110705067 洪薇欣

6.67

題目：Use the MATLAB command **pzmap** to plot the poles and zeros of the following systems:

(a)

$$H(s) = \frac{s^3 + 1}{s^4 + 2s^2 + 1}$$

(b)

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 1 & -6 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ 2 \end{bmatrix},$$
$$\mathbf{c} = [0 \quad 1], \quad D = [0]$$

作法：

The command r = roots (a) finds the roots of a polynomial described by the vector a and thus may be used to determine the zeros and poles of a Laplace transform expressed as a ratio of polynomials ins. The elements of the vector a correspond to descending powers of s.

The command H = zpk (z , p , k) (sys = zpk(zeros,poles,gain)) creates an LTI object (continuous-time zero-pole-gain model) representing the pole-zero-gain form of system description. The zeros and poles are described by the vectors z and p, respectively, and the gain is represented by the scalar k. The output sys is a zpk model object storing the model data. Set zeros or poles to [] for systems without zeros or poles. These two inputs need not have equal length and the model need not be proper (that is, have an excess of poles).

The commands tf convert among models when applied to an LTI object of a different form. For example, if syszpk is an LTI object representing a system in zero-pole-gain form, then the command sys = tf(syszpk) converts the dynamic system model syszpk to a transfer function model.

pzmap (sys) produces a pole-zero plot.

sys = ss(A,B,C,D) creates a continuous-time state-space model object of the following form:
x=Ax+Bu, y=Cx+Du

For instance, consider a plant with Nx states, Ny outputs, and Nu inputs. The state-space matrices are:
A is an Nx-by-Nx real- or complex-valued matrix.
B is an Nx-by-Nu real- or complex-valued matrix.
C is an Ny-by-Nx real- or complex-valued matrix.
D is an Ny-by-Nu real- or complex-valued matrix.

(a) 先用 roots 找出 zeros and poles (described by the vectors z and p)，再用 pzmap 畫出 pole-zero plot
(b) 先用 ss(A,B,C,D) creates a continuous-time state-space model object，再用 pzmap 畫出 pole-zero plot

(a)

```
%6.67a

z=roots([1,0,0,1]);

p=roots([1,0,2,0,1]);

k=1;

syszpk=zpk(z,p,k);

systf=tf(syszpk);

pzmap(systf);
```

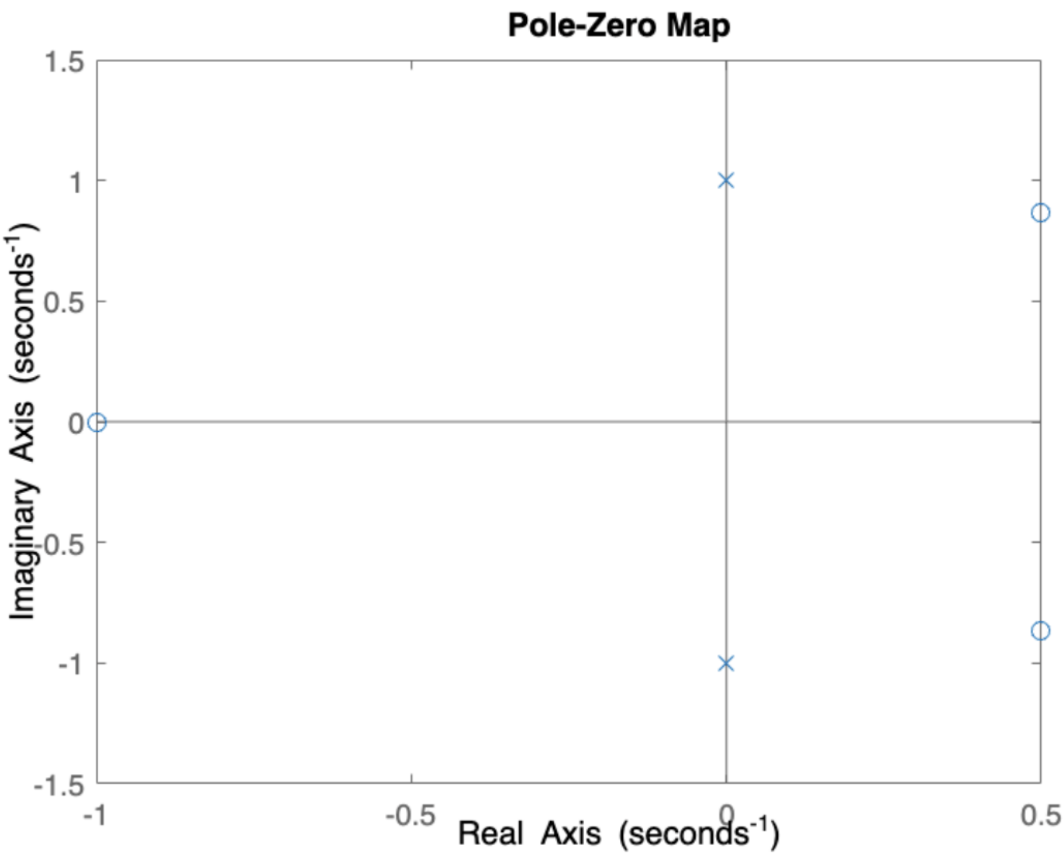
(b)

```
%6.67b

A=[1,2;1,-6];

b=[1;2];

c=[0,1];

D=[0];
```

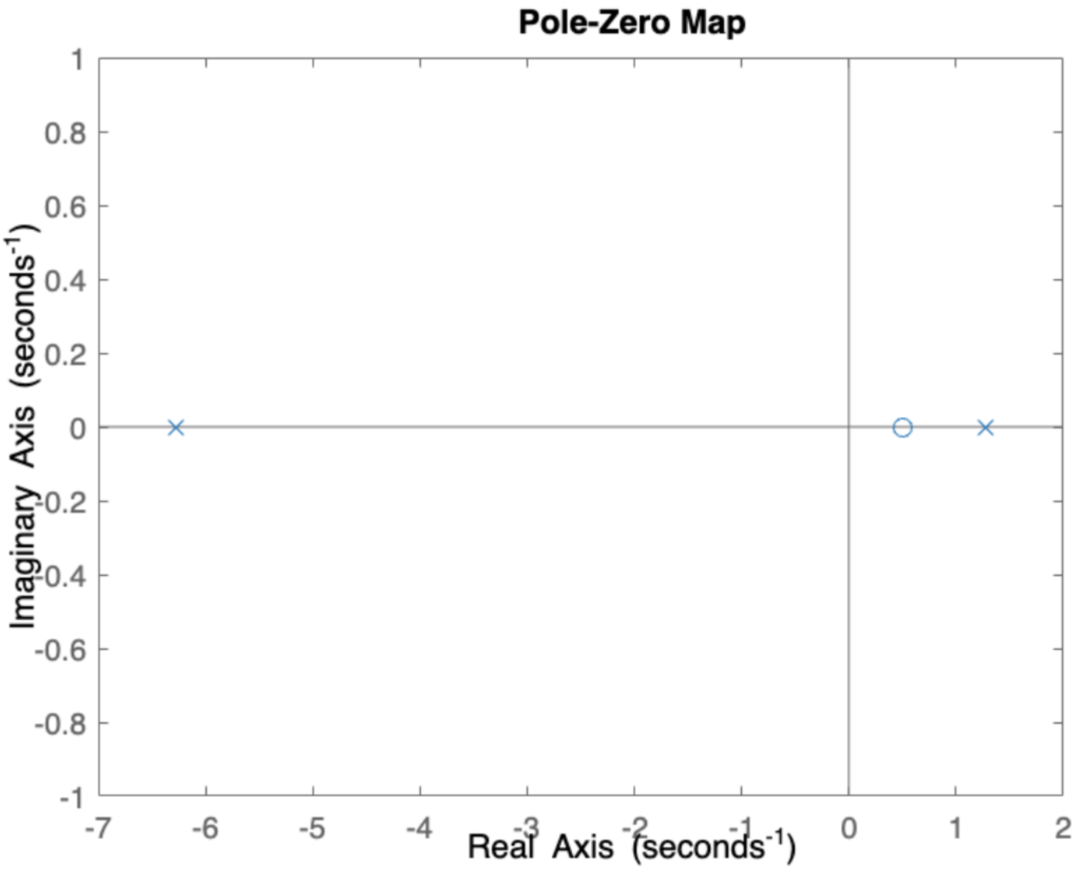
```
pzmap(ss(A,b,c,D));
```

結果：

(a)



(b)



6.70

題目：Use your knowledge of the effect of poles and zeros on the magnitude response, to design systems having the specified magnitude response. Place poles and zeros in the s-plane, and evaluate the corresponding magnitude response using the MATLAB command **fregresp** . Repeat this process until you find poles and zeros that satisfy the specifications.

(a) Design a high-pass filter with two poles and two zeros that satisfies $|H(j0)|= 0$, $0.8 \leq |H(j\omega)| \leq 1.2$, for $|\omega| > 100\pi$ and that has real-valued coefficients.

(b) Design a low-pass filter with real-valued coefficients that satisfies $0.8 \leq |H(j\omega)| \leq 1.2$ for $|\omega| < \pi$ and $|H(j\omega)| < 0.1$ for $|\omega| > 10\pi$.

作法：

The command H = zpk (z , p , k) (sys = zpk(zeros,poles,gain)) creates an LTI object (continuous-time zero-pole-gain model) representing the pole-zero-gain form of system description. The zeros and poles are described by the vectors z and p, respectively, and the gain is represented by the scalar k. The output sys is a zpk model object storing the model data. Set zeros or poles to [] for systems without zeros or poles. These two inputs need not have equal length and the model need not be proper (that is, have an excess of poles).

The command H = tf (b , a) creates an LTI object H representing a transfer function with numerator and denominator polynomials defined by the coefficients in b and a, ordered in descending powers of s.The commands tf convert among models when applied to an LTI object of a different form. For example, if syszpk is an LTI object representing a system in zero-pole-gain form, then the command sys = tf(syszpk) converts the dynamic system model syszpk to a transfer function model.

Additional commands that apply directly to LTI objects include freqresp for determining the frequency response. [response, outputfreq] = freqresp(fit,inputfreq) calculates the frequency response, response of the fit of a rationalfit function object or a rational object at the specified input frequencies, inputfreq.

B = squeeze(A) returns an array with the same elements as the input array A, but with dimensions of length 1 removed. For example, if A is a 3-by-1-by-1-by-2 array, then squeeze(A) returns a 3-by-2 matrix.
If A is a row vector, column vector, scalar, or an array with no dimensions of length 1, then squeeze returns the input A.

將符合條件的一個 transfer function 用 zpk(z,p,k) 做出 continuous-time zero-pole-gain model，z 為 zeros, p 為 poles，再用 tf 函式轉換出 transfer function。再用 freqresp 做出 frequency response，然後用 squeeze 得到其元素與輸入 array 相同，但刪除了長度為 1 的 array，最後用 plot 將 frequency response 畫出。

Transfer function:
(a) $H(s) = s^2 /((s+25+j10\pi)(s+25-j10\pi))$
(b) $H(s) = (0.005(s-j50)(s+j50)) /((s+2+j\pi)(s+2-j\pi))$

(a)

```
%6.70a

z=[0,0];

k=1;

p=[25+j*10*pi,25-j*10*pi];
```

```
syszpk=zpk(z,p,k);

systf=tf(syszpk); %convert to transfer function form

w=[0:499]*200*pi/500; %Frequencies from 0 to 628 rad/s

H=freqresp(systf,w);

Hmag=abs(squeeze(H));

plot(w,Hmag);

title('High Pass Filter');

xlabel('w:rad/s');

ylabel('|H(w)|');
```

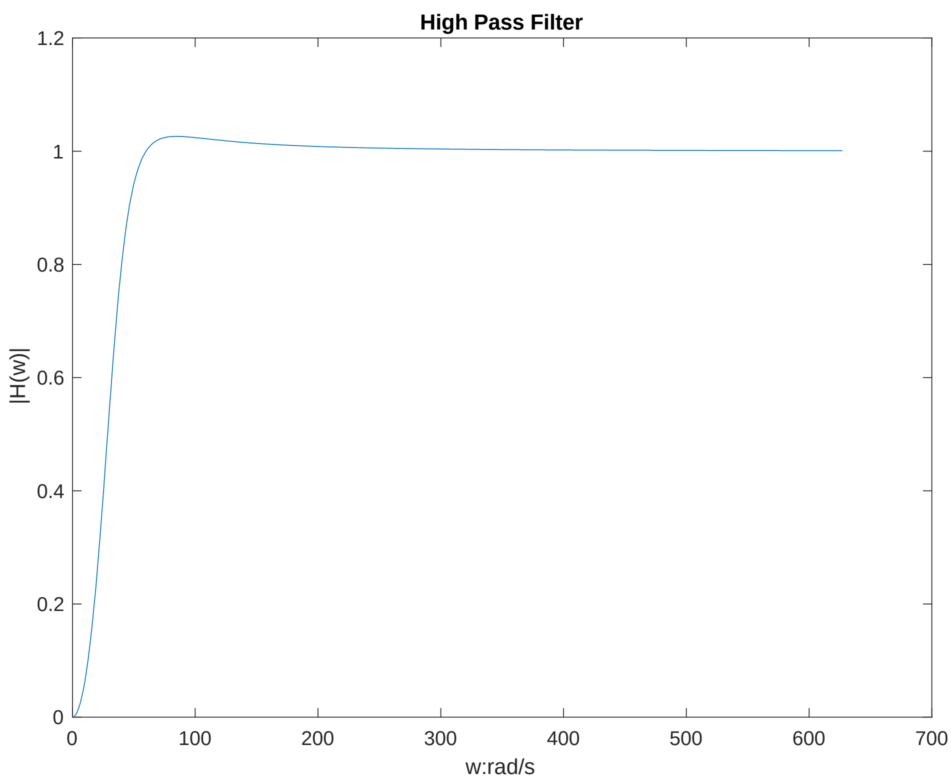
(b)

```
%6.70b

z=[j*50,-j*50];

k=0.005;

p=[2+j*pi,2-j*pi];

syszpk=zpk(z,p,k);

systf=tf(syszpk); %convert to transfer function form

w=[0:50]*200*pi/500; %Frequencies from 0 to 62.8 rad/s

H=freqresp(systf,w);

Hmag=abs(squeeze(H));
```

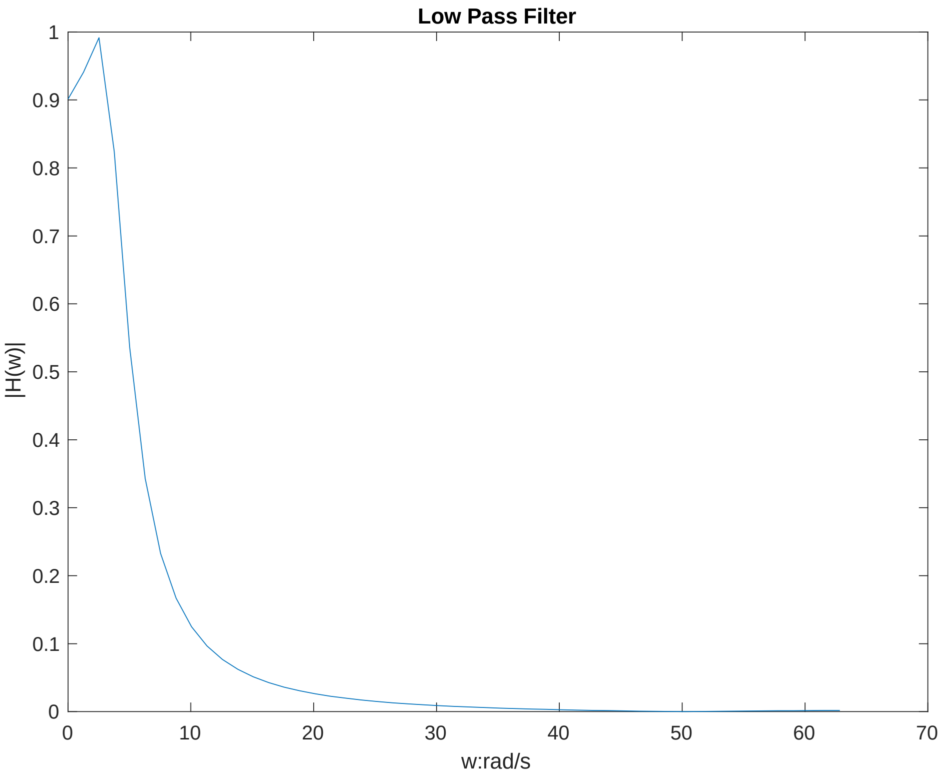
```
plot(w,Hmag);

title('Low Pass Filter');

xlabel('w:rad/s');

ylabel('|H(w)|');
```

結果：

(a)



(b)



6.73

題目：Use the MATLAB command **tf** to find transfer function descriptions of the systems in Problem 6.49.

Problem 6.49:

(a) Use the time-differentiation property to show that the transfer function of an LTI system is expressed in terms of the state-variable description as

$$H(s) = \mathbf{c}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{b} + D.$$

(b) Determine the transfer function, impulse response, and differential-equation descriptions of a stable LTI system represented by the following state-variable descriptions:

(i) $\mathbf{A} = \begin{bmatrix} -1 & 1 \\ 0 & -2 \end{bmatrix}$, $\mathbf{b} = \begin{bmatrix} 3 \\ -1 \end{bmatrix}$,
$\mathbf{c} = [1 \quad 2]$, $D = [0]$

(ii) $\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 1 & -6 \end{bmatrix}$, $\mathbf{b} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$,
$\mathbf{c} = [0 \quad 1]$, $D = [0]$

作法：

[b,a] = ss2tf(A,B,C,D) converts a state-space representation of a system into an equivalent transfer function. ss2tf returns the Laplace-transform transfer function for continuous-time systems and the Z-transform transfer function for discrete-time systems.

The command H = tf (b , a) creates an LTI object H representing a transfer function with numerator and denominator polynomials defined by the coefficients in b and a, ordered in descending powers of s. sys = tf(numerator,denominator) creates a continuous-time transfer function model, setting the Numerator and Denominator properties. For instance, consider a continuous-time SISO dynamic system represented by the transfer function sys(s) = N(s)/D(s), the input arguments numerator and denominator are the coefficients of N(s) and D(s), respectively.

bode for determining the Bode plot.

用 ss2tf(A,B,C,D) 將 state-space representation 轉成 transfer function，再將用 ss2tf(A,B,C,D)得到的 b : transfer function numerator coefficients 和 a : transfer function denominator coefficients 用 tf (b , a) 產生出 transfer function，再用 bode 畫出 bode plot。

(b) (i)

```
%6.73bi

A=[-1,1;0,-2];

B=[3;-1];

C=[1,2];

D=[0];

[b,a] = ss2tf(A,B,C,D);

sys=tf(b,a);

bode(sys);
```

(b) (ii)

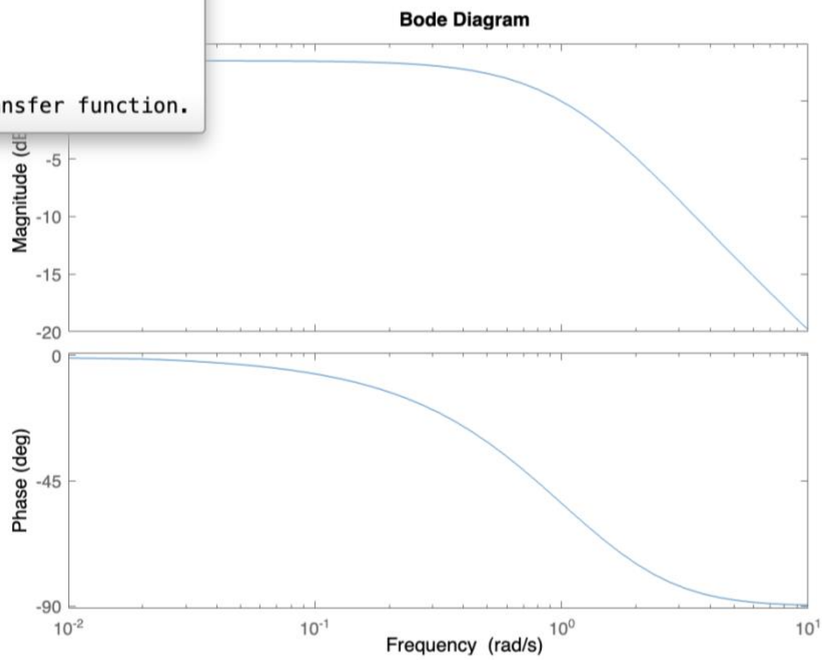
```
%6.73bii

A=[1,2;1,-6];

B=[1;2];

C=[0,1];

D=[0];

[b,a] = ss2tf(A,B,C,D);

sys=tf(b,a);

bode(sys);
```

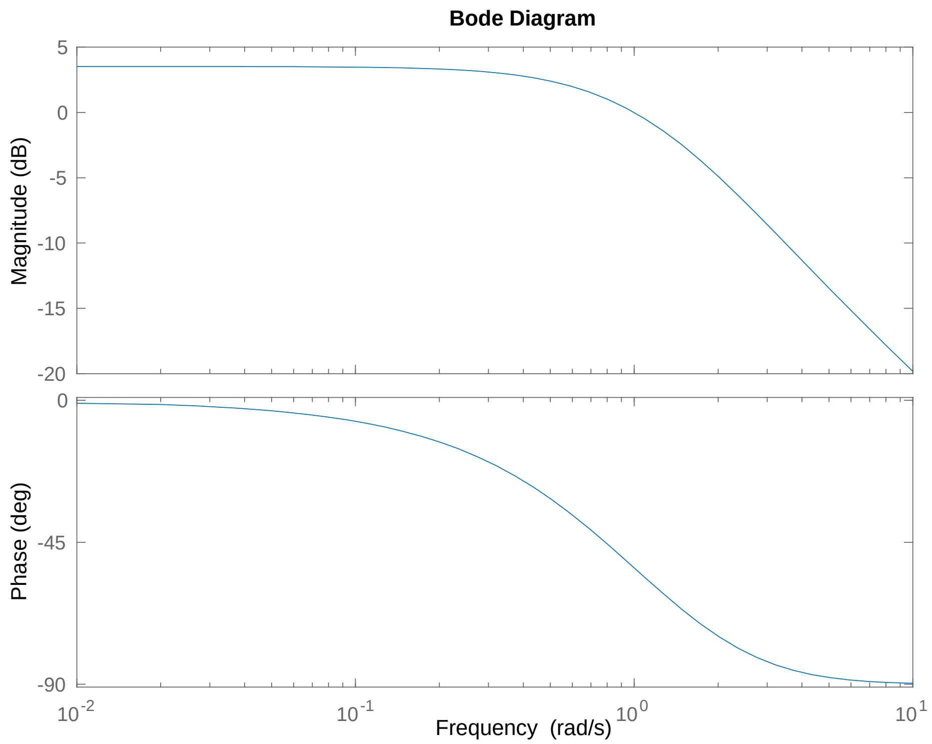
結果：

(b) (i)

```
%6.73bi
A=[-1,1;0,-2];
B=[3;-1];
C=[1,2];
D=[0];
[b,a] = ss2tf(A,B,C,D);
sys=tf(b,a);
bode(sys);
```

```
sys =

      s + 3
  -------------
  s^2 + 3 s + 2

Continuous-time transfer function.
```



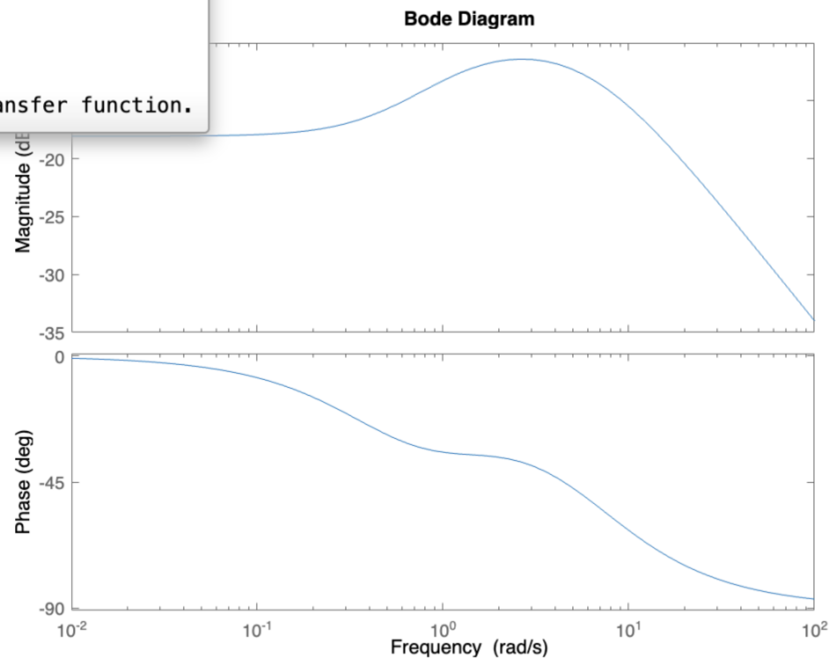


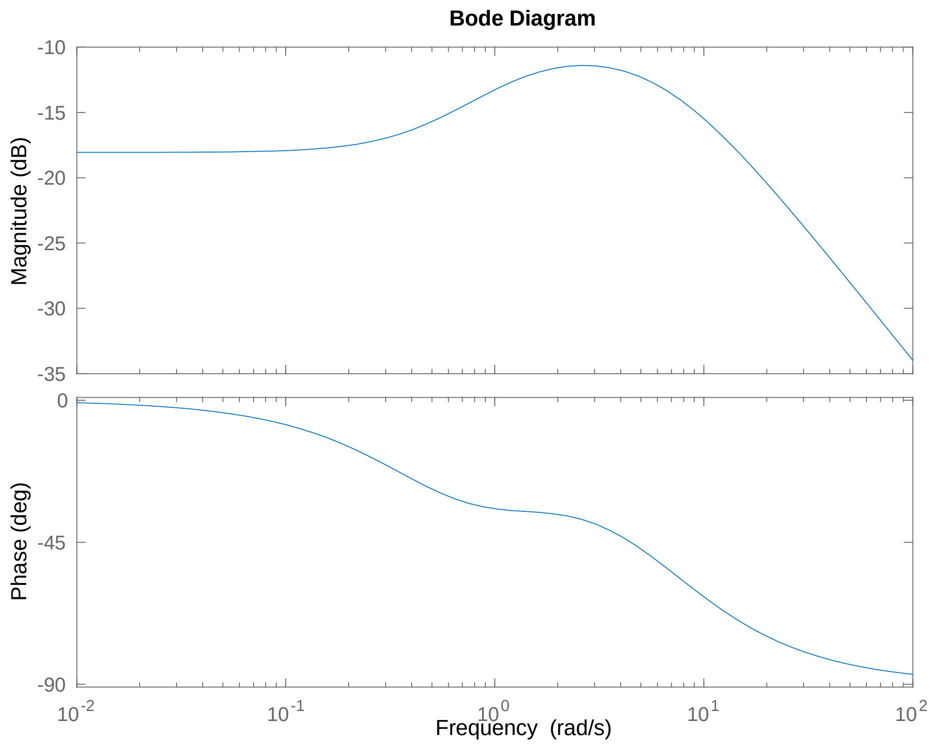
Bode Diagram
5
0
-5
-10
-15
-20
Magnitude (dB)
0
-45
-90
Phase (deg)
10-2
10-1
100
101
Frequency (rad/s)

(b) (ii)

```
%6.73bii
A=[1,2;1,-6];
B=[1;2];
C=[0,1];
D=[0];
[b,a] = ss2tf(A,B,C,D);
sys=tf(b,a);
bode(sys);
```

```
sys =

     2 s - 1
  -------------
  s^2 + 5 s - 8

Continuous-time transfer function.
```




Bode Diagram
Magnitude (dB)
-10
-15
-20
-25
-30
-35
Phase (deg)
0
-45
-90
10^-2
10^-1
10^0
10^1
10^2
Frequency (rad/s)