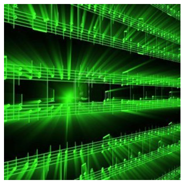
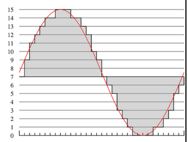
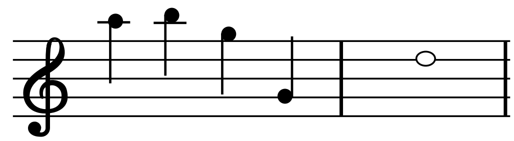




Lecture #4 – Converting from time to frequency domain

ESE 150 – DIGITAL AUDIO BASICS

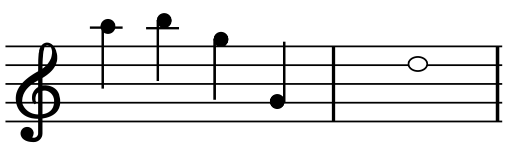
TEASER

- Play this on piano:



TEASER

- Play



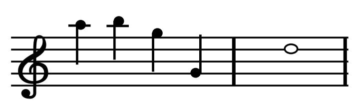
4 quarter notes

1 whole note

Cheat: A5, B5, G5, G4, D5

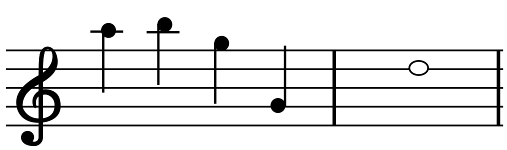
INFORMATION

- 1s / quarter note ➔ 8s of sound
- How many bits to represent 8s of sound with 16b samples and 44KHz sampling?
  - 44K Hz x 16b/sample x 8s = 5632K =5Mbits



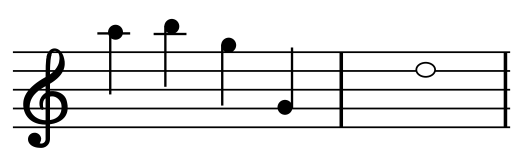
REPRESENTATION

- 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

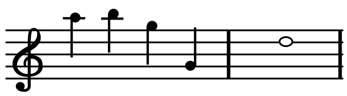


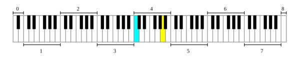
880 988 784 392 587

Frequencies in Hertz

Frequency Representation

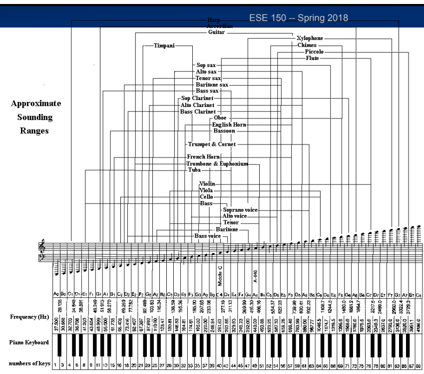
- How much information is this musical staff communicating?
- How many keys on piano? → bits/note





Hamburg Steinway D-274 Piano photo by Karl Kunde

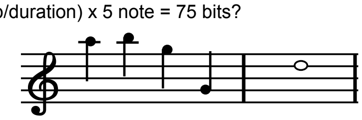
https://commons.wikimedia.org/wiki/File:D274.jpg



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 5 notes?
  - (7b/note+8b/duration) x 5 note = 75 bits?

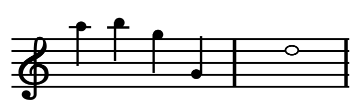


Lab 2 Postlab

- 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

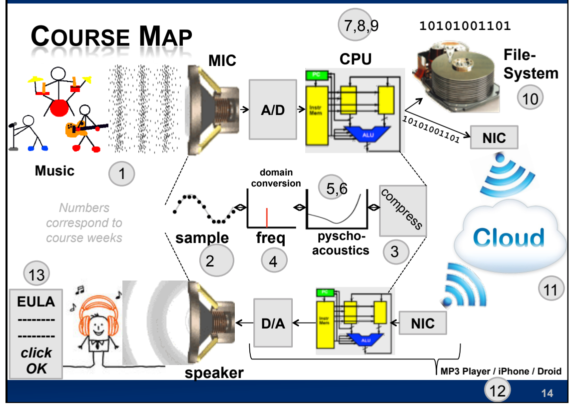
- Can represent common sounds much more compactly in frequency domain than in time-sample domain
  - Frequency domain ~ 75b
  - Time-sample domain ~ 5Mb



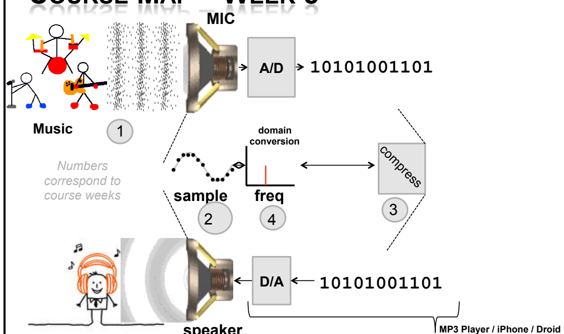
Lecture Topics

- 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
- The Discrete Fourier Transform (DFT) – can translate
  - Change of basis
- Next Lab
- References

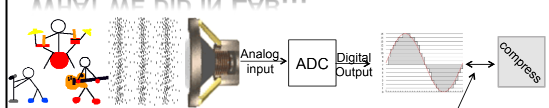
Course Map



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

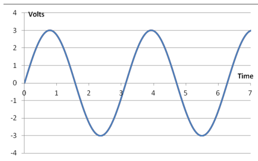
What is the Frequency Domain?

Musical Representation

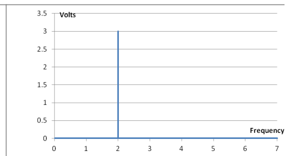
- With this compact notation
  - Could communicate a sound to pianist
  - Much more compact than 44KHz time-sample amplitudes (fewer bits to represent)
  - Represent frequencies

Time-Domain & Frequency-Domain

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



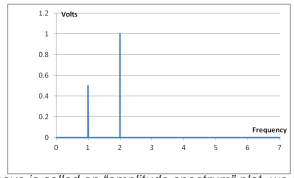
Time domain representation



Frequency domain representation

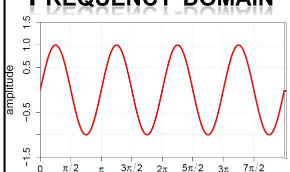
Frequency-Domain

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

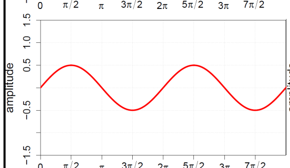


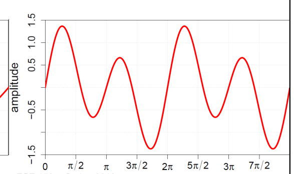
Above is called an "amplitude spectrum" plot, we also have "phase spectrum" plots in the frequency domain

Frequency-Domain



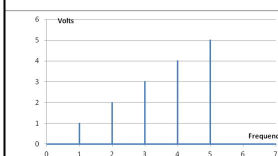
- **How about the time-domain?**
  - Plot $sin(2t)$
  - Plot $\frac{1}{2}sin(t)$
  - Sum: $s(t) = sin(2t) + \frac{1}{2}sin(t)$
  - ***Notice how it was easier to plot the frequency domain representation***

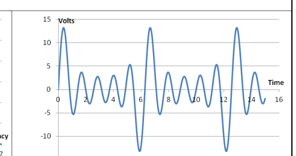




Frequency-Domain

- **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

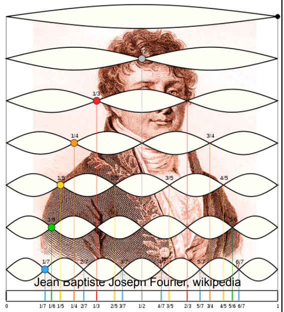
- **So far…**
  - we have seen how a signal written as:
    - a sum of sines of different frequencies
  - can have a ***frequency domain representation***
- **Each sine component…**
  - is more or less important depending on its **coefficient**
  - Example: $s(t) = \mathbf{1}sin(2t) + \frac{1}{2}sin(t)$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

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(*nt*) and cos(*nt*)
  - where *n* = 1, 2, 3, …
- These are called the ***harmonics*** of the signal



Jean Baptiste Joseph Fourier, wikipedia

FOURIER SERIES IN A NUTSHELL

- **States that any PERIODIC function can be written as a sum of sines and cosines!**
  - This is huge!
  - It also gives us a way to breakup a periodic function into its sine/cosine parts
- **Why do we care?**
  - Well if music was periodic, we could break it into its sine/cosine pieces and store only the amp/freq/phase
    - Huge storage advantage
  - But also, once we break it down, we could modify them
    - Say we want to enhance the bass or treble of a signal?
    - By increasing amplitude of bass frequencies, we'll get more bass

FOURIER SERIES IN A NUTSHELL

- **Other implications…**
  - If a signal is periodic, we can convert it from time to frequency
- **Imagine anything periodic**
  - If Earthquake vibrations could be separated into sinusoids…
    - Think vibrations at different speeds and strengths
  - We could design buildings/bridges to avoid interacting with the strongest frequency vibrations (and not just "all of them")
  - If a radio-wave is our signal and its periodic
    - We could hone in on a particular "channel" and listen only to that one…like a radio/TV!
  - It has endless applications in: communications, astronomy, geology, optics
- **There is one problem…**
  - The above examples aren't exactly periodic!
  - But we'll address that soon…first let's see the Fourier series

FOURIER SERIES – MORE FORMALLY

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

$$f(t) = a_o + \sum_{n=1}^{\infty}\left(a_n \cos(n\omega_o t) + b_n \sin(n\omega_o t)\right)$$

Where ω_0 is the fundamental angular frequency of $f(t)$ $\quad \omega_0 = \frac{2\pi}{T}$

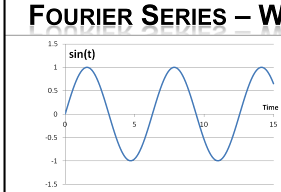
The summation is an infinite series whose first pair of terms (for n=1) involve: $\cos(\omega_0 t)$ and $\sin(\omega_0 t)$.
Higher values of n involve sine and cosine function at harmonic multiples of ω_0

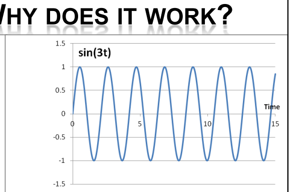
The constants: a_0, a_n , and b_n are called the Fourier coefficients of *f(t)*
Their values are determined by evaluating integral expressions involving *f(t)*

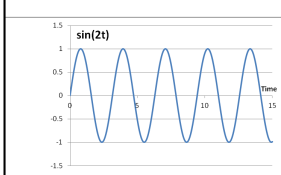
$$a_0 = \frac{1}{T}\int_0^T f(t)dt \qquad a_n = \frac{2}{T}\int_0^T f(t)\cos(n\omega_0 t)\,dt \qquad b_n = \frac{2}{T}\int_0^T f(t)\sin(n\omega_0 t)\,dt$$

a_n and b_n act as the "weights" for contributions for each harmonic

FOURIER SERIES – WHY DOES IT WORK?



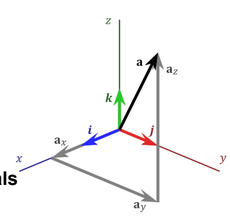




The cos(*nx*) and sin(*nx*) functions form an orthogonal basis:
- allow us to represent any periodic signal by taking a *linear combination* of the basis functions without interfering with one another
- AKA: superposition works!

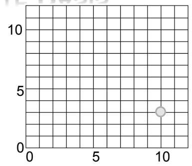
ORTHOGONAL BASIS

- **What is an orthogonal basis?**
- **Example: 3D space**
  - **Basis**: [**i j k**]
  - **Linear combination**: *x***i** + *y***j** + *z***k**
  - Coordinate representation: [*x y z*]
- **Example: Quadratic polynomials**
  - **Basis**: [**x**^{2} **x 1**]
  - **Linear combination**: *a***x**^{2} + *b***x** + *c*
  - Coordinate representation: [*a b c*]
- **All vectors in the space can be represented as sequences of coordinates, or ordered coefficients of the basis vectors**
  - This is what the Fourier Series does…
    - just uses cos(nt) and sin(nt) as its "Basis"



Fourier Series has Infinite Basis…

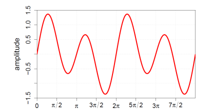
- Consider the xy-plane:
- We can address a point by its two coordinates (x,y)
  - the grey point is located at (10, 3)



- In Fourier series, we have a basis with *infinite* dimension!
  - so we sum up over an infinite number of ***harmonics***
  - harmonics are 1, cos(t), cos(2t), cos(3t), …, sin(t), sin(2t), …

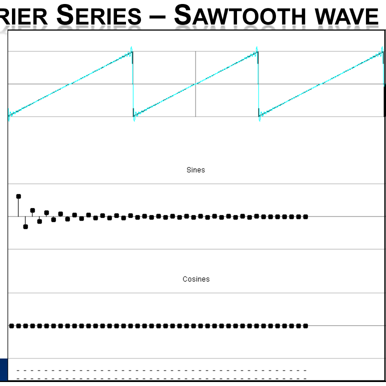
Fourier Series has Infinite Basis…

- Although we have an infinite number of basis elements, we don't need to use all of them
- The components with the largest coefficients are the most significant
  - Recall: $s(t) = 1sin(2t) + \frac{1}{2}sin(t)$



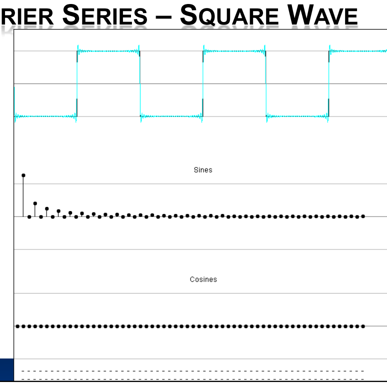
- More components we add, the closer to function we get.
  - i.e.: As $n \rightarrow \infty$, $error \rightarrow 0$
- Let's show some examples now!

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

- Close Encounters Mothership
- https://www.youtube.com/watch?v=S4PYI6TzqYk

THE FOURIER TRANSFORM

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

PRECLASS 4

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

- **Compute Dot Products with frequencies**
  - (equivalently, compute Fourier Components)

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

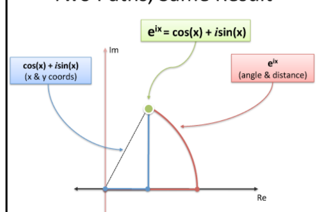
i	0	1	2	3	4	5	6	7	8	9	10
k=1	0.00	0.59	0.95	0.95	0.59	0.00	-0.59	-0.95	-0.95	-0.59	0.00
k=2	0.00	0.95	0.59	-0.59	-0.95	0.00	0.95	0.59	-0.59	-0.95	0.00
k=3	0.00	0.95	-0.59	-0.59	0.95	0.00	-0.95	0.59	0.59	-0.95	0.00
k=4	0.00	0.59	-0.95	0.95	-0.59	0.00	0.59	-0.95	0.95	-0.59	0.00
k=5	0.00	0.00	0.00	0.00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

OBSERVE

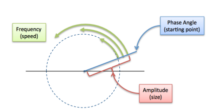
- **Can identify frequencies with dot product**
  - Identifying projection onto each basis vector in Fourier Series
- **Works because they are orthogonal**
- **Performing a change of basis**
  - From time-sample basis
  - To Fourier (sine, cosine) basis

EULER'S FORMULA

Two Paths, Same Result



Describing A Circular Path



Another choice of bases.
Change of basis.

Exponential Fourier Series

Fourier Series:

$$f(t) = a_o + \sum_{n=1}^{\infty}\left(a_n \cos(n\omega_o t) + b_n \sin(n\omega_o t)\right)$$

Euler's Identity:

$$\cos n\omega_0 t = \frac{1}{2}(e^{jn\omega_0 t} + e^{-jn\omega_0 t}),$$

$$\sin n\omega_0 t = \frac{1}{j2}(e^{jn\omega_0 t} - e^{-jn\omega_0 t}).$$

Euler's Identity in the Fourier Series:

$$f(t) = a_0 + \sum_{n=1}^{\infty}\left[\frac{a_n}{2}(e^{jn\omega_0 t} + e^{-jn\omega_0 t}) + \frac{b_n}{j2}(e^{jn\omega_0 t} - e^{-jn\omega_0 t})\right]$$

$$= a_0 + \sum_{n=1}^{\infty}\left[\left(\frac{a_n - jb_n}{2}\right)e^{jn\omega_0 t} + \left(\frac{a_n + jb_n}{2}\right)e^{-jn\omega_0 t}\right]$$

$$= a_0 + \sum_{n=1}^{\infty}[\mathbf{c}_n e^{jn\omega_0 t} + \mathbf{c}_{-n} e^{-jn\omega_0 t}].$$

$$f(t) = \sum_{n=-\infty}^{\infty} \mathbf{c}_n e^{jn\omega_0 t},$$

(exponential representation)

$$\mathbf{c}_n = \frac{1}{T}\int_{-T/2}^{T/2} f(t)\, e^{-jn\omega_0 t}\, dt.$$

$$\mathbf{c}_n = \frac{a_n - jb_n}{2}$$

and

$$\mathbf{c}_{-n} = \frac{a_n + jb_n}{2} = \mathbf{c}_n^*.$$

Fourier Series/Transform/DFT

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

Fourier Transform – for continuous functions:

Output: Frequency Domain

Input: Time Domain

$$\mathcal{F}\{g(t)\} = G(f) = \int_{-\infty}^{\infty} g(t) e^{-i2\pi ft}\, dt$$

$$\mathcal{F}^{-1}\{G(f)\} = g(t) = \int_{-\infty}^{\infty} G(f) e^{i2\pi ft}\, df$$

Visual Demonstration

- **Instead of thinking only in sinusoids…**
  - Complex sinusoids can be thought of as cycles…
  - Let's visual complex #s a bit…
  - http://betterexplained.com/examples/fourier/

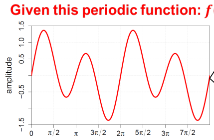
Frequency-Domain

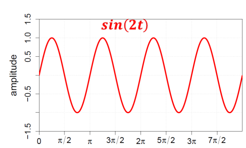
- **How to make a song appear "periodic:"**
  - Treat the entire song as 1 period of a very complicated sinusoid!
  - *This is the assumption of the Fourier Transform*

Fourier Series in general form:

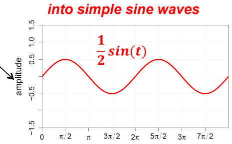
$$f(t) = \frac{a_o}{2} + \sum_{n=1}^{N}[a_n \cos(nt) + b_n \sin(nt)]$$

Given this periodic function: $f(t)$





Fourier Series says it can be broken into simple sine waves



Fourier Series for this f(t):

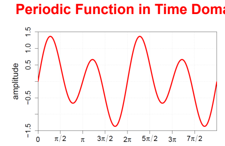
$$f(t) = \sin(2t) + \frac{1}{2}\sin(t)$$

Fourier Series – decomposes f(t) into a "weighted sum" of sine waves

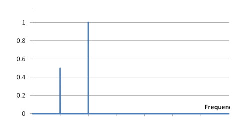
We only need to store amplitude and frequency of each harmonic of the fund. freq.

Time Domain to Frequency Domain

Periodic Function in Time Domain



Periodic Function in Frequency Domain



Fourier Series for this f(t):

$$f(t) = \sin(2t) + \frac{1}{2}\sin(t)$$

Store only frequency information:

1st Fundamental amplitude: 0.5
2nd Fundamental amplitude: 1
And so on…

Alternate view of Fourier Series
…converting signals in the "time domain" to the "frequency domain"

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: $x(t) \rightarrow x_n$
  - The signal contains **N** samples: $0 \le n \le N-1$
  - Again Euler's formula helps us: $e^{it} = \cos(t) + i\sin(t)$?

Discrete Time to Frequency:

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

The frequency-domain signal: X_k also has N-1 elements: $0 \le k \le N-1$

Frequency back to Time:

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

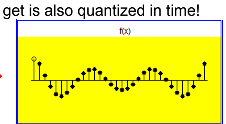
The time-domain signal: x_n can be reconstructed from X_k

ESE 150 -- Spring 2018

Discrete Fourier Transforms

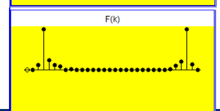
- **A DFT transforms N samples of a signal in time domain**
  - into a (periodic) frequency representation with N samples
  - So we don't have to deal with real signals anymore
- **We work with sampled signals (quantized in time),**
  - and the frequency representation we get is also quantized in time!

Music Signal in Discrete Time →



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

Music Signal "Transformed" To Frequency Domain →



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

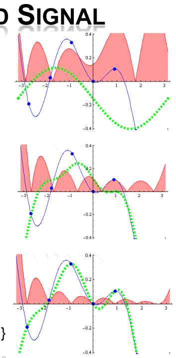
Discrete Fourier Transforms

- **A smaller sampling period means:**
  - **-> more points to represent the signal**
    - **larger N**
  - **-> more harmonics used in DFT**
    - **N harmonics**
  - **-> Smaller error compared to actual analog signal we capture/produce**
- **DFTs are extensively used in practice, since computers can handle them**

Approximating the Sampled Signal

- **A signal sampled in time can be approximated *arbitrarily* closely from the time-sampled values**
- **With a DFT, each sample gives us knowledge of one harmonic**
- **Each harmonic is a component used in the reconstruction of the signal**
- **The more harmonics we use, the better the reconstruction**

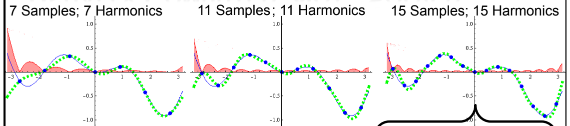
{ Cos[0*t*], Sin[1*t*], Cos[1*t*] , Sin[2*t*], Cos[2*t*] , Sin[3*t*], Cos[3*t*] }



ESE 150 -- Spring 2018

Usually Computed, Not "Solved"

7 Samples; 7 Harmonics — 11 Samples; 11 Harmonics — 15 Samples; 15 Harmonics



7 Samples; 7 Harmonics:

t	v
-2.7	-0.2
-1.8	0
-0.9	0.3
0	0
0.9	0.1
1.8	-0.4
2.7	-0.9

DFT →

f	A
0	-1.1
1s	-0.8
1c	1.3
2s	0.2
2c	0.7
3s	0.5
3c	-0.3

11 Samples; 11 Harmonics:

t	v
-2.9	0
-2.3	-0.3
-1.7	0.1
-1.1	0.4
-0.6	0.2
0	0
0.6	0.1
1.1	0.1
1.7	-0.3
2.3	-0.9
2.9	-0.7

DFT →

f	A
0	-1.3
1s	-1.5
1c	1.9
2s	0.1
2c	0.9
3s	1.1
3c	-0.5
4s	0.6
4c	-0.2
5s	0.3
5c	-0.1

15 Samples; 15 Harmonics:

t	v
-2.9	0.1
-2.5	-0.3
-2.1	-0.2
-1.7	0.1
-1.3	0.3
-0.8	0.3
-0.4	0.1
0	0
0.4	0
0.8	0.1
1.3	0
1.7	-0.3
2.1	-0.7
2.5	-0.9
2.9	-0.7

DFT →

f	A
0	-2
1s	-2
1c	2.3
2s	-0
2c	1.1
3s	1.5
3c	-0
4s	0.8
4c	-0
5s	0.6
5c	0
6s	0.5
6c	-0
7s	0.5
7c	0.1

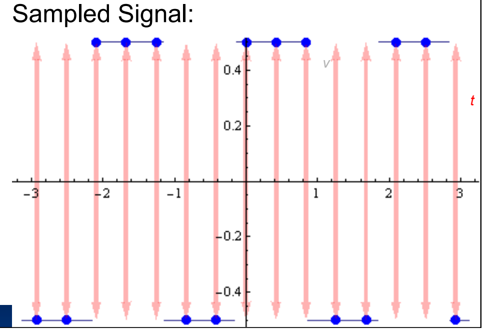
ESE 150 -- Spring 2018

Yet Another Sampled (Real) Signal

Measured Data:

t	v
$-\frac{14\pi}{15}$	$-\frac{1}{2}$
$-\frac{4\pi}{5}$	$-\frac{1}{2}$
$-\frac{2\pi}{3}$	$\frac{1}{2}$
$-\frac{8\pi}{15}$	$\frac{1}{2}$
$-\frac{2\pi}{5}$	$\frac{1}{2}$
$-\frac{4\pi}{15}$	$-\frac{1}{2}$
$-\frac{2\pi}{15}$	$-\frac{1}{2}$
0	$\frac{1}{2}$
$\frac{2\pi}{15}$	$\frac{1}{2}$
$\frac{4\pi}{15}$	$\frac{1}{2}$
$\frac{2\pi}{5}$	$-\frac{1}{2}$
$\frac{8\pi}{15}$	$-\frac{1}{2}$
$\frac{2\pi}{3}$	$\frac{1}{2}$
$\frac{4\pi}{5}$	$\frac{1}{2}$
$\frac{14\pi}{15}$	$-\frac{1}{2}$

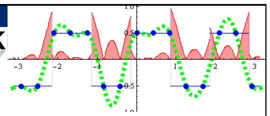
Sampled Signal:

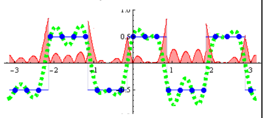


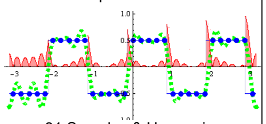
Some Signals Dislike Some Harmonics

- **Approximate Reconstruction**
  - although always achievable
  - may require a lot of samples
  - to get good performance

Sometimes time is better than frequency!


15 Samples & Harmonics


21 Samples & Harmonics

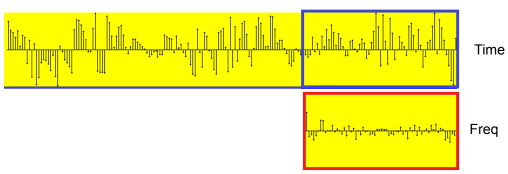

31 Samples & Harmonics

Saving Resources

- **However, N can get very large**
  - e.g., with a sampling rate of 48,000Hz
  - 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**



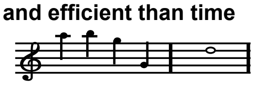
(sepwww.stanford.edu/oldsep/hale/FftLab.html)

Reconstruction

- **Not really connect-the-dots in time**
- **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
- **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/