Outline

• A closer look at quantization
  • Sidebar: deciBels (dB)
• Representing negative integers in binary number systems
• Digitizing other forms of information:
  • Text
  • Images and Video
Reminder: The role of ADC and DAC

- Analog-to-Digital conversion enables the use of digital processing and transmission technologies.
- ADC involves:
  - Converting from continuous-time to discrete-time: **Sampling**.
  - Converting from continuous amplitudes to discrete amplitudes levels: **Quantization**.
- The reverse process is called Digital-to-Analog conversion.

A CLOSER LOOK AT QUANTIZATION
Quantization Introduces Errors

- The Sampling Theorem assures us that the process of sampling a continuous-time signal can be reversed without error.
  - As long as the sampling rate is high enough.
- In contrast, quantization introduces distortion that is not reversible.
  - The rounding in the quantization process leads to errors that cannot be reversed.
  - This error is called quantization noise.
  - We will quantify how quantization noise is related to the resolution (number of bits) of an ADC.

Experiment Design

1. Take a simple analog signal,
   - Expressed as a mathematical formula,
2. Quantize samples of that signal
   - To a given resolution (N bits).
3. Compute the quantization error for each sample.
   - I.e., the difference between quantized and unquantized samples.
4. Measure the strength of the quantization noise relative to the strength of the signal.
   - The resulting ratio is called the signal-to-noise ratio (SNR).
5. Observe how the SNR depends on the resolution N.
Experiment Details

- For our experiment, we will use the signal \( x(t) = 5 \cos(2\pi 50 t) \exp(-50 t) \).
- The digital-to-analog conversion will employ:
  - Sampling rate: \( f_s = 1000 \) samples/second.
  - Input range: \( V_{\text{max}} = 5 \).
  - Resolution: \( N \) bits
    - We will vary the resolution \( N \) and study how the quantization noise changes.

Quantization

- For each un-quantized sample \( x[n] = x(nT_s) \), the quantization process involves:
  1. Sign and magnitude:
     - \( \text{sign}[n] = 1 \) if \( x[n] < 0 \), \( \text{sign}[n] = 0 \) otherwise.
     - Magnitude: \( |x[n]| \).
  2. Scaling: the un-quantized magnitudes \( |x[n]| \) are multiplied by \( (2^{(N-1)}-1)/V_{\text{max}} \):
     - Resulting values are between 0 and \( +2^{(N-1)} \).
     - Samples outside this range are clipped to \( +2^{(N-1)} \).
  3. Rounding: The magnitudes \( |x[n]| \) are rounded to the nearest integer.
     - Denote the result as \( m_q[n] \).
     - This step introduces the quantization error.
  4. Unscaling: To allow comparison to the original samples, we reverse the scaling and decomposition into sign and magnitude:
     \[ x_q[n] = (-1)^{\text{sign}[n]} \times m_q[n] \times \frac{V_{\text{max}}}{(2^{(N-1)}-1)} \]
    - The quantized samples \( x_q[n] \) should be close to the original \( x[n] \).
Example

- Assume $x[n] = -3.7$, $V_{\text{max}} = 5$, and $N = 8$.
  1. Sign and magnitude:
     $\text{sign}[n] = 1$, $|x[n]| = 3.7$.
  2. Scaling:
     $|x[n]| \left(2^{(N-1)}-1\right)/V_{\text{max}} = 93.98$.
  3. Rounding:
     Nearest integer is $m_q[n] = 94$.
  4. Unscaling:
     $x_q[n] = (-1)^{\text{sign}[n]} \times m_q[n] \times V_{\text{max}}/(2^{(N-1)}-1) = -3.70079$

Illustration: $N=3$
Quantization Error

- The difference between the original and the quantized samples is called the quantization error:
  \[ e_q[n] = x[n] - x_q[n] \].
- The maximum possible quantization error for a given resolution \( N \) and input range \( V_{\text{max}} \) is:
  \[ \text{max. error} = \frac{1}{2} V_{\text{max}}/(2^{(N-1)}-1). \]

| n  | nT_s | x[n]   | sign(x[n]) | |x[n]| | m_x[n] | x_q[n] | e_q[n] |
|----|------|--------|------------|-----|-----|--------|-------|-------|
| 0  | 0.001| 4.52337833 | 0          | 4.52337833 | 3   | -0.4766217 |
| 2  | 0.002| 3.66019352 | 0          | 3.66019352 | 2   | 0.3298019 |
| 3  | 0.003| 2.52965405 | 0          | 2.52965405 | 2   | -0.3333333 |
| 4  | 0.004| 1.26515287 | 0          | 1.26515287 | 1   | 0.4015138 |
| 5  | 0.005| 0.0001804  | 0          | 0.0001804  | 0   | -0.3333333 |
| 6  | 0.006| -1.1444313  | 1          | 1.14443126 | 1   | -0.6666667 |
| 7  | 0.007| -2.0708415  | 1          | 2.07084145 | 1   | -0.4041748 |
| 8  | 0.008| -2.7113555  | 1          | 2.71135551 | 2   | -0.3333333 |
| 9  | 0.009| -3.0320199  | 1          | 3.03201988 | 2   | -0.3333333 |
| 10 | 0.01 | -3.0326533  | 1          | 3.03265329 | 2   | -0.3333333 |
| 11 | 0.011| -2.7436502  | 1          | 2.74365022 | 2   | -0.3333333 |
| 12 | 0.012| -2.220169   | 1          | 2.22016902 | 1   | -0.6666667 |

Spread sheet: Quantization.xlsx
Quantized and Original Signal

For the chart below: N=3

Quantization Error

For the chart below: N=3
Summarizing Quantization Error

- It is desirable to summarize the effects of quantization in a single metric.
  - Want to capture “strength” of quantization error,
  - Should relate that strength to the strength of the signal.
- Strength of signal is measured by its power.
- Metric for capturing the impact of quantization: signal-to-noise ratio (SNR).
  - SNR is defined as the ratio of signal power to power of quantization error.

Measuring the Power of a Signal

- The power $P$ of a discrete-time signal $x[n]$ with $M$ samples is defined as the average of the squared magnitudes of the samples:
  $$P = \frac{1}{M} \sum_{n=0}^{M-1} |x[n]|^2$$
- To compute the power of a set of samples:
  1. Compute the square $x[n]^2$ for each sample.
  2. Sum all squares.
  3. Divide by the number of samples.
How Does SNR Vary with Resolution?

- To finish our experiment, we will measure the SNR for a number of different resolutions N.
- Specifically, we will vary N from 2 to 12 and record the resulting SNR.
- We will then plot SNR vs N to explore the relationship between the two quantities.
- The above is straightforward using the Excel spreadsheet Quantization.xlsx.

SNR vs Resolution

<table>
<thead>
<tr>
<th>Resolution</th>
<th>SNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2.1</td>
</tr>
<tr>
<td>3</td>
<td>16.4</td>
</tr>
<tr>
<td>4</td>
<td>93.9</td>
</tr>
<tr>
<td>5</td>
<td>478.2</td>
</tr>
<tr>
<td>6</td>
<td>2250.5</td>
</tr>
<tr>
<td>7</td>
<td>11275.5</td>
</tr>
<tr>
<td>8</td>
<td>34888.0</td>
</tr>
<tr>
<td>9</td>
<td>133188.1</td>
</tr>
<tr>
<td>10</td>
<td>786920.1</td>
</tr>
<tr>
<td>11</td>
<td>1908634.2</td>
</tr>
<tr>
<td>12</td>
<td>8913819.4</td>
</tr>
</tbody>
</table>

Relationship is difficult to discern!?
Exponential increase in SNR?
The deciBel

- When observations vary over several orders of magnitude, analysis or interpretation is difficult.
  - On a plot, small values are indistinguishable.
  - Most important aspect may be the order of magnitude.
    - True for exponentially increasing or decreasing data.
- In such cases, it is appropriate to transform data to a logarithmic scale.
  - Engineers use the logarithmic transformation:
    \[ x \text{ in dB} = 10 \log_{10}(x) \]
  - dB stands for deciBel.
- Property: If data exhibit an exponential relationship
  \[ x[n] = A b^n, \]
  then the transformation to dB leads to a linear relationship:
  \[ x[n] \text{ in dB} = 10 \log_{10}(A) + n \log_{10}(b). \]
- A value given dB is transformed back to linear (normal) scale via:
  \[ x = 10^{(x \text{ in dB}/10)} \]

<table>
<thead>
<tr>
<th>Resolution</th>
<th>SNR</th>
<th>SNR in dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2.1</td>
<td>3.3</td>
</tr>
<tr>
<td>3</td>
<td>16.4</td>
<td>12.2</td>
</tr>
<tr>
<td>4</td>
<td>93.9</td>
<td>19.7</td>
</tr>
<tr>
<td>5</td>
<td>478.2</td>
<td>26.8</td>
</tr>
<tr>
<td>6</td>
<td>2250.5</td>
<td>33.5</td>
</tr>
<tr>
<td>7</td>
<td>11275.5</td>
<td>40.5</td>
</tr>
<tr>
<td>8</td>
<td>34888.0</td>
<td>45.4</td>
</tr>
<tr>
<td>9</td>
<td>133188.1</td>
<td>51.2</td>
</tr>
<tr>
<td>10</td>
<td>786920.1</td>
<td>59.0</td>
</tr>
<tr>
<td>11</td>
<td>1908634.2</td>
<td>62.8</td>
</tr>
<tr>
<td>12</td>
<td>8913819.4</td>
<td>69.5</td>
</tr>
</tbody>
</table>

SNR in dB vs Resolution

- Linear relationship is clearly discernible: slope 6dB/bit
  - Implies that SNR increases exponentially like \(4^N\).
Does this Make Sense?

- When one bit of resolution is added,
  - the number of quantization intervals is doubled,
  - and the quantization errors are (approximately) halved.
- The power of the quantization noise is reduced by a factor \(2^2 = 4\).
  - Recall that power involves the square of the magnitudes.
- Since the signal power is independent of resolution, SNR increases by a factor of 4.

REPRESENTING NEGATIVE NUMBERS IN BINARY FORM
Signed Magnitude Representation

- So far, we have represented negative numbers in binary form by:
  - Designating the MSB as the sign bit, and
  - Representing the magnitude using the remaining N-1 bits.
- This representation was used in very early versions of digital processors.
- Problems:
  - Two representations of 0:
    - 0000 and 1000 (i.e., +0 and -0)
  - Addition doesn’t work right:
    - 3+(-2): 011 + 110 = 1001 (that’s -1, not 1).

One’s Complement

- Another way to represent negative numbers is one’s complement.
- In one’s complement,
  - Positive numbers have a 0 in the MSB followed by binary representation of the magnitude in the remaining N-1 bits.
    - Same as signed magnitude.
  - Negative numbers are represented by inverting ALL bits of the corresponding positive number.
    - Example: +3 is represented as 0011 and -3 is represented as 1100.
One’s Complement

- One’s complement is used for various checksum algorithms.
- With one’s complement,
  - There are still two representations of 0:
    - +0: 0000, -0: 1111.
  - Addition can be fixed by adding an “end-around carry”:
    - Example: $3 + (-2) = 1$
      
      \[
      \begin{array}{c|c}
      & 0011 \\
      + & 1101 \\
      \hline
      & 0000 \\
      \end{array}
      \]
      \[
      \begin{array}{c|c}
      & 0001 \\
      \hline
      & 0001 \\
      \end{array}
      \]
    - If a carry occurs, it is added to the result.

Two’s complement

- The most widely used format for representing negative numbers is two’s complement.
- To convert between a positive number and it’s negative:
  - Form the one’s complement, then add 1.
  - Example:
    - 3 is represented as 0011 (same as signed magnitude and one’s complement).
    - The one’s complement is 1100.
    - Adding one yields -3 in two’s complement: 1101.
  - To convert a negative number to it’s positive, one uses the same procedure:
    - Example:
      - -3 in two’s complement: 1101
      - One’s complement: 0010.
      - Add 1: 0011 (which is the representation of +3).
Properties of Two’s Complement

- Only one representation of 0: 0000.
- Largest positive number: $2^{N-1}-1$ (0111 = 7₁₀)
- Most negative number: $-2^N$ (1000 = -8₁₀)
- Addition “just works” if carry is discarded:
  - Example: $3 + (-2) = 1$
    
    
    \[
    \begin{array}{c}
    0011 \\
    1110 \\
    \hline
    \ast 0001
    \end{array}
    \]

Hexadecimal Representation

- Binary notation is tedious for humans.
  - Binary numbers are approximately three times longer than their decimal counterparts.
  - Binary representation of text is 8 (or even 32) times longer than the text itself.
- To reduce the length of binary representations, it is customary
  - to group sequences of bits into groups of four, and
  - to assign each group a corresponding “digit”.

Hexadecimal Table

<table>
<thead>
<tr>
<th>Group of Bits</th>
<th>Digit</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0</td>
</tr>
<tr>
<td>0001</td>
<td>1</td>
</tr>
<tr>
<td>0010</td>
<td>2</td>
</tr>
<tr>
<td>0011</td>
<td>3</td>
</tr>
<tr>
<td>0100</td>
<td>4</td>
</tr>
<tr>
<td>0101</td>
<td>5</td>
</tr>
<tr>
<td>0110</td>
<td>6</td>
</tr>
<tr>
<td>0111</td>
<td>7</td>
</tr>
<tr>
<td>1000</td>
<td>8</td>
</tr>
<tr>
<td>1001</td>
<td>9</td>
</tr>
<tr>
<td>1010</td>
<td>A</td>
</tr>
<tr>
<td>1011</td>
<td>B</td>
</tr>
<tr>
<td>1100</td>
<td>C</td>
</tr>
<tr>
<td>1101</td>
<td>D</td>
</tr>
<tr>
<td>1110</td>
<td>E</td>
</tr>
<tr>
<td>1111</td>
<td>F</td>
</tr>
</tbody>
</table>

Example: $4A_{\text{hex}} = 0100\ 1010_2 = 74_{10}$

Exercise

- Using 4-bit binary representations, express the numbers 4 and -6 in
  - Signed-magnitude form
  - One’s complement
  - Two’s complement
- For one’s complement and two’s complement, add the two numbers.
Other Forms of Information

- The techniques we have discussed to this point are very general and can be applied to any continuous-time signal.
  - Speech is primary example,
  - Any measurement recorded by appropriate sensors:
    - Seismic, current or voltage used by an appliance, speed, fuel consumption rate, ….
- Textual information is fundamentally different:
  - No need to quantize or sample.
  - Must still represent “letters” in binary form.
- Images and video signals are inherently multi-dimensional.
  - Not all that different from signals considered so far.
Representing Text

- Representing textual information is fundamentally simpler than analog signals.
- Written information is already discrete:
  - Basic unit of text: letters (or characters).
- All binary representations of text rely on look-up tables to translate between binary and written representation.

Example: ASCII

- ASCII stands for American Standard Code for Information Interchange.
- In ASCII, 8 bits are used to encode characters that are common in English text (plus a number of control characters).
  - Actually only 7 bits are used – the MSB is always 0.
- ASCII tables explicitly list the correspondence of 128 bit patterns and characters.
A Few Examples

<table>
<thead>
<tr>
<th>Character</th>
<th>Hexadecimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>,</td>
<td>2C</td>
</tr>
<tr>
<td>-</td>
<td>2D</td>
</tr>
<tr>
<td>.</td>
<td>2E</td>
</tr>
<tr>
<td>/</td>
<td>2F</td>
</tr>
<tr>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>1</td>
<td>31</td>
</tr>
<tr>
<td>9</td>
<td>39</td>
</tr>
<tr>
<td>3A</td>
<td>:</td>
</tr>
<tr>
<td>40</td>
<td>@</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Character</th>
<th>Hexadecimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>41</td>
</tr>
<tr>
<td>B</td>
<td>42</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Z</td>
<td>5A</td>
</tr>
<tr>
<td>[</td>
<td>5B</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>.</td>
<td>60</td>
</tr>
<tr>
<td>a</td>
<td>61</td>
</tr>
<tr>
<td>b</td>
<td>62</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>z</td>
<td>2A</td>
</tr>
</tbody>
</table>

Example: Unicode

- In an increasingly multi-lingual world, provisions are needed to support character sets other than those used in the English language.
- Unicode provides a framework for supporting a virtually arbitrary number of character sets.
- To facilitate this, Unicode provides 32 bits for representing each character.
  - This allows for up to 4 Billion characters to be encoded.
  - A prior version of Unicode used 16 bits.
- Unicode bit sequences are written a U-XXXXXXXX where each X is a hexadecimal digit.
  - The ASCII characters are U-00000000 to U-0000007F.
Digitizing Images

- To digitize an image, light is projected onto an array of light-sensitive sensors.
  - The most prevalent transducers used in digital cameras are charge-coupled devices (CCD).
  - These were invented in 1969 at Bell Labs by Boyle and Smith.
    - They received the 2009 Nobel Prize for Physics.
  - The output voltage of each CCD is sampled and A/D converted and then transferred to memory.

Images are Two-Dimensional

- Each pixel in the array produces a sample.
- Since the array is two-dimensional, the resulting signal is best thought of as a two-dimensional signal.
- For processing, storage, and transmission the two-dimensional signal must be serialized (i.e., made one-dimensional).
  - This is accomplished by scanning.
Color

• CCDs transduce the intensity of the incident light.
  • This leads to grey-scale images.
• To create color images, filters for red, blue, and green light are used.
  • Pixels will indicate only how much red, green, or blue light is present at its location.

Video

• Video signals are created by periodic (in time) capture of images.
  • At a periodic rate, an image is taken and the resulting sequence of images is the video signal.
• In addition to the spatial sampling involved in capturing an image, temporal sampling is needed for creating video.
  • Consequently, video signals are three-dimensional.
    • Two spatial and one time dimension.