"A penny for your thoughts, Mr. Griscom."

EE 311 **March 18-20, 2019**
Lecture 14-15

Chapter 6
Analysis and Design of IIR Filters

• IIR filters have feedback which results in a transfer function that has poles at locations other than the origin.
• Using poles and zeros together can result in a computationally efficient design with narrower transition bands than comparable FIR filters.
• Why we design FIR filters at all if IIR filters are, in general, more computationally efficient.
  stability and linear phase.
• IIR filters can be designed from existing analog filters by finding an appropriate mapping function to map the s-plane to the z-plane.
• Butterworth, Chebyshev, and elliptic (or Cauer) analog filters make good templates for creating digital filters using a mapping function such as the bilinear transform.

6.1 Fundamental IIR design Using the Bilinear Transform
• A low pass continuous time filter has a transfer function in *s* which can be written as

$$H(s) = \frac{Y(s)}{X(s)} = \frac{a_0 s^m + a_1 s^{m-1} + \cdots + a_m}{s^n + b_1 s^{n-1} + \cdots + b_n} \quad \text{where } n > m$$

$$Y(s) = a_0 s^{m-n} X(s) + \cdots + a_m s^{-n} X(S) - b_1 s^{-1} Y(s) - \cdots - b_n s^{-n} Y(s) \qquad (6.1)$$

Let $y(nT) = \int_0^{nT} f(t) \cdot dt$ which gives a difference equation of

• Derive this again!

$$y(nT) = y([n-1]T) + (T/2)[f(nT) + f([n-1]T)]$$

In (6.1), values of *Y*(*s*) and *X(s*) are divided by powers of *s*. But, for an arbitrary function *g*(*t*) which has a LaPlace transform *G*(*s*), the inverse LaPlace transform of *G*(*s*)/*s* corresponds to $\int g(t)dt$. In (6.1), a term such as $s^{-1}Y(s)$ could get replaced by the difference equation term y([n-1]T) + T*[y(nT) + y([n-1]T)]/2. Or, in terms of the *z*-transform this corresponds to $z^{-1}Y(z) + TY(z)*(1+z^{-1})/2$. We could get the same result by replacing

$Y(s) \rightarrow Y(z)$

and

$$s \rightarrow \frac{2}{T}\frac{z-1}{z+1}, \text{ or } Y(z) = Y(s)\Big|_{s=\frac{2}{T}\frac{z-1}{z+1}}. \qquad \textit{Bilinear Transform}$$

The bilinear transform used here has been derived from numerical integration using trapezoids. It could have been derived based solely on its mapping properties from the *s*-plane to the *z*-plane as we shall see.

Example 6.1

Use the bilinear transform to find an equivalent digital filter for the RC network in Figure 6.1. Use a sample frequency f_s = 1000Hz. R = 1000 Ω, C = 1μf.

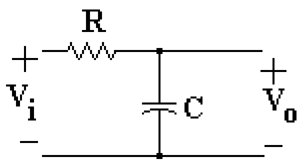


Figure 6.1
An RC circuit. Find the equivalent digital filter using the BLT.

Solution:
The transfer function for the analog filter is

$$H(s) = \frac{1000}{s + 1000}$$

Substituting the equation for the BLT into this equation gives the transfer function for the digital filter.

$$H(z) = \frac{1000}{\frac{2}{T} \cdot \frac{z-1}{z+1} + 1000} = \frac{1000T(z+1)}{z(1000T+2)+(1000T-2)}$$

Using a sample frequency of 1000 Hz (T = .001) reduces this equation to the following:

$$H(z) = \frac{1}{3} \frac{(z+1)}{z - 1/3}$$

• Notice that the transfer function in S has no zeros and the one real pole and this resulted in a transfer function in *z* which has a zero at z = -1 and a single pole.

• In general, when the z-transform is applied to a transfer function in *s* with numerator of order M and denominator of order N there will be N – M zeros at the z = -1 point.

```
%RC.m
fs = 1000;T = 1/fs;
R = 1000;C = 0.000001;
Tau = R*C;
num = [1 1];
den = [1 -exp(-T/Tau)];
figure(1);clf;
[H f] = freqz(num, den, 1024, fs);
Habs = abs(H)/max(abs(H));
subplot(2, 1, 1);
plot(f, Habs);
subplot(2, 1, 2);
plot(f, angle(H)*180/pi);
numA = 1/Tau;denA = [1 1/Tau];
Ha = freqs(numA, denA, 2*pi*f);
subplot(2, 1, 1);hold on;
plot(f, abs(Ha), 'r');
title('Low pass RC Gain response');
xlabel('frequency in Hz');
ylabel('Gain');
subplot(2, 1, 2);hold on;
plot(f, angle(Ha)*180/pi, 'r');
title('Low pass RC Phase resp');
xlabel('frequency in Hz');
ylabel('Phase in degrees');
```

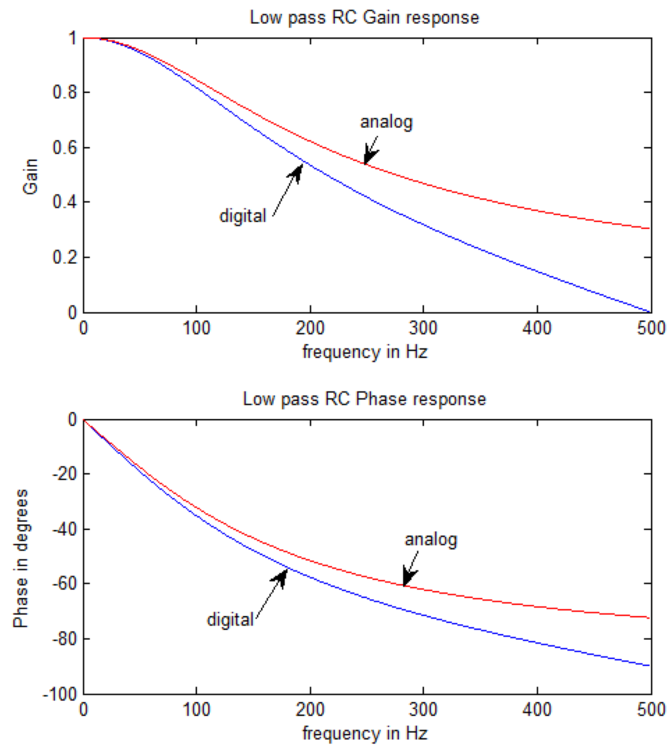


Figure 6.2
Gain and phase vs frequency for the analog low pass RC circuit with the corresponding digital version from the BLT.

• The analysis shows that there is some distortion of both the magnitude and phase in the frequency plot which becomes more pronounced at higher frequencies.

$$z = \frac{2+sT}{2-sT}$$

Let

$s = \sigma + j\omega$ and $z = \alpha + j\beta$

Making these substitutions in the mapping function gives the following:

$$\alpha + j\beta = \frac{(2+\sigma T)+j\Omega}{(2-\sigma T)-j\Omega}, \text{ where } \Omega = \omega T .$$

Multiplying numerator and denominator by the conjugate of the denominator gives:

$$\alpha + j\beta = \frac{-(\sigma T)^2 - (\Omega)^2 + 4 + j(4\Omega)}{(\sigma T)^2 - 4\sigma T + (\Omega)^2 + 4} \tag{6.3}$$

To determine how the jω axis maps into the *z*-plane we let $\sigma = 0$. Equation (6.3) reduces to

$$\alpha + j\beta = \frac{4-(\Omega)^2 + 4j\Omega}{4+(\Omega)^2}$$

Evaluating the magnitude of this equation gives

$$|\alpha + j\beta| = \frac{[16-8(\Omega)^2+(\Omega)^4+16(\Omega)^2]^{1/2}}{[4+(\Omega)^2]} = 1$$

• Thus we see that the $j\omega$ axis in the s-plane maps to the unit circle ($\alpha^2+\beta^2=1$) in the *z*-plane.
• This distortion is referred to as "warping" and the *z*-plane frequency space is said to be "warped".
• The practical implication is that the BLT will move all of the band edges for a filter in *s* with the higher frequencies being moved the most.

s-plane ω	z-plane BLT mapping
0	+1
$\omega_s/4$	$1\angle 76.3^o$
$\omega_s/2$	$1\angle 115.0^o$
ω_s	$1\angle 144.7^o$
$2\omega_s$	$1\angle 161.9^o$
∞	-1

Table 6.1
This table shows how frequencies get mapped from the s-plane to the z-plane by the BLT.

Since we know that the $j\omega$ axis gets mapped to the unit circle we can write

$$s = \frac{2}{T}\frac{z-1}{z+1} \quad \text{and} \quad j\omega = \frac{2}{T}\frac{e^{j\Omega}-1}{e^{j\Omega}+1}$$

Factoring $e^{j\omega T/2}$ out of the numerator and denominator of this equation gives

$$j\omega = \frac{2}{T}\frac{e^{j\Omega/2}-e^{-j\Omega/2}}{e^{j\Omega/2}+e^{-j\Omega/2}}$$

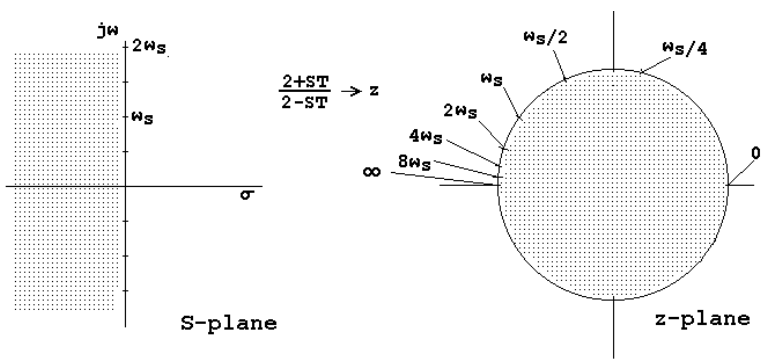


Figure 6.3
This figure illustrates the mapping of the s-plane to the z-plane by the BLT. The jω axis maps to the unit circle. The entire left half s-plane maps to the inside of the unit circle.

Using Euler's identity this equation reduces to the following form:

$$\omega = \frac{2}{T}\frac{\sin(\Omega/2)}{\cos(\Omega/2)}$$

or,

$$\textit{Prewarping Equation:} \quad \omega = \frac{2}{T}\tan(\Omega/2) \tag{6.4}$$

Figure 6.4 shows a graph of this equation with $f_s = 1$. The dashed line is a straight line relationship.

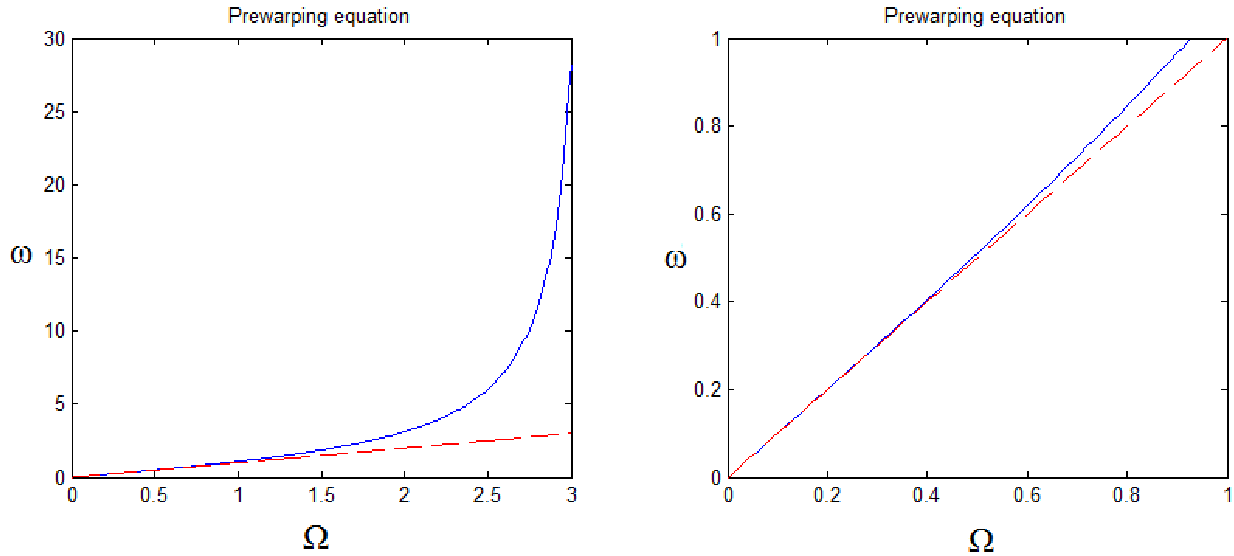


Figure 6.4
This graph shows the prewarping equation with $f_s = 1$. Notice that at frequencies below f_s the relationship is nearly linear. The dashed line is a straight line relationship.

• For example, suppose we have a low pass filter in *s*, which has a cutoff frequency at 250 Hz. We want to use the BLT to create an equivalent digital filter with a sample frequency of f_s = 1000Hz.

• We know that if we apply the BLT directly the established cutoff frequency will be moved slightly.
• To avoid this problem we move the cutoff frequency in *s* from 250 Hz to a new value dictated by the prewarping equation.
• Since we want the cutoff frequency in the *z*-plane to be 250 Hz, we set $\omega = 2\pi(250) = 500\pi$ in (6.4) and get a "prewarped" cutoff frequency in the *s*-plane of

$$\omega_C = (2/T)\tan(500T/2) = 2000\tan(0.5\pi/2) = 2000 \text{ radians} = 318.31\text{Hz}$$

Thus, we "prewarp" our filter in *s* to have a cutoff frequency of 318.31 Hz. When we apply the BLT to this prewarped filter the digital filter will have a cutoff frequency of 250Hz as planned. Example 6.2 illustrates this procedure with a more complex filter.

Example 6.2
Given the low pass Butterworth filter (in *s*) below which has a cutoff frequency of 190 Hz, design a digital version of this filter using the BLT. Prewarp the equation in *s* so that

the digital filter also has a cutoff frequency of 190 Hz. Use 1000Hz for a sample frequency.

$$H(s) = \frac{K}{(s + 1102.9 \pm j456.85)(s + 456.85 \pm j1102.9)}$$

or,

$$H(s) = \frac{K}{s^4 + 3119.6S^3 + 4.8658x10^6 s^2 + 4.4459x10^9 s + 2.0311x10^{12}}$$

Solution:

We begin solving this problem by applying prewarping (6.4) to the cutoff frequency with T = .001 seconds.

$$\omega = (2/.001)\tan(.001(2\pi \cdot 190)/2) = 1359.2 radians / sec = 216.32Hz$$

We must use this new value for the cutoff frequency to design a new analog Butterworth filter with a cutoff frequency of 216.32Hz. Using MATLAB® to produce a new Butterworth filter with the pre-warped cut off frequency gives the following transfer function.

```
[num den] = butter(4, 2*pi*216.32, 's');
```

$$H(s) = \frac{K}{(s + 1255.7 \pm j520.14)(s + 520.14 \pm j1255.7)}$$

or,

$$H(s) = \frac{K}{s^4 + 3551.7s^3 + 6.3073x10^6 s^2 + 6.5613x10^9 s + 3.4128x10^{12}}$$

Applying the BLT (6.2) to this equation gives the transfer function for the digital Butterworth filter.

$$H(z) = \frac{K_z (z+1)^4}{z^4 - .93911z^3 + .76607z^2 - .22937z + .036038}$$

Figure 6.5 shows the magnitude plot for the original Butterworth filter and for the digital version.

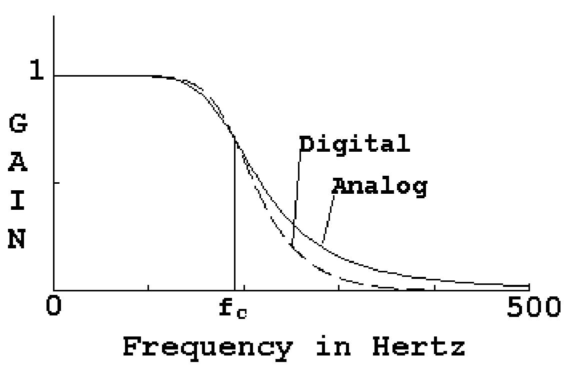


Figure 6.5

A fourth order analog Butterworth filter and its digital counterpart created by using the BLT.

• The design rules for creating a digital filter from an analog filter using the BLT are summarized below.

Design rules for the BLT

1. Begin with a transfer function in s for an analog filter which meets the design specifications.

2. Determine which frequencies need to be preserved in the equivalent digital filter. This is typically the cut off frequency for a low pass filter but for band pass or band stop filters there may be multiple band edges.

3. If there are multiple frequencies to be preserved you must use the prewarping equation (6.4) to find new prewarped values for each frequency. The analog filter must then be reformulated with these prewarped frequencies. If there is but a single frequency to be preserved you may use (6.6) to determine an appropriate constant for the BLT.

4. Apply the BLT to the analog filter to arrive at a transfer function in z. It is typically easier to disregard the gain constant at design time and normalize the filter gain after the design is complete.

• MATLAB® has automated the design of digital filters using the BLT so that the process is not as tedious as the last few examples would lead you to believe.
• The MATLAB® function *bilinear* will do the BLT on a transfer function in *s* to produce a filter in *z*.

Example 6.4
Use MATLAB® to design a digital Butterworth filter to meet the following specifications:

sample frequency:	44,100Hz
pass band frequency range:	0 – 4,000Hz
pass band gain:	1
pass band ripple:	0.01
stop band frequency range:	6000Hz – $f_s/2$
stop band gain:	0
stop band ripple:	0.03

The MATLAB® code below produces a filter to meet these specifications and draws its frequency vs gain plot in Figure 6.6.

```
%Creates a Butterworth filter such that
%fs = 44,100Hz
%pass band 0 - 4000Hz; stop band 8000Hz to fs/2
%pass band ripple = .01; stop band ripple = .03
fs = 44100;
fs2 = fs/2;
fpass = 4000; Rp = .01; RpDB = -20*log10(1-Rp);
fstop = 8000; Rs = .03; RsDB = -20*log10(Rs);
[N fc] = buttord(fpass/fs2, fstop/fs2, RpDB, RsDB);
```

```
[num den] = butter(N, fc);
[H f] = freqz(num, den, 1024, fs);
figure(1);clf;
subplot(2, 1, 1);
plot(f, abs(H));
title('Low pass Butterworth filter');
axis([0 fs/2 0 1.2]);
xlabel('frequency in Hertz');
ylabel('gain');
subplot(2, 1, 2);
plot(f, 180*angle(H)/pi);
axis([0 fs/2 -180 180]);
xlabel('frequency in Hertz');
ylabel('Phase in degrees');
```

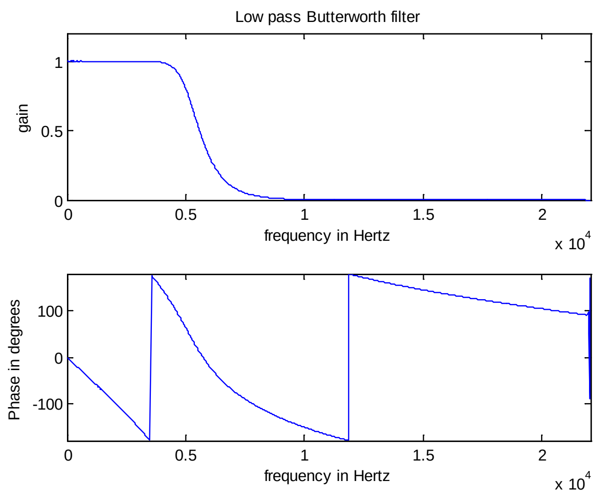


Figure 6.6
A low pass Butterworth filter. The order is 7. For the MATLAB® code see Appendix D for syntax of the `buttord`, `butter`, and `freqz` functions.

6.2 Stability of IIR Filters

• For a linear system, if the input is zero, the steady state output should be zero or, it should settle toward zero.
• Putting an impulse into a stable filter should result in a response which approaches zero after some, possibly long, period of time.
• Mathematically, we can write the transfer function for a filter as:

$$H(z) = k \cdot \frac{b_0 z^M + b_1 z^{M-1} + \cdots + b_{M-1} z + b_M}{z^N + a_1 z^{N-1} + \cdots + a_{N-1} z + a_N}$$

In this equation k is a gain constant and M ≤ N for a causal system.

$$H(z) = k\left(k_0 + \frac{k_1 z}{z - p_1} + \frac{k_2 z}{z - p_2} + \cdots + \frac{k_N z}{z - p_N} \right)$$

In this equation p_i is a pole of H(z) and k_i is a constant. If M < N then $k_0 = 0$.

Each of the terms in this expansion has an inverse z transform of the form:

$$\frac{k_i z}{z - p_i} \leftrightarrow k_i p_i^n$$

When looking at this term for stability consideration we see that there are three cases to consider:

Case 1: $|p_i| < 1$

In this case we see that as n gets larger the value of p_i^n approaches zero the system is said to be stable.

Case 2: $|p_i| = 1$

If the pole is exactly equal to 1, the value of p_i^n does not approach zero but neither does it grow without bound. The system is said to be marginally stable. For a filter with a pole whose magnitude is 1, the system needs no input to produce an output. Any noise disturbance (including numeric round off noise) or non-zero initial conditions will produce an output forever with no need for an input. A system with an output but no input is an oscillator. The filter oscillates indefinitely.

Case 3: $|p_i| > 1$

If the pole of H(z) has a magnitude greater than one, the value of p_i^n grows without bound and the system is said to be unstable.

Since the poles may be complex we can conclude that a filter having a pole outside a circle of radius 1 in the z plane is unstable.

Example 6.5

Consider a first order system in z with a transfer function in the form of $H(z) = \frac{Y(z)}{U(z)} = k\frac{z - z_1}{z - p_1}$ where z_1 is a real number representing a zero of the function, p_1 is a real number representing a pole of the function, and *k* is a gain constant. We can convert this transfer function to a difference equation by writing it in negative powers of *z*, cross multiplying, and taking the inverse transform using the time shifting property. This operation results in the following equation: $y(n) = ku(n) - kz_1u(n-1) + p_1y(n-1)$

Take $z_1 = 1$ and $k = 1$ and plot the impulse response for this system when $p_1 = 0, 0.3, 0.8, 0.95, 1.0$, and 1.2.

Solution

```
%IIRImp.m
T = 1;                  % Sample period
r = [0 .1 .3 .8 .95 1.0 1.2];
num = [1 1];            % numerator vector
figure(1);clf;
for i = 1:7
  den = [1 -r(i)];     % denominator vector
```

```
  [h nT] = impz(num, den, 40);
  subplot(7, 2, 2*i-1);
  stem(nT, h, 'MarkerEdgeColor', 'w');% stem plots the result
  ylabel('h(nT)');
  subplot(7, 2, 2*i);
  zplane(num, den);
  ylabel('Imag');
end
```

Instability is best shown in an impulse or step response as in Figure 6.7. Having poles outside the unit circle does not show up in the typical gain versus frequency response. If a system is unstable but linear then a sinusoidal input will produce a sinusoidal output at the same frequency.

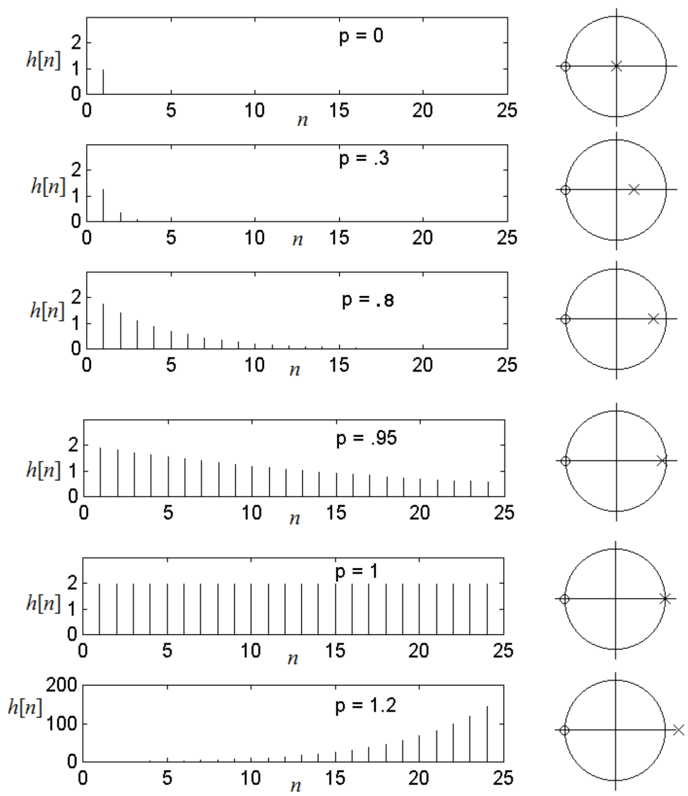


Figure 6.7

The impulse response for a first order system with a zero at z = -1 and a pole which moves from 0 to 1.2.

6.3 Frequency transformations

• Filter design often begins with the design of a *normalized* low pass filter.
• *normalize* means we have chosen a cutoff frequency $\omega_c = 1$ in the analog filter case or we choose a sample frequency $f_s = 2\pi$ in the digital case.
• To "unnormalize" an analog filter it is necessary to replace $s = j\omega$ by s/ω_c.
• This remapping of the frequency axis is referred to as a low pass to low pass transformation. Its effect is to create a low pass filter with a cut off frequency at a new value.
• Similar remapping equations can be developed for creating other filter types from low pass filters.

To illustrate

$$H_L(s) = \frac{1/(R_L C_L)}{s + 1/(R_L C_L)} \tag{6.7}$$

and

$$H_H(s) = \frac{s}{s + 1/(R_H C_H)} \tag{6.8}$$

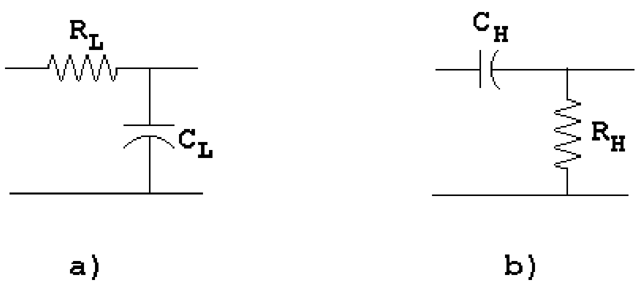


Figure 6.8
a) A low pass RC circuit and b) a high pass RC circuit.

In (6.7) we can replace s by λ/s. The new remapped equation becomes

$$H_R(s) = H_L(s)\big|_{s \leftarrow \lambda/s} = \frac{s}{s + \lambda R_L C_L} \tag{6.9}$$

Equation (6.9) has the same form as (6.8). Thus by choosing the proper value of λ we can use $s \leftarrow \lambda/s$ as a frequency transformation equation for a low pass to high pass conversion.

With a bit of algebra we can find other transformation equations. These are listed in Table 6.2.

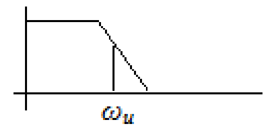
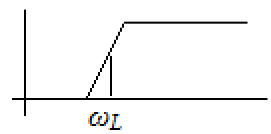
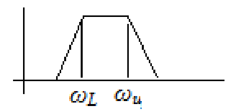
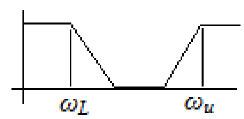
Transformation	Mapping Function	New Filter
low pass to low pass	$s \leftarrow \omega_c s / \omega_u$	
low pass to high pass	$s \leftarrow \omega_c \omega_L / s$	
low pass to band pass	$s \leftarrow \omega_p \dfrac{s^2 + \omega_L \omega_u}{s(\omega_u - \omega_L)}$	
low pass to band stop	$s \leftarrow \omega_p \dfrac{s(\omega_u - \omega_L)}{s^2 + \omega_u \omega_L}$	

Table 6.2
Analog frequency transformations. The low pass filter that is transformed has a cut off frequency of ω_c and a pass band edge at ω_p. The new cut-off frequencies are ω_u for the low-pass cut-off and ω_L for the high-pass cut-off.

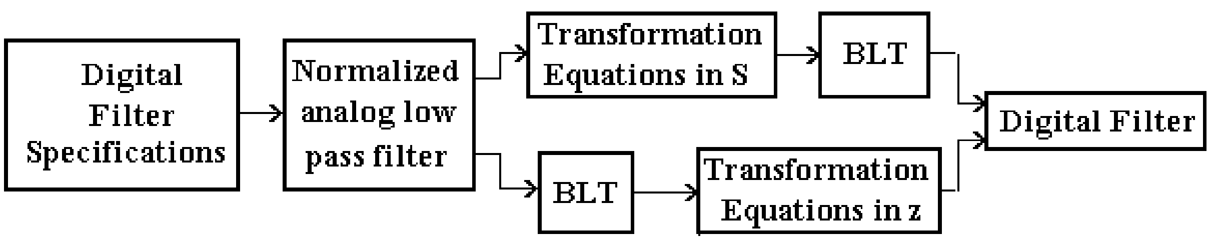


Figure 6.9
The design process for getting a digital filter from specifications by way of frequency transformations.

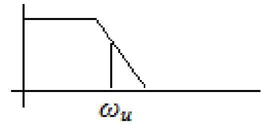
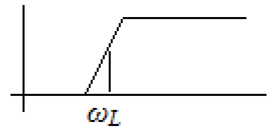
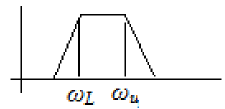
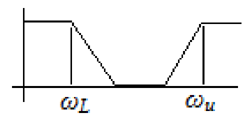
Transform	Mapping Function	Constants	New Filter
low pass to low pass	$z \leftarrow \frac{z-\alpha}{1-\alpha z}$	$\alpha = \frac{\sin[(\omega_c - \omega_u)T/2]}{\sin[(\omega_c + \omega_u)T/2]}$	
low pass to high pass	$z \leftarrow \frac{\alpha - z}{1-\alpha z}$	$\alpha = \frac{\cos[(\omega_L + \omega_c)T/2]}{\cos[(\omega_L - \omega_c)T/2]}$	
low pass to band pass	$z \leftarrow \frac{-z^2 + \frac{2\alpha k}{k+1} z - \frac{k-1}{k+1}}{\frac{k-1}{k+1} z^2 - \frac{2\alpha k z}{k+1} + 1}$	$\alpha = \frac{\cos[(\omega_u + \omega_L)T/2]}{\cos[(\omega_u - \omega_L)T/2]}$ $k = \frac{\tan(\omega_c T/2)}{\tan[(\omega_u - \omega_L)T/2]}$	
low pass to band stop	$z \leftarrow \frac{z^2 - \frac{2\alpha}{k+1} z + \frac{1-k}{1+k}}{\frac{1-k}{1+k} z^2 - \frac{2\alpha z}{k+1} + 1}$	$\alpha = \frac{\cos[(\omega_u + \omega_L)T/2]}{\cos[(\omega_u - \omega_L)T/2]}$ $k = \frac{\tan(\omega_c T/2)}{\cot[(\omega_u - \omega_L)T/2]}$	

Table 6.3

Digital frequency transformations. The low pass filter that is transformed has cut-off frequency, ω_c. The symbols ω_u and ω_L designate upper and lower cut-off frequencies, respectively, in the new filter.