Penn Engineering ESE

Lecture #1 – A2D, D2A

## ESE 150 – Digital Audio Basics

Based on slides © 2009--2018 Koitschek & DeHon
Additional Material © 2014--2017 Farmer

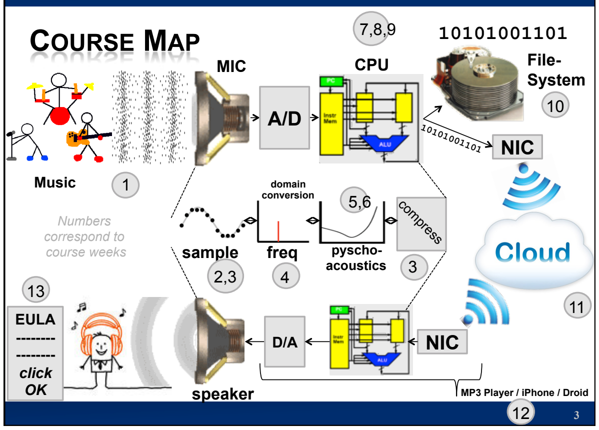


## Lecture Topics

- Where are we on course map?
- Sound / Sound Pressure
- Sampling & Quantization
- Effects of Quantization
- Limits of Sampling
- System Capacity
- Copying vs. Sampling
- Summary
- References

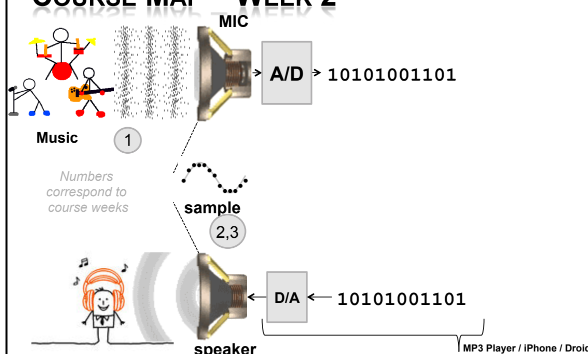


## Course Map



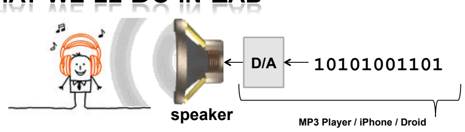



## Course Map – Week 2






## What We'll Do in Lab



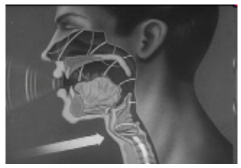

- **Microphones / Speaker:**
  - Convert pressure waves to electric signal (and reverse)
- **Sound (in nature):**
  - a "pressure wave" that changes air molecules w/ respect to time
- **Sound (in our machine):**
  - a "voltage wave" that changes amplitude w/ respect to time
- **MP3 player converted digital signal back to an analog signal**
  - We were able to look at "voltage wave" on an oscilloscope



## Sound Waves

## Introduction to Sound

- **Sound is a pressure wave**

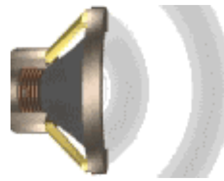


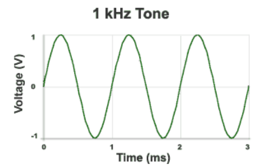
http://www.archive.org/details/SoundWavesAn



## Introduction to Sound Waves

Recall from Math: Hertz (Hz) = 1 cycle per second
1kHz = 1000 cycles/s

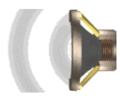





Source: http://www.mediacollege.com/audio/01/sound-waves.html

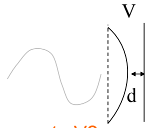


## Pressure to Voltage



- Microphones convert pressure to voltage
  - (speakers/headphones voltage to pressure)
- Parallel plate capacitor with pressure activated plate
  - over short time scale:
    - charge (Q) is conserved (not changing)
    - if *d* changes, what happens to C?

$$C \propto \frac{A}{d}$$




    - and when C changes, Q holds, what happens to V?

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

Δd → ΔC → ΔV

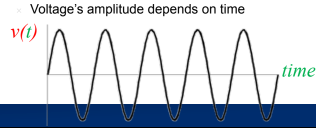


## Signals



## 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



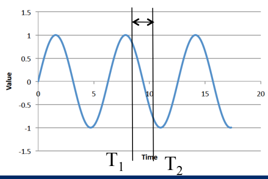



## We need to quantify some things

- **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 temp
  - 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 1 thing…2-bits, represents 4 things
        - What's the generalization? (n-bits → how many things?)

## Big Question

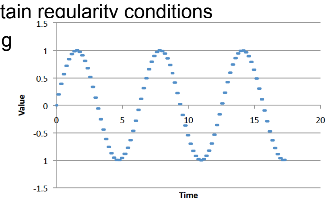
- **How represent and process continuous information on a digital computer with finite memory?**
  - Note: continuous means signal may take on infinite number of values between any $T_1$ and $T_2$






## Connect the Dots

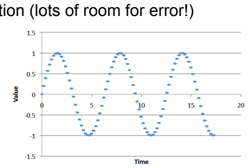
- **Intuition, with enough dots not hard to "connect-the-dots" to reconstruct (understand) the continuous signal.**
  - What is the continuous signal here?
  - Assumes certain regularity conditions
  - What is enoug






## Definitions

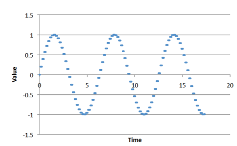
- **Analog-to-Digital (ADC) Conversion**
  - Process of converting *continuous* signal to *discrete* signal
  - Going from analog to digital "domain"
  - Often called: digitization
  - Use a subset of real #'s to represent all real #'s
    - Involves a lot of approximation (lots of room for error!)
-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



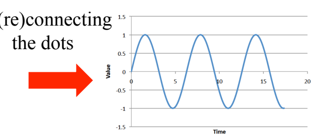



## Definitions

- **Digital-to-Analog (DAC) Conversion**
  - Process of converting *discrete* signal to *continuous* 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)




(re)connecting the dots






## Sampling & Quantization



## When it comes to music, what is sampling?

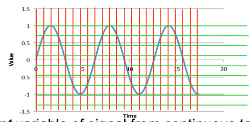
When this guy stole a bassline from David Bowie, "*sampling*" became part of the vernacular…

…but that's not what we're talking about!

Sampling? **No, No Baby**

## ADC – Sampling & Quantization

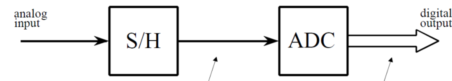
- **Analog-to-Digital (ADC) Conversion**
  - Converting analog (continuous) signal to digital signal
  - Digitization process has two important aspects:
    - *1) Sampling*
      - Converting ***independent*** variable of signal from continuous to discrete
      - e.g.: breaking continuous *time* down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



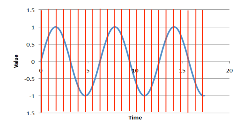

    - *2) Quantization*
      - Converting ***dependent*** variable of signal from continuous to discrete
      - e.g.: breaking continuous *voltage* down into levels



## ADC – Broken into two parts



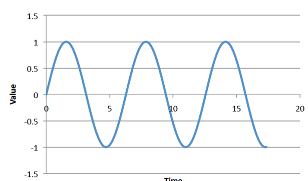

Performs sampling Performs quantization






## Preclass 1 and 2

- **Frequency of sine wave?**

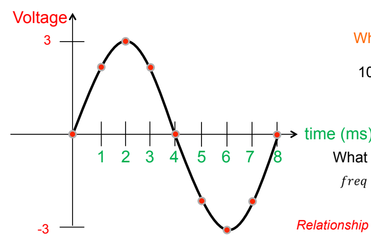



- **Relationship between period and frequency?**



## 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?

1000 samples per second
1 kiloSamples / s

What is frequency of this signal?

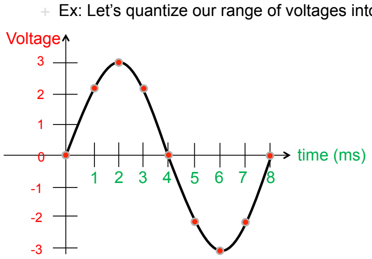
$$freq = \frac{1}{period} = \frac{1}{8ms} = 125\ Hz$$

*Spoiler alert:*
*Relationship between sample rate & frequency!*



## ADC – Quantization

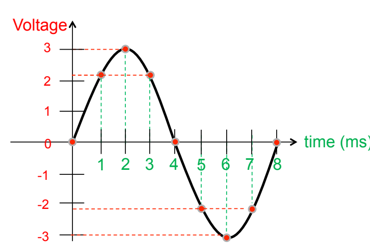
- **Analog-to-Digital (ADC) Conversion**
  - ***Quantization***: breaking dependent variable (voltage) into levels
  - Ex: Let's quantize our range of voltages into 7 levels (1 Volt each)






## ADC – Sampling & Quantization

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

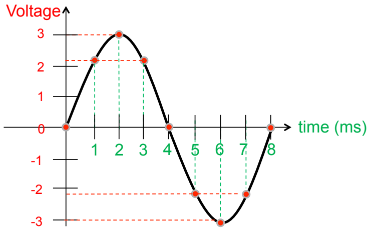



Samples:
{ 0 ms, 0 Volts }
{ 1 ms, 2 Volts }
{ 2 ms, 3 Volts }
{ 3 ms, 2 Volts }
{ 4 ms, 0 Volts }
{ 5 ms, -2 Volts }
{ 6 ms, -3 Volts }
{ 7 ms, -2 Volts }
{ 8 ms, 0 Volts }

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**

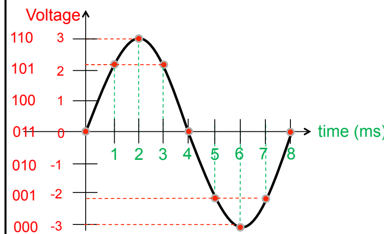



Samples:
{ 0 ms, 0 Volts }
{ 1 ms, 2 Volts }
{ 2 ms, 3 Volts }
{ 3 ms, 2 Volts }
{ 4 ms, 0 Volts }
{ 5 ms, -2 Volts }
{ 6 ms, -3 Volts }
{ 7 ms, -2 Volts }
{ 8 ms, 0 Volts }



## 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) 

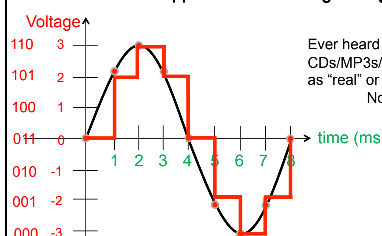
## ADC – Storing the Data

- **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

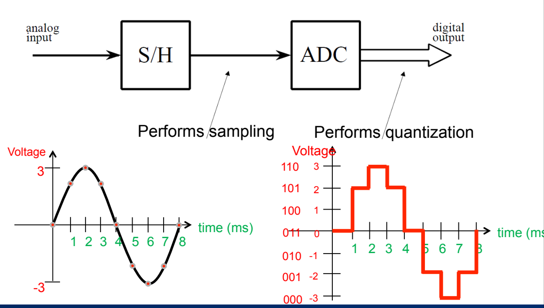
- **Analog-to-Digital (ADC) Conversion**
  - **Continuous analog signal overlaid with discrete digital signal**
  - **At best an approximation of original signal**




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



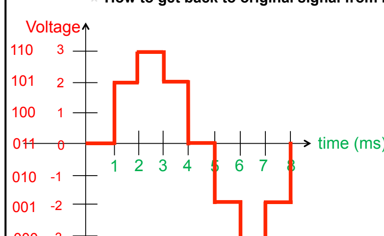
## ADC – Broken into two parts





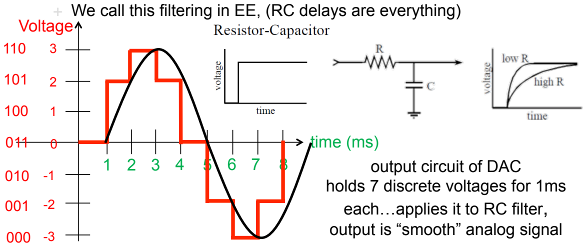

## ADC – An Approximation at Best

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



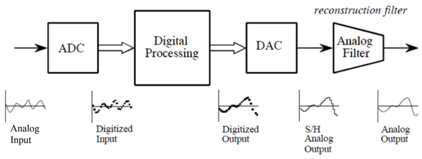
## DAC - Filtering

- **Digital-to-Analog (DAC) Conversion**
  - What a capacitor does?
  - What happens when apply voltage across a resistor/capacitor?
  - We call this filtering in EE, (RC delays are everything)





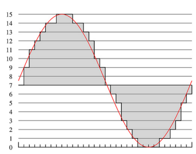

## ADC / DAC – The full picture




- **Formally:**
  - Analog input signal that varies with time: $s(t)$
  - Signal processing algorithm to digitize analog input signal: $f[s(t)]$
  - 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

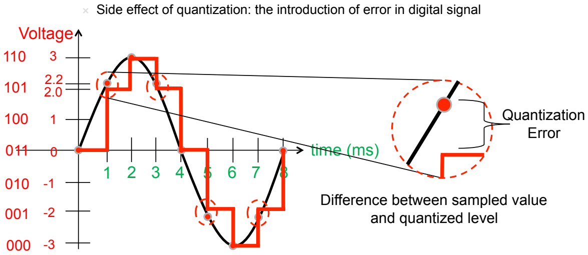


## Effects of Quantization



## Quantization Error

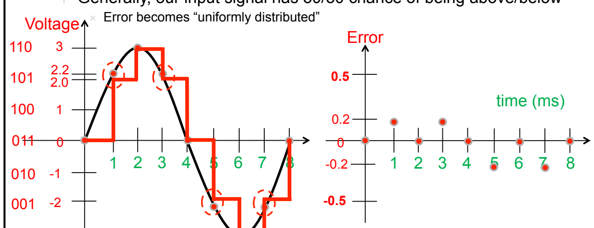
- **In example, quantization algorithm required us to round**
  - At sample time, t=1ms, input signal was: 2.2V
  - It was lower than 2.5V, we rounded down to quantized level of 2.0V
    - Side effect of quantization: the introduction of error in digital signal





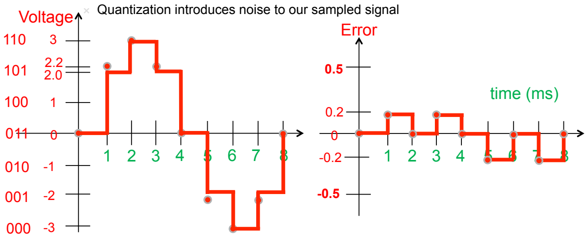

## Quantization Error

- **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 a lot like random noise
  - 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?
    $$\frac{InputRange}{NumLevels - 1}$$
  - What is LSB for our example (3V to -3V, 7 levels)?
  - Also known as: resolution of ADC
  - Quantization error = ± LSB



## Quantization Error / Design

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



## Limits of Sampling

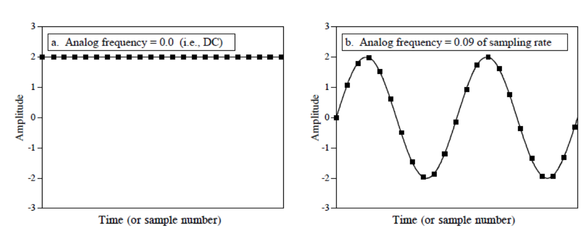


## Sampling

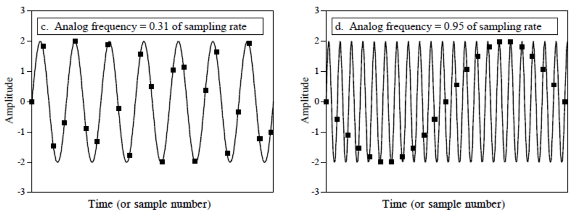
- **Definition of proper sampling**
  - If you can exactly reconstruct analog signal from samples,
  - you have done the sampling properly
    - Essentially: you have captured the key information from the signal to process can be reversed
- **Milestone of digital signal processing (DSP):**
  - Nyquist-Shannon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



## Sampling Rates: How much is necessary?



## Sampling Rates: How much is necessary?




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

## System Capacity



## Quantization, Sampling, Capacity

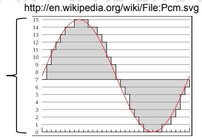
- **Quantization and Sampling**
  - Play enormous role in determining storage capacity of digital system
  - # of quantization levels = # of bits per sample
    - Increasing resolution of ADC, reduces quantization noise…
    - But also increases amount of data we must store for each sample
  - Sampling 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

Discrete Voltage Levels



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)

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



## Calculating Capacity

- **How many songs can we fit on a CD?**
  - How many MB can a CD hold?
    - 650 MB (for Data)
    - About 780 MB (for Audio)
  - If 3 minute song = 15.1 MB…
    - 1 minute audio ~ 5.04MB
    - But, we record in stereo: (Right, Left audio tracks)
      - 5.04 MB x 2 = 10.07 MB per minute
    - How many minutes do we get on a 780MB CD?

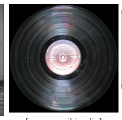


## Copying vs. Sampling

## Recording: From Copying to Sampling

- **Analog Recording: copying**
  - from tinfoil
  - to vinyl
  - to metal filings
  - … to history …

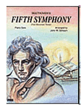

[source: wikipedia]


[source: wikipedia]

[source: wikipedia]

- **Digital Recording: sampling**
  - In time
  - In value


[source: wikipedia]



[source: sheetmusicplus]





## Compact Disc (CD)

- **CD (late 20th Century)**
  - ~ 600 MB capacity
  - ~ 1 hour of music
  - "Transparent" sound quality
    - Indistinguishable from best analog recording
    - To "almost all" humans


[source: wikipedia]

## This week in Lab

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

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



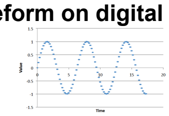
## Learn More

- **ESE215 – basic analog circuitry, RLC circuits, simple filters**
- **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



- **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**
  - Read lab, work prelab
  - Bring USB flash drive
- **Remember feedback**
  - Including office hour polling

## 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


55