A LEAST MEAN SQUARE BASED METHOD OF LOWPASS FIR FILTER DESIGN USING THE MATLAB TOOLBOX

Kwaha B.J, Gyang B.N & Amalu P.C
Department of Physics, University of Jos, Jos.
+Department of Physical Sciences, Ajayi Crowther University, Oyo.

ABSTRACT

A lowpass digital filter was designed using the MATLAB toolbox. The programme code was based on the least mean square error criterion method and was run in two steps – viz- filter order estimation from given specifications, and determination of the coefficients of the transfer function using the estimated order and filter specifications. The amplitude response of the linear phase FIR filter was used to obtain a realizable filter transfer function for filter length, N = 40. It was observed that for a longer length the FIR filter has a sharper cut off FIR filter and a narrow transition band. This makes it suitable for processing baseband signals especially digitized voice, music and data signals. All tests were carried out with the aid of the command window, which was unlocked from the workspace desktop for easy accessibility.

Keywords: *Lowpass FIR Filter Design on MATLAB*

1. INTRODUCTION

Mitra [1] describes a filter as a frequency selective device in the sense it functions by accepting an input signal, blocking off specified frequency components and then passing the original signal minus those components to the output. For instance, a phone limits frequencies to a range considerably lower than the human audible range [2]. This is why listening to CD-quality music over the phone is not as pleasing to the ear as listening directly to it.
Kaiser [3] and Rabiner [4]subdivided filters into two groups - *Analog Filters* and *Digital filters*, which are categorized into low pass, high pass, band pass and band stop. Smith [5] outlines two important uses of digital filters – viz *Separation of signals that have been combined with noise* and *restoration of signals that have been distorted in some way*. Analog filters can be used for these same tasks but with inferior results . A simplified low pass filter (LPF) is such that signal passes through an inductor and a shunted capacitor provides a path to earth. This way it admits frequencies from dc to a specified cutoff, f_1 and rejects all others. And if the time constant of the circuit is T' then the gain is given as

$$A = \frac{V_{out}}{V_{in}} = \frac{1}{1 + j\omega T'} \qquad (1)$$

where V_{out} is the output voltage, V_{in} the input voltage and ω the angular frequency[5] . $w = {}^{1}\!/_{T}$, then $\left| {}^{V_o}\!/_{V_i} \right|$ = 0.707
The need to have a versatile and flexible tool in the design and implementation of digital filters and the rapid evolution in the field of computing and the availability of various digital hardwares at relatively cheap costs with high performances coupled with the advantages derivable from DSP stimulated this research. It is intended that a software package based on *least mean square method* using the MATLAB tool is used to design and implement a finite input response (FIR) low pass digital filter.

2. DESIGN APPROACH

The method considered in this work is based on the observation that, for a length – N FIR digital, N distinct equally spaced frequency samples of its frequency response constitute the N – point DFT of its impulse response and hence, the impulse response sequence can readily be computed by applying an inverse DFT to the frequency samples taken [1]. Identical discrete time sequences may result from the sampling of more than one distinct continuous time function. This makes it possible to relate a unique continuous time signal to a given discrete time sequence, and also possible to recover the original continuous time signal from its sampled values [1]
Smith [5] and Rabina [4] gave the transfer function of a digital filter as

$$H(z) = {}^{Y(z)}\!/_{X(z)} \qquad (2)$$

where $X_{(z)}$ is the z-transform of the input signal $x_{(n)}$ and $Y_{(z)}$ is the z transform of the output signal $y_{(n)}$.

$$Y_{(z)} = X_{(z)} + g_1z^{-M1}X_{(z)} - g_2z^{-M}{}_2Y_{(z)} \qquad (3)$$

$$Y_{(z)} + g_2z^{-M2}Y_{(z)} = X_{(z)} + g_1z^{-M1} X_{(z)}$$

$$H_{(z)} \cong \frac{Y_{(z)}}{X_{(z)}} = \frac{1+g_1z^{-M_1}}{1+g_2z^{-M_2}} \qquad (4)$$

Equation (4) gives the transfer function of the digital filter. Kale [3]and Mitra [1] gave the Discrete Time Fourier Transform X(e^{jw}) of an input signal sequence x(n) as

$$X\left(e^{j\omega}\right) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} \qquad (5)$$

where X(e^{jw}) is a complex function of the real variable w.

3. DIGITAL FILTER RESPONSES

A filter has an impulses response, a stop response and a frequency response [3][5] . Each of these responses contains complete information about the filter, although in a different form. If one of these three responses is specified, the other two are fixed and can be directly calculated.

In most frequency domain applications frequency response is most important because it includes the pass band, stop band and the transition band. The impulse response is used in implementing a digital filter. To create a finite impulse response (FIR), the impulse response h(n) is convolved with the input signal x(n) using[3][1] .

$$Y(n) = x(n) \quad h(n) = \sum_{\bar{2}=1}^{m-1} x(n-i) \; h(i) \qquad (6)$$

where M is the number of points used to express the convolution. In synthesizing FIR digital filters, the following specifications are essential [3][1] .

(i) pass band edge angular frequency, F_p

(ii) stop band edge angular frequency, F_s

(iii) peak pass band ripple, δ_p

(iv) sampling frequency, F_r

These expressions are used to obtain the normalized form of F_p and F_s

$$\omega_p = \frac{\Omega_p}{F_T} = \frac{2\pi F_p}{F_T} = 2\pi F_p T \qquad (7)$$

$$\omega_s = \frac{\Omega_s}{F_T} = \frac{2\pi F_s}{F_T} = 2\pi F_s T \qquad (8)$$

The pass band edge and the stop band edge frequencies are specified in Hertz (Hz), along with the sampling rate of the digital filter. The filter design technique is developed in terms of normalized angular frequencies ω_p and ω_s. Rabiner [4] and Mitra [1] presented an expression to determine the peak pass band ripple α_p and the minimum stop band attenuation α_s.

$$\alpha_p. = -20\log_{10} (1-\delta_p) \text{ dB} \qquad (9)$$

$$\alpha_s. = -20\log_{10} \ \delta_s) \text{ dB} \qquad (10)$$

Kaiser (1976) developed formula to determine the filter order.

$$N \cong \frac{20\log_{10}\left(\sqrt{\delta_p\,\delta_s}\right) - 13}{14.6\left(\omega_s - \omega_p\right)/2\pi} \qquad (11)$$

4. DESIGN SPECIFICATIONS AND METHOD

Pass band edge frequency, F_p = 1.8kHz Stop band edge frequency, F_s = 2.0kHz

Pass band ripple, δ_p = 0.0114469 Stop band ripple, δ_s = 0.01778273

Sampling rate, F_T = 12kHz.

Given, F_p = 1.8kHz, F_s = 2.0kHz, δ_p = 0.0114463, δ_s = 0.01778279 and F_T = 12kHz

$\alpha_p = -20 \log_{10}(1-\delta_p)$ dB = -0.099999953dB ≅ 0.1dB.

Also, $\alpha_s = -20 \log_{10}(\delta_s)$ dB = 35.000002dB ≅ 35dB

And from equations (7) and (8), ω_p = 0.9425 and ω_s = 1.0472. And from equation (11), the filter order is = 98

The MATLAB method using Least Squares Error approach in designing a digital FIR low pass filter involves two steps namely – viz- filter order estimation from given specifications and determination of the coefficients of the transfer function using the estimated order and filter specifications [3][1][6]

The specifications were used with the M-files in the signal processing toolbox of the MATLAB software. The M-files consists of data and parameters that each filter type can be associated with. The low pass filter has its own M-files and specifications, which could be stretched to suit a given purpose, provided it suits the baseband range. The M-files are as follows:

b = firls (N, fpts, mag) b = firls (N, fpts, mag, 'f type')
b = firls (N, fpts, mag wt) b = firls (N, fpts, mag, wt, 'f type')

These are the various options available used for the function firls (an acronym for Finite Impulse Response – Least Squares) in the toolbox of MATLAB.

4.1 FIR Type 1

FIR 1 is of even order. The following specifications were used to generate a linear phase FIR 1 filter type. F_p = 800Hz, α_p = 0.5dB, F_s = 1000Hz, α_s = 40dB, F_T = 4000Hz, δ_p = 0.0559 and δ_s = 0.01. The input data for the program are N = 40, fpts = [0 0.05 0.4 1], mag = [1 1 0 0] and ω = [1 1]

4.2 FIR Type 2

FIR 2 is of odd order. The following specifications were used to generate a linear phase FIR 2 filter. F_p = 800Hz, α_p = 0.5dB, F_s = 1000Hz, α_s = 40dB, F_T = 4000Hz, δ_p = 0.0559 and δ_s = 0.01. The input data for the program are N = 40, fpts = [0 0.05 0.4 1], mag = [1 1 0 0] and ω = [1 1].

5. LEAST MEAN SQUARES ERROR METHOD

The Least – Squares Error method employs the use of a minimum mean square error criterion given as ;

$$\varepsilon = \sum_{i=1}^{k} \left\{ W(\omega_i) \left[\overline{H}(\omega_i) - D(\omega_i) \right] \right\}^2 \qquad (12)$$

where $\overline{H}$ (ω) is the amplitude response of the designed filter, D(ω) is the desired amplitude response, W(ω) is the weighting function. The amplitude response for a linear phase FIR filter is expressed as[4][1] .

$$\overline{H}(\omega) = Q(\omega) \sum_{k=0}^{L} \overline{a}[k] \cos(\omega k) \qquad (13)$$

This is used to obtain a realizable transfer function. Firls designs a linear phase FIR filter that minimizes the weighted, integrated squared error between an ideal linear function and the magnitude response of the filter over a set of derived frequency bands [5] .

Table 1. Filter Specifications

N	α_P	α_S	δ_P	δ_S	F_P	F_S	F_T
	(dB)	(dB)			(kHz)	(kHz)	(kHz)
40	0.5	40	0.059	0.01	0.8	1.0	4.0
37	0.5	40	0.059	0.01	0.8	1.0	4.0
30	0.5	40	0.059	0.01	0.8	1.0	4.0
27	0.5	40	0.059	0.01	0.8	1.0	4.0

The magnitude (Gain (dB)) responses for the specifications for N = 40 is given in Fig. 1

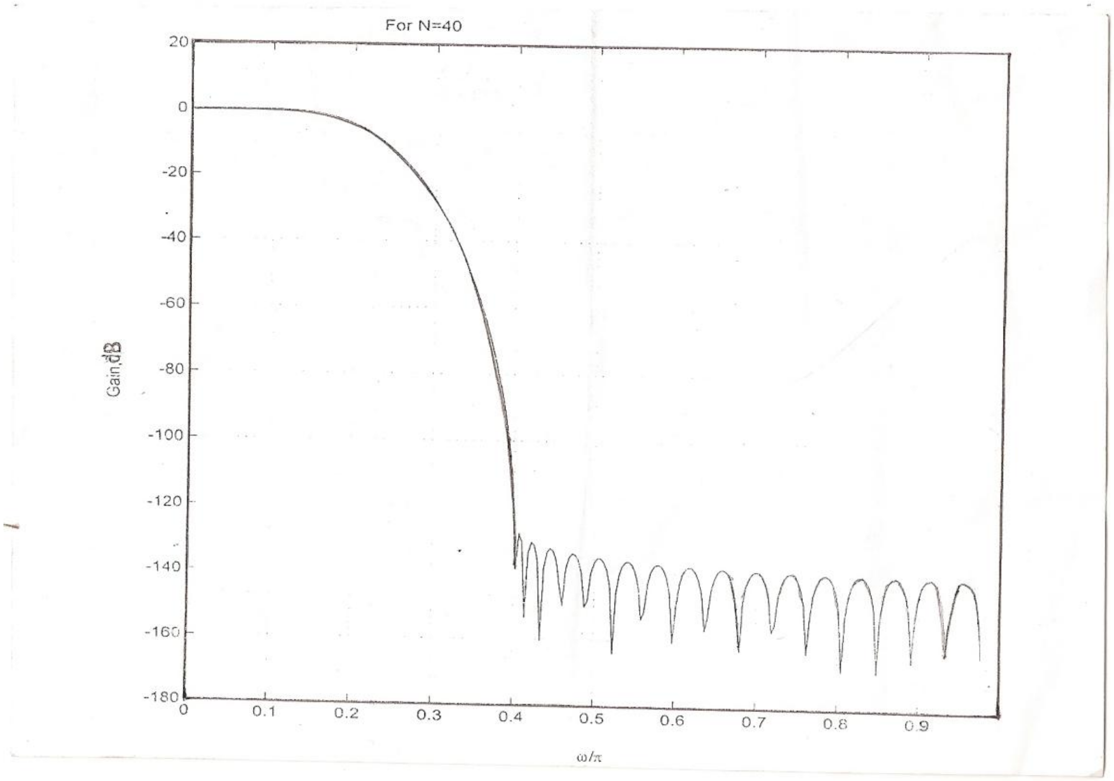

Fig. 1

6. DISCUSSION

In the design of this lowpass digital filter using MATLAB signal processing toolbox, the strings for the low pass filter type using the Least Square Error algorithm was studied closely before developing the code. From the design, the filter length is inversely proportional to the transition bandwidth and does not depend on the actual location of the transition band. This implies that a sharp cut off FIR filter with a narrow transition band would be of a very long length, whereas a filter with a wide transition band will have a very short length. This is quite suitable in processing baseband signals especially digitized voice and data signals.

All tests were carried out with the aid of the command window, which was unlocked from the workspace desktop for easy accessibility.

7. CONCLUSION

The running and debugging of the program was carried out several times, before it finally reached the final stage of presentation. A digital low pass filter was designed using the MATLAB software and its gain (dB) response was determined. It was observed that higher order, FIR low pass filters usually give a cleaner pass band (cleaner in the sense that the ripples are usually minimized) than lower order FIR filters, thereby making higher order FIR filters much more stable.

8. REFERENCES

[1]. Mitra, S. K. (2002). Digital Signal Processing – A Computer based approach. Second Edition, New York, USA: McGraw-Hill Irwin

[2]. Wu, O. (2004). Biomedical Signal and Image Processing. Spring 2001. HST-582J/6.555J/16.456J.

[3]. Kale, I. (2004). www.http.Electronics>Filter DES CMSA Web Based Filter Designer. Htm

[4]. Rabiner L.R. (1978) Linear Programming and Design of FIR Filters IEEE Proceeding Vol.20.

[5]. Smith, W. S. (2003)Digital Filter Design. www.wikepedia.org/wiki/electronic

[6]. Kaiser J.F.(1976) Digital Filter:- System Analysis By Digital Computers . McGraw Hill Inc.