# Program – Butterworth Band-pass Filter Design

In the previous program, a filter bank of 10 $4^{th}$ order Butterworth band pass filters was designed. The filters, defined by their *b* and *a* coefficients, were designed using MATLAB's built in '*butter*' function. In this program, you write your own filter design function that replaces the call to the MATLABbuilt-in.

Like the previous programs, once you save your function with the name butter_bpf4.m in the same folder as the main program, your function should be picked up and used instead of the MATLAB function.

This document describes how to design a $4^{th}$ order Butterworth BPF. It does not have any code, but it should help you write your own.

**Requirement**: We need to design a digital $4^{th}$ order Butterworth band pass filter with cutoffs $\omega_{c1}$ and $\omega_{c2}$ (radians). I.e. we need to generate something of the form of

$$H(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + b_3 \cdot z^{-3} + b_4 \cdot z^{-4}}{a_0 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + a_3 \cdot z^{-3} + a_4 \cdot z^{-4}}$$

A function butter_bpf4 should be written that accepts cutoffs $\omega_{c1}$ and $\omega_{c2}$as arguments and returns the filter coefficients $a_0$ to $a_4$, and $b_0$ to $b_4$as vectors ***a*** and***b***.

```
function [ba] = butter_bpf4(wc1, wc2)
```

**Steps**:

1. Calculate analog filter characteristics by frequency warping (you may assume T=2):
   a. $\boldsymbol{\Omega_{c1} = \frac{2}{T} tan \frac{\omega_{c1}}{2}}$
   b. $\boldsymbol{\Omega_{c2} = \frac{2}{T} tan \frac{\omega_{c2}}{2}}$
   c. $\boldsymbol{\Omega_0 = \sqrt{\Omega_{c1} \cdot \Omega_{c2}}}$
   d. **BW =** $\boldsymbol{\Omega_{c2} - \Omega_{c1}}$

2. Since we need a $4^{th}$ order BPF, we need to start with a $2^{nd}$ order LPF:
   a. We know the normalized $2^{nd}$ order LPF is

   $$H_2(s) = \frac{1}{s^2 + \sqrt{2}s + 1}$$

   b. The transformation that converts a normalized LPF $H_1(s)$ to a band pass filter with the given characteristics is

   $$s_{LP} \leftarrow \frac{s^2 + \Omega_0^2}{s \cdot BW}$$

   c. This gives us a $4^{th}$ order BPF

   $$H'(s) = \frac{BW^2 s^2}{s^4 + \sqrt{2} \cdot BW \cdot s^3 + (BW^2 + 2 \cdot \Omega_0^2) \cdot s^2 + \sqrt{2} \cdot BW \cdot \Omega_0^2 \cdot s + \Omega_0^4}$$

d. Let us rewrite it in terms of symbols c, d, e, f and g for simplicity

$$H'(s) = \frac{c \cdot s^2}{s^4 + d \cdot s^3 + e \cdot s^2 + f \cdot s + g}$$

where $c = BW^2, d = \sqrt{2} \cdot BW, e = (BW^2 + 2 \cdot \Omega_0^2), f = \sqrt{2} \cdot BW \cdot \Omega_0^2, g = \Omega_0^4$.

**In your code, compute c, d, e, f, and g using the formulas obtained above.**

3. Convert the analog LPF to a digital one using the Bilinear transform (assuming T = 2)

a. Find H(z) as

$$H(z) = H'(s) \mid s \leftarrow \frac{1 - z^{-1}}{1 + z^{-1}}$$

Upon substitution and simplification, we get,

$$H(z) = \frac{c - 2 \cdot c \cdot z^{-2} + c \cdot z^{-4}}{(1 + d + e + f + g) + (-2 \cdot d + 2 \cdot f + 4 \cdot g - 4) \cdot z^{-1} + (6 \cdot g - 2 \cdot e + 6) \cdot z^{-2} + (2 \cdot d - 2 \cdot f + 4 \cdot g - 4) \cdot z^{-3} + (1 - d + e - f + g) \cdot z^{-4}}$$

b. Comparing H(z) with

$$H(z) = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + b_3 \cdot z^{-3} + b_4 \cdot z^{-4}}{a_0 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + a_3 \cdot z^{-3} + a_4 \cdot z^{-4}}$$

where$b_0 = c,\ b_1 = 0,\ b_2 = -2 \cdot c, \ldots, a_0 = (1 + d + e + f + g), a_1 = (-2 \cdot d + 2 \cdot f + 4 \cdot g - 4)$, etc..

**In your code, compute *b* and *a* coefficients using the formulas above. Remember, in MATLAB, $b_0$has to be stored in b(1), $b_1$ in b(2), and so on. Similarly, $a_0$ gets stored in a(1), etc.**

c. **Calculate K = 1/$a_0$. Multiply all coefficients $a_n$ and$b_n$ by K. This changes $a_0$ to 1 without changing the equation itself.** (It is the convention to force $a_0$ = 1, since it is the coefficient of y(n). This avoids a division by $a_0$ when calculating y(n).)

You have completed the design of a $4^{th}$ order Butterworth band pass filter. Your function returns the ***b*** and ***a*** coefficients obtained above.

**Verification**:

Let wc1 = 1 and wc2 = 2. Run your filter design function as:[b a] = butter_bpf4(wc1, wc2)

RunMATLAB's built-in Butterworthdesign functionas:[b1a1] = butter(2, [wc1 wc2]/pi)

Check that b1 and a1 should be identical to the b and a obtained from your function.

**Troubleshooting:**

If your coefficients don't match in the verification step, set wc1=1, wc2=2 and check all intermediary calculations by printing them out in your program:

For wc1=1 and wc2=2:

oc1 = 0.5463, oc2 = 1.5574, o0 = 0.9224, BW = 1.0111,

c = 1.0223, d = 1.4299, e = 2.7240, f = 1.2166, g = 0.7239

Before multiplication by K: b0 = 1.0223, b1 = 0, b2= -2.0447, b3 = 0, b4 = 1.0223

Before multiplication by K: a0 = 7.0944, a1 = -1.5311, a2 = 4.8954, a3 = -0.6778, a4 = 1.8013

After multiplication by K: b = 0.1441 0 -0.2882 0 0.1441

After multiplication by K: a = 1.0000 -0.2158 0.6900 -0.0955 0.2539

Don't hardcode these values into your program!

---

***END***

*Rohit Rawat*

*08/12/2013*