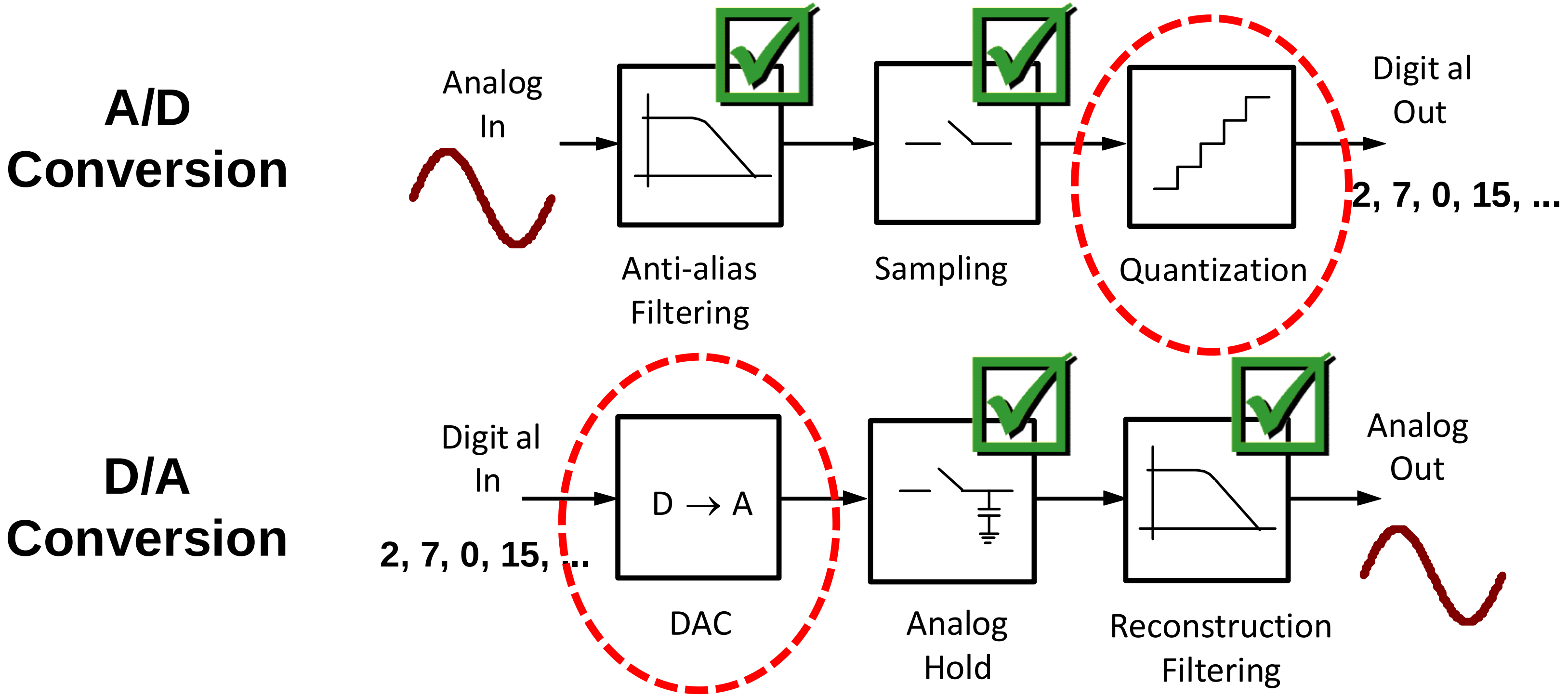
EE 240C
Analog-Digital Interface Integrated Circuits

Amplitude Quantization

Recap

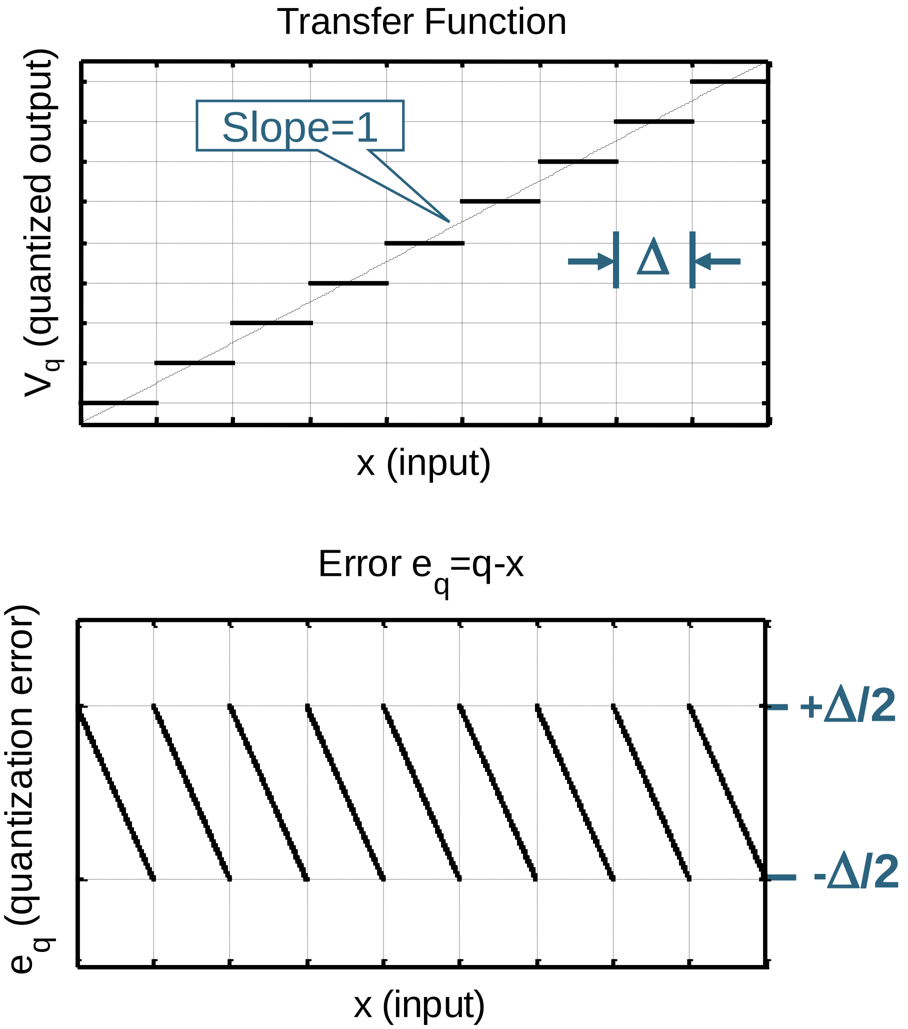


A/D Conversion

D/A Conversion

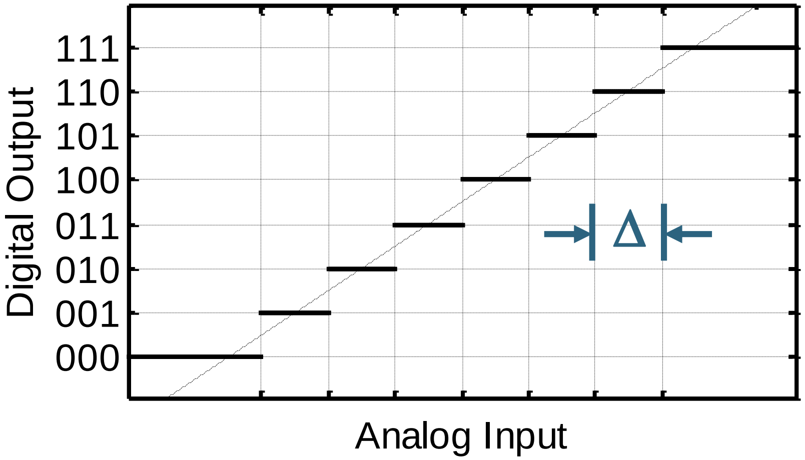
- Next, look at
  - Transfer functions of quantizer and DAC
  - Model for quantization error

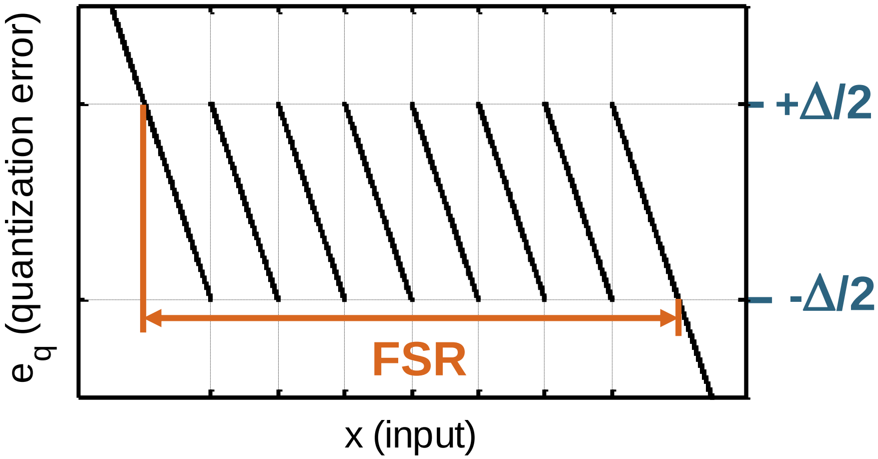
Ideal Quantizer



- Quantization step Δ
- Quantization error has sawtooth shape
  - Bounded by –Δ/2, +Δ/2
- Ideally
  - Infinite input range and infinite number of quantization levels
- In practice
  - Output is a digital word with finite number of bits (all digital signals are quantized, even quadruple precision floating point)
  - Analog signals are not quantized. Is analog more accurate than digital?

Quantizer with Finite Input Range

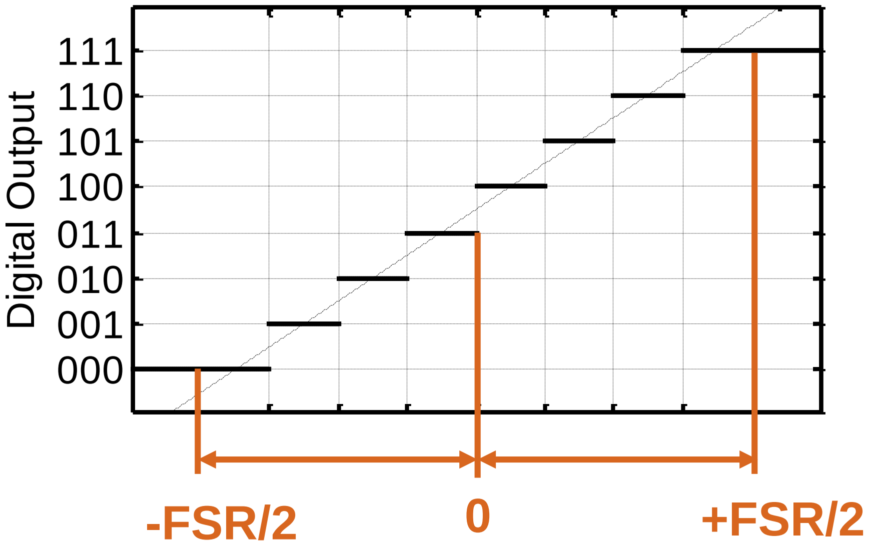




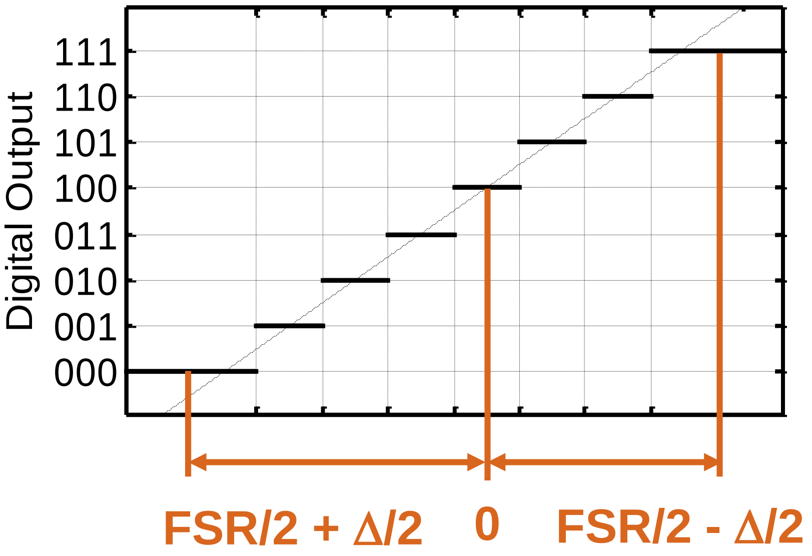
- Example: B=3
  - $2^3=8$ distinct output codes
  - Diagram on the left shows "straight-binary encoding"
  - See e.g. Analog Devices "The Data Conversion Handbook" for other encoding schemes
    - http://www.analog.com/media/en/training-seminars/design-handbooks/Data-Conversion-Handbook/Chapter2.pdf
- Quantization error grows out of bounds beyond code boundaries
- We define the full scale range (FSR) as the maximum input range that satisfies $|e_q| \leq \Delta/2$
  - Implies that $\text{FSR}=2^B \cdot \Delta$

Technicality: Quantizer Offset

Mid-Rise Quantizer

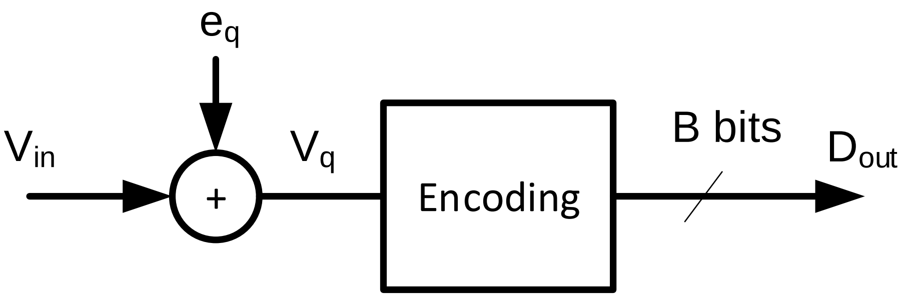


Mid-Tread Quantizer



- Many ADCs have offsets >> Δ
- In many applications small offsets are inconsequential
- “Yet another option”: unipolar quantizer

Quantizer Model



- Encoding block determines how quantized levels are mapped into digital codes
- Note that this model is not meant to represent an actual hardware implementation
  - Its purpose is to show that quantization and encoding are conceptually separate operations
  - Changing the encoding of a quantizer has no interesting implications on its function

Quantization “Noise”

- In principle, the quantization error e_q is a deterministic signal:
  - The input uniquely defines the error
  - Strictly it cannot be characterized as “noise”, which implies random nature
  - But it is kind of random …
  - And we have a pretty powerful way of dealing with (thermal and 1/f) noise in electronic circuits – wouldn’t it be nice to do use the same approach with quantization errors?

- We are in luck: in practice, quantization error closely resembles noise provided that
  - Input spreads over several quantizer levels
  - Is busy
  - Quantizer does not clip

- Let’s look at a few specific situations and then derive an expression for the variance of the quantization “noise” and its spectrum

EE 240C
Analog-Digital Interface Integrated Circuits

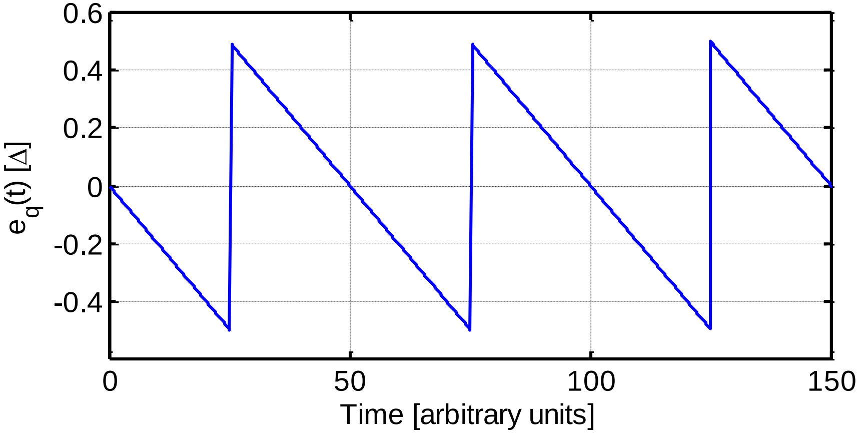
Quantization Error: Statistics

Quantization Error Model

- Two aspects
  - How much noise power does quantization add to samples?
  - How is this noise power distributed in frequency?
- Quantization error is a deterministic function of the signal
  - Should be able answer above questions using a deterministic analysis
  - But, unfortunately, such an analysis strongly depends on the chosen signal and can be very complex
- Strategy
  - Build basic intuition using simple deterministic signals
  - Next, abandon idea of deterministic representation and revert to a "general" statistical model (to be used with caution!)

Example 1: Ramp Input

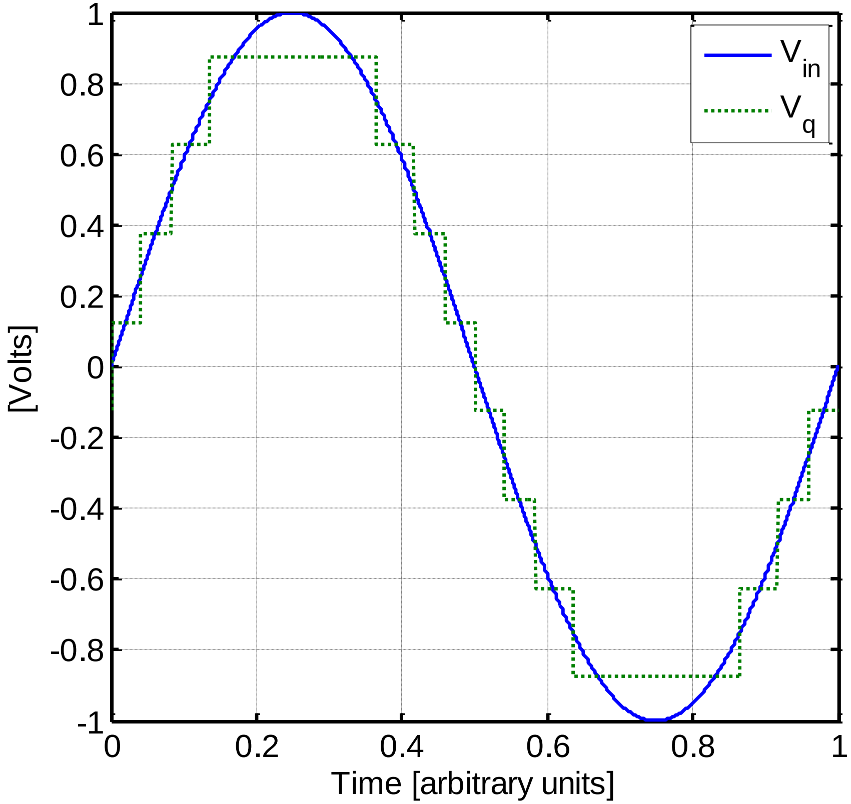
- Applying a ramp signal at the input of the quantizer results in the following time domain waveform for the quantizat

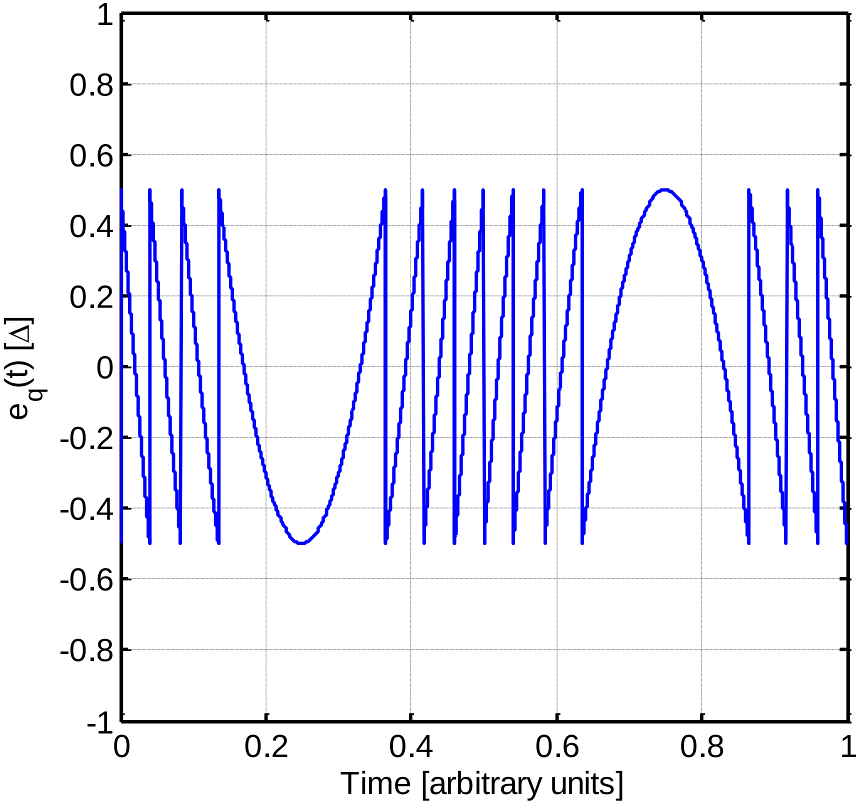


- What is the average power of this waveform?
- Integrate over one period

$$\overline{e_q^2} = \frac{1}{T}\int_{-T/2}^{T/2} e_q^2(t)dt \qquad e_q(t) = \frac{\Delta}{T}\cdot t \qquad \boxed{\therefore\ \overline{e_q^2} = \frac{\Delta^2}{12}}$$

Example 2: Sine Wave Input

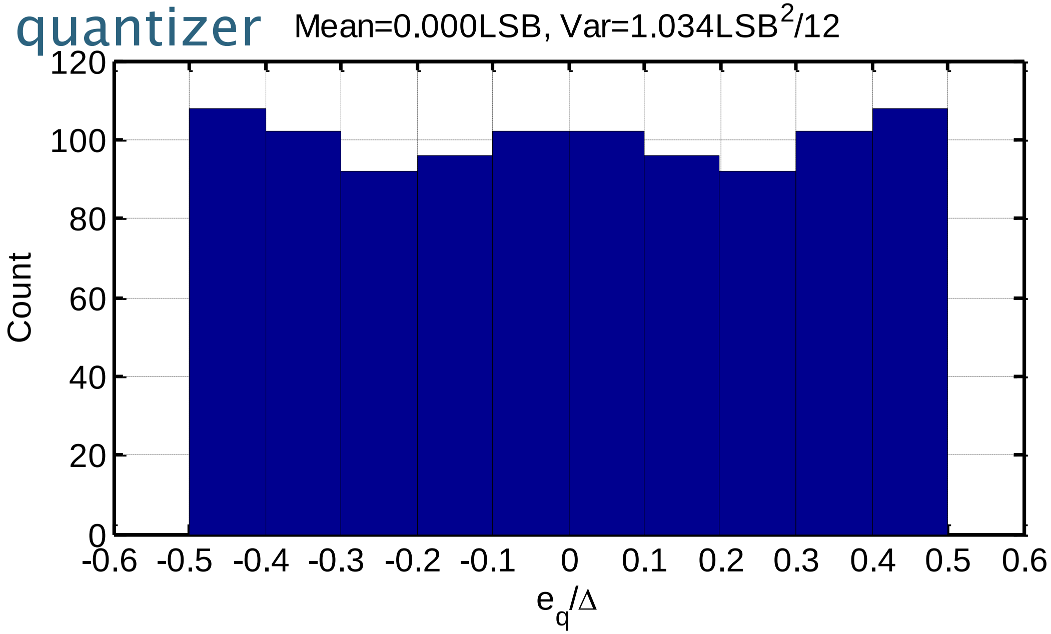




- Integration is not straightforward
- Let’s ask a computer to gain insight into what’s going on …

Quantization Error Histogram

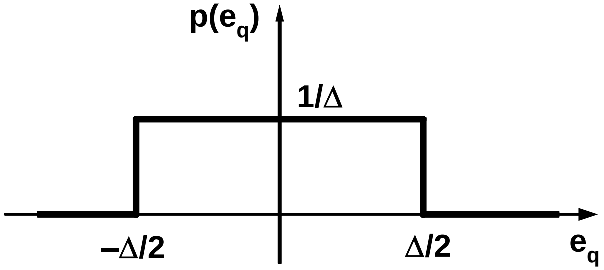
- Sinusoidal input signal with f_{sig}=101Hz, sampled at f_s=1000Hz
- 8–bit quantizer



- Distribution is "almost" uniform
- Approximate average power by integrating uniform distribution

Statistical Model of Quantization Error

- Assumption: $e_q(x)$ has a uniform probability density
- This approximation holds reasonably well in practice when
  - Signal spans large number of quantization steps
  - Signal is "sufficiently active"
  - Quantizer does not overload (clip)

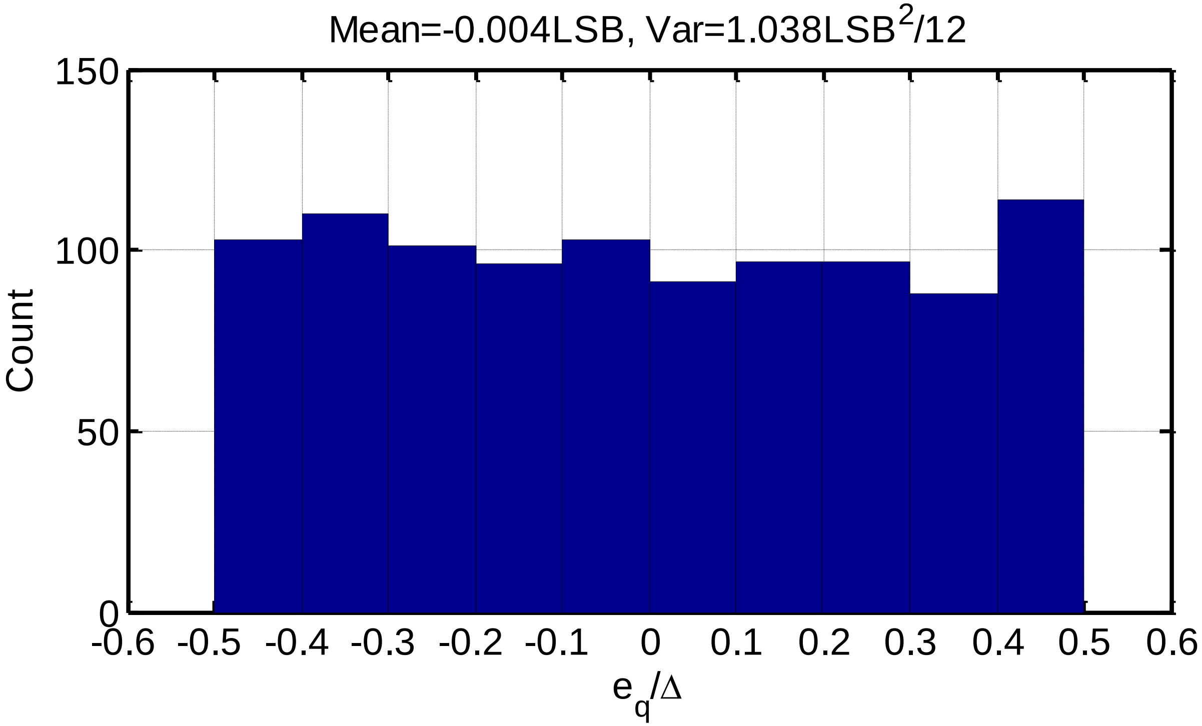


Mean $\overline{e_q} = 0$

Variance $$\boxed{\overline{e_q^2} = \int_{-\Delta/2}^{+\Delta/2} \frac{{e_q}^2}{\Delta} de_q = \frac{\Delta^2}{12}}$$

Reality Check (1)

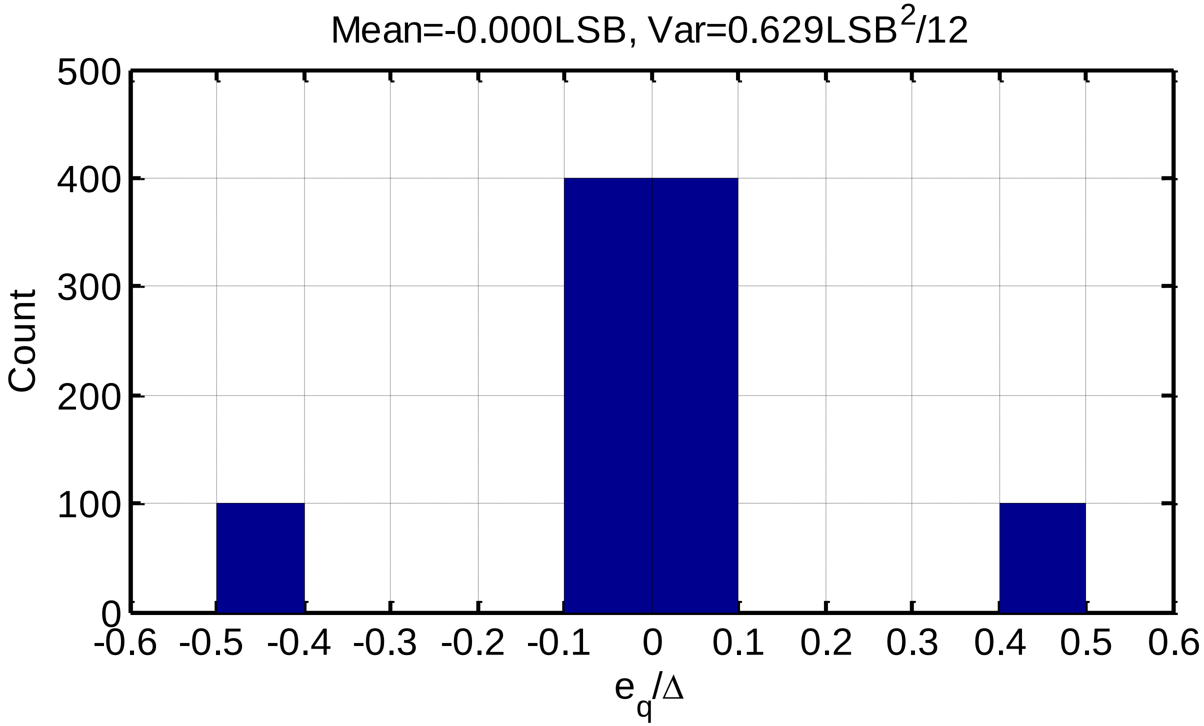
- Input sequence consists of 1000 samples drawn from Gaussian distribution, 4σ=FSR



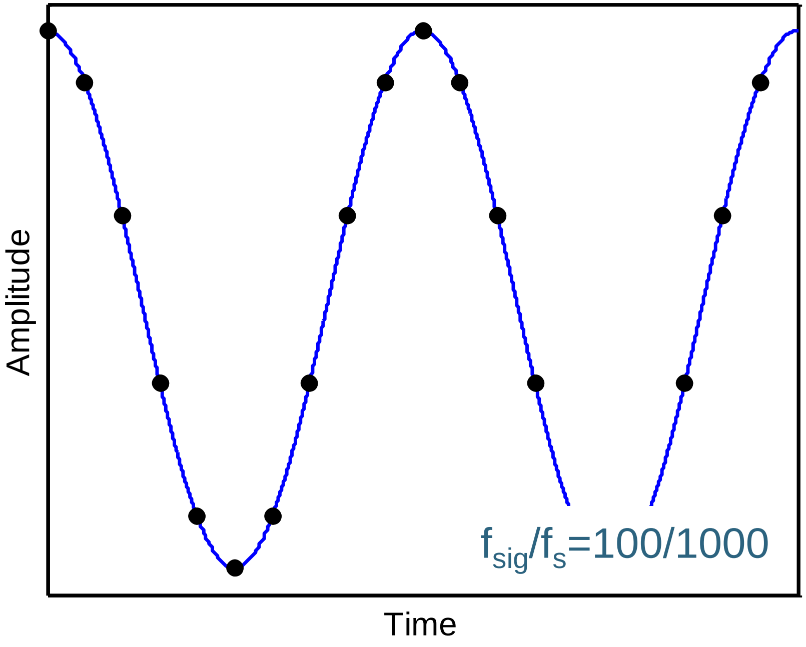
- Error power close to that of uniform approximation (3.8% larger)

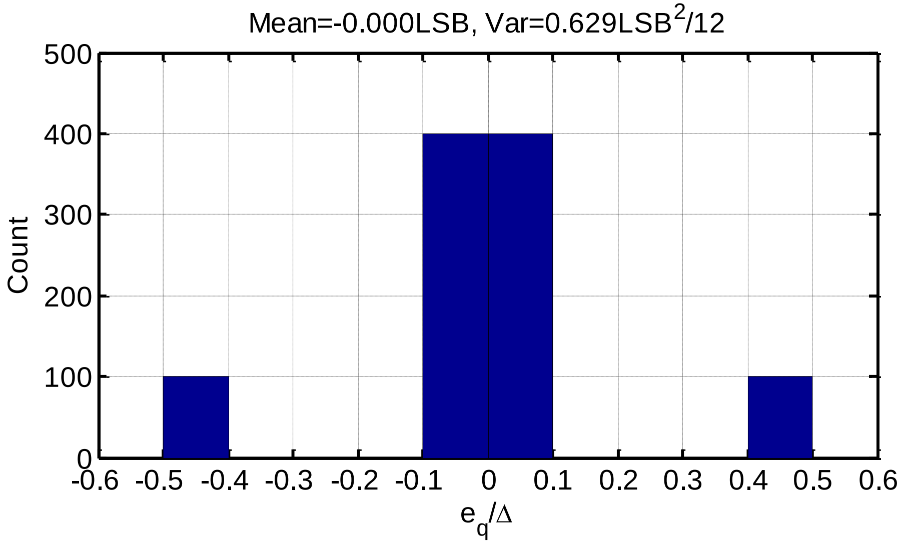
Reality Check (2)

- Another sine wave example, but now $f_{sig}/f_s=100/1000$
- What's going on here?



Analysis





- Sampled signal is repetitive and has only a few distinct values
  - This also means that the quantizer produces only a few distinct values of e_q that repeat over and over – not a uniform distribution
  - Hence the lumpy histogram

EE 240C
Analog-Digital Interface Integrated Circuits

Quantization Error: SNR and Spectrum

Signal-to-Quantization-Noise Ratio

- Assuming uniform distribution of e_q and a full-scale sinusoidal input

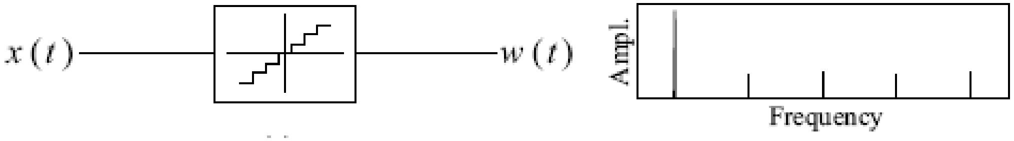
$$SQNR = \frac{P_{sig}}{P_{qnoise}} = \frac{\frac{1}{2}\left(\frac{2^B\Delta}{2}\right)^2}{\frac{\Delta^2}{12}} = 1.5 \times 2^{2B} = 6.02B + 1.76 \text{ dB}$$

B (Number of Bits)	SQNR
8	50 dB
12	74 dB
16	98 dB

- Questions:
  - What is the maximum SQNR of a 32 Bit integer?
  - Of a 64 Bit floating point number?
  - What limits the SQNR of practical ADCs?

Quantization Noise Spectrum (1)

- How is the quantization noise power distributed in frequency?
  - First think about applying a sine wave to a quantizer, without sampling (output is continuous time)



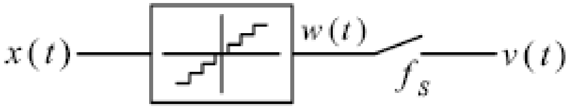
+ many more harmonics

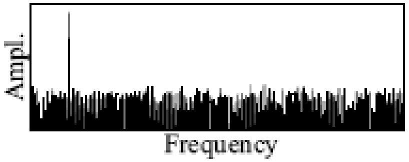
[Y. Tsividis, ICASSP 2004]

- Quantization results in an "infinite" number of harmonics

Quantization Noise Spectrum (2)

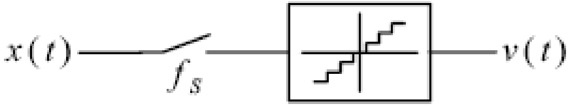
- Now sample the signal at the output
  - All harmonics (an "infinite" number of them) will alias into band from 0 to $f_s/2$
  - Quantization noise spectrum becomes "white"

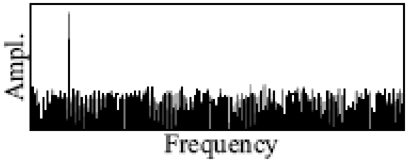




[Y. Tsividis, ICASSP 2004]

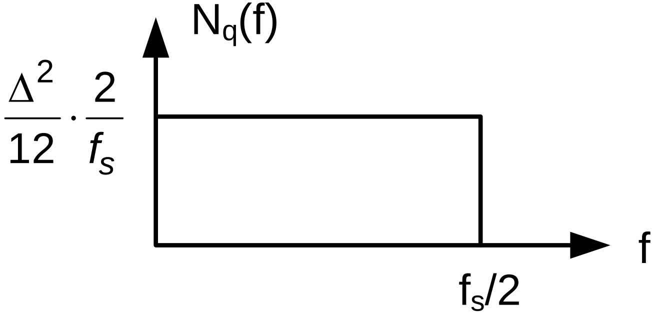
- Interchanging sampling and quantization won't change this situation





Quantization Noise Spectrum (3)

- Can show that the quantization noise power is indeed distributed (approximately) uniformly in frequency
  - Again, this is provided that the quantizer input is “busy” and error is "sufficiently random"



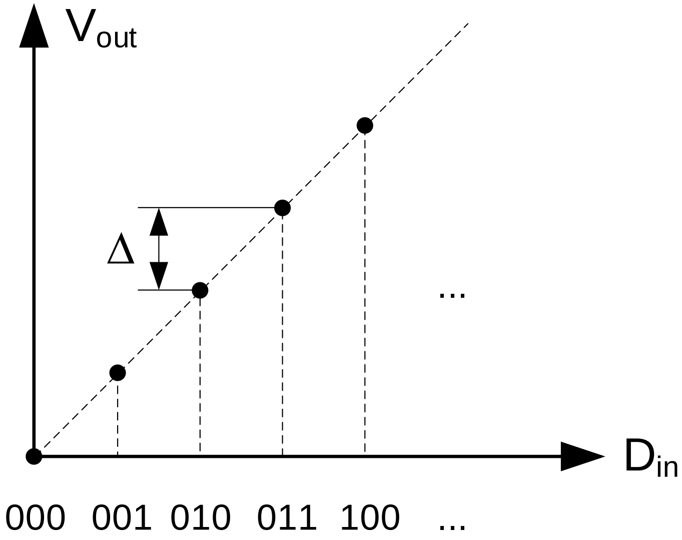
- References
  - W. R. Bennett, "Spectra of quantized signals," Bell Syst. Tech. J., pp. 446-72, July 1948.
  - B. Widrow, "A study of rough amplitude quantization by means of Nyquist sampling theory," IRE Trans. Circuit Theory, vol. CT-3, pp. 266-76, 1956.
  - A. Sripad and D. A. Snyder, "A necessary and sufficient condition for quantization errors to be uniform and white," IEEE Trans. Acoustics, Speech, and Signal Processing, pp. 442-448, Oct 1977.

Summary

- Conditions: input signal is "busy" (no DC, obviously) and ranges over many quantization thresholds
- Then, the quantizer error
  - Has a nearly white spectrum, 1-sided power density $\overline{S_q^2} = \frac{\Delta^2}{12} \frac{1}{\frac{f_s}{2}}$
  - Has variance $\overline{e_q^2} = \frac{\Delta^2}{12}$
- Question: what's the quantization error of a DAC?

Ideal DAC

- Essentially a digitally controlled voltage or current source
  - Example below is for unipolar DAC
- Ideal DAC does not introduce quantization error!



EE 240C
Analog-Digital Interface Integrated Circuits

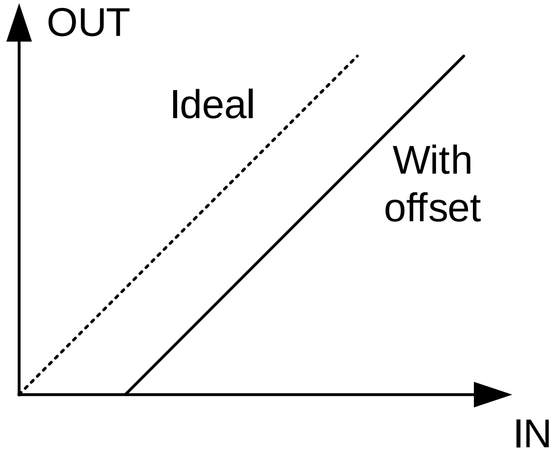
Static Quantization Errors (Nonidealities)

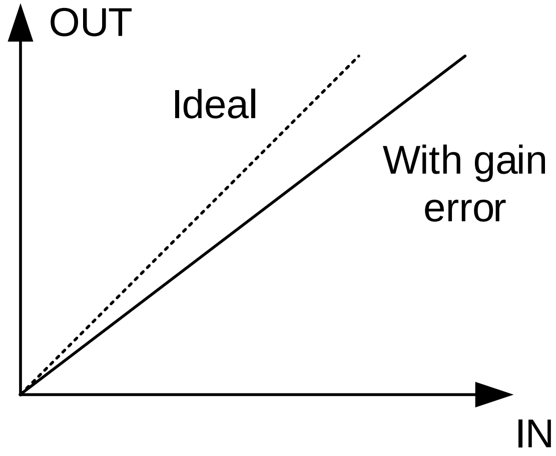
Static Nonidealities

- Static deviations of transfer characteristics from ideality
  - Offset
  - Gain error
  - Nonlinearity
- References
  - “An Introduction to Mixed-Signal IC Test & Measurement”, Oxford University Press
  - IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters (1241)
  - Analog Devices MT-010: The Importance of Data Converter Static Specifications
  - "Understanding Data Converters," Texas Instruments Application Report LAA013, 1995.
    - http://focus.ti.com/lit/an/slaa013/slaa013.pdf

Offset and Gain Error

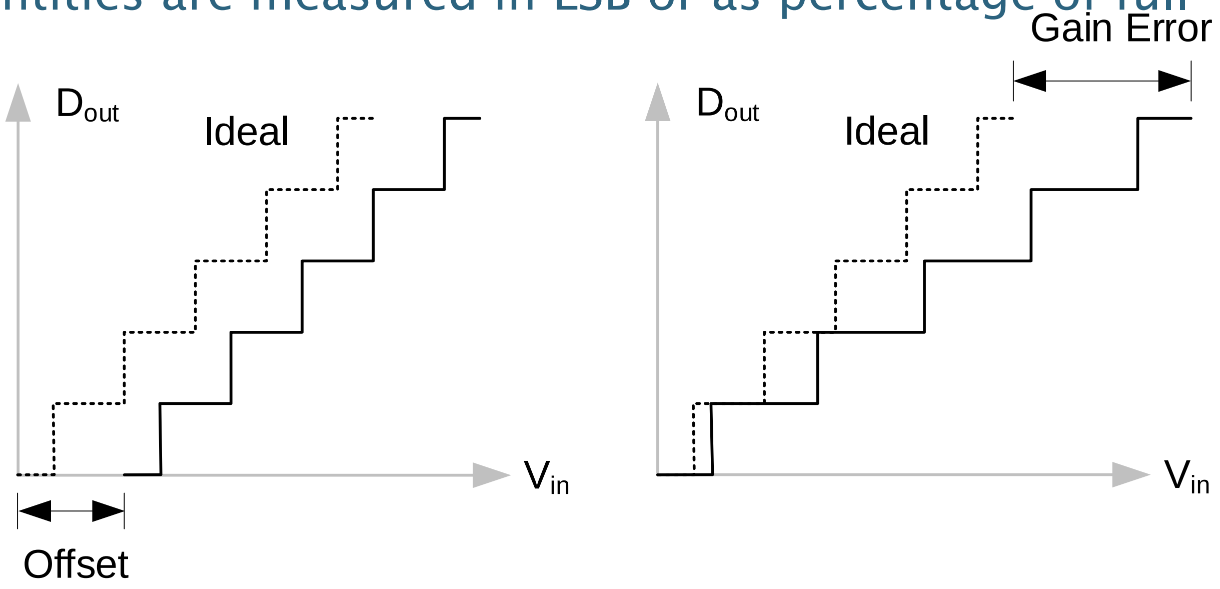
- Conceptually simple, but lots of (uninteresting) subtleties in how exactly these errors should be defined
  - Unipolar versus bipolar, endpoint versus midpoint specification
  - Definition in presence of nonlinearities
- General idea (neglecting staircase nature of transfer functions):





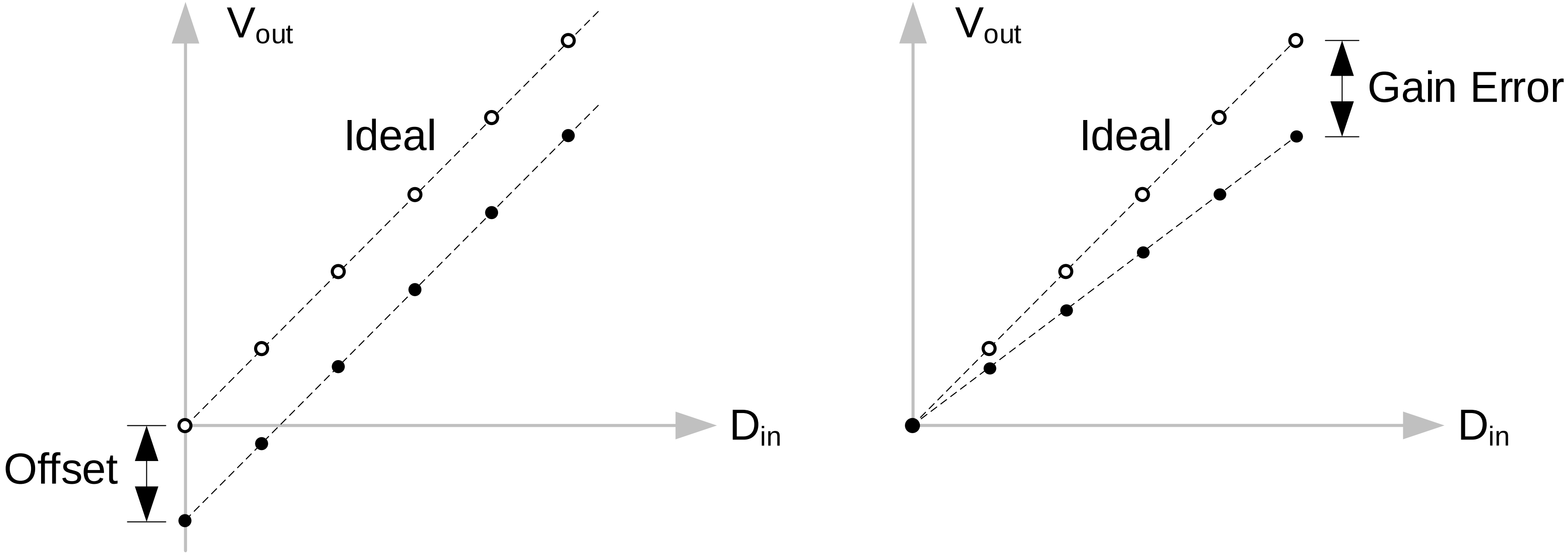
ADC Offset and Gain Error

- Definitions based on bottom and top endpoints of transfer characteristic
  - ½ LSB before first transition and ½ LSB after last transition
  - Offset is the deviation of bottom endpoint from its ideal location
  - Gain error is the deviation of top endpoint from its ideal location with offset removed
- Both quantities are measured in LSB or as percentage of full-scale range



DAC Offset and Gain Error

- Same idea, except that endpoints are directly defined by analog output values at minimum and maximum digital input
- Also note that errors are specified along the vertical axis



Comments on Offset and Gain Errors

- Definitions on the previous slides are the ones typically used in industry
  - IEEE Standard suggest somewhat more sophisticated definitions based on least square curve fitting
    - Technically more suitable metric when the transfer characteristics are significantly non-uniform or nonlinear
- Generally, it is non-trivial to build a converter with very good gain/offset specifications
  - Nevertheless, since gain and offset affect all codes uniformly, these errors tend to be easy to correct
    - E.g. using a digital pre- or post-processing operation
  - Also, many applications are insensitive to a certain level of gain and offset errors
    - E.g. audio signals, communication-type signals, ...
  - Gain and offset errors, and temperature coefficients, import for sensors
- More interesting aspect: linearity
  - DNL and INL

EE 240C
Analog-Digital Interface Integrated Circuits

Differential Nonlinearity (DNL)

Types of Nonlinearity

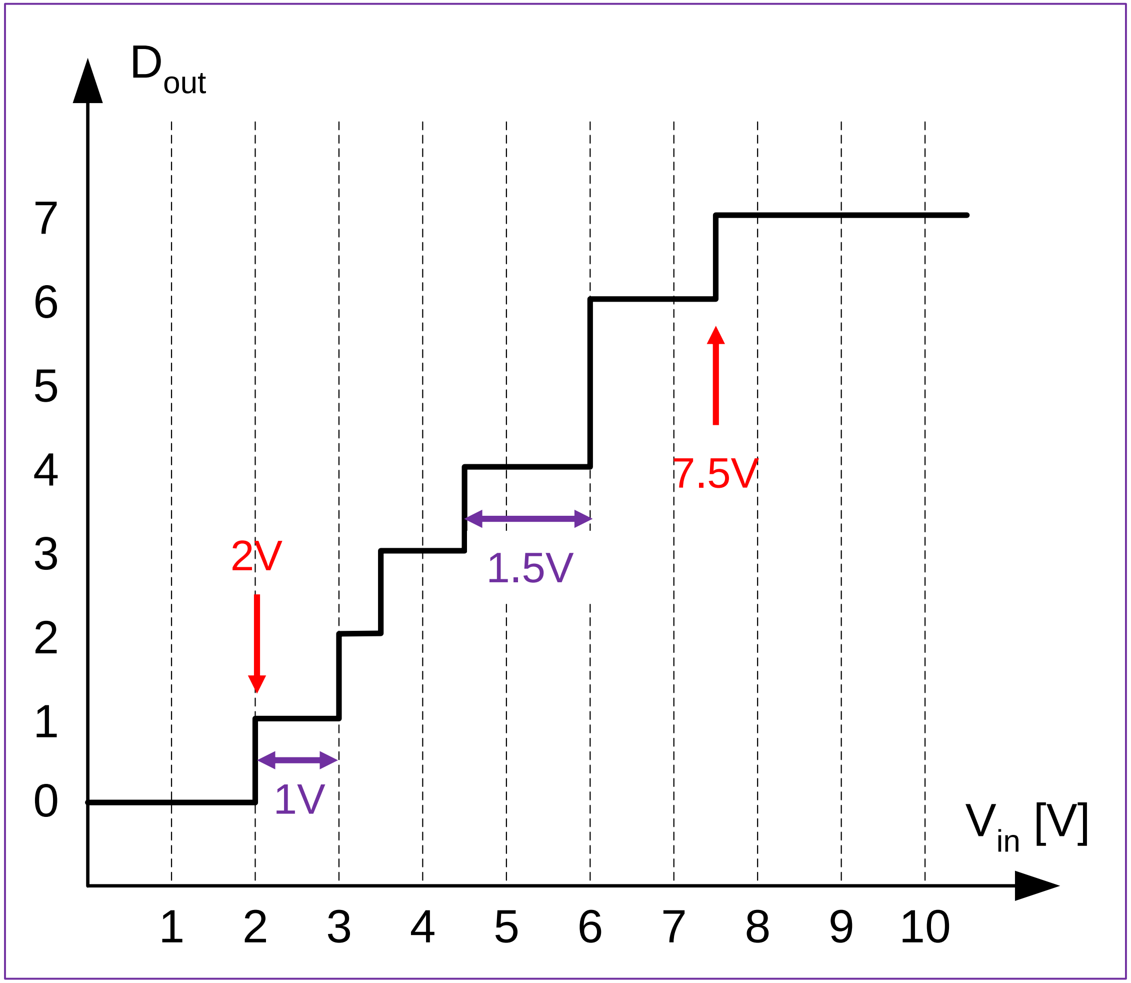
- Because of differences how these errors affect signals, quantizer nonlinearity is usually broken up into two components:
  1. Differential Nonlinearity (DNL)
     Deviation of each step from the ideal value, Δ
  2. Integral Nonlinearity (INL)
     Deviation of the I/O characteristic from a straight line

- The DNL and INL for each code is generally different, but statistics such as minimum/maximum are often sufficient to assess system level performance

Differential Nonlinearity (DNL)

- In an ideal world, all ADC codes would have equal width Δ; all DAC output increments would have same size
- DNL(k) is a vector that quantifies for each code k the deviation of this width from the "average" width Δ (step size)
- DNL(k) is a measure of uniformity, it does not depend on gain and offset errors
  - Scaling and shifting a transfer characteristic does not alter its uniformity and hence DNL(k)
- Let's look at an example

ADC DNL Example (1)



Code (k)	width [V]
0	undefined
1	1
2	0.5
3	1
4	1.5
5	0
6	1.5
7	undefined

ADC DNL Example (2)

- What is the average code width?
  - ADC with perfect uniformity would divide the range between first and last transition into 6 equal pieces
  - Hence calculate average code width (i.e. LSB size) as

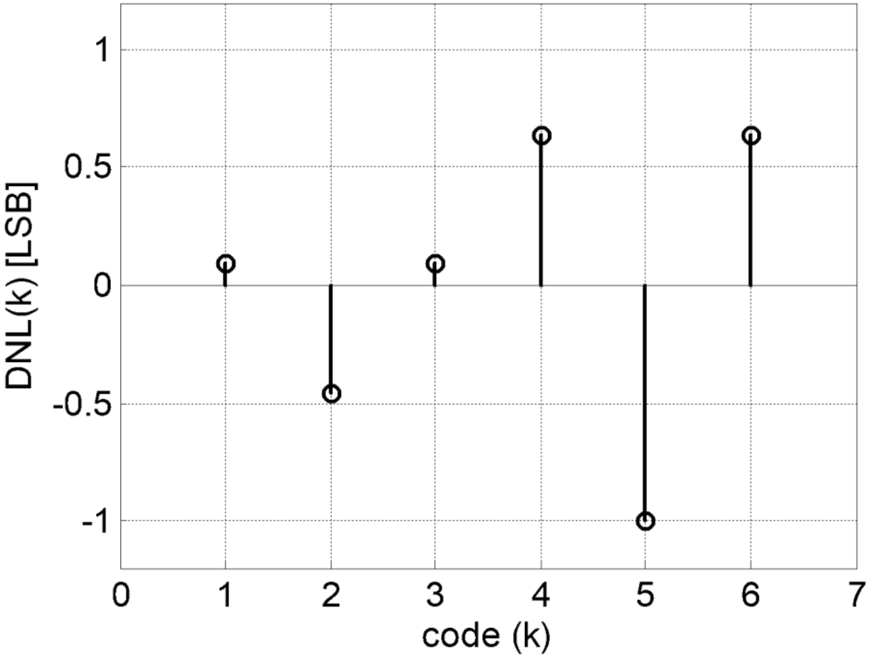
$$W_{\mathrm{avg}} = \frac{7.5V - 2V}{6} = 0.9167V$$

- Now calculate DNL(k) for each code k using

$$\mathrm{DNL}(k) = \frac{W(k) - W_{\mathrm{avg}}}{W_{\mathrm{avg}}}$$

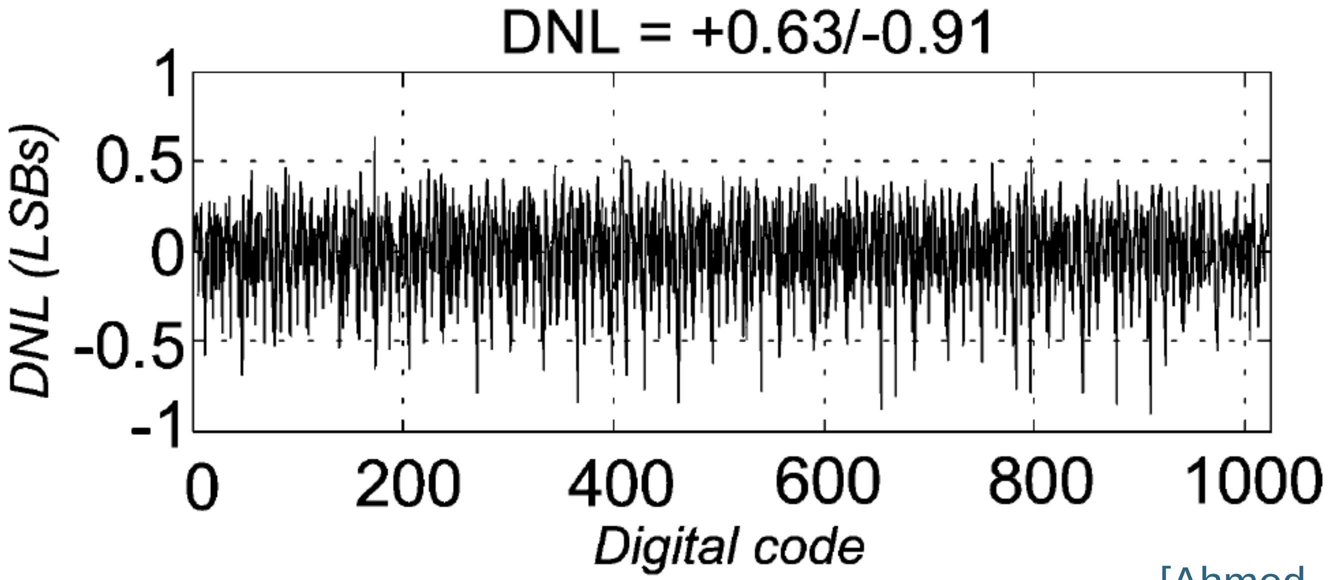
Result

Code (k)	DNL [LSB]
1	0.09
2	-0.45
3	0.09
4	0.64
5	-1.00
6	0.64



- Positive DNL: wide code, negative DNL: narrow code
- DNL = –1 LSB implies missing code
- Impossible to have DNL < –1 LSB for an ADC
  - But possible to have DNL > +1 LSB
- Sum over all DNL(k) is equal to zero

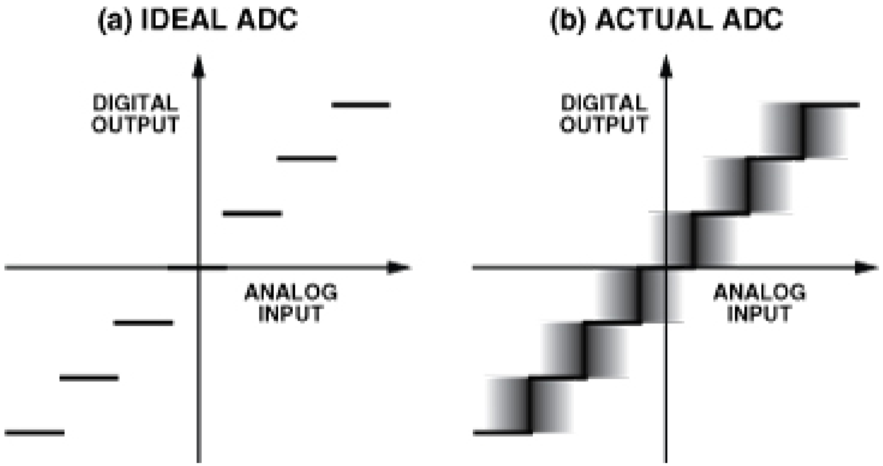
A Typical ADC DNL Plot



[Ahmed, JSSC 12/2005]

- People speak about DNL often only in terms of min/max number across all codes
  - E.g. DNL = +0.63/−0.91 LSB
- Any code with DNL < −0.9 LSB is essentially a missing code. Why?

Impact of Noise



[W. Kester, "ADC Input Noise: The Good, The Bad, and The Ugly. Is No Noise Good Noise?" Analogue Dialogue, Feb. 2006]

- In essentially all moderate to high-resolution ADCs, the transition levels carry noise that is somewhat comparable to the size of an LSB
  - Noise "smears out" DNL, can hide missing codes
- Especially for converters whose input referred (thermal) noise is larger than an LSB, DNL is a fairly useless metric

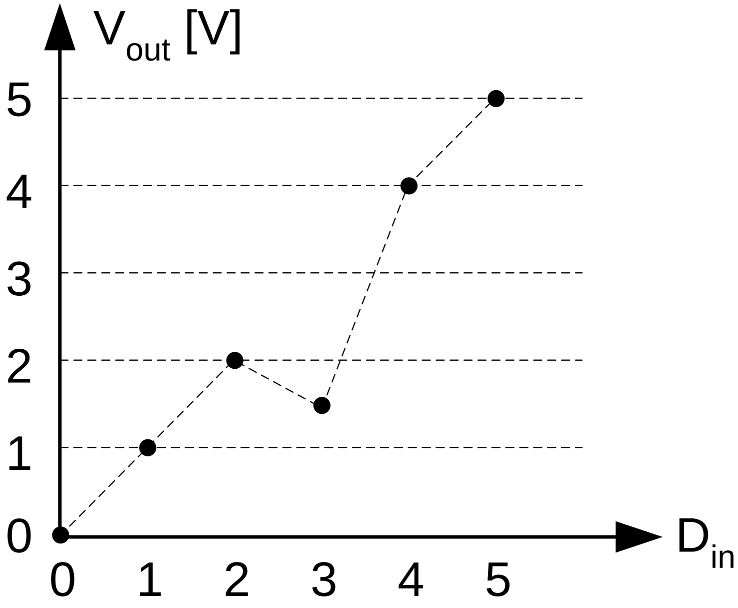
DAC DNL

- Same idea applies
  - Find output increments for each digital code
  - Find increment that divides range into equal steps
  - Calculate DNL for each code k using

$$\mathrm{DNL}(k) = \frac{\mathrm{Step}(k) - \mathrm{Step}_{\mathrm{avg}}}{\mathrm{Step}_{\mathrm{avg}}}$$

- DAC DNL can be less than -1 LSB

Non-Monotonic DAC

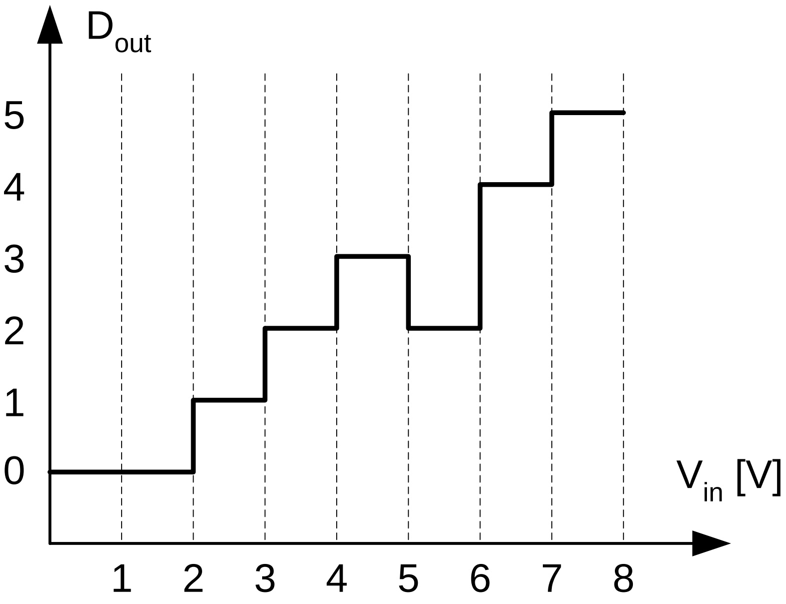


$$\mathrm{DNL}(3) = \frac{\mathrm{Step}(3) - \mathrm{Step}_{\mathrm{avg}}}{\mathrm{Step}_{\mathrm{avg}}}$$

$$= \frac{-0.5V - 1V}{1V}$$

- In a DAC, DNL < -1LSB implies non-monotinicity
- How about a non-monotonic ADC?

Non-Monotonic ADC



- Code 2 has two transition levels ⇒ W(2) is ambiguous
  - DNL is ill–defined!
- Not a very big issue, because a non–monotonic ADC is usually not what we design for in practice…

Question

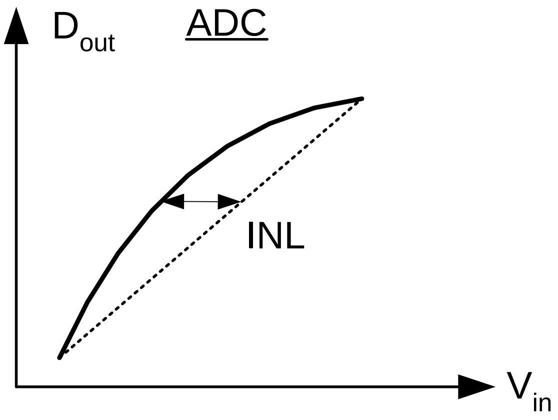
- Let's say I built a 16-Bit converter with a ±1 LSB DNL
- Marketing can get orders only for converters with ≤ ±0.5 LSB DNL
- What's the easiest "fix" to get ±0.5 LSB DNL?
  (short of hiring a new marketing team).

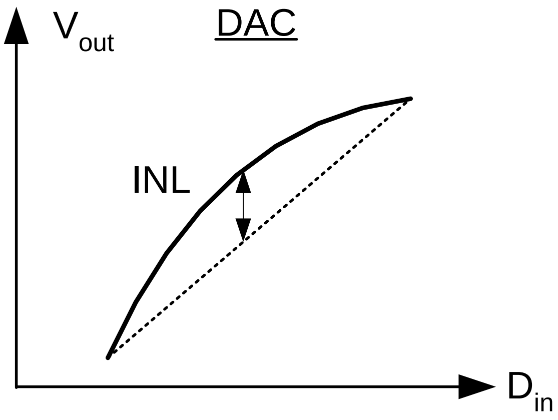
EE 240C
Analog-Digital Interface Integrated Circuits

Integral Nonlinearity (INL)

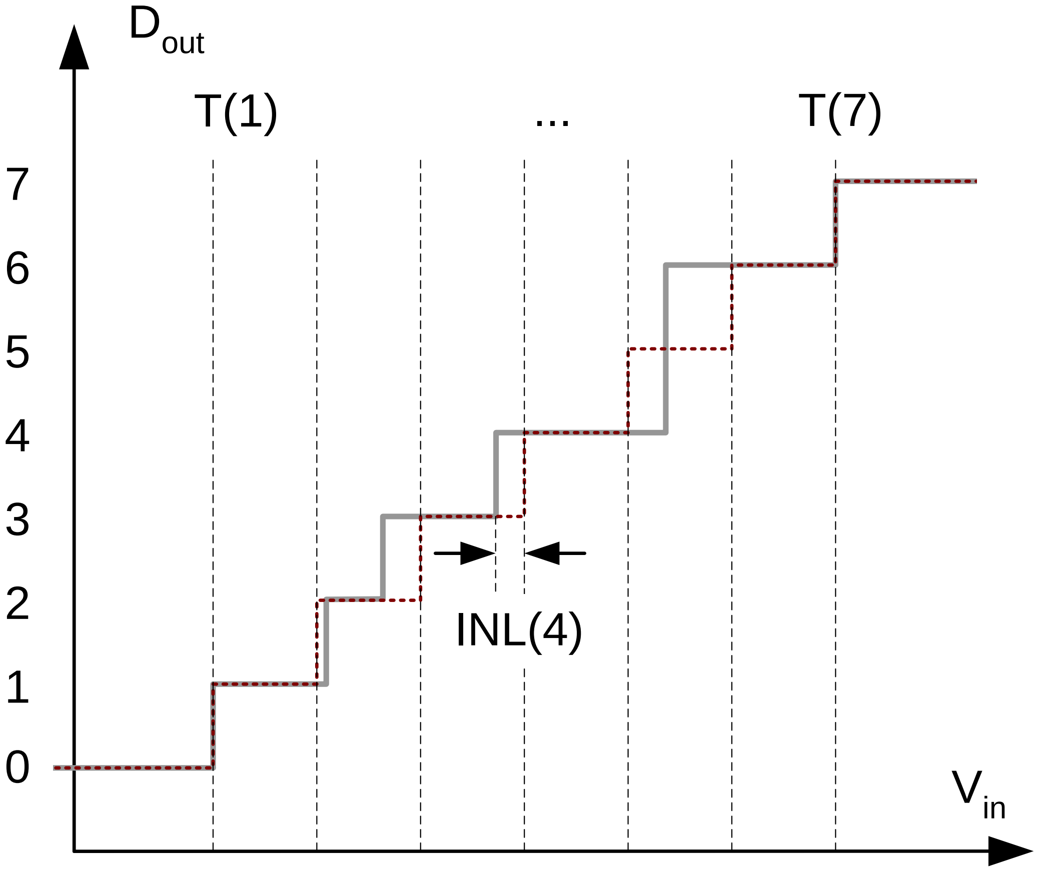
Integral Nonlinearity (INL)

- General idea
  - For each "relevant point" of the transfer characteristic, quantify distance from a straight line drawn through the endpoints
    - An alternative definition uses a least square fit line as a reference
  - Just as with DNL, the INL of a converter is by definition independent of gain and offset errors





ADC INL Example (1)



- "Straight line" reference is uniform staircase between first and last transition

  INL for each code is

$$\mathrm{INL}(k) = \frac{T(k) - T_{\mathrm{uniform}}(k)}{W_{\mathrm{avg}}}$$

- INL(1) = 0 and INL(7) = 0
- INL(0) is undefined

ADC INL Example (2)

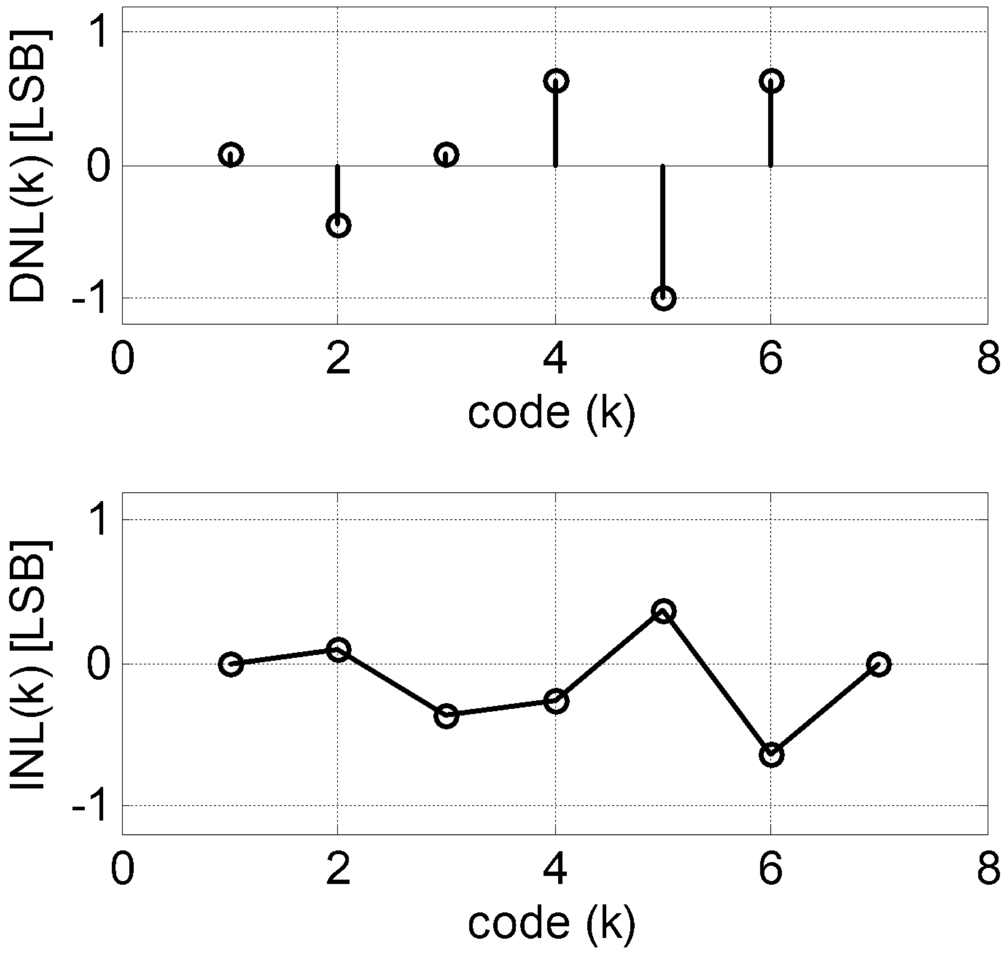
- INL versus DNL

$$\mathrm{INL}(k) = \sum_{i=1}^{k-1} \mathrm{DNL}(i)$$

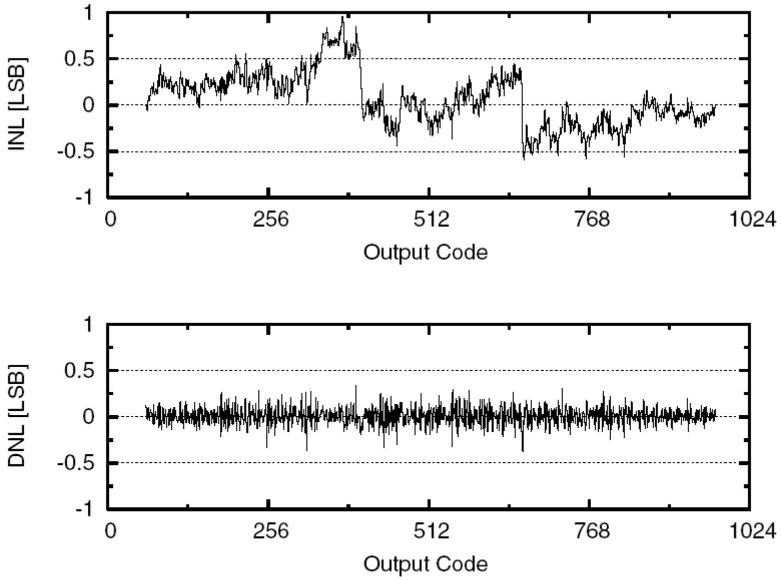
- Once we computed DNL, we can easily find INL using a cumulative sum operation on the DNL vector
- Using values from the DNL example, we find

Code (k)	DNL [LSB]	INL [LSB]
1	0.09	0
2	-0.45	0.09
3	0.09	-0.36
4	0.64	-0.27
5	-1.00	0.36
6	0.64	-0.64
7	undefined	0

Result



A Typical ADC DNL/INL Plot



[Ishii, Custom Integrated Circuits Conference, 2005]

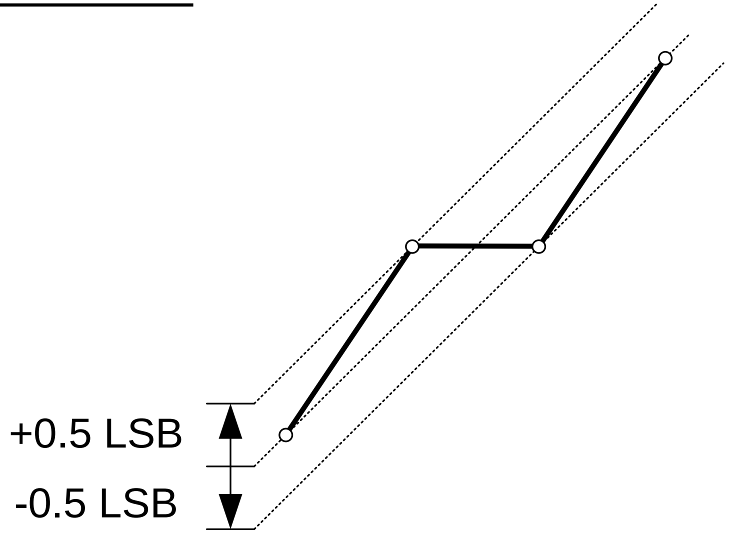
- DNL/INL signature often reveals architectural details
  - E.g. major transitions
  - We'll see more examples in the context of DACs
- Since INL is a cumulative measure, it turns out to be less sensitive than DNL to thermal noise "smearing"

DAC INL

- Same idea applies
  - Find ideal output values that lie on a straight line between endpoints
  - Calculate INL for each code k using

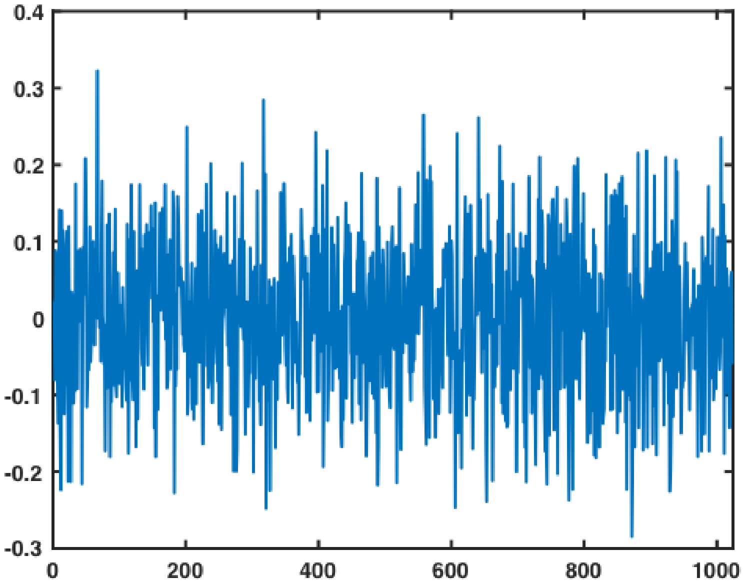
$$\mathrm{INL}(k) = \frac{V_{\mathrm{out}}(k) - V_{\mathrm{out}_{\mathrm{uniform}}}(k)}{\mathrm{Step}_{\mathrm{avg}}}$$

- Property of DAC INL
  - If for all codes |INL| < 0.5 LSB, it follows that all |DNL| < 1 LSB
  - A sufficient (but not necessary) condition for monotonicity

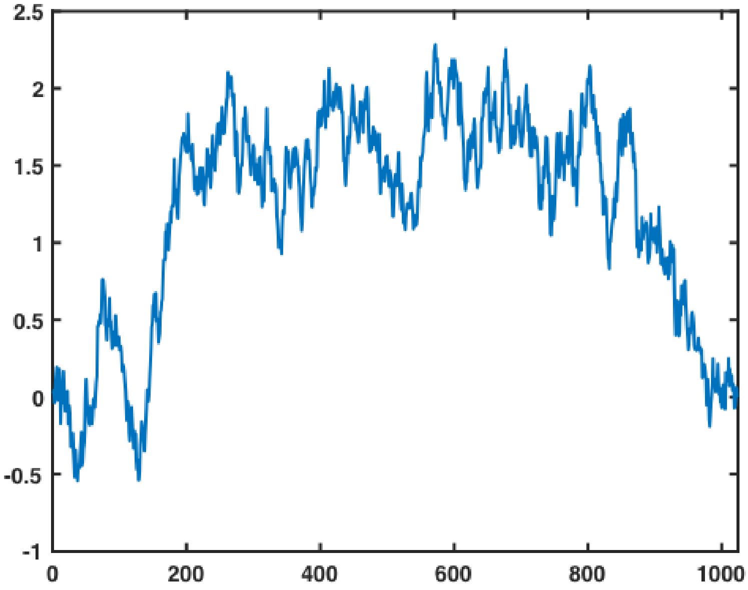


Example: Unit Element DAC

DNL

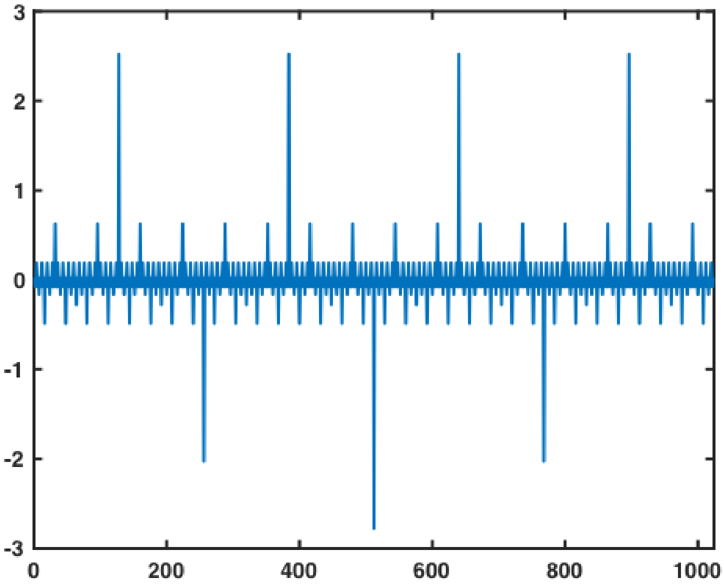


INL

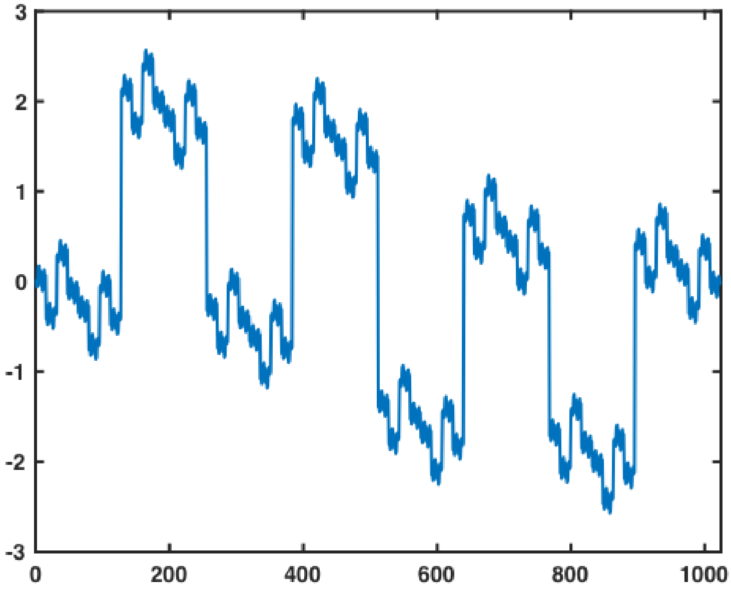


Example: Binary weighted DAC

DNL



INL



Same unit element weights !

Question

- INL and DNL reveal a lot about a converter
- But they miss a lot, too
  - What?

EE 240C
Analog-Digital Interface Integrated Circuits

Histogram Testing

Post-Processing

- DAC
  - "Trivial", apply codes and use "a good voltmeter" to measure outputs
- ADC
  - Need to find "decision levels", i.e. input voltages at all code boundaries
  - One way: Adjust voltage source to find exact code transitions
    - "code boundary servo"
  - More elegant: Histogram testing

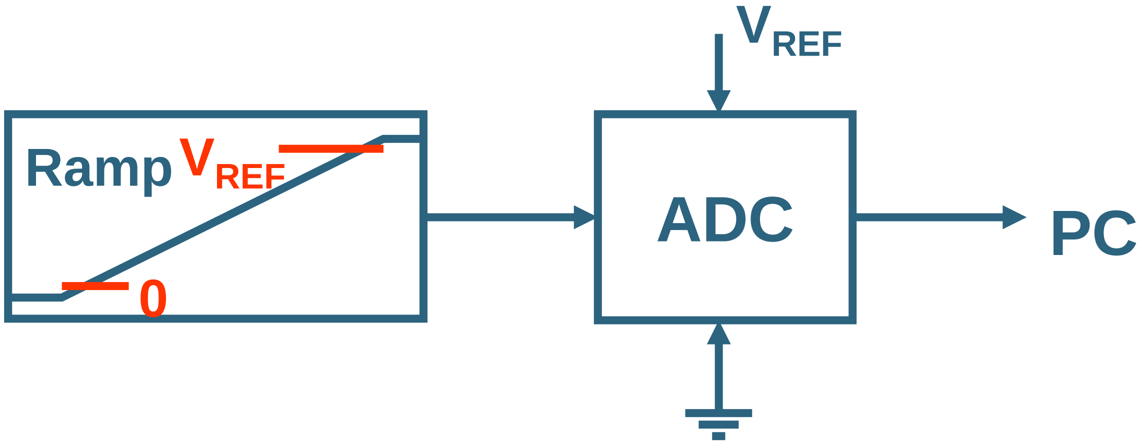
$$\mathrm{DNL}(k) = \frac{\mathrm{Step}(k) - \mathrm{Step}_{avg}}{\mathrm{Step}_{avg}}$$

$$\mathrm{Step}(k) = T(k+1) - T(k)$$

$$\mathrm{INL}(k) = \frac{T(k) - T_{uniform}(k)}{W_{avg}}$$

$$\mathrm{INL}(k) = \sum_{i=1}^{k-1} \mathrm{DNL}(i)$$

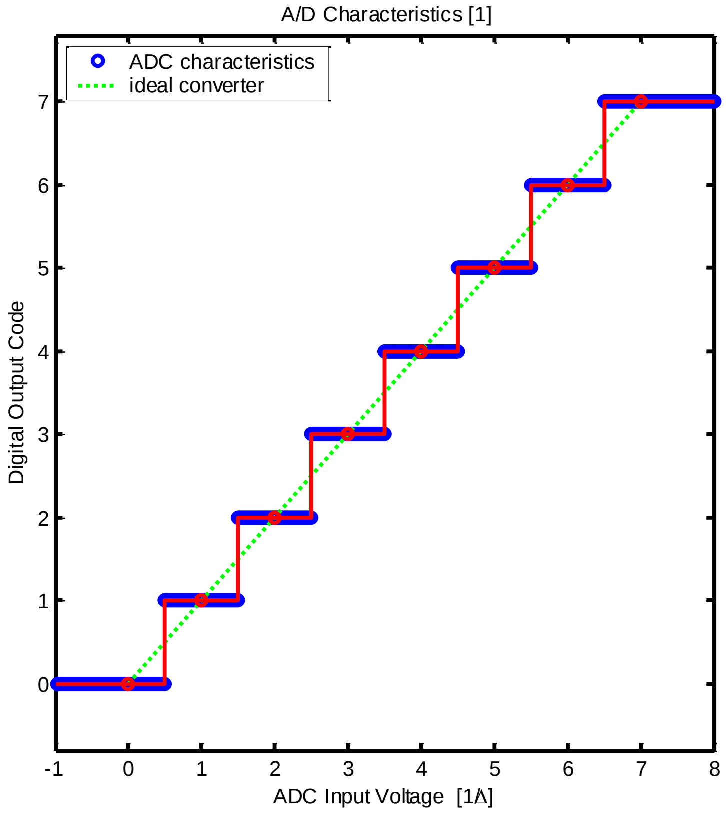
Basic Histogram Test Setup

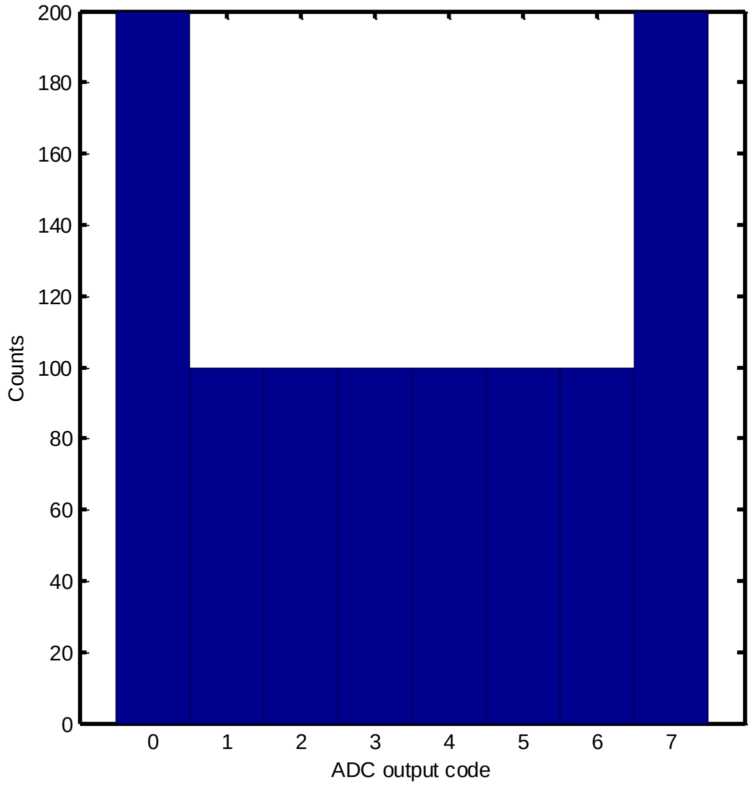


- DNL follows from total number of occurrences of each code
- Ramp speed is adjusted to provide e.g. an average of 100 outputs of each ADC code (for 1/100 LSB resolution)
- Ramps can be quite slow for high resolution ADCs

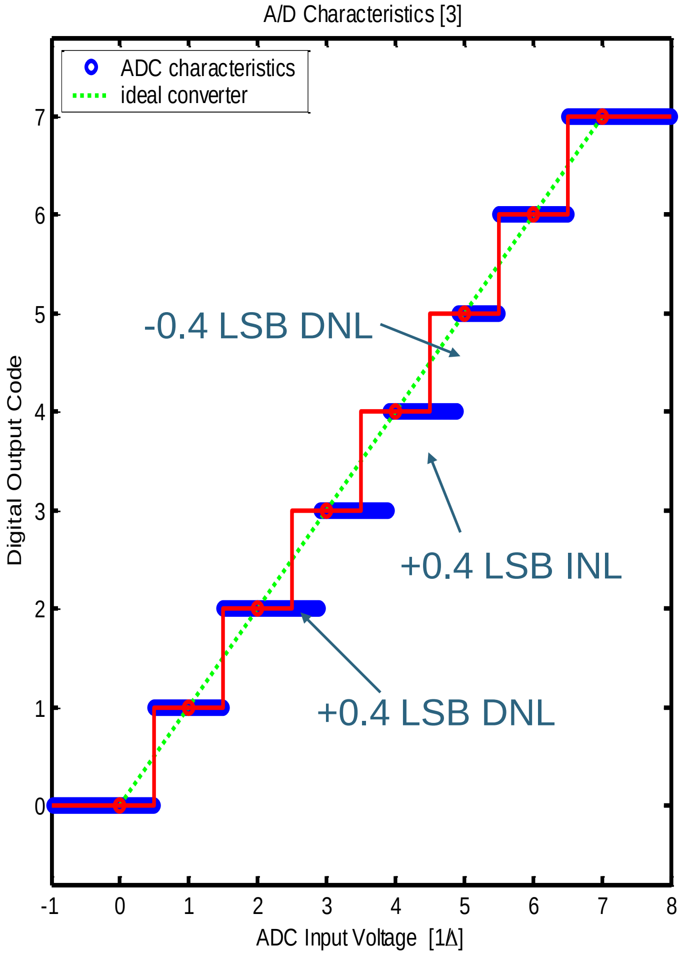
$$\frac{\text{(65,536 codes)(100 conversions/code)}}{\text{100,000 conversions/sec}} = \text{65.6 sec}$$

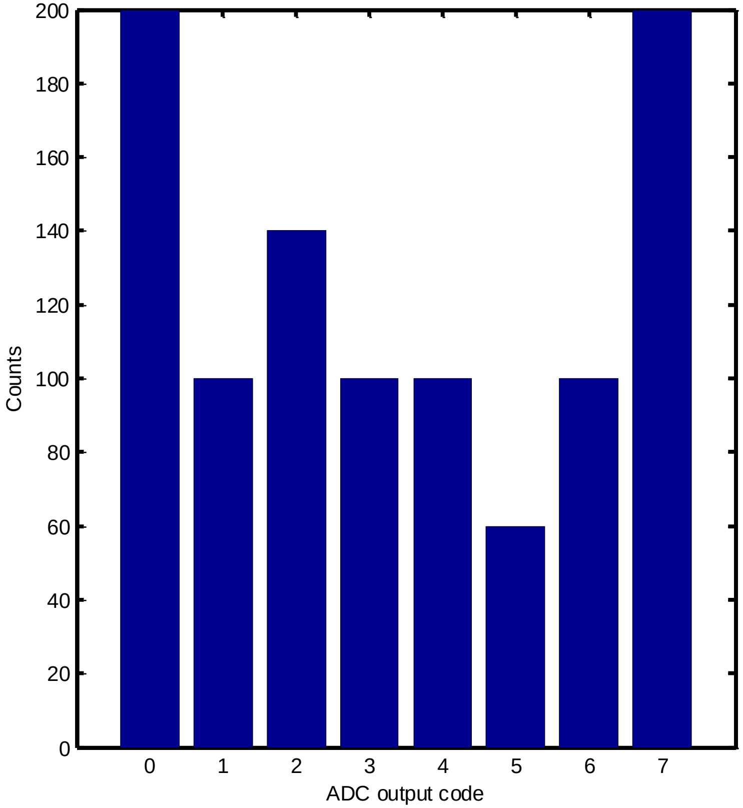
Histogram of Ideal 3 Bit ADC





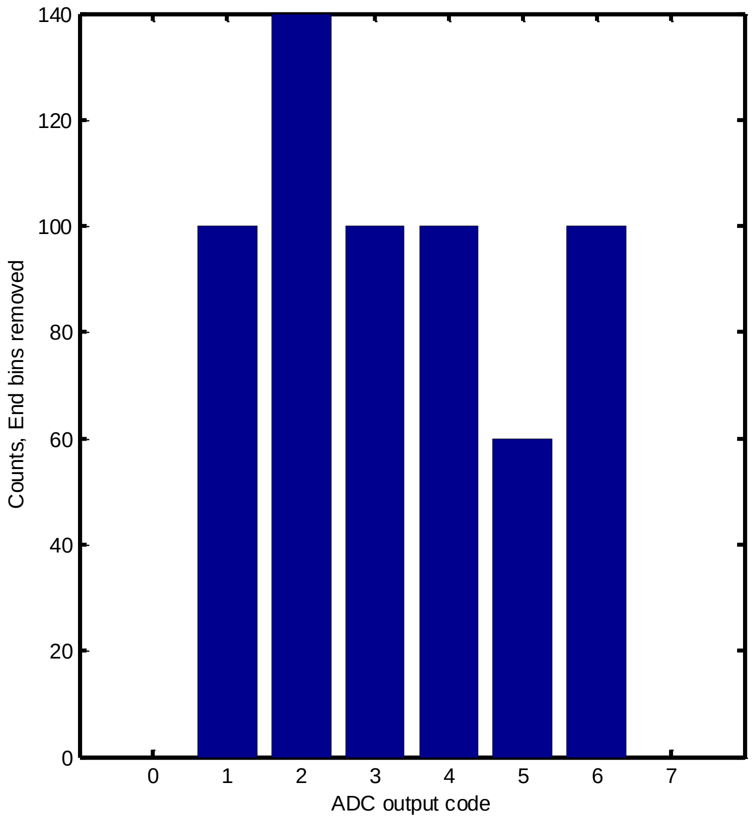
Histogram of Sample 3 Bit ADC





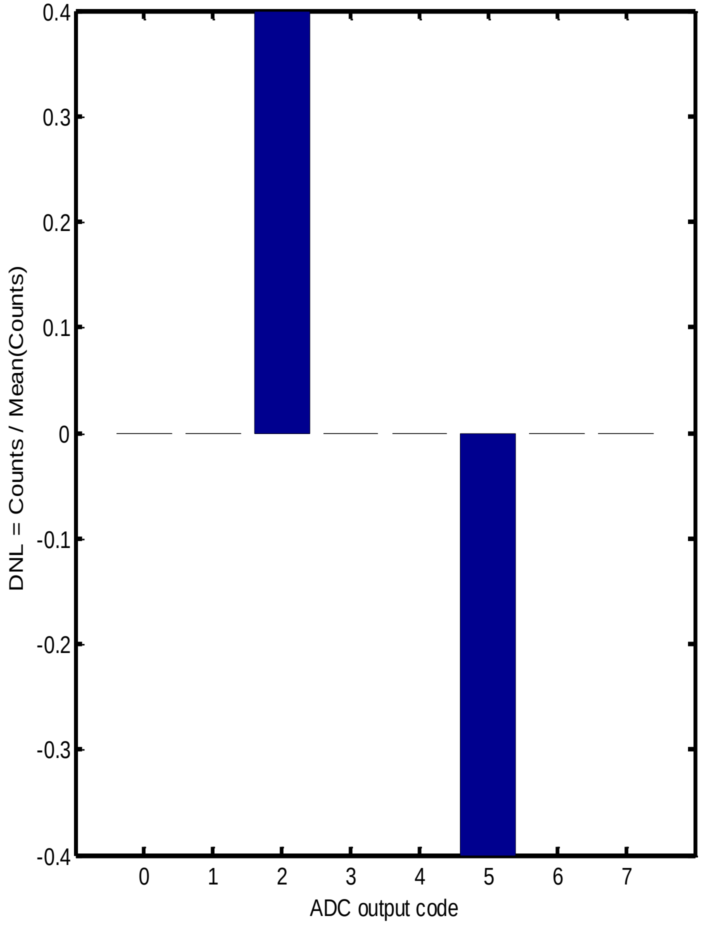
DNL from Histogram (1)

- Step 1
  - Remove “over-range bins” (0 and 7)



DNL from Histogram (2)

- Step 2
  - Divide by average count
- Step 3
  - Subtract 1
  - Ideal bins have exactly the average count, which corresponds to 1 after normalization
- Result is DNL

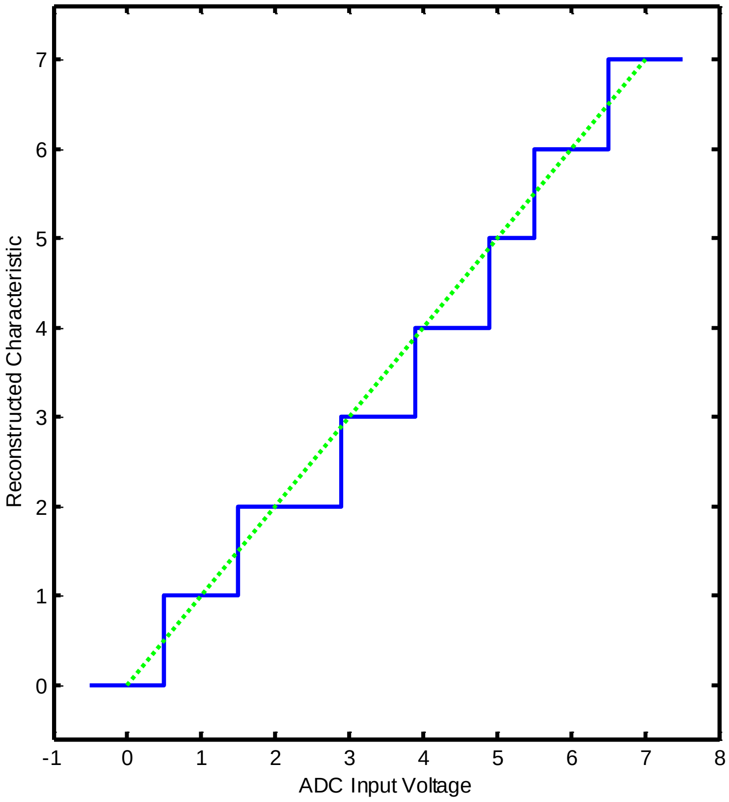


INL from Histogram

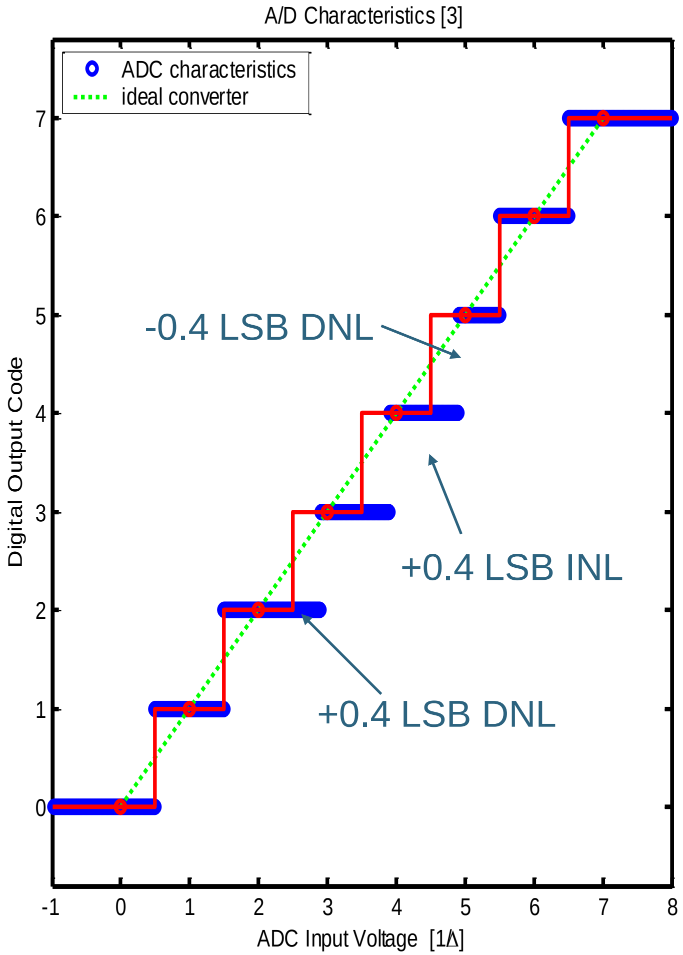
- INL is simply running sum of DNL
- The DNL information can also be used directly to construct the converter transfer function
  - Simply add up all bin-widths to find transition levels

$$\mathrm{INL}(k) = \sum_{i=1}^{k-1} \mathrm{DNL}(i)$$

$$\mathrm{INL}(k) = \frac{T(k) - T_{\mathrm{uniform}}(k)}{W_{\mathrm{avg}}}$$



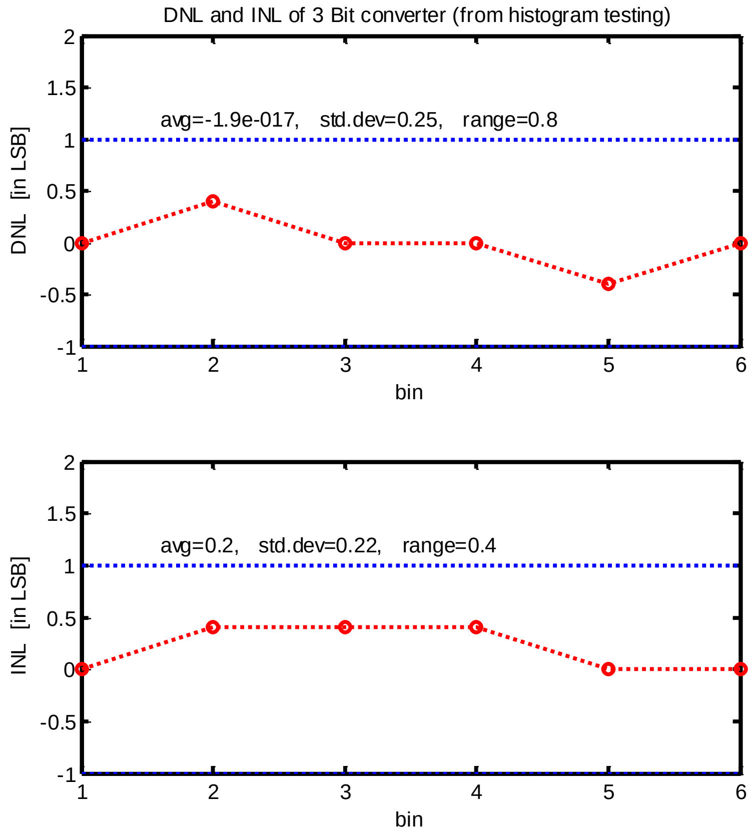
DNL and INL of Sample ADC



-0.4 LSB DNL

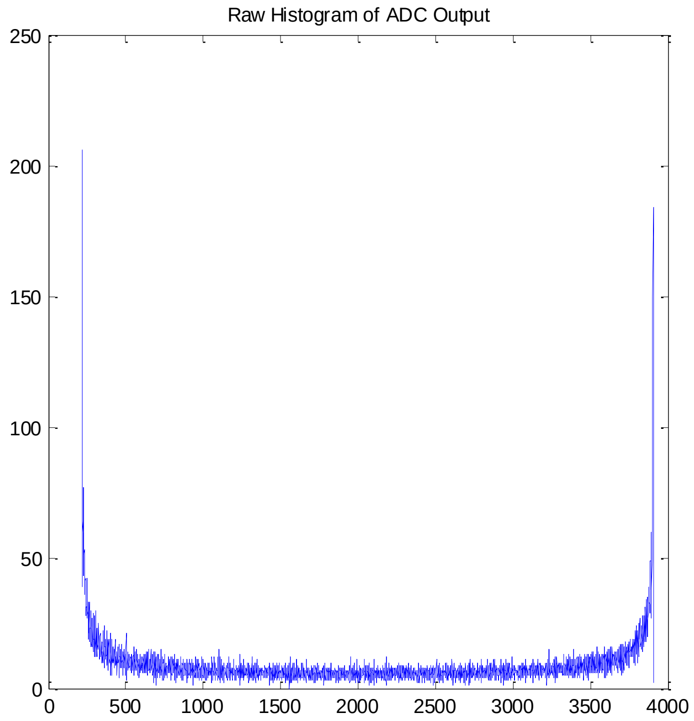
+0.4 LSB INL

+0.4 LSB DNL

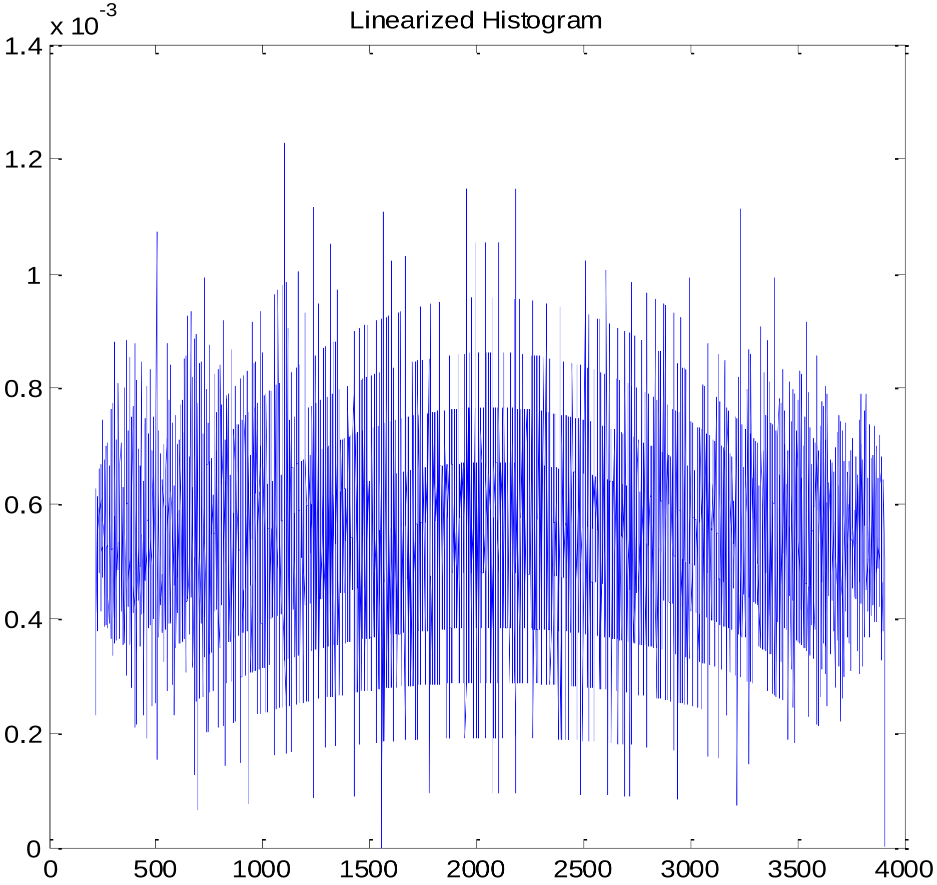


Sinusoidal Inputs

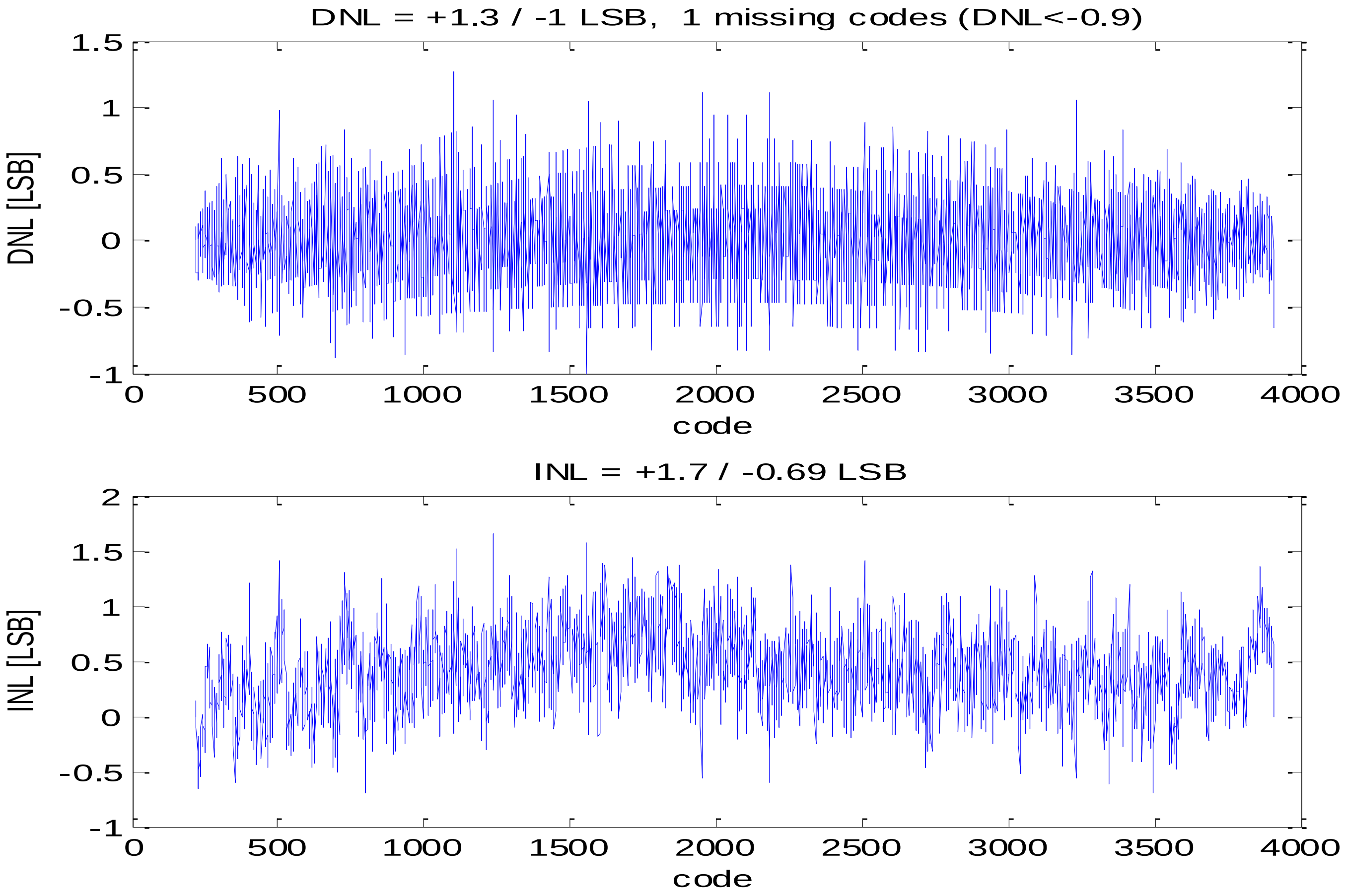
- Precise ramps are hard to generate
- Solution
  - Use sinusoidal test signal
- Problem
  - Ideal histogram is not flat but has "bath-tub shape"



After Correction for Sinusoidal pdf



Resulting DNL and INL



Correction for Sinusoidal pdf

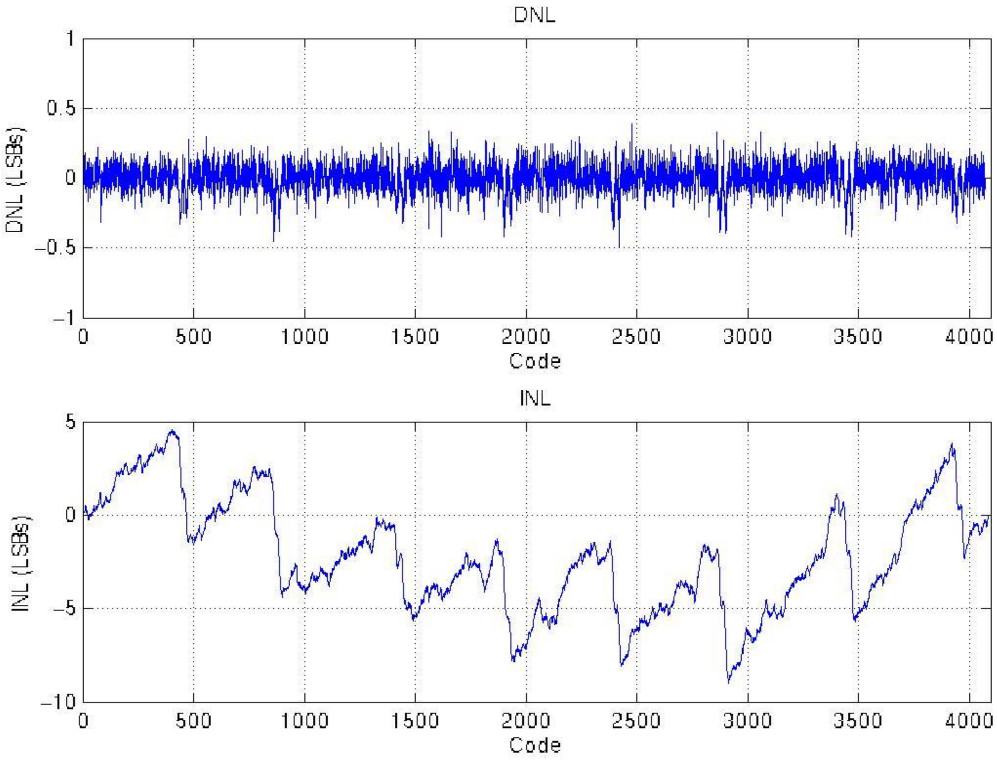
- References
  - M. V. Bossche, J. Schoukens, and J. Renneboog, "Dynamic Testing and Diagnostics of A/D Converters," IEEE TCAS, Aug. 1986.
  - IEEE Standard 1241
- Note: it is not necessary to know the exact amplitude, offset, and frequency of the test sine wave

Limitations of Histogram Testing

- The histogram test (as any ADC test, of course) characterizes one particular converter
  - Must test many devices to get valid statistics
- Histogram testing assumes monotonicity
  - E.g. “code flips” will not be detected.
- Dynamic sparkle codes produce only minor DNL/INL errors
  - E.g. 123, 123, …, 123, 0, 124, 124, …
  - Must look directly at ADC output to detect
- Noise not detected and can actually “improve” DNL
  - E.g. 9, 9, 9, 10, 9, 9, 9, 10, 9, 10, 10, 10, …
- Reference
  - B. Ginetti and P. Jespers, “Reliability of Code Density Test for High Resolution ADCs,” Electron. Letters, pp. 2231–2233, Nov. 1991.

Hiding Problems in the Noise

- INL looks a lot like there are 5 missing codes
- DNL "smeared out" by noise!
- Always look at both DNL/INL
- INL usually does not lie …



[Source: David Robertson, Analog Devices]