Penn Engineering

ESE

Lecture #3 – Quantization

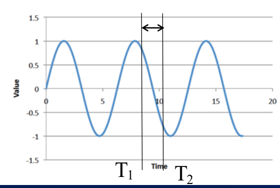
ESE 150 – Digital Audio Basics

Lecture Topics

- Reminder: Sampling and Quantization
- Expressing Mathematically
- <interlude: image resolution>
- Effects of Quantization
- System Capacity
- Summary
- References

Big Question

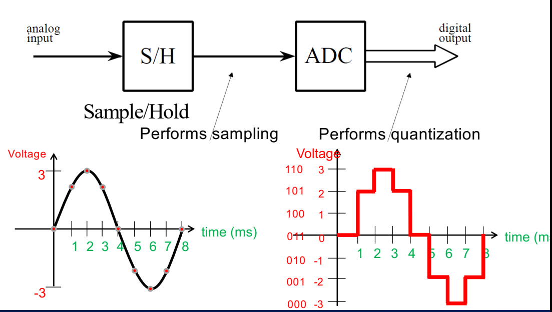
- **How represent and process *continuous* information on a digital computer with *finite* memory?**
  - Note: continuous means signal may take on infinite number of values between any T_1 and T_2



Strategy

- **Sample at periodic time intervals**
  - Discretize independent variable
- **Quantize to discrete levels**
  - Discretize the value of the dependent variable

ADC – Broken into two parts



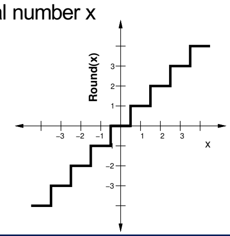
Problem Decomposition

- **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
- **Convergent:** limit of infinite samples, levels
- **Once have strategy, reduces to a well-defined optimization problem**
- **Parameterization admits to tuning for tradeoffs**

MATHEMATICAL EXPRESSION

ROUND

- **Rounding – select nearest discrete value as approximation of continuous value**
- **For sake of concreteness, we will define:**
  - Round(x) – nearest integer to real number x
    - Round(0.7) = 1
    - Round(-0.1) = 0
    - Round(2.4999) = 2
    - Round(1.50001) = 2
- **What is Round(3.3) ?**



QUANTIZE

- **We will quantize to some level L**
- **Define as number of values between integers**
- **So, we have L steps of 1/L between integers**
  - (or only represent every L'th integer if L<1)
- **In terms of Round**
  - $Quantize_L(x) = Round(L*x)/L$
  - E.g. $Quantize_8(0.7)=Round(8*0.7)/8=6/8=0.75$

PRECLASS 1

		0.333	1.246	-1.123
L=4	$Quantize_L(x)$			
	$QuantizationError_L(x)$			
L=16	$Quantize_L(x)$			
	$QuantizationError_L(x)$			

$Quantize_L(x) = Round(L*x)/L$

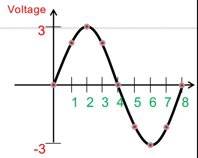
BITS

- **If we quantize to L levels per integer**
- **Represent values between integers**
  - Max
  - Min
- **How many bits required per quantized value?**

BITS PER QUANTIZED VALUE

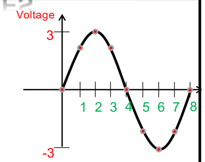
- **Bits = $\lceil \log_2((Max-Min)*L+1) \rceil$**

Sample Values



- When sample at time interval T
  - (for frequency = 1/T)
- We collect the points at times:
  - 0, T, 2T, 3T, 4T, …
- For signal f(t), we are collecting values:
  - f(0), f(T), f(2T), f(3T), f(4T), …
- Or in general, we are collecting the points
  - f(i*T) for i=0 to MaxSamples (or MaxTime/T)

Sample and Quantize Values

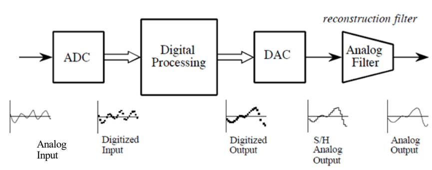


- When sample at time interval T
  - (for frequency = 1/T)
- We collect the points at times:
  - 0, T, 2T, 3T, 4T, …
- For signal f(t), we are collecting:
  - f(0), f(T), f(2T), f(3T), f(4T), …
- If we then Quantize the values to level L
  - Quantize_L(f(0)), Quantize_L(f(T)), Quantize_L(f(2T)), …

Sample and Quantize Values

- If we then Quantize the values to level L
  - Quantize_L(f(0)), Quantize_L(f(T)), Quantize_L(f(2T)), …
- Or in general, we are collecting the points
  - Quantize_L(f(i*T)) for i=0 to MaxSamples (or MaxTime/T)
- We store them in an array (or vector) F of MaxSamples+1
  - For i from 0 to MaxSamples: F[i]= Quantize_L(f(i*T))
  - This is what you collected on Monday!

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)*L)/L
      - T is sample period
  - Digitized signal produced by $F[\ ]$: $s_f(t)$

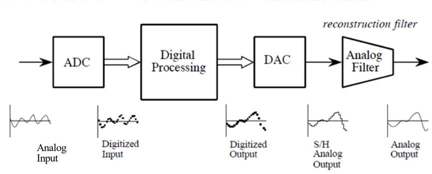
Preclass 2

Sample f(t)=sin(2π*110*t) at T=0.3ms, quantize L=10

i	0	1	2	3	4
i*T	0	0.3ms			
Value f(i*T)	0				
F[i]	0				
Quantization Error	0				

Sine goes -1 to 1: how many bits for these 5 samples?

ADC / DAC – The Full Picture



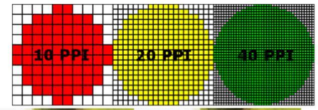
- Formally:
  - Analog input signal that varies with time: $s(t)$
  - Signal processing algorithm to digitize analog input signal:
    - F[i]=Round(s(i*T)*L)/L
      - T is sample period
  - Digitized signal produced by $F[\ ]$: $s_f(t)$

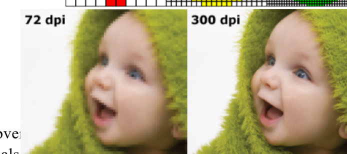
![Penn Engineering ESE]

Interlude: 2D Images

Same Phenomena in Images

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





http://www.morefill.com/wpve
http://dslrphotographytutorials.com/wp-content/uploads/2014/03/111-Compare.png
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!

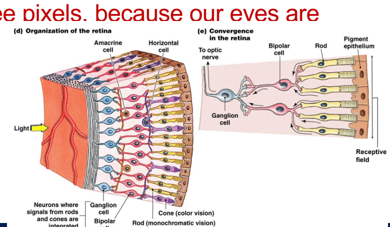


Fig. 10-35

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 (Dots-Per-Inch) at 20 inches

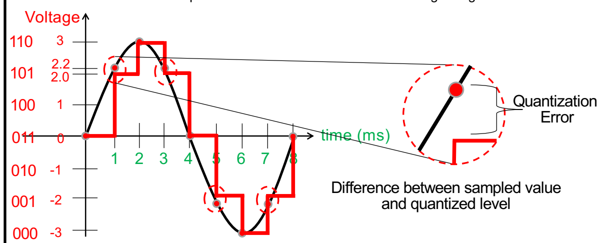
Effects of Quantization

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)
    - R(t) what we receive
  - Or, equivalently:
    - n(t)=S(t)-R(t)

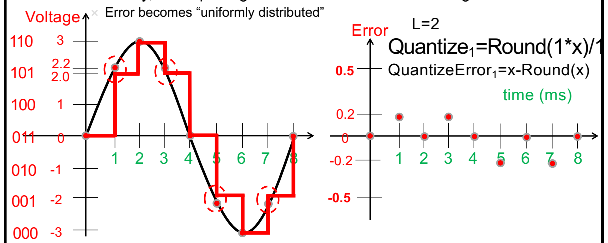
Quantization Error

- **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



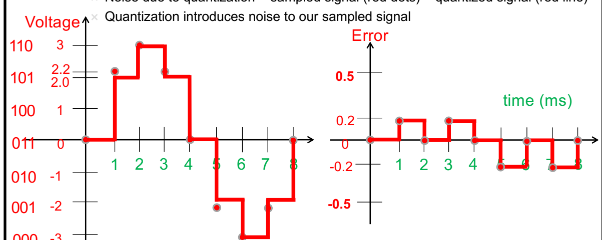
Quantization Error

- **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

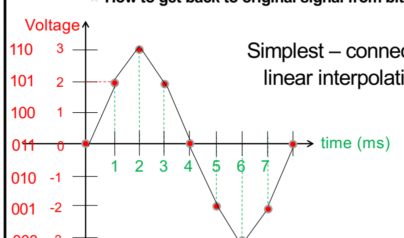


Continuous Error

- **So far, focus on error at sample points**
- **Can also define error at any point**
- **Depends on reconstruction model**
  - How estimate/create continuous from samples
- **Simple connect-the-dots model linear interpolation**

ADC – An Approximation at Best

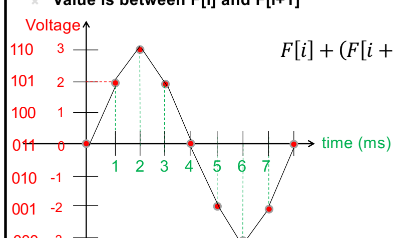
- **Digital-to-Analog (DAC) Conversion**
  - Process of converting *discrete* signal to *continuous* signal
    - **How to get back to original signal from bits?**



Simplest – connect the dots
linear interpolation between points.

Continous Reconstruction

- **What is reconstructed signal at time t?**
- **Find samples that bracket --- i: i*T<t<(i+1)*T**
- **Value is between F[i] and F[i+1]**



$$F[i] + (F[i+1] - F[i]) \times \left(\frac{t - i \times T}{T}\right)$$

Preclass 3

- Sampling and quantization error at t=1ms?
- T=0.3ms
- i?
- F[i], F[i+1]
- Reconstructed value?
- f(1ms)
- Error=f(1ms)-ReconstructedValue(1ms)

$$F[i] + (F[i+1] - F[i]) \times \left(\frac{t - i \times T}{T}\right)$$

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 →(L) bits → makes more expensive
    - Bits = ⌈log2((Max-Min)*L+1)⌉
  - Increase L 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)

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
- **Anyone:** Sample frequently with high resolution
- **Engineer ask:** how few bits can I use without sacrificing quality?
- Engineering is about tradeoffs
  - Quality vs. Cost

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

System Capacity

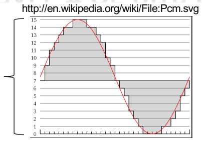
Quantization, Sampling, Capacity

- 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
    - Bits/sample = ⌈log2((Max-Min)*L+1)⌉
  - 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?

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

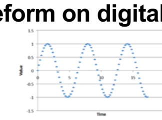
Limits of Sampling

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 (Friday)
    - Tells us our sampling rate should be:
      - twice the frequency of the signal!

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 Friday on syllabus**
- **Remember feedback**
  - Includes Lab 1
- **Office Hours**
  - T7, W2, R8
  - Zoom links on Piazza (web, canvas)
- **Lab 1 Reports due Friday**

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