IIR filter : Butterworth digital filter

Samuel Dupont

Samuel.dupont.etu@univ-lemans.fr

November 25, 2016
Last update: November 28, 2016

Abstract

This article aim to do a recap to produce IIR filter. The butterworth filter type is one of most well known due to it's simplicity. After seeing how to generate the digital filter from the continuous form using bilinear transform, the article will presents how to implement it in biquad form. It allows a fast response of the filter using cascade of second order filters.

The matlab programs used to implement the different IIR (normal and biquad) filter can be found on the following link: https://github.com/Nuopel/Filtering/tree/master/IIR

I. Introduction

Butterworth filters are well known for their flat frequency response in the bandpass, and the fact that they have a -3dB cut off frequency regardless of the filter order.

In order to show how work the filter, low pass filter will be used as example.
Its transfer function is characterised by :

$$H(j\omega) = \frac{1}{\sqrt{(1 + (\frac{j\omega}{\omega_c})^{2n})}}, \tag{1}$$

where ω is the angular frequency $\omega = 2\pi f$, f being the frequency, ω_c, the angular cut off frequency and n the order of the filter.
It's easily discernible that at the cut off frequency $\omega = \omega_c$, the filter response is -3dB ($H(\omega_c) = \frac{1}{\sqrt{2}}$) and this will be true for all order n.

II. Define the general continuous expression

As the aim is to determine a general filter transfer function the Laplace domain is used.
Departing from the previous equation, looking at the gain $|H(j\omega)|^2$ but using Laplace formulation ($s = j\omega$) and using the fact that $|H(s)|^2 = H(s)\overline{H(s)}$ and $\overline{j\omega} = -j\omega$, it can be written:

$$H(s)H(-s) = \frac{G_0^2}{1 + (\frac{-s^2}{\omega_c^2})^n}, \tag{2}$$

where G_0 is the gain.
This equation have a convenient pole form, which in signal processing is useful to determine the stability of continuous filter. The condition is that the pole must be located in the negative part of the s plan.
Fulfil this condition means:

$$-\frac{s_k^2}{\omega_c^2} = (-1)^{\frac{1}{n}} = e^{j\pi\frac{2k+n-1}{2n}}, \tag{3}$$

[1]where $k = 1, 2, 3, ..., n$.
Thus, to find the pole denoted s_k:

$$s_k = \omega_c e^{j\pi\frac{2k+n-1}{2n}} \tag{4}$$

The general pole form of the low pass Butterworth is assumed as:

$$H_a(s) = \frac{G_0}{\prod_{k=1}^{n}(s - s_k)/\omega_c} \tag{5}$$

Now that the continuous expression is established as well as the pole value $p_k = s_k$, it's time to see how to create a numerical filter. A plot of a continuous filter is presented Figure 1

III. Using bilinear transform to produce a digital filter

The continuous Laplace domain can't be used on numerical data, thus it will be seen now how to convert a continuous filter to the discrete domain, referred as the Z domain.
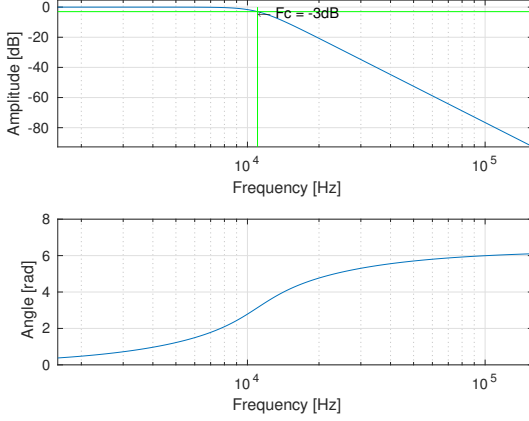
---

[1]$e^{j(\pi+2n)} = -1$

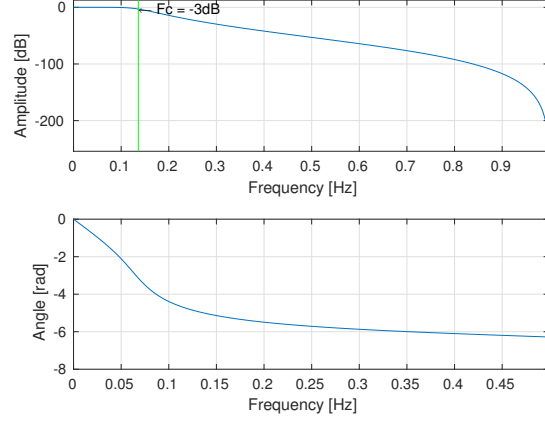**Figure 1:** *Continuous low-pass filter order 4 $f_c = 11025$*

Figure 2: *Discrete low pass filter order 4 $f_c = 3000$, $f_s = 44100$*

To pass from a domain to another the bilinear transform is used:

$$s \approx \frac{2}{T_s} \frac{1 - z^{-1}}{1 + z^{-1}} \qquad (6)$$

Using the 6 into the equation 5 it's obtained:

$$H_d(z) = G_0 \prod_{k=1}^{n} \frac{1}{s_k + A} \frac{z^{-1} + 1}{z^{-1} - \frac{A - s_k}{A + s_k}}, \qquad (7)$$

where $A = \frac{2}{T_s * \omega_c}$, T_s being the inverse of the sampling frequency.
The discrete pole p_{zk} are easily recognisable:

$$p_{zk} = \frac{A - \frac{s_k}{\omega_c}}{A + \frac{s_k}{\omega_c}} = \frac{\frac{2}{T_s * \omega_c} - \frac{s_k}{\omega_c}}{\frac{2}{T_s * \omega_c} + \frac{s_k}{\omega_c}}. \qquad (8)$$

The gain of the filter G_z can also be derived from Eq. 7:

$$G_z = G_0 \prod_{k=1}^{n} \frac{1}{\frac{2}{T_s * \omega_c} - \frac{s_k}{\omega_c}}. \qquad (9)$$

The zeros are easy to define looking at 7, there is as much zeros equal to -1 than the order:

$$z_{zk} = -1. \qquad (10)$$

The Figure 2 shows a discrete low pass filter generated using the previous method.

i. Warping effect of bilinear transform

In order to determine the frequency response of the Laplace Eq. 5, s is evaluated at $s = j\omega$,

thus as the Z plane bilinear transform refers to $z = e^{sT_s}$ it means that $z = e^{j\omega T_s}$ (the form above is the first order Taylor expansion).
The z filter is evaluated at $H_d(z)$ while the continuous one is evaluated at $H_a(s)$, thus it's observed $e^{j\omega T_s}$ which maps ω onto πF_s.
The discrete filter is mapped to $z = \omega_d = e^{j\omega T_s}$, while the continuous is mapped to $s = j\omega_a \approx \frac{2}{T} \frac{z-1}{z+1}$:

$$H_d(z) \approx H_a\left(\frac{2}{T_s} \frac{z-1}{z+1}\right),$$

$$H_d(e^{j\omega T_s}) \approx H_a\left(\frac{2}{T_s} \frac{e^{j\omega T_s} - 1}{e^{j\omega T_s} + 1}\right), \qquad (11)$$

$$H_d(j\omega_a) \approx H_a\left(j\frac{2}{T_s} tan\left(\frac{\omega T_s}{2}\right)\right).$$

This shows that the points of H_d are mapped onto the continuous s plane, $s = j\omega_a = e^{j\omega T_s}$. Thus, the discrete-time to continuous-time frequency mapping/warping of the bilinear transform is :

$$\omega_a = \frac{2}{T_s} tan\left(\frac{\omega T_s}{2}\right). \qquad (12)$$

The warping effect of the bilinear can be compensated by setting the cut off frequency ω_c to $\frac{2}{T_s} tan\left(\frac{\omega_c T_s}{2}\right)$.
The warping effect is effective only in high frequency, if $\omega << F_s/2$ then $tan(\omega) = \omega$. It's illustrated by the Figure 3, which shows two filters at low and high frequency in both warped and unwarped case.
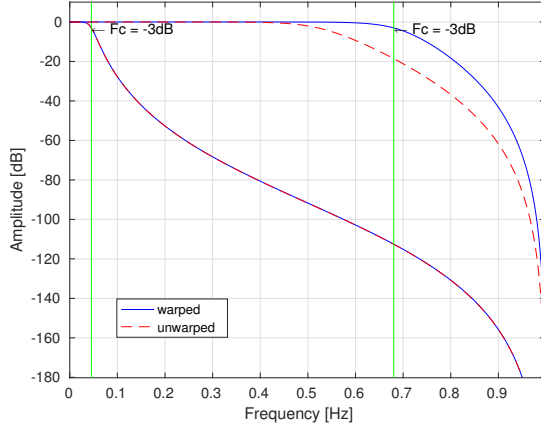
Figure 3: *Effect of warping on two filters $f_c = 3000Hz$, $f_c = 15000Hz$, $F_s = 44100Hz$*

IV. TOWARD A REAL IMPLEMENTATION

As the aim of the Butterworth filters is to be implemented this section will show how to implement them and some ways to reduce the calculation as well as things to pay attention.

i. Precision of the variable

Taking the Eq.4 and Eq.5 for a straight forward implementation would lead to an impossibility of calculating high order of Butterworth filter. Indeed the calculated gain would be related to $G = G_0 * (\omega_c)^n$ and would rise really fast until that there is not enough precision to calculate. A simple alternative is to take into account this value at each pole modifying s_k to :

$$s_k = e^{j\pi \frac{2k+n-1}{2n}}. \tag{13}$$

So that the pole would be written as :

$$p_{zk} = \frac{A - s_k}{A + s_k} = \frac{\frac{2}{Ts*\omega_c} - s_k}{\frac{2}{Ts*\omega_c} + s_k}. \tag{14}$$

The gain would be $G_z = G_0$.

ii. Pre-calculation

If you look at the value A it can be seen that to calculate the parameters of the filter you don't really need to know F_s or ω_c but the ratio $\frac{F_s}{\omega_c}$. Thus this value should be calculated beforehand to reduce calculation.

V. USING BIQUAD IMPLEMENTATION

As the order of the filter is rising, the number of coefficients of the IIR filter is growing as well. This imply that the response of the filter is becoming slow, to bear with this the filter can be transformed in a cascade of second order filter, improving the speed.

i. Biquad continuous expression

The coefficients of these second order filters can be derived from the the analytical formulation of the continuous pole Eq.4. A smart way to reduce the cost is to pair two poles so that the resulting second order equation as real coefficients (instead of complex ones).
Departing form the pole equation it can be observed for $k > 1$ and $k \neq \frac{n+1}{2}$, for n odd :

$$
\begin{aligned}
s_k &= \overline{s_{n-k}}, \\
e^{j\pi \frac{2k+n-1}{2n}} &= \overline{e^{j\pi \frac{2(n-k)+n-1}{2n}}}, \\
&= \overline{e^{j\pi \frac{-2k+3n-1}{2n}}} = \overline{e^{j\pi \frac{-2k-n-1}{2n}} e^{j\pi \frac{4n}{2n}}}, \quad (15) \\
&= \overline{e^{-j\pi \frac{2k+n-1}{2n}} * 1}, \\
&= e^{j\pi \frac{2k+n-1}{2n}}.
\end{aligned}
$$

Thus pairing two conjugates poles (using $s = \frac{s}{\omega_c}$) lead to:

$$
\begin{aligned}
(s - s_k)(s - s_{n-k}) &= s^2 - s(s_k + s_{n-k}) + s_k s_{n-k}, \\
&= s^2 - s(e^{j\theta} + e^{-j\theta}) + e^{j\theta} e^{-j\theta}, \\
&= s^2 - s2cos(\theta) + 1,
\end{aligned}
$$
$$(16)$$

where $\theta = \pi \frac{2k+n-1}{2n}$. It can be also observed that for odd order filter at $k = \frac{n+1}{2}$:

$$
\begin{aligned}
s_{\frac{n+1}{2}} &= e^{j\pi \frac{2(\frac{n+1}{2})+n-1}{2n}}, \\
&= e^{j\pi \frac{2n}{2n}}, \\
&= e^{-j\pi} = -1.
\end{aligned}
\tag{17}
$$

Using the previous statement a general formulation, known as the normalized Butterworth

3

polynomial B_n, can be written :

For n even:

$$B_n(s) = \prod_{k=1}^{\frac{n}{2}} [s^2 - 2s\cos(\pi \frac{2k+n-1}{2n}) + 1].$$

For n odd:

$$B_n(s) = (s+1) \prod_{k=1}^{\frac{n+1}{2}} [s^2 - 2s\cos(\pi \frac{2k+n-1}{2n}) + 1].$$

(18)

The continuous expression of a low-pass filter is then expressed as:

$$H_a(s) = \frac{G_0}{B_n(s)}$$

(19)

It must be kept in mind that for a desired cut off frequency ω_c, the previous equation should replace $s = \frac{s}{\omega_c}$.

ii. Biquad digital expression

Now that the continuous second order coefficients are known, the digital coefficients must be calculated. Departing from the continuous second order transfer function:

$$H_A(s) = \frac{B_0 s^2 + B_1 s + B_2}{A_0 s^2 + A_1 s + A_2},$$

(20)

and applying the bilinear transform with a warped ω_c showed before :

$$s = K \frac{1 - z^{-1}}{1 + z^{-1}},$$

(21)

where $K = \frac{2}{T_s * \omega_c}$, leads to the digital transfer function:

$$H_d(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{a_0 + a_1 z^{-1} + a_2 z^{-2}},$$

(22)

where the coefficients B_i and A_i of $H_d(z)$ calculated from $H(s)$ coefficients are :

$$b_0 = \frac{B_0 K^2 + B_1 K + B_2}{A_0 K^2 + A_1 K + A_2}$$

$$b_1 = \frac{2B_2 - 2B_0 K^2}{A_0 K^2 + A_1 K + A_2}$$

$$b_2 = \frac{B_0 K^2 - B_1 K + B_2}{A_0 K^2 + A_1 K + A_2}$$

$$a_0 = 1$$

$$a_1 = \frac{2A_2 - 2A_0 K^2}{A_0 K^2 + A_1 K + A_2}$$

$$a_2 = \frac{A_0 K^2 - A_1 K + A_2}{A_0 K^2 + A_1 K + A_2}$$

(23)

It can be seen that $a_0 = 1$ result from a normalisation of the filter, so that the original value was $a_0 = A_0 K^2 + A_1 K + A_2$.

To go further for the optimisation, if $n > 2$, it can be a good idea to normalise all the biquad filter except the first one by their respective b_0 so that these denominator filter coefficients will be:

$$b_0 = 1$$

$$b_1 = \frac{2B_2 - 2B_0 K^2}{B_0 K^2 + B_1 K + B_2}$$

$$b_2 = \frac{B_0 K^2 - B_1 K + B_2}{B_0 K^2 + B_1 K + B_2}$$

(24)

A general gain should take into account each normalisation, for the number total of biquad p:

$$G_z = \prod_{k=2}^{p} \frac{B_{p0} K^2 + B_{p1} K + B_{p2}}{A_{p0} K^2 + A_{p1} K + A_{p2}}.$$

(25)

This gain should multiply the first filter which depends if the order is odd or even :

For n even:

$$b_{1,0} = \frac{B_0 K^2 + B_1 K + B_2}{A_0 K^2 + A_1 K + A_2} * G_z$$

$$b_{1,1} = \frac{2B_2 - 2B_0 K^2}{A_0 K^2 + A_1 K + A_2} * G_z$$

$$b_{1,2} = \frac{B_0 K^2 - B_1 K + B_2}{A_0 K^2 + A_1 K + A_2} * G_z$$

For n odd:

$$b_{1,0} = \frac{B_1 K + B_2}{A_1 K + A_2} * G_z$$

$$b_{1,1} = \frac{-B_1 K + B_2}{A_1 K + A_2} * G_z$$

$$b_{1,2} = 0$$

$$a_{1,0} = 1$$

$$a_{1,1} = \frac{-A_1 K + A_2}{A_1 K + A_2}$$

$$a_{1,2} = 0$$

(26)

Finally, it can also be take into account that for low pass filter B_0, B_1 are null and A_0, A_2 are equal to one.

VI. Other kind of filter

i. High-pass filter

A simple way to obtain a high-pass filter using the coefficient of the low-pass is to use :

$$\frac{s}{\omega_c} \to \frac{\omega_c}{s} \tag{27}$$

Departing from the continuous transfers function:

$$\frac{B_0 s^2 + B_1 s + B_2}{A_0 s^2 + A_1 s + A_2} \to \frac{B_0 + B_1 s + B_2 s^2}{A_0 + A_1 s + A_2 s^2}, \tag{28}$$

thus it's just needed to swap B_0 to B_2 and A_0 to A_2 in the previous low pass equation.

It obtained for all the p filter expect the first one:

$$b_{p0} = 1$$

$$b_{p1} = \frac{2B_0 - 2B_2 K^2}{B_2 K^2 + B_1 K + B_0}$$

$$b_{p2} = \frac{B_2 K^2 - B_1 K + B_0}{B_2 K^2 + B_1 K + B_0}$$

$$a_{p0} = 1$$

$$a_{p1} = \frac{2A_0 - 2A_2 K^2}{A_2 K^2 + A_1 K + A_0}$$

$$a_{p2} = \frac{A_2 K^2 - A_1 K + A_0}{A_2 K^2 + A_1 K + A_0}$$

(29)

and for $p = 1$:

For n even:

$$b_{1,0} = \frac{B_2 K^2 + B_1 K + B_0}{A_2 K^2 + A_1 K + A_0} * G_z$$

$$b_{1,1} = \frac{2B_0 - 2B_2 K^2}{A_2 K^2 + A_1 K + A_0} * G_z$$

$$b_{1,2} = \frac{B_2 K^2 - B_1 K + B_2}{A_2 K^2 + A_1 K + A_0} * G_z$$

For n odd:

$$b_{1,0} = \frac{B_2 K + B_1}{A_2 K + A_1} * G_z$$

$$b_{1,1} = \frac{-B_2 K + B_1}{A_2 K + A_1} * G_z$$

$$b_{1,2} = 0$$

$$a_{1,0} = 1$$

$$a_{1,1} = \frac{-A_2 K + A_1}{A_2 K + A_1}$$

$$a_{1,2} = 0$$

(30)

With:

$$G_z = \prod_{k=2}^{p} \frac{B_{p2} K^2 + B_{p1} K + B_{p0}}{A_{p2} K^2 + A_{p1} K + A_{p0}}. \tag{31}$$

It's also possible to directly use the digital coefficients of the low-pass filter using the properties of the Butterworth filter. Indeed, the coefficients A_0 and A_2 are equal to 1, and the coefficient B_0 and B_1 equal to 0. Thus by observing the discrete coefficients of low pass and high pass filter it seen that the only difference is :

$$b_{1hp} = -b_{1lp}, \tag{32}$$

and

$$G_{zhp} = G_{zlp} * K^n, \tag{33}$$

Thus inverse the sign of b_1 and multiply the first filter numerator by K^n.

ii. Bandpass filter

A way to obtain band pass filter using the coefficient of the low-pass is to use :

$$\frac{s}{\omega_c} \rightarrow \frac{\omega_c}{\Delta\omega}\left(\frac{s}{\omega_o} + \frac{\omega_0}{s}\right), \qquad (34)$$

with $\omega_0 = \sqrt{\omega_1\omega_2}$ and $\Delta\omega = \omega_2 - \omega_1$.
This transformation involve a complex transformation of the filter resulting a doubled order filter (for low frequency and high frequency filtering).
The easiest way is to use a low pass filter combined with a high pass filter.

References

[https://www.mathworks.com/help/signal/ref/bilinear.html]

[https://en.wikipedia.org/wiki/Bilinear_transform]

[https://en.wikipedia.org/wiki/Butterworth_filter]

A. Bilinear demo Eq. 7

$$\frac{1}{\frac{w_c}{s-s_k}} \rightarrow \frac{1}{\frac{A(1-z^{-1})}{1+z^{-1}} - \frac{s_k}{\omega_c}} = \frac{z^{-1}+1}{B - Bz^{-1} - \frac{s_k}{\omega_c} - z^{-1}\frac{s_k}{\omega_c}}$$

$$= \frac{z^{-1}+1}{z^{-1}\left(-\frac{s_k}{\omega_c} - B\right) - B - \frac{s_k}{\omega_c}}$$

B. Warping demo Eq. 11

$$\frac{e^{j\omega T_s} - 1}{e^{j\omega T_s} + 1} = \frac{e^{j\omega T_s/2}\left(e^{j\omega T_s/2} - e^{j\omega T_s/2}\right)}{e^{j\omega T_s/2}\left(e^{j\omega T_s/2} + e^{j\omega T_s/2}\right)}$$

$$= \frac{j\left(e^{j\omega T_s/2} - e^{j\omega T_s/2}\right)/2j}{\left(e^{j\omega T_s/2} + e^{j\omega T_s/2}\right)/2}$$

$$= j\frac{sin(\omega T_s/2)}{cos(\omega T_s/2)}$$

$$= jtan\left(\frac{\omega T_s}{2}\right)$$