TYPES OF IIR FILTERS

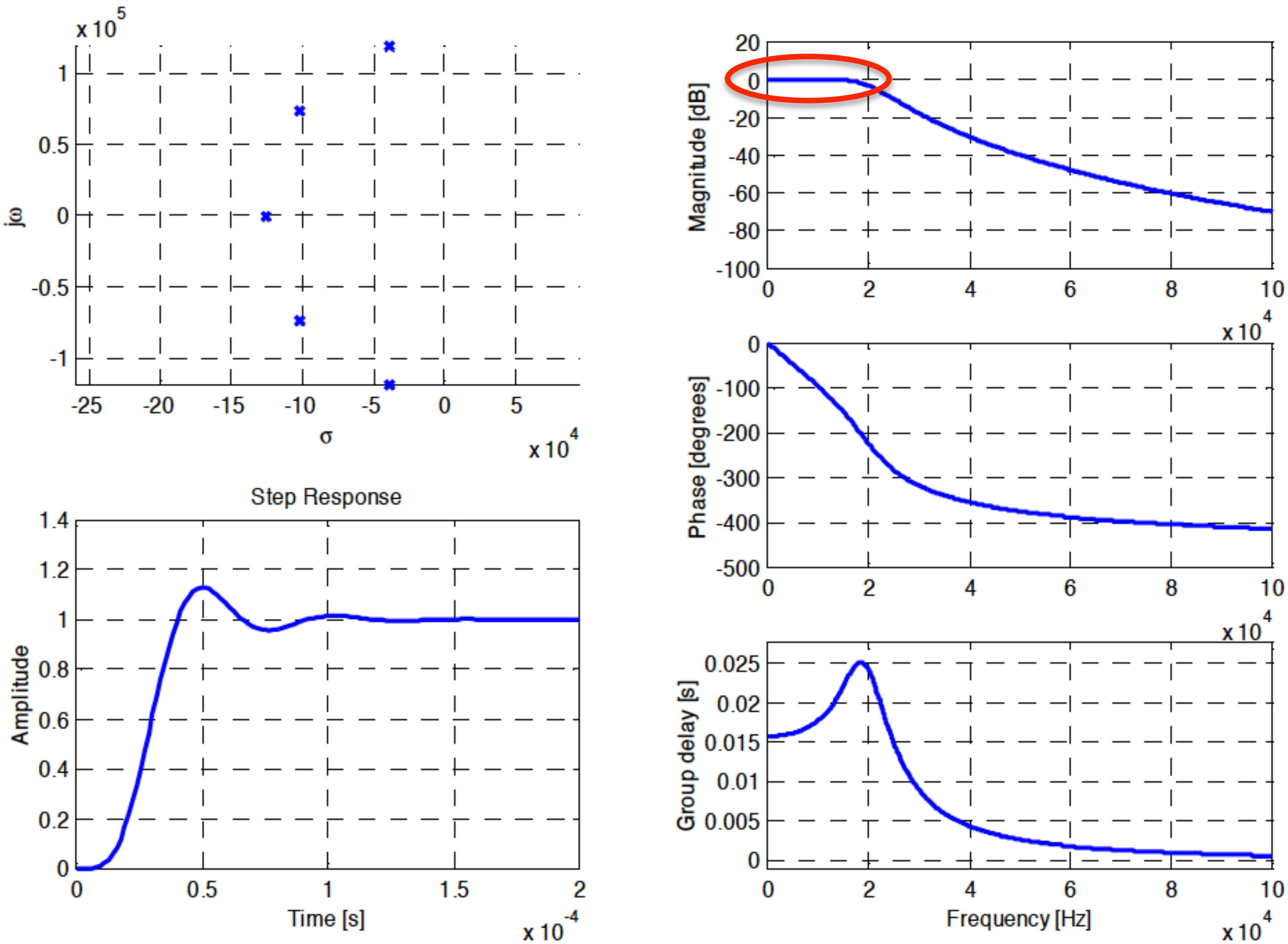
- **Butterworth**
  - Maximally flat magnitude in the passband
- **Chebyshev type I**
  - Equiripple in the passband
- **Chebyshev type II**
  - Equiripple in the stopband
- **Elliptic**
  - Equiripple in passband and stopband
- **Bessel**
  - Maximally flat group delay

CONTINUOUS-TIME FILTERS

- MATLAB command
  - -buttord, butter (BUTTERWORTH)
  - -cheb1ord, cheb1 (CHEBYSHEV TYPE I)
  - -cheb2ord, cheb2 (CHEBYSHEV TYPE II)
  - -ellipord, ellip (ELLIPTIC)
  - -besself (BESSEL)
- Use with ‘s’ option for continuous time
- Frequency are in rad/s
- Attenuation and ripple values are in dB

BUTTERWORTH

```
N = 5; Wn = 2*pi*20e3;
```



BUTTERWORTH

- To design the n^{th} order Butterworth filter use **butter** command:

[B, A] = butter(N, Wn, ‘ftype’, ‘s’);

Where **N** is filter order, **Wn** [rad/s] is cutoff frequency, **ftype** could be either ‘low’, ‘high’, and ‘stop.’ Argument ‘s’ specify for continuous time, if not specified it is used us digital filter

Butter function returns **B** (numerator) and **A** (denominator)

- If order and cutoff frequency is unknown use **buttord** command:

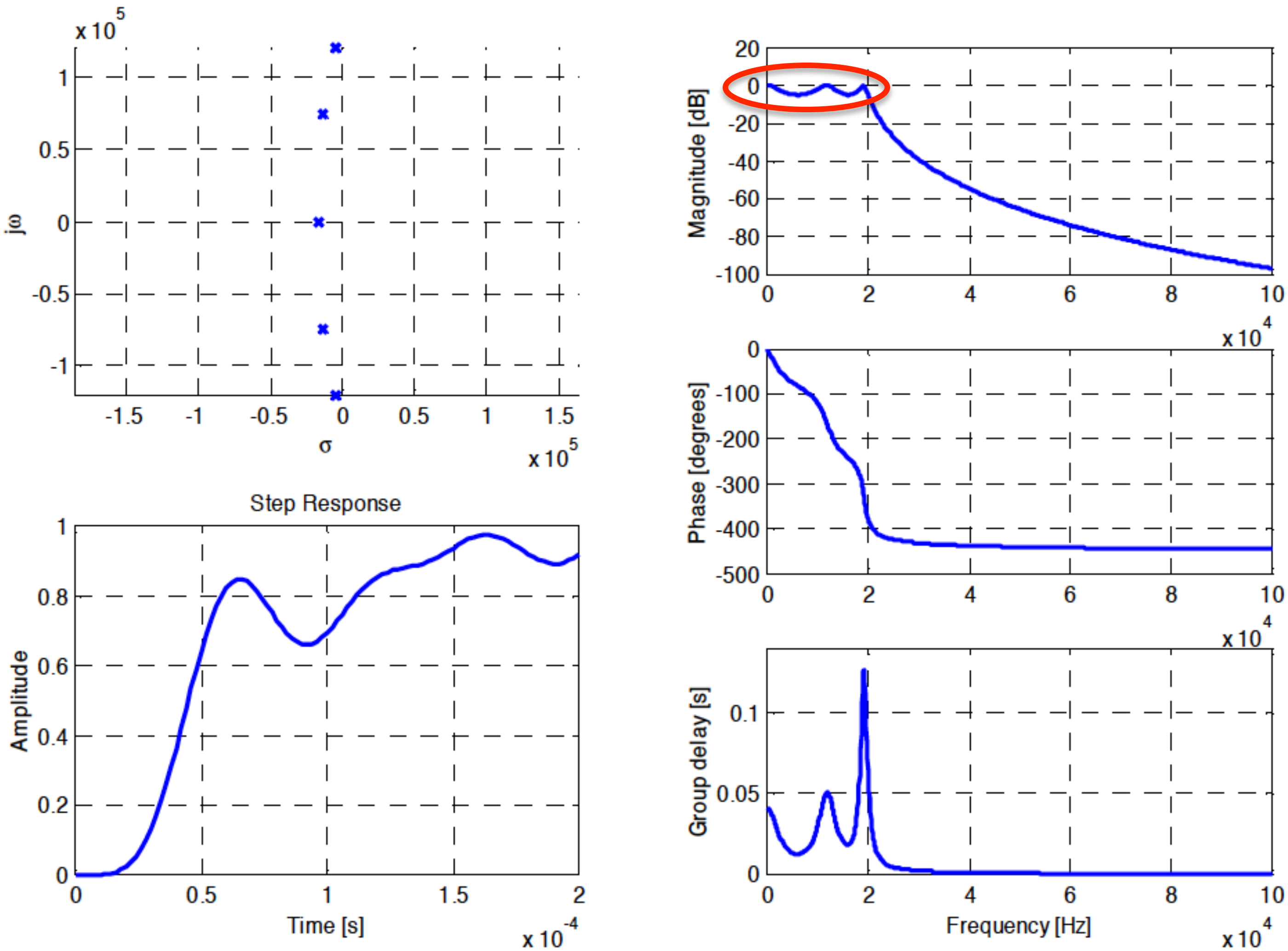
[N, Wn] = buttord(Wp, Ws, Rp, Rs, ‘s’);

Where **Wp**, and **Ws** are the passband and stopband edge frequency in [rad/s]
Rp, and **Rs** are the passband ripple and the stopband attenuation in dB

buttord function returns **N** (minimum filter order) and **Wn** (cutoff frequency)

CHEBYSHEV TYPE I

N = 5; Wn = 2*pi*20e3; Rp = 5 dB;



CHEBYSHEV TYPE I

• To design the n^{th} order Chebyshev type I filter use **cheby1** command:

[B, A] = cheby1(N, R, Wp, ‘ftype’, ‘s’);

Where **N** is filter order, **R** [dB] is peak to peak passband ripple, **Wp** [rad/s] is cutoff frequency, **ftype** could be either ‘low’, ‘high’, and ‘stop.’ Argument ‘s’ specify for continuous time, if not specified it is used us digital filter

cheby1 function returns **B** (numerator) and **A** (denominator)

• If filter order is unknown use **cheb1ord** command:

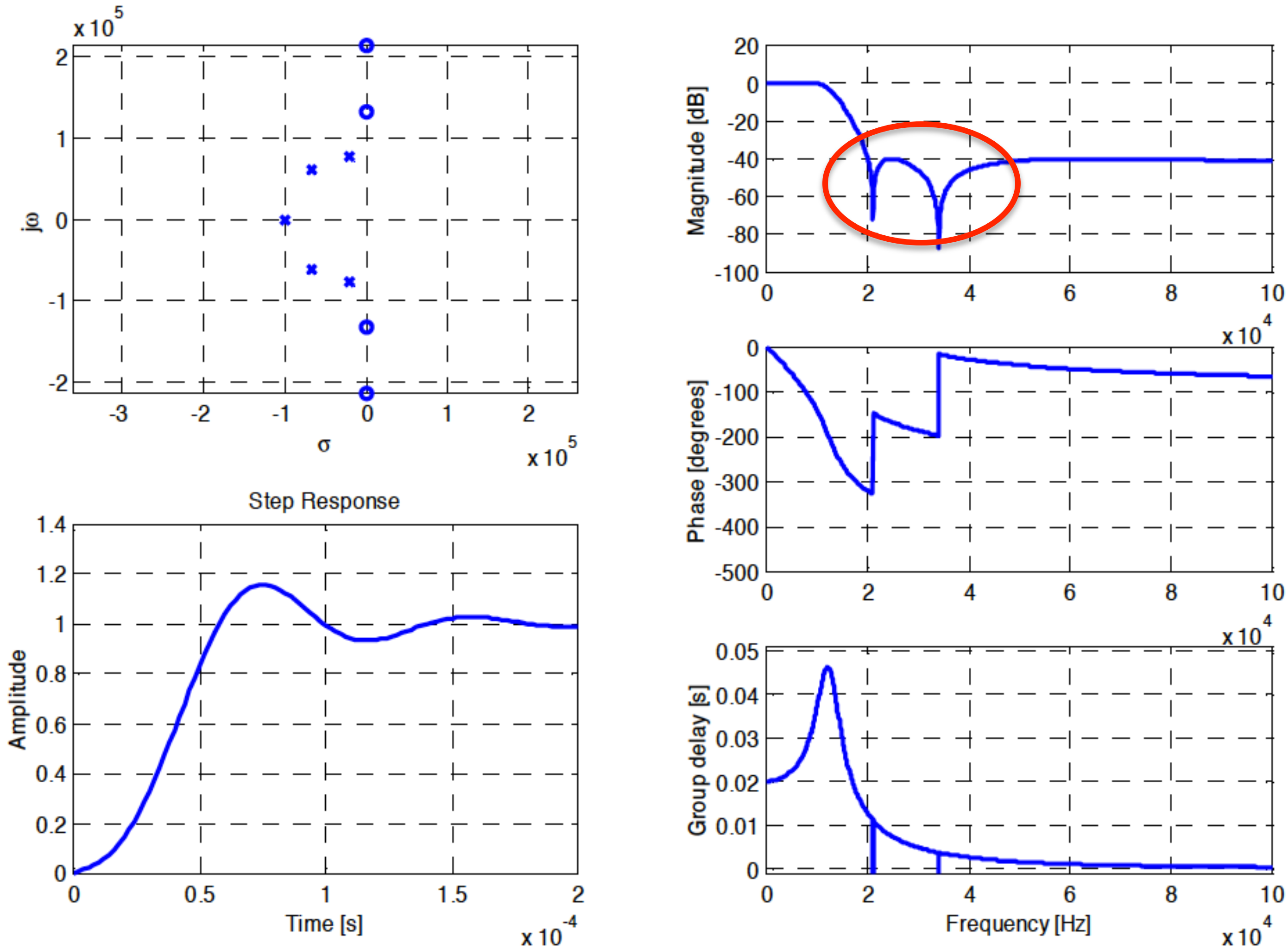
[N, Wp] = cheb1ord(Wp, Ws, Rp, Rs, ‘s’);

Where **Wp**, and **Ws** are the passband and stopband edge frequency in [rad/s]
Rp, and **Rs** are the passband ripple and the stopband attenuation in dB

cheb1ord function returns **N** (minimum filter order) and **Wp** (cutoff frequency)

CHEBYSHEV TYPE II

N = 5; Wn = 2*pi*20e3; Rs = 40 dB



CHEBYSHEV TYPE II

• To design the nth order Chebyshev type II filter use **cheby2** command:

[B, A] = cheby2(N, R, Ws, 'ftype', 's');

Where **N** is filter order, **R** [dB] is peak to peak stopband ripple, **Ws** [rad/s] is stopband edge frequency, **ftype** could be either 'low', 'high', and 'stop.' Argument 's' specify for continuous time, if not specified it is used us digital filter

cheby2 function returns **B** (numerator) and **A** (denominator)

• If filter order is unknown use **cheb2ord** command:

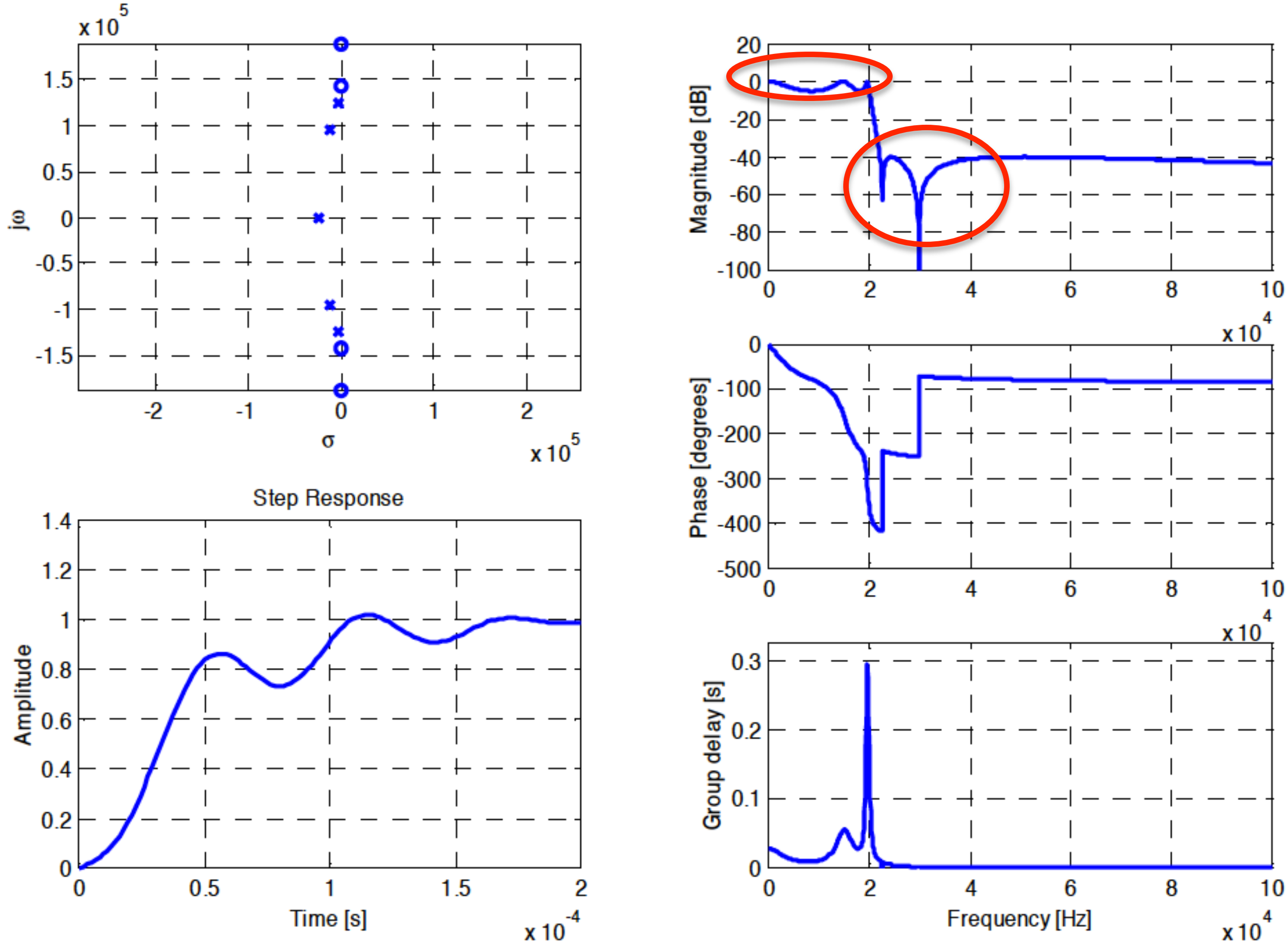
[N, Ws] = cheb2ord(Wp, Ws, Rp, Rs, 's');

Where **Wp**, and **Ws** are the passband and stopband edge frequency in [rad/s]
Rp, and **Rs** are the passband ripple and the stopband attenuation in dB

cheb2ord function returns **N** (minimum filter order) and **Ws** (stopband frequency)

ELLIPTIC

N = 5; Wn = 2*pi*20e3; Rp = 5 dB; Rs = 40 dB



ELLIPTIC

- To design the n^{th} order elliptic filter use **ellip** command:

[B, A] = ellip(N, Rp, Rs, Wp, 'ftype', 's');

Where **N** is filter order, **Rp** and **Rs** [dB] is peak to peak passband and stopband ripple, **Wp** [rad/s] is stopband edge frequency, **ftype** could be either 'low', 'high', and 'stop.' Argument 's' specify for continuous time, if not specified it is used us digital filter

ellip function returns **B** (numerator) and **A** (denominator)

- If filter order is unknown use **ellipord** command:

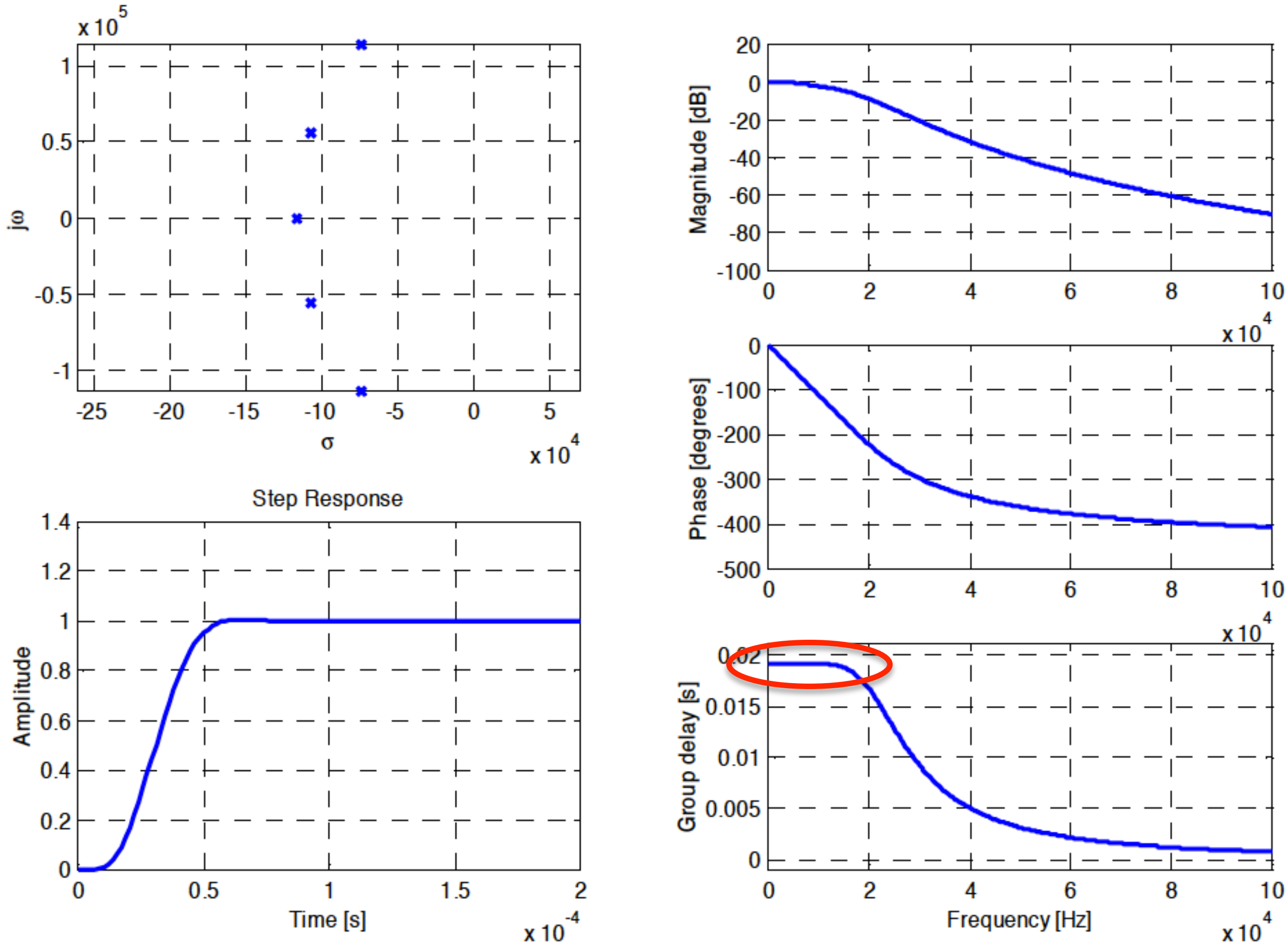
[N, Wp] = ellipord(Wp, Ws, Rp, Rs, 's');

Where **Wp**, and **Ws** are the passband and stopband edge frequency in [rad/s]
Rp, and **Rs** are the passband ripple and the stopband attenuation in dB

ellipord function returns **N** (minimum filter order) and **Wp** (passband frequency)

BESSEL

```
N = 5; Wn = 2*pi*20e3;
```



BESSEL

- To design the n^{th} order bessel filter use **besself** command:

    [B, A] = besself(N, Wo);

Where **N** is filter order, **Wo** [rad/s] is the frequency up to which the group delay is approximately constant

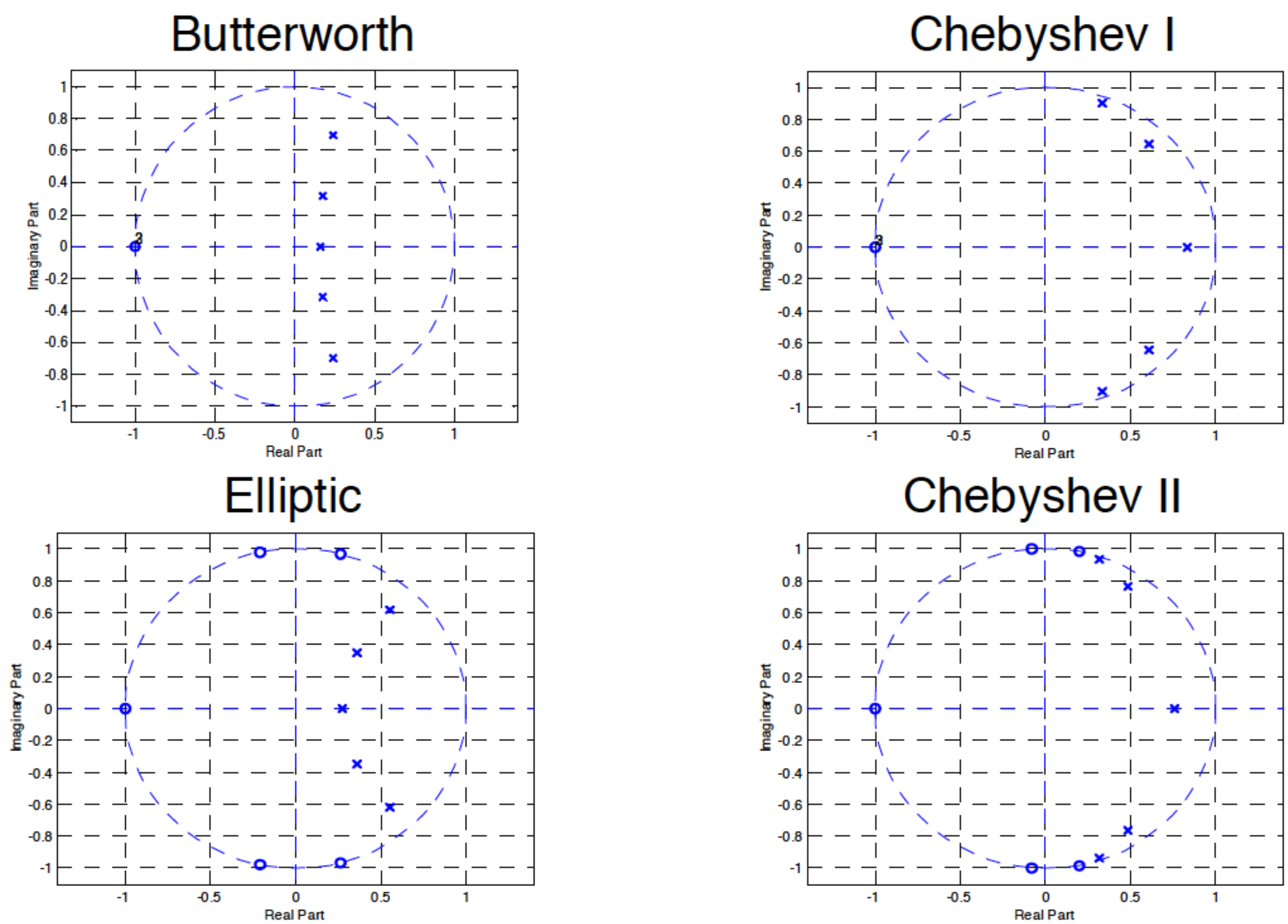
besself function returns **B** (numerator) and **A** (denominator)

- **besself** function can only design analog filters

DISCRETE-TIME FILTERS

- Use same command without 's' option
- Frequency are normalized to $F_s/2$
  - $Wn = 1 \Rightarrow f = F_s/2$

Z-PLANE REPRESENTATION

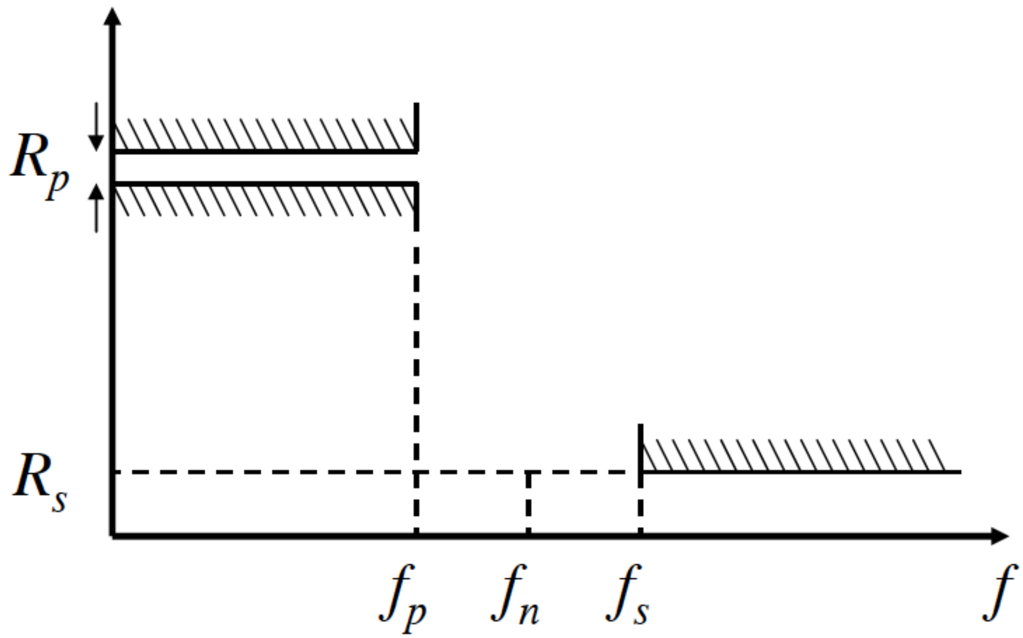


SUMMARY

FILTER TYPE	GAIN ROLLOFF	PHASE LINEARITY
Butterworth	Poor	Better
Chebyshev	Better	Poor
Elliptic	Best	Worst
Bessel	Worst	Best

DESIGN EXAMPLE

- Specs: design a lowpass filter with a passband of 20 kHz, no more than 0.2 dB of passband ripple, and with a minimum attenuation of 60 dB for frequencies above 40 kHz.



DESIGN EXAMPLE

- State the spec:

```
Wp = 2*pi*20e3;   Ws = 2*pi*40e3;
Rp  = 0.2;  Rs = 60;
```

- Choose filter type. Find order and cutoff frequency:

```
[N, Wn] = cheb1ord(Wp, Ws, Rp, Rs, 's')
    >> N = 7
       Wn = 1.2566+005
```

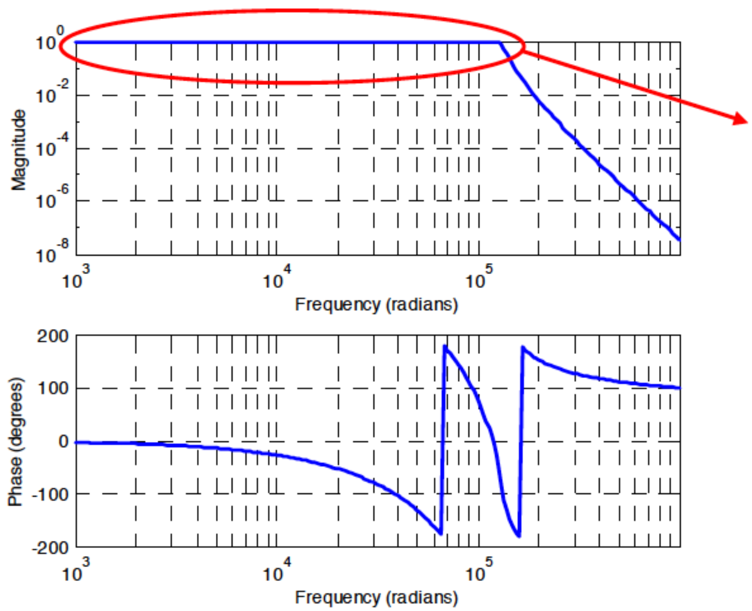
- Find numerator and denominator of filter transfer function:

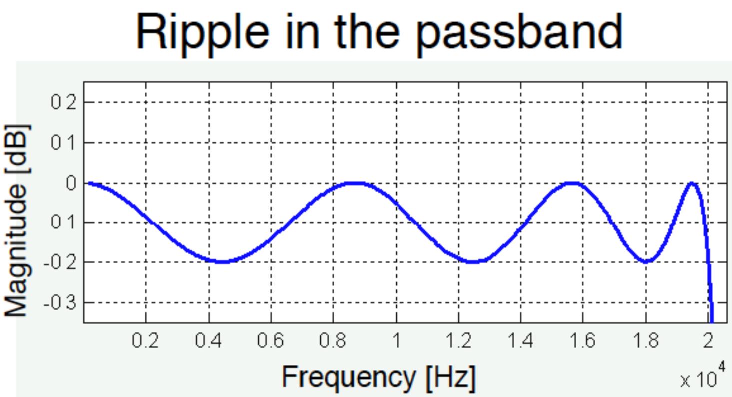
```
[num, den] = cheby1(N, Rp, Wn, 's');
tf(num, den)

    >> Transfer function:
                                              3.562e34
    -----------------------------------------------------------------------------------------------------------
    s^7 + 1.831e05 s^6 + 4.441e10 s^5 + 5.068e15 s^4 + 5.422e20 s^3 + 3.502e25 s^2 + 1.637e30 s + 3.562e34
```

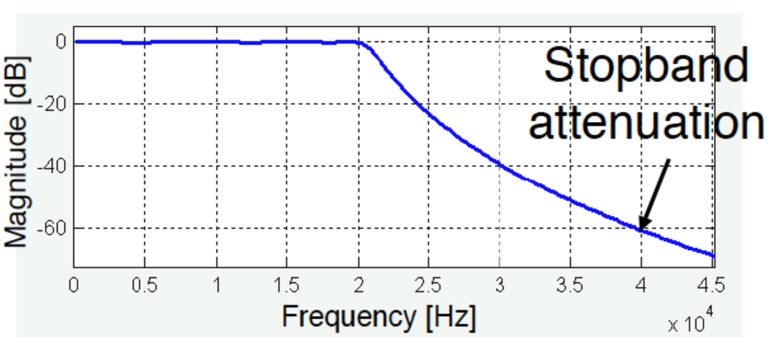
DESIGN EXAMPLE

- To plot magnitude and phase:

```
[H, w] = freqs(num, den);
semilogx(w, 20*log10(abs(H)));
semilogx(w, 180/pi*phase(H));
```







DESIGN EXAMPLE

- To plot step and group delay response:

```
T = tf(num, den);
step(T)
gd = -diff(phase(H))./diff(w);
plot(w(1:length(w)-1), gd);
```