Indian Journal of Science and Technology Vol. 4 No. 4 (April 2011) ISSN: 0974- 6846

The design and implementation of a digital infinite impulse response (IIR) Lowpass Butterworth filter- A comparison of Matlab and Bilinear transformation methods

B. J. Kwaha[1], E.A. Kolawole[1], and A. M. Batu[2]
[1]Dept. of Physics, University of Jos, Jos. Nigeria; [2]Dept. of Science and Technology, Kaduna Polytechnic, Nigeria
bold2k4@yahoo.com; kwahab68@yahoo.com

Abstract

This research compares two IIR digital filter design and implementation methods- viz MATLAB direct method and Bilinear transformation method. Programs to design and implement Butterworth IIR lowpass digital filters were developed and executed using both methods. In the design, at F_{pass} of 1Hz, F_{stop} of 2.414 Hz, using MATLAB method the 3 dB frequency is at 1.605 Hz while using the Bilinear transformation method the 3 dB frequency is at 1.395 Hz. At F_{pass} of 0.7839Hz, F_{stop} of 2.007 Hz, using MATLAB method the 3 dB frequency is at 1.181 Hz while using the Bilinear method the 3 dB frequency is at 1.001 Hz. From the design it was observed that the Butterworth filter has maximally flat passband in its frequency response and a poor roll-off at its bandstop which is in line with predicted theory. The Butterworth filters designed using Bilinear transformation were more stable with a monotonically decreasing gain response due to its poor roll-off which may be associated to discrete component performance. The magnitude of the passband decreases as the order of the filter increases. The designs also show that higher order filters have sharper skirts and may be used in phase modulated wave (PMW) applications.

Keywords: Matlab, Bilinear transformation methods, filter design.

Introduction

A filter is a device that passes certain frequency components in a signal spectrum through a system without any distortion and blocks other frequency components (Mitra, 1998). Filters perform direct manipulations on the spectra of signals and are basically classified into digital and analog filters (Liman, 1996). 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, and then a mapping from s to z plane 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 are (Smith, 2006):

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{n=0}^{M} b_n z^{-n}}{\sum_{n=0}^{M} a_n z^{-n}} = \frac{b_0 + b_1 z^{-1} + \dots\dots\dots + b_{M^z} - M}{a_0 + a_1 z^{-1} + \dots\dots\dots + a_{N^z} - N}, a_0 = 1 \qquad (1)$$

From this expression, the output difference equation y(n) can be obtained as:

$$y(n) = \sum_{m=0}^{M} b_m\, x(n-m) - \sum_{n=0}^{N} a_m\, y(n-m) \qquad (2)$$

The frequency domain response function can be obtained from its z-transform (Smith, 2006):

$$Y(z) = X(z)\sum_{k=0}^{N} b_k z^{-k} + Y(z)\sum_{k=1}^{M} a_k z^{-k} \qquad (3)$$

Then the transfer function is given by

$$H(z) \equiv \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{N} b_k z^{-k}}{1 - \sum_{k=1}^{M} a_k z^{-k}} \qquad (4)$$

A digital filter is programmable, easily designed, tested and implemented, extremely stable with respect to both time and temperature, can handle low frequency signals accurately and are more versatile in their ability to process signals in a variety of ways.

IIR Butterworth filter design methods

Bilinear transformation method

The bilinear transformation is a nonlinear mapping of the continuous s-domain to the discrete z-domain given by z = (k+s)/(k-s) (Mitra, 1998). The transfer function H(s) of the ideal analogue filter is mapped to the transfer function H(z) of a corresponding discrete filter. By default, the frequency warping constant k is set equal to twice the sampling frequency. The bilinear transformation method involves deducing a continuous-time transfer function using classical methods. A discrete-time transfer function is obtained by applying the bilinear transformation to the continuous-time transfer function. The exact relation between the imaginary axis in the s-plane (s = jΩ) and the unit circle in the z-plane ($z = \lambda^j$) is of interest. With T representing the step size in the numerical integration, = 2, it follows that (Mitra, 2002):

$$j\Omega = \frac{1 - \lambda^{j\omega}}{1 + \lambda^{j\omega}} = j\tan\left(\omega/2\right) \qquad (5)$$

Where

$$\Omega = \tan\left(\omega/2\right)$$

Direct IIR design

Unlike the analogue prototyping method, direct designs are not constrained to the standard lowpass, high-pass, bandpass or bandstop configurations. Filters with an arbitrary, perhaps multiband, frequency response are possible. Mitra (1998) reports that the signal processing toolbox order selection functions estimate the minimum filter order that meets a given set of requirements. There are various direct IIR design methods e.g. the Yule walk method.

Butterworth filter design procedure

Usually, the specifications are given as:

1) Ω_p the pass-band edge,
2) Ω_s the stop-band edge,
3) $|H_a(j\Omega_p)|=\frac{1}{\sqrt{1+\varepsilon^2}}$, the maximum pass-band attenuation or
4) $|H_a(j\Omega_s)|=1/A$, the minimum stop-band attenuation or ripple

These 4 pieces of known information must be used to compute filter order n, 3 dB cut-off frequency Ω_c and filter transfer function $H_a(j\Omega)$. The Matlab signal processing toolbox function Buttord uses the Matlab definitions as: Wp for the pass-band edge in radians/second, Ws for the stop-band edge in radians/second, R_p for the maximum pass-band ripple (the attenuation, in dB, at W_p), R_s for the minimum stop-band ripple (the attenuation, in dB, at Ws). Filter order n and 3dB down frequency W_c will be computed with the command-line statement

$$[n, W_c] = buttord(W_p, W_s, R_p, R_s, 's')$$

Specifications & calculations using MATLAB method

Using Matlab, the low-pass digital filter is designed using the analogue prototype: Butterworth. The optimum filter type is chosen on the basis of implementation complexity, magnitude response, and phase response. The design specifications for the filter are: Pass band edge frequency, F_p = 1 Hz; Stop band edge frequency, F_s= 2.414 Hz; Sample frequency , F_t= 7 Hz; Pass band ripple, $\square_p$= 0.5 dB; Stop band ripple $\square_s$ = 20 dB; Transition band = 100 Hz.

To obtain the normalized angular frequencies, we use the following formula:

W_p = Cutoff frequency÷Sample frequency (6)

W_s= Stop band edge frequency÷Sample frequency (7)

Substituting in values from our specifications into the equation 6 and 7 for

W_p= 0.1429 and W_s= 0.3449

With these specifications, MATLAB is employed to compute the order and the F3db of the filter to be designed. The order and F3 db for the program whose source code is in appendix B are

N=3 and F3 db=1.605 Hz

For the Matlab source code in appendix C, the frequency required is 1 kHz and to show the plot, the sample frequency ranges from 0 to 8 kHz to allow the magnitude response to be visible. Magnitudes below 20 dB should not be plotted and a pass-band ripple of 0.5 dB should allow for realistic simulation.

Specifications and calculations using Bilinear transformation method

Step1- Pre-warp the frequencies:

Using T=1s and θ_1=0.5π and θ_2=0.75π

The pre-warped frequencies are from equation (5):

$\Omega_1=\tan(\theta_1/2) = \tan(0.5\pi/2)=1.0$

$\Omega_2=\tan(\theta_2/2) = \tan(0.75\pi/2)=2.4141$

Step 2- Design an analogue LP filter that satisfies the following criteria

$K_1=20 \log/H(j\,\Omega_1)/\geq -3.01$ dB

$K_2=20 \log/H(j\,\Omega_2)/\geq -20$ dB

The filter Butterworth filter order can be calculated as follows:

$$n= \frac{\log 10\,(10-k1/10-1)/(10-k2/10-1)}{2\log 10\,(\Omega 1/\ \Omega 2)} = 2.6\approx 3 \quad (8)$$

Step3- Obtain H(s) from appendix A for n= 3

$$H(s) = \frac{1}{(P^2+P+1)(P+1)} \quad (9)$$

This is the normalized polynomial. Since Ω_1=1.0 the polynomial need not be de-normalized.

Converting H (s) to the equivalent H (z) (Mitra, 2002)

$$H(s) = \frac{1}{(P^3+2P^2+2P+1)} \quad (10)$$

and by manipulation (Mitra, 2002)

$$\frac{Y(Z)}{X(Z)} = \frac{1+3Z^{-1}+3Z^{-2}+Z^{-3}}{6+2Z^{-2}} \quad (11)$$

Cross multiplying equation 11

$$Y(z)(6+2z^{-2})= (1 + 3z^{-1}+3z^{-2}+z^{-3})X(z) \quad (12)$$

Applying the inverse z- transform we obtain (Mitra, 2002)

6y[n] +2y [n-2] = x[n] +3x [n-1] +3x [n-2] +x [n-3] (13)

And the difference equation of the IIR filter is

Y[n]= 1/6x[n]+1/2x[n-1]+1/2x[n-2]+1/6x[n-3]-1/3y[n-2] (14)

The magnitude of the Frequency Amplitude Response is obtained from the transfer function H(z) defined by equation (11) and so:

$$H(z)=\frac{1+3Z^{-1}+3Z^{-2}+Z^{-3}}{6+2Z^{-2}}$$

The magnitude response is calculated replacing z with $z=e^{j\theta}$ in equation 11 and using Euler's identity: $e^{j\theta}$= cos (θ) +jsin (θ) we obtain:

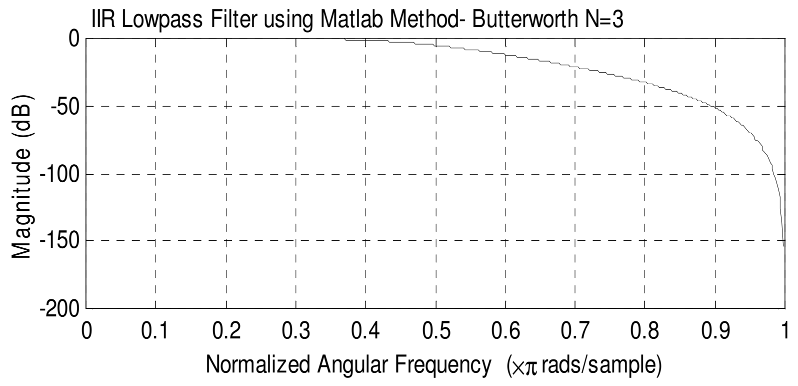
$$|H(ej^{\theta})|=\frac{\sqrt{(1+3\cos(\theta)+3\cos(2\theta)+\cos(3\theta))2+(3\sin(\theta)+3\sin(2\theta)+\sin(3\theta))2}}{\sqrt{(6+2\cos(2\theta))2+(2\sin(2\theta))2}} \quad (15)$$

Equation 11 can be implemented using the MATLAB program in appendix D.

Results and discussion

This research was based on the design and implementation of IIR Lowpass Butterworth digital filters through computation and simulations using Matlab

Fig 1a: Magnitude Response for N=3 using Matlab



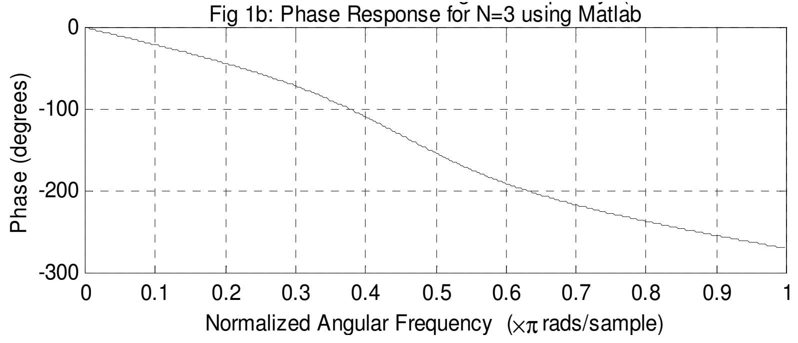
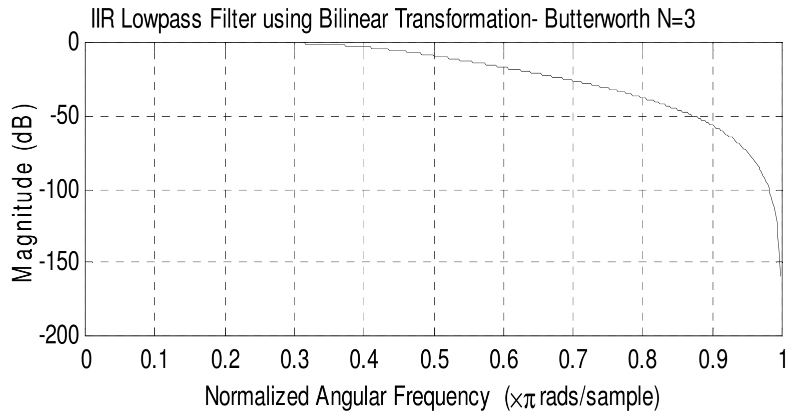


Fig 2: Magnitude Response for N=3 using Bilinear Transformation



method, Bilinear transformation methods and comparing them. Alongside, there were needs to determine some important parameters that give the order, 3dB points, magnitude and phase response curves, and the passbands that characterized the performance of the IIR digital filters. Programs written based on the necessary equations underlying the operations of the Butterworth filters to design and implement the IIR digital filters were developed and executed using MATLAB Toolbox. The programs were tried out several times to ascertain its flexibility and accuracy before presentation. Equation 15 gives the difference equation that analyses the operations of the digital filter. This equation allows us to analyze in the frequency and time domains. These derivatives terminate with the attainment of the digital filters transfer function [equation 11] which is used to determine the magnitude response of the digital filter. Also the derivations sequel to equations 16 and 17 were used to calculate the normalized angular frequencies for the digital filters designed by Matlab which are used to determine the magnitude and phase response of the filters. All other equations used in the development of these programs are related to theses ones (Kwaha, 2007).

At F_{pass} of 1Hz, F_{stop} of 2.414 Hz, using Matlab Method the 3 dB frequency is at 1.605 Hz while using the Bilinear method the 3 dB frequency is at 1.395 Hz. The Butterworth lowpass filters are used in heart monitoring equipment (ECG), music, voice transmission and audio processing. Butterworth filters designed using the direct Matlab and Bilinear transformation methods shown on Figures 1a and 2 both have maximally flat passband and a poor roll-off at bandstop which is in line with predicted theory (Ogunbuike, 1986).

Appendix B is a program written to display the response curves of the designed filter, appendix D is the Matlab source code for the designed filter n = 3 using bilinear transformation and appendix C the Matlab source code, the frequency required is 1 kHz and to show the plot, the sample frequency ranges from 0 to 8 kHz to allow the magnitude response to be visible. Magnitudes below 20 dB should not be plotted and a pass-band ripple of 0.5 dB should allow for realistic simulation.

Further work was done in the designs of IIR digital lowpass Butterworth filters for n = 2, 32 and 43 and it was seen that higher order filters have slopes with sharper skirts and may be applied in phase modulated waves (PMW) applications. From Table 1 & 2 of result, it was observed that as the filter order increases, the magnitude of the passband decreases.

Summary and conclusion

A fundamental aspect of signal processing is filtering. With the aid of computer programs performing filter design algorithms, designing and optimizing filters can be done relatively quickly. This research was based on the design and implementation of IIR Lowpass Butterworth digital filters using Matlab and Bilinear transformation methods through computation and simulation and comparing both methods. Programs to design and implement the IIR digital filters were developed and executed using MATLAB toolbox. The results obtained shows that the Butterworth filter has a maximally flat passband and a poor roll-off at bandstop which is in line with predicted theory. This makes it suitable for processing baseband signals in music and speech applications. The magnitude response of Fig. 1a & 1b and 2 with the same specifications using Matlab and Bilinear transformation methods have almost equal transition band with a monotonically decreasing gain response due to its poor roll-off which may be associated to discrete component performance. Fig. 2 is more stable as lower order filters designed using bilinear transformation are more stable than higher order filters as it is sensitive to quantization errors which affects the transfer function. This error occurs in filter order from as low as 8. Matlab method is preferred in higher order filters. These results show that in designing IIR low pass Butterworth filters Matlab method gives both the magnitude and the phase response curves whereas the bilinear transform method gives only the magnitude response curve.

Table 1. Specifications for a digital IIR Lowpass Butterworth filter using Matlab (n=3).

Filter specifications	Responses			Passband	
	Normalized angular Frequency (x□ rad/sample)	Magnitude (dB)	Phase (degrees)	Frequency (Hz)	Magnitude (dB)
$F_{p=\ 1Hz}$	0.1	0	-21.4286	0.2	0
$F_{s=\ 2.414Hz}$	0.	0	-42.8571	0.4	0
$F_{t=\ 7Hz}$	0.2	0	-75.7143	0.45	-0.0029
$□_{p=\ 0.5dB}$	0.4	-3.6000	-112.8571	0.5	-0.0044
$□_{s=\ 20dB}$	0.5	-7.2000	-155.7143	0.55	-0.0056
$\omega_{p=0.1429}$	0.6	-12.8000	-190.7143	0.6	-0.0071
$\omega_{s=\ 0.3449}$	0.7	-20.8000	-214.2857	0.65	-0.0094
F3dB=1.605Hz	0.8	-34.0000	-242.8571	0.7	-0.0115
	0.85	-40.0000	-248.5714	0.8	-0.0250
	0.9	-52.0000	-257.1429	0.85	-0.0382
	0.95	-70.8000	-265.0000	0.9	-0.0583
	1.0	-156.0000	-271.4286	1.0	-0.1200

Table 2. Specifications for a digital IIR Lowpass filter using Bilinear transformation (n=3).

Filter specifications	Responses		Passband	
	Normalized Angular frequency (xπ ad/sample)	Magnitude (dB)	Frequency (Hz)	Magnitude (dB)
$H(z)=\frac{(1+3z^{-1}+3z^{-2}+z^{-3})}{(6+2z^{-2})}$	0.1	0	0.2	0
$F_{p=\ 1Hz}$	0.2	0	0.4	0
$F_{s=\ 2.414Hz}$	0.2	-2.0000	0.45	0
	0.4	-4.8000	0.5	-0.0041
	0.5	-11.2000	0.55	-0.0044
	0.6	-17.2000	0.6	-0.0062
	0.7	-27.6000	0.65	-0.0091
	0.8	-39.6000	0.7	-0.0147
	0.85	-48.0000	0.8	-0.0232
	0.9	-56.8000	0.85	-0.0353
	0.95	-74.4000	0.9	-0.0357
	1.0	-159.6000	1.0	-0.0359

Indian Journal of Science and Technology Vol. 4 No. 4 (April 2011) ISSN: 0974- 6846

References

1. Kwaha BJ (2007) The development of digital filtering techniques for application in continuous wave (CW) radar systems. Ph. D thesis in the postgraduate school, University of Jos, Nigeria.
2. Liman MS (1996) Design and implementation of digital filters using windows. M.Sc. thesis in the post graduate school, University of Jos, Nigeria.
3. Mitra SK (1998) Digital signal processing (A computer based Approach), New York, NY: McGraw-Hill, 2nd edn.
4. Mitra SK (2002) Digital signal processing (A computer based approach). McGraw Hill book company, New York, 3rd edn.
5. Ogubuike SC (1986) Electrical filters. B. Sc., thesis, University of Jos, Nigeria.
6. Smith JO (2006) Introduction to Digital filters. www.http://ccrma.stanford.edu/ jos/filters06/.

Appendix A: Butterworth polynomials.

N	
1	(s + 1)
2	(s^2 + 1.4142s + 1)
3	(s + 1)(s2 + s + 1)
4	(s^2+ 0.7654s + 1)(s2 + 1.8478s + 1)
5	(s + 1)(s2 + 0.6180s + 1)(s^2 + 1.6180s + 1)
6	(s^2 + 0.5176s + 1)(s^2 + 1.4142s + 1)(s^2 + 1.9319)
7	(s + 1)(s^2 + 0.4450s + 1)(s^2 + 1.2470s + 1)(s^2 + 1.8019s + 1)
8	(s^2 + 0.3902s + 1)(s^2 + 1.1111s + 1)(s^2 + 1.6629s + 1)(s^2 + 1.9616s + 1)
9	(s+1)(s^2 + 0.3473s + 1)(s^2 + 1.0000s + 1)(s^2+ 1.5321s + 1)(s^2+ 1.8794s + 1)
10	(s^2+0.3129s+1)(s^2+0.9080s+1)(s^2+1.4142s+1)(s^2+1.7820s+1)(s^2+1.9754s +1)

Appendix B: Program written to display the response curves

```
A=filt1.tf.den;
B=filt2.tf.num;
Freqz(B, A)
Impz(B,A,100,8000)
```

Appendix C

Matlab Code (Butterworth):

```
% Lowpass digital filter with Butterworth analog prototype
%
% Digital Filter Specifications:
wp = wp*2*pi; % digital passband frequency in Hz (normalized)
ws = ws*2*pi; % digital stopband frequency in Hz (normalized)
Rp = 0.5; % passband ripple in dB
As = 20; % stopband attenuation in dB
% Analog Prototype Specifications:
Fs = 1; T = 1/Fs;
OmegaP = (2/T)*tan(wp/2); % prewarp prototype passband frequency
OmegaS = (2/T)*tan(ws/2); % prewarp prototype stopband frequency
% Analog Butterworth Prototype Filter Calculation:
[c, d] = butterworth(OmegaP, OmegaS, Rp, As);
% Bilinear Transformation:
[b, a] = bilinear (cs, ds, Fs);
%
[db,mag,pha,grd,w] = freqz(b,a);
plot(w*8000/2/pi,db);
xlabel('frequency (Hz)');
ylabel('decibels'); title('Magnitude in dB');
```

Appendix D

```
% Program Description:
% To design and realize an IIR Filter using the Bilinear
% transformation. The Butterworth LP analog filter has n=3
% The Transfer Function is:
% H(z) = (1 + 3z^-1 + 3z^-2 + z^-3)/(6 + 2z^-2)
%*****************************************************************
theta=0;
delta=pi/200;
i=1;
while theta < 0.9*pi
Num=
sqrt((1+3*cos(theta)+3*cos(2*theta)+cos(3*theta))^2 +
(3*sin(theta)+3*sin(2*theta)+sin(3*theta))^2);
Den = sqrt((6+2*cos(2*theta))^2 + (2*sin(2*theta))^2);
H(i)= 20*log10(abs(Num/Den));
w(i)=theta;
theta=theta+delta;
i=i+1;
end
%Plot the Frequency Response:
plot(w,H);
grid on;
title('IIR Filter using Bilinear Transformation - Butterworth N=3');
ylabel('Amplitude Response (dB)');
xlabel('Frequency (radians)');
%End of Program
```