623.817324
1991
FAR

Design and Implementation of a Digital Filter Bank

BY

Farook Sattar

A Thesis

Submitted to the Department of Electrical and Electronic Engineering, BUET, in partial fulfilment of the requirements for the degree

OF

79626
79626
6/5/91

MASTER OF SCIENCE IN ELECTRICAL AND ELECTRONIC ENGINEERING

BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY, DHAKA.

JANUARY, 1991

#79626#

The thesis titled"Design and Implementation of a Digital Filter Bank" Accepted as satisfactory in partial fulfillment of the requirements for the degree of Master of Science in Engineering (M. Sc. Engg.) in Electrical and Electronic Engineering of Farook Sattar, Roll no.- 881309F, Session:1986-1987.

Board of Examiners

1. ---------------------------- 21/1/91
(Dr. Kazi Mohiuddin Ahmed)
Associate Professor,
Department of Electrical and
Electronic Engineering,
BUET, Dhaka.

Chairman
(Supervisor)

2. ----------------------------
(Prof. Goran Salomonsson)
Department of Telecomm. Theory,
University of Lund, Sweden.

Co-Supervisor

3. ---------------------------- 21.1.91.
(Dr. S. F. Rahman)
Professor and Head,
Department of Electrical and
Electronic Engineering,
BUET, Dhaka.

Member
(Ex-Officio)

4. ---------------------------- 21.1.1991
(Dr. M. A. Matin)
Professor,
Department of Electrical and
Electronic Engineering,
BUET, Dhaka.

Member

5. Saiful Islam ---------------------------- 21-1-91
(Dr. Saiful Islam)
Professor,
Department of Electrical and
Electronic Engineering,
BUET, Dhaka.

Member

6. ---------------------------- 21.1.91
(Dr. Syed Mahbubur Rahman)
Associate Professor and Head,
Department of Computer Science and
Engineering, BUET, Dhaka.

Member
(External)

CERTIFICATE

This is to certify that this work has been done by me and it has not been submitted elsewhere for the award of any degree or diploma.

Signature of the student

Farook Sattar

21-01-91

(*Farook Sattar*)

Dedicated to Dr. Alia Ahmad

CONTENTS

LIST OF FIGURES

APPENDIX

Appendix A	MATLAB programs relating to perfect reconstruction of the signal and frequency sub-bands selection
Appendix B	MATLAB programs involving echo cancelling using white noise as the input signal
Appendix C	MATLAB programs involving echo cancelling using echo speech as the input signal
Appendix D	Values of the coefficients of the two analysis filters and the hybrid filter

Acknowledgement

The author is much obliged to his thesis supervisor Dr. Kazi Mohiuddin Ahmed, Associate Professor of Electrical and Electronic Engineering Department, BUET, Dhaka for entrusting him with this work. He is grateful to him for his supervision, necessary and valuable suggestions, correction and evaluation while going through the manuscript.

Special thanks and sincere gratitude are extended to Prof. Göran Salomonsson, Dept. of Telecomm. Theory, University of Lund, Sweden for his active guidance and supervision of the thesis work, carried on at the Dept. of Telecomm. Theory of the University of Lund, Sweden.

The author likes to express his thanks to Dr. S. F. Rahman, Professor and Head , Dept. of Electrical and Electronic Engineering, BUET, Dhaka for his help and co-operation to complete this work.

The author is humbly grateful to Prof. M. A. Matin, Dept. of Electrical and Electronic Engineering, BUET, Dhaka for his encouragement and sincere wishes to complete the work successfully.

Thanks are due to Mr. Benny Lövström and Mr. Leif Karlsson of the Department of Telecomm. Theory of Lund University, Sweden for their kind co-operation and assistance in the revision of the text and computer usage during the work.

Lastly the author acknowledges the kind assistance from Dr. Shahidul Hassan, Associate Professor, Dr. M. A. Choudhury, Assistant Professor and Mr. A. H. M. Zahirul Alam, Assistant Professor of the Dept. of EEE., BUET, Dhaka during his computer usage in BUET, Dhaka.

Abstract

There has been a growing interest in the field of digital filter banks, especially in construction of distortionfree filter banks with linear phase. It has been applied especially in design of low-bit rate speech coder. Filter banks are also valuable when parallel processing is used. Real time digital filter banks, with internal processing capability, is used in echo cancelling, speech processing and biomedical signal applications. In this thesis a QMF (quadrature mirror filter) based FIR filter bank has been implemented as a sub-band selector and as an echo canceller. Theoretical analysis and computer simulation shows that the digital filter bank splits the input signal into several frequency bands from which it can reconstruct a delayed replica of the input signal. When processing is performed in the sub-bands, the filter bank works as a sub-band selector and as an echo canceller respectively.

Chapter 1

79626
6-5-91
লাইব্রেরী, ঢাকা

Introduction

Over the past several years, Finite Impulse Response (FIR) digital filters, i.e., digital filters for which the impulse response is of finite duration, have become increasingly important. The way of designing FIR filters by different methods provides an efficient means for implementing high order digital filters. As an additional consideration, FIR filters with linear-phase have been designed and people are trying to implement them in the various fields of digital system.

In the mid-1960s when the signal processing based on digital technology had emerged as a new branch, the importance of digital filters for the digital system has arisen. Prior to this, different types of analog filters such as passive and active filters were used extensively for the analog/continuous-time system. Switched capacitor filters have then come which are discrete-time but not digital; i.e., signal values are clocked so that the time is quantized but the signal amplitudes are represented in analog form. With the developement of digital system, the necessity of digital filters has gained the focus of attention which involves both the time and amplitude quantization. Gradually different types of digital filters have been designed, improved and implemented with the developement of digital system. First the Infinite impulse response (IIR) digital filters have been designed by transforming the frequency response of the corresponding analog filters with the help of different designing methods. FIR filters have been designed from the IIR filters. Historically the ideas for designing the FIR filters was obtained from the truncation or modification of the corresponding infinitely long impulse response of the IIR filters. This has introduced a new "window" method which became the most widely used technique for the approximate design of FIR filters. By the "window" method the use of a variety of windows including Kaiser window and the Dolph-Chebyshev window have become effective. But the window method does not permit individual control over the approximation error in different bands. So the next approach was to design the FIR filters which have given better filter result from minimization the maximum error or a frequency weighted error criteria on the basis of some algorithms. In the mean time Herrmann (1970)[1], Herrmann and Schussler (1970)[2], Hofstetter (1971)[3], Parks and McClellan (1972)[4], Rabiner (1972)[5] have developed different procedures but the Parks-McClellan (1972)[4] algorithm has become the dominent method for the optimal approximation design of FIR filters. Another approach to FIR filter design corresponds to a non-linear opimization of the filter characteristics proposed by Herrmann and Schuessler[6] constraining the equiripple frequency characteristic by means of a set of non-linear equations. In the mean time FIR adaptive filters have come which use different adaptive algorithms for the adaptation of their coefficients/weights. In recent years, the QMF filters have been of great interest and are being used in many speech and communications applications. These filters provide significant improvement over the conventional optimal equiripple and window designs especially for their usage in the analysis/reconstruction system.

Our purpose is to design a multiple channel digital filter bank using quadrature mirror filters (QMF)[7] and use it in some applications. Digital filters operate on digital inputs and generate digital outputs. All digital filters restrict to either of two forms: non-recursive and recursive. A non-recursive filter generates its output by simply weighing the inputs by constants

(coefficients) and then summing the weighted inputs. The coefficients determine the filter. The second form of digital filter is recursive. In this case, the output is not only a function of the inputs, but it also depends on the past outputs. They are classified also into Finite ΩImpulse Response (FIR) and Infinite Impulse Response (IIR) filters, having finite and infinite length respectively in their impulse response. The QMFs are two filters which have a cut-off frequency at the quadrature (one-fourth) of the sampling frequency and where one of the filters is a mirror or image of the other. In other words QMF is a product of low-pass and high-pass filters having the same cut-off frequency. The digital filter bank splits the signal into several frequency bands from which the signal can be reconstructed without distortion. The signal has only been delayed and it is possible to know the delay from the direct relation between delay and the order of the filter. The condition for the perfect reconstruction is that the CQF (conjugate quadrature filters), i.e. the product of the analysis and synthesis filters, should have a linear phase. CQF is a product of two filters which are conjugate to each-other (for a real sequence) and have a cut-off frequency which is quadrature (one-fourth) of the sampling frequency. The use of the filter bank comes when processing is performed in the subbands. With processing, the filter bank works as a subband selector or as an echo canceler. Using an encoder in the processor, any of the subbands from the total frequency band can be selected. Adaptive processing can be done for echo canceling. Echo, an undesired extraneous signal in telephone systems, cancels out by the adaptive filters in the processor.

1.1 Objectives of the thesis

The objectives of the thesis are as follows:

- To reconstruct a signal perfectly with only a delay in the absence of any processor.
- To split and select the frequency sub-bands from the frequency band of the signal.
- To cancel the echo of the speech signal for long distance telephone lines.

1.2 Outline of the thesis

Outlining the organisation of this thesis, in Chapter 2 digital filters including their classification, structures, designs, and applications etc. have been discussed briefly. The basic structure of the digital filter bank has been described in Chapter 3. Theoretical analysis related to the exact reconstruction of the signal are shown there. The computer simulation results for both the processing and nonprocessing conditions are also given in Chapter 3. The principle of an echo canceller, an echo canceller using the filter bank, optimal weights of adaptive filters and the LMS (least mean square) algorithm [8] are presented in Chapter 4. The performance of the filter bank as an echo canceller is shown in Chapter 5 which presents simulation results using noise or speech signal as input. In Chapter 6 the conclusions of the paper with summary and discussions are drawn.

In the following chapter we are going to describe briefly about the digital filters including their classification, structures, designs and applications etc.

Chapter 2

Digital filters

Filters are an important class of linear time-invarient systems. Strictly speaking, the term filter suggests a system that passes certain frequency components and totally rejects all others, but in a broader sense any system that modifies certain frequencies relative to others is also called a filter[6]. The filter used in discrete-time systems are digital filters. Digital filters operate on digital inputs and generate digital outputs.

2.1 Classification of digital filters

Digital filters can be classified according to the length of their impulse response. FIR (Finite impulse response) and IIR (Infinite impulse response) are the two types of digital filters which have finite and infinite length/ duration of impulse response respectively. FIR and IIR filters can be classified into recursive and non-recursive filters according to their structure. Recursive filters depend on previous output values (having feedback with poles and zeros in their system function) whereas non-recursive filters never depend on previous output values (no feedback with zeros and poles only at $z = 0$ or $z = \infty$ in their transfer function). They can be said the memoryless filters. FIR filters are also called tapped delay line or transversal filters when the coefficients of the FIR filters are automatically changed by means of a step- by - step process, that minimizes r.m.s. difference between the received signal and the desired signal. According to the degrees of freedom, FIR filters are classified into 1-D (1-dimentional) and 2-D (2-dimentional) filters. The spectrum of the 2-D digital filter is shown below:

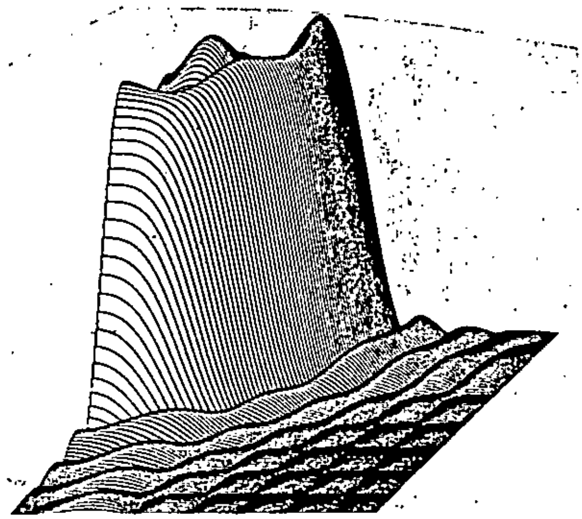

Figure 2.1: 2-D digital filter (low-pass)[9].

In describing digital filters, one usually comes across the following specially named digital filters.

Analysis filters:

In the digital system the filters which split/divide the frequency band of the signal into several frequency sub-bands are called analysis filters.

Synthesis filters:

In the digital system the purpose of synthesis filters is to combine the splitted sub-bands of a signal into a single band.

QMF filters:

QMF (quadrature mirror filters) are those filters which are mirror/image to each other and have a cut- off frequency of one- fourth of the sampling frequency. So, low pass and high pass filters having the cut- off frequency stated above, are QMF filters.

Symmetrical half-band FIR filters:

Symmetrical half-band FIR filters have impulse response which are symmetrical and finite in length having one-half (1/2) spectral energy.

Adaptive filters:

Adaptive filters are a type of FIR filters whose coefficients/weights are changed automatically in step-by-step process by using certain algorithms. The LMS (least mean square) adaptive filters automatically change its coefficients/weights to minimize the error between the received signal and the desired signal.

2.2 Structure of digital filters

The block diagram of the FIR and IIR filters are given in Fig. 2.2 and Fig. 2.3:

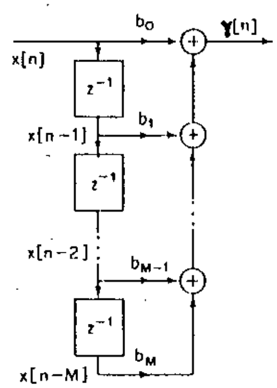

Figure 2.2: Signal flow graph of the FIR filter.

We can express the above FIR and IIR filters by the following difference equations and their corresponding transfer functions.

For FIR filters

$$y(n) = \sum_{k=0}^{M} b_k x(n-k)$$

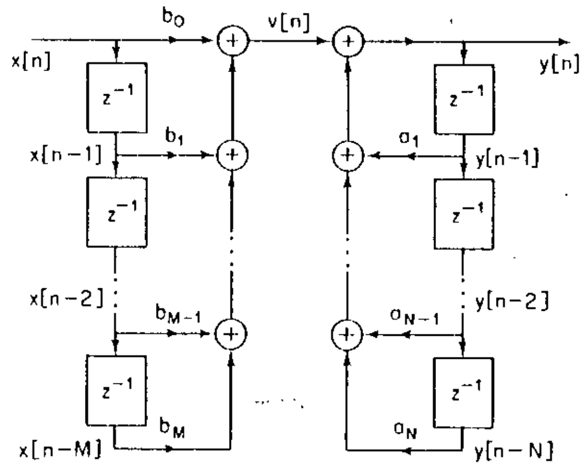

Figure 2.3: Signal flow graph of the IIR filter of the Nth order system.

with the corresponding transfer function

$$H(z) = \sum_{k=0}^{M} b_k z^{-k}$$

For IIR filters

$$y(n) - \sum_{k=1}^{N} a_k y(n-k) = \sum_{k=0}^{M} b_k x(n-k)$$

with the corresponding rational transfer function,

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{(1 - \sum_{k=1}^{N} a_k z^{-k})}$$

IIR filters, for example, have the advantage that they can be designed easily using closed form designed formulas whereas FIR filters has an advantage over IIR filter because of its all-time stability due to the absence of any poles (except at $z = 0$) in their transfer functions.

2.3 Design of digital filters

IIR filters can be designed easily using closed - form designed formulas. That is, once the problem has been specified approximate for a given approximation method (e.g., Butterworth, Chebyshev or elliptic), then the order of the filter that will meet the specifications can be computed and the coefficients (or poles and zeros) of the digital filter can be obtained by straightforward substitution into a set of design equations. Impulse invarience and Bilinear transformation are the two methods for the design of IIR filters. The basis for impulse invarience is to choose an impulse response for the discrete - time filter that is similar in some sense to the impulse response of the continuous - time filter. It is appropriate only for the bandlimited signals due to avoid the severe aliasing distortions. Bilinear transformation, an algebric transformation between the variable s and z, maps causal suitable continuous time filters into causal stable discrete-time filters. Bilinear transformation of analog filters designed by Butterworth, Chebyshev, or Elliptic approximation methods is a standard method for the design of IIR digital filters. It is a non - linear transformation and free from the aliasing effects.

Closed-form design equations do not exist for FIR filters. FIR filters are almost entirely restricted to discrete-time implementations. Consequently the design techniques for FIR filters are based on directly approximating the desired frequency response of the discrete - time system.

The simplest method of FIR filter design is called the window method. This method generally begins with an ideal desired frequency response that can be represented as

$$H_d(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h_d(n)(e^{-j\omega n}),$$

where $h_d(n)$ is the corresponding impulse response sequence, which can be expressed in terms of $H_d(e^{j\omega})$ as

$$h_d(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\omega n})d\omega$$

The impulse response above is non-causal and infinitely long. The most straightforward approach to obtain a causal FIR approximation to such systems is to truncate the ideal response.

The simplest way to obtain a causal FIR filter from $h_d(n)$ is to define a new system with impulse response $h(n)$ given by

$$\begin{aligned} h(n) &= h_d(n), \quad 0 \leq n \leq M \\ &= 0, \qquad otherwise \end{aligned} \tag{2.1}$$

More generally, we can represent $h(n)$ as the product of the desired impulse response and a finite-duration "window" $w(n)$; i.e.,

$$h(n) = h_d(n)w(n),$$

where for simple truncation as in Eq.2.1, the window is the rectangular window

$$\begin{aligned} w(n) &= 1, \quad 0 \leq n \leq M, \\ &= 0, \quad otherwise. \end{aligned} \tag{2.2}$$

It follows from the modulation or windowing theorem that

$$H(e^{j\omega}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\theta})W(e^{j(\omega-\theta)})d\theta \tag{2.3}$$

That is, $H(e^{j\theta})$ is the periodic convolution of the desired ideal frequency response with the Fourier transform of the window.

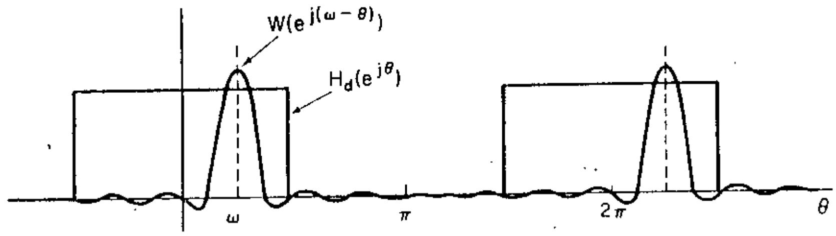

Figure 2.4: Convolution process implied by truncation of the ideal impulse response.

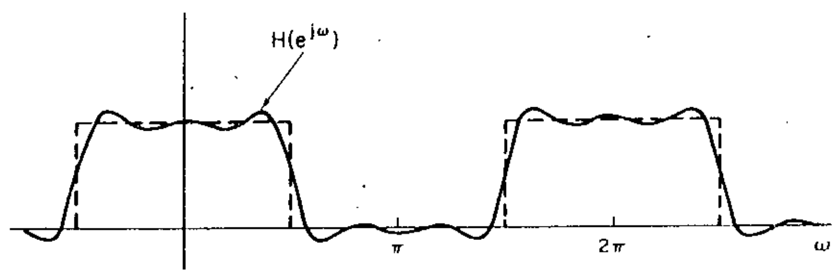

Figure 2.5: Typical approximation resulting from windowing the ideal impulse response.

The choice of window is governed by the desire to have $w(n)$ as short as possible in duration so as to minimize computation in the implementation of the filter while having $W(e^{j\omega})$ approximate an impulse; i.e., we want $W(e^{j\omega})$ to be highly concentrated in frequency so that the convolution of Eq. 2.3 faithfully reproduces the desired frequency response.

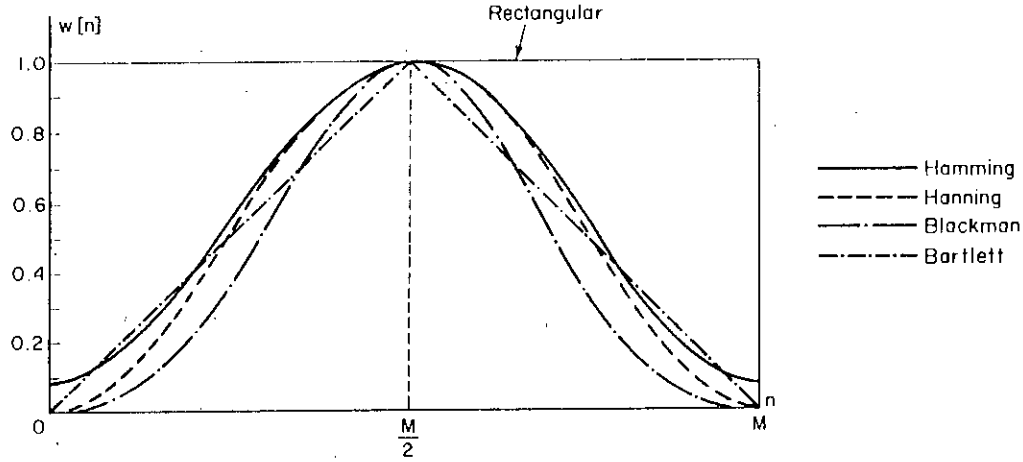

Figure 2.6: Commonly used windows.

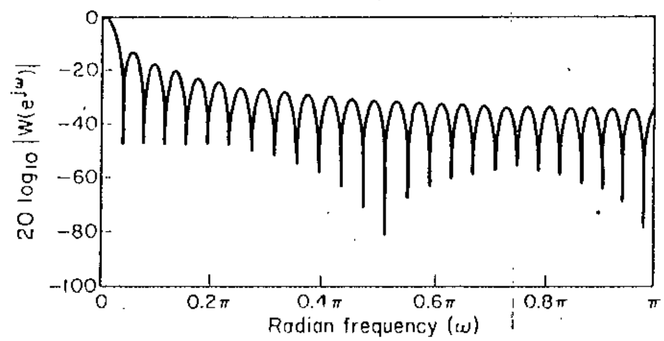

Figure 2.7: Fourier transform of Rectangular window.

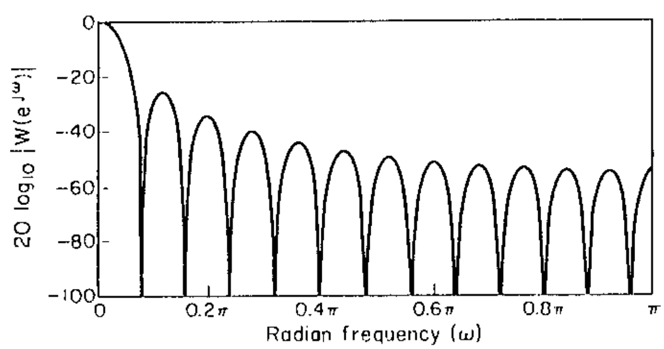

Figure 2.8: Fourier transform of Barlett window.

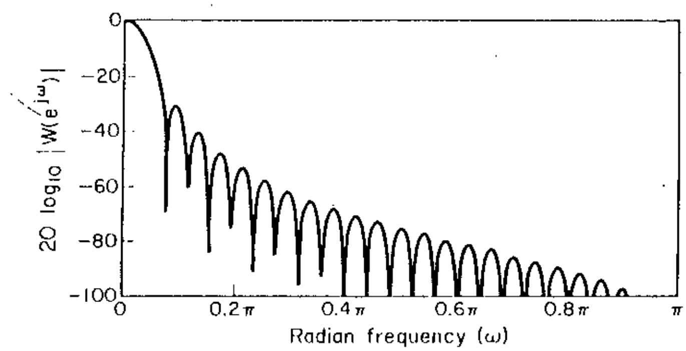

Figure 2.9: Fourier transform of Hanning window.

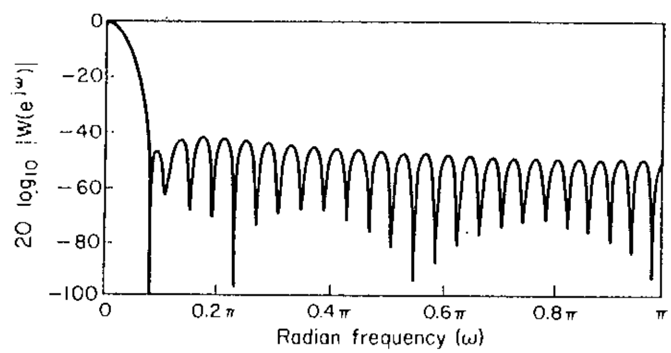

Figure 2.10: Fourier transform of Hamming window.

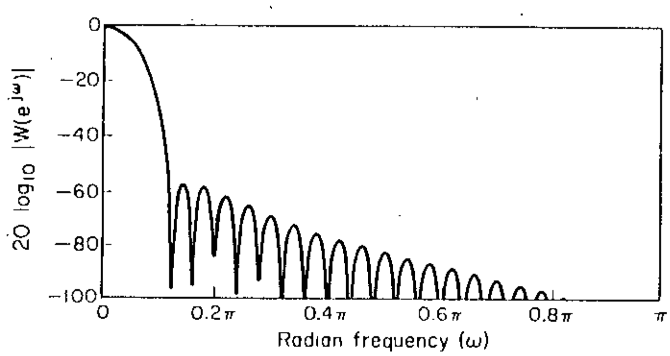

Figure 2.11: Fourier transform of Blackman window.

Window Type	Peak Sidelobe Amplitude (Relative)	Approximate Width of Mainlobe	Peak Approximation Error $20 \log_{10} \delta$ (dB)	Equivalent Kaiser Window β	Transition Width of Equivalent Kaiser Window
Rectangular	-13	$4\pi/(M+1)$	-21	0	$1.81\pi/M$
Bartlett	-25	$8\pi/M$	-25	1.33	$2.37\pi/M$
Hanning	-31	$8\pi/M$	-44	3.86	$5.01\pi/M$
Hamming	-41	$8\pi/M$	-53	4.86	$6.27\pi/M$
Blackman	-57	$12\pi/M$	-74	7.04	$9.19\pi/M$

Figure 2.12: Comparison of commonly used windows.

However Kaiser (1966, 1974) found that a window could be formed using the zeroth-order modified Bessel function of the first kind, a function that is much easier to compute. The Kaiser window is defined as

$$w[n] = \begin{cases} \dfrac{I_0[\beta(1 - [(n-\alpha)/\alpha]^2)^{1/2}]}{I_0(\beta)}, & 0 \le n \le M, \\ 0, & \text{otherwise}, \end{cases}$$

where $\alpha = M(order)/2$, and $I_0(.)$ represents the zeroth - order modified Bessel function of the first kind. In contrast to the other windows, the Kaiser window has two parameters; the length (M + 1) and a shape parameter β. By varying (M + 1) and β, the window length and shape can be adjusted to trade sidelobe amplitude for magnitude width.

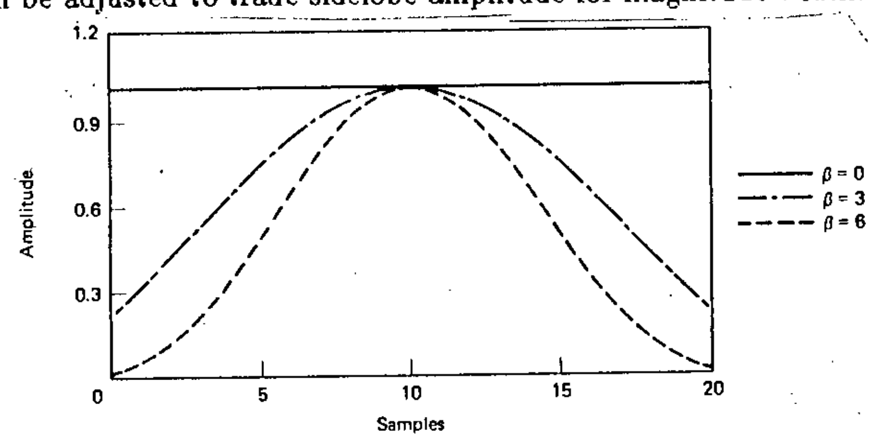

Figure 2.13: Kaiser windows for $\beta = 0, 3$, and 6 and $M = 20$.

The design problem for FIR filters is much more under control than the IIR design problem because of the existance of optimality theorem for FIR filters from which Park-Maclellan (1972

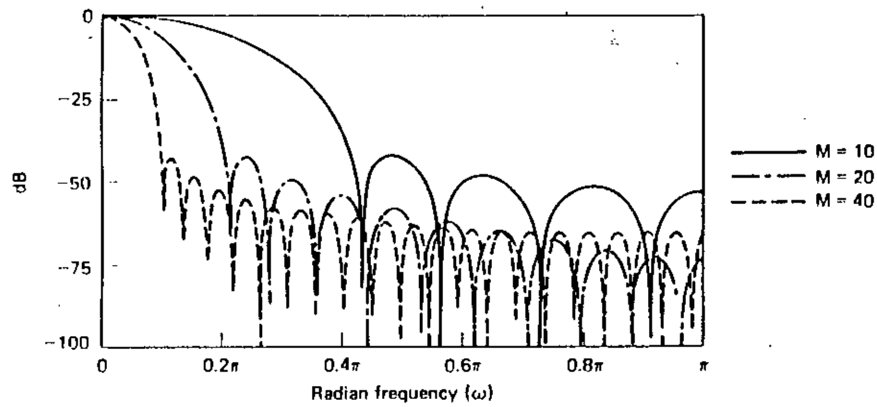

Figure 2.14: Fourier transforms of Kaiser windows with $\beta = 6$ and $M = 10, 20,$ and 40.

)[4] algorithm is used for the design of FIR filter especially lower order filters. Design techniques for FIR filters with out linear phase have been given by Chen and Parks (1987)[12], Parks and Burrus (1987)[13], and Schussler and Steffen (1988)[14].

2.4 Decimation

The operation of reducing the sampling rate is called decimation/downsampling. If the sequence of the signal is $x(n)$, after downsampling by $M(\downarrow M)$,the sequence will become as

$$x_d(n) = x(Mn); n = 0, 1, 2, ...n$$

which means that it takes every m-th sample value.

The downsampling causes the overlapping of the signal if the signal is not sufficiently band-limited.

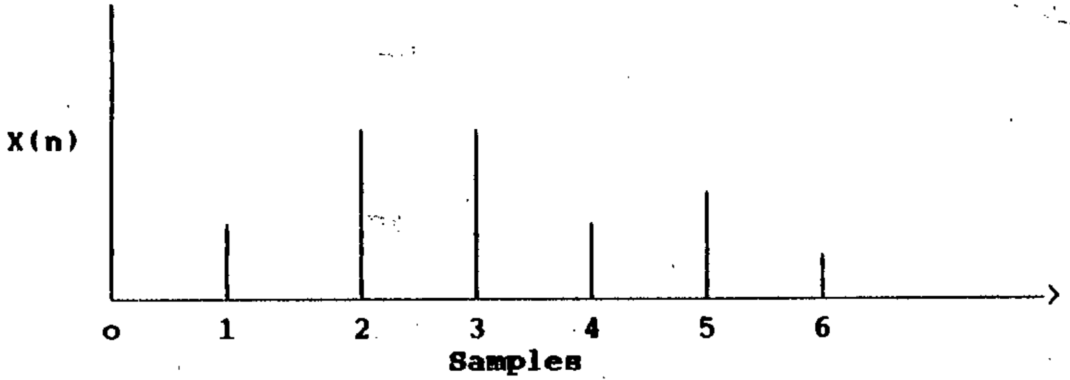

Figure 2.15: Sequence of signal before downsampling.

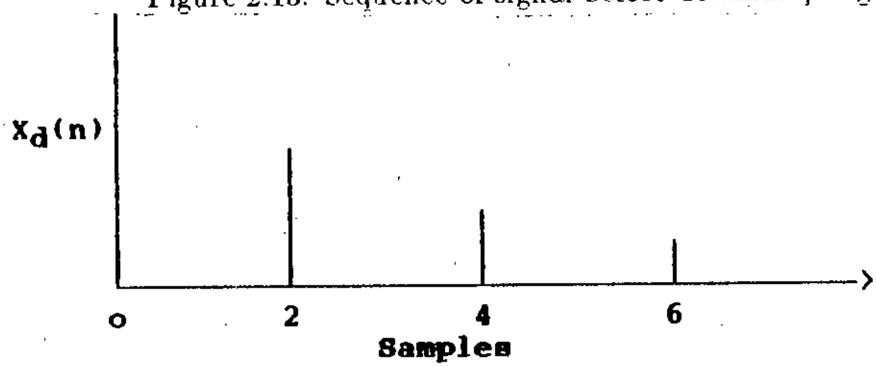

Figure 2.16: Sequence of signal after downsampling (by 2).

2.5 Interpolation

The operation of increasing the sampling rate will be called interpolation/ upsampling. If the sequence of the signal is $x(n)$, after upsampling by $L(\uparrow L)$, the sequence will be as follows:

$$\begin{aligned} x_i(n) &= x(n/L), \quad n = 0, L, 2L \\ &= 0, \qquad\quad \textit{otherwise} \end{aligned}$$

Ω Interpolation/upsampling relates to the digital to analog (D/A) conversion during the reconstruction of the signal.

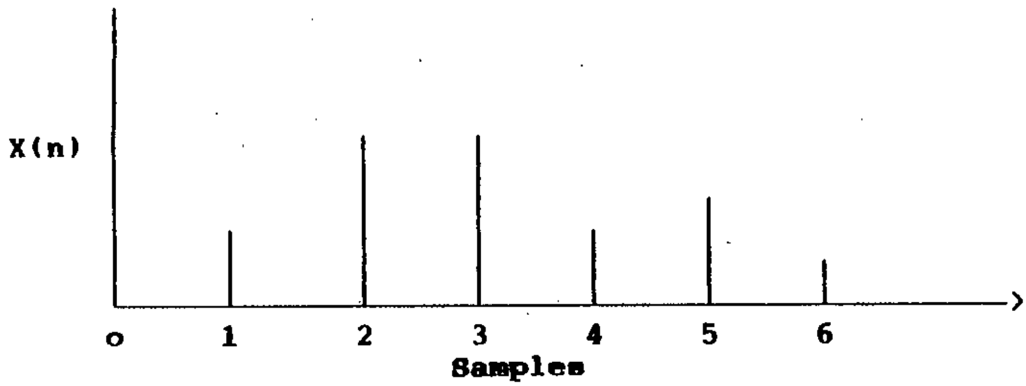

Figure 2.17: Sequence of signal before upsampling.

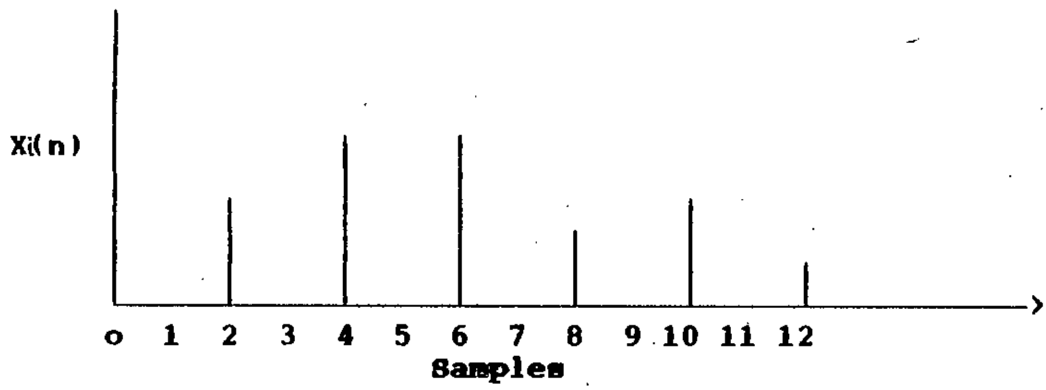

Figure 2.18: Sequence of signal after upsampling.

2.6 z-transform

The z-transform for the discrete-time signals is the counter-part of the Laplace transform for continuous-time signals, and they each have a similar relationship to the corresponding Fourier transform. The z-transform has several advantages over Fourier transform. Fourier transform does not converge for all sequences like the z-transform. A second advantage is that in analytical problems the z-transform notation is often more convenient than the Fourier transform notation.

The Fourier transform $X(e^{j\omega})$ of a sequence x(n) is defined as

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n}$$

The z- transform $X(z)$ of a sequence $x(n)$ is defined as

$$Z[x(n)] = X(z) = \sum_{n=-\infty}^{\infty} x(n) z^{-n} \tag{2.4}$$

The correspondence between a sequence and its z-transform is indicated by the notation

$$x(n) \overset{z}{\longleftrightarrow} X(z)$$

The z-transform as we have defined it in Eq. 2.4 with the complex variable $e^{j\omega}$, then the z-transform reduces to the Fourier transform.

2.7 Applications of digital filters

In digital signal processing techniques digital filters are widely used. It is useful in many important areas such as speech signal processing, digital telephony and communications, facsimile and TV image processing, radar - soner systems, biomedicine, space-research and geoscience, etc. 1-D and 2-D digital filters and adaptive filters are used in several specific areas stated above. They can be applied in many and different parts of signal and image processiong transmission systems [9] .

Two typical examples of systems in which digital filtering can be inserted are:

- Local digital processing system.
- Digital communication system.

The 1-D or 2-D digital filter in the first system of Fig.2.19 help to get some specific frequencies by using suitable transfer functions. In the second system of Fig. 2.20, the 1-D or 2-D digital filters are required to select and delimit the bandwidths or to perform suitable corrections on the signals.

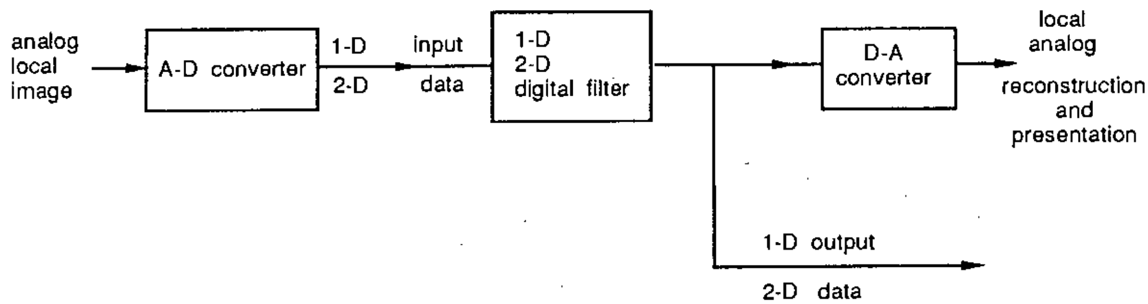

Figure 2.19: General structure of a local digital processing system.

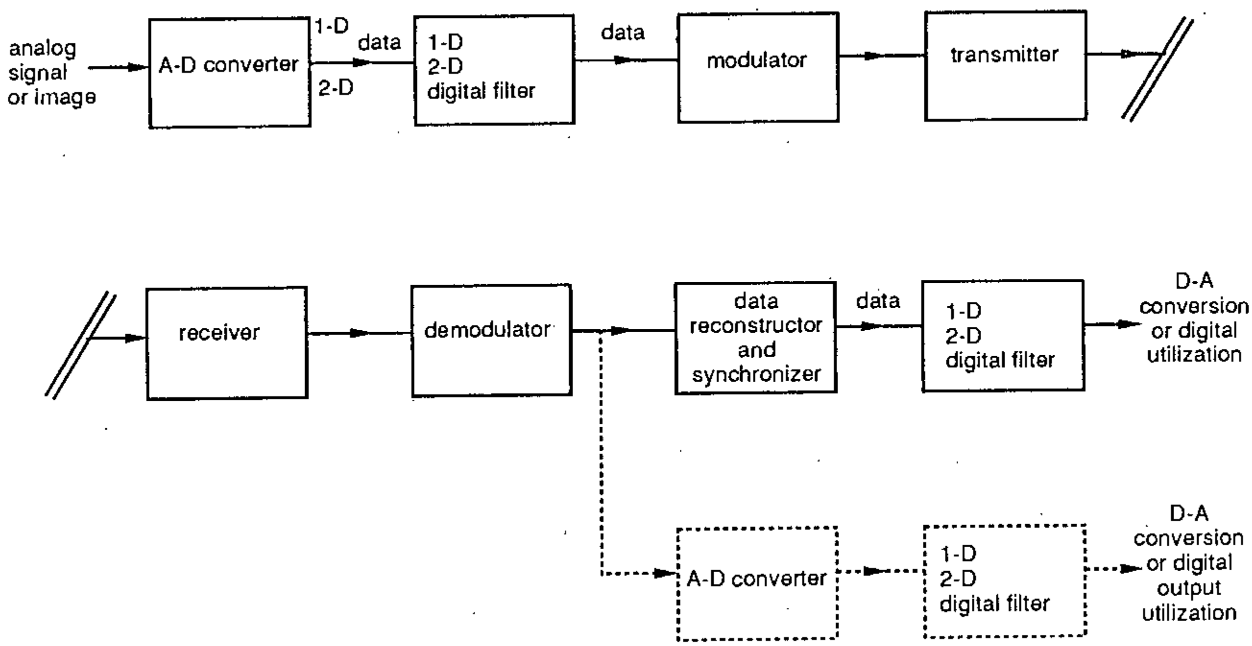

Figure 2.20: General structure of a digital communication system.

Chapter 3

An exact reconstruction digital filter bank

3.1 Basic structure of the digital filter bank

The two channel digital filter bank, shown below, has two individual sections which may be or may not be separated by the processor according to our demand. The first part is the analyser and the second part is the synthesizer. The analyser has two QMF (quadrature mirror filters) $H_0(z)$ and $H_1(z)$, a low pass and a high pass filter respectively, followed by a decimator. The purpose of $H_0(z)$ and $H_1(z)$ is to separate the input signal into two frequency bands (lowpass and highpass), being as independent as possible. The processor has suitable processing, according to the application.

The synthesizer has two interpolators followed by the two filters $G_0(z)$ and $G_1(z)$ which eliminate aliasing caused by the decimator-interpolator system. The filters used in the filter bank are FIR (finite impulse response) filters. The filters (the analysis and synthesis filters) should be symmetrical, even and linear phase. Then it is possible to have an output which is a delayed replica of the input. Supposing that the input is an impulse, the output ideally

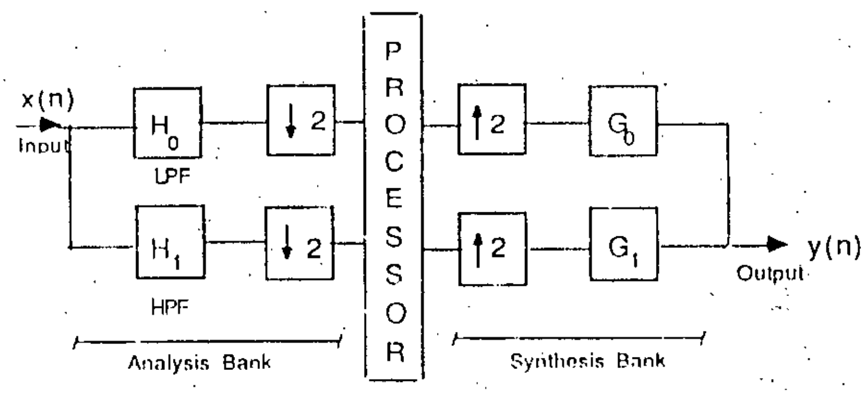

Figure 3.1: Two channel digital filter bank

is a delayed impulse. The following figure shows the frequency response of the input signal schematically at every step of the filter bank. It is a maximally decimated 2-channel quadrature mirror filter (QMF) bank. The analysis filters $H_k(z)$ split the input signal into 2 subband signals which in turn are decimated by 2 in the analysis bank and then transmitted. At the receiving end, the 2 subbands signals are interpolated and recombined using the synthesis filters $G_k(z)$ in the synthesis bank. Aliasing effects due to decimation are cancelled by the imaging effects due to interpolation and so that a perfectly reconstructed output signal has found.

3.2 Conditions for exact reconstruction

We want to show the conditions for the perfect reconstruction of the signal. If we take any path of the filter bank, we find a decimation there. Decimation/ downsampling relates to the operation of reducing the sampling rate of the sequence. Decimation causes a compression in the time domain and stretching in the frequency domain. If the input signal $x(n)$ is decimated by 2 then we can write the result in Z- domain (output $Y(z)$) [3].

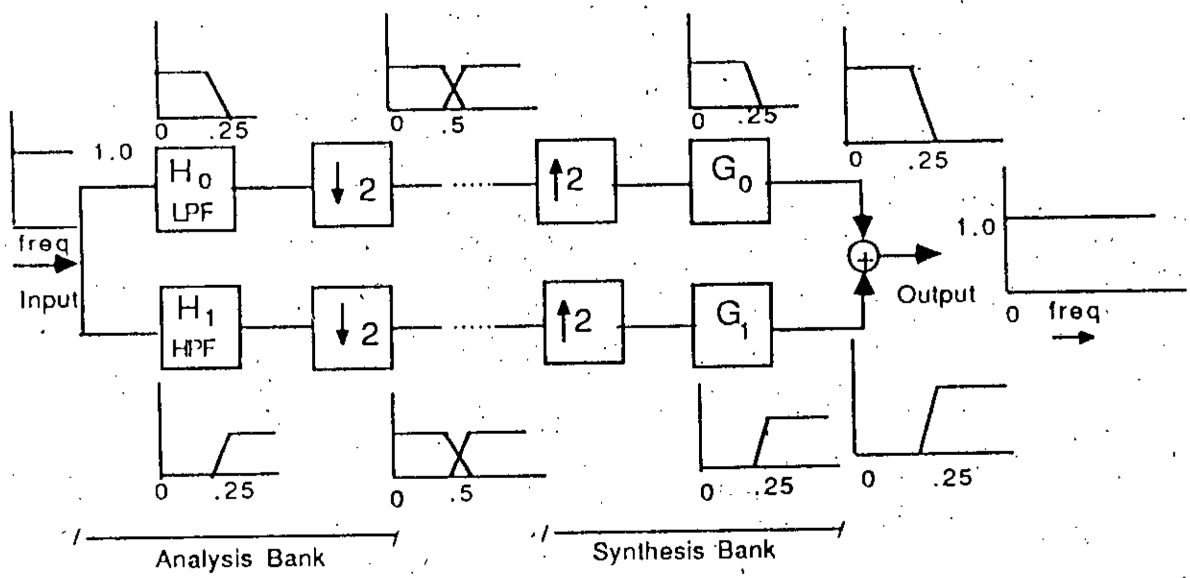

Figure 3.2: Filter bank with frequency functions for the signals and filters.

$$Y(z) = \frac{1}{2}\left[X(z^{\frac{1}{2}}) + X(-z^{\frac{1}{2}})\right] \tag{3.1}$$

The first term in the above expression is $X(z^{\frac{1}{2}})$, which is a stretched version of $X(z)$ and the second term is $X(-z^{\frac{1}{2}})$, which is a stretched version of the frequency shifted version of $X(z)$ (shifted by amount π). In the path there is also interpolation. Interpolation/upsampling is opposite to decimation/downsampling. It relates to the operation of increasing the sampling rate. It causes stretching in the time domain and a compression in the frequency domain. If the signal $x(n)$ is interpolated by 2 then in Z- domain, we can write

$$Y(z) = X(z^2) \tag{3.2}$$

$Y(z)$ has no aliasing term and $X(z^2)$ is the compressed version of $X(z)$.

Ω

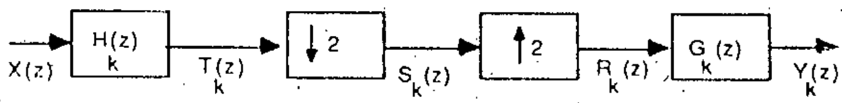

Figure 3.3: Calculation of conditions for exact reconstruction

Let in Z- domain $X(z)$ and $H_k(z)$ mean the transfer function of the input signal and the analysis filter banks respectively (see Fig. 3.3). After filtering the output $T(z)$ becomes the product of $X(z)$ and $H_k(z)$.

$$T_k(z) = X(z)H_k(z) \tag{3.3}$$

Then $T_k(z)$ is decimated by 2 and according to Eq.3.1, output $S_k(z)$ can be calculated as

$$S_k(z) = \frac{1}{2}\left[X(z^{\frac{1}{2}})H_k(z^{\frac{1}{2}}) + X(-z^{\frac{1}{2}})H_k(-z^{\frac{1}{2}})\right] \tag{3.4}$$

When the signal come to the synthesizer, it is upsampled by 2. According to Eq.3.2 the output $T_k(z)$ after upsampling becomes:

$$R_k(z) = \frac{1}{2}[X(z)H_k(z) + X(-z)H_k(-z)] \tag{3.5}$$

Let $G_k(z)$ is the transfer function of the synthesis filter. Then after final filtering the output $Y_k(z)$ of the filter bank is the product of $R_k(z)$ and $G_k(z)$.

$$Y_k(z) = \frac{1}{2}[X(z)H_k(z) + X(-z)H_k(-z)]\,G_k(z) \tag{3.6}$$

If we simplify the above expression for a two path filter bank, we can get the output as follows:

$$\begin{aligned} Y(z) = Y_0(z) + Y_1(z) \;&= \tfrac{1}{2}\left[H_0(z)G_0(z) + H_1(z)G_1(z)\right]X(z) \\ &+ \tfrac{1}{2}\left[H_0(-z)G_0(z) + H_1(-z)G_1(z)\right]X(-z) \end{aligned} \tag{3.7}$$

The second term stands for aliasing. Under the following conditions the second term becomes zero, which means that there is no aliasing effect in the over-all system.

$$\begin{aligned} H_1(z) &= H_0(-z) \\ G_0(z) &= H_0(z) \\ G_1(z) &= -H_1(z) \end{aligned} \tag{3.8}$$

and the overall transfer fuction becomes

$$\frac{Y(z)}{X(z)} = \frac{1}{2}\left[H_0(z)G_0(z) + H_1(z)G_1(z)\right] = \frac{1}{2}\left[H_0^2(z) + (-H_0^2(-z))\right] \tag{3.9}$$

For perfect reconstruction the magnitude of the overall transfer function must be unity. This is achieved by the following demand, written for the frequency function of the filter:

$$\mid H_0(\nu) \mid^2 + \mid H_0(\nu - \frac{1}{2}) \mid^2 = 2 \tag{3.10}$$

where $\nu = \frac{\omega}{2\pi}$ is the normalized frequency.

3.3 The delay in the filter bank

In this section the delay in the filter bank is calculated. The delay in the two channel filter bank is equal to the order N of the filter[1]. We can verify this by using the Fourier transform. Let $H(z)$ be a symmetrical half-band FIR filter. Then the impulse response $h(n)$ satisfies the following relation:

$$\begin{aligned} h(n) &= h(N-n), \quad 0 \le n \le N \\ &= 0, \qquad\qquad \text{otherwise} \end{aligned} \tag{3.11}$$

The Fourier transform $H(e^{j\omega})$ can be expressed in terms of its magnitude $H(\omega)$ [3]:

$$H(e^{j\omega}) = H(\omega)e^{(-j\omega N/2)}$$

Substituting in Eq. 3.9 and letting $z = e^{j\omega}$ (for unit circle) gives:

$$\begin{aligned} Y(e^{j\omega}) &= \tfrac{1}{2}\left[H^2(e^{j\omega}) - H^2(-e^{j\omega})\right] X(e^{j\omega}) \\ &= \tfrac{1}{2}\left[H^2(e^{j\omega}) - H^2(e^{j(\omega+\pi)})\right] X(e^{j\omega}) \\ &= \tfrac{1}{2}\left[H^2(\omega) - H^2(\omega+\pi)\exp^{-j\pi N}\right] e^{-j\omega N} X(e^{j\omega}) \end{aligned} \tag{3.12}$$

We can show the following two cases from the expression of $Y(e^{j\omega})$.

If the order of the filter N is odd, i.e. the length $L(= N+1)$ is even, then the expression $Y(e^{j\omega})$ becomes:

$$\begin{aligned} Y(e^{j\omega}) &= \tfrac{1}{2}\left[H^2(\omega) + H^2(\omega+\pi)\right] e^{-j\omega N} X(e^{j\omega}) \\ &= X(e^{j\omega})e^{-j\omega N} \end{aligned}$$

In Z-transform the above final expression becomes:

$$Y(z) = X(z)z^{-N} \tag{3.13}$$

or in time domain $y(n) = x(n - N)$

If N is even, i.e. the length L (=N+1) is odd, then $Y(e^{j\omega})$ becomes:

$$Y(e^{j\omega}) = \frac{1}{2}\left[H^2(\omega) - H^2(\omega+\pi)\right] e^{-j\omega N} X(e^{j\omega})$$

In this case the original signal can not be perfectly constructed because $[H^2(\omega) - H^2(\omega + \omega_s/2)] \neq 2$ or can not be constant for all ω. At $\omega = \omega_s/4$, the amplitude is always zero.

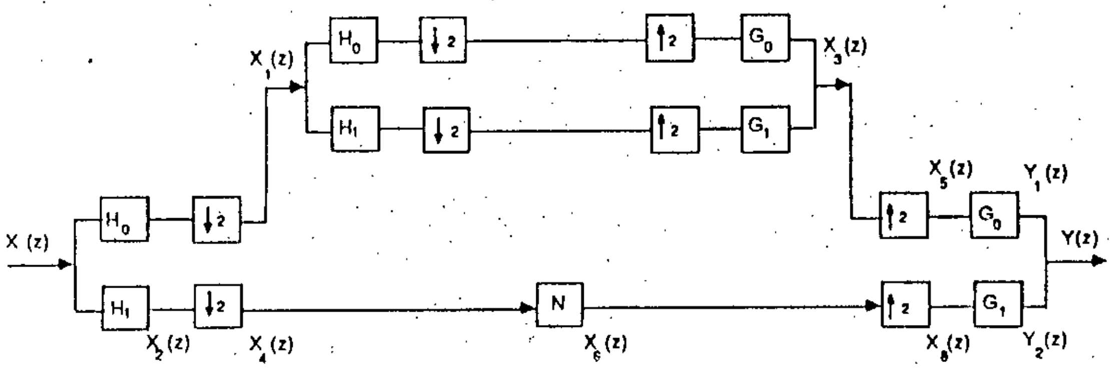

Figure 3.4: Calculation of the delay

We can write according to Eq.3.9 with notations from Fig.2.4

$$X_3(z) = \frac{1}{2}\left[H_0^2(z) - H_0^2(-z)\right] X_1(z) \tag{3.14}$$

where

$$X_1(z) = \frac{1}{2}\left[H_0(z^{\frac{1}{2}})X(z^{\frac{1}{2}}) + H_0(-z^{\frac{1}{2}})X(-z^{\frac{1}{2}})\right] \tag{3.15}$$

Combining Eqs.(3.2), (3.10) and (3.14), we get

$$\begin{aligned} Y_1(z) &= G_0(z)\cdot X_5(z) \\ &= G_0(z)\cdot \tfrac{1}{2}\left[H_0^2(z^2) - H_0^2(-z^2)\right]\cdot \tfrac{1}{2}\left[H_0(z)X(z) + H_0(-z)X(-z)\right] \end{aligned} \tag{3.16}$$

Since the product filters have a linear phase, the phase of $[H_0^2(z) - H_0^2(-z)]$ is $2\pi\nu N$, where N is the order of the filter. That is [1]

$$\frac{1}{2}\left[H_0^2(z) - H_0^2(-z)\right] = [1]\, z^{-N} \tag{3.17}$$

From Fig.3.3,

$$X_8(z) = X_6(z^2) = X_4(z^2)z^{-2N} \tag{3.18}$$

According to Eq. (3.1)

$$X_4(z) = \frac{1}{2}\left[X_2(z^{\frac{1}{2}}) + X_2(-z^{\frac{1}{2}})\right]$$

and Eq.(3.18) yields

$$X_8(z) = \frac{1}{2}\left[H_1(z)X(z) + H_1(-z)X(-z)\right]z^{-2N}$$

Therefore ,

$$\begin{aligned} Y_2(z) &= G_1(z)\cdot X_8(z) \\ &= \tfrac{1}{2}G_1(z)\left[H_1(z)X(z) + H_1(-z)X(-z)\right]z^{-2N} \end{aligned} \tag{3.19}$$

Adding Eq. (3.16) , (3.19) and then simplifying,

$$\begin{aligned} Y(z) &= \tfrac{1}{2}\left[G_0(z)\cdot\tfrac{1}{2}(H_0^2(z^{4}) - H_0^2(-z^{4}))H_0(z) + H_1(z)G_1(z)z^{-2N}\right]X(z) \\ &+\tfrac{1}{2}\left[G_0(z)\cdot\tfrac{1}{2}(H_0^2(z^{4}) - H_0^2(-z^{4}))H_0(-z) + H_1(-z)G_1(z)z^{-2N}\right]X(-z) \end{aligned} \tag{3.20}$$

Using Eq.(3.17), Eq.(3.20) becomes

$$\begin{aligned} Y(z) &= \tfrac{1}{2}\left[H_0(z)G_0(z) + H_1(z)G_1(z)\right]z^{-2N}X(z) \\ &+\tfrac{1}{2}\left[H_0(-z)G_0(z) + H_1(-z)G_1(z)\right]z^{-2N}X(-z) \end{aligned} \tag{3.21}$$

From the relations in Eq. (3.8), Eq. (3.21) finally becomes

$$\begin{aligned} Y(z) &= \tfrac{1}{2}\left[H_0^2(z) - H_0^2(-z)\right]z^{-2N}X(z) \\ &= X(z)z^{-3N} \end{aligned} \tag{3.22}$$

Refering to the Eq.(3.22), the delay in the filter bank is 3 times order of the filter. Similarly,

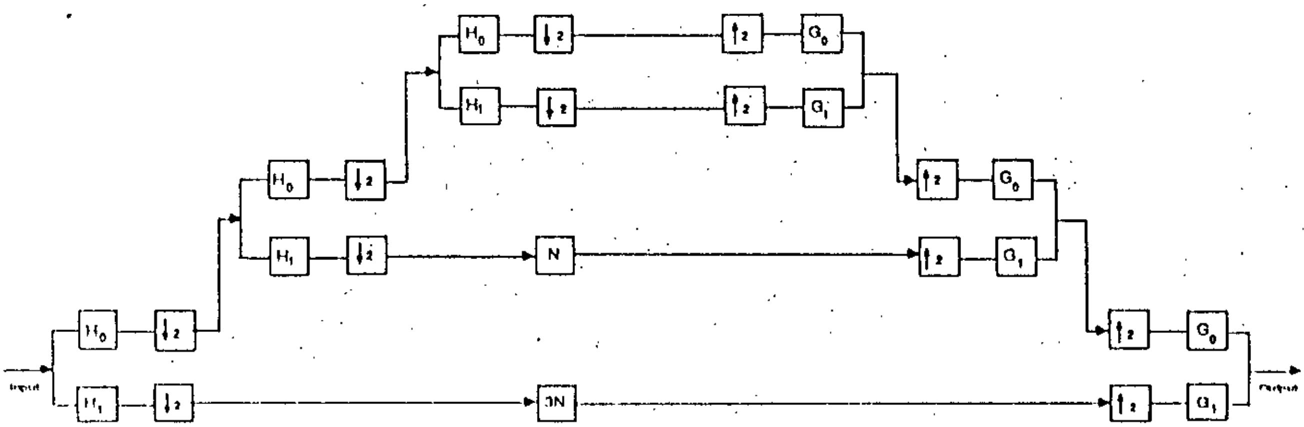

Figure 3.5: Calculation of the total delay for the four channel filter bank

taking Eq.(2.2), (2.14), (3.22), (3.17) and then simplifying,

we get with notations from Fig 3.5,

$$\begin{aligned} Y_3(z) &= G_0(z) \cdot Y_1(z) \\ &= \tfrac{1}{2}\left[H_0(z)\,X(z) + H_0(-z)\,X(-z)\right] G_0(z) z^{-6N} \end{aligned} \tag{3.23}$$

and from Eq.(2.2), (2.14)

$$\begin{aligned} Y_6(z) &= G_1(z) \cdot Y_4(z) \\ &= \tfrac{1}{2}\left[H_1(z)\,X(z) + H_1(-z)\,X(-z)\right] G_1(z) z^{-6N} \end{aligned} \tag{3.24}$$

Adding Eq.(3.23), (3.24) and simplifying, we get

$$\begin{aligned} X(z) &= \tfrac{1}{2}\left[H_0(z)\,G_0(z) + H_1(z)\,G_1(z)\right] z^{-6N} X(z) \\ &+ \tfrac{1}{2}\left[H_0(-z)\,G_0(z) + H_1(-z)\,G_1(z)\right] z^{-6N} X(-z) \end{aligned} \tag{3.25}$$

Using Eq.(3.8) and (3.17) finally we get

$$\overline{X}(z) = z^{-7N} X(z) \tag{3.26}$$

Referring to Eq.(3.26), the total delay for the four channel filter bank is 7 times order of the filter. More general, we can find the following expression for the delay.

$$\Delta = (2^L - 1)N \tag{3.27}$$

where Δ and N stand for delay and the order of the filter respectively and $L = 1, 2, 3, 4, 5, \ldots$ represents the depth of the filter bank.

3.4 Using the filter bank as a selector of sub-bands

The four channel digital filter bank splits an input signal into four different frequency sub-

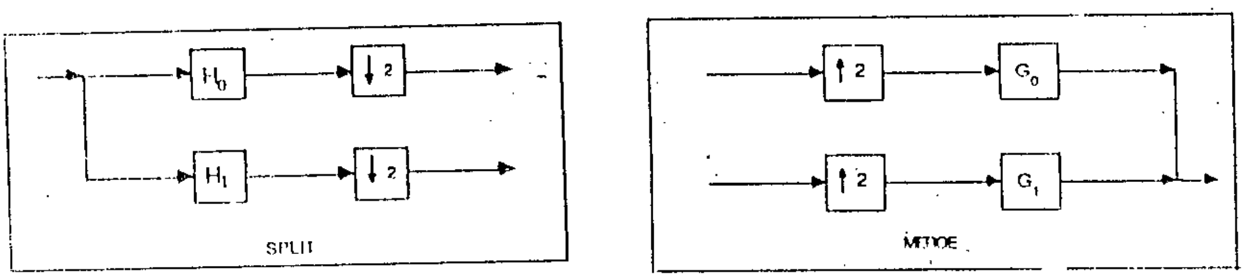

Figure 3.6: Defination of the 'SPLIT' and 'MERGE'.

bands in the analyser and recombine these four sub-bands into a single band in the synthesizer, but one can easily select one or more frequency sub-bands at the output by using a selector. The filter bank then acts as a sub-band selector. Defining the analyser and synthesizer by 'SPLIT'and 'MERGE'. (see Fig. 3.6), sub-band selector can be represented as in Fig.3.7.

If an unit impulse is used as an input to the sub-band selector and all bands are selected, the output will be a delayed impulse which is shown in Fig. 3.8. The delay of 889 samples in Fig.3.9 certifies our calculation of total delay for the four channel filter bank which is 7 times the order (N) of the filter. Fig. 3.10 shows the output when the input is an impulse and the order N is even.

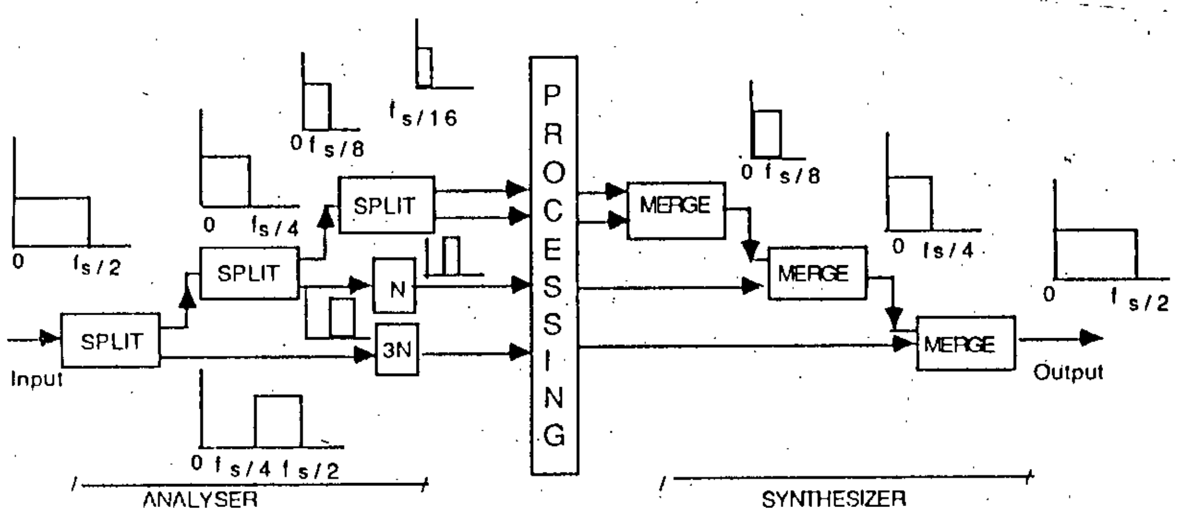

Figure 3.7: Subband selecter filter bank.

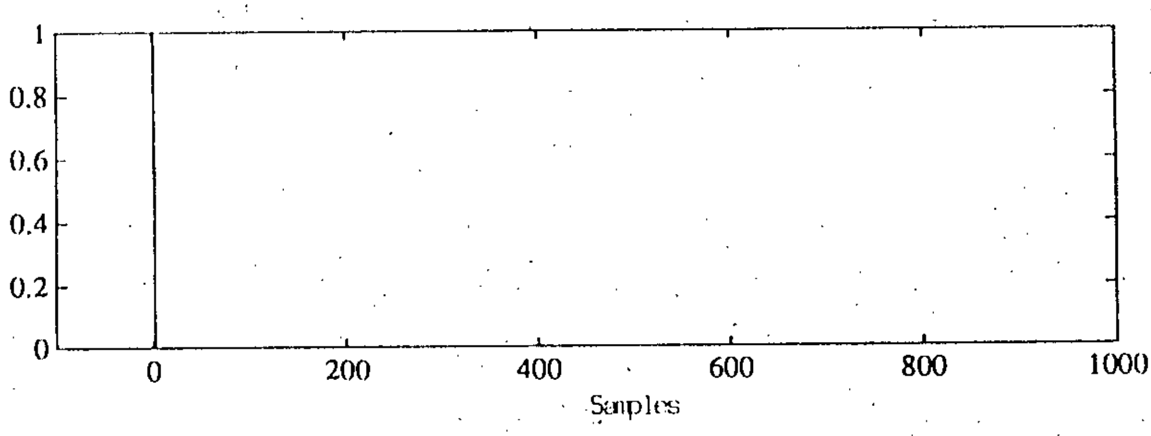

Figure 3.8: Unit Impulse.

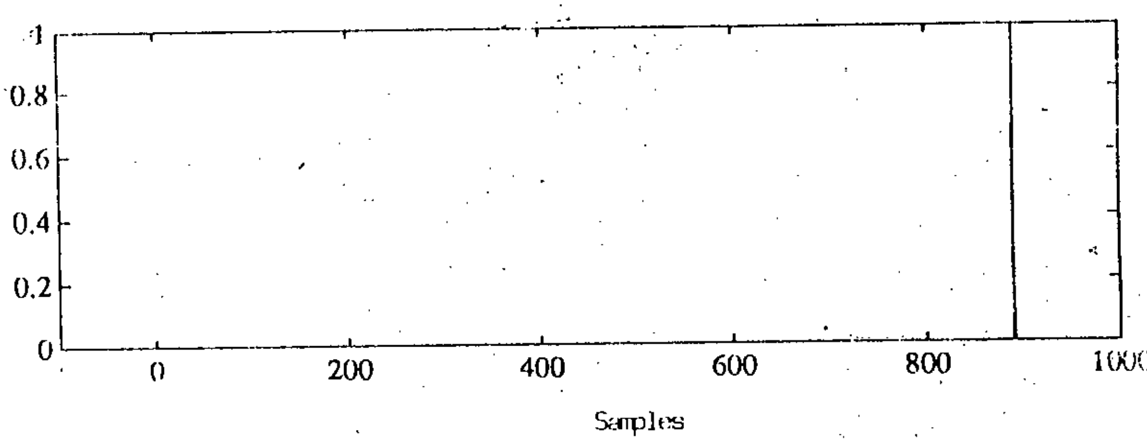

Figure 3.9: Delayed unit impulse.

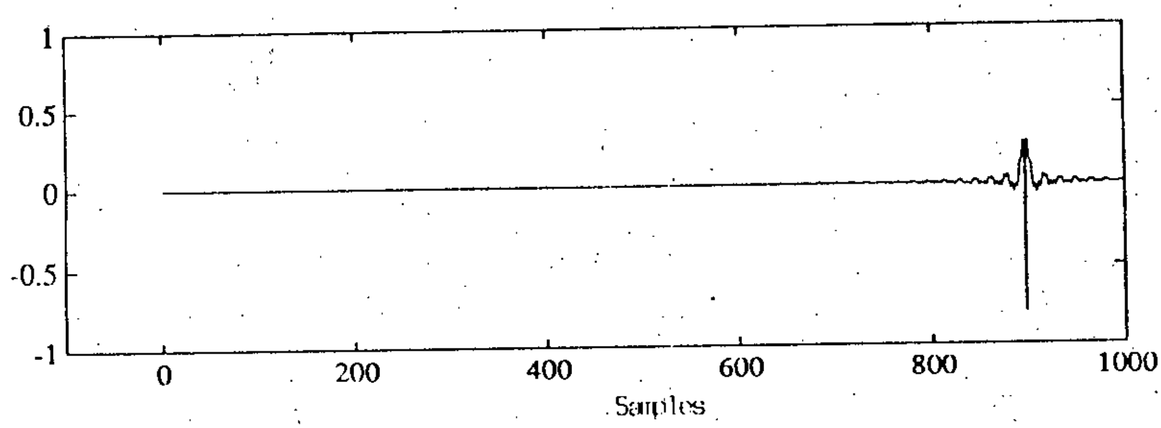

Figure 3.10: Output for the odd length filters.

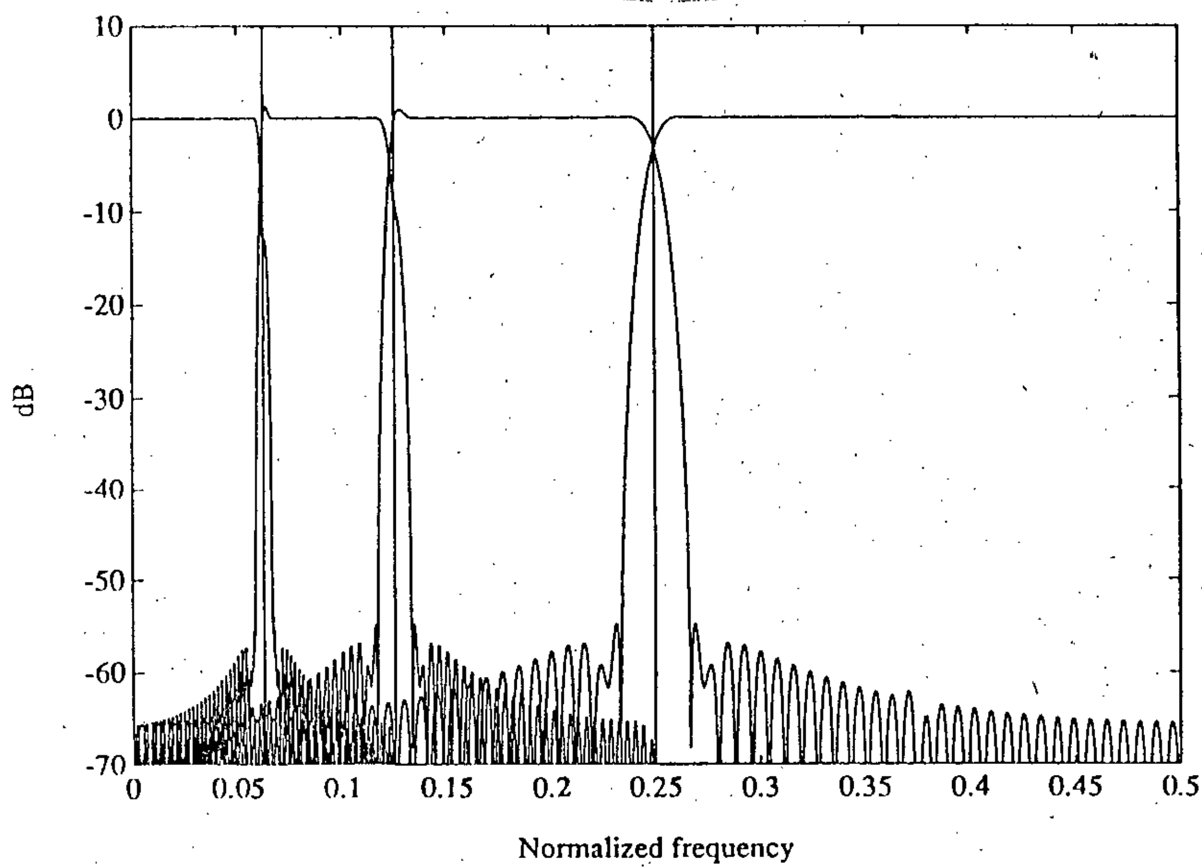

Figure 3.11: Frequency subbands selected from the subband selector filter bank.

Fig. 3.11 shows the different frequency subbands in the subband selector filter bank. By using different gain in the selector paths, one or more subbands of different magnitude can be selected. If we consider the described four channel filter bank, the first to last subbands in the figures are related to the output of different channels of the filter bank which are low pass, band pass, band pass and high pass in nature respectively.

Fig. 3.12 and 3.13 show the speech signal and its delayed output when all channels are selected. Then it works like an all-pass system. Fig. 3.14 - 3.17 show outputs of the speech signal when low pass, band pass, band pass and high pass channels are selected respectively. It means that one input signal can be splitted into four output signals of different frequency relating to the channels in the filter bank.

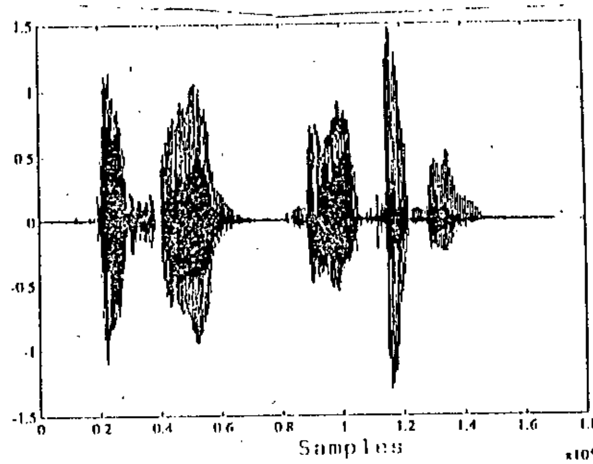

Figure 3.12: Speech signal.

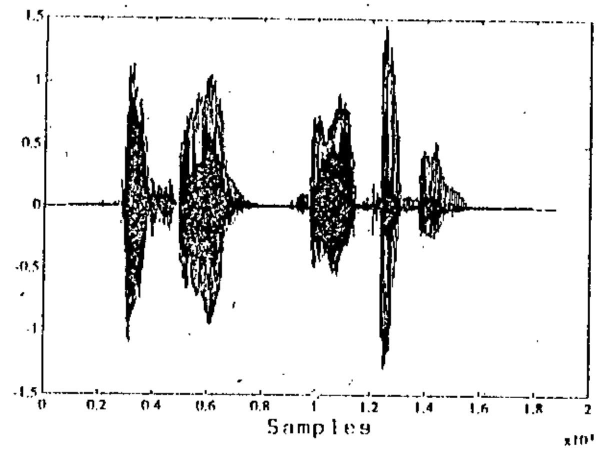

Figure 3.13: Delayed speech signal.

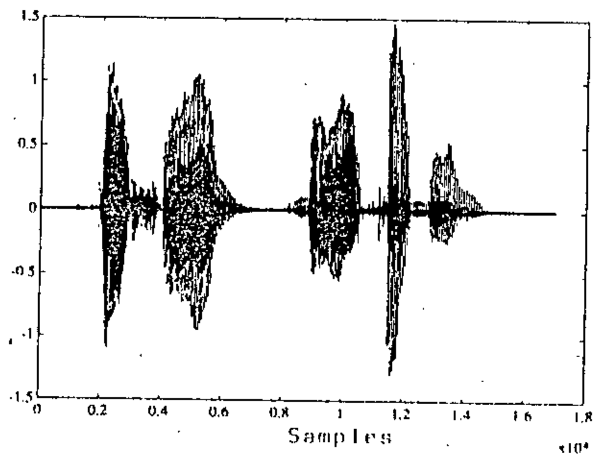

Figure 3.14: Low pass signal.

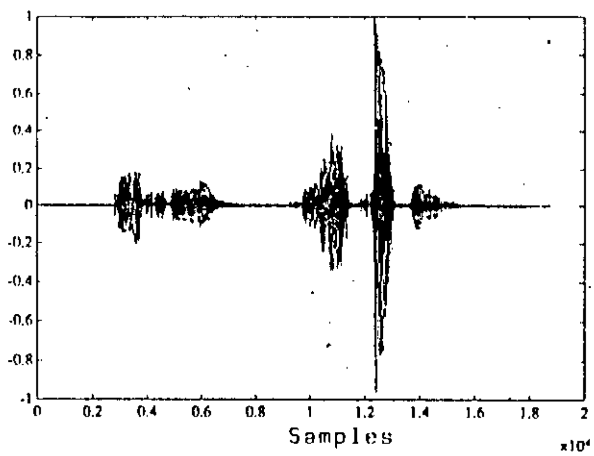

Figure 3.15: Band pass signal.

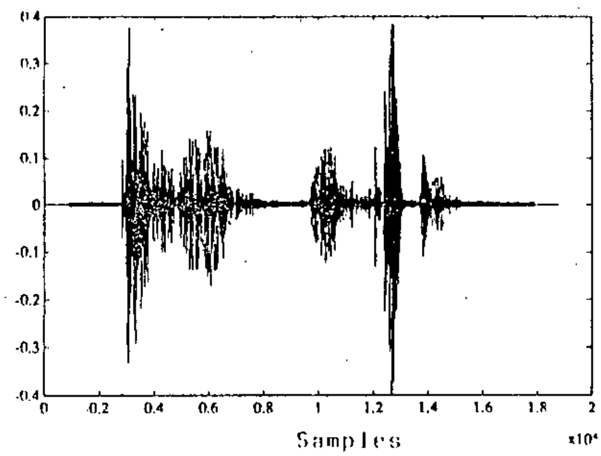

Figure 3.16: Band pass signal.

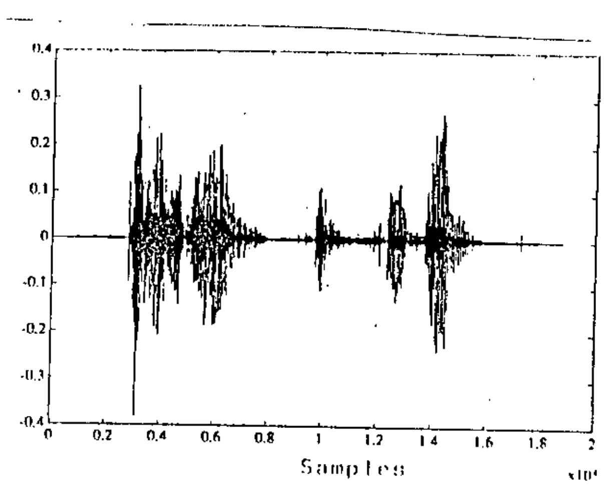

Figure 3.17: High pass signal.

Chapter 4

Filter bank in echo cancelling

4.1 Principle of echo canceller

When we transmit speech through the telephone line, the primary problem is undesired feed through of the transmitted speech into the receiver through the hybrid. The extraneous signal is called echo. The echo canceller is an adaptive finite impulse response (FIR) / transversal filter that adaptively learns the response of the hybrid filter, and generates a replica of the response which is subtracted from the hybrid output to yield an echo-free received signal.

This can be explained by using local transmitter and local receiver as follows:

The local transmitter signal y(t) at port A generates the undesired echo signal r(t). This signal is superimposed at the output of the hybrid (port D) with the far transmitter signal x(t). The canceller generates a replica of the echo, $\hat{r}(t)$ from the knowledge of local transmitter signal. This replica is subtracted from the echo plus far transmitter signal to yield e(t), which ideally contains the far transmitter signal x(t) alone.

The echo canceller is usually implemented in discrete-time as a FIR filter. Fig.4.3 is a block

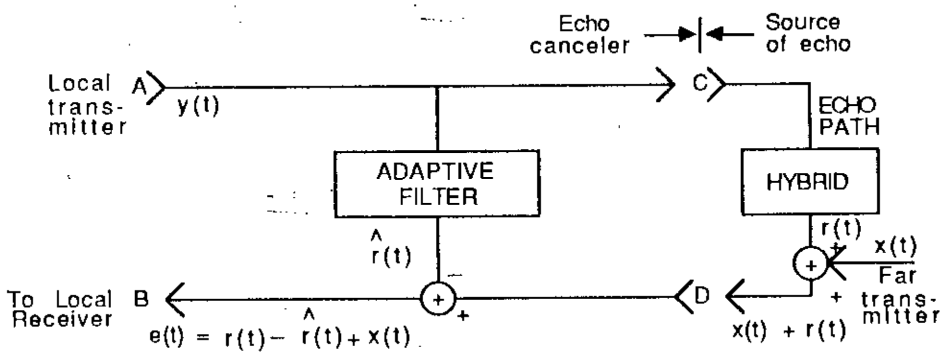

Figure 4.1: The principle of an echo canceller.

diagram of the common structure of an echo canceller.

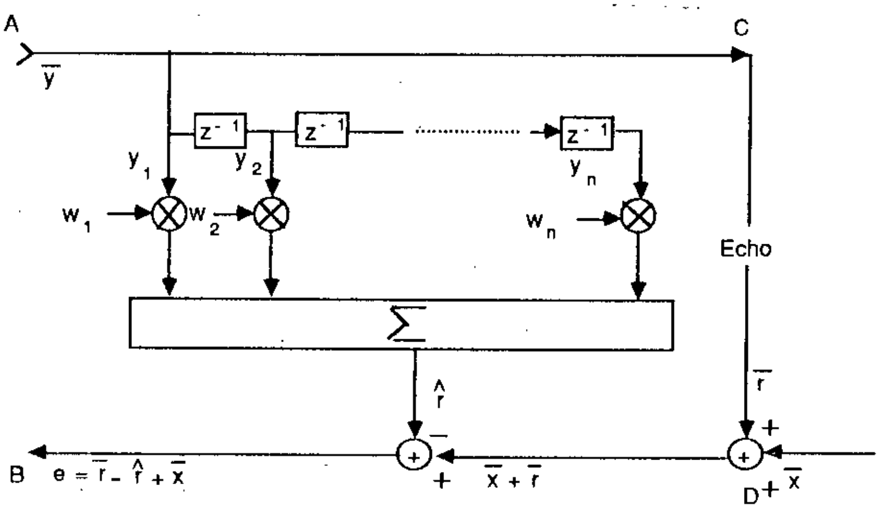

Figure 4.2: A transversal filter echo canceller.

4.2 Echo canceller using the filter bank

Following figure shows the block diagram of an echo canceller using the filter bank. The filter bank echo canceller can be divided into two sections. One section of the echo canceller consists of two parallel paths. One path has the analysis bank of the filter bank and the set of transversal adaptive filters respectively. Another path consists of a hybrid filter and the analysis bank of the filter bank respectively. Another section of the filter bank echo canceller has the synthesis bank of the filter bank. The input signal follows the two paths for the adaptation process in the echo canceller. The echo path consists of the hybrid filter and an analysis bank. The other path consists of an analysis bank and the transversal adaptive filter. Adaptive processing is performed separately in every channel of the filter bank. After the adaptive process, the error signals of all channels pass through the synthesis bank. The output from the synthesis bank is in the ideal case the desired echo-free signal. In the Fig. 4.3 W_1, W_2, W_3, W_4 represent the weights of the transversal adaptive filters for the four channels of the filter bank respectively. It can be mentioned that hybrid filter, widely used in long-distance telephone lines is a two-wire/ four-wire device which ideally receives and transmits the signal to the transmitting-end with out any reflection/leakage of the transmitted signal to the receiving-end. But for many reasons, the hybrid device can not provide ideal performance due to the variations in instrument characteristics, loop length, impedence etc and produces an echo from the leakage of the transmitting signal.

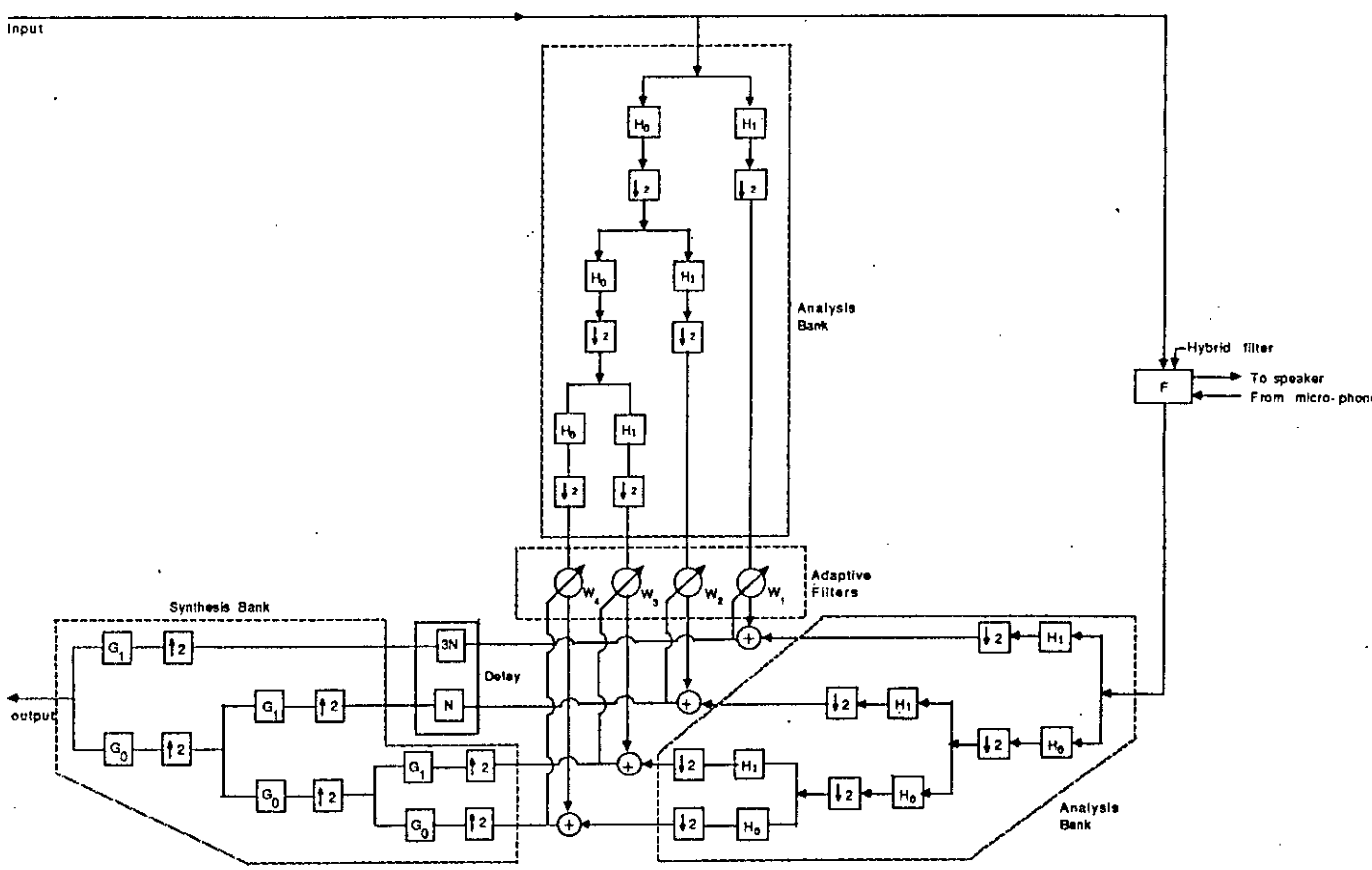

Figure 4.3: Block diagram of a filter bank echo canceller.

4.3 Finding the optimal weights.

In this section the optimal weights of the adaptive filters in the echo canceler are calculated. First we study path number 1, in search for the optimal weights $\overline{W_1}$.

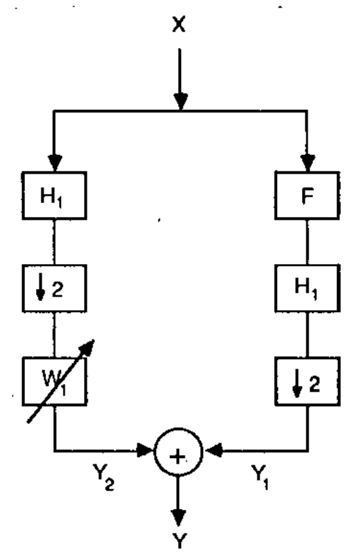

Figure 4.4: Calculation of the optimal weights.

Let the input signal be $X(z)$, the output signals from the echo path and the canceler path respectively are $Y_1(z)$, $Y_2(z)$ and let the optimal weights be $\overline{W}_1(z)$,(see Fig.4.4).
The outputs then can be calculated as follows:

$$Y_1(z) = \frac{1}{2}\left[F(z^{\frac{1}{2}})H_1(z^{\frac{1}{2}})X(z^{\frac{1}{2}}) + F(-z^{\frac{1}{2}})H_1(-z^{\frac{1}{2}})X(-z^{\frac{1}{2}})\right] \quad (4.1)$$

and

$$Y_2(z) = \frac{1}{2}\left[H_1(z^{\frac{1}{2}})X(z^{\frac{1}{2}}) + H_1(-z^{\frac{1}{2}})X(-z^{\frac{1}{2}})\right]\overline{W}_1(z) \quad (4.2)$$

For the optimal weights $Y_1(z)$ and $Y_2(z)$ are equal. Equating Eq.(4.1) and (4.2), we get

$$\overline{W}_1(z) = \frac{\left[F(z^{\frac{1}{2}})H_1(z^{\frac{1}{2}})X(z^{\frac{1}{2}}) + F(-z^{\frac{1}{2}})H_1(-z^{\frac{1}{2}})X(-z^{\frac{1}{2}})\right]}{\left[H_1(z^{\frac{1}{2}})X(z^{\frac{1}{2}}) + H_1(-z^{\frac{1}{2}})X(-z^{\frac{1}{2}})\right]} \quad (4.3)$$

When we want to find the optimal weights $\overline{W_3}$, we can draw Fig.4.5 for path number 3. Following the notation of Fig.4.5, we can express the two outputs as follows:

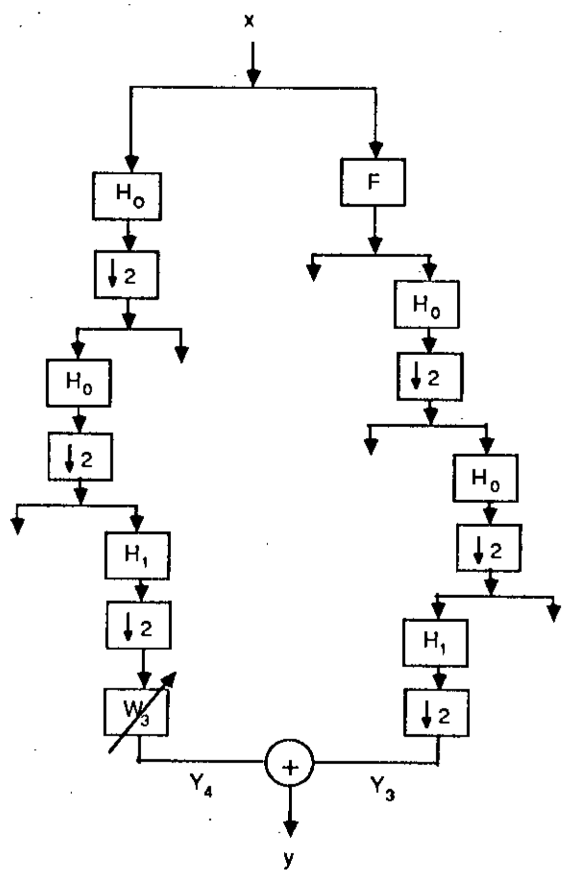

Figure 4.5: Calculation of the optimal weights $\overline{W}_3$

$$Y_3(z) = \frac{1}{8}\left[\left(H_1(z^{\frac{1}{2}}) + H_1(-z^{\frac{1}{2}})\right)\left(H_1(z^{\frac{1}{4}}) + H_1(-z^{\frac{1}{4}})\right)\right] \left[\left(F(z^{\frac{1}{8}})H_0(z^{\frac{1}{8}})X(z^{\frac{1}{8}}) + F(-z^{\frac{1}{8}})H_0(-z^{\frac{1}{8}})X(-z^{\frac{1}{8}})\right)\right] \tag{4.4}$$

$$Y_4(z) = \frac{1}{8}\left[\left(H_1(z^{\frac{1}{2}}) + H_1(-z^{\frac{1}{2}})\right)\delta\left(H_1(z^{\frac{1}{4}}) + H_1(-z^{\frac{1}{4}})\right)\right] \left[\left(H_0(z^{\frac{1}{8}})X(z^{\frac{1}{8}}) + H_0(-z^{\frac{1}{8}})X(-z^{\frac{1}{2}})\right)\right]\overline{W}_3(z) \tag{4.5}$$

We equate $Y_3(z)$ and $Y_4(z)$ and find the expression of $\overline{W}_3$,

$$\overline{W}_3(z) = \frac{\left[F(z^{\frac{1}{8}})H_1(z^{\frac{1}{8}})X(z^{\frac{1}{8}}) + F(-z^{\frac{1}{8}})H_1(-z^{\frac{1}{8}})X(-z^{\frac{1}{8}})\right]}{\left[H_1(z^{\frac{1}{8}})X(z^{\frac{1}{8}}) + H_1(-z^{\frac{1}{8}})X(-z^{\frac{1}{8}})\right]} \tag{1.6}$$

In the similar way, we can find the expression for $\overline{W}_2$ and $\overline{W}_4$.

$$\overline{W}_2(z) = \frac{\left[F(z^{\frac{1}{4}})H_1(z^{\frac{1}{4}})X(z^{\frac{1}{4}}) + F(-z^{\frac{1}{4}})H_1(-z^{\frac{1}{4}})X(-z^{\frac{1}{4}})\right]}{\left[H_1(z^{\frac{1}{4}})X(z^{\frac{1}{4}}) + H_1(-z^{\frac{1}{4}})X(-z^{\frac{1}{4}})\right]} \tag{4.7}$$

$$\overline{W}_4(z) = \frac{\left[F(z^{\frac{1}{8}})H_0(z^{\frac{1}{8}})X(z^{\frac{1}{8}}) + F(-z^{\frac{1}{8}})H_0(-z^{\frac{1}{8}})X(-z^{\frac{1}{8}})\right]}{\left[H_0(z^{\frac{1}{8}})X(z^{\frac{1}{8}}) + H_0(-z^{\frac{1}{8}})X(-z^{\frac{1}{8}})\right]} \tag{4.8}$$

If we look for the expression of weights, we see that these weights depend on the input signal, the hybrid filter, the analysis filter, and the decimation factor. It is also noticed that for $X(z)=1$(when the input signal $x(n)$ is an impulse), the optimal weights depend only on the hybrid filter and the decimation factor and the depth of the filter bank and it is as follows:

$$\overline{W}_L(z) = F(z^{\frac{1}{2^L}})$$

where L is the depth of the filter bank.

4.4 LMS algorithm

The LMS (Least mean square) algorithm is an implementation of the method of steepest descent [2]. According to the LMS method, the ' next weight ' vector $\overline{W}(j+1)$ of the adaptive filter is equal to the ' present weight ' vector $\overline{W}(j)$ plus a change to the negative gradient:

$$\overline{W}(j+1) = \overline{W}(j) - \mu\hat{\nabla}(j)$$

where

$\overline{W}(j)$ = current weight vector.
$\overline{W}(j+1)$ = weight vector at next time instant.
μ = a factor that controls stability and rate of the convergence (stepsize).
$\hat{\nabla}$ = estimate of gradient of the mean square error with respect to $\overline{W}$.
j = iteration number of the adaptive process.

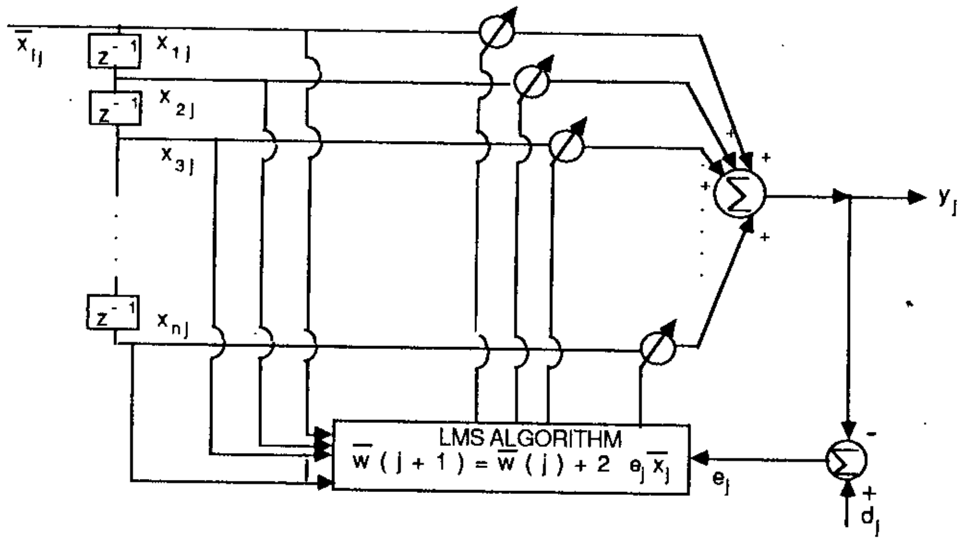

Figure 4.6: Digital implementation of LMS weight adjustment algorithm.

The method for obtaining the estimated gradient of the mean-square-error function is to take the gradient of a single time sample of the squared error,

$$\begin{aligned} \hat{\nabla}(j) &= \nabla[e^2(j)] \\ &= 2e(j)\nabla[e(j)] \end{aligned}$$

Error vector $e(j)$ is the difference between the desired response $d(j)$ and the output $Y(j)$.

$$e(j) = d(j) - Y(j) = d(j) - \overline{W}^T(j)\overline{X}(j)$$

$$\overline{X}(j) = \begin{bmatrix} x_{1j} \\ x_{2j} \\ x_{3j} \\ \cdot \\ \cdot \\ \cdot \\ x_{nj} \end{bmatrix},$$

The following figure represents the Eq.4.9 in terms of block diagram.

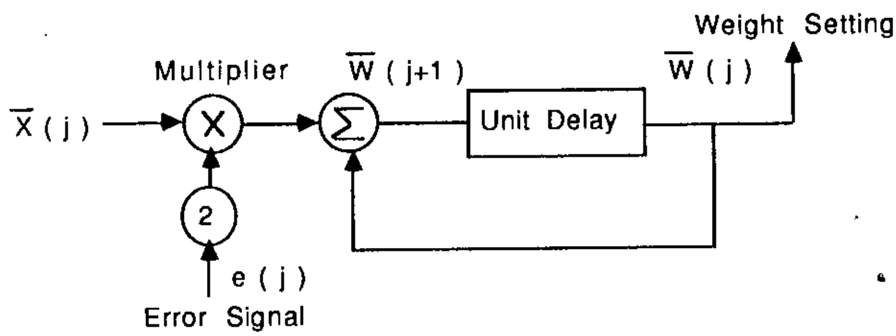

Figure 4.7: Block diagram representation of the LMS algorithm

$$\overline{W}(j) = \begin{bmatrix} w_1 j \\ w_2 j \\ w_3 j \\ \cdot \\ \cdot \\ \cdot \\ w_n j \end{bmatrix}$$

Thus the expression for the gradient estimate simplifies to

$$\hat{\nabla}(j) = -2e(j)\overline{X}(j)$$

Using this estimate in place of true gradient yields the LMS algorithm:

$$\overline{W}(j+1) = \overline{W}(j) + 2\mu e(j)\overline{X}(j) \tag{4.9}$$

This is directly usable as a weight adaptation formula.

The LMS algorithm is simple and generally easy to implement. Although it is based on gradients of mean-square-error functions, it does not require squaring, averaging, or differentiation.

The gradient estimate used in the LMS algorithm is unbiased (the mean value is not zero) and the expected value of the weight vector converges to the Wiener vector or the optimal vector $\overline{W}$, which can be calculated by setting the gradient of the mean- square- error function to zero. If R and P are auto-correlation matrix of the input signal and cross correlation matrix of the input signal and the desired signal respectively, the optimal weights can be written $\overline{W} = R^{-1}P$. Starting with an arbitrary initial weight vector, the algorithm will converge and will remain stable, given that the parameter μ is greater than zero and less than the reciprocal of the largest eigenvalue λ_{max} of the matrix R. We can see that λ_{max} can not be greater than the trace of R which is the sum of the diagonal elements of R. Furthermore, a transversal adaptive filter gives $tr[R]$ as just L (length of the filter) times the input signal power. Thus convergence of the weight vector mean is assured by:

In general	$0 < \mu < \frac{1}{tr[R]}$
Transversal filter	$0 < \mu < \frac{1}{L(signalpower)}$

The more weights used in the transversal filter, the closer its impulse response will be to that of the optimal Wiener filter (if the optimal filter is not finite). Increasing the number of weights, however, slows the adaptive process and increases the cost of implementation and introduces noise. So the performance requirments should be carefully considered before a filter is designed for a particular application.

Chapter 5

Simulations

In this chapter we describe the echo cancelling system and the results of simulations when white noise and speech signal are used respectively as the input to the system.

The system according to Fig. 4.3 is implemented in MATLAB, an interactive system for matrix calculations and signal processing. The analysis filters (H_0 and H_1) have been chosen as half-band Hamming window FIR filters with 128 (even number) coefficients and a -3 dB cut-off frequency of 0.25 times the sampling frequency. We have selected a hybrid filter which is a lowpass, 32 coefficient FIR filter and has a cut-off frequency of 0.4 times the sampling frequency. Adaptive filters with 32 weights have been chosen, the LMS algorithm is used for the adaptation. The maximum value of the step size μ_{max} has been calculated from the level of the variance of the input signal and the number of weights of the adaptive filter. A suitable fractional value of μ_{max} has been taken to guarantee the stability of the adaptive filter.

The following figures show the impulse response and the frequency response of the filters which are used in the system:

Fig. 5.1 shows the impulse response of the low-pass analysis filter. It has 128 number of coefficients and is symmtrical.

Fig. 5.2 represents the frequency response of the low-pass analysis filter in terms of normalized frequncy which has a cut-off frequency of (1/4) times the sampling frequency. It has also highly attenuated (60 dB) ripples in the stop-band.

Fig. 5.3 shows the impulse response of the high-pass filter. It has 128 number of coefficients and is symmetrical. Ω

Fig. 5.4 is the frequency response of the high pass analysis filter in terms of normalized frequency. It has a cut-off frequency of (1/4) times the sampling frequency and highly attenuated (60 dB) ripples in the stop band.

Fig. 5.5 shows the impulse response of the hybrid filter which has 32 number of coefficients and is symmetrical.

Fig. 5.6 represents the frequency response of the hybrid filter in terms of normalized frequency It has a cut-off frequency of (2/5) times the sampling frequency and highly attenuated (60 dB) ripples in its stop band.

5.1 Echo cancelling using noise.

In this section the results of simulations using noise as input will be presented. The noise is white gaussian with a variance of 1.

Fig. 5.7 and 5.8 show the input (white noise) and the output after simulation for the noise of 7000 samples respectively. It was found that the convergence/reduction of noise is quite fast. After 2000 samples (i.e. 0.25 second when sampling rate 8 kHz is used), the output comes close to zero.

Fig. 5.9 - 5.12 show the error signals after the adaptation in every channel of the filter bank. The different length of the error signals are due to the different amount of decimation preceeding the adaptive filters.

Fig.5.13 - 5.16 show the weights of the adaptive filters for every channel of the filter bank. The adaptive filters adapt all the weights (32 weights) of the hybrid filter in every channel. During adaptation we used 16 samples delay preceeding the hybrid filter. This means that the maximum value of the hybrid coefficients will be at $16 + (\frac{1}{2})$length of hybrid = 32. Due to the decimation the weights in Fig.5.13 have their maximum value at sample number 16. For the paths with larger decimation factor the influence of the delay will be less, i.e. the maximum value will occur at sample number 8 and 4 respectively.

Fig.5.17 shows the ratio between the short term energy (STE) of the echo and the STE of the signal after echo cancellation. This is a measurement of the performance of the echo canceller, and a high value means a large echo suppression. The figure shows that the ratio is growing to a certain level (approximately 70 dB) during the first 3000 samples, then the increase is stopped and the level remains constant. The slope of the curve can be changed by varying the stepsize (μ). As the stepsize increases the slope of the curve decreases and vice versa.

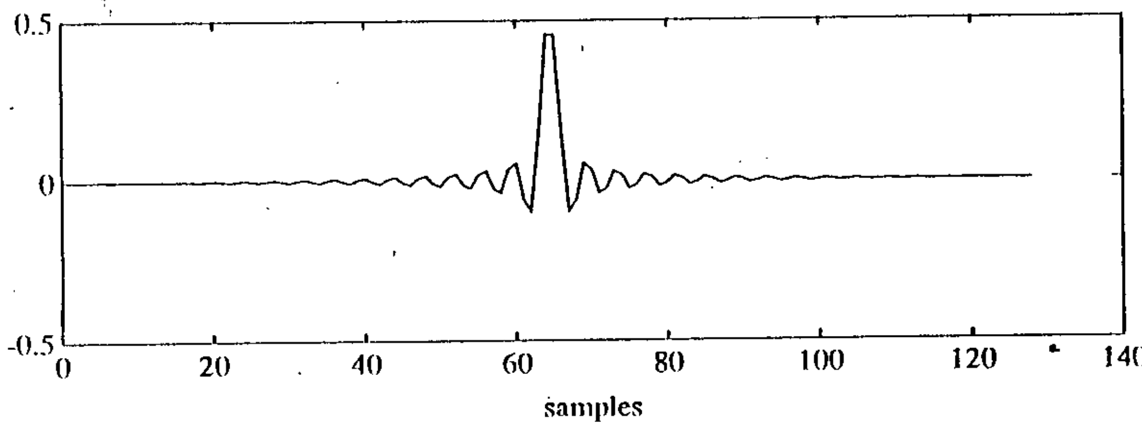

Figure 5.1: Impulse response of the analysis filter(h_0).

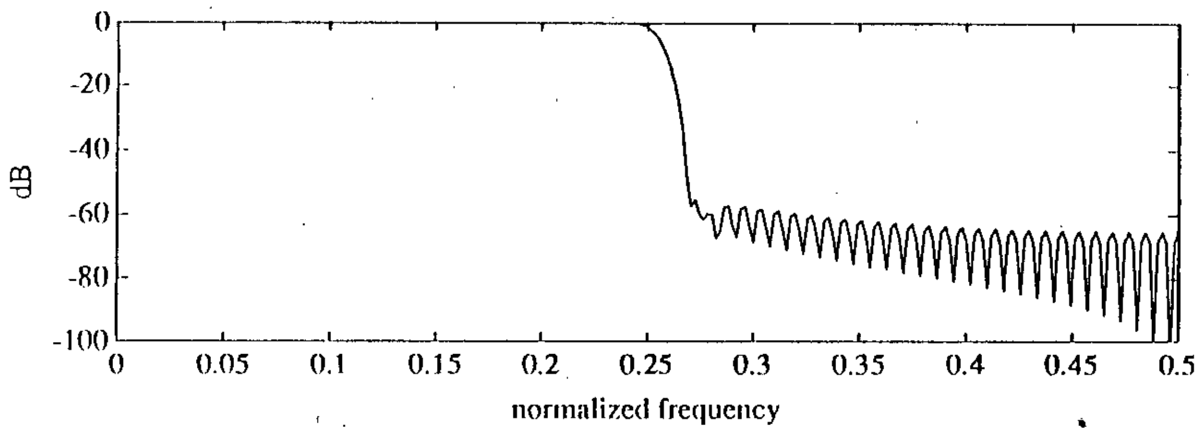

Figure 5.2: Frequency response of the analysis filter(H_0).

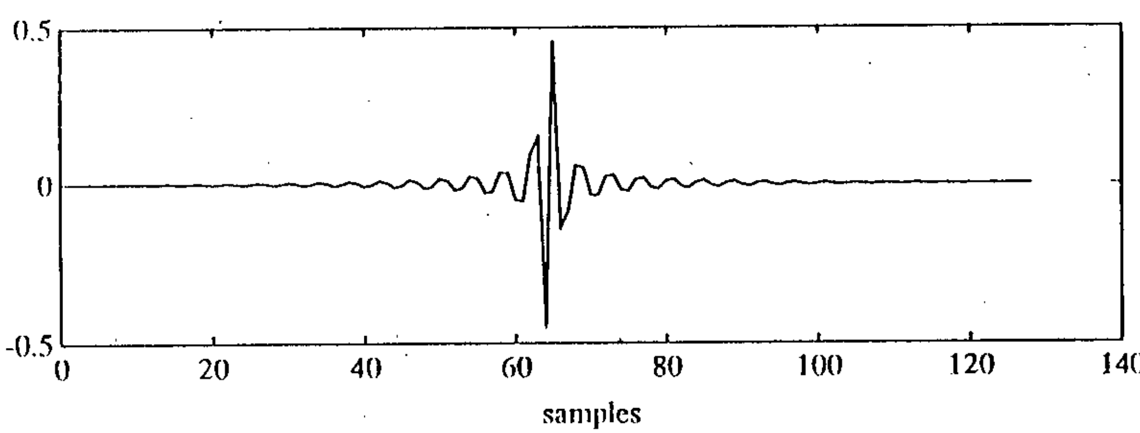

Figure 5.3: Impulse response of the analysis filter(h_1).

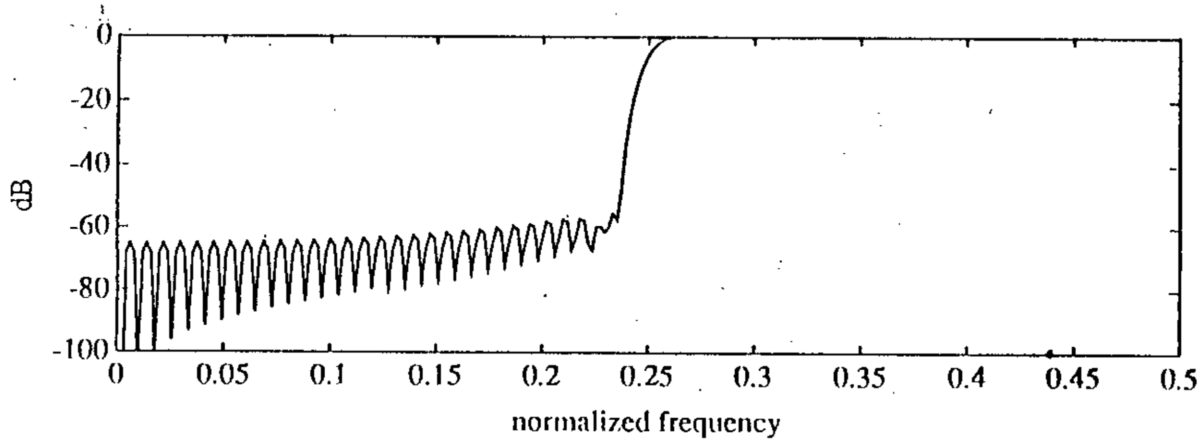

Figure 5.4: Frequency response of the analysis filter(H_1).

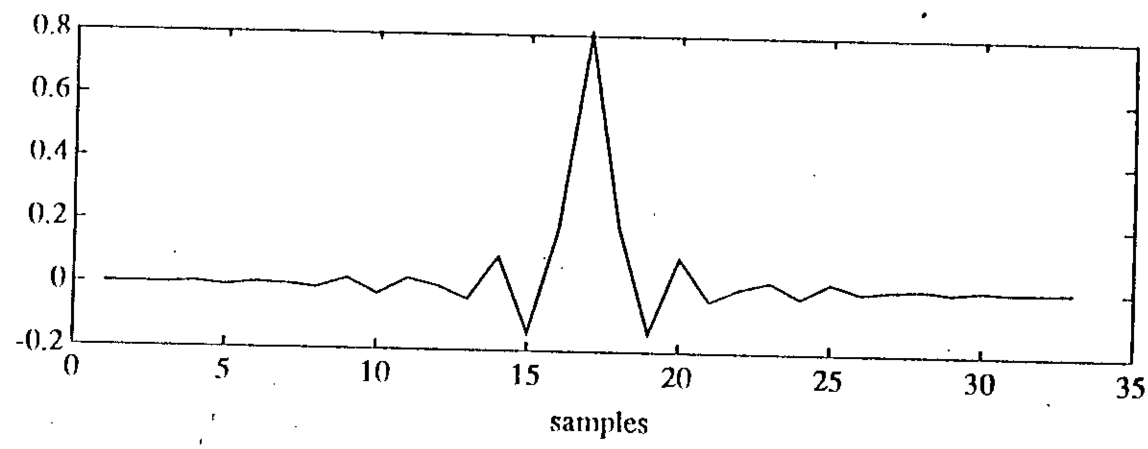

Figure 5.5: Impulse response of the hybrid filter.

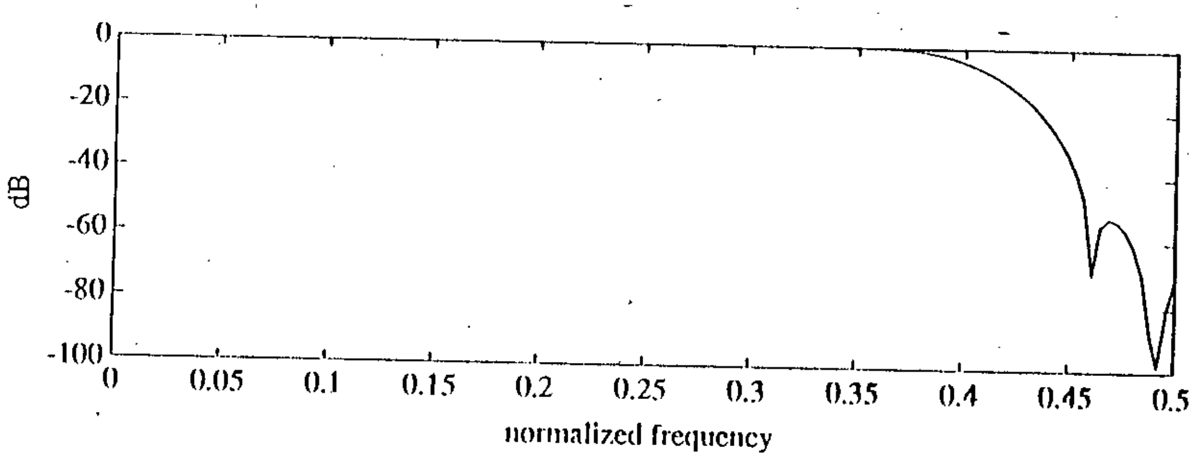

Figure 5.6: Frequency response of the hybrid filter.

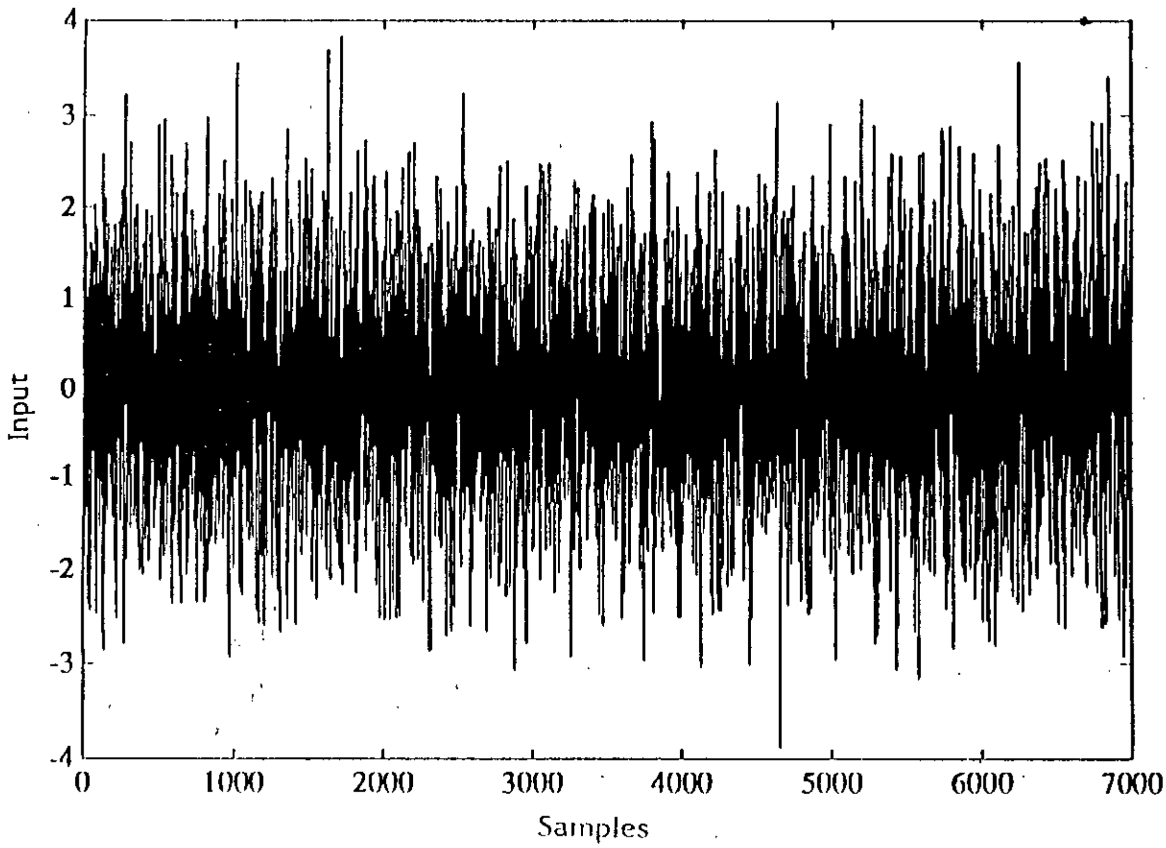

Figure 5.7: White noise.

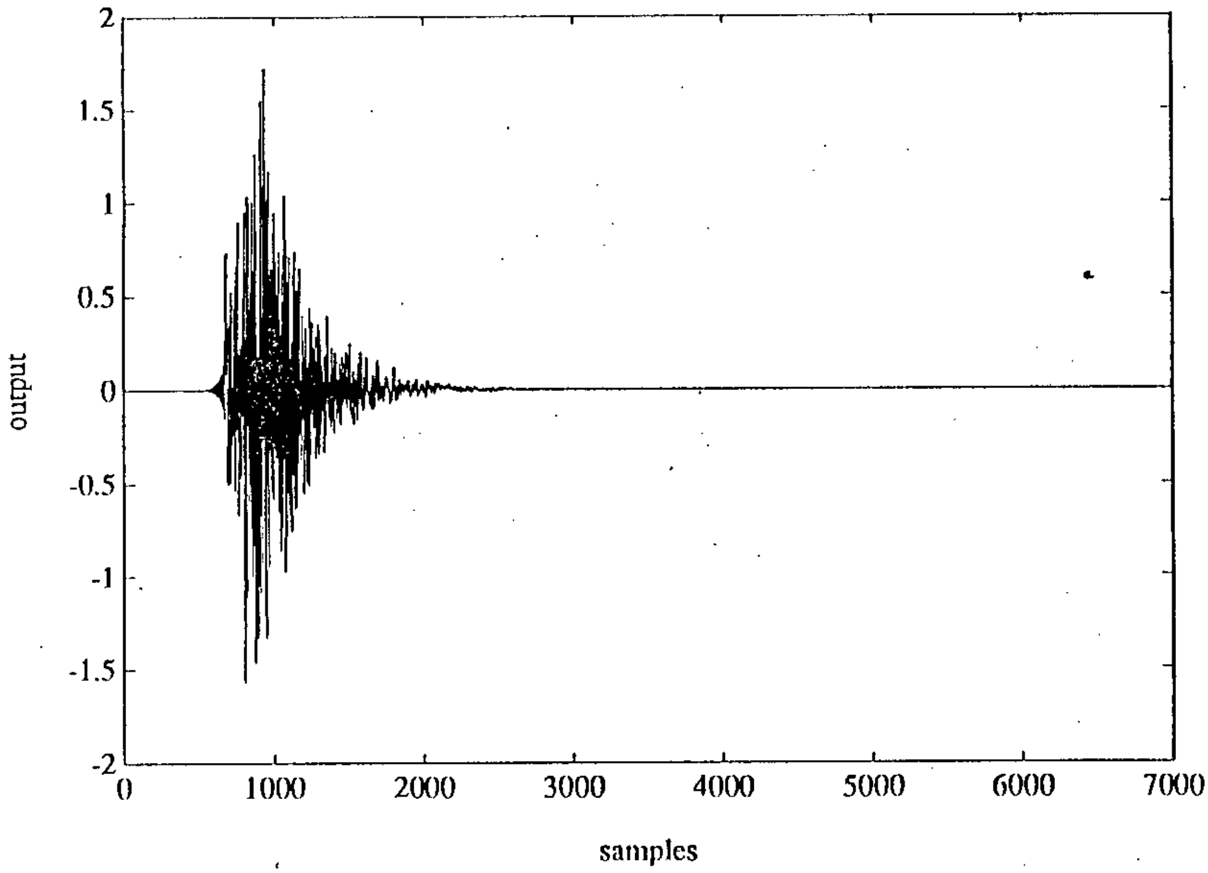

Figure 5.8: Output versus iteration number when the input is noise.

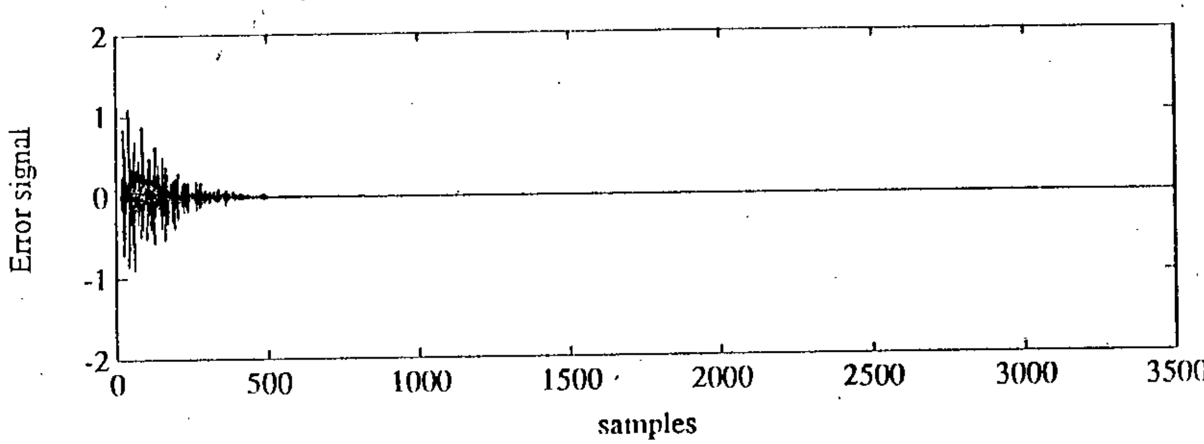

Figure 5.9: Error signal for channel 1 (high pass).

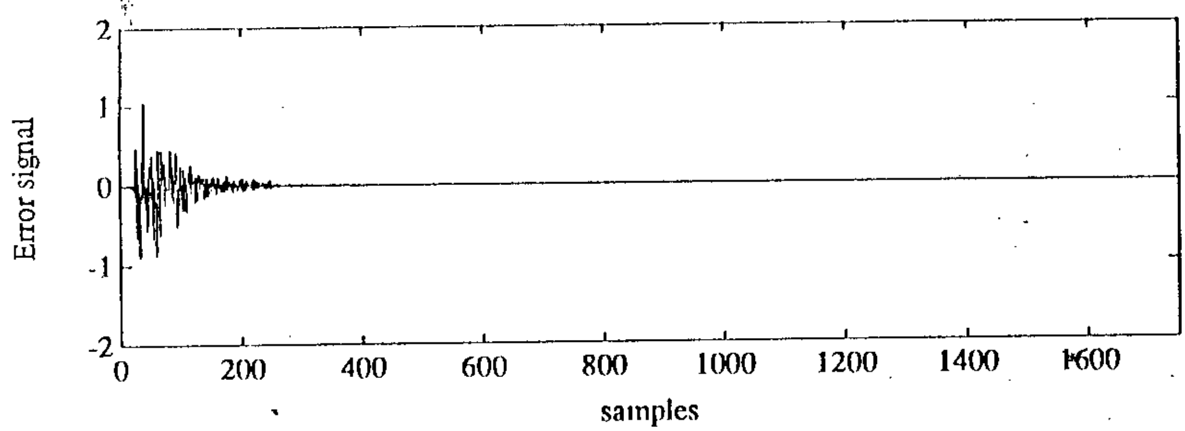

Figure 5.10: Error signal for channel 2 (band pass).

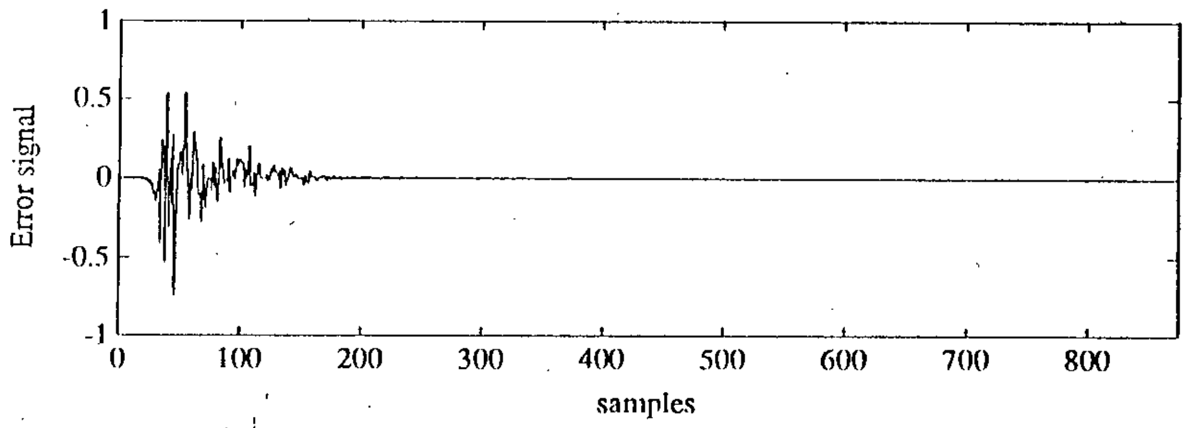

Figure 5.11: Error signal for channel 3 (band pass).

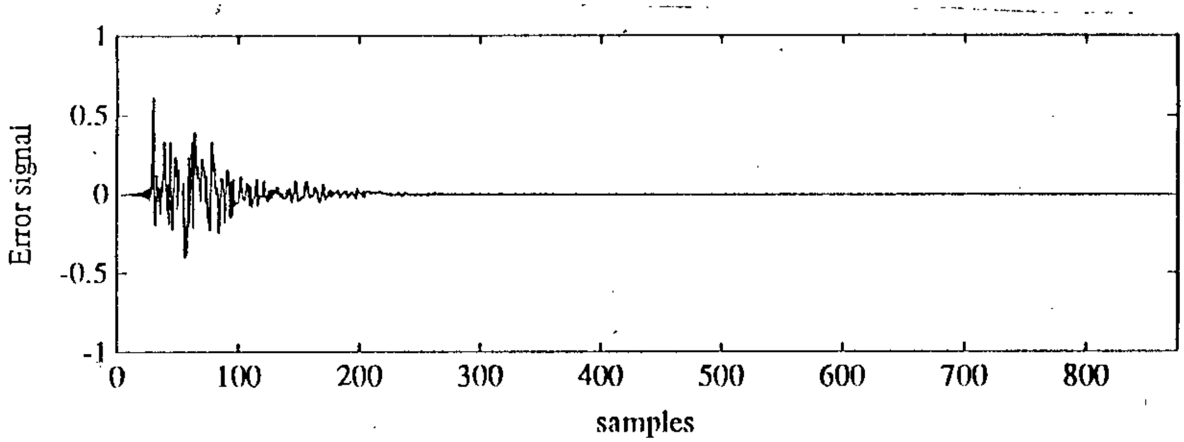

Figure 5.12: Error signal for channel 4 (low pass).

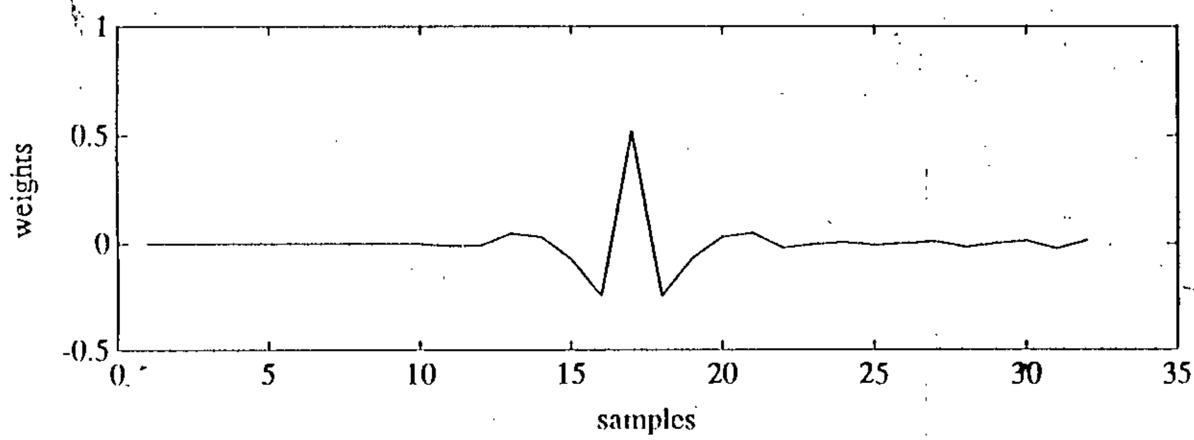

Figure 5.13: Weights for high pass channel(W_1).

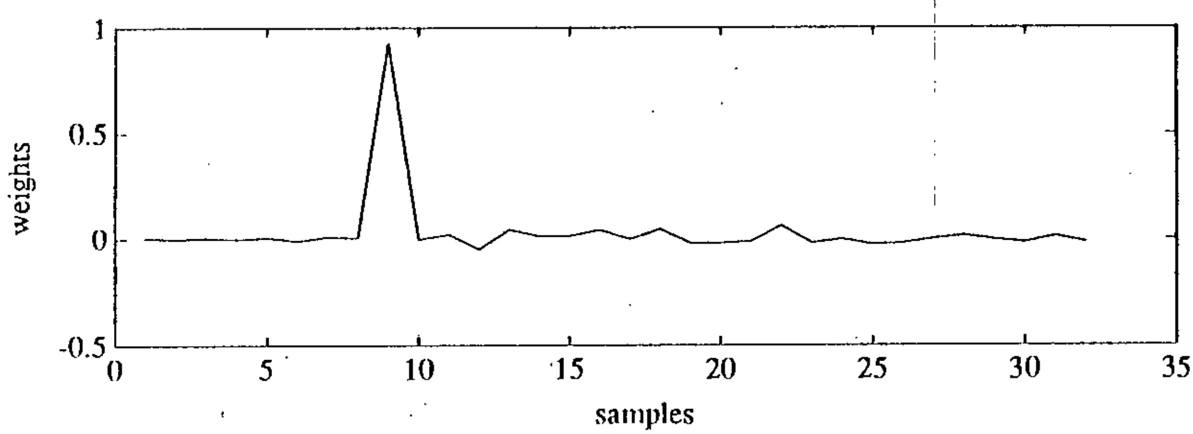

Figure 5.14: Weights for band pass channel(W_2).

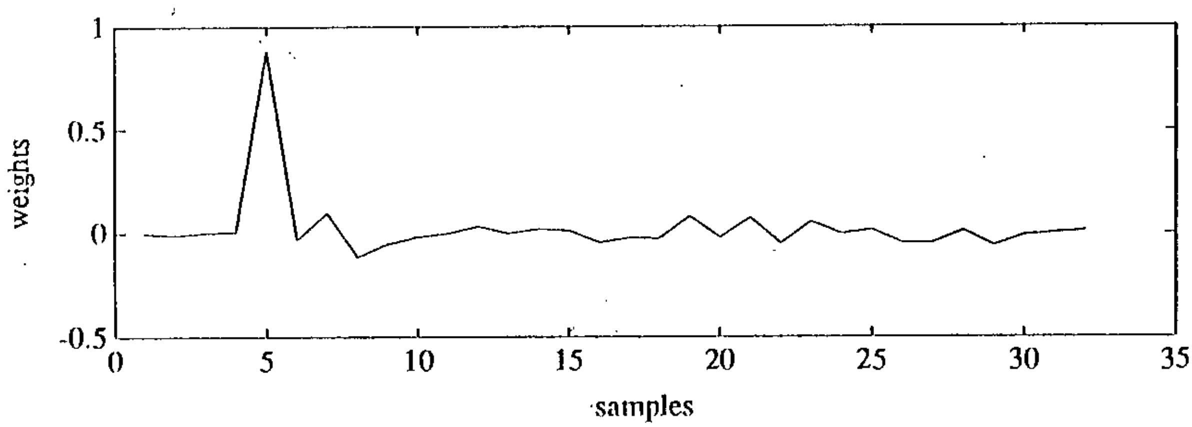

Figure 5.15: Weights for band pass channel(W_3).

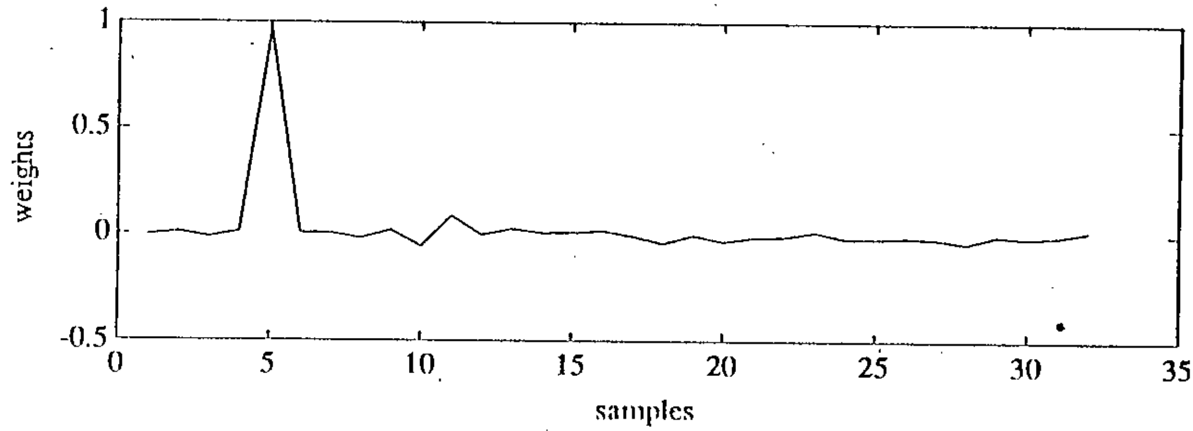

Figure 5.16: Weights for low pass channel(W_4).

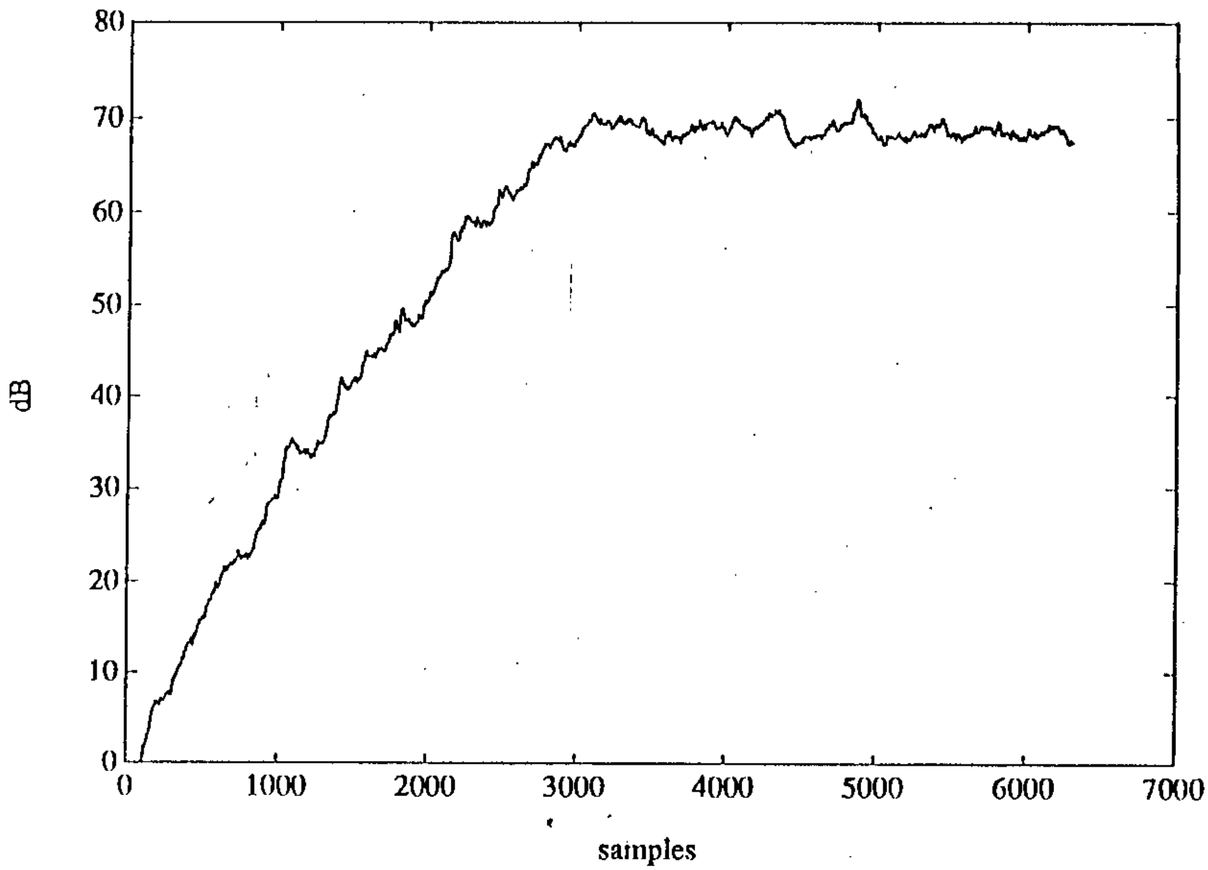

Figure 5.17: Echo suppression ratio for noise input case.

5.2 Echo cancelling using speech signal

In this section the results of simulations using speech signal as input will be presented.

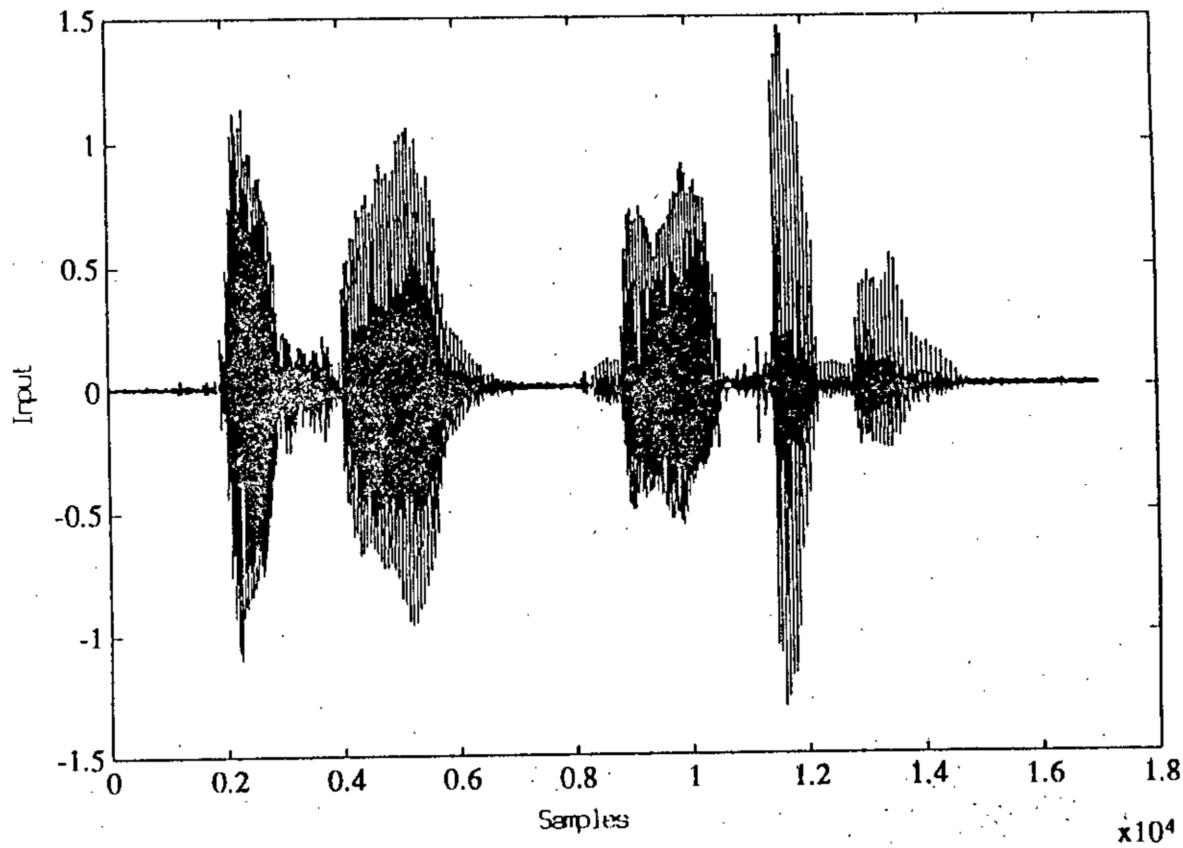

Figure 5.18: Speech signal.

The speech signal of 17000 samples (2.125*sec*) is used here. Fig. 5.18 refers the speech signal whereas Fig. 5.19 shows its output signal during cancelation. In the Fig. 5.19 we find that the suppression of the speech signal occurs at all frequency bands. In the speech signal frequency bands have different amount of energy which caus different step size (μ) for best adaptation. For this reason the μ value is normalized by the power in each band.

The suppresion of echo can be recognized significantly if we compare the magnitude of the

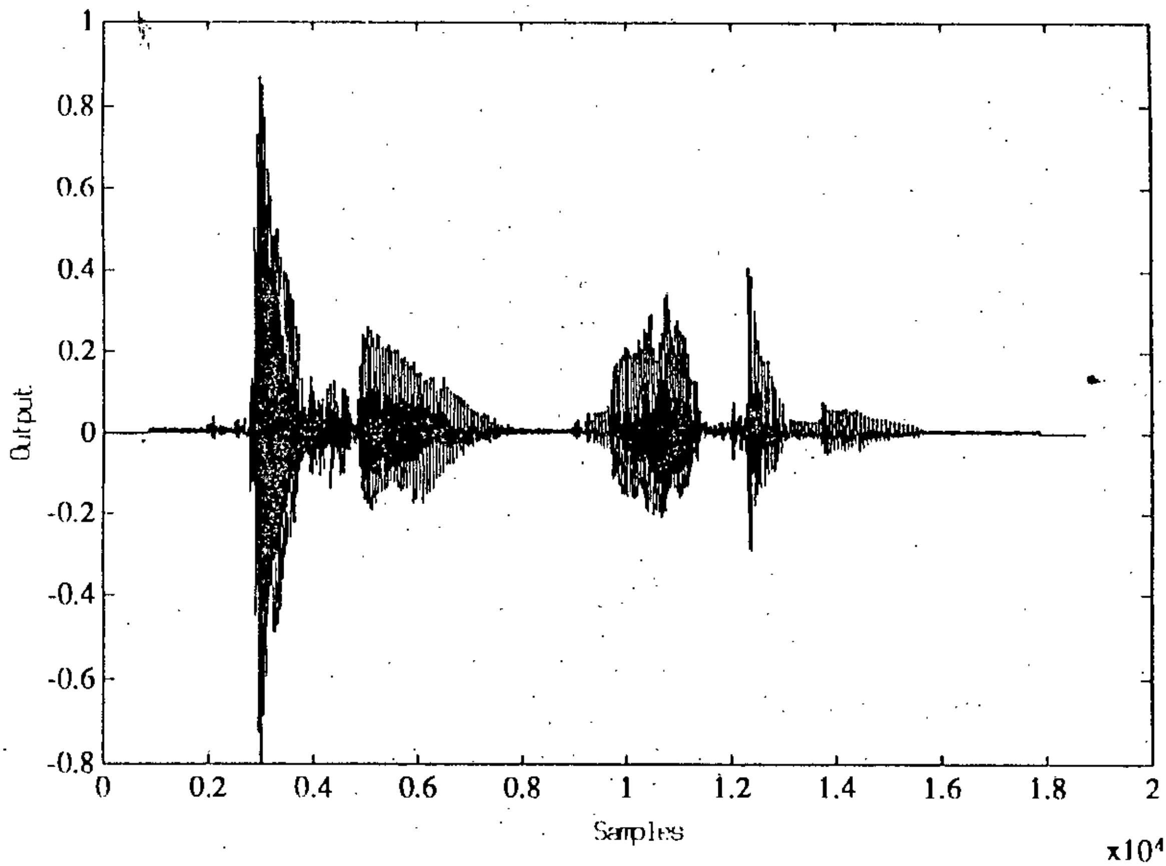

Figure 5.19: Output versus iteration number using speech signal as input.

input and the output of the speech signal in Fig. 5.18 and 5.19 respectively. We find that the adaptation goes only during the presence of information content but not during the pause in the speech signal.

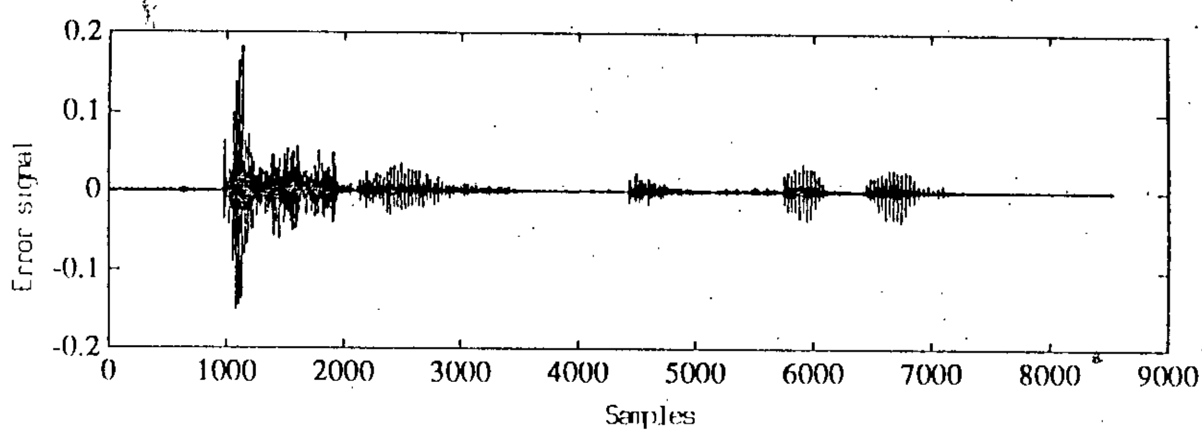

Figure 5.20: Error signal for channel 1 (high pass).

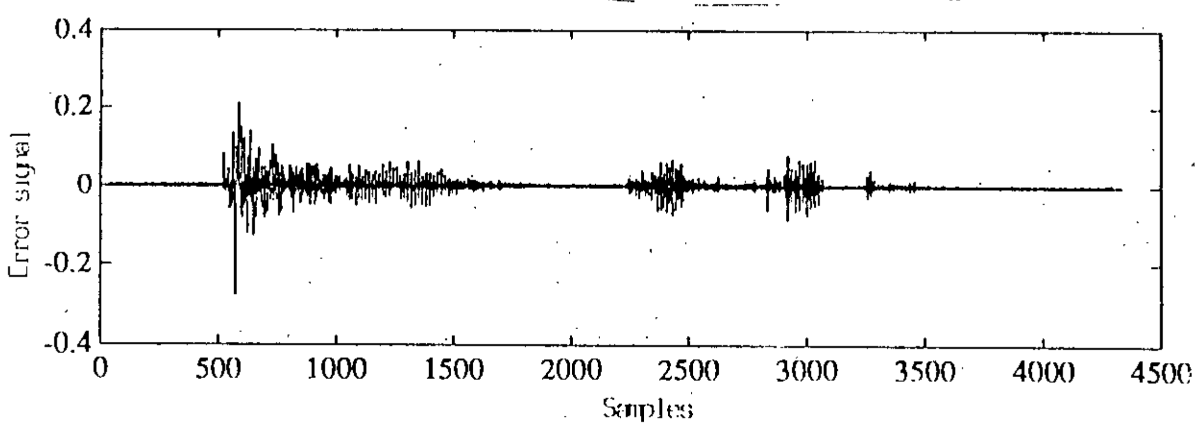

Figure 5.21: Error signal for channel 2 (band pass).

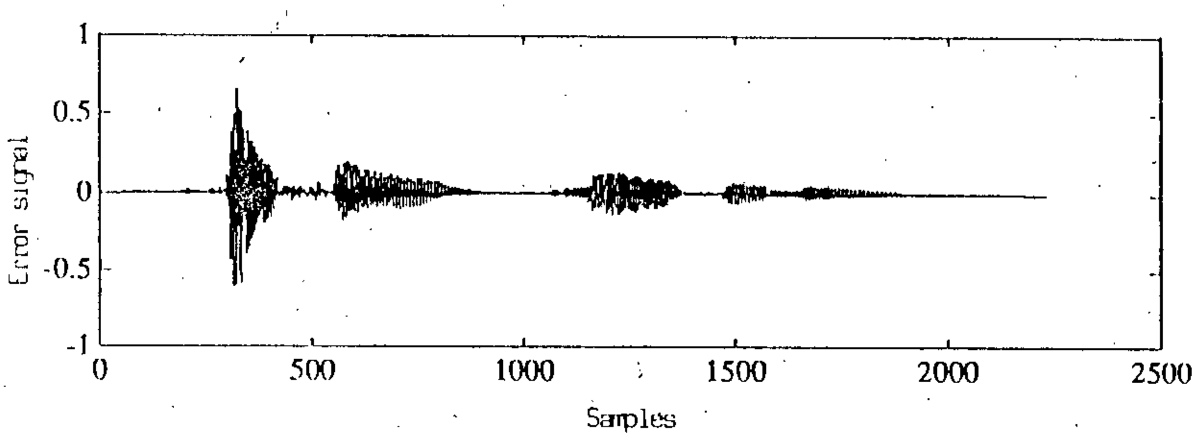

Figure 5.22: Error signal for channel 3 (band pass).

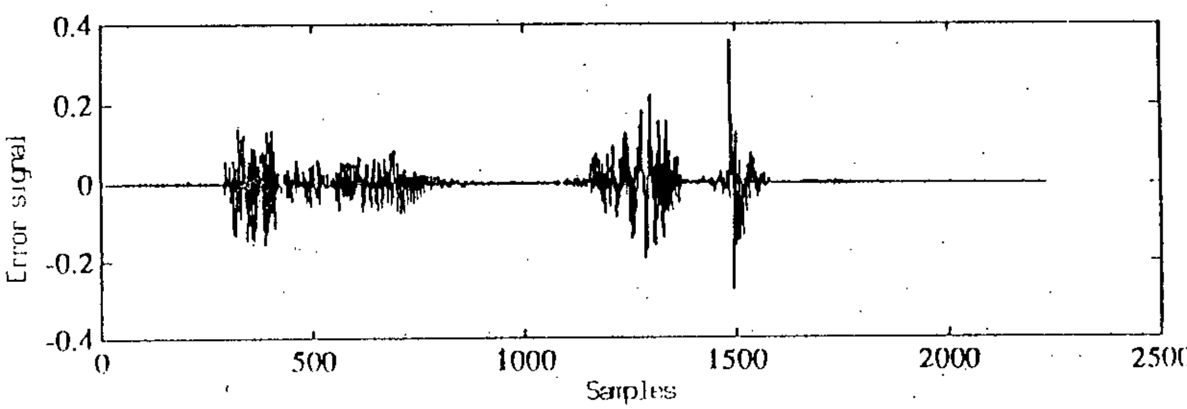

Figure 5.23: Error signal for channel 4 (low pass).

Fig.5.20 - 5.23 refer the error signals after the adaptation in every channel of the filter bank. It is also noted that the different length of the error signals are due to the different amount of decimation preceeding the adaptive filters.

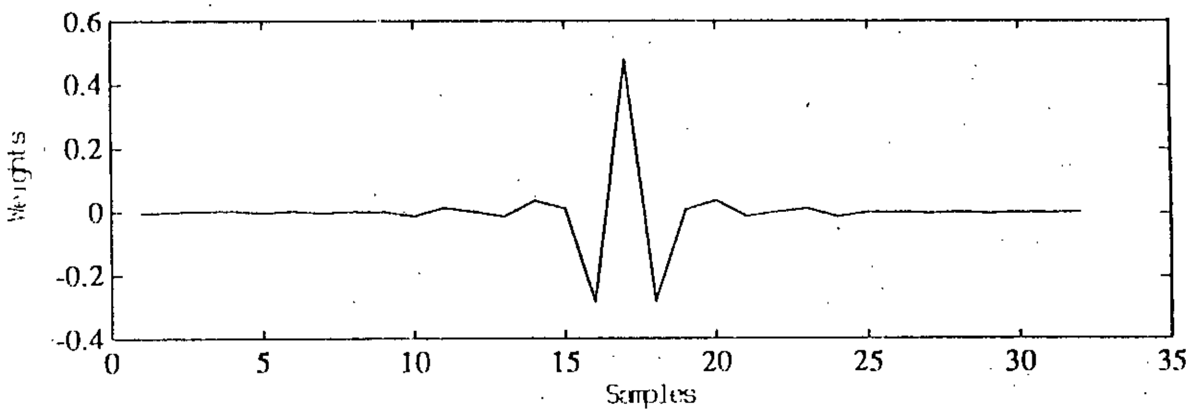

Figure 5.24: Weights for high pass channel(W_1).

Fig. 5.24 - 5.27 show the weights of the hybrid filter in the four channels. During adaptation we use 16 samples delay preceeding the hybrid filter. This means that the maximum value of the hybrid coefficients will be at $16 + (\frac{1}{2})$length of hybrid= 32. Due to the decimation the weights in Figure 5.24 have their maximum value at sample number 16. For the paths with longer decimation factor the influence of the delay will be less, i.e. the maximum value will occur at sample number 8 and 4 respectively.

Fig.5.28 shows the ratio between the short term energy (STE) of the echo and the STE of the signal after echo cancellation. The variation of the curve in Fig. 5.28 is quite remarkable. This happens due to the variation of energy of the speech signal (echo). Due to the high and low energy of the speech signal or the echo, the echo suppression ratio becomes high and low respectively. Thats why during the pause in the speech signal there are no energy and it makes the echo suppression ratio zero and sometimes slightly below zero. Negative suppression ratio occurs due to the variance in the estimation of the STE or due to the the aliasing effects in the

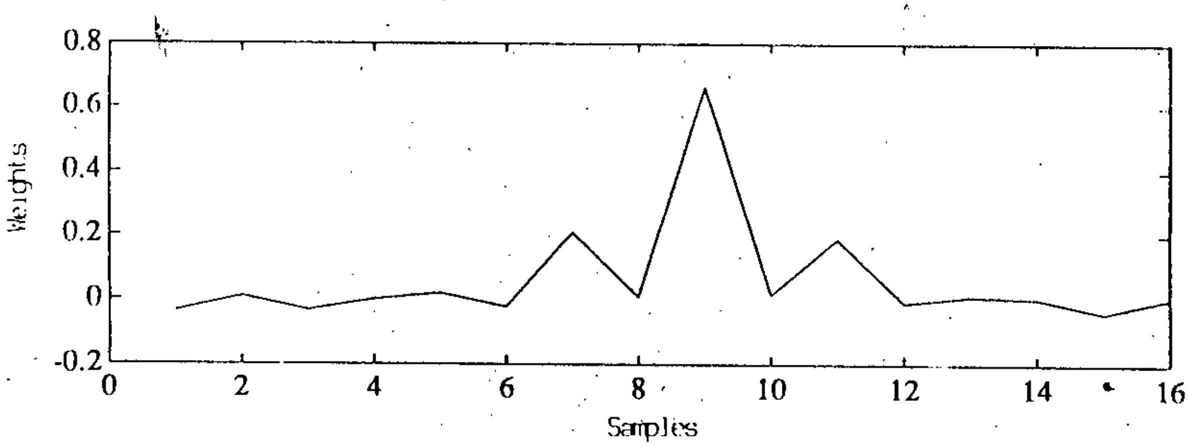

Figure 5.25: Weights for band pass channel(W_2).

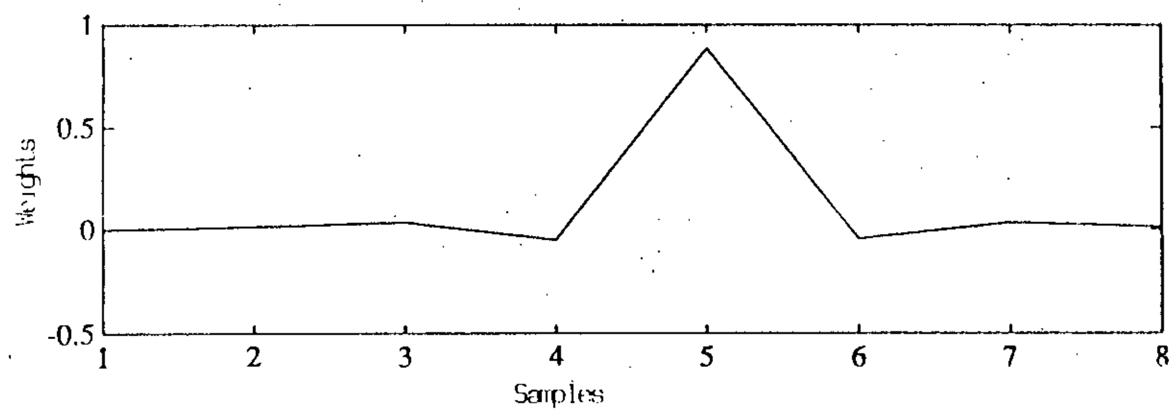

Figure 5.26: Weights for band pass channel(W_3).

analysis bank.

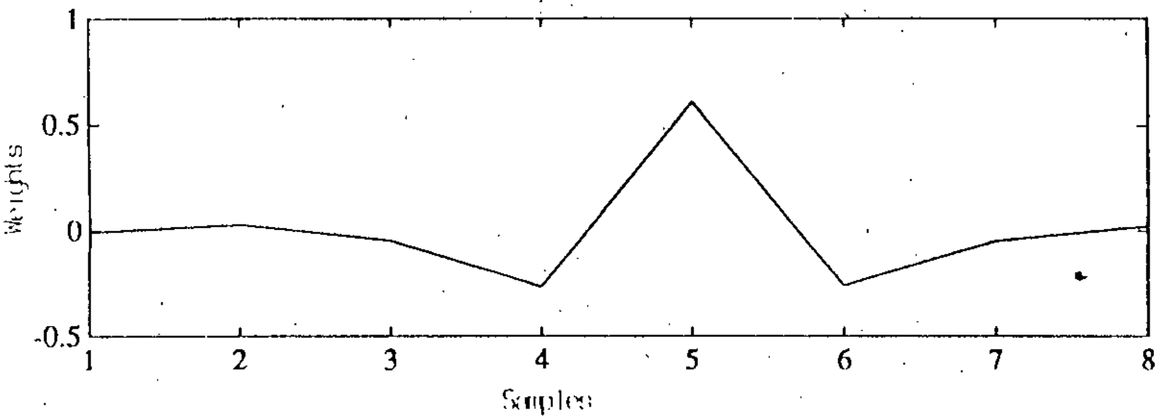

Figure 5.27: Weights for low pass channel(W_4).

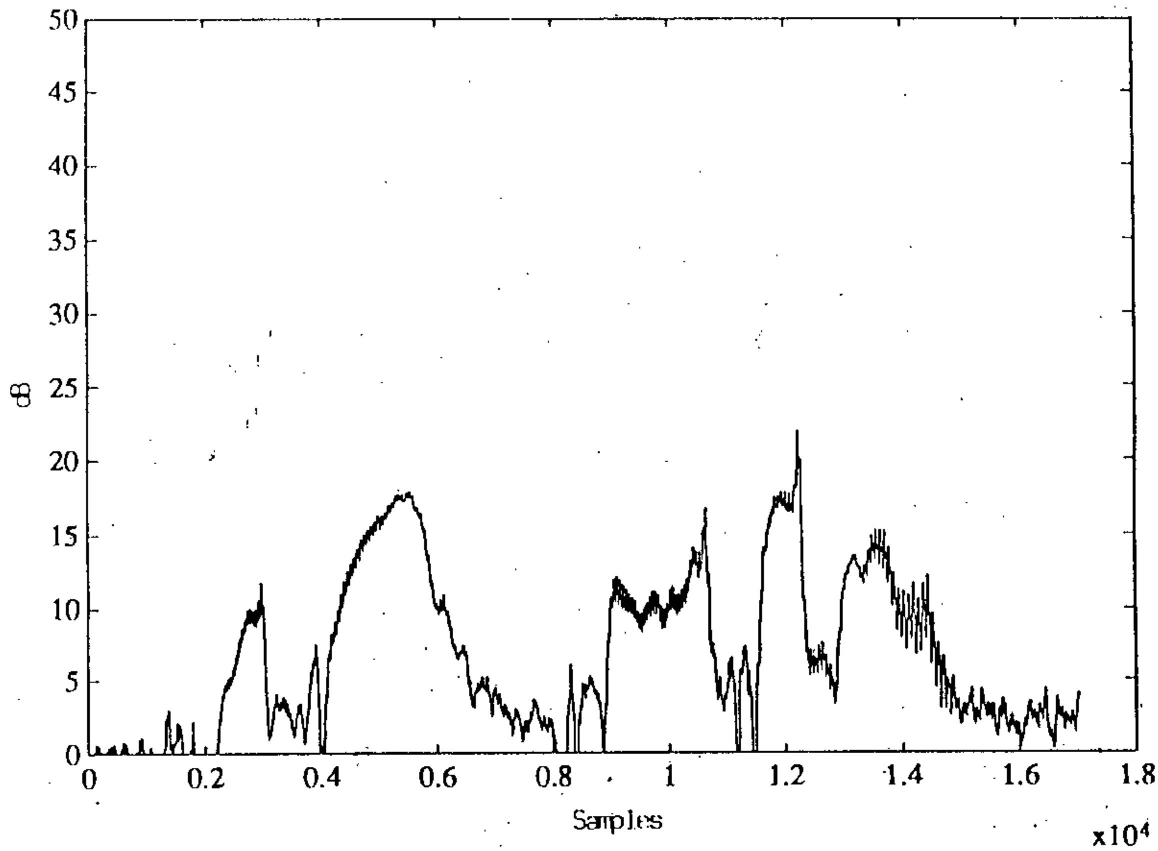

Figure 5.28: Echo supression ratio for speech signal input case.

Chapter 6

Discussions

We have derived in this thesis the design of a FIR digital filter bank and its implementation by computer simulation using MATLAB (matrix laboratory) program. The filter bank is of importance for among others the following reasons:

- to split the frequency band of the signal into several sub-bands.
- to cancel the echo of a speech signal for telephone lines.

The FIR filter bank consists of three sections: analyser, processor and synthesizer. The QMFs (quadrature mirror filters) in the analyser should have a finite even number of coefficients and be symmetrical half-band filter for perfect reconstruction of signals. Symmetrical but odd number of coefficients filters can not reconstruct a signal perfectly. The delay in the filter bank depends upon the order of the filter and the depth of the filter bank.

The filter bank can split the frequency band of the signal into different sub-bands. The number of sub-bands and their size depend upon the depth of the filter bank. As the depth increases, the size of the sub-bands decreases. The quality of the division into frequency bands depends on the lengths of the QMF filters. When narrow bands with little overlapping is wanted long filters has to be used.

When adaptive filters are used for each channel of the echo canceller filter bank, they adapt with satisfactory and cancel the echo from the hybrid filter. The echo supression is quite high especially when the input is noise. The Least-Mean-Square (LMS) algorithm is used for adaptation which provides a simple implementation for echo cancelling.

We use a four channel (depth = 3) digital filter bank but a filter bank with any number of channels can be used for the applications studied.

The filter bank echo canceller has an advantage over the ordinary echo canceller The ordinary echo canceller suppresses echo by using only one adaptive filter to adapt information at all frequencies while the filter bank echo canceller has adaptive filters in several frequency bands. This means that it can adapt information of different frequency separately using different number of weights determined by considering the information content or the energy of the signals for the different frequency bands. If the information content or the energy of the signals is low then less number of weights for adaptation are required and vice versa. Thats why filter bank echo canceller could be much faster than the ordinary echo canceller.

Finally it can be mentioned that this FIR filter bank could be implemented in other important applications. As an example, it can be used in some acoustic applications such as cancelling interference from different kinds of sources. It can also be applied for echo cancelling of high frequency signals.

REFERENCES

[1] O. Herrmann, " On the design of non-recursive digital filters with linear phase," Elec. Lett., Vol. 6, No.11, pp. 328-329, 1970.

[2] O. Herrmann and W. Schussler, " Design of non-recursive digital filter with minimum phase," Elec. Lett., Vol. 6, No. 11, pp. 329-330, 1970.

[3] E. Hofstetter, A. V. Oppenheim and J. Siegel, "On the optimum non-recursive digital filters," Proc. 9th Allerton Conf. Circuit System Theory, Oct., 1971.

[4] T. W. Parks and J.H. McClellan, " A program for the design of linear phase finite impulse response filters," IEEE Trans. Audio Electroacoustics, Vol. AU-20, No. 3, pp. 195-199, Aug., 1972.

[5] L. R. Rabiner, " Linear program design of finite impulse response digital filters,"IEEE Trans. On Audio and Electro-acoustics, Vol. AU-20, No. 4, pp. 280-288, Oct., 1972.

[6] A. V. Oppenheim, R. W. Schafer, " Digital Signal Processing," Prentice Hall, 1989, pp. 102-105, 244-250.

[7] M. J. T. Smith and T. P. Barnwell, " Exact Reconstruction Techniques for Tree-Structured Subband Coders, "IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-34, pp. 434-441, June, 1986.

[8] B. Widrow and S. D. Stearns, " Adaptive Signal Processing, " Prentice-Hall, 1985, pp. 99-114.

[9] V. Cappellini, A. G. Constantinides and P. Emiliani," Digital filters and their applications", Academic Press LTD. (london), 1981.

[10] E. A. Lee, D. G. Messerschmitt, " Digital Communication," Kluwer Academic Publishers, 1988, pp. 643-646.

[11] B. Widrow, " Stationary and Nonstationary Learning Characteristics of the LMS Adaptive Filter", Proc. IEEE, Vol. 64, No. 8, pp. 1151-1162, Aug., 1976.

[12] T. W. Parks and X. Chen, " Design of digital filters in the Complex Domain," IEEE Trans. on Acoustics, Speech, and Signal Processing, Vol. ASSP-35, pp. 144-153, 1987

[13] T. W. Parks and C. S. Burrus, " Digital Filter Design," John Wiley, New York, 1987.

[14] H. W. Schussler and P. Steffen, " Some Advanced Topics in Filter Design," in Advanced Topics in Signal Processing, J. S. Lim and A. V. Oppenheim, Eds.,Prentice Hall, Englewood Cliffs, NJ, 1988.

APPENDIX

Appendix A	MATLAB programs relating to perfect reconstruction of the signal and frequency sub-bands selection
Appendix B	MATLAB programs involving echo cancelling using white noise as the input signal
Appendix C	MATLAB programs involving echo cancelling using echo speech as the input signal
Appendix D	Values of the coefficients of the two analysis filters and the hybrid filter

APPENDIX-A

_WIDROW$DUA0:[TTTUSER.FS.BANK]BANK.M;4

```
% Bandpass filter bank.
%The impulse response B and the input signal in (column vectors!)
%must be defined before this program is run.
%

c= input ('Give the four coefficients!')
%analysis
%
lh=length(B);
n=[0:lh-1]';
B1= (-1).^n.*B(lh-n);
[out1,out2] = split1(in,B,B1);
[out3,out4] = split1(out1,B,B1);
[out5,out6] = split1(out3,B,B1);
%
%Processing step
%
out5 = c(1)*out5;
out6 = c(2)*out6;
out4 = c(3)*out4;
out2 = c(4)*out2;
%
%Reconstruction
%
G = B(lh-n);
G1 = -(-1).^n.*B(n+1);
m3 = merge1(out5,out6,G,G1);
d4 = [zeros(lh-1,1);out4];
m2 = merge1 (m3,d4,G,G1);
d2 = [zeros(3*(lh-1),1);out2];
out = merge1(m2,d2,G,G1);
disp( 'The output of the filterbank is in the vector out')
```

```
function [out1,out2] = split1(in,b0,b1)
%
% function [out1,out2] = split1(in,b0,b1)
% For use together with merge1 and bank to realize a filterbank.

x1= conv(b0,in);
x2= conv(b1,in);
out1= x1(1:2:length(x1));
out2= x2(1:2:length(x2));
```

```
function out = merge1(in1,in2,b0,b1)
%
% function out = merge1(in1,in2,b0,b1)
% For use together with split1 and bank to realize a filterbank.

%
% Upsampling:
y1(1:2:2*length(in1))=2*in1;
y2(1:2:2*length(in2))=2*in2;
%
% Filtering:
z1=conv(b0,y1);
z2=conv(b1,y2);
%
% Addition:
maxlen=max([length(z1) length(z2)]);
z11 = [z1';zeros(maxlen-length(z1),1)];
z22 = [z2';zeros(maxlen-length(z2),1)];
out = z11+z22;
```

_SHANNO$DUA0:[TTTUSER.FS]BANK210.COM;1

```
$ matlab
rand('normal')
in =rand(1:7000);
F =firl(32,.8);
B=firl(127,.5059)';
realfilefast
7000
0.008
32
16
quality
erle1= qualityx(out22,e1);
erle2= qualityx(out44,e2);
erle3= qualityx(out66,e3);
erle4= qualityx(out55,e4);
save speech210 out e1 e2 e3 e4 xf h1 h2 h3 h4 mu len l dl F B
save erle210 erle erle1 erle2 erle3 erle4
exit
```

_SHANNO$DUA0:[TTTUSER.FS]REALFILEFAST.M;18

```
len=input('Length');
mu0=input('Stepsize mu0, 0<mu0<1, normalized! ');
%mu=input('Stepsize mu');
l=input('No of weights');
dl=input('Delay');
disp('Analyzing.........');
%rand('normal');
din=[zeros(dl,1);in'];
xf=filter(F,1,din);
clear din                          %We don't need din anymore. Save space!
lh=length(B);
n=[0:lh-1]';
B1=(-1).^n.*B(lh-n);
[out1,out2]=split1(in,B,B1);
clear in                           %We don't need in anymore. Save space!
[out3,out4]=split1(out1,B,B1);
[out5,out6]=split1(out3,B,B1);
[out11,out22]=split1(xf,B,B1);
[out33,out44]=split1(out11,B,B1);
[out55,out66]=split1(out33,B,B1);
clear out1
clear out3
clear out11
clear out33
%
% Adaptive filter and LMS simulation step;
%
pow1=sum(out2.^2)/length(out2);
pow2=sum(out4.^2)/length(out4);
pow3=sum(out6.^2)/length(out6);
pow4=sum(out5.^2)/length(out5);
mu1=mu0/(pow1*l)
mu2=mu0/(pow2*l)
mu3=mu0/(pow3*l)
mu4=mu0/(pow4*l)
disp('Adapting.........');
[h1,e1]=lmsf(out2,out22,mu1,l,1,len);
[h2,e2]=lmsf(out4,out44,mu2,l,1,len);
[h3,e3]=lmsf(out6,out66,mu3,l,1,len);
[h4,e4]=lmsf(out5,out55,mu4,l,1,len);
%
%Reconstruction
%
disp('Reconstructing.........');
G=B(lh-n);
G1=-(-1).^n.*B(n+1);
m3=merge1(e4,e3,G,G1);
d4=[zeros(lh-1,1);e2'];
m2=merge1(m3,d4,G,G1);
d2=[zeros(3*(lh-1),1);e1'];
out=merge1(m2,d2,G,G1);
disp('The output of the filterbank is in the vector out')
```

_SHANNO$DUA0:[TTTUSER.FS]LMSF.M;2

```
function [h, e] = lmsf(y,x,mu,n,l1,l2);
% function [h, e] = lmsf(y,x,mu,n,l1,l2);
% Transversal filter simulation algorithm that runs the
% LMS-algorithm on the input signal y, modifies the weights h
% to make the output signal follow the desired signal x.
% mu is the stepsize and n denotes the order of the
% transversal filter. Output is the weights (h) and the error (e).
% Observe that the
% algorithm starts updating coefficients at time=max(n,l1) and
% finishes at time=min(length(y),length(x),l2).
disp('iteration startar')
h = zeros(1:n);
ny = min(length(x),length(y));
nny=min(ny,l2);
e = [];
for i=n:ny,
  xhatn = h*(y(i:-1:(i-n+1)))';
  en = x(i)-xhatn;
  if i >= l1
     if i <= l2
        h = h + 2*mu*en*y(i:-1:(i-n+1));
     end;
  end;
  ii=i-n+1;
  e(ii) = en;
end
disp('iteration klar')
end
```

_SHANNO$DUA0:[TTTUSER.FS]QUALITY.M;12

```
% PROGRAM ERLE
%
%
xxf=filter(ones(128,1)/128,1,abs(xf.^2));
o=filter(ones(128,1)/128,1,abs(out.^2));
o = o(7*length(B):length(o));          %Adjust for the delay of the Filter Bank
o= o+0.00000001;
lxxf=length(xxf);
lo=length(o);
minl = min(lo,lxxf);
r=xxf(1:minl)./ o(1:minl);
erle=10*log10(abs(r)+.000000001);
```

_SHANNO$DUA0:[TTTUSER.FS]QUALITYX.M;5

```
function erle=qualityx(x,e)
% function erle=qualityx(x,e)
%
%
x=filter(ones(128,1)/128,1,abs(x.^2));
e=filter(ones(128,1)/128,1,abs(e.^2));
e=e+.000000001;
ml=min(length(x),length(e));
e=e(:);
r=r(:);
r=x(1:ml) ./ e(1:ml);
erle=10*log10(abs(r)+.000000001);
```

_SHANNO$DUA0:[TTTUSER.FS]REALFILEFAST1.M;4

```
len=input('Length');
mu0=input('Stepsize mu0, 0<mu0<1, normalized! ');
%mu=input('Stepsize mu');
l=input('No of weights');
dl=input('Delay');
disp('Analyzing.........');
%rand('normal');
din=[zeros(dl,1);in'];
xf=filter(F,1,din);
clear din                       %We don't need din anymore. Save space!
lh=length(B);
n=[0:lh-1]';
B1=(-1).^n.*B(lh-n);
[out1,out2]=split1(in,B,B1);
clear in                        %We don't need in anymore. Save space!
[out3,out4]=split1(out1,B,B1);
[out5,out6]=split1(out3,B,B1);
[out11,out22]=split1(xf,B,B1);
[out33,out44]=split1(out11,B,B1);
[out55,out66]=split1(out33,B,B1);
clear out1
clear out3
clear out11
clear out33
%
% Adaptive filter and LMS simulation step;
%
pow1=sum(out2.^2)/length(out2);
pow2=sum(out4.^2)/length(out4);
pow3=sum(out6.^2)/length(out6);
pow4=sum(out5.^2)/length(out5);
mu1=mu0/(pow1*l)
mu2=mu0/(pow2*.5*l)
mu3=mu0/(pow3*.25*l)
mu4=mu0/(pow4*.25*l)
disp('Adapting.........');
[h1,e1]=lmsf(out2,out22,mu1,l,1,len);
[h2,e2]=lmsf(out4,out44,mu2,.5*l,1,len);
[h3,e3]=lmsf(out6,out66,mu3,.25*l,1,len);
[h4,e4]=lmsf(out5,out55,mu4,.25*l,1,len);
%
%Reconstruction
%
disp('Reconstructing.........');
G=B(lh-n);
G1=-(-1).^n.*B(n+1);
m3=merge1(e4,e3,G,G1);
d4=[zeros(lh-1,1);e2'];
m2=merge1(m3,d4,G,G1);
d2=[zeros(3*(lh-1),1);e1'];
out=merge1(m2,d2,G,G1);
disp('The output of the filterbank is in the vector out')
```

```
$ matlab
rand('normal')
load farook;
in =(farook/1000)';
clear farook;
F =fir1(32,.8);
B=fir1(127,.5059)';
realfilefast1
17000
0.005
32
16
quality
erle1= qualityx(out22,e1);
erle2= qualityx(out44,e2);
erle3= qualityx(out66,e3);
erle4= qualityx(out55,e4);
save speech414 out e1 e2 e3 e4 xf h1 h2 h3 h4 mu0 len l dl F B
save erle414 erle erle1 erle2 erle3 erle4 mu1 mu2 mu3 mu4
exit
```

APPENDIX-D

_WIDROW$DUA0:[TTTUSER.FS]H00.;1

```
 1.5308859e-04
-3.8216094e-04
-1.4781471e-04
 4.2257955e-04
 1.4861901e-04
-4.8890645e-04
-1.5340511e-04
 5.8408147e-04
 1.5971302e-04
-7.1100694e-04
-1.6471336e-04
 8.7254241e-04
 1.6520000e-04
-1.0715072e-03
-1.5757963e-04
 1.3106930e-03
 1.3785600e-04
-1.5928891e-03
-1.0160575e-04
 1.9209233e-03
 4.3942216e-05
-2.2977237e-03
 4.0538502e-05
 2.7264070e-03
-1.5783627e-04
-3.2104011e-03
 3.1466096e-04
 3.7536157e-03
-5.1859656e-04
-4.3606764e-03
 7.7833692e-04
 5.0372493e-03
-1.1040257e-03
-5.7904959e-03
 1.5077499e-03
 6.6297180e-03
-2.0042655e-03
-7.5672936e-03
 2.6120820e-03
 8.6200650e-03
-3.3551191e-03
-9.8114593e-03
 4.2653070e-03
 1.1174838e-02
-5.3868052e-03
-1.2759004e-02
 6.7831342e-03
 1.4637692e-02
-8.5498460e-03
-1.6926888e-02
 1.0838475e-02
 1.9818743e-02
-1.3905496e-02
-2.3654201e-02
 1.8223137e-02
 2.9098108e-02
-2.4767502e-02
-3.7637535e-02
 3.5935318e-02
 5.3408839e-02
-5.9603802e-02
-9.3794482e-02
 1.4566938e-01
 4.5434241e-01
 4.5434241e-01
 1.4566938e-01
```

```
-9.3794482e-02
-5.9603802e-02
 5.3408839e-02
 3.5935318e-02
-3.7637535e-02
-2.4767502e-02
 2.9098108e-02
 1.8223137e-02
-2.3654201e-02
-1.3905496e-02
 1.9818743e-02
 1.0838475e-02
-1.6926888e-02
-8.5498460e-03
 1.4637692e-02
 6.7831342e-03
-1.2759004e-02
-5.3868052e-03
 1.1174838e-02
 4.2653070e-03
-9.8114593e-03
-3.3551191e-03
 8.6200650e-03
 2.6120820e-03
-7.5672936e-03
-2.0042655e-03
 6.6297180e-03
 1.5077499e-03
-5.7904959e-03
-1.1040257e-03
 5.0372493e-03
 7.7833692e-04
-4.3606764e-03
-5.1859656e-04
 3.7536157e-03
 3.1466096e-04
-3.2104011e-03
-1.5783627e-04
 2.7264070e-03
 4.0538502e-05
-2.2977237e-03
 4.3942216e-05
 1.9209233e-03
-1.0160575e-04
-1.5928891e-03
 1.3785600e-04
 1.3106930e-03
-1.5757963e-04
-1.0715072e-03
 1.6520000e-04
 8.7254241e-04
-1.6471336e-04
-7.1100694e-04
 1.5971302e-04
 5.8408147e-04
-1.5340511e-04
-4.8890645e-04
 1.4861901e-04
 4.2257955e-04
-1.4781471e-04
-3.8216094e-04
 1.5308859e-04
```

```
 1.5308859e-04
 3.8216094e-04
-1.4781471e-04
-4.2257955e-04
 1.4861901e-04
 4.8890645e-04
-1.5340511e-04
-5.8408147e-04
 1.5971302e-04
 7.1100694e-04
-1.6471336e-04
-8.7254241e-04
 1.6520000e-04
 1.0715072e-03
-1.5757963e-04
-1.3106930e-03
 1.3785600e-04
 1.5928891e-03
-1.0160575e-04
-1.9209233e-03
 4.3942216e-05
 2.2977237e-03
 4.0538502e-05
-2.7264070e-03
-1.5783627e-04
 3.2104011e-03
 3.1466096e-04
-3.7536157e-03
-5.1859656e-04
 4.3606764e-03
 7.7833692e-04
-5.0372493e-03
-1.1040257e-03
 5.7904959e-03
 1.5077499e-03
-6.6297180e-03
-2.0042655e-03
 7.5672936e-03
 2.6120820e-03
-8.6200650e-03
-3.3551191e-03
 9.8114593e-03
 4.2653070e-03
-1.1174838e-02
-5.3868052e-03
 1.2759004e-02
 6.7831342e-03
-1.4637692e-02
-8.5498460e-03
 1.6926888e-02
 1.0838475e-02
-1.9818743e-02
-1.3905496e-02
 2.3654201e-02
 1.8223137e-02
-2.9098108e-02
-2.4767502e-02
 3.7637535e-02
 3.5935318e-02
-5.3408839e-02
-5.9603802e-02
 9.3794482e-02
 1.4566938e-01
-4.5434241e-01
 4.5434241e-01
-1.4566938e-01
```

_WIDROW$DUA0:[TTTUSER.FS]H1.;1

```
-9.3794482e-02
 5.9603802e-02
 5.3408839e-02
-3.5935318e-02
-3.7637535e-02
 2.4767502e-02
 2.9098108e-02
-1.8223137e-02
-2.3654201e-02
 1.3905496e-02
 1.9818743e-02
-1.0838475e-02
-1.6926888e-02
 8.5498460e-03
 1.4637692e-02
-6.7831342e-03
-1.2759004e-02
 5.3868052e-03
 1.1174838e-02
-4.2653070e-03
-9.8114593e-03
 3.3551191e-03
 8.6200650e-03
-2.6120820e-03
-7.5672936e-03
 2.0042655e-03
 6.6297180e-03
-1.5077499e-03
-5.7904959e-03
 1.1040257e-03
 5.0372493e-03
-7.7833692e-04
-4.3606764e-03
 5.1859656e-04
 3.7536157e-03
-3.1466096e-04
-3.2104011e-03
 1.5783627e-04
 2.7264070e-03
-4.0538502e-05
-2.2977237e-03
-4.3942216e-05
 1.9209233e-03
 1.0160575e-04
-1.5928891e-03
-1.3785600e-04
 1.3106930e-03
 1.5757963e-04
-1.0715072e-03
-1.6520000e-04
 8.7254241e-04
 1.6471336e-04
-7.1100694e-04
-1.5971302e-04
 5.8408147e-04
 1.5340511e-04
-4.8890645e-04
-1.4861901e-04
 4.2257955e-04
 1.4781471e-04
-3.8216094e-04
-1.5308859e-04
```

```
 9.3427708e-04
 1.0611036e-17
-1.5350895e-03
 3.6635038e-03
-5.4101162e-03
 4.8317021e-03
-1.1335696e-17
-9.3481367e-03
 2.0407843e-02
-2.7199294e-02
 2.2299150e-02
-2.4777760e-17
-4.0420055e-02
 9.2966646e-02
-1.4587595e-01
 1.8520384e-01
 7.9896336e-01
 1.8520384e-01
-1.4587595e-01
 9.2966646e-02
-4.0420055e-02
-2.4777760e-17
 2.2299150e-02
-2.7199294e-02
 2.0407843e-02
-9.3481367e-03
-1.1335696e-17
 4.8317021e-03
-5.4101162e-03
 3.6635038e-03
-1.5350895e-03
 1.0611036e-17
 9.3427708e-04
```

79626
6/5/91
লাইব্রেরী, ঢাকা