Overview

- In the previous lecture, we have learned the key concepts in signal and signal processing.

- In this lecture, we will look at the basics of low pass filter. We expect that you can design simple analog filters and simple software digital filters using MATLAB in EWS

Frequency Domain View of Signals

Fourier discovered that any wave form can be represented as sum of sinusoids.

Q1: what does bandwidth mean?

Q2: Which figure represents a periodic signal and which represents an non-periodic signal (also called energy signal)?

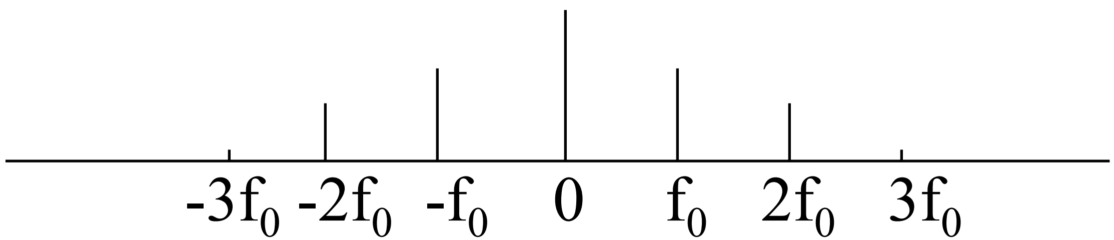


Figure 1

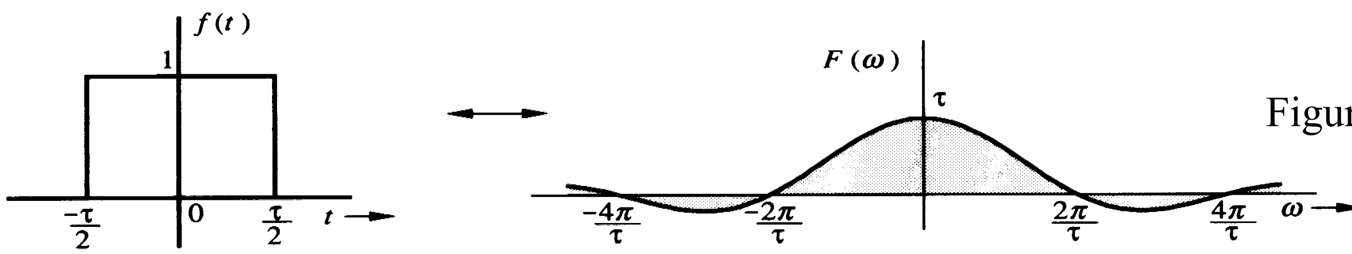


Figure 2

Review Sampling

- According to Nyquist, how fast should you sample? What is the rule of thumb in practice?

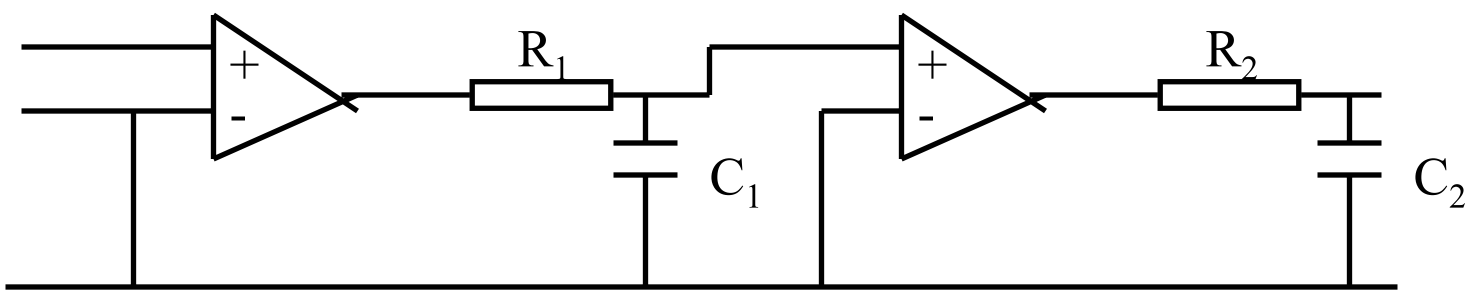
- If you have signal ranges from 1 - 10 Hz and noise ranges from 50 to 1000 Hz, what is the correct practical lower bound in sampling rate if you decide to sample and the filter the noise out from the sampled data?

- A. 20 Hz
- B. 60 Hz
- C. 150 Hz
- D. 3000 Hz
- E. 5000 Hz

- What should you do if you only have a standard PC?

Basic Signal Processing Concepts:
Low Pass filters - 1

Signals often contaminated by high frequency noises that need to be removed. The following is a simple two stage active filter.

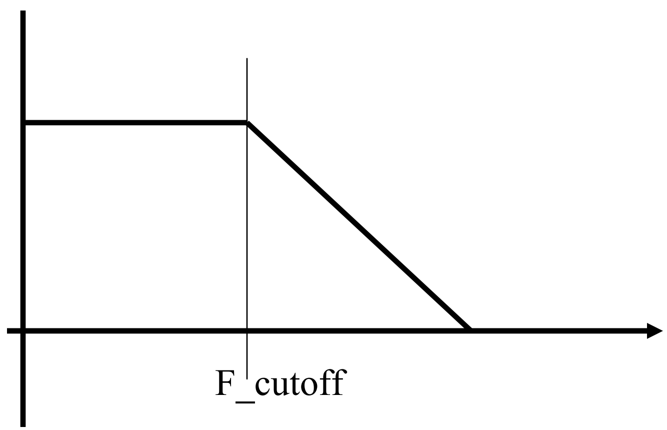


The Op-Am isolates the interactions between two RC circuits.

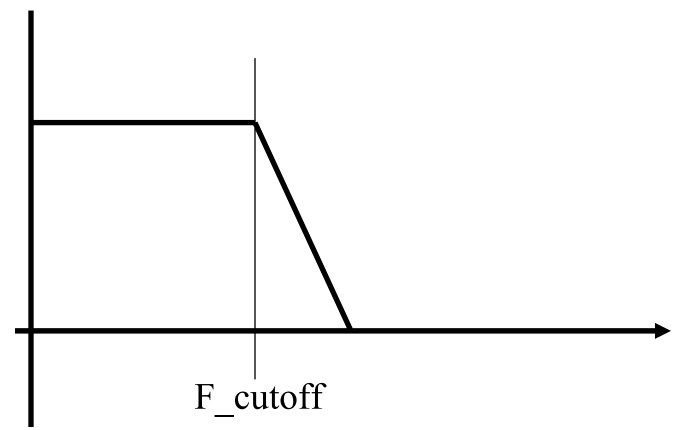
This allows a simple analysis. You can get such filters on a chip

Basic Signal Processing Concepts: Low Pass filters - 2

- The more are the stages, the sharper is the rate of reduction of the signal after the cut off frequency. If you want sharper rate of reduction, you need more stages. (In filter literature, the number of stages is called the order of the filter.)



1^{st} order



2^{nd} order

Basic Signal Process Concepts: Low Pass Filters 3

The magnitude attenuation and the phase delay of a N stage filter illustrated in page 4 are as follows, where $\omega_{cutoff} = R_1C_1 = R_2C_{2.} = \ldots.$

$$|a(\omega)| = |a_0|\left(\frac{1}{\sqrt{1+(\frac{\omega}{\omega\_cutoff})^2}}\right)^N$$

$$\theta(\omega) = -N(\arctan\frac{\omega}{\omega\_cutoff})$$

The formula for the commonly used Butterworth filter is SQRT(1 /(1 + (w/w_cutoff)^{2N}))

Application Notes

We have two control variables, the cutoff frequency and the order of the filter.

1. The lower the cutoff frequency, the more effective is the filtering. But if the cutoff frequency is too close to the useful signal, the signal will also be reduced.

2. The higher the order, the more powerful is the filter. But it also introduce more phase delays (the shift of t he wave to the right on the time line)

The design of a filter is iterative. You have to adjust the cutoff frequency and the order of the filter until the requirements are met.

Class Exercise

- The frequencies of interest in the signal are from 10 to 60 Hz. There are serious noises with frequencies in the range 500 - 1000 Hz seen on the scope. The simple anti-aliasing filter (pp 4 & 6) is used to filter out the high frequency noises.

- Suppose that we pick cutoff frequency at 100 Hz, what should be the order (stages) of the filter so that the signal will be reduced no more that 30% while the noise will be reduced at least 96%?

Class Exercise

- The frequencies of interest in the signal are from 10 to 60 Hz. There are serious noises with frequencies in the range 500 - 1000 Hz seen on the scope. An anti-aliasing filter as illustrated in page 4 is used to filter out the high frequency noises. Suppose that we pick cutoff frequency at 100 Hz, what should be the order (stages) of the filter so that the signal will be reduced no more that 30% while the noise will be reduced at least 96%?

- Max magnitude reduction in signal. $$\left(\frac{1}{\sqrt{1+(\frac{60}{100})^2}}\right)^{2\ (order)} = 0.74$$

- Min magnitude reduction in noise. $$\left(\frac{1}{\sqrt{1+(\frac{500}{100})^2}}\right)^{2} = 0.038$$

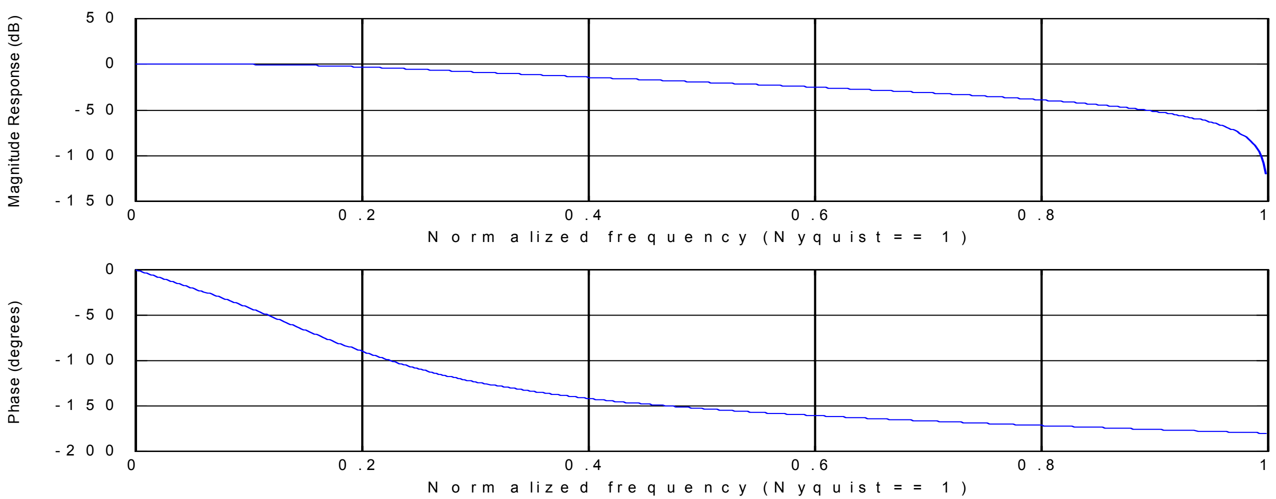
Design a Digital Low Pass Filter

- The general structure of a digital filter is:
- y = a1*y_1 + a2*y_2 + .. + b0*x+b1*x_1 + b2*x_2, where
  - y is the current (filtered) output that you want to compute
  - x is the current raw input before filter
  - y_k and x_k are k-step previous output and input respectively.
- You can find a's and b's using Matlab.
- [b, a] = butter[n, wn] gives a nth order (n stage) butterworth filter with cut off frequency wn. wn must be in the form of percentage of the Nyquist frequency, which is sampling rate/2.0.
- Suppose that sampling rate is 200 Hz and the cutoff frequency is 20, then the Nyquist frequency is 200/2 = 100 Hz and wn = 20/100 = 0.2.

Digital Filter Design - 2

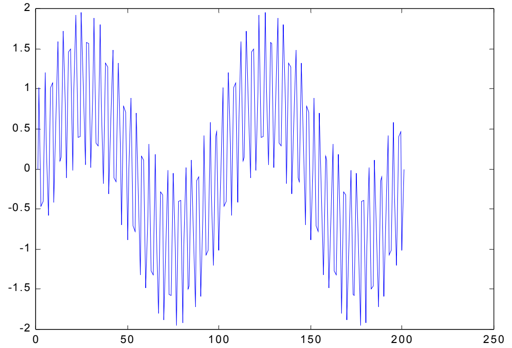
- [b, a] = butter(2, 0.2)
- b = 0.0675 0.1349 0.0675
- a = 1.0000 -1.1430 0.4128
- y(n) = b(1)*x(n) + b(2)*x(n-1) + ... + b(nb+1)*x(n-nb) - a(2)*y(n-1) - ... - a(na+1)*y(n-na)
- y(n) = 0.0675x(n) + 0.01349x(n-1) +0.0675x(n-2) + 1.143y(n-1) – 0.4128y(n-2)

- freqz[b,a] //decibel 20 log (v1/v2), the base is 10.

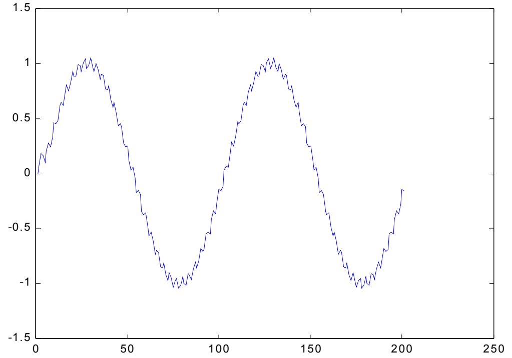


Example

- t = 0:0.005:2; %sample at 200 Hz for 2 sec.
- x = sin(2*pi*t); % 1 Hz signal
- y = sin(2*pi*30*t); % 30 Hz noise
- plot(x+y)



- wn = 20/(200*0.5)=0.2 % cut off at 20 Hz
- Nyquist
- [b, a] = butter(2, 0.2) % 2^{nd} order filter
  % cutoff 20 Hz
- z = filter(b, a, (x+y));
- plot(z)



- "%" comment. ";" suppress output.

Summary

- In this lecture, we have reviewed
  - How to design an analog filter that is needed for anti-aliasing.
  - How to design a digital filter using Matlab