Sampling and Quantization

Laboratory Report

Prepared by:
ENG-219-058/2021

Lecturer:
Mr Martin Wafula

Multimedia Universirty of Kenya
Electrical Engineering

November 30, 2024

1. Abstract

This report investigates the Sampling Theorem, which states that a continuous-time signal can be accurately reconstructed from its samples as long as the sampling frequency is at least double the highest frequency component of the signal, known as the Nyquist rate. The study provides a hands-on examination of this theorem by sampling and reconstructing signals in both time and frequency domains. Utilizing Matlab, experiments were conducted to perform signal sampling, analyze frequency spectra, and reconstruct signals, allowing for an evaluation of how varying sampling rates impact the overall results.

2. Introduction

To give a thorough overview of Sampling Theory and Quantization, I will delve into the mathematical principles underlying these concepts, including the application of Fourier analysis to the Sampling Theorem and the analysis of quantization errors. This detailed explanation will illustrate how these mathematical concepts form the basis of digital communication.

3. Objective

- To analyze and verify the Sampling Theorem.
- To reconstruct the original signal from sampled data.
- To perform quantization.

4. Theoretical Background

Sampling Theorem

The Sampling Theorem ensures that a continuous-time signal can be represented accurately in discrete form if sampled at a rate greater than or equal to twice its highest frequency. This theorem forms the basis for digital signal processing, enabling the conversion of analog signals into digital form for storage and analysis without losing crucial information.

Mathematical Formulations for Sampling

1. Sampling a Continuous-Time Signal

To sample a continuous-time signal $x(t)$ with frequency content up to W, we multiply it by an impulse train $\delta(t - nT)$, where each impulse marks a sampling point:

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

where $T = \frac{1}{2W}$ is the sampling period, ensuring that we sample at a rate high enough to capture the signal's content.

2. Frequency Domain Representation of the Sampled Signal

Sampling in the time domain causes the signal's frequency spectrum to repeat periodically. The Fourier transform of the sampled signal $x_s(t)$, denoted $X_s(f)$, is given by:

$$X_s(f) = \mathcal{F}\{x(t) \cdot \sum_{n=-\infty}^{\infty} \delta(t - nT)\}$$

The Fourier transform of an impulse train $\sum_{n=-\infty}^{\infty} \delta(t - nT)$ is another impulse train in the frequency domain:

$$\mathcal{F}\left\{\sum_{n=-\infty}^{\infty} \delta(t - nT)\right\} = \frac{1}{T} \sum_{k=-\infty}^{\infty} \delta\left(f - \frac{k}{T}\right)$$

So, the Fourier transform of the sampled signal becomes:

$$X_s(f) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X\left(f - \frac{k}{T}\right)$$

where $X(f)$ is the Fourier transform of the original signal $x(t)$.

3. Preventing Aliasing

To prevent aliasing (where frequency components overlap and distort the signal), the sampling frequency f_s must be at least twice the highest frequency W in the original signal:

$$f_s = \frac{1}{T} \geq 2W$$

This ensures that the repeated copies of the spectrum in $X_s(f)$ do not o

These formulations describe the relationship between the continuous-time signal, its sampled version, and the resulting frequency domain representation, including the conditions necessary to prevent aliasing. The Sampling Theorem can be mathematically described as follows:

$$f_s \geq 2B$$

where:

- f_s: Sampling frequency (samples per second).
- B: The highest frequency component of the continuous-time signal.

When a continuous-time signal $x(t)$ is sampled at a rate f_s, the discrete signal $x[n]$ is obtained by:

$$x[n] = x(nT), \quad \text{where} \quad T = \frac{1}{f_s}$$

Reconstruction of Sampled Signals

To perfectly reconstruct the signal from its samples, f_s must be at least twice the signal's bandwidth. If the sampling rate is lower than the Nyquist rate, aliasing occurs, distorting the reconstructed signal.

The reconstruction of a sampled signal relies on the ideal interpolation formula:

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

where:

- $x(t)$: The reconstructed continuous-time signal.
- $x[n]$: The discrete samples of the signal.
- T: The sampling period, related to the sampling frequency by $f_s = \frac{1}{T}$.
- $\text{sinc}(t) = \frac{\sin(\pi t)}{\pi t}$: The sinc function, acting as an ideal low-pass filter.

In practice, sinc interpolation is often approximated by simpler methods like linear or spline interpolation, though sinc interpolation provides theoretically perfect reconstruction when sampling above the Nyquist rate.

Quantization for Analog Sequence Sources

In digital communication, after sampling a continuous-time signal, the next step is quantization. This process converts each sample into one of a finite set of discrete values, known as quantization levels. Quantization is essential because, while sampled signals can take any real value, digital systems can only represent and process finite, discrete levels.

Quantization Process and Levels

1. **Defining Quantization Regions:** The range of sampled values is divided into quantization regions R_i, where each region is mapped to a corresponding quantized value or level q_i.

2. **Mapping to the Nearest Quantization Level:** Each sample s within a region R_i is approximated by the quantization level q_i that is closest to s. Mathematically, the quantized value q_i is chosen such that:

Quantization Error

For a sample s within a quantization region R_i, the quantization error $e_i(s)$ is defined as:

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

The mean squared quantization error (MSE) is used to evaluate the accuracy of quantization across all quantization levels. If $f_S(s)$ represents the probability density function of the source signal S, then the MSE over all quantization levels is:

$$\text{MSE} = \int_{-\infty}^{\infty} e^2(s) \cdot f_S(s)\, ds$$

This expression calculates the expected squared error and serves as a measure of the distortion introduced by quantization.

5. Methodology

5.1 Sampling Theorem

1. Define the message signal with 1 Hz and 3 Hz sinusoidal components.
2. Plot the message signal in the time domain.
3. Compute and plot its frequency spectrum using FFT.
4. Sample the signal with a sampling period, e.g., 0.02 seconds (50 Hz).
5. Plot the sampled signal in the discrete-time domain.
6. Compute and plot the spectrum of the sampled signal.

MATLAB Code

Below is a simplified version of the MATLAB code used to verify the Sampling Theorem:

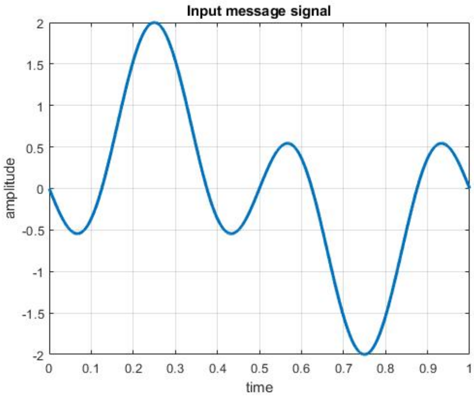
Listing 1: Analysis of Sampling Theorem

```
%% Copyright @ Dr Sudip Mandal

%% ANALYSIS OF SAMPLING THEOREM

clear all;
close all;
clc;

% Define the message signal
tot = 1;
td = 0.002;
t = 0:td:tot;
L = length(t);
x = sin(2*pi*t) - sin(6*pi*t);

% Plot the message signal in time domain
figure(1);
plot(t, x, 'linewidth', 2);
xlabel('time'); ylabel('amplitude');
grid;
title('Input message signal');
```

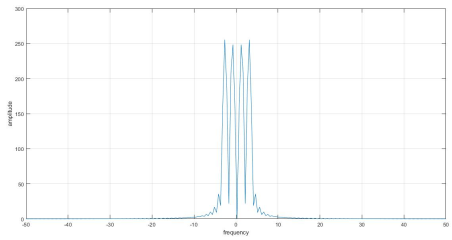
```
% Plot the signal in frequency domain
Lf = length(x);
Lfft = 2^ceil(log2(Lf) + 1);
fmax = 1 / (2 * td);
Faxis = linspace(-fmax, fmax, Lfft);
xfft = fftshift(fft(x, Lfft));

figure(2);
plot(Faxis, abs(xfft));
xlabel('frequency'); ylabel('amplitude');
axis([-50 50 0 300]);
grid;

% Sample the message signal
ts = 0.02;
Nfactor = round(ts / td);
xsm = downsample(x, Nfactor);
tsm = 0:ts:tot;

% Plot the sampled signal (discrete time version)
figure(3);
stem(tsm, xsm, 'linewidth', 2);
xlabel('time'); ylabel('amplitude');
grid;
title('Sampled Signal');

% Compute the spectrum of sampled signal
xsmu = upsample(xsm, Nfactor);
Lfu = length(xsmu);
Lffu = 2^ceil(log2(Lfu) + 1);
fmaxu = 1 / (2 * td);
Faxisu = linspace(-fmaxu, fmaxu, Lffu);
xfftu = fftshift(fft(xsmu, Lffu));

% Plot the spectrum of the sampled signal
figure(4);
plot(Faxisu, abs(xfftu));
xlabel('frequency'); ylabel('amplitude');
title('Spectrum of Sampled signal');
grid;
```

5.2 Reconstruction from the Sampled Signal

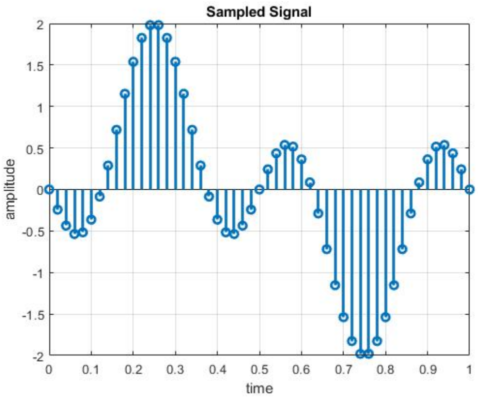
1. Define parameters and generate the signal.
2. Upsample the sampled signal by inserting zeros.
3. Analyze the frequency spectrum of the upsampled signal using FFT.



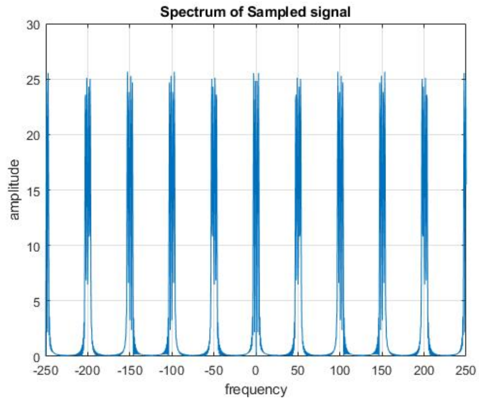
(a) Input message signal



(b) Fast Fourier Transform



(c) Sampled signal



(d) Spectrum of Sampled signal

Figure 1: Sampling Theorem

4. Design a low-pass filter to retain frequencies between $-10\,\mathrm{Hz}$ and $10\,\mathrm{Hz}$.
5. Filter the upsampled signal using the LPF.
6. Apply inverse FFT to convert the filtered signal to the time domain and compare it with the original signal.

MATLAB Code

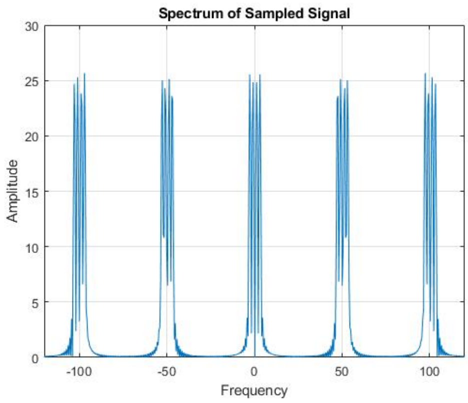
The following MATLAB code is used to implement the reconstruction process:

Listing 2: Reconstruction from Sampled Signal

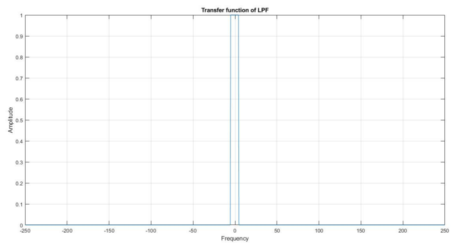
```
%% Copyright @ Dr Sudip Mandal

%% Reconstruction from Sampled Signal

clear all;
close all;
clc;

% Define Parameters and Generate Signal
tot = 1;
```

```
td = 0.002;
t = 0:td:tot;
L = length(t);
x = sin(2*pi*t) - sin(6*pi*t);

ts = 0.02;

% Upsample and zero fill the sampled signal
Nfactor = round(ts / td);

xsm = downsample(x, Nfactor);

xsmu = upsample(xsm, Nfactor);

% Frequency Spectrum of Sampled Signal
Lfu = length(xsmu);
Lffu = 2^ceil(log2(Lfu) + 1);
fmaxu = 1 / (2 * td);
Faxisu = linspace(-fmaxu, fmaxu, Lffu);
xfftu = fftshift(fft(xsmu, Lffu));

% Plot the spectrum of the Sampled Signal
figure(1);
plot(Faxisu, abs(xfftu));
xlabel('Frequency'); ylabel('Amplitude');
axis([-120 120 0 300 / Nfactor]);
title('Spectrum of Sampled Signal');
grid;

% Design a Low Pass Filter
BW = 10;
H_lpf = zeros(1, Lffu);
H_lpf(Lffu / 2 - BW : Lffu / 2 + BW - 1) = 1;

figure(2);
plot(Faxisu, H_lpf);
xlabel('Frequency'); ylabel('Amplitude');
title('Transfer function of LPF');
grid;

% Filter the Sampled Signal
x_recv = Nfactor * (xfftu) .* H_lpf;

figure(3);
plot(Faxisu, abs(x_recv));
xlabel('Frequency'); ylabel('Amplitude');
axis([-120 120 0 300]);
title('Spectrum of LPF output');
```

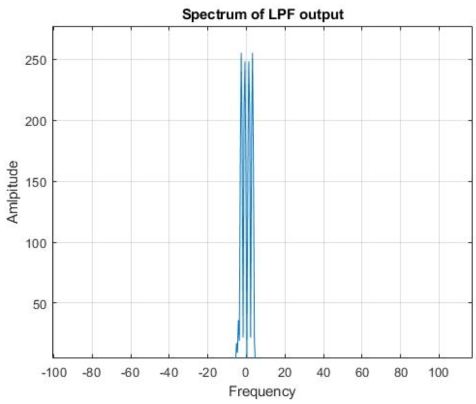
```
grid;

% Inverse FFT for Time domain representation
x_recv1 = real(ifft(fftshift(x_recv)));
x_recv2 = x_recv1(1:L);

figure(4);
plot(t, x, 'r', t, x_recv2, 'b--', 'linewidth', 2);
xlabel('Time'); ylabel('Amplitude');
title('Original vs. Reconstructed Message Signal');
grid;
```



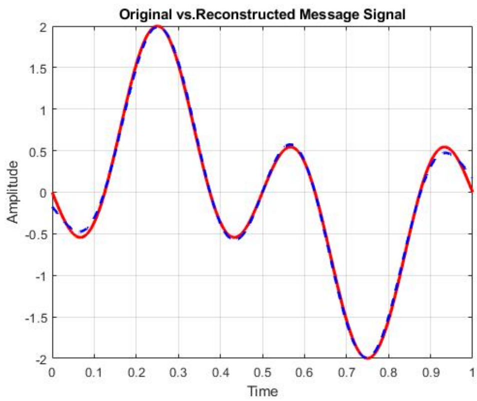
(a) Spectrum of Sampled Signal



(b) Transfer Function



(c) Spectrum of Low Pass Filter



(d) Output vs Original

Figure 2: Reconstruction of Sampled Signal

5.3 Quantization

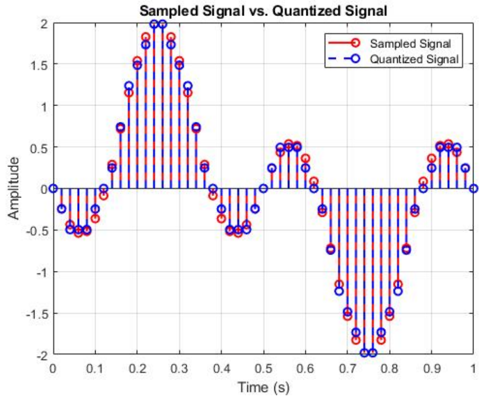
1. Define quantization levels (e.g., 8, 16, 32).
2. Quantize the signal by mapping samples to the nearest levels.
3. Plot the quantized signal and calculate quantization error.

MATLAB Code Outline

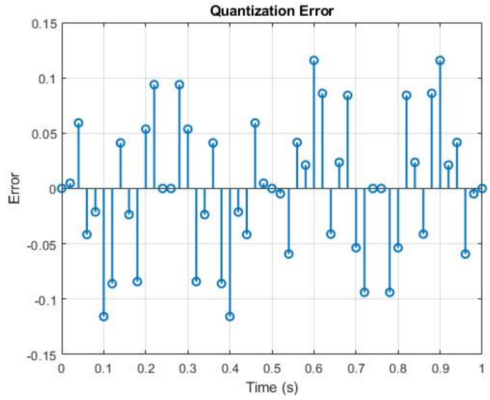
Listing 3: Quantization Process in MATLAB

```
%% Copyright @ Dr Sudip Mandal
%% Quantization

clear all;
close all;
clc;

% Define the message signal
tot = 1;
td = 0.002;
t = 0:td:tot;
L = length(t);
x = sin(2*pi*t) - sin(6*pi*t);

% Plot the message signal in time domain
figure(1);
plot(t, x, 'linewidth', 2);
xlabel('time'); ylabel('amplitude');
grid;
title('Input message signal');

% Plot the signal in frequency domain
Lf = length(x);
Lfft = 2^ceil(log2(Lf) + 1);
fmax = 1 / (2 * td);
Faxis = linspace(-fmax, fmax, Lfft);
xfft = fftshift(fft(x, Lfft));

figure(2);
plot(Faxis, abs(xfft));
xlabel('frequency'); ylabel('amplitude');
axis([-50 50 0 300]);
grid;

% Sample the message signal
ts = 0.02;
Nfactor = round(ts / td);
xsm = downsample(x, Nfactor);
tsm = 0:ts:tot;

% Plot the sampled signal (discrete time version)
figure(3);
stem(tsm, xsm, 'linewidth', 2);
xlabel('time'); ylabel('amplitude');
grid;
```

```
title('Sampled Signal');

% Compute the spectrum of sampled signal
xsmu = upsample(xsm, Nfactor);
Lfu = length(xsmu);
Lffu = 2^ceil(log2(Lfu) + 1);
fmaxu = 1 / (2 * td);
Faxisu = linspace(-fmaxu, fmaxu, Lffu);
xfftu = fftshift(fft(xsmu, Lffu));

% Plot the spectrum of the sampled signal
figure(4);
plot(Faxisu, abs(xfftu));
xlabel('frequency'); ylabel('amplitude');
title('Spectrum of Sampled signal');
grid;

% Quantization process
levels = 16;
x_min = min(xsm);
x_max = max(xsm);
step = (x_max - x_min) / levels;

% Quantize the sampled signal
x_quantized = step * round((xsm - x_min) / step) + x_min;

% Plot quantized vs. sampled signal
figure(5);
stem(tsm, xsm, 'r', 'LineWidth', 1.5); hold on;
stem(tsm, x_quantized, 'b--', 'LineWidth', 1.5);
xlabel('Time (s)');
ylabel('Amplitude');
title('Sampled Signal vs. Quantized Signal');
legend('Sampled Signal', 'Quantized Signal');
grid on;

% Quantization error
quantization_error = xsm - x_quantized;
figure(6);
stem(tsm, quantization_error, 'LineWidth', 1.5);
xlabel('Time (s)');
ylabel('Error');
title('Quantization Error');
grid on;
```



(a) Sampled vs Quantized



(b) Quantization Error

Figure 3: Quantization

6. Discussion

1. Sampling Results and Analysis

The main purpose of sampling is to convert a continuous-time analog signal into a discrete sequence form without any loss of information. From Nyquist-Shannon Sampling theorem, a signal can be completely reconstructed if sampled at or above the Nyquist rate.

Observations from Graphs

- **When sampling above the Nyquist rate** The reconstructed signal closely matches the original. In our graphs, when we sampled a sinusoidal signal with frequencies of 1 Hz and 3 Hz at 50 Hz (significantly above the Nyquist rate of 6 Hz), the resulting discrete-time signal preserved the characteristics 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.

2. Reconstruction and Sinc Interpolation

- **By using sinc interpolation,** we successfully reconstructed the continuous-time signal from discrete samples taken at the Nyquist rate or above. The sinc function serves as an ideal low-pass filter. However, in practice, truncating the sinc function to simplify computation introduces a small error in the reconstructed signal. This error can be minimized by using a higher sampling rate.

3.Quantization Results and Analysis

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

Observations from Graphs

When using a small number of quantization levels (e.g., 8 levels), the resolution of the quantized signal was noticeably lower, especially in regions with fast amplitude variations. As the number of quantization levels increased, the quantized signal more closely matched the original sampled signal. The quantization error, which is the difference between the sampled values and their quantized counterparts, was more significant with fewer levels. This error diminished as the number of levels grew, illustrating the trade-off between bit depth (number of levels) and signal accuracy.

Discussion Questions

1. Theory: Explain why the Nyquist rate is important for the sampling process.

The Nyquist rate is important in the sampling process because it ensures that a continuous signal can be accurately sampled and reconstructed without loss of information.The Nyquist-Shannon Sampling Theorem states that a continuous-time signal can be perfectly represented and reconstructed from its discrete samples if it is band- limited and the sampling frequency is at least twice the maximum frequency present in the signal.(the Nyquist rate) Sampling below the Nyquist rate can result to aliasing which causes overlapping of frequency components, leading to distortion and loss of the original signal's message.

2. Spectrum Analysis: Describe the frequency spectrum of the sampled signal. How does it change with different sampling rates?

The definition of spectrum in the case of sampling a continuous signal is just the periodic repetition of the original signal's spectrum which occurs at multiples of sampling frequency. When the sampling rate is below the Nyquist rate the spectra overlap causing aliasing whereas if the sampling rate is above the Nyquist rate the spectra do not overlap providing an accurate reconstruction of the signal.

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 removes high-frequency components, allowing only the frequencies within the desired range to pass through which ensures that the reconstructed signal matches with the original analog signal. Reducing the filter's bandwidth below the Nyquist rate attenuates some of the original signal's high frequency components which leads to loss of information. On the other hand increasing the filter's bandwidth above the Nyquist rate allows high-frequency noise and aliasing artifacts to pass through, causing distortion in the reconstructed signal.

4. 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. In the frequency spectrum it appears as mirrored frequencies that fold back into the spectrum. This is caused when higher frequency components fold back into the lower frequency range resulting in aliasing. To avoid aliasing one can use an anti- aliasing filter that removes frequencies higher than half the sampling rate which ensures only the frequencies that can be accurately sampled are present. Another way is by sampling at or above the Nyquist rate.

5. Effects of Undersampling: How does undersampling affect the reconstruction in the time and frequency domains?

Time Domain: If the sampling rate is below the Nyquist rate, the reconstructed signal will appear distorted. The signal may be reshaped , 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: Why might we choose a sampling rate higher than the minimum Nyquist rate?

In practical systems, a sampling rate higher than the minimum Nyquist rate is often chosen to address imperfections such as noise, non-ideal filter performance, and variations in signal characteristics. This higher sampling rate enhances signal reconstruction accuracy, provides a safety margin to prevent aliasing, and improves representation of signals, particularly those with components near the Nyquist frequency.

Additional Questions

1. Quantization Error

Quantization error is the difference between the actual analog signal value and the nearest quantized digital value. Increasing the number of quantization level decreases the quantization error since the quantization steps become finer. Here is a detailed relationship:

- Fewer quantization levels result in each level covering a broader range of analog values, which increases quantization error and decreases signal fidelity.
- Increasing the number of quantization levels reduces the error, as each step covers a narrower range, thereby enhancing the digital representation of the signal.

Therefore, increasing the quantization levels reduces quantization error, thereby improving signal quality, but it also requires more bits per sample.

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

The Signal-to-Noise Ratio (SNR) in terms 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:

$$\text{SNR (dB)} = 6.02n + 1.76$$

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

- Increasing the number of quantization bits raises the SNR, thus reducing the noise introduced by quantization and improving signal quality.
- Higher quantization levels allow for more precise signal representation.

3. Bitrate Calculation

For digital transmission of a signal the bitrate required is dependent on the sampling rate and the number of quantization levels. . For a signal with a sampling rate f_s and n bits per sample:

$$\text{Bitrate} = f_s \times n$$

Increasing the sampling rate or the number of quantization levels raises the bitrate, as more samples per second or more bits per sample are required, highlighting the trade-offs in digital communication where higher accuracy and fidelity demand 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:

- **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.
- **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.

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

The sampling rate, quantization levels, and signal quality should be balanced to meet specific application needs:

- 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.
- High-fidelity audio or video might demand high sampling rates and high quantization levels, increasing data requirements but enhancing quality.

7. Conclusion

The main goal of this experiment was to demonstrate the sampling, understand the reconstruction of a sampled signal and know about 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 concepts provides a key foundation in understanding the representation and recontruction of analog signal used in digital communication.

Summary of Results and Observations

The experiment demonstrated the crucial role of the Nyquist rate in sampling. Sampling at or above twice the highest signal frequency preserved the original data, enabling accurate reconstruction using sinc interpolation. When the sampling rate fell below the Nyquist rate, aliasing caused frequency distortion, preventing accurate signal reconstruction.

During quantization, we noted that fewer quantization levels increased quantization error as each sample was rounded to the nearest level. This error decreased with more quantization levels, consistent with theoretical expectations. The results also highlighted the trade-off between quantization accuracy and bit depth, emphasizing the need to balance data quality and resource usage.

Interpretation and Relation to Theory

This experiment confirmed the Sampling Theorem, showing that accurate signal reconstruction needs a sampling rate at least twice the maximum signal frequency. We observed aliasing in undersampled signals and precise reconstruction at or above the Nyquist rate, supporting the theorem. The quantization results also validated the theoretical model, indicating that higher quantization levels enhance digital representation fidelity but require more data.

Recommendations for Future Work

To expand on the current understanding of sampling and quantization, several new avenues for research and experimentation can be considered:

1. **Using Machine Learning for Optimal Sampling:** How to applying machine learning techniques to optimize sampling rates and quantization levels dynamically based on the characteristics of incoming signals to ensure accurate signal presentation and reconstruction.

2. **Comparing different Quantization Algorithm:** Conducting comparative studies of various quantization algorithms, such as uniform, non-uniform, and vector quantization, can provide insights into which methods best balance signal quality and computational efficiency for specific applications.

3. **Adaptive Quantization Techniques:** Investigating adaptive quantization methods that adjust the quantization step size based on the signal's dynamic range could improve signal fidelity and reduce quantization error, especially for signals with varying amplitude levels.

These future research directions and experiments would enhance the practical understanding and application of sampling and quantization, leading to better-designed systems for digital signal processing and communication technologies.

8. Acknowledgement

The MATLAB code used in this report was provided by Dr. Sudip Mandal an Assistant Professor at Jalpaiguri Government Engineering College, West Bengal, India, which was referenced and adapted for this experiment, as seen on his YouTube channel: Digital Communication Labs using MATLAB.

9. References

- **Digital Signal Processing with Matlab:**
  Hussain, Z. M., Sadik, A. Z., & O'Shea, P. (2011). *Digital Signal Processing: An Introduction with MATLAB and Applications*. Springer Science & Business Media.

- **MATLAB's built-in plotting functions.**
  Mustafy, T., & Rahman, M. T. U. (2024). MATLAB. In *Statistics and Data Analysis for Engineers and Scientists* (pp. 37-80). Singapore: Springer Nature Singapore.

- **Lecture Notes:**
  Wafula, Martin. Lecture notes. ECE 2414, Digital Communication, November 29, 2024. Personal communication. Multimedia University of Kenya.