ANALYSIS OF SAMPLING THEOREM USING PYTHON JUPYTER NOTEBOOK

Course Code and Name

CHARLES GITHINJI

ENG-219-017/2021

ECE 2414: DIGITAL COMMUNICATION 1

Date of Submission 7TH NOV 2024

LECTURER

MARTIN WAFULA

Contents

1 Abstract

This lab report explores the Sampling Theorem, which asserts that a continuous-time signal can be perfectly reconstructed from its samples if the sampling frequency is at least twice the maximum frequency of the signal (the Nyquist rate). The lab includes practical analysis of the theorem through the sampling and reconstruction of signals in both the time and frequency domains. Using Python and Jupyter Notebook, signal sampling, frequency spectrum analysis, and signal reconstruction were carried out to observe the effects of different sampling rates

2 Introduction

To provide a comprehensive understanding of Sampling Theory and Quantization, I'll expand on the mathematics behind these concepts, including Fourier analysis as applied to the Sampling Theorem and quantization errors. This breakdown will help you see how these mathematical foundations support digital communication.

Sampling Theory for Analog Waveform Sources

Overview of the Sampling Theorem

The Sampling Theorem is pivotal in digital signal processing, as it establishes the conditions for converting a continuous signal into a discrete sequence of samples without losing information. The theorem states that if an analog signal is band-limited (i.e., it has a maximum frequency component W , then it can be completely represented by its samples taken at intervals of T = 1/2W or faster. This minimum sampling rate, 2W , is called the Nyquist rate.

Mathematical Formulation

For a continuous-time signal x(t) with frequency content restricted to -W<=f<=W, the sampled signal xs(t) is obtained by multiplying x(t) with an impulse train where each impulse represents a sampling point:

$$x_s(t) = x(t) \cdot \delta_T(t)$$

The impulse train can be represented as:

$$\delta_T(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT)$$

Where T= 1/2W. Consequently, the sampled signal xs (t) becomes:

$$x_s(t) = \sum_{n=-\infty}^{\infty} x(nT) \cdot \delta(t - nT)$$

In the frequency domain, sampling in the time domain results in periodic replication of the signal's spectrum in the frequency domain. This replication can be derived using the Fourier Transformof the sampled signal. If X(f)) is the Fourier transform of x(t), then the Fourier transform of xs(t), denoted by Xs(f), is given by:

$$X_s(f) = \sum_{k=-\infty}^{\infty} X(f - kf_s)$$

Where fs =1/T is the sampling frequency. This equation shows that the spectrum of the sampled signal Xs(f) consists of copies of\X(f) spaced fs apart. To avoid aliasing (overlapping of spectral copies), the sampling frequency fs must be at least twice the highest frequency W of x(t), i.e., fs >= 2W.

Signal Reconstruction using Sinc Interpolation

The original signal Xt can be reconstructed from its samples x(nT) by using a sinc function, which acts as an ideal low-pass filter:

$$x(t) = \sum_{n=-\infty}^{\infty} x(nT) \cdot \text{sinc}\left(\frac{t - nT}{T}\right)$$

where the sinc function $\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$ interpolates between sample points. This sinc interpolation reconstructs the continuous waveform from discrete samples, provided the sampling frequency satisfies the Nyquist criterion.

Quantization for Analog Sequence Sources

Purpose of Quantization

After sampling, the next step in digital communication is quantization, which converts each sample into one of a finite number of discrete values or quantization levels. Quantization is necessary because each sample could take on any real value, while digital systems can only store or transmit finite, discrete levels.

Quantization Process and Quantization Levels

1. Defining Quantization Regions: Let the range of sample values be divided into quantization regions Ri such that each region is mapped to a corresponding quantized value or level qi
2. Mapping to Nearest Quantization Level: Each sample value s within a region Ri is approximated by the quantization level qi that minimizes the distance to s.

For a sample s within a region Ri, the quantization error ei(s) is given by:

$$e_i(s) = s - q_i$$

The mean squared quantization error (MSE) across all regions provides a measure of the accuracy of quantization. For a source symbol S with probability density, f S (s) the MSE over all quantization levels is:

$$\text{MSE} = \sum_{i=1}^{M} \int_{R_i} f_S(s)(s - q_i)^2 \, ds$$

This expression calculates the expected squared error, providing a measure of the distortion introduced by quantization.

Uniform vs. Non-Uniform Quantization

1. Uniform Quantization: In *uniform quantization*, the quantization intervals Ri are equal in width. This approach is suitable for signals with values distributed uniformly across the range.

2. Non-Uniform Quantization: For signals where values are not uniformly distributed, non-uniform quantization allocates smaller intervals where values are more frequent, reducing quantization error for these values. Techniques like *μ-law* and *A-law* companding are used in audio and telecommunications to achieve non-uniform quantization, effectively reducing perceptible noise in the reconstructed signal.

Practical Example of Quantization Error Calculation

Consider an 8-level quantizer with levels q1, q2, q8 that are optimally placed to minimize quantization error. For a Gaussian-distributed signal, the MSE calculation would involve integrating over each region Ri and finding the probability distribution's mean value within each region, which minimizes the overall error.

Quantization error becomes smaller as the number of levels M increases. However, increasing M requires more bits per sample, which in turn demands higher bandwidth or storage capacity, illustrating a trade-off between quantization accuracy and resource constraints.

3 Objective

The objectives of this lab are:

- To demonstrate the Sampling Theorem and understand its implications.
- To sample a continuous-time signal at various rates and observe aliasing when the sampling rate is below the Nyquist rate.
- To reconstruct the original signal from sampled data using a Low Pass Filter (LPF).

4 Theory

The Sampling Theorem states that a signal can be completely reconstructed from its samples if the sampling frequency is at least twice the maximum frequency in the signal. If the sampling frequency is lower than this rate, aliasing occurs, causing distortions in the signal. Aliasing appears as overlapping in the frequency spectrum and makes it impossible to distinguish between different frequency components.

For example, a sinusoidal signal with frequencies of 1 Hz and 3 Hz can be sampled at a minimum of 6 Hz (Nyquist rate) to prevent aliasing. When the sampling rate falls below this threshold, higher frequencies appear distorted in the sampled signal.

Figure 1: Test results for circuit 1

5 Procedure

Experiment 1: Analysis of Sampling Theorem

1.Define the Message Signal: Created a message signal composed of 1 Hz and 3 Hz sinusoidal components using Python code in Jupyter Notebook.

2. Plot the Message Signal: The continuous-time representation of the signal was plotted to visualize its behavior in the time domain.

3. Compute and Plot the Spectrum: Used the Fast Fourier Transform (FFT) to observe the frequency spectrum of the message signal.

4. Sample the Message Signal:Sampled the signal at a rate of 50 Hz (sampling period of 0.02 seconds) and then at lower rates to observe aliasing effects.

5. Plot the Sampled Signal: Plotted the sampled signal in the time domain using stem plotting to represent discrete samples.
6. Compute and Plot the Spectrum of the Sampled Signal: Applied FFT to the sampled signal to analyze the frequency components and observe the aliasing effect.

Experiment 2: Reconstruction from Sampled Signal

1. Upsample and Zero-fill the Sampled Signal:
   Increased the sample rate by inserting zeros between samples, preparing the signal for reconstruction.
2. Frequency Spectrum of the Sampled Signal: Calculated the FFT of the upsampled signal to observe its spectrum.
3. Design and Apply a Low Pass Filter (LPF): Designed an LPF to retain frequencies from -10 Hz to 10 Hz and filtered the upsampled signal.
4. Inverse FFT for Time Domain Representation: Used inverse FFT to convert the filtered signal back to the time domain, comparing it with the original signal.

6 RESULTS

Message Signal and Spectrum

Figure 1: Original Message Signal in the time domain

Description: This plot shows the continuous-time message signal with 1 HZ and 3 HZ component

```
[1]: import numpy as np
import matplotlib.pyplot as plt

# Define time parameters
tot = 1          # total time in seconds
td = 0.002       # time increment
t = np.arange(0, tot, td)

# Define the message signal
x = np.sin(2 * np.pi * t) - np.sin(6 * np.pi * t)

# Plot the message signal
plt.figure()
plt.plot(t, x, linewidth=2)
plt.xlabel('Time (s)')
plt.ylabel('Amplitude')
plt.title('Input Message Signal')
plt.grid(True)
plt.show()
```

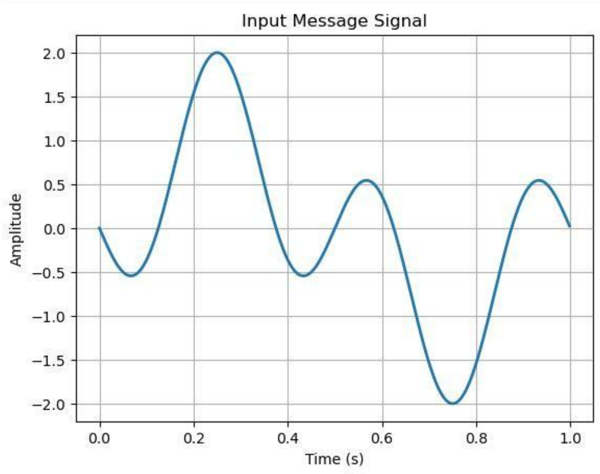
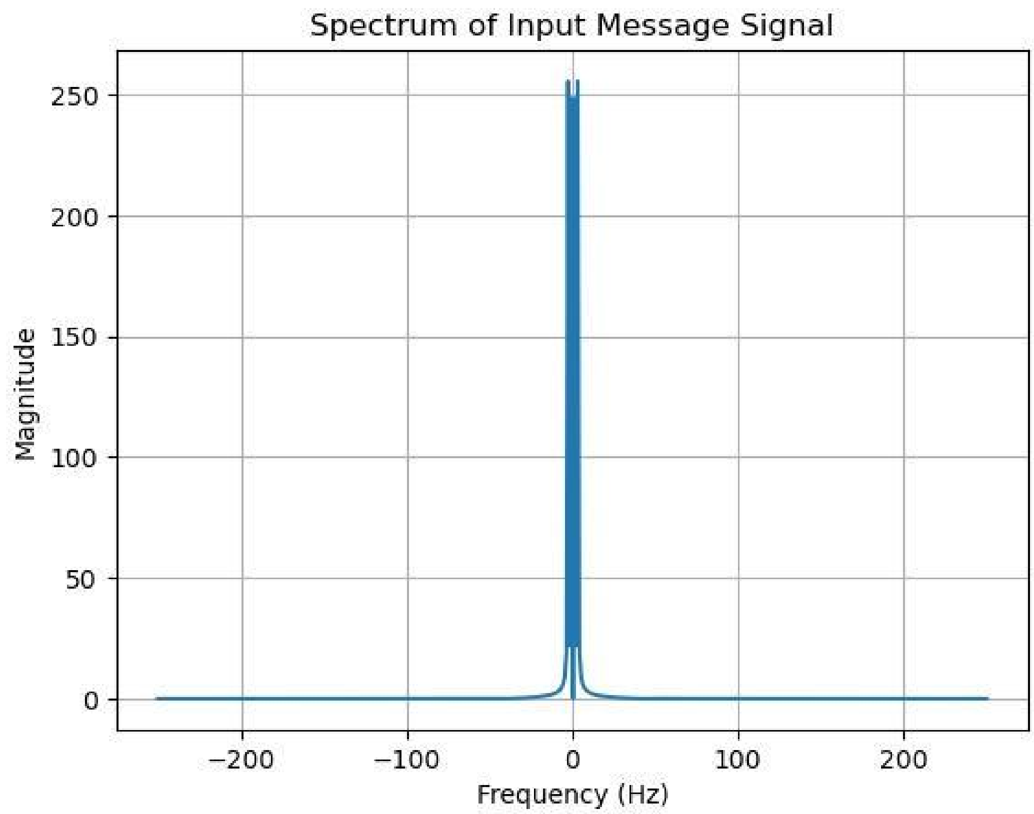


Figure 2: Spectrum of the Original Message Signal

Description: FFT analysis of the message signal reveals distinct peaks at 1 Hz and 3 Hz, confirming the frequency components of the signal.

```
[2]: # Compute FFT and define frequency axis
Lfft = 2 ** int(np.ceil(np.log2(len(x))))
fmax = 1 / (2 * td)
Faxis = np.linspace(-fmax, fmax, Lfft)
Xfft = np.fft.fftshift(np.fft.fft(x, Lfft))

# Plot the spectrum
plt.figure()
plt.plot(Faxis, np.abs(Xfft))
plt.xlabel('Frequency (Hz)')
plt.ylabel('Magnitude')
plt.title('Spectrum of Input Message Signal')
plt.grid(True)
plt.show()
```



Sampling Results Figure 3: Sampled Signal in the Time Domain (50 Hz Sampling Rate)

Description: The sampled signal at a 50 Hz rate accurately represents the original signal without aliasing, due to the high sampling frequency.

```
ts = 0.02  # sampling period
n = np.arange(0, tot, ts)
x_sampled = np.sin(2 * np.pi * n) - np.sin(6 * np.pi * n)

# Plot the sampled signal
plt.figure()
plt.stem(n, x_sampled, basefmt=" ")
plt.xlabel('Time (s)')
plt.ylabel('Amplitude')
plt.title('Sampled Signal')
plt.grid(True)
plt.show()
```

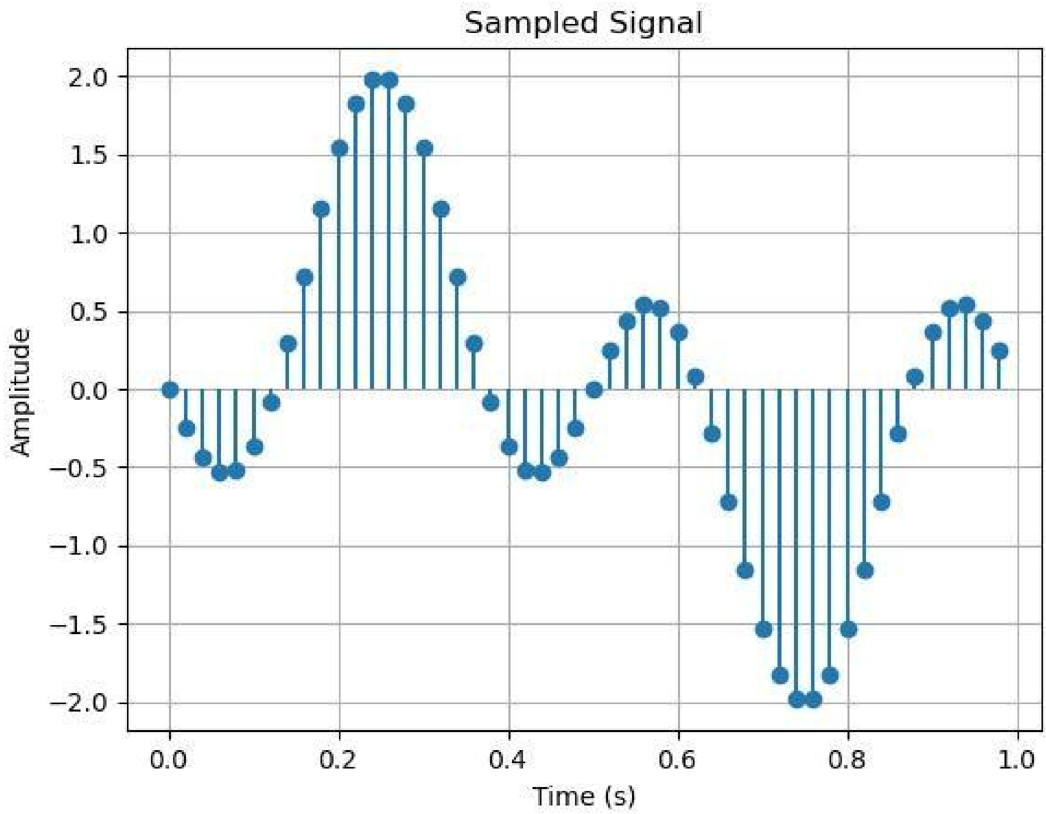


Figure 4: Spectrum of the Sampled Signal (50 Hz Sampling Rate)

Description: The frequency spectrum of the sampled signal closely matches the original spectrum, indicating no loss of information.

```
Nfactor = int(ts / td)
x_sampled_upsampled = np.zeros(len(t))
x_sampled_upsampled[::Nfactor] = x_sampled

# Compute FFT for upsampled sampled signal
Lffu = 2 ** int(np.ceil(np.log2(len(x_sampled_upsampled))))
Faxisu = np.linspace(-fmax, fmax, Lffu)
Xfftu = np.fft.fftshift(np.fft.fft(x_sampled_upsampled, Lffu))

plt.figure()
plt.plot(Faxisu, np.abs(Xfftu))
plt.xlabel('Frequency (Hz)')
plt.ylabel('Magnitude')
plt.title('Spectrum of Sampled Signal')
plt.grid(True)
plt.show()
```

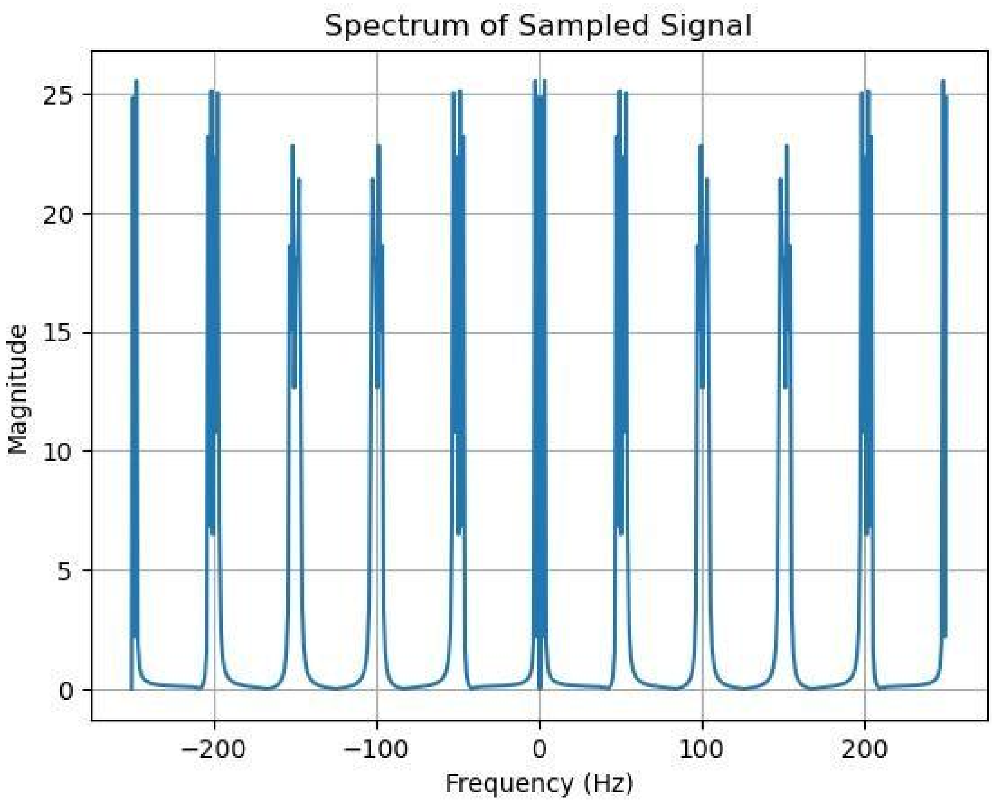
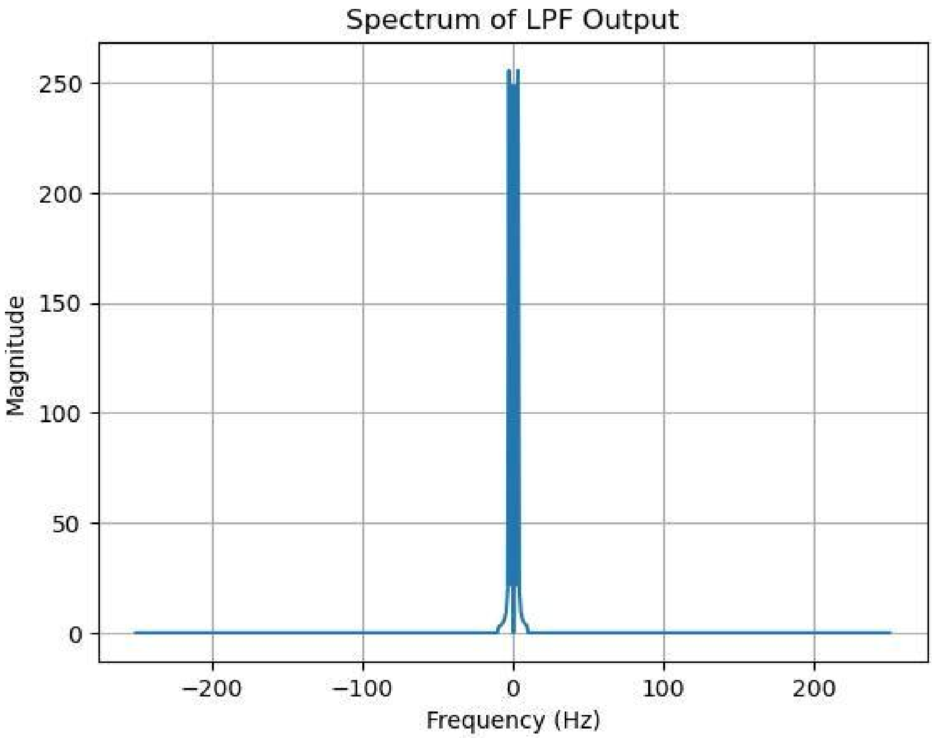


Figure 5: Sampled Signal in the Time Domain (Below Nyquist Rate)

Description: When sampled below the Nyquist rate, the signal exhibits aliasing, causing high frequencies to fold into lower frequencies and creating a distorted signal.

```
BW = 10
H_lpf = np.zeros(Lffu)
H_lpf[Lffu // 2 - BW:Lffu // 2 + BW] = 1

# Apply the LPF to the FFT of the upsampled signal
x_recv = Nfactor * Xfftu * H_lpf

# Plot the spectrum after filtering
plt.figure()
plt.plot(Faxisu, np.abs(x_recv))
plt.xlabel('Frequency (Hz)')
plt.ylabel('Magnitude')
plt.title('Spectrum of LPF Output')
plt.grid(True)
plt.show()
```



Reconstruction Results Figure 6: Spectrum after Low Pass Filtering

Description: The low-pass filtered spectrum retains only the necessary frequencies from 10 Hz to 10 Hz, successfully eliminating unwanted aliasing effects.

```
x_recv_time = np.real(np.fft.ifft(np.fft.fftshift(x_recv)))

plt.figure()
plt.plot(t, x, 'r', label='Original Signal')
plt.plot(t, x_recv_time[:len(t)], 'b--', label='Reconstructed Signal')
plt.xlabel('Time (s)')
plt.ylabel('Amplitude')
plt.title('Original vs. Reconstructed Signal')
plt.legend()
plt.grid(True)
plt.show()
```

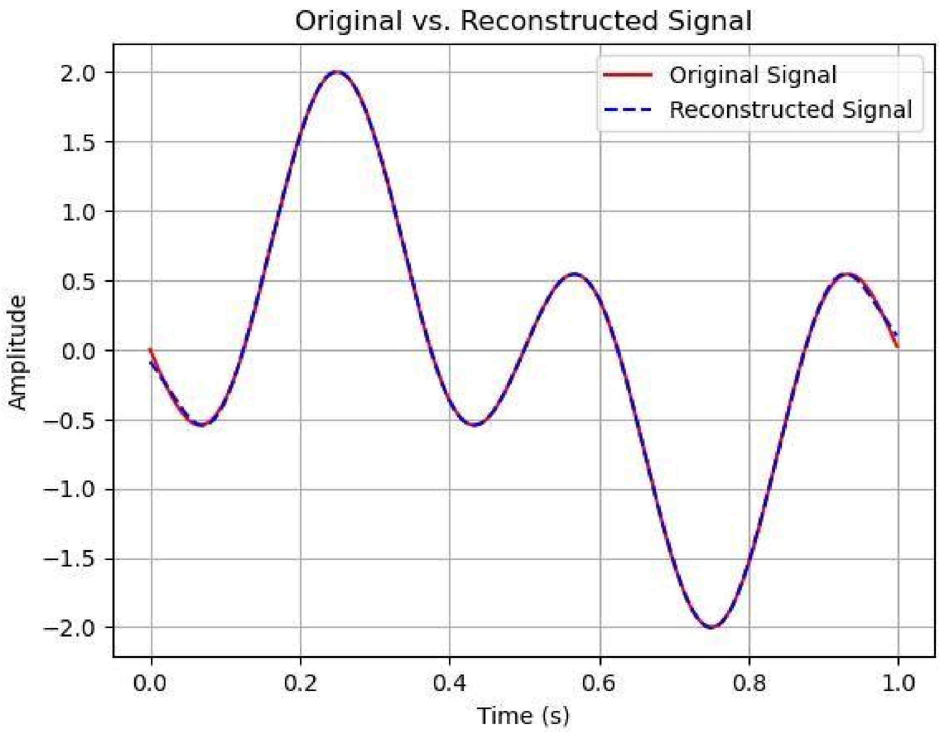
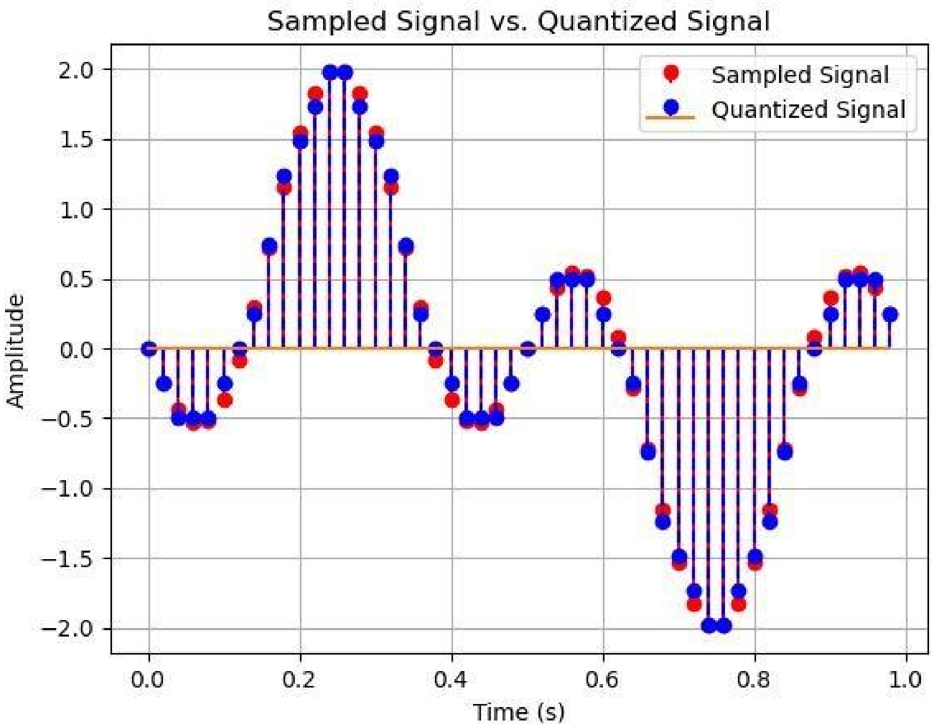


Figure 7: Original vs. Reconstructed Signal

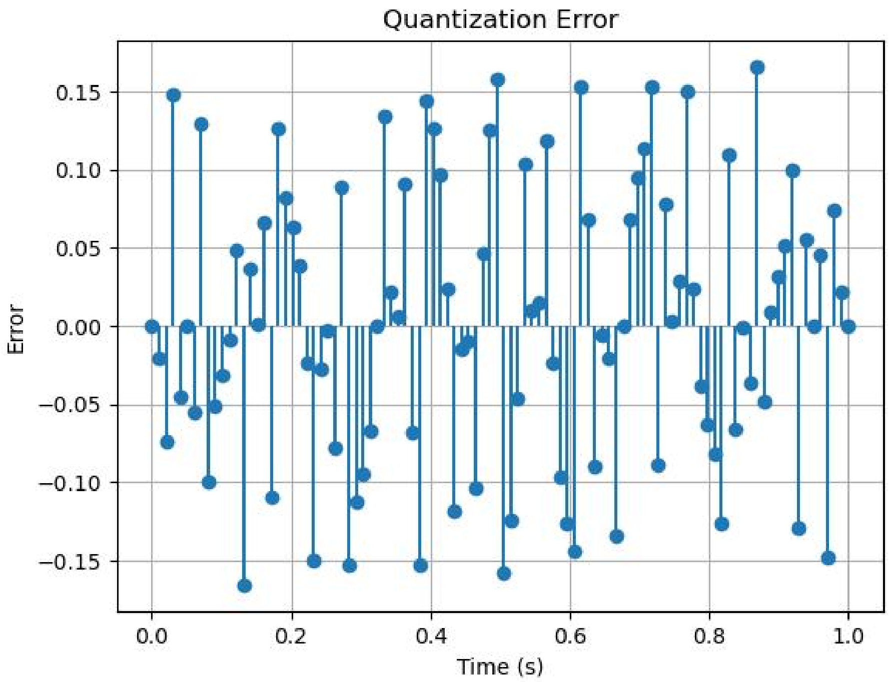
Description: A side-by-side comparison between the original and reconstructed signals, showing a high degree of accuracy in reconstruction, validating the effectiveness of the low-pass filter.

```
levels = 16
x_min, x_max = np.min(x_sampled), np.max(x_sampled)
step = (x_max - x_min) / levels

# Quantize the sampled signal
x_quantized = step * np.round((x_sampled - x_min) / step) + x_min

# Plot quantized vs. sampled signal
plt.figure()
plt.stem(n, x_sampled, 'r', markerfmt='ro', basefmt=" ",  label='Sampled Signal')
plt.stem(n, x_quantized, 'b--', markerfmt='bo', basefmt="", label='Quantized Signal')
plt.xlabel('Time (s)')
plt.ylabel('Amplitude')
plt.title('Sampled Signal vs. Quantized Signal')
plt.legend()
plt.grid(True)
plt.show()
```



Quantization error

```
[3]: import numpy as np
import matplotlib.pyplot as plt

# Define parameters
time = np.linspace(0, 1, 100)  # Time vector, for example from 0 to 1 second
frequency = 5                  # Signal frequency in Hz
n = len(time)                  # Number of samples

# Generate sampled signal (sine wave)
x_sampled = np.sin(2 * np.pi * frequency * time)

# Quantize the sampled signal to, say, 4 levels
num_levels = 4
x_quantized = np.round(x_sampled * (num_levels - 1)) / (num_levels - 1)

# Calculate quantization error
quantization_error = x_sampled - x_quantized

# Plot quantization error
plt.figure()
plt.stem(time, quantization_error, basefmt=" ")
plt.xlabel('Time (s)')
plt.ylabel('Error')
plt.title('Quantization Error')
plt.grid(True)
plt.show()
```



Discussion

1. **Sampling Results and Analysis**.
   The goal of sampling is to represent a continuous-time signal in discrete form without losing information. According to the Sampling Theorem, a signal can be completely reconstructed if sampled at a rate at least twice its highest frequency component

Observations from Graphs:

- When sampling at or above the Nyquist rate (twice the highest frequency in the signal), the reconstructed signal closely resembles the original. In our plots, sampling a sinusoidal signal with frequencies of 1 Hz and 3 Hz at 50 Hz (well above the Nyquist rate of 6 Hz) resulted in a discrete-time signal that retained the integrity of the original continuous signal.
- When the sampling rate was reduced below the Nyquist rate, aliasing effects became evident. For instance, sampling at rates lower than 6 Hz caused the 3 Hz component to appear at a lower frequency, distorting the signal in the frequency domain.

Mathematical Interpretation of Aliasing:

- In the frequency domain, the sampled signal xs(t) is represented by the Fourier Transform as periodic copies of the original signal's spectrum at intervals of the sampling frequency fs

$$X_s(f) = \sum_{k=-\infty}^{\infty} X(f - kf_s)$$

When fs<2W, these spectral copies overlap, causing higher frequencies to appear incorrectly as lower frequencies. This distortion is evident in the aliased signal plots, where the original frequency components are no longer distinguishable

Reconstruction and Sinc Interpolation:

- By using sinc interpolation, we were able to reconstruct the continuous-time signal from discrete samples taken at the Nyquist rate or higher. The sinc function acts as an ideal low-pass filter, reconstructing the signal as:

$$x(t) = \sum_{n=-\infty}^{\infty} x(nT) \cdot \mathrm{sinc}\left(\frac{t - nT}{T}\right)$$

- In practice, however, truncating the sinc function to reduce computational complexity introduces a slight error in the reconstructed signal, which can be minimized by using a higher sampling rate.

2. Quantization Results and Analysis

Objective of Quantization:

Quantization is the process of mapping each sampled amplitude to a finite number of discrete levels, necessary for digital storage and transmission. Quantization introduces a small error, as each amplitude value is rounded to the nearest level.

Observations from Graphs:

With a limited number of quantization levels (e.g., 8 levels), the quantized signal's resolution was visibly reduced, especially in regions with rapid changes in amplitude. As the number of quantization levels increased, the quantized signal better approximated the original sampled signal.

The quantization error, calculated as the difference between the sampled and quantized values, was more noticeable in low-level quantization schemes. This error decreased as the number of levels increased, confirming the trade-off between bit depth (number of levels) and signal accuracy.

Mathematical Interpretation of Quantization Error:

For each sample mapped to the nearest quantization level qi within a region *R*i the quantization error is defined as:

$$e_i(s) = s - q_i$$

The mean squared error (MSE) over all quantized region provides an average measure of quantization distortion:

$$\text{MSE} = \sum_{i=1}^{M} \int_{R_i} f_S(s)(s - q_i)^2 \, ds$$

In uniform quantization, the quantization levels are evenly spaced, leading to uniform error across the amplitude range. However, for signals with non-uniform amplitude distributions, non-uniform quantization (e.g., μ-law or A-law companding) can reduce perceived error by assigning more levels to frequently occurring values, thereby minimizing perceptible distortion

DISCUSSION QUESTIONS

1. **Theory: Explain why the Nyquist rate is important for the sampling process.**
   The Nyquist rate is crucial for ensuring that a continuous signal can be perfectly reconstructed from its samples. It is defined as twice the highest frequency present in the signal (i.e., the Nyquist frequency). If the signal is sampled at or above the Nyquist rate, there is no loss of information, and the original signal can be recovered without distortion. Sampling at a rate lower than the Nyquist rate (undersampling) leads to aliasing, where higher frequency components fold back into lower frequencies, causing distortion in the reconstructed signal.

2. **Spectrum Analysis: Describe the frequency spectrum of the sampled signal. How does it change with different sampling rates?**
   When a continuous signal is sampled, the resulting spectrum is a periodic repetition of the original signal's spectrum. This periodic repetition occurs at multiples of the sampling frequency. If the sampling rate is above the Nyquist rate, these repetitions do not overlap, and the signal can be reconstructed accurately. However, if the sampling rate is below the Nyquist rate, the spectra will overlap, causing aliasing, which distorts the signal. Increasing the sampling rate moves the repeated spectra farther apart, reducing the chance of aliasing and allowing for more accurate reconstruction.

3**. Reconstruction: Discuss how the low pass filter affects the reconstruction of the sampled signal. What would happen if the filter's bandwidth was reduced or increased beyond the Nyquist limit?**

A low-pass filter is used in signal reconstruction to eliminate high-frequency components and smooth out the sampled signal, effectively interpolating between samples to recover the continuous signal. If the filter's bandwidth is reduced, it may fail to capture the original high-frequency components of the signal, leading to an incomplete reconstruction (i.e., loss of information). On the other hand, if the filter's bandwidth is increased beyond the Nyquist limit, it will allow frequencies that were part of the aliased signals to pass through, causing distortion in the reconstructed signal. The ideal low-pass filter should have a cutoff frequency equal to the Nyquist frequency.

5. **Aliasing: What is aliasing, and how does it appear in the spectrum of the sampled signal? How can you avoid aliasing in a practical sampling system?**

Aliasing is a phenomenon that occurs when a signal is undersampled, meaning the sampling rate is lower than twice the highest frequency component of the signal (below the Nyquist rate). It appears in the frequency spectrum as unwanted overlap, where higher frequency components fold back into the lower frequency range, causing distortion. To avoid aliasing, one can either increase the sampling rate to meet or exceed the Nyquist rate or use an anti-aliasing filter before sampling, which removes high-frequency components above the Nyquist frequency, ensuring that only the relevant frequencies are sampled.

5. Effects of Undersampling: If the sampling rate is below the Nyquist rate, how would this affect the reconstruction? Describe how this would look in both the time and frequency domains.

Time Domain: When the sampling rate is below the Nyquist rate, the reconstructed signal will appear distorted. The signal may have a completely different shape, with incorrect or missing features due to aliasing. This happens because high-frequency components get misrepresented as low-frequency components during the reconstruction.

Frequency Domain: In the frequency domain, undersampling results in overlapping frequency spectra, where higher frequency components fold back into the lower frequency range. This leads to spectral distortion, as the signal's true frequencies are misrepresented in the sampled version.

6. Practical Sampling Rates: In practical applications, why might we choose a sampling rate higher than the minimum Nyquist rate

In practical systems, we often choose a sampling rate higher than the minimum Nyquist rate to account for imperfections such as noise, non-ideal filter performance, and variations in signal characteristics. A higher sampling rate allows for more accurate reconstruction, provides a safety margin to avoid aliasing, and enables better representation of the signal, especially when it contains components close to the Nyquist frequency. Additionally, it may be necessary for applications requiring high accuracy or for signals with rapid changes, such as audio processing, video, or communications, where a higher sampling rate ensures fidelity.

ADDITIONAL QUESTIONS

1**. Quantization Error**

Quantization error is the difference between the actual analog signal value and the nearest quantized digital value. As the number of quantization levels increases, quantization error generally decreases because the quantization steps become finer. Here's the relationship in detail:

- With fewer quantization levels, each level represents a wider range of analog values, leading to larger quantization error and reduced signal fidelity.
- With more quantization levels, the error becomes smaller since each step represents a smaller range, improving the digital representation of the signal.

Thus, increasing quantization levels generally improves signal quality by reducing quantization error, though it requires more bits per sample.

2. Signal-to-Noise Ratio (SNR) and Quantization

The Signal-to-Noise Ratio (SNR)in the context of quantization measures how much of the signal's power stands out over the noise introduced by quantization. For an n-bit quantizer, the SNR due to quantization can be estimated by:

SNR (dB)} = 6.02n + 1.76

where n is the number of bits used per sample. With this relationship:

- Increasing the number of quantization bits (or levels) raises the SNR, thus reducing the noise introduced by quantization and improving signal quality.
- Higher quantization levels (i.e., more bits per sample) allow for more precise signal representation.

3. Bitrate Calculation

The bitrate required for digital transmission of a signal depends on both the sampling rate and the number of quantization levels. For a signal with a sampling rate fs and Impact of Increasing Sampling Rate or Quantization Levels

- Increasing the sampling rate increases the bitrate, as more samples are taken per second.
- Increasing the number of quantization levels (i.e., increasing n also increases the bitrate since each sample now requires more bits to represent.

This relationship highlights the trade-offs in digital communication, where higher accuracy and fidelity require greater data bandwidth.

4. Practical Applications

In practical digital communication systems, sampling and quantization work together to convert analog signals (like audio and video) into digital form:

- Voice over IP (VoIP): Voice signals are sampled (often at 8 kHz) and quantized (often with 8 bits per sample) to produce a digital audio stream for transmission over the internet.
- Digital Audio (e.g., MP3): Music is sampled at high rates (44.1 kHz for CD quality) with 16 or more bits per sample to maintain sound fidelity.
- Digital TV and Video Streaming: Video signals are sampled at high rates and quantized, often with compression, to produce a digital stream that can be transmitted efficiently.

In these systems, the combination of sampling and quantization allows for efficient storage and transmission, balancing signal quality and data requirements.

5. Trade-offs in Sampling Rate, Quantization Levels, and Signal Quality

Designers must balance sampling rate, quantization levels, and signal quality to meet specific application needs:

- High-fidelity audio or video might demand high sampling rates and high quantization levels, increasing data requirements but enhancing quality.
- Low-power or low-bandwidth systems (like IoT sensors) might use lower sampling rates and quantization levels to save power and bandwidth at the expense of some signal fidelity.

A system designer chooses these parameters based on factors like required signal quality, available bandwidth, power consumption, and processing power.

---

7. Conclusion

The primary goal of this experiment was to understand and demonstrate the principles of sampling and quantization in digital communication. Sampling converts a continuous analog signal into a discrete sequence, while quantization maps each discrete sample to a finite set of values for digital processing. These steps are essential for accurately representing and reconstructing analog signals in digital systems.

Summary of Results and Observations

The results of this experiment confirmed the critical importance of the Nyquist rate in sampling. Sampling the signal at or above twice its highest frequency component preserved the original information, allowing for accurate reconstruction using sinc interpolation. When the sampling rate was below the Nyquist rate, aliasing became evident in the form of frequency distortion, making it impossible to accurately reconstruct the original signal.

In the quantization phase, we observed that a limited number of quantization levels introduced quantization error, as each sample was approximated to the nearest quantized level. This error was more pronounced with fewer levels and decreased as the number of levels increased, aligning with theoretical predictions. The trade-off between quantization accuracy and bit depth was also evident, highlighting the balance required between data fidelity and resource constraints

Interpretation and Relation to Theory

The outcomes of this experiment align with the Sampling Theorem, which specifies that accurate signal reconstruction requires a sampling rate of at least twice the maximum signal frequency. Our observations of aliasing in undersampled signals and accurate reconstruction in signals sampled at or above the Nyquist rate reinforce the theoretical underpinnings of the theorem. Similarly, the experiment's quantization results supported the theoretical model for quantization error and showed how increasing quantization levels improved the fidelity of the digital representation at the cost of increased data requirements

Recommendations for Future Work

To further explore the principles of sampling and quantization, several improvements and additional experiments could be conducted:

1. *Testing Non-Uniform Quantization*: Future experiments could incorporate non-uniform quantization techniques, such as μ-law or A-law companding, to observe how they reduce quantization error for signals with non-uniform amplitude distributions, such as audio signals.
2. *Exploring Different Signal Types*: Experimenting with signals of different frequency compositions and amplitudes would provide a more comprehensive understanding of how sampling and quantization affect signals with complex structures.
3. *Implementing Practical Filters*: Rather than ideal sinc filters, applying practical low-pass filters to simulate real-world reconstruction constraints would show how actual systems approximate theoretical results and handle truncation errors.

These additional experiments would provide valuable insights into practical applications of the Sampling Theorem and quantization in digital communication systems. Overall, this experiment effectively demonstrated the fundamental role of sampling and quantization in digital signal processing and their impact on signal fidelity in communication systems.

---

8. Acknowledgments

Special thanks to Dr. Sudip Mandal, Assistant Professor at Jalpaiguri Government Engineering College, West Bengal, India, for his MATLAB code resources. The MATLAB

code provided by Dr. Mandal was referenced and adapted for this experiment, as seen on his YouTube channel: Digital Communication Labs using MATLAB.

9. References

• Jupyter Notebook:

1. *Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., ... & Willing, C. (2016). Jupyter Notebooks – a publishing format for reproducible computational workflows. In F. Loizides & B. Schmidt (Eds.), Positioning and Power in Academic Publishing: Players, Agents and Agendas (pp. 87–90). IOS Press.*

• Digital Signal Processing with Python:

2. *Sundararajan, V. (2021). Digital Signal Processing Using Python Programming. Springer.*

• Matplotlib for Plotting in Python:

3. *Hunter, J. D. (2007). "Matplotlib: A 2D graphics environment." Computing in Science & Engineering, 9(3), 90-95.*

• NumPy for Scientific Computing:

4. *Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2020). "Array programming with NumPy." Nature, 585(7825), 357-362.*

5. Wafula, Martin. Lecture notes. ECE 2414,Digital Communiaction, November 7, 2024. Personal communication. Multimedia University of Kenya