

CRANFIELD UNIVERSITY



**COLLEGE OF AERONAUTICS
ELECTRONIC SYSTEM DESIGN DEPARTMENT
Msc Thesis**

Academic Year 1994 - 1995

ADAPTIVE FILTER DESIGN

Supervisor : Dr. S.J. Handley

A. Esuola

September 1995

This thesis is submitted in partial fulfilment of the requirements for the Degree of Master of Science

TO WHOM IT MAY CONCERN



Following the revision of this thesis, the following have been taken into account-

- a) All pages have been reformatted to meet the Cranfield University standard
- b) Chapter separator have been removed
- c) Page numbering have been adjusted as well as contents list.
- d) The entire graphs have been reproduced and are now legible.

Amongst other corrections the following have been made -

The filter's coefficient previously on page 51 now appears on page 24. Also previously on page 44, it mentioned that the specification of the filter can be met with a second order analogue filter was valid. However, this have been reworded and now appear in section 4.4.2 page 19-22. Chapter 5 have been rewritten and now appears as chapter 8.

However, the following addition have been made,

chapter 6 - Simulation subdivided into

- a) Coupled system
- b) Uncoupled system

Finally more results are been made available on the programming diskette, which cannot be included in this thesis primarily because of the time scale and printing difficulties coupled with cost.

ABSTRACT - A heart pulse signal was sampled using a ECG single ended amplifier. The heart pulse signal was reconstructed after appropriate filtering, displaying this simultaneously with the acquired signal, on the oscilloscope.

In some applications, it is necessary to perform simultaneous measurement on weak signals in extraneous environment, tracking and eliminating the unwanted signals, such as the mains frequency, and making comparison of discernible parameters.

This project was aimed at designing and implementing an adaptive filter, using adaptive noise cancelling technique to track and eliminate the mains' noise frequency in a ECG signal, in real-time.

Furthermore a model was developed for the system and was simulated to obtain an optimum value of μ , a tracking parameter, for the adaptive filter. Consequently software was written to implement the design.

ACKNOWLEDGEMENTS

I would like to thank Dr. S.J Handley, my supervisor for his invaluable suggestions and advice.

I extend my sincere gratitude to John Waytt, for making the laboratory equipment available as necessary and offering suggestions in debugging of the programming codes.

To all my friends, I say thank you. I say thanks to Dr. Ikpe, for his time and suggestion, thanks to Mide for making available his computing facilities, Emeka Lawson for offering suggestions to programming codes, Steve Welch, for providing printing and work space facilities.

I also say thank you to Tandra Gill, for showing interest in my academics and wishing me well at all times.

Finally, I give Glory to God.

CONTENTS

		<i>Pages</i>
<i>Abstract</i>		<i>i</i>
<i>Acknowledgements</i>		<i>ii</i>
<i>Contents</i>		<i>iii-iv</i>
<i>List of figures</i>		<i>v-ix</i>
<i>List of symbols</i>		<i>x-xi</i>
Chapter 1	Introduction	1-2
Chapter 2	Application of adaptive filters	3-4
2.1	Introduction	3
2.2	Objectives	4
Chapter 3	Principal theories	5-14
3.1	Introduction	5
3.2	Digital filters	5
3.3	Types of digital filters	5-7
3.4	Finite Impulse Response (FIR)	7-12
3.5	Infinite Impulse Respons (IIR)	12
3.6	Sampling theorem	12-13
3.7	Adaptive algorithm	13-14
Chapter 4	Implementation	15-32
4.1	Introduction	15
4.2	Equipment	15
4.3	Settings	15-16
4.4	Initial work	17-27
4.5	Adaptive filter theory	27-32
Chapter 5	Differential Equation Development	33-35
Chapter 6	Model Simulation	36-47
6.1	Introduction	36
6.2	Coupled system	36-41
6.3	Uncoupled system	41-47
Chapter 7	Software Design	48-54
7.1	Introduction	48
7.2	Main program	48-51

7.3	Design filter	51
7.4	Off-line digital filtering	52
7.5	Real-time digital filtering	53
7.6	Adaptive	53-54
Chapter 8	Testing and Result	55-87
8.1	Introduction	55
8.2	Initial testing	55
8.3	Software testing and signal acquisition	56-57
8.4	Acquisition of physiological signal	58
8.5	Acquisition of noisy physiological signal	59
8.6	Acquisition of ECG signal	60-62
8.7	Real-time testing and filtering response	63-77
8.8	Investigation of adaptive parameter, μ	77-80
8.9	Adaptive filtering	80-87
Chapter 9	Conclusions	89-90
	Scope for further studies	91
	References	
	Appendix A Derivation of differential equation	
	Appendix B Estimation of wordlength	
	Appendix C Flow chart	
	Appendix D Gain as defined by variables	

LIF

600

600

600

600

600

600

600

IN MEMORY OF MY FATHER DARAMOLA OLAYIMIPO ESUOLA

LIST OF FIGURES***Pages***

Fig 3.1	Analogue to digital and digital to analogue conversion	5
Fig 3.2	Conceptual representation of a digital filter	6
Fig 3.3	Illustration of a tolerance scheme for a lowpass filter	7
Fig 3.4	Direct realisation of a fourth - order IIR filter	9
Fig 3.5	Cascade realisation of a fourth - order IIR filter	9
Fig 3.6	Parallel realisation of a fourth - order filter	10
Fig 3.7	Realisation structure for FIR filter	10
Fig 3.8(a)	Illustration of Frequency sampling structure for a FIR filter	11
Fig 3.8(b)	Illustration of fast convolution technique for FIR filters	11
Fig 3.9	An example of aliasing in the time domain	13
Fig 4.1	Illustration of an ideal impulse response filter	17
Fig 4.2	Cascade structure for an IIR bandpass filter	22
Fig 4.3	Illustration of notch filter design using pole-placement method	24
Fig 4.4	Illustration of the cononical structure for notch filter design	25
Fig 4.5	Illustration of stable regions within a triangle	26
Fig 4.6	Illustration of adaptive noise cancelling configuration	28
Fig 4.7	Illustration of error performance surface	30
Fig 6.1	Illustration of analogue simulation diagram for the model	37
Fig 6.2	Illustration of the simulation for the system at 18kHz (case1) [coupled system]	37
Fig 6.3	Illustration of the simulation for the system at 18kHz (case2)	38
Fig 6.4	Illustration of the simulation for the system at 6Hz (case 8)	38
Fig 6.5	Illustration of the simulation for the system at 6Hz (case 9)	39

Fig 6.6	Illustration of the simulation for the system at 6Hz (case 21)	40
Fig 6.7	Illustration of the simulation for the system at 6Hz (case 22)	40
Fig 6.8	Illustration of the simulation for the system at 6Hz (case 24)	41
Fig 6.9	Illustration of the simulation for the system at 6Hz (case 7) [decoupled system]	42
Fig 6.10	Illustration of the simulation for the system at 6Hz (case 8)	43
Fig 6.11	Illustration of the simulation for the system at 6Hz (case 9)	43
Fig 6.12	Illustration of the simulation for the system at 6Hz (case 21)	44
Fig 6.13	Illustration of the simulation for the system at 6Hz (case 22)	44
Fig 6.14	Illustration of the simulation for the system at 6Hz (case 23)	45
Fig 6.15	Illustration of the simulation for the system at 6Hz (case 24)	45
Fig 6.16	Illustration of the simulation for the system using a square wave	46
Fig 6.17	Illustration of the simulation for the system using a square wave	46
Fig 6.18	Illustration of the simulation for the system using a square wave	47
Fig 6.19	Illustration of the simulation for the system using a square wave	47
Fig 7.1	Illustration of the structural diagram for adaptive filter design	49
Fig 7.2	Illustration of flowchart diagram for displaying a menu	50
Fig 7.3(a)	Illustration of window program design	51
Fig 7.3(b)	Illustration of window program design continued	52
Fig 7.4	Adaptive filter algorithm	54
Fig 8.1	Illustration of experimental setup to test for input signal	55
Fig 8.2(a)	Illustration of experimental setup for simultaneous display of input and output signal	56
Fig 8.2(b)	Illustration of simultaneous display of sampled data in real-time	57
Fig 8.2(c)	Illustration of the acquired 50Hz reference signal	58

Fig 8.3(a)	Illustration of experimental setup for acquisition of low frequency signal	58
Fig 8.3(b)	Illustration of the wave profile of the acquired signal	59
Fig 8.4(a)	Illustration of experimental setup to acquire a noisy simulated physiological signal	59
Fig 8.4(b)	Illustration of the simulated physiological signal	60
Fig 8.5(a)	Illustration of experimental setup for acquisition of ECG signal sampled at 18kHz	60
Fig 8.5(b)	Illustration of ECG signal acquired at 18kHz	61
Fig 8.5(c)	Illustration of ECG signal acquired at 250Hz	61
Fig 8.5(d)	Illustration of experimental setup for simultaneous display of the ECG signal	62
Fig 8.5(e)	Illustration of simultaneous display of ECG signal in real-time	62
Fig 8.6	Illustration of lowpass filter response	63
Fig 8.7	Illustration of bandpass filter response	64
Fig 8.8(a)	Illustration of notch filter response	65
Fig 8.8(b)	Illustration of the magnituded response for the notch filter	66
Fig 8.8(c)	Illustration of the phase response for the notch filter	66
Fig 8.9(a)	Illustration of experimental setup for lowpass filtering	67
Fig 8.9(b)	Illustration of action of the lowpass filter on the ECG acquired at 18kHz	67
Fig 8.9(c)	Illustration of nagnified ECG signal	68
Fig 8.9(d)	Illustration of action of the lowpass filter on the ECG acquired at 250Hz	69
Fig 8.9(e)	Illustration of several peaks in ECG signal	69
Fig 8.9(f)	Illustration of lowpass filter on simulated physiological signal	70

Fig 8.10	Illustration of experimental setup for bandpass filtering	70
Fig 8.11(a)	Illustration of action of bandpass filter on a ECG signal acquired at 18kHz.	71
Fig 8.11(b)	Illustration of action bandpass filter on a ECG signal acquired at 250Hz	71
Fig 8.11(c)	Illustration of P,R, and T peaks in a ECG signal	72
Fig 8.12(a)	Action of bandpass filter on the reference signal	72
Fig 8.12(b)	Action of bandpass filter on the simulated signal	73
Fig 8.13	Illustration of experimental setup for notch filtering	73
Fig 8.14(a)	Illustration of action of notch filter on a ECG signal sampled at 18kHz	74
Fig 8.14(b)	Illustration of magnified ECG signal sampled at 18kHz after notch filtering	74
Fig 8.14(c)	Further magnification of the ECG signal sampled at 18kHz after notch filtering	75
Fig 8.15(a)	Illustration of combined effects of notch and bandpass filtering on ECG signal sampled at 18kHz	76
Fig 8.15(b)	Magnified ECG signal after notch and lowpass filtering	76
Fig 8.15(c)	Illustration of combined effect of lowpass and bandpass filters on ECG signal acquired at 250Hz	77
Fig 8.16(a)	Illustration of the vectors of the ECG signal sampled at 250Hz	79
Fig 8.16(b)	Illustration of the covariance matrix of the ECG signal sampled at 250Hz	79
Fig 8.16(c)	Illustration of the eigenvalues of the covariance matrix of the ECG signal sampled at 250Hz	80
Fig 8.17(a)	Illustration of the estimate of the noise signal	81
Fig 8.17(b)	Illustration of the error in estimating the noise signal	82
Fig 8.17(c)	Illustration of the new weight generated by the adaptive filter	82

Fig 8.17(d)	Illustration of the action of the bandpass filter on an ECG sampled at 250Hz	83
Fig 8.17(e)	Illustration of the estimate of the noise signal	84
Fig 8.17(f)	Illustration of the error in estimating the noise signal	84
Fig 8.17(g)	Illustration of the new weight generated by the adaptive algorithm employing the notch filter	85
Fig 8.17(h)	Illustration of the action of the adaptive filter on a ECG signal acquired at 18kHz	85
Fig 8.17(i)	Illustration of the estimate of the noise signal employing a lowpass filter	86
Fig 8.17(j)	Illustration of the error in estimating the noise signal in the corrupted signal	86
Fig 8.17(k)	Illustration of the new weight generated by the filter	87
Fig 8.17(l)	Illustration of action of the new adaptive filter on ECG acquired at 250Hz	87

LIST OF SYMBOLS

A	-	Amplitude of the signal
$A_{\min}$	-	Minimum attenuation
$\dot{x}(t)$	-	State derivative of $\underline{x}$
$\underline{x}(t)$	-	State vector
x_1, x_2	-	State variable
$x(t), x_k$	-	Input signal
$x(n-k)$	-	Delay in input signal by a factor k
$\dot{y}(t)$	-	Derivative/State derivative of vector $\underline{y}$
$\ddot{y}(t)$	-	Second derivative of y
q	-	differential operator
$y(n-k)$	-	Delay in output signal by a factor k
H(z)	-	Transfer function for discrete time system
$H_D(\omega)$	-	Ideal response
β	-	Ripple parameter
μ	-	Tracking and convergence parameter
bw	-	Bandwidth
!	-	Factorial
$I_0(x)$	-	Zero order modified Bessel function
$\Sigma, \int$	-	Summation
$\Rightarrow$	-	Implies
$\therefore$	-	Therefore
$h_D(\omega)$	-	Practical response
z^{-k}	-	Delay in z by a factor k
H(z)	-	Transfer function of a discrete time system
z	-	Polynomial in z
H(s)	-	Transfer function of a continuous time system
s	-	Polynomial in s
Δf	-	Normalise frequency

f_s	-	Sampling frequency
$f_{\max}$	-	Maximum frequency
f_c	-	Cut-off frequency
T	-	Period
T_s	-	Sampling period
$f(\xi)$	-	Probability density function
τ	-	Sampling interval
$r(t), r_k$	-	Reference signal
r	-	Radius
R	-	Auto-correlation vector
∇_k	-	True gradient vector
∇	-	Gradient vector
P	-	Cross-Correlation vector, Covariance matrix
W	-	Weight vector
W_{opt}	-	Optimum coefficient or weight
$W(n)$	-	Window function
J	-	Mean Square Error
$E[.$	-	Expectation
N	-	Filter length
$n(t), n_k$	-	Noise in the signal
$\hat{n}_k$	-	Estimated noise in the signal
$\hat{s}_k$	-	Estimate of the desired signal
$s(t), s_k$	-	Desired signal
ω_o	-	Centre frequency
ω_1	-	Lower band edge frequency
ω_2	-	Upper band edge frequency
δ_p	-	Passband attenuation
δ_s	-	Stopband attenuation
$\#$	-	Channel number

LIST OF SYMBOLS

A	-	Amplitude of the signal
$A_{\min}$	-	Minimum attenuation
$\dot{x}(t)$	-	State derivative of $\underline{x}$
$\underline{x}(t)$	-	State vector
x_1, x_2	-	State variable
$x(t), x_k$	-	Input signal
$x(n-k)$	-	Delay in input signal by a factor k
$\dot{y}(t)$	-	Derivative/State derivative of vector $\underline{y}$
$\ddot{y}(t)$	-	Second derivative of y
q	-	differential operator
$y(n-k)$	-	Delay in output signal by a factor k
H(z)	-	Transfer function for discrete time system
$H_D(\omega)$	-	Ideal response
β	-	Ripple parameter
μ	-	Tracking and convergence parameter
bw	-	Bandwidth
!	-	Factorial
$I_0(x)$	-	Zero order modified Bessel function
$\Sigma, \int$	-	Summation
$\Rightarrow$	-	Implies
$\therefore$	-	Therefore
$h_D(\omega)$	-	Practical response
z^{-k}	-	Delay in z by a factor k
H(z)	-	Transfer function of a discrete time system
z	-	Polynomial in z
H(s)	-	Transfer function of a continuous time system
s	-	Polynomial in s
Δf	-	Normalise frequency

f_s	-	Sampling frequency
$f_{\max}$	-	Maximum frequency
f_c	-	Cut-off frequency
T	-	Period
T_s	-	Sampling period
$f(\xi)$	-	Probability density function
τ	-	Sampling interval
$r(t), r_k$	-	Reference signal
r	-	Radius
R	-	Auto-correlation vector
∇_k	-	True gradient vector
∇	-	Gradient vector
P	-	Cross-Correlation vector, Covariance matrix
W	-	Weight vector
W_{opt}	-	Optimum coefficient or weight
$W(n)$	-	Window function
J	-	Mean Square Error
$E[$	-	Expectation
N	-	Filter length
$n(t), n_k$	-	Noise in the signal
$\hat{n}_k$	-	Estimated noise in the signal
$\hat{s}_k$	-	Estimate of the desired signal
$s(t), s_k$	-	Desired signal
ω_o	-	Centre frequency
ω_1	-	Lower band edge frequency
ω_2	-	Upper band edge frequency
δ_p	-	Passband attenuation
δ_s	-	Stopband attenuation
$\#$	-	Channel number

CHAPTER 1 INTRODUCTION

Recent advances in adaptive digital filtering, have found a widespread application in, for example, adaptive control, communications systems, medicine and digital signal processing, to mention a few.

Fundamental connections between adaptive filtering, identification, and control have been widely acknowledge since their inception in the late 1960s. However, early on these specialisations spawned separate literature and experts. Explicit attempts at rapprochement emerged in the latter half of the 1970s. Three successful applications of adaptive filtering were reported in communication systems (network echo cancellation, differential pulse code modulation and linear channel equalisation) that emerged concurrently in the 1960s, [1].

Recently, the application of adaptive filtering has gained ground in the field of medicine. This interest seems to have stemmed from the difficulty posed by selective reception of weak signals echoed by different tissues samples, where all the samples have been subjected to a degree of interference, such as the powerline noise. This has been of great interest in the biomedical research field. One of such application is the human Electroencephalogram (EEG), [2]. This is a process whereby the useful diagnostic information on variety of neurological disorders can be obtained from the electrical activity of the brain. Other applications includes foetal monitoring - cancelling of maternal Electrocardiogram (ECG), has also been reported, [3], [10].

Domestic and commercial applications have exploited this technology to a great extent. Loudspeaking telephones employ the hybrid network to separate the transmit and receive paths. Teleconferencing networks also exploit the use of adaptive filtering to obviate similar problem in the loudspeaking telephone.

Another extensive application has been observed in the military sphere. Extremely Low Frequency (ELF) communication with submarines has been made possible, [4]. Several model, such as *Hidden Markov model* (HHM) have been suggested for tracking multiple tones sources which vary with time, an important technique in radar signal processing.

The objective of this project, focuses on practical problems which demand the use of adaptive filtering technique. One of such problem is the convenient elimination of powerline noise from a biomedical signal. This has been the focus of researchers for some time, [5], [6], [7], [8].

While proper grounding and electrical shielding is paramount in analogue recordings, the modern use of computers has suggested the investigation of digital approaches to solve the powerline interference problems. In early applications, powerline noise was removed by either a lowpass or bandstop and a notch filter with proper cut-off frequencies. A lowpass filter is adequate only when the true signal has all frequency content below the cut-off frequency. This is usually not the case for general biomedical signal processing. When the frequency of the powerline noise is stable, a fixed notch filter is sufficient. However, the frequency of the powerline noise is not always stable at exactly 50Hz. This frequency variation is always assumed to be Gaussian [9]. This suggests that the distribution may or may not be skewed in one direction.

A fixed notch filter may eliminate the noise when the distribution is centered exactly at the frequency for which the filter was designed. However, if the distribution is not centred at the frequency of the fixed notch filter, it may be ineffective.

Based on incremental estimation, a local self-corrective method was proposed for powerline noise elimination in biomedical signal processing, [10]. The method was simple, functioning in real time, and suitable for a microprocessor. It was later pointed out that the method is actually equivalent to a fixed notch filter where only the bandwidth is being adapted, [11].

In this project, the biomedical source was assumed to be a low frequency signal, in the region of (2-10)Hz. The interfering signal of the powerline source was assumed to have a centre frequency which varies between (45-55)Hz. These signals are supplied by a Thandar TG2002 20MHz sweep/function generator.

Initial work began by designing a digital lowpass filter to accept frequencies of up to 20Hz. This ensures that all signal of interest remain intact. The filtering process continued by designing a digital notch filter with centre frequency at 50Hz and a bandwidth of ± 5 Hz to remove the powerline noise interference. Finally a digital bandpass filter was applied to pass the signal of interest i.e (2-10Hz).

To achieve real-time adaptive filtering, the *Least Mean Square*(LMS) was favoured. As a consequence, a differential equation was developed for the noise cancellation adaptive algorithm configuration. The discernible parameter from the solution of the differential equation, is the bandwidth of the notch filter, designated as μ . Where μ constitutes the adaptive parameter, providing the variation in the centre frequency is not too erratic.

2.1 INTRODUCTION

Why *adaptive filters*? The contamination of a signal of interest by unwanted, larger signals or noise is a problem often encountered in many applications. Where the signal and noise occupy fixed and separate frequency bands, conventional linear filters with fixed coefficients are normally used to extract the signal. However, there are many instances when it is necessary for the filter characteristics to be variable, adapting to changing signal characteristics, or to be altered intelligently. In such a case the coefficients of the filter must vary and cannot be specified in advance. Such is the case when the spectral overlap between the signal and noise or if the band occupied by the noise is unknown or varies with time. Typical applications where fixed coefficient filters are inappropriate are the following -

2.1.1 Foetal monitoring

Cancelling of maternal electrocardiogram (ECG) during labour. Information derived from the foetal ECG, such as the foetal heart rate pattern, is valuable in assessing the condition of the baby before or during child birth, [2].

2.1.2 Digital communication

Using a spread spectrum where a larger jamming signal, possibly intended to disrupt communication, could interfere with the desired signal. The interference often occupies a narrow but unknown band within the wideband spectrum, and can only be effectively dealt with adaptively, [10].

2.1.3 In digital data communication

Over the telephone channel at a high rate. Signal distortions caused by the poor amplitude and phase response characteristic channel lead to pulses representing digital codes to interfere with each other (intersymbol interference), making it difficult to detect the codes reliably at the receiving end. To compensate for channel distortions which may be varying with time or of unknown characteristics at the receiving end, adaptive equalisation is used, [1].

2.1.4 Electroencephalograph(EEG)

where artefacts or signal contamination produced by eye movements or blinks is much larger than the genuine electrical activity of the brain and shares the same frequency band with signals of clinical interest. It is not possible to use conventional linear filters to remove the artefacts while preserving the signals of clinical interest, [2].

The above applications is not an exhaustive list of adaptive digital filters.

The task at hand requires that the desired signal (the biomedical signal) should be separated from the unwanted signal or noise. Furthermore this project also requires that a variation in the frequency mains should be tracked accordingly in real-time as specified in the objectives to be stated below.

2.2 OBJECTIVES

The main objectives of this project is to filter the powerline mains interfering signal from a given source in real-time.

2.3 REQUIREMENTS AND SPECIFICATIONS

Required to design and obtain the coefficient of an FIR passband filter to meet the following specifications -

passband edge frequency	2 - 10Hz
transition width	5Hz
stop band attenuation	>30dB
sampling frequency	250Hz

Lowpass filter with the following specifications -

passband	20Hz
transition width	15Hz
stopband attenuation	>15dB
sampling frequency	250Hz
use butterworth characteristics	

A notch filter with the following specification -

notch frequency	50Hz
attenuation at notch	>60dB
bandwidth	± 5 Hz
sampling frequency	250Hz

The above specifications are typical values employed in the on going research activities at Plymouth University.

2.4 CHOICE OF THE HIGH LEVEL LANGUAGE

The choice of high level language is not crucial. However, it is convenient working with a high level programming language such as C, Ada or Modular-2 as opposed to assembly language.

However, the C language was chosen because of its wide acceptance in the industry.

CHAPTER 3 PRINCIPAL THEORIES

3.1 INTRODUCTION

The aim of this chapter is to overview the theories surrounding digital filters and consequently adaptive filters, and to establish a relationship between discernible quantities.

3.2 DIGITAL FILTER

A filter is essentially a system or network that selectively changes the waveshape, amplitude-frequency or phase-frequency characteristics of a signal in some desired manner. Common filter objectives are to improve the quality of a signal (for example, to remove or reduced noise), to extract information from signals or to separate two or more signals previously combined to make, for example, efficient use of an available communication channel.

A digital filter, as it shall be seen later, is a mathematical algorithm implemented in hardware or software that operates on a digital input signal to produce a digital output signal for the purpose of achieving a filtering objective. Digital filter refers to the specific hardware or software routines that performs a filtering algorithm. Digital filter often operates on digitised analogue signals or just numbers representing some variables stored in a computer memory.

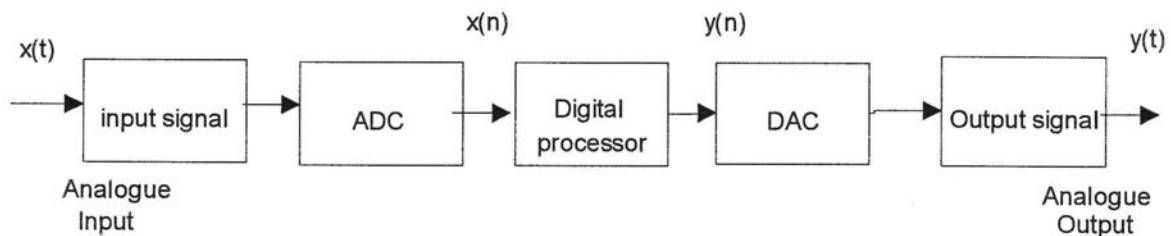


Fig 3.1

Analogue-to-Digital Digital-to-Analogue Conversion

The figure 3.1 illustrates a simplified block diagram showing how a real-time signal would be sampled and reconstructed in the later part of this task, via an analogue-to-digital, (ADC) and digital-to-analogue, (DAC) converters, verifying that the input and output signals are identical, section 8.3.

3.3 TYPES OF DIGITAL FILTER

Digital filters are broadly divided into two classes namely infinite impulse response (IIR) and finite impulse response (FIR) filters. Either type of filter, in its basic form, can be represented by its impulse response sequence, $a(k)$ ($k = 0, 1, 2, \dots$), as in figure 3.2.

$$y(n) = \sum_{k=0}^{\infty} a(k)x(n-k) \quad \dots\dots\dots (3.3.1)$$

$$y(n) = \sum_{k=0}^{N-1} a(k)x(n-k) \quad \dots\dots\dots (3.3.2)$$

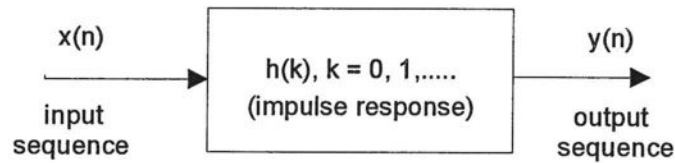


Fig 3.2
Conceptual representation of a digital filter

The input and output signals are related by linear convolution, which is given in equations (3.3.1) for the IIR and in (3.3.2) for the FIR filter.

It is evident from these equations that for an IIR filter, the impulse response is of infinite duration whereas for an FIR filter, it is of finite duration; since $a(k)$ for the FIR has only N values. In practice it is not feasible to compute the output of the IIR using equation 3.3.1, because the length of its impulse response is too long (i.e infinite in theory). Instead the IIR filtering equation is expressed in the recursive form:

$$y(n) = \sum_{k=0}^N a_k x(n-k) - \sum_{k=1}^M b_k y(n-k) \quad \dots\dots\dots (3.3.3)$$

where the a_k and b_k are the coefficient of the filter. Thus equation 3.3.2 and 3.3.3 are the difference equations for the FIR and IIR filters respectively. These equations, and in particular the values of $a(k)$, for the FIR, or a_k and b_k , for the IIR, are often very important objectives of most filter design problems.

Alternative expressions for the FIR and IIR filters are given in equations 3.3.4 and 3.3.5 respectively. These are the transfer functions for these filters and are very useful in evaluating their frequency responses.

$$H(z) = \sum_{k=0}^{N-1} a(k)z^{-k} \quad \dots\dots\dots (3.3.4)$$

$$H(z) = \sum_{k=0}^N a_k z^{-k} / (1 + \sum_{k=1}^M b_k z^{-k}) \quad \dots\dots\dots (3.3.5)$$

3.4 FINITE IMPULSE RESPONSE (FIR)

Filter design steps - The design of digital filters involves five steps:

- * *specification of filter requirements;*
- * *calculation of suitable filter coefficients;*
- * *representation of the filter by a suitable structure (realisation);*
- * *analysis of the effects of finite wordlength of the filter performance;*
- * *implementation of filter in software and hardware*

These steps are not necessarily independent; nor are they always performed in the order given.

3.4.1 Requirement specifications

This includes specifying the - signal characteristics type of signal source and sink, I/O interface, data rates width, and highest frequency of interest, (see section 2.3).

The characteristics of the filter - the desired amplitude and phase responses and their tolerances (if any), the speed of operation and modes of filtering (real-time or batch) need to be specified.

The manner of implementation - for example, as a high level language routine in a computer or as a DSP processor-based system, and other design constraints (for example, the cost of the filter). The designer may not have enough information to specify the filter completely at the outset, but as many of the filter requirements as possible should be specified to simplify the design process.

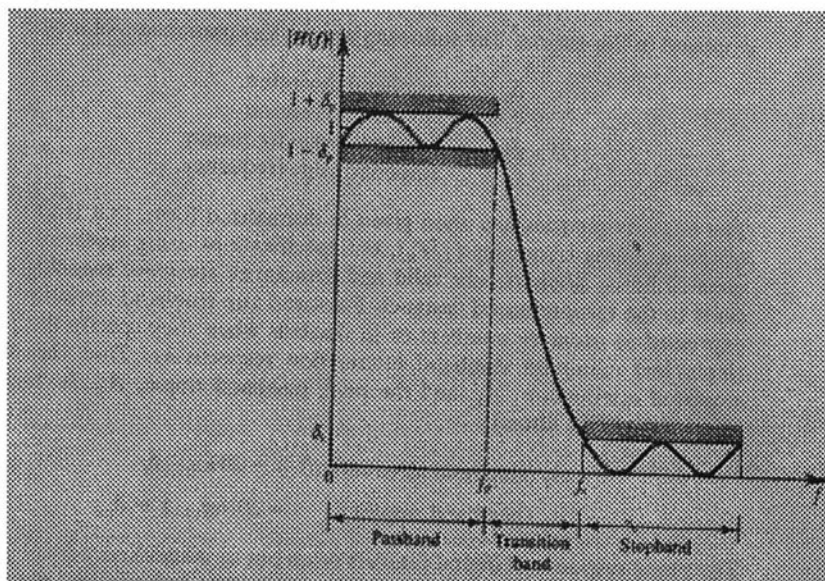


fig 3.3
illustration of a tolerance scheme for a lowpass filter, [2].

For frequency selective filters, such as lowpass and bandpass filters, the specifications are often in the form of tolerance scheme. Figure 3.3 depicts such a scheme for a lowpass filter. The shaded horizontal lines indicate the tolerance limits. In the passband, the magnitude response has a peak deviation of δ_p and, in the stopband, it has a maximum deviation of δ_s . The width of the transition width determines how sharp the filter is. The magnitude response should decrease monotonically from the passband to the stop band region.

3.4.2 Coefficient calculation

In step, we select one of a number of approximation methods and calculate the values of the coefficients, $a(k)$, for FIR, or a_k and b_k , for IIR, such that the filter characteristics given in section 2.3 are satisfied. The method used to calculate the filter coefficients depends on whether the filter is IIR or FIR type.

Calculation of IIR filter coefficients are traditionally based on transformation of known analogue filter characteristics into equivalent digital filters. The two basic methods used are the impulse invariant and the bilinear transformation methods. However, discussion will only be limited to the bilinear method because of its wide acceptability and is well suited to the calculation of coefficients of frequency selective filters. Furthermore digital filters resulting from the bilinear transform method will, in general, preserve the magnitude response characteristics of the analogue filter but not the time domain properties.

The pole-zero placement method offers an alternative approach to calculating the coefficients of IIR filters. It is an easy way of calculating the coefficients of very simple filters. However, for filters with good amplitude response it is not recommended as it relies on 'trial and error' shuffling of the pole and zero positions.

As with IIR filters there are several methods of calculating the coefficients of FIR filters. Three most popular methods are mentioned in this task, window, frequency sampling, and the Optimal (*Parks-McClellan algorithm*), [12]. The following provide a summary of some of the methods for calculating filter coefficients which are most widely used :

* impulse invariant	(IIR)	* bilinear transformation	(IIR)
* pole-zero placement	(IIR)	* window	(FIR)
* frequency sampling	(FIR)	* optimal	(FIR)

From these we choose the method that best suits a particular application. This choice will be influenced by the critical requirements in the specifications, (i.e in section 2.3)

3.4.3 Representation of filter structure

Realisation involves converting a given transfer function, $H(z)$, into a suitable filter structure. Block or flow diagrams are often used to depict these structures.

For IIR filters, three structures commonly used are the direct, cascade and parallel forms. The direct form is simply a straightforward representation of the IIR transfer function. In the cascade form, the transfer function of the IIR filter, equation 3.3.5, is factored and expressed as the product of second-order sections. In the parallel form, $H(z)$ is expanded using partial fractions as the sum of second-order sections. As an illustration and for comparison, figures 3.4, 3.5 and 3.6 shows the block diagrams for a fourth-order, (that is, $N = 4$) for IIR filter

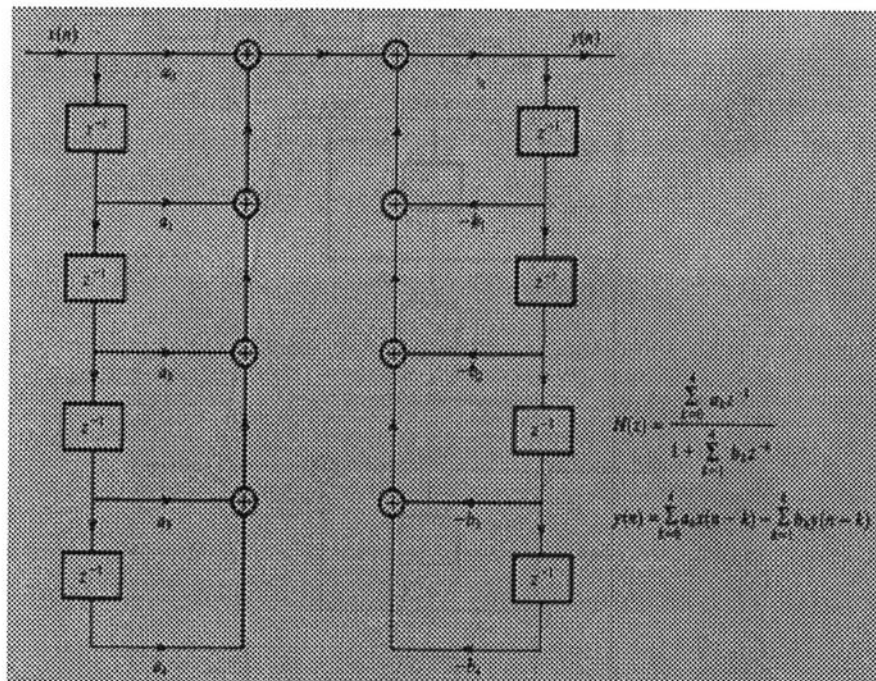


Fig 3.4
Direct realisation of a fourth-order IIR filter, [2].

represented in the direct, cascade and parallel structures respectively. The corresponding set of transfer functions and the difference equations describing these filter structures are also given in the figures.

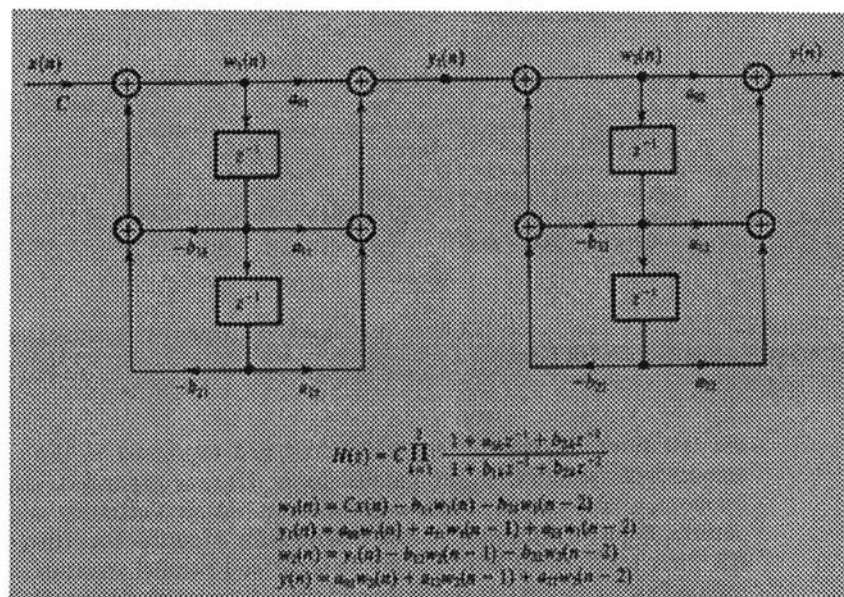


Fig 3.5
cascade realisation of a forth-order IIR filter, [2].

The parallel and cascade structures are the most widely used for IIR because they lead to simpler filtering algorithms and are far less sensitive to the effects of implementing the filter using a finite number of bits than the direct structure. The direct structure suffers severe coefficient sensitivity problems, especially for high order filters, and it has been suggested not to be use in this instance, [2].

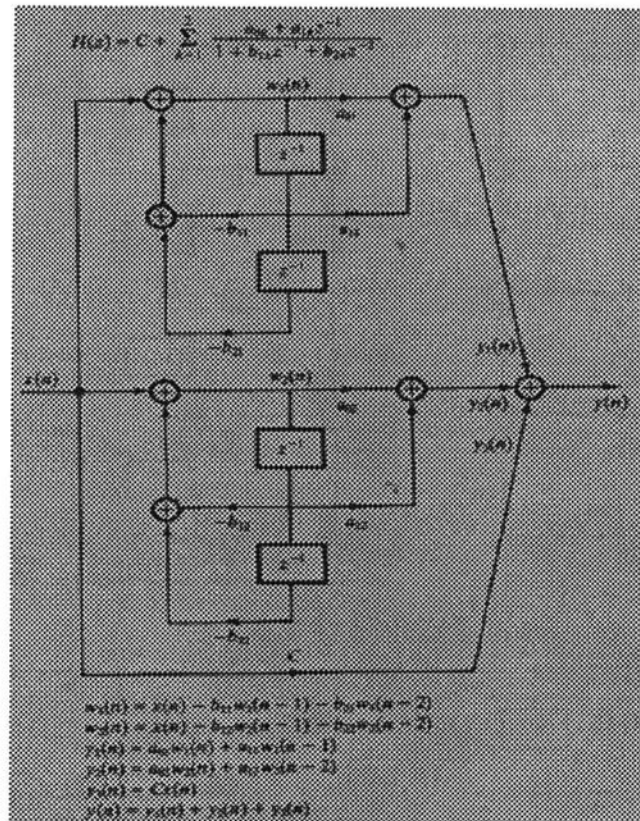


Fig 3.6
Parallel realisation of a fourth-order filter, [2].

However, the most widely used structures for the FIR, is the direct form, figure 3.7. This is because it is easy to implement. In this form, this structure is sometimes called tapped delay line.

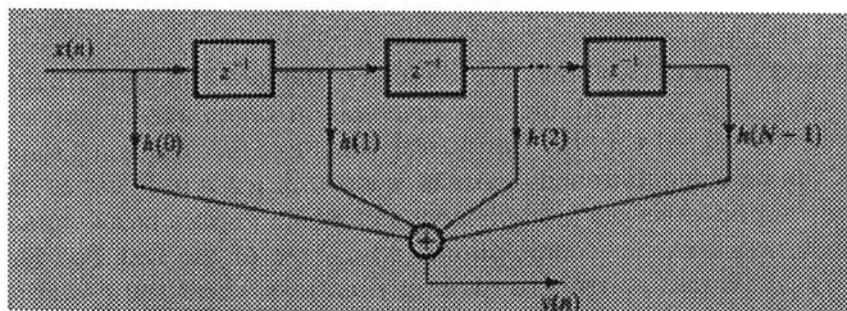


Fig 3.7
Realisation structure for FIR filter, [2].

Other structures used are the frequency sampling structure and the fast convolution technique, figure 3.8(a) and 3.8(b). Compared with the transversal structure, the frequency sampling structure can be computationally more efficient

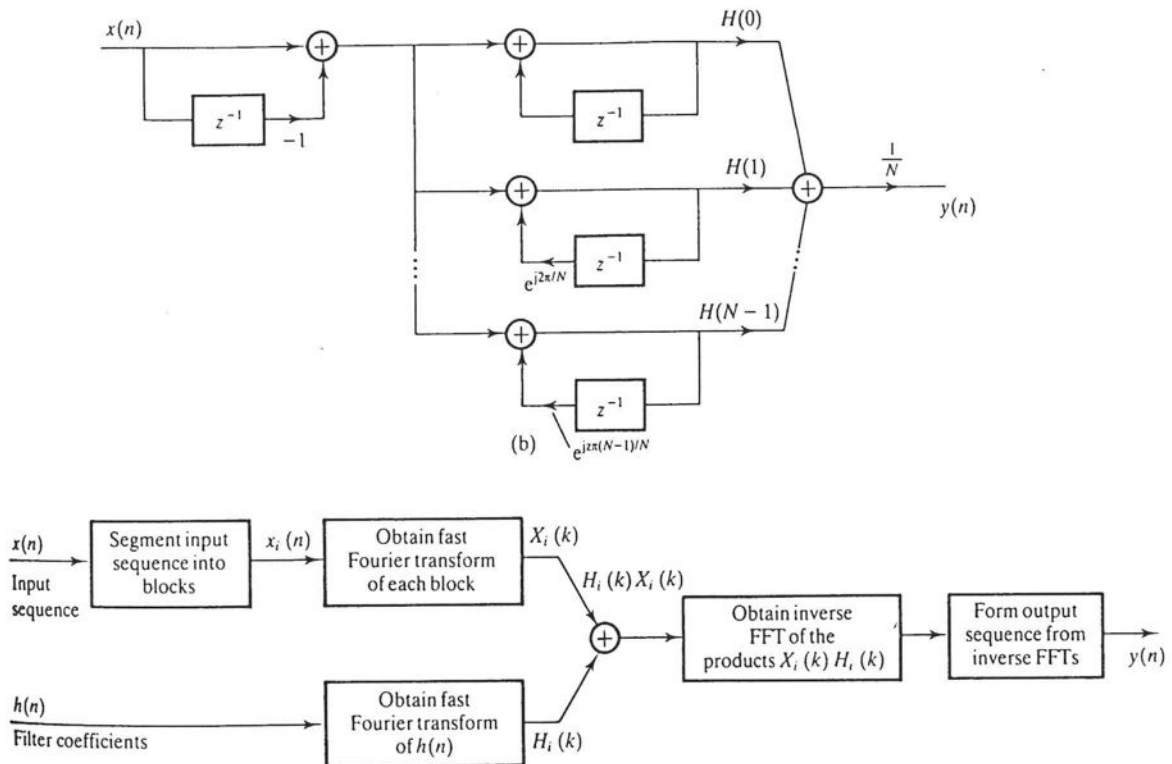


fig 3.8(a) and fig 3.8(b)
illustrating the frequency sampling and fast convolution technique

as it leads to fewer coefficients, but it may not be as simple to implement and would require more storage. The fast convolution uses the computational advantage of the *Fast Fourier Transform* (FFT) and is particularly attractive in situations where the power spectrum of the signal is also required.

In summary, the following are the commonly used realisation structures for FIR and IIR filters:

- | | | | |
|----------------------|--------|--------------------|--------|
| * Frequency sampling | (FIR); | * Fast convolution | (FIR); |
| * Direct form | (IIR); | * Cascade | (IIR); |
| * Parallel | (IIR); | | |

This task will only emphasise the transversal and the cascade structures wherever applicable.

3.4.4 Analysis of finite wordlength effects

The approximation and realisation steps assume infinite or very high precision. However, in actual implementation it is often necessary to represent the filter coefficients using limited number of bits, typically 8 to 16-bits, and the arithmetic operations indicated in the difference equation are performed using finite precision arithmetic.

The effects of using finite number of bits are to degrade the performance of the filter and in some cases to make it unusable. As a designer the analysis of these effects was put into consideration and a suitable wordlengths (that is, numbers of bits), was chosen for the filter coefficients, filter variables, that is the input and output samples, for the arithmetic operations within the filter.

The main sources of performance degradation in digital filters are as follows -

3.4.4.1 *Input/output quantization.*

In particular, the ADC noise due to quantizing of the input signal samples is significant.

3.4.4.2 *Coefficient quantization*

This leads to deviations in the frequency response of both FIR and IIR filters, and possibly to instability in IIR filters.

3.4.4.3 *Arithmetic roundoff errors*

The use of finite precision arithmetic to perform filtering operations yields results that require additional bits to represent. When these are quantized to the permissible wordlength, often by rounding, roundoff noise is the result. This can cause undesirable effects such as instability in an FIR filter.

3.4.4.4 *Overflow*

This occurs when the result of an addition exceeds permissible wordlength. It leads to wrong output samples and to possible instability in IIR filters.

3.5 INFINITE IMPULSE RESPONSE (IIR)

The design stages of an IIR digital filter are quite similar to that of FIR filters. The specification of the filter need be specified as with the case of an FIR filter, the coefficient calculation had been mentioned earlier. This could be based on bilinear transform or pole placement methods, depending on the specification requirements.

However, the realisation of an IIR filter differ slightly. Whilst direct realisation is inadvisable, parallel and cascade structures have gained wide acceptance both for industrial and research applications.

Analysis of error is similar to that of FIR filters and would not be discuss any further.

3.6 SAMPLING THEOREM

For any meaningful real-time digital filtering to be made possible, a signal source will be required as an input to the filter. Such signal are usually in analogue form. An equivalent

digital signal can be achieved by using an analogue to digital converter, (ADC). This ADC is based on sample hold method.

There are certain criteria that must be fulfilled for an analogue signal to be correctly represented in a digitised form, these criteria form a basis for the sampling theorem.

The **Sampling Theorem** - states that, for a given signal say $f(t)$, to reconstruct the signal correctly, the signal must be sampled at a rate that is at least twice the highest frequency component $f_{\max}$ i.e

$$f_s \geq f_{\max} \quad \dots\dots\dots (3.6)$$

where f_s , is the sampling frequency and $f_{\max}$ is the maximum frequency component.

However, if was decided to sample a time domain signal at time intervals of T , that is with a sampling frequency of $1/T$, then, as fig 3.9 reveals, another frequency component with the same set of samples as the signal of interest, may exist.

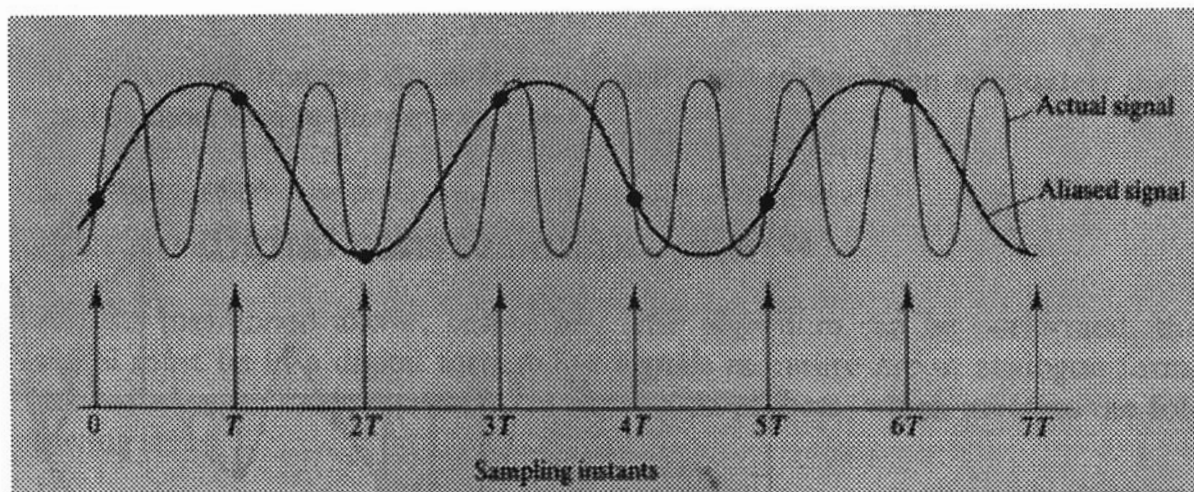


fig 3.9
An example of aliasing in the time domain, [14].

Thus the frequency component can be mistaken for the lower frequency components. This phenomenon is known as aliasing. This condition leads to irrecoverable or loss of information.

3.7 ADAPTIVE ALGORITHMS

Adaptive algorithms are used to adjust the coefficients of the digital filter, such that the error signal say e_k given by the difference between the measured signal and the estimate of the desired signal, is used to compute the weight of the adaptive filter. This will become clearer in the latter part of this task.

Quite a number of algorithms have found widespread applications, such as the least mean square (LMS), the recursive least square (RLS), and **Kalman** algorithms. In terms of

computation and storage requirements, the LMS algorithm is the most efficient. Furthermore, it does not suffer from the numeric instability problem inherent in the other two algorithms. As a consequence the LMS algorithm was preferred, although it has been established that the RLS algorithm has a superior convergence properties, [2].

CHAPTER 4 IMPLEMENTATION

4.1 INTRODUCTION

This chapter describes the actual work carried out aimed at meeting the objectives stated in sec. 2.3. Furthermore it demonstrates how fundamental theories have been implemented.

4.2 EQUIPMENT

The equipment supplied include the following

- * PCL-812 (12-bit) ADC, counter and digital input-output interface card accompanied with a driver routine
- * Thandar TG2002 20MHz sweep/function Generator
- * A 486-33MHz PC
- * Hitachi VC-6045 Digital Storage Oscilloscope
- * two coaxial connectors
- * Single ended ECG amplifier
- * X-Y plotter
- * Two breakout boxes
- * Electrolyte gel

4.3 SETTINGS

- * The input-output was set at address 0x210 hexadecimal.
- * The analogue to digital data registers use address BASE +4 and +5.

ADC	Low byte and data							
BASE +4	D7	D6	D5	D4	D3	D2	D1	D0
	A7	A6	A5	A4	A3	A2	A1	A0

ADC	High byte							
BASE +5	D7	D6	D5	D4	D3	D2	D1	D0
	0	0	0	DRDY	AD11	AD10	AD9	AD8

Legend:

AD11 to AD0 - Where AD0 is the least significant bit (LSB) and AD11 is the most significant bit (MSB) of the analogue-to-digital data.

DRDY - Data Ready signal.

0: Data ready,

1: Data not ready.

The register (BASE + 5) will set this bit to 1 and it is cleared when an analogue-to-digital (A/D) conversion is completed.

4.3.1 **Multiplexer scan #0 and #1**

The multiplexer scan channel 0 and 1 (#0 and #1) were chosen. Legend : CL3 to CL0 denotes the scan channel number.

4.3.2 **DAC Output**

The digital-to-analogue (D/A) registers are write registers, using address BASE, +6 and +7.

Data format:

BASE +6	D7	D6	D5	D4	D3	D2	D1	D0
---------	----	----	----	----	----	----	----	----

D/A #1 low	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
BASE +7	D7	D6	D5	D4	D3	D2	D1	D0
D/A #1 high	X	X	X	X	DA11	DA10	DA9	DA8

Legend: D/A11 to D/A0 describes 11 bits addressable register. The label DA0 is the least significant bit (LSB) and AD11 is the most significant bit (MSB) of the D/A data. Similarly D/A #2 was formatted as above. The D/A #1 was enabled.

4.3.3 PCL-812 Control Register

The PCL-812 control register is a write-only register using address BASE +11. This register provides information on the operating modes of the PCL-812.

Data format:

BASE +11	D7	D6	D5	D4	D3	D2	D1	D0
PCL-812CNTL	X	X	X	X	X	S2	S1	S0

A. under internal trigger condition

S2 S1 S0

0 0 0 : Disable software & pace trigger

0 0 1 : Enable software trigger only

0 1 0 : Enable pacer trigger using Dynamic Memory Access transfer only

1 1 0 : Pacer trigger using program transfer or interrupt transfer

1 1 0 : was selected, i.e enabling pacer trigger transfer program transfer or interrupt transfer.

B. External trigger source was not used.

4.3.4 Programmable Interval Timer/Counter Register

Four registers were dedicated for the Intel 8253 programmable timer/counter, located at address BASE 0, 1, 2, and 3. The 8253 timer/counter device consist of three independent 16-bit down counters. Each counter has a clock input, control gate and an output. It can be programmed to have a count from 2 up to 65535.

The maximum clock input frequency is 5MHz for version 5 of the 8253. The PCL-812 provides a 2MHz input frequency through an on-board crystal oscillator. Counters 1 and 2 are cascaded and operated in a fixed divide configuration. The counter 2 input is connected to the 2MHz frequency and the output of counter 2 is connected to the input of counter 1. The output of counter 1 is internally configured to provide a trigger pulse to the A/D converter.

The settings above are crucial, if the interface card was to function as intended. Furthermore this will enhance user friendly software, as it would be seen in the latter part of this project. A PCL-812 driver routine was provided on board. This routine was written in assembly language enhancing input and output of data with minimal overhead.

4.4 INITIAL WORK

Initial work began by trying to estimate the number of bits that would be required to represent the coefficient for the different filters to meet the required specification, (see Appendix B).

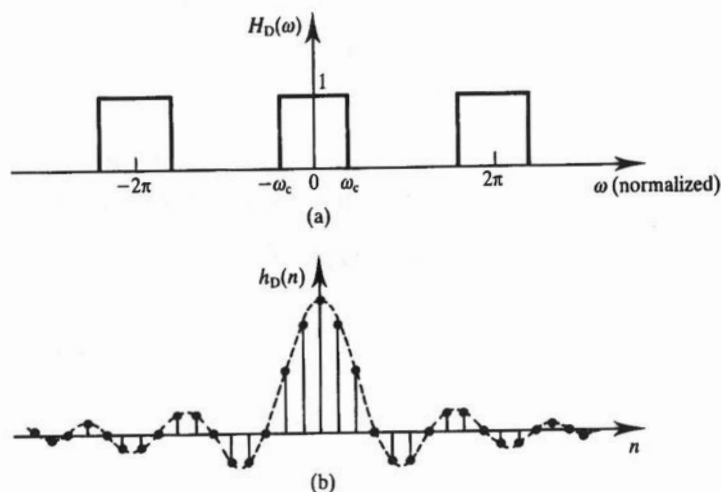
4.4.1 Lowpass filter design

The coefficient for this filter was obtained using a window method. This method makes use of the fact that the frequency response of a filter $H_D(\omega)$, and the corresponding impulse response, $h_D(n)$, are related by the Inverse Fourier transform:

$$h_D(n) = 1/2\pi \int_{-\pi}^{\pi} H_D(\omega) e^{j\omega n} d\omega \dots\dots\dots (4.4.1)$$

Where the subscript D , was used to distinguish between the ideal and practical impulse responses. If $H_D(\omega)$ is known then we can obtain $h_D(n)$ by evaluating the inverse Fourier transform of equation 4.4.1. In general, a lowpass, highpass, bandpass and bandstop filters can be designed from an ideal lowpass response, by using the appropriate relationship.

The figure 4.1 shows an ideal impulse response for a lowpass filter.



(Fig 4.1)
illustration on an ideal impulse response filter,[2]

It was noted that

$$h_D(n) = h_D(-n) \dots\dots\dots 4.4.2$$

so that the filter will have a linear phase response. The consequence of this is that the input and output signal will be in phase. However, there arises a problem with this approach. In as much as we move away from $n = 0$, $h_D(n)$ decreases and theoretically this carries on to $n = \pm \infty$. As a result the filter is no longer an FIR filter.

A probable solution is to truncate the ideal impulse response by setting $h_D(n) = 0$ for n greater than some value say N . However, this approach introduces an undesirable ripples and overshoots - called the **Gibb's** phenomenon. Notably this filter spectrum approaches the ideal impulse response by increasing the filter coefficient. Direct truncation of $h_D(n)$ as described above is equivalent to multiplying the ideal impulse response by a rectangular window of the form

$$W(n) = \begin{cases} 1 & n = 0, 1, \dots, M-1 \\ 0 & \text{elsewhere} \end{cases} \quad (4.4.3)$$

In the frequency domain this is equivalent to convolving $H_D(\omega)$ and $W(\omega)$, where $W(\omega)$ is the Fourier transform of $W(n)$. As $W(\omega)$ has the classic shape of $\sin(x)/x$, truncation of $h_D(n)$ leads to the overshoot and ripples in the frequency response as expected.

A Kaiser window was chosen out of the many windows available. This was because it satisfies the required specifications. Although the Blackman's window, also meet the required specifications, see sec 2.3. But the Kaiser window have the added advantage such that its characteristics, the transition width and stopband attenuation are tradable. This is not true with a Blackman, rectangular, Hamming windows etc. The Kaiser window incorporate a ripple control parameter, β , which allows the designer to trade off the transition width against the ripple. The definition of the Kaiser window is given by

$$W(n) = \begin{cases} I_0\{\beta[1-(2n/N-1)^2]^{1/2}\}I_0(\beta) & -(N-1)/2 \leq n \leq (N-1)/2 \\ 0 & \text{elsewhere} \end{cases} \quad (4.4.4)$$

where $I_0(x)$ is the zero-order modified Bessel function of the first kind. $I_0(x)$ is normally evaluated using the following power series expansion.

$$I_0(x) = 1 + \sum_{k=1}^L [(x/2)^k / k!]^2 \quad (4.4.5)$$

where typically $L < 25$, [13].

From the specification the stopband deviation is required to be greater than 60dB

$$\begin{aligned} & -20\log_{10}(\delta_s) \geq 60\text{dB} \\ \Rightarrow & \log_{10}(\delta_s) \geq -3 \\ \text{or} & \delta_s \geq 10^{-3} \end{aligned} \quad (4.4.6)$$

The number of the filter's coefficient would be given by

$$N \geq \frac{A - 7.95}{14.36\Delta F} \quad (4.4.7)$$

Where A is the stopband attenuation in decibels (dB), ΔF is the normalised sampling frequency and N is the filter length.

$$N \geq \frac{60 - 7.95}{14.36(15/250)} \dots\dots\dots (4.4.8)$$

giving
 $\therefore$

$$N \geq 60.4$$

$$N = 61$$

The ripple parameter would be given by

$$\beta = 0.1102(A - 8.7) \text{ if } A \geq 50\text{dB} \dots\dots\dots (4.4.9)$$

$$\beta = 0.1102(60 - 8.7)$$

$$= 5.653$$

A program "Window.c" was written to compute this coefficient, that was saved in file called "winlowps.dat". However, to allow for smearing effects of the window functions in computing the ideal impulse response, cut-off frequencies of $f_{c1} - \Delta f/2$ and $f_{c2} + \Delta f/2$ were used, i.e. f_{c1} and f_{c2} are 22.5Hz and 27.5Hz respectively.

For a Blackman window, an estimate of the number of filter coefficients was obtained as

$$N = 5.5/\Delta f$$

$$N = 91$$

This is quite large, therefore unacceptable.

4.4.2 Bandpass filter design

To obtain the coefficient for this filter, the bilinear Z transform (BZT) was used.

The transition width of the filter was given as 5Hz, therefore giving a cut-off frequency at 15Hz. However, there are different ways of achieving the design of this filter, but the method adopted is one that employed a suitable normalised lowpass analogue filter, with transfer function $H(s)$. This filter was then transformed to the desired digital bandpass filter by employing the appropriate transformation of an analogue to digital then followed by the z-transformation. This was achieved by defining $H(s)$ as follows -

$$H(s) = \frac{1}{S + 1} \dots\dots\dots (4.4.10)$$

It must be noted that $H(s)$ is a polynomial of degree n , where n is an integer to be determined.

The relationship between the filter's order and its characteristics would be given by

$$N \geq \{\log_{10}[(1/\delta_s) - 1]\} / 2\log_{10}(\omega_s'/\omega_p) \dots\dots\dots (4.4.11)$$

Where δ_s is the stopband deviation, ω_s and ω_p are the stopband and passband frequency respectively.

$$-20\log_{10}(\delta_s) \geq 15\text{dB}$$

$$\begin{aligned}
 \omega_1 &= 2\pi \times 10 \\
 \omega_2 &= 2\pi \times 15 \\
 \therefore N &\geq \log_{10} \left[\frac{5.62341 - 1}{2\log_{10} \left[\frac{2\pi \times 15}{2\pi \times 10} \right]} \right] = 1.888.. \\
 N &= 2
 \end{aligned}$$

i.e the order of the filter is 2.

This was then followed by determining the prewarped critical frequencies of the digital filter. In this case we have the lower and upper band edge frequencies, ω_1' and ω_2' .

$$\omega_1' = \tan(2\pi \times f_1/f_s) \quad (4.4.12)$$

$$\omega_2' = \tan(2\pi \times f_2/f_s) \quad (4.4.13)$$

Then s was replaced in the transfer function, H(s), using the transformation below

$$s = \frac{s^2 + \omega^2}{Ws} \quad (4.4.14)$$

i.e. lowpass to bandpass. Where $\omega_0^2 = \omega_1' \omega_2'$ and $W = \omega_2' - \omega_1'$.

By applying the BZT to the new $H'(s)$:

$$s = \frac{z-1}{z+1} \quad (4.4.15)$$

the z-transform can be computed.

Although it must be stressed that employing the butterworth characteristics will preserve the phase response of the filter but the elliptic characteristics produces a more efficient filter.

Using a first order normalised analogue lowpass filter $H_i(s)$, where (i) is an integer.

$$\begin{aligned}
 \therefore H(s) &= H_1(s).H_2(s) \\
 &= \frac{1}{s_1 + 1} \cdot \frac{1}{s_2 + 1} \quad (4.4.16)
 \end{aligned}$$

The prewarped passband edge frequencies are

$$\begin{aligned}
 \omega_1' &= \tan(\omega_1 T)/2 \\
 &= \tan(2\pi \times 10) / (2 \times 250) \\
 &= 0.126329 \\
 \omega_2' &= \tan(\omega_2 T)/2
 \end{aligned}$$

$$\begin{aligned}
 &= \frac{\tan(2\pi \times 15)}{2 \times 250} \\
 &= 0.190760
 \end{aligned}$$

$$\omega_0^2 = 0.024098617 \text{ and } W = 0.064430823$$

Using the lowpass-to-bandpass transformation equation (4.4.14), we have

$$\begin{aligned}
 H'(s) &= H(s) \Big|_{s = \frac{s^2 + \omega_0^2}{Ws}} \dots\dots\dots (4.4.17) \\
 &= \frac{1}{s^2 + \omega_0^2 / Ws + 1}
 \end{aligned}$$

$$H'(s) = \frac{Ws}{s^2 + Ws + \omega_0^2} \dots\dots\dots (4.4.18)$$

Applying the BZT to the analogue bandpass filter we have

$$\begin{aligned}
 H(z) &= H'(s) \Big|_{s = \frac{z-1}{z+1}} \dots\dots\dots (4.4.19) \\
 &= \frac{W(z-1)/(z+1)}{[(z-1)/(z+1)]^2 + W(z-1)/(z+1) + \omega_0^2}
 \end{aligned}$$

$$= \frac{W(z^2 - 1)/(1 + W + \omega_0^2)}{z^2 + [2(\omega_0^2 - 1)/(1 + W + \omega_0^2)]z + (1 - W + \omega_0^2)/(1 + W + \omega_0^2)} \dots\dots\dots (4.4.20)$$

substituting known values of ω_0 and W and simplifying we have

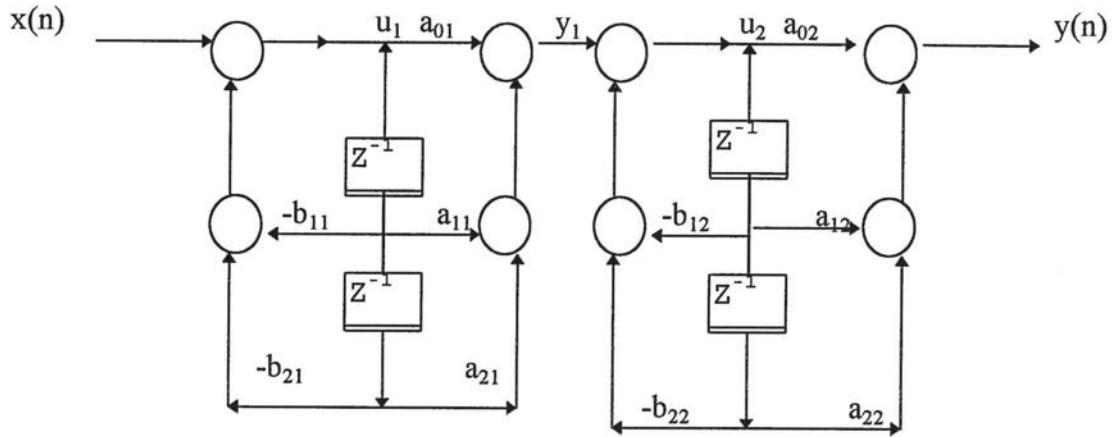
$$\begin{aligned}
 H(z) &= H_1(z).H_2(z) \\
 \Rightarrow H_1(z) &= \frac{0.11310(z^2 - 1)}{1.11468z^2 - 1.99684z + 0.99955} \quad (5 \text{ s.f.f}) \dots\dots\dots (4.4.21)
 \end{aligned}$$

similar expression holds for $H_2(z)$. As a consequence, the overall transfer function $H(z)$, is a fourth order, and the difference equation cannot be implemented by direct realisation, as this may cause overflow at the summing junction, and consequently resulting to an unstable filter. As a result $H(z)$ was broken down into smaller sections of second order canonical blocks to facilitate the computation of the difference equation. These blocks were then connected in cascade.

The figure 4.2 illustrated how the computation was implemented.

From the canonical diagram, the method of signal flow was adopted. This involves multiplying the input at one end of the arrow with a scalar quantity on the arrow and the resulting solution represents the output in the direction of the arrow.

From the canonical diagram, the method of signal flow was adopted. This involves multiplying the input at one end of the arrow with a scalar quantity on the arrow and the resulting solution represents the output in the direction of the arrow.



(fig 4.2)
Cascade structure for an IIR bandpass filter

$$H(z) = \frac{Y(z)}{X(z)} = H_1(z) \cdot H_2(z)$$

$$H(z) = \frac{0.11310(z^2 - 1)}{1.11468z^2 - 1.99684z + 0.99955} \cdot \frac{0.11310(z^2 - 1)}{1.11468z^2 - 1.99684z + 0.99955}$$

$$a_{01} = a_{02} = 0.11310; \quad a_{11} = a_{12} = 0; \quad a_{21} = a_{22} = -0.11310$$

$$b_{11} = b_{12} = -1.99684; \quad b_{21} = b_{22} = 0.99955$$

substituting known values into the difference equation, the following was obtained -

$$u_1(n) = x(n) - b_{11}u_1(n-1) - b_{21}u_1(n-2)$$

$$y_1(n) = a_{01}u_1(n) + a_{12}u_1(n-1) + a_{21}u_1(n-2)$$

$$u_2(n) = y_1(n) - b_{12}u_2(n-1) - b_{22}u_2(n-2)$$

$$y(n) = a_{02}u_2(n) + a_{12}u_2(n-1) + a_{22}u_2(n-2)$$

This difference equation so derived lend itself to computation. Finally the notch filter was designed using the following procedure.

4.4.3 Notch filter design

As a matter of convenience the pole-zero design methodology provides a solution that can be easily implemented, without any significant loss of accuracy in the filter design. As a result this method was adopted.

In order to reject the component at 50Hz, this requires that a pair of complex zeros are placed at points on the unit circle corresponding to 50Hz. The equivalent angles are

$$\begin{aligned}\theta &= 360^\circ \times \frac{f}{f_s} \dots\dots\dots (4.4.22) \\ &= 360^\circ \times \frac{50}{250} \\ &= \pm 72^\circ\end{aligned}$$

To achieve a sharp notch filter and improved amplitude response on either side of the notch frequency, a pair of complex conjugate poles are placed at a radius $r < 1$. The width of the notch is determined by the location of the poles. And the relationship between the bandwidth and the radius would be given by

$$r \approx 1 - (bw/F_s)\pi \dots\dots\dots (4.4.23)$$

With a bandwidth of 5Hz, the poles should be placed at

$$\begin{aligned}1 - (5/250)\pi \\ = 0.937 \text{ unit}\end{aligned}$$

Therefore the poles must be located at

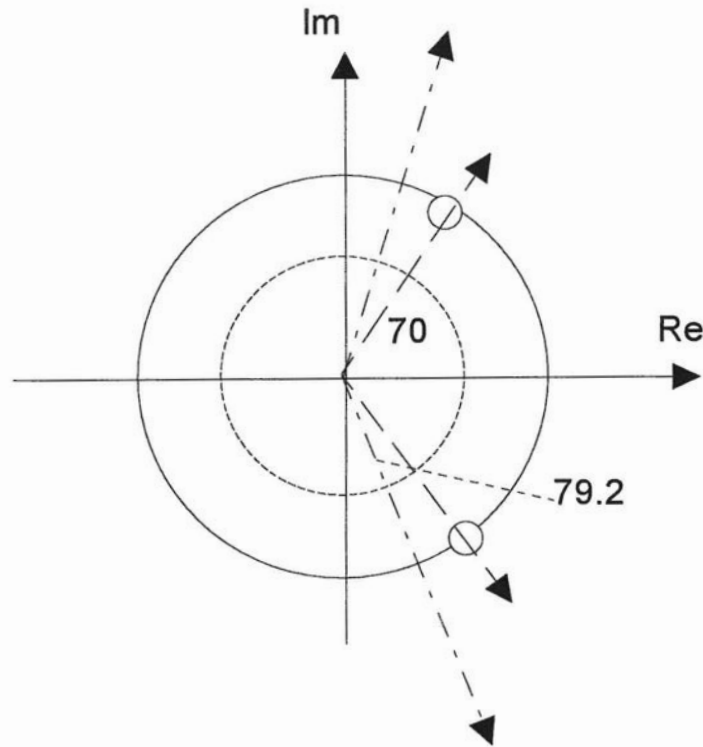
$$\pm = 360^\circ \times \frac{5}{250}$$

$\Rightarrow$ 5Hz corresponds to $= 7.2^\circ$.

Therefore the poles need be located at 79.2° . The corresponding transfer function say $H(z)$ would be of the form.

$$H(z) = \frac{(z - e^{j\theta})(z - e^{-j\theta})}{(z - re^{j\theta})(z - re^{-j\theta})} \dots\dots\dots (4.4.24)$$

The pole-zero diagram will provide an insight into this problem.



(Fig 4.3)
Notch filter design using pole placement method

As a consequence, from the diagram the transfer function can be written by inspection, thus

$$H(z) = \frac{(z - e^{j72^\circ})(z - e^{-j72^\circ})}{(z - re^{j79.2^\circ})(z - re^{-j79.2^\circ})}$$

$$re^{j79.2} = [\cos 79.2 + j\sin 79.2]r$$

$$re^{-j79.2} = [\cos 79.2 - j\sin 79.2]r$$

where

$$r = 0.937, z^2 - 2r\cos 79.2^\circ z + r^2$$

$$\Rightarrow H(z) = \frac{z^2 - 0.618034z + 1}{z^2 - 0.35122z + 0.87828} \quad \dots\dots\dots (4.4.25)$$

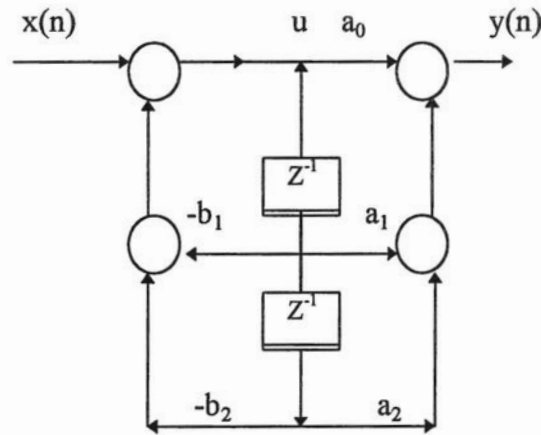
$$a_0 = a_2 = 1;$$

$$a_1 = -0.618034$$

$$b_1 = -0.35122;$$

$$b_2 = 0.87828$$

From the canonical structure in fig 4.4, it illustrated how the filter was finally implemented and how the required difference equation was computed.



(fig 4.4)
illustration of the canonical structure for the notch filter design

$$u(n) = x(n) - b_1 u(n-1) - b_2 u(n-2)$$

$$y(n) = a_0 u(n) + a_1 u(n-1) + a_2 u(n-2)$$

4.4.4 Stability

Having establish these difference equations for both the bandpass and the notch filters, there is a tendency for the system of equations to exhibit instability and consequently need to be investigated. From the knowledge of the transfer function, $H(z)$, for a given system of equation, the concept of the stability triangle is applicable. Consider the general equation for a second order z -transfer function, can be written thus,

$$H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}} \quad \dots\dots\dots (4.4.26)$$

$$y(n) = \sum_{k=0}^2 a_k x(n-k) - \sum_{k=1}^2 b_k y(n-k) \quad \dots\dots\dots (4.4.27)$$

The poles (or root of the denominator) are located at

$$p_1 = 1/2[-b_1 + (b_1^2 - 4b_2)^{1/2}] \quad \dots\dots\dots (4.4.28)$$

$$p_2 = 1/2[-b_1 - (b_1^2 - 4b_2)^{1/2}] \quad \dots\dots\dots (4.4.29)$$

For each second order equation, three types of poles arise: Complex conjugate poles, poles that are real and unequal, real equal poles. Complex poles are the most common and this occur if

$$b_1^2 < 4b_2$$

For this case, the poles are each located at a radius, r , from the origin and at an angle θ given by :

$$p_1 = r < \theta$$

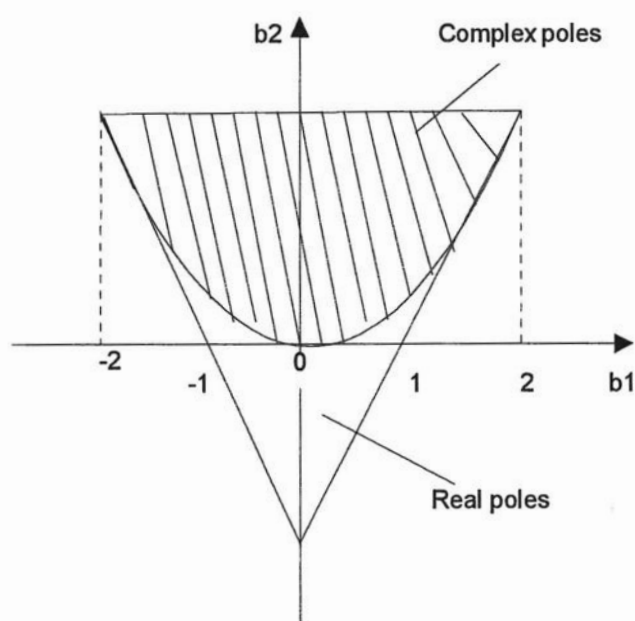
$$p_2 = r < -\theta$$

where

$$r = (b_2)^{1/2} \dots\dots\dots (4.4.30)$$

$$\theta = \cos^{-1}(-b_1/2r) \dots\dots\dots (4.4.31)$$

Small changes in the coefficients b_1 and b_2 ,



(Fig 4.5)
illustrates the stable region within the triangle

whatever their cause, will produce a corresponding change in both r and θ . However, for stability of the coefficients of the filter to be guaranteed, they must lie inside the stability triangle (fig 4.5), and bounded by the following criteria

$$0 \leq |b_2| < 1 \dots\dots\dots (4.4.32)$$

$$|b_1| \leq 1 + b_2 \dots\dots\dots (4.4.33)$$

Using this criteria, the transfer function derived earlier in equ (4.4.21) and (4.4.25) can now be examined for stability.

4.4.4.1 Stability of the bandpass filter

From equation (4.4.21) using use of only one of the building blocks, since $H_1(z)$ and $H_2(z)$ are identical, consideration will be limited to $H_1(z)$ only.

From the canonical structure diagram, this can be verified.

$$H_1(z) = \frac{0.11310(z^2 - 1)}{1.11468z^2 - 1.99684z + 0.99955}$$

$$b_{11} = -1.99684 \quad b_{21} = 0.99955$$

These coefficients satisfies the required stability criteria.

4.4.4.2 Stability of the notch filter

Similarly from equation (4.4.25), the transfer function was found to be

$$H(z) = \frac{z^2 - 0.618034z + 1}{z^2 - 0.35122z + 0.87828}$$

$$b_1 = -0.35122 \quad b_2 = 0.87823$$

These coefficients conforms to the stability criteria.

When the above design was completed, several tests were carried out to confirm that they function as intended. However, the result will discuss in section 6.

4.5 ADAPTIVE FILTER THEORY

As mentioned earlier, an adaptive filter is essentially a digital filter with adjustable coefficients, and an adaptive algorithm which is used to adjust or modify the coefficients of the filter. As a result the filter adapts, automatically to changes in its input signals, hence, term as adaptive.

Adaptive algorithm has many wide applications, with varied configuration which is application dependent. However, this task focuses only on adaptive noise cancelling principles.

4.5.1 Configuration of the system

The fig 4.5, illustrates how the system will be modelled. This configuration will form the basis for developing a mathematical solution for the problem in question, (i.e. noise canceling technique).

From this configuration, it is evident that if $y(t)$ is taken as the output signal, this output is identically the output of the digital filter. However on the contrary if the $e(t)$ is taken as the output signal, it constitute the error signal which is subsequently feedback to the adaptive filter, providing it with a means for updating the filter coefficient or weights.

The following terms were defined - the two input signals, the heart pulse, $x(t)$, which consist of the desired signal, s_k , and some noise, the mains frequency, say n_k , are assumed to be uncorrelated with each other; and the reference signal, r_k , produces an estimate of the noise

or unwanted signal in the measured signal x_k . This is assumed to be correlated in some way with n_k . r_k is being processed by the digital filter to produce an estimated, $\hat{n}_k$, of n_k .

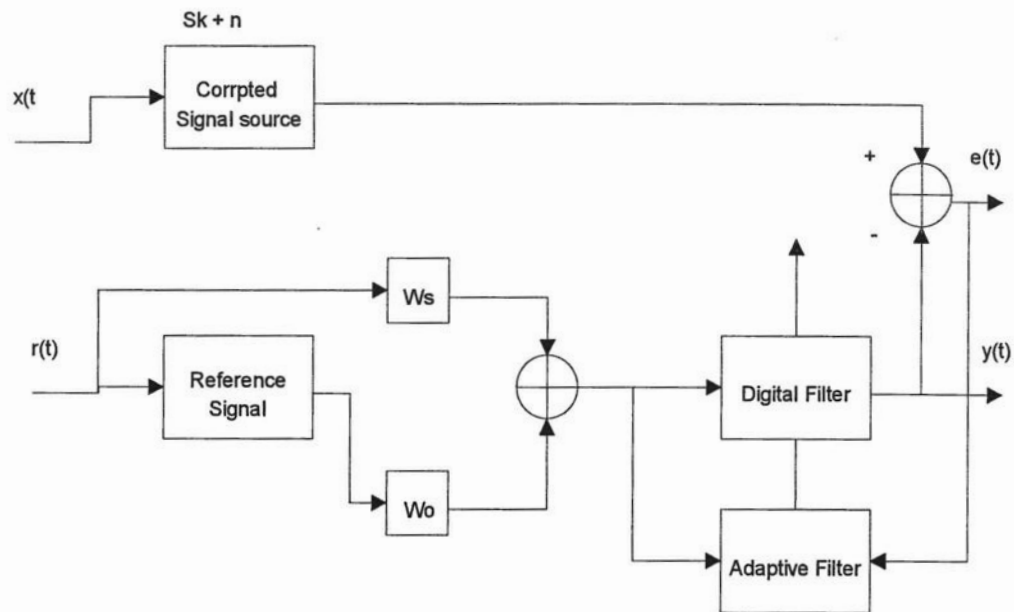


Fig 4.6
Adaptive noise cancelling configuration

Now
$$x_k = s_k + n_k \quad \dots\dots\dots (4.5.1)$$

However, the error signal e_k , is to be minimized according to some criterion, for example in the least squares sense. Quite a number of algorithms have found widespread applications, such as the least mean square (LMS), the recursive least square (RLS), and Kalman algorithms. In terms of computation and storage requirements, the LMS algorithm is the most efficient. Furthermore, it does not suffer from numeric instability problem inherent in the other two algorithms. As consequence the LMS algorithm was chosen as the first choice. Although it has been established that the RLS algorithm has a superior convergence properties.

If r_k is a measure of n_k , then it can be argued that

$$r_k = \hat{n}_k \quad \dots\dots\dots (4.5.2)$$

i.e an estimate of the contaminating signal. Therefore, the estimate of the desired signal, $\hat{s}_k$, would be given by

$$\begin{aligned} \hat{s}_k &= x_k - r_k \\ \hat{s}_k &= s_k + n_k - \hat{n}_k \quad \dots\dots\dots (4.5.3) \end{aligned}$$

The main objective in a noise canceling adaptive filter is to produce an optimum estimate of the noise in the corrupted signal and hence an optimum estimate of the desired signal. This can be achieved by using $\hat{s}_k$, in a feedback arrangement to adjust the digital filter coefficients, via a suitable adaptive algorithm, to minimize the noise in $\hat{s}_k$.

The output signal serves two purposes:

- (a) as an estimate of the desired signal and
- (b) as an error signal which is used to adjust the filter coefficients

If the filter structure was assumed to be FIR with N coefficients, the error e_k between the **Wiener filter** output and the primary signal, x_k is given by

$$\begin{aligned} e_k &= x_k - \hat{n}_k \\ &= x_k - W^T r_k \\ &= x_k - \sum_{i=0}^{N-1} W(i) r_{k-i} \end{aligned} \quad \dots\dots\dots (4.5.4)$$

Where r_k and W are the output signal vector and weight vector respectively, and are given by

$$r = \begin{bmatrix} r_k \\ r_{k-1} \\ | \\ r_{k-(N-1)} \end{bmatrix} \quad W = \begin{bmatrix} W(0) \\ W(1) \\ | \\ W(N-1) \end{bmatrix}$$

The square of the error would be given by

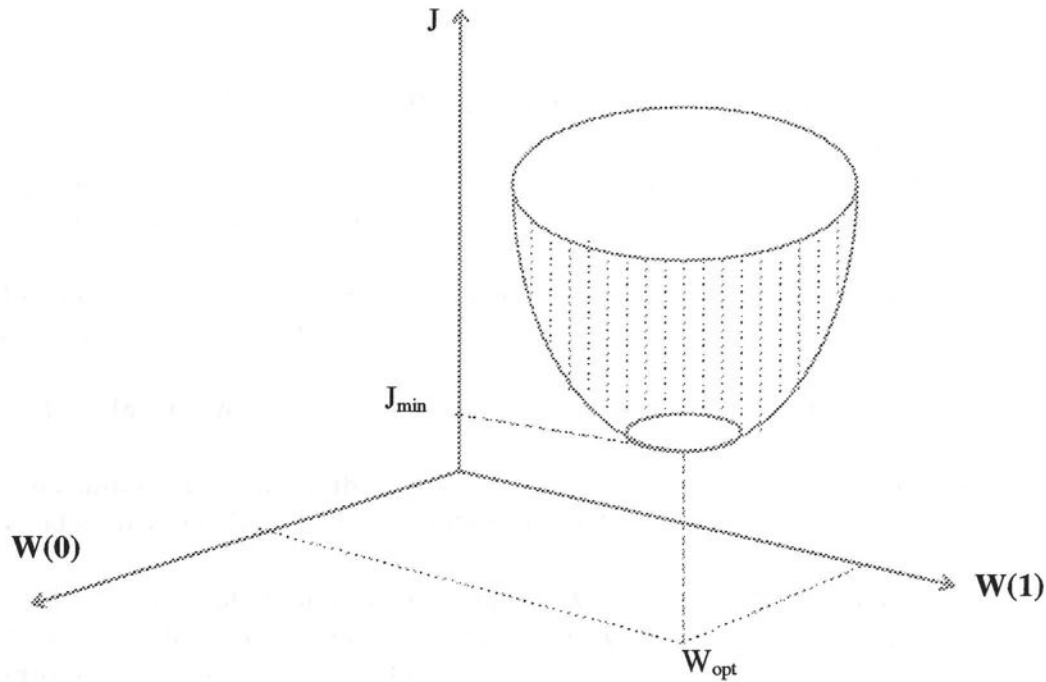
$$\begin{aligned} e_k^2 &= (x_k - W^T r_k)^2 \\ &= x_k^2 - 2x_k r_k^T W + W^T r_k r_k^T W \end{aligned} \quad \dots\dots\dots (4.5.5)$$

The mean square error (MSE), J , can be obtained by taking expectations of both sides of the equations

$$\begin{aligned} E[e_k^2] &= E[x_k^2] - 2E[x_k r_k^T W] + E[W^T r_k r_k^T W] \\ J &= \sigma^2 + 2P^T W + W^T R W \end{aligned} \quad \dots\dots\dots (4.5.6)$$

where $E[.]$ symbolises expectation, $\sigma^2 = E[x_k^2]$, is the variance of x_k , $P = E[x_k r_k^T]$ in the N length cross-correlation vector and $R = [x_k x_k^T]$ is the NxN autocorrelation matrix. A plot of the MSE against the filter coefficients, W , is a bowl shaped with a unique bottom, (see fig 4.7). This figure is known as the performance surface and is non negative.

The gradient, ∇ of the MSE is obtained by differentiating equation (4.5.6), with respect to the weight vector W , and setting the result to zero, [16].



(Fig 4.7)
showing the error performance surface

$$MSE = J = \sigma^2 + 2p^T W + W^T R W$$

$$\nabla = \frac{dJ}{dW} = \frac{d\sigma^2}{dW} + 2 \frac{d(p^T W)}{dW} + \frac{d(W^T R W)}{dW}$$

$$\frac{d\sigma^2}{dW} = 0$$

$$\frac{d(2p^T W)}{dW} = -2p$$

$$\frac{d(W^T R W)}{dW} = 2RW$$

Using these results, and setting $\nabla = 0$, equation (4.5.6) becomes

$$\nabla = \frac{dJ}{dW} = -2P + 2RW = 0 \quad \dots\dots\dots (4.5.7)$$

$W(i)$, where $(i = 0, 1, 2, 3 \dots\dots\dots N-1)$,

corresponds to a point on the surface. At the minimum point of the surface, the gradient is zero and the filter weight vector has an optimum value, W_{opt} . Therefore

$$W_{opt} = R^{-1}P \quad \text{.....} \quad (4.5.8)$$

equation (4.5.8), is known as the **Wiener-Hopf** equation or solution. The task in adaptive filtering is to adjust a filter weights, $W(0)$, $W(1)$, using a suitable algorithm, to find the optimum point on the performance surface. However, the Wiener filter has a limited practical usefulness, due to the following:

- (a) **Correlation** - It requires the autocorrelation matrix, **R**, and the cross-correlation vector, **P**, both of which are not known a priori.
- (b) **Matrix Inversion** - it involves matrix inversion which is time consuming, and
- (c) **Nonstationary** signal - if the signals are nonstationary, then both **R** and **P** will change with time and so W_{opt} will have to be computed repeatedly.

For real-time application, a way of obtaining W_{opt} on a sample-by-sample basis is required. Adaptive algorithms have been used to achieve this without having to compute **R** and **P** explicitly or performing matrix inversion, [17].

4.5.2 The basis for LMS adaptive algorithm

The LMS adaptive algorithm was one of the most successful. The computation of W_{opt} in one go as suggested by equation (4.5.8), was obviated. Instead, in the LMS the coefficients are adjusted from sample to sample in such a way as to minimize the MSE. This amounts to descending along the surface of fig 4.7, towards its bottom.

The LMS is based on the **steepest descent** algorithm where the weight vector is updated from sample to sample as follows:

$$W_{k+1} = W_k - \mu \nabla_k \quad \text{.....} \quad (4.5.9)$$

Where W_k and ∇ are the weight and the true gradient vectors, respectively, at the k th sampling instant and μ controls the stability and rate of convergence.

The steepest descent algorithm still requires the knowledge of **R** and **P**, since ∇_k is obtained by evaluating equation 4.5.8. The LMS is a practical method of obtaining estimates of the filter weights W_k in real-time without a matrix inversion in equation 4.5.9 or the direct computation of the autocorrelation and cross-correlation. The Widrow-Hopf LMS algorithm for updating the weights from sample to sample is given by

$$W_{k+1} = W_k - 2\mu e_k x_k \quad \text{.....} \quad (4.5.10)$$

where

$$e_k = x_k - W_k^T r_k \quad (\text{equ 4.5.4})$$

It is now very obvious that the LMS algorithm above does not require a prior knowledge of the signal statistics (ie the correlations $\mathbf{R}$ and $\mathbf{P}$), but instead uses their instantaneous estimates. The weights obtained by the LMS algorithm are only estimates, but these estimates improve gradually with time as the weights are adjusted and the filter learns the characteristics of the signals. Eventually the weights converge. The condition for convergence is:

$$0 < \mu < \frac{1}{\lambda_{\max}} \dots\dots\dots (4.5.11)$$

where $\lambda_{\max}$ is the maximum eigenvalue of the input data covariance matrix, [2].

CHAPTER 5 DIFFERENTIAL EQUATION DEVELOPMENT

From figure 4.6, section 4, the development proceeded by considering $x(t)$, the corrupted signal as follows -

$$\text{Given that } x(t) = s(t) + n(t) \dots\dots\dots (5.1)$$

$$r(t) = r_s(t) = A \sin 2\pi f_0 t \dots\dots\dots (5.2)$$

$$r(t) = A \cos 2\pi f_0 t \dots\dots\dots (5.3)$$

According to the LMS adaptive algorithm, weights iteration formula is given by

$$W_s(k+1) = W_s(k) + 2\mu e(k)r_s(k) \dots\dots\dots (5.4)$$

$$W_o(k+1) = W_o(k) + 2\mu e(k)r_o(k) \dots\dots\dots (5.5)$$

$$W_o(k+1) = 2\mu \sum_{n=0}^k e(n)r_o(n) \dots\dots\dots (5.6)$$

$$y(k) = y_s(k) + y_o(k) \dots\dots\dots (5.7)$$

$$2\mu A^2 \left\{ \sin(\omega_o k T_s) \sum_{n=0}^k e(n) \sin(\omega_o n T_s) + \cos(\omega_o k T_s) \sum_{n=0}^k e(n) \cos(\omega_o n T_s) \right\} \dots\dots\dots (5.8)$$

$$y(t) = 2\mu A^2 / T_s \left\{ \sin \omega_o t \int_0^t e(t) \sin \omega_o t dt + \cos \omega_o t \int_0^t e(t) \cos \omega_o t dt \right\} \dots\dots\dots (5.9)$$

putting $e(t) = x(t) - y(t)$ into (5.9), the following differential equation can be derived, (see appendix A) .

$$y''(t) + 2\mu A^2 / T_s y'(t) + \omega_o^2 y(t) = 2\mu A^2 / T_s x'(t) \dots\dots\dots (5.10)$$

The solution of the above equation is of the form

$$y(t) = C_1 e^{-\eta t} + C_2 e^{-\zeta t} \dots\dots\dots (5.11)$$

$$\text{where } \eta = \mu A^2 / T_s + \sqrt{(\mu A^2 / T_s^2)^2 - \omega_o^2} \dots\dots\dots (5.12)$$

C_1 and C_2 are constant. According to the equations (5.11) and (5.12)

$$\zeta = \eta = \mu A^2 / T_s + \sqrt{(\mu A^2 / T_s^2)^2 - \omega_o^2} \dots\dots\dots (5.13)$$

It is known that the system of equation will converge when $2\mu A^2/T_s \geq \omega_o^2$, [4]. It converges monotonically when $\eta = 2\mu A^2/T_s < \omega_o^2$ oscillating convergence. The greater step μ is, the faster the speed of convergence is. In order to reduce the weight noise during adaptive iteration, the value of μ can not be too great. Generally, we choose

$$2\mu A^2/T_s < \omega_o^2$$

and the system in oscillating convergence. The convergence time of the system is

$$\tau = T_s / 2\mu A^2$$

In order to get the frequency response of the filter, (5.10) is transformed using the Laplace, with the initial values

$$y'(0) = 0, y(0) = 0, x(0) = 0$$

$$\text{Then } S^2 Y(S) + 2\mu A^2/T_s S Y(S) + \omega_o^2 Y(S) \dots\dots\dots (5.14)$$

$$= 2\mu A^2/T_s S X(S) \dots\dots\dots (5.15)$$

$$\frac{Y(S)}{X(S)} = \frac{(2\mu A^2 / T_s)}{S^2 + (2\mu A^2 / T_s)S + \omega_o^2} \dots\dots\dots (5.16)$$

The stationary frequency response of the system is :

$$H_f(\omega) = \frac{1}{1 + \frac{((\omega_o - \omega)T_s)^2}{2\mu A^2\omega}} \dots\dots\dots (5.17)$$

The bandwidth of the adaptive filter is

$$B = 2\mu A^2/T_s H(z) \dots\dots\dots (5.18)$$

If the $e(t)$ is used as output the system is a notch filter. Its stationary frequency is :

$$|H_N(\omega)| = \frac{1}{1 + \frac{(2\mu A^2)^{1/2}}{((\omega_o - \omega)T_s)^2}} \dots\dots\dots (5.19)$$

The notch frequency is above adaptive notch filter, the frequency of the reference. Since the notch frequency (or the frequency of powerline interference) is variable, if the reference signal frequency in the notch filter was fixed, its notch bandwidth must be wide.

i. Initially, set each weight $W_k(i)$, $i=0, 1, \dots, N-1$, to an arbitrary fixed value such as 0. For each subsequent sampling instants, $k = 1, 2, \dots$ carry out steps ii. to iv.

ii. Compute filter output $\hat{n}_k = \sum_{i=0}^{N-1} W_k(i)x_{k-i}$

iii. Compute the error estimate $e_k = x_k - \hat{n}$

iv. Update the next filter weights $W_{k+1}(i) = W_k(i) + 2\mu e_k x_{k-i}$

6.1 INTRODUCTION

Previous section have examined filter design, adaptive and noise cancelling technique and have developed a solution to a general equation that describes the system under consideration. However, this section will investigate the behaviour of the system for different input signals and attempt to predict what values of μ and perhaps the characteristics of the input signal, that will cause the system to be unstable. If this value can be established, then the convergence factor μ , for the proposed adaptive filter can be estimated with the added advantage that number of iteration to converge to the true value will be reduced.

The findings from the simulation have been subdivided into two section, namely the coupled system and the decoupled system.

6.2 COUPLED SYSTEM

The simulation criteria for the coupled system and the decoupled one, are identical except that an initial conditions was imposed on the uncoupled system.

Considering equation 5.10,

$$\ddot{y}(t) + \frac{2\mu A^2}{T_s} \dot{y}(t) + \omega_o^2 y = \frac{2\mu A^2}{T_s} \dot{x}(t)$$

by putting

$$\frac{d}{dt} = q$$

then equation 6.2.1 can be rewritten thus

$$q^2 \dot{y} + \frac{2\mu A^2 q y}{T_s} + \omega_o^2 y = \frac{2\mu A^2 q x}{T_s} \dots\dots\dots (6.2.1)$$

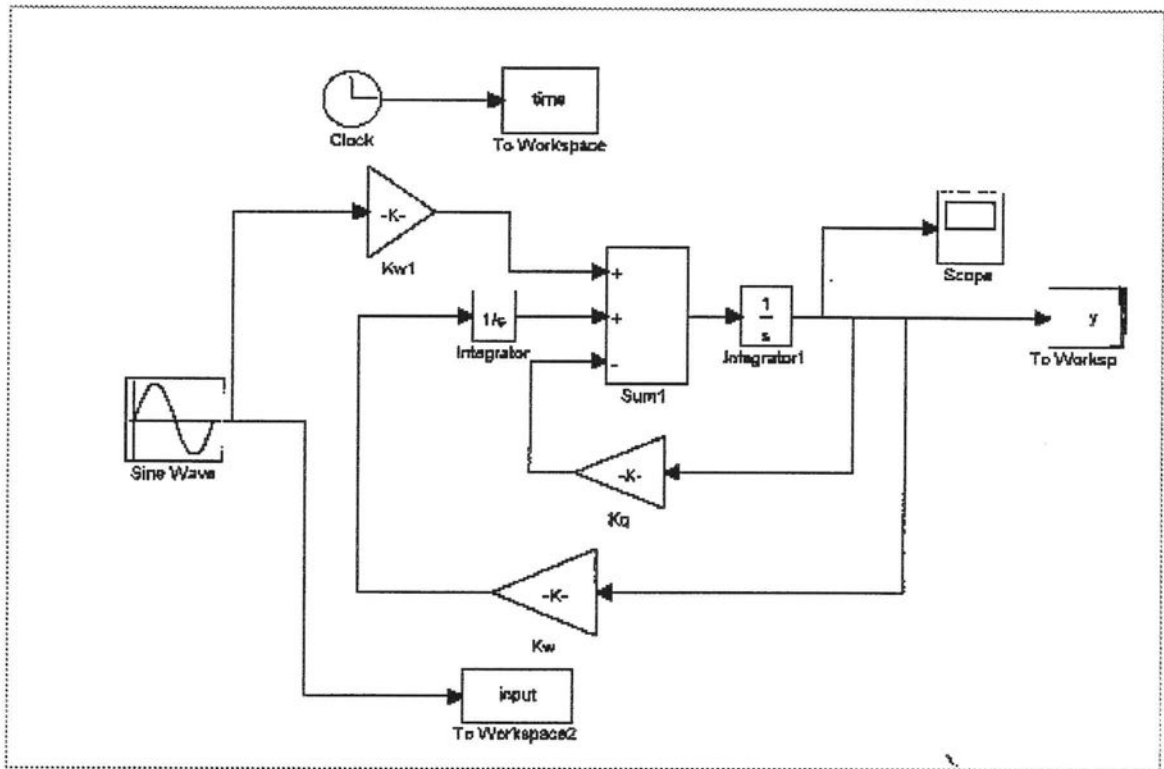
$$y = \frac{1}{q} \frac{2\mu A^2}{T_s} (x - y) - \frac{1}{q^2} \omega_o^2 y \dots\dots\dots (6.2.2)$$

To simulate this system, analogue equivalent was developed with appropriate integrators as shown in figure 6.1.

Work began by employing a sinusoidal function generator provided by the modelling package, Simulink, and the parameters of interest are the input frequency and the gain of the system.

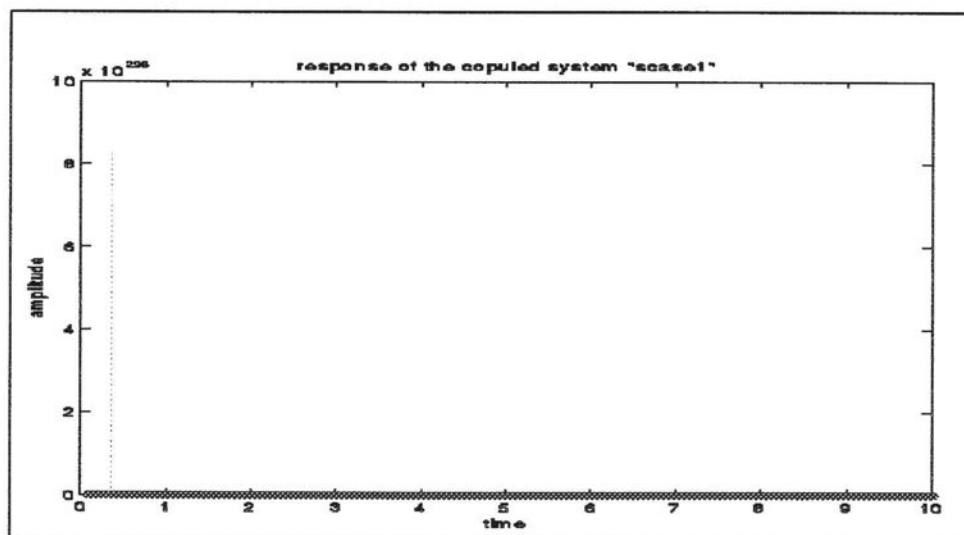
However, the gain imposed on the system subsequently depends on the parameter of the input signal such as its amplitude and the sampling frequency, (chapter 5). As a result, these quantities were employed to determine the gain on the system, (see Appendix D). First case considered was a very high gain, (see case1), and the response of the system is as shown in

figure 6.2. The figure does not reflect the nature of the input signal frequency at 18kHz and amplitude 10units.



(Fig 6.1)
Analogue simulation model

The output of the system was found to be extremely high and consequently unstable. Many other simulation with different parameters (cases), was considered.

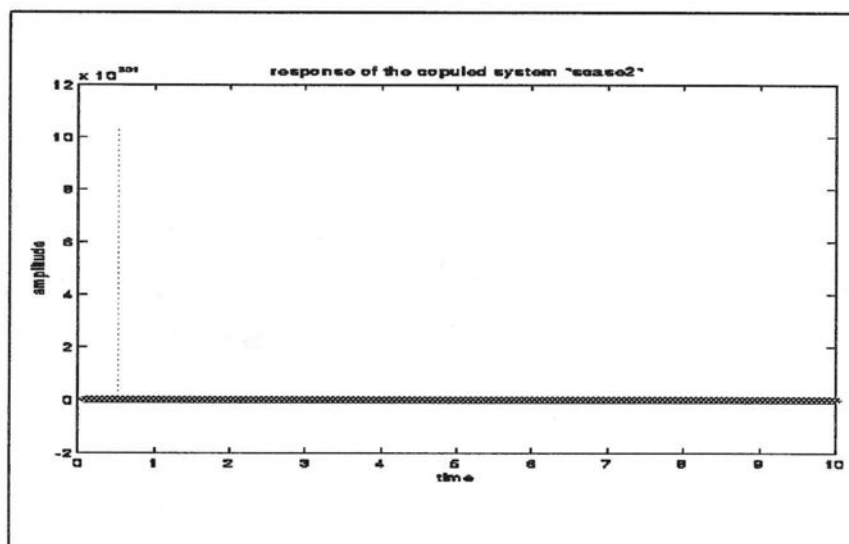


(Fig 6.2)
Output simulation of the system at 18kHz, (case1)

It was noticed from the diagram, that the output of the system are identically unstable. However, the frequency of the input signal was reduced from 18kHz to 250Hz and the gain was made as low as 31.4units and system was simulated similarly as for the case of the 18kHz signal but the system was found to be equally unstable.

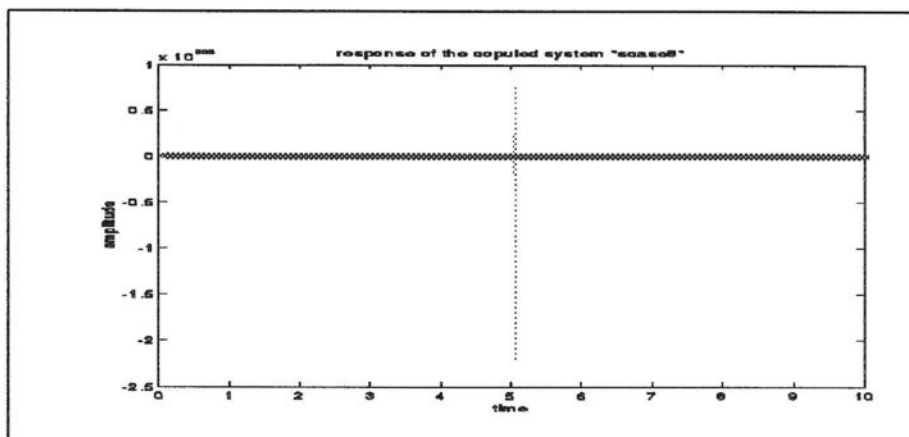
Some of the other result obtained is as shown in appendix D. The output of the system are identical

cases was considered, (see listing in Appendix D) and the simulation taken around this frequency region and varying other parameters such as the amplitude and the convergence factor, μ , the result reflects that the system cannot be stable in this region.



(Fig 6.3)
Output simulation of the system at 18kHz, (case2)

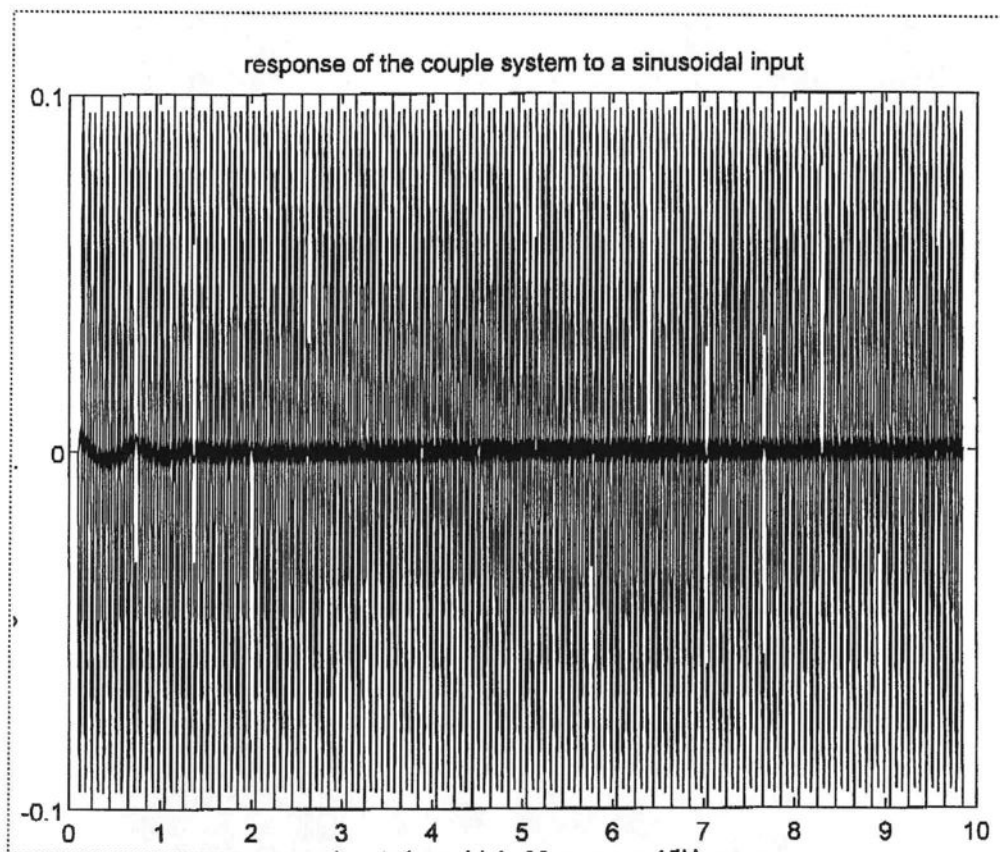
It was noted that for certain combination of amplitude and frequency, the output of the system was negative, hence it behaves like a comparator, (see figure 6.4).



(Fig 6.4)
Output simulation for the system at 18kHz

The observation made for other cases were identical, mostly with a positive response especially when the amplitude of the input signal is greater than unity, as a result some of this figures will not be included.

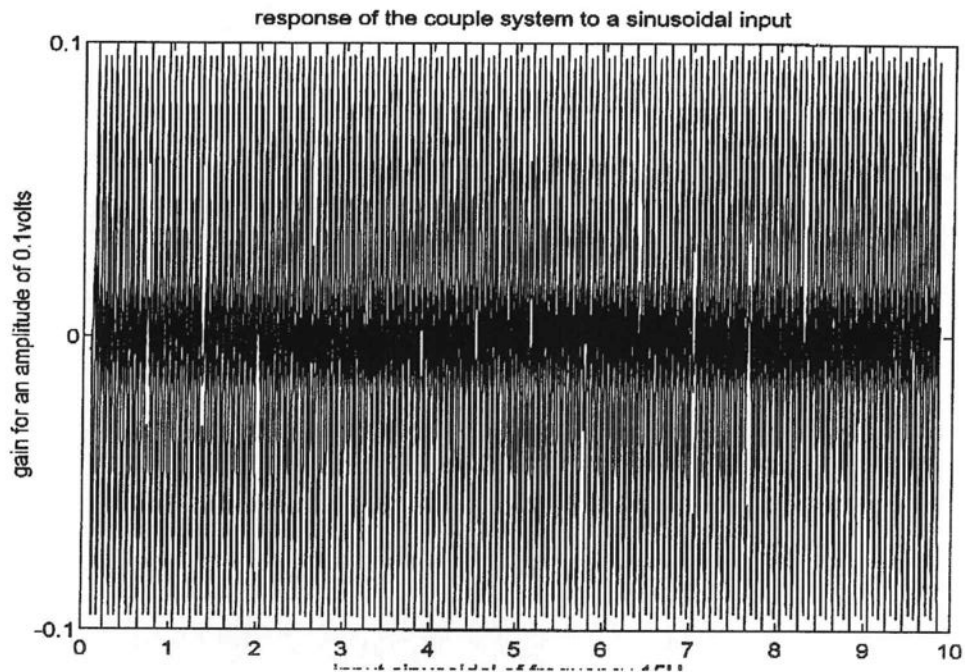
However for extremely low frequency of 6Hz, amplitude of 0.1unit with μ as 0.01 and consequently low gain, the system began to respond to the sinusoidal input as shown in figure 6.5, but with a very low amplitude about the zero origin about the amplitude axis.



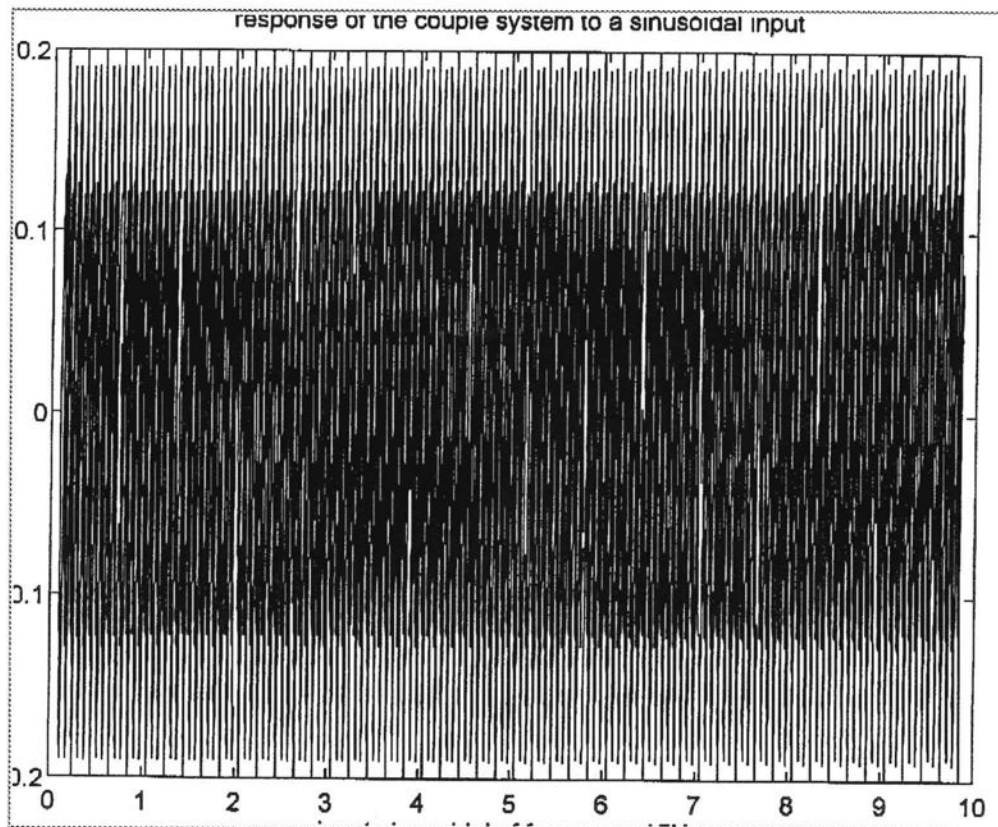
(Fig 6.5)
Output simulation of the system at 6Hz, (case9)

Other parameters of the input signal was kept constant at this instance except for its amplitude which was varied and the results obtained are depicted as in figures 6.6 and 6.7.

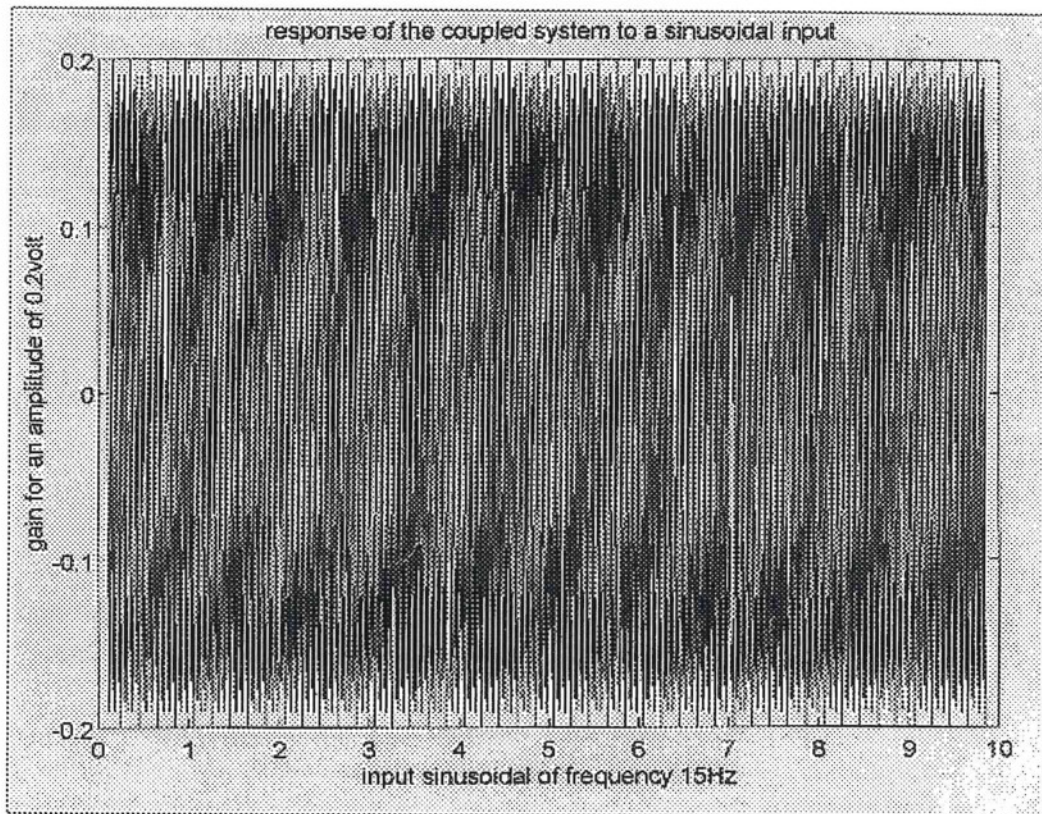
The larger amplitude represent the input signal as the output rises gradually.



(Fig 6.6)
Output simulation of the system at 6Hz, (case21)



(Fig 6.7)
Output simulation of the system at 6Hz, (case22)



(Fig 6.8)
Output simulation of the system at 6Hz, (case24)

It was found that for a value of μ equal 0.09, amplitude 0.2unit and a gain of 129.6, the input and output signal becomes unity, figure 6.8, after which this system began to exhibit instability. Therefore it was concluded that these values are the optimum value for the system without degradation.

6.3 UNCOUPLED SYSTEM

Further investigation was conducted by decoupling the system and examining the system response for identical input as the coupled system. Simulation result reveals that the gain for the decoupled system cannot be imposed arbitrarily, otherwise a solution may not exist for the system.

To achieve this, it is desirable to write equation 5.10 in a *state space* form, that is

$$\dot{x}(t) = A x(t) + B u(t) \quad \dots\dots\dots (6.3.1)$$

$$y(t) = C x(t) + D u(t) \quad \dots\dots\dots (6.3.2)$$

where $\dot{x}(t)$ is the state derivative vector, **A**, the state matrix, **B**, the state input, $x(t)$ and $u(t)$ are the state and input vectors respectively, $y(t)$, is the output vector, and **C** and **D** are the output matrix respectively. For most simple system **D** is usually equal to zero and so is the case with this system under consideration.

From the analogue diagram of figure 6.1, the state space equations can be written directly. The integrator to the right is the output to the world and this was denoted as y . Noting that the output of this integrator was x_1 and this is the same as y . The output of second integrator was designated x_2 . With this notation, therefore the state space equation can be written thus -

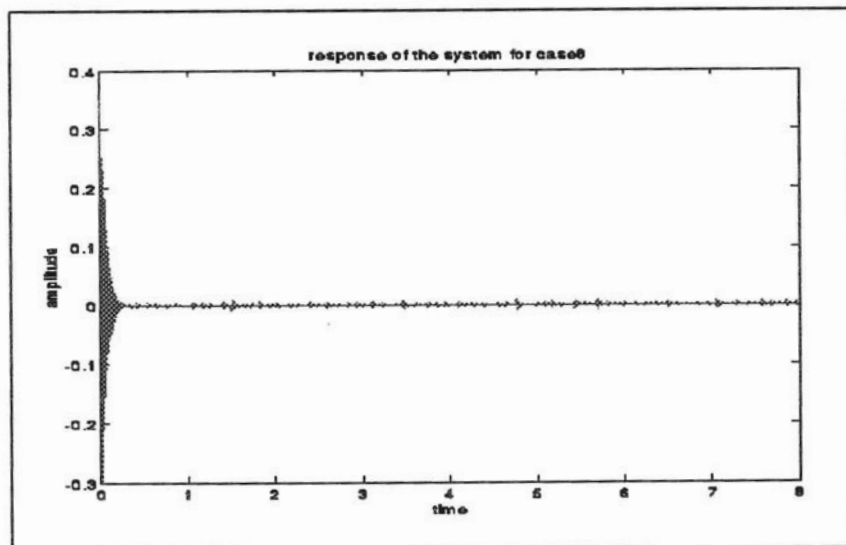
$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \frac{-2\mu A^2}{T_s} & 1 \\ -\omega_o^2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{-2\mu A^2}{T_s} \\ 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \dots\dots (6.3.3)$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \dots\dots\dots (6.3.4)$$

To proceed with the simulation, the problem requires some initial conditions. The initial condition imposed on the system therefore was x_1 equal zero and x_2 equal 0.35.

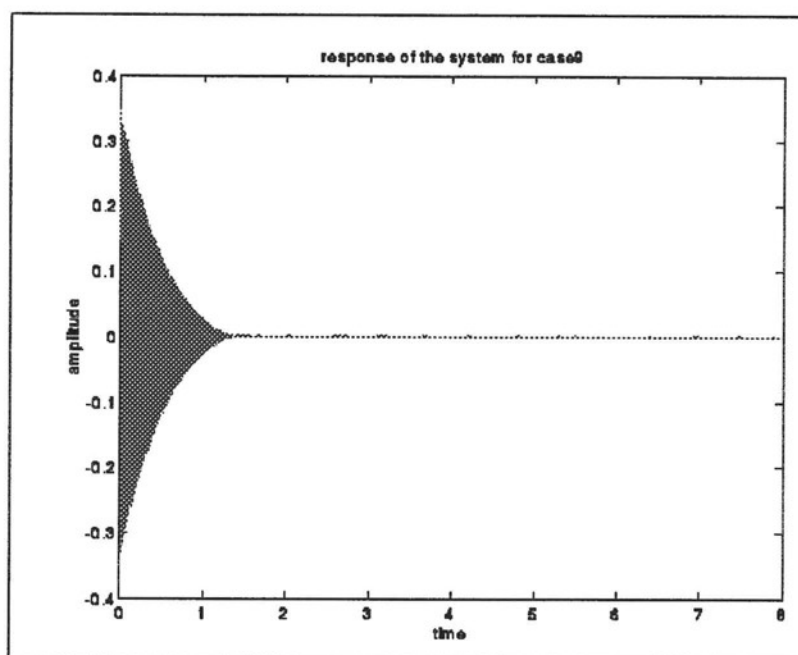
However, following the explanation given earlier, for high gains, there was no solution to the problem, unlike the case of the analogue simulation. The method of solution considered was the *Runge-Kutta* approximation. Since solution does not exist for very high gains in this instance, (the equation was unable to converge), and hence simulation results could not be obtained for cases 1 through 6.

Proceeding with case 7, the following result was obtained as depicted in figure 6.9. With a smaller amplitude and lower frequency input signal, the system appears to converge but grossly unstable.



(Fig 6.9)
Output simulation of the system at 6Hz, (case7)

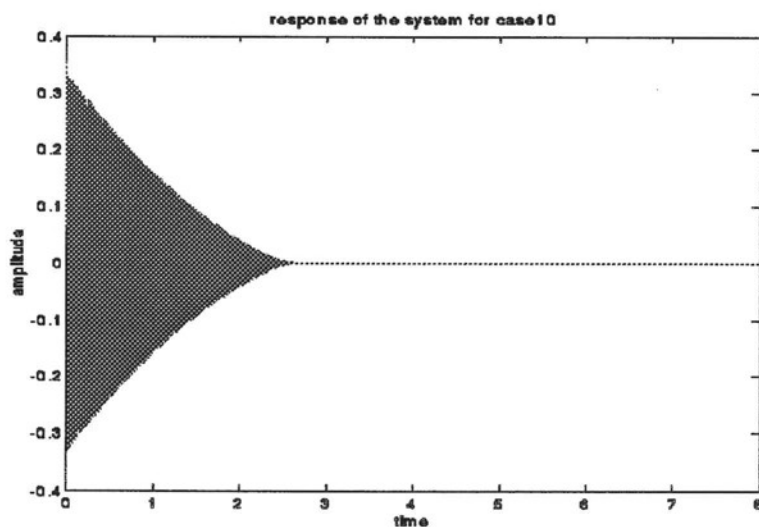
Several other simulation was obtained around this region of response, and the results obtained was depicted in figure 6.10 and 6.11 respectively. Therefore it was concluded that the system cannot be stable in this region.



(Fig 6.10)

Output simulation of the system at 6Hz, (case8)

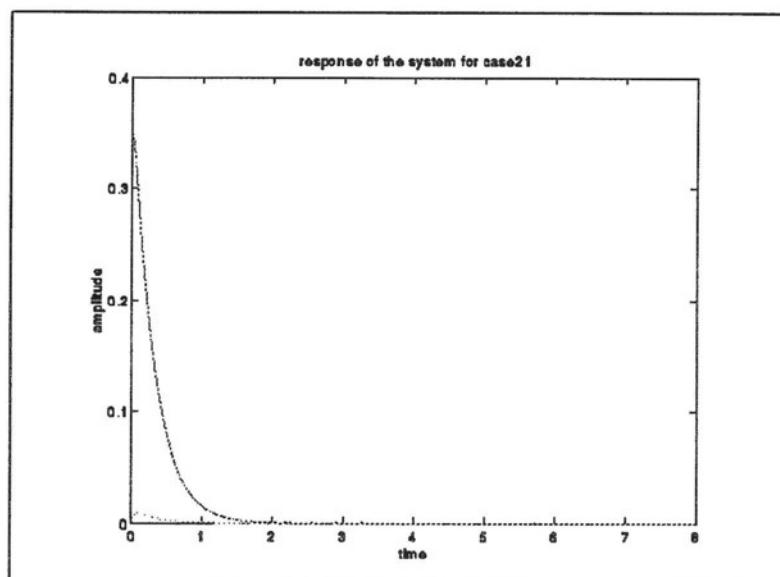
However, it was found that when the parameters of the input as it was the case with the coupled system, the system response sharply. It was found that the system decays from the initial condition, $x_2 = 0.35$ and converges to zero, and was stable as shown in figure 6.12.



(Fig 6.11)

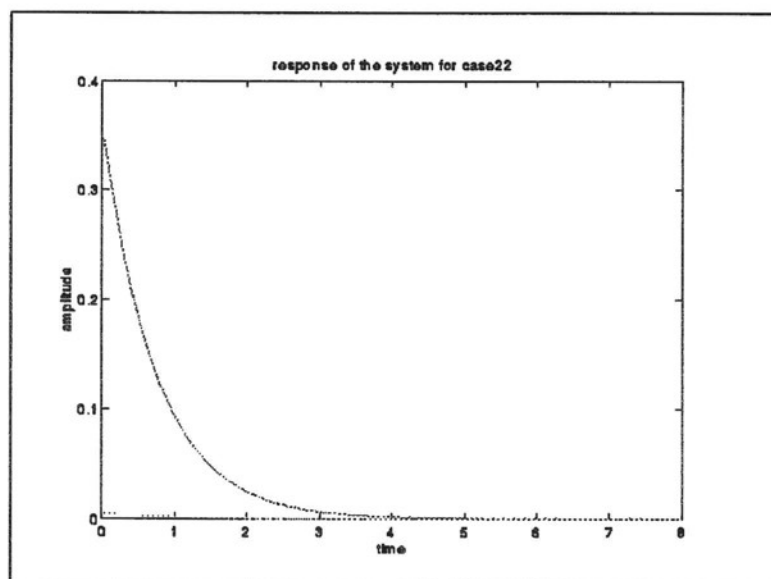
Output simulation of the system at 6Hz, (case9)

Other cases considered were cases 22 through to 24, and the result obtained are equally shown. Hence it can be concluded that the proposed adaptive filter will work effectively only for low frequency input signals.

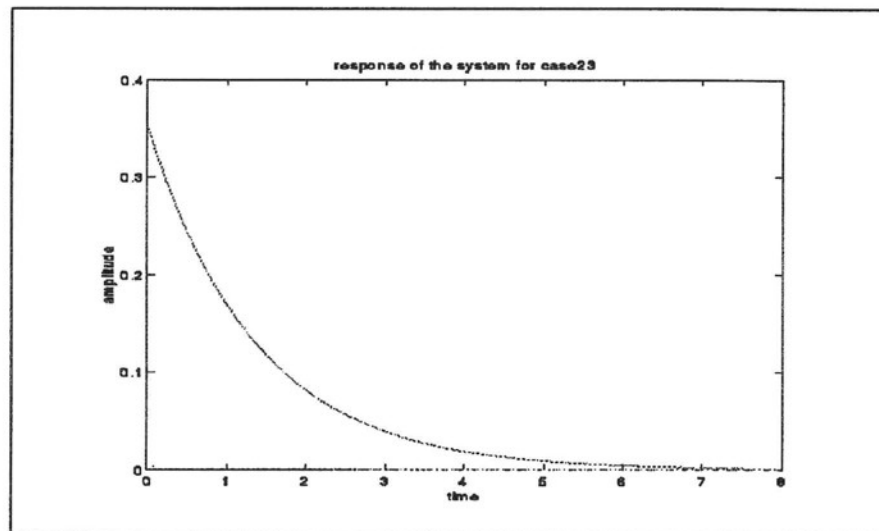


(Fig 6.12)
Output simulation of the system at 6Hz, (case21)

With a gain less than the case of 21, then convergence was gently approached, as shown in figure 6.13.

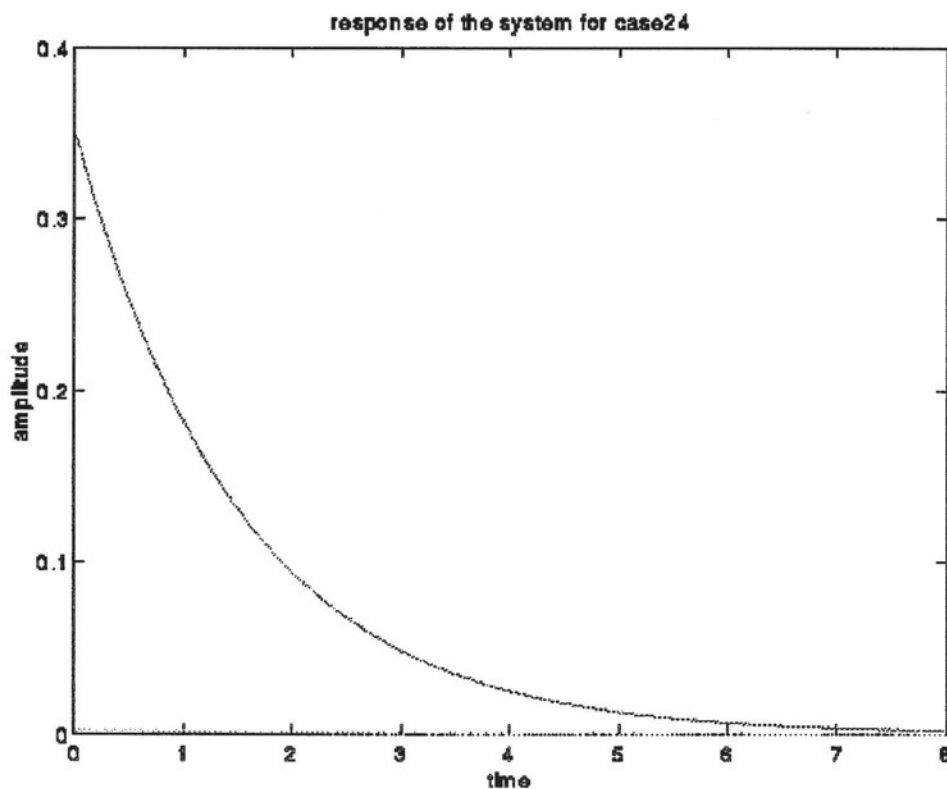


(Fig 6.13)
Output simulation of the system at 6Hz, (case22)



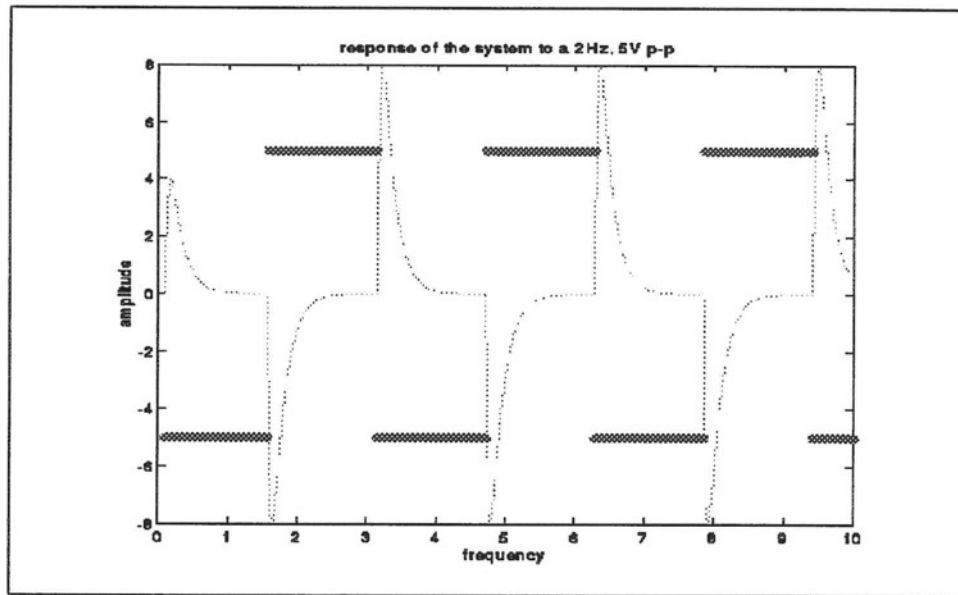
(Fig 6.14)
Output simulation of the system at 6Hz, (case23)

It can be seen that the second variable, x_1 remains on the x-axis, this was because of the initial condition placed on the system. Although it was found that with any arbitrary values of x_1 , if the solution exist, it converges faster compared to x_2 , not shown in this instance.



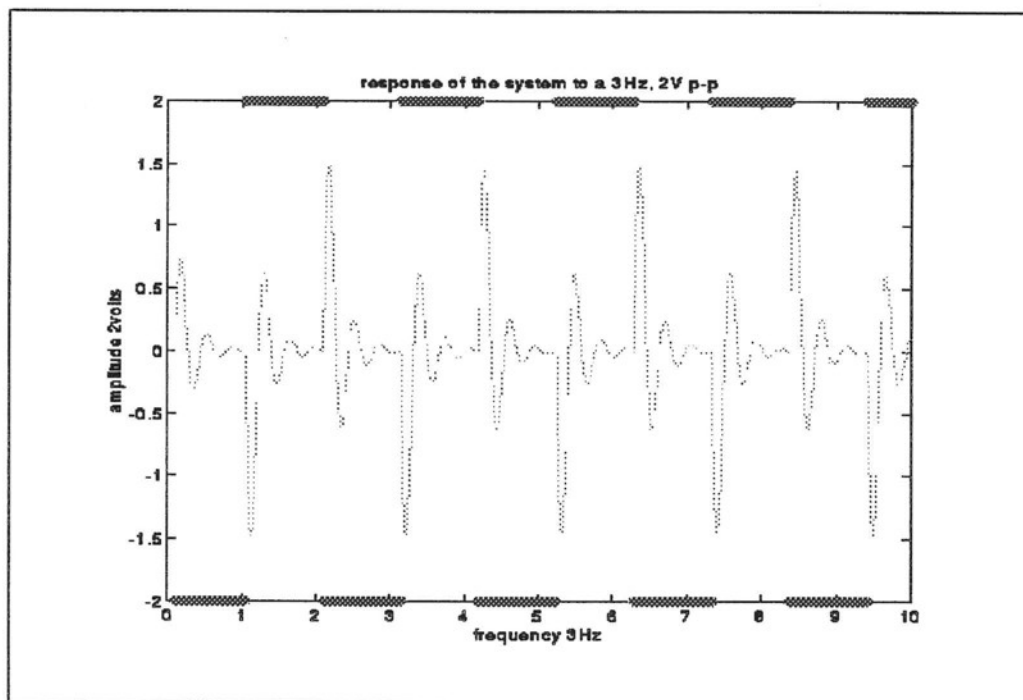
(Fig 6.15)
Output simulation of the system at 6Hz, (case24)

With these result it was concluded that gain imposed on the system cannot be too high without inducing stability in the system.

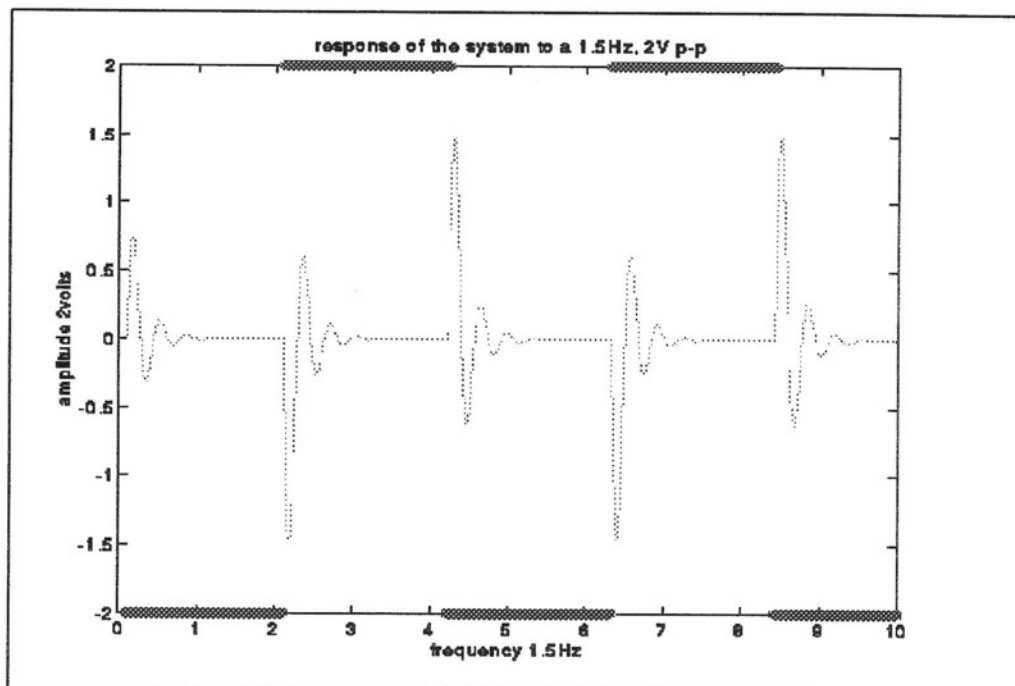


(Fig 6.16)
Output simulation of the system at 6Hz, (case22)

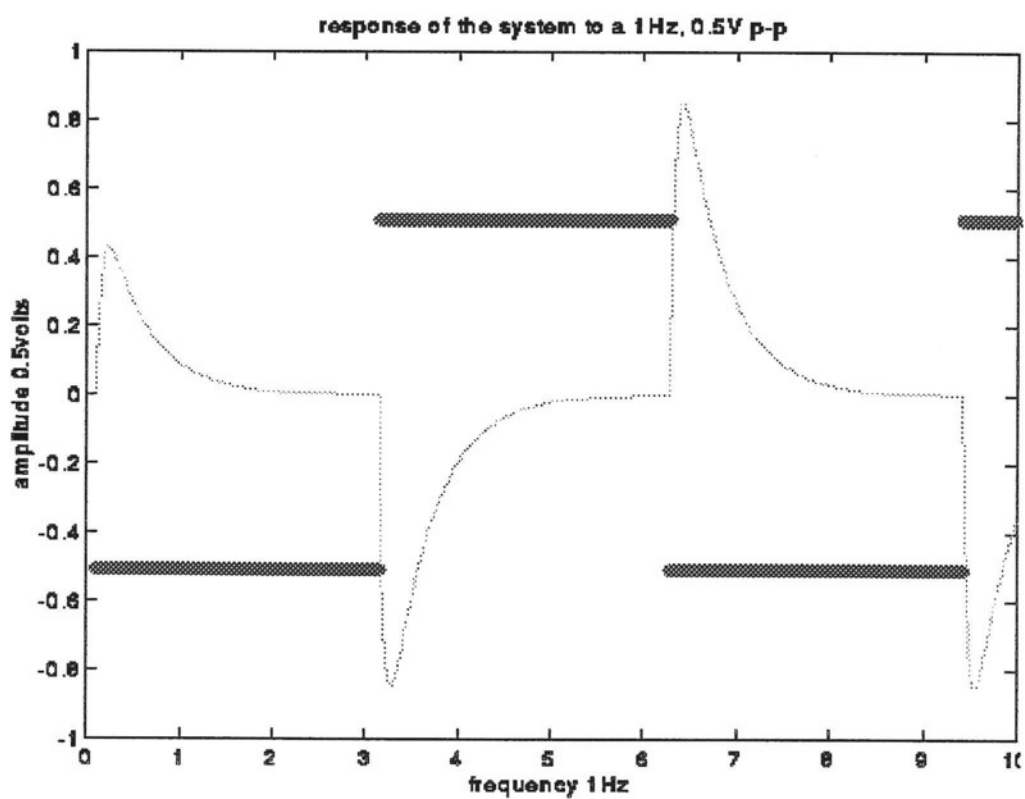
Following these observation other input, such as the square wave was considered for low frequency 5 to 1Hz respectively and the result obtained was depicted in figures 6.16 through 6.18. The simulation began with a frequency of 5Hz and amplitude of 5v.



(Fig 6.17)
Output simulation of the system at 6Hz, (case23)



(Fig 6.18)
Output simulation of the system at 6Hz, (case24)



(Fig 6.19)
Output simulation of the system at 6Hz, (case24)

CHAPTER 7 SOFTWARE DESIGN

7.1 INTRODUCTION

The design of the software employ the top-down design approach. This concept highlights the inter relationship between individual modules as it would be seen later. Then each module was further expanded by a flow chart diagram which conforms with the British Standard (BS6224).

This modularity concept was deemed vital to the overall success of this project. Firstly it aids debugging and secondly it lends itself the solution to complex activities which would otherwise be incomprehensible should one attempt to solve the problem as a single entity.

7.2 MAIN PROGRAM

The main program encompasses all the individual modules. It is a menu driven program and the structural diagram was shown in fig 7.1. The structural diagram closely reflects the function calls made during the design process of the program.

The program runs by inviting the user to select an option via the module, *display menu*. The high-level code designed for the module that display the menu program was depicted in figure 7.2. This figure is usually referred to as the *Jackson Structure Diagram(JSP)*. The options available are as follows:

1. Design Filter
2. Digital to Analogue conversion
3. Off-line digital filter
4. Real-time digital filter
5. Off-line adaptive filter
6. Real-time adaptive filter
7. Exit

Each of this option will be chosen and elaborating on the important features of the high-level code.

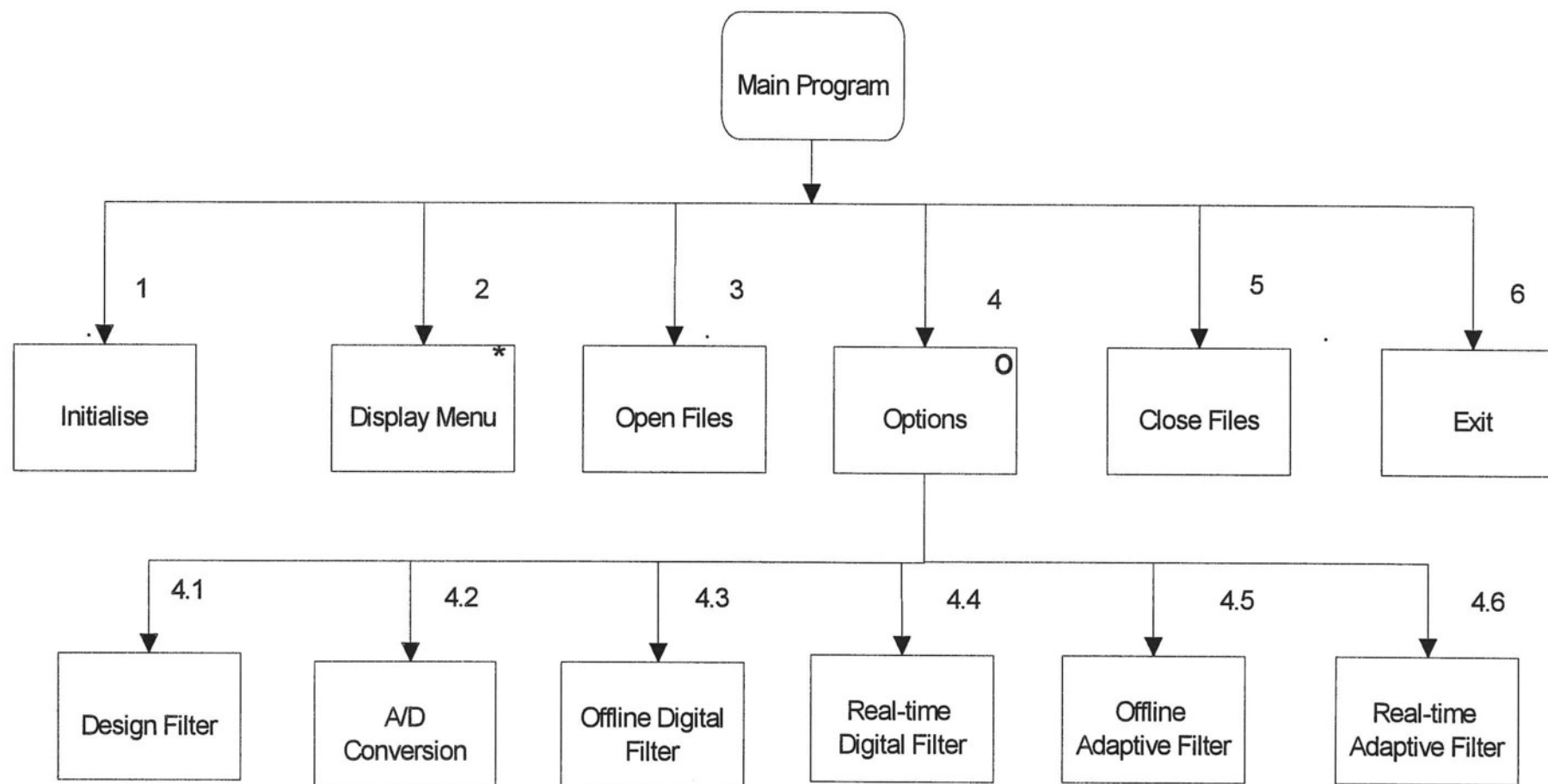


Fig7.1
Structural Diagram for Adaptive Filter Design

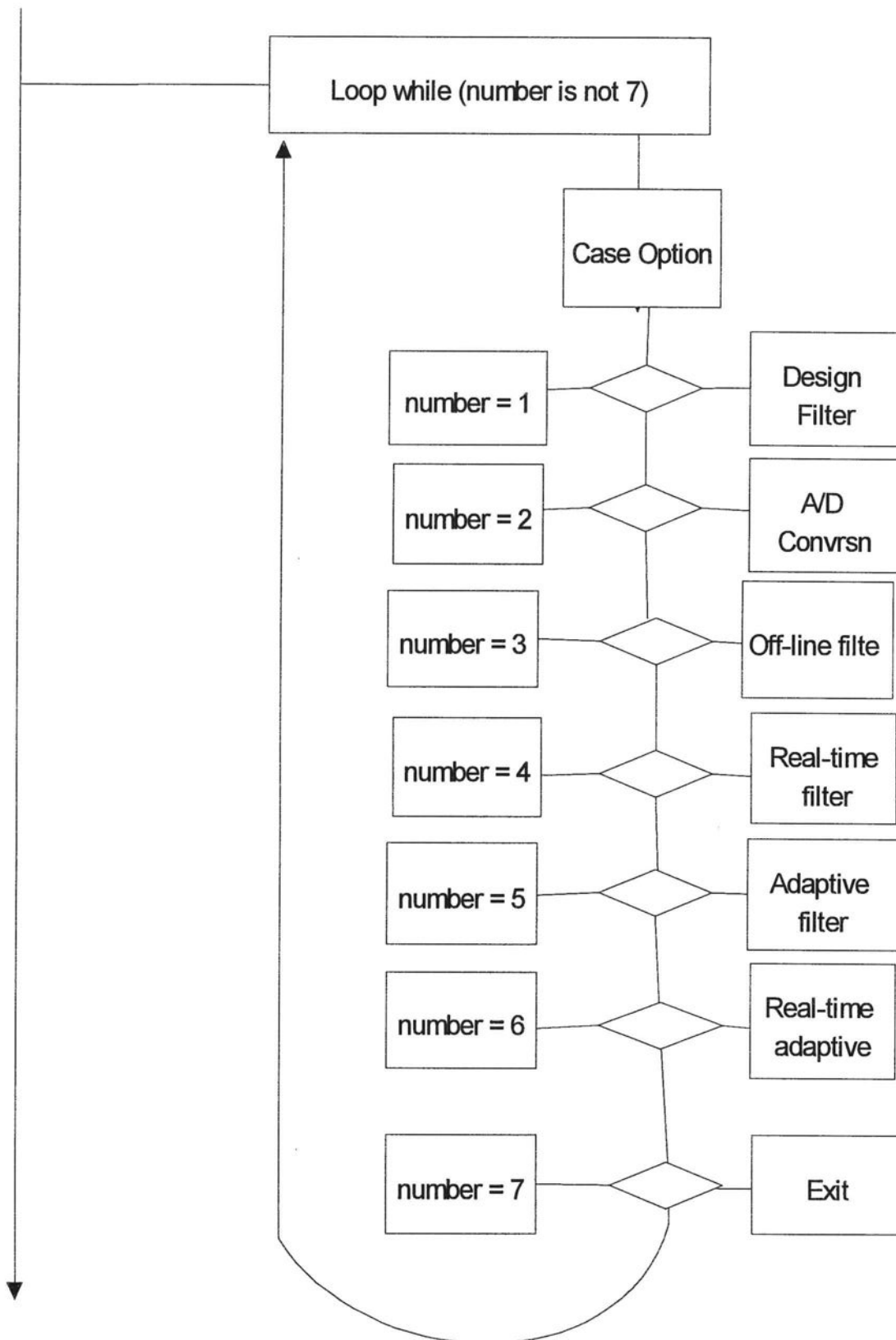


Figure 7.2
illustration of flowchart diagram for displaying menu

7.3 **DESIGN FILTER**

This module was written under the filename "*Window.c*". It employed the window method described earlier sec 4.4.1, to calculate the coefficient of the desired filter. Five Windows featured in this program, namely

- 7.3.1 Rectangular window
- 7.3.2 Hamming window
- 7.3.3 Hanning window
- 7.3.4 Blackman
- 7.3.5 Kaiser window

However, the Kaiser window was employed in this thesis and its consequently its design would be expatiated using the BS6224 standard. Other windows conform similarly to this standard and would not be discussed.

Moving down the chart, if option 1 (i.e number = 1) was selected, the program passes control to the filter design module. The fig 7.3(a) and 7.3(b) shows he top-down design structure for the window program.

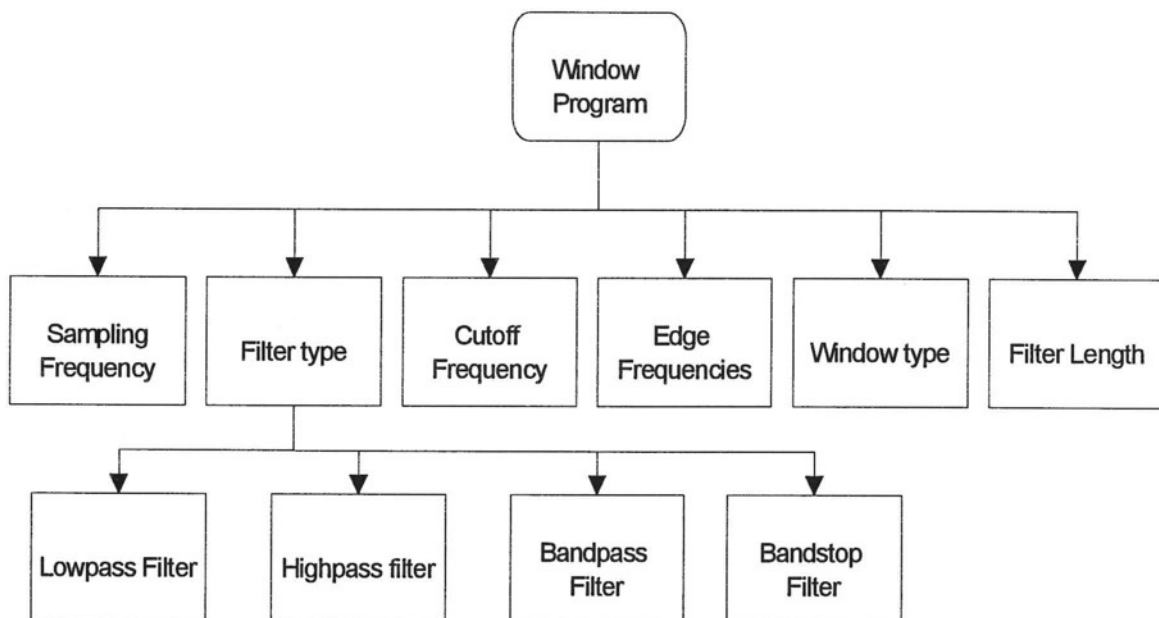


Figure 7.3(a)
Structural diagram for window filter design

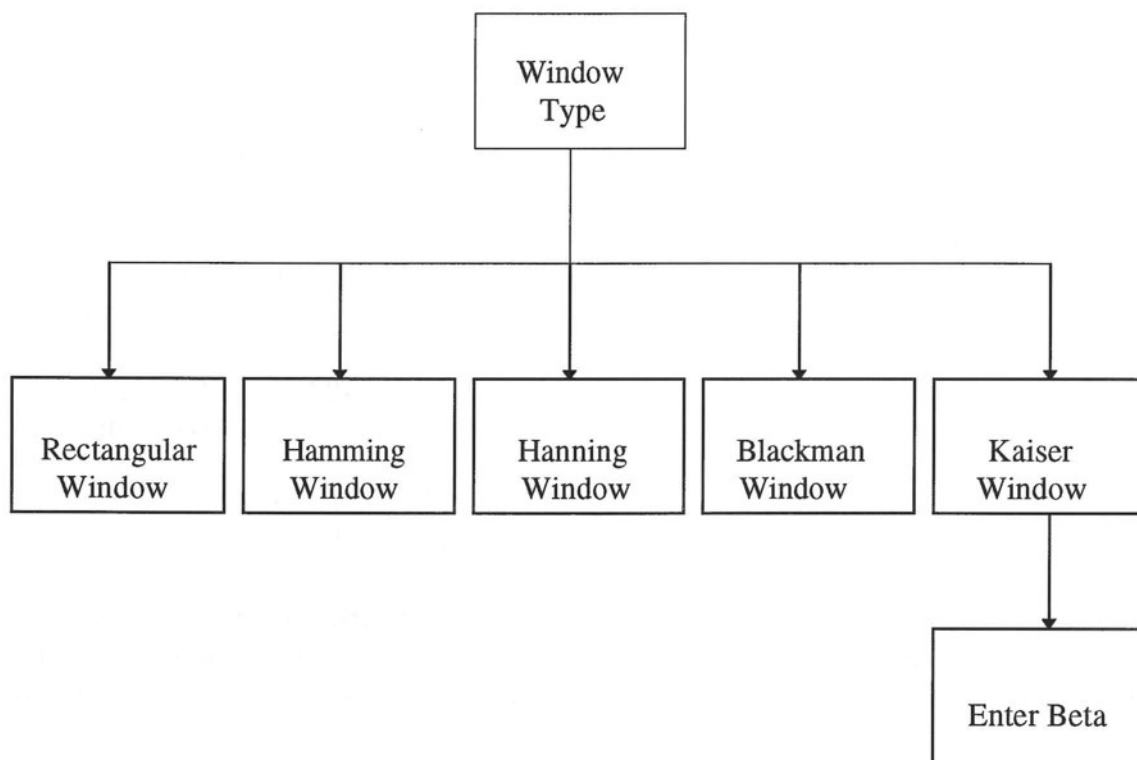


Fig 7.3 (b)
Structural diagram for window program design

Furthermore appendix C elaborate on this structure revealing the relationship between the high-level language and the actual coding.

7.4 Analogue to Digital conversion

This module was design to appreciate how the input and output signal works via the ADC converter. The ADC conversion was achieved using the PCL-812 interface card, (see equipment list, sec 4.1).

This module was written under the filename "*sample.c*", the project file employed was called "*sample.prj*". The linker files, "*812TCL.obj*" (i.e files making up the program) was also included in the directory before the program could run. However, the program was run and the result was as expected. The output signal was taken from the digital-analogue #1 and fed into the oscilloscope, and a step-like sinusoidal waveform was expected, however, the sampling frequency (i.e 18kHz) was employed in this instance was quite high and consequently this phenomenon was not visible. The nex module considered was the off-line digital filtering.

7.5 OFF-LINE DIGITAL FILTERING

This module was design as an intermediate step before the actual on-line (or real-time) digital filter. Following this approach provides an ample opportunity to examine the nature of the data stored in the output file after performing the digital filtering. However, the results obtained would be discussed further in chapter 8. The datafiles for this operation were called "*winlowps.dat*," "*testnch.dat*" and "*stbndps1.dat*" for the lowpass filter, notch filter and the bandpass filter respectively.

7.6 REAL-TIME DIGITAL FILTER

This module employed the program "newdiga.c" It was identical to program sample.c except that instead of writing to a file the output was directed to the oscilloscope directly via the interface card after employing some suitable scaling algorithm. The accompanying files were called "newdiga.prj" and "812TCL.obj." The filters designed were then tested in turn and the results obtained were examined in detail in sec 8.7.2.1, 8.7.2.2 and 8.7.2.3 respectively.

7.7 ADAPTIVE FILTERING

This program was made up of two parts, namely -

7.7.1 Off-line adaptive filtering

7.7.2 Real-time adaptive filtering

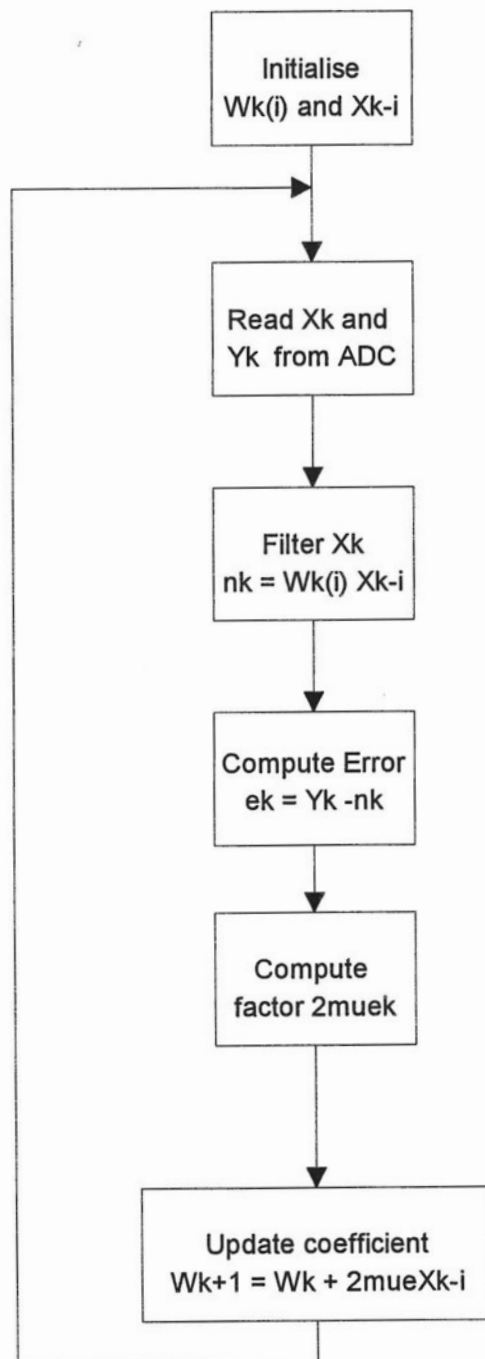
7.7.1 Off-line adaptive filtering

Work began on this module, by adopting the algorithm proposed in chapter 5 and this was also depicted in fig 7.4., as a block diagram illustrating the logic on how to achieve the filtering process.

For this task, a file was created with the name "myadapt.c." It works by initialising the arrays which holds the reference signal data r_k , the corrupted signal y_k , the coefficient of the filters, the error signal data and the filter weights, w_k , to mention a few. In this instance the filename "heart3.dat," was loaded and read into the y_k array, (ie memory holding the corrupted signal). These arrays are filled by loading corresponding data files using a pointer variable.

7.7.2 Real-time adaptive filtering

As at the time of the report, further research was still going on. However, the success of this section depends very much on the outcome of sec 7.7.1. And consequently it would provide a basis for further investigation.



(Fig 7.4)
adaptive filter algorithm

CHAPTER 8 TESTING AND RESULT

8.1 INTRODUCTION

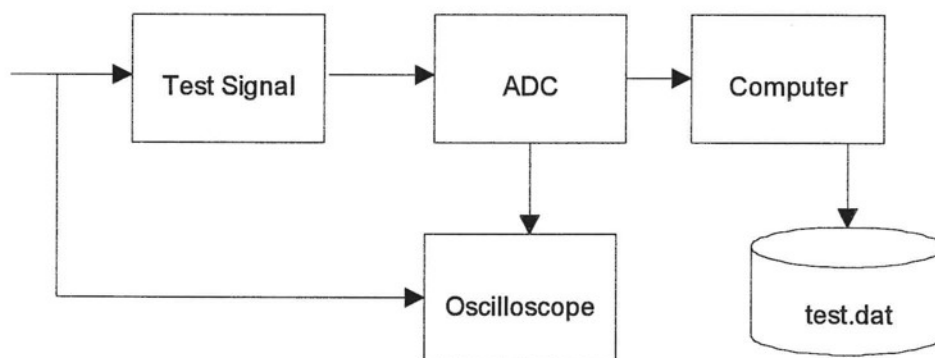
This part of the task describes all the result of the procedures that were carried out to accomplishing the set objectives. It further compare the some of these result to some known standards, for example the filtered ECG signals obtained was compared with some results obtained in cardiology studies and medical research, and attempt was made to offer an explanation wherever a discrepancy exist.

The entire task can be classified under three subheadings, filter and software design, and simulation.

The design of the filters have been fully covered in chapter 4, simulation of the model have also been discussed substantially in chapter 6 and the design of the software have been elaborated in chapter 7. This chapter will begin by describing how the experiments were set up and the results achieved. Furthermore in the latter part of this chapter, analysis of the ECG (vector) signal was carried out and investigation of the covariance matrix was conducted and consequently an estimation of the parameter, μ , for the ECG signal was made.

Finally the adaptive algorithm suggested in chapter 5, was fully implemented employing the value of μ , estimated from the covariance matrix, in the first instance, and then employing the simulated value of μ and comparing the results obtained.

8.2 INITIAL TESTING



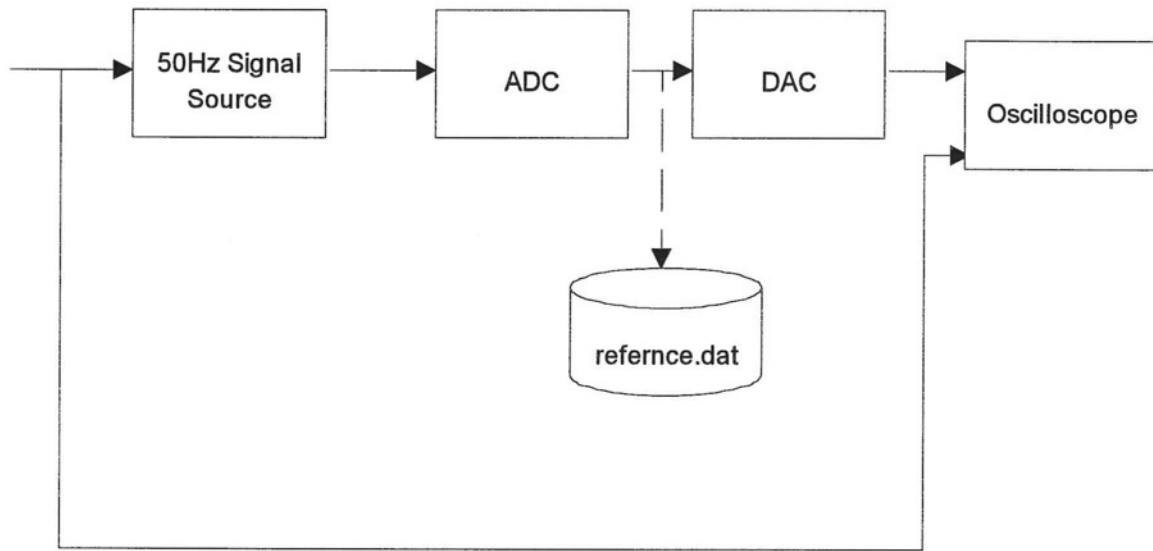
(Fig 8.1)
illustrating the set-up used to verifying the presence of a test signal

Initial testing began by designing software that would sample a test signal obtained from a signal generator, fed into channel #0 of the ADC converter. The data acquired was then stored in a datafile called "*test.dat*." If the "*test.dat*" datafile was created with data successfully written to it, then by typing `dir "test.dat"` at the Dos prompt, a value other than 0 byte signifies that the content of the file was not empty. Observation confirms the success of this test. To investigate further if the content of the file is as expected, a '`type`' command can be issued at the Dos, followed by the filename.

8.3 SOFTWARE TESTING AND SIGNAL ACQUISITION

8.3.1 Acquisition of 50Hz signal in real-time

Following the success of the test in the set-up in figure 8.1, the next stage of the experiment was designed to send back data that was acquired in real-time, so that the output signal follows the input signal exactly. This particular test was considered important so as to get to know how the ADC interface card works. As a consequence the signal generator was set to 50Hz, identical to that of the mains frequency, see figure 8.2(a).



(fig 8.2(a))

experimental set-up to show simultaneous display of the 50Hz signal

The software design to meet this criteria was called "*Sample.c*." It consists of the project file called "*sample.prj*," and an object file, "*812TCS.obj*." The program called "*sampfreq.c*," was used to set the sampling frequency of the interface card to 250Hz and the program *sample.c* was written to acquire the 50Hz signal. The signal was sampled via #1 of the ADC and sent back via the #2 of DAC and the obtained wave form, in real-time is as shown in figure 8.2(b).

The plot was achieved by utilising the storage facility on the oscilloscope. When the signal was captured and stored, a plot button was depressed on the oscilloscope. Data stored in the memory were routed from the oscilloscope to the X-Y plotter. From fig 8.2(b), the input signal was depicted by the sinusoidal wave in the upper region of the window and the sampled version was displayed below, hence the following deduction can be made.

Period T,

$$\begin{aligned} T &= 4 \times 5\text{ms} \\ &= 20\text{ms} \\ &= 1/20\text{ms} \\ &= 50\text{Hz} \end{aligned}$$

but $f = 1/T$

This verifies the setting of the frequency intended. Now if the number of samples per envelop cycle, Q is known, then confirmation of the sampling frequency can be made. The number of samples per envelope cycle would be given by

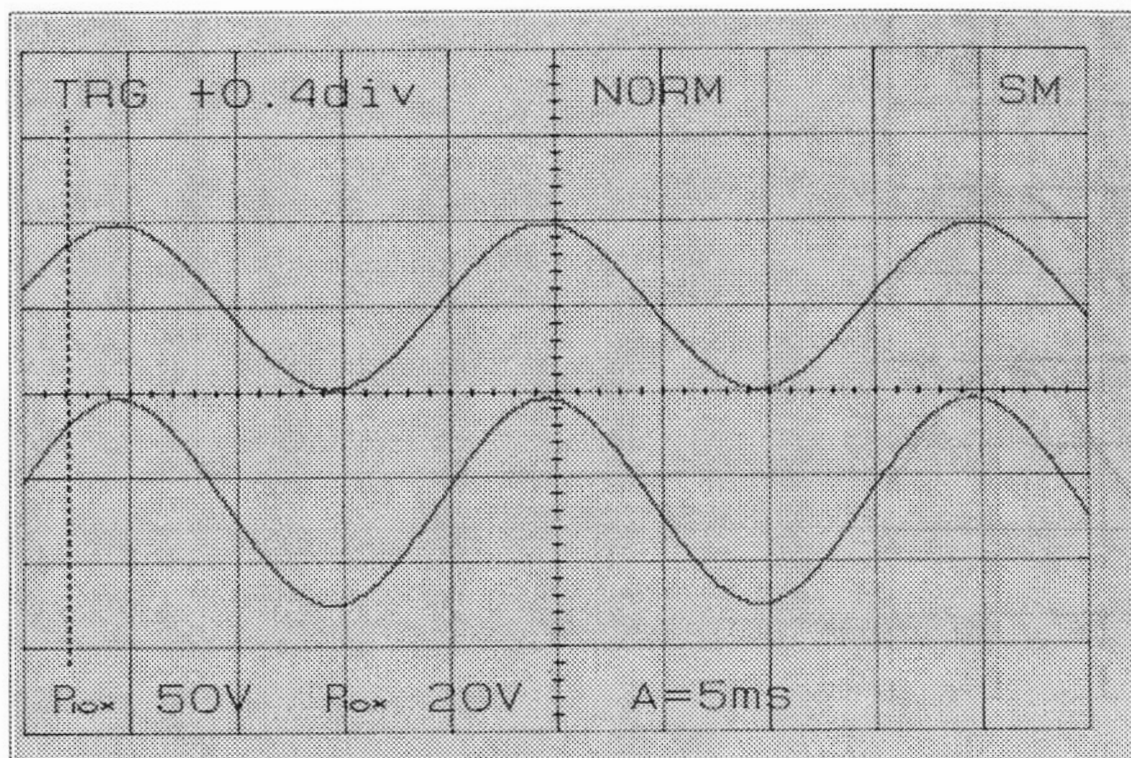


fig 8.2(b)
illustration of a simultaneous display of a sampled data in real-time

$$Q = \tau/T \dots\dots\dots (8.3.1)$$

$$= 2\pi/\omega T$$

where τ , is the sampling interval, (i.e the width of the staircase), although not visible in this instance, and T , the period of the analogue signal.

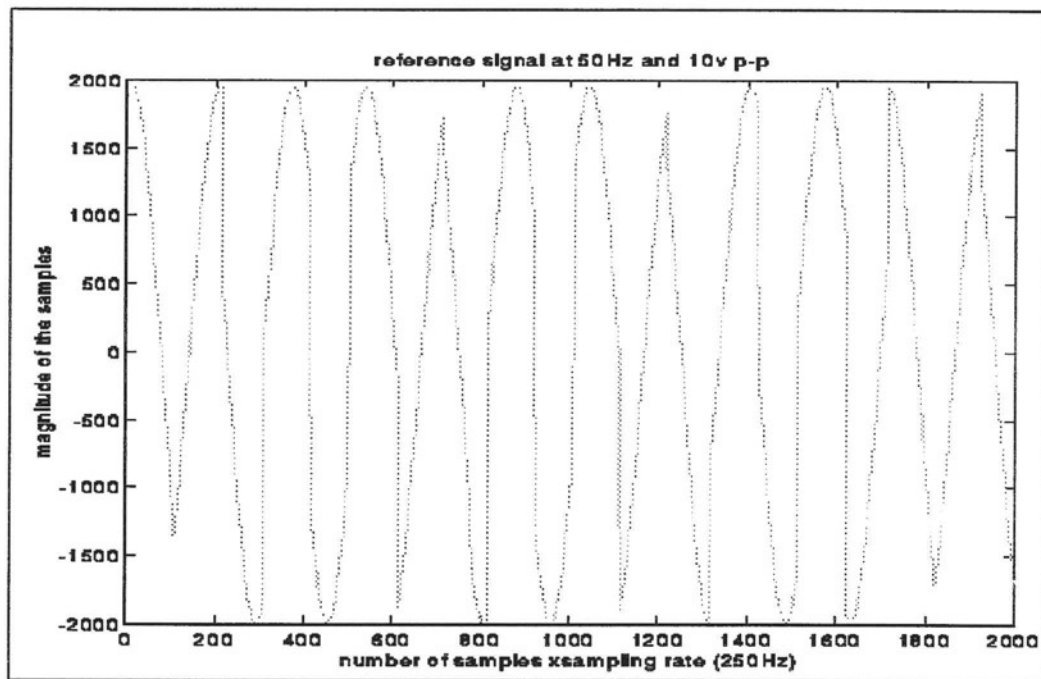
but $T = 1/f_s$

$\Rightarrow f_s/f = Q$

$\therefore f_s = Q \times f$

The desired sampling frequency.

For the latter part of the experiment it was necessary to acquire and store this same data, (ie the reference signal)to a file. As a result another program was written called "*store.c*" which samples the signal at a rate of 250Hz and stores the 50Hz signal in a datafile called "*refernce.dat*." An illustration of the sample obtained was depicted in figure 8.2(c).



(fig 8.2(c))
illustration of the acquired 50Hz reference signal

8.4 ACQUISITION OF PHYSIOLOGICAL SIGNAL

The physiological signal was obtained from the signal generator, this time the signal was set to 6Hz and the same program as sample.c was run except for the slight modification, an output datafile was included called "*lowfresg.dat*," and the acquired data was depicted as in figure 8.3(b). The sampling frequency employed was the same as that of the reference signal, (250Hz). Generally speaking this sampling frequency should be assumed for the rest of this task unless it is stated otherwise.

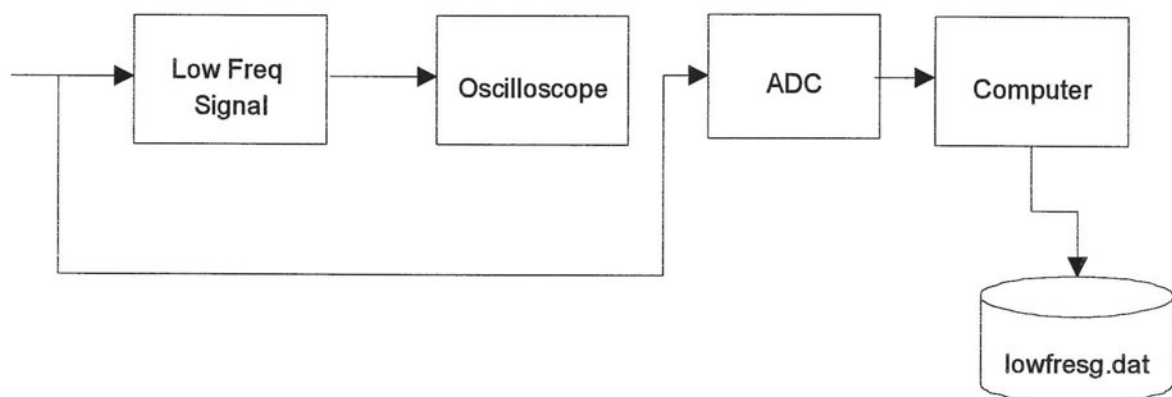
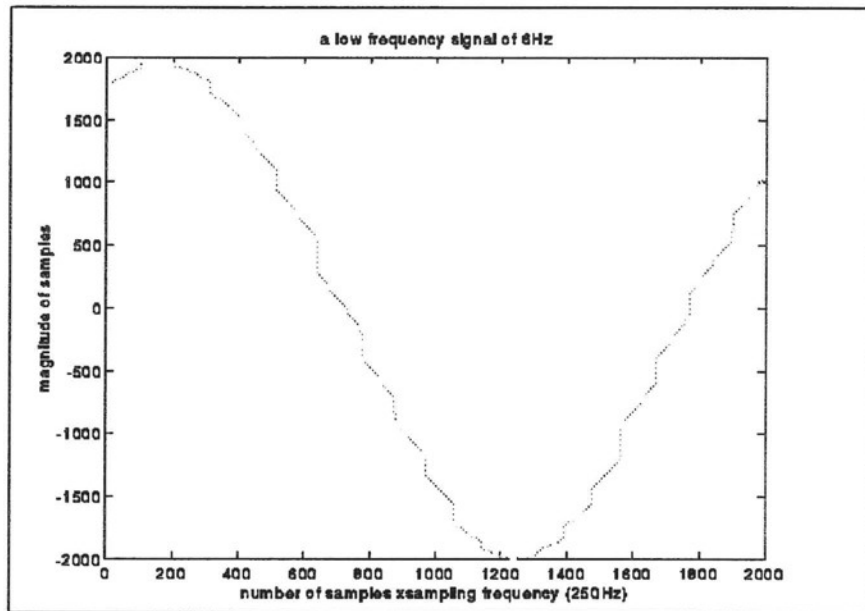
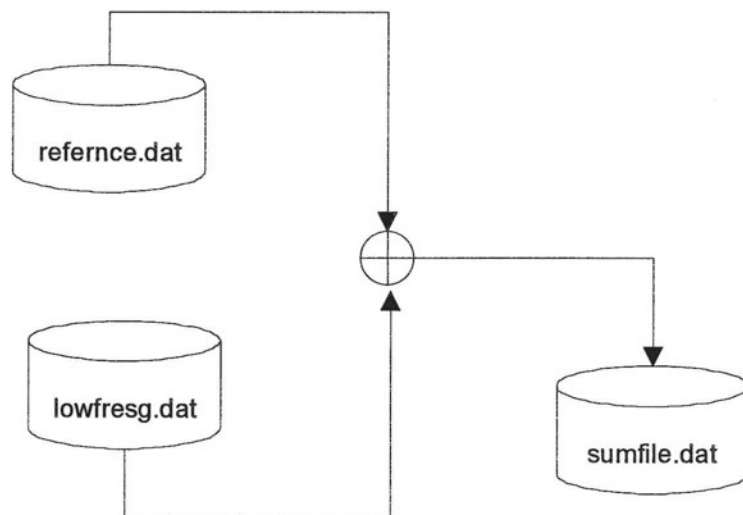


fig 8.3(a)
illustration of the experimental set-up for acquisition of the low frequency signal



(fig 8.3(b))
illustration of the wave profile of the acquired signal

8.5 ACQUISITION OF NOISY PHYSIOLOGICAL SIGNAL



(fig 8.4(a))
illustration of the experimental set-up to acquire a noisy physiological signal

This signal was simulated by adding the content of the files "refernce.dat" and "lowfresg.dat" obtained in sec 8.3 and 8.4 respectively. Since this addition is linear, the principles for processing of signals is applicable. As a result, another programming module was written, called "*read.c*." This program works by prompting the user to supply the names of the file whose data were to be added. The files "*refernce.dat*" and "*lowfresg.dat*" were supplied

respectively, and the resulting data was stored in another datafile called "*sumfile.dat*." The fig 8.4(b) illustrates the acquisition of the noisy physiological signal.

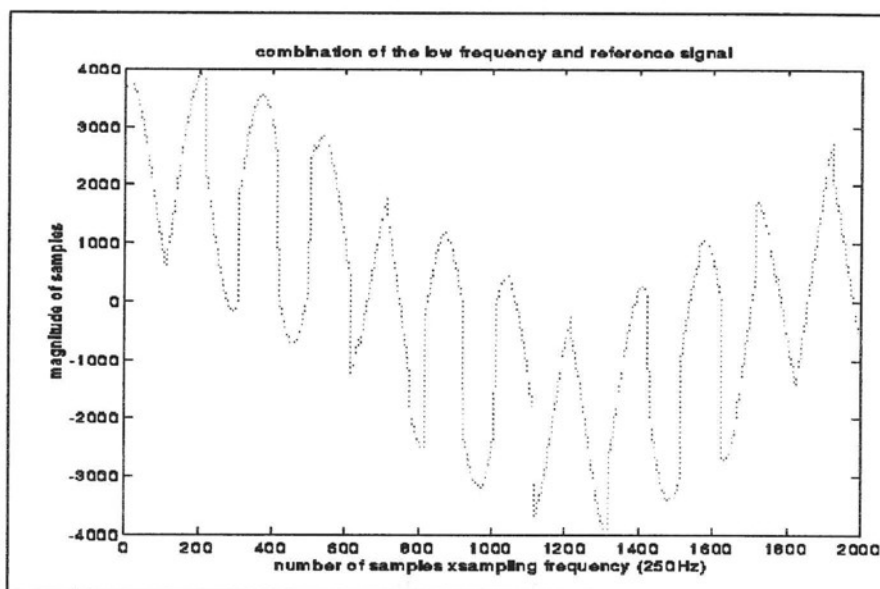


fig 8.4(b)
illustration of the noisy physiological signal

8.6 ACQUISITION OF ECG SIGNAL

8.6.1 ECG signal sampled at 18kHz

The figure 8.5(a) shows the block diagram for acquisition of the ECG signal sampled at 18kHz. One thousand samples was acquired, and the contents of the datafile called "*heart2.dat*" was depicted as in fig 8.5(b)

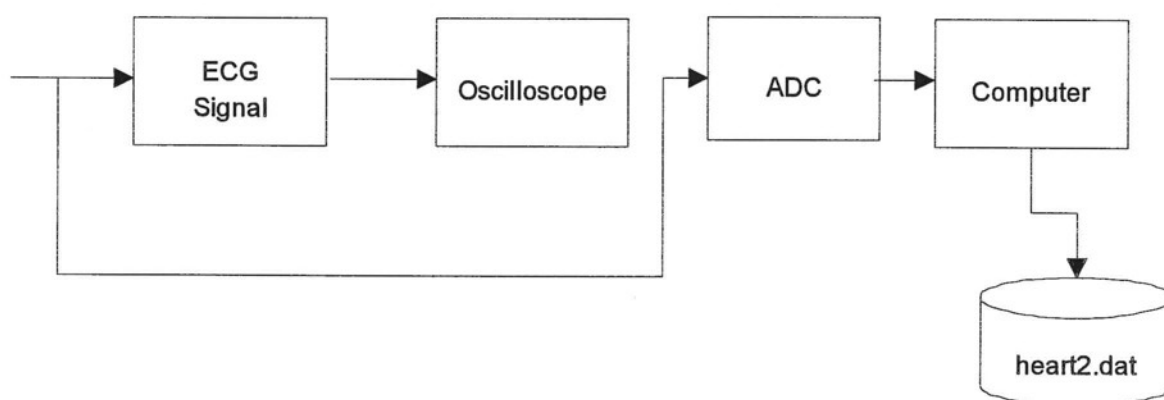


fig 8.5(a)
illustrating the experimental set-up to acquiring the ECG signal

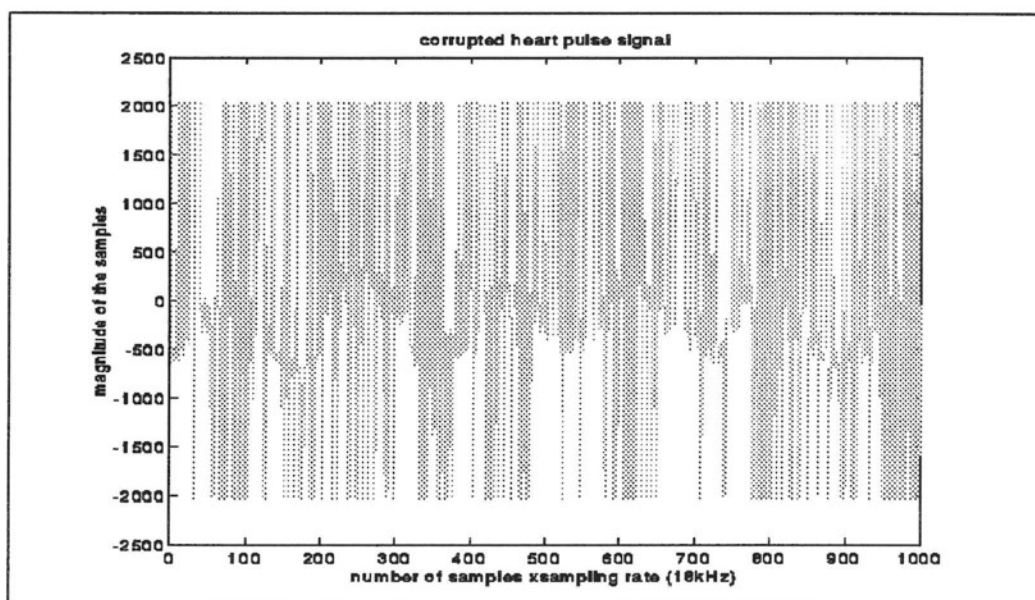
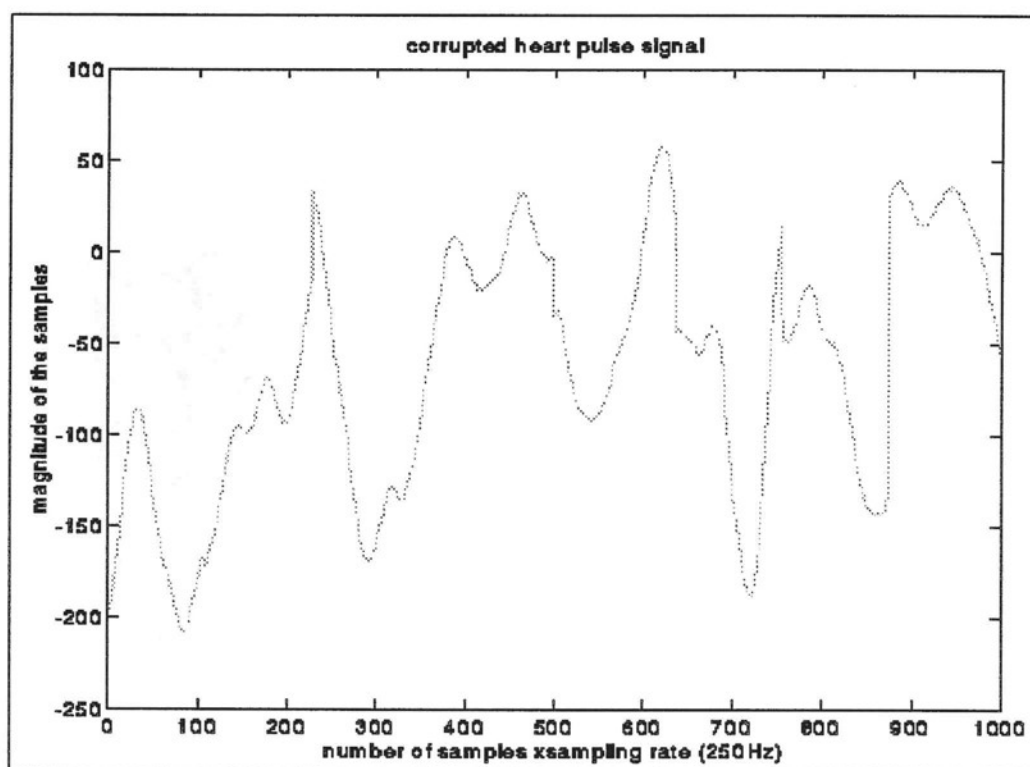


fig 8.5(b))
illustrating the content of datafile heart2.dat of the ECG signal

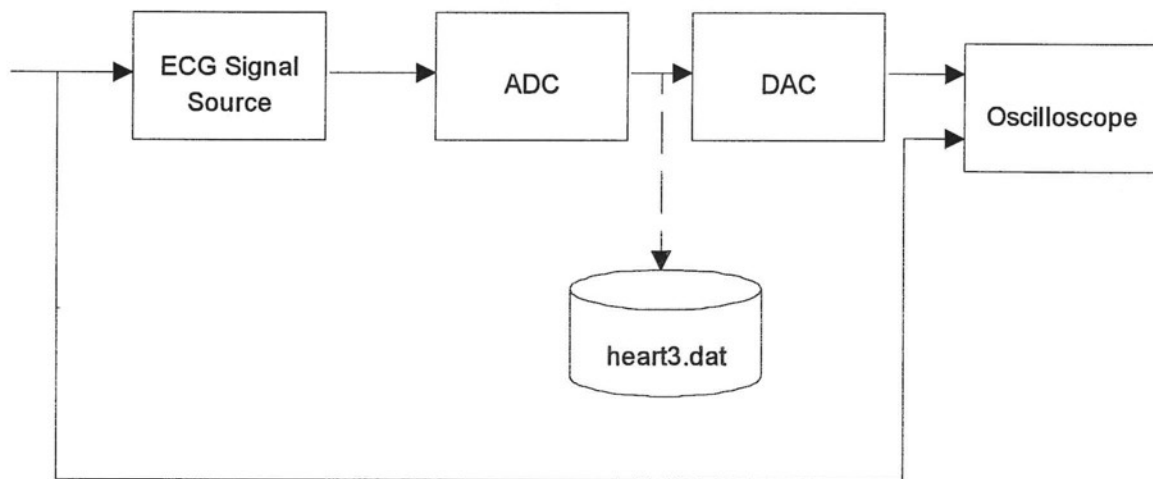
With the same experimental set-up another sample was also taken with a sampling frequency of 250Hz, and the data was stored in another file called "*heart3.dat*". The figure 8.5(c) illustrates the content of the stored data.



(fig 8.5(c))
illustrating the content of datafile heart3.dat of the ECG signal

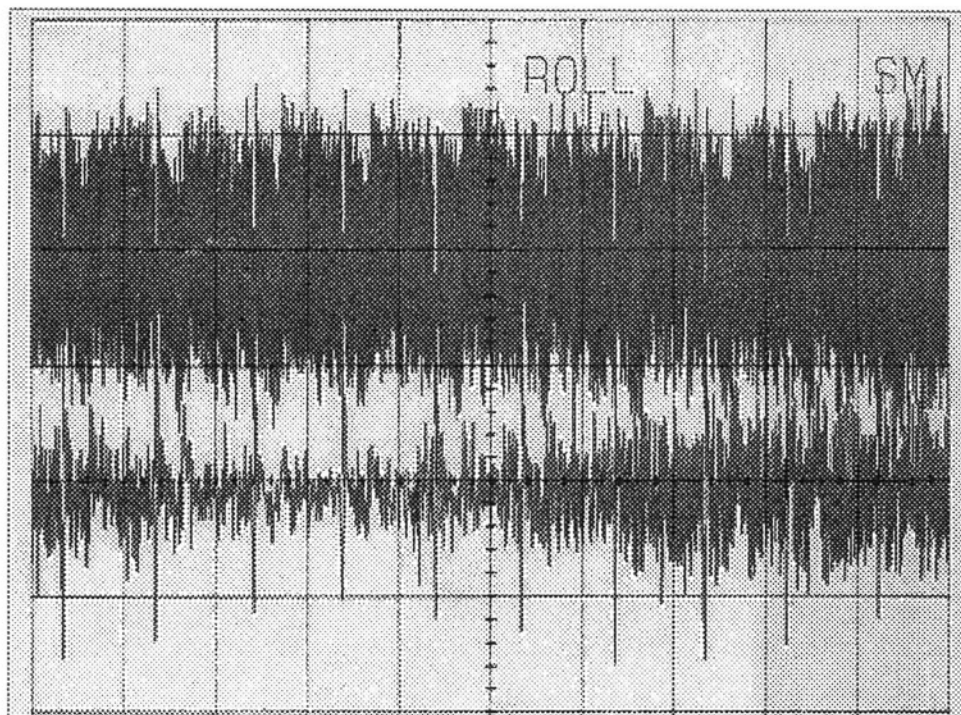
In either instance 1000 data points was considered. Comparing both ECG signals the sample with the higher sampling frequency appears more noisy compared to that obtained at 250Hz, (i.e fig 8.5(c)). This could be compounded by the noise generated by the ADC converter at that high frequency.

The next stage of the task consider the simultaneous plot of the ECG signal followed by examining the action of each of the filter on the acquired signals.



(fig 8.5(d))

illustration of the experimental set-up for the simultaneous display of the sampled ECG signal in real-time



(fig 8.5(e))

illustration of a simultaneous display of a ECG signal in real-time

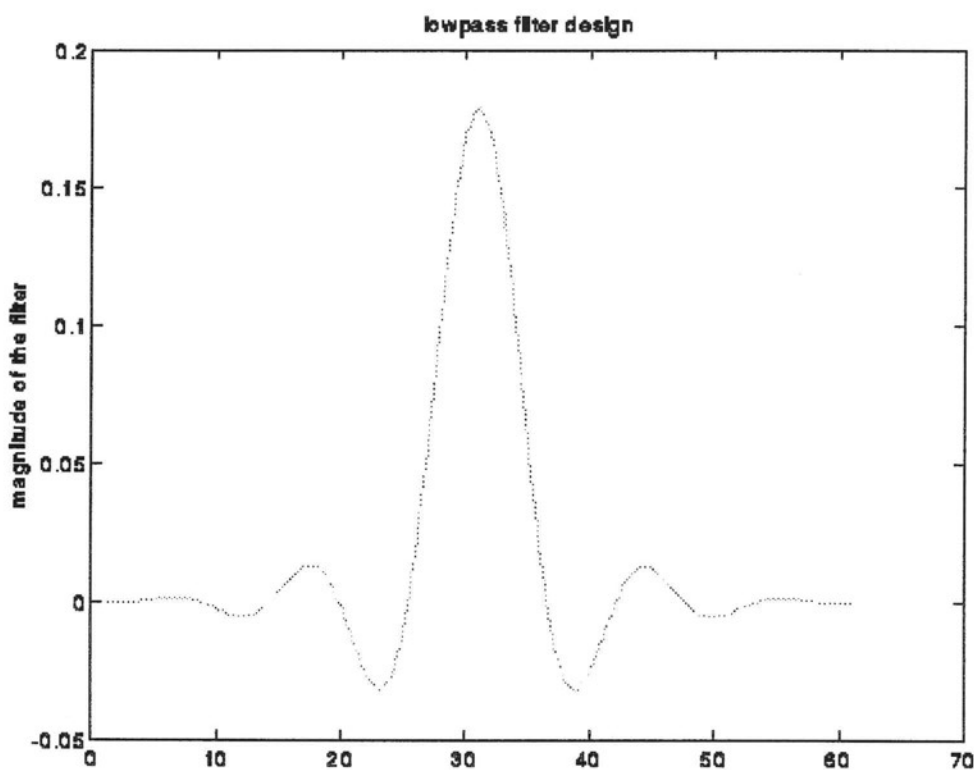
To acquire this signal the experiment was as setup in figure 8.5(d), the signal was acquired via #1 of the ADC and sent back via #2 of DAC. The display obtained was as shown in figure 8.5(e). The input signal was fed into #1 of the oscilloscope, the display in the upper part of the window and reconstructed was fed into #2 of the oscilloscope, depicted by the lower display. At this stage the action of the designed filter on the ECG signal was considered.

8.7 ***REAL-TIME TESTING AND FILTER RESPONSE***

To achieve real-time filtering, the same experimental set-up as in figure 8.2(a) was adopted, in addition the appropriate filter was incorporated to achieve the desired filtering operations. Furthermore data were output directly to the #1 of the DAC on the interface card, as before.

8.7.1 **Filter response**

The software designed for this purpose employed two basic methods, namely the window design and a mathematical algorithm, known as the *difference equation*. The window design was achieved using standard equations defined for the several windows, to compute the desired coefficients, and the program that does the computation was called "window.c." The lowpass filter was designed using this method, (see sec 4.1.1).

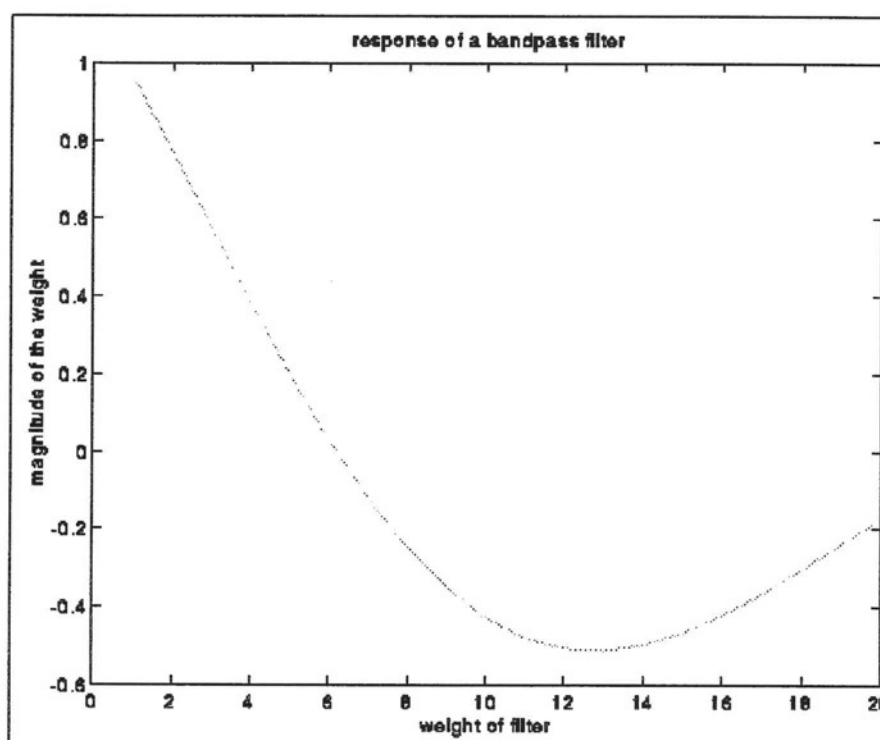


(Fig 8.6)
illustrating lowpass filter response

8.7.1.1 Lowpass Filter

Running this program several parameters were supplied, such as the sampling frequency, the cut-off frequency, type of window etc., (see chapter 7, sec 7.3). If Kaiser window was selected as in this case, the program request for the value of β , the ripple parameter, and the length of the filter, N . With these parameters supplied according to the specifications, the figure 8.6 illustrates the response of the desired filter.

The coefficient of this filter was saved under the filename "*winlowps.dat*." From the response, the oscillation about the weight axis can be seen to be finite as opposed to infinity for an ideal response, hence the term window method.



(Fig 8.7)
illustrating bandpass filter response

8.7.1.2 Bandpass filter

The design of the filter have been described in chapter 4.4.2, using the BZT. Also derived was the difference equation showing how the coefficients were computed and figure 8.7 depicts the response of the filter for a step input. The coefficient of this filter was saved under the filename "*bandps1.dat*." From the diagram the minimum point illustrate the region where the signal of interest was passed by the filter.

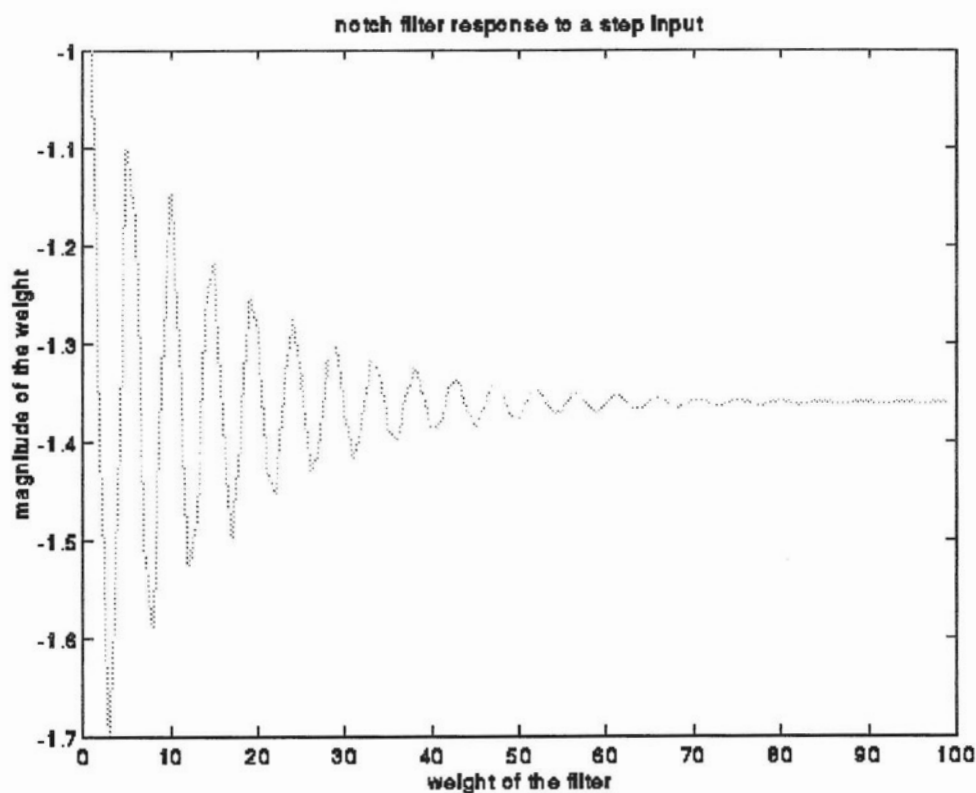
8.7.1.3 Notch filter

Like the bandpass filter, the design of this filter was also based on difference equation, (see sec 4.4.3), performing computation in a similar manner as with the implementation of the passband filter. The illustration of the response was as shown in figure 8.8(a).

From the response of the plot, the notch of the filter can be seen to extend down to the weight axis, this point corresponds roughly to the frequency to be eliminated, 50Hz. In addition to this plot, the plot of the magnitude and phase was considered, and these are illustrated in figures 8.8(b) and 8.8(c) respectively.

The frequency axis was normalised, and fifty points were considered for a period of 2π radians.

From the magnitude and phase plot, it can be seen that a maximum and a minimum response was attained at the notch frequency respectively. The coefficient of this filter was saved in a file called "testntch.dat."

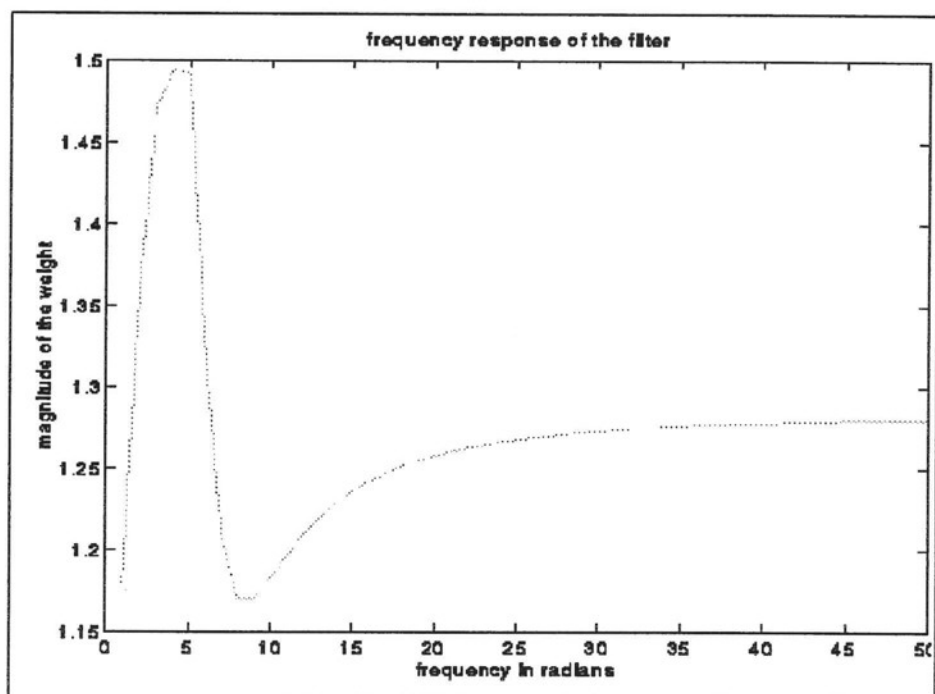


(fig 8.8(a))
illustration of the step response of the notch filter

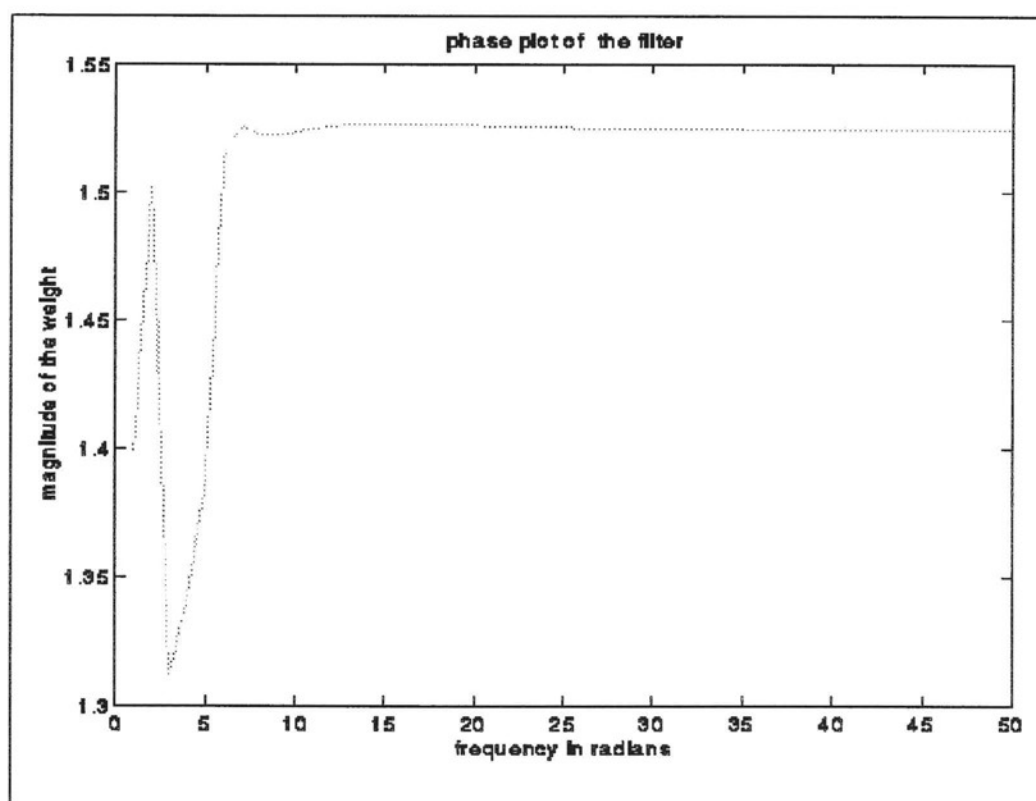
The coefficients of the magnitude and phase plot are stored in "testnmag.dat" and "testnphs.dat" respectively.

8.7.2 DIGITAL FILTERING

The digital filtering proceeded by considering the effect of each of the filter on the acquired signals, such as the reference, low frequency, simulated physiological and ECG signals. Finally the filter or combination of filters that prove most effective in eliminating the unwanted signal, will be employed in the adaptive filtering implementation.



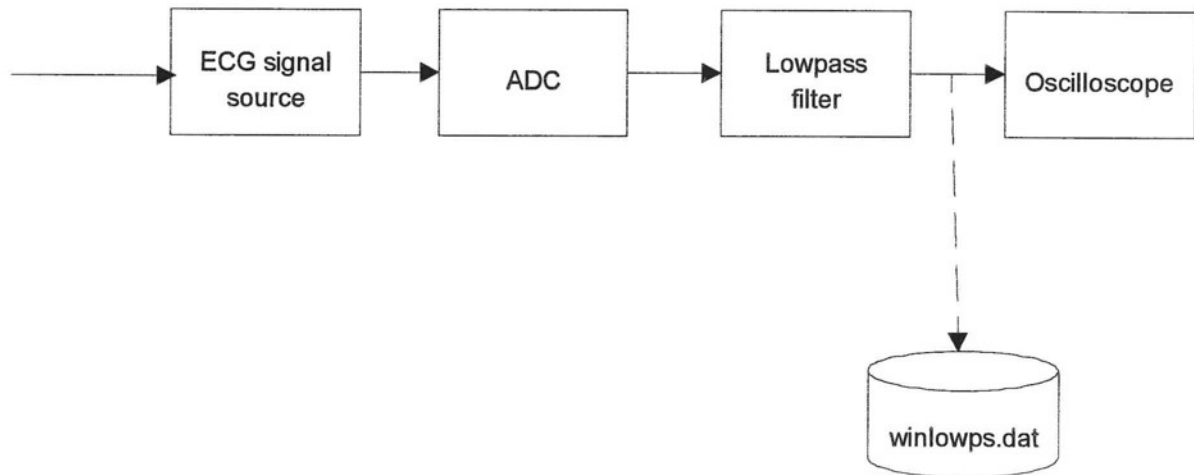
(Fig 8.8(b))
illustrating the magnitude of the Notch filter



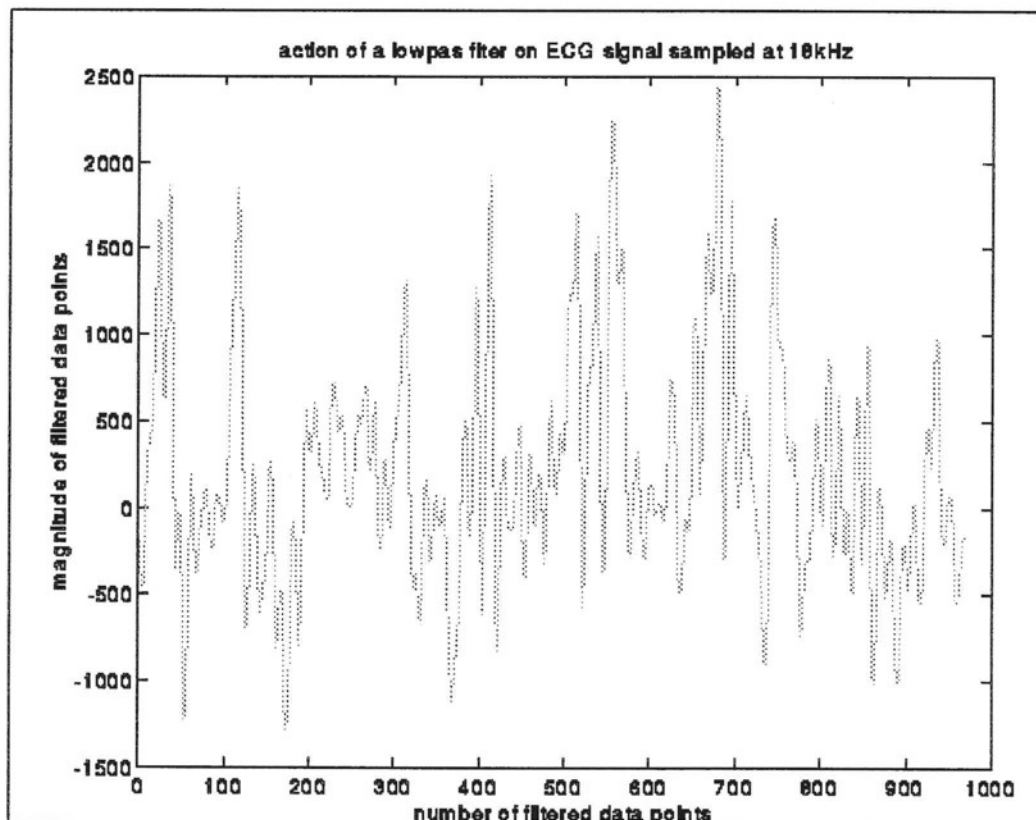
(Fig 8.8(c))
illustrating the phase response of the notch filter

8.7.2.1 Lowpass filtering

This experiment began by designing a software called "*firfilt.c*." This program works by prompting the user to supply the name of the file to be filtered and the filter coefficient file. When these files were supplied, (including the "*winlowps.dat*"), then the filtering operation begins, and the duration however, depends on the length of the filter's coefficient. In

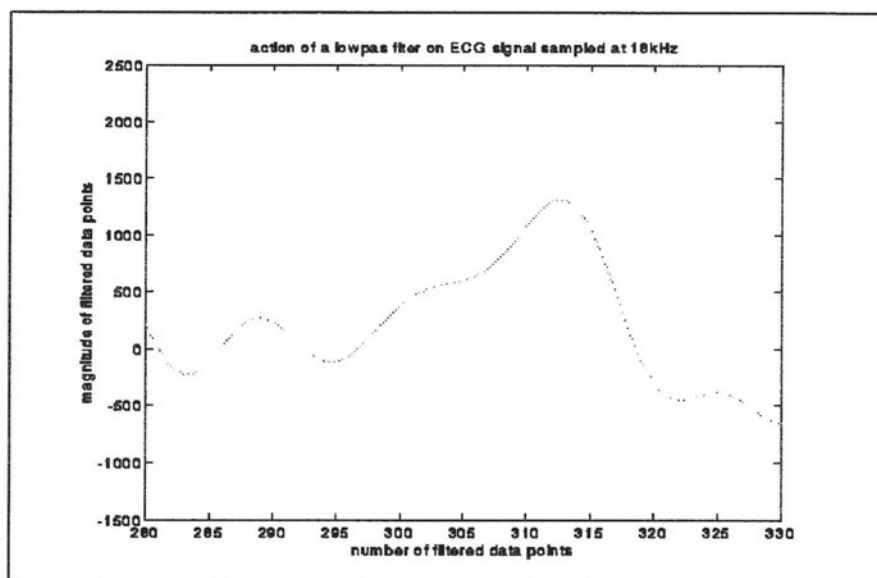


(Fig 8.9(a))
illustration of the experimental set-up for lowpass filtering



(fig 8.9(b))
illustrating action of the lowpass filter on the acquired ECG signal

general the duration of the filtering process is proportional to the filter length. The figure 8.9(a) shows the experimental set-up for the lowpass filtering procedure.

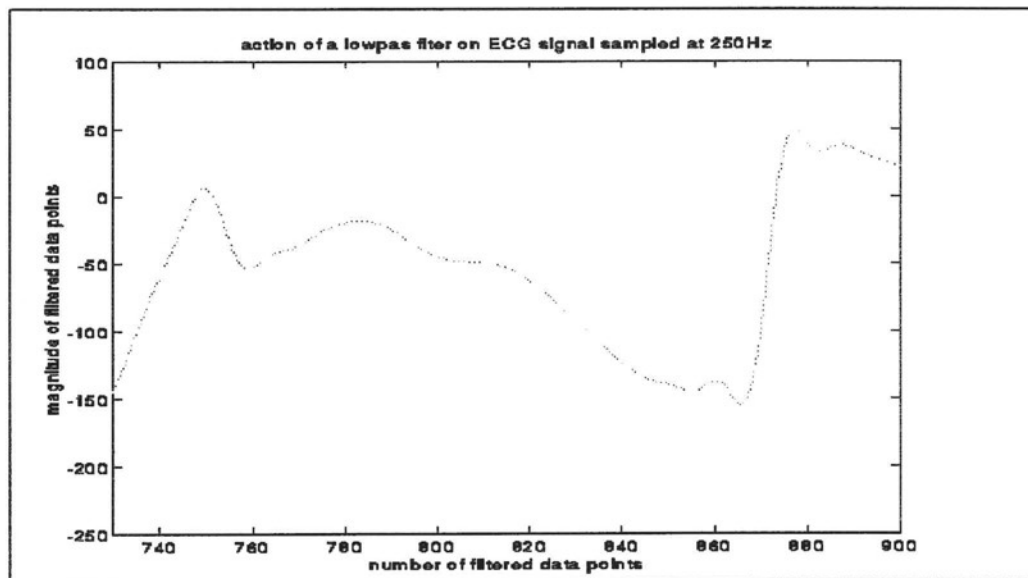


(fig 8.9(c))
illustration of magnified ECG signal sampled at 18kHz

The first case considered was the reduction of the noise in the ECG signals sampled at 18kHz and 250Hz respectively, followed by the action of the filter on other acquired signals. The results obtained are shown in figure 8.9(b) through 8.9(d) respectively.

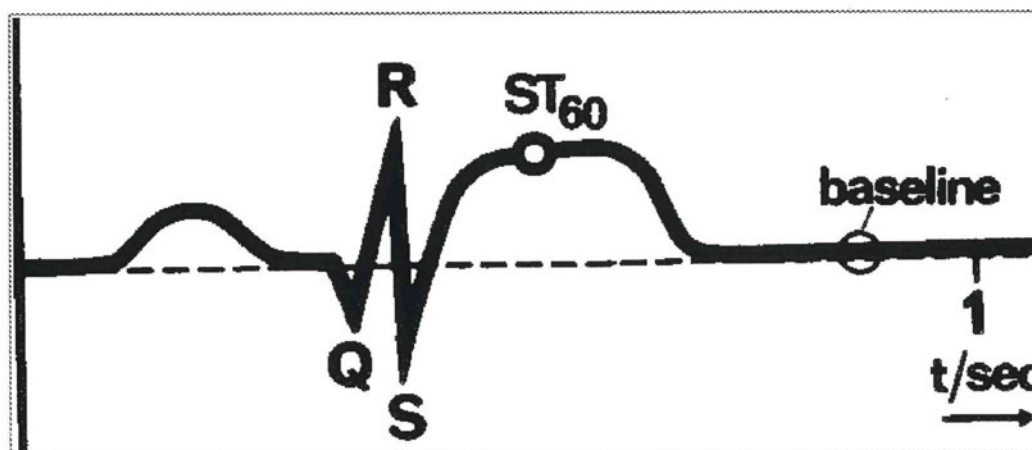
The filtered data was further magnified by considering about 100 data points as shown in figure 8.9(c) and about 200 data points in figure 8.9(d), for the signal acquired at 250Hz. From left to right of the figure 8.9(c), 3 peaks are visible with varying magnitudes. The highest peak in this figure is usually referred to as the **R-peak** and the interval between such peaks are referred to as the **RR interval**, see figure 8.9(e). Although R-peak chosen in this instance seems to be broad as opposed to the usual sharp peak. This can be observed in figure 8.9(d). To the left of figure 8.9(b) is a peak called the **T-peak**, and to the right of the first peak, figure 8.9(d) is the peak described as the **P-peak**, and this was better presented by the action of the bandpass filter (see figure 8.11(a)), on the ECG signal. If a line can be drawn such that the different peaks and valley, (ie the R,Q and S peaks), lies on either side of this line, such line is described as the **baseline** and measurement from the baseline to the to any of the peaks or valley is a measure of the amplitude followed by the name of the peak, eg T-peak amplitude, P-peak amplitude etc.

The figure 8.9(c) and 8.9(d) can be seen to reveals identical information as expected although the quality of the filtered data differs.



(fig 8.9(d))
illustrating about 200 filtered data points of a ECG signal sampled at 250Hz

As explained earlier, from figure 8.9(d), the first peak from the left corresponds to the R-peak in figure 8.9(e). The T-peak in figure 8.9(e) seems to be slightly exaggerated, (see figure 8.11(c)).



(Fig 8.9(e))
illustrating several peaks observed in ECG signal, [21]

The filtering of the simulated physiological signal and the reference signal was insignificant using the lowpas filter. This result was as expected, since the cut-off frequency of the filter was in the region of 20Hz, allowing for tolerance, (see figure 8.9(f)).

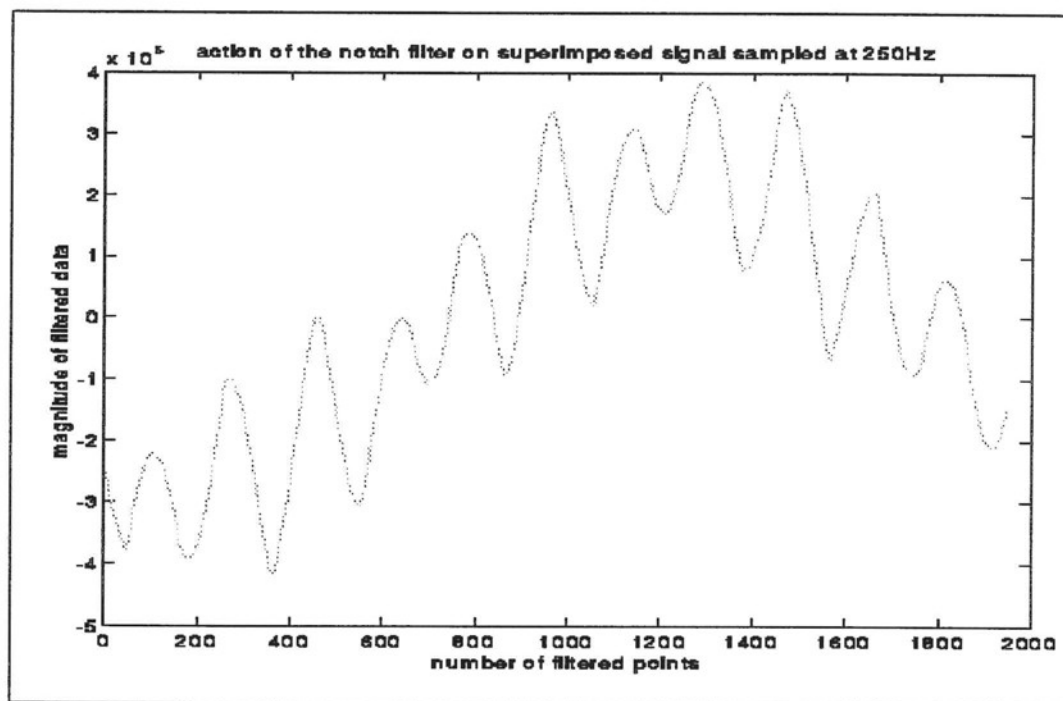
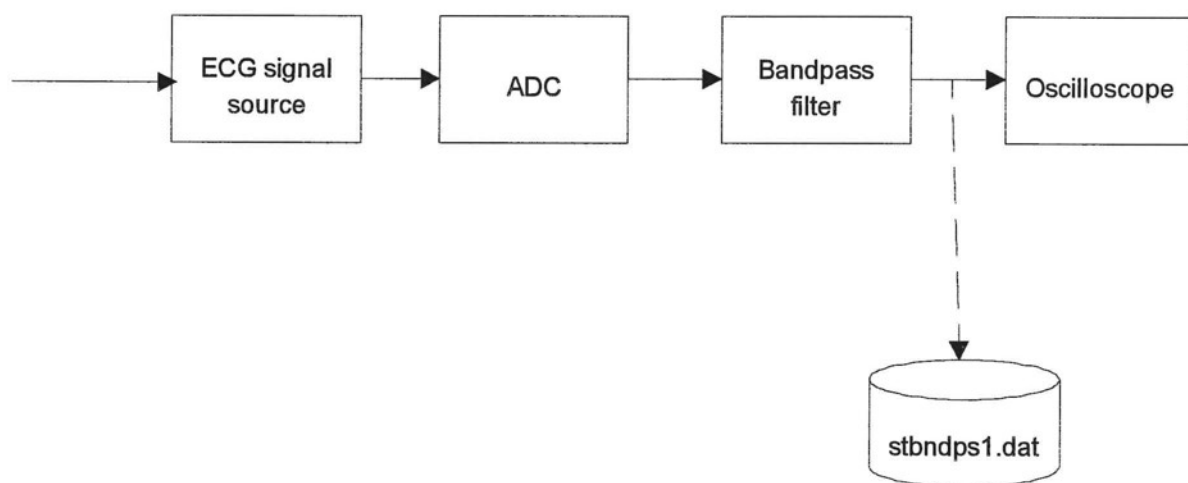


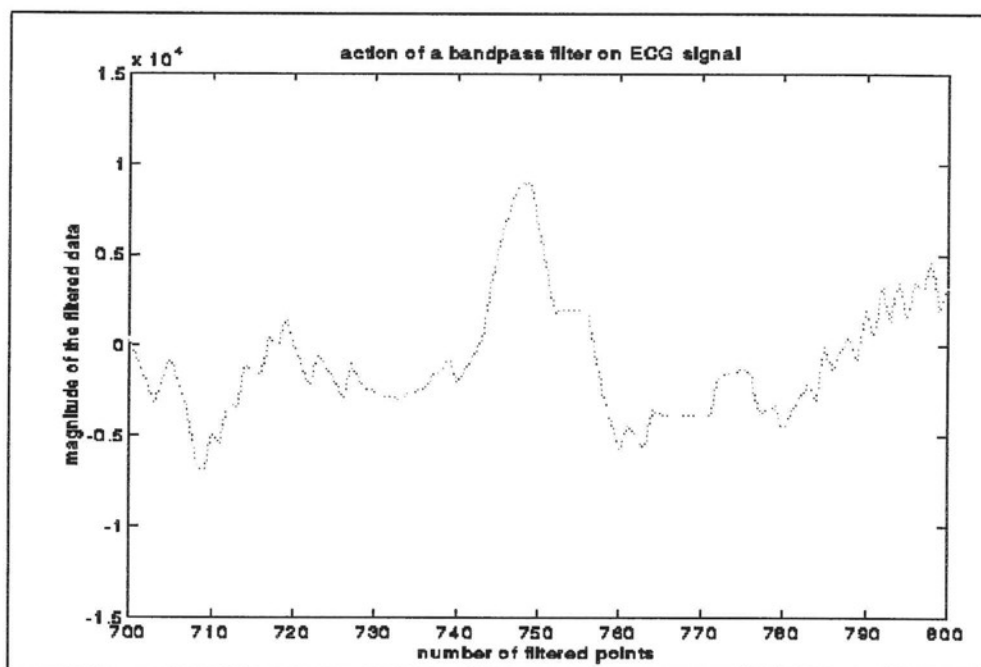
Fig 8.9(f)
illustrating the action of the lowpass filter on the simulated physiological signal

8.7.2.2 Bandpass filtering



(Fig 8.10)
illustration of the experimental set-up for bandpass filtering

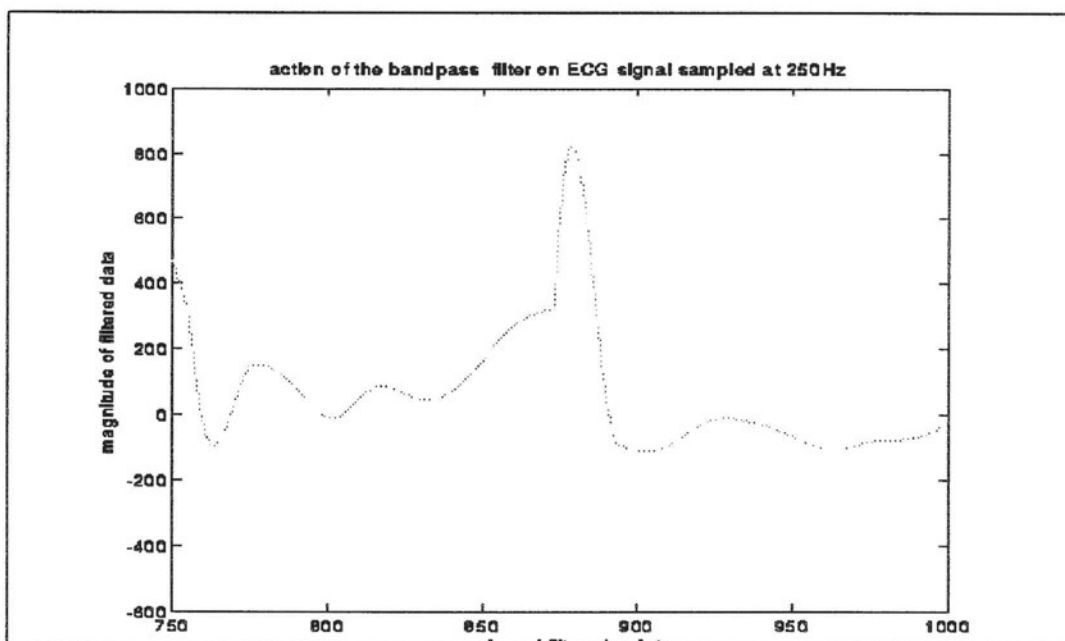
Similarly like the lowpass experiment, the 18kHz signal was first considered, and the acquired signal was depicted in figure 8.11(a), giving consideration to 100 samples of the filtered data.



(fig 8.11(a))

illustrating the action of the bandpass filter on a ECG signal acquired at 18kHz

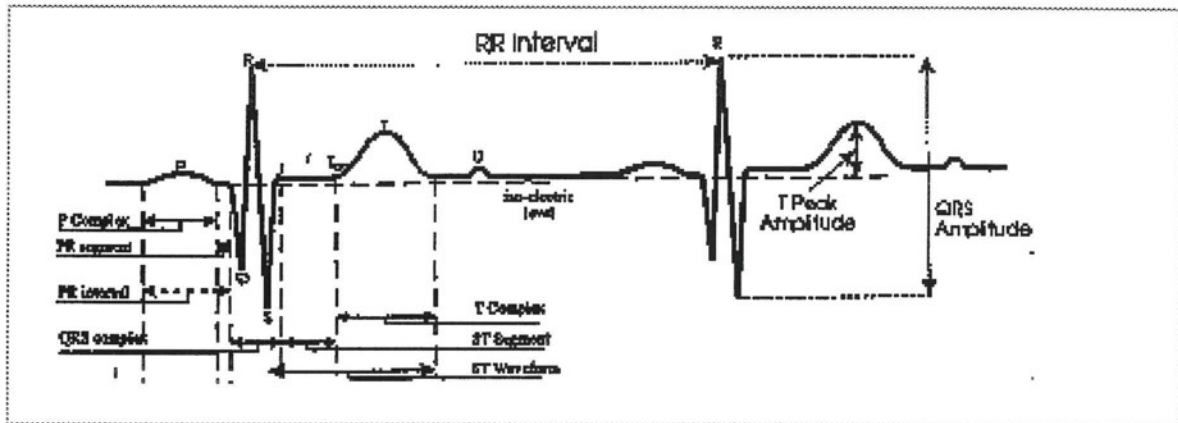
The information present in figure 8.11(a) was identical to those presented in fig 8.9(c) and 8.9(d) respectively. Furthermore the features of the ECG signal are distinct employing the bandpass filter. However, the presence of the sinusoid and noise are quite visible but have been reduced considerably, (see fig 8.5(b)). From this figure the highest peak is the R-peak as described earlier, and the peak immediately to the right projecting from the valley is identically the T-peak previously identified, see figure 8.11(c).



(fig 8.11(b))

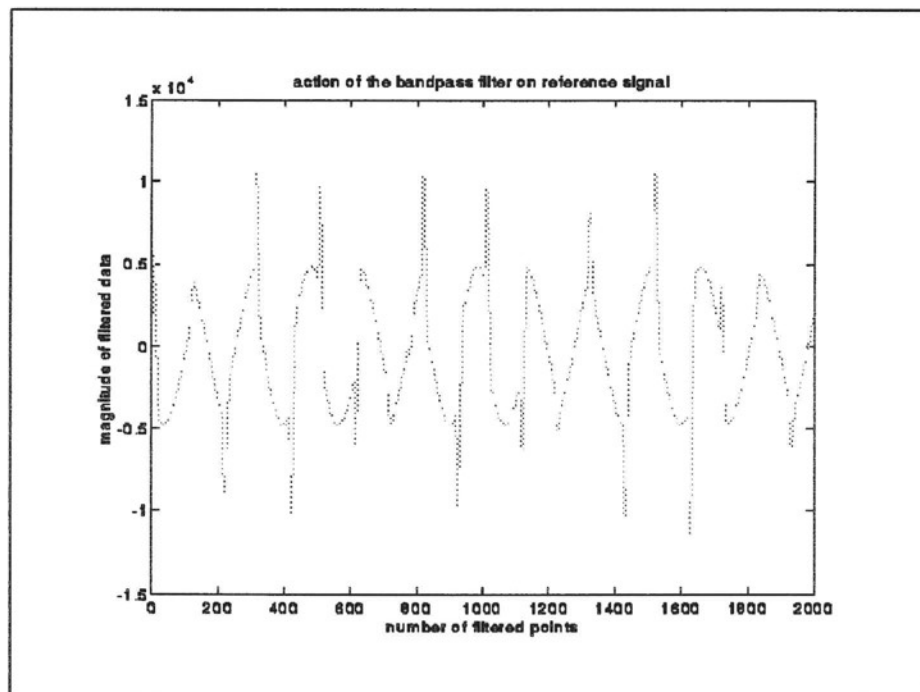
illustrating the action of the bandpass filter on the ECG signal sampled at 250Hz

This phenomenon whereby the ECG profile is superimposed on a sinusoid is described as the baseline shift. Method of interpolation have been suggested to eliminate this undesirable effect, [22].

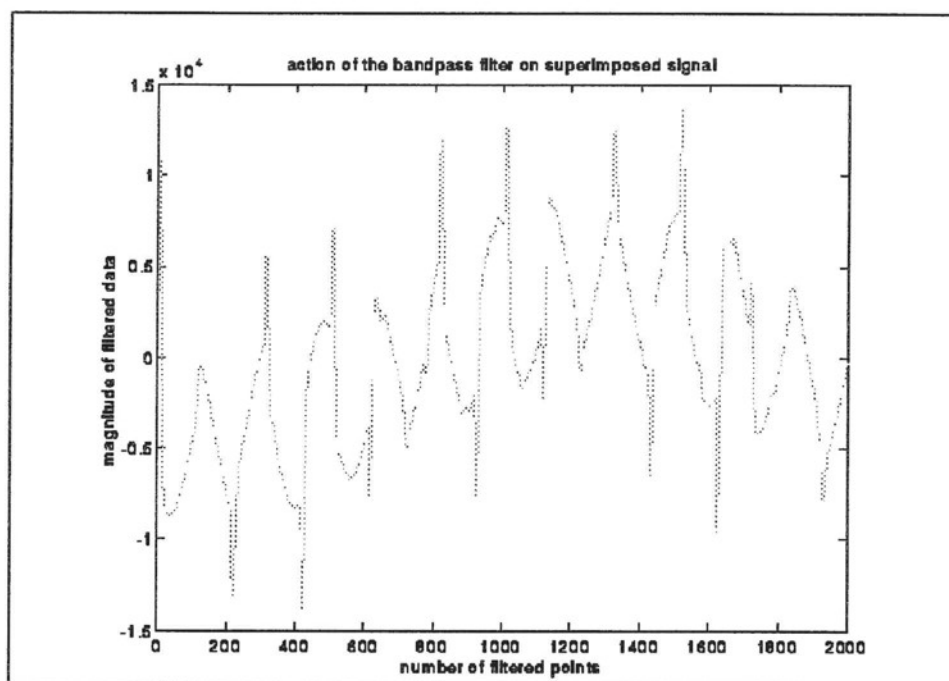


(Fig 8.11(c))
illustration of P, R and T peaks in a ECG signal, [22].

The result shows that the filter was able to get rid of part of the unwanted signal, (see figure 8.4(b) prior to filtering), but have not eliminate the presence of this signal completely, (see figure 8.12(a) and 8.12(b)). This result was as expected, because the filter was designed to pass certain frequency signal strongly while unwanted signals are passed weakly.



(fig 8.12(a))
illustrating the action of the bandpass filter on the reference signal

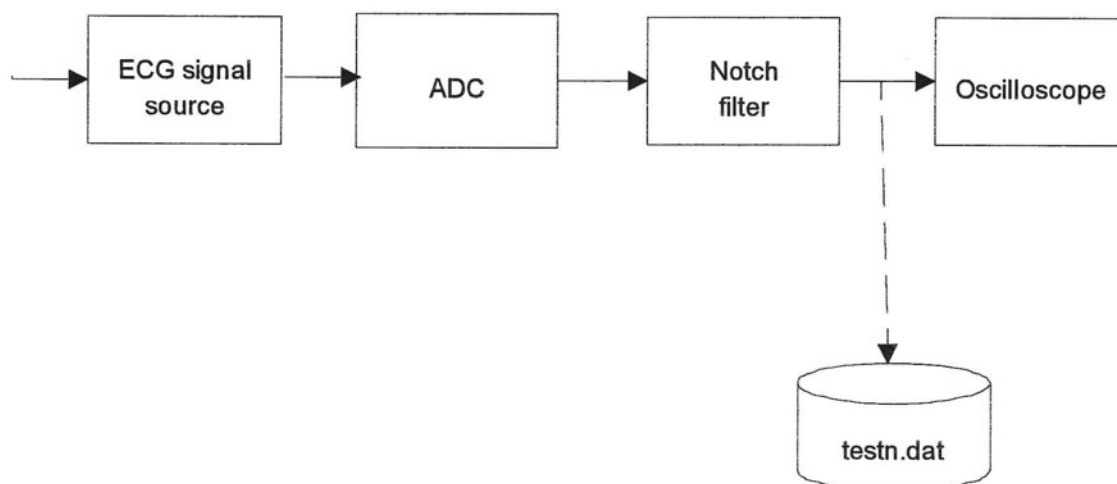


(Fig 8.12(b))

illustrating the action of the bandpass filter on a simulated physiological signal

8.7.2.3 Notch filtering

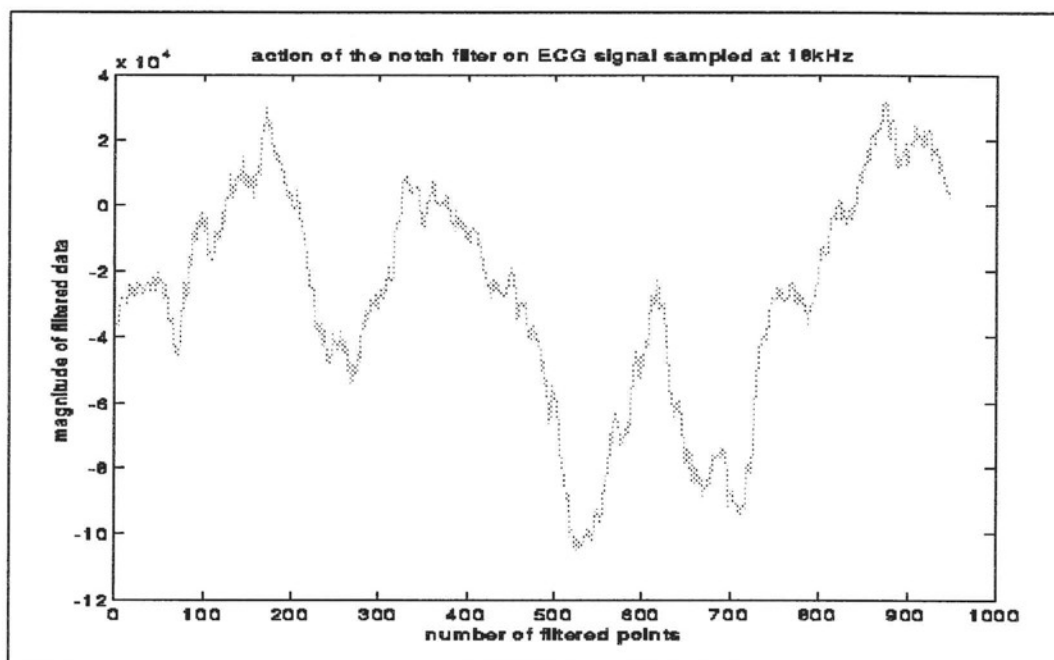
Like with other filters, the notch filtering began by examining the acquired ECG signals and the physiological signal. The experimental set-up was depicted in figure 8.16.



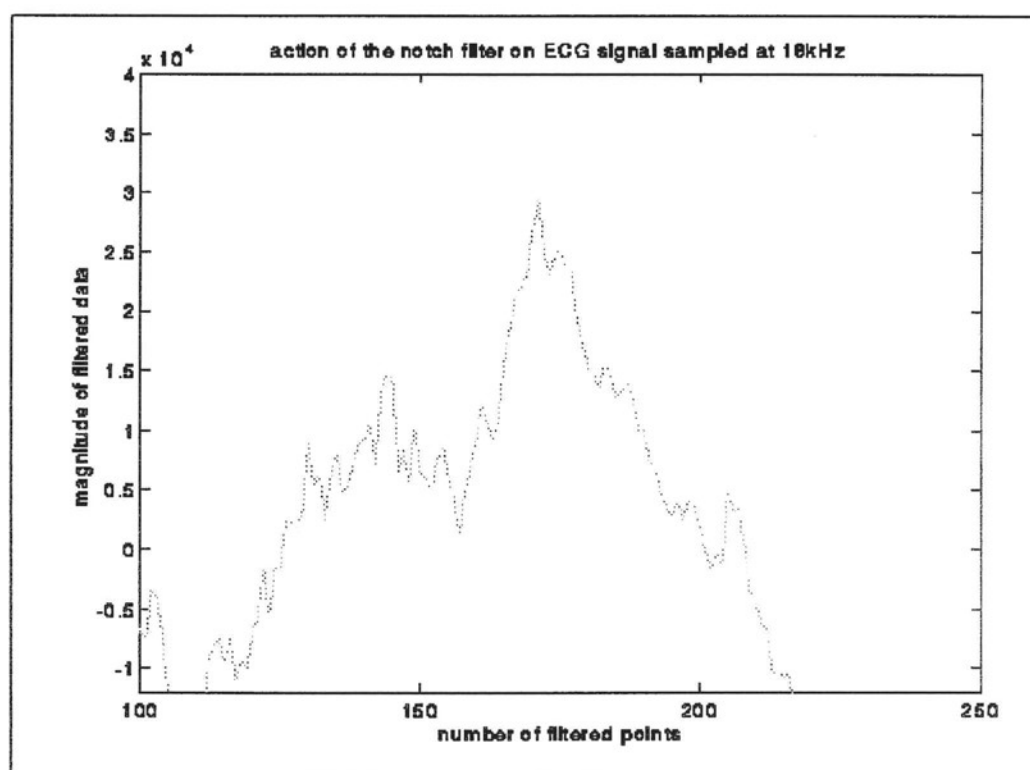
(fig 8.13)

illustrating the experimental set-up for notch filtering signal

After the filtering process, figure 8.14(a) can now be compare with figure 8.5(b), the filtered data. The traces of noise has been reduced considerably although have not been eliminated. However, the signal of interest appears cluttered and therefore needs to be magnify and consequently a sample of 150 data points were considered and expanded as shown in figure 8.14(b) and 8.14(c) respectively.



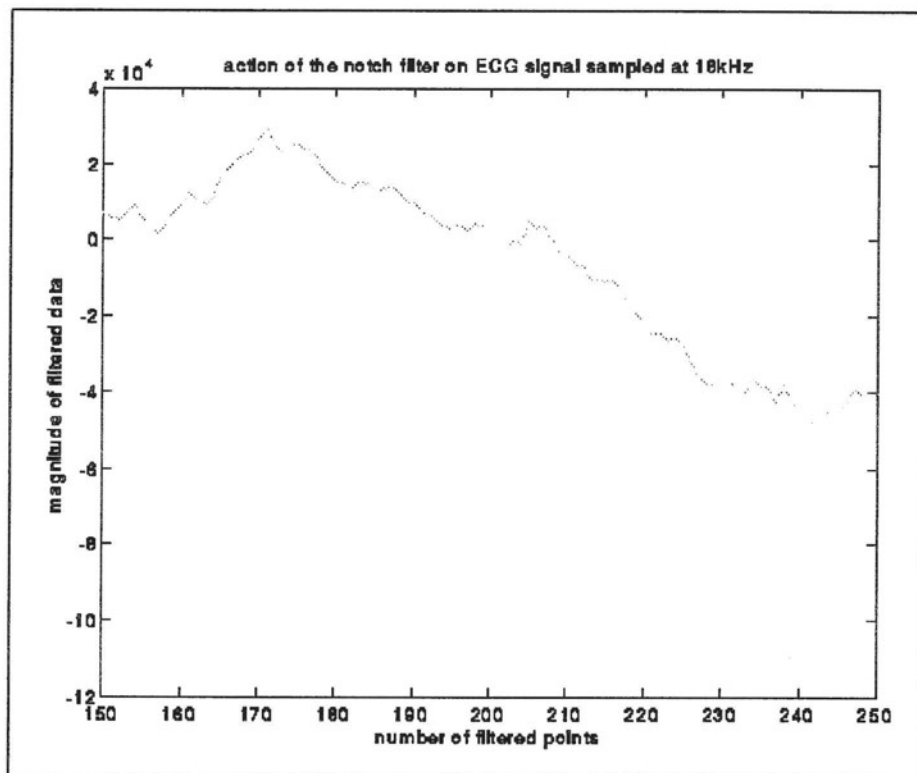
(fig 8.14(a))
illustrating the action of the notch filter on the ECG signal sampled at 18kHz



(fig 8.14(b))
illustrating the action of the notch filter on the ECG signal

It can be seen that the R and Q-peaks presented by notch filtering was shaper and distinct compare to those of the lowpass and bandpass filters respectively. However, the desired P and T peaks are not visible in this instance. A reduced samples of 100 points was further considered and the desired P and T-peaks were found to assume different magnitude within the filtered data.

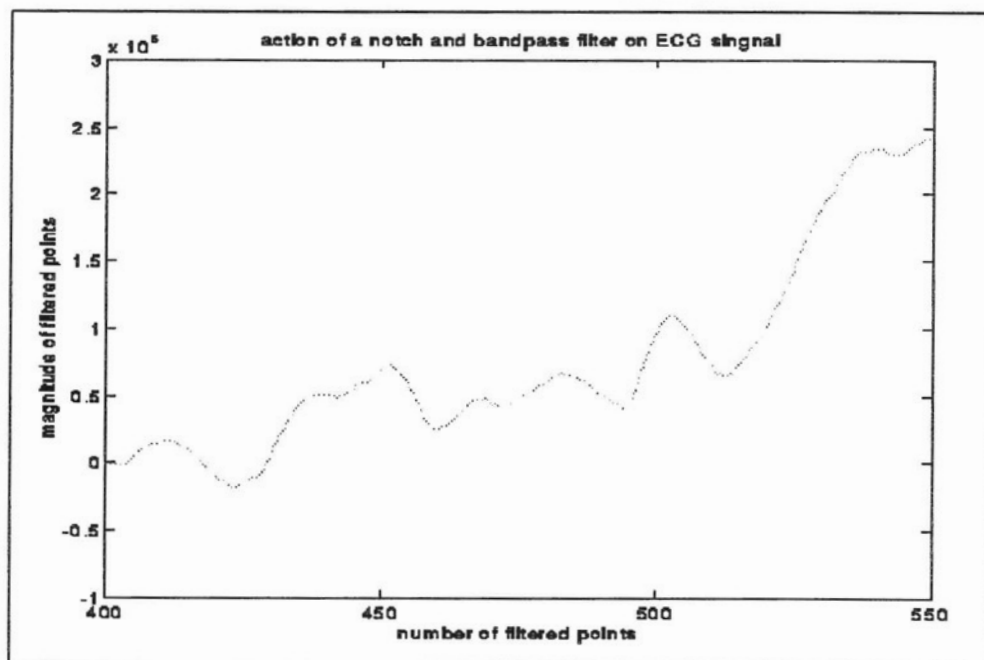
Investigation of the effectiveness of this filter continued by examining the ECG signal sampled at 250Hz, and this was shown in figure 8.14(c).



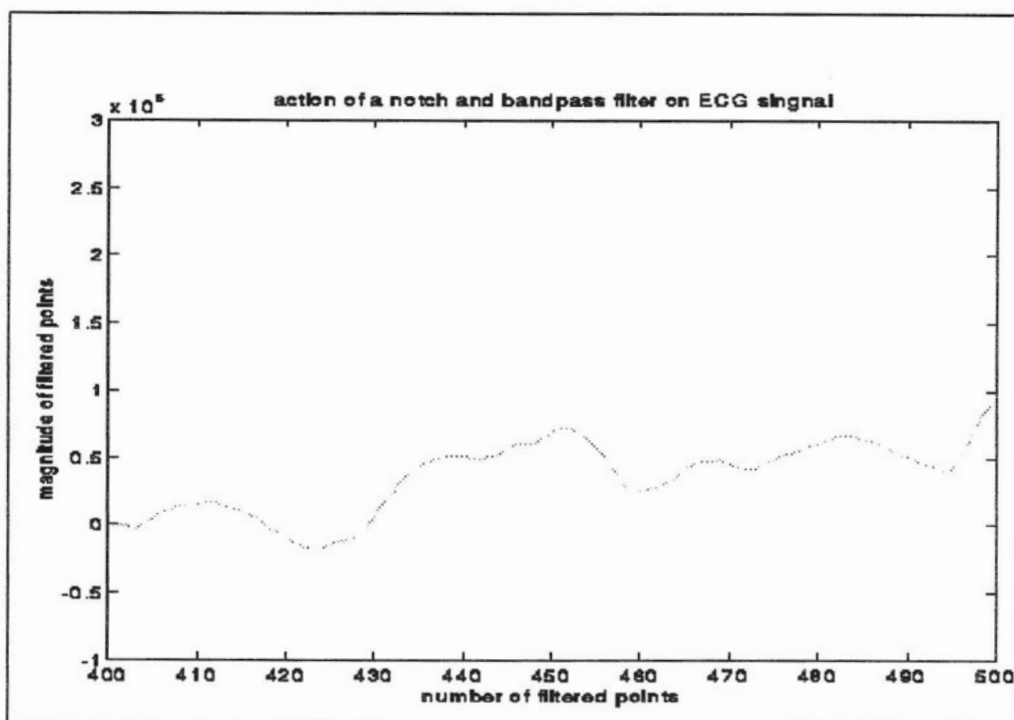
(fig 8.14(c))
illustrating the action of the notch filter on the ECG signal acquired at 250Hz

Action of the notch filter was also considered on the simulated physiological signal and the result obtained was shown in figure 8.14(d). It was found that part of the signal was notched although other frequency was allowed to pass as would be expected, (the filter was designed for 50Hz).

After all these experiment, the question arises, are there any combination of the filters such that when combined during the filtering process will produce optimum value of the desired signal. If this were the case, therefore such combination of filters will be ideal and consequently be employed in the adaptive filtering process. As a result of this, further investigation continued by examining action of a bandpass filter prior to notching and lowpass filtering followed by bandpass filtering and appropriate comparison was made. Figures 8.15 began by examining the ECG signal sampled at 18kHz followed by the ECG signal acquired at 250Hz.



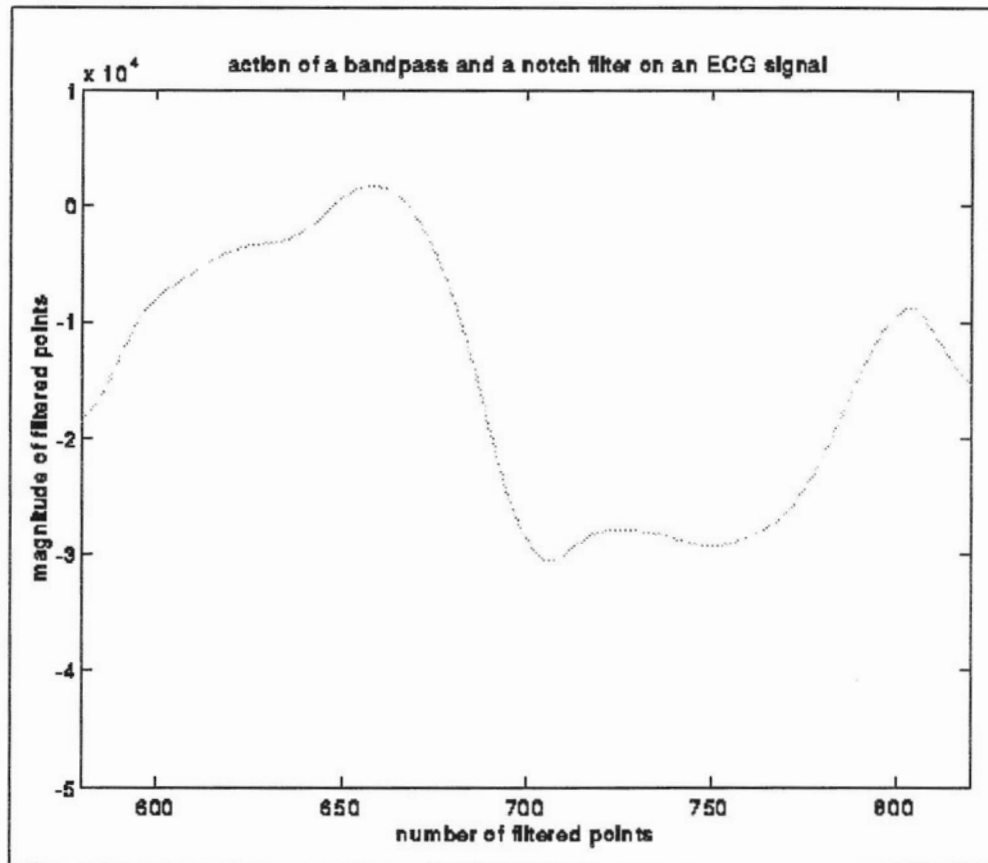
(fig 8.15(a))
 illustrating the action of a notch and
 bandpass filter on the ECG signal acquired at 18kHz



(fig 8.15(b))
 illustrating the action of a notch and
 bandpass filter on the ECG signal acquired at 18kHz

The figure 8.15(a) depicts the action of combined effects of two filters, namely the notch and the passband filter on the ECG signal sampled at 18kHz. This procedure began by considering about 150 filtered sample point. And figure 8.15(b) expand this figure further by taking a reduced sample of 100 filtered points.

Similarly the combined effect of the lowpass filter and the bandpass filter was also considered, figure 8.15(c) for the ECG signal sampled at 250Hz.



(fig 8.15(c))
*illustrating the action of a lowpass and
bandpass filter on the ECG signal acquired at 250Hz*

8.8 INVESTIGATION OF THE ADAPTIVE PARAMETER, (μ)

Several filtering process have been carried out in order to investigate the performance and action of the filters on the acquired signals. The results obtained have given a clear insight to their operation.

The next and final part of this task was to employ these filters as part of a complete system, in this case, an adaptive filter. It has been mentioned earlier the need for an adaptive filter, (see chapter 2), several methods available, chapter 3 and how it can be achieved, (chapter 5 and chapter 6).

However, the algorithm in chapter 5 incorporates the parameter μ , which signifies the iterative steps in the adaptive algorithm. The question at hand is what initial value should be given to μ ? Should it be its minimum value, maximum value or perhaps the mid range between these values. If so, what are these values?

To answer this question, it is appropriate at this stage to mention that the algorithm developed would be dedicated to monitoring the ECG signal at a given instance. As a consequence the nature of the signal is known, (though a low frequency signal), and so are the actual values. Supposing that some statistics of these signals are available, then it may be possible to assign a value for μ at random employing knowledge of this statistics. This is the line of reasoning that would be followed and employed for the latter part of this task.

8.8.1 Determination of the covariance matrix for the ECG signal

From properties of random signals, the *probability density function* (P.D.F), is the probability that a random vector $\underline{x}(t)$ takes a particular value. If $f(\xi)$ represent the probability that $\underline{x}(t)$ takes on a particular value within a band $d\xi$ centred at ξ , then the Gaussian probability distribution is such that:

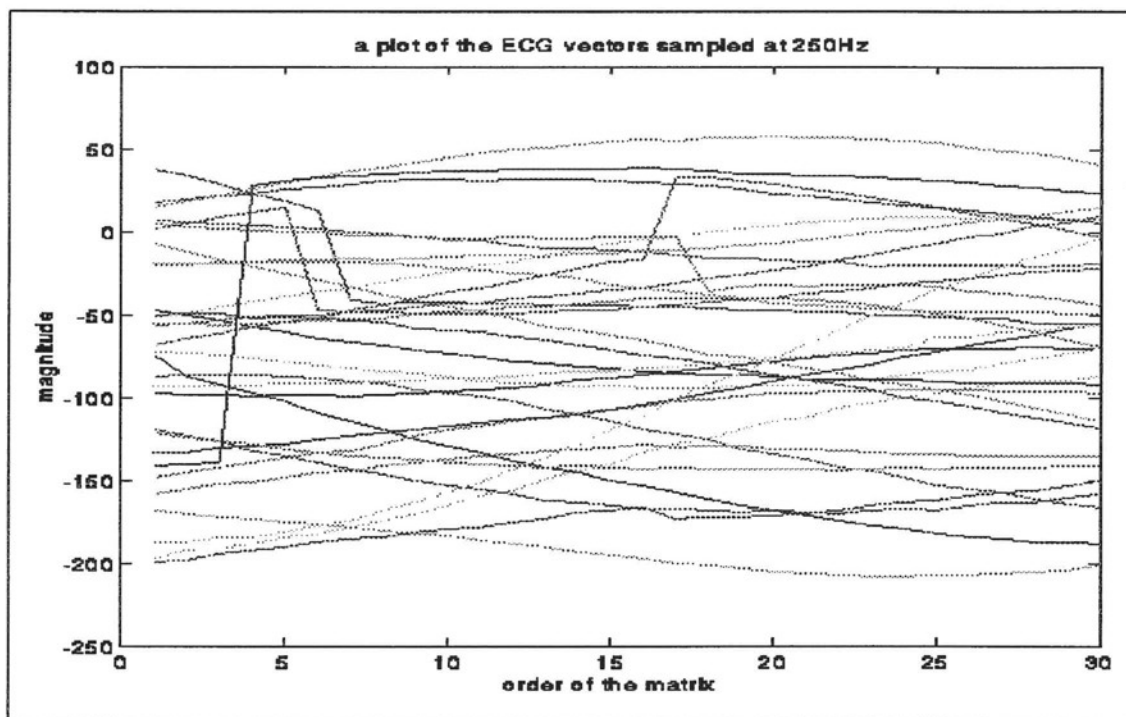
$$f(\xi) = \frac{1}{\sqrt{2\pi P}} e^{-\frac{(\xi - \bar{x})^2}{2P}} \dots\dots\dots (8.1)$$

if $\underline{x}(t)$ is an $m \times 1$ vector then P is an $m \times m$ matrix. P is called the *covariance matrix* or the *spectra density matrix*. If the probability density function does not vary with time then the signal is said to be stationary. This notion was assumed in this particular task.

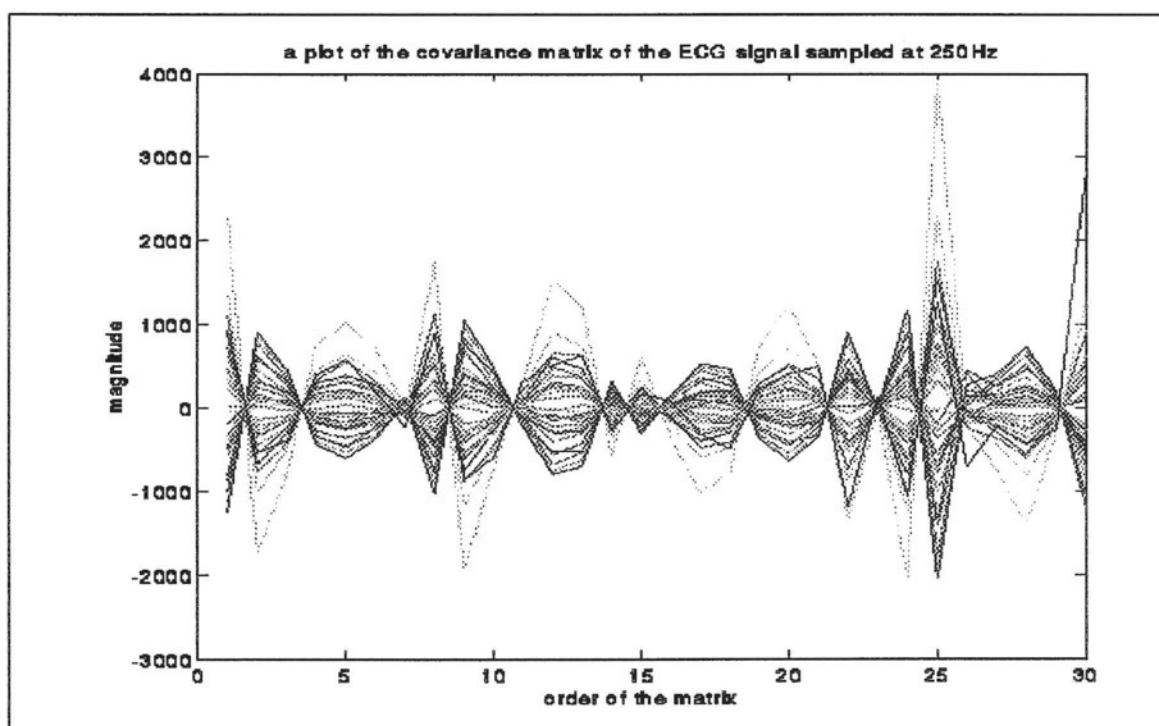
In order to determine the covariance matrix, P , the acquired ECG signal sampled at 250Hz was employed. However, the sample acquired at 18kHz can equally be used.

The sampled data was loaded into a specialised package designed for mathematical analysis.

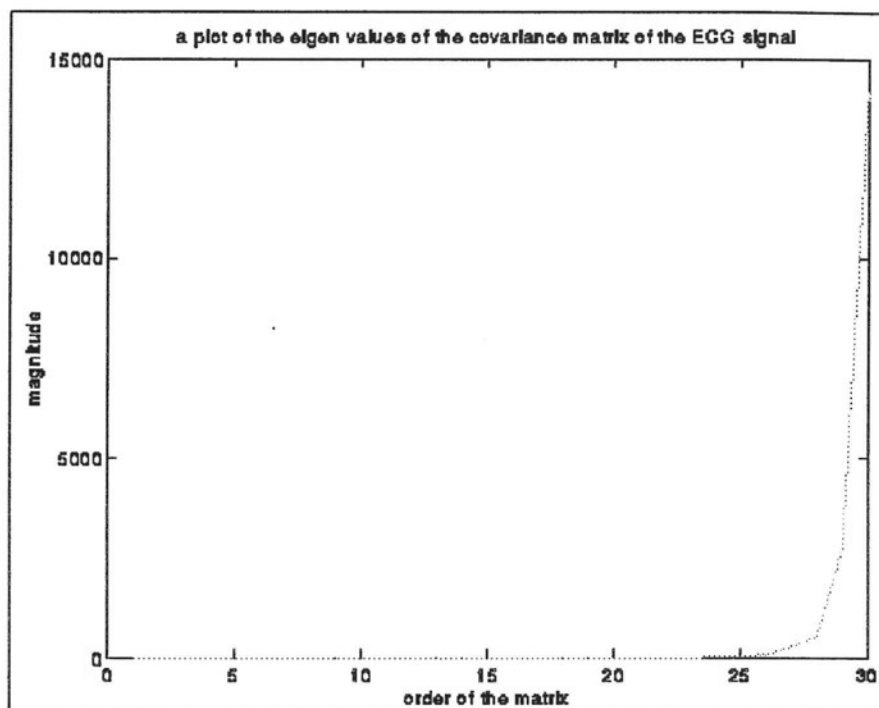
The datafile called "heart3.dat" was loaded in Matlab, a mathematical package, and the 1×1000 column vector originally stored on file was formulated into 30×30 matrix, by making the first 30 elements of the file into a column vector. The figure 8.16(a) depicts the plot of this vectors, [23]. It is however desirable to find the covariance matrix and this was achieved by using a special command in Matlab which generated the desired matrix. This matrix was depicted in figure 8.15(b). However, the diagonal elements of this covariance matrix signifies the range of convergence values for μ , (see equ 4.5.11). A plot of this was obtained and it was shown in figure 8.16(c).



(fig 8.16(a))
 illustration of the vectors of the ECG signal sampled at 250Hz



(fig 8.16(b))
 illustration of the covariance matrix of the ECG signal sampled at 250Hz



(fig 8.20(c))
*illustration of the eigenvalues
of the covariance matrix of the ECG signal sampled at 250Hz*

The expected value or mean value, $\bar{x}$ of $\underline{x}(t)$ is defined as

$$E(x(t)) = \int_{-\infty}^{\infty} \xi f(\xi) d\xi \quad \dots\dots\dots (8.2)$$

The covariance matrix P of $\underline{x}(t)$ is defined as

$$P = E \left\{ (x(t) - \bar{x})(x(t) - \bar{x})^T \right\} \quad \dots\dots\dots (8.3)$$

8.9 ADAPTIVE FILTERING

The adaptive filtering began by considering off-line filtering. This was achieved by adopting the algorithm proposed in chapter 5 and following the methodology of the logic in chapter 7. To proceed with this task requires that the input signals namely the reference and the corrupted signal need be feed into the digital filter to produce a new coefficient or weights which puts into account the prevailing environmental conditions such as noise. As a result of this several files were opened in this task, and they can be classified into input and output files. The input files consist of

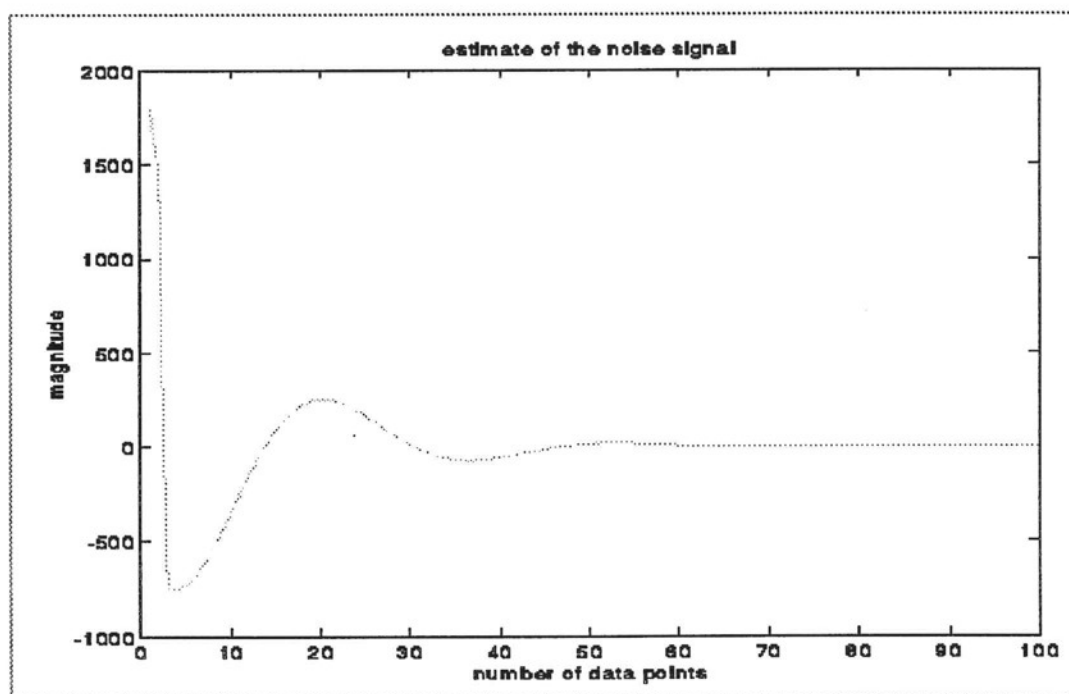
- a. The reference signal, used to emulate the noise of the mains frequency and serves as input for the digital filter.

- b. The corrupted signal signal, which contains the desired signal or data as previously discussed.
- c. The coefficient file supplied were the bandpass, lowpass and the notch filter coefficient respectively.

The output initially sought was to investigate how the error, estimate of the noise signal and the new weight or adapting coefficients of the filter, varies for different values of μ . The value of μ considered are 0.09, 0.0002 and 0.1 respectively. These have a corresponding value of λ equal 11.11, 5000 and 10 respectively, (see figure 8.16(c)). However, all the pictures acquired cannot be displayed. But the data are available on the programming diskette.

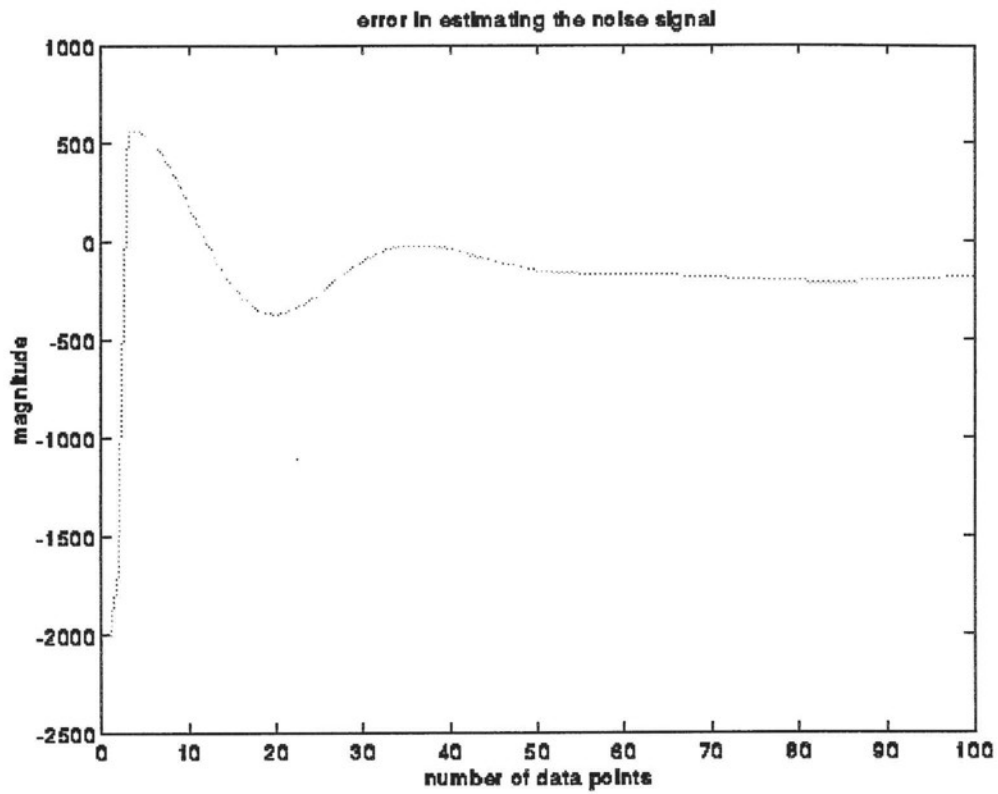
8.9.1 Bandpass adaptive filtering

The first case considered was the bandpass filter. The mentioned input file were supplied to the bandpass filter and the noise was estimated accordingly as well as the error, the result obtained are depicted below :



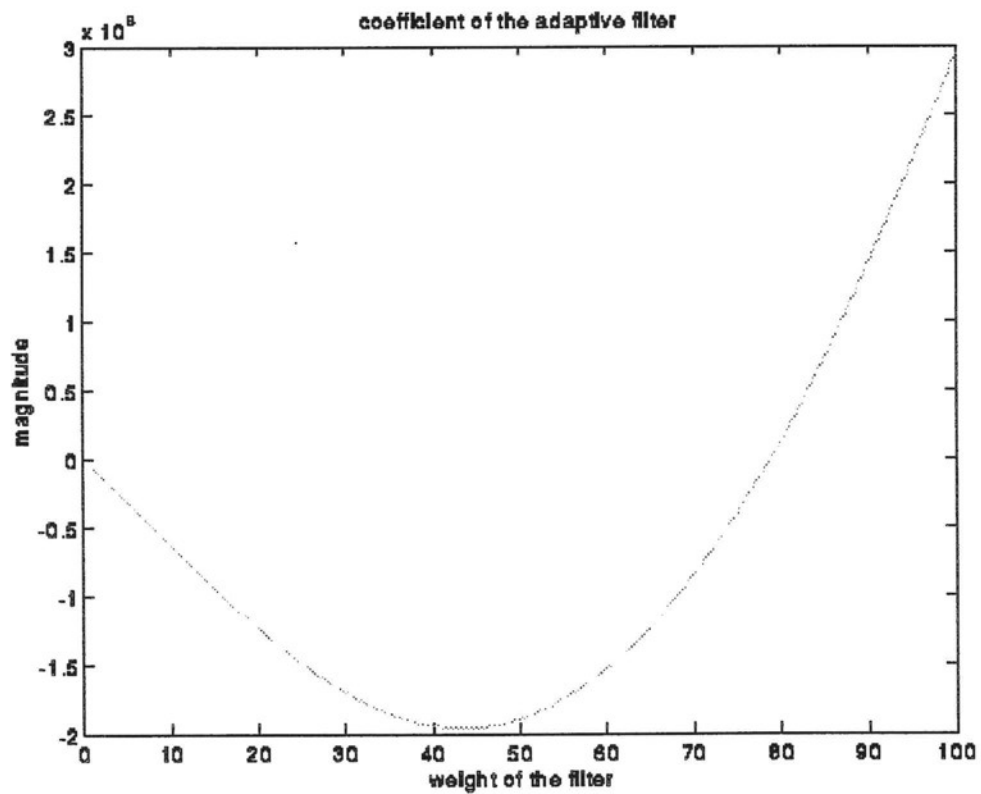
(Fig 8.17(a))
illustration of the estimate of the noise signal

Figure 8.17(a) illustrate the estimate of the noise signal in the corrupted data with μ equal 0.09. Figures 8.17(b) and 8.17(c) illustrate the error signal which was fed back to the digital filter to produce a corresponding new filter weight.



(Fig 8.17(b))

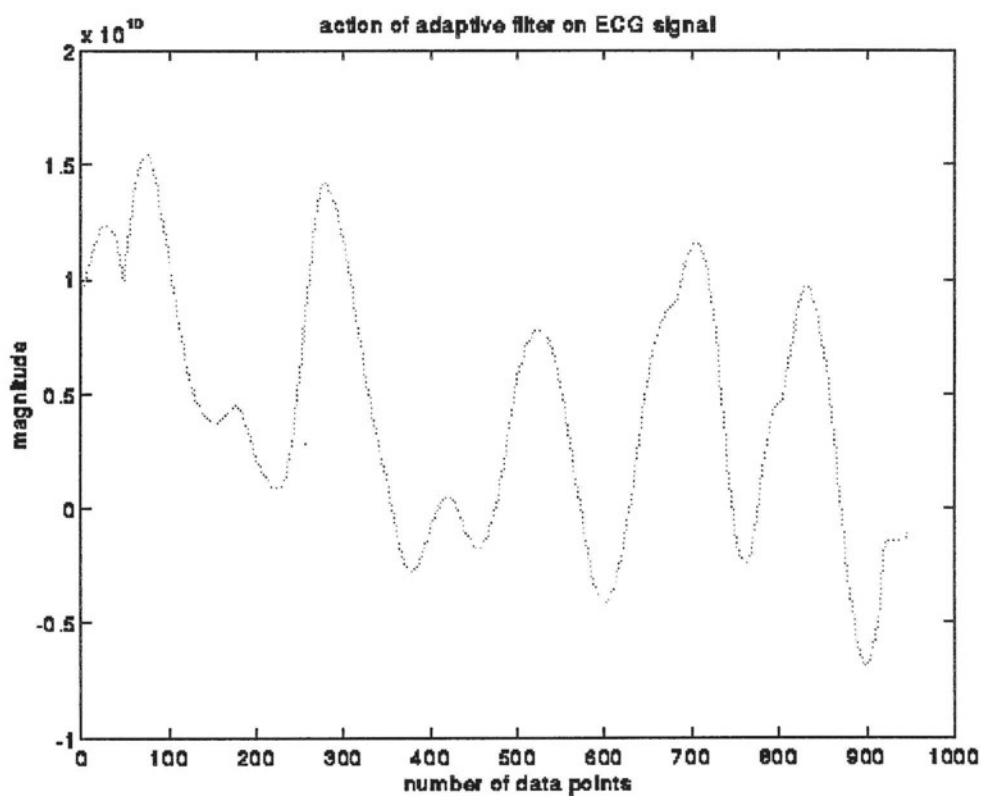
illustration of the error signal generated by the digital bandpass filter



(Fig 8.17(c))

illustration of the new weight generated by the filter

Having generated this datafile, they were then employed to filter the ECG signal by using the previous program "firfilt.c." The first case considered was the signal acquired at 250Hz.

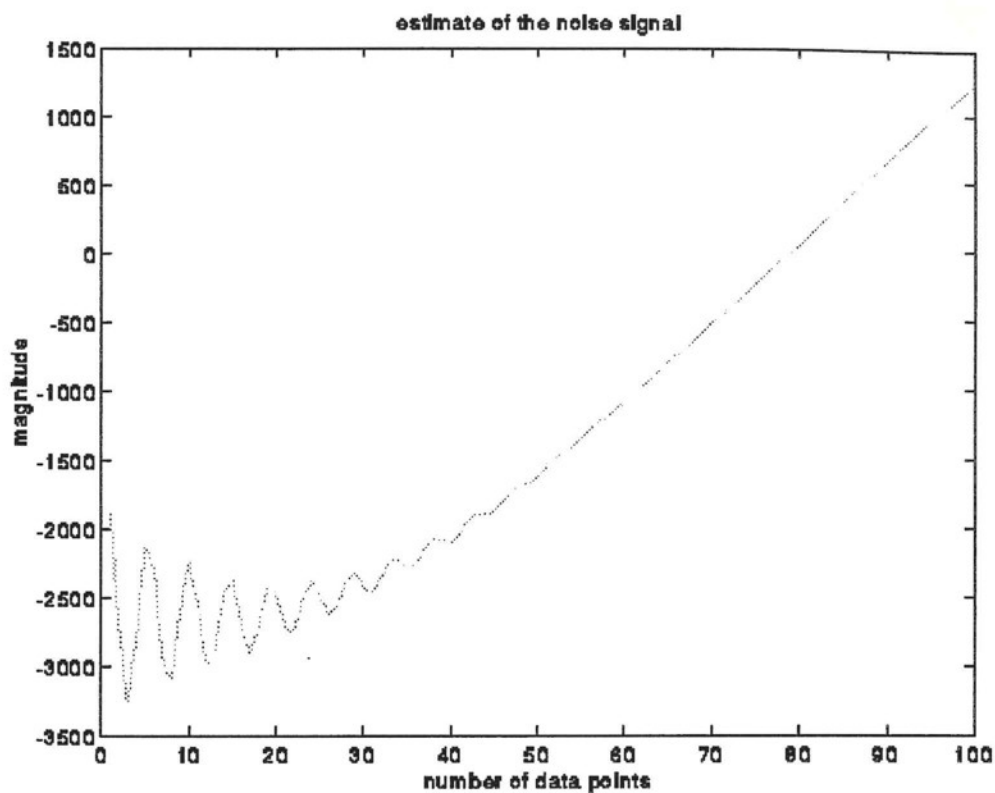


(Fig 8.17(d))

illustration of the action of the bandpass filter on an ECG signal sampled at 250Hz

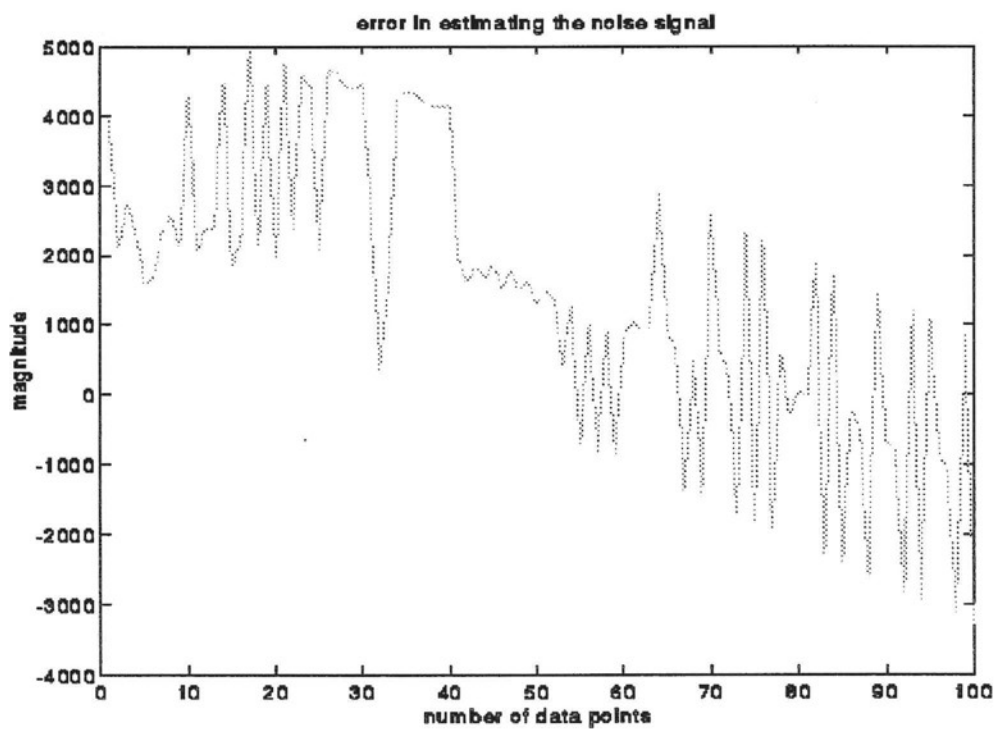
8.9.2 Notch adaptive filtering

By similar consideration figures 8.17(e) through 8.17(h) depicted the result obtained for the notch filter.



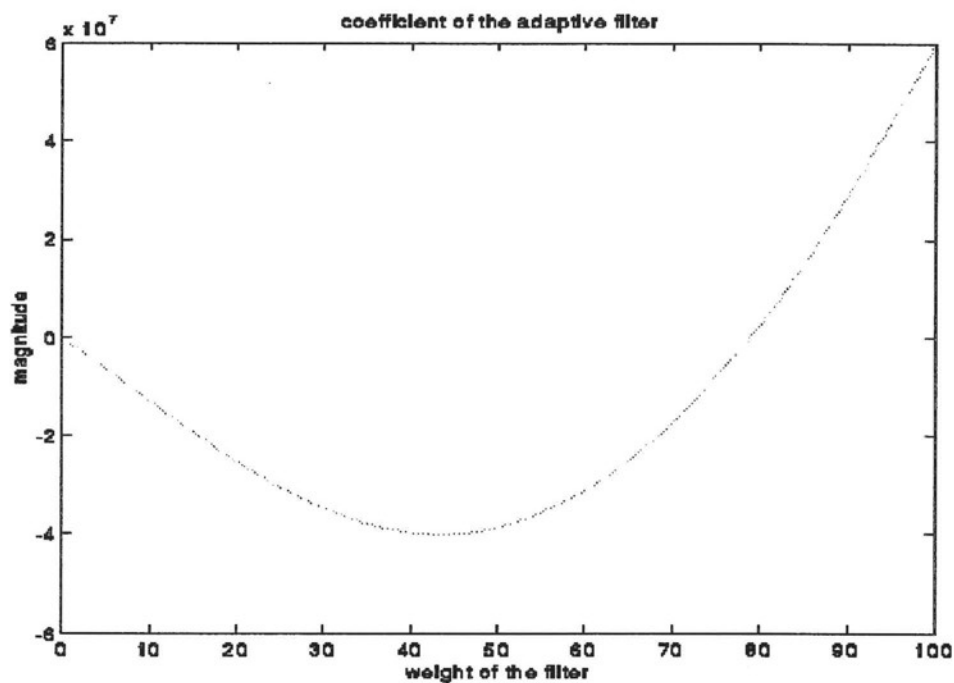
(Fig 8.17(e))

illustration of the error signal as seen by the notch filter

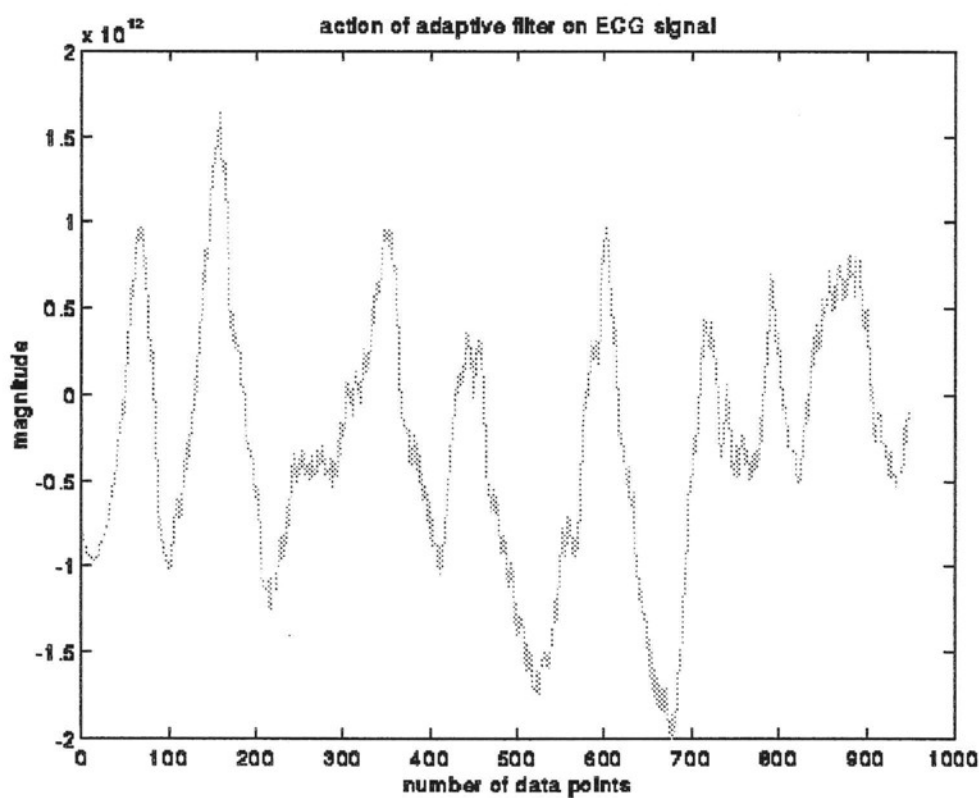


(Fig 8.17(f))

illustration of the error generated in estimating the noise in the ECG signal sampled at 18kHz



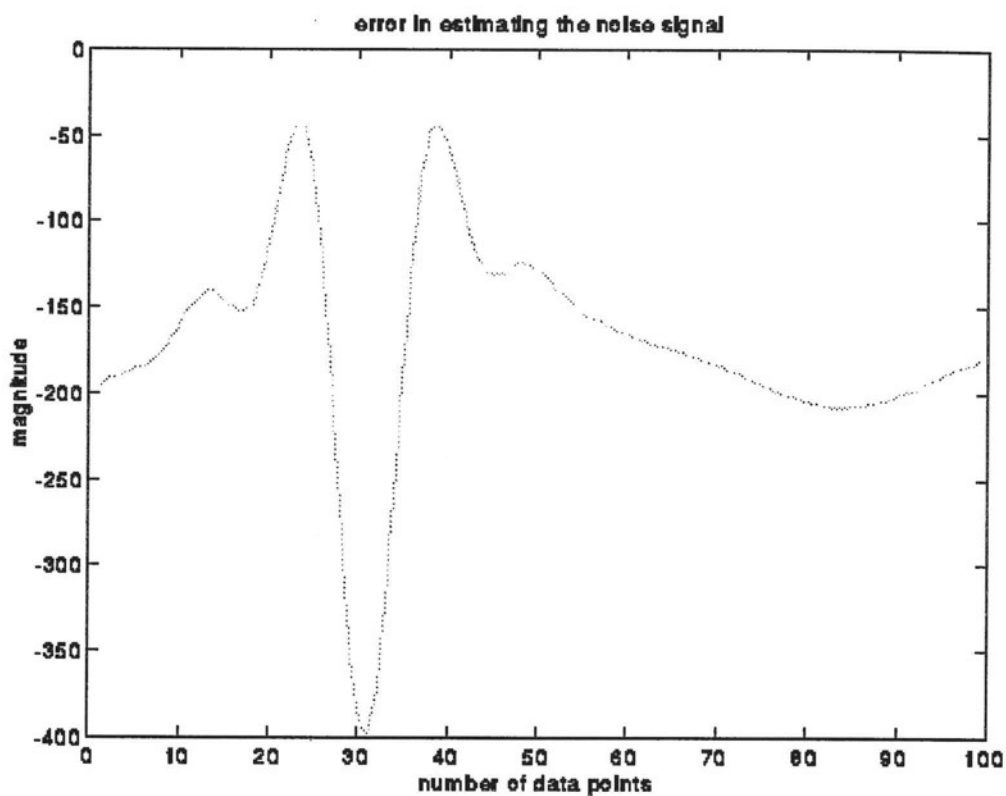
(Fig 8.17(g))
illustration of the new weight generated by the adaptive algorithm employing the notch filter



(Fig 8.17(h))
illustrating the action of the new filter weight on the ECG signal acquired at 18kHz

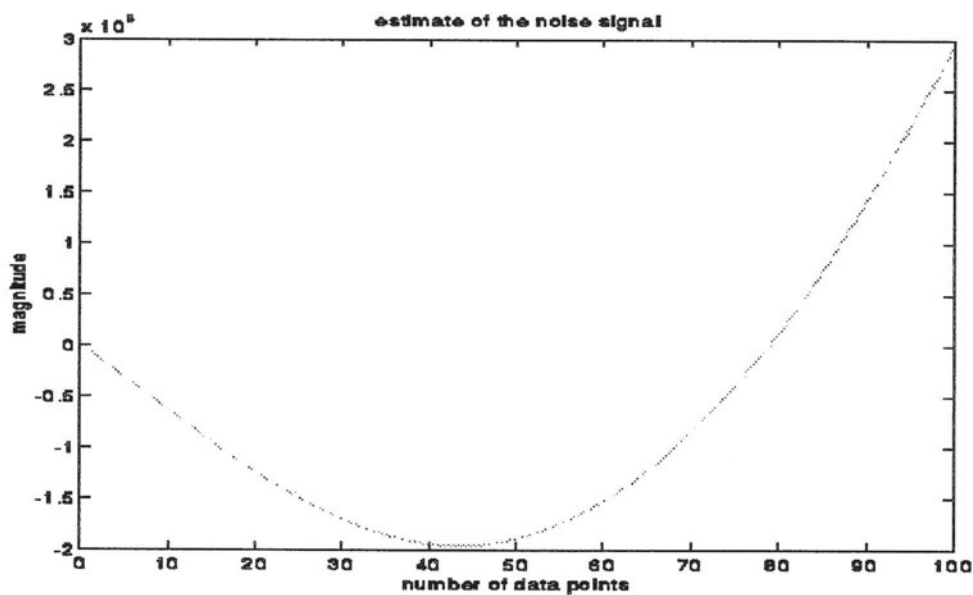
8.93 Lowpass filtering

By similar reasoning the figures 8.17(i) through 8.17(l) was obtained and depicted as follows:



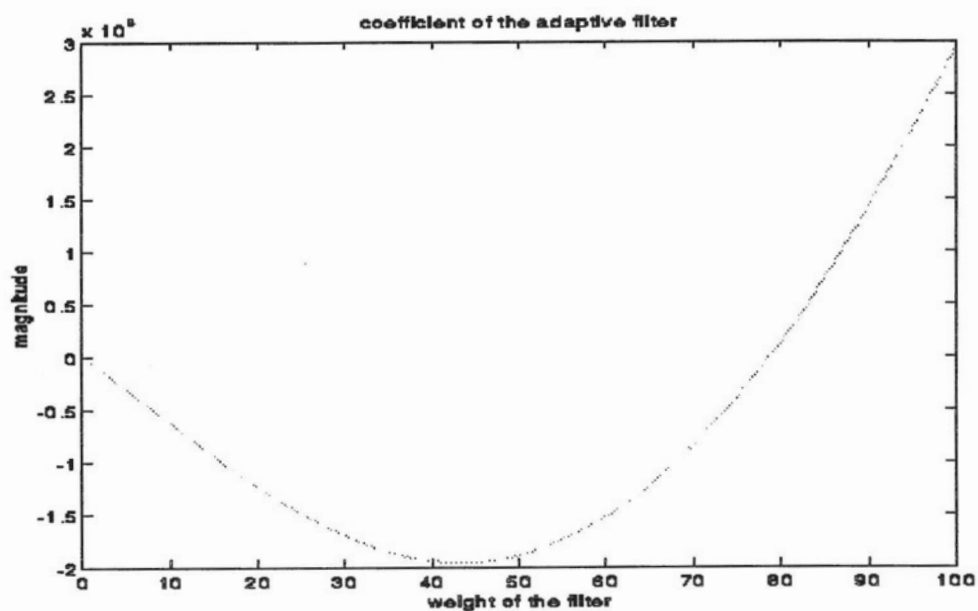
(Fig 8.17(i))

illustrating the estimate of the noise in the acquired signal



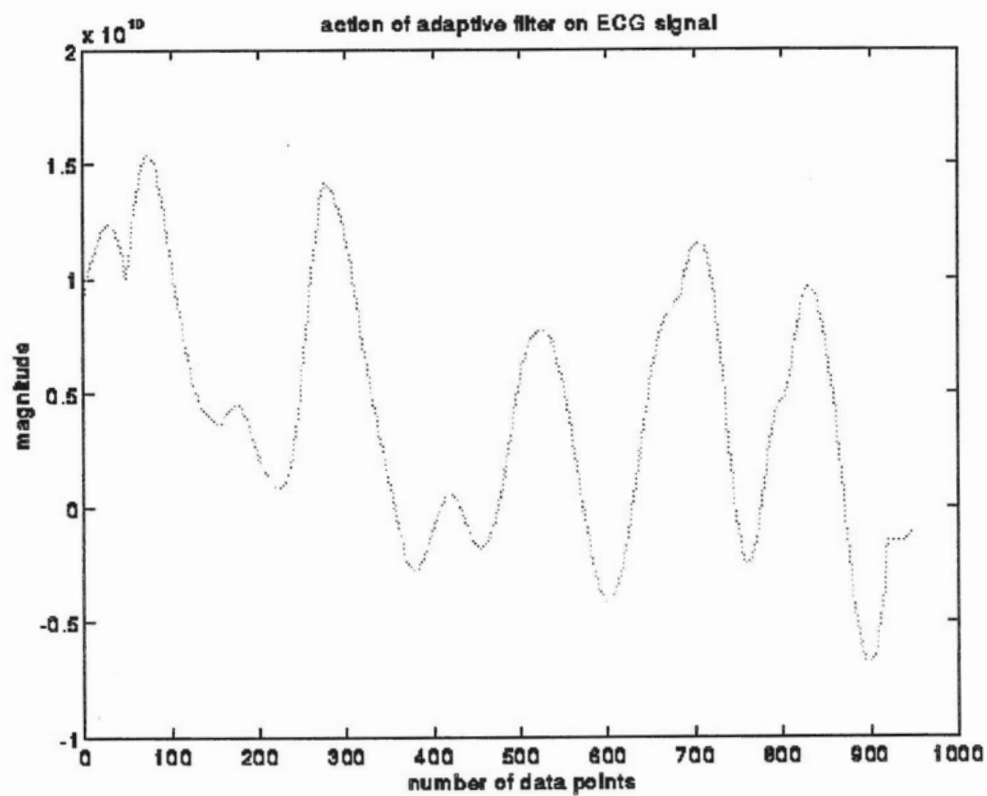
(Fig 8.17(j))

illustration of the error in estimating the noise in the corrupted signal



(Fig 8.17(k))

illustration of the new weight generated by the lowpass filter



(Fig 8.17(l))

illustration of the action of the new adaptive filter on ECG acquired at 250Hz

CHAPTER 9 CONCLUSION

Other than adaptive filtering in real-time, the set objectives of this project have been met. Namely, digital filter design for reduction of noise in a low frequency signal, development of differential equation for modelling of adaptive noise cancelling technique, identification of the optimum value of the tracking parameter, μ , and software design.

However, the adaptive filter, operational in real-time could be achieve given some more time.

Nevertheless the results achieved from the digital filter design are satisfactory. The actions of digital lowpass filter, bandpass filter and the notch filter on an ECG signal have been demonstrated prior to and after the filtering processes, (see fig 8.5(b), 8.5(c), 8.9(b) and 8.10). However, the bandpass filter appears to be the most effective of all the filter designed in reconstructing the ECG signal, (see fig 8.11(a)).

It was noted that the length, N , of the lowpass filter design can be considerably high using window method, especially when application requires a sharp transition width. For example in this design, should the transition width be reduced from 15 to 5Hz the length of the filter was found to be 171 and Blackman window was of the order of 275. Therefore this method is not recommended. Finally the pole-zero placement filter design was quite simple, and easy to implement. Although the error in realising the exact notch frequency is negligible, (i.e 4 parts in 10000), see appendix B, and the magnitude and phase response also confirms this. Consequently this obviates the need to consider more complex approach such as the impulse invariant or the bilinear z-transformation.

The optimum value obtained for the tracking parameter, μ may require further investigation about the frequency of interest. Several other cases may have to be considered, other than those considered in appendix D, hence there may exist some other combination of parameter to give optimum value of μ . Time constraint does not permit such an investigation. However, with the system decoupled, it may be possible to operate at higher frequency, than the frequency predicted from its simulation. This can be done by obtaining some feedback matrix (or gain matrix) which will stabilise the system at a higher frequency of interest. Although it must be pointed out tha the frequency cannot be increase indiscriminately.

The results obtained are comparable with on-going research. Although it have to be bore in mind that these results suffers from two set backs -

- a) method of acquisition of the ECG signal
- b) ability to remove the baseline (see fig 8.11(a))

In this field of research acquisition of the ECG signals are usually achieved by placing electrodes at some prescribed part of the body e.g by the side of the ribs, wrist etc., as opposed from the palm. Some of the desired peaks may not be well define in this instance and this is true for this particular case, [22].

The new filter weights generated from the adaptive filter algorithm appears identical. This will need to be investigated further.

CHAPTER 9 CONCLUSIONS

Other than adaptive filtering in real-time, the set objectives of this project have been met. Namely, digital filter design for reduction of noise in a low frequency signal, development of differential equation for modeling of adaptive noise canceling technique, identification of the optimum value of the tracking parameter, μ , and software design.

However, operation of the adaptive filter, in real-time could be achieve given more time.

Nevertheless, the result achieved from the digital filter design are satisfactory. The actions of digital lowpass filter, bandpass filter and the notch filter on an ECG signal have been demonstrated prior to and after the filtering processes, (see fig 8.5(b), 8.5(c), 8.9(b) and 8.10). However, the bandpass filter appears to be the most effective of all the filters designed in the reconstructing the ECG signal, (see fig 8.11(a)).

It was noted that the length, N of the lowpass filter design can be considerably higher using the window method, especially when the application requires a sharp transition width. For example, in this design, should the transition width be reduced from 15 to 5Hz the length of the filter was found to be 171 and 275 using Kaiser and Blackman windows respectively. Therefore this method is not recommended. Finally the pole-zero placement filter design was quite simple, and easy to implement. The error in realising the exact notch frequency is negligible, (i.e 4 parts in 10000), see appendix B, and the magnitude and phase response also confirms this. Consequently this obviates the need to consider a more complex approach such as the impulse invariant or the bilinear z-transformation.

The optimum value obtained for the tracking parameter μ may require further investigation about the frequency of interest. Several other cases may have to be considered, other than those considered in appendix D, hence there may exist some other combination of parameters to give an optimum value of μ . Time has not permitted such an investigation. However, with the system decoupled, it may be possible to operate at a higher frequency than the frequency predicted from its simulation. This can be done by obtaining some feedback matrix (or gain matrix) which will stabilise the system at a higher frequency of interest. Although it must be stressed that the frequency cannot be increased indiscriminately.

The results obtained are comparable with on-going research. Although it has to be borne in mind that these results suffer from two set backs :-

- a) method of acquisition of the ECG signal,
- b) ability to remove the baseline (see fig 8.11(a)).

In this field of research, acquisition of the ECG signals is usually achieved by placing electrodes at some prescribed part of the body e.g by the side of the ribs, wrist etc., as

opposed to the palm. Some of the desired peaks may not be well define in this instance and this is true for this particular case, [22].

The new filter weights generated from the adaptive filter algorithm appears identical. This will require further investigation.

SCOPE FOR FURTHER STUDIES

The filters designed in this project can be achieved by different methods as indicated in the design methodology, see chapter 4.

However, for the lowpass and notch filters, as mentioned earlier. In the case of the lowpass filter, the window method does not provide very much flexibility regarding the passband and stopband ripples. Furthermore, this method did not produce the minimum number of filter coefficients, which is vital when considering real-time applications. This problem can be circumvented by considering the Remez Exchange algorithm or optimum algorithm as indicated earlier [12]. While this offers an attractive solution, the computation involved could be demanding. Apart from this rounding errors, due to finite wordlength effects, would have to be given serious consideration in this instance as the convergence state may not be attained, and subsequently the filter characteristics will be unobtainable.

For filters with length more than a hundred, it may be advisable to employ the multirate technique during the acquisition of the signal, as this technique would enhance the reconstruction of desired signal and efficient processing of the filter coefficients, especially in real-time, [2].

Further work is desirable, especially in estimating the magnitude of the noise source and consequently the signal to noise ratio. This information is invaluable in determining the effectiveness of the implemented filters. Attention should equally be paid to roundoff errors, quantisation error and finite wordlength effects.

Finally the success of this project will depend on the ability to generate a suitable reference signal.

REFERENCES

- [1]. Richard, C., and Johnson, J.R.,
On the interaction of adaptive filtering, identification and control.
IEEE Signal Processing, **vol.12**, no.2, pp. 462-72, March 1995.
- [2]. Emmanuel, C. I., and Barry, W.J.,
Digital signal processing practical approach.,
Addison Wesley Publishers Ltd, New York, 1993.
- [3]. Glover, J.R.,
Comments on digital filters for real time ECG processing using
microprocessors,
IEEE Transaction Biomedical Engineering,
vol. BME-34, pp. 962-963, December 1987.
- [4]. Gao Jun, and Luo Bin Feng,
The study of frequency tracking adaptive notch filter in ELF communication,
ICCT '92 proceedings of 1992.
International conference on communication technology,
Beijin, China, **vol.2**, pp. 6-18, Sept.1992.
- [5]. Girgis, A., and Bing Chen,
Design of programmable FIR digital filter to optimise the detection of signals
embedded in white noise.
IEEE Instrumentation and Measurement technology conference.
pp. 224A - 224F, May 1992.
- [6]. Lowen, C.F.N.,
Adaptive filters,
Englewood cliffs, NJ, Prentice - Hall, Inc., 1985.
- [7]. Johnson, C.R.,
Adaptive IIR filtering, current results and open issue,
IEEE Trans. Information Theory,
vol. IT-30, pp.237 - 249, March 1984.
- [8]. Ferntuch, P.C.,
Adaptive recursive LMS filter,
Proceedings IEEE, **vol.64**, pp.1622 - 1624, Nov.1976.

- [9]. Flynn, M.O.,
Probabilities random variables and random processes,
Harper and Row, New York, 1982.
- [10]. Mortara, D.W.,
Digital filters ECG signals,
Comput. Lardio., pp. 511 - 514, 1977.
- [11]. Ferdjalhah, M., and Ronald, E.B.,
Adaptive digital notch filter design on the unit circle for the removal of
powerline noise from biomedical signals,
IEEE Transaction on Biomedical Engineer, **vol.41**, no.6, June 1994.
- [12]. McClellan, J.H., Parks, T.W, and Rabiner, L.R.,
A computer program for designing optimum FIR linear phase digital filters,
IEEE Trans. Audio Electroacoustics, **vol. 21**, pp. 506-26, 1973.
- [13]. McClellan, J.H., Parks, T.W., and Rabiner, L.R.,
A computer program for designing optimum FIR linear phase digital filters,
IEEE Trans.Audio Electroacoustics,
vol.21, pp.506-26, 1973.
- [14]. Bozic, S.M.,
Digital and Kalman filtering,
Unwin brothers Ltd, surrey, 1979.
- [15]. Thomas Alexander, S.,
adaptive signal processing theory and applications,
Springer - Verlag, London, 1986.
- [16]. Holykin Simon,
Introduction to adaptive filters, Macmillan,
Inc, New York, 1984.
- [17]. Widrow B.
stationary and nonstationary learning characteristics of the LMS adaptive
filter. Proc IEEE, **vol 64** pp 1151-62.
- [18]. Kris, A., and Jamsa, I.,
Programming in C
Irwin, Homewood, Illinois 1986.
- [19]. John Thomas Berry,
Advanced C programming,
Prentice Hall press, NJ, 1986.

- [20]. Brandy, C express,
Robert Jourdain programmers library,
Brady utility, New York, 1989.

- [21]. R. Hinsen, R. V. Essen, J. Silny, W. Merx, G. Rau and S. Effert.,
Monitoring of Myocardial Ischemia and Necrosis in Acute Myocardial
Infarction, Computers in Cardiology, **section 3**, pp. 71-76, 1979.

- [22]. Emmanuel C. Ifeachor,
Medical Applications of DSP
DSP: Theory, Applications and Implementation,
IEE Colloquium Younger Members Section
pp. 4/1- 4/13, 17 January 1996.

- [23.] Mr. D. J. Lewis,
Systems Theory
ESD Department
College of Aeronautics
Cranfield University

APPENDIX A

DERIVATION OF DIFFERENTIAL EQUATION

From the noise cancelling adaptive configuration, sec 4.5, fig 4.6. It was established that

$$\begin{aligned}
 1. \quad y(t) &= \frac{2\mu A^2}{T_s} \left\{ \sin \omega_o t \int_0^t e(t) \sin \omega_o t dt + \cos \omega_o t \int_0^t e(t) \cos \omega_o t dt \right\} \\
 \ddot{y}(t) &= \frac{2\mu A^2}{T_s} \left\{ \begin{aligned} &\omega_o \cos \omega_o t \int_0^t e(t) \sin \omega_o t dt + \sin^2 \omega_o t e(t) \cos^2 \omega_o t e(t) \\ &-\omega_o \sin \omega_o t \int_0^t e(t) \cos \omega_o t dt \end{aligned} \right\} \\
 &= \frac{2\mu A^2}{T_s} \left\{ \begin{aligned} &e(t) [\sin^2 \omega_o t + \cos^2 \omega_o t] + \omega_o \cos \omega_o t \int_0^t e(t) \sin \omega_o t dt \\ &-\omega_o \sin \omega_o t \int_0^t e(t) \cos \omega_o t dt \end{aligned} \right\} \\
 \Rightarrow \quad &\frac{2\mu A^2}{T_s} \left\{ e(t) + \omega_o \cos \omega_o t \int_0^t e(t) \sin \omega_o t dt - \omega_o \sin \omega_o t \int_0^t e(t) \cos \omega_o t dt \right\} \\
 \ddot{y}(t) &= \frac{2\mu A^2}{T_s} \left\{ \begin{aligned} &e'(t) - \omega_o^2 \sin \omega_o t \int_0^t e(t) \sin \omega_o t dt + \omega_o \sin \omega_o t \cos \omega_o t e(t) \\ &-\omega_o \sin \omega_o t \cos \omega_o t e(t) + \omega_o^2 \cos \omega_o t \int_0^t e(t) \cos \omega_o t dt \end{aligned} \right\} \\
 \ddot{y}(t) &= \frac{2\mu A^2}{T_s} \left\{ e'(t) - \omega_o^2 \left[\sin \omega_o t \int_0^t e(t) \sin \omega_o t dt + \cos \omega_o t \int_0^t e(t) \cos \omega_o t dt \right] \right\} \\
 &= \frac{2\mu A^2}{T_s} e'(t) - \omega_o^2 y(t)
 \end{aligned}$$

Now consider the R.H.S

$$\ddot{y}(t) = \frac{2\mu A^2}{T_s} \omega_o^2 y'(t) + \omega_o^2 y(t)$$

$$\Rightarrow = \frac{2\mu A^2}{T_s} e'(t) - \omega_0^2 y(t) + \frac{2\mu A^2}{T_s} \omega_0^2 y'(t) + \omega_0^2 y(t)$$

$$\Rightarrow = \frac{2\mu A^2}{T_s} [e'(t) - y(t)]$$

but $e(t) = x(t) - y(t)$

$$\Rightarrow \frac{2\mu A^2}{T_s} x'(t)$$

ie R.H.S = L.H.S as required

APENDIX B

ESTIMATION OF WORDLENGTH

Before embarking on filter design, it is desirable to investigate the number of bits that are needed to represent the coefficient of the filters.

Bandpass filter

Reference to sec 4.4.2, the z-transfer function of the filter was of the form $H(z) = H_1(z).H_2(z)$

$$H_1(z) = \frac{0.11310(z^2 - 1)}{1.11468z^2 - 1.99684z + 0.99955}$$

since $H_1(z)$ is identical to $H_2(z)$, similar analysis holds for $H_2(z)$. The denominator coefficient of each of the second-order section were quantised by rounding to B-bits (B= 2, 3,.....) including the sign bits. For each value of B, the quantised coefficients and the pole location in polar form was computed as follows:

For B = 8 bits, the coefficients are quantised by rounding

$$\begin{aligned} b_1 &= (-1.99684 \times 2^7 + 0.5) \\ &= 256.09552 \text{ (5d.p)} \\ &= -256 \end{aligned}$$

$$\begin{aligned} b_2 &= (0.99955 \times 2^7 + 0.5) \\ &= 128.4424 \\ &= 128 \end{aligned}$$

In fractional notation, the coefficients are

$$\begin{aligned} b_1 &= -256/128 \\ &= -2.00000 \\ b_2 &= 128/128 \\ &= 1.00000 \end{aligned}$$

from section (4.4.4)

the pole-radius and angle for the 8-bits are given by from equ (4.4.30) and (4.4.31)

$$\begin{aligned} r &= \sqrt{1.00000} \\ r &= 1 \text{ unit} \\ \theta &= \cos^{-1}(2/2(1)) \\ &= 0^\circ \end{aligned}$$

Clearly, for B = 8-bits, $r = 1$ unit signifies that stability cannot be maintain for this bit size, therefore need to consider higher number of bits.

For B = 10-bits

$$\begin{aligned}b_1 &= (-1.99684 \times 2^9 + 0.5) \\&= -1022.88208 \text{ (5d.p)} \\&= -1023\end{aligned}$$

$$\begin{aligned}b_2 &= (0.99955 \times 2^9 + 0.5) \\&= 512.2696 \\&= 512\end{aligned}$$

in fractional notation, the coefficients are

$$\begin{aligned}b_1 &= -1028/512 \\&= -1.99805 \text{ (5d.p)}\end{aligned}$$

$$\begin{aligned}b_2 &= 512/512 \\&= 1.00000\end{aligned}$$

similarly using the pole-radius angle for the 10-bits, are given by

$$\begin{aligned}r &= \sqrt{1.998050} \\r &= 1.41352 \text{ unit (5d.p)} \\ \theta &= \cos^{-1}(1/2(1.41352)) \\&= 69.28^\circ\end{aligned}$$

better still $r = 1.41352$ reflects that the filter cannot be stable for B= 10.

Consider B= 12-bits

∴

$$\begin{aligned}b_1 &= (-1.99684 \times 2^{11} + 0.5) \\&= -4090.02832 \text{ (5d.p)} \\&= -4090 \\b_2 &= (2047.578) \\&= 2048\end{aligned}$$

in fractional notation, the coefficient are

$$b_1 = -1.99707(5\text{d.p})$$

$$\begin{aligned}b_2 &= 2048/2048 \\&= 1.00000\end{aligned}$$

similarly using the pole-radius angle for the 10-bits, are given by

$$\begin{aligned}r &= \sqrt{1.99707} \\r &= 1.41318 \text{ unit (5d.p)} \\ \theta &= \cos^{-1}(0.70659) \\&= 45^\circ\end{aligned}$$

Consider B=16

$$\begin{aligned}\therefore b_1 &= (-1.99684 \times 2^{15} + 0.5) \\ &= -65432.95312 \text{ (5d.p)} \\ &= -65433\end{aligned}$$

$$\begin{aligned}b_2 &= (0.99955 \times 2^{15} + 0.5) \\ &= 32753.7544 \\ &= 32754\end{aligned}$$

in fractional notation, the coefficient are

$$\begin{aligned}b_1 &= -65433/32768 \\ &= -1.99686 \text{ (5d.p)}\end{aligned}$$

$$\begin{aligned}b_2 &= 32754/32768 \\ &= 0.99957 \\ &= \sqrt{0.99957}\end{aligned}$$

$$\begin{aligned}r &= 0.99979 \text{ unit (5d.p)} \\ \theta &= \cos^{-1}(0.99864) \\ &= 2.98^\circ\end{aligned}$$

For B= 16-bits the filter would be stable. In practice however, consideration may have to be given to scaling of the input signal.

Notch filter

similarly from sec 4.4.3 equ 4.4.25, the z-transfer function for the notch filter design was found to be

$$H(z) = \frac{z^2 - 0.618034z + 1}{z^2 - 0.35122z + 0.87828}$$

Considering the denominator as with case of the bandpass filter,

$$b_1 = 0.35122 ; \quad b_2 = 0.87828$$

$\therefore$ for B = 8-bits

$$\begin{aligned}b_1 &= (-0.35122 \times 2^7 + 0.5) \\ &= -45.45616 \text{ (5d.p)} \\ &= -45 \\ b_2 &= (0.87828 \times 2^7 + 0.5) \\ &= 112.91984 \\ &= 113\end{aligned}$$

In fractional notation

$$\begin{aligned}b_1 &= -45/128 \\ &= -0.35156 \text{ (5d.p)} \\ b_2 &= 113/128 \\ &= 0.88281 \\ &= \sqrt{0.35156} \\ r &= 0.59292 \text{ unit (5d.p)} \\ \theta &= \cos^{-1}(0.29646) \\ &= 72.75^\circ\end{aligned}$$

For B = 8-bits the filter was found to be stable.

Consider B = 16-bits

$$\begin{aligned}b_1 &= (-0.35122 \times 2^{15} + 0.5) \\ &= -11509.27696 \text{ (5d.p)} \\ &= -11509 \\ b_2 &= (0.87828 \times 2^{15} + 0.5) \\ &= 28779.97904 \\ &= 28780\end{aligned}$$

In fractional notation

$$\begin{aligned}b_1 &= -11509/32768 \\ &= -0.35123 \text{ (5d.p)} \\ b_2 &= 28780/32768 \\ &= 0.87830 \\ &= \sqrt{0.87830} \\ r &= 0.93717 \text{ unit (5d.p)} \\ \theta &= \cos^{-1}(0.18739) \\ &= 79.199^\circ\end{aligned}$$

To two decimal place, this angle is found to be exactly equal to the calculated value, (see sec 4.4.3.) The rounding error of the coefficient is

$$\begin{aligned}\text{round off error} &= 79.19963/79.2 \times 100\% \\ &= 4.67 \times 10^{-6}\end{aligned}$$

ie is approximately 5 parts per million.

Having said this, it was found that 8-bits was sufficient to guarantee the filter stability. However, this number of bits may not necessarily meet the specifications for the passband and stopband attenuation, and consequently these parameters need to be investigated.

The minimum stopband attenuation $A_{\min}$ would be given by

$$A_{\min} = 20\log_{10}((\sqrt{1.5 \times 2^{B+1}})|_{B=8})$$

$$= 28.85\text{dB}$$

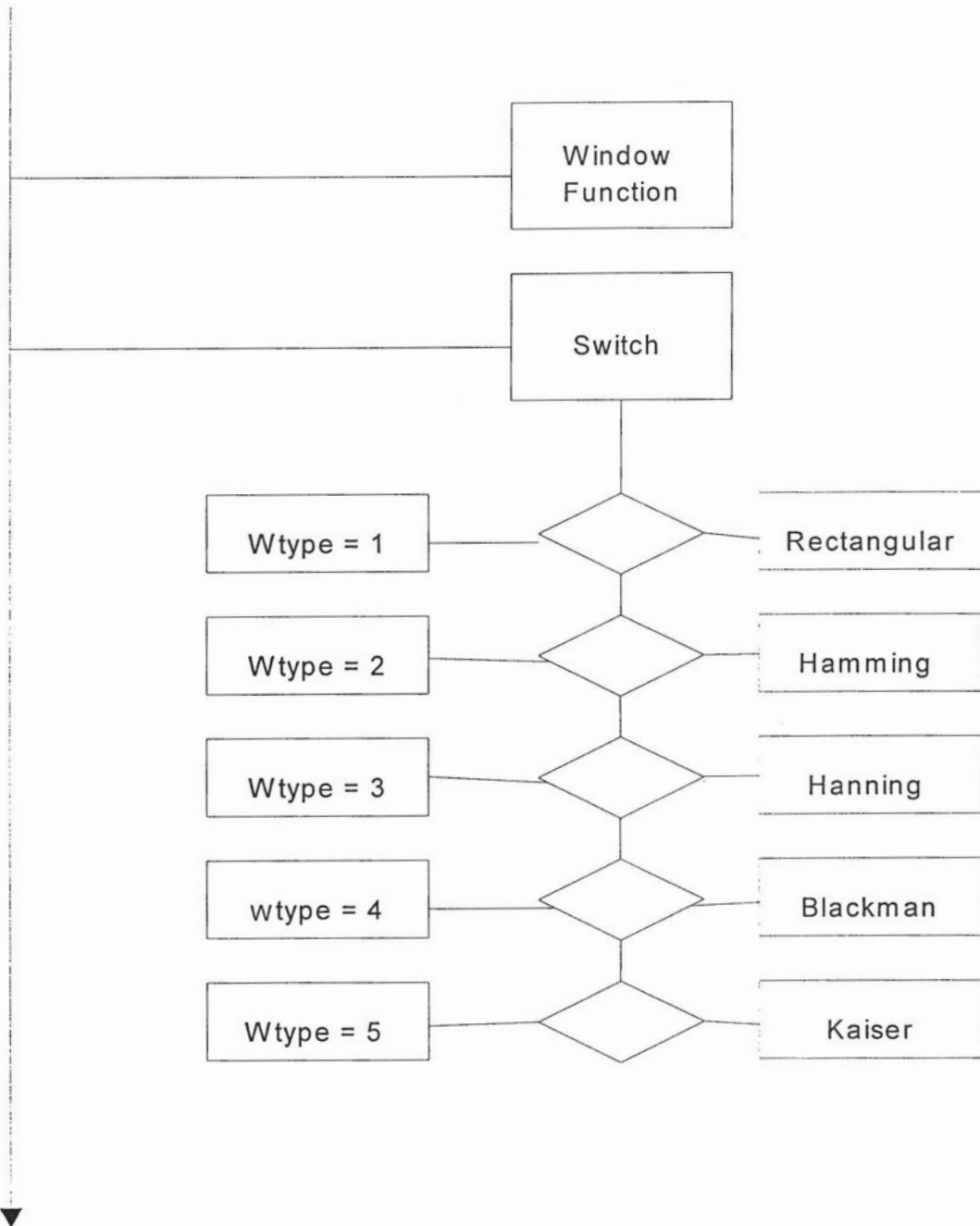
Clearly this attenuation does not meet the specified requirement.

For B = 10-bits, the attenuation at notch was found to be 34.87dB.

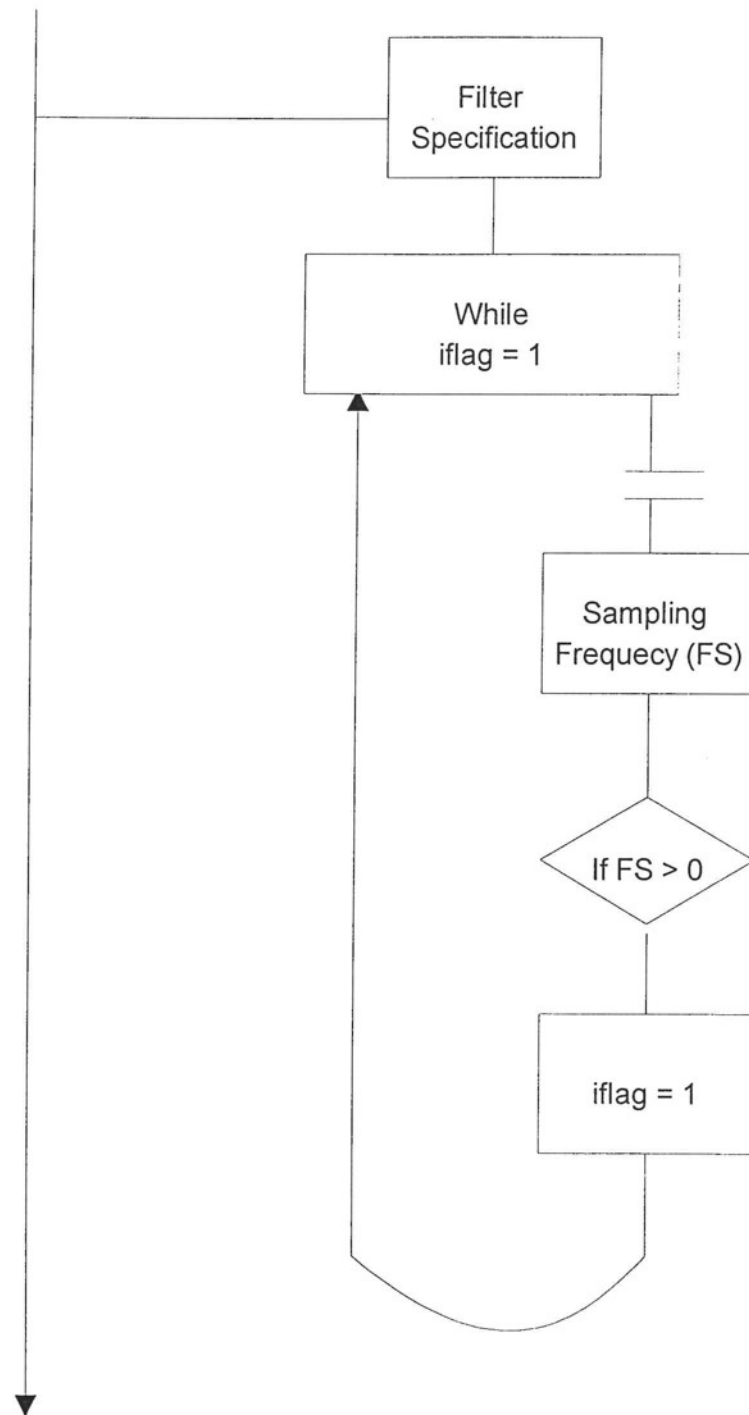
The actual number of bits used was 16. And the attenuation was found to be 52.93dB.

The error in attenuation requirement was found to be 11.8%.

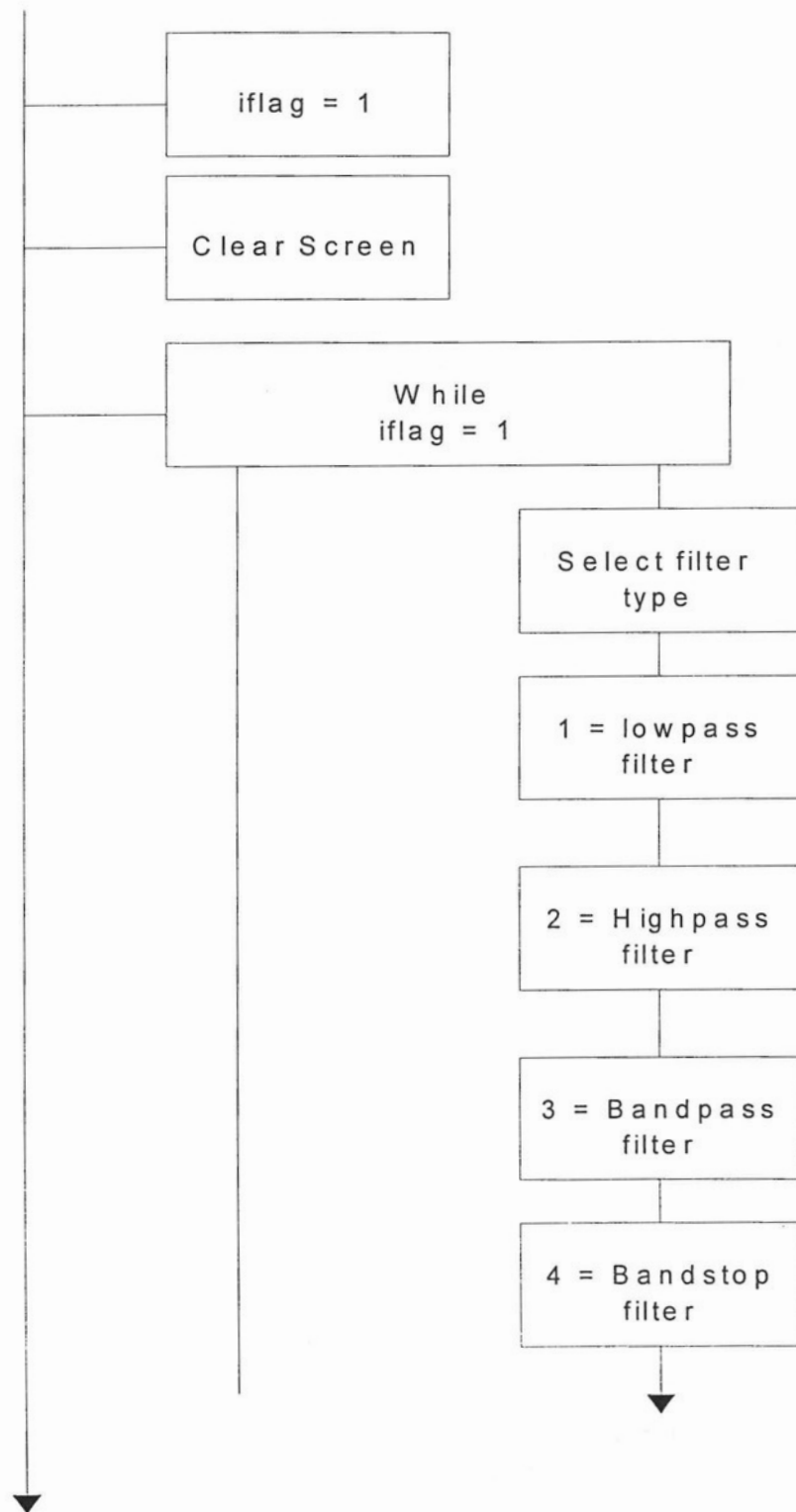
APPENDIX (c)



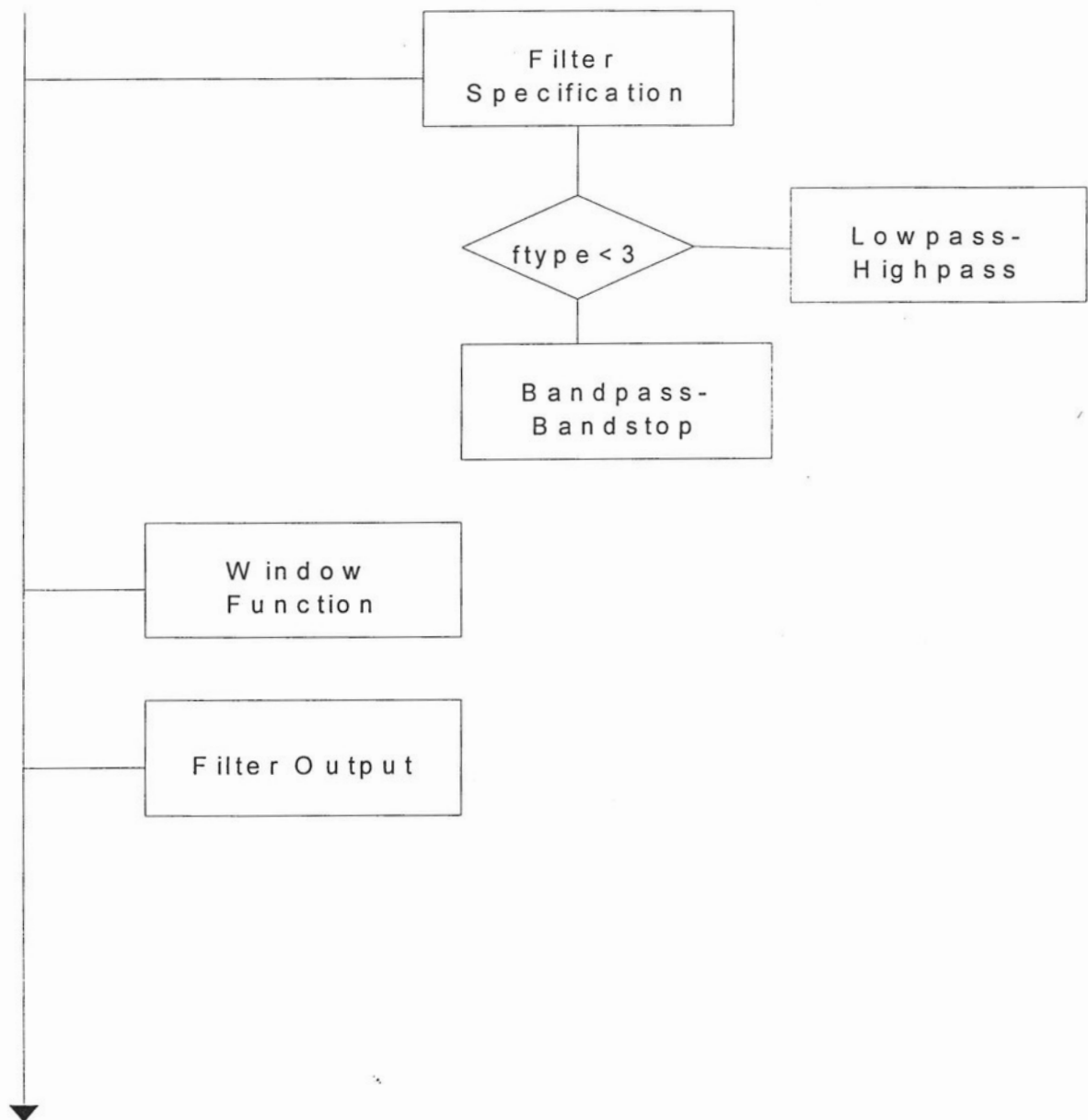
APPENDIX (c)



APPENDIX (c)



APPENDIX (c)



Flowchart illustrating window software design

APPENDIX D

Cases	f_s/kHz	A	μ	$\omega_o^2/\text{radians}$	Gain = $\frac{2\mu A^2}{T_s}$
1	18	10	1	98696	1130973
2	18	1	1	98696	1103697
3	18	0.1	1	98696	1131
4	18	1	0.1	98696	11309
5	18	10	0.01	98696	113
6	0.25	10	1	314	31400
7	0.25	1	1	314	3140
8	0.25	0.1	0.1	314	31.4
9	0.25	0.1	0.01	314	3.14
10	0.314	0.2	0.01	314	0.628
11	0.314	0.2	0.1	314	6.28
12	18	4.8	1	94.25	829440
13	18	10	1	94.25	36×10^5
14	18	1	1	94.25	36000
15	18	0.1	1	94.25	360
21	0.000006	0.1	0.09	94.25	32.4
22	0.000006	0.2	0.05	94.25	72
23	0.000006	0.2	0.09	94.25	129.6
24	0.000006	0.2	0.1	94.25	144