EE323 · Digital Signal Processing

Lab 6: z-transform and LTI system in Transformed Domain

任振裕 / 11812214

December 2, 2020

Contents

6.1 Introduction

There are mainly two parts of this lab:

- **z-transform:**
  The z-transform is useful for the manipulation of discrete data sequences and has acquired a new significance in the formulation and analysis of discrete-time systems. And in this part of the lab, we will both discuss z-transform and inverse z-transform deeply through many aspects: ROC regions, poles, zeros etc. We will also discuss a special case of z-transform that $z = e^{j\omega}$ to calculate the frequency response of a LTI system. Through this part, we could have a deeper understanding towards z-transform and LTI system. And this will help us a lot for the further filter design.
- **LTI system in Transformed Domain:**
  In this part, we will mainly consider two digital filters: Linear phase FIR filters and IIR filters. Linear Phase FIR filters are an important class of FIR filters. We will use four basic types of it to have a further exploration for it. In the part of IIR filter, we will involve a simple first order filter. All these could help us have a first insight into the digital filter designs.

Warning: In order to simplify the bandwidth analysis, all magnitude responses in dB use the formula of squared magnitude:

$$\text{magnitude response in dB} = 20\log_{10}\text{magnitude response}$$

6.2 z-transform

6.2.2 Poles and zeros of z-transform

- Three functions considered in this section:

$$H_1(z) = -1 + 2z^{-1} - 3z^{-2} + 6z^{-3} - 3z^{-4} + 2z^{-5} - z^{-6}$$

$$G_1(z) = \frac{3z^4 - 2.4z^3 + 15.36z^2 + 3.84z + 9}{5z^4 - 8.5z^3 + 17.6z^2 + 4.7z - 6}$$

$$G_2(z) = \frac{2z^4 + 0.2z^3 + 6.4z^2 + 4.6z + 2.4}{5z^4 + z^3 + 6.6z^2 + 4.2z + 24}$$

- MATLAB codes for this section:

q6_2_2.m

```
clear;
P1 = [-1 2 -3 6 -3 2 -1];
D1 = [1 zeros(1,6)];
P2 = [3 -2.4 15.36 3.84 9];
D2 = [5 -8.5 17.6 4.7 -6];
P3 = [2 0.2 6.4 4.6 2.4];
D3 = [5 1 6.6 4.2 24];
zeros1 = roots(P1);poles1 = roots(D1);display(zeros1);display(poles1);
zeros2 = roots(P2);poles2 = roots(D2);display(zeros2);display(poles2);
zeros3 = roots(P3);poles3 = roots(D3);display(zeros3);display(poles3);
subplot(131),zplane(zeros1,poles1),title('H_1(z)');
subplot(132),zplane(zeros2,poles2),title('G_1(z)');
subplot(133),zplane(zeros3,poles3),title('G_2(z)');
```

- Zero-Pole plots:

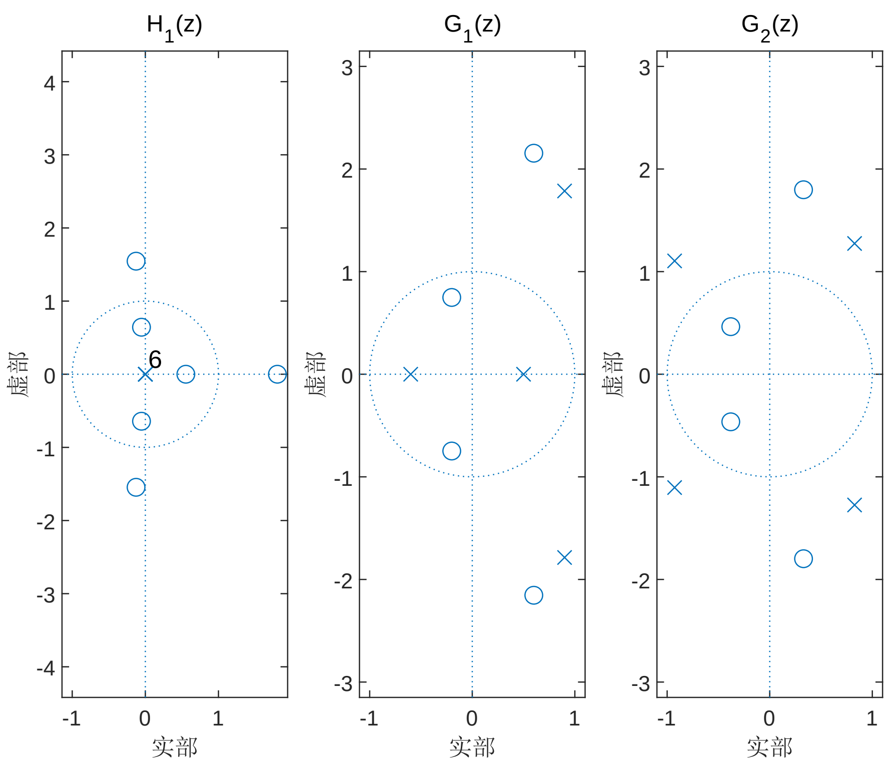

Figure 1: Zero-Pole plots

> **Info:** To decide the possible ROC region, we used the rules below:
>
> - For right-sided sequences: ROC extends outward from the outermost pole to infinity.
> - For left-sided: ROC inwards from the innermost pole to the original point.
> - For two-sided: ROC either is a ring or do not exist.

- For $H_1(z)$:
  - Poles: 0, 0, 0, 0, 0, 0 (Or a 6$^{\text{th}}$ pole at zero).
  - Zeros: 1.8054 + 0.0000i, -0.1269 + 1.5457i, -0.1269 - 1.5457i, -0.0528 + 0.6426i, -0.0528 - 0.6426i, 0.5539 + 0.0000i.
  - Possible ROC:
    * left-sided: Impossible since poles are all zero.
    * right-sided: ROC = $\{z : |z| > 0\} = \{-\infty, 0\} \cup \{0, +\infty\}$.
    * two-sided: Impossible since poles are all zero.
- For $G_1(z)$:
  - Poles: 0.9000 + 1.7861i, 0.9000 - 1.7861i, -0.6000 + 0.0000i, 0.5000 + 0.0000i.
  - Zeros: 0.6000 + 2.1541i, 0.6000 - 2.1541i, -0.2000 + 0.7483i, -0.2000 - 0.7483i.
  - Possible ROC:
    * To get the possible ROC, first derive the outermost and innermost poles:

Command Line

```
» max(abs(poles2))
ans =
2.0000
» min(abs(poles2))
ans =
0.5000
```

    * left-sided: ROC = $\{z : |z| < 0.5\}$.
    * right-sided: ROC = $\{z : |z| > 2\}$.
    * two-sided: ROC = $\{z : 0.5 < |z| < 2\}$.
- For $G_2(z)$:
  - Poles: 0.8272 + 1.2744i, 0.8272 - 1.2744i, -0.9272 + 1.1045i, -0.9272 - 1.1045i.
  - Zeros: 0.3296 + 1.7980i, 0.3296 - 1.7980i, -0.3796 + 0.4637i, -0.3796 - 0.4637i.
  - Possible ROC:
    * To get the possible ROC, first derive the outermost and innermost poles:

Command Line

» max(abs(poles3))

ans =

1.5193

» min(abs(poles3))

ans =

1.4421

* left-sided: ROC = $\{z : |z| < 1.4421\}$.
* right-sided: ROC = $\{z : |z| > 1.5193\}$
* two-sided: ROC = $\{z : 1.4421 < |z| < 1.5193\}$.

6.2.3 z-transform and frequency response

Info: For this section, we used the property that $H(e^{j\omega}) = H(z)|_{|z|=1}$ to plot magnitude and phase response.

- Two transfer functions for this section:
  - $H(z) = z^{-4} + 2z^{-3} + 2z^{-1} + 1$.
  - $H(z) = \frac{z^2-1}{z^2-1 \cdot 2z+0.95}$.
- MATLAB function for this section:

FreRes.m

```
function [mag,phase] = FreRes(num,den)
% num, and den are vectors specifying the coefficients of numerator and
    denominator.
% Calculate the function expression and then use fplot to plot it.
nP = 0:-1:1-length(num);% The index of z for numerator: z^0, z^(-1),z
    ^(-2),...
nD = 0:-1:1-length(den);% The index of z for denominator: z^0, z^(-1),z
    ^(-2),...
syms w;
P = sum(num.*exp(j*w).^nP)% numerator
D = sum(den.*exp(j*w).^nD)% denonminator
H = P/D % Obtain the function expression of the transfer function.
mag = abs(H); phase = angle(H);
% sgtitle({'Transfer function: ',char(H)},'FontSize',10)
subplot(131),fplot(w,(abs(H)),[-pi,pi]);
xlabel('w(rad)'),ylabel('magnitude response');
title('magnitude response');
subplot(132),fplot(w,20*log10(abs(H)),[-pi,pi]);% magnitude in dB
xlabel('w(rad)'),ylabel('magnitude response(dB)');
title('magnitude response(dB)');hold on;
subplot(133),fplot(w,angle(H),[-pi,pi]);% phase
xlabel('w(rad)'),ylabel('phase response');
title('phase response');hold on;
end
```

- For $H(z) = z^{-4} + 2z^{-3} + 2z^{-1} + 1$:

Command Line

```
» FreRes([1 2 2 1],1)
P =
2*exp(-w*1i) + 2*exp(-w*2i) + exp(-w*3i) + 1
D =
1
H =
2*exp(-w*1i) + 2*exp(-w*2i) + exp(-w*3i) + 1
```

Magnitude and phase response:

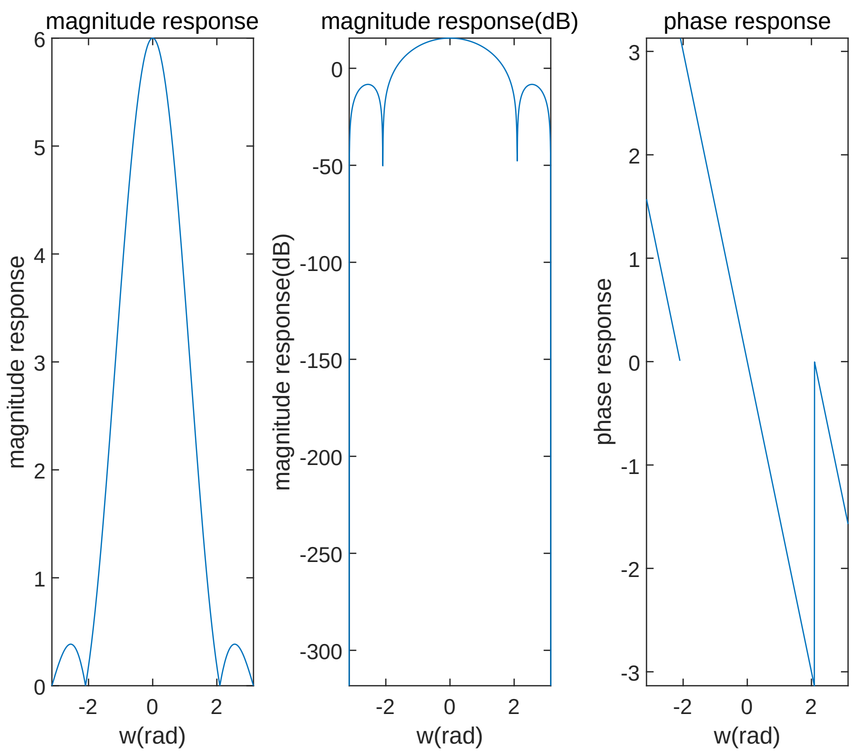

Figure 2: Magnitude and phase response for $H(z) = z^{-4} + 2z^{-3} + 2z^{-1} + 1$

- For $H(z) = \frac{z^2-1}{z^2-1\cdot 2z+0.95} = \frac{1-z^{-2}}{1-1.2z^{-1}+0.95z^{-2}}$:

Command Line

```
» FreRes([1 0 -1],[1 -1.2 0.95])
P =
1 - exp(-w*2i)
D =
1 + (19*exp(-w*2i))/20 - (6*exp(-w*1i))/5
H =
-(exp(-w*2i) - 1)/(1 + (19*exp(-w*2i))/20 - (6*exp(-w*1i))/5)
```

Magnitude and phase response:

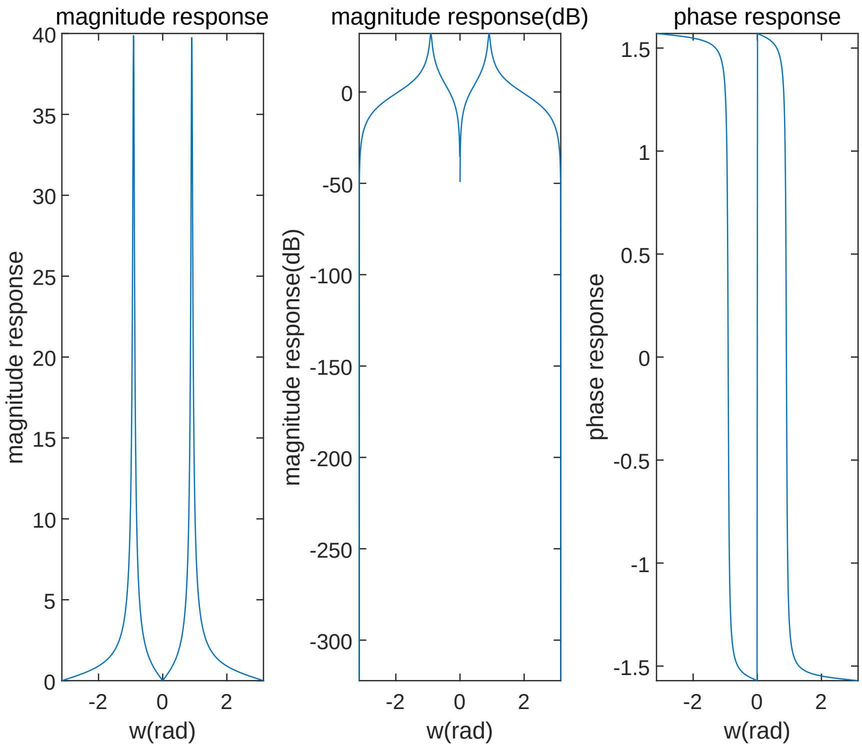

Figure 3: Magnitude and phase response for $H(z) = \frac{z^2-1}{z^2-1.2z+0.95}$

6.2.4 Inverse z-transform

- Two functions considered in this section:
  - $X(z) = \frac{3-7.8z^{-1}}{(1-0.7z^{-1})(1+1.6z^{-1})}$.
  - $Y(z) = \frac{3z^2+1.8z+1.28}{(z-0.5)(z+0.4)}$.
- The MATLAB function I designed to calculate the partial fraction form:

PartialFrac.m

```
function [rho,lambda,k] = PartialFrac(b,a)
% input b=[bm ... b1 b0] refers to coefficients of [z^(-m) ... z^(-1) z
    ^0].
```

```
% input a=[an ... a1 a0] refers to coefficients of [z^(-m) ... z^(-1) z
    ^0].
% b for numerator and a for denominator;
% Partial fraction format: K(z) + sum(rho/(1-lambda*z^(-1)));
[r, p, k] = residue(b,a);
rho = -r./p;
lambda = 1./p;
end
```

Info:

- The partial fraction expansion of z-transform could be expressed as:

$$X(z) = K(z) + \frac{P_1(z)}{D(z)} = K(z) + \sum_{l=1}^{N}\left(\frac{\rho_l}{1-\lambda_l z^{-1}}\right)$$

- MATLAB function **residue.m** gives: [r,p,k] = residue(b,a);

$$\frac{b(s)}{a(s)} = \frac{b_m s^m + b_{m-1}s^{m-1} + \ldots + b_1 s + b_0}{a_n s^n + a_{n-1}s^{n-1} + \ldots + a_1 s + a_0} = \frac{r_n}{s-p_n} + \ldots + \frac{r_2}{s-p_2} + \frac{r_1}{s-p_1} + k(s)$$

- To simplify the computation, I used below substitutions in my code:

$$s = z^{-1} \text{ (Use } z^{-1} \text{ to be the variable of the polynomials);}$$
$$\frac{r_l}{s-p_l} \rightarrow \left(\frac{\rho_l}{1-\lambda_l z^{-1}}\right)$$

- To calculate the inverse z-transform in this section, we used below commonly used z-transformed pairs:
  - $-\alpha^n\mu[-n-1] \leftrightarrow \frac{1}{1-\alpha z^{-1}} \quad |z| < |\alpha|$.
  - $\delta[n-m] \leftrightarrow z^{-m} \quad$ All z, except 0(if $m > 0$) or ∞(if $m < 0$).
  - $\alpha^n\mu[n] \leftrightarrow \frac{1}{1-\alpha z^{-1}} \quad |z| > |\alpha|$.

- For $X(z) = \frac{3-7.8z^{-1}}{(1-0.7z^{-1})(1+1.6z^{-1})}$:
  - $X(z) = \frac{3-7.8z^{-1}}{(1-0.7z^{-1})(1+1.6z^{-1})} = \frac{-7.8z^{-1}+3}{-1.12z^{-2}+0.9z^{-1}+1}$.
  - Type below codes in command window of MATLAB:

Command Line

```
» [rho,lambda,k] = PartialFrac([-7.8 3],[-1.12 0.9 1])
rho =
-2.4783
5.4783
lambda =
0.7000
-1.6000
k =
[]
```

- The partial expansion of $X(z)$'s z-transform: $X(z) = \frac{-2.4783}{1-0.7z^{-1}} + \frac{5.4783}{1+1.6z^{-1}}$.
  * When ROC $= \{z : |z| > 1.6\}$, then $x[n]$ is a right-sided sequence.
    $\therefore x[n] = -2.4783(0.7)^n u[n] + 5.4783(-1.6)^n u[n]$.
  * When ROC $= \{z : |z| < 0.7\}$, then $x[n]$ is a left-sided sequence.
    $\therefore x[n] = 2.4783(0.7)^n u[-n-1] - 5.4783(-1.6)^n u[-n-1]$.
  * When ROC $= \{z : 0.7 < |z| < 1.6\}$, then $x[n]$ is a two-sided sequence.
    $\therefore x[n] = -2.4783(0.7)^n u[n] - 5.4783(-1.6)^n u[-n-1]$.

- For $Y(z) = \frac{3z^2+1.8z+1.28}{(z-0.5)(z+0.4)}$:
  - $Y(z) = \frac{3z^2+1.8z+1.28}{(z-0.5)(z+0.4)} = \frac{3z^2+1.8z+1.28}{z^2-0.1z-0.2} = \frac{1.28z^{-2}+1.8z^{-1}+3}{-0.2z^{-2}-0.1z^{-1}+1}$.
  - Type below codes in command window of MATLAB:

Command Line

```
» [rho,lambda,k] = PartialFrac([1.28 1.8 3],[-0.2 -0.1 1])
rho =
2.8889
6.5111
lambda =
-0.4000
0.5000
k =
-6.4000
```

  - The partial expansion of $Y(z)$'s z-transform: $Y(z) = -6.4 + \frac{2.8889}{1+0.4z^{-1}} + \frac{6.5111}{1-0.5z^{-1}}$.
    * When ROC $= \{z : |z| > 0.5\}$, then $y[n]$ is a right-sided sequence.
      $\therefore y[n] = -6.4\delta[n] + 2.8889(-0.4)^n u[n] + 6.5111(0.5)^n u[n]$.
    * When ROC $= \{z : |z| < 0.4\}$, then $y[n]$ is a left-sided sequence.
      $\therefore y[n] = 6.4\delta[n] - 2.8889(-0.4)^n u[-n-1] - 6.5111(0.5)^n u[-n-1]$.
    * When ROC $= \{z : 0.4 < |z| < 0.5\}$, then $y[n]$ is a two-sided sequence.
      $\therefore$ ROC $= 6.4\delta[n] + 2.8889(-0.4)^n u[n] - 6.5111(0.5)^n u[-n-1]$.

6.2.5 Stability Conditions

- Two functions used in this section:
  - $H_1(z) = \frac{1}{1-1.845z^{-1}+0.850586z^{-2}}$.
  - $H_2(z) = \frac{1}{1-1.85z^{-1}+0.85z^{-2}}$(The system is implemented by keeping 2 digits after the decimal points of the coefficients.).
- MATLAB codes for this section:

q6_2_5.m

```
clear;
poles1 = [0.943 0.902]; zeros1 = [0 0];
b1 = [1]; a1 = [1 -1.845 0.850586];
[h1,t1] = impz(b1,a1);
poles2 = [1 0.85]; zeros2 = [0 0];
b2 = [1]; a2 = [1 -1.85 0.85];
```

```
[h2,t2] = impz(b2,a2);
figure(1); %H1
sgtitle('Zero-Pole plot and Impulse Response of H_1(z)')
subplot(121),zplane(zeros1',poles1'),title('Zero-Pole plot');
subplot(122),stem(t1,h1),xlabel('t'),ylabel('h'),title('Impulse Response'
    );
figure(2); %H2
sgtitle('Zero-Pole plot and Impulse Response of H_2(z)')
subplot(121),zplane(zeros2',poles2'),title('Zero-Pole plot');
subplot(122),stem(t2,h2),xlabel('t'),ylabel('h'),title('Impulse Response'
    );
```

- For $H_1(z) = \frac{1}{1-1.845z^{-1}+0.850586z^{-2}} = \frac{z^2}{z^2-1.845z^1+0.850586}$:
  - Poles of $H_1(z)$: $z_p = \frac{1.845\pm\sqrt{1.845^2-4\times0.850586}}{2} = 0.943,$ or 0.902.
  - Roots of $H_1(z)$: 0,0 (Or a 2^{th} root at zero).
  - Zero-pole plot and impulse response of $H_1(z)$:

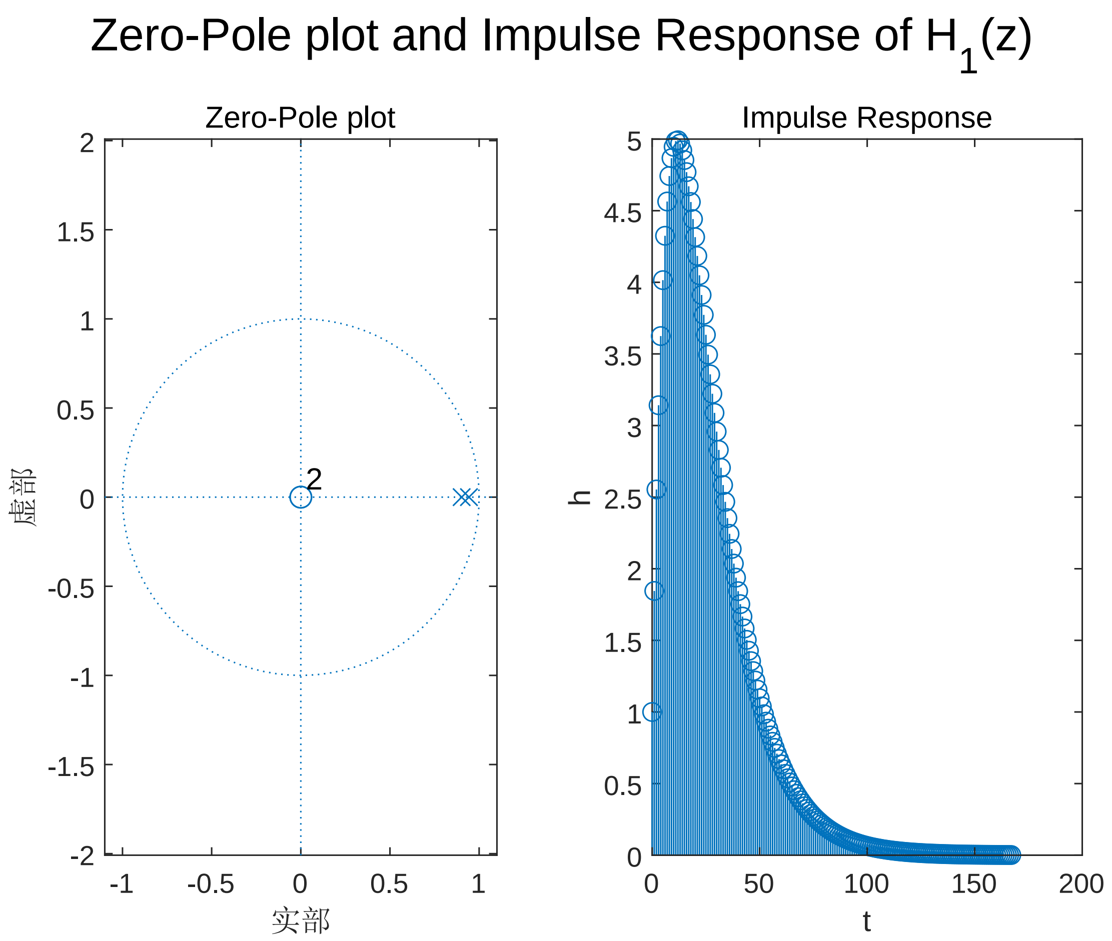

Figure 4: Zero-pole plot and impulse response of $H_1(z)$

- For $H_2(z) = \frac{1}{1-1.85z^{-1}+0.85z^{-2}} = \frac{z^2}{z^2-1.85z+0.85}$:
  - Poles of $H_2(z)$: $z_p = \frac{1.85\pm\sqrt{1.85^2-4\times0.85}}{2} = 1,$ or 0.85.

- Roots of $H_2(z)$: 0,0 (Or a 2^{th} root at zero).
- Zero-pole plot and impulse response of $H_2(z)$:

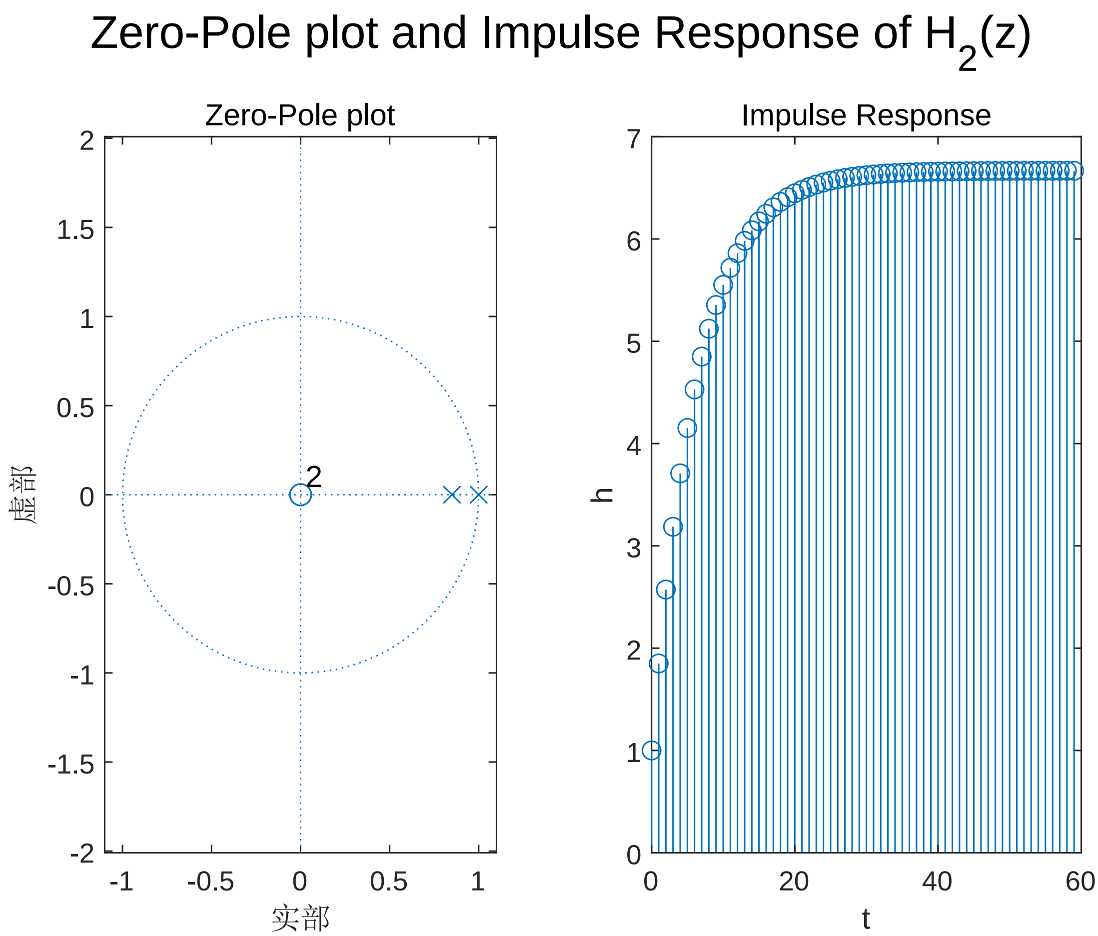

Figure 5: Zero-pole plot and impulse response of $H_2(z)$

- Analysis:
  - Statistically speaking, magnitudes of poles of $H_1(z)$ are all lower than one while those of $H_2(z)$ contain one. This means that $H_1(z)$ is stable while $H_2(z)$ is not stable.
  - Graphically speaking, according to Figures 3 and 4, all the poles of $H_1(z)$ are within the unit circle while one of $H_2(z)$'s poles is on the unit circle, which means that the ROC region of $H_1(z)$ contains the unit circle while the ROC region of $H_2(z)$ does not, therefore $H_1(z)$ is stable but $H_2(z)$ is not.
  - All these changes are caused by ping 2 digits after the decimal pointsof the coefficients. This means that an originally stable IIR filter characterized by infinite precision coefficients and with all poles inside the unit circle may become unstable after implementation due to the unavoidable quantization of all coefficients.

6.3 Linear Phase FIR Filters

6.3.1 Four Types of Linear Phase FIR Filters

A. INLAB report I for section 6.3.1

- Type I Linear Phase FIR Filter $h[n]$ considered in this section:

$$\{h[n]\} = \{-0.0035, -0.0039, 0.0072, 0.0201, -0.0000, -0.0517, -0.0506, 0.0855, 0.2965, 0.4008 \\ 0.2965, 0.0855, -0.0506, -0.0517, -0.0000, 0.0201, 0.0072, -0.0039, -0.0035\} \quad \text{for } 0 \le n \le 18;$$

Evidently, $h[n]$ is a symmetric impulse response that satisfies the Type I, therefore we could get its z-transform in the below format:

$$H(z) = -0.0035 - 0.0039z^{-1} + 0.0072z^{-2} + 0.0201z^{-3} - 0.0517z^{-5} - 0.0506z^{-6} + 0.0855z^{-7} + 0.2965z^{-8} \\ 0.4008z^{-9} + 0.2965z^{-10} + 0.0855z^{-11} - 0.0506z^{-12} - 0.0517z^{-13} + 0.0201z^{-15} + 0.0072z^{-16} \\ - 0.0039z^{-17} - 0.0035z^{-18}$$

- MATLAB codes in this section:

q6_3_1a.m

```
clear;
h=[-0.0035,-0.0039,0.0072,0.0201,-0.0000,-0.0517,-0.0506,0.0855,0.2965,
0.4008,0.2965,0.0855,-0.0506,-0.0517,-0.0000,0.0201,0.0072,
-0.0039,-0.0035]
figure(1);
[mag,phase]=FreRes(h,1);
subplot(132),plot(linspace(-pi,pi,1000),-3*ones(1,1000),'.-')
figure(2);
P=h;%Since h is symmetrical impulse response;
D=[1 zeros(1,18)];
zeros = roots(P)
poles = roots(D)
zplane(zeros,poles)
```

- Use **FreRes.m** to calculate the magnitude and frequency responses:

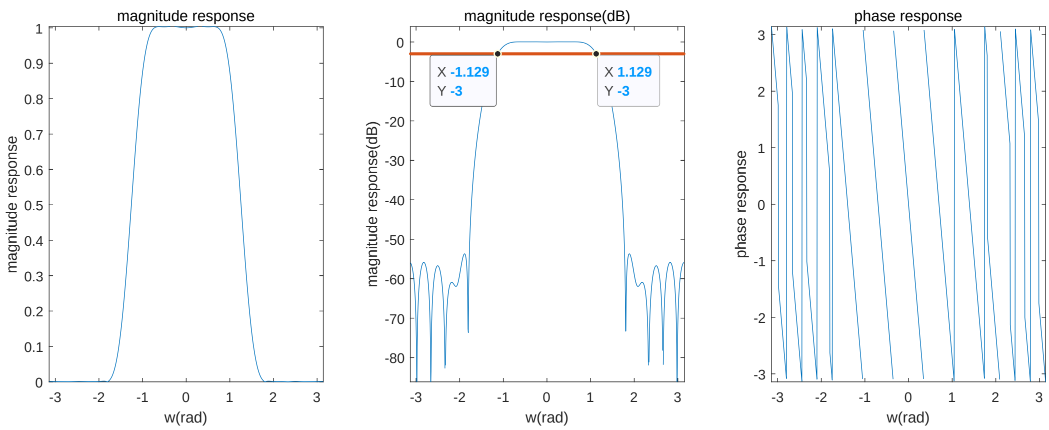

Figure 6: Frequency response of the filter $h[n]$

- zero-pole plots of the filter:

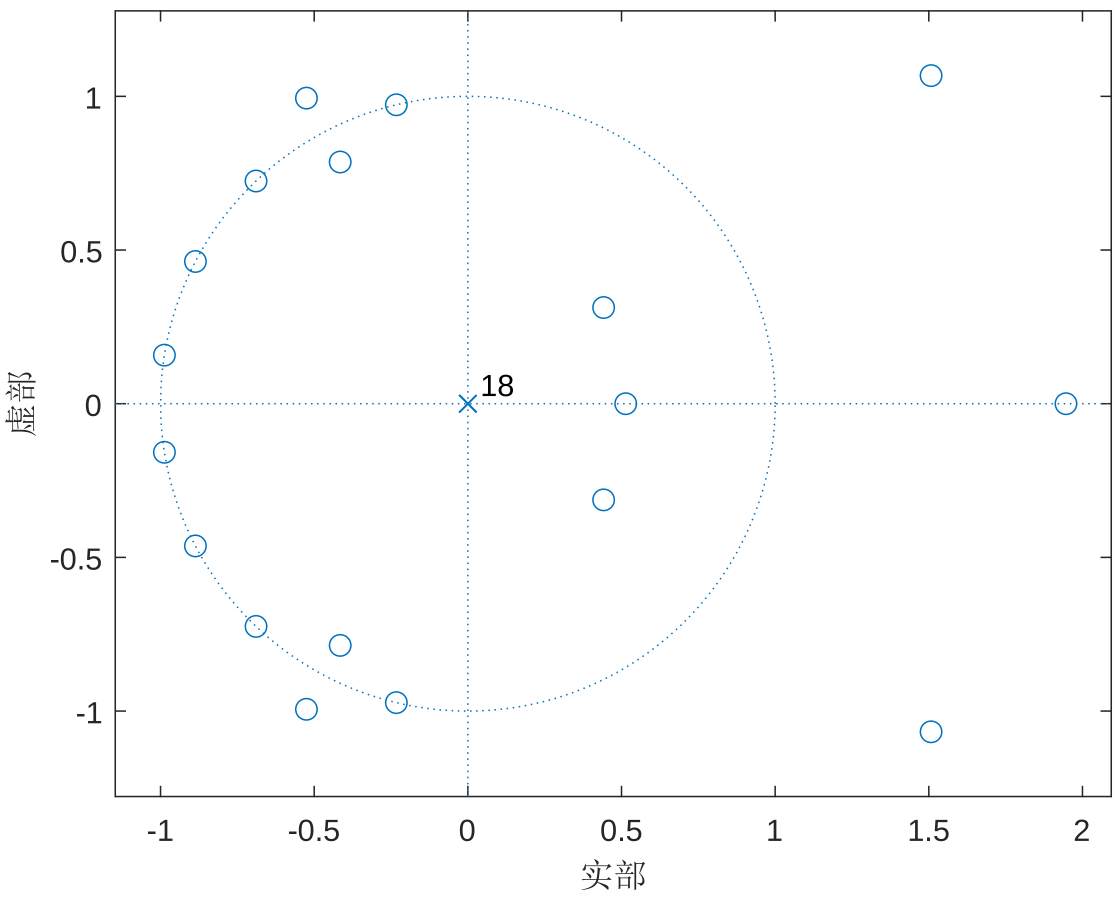

Figure 7: zero-pole plots of the filter

- Analysis for figure 7:
  The zero-pole plots of figure 7 just suit the property of Type I Linear Phase FIR filter such that Either an even number or no zeros at $z = 1$ and $z = -1$ and $H(z) = z^{-(N-1)}H\left(z^{-1}\right)$(Mirror Image Symmetry of Zeros).
- Answers to the two questions:
  - Question 1 (magnitude characteristic):
    * According to Figure 6, $H(e^{j\omega})$ is a lowpass filter but not an ideal filter since the filter not exactly has a magnitude response equal to one in the passband and zero in the stopband, and has a zero phase everywhere.
    * And if we use 3dB cutoff frequency to identify the passband, as shown in the markers of figure, the cutoff frequency ω_c equals approximately ± 1.129(rad).
    * The passband of the lowpass filter is $\omega \in [-1.129, 1.129]$ in rad with a length 2.258(rad).
    * The stopband of the lowpass filter is $\omega \in [-\pi, -1.129] \cup [1.129, \pi]$.
  - Question 2 (If a type III filter may be designed to be a lowpass filter?):
    * Evidently, type III filter can not be designed as a lowpass filter. And I just give a counterexample that just satisfies Anti-symmetrical impulse response with h=[1 -2 3 0 -3 2 -1].
    * MATLAB code for counterexample:

q6_3_1b.m

```
clear;
h=[1 -2 3 0 -3 2 -1];
[mag,phase]=FreRes(h,1);
```

* Result:

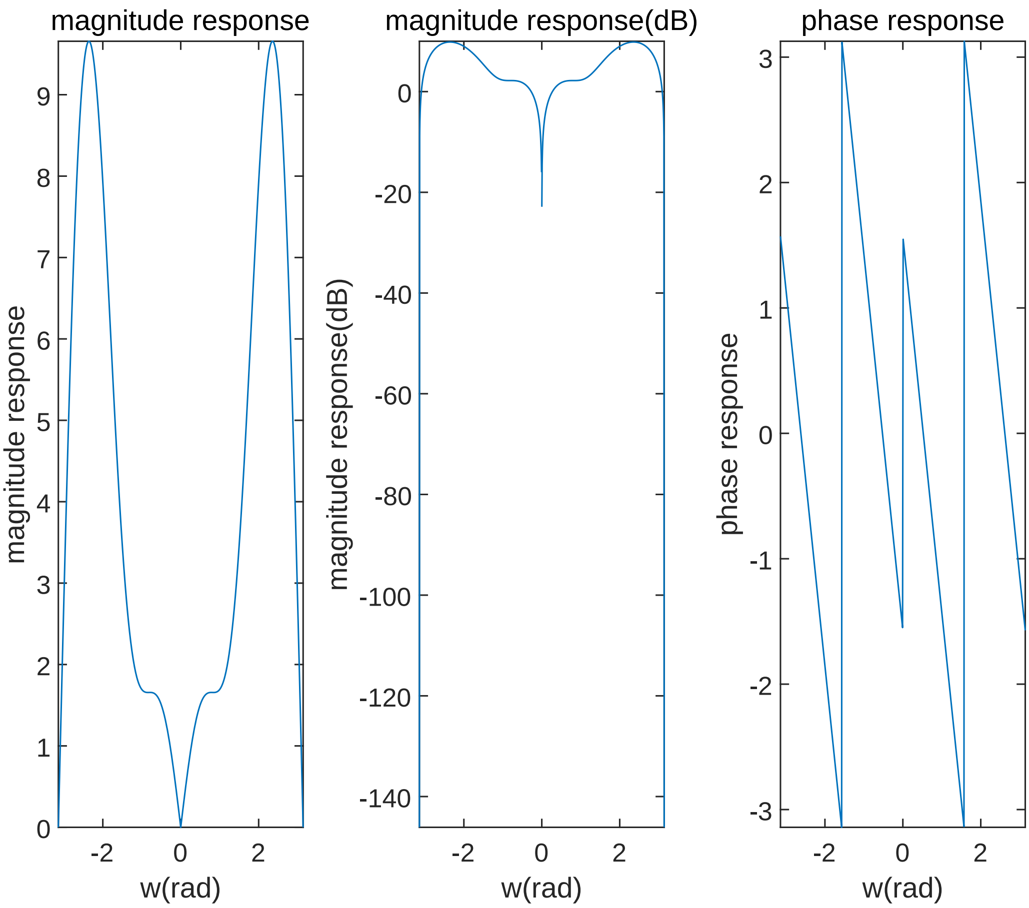

Figure 8: Counterexample for Question 2

* Analysis:
According to above figure, the frequency response of Type III can not be a lowpass filter but more like a bandpass filter.

B. INLAB report II for section 6.3.1

- $h_1[n]$ and $h_2[n]$ considered in this section:

$$h_1[n] = \begin{cases} h[n], & \text{for even } n \\ -h[n], & \text{for odd } n \end{cases}$$

$$h_2[n] = \begin{cases} h[n/5], & \text{for } n = 5k \text{ for interger } k \\ 0, & \text{otherwise} \end{cases}$$

- MATLAB code for this section:

q6_3_1c.m

```
clear;
n=0:18;
h=[-0.0035,-0.0039,0.0072,0.0201,-0.0000,-0.0517,-0.0506,0.0855,0.2965,
0.4008,0.2965,0.0855,-0.0506,-0.0517,-0.0000,0.0201,
0.0072,-0.0039,-0.0035]
h1=h;h2=zeros(1,91);
h1(2:2:19)=-h1(2:2:19);
h2(1:5:91)=h(1:1:19);
figure(1);% frequency response of h1[n]
[mag1,phase1]=FreRes(h1,1);
figure(2);% frequency response of h2[n]
[mag2,phase2]=FreRes(h2,1);
```

- Frequency response of $h_1[n]$:

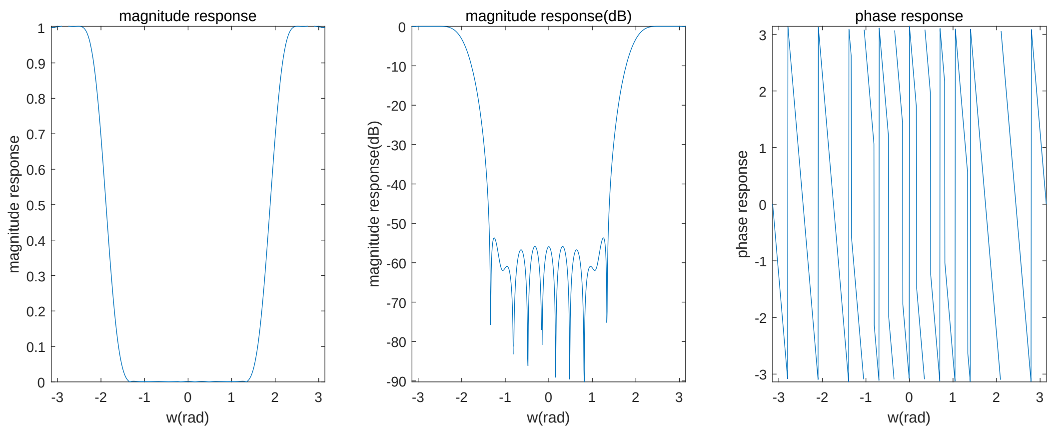

Figure 9: Frequency response of $h_1[n]$

- Frequency response of $h_2[n]$:

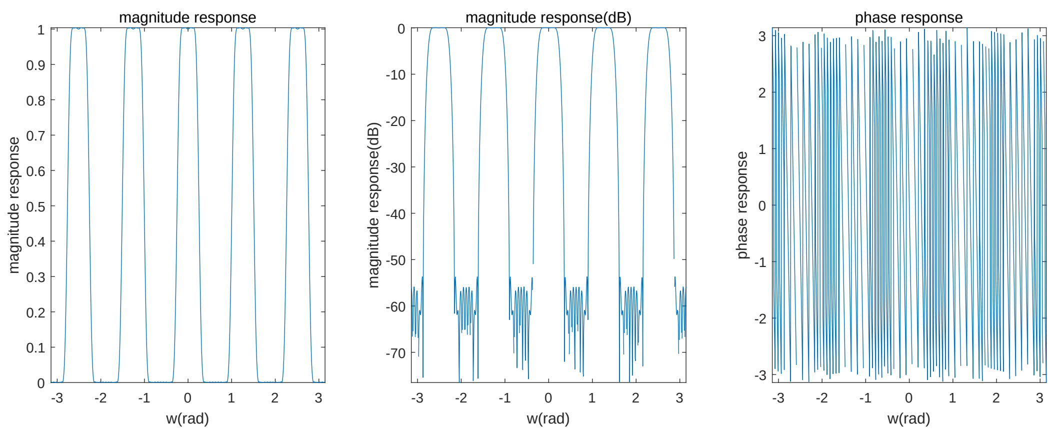

Figure 10: Frequency response of $h_2[n]$

- Analysis:

– For $h_1[n]$, we firstly derive the relationship of z-transform between $h_1[n]$ and $h[n]$:

$$\begin{aligned}
H(-z) &= \sum_{n=-\infty}^{\infty} h[n](-z)^{-n} \\
&= \sum_{n=-\infty}^{\infty} h[n](z)^{-n}(-1)^{-n} \\
&= \sum_{n=-\infty}^{\infty} h\left[n_{\text{even}}\right](z)^{-n_{\text{even}}} - \sum_{n=-\infty}^{\infty} h\left[n_{\text{odd}}\right](z)^{-n_{\text{odd}}} \\
&= H_1(z)
\end{aligned}$$

$\therefore$ we have $H_1(e^{j\omega}) = H(-e^{j\omega}) = H(e^{j(\omega+\pi)})$. This will make the frequency response of $H_1(z)$ is a right frequncy shift with π of $H(z)$.
By comparing the frequency responses of Figures 6 and 9, we could verify the above conclusion.
Some interesting properties of $H_1(e^{j\omega})$:

* $H_1(e^{j\omega})$ is a highpass filter.
* The bandwidth of $H_1(e^{j\omega})$ equlas $\pi-$bandwidthof $H(e^{j\omega})$.

– For $h_2[n]$, we have below general property for time-scaling signal:

Info: If $x_{(k)}[n]$ satisfies:

$$x_{(k)}[n] = \begin{cases} x[n/k], & \text{if } n \text{ is a multiple of } k \\ 0, & \text{if } n \text{ is not a multiple of } k \end{cases}$$

Then we have:

$$X_{(k)}\left(e^{j\omega}\right) = \sum_{n=-\infty}^{+\infty} x_{(k)}[n]e^{-j\omega n} = \sum_{r=-\infty}^{+\infty} x_{(k)}[rk]e^{-j\omega rk}$$

Furthermore, since $x_{(k)}[rk] = x[r]$, we can find that:

$$X_{(k)}\left(e^{j\omega}\right) = \sum_{r=-\infty}^{+\infty} x[r]e^{-j(k\omega)r} = X\left(e^{jk\omega}\right)$$

Finally, that is:

$$x_{(k)}[n] \stackrel{\text{Ft}}{\longleftrightarrow} X\left(e^{jk\omega}\right)$$

$\therefore$ For this case, we have $H_2(e^{j\omega}) = H(e^{j5\omega})$.
This means that the period of $H_2(e^{j\omega})$ is 1/5 of $H(e^{j5\omega})$. And if we compare the Figrues 6 and 10, we can verify this conclusion.

6.3.2 Design of Simple FIR Filters

Info:

- In this section, we need to design a linear phase lowpass FIR filter with a 3-dB cutoff frequency at 0.3π using two methods:
  - A cascade of first-order lowpass filters.
  - A higher order moving average filter.
- First-order lowpass filter: $H_0(z) = \frac{1+z^{-1}}{2} = e^{-j\omega/2}\cos(\omega/2)$.
  M-order lowpass filter:
$$H(z) = \left(\frac{1+z^{-1}}{2}\right)^M$$
$$H(e^{j\omega}) = e^{-jM\omega/2}\cos^M(\omega/2)$$

Info:

- To get the number of stages of the lowpass filter:
$$\text{Solve the equation} \qquad \cos^M(\omega_c/2) = \frac{\sqrt{2}}{2}$$
  $\therefore \omega_c = 2\cos^{-1}\left(2^{-1/(2M)}\right)$.
  When $\omega_c = 0.3\pi \longrightarrow M = 3$.
- A cascade of first-order lowpass filters:
$$H(z) = \left(\frac{1+z^{-1}}{2}\right)^M = \left(\frac{1+z^{-1}}{2}\right)^3$$
$$H(e^{j\omega}) = e^{-j3\omega/2}\cos^3(\omega/2)$$
- A higher order moving average filter:
$$H(z) = \frac{1}{M}\sum_{m=0}^{M-1} z^{-m} \longrightarrow H\left(e^{j\omega}\right) = \frac{1}{M}\cdot\frac{\sin\left(\frac{M\omega}{2}\right)}{\sin\left(\frac{\omega}{2}\right)}e^{-\frac{j(M-1)\omega}{2}}$$
$$H\left(e^{j\omega}\right) = \frac{1}{3}\cdot\frac{\sin\left(\frac{3\omega}{2}\right)}{\sin\left(\frac{\omega}{2}\right)}e^{-\frac{j2\omega}{2}}$$

- MATLAB code for this section:

q6_3_2.m

```
clear;
w = linspace(0,pi,5000);
M = 3;
H_cascade = exp(-j*3*w/2).*(cos(w/2)).^3;
H_moving_average = (1/M)*sin(M.*w/2)./sin(w/2).*exp(-1j*(M-1).*w./2);
plot(w,20*log10(abs(H_cascade)));
hold on;
plot(w,20*log10(abs(H_moving_average)));
plot(w,-3*ones(1,5000));
ylim([-50,0]);
legend('the cascade filter','higher order moving average filter','3-dB
    cutoff line');
```

- The plot of two designs and 3dB cutoff frequency:

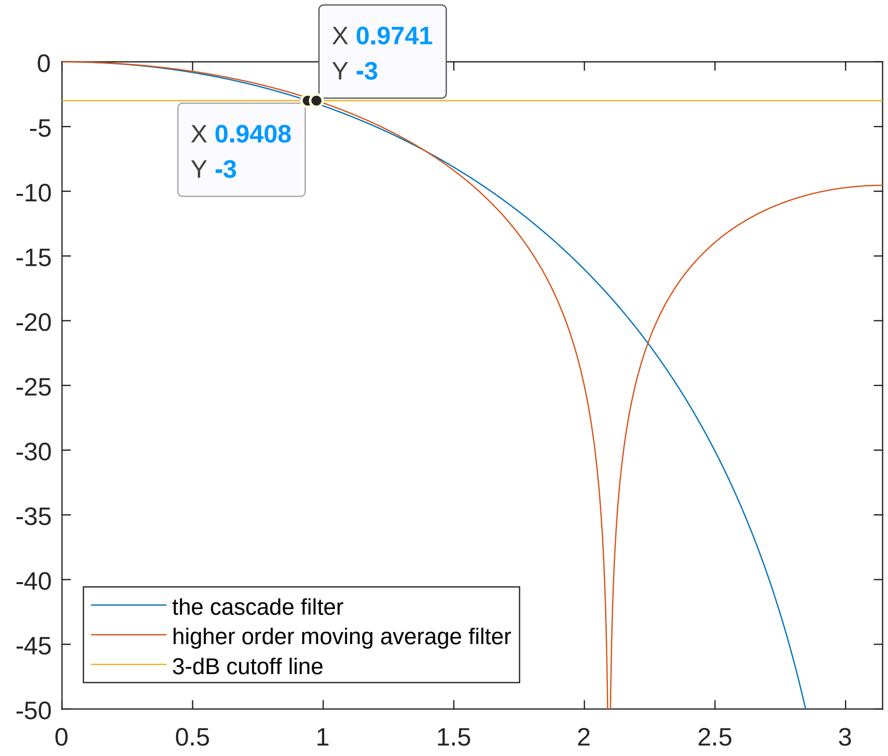

Figure 11: The plot of two designs and 3dB cutoff frequency

- A tiny look for cutoff frequency:

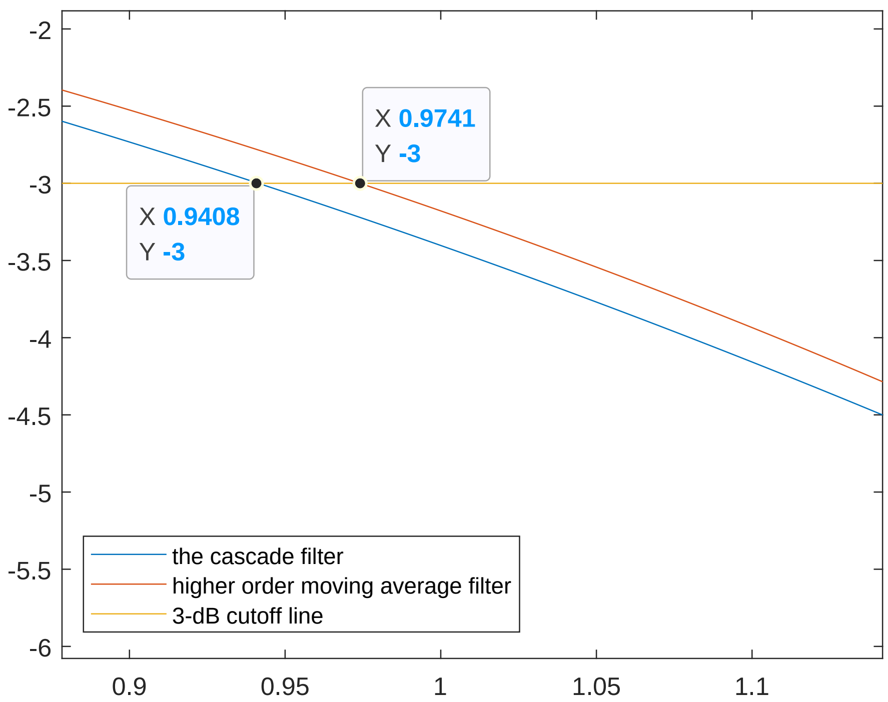

Figure 12: A tiny look for cutoff frequency

- Analysis:
  Cutoff frequencies for cascade filter and moving average filter are respectively 0.9408(rad) and 0.9471(rad), which approximately equal to 0.3π, therefore the design of the two filters are successful.

6.4 IIR Filters

Info:

- In this section, we will design highpass IIR filter with a 3-dB cutoff frequency of 0.2π.
- Compute the 3-dB cutoff frequency for highpass filter: Given the first order highpass IIR filter with a zero at $z = 1$:

$$H(z) = \frac{1+\alpha}{2}\frac{1-z^{-1}}{1-\alpha z^{-1}}, \quad 0 < |\alpha| < 1$$

  We could calculate the squared magnitude function:

$$\left|H_{HP}\left(e^{jw}\right)\right|^2 = H\left(e^{jw}\right)H\left(e^{-jw}\right) = \frac{(1+\alpha)^2(1-cosw)}{2\left(1+\alpha^2-2\alpha\cos w\right)}$$

  – Situation 1: Given α, calculate ω_c:

$$\frac{(1+\alpha)^2(1-cosw)}{2\left(1+\alpha^2-2\alpha\cos w\right)} = \frac{1}{2}$$

  which, when solved, yields:

$$\cos\omega_c = \frac{2\alpha}{1+\alpha^2} \longrightarrow \omega_c = \cos^{-1}\frac{2\alpha}{1+\alpha^2}$$

  – Situation 2: Given ω_c, calculate α:

$$\alpha^2\cos\omega_c - 2\alpha + \cos\omega_c = 0 \longrightarrow \alpha = \frac{1-\sqrt{1-\cos^2\omega_c}}{\cos\omega_c}$$

  $\therefore \alpha = \frac{1-\sqrt{1-\cos^2\omega_c}}{\cos\omega_c}$

- In this case, when $\omega_c = 0.2\pi$, $\alpha = \frac{1-\sqrt{1-\cos^2 0.2\pi}}{\cos 0.2\pi} \approx 0.5095$

- MATLAB codes in this section:

q6_4.m

```
clear;
alpha = (1-sqrt(1-cos(0.2*pi)^2))/cos(0.2*pi);
w = linspace(0,pi,5000);
H = sqrt(((1+alpha)^2*(1-cos(w))./(2*(1+alpha^2-2*alpha*cos(w))));
plot(w,20*log10(abs(H)));
hold on;
plot(w,-3*ones(1,5000));
xlabel('w(rad)'),ylabel('magnitude response in dB'),title('Simple
    highpass IIR filter');
legend('Simple highpass IIR filter','3-dB cutoff line');
```

- Simple highpass IIR filter:

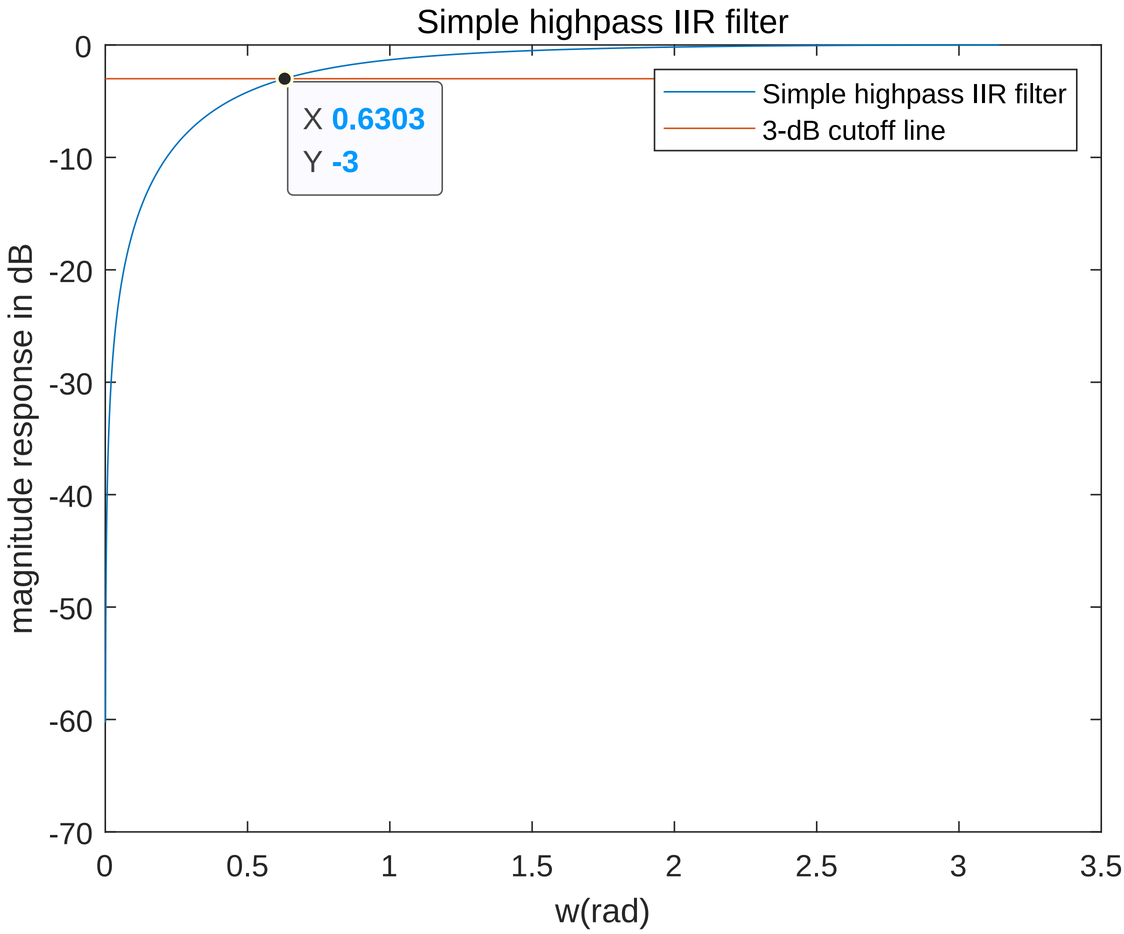

Figure 13: Simple highpass IIR filter

- Analysis:
  Resulted cutoff frequency is approximately 0.6303, almost equal to 0.2π. This indicates the design of the lowpass filter is successful.

6.5 Summary & Experience

1. In this lab, we mainly completed some specific algorithms of z-transform and some basic designs of linear phase filter and IIR filters.
2. Through this lab, we could easily recover the results both in the labs and lectures, and therefore get a deeper understanding of the related knowledge.
3. Through the designs of some simple filters, the relationship between DTFT and z-transform is emphasized again, which really makes a sense to the applications of z-transform.
4. An interesting bug I found: When it comes to drawing zero-pole plane, the inputs of MATLAB function **zplane.m** must be column vectors, or we will not get the correct output.