Engin. 100: Music Signal Processing
Lab #3: Signal Spectra and Filtering

- How do we define the spectrum of a signal?
- Why can we sample without losing information?
- How do we compute the spectrum of a signal?
- How do we use Matlab to compute the spectrum?
- Why should we bother to compute the spectrum?

What is the spectrum of a signal?

- In Engin. 100, we will only define spectra of periodic (repeating) signals. How come?

1. Each musical note played by an instrument is a periodic signal, so this is what we care about.
2. Definition and computation are much easier.
3. Actually, a non-periodic signal can be viewed as a periodic signal with a VERY long period.

Fourier Series of Periodic Signals

- Any real-world periodic signal with period=T (so x(t)=x(t+T) for all t) can be expanded as:
  \[ x(t) = c_0 + c_1 \cos(2\pi t/T-\theta_1) + c_2 \cos(4\pi t/T-\theta_2) + \ldots \]

- DC constant
- Fundamental
- Harmonic
- Period=T
- Period=T/2

The spectrum of x(t) is just a stem plot of the amplitudes \( c_k \) vs. the frequencies \( k/T \) Hertz.

Fourier Series of Musical Signals

- Phase shift \( \theta \) cannot be perceived by the ear.
- No DC term in musical signals. So can use:
  \[ x(t) = a_1 \cos(2\pi t/T) + a_2 \cos(4\pi t/T) + a_3 \cos(6\pi t/T) + \ldots \]
  fundamental harmonic harmonic

- This model originally proposed by Bernoulli.
- Coefficients \( a_k \) →timbre (TAM-ber) of sound:
  - Is why a clarinet sounds different from a flute.
  - Plot of \( a_k \) vs. frequency \( k/T \) Hertz: sound spectrum.

Clarinet Time Waveform and Spectrum

Note “G” (392 Hz) Sampled 44100 Hz

clear; [X S]=audioread('proj3.wav');N=32768;XX=reshape(X,N,13,4);
Y=XX(:,1,2);dt=1/S;df=S/N;t=[0:199];subplot(211),plot(t,Y(1:200))
FY=2*abs(fft(Y))/N;f=df*[0:3699];subplot(212),plot(f,FY(1:3700))
Trumpet Time Waveform and Spectrum
Note “G” (392 Hz) Sampled 44100 Hz

clear; [X,S]=audioread('proj3.wav');N=32768;XX=reshape(X,N,13,4);
Y=XX(:,1,3);dt=1/S;df=S/N;t=dt*[0:199];subplot(211),plot(t,Y(1:200))
FY=2*abs(fft(Y))/N;f=df*[0:3699];subplot(212),plot(f,FY(1:3700))

Engin. 100: Music Signal Processing
Lab #3: Signal Spectra and Filtering
• How do we define the spectrum of a signal?
• Why can we sample without losing information?
• How do we compute the spectrum?
• How do we use Matlab to compute the spectrum?
• Why should we bother to compute the spectrum?

Fourier Series of Periodic Signals
• Any real-world periodic signal with period=T can be expanded in either of these two forms:
  \[ x(t) = c_0 + c_1 \cos(2\pi t/T-\theta_1) + c_2 \cos(4\pi t/T-\theta_2) + \ldots \]
  \[ x(t) = a_0 + a_1 \cos(2\pi t/T) + a_2 \cos(4\pi t/T) + \ldots \]
• Coefficients in these two forms are related by:
  \[ A \cos(t) + B \sin(t) = C \cos(t-\theta) \] (cf. Lab #1) where \[ C = (A^2 + B^2)^{1/2} \]; \[ \tan \theta = B/A; A = C \cos \theta; B = C \sin \theta. \]

Example: Period=0.001 seconds
Bandlimited to 8000 Hertz
• \[ x(t) = a_0 + a_1 \cos(2\pi 1000t) + a_2 \cos(2\pi 2000t) + \ldots \]
  \[ + b_1 \sin(2\pi 1000t) + b_2 \sin(2\pi 2000t) + \ldots \]
• 0 1000 Hz (fund.) 2000 Hz (harmonic)
• In general, this expansion is infinite and requires infinitely high frequencies at integer multiples of the fundamental frequency=1/period=1000 Hz.
• But what if the signal is bandlimited to 8000 Hz?
• Means: It has no sinusoids with freqs>8000 Hz.

Sampling Signals with Period=T seconds and Bandlimited to B Hertz
• IDEA: If a signal is completely characterized by 2BT+1 numbers, we can compute those 2BT+1 numbers by sampling 2BT+1 times per period!
  • THAT IS: Sample faster than 2BT times/period.
  • Sampling rate>(2BT times)/(T second)=2B Hertz. So need: sampling rate>2(maximum frequency).
Sampling Signals with Period=T seconds and Bandlimited to B Hertz

- **Sampling rate** > \(2BT\) times/(T second) = 2B Hertz. **Need**: sampling rate > 2(maximum frequency).
- **Note**: Period=T does not matter to sampling rate! We can make T arbitrarily large; this still holds.
- **Sampling theorem**: co-discovered Claude Shannon, U-M CoE Class of 1936 (EE). Bust outside EECS.

Engin. 100: Music Signal Processing Lab #3: Signal Spectra and Filtering

- How do we define the spectrum of a signal?
- Why can we sample without losing information?
- How do we compute the spectrum of a signal?
- How do we use Matlab to compute the spectrum?
- Why should we bother to compute the spectrum?

Closed-Form Solution to the Linear System of Equations

- **General**: Signal is sampled at S sample/seconds; has period=T seconds; bandlimited to B Hertz.
- **Data**: \(\{x(1/S), x(2/S), x(3/S), \ldots, x(ST/S) = x(T)\}\) (sampling every 1/S seconds, up to T seconds).
- **Solve**: ST=N equations in 2BT+1 unknowns:
  - Set \(t = \{1/S, 2/S, \ldots, (ST)/S = T\}\) in the equation:
  - \(x(t) = a_0 + a_1 \cos(2\pi t/T) + \ldots + a_{BT} \cos(2\pi BTt/T)
  - \(+ b_1 \sin(2\pi t/T) + \ldots + b_{BT} \sin(2\pi BTt/T)\)

Closed-Form Solution to the Linear System of Equations

- There is a closed-form solution (formula):
  - \(a_k = (2/N) \sum x(n/S) \cos(2\pi nk/N)\) where N=ST.
  - \(b_k = (2/N) \sum x(n/S) \sin(2\pi nk/N)\) where N=ST.
  - \(a_0 = (1/N) \sum x(n/S) = \text{average value of } x(n/S).\)
  - Computed for \(k = 0\) to \(B\) (2BT+1 numbers).
  - All sums are taken from \(n = 1\) to \(n = N = ST\).
  - Where does this formula come from? See the Appendix of Lab #3 (MUCH algebra!).

Using Matlab to Compute Spectra

- Matlab’s “F=fft(X)” computes the following:
  - \(F(X) = \sum x(n-1) e^{-i2\pi (n-1)(k-1)/N}\) for \(k = 1, \ldots, N\) where sum is from \(n = 1\) to \(N\), and \(e^{-i\theta} = \cos(\theta) - i \sin(\theta)\).
  - \(a_{k-1} = k^{th}\) element of “\(2/N * \text{real}(fft(X))\)”, \(k = 1, 2, \ldots, B\).
  - \(b_{k-1} = k^{th}\) element of “\(-2/N * \text{imag}(fft(X))\)”, \(k = 1, 2, \ldots, B\).
  - \(c_{k-1} = k^{th}\) element of “\(2/N * \text{abs}(fft(X))\)”, \(k = 1, 2, \ldots, B\).
  - \(X = \text{vector of sampled signal}; N = ST\) (integer); we need to ensure \(S > 2\) (maximum frequency of \(x(t)\)).
  - \(T = \text{period in seconds} = \#\text{samples per period}/S.\)
Using Matlab’s “fft” to compute spectra

- “fft” output has mirror symmetry \( e^{i2\pi(n-1)(k-1)/N} \).
- **DELETE THE 2\(^{nd}\) HALF OF ITS OUTPUT!**
- Watch indexing: Matlab: 1 to N, Math: 0 to N-1.
- \((2/N)\text{real}(\text{fft}([x(0)\ldots x(N-1)],N))\) gives
  \([2a_0 \ a_1 \ a_2 \ldots a_{N/2-2} \ a_{N/2-1} \ b_{N/2} \ b_{N/2-2} \ldots b_1]\)
- \(-(2/N)\text{imag}(\text{fft}([x(0)\ldots x(N-1)],N))\) gives
  \([0 \ b_1 \ b_2 \ldots b_{N/2-2} \ b_{N/2-1} \ -b_{N/2} \ -b_{N/2-2} \ldots -b_1]\)
- \((2/N)\text{abs}(\text{fft}([x(0)\ldots x(N-1)],N))\) gives
  \([2c_0 \ c_1 \ c_2 \ldots c_{N/2-2} \ c_{N/2-1} \ c_{N/2} \ c_{N/2-1} \ c_{N/2-2} \ldots c_2 \ c_1]\)

Small Illustrative Numerical Example

- GIVEN: \(x(t)=\cos(2\pi440t)\) sampled at 8192 Hertz.
- \(>>X=\cos(2\pi*440*[0:8191]/8192)\); \(\%T=1\) second.
- \(>>F=(2/8192)\text{abs}(\text{fft}(X,8192))\) outputs all zeros except one at \(F(441)=1\) and \(F(7753=8193-440)=1\).
- So how do we interpret this Matlab output?
- Remember to delete the mirror image output.
- There is a single sinusoid, but what frequency?

Engin. 100: Music Signal Processing Lab #3: Signal Spectra and Filtering

- How do we define the spectrum of a signal?
- Why can we sample without losing information?
- How do we compute the spectrum of a signal?
- How do we use Matlab to compute the spectrum?
- Why should we bother to compute the spectrum?

Spectral filtering of noisy signals

- **GIVEN**: Observe: \(y=x+n=\text{desired signal+noise}\).
- **GOAL**: Filter out as much noise \(n\) as possible.
- **KNOW**: Signal bandlimited to \(f=(K-1)S/N\) Hertz for \(k=1,2\ldots N/2\) (require maximum frequency<\(S/2\)).
- Frequency resolution (separation) is \(1/T\) Hertz.

- HERE: \(X\) is samples of one sinusoid at frequency \((441-1)(8192\ \text{sample/sec.})/(8192\ \text{samples})=440\ \text{Hz}\).
- Plot only **first half of \(F\)** to get the spectrum of \(X\)!

Low-pass filtering a noisy EKG signal

- **EKG**=Electrocardiogram (heart) electric signal.
- Periodic (we hope!) with period about 1 second (note 60 beats per minute=1 beat per second).
- Components at 1,2,3…Hertz up to about 7 Hertz.
- **So**: We eliminate all frequencies above 7 Hertz.
- **Result**: Most noise gone; not low frequencies.
- See next slide for signals and their spectra.
Noisy Signal (from First Lecture)
What is the signal buried in noise?
How do we recover the signal?

Spectrum of noisy signal: abs(fft(X))
Huh? But look at the vertical scale
Spectrum also looks random.
Look carefully: there is a large component very close to zero, at the extreme left. Zoom in to this (see next slide).

Spectrum of noisy signal: Zoom in.
Now it’s clear what the signal is!
There is a large component (peak) at index K=3.
This is a sinusoid at the frequency
\( f = (K-1)SN = (3-1)8000/8000 = 2 \text{ Hertz (is right).} \)
Next, we filter it.

Filtering Noisy Signal: Set all of the components to zero except K=3,7999
Matlab code used:
\[
T = [0:7999]/8000;
X = \cos(4\pi t^T);
N = \text{randn(1,8000)};
Y = X + 10N;
F = \text{fft}(Y);
F(4:7998) = 0;
F([1 2 8000]) = 0;
Z = \text{real(ifft}(F));
\]

CONCLUSION
- Periodic signals (including musical notes) can be expanded in a Fourier series.
- The signal is a sum of sinusoids at frequencies that are integer multiples of 1/(period of signal).
- If signal is both bandlimited and periodic, then the Fourier series has only a finite #terms, and we can compute coefficients from its samples.
- There is a closed-form solution to the resulting linear system; Matlab can compute it easily.
- Use \( f = (K-1)SN \) to get frequencies from index.

NOTE: Need to keep both K=3 and its mirror index K=8000-3.
Mirror index of K is N-2-K due to Matlab indexing starting at 1. If counting from k=0 to N-1, so f=KS/N, the mirror index of k is N-k.