Basics of Signal Processing

Raj Rajkumar
Lecture #23

Previous Lecture

- Feedback control
  - Stability
  - Instability
  - Marginal stability
- Feedback controllers
  - Proportional control
  - Proportional + Derivative control
  - Proportional + Derivative + Integral control
- Simulation diagram concepts

Outline of This Lecture

- Signal and data are at the interface of embedded software engineering and ECE.
- **Objective**:
  - Understand the basic concepts
  - Master issues that a software engineer should do and can do
  - Ask for expert help intelligently
- **Today**: Signals and data acquisition
  - Source of deterministic errors and random noises
  - Basics of signal spectrum
  - Basic filters
    - What you can do and when you should ask for help

Signals

- In many real-time systems, we want to measure environmental variables like temperature, pressure, humidity, light, sound, the location of a valve, etc.
- We may also be receiving signals broadcast (say AM radio, FM radio, satellite radio, TV, etc.) at an antenna
  - Broadcast signals are often at some basic carrier frequency
- The underlying variable is typically changing at a certain (maximum) rate
- The "signal" is the underlying information that we want to capture as precisely as possible
  - But, what we measure/receive often includes "noise"
  - Signal-to-Noise Ratio (SNR) is a term for the power ratio between a signal (meaningful information) and the background noise:

$$SNR = \frac{P_{signal}}{P_{noise}}$$

Deterministic Errors

- **Bias** adds an offset to the true value. Most analog components have this problem over time
- *Solution*: **calibration**
  - By adjusting resistors *or*
  - By compensating **for** the bias in software.
- **Quantization errors**: the resolution is not fine enough for application needs.
  - Proper configuration of voltage ranges in A/D and D/A.
  - A standard A/D and D/A card may give a 12-bit resolution over a given range. However, higher resolution ones are also available.
- **Aliasing**: discussed later.

Sources of Random Noises

- **Noise** is the electrical signal that you do *not* want.
  - In fact, the music broadcast from radio stations is a common source of electro-magnetic interference (EMI).
- **Internal** from the electronics of the A/D - D/A card
  - It is usually very small.
- **Environmental** :
  - Noise from **equipment power transformers**. Switching transformers are cheap. The very inexpensive ones are quite noisy. A linear transformer is heavy, expensive but quiet.
  - Noise from the **power line** due to power tools, elevators etc. Buy line voltage conditioner or move to another place to do your work.
- **EMI**: electrical shielding and/or using differential mode.
  - Do you still remember differential mode inputs?
  - The *oscilloscope* is your best friend to identify the source of noise
- **Poor grounding** is a common source of noise problems. Before trying anything fancy, make sure that your equipment is properly grounded.

Key Concepts in Signal Processing - 1

- **Fourier** discovered that *a signal can be decomposed into a sum of sinusoids*.
- If you have a set of samples, you can use FFT (**F**ast **F**ourier **T**ransform) in tools like MATLAB to take a look at its frequency components.
- **Bandwidth** refers to the frequency components of a signal that have non-negligible magnitudes with respect to the application at hand.

Joseph Fourier, 1768-1830

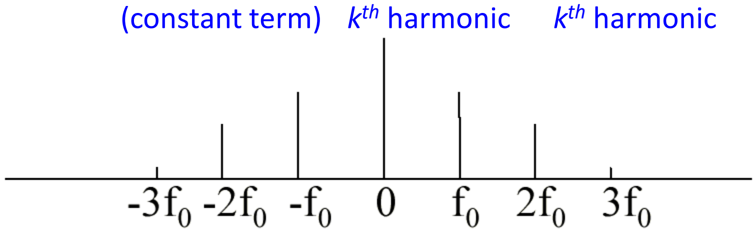
Basic Signal Processing Concepts

Fourier Series

- For *any* periodic function, whose period is $T = 1/ f_0$

$$f(t) = \boxed{C_0} + \sum_{k=1}^{\infty} \boxed{C_k} \cos(2\pi\ f_0 k t + \boxed{\theta_k})$$

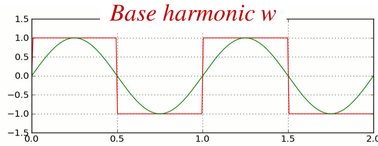
"DC" offset (constant term) — Amplitude of k^{th} harmonic — Phasing of k^{th} harmonic



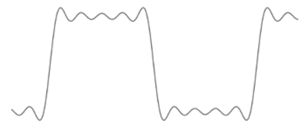
The Fourier Equivalent for a Square Wave

$$x_{\text{square}}(t) = \frac{4}{\pi}\sum_{k=1}^{\infty}\frac{\sin\left((2k-1)2\pi f t\right)}{(2k-1)}$$
$$= \frac{4}{\pi}\left(\sin(2\pi f t) + \frac{1}{3}\sin(6\pi f t) + \frac{1}{5}\sin(10\pi f t) + \cdots\right).$$

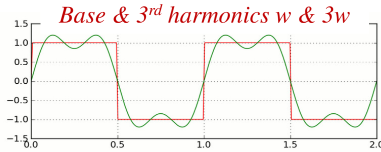
Base harmonic w



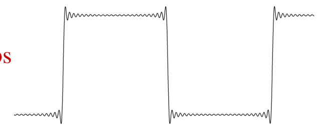
in 5 steps



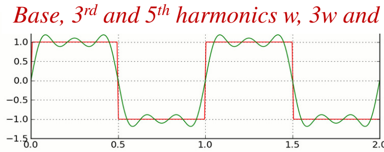
Base & 3rd harmonics w & 3w



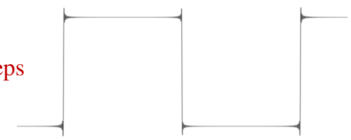
in 25 steps



Base, 3rd and 5th harmonics w, 3w and 5w



in 125 steps



Key Concepts in Signal Processing - 2

- The most common noises are **high-frequency noises**. Hence, **low-pass filters** are popular.
- Noise in the same frequency range as the signal can only be filtered by **model-based filtering**, e.g., ***Kalman filters***
- If you have significant noise in your signal range, try to prevent them *at the source*
  - Shielding, differential inputs, better transformers, proper grounding, …
- If this still does not work, ask a professional in signal processing to help. (How can you know if you have this problem?)

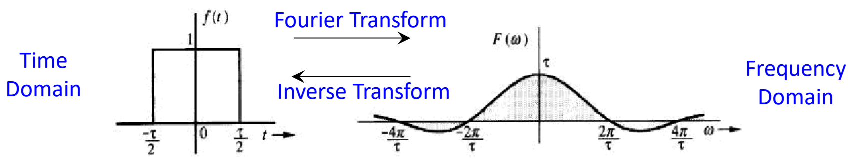
Basic Signal Processing Concepts

Fourier Transform

- Transforms one complex-valued function of a real variable into another
- The domain of the original function is typically time and is accordingly called the *time domain*. The domain of the function is frequency, and so the Fourier transform is often called the *frequency domain representation* of the original function. It describes which frequencies are present in the original function.

A signal f(t) is called an energy signal if $\int_{-\infty}^{\infty} f(t)\, dt < \infty$

The **fourier transform** of the signal is given by $F(f) = \int_{-\infty}^{\infty} f(t) e^{-j2\pi t} dt$



- *Key observation*: the narrower the pulse, the wider is the frequency range.

Key Concepts in Sampling

- Nyquist Rate
  - To correctly capture an analog signal digitally,

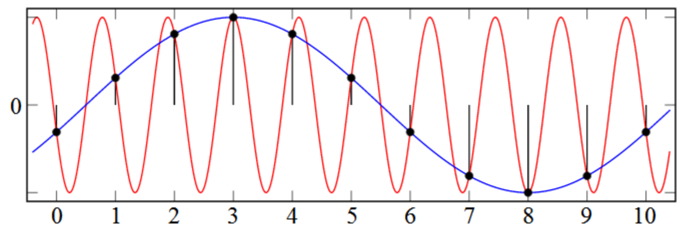
  The sampling rate must be at least twice the bandwidth of the signal.
- This "2 times" result assumes perfection in the sampling and filtering process
- You will *not* have such perfection in practice. So, the practical rule of thumb is
  - *Sample at least 3 times* and *preferably 5 times* higher than the base frequency if you can afford it.

Correctly sampled

Incorrectly sampled

Aliasing

- **Aliasing**:
  - If you sample too slowly, the high-frequency components will become irregular noise at the *sampling* frequency
  - Beware: These are in the same frequency range of your signal!!!
- Look at the samples below alone
- Can you tell which of the two frequencies the sampled series represents?
- Either of the two signals could produce the samples, i.e., the signals are "aliases" of each other



Applications of Nyquist Theorem - 1

- "The signal is in the eye of the beholder"
- Nyquist was interested in digital samples that capture all the information in the electrical waveform.
- The term "signal" in **Nyquist Theorem** means the total information in the waveform. His signal means the sum of
  - The "real signal" that you must love and
  - The noise that you will hate
- This "*love-hate situation*" makes the correct application of the Nyquist Theorem interesting.
- Suppose that we have a signal which has frequency components in the range of 10 to 60 Hz while noise ranges from 100 to 500 Hz.
  - What is a practical lower bound for sampling?

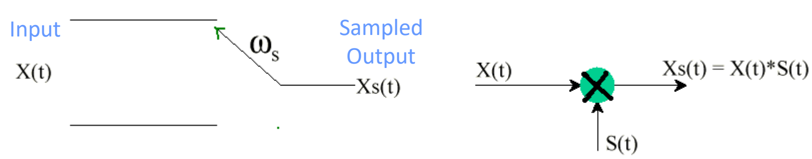
Applications of Nyquist Theorem - 2

- If you are sampling below 1500 Hz, a significant portion of the 500 Hz noise will be transformed into lower-frequency noises that you cannot filter away using classical frequency domain filters.
- 1500 Hz may be way too fast for a standard PC with a normal 10 *ms* timer resolution. You need DSP hardware if you want to digitally filter out the noise after sampling.
- Another way is to use an *anti-aliasing* filter.
  - A simple hardware analog filter **before** the sampling process.

Signal + Noise → Analog Low-Pass Filter → Sampling & A/D Conversion

Signal + Noise	Analog Low-Pass Filter	Sampling & A/D Conversion
≤ 60Hz signal + ≥ 100Hz noise	*Cut-off at (say)80Hz*	*Sample at Nyquist Rate or Higher*

A/D Sampling



$$S(t) = \frac{\tau}{T_s}(1 + \sum_{n=1}^{\infty} 2A_n \cos(n\omega_s t))$$

- *S(t)* is a rectangular pulse train, where τ is the contact time and ω_s is the sampling rate.

i.e. when you digitally sample a continuous signal, you end up multiplying the original signal by cosine functions

Modulation Theorem

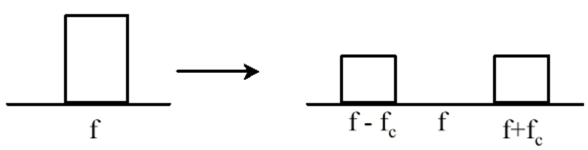
- The effect of "chopper sampling" is to multiply a *harmonic series of cosine functions* to the signal.
- By the **Modulation Theorem**, the cosine function splits and frequency-shifts the original signal.

Basic Signal Processing Concepts

Modulation Theorem

$$v(t) \cos \omega_c t \rightarrow \tfrac{1}{2} (V(f - f_c) + V(f + fc))$$

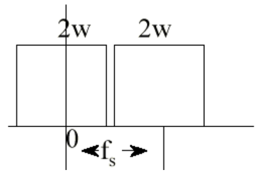
- Multiply a signal *v(t)* by a cosine function with frequency f_c splits the signal into two parts and frequency shifted by f_c and $-f_c$



The higher the sampling rate, the bigger is the frequency shift and the separation between the shifted signals.

Nyquist Rate and Aliasing

- For a signal with bandwidth W, the amount of frequency shift, f_s , must be greater than $2W$ or there will be an overlap between the spectra of the shifted signals.
- The overlap, if any, is called **aliasing**.



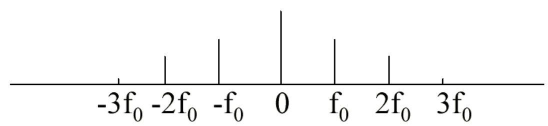
- It follows that the sampling rate must be at least two times the bandwidth, the highest frequency in the signal, to avoid aliasing. This is known as the **Nyquist Rate**.
  – If there is no aliasing, then the signal can be recovered perfectly *in theory* using ideal low-pass filters.

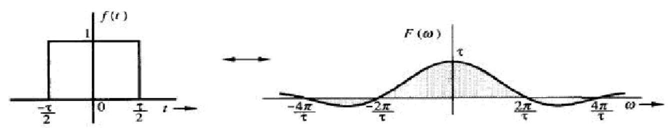
Overview of Filters

- Let's look at the basics of a low-pass filter.
- It is reasonable to expect that you can design simple analog filters and simple software digital filters using tools like Matlab.

Frequency Domain View of Signals

- Fourier: **any waveform can be represented as the sum of sinusoids.**
  - Q1: what does *bandwidth* mean?
  - Q2: Which figure represents a periodic signal and which represents a non-periodic signal (also called **energy signal**)?





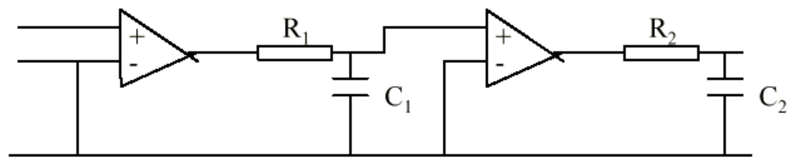
Review Sampling

- 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 <u>practical</u> lower bound in sampling rate if you decide to process the signal digitally?
  A. *20 Hz*
  B. *60 Hz*
  C. *150 Hz*
  D. *3000 Hz*
  E. *5000 Hz*

Low Pass Filters – 1

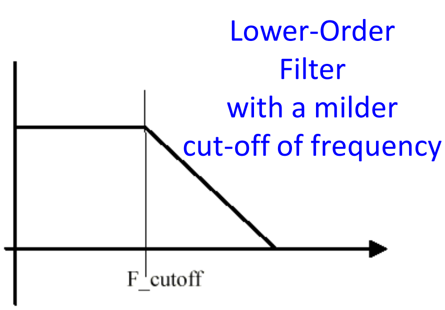
- Signals are often contaminated by high-frequency noise that need to be removed.
- The following is a simple *two-stage active* ***analog*** *filter*:

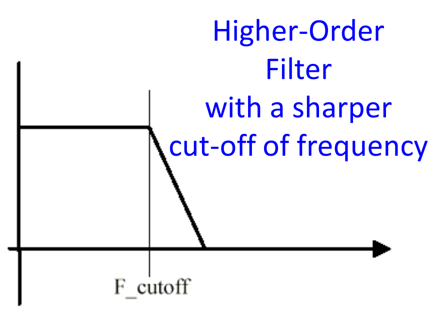


- The Op-Amp isolates the interactions between two RC circuits.
- This allows a simple analysis. You can get such filters on a chip.

Low Pass Filters – 2

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





Low Pass Filters – 3

- The magnitude attenuation and the phase delay of a N-stage low-pass filter are as follows, where ω_{cutoff} = R_1C_1 = R_2C_2 = …

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

Amplitude Of Filter Output

$$\theta(\omega) = -N\left(\arctan\frac{\omega}{\omega_{cutoff}} \right)$$

Frequency Of Filter Output

- The formula for the output amplitude for the commonly used Butterworth **digital** filter is *sqrt*(1 /(1 + $(\omega/\omega_{cutoff})^{2N}$))

Application Notes

- We have two control variables, the cut-off 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.

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

Application Example

- The frequencies of interest in the signal are from 10 to 60 Hz. There is significant noise with frequencies in the range 500 - 1000 Hz seen on the scope.
  - An *anti-aliasing filter* is used to filter out the high-frequency noises.
- Suppose that we pick a cut-off frequency at 100 Hz. What should be the order (stages) of the filter so that the signal will be reduced no more than 30% while the noise will be reduced at least 96%?
- Max. magnitude reduction in signal: $\left(\frac{1}{\sqrt{1+(\frac{60}{100})^2}}\right)^2 = 0.74$
- Min. magnitude reduction in noise: $\left(\frac{1}{\sqrt{1+(\frac{500}{100})^2}}\right)^2 = 0.038$

Design of a Digital Low Pass Filter

- The general structure of a digital filter is:

  $y = a_1{*}y_1 + a_2{*}y_2 + .. + b_0{*}x + b_1{*}x_1 + b_2{*}x_2$, 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 the k-step previous output and input respectively
    - Corresponding to the most recent samples (x) and outputs (y)
- You can find the values of coefficients a and b using tools like Matlab
- *[b, a] = butter[n, ω_n]* gives an n^{th} order (n-stage) Butterworth filter with cut-off frequency ω_n (expressed as a ratio).
  - ω_n must be in the form of percentage of the **Nyquist Frequency**, which is defined as 0.5 of the sampling rate.
- Suppose that the sampling rate is 200 Hz and the cut-off frequency is 20, then the Nyquist frequency is 200*0.5= 100 Hz and ω_n = 20/100 = 0.2

Digital Filter Design Example

- `[b,a] = butter[2, 0.2]`

  ```
  b = 0.0675    0.1349    0.0675
  a = 1.0000   -1.1430    0.4128
  ```
  ← in Matlab

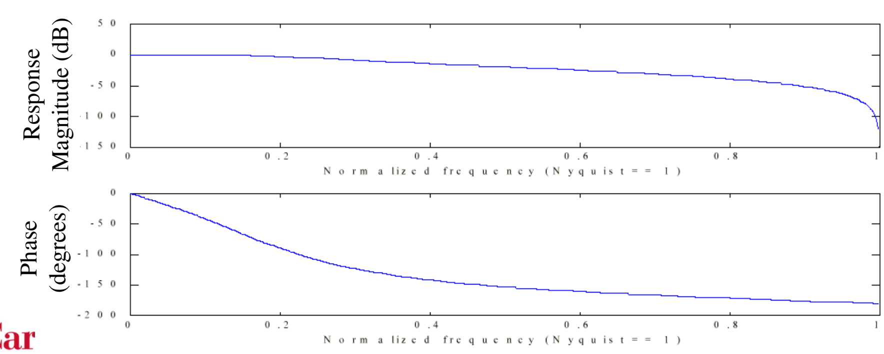
  which is used as

  $y = -1.143*y_1 + 0.4128*y_2 + 0.0675*x + 0.1349*x_1 + 0.0675*x_2$
- What this means:
  - You "sample" the input signal at 200Hz (i.e. every 5*ms*)
  - Take the current sample value (x) and scale it by 0.0675, the previous sample (x_1) by 0.1349 and the even older sample (x_2) by 0.0675
  - Take the previous output value (y_1) and scale it by -1.143 and the next older output (y_2) by 0.4128.
  - Add them up: you have the filtered version of the input signal!
    - Filter cut-off frequency is 20% of the Nyquist frequency
  - Very easy and quick to compute in real-time
  - Note that the constants play a HUGE role.
- The current output becomes the previous output and the previous output becomes the next older output for the next iteration
- Just repeat!

Frequency Response of a Filter

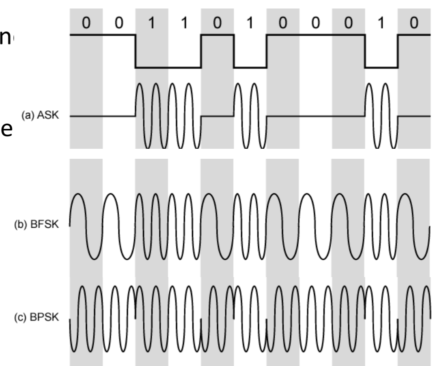
- There are many different types of filter circuits, with different responses to changing frequency.
- The **transfer function** of a filter is the ratio of the output signal to that of the input signal as a function of the (complex) frequency
- The **frequency response** of a filter is generally represented using something called a Bode plot (with frequency along the x-axis and magnitude along the y-axis).

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



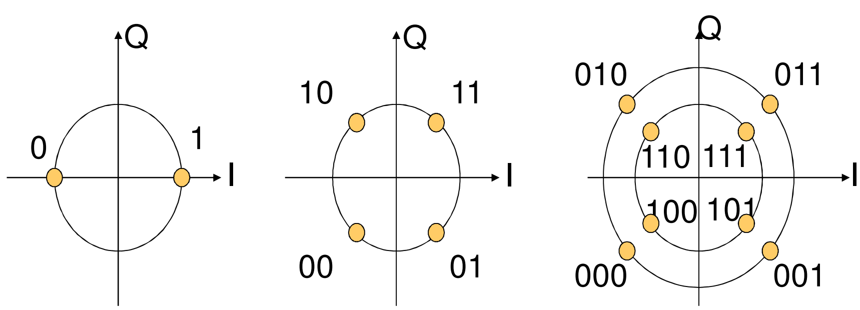
Digital Data to Analog Signal (1 of 2)

- ASK: Amplitude Shift Key
  - Amplitude difference of carrier frequen
- FSK: Frequency Shift Key
  - Frequency difference around carrier fre
- PSK: Phase Shift Key
  - Phase change on carrier frequency



Digital Data to Analog Signal (2 of 2)

- QPSK: Quadrature PSK
- QAM (Quadrature Amplitude Modulation): ASK + QPSK



(a) BPSK (b) QPSK (c) QAM

Summary of Lecture

- Source of deterministic errors and random noises
- Basics of signal spectrum
  - Nyquist sampling
  - Fourier transform
- Basic filters
  - What you can do and when you should ask for help