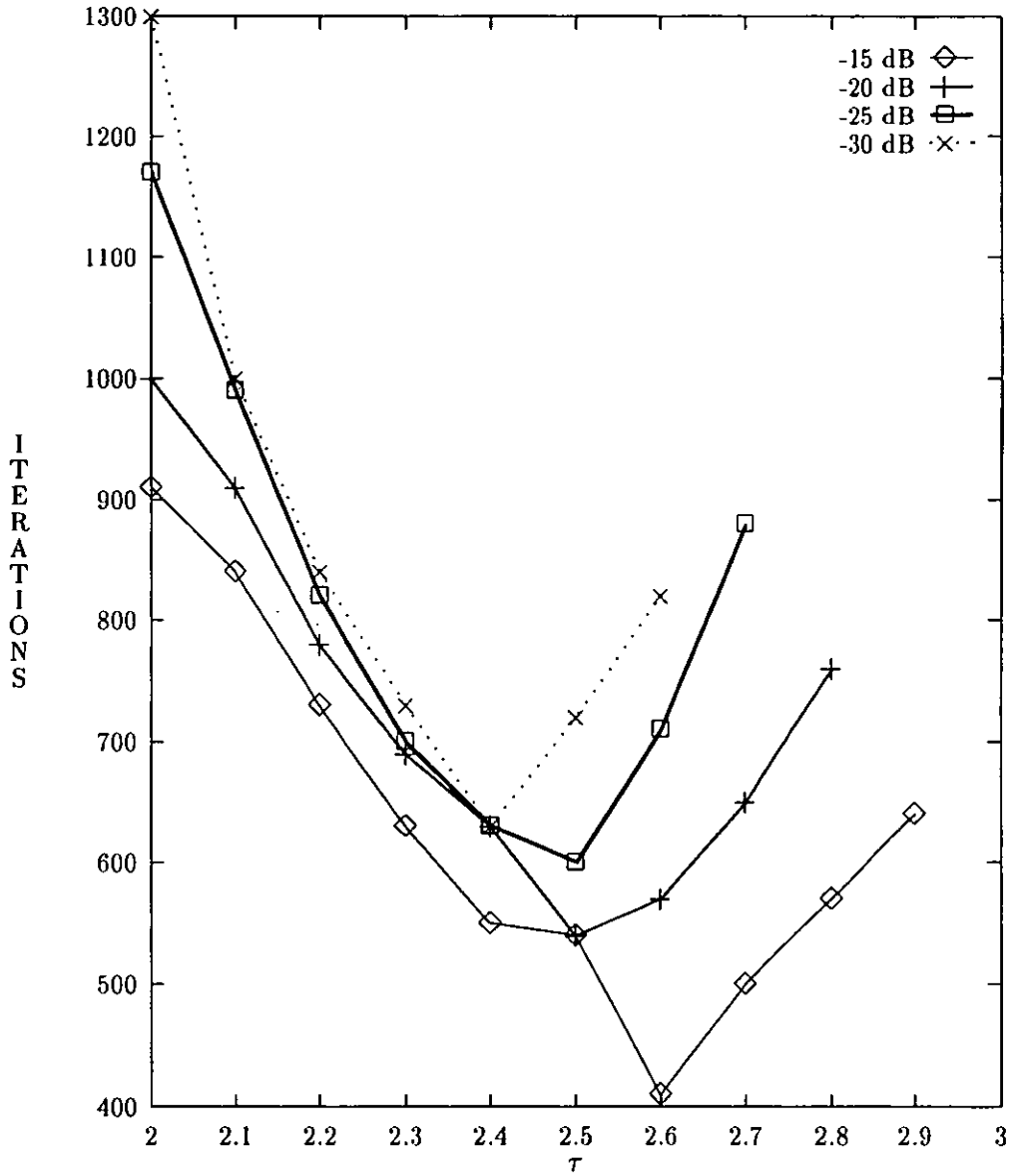


Figure 5.7: Convergence time versus cost function power τ is plotted for the third echo path model and all four far-end signal levels of -15, -20, -25 and -30 dB.

Learning curves for the third model in Figure 5.1

Figures 5.8–5.17 are the learning curves for the third echo path model and far-end signal level of -15 dB. Each of these curves constitute a single point in the lowest curve in Figure 5.1. Normalised tap-error vector is plotted versus number of iterations. Each curve is averaged 20 times. Convergence level is set at -35 dB. We continue achieving this level till $\tau = 2.6$. For $\tau = 2.7$ we cannot achieve this convergence level as the algorithm becomes unstable. We can, however, achieve a convergence level lower than -35 dB. μ is decreased as well to keep the algorithm stable. We can now measure the convergence time (in terms of number of iterations), when learning curve crosses the -35 dB level. Convergence is comparatively slower when achieved at a lower level. This effect results in the turning point in the lowest curve in Figure 5.1. From this point onwards, higher values of τ results in further slowing down of the convergence process as we have to decrease μ every time to keep the algorithm stable.

Simulations with identical (unit) power of the echo path models

Simulations were also performed with identical power of the impulse responses of the three echo path models. To normalise the power of each of the impulse response to 1, each weight of a particular impulse response was divided by the square root of the sum of the squares of all the weights in a model. Computer simulations were performed in a similar way as for the results shown in Figure 5.1. The results of these simulations are presented in Figure 5.18. A comparison of Figures 5.1 and 5.18 reveals that there isn't much difference especially in the general shape of the curves. The only difference is that comparatively higher values of τ are achieved for all the three models. These simulation re-

sults show that different echo paths behave differently even when their power is normalised.

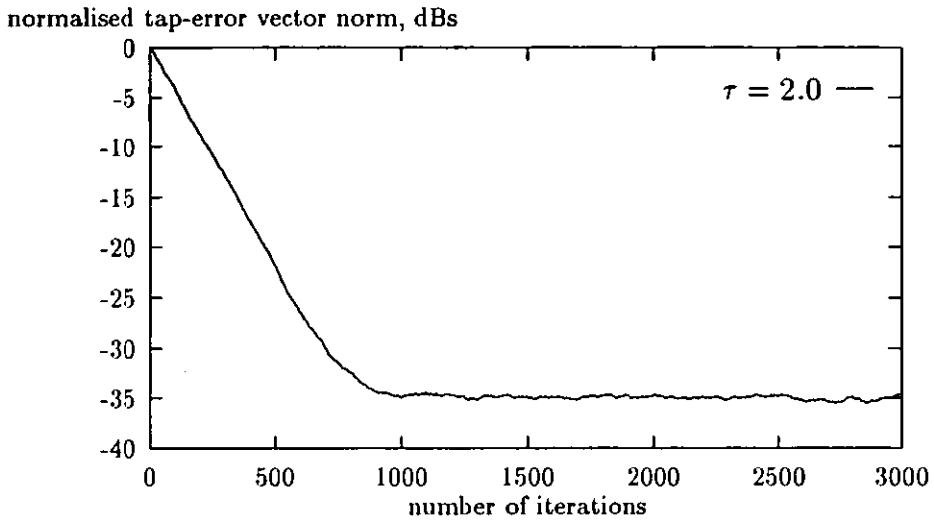


Figure 5.8: Learning curve for $\tau = 2.0$.

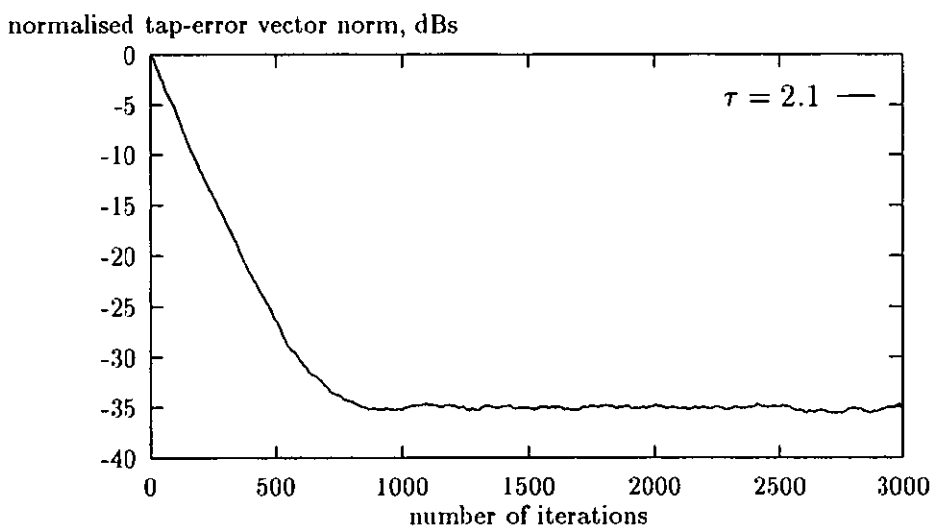


Figure 5.9: Learning curve for $\tau = 2.1$.

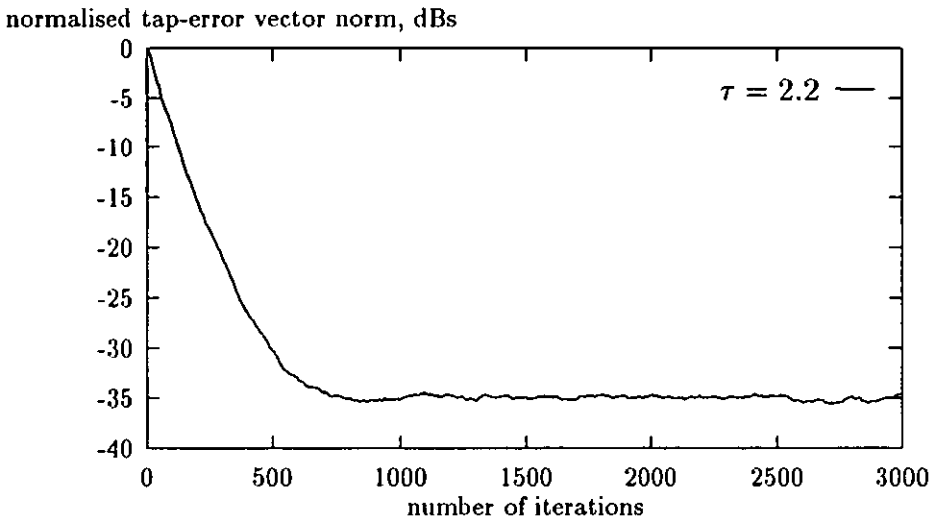


Figure 5.10: Learning curve for $\tau = 2.2$.

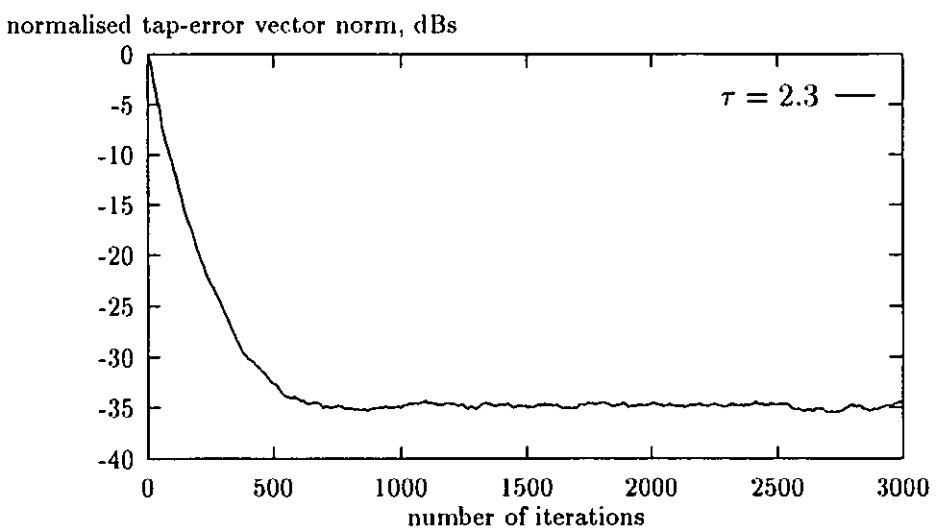


Figure 5.11: Learning curve for $\tau = 2.3$.

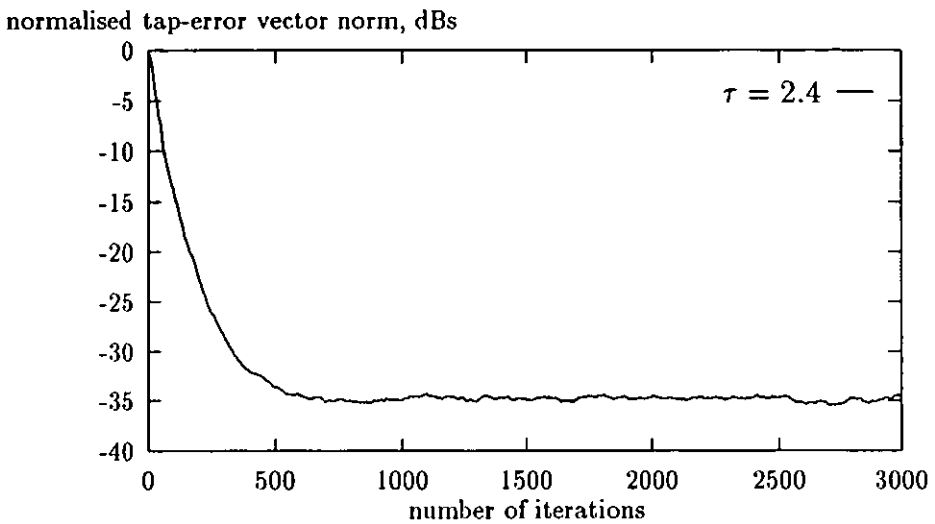


Figure 5.12: Learning curve for $\tau = 2.4$.

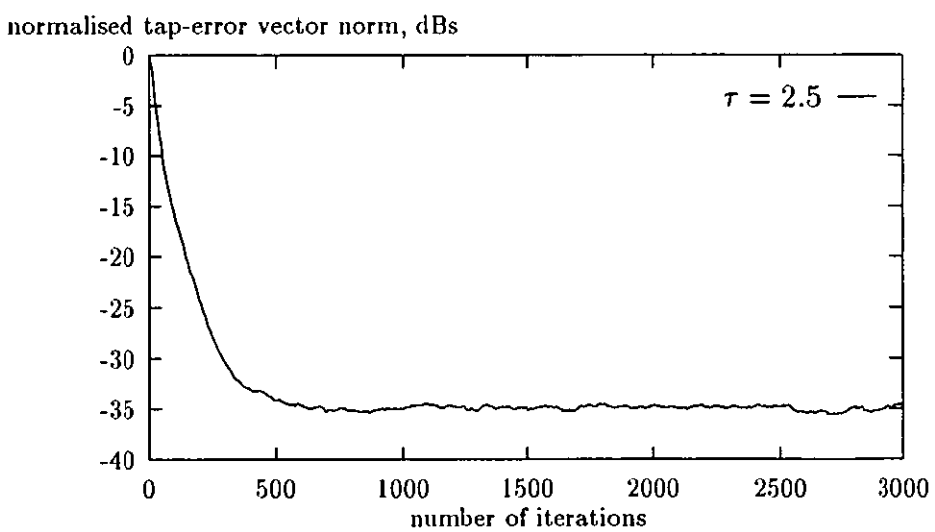


Figure 5.13: Learning curve for $\tau = 2.5$.

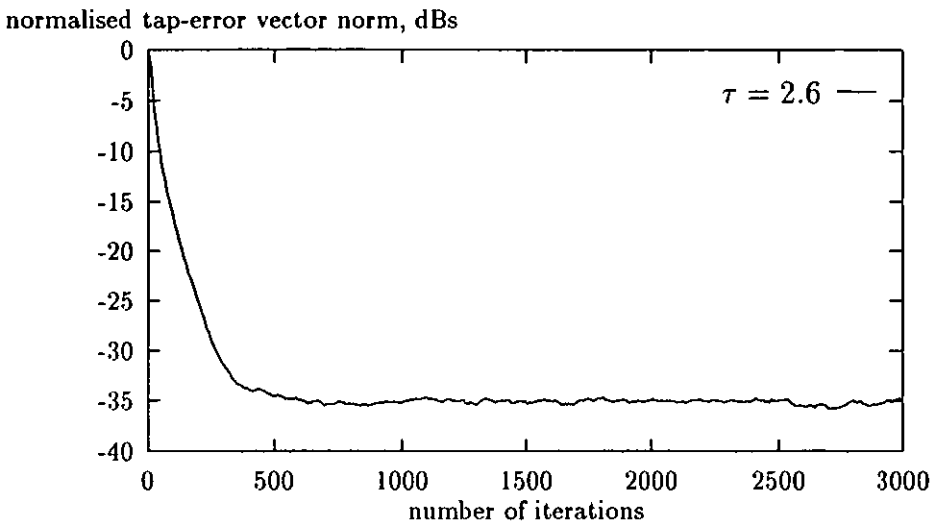


Figure 5.14: Learning curve for $\tau = 2.6$.

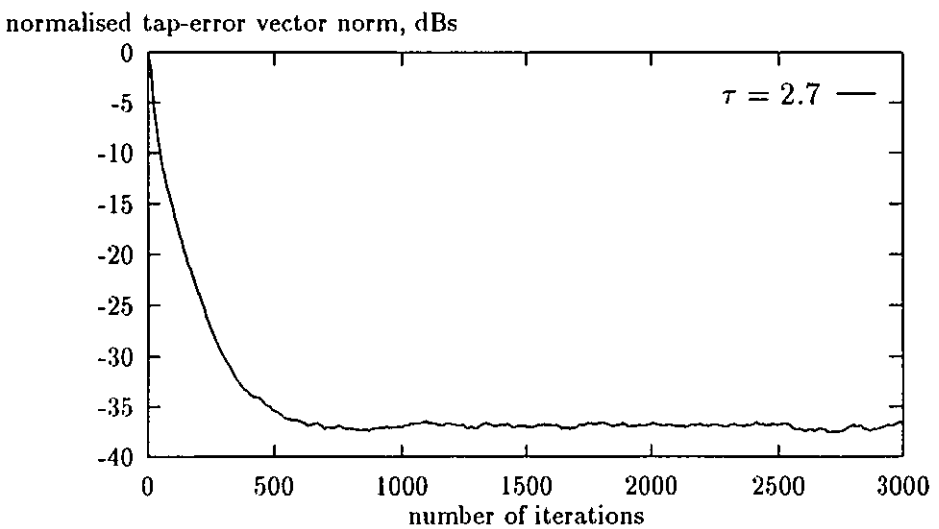


Figure 5.15: Learning curve for $\tau = 2.7$.

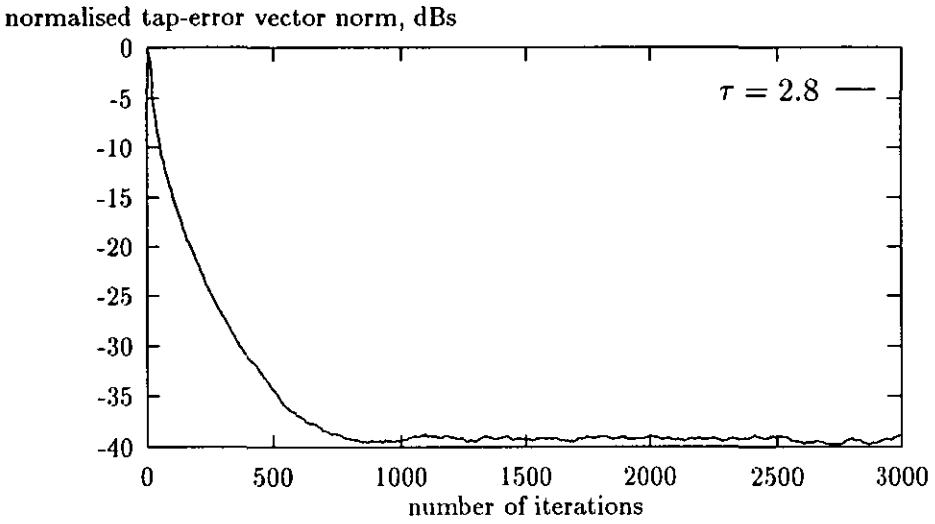


Figure 5.16: Learning curve for $\tau = 2.8$.

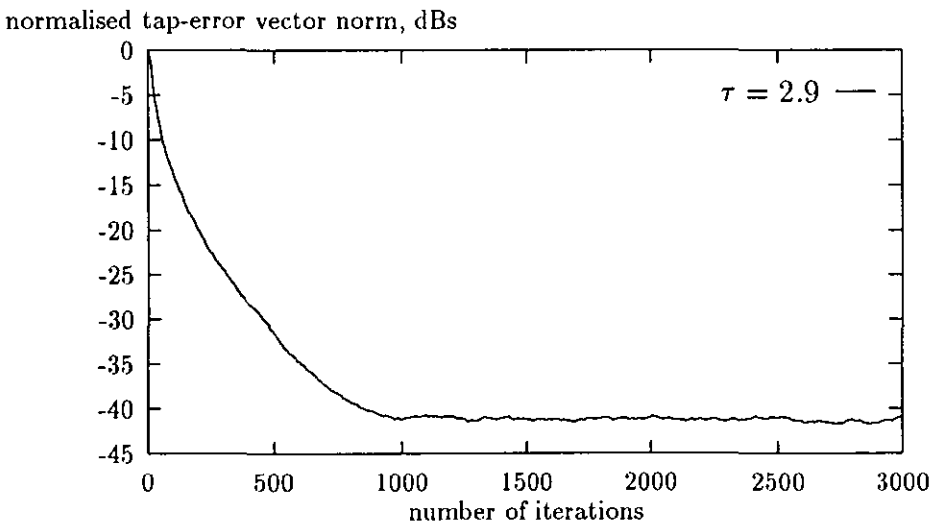


Figure 5.17: Learning curve for $\tau = 2.9$.

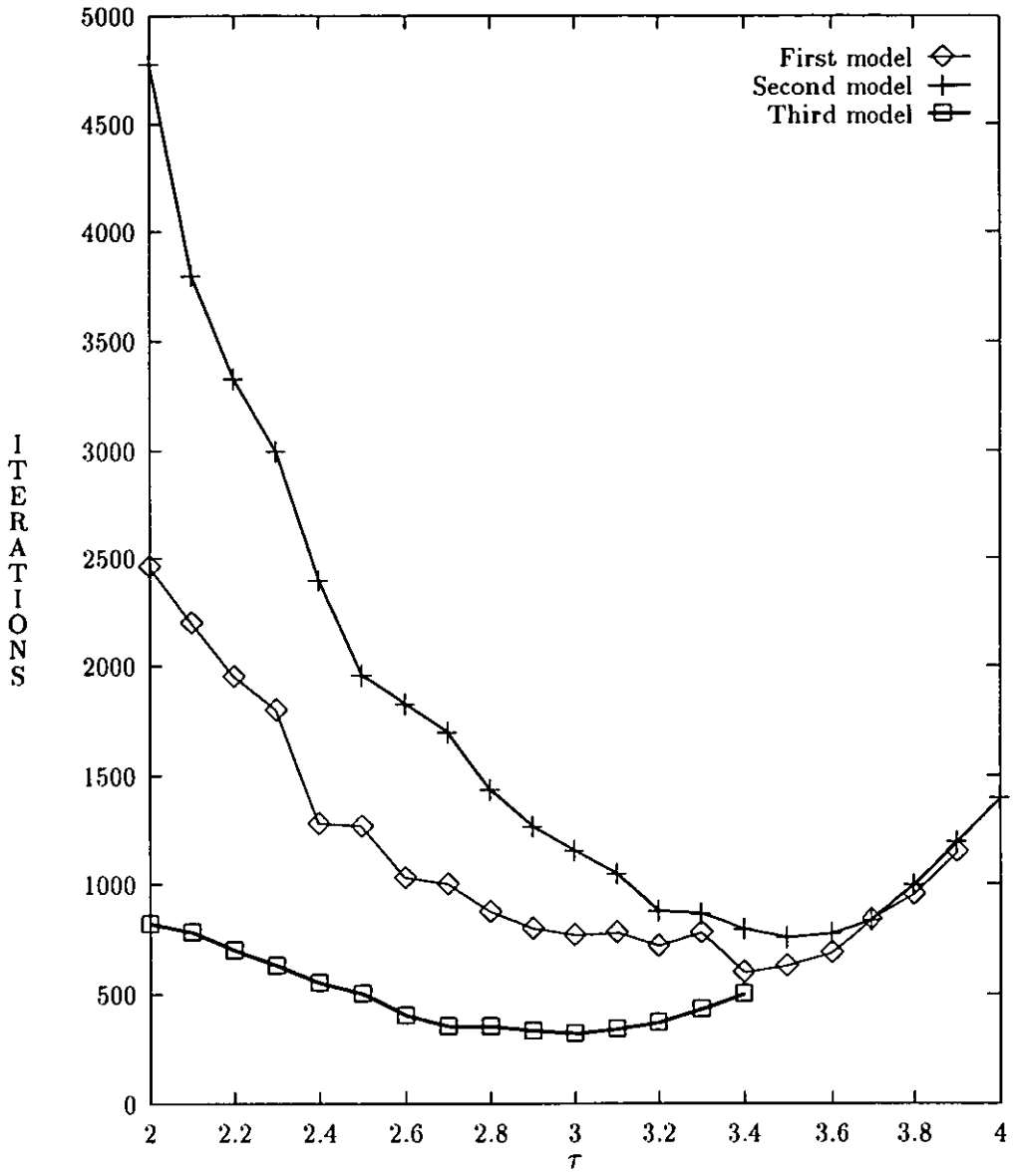


Figure 5.18: Convergence time versus cost function power τ is plotted for all three echo path models. Far-end signal is at -15 dB and convergence was achieved at -35 dB as compared with that of the near-end signal. All the echo path models used are normalised to 1.

5.3 Convergence level and some aspects of calculated and measured step size of the convergence process.

We can calculate the upper bound on step size μ using Equation (3.29) for each value of cost function power τ and various far-end signal levels. These calculated values of the upper bounds on μ can be compared with those obtained by experiment. Computer simulations were performed to obtain the experimental maximum achievable values of μ. The simulation results show that in all cases, experimentally measured values of μ falls well below the upper bound set by Equation (3.29).

Computer simulations for this purpose were performed using binary data sequences, for the first echo path model, and without switching back to lower gradient for $|e_k| \geq 1$ during convergence. All the four cases of far-end signal level i.e. -15, -20, -25 and -30 dB were studied. For each simulation, step size μ was increased, keeping τ constant, and without putting a restriction on convergence level. Results were obtained after averaging three times for relatively better readability of convergence level. For the same τ, μ was further increased in the next run. This process continued until the algorithm became unstable. In this way it is possible to obtain the maximum achievable value of μ, μ_{max} *measured* with a certain τ and far-end signal level.

Calculated upper bounds on μ, μ_{max} *calculated* from Equation (3.29) for various values of τ are recorded in tabular form along with the measured values of μ_{max} and corresponding convergence levels. Tables 5.2–5.5 give a comparison of the calculated and measured values of μ_{max} with the relative information of

convergence level and τ for the far-end signal levels of -15, -20, -25 and -30 dB respectively.

A few deductions can be made by a close look at Tables 5.2–5.5. The first one is that μ_{max} *measured* is well below the upper bounds μ_{max} *calculated* set by Equation (3.29) for all the tabulated cases. This fact is obvious from the comparison of the second and third columns of the said tables.

The last columns of the tables under consideration, provide the corresponding convergence levels for each μ_{max} *measured*. As we know that convergence level increases with the increase in μ and vice versa, while keeping the rest of the parameters constant. This means that these convergence levels (given in the fourth columns) are, the maximum achievable convergence levels for the particular values of τ as they correspond to the maximum measured value of μ. We now make a unique conclusion that for a particular far-end signal level and echo path, each τ has a maximum achievable value of convergence level. Convergence can be achieved, in principle, anywhere below this level, as μ can be decreased as required, without compromising the stability of the convergence process. The stability of the convergence process is disturbed only when μ is increased beyond μ_{max} *measured*.

It is also observed that the maximum achievable convergence level decreases with the increase in τ as well as with the decrease in the far-end signal level. For example in Table 5.2, maximum achievable convergence level is -10 dB for τ = 2.0, -24 dB for τ = 2.5 and -34 dB for τ = 3.0. If we have to compare different levels of the far-end signal for the same value of τ, say τ = 3.0. Then it is -34, -42, -49 and -57 dB for the corresponding levels of far-end signal as -15, -20, -25 and -30 dB (Tables 5.2–5.5).

In Tables 5.2–5.5 μ_{max} *measured* decreases with the increase in τ. Initial

higher and later lower values of μ_{max} *measured* are the result of the corresponding initial higher and later lower values of convergence level. However in the simulations of Section 5.2, optimum value of μ, increases with the increase in τ. That is a result of keeping the convergence level constant while increasing τ. In Tables 5.2–5.5, μ_{max} *calculated* increases with the increase in τ. This is because of the presence of the term $|f_k|^{\tau-2}$ in Equation (3.29). As $|f| < 1$, an increase in τ results in a decrease in $|f_k|^{\tau-2}$. If a term decreases in the denominator of a fraction, the whole fraction increases.

The big difference between μ_{max} *calculated* and μ_{max} *measured* could be the result of two factors. The first one is the assumption that we are already very near to the minimum point at the start of our analysis. μ_{max} *calculated* is obtained from the results of these analysis. This assumption (Section 3.4) may have contributed in widening the gap between μ_{max} *calculated* and μ_{max} *measured.* The second factor could be the approximations made in the analysis while obtaining the expression for the calculation of μ_{max} *calculated.* This may as well have resulted in widening the gap between μ_{max} *calculated* and μ_{max} *measured.*

cost function power τ	far-end signal is at -15 dB		
	μ_{max} calculated	μ_{max} measured	convergence level, dBs
2.0	0.031	0.028	-10
2.1	0.032	0.026	-12
2.2	0.033	0.023	-16
2.3	0.035	0.021	-18
2.4	0.037	0.019	-21
2.5	0.040	0.016	-24
2.6	0.042	0.014	-26
2.7	0.046	0.012	-28
2.8	0.049	0.010	-30
2.9	0.054	0.009	-32
3.0	0.059	0.007	-34
3.1	0.064	0.006	-35
⋮	⋮	⋮	⋮

Table 5.2: Calculated and measured values of step size μ corresponding to various values of cost function power τ along with the maximum achievable values of convergence level in decibels. Signal sequences are binary. Far-end signal is at -15 dB as compared with that of the near-end signal. The first echo path model is used.

cost function power τ	far-end signal is at -20 dB		
	μ_{max} calculated	μ_{max} measured	convergence level, dBs
2.0	0.031	0.028	-15
2.1	0.034	0.027	-18
2.2	0.038	0.024	-23
2.3	0.042	0.023	-25
2.4	0.047	0.019	-28
2.5	0.053	0.017	-30
2.6	0.060	0.014	-33
2.7	0.068	0.012	-36
2.8	0.078	0.011	-38
2.9	0.090	0.009	-40
3.0	0.104	0.008	-42
3.1	0.121	0.007	-44
⋮	⋮	⋮	⋮

Table 5.3: Calculated and measured values of step size μ corresponding to various values of cost function power τ along with the maximum achievable values of convergence level in decibels. Signal sequences are binary. Far-end signal is at -20 dB as compared with that of the near-end signal. The first echo path model is used.

cost function power τ	far-end signal is at -25 dB		
	μ_{max} calculated	μ_{max} measured	convergence level, dBs
2.0	0.031	0.029	-18
2.1	0.036	0.028	-24
2.2	0.042	0.025	-28
2.3	0.050	0.022	-32
2.4	0.059	0.019	-35
2.5	0.070	0.017	-37
2.6	0.084	0.015	-40
2.7	0.102	0.013	-42
2.8	0.124	0.011	-45
2.9	0.151	0.009	-47
3.0	0.185	0.008	-49
3.1	0.228	0.007	-51
⋮	⋮	⋮	⋮

Table 5.4: Calculated and measured values of step size μ corresponding to various values of cost function power τ along with the maximum achievable values of convergence level in decibels. Signal sequences are binary. Far-end signal is at -25 dB as compared with that of the near-end signal. The first echo path model is used.

cost function power τ	far-end signal is at -30 dB		
	μ_{max} calculated	μ_{max} measured	convergence level, dBs
2.0	0.031	0.030	-21
2.1	0.038	0.028	-30
2.2	0.047	0.025	-35
2.3	0.059	0.022	-38
2.4	0.074	0.020	-42
2.5	0.094	0.017	-45
2.6	0.119	0.015	-47
2.7	0.153	0.013	-50
2.8	0.197	0.011	-53
2.9	0.254	0.010	-54
3.0	0.329	0.008	-57
3.1	0.429	0.007	-59
⋮	⋮	⋮	⋮

Table 5.5: Calculated and measured values of step size μ corresponding to various values of cost function power τ along with the maximum achievable values of convergence level in decibels. Signal sequences are binary. Far-end signal is at -30 dB as compared with that of the near-end signal. The first echo path model is used.

5.4 Dispersion of far-end signal in addition to attenuation

For the preceding computer simulation purposes we have used only attenuation in the far-end signal. The present thesis is based on various levels of attenuation such as -15, -20, -25 and -30 dB but no dispersion at all in the incoming far-end signal. The real systems may introduce both the attenuation and dispersion in a signal during the course of its passage through the channel. We therefore, also look at whether the algorithm behaves similarly when dispersion is introduced into the far-end signal in addition to the attenuation.

The addition of dispersion into the far-end signal is modelled by passing the signal through a transversal digital filter after attenuation. The required dispersion can be added simply by setting the coefficients of the digital filter with certain values. Let us call this filter the far-end dispersion filter. For simplicity we choose a 32 tap FIR filter similar to the one used for the first and second echo path models. i.e. the transfer function of the filter is $z/(z-a)$. The feedback coefficient a is selected arbitrarily in such a way that the 32nd coefficient of the far-end dispersion filter is at -100 dB as compared with that of the first one. A detailed procedure for the calculation of the feedback coefficient a is explained in Section 4.3. Using this procedure, a came out to be equal to 0.689778. Numerical values of all the 32 tap coefficients are summarised in Appendix A.2.3. The simulation setup used is given in Figure 4.5. For computer simulations the attenuation level of the far-end signal is arbitrarily chosen as -15 dB with the corresponding convergence level as -35 dB. The rest of the simulation procedure is similar to that explained in Section 4.6 and 5.2. Starting

from 2.0, τ was incremented by 0.1 at each simulation. The corresponding optimum value of step size μ was used which was calculated using the steps described in Section 4.6. Each simulation result was ensemble averaged by 20 times before recording the convergence time. The resulting convergence times are plotted versus cost function powers τ, in Figure 5.19. A look at this figure shows that the introduction of dispersion in the far-end signal does not violate the basic argument, i.e. convergence time reduces with the increase in τ. Ignoring the additional noise and absolute values in Figure 5.19, the basic shape is similar to that of the middle (first echo path model) curve of Figure 5.1. So the new adaptation algorithm remains valid whether the dispersion is present in the far-end signal or not.

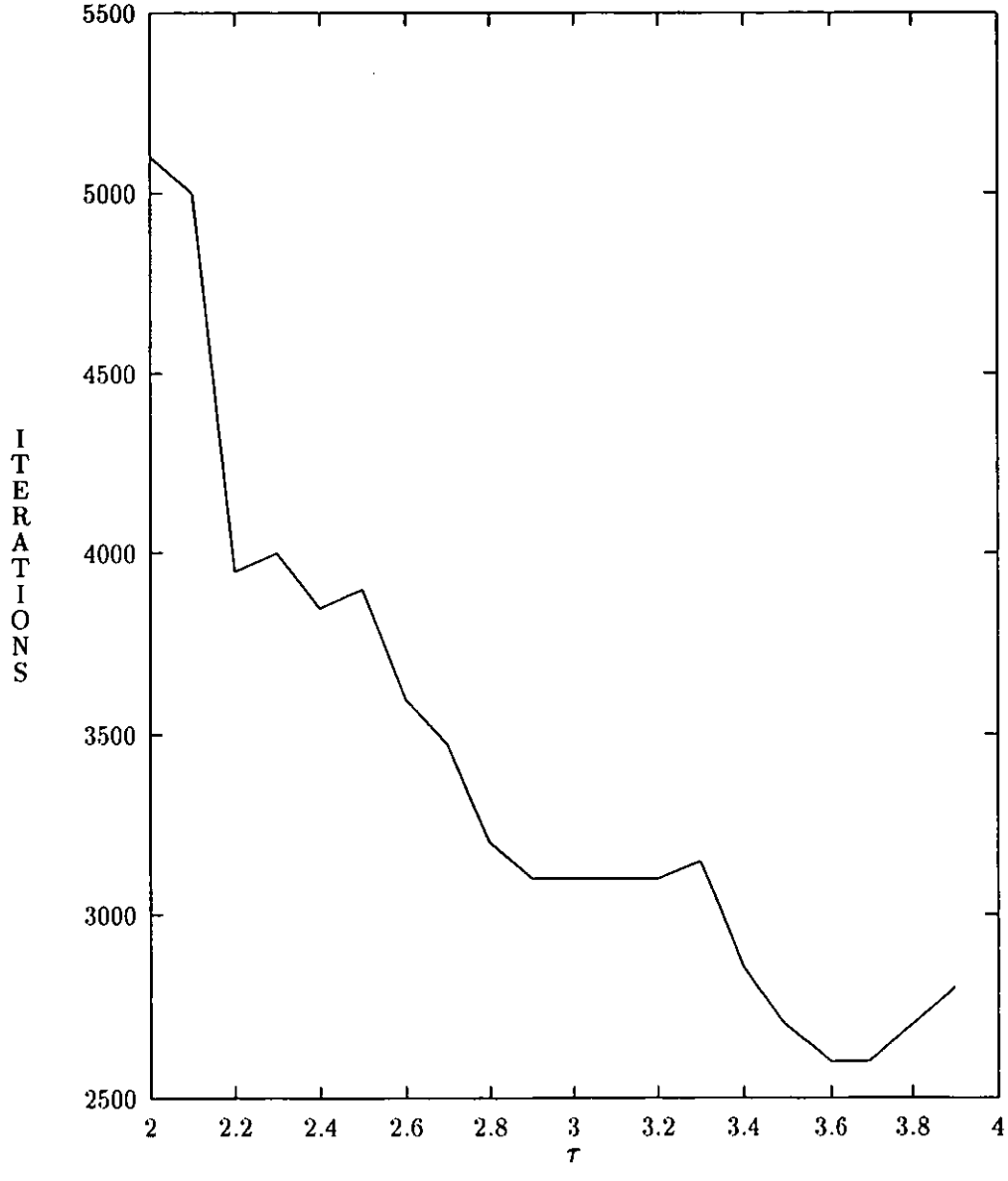


Figure 5.19: Convergence time versus cost function power τ is plotted. Convergence was achieved at -35 dB. Far-end signal is attenuated at -15 dB and passed through a dispersion.

5.5 Achievement of extended tolerance

Referring to the figures of Section 5.2 we have a maximum value of τ for the fastest convergence, in each case. Now is it the absolute maximum or can we play with and stretch the curve a bit further in search of a wide region of the low sensitivity of convergence time to the variations in τ? The answer to the above question is that let us first investigate the parameters causing instability. Then we may be able to control some of these.

If we look at the adaptation algorithm (3.11) we find that this is due to the high values of the gradient and error signal e_k. In the modified stochastic gradient algorithm there is a term $|e_k|^{(\tau-1)}$. As e_k crosses the limits of ± 1, with $\tau > 2$ or $(\tau - 1) > 1$, a situation arises where a positive number greater than 1 is being raised to the power of another positive number greater than 1. This results in a bigger number which in turn can produce a larger e_k in the next iteration resulting again in a larger than the previous $|e_k|^{(\tau-1)}$ and so on, leading towards instability. A typical error signal is plotted against number of iterations in Figure 5.20.

The instability can be handled for a while by reducing the gradient i.e., by reducing either the exponent τ or the error signal e_k. In search of a wider range of low sensitivity region near the ends of the curves and to explore further the usefulness of the algorithm, the exponent τ was switched from a higher value to a lower one during the convergence whenever e_k reaches or crosses the limits of ± 1. And was switched back to the respective higher value when e_k fell again in the limits of ± 1. For this purpose a check was performed on e_k during each iteration and action was taken accordingly. The switching was continued throughout the whole simulation and in all simulations for the sake

of consistency. The value chosen for lower cost function power was 2.0 as in the LMS algorithm. Results of the computer simulations performed in this way are given in the next section.

The exponent τ and its relative optimum step size μ were switched from higher values to lower ones (those for LMS algorithm) and back, depending upon the value of the error signal e_k, as given below

$$\text{if } |e_k| < 1 \text{ then, cost function's exponent} = \tau, \text{ step size} = \mu$$

$$\text{where } \tau > 2.0, \mu \text{ is the optimum value of step size for } \tau$$

$$\text{if } |e_k| \geq 1 \text{ then, cost function's exponent} = 2.0, \text{ step size} = \mu_2$$

$$\mu_2 \text{ is the optimum value of step size for } \tau = 2.0$$

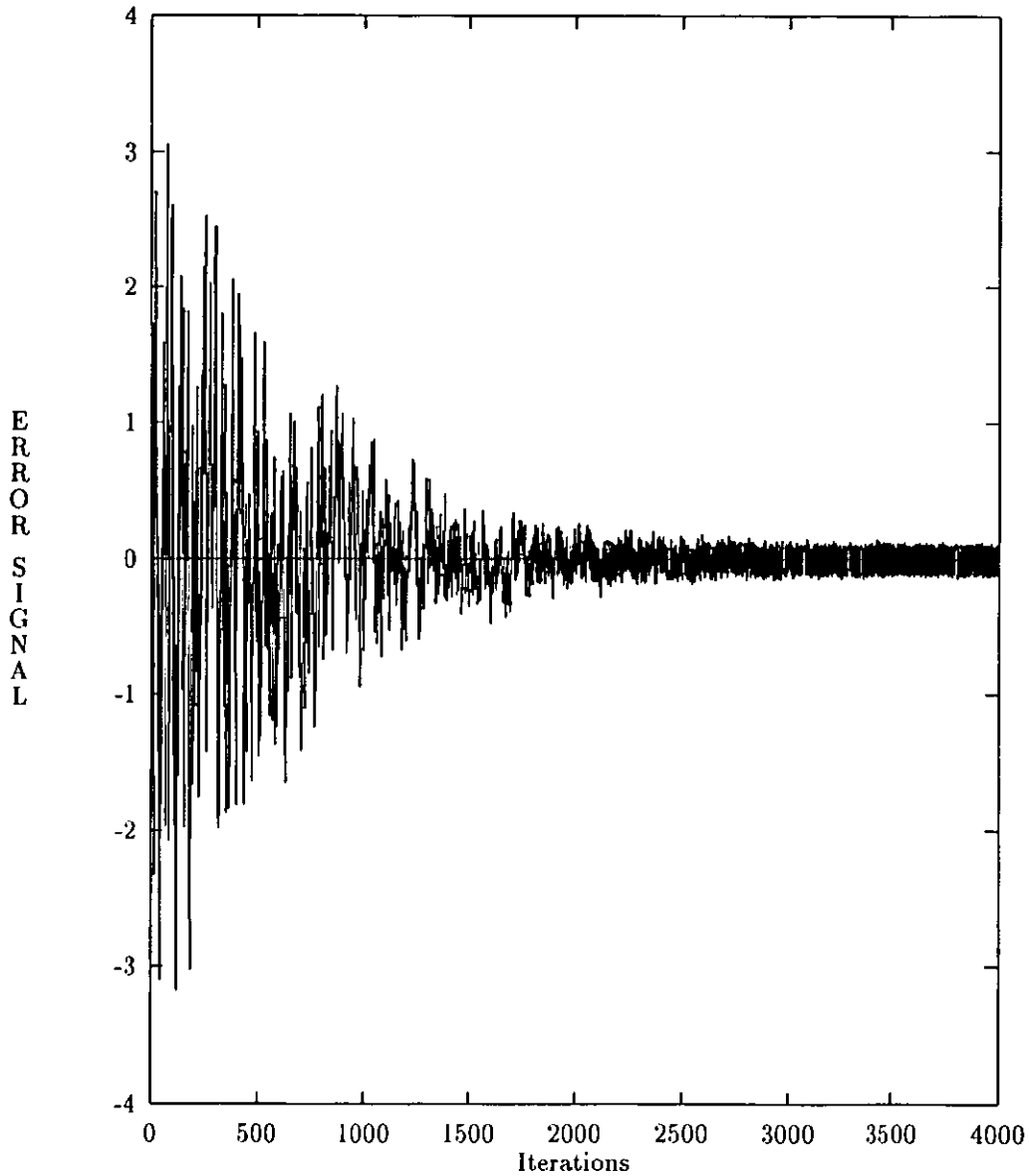


Figure 5.20: A typical error signal e_k versus convergence time in terms of the number of iterations. This error signal corresponds to $\tau = 2.0$, $\mu = 0.00085$, far-end signal level at -20 dB, convergence level at -40 dB and binary (± 1) data for the first echo path model.

5.6 Stretching tolerance by switching to lower gradient during convergence

The simulation results presented in this section are obtained by performing computer simulations in a similar way to that of Section 5.2. The computer simulations were performed with the same parameters, i.e. the signal sequences were pseudo random and binary. The same four attenuation levels of far-end signals were used, which are -15, -20, -25 and -30 dB. Resulting in four convergence levels of -35, -40, -45 and -50 dB respectively. The difference is that in the present case the cost function power τ was switched back and forth along with the corresponding step size μ, during the course of convergence. This was done according to the criteria set in Section 5.5. The rest of the procedures remain the same as previously described in Section 4.6 and 5.2. Different power levels used in simulations along with the maximum values of cost function power τ attained are summarised in Table 5.6.

Figures 5.21–5.24 show the results for the far-end signal levels of -15, -20, -25 and -30 dB respectively. Each of the figures contain the results of all three echo path models for a particular far-end signal level and convergence level. By observing the curves in these four figures, we can easily conclude that there are some features which are common with those of the Figures 5.1–5.4. Convergence time decreases as we increase cost function power τ. Initially the decrease in convergence time is fast. This decrease becomes slower in the next part of the curves. In these later parts of the curves convergence time does not change while we keep on changing τ. These parts of the curves provide us with the tolerance. Convergence time is tolerant to the changes in τ for a certain range

near-end signal level dBs	far-end signal level dBs	conver -gence level dBs	(far-end signal level) -(convergence level) dBs	max. value of τ attained		
				1st model	2nd model	3rd model
0	-15	-35	20	3.7	3.9	3.1
0	-20	-40	20	3.3	3.4	2.9
0	-25	-45	20	3.1	3.2	2.7
0	-30	-50	20	2.9	3.0	2.6

Table 5.6: Various combinations of power levels and the maximum value of cost function power τ for fastest convergence with each combination. Data streams were binary. With switching of gradient during convergence.

of τ depending upon the other parameters. This tolerance gives us a working range of τ, a better choice of design for real systems. A bathtub like feature is emerging from the curves. This means that after the plateau of tolerance, convergence time once again starts increasing with increments in τ. The present form of the algorithm (i.e. with switching to lower gradient during convergence as per requirement) results in an extended range of τ as compared with that of the non-switched case.

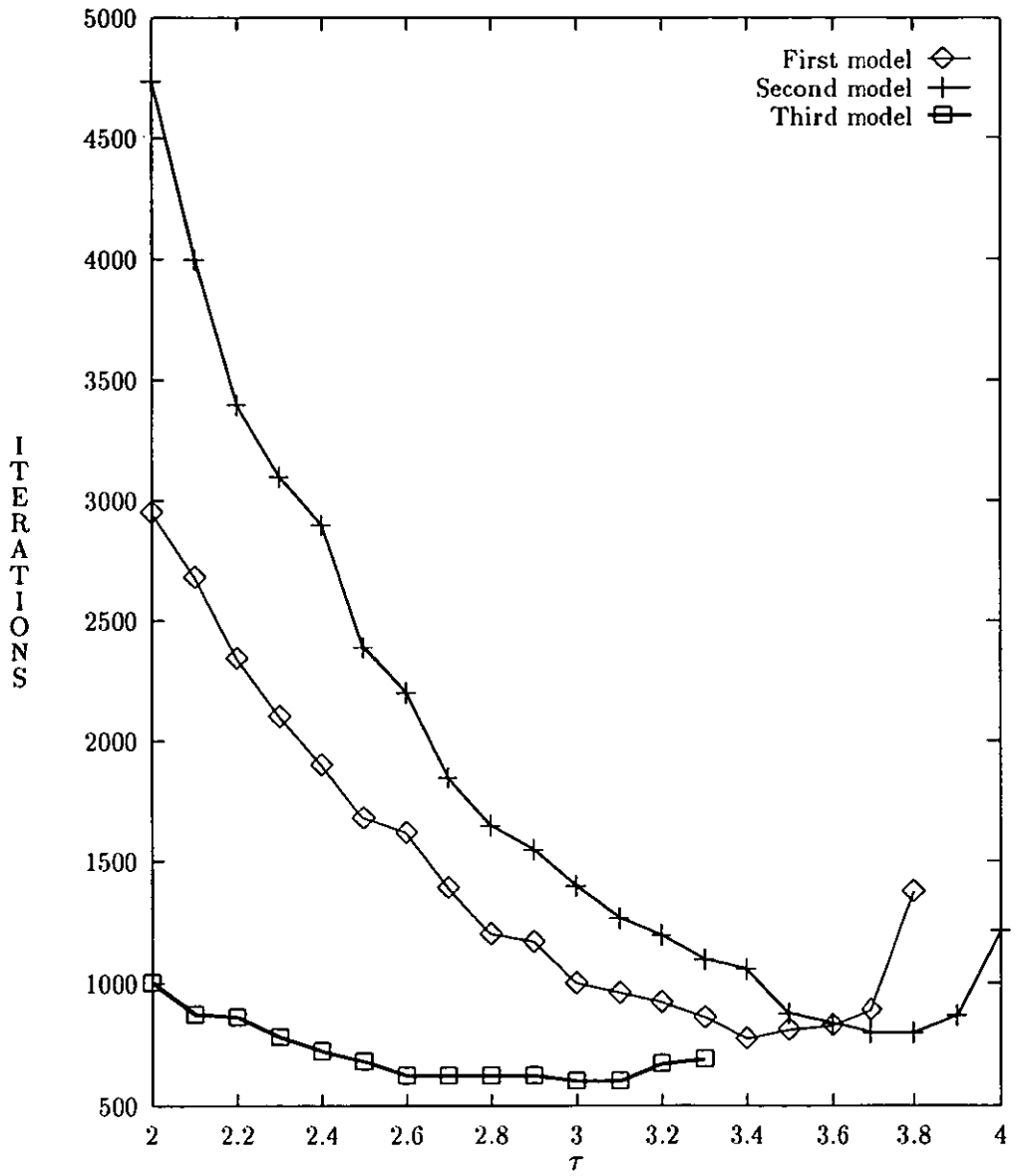


Figure 5.21: Convergence time versus cost function power τ is plotted for all three echo path models. Far-end signal is at -15 dB and convergence was achieved at -35 dB as compared with that of the near-end signal. Gradient was switched back and forth during convergence.

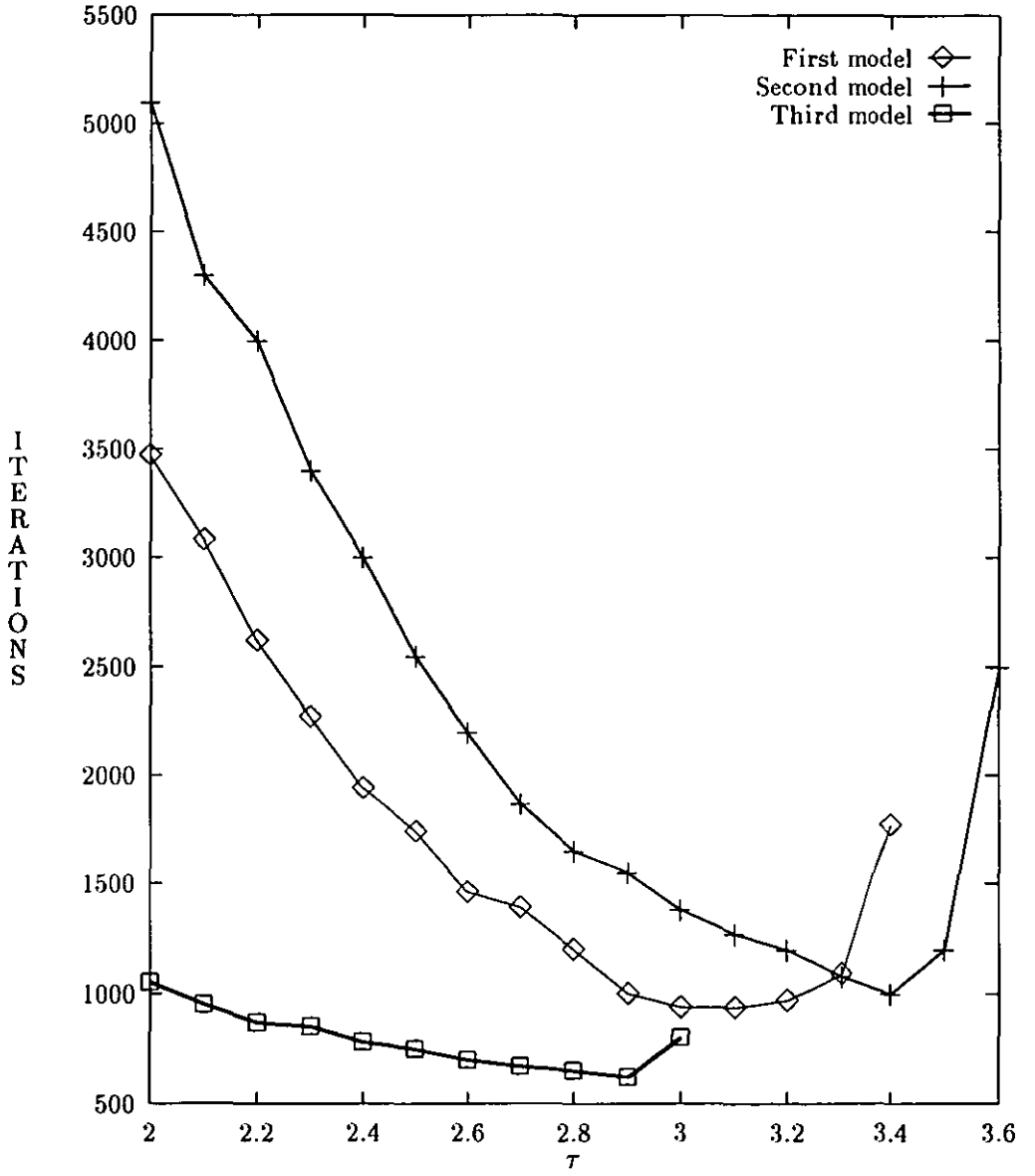


Figure 5.22: Convergence time versus cost function power τ is plotted for all three echo path models. Far-end signal is at -20 dB and convergence was achieved at -40 dB as compared with that of the near-end signal. Gradient was switched back and forth during convergence.

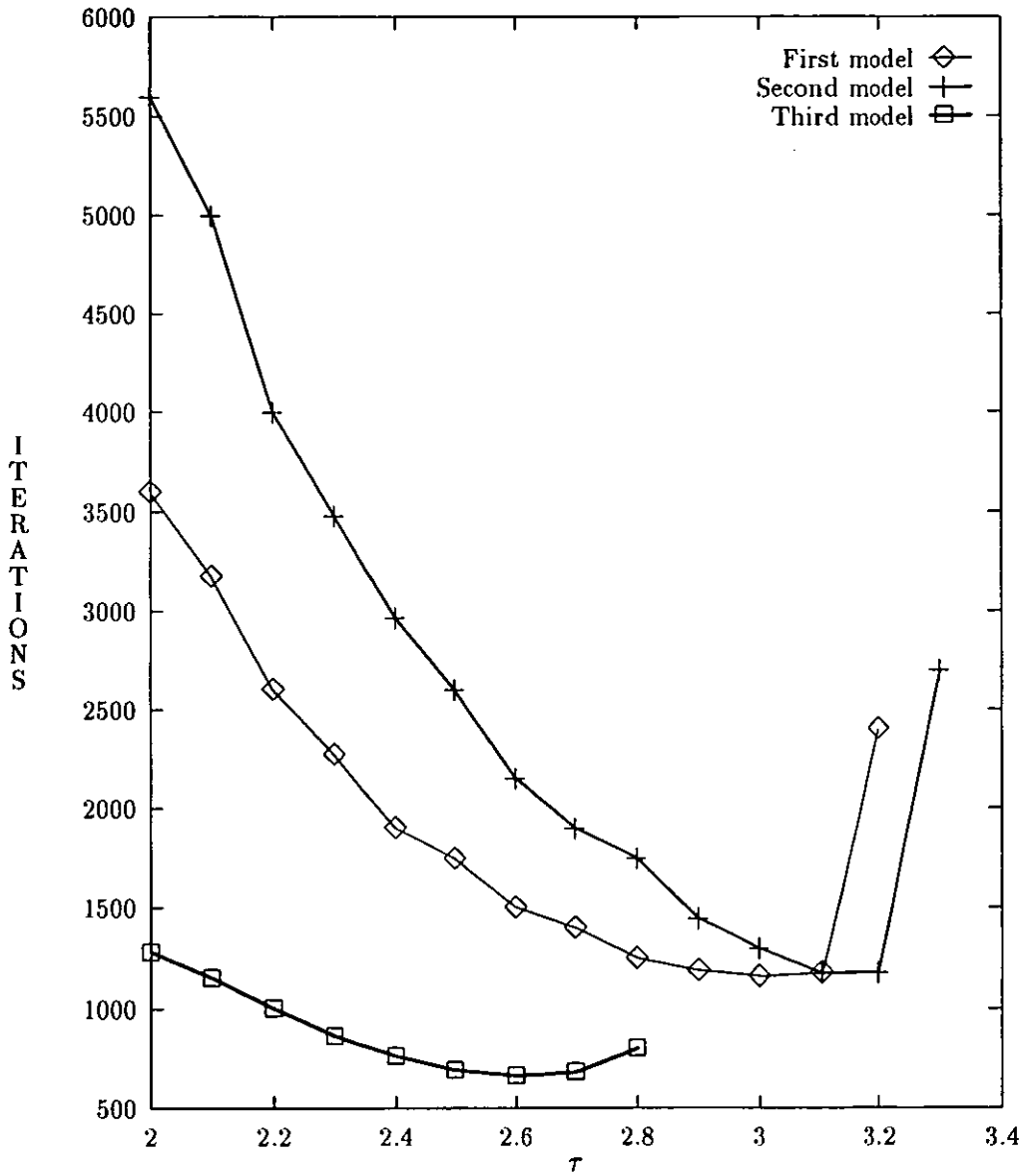


Figure 5.23: Convergence time versus cost function power τ is plotted for all three echo path models. Far-end signal is at -25 dB and convergence was achieved at -45 dB as compared with that of the near-end signal. Gradient was switched back and forth during convergence.

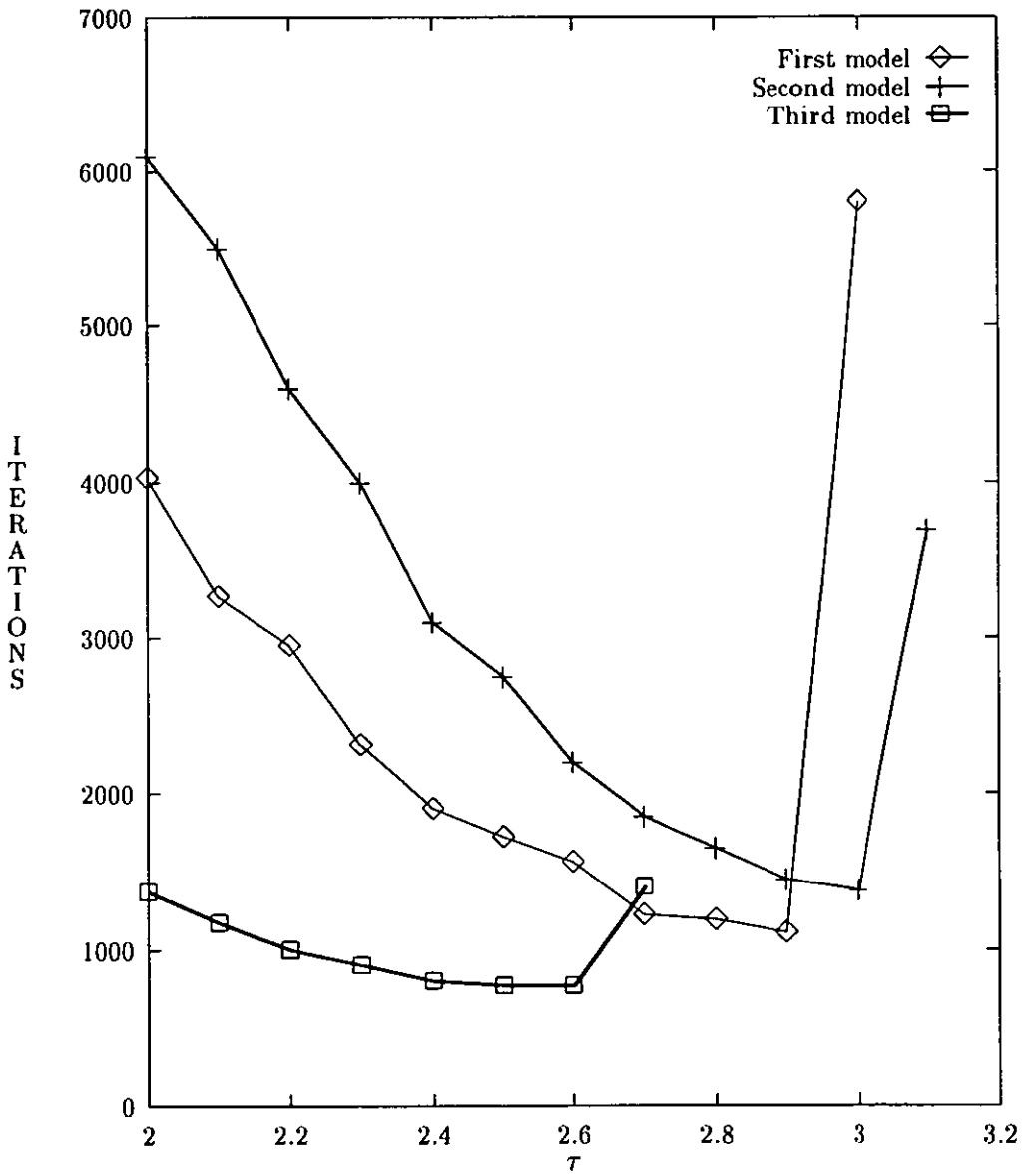


Figure 5.24: Convergence time versus cost function power τ is plotted for all three echo path models. Far-end signal is at -30 dB and convergence was achieved at -50 dB as compared with that of the near-end signal. Gradient was switched back and forth during convergence.

Some features of the curves in these figures are similar to that of the non-switched case. e.g. initial points of the curves are higher for comparatively smaller area transfer function echo path models (refer to Equations (5.1) and (5.2)). Which means more percentage decrease in convergence time if the echo path is of small area transfer function type. The lengths of the curves are also dependent upon the characteristics of the transfer function. The smaller the area of the transfer function is the larger the length of these curves will be. In other words we will achieve a higher value of τ for the fastest convergence.

The most important feature of interest, after reduction in convergence time, is the extended tolerance of the convergence time to the variations in τ.

As in Section 5.2, the same results (Figures 5.21–5.24) are presented in a different way. All the curves for a particular echo path model are combined in the same figure for comparison among various levels of far-end signal. Figures 5.25–5.27 represent results of simulations for all the four far-end signal levels for the first, second and third echo path models respectively.

As it is obvious from these figures, the curves are similar to that of Figures 5.5–5.7 except for the extent of the tolerant portion of the curves. After the separate start all the curves in a figure trace nearly the same path, before jumping up. Generally the less attenuated far-end signal curve starts from a relatively lower point and more attenuated far-end signal curve starts from a relatively higher point. The not so strict observance of this rule here, is, because of the measurement and observation noise and the close proximity of the points to each other. The tails of the curves end at different values of τ along x-axis. If the far-end signal is less attenuated, the corresponding curve has a relatively longer tail and vice versa.

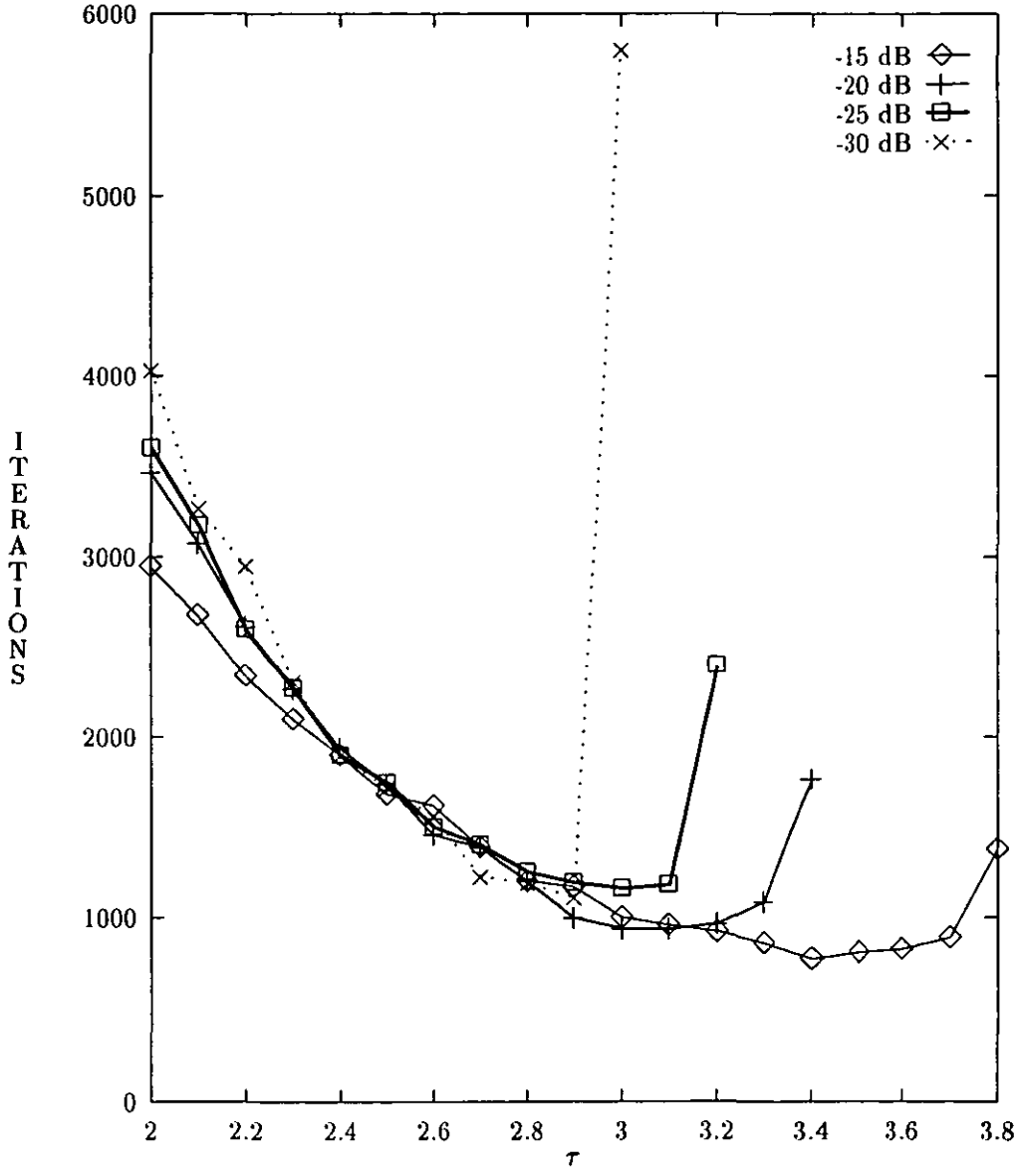


Figure 5.25: Convergence time versus cost function power τ is plotted for the first echo path model and all four far-end signal levels of -15, -20, -25 and -30 dB. Gradient was switched back and forth during convergence.

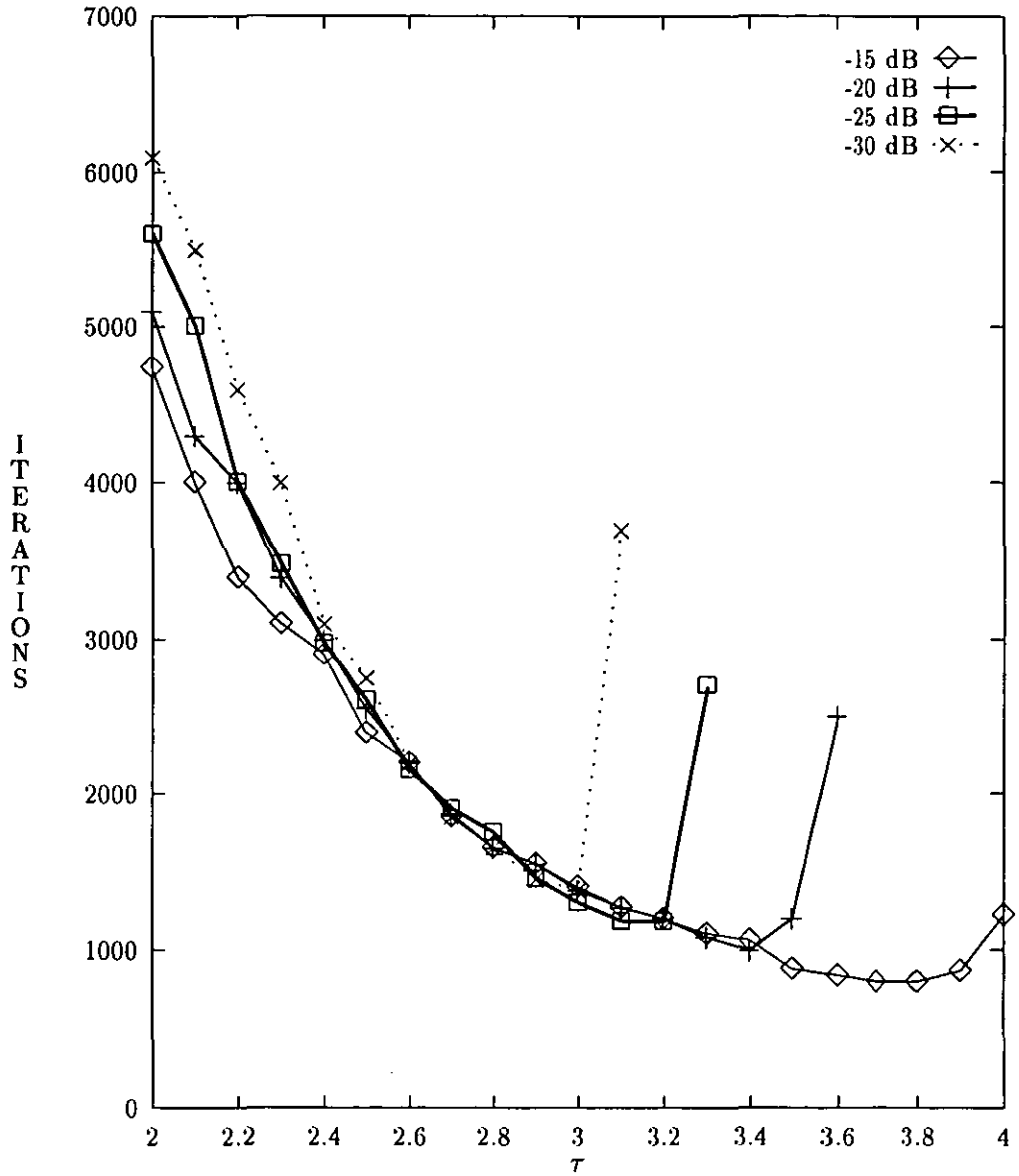


Figure 5.26: Convergence time versus cost function power τ is plotted for the second echo path model and all four far-end signal levels of -15, -20, -25 and -30 dB. Gradient was switched back and forth during convergence.

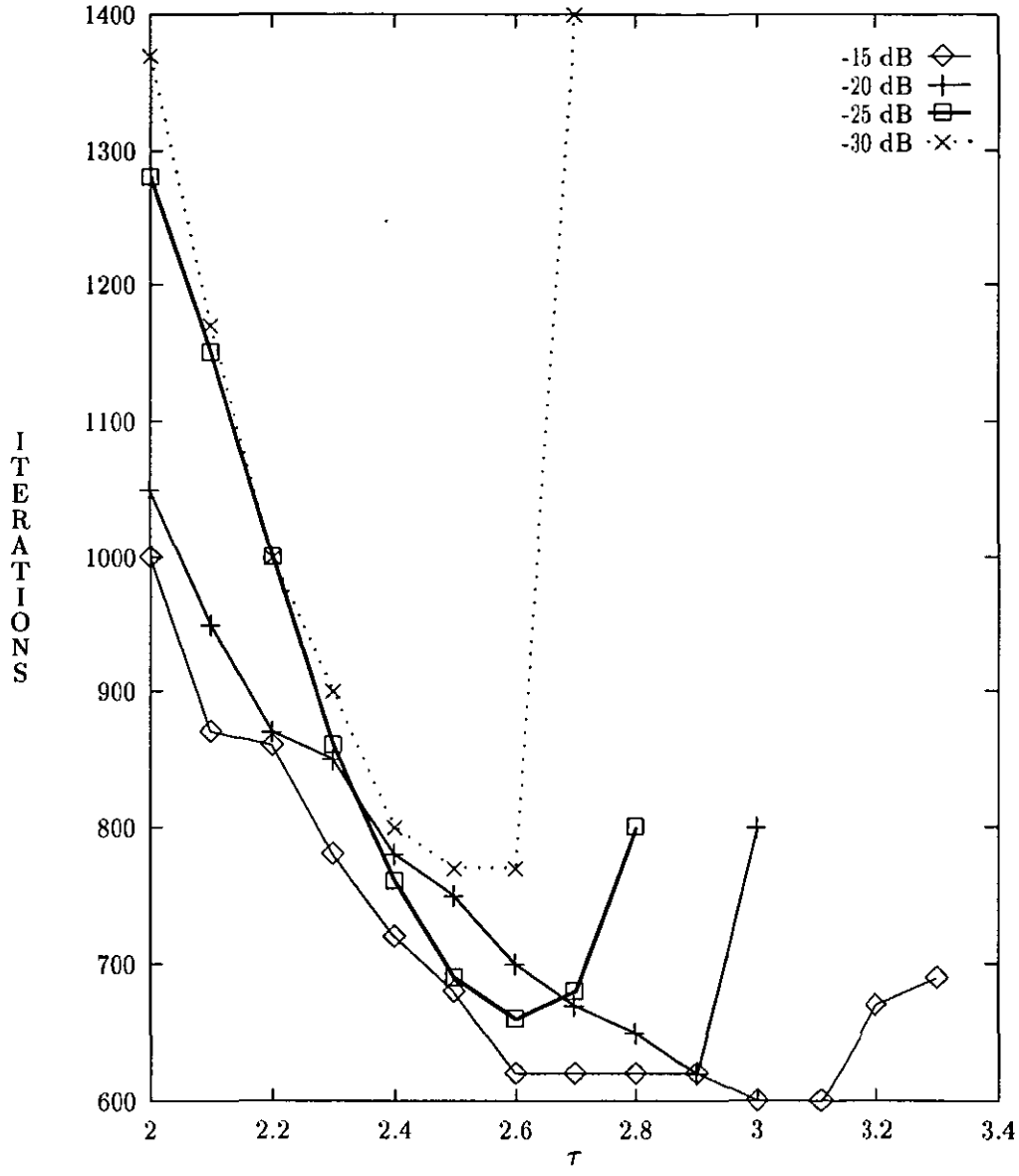


Figure 5.27: Convergence time versus cost function power τ is plotted for the third echo path model and all four far-end signal levels of -15, -20, -25 and -30 dB. Gradient was switched back and forth during convergence.

5.7 Switching behaviour

It is interesting to know how often the gradient is switched during convergence and how long it keeps going. The case of the first echo path model with far-end signal at -15 dB is chosen arbitrarily for computer simulations to find the number of switchings. Simulations were performed in a similar way as described earlier in this section. Convergence was achieved at -35 dB in each case without averaging. It was observed after the simulations that switchings take place for comparatively small amounts of time. The number of switchings back to lower gradient remain around 200 or below, for this particular case of first echo path model and -15 dB far-end signal. It is also worth noting that no switching occurs beyond 1000 iterations. The number of switchings is decreased as τ is increased. After reaching a minimum at $\tau = 3.4$, the number of switchings started increasing once again. This effect is similar to that of the decrease in convergence time with the increase in τ up to a certain τ, and increase in convergence time with the increase in τ after that. The result is presented in graphical form in Figure 5.28 where the number of switchings is plotted versus τ. If we ignore absolute values, the shape of this figure is similar to the middle curve of the Figure 5.21. A selection of the switching sequences for the four different values of τ, i.e. $\tau = 2.1$, 2.7, 3.4 and 3.8 are shown in Figures 5.29–5.32. From these figures we can readily observe the decrease in the number of switchings as τ is increased, and near the end an increase in the number of switchings as τ is increased.

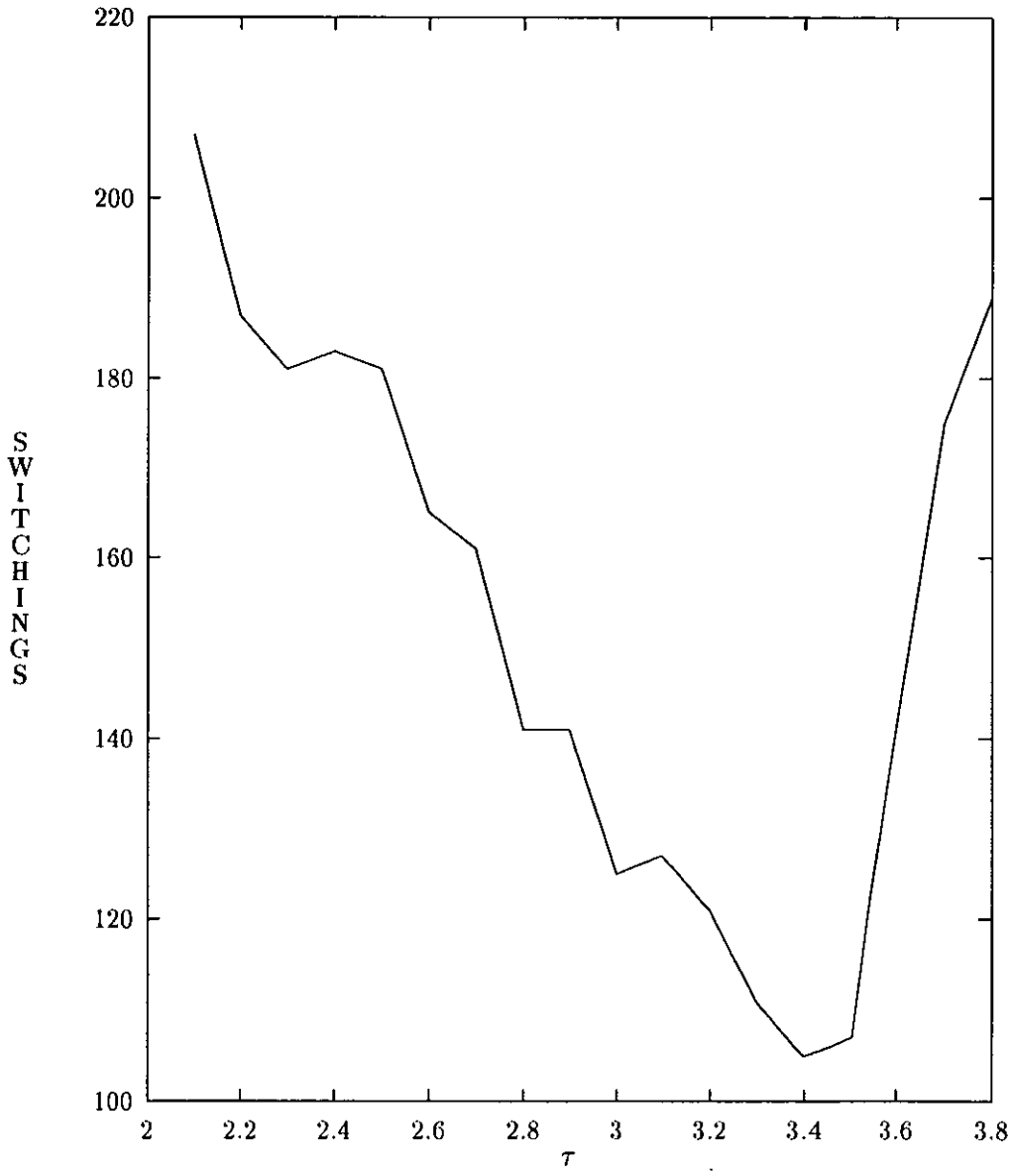


Figure 5.28: Number of switchings versus cost function power τ. For the case of binary signal sequences when far-end signal is at -15 dB and convergence is achieved at -35 dB for the first echo path model.

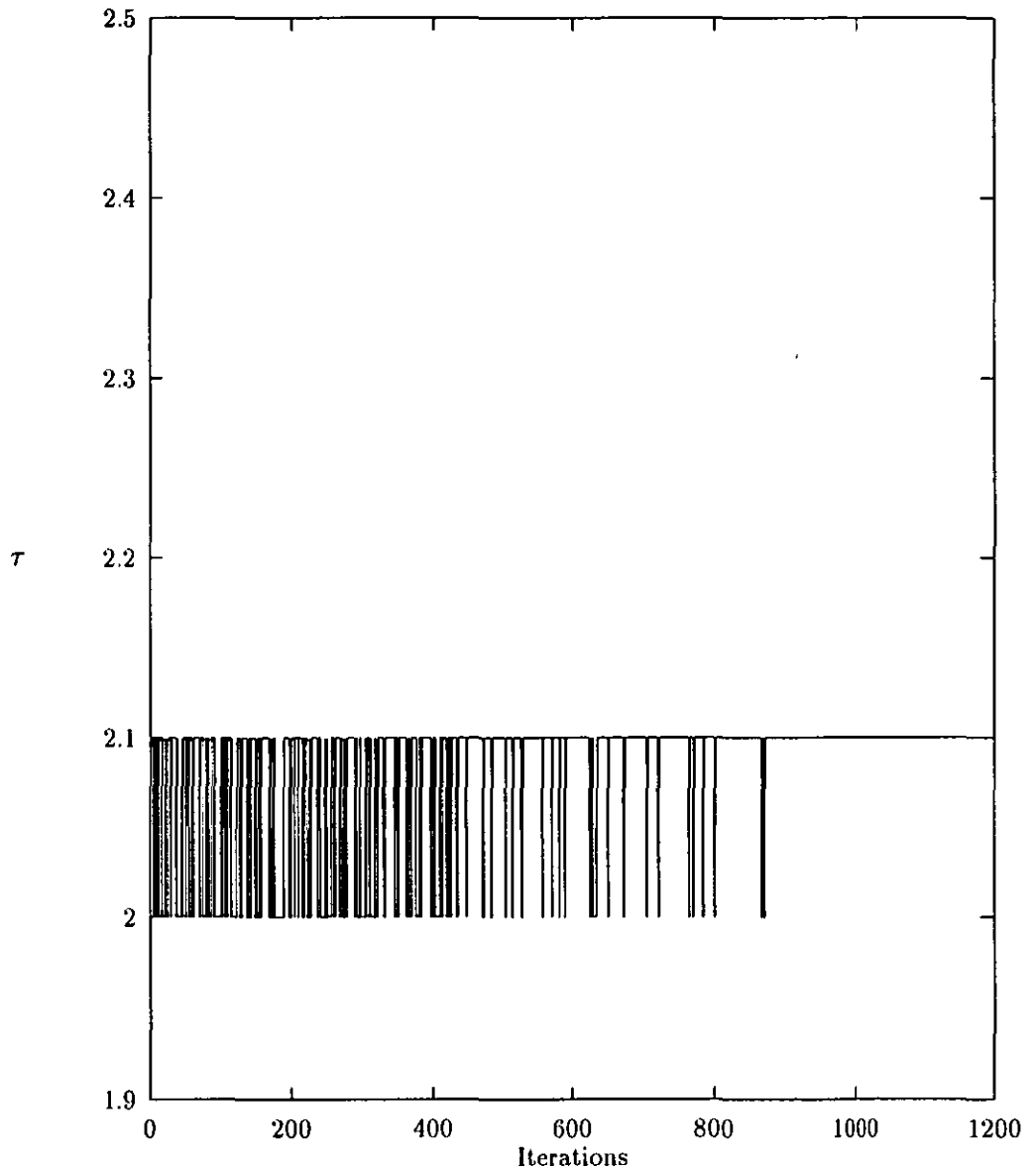


Figure 5.29: Switching sequence for $\tau = 2.1$. Simulations are performed with binary data sequences when far-end signal is at -15 dB and convergence is achieved at -35 dB for the first echo path model.

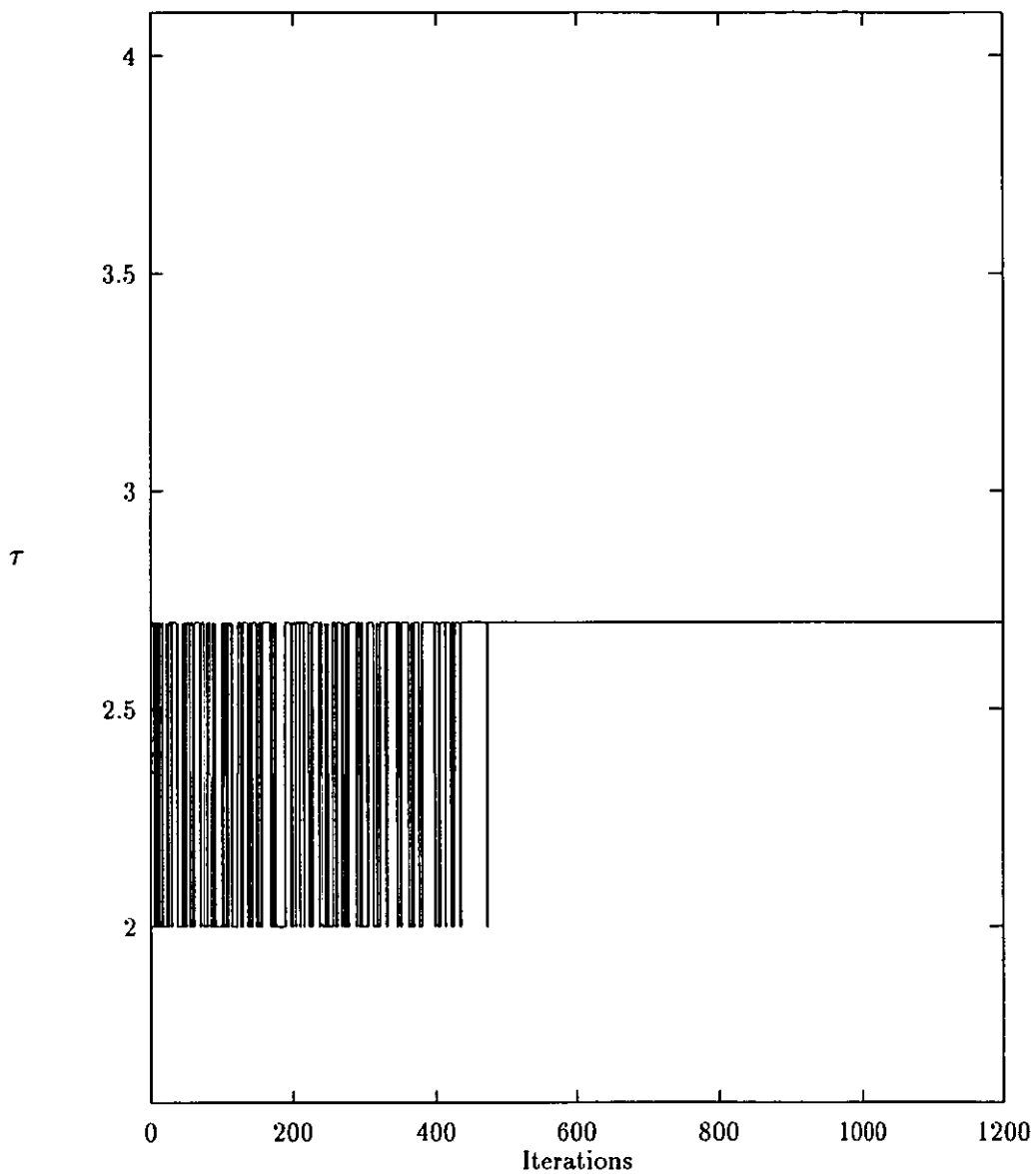


Figure 5.30: Switching sequence for $\tau = 2.7$. Simulations are performed with binary data sequences when far-end signal is at -15 dB and convergence is achieved at -35 dB for the first echo path model.

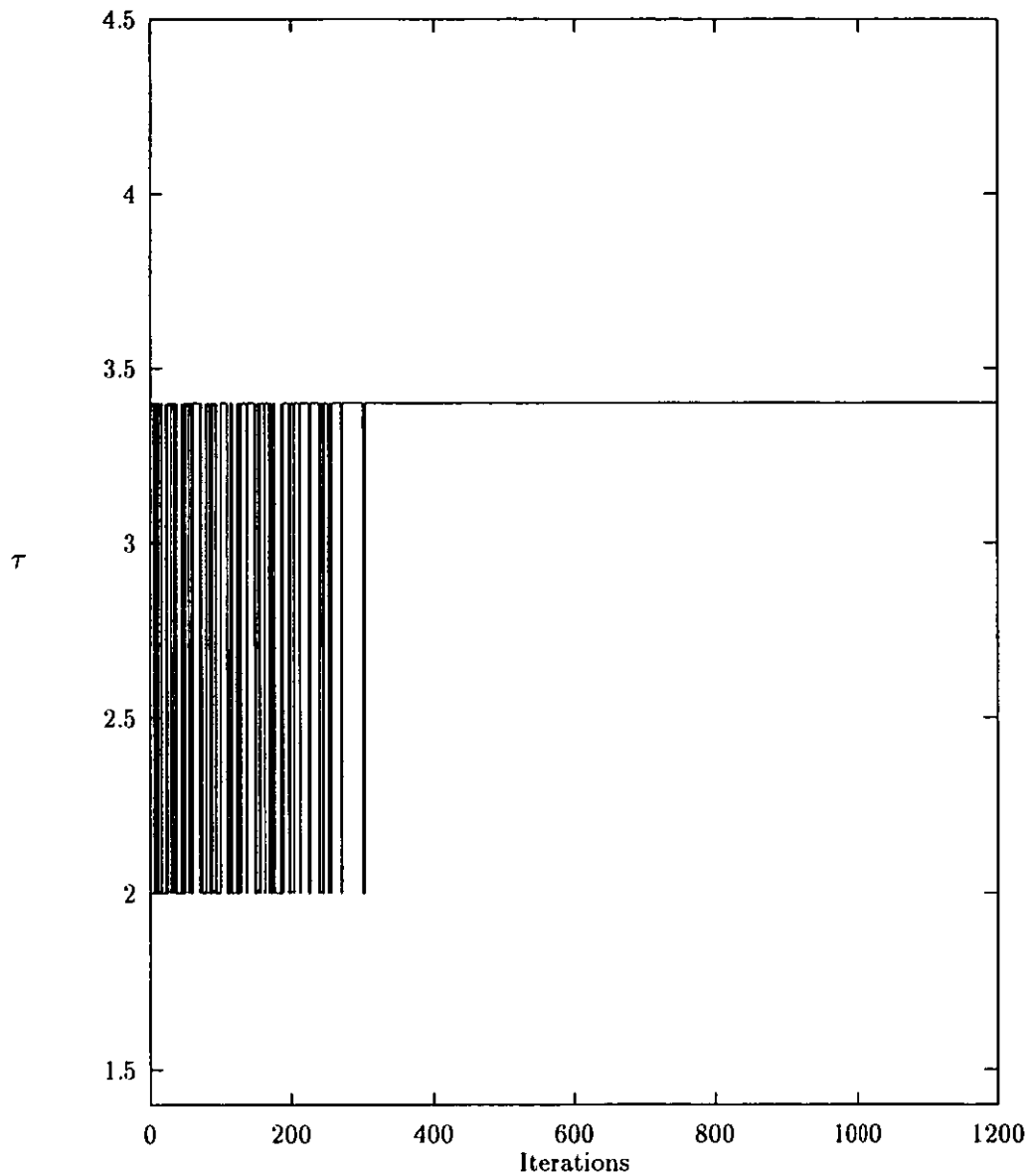


Figure 5.31: Switching sequence for $\tau = 3.4$. Simulations are performed with binary data sequences when far-end signal is at -15 dB and convergence is achieved at -35 dB for the first echo path model.

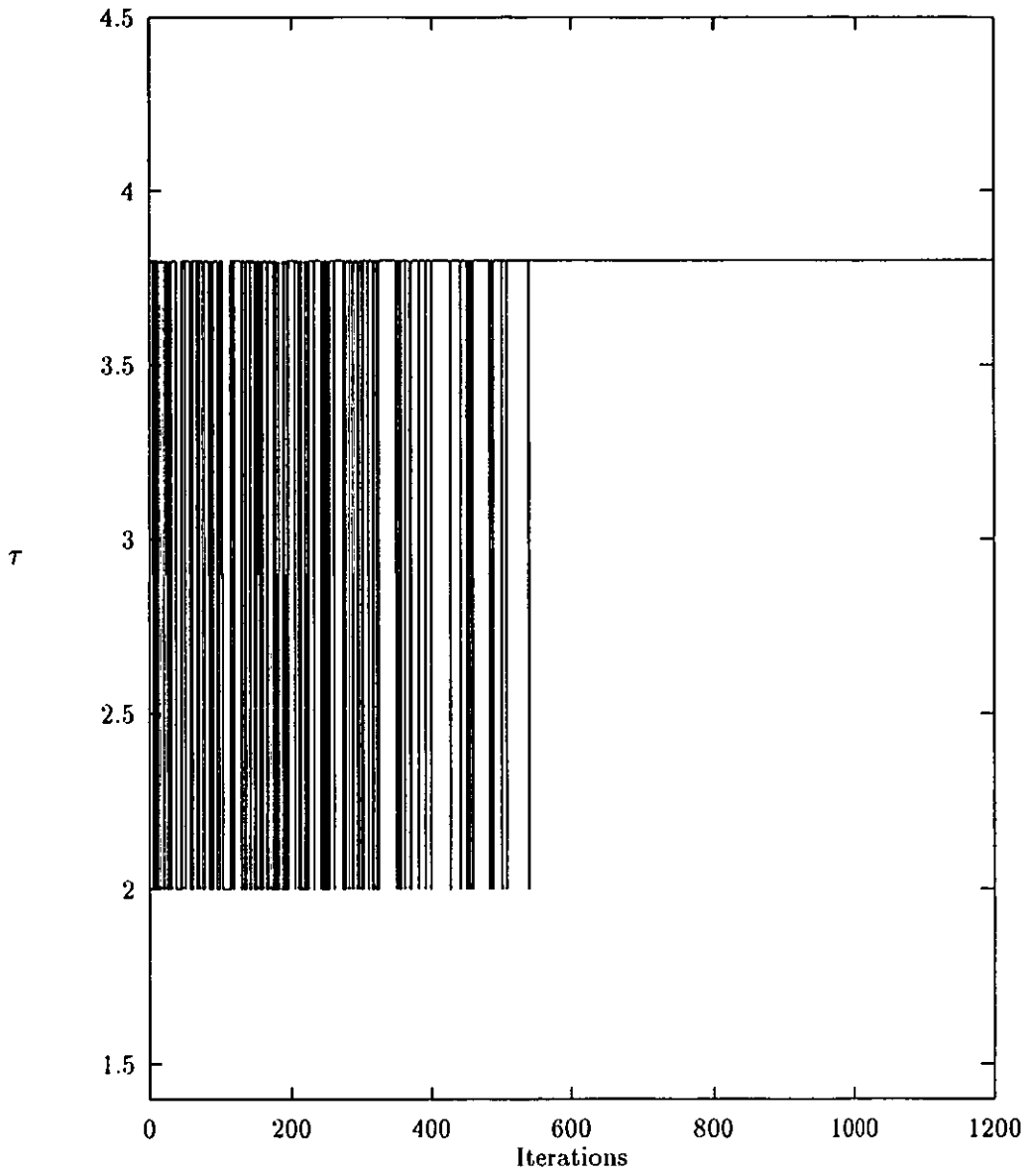


Figure 5.32: Switching sequence for $\tau = 3.8$. Simulations are performed with binary data sequences when far-end signal is at -15 dB and convergence is achieved at -35 dB for the first echo path model.

5.8 Conclusions

We can safely conclude that the non-quadratic cost function reduces the convergence time considerably as compared with that of the LMS algorithm.

Reductions in convergence time largely depends upon the area under the transfer function curve of the echo path model (refer to Equations (5.1) and (5.2)). Small area transfer functions tend to achieve relatively larger reductions in convergence time and vice versa. Large area transfer functions have comparatively better convergence times for mean square function anyway. Reductions in convergence time are almost independent of the level of far-end signal. The power of the far-end signal has a larger effect on the tails of the curves. The larger the power, the longer the tails i.e. a higher value of maximum τ for fastest convergence and vice versa.

We also conclude that the gradient can be switched back and forth from a higher to a lower and from a lower to a higher value during convergence process. The result is the achievement of an extended portion of the curve with greater tolerance and lower sensitivity to the variations in τ. This tolerance gives a better choice to the designer for the range of τ while designing real systems. Switchings occur only during the initial iterations of the convergence process. The number of switchings tend to decrease with the increase in τ. After reaching a certain minimum, the number of switchings starts increasing once again near the tail of the convergence time curve. This observation is similar to that of the decrease in the convergence time with the increase in τ.

The argument of reduction in convergence time with increase in τ remains valid after adding dispersion in the far-end signal sequence.

Chapter 6

Simulations with quaternary data

6.1 Introduction

Quaternary data simulations and results are presented in this chapter. Computer simulations were performed with pseudo random non-Gaussian data with four levels of +3, +1, -1 and -3, as described in Section 4.6. In order to illustrate a region of possible operating conditions, four cases with different far-end signal levels added to the returned signal were studied, each in two different ways. The far-end signal levels used were -15, -20, -25 and -30 dB. In all these cases, convergence was achieved 20 dB below the far-end signal level. Three different echo path models, described in Section 4.3, were used. Simulations are performed in the normal way as well as with the higher value of τ switched back to 2.0 and vice versa during the course of convergence.

6.2 Simulations with quaternary data

A pseudo random sequence of values +3, +1, -1 and -3 is used as a non-Gaussian quaternary signal at the input (transmitter) of the near-end of our echo canceller model. The echo of the input, or the reflected portion of the input signal sequence is received at the near-end along with the attenuated far-end signal. The far-end signal sequence is also modelled from a similar but independent sequence of values +3, +1, -1 and -3 with some attenuation introduced into it. Modelling of the four level data streams is discussed in detail in Section 4.2. The transfer function of the echo path is modelled as a digital echo path filter of length 32. The three echo path models used here are the same as for binary data sequences. Modelling of the echo paths is described in Section 4.3. The transfer function of the echo path is estimated by a digital adaptive filter of length 32 using the newly developed adaptation algorithm (3.11). The estimate of the echo path is then compared with that of the actual echo path. The resultant error signal is then used to obtain the new estimate of the echo path. In the process, we obtain a better than previous estimate at each iteration, which leads to convergence. The rest of the procedure is similar to that already described in the first few paragraphs of Section 5.2. After performing computer simulations we obtain the desired convergence curves. Convergence times in terms of the number of iterations were observed and recorded from these curves. These data are presented here in graphical form in Figures 6.1–6.4, from which we can easily see the extent of reduction in convergence time with the increase in cost function power τ.

Four different levels of far-end signal and convergence are used in the simulations with four level data, similar to that with the binary data. Convergence

near-end signal level dBs	far-end signal level dBs	conver -gence level dBs	(far-end signal level) -(convergence level) dBs	max. value of τ attained		
				1st model	2nd model	3rd model
0	-15	-35	20	3.7	4.0	2.8
0	-20	-40	20	3.3	3.8	2.6
0	-25	-45	20	3.2	3.2	2.6
0	-30	-50	20	2.9	3.1	2.5

Table 6.1: Various combinations of power levels and the maximum values of cost function power τ attained for the fastest convergence with each combination. Data streams were quaternary.

was always achieved 20 dB below the far-end signal levels. Respective power levels in decibels, along with the maximum values of τ attained for the fastest convergence in individual cases, are summarised in Table 6.1.

The results of the simulations with four level data are presented in a similar graphical form as were the simulations with binary data in the previous chapter. Convergence time in terms of the number of iterations versus cost function power τ is plotted. Related curves are combined in single figures, so that a comparison could be made among various features of the curves, in addition to the most significant observation of the decrease in convergence time with the increase in cost function power τ. Four figures are presented here. Each figure contains simulation results for all three echo path models for a particular attenuation level of the far-end signal. This way of presenting results provides a chance to

compare the effects of different echo path models.

Figures 6.1–6.4 show the results of simulations for the far-end signal levels of -15, -20, -25 and -30 dB respectively. The shape of all the curves in these figures is essentially similar. Absolute values are not the same, but there are some general trends. e.g. the convergence time (in terms of the number of iterations) tends to decrease as we go on increasing cost function power τ. In the initial portions of the curves, convergence time is more sensitive to the changes in cost function power τ. As we go on increasing cost function power τ, sensitiveness gradually decreases similar to the case of simulations with binary data. In the case of four level data, however, there is more noise in the curves. At a certain value of τ the algorithm stops converging exactly at a particular target level, however, it may still converge below that level. We may continue increasing τ by achieving convergence below the target level and recording the convergence time of the crossing point of the convergence curve and the target convergence level. The convergence process starts slowing down with further increments in the value of τ. The end result is that we obtain a bathtub like feature by plotting convergence time versus τ. A certain value of τ or a range of the values of τ achieve the minimum convergence time (or the fastest convergence), for a particular combination of echo path model and far-end signal. These maximum values of τ for the minimum possible convergence time are summarised in Table 6.1.

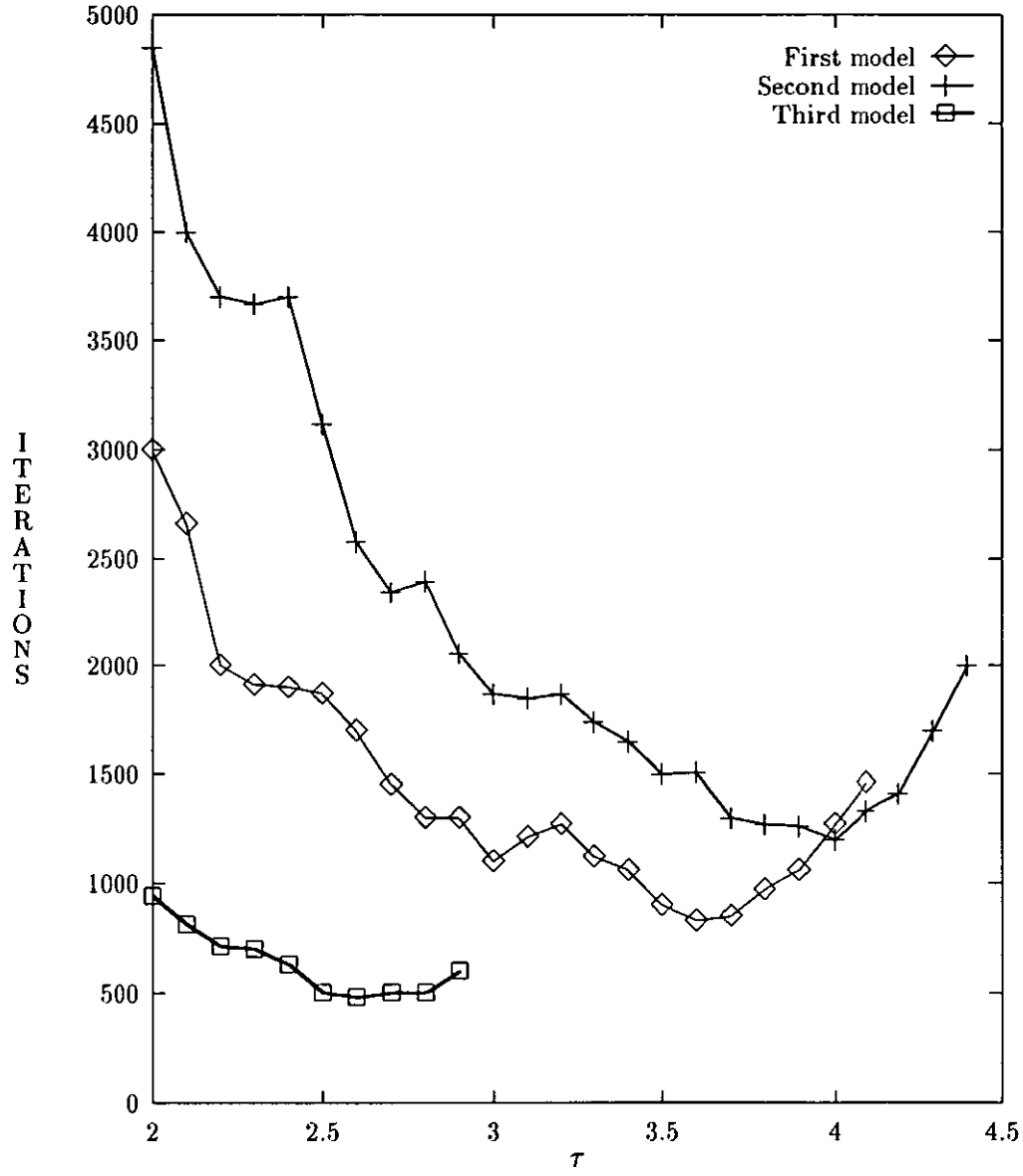


Figure 6.1: Convergence time versus cost function power τ is plotted for all three echo path models. Far-end signal is at -15 dB and convergence was achieved at -35 dB as compared with that of the near-end signal.

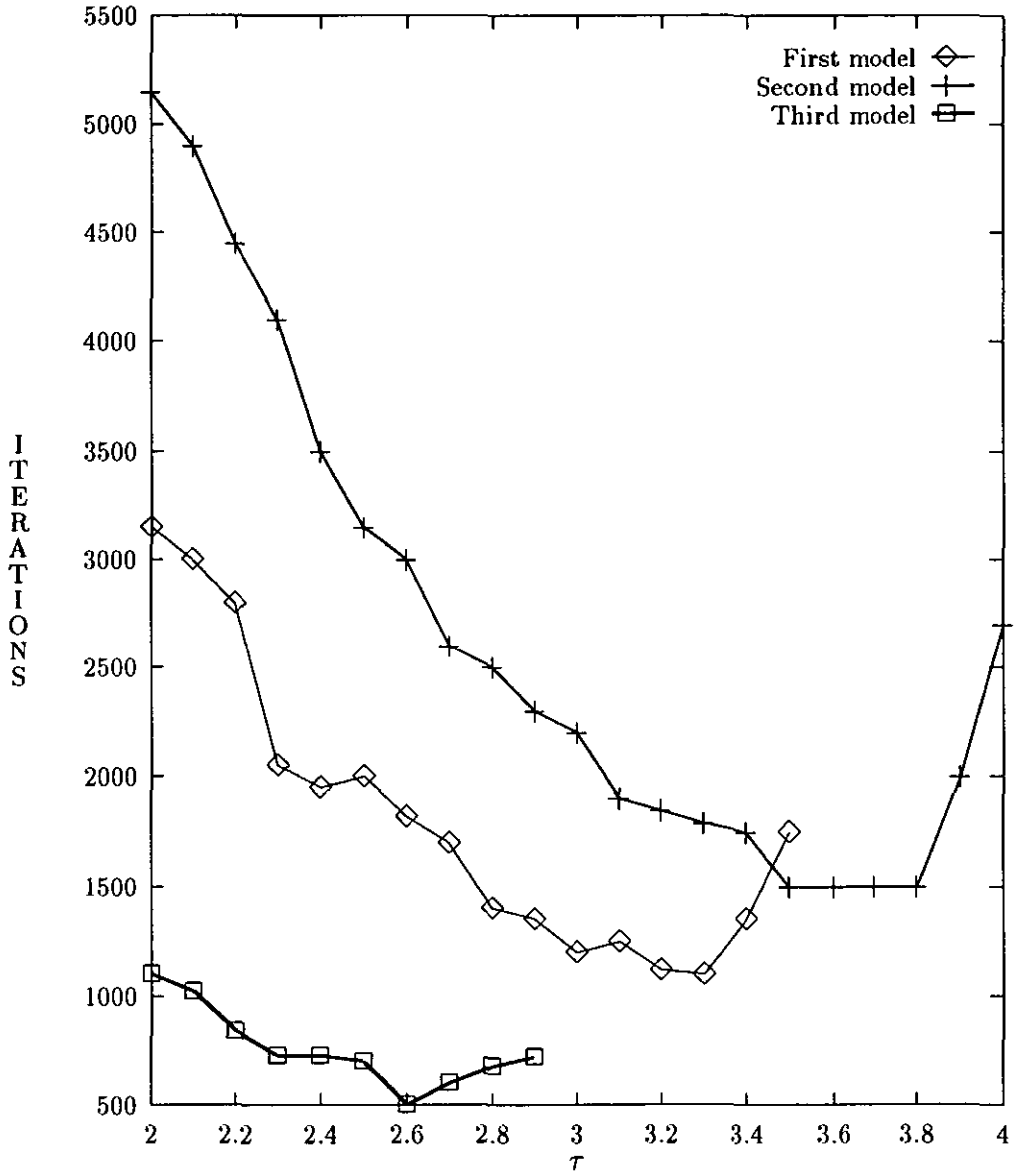


Figure 6.2: Convergence time versus cost function power τ is plotted for all three echo path models. Far-end signal is at -20 dB and convergence was achieved at -40 dB as compared with that of the near-end signal.

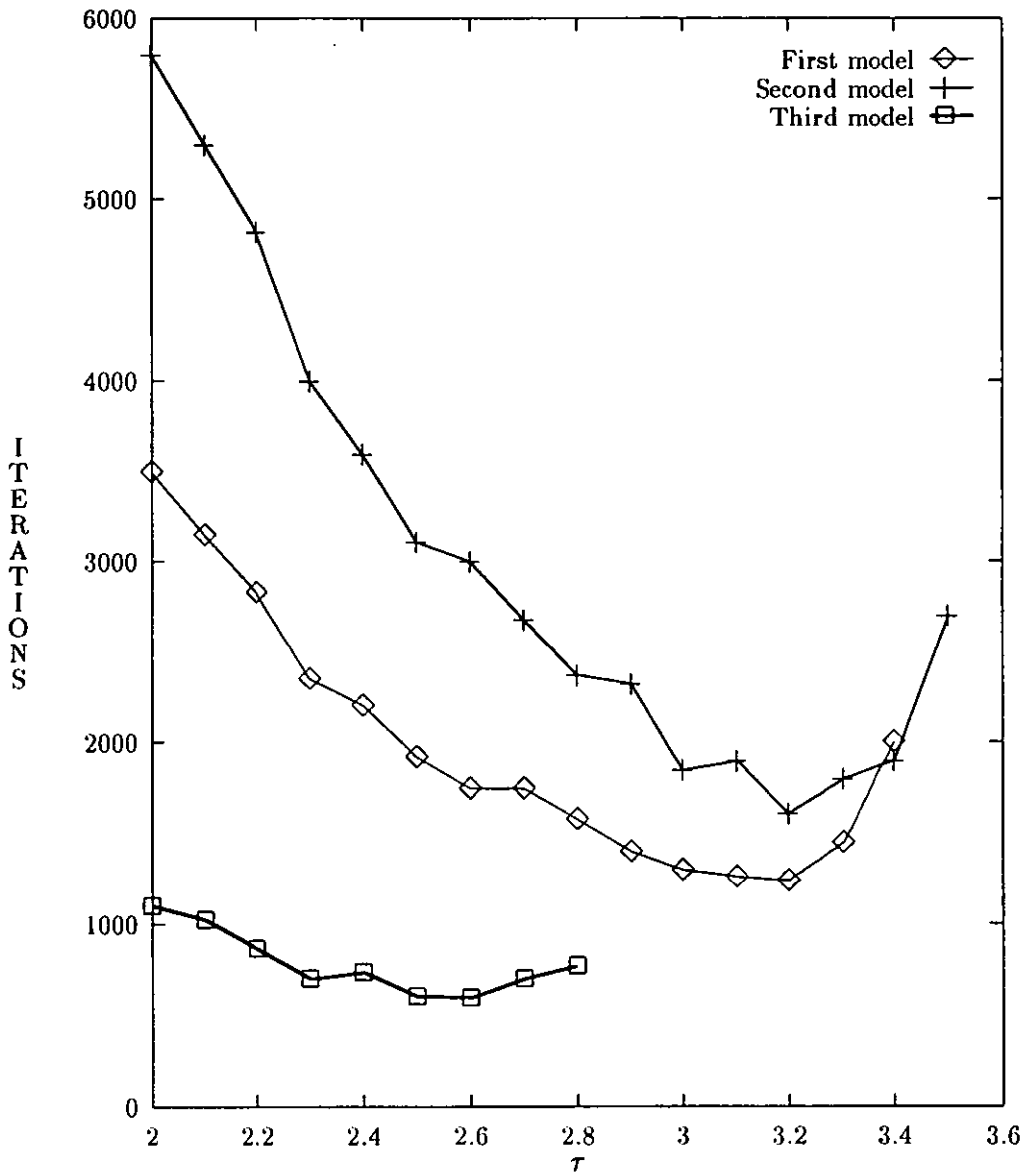


Figure 6.3: Convergence time versus cost function power τ is plotted for all three echo path models. Far-end signal is at -25 dB and convergence was achieved at -45 dB as compared with that of the near-end signal.

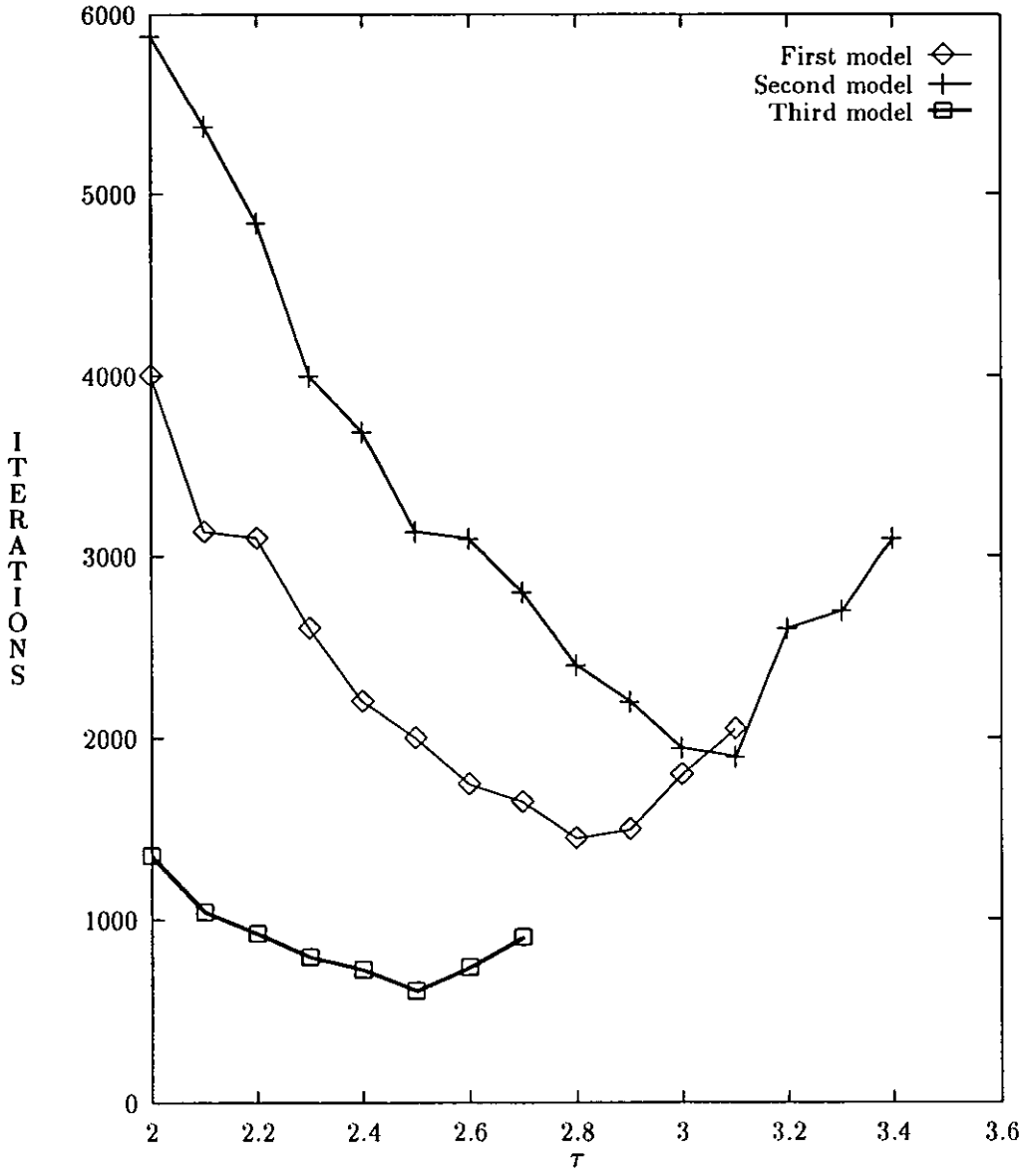


Figure 6.4: Convergence time versus cost function power τ is plotted for all three echo path models. Far-end signal is at -30 dB and convergence was achieved at -50 dB as compared with that of the near-end signal.

The effects of the characteristics of the echo path models are evident from these figures. The type of echo path model determines the relative length, percentage reduction in convergence time, etc. All three echo path models (see Figures 4.1–4.3) used here have different areas under the curves of their respective transfer functions (refer to Equations (5.1) and (5.2). The echo path models with small transfer function area result in relatively large decreases in convergence time and vice versa. e.g. in Figure 6.1, convergence time reduces by 74% for the second echo path model, whereas it reduces by 47% for the third echo path model. Reductions in convergence time are computed by comparing the minimum convergence time with that of the mean square convergence one. Mean square convergence time for the third echo path model is already better than the first and the second ones. Therefore, comparatively less reduction in convergence time may be achieved for an echo path model, which already has comparatively better convergence time for the mean square function, and vice versa.

The improvements in convergence time are greater for models with relatively small transfer function areas (refer to Equations (5.1) and (5.2)) (e.g. the second echo path model), and vice versa. A smaller transfer function area means a higher value of the maximum τ for the minimum possible convergence, and vice versa. Nonetheless all the three echo path models exhibit a general trend of decrease in convergence time for all four far-end signal attenuation levels.

The above mentioned results can be presented in a different way to extract different information. Instead of combining curves belonging to the same far-end signal level and to the different echo path models in a single diagram, we may combine curves belonging to the same echo path model and to the different levels of the far-end signal in a single diagram. Figures 5.5–5.7 show the four cases of

the far-end signal levels for particular echo path models, in single diagrams.

The curves in these figures almost overlap each other. We can safely conclude that the variations in the attenuation level of the far-end signal affect the length, etc. of the curves. Reductions in convergence time remain nearly constant for all four far-end signal power attenuation levels for a particular echo path model. It can be concluded from this observation that the reduction in convergence time is almost independent of the variations in the power attenuation level of the far-end signal. Figures 5.5–5.7 also indicate that the maximum value of τ, for the lowest convergence rate, is higher, for the less attenuated signals, and vice versa.

Although the curves overlap with each other, generally each one has a slightly different convergence time for the mean square function. These can be observed as the starting points of the curves in Figures 5.5–5.7. The starting points are higher for more attenuated far-end signals, and vice versa.

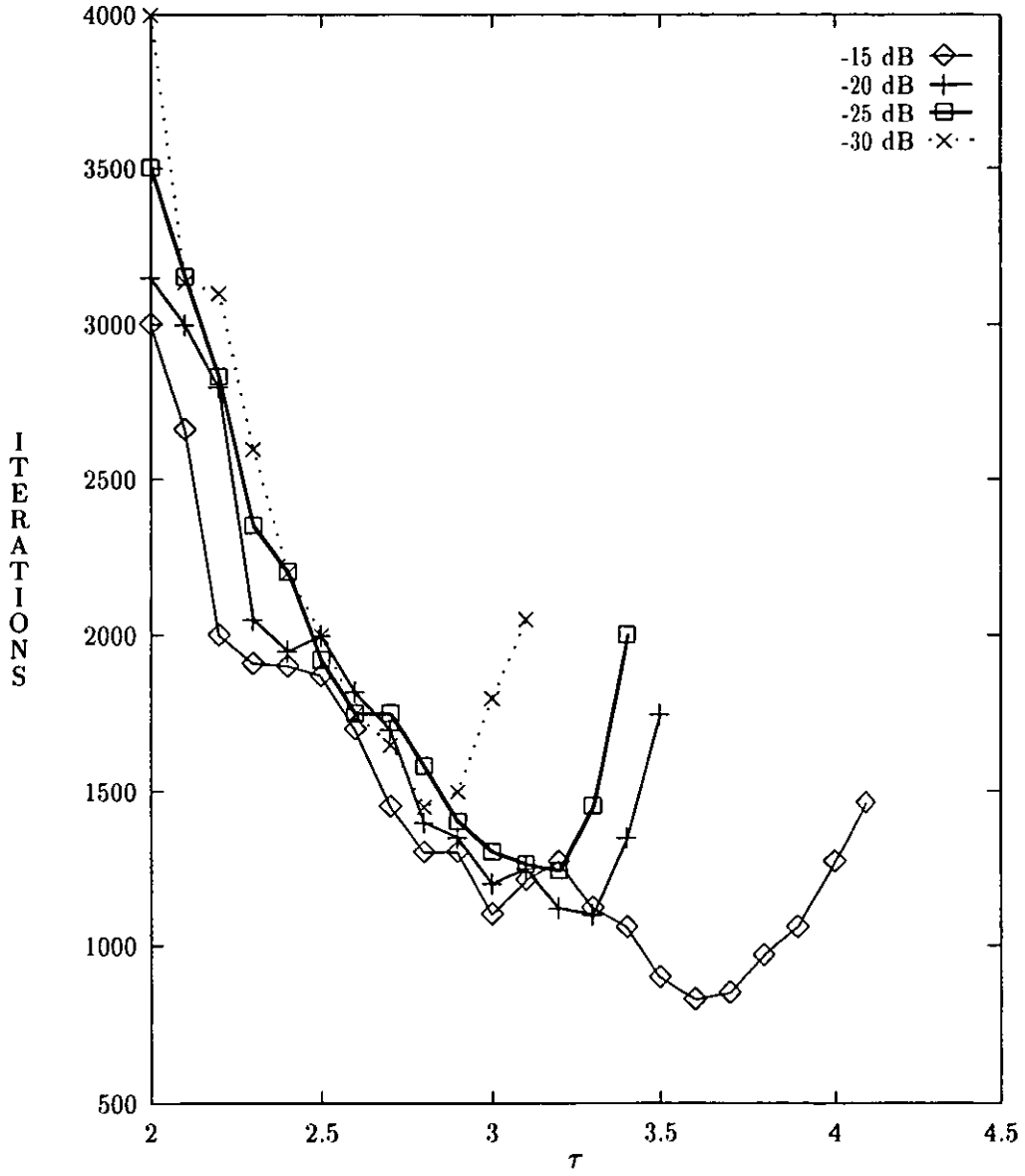


Figure 6.5: Convergence time versus cost function power τ is plotted for the first echo path model and all four far-end signal levels of -15, -20, -25 and -30 dB.

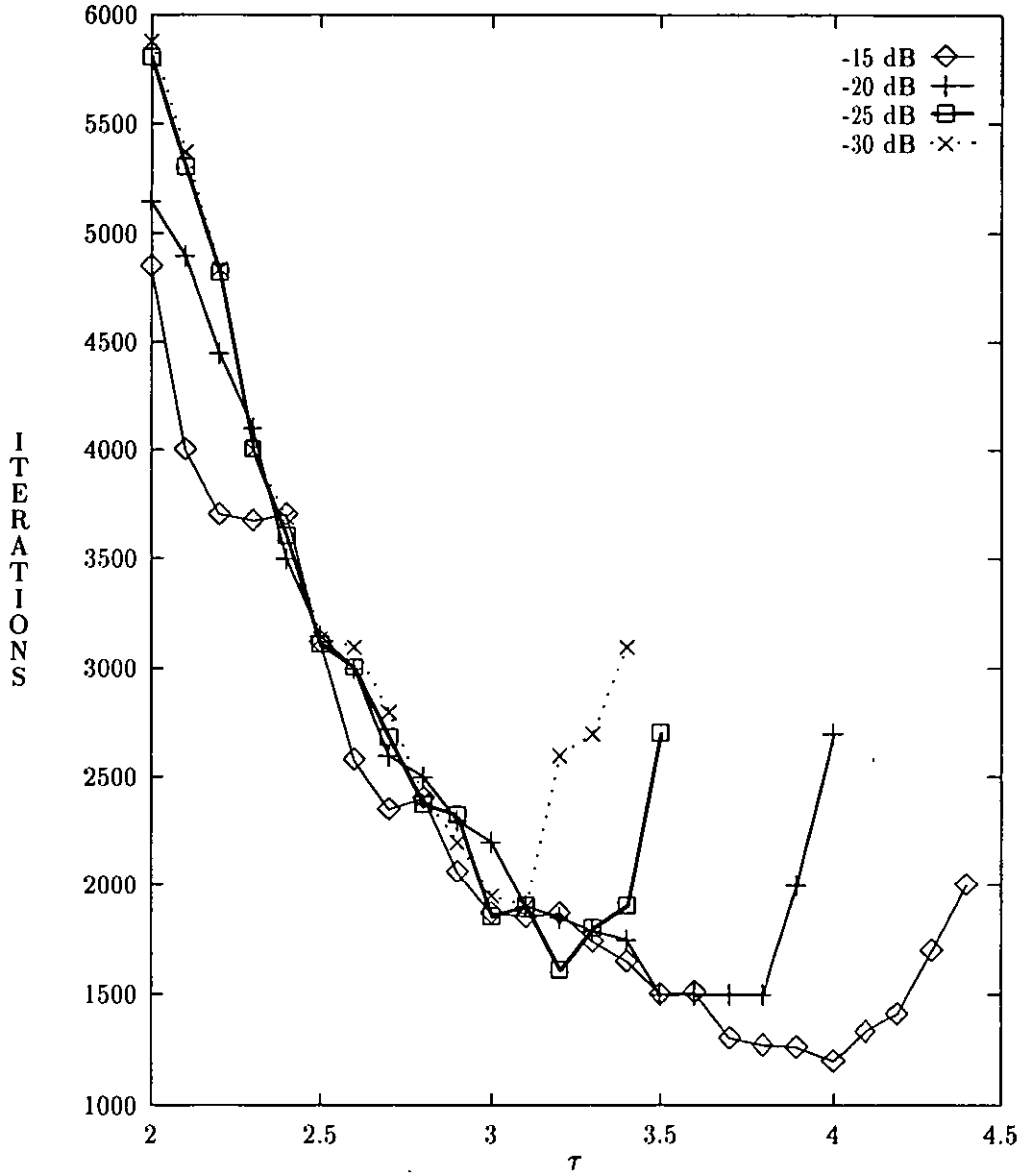


Figure 6.6: Convergence time versus cost function power τ is plotted for the second echo path model and all four far-end signal levels of -15, -20, -25 and -30 dB.

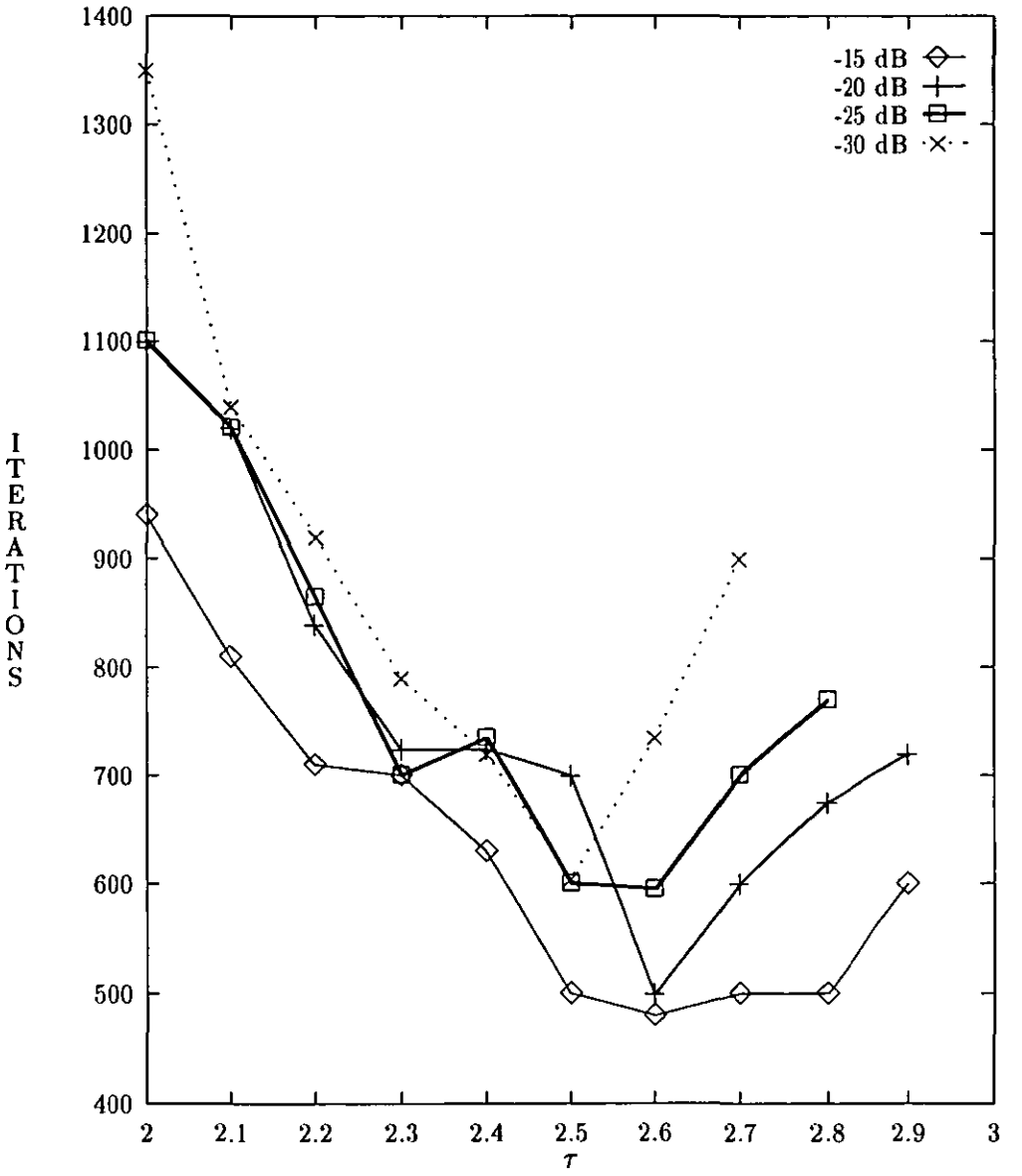


Figure 6.7: Convergence time versus cost function power τ is plotted for the third echo path model and all four far-end signal levels of -15, -20, -25 and -30 dB.

6.3 Stretching tolerance by switching to lower gradient during convergence

A somewhat detailed discussion about switching from higher to lower gradient and vice versa during convergence is given in Section 5.5. The same reasoning can well be applied here on four level data. The required effect is the extension in the tolerance range in each of the individual curves described in the previous section. The simulation results presented in this section are obtained by performing computer simulations in a similar way to that of Section 6.2. Most of the parameters remain the same. i.e. the signal sequences remain pseudo random quaternary from the set {-3, -1, +1, +3}. The same four attenuation levels of -15, -20, -25 and -30 dB are used for the far-end signal. This results in four convergence levels of -35, -40, -45 and -50 dB respectively. Optimum step size μ was empirically determined for each individual case. The only difference is that in the present case the cost function power τ was switched back and forth along with the corresponding step size μ, during the course of the convergence. This was done according to the criteria set in Section 5.5. The rest of the procedure remains the same as previously described in Section 6.2. Different power levels used in simulations along with the maximum values of cost function power τ attained for the fastest convergence, are summarised in Table 6.2.

Figures 6.8–6.11 show the results for various far-end signal levels. Each of the figures contain the results of all three echo path models for a particular far-end signal level and the corresponding convergence level. By inspecting the convergence time curves in these four figures we observe that some of the features are similar to those of the Figures 6.1–6.4. Convergence time decreases as we

near-end signal level dBs	far-end signal level dBs	conver -gence level dBs	(far-end signal level) -(convergence level) dBs	max. value of τ attained		
				1st model	2nd model	3rd model
0	-15	-35	20	6.0	6.6	4.5
0	-20	-40	20	4.7	5.0	3.8
0	-25	-45	20	3.8	3.9	3.3
0	-30	-50	20	3.3	3.6	3.0

Table 6.2: Various combinations of power levels and the maximum values of cost function power τ attained for the fastest convergence with each combination. Data streams were quaternary. Switching of gradient was performed during convergence.

increase cost function power τ. Initially the decrease in convergence time is fast. Gradually it becomes slower. Then no change occurs in convergence time with the increase in τ. These later parts of the curves provide us with the tolerance. Convergence time tolerates the changes in τ for a certain range of τ. A bathtub like feature emerges when convergence time once again starts increasing with the increase in τ. The present form of the algorithm (i.e. with switching to lower gradient during convergence as per requirement) results in an extended range of τ (Figures 6.8–6.11), as compared with that of the non-switched case (Figures 6.1–6.4).

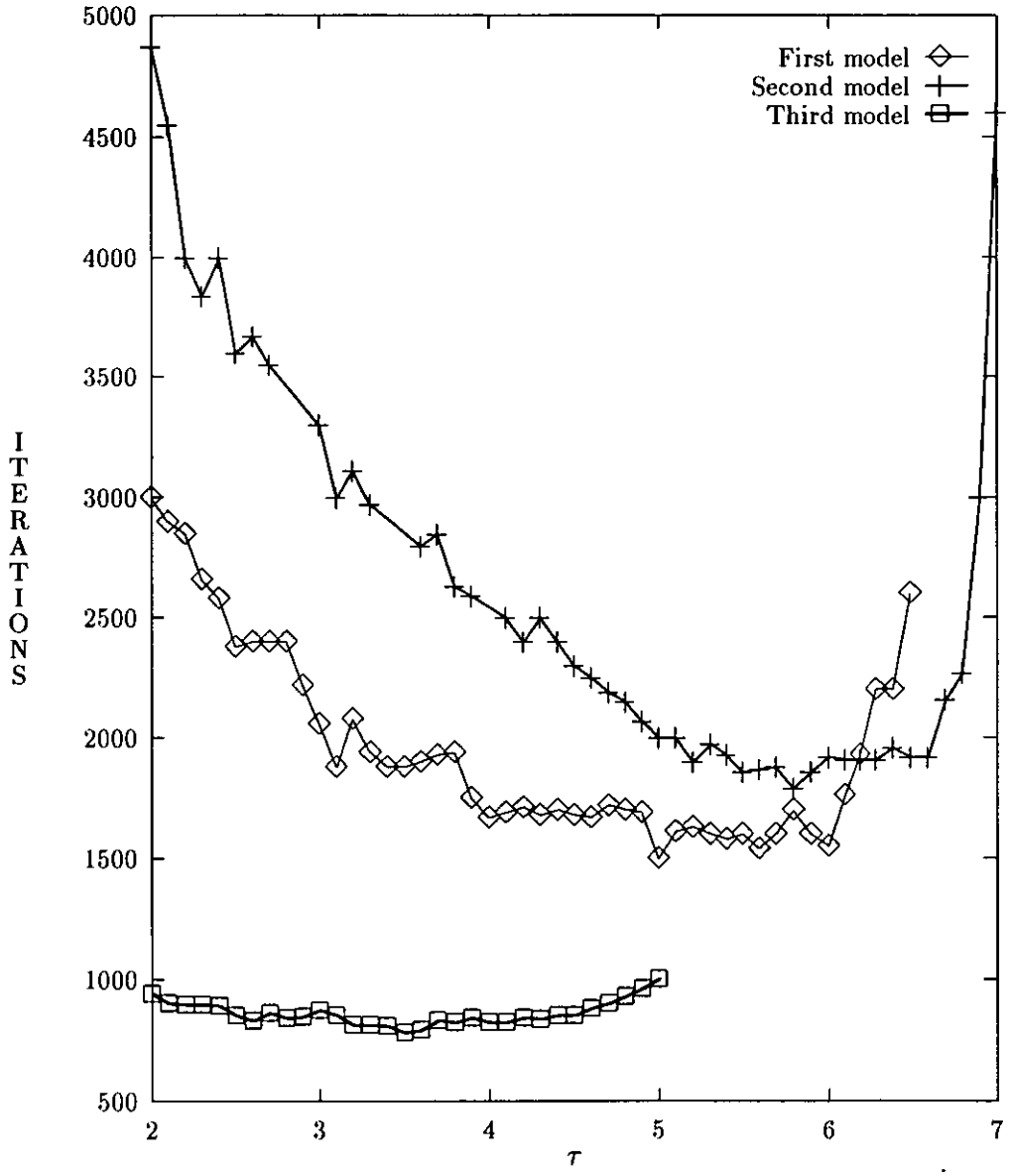


Figure 6.8: Convergence time versus cost function power τ is plotted for all three echo path models. Far-end signal is at -15 dB and convergence was achieved at -35 dB as compared with that of the near-end signal. Gradient was switched back and forth during convergence.

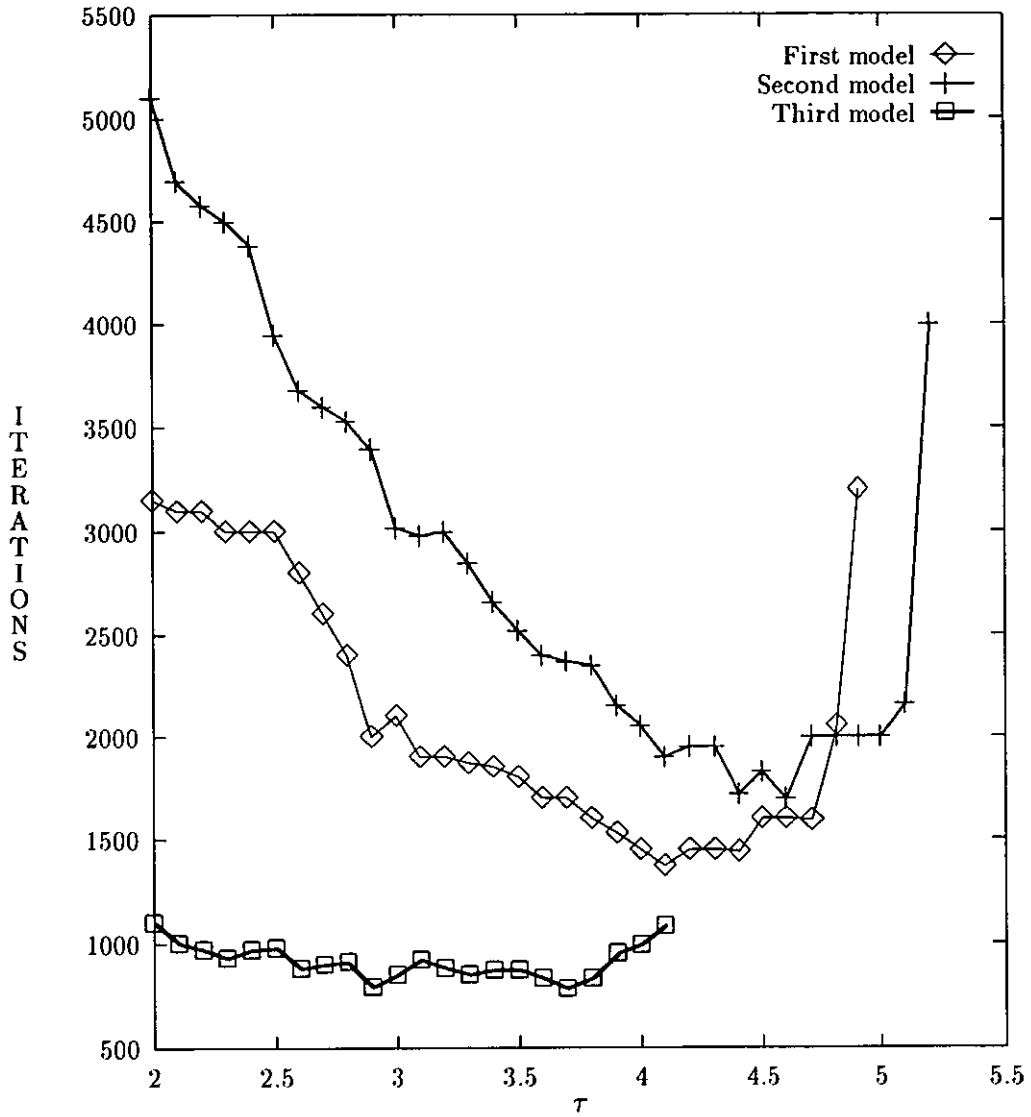


Figure 6.9: Convergence time versus cost function power τ is plotted for all three echo path models. Far-end signal is at -20 dB and convergence was achieved at -40 dB as compared with that of the near-end signal. Gradient was switched back and forth during convergence.

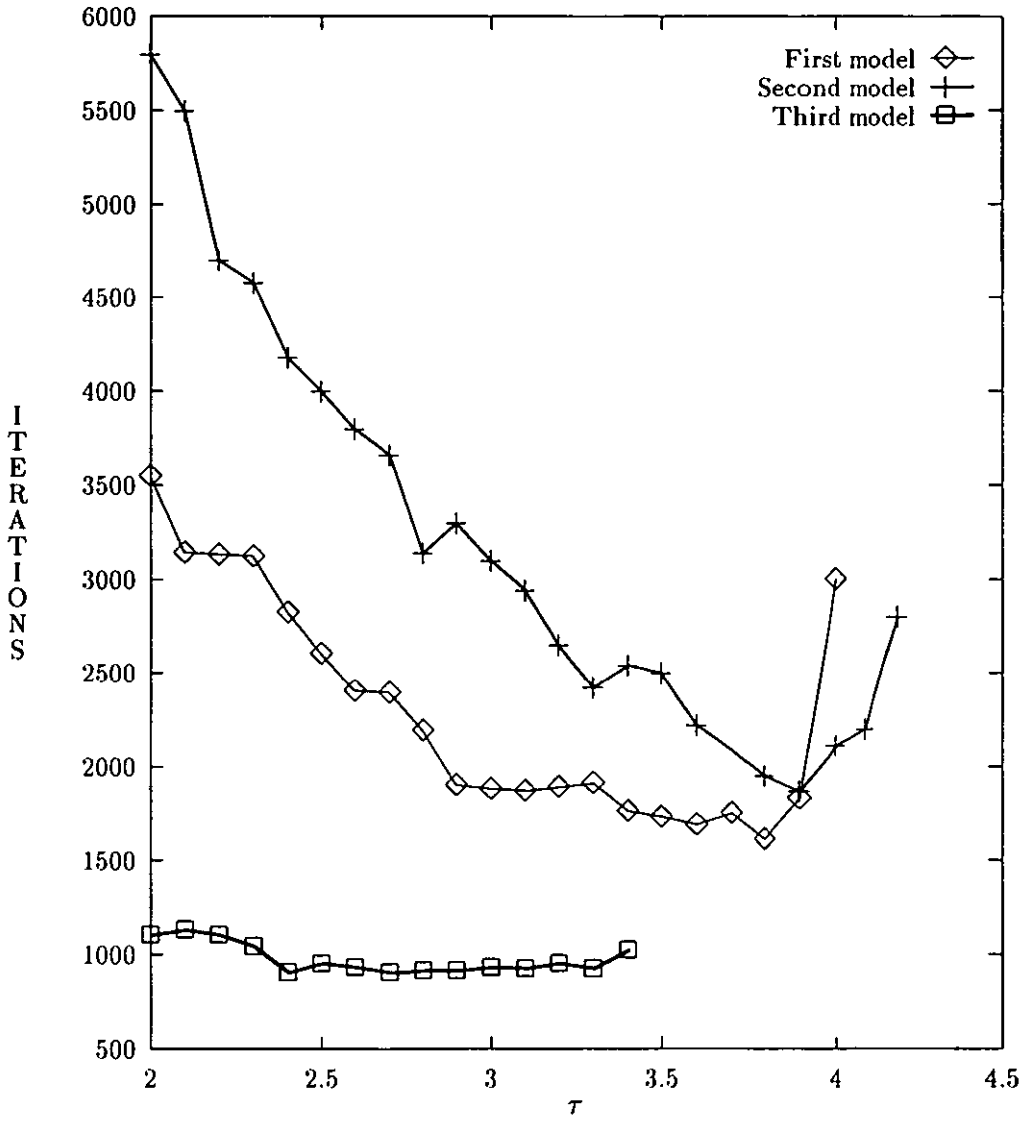


Figure 6.10: Convergence time versus cost function power τ is plotted for all three echo path models. Far-end signal is at -25 dB and convergence was achieved at -45 dB as compared with that of the near-end signal. Gradient was switched back and forth during convergence.

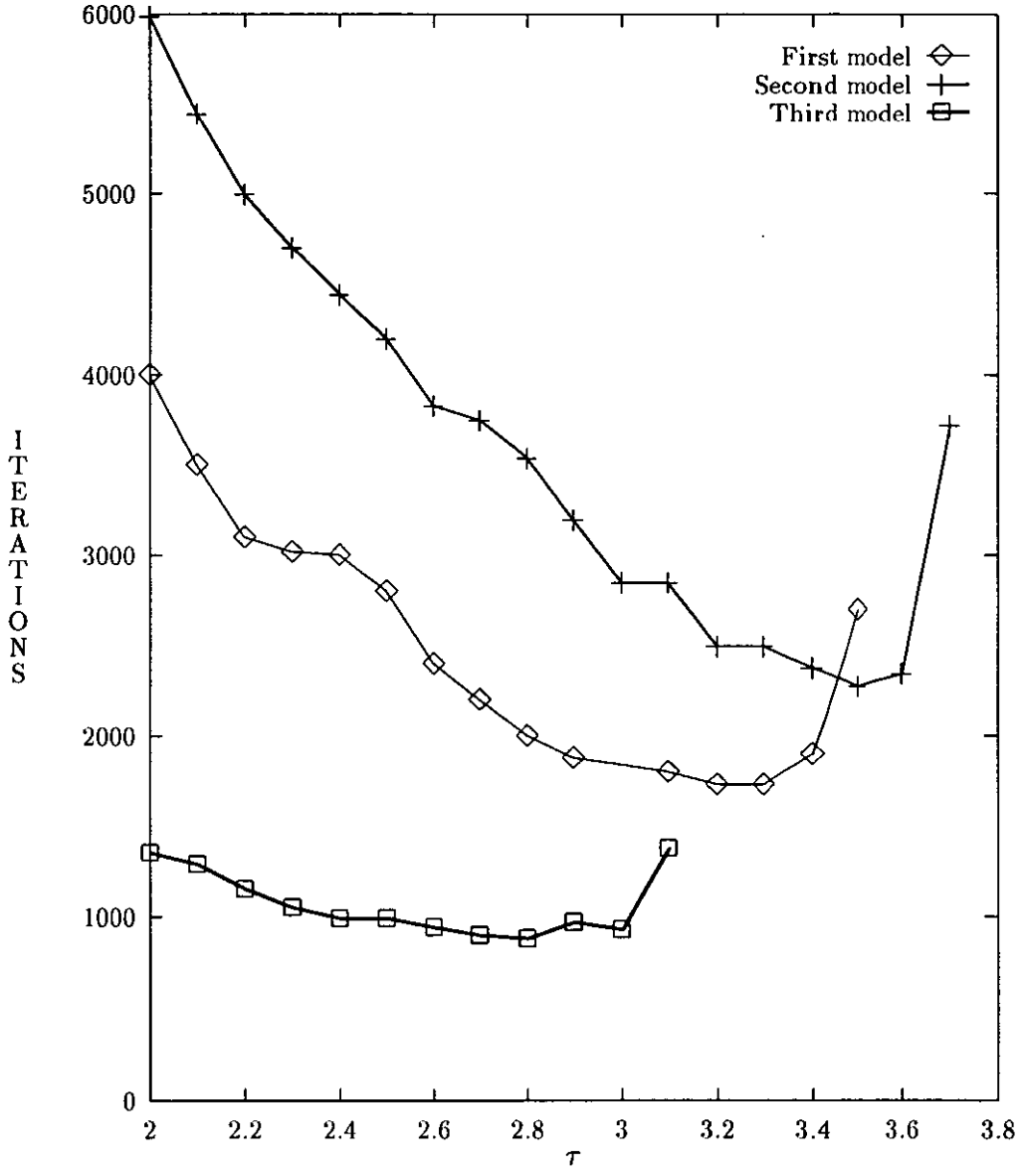


Figure 6.11: Convergence time versus cost function power τ is plotted for all three echo path models. Far-end signal is at -30 dB and convergence was achieved at -50 dB as compared with that of the near-end signal. Gradient was switched back and forth during convergence.

Some features of the curves are similar to those of the curves for the non-switched case. e.g. initial points of the curves on the y-axis (mean square convergence time), are higher, for the small area transfer function of the echo path models (refer to Equations (5.1) and (5.2)), and vice versa. More percentage reduction in convergence time for small area transfer function echo paths, and vice versa. The lengths of the curves are also dependent upon the characteristics of the echo path model. The smaller the area of the transfer function of the echo path model, the greater the lengths of these curves will be. In other words, we will achieve a higher value of τ for the fastest convergence in each individual case. The most important feature of our interest after the reduction in convergence time, is, the extended tolerance of the convergence time to the variations in τ for a certain range of τ.

For comparison of the effects of various far-end signal levels for a particular echo path model, the same convergence time curves are presented in a different way. i.e. by collection of all four cases of far-end signal levels in a single diagram. One diagram (Figures 6.12–6.14) for each of the echo path models. These are the same curves as were in Figures 6.8–6.11, but are combined differently to extract different information.

A look at these figures shows that the curves in each of the figures overlap each other before jumping up, as in a similar case for binary data sequences. The starting points on the y-axis (convergence time in terms of the number of iterations) are generally higher for more attenuation in the far-end signal and vice versa. These are not so strictly followed rules because of the close proximity of the curves with each other in the presence of observational noise. Attenuation in the far-end signal affects the lengths of the curves. Less attenuation, longer tails, larger regions of convergence time tolerance to τ, and vice versa.

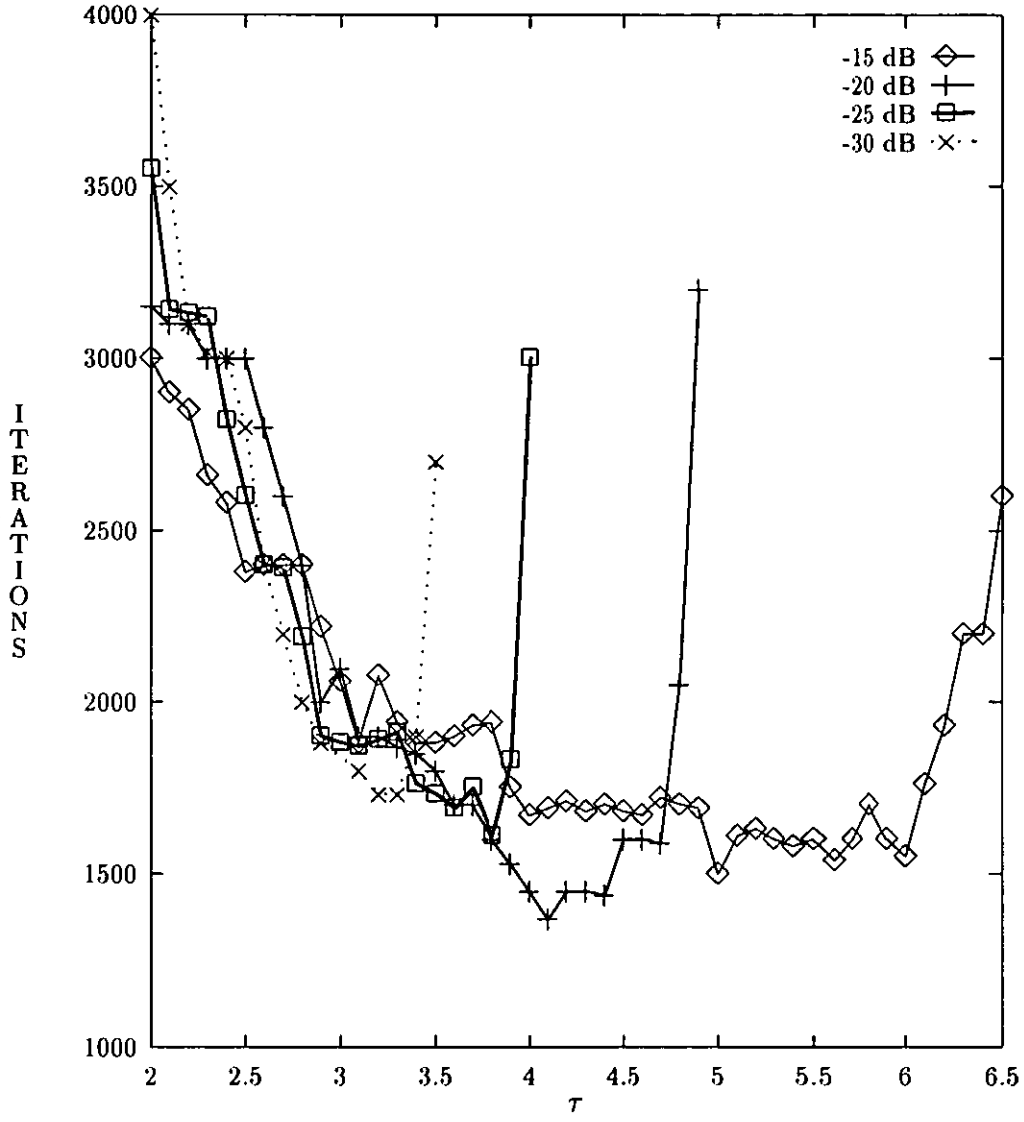


Figure 6.12: Convergence time versus cost function power τ is plotted for the first echo path model and all four far-end signal levels of -15, -20, -25 and -30 dB. Gradient was switched back and forth during convergence.

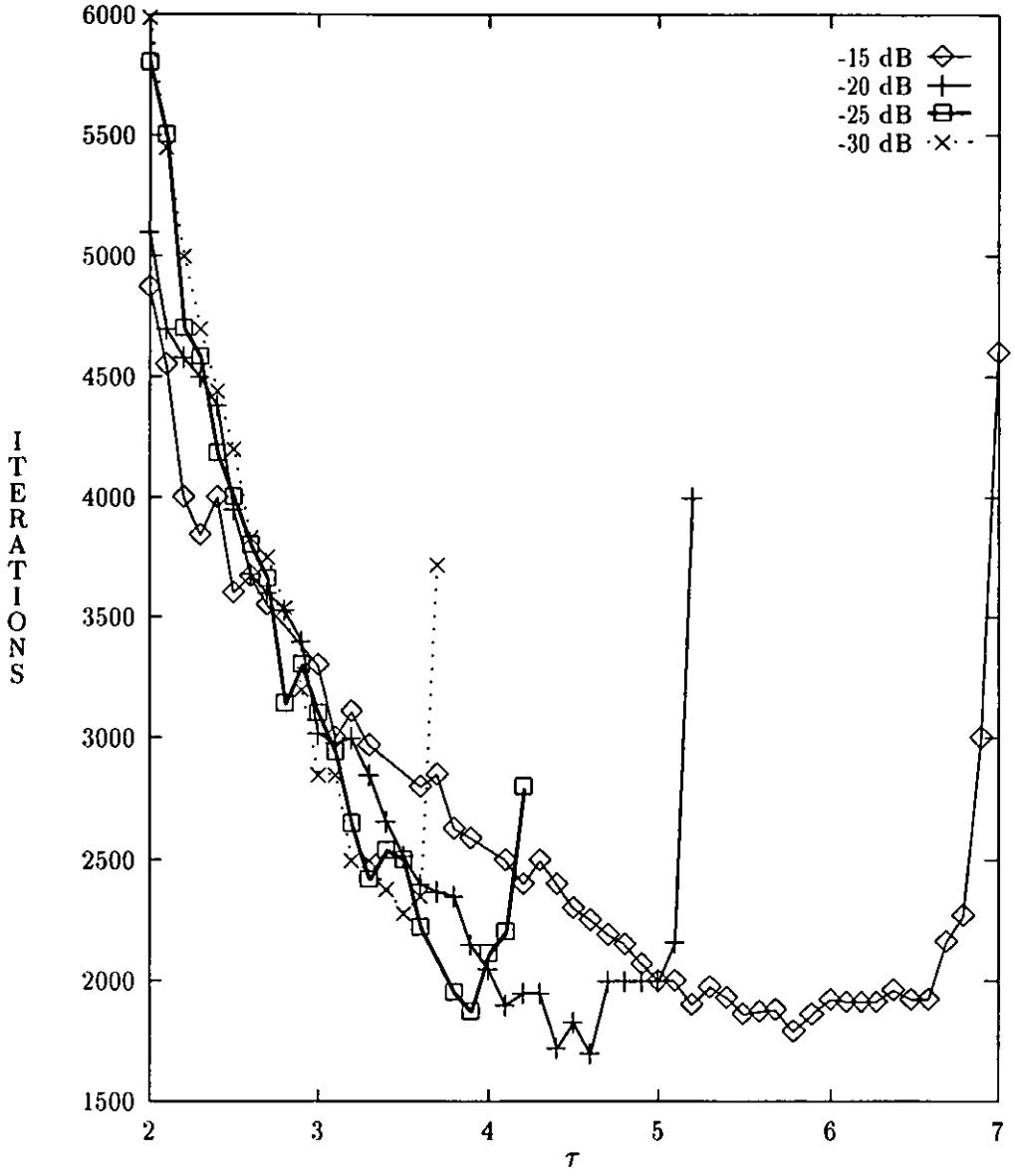


Figure 6.13: Convergence time versus cost function power τ is plotted for the second echo path model and all four far-end signal levels of -15, -20, -25 and -30 dB. Gradient was switched back and forth during convergence.

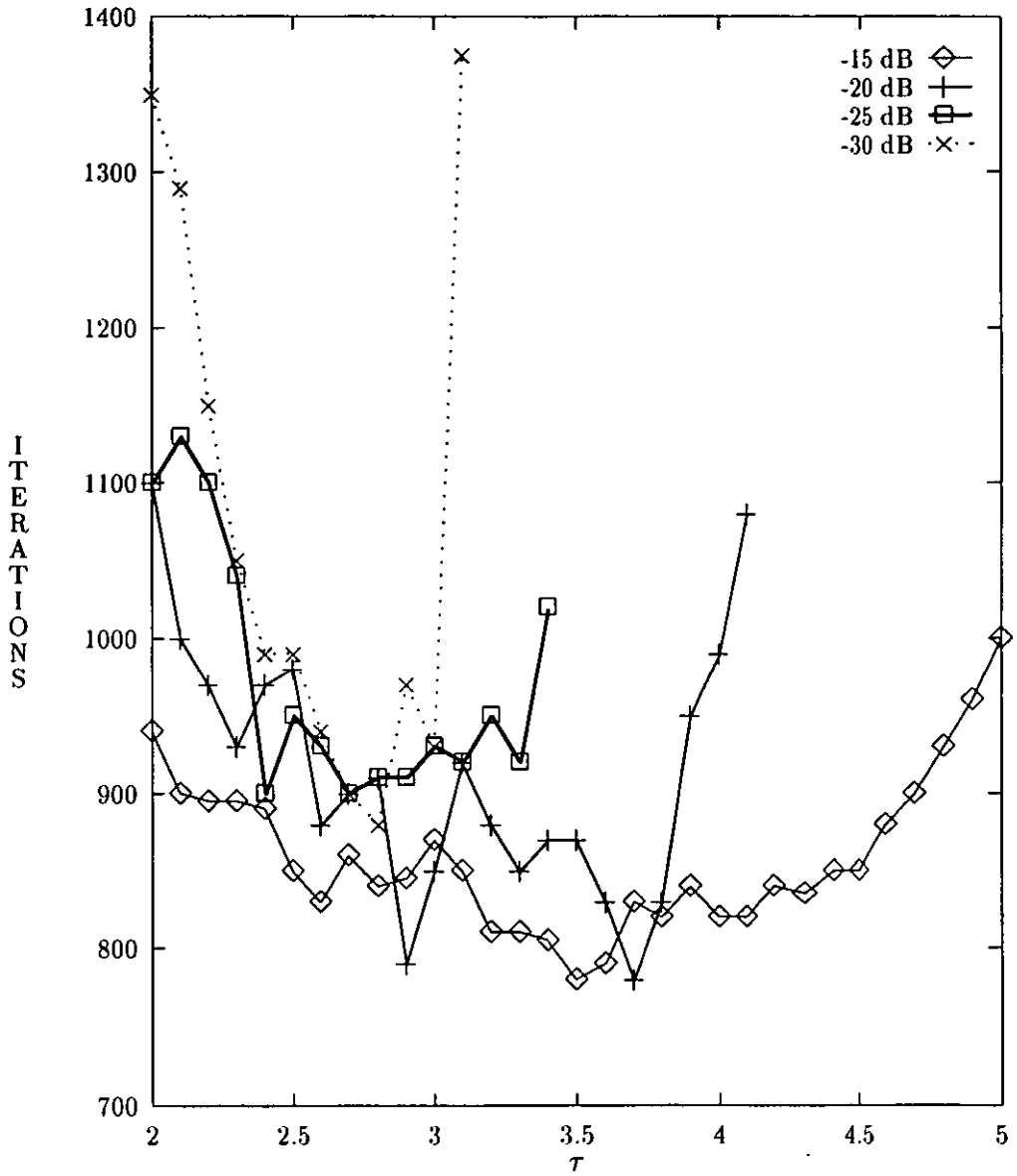


Figure 6.14: Convergence time versus cost function power τ is plotted for the third echo path model and all four far-end signal levels of -15, -20, -25 and -30 dB. Gradient was switched back and forth during convergence.

6.4 Switching behaviour

The switching behaviour of nonquadratic adaptation algorithms in the case of the four level data sequences can be obtained in a similar way as in the case of binary data sequences. The first echo path model with the far-end signal at -15 dB was chosen for simulations. Simulations were performed in a similar way as described in Section 6.3. Convergence is achieved at -35 dB. No averaging is done. After compiling the results of the simulations it is observed that a relatively small number of switchings take place. However, in absolute terms they are nearly double to the ones for binary data sequences. i.e. the number of switchings remain below or around 400, as compared with that of the figure of 200, in the previous chapter. The number of switchings versus τ is plotted in Figure 6.15

Figure 6.15 shows that the number of switchings decrease initially as τ is increased. They increase once again near the end of the curve with the increase of τ. The number of switchings is higher for $\tau = 2.1$. It keeps on decreasing as τ is increased for some time and then fluctuates in the remaining part of the curve. These fluctuations are due to the four level signal sequences. The Figure 6.15 shows that the reduction in the number of switchings is less for the present case as compared with the ones with binary data sequences shown in Figure 5.28. A selection of the switching sequences for the four different values of τ, i.e. $\tau = 2.1$, 3.7, 5.2 and 6.7 are shown in Figures 6.16–6.19. From these figures we can readily observe the decrease in the number of switchings, with the increase in τ and then increase once again at the end.

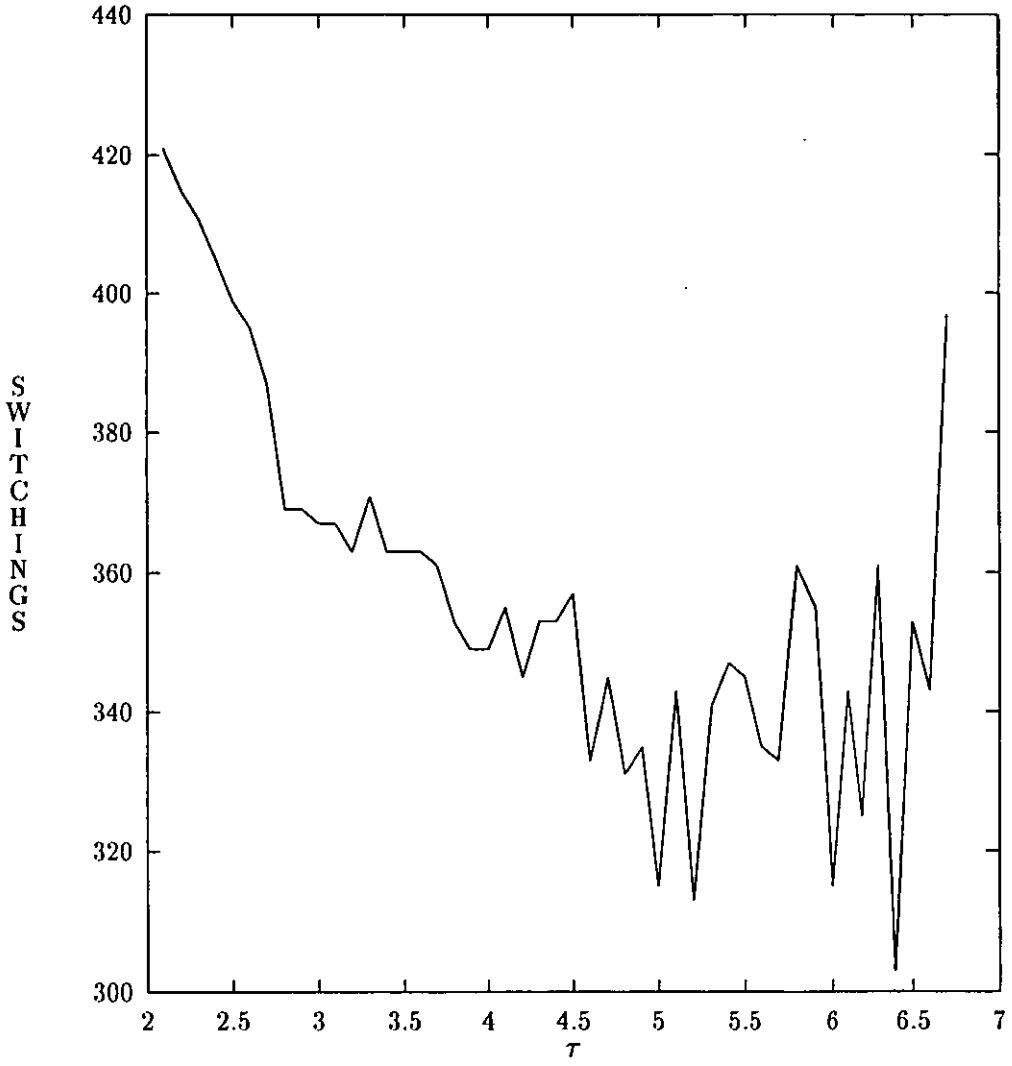


Figure 6.15: Number of switchings versus cost function power τ. For the case of quaternary signal sequences when far-end signal is at -15 dB and convergence is achieved at -35 dB for the first echo path model.

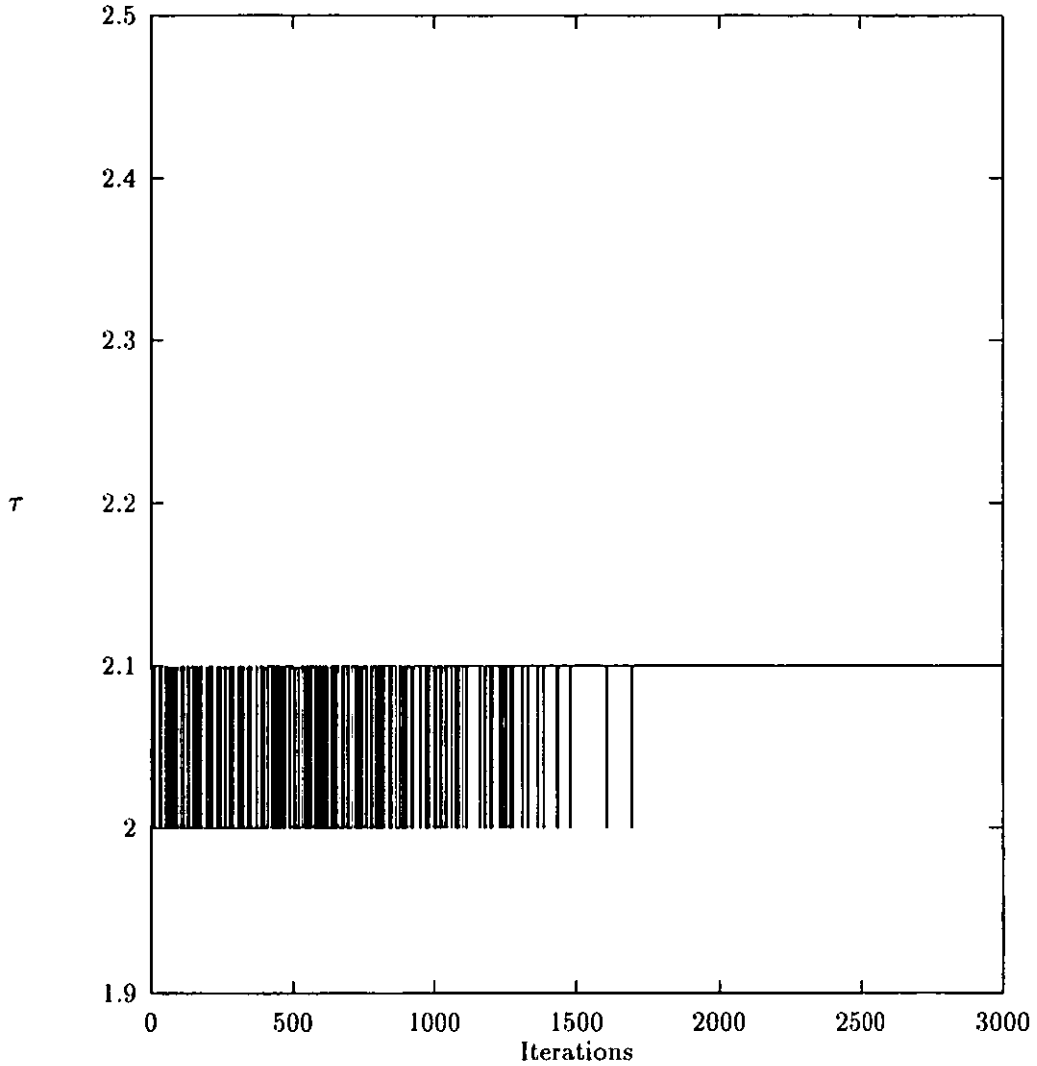


Figure 6.16: Switching sequence for $\tau = 2.1$. Simulations are performed with quaternary data sequences when far-end signal is at -15 dB and convergence is achieved at -35 dB for the first echo path model.

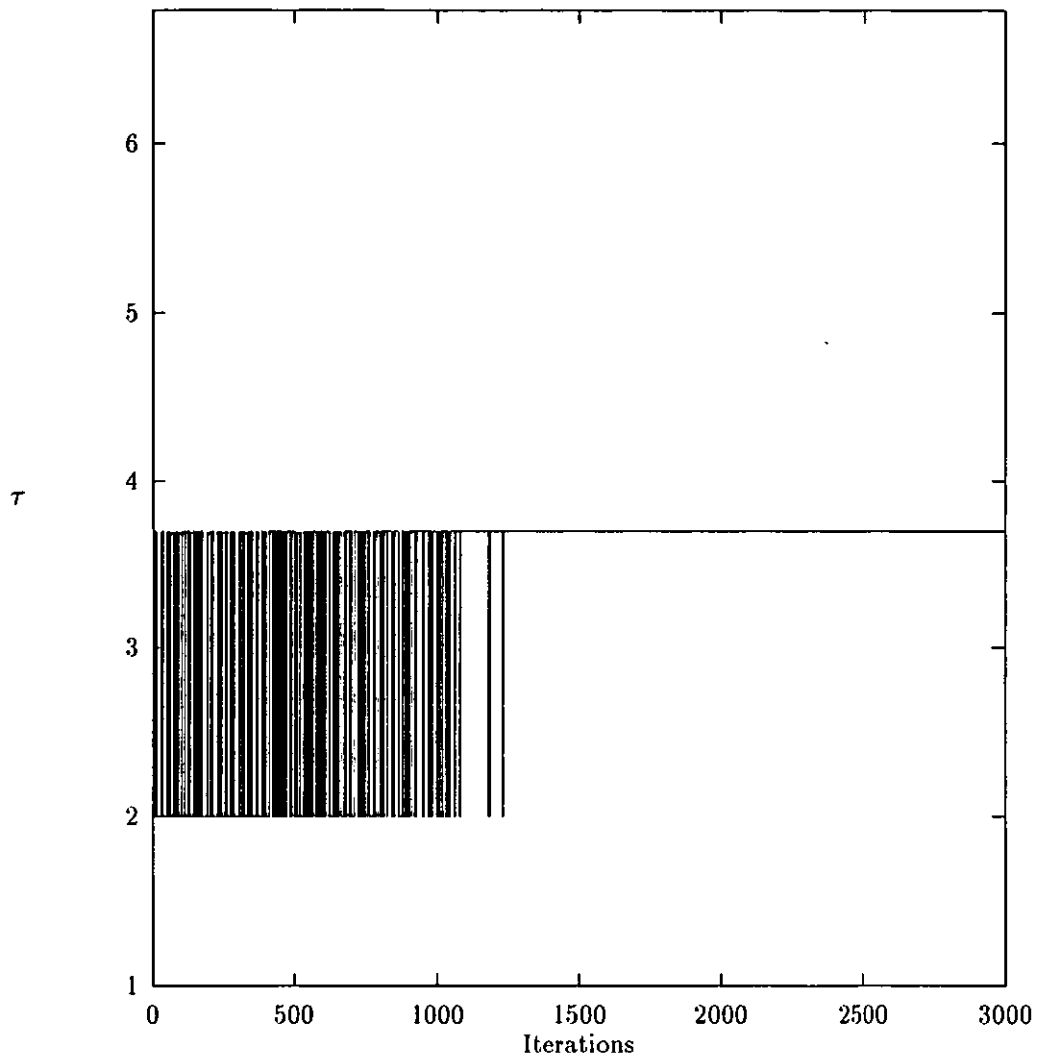


Figure 6.17: Switching sequence for $\tau = 3.7$. Simulations are performed with quaternary data sequences when far-end signal is at -15 dB and convergence is achieved at -35 dB for the first echo path model.

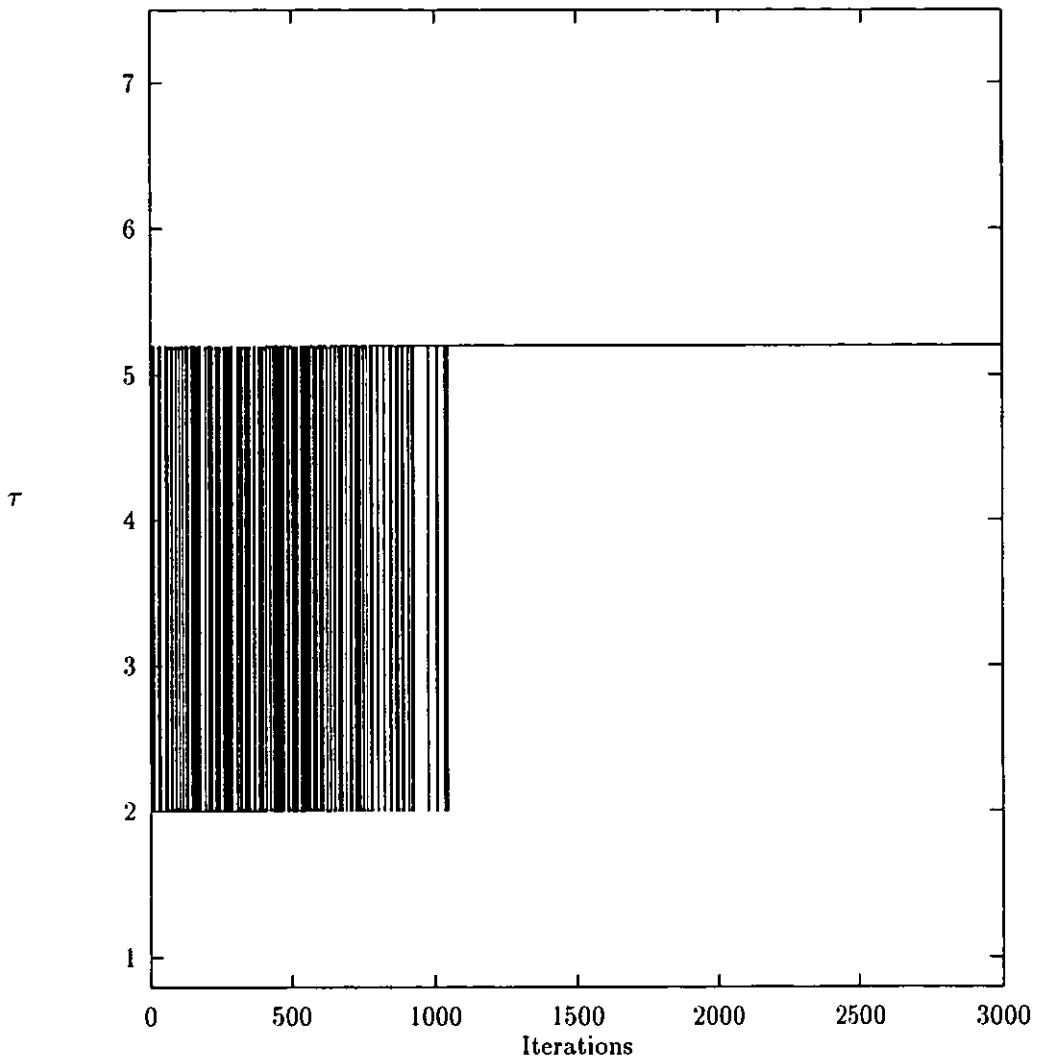


Figure 6.18: Switching sequence for $\tau = 5.2$. Simulations are performed with quaternary data sequences when far-end signal is at -15 dB and convergence is achieved at -35 dB for the first echo path model.

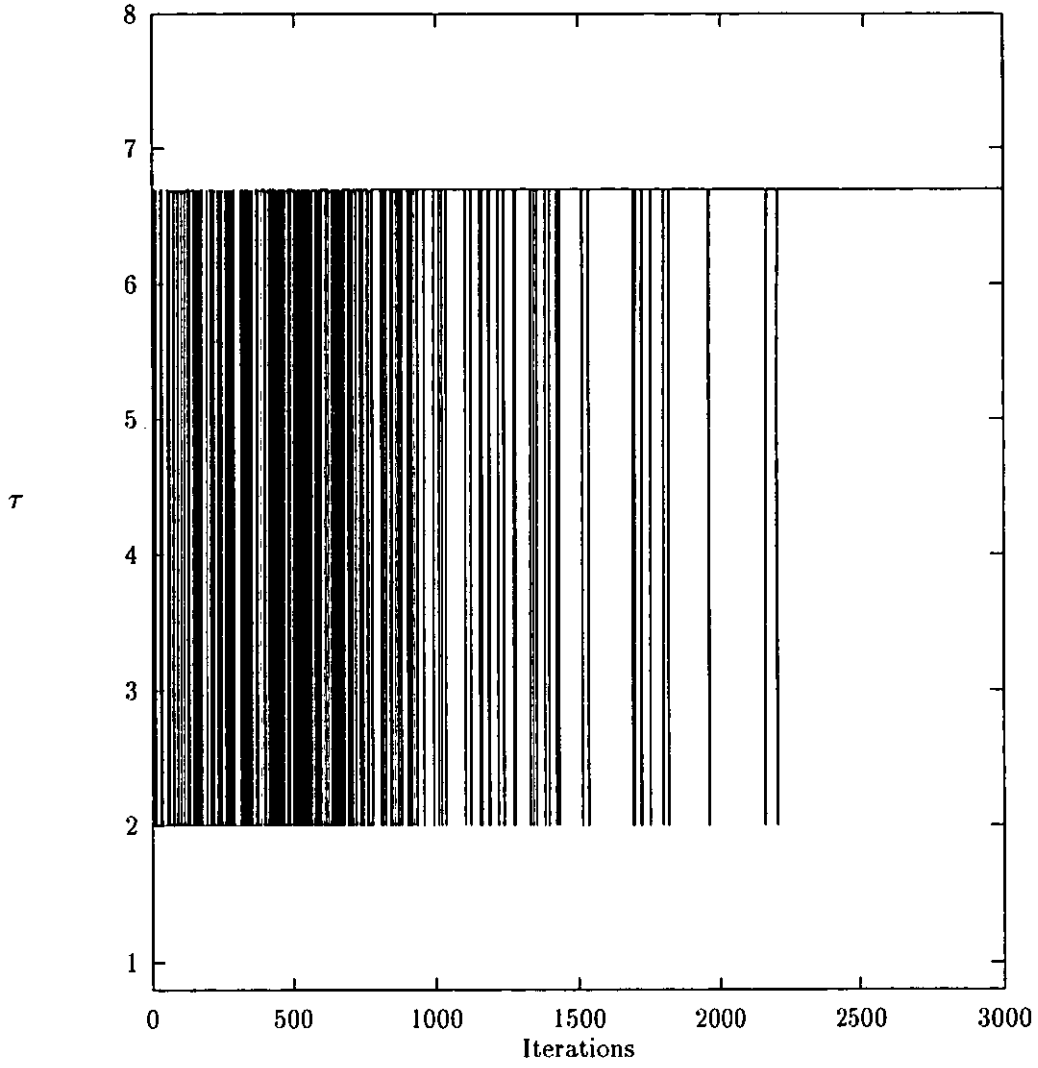


Figure 6.19: Switching sequence for $\tau = 6.7$. Simulations are performed with quaternary data sequences when far-end signal is at -15 dB and convergence is achieved at -35 dB for the first echo path model.

6.5 Conclusions

In the case of the four level data sequences, the non-square cost functions reduce the convergence time considerably as in the case of binary data sequences.

Reductions in convergence time mainly depends upon the characteristics of the echo path. An echo path model with a small area under the transfer function curve (refer to Equations (5.1) and (5.2)) results in relatively large reductions in convergence time and vice versa. This effect is similar to that for the case of binary data. We obtain a comparatively larger range of low sensitive region to τ for an echo path model with a relatively smaller area under the transfer function curve.

Convergence time reductions are nearly independent of the level of the far-end signals. We obtain, in general, a larger range of low sensitivity to τ for a less attenuated far-end signal and vice versa.

As for the case of binary signal sequences, the gradient along with its corresponding optimum step size μ can be switched back and forth, for the four level signal sequences as well, according to the requirements during the process of convergence. The switching results in an extended range of low sensitivity to τ. A similar bathtub like feature emerges in the convergence time curves of the switched simulations, i.e. at the tail of the curves, convergence time increases once again with the increase in cost function power τ. As the extension of low sensitivity to τ is concerned, the switched case responds more to the variations in the power attenuation level of the far-end signal sequences than to the variations in the characteristics of the echo path.

Chapter 7

Conclusions

7.1 Summary of achievements

New adaptation algorithms based on cost function $E[|e_k|^\tau]$, where τ is a rational number greater than or equal to 2.0, were derived and analytically verified for convergence. Stochastic gradient techniques were used for development of these adaptation algorithms. Although mean square algorithms have already been proved optimum for Gaussian data in contemporary literature, the idea behind this study was to develop algorithms for non-Gaussian data, that are relatively faster in convergence as compared with the mean square ones. As faster algorithms are required for applications in digital data communications systems, the proposed adaptation algorithms were tested in digital data echo cancellation. Simulation results have shown that the new algorithms are superior to the mean square ones with respect to substantial improvements in convergence time. A great number of simulations were performed in the presence of far-end signals of various power attenuation levels, with different echo path models, with two

level as well as four level digital data and dispersion in the far-end signal in addition to the attenuation. Results of all these experiments have confirmed the substantial improvements in convergence time comparative to mean square algorithms.

Simulations with binary data

Four power attenuation levels of far-end signal (-15, -20, -25 and -30 dB) and three echo path models were used for computer simulations incorporating two level digital data streams. Simulation results have been presented in Chapter 5. From these results we can conclude that the adaptation algorithms using nonquadratic cost function powers reduce the convergence time considerably as compared with that of the LMS algorithm.

Decrease in convergence time largely depends upon the area under the transfer function curve of the echo path model (refer to Equations (5.1) and (5.2)). Lesser area transfer functions tend to achieve relatively higher reductions in convergence time and vice versa. Reductions in convergence time are almost independent of the attenuation level of the far-end signal. Relative power of the far-end signal has a larger effect on the tails of the convergence time curves. Less attenuation (more power) results in longer tails, i.e. a higher value of τ for the fastest convergence, larger region of tolerance of convergence time to the variations in τ and vice versa.

We also experimented with changing cost function power τ during convergence, according to certain criteria described in Chapter 5. From there we can conclude that the gradient can be switched back and forth from a higher to a lower value and vice versa, during the convergence process. The result of this study is the achievement of the extended portions of the convergence time

curves with greater tolerance and lower sensitivity to the variations in τ. That means convergence rates do not change appreciably by changing cost function power τ in that region. This effect gives a better choice of τ to the designer while designing real systems. At the end of the convergence time curves, convergence time starts increasing once again with the emergence of a bathtub like feature. It was observed that switchings occur only during initial part of the convergence process. The number of switchings tends to decrease with the increase in τ. After reaching a certain minimum, the number of switchings start increasing once again and that happens near the tail of the convergence time curve.

Dispersion was also added to the far-end signal in addition to the attenuation for one case. Simulation results show that the basic argument of reduction in convergence time or increase in convergence rate remains valid for this case.

Simulations with quaternary data

The same four far-end power attenuation levels (i.e. -15, -20, -25 and -30 dB) and the same three echo path models were used for computer simulations incorporating four level data streams, as in the case of binary signals. It is concluded that in the case of four level data sequences as well, the nonquadratic cost function increases the convergence rate considerably.

Increase in convergence rate depends upon the characteristics of the echo path. An echo path with smaller area under the transfer function curve (refer to Equations (5.1) and (5.2)) tends to produce higher convergence rates and vice versa. This observation is similar to that with the binary data one. We obtain comparatively larger range of low sensitive region to τ for an echo path model with relatively smaller area under the transfer function curve.

Increase in convergence rate is almost independent of the attenuation level of the far-end signal. We obtain, in general, a larger range of low sensitivity to τ for a less attenuated far-end signal and vice versa.

Cost function power, along with the corresponding optimum step size μ were switched back and forth as per requirements during the convergence, for the four level data sequences as well. The switchings have resulted in an extended range of low sensitivity to τ. A similar bathtub like feature have emerged in the convergence time curves of the switched simulations. i.e. at the tail of the curves, convergence time increases once again with the increase in cost function power τ. Lengths of the convergence time curves, obtained with the switched gradient, are more sensitive to the attenuation level in the far-end signal than to the variations in the characteristics of the echo path.

7.2 Suggestions for future research

Nonquadratic error criteria $|e_k|^\tau$ based on rational numbers (rather than just 2.0 or its integer powers) have huge potential of applications in areas like digital signal processing and control. At present, most of the adaptive signal processing and adaptive control applications revolve around the mean square error criterion for its simplicity and ease in implementation, ease in mathematical analysis and proven convergence properties. However, mean square error criterion is bounded with the Gaussian nature of the data. For non-Gaussian data the superiority of the nonquadratic cost functions over the quadratic ones have been established in this thesis for at least one application of adaptive filtering (i.e. digital data echo cancellation).

One multifacet area for future work could be to look into all kinds of appli-

cations of adaptive filters in adaptive signal processing (e.g. digital data equalisation), and adaptive control, where the data sequences are non-Gaussian.

Nonquadratic cost functions are applied only on stochastic gradient methods for the work presented in this thesis. A future work could be to investigate the application of nonquadratic cost functions with other algorithms, e.g. recursive least squares (RLS), etc.

References

[1] Shah, S. A. H. and Cowan, C. F. N.; "Modified stochastic gradient algorithm using nonquadratic cost functions for data echo cancellation," IEE Proceedings on Vision, Image, and Signal Processing, vol. 142, no. 3, pp. 187–191, June 1995.

[2] Adams, P. F.; "Adaptive filters in Telecommunications" in *Adaptive filters*, by Cowan, C. F. N. and Grant, P. M.; Prentice Hall, Englewood Cliffs, New Jeresy, 1985.

[3] Walach, E. and Widrow, B.; "The Least Mean Fourth (LMF) Adaptive Algorithm and its Family," IEEE Transactions on Information Theory, vol. IT-30, no. 2, pp. 275–283, March 1984.

[4] McCool, J. M. and Widrow, B.; "Principles and applications of adaptive filters: A tutorial review," IEEE International Symposium on circuits and Systems, Houston, vol. 3, pp. 1143–1157, April 1980.

[5] Koford, J. S. and Groner, G. F.; "The use of an adaptive threshold element to design a linear optimal pattern classifier," IEEE Transactions on Information Theory, vol. IT-12, no. 1, pp. 42–50, January 1966.

[6] Krishna, V. H. and Reddy, D. C.; "Design of adaptive filters for a class of non-uniformly sampled signals," Signal Processing, vol. 31, no. 1, pp. 81–89, March 1993.

[7] Grant, P. M.; "Digital signal processing, part 1: digital filters and the DFT," Electronics & Communication Engineering Journal, vol. 5, no. 1, pp. 13–21, February 1993.

[8] Gitlin, R. D., Mazo, J. E. and Taylor, M. G.; "On the design of gradient algorithms for digitally implemented adaptive filters," IEEE Transactions on Circuit Theory, vol. CT-20, no. 2, pp. 125–136, March 1973.

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

[10] Kwan, H. K. and Li, Q. P.; "High-speed realisation of adaptive linear phase FIR digital filters," IEE Proceedings, Part F, vol. 140, no. 1, pp. 48–54, February 1993.

[11] Cowan, C. F. N.; "Performance comparisons of finite linear adaptive filters," IEE Proceedings, Part F, vol. 134, no. 3, pp. 211–216, June 1987.

[12] Honig, M. L. and Messerschmitt, D. G.; *Adaptive filters: structures, algorithms and applications*, Kluwer Academic Publishers, Norwell, Massachusittes, 1984.

[13] Cowan, C. F. N. and Grant, P. M.; *Adaptive filters*, Prentice Hall, Englewood Cliffs, New Jeresy, 1985.

[14] Mulgrew, B. and Cowan, C. F. N.; *Adaptive filters and equalisers*, Kluwer Academic Publishers, Norwell, Massachusettes, 1988.

[15] Widrow, B., Stearns, S. D.; *Adaptive signal processing*, Prentice Hall, Englewood Cliffs, New Jeresy, 1984.

[16] Haykin, S.; *Adaptive filter theory*, 2nd edition, Prentice Hall, Englewood Cliffs, New Jeresy, 1991.

[17] Alexander, S. T.; *Adaptive signal processing: theory and applications*, Springer-Verlag, New York, 1986.

[18] Haykin, S.; *Introduction to adaptive filters*, Macmillan, New York 1984.

[19] Monzingo, R. A. and Miller, T. W.; *Introduction to adaptive arrays*, Wiley-Interscience, New York, 1980.

[20] Goodwin, G. C. and Sin, K. S.; *Adaptive filtering, prediction and control*, Prentice Hall, Englewood Cliffs, New Jeresy, 1984.

[21] Goodwin, G. C. and Payne, R. L.; *Dynamic system identification: experiment design and data analysis*, Academic Press, New York, 1977.

[22] Chen, W. Y. and Haddad, R. A.; "Dual mode adaptive signal processing," Computers & Electrical Engineering, vol. 18, nos. 3-4, pp. 261–275, 1992.

[23] Duttweiler, D. L.; "A twelve channel digital echo canceller," IEEE Transactions on Communications, vol. COM-26, no. 5, pp. 647–653, May 1978.

[24] Honig, M. L.; "Echo cancellation of voiceband data signals using RLS and stochastic gradient algorithms," IEEE Transactions on Communications, vol. COM-33, no. 1, pp. 65–73, January 1985.

[25] Gritton, C. W. K. and Lin, D. W.; "Echo Cancellation Algorithms," IEEE ASSP Magazine, vol. 1, no. 2, pp. 30–38, April 1984.

[26] Sondhi, M. M. and Berkley, D. A.; "Silencing Echoes on the Telephone network," Proceedings of the IEEE, vol. 68, no. 8, pp. 948–963, August 1980.

[27] Sondhi, M. M. and Presti, A. U.; "A Self-Adaptive Echo Canceller," Bell Systems Technical Journal Briefs, vol. 45, no. 10, pp. 1851–1854, December 1966.

[28] Godard, D.; "Channel equalisation using a Kalman filter for fast data transmission," IBM Journal on Research and Development, vol. 18, no. 3, pp. 267–273, May 1974.

[29] Lucky, R.; "Automatic equalisation for digital communication," Bell Systems Technical Journal, vol. 44, no. 4, pp. 547–588, April 1965.

[30] Gersho, A.; "Adaptive equalization of highly dispersive channels for data transmission," Bell Systems Technical Journal, vol. 48, no. 1, pp. 55–70, January 1969.

[31] Nissen, C. W.; "Automatic channel equalisation algorithm," Proceedings of IEEE, vol. 55, no. 5, p. 698, May 1967.

[32] Gibson, G. J., Siu, S. and Cowan, C. F. N.; "The application of nonlinear structures to the reconstruction of binary signals," IEEE Transactions on Signal Processing, vol. 39, no. 8, pp. 1877–1884, August 1991.

[33] Proakis, J. G.; "Adaptive digital filters for equalisation of telephone channels," IEEE Transactions on Audio Electroacoustics, vol. AU-18, no. 2, pp. 195–200, June 1970.

[34] Makhoul, J.; "Linear prediction: a tutorial review," Proceedings of IEEE, vol. 63, no. 4, pp. 561–580, April 1975.

[35] Gray, A. H. and Markel, J. D.; *Linear Prediction of Speech*, Springer-Verlag, Berlin, 1976.

[36] Atal, B. S. and Schroeder, M. R.; "Adaptive predictive coding of speech signals," Bell Systems Technical Journal, vol. 49, no. 8, pp. 1973–1986, October 1970.

[37] Atal, B. S. and Schroeder, M. R.; "Predictive coding of speech and subjective error criteria," IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP-27, no. 3, pp. 247–254, June 1979.

[38] Widrow, B. et al.; "Adaptive noise cancellation: principles and applications," Proceedings of IEEE, vol. 63, no. 12, pp. 1692–1716, December 1975.

[39] Friedlander, B.; "System identification techniques for adaptive noise cancelling," IEEE Transactions on Acoustics, Speech, and signal Processing, vol. 1, no. 1, pp. 699–709, October 1982.

[40] Harrison, W. A., Lim, J. S. and Singer, E.; "A new application of adaptive noise cancellation," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-34, no. 1, pp. 21–27, February 1986.

[41] Hamming, R. W.; *Digital filters*, 2nd ed., Prentice-Hall, Englewood Cliffs, New Jeresy, 1983.

[42] Gold, B. and Rader, C. M.; *Digital processing of signals*, McGraw-Hill, New YOrk, 1969.

[43] Peled, A. and Liu, B.; *Digital signal processing: theory design and implementation*, Wiley, New York, 1976.

[44] Adams, P. F., Harbridge, J. R. and Macmillan, R. H.; "A MOS integrated circuit for digital filtering and level detection," IEEE Journal of Solid State Circuits, vol. SC-16, no. 3, pp. 183–190, June 1981.

[45] Kallman, H. E.; "Transversal filters," Proceedings of IRE, vol. 28, no. 7, pp. 302–310, July 1940.

[46] Zohar, S.; "New hardware realisations of non-recursive digital filters," IEEE Transactions on Computers, vol. C-22, no. 4, pp. 328-347, April 1973.

[47] Herrmann, O. and Schuessler, W.; "Design of nonrecursive digital filters with minimum phase," Electronic Letters, vol. 6, no. 11, pp. 329-330, 28th May 1970.

[48] Becker, F. K. and Rudin, H. R.; "Application of Automatic Transversal Filters to the Problem of Echo Suppression," Bell Systems Technical Journal, vol. 45, no. 10, pp. 1847–1850, December 1966.

[49] Horna, O. A.; "Echo canceler with adaptive transversal filter utilizing pseudo-logarithmic coding," COMSAT Technical Review, vol. 12, no. 2, pp. 393–428, Fall 1977.

[50] Roberts, R. A. and Mullis, C. T.; *Digital signal processing*, Addison Wesley, Reading, Massachusettes, 1987.

[51] Sherman, S.; "Non-mean square error criteria," IRE Transactions on Information Theory, vol. IT-4, no. 3, pp. 125–126, September 1958.

[52] Brown, J. L.; "Asymmetric nonmean square error criteria," IRE Transactions on Automatic Control, vol. AC-7, no. 1, pp. 64–66, January 1962.

[53] Zakai, M.; "General error criteria," IEEE Transactions on Information Theory, vol. IT-10, no. 1, pp. 94–95, January 1964.

[54] Douglas, S. C. and Meng, T. H. -Y.; "Stochastic gradient adaptation under general error criteria," IEEE Transactions on Signal Processing, vol. 42, no. 6, pp. 1335–1351, June 1994.

[55] Schultz, W. C. and Rideout, V. C.; "Control system performance measures: past, present and future," IRE Transactions on Automatic Control, vol. AC-6, no. 1, pp. 22–35, February 1961.

[56] Chambers, J. A., Tanrikulu, O. and Constantinides, A. G.; "Least mean mixed-norm adaptive filtering," Electronics Letters, vol. 30, no. 19, pp. 1574–1575, 15th September 1994.

[57] Sethares, W. A.; "Adaptive algorithms with nonlinear data and error functions," IEEE Transactions on Signal Processing, vol. 40, no. 9, pp. 2199–2206, September 1992.

[58] Jun, B. -E. and Park, D. -J.; 'Novel steepest descent adaptive filter derived from new performance function with additional exponential term," Signal Processing, vol. 36, no. 2, pp. 189–199, March 1994.

[59] Keerthi, S. S. and Gilbert, E. G.; "Optimal infinite-horizon control and the stabilization of linear discrete-time systems: state-control constraints and nonquadratic cost functions," IEEE Transactions on Automatic Control, vol. AC-31, no. 3, pp. 264–266, March 1986.

[60] Fang, G. S.; "Voice channel echo cancellation," IEEE Communications Magazine, vol. 21, no. 9, pp. 11–14, December 1983.

[61] Gould, R. G. and Helder, G. K.; "Transmission Delay and Echo Suppression," IEEE Spectrum, vol. 7, no. 4, pp. 47–54, April 1970.

[62] Brady, P. T.; "A Statistical analysis of ON-Off Patterns in sixteen conversations," Bell Systems Technical Journal, vol. 47, no. 1, pp. 73–91, January 1968.

[63] Marcos, S. and Macchi, O.; Joint adaptive echo cancellation and channel equalization for data transmission," Signal Processing, vol. 20, no. 1, pp. 43–65, May 1990.

[64] Bershad, N. J. and Bonnet, M.; "Saturation effects in LMS adaptive echo cancellation for binary data," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 38, no. 10, pp. 1687–1696, October 1990.

[65] Messerschmitt, D. G.; "Echo cancellation in speech and data transmission," IEEE Journal of Selected Areas in Communications, vol. SAC-2, no. 2, pp. 283–297, March 1984.

[66] Koll, V. G. and Weinstein, S. B.; "Simultaneous two-way data transmission over a two-wire circuit," IEEE Transactions on Communications, vol. COM-21, no. 2, pp. 143–147, February 1973.

[67] Gilsanz, M., Pedron, F. and Siles, J. A.; "Adaptive echo canceling for baseband data transmission," Electrical Communication, vol. 59, no. 3, pp. 338–344, 1985.

[68] Agazzi, O., Messerschmitt, D. G. and Hodges, D. A.; "Nonlinear echo cancellation of data signals," IEEE Transactions on Communications, vol. COM-30, no. 11, pp. 2421–2433, November 1982.

[69] Hu, R. and Ahmed, H. M.; "Echo cancellation in high speed data transmission systems using adaptive layered bilinear filters," IEEE Transactions on communications, vol. 42, nos. 2–4, pp. 655–663, February/March/April 1994.

[70] Borys, A., Rupperecht, W. and Trick, U.; 'Influence of nonlinearities on echo cancellation in two-wire full-duplex data transmission," NTZ Archiv, vol. 8, Part 8, pp. 185–190, 1986.

[71] Murano, K., Unagami, S. and Amano, F.; "Echo cancellation and applications," IEEE Communications Magazine, vol. 28, no. 1, pp. 49–55, January 1990.

[72] Cioffi, J. M. and Kailath, T.; "An efficient RLS data-driven echo canceller for fast initialization of full-duplex data transmission," IEEE Transactions on Communications, vol. COM-33, no. 7, pp. 601–611, July 1985.

[73] Dimolitsas, S. and Gunn, J. E.; "A length adaptive, transversal data echo cancelor," Signal Processing, vol. 12, no. 3, pp. 321–324, 1987.

[74] Yip, P. C. -W. and Etter, D. M.; "An adaptive multiple echo canceller for slowly time-varying echo paths," IEEE Transactions on Communications, vol. 38, no. 10, pp. 1693–1698, October 1990.

[75] Weinstein, S. B.; "A passband data-driven echo canceller for full-duplex transmission on two-wire circuits," IEEE Transactions on Communications, vol. COM-25, no. 7, pp. 654–666, July 1977.

[76] Mueller, K. H.; "A new digital echo canceler for two-wire full-duplex data transmission," IEEE Transactions on Communications, vol. COM-24, no. 9, pp. 956–962, September 1976.

[77] Macchi, O. and Marcos, S.; "Modeling and asynchronous data echo canceller," IEEE Transactions on Communications, vol. 37, no. 1, pp. 75–79, January 1989.

[78] Kanemasa, A. and Niwa, K.; "An adaptive-step sign algorithm for fast convergence of a data echo canceller," IEEE Transactions on Communications, vol. COM-35, no. 10, pp. 1102–1108, October 1987.

[79] Hsu, W., Chui, F. and Hodges, D. A.; "An acoustic echo canceler," IEEE Journal of Solid-State Circuits, vol. 24, no. 6, pp. 1639–1646, December 1989.

[80] Cioffi, J. M. and Ho, M.; "A finite precision analysis of the block-gradient adaptive data-driven echo canceller," IEEE Transactions on Communications, vol. 40, no. 5, pp. 940–946, May 1992.

[81] Cioffi, J. M.; "A fast echo canceller initialization method for the CCITT V.32 modem," IEEE Transactions on Communications, vol. 38, no. 5, pp. 629– 638, May 1990.

[82] Alvestad, T. and Eriksen, T. J. -C.; "Echo canceler for two-wire data modems," Electrical Communication, vol. 59, no. 3, pp. 333–337, 1985.

[83] Sondhi, M. M.; "An Adaptive Echo Canceller," Bell Systems Technical Journal, vol. 46, no. 3, pp. 497–511, March 1967.

[84] Siu, S. and Cowan, C. F. N.; "Performance analysis of the L_p norm back propagation algorithm for the adaptive equalisation," IEE Proceedings-F Radar and Signal Processing, vol. 140, no. 1, pp. 43–47, February 1993.

[85] Cowan, C. F. N. and Mirza, M. R.; "On the use of non-Euclidean cost functions in adaptive echo cancellers," IEE Coloquium on 'New Directions in Adaptive Signal Processing' (Digest no. 039), London, UK, pp. 10/1–3, 16 February 1993.

[86] Caraiscos, C. and Liu, B.; "A roundoff error analysis of the LMS adaptive algorithm," IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP-32, no. 1, pp. 34–41, February 1984.

[87] Kwong, R. H. and Johnston, E. W.; "A variable step size LMS algorithm," IEEE Transactions on Signal Processing, vol. 40, no. 7, pp. 1633–1642, July 1992.

[88] Bitmead, R. R.; "Convergence in distribution of LMS-type adaptive parameter estimation," IEEE Transactions on Automatic Control, vol. AC-28, no. 1, pp. 54–60, January 1983.

[89] Mathews, V. J. and Xie, Z.; "A stochastic gradient adaptive filter with gradient adaptive step size," IEEE Transactions on signal Processing, vol. 41, no. 6, pp. 2075–2087, June 1993.

[90] Mathews, V. J. and Cho, S. H.; "Improved convergence analysis of stochastic gradient adaptive filters using the sign algorithm," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-35, no. 4, pp. 450–454, April 1987.

[91] Bajpai, A. C., Calus, I. M. and Fairley, J. A.; *Mathematics for engineers and scientists, Volume 1*, Wiley, London, 1973.

[92] Haykin, S.; *Communication systems*, Wiley, New York, 1978.

Appendix A

Numerical values

A.1 A calculation of the numerical value of the far-end signal attenuation

A calculation is shown below to determine the multiplier f for the two alphabets of the bipolar far-end signal sequence. We take the case when far-end signal power level is 15 dB below as compared with that of the near-end signal power.

$$10 \log_{10} \left(\frac{Power_{far-end}}{Power_{near-end}} \right) = -15 \text{ dB}$$

$$\text{or} \quad 20 \log_{10} \left(\frac{| V_{far-end} |}{| V_{near-end} |} \right) = -15$$

$$\text{or} \quad \frac{| V_{far-end} |}{| V_{near-end} |} = 10^{-\frac{15}{20}}$$

If we take the near-end signal as our reference point with its power at 0 dB, then $| V_{near-end} |$ becomes 1 and the above equation reduces to

$$| V_{far-end} | = 0.177827941$$

near-end signal power level dBs	far-end signal power level dBs	multiplier f
0	-15	0.177827941
0	-20	0.1
0	-25	0.056234133
0	-30	0.031622777

Table A.1: Various far-end signal levels used in the simulations along with the corresponding values of multiplier f used to obtain the far-end signal sequences from the sequence {1,-1} and {3,1,-1,-3}.

$$\text{or} \quad V_{far-end} = 0.177827941, -0.177827941$$

Thus the far-end signal is a pseudo random sequence taken from the set $\{0.177827941, -0.177827941\}$. Thus the number 0.177827941 is the multiplier f to the sequence $\{1, -1\}$, to obtain the far-end signal level of -15 dB. Similar calculations were performed to find out the alphabets of the far-end signal sequences for the far-end signal levels of -20, -25, and -30 dBs. Different far-end signal levels used in the simulations and the relative multipliers f to obtain these, by multiplying with the sequence from the set {1,-1} are shown in Table A.1.

A.2 Numerical values of filter coefficients

Numerical values of the filters' coefficients or tap gains for the first and second echo path models and of the dispersion filter of the far-end signal are recorded

here for reference. The numerical values of the coefficients of the the third echo path model are given in Section 4.3.

A.2.1 Numerical values of the filter coefficients for the first echo path model

A calculation to obtain these values is given in Section 4.3. The filter coefficients thus obtained upto 6th decimal place are as below.

$$\{h\} = \{1.000000, 0.800250, 0.640400, 0.512480, 0.410112, 0.328192,$$
$$0.262635, 0.210174, 0.168192, 0.134596, 0.107710, 0.086195, 0.068978,$$
$$0.055199, 0.044173, 0.035350, 0.028289, 0.022638, 0.018116, 0.014497,$$
$$0.011601, 0.009284, 0.007430, 0.005946, 0.004758, 0.003808, 0.003047,$$
$$0.002438, 0.001951, 0.001562, 0.001250, 0.001000\}$$

A.2.2 Numerical values of the filter coefficients for the second echo path model

A calculation to obtain these values is given in Section 4.3. The filter coefficients thus obtained upto 6th decimal place are as below.

$$\{h\} = \{1.000000, 0.640400, 0.410112, 0.262636, 0.168192, 0.107710,$$
$$0.068978, 0.044173, 0.028289, 0.018116, 0.011601, 0.007430, 0.004758,$$
$$0.003047, 0.001951, 0.001250, 0.000800, 0.000512, 0.000328, 0.000210,$$
$$0.000135, 0.000086, 0.000055, 0.000035, 0.000023, 0.000014, 0.000009,$$
$$0.000005, 0.000003, 0.000002, 0.000001, 0.000001\}$$

A.2.3 Numerical values of the filter coefficients used for the dispersion of the far-end signal

A calculation to obtain these values is given in Section 4.3. The filter coefficients thus obtained upto 6th decimal place are as below.

$$\{h\} = \{1.000000, 0.689779, 0.475794, 0.328193, 0.226380, 0.156152,$$

$$0.107711, 0.074296, 0.051248, 0.035350, 0.024384, 0.016819, 0.011602,$$

$$0.008003, 0.005520, 0.003808, 0.002626, 0.001812, 0.001250, 0.000862,$$

$$0.000595, 0.000410, 0.000283, 0.000195, 0.000135, 0.000093, 0.000064,$$

$$0.000044, 0.000030, 0.000021, 0.000014, 0.000010\}$$

Appendix B

Computer programmes

B.1 Fortran source code used for binary data sequences without switching of gradient during convergence

```
      program noquad1
c
c Declarations of variables
c
      double precision taphat(0:31),taptar(0:31)
      double precision yk,yhat,ek,taperror(0:31),fx
      double precision taphatn(0:31)
      double precision dsumhi,dsumhi_h,tau,a,tau1
      double precision x(-31:0),loopgain,s,loopgain1
      real bit,z1,z2,average1(50000)
      real hsum,q,pq
      integer n,m,i,k,p,pp,k1,l,l1,l2,mp
      integer average2(50000)
c
c Opening the input data file for echo path models
c
      open(2,file='model.1',status='unknown',
     + access='sequential',form='formatted')
c
c Opening the output file
```

```
c
      open(1,file='fort.1',status='unknown',
     + access='sequential',form='formatted')
c
c Input from keyboard starts
c
      write(*,*)
      write(*,*)"cost function power tau"
      read(*,*)tau1 ! Read current value of tau
      write(*,*)
      write(*,*)"number of points for iteration"
      read(*,*)m ! Read number of iterations
      write(*,*)
      write(*,*)"interval for plotting"
      read(*,*)pp !Used only for storing and plotting
      write(*,*)
      write(*,*)"step size"
      read(*,*)loopgain1 ! Read current value of step size
      write(*,*)
c
c Input from keyboard ends
c
      n = 32 ! Number of filter coefficients
      mp = m/pp ! Number of total plotting points
      l1 = 20 ! Number of averagings
      dsumhi = 0.0
      do 10 i = 0,n-1
            read(2,*)a ! Read echo path data
            taptar(i) = a ! Generation of target filter co-efficients
            dsumhi = dsumhi + (taptar(i))**2
10      continue
c
      do 40 l = 1,mp
            average1(l) = 0.0
            average2(l) = 0
40      continue
c
c Start of the averaging loop
```

```
c
      do 90 l = 1,l1
            l2 = 0
            do 20 i = -n+1,0
                  x(i) = 0.0 ! Forcing initial conditions to zero
                  taphat(-i) = 0.0 ! Initialisation
20          continue
            k1 = 0
c
c Start of iteration loop
c
            do 80 k = 0,m
                  z1 = rand(0) ! Generation of near-end signal starts
                  if(z1 .ge. 0.5) then
                        bit = 1.0
                  else
                        bit = -1.0
                  end if
                  x(0) = bit ! Near-end signal
                  z2 = rand(0) ! Generation of far-end signal starts
                  if(z2 .ge. 0.5) then
                        bit = 1.0
                  else
                        bit = -1.0
                  end if
                  fx = bit*0.177827941 ! attenuated far-end signal
                  yk = 0.0
                  yhat = 0.0
                  do 50 i = 0,n-1
                        yk = yk + taptar(i)*x(-i) ! Target or echo path fil-
ter output
                        yhat = yhat + taphat(i)*x(-i) ! Adaptive filter out-
put
50                continue
                  ek = (yk + fx) - yhat ! Error.signal
                  tau = tau1
                  loopgain = loopgain1
                  if (ek.lt.0.0) then
```

```
                        s = -1.0
                    else
                        s = 1.0
                    end if
                    dsumhi_h = 0.0
                    do 70 i = 0,n-1
                        taphatn(i)=taphat(i)+tau*loopgain*x(0-i)*
     +                 ((dabs(ek))**(tau-1.0))*s
                        taperror(i) = taptar(i)-taphatn(i)
                        dsumhi_h = dsumhi_h + (taperror(i))**2
                        taphat(i) = taphatn(i)
70                  continue
                    p = k/pp
                    pq = pp
                    q = float(k)/pq
                    if(q.eq.float(p))then
                        hsum = dsumhi_h/dsumhi
                        l2 = l2 + 1
                        average1(l2) = average1(l2) + (10.0*log10(hsum))
                        if(l.eq.1)then
                            average2(l2) = k
                        end if
                    end if
                    do 78 i = -31,-1
                        x(i) = x(i+1)
78                  continue
80              continue ! End of iteration loop
                write(*,*)" averaging loop no",l ! Echo the counter on screen
90      continue ! End of the averaging loop
        do 100 l = 1,mp ! Write to output file
            write(1,*)average2(l),average1(l)/(float(l1))
100         continue
        close(1,status='keep') ! Close output file
        close(2,status='keep') ! Close input file
        stop
        end
```

B.2 Fortran source code used for binary data sequences with switching of gradient during convergence

```
      program noquad2
c
c Declarations of variables
c
      double precision taphat(0:31),taptar(0:31)
      double precision yk,yhat,ek,taperror(0:31),fx
      double precision taphatn(0:31)
      double precision dsumhi,dsumhi_h,tau,a,tau1
      double precision x(-31:0),loopgain,s,loopgain1
      real bit,z1,z2,average1(50000)
      real hsum,q,pq
      integer n,m,i,k,p,pp,k1,l,l1,l2,mp
      integer average2(50000)
c
c Opening the input data file for echo path models
c
      open(2,file='model.1',status='unknown',
     + access='sequential',form='formatted')
c
c Opening the output file
c
      open(1,file='fort.1',status='unknown',
     + access='sequential',form='formatted')
c
c Input from keyboard starts
c
      write(*,*)
      write(*,*)"cost function power tau"
      read(*,*)tau1 ! Read current value of tau
      write(*,*)
      write(*,*)"number of points for iteration"
      read(*,*)m ! Read number of iterations
      write(*,*)
```

```
      write(*,*)"interval for plotting"
      read(*,*)pp !Used only for storing and plotting
      write(*,*)
      write(*,*)"step size"
      read(*,*)loopgain1 ! Read current value of step size
      write(*,*)
c
c Input from keyboard ends
c
      n = 32 ! Number of filter coefficients
      mp = m/pp ! Number of total plotting points
      l1 = 20 ! Number of averagings
      dsumhi = 0.0
      do 10 i = 0,n-1
            read(2,*)a ! Read echo path data
            taptar(i) = a ! Generation of target filter co-efficients
            dsumhi = dsumhi + (taptar(i))**2
10       continue
c
      do 40 l = 1,mp
            average1(l) = 0.0
            average2(l) = 0
40       continue
c
c Start of the averaging loop
c
      do 90 l = 1,l1
            l2 = 0
            do 20 i = -n+1,0
                  x(i) = 0.0 ! Forcing initial conditions to zero
                  taphat(-i) = 0.0 ! Initialisation
20          continue
            k1 = 0
c
c Start of iteration loop
c
            do 80 k = 0,m
                  z1 = rand(0) ! Generation of near-end signal starts
```

```
                    if(z1 .ge. 0.5) then
                          bit = 1.0
                    else
                          bit = -1.0
                    end if
                    x(0) = bit ! Near-end signal
                    z2 = rand(0) ! Generation of far-end signal starts
                    if(z2 .ge. 0.5) then
                          bit = 1.0
                    else
                          bit = -1.0
                    end if
                    fx = bit*0.177827941 ! attenuated far-end signal
                    yk = 0.0
                    yhat = 0.0
                    do 50 i = 0,n-1
                          yk = yk + taptar(i)*x(-i) ! Target or echo path fil-
ter output
                          yhat = yhat + taphat(i)*x(-i) ! Adaptive filter out-
put
50                  continue
                    ek = (yk + fx) - yhat ! Error signal
                    if(dabs(ek).ge.1.0)then ! Switching of gradient starts
                          tau = 2.0 ! Switch back to lowest value of τ
                          loopgain = 0.0009 ! Switch back to μ for τ = 2
                    else
                          tau = tau1
                          loopgain = loopgain1
                    end if
                    if (ek.lt.0.0) then
                          s = -1.0
                    else
                          s = 1.0
                    end if
                    dsumhi_h = 0.0
                    do 70 i = 0,n-1
                          taphatn(i)=taphat(i)+tau*loopgain*x(0-i)*
   +                     ((dabs(ek))**(tau-1.0))*s
```

```
                        taperror(i) = taptar(i)-taphatn(i)
                        dsumhi_h = dsumhi_h + (taperror(i))**2
                        taphat(i) = taphatn(i)
70                  continue
                    p = k/pp
                    pq = pp
                    q = float(k)/pq
                    if(q.eq.float(p))then
                        hsum = dsumhi_h/dsumhi
                        l2 = l2 + 1
                        average1(l2) = average1(l2) + (10.0*log10(hsum))
                        if(l.eq.1)then
                            average2(l2) = k
                        end if
                    end if
                    do 78 i = -31,-1
                        x(i) = x(i+1)
78                  continue
80              continue ! End of iteration loop
                write(*,*)" averaging loop no",l ! Echo the counter on screen
90      continue ! End of the averaging loop
        do 100 l = 1,mp ! Write to output file
            write(1,*)average2(l),average1(l)/(float(l1))
100         continue
        close(1,status='keep') ! Close output file
        close(2,status='keep') ! Close input file
        stop
        end
```

B.3 Fortran source code used for quaternary data sequences without switching of gradient during convergence

```
      program noquad3
c
c Declarations of variables
c
      double precision taphat(0:31),taptar(0:31)
      double precision yk,yhat,ek,taperror(0:31),fx
      double precision taphatn(0:31)
      double precision dsumhi,dsumhi_h,tau,a,tau1
      double precision x(-31:0),loopgain,s,loopgain1
      real bit,z1,z2,average1(50000)
      real hsum,q,pq
      integer n,m,i,k,p,pp,k1,l,l1,l2,mp
      integer average2(50000)
c
c Opening the input data file for echo path models
c
      open(2,file='model.1',status='unknown',
     + access='sequential',form='formatted')
c
c Opening the output file
c
      open(1,file='fort.1',status='unknown',
     + access='sequential',form='formatted')
c
c Input from keyboard starts
c
      write(*,*)
      write(*,*)"cost function power tau"
      read(*,*)tau1 ! Read current value of tau
      write(*,*)
      write(*,*)"number of points for iteration"
      read(*,*)m ! Read number of iterations
      write(*,*)
```

```
      write(*,*)"interval for plotting"
      read(*,*)pp !Used only for storing and plotting
      write(*,*)
      write(*,*)"step size"
      read(*,*)loopgain1 ! Read current value of step size
      write(*,*)
c
c Input from keyboard ends
c
      n = 32 ! Number of filter coefficients
      mp = m/pp ! Number of total plotting points
      l1 = 20 ! Number of averagings
      dsumhi = 0.0
      do 10 i = 0,n-1
            read(2,*)a ! Read echo path data
            taptar(i) = a ! Generation of target filter co-efficients
            dsumhi = dsumhi + (taptar(i))**2
10       continue
c
      do 40 l = 1,mp
            average1(l) = 0.0
            average2(l) = 0
40       continue
c
c Start of the averaging loop
c
      do 90 l = 1,l1
            l2 = 0
            do 20 i = -n+1,0
                  x(i) = 0.0 ! Forcing initial conditions to zero
                  taphat(-i) = 0.0 ! Initialisation
20          continue
            k1 = 0
c
c Start of iteration loop
c
            do 80 k = 0,m
                  z1 = rand(0) ! Generation of near-end signal starts
```

```
                    if(z1 .ge. 0.75) then
                          bit = 3.0
                    elseif(z1 .ge. 0.5) then
                          bit = 1.0
                    elseif(z1 .ge. 0.25) then
                          bit = -1.0
                    else
                          bit = -3.0
                    end if
                    x(0) = bit ! Near-end signal
                    z2 = rand(0) ! Generation of far-end signal starts
                    if(z2 .ge. 0.75) then
                          bit = 3.0
                    elseif(z2 .ge. 0.5) then
                          bit = 1.0
                    elseif(z2 .ge. 0.25) then
                          bit = -1.0
                    else
                          bit = -3.0
                    end if
                    fx = bit*0.177827941 ! attenuated far-end signal
                    yk = 0.0
                    yhat = 0.0
                    do 50 i = 0,n-1
                          yk = yk + taptar(i)*x(-i) ! Target or echo path fil-
ter output

                          yhat = yhat + taphat(i)*x(-i) ! Adaptive filter out-
put
50                  continue
                    ek = (yk + fx) - yhat ! Error signal
                    tau = tau1
                    loopgain = loopgain1
                    if (ek.lt.0.0) then
                          s = -1.0
                    else
                          s = 1.0
                    end if
                    dsumhi_h = 0.0
```

```
                    do 70 i = 0,n-1
                          taphatn(i)=taphat(i)+tau*loopgain*x(0-i)*
     +                   ((dabs(ek))**(tau-1.0))*s
                          taperror(i) = taptar(i)-taphatn(i)
                          dsumhi_h = dsumhi_h + (taperror(i))**2
                          taphat(i) = taphatn(i)
70                  continue
                    p = k/pp
                    pq = pp
                    q = float(k)/pq
                    if(q.eq.float(p))then
                          hsum = dsumhi_h/dsumhi
                          l2 = l2 + 1
                          average1(l2) = average1(l2) + (10.0*log10(hsum))
                          if(l.eq.1)then
                                average2(l2) = k
                          end if
                    end if
                    do 78 i = -31,-1
                          x(i) = x(i+1)
78                  continue
80            continue ! End of iteration loop
              write(*,*)" averaging loop no",l ! Echo the counter on screen
90      continue ! End of the averaging loop
        do 100 l = 1,mp ! Write to output file
              write(1,*)average2(l),average1(l)/(float(l1))
100         continue
        close(1,status='keep') ! Close output file
        close(2,status='keep') ! Close input file
        stop
        end
```

B.4 Fortran source code used for quaternary data sequences with switching of gradient during convergence

```
      program noquad4
c
c Declarations of variables
c
      double precision taphat(0:31),taptar(0:31)
      double precision yk,yhat,ek,taperror(0:31),fx
      double precision taphatn(0:31)
      double precision dsumhi,dsumhi_h,tau,a,tau1
      double precision x(-31:0),loopgain,s,loopgain1
      real bit,z1,z2,average1(50000)
      real hsum,q,pq
      integer n,m,i,k,p,pp,k1,l,l1,l2,mp
      integer average2(50000)
c
c Opening the input data file for echo path models
c
      open(2,file='model.1',status='unknown',
     + access='sequential',form='formatted')
c
c Opening the output file
c
      open(1,file='fort.1',status='unknown',
     + access='sequential',form='formatted')
c
c Input from keyboard starts
c
      write(*,*)
      write(*,*)"cost function power tau"
      read(*,*)tau1 ! Read current value of tau
      write(*,*)
      write(*,*)"number of points for iteration"
      read(*,*)m ! Read number of iterations
      write(*,*)
```

```
      write(*,*)"interval for plotting"
      read(*,*)pp !Used only for storing and plotting
      write(*,*)
      write(*,*)"step size"
      read(*,*)loopgain1 ! Read current value of step size
      write(*,*)
c
c Input from keyboard ends
c
      n = 32 ! Number of filter coefficients
      mp = m/pp ! Number of total plotting points
      l1 = 20 ! Number of averagings
      dsumhi = 0.0
      do 10 i = 0,n-1
            read(2,*)a ! Read echo path data
            taptar(i) = a ! Generation of target filter co-efficients
            dsumhi = dsumhi + (taptar(i))**2
10       continue
c
      do 40 l = 1,mp
            average1(l) = 0.0
            average2(l) = 0
40       continue
c
c Start of the averaging loop
c
      do 90 l = 1,l1
            l2 = 0
            do 20 i = -n+1,0
                  x(i) = 0.0 ! Forcing initial conditions to zero
                  taphat(-i) = 0.0 ! Initialisation
20          continue
            k1 = 0
c
c Start of iteration loop
c
            do 80 k = 0,m
                  z1 = rand(0) ! Generation of near-end signal starts
```

```
                    if(z1 .ge. 0.75) then
                          bit = 3.0
                    elseif(z1 .ge. 0.5) then
                          bit = 1.0
                    elseif(z1 .ge. 0.25) then
                          bit = -1.0
                    else
                          bit = -3.0
                    end if
                    x(0) = bit ! Near-end signal
                    z2 = rand(0) ! Generation of far-end signal starts
                    if(z2 .ge. 0.75) then
                          bit = 3.0
                    elseif(z2 .ge. 0.5) then
                          bit = 1.0
                    elseif(z2 .ge. 0.25) then
                          bit = -1.0
                    else
                          bit = -3.0
                    end if
                    fx = bit*0.177827941 ! attenuated far-end signal
                    yk = 0.0
                    yhat = 0.0
                    do 50 i = 0,n-1
                          yk = yk + taptar(i)*x(-i) ! Target or echo path fil-
ter output

                          yhat = yhat + taphat(i)*x(-i) ! Adaptive filter out-
put
50                  continue
                    ek = (yk + fx) - yhat ! Error signal
                    if(dabs(ek).ge.1.0)then ! Switching of gradient starts
                          tau = 2.0 ! Switch back to lowest value of τ
                          loopgain = 0.0009 ! Switch back to μ for τ = 2
                    else
                          tau = tau1
                          loopgain = loopgain1
                    end if
                    if (ek.lt.0.0) then
```

```
                        s = -1.0
                    else
                        s = 1.0
                    end if
                    dsumhi_h = 0.0
                    do 70 i = 0,n-1
                        taphatn(i)=taphat(i)+tau*loopgain*x(0-i)*
   +                   ((dabs(ek))**(tau-1.0))*s
                        taperror(i) = taptar(i)-taphatn(i)
                        dsumhi_h = dsumhi_h + (taperror(i))**2
                        taphat(i) = taphatn(i)
70                  continue
                    p = k/pp
                    pq = pp
                    q = float(k)/pq
                    if(q.eq.float(p))then
                        hsum = dsumhi_h/dsumhi
                        l2 = l2 + 1
                        average1(l2) = average1(l2) + (10.0*log10(hsum))
                        if(l.eq.1)then
                            average2(l2) = k
                        end if
                    end if
                    do 78 i = -31,-1
                        x(i) = x(i+1)
78                  continue
80              continue ! End of iteration loop
                write(*,*)" averaging loop no",l ! Echo the counter on screen
90      continue ! End of the averaging loop
        do 100 l = 1,mp ! Write to output file
            write(1,*)average2(l),average1(l)/(float(l1))
100         continue
        close(1,status='keep') ! Close output file
        close(2,status='keep') ! Close input file
        stop
        end
```

B.5 Fortran source code used for binary data sequences with dispersion added to far-end signal

```
      program noquad5
c
c Declarations of variables
c
      double precision taphat(0:31),taptar(0:31)
      double precision yk,yhat,ek,taperror(0:31),fx(-31:0),fd
      double precision taphatn(0:31)
      double precision dsumhi,dsumhi_h,tau,a,tau1,b
      double precision x(-31:0),loopgain,s,loopgain1
      real bit,z1,z2,average1(50000)
      real hsum,q,pq
      integer n,m,i,k,p,pp,k1,l,l1,l2,mp
      integer average2(50000)
c
c Opening the input data file for echo path models
c
      open(2,file='model.1',status='unknown',
     + access='sequential',form='formatted')
c
c Opening the output file
c
      open(1,file='fort.1',status='unknown',
     + access='sequential',form='formatted')
c
c Opening the input dispersion data file
c
      open(3,file='disp',status='unknown',
     + access='sequential',form='formatted')
c
c Input from keyboard starts
c
      write(*,*)
      write(*,*)"cost function power tau"
```

```
      read(*,*)taul ! Read current value of tau
      write(*,*)
      write(*,*)"number of points for iteration"
      read(*,*)m ! Read number of iterations
      write(*,*)
      write(*,*)"interval for plotting"
      read(*,*)pp !Used only for storing and plotting
      write(*,*)
      write(*,*)"step size"
      read(*,*)loopgain1 ! Read current value of step size
      write(*,*)
c
c Input from keyboard ends
c
      n = 32 ! Number of filter coefficients
      mp = m/pp ! Number of total plotting points
      l1 = 20 ! Number of averagings
      dsumhi = 0.0
      do 10 i = 0,n-1
            read(2,*)a ! Read echo path data
            taptar(i) = a ! Generation of target filter co-efficients
            read(2,*)b ! Read dispersion filter data
            ftar(i) = b ! Generation of dispersion filter co-efficients
            dsumhi = dsumhi + (taptar(i))**2
10      continue
c
      do 40 l = 1,mp
            average1(l) = 0.0
            average2(l) = 0
40      continue
c
c Start of the averaging loop
c
      do 90 l = 1,l1
            l2 = 0
            do 20 i = -n+1,0
                  x(i) = 0.0 ! Forcing initial conditions to zero
                  taphat(-i) = 0.0 ! Initialisation
```

```
20          continue
            k1 = 0
c
c Start of iteration loop
c
            do 80 k = 0,m
                  z1 = rand(0) ! Generation of near-end signal starts
                  if(z1 .ge. 0.5) then
                        bit = 1.0
                  else
                        bit = -1.0
                  end if
                  x(0) = bit ! Near-end signal
                  z2 = rand(0) ! Generation of far-end signal starts
                  if(z2 .ge. 0.5) then
                        bit = 1.0
                  else
                        bit = -1.0
                  end if
                  fx(0) = bit*0.177827941 ! attenuated far-end signal
                  yk = 0.0
                  yhat = 0.0
                  fd = 0.0
                  do 50 i = 0,n-1
                        yk = yk + taptar(i)*x(-i) ! Target or echo path fil-
ter output
                        yhat = yhat + taphat(i)*x(-i) ! Adaptive filter out-
put
                        fd = fd + ftar(i)*x(-i) ! Dispersion filter output
50                continue
                  ek = (yk + fd) - yhat ! Error signal
                  tau = tau1
                  loopgain = loopgain1
                  if (ek.lt.0.0) then
                        s = -1.0
                  else
                        s = 1.0
                  end if
```

```
                    dsumhi_h = 0.0
                    do 70 i = 0,n-1
                          taphatn(i)=taphat(i)+tau*loopgain*x(0-i)*
     +                   ((dabs(ek))**(tau-1.0))*s
                          taperror(i) = taptar(i)-taphatn(i)
                          dsumhi_h = dsumhi_h + (taperror(i))**2
                          taphat(i) = taphatn(i)
70                  continue
                    p = k/pp
                    pq = pp
                    q = float(k)/pq
                    if(q.eq.float(p))then
                          hsum = dsumhi_h/dsumhi
                          l2 = l2 + 1
                          average1(l2) = average1(l2) + (10.0*log10(hsum))
                          if(l.eq.1)then
                                average2(l2) = k
                          end if
                    end if
                    do 78 i = -31,-1
                          x(i) = x(i+1)
                          fx(i) = fx(i+1)
78                  continue
80            continue ! End of iteration loop
              write(*,*)" averaging loop no",l ! Echo the counter on screen
90    continue ! End of the averaging loop
      do 100 l = 1,mp ! Write to output file
            write(1,*)average2(l),average1(l)/(float(l1))
100       continue
      close(1,status='keep') ! Close output file
      close(2,status='keep') ! Close input file
      close(3,status='keep') ! Close dispersion data file
      stop
      end
```