College of Engineering Trivandrum

Department of Electronics and Communication Engineering

ECL332 Communication Engineering Laboratory

Experiment No. 2

Pulse Code Modulation

Name: _____ Date: _____

Roll No.: _____ Batch: _____

Learning Objectives

After completing this lab, you will be able to:

- Explain PCM as a **source coding + formatting** pipeline for real-world data (image).
- Perform **uniform quantization** with different levels $L = 2^b$ and generate a **PCM bitstream**.
- Implement **binary encoder/decoder** and reconstruct the image from the received bits.
- Model a simple digital channel using **bit errors (BSC)** and **AWGN with BPSK** and compare their effects.
- Compute **MSE**, **SNR**, and **PSNR**, and plot quality vs **bits per pixel** b.
- Make an engineering choice of b under a **bit-rate constraint**.

Application Context (Scenario)

You are building a basic "image telemetry" link (e.g., sending a grayscale image from a drone / CCTV / medical device). You must choose the **bit depth** b to balance:

- **Quality**: higher b reduces quantization distortion.
- **Bit-rate**: higher b increases the number of transmitted bits.
- **Channel robustness**: noisy links introduce bit errors that can severely degrade reconstructed images.

Aim

To implement an end-to-end PCM system for a grayscale image: **encode** (quantize + binary), transmit through a simulated **digital channel**, **decode** and **reconstruct**, and evaluate the performance for varying b.

Resources

- Use Python for Programming
- The image is available in https://jinujayachandran.github.io/ay2025-26_ecl332_commlab/.

Theoretical Background

1) What PCM does (and what it does NOT do)

PCM converts **amplitude samples** of a signal into a **binary sequence**. For an image, each pixel value is treated like a sample of a 2D signal.

PCM includes:

- **Sampling**: selecting discrete values. (For digital images, pixels are already sampled.)
- **Quantization**: mapping a continuous-amplitude value to a finite set of levels.
- **Encoding**: representing quantized indices in binary.

PCM does NOT include: compression like JPEG/PNG transforms (DCT/wavelets, entropy coding). This lab focuses on the fundamental digitization step and its trade-offs.

2) Signal model for an image

Let the grayscale image be a matrix $X \in \mathbb{R}^{M \times N}$ with pixel values 0–255 (8-bit). Normalize to:

$$x[m,n] = \frac{X[m,n]}{255}, \qquad 0 \le x[m,n] \le 1$$

Vectorize the image for processing:

$$\mathbf{x} = \mathrm{vec}(x) \in \mathbb{R}^{MN}$$

PCM will operate on each element $x[i]$.

3) Uniform quantization

Choose b bits per pixel $\Rightarrow L = 2^b$ quantization levels.

Step size:

$$\Delta = \frac{x_{\max} - x_{\min}}{L} = \frac{1}{L}$$

Quantizer index (encoder output):

$$k[i] = \left\lfloor \frac{x[i]}{\Delta} \right\rfloor, \quad k[i] \in \{0, 1, \ldots, L-1\} \tag{1}$$

Saturation/clipping is required for boundary cases (e.g., $x = 1$).

Reconstruction (dequantization at receiver): A common mid-rise reconstruction is:

$$\hat{x}[i] = \left(k[i] + \frac{1}{2}\right)\Delta$$

Alternative consistent rules are acceptable if the same is used in encoder/decoder.

4) Signal-to-Quantization Noise Ratio (SQNR)

SQNR measures the strength of a signal relative to the distortion introduced by the quantizer. If $x[i]$ is the original signal/image sample and $\hat{x}[i]$ is the reconstructed value after dequantization, the quantization error is

$$e[i] = x[i] - \hat{x}[i].$$

The signal power and quantization-noise power are computed as

$$P_{\text{signal}} = \frac{1}{N}\sum_{i=1}^{N} x[i]^2, \qquad P_{\text{q-noise}} = \frac{1}{N}\sum_{i=1}^{N} e[i]^2 = \frac{1}{N}\sum_{i=1}^{N} (x[i]-\hat{x}[i])^2. \tag{2}$$

Hence,

$$\text{SQNR} = \frac{P_{\text{signal}}}{P_{\text{q-noise}}}, \qquad \text{SQNR}_{\text{dB}} = 10\log_{10}\left(\frac{P_{\text{signal}}}{P_{\text{q-noise}}}\right) = 10\log_{10}\left(\frac{\sum x[i]^2}{\sum (x[i]-\hat{x}[i])^2}\right). \tag{3}$$

When only quantization noise is present (no channel noise), $P_{\text{q-noise}}$ equals the MSE between x and $\hat{x}$.

Ideal uniform quantizer (approx.): For step size Δ and no overload, the quantization noise power is $\sigma_e^2 \approx \Delta^2/12$, leading to

$$\text{SQNR}_{\text{dB}} \approx 10\log_{10}\left(\frac{12P_{\text{signal}}}{\Delta^2}\right).$$

For a full-scale sinusoid quantized with b bits, a commonly used approximation is

$$\text{SQNR}_{\text{dB}} \approx 6.02\,b + 1.76 \text{ dB}.$$

5) Binary encoding of indices

Each quantized index $k[i]$ is represented using exactly b bits:

$$k[i] \to \text{bin}(k[i]) \in \{0,1\}^b \tag{4}$$

Concatenating all codewords produces a serialized PCM stream:

$$\mathbf{b}_{\text{tx}} \in \{0,1\}^{bMN}$$

6) Why channel modeling is needed

In real links, transmitted bits can be corrupted due to noise, interference, fading, or synchronization errors. Unlike quantization distortion (which is *small* and *distributed*), bit errors can cause **large, structured artifacts** because a single flipped bit changes the decoded index.

7) Two simple channel models used in this lab

(A) Binary Symmetric Channel (BSC):

directly flips bits with probability p:

$$b_{\text{rx}} = b_{\text{tx}} \oplus z, \qquad z \sim \text{Bernoulli}(p)$$

This is the simplest abstraction of bit errors.

(B) AWGN channel with BPSK:

Map bits to symbols (let us follow BPSK Mapping):

$$0 \to +1, \quad 1 \to -1$$

Transmit over:

$$y = s + n, \qquad n \sim \mathcal{N}(0, \sigma^2)$$

Receiver decides:

$$\hat{s} = \text{sign}(y) \Rightarrow \hat{b}$$

Noise strength is controlled via E_b/N_0 (or SNR). This is more "physical" than BSC. *(More details in the section for implementation of Channel Models)*

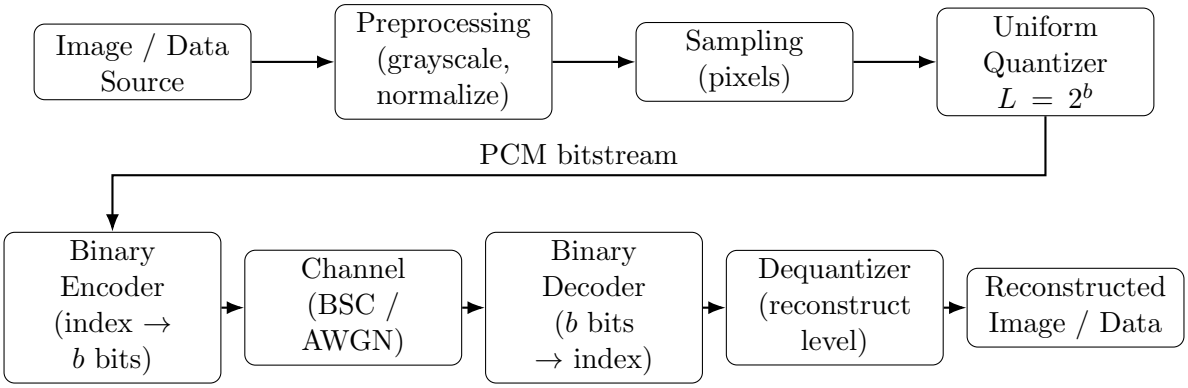

Figure 1: End-to-end PCM system.

Pre-Lab

1. For normalized pixels in $[0, 1]$, compute Δ for $b = 1, 2, 4, 8$.

2. For $b = 3$ ($L = 8$), quantize $x = \{0.12, 0.38, 0.77, 0.99\}$ and compute k and $\hat{x}$.

3. A 256×256 image is transmitted at 10 frames/s. Compute bit-rate for $b = 2, 4, 6, 8$.

4. Explain (2–3 lines): Why is PSNR used for images? What does "higher PSNR" visually

mean?

5. (Channel) If the bit flip probability is $p = 10^{-3}$, approximately how many bits are flipped in a 1 Mbit stream?

Experiment: Pulse Code Modulation and Demodulation

The experiment is divided into three parts - PCM Encoder, Channel Modeling and PCM Decoder.

Part A: PCM Encoder

Task 1: Load and prepare the image

T1-1 Load an image, convert to grayscale.
T1-2 Resize (recommended: 256×256) to keep runtime reasonable.
T1-3 Normalize to $[0, 1]$ as $x = X/255$.

PTQ1. The size of the image $M \times N =$ _____

PTQ2. The maximum and minimum pixel values of the image before normalization is _____ and _____.

PTQ3. The maximum and minimum pixel values of the image after normalization is _____ and _____.

.

Task 2: PCM Encoder (Quantizer + Bitstream)

For each $b \in \{1, 2, 3, 4, 5, 6, 7, 8\}$:

T2-1 Set $L = 2^b$ and $\Delta = 1/L$.

T2-2 Compute index array $k[i]$ for all pixels (See equation (1)).

T2-3 Convert each $k[i]$ to a b-bit codeword and serialize to bitstream $\mathbf{b}_{tx}$ (See equation (4)).

T2-4 Compute the SQNR (See equations (2) and (3)).

PTQ4. The bitstream length $= bMN$ for each b respectively are _____.

PTQ5. The value of $k[i]$ for $i = 10$, and $b = 4$ is _____.

PTQ6. Write the first 8 indices $k[1:8]$ and first 8 bits for $b = 2$ and $b = 8$.

PTQ7. Fill the following SQNR values

b	SQNR
1	
2	
3	
4	

b	SQNR
5	
6	
7	
8	

Common Mistake

Do NOT forget to clip indices: when $x = 1$, $\lfloor 1/\Delta \rfloor = L$, which is invalid. Use $k \leftarrow \min(k, L - 1)$.

Part B: Channel Modeling

In a practical communication system, the transmitted PCM bitstream is affected by **channel impairments** such as noise and interference. In this lab, we model the channel at the **bit level** using:

- **Binary Symmetric Channel (BSC)**: flips bits with probability p (simple abstract error model).
- **BPSK + AWGN channel**: maps bits to symbols, adds Gaussian noise, and performs detection (more physical/realistic baseband model).

Note: The BSC and AWGN are two different types of channel models. The PCM bit stream is send through either of the channels at a time.

Key Terms

1) What is SNR?

Signal-to-Noise Ratio (SNR) measures how strong the signal is compared to noise:

$$\text{SNR} = \frac{P_{\text{signal}}}{P_{\text{noise}}} \qquad \text{SNR}_{\text{dB}} = 10\log_{10}\left(\frac{P_{\text{signal}}}{P_{\text{noise}}}\right)$$

In a digital modulation setting (e.g., BPSK), "signal" refers to the transmitted symbol energy/power, and "noise" refers to the AWGN variance.

2) What is E_b? (Energy per bit)

E_b is the **average energy used to transmit one bit**. If we use BPSK mapping $0 \to +1$, $1 \to -1$, and each symbol carries 1 bit, then for amplitude ± 1:

$$E_b = 1$$

(assuming symbol duration is normalized and the symbol energy is 1. Energy of a symbol is calculated as square of absolute value of the symbol.)

3) What is N_0? (Noise spectral density)

In AWGN, noise is modeled as a random process with **two-sided power spectral density** $N_0/2$. For baseband discrete-time simulation, we typically use:

$$n \sim \mathcal{N}(0, \sigma^2), \quad \sigma^2 = \frac{N_0}{2}$$

So once you choose N_0, you immediately know the Gaussian noise variance σ^2.

4) What is E_b/N_0 and why use it?

$$\frac{E_b}{N_0}$$

is a standard measure of link quality. It compares **energy per bit** to **noise density**. Higher E_b/N_0 means a cleaner channel, lower BER, and better reconstructed image.

In dB:

$$\left(\frac{E_b}{N_0}\right)_{\text{dB}} = 10\log_{10}\left(\frac{E_b}{N_0}\right)$$

5) How to "set SNR" in simulation?

In BPSK+AWGN, you control the noise by choosing a desired E_b/N_0 (or SNR) and then computing noise variance. For BPSK with $E_b = 1$:

$$N_0 = \frac{E_b}{10^{(E_b/N_0)_{\text{dB}}/10}} = \frac{1}{10^{(E_b/N_0)_{\text{dB}}/10}} \quad \Rightarrow \quad \sigma^2 = \frac{N_0}{2}$$

Then generate $n \sim \mathcal{N}(0, \sigma^2)$ and add to the transmitted symbols.

Tip

For this lab: Use E_b/N_0 (0 dB, 5 dB, 10 dB) as the knob to control channel quality. For BSC: Use bit flip probability p as the knob.

Channel A: Binary Symmetric Channel (BSC)

Task 3A: Implement BSC

T3A-1 Choose $p \in \{0,\ 10^{-4},\ 10^{-3}\}$.

T3A-2 Generate an error vector $\mathbf{z}$ of the same length as $\mathbf{b}_{tx}$, where $z[j] = 1$ with probability p.

T3A-3 Apply XOR: $\mathbf{b}_{rx} = \mathbf{b}_{tx} \oplus \mathbf{z}$.

T3A-4 Estimate observed BER as $\widehat{p} = \frac{\#\text{flipped bits}}{\text{total bits}}$.

Python snippet: BSC

```
import numpy as np

def channel_bsc(b_tx, p, rng=None):
    """
    Binary Symmetric Channel:
    flips each bit independently with probability p.

    b_tx: numpy array of 0/1 bits
    p: bit flip probability (BER parameter)
    """
    if rng is None:
        rng = np.random.default_rng()

    # z[j] = 1 with probability p (bit flip mask)
    z = (rng.random(b_tx.shape) < p).astype(np.uint8)

    # XOR to flip bits where z==1
    b_rx = np.bitwise_xor(b_tx.astype(np.uint8), z)

    # measured BER
    ber_meas = np.mean(b_rx != b_tx)
    return b_rx, ber_meas
```

Interpretation: BSC directly enforces a BER-like behavior. It does not model modulation or waveform; it is an abstract bit error model.

Channel B: BPSK + AWGN — closer to physical link

Task 3B: Implement BPSK+AWGN

1. Map bits to BPSK symbols: $0 \to +1$, $1 \to -1$ to obtain $\mathbf{s}$.
2. Choose E_b/N_0 in dB (Take values in dB: 0 to 20 in increments of 2).
3. Convert E_b/N_0 to noise variance. For unit-energy BPSK ($E_b = 1$):

$$N_0 = \frac{1}{10^{(E_b/N_0)_{\text{dB}}/10}}, \quad \sigma^2 = \frac{N_0}{2}$$

4. Generate AWGN $n \sim \mathcal{N}(0, \sigma^2)$ and compute $y = s + n$.
5. Hard decision: $\hat{s} = \text{sign}(y)$ and map back to bits $\mathbf{b}_{rx}$ (If the sign of y is positive then 0 is assumed to transmitted otherwise 1.)

Python snippet: BPSK mapping, AWGN addition, hard decision

```
import numpy as np

def bpsk_modulate(b):
    """
    BPSK mapping:
    0 -> +1
    1 -> -1
    """
    b = b.astype(np.uint8)
    return 1 - 2*b   # 0->1, 1->-1

def bpsk_demodulate_hard(y):
    """
    Hard decision:
    y >= 0 -> bit 0
    y < 0  -> bit 1
    """
    return (y < 0).astype(np.uint8)

def channel_awgn_bpsk(b_tx, ebn0_db, rng=None):
    """
    AWGN channel with BPSK.
    ebn0_db: desired Eb/N0 in dB

    Assumptions:
    - BPSK symbols are +/-1, so Eb = 1 (one bit per symbol).
    - noise variance sigma^2 = N0/2 with N0 = Eb / (Eb/N0).
    """
    if rng is None:
        rng = np.random.default_rng()

    s = bpsk_modulate(b_tx)  # transmit symbols (+1/-1)

    ebn0_lin = 10**(ebn0_db/10.0)  # convert dB to linear
    Eb = 1.0
```

```
    N0 = Eb / ebn0_lin
    sigma2 = N0 / 2.0
    sigma = np.sqrt(sigma2)

    n = rng.normal(loc=0.0, scale=sigma, size=s.shape)  # AWGN
    y = s + n

    b_rx = bpsk_demodulate_hard(y)

    ber_meas = np.mean(b_rx != b_tx)
    return b_rx, ber_meas
```

How does E_b/N_0 relate to SNR here?

For BPSK with one bit per symbol and unit-energy symbols:

$$E_b = E_s = 1$$

Thus choosing E_b/N_0 effectively sets the "SNR per bit". Higher E_b/N_0 means smaller σ^2 (less noise) and hence lower BER.

Tip

Typical observations:

- At 0 dB: noticeable BER, visible artifacts in reconstructed image.
- At 5 dB: BER drops significantly, image improves.
- At 10 dB: BER becomes very small, image is close to the no-channel case.

Mini-demo loop (BER vs E_b/N_0)

```
ebn0_list = [0, 5, 10]
for ebn0_db in ebn0_list:
    b_rx, ber = channel_awgn_bpsk(b_tx, ebn0_db)
    print(f"Eb/N0 = {ebn0_db:2d} dB  -->  measured BER = {ber:.6f}")
```

Tip

BSC is parameterized directly by BER p. AWGN is parameterized by E_b/N_0; BER is an outcome after detection.

Part C: PCM Decoder

Task 4: PCM Decoder (Bitstream to Image)

For each chosen b:

T4-1 Group received bits $\mathbf{b}_{rx}$ into blocks of b bits.

T4-2 Convert each block to decimal index $\hat{k}[i]$.

T4-3 Reconstruct $\hat{x}[i] = (\hat{k}[i] + \frac{1}{2})\Delta$.

T4-4 Reshape into $\hat{X}$ of size $M \times N$ and display.

Task 5: Compute Metrics and Plot Trends

For each b:

T5-1 Compute MSE, SNR(dB), PSNR(dB) between original and reconstructed images.

T5-2 Plot **PSNR vs** b.

T5-3 Plot **SNR vs** b.

T5-4 Repeat the above for No channel, BSC and AWGN channels.

Peak Signal-to-Noise Ratio (PSNR)

Peak Signal-to-Noise Ratio (PSNR) is a widely used objective metric to quantify the quality of a reconstructed image (after quantization/PCM decoding, compression, or transmission errors) with respect to the original image. A higher PSNR generally indicates that the reconstructed image is closer to the original.

1) Mean Squared Error (MSE)

Let X be the original image and $\hat{X}$ be the reconstructed image of size $M \times N$. The **mean squared error (MSE)** is defined as

$$\mathrm{MSE} = \frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} \Big(X[m,n] - \hat{X}[m,n] \Big)^2 .$$

Smaller MSE implies smaller reconstruction error.

2) PSNR Definition

PSNR is defined in decibels (dB) as

$$\mathrm{PSNR\,(dB)} = 10 \log_{10} \left(\frac{\mathrm{MAX}^2}{\mathrm{MSE}} \right),$$

where MAX is the maximum possible pixel value of the image.

- For 8-bit grayscale images with pixel range $[0, 255]$, $\quad \mathrm{MAX} = 255$.
- If pixels are normalized to the range $[0, 1]$, $\quad \mathrm{MAX} = 1$.

3) Practical Interpretation (Rule of Thumb)

Although the exact perception depends on the image content, typical PSNR ranges are:

- PSNR < 20 dB: poor quality (heavy distortion)
- 20–30 dB: noticeable distortion
- 30–40 dB: good quality
- > 40 dB: very high quality (differences are difficult to notice)

4) Note

If MSE $= 0$ (perfect reconstruction), PSNR becomes infinite since $\log_{10}(\mathrm{MAX}^2/0) \to \infty$.

Observations / Results

Table A: No channel (quantization only)

b	L	Bitstream length (bMN)	MSE	SNR (dB)	PSNR (dB)
1					
2					
3					
4					
5					
6					
7					
8					

Table B: With BSC channel

Parameter (p or E_b/N_0): _____

b	Bit errors / BER	MSE	SNR (dB)	PSNR (dB)	Visual artifacts (short note)
2					
4					
6					
8					

Table C: With AWGN channel

Parameter (p or E_b/N_0): _____

b	Bit errors / BER	MSE	SNR (dB)	PSNR (dB)	Visual artifacts (short note)
2					
4					
6					
8					

Plots

1. Plot of SNR vs b in AWGN channel

2. Plot of SNR vs b in BSC channel

Post-Lab Questions

Answer using your own results/plots/tables wherever possible.

1. From your Table A, how much does PSNR increase when b increases from 2 to 3, and from

7 to 8? Do you observe a roughly constant increment? Explain using Δ.

2. In your implementation, what reconstruction rule did you use: $\hat{x} = (k + \frac{1}{2})\Delta$ or $\hat{x} = k\Delta$ or something else? How would the choice affect bias and MSE?

3. For the same image, why does the **visual improvement** from $b = 1$ to $b = 3$ appear larger than from $b = 6$ to $b = 8$, even if PSNR increases in both cases?

4. Compare BSC and AWGN: For similar measured BER, do the reconstructed images look similar? Why/why not?

5. In your encoder, did you serialize bits as MSB→LSB or LSB→MSB? Demonstrate with one example index (e.g., $k = 5$ with $b = 3$) and show its bit pattern. What happens if transmitter and receiver disagree on bit order?

6. (Sensitivity) Which bit position (MSB or LSB) is more critical in your PCM codeword? Propose and test a simple experiment: flip only MSB bits (or only LSB bits) and compare reconstruction quality.

7. Suggest one method to improve robustness without changing b (e.g., repetition coding, parity, CRC + retransmission). Where would it be inserted in your pipeline?

8. If you were to transmit a **color image**, how would the PCM bitstream size change? State

assumptions and compute the new bit-rate for 256×256 at 10 fps with $b = 6$ per channel.

Concluding Remarks

(*Give briefly what you have understood from the experiment.*)

Attach Code: The code should contain the following plots

1. Original image and reconstructed images grid without a channel.
2. Reconstructed images with channel corruption using BSC and AWGN channels.
3. Plot of SQNR vs b.
4. Plot of PSNR vs b using BSC and AWGN channels.
5. Plot of SNR vs b using BSC and AWGN channels.

Student's Signature: _____