ELC 4351: Digital Signal Processing

Liang Dong

Department of Electrical and Computer Engineering
Baylor University

liang_dong@baylor.edu

January 19, 2017
Errors in Computing Systems:

- Numbers are represented by a finite number of bits. The resulting errors are called the finite-wordlength or finite-precision effects.
Quantization Errors

Errors in Computing Systems:

- Numbers are represented by a finite number of bits. The resulting errors are called the finite-wordlength or finite-precision effects.

- Quantization errors:
  - Signal quantization
  - Coefficient quantization

Quantization

Quantization Errors

Signal Quantization

Signal to Quantization Noise Ratio

Coefficient Quantization

Roundoff Noise

Overflow

Scaling of Signals
Errors in Computing Systems:

- Numbers are represented by a finite number of bits. The resulting errors are called the finite-wordlength or finite-precision effects.

- Quantization errors:
  - Signal quantization
  - Coefficient quantization

- Arithmetic errors:
  - Roundoff or truncation
  - Overflow
Analog signal $x(t) \Rightarrow \text{ADC} \Rightarrow$ digital signal $x[n]$. 
Analog signal $x(t) \Rightarrow \text{ADC} \Rightarrow$ digital signal $x[n]$.

First, $x(t)$ is sampled and becomes a discrete-time signal $x(nT)$. 
Analog signal $x(t) \Rightarrow$ ADC $\Rightarrow$ digital signal $x[n]$.

First, $x(t)$ is sampled and becomes a discrete-time signal $x(nT)$.

Then, $x(nT)$ is encoded using $B$ bits and becomes a digital signal $x[n]$. 
Suppose that $-1 \leq x[n] < 1$. 

Dynamic range = 2. $B$ bits represent a sample, the number of quantization levels is $2^B$. The quantization step (resolution): $\Delta = 2^{\frac{1}{2^B}} = 2^{\frac{1}{2^B} + 1}$. 

[Notes on signal quantization, coefficient quantization, roundoff noise, overflow, scaling of signals]
Suppose that $-1 \leq x[n] < 1$.

- Dynamic range $= 2$. 
Suppose that \(-1 \leq x[n] < 1\).

- Dynamic range = 2.

- \(B\) bits represent a sample, the number of quantization levels is \(2^B\).
Suppose that $-1 \leq x[n] < 1$.

- Dynamic range $= 2$.

- $B$ bits represent a sample, the number of quantization levels is $2^B$.

- The quantization step (resolution): $\Delta = \frac{2}{2^B} = 2^{-B+1}$. 
A 3-bit ADC:
Quantization error/noise: $e(n) = x(n) - x(nT)$. 

Rounding:

$|e(n)| \leq \frac{\Delta}{2}$. 

The quantization noise depends on the quantization step. More bits $\Rightarrow$ smaller quantization step $\Rightarrow$ lower quantization noise.
Rounding Error

- Quantization error/noise: \( e(n) = x(n) - x(nT) \).
- Rounding: \( |e(n)| \leq \Delta/2 \).
Rounding Error

- Quantization error/noise: $e(n) = x(n) - x(nT)$.

- Rounding: $|e(n)| \leq \Delta/2$.

- The quantization noise depends on the quantization step.
Rounding Error

- Quantization error/noise: \( e(n) = x(n) - x(nT) \).

- Rounding: \( |e(n)| \leq \Delta / 2 \).

- The quantization noise depends on the quantization step.

- More bits \( \Rightarrow \) smaller quantization step \( \Rightarrow \) lower quantization noise.
The nonlinear operation of quantizer: $x(n) = Q[x(nT)]$

Linear operation: $x(n) = Q[x(nT)] = x(nT) + e(n)$
Common Assumptions

- Assume that the quantization error $e(n)$ is uncorrelated with $x(n)$. 
Common Assumptions

- Assume that the quantization error $e(n)$ is uncorrelated with $x(n)$.

- Assume $e(n)$ is a random variable uniformly distributed in the interval $[-\Delta/2, \Delta/2]$. 

Assume that the quantization error $e(n)$ is uncorrelated with $x(n)$. Assume $e(n)$ is a random variable uniformly distributed in the interval $[-\Delta/2, \Delta/2]$. Large wordlength $B$ leads to small quantization error $\sigma^2_e = \frac{\Delta^2}{12} = \frac{1}{2}B^2$. 

Overflow
Common Assumptions

- Assume that the quantization error $e(n)$ is uncorrelated with $x(n)$.

- Assume $e(n)$ is a random variable uniformly distributed in the interval $[-\Delta/2, \Delta/2]$.

- Therefore, $E[e(n)] = (-\Delta/2 + \Delta/2)/2 = 0$;

  and variance: $\sigma_e^2 = \frac{\Delta^2}{12} = \frac{2^{-2B}}{3}$.

Large wordlength $B$ leads to small quantization error $\sigma_e^2$. 
Signal to Quantization Noise Ratio

- \( \text{SNR} = 10 \log_{10}(\sigma_x^2/\sigma_e^2) \).
Signal to Quantization Noise Ratio

- \( \text{SNR} = 10 \log_{10}(\sigma_x^2/\sigma_e^2) \).

- With \( \sigma_e^2 = 2^{-2B}/3 \), we have

\[
\text{SNR} = 10 \log_{10}(3 \times 2^{2B} \sigma_x^2) \\
= 10 \log_{10} 3 + 20B \log_{10} 2 + 10 \log_{10} \sigma_x^2 \\
= 4.77 + 6.02B + 10 \log_{10} \sigma_x^2
\]

For each additional bit, the ADC provides about 6-dB gain. SNR is proportional to \( \sigma_x^2 \). Keep signal power as large as possible.
Signal to Quantization Noise Ratio

- \[ \text{SNR} = 10 \log_{10}(\sigma_x^2/\sigma_e^2). \]
- With \( \sigma_e^2 = 2^{-2B}/3 \), we have
  \[
  \text{SNR} = 10 \log_{10}(3 \times 2^{2B} \sigma_x^2) \\
  = 10 \log_{10} 3 + 20B \log_{10} 2 + 10 \log_{10} \sigma_x^2 \\
  = 4.77 + 6.02B + 10 \log_{10} \sigma_x^2
  \]
- For each additional bit, the ADC provides about 6-dB gain.
Signal to Quantization Noise Ratio

- SNR = $10 \log_{10}(\sigma_x^2/\sigma_e^2)$.

- With $\sigma_e^2 = 2^{-2B}/3$, we have
  \[
  \text{SNR} = 10 \log_{10}(3 \times 2^{2B} \sigma_x^2) \\
  = 10 \log_{10} 3 + 20B \log_{10} 2 + 10 \log_{10} \sigma_x^2 \\
  = 4.77 + 6.02B + 10 \log_{10} \sigma_x^2
  \]

- For each additional bit, the ADC provides about 6-dB gain.

- SNR is proportional to $\sigma_x^2$. Keep signal power as large as possible.
The filter coefficients $b_n, a_m$ are quantized for a given fixed-point processor.
The filter coefficients $b_n, a_m$ are quantized for a given fixed-point processor.

Coefficient quantization can cause serious problems if the poles of designed IIR filters are too close to the unit circle.
The filter coefficients $b_n$, $a_m$ are quantized for a given fixed-point processor.

Coefficient quantization can cause serious problems if the poles of designed IIR filters are too close to the unit circle.

This is because those poles may move outside the unit circle due to coefficient quantization, resulting in an unstable implementation.
Roundoff Noise

\[ y(n) = \alpha x(n) \]
Roundoff Noise

\[ y(n) = \alpha x(n) \]

- \( x(n) \) and \( \alpha \) are \( B \)-bit, the product \( y(n) \) will be \( 2B \)-bit.
Roundoff Noise

\[ y(n) = \alpha x(n) \]

- \( x(n) \) and \( \alpha \) are \( B \)-bit, the product \( y(n) \) will be \( 2B \)-bit.
- Usually, the result will be stored in \( B \)-bit memory.
Roundoff Noise

- $y(n) = \alpha x(n)$

- $x(n)$ and $\alpha$ are $B$-bit, the product $y(n)$ will be $2B$-bit.

- Usually, the result will be stored in $B$-bit memory.

- Truncation or rounding brings the roundoff noise.
Roundoff Noise

- $y(n) = \alpha x(n)$
- $x(n)$ and $\alpha$ are $B$-bit, the product $y(n)$ will be $2B$-bit.
- Usually, the result will be stored in $B$-bit memory.
- Truncation or rounding brings the roundoff noise.
- $y(n) = Q[\alpha x(n)] = \alpha x(n) + e(n)$
Roundoff Noise

- \( y(n) = \alpha x(n) \)

- \( x(n) \) and \( \alpha \) are \( B \)-bit, the product \( y(n) \) will be \( 2B \)-bit.

- Usually, the result will be stored in \( B \)-bit memory.

- Truncation or rounding brings the roundoff noise.

- \( y(n) = Q[\alpha x(n)] = \alpha x(n) + e(n) \)

- Is this noise larger?
When the dynamic range of signals is fixed, the result of an arithmetic addition may exceed the capacity of the register.
Overflow

- When the dynamic range of signals is fixed, the result of an arithmetic addition may exceed the capacity of the register.

- This overflow results in severe distortion of the signal output.
When the dynamic range of signals is fixed, the result of an arithmetic addition may exceed the capacity of the register.

This overflow results in severe distortion of the signal output.

We need saturation algorithm or proper scaling.
Saturation Algorithm

- Saturation arithmetic prevents overflow by keeping the result at a maximum value.

- Saturation algorithm is a nonlinear operation that clips the desired waveform.

\[
y = \begin{cases} 
1 - 2^{-M}, & x \geq 1 - 2^{-M} \\
1 - 2^{-M}, & -1 \leq x < 1 \\
1, & x < -1
\end{cases}
\]
An effective technique in preventing overflow is by scaling down the signal.
An effective technique in preventing overflow is by scaling down the signal.

If the signal $x(n)$ is scaled by $\beta$, the corresponding signal variance changes to $\beta^2 \sigma^2_x$. 
Scaling of Signals

- \[ \text{SNR} = 10 \log_{10}(\beta^2 \frac{\sigma_x^2}{\sigma_e^2}) \]
  
  \[ = 4.77 + 6.02B + 10 \log_{10} \sigma_x^2 + 20 \log_{10} \beta \]
Scaling of Signals

- \[ \text{SNR} = 10 \log_{10} \left( \beta^2 \frac{\sigma_x^2}{\sigma_e^2} \right) \]
  \[ = 4.77 + 6.02B + 10 \log_{10} \sigma_x^2 + 20 \log_{10} \beta \]

- For down scaling, \( \beta < 1. \)
Scaling of Signals

\[ \text{SNR} = 10 \log_{10} \left( \frac{\beta^2 \sigma_x^2}{\sigma_e^2} \right) = 4.77 + 6.02 B + 10 \log_{10} \sigma_x^2 + 20 \log_{10} \beta \]

- For down scaling, \( \beta < 1 \).
- The term \( 20 \log_{10} \beta \) is negative, and the SNR reduces.
Scaling of Signals

- SNR = 10 log_{10}(\beta^2 \sigma_x^2 / \sigma_e^2)
  = 4.77 + 6.02B + 10 \log_{10} \sigma_x^2 + 20 \log_{10} \beta

- For down scaling, \( \beta < 1 \).

- The term 20 \log_{10} \beta is negative, and the SNR reduces.

- For example, when \( \beta = 0.5 \), 20 \log_{10} \beta = -6.02 \text{ dB}, thus reducing the SNR of the input signal by about 6 dB.
Scaling of Signals

- SNR = 10 log_{10} \left( \beta^2 \sigma_x^2 / \sigma_e^2 \right)
  = 4.77 + 6.02B + 10 \log_{10} \sigma_x^2 + 20 \log_{10} \beta

- For down scaling, $\beta < 1$.

- The term $20 \log_{10} \beta$ is negative, and the SNR reduces.

- For example, when $\beta = 0.5$, $20 \log_{10} \beta = -6.02$ dB, thus reducing the SNR of the input signal by about 6 dB.

- This is equivalent to losing 1 bit in representing the signal. Why?