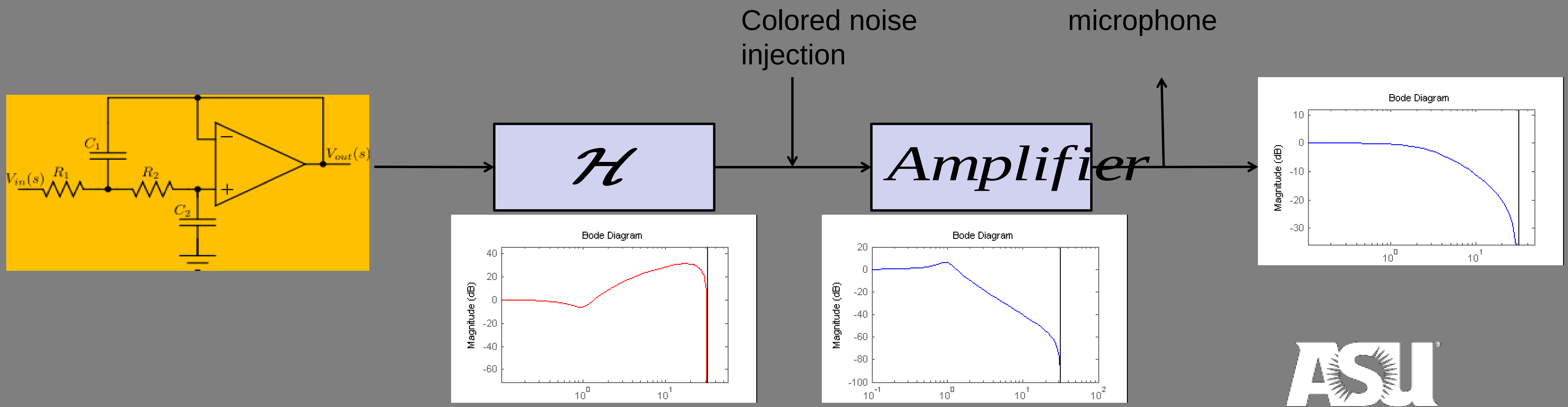
EEE304

Week 2: Filtering



ARIZONA STATE UNIVERSITY

EEE304

Lecture 2.1: Filtering Generalities

Arizona State University

LTI Filters: Generalites

$$y = \mathcal{H}[x] = h * x$$

$$y(t) = \int_{-\infty}^{\infty} h(t-\tau)x(\tau)d\tau; \quad Y(s) = H(s)X(s)$$

$$y(n) = \sum_{k} h(n-k)x(k); \quad Y(z) = H(z)X(z)$$

- Recalling the basic expressions for the response of LTI systems

$$y = \mathcal{H}[x]; \quad Y(s) = H(s)X(s); \quad Y(z) = H(z)X(z)$$

we obtain the following responses to complex exponentials

$$x(t) = e^{j\omega t} \Rightarrow y(t) = H(j\omega)e^{j\omega t}$$

$$x(n) = e^{j\Omega n} \Rightarrow y(n) = H(e^{j\Omega})e^{j\Omega n}$$

$\leftarrow \quad \Omega = \frac{2\pi}{N}$ for DT periodic

- In other words, LTI systems separate parts of signals based on their frequency content, i.e., they are frequency discriminators.

- Filtering, in its narrow interpretation, is exactly this processing of signals whereby different frequencies are multiplied with different gains

LTI Filters: Types

$$y = \mathcal{H}[x] = h * x$$

$$y(t) = \int_{-\infty}^{\infty} h(t-\tau)x(\tau)d\tau; \quad Y(s) = H(s)X(s)$$

$$y(n) = \sum_{k} h(n-k)x(k); \quad Y(z) = H(z)X(z)$$

- Here we consider the following general types of filters

1. Ideal filters: $H(j\omega) = \begin{cases} 1 & \text{if } \omega \in \mathrm{I} \\ 0 & \text{otherwise} \end{cases}$ (I is a set of frequencies)
2. Actual low-pass filters, Butterworth: An implementable low-pass filter with certain optimality characteristics.
3. Other high-pass, notch, and band-pass filters
4. DT filters: IIR and computational implementation issues. Finite Impulse Response (FIR) DT filters: $y(n) = \sum_{k=0}^{N} h(n-k)x(k)$ and implementation with Digital Signal Processors(DSP)
5. Zero Order Hold (ZOH): $H(j\omega) = \dfrac{1-e^{-j\omega T}}{j\omega T}$ A special type of filter arising in Digital to Analog Converters (DAC)

Ideal LTI Filters

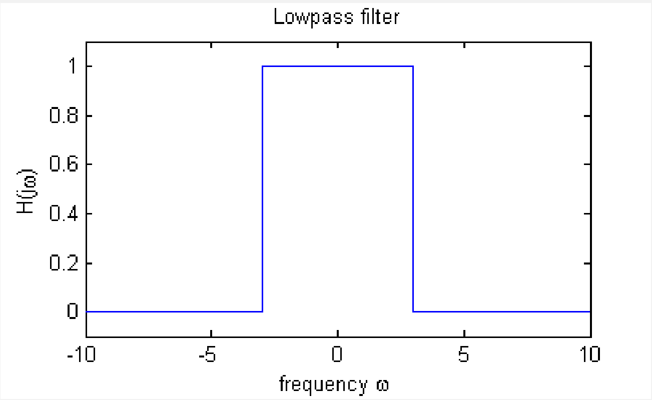
- Ideal filters can be distinguished to

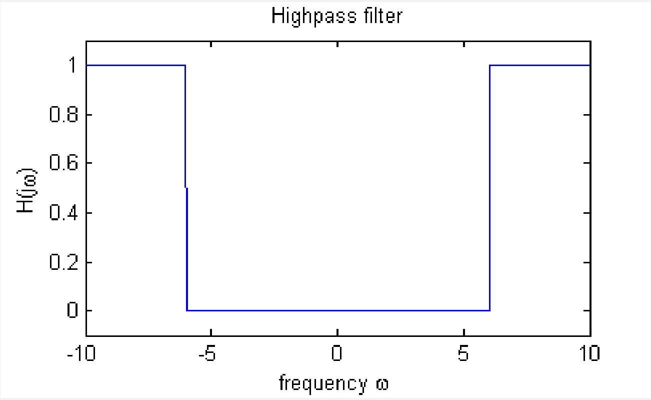
1. Lowpass filters $H(j\omega) = \begin{cases} 1 & \text{if } |\omega| < \omega_c \\ 0 & \text{otherwise} \end{cases}$

2. Highpass filters $H(j\omega) = \begin{cases} 1 & \text{if } |\omega| > \omega_c \\ 0 & \text{otherwise} \end{cases}$

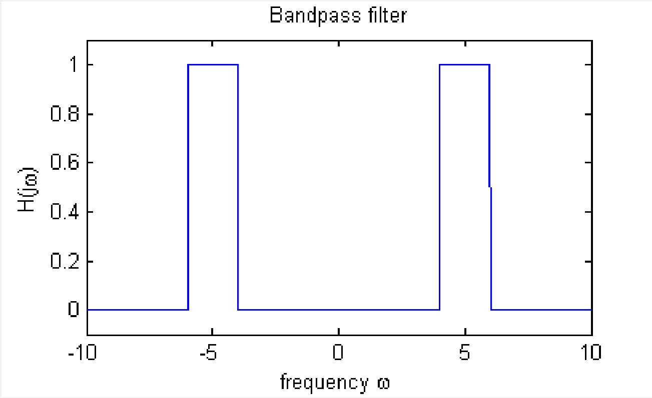
3. Bandpass filters $H(j\omega) = \begin{cases} 1 & \text{if } |\omega \pm w_0| < \omega_c \\ 0 & \text{otherwise} \end{cases}$

4. Bandstop filters $H(j\omega) = \begin{cases} 0 & \text{if } |\omega - w_0| > \omega_c \\ 1 & \text{otherwise} \end{cases}$

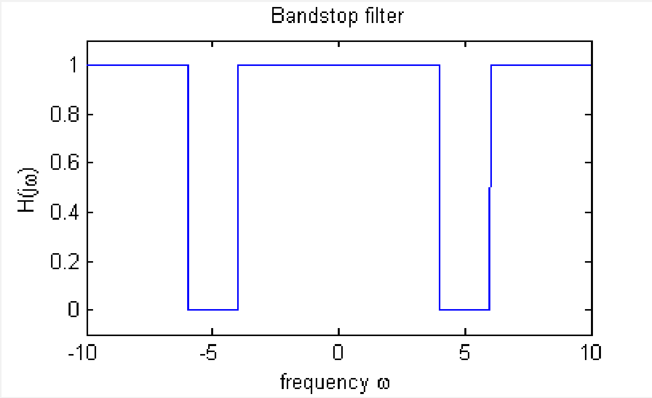




1 – Lowpass



$\text{Lowpass}_1 - \text{Lowpass}_2$

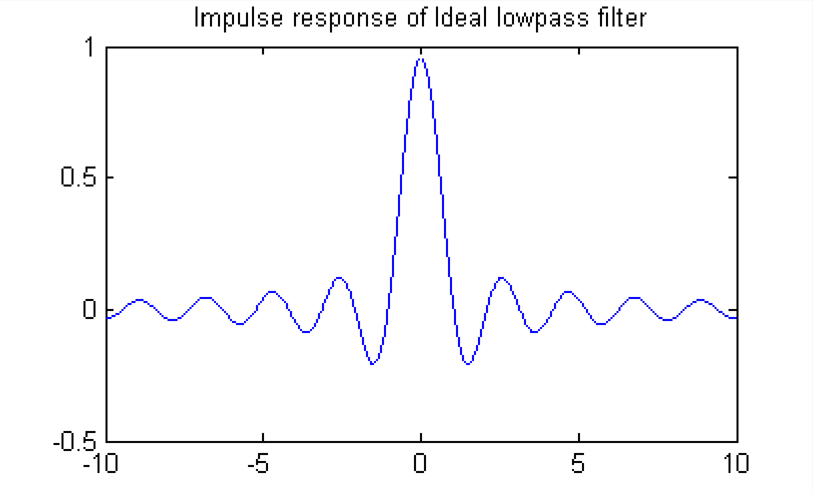


1 – Bandpass

Implementable filters

- Ideal low pass filters are useful in analysis, but must be approximated for implementation purposes
  - The impulse response of an ideal filter is non-causal

    e.g., $H(j\omega)=1, \quad |\omega|<3 \Rightarrow h(t)=\frac{\sin 3t}{\pi t}$
  - It is also unstable (IR is not absolutely integrable)



- There can be many approximations of the ideal low pass filter. Here we focus on the popular Butterworth filters, which have very good properties in the pass-band and are relatively easy to design. Other types of filters include Chebyshev, Elliptic, Bessel, Gaussian, etc. each one satisfying a different optimality criterion.

Butterworth filters

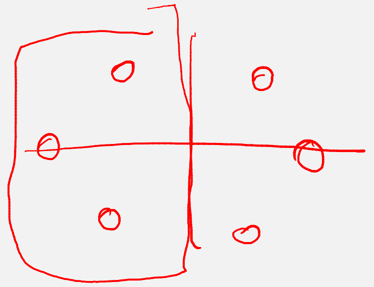
- Butterworth filters offer a simple expression for the transfer function whose gain is

$$|H(j\omega)| = \sqrt{\frac{1}{1+(\omega/\omega_c)^{2n}}}; \quad \omega_c : \text{cutoff frequency}$$

- The Butterworth filter is maximally flat in the passband region, but does not roll-off as fast as other filters in the stopband region.

- As n increases, it approximates better the magnitude of the ideal low pass filter, but the response delay increases.

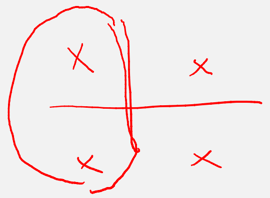
n=3



- Its transfer function is the stable part of $\boxed{H(s)H(-s) = \dfrac{1}{1+(s/j\omega_c)^{2n}}}$

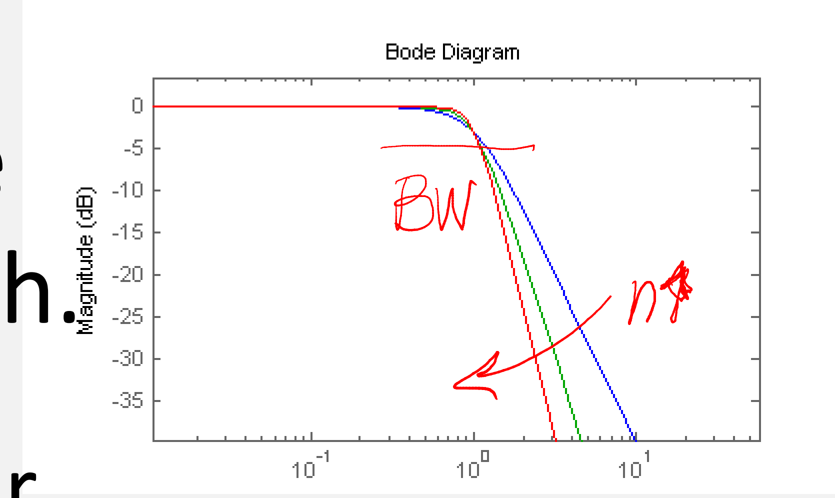
- Or, $H(s) = \dfrac{\omega_c{}^n}{\prod_{k=1:n}\left(s+\omega_c e^{j\frac{2k+n-1}{2n}\pi}\right)}$; Second order: $H(s) = \dfrac{\omega_c^2}{s^2+1.414\omega_c s+\omega_c^2}$; $\zeta = 1/\sqrt{2}$

n=2

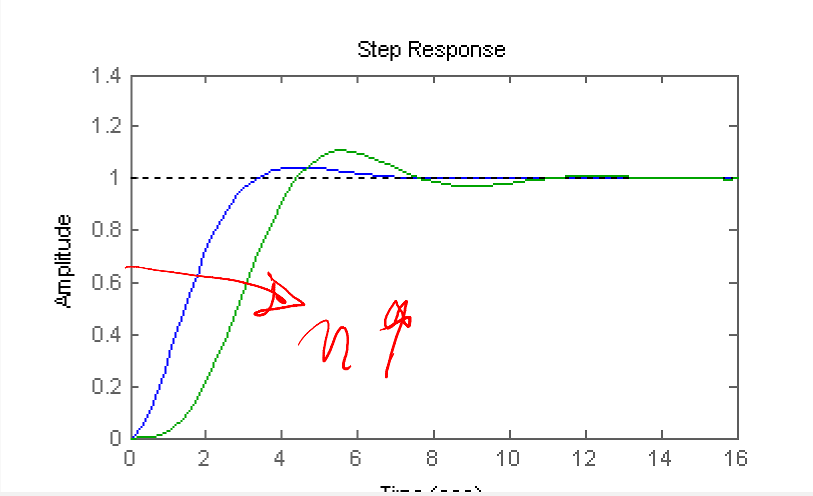


Butterworth filters

- The Bode plots of 2nd, 3rd and 4th order Butterworth are shown in the figure. Notice that all hit the -3dB mark at the cutoff frequency (here 1rad/s), which is the filter Bandwidth.



  - The 4th order Butterworth rolls-off faster than the second order, but its step response has larger delays



  - Relevant MATLAB commands: [num,den]=butter(n,wc,'s');

```
>> [num,den]=butter(2,1,'s');H2=tf(num,den)
        1
---------------------
s^2 + 1.414 s + 1
 >> [num,den]=butter(4,1,'s');H4=tf(num,den)
                  1
--------------------------------------------------
s^4 + 2.613 s^3 + 3.414 s^2 + 2.613 s + 1
>> bodemag(H2,H4), pause, step(H2,H4)
```

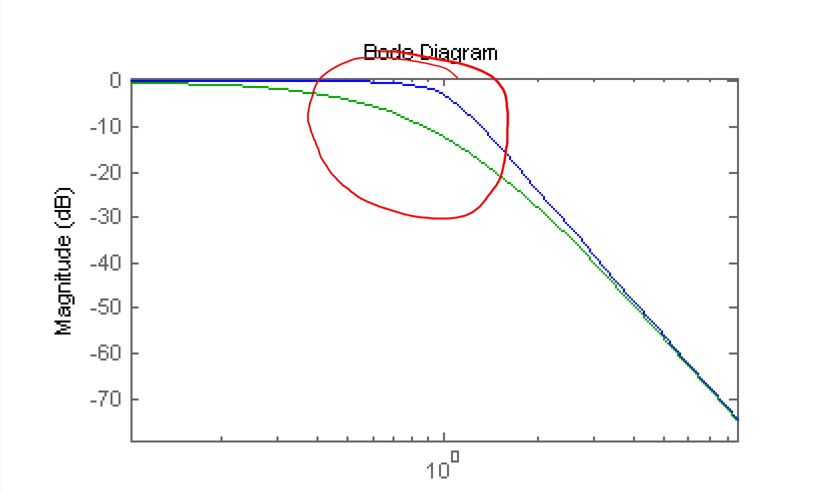
Other Low-pass filters

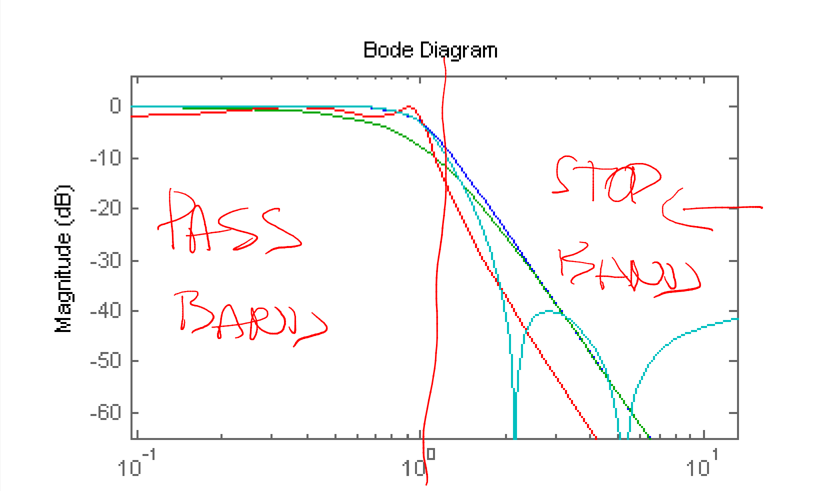
- On the other hand, the naïve implementation of a 4th order low pass filter as a cascade of first order filters

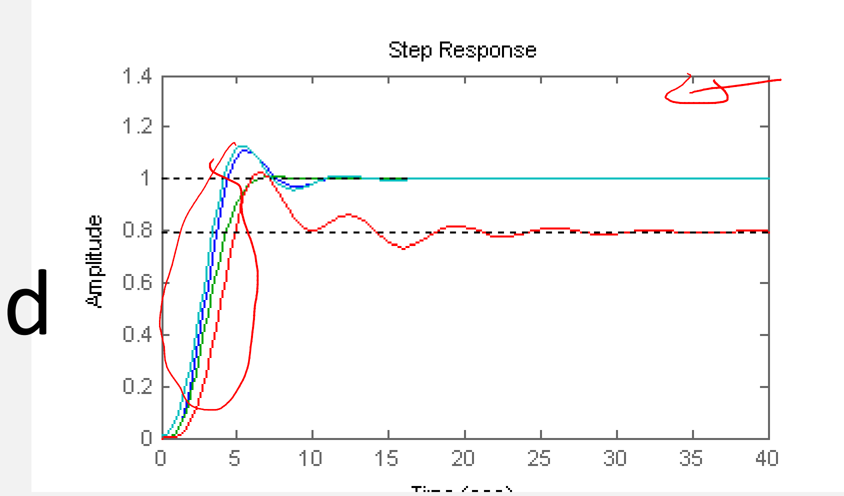
$$H(s) = \frac{1}{(1 + s/\omega_c)^4}$$

  can produce a much worse transition between pass band and stop band. The difference is less pronounced at low orders n but gets worse for large n.

- The adjacent plot shows the Bode plots for 4th order filters: Butterworth, Bessel (linear phase-constant group delay), Chebyshev 1 (passband), and Chebyshev 2 (stopband)
  - The step response of the 4th order Bessel has no overshoot and the 4th order Chebyshev 1 has non-unity DC gain.





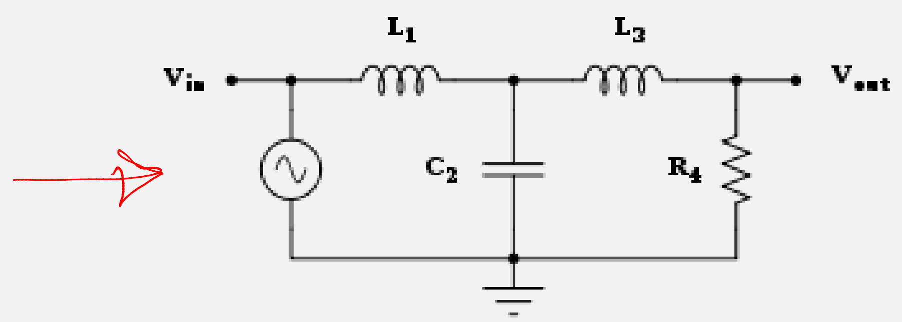


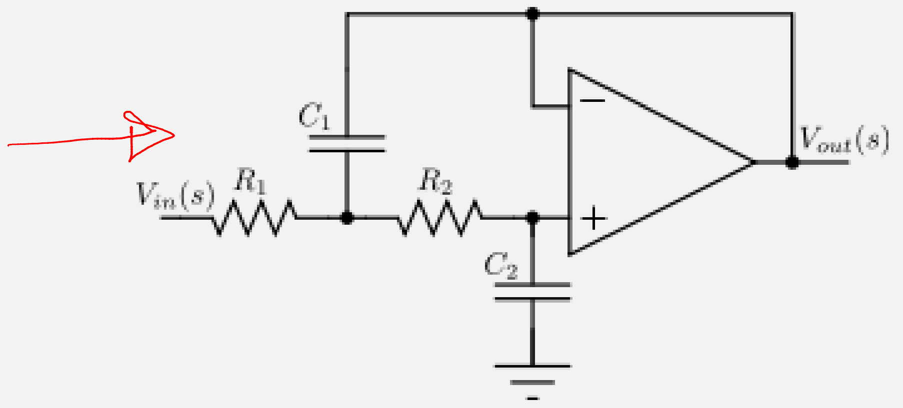
EEE304

Lecture 2.2: Analog Filter Implementation

Analog filter implementation

- Standard circuits for filter implementation:
  - Passive RLC (Cauer topology)
  - Active RC (Sallen-Key topology)





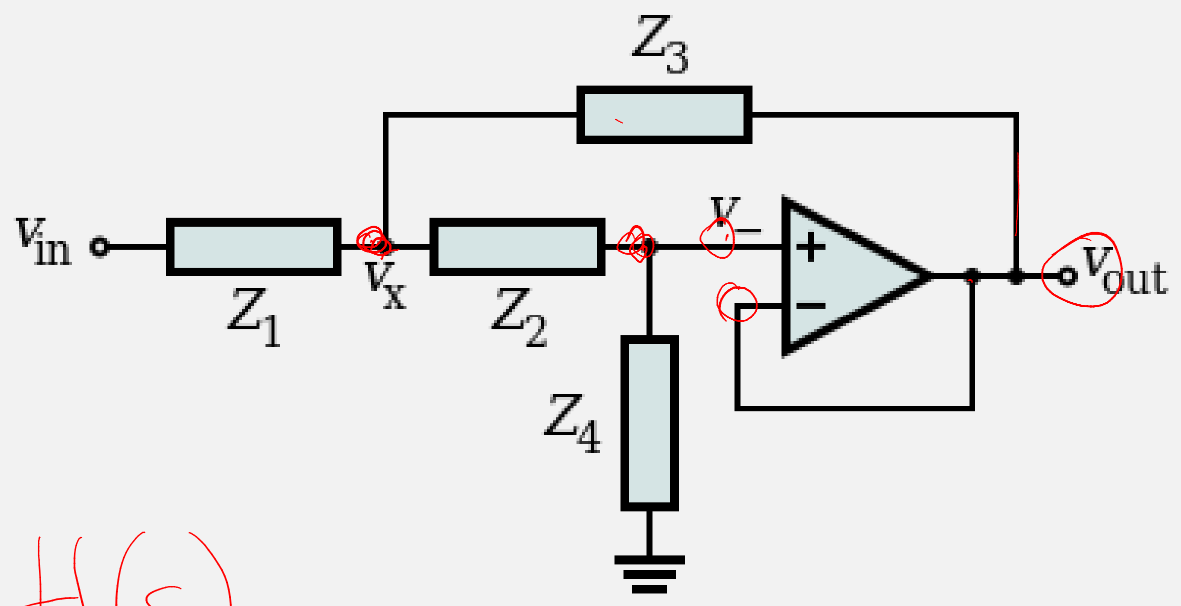
- Generic Sallen-Key (MIT Lincoln Lab, 1955)

$$V_{out} = V_-$$

$$\frac{V_{in} - V_x}{Z_1} + \frac{V_{out} - V_x}{Z_3} + \frac{V_- - V_x}{Z_2} = 0$$

$$\frac{V_x - V_-}{Z_2} + \frac{0 - V_-}{Z_4} = 0$$

$$\frac{V_{out}}{V_{in}} = \frac{Z_3 Z_4}{Z_1 Z_2 + Z_3(Z_1 + Z_2) + Z_3 Z_4} \triangleq H(s)$$



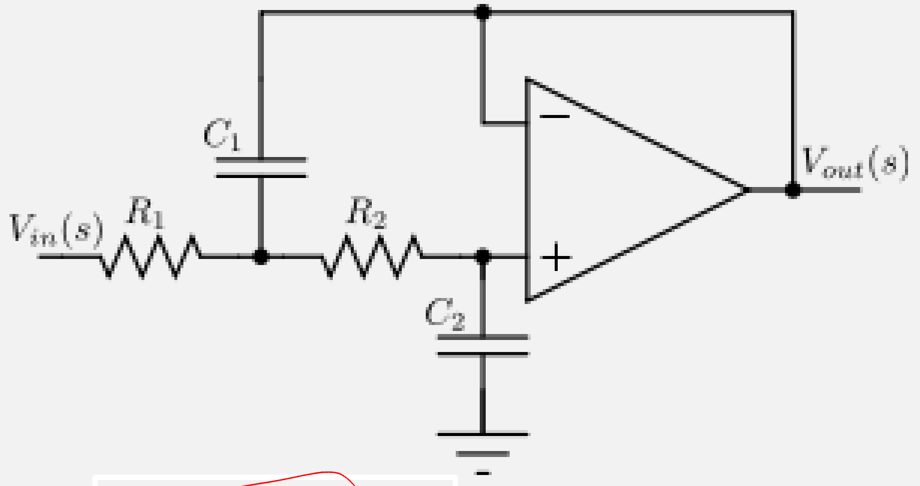
Butterworth filter circuit

- 2nd Order Butterworth (Sallen-Key)

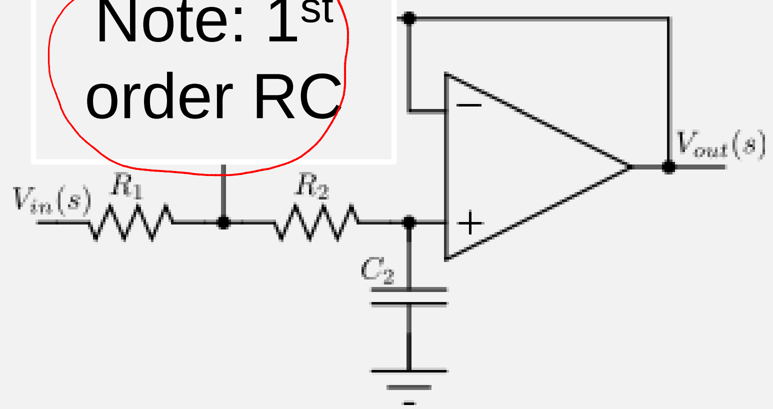
$$\frac{V_{out}}{V_{in}} = \frac{\frac{1}{C_1 s}\frac{1}{C_2 s}}{R_1 R_2 + \frac{1}{C_2 s}(R_1 + R_2) + \frac{1}{C_1 s}\frac{1}{C_2 s}}$$

$$= \boxed{\frac{1}{C_1 C_2 R_1 R_2 s^2 + (R_1 + R_2) C_2 s + 1}} \quad H(s)$$

$$\omega_c = \frac{1}{\sqrt{C_1 C_2 R_1 R_2}}, \quad Q = \frac{\sqrt{C_1 C_2 R_1 R_2}}{C_2 (R_1 + R_2)} = \frac{1}{\sqrt{2}}, \quad \zeta = \frac{C_2 (R_1 + R_2)}{2\sqrt{C_1 C_2 R_1 R_2}} = \frac{1}{\sqrt{2}}$$



Note: 1st order RC



- *"Q"* is the Quality factor (1/(2 damping ratio) and signifies the selectivity of the filter. Q = cutoff frequency / bandwidth and it is < ½ for overdamped and > ½ for underdamped ciruits.

$$H(s) = \frac{\omega_c^{\ 2}}{s^2 + 2\zeta\omega_c s + \omega_c^{\ 2}} = \frac{\omega_c^{\ 2}}{s^2 + \frac{\omega_c}{Q} s + \omega_c^{\ 2}}$$

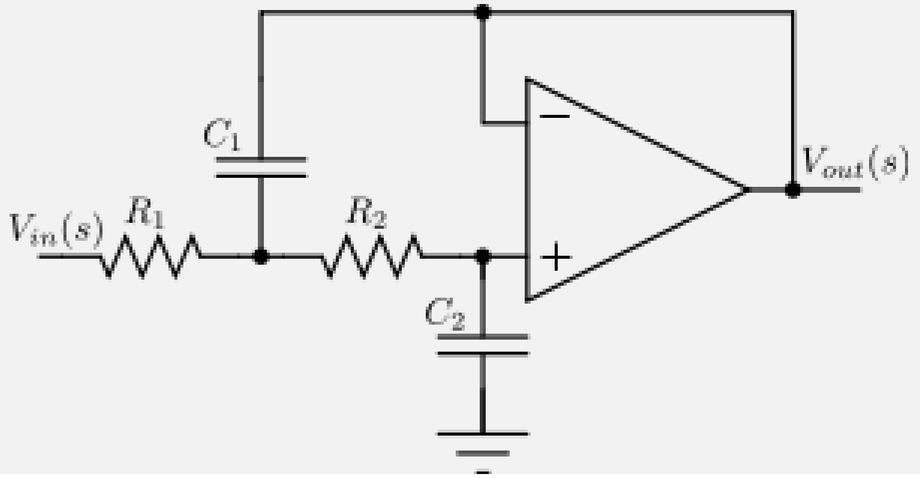
Butterworth filter circuit

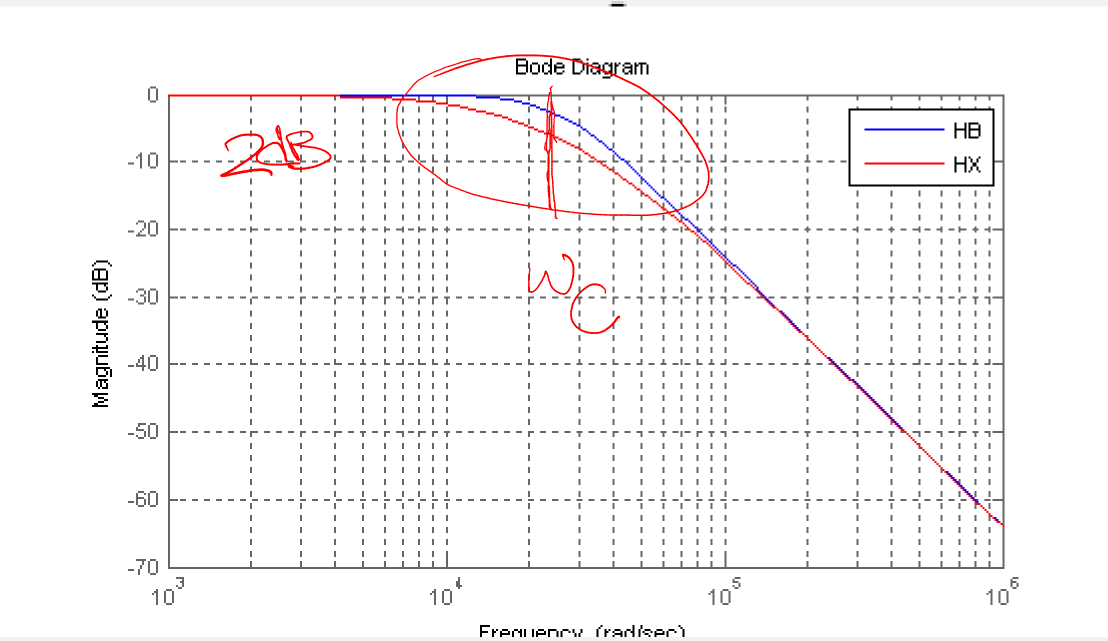
- 2^{nd} Order Butterworth (Sallen-Key) design choices

$$C_1 = nC, \quad C_2 = C, \quad R_1 = mR, \quad R_2 = R$$

$$\omega_c = \frac{1}{RC\sqrt{mn}},$$

$$Q = \frac{\sqrt{mn}}{(m+1)} = \frac{1}{\sqrt{2}} \qquad (=> n > 1)$$





- Say $\omega_c = 4kHz = 25133 rad/s$

$$n = 2 \quad => m = 1$$

$$\omega_c = \frac{1}{RC\sqrt{2}} = 4000 \times 2\pi => \boxed{\frac{1}{RC} = 35543}$$

$$C = 1nF \quad => R = 28135\Omega$$

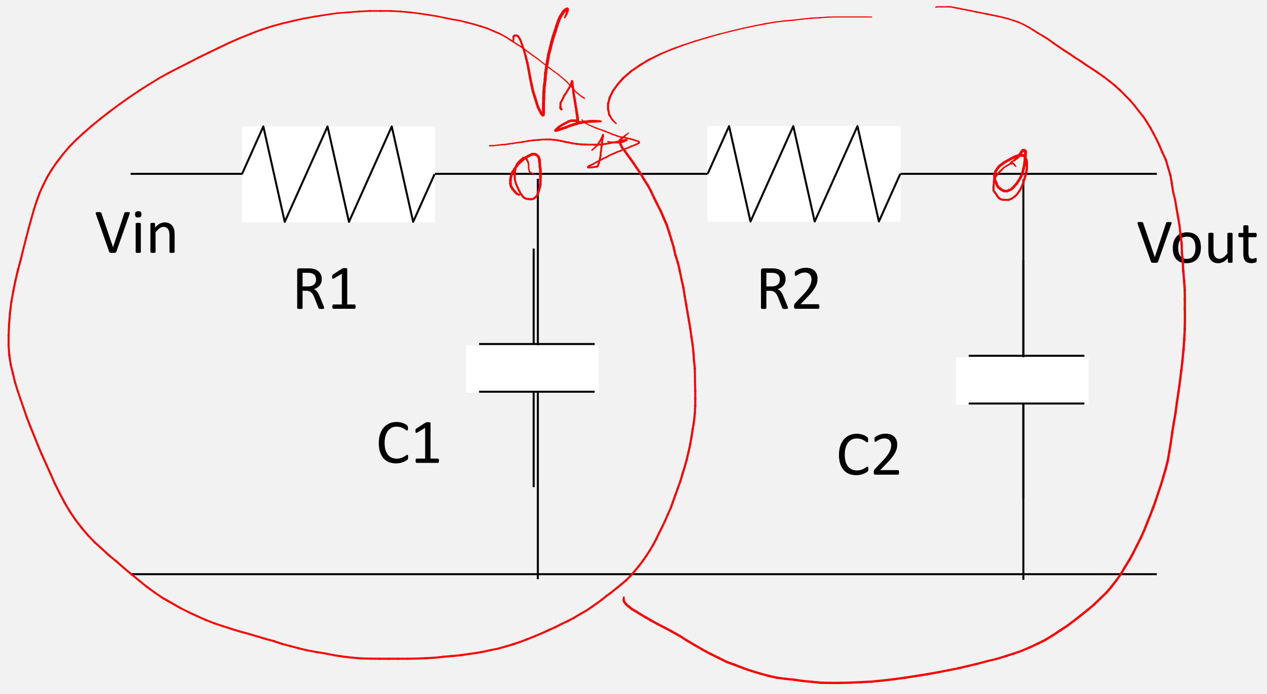
$$=> C_1 = 2nF, \quad C_2 = 1nF, \quad R_1 = 28.1k\Omega, \quad R_2 = 28.1k\Omega$$

$$=> H_B(s) = \frac{1}{C_1C_2R_1R_2s^2 + (R_1 + R_2)C_2s + 1}$$

$$= \frac{1}{1.579\text{e-}009\, s^2 + 5.62\text{e-}005\, s + 1}$$

RC filter circuit

$R_2 >> R_1$



- A passive RC alternative

$$\frac{V_{in}-V_1}{R_1}+\frac{V_{out}-V_1}{R_2}+\frac{-V_1}{Z_1}=0$$

$$\frac{V_1-V_{out}}{R_2}+\frac{0-V_{out}}{Z_2}=0$$

$$\frac{V_{out}}{V_{in}}=\frac{1}{\left[1+\frac{R_1}{Z_1}+\frac{R_2}{Z_2}+\frac{R_1}{Z_2}+\frac{R_1R_2}{Z_1Z_2}\right]}=\frac{1}{R_1R_2C_1C_2s^2+(R_1C_1+R_2C_2+R_1C_2)s+1} = H(s)$$

- R_1C_2s is the "loading" term. This circuit will always be overdamped but will approach critical damping for small loading terms (large R2, does not draw much current from 1st stage, but has high output impedance and requires an op-amp follower (buffer) at the end.

$$\omega_c=\frac{1}{RC}=4000\times 2\pi \Rightarrow \frac{1}{RC}=25133$$

$$C=10nF \quad \Rightarrow R=3979\Omega$$

$$\Rightarrow C_1=1nF,\quad C_2=10nF,\quad R_1=4.0k\Omega,\quad R_2=39.8k\Omega$$

$$\Rightarrow H_X(s)=\frac{1}{1.592\text{e-}009\,s^2+8.38\text{e-}005\,s+1}$$

$$(\zeta=1.05)$$

Filtering example

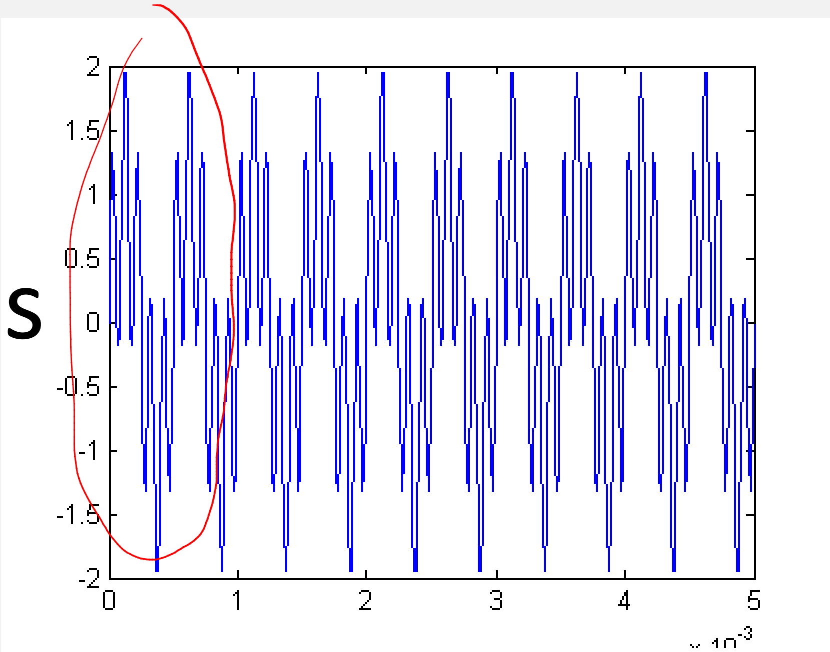
- Consider the problem of filtering out high frequency noise (say, 10kHz), from a 2kHz vibration signal using our low pass filters HB,HX

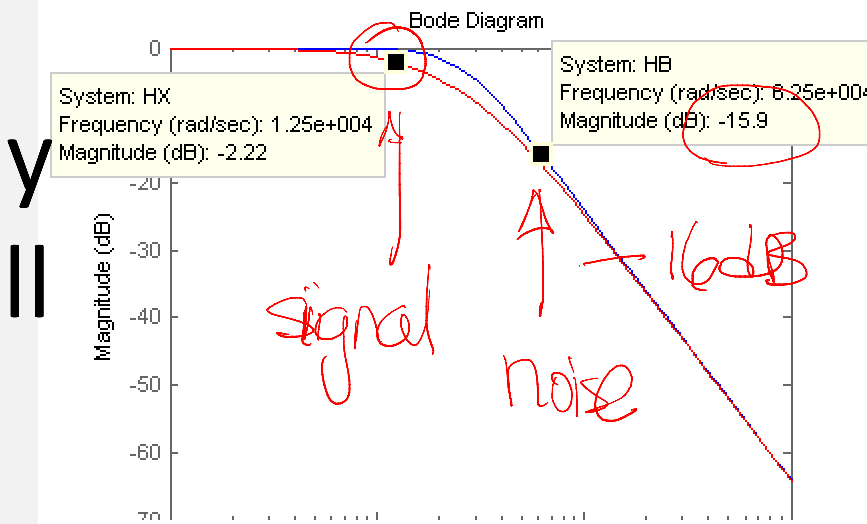
$$x(t) = x_s(t) + x_n(t) = \sin(2\pi 2000 t) + \sin(2\pi 10000 t)$$

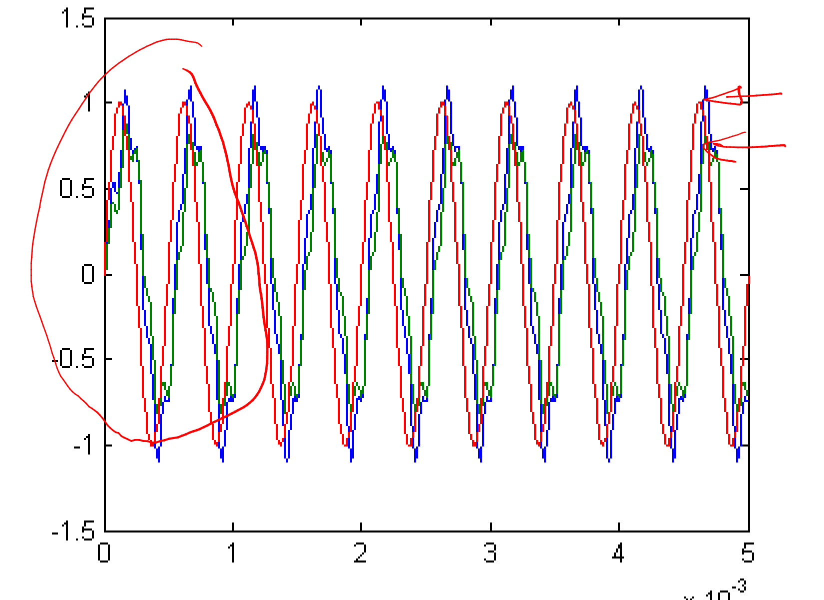
$$y_B = H_B[x], \quad y_X = H_X[x]$$

- The filters attenuate the high frequency signal considerably but due to its proximity to the cutoff frequency there is still significant residual. Also notice the delay of the low frequency signal relative to the original (red).

```
>> t=(0:1e-5:.005)';
>> x=sin(2*pi*2000*t)+sin(2*pi*10000*t);
>> yB=lsim(HB,x,t); yX=lsim(HX,x,t);
>> plot(t,x), plot(t,yB,t,yX,t,sin(2*pi*2000*t))
```



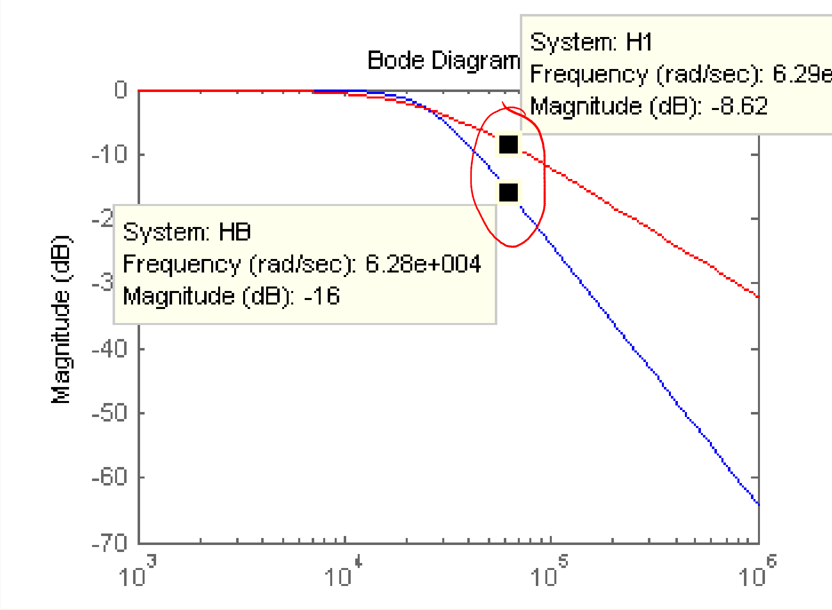




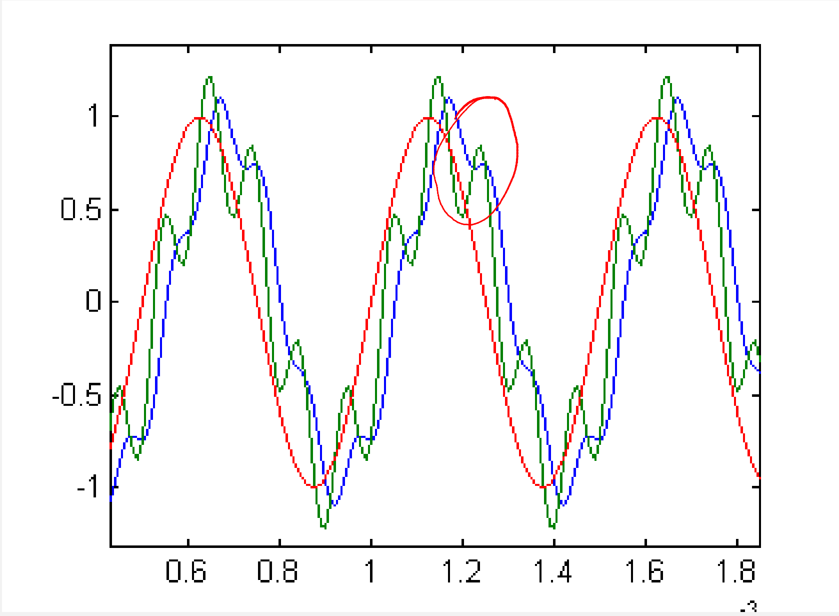
Filtering example

- Let us also compare with a 1st order filter response. Such filters are very simple and easy to implement so they can have an advantage if they offer adequate performance. In this case RC is

$$\omega_c = \frac{1}{RC} = 4000\times 2\pi => \frac{1}{RC} = 25133 \quad => C = 1nF \quad => R = 39.8k\Omega$$



Green: 1st ord



- Once can now see the increased contribution of the noise to the output (blue-green) by about 7dB (x2).

- A variant of this problem is the design of an antialiasing filter that trades off complexity and cost with the transmission of the useful low frequency signal up to 2kHz, while it reduces the high frequency components (above Nyquist frequency) by 40dB.
  - This is a two-parameter study (bandwidth-order), suitable for numerical optimization

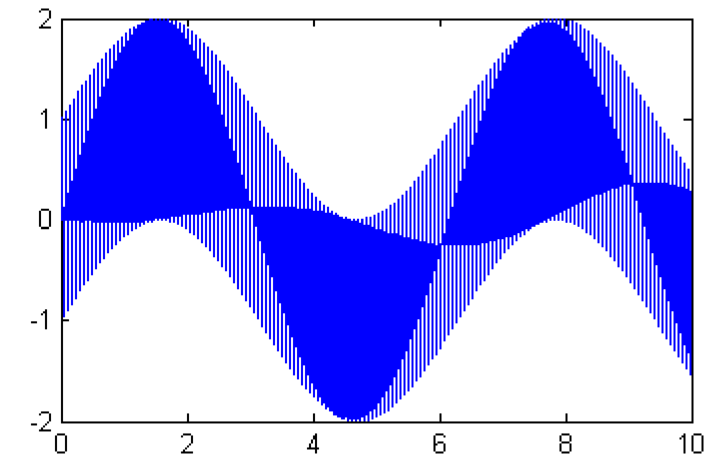
Filter design example

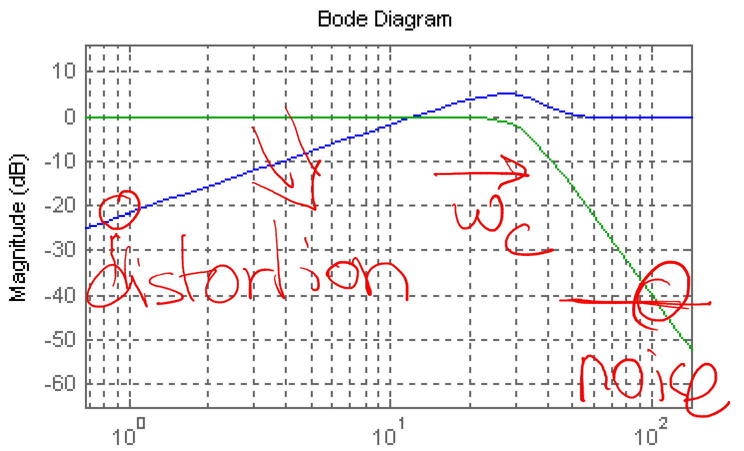
- Consider the problem of separating a useful low frequency signal (ws = 1rad/s), from additive high-frequency noise (wn = 100rad/s)

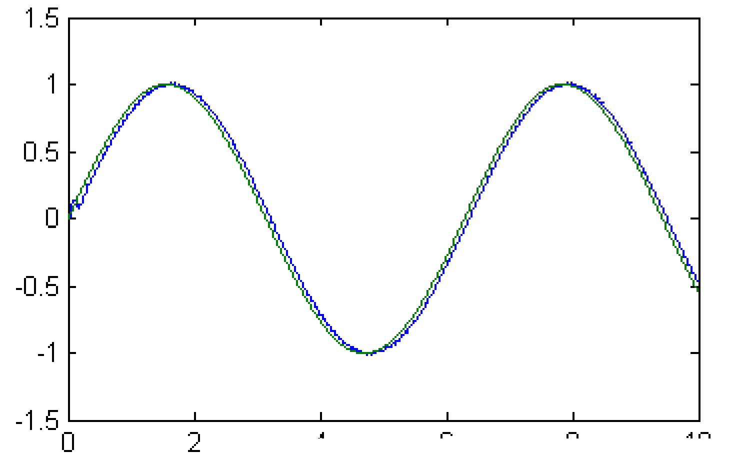
$$y(t) = x_s(t) + x_n(t) = \sin(t) + \sin(100t)$$

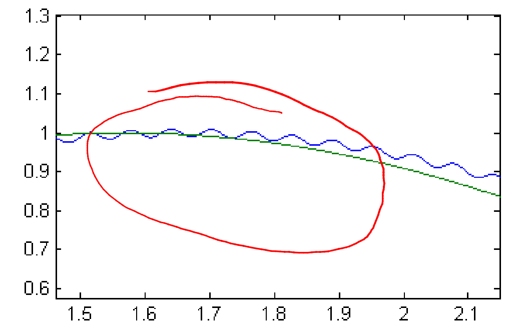
We want to design a 4th order Butterworth filter that will reduce the noise amplitude by 40dB and minimize the distortion of the useful signal.

  - The error is $x_s - H[y] = x_s - H[x_s + x_n] = \overbrace{(1-H)[x_s]}^{\text{distortion}} - \overbrace{H[x_n]}^{\text{noise}}$
  - Observing the Bode plots of 1-H and H (MATLAB: bode(1-H,H)), we design the Butterworth cutoff frequency as high as possible to achieve magnitude -40dB at 100rad/s: Adec*(4x20dB/dec)=40 => A = ½ dec = 3.16 => wc = 31.6rad/s.
  - Compute the response and compare with xs(t): plot(t,lsim(H4,y,t),t,xs)







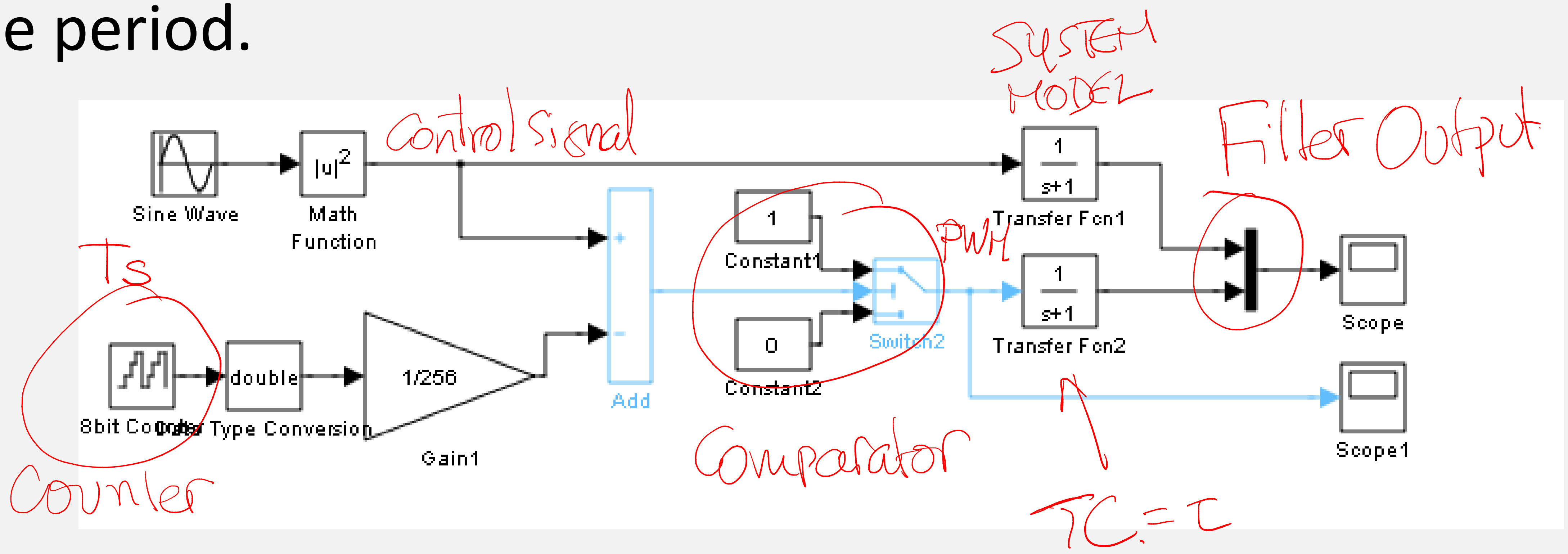


Filtering in a PWM application

- Pulse Width Modulation (PWM) is a widespread and power-efficient method to apply analog control signals to systems where the important quantity is the average energy of the input signal but not its exact shape.
  - Typical examples of PWM applications are in heat generation, motor speed control, DC-DC converters, power electronics. On the other hand, an application like the motor position control or valve position regulation for flow control may not be appropriate.
- The main concept behind PWM is that the control signal power modulates the duty cycle of a square wave (easily done with simple electronics) and this signal is effectively filtered by the system.
- Our interest is to define the PWM period so that all the harmonics are sufficiently attenuated by the system and its output looks smooth enough. (Essentially an inverse problem where we are given the filter and design the input.)

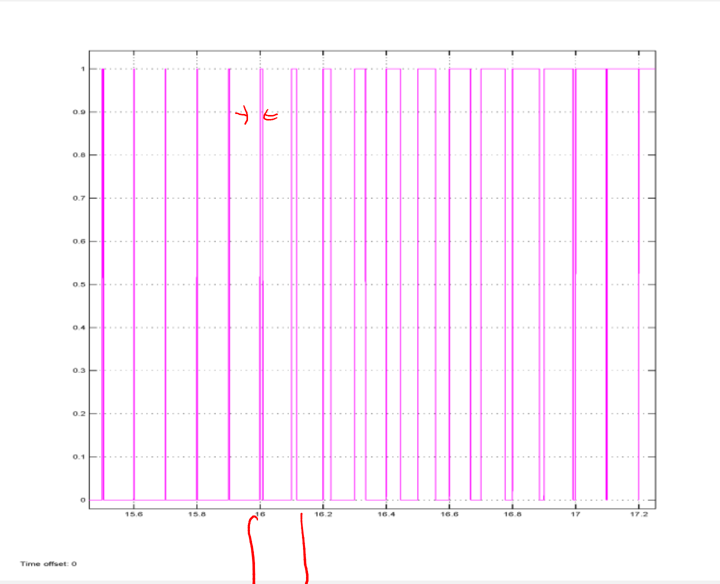
Filtering in a PWM application

- Ignoring some of the details (exact scaling, feedback correction), a simple PWM scheme is obtained by comparing our signal, expressed in a 0-1 scale, with a counter that resets to 0 at the end of the period.

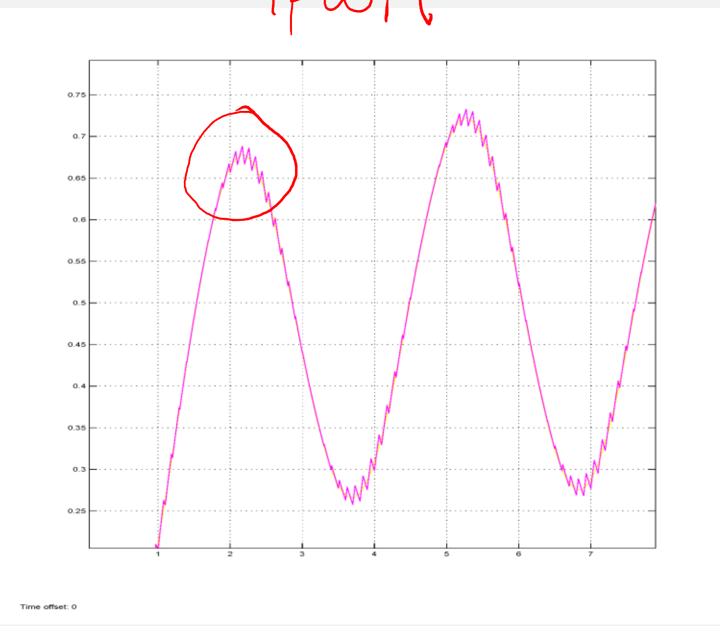


- We compare the response of the analog input with its PWM conversion, filtered by a 1st order filter of bandwidth 1. The top signal is the PWM itself, while the bottom plot shows the outputs of the two filters.

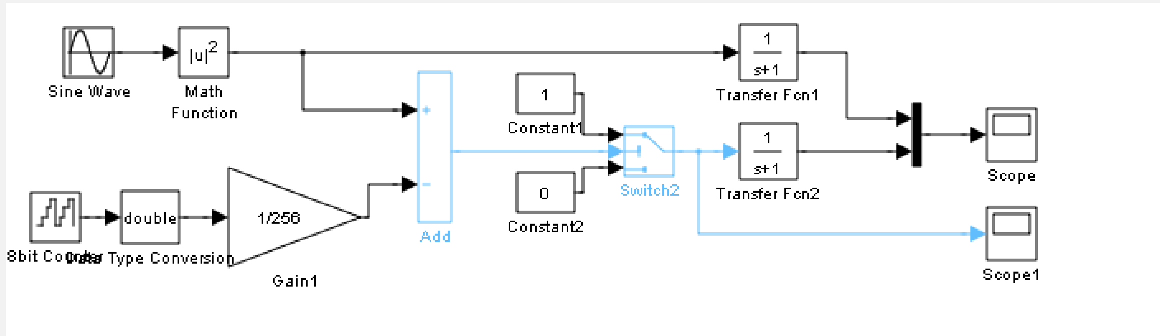
CONTROL SIGNAL



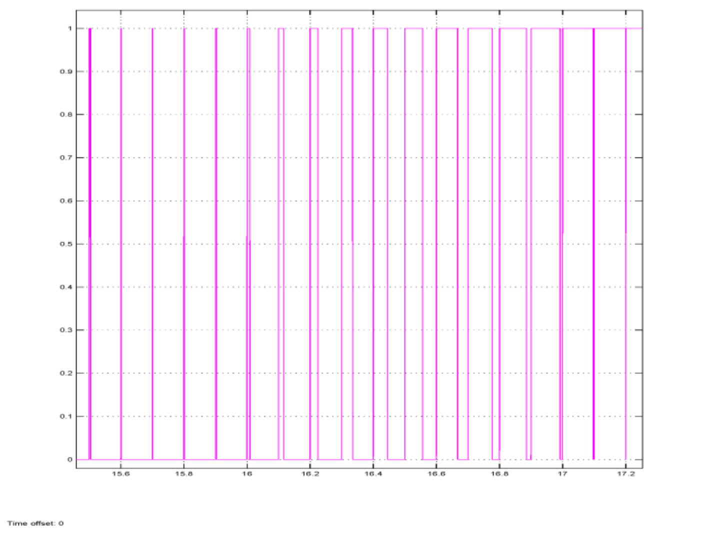
TPWM



Filtering in a PWM application

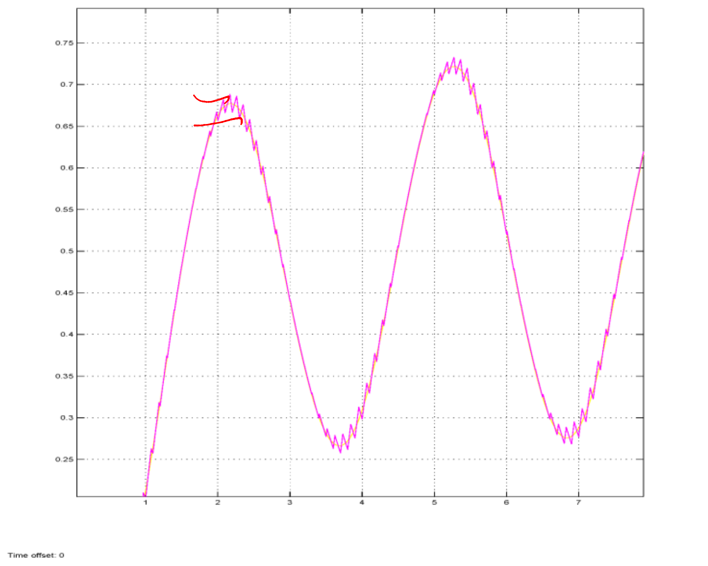


- There are two basic parameters in the PWM definition. One is the sampling time of the counter, say Ts, when it is incremented by 1. The other is the number of bits in the counter, from which the PWM period is Ts * 2^bits. In our case, we choose Ts = 0.4ms and an 8 bit counter, yielding a PWM period 0.1s.



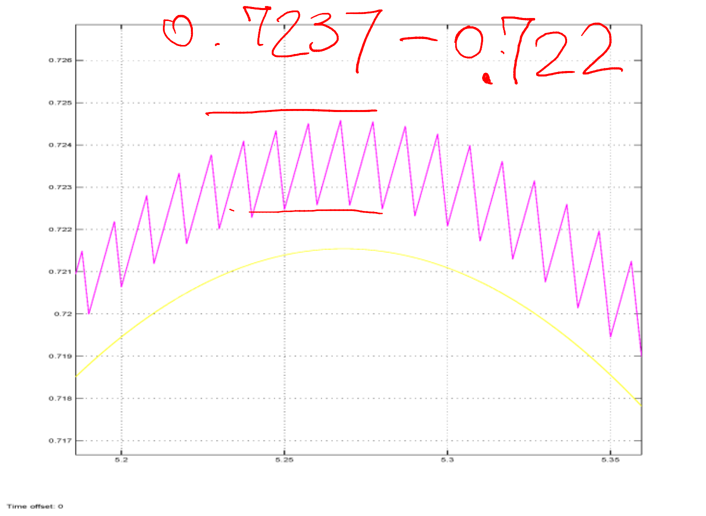
- Thus, the first harmonic of the PWM signal (ignoring the slow variation of the sinusoid) will be at 10Hz, where the filter amplitude is 0.016: >> m=bode(tf(1,[1 1]),10*2*pi). This is consistent with the observed ripple in the response.

frequency rad/s



- Since the filter rolls-off by -20dB/dec, we anticipate that the ripple amplitude will drop by an order of magnitude if we increase the PWM frequency by a factor of 10. Another simulation verifies this expectation.

Ts = 0.04 ms



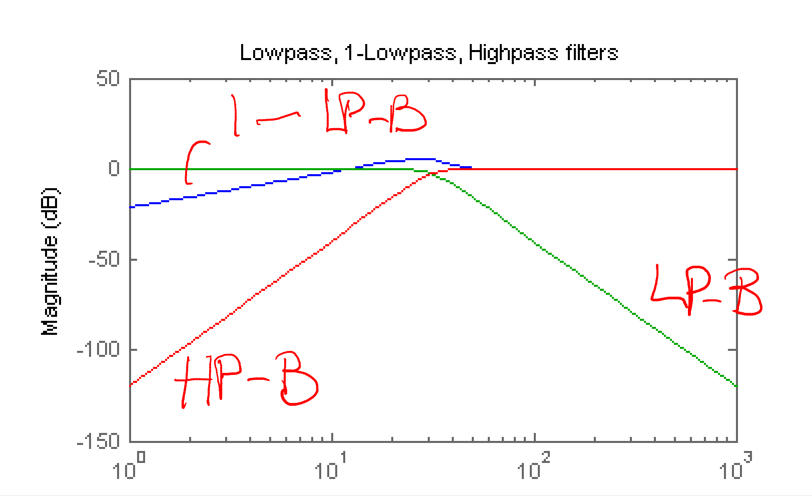
EEE304

Lecture 2.3: More Analog Filters

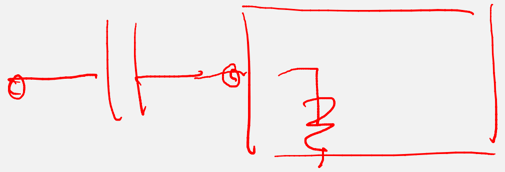
ARIZONA STATE UNIVERSITY

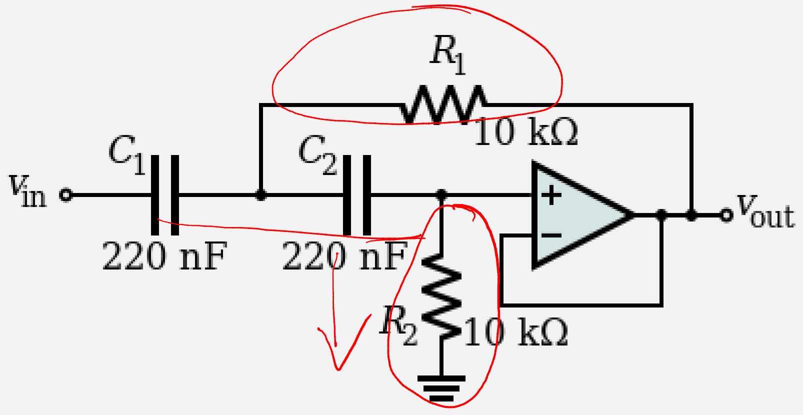
High-pass filters

- Consider the problem of separating a useful high frequency audio signal (ws = 1kHz), from additive low-frequency drift (wn = 1Hz).
  - Here we need a high pass filter. Notice that although for H=lowpass , 1-H is high pass, it is not a good one, except for the ideal case. For the Butterworth case, the high pass Butterworth is $H_{HP}(s) = s^n H_{LP}(s)$

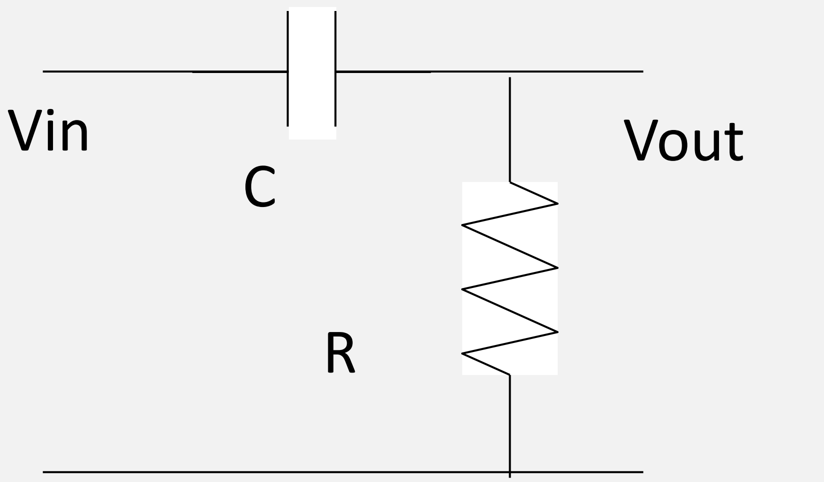


- The Sallen-Key topology for a high pass Butterworth remains similar, exchanging resistors and capacitors. (Omitting the first stage also yields the corresponding 1st order high pass filter with a buffer op-amp.
  - It is worthwhile to note that the filter time constant, determined by RC is also responsible for the settling time of the filter and that may be an important quantity for applications.





High-pass filters



- Returning to our problem, we want the filter corner frequency to be less than 1kHz, and higher than 1Hz. Let's choose a first order RC with corner frequency the geometric mean $\sqrt{1 \cdot 1000}$

$$\omega_c = \frac{1}{RC} = 32 \times 2\pi => \frac{1}{RC} = 200 \quad => C = 100nF \quad => R = 50k\Omega$$

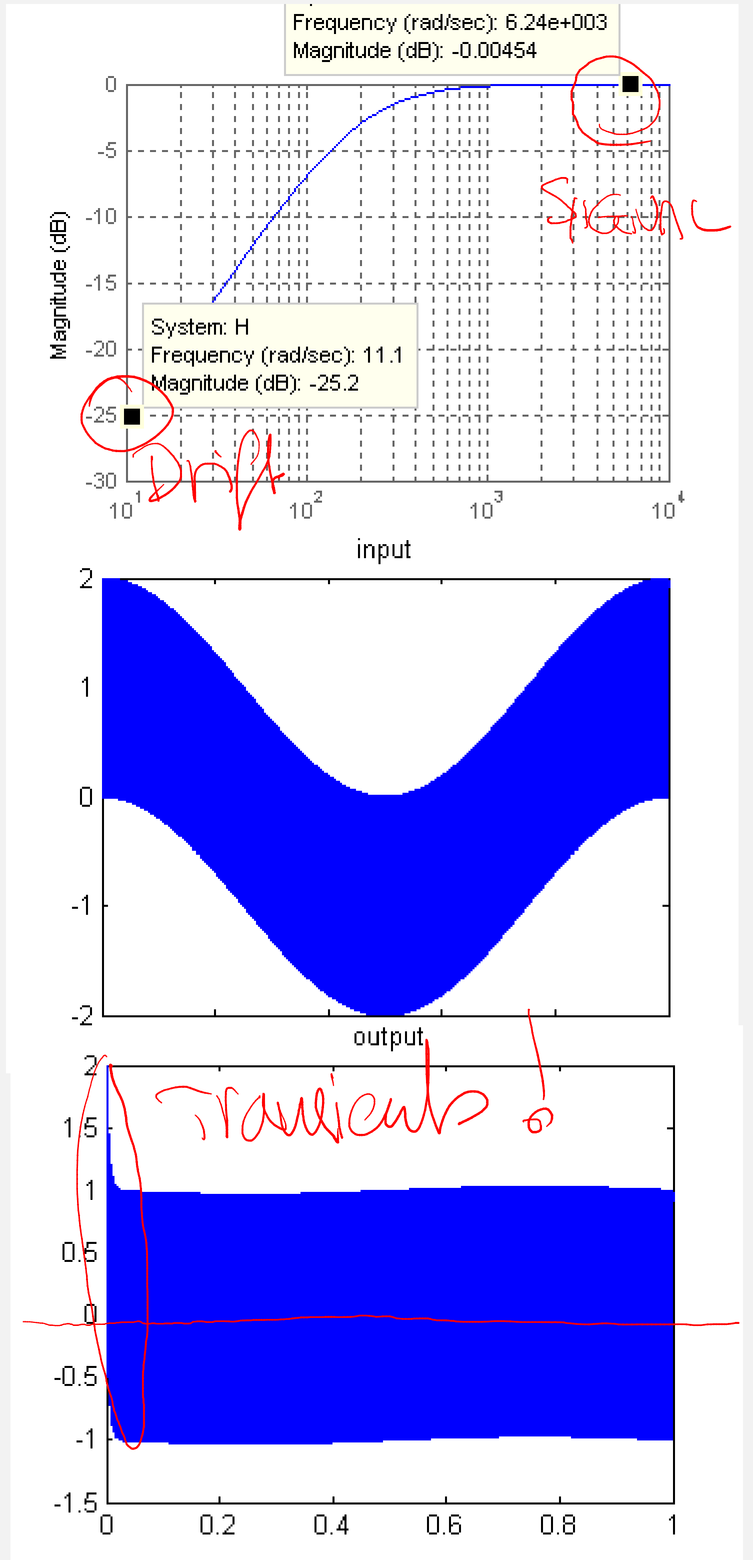
- Applying a simple voltage division we find the transfer function

$$\frac{V_{out}}{V_{in}} = \frac{Z_R}{Z_R + Z_C} = \frac{R}{R + 1/sC} = \boxed{\frac{RCs}{RCs+1} = H(s)}$$

- And evaluate its Bode plot and response to a test signal

$$x(t) = \cos(2\pi t) + \cos(2\pi 1000t)$$

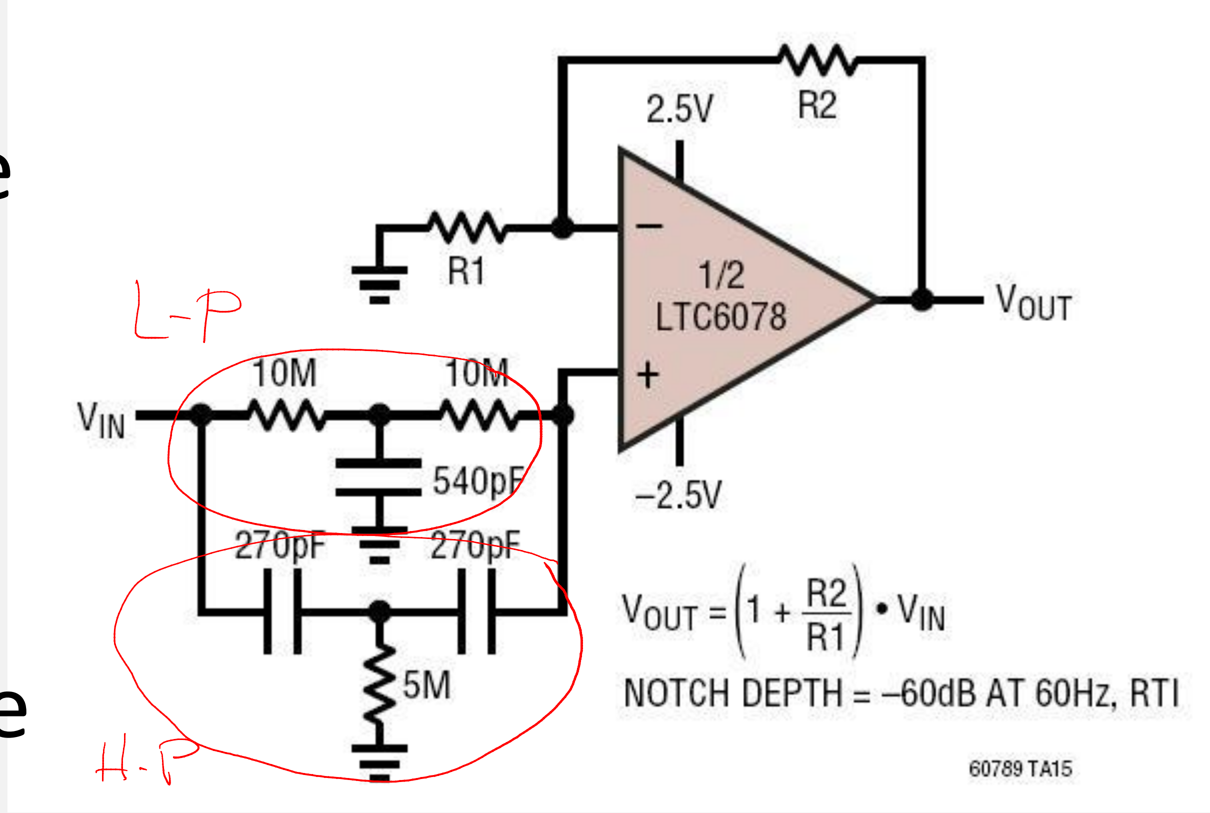
- At the output, the slow oscillation is barely visible. But notice the initial transient period of about 20ms (4 x RC –time constants). If only a capacitor is used in series, one then relies on the input impedance of the rest of the circuit. That can be very high resulting in unpredictable initial transients.



Band-stop or notch filters

- Notch filters reject a narrow range of frequencies. A typical need for a notch filter is the EM interference due to AC currents (60Hz).

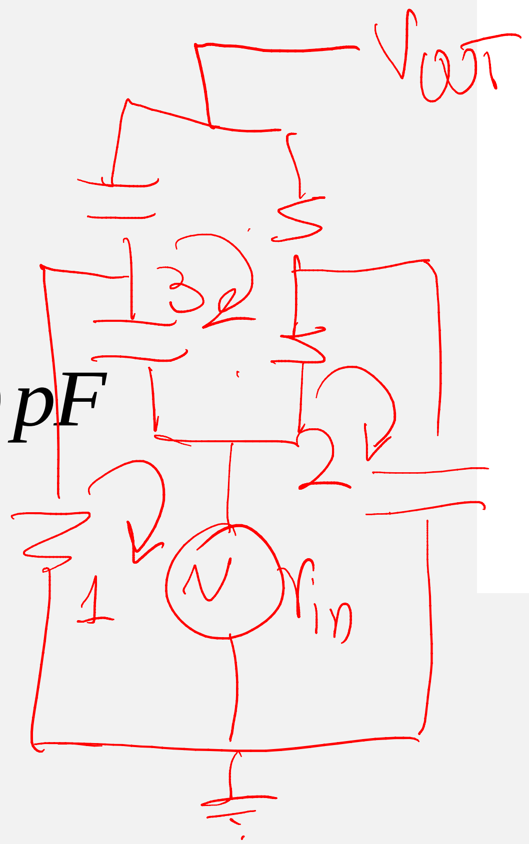
- One popular topology is the “twin-T”, a name originating from its use of a high pass and a low pass configuration of resistors and capacitors. One advantage is the use of only R and C but their quality factors are not very high.



Circuit from Linear Technology solutions

- An interesting and useful exercise with this more complicated filter is the derivation of its transfer function using computational MATLAB tools. In particular, we use the transfer function overloaded methods to solve the nodal (or loop) equations numerically.

Band-stop or notch filters

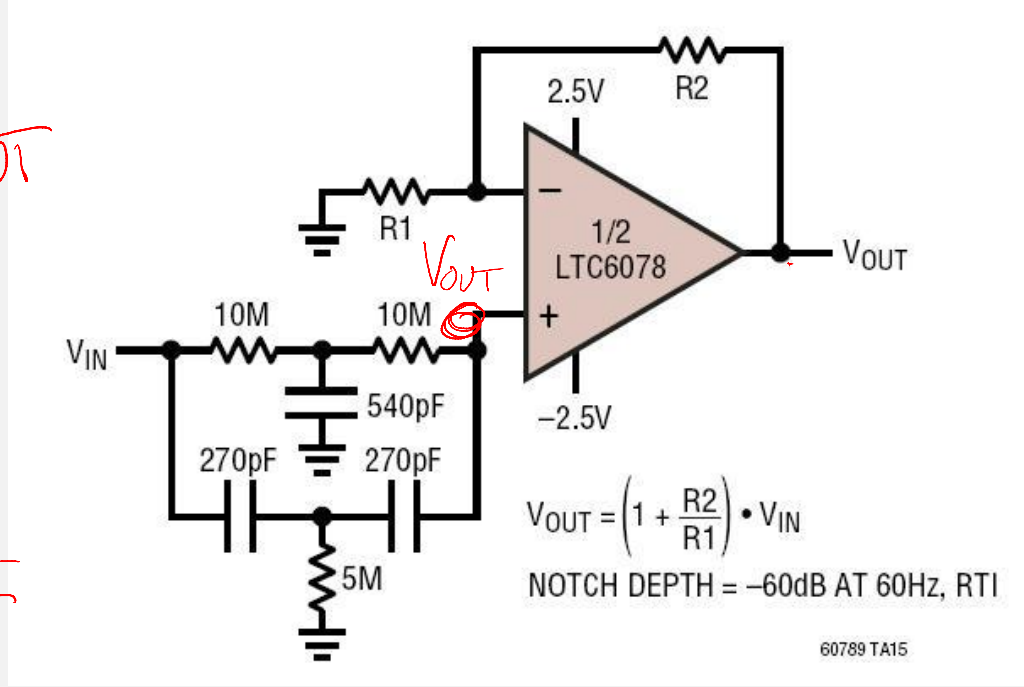


- Define the impedances (omit any gain resistors)
  $Z_1 = 10M\Omega, \quad Z_2 = 5M\Omega, \quad Z_3 = 1/s270pF, \quad Z_4 = 1/s540pF$
- And write the loop equations (with $V_+ = V_- = V_{out}$)

$$I_1 Z_2 + I_1 Z_3 - I_3 Z_3 + V_{in} = 0$$

$$I_2 Z_1 - I_3 Z_1 + I_2 Z_4 - V_{in} = 0$$

$$I_3 Z_3 - I_1 Z_3 + I_3 Z_3 + I_3 Z_1 + I_3 Z_1 - I_2 Z_1 = 0$$



Circuit from Linear Technology solutions

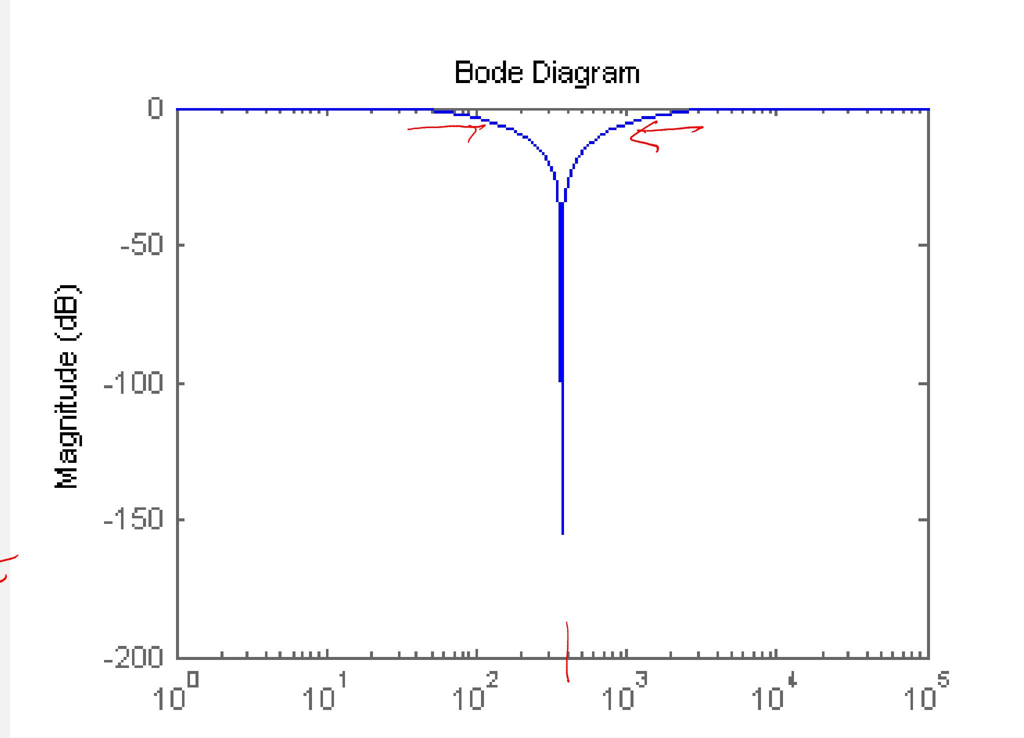
- Then, enter the MATLAB matrices for the problem $AI = bV_{in}, \quad V_{out} = cI + dV_{in}$, where the entries of A,b,c,d may be transfer functions (systems)

```
>> R1=10e6;R2= 5e6;C1=270e-12;C2=540e-12;
>> s=tf('s');Z1=R1;Z2=R2;Z3=1/C1/s;Z4=1/C2/s;
>> A=[Z2+Z3 0 -Z3;0 Z1+Z4 -Z1;-Z3 -Z1 2*Z1+2*Z3];
>> b=[-1;1;0];c=[0 -Z1 2*Z1];d=1;
>> H=minreal(d+c*inv(A)*b)
s^2 - 3.068e-012 s + 1.372e005
--------------------------------------
  s^2 + 1481 s + 1.372e005
>> bodemag(H)
```

s= tf([1,0],1)

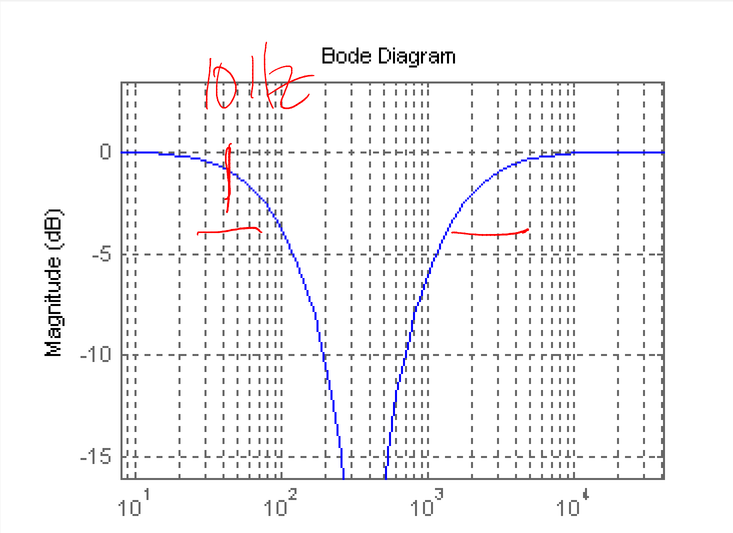
minreal = cancel common factors

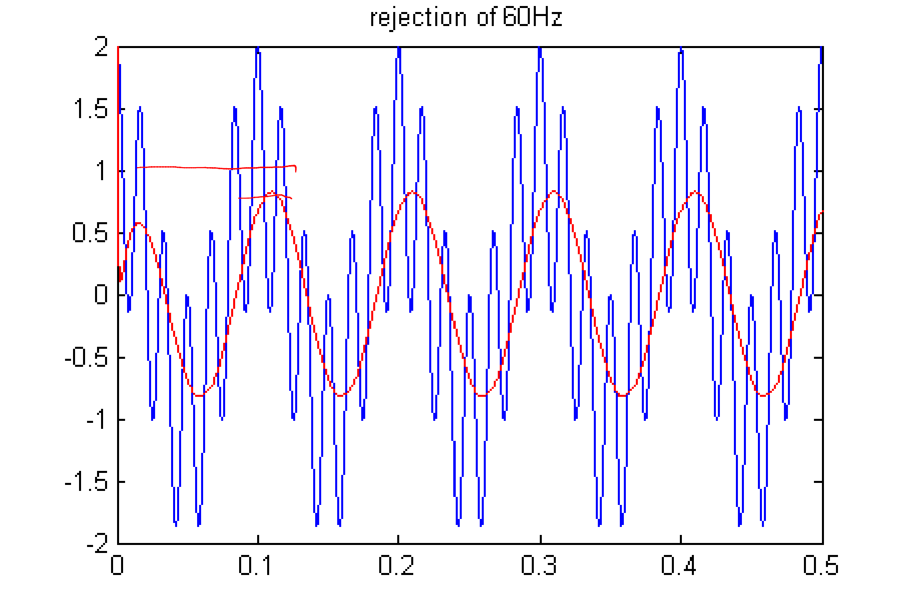
zero at $\omega_0 = \sqrt{1.372E5} = 60Hz$

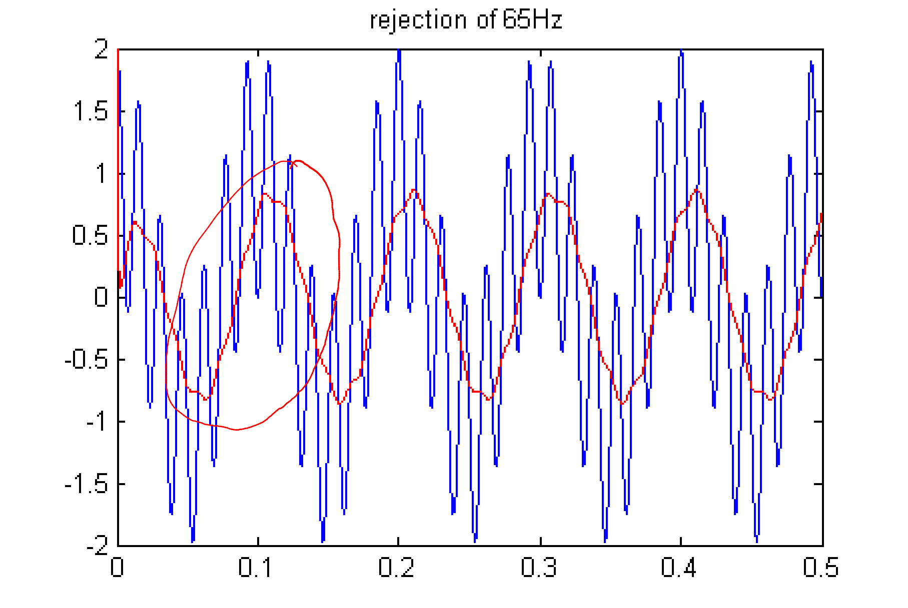


Band-stop or notch filters

- The filter has a zero exactly at 60Hz by construction but its quality factor is not very high (Df ~ 900)
- In the time-domain simulations we observe the rejection of a 60Hz cosine while the 10Hz cosine is attenuated by ~10%.
- On the other hand, the 65Hz cosine is also attenuated showing only moderate selectivity.
- A different notch filter, based on RLC combination, is discussed next. Such filters are more selective but more sensitive to component variability. Other possible choices are the use of discrete-time implementation of high-order filters (e.g., MATLAB’s “designfilt” command).

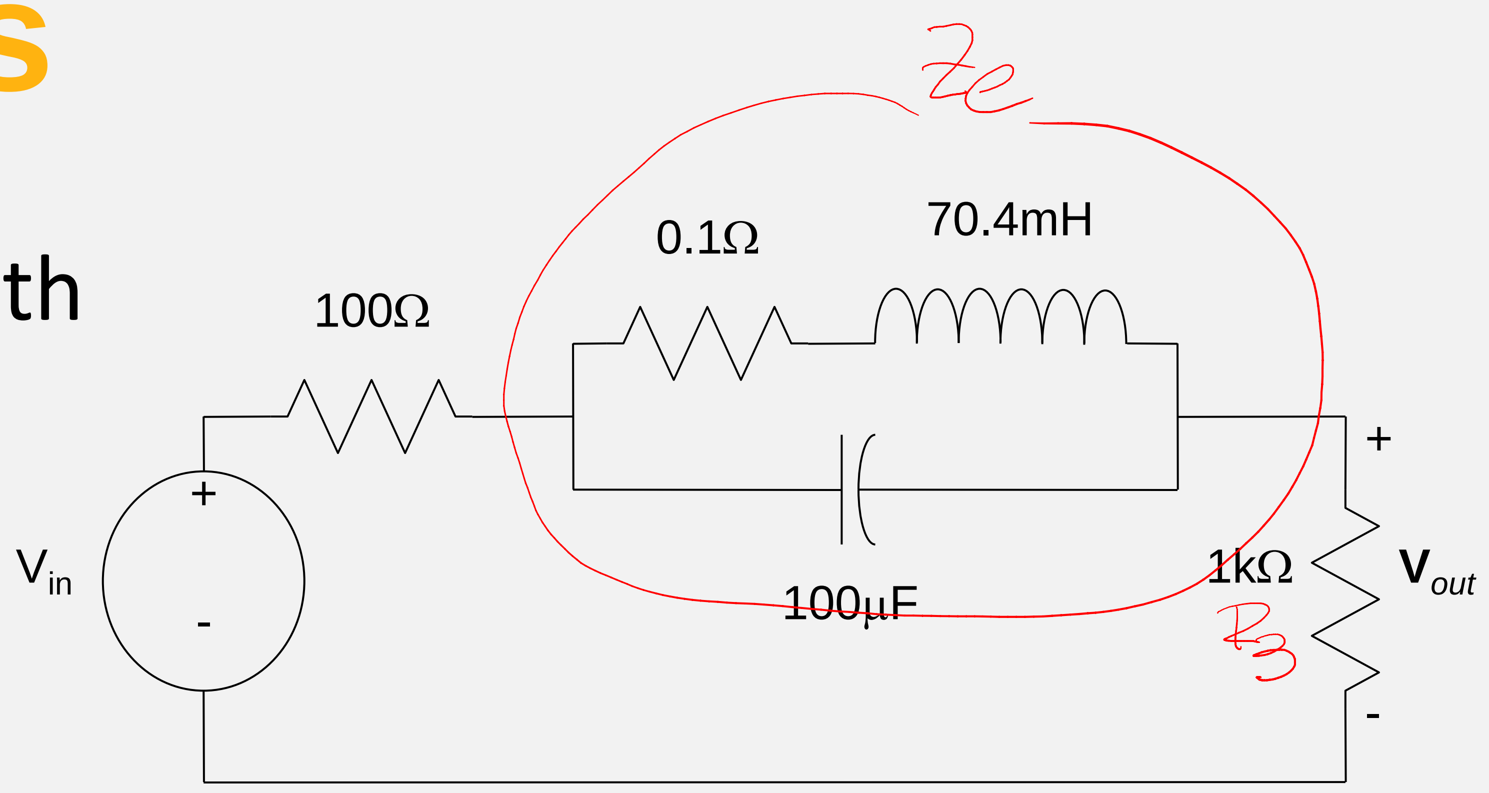






Band-stop or notch filters

- This filter uses an LC loop to zero-out 60Hz with high selectivity. Its transfer function can be obtained with fairly standard computations (impedance parallel/series combinations)



$$V_{in} = ZI; \quad V_{out} = R_3 I = \frac{R_3}{Z} V_{in}$$

$$Z = R_1 + R_3 + Z_e; \quad Z_e = \left(\frac{1}{sL + R_2} + sC \right)^{-1} = \frac{sL + R_2}{s^2 LC + sCR_2 + 1}$$

$$Z = \frac{s^2 LC(R_1 + R_3) + s[CR_2(R_1 + R_3) + L] + (R_1 + R_2 + R_3)}{s^2 LC + sCR_2 + 1}$$

$$V_{out} = \frac{s^2 LCR_3 + sCR_2 R_3 + R_3}{s^2 LC(R_1 + R_3) + s[CR_2(R_1 + R_3) + L] + (R_1 + R_2 + R_3)} V_{in}$$

MATLAB commands

$$V_{out} = \frac{R_3}{Z_1 + R_3} V_{in}$$

$$>> s = tf([1,0],1)$$

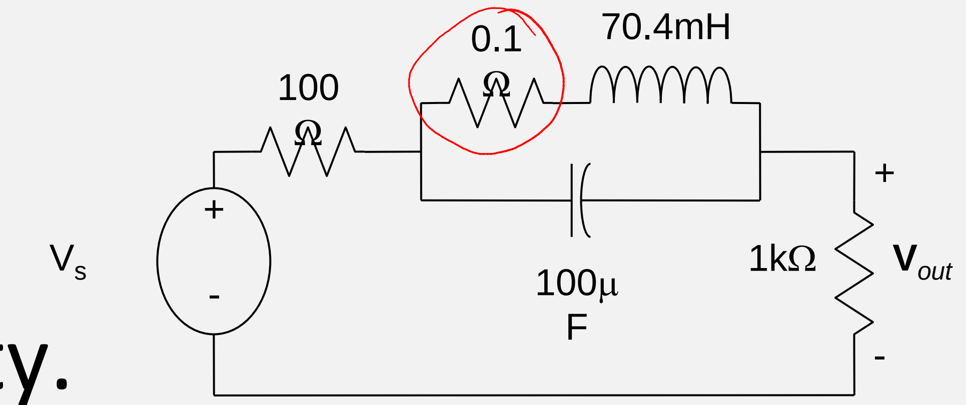
$$>> Z_e = inv\left(1/(s * L + R_2) + sC\right)$$

$$>> Z_1 = R_1 + Z_e$$

$$>> G = \mathrm{minreal}(R_3/(Z_1 + R_3))$$

cancellations

Band-stop or notch filters

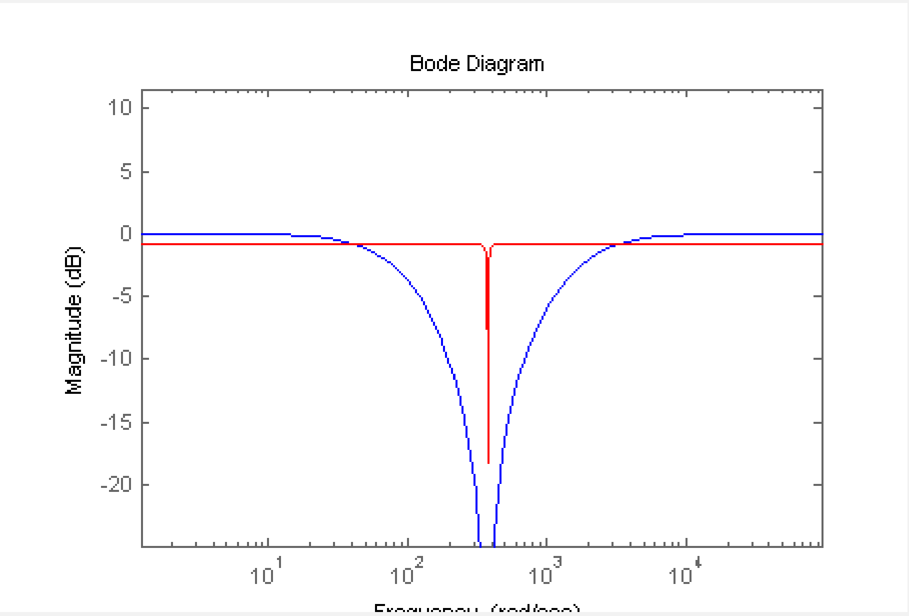


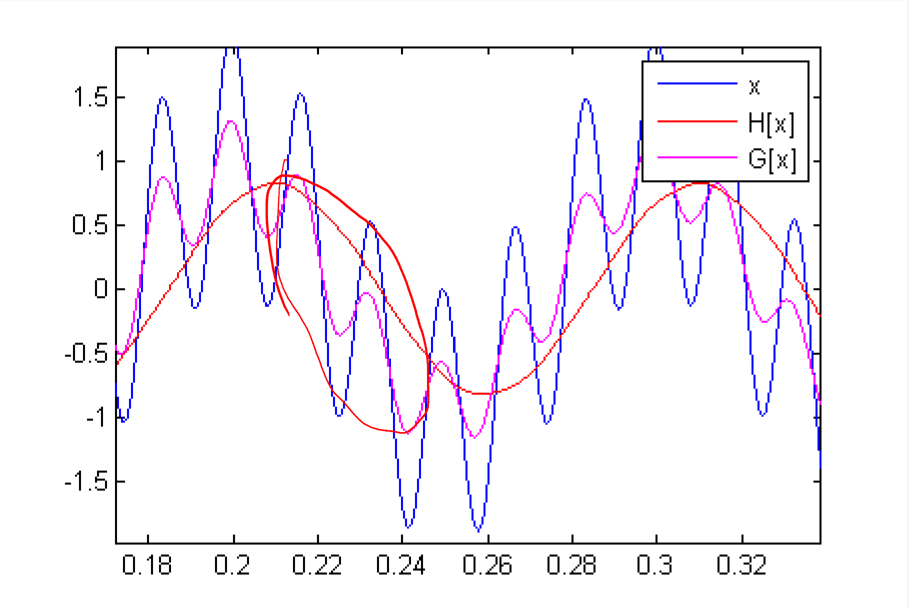
- This filter uses an LC loop to zero-out 60Hz with high selectivity. Its transfer function can be obtained with fairly standard computations (impedance parallel/series combinations)

```
>> L=70.4e-3;R1=100;R2=0.1;R3=1e3;C=100e-6;
>>G=(s*s*L*C*R3+s*C*R2*R3+R3)/(s*s*L*C*(R1+R3)+s*(C*R2*(R1+R3)+L)+R1
      +R2+R3)
```

0.00704 s^2 + 0.01 s + 1000 ~> 60 Hz zero

0.007744 s^2 + 0.0814 s + 1100





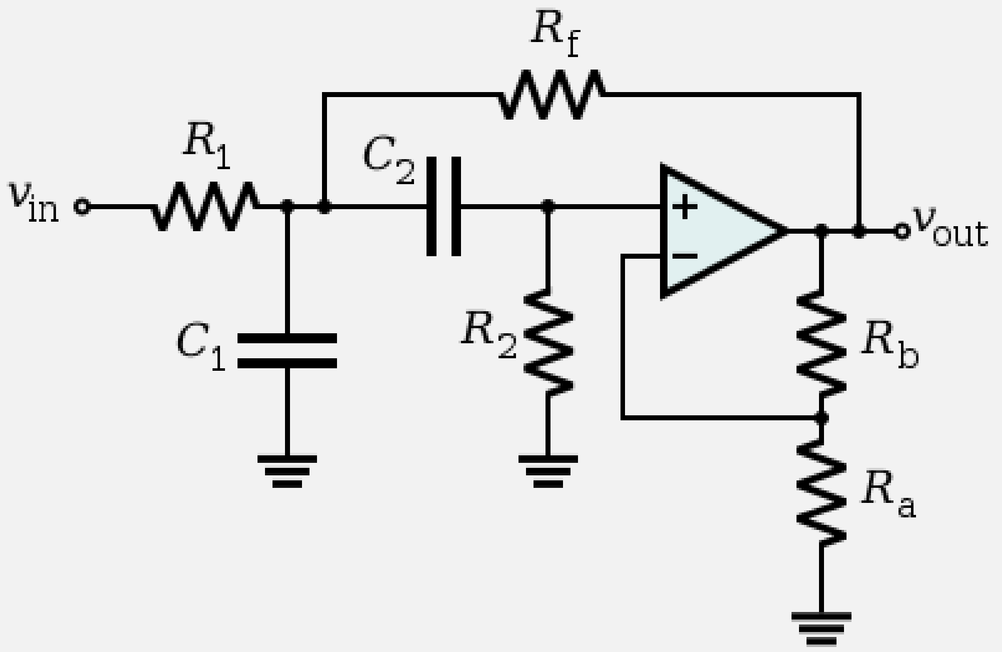
- This filter has only an approximate zero at 60Hz because it includes more realistic resistance values. It has high selectivity due to its underdamped poles. Also note that the filter has poles near the jw-axis so its settling time is large.

Band-pass filters

- Band-pass filters can be implemented with a similar (but non-unity gain) topology as the Sallen-Key.

$$H_{BP}(s) = \frac{\left(1+\frac{R_b}{R_a}\right)\frac{1}{R_1C_1}s}{s^2+\left(1+\frac{1}{R_1C_1}+\frac{1}{R_2C_1}+\frac{1}{R_2C_2}+\frac{R_b}{R_aR_fC_1}\right)s+\frac{R_1+R_f}{R_1R_2R_fC_1C_2}}$$

$$f_{center} = \frac{1}{2\pi}\sqrt{\frac{R_1+R_f}{R_1R_2R_fC_1C_2}}$$



- Band-pass filters find applications in audio, bio-signal analysis, identification-based modeling, to name a few.

EEE304

Lecture 2.4: Discrete-Time Filters

ARIZONA STATE UNIVERSITY

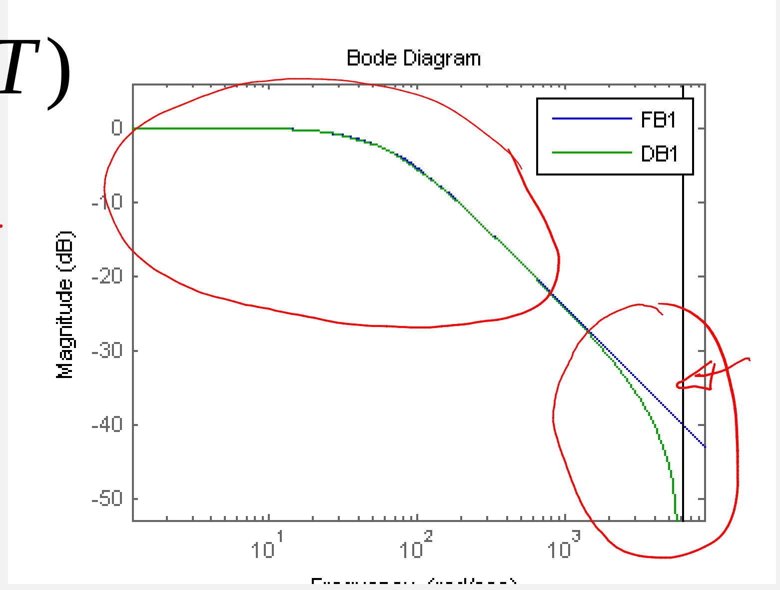
Filtering examples: The DT case

- Consider the problem of separating the high frequency noise above (wn = 100Hz), from the useful signal (ws = 1Hz), with a DT filter.
  - Here we focus only the filtering part and transfer function properties, while further details on sampled-data problems are discussed in subsequent lectures. We assume that the sampling rate is 2kHz so that the discrete frequency is Nyquist Freq. 1 kHz

$$\Omega(rad/sample) = \omega(rad/\sec)T(\sec/sample)$$

  - Let us define a first order analog filter with cutoff frequency 1 dec below the noise (10Hz, 62.8rad/s, for -20dB attenuation, blue trace). The corresponding DT Butterworth would have cutoff 31.4e-3rad/sample and normalized frequency 10Hz/(2kHz/2) = 0.01.

$$FB1 = \frac{1}{0.01592s+1}, \quad DB1 = \frac{0.01547\,z + 0.01547}{z - 0.9691} \quad (\text{Note}: \approx \frac{\lambda}{z-(1-\lambda)}, \lambda = w_c T)$$



  - CT corner frequency = 1/0.0159 = 62.8rad/s.
  - DT pole= 0.969= exp(-62.8/2000) ~ 1 – 0.0314 (z = exp(sT))

Filtering examples: The DT case

- We compare 1st and 2nd order Butterworth filters in terms of noise attenuation and signal distortion. Notice that the DT filter can only attenuate signals up to 1kHz, which is the Nyquist frequency (half the sampling rate).

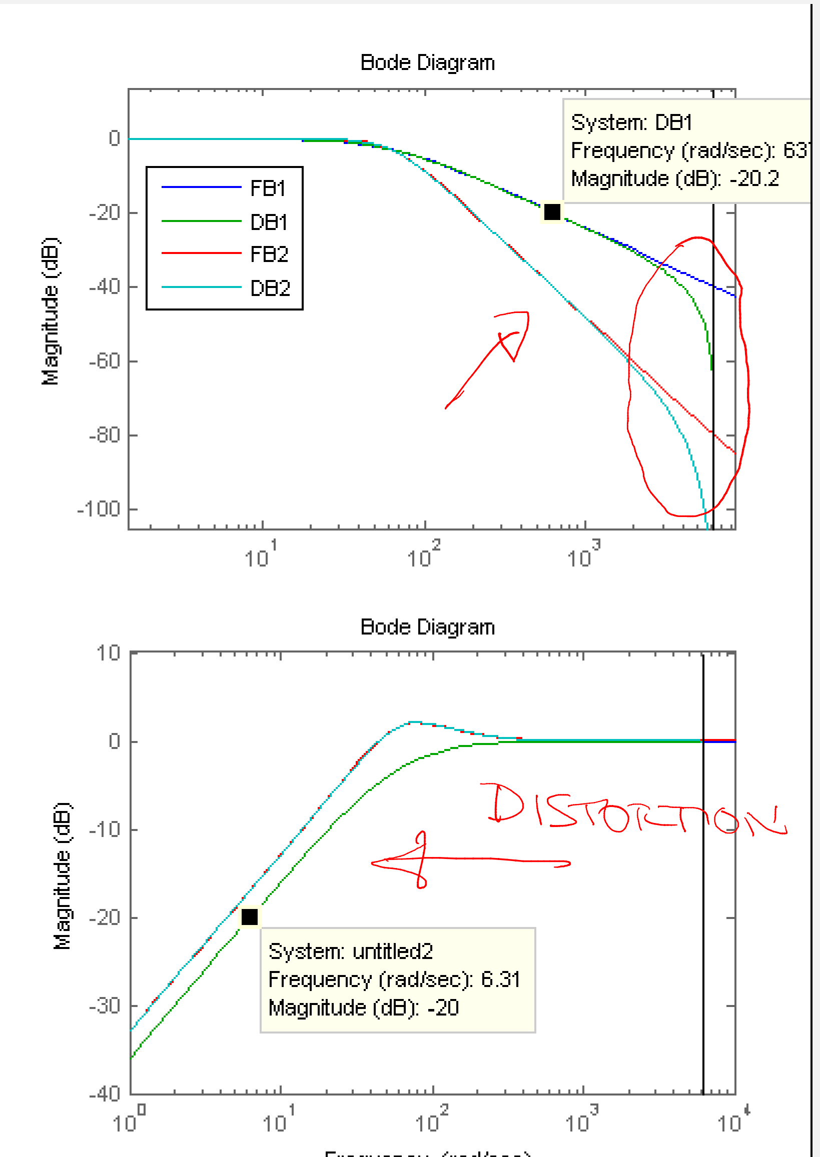
- The corresponding DT Butterworth would have cutoff 62.8/2000 = 31.4e-3rad/sample. Its normalized frequency is 0.01. The corresponding MATLAB command for the Butterworth filter in DT is

```
>> T=1/2000; wc=0.01; n=2
>> [num,den]=butter(2,wc); DB2=tf(num,den,T)
```

```
0.0002414 z^2 + 0.0004827 z + 0.0002414
-------------------------------------------------------
        z^2 - 1.956 z + 0.9565
```

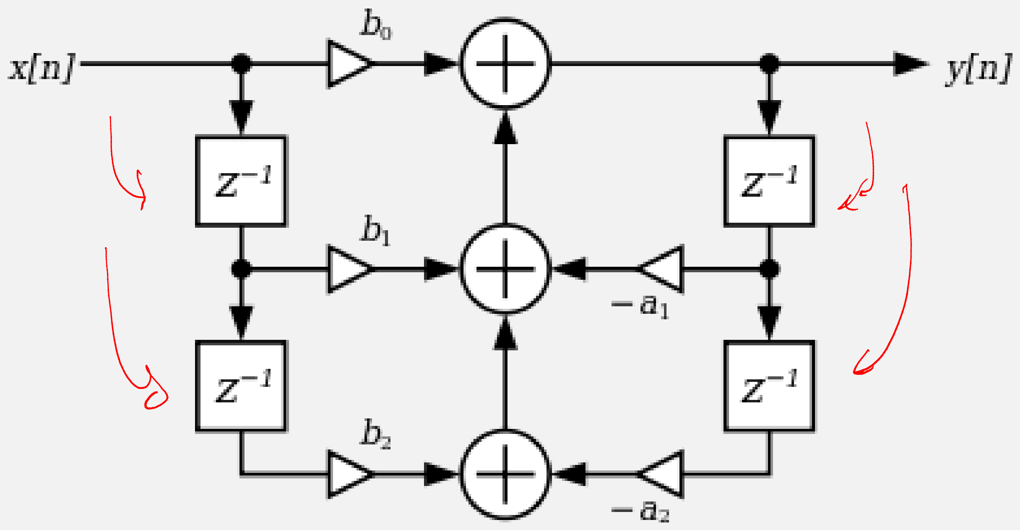
- We evaluate the noise amplitude from the Bode plot of FBn, DBn and the distortion from the Bode plot of (1-FBn), (1-DBn)

(We see more on the “equivalence” between DT and CT filters in the sampling and reconstruction lectures)

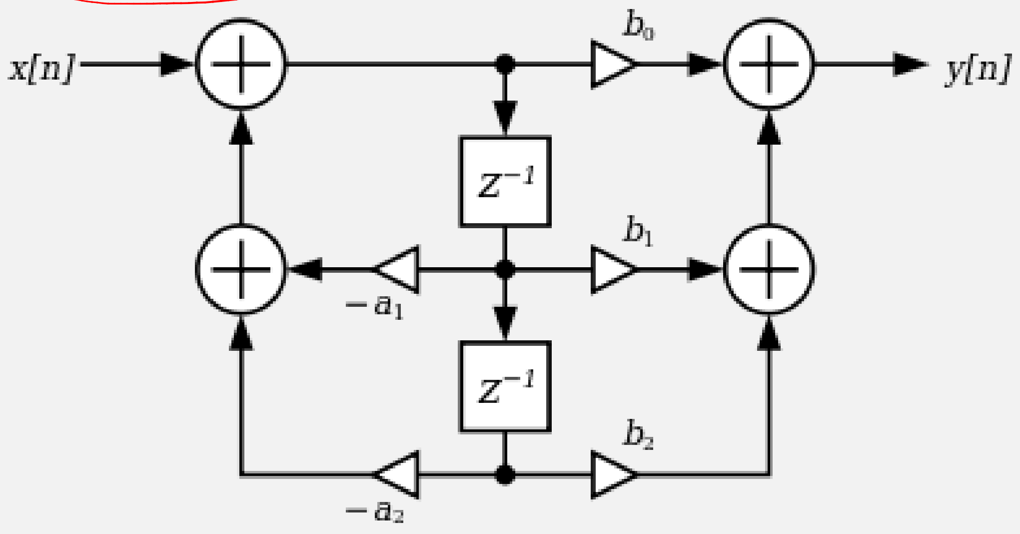


IIR filter structures

- IIR DT filters: $y(n)+\sum_{k=1}^{N} a_k y(n-k)=\sum_{k=0}^{M} b_k x(n-k) \Leftrightarrow H(z)=\dfrac{b_0+b_1 z^{-1}+b_2 z^{-2}+\ldots}{1+a_1 z^{-1}+a_2 z^{-2}+\ldots}$
- Recursive computation of the output, y(n) depends on past I/O samples
  - Easy implementation in terms of “taps” or delays



Direct Form I



Direct Form II

  - Both forms (especially II) can exhibit numerical sensitivity/instability due to quantization and finite precision arithmetic overflow. Usually recommended to keep the order low (<=2). Many alternatives exist in literature, but none is universal.

IIR filter structures

- IIR example: The 8th order Butterworth for cutoff frequency 0.01 is

$$\frac{3.442e-015\,z^8+2.753e-014\,z^7+9.637e-014\,z^6+1.927e-013\,z^5+\ldots+2.753e-014\,z+3.442e-015}{z^8-7.839\,z^7+26.89\,z^6-52.7\,z^5+64.55\,z^4-50.62\,z^3+24.81z^2-6.947\,z+0.8513}$$

- Clearly, this will have implementation problems, even in double precision

- Factoring out the scaling factor in the numerator, we still see the problem in the dynamic range

$$3.4417e-015\frac{z^8+8\,z^7+28\,z^6+56\,z^5+70\,z^4+56\,z^3+28\,z^2+8\,z+1}{z^8-7.839\,z^7+26.89\,z^6-52.7\,z^5+64.55\,z^4-50.62\,z^3+24.81z^2-6.947\,z+0.8513}$$

- But, analyzing in terms of factors, we can have a reasonable implementation as a cascade of four 2nd order filters (still need to make sure the quantization of coefficients is accurate enough)

$$\frac{[2.4221e-004(z+1)^2]^4}{(z-0.9649)\,(z-0.9807)\,(z^2-1.94z+0.9416)\,(z^2-1.965z+0.9664)\,(z^2-1.988z+0.9887)}$$

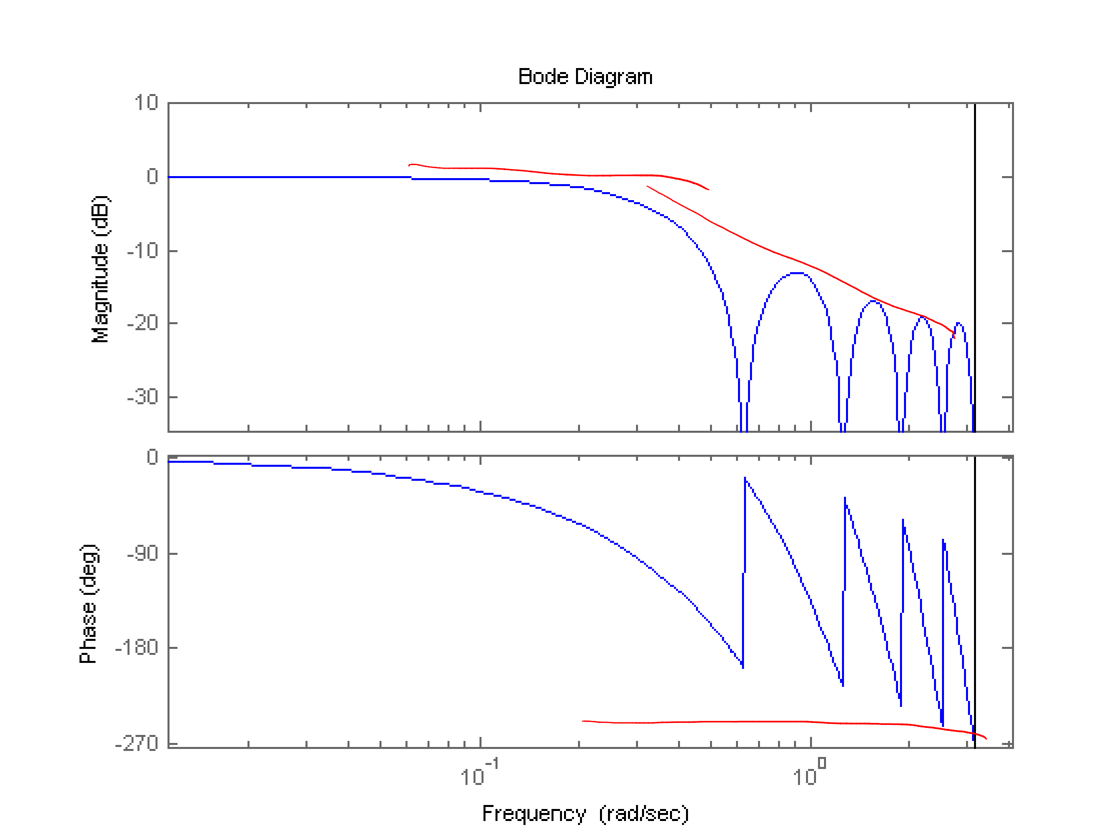
FIR filter structures

- FIR DT filters: $y(n) = \sum_{k=0}^{N} h(n-k)x(k)$
- DSP's: efficient implementation of filters with applications in inversion, equalization, noise-cancellation problems.
  - Interesting problems associated with this implementation is the approximation in the CT to DT (C2D) conversion (including sampling frequency, order, and quantization selection).
- Advantages: ability to implement "arbitrary" filters but may require high order, easier to get "linear phase" (constant group delay), well-suited for finite precision arithmetic, always stable.
- Disadvantages: unsuitable for long IR, may have excessive memory requirements and require large number of computations (but parallelizable).

FIR example

- A classic example of a simple FIR implementation
  - Choose $h(n) = \frac{1}{N}; \quad n = 1,\ldots,N$ (0 otherwise)
  - This is a simple averaging filter over a rolling window. It has low pass properties.

$$H(z) = \frac{1}{N}\left(z^{-1} + z^{-2} + \ldots + z^{-N}\right); \quad n = 1,\ldots,N$$



  - Other widely used FIR filters are the so-called EWMA (exponentially-weighted, moving-average) where old samples are de-weighted exponentially.

For example, $h(n) = \frac{\lambda^n}{M}; \quad n = 1,\ldots,N$, approximating a recursive lowpass filter, but with FIR properties.

FIR note: Group delay

- FIR filters are noted for their convenience in designing filters with constant "group delay", defined as $\tau(\Omega) = -\frac{d}{d\Omega}\left\{\angle H(e^{j\Omega})\right\}$

  Recalling the LTI response to sinusoids and expressing the phase shift as a delay

  $$y(n) = |H(e^{j\Omega})|\cos(\Omega n + \angle H(e^{j\Omega})) = |H(e^{j\Omega})|\cos\left(\Omega\left[n + \frac{\angle H(e^{j\Omega})}{\Omega}\right]\right)$$

  we observe that frequencies around a constant group delay exhibit the same delay at the output of the filter (similarly for CT filters). This is particularly important in communication systems.

  Filters with constant group delay are also referred to as "linear phase" systems. Cascades of linear phase systems are also linear phase. A nontrivial example of a linear-phase system is the running average $H(z) = \frac{1}{2}(z^{-1} + z^{-2})$

  The phase of H is $\angle H(e^{j\Omega}) = \angle e^{j2\Omega} + \angle(z+1) = \boxed{2\Omega + \tan^{-1}\frac{\sin\Omega}{1+\cos\Omega}}$ $\quad \frac{d}{d\Omega}(\cdot) = \text{constant.}$

  Differentiating and after calculations it follows that the phase is constant

FIR example

- An example of a low pass filter FIR implementation
  - Choose $h(t) = \dfrac{\sin 10t}{\pi t}; \quad t = -\pi : 0.01 : \pi$

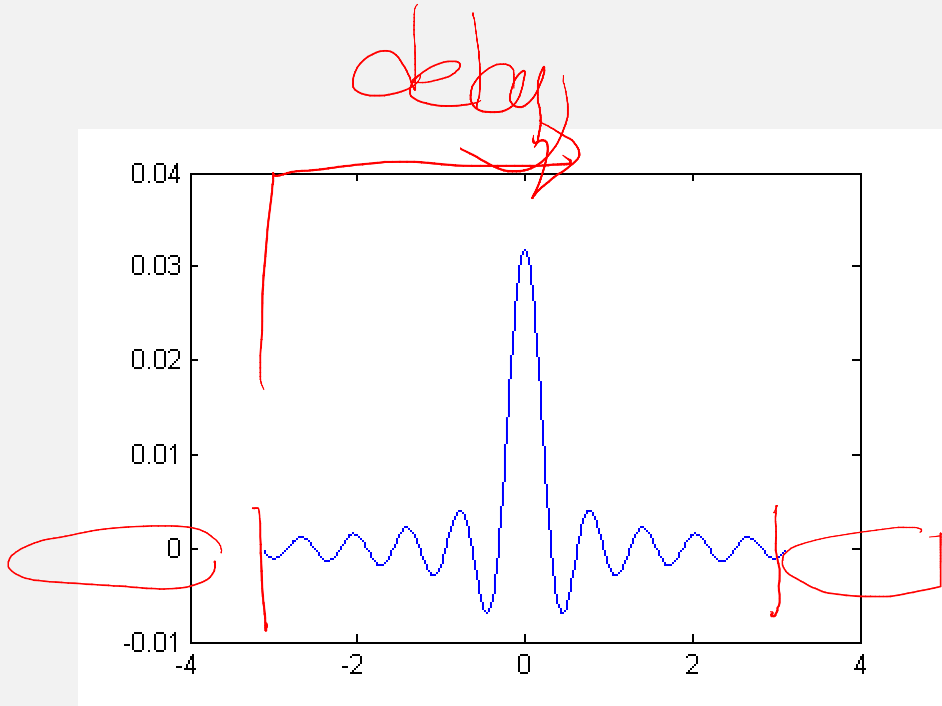
(from the truncation of the ideal low pass IR)

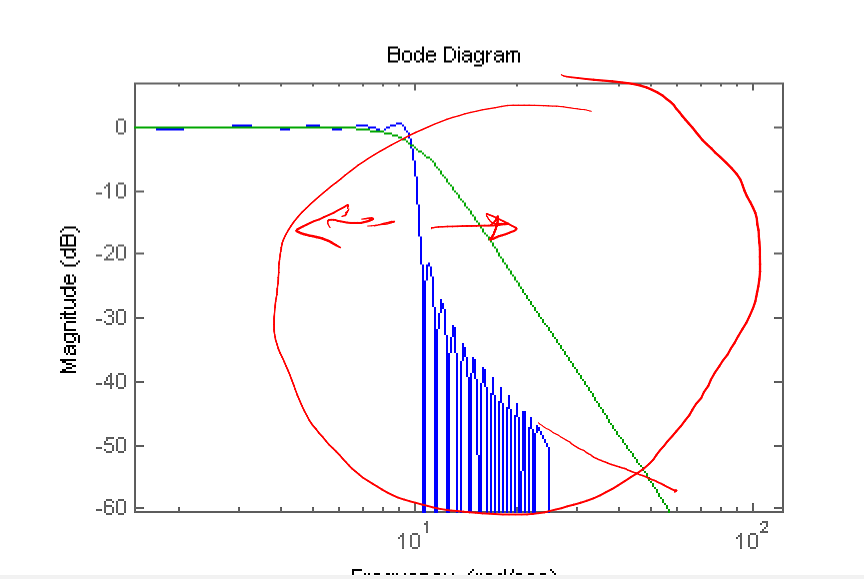
- Define the FIR filter with num = h, den = z^N, N=629, H=tf(num,den,Ts). This also implies the addition of a delay of N/2 samples to make the filter causal.

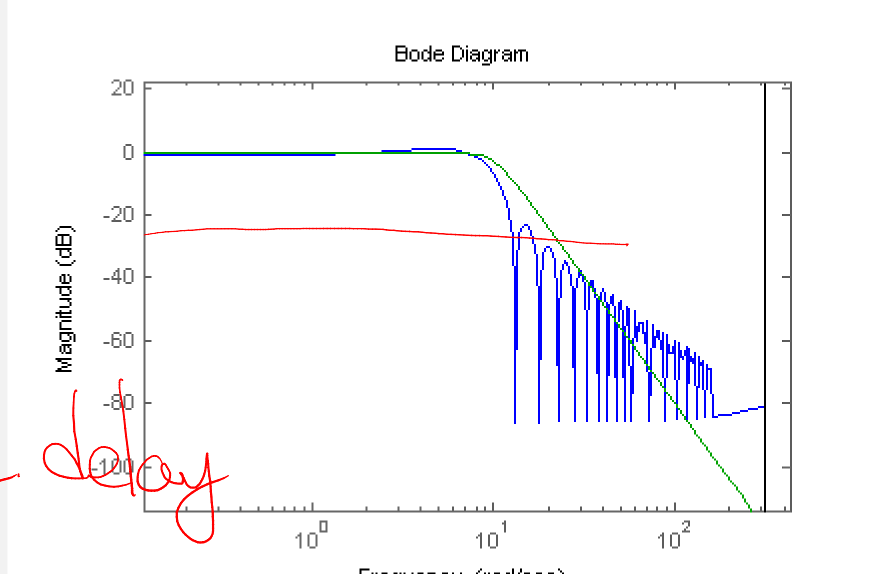
- The figure shows its frequency response, compared to a 4th order Butterworth filter

- To reduce the order, we try $t = -\pi/5 : 0.01 : \pi/5$. Now N=126 but the filter response has deteriorated.

0.63 sec delay

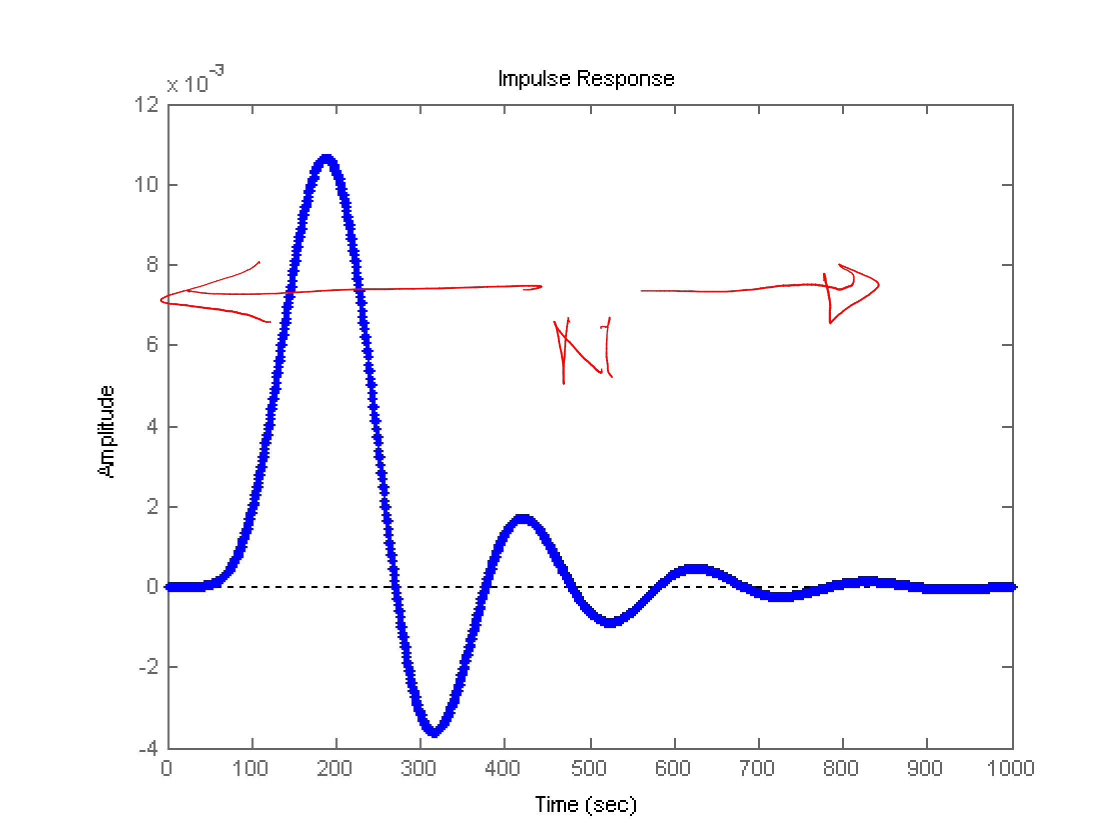


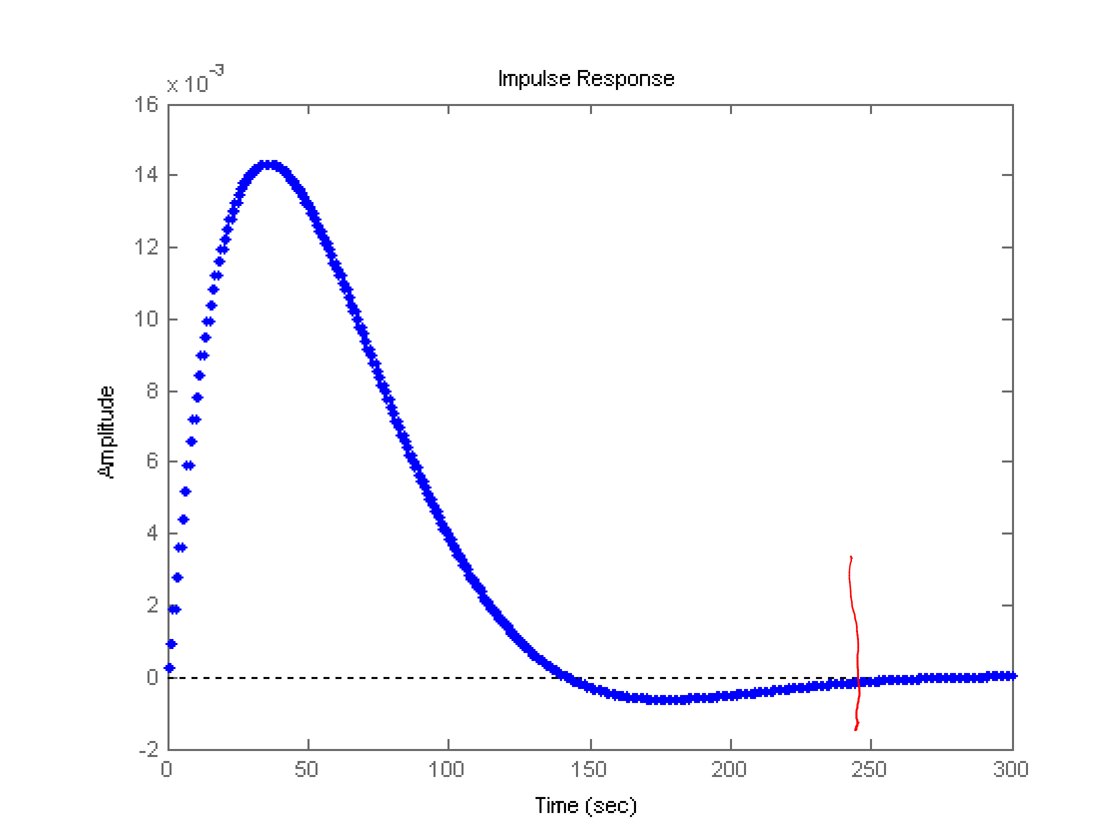




FIR example

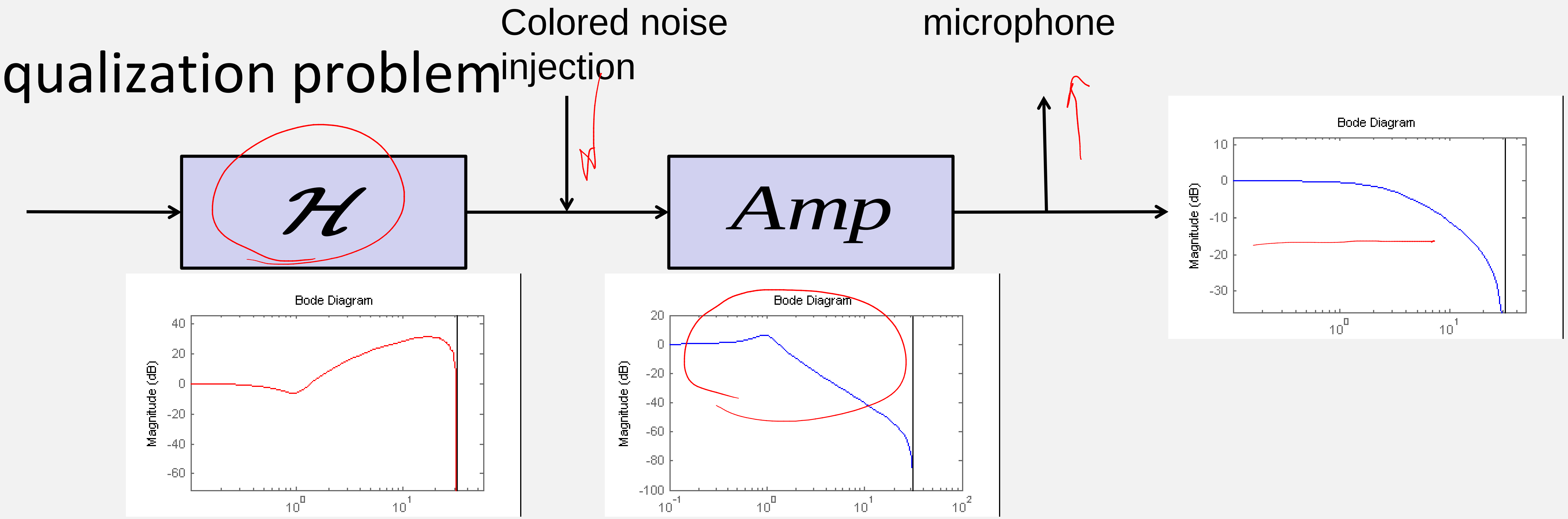
- Let us also consider the 8th order Butterworth filter and attempt an FIR implementation.
- The IR coefficients have much more reasonable size but we still need about 1000 terms
- Even the second order Butterworth would require about 300 terms for its implementation.
  - Here, IIR implementations have a clear advantage, but there are other methods (e.g., Parks-McClellan) to compute an optimal FIR implementation of a giver IR





Equalization application

- The equalization problem



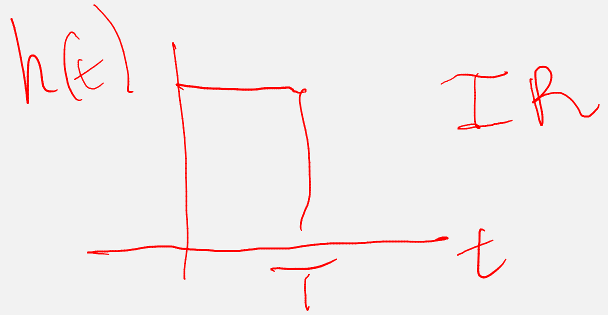
- An Amplifier response is to be "shaped" to correct for underdamped behavior and reduced bandwidth.

- An "equalizer" (or "prefilter") $\mathcal{H}$ is used to adjust the signal sent to the amplifier so that the output has critically damped characteristics and higher bandwidth. The Amplifier model could be obtained by injecting colored noise and reading the output with a microphone. The prefilter can be implemented as a high order FIR approximation using a DSP, that can be adjusted on-demand by the user.

Other filter structures: ZOH

- Zero Order Hold: It is a CT filter that arises in ADC/DAC operations and holds the value constant for a given interval T. Leaving aside the sampling operation, the transfer function of this filter is

$$H(s) = \frac{1 - e^{-sT}}{sT}$$



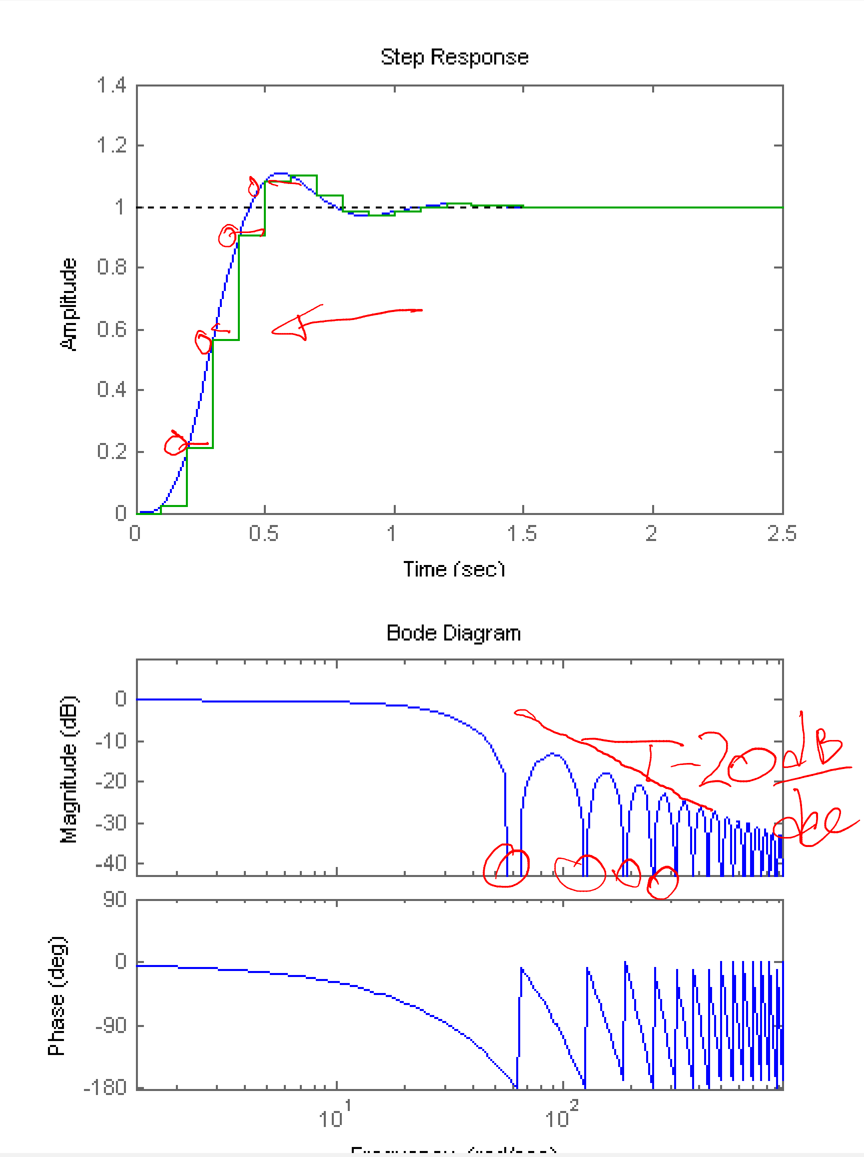
- The filter is stable (it has no finite poles since the root of the denominator at 0 is canceled by the numerator zero)

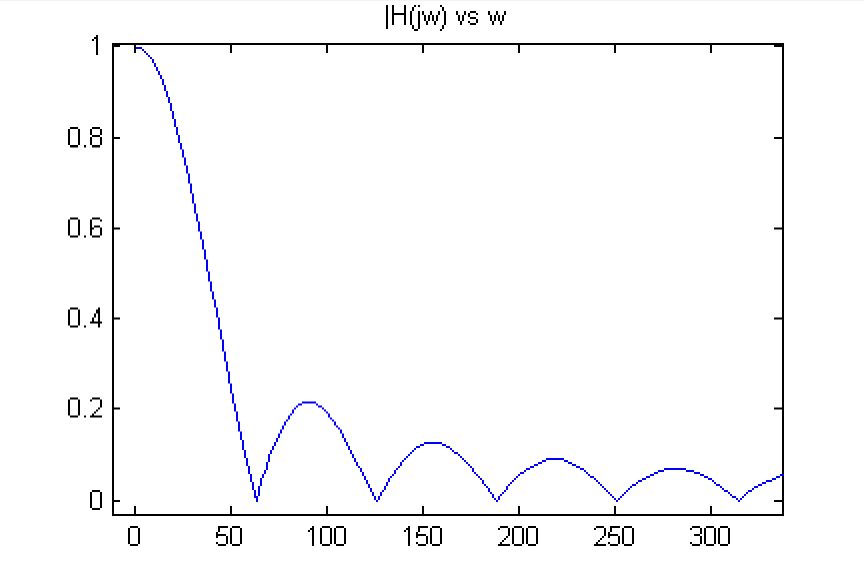
- Its frequency response is 1 at w = 0, has zeros at

$$\cos\omega T = 1 \Rightarrow \omega = 2\pi n / T,\, n = \pm 1, \pm 2 \ldots$$

- The filter rolls-off with -20dB/dec, so it has similar properties to a 1st order lowpass filter.

- It has phase like a ½-sample delay $\sim e^{-j\omega T/2}$





EEE304

Lecture 2.5: Filter Output Computations

ARIZONA STATE UNIVERSITY

Filter output computations

- The recurring theme in the filtering problems is:
  - The computation of the filter transfer function
  - The evaluation of the transfer function at the required frequencies
  - Depending on the problem statement, this may lead to an optimization problem or an interpolation problem
- Transient computation problems are also of interest but the typical practical application makes use of computational software so their analytical evaluation is infrequent. The same is true for the frequency response of filters, although in this case, the designer should have a more detailed insight to expedite the design and debugging processes.

Filter output computations: Steady-state response

- One special problem arising in filtering problems is the computation of the “steady-state” response.

- The solution of this problem can be greatly simplified if the filter is stable, whereby any transient terms vanish asymptotically. Then, the final answer follows the simpler Fourier transform computations. In addition, assuming that the transfer functions have real coefficients, the following formulae can be used:

- Continuous Time

$$x(t) = e^{j\omega_o t} \Rightarrow y(t) = H(j\omega_o)e^{j\omega_o t}$$

$$x(t) = \cos(\omega_o t) \Rightarrow y(t) = |H(j\omega_o)|\cos(\omega_o t + \angle H(j\omega_o))$$

- Discrete Time

$$x(n) = e^{j\Omega_o n} \Rightarrow y(n) = H(e^{j\Omega_o})e^{j\Omega_o n}$$

$$x(n) = \cos(\Omega_o n) \Rightarrow y(n) = |H(e^{j\Omega_o})|\cos(\Omega_o n + \angle H(e^{j\Omega_o}))$$

Filter output computations: St.st. response

- CT Example 1: Consider the causal filter described by the differential equation $\ddot{y} + 2\dot{y} + 4y = -0.4\dot{x} + 4x$

  3

  Compute the steady state response to $x(t) = \boxed{\cos(2t)}u(t-10) + \cancel{e^{-2t}u(t)}$

  1   1

- The filter transfer function is: $H(s) = \dfrac{-0.4s + 4}{s^2 + 2s + 4}$

  2

  The filter is stable since the poles (-1 +/-1.73j) have negative real parts. Hence, the steady-state response is well defined and can be computed for the steady-state part of the input $\boxed{x_{ss}(t) = \cos(2t)}$

  Im

$$y_{ss}(t) = |H(j\omega_o)|\cos(\omega_o t + \angle H(j\omega_o)) = \frac{\sqrt{4^2 + (-0.4\times 2)^2}}{\sqrt{(-2^2+4)^2 + (4)^2}}\cos(2t + \tan^{-1}\frac{-0.8}{4} - 90^o)$$

rad   Re   Re=0

$$= 1.019\cos(2t - 101.3^o)$$

deg or rad

Filter output computations: St.st. response

- CT Example 2: Consider the causal filter described by the differential equation $\ddot{y} + 2\dot{y} + 4y = -0.4\dot{x} + 4x$

  Compute the steady state response to $x(t) = \cos(20t)u(t-10) + 10u(t)$

  1 1

- The filter transfer function is: $H(s) = \dfrac{-0.4s + 4}{s^2 + 2s + 4}$

  The filter is stable and the steady-state response is well defined and can be computed for the steady-state part of the input $x_{ss}(t) = \cos(20t) + 10$

$$y_{ss}(t) = |H(j\omega_o)| \cos(\omega_o t + \angle H(j\omega_o)) + H(0)(10)$$

4 DC-gain

$$= \frac{\sqrt{4^2 + (-0.4 \times 20)^2}}{\sqrt{(-20^2 + 4)^2 + (40)^2}} \cos(20t + \tan^{-1}\tfrac{-8}{4} - \tan^{-1}\tfrac{40}{-196} + 180^o) + 10 = \boxed{0.0225\cos(20t + 122.3^o)} + \boxed{10}$$

4 abs deg 4 Re(den) < 0 !

For MATLAB verification, use >> [m,p]=bode(H,20) → freq.

Filter output computations: St.st. response

- CT Example 3: Consider the causal filter described by the differential equation $\ddot{y} - 2\dot{y} + 4y = -0.4\dot{x} + 4x$

  Compute the steady state response to $x(t) = \cos(20t)u(t-10) + 10u(t)$

- The filter transfer function is: $H(s) = \dfrac{-0.4s + 4}{s^2 - 2s + 4}$

  The filter is not stable since the poles of the transfer function are (1 +/- 1.73j) and have positive real parts. Hence, the steady-state is not well defined.

Filter output computations: St.st. response

- DT Example 1: Consider the causal filter described by the difference equation $y[n]=\frac{1}{2}y[n-1]+\frac{1}{4}x[n-1]$

  Compute the steady state response to $x(n)=\sin\left(\frac{n\pi}{16}\right)u(n-10)$

- The filter transfer function is: $H(z)=\frac{\frac{1}{4}z^{-1}}{1-\frac{1}{2}z^{-1}}=\frac{\frac{1}{4}}{z-\frac{1}{2}}$

  The filter is stable since the pole (1/2) has magnitude less than one. Hence, the steady-state response is well defined and

$$y_{ss}(n)=\left|H\left(e^{j\Omega}\right)\right|\sin\left(\Omega n+\angle H\left(e^{j\Omega}\right)\right),$$

$$\Omega=\frac{\pi}{16}$$

$e^{j\Omega}=\cos\Omega+j\sin\Omega$

Im

$$y_{ss}(n)=\frac{\frac{1}{4}}{\sqrt{\left\{\left[\cos\left(\frac{\pi}{16}\right)-\frac{1}{2}\right]^2+\left[\sin\left(\frac{\pi}{16}\right)\right]^2\right\}}}\sin\left(\frac{\pi}{16}n-\operatorname{atan}\frac{\sin\left(\frac{\pi}{16}\right)}{\cos\left(\frac{\pi}{16}\right)-\frac{1}{2}}\right)=0.482\sin\left(\frac{\pi}{16}n-22.1^o\right)$$

Re could be -ve

Filter output computations: St.st. response

- DT Example 2: Consider the causal filter described by the difference equation $y[n]=\frac{1}{2}y[n-1]+\frac{1}{4}x[n-1]-\frac{1}{4}x[n-2]$

  Compute the steady state response to $x(n)=\cos\left(\frac{n\pi}{2}\right)u(n-10)+u(n)$

  Ω_o 1 1

- The filter transfer function is: $H(z)=\frac{1/4\, z^{-1}-1/4\, z^{-2}}{1-1/2\, z^{-1}}=\frac{z-1}{4z(z-0.5)}$

  The filter is stable since the poles (1/2, 0) have magnitude less than one. Hence, the steady-state response is well defined and

DC gain

$z=e^{j\Omega_o}=\cos\Omega_o+j\sin\Omega_o$

$$y_{ss}(n)=|H(e^{j\Omega_o})|\cos(\Omega_o n+\angle H(e^{j\Omega_o}))+H(e^{j0})(1);\qquad \Omega_o=\pi/2$$

$\angle 1/2$

$|z|$

$$=\frac{\sqrt{(\cos\Omega_o-1)^2+\sin^2\Omega_o}}{(4)(1)\sqrt{(\cos\Omega_o-0.5)^2+\sin^2\Omega_o}}\cos\left(\tfrac{\pi}{2}n+\tan^{-1}\tfrac{1}{-1}+180^o-\tfrac{\pi}{2}-\tan^{-1}\tfrac{1}{-0.5}+180^o\right)+(0)(1)$$

Re < 0 Re < 0

$$=0.316\cos\left(\tfrac{\pi}{2}n-71.6^o\right)$$

```
>> H=tf([1 -1],[1 -.5 0]*4,1)
>> [m,p]=bode(H,pi/2)
```

system

freq. resp.

EEE304

Lecture 2.6: Filter computations: Bode Plots

ARIZONA STATE UNIVERSITY

Computations for Bode plots (CT)

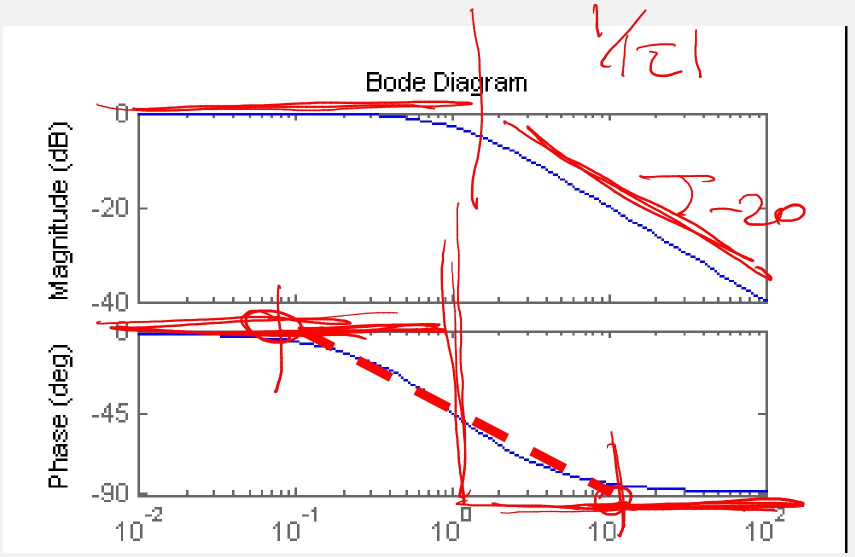
- Factorize numerator and denominator to 1st or 2nd order terms.
- The more convenient form is the time-constant $(\tau s+1)$. Then the magnitude/phase for the term is

Re = 1

$$(\tau j\omega+1)=\sqrt{\tau^2\omega^2+1}\,\angle\tan^{-1}(\tau\omega);\quad \frac{1}{(\tau j\omega+1)}=\frac{1}{\sqrt{\tau^2\omega^2+1}}\angle-\tan^{-1}(\tau\omega)$$

- All the magnitudes are 1 (0dB) at low frequencies and no phase correction is needed since the real part is positive.

CORNER FREQ



- Low frequency asymptote: 0dB, 0deg
- High frequency asymptote: +/-20dB/dec, +/-90deg
- Interpolate phase with a straight line 1 dec above and below

Computations for Bode plots (CT)

- Zeros or Poles at the origin $(s, \quad 1/s)$: No 0dB asymptote. Both go through 0dB at 1rad/s.

- Quadratics, underdamped:

$\zeta < 1$

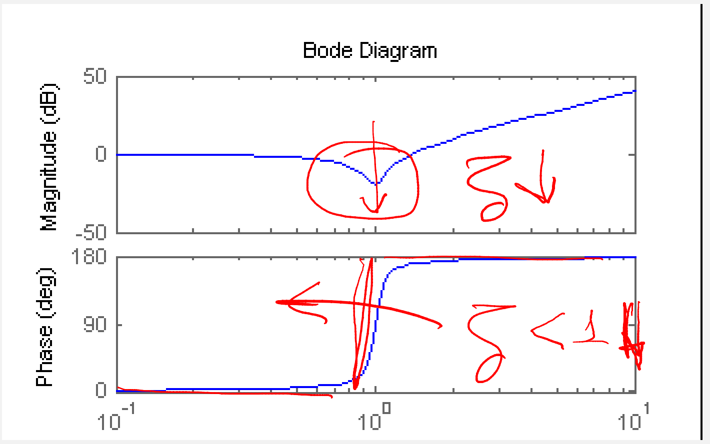
quadrant correction

$$(s^2 + 2\zeta\omega_0 s + \omega_0^{\,2})\Big|_{s=j\omega} = \sqrt{(\omega_0^{\,2} - \omega^2)^2 + 4\zeta^2\omega_0^{\,2}\omega^2}\ \angle \tan^{-1}\left(\frac{2\zeta\omega_0\omega}{(\omega_0^{\,2} - \omega^2)}\right) + 180u(\omega^2 - \omega_0^{\,2})$$

- 180 deg phase correction is needed when the real part is negative.

- Low frequency asymptote: 20log10($\omega_0^{\,2}$)dB, 0deg

- High frequency asymptote: 40dB/dec, 180deg



- Magnitude dipping (or peaking for a denominator) and phase interpolation depends on damping ratio (TF shown: $s^2 + 0.1s + 1$).

Computations for Bode plots (DT)

- Factorize numerator and denominator to 1st or 2nd order terms.

- All forms yield similar complexity. Assume $(z-a)$. Then the magnitude/phase for the term is

$z = e^{j\Omega}$

Im

$$(e^{j\Omega}-a)=\sqrt{(\cos\Omega-a)^2+\sin^2\Omega}\,\angle\tan^{-1}\left(\frac{\sin\Omega}{\cos\Omega-a}\right)+180u(a-\cos\Omega)$$

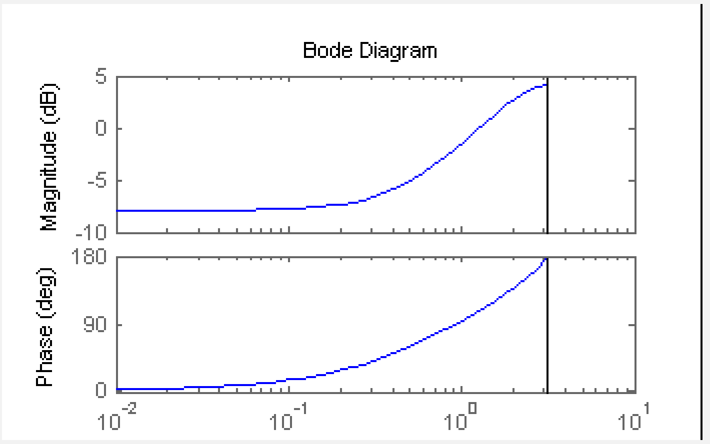
Re

quadrant correction

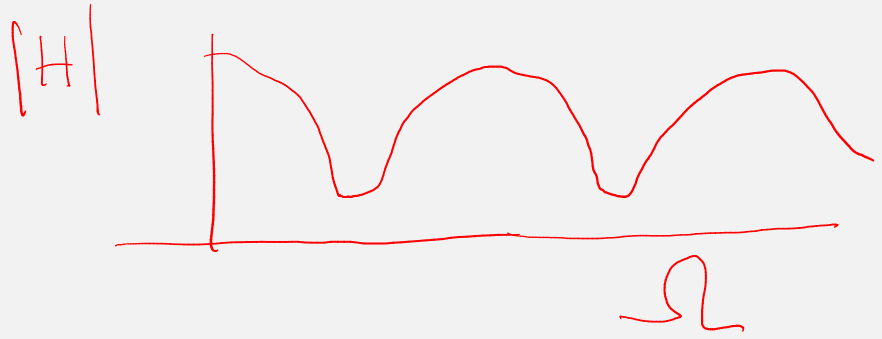
- Low frequency asymptote: 20log10(|1-*a*|)dB, 0/180deg

$z = e^{j0} = 1$

- High frequency asymptote: none (TF shown: z-0.6)



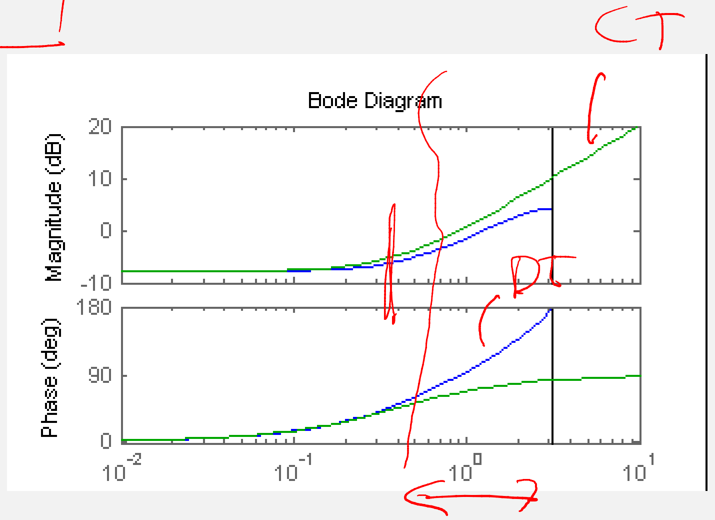
TF is periodic in Ω, with period 2π



Computations for Bode plots (DT)

- An approximate asymptote to help visualize the DT Bode plot can be obtained by converting to an “equivalent” CT system. (More details on DT-CT equivalence are discussed in subsequent lectures.)
- We use the (so called “Forward Euler”) transformation $z = e^{sT} \cong 1 + sT; \; T = 1$
- Then the CT Bode plot approximates the DT plot for $\boxed{(z-a), a > 0}$, and frequencies well-below Nyquist (3.14rad/sec), e.g., 0.3rad/sec

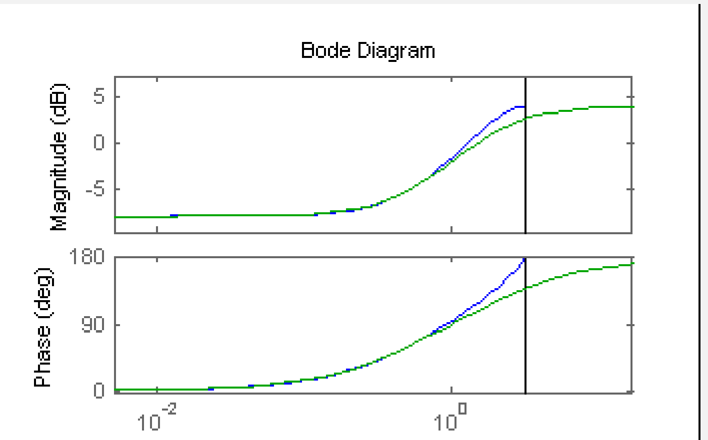
(T=1)



- E.g., $(z-0.6) \rightarrow (1+s-0.6) = (s+0.4)$

→ 1+s

- The high frequency asymptote will provide a better approximation as the DT corner frequency approaches 1 (slower poles). Other transformations may yield better approximations, e.g., Tustin: $z = \dfrac{e^{sT/2}}{e^{-sT/2}} \cong \dfrac{1 + sT/2}{1 - sT/2}$



Computations for Bode plots (DT)

$$z = e^{sT} \cong 1 + sT; \; T = 1$$

- Example: Visualize the frequency response of the system

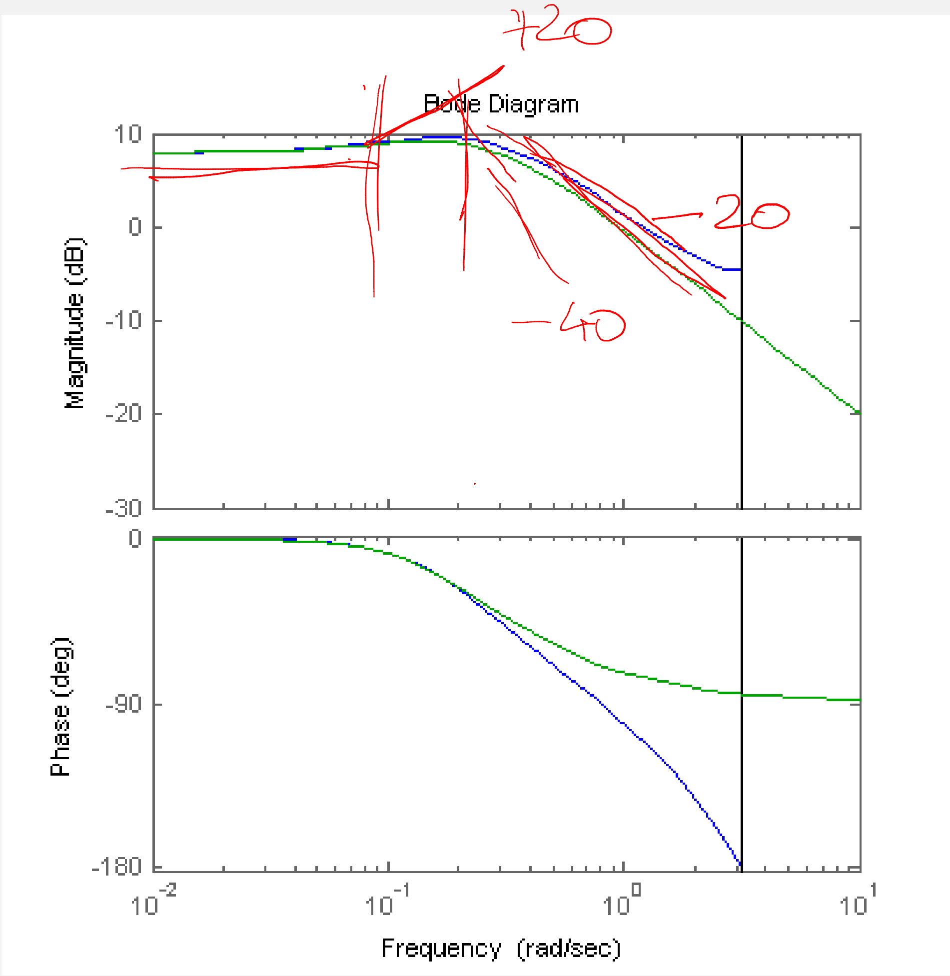
$$H(z) = \frac{z - 0.9}{(z - 0.8)^2}$$

- Using “Forward Euler” $z = 1 + s$

$$H_c(s) = \frac{s + 1 - 0.9}{(s + 1 - 0.8)^2} = \frac{s + 0.1}{(s + 0.2)^2}$$

CORNER FREQ

- Its frequency response has corner frequencies at 0.1 (zero), 0.2 (two poles) which are well-below the Nyquist frequency 3.14. We therefore expect a reasonable approximation of the DT system frequency response by the equivalent CT system up to 0.3 frequency (barely able to see the asymptotes.



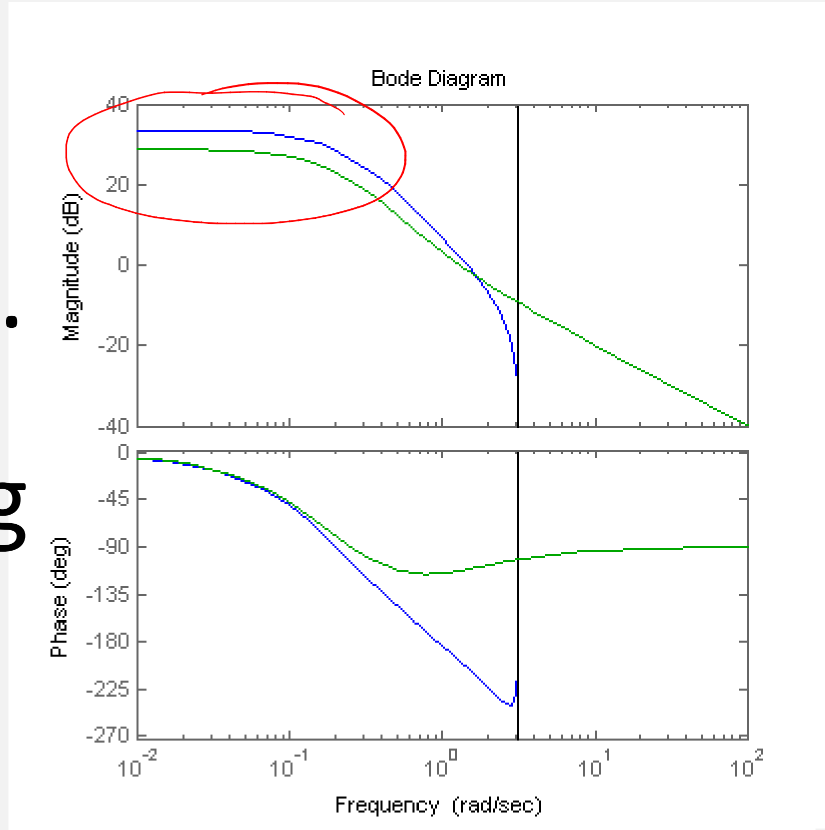
$z = e^{sT} \cong 1 + sT;\ T = 1$

Computations for Bode plots (DT)

- Example: Visualize the frequency response of the system $H(z) = \dfrac{z+0.9}{(z-0.8)^2}$

  Using “Forward Euler” $H_c(s) = \dfrac{s+1+0.9}{(s+1-0.8)^2} = \dfrac{s+1.1}{(s+0.2)^2}$

  This system has a “ringing zero” and the approximation fails.



- Even more pronounced differences are obtained for “ringing poles”, e.g., $H(z) = \dfrac{z-0.9}{(z-0.8)(z+0.8)}$

  Now $H_c(s) = \dfrac{s+1-0.9}{(s+1-0.8)(s+1+0.8)} = \dfrac{s+0.1}{(s+0.2)(s+1.8)}$

  but the two systems have fairly different frequency responses and dramatically different step responses.