Lab 9 Report

IIR Filters

Alex Hirzel

Submitted to Yang Liu for EE4252

Due December 6, 2012

Michigan Technological University

Project 1: Basic Routines for Digital Filter Design

Exercise 1: IIR Filter Design

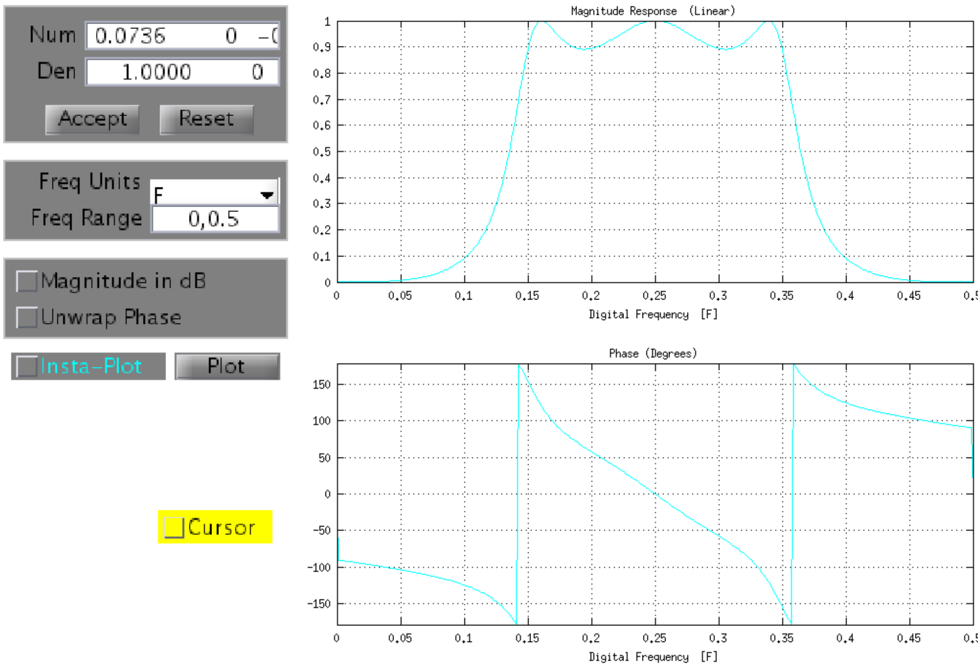


Figure 1: Hard copy of `dfdiir` showing the design of the desired filter.

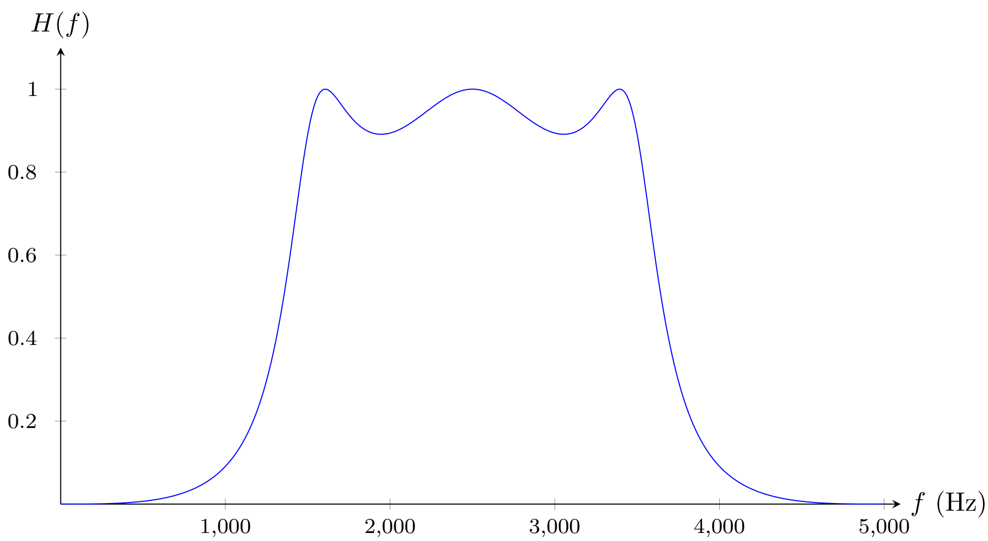


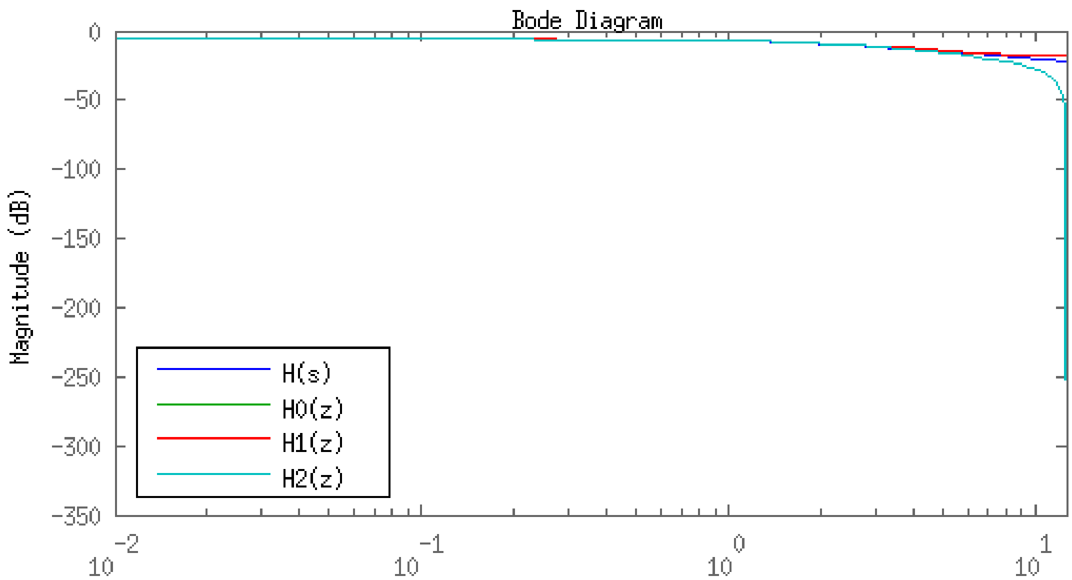
Figure 2: Plot of $\left|H(f)\right|$ versus frequency in Hz.

Exercise 2: s2z mappings

(a) The results of converting $H(s) = 1/(s+2)$ are:

$$H_0(z) = \frac{0.1967z}{z - 0.6065} \qquad H_1(z) = \frac{0.1967z}{z - 0.6065} \qquad H_2(z) = \frac{0.1 - 0.1z}{z - 0.6}$$

(b) All filters are overplotted below.



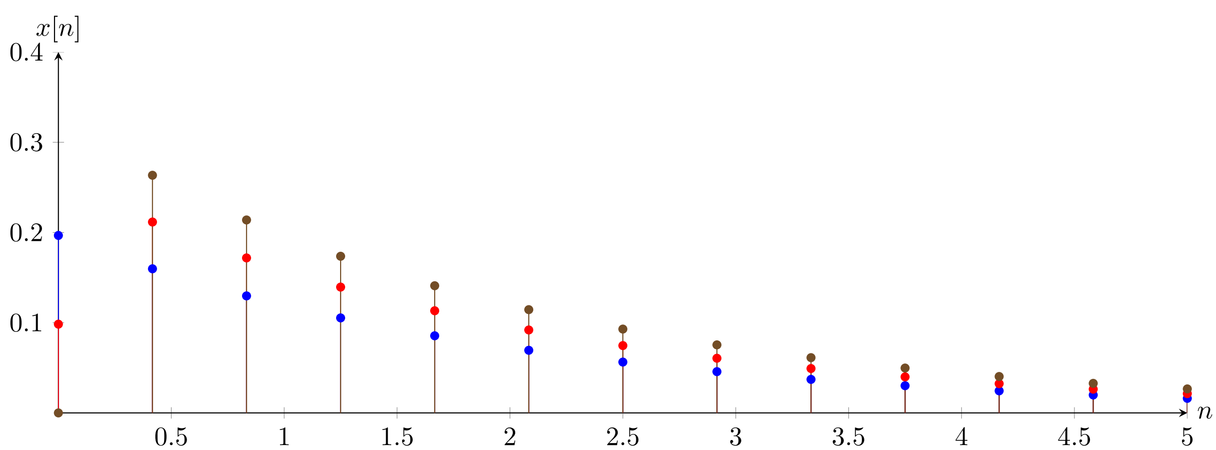
(c) It can be seen from above that $H_1(z)$ from above most closely matches $H(s)$. This is because the red and blue lines are closer to one another than the blue and teal lines.

(d) Shown below are their impulse responses $h_{1\{0,1,2\}}[n]$.

$$H_{10}(z) = \frac{0.1967z}{z - 0.6065} \qquad h_{10}[n] = 0.19673(0.6065)^n u[n]$$

$$H_{11}(z) = \frac{0.9837z + 0.9837}{z - 0.6065} \qquad h_{11}[n] = 0.26056(0.6065)^n u[n] - 0.16219\delta[n]$$

$$H_{12}(z) = \frac{0.1 - 0.1z}{z - 0.6} \qquad h_{12}[n] = 0.32438(0.6065)^n u[n] - 0.32438\delta[n]$$



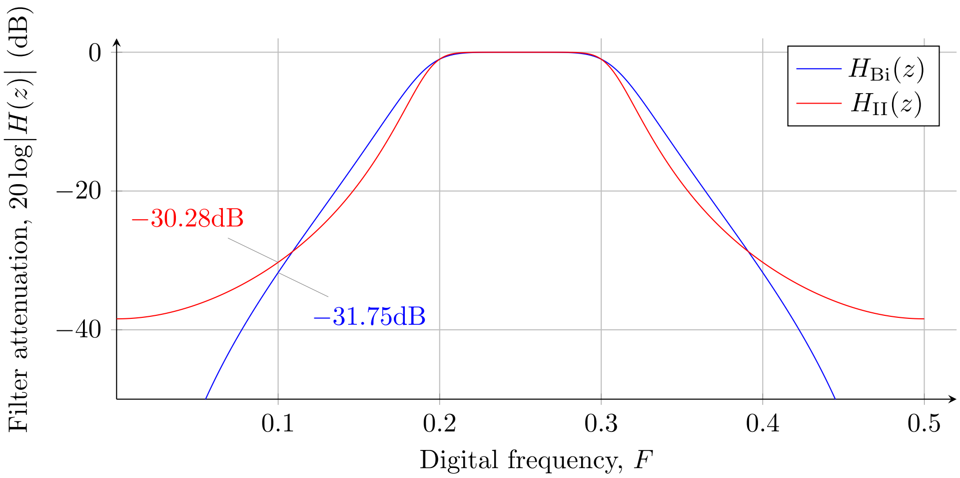
Project 2: GUI-Based Digital Filter Design

The impulse invariant and bilinear transform Butterworth designs produced by `diirgui` are:

$$H_{\mathrm{II}}(z) = \frac{0.0005238z^{10} + 0.02412z^8 + 0.09109z^6 + 0.01531z^4 - 0.006237z^2 - 0.001179}{z^{10} + 2.811z^8 + 3.432z^6 + 2.216z^4 + 0.746z^2 + 0.1038}$$

$$H_{\mathrm{Bi}}(z) = \frac{0.03047z^6 - 0.0914z^4 + 0.0914z^2 - 0.03047}{z^6 + 1.483z^4 + 0.9296z^2 + 0.2033}$$

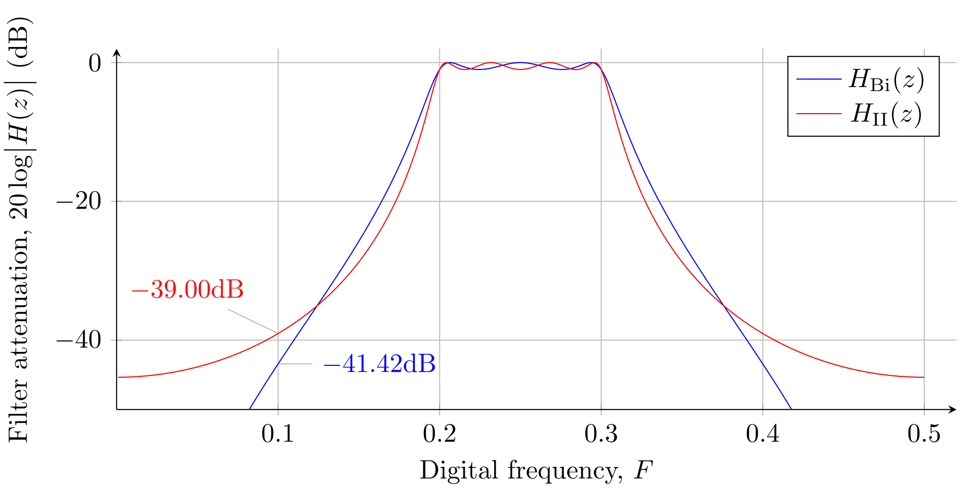
Each filter design meets specs. The two yield filters that are *not* the same order. The filter designed via the bilinear transform is more economical with order six rather than ten.



The Chebyshev filters also have different orders (eight and six, respectively, for the impulse invariant and bilinear transform designs). Both designs still meet specs. The filter designed via the bilinear transform is more economical with order six rather than eight.

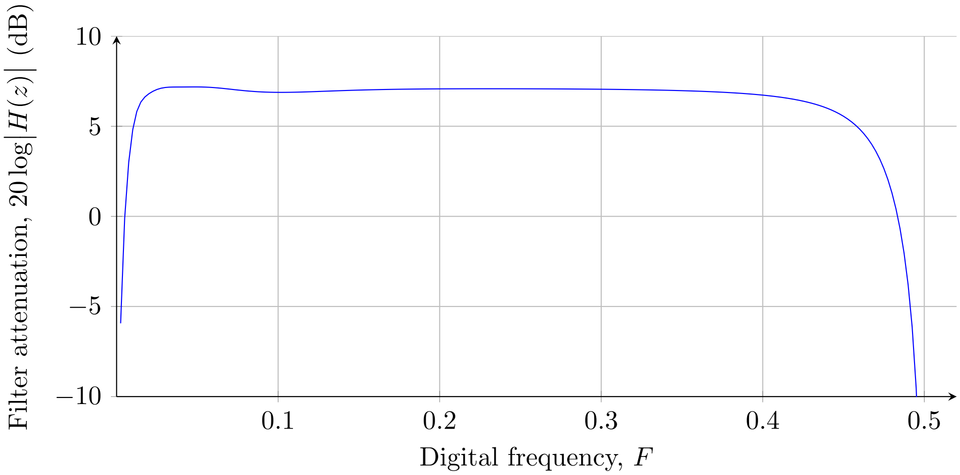
$$H_{\mathrm{II}}(z) = \frac{-z^8 - 3.067z^6 - 3.837z^4 - 2.29z^2 - 0.5474}{0.001235z^8 - 0.005121z^6 - 0.04221z^4 - 0.0116z^2 - 0.0003102}$$

$$H_{\mathrm{Bi}}(z) = \frac{z^6 + 2.138z^4 + 1.769z^2 + 0.5398}{0.01147z^6 - 0.03442z^4 + 0.03442z^2 - 0.01147}$$



Project 3: A Digital Audio Equalizer

The response varies as an equalizer would be expected to work: there are five controllable peaks corresponding to the `g` input to `audioeq`, and the flattest response was found using `audioeq(0.5, [1 1 1 0.5 2.1])`. The passband ripple is on the order of 1dB.

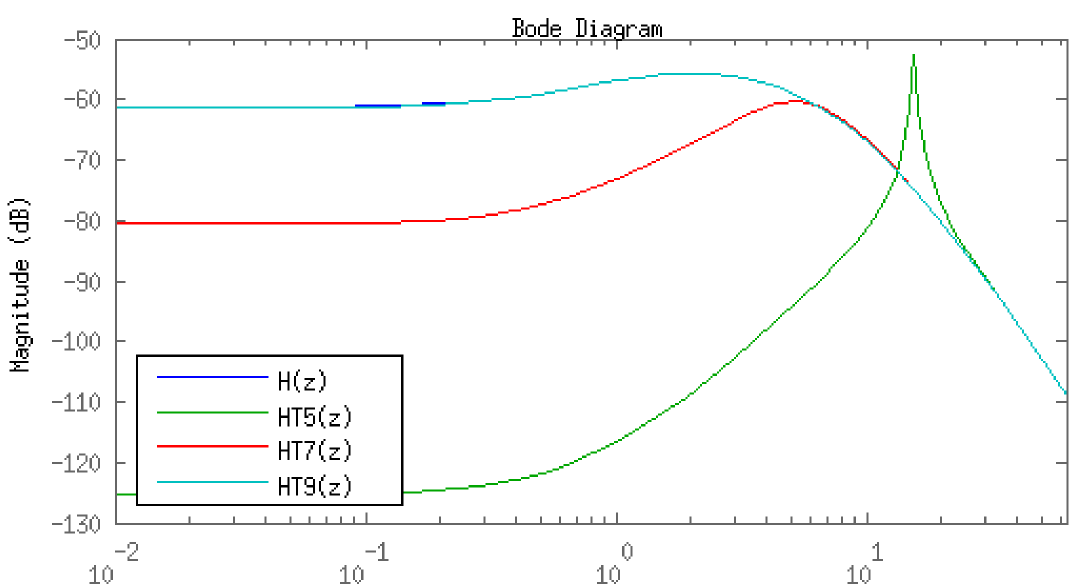


Project 4: Stability of IIR Filters

(a) The result is approximately

$$H(z) \approx \frac{10^{-7}(1.103z^5 + 1.125z^4 - 2.163z^3 - 2.207z^2 + 1.06z + 1.082)}{z^5 - 4.736z^4 + 8.969z^3 - 8.488z^2 + 4.015z - 0.7594}.$$

(c) The bode plot is shown below. The truncated version is shown in red.



(d) The numerator of both $H(z)$ and $H_T(z)$ have zeroes that are on the unit circle (magnitude 1.000002633680229). For the purposes of this analysis, then, the filter will be minimum-phase if denominator poles are inside the unit circle. In the case of $H(z)$, all poles are inside the unit circle, so $H(z)$ is stable and minimum-phase.

(e) $H_T(z)$ has the same numerator as $H(z)$, but the denominator of $H_T(z)$ has a pole that is outside the unit circle. Therefore, $H_T(z)$ is not stable nor is it minimum-phase.

(f) Truncation to 9 significant figures results in a stable filter (which is therefore minimum-phase). Expectedly, truncation to 5 significant figures results in a filter that is even less stable than $H_T(z)$ (i.e. pole radius is even larger), a non-minimum-phase result.

Appendix: MATLAB Source Code

What follows is a listing of the MATLAB source code (Listing 1) and the text output of this code (Listing 2) used to generate the figures and other information presented in this report.

Listing 1: The MATLAB script used for this report, `Lab09_ahirzel.m`.

```
addpath ../../ClassWorkspace;
delete 'generated/diary.txt'; diary 'generated/diary.txt'; diary on

evalztf = @(N, D, z) polyval(N, z) ./ polyval(D, z);
evalztfF = @(N, D, F) evalztf(N, D, exp(1j*2*pi*F));


% Project 1: Basic Routines for Digital Filter Design %%%%%%%%%%%%%%%%%%%%%

[N, D] = dfdiir('c1', 'bp', 'trap', [1 20], 10000, [1500 3500], [1000 4000]);
load p1e1; csvwrite('generated/p1e1.csv', [p1e1_dfreq .* 10000 p1e1_Mtf]);

s  = tf([0 1], [1 2]);
[N0, D0] = s2zinvar([0 1], [1 2], 4, 'i', 0); z0 = tf(N0, D0, 1/4, 'Variable', 'z')
[N1, D1] = s2zmatch([0 1], [1 2], 4, '0', 0); z1 = tf(N1, D1, 1/4, 'Variable', 'z')
[N2, D2] = s2zni([0 1], [1 2], 4, 'trap', 0); z2 = tf(N2, D2, 1/4, 'Variable', 'z')
bodemag(s, z0, z1, z2, {0.01, 2*2*pi})
legend('H(s)', 'H0(z)', 'H1(z)', 'H2(z)', 'Location', 'Southwest');
mysaveas('p1e2-magnitudes', 4, 2);

[N10, D10] = s2zmatch([0 1], [1 2], 4, '0', 0); tf(N10, D10, 1/4, 'Variable', 'z')
[N11, D11] = s2zmatch([0 1], [1 2], 4, '1', 0); tf(N11, D11, 1/4, 'Variable', 'z')
[N12, D12] = s2zmatch([0 1], [1 2], 4, '2', 0); tf(N12, D12, 1/4, 'Variable', 'z')


% Project 2: GUI-Based Digital Filter Design %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

load p2butt; load p2cheb; F = 0:0.001:0.5; S = 100;

tf(D_Butt_II, N_Butt_II, 1/S, 'Variable', 'z')
tf(D_Butt_Bi, N_Butt_Bi, 1/S, 'Variable', 'z')
tf(D_Cheb_II, N_Cheb_II, 1/S, 'Variable', 'z')
tf(D_Cheb_Bi, N_Cheb_Bi, 1/S, 'Variable', 'z')

csvwrite('generated/p2-butt.csv', [F', 20*log10(abs(evalztfF(N_Butt_Bi, D_Butt_Bi, F)))', ...
                                       20*log10(abs(evalztfF(N_Butt_II, D_Butt_II, F)))'] );
csvwrite('generated/p2-cheb.csv', [F', 20*log10(abs(evalztfF(N_Cheb_Bi, D_Cheb_Bi, F)))', ...
                                       20*log10(abs(evalztfF(N_Cheb_II, D_Cheb_II, F)))'] );


% Project 3: A Digital Audio Equalizer %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[N, D, ht, f] = audioeq(0.5, [1 1 1 0.5 2.1]); close(gcf);
csvwrite('generated/p3-flattest.csv', [f' 20*log10(abs(ht))']);


% Project 4: Stability of IIR Filters %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

N = [0 0 0 1 2 0.75]; D = [1 27.5 261.5 1039 1668 864]; S = 100;

[Nz, Dz] = s2zni(N, D, S, 'bili');
Dzt9 = numdig(Dz, 9); Dzt7 = numdig(Dz, 7); Dzt5 = numdig(Dz, 5);
bodemag(tf(Nz, Dz, 1/S, 'Variable', 'z'), ...
        tf(Nz, Dzt5, 1/S, 'Variable', 'z'), ...
        tf(Nz, Dzt7, 1/S, 'Variable', 'z'), ...
        tf(Nz, Dzt9, 1/S, 'Variable', 'z'), ...
        {0.01, 0.1*S*2*pi});
legend('H(z)', 'HT5(z)', 'HT7(z)', 'HT9(z)', 'Location', 'Southwest');
mysaveas('p4', 4, 2);

disp('Zeroes of H(z):'); roots(Nz)
disp('Poles of H(z):'); roots(Dz)
disp('Poles of H_T(z) truncated to 9 significant figures:'); roots(Dzt5)
disp('Poles of H_T(z) truncated to 9 significant figures:'); roots(Dzt7)
disp('Poles of H_T(z) truncated to 9 significant figures:'); roots(Dzt9)

diary off
```

Listing 2: The output of listing 1, `diary.txt`.

```
z0 =

   0.1967 z
  ----------
  z - 0.6065

Sample time: 0.25 seconds
Discrete-time transfer function.


z1 =

   0.1967 z
  ----------
  z - 0.6065

Sample time: 0.25 seconds
Discrete-time transfer function.


z2 =

  0.1 z + 0.1
  -----------
    z - 0.6

Sample time: 0.25 seconds
Discrete-time transfer function.


ans =

   0.1967 z
  ----------
  z - 0.6065

Sample time: 0.25 seconds
Discrete-time transfer function.


ans =

  0.09837 z + 0.09837
  -------------------
      z - 0.6065

Sample time: 0.25 seconds
Discrete-time transfer function.


ans =

    0.1967
  ----------
  z - 0.6065

Sample time: 0.25 seconds
Discrete-time transfer function.


ans =

            z^10 + 2.811 z^8 + 3.432 z^6 + 2.216 z^4 + 0.746 z^2 + 0.1038
  --------------------------------------------------------------------------------
  0.0005238 z^10 + 0.02412 z^8 + 0.09109 z^6 + 0.01531 z^4 - 0.006237 z^2 - 0.001179

Sample time: 0.01 seconds
Discrete-time transfer function.


ans =

       z^6 + 1.483 z^4 + 0.9296 z^2 + 0.2033
  -----------------------------------------------
  0.03047 z^6 - 0.0914 z^4 + 0.0914 z^2 - 0.03047

Sample time: 0.01 seconds
Discrete-time transfer function.


ans =
```

```
           -z^8 - 3.067 z^6 - 3.837 z^4 - 2.29 z^2 - 0.5474
  -----------------------------------------------------------------
  0.001235 z^8 - 0.005121 z^6 - 0.04221 z^4 - 0.0116 z^2 - 0.0003102

Sample time: 0.01 seconds
Discrete-time transfer function.


ans =

        z^6 + 2.138 z^4 + 1.769 z^2 + 0.5398
  ------------------------------------------------
  0.01147 z^6 - 0.03442 z^4 + 0.03442 z^2 - 0.01147

Sample time: 0.01 seconds
Discrete-time transfer function.

Zeroes of H(z):

ans =

   0.9950
   0.9851
  -1.0000 + 0.0000i
  -1.0000 - 0.0000i
  -1.0000

Poles of H(z):

ans =

    0.9900
    0.9802
    0.9560
    0.9231
    0.8868

Poles of H_T(z) truncated to 9 significant figures:

ans =

   1.1227
   0.9906 + 0.1546i
   0.9906 - 0.1546i
   0.8161 + 0.0831i
   0.8161 - 0.0831i

Poles of H_T(z) truncated to 9 significant figures:

ans =

   1.0177
   0.9701 + 0.0468i
   0.9701 - 0.0468i
   0.8892 + 0.0238i
   0.8892 - 0.0238i

Poles of H_T(z) truncated to 9 significant figures:

ans =

    0.9895
    0.9811
    0.9557
    0.9231
    0.8868
```