EE 505

Lecture 3

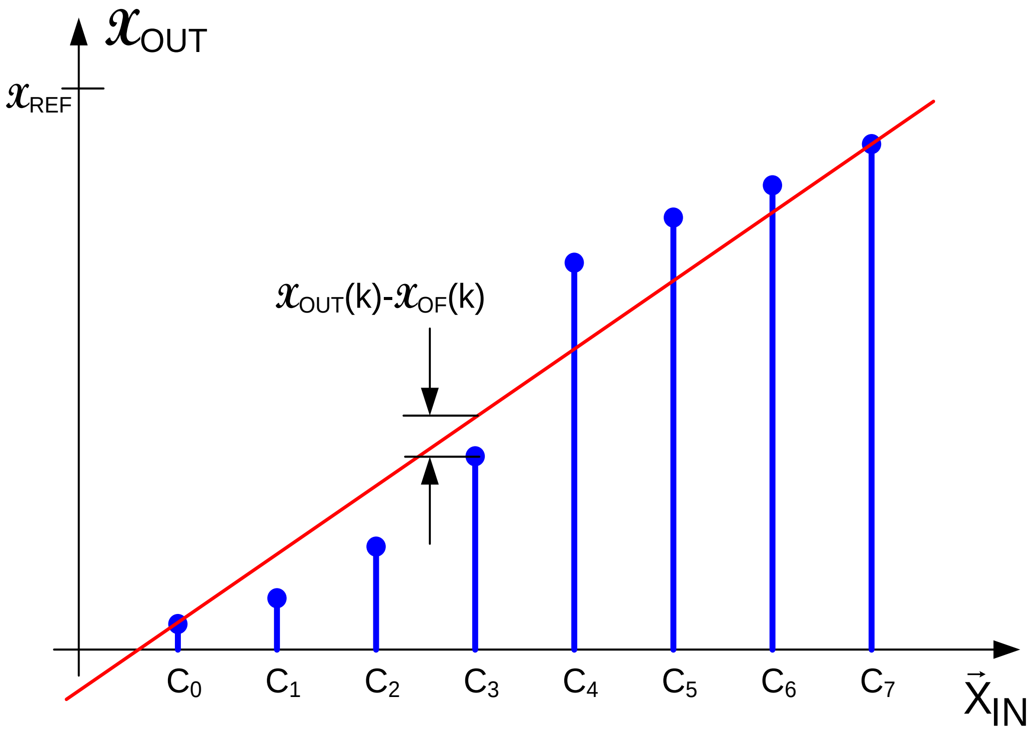
Data Converter Operation and Characterization

-- Linearity Metrics

REVIEW

Integral Nonlinearity (DAC)

Nonideal DAC



$$INL_k = \mathcal{X}_{OUT}(k) - \mathcal{X}_{OF}(k)$$

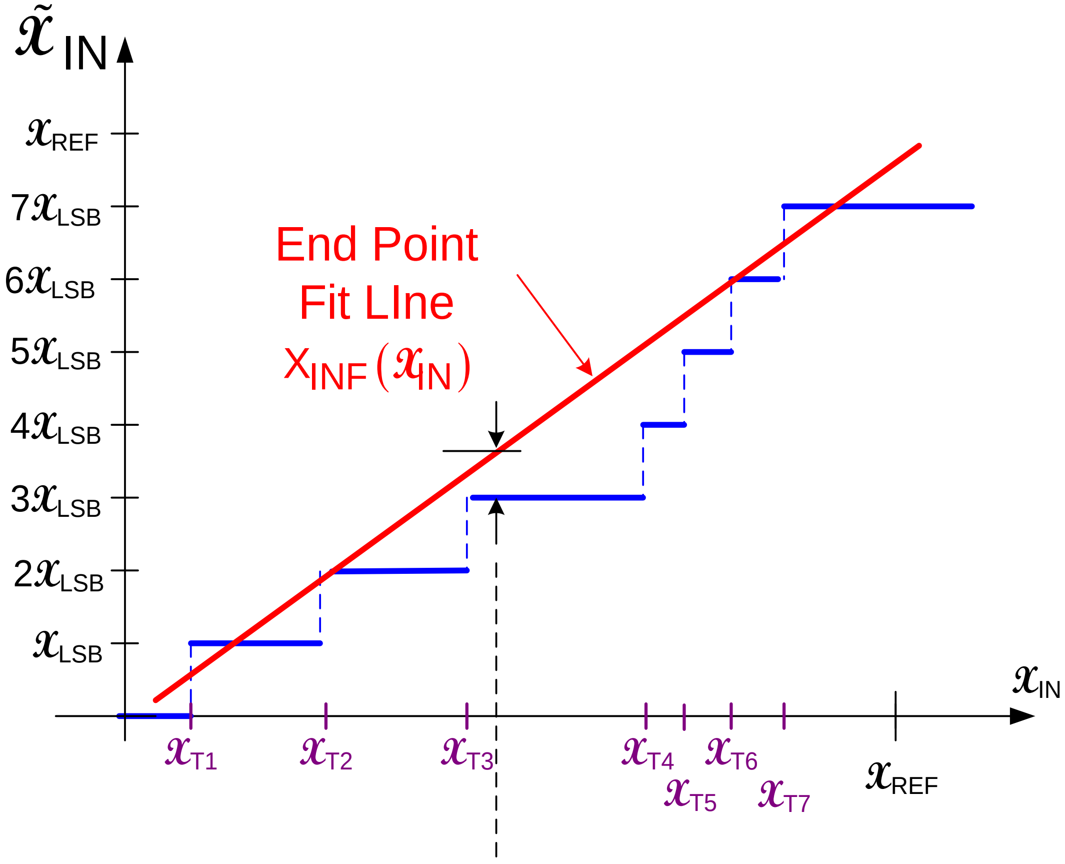
$$INL = \max_{0 \le k \le N-1} \left\{ \left| INL_k \right| \right\}$$

REVIEW

Integral Nonlinearity (ADC)

Nonideal ADC

Continuous-input based INL definition



$$INL(x_{IN}) = \tilde{x}_{IN}(x_{IN}) - X_{INF}(x_{IN})$$

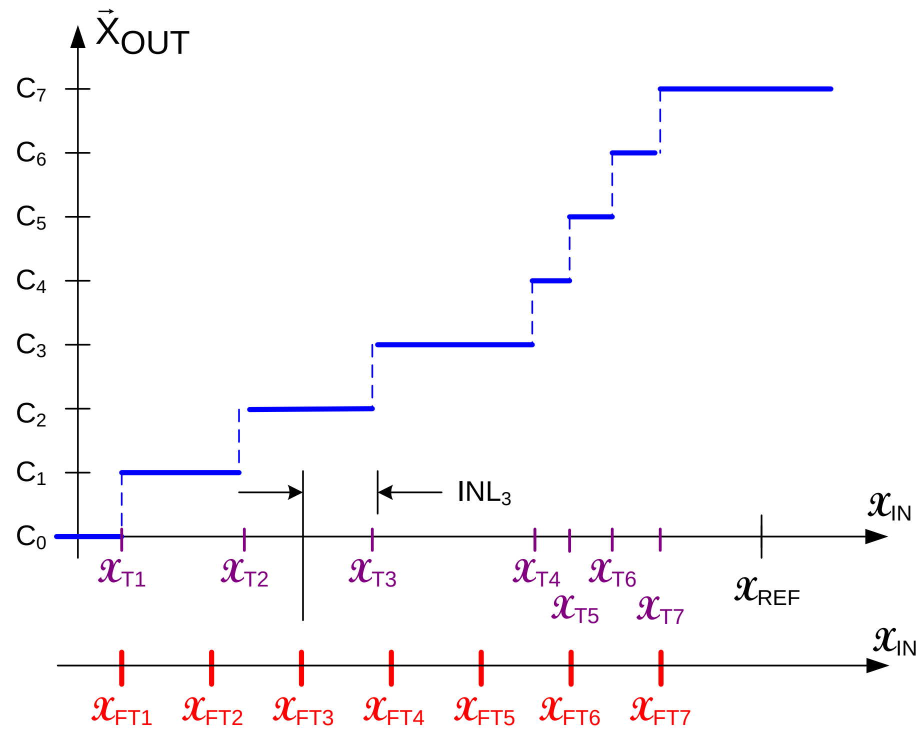
$$INL = \max_{0 \le x_{IN} \le x_{REF}} \left\{ \left| INL(x_{IN}) \right| \right\}$$

REVIEW

Integral Nonlinearity (ADC)

Nonideal ADC

Break-point INL definition



Place N-3 uniformly spaced points between X_{T1} and $X_{T(N-1)}$ designated x_{FTk}

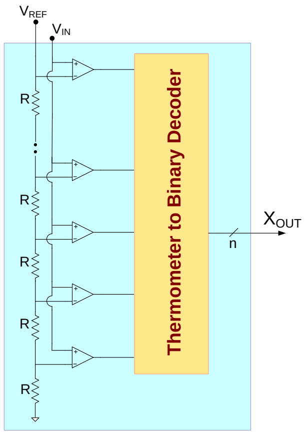
$$INL_k = x_{Tk} - x_{FTl} \qquad 1 \le k \le N-2$$

$$INL = \max_{2 \le k \le N-2} \left\{ \left| INL_k \right| \right\}$$

REVIEW

How important is statistical analysis?

Example: 7-bit FLASH ADC with R-string DAC



Case 1 σ_{VOS}=5mV

$P_{COMP} = 0.565$

Since all comparators must be good, the ADC yield is

$$Y_{ADC}=(P_{COMP})^{127}=(0.565)^{127}$$

$$Y_{ADC}=3.2\bullet 10^{-32}$$

This yield is essentially 0 and a standard deviation of 5mV is even not trivial to obtain with MOS comparators !

The effects of statistical variation can have dramatic effects on yield of data converters !

REVIEW

INL-based ENOB

Consider initially the continuous INL definition for an ADC where the INL of an ideal ADC is $X_{LSB}/2$

Assume $$\text{INL}= \upsilon X_{LSBR} = \upsilon \frac{X_{REF}}{2^{n_R}}$$

where X_{LSBR} is the LSB based upon the defined resolution , n_R

Define the equivalent LSB by $$X_{LSBE}=\frac{X_{REF}}{2^{n_{EQ}}}$$

Thus (substituting for X_{REF} into INL expression):

$$\text{INL}=\upsilon\frac{2^{n_{EQ}}}{2^{n_R}}X_{LSBE} = \left[\upsilon 2^{n_{EQ}+1-n_R}\right]\frac{X_{LSBE}}{2}$$

Since an ideal ADC has an INL of $X_{LSB}/2$, Setting term in [ ] to 1, can solve for n_{EQ} to obtain

$$\text{ENOB} = n_{EQ} = \log_2\left(\frac{1}{2\theta}\right) = n_R\text{-}1\text{-}\log_2(\upsilon)$$

where n_R is the defined resolution

INL-based ENOB

Since the break-point INL is ideally 0, it is not related to either X_{LSB} or X_{REF}. As such, the magnitude of the break-point INL is independent of the resolution. It is thus difficult to naturally define the effective number of bits (ENOB) directly from the INL. However, since the gain (from input to interpreted output) of an ADC is ideally 1, the break-point INL is conveying about the same linearity information as the continuous-input INL. As such, the ENOB based upon the break-point INL is also defined by the same expression.

The ENOB based upon INL for both DACs and for ADCs is defined by the expression

$$\mathrm{ENOB} = n_R - 1 - \log_2(\upsilon)$$

where n_R is the specified resolution and v is the resolution in LSB at the n_R bit level.

The ENOB based upon INL for both DACs and for ADCs is defined by the expression

$$\text{ENOB} = n_R - 1 - \log_2(\upsilon)$$

where n_R is the specified resolution and v is the resolution in LSB at the n_R bit level.

Question: With this definition, is it possible for a data converter to have an ENOB that is actually larger than n_R? YES !

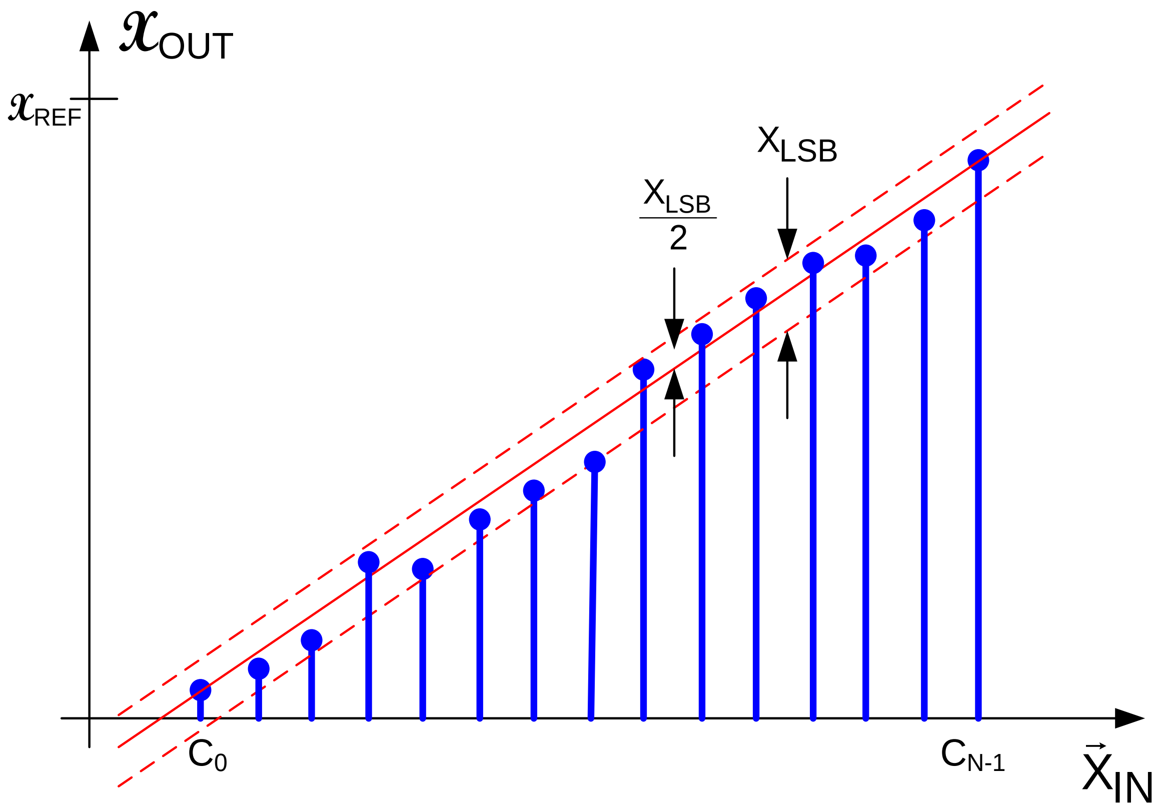
Question: What is the ENOB of any 1-bit ADC? ∞ !

Question: Is it easy to design a 4-bit ADC with an ENOB of 7 bits? YES !

Question: Is it easy to design a 14-bit ADC with an ENOB of 16 bits? No !

Question: Is ENOB (based on INL) a systematic metric?

INL-based ENOB



Interpretation of ENOB definition for a DAC:

A DAC with n_{EFF} bits (ENOB) of resolution should have all outputs bounded by +/- $X_{LSB}/2$ from the fit line so distance between fit line and upper/lower bounding lines determines the ENOB

INL-based ENOB

The ENOB based upon INL for both DACs and for ADCs is defined by the expression

$$\text{ENOB} = n_R - 1 - \log_2(\upsilon)$$

where n_R is the specified resolution and v is the resolution in LSB at the n_R bit level.

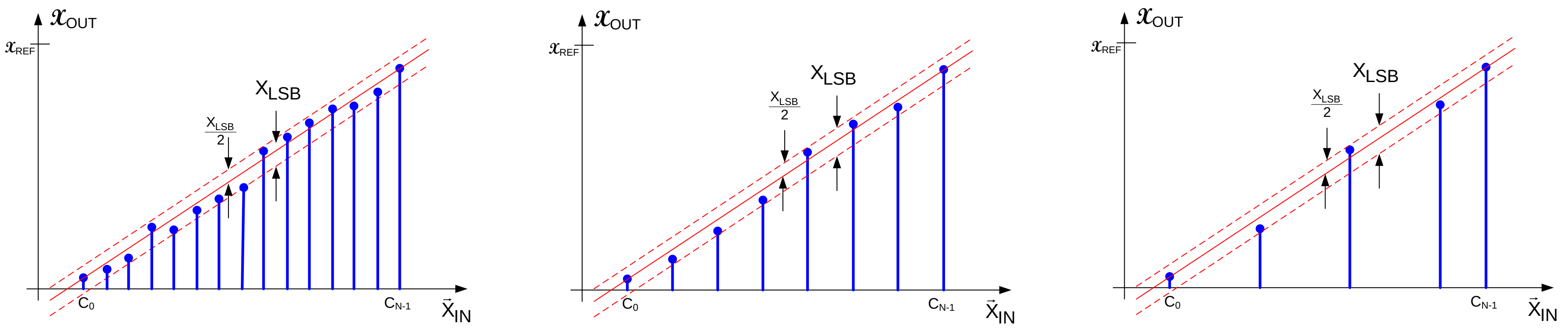
Observation: The ENOB was defined relative to a fit line and was not dependent upon the number of DAC levels or the number of break points in the ADC

Question: Then, why does n_R appear in the ENOB expression?

INL-based ENOB

Question: Then, why does n_R appear in the ENOB expression?

$$\text{ENOB} = n_R - 1 - \log_2(\upsilon)$$



Normalization was with respect to the LSB which is dependent upon n_R

INL-based ENOB

Theorem: The INL ENOB is an inherent property of a data converter independent of the number of bits of resolution specified for a data converter

Proof: Assume a data converter has n_{RA} bits of resolution and an INL of v_A LSB and a converter with the same linearity was specified with n_{RB} bits of resolution and an INL of v_B LSB.

Since there are simply two representations of the same nonlinearity, the absolute INL will be the same for both representations. That is,

$$INL_A = INL_B \qquad (1)$$

Based upon the first specification, the INL can be expressed as

$$INL_A = \upsilon_A X_{LSBA} \qquad (2)$$

But since it is assumed to have n_{RA} bits of resolution

$$\frac{X_{LSBA}}{X_{REF}} = 2^{-n_{RA}} \qquad (3)$$

INL-based ENOB

Proof (cont)

Thus we obtain the expression

$$INL_A = \upsilon_A 2^{-n_{RA}} X_{REF} \qquad (4)$$

and the ENOB is given by

$$ENOB_A = n_{RA} - 1 - \log_2(\upsilon_A) \qquad (5)$$

Substituting from (4) into (5) we obtain

$$ENOB_A = \log_2(X_{REF}) - 1 - \log_2(INL_A) \qquad (6)$$

By a similar argument we obtain

$$ENOB_B = n_{RB} - 1 - \log_2(\upsilon_B) \qquad (7)$$

and

$$ENOB_B = \log_2(X_{REF}) - 1 - \log_2(INL_B) \qquad (8)$$

Now,since $INL_A = INL_B$, it follows that

$$ENOB_A = ENOB_B \qquad \blacksquare$$

INL-based ENOB

Theorem: The INL-based ENOB can be equivalently expressed as

$$\text{ENOB}=\log_2\left(X_{REF}\right)-\log_2\left(INL_{REF}\right)-1$$

where INL_{REF} is the INL expressed relative to X_{REF}.

Proof: follows directly from proof of previous theorem

$$\text{ENOB}=\text{ENOB}_A= \log_2\left(X_{REF}\right)-1-\log_2\left(INL_A\right)=\log_2\left(X_{REF}\right)-1-\log_2\left(INL_{REF}\right)$$

To avoid possible misinterpretation, INL_{REF} defined below

$$INL_{REF}=\frac{INL_V}{V_{REF}}$$

where INL_V is the deviation in volts from the end-point fit line and $X_{REF}=V_{REF}$

INL-based ENOB

Theorem: The INL-based ENOB can be equivalently expressed as

$$\text{ENOB}=\log_2\left(X_{REF}\right)-\log_2\left(\text{INL}_{REF}\right)-1$$

where INL_{REF} is the INL expressed relative to X_{REF}.

Observe the INL-based ENOB does not depend upon the number of bits of resolution !

Can the INL-based ENOB on an n-bit ADC or DAC exceed n?

The answer is YES but in such a data converter it would probably be relatively easy to increase the number of bits while maintaining the ENOB and without increasing the number of bits, applications would probably be limited

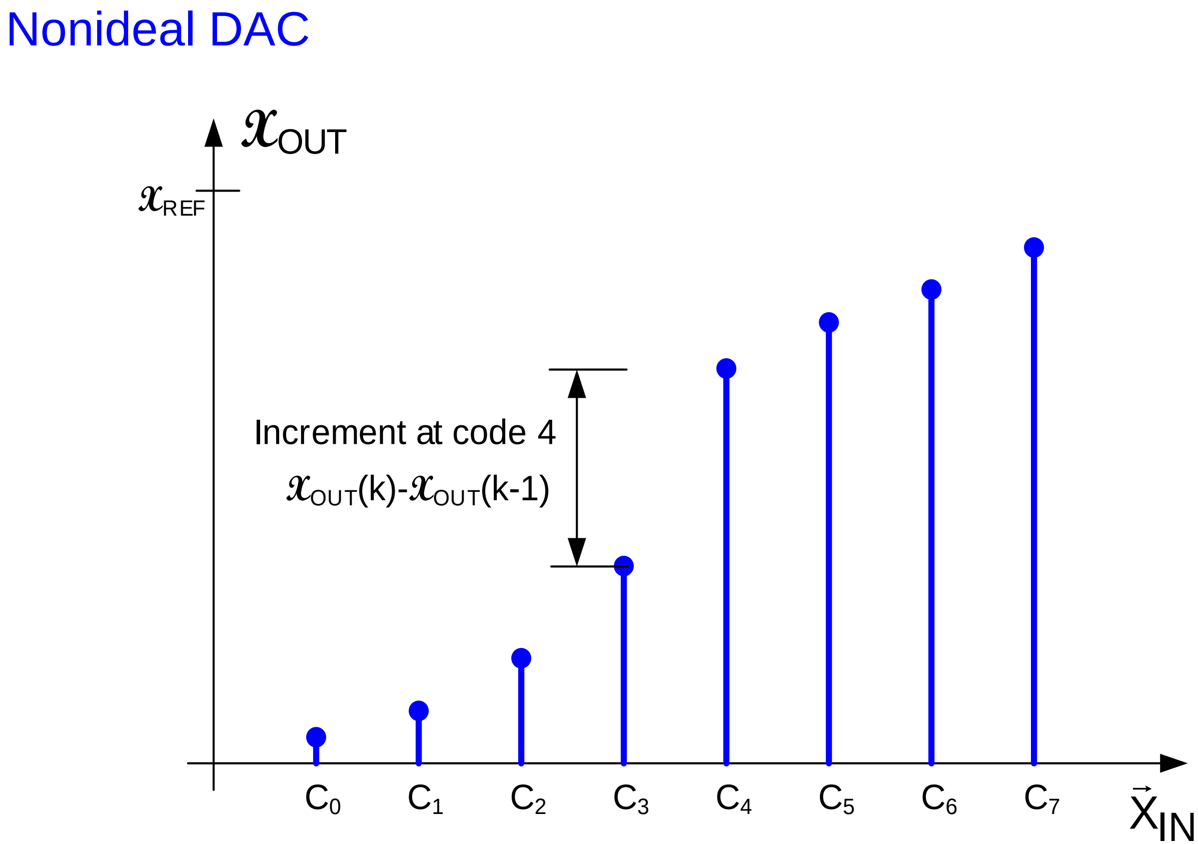
If the INL-based ENOB of a data converter exceeds n, it is probably over-designed

Designing a data converter of more than 1 bit that has a high number of bits of linearity is challenging

Performance Characterization of Data Converters

- Static characteristics
  - Resolution
  - Least Significant Bit (LSB)
  - Offset and Gain Errors
  - Absolute Accuracy
  - Relative Accuracy
  - Integral Nonlinearity (INL)
  - Differential Nonlinearity (DNL)
  - Monotonicity (DAC)
  - Missing Codes (ADC)
  - Quantization Noise
  - Low-f Spurious Free Dynamic Range (SFDR)
  - Low-f Total Harmonic Distortion (THD)
  - Effective Number of Bits (ENOB)
  - Power Dissipation

Differential Nonlinearity (DAC)

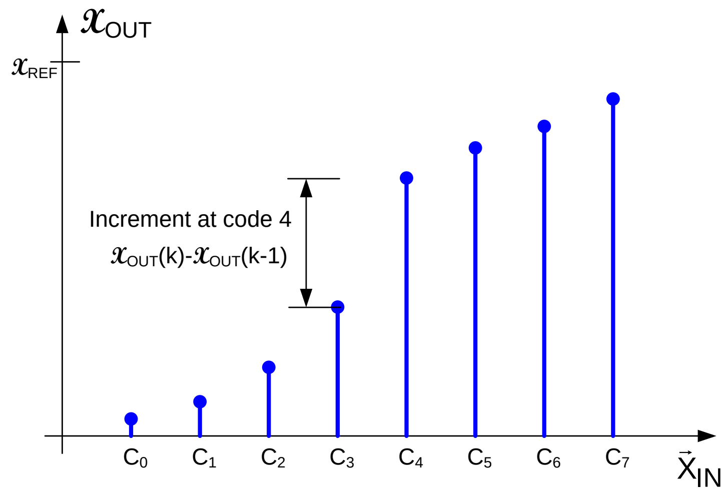


DNL(k) is the actual increment from code (k-1) to code k minus the ideal increment normalized to X_{LSB}

$$\mathrm{DNL}(k)=\frac{X_{OUT}(k)-X_{OUT}(k-1)-X_{LSB}}{X_{LSB}}$$

Differential Nonlinearity (DAC)

Nonideal DAC



Increment at code k is a signed quantity and will be negative if $X_{OUT}(k)<X_{OUT}(k-1)$

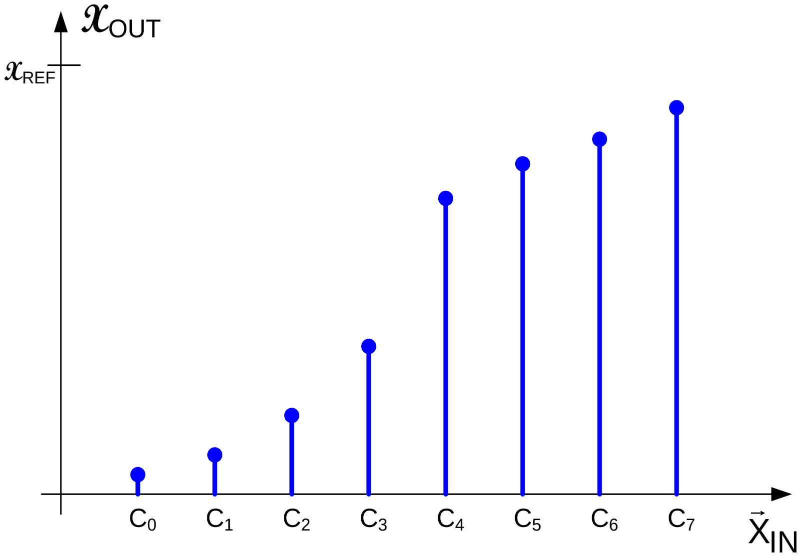
$$\mathrm{DNL}(k)=\frac{X_{OUT}(k)-X_{OUT}(k-1)-X_{LSB}}{X_{LSB}}$$

$$\mathrm{DNL}=\max_{1\leq k\leq N-1}\left\{\left|\mathrm{DNL}(k)\right|\right\}$$

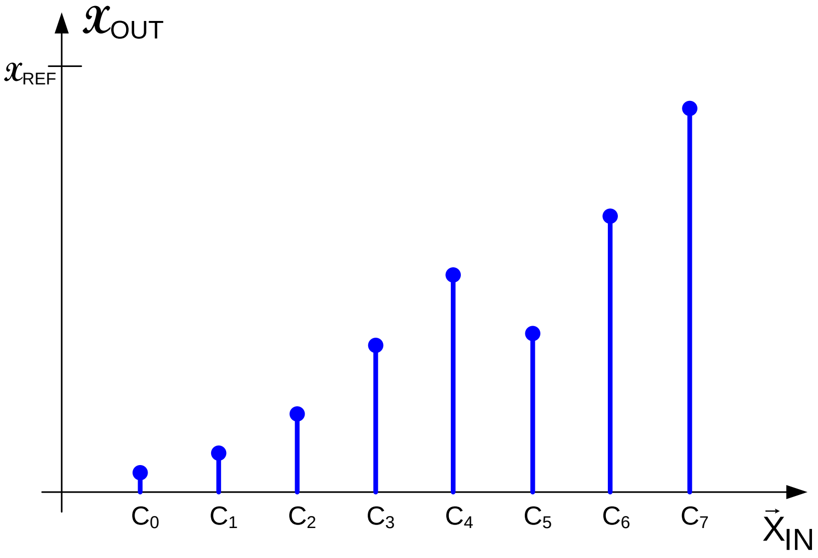
DNL=0 for an ideal DAC

Monotonicity (DAC)

Nonideal DAC



Monotone DAC



Non-monotone DAC

Definition:

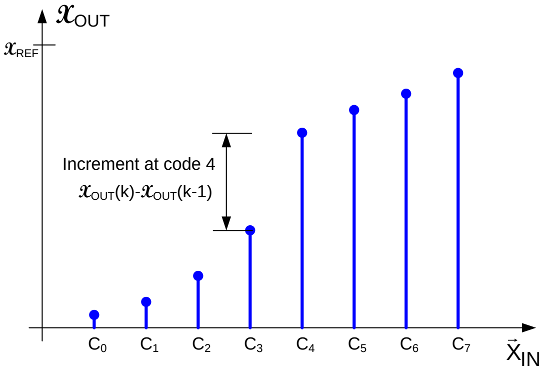
A DAC is monotone if $x_{OUT}(k) > x_{OUT}(k-1)$ for all k

Theorem:

A DAC is monotone if DNL(k)> -1 for all k

Differential Nonlinearity (DAC)

Nonideal DAC



Theorem: The INL_k of a DAC (when corrected for gain error and offset) can be obtained from the DNL by the expression

$$INL_k = \sum_{i=1}^{k} DNL(i)$$

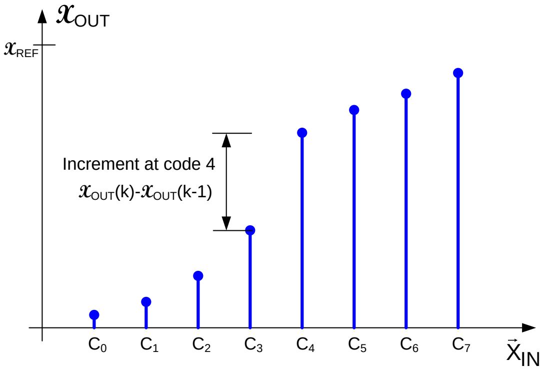
Caution: Be careful about using this theorem to measure the INL since errors in DNL measurement (or simulation) can accumulate

Corollary: The DNL of a DAC (when corrected for gain error and offset) can be expressed as

$$DNL(k) = INL_k - INL_{k-1}$$

Differential Nonlinearity (DAC)

Nonideal DAC



Theorem: If the INL of a DAC satisfies the relationship

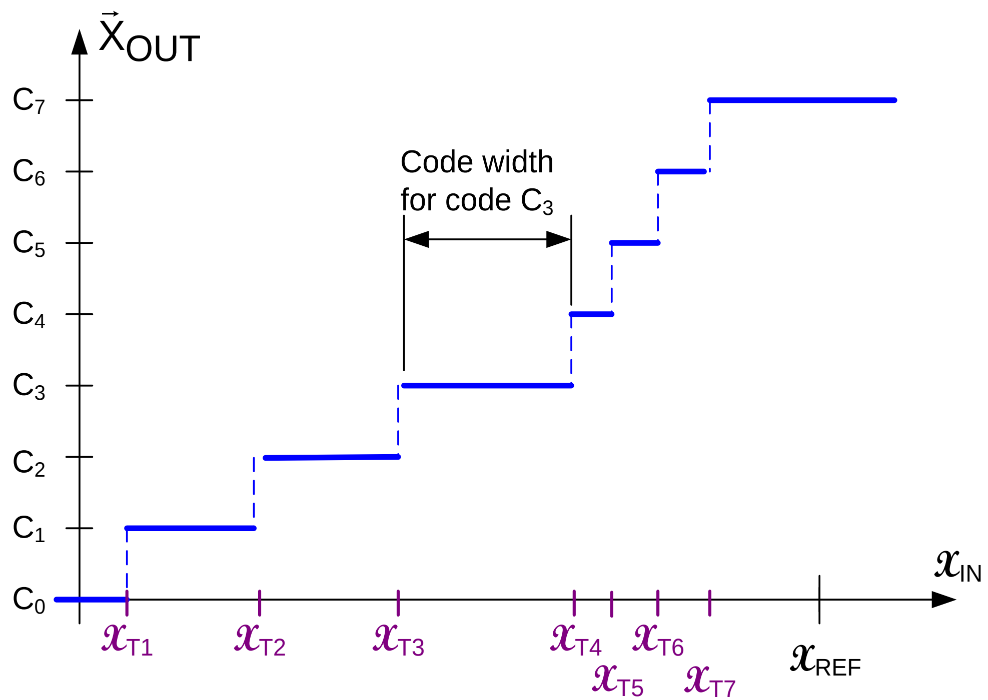
$$\text{INL} < \frac{1}{2} X_{LSB}$$

then the DAC is monotone

Note: This is a necessary but not sufficient condition for monotonicity

Differential Nonlinearity (ADC)

Nonideal ADC

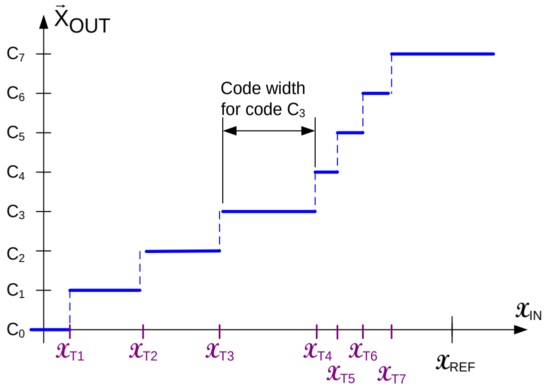


DNL(k) is the code width for code k – ideal code width normalized to X_{LSB}

$$DNL(k)=\frac{x_{T(k+1)}-x_{Tk}-x_{LSB}}{x_{LSB}}$$

Differential Nonlinearity (ADC)

Nonideal ADC



$$\mathrm{DNL}(k)=\frac{x_{T(k+1)}-x_{Tk}-x_{LSB}}{x_{LSB}}$$

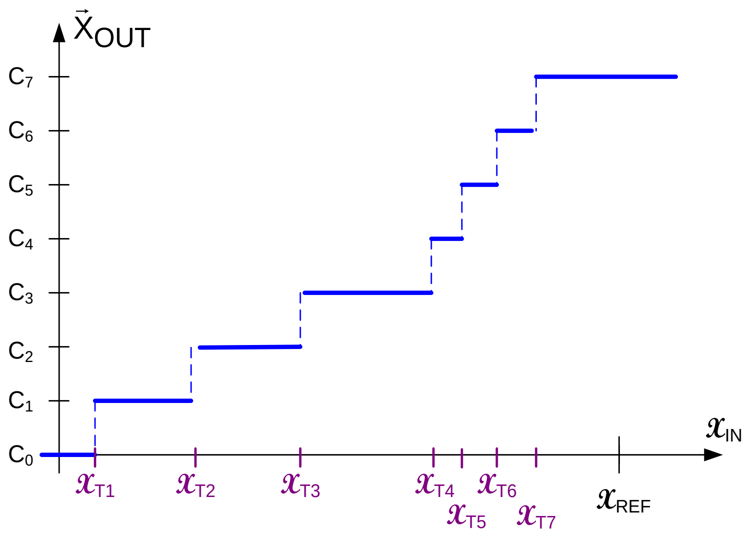
$$\mathrm{DNL}=\max_{2\le k\le N-1}\left\{\left|\mathrm{DNL}(k)\right|\right\}$$

DNL=0 for an ideal ADC

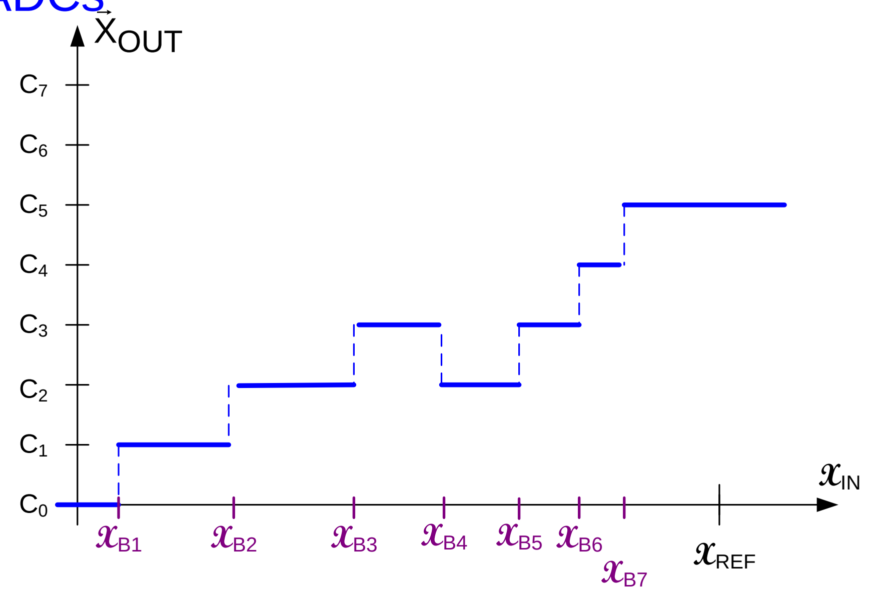
Note: In some nonideal ADCs, two or more break points could cause transitions to the same code C_k making the definition of DNL ambiguous

Monotonicity in an ADC

Nonideal ADCs



Monotone ADC



Nonmonotone ADC

Definition: An ADC is monotone if the

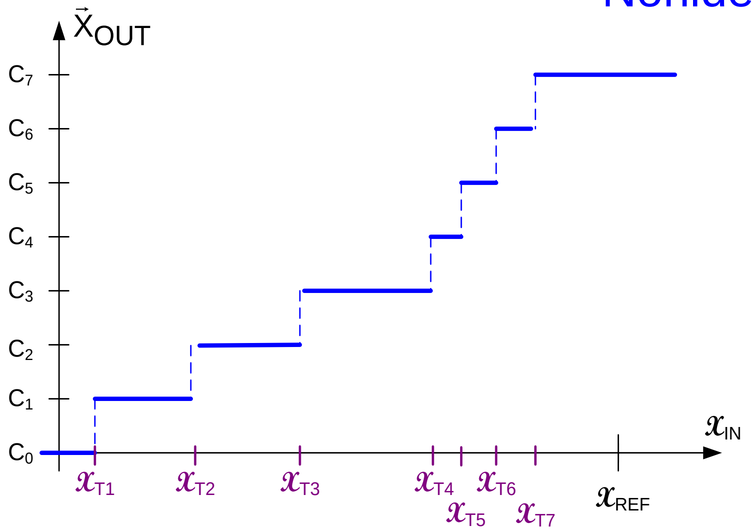
$$\vec{X}_{OUT}(x_k) \geq \vec{X}_{OUT}(x_m) \quad \text{whenever} \quad x_k \geq x_m$$

Note: Have used x_{Bk} instead of x_{Tk} in figure on right since more than one transition point corresponds to a given code

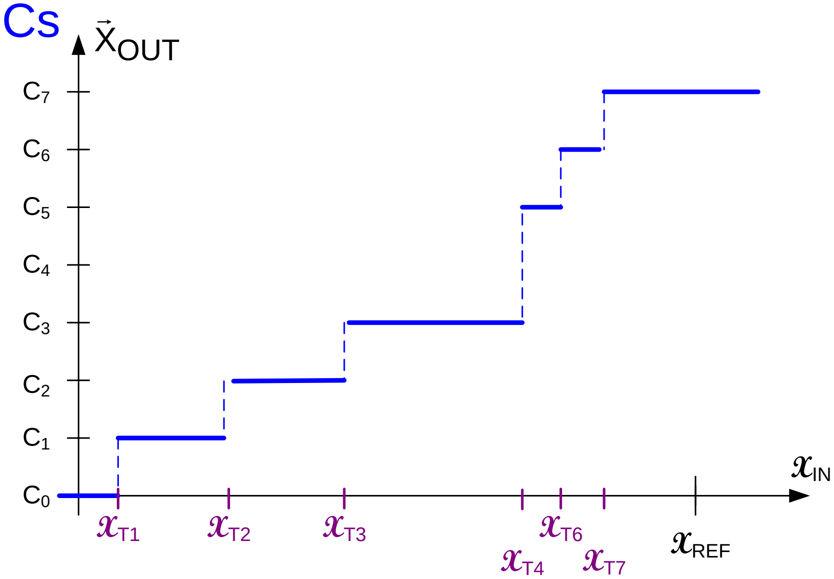
Note: Some authors do not define monotonicity in an ADC.

Missing Codes (ADC)

Nonideal ADCs



No missing codes



One missing code

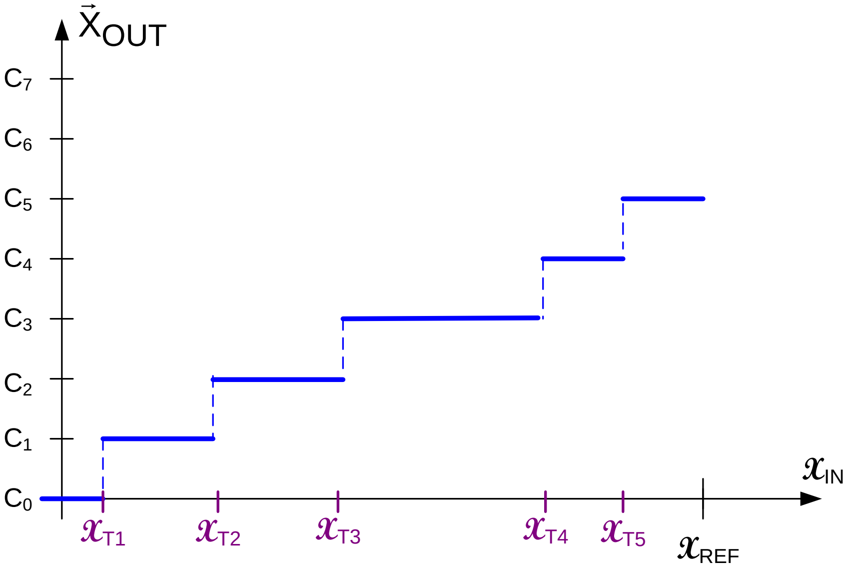
Definition: An ADC has no missing codes if there are N-1 transition points and a single LSB code increment occurs at each transition point. If these criteria are <u>not</u> satisfied, we say the ADC has missing code(s).

Note: With this definition, all codes can be present but we still say it has "missing codes"

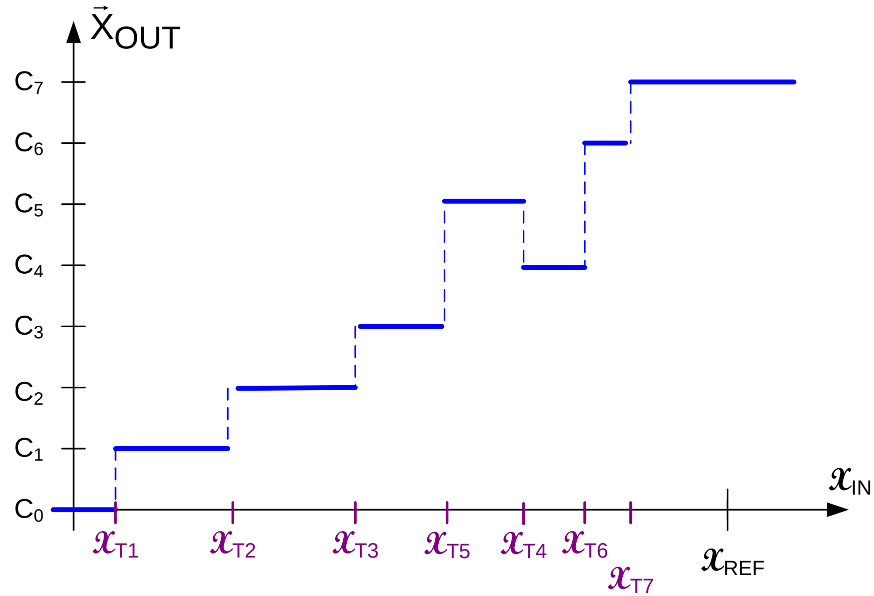
Note: Some authors claim that missing codes in an ADC are the counterpart to nonmonotonicity in a DAC. This association is questionable.

Missing Codes (ADC)

Nonideal ADCs



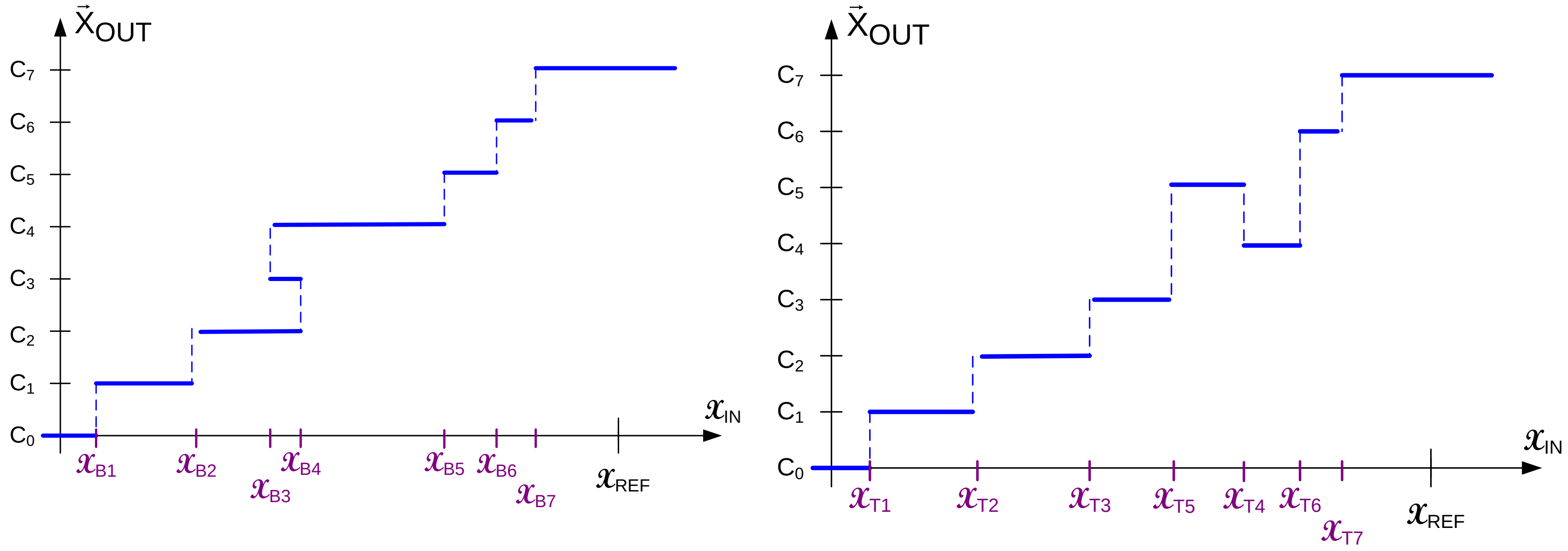
Missing codes



Missing code with all codes present

Weird Things Can Happen

Nonideal ADCs



- Multiple outputs for given inputs
- All codes present but missing codes

Be careful on definition and measurement of linearity parameters to avoid having weird behavior convolute analysis, simulation or measurements

Most authors (including manufacturers) are sloppy with their definitions of data converter performance parameters and are not robust to some weird operation

LSB Definition

X_{LSB} appears in many performance specifications but the definition of X_{LSB} is generally not given. This can cause modest inconsistencies in the definition of some performance specifications.

What is X_{LSB}?

LSB Definition

X_{LSB} appears in many performance specifications but the definition of X_{LSB} is generally not given. This can cause modest inconsistencies in the definition of some performance specifications.

What is X_{LSB}?

Conventional Wisdom X_{LSB}

$$X_{LSB} = \frac{X_{REF}}{2^{n_R}}$$

(X_{LSB} determined by specified resolution and can not be measured)

Alternate LSB Definition

X_{LSB} appears in many performance specifications but a distinction in X_{LSB} that differs from that obtained from specified values for X_{REF} and n_R is generally not given. This can cause modest inconsistencies in the definition of some performance specifications.

DAC

Alternate definitions of X_{LSB}
where N is the measured number of DAC output levels

$$X_{LSB}=\frac{X_{REF}}{N}$$

---

where N is the measured number of DAC output levels and $X_0(N\text{-}1)$ and $X_0(0)$ are last and first outputs respectively

$$X_{LSB}=\frac{X_0(N\text{-}1)-X_0(0)}{N\text{-}1}$$

---

useful when extreme values do not occur at minimum and maximum input codes

$$X_{LSB}=\frac{\max\limits_{k}\{X_0(k)\}-\min\limits_{k}\{X_0(k)\}}{N\text{-}1}$$

---

useful for determining worst-case resolution of a DAC

$$X_{LSB}=\max_{k}\{X_0(k)-X_0(k\text{-}1)\}$$

ADC

Similar definitions can be made for X_{LSB} of an ADC based upon the breakpoints

Alternate LSB Definition

Is the concept of an LSB that is based upon measurements useful?

In many control applications, the largest gap between outputs of a DAC is often of interest and though that is ideally V_{LSB}, it may differ significantly

ENOB based upon DNL

If it is assumed that an acceptable DNL for an n-bit data converter is $X_{LSB}/2$, then if the DNL is different from $X_{LSB}/2$, the effective number of bits essentially changes.

An ENOB based upon the DNL can be defined (homework problem)

ENOB relative to resolution

If an n-bit data converter has an INL of ¼ LSB, it is really performing from a linearity viewpoint at the n+1 bit level and if it has an INL of 1/8 LSB it is really performing at the n+2 bit level

Correspondingly, if it has a DNL of ¼ LSB, it is also performing from a differential linearity viewpoint at the n+1 bit level

Observation: The ENOB of a data converter can exceed the number of bits of resolution of the data converter

Observations: Some applications benefit from an ENOB that exceeds the resolution of the data converter

Limitations of INL & DNL in Characterizing Linearity

See Lecture 4

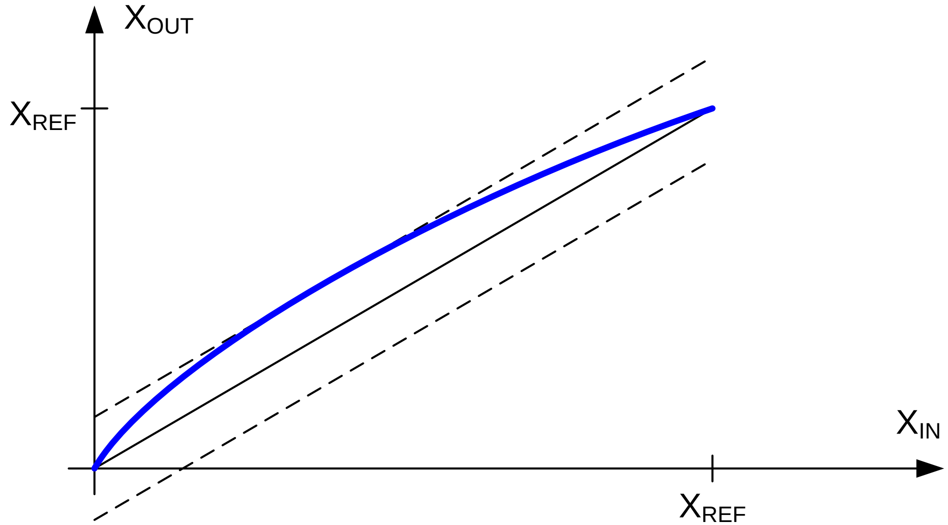
- **INL is a key parameter that is attempting to characterize the overall linearity of a DAC !**
- **INL is a key parameter that is attempting to characterize the overall linearity of an ADC !**
- **DNL is a key parameter that is attempts to characterize the local linearity of a DAC !**
- **DNL is a key parameter that is attempts to characterize the local linearity of an ADC !**

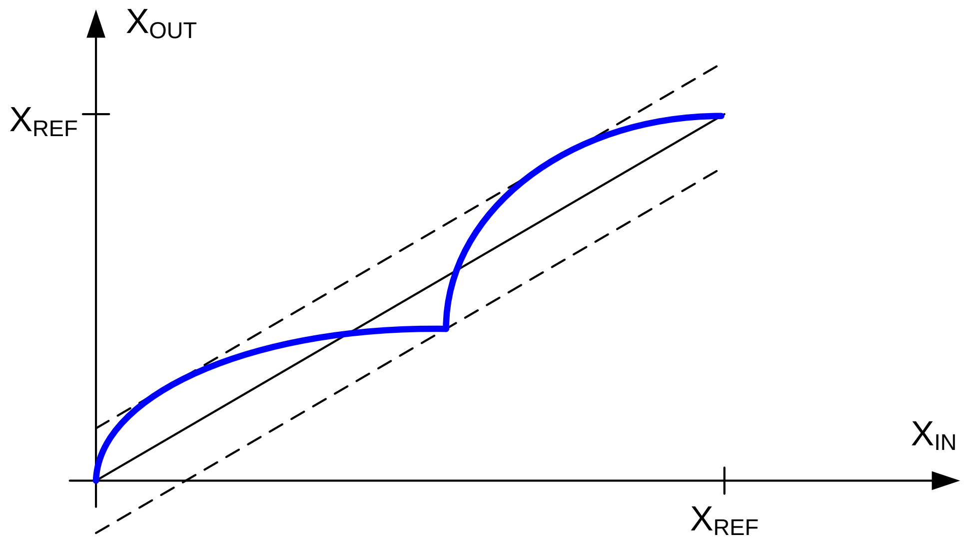
Are INL and DNL effective at characterizing the linearity of a data converter?

Limitations of INL & DNL in Characterizing Linearity

Consider the following 4 transfer characteristics, all of which have the same INL

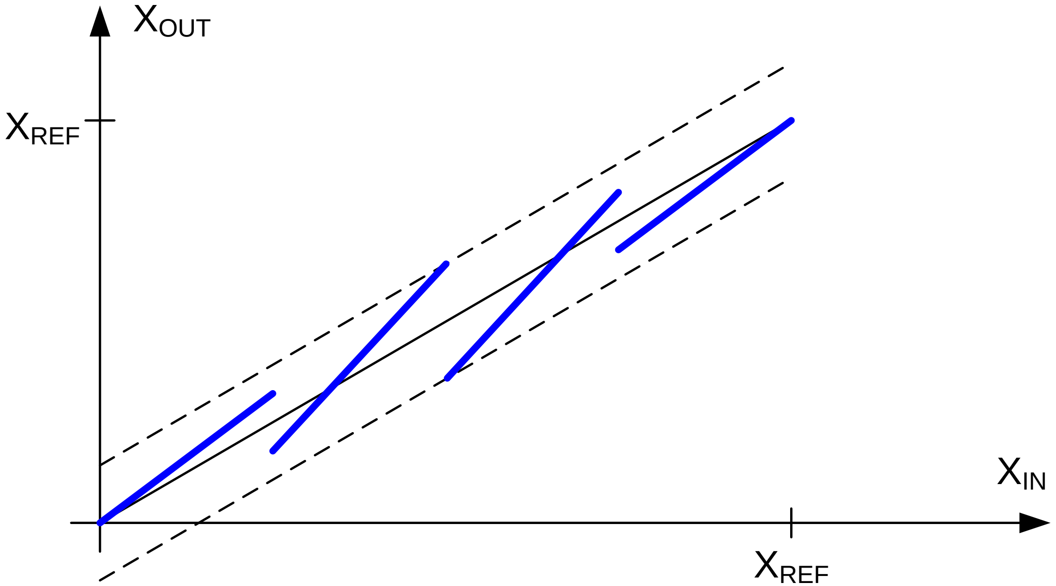
See Lecture 4

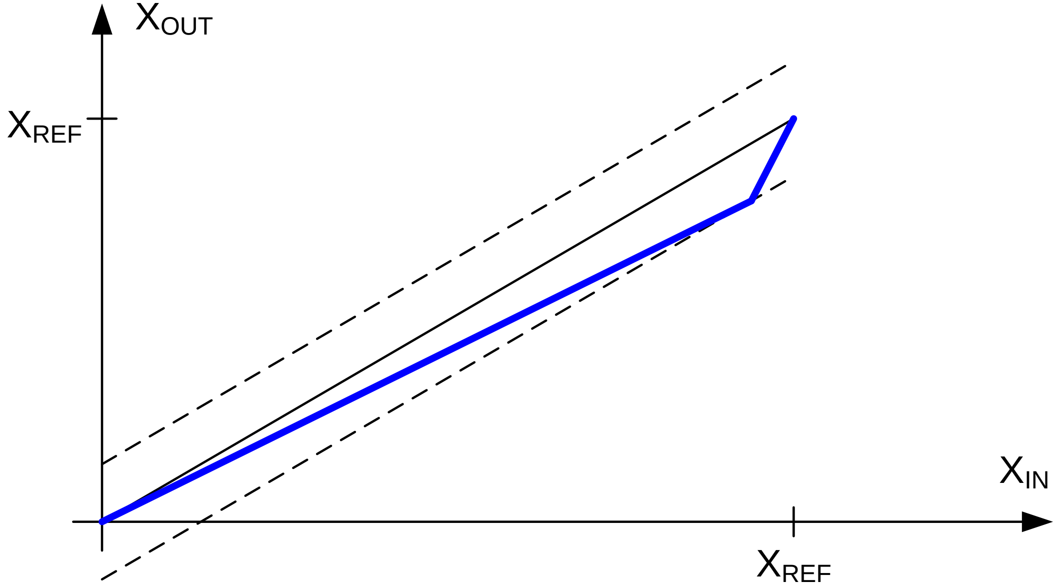




Limitations of INL & DNL in Characterizing Linearity

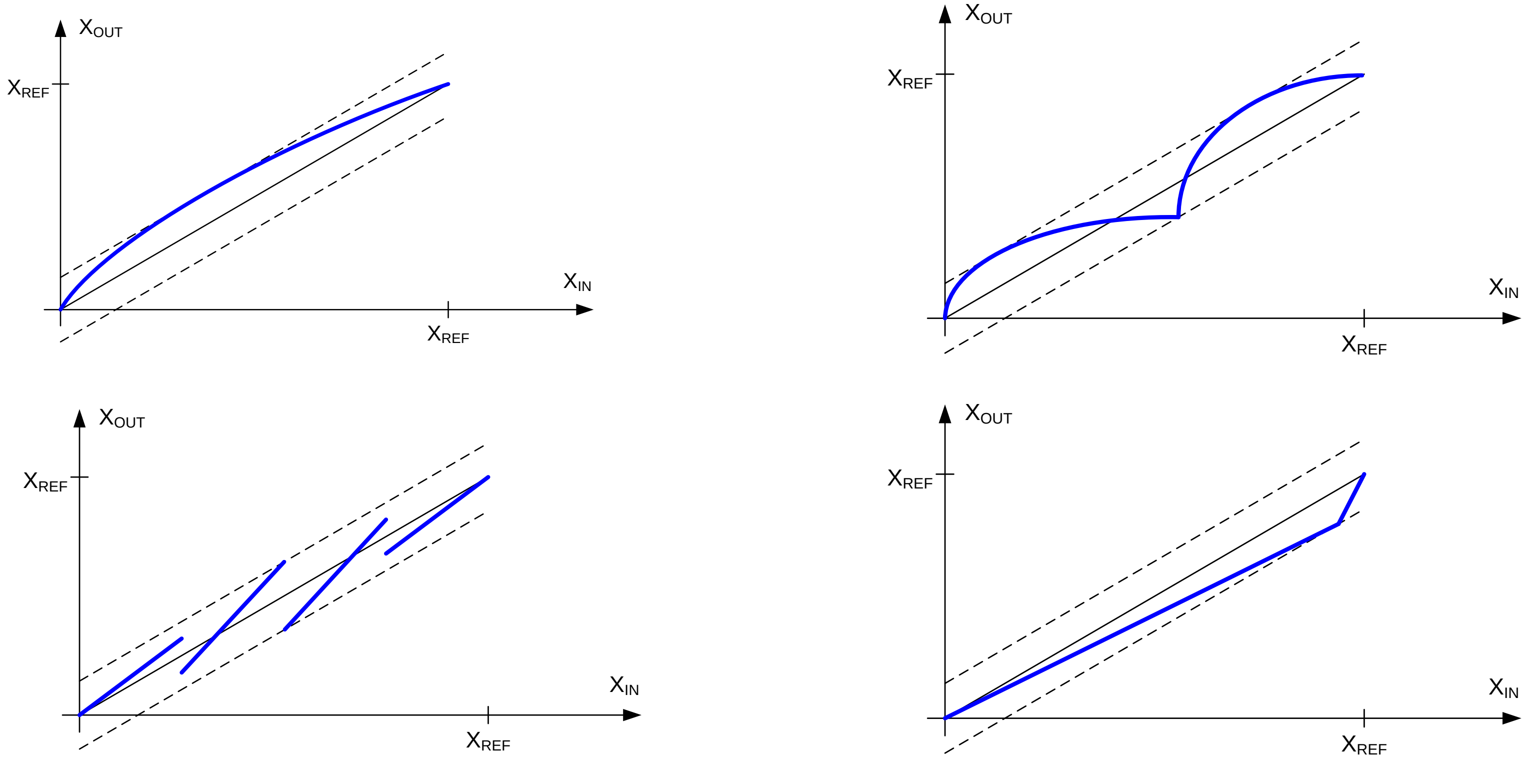
See Lecture 4





Limitations of INL & DNL in Characterizing Linearity

See Lecture 4



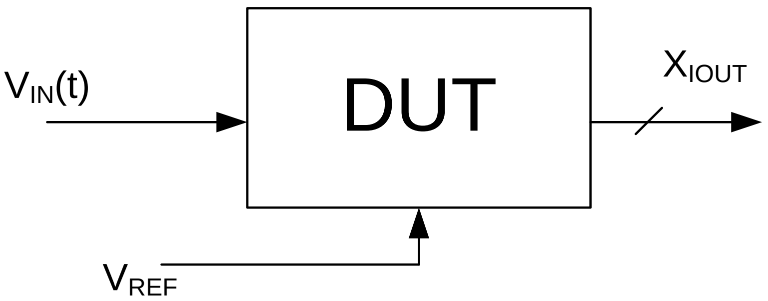
Although same INL, dramatic difference in performance particularly when inputs are sinusoidal-type excitations

INL also gives little indication of how performance degrades at higher frequencies

Spectral Analysis often used as an alternative (and often more useful in many applications) linearity measure for data converters

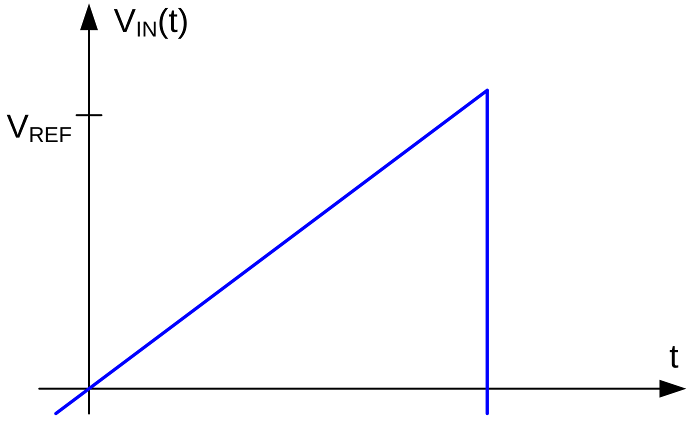
Linearity Measurements (testing)

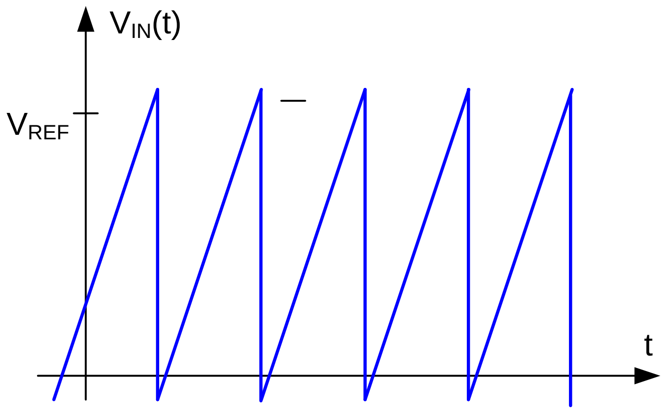
Consider ADC



Linearity testing often based upon code density testing

Code density testing:



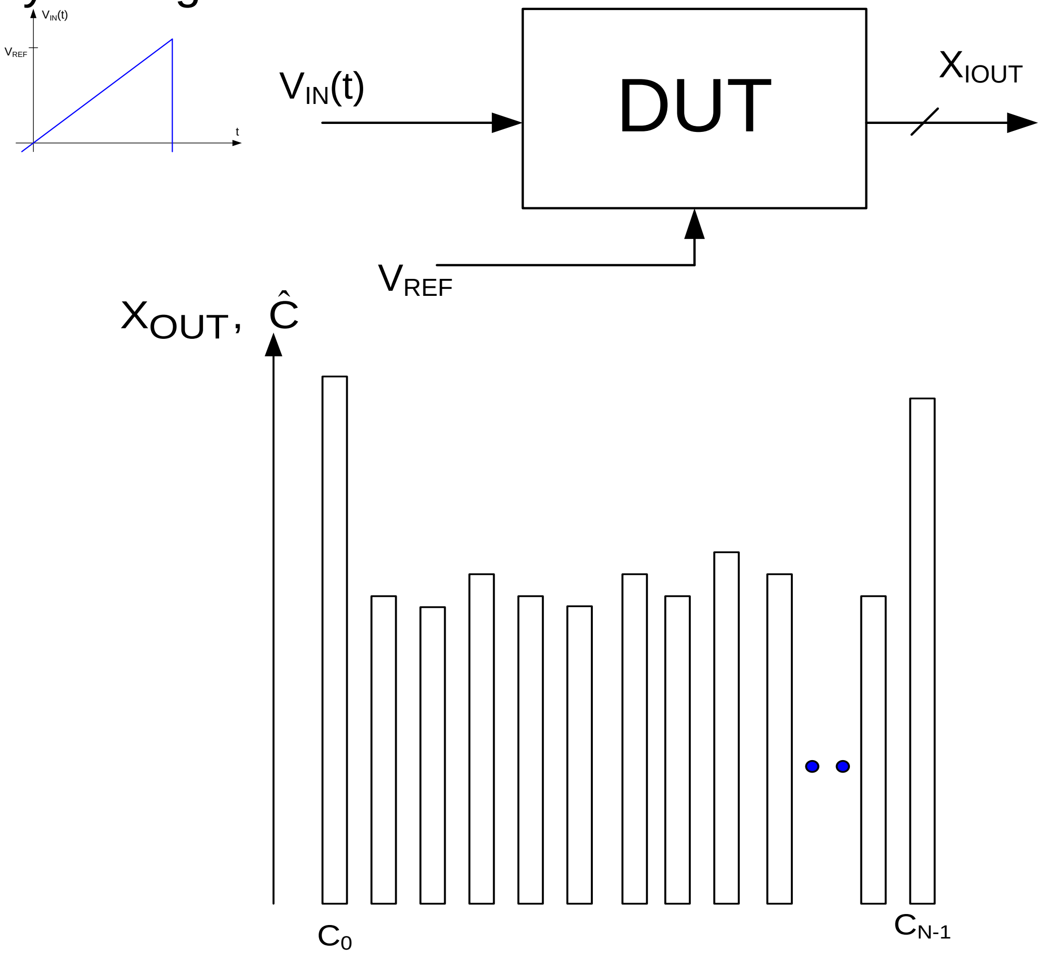


Ramp or multiple ramps often used for excitation

Linearity of test signal is critical (typically 3 or 4 bits more linear than DUT)

Linearity Measurements (testing)

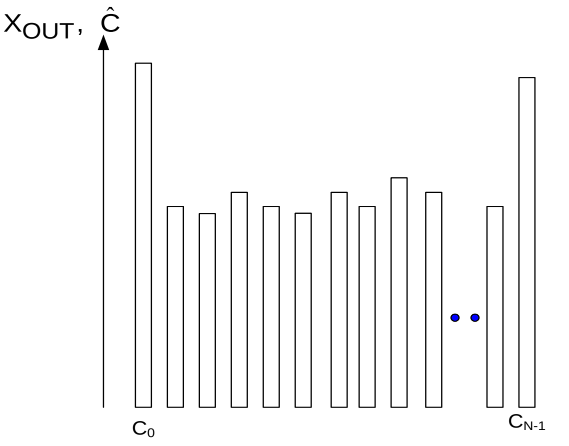
Code density testing:



- First and last bins generally have many extra counts (and thus no useful information)
- Typically average 16 or 32 hits per code

Linearity Measurements (testing)

Code density testing:



$$\bar{C} = \frac{\sum_{i=1}^{N-2} \hat{C}_i}{N-2}$$

$$DNL_i = \frac{\hat{C}_i - \bar{C}}{\bar{C}}$$

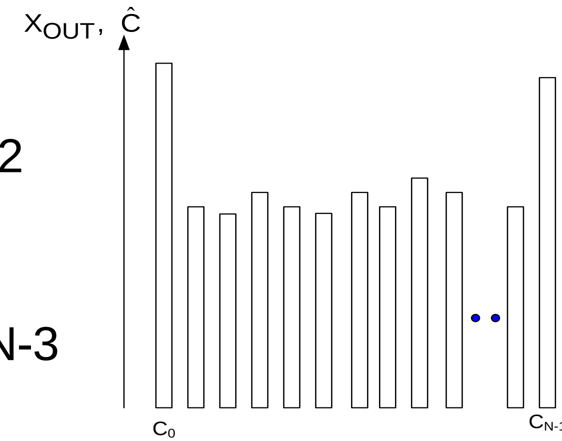
$$INL_i = \begin{cases} 0 & i=0,N-2 \\ \dfrac{\left[\sum_{k=1}^{i} \hat{C}_k\right] - i\bar{C}}{\bar{C}} & 1 \le i \le N-3 \end{cases}$$

$$DNL = \max_{1 \le i \le N-2} \left\{ \left| DNL_i \right| \right\}$$

$$INL = \max_{1 \le i \le N-3} \left\{ \left| INL_i \right| \right\}$$

Linearity Measurements (testing)

Code density testing:



$$DNL_i = \frac{\hat{C}_i - \overline{C}}{\overline{C}} \qquad INL_i = \begin{cases} 0 & i=0, N-2 \\ \dfrac{\left[\sum\limits_{k=1}^{i} \hat{C}_k\right] - i\overline{C}}{\overline{C}} & 1 \le i \le N-3 \end{cases}$$

$$DNL = \max_{1 \le i \le N-2} \left\{ \left| DNL_i \right| \right\} \qquad INL = \max_{1 \le i \le N-3} \left\{ \left| INL_i \right| \right\}$$

- **Code Density Measurements are Indirect Measurements of the INL and DNL**
- **Can give very wrong information under some nonmonotone missing code scenarios**
- **Often use an average of 16 or 32 samples per code**
- **Measurement noise often 1 lsb or larger but averages out**
- **Sometimes use good sinusoidal waveform but must correct code density for this distinction**
- **Full code-density testing is costly for high-resolution low-speed data converters because of data acquisition costs**
- **Reduced code testing using servo methods is often a less costly alternative but may miss some errors**

Performance Characterization of Data Converters

- Static characteristics
  - Resolution
  - Least Significant Bit (LSB)
  - Offset and Gain Errors
  - Absolute Accuracy
  - Relative Accuracy
  - Integral Nonlinearity (INL)
  - Differential Nonlinearity (DNL)
  - Monotonicity (DAC)
  - Missing Codes (ADC)
  - Quantization Noise
  - Low-f Spurious Free Dynamic Range (SFDR)
  - Low-f Total Harmonic Distortion (THD)
  - Effective Number of Bits (ENOB)
  - Power Dissipation

Quantization Noise

- DACs and ADCs generally quantize both amplitude and time
- If converting a continuous-time signal (ADC) or generating a desired continuous-time signal (DAC) these quantizations cause a difference in time and amplitude from the desired signal
- First a few comments about Noise

What is Noise in a data converter?

Noise is a term applied to some nonideal effects of a data converter

Precise definition of noise is probably not useful

Some differences in views about what nonideal characteristics of a data converter should be referred to as noise

Types of noise:

- Random perturbations in V or I due to movement of electrons in electronic circuits
- Interfering signals generated by other systems
- Interfering signals generated by a circuit or system itself
- Error signals associated with imperfect signal processing algorithms or circuits
  - Quantization noise
  - Sample Jitter
  - Harmonic Distortion

Noise

All of these types of noise are present in data converters and are of concern when designing most data converters

Can not eliminate any of these noise types but with careful design can manage their effects to certain levels

Noise (in particular the random noise) is often the major factor limiting the ultimate performance potential of many if not most data converters

Noise

Types of noise:

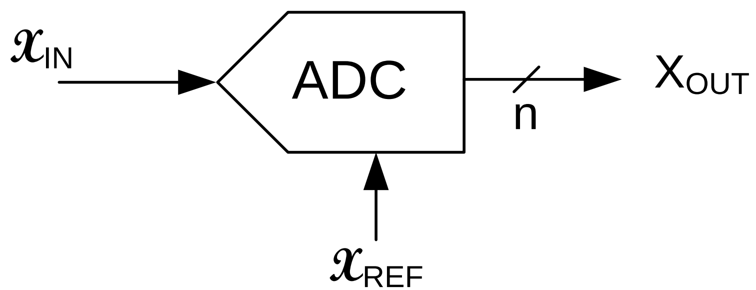
- Perturbations in V or I due to movement of electrons in electronic circuits
- Interfering signals generated by other systems
- Interfering signals generated by a circuit or system itself
- Error signals associated with imperfect signal processing algorithms or circuits
  - Quantization noise
  - Sample Jitter
  - Harmonic Distortion

Quantization noise is a significant component of this noise in ADCs and DACs and is present even if the ADC or DAC is ideal

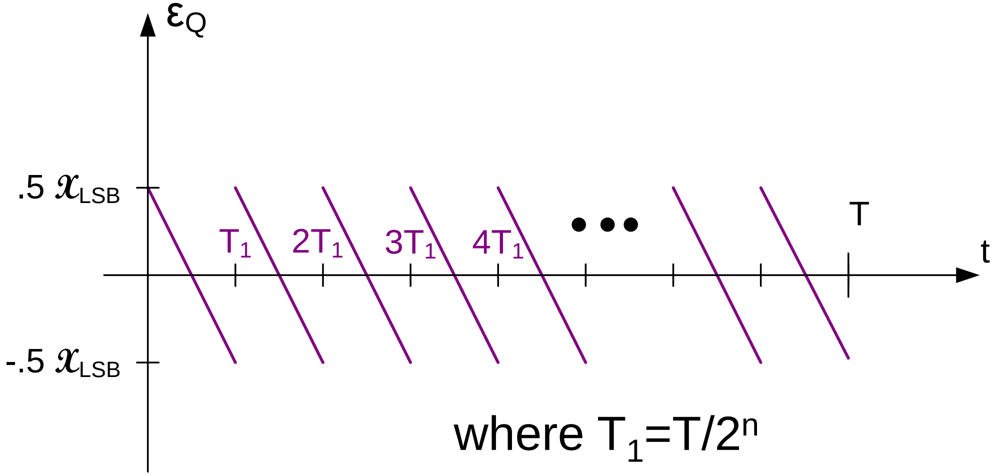
Quantization Noise in ADC

(same concepts apply to DACs)

Consider an Ideal ADC with first transition point at $0.5X_{LSB}$



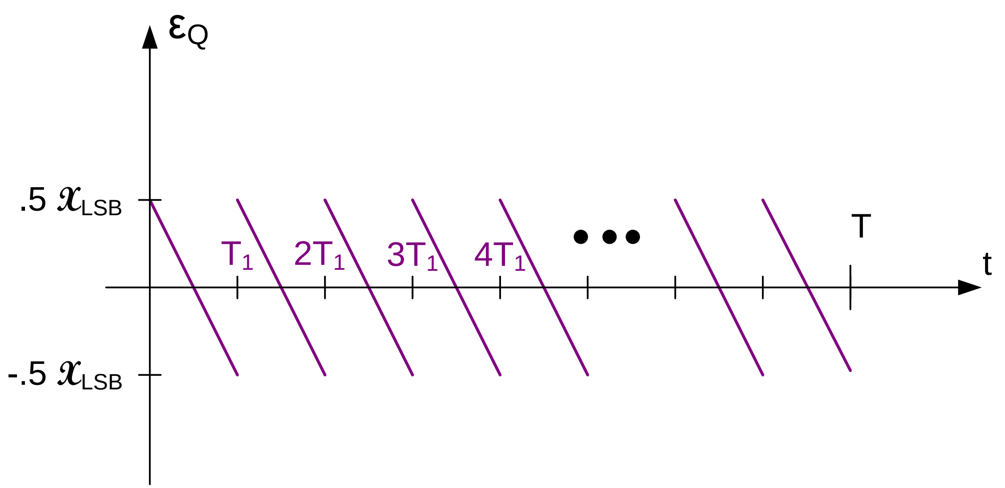
If the input is a low frequency sawtooth waveform of period T that goes from 0 to X_{REF} , the error signal in the time domain will be:



where $T_1 = T/2^n$

This time-domain waveform (after dc offset is removed) is termed the Quantization Noise for the ADC with a sawtooth (or triangular) input

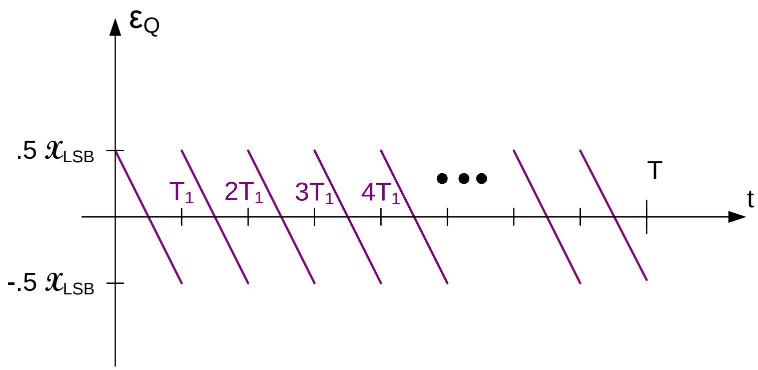
Quantization Noise in ADC

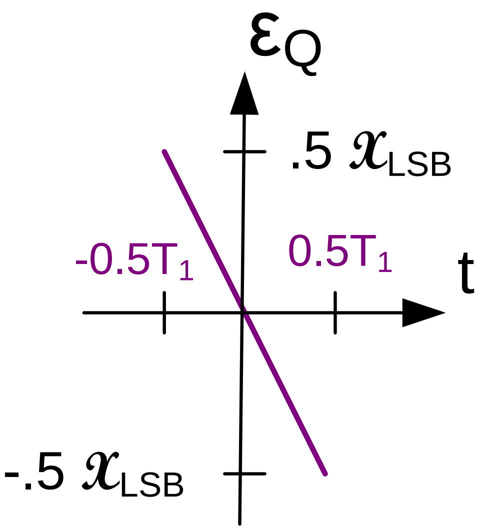


For large n, this periodic waveform "behaves" much like a random noise source that is uncorrelated with the input and can be characterized by its RMS value which can be obtained by integrating over any interval of length T_1. For notational convenience, shift the waveform to the left by $T_1/2$ units

$$E_{RMS} = \sqrt{\frac{1}{T_1}\int_{-T_1/2}^{T_1/2} \varepsilon_Q^{2}(t)\,dt}$$

Quantization Noise in ADC





$$E_{RMS} = \sqrt{\frac{1}{T_1}\int_{-T_1/2}^{T_1/2} \varepsilon_Q{}^2(t)\,dt}$$

In this interval, ε_Q can be expressed as

$$\varepsilon_Q(t) = -\left(\frac{X_{LSB}}{T_1}\right)t$$

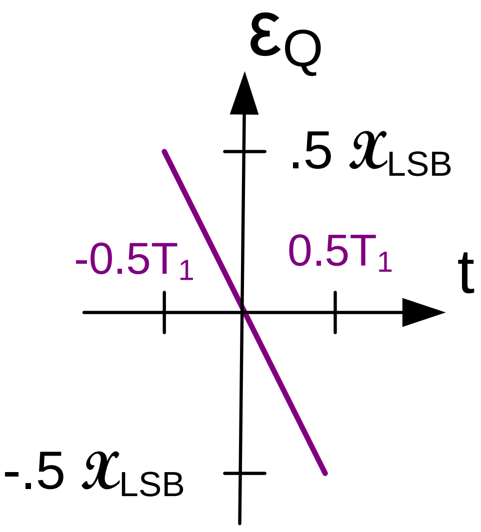
Quantization Noise in ADC

$$E_{RMS} = \sqrt{\frac{1}{T_1}\int_{-T_1/2}^{T_1/2} \varepsilon_Q{}^2(t)\,dt}$$

$$E_{RMS} = \sqrt{\frac{1}{T_1}\int_{-T_1/2}^{T_1/2} \left(-\frac{x_{LSB}}{T_1}\right)^2 t^2\,dt}$$

$$E_{RMS} = x_{LSB}\sqrt{\frac{1}{T_1^3}\left.\frac{t^3}{3}\right|_{-T_1/2}^{T_1/2}}$$

$$E_{RMS} = \frac{x_{LSB}}{\sqrt{12}}$$



$$\varepsilon_Q(t) = -\left(\frac{X_{LSB}}{T_1}\right)t$$

Quantization Noise in ADC

$$E_{RMS} = \frac{x_{LSB}}{\sqrt{12}}$$

The signal to quantization noise ratio (SNR) can now be determined. Since the input signal is a sawtooth waveform of period T and amplitude X_{REF}, it follows by the same analysis that it has an RMS value of

$$x_{RMS} = \frac{x_{REF}}{\sqrt{12}}$$

Thus the SNR is given by

$$SNR = \frac{x_{RMS}}{E_{RMS}} = \frac{x_{RMS}}{x_{LSB}} = 2^n$$

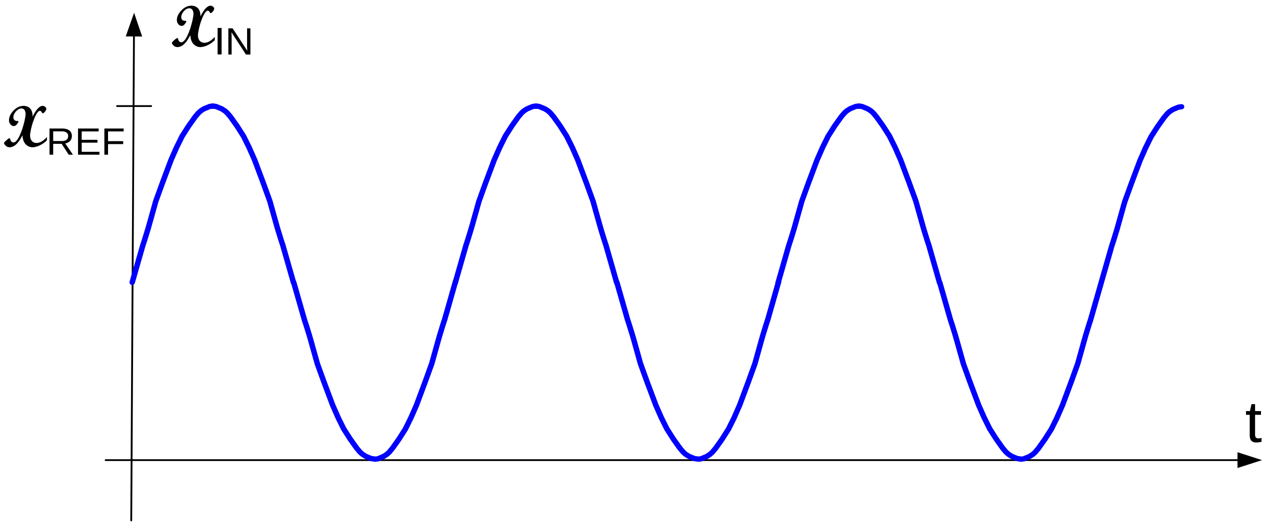
or, in dB,

$$SNR_{dB} = 20\left(n \bullet \log 2\right) = 6.02n$$

Note: dB subscript often neglected when not concerned about confusion

Quantization Noise in ADC

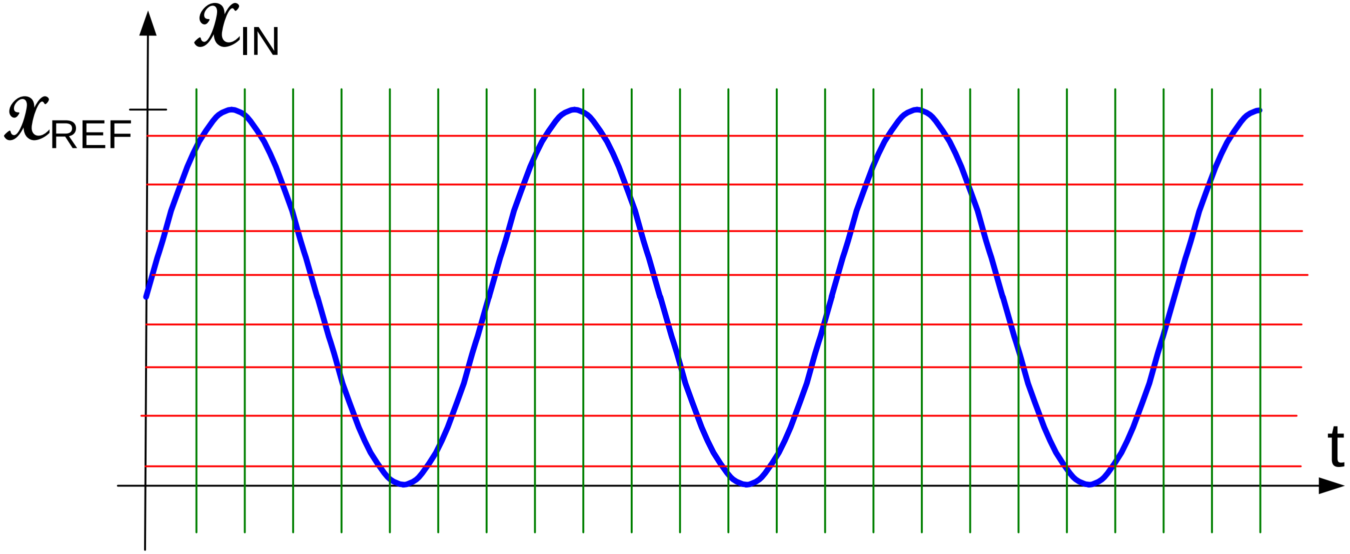
How does the SNR change if the input is a sinusoid that goes from 0 to x_{REF} centered at $x_{REF}/2$?



$$SNR = 20(n \bullet \log 2) = 6.02n$$

Quantization Noise in ADC

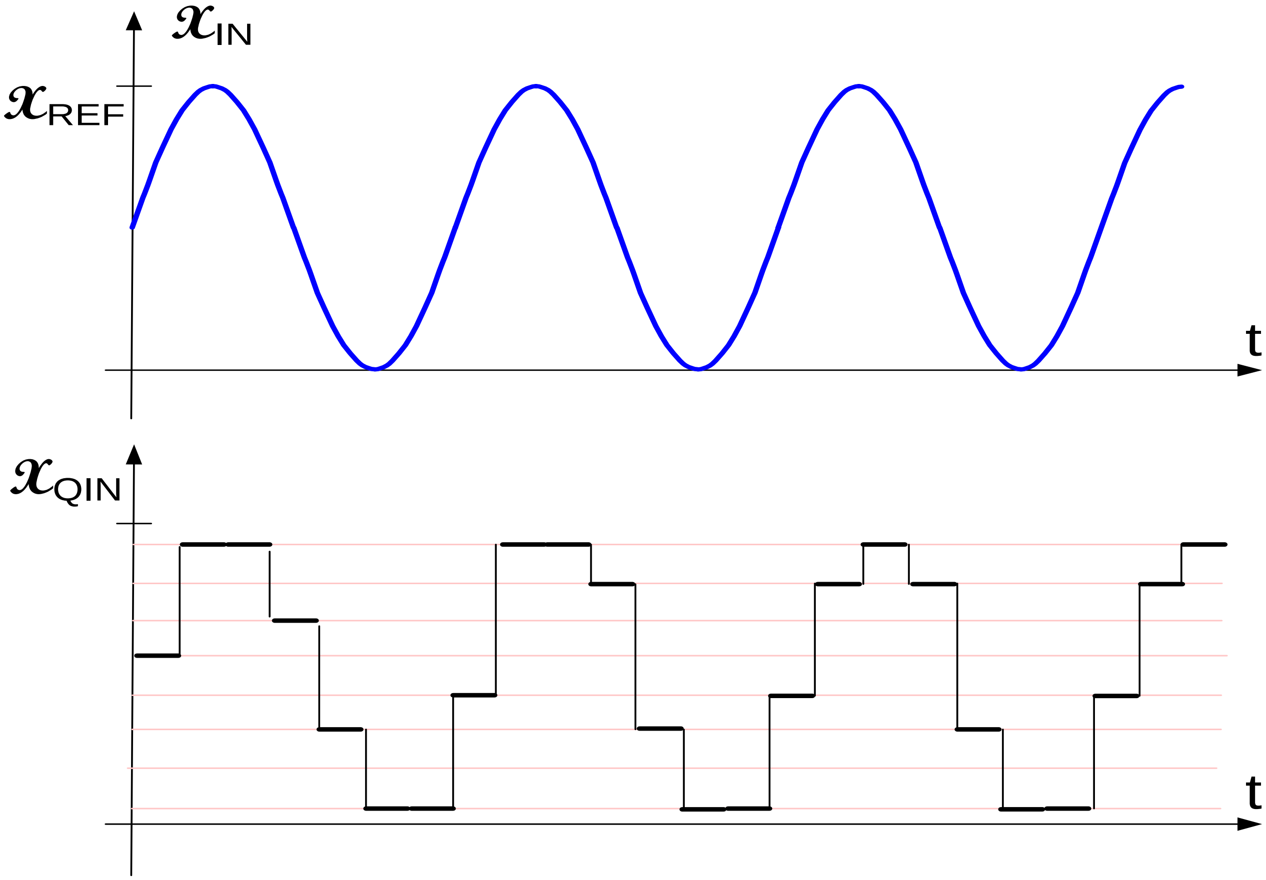
How does the SNR change if the input is a sinusoid that goes from 0 to x_{REF} centered at $x_{REF}/2$?



Time and amplitude quantization points

Quantization Noise in ADC

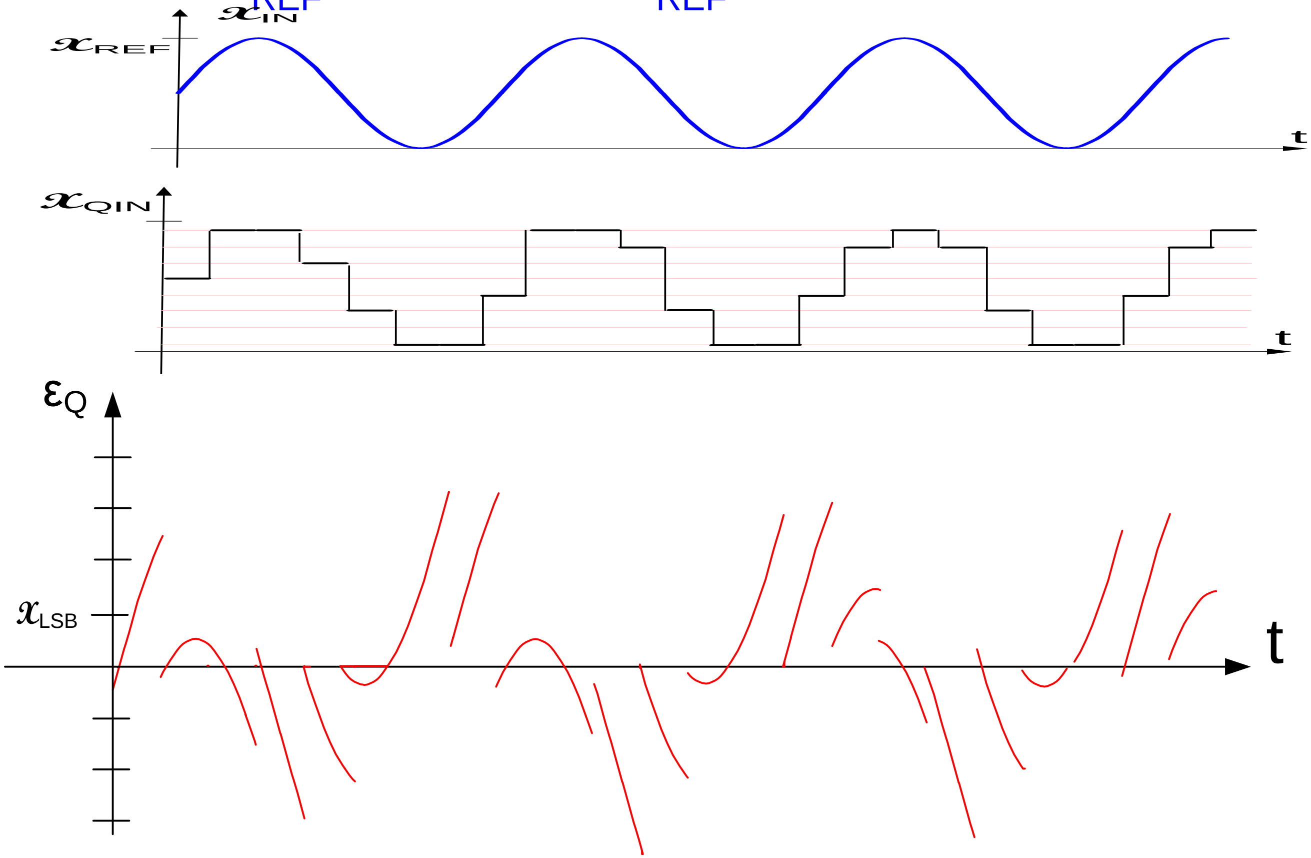
How does the SNR change if the input is a sinusoid that goes from 0 to x_{REF} centered at $x_{REF}/2$?



Time and Amplitude Quantized Waveform

Quantization Noise in ADC

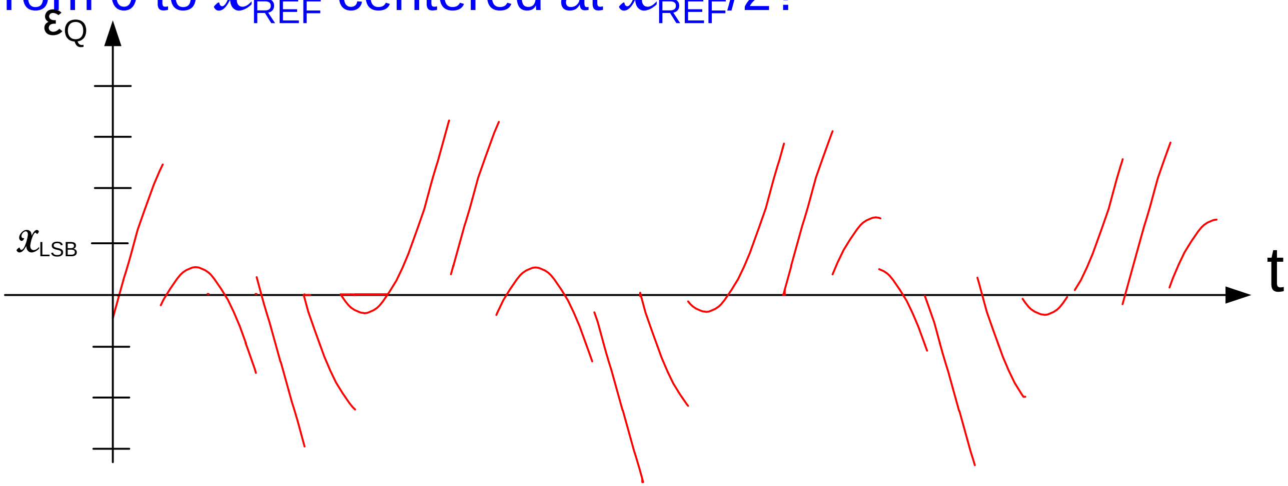
How does the SNR change if the input is a sinusoid that goes from 0 to $\mathcal{X}_{REF}$ centered at $\mathcal{X}_{REF}/2$?



Error waveform

Quantization Noise in ADC

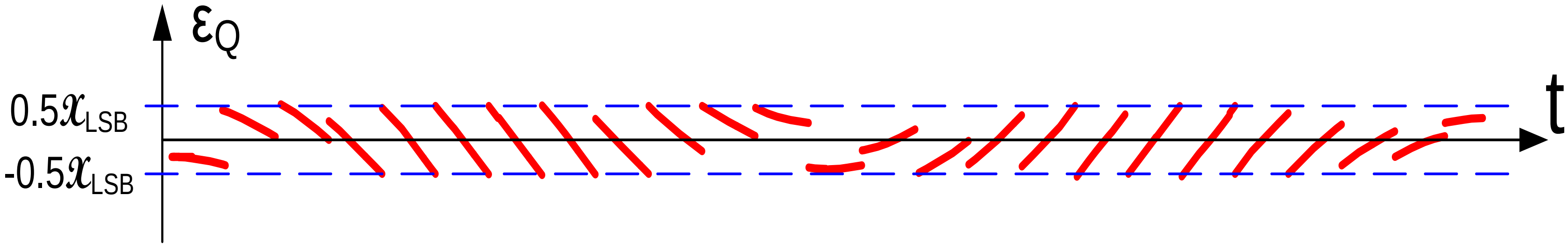
How does the SNR change if the input is a sinusoid that goes from 0 to $\mathcal{X}_{REF}$ centered at $\mathcal{X}_{REF}/2$?



- Appears to be highly uncorrelated with input even though deterministic
- Mathematical expression for ε_Q very messy
- Excursions exceed X_{LSB} (but will be smaller and bounded by $\pm X_{LSB}/2$ for lower frequency signal/frequency clock ratios)
- For lower frequency inputs and higher resolution, at any time, errors are approximately uniformly distributed between $-X_{LSB}/2$ and $X_{LSB}/2$
- Analytical form for ε_{QRMS} essentially impossible to obtain from $\varepsilon_Q(t)$

Quantization Noise in ADC

How does the SNR change if the input is a sinusoid that goes from 0 to $\mathcal{X}_{REF}$ centered at $\mathcal{X}_{REF}/2$?



For low f_{SIG}/f_{CL} ratios, bounded by ±XLB and at any point in time, behaves almost as if a uniformly distributed random variable

$$\varepsilon_Q \sim U[-0.5X_{LSB},\ 0.5X_{LSB}]$$

Quantization Noise in ADC

Recall:

If the random variable f is uniformly distributed in the interval [A,B]
f : U[A,B]   then the mean and standard deviation of f are given by

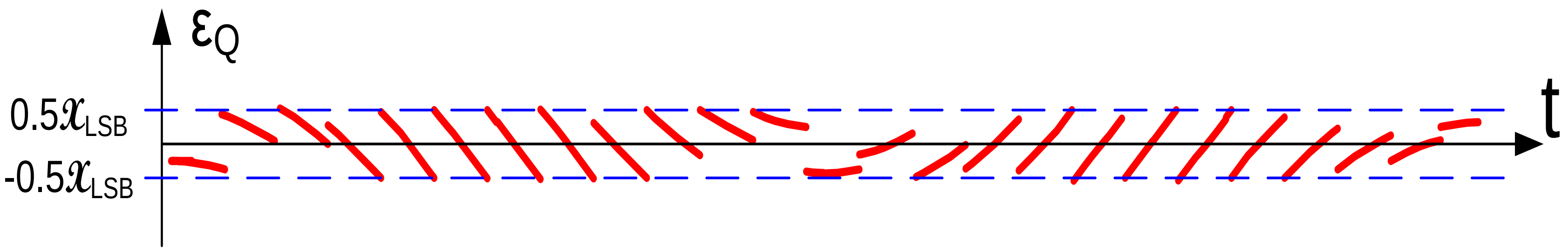
$$\mu_f = \frac{A+B}{2} \qquad \sigma_f = \frac{B-A}{\sqrt{12}}$$

Theorem: If n(t) is a random process, then for large T,

$$V_{RMS} = \sqrt{\frac{1}{T}\int_{t_1}^{t_1+T} n^2(t)\,dt} = \sqrt{\sigma_n^2 + \mu_n^2}$$

Quantization Noise in ADC

How does the SNR change if the input is a sinusoid that goes from 0 to $\mathcal{X}_{REF}$ centered at $\mathcal{X}_{REF}/2$?



$$\varepsilon_Q \sim U[-0.5X_{LSB},\ 0.5X_{LSB}]$$

$$\mu_{\varepsilon_Q} = \frac{A+B}{2} = 0 \qquad \sigma_f = \frac{B-A}{\sqrt{12}} = \frac{X_{LSB}}{\sqrt{12}}$$

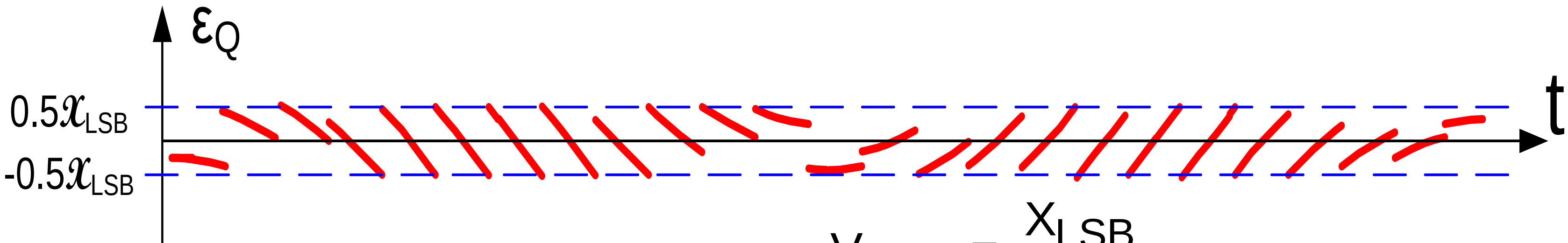
$$V_{RMS} = \sqrt{\frac{1}{T}\int_{t_1}^{t_1+T} n^2(t)\,dt} = \sqrt{\sigma_n^2 + \mu_n^2}$$

$$V_{RMS} = \sigma_{\varepsilon_Q} = \frac{X_{LSB}}{\sqrt{12}}$$

Note this is the same RMS noise that was present with a triangular input

Quantization Noise in ADC

How does the SNR change if the input is a sinusoid that goes from 0 to $\mathcal{X}_{REF}$ centered at $\mathcal{X}_{REF}/2$?



$$V_{RMS} = \frac{X_{LSB}}{\sqrt{12}}$$

But $\quad V_{INRMS} = \left(\frac{X_{REF}}{2}\right)\frac{1}{\sqrt{2}}$

Thus obtain

$$SNR = \frac{\dfrac{X_{REF}}{2\sqrt{2}}}{\dfrac{X_{LSB}}{\sqrt{12}}} = 2^n\sqrt{\frac{3}{2}}$$

Finally, in db,

$$SNR_{dB} = 20\log\left(2^n\sqrt{\frac{3}{2}}\right) = 6.02\,n + 1.76$$

ENOB based upon Quantization Noise

$$SNR = 6.02\ n + 1.76$$

Solving for n, obtain

$$ENOB = \frac{SNR_{dB} - 1.76}{6.02}$$

Note: could have used the SNR_{dB} for a triangle input and would have obtained the expression

$$ENOB = \frac{SNR_{dB}}{6.02}$$

But the earlier expression is more widely used when specifying the ENOB based upon the noise level present in a data converter

ENOB based upon Quantization Noise

For very low resolution levels, the assumption that the quantization noise is uncorrelated with the signal is not valid and the ENOB expression will cause a modest error

from van de Plassche (p13)

$$SNR_{corr} \cong \left(2^n - 2 + \frac{4}{\pi}\right)\sqrt{\frac{3}{2}}$$

Res (n)	SNR_{corr}	SNR
1	3.86	7.78
2	12.06	13.8
3	19.0	19.82
4	25.44	25.84
5	31.66	31.86
6	37.79	37.88
8	49.90	49.92
10	61.95	61.96

SNR = 6.02 n +1.76

Table values in dB

Almost no difference for n ≥ 3

Stay Safe and Stay Healthy !

End of Lecture 3