EE 274 / CoE 197E Filter Structures Lab Exercise

Rhandley D. Cajote

1. **FIR Filter Structures**. A causal FIR filter of length M is characterized by a transfer function H(z):

$$H(z) = \sum_{k=0}^{M-1} h(n) z^{-k}$$

which is a polynomial in z^{-1} of degree $M-1$. In the time domain the input-output relation of the above FIR filter is given by

$$y(n) = \sum_{k=0}^{M-1} h(k) x(n-k)$$

Where y(n) and x(n) are the output and input sequences, respectively.

a. **Cascade structure**. The factored form of a causal FIR transfer function H(z) of order M-1, can be determined from its polynomial form representation given by the above equations which can be utilized to realize H(z) in cascaded form. An example of how to convert the rational transfer function to its factored form or second order sections is shown.

```
% Program P6_1
% Conversion of a rational transfer function
% to its factored form
num = input('Numerator coefficient vector = ');
den = input('Denominator coefficient vector = ');
[z,p,k] = tf2zp(num,den);
sos = zp2sos(z,p,k);
```

You have to input a valid numerator and denominator polynomial coefficients. The rational H(z) must be proper, the order of the denominator is greater than the order of the numerator. Type help zp2sos to understand the format of the output.

b. Use the above example code to generate a cascade realization of the filter described by:

$$H_1(z) = 2 + 10z^{-1} + 23z^{-2} + 34z^{-3} + 31z^{-4} + 16z^{-5} + 4z^{-6}$$

Draw the block diagram of the cascade structure and plot the frequency response. Is the system linear phase? Compare the frequency response of the $H_1(z)$ using freqz() and the second-order (sos) cascade using fvtool(). Is there any noticeable difference?

c. Use the above example code to generate a cascade realization of the filter described by:

$$H_2(z) = 6 + 31z^{-1} + 74z^{-2} + 102z^{-3} + 74z^{-4} + 31z^{-5} + 6z^{-6}$$

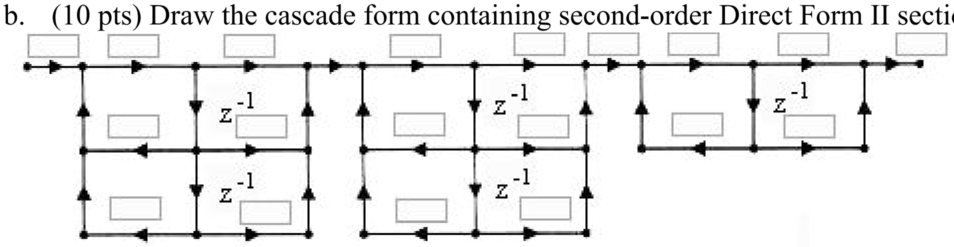
Is the system linear phase? Draw the block diagram of the cascade structure of $H_2(z)$ with fewer multipliers than $H_1(z)$. Is the system linear phase? Compare the frequency response of the $H_2(z)$ using freqz() and the second-order (sos) cascade using fvtool(). Is there any noticeable difference?

2. **IIR Filter Structures**. An IIR filter is described by the following function below. Determine the corresponding coefficient and redraw each filter structure with the computed coefficients. Compute and plot the frequency response of the system. Plot the response to an input $x(n) = u(n)$ for $0 \le n \le 100$, or the step response using the corresponding structure in each case.

   Note: There are several ways on how to implement the filter in Matlab, these are just suggestions. You can use filtfilt() to filter data using second order sections. You can use filter() to filter data using rational polynomial coefficients of H(z). You can also use the Matlab dfilt() package, this package can create a digital filter using the supported filter structures. Type >> help dfilt for the list of packages, for example type >> help dfilt.df1sos on how to create a digital filter using second-order section direct form 1.

$$H(z) = 2\left(\frac{1 + 0z^{-1} + z^{-2}}{1 + 0.8z^{-1} + 0.64z^{-2}}\right)\left(\frac{2 - z^{-1}}{1 - 0.75z^{-1}}\right)\left(\frac{1 + 2z^{-1} + z^{-2}}{1 + 0.81z^{-2}}\right)$$

   a. (10 points) Draw the Direct Form I and Direct Form II realizations of H(z). Use dfilt() package to create a filter using DFI or DF II.
   b. (10 pts) Draw the cascade form containing second-order Direct Form II section

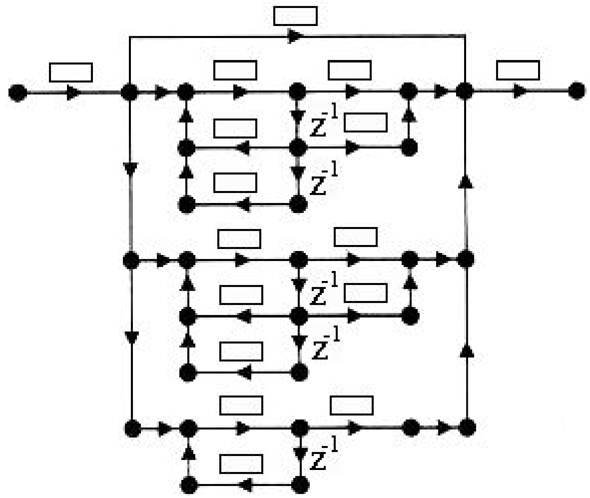


   c. (10 pts) Draw the Parallel form containing second-order Direct Form II sections.

   Use the residuez() and residues() function to determine the partial fraction expansion of the rational H(z) which you will use to determine the coefficients of the parallel filter section. For example:

```
% Program P6_2
% Parallel Form Realizations of an IIR Transfer
num = input('Numerator coefficient vector = ');
den = input('Denominator coefficient vector = ');
[r1,p1,k1] = residuez(num,den);
[r2,p2,k2] = residue(num,den);
disp('Parallel Form I')
disp('Residues are');disp(r1);
disp('Poles are at');disp(p1);
disp('Constant value');disp(k1);
disp('Parallel Form II')
disp('Residues are');disp(r2);
disp('Poles are at');disp(p2);
disp('Constant value');disp(k2);
```

   Use dfilt.parallel to create a digital filter composed of parallel section.



3. **(For Graduate Students only) Effects of coefficient quantization.**
   The effect of quantization can be demonstrated by the change in the magnitude response and in the location of the poles and zeros as illustrated by the code below.

```
% Program P9_1
% Coefficient Quantization Effects on Direct Form
% Realization of an IIR Transfer Function
clf;
[b,a] = ellip(6,0.05,60,0.4);
[g,w] = Gain(b,a);
bq = a2dT(b,5);aq = a2dT(a,5);
[gq,w] = Gain(bq, aq);
figure; plot(w/pi,g,'b', w/pi,gq,'r--');
axis([0 1 -80 1]);grid
xlabel('\omega /\pi');ylabel('Gain, dB');
legend('original', 'quantized');
pause
figure; zplane(b,a);
hold on
%zplane(bq, aq);
pzplot(bq,aq);
hold off
title('Original pole-zero locations: x, o; New pole-zero
locations: +, *')

function beq = a2dT(d,n)
% BEQ = A2DT(D, N) generates the decimal
% equivalent beq of the binary representation
% of a decimal number D with N bits for the
% magnitude part obtained by truncation
%
m = 1; d1 = abs(d);
while fix(d1) > 0
    d1 = abs(d)/(10^m);
    m = m+1;
end
```

```
beq = 0;
for k = 1:n
    beq = fix(d1*2)/(2^k) + beq;
    d1 = (d1*2) - fix(d1*2);
end
beq = sign(d).*beq.*10^(m-1);

function [g,w] = Gain(num,den)
% Computes the gain function in dB of a
% transfer function at 256 equally spaced points
% on the unit circle
% Numerator coefficients are in vector num
% Denominator coefficients are in vector den
% Frequency values are returned in vector w
% Gain values are returned in vector g
w = 0:pi/255:pi;
h = freqz(num,den,w);
g = 20*log10(abs(h));
```

The effect of quantization in the IIR implementation of a Direct Form II filter and a SOS can also be illustrated using the code below.

```
% Program P9_2
% Coefficient Quantization Effects on Cascade
% Realization of an IIR Transfer Function
clf;
[z,p,k] = ellip(6,0.05,60,0.4);
[b,a] = zp2tf(z,p,k);
[g,w] = Gain(b,a);
sos = zp2sos(z,p,k);
sosq = a2dR(sos,6);
R1 = sosq(1,:);R2 = sosq(2,:);R3 = sosq(3,:);
b1 = conv(R1(1:3),R2(1:3));bq = conv(R3(1:3),b1);
a1 = conv(R1(4:6),R2(4:6));aq = conv(R3(4:6),a1);
[gq,w] = Gain(bq, aq);
plot(w/pi,g,'b-',w/pi,gq,'r--');
axis([0 1 -80 20]);grid
xlabel('\omega /¬π');ylabel('Gain, dB');
title('original - solid line; quantized - dashed line');
%pause
figure; zplane(b,a);
hold on
pzplot(bq,aq);
hold off
title('Original pole-zero locations: x, o; New pole-zero
locations: +, *');
```

Modify the Matlab code P9_2 given above to investigate the effects of quantization on the filter coefficient on an eighth-order elliptic highpass transfer function with a passband ripple of 0.1 dB, a minimum stopband attenuation of 70 dB, and a normalized cutoff frequency at 0.55 rad/sec. Assign five bits to the fractional part of the binary representations. Run the modified Matlab code and compare the frequency response, pole-zero plot and filter output with quantized and unquantized coefficients. Comment on your results.