Signal Processing Lab Report 6

Aasrith Reddy and Ritama sanyal

1 Continuous-time Fourier Transform

1.1 continuous FT Function

The Inputs to this function are:

- t - Symbolic variable.
- x(t) - signal whose FT is to be computed.
- a,b – the signal is equal to x(t) in the interval [a, b] and zero outside.
- ω – the vector ω contains the values of frequency where FT is to be computed.

Initialization:

- `X = zeros(size(w));`: Initializes the output `X` as a zero vector with the same size as the frequency vector `w`. This will hold the CTFT values.

For Loop (Frequency Iteration):

- The loop iterates over each frequency value in `w`.
- `integrand = x_t .* exp(-1j * w(k) * t);`: Computes the integrand of the CTFT, which is the product of the signal `x_t` and the complex exponential term `exp(-1j * w(k) * t)` for the `k`-th frequency.

Integration:

- `X(k) = int(integrand, t, a, b);`: Integrates the `integrand` over the time interval $[a, b]$ to compute the CTFT value for the `k`-th frequency and stores it in `X(k)`.

The function returns `X`, which contains the CTFT values for each frequency in `w`.

1.2 when input is Rectangular pulse

Considering a Rectangular pulse of unit amplitude in [-T,T]. Considering $\omega = -5 : 0.1 : 5$

CTFT of a Rectangular Pulse

The continuous-time Fourier transform (CTFT) of a rectangular pulse of unit amplitude defined over the interval $[-T, T]$ is given by:

$$X(\omega) = \int_{-T}^{T} 1 \cdot e^{-j\omega t} \, dt$$

Evaluating the integral:

$$X(\omega) = \int_{-T}^{T} e^{-j\omega t} \, dt$$

This integral can be solved as:

$$X(\omega) = \left[\frac{e^{-j\omega t}}{-j\omega} \right]_{-T}^{T}$$

$$X(\omega) = \frac{1}{-j\omega} \left(e^{-j\omega T} - e^{j\omega T} \right)$$

Using the identity $e^{-j\theta} - e^{j\theta} = -2j\sin(\theta)$:

$$X(\omega) = \frac{1}{-j\omega} \cdot (-2j)\sin(\omega T)$$

$$X(\omega) = \frac{2\sin(\omega T)}{\omega}$$

Thus, the CTFT of the rectangular pulse is:

$$X(\omega) = 2T \cdot \operatorname{sinc}\left(\frac{\omega T}{\pi} \right)$$

where the sinc function is defined as:

$$\operatorname{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$$

So:

$$X(\omega) = 2T \cdot \frac{\sin(\omega T)}{\omega T}$$

This result shows that the CTFT of a rectangular pulse is a sinc function, which is centered at $\omega = 0$ and whose width is inversely proportional to T. As T increases, the pulse in the frequency domain becomes narrower, indicating that the time-domain width and frequency-domain width are inversely related.

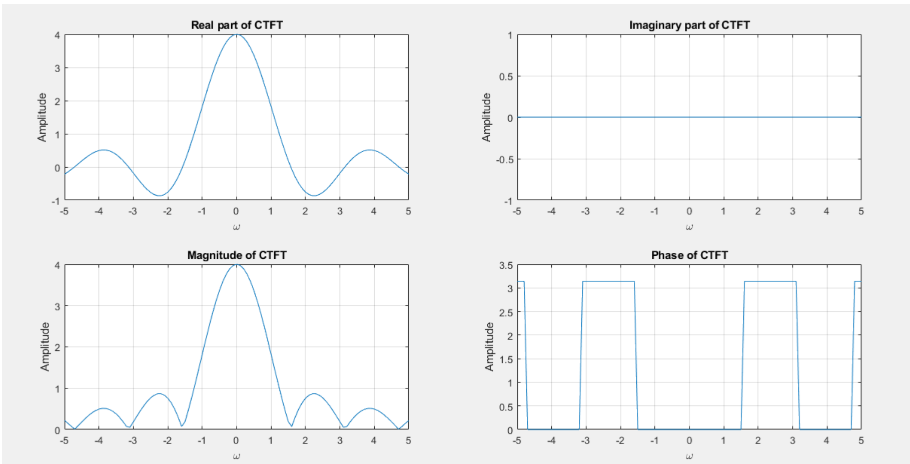

Figure 1.1: CTFT of Rectangular Pulse of unit amplitude in [-2,2]

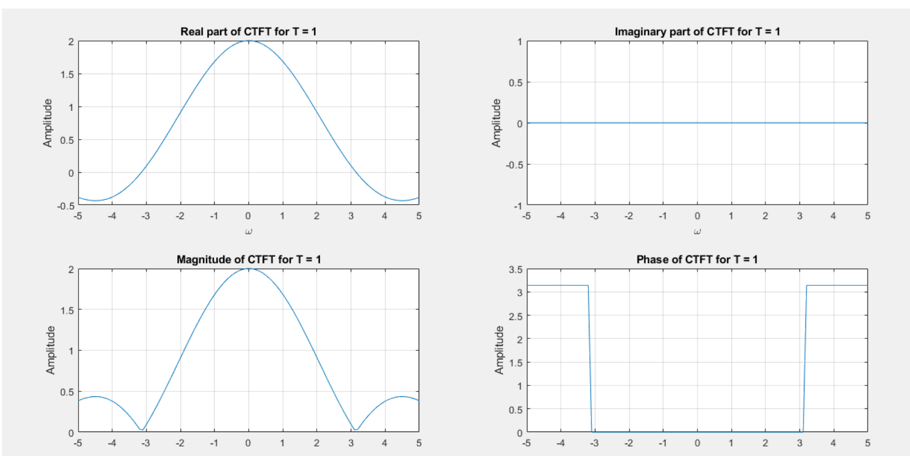

Figure 1.2: CTFT of Rectangular Pulse of unit amplitude in [-1,1]

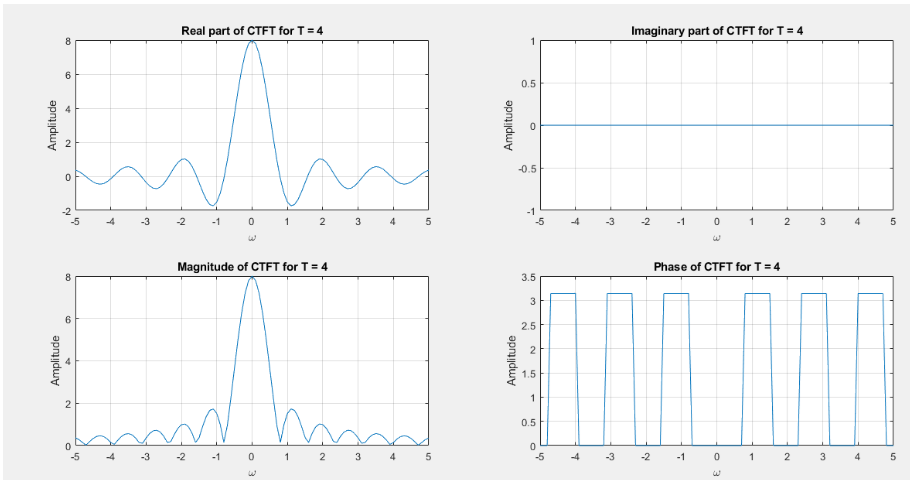

Figure 1.3: CTFT of Rectangular Pulse of unit amplitude in [-4,4]

Observation for Different Values of T:

- As T changes, the width of the rectangular pulse affects the CTFT. This observation is supported by the **scaling property** of the Fourier transform, which states that increasing the width of the time-domain signal (rectangular pulse) results in a narrower and more concentrated spectrum in the frequency domain, and vice versa.

1.3 when input is e^{jt}

The continuous-time Fourier transform (CTFT) of e^{jt} is given by:

$$X(\omega) = \int_{-\infty}^{\infty} e^{jt} \cdot e^{-j\omega t}\, dt$$

$$X(\omega) = \int_{-\infty}^{\infty} e^{j(1-\omega)t}\, dt$$

To evaluate this integral, note that it represents the Fourier transform of a complex exponential function. The result is:

$$X(\omega) = 2\pi\delta(\omega - 1)$$

where $\delta(\omega - 1)$ is the Dirac delta function centered at $\omega = 1$. This indicates that the CTFT of e^{jt} is an impulse located at $\omega = 1$ with a magnitude of 2π.
Expected FT and Observations:

- For $x(t) = e^{jt}$, the CTFT is expected to be a shifted impulse at $\omega = 1$, as the Fourier transform of $e^{j\omega_0 t}$ is $2\pi\delta(\omega - \omega_0)$.

- The plots show these characteristics, with the magnitude showing peaks at expected frequencies and the phase plots showing corresponding jumps.

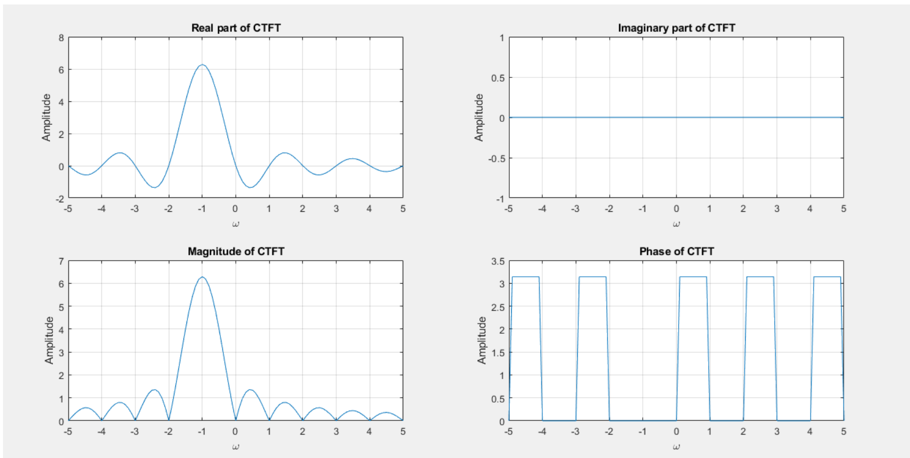

Figure 1.4: CTFT of e^{jt}

1.4 when input is cos(t)

The continuous-time Fourier transform (CTFT) of $\cos(t)$ is:

$$X(\omega) = \int_{-\infty}^{\infty} \cos(t) \cdot e^{-j\omega t}\, dt$$

Using the Euler's formula, $\cos(t) = \frac{1}{2}(e^{jt} + e^{-jt})$, the expression becomes:

$$X(\omega) = \int_{-\infty}^{\infty} \frac{1}{2}\left(e^{jt} + e^{-jt}\right) \cdot e^{-j\omega t}\, dt$$

$$X(\omega) = \frac{1}{2}\int_{-\infty}^{\infty} e^{j(1-\omega)t}\, dt + \frac{1}{2}\int_{-\infty}^{\infty} e^{-j(1+\omega)t}\, dt$$

The integral of a complex exponential results in a Dirac delta function. Therefore:

$$X(\omega) = \frac{1}{2} \cdot 2\pi\delta(\omega - 1) + \frac{1}{2} \cdot 2\pi\delta(\omega + 1)$$

$$X(\omega) = \pi\delta(\omega - 1) + \pi\delta(\omega + 1)$$

This result indicates that the CTFT of $\cos(t)$ consists of two impulses located at $\omega = 1$ and $\omega = -1$, each with a magnitude of π.

Expected FT and Observations:

- For $x(t) = \cos(t)$, the CTFT is expected to consist of two impulses at $\omega = \pm 1$, as $\cos(t)$ can be represented as $\frac{1}{2}(e^{jt} + e^{-jt})$.

- The plots show these characteristics, with the magnitude showing peaks at expected frequencies and the phase plots showing corresponding jumps.

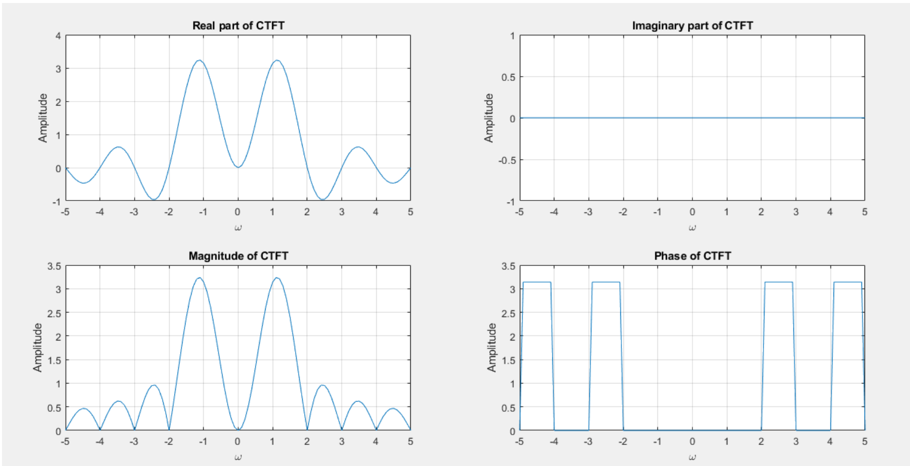

Figure 1.5: CTFT of $cos(t)$

1.5 when input is a triangle pulse of height 1 and base/support [-1,1]

The triangular pulse $x(t)$ of height 1 and base/support $[-1, 1]$ is defined as:

$$x(t) = \begin{cases} 1 - |t|, & |t| \leq 1 \\ 0, & \text{otherwise} \end{cases}$$

To compute the CTFT, we start with the definition:

$$X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} \, dt$$

Since $x(t)$ is non-zero only in the interval $[-1, 1]$, we can rewrite the integral as:

$$X(\omega) = \int_{-1}^{1} (1 - |t|) e^{-j\omega t} \, dt$$

Next, we can split this integral into two parts due to the absolute value:

$$X(\omega) = \int_{-1}^{0} (1 + t) e^{-j\omega t} \, dt + \int_{0}^{1} (1 - t) e^{-j\omega t} \, dt$$

Calculating these integrals separately:
1. For the first integral:

$$X_1(\omega) = \int_{-1}^{0} (1 + t) e^{-j\omega t} \, dt$$

2. For the second integral:

$$X_2(\omega) = \int_0^1 (1 - t)e^{-j\omega t}\, dt$$

Combining both results, we get:

$$X(\omega) = X_1(\omega) + X_2(\omega)$$

The result of these integrals yields the Fourier transform of the triangular pulse, which can be simplified to:

$$X(\omega) = \frac{2\sin^2\left(\frac{\omega}{2}\right)}{\left(\frac{\omega}{2}\right)^2} = 2\text{sinc}^2\left(\frac{\omega}{2}\right)$$

Thus, the CTFT of the triangular pulse of height 1 and base/support $[-1, 1]$ is given by:

$$X(\omega) = 2\text{sinc}^2\left(\frac{\omega}{2}\right)$$

A triangular pulse can be viewed as the convolution of two rectangular pulses. If we denote a rectangular pulse of height 1 and width 1 as $\text{rect}(t)$, the triangular pulse can be expressed as:

$$x(t) = \text{rect}(t) * \text{rect}(t)$$

The Fourier transform property states that the FT of the convolution of two signals is the product of their individual FTs. Since the FT of a rectangular pulse $\text{rect}(t)$ is a sinc function:

$$\text{FT}\{\text{rect}(t)\} = \text{sinc}(\omega)$$

The FT of the triangular pulse $x(t)$ becomes:

$$X(\omega) = \text{sinc}^2\left(\frac{\omega}{2}\right)$$

So, the expected FT is a squared sinc function, centered at $\omega = 0$.

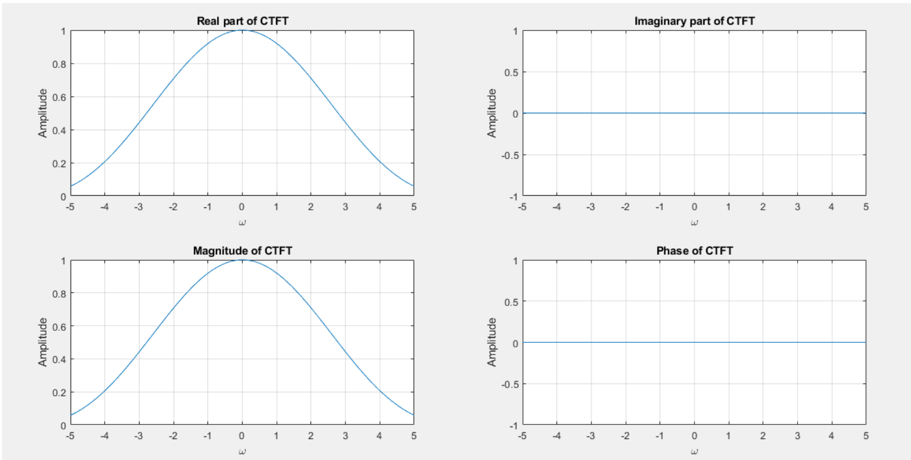

Figure 1.6: CTFT of a triangular pulse

2 Fast Fourier Transform (Radix-2)

2.1 radix2fft Function:

The Fast Fourier Transform (FFT) is an efficient algorithm to compute the Discrete Fourier Transform (DFT) and its inverse. The DFT of a sequence $x[n]$ of length N is defined as:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}, \quad k = 0, 1, \ldots, N-1$$

Computing the DFT directly requires $O(N^2)$ operations. The FFT reduces this complexity to $O(N \log N)$, making it a critical tool in signal processing, data analysis, and many other fields.

Radix-N FFT

The radix-N FFT algorithm computes the DFT by recursively breaking down a DFT of size N into N smaller DFTs. The most common versions are radix-2 (where $N = 2$), radix-4 (where $N = 4$), and radix-8 (where $N = 8$).

Recursive Structure

The radix-N FFT operates on a sequence of length $N = N_1 \cdot N_2$, where N_1 and N_2 are the sizes of the subproblems. The sequence is divided into N_1 groups, each containing N_2 elements.

Decimation-In-Time (DIT) Algorithm

The Decimation-In-Time (DIT) approach is a popular implementation of the radix-2 FFT. In this approach, the input sequence is divided into even and odd indexed samples, allowing the FFT to be computed recursively.

Splitting the Sequence

For a sequence of length N, the input $x[n]$ is split as follows:

$$x_{\text{even}}[k] = x[2k], \quad x_{\text{odd}}[k] = x[2k+1]$$

for $k = 0, 1, \ldots, \frac{N}{2} - 1$.
Combining Results
After computing the FFT for the even and odd indexed parts, the results are combined using twiddle factors:

$$X[k] = X_{\text{even}}[k] + W_N^k \cdot X_{\text{odd}}[k]$$

$$X[k + \frac{N}{2}] = X_{\text{even}}[k] - W_N^k \cdot X_{\text{odd}}[k]$$

where $W_N = e^{-j\frac{2\pi}{N}}$ is the twiddle factor.
Properties of FFT

- **Periodicity**: The DFT is periodic with period N.
- **Symmetry**: For real-valued input signals, the DFT exhibits conjugate symmetry.
- **Linearity**: The DFT is a linear operation, meaning that the transform of a sum is the sum of the transforms.

Applications The FFT algorithm is widely used in various fields such as:

- Signal processing (filter design, spectral analysis)
- Image processing (image compression, feature extraction)
- Communications (modulation, demodulation)

Conclusion The radix-N FFT algorithm significantly improves the efficiency of computing the DFT, making it a fundamental tool in digital signal processing. Understanding its structure and implementation is crucial for effectively utilizing Fourier analysis in practical applications.
Input Validation The function checks if the length of the input vector N is a power of 2. If not, it throws an error.
Base Case If $N = 2$, the function directly computes the DFT for the two-element case:

$$X = \begin{bmatrix} x(1) + x(2) \\ x(1) - x(2) \end{bmatrix}$$

Recursive Case The input vector x is split into even-indexed and odd-indexed elements, and `radix2fft` is called recursively on both halves.

Combining Results

After obtaining the FFTs of the even and odd parts, it combines them using the twiddle factors, which are calculated as:

$$e^{-\frac{j2\pi k}{N}}$$

where k ranges from 0 to $\frac{N}{2} - 1$.

```
Output from radix2fft:
     0
     2

Output from fft:
     0     2
```

Figure 2.1: Display Output

3 Quantization

3.1 Explanation of quadrating quant

The provided function `quadratic_quant` is specifically a non-uniform quadratic quantizer.

- The function `quadratic_quant` is designed to quantize a signal `x` using `B` bits and a range parameter `a`.
- It divides the interval $[0, a)$ into $L = 2^{(B-1)}$ equal parts for quantization.
- The midpoints for quantization are calculated in both positive and negative ranges (`midpoints_pos` and `midpoints_neg`).
- The function then loops through each element of the input signal `x`:
  - If a value exceeds the range $[-a, a)$, it is mapped to the nearest extreme quantization level.
  - For values within the range, they are mapped to the nearest midpoint based on the appropriate quantization interval.

3.2 Using above quantized function

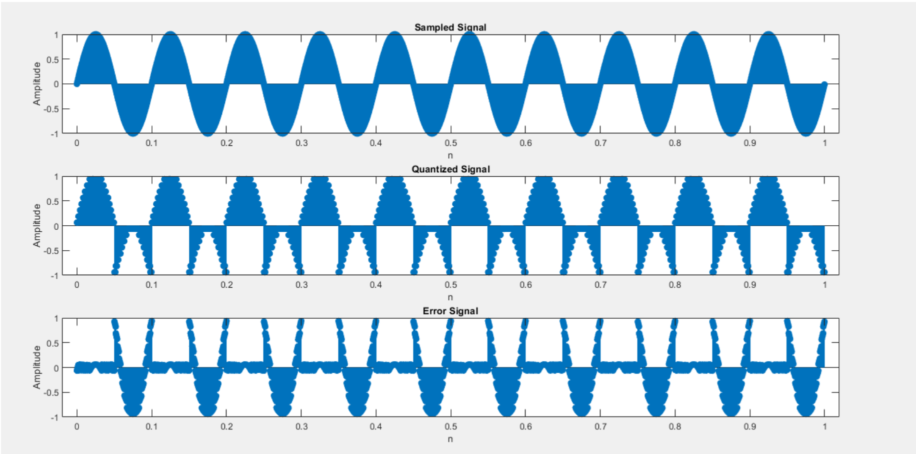

Figure 3.1: Sampled signal Quantised signal Error signal

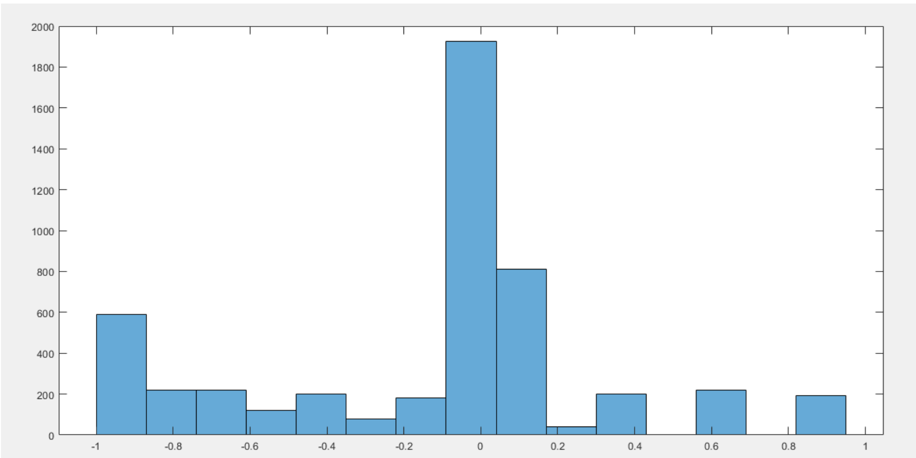

Figure 3.2: Histogram of quantization error

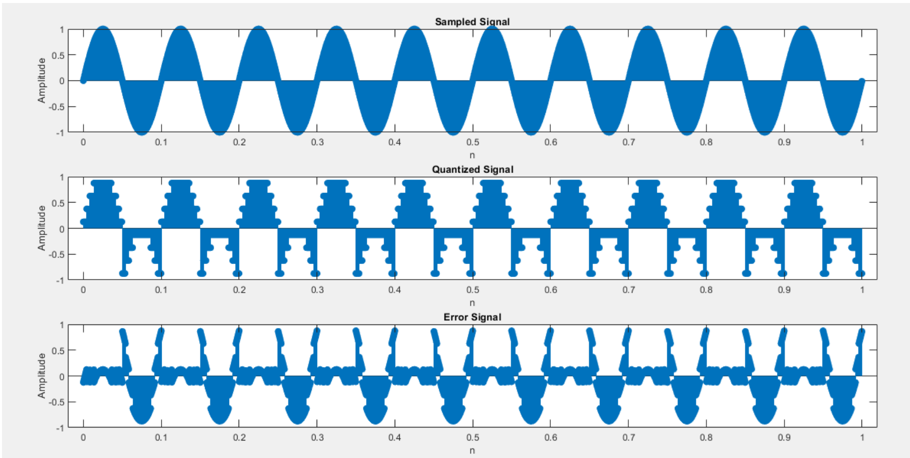

Figure 3.3: Same experiment for B=3

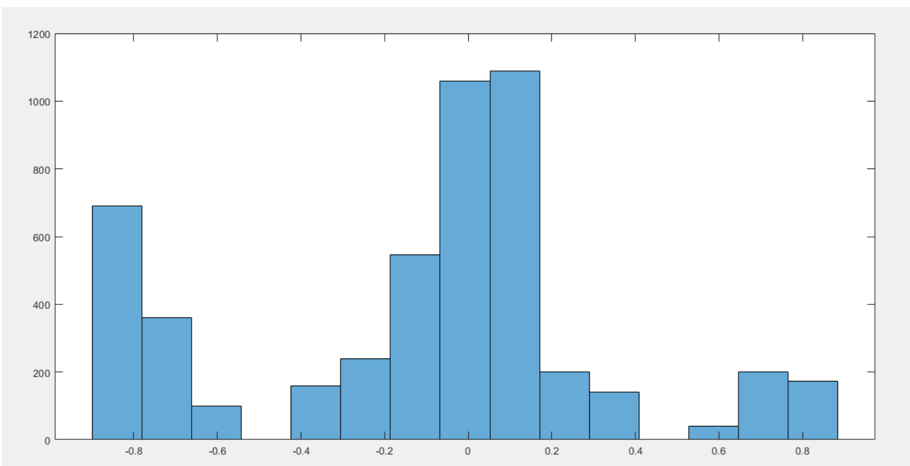

Figure 3.4: Histogram for B=3

Key Differences:

- **First Histogram (Higher B):**
  - The quantization error is more concentrated around 0, indicating fewer large errors.
  - This suggests that with more bits for quantization, the signal is represented more accurately, resulting in smaller quantization errors.
- **Second Histogram (Lower B):**

– The quantization error is more spread out, with larger error magnitudes.
– With fewer bits, there is less precision in the representation of the signal, leading to larger quantization errors.

3.3 Maximum Absolute Quantisation Error

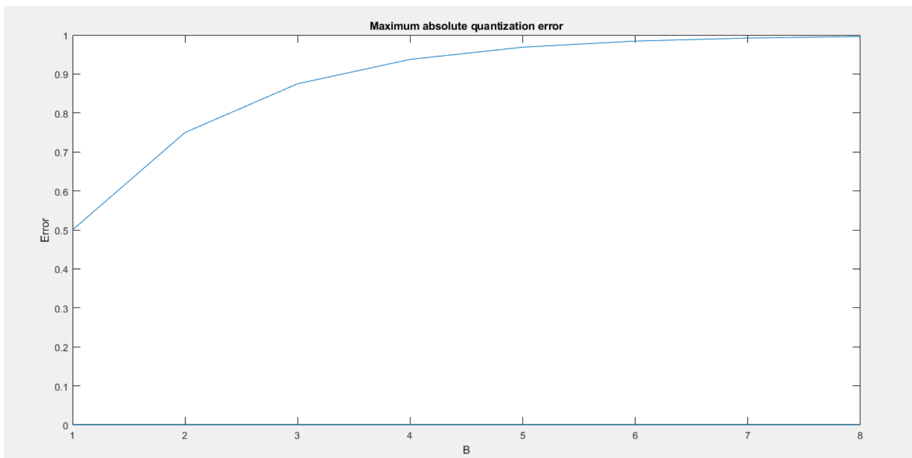

Figure 3.5: Maximum absolute quantization error

Observations:

- As the number of bits B increases, the maximum absolute quantization error decreases.
- The curve initially decreases sharply, indicating that increasing B from a small value significantly reduces the error.
- After a certain point (around $B = 4$), the reduction in quantization error becomes more gradual, and eventually reaches a near-flat region, suggesting diminishing returns in error reduction with higher values of B.
- This implies that beyond a certain number of bits, additional bits do not significantly improve the accuracy of the quantization.

Conclusion: Increasing the number of bits improves the accuracy of quantization, but after a certain threshold, the improvements become marginal.

3.4 Signal to Quantization Noise Ratio

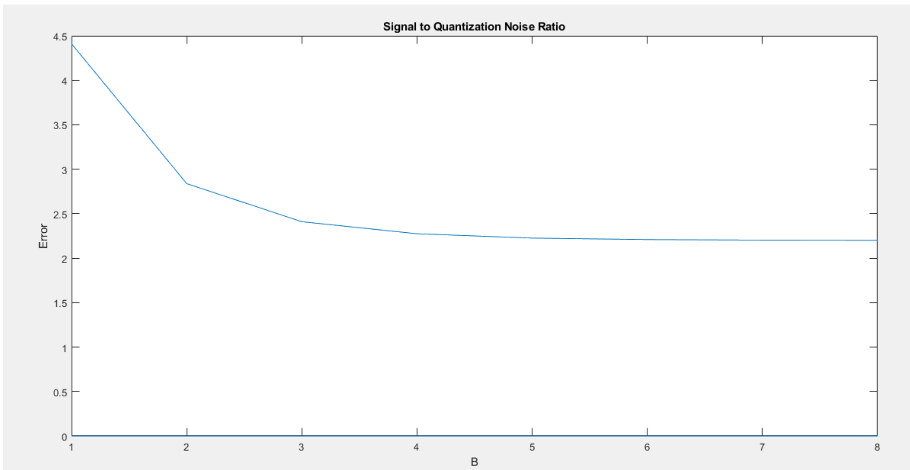

Figure 3.6: SQNR

Observations:

- As the number of bits B increases, the maximum absolute quantization error decreases.
- The curve initially decreases sharply, indicating that increasing B from a small value significantly reduces the error.
- After a certain point (around $B = 4$), the reduction in quantization error becomes more gradual, and eventually reaches a near-flat region, suggesting diminishing returns in error reduction with higher values of B.
- This implies that beyond a certain number of bits, additional bits do not significantly improve the accuracy of the quantization.

Conclusion: Increasing the number of bits improves the accuracy of quantization, but after a certain threshold, the improvements become marginal.

3.5 part e

The non-uniform quantizer offers higher accuracy near zero, with finer quantization levels in the central region of $[-a, a)$, making it ideal for signals where small values are more frequent (e.g., audio). However, accuracy decreases towards the extremes ($\pm a$) due to larger step sizes.

In contrast, the uniform quantizer divides the interval into equal steps, providing consistent accuracy across the range but lacking the finer resolution near zero that the non-uniform quantizer offers.

Comparison:

- **Non-uniform quantizer**: More accurate near zero, less accurate at extremes.
- **Uniform quantizer**: Equal accuracy throughout, but less precision for small values.

Conclusion: Non-uniform quantization is better for signals with small, frequently occurring values, while uniform quantization ensures consistent accuracy across the entire range.

4 Quantization of Audio signals

4.1 part a

Comparison of Sound Quality:

- With $B = 3$, the quantized signal will likely have noticeable distortion compared to the original signal.
- The quantization introduces fewer levels (only 8 possible values since $2^3 = 8$), leading to a loss of detail in the signal.
- The degradation in sound quality is more prominent in the smoother parts of the audio where finer variations in amplitude are essential for preserving the sound's fidelity.
- The original signal should sound much smoother, while the quantized version may exhibit a more "robotic" quality due to the reduced number of quantization levels.

4.2 part b

Observations:

- For small values of B (e.g., $B = 1$), the sound is highly distorted, with large quantization errors, making the audio sound harsh and unrecognizable.
- As B increases, the sound quality improves noticeably. By $B = 4$, the sound becomes smoother, although some distortion is still present.
- For $B = 6$ and above, the sound quality becomes much closer to the original signal, with only minimal audible differences.
- At $B = 8$, the sound is nearly indistinguishable from the original, with very fine quantization steps that preserve much of the signal's detail.

Conclusion: Increasing the number of quantization levels (B) significantly enhances the quality of the quantized signal, reducing distortion and making it sound more similar to the original.

4.3 part c

Effect of Quantization on Frequency Content:

- **Quantization Noise:** Quantization introduces noise, which is spread across a wide frequency range. This noise can distort the signal's original frequency content, especially at low B values.
- **Impact on Harmonics:** Lower values of B result in a rough approximation of the signal, introducing additional harmonics in the frequency spectrum and distorting the original frequencies.
- **Smoothing with Higher B:** As B increases, the quantization noise decreases, and the signal's frequency content becomes closer to the original, with fewer high-frequency artifacts.

Role of B:

- **Low B:** At low bit levels (e.g., $B = 1$ or $B = 2$), the signal is heavily distorted, introducing significant high-frequency noise and affecting the frequency content.
- **High B:** With higher values of B, the quantization error is reduced, leading to a more accurate representation of the signal's original frequency spectrum with minimal distortion.

Conclusion: Quantization at low B levels distorts the frequency content by introducing high-frequency noise. As B increases, the quantized signal retains more of its original frequency characteristics with less distortion.