

CHAPTER 13 IIR FILTER DESIGN

13.1 Analog/Digital Lowpass Butterworth Filter

This document designs a lowpass digital IIR filter of the Butterworth type. A bilinear transformation is performed to create a digital filter from the analog design. You specify:

- **T**, the sampling rate
- **fp**, the passband edge frequency
- **fs**, the stopband edge frequency
- **G**, the minimum stopband attenuation

Mathcad calculates the required filter order and constructs the transfer function in the **s**-plane.

This document prewarps the analog filter frequencies, and subsequently gives the digital transfer function. Filter output is displayed for some simple input sequences in two ways: by direct convolution with the truncated impulse response, and by sequence generation using a difference equation.

Background

Digital filters are often designed using analog techniques, and then converting the analog transfer function into digital terms. This approach was adopted because analog filter design is a well-understood and refined science, and in some cases easier to implement.

Bilinear Transformation

There are a number of methods for converting functions from the continuous-time Laplace Transform domain to the discrete-time **z**-transform domain. One of the most universal employs a bilinear transformation between the **s**-plane and the **z**-plane:

$$s = \frac{2}{T} \cdot \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right)$$

where **T** is the sampling rate.

Frequency and Phase Warping

The continuous-time frequency axis is mapped onto the unit circle in the **z**-plane. Since the transformation is, necessarily, nonlinear, there will be some "warping" of the frequencies as they convert from the analog to digital domain. It is not so critical that some frequencies are unevenly distributed, except in the case of the design frequencies at the band edges. It is possible to "prewarp" these frequencies so the digital filter still meets design constraints, using the relation

$$\omega_s = \frac{2}{T} \cdot \tan\left(\frac{\omega_z}{2}\right)$$

where ω_z varies between 0 and 2π . The bilinear transformation also warps the phase of the filter, particularly at the edges of the frequency range. The phase characteristic of all filters designed by this technique is nonlinear.

Mathcad Implementation

The filter design depends on four parameters: the sampling interval, passband and stopband edges, and the desired attenuation in the stop band.

Lowpass Filter Specifications

sampling interval (sec):	$T := 0.001 \text{ s}$
implies a bandlimit (Hz) of:	$f_{max} := \frac{1}{2 \cdot T} = 500 \text{ Hz}$
passband edge (Hz):	$f_p := 90 \text{ Hz}$
stopband edge (Hz):	$f_s := 100 \text{ Hz}$
attenuation for frequencies above the stopband edge:	$G := .2$

To prepare for analog filter design, prewarp the passband and stopband frequencies, and normalize the frequency range so that the passband edge is at 1 (see Parks and Burrus).

Butterworth Filter Design

Prewarp frequencies for the analog filter:

$$\omega\pi_{\pi} := 2 \cdot \pi \cdot f_p = 565.487 \text{ Hz}$$

$$\omega\sigma_{\sigma} := 2 \cdot \pi \cdot f_s = 628.319 \text{ Hz}$$

$$u_p := \frac{2}{T} \cdot \tan\left(\omega\pi_{\pi} \cdot \frac{T}{2}\right) = 581.054 \text{ Hz}$$

$$u_s := \frac{2}{T} \cdot \tan\left(\omega\sigma_{\sigma} \cdot \frac{T}{2}\right) = 649.839 \text{ Hz}$$

Frequency normalization:

$$u := \frac{u_s}{u_p} = 1.118$$

The document then calculates the approximate order for a Butterworth-type filter that will meet the specifications, and stores the poles of the transfer function in the array **p**.

$$N := \text{ceil}\left(\frac{\log\left(\left(\frac{1}{G}\right)^2 - 1\right)}{2 \cdot \log(u)}\right) = 15$$

If the filter does not meet specs, redefine **N** here for a higher order filter :

$$N = 15$$

$$k := 0 \dots N - 1$$

$$p_k := -\sin\left(\frac{\pi \cdot (2 \cdot k + 1)}{2 \cdot N}\right) + 1j \cdot \cos\left(\frac{\pi \cdot (2 \cdot k + 1)}{2 \cdot N}\right)$$

The analog transfer function is:

$$H_s(s) := \prod_k \frac{1}{s - p_k}$$

$$x := .5, .55 \dots 1.45$$

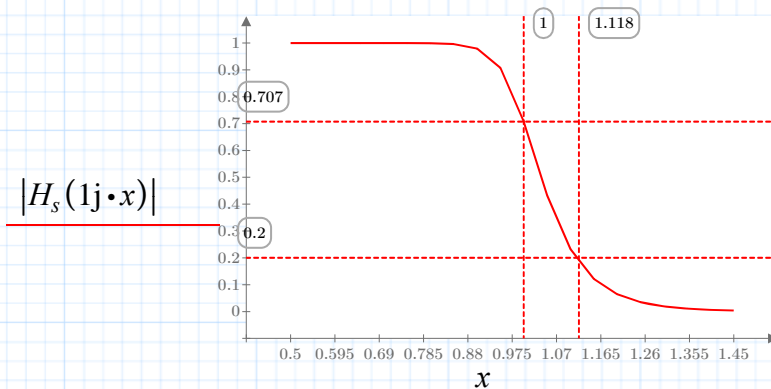


Fig. 13.1 Check the normalized frequency response of the analog filter

Digital Filter Transformation

Find the corresponding digital transfer function using a bilinear transformation.

$$q := \frac{1}{\tan\left(\omega\pi \cdot \frac{T}{2}\right)} \quad H(z) := H_s\left(q \cdot \frac{z-1}{z+1}\right)$$

A plot of the digital frequency filter response shows that we have correctly met our stopband and passband requirements.

$$f := 0 \text{ Hz}, \frac{.01}{T} \dots \frac{1}{2 \cdot T}$$

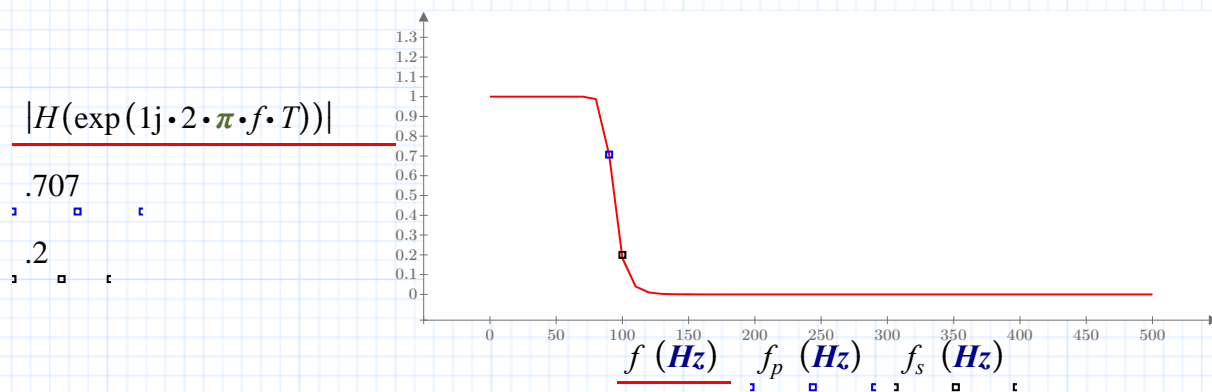
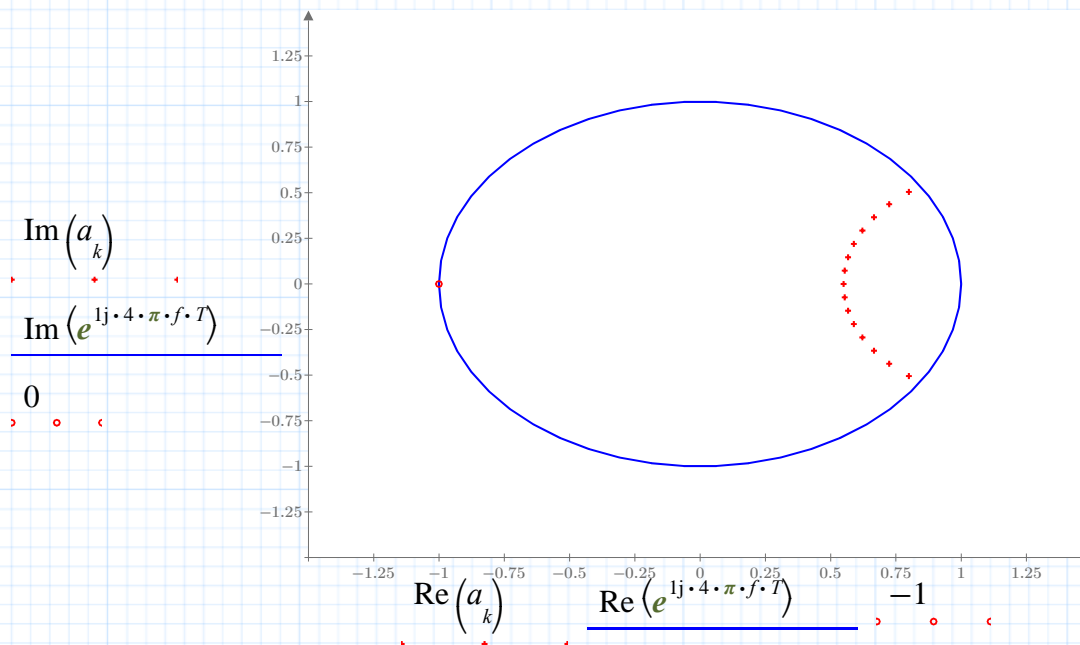


Fig. 13.2 Digital frequency response in hertz

The N zeros of the transfer function are all at $z = -1$.

The N poles are given by:

$$a_k := \frac{q + p_k}{q - p_k}$$



Transform Expansion in Partial Fractions

The previous plot shows the location of the poles of $\mathbf{H}z$ inside the unit circle. These poles are in conjugate pairs, with one real pole if the filter length is odd. The denominator can be written as a product of quadratic factors of the form $z^2 + bz + c$. If N is odd, there is one linear factor of the form $z - \mathbf{blin}$. The coefficients \mathbf{b} and \mathbf{c} are given by:

$$odd := \text{mod}(N, 2) > 0$$

$$m_{max} := \text{if} \left(odd, \text{floor} \left(\frac{N}{2} \right), \frac{N}{2} - 1 \right)$$

$$m := 0 \dots m_{max}$$

$$b_m := -2 \cdot \text{Re} \left(a_m \right)$$

$$c_m := \left(|a_m| \right)^2$$

$$b_{m_{max}} := \text{if} \left(odd, a_{\frac{N-1}{2}}, -2 \cdot \text{Re} \left(a_{m_{max}} \right) \right)$$

$$c_{m_{max}} := \text{if} \left(odd, 0, \left(|a_{m_{max}}| \right)^2 \right)$$

$$b_m = \begin{bmatrix} -1.599 \\ \vdots \end{bmatrix}$$

$$c_m = \begin{bmatrix} 0.894 \\ \vdots \end{bmatrix}$$

When the transfer function is expressed as a rational function in z , the numerator is

$$M \cdot (z + 1)^N$$

where

$$M := H_s(q) = 5.648 \cdot 10^{-10} + 1.809j \cdot 10^{-25}$$

Filter Impulse Response

To generate the impulse response for the digital filter, expand \mathbf{H} in partial fractions. The coefficients for this expansion can be conveniently generated in Mathcad by carrying out a contour integral which finds the residue at each pole. The plot of the impulse response below shows the first 50 terms, calculated from the previously generated poles and partial fraction coefficients.

$$r := .001$$

$$K_k := \frac{1}{2 \cdot \pi} \cdot \int_0^{2 \cdot \pi} H \left(r \cdot \exp(1j \cdot \theta) + a_k \right) \cdot r \cdot \exp(1j \cdot \theta) d\theta$$

H expanded in partial fractions is:

$$H(z) = M + \sum_k \frac{K_k}{z - a_k}$$

The coefficients K are:

$$K_k = \begin{bmatrix} 0.104 + 0.079i \\ \vdots \end{bmatrix}$$

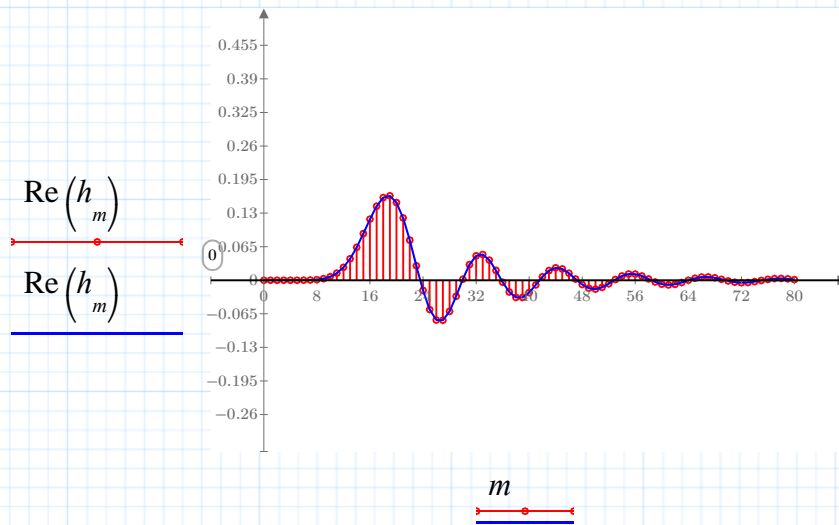
For more information on partial fraction expansion, see **Section 11.1 z-Transform and Inverse Transform**

The beginning of the infinite impulse response sequence is given by:

$$m := 0 \dots 80$$

$$h_m := \sum_k \left(K_k \cdot (a_k)^{m-1} \right)$$

$$h_0 := M$$



Taking real values since imaginary parts are negligible (approximation errors).

Fig. 13.4 Infinite impulse response of filter

Design Check

To calculate the filter output for a sine-wave input, convolve the input signals with the first 50 terms of the impulse response. Two input signals with different frequencies are examined. The low-frequency input **x1** is not attenuated, whereas the higher-frequency input **x2** is, as we would expect. Also, notice the phase shifting, as discussed in the Background section.

$$t := 0 \dots 80$$

Define low and high frequency inputs and their output convolution. **Note:** Frequencies above the band edge should not be used in this calculation: they will not be adequately sampled.

$$\omega_1 := 2 \cdot \pi \cdot T \cdot 9 \cdot f_p$$

$$x1(t) := \sin(\omega_1 \cdot t)$$

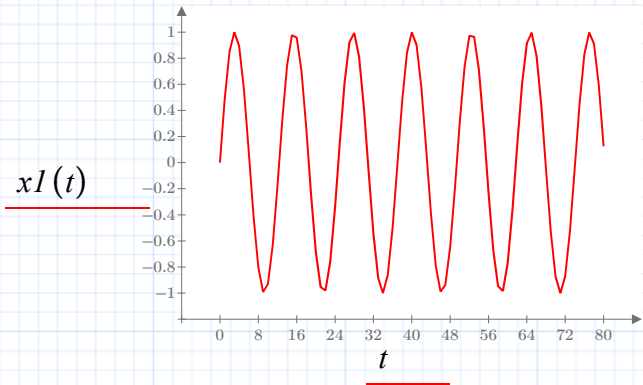
$$y1_t := \sum_m \left(x1(t-m) \cdot h_m\right)$$

$$\omega_2 := 2 \cdot \pi \cdot T \cdot 1.1 \cdot f_s$$

$$x2(t) := \sin(\omega_2 \cdot t)$$

$$y2_t := \sum_m \left(x2(t-m) \cdot h_m\right)$$

Input signals



Outputs

