ELECTRONOTES
1 Pheasant Lane
Ithaca, NY 14850
(607)-273-8030

APPLICATION NOTE NO. 336

January 1996

LOW-PASS TO BAND-PASS TRANSFORMATIONS ANALOG AND DIGITAL APPROACHES

0. INTRODUCTION

In the previous application note, we discussed the conversion of low-pass filters to high-pass filters by various means. Here we will do some of the same things for low-pass to band-pass transformations. Again we will find that we can consider transformations in both the analog and the digital domains. One major complication is that the order of the filters we end up with will in general be twice that of the prototype. This is best understood in terms of the diagram of Fig. 1 where we see that a low-pass response, having both a positive and negative side, is translated upward away from a center about zero to a new center (the response may also be spread out or shrunk as well). At the same time, a new negative side (corresponding to the positive side in terms of its requirements) is generated.

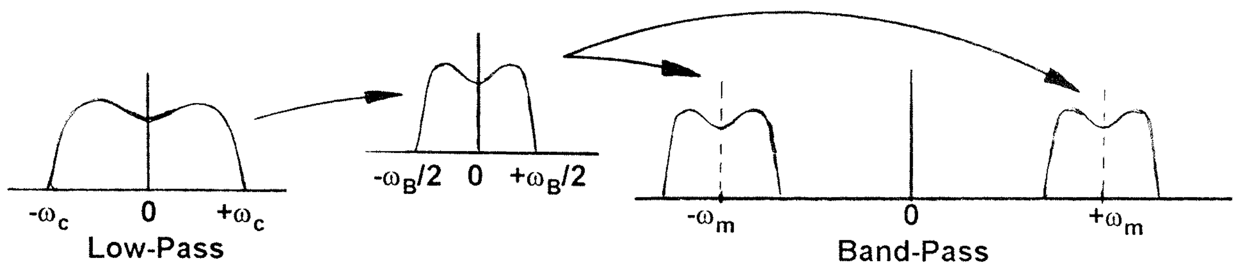


Fig. 1 Low-Pass to Band-Pass

1. ANALOG LOW-PASS TO BAND-PASS

In order to arrive at a basic notion of a low-pass to band-pass mapping within the s-plane, we will look at a simple low-pass to band-pass conversion at the component level. Fig. 2a shows our familiar first-order low-pass R-C filter. We can convert this easily to a band-pass by putting an inductor in parallel with the capacitor (Fig. 2b).

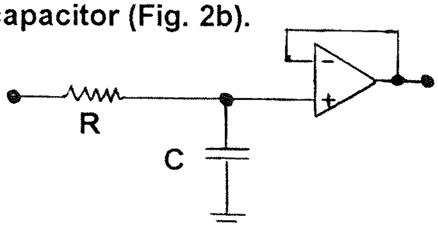


Fig. 2a Low-Pass

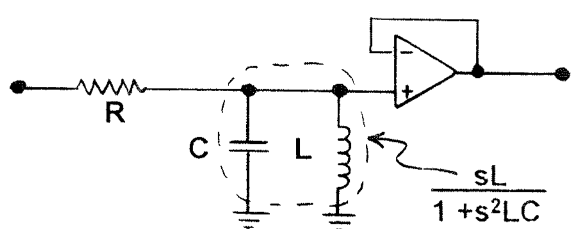


Fig. 2b Band-Pass

The transfer function of the band-pass is obtained from the voltage divider:

$$T_B(s) = [sL/(1+s^2LC)] / [\, R + \quad sL/(1+s^2LC) \,]$$

$$= (s/RC) / [s^2 + (\omega_m/Q)s + \omega_m{}^2] \qquad (1)$$

where the frequency of maximum response (peak) is:

$$\omega_m{}^2 = 1/LC \qquad (2)$$

and:

$$Q = R(C/L)^{1/2} \qquad (3)$$

This puts the result in the form of a standard band-pass filter. The component substitution has resulted in a second-order band-pass based on a first-order low-pass, as suggested in the introduction. Note that when $s = j\omega_m$, we find that $T_B(s)=1$. This is consistent with our physical understanding of the L-C parallel circuit having infinite impedance at the frequency $\omega=(LC)^{-1/2}$, in which case there is no current flowing through the resistor R, and hence no voltage drop from input to output at that frequency. The above analysis is perhaps abbreviated and is mainly for orientation. It is not essential for what follows, which is based upon the component substitution.

We see that the substitution of a parallel combination of a capacitor and an inductor for a capacitor suggests the impedance substitution:

$$1/s_LC \quad \longleftarrow \quad (1/s_BC)(s_BL) / [\, (1/s_BC) + s_BL \,] \qquad (4)$$

Bringing in the ω_m notation, we can rewrite equation (4) as:

$$s_L \longleftarrow (\omega_m{}^2 + s_B{}^2)/s_B \qquad (5)$$

This is almost the standard form of the substitution that is generally employed. Here we will add one more parameter, ω_B, as:

$$s_L \longleftarrow (\omega_m{}^2 + s_B{}^2)/s_B\omega_B \qquad (6)$$

Another way to write this is as:

$$\omega_B s_L \longleftarrow (\omega_m{}^2 + s_B{}^2)/s_B \qquad (7)$$

Here while we have written ω_B as though it were a radial frequency with units of inverse seconds, it probably is better thought of as a dimensionless scaling factor that relates width in the low-pass plane to width in the band-pass plane. Indeed, we will treat it as the bandwidth of the band-pass filter.

If we multiply out equation (7) and solve for s_B, we get:

$$s_B = [\, s_L\omega_B \pm (s_L^2\omega_B^2 - 4\omega_m^2)^{1/2} \,] / 2 \qquad (8)$$

which is nothing more than an application of the quadratic formula. We note the ± sign which gives us two value of s_B for each s_L. (Note that we end up with pairs of poles at the same angle in the s-plane.) This again shows the doubling of the order that we expect in going from low-pass to band-pass. Note also that the quantity inside the square root of equation (8) is already, in general, complex. Accordingly, any program we write must be able to handle this possibility.

Program 1 below, abp.m, is a MATLAB™ program for an analog band-pass. This program has only four parameters to specify. The specification begins with the order of the prototype analog low-pass, N. The resulting band-pass will have twice this order, 2N. The second parameter, r, is a reduction factor for the real part of the Butterworth poles. The design of the low-pass prototype begins with a Butterworth low-pass. This Butterworth low-pass may of course be used as is for a flat passband, or it can be converted to a Chebyshev (equi-ripple) low-pass. The conversion to Chebyshev is a matter of multiplying the real part of all the Butterworth poles by some reduction factor r where $1 \geq r > 0$. To keep Butterworth, r is specified as 1. While there is a mathematical relationship between the ripple and the reduction factor r, trial and error is certainly practical for this case, especially given that ripple is something we are willing to tolerate rather than something we actually want. Perhaps surprisingly, r needs to be relatively small for significant ripple. For our example, r = 0.3 gives slightly less than 10% ripple (Fig. 3c). The remaining two input parameters are the lower and upper cutoff frequencies of the band-pass filter. Note that we use these to generate the parameters ω_m and ω_B that we need to implement equation (8).

Fig. 3 shows the graphical results of a run of the program using the command line:

```
[bp,bz,bd,bn]=abp(5, .3, 1, 2.5)                                   (9)
```

The output parameters of the function are the band-pass poles, the band-pass zeros, the band-pass denominator, and the band-pass numerator, as well as the plots shown. The low-pass plots are shown mainly for reference purposes. In the case of Butterworth, and to a good approximation for the case of Chebyshev low-pass prototypes, the low-pass cutoff will be close to 1.0 (Fig 3c).

The doubling of the order as we go from low-pass to band-pass is evident from Fig. 3b and Fig. 3d. In Fig. 3b, we have a pole/zero plot for the analog band-pass. Note that we now have 10 poles total, in two "arrays" of five each, displaced in the imaginary dimension. (Keep in mind that Fig. 3a and Fig. 3b represent the s-plane). Further, we see that five zeros have appeared at s=0. Previously the low-pass had five zeros at infinity. The band-pass has five zeros at infinity as well.

PROGRAM 1: abp.m - ANALOG BANDPASS DESIGN

```
function [bp,bz,bd,bn]=abp(N,r,wl,wu)
%   function [bp,bz,bd,bn]=abp(N,r,wl,wu)
%
%              ANALOG BAND-PASS
%
%       N       = order of low-pass prototype
%                 (order of band-pass = 2N)
%       r       = reduction factor for Butterworth poles
%       wl      = lower band-pass cutoff
%       wu      = upper band-pass cutoff
%
%       example:   [bp,bz,bd,bn]=abp(5,.3,1,2.5)
%                  Gives a 10th-order bandpass with about
%                  10% ripple with passband from 1 to 2.5.

%  B. Hutchins                                     Fall 1995

wm=sqrt(wu*wl);   % max (center) frequency
wb=wu-wl;         % bandwidth

% get Chebyshev low-pass prototype
sa=pi/N;
for k=1:N
   poleang=pi/2 +(k-1)*sa + sa/2;
   lp(k)=r*cos(poleang) + j*sin(poleang);
end
lp=lp.'
figure(1);subplot(221)
plot(real(lp),imag(lp),'x')
axis([-1.2 .2 -1.2 1.2]);
grid
title('Analog LP Poles')

%     Low-Pass Magnitude response
ld=poly(lp);          % low-pass denominator
subplot(223)
w=0:.01:2;
H=abs(freqs([zeros(1,N-1),1],ld,w));
H=H/H(1);
plot(w,H)
axis([-0.2 2 -.1 1.5])
grid
title('Low-Pass')


% Low-Pass Poles ---> Band-Pass Poles/Zeros
llp=length(lp)
%   two BP poles and one zero for each LP pole
for k = 1:llp
     bp(k)= lp(k)*wb/2 +( (lp(k)^2*wb^2 - 4*wm^2)^(1/2) )/2;
     bp(llp+k)= lp(k)*wb/2 -( (lp(k)^2*wb^2 - 4*wm^2)^(1/2) )/2;
     bz(k)=0;
end
subplot(222)
plot(bp,'x')
hold on
plot(real(bz),imag(bz),'o')
grid
axis([-1.2 .2 -wm-wb-1 +wm+wb+1])
title('Band-Pass Poles/Zeros')
hold off
```

```
% Band-Pass Magnitude Response
bn=poly(bz);              % band-pass numerator
bd=poly(bp);              % band-pass denominator
w=0:.01:2*wm+2*wb;
subplot(224)
HB=abs(freqs(bn,bd,w));
HB=HB/abs(freqs(bn,bd,wm));
plot(w,HB)
axis([-.1 wm+wb+1 -.1 1.5]);
grid
title('Band-Pass')
```

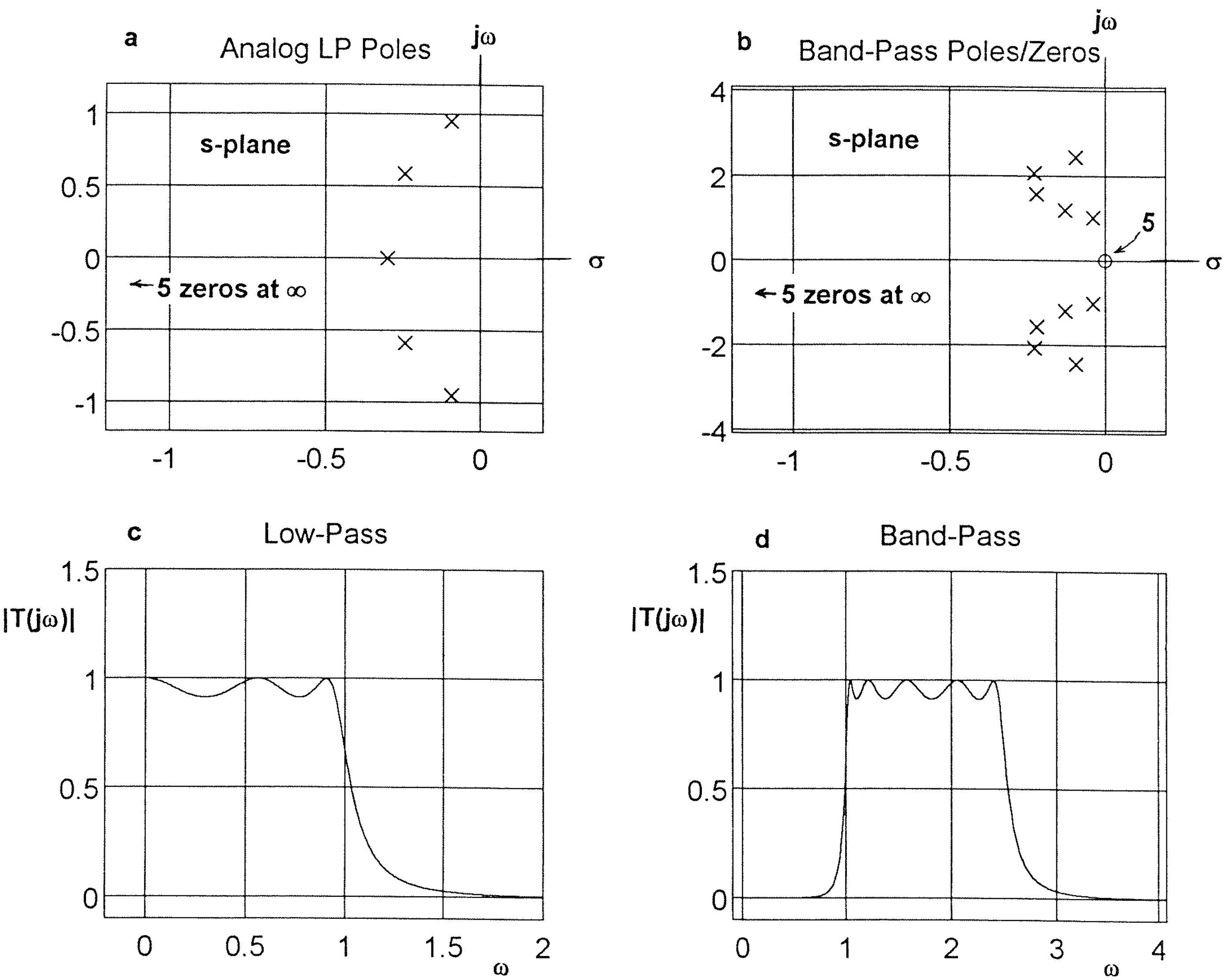


Fig. 3 Analog Low-Pass to Band-Pass

By looking at the magnitude response of the band-pass, Fig. 3d, we also see that the order is doubled, since the band-pass response has five full ripples while the low-pass has 2.5 ripples. Another way to look at this is to say that the low-pass has five full ripples (five poles) when we count the negative side of the response not shown, and the band-pass has 10 full ripples (10 poles) when both sides of the response are considered. In this sense, the entire low-pass response, which is originally centered about zero, can be considered to have been translated upward in frequency (and stretched or compressed) and then replicated on the negative side (Fig. 1). Note that the upper and lower cutoff frequencies are pretty much as specified in the function call.

If we are thinking in frequency units, the units here are radians/second. Thus the original low-pass has a cutoff of 1 rad/sec, and the band-pass has a lower cutoff at 1 rad/sec and an upper cutoff of 2.5 rad/sec. This is important if we are considering the transfer function generated from bn and bd. Of course, these analog filters are easily scaled to any frequencies necessary.

2. IIR DIGITAL LOW-PASS TO BAND-PASS

One approach to designing digital band-pass filters would be to use a direct design such as any one of the familiar FIR design methods and programs. Another would be to convert a suitable analog low-pass bprototype to a corresponding analog band-pass using the methods of Section 1 above, and then form the corresponding digital, IIR, band-pass using Bilinear z-Transform. The method discussed here however is a transformation method from a digital low-pass to a digital band-pass [1-3].

This transformation is a mapping from a low-pass z-plane (z_L) to a band-pass z-plane (z_B):

$$z_L^{-1} \longleftarrow - (z_B^{-1} + a_1 z_B^{-1} + a_2) / (a_2 z_B^{-2} + a_1 z_B^{-1} + 1) \qquad (10)$$

where:

$$a_1 = -2\alpha K/(K+1) \qquad (11)$$

$$a_2 = (K-1)/(K+1) \qquad (12)$$

$$\alpha = \cos[(\omega_u+\omega_l)/2] \,/\, \cos[(\omega_u-\omega_l)/2] \qquad (13)$$

$$K = \cot[(\omega_u-\omega_l)/2]\tan(\omega_c/2) \qquad (14)$$

PROGRAM 2: dbp.m - DIGITAL LOW-PASS TO BAND-PASS

```
function [bp,bz,bn,bd]=dbp(N,r,wc,wl,wu)
%   function [bp,bz,bn,bd]=dbp(N,r,wc,wl,wu)
%
%   DIGITAL BAND-PASS FROM DIGITAL LOW-PASS
%
%   bp      =   band-pass poles
%   bz      =   band-pass zeros
%   bn      =   band-pass numerator
%   bd      =   band-pass denominator
%
%   N       =   order of low-pass (analog or digital) prototype
%   r       =   reduction factor for real part of Butterworth poles
%                 frequencies relative to sampling frequency = 1
%   wc      =   low-pass cutoff (0 to 0.5)
%   wl      =   lower bp cutoff (0 to 0.5)
%   wu      =   upper bp cutoff (0 to 0.5)
%
%                                N  r  wc  wl  wu
%   example:  [bp,bz,bn,bd]=dbp(5,.3,.10,.12,.30)
%      Designs a 10th-order digital Chebyshev with about 10%
%      ripple and with a bandwidth from 0.12 to 0.3 times the
%      sampling frequency.

%   B. Hutchins                    Fall 1995 (rev. Spring 1996)

sa=pi/N;                                % Compute
for k=1:N                               % analog Butterworth
   poleang=pi/2 +(k-1)*sa + sa/2;       % poles and reduce
   p(k)=r*cos(poleang) + j*sin(poleang); % real part for
end                                     % Chebyshev low-pass

ws=2*pi;                                % Bilinear z
wcwarp=(ws/pi)*tan(pi*wc/ws);           % to digital
p=p*wcwarp;                             % low-pass
fs=ws/(2*pi);                           % poles
lp=(2*fs + 2*pi*p)./(2*fs - 2*pi*p);    %

ln=[1 1];                   %   Add Zeros
for k=1:(N-1)               %
     ln=conv(ln,[1 1]);     %   Low-pass numerator
end                         %
lz=roots(ln);               %   Low-pass zeros
ld=poly(lp);                %   Low-pass denominator

%          PLOT DIGITAL LOW-PASS
figure(1)
subplot(221)
pzplot(ln,ld)
axis([-1.1 1.1 -1.1 1.1]);
title('Low-Pass Poles/Zeros')
subplot(223)
HL=abs(freqz(ln,ld,500));
HL=HL/HL(1);
plot([0:.001:.499],HL)
axis([-.05 .5 -.1 1.5]);
grid
title('Low-Pass Magnitude')
%           END OF LOW PASS DESIGN

wu=2*pi*wu;
wl=2*pi*wl;
wc=2*pi*wc;
```

```
%       Conversion Parameters
alpha = cos((wu+wl)/2) / cos((wu-wl)/2);
k     = cot((wu-wl)/2)*tan(wc/2);
a1    = -2*alpha*k/(k+1)
a2    = (k-1)/(k+1)

%       Compute Band-Pass Poles/Zeros
llp=length(lp);
lpi=lp.^(-1);

for m=1:llp
     rad=sqrt(a1^2*(lpi(m)+1)^2 - 4*(a2*lpi(m)+1)*(lpi(m)+a2));
     bp(m)    =(-a1*(lpi(m)+1)+rad)/(2*(a2*lpi(m)+1));
     bp(m+llp)=(-a1*(lpi(m)+1)-rad)/(2*(a2*lpi(m)+1));
     bz(m)=1;
     bz(m+llp)=-1;
end
bp=bp.^(-1);

bn=poly(bz);
bd=poly(bp);

%           PLOT DIGITAL BAND-PASS
subplot(222)
pzplot(bn,bd)
axis([-1.1 1.1 -1.1 1.1]);
title('Band-Pass Poles/Zeros')
subplot(224)
HB=abs(freqz(bn,bd,500));
m=round(500*sqrt(wu*wl)/pi)
HB=HB/HB(m);
plot([0:.001:.499],HB)
axis([-.05 .5 -.1 1.5]);
grid
title('Band-Pass Magnitude')
figure(1)
```

where ω_l and ω_u are the lower and upper cutoff frequencies of the digital band-pass, given in terms of a sampling frequency of 2π. [The actual numbers input to our program will be on the interval 0 to 0.5 as noted in the heading to the program. The multiplication by 2π is internal.] The parameter ω_c is the cutoff frequency of the prototype low-pass filter, and in general makes little difference to the final design. As with the analog-to-analog mapping, the low-pass is perhaps most interesting as a point of reference. Some users may prefer to modify the program with ω_c fixed. In such a case, a relatively small value of ω_c is suggested.
It remains to solve equation (10) using the quadratic formula, which gives us z_B in terms of z_L and the constants:

$$z_B^{-1} = \{ - a_1(z_L^{-1}+1) \pm [a_1^2(z_L^{-1}+1)^2-4(a_2z_L^{-1}+1)(z_L^{-1}+a_2)]^{1/2} \} \;/\; 2(a_2z_L^{-1}+1) \qquad (15)$$

The actual program here is Program 2, dbp.m, as shown. The program begins with the design of an analog low-pass prototype, as in Program 1. This prototype is then converted to a digital low-pass using Bilinear z-Transform, giving the

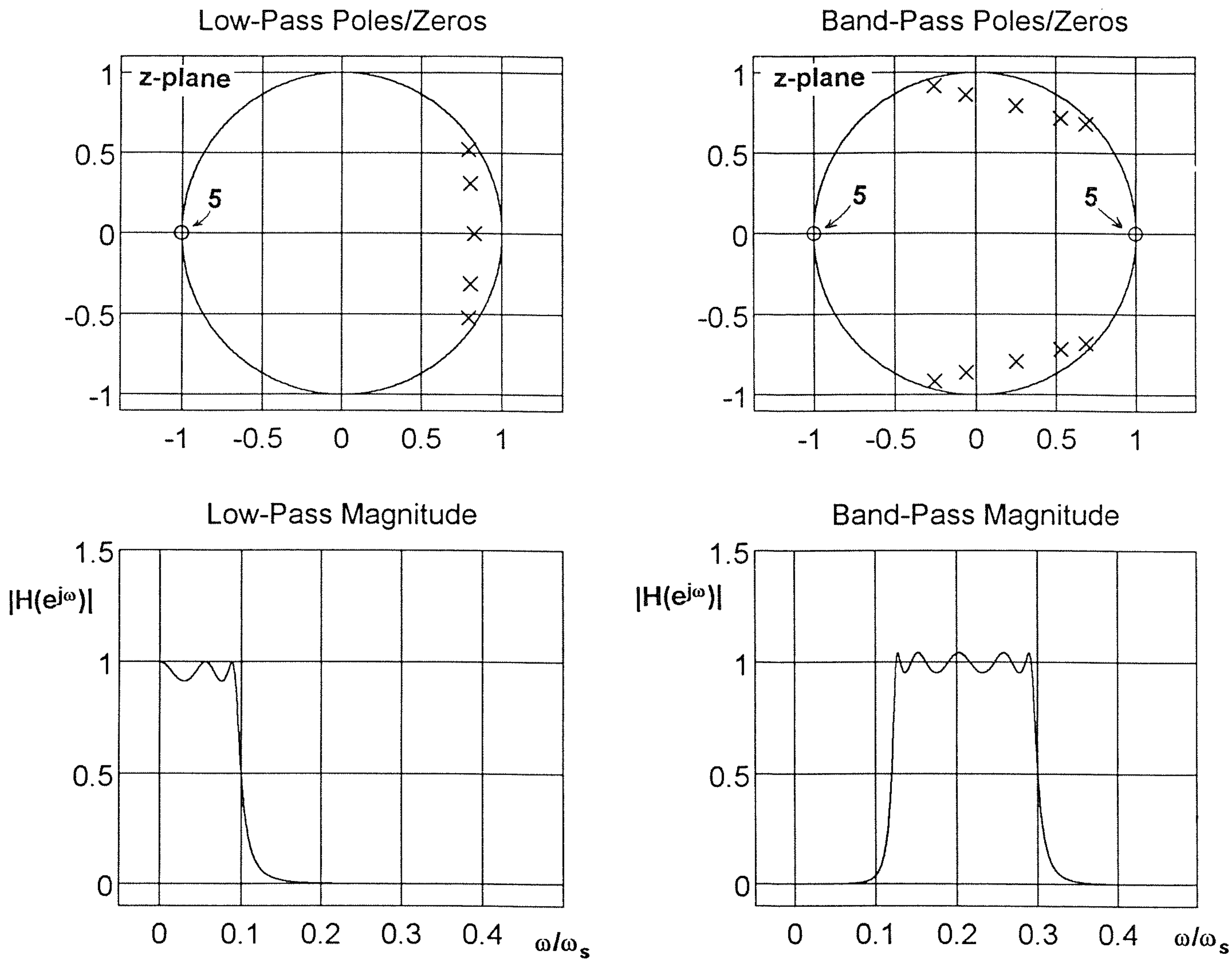


Fig. 4 Digital Low-Pass to Band-Pass

digital low-pass prototype to be converted to digital band-pass. The actual band-pass design is a straightforward implementation of equations (10) through (15).

Figure 4 shows an example corresponding to the suggested example in the heading to the program:

[bp,bz,bn,bd]=dbp(5,.3,.10,.12,.30) (16)

This begins with our familiar 5th-order low-pass with 10% ripple. The digital low-pass has a cutoff of 0.1 (0.2π actually), and as noted, is mainly of concern here for the illustration. The actual band-pass is to have a lower cutoff at 0.12 (0.24π) and an upper cutoff at 0.3 (0.6π).

The band-pass, Fig. 4b and Fig. 4d is similar to the analog case (Fig. 3) except here the two replications of the low-pass positioned are with reference to the unit circle, not the $j\omega$-axis. From the pole/zero plot we see the ten poles. There are also 5 zeros at z=1 (corresponding to the 5 zeros at s=0 for the analog case). Here, in addition, we have 5 zeros at z=-1. These are the 5 zeros that were at $s=\infty$ in the analog case, transformed exactly as we would have gotten by Bilinear z-Transform of the analog band-pass.

3. FIR DIGITAL LOW-PASS TO BAND-PASS?

Above we discussed the fact that FIR design methods for band-pass digital filters (and for most other types) are readily available, so we may well be content to use our low-pass to band-pass transformations only for IIR filters where they work well. But, do they in fact work for FIR? On the one hand, we might well note that FIR filters are a special case of IIR filters, and accordingly, a methods that works for IIR should also work for FIR. On the other hand, we might also note that IIR filters of a given order have poles as well as zeros, and these in effect constitute a doubling of the number of manipulable design parameters, a fact that could well indicate a significant difference. In fact, without the poles, the FIR low-pass to band-pass transformation is extremely limited (the poles of the FIR at z=0 of course have no effect on the response).

With some special setup, we can manipulate the FIR low-pass to band-pass transformation into working; pretty much as a curious illustration. As in many cases, it is useful to begin with the simplest "toy filter" that we can come up with. This is the two-tap moving-average FIR low-pass (Fig. 5). Here we have a transfer function:

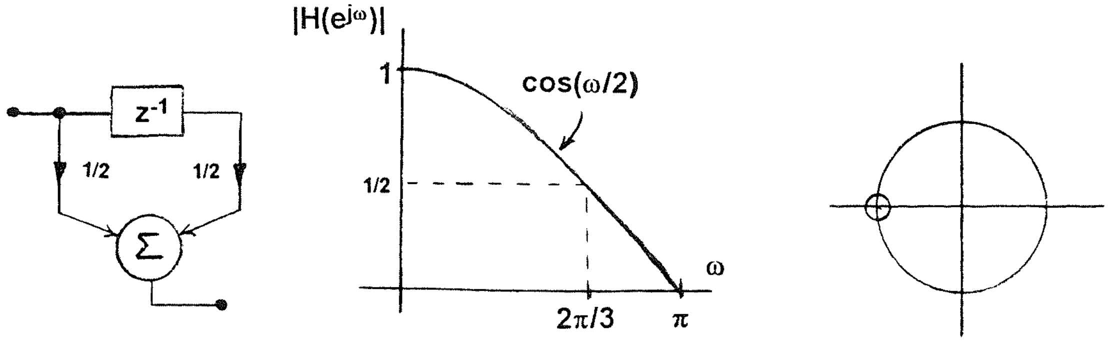
$$H_L(z) = 1/2 + (1/2)z^{-1} \tag{17}$$

with corresponding magnitude response:

$$|H_L(e^{j\omega})| = \cos(\omega/2) \tag{18}$$

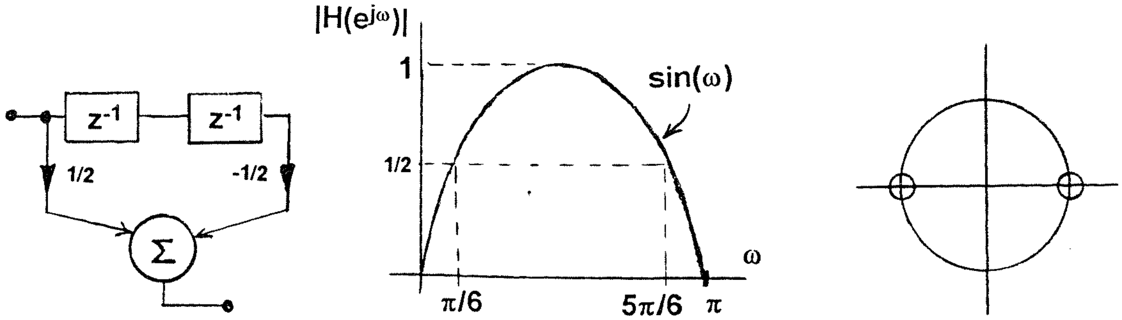
Fig. 5a shows the filter's structure, Fig. 5b the magnitude response, and Fig. 5c shows the zero at z=-1 in the z-plane. Note that we might choose $|H_L(e^{j\omega})|=1/2$ as a defined cutoff point (somewhat arbitrarily). This occurs [see equation (18)] at $\omega=2\pi/3$ (i.e., $\omega/\omega_s=1/3$).

Having familiarized ourselves with this low-pass setup, we need to imagine what could happen if we try to take the low-pass zero through the low-pass to band-pass transformation used above. We simply get two zeros instead of one. Where should these two zeros end up? Clearly we would have to have two real zeros, or a complex-conjugate pair of zeros. A pair of complex conjugate zeros is



(a) Structure (b) Magnitude Response (c) z-Plane

Fig. 5 Prototype FIR Low-Pass



(a) Structure (b) Magnitude Response (c) z-Plane

Fig. 6 Transformed to Band-Pass

clearly suitable to notch-like responses - pretty much the opposite of a band-pass. Accordingly, we consider two real zeros, and one obvious choice would be one zero at z=-1 and the other at z=+1 (Fig. 6). This filter has a transfer function:

$$H_B(z) = (1/2)(1 - z^{-1})(1 + z^{-1}) = 1/2 - (1/2)z^{-2} \quad (19)$$

with corresponding magnitude response:

$$|H_B(e^{j\omega})| = \sin(\omega) \quad (20)$$

Fig. 6a shows the filter structure, Fig. 6b shows the magnitude response, and Fig. 6c shows the zeros at z=-1 and at z=+1. Note that if we keep our cutoff value of $|H_B(e^{j\omega})| = 1/2$, we see from Fig. 6b that we have $\omega_l=\omega_s/12$ and $\omega_u=5\omega_s/12$ (at $\pi/6$ and at $5\pi/6$ for a sampling frequency of 2π).

Now, suppose that we had chosen these as the start of our design. That is, we have not yet looked at the above possibilities as outlined in Fig. 5 and Fig. 6. Thus we have a low-pass with a cutoff $\omega_c=2\pi/3$ and we want a band-pass with cutoff at $\pi/6$ and at $5\pi/6$. We use the transformation procedure for digital low-pass to digital band-pass as used above. Equations (13), (14), (11) and (12) yield:

$$\alpha = 0 \tag{21a}$$

$$K = 1 \tag{21b}$$

$$a_1 = 0 \tag{21c}$$

$$a_2 = 0 \tag{21d}$$

so we are clearly in the realm of some special case. Transforming the zero at $z_L = -1$ using equation (15) gives:

$$z_B = \pm 1 \tag{22}$$

the result we guessed, which suggests that this works, although we stacked the deck for sure.

Perhaps another way to see what has happened here is to consider how the impulse response of the low-pass, equation (17) can be manipulated to form the impulse response of the band-pass, equation (19). It is neither obvious nor well demonstrated by this short filter, but nonetheless correct, that we can convert the low-pass to a band-pass by multiplying the low-pass impulse response by $(-1)^n$ and then zero-padding the result. This is the same as modulating the low-pass response to center it around one quarter of the sampling frequency (typical samples: ... 0 1 0 -1 0 1 0 -1 .....). The zero-padding doubles the length of the time-domain description and thus compresses the width of the frequency-domain description by a factor of two, while at the same time rotating it by 90° on the unit circle. Hence we understand the change from $\cos(\omega/2)$ to $\sin(\omega)$.

While it is possible to continue to investigate special cases of this attempt to use the digital type-transformations on an FIR response, little more in the way of systematic calculation or practical insight results, and we again would likely just recommend direct design FIR programs.

REFERENCES

[1] A.G. Constantinides, "Spectral Transformations for Digital Filters," Proc. IEE, Vol. 117, No. 8, pp 1587-1593, August 1970

[2] A.V. Oppenheim & R.W. Schafer, Discrete-Time Signal Processing, Prentice-Hall (1989) pp 430-438

[3] J.G. Proakis & D.G. Manolakis, Digital Signal Processing; Principles, Algorithms, and Applications (2nd Ed), Macmillan (1992), pp 648-650