Digital Butterworth LP IIR Filter Design Using the Bilinear Transformation Method

Author: Paul Gomez, PhD

October 22, 2001

Objective:

This document shows how to design and realize a digital LP filter using the Bilinear Transformation method with a specific example. Although we are implementing a specific filter, the method is generic. The frequency response will be plotted using two different scripts written in MATLAB and Python. The scripts are in the same folder where this document is placed, along with the resulting plot.

Problem: Design a LP filter to satisfy the following specifications:

1) Monothonic stopband and passband
2) –3.01dB cutoff frequency of 0.5π radians
3) Magnitude down at least 20dB at 0.75π radians

Solution:

1) A Butterworth LP filter accomplishes the monotonic stopband and passband requirement

Step 1. Pre-Warp the frequencies:

Using T=1sec and $\theta_1 = 0.5\pi$ and $\theta_1 = 0.75\pi$ the pre-warped frequencies are:

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

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

Step 2. Design an analog LP filter that satisfies the following criteria

$$k1 = 20\log\left|H(j\Omega_1)\right| \geq -3.01dB$$

$$k2 = 20\log\left|H(j\Omega_2)\right| \geq -20dB$$

The Butterworth Filter Order, n, can be calculated as follows:

$$n = \frac{\log_{10}\left[(10^{-k1/10}-1)/(10^{-k2/10}-1)\right]}{2\log_{10}(\Omega_1/\Omega_2)}$$

$$n = 2.6 \approx 3$$

Step 3. Obtain H(s) from the Butterworth Table for n=3:

$$H(s) = \frac{1}{(p^2+p+1)(p+1)}$$

which is the normalized polynomial. Since $\Omega_1 = 1.0$ we don't have to denormalize the polynomial.

Let's convert H(s) to the equivalent H(z):

$$H(s) = \frac{1}{p^3+2p^2+2p+1}$$

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

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

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

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

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

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

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

Applying the Inverse Z-Transform on both sides of the equation we obtain:

$$6y[n]+2y[n-2] = x[n]+3x[n-1]+3x[n-2]+x[n-3]$$

And the Difference Equation of the **IIR filter** is:

$$y[n] = \frac{1}{6}x[n]+\frac{1}{2}x[n-1]+\frac{1}{2}x[n-2]+\frac{1}{6}x[n-3]-\frac{1}{3}y[n-2]$$

Frequency Amplitude Response

The Magnitude Response is obtained from the Transfer Function H(z) defined by:

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

The magnitude response is calculated replacing $z = e^{j\theta}$ in the above equation, and using $e^{j\theta} = \cos(\theta) + j\sin(\theta)$ we arrive at:

$$H(e^{j\theta}) = \frac{1+3(\cos(\theta)-j\sin(\theta))+3(\cos(2\theta)-j\sin(2\theta))+(\cos(3\theta)-j\sin(3\theta)}{6+2(\cos(2\theta)-j\sin(2\theta))}$$

$$H(e^{j\theta}) = \frac{1+3\cos(\theta)-3j\sin(\theta)+3\cos(2\theta)-3j\sin(2\theta)+\cos(3\theta)-j\sin(3\theta)}{6+2\cos(2\theta)-2j\sin(2\theta)}$$

$$H(e^{j\theta}) = \frac{(1+3\cos(\theta)+3\cos(2\theta)+\cos(3\theta))-j(3\sin(\theta)+3\sin(2\theta)+\sin(3\theta))}{(6+2\cos(2\theta))-j(2\sin(2\theta))}$$

$$\left|H(e^{j\theta})\right| = \frac{\sqrt{(1+3\cos(\theta)+3\cos(2\theta)+\cos(3\theta))^2+(3\sin(\theta)+3\sin(2\theta)+\sin(3\theta))^2}}{\sqrt{(6+2\cos(2\theta))^2+(2\sin(2\theta))^2}}$$

The previous equation was implemented using MATLAB and Python. Both scripts are shown at the end of this document. The program also plots the Frequency Amplitude Response. The attenuation at frequency 0.5π radians is –3.01dB and the attenuation at frequency 0.75π radians is –22.98dB. The performance of the designed filter exceeds the requirements.

THE MATLAB code is:

```
%****************************************************************
% 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)
%
% Author: Paul Gomez, PhD
% Date: October 22, 2001
%****************************************************************
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
```

The Python script is shown below:

```
# ************************************************************
# Butterworth IIR LP Filter using the Bilinear Transformation
```

```
# Paul Gomez, PhD
# March 18, 2024
# The equations used in this script are explained in the design
# document located on the same folder as this script.
# ************************************************************

from scipy import signal # not used
import numpy as np
import matplotlib.pyplot as plt

theta=0;
delta=np.pi/200;

H = np.array(0.00)
w = np.array(0.00)

i=1;
while theta < 0.9 * np.pi:
   # filter numerator:
   Num = np.sqrt((1+3*np.cos(theta)+3*np.cos(2*theta)+np.cos(3*theta))**2 +
(3*np.sin(theta)+3*np.sin(2*theta)+np.sin(3*theta))**2)
   # filter denominator:
   Den = np.sqrt((6+2*np.cos(2*theta))**2 + (2*np.sin(2*theta))**2)
   # Amplitude response:
   H = np.append(H,20*np.log10(abs(Num/Den)) )
   w = np.append(w,theta)
   theta=theta+delta
   i=i+1


# Plot the Frequency Response:
fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
ax.plot(w, H)
ax.set_title('IIR Filter using Bilinear Transformation - Butterworth N=3')
ax.set_xlabel('Frequency [Radians]')
ax.set_ylabel('Amplitude [dB]')
ax.grid(which='both', axis='both')

plt.show()

# *** end of program ***
```