Conversion of Analog to Digital Transfer Functions^{*}

C. Sidney Burrus

This work is produced by OpenStax-CNX and licensed under the Creative Commons Attribution License 3.0^\dagger

For mathematical convenience, the four classical IIR filter transfer functions were developed in terms of the Laplace transform rather than the z-transform. The prototype Laplace-transform transfer functions are descriptions of analog filters. In this section they are converted to z-transform transfer functions for implementation as IIR digital filters.

There have been several different methods of converting analog systems to digital described over the history of digital filters. Two have proven to be useful for most applications. The first is called the impulse-invariant method and results in a digital filter with an impulse response exactly equal to samples of the prototype analog filter. The second method uses a frequency mapping to convert the analog filter to a digital filter. It has the desirable property of preserving the optimality of the four classical approximations developed in the last section. This section will develop the theory and design formulas to implement both of these conversion approaches.

1 The Impulse-Invariant Method

Although the transfer functions in Continuous Frequency Definition of Error^1 were designed with criteria in the frequency domain, the impulse-invariant method will convert them into digital transfer functions using a time-domain constraint [7], [5], [8]. The digital filter designed by the impulse-invariant method is required to have an impulse response that is exactly equal to equally spaced samples of the impulse response of the prototype analog filter. If the analog filter has a transfer function F(s) with an impulse response f(t), the impulse response of the digital filter h(n) is required to match the samples of f(t). For samples at T second intervals, the impulse response is

$$h(n) = F(T)|_{t=Tn} = F(Tn)$$

$$\tag{1}$$

The transfer function of the digital filter is the z-transform of the impulse response of the filter, which is given by

$$H(z) = \sum_{n=0}^{\infty} h(n) z^{-n}$$
(2)

¹"Least Squared Error Design of FIR Filters": Section Continuous Frequency Definition of Error

^{*}Version 1.2: Nov 17, 2012 6:02 pm -0600

[†]http://creativecommons.org/licenses/by/3.0/

 $<\!http://cnx.org/content/m16892/latest/\#uid28\!>$

The transfer function of the prototype analog filter is always a rational function written as

$$F(s) = \frac{B(s)}{A(s)} \tag{3}$$

where B(s) is the numerator polynomial with roots that are the zeros of F(s), and A(s) is the denominator with roots that are the poles of F(s). If F(s) is expanded in terms of partial fractions, it can be written as

$$F(s) = \sum_{i=1}^{N} \frac{K_i}{s+s_i} \tag{4}$$

The impulse response of this filter is the inverse-Laplace transform of (4), which is

$$f(t) = \sum_{i=1}^{N} K e^{s_i t}$$
(5)

Sampling this impulse response every T seconds gives

$$f(nT) = \sum_{i=1}^{N} K_i \ e^{-s_i nT} = \sum_{i=1}^{N} K_i \left(e^{-s_i T}\right)^n \tag{6}$$

The basic requirement of (1) gives

$$H(z) = \sum_{n=0}^{\infty} \left[\sum_{i=1}^{N} K_i \left(e^{-sIT} \right)^n \right]$$
(7)

$$H\left(z\right) = \sum_{i=1}^{N} \frac{K_i z}{z - e^{sIT}} \tag{8}$$

which is clearly a rational function of z and is the transfer function of the digital filter, which has samples of the prototype analog filter as its impulse response.

This method has its requirements set in the time domain, but the frequency response is important. In most cases, the prototype analog filter is one of the classical types, which is optimal in the frequency domain. If the frequency response of the analog filter is denoted by $F(j\omega)$ and the frequency response of the digital filter designed by the impulse- invariant method is $H(\omega)$, it can be shown in a development similar to that used for the sampling theorem

$$H(\omega) = (1/T) \sum_{k=-\infty}^{\infty} F(j(\omega - 2\pi k/T))$$
(9)

The frequency response of the digital filter is a periodically repeated version of the frequency response of the analog filter. This results in an overlapping of the analog response, thus not preserving optimality in the same sense the analog prototype was optimal. It is a similar phenomenon to the aliasing that occurs when sampling a continuous-time signal to obtain a digital signal in A-to-D conversion. If $F(j\omega)$ is an analog lowpass filter that goes to zero as ω goes to infinity, the effects of the folding can be made small by high sampling rates (small T).

The impulse-invariant design method can be summarized in the following steps:

- 1. Design a prototype analog filter with transfer function $F(j\omega)$.
- 2. Make a partial fraction expansion of $F(j\omega)$ to obtain the N values for K_i and s_i .
- 3. Form the digital transfer function H(z) from (8) to give the desired design.

The characteristics of the designed filter are the following:

- It has N poles, the same as the analog filter.
- It is stable if the analog filter was stable. This is seen from the change of variables in the denominator of (6.70) which maps the left-half s-plane inside the unit circle in the z-plane.
- The frequency response is a folded version of the analog filter, and the optimal properties of the analog filter are not preserved.
- The cascade of two impulse-invariant designed filters are not impulse-invariant with the cascade of the two analog prototypes. In other words, the filter must be designed in one step.

This method is sometimes used to design digital filters, but because the relation of the analog and digital system is specified in the time domain, it is more useful in designing a digital simulation of an analog system. Unfortunately, the properties of this class of filters depend on the input. If a filter is designed so that its impulse response is the sampled impulse response of the analog filter, its step response will not be the sampled step response of the analog filter.

A step-invariant filter can be designed by first multiplying the analog filter transfer function F(s) by 1/s, which is the Laplace transform of a step function. This product is then expanded in partial fraction just as F(s) was in (4) and the same substitution made as in (8) giving a z-transform. After the z-transform of a step is removed, the digital filter has the step-invariant property. This idea can be extended to other input functions, but the impulse-invariant version is the most common. Another modification to the impulse-invariant method is known as the matched z transform covered in [7], but it is less useful.

There can be a problem with the classical impulse-invariant method when the number of finite zeros is too large. This is addressed in [3], [4].

An example of a Butterworth lowpass filter used to design a digital filter by the impulse-invariant method can be shown. Note that the frequency response does not go to zero at the highest frequency of w = p. It can be made as small as desired by increasing the sampling rate, but this is more expensive to implement. Because the frequency response of the prototype analog filter for an inverse-Chebyshev or elliptic-function filter does not necessarily go to zero as w goes to infinity, the effects of folding on the digital frequency response are poor. No amount of sampling rate increase will change this. The same problem exists for a highpass filter. This shows the care that must be exercised in using the impulse-invariant design method.

2 The Bilinear Transformation

A second method for converting an analog prototype filter into a desired digital filter is the bilinear transformation. This method is entirely a frequency-domain method, and as a result, some of the optimal properties of the analog filter are preserved. As was the case with the impulse-invariant method, the time interval is not normalized to one, but is explicitly denoted by the sampling interval T with units of seconds. The bilinear transformation is a change of variables (a mapping) that is linear in both the numerator and denominator [7], [5], [2], [6]. The usual form is

$$s = \frac{2}{T} \frac{z - 1}{z + 1} \tag{10}$$

The z-transform transfer function of the digital filter H(z) is obtained from the Laplace transform transfer function F(s) of the prototype filter by substituting for s the bilinear form of (10).

$$H(z) = F\left(\frac{2(z-1)}{T(z+1)}\right)$$
(11)

This operation can be reversed by solving (10) for z and substituting this into H(z) to obtain F(s). This reverse operation is also bilinear of the form

$$z = \frac{2/T + s}{2/T - s}$$
(12)

To consider the frequency response, the Laplace variable s is evaluated on the imaginary axis and the z-transform variable z is evaluated on the unit circle. This is achieved by

$$s = ju \quad \text{and} \quad z = e^{j\omega T}$$
 (13)

which gives the relation of the analog frequency variable u to the digital frequency variable ω from (13) and (10) to be

$$u = (2/T) \tan\left(\left(\omega T\right)/2\right) \tag{14}$$

The bilinear transform maps the infinite imaginary axis in the-s plane onto the unit circle in the z-plane. It maps the infinite interval of $-\infty < u < \infty$ of the analog frequency axis on to the finite interval of $-\pi/2 < \omega < \pi/2$ of the digital frequency axis. This is illustrated in Figure 1.



Figure 1: The Frequency Map of the Bilinear Transform

There is no folding or aliasing of the prototype frequency response, but there is a compression of the frequency axis, which becomes extreme at high frequencies. This is shown in Figure 2 from the relation of (14).



Figure 2: The Frequency Mapping of the Bilinear Transform

Near zero frequency, the relation of u and ω is essentially linear. The compression increases as the digital frequency w nears $\pi/2$. This nonlinear compression is called frequency warping. The conversion of F(s) to H(z) with the bilinear transformation does not change the values of the frequency response, but it changes the frequencies where the values occur.

In the design of a digital filter, the effects of the frequency warping must be taken into account. The prototype filter frequency scale must be prewarped so that after the bilinear transform, the critical frequencies are in the correct places. This prewarping or scaling of the prototype frequency scale is done by replacing s with Ks. Because the bilinear transform is also a change of variables, both can be performed in one step if that is desirable.

If the critical frequency for the prototype filter is u_o and the desired critical frequency for the digital filter is ω_o , the two frequency responses are related by

$$F(ju_0) = H(\omega_0) = F^*$$
 (15)

The prewarping scaling is given by

$$u_0 = \frac{2}{T} tan\left(\frac{\omega_0 T}{2}\right) \tag{16}$$

Combining the prewarping scale and the bilinear transformation give

$$u_0 = \frac{2K}{T} tan\left(\frac{\omega_0 T}{2}\right) \tag{17}$$

Solving for K and combining with (10) give

$$s = \frac{u_0}{\tan(\omega_0 T/2)} \frac{z-1}{z+1}$$
(18)

All of the optimal filters developed in Continuous Frequency Definition of Error^2 and most other prototype filters are designed with a normalized critical frequency of $u_0 = 1$. Recall that ω_0 is in radians per second. Most specifications are given in terms of frequency f in Hertz (cycles per second) which is related to ω or uby

$$\omega = 2\pi f \tag{19}$$

Care must be taken with the elliptic-function filter where there are two critical frequencies that determine the transition region. Both frequencies must be prewarped.

The characteristics of the bilinear transform are the following:

- The order of the digital filter is the same as the prototype filter.
- The left-half s-plane is mapped into the unit circle on the z-plane. This means stability is preserved.
- Optimal approximations to piecewise constant prototype filters, such as the four cases in Continuous Frequency Definition of Error³, transform into optimal digital filters.
- The cascade of sections designed by the bilinear transform is the same as obtained by transforming the total system.

The bilinear transform is probably the most used method of converting a prototype Laplace transform transfer function into a digital transfer function. It is the one used in most popular filter design programs [1], because of characteristic 3 above that states optimality is preserved. The maximally flat prototype is transformed into a maximally flat digital filter. This property only holds for approximations to piecewise constant ideal frequency responses, because the frequency warping does not change the shape of a constant. If the prototype is an optimal approximation to a differentiator or to a linear-phase characteristic, the bilinear transform will destroy the optimality. Those approximations have to be made directly in the digital frequency domain.

Example 1: The Bilinear Transformation

To illustrate the bilinear transformation, the third-order Butterworth lowpass filter designed in the Example is converted into a digital filter. The prototype filter transfer function is

$$F(s) = \frac{1}{(s+1)(s^2+s+1)}$$
(20)

The prototype analog filter has a passband edge at $u_0 = 1$. A data rate of 1000 samples per second corresponding to T = 0.001 seconds is assumed. If the desired digital passband edge is $f_0 = 200$ Hz, then $\omega_0 = (2\pi) (200)$ radians per second, and the total prewarped bilinear transformation from (18) is

$$s = 1.376382 \frac{z-1}{z+1} \tag{21}$$

 $^{^2&}quot;Least$ Squared Error Design of FIR Filters": Section Continuous Frequency Definition of Error $<\!http://cnx.org/content/m16892/latest/#uid28>$

³"Least Squared Error Design of FIR Filters": Section Continuous Frequency Definition of Error http://cnx.org/content/m16892/latest/#uid28>

The digital transfer function in (20) becomes

$$H(z) = \frac{0.09853116(z+1)^3}{(z-0.158384)(z^2-0.418856z+0.355447)}$$
(22)

Note the locations of the poles and zeros in the z-plane. Zeros at infinity in the s-plane always map into the z = -1 point. The example illustrate a third-order elliptic-function filter designed using the bilinear transform.

3 Frequency Transformations

For the design of highpass, bandpass, and band reject filters, a particularly powerful combination consists of using the frequency transformations described in Section elsewhere together with the bilinear transformation. When using this combination, some care must be taken in scaling the specifications properly. This is illustrated by considering the steps in the design of a bandpass filter:

- 1. First, the lower and upper digital bandedge frequencies are specified as ω_1 and or ω_1 , ω_2 , ω_3 , and ω_4 if an elliptic-function approximation is used.
- 2. These frequencies are prewarped using (16) to give theband edges of the prototype bandpass analog filter.
- 3. These frequencies are converted into a single band- edge ω_p or ω_s for the Butterworth and Chebyshev and into ω_p and ω_s for the elliptic-function approximation of the prototype lowpass filter by using Equation 2 from Frequency Transformations⁴ and Equation 4 from Frequency Transformations⁵.
- 4. The lowpass filter is designed for this ω_p and/or ω_s by using one of the four approximations in the sections in Continuous Frequency Definition of Error⁶ or some other method.
- 5. This lowpass analog filter is converted into a bandpass analog filter with the frequency transformation Equation 6 from Frequency Transformations⁷.
- 6. The bandpass analog filter is then transformed into the desired bandpass digital filter using the bilinear transformation (10).

This is the procedure used in the design Program 8 in the appendix.

When designing a bandpass elliptic-function filter, four frequencies must be specified: the lower stopband edge, the lower passband edge, the upper passband edge, and the upper stopband edge. All four must be prewarped to the equivalent analog values. A problem occurs when the two transition bands of the bandpass filter are converted into the single transition band of the lowpass prototype filter. In general they will be inconsistant; therefore, the narrower of the two transition bands should be used to specify the lowpass filter. The same problem occurs in designing a bandreject elliptic-function filter. Program 8 in the appendix should be studied to understand how this is carried out.

An alternative to the process of converting a lowpass analog into a bandpass analog filter which is then converted into a digital filter, is to first convert the prototype lowpass analog filter into a lowpass digital filter and then make the conversion into a bandpass filter. If the prototype digital filter transfer function is $H_p(z)$ and the frequency transformation is f(z), the desired transformed digital filter is described by

$$H(z) = H_p(f(z)) \tag{23}$$

Since the frequency response of both H(z) and $H_p(z)$ is obtained by evaluating them on the unit circle in the z plane, f(z) should map the unit circle onto the unit circle (|z| = 1 => |f(z)| = 1). Both H(z) and

⁴"Frequency Transformations", (2) <http://cnx.org/content/m16913/latest/#uid7>

⁵"Frequency Transformations", (4) <http://cnx.org/content/m16913/latest/#uid9>

⁶"Least Squared Error Design of FIR Filters": Section Continuous Frequency Definition of Error <<u>http://cnx.org/content/m16892/latest/#uid28></u>

⁷"Frequency Transformations", (6) < http://cnx.org/content/m16913/latest/#uid11>

 $H_p(z)$ should be stable; therefore, f(z) should map the interior of the unit circle into the interior of the unit circle (|z| < 1 => |f(z)| < 1). If f(z) were viewed as a filter, it would be an "all-pass" filter with a unity magnitude frequency response of the form

$$f(z) = \frac{p(z)}{z^n p(1/z)} = \frac{a_n z^n + a_{n-1} z^{n-1} + \dots + a_0}{a_0 z^n + a_1 z^{n-1} + \dots + a_n}$$
(24)

The prototype digital lowpass filter is usually designed with bandedges at $\pm \pi/2$. Determining the frequency transformation then becomes the problem of solving the n + 1 equations

$$f(e^{j\omega_i}) = e^{\pm j\pi/2} = (-1)^i j \tag{25}$$

for the unknown a_k where $i = 0, 1, 2, \dots n$ and the ω_i are the bandedges of the desired transformed frequency response put in ascending order. The resulting simultaneous equations have a special structure that allow a recursive solution. Details of this approach can be found in [6].

This is an extremely general approach that allows multiple passbands of arbitrary width. If ellipticfunction approximations are used, only one of the transition bandwidths can be independently specified. If more than one passband or rejectband is desired, f(z) will be higher order than second order and, therefore, the transformed transfer function H(f(z)) will have to be factored using a root finder.

To illustrate the results of using transform methods to design filters, three examples are given which are designed with Program 8 from the appendix.

Example 2: Design of an Chebyshev Highpass Filter

The specifications are given for a highpass Chebyshev frequency response with a passband edge at $f_p = 0.3$ Hertz with a sampling rate of one sample per second. The order is set at N = 5 and the passband ripple at 0.91515 dB. The transfer function is

$$H(z) = \frac{(z-1)(z^2-2z+1)(z^2-2z+1)}{(z+0.64334)(z^2+0.97495z+0.55567)(z^2+0.57327z+0.83827)}$$
(26)

The frequency response plot is given in Figure 3.



Figure 3: Fifth Order Digital Chebyshev Highpass Filter

Example 3: Design of an Elliptic-Function Bandpass Filter

This filter requires a bandpass frequency response with an elliptic-function approximation. The maximum passpand ripple is one dB, the minimum stopband attenuation is 30 dB, the lower stopband edge $f_1 = 0.19$, the lower passband edge $f_2 = 0.2$, the upper passband edge $f_3 = 0.3$, and the upper stopband edge $f_4 = 0.31$ Hertz with a sampling rate of one sample per second. The design program calculated a required prototype order of N = 6 and, therefore, a total order of 10. The frequency response plot is shown in Figure 4.



Figure 4: Tenth Order Digital Elliptic Bandpass Filter

Example 4: Design of an Inverse-Chebyshev Bandreject Filter

The specifications are given for a bandreject Inverse- Chebyshev frequency response with bandedges at $f_s = 0.1$ and 0.2 Hertz with a sampling rate of one sample per second. The order is set at N = 11 and the minimum stopband attenuation at 30 dB. The frequency response plot is given in Figure 5.



Figure 5: Twenty second Order Digital Inverse Chebyshev Band Reject Filter

3.1 Summary

This section has described the two most popular and useful methods for transforming a prototype analog filter into a digital filter. The analog frequency variable is used because a literature on analog filter design exists, but more importantly, many approximation theories are more straightforward in terms of the Laplacetransform variable than the z-transform variable. The impulse-invariant method is particularly valuable when time- domain characteristics are important. The bilinear-transform method is the most common when frequency-domain performance is the main interest. Use of the BLT warps the frequency scale and, therefore, the digital band edges must be prewarped to obtain the necessary band edges for the analog filter design. Formulas that transform the analog prototype filters into the desired digital filters and for prewarping specifications were derived.

The use of frequency transformations to convert lowpass filters into highpass, bandpass, and bandreject filters was discussed as a particularly useful combination with the bilinear transformation. These are implemented in Program 8 and design examples from this program were shown.

There are cases where no analytic results are possible or where the desired frequency response is not piecewise constant and transformation methods are not appropriate. Direct methods for these cases are developed in other sections.

References

- [1] DSP Committee, editor. Programs for Digital Signal Processing. IEEE Press, New York, 1979.
- [2] B. Gold and C. M. Rader. Digital Processing of Signals. McGraw-Hill, New York, 1969.
- [3] L. B. Jackson. A correction to impulse invariance. IEEE Signal Processing Letters, 7(10):2738211;275, October 2000.
- [4] W. F. G. Mecklenbrauker. Remarks on and correction to the impulse invariant method for the design of iir digital filters. Signal Processing, 80:16878211;1690, August 2000.
- [5] A. V. Oppenheim and R. W. Schafer. Discrete-Time Signal Processing. Prentice-Hall, Englewood Cliffs, NJ, second edition, 1999. Earlier editions in 1975 and 1989.
- [6] T. W. Parks and C. S. Burrus. Digital Filter Design. John Wiley & Sons, New York, 1987.
- [7] L. R. Rabiner and B. Gold. Theory and Application of Digital Signal Processing. Prentice-Hall, Englewood Cliffs, NJ, 1975.
- [8] F. J. Taylor. Digital Filter Design Handbook. Marcel Dekker, Inc., New York, 1983.