Kuwait University
College of Engineering and Petroleum

جامعة الكويت
KUWAIT UNIVERSITY

ME319 MECHATRONICS

Part III: The Senses – Sensors and Signals

Lecture 1: Signal Conditioning and Filtering

Spring 2021

Ali AlSaibie

Lesson Objectives

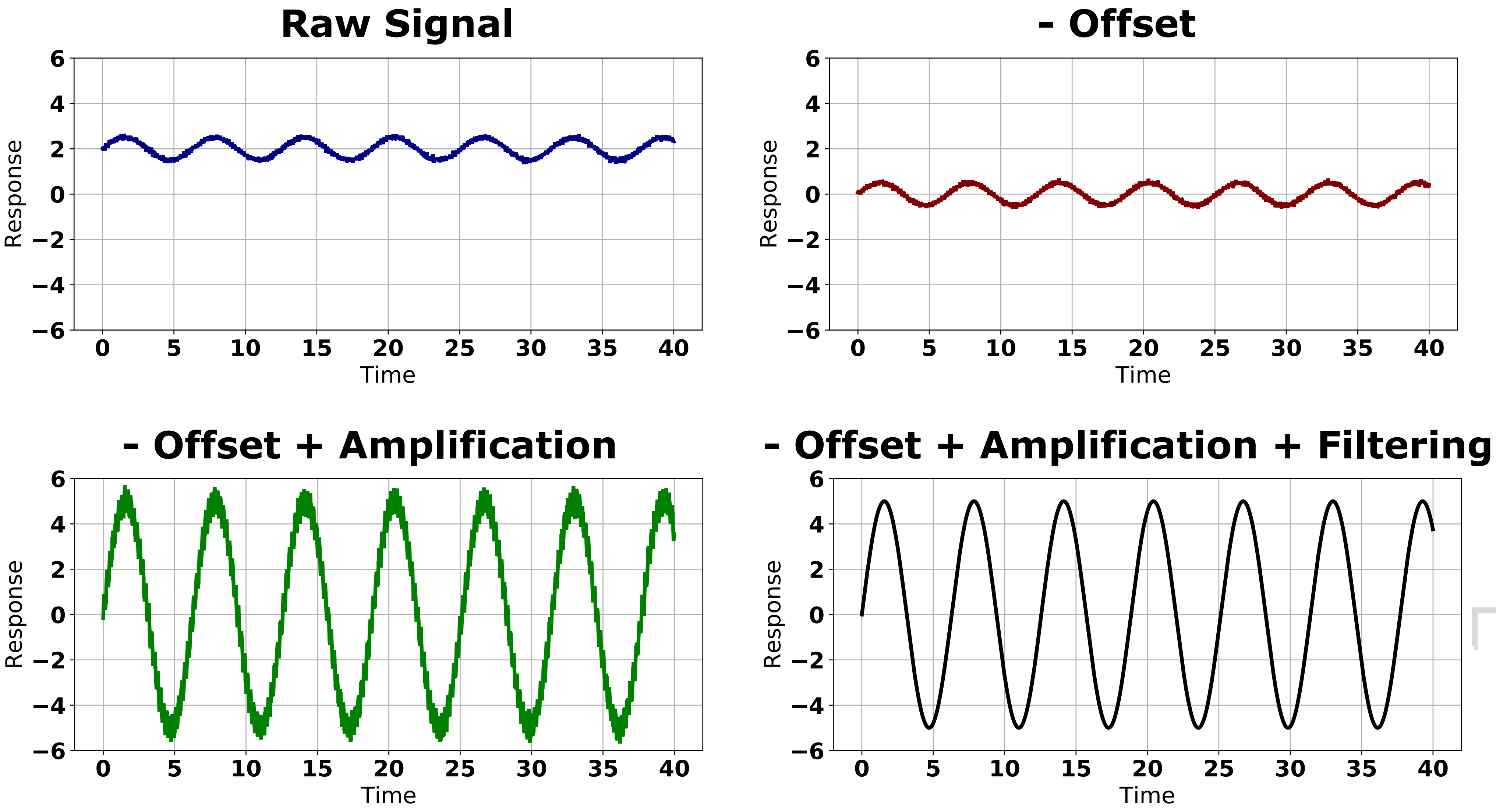
- Review the basics elements of signal conditioning
- Discuss passive filtering techniques
- Discuss digital filtering techniques

جامعة الكويت
KUWAIT UNIVERSITY

- An analog signal, coming from a sensor for example, can have an
  - Offset, or bias: A DC shift from the mean or actual value
  - Poor range: A small voltage range, reduces reading precision
  - Noise: Signal components not of interest
- Signal Conditioning is the process of eliminating the above issues
  - A signal is offset, then
  - The signal is scaled to maximize measurement resolution, then
  - Signal noise is removed through noise **filtering** techniques.

جامعة الكويت
KUWAIT UNIVERSITY

- The following figure illustrates the steps involved in signal conditioning.



جامعة الكويت
KUWAIT UNIVERSITY

Analog or Digital Signal Conditioning?

- While modern MCUs/DSPs can perform a wide range of signal conditioning operations,
- There is always a good case for applying signal conditioning in circuit (Analog)
- A few of the reasons why:
  - Lower CPU overhead
  - Sampling limitations on the digital side
  - Capturing a wider range of the “useful” signal
- A few of the limitations of analog signal processing:
  - Varying signal dynamics
  - Signal modeling uncertainty
  - Cost/Complexity/Difficulty

Offset Removal

- Consider the following voltage signal:

$$V(t) = 2 + 3.3\sin(\omega t)$$

- If this signal is fed into an ADC, which can only handle $V_{range} = [-3.3V, 3.3V]$
- The above signal, will clearly exceed the MCU input range

- We can amplify (scale down) the signal, to limit the maximum value to 3.3V
  - But we will loose signal resolution on the ADC side

- Instead, we can remove the DC offset: $V_{DC} = 2$, from the signal, to achieve

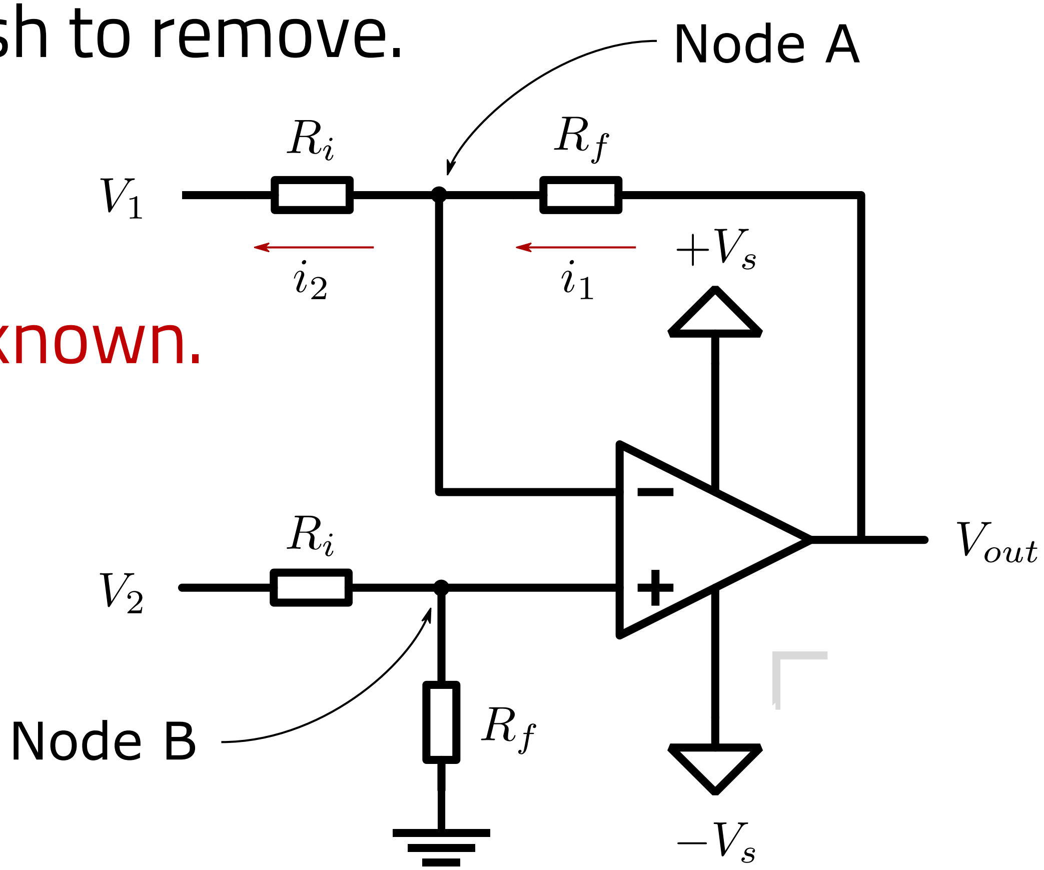
$$V(t)_{-offset} = 3.3\sin(\omega t)$$

جامعة الكويت
KUWAIT UNIVERSITY

- Offset removal can be achieved using a difference op-amp configuration

$$V_{out} = (V_2 - V_1)\frac{R_f}{R_i}$$

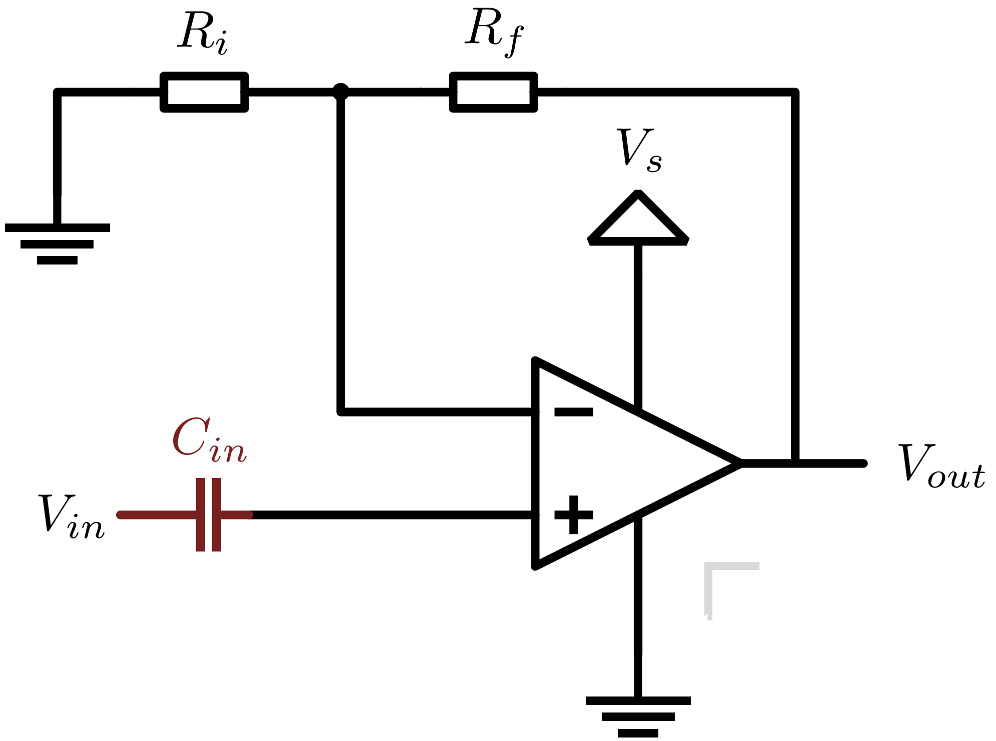
- Where $V_1 = V_{DC}$, the DC offset we wish to remove.
- Limitation:
  - The DC offset must be accurately known.



- If the DC offset is unknown, or varying, and we wish to completely remove the DC component, we can add a capacitor in series to the op-amp input
  - Removing DC components may not always be desired/required

$$V_{out} = (V_{in} - V_{DC})\left(1 + \frac{R_f}{R_i}\right)$$

- We can remove DC offset and amplify



- Often, there are sensors that output values in the mV range.
- If the MCU ADC resolution is $2mV$ for example, and the incoming signal range is $[0,10mV]$, there isn't much resolution in the ADC converted signal.
  - The digital value is practically useless.
  - The precision is $\pm 1mV$, a 20% uncertainty of range.
- So, we try to amplify the signal to the full range of the ADC input.
- As discussed in the Op-Amps section.
- We amplify, linearly, the $[0,10mV]$ range signal to $[0,3.3V]$ for example.

جامعة الكويت
KUWAIT UNIVERSITY

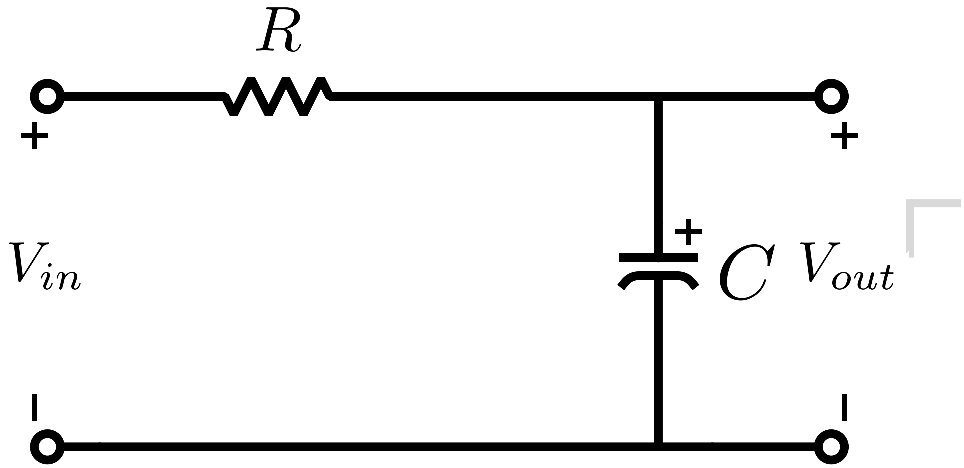
- To discuss filtering, it is important to review the concepts of frequency response.
- Consider the following low-pass filter circuit.

$$V_{in}(t) = Ri(t) + \frac{1}{C}\int i(t)dt \Rightarrow V_{in}(s) = RI(s) + \frac{1}{Cs}I(s)$$

$$V_{out}(t) = \frac{1}{C}\int i(t)dt \Rightarrow V_{out}(s) = \frac{1}{Cs}I(s)$$

The RC Filter transfer function:

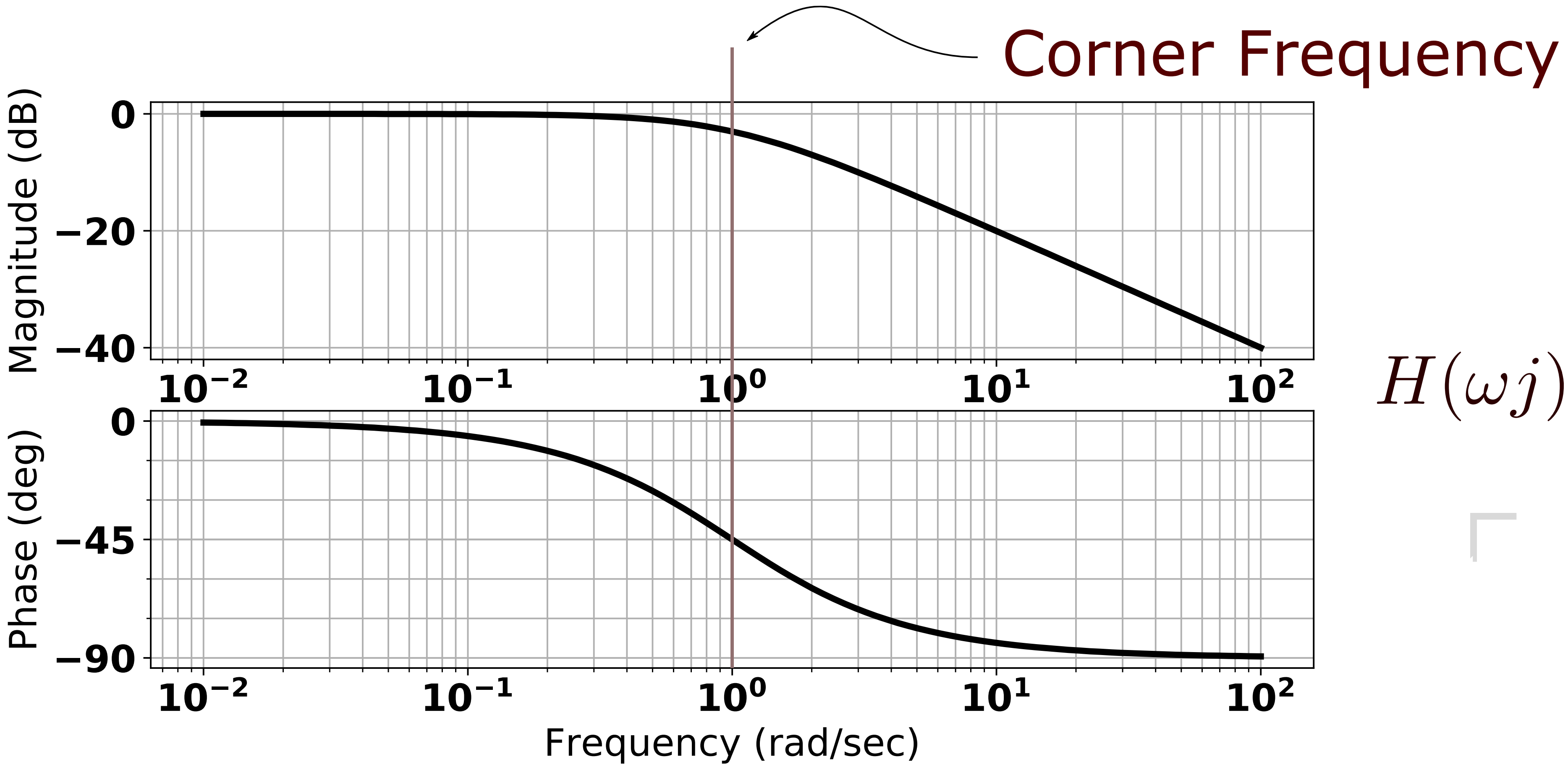
$$H(s) = \frac{V_{out}}{V_{in}} = \frac{1}{RCs + 1}$$



- At steady-state $s \to \omega j$, the RC Filter transfer function becomes

$H(\omega j) = \frac{V_{out}}{V_{in}} = \frac{1}{RC\omega j+1}$. This is a first-order system. With a corner frequency of $\omega_c = \frac{1}{RC}$

- If we plot the Magnitude $|H(\omega j)|$ and Phase response $\angle H(\omega j)$, for varying ω, we get the Bode Plot



$$H(\omega j) = \frac{1}{\omega j+1}$$

- A bode plot is the pair of magnitude response and phase response plots.
  - The frequency is plotted on a log-scale
  - The magnitude response is either plotted on a log scale or decibels (dB)
    - $20dB = 20\log(10), -20dB = 20\log(0.1)$
  - Phase is plotted in degrees

- To convert from dB to decimal ratio
  - $A[ratio] = 10^{\frac{A[dB]}{20}}$
- To convert from decimal ratio to dB
  - $A[dB] = 20\log_{10}A[ratio]$

جامعة الكويت
KUWAIT UNIVERSITY

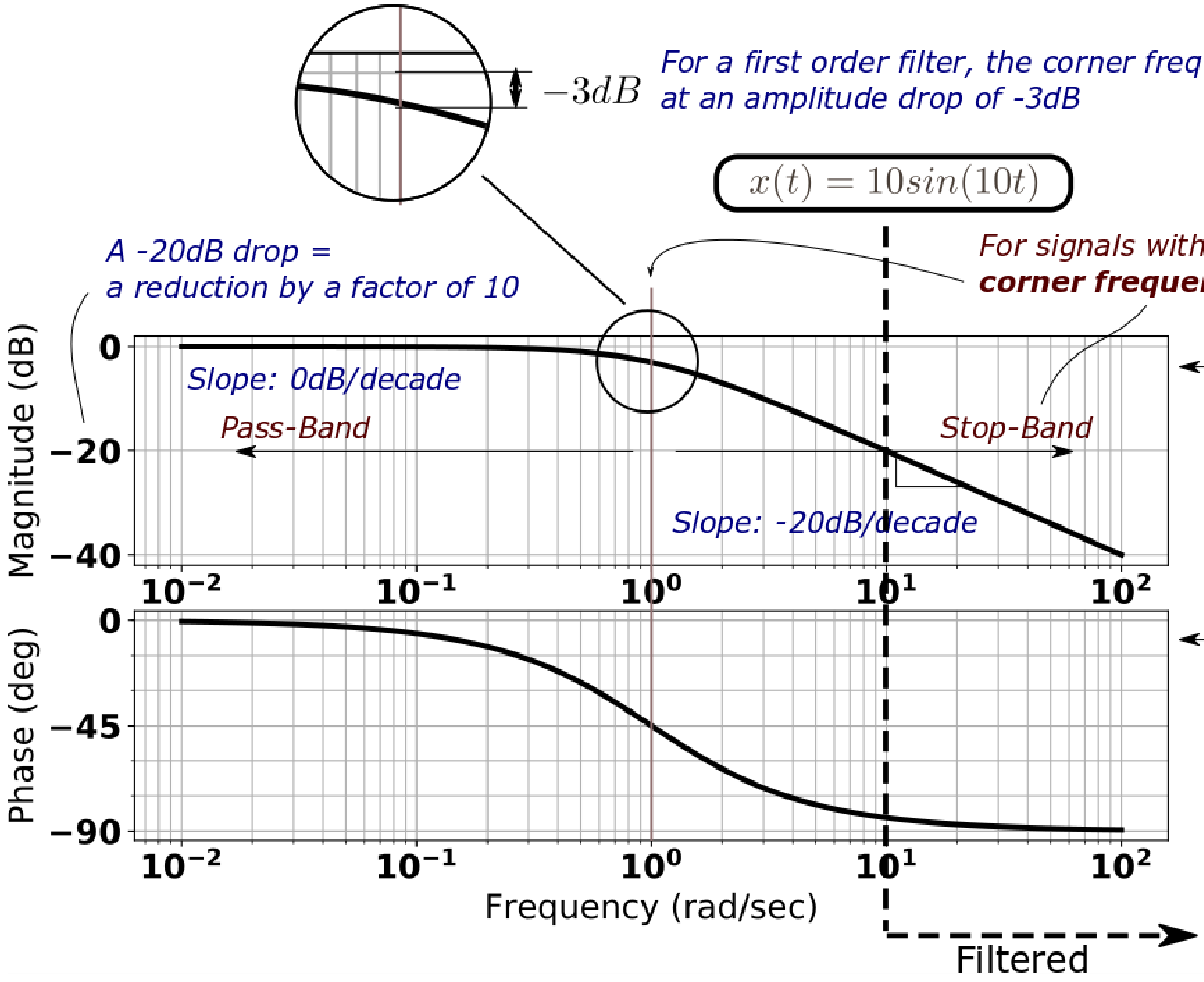
Low Pass Filter (1st Order)

$$H(\omega j) = \frac{Y}{X} = \frac{1}{\omega j + 1}$$

For a first order filter, the corner frequency is defined at an amplitude drop of -3dB

$$x(t) = 10sin(10t)$$

*For signals with a frequency higher than the cut-off, or **corner frequency**, the signal amplitude is **attentuated***



Magnitude Response
Shows ratio of output to input amplitude for a specific excitation frequency

Phase Response
Shows time shift between input and output for a specific excitation frequency

Negative phase angle means the output lags, positive phase angle means the output leads the input

$$y(t) = 0.99sin(10t - 1.47)$$

- Using Fourier Transform:
  - Continuous: $x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)e^{j\omega t}d\omega$
  - Discrete: $x[k] = \frac{1}{N}\sum_{k=0}^{N-1} X_k e^{i2\pi kn/N}$
- Fourier Analysis is outside the scope of this course, but the main idea is
- *A signal is a weighted sum of single frequency components.*
- Many signals we deal with can be approximated to have a small finite number of frequency components.
- *Example:* $x(t) = 10\sin(1t+\pi) + 20\cos\left(100t + \frac{\pi}{2}\right) + 0.5\sin(1000t)$
- *Is a **weighted** sum of **three** frequency components*

  $@\omega = 1,\ 100\ \&\ 1000$

- When filtering a signal, each frequency component gets amplified and shifted independently, the output signal is the sum of the filtered frequency specific components

$$x(t) = 10sin(1t + \pi) + 20cos(100t + \tfrac{\pi}{2}) + 0.5sin(1000t) \longrightarrow \boxed{H(\omega j) = \tfrac{1}{\omega j + 1}} \longrightarrow y(t) = ?$$

$$\Downarrow$$

$$x(t)_1 = 10sin(1t + \pi) \longrightarrow \boxed{H(\omega j) = \tfrac{1}{\omega j + 1}} \longrightarrow y(t)_1 = 7.07sin(1t + \pi - 0.83)$$

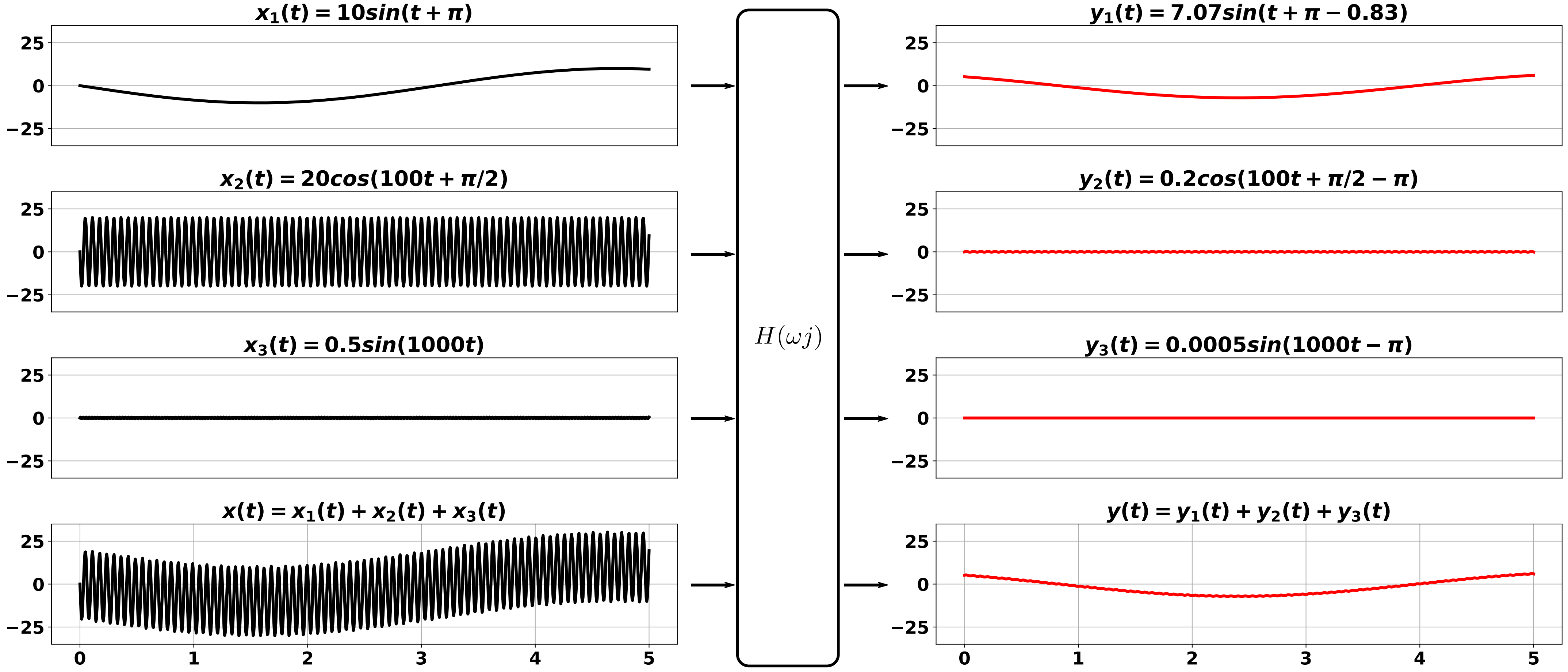
$$x(t)_2 = 20cos(100t + \tfrac{\pi}{2}) \longrightarrow \boxed{H(\omega j) = \tfrac{1}{\omega j + 1}} \longrightarrow y(t)_2 = 0.2cos(100t + \tfrac{\pi}{2} - \pi)$$

$$x(t)_3 = 10sin(1000t) \longrightarrow \boxed{H(\omega j) = \tfrac{1}{\omega j + 1}} \longrightarrow y(t)_3 = 0.0005sin(1000t - \pi)$$

$$y(t) = 7.07sin(1t + \pi - 0.83) + 0.2cos(100t + \tfrac{\pi}{2} - \pi) + 0.0005sin(1000t - \pi)$$

جامعة الكويت
KUWAIT UNIVERSITY

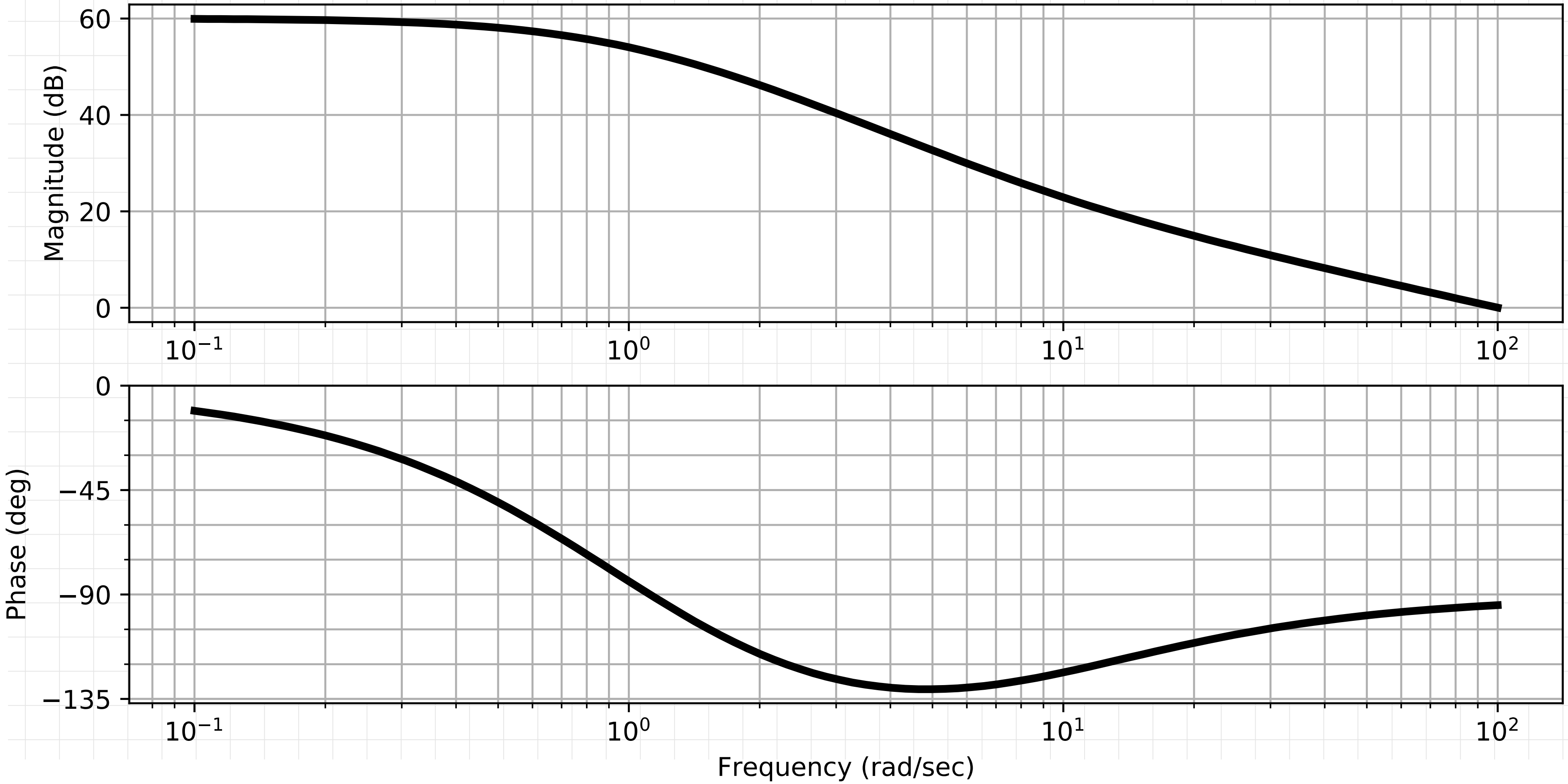
Filtering a Signal



جامعة الكويت
KUWAIT UNIVERSITY

Derive the output signal function given the following Bode Plot and the following input signal:

$$u_{in} = 10\sin(20t + \pi) + 100\cos(100t)$$



Example

Perform the following conversions

a. -20dB to decimal

b. 100 to dB

c. 0.01 to dB

KUWAIT UNIVERSITY

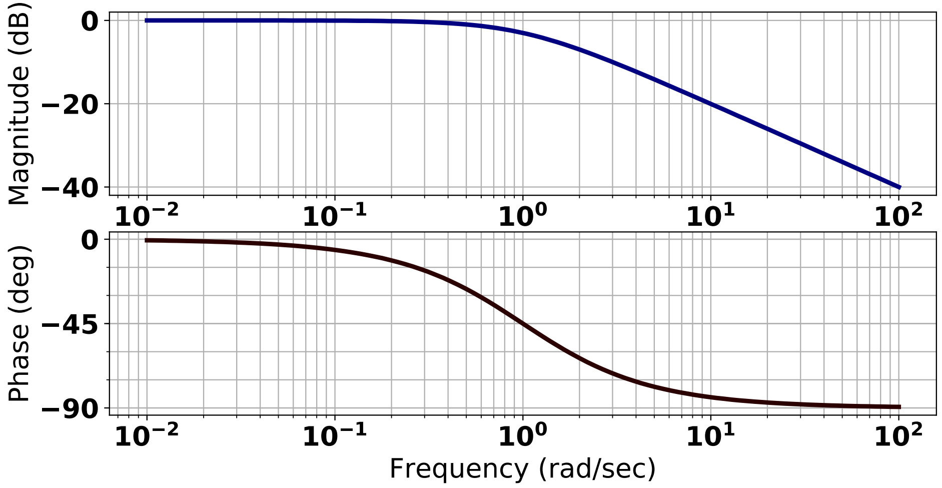
Filters

- Filters are categorized into the following

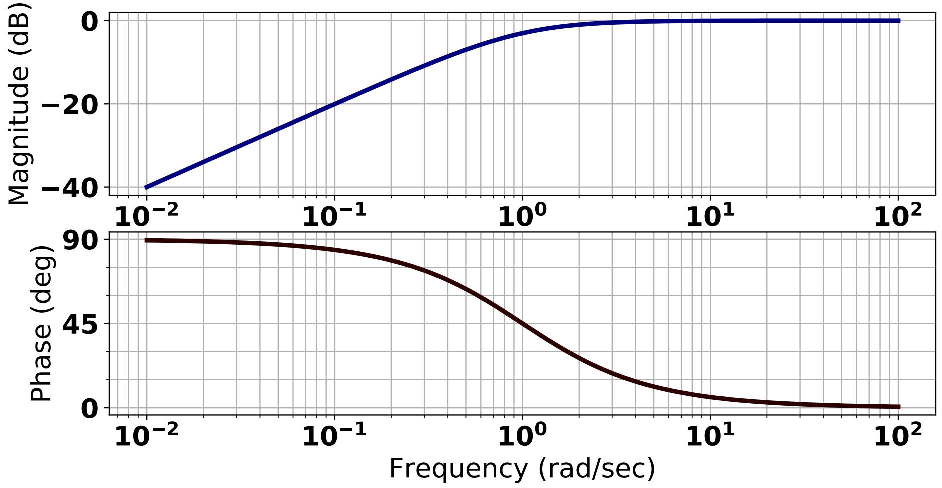
1. Low Pass Filter
    - *Low frequency components are preserved, high frequency ones blocked*
2. High Pass Filter
    - *High frequency components are preserved, low frequency ones blocked*
3. Band Pass Filter
    - *A range of frequency components preserved, higher and lower ones blocked*
4. Notch Filter
    - *A narrow range of frequency components are **preserved***
    - *Or, a narrow range of frequency components are **removed**.*

جامعة الكويت
KUWAIT UNIVERSITY

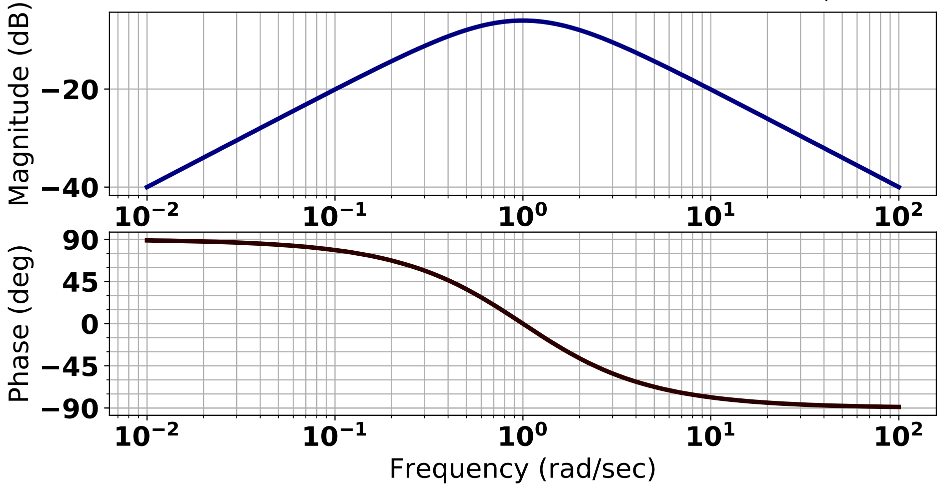
Low Pass Filter: $H(\omega j) = \frac{1}{RCs+1}$



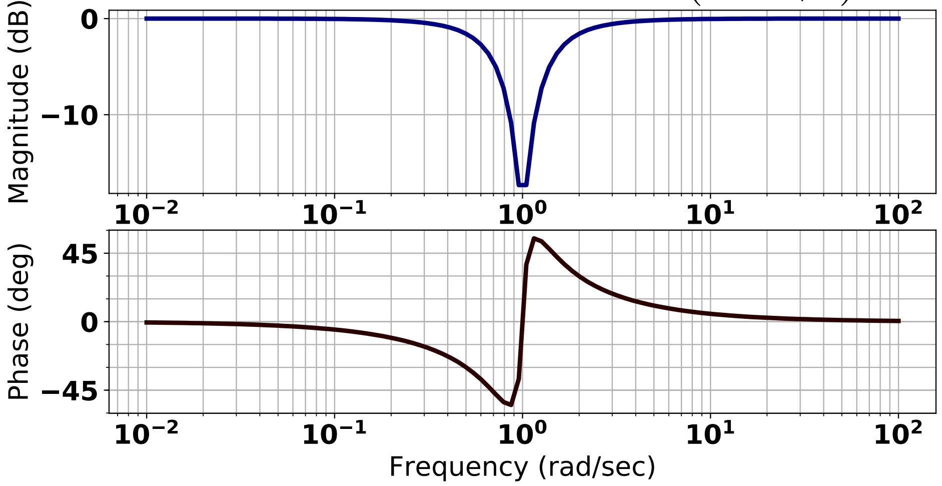
High Pass Filter: $H(\omega j) = \frac{s}{1/RCs+1}$



Band Pass Filter: $H(\omega j) = \frac{1}{RCs+1}\frac{s}{1/RCs+1}$



Notch Filter: $H(\omega j) = \frac{(RCs^2+0.1s+1)}{(RCs+1)^2}$



جامعة الكويت
KUWAIT UNIVERSITY

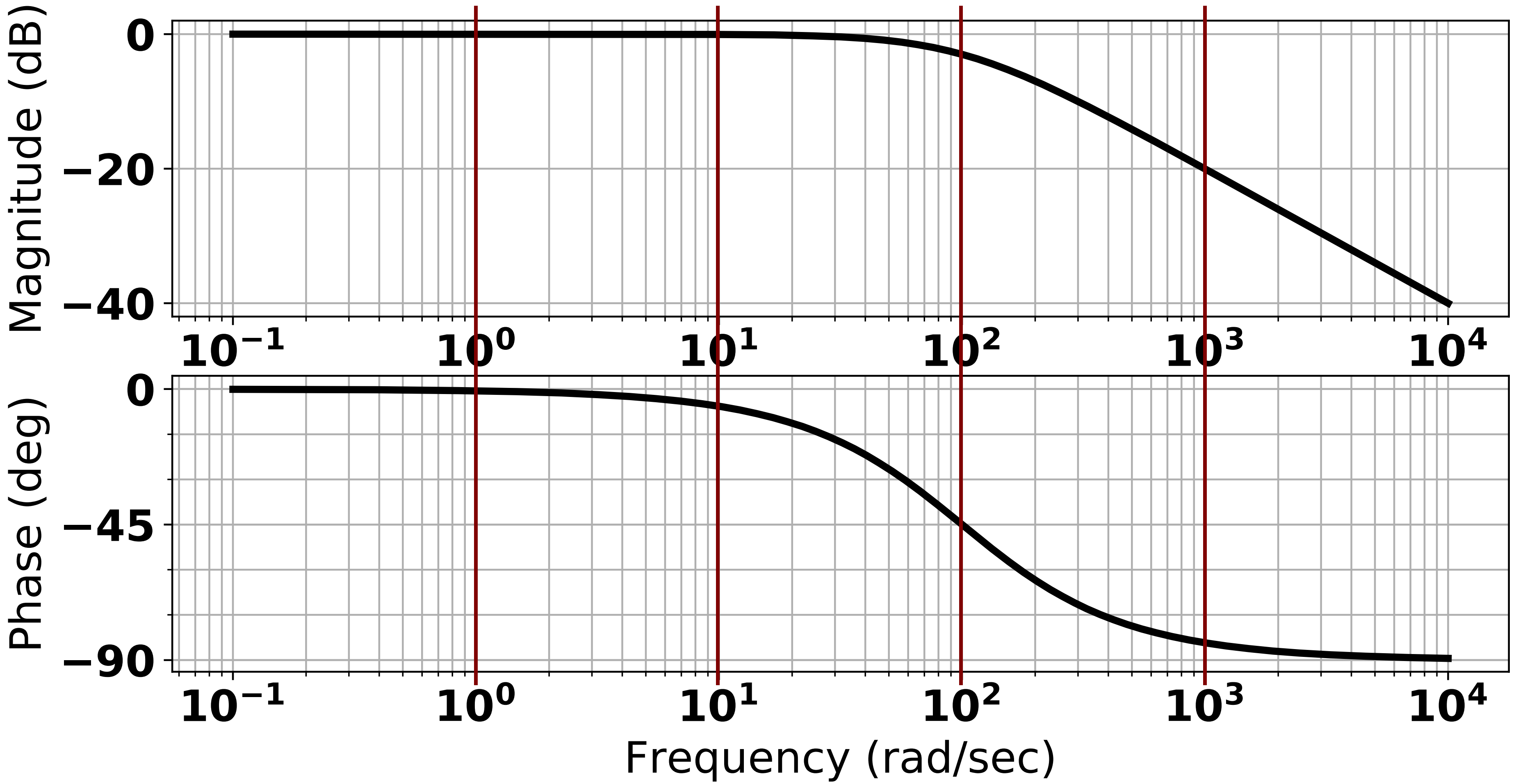
Simple First Order Low Pass Filter

- A first order filter is simple to implement
- It works well for noise that is at a much higher frequency than the signals'
- As the noise frequency approaches the signal frequency, it becomes hard to implement a low pass filter successfully.
  - The phase shift of a low pass filter starts early
    - *The required signal will be delayed*
  - The attenuation slope is slow (slow rollover rate)
    - *The noise can't be attenuated well*
- Higher order filters, with a sharper attenuation slop and sharp phase delay curve can be used in such cases.

- Consider the first-order low pass filter given by the Bode Plot: $\omega_c = 100 rad/s$

$$u_{1\,in} = 10\sin(1t) + 10\sin(1000t) \Rightarrow u_{1\,out} = 9.99\sin(1t - 0.01) + 0.99\sin(1000t - 1.47)$$

$$u_{2\,in} = 10\sin(10t) + 10\sin(100t) \Rightarrow u_{2\,out} = 9.95\sin(1t - 0.1) + 7\sin(100t - 0.78)$$



Higher Order Filters

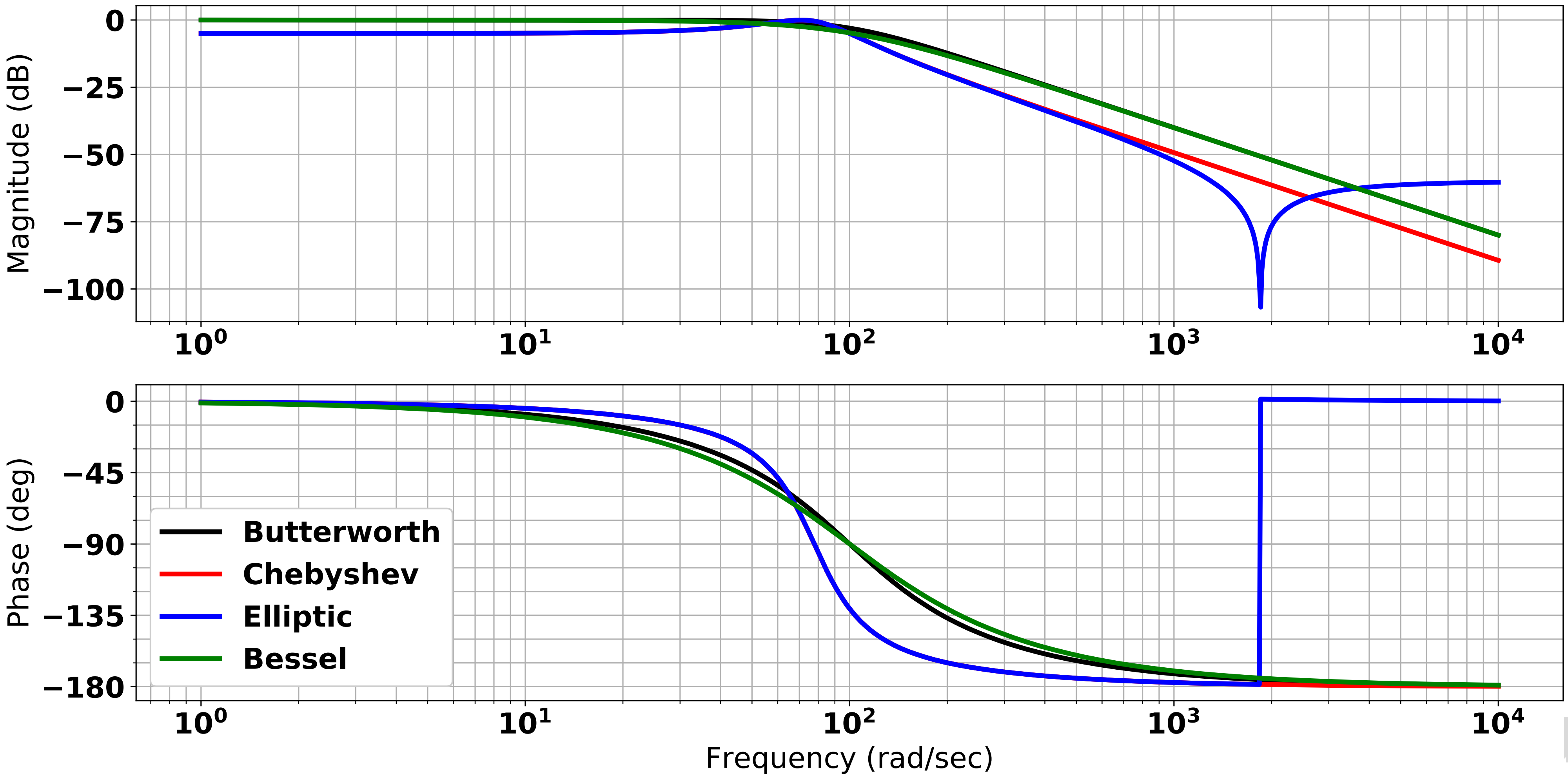
- The four classic analog filters are (comments are general guides, not always accurate)

1. **Butterworth**
   - Flat pass-band, poor attenuation rate, good phase response
2. **Chebyshev**
   - Some pass-band ripple, good attenuation rate, good phase response
   - For same order as Butterworth, sharper pass to stop band transition
3. **Elliptic**
   - Some pass and stop band ripple, but best roll off rate (sharpest)
4. **Bessel**
   - Poor roll off rate, but good phase response of all.

- The above filters don't have a specific order. For each, the order is chosen.
- For Chebyshev and Elliptic, the ripple tolerance must be specified.

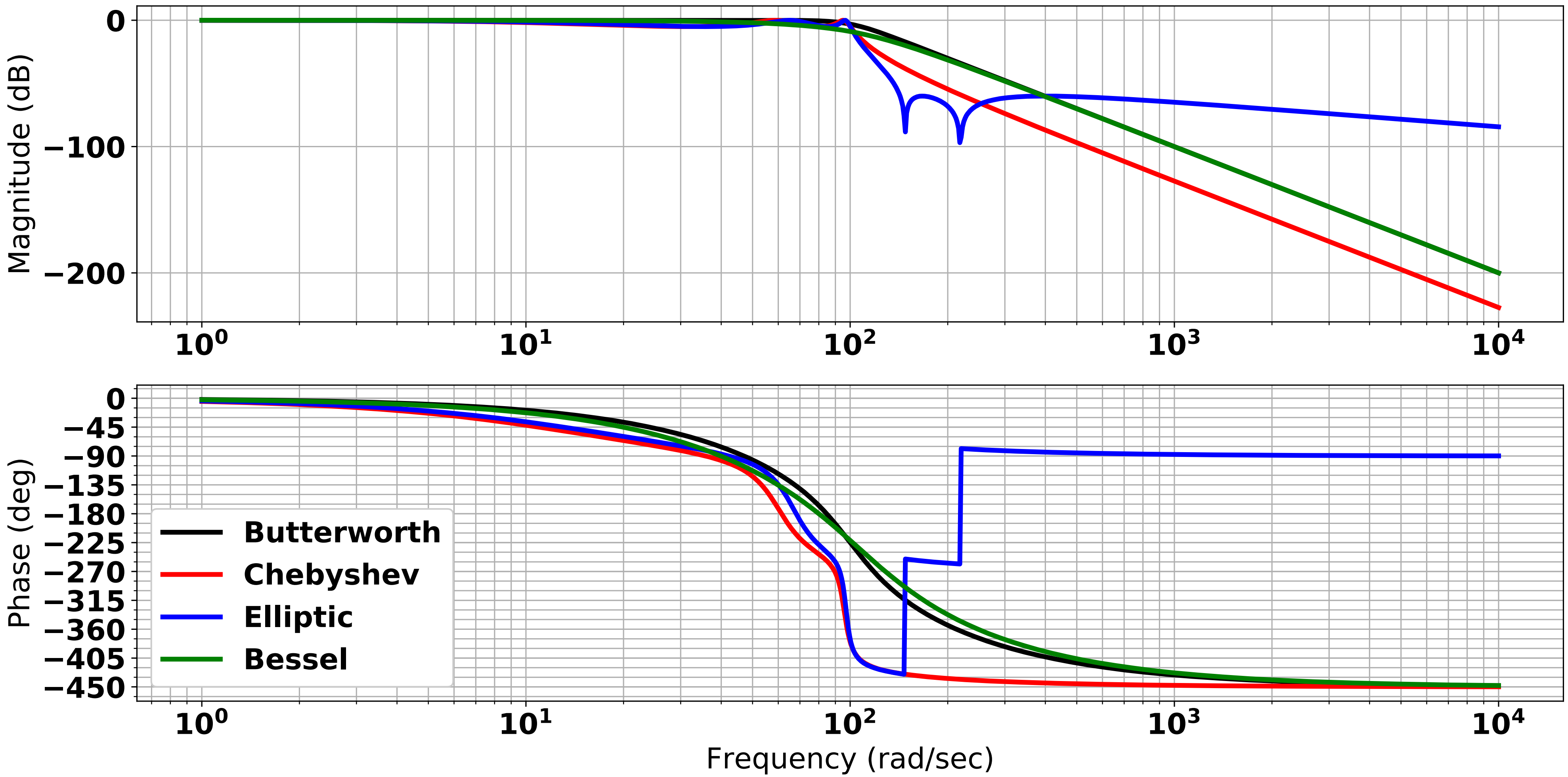
جامعة الكويت
KUWAIT UNIVERSITY

Analog Low Pass Filters – 2nd Order



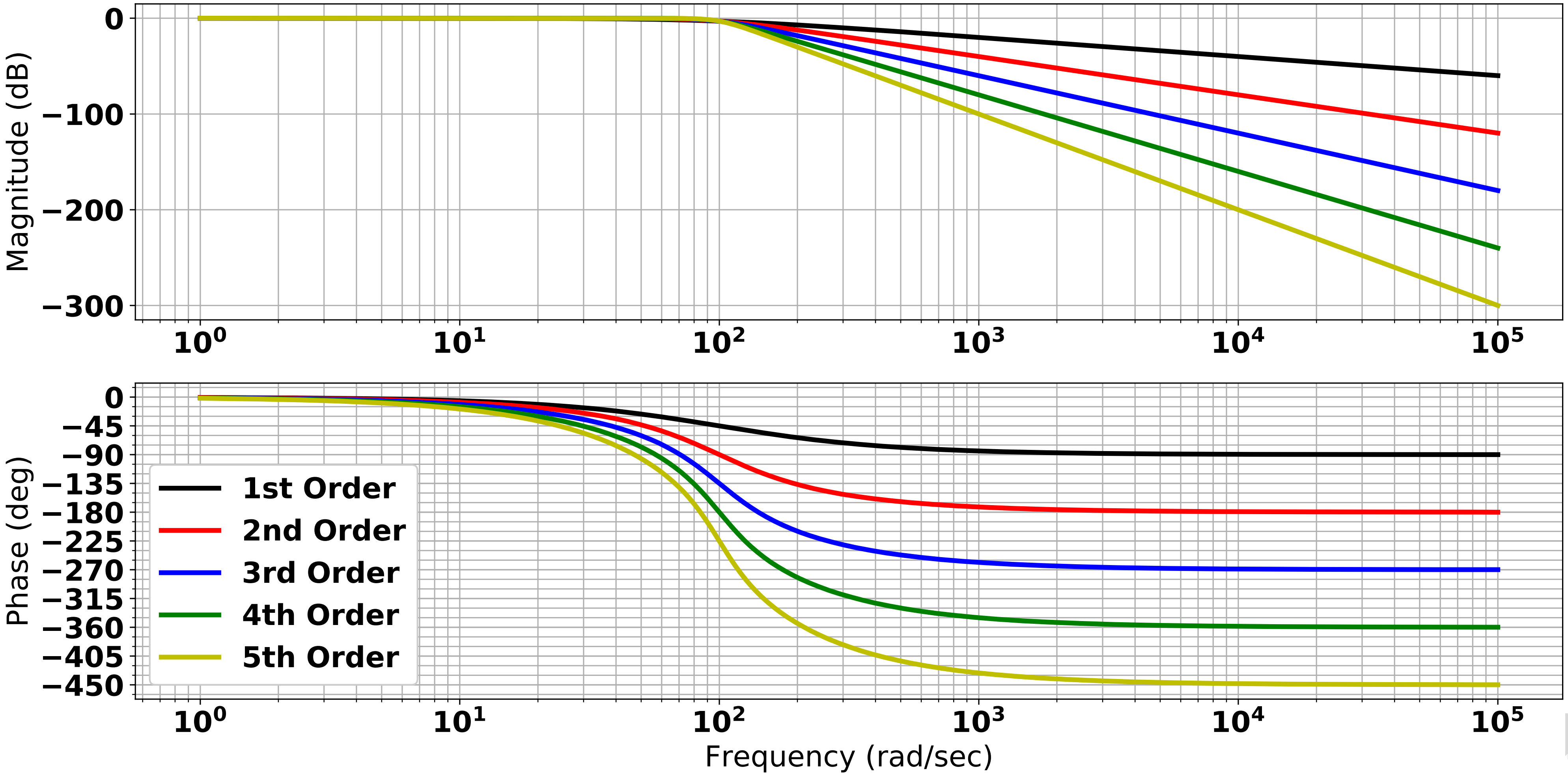
جامعة الكويت
KUWAIT UNIVERSITY

Analog Low Pass Filters – 5th Order



جامعة الكويت
KUWAIT UNIVERSITY

Butterworth Low Pass Filter – Varying Orders



Analog Filters

- A filter can be constructed using **passive electronic** elements
  - Using resisters, capacitors and inductors
  - An RC Filter is a passive filter
- A filter can be constructed using **active electronic** elements
  - Using op-amps and other components that **require energy supply**.

Digital Filters

- Filtering can be done on the software side inside a microcontroller
  - Given sufficient sampling and signal resolution, a software filter can emulate the effect of an electronic (in-circuit) filter.

جامعة الكويت
KUWAIT UNIVERSITY

- The same analog filters (and more), can be implemented in software as digital filters.
- With digital implementation, the sampling time or the simulation time-step, affects the performance of the filter.
- We can design a filter in the continuous domain and convert it into discrete form. Then from the discrete filter transfer function we can get a difference equation to implement in software

$$\underbrace{H(s) = \frac{Y(s)}{U(s)}}_{Continous\, T.F.} \to^{T} \underbrace{H(z) = \frac{Y(z)}{U(z)}}_{Discrete\, T.F.} \to \underbrace{y[k] = \cdots}_{Difference\, Equation}$$

KUWAIT UNIVERSITY
جامعة الكويت

- An analog filter can be expressed via a continuous transfer function $H(s)$
- Digital filters can be expressed with a discrete transfer function $H(z)$
- s is the continuous domain complex variable, z is the discrete domain variable
- $z = e^{sT}$, where T is the sampling time, or integration timestep.
- z can be approximated via bilinear transform: $z = \frac{1+sT/2}{1-sT/2} \rightarrow s = \frac{2}{T}\frac{1-z}{1+z}$
- In MATLAB, given a continuous domain transfer function
  - Can discretize via *c2d()*: $G(s) = \frac{1}{s/100+1} \rightarrow^{c2d:T=0.01} \quad G(z) = \frac{0.632}{z-0.367}$

```
s = tf('s')
Gs = 1 / (s/100 + 1);
T = 0.01;
Gz = c2d(Gs,T)
```

جامعة الكويت
KUWAIT UNIVERSITY

Discrete T.F. to Difference Equation

- A discrete transfer function can be conveniently converted into a difference equation. (Analogous to converting a continuous T.F. to a differential eq.)
- A difference equation can be directly implemented in software.
- Given

$$H(z)=\frac{Y(z)}{U(z)}=\frac{az+b}{z^2+dz+e}=\frac{z^{-2}}{z^{-2}}\frac{az+b}{z^2+dz+e}=\frac{az^{-1}+bz^{-2}}{1+dz^{-1}+ez^{-2}}$$

$$(1+dz^{-1}+ez^{-2})Y(z)=(az^{-1}+bz^{-2})U(z)$$

$$y[k]+dy[k-1]+ey[k-2]=au[k-1]+bu[k-2]$$

$$\boldsymbol{y[k]=-dy[k-1]-ey[k-2]+au[k-1]+bu[k-2]}$$

جامعة الكويت
KUWAIT UNIVERSITY

Example

Given the following discrete transfer functions, derive the difference equation.

a. $H(z) = \frac{0.8}{z-0.5}$

b. $H(z) = \frac{0.2452z+0.254}{z-0.59}$

جامعة الكويت
KUWAIT UNIVERSITY

Software Implementation of a Difference Equation

- Implement: $y[k] = -cy[k-1] + au[k] + bu[k-1]$, in software

```
#include <iostream>
#include <cmath>
int main(){
    /* Create some input */
    float u[100];
    for (int k = 0; k<100; k++){ u[k] = sin(2*3.14*k/100); }
    /* Apply the filter H(z) = (a+bz^-1) / (1+cz^-1)
     * Apply the following difference equation
     * y[k] = a*u[k] + b*u[k-1] - c * y[k-1]
     */
    float y[100] ={0}; /* initialize to zero */
    int idx = 1; /* start from to reference idx - 1 */
    const int a=1, b=.5, c=.1;
    while(idx++ < 100){
        y[idx] = a*u[idx] + b*u[idx-1] - c * y[idx - 1];
        std::cout <<  y[idx] << std::endl;
    }
    return 0;
}
```

جامعة الكويت
KUWAIT UNIVERSITY