866 United Nations Plaza • Suite 566 • New York, NY 10017
Phone: +1 212 223 3552 • Fax: +1 212 421 6608
info@egartech.com • www.egartech.com

On the realization of the Butterworth filter in MATLAB

(Warning to quantitative analyst)

Mark Ioffe

Abstract

The analysis of time series is an essential tool of the financial analyst. In analysis time series digital filters and in particular low frequency play an important role. In this paper from mathematical point of view is considered low-pass Butterworth filter. Digital filter design, which consists in finding the coefficients of a finite difference scheme, is reduced to finding the coefficients of the continuous filter, for which the coefficients are the roots of the imaginary unit. To recalculate the cutoff frequency of the discrete to continuous time no additional parameters are required. As an example, compare the amplitude characteristics of the digital filter obtained by means of this method with the filter obtained in MATLAB function butter (n, f0). The reason of differences is that in MATLAB for frequency conversion from discrete to continuous one is using additional time parameter.

There are two types of Butterworth filters: continuous and discrete. From a mathematical point of view, continuous filter is described by some differential equation, discrete filter by finite-difference equation. We first discuss the continuous filter. Consider the heterogeneous linear differential equation of order n with constant coefficients a (1), a (2) .... a (n +1) of the form:

$$a(1)\frac{d^n y}{dt^n} + a(2)\frac{d^{n-1} y}{dt^{n-1}} + ...a(n)\frac{dy}{dt} + a(n+1)y = x(t)$$

(1)

Assume that the right-hand side of the equation, the function x (t) is a harmonic oscillation with a unit complex amplitude, that is:

$$x(t) = e^{i\mu t}$$

(2)

Obviously, the steady-state solution (1) is also a harmonic oscillation of the same frequency with complex amplitude. That is:

$$H(i\mu) = \left(\sum_{k=1}^{n+1} a(k)(i\mu)^{n+1-k} \right)^{-1}$$

$$y(t) = H(i\mu)e^{i\mu t}$$

(3)

The square of the amplitude of the output signal, obviously, is:

$$A^2(\mu) = H(i\mu)H(-i\mu)$$

(4)

Butterworth low-pass filter with a cutoff frequency ω is a special case of equation (1), for which the square of the amplitude of the output signal is:

$$A^2(\mu) = \frac{1}{1 + \left(\frac{\mu}{\mu_0} \right)^{2n}}$$

(5)

Design of such a filter is to find such n +1 coefficients a (1), a (2), ... a (n +1), which provide the equality:

$$\sum_{k=1}^{n+1} a(k)(i\mu)^{n+1-k} \sum_{k=1}^{n+1} a(k)(-i\mu)^{n+1-k} = 1 + \left(\frac{\mu}{\mu_0} \right)^{2n}$$

(6)

Obviously, the

$$a(n+1) = 1$$

(7)

To find the remaining coefficients we use the fact that the left and right sides of (6) are polynomials of degree 2n. The roots of the polynomial on right-hand side are:

$$a_k = \mu_0 \sqrt[2n]{-1} = \mu_0 e^{i(\text{®} + 2\text{®}k(k-1))\frac{1}{2n}}$$

$$k = 1,2,....2n$$

(8)

Polynomial, which is in the left side, is the product of two polynomials of variable $i\mu$ in degree n with real coefficients. Obviously, with the change of variable $i\mu$ on μ roots of new polynomial with the same real coefficients will be $i^{ª}{}_{k}$, k = 0,1, 2n-1. Since the polynomial with real coefficients has either real or complex conjugate pairs of numbers, and the roots of two polynomials, located on the left side have opposite signs, we find that

$$\sum_{k=1}^{n+1} a(k)s^{n+1-k} = \frac{1}{μ_0^n}\prod_{k=1}^{b}(s - «_k)$$

$$«_k = μ_0 e^{i®\left(\frac{(2k+1)}{2n}+\frac{1}{2}\right)}$$

(9)

For given roots of the polynomial its coefficients can be calculated by recursion. Let b (n, n +1), n = 1,2, ... N is a matrix of n +1 coefficients of the order n polynomial with roots $«_k$. Obviously

$$b(1,1) = 1$$
$$b(1,2) = -«_1$$

(10)

If n> = 2, then

$$b(n,1) = 1$$
$$b(n, n+1) = -«_n b(n-1, n)$$
$$b(n,k) = b(n-1, k-1) - «_n b(n-1, k)$$

(11)

Note that the matrix b (n, n +1) corresponds to a polynomial whose coefficient of the highest degree is 1.

How found above continuous filter coefficients can be used in computer to create a program that realizes that for which the filter is intended? That is, to separate the useful signal from the same signal measured with noise. Direct method is the solution of the differential equation (1), right-hand side is a known function of time. More precisely, since the equation of order n we must solve the n-dimensional system of first order differential equations. For solutions can be used, for example, MATLAB function ode45. Vector of variables in this case is:

$$\bar{y} = (\bar{y}(1), \bar{y}(2).....\bar{y}(n)) = (y, \frac{dy}{dt}.....\frac{d^{n-2}y}{dt^{n-2}}, \frac{d^{n-1}y}{dt^{n-1}})$$

(12)

Vector of right-hand sides, as a function of time and the vector of variables:

$$\bar{z} = \frac{d\bar{y}}{dt} = (\bar{y}(2), \bar{y}(3), \ldots . \bar{y}(n), \bar{z}(n))$$

$$\bar{z}(n) = \frac{1}{a(1)}(x(t) - \sum_{k=2}^{n} a(k)\bar{y}(n-k+1))$$

(13)

The initial conditions is naturally selected vector:

$$\bar{y}(0) = (x(0), 0, 0 \ldots 0)$$

(14)

We next consider the discrete Butterworth n order filter. It corresponds to the following finite difference equation:

$$b(1)y(m) + b(2)y(m-1) + \ldots . b(n)y(m-n+1) + b(n+1)y(m-n) = c(1)x(m) + c(2)x(m-1)$$
$$+ \ldots . c(n)x(m-n+1) + c(n+1)x(m-n)$$

(15)

In the analysis of filtration properties of the filter is commonly used analog to harmonic oscillation function:

$$x(k) = e^{i2®fk}$$

(16)

Obviously, this function is a periodic function of the parameter f with period 1. At the entrance, defined by the function (16), the output obviously is:

$$y(k) = HH(e^{i2®f})x(k) = \frac{\sum_{j=1}^{n+1} c(j)e^{i2®f(n-j+1)}}{\sum_{j=1}^{n+1} b(j)e^{i2®f(n-j+1)}} x(k)$$

(17)

Function $HH(z = e^{i2®f})$ is a rational function of the complex variable z with real coefficients. Consider the so-called bilinear transformation of the complex variable z in the complex variable s:

$$s = \frac{z-1}{z+1}$$

(18)

Note that in (18), there is no time constant. If $z = e^{i2®f}$, that is lies on a circle of unit radius, the value of s is equal to

$$s = \frac{e^{i2®f} - 1}{e^{i2®f} + 1} = itg(®f)$$

(19)

In this case the value of s lies on the imaginary axis, that is, each value of f can be associated with the value of the angular frequency ω, in accordance with the formula:

$$\mu = tg(®f)$$

(20)

Thus, each of the digital Butterworth filter can be associated with a continuous Butterworth filter. In particular, the low-pass continuous filter with a cutoff frequency μ_0 can be associated with low-pass digital filter with a cutoff frequency f_0 is equal to

$$f_0 = \frac{1}{®} Atg(\mu_0)$$

(21)

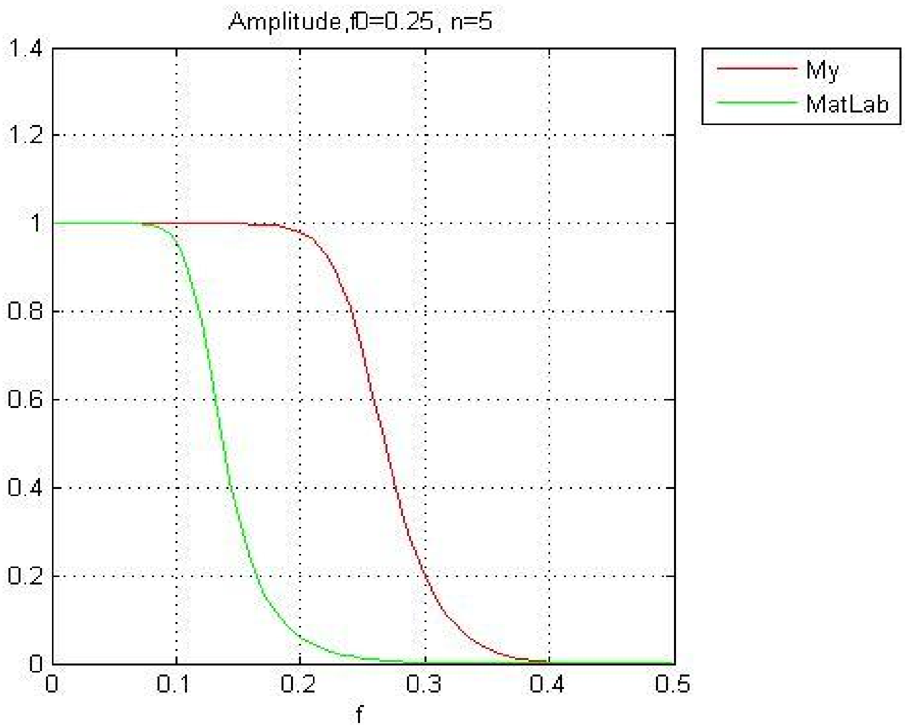
To find the corresponding coefficients of the digital filter to a polynomial with coefficients a (1), a (2), ... a (n +1) we have to substitute (18) and compute the corresponding coefficients of the numerator and denominator. Thus, the

$$\frac{\sum_{k=1}^{n+1} a(k)(z-1)^{n+1-k}(z+1)^{k-1}}{(z+1)^n} = \frac{\sum_{k=1}^{n+1} b(k) z^{n+1-k}}{\sum_{k=1}^{n+1} c(k) z^{n+1-k}}$$

(22)

Thus, finding the appropriate low-pass coefficients of the digital Butterworth filter reduced to finding the coefficient of the continuous filter. Note that the above method of determining the coefficients of the digital Butterworth filter is independent of any time parameter. Consider as an example the amplitude characteristics of the low-pass Butterworth filter of 5th order with a cutoff frequency of 0.25 obtained by the method described above, as well as through MATLAB function butter (n, f0).

866 United Nations Plaza • Suite 566 • New York, NY 10017
Phone: +1 212 223 3552 • Fax: +1 212 421 6608
info@egartech.com • www.egartech.com



As follows from the chart we obtained better fits the filter cutoff parameter $f_0 = 0.25$. The difference is explained by the fact that in MATLAB instead of bilinear transformation (18) is used depends on the parameter T conversion:

$$s = \frac{2}{T}\frac{z-1}{z-1}$$

(23)

For the calculation frequency of a continuous filter using the formula:

$$\mu = 2\,fs * tg\left(\frac{\text{®} f_0}{fs}\right)$$

$$fs = 2$$

(24)