MatLab Based Design and Implemetation of Digita Filter

Zena Ez Dallalbashi

Technical Institute/ Electronic Department Northren Technical University

Abstract

Digital filters are essential in the wider field of digital signal processing applications, such as spectrum analysis, digital image processing, and pattern recognition. In signal processing, the function of a filter is to remove unwanted parts of the signal, such as random noise, or to extract useful parts of the signal, such as the components lying within a certain frequency range. A digital filter uses a digital processor to perform numerical calculations on sampled values of the signal. The processor may be a general-purpose computer such as a PC, or a specialised DSP (Digital Signal Processor). With the basic building blocks at hand. Finite impulse response filters (FIR) and infinite impulse response filters (IIR) are the two principal techniques for filtering digital signals. The work is to design PC- based Digital Filter including the two principal techniques using the Graphical User Interface (GUI) for the Mat lab software; front panel is designed as control panel involving all functions and tools necessary for filtering. Hardware requirements for interfacing the signal to the PC are considered and designed. The sound card used for signals in the audio range. External signals can be recorded, displayed, stored and processed.

Key word:

Digital filter, GUI, Mat lab software, FIR, IIR, Panel

1. Introduction

Digital filters are used in a wide variety of signal processing applications, such as spectrum analysis, digital image processing, and pattern recognition. Digital filters eliminate a number of problems associated with their classical

analog counterparts and thus are preferably used in place of analog filters. Digital filters belong to the class of discrete-time LTI (linear time invariant) systems, which are characterized by the properties of causality, recursibility,

and stability. They can be characterized in the time domain by their unit-impulse response, and in the transform domain by their transfer function. Obviously, the unit-impulse response sequence of a causal LTI system could be of either finite or infinite duration and this property determines their classification into either finite impulse response (FIR) or infinite impulse response (IIR) system. [2]

Literature Review

The development of digital filter began more than half of century, many auditory filter models have been developed, analyzed, and applied. Herrmann was the first to develop a method for designning optimal (in a Chebyshevsense) FIR filters. By assuming that the frequency response of the optimal low-pass filter was equiripple in both the passband and the stopband, and by fixing the number of ripples in each band, 1970[1]. Rahman Jamal, Mike Cerna, John Hanks Design Filters Using digital filter Tool Kit using LabviewTM software December 1996 [2]. Timothy J. Schlichter discusses the different options in Matlab and gives examples of lowpass, highpass, and bandpass filter designs 5/2/99[3]. Ricardo A. Losada The MathWorks, Inc. MA 01760, USA illustrates practical aspects of FIR filter design and fixed-point implementation along with the algorithms available in the Filter Design Toolbox and the Signal Processing Toolbox for this purpose 2004[5].

Aim and structure

The aim of the research is to design and implement digital filter including digital filter family that are the finite impulse response, infinite impulse response. using the Graphical User Interface (GUI) for mat lab software. The paper falls into five sections. After this introduction, a basic theory for the digital filter tools is presented in section two. Section three explains the design of the digital filter. Results and description of filtering signals is presented in section four. Conclusions are introduced in section five.

2. Theory

2.1 Digital Filter:-

A digital filter is characterized by its transfer function, or equivalently, its difference equation. Mathematical analysis of the transfer function can describe how it will respond to any input. As such, designing a filter consists of developing specifications appropriate to the problem and then producing a transfer function which meets the specifications. There are two basic types of digital filters, Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filters.The general form of the digital filter difference equation is:-

$$y(n)=\sum_{i=0}^{N}a_i x(n-i)-\sum_{n=1}^{N}b_i y(n-i) \qquad (1)$$

Manuscript received January 5, 2020
Manuscript revised January 20, 2020

where y(n) is the current filter output, the y(n-i)'s are previous outputs, the x(n-i)'s are current or previous filter inputs, the ai's are the filter's feed forward coefficients corresponding to the zeros of the filter, the bi's are the filter's feedback coefficients corresponding to the poles of the filter, and N is the filter's order. IIR filters have one or more nonzero feedback coefficients. That is, as a result of the feedback term, if the filter has one or more poles, once the filter has been excited with an impulse there is always an output. FIR filters have no non-zero feedback coefficient.That is, the filter has only zeros, and once it has been excited with an impulse, the output is present for only a finite (N) number of computational cycles.Because an IIR filter uses both a feed-forward polynomial (zeros as the roots) and a feedback polynomial (poles as theroots), it has a much sharper transition characteristic for a given filter order[6].

Non-recursive filters FIR Filter:-

A finite impulse response (FIR) filter is a type of a signal processing filter whose impulse response (or response to any finite length input) is of finite duration, because it settles to zero in finite time. The impulse response of an Nth-order discrete-time FIR filter (i.e. with a Kronecker delta impulse input) lasts for N+1 samples, and then dies to zero.FIR filters can be discrete-time or continuous-time, and digital or analog.

figure 1 shows a discrete-time FIR filter of order *N*. The top part is an *N*-stage delay line with *N*+1 taps. Each unit delay is a z^{-1} operator in Z-transform notation.The output *y* of a linear time invariant system is determined by convolving its input signal *x* with its impulse response *b*.For a discrete-time FIR filter, the output is a weighted sum of the current and a finite number of previous values of the input. The operation is described by the following equation, which defines the output sequence *y[n]* in terms of its input sequence *x[n]*:

$$y(n) = b_0 x[n] + b_1 x[n-1] + ... + b_N x[n-N]$$

.

$$y([n] = \sum_{i=0}^{N} b_i x[n-i] \qquad (2)$$

b_i are the filter coefficients, also known as tap weights, that make up the impulse response,

N is the filter order; an *N*th-order filter has (*N* + 1) terms on the right-hand side. The *x*[*n* − *i*] in these terms are commonly referred to as taps, based on the structure of a tapped delay line that in many implementations or block diagrams provides the delayed inputs to the multiplication operations. One may speak of a "5th order/6-tap filter", for instance.

An FIR filter has a number of useful properties:

1- Are inherently stable. This is due to the fact that, because there is no feedback, all the poles are located at the origin and thus are located within the unit circle.

2-Require no feedback. This means that any rounding errors are not compounded by summed iterations.

3-They can easily be designed to be linear phase by making the coefficient sequence symmetric; linear phase, or phase change proportional to frequency, corresponds to equal delay at all frequencies.

The main disadvantage of FIR filters is that considerably more computation power in a general purpose processor is required.[7]

Recursive Filters IIR Filter:-

Infinite Impulse Response (IIR) filters are fundamental elements of Digital Signal Processing whose impulse response extends for an infinite period period of time, this is because they are recursive. IIR filters are commonly implemented using a feedback (recursive) structure, while FIR filters usually require no feedback (non-recursive). In the design of IIR filters, a commonly used approach is called the bilinear transformation. This design begins with the transfer function of an analog filter, then performs a mapping from the s-domain to the z-domain. Using differential equations, it can be shown that the mapping from the s-plane to the z-plane is:-

$$S = \frac{2}{T}\left(\frac{1-Z^{-1}}{1+Z^{-1}}\right) \qquad (3)$$

This mapping results in a general form for an IIR filter with an arbitrary number of poles and zeros. The system response and the difference equation for this filter is as follows:

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{n=0}^{M} b_n z^{-n}}{\sum_{n=0}^{N} a_n z^{-n}} \qquad (4)$$

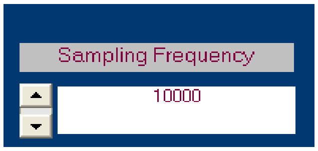
$$= \frac{b_0 + b_1 z^{-1} + - - b_M z^{-m}}{1 + a_1 z^{-1} + - - a_N z^{-N}}, a_{0=1} \qquad (5)$$

IIR filters have traditional analog counterparts (Butterworth, Chebyshev, Elliptic, and Bessel) and can be analyzed and synthesized using more familiar traditional filter design techniques[3]. IIR filters are generally implemented in two-pole sections called biquads because they are described with a biquadratic equation in the z-domain. Higher order filters are designed using cascaded biquad sections, the basic IIR biquad is shown in Figure 2. The zeros are formed by the feedforward coefficients b0,

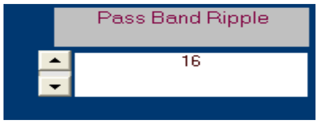
b1, and b2; the poles are formed by the feedback coefficients a1, and a2.Because an IIR filter uses both a feed-forward polynomial (zeros as the roots) and a feedback polynomial (poles as the roots), it has a much sharper transition characteristic for a given filter order.
A disadvantage of IIR filters is that they usually have nonlinear phase.
Some minor signal distortion is a result.

3. Digital Filter Design:-

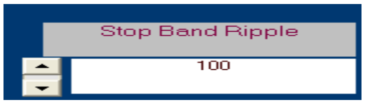
Figure (3) shows the front panel of the Digital Filter including all figure places and control icons requiring for displaying, read, measure, and manipulate the results of the signal filtering. GUI Toolbox, under Mat lab v6.5 software was used to design the front panel, acquire, process, and data filtering [4].
III.1 the following sections explain how to manipulate controls and read indicators in the digital filter.



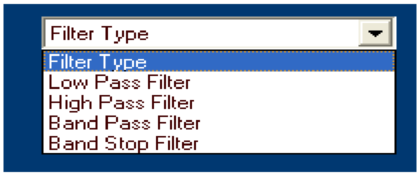
Setting The Sampling Frequency: - The samp. Freq control is used to specify the sampling frequency. The sampling frequency entered should match the sampling frequency that was used to generate the FIR, IIR filter coefficients. This parameter is used by the tool to generate sample data across valid frequencies and evaluate the frequency response using the output time samples.



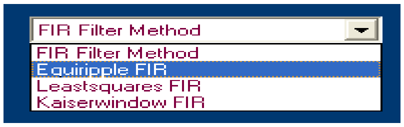
Pass Band Ripple: - The Pass Band Ripple control is used to specify the Pass band ripple: deviation from maximum gain (dB) in the pass band and filter order.

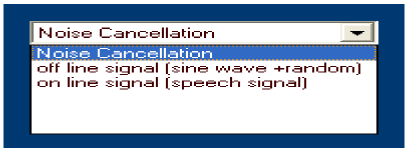


Stop Band Ripple: - The Stop Band Ripple control is used to specify Stop band attenuation: deviation from 0 gains (dB) in the stop band: filter order and coefficient calculation.

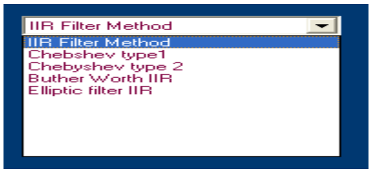


Filter Type :- The Filter Type control is used to specify the type of filter such as Low pass Filter, High pass Filter , Band pass Filter , Band stop Filter.




Noise Cancellation
Noise Cancellation
off line signal (sine wave +random)
on line signal (speech signal)

FIR Filter Methods control**: -** A full suite of design methods is available for finite impulse response (FIR). These methods support the rapid design and evaluation of low pass, high pass, band pass, and band stop, filters such as equiripple , lest squares and Kaiser window.



IIR Filter Methods control**: -** A full suite of design methods is available for infinite impulse response (IIR). These methods support the rapid design and evaluation of low pass, high pass, and band pass, and band stop, filters such as Butterworth, Chebyshev type1, Chebyshev type1, elliptic

Amplitude Response of IIR Filter:- clicking this Button shows the Amplitude response and pole zeros for the each method in two axes.

Amplitude Response of FIR Filter:- this Button shows the frequency response and pole zeros for the each method in two axes

Filter performance:-clicking this button simple discussion on digital filter is displayed as shown in figure 4

NoiseCancellation Control: - after specifying the filter requirement clicking this control signal can be filtered.

Selecting Finite Impulse Response Method (FIR):-

The FIR method control is used to select the filtering method. These methods support the rapid design and evaluation of low pass, high pass, band pass, and band stop, filters such as Equiripple , lest squares and Kaiser window .

FIR filters have the following primary advantages [8].

1-They can have exactly linear phase.
2-They are always stable.
3-The design methods are generally linear.

Designing Low Pass Filter:-

A low pass filter will be designed that passes all frequencies less than or equal to 20% of the Nyquist frequency (Half the sampling frequency) and attenuates frequencies greater than or equal to the Nyquist frequency. An FIR Equiripple filter will be designed with these specifications [3]:

Pass band attenuation 1 dB
Stop band attenuation 80 dB
A pass band frequency 0.2 [Normalized (0 to 1)]
A stop band frequency 0.5 [Normalized (0 to 1)]

To implement this design, the following specifications will used:

1- Select Low pass filter from the Filter Type control and Equiripple method from FIR Filter Method control.

2- Select Normalized (0 to 1) in the Pass band frequency control and Stop band frequency control.

3- Enter 0.1 for wpass and 0.5 for wstop. In the Pass band frequency control and Stop band frequency control.

4- Enter 1dB for pass band ripple and 80 dB for stop band ripple. In the Pass band control and Stop band control.

5-After setting the design specifications, click the amplitude response of FIR filter button the upper right of the GUI to design the filter. The amplitude response of the filter and the poles zeros is displayed in the Filter Analysis area after the coefficients are computed as shown in figure 5 .

Designing High Pass Filter

A high pass filter will be designed that passes all frequencies greater than or equal to the Nyquist frequency (Half the sampling frequency) and attenuates frequencies less than or equal to 20% of the Nyquist frequency. An IIR Cheby typ1 high pass filter will be designed with these specifications:

Pass band attenuation 2 dB, Stop band attenuation 60 dB, A pass band frequency 0.2 [Normalized (0 to 1)], A stop band frequency 0.5 [Normalized (0 to 1)]. To implement this design, the following specifications will used:

1-Select High pass filter from the Filter Type control and Cheby1 method from IIR Filter Method control.

2- Select Normalized (0 to 1) in the Pass band frequency control and Stop band frequency control.

3- Enter 0.2 for wpass and 0.5 for wstop. In the Pass band frequency control and Stop band frequency control.

4- Enter 2 dB for pass band ripple and 60 dB for stop band ripple. In the Pass band control and Stop band control.

5-After setting the design specifications, click Amplitude response of the IIR filter button the upper of the GUI to design the filter. The amplitude response of the filter and the poles zeros is displayed in the Filter Analysis area after the coefficients are computed as shown in figure 6. The number of circles in the axes of pole zeros determine order of the filter.

4. Results:

The purpose of filter design is to perform frequency dependent alteration of a data sequence. A possible requirement might be to remove noise above 300 Hz from a data sequence sampled at 10000 Hz. A more rigorous specification might call for a specific amount of pass band ripple, stop band attenuation, or transition width. A very precise specification could ask to achieve the performance goals with the minimum filter order, or it could call for an arbitrary magnitude shape, or it might require an FIR filter. A Butterworth IIR filter is often sufficient. To design a third -order 300 Hz low pass Butterworth filter and apply it to the data in vector x :

```
[M,Wn] =buttord(Wp,Ws,Rp, Rs);      (5)
[b,a]=butter(M,Wn);                 (6)
Filter(b,a,x);                      (7)
```

The upper left of the gui control panel in Figure 7 shows a sinusoidal wave

Corrupted with random noise. A low pass filter with about 300 Hz cut off frequency is applied to the wave, the noise is cancelled from the wave as shown in the left bottom of the GUI control panel.

speech processing using filter:-

Speech is a very basic way for humans to convey information to one another with a bandwidth of only another with a bandwidth of only 4 kHz; speech can convey information with the emotion of a human voice.

The speech signal has certain properties: It is a one-dimensional signal, with time as its independent variable, it is random in nature, it is non-stationary, and i.e. the frequency spectrum is not constant in time. Although human beings have an audible frequency range of 20Hz to 20 kHz, the human speech has significant frequency components only up to 4 kHz. The most frequent problem in speech processing is the consequence of interference noise in speech signals. Interference noise in a certain way modulates the speech signal and reduces its clearness. Interference noise can be produced from acoustical sources such as ventilation equipment, echoes, crowds. Noise cancellation is a special case of filtering which can be applied when some information about the reference noise signal is available [9]. Figure 8 shows speech signal processed using butter worth low pass filter but with cut off frequency of about 800Hz and of order 3.

5. Conclusions

Digital filters have been the most frequent application of digital signal processing. By digitizing any system, one is ensured that the signal can be reproduced with exactly the same characteristics. The main advantages of digital filters are being used instead of analog filters are the following: it is possible to reprogram the DSP, and one can update the filter coefficients while the program is running. Two different filter structures can easily be implemented. These two structures are Infinite Impulse Response (IIR) and Finite Impulse Response (FIR). IIR filters are commonly implemented using a feedback (recursive) structure, while FIR filters usually require no feedback (non-recursive) [3].This paper discusses the use of Mat lab, a mathematical software package, to design, manipulate, and analyze digital
filters. This paper concludes of how the data given by Mat lab for various filters can be used to implement filters on real digital signal processors. A visual screen is designed involving all tools needed to display the signals in the frequency domain. The Mat lab swing GUI was used to facilitate the interpretation through graphical displays and parametric controls of various signal-processing coefficients

References :

[1] 0. Herrmann, “Design of nonrecuffive digital filters with linear phase ,” Electron Lett., vol. 6, no. 11, pp. 328-329, 1970.

[2] Designing Filters Using the Digital Filter Design Toolkit Rahman Jamal, Mike Cerna, John Hanks 1996

[3] Digital Filter Design Using Matlab By Timothy J. Schlichter EE 4000 Introduction to Digital Filtering 5/2/99

[4] The Math Works, Inc. "Mat lab the Language of Technical Computing",

[5] Version 6.5,2002,MA,USA.[5] Practical FIR Filter Design in MATLABR Revision 1.1 Ricardo A. Losada The MathWorks, Inc.3 Apple Hill Dr. Natick, MA 01760, USA January 12, 2004

[6] [6] An Introduction to Digital Filters Application Note 9603 january 1999

[7] [7] FIR Filter Design: The Window Design Method, by Nguyen Huu Phuong

[8] [8] Design and implementation of Digital FIR Equiripple Notch Filter on MAHESH S. CHAVAN, * RA.AGARWALA, ** M.D.UPLANE.

[9] [9] School of Mathematics and Systems Engineering Reports from MSI - Rapporter från MSI Implementation of the LMS Algorithm for Noise Cancellation on Speech.

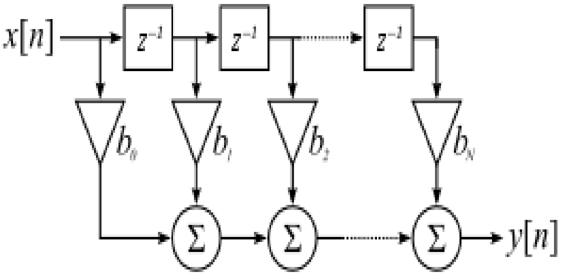


Fig. 1 FIR Filter

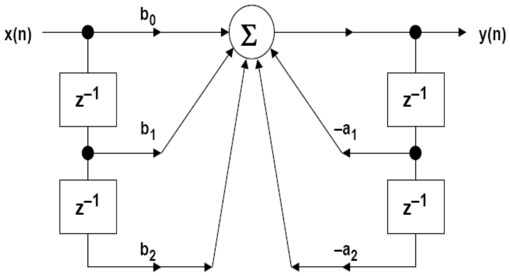
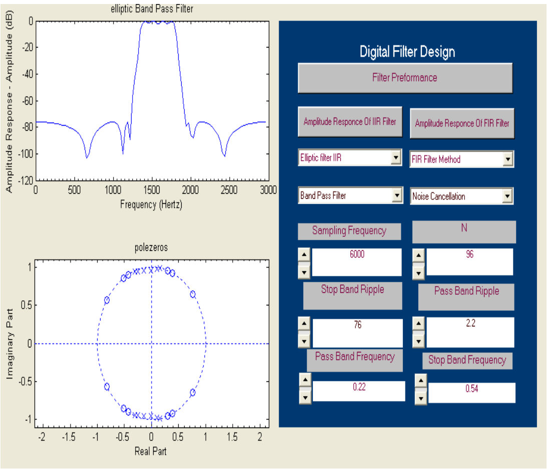


Figure. 2 IIR Filter



Elliptic filter IIR

FIR Filter Method

Band Pass Filter

Noise Cancellation

Sampling Frequency: 6000

N: 96

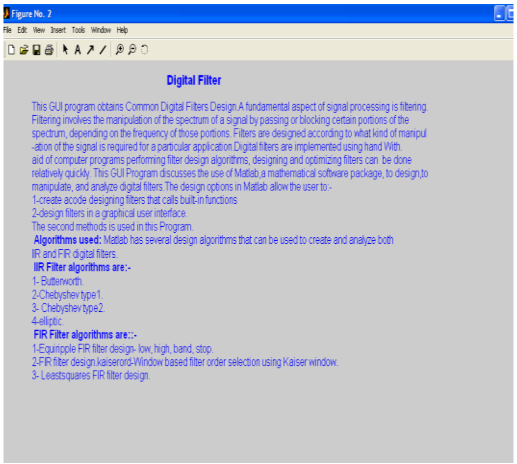
Stop Band Ripple: 76

Pass Band Ripple: 2.2

Pass Band Frequency: 0.22

Stop Band Frequency: 0.54

Fig. 3



Digital Filter

This GUI program obtains Common Digital Filters Design.A fundamental aspect of signal processing is filtering. Filtering involves the manipulation of the spectrum of a signal by passing or blocking certain portions of the spectrum, depending on the frequency of those portions. Filters are designed according to what kind of manipul -ation of the signal is required for a particular application.Digital filters are implemented using hand With. aid of computer programs performing filter design algorithms, designing and optimizing filters can be done relatively quickly. This GUI Program discusses the use of Matlab,a mathematical software package, to design,to manipulate, and analyze digital filters.The design options in Matlab allow the user to:-

1-create acode designing filters that calls built-in functions

2-design filters in a graphical user interface.

The second methods is used in this Program.

Algorithms used: Matlab has several design algorithms that can be used to create and analyze both IR and FIR digital filters.

IIR Filter algorithms are:-

1- Butterworth.

2-Chebyshev type1.

3- Chebyshev type2.

4-elliptic.

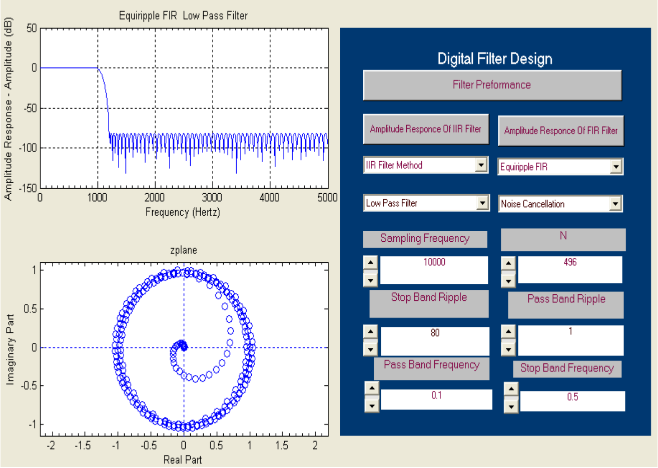
FIR Filter algorithms are::-

1-Equiripple FIR filter design- low, high, band, stop.

2-FIR filter design.kaiserord-Window based filter order selection using Kaiser window.

3- Leastsquares FIR filter design.

Fig. 4



IIR Filter Method

Equiripple FIR

Low Pass Filter

Noise Cancellation

Sampling Frequency: 10000

N: 496

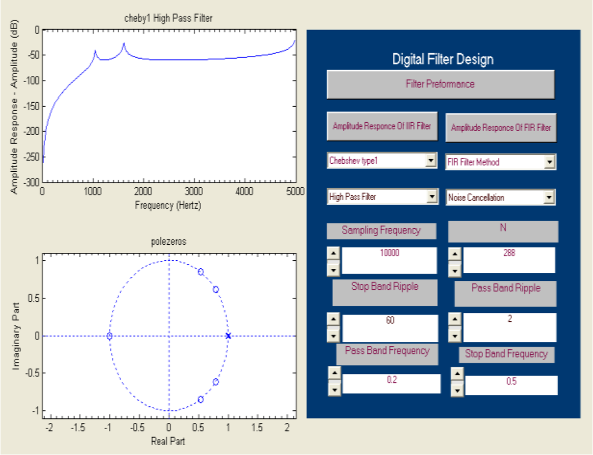
Stop Band Ripple: 80

Pass Band Ripple: 1

Pass Band Frequency: 0.1

Stop Band Frequency: 0.5

Fig. 5 Equiripple FIR low pass Filter



Chebshev type1

FIR Filter Method

High Pass Filter

Noise Cancellation

Sampling Frequency: 10000

N: 288

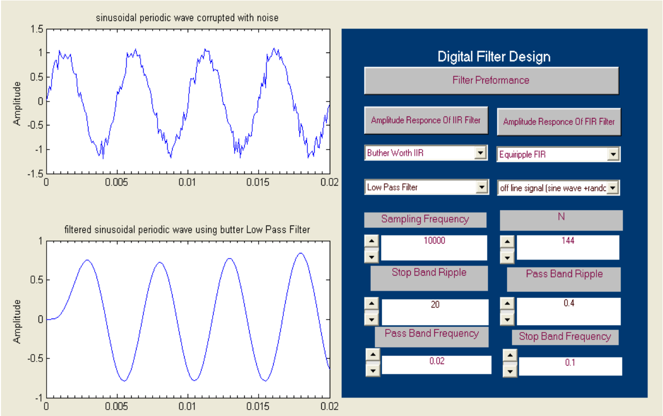
Stop Band Ripple: 60

Pass Band Ripple: 2

Pass Band Frequency: 0.2

Stop Band Frequency: 0.5

Fig. 6 Cheby type 1 high pass filter


sinusoidal periodic wave corrupted with noise
Amplitude
1.5
1
0.5
0
-0.5
-1
-1.5
0
0.005
0.01
0.015
0.02
filtered sinusoidal periodic wave using butter Low Pass Filter
Amplitude
1
0.5
0
-0.5
-1
0
0.005
0.01
0.015
0.02
Digital Filter Design
Filter Preformance
Amplitude Responce Of IIR Filter
Amplitude Responce Of FIR Filter

Buther Worth IIR

Equipple FIR

Low Pass Filter

off line signal (sine wave +randc

Sampling Frequency: 10000

N: 144

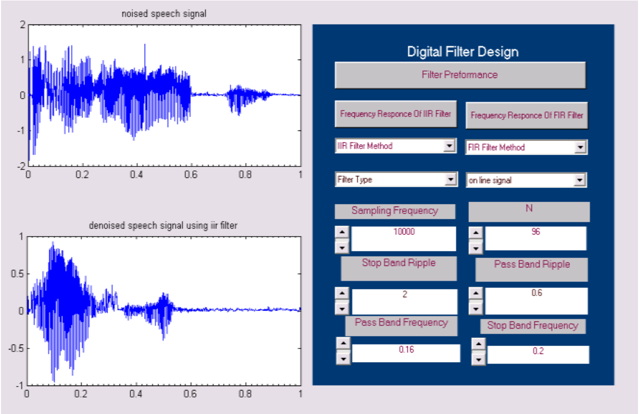
Stop Band Ripple: 20

Pass Band Ripple: 0.4

Pass Band Frequency: 0.02

Stop Band Frequency: 0.1

Fig. 7



IIR Filter Method

FIR Filter Method

Filter Type

on line signal

Sampling Frequency: 10000

N: 96

Stop Band Ripple: 2

Pass Band Ripple: 0.6

Pass Band Frequency: 0.16

Stop Band Frequency: 0.2

Fig. 8 speech signal filtered using butter low pass filter