Marwin B. Alejo 2020-20221
EE274_ProgEx05

Also accessible through www.github.com/soymarwin/ee274/EE274_ProgEx05 for history tracking.

Disclaimer: For variable mapping, I use the section number in reverse followed by the actual var name in the original code. This is to ensure that variables in each section does not share the same memory hence, latter codes will not be affected by preceding sections. Each code must be executed section by section as Simulink does not work as is with procedural execution – publish command. Also, all filter realizations are generated with auto-routing in each node as such blocks and details are visible on screen.

1. FIR Filter Structure

a. Cascade Structure

```
% tf2zp() require input vectors of the same length hence, the original code
% must be modified as shown below.
a1_num = input('Numerator coef vector = ');
a1_den = input('Denominator coef vector = ');
[a1_b, a1_a] = eqtflength(a1_num, a1_den);
[a1_z, a1_p, a1_k] = tf2zp(a1_b, a1_a);
a1_sos = zp2sos(a1_z, a1_p, a1_k);
```

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

```
b1_num = [2 10 23 34 31 16 4];
b1_den = 1;
[b1_b, b1_a] = eqtflength(b1_num, b1_den);
[b1_z, b1_p, b1_k] = tf2zp(b1_b, b1_a);
b1_sos=zp2sos(b1_z, b1_p, b1_k);
[b1h,b1w]=freqz(b1_num,b1_den);
plot(b1w/pi,20*log10(abs(b1h))); title('Frequency Response of H_{1}(z)');
xlabel('Normalized Frquency (\times\pi rad/sample)'); ylabel('Magnitude(dB)');
fvtool(b1_sos);
b1_cscd=dfilt.dffir(b1_sos);
realizemdl(b1_cscd); % see b1_cscd_mdl.slx for complete cascade generalization diagram of H1(z)
```

Generalization: $H_1(z)$ is not a linear-phase transfer function due to its coefficients not having the required symmetry. Moreover, there is no clear or noticeable difference between the plots generated using freqz() and fvtool() except that the plot in freqz() (figure 1) is from edge to edge (0:1) while in fvtool() (figure 2) is from 0 to ~1. Figures 3 to 5 show the cascade diagram of $H_1(z)$. Also, this cascade diagram is digitally stored in b1_cscd_mdl.slx MATLAB object.

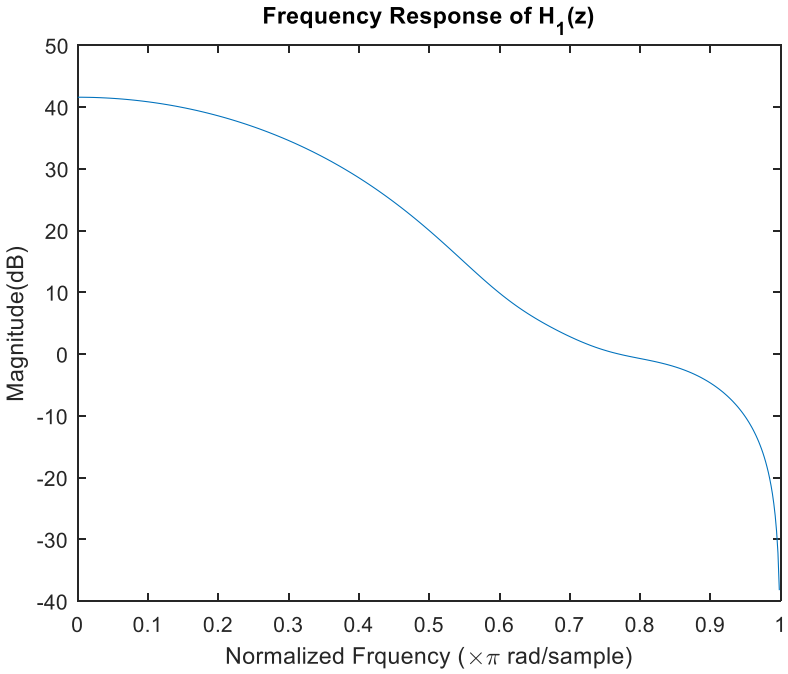

Figure 1. H1(z) FR using freqz()

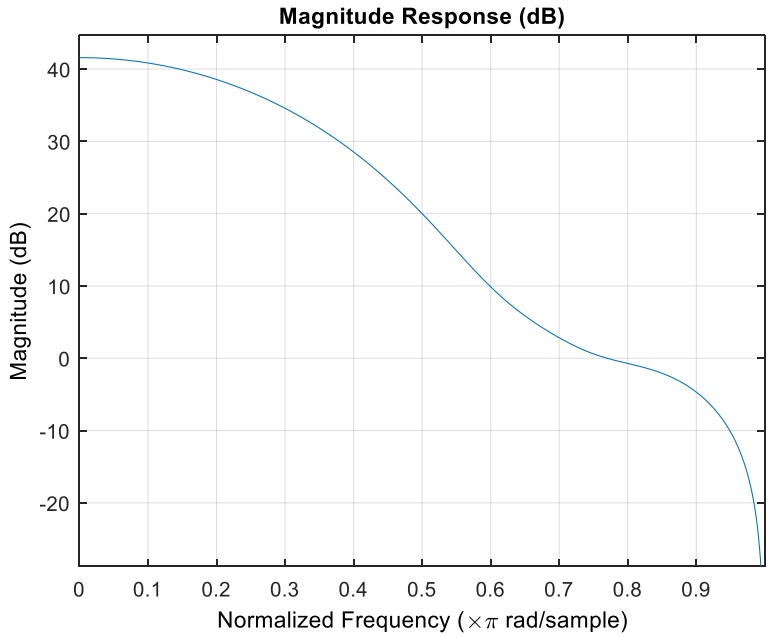

Figure 2. H1(z) FR using fvtool()

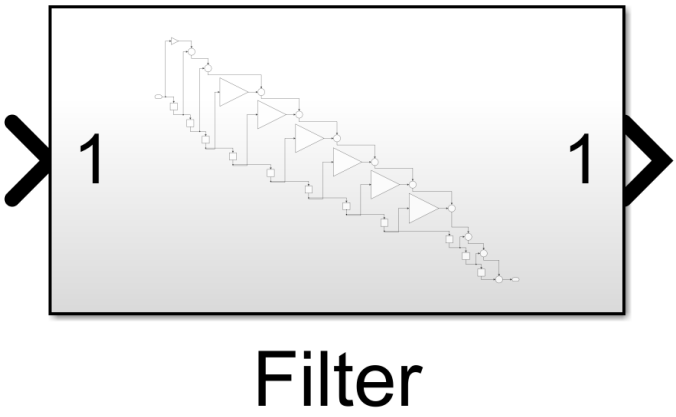

Figure 3. H1(z) cascade diagram (filter inside)

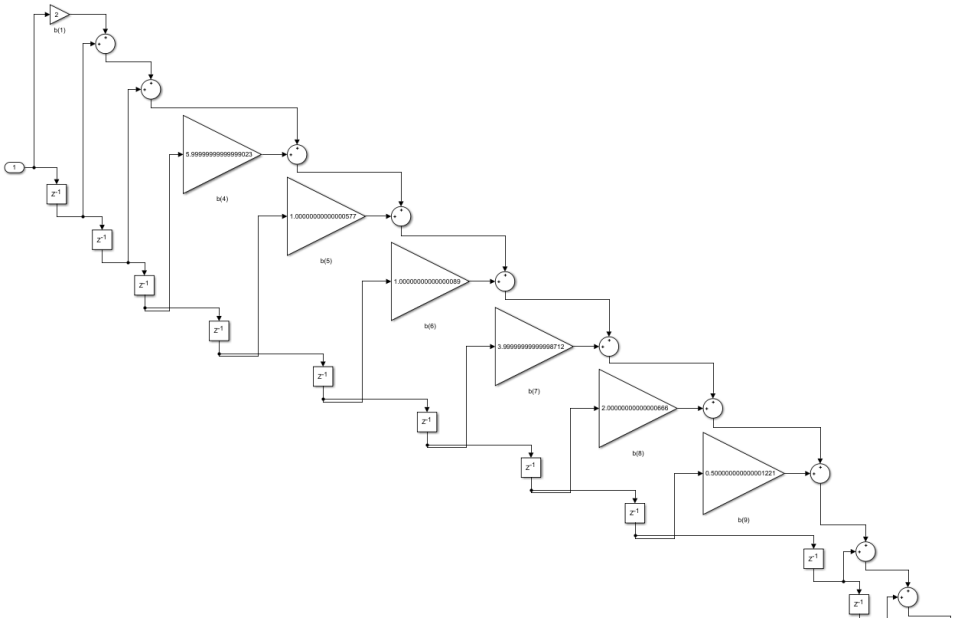

Figure 4. H1(z) cascade diagram (filter)

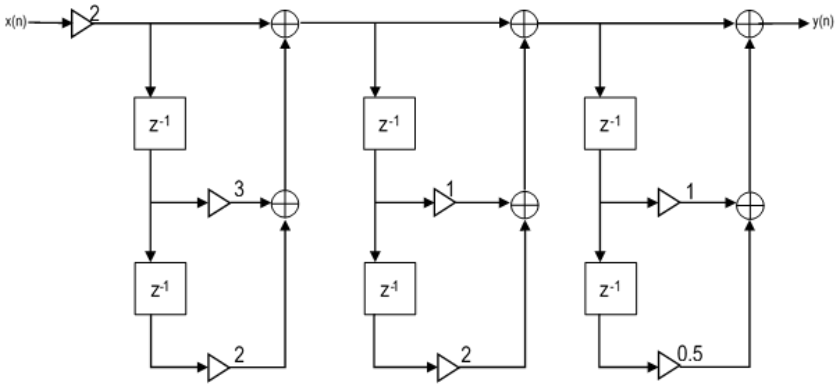

Figure 5. H1(z) Cascade diagram (simplified)

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

```
c1_num = [6 31 74 102 74 31 6];
c1_den = 1;
[c1_b, c1_a] = eqtflength(c1_num, c1_den);
[c1_z, c1_p, c1_k] = tf2zp(c1_b, c1_a);
```

```
c1_sos = zp2sos(c1_z, c1_p, c1_k);
[c1h,c1w]=freqz(c1_num,c1_den);
plot(c1w/pi,20*log10(abs(c1h)));title('Frequency Response of H_{2}(z)');
xlabel('Normalized Frquency (\times\pi rad/sample)'); ylabel('Magnitude(dB)');
fvtool(c1_sos);
c1_cscd=dfilt.dffir(c1_sos);
realizemdl(c1_cscd); % see c1_cscd_mdl.slx for complete cascade generalization diagram of H2(z)
```

Generalization: $H_2(z)$ is a linear-phase transfer function (type-1) with an odd length and even symmetry. Moreover, there is no clear or noticeable difference between the plots generated using freqz() (figure 6) and fvtool() (figure 7). Figures 8 to 10 shows the cascade diagram of $H_2(z)$. Also, this cascade diagram is digitally stored in c1_cscd_mdl.slx MATLAB object.

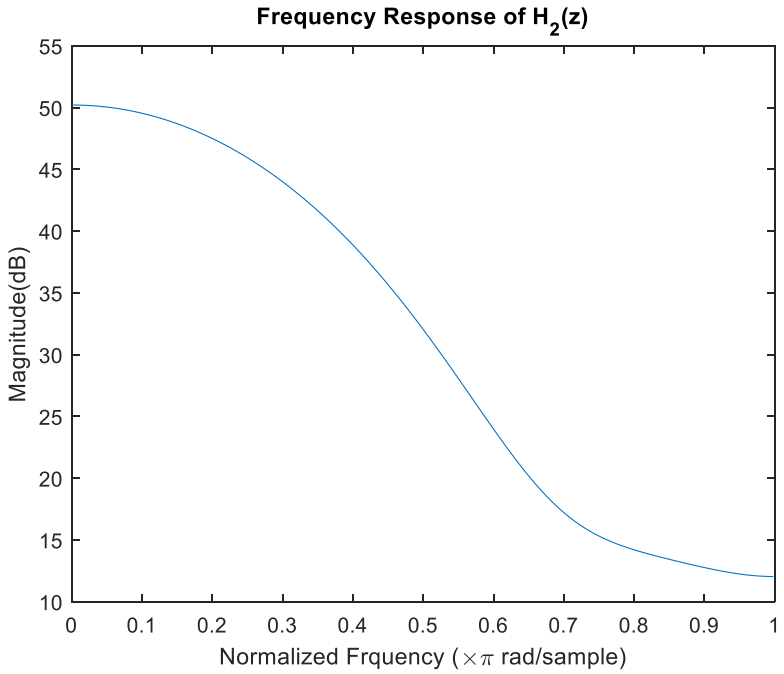

Figure 6. H2(z) FR using freqz()

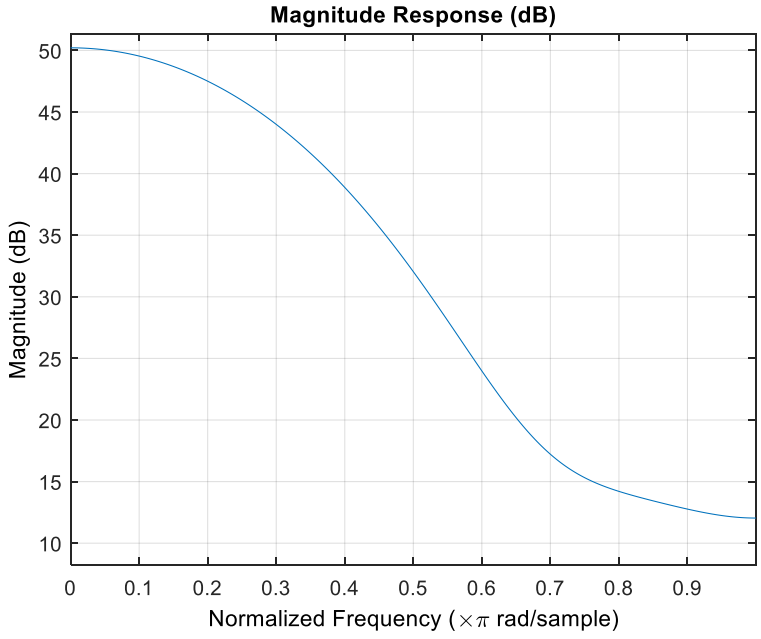

Figure 7. H2(z) FR using fvtool()

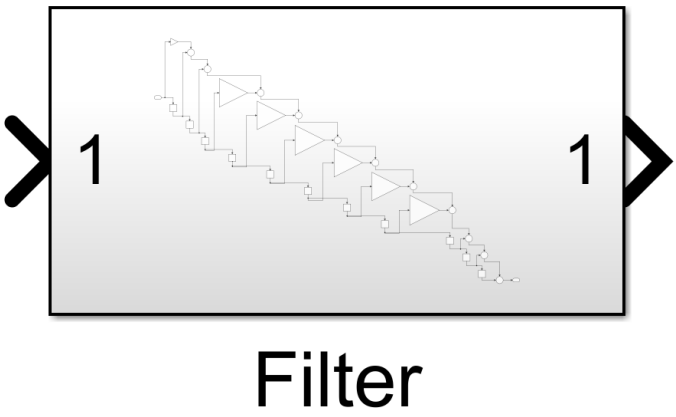

Figure 8. H2(z) cascade diagram (filter inside)

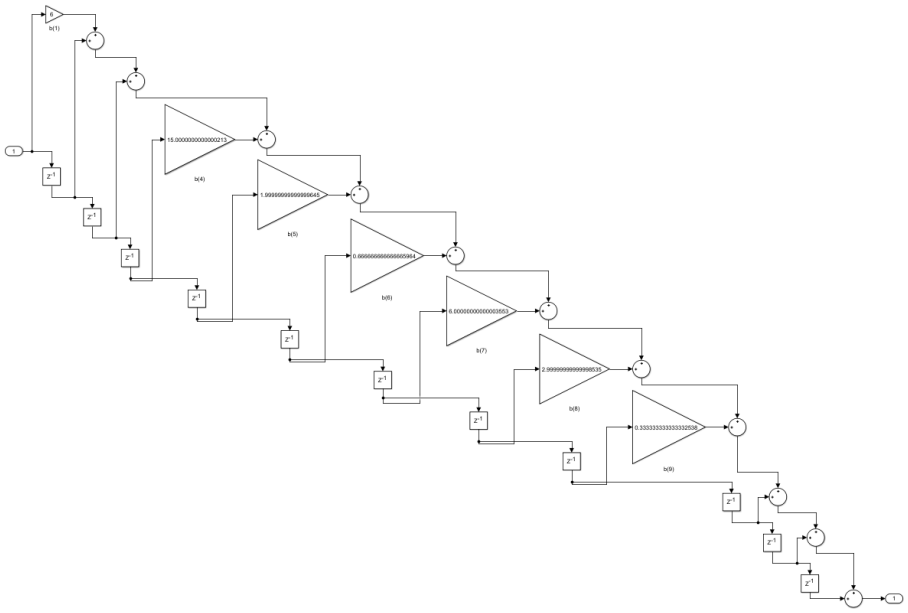

Figure 9. H2(z) cascade diagram (filter)

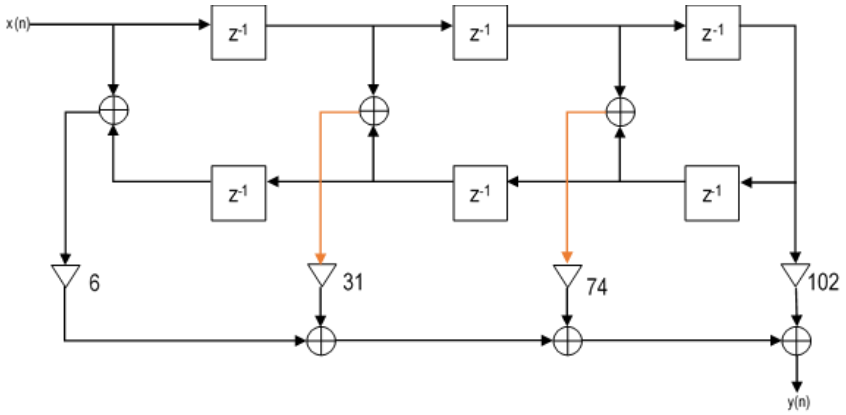

Figure 10. H2(z) Cascade diagram (simplified)

2. IIR Filter Structure

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

```
% From TF
fb0_2=2; fb1_2=[1 2 1]; fb2_2=[0 1 -1 2 1];
fa1_2=[1 1 1]; fa2_2=[0.8 0.64 -0.75 0 0.81];
fb_2=fb0_2*conv(fb1_2,fb2_2); fa_2=conv(fa1_2,fa2_2);
[fh_2,fw_2]=freqz(fb_2, fa_2,'whole',101); figure(), plot(fw_2/pi,abs(fh_2));
title('Frequency Response of H(z) with 0 \leq n \leq 100');
xlabel('Normalized Frquency (\times\pi rad/sample)'); ylabel('Magnitude(rad)');
```

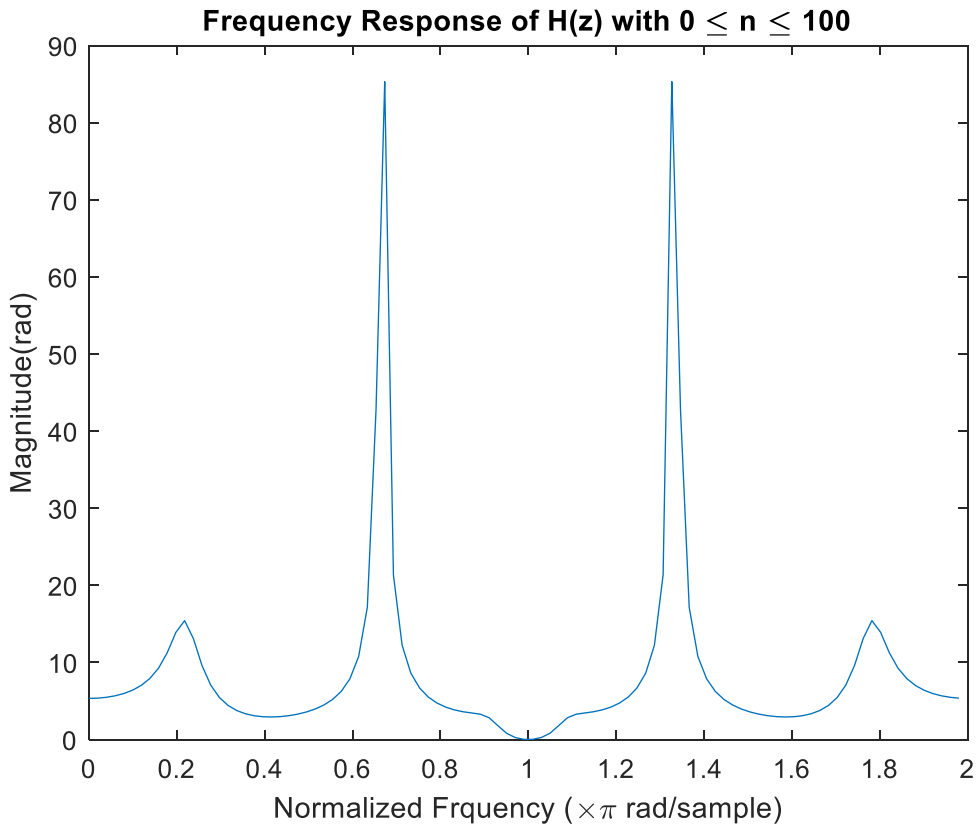

Figure 11. H(z) Frequency Response

a. Draw DF1 and DF2 realizations of H(z)

```
a2_df1=dfilt.df1(fb_2,fa_2);
a2_df2=dfilt.df2(fb_2,fa_2);
realizemdl(a2_df1); % see a2_df1_mdl.slx for complete DF1 generalization diagram of H(z)
realizemdl(a2_df2); % see a2_df2_mdl.slx for complete DF2 generalization diagram of H(z)
```

Figure 12 and figure 13 shows the DF1 realization of H(z). Moreover, a2_cscd_mdl.slx contain the following below. On the other hand, figure 14 and figure 15 shows the DF2 realization of H(z). Moreover, a2_df2_mdl.slx contain the following below.

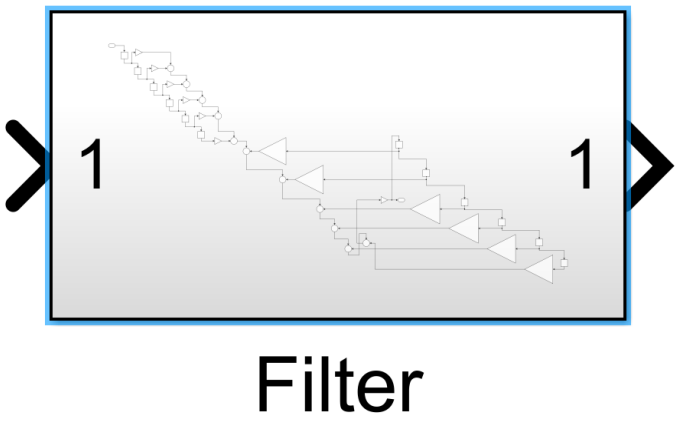

Figure 12. H(z) DF1 realization using dfilt.df1() and Simulink

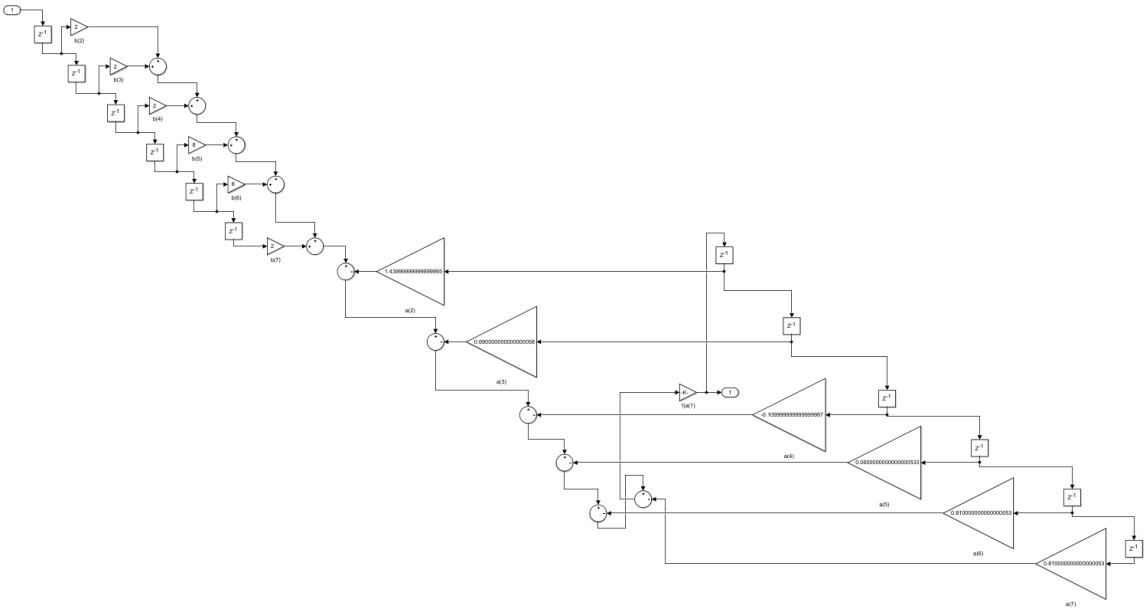

Figure 13. H(z) DF1 realization (filter)

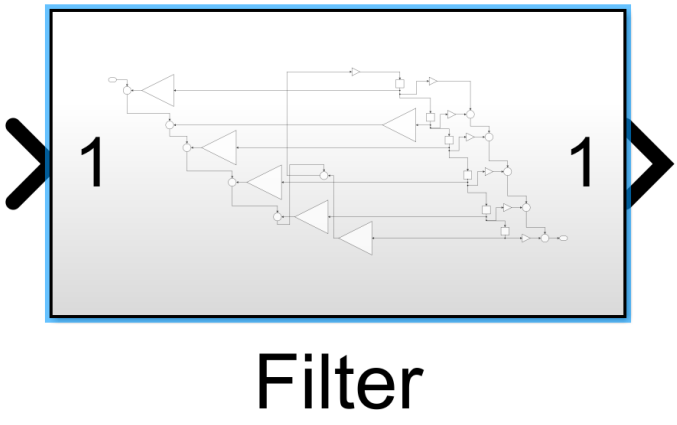

Figure 14. H(z) DF2 realization using dfilt.df2() and Simulink

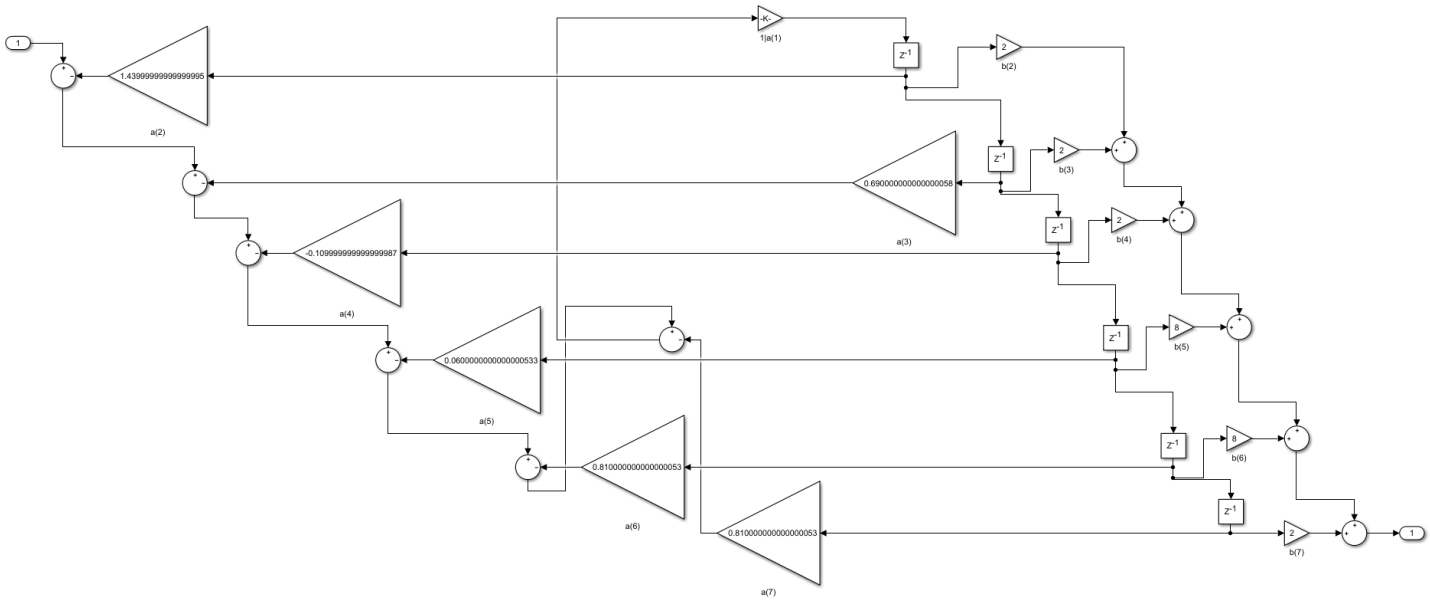

Figure 15. H(z) DF1 realization (filter)

b. Draw cascade containing sos-DF2 realization of H(z)

```
[fb2_2, fa2_2] = eqtflength([1 0 1], [1 0.8 0.64]);
b2_df2=dfilt.df2sos(fb2_2, fa2_2);
[fb2_21, fa2_21] = eqtflength([2 -1], 1 -0.75);
b2_df21=dfilt.df2sos(fb2_21, fa2_21);
[fb2_22, fa2_22] = eqtflength([1 2 1], [1 0.81]);
b2_df22=dfilt.df2sos(fb2_22, fa2_22);
b2_cscd=dfilt.cascade(b2_df2,b2_df21,b2_df22);
realizemdl(b2_cscd); % see b2_df2sos_mdl.slx for complete DF2sos realization diagram
```

Figures 16 to 19 shows the cascade realization of H(z).

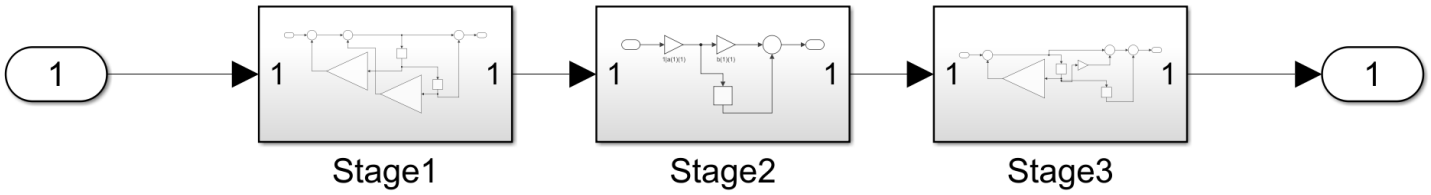

Figure 16. H(z) cascade realization using dfilt.df2sos() and Simulink

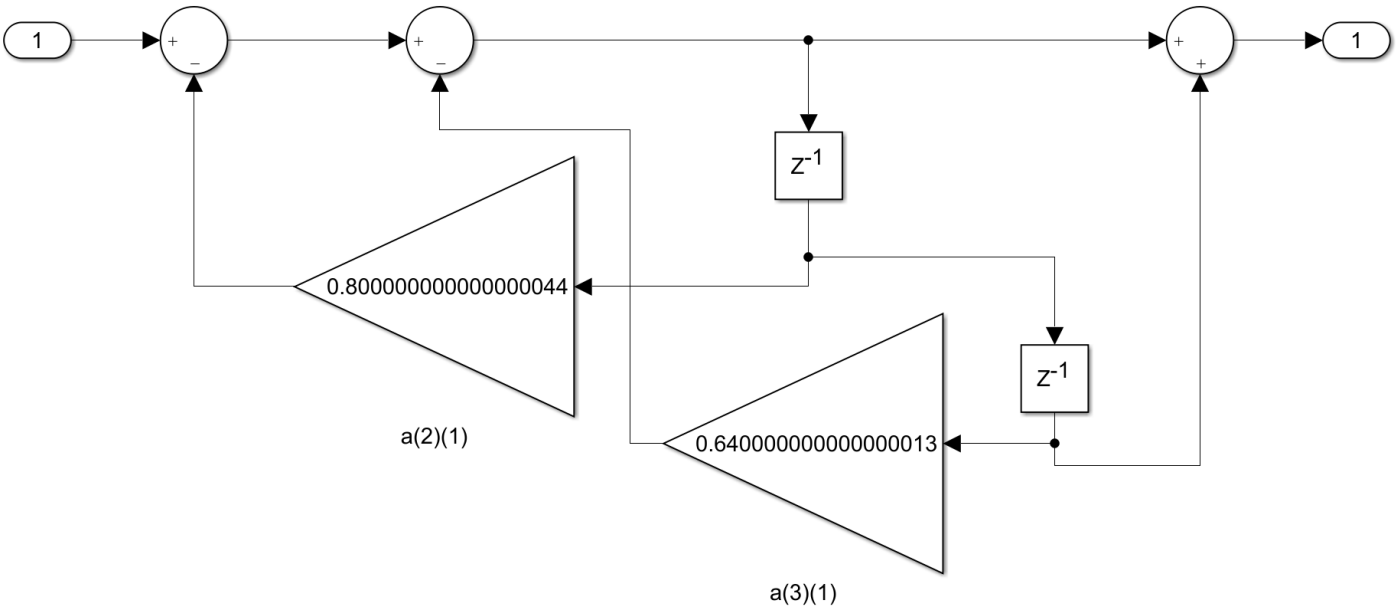

Figure 17. H(z) cascade realization Stage 1

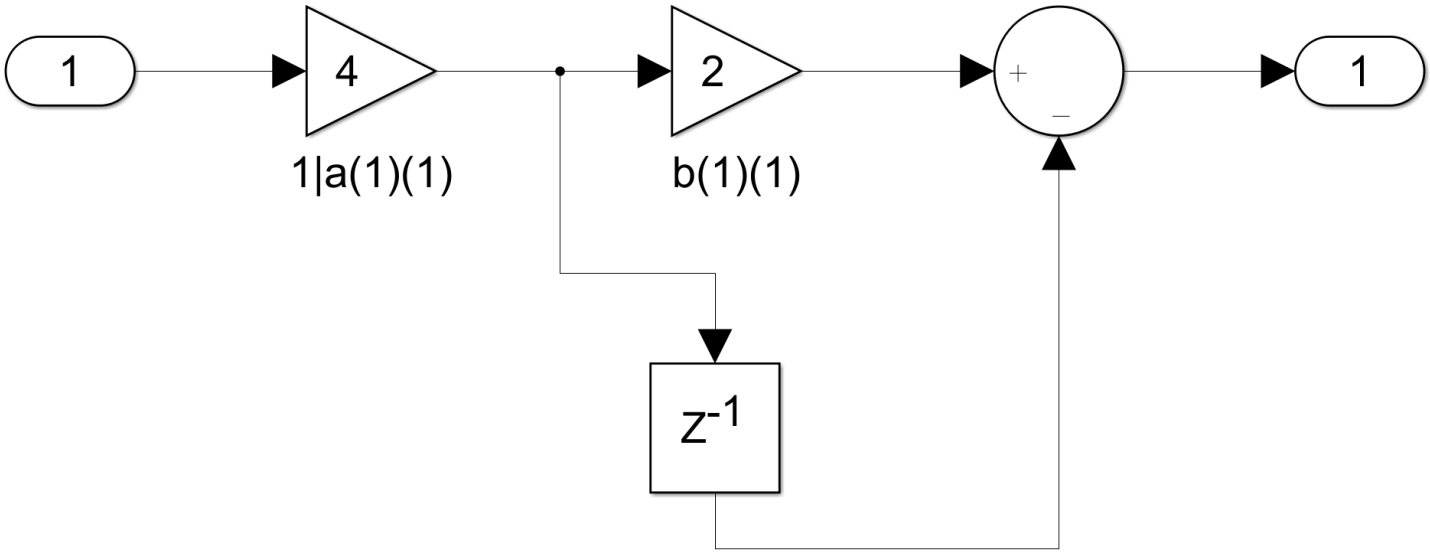

Figure 18. H(z) cascade realization Stage 2

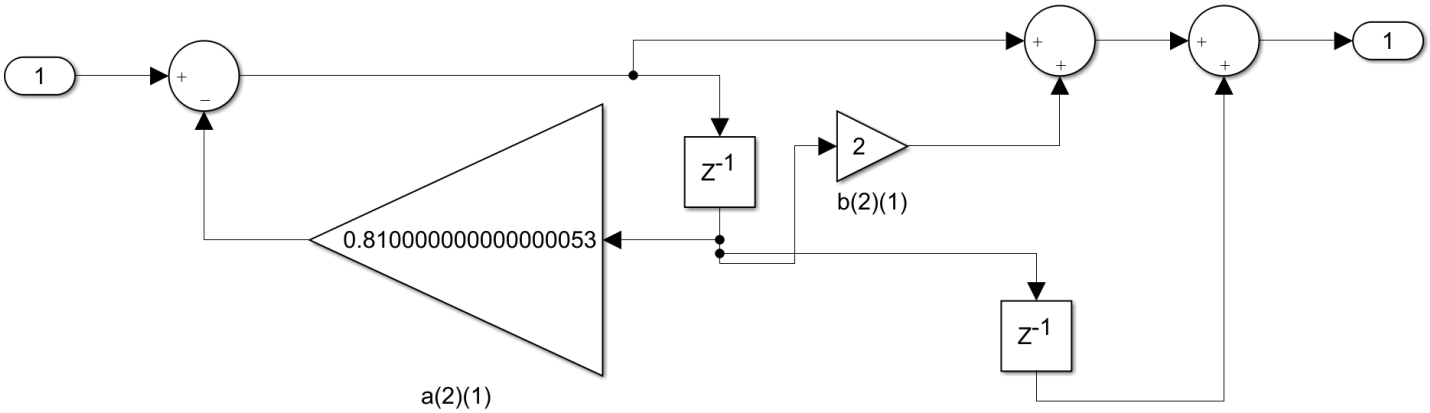

Figure 19. H(z) cascade realization Stage 3

c. Draw parallel containing sos-DF2 realization of H(z)

```
c2_cscd=dfilt.parallel(b2_df2,b2_df21,b2_df22);
realizemdl(c2_cscd); % see c2_df2sos_mdl.slx for complete DF2sos realization diagram
```

Figures 20 to 23 shows the cascade realization of H(z).

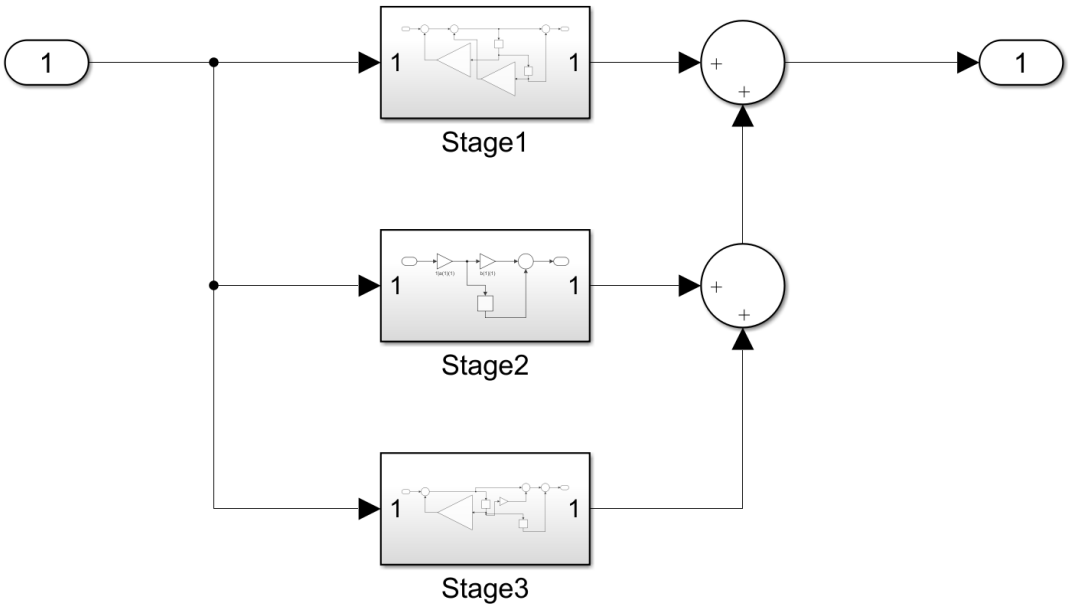

Figure 20. H(z) parallel realization using dfilt.df2sos() and Simulink

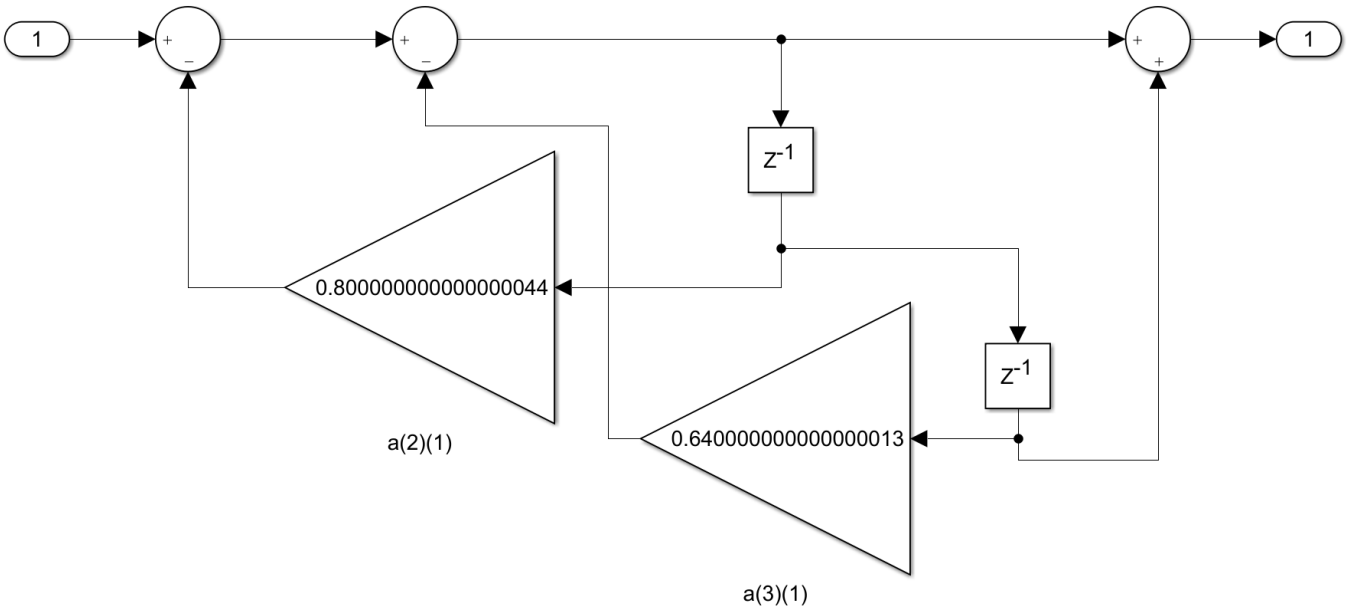

Figure 21. H(z) parallel realization Stage 1

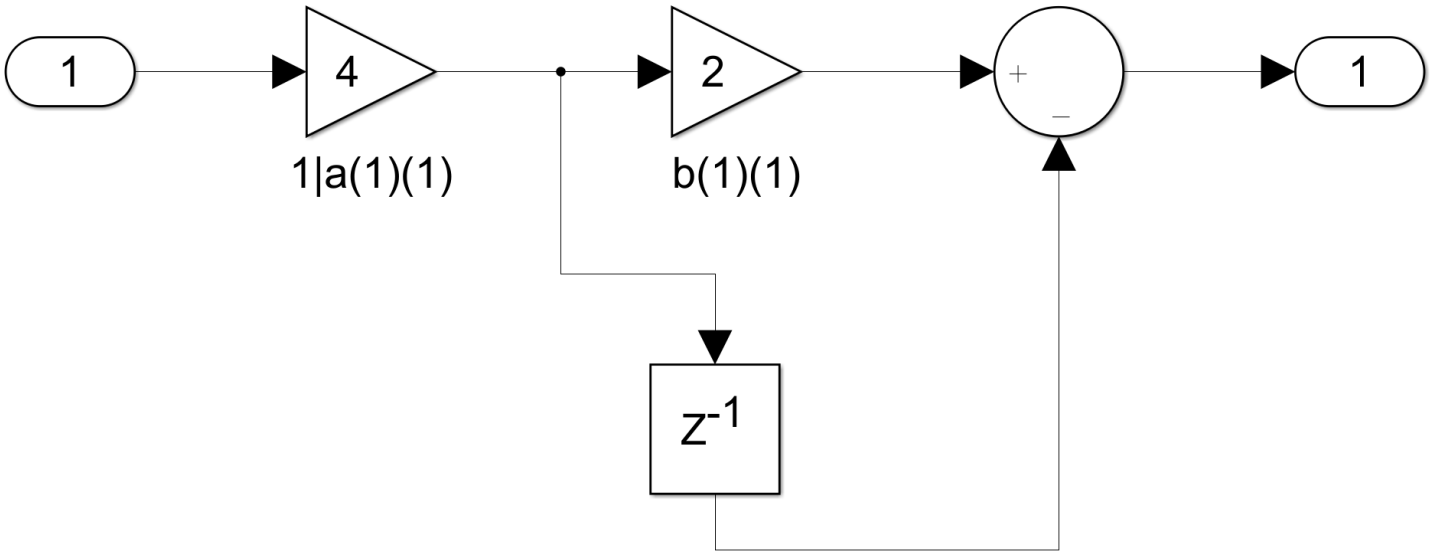

Figure 22. H(z) parallel realization Stage 2

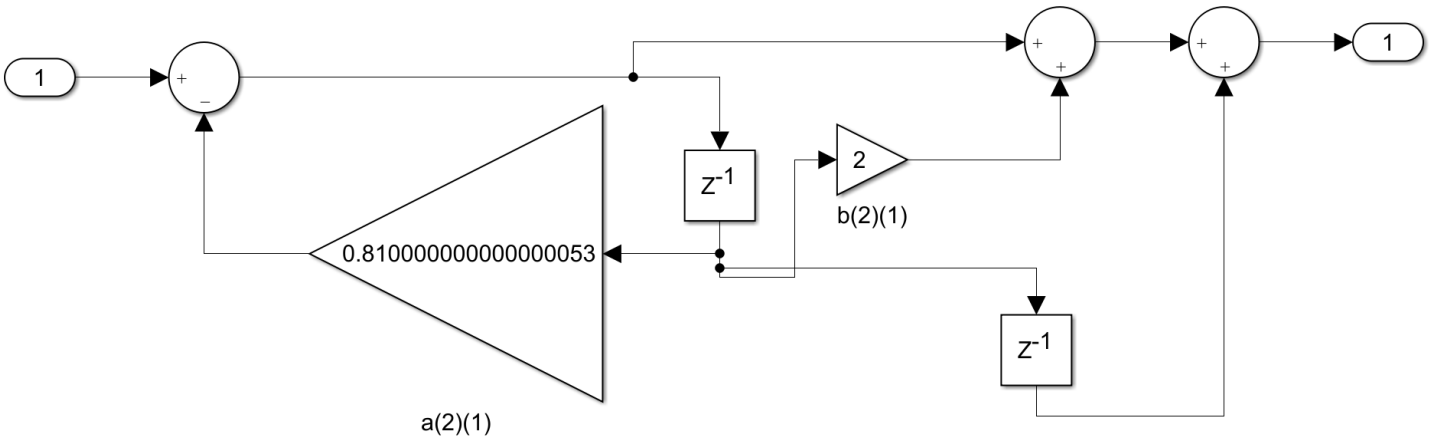

Figure 23. H(z) parallel realization Stage 3

3. Effects of coefficient quantization (for graduates)

Task: Modify P9_2 to investigate the effects of quantization on the filter coefficient on an eight-order elliptic high pass transfer function with a passband ripple of 0.1dB, min stopband attenuation of 70dB, and a normalized cutoff frequency of 0.55 rad/sec. Assign five bits to fractional part of the binary representations. Run the modified MATLAB code and compared the frequency response, pole-zero plot, and filter output with quantized and unquantized coef.

The following codes below are P9_1, P9_2, and the modified P9_2. Figures 24 and 25 are the plots from P9_1, figures 26 and 27 are the plots from P9_2, and figure 28 and 29 are the plots from the modified P9_2 relevant to the task above..

```
% Program P9_1
% Coefficient Quantization Effects on Direct Form
% Realization of an IIR Transfer Function
[b391,a391] = ellip(6,0.05,60,0.4);
[g391,w391]=Gain(b391,a391);
bq391 = a2dT(b391,5); aq391 = a2dT(a391,5);
[gq391,w391] = Gain(bq391, aq391);
figure(); plot(w391/pi,g391,'b', w391/pi,gq391,'r--');
axis([0 1 -80 1]);grid
xlabel('\omega /\pi');ylabel('Gain, dB');
legend('original', 'quantized');
figure(); zplane(b391,a391);
hold on;
pzplot(bq391,aq391);
hold off;
title('Original pole-zero locations: x, o; New pole-zero locations: +, *')
```

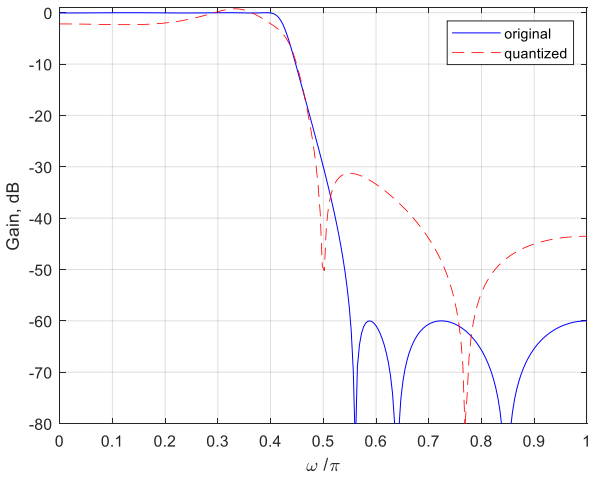

Figure 24. Frequency Response of P9_1

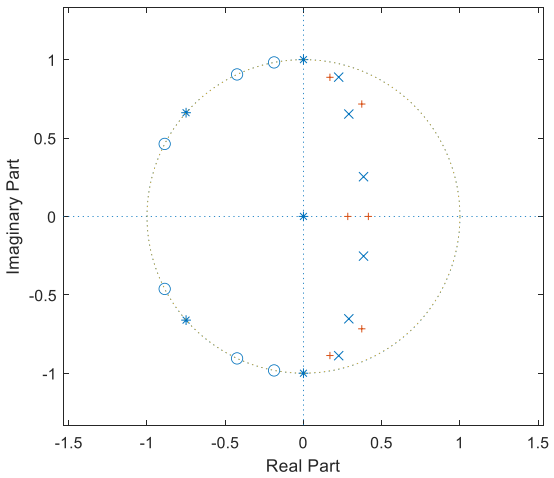

Figure 25. Zero-pole of P9_1

```
% Program P9_2
% Coefficient Quantization Effects on Cascade
% Realization of an IIR Transfer Function
[z923,p923,k923] = ellip(6,0.05,60,0.4);
[b923,a923] = zp2tf(z923,p923,k923);
[g923] = Gain(b923,a923);
sos923 = zp2sos(z923,p923,k923);
sosq923 = a2dR(sos923,6);
R1923 = sosq923(1,:);R2923 = sosq923(2,:);R3923 = sosq923(3,:);
b1923 = conv(R1923(1:3),R2923(1:3));bq923 = conv(R3923(1:3),b1923);
a1923 = conv(R1923(4:6),R2923(4:6));aq923 = conv(R3923(4:6),a1923);
[gq923,w923] = Gain(bq923, aq923);
figure(); plot(w923/pi,g923,'b-',w923/pi,gq923,'r--');
axis([0 1 -80 20]);grid
xlabel('\omega /pi');ylabel('Gain, dB');
title('original - solid line; quantized - dashed line');
figure(); zplane(b923,a923);
hold on
pzplot(bq923,aq923);
hold off
title('Original pole-zero locations: x, o; New pole-zero locations: +, *')
```

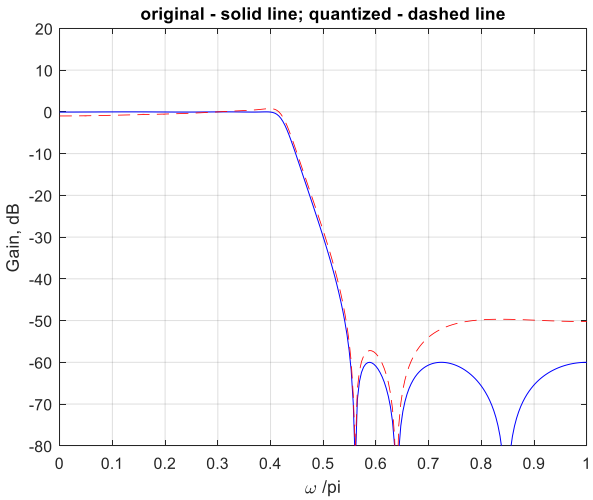

Figure 26. Frequency Response of P9_2 (blue is original; red is quantized)

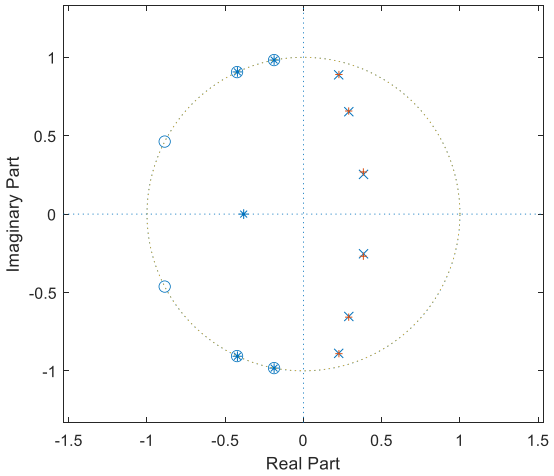

Figure 27. Zero-pole of P9_2 (blue is original; red is quantized)

The modified P9_2 satisfies the requirements given in the task above. As shown in the code (excluding the 3-line comment) below, lines 1 and 5 are altered as such the parameter values of ellip() and a2dR() are set to the following values below, respectively.

```
% Modified Program P9_2
% Coefficient Quantization Effects on Cascade
% Realization of an IIR Transfer Function
[z923_1,p923_1,k923_1] = ellip(8,0.1,70,0.55);
[b923_1,a923_1] = zp2tf(z923_1,p923_1,k923_1);
[g923_1] = Gain(b923_1,a923_1);
sos923_1 = zp2sos(z923_1,p923_1,k923_1);
sosq923_1 = a2dR(sos923_1,5);
R1923_1 = sosq923_1(1,:);
R2923_1 = sosq923_1(2,:);
R3923_1 = sosq923_1(3,:);
b1923_1 = conv(R1923_1(1:3),R2923_1(1:3));
bq923_1 = conv(R3923_1(1:3),b1923_1);
a1923_1 = conv(R1923_1(4:6),R2923_1(4:6));
aq923_1 = conv(R3923_1(4:6),a1923_1);
[gq923_1,w923_1] = Gain(bq923_1, aq923_1);
figure(); plot(w923_1/pi,g923_1,'b-',w923_1/pi,gq923_1,'r--');
axis([0 1 -80 20]); grid; xlabel('\omega /pi');ylabel('Gain, dB');
title('original - solid line; quantized - dashed line');
figure(); zplane(b923_1,a923_1); hold on; pzplot(bq923_1,aq923_1);
title('Original pole-zero locations: x, o; New pole-zero locations: +, *')
```

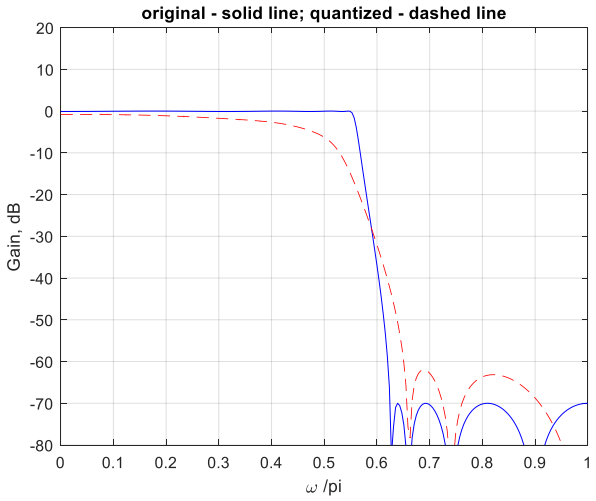

Figure 28. Frequency Response of Modified P9_2 (blue is original; red is quantized)

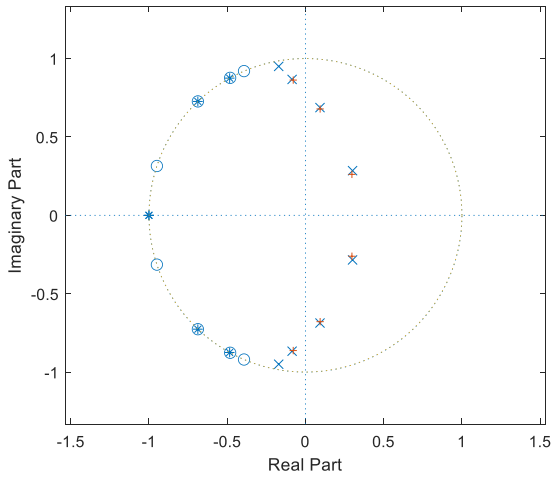

Figure 29. Zero-pole of Modified P9_2 (blue is original; red is quantized)

Generalizations:

- Figure 28 shows the frequency response of the unquantized filter (blue) as well as the quantized filter (red). By inspection, the coefficients after 5-bit quantization has adversely affected the frequency response. Frequency response of the quantized filter turned-out differently from that of the original designed filter.
- Figure 29 shows the zero-pole plot of the quantized (red) and unquantized (blue) filter. No poles are present outside the unit circle hence, both the unquantized and quantized filter is stable. Since the filter is quantized on the 8th-order of a cascade of second-order sections, quantized filter's sensitivity possibly reduced hence, becomes stable.