Experiment# 2
Sampling, Quantization, and Pulse Code Modulation (PCM)

Introduction:
Although a significant portion of communication today is in analog form, it is being replaced rapidly by digital communication. Within the next decade most of communication will become digital, with analog communication playing a minor role.

Today's lab may be viewed as a transition from analog to digital communications; it will consider the first important step in any digital communication system, transforming the source information to a form that is compatible with a digital system. We will treat various aspects of sampling, quantization (both uniform and nonuniform), and pulse code modulation (PCM). Finally, as an application, we will build a complete digital system that deals with a speech signal to investigate the effect of the system parameters on the quality of the reconstructed speech signal.

Prelab
1. How many bits would be required to represent an analog signal with values ranging from (-1) to 1 Volt if the resulting quantized signal is to have a resolution of 0.125 V? (round to the nearest bit)
2. Assuming a maximum source coding data rate of 50 kbits/sec, what is the maximum signal bandwidth we can transmit using PCM with the number of bits found in question 1? What is the corresponding Nyquist rate?
3. For the results in questions 1 and 2, compute the Signal to Quantization Noise Ratio (SQNR) of the system assuming the message signal has a peak to average power ratio \( \frac{V_{\text{max}}}{P_{\text{av}}} \) of 3 dB and 14 dB.
4. Derive the the SQNR for a sine wave in term of the number of levels.
5. Use the simulink to draw the characteristic of a quantizer having the following I/O relation:
   \[
   y = \begin{cases} 
   0.75 & x \geq 0.5 \\
   0.25 & 0 \leq x < 0.5 \\
   -0.25 & -0.5 \leq x < 0 \\
   -0.75 & x < -0.5 
   \end{cases}
   \]
6. For the system shown in page 10, calculate theoretically:
   a. the first 5 samples
   b. quantized samples
   c. PCM code words
   d. the decoded quantized samples for the PCM
7. Repeat the above question if the system having a μ-law compandor (compressor & expander) with \( \mu = 255 \).
**Sampling process:**

The sampling process is usually described in the time domain. It is an operation that is basic to digital signal processing and digital communication. Through the use of the sampling process, an analog signal is converted into corresponding sequence of samples that is usually spaced uniformly in time i.e. discrete in time. Clearly, for such a procedure to have a practical utility, it is necessary that we choose the sampling rate properly so that the sequence of samples uniquely defines the original analog signal. The sampling theorem is the basis for determining the proper sampling rate for a given band-limited signal. It states that,

*A band-limited signal of finite energy, which has no frequency component higher than \( W \) hertz, is completely described by specifying the values of the signal at instants of time separated by \( 1/2W \) seconds (i.e. at a rate of \( 2W \) samples per second).*

The process of reconstructing a continuous-time signal from its samples is also known as interpolation. In which we pass the sampled signal through an ideal low-pass filter of bandwidth \( W \) Hz. As you may note, the use of a sampling rate higher than the Nyquist rate has a beneficial effect of easing the design of the reconstruction filter used to recover the original signal from its sampled version

**Time-Division Multiplexing:**

The sampling theorem provides the basis for transmitting the information contained in a band-limited message signal as a sequence of samples. An important feature of the sampling process is conservation in time. That is, the transmission of the message samples engages the communication channel for only a fraction of the sampling interval on a periodic basis, and in this way some of the time interval between adjacent samples is cleared for use by other independent message sources on a time-shared basis. We thereby obtain a time-division multiplex (TDM) system, which enable the joint utilization of a common communication channel by allowing all signals to share the transmission link, with each signal connected to the link for only a short time.

The rate of the commutator (electronic switching circuit) at the transmitter side also obeys the sampling theorem:

\[ f_s = N \times 2 \times W, \]  
where \( N \) is the number of messages and \( W = \max(W_i), i=1,2,\ldots N \).

It is clear that the use of TDM introduces a bandwidth expansion factor \( N \), because the scheme must squeeze \( N \) samples derived from \( N \) independent message sources into a time slot equal to one sampling interval.

At the receiving end of the system, the received signal is applied to a pulse demodulator which consists of a decommutator and a LPF. The decommutator is in synchronization with the commutator in the transmitter.

**Quantization:**

A continuous time signal, such as voice, has a continuous range of amplitudes and therefore its samples have a continuous amplitude range i.e. they are only discrete in time not in amplitude. In other words, within the finite amplitude range of the signal, we find an infinite number of amplitude levels. It is not necessary in fact to transmit the exact amplitude of the samples. Any human sense (the ear or the eye), as ultimate receiver, can detect only finite intensity differences. This means that the original continuous time signal may be approximated by a signal constructed of discrete amplitudes selected on a minimum error basis from an available set. Clearly, if we assign the discrete amplitude levels with sufficiently close spacing we may take the approximated signal practically indistinguishable from the original continuous signal.

Amplitude quantization is defined as the process of transforming the sample amplitude \( m(nT_s) \) of a message signal \( m(t) \) at time \( t=nT_s \) into a discrete amplitude \( v(nT_s) \) taken from a finite set of possible amplitudes.
The existence of a finite number of discrete amplitude levels is a basic condition of pulse code modulation. So Quantization is an important stage in forming the PCM signal where the output of the sampling process is quantized to provide a new representation that is discrete in both time and amplitude.

Quantization can be of a uniform or nonuniform type. In a uniform quantizer, the representation levels are uniformly spaced; otherwise, the quantizer is nonuniform. In system that uses uniform quantizer, the quantization noise is the same for all signal magnitude. Therefore, in uniform quantization, the SNR is worse at low level signals than for high level signals. In telephone systems, it was found that for most voice communication channel, very low speech volumes predominate; 50% of the time, the voltage characterizing detected speech energy is less than 1/4 of the rms voltage. Large amplitude values are relatively rare; only 15% of the time does the voltage exceeds the rms value. Also as you may have noticed that the quantization noise depends on the step size, so a uniform quantizer would be wasteful for speech signal.

Nonuniform quantization, in which the step size increases as the separation from the origin of the input-output amplitude increases, can provide fine quantization to the weak signals and coarse quantization of the strong signals. In other words, the weak passages, which need more protection, are favored at the expense of the loud passages. Thus in the case of nonuniform quantization, quantization noise can be made proportional to signal size. The effect is to improve the overall SNR by reducing the noise for the predominant weak signals, at the expense of an increase in noise for the rarely occurring signals.

The use of non-uniform quantizer is equivalent to passing the baseband signal through a compressor and then applying the compressed signal to a uniform quantizer.

Quantization and Compression
Quantization is sometimes used for compression. As an example, suppose we have a digital image which is represented by 8 different gray levels: [0 31 63 95 159 191 223 255]. To directly store each of the image values, we need at least 8-bits for each pixel since the values range from 0 to 255. However, since the image only takes on 8 different values, we can assign a different 3-bit integer (a code) to represent each pixel: [000 001 ... 111]. Then, instead of storing the actual gray levels, we can store the 3-bit code for each pixel. A look-up table, possibly stored at the beginning of the file, would be used to decode the image. This lowers the cost of an image considerably: less hard drive space is needed, and less bandwidth is required to transmit the image (i.e. it downloads quicker). In practice, there are much more sophisticated methods of quantizing images which rely on quantization.

Pulse Code Modulation
Pulse Code Modulation (hereinafter referred to as PCM) is a sampled modulation similar to Pulse Amplitude Modulation. Since PCM encodes a message into bits of 1’s and 0’s, it is often referred to as a source code. PCM does not yield waveforms that vary linearly with the message however.
Nyquist criteria apply to PCM since it is obtained through sampling. That means that the sampling frequency must be at least twice the highest frequency in the message.

PCM offers advantages over other modulation methods in its resistance to noise and its ability to be processed digitally. PCM is particularly resistant to noise added once it has been modulated in the communication channel. PCM can be processed entirely in the digital domain, allowing any desired signal alterations to be performed that would otherwise be impossible in the analog domain.

An Analog to Digital or A/D converter is used to convert the continuous message signal into a series of digital numbers, with each of the numbers representing a level of the quantized message signal. This stream of digital number is the PCM signal. Once the quantized signal is in the digital domain, it is easy to perform any signal processing desired by simply performing the appropriate mathematical operation.

A D/A converter performs an inverse operation to that of the A/D converter, changing digital numbers to analog voltages. The A/D and D/A converters must match with respect to word size, sampling rate, and mapping in order for the PCM signal to be properly demodulated.
Laboratory Assignment

1. Image Quantization

Download the file 'fountainbw.tif' from the W:\lab1utilities\statistics. Place it in the present working directory or in a directory on the path. The image 'fountainbw.tif' is an 8-bit grayscale image. We will investigate what happens when we quantize it to smaller numbers of bits/pixel.

a) Load it into Matlab and display it using the following sequence of commands.

```matlab
y = imread('fountainbw.tif');
image(y);
colormap(gray(256));
axis('image');
```

b) The image array will initially be of type uint8, so you will need to convert the image matrix to type double before performing any computation. Use the command

```matlab
z=double(y)
```

c) Uniform quantizer implementation:

There is an easy way to uniformly quantize a signal. Let

\[ \Delta = \frac{Max(X) - Min(X)}{N - 1} \]

where X is the signal to be quantized, and N is the number of quantization levels. To force the data to have a uniform quantization step of \( \Delta \),

- Subtract Min(X) from the data and divide the result by \( \Delta \).
- Round the data to the nearest integer
- Multiply the rounded data by \( \Delta \) and add Min(X) to convert the data back to its original scale.

d) Write a Matlab function \( Y = Uquant(X,N) \) which will uniformly quantize an input array X (either a vector or a matrix) to N discrete levels.

e) Use this function to quantize the fountain image to 7 b/pel, 6, 5, 4, 3, 2, 1 b/pel, and observe the output images. Print hard copies of only the 7, 4, 2, and 1 b/pel images, as well as the original.

INLAB REPORT:

1. Describe the artifacts (errors) that appear in the image as the number of bits is lowered?
2. Note the number of b/pel at which the image quality noticeably deteriorates.
3. Hand in the printouts of the above four quantized images and the original.
4. Compare each of these four quantized images to the original.
2. Audio Quantization

If an audio signal is to be coded, either for compression or for digital transmission, it must undergo some form of quantization. Most often, a general technique known as vector quantization is employed for this task, but this technique must be tailored to the specific application so it will not be addressed here. In this assignment, we will observe the effect of uniformly quantizing the samples of two audio signals.

a) Download the file 'speech.au' and 'music.au' from the W:\lab\utilities\statistics. Place it in the present working directory or in a directory on the path.
b) Use your Uquant function to quantize each of these signals to 7, 4, 2 and 1 bits/sample.
c) Listen to the original and quantized signals and answer the following questions:
   - For each signal, describe the change in quality as the number of b/sample is reduced?
   - For each signal, is there a point at which the signal quality deteriorates drastically?
   - At what point (if any) does it become incomprehensible?
   - Which signal's quality deteriorates faster as the number of levels decreases?
   - Do you think 4 b/sample is acceptable for telephone systems? ... 2 b/sample?
d) Use subplot to plot in the same figure, the four quantized speech signals over the index range 7201:7400.
e) Generate a similar figure for the music signal, using the same indices. Make sure to use orient tall before printing these out.

INLAB REPORT:
Hand in answers to the above questions, and the two Matlab figures.

3. Error Analysis

As we have clearly observed, quantization produces errors in a signal. The most effective methods of the analysis of the error signal turn out to be probabilistic. In order to apply these methods, however, one needs to have a clear understanding of the error signal's statistical properties. For example, can we assume that the error signal is white noise? Can we assume that it is uncorrelated with the quantized signal? As you will see in this exercise, both of these are good assumptions if the quantization intervals are small compared with sample-to-sample variations in the signal.

If the original signal is X, and the quantized signal is Y, the error signal is defined by the following:

\[ E = Y - X \]

When the spacing, \( \Delta \), between quantization levels is sufficiently small, a common statistical model for the error is a uniform distribution from \(-\Delta/2\) to \(\Delta/2\).

a) Compute the error signal for the quantized speech for 7, 4, 2 and 1 b/sample.
b) Use the command hist(E,20) to generate a 20-bin histogram for each of the four error signals. Use subplot to place the four histograms in the same figure.
1. Hand in the histogram figure.
2. How does the number of quantization levels seem to affect the shape of the distribution?
3. Explain why the error histograms you obtain might not be uniform?

4. **Signal to Noise Ratio**

One way to measure the quality of a quantized signal is by the Power Signal-to-Noise Ratio (PSNR). This is defined by the ratio of the power in the quantized speech to power in the noise.

\[
PSNR = \frac{P_y}{P_E}
\]

In this expression, the noise is the error signal \( E \). Generally, this means that a higher PSNR implies a less noisy signal.

From previous labs we know the power of a sampled signal, \( x(n) \), is defined by

\[
P_x = \frac{1}{L} \sum_{n=1}^{L} x^2(n)
\]

where \( L \) is the length of \( x(n) \). Compute the PSNR for the four quantized speech signals from the previous section.

In evaluating quantization (or compression) algorithms, a graph called a "rate-distortion curve" is often used. This curve plots signal distortion vs. bit rate. Here, we can measure the distortion by \( \frac{1}{PSNR} \), and determine the bit rate from the number of quantization levels and sampling rate. For example, if the sampling rate is 8000 samples/sec, and we are using 7 bits/sample, the bit rate is 56 kilobits/sec (kbps).

- **Assuming that the speech is sampled at 8kHz:**

Plot the rate distortion curve using \( \frac{1}{PSNR} \) as the measure of distortion. Generate this curve by computing the PSNR for 7, 6, 5,..., 1 bits/sample. Make sure the axes of the graph are in terms of distortion and bit rate.

**INLAB REPORT:**
Hand in a list of the 4 PSNR values, and the rate-distortion curve.
Experiment# 3  
Sampling, Quantization, and PCM using Simulink

In this experiment we will use simulink to generate a whole PCM waveform (Transmitter, and receiver).

- we will use the blocks below:

![Simulink Block Diagram]

**Prelab:**

Try to be familiar with the above blocks

**Laboratory Assignment**

Part (1): Sampling

Sampling Blocks:

![Sampling Block Diagram]

Blocks parameters:

**(a) Pulse Generator:**

- Pulse type: Time based
- Time: Use simulation time
- Amplitude: 1
- Period: 2
- Pulse width(% of period): 50, see the effect of varying the pulse width by taking the pulse width to be 10, 20, 30, 40, but take the figure of the output only for the case of 50 and comment about previous cases(10, … 40) without figures.
- Phase delay: 0

**(b) Signal Generator:**

- Waveform: Sine
- Amplitude: 1
- Frequency: 1 Hz
Part (2) Quantization Block:
Quantization Blocks:

Blocks parameters:
- a) Signal generator parameters as above
- b) Scalar Quantizer:
  - Quantization partition (=value of quantization levels): [-0.75 -0.25 0.25 0.75]
  - Quantization codebook: [-0.825 -0.5 0.5 0.825]
  - I/P signal vector length: 1

  Sample time: 0.1, see the effect of varying the sampling time by taking the sampling time be 0.001, 0.05, 0.2, 0.3, 0.5, but take the figure of the output only for the case of 0.1 and comment about previous cases(0.001, ... 0.5) without figures.

Part (3): Quantization after sampling

The parameters is as above
Part (4): Quantization & Encoding Block:

Useful Notes:
1) The block called "Integer to Bit Converter" represents the encoding process.
2) The block called "Scalar Quantizer" represents the quantization process.
3) We use the block called "Simout" to transform from simulink to workspace, so we can deal with the parameters we want.
4) We can't deal with simout in work space unless it was transformed from structure to array, which can be done by double click on the block called simout then transform structure to array.
5) We can find the root mean squared value (RMSV) of the quantization error from the output of the simout block:

Part (5): PCM Block(sampling, quantization, encoding)
Transmitter & Receiver:
1- Construct the shown model in figure below:
The I/P parameters are:

(a) **Pulse Generator:**
- Pulse type: Time based
- Time: Use simulation time
- Amplitude: 1
- Period: 0.01
- Pulse width (% of period): 1
- Phase delay: 0
Choose the bottom check box.

(b) **Signal Generator:**
- Waveform: Sine
- Amplitude: 2
- Frequency: 5 Hz
Choose the bottom check box.

(c) **Scalar Quantizer:**
- Quantization partition (=value of quantization levels): [-.5 0 .5]
- Quantization codebook: [-0.75 -0.25 0.25 0.75]
- I/P signal vector length: 1
- Sample time: 0.01

(c) **Integer to Bit Converter:**
- Number of bits per integer: 4

(d) **Bit to Integer Converter:**
- Number of bits per integer: 4

(e) **Quantizer decode:**
- Quantization codebook: [-0.825 -0.5 0 .5 0.825]
If the filtered signal (scope 3) is not the same as the original signal change fc till you get the desired signal (try and error method).

* keep in mind that the filtered signal is affected by the following factors:
  a) Run time.
  b) Frequency of original signal.
  c) Cutoff frequency fc.