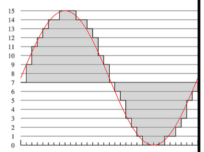
Penn Engineering

ESE



Lecture #1 – A2D, D2A

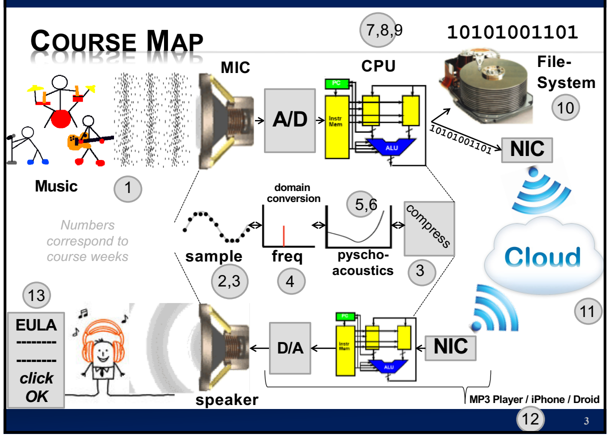
ESE 150 – DIGITAL AUDIO BASICS

Based on slides © 2009--2020 Koditschek & DeHon
Additional Material © 2014--2017 Farmer

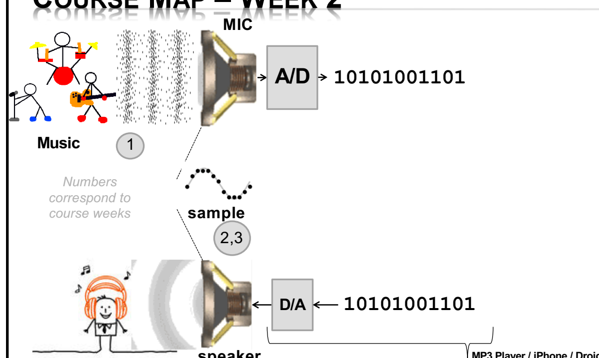
LECTURE TOPICS

- Where are we on course map?
- Sound / Sound Pressure
- Sampling & Quantization
- <interlude>
- Effects of Quantization
- Limits of Sampling
- System Capacity
- Summary
- References

COURSE MAP



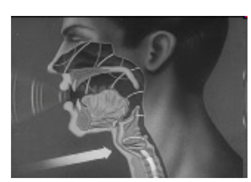
COURSE MAP – WEEK 2



SOUND WAVES

INTRODUCTION TO SOUND

- Sound is a pressure wave

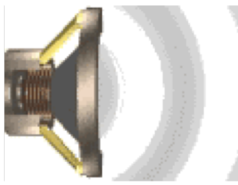


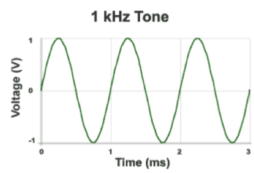
http://www.archive.org/details/SoundWavesAn

Week 1: Introduction to Sound Waves

Cycle = 1 iteration of sine wave
Hertz (Hz) = 1 cycle per second

1kHz = 1000 cycles/s

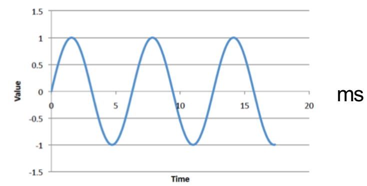




Source: http://www.mediacollege.com/audio/01/sound-waves.html

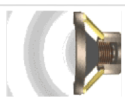
Preclass 1 and 2

- **Frequency of sine wave?**



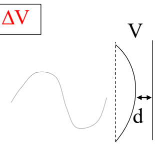
- **Relationship between period and frequency?**

Week 1: Pressure to Voltage



- Microphones convert pressure to voltage
  - (speakers/headphones voltage to pressure)
  - Physical position to voltage

$\Delta d \rightarrow \Delta C \rightarrow \Delta V$



- Reason as parallel place capacitor
  - ESE 112 or PHYS 151

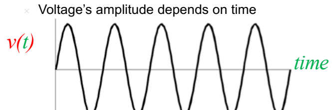
$$C = \frac{\varepsilon A}{d}$$

$$Q = CV \qquad V = \frac{Q}{C}$$

Signals

We need to quantify some things

- **What is a signal?**
  - Something that carries information
  - A description of how one parameter depends on another
    - Common Engineering Example:
      - Voltage that varies with time
        - E.g. Amplitude of voltage changes as time moves forward
      - Time = ***independent*** variable (x-axis): time
        - Depends on nothing!
      - Voltage = ***dependent*** variable (y-axis): $v(t)$
        - Voltage's amplitude depends on time

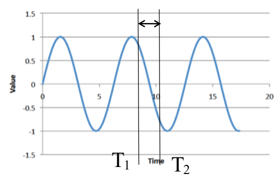


We need to quantify some things

- **Most signals encountered in nature…**
  - …are "**continuous**" / analog
    - Continuous range of values (any real #)
    - Examples: 1) Light intensity that changes with distance
    - 2) Voltage that varies over time $v(t)$
      - *We will see in lab this week: MUSIC signal represented with voltage*
    - 3) Chemical reaction rate that depends on temperature
  - as opposed to "**non-continuous**" / discrete signals
    - Only a discrete range of values possible (limited subset of real #s)
    - How a computer must represent signals
      - Fundamental unit of information: **bit**
      - Cannot represent all possible real #'s
      - Uses binary digit (bit) to represent #'s:
        - 1-bit, represents 2 things…2-bits, represents 4 things
        - What's the generalization? (n-bits → how many things?)

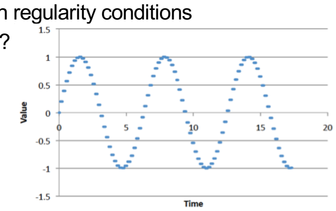
Big Question

- **How represent and process *continuous* information on a digital computer with *finite* memory?**
  - Note: continuous means signal may take on infinite number of values between any T_1 and T_2



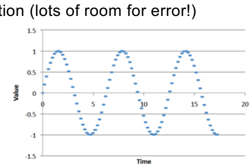
Connect the Dots

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



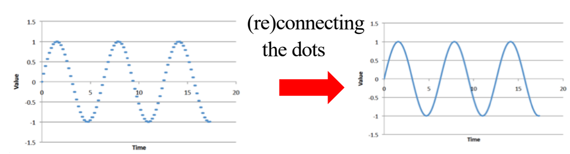
Definitions

- **Analog-to-Digital (ADC) Conversion**
  - Process of converting <u>*continuous*</u> signal to <u>*discrete*</u> signal
  - Going from analog to digital "domain"
  - Often called: <u>digitization</u>
  - Use a subset of real #'s to represent all real #'s
    - Involves a lot of approximation (lots of room for error!)
- **…collecting** the dots



Definitions

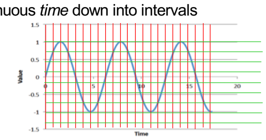
- **Digital-to-Analog (DAC) Conversion**
  - Process of converting <u>*discrete*</u> signal to <u>*continuous*</u> signal
  - Going from digital to analog "domain"
  - Converting "bits" to a continuous waveform
    - Our MP3/Music players do this all the time (will do in lab)



Sampling & Quantization

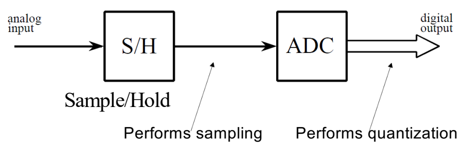
ADC – Sampling & Quantization

- **Analog-to-Digital (ADC) Conversion**
  - Converting analog (continuous) signal to digital signal
  - Digitization process has two important aspects:
    - *1) Sampling*
      - Converting ***independent*** variable of signal from continuous to discrete
      - e.g.: breaking continuous *time* down into intervals



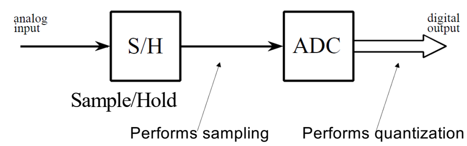
- *2) Quantization*
  - Converting ***dependent*** variable of signal from continuous to discrete
  - e.g.: breaking continuous *voltage* down into levels

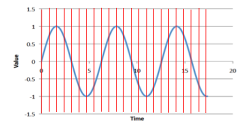
ADC – Broken into Two Parts



Figures from reading: *The Scientist and Engineer's Guide to Digital Signal Processing*, By Steven W. Smith

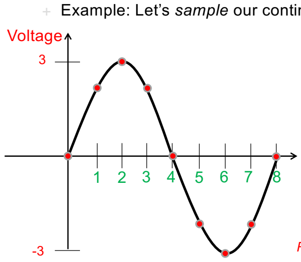
ADC – Broken into Two Parts





ADC – Sampling

- Analog-to-Digital (ADC) Conversion
  - ***Sampling***: breaking independent variable (time) into intervals
  - Example: Let's *sample* our continuous signal @ 1 ms intervals:



What is our sampling rate?
(How many samples per second?)

1000 samples per second
1 kiloSamples / s

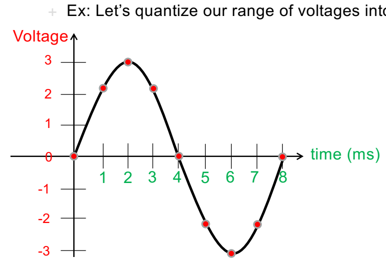
What is frequency of this signal?

$freq = \frac{1}{period} = \frac{1}{8ms} = 125\ Hz$

Spoiler alert:
Relationship between sample rate & frequency!

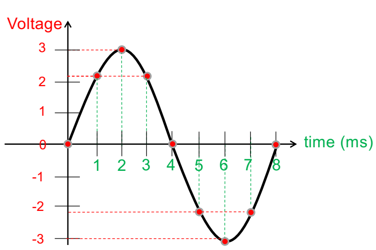
ADC – Quantization

- Analog-to-Digital (ADC) Conversion
  - ***Quantization***: breaking dependent variable (voltage) into levels
  - Ex: Let's quantize our range of voltages into 7 levels (1 Volt each)



ADC – Sampling & Quantization

- Analog-to-Digital (ADC) Conversion
  - **Let's collect our samples at the quantized levels**

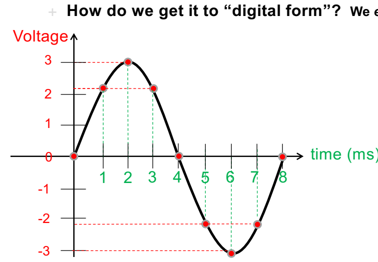


Samples:
{ 0 ms, 0 Volts }
{ 1 ms, 2 Volts }
{ 2 ms, 3 Volts }
{ 3 ms, 2 Volts }
{ 4 ms, 0 Volts }
{ 5 ms, -2 Volts }
{ 6 ms, -3 Volts }
{ 7 ms, -2 Volts }
{ 8 ms, 0 Volts }

Notice, we are rounding! Error is inherent in this process

ADC – Digital Conversion / Encoding

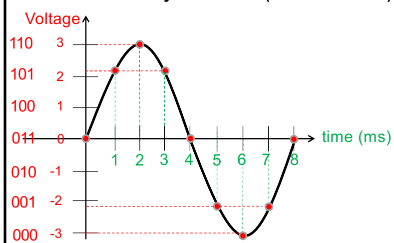
- Analog-to-Digital (ADC) Conversion
  - **We've converted something continuous into discrete form**
  - **How do we get it to "digital form"?** **We *encode* it…(map to another format**



Samples:
{ 0 ms, 0 Volts }
{ 1 ms, 2 Volts }
{ 2 ms, 3 Volts }
{ 3 ms, 2 Volts }
{ 4 ms, 0 Volts }
{ 5 ms, -2 Volts }
{ 6 ms, -3 Volts }
{ 7 ms, -2 Volts }
{ 8 ms, 0 Volts }

ADC – Digital Conversion / Encoding

- **Analog-to-Digital (ADC) Conversion**
  - We have 7 discrete voltages, # of bits to represent 7 things?
  - 3-bits! Why? $2^{3\text{-bits}} = 8$ (1 unused state)



Samples:	Binary Encoding:
{ 0 ms, 0 Volts }	011
{ 1 ms, 2 Volts }	101
{ 2 ms, 3 Volts }	110
{ 3 ms, 2 Volts }	101
{ 4 ms, 0 Volts }	011
{ 5 ms, -2 Volts }	001
{ 6 ms, -3 Volts }	000
{ 7 ms, -2 Volts }	001
{ 8 ms, 0 Volts }	011

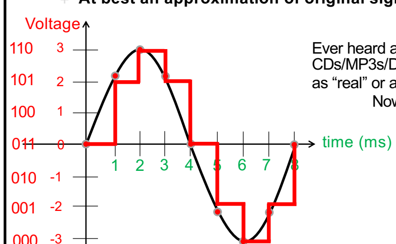
Encoding: mapping data from one form to another (not always conversion)

ADC – Storing the Data

- **Analog-to-Digital (ADC) Conversion**
  - What do we store? Just the encoded bits:
    - Our digitized signal: {011, 101, 110, 101, 011, 001, 000, 001, 011}
    - It is now discrete & in digital format, store bits in MP3 player!
  - Why can we avoid storing the time?
    - It's repetitive! Just store sampling rate: 1 kilo-samples/ sec
    - Later, if we wish to restore signal, each "sample" occurred at 1ms
  - In this example:
    - Sampling rate: 1 k-samples/sec
    - Resolution: 3-bits
    - Our digitized signal: {011, 101, 110, 101, 011, 001, 000, 001, 011}

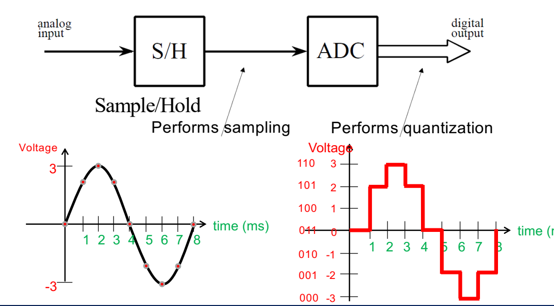
ADC – An Approximation at Best

- **Analog-to-Digital (ADC) Conversion**
  - Continuous analog signal overlaid with discrete digital signal
  - At best an approximation of original signal



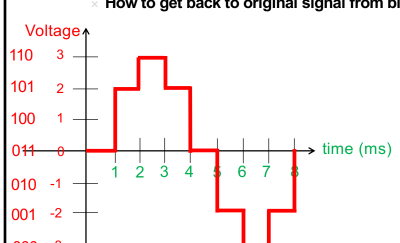
Ever heard an audiophile complain:
CDs/MP3s/Digital music doesn't sound as "real" or as good old records?
Now you know why!

ADC – Broken into Two Parts



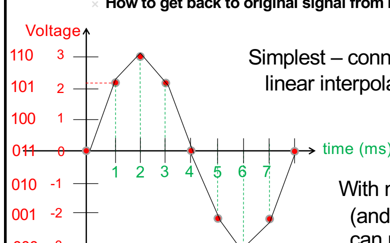
ADC – An Approximation at Best

- **Digital-to-Analog (DAC) Conversion**
  - Process of converting *discrete* signal to *continuous* signal
    - How to get back to original signal from bits?



ADC – An Approximation at Best

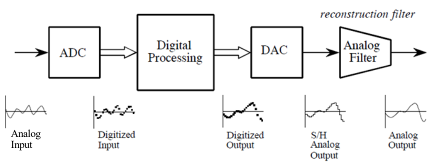
- **Digital-to-Analog (DAC) Conversion**
  - Process of converting *discrete* signal to *continuous* signal
    - How to get back to original signal from bits?



Simplest – connect the dots
linear interpolation between points.

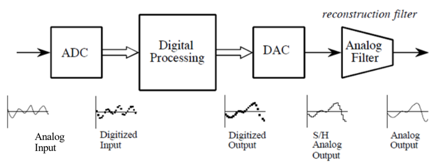
With more sophistication
(and typical circuits)
can make smoother links.

ADC / DAC – The full picture



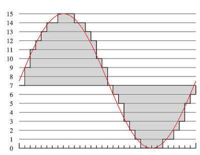
Figures from reading: *The Scientist and Engineer's Guide to Digital Signal Processing*, By Steven W. Smith

ADC / DAC – The full picture



- **Formally:**
  - Analog input signal that varies with time: $s(t)$
  - Signal processing algorithm to digitize analog input signal:
    - f[i]=Round(*s*(i*T))
      - T is sample period
  - Digitized signal produced by f[ ]: $s_f(t)$

PCM



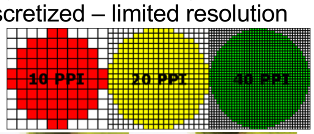
- **Quantization & Sampling Technique described:**
  - Called Pulse-Code-Modulation (PCM)
    - Patented in 1943
    - PCM process is the ADC process
    - Developed for telecommunications

Penn Engineering ESE

Interlude: 2D Images

Same Phenomena in Images

- World continuous
- Digital images on projector, TV, paper (even photographs) are discretized – limited resolution
  - This projector…



http://www.morefill.com/wpve
http://dslrphotographytutorials.c... ng
http://blog.inkjetwholesale.com.au/wp-content/uploads/2015/07/DPI-comparison.png

Apple Retina Display

- **Why called retina?**
- **Claim (goal):** as much resolution as you have in your retina (at typical viewing distance)
- We cannot see pixels, because our eyes are themselves discrete!

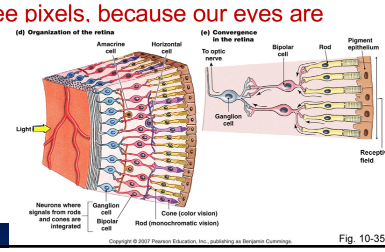


Fig. 10-35

Apple Retina Display

- **Why called retina?**
- **Claim (goal):** as much resolution as you have in your retina (at typical viewing distance)
- We cannot see pixels, because our eyes have discrete photo receptors (rods, cones)
- Human eye resolution 0.5 arc-minute (0.02 degrees)
  - Around 300 DPI at 20 inches

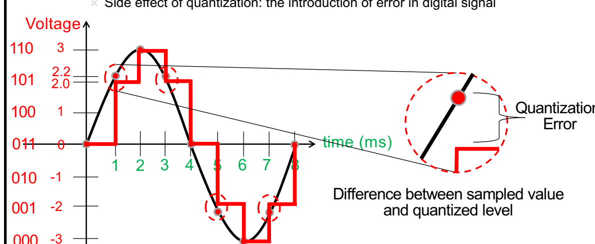
Effects of Sampling and Quantization

Noise -- ``Formal'' Definition

- **Noise – difference between our ideal signal and the actual signal**
  - The one that we actually hear
  - The one that shows up when we transmit data
  - The one we store or reconstruct
- **Sometimes will see**
  - R(t) = S(t) + n(t)
    - Noise n(t) is added to the ideal signal S(t)
  - Or, equivalently:
    - n(t)=S(t)-R(t)

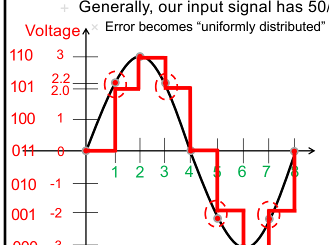
Quantization Error

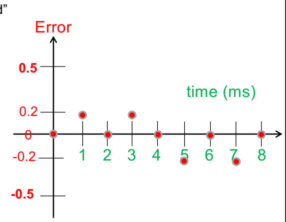
- **In example, quantization algorithm required us to round**
  - At sample time, t=1ms, input signal was: 2.2V
  - It was lower than 2.5V, we rounded down to quantized level of 2.0V
    - Side effect of quantization: the introduction of error in digital signal



Quantization Error

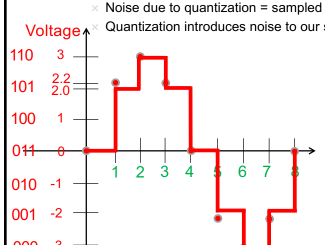
- **How much error?**
  - In our case, we round up if equal to or above ½ a level…
    - …round down if below ½ a level
  - Generally, our input signal has 50/50 chance of being above/below
    - Error becomes "uniformly distributed"

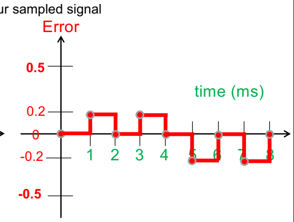




Quantization Error / Noise

- **How much error?**
  - Looking at the plot of error, looks random
  - Sets up a way for us to model quantization error as noisy signal
    - Noise due to quantization = sampled signal (red dots) – quantized signal (red line)
    - Quantization introduces noise to our sampled signal





Quantization Error / LSB

- **"Least Significant Bit"**
  - How much value is added with each addition of the least-significant bit?

    LSB = InputRange/(Levels-1)

  - What is LSB for our example (3V to -3V, 7 levels)?
  - Also known as: *resolution* of ADC
    - What is the smallest difference the ADC can represent
  - Quantization error = ± LSB/2

45

Least Significant Bit

- **Quantization error = ± LSB/2**
- **The LSB is our resolution**
  - Like the DPI
  - Or, more accurately, the distance between pixels (1/DPI)
- **How close do we get to the original signal?**
- **What is the magnitude of the error introduced due to quantization?**

46

Quantization Error / Design

- **Why model quantization error as noise?**
- **There is always noise present**
  - Something other than the signal we intend
  - Wires, electronics, background
  - Not gaining much if quantization noise < other noise
- **Quantization adds noise**
  - Reduce by increasing sampling, increasing resolution
  - More levels → bits → makes more expensive
  - Increase until reach desired noise level
    - Until other sources dominate quantization noise
- **SNR = Signal-to-Noise Ratio**
  - How much larger is the signal compare to noise?
  - Mean (average) value of signal / std. dev. of noise
  - Usually what we are optimizing in the system (including ADC)

47

Engineering

- **"An Engineer can do for a dime what anyone else can do for a dollar."**
- **Engineering is about optimization and efficiency**
- **Bits are costly**
- **Engineer ask:** how few bits can I use without sacrificing quality?
- **Engineering is about tradeoffs**
  - Quality vs. Cost

48

Limits of Sampling

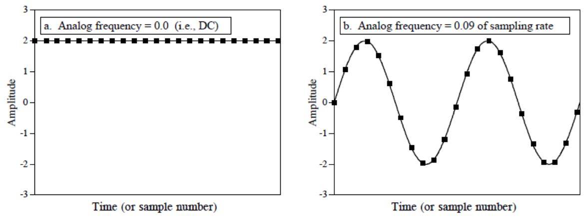
49

Sampling

- **Definition of proper sampling**
  - If you can exactly reconstruct analog signal from samples,
  - you have done the sampling properly
    - Essentially: you have captured the key information from the signal to process can be reversed
- **Milestone of digital signal processing (DSP):**
  - Nyquist-Shannon Theorem (next week)
    - Tells us our sampling rate should be:
      - twice the frequency of the signal!

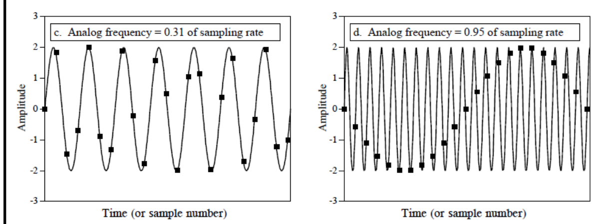
50

Sampling Rates: How much is necessary?



Figures from reading: *The Scientist and Engineer's Guide to Digital Signal Processing*, By Steven W. Smith

Sampling Rates: How much is necessary?



Signal can appear to change frequency if not adequately sampled
called: aliasing (next week)

Figures from reading: *The Scientist and Engineer's Guide to Digital Signal Processing*, By Steven W. Smith

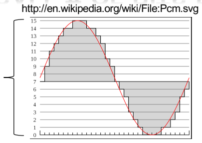
System Capacity

Quantization, Sampling, Capacity

- **Quantization and Sampling**
  - Play enormous role in determining storage capacity of digital system
  - # of quantization levels ➔ # of bits per sample
    - Increasing resolution of ADC, reduces quantization noise…
    - But also increases amount of data we must store for each sample
  - Sampling rate = how often we collect # of bits per sample
    - Typically sampling rate = twice frequency of signal (next week)
    - Increasing the rate, increases the amount of data to store!

Recall # of bits for typical song

http://en.wikipedia.org/wiki/File:Pcm.svg



http://en.wikipedia.org/wiki/File:Compact_disc.svg

- **Sampling rate & resolution effect on storage**
  - Compact Disks: 16bits at 44KHz
    - How many bits is a typical 3-minute song? (preclass 4)

$$\left(44{,}000\ \frac{samples}{1\ sec}\right)\left(16\ \frac{bits}{sample}\right)\left(60\ \frac{sec}{1\ min}\right)\left(3\ \frac{min}{song}\right) = 15.1\ \frac{MB}{song}$$

Problem Decomposition

- **Powerful Engineering technique**
  - Formulate a parameterized solution strategy
  - Then identify the right parameters
- **Divides the problem**
- **Here**
  - Strategy of sampling and quantization
  - Then identify the right sampling rate, quantization level
- **Once have strategy, reduces to a well-defined optimization problem**
- **Parameterization admits to tuning for tradeoffs**

This Week in Lab

- **Look at waveforms of Sound**
- **Sample and Quantize sounds waveforms**

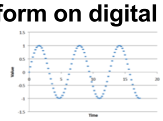
- **Remember:**
  - Read Lab
  - Work Prelab
  - Bring USB Flash Drive to lab
  - Partner assignments…out by Monday morning

Learn More

- **ESE215 – basic analog circuitry, RLC circuits, simple filters**
  - Including why typical circuits give smoother (not linear) connection of dots
- **ESE568 – Mixed Signal Integrated Circuits**
  - Build A2D, D2A

Big Ideas

- **Approximate continuous waveform on digital media by**
  - Discretize in all dimension
  - For audio: in time and amplitude
    - Sample in time; quantize voltage



- **Allows us to store audio signal as sequence of bits**
- **Reconstruct by "connecting-the-dots"**
  - If our dots are frequent enough to represent the signal
- **Introduce error → noise**
  - Reason about tolerable (or noticeable) noise

Admin

- **Reading for today, next Wednesday on syllabus**
- **In Lab (Detkin) on Monday**
  - Lab posted
  - Read lab, work prelab
  - Bring USB flash drive
- **Remember feedback**

References

- **S. Smith, "The Scientists and Engineer's Guide to Digital Signal Processing," 1997.**
- **Wikipedia,** http://en.wikipedia.org/wiki/Analog-to-digital_converter
- **Wikipedia:** http://en.wikipedia.org/wiki/Pulse-code_modulation