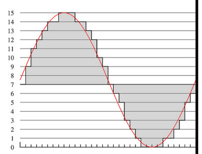
Penn Engineering

ESE



Lecture #4 – Converting from time to frequency domain

ESE 150 – DIGITAL AUDIO BASICS

TEASER

- **Play this on piano:**

TEASER

- **Play**

3 eighth notes   1 half note

Cheat: G4 E4ᵇ F4 D4q

INFORMATION

- **1s / quarter note ➔ 10s of sound**
- **How many bits to represent 10s of sound with 16b samples and 44KHz sampling?**
  - 44K Hz x 16b/sample x 10s = 7040K =7Mbits

REPRESENTATION

- **How does musical staff represent sound?**
  - What does vertical position represent?
  - Note shape?

FREQUENCY REPRESENTATION

- **There are other ways to represent**
  - Frequency representation particularly efficient

392   311   348   294

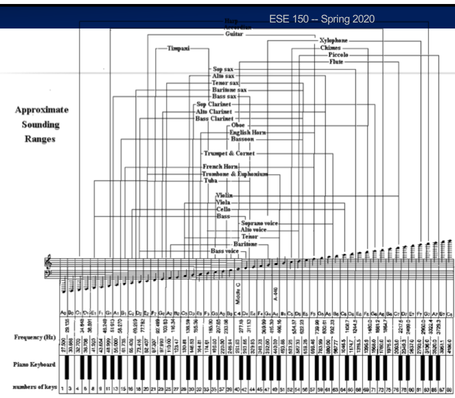
Frequencies in Hertz

Frequency Representation

- How much information is this musical staff communicating?
- How many keys on piano? → bits/note

Hamburg Steinway D-274 Piano photo by Karl Kunde

https://commons.wikimedia.org/wiki/File:D274.jpg



Larry Solomn: http://solomonsmusic.net/insrange.htm

Frequency Representation

- How much information is this musical staff communicating?
- How many keys on piano? → bits/note
- Let's say 8b duration
- How many bits for 8 notes?
  - (7b/note+8b/duration) x 8 note = 120 bits?

Lab 2 Postlab

- You reproduced 800 samples of a 300Hz sine wave at 1000Hz with 8b precision
  - 6400b
- What did you need to specify to do that?
- How may bits to represent that?

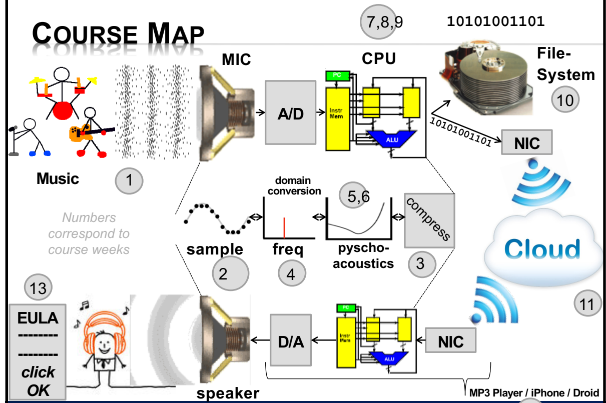
Conclude

- Can represent common sounds much more compactly in frequency domain than in time-sample domain
  - Frequency domain ~ 120b
  - Time-sample domain ~ 7Mb

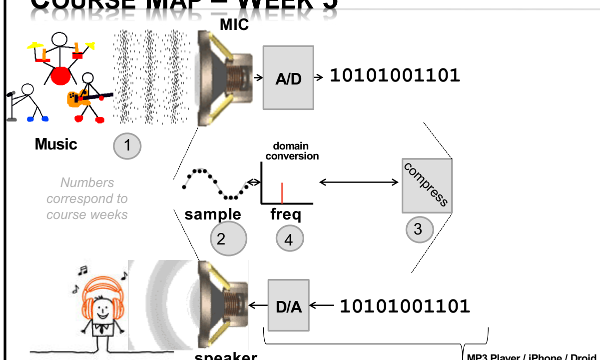
Lecture Topics

- Teaser: frequency representation
- Where are we on course map?
- What we did in lab last week
  - How it relates to this week
- The Fourier Series
  - can represent any signal in frequency domain
- The Discrete Fourier Transform (DFT)
  - can translate any signal between time and frequency domain
  - change of basis
- Next Lab
- References

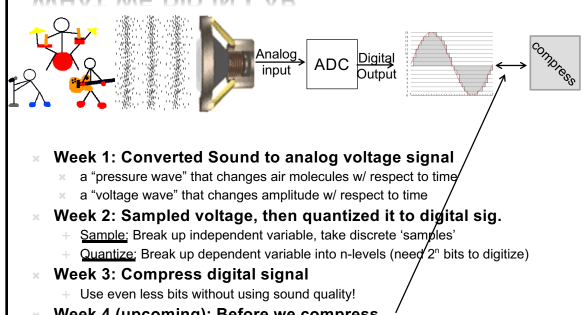
Course Map



Course Map – Week 5



What we did in Lab…



- Week 1: Converted Sound to analog voltage signal
  - a "pressure wave" that changes air molecules w/ respect to time
  - a "voltage wave" that changes amplitude w/ respect to time
- Week 2: Sampled voltage, then quantized it to digital sig.
  - Sample: Break up independent variable, take discrete 'samples'
  - Quantize: Break up dependent variable into n-levels (need 2^n bits to digitize)
- Week 3: Compress digital signal
  - Use even less bits without using sound quality!
- Week 4 (upcoming): Before we compress…
  - Put our 'digital' data into another form…BEFORE we compress…less stuff to compress!

Background

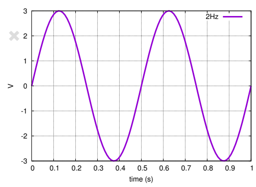
What is the Frequency Domain?

Musical Representation

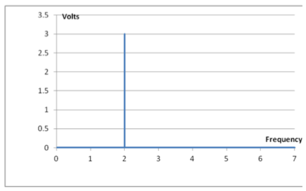
- With this compact notation
  - Could communicate a sound to pianist
  - Much more compact than 44KHz time-sample amplitudes (fewer bits to represent)
  - Represent frequencies

Time-Domain & Frequency-Domain

- **As an example…let's say we have a pure tone**
  - If period: T = 1/2 and Amplitude = 3 Volts
  - $s(t) = A\,sin(2\pi f t) = A\,sin(2\pi 2 t)$



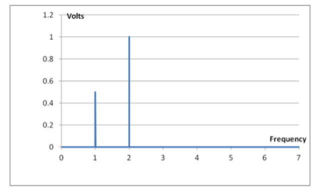
Time domain representation



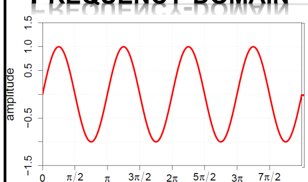
Frequency domain representation

Frequency-Domain

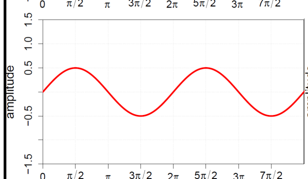
- **Of course, not all signals are this simple**
- **For example:** $s(t) = \sin(2\times 2\pi\times t) + \frac{1}{2}\sin(2\pi\times t)$
- **Question: What will the frequency representation look like?**

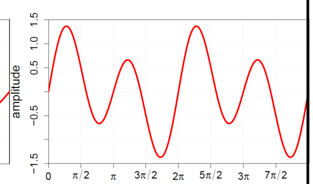


Frequency-Domain

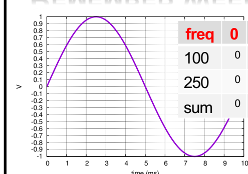


- **How about the time-domain?**
  - Plot $\sin(2\times 2\pi\times t)$
  - Plot $\frac{1}{2}\sin(2\pi\times t)$
  - Sum: $\sin(2\times 2\pi\times t) + \frac{1}{2}\sin(2\pi\times t)$
  - ***Notice how it was easier to plot the frequency domain representation***



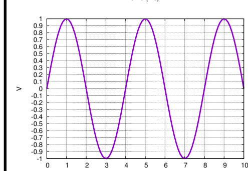


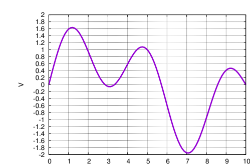
Remember Week 2: Preclass 3



freq	0	1	2	3	4	5	6	7	8	9	10
100	0	0.6	0.95	0.95	0.6	0	-0.6	-0.95	-0.96	-0.6	0
250	0	1.0	0	-1	0	1	0	-1	0	1	0
sum	0	1.6	0.95	-0.05	0.6	1	-0.6	-1/95	-0.95	0.4	0

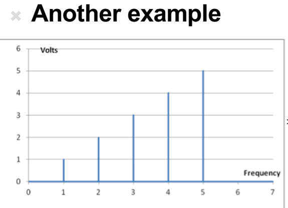
$\sin(100\times 2\pi\times t) + \sin(250\times 2\pi\times t)$

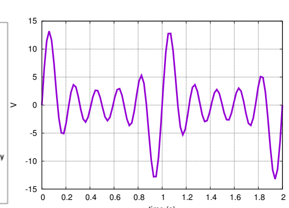




Frequency-Domain

- **Another example**





The time domain plot on the right is really the sum of 5 sinusoids, where 5 Hz is the strongest component of the signal

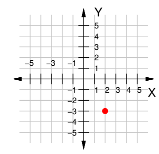
Frequency-Domain

- **So far…**
  - we have seen how a signal written as:
    - a sum of sines of different frequencies
  - can have a ***frequency domain representation***
- **Each sine component…**
  - is more or less important depending on its **coefficient**
  - Example: $s(t) = 1\,\sin(2\times 2\pi\times t) + \frac{1}{2}\sin(2\pi\times t)$
- **Can <u>any</u> arbitrary signal be represented as a sum of sines?**
  - No. But the idea has potential, let's explore it!

Vector Background

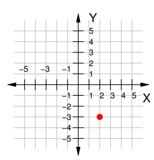
Vector Space

- **We're familiar with multi-dimensional spaces and vector representation**
  - E.g. Cartesian Coordinates in 2 Space
    - 2 dimensions X, Y
    - Represent points as vector with 2 elements (x,y)
  - Preclass 4a
    - What is the (x,y) coordinate of the red dot?



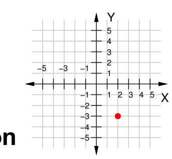
Vector Space

- **We're familiar with multi-dimensional spaces and vector representation**
  - E.g. Cartesian Coordinates in 2 Space
    - 2 dimensions X, Y
    - Represent points as vector with 2 elements (x,y)
  - Can easily extend to 3 Space
    - (x,y,z)
  - Harder to visualize, but could extend to any number of dimensions
    - (d1,d2,d3,d4,d5,….)



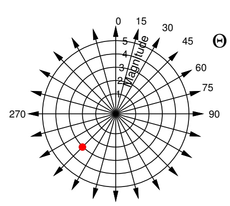
Orthogonal Basis

- **We can describe any point in the space by a linear combination of orthogonal basis elements**
  - E.g. Cartesian Coordinates in 2 Space
    - x -- [1,0]
    - y -- [0,1]
    - Any point:
      - a*x + b*y = [a,b]
  - Orthogonal – no linear scaling of one gives the other
    - Dot products are zero
- **Combine by linear superposition**



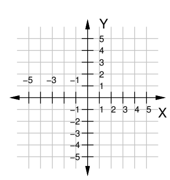
Different Representations

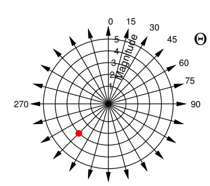
- **We can also represent points in 2-space in polar coordinates**
  - A different orthogonal basis
    - (magnitude, Θ)
- **What is the polar coordinate of the red dot? (4b)**



Can Change Representations

- **Both Cartesian and Polar Coordinates can describe points in the same space.**
  - How do we change polar to Cartesian? (4c)
  - What is the Cartesian coordinate for the red dot? (4d)





Complex Numbers

- **Complex Numbers are an example of this**
  - Real dimension
  - Imaginary dimension
- **Cartesian version:** a+b*i*
- **Polar (Magnitude, angle) version:** $M \times e^{i\theta}$
- Euler's Formula: $e^{i\theta} = \cos\theta + i\sin\theta$

The frequency domain &

The Fourier Series

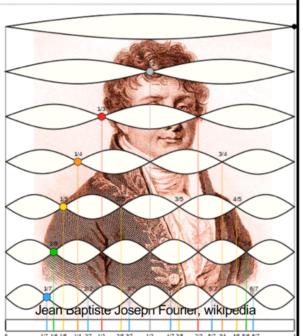
History…

- **Fourier series:**
  - Any ***periodic*** signal can be represented as a sum of simple periodic functions: *sin* and *cos*

  sin(*n*t) and cos(*n*t)

  where *n* = 1, 2, 3, …

  These are called the ***harmonics*** of the signal



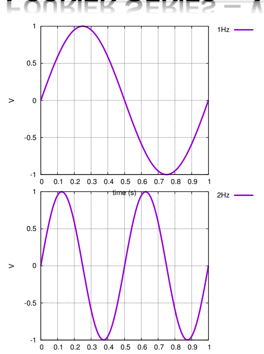
Fourier Series – More Formally

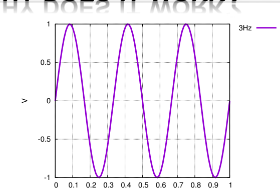
The Fourier Theorem states that any *periodic* function f(t) of period L can be cast in the form:

$$f(t) = a_0 + \sum_{n=1}^{\infty}\left(a_n \cos\frac{n\pi t}{L} + b_n \sin\frac{n\pi t}{L}\right)$$

The constants: a_0, a_n , and b_n are called the Fourier coefficients of *f(t)*

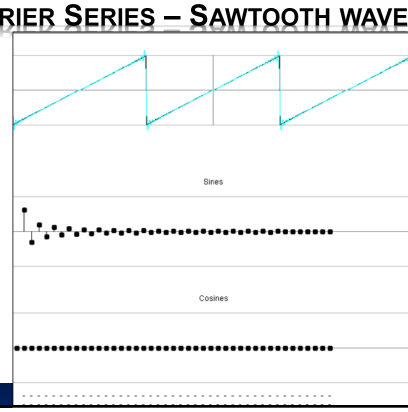
Fourier Series – Why does it work?





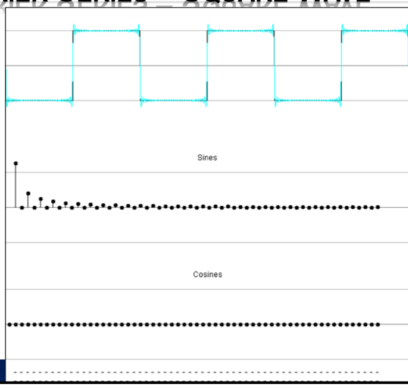
- The cos(*n*x) and sin(*n*x) functions form an orthogonal basis:
  - allow us to represent any periodic signal by taking a *linear combination*
  - of the basis functions without interfering with one another
  - AKA: superposition works!

Fourier Series – Sawtooth Wave



(falstad.com/fourie

Fourier Series – Square Wave



(falstad.com/fourie

Fourier Series (Review of key points)

- **The idea of the series:**
  - Any ***PERIODIC*** wave can be represented as simple sum of sine waves
- **2 Caveats:**
  - Linearity:
    - The series only holds while the system it is describing is *linear* because it relies on the superposition principle
    - -aka – adding up all the sine waves is superposition in action
  - Periodicity:
    - The series only holds if the waves it is describing are periodic
    - Non-periodic waves are dealt with by the Fourier Transform
      - We will examine that in the 2nd half of lecture

Nyquist

- **Remember we said we needed to sample at twice the maximum frequency**
  - Now see all signals can be represented as a linear sum of frequencies
  - …and the frequency components are orthogonal
    - Can be extracted and treated independently

Interlude

- **Wellness**
- **https://docs.google.com/presentation/d/1q55pAhZbpjigHSr0jW94zQkhWi5tyXtZkokMDSOZVVA/edit#slide=id.p**

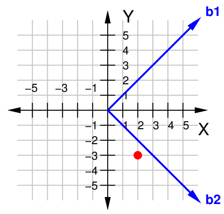
What now?

- **In the first half of the lecture we introduced:**
  - The idea of frequency domain
  - The Fourier Series
- **In the second half of the lecture:**
  - Fourier Transform
  - See how to perform this time-frequency translation
  - Examples

Vector Background

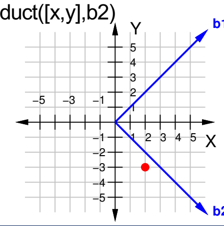
Change of Bases

- **There are more than one set of basis vectors that span a space**
  - For example, might rotate 90 degrees in Cartesian coordinates
    - $b1 = [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}],\ b2 = [\frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}]$
    - Note dotproduct(b1,b2)=0
    - Represent points as linear combination: a*b1+c*b2



Change of Bases

- **Can change basis by performing dot product**
  - $b1 = [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}],\ b2 = [\frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}]$
  - Represent points as linear combination: a*b1+c*b2
  - a=dotproduct([x,y],b1); c=dotproduct([x,y],b2)
    - What are a and c for preclass 4a case?
    - Check correct by seeing that a*b1+c*b2 is what we expect



The Fourier Transform

Preclass 5

- **Compute Dot Products – what did we get?**

i	0	1	2	3	4	5	6	7	8	9	10
time	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Sample	0	0.95	0.59	-0.59	-0.95	0	0.95	0.59	-0.59	-0.95	0

i	0	1	2	3	4	5	6	7	8	9	10
B1	0.00	0.59	0.95	0.95	0.59	0.00	-0.59	-0.95	-0.95	-0.59	0.00
B2	0.00	0.95	0.59	-0.59	-0.95	0.00	0.95	0.59	-0.59	-0.95	0.00
B3	0.00	0.95	-0.59	-0.59	0.95	0.00	-0.95	0.59	0.59	-0.95	0.00
B4	0.00	0.59	-0.95	0.95	-0.59	0.00	0.59	-0.95	0.95	-0.59	0.00
B5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Preclass 5

$$Sample[i] = \sin\left(\frac{2\pi k \times i}{T}\right)$$

- **Note B1 to B5 were sample values from 1, 2, 3, 4, 5 Hz sine waves**

i	0	1	2	3	4	5	6	7	8	9	10
time	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Sample	0	0.95	0.59	-0.59	-0.95	0	0.95	0.59	-0.59	-0.95	0

i	0	1	2	3	4	5	6	7	8	9	10
K=1	0.00	0.59	0.95	0.95	0.59	0.00	-0.59	-0.95	-0.95	-0.59	0.00
K=2	0.00	0.95	0.59	-0.59	-0.95	0.00	0.95	0.59	-0.59	-0.95	0.00
K=3	0.00	0.95	-0.59	-0.59	0.95	0.00	-0.95	0.59	0.59	-0.95	0.00
K=4	0.00	0.59	-0.95	0.95	-0.59	0.00	0.59	-0.95	0.95	-0.59	0.00
k=5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Observe

$$Sample[i] = \sin\left(\frac{2\pi k \times i}{T}\right)$$

- When we compute the dot-product with discrete frequency samples, the only non-zero was the frequency in the signal!

i	0	1	2	3	4	5	6	7	8	9	10
time	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Sample	0	0.95	0.59	-0.59	-0.95	0	0.95	0.59	-0.59	-0.95	0

i	0	1	2	3	4	5	6	7	8	9	10
K=1	0.00	0.59	0.95	0.95	0.59	0.00	-0.59	-0.95	-0.95	-0.59	0.00
K=2	0.00	0.95	0.59	-0.59	-0.95	0.00	0.95	0.59	-0.59	-0.95	0.00
K=3	0.00	0.95	-0.59	-0.59	0.95	0.00	-0.95	0.59	0.59	-0.95	0.00
K=4	0.00	0.59	-0.95	0.95	-0.59	0.00	0.59	-0.95	0.95	-0.59	0.00
k=5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Observe

- **Can identify frequencies with dot product**
  - Identifying projection onto each basis vector in Fourier Series
- **Works because frequency sine waves are orthogonal**
- **Performing a change of basis**
  - From time-sample basis
  - To Fourier (sine, cosine) basis

Time and Frequency Bases

- **Time Sample basis**
  - Also a multi-dimensional space
  - Dimension = # time samples
  - Vector $[t_0,t_1,t_2,t_3, \ldots]$
- **Frequency basis**
  - Multi-dimensional
  - Dimensions = Coefficients of sine and cosine components
  - $f(t) = a_0 + \sum_{n=1}^{\infty}\left(a_n \cos\frac{n2\pi t}{L} + b_n \sin\frac{n2\pi t}{L}\right)$
  - Vector $[a_0,a_1,b_1,a_2,b_2,\ldots]$

Frequency-Domain

- **How to make a song appear "periodic"**
  - Treat the entire song as 1 period of a very complicated sinusoid!
    - (or some length of time, linke 25ms or 1024 samples)
  - *This is the assumption of the Fourier Transform*

Discrete Fourier Transforms

- **Fourier Transforms are nice,**
  - but we want to store and process our signals with computers
- **We extend Fourier Transforms into Discrete Fourier Transforms, or DFT**
  - We know our music signal is now discrete: $x(t) \rightarrow f[k]$
  - The signal contains **N** samples: $0 \leq k \leq N-1$

Warning

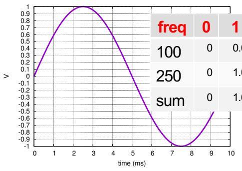
- **Don't get lost in mathematical notation**
- **k – sample – correspond to a time point**
- **n -- frequency component**
- **(put on board)**

DFT – Discrete Fourier Transform

- **Represent any sequence of time samples as**

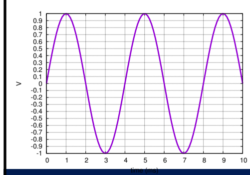
$$f[k] = a_0 + \sum_{n=1}^{N}\left(a_n \cos\frac{n2\pi k}{N} + b_n \sin\frac{n2\pi k}{N}\right)$$

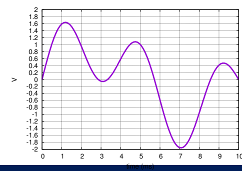
Remember Week 2: Preclass 3



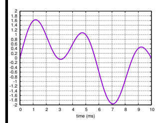
freq	0	1	2	3	4	5	6	7	8	9	10
100	0	0.6	0.95	0.95	0.6	0	-0.6	-0.95	-0.96	-0.6	0
250	0	1.0	0	-1	0	1	0	-1	0	1	0
sum	0	1.6	0.95	-0.05	0.6	1	-0.6	-1/95	-0.95	0.4	0

$\sin(100\times 2\pi\times t)+\sin(250\times 2\pi\times t)$





Remember Week 2: Preclass 3



freq	0	1	2	3	4	5	6	7	8	9	10
100	0	0.6	0.95	0.95	0.6	0	-0.6	-0.95	-0.96	-0.6	0
250	0	1.0	0	-1	0	1	0	-1	0	1	0
sum	0	1.6	0.95	-0.05	0.6	1	-0.6	-1/95	-0.95	0.4	0

n

k

$\sin(100\times 2\pi\times t)+\sin(250\times 2\pi\times t)$

$$f[k]=a_0+\sum_{n=1}^{N}\left(a_n\cos\frac{n2\pi k}{N}+b_n\sin\frac{n2\pi k}{N}\right)$$

$b_{100}=1$, $b_{500}=1$ … rest a_i, $b_i = 0$

[a0,a1,b1,a2,b2,….a500,b500] = [0,0,0,0,….0,1,0,0,0…..,0,1]

DFT – Discrete Fourier Transform

- **Represent any sequence of time samples as**

$$f[k]=a_0+\sum_{n=1}^{N}\left(a_n\cos\frac{n2\pi k}{N}+b_n\sin\frac{n2\pi k}{N}\right)$$

- **Compute a_n, b_n by dot product – preclass 5!**
  - $a_n=\left(\frac{2}{N}\right)\sum_{k=0}^{k=N}\left(Sample[k]\times\cos\left(\frac{n2\pi k}{N}\right)\right)$
  - $b_n=\left(\frac{2}{N}\right)\sum_{k=0}^{k=N}\left(Sample[k]\times\sin\left(\frac{n2\pi k}{N}\right)\right)$

DFT – Discrete Fourier Transform

- **Compute a_n, b_n by dot product – preclass 5!**
  - $a_n=\left(\frac{2}{N}\right)\sum_{k=0}^{k=N}\left(Sample[k]\times\cos\left(\frac{n2\pi k}{N}\right)\right)$
  - $b_n=\left(\frac{2}{N}\right)\sum_{k=0}^{k=N}\left(Sample[k]\times\sin\left(\frac{n2\pi k}{N}\right)\right)$

k

n

i	0	1	2	3	4	5	6	7	8	9	10
n=1	0.00	0.59	0.95	0.95	0.59	0.00	-0.59	-0.95	-0.95	-0.59	0.00
n=2	0.00	0.95	0.59	-0.59	-0.95	0.00	0.95	0.59	-0.59	-0.95	0.00
n=3	0.00	0.95	-0.59	-0.59	0.95	0.00	-0.95	0.59	0.59	-0.95	0.00
n=4	0.00	0.59	-0.95	0.95	-0.59	0.00	0.59	-0.95	0.95	-0.59	0.00
n=5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

DFT – Discrete Fourier Transform

- **Represent any sequence of time samples as**

$$f[k]=a_0+\sum_{n=1}^{N}\left(a_n\cos\frac{n2\pi k}{N}+b_n\sin\frac{n2\pi k}{N}\right)$$

- **Compute a_n, b_n by dot product**
  - $a_n=\left(\frac{2}{N}\right)\sum_{k=0}^{k=N}\left(Sample[k]\times\cos\left(\frac{n2\pi k}{N}\right)\right)$
  - $b_n=\left(\frac{2}{N}\right)\sum_{k=0}^{k=N}\left(Sample[k]\times\sin\left(\frac{n2\pi k}{N}\right)\right)$

Note: sum over different dimensions!

DFT – Discrete Fourier Transform (Complex Representation)

- **Represent any sequence of time samples as**

$$f[k]=a_0+\sum_{n=1}^{N}\left(a_n\cos\frac{n2\pi k}{N}+b_n\sin\frac{n2\pi k}{N}\right)$$

- **From Euler's formula $e^{i\theta}=\cos\theta+i\sin\theta$, can also express as exponential**

$$f(k)=\frac{1}{N}\sum_{n=0}^{N-1}\boldsymbol{X_n}\,e^{-i\left(\frac{2\pi nk}{N}\right)}$$

- Representation vector is $[X_0, X_1, \ldots X_{N-1}]$; X_n complex

DFT – Discrete Fourier Transform (Complex Representation)

- Represent any sequence of time samples as

$$f(k) = \frac{1}{N}\sum_{n=0}^{N-1} X_n\, e^{-i\left(\frac{2\pi nk}{N}\right)}$$

- Compute X_n by dot product

$$X_n = \sum_{k=0}^{N-1} x_k \times e^{-i\frac{2\pi\, k\, n}{N}}$$

DFT – Discrete Fourier Transform (Complex Representation)

- Compute X_n by dot product

$$X_n = \sum_{k=0}^{N-1} x_k \times e^{-i\frac{2\pi\, k\, n}{N}}$$

- Same as … compute a_n, b_n by dot product
  - $a_n = \left(\frac{2}{N}\right)\sum_{k=0}^{k=N}\left(Sample[k]\times\cos\left(\frac{n2\pi k}{N}\right)\right)$
  - $b_n = \left(\frac{2}{N}\right)\sum_{k=0}^{k=N}\left(Sample[k]\times\sin\left(\frac{n2\pi k}{N}\right)\right)$

DFT – Discrete Fourier Transform (Complex Representation)

- Compute X_K by dot product

$$X_n = \sum_{k=0}^{N-1} x_k \times e^{-i\frac{2\pi\, k\, n}{N}}$$

- To find the **energy** at a **particular frequency**, spin **your signal** around a circle **at that frequency**, and **average a bunch of points along that path**.

Don't let notation confuse you Expanding….

$$b_n = \left(\frac{2}{N}\right)\sum_{k=0}^{k=N}\left(Sample[k]\times\sin\left(\frac{n2\pi k}{N}\right)\right)$$

- E.g. for n=2, this says

$$b_2 = \left(\frac{2}{N}\right) dotproduct\left(Sample, \left[\sin(0), \sin\left(\frac{2*2\pi*1}{N}\right), \sin\left(\frac{2*2\pi*2}{N}\right), \ldots\right]\right)$$

$$b_2 = \left(\frac{2}{N}\right) dotproduct(Sample, [0, 0.95, 0.59, -0.59, -0.95, 0, 0.95, 0.59, -0.59, -0.95, 0])$$

- …which is dot product we performed in preclass 5

Discrete Fourier Transforms

- A DFT transforms N samples of a signal in time domain
  - into a (periodic) frequency representation with N samples
  - So we don't have to deal with real signals anymore
- We work with sampled signals (quantized in time),
  - and the frequency representation we get is also quantized frequency!

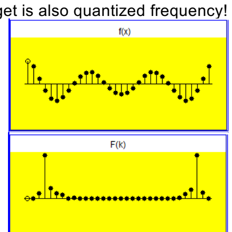
Music Signal in Discrete Time →

↓

$$DFT(x_n) = X_n = \sum_{k=0}^{N-1} x_k \times e^{-i\frac{2\pi\, k\, n}{N}}$$

↓

Music Signal "Transformed" To Frequency Domain →



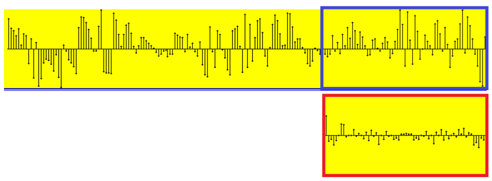
(sepwww.stanford.edu/oldsep/hale/FftLab.html)

Saving Resources

- However, N can get very large
  - e.g., with a sampling rate of 44,000Hz
  - How big is N for a 4 minute song?
- How many operations does this translate to?
  - To compute one frequency component?
  - To compute all N frequency components?
- This is not practical. Instead, we use a window of values to which we apply the transform
  - Typical size: …, 512, 1024, 2048, …

A Window Operation

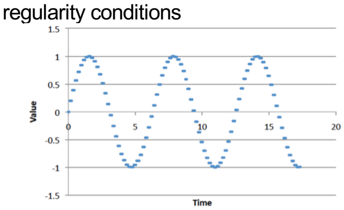
- In the example below, we traverse the signal but only look at 64 samples at a time



(sepwww.stanford.edu/oldsep/hale/FftLab.html)

Connect the Dots

- Intuition, with enough dots, not hard to "connect-the-dots" to reconstruct (understand) the continuous signal.
  - What is the continuous signal here? (Lecture 1: preclass 3)
  - Assumes certain regularity conditions
  - What is enough?



Reconstruction

- Not really connect-the-dots in time
  - (previous explanation was oversimplified)
- Recall near Nyquist rate
  - Could often miss the peak
  - Get poor sine waves
    - …look like peak moves around even if sampled above Nyquist rate
- Better reconstruction
  - Convert to frequency
    - Which can perfectly represent up to half sampling rate
  - Reconstruct from frequency basis

Big Ideas

- Can represent signals in frequency domain
  - Different basis – basis vectors of sines and cosines
- Often more convenient and efficient than time domain
  - Remember musical staff

- Can convert between time and frequency domain
  - Using a dot-product to calculate time or frequency components

$$f(t) = \frac{a_o}{2} + \sum_{n=1}^{N} [a_n \cos(nt) + b_n \sin(nt)]$$

This Week In Lab

- Lab 4:
  - You will identify Frequency components using FFT in Matlab
    - Bring headphones
- Remember:
  - Lab 3 report is due on Friday

Learn More

- ESE325 – whole course on Fourier Analysis
- ESE224 – signal processing
- ESE215, 319, 419 – reason about behavior of circuits in time and frequency domains

References

- S. Smith, "The Scientists and Engineer's Guide to Digital Signal Processing," 1997.
- https://betterexplained.com/articles/an-interactive-guide-to-the-fourier-transform/