Analysis of Sampling Theorem
Using Matlab

Ian Kamau

ENG-219-004/2021

Course Code and Name:
ECE 2414: Digital Communication 1

Date of Submission:
29th November 2024

Lecturer:
Martin Wafula

Contents

1. Abstract

This report examines the Sampling Theorem, which asserts that a continuous-time signal can be precisely reconstructed from its discrete samples if the sampling frequency is at least twice the signal's maximum frequency, referred to as the Nyquist rate. Through practical analysis, the study explores the theorem by conducting signal sampling and reconstruction in both time and frequency domains. Using MATLAB, experiments were carried out to sample signals, study their frequency spectra, and reconstruct the original signals. The investigation highlights the effects of different sampling rates on the accuracy and quality of signal reconstruction.

2. Introduction

For broader understanding of Sampling and Quantization as used in Digital communication, I will explore the mathematical foundations of these concepts, focusing on Fourier analysis as it applies to the Sampling Theorem and the study of quantization errors. By examining these principles in detail, the explanation will demonstrate how they underpin the core mechanisms of digital communication systems.

3. Objective

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

4. Theoretical Background

Sampling Theorem

The Sampling Theorem establishes that a continuous-time signal can be accurately represented in a discrete form if it is sampled at a rate equal to or greater than twice its highest frequency component, commonly referred to as the **Nyquist rate**. This principle forms the foundation of digital signal processing, enabling the conversion of analog signals into digital form without losing critical information.

Sampling involves capturing a continuous-time signal $x(t)$ at discrete intervals using an impulse train $\delta(t - nT)$, where each impulse corresponds to a sampling point. The sampled signal $x_s(t)$ can be expressed as:

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

Here, $T = \frac{1}{2W}$ is the sampling period, ensuring a high enough sampling rate to preserve the signal's information.

In the frequency domain, sampling creates periodic replicas of the signal's spectrum. If $X(f)$ is the Fourier transform of $x(t)$, the Fourier transform of the sampled signal $x_s(t)$, denoted $X_s(f)$, is given by:

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

This periodic repetition occurs at intervals of $\frac{1}{T}$, the sampling frequency.

Aliasing occurs when the sampling frequency f_s is less than twice the maximum frequency W, leading to overlapping spectral replicas. To avoid this distortion, the sampling frequency must satisfy:

$$f_s \geq 2W$$

This ensures non-overlapping spectra and retains the signal's integrity.

Reconstruction of Sampled Signals

To reconstruct a continuous-time signal from its samples, the sampling rate must meet or exceed the Nyquist rate. The reconstruction formula using ideal sinc interpolation is:

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

Where:

- $x(t)$: Reconstructed signal.
- $x[n]$: Sampled values.
- T: Sampling period ($T = \frac{1}{f_s}$).
- $\text{sinc}(t) = \frac{\sin(\pi t)}{\pi t}$: Sinc function, acting as an ideal low-pass filter.

In practice, sinc interpolation is often replaced with simpler methods like linear or spline interpolation, though the sinc function provides theoretically perfect reconstruction when the sampling criterion is met.

Quantization for Analog Sequence Sources

Quantization involves mapping an analog sequence of real numbers $s_1, s_2, \ldots$ to a finite set of discrete representation points $\{q_1, q_2, \ldots, q_M\}$, each associated with a quantization region R_i. The process introduces a measure of accuracy based on the squared error, and optimizing the quantization requires careful selection of both quantization regions and representation points.

1. Quantization Regions

Given fixed representation points $\{q_1, q_2, \ldots, q_M\}$, quantization regions R_i are chosen to minimize the squared error between a source value s and the closest representation point q_i. The optimal regions are defined as intervals determined by midpoints between consecutive points:

$$R_i = \begin{cases} (-\infty, \frac{q_1+q_2}{2}] & \text{if } i = 1, \\ (\frac{q_{i-1}+q_i}{2}, \frac{q_i+q_{i+1}}{2}] & \text{if } 1 < i < M, \\ (\frac{q_{M-1}+q_M}{2}, \infty) & \text{if } i = M. \end{cases}$$

2. Representation Points

For fixed quantization regions R_i, the optimal representation point q_i minimizes the squared error within R_i. The representation point is chosen as the conditional mean of the source values within the region:

$$q_i = E[S \mid S \in R_i] = \int_{R_i} s\, f_i(s)\, ds$$

where $f_i(s)$ is the conditional probability density function of S given $S \in R_i$.

3. Mean Squared Error (MSE)

The overall distortion introduced by quantization is measured by the Mean Squared Error (MSE), which is calculated as:

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

Here:

- $f_S(s)$: Probability density function of the source S.
- $Q(S)$: Quantization function mapping S to the nearest q_i.
- R_i: Quantization region corresponding to q_i.

The MSE is partitioned into contributions from each quantization region, allowing separate optimization for each.

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 the MATLAB code used for verifying the Sampling Theorem:

```
%% Copyright @ Dr Sudip Mandal
% Digital Communication Lab

%% ANALYSIS OF SAMPLING THEOREM
clear all; close all; clc;

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

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

% FFT of the message signal
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');
title('Spectrum of Message Signal'); grid;

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

figure(3);
stem(tsm, xsm, 'linewidth', 2);
xlabel('time'); ylabel('amplitude');
title('Sampled Signal'); grid;

% FFT of the sampled signal
xsmu = upsample(xsm, Nfactor);
xfftu = fftshift(fft(xsmu, Lfft));

figure(4);
plot(Faxis, 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.
4. Design a low-pass filter to retain frequencies between $-10\,\text{Hz}$ and $10\,\text{Hz}$.
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:

```
%% Copyright @ Dr Sudip Mandal
% Digital Communication Lab

%% 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 Siganl
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('Amlpitude');
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;
```

5.3 Quantization

1. Define quantization levels (e.g., 8, 16, 32).
2. Quantize the signal by mapping samples to the nearest levels.

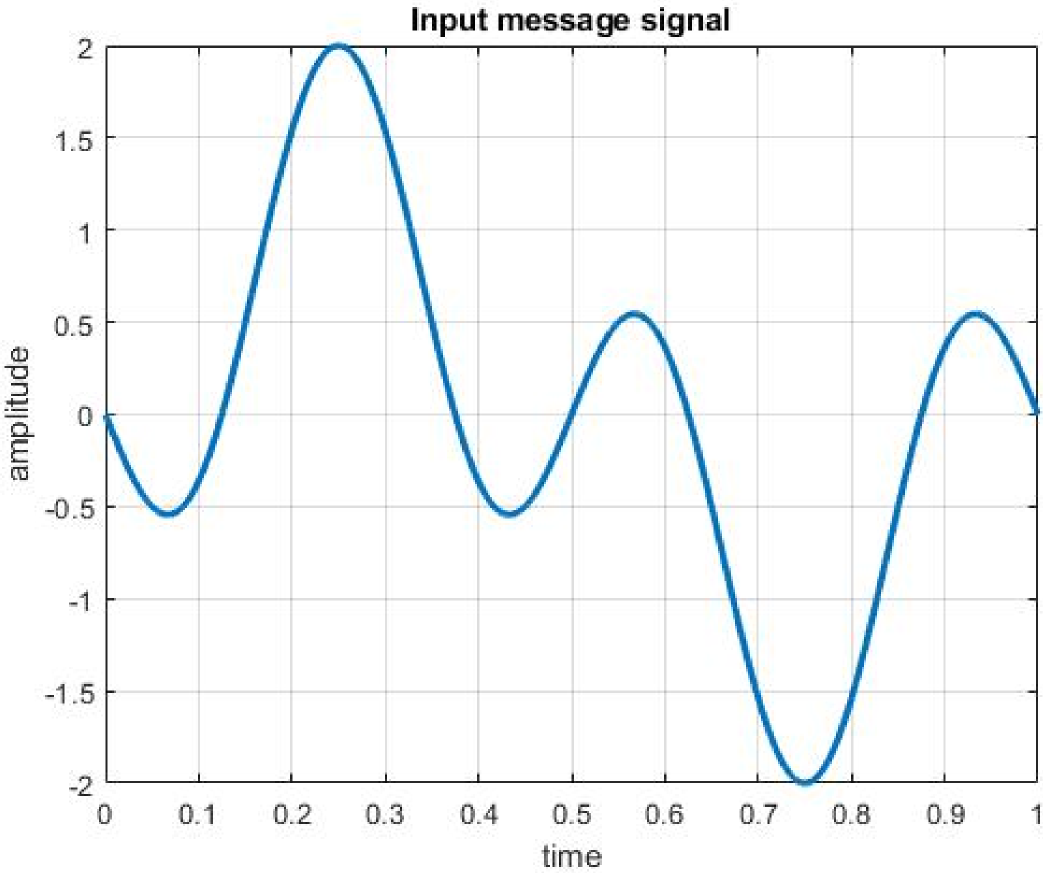
3. Plot the quantized signal and calculate quantization error.

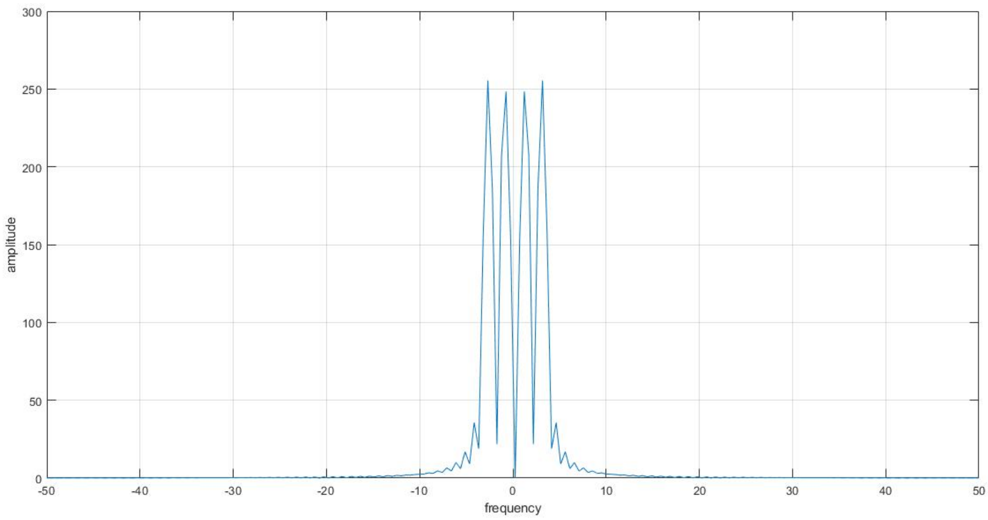
MATLAB Code Outline

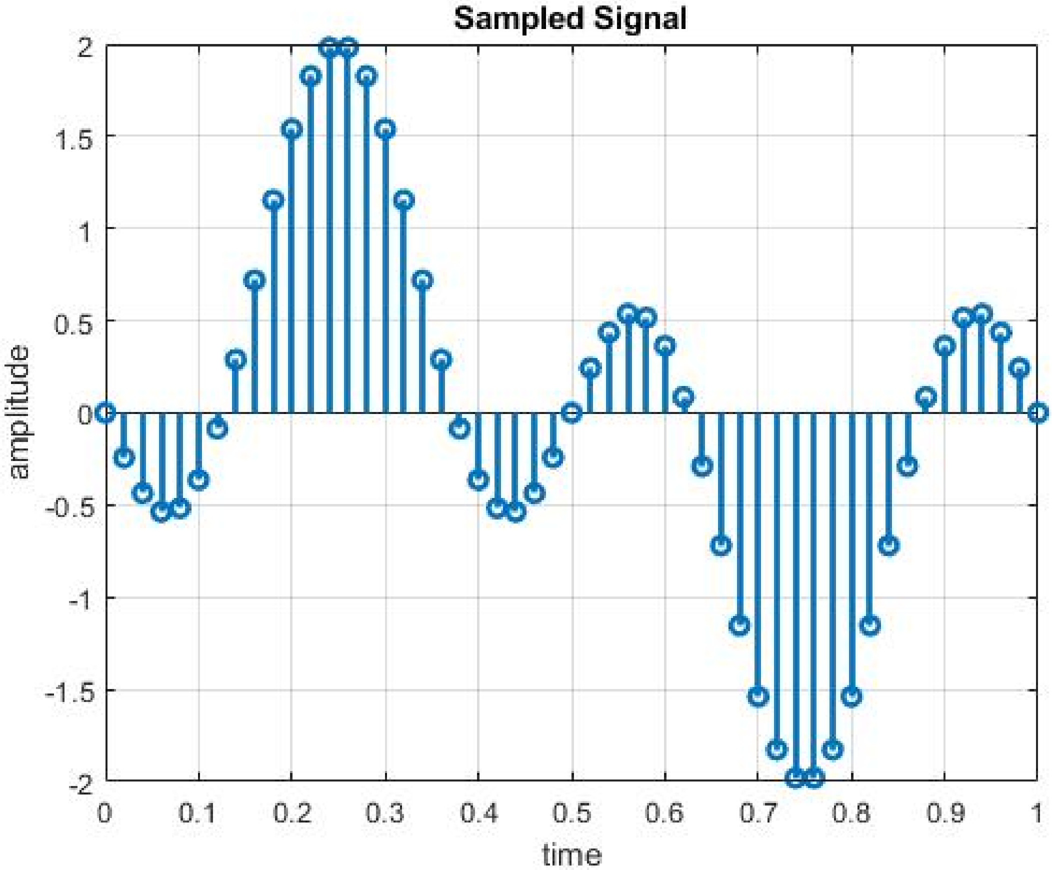
```
% Quantization Process in MATLAB

clear all; close all; clc;

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

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

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

% Quantize the sampled signal
levels = 16;
x_min = min(xsm); x_max = max(xsm);
step = (x_max - x_min) / levels;
x_quantized = step * round((xsm - x_min) / step) + x_min;

% Plot sampled vs. quantized signals
figure(2);
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;

% Quantization Error
quantization_error = xsm - x_quantized;
figure(3);
stem(tsm, quantization_error, 'LineWidth', 1.5);
xlabel('Time (s)'); ylabel('Error');
title('Quantization Error'); grid;
```

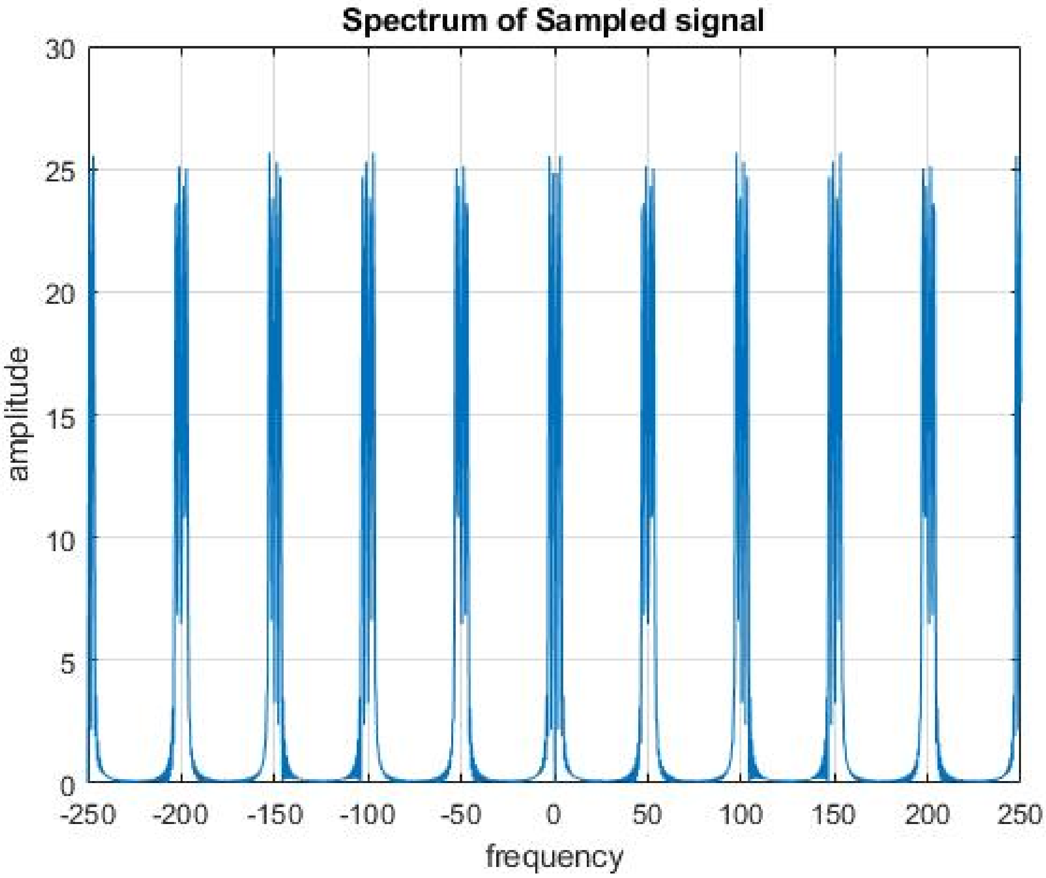
6. Discussion

1. Sampling Results and Analysis





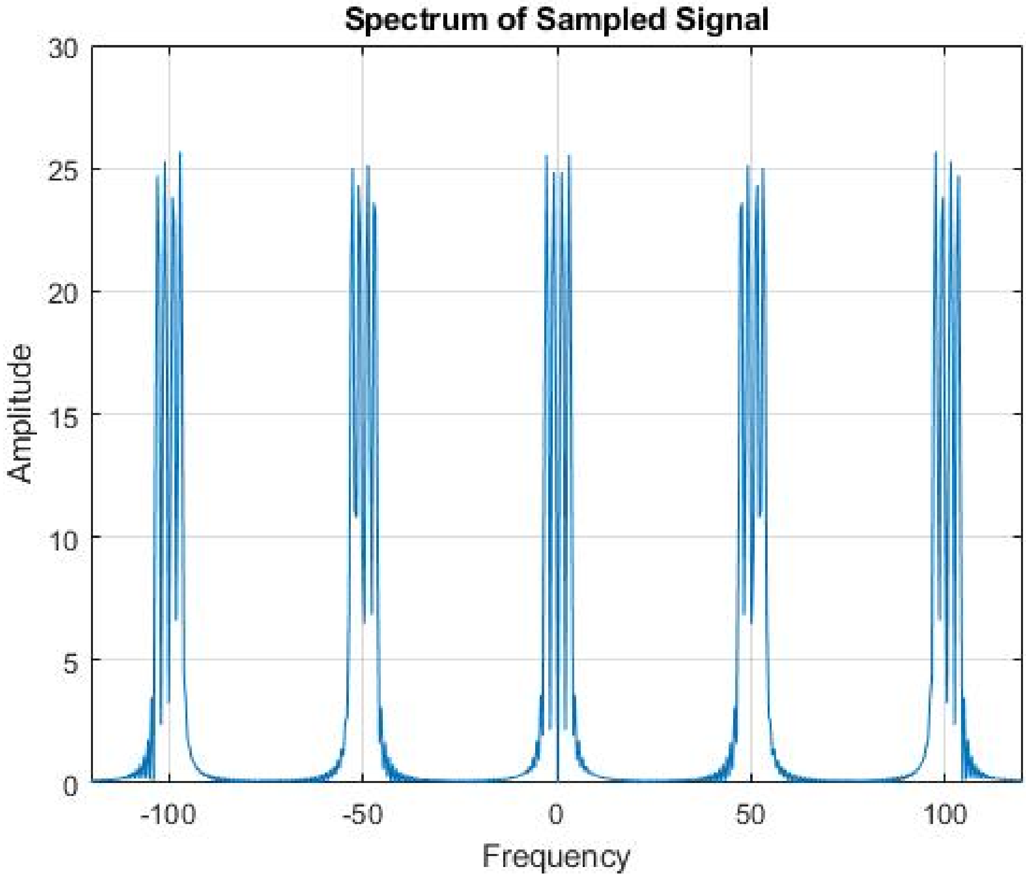

Sampled Signal
amplitude
time
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

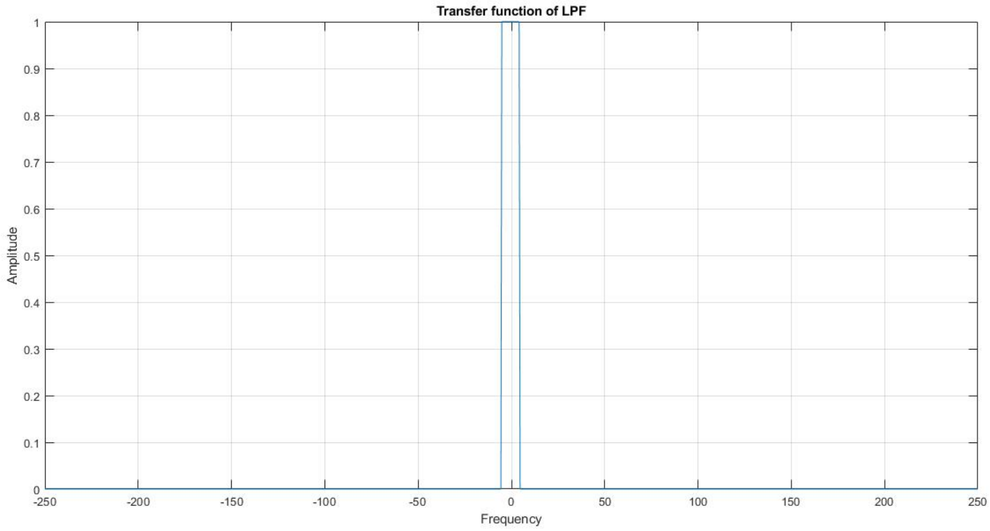


Observations from Graphs

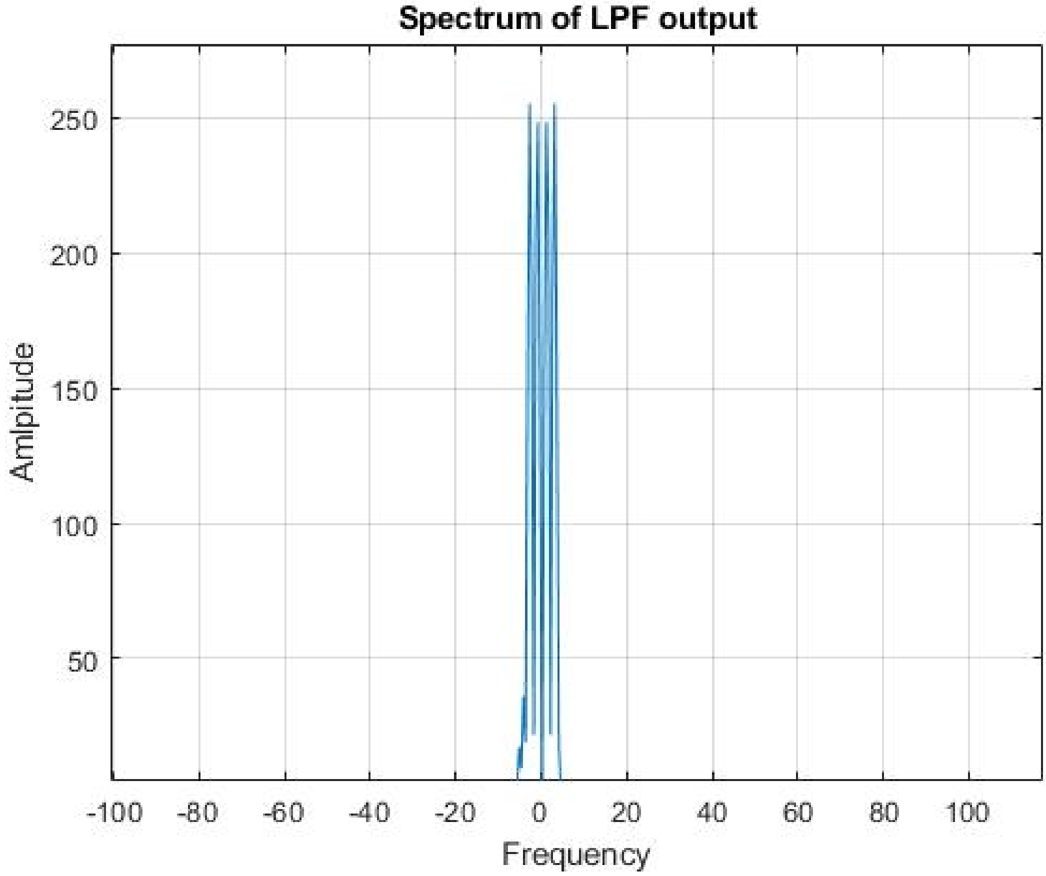
When the sampling rate exceeded the Nyquist rate, the reconstructed signal closely resembled the original. For example, sampling a sinusoidal signal with frequencies of 1 Hz and 3 Hz at 50 Hz (well above the Nyquist rate of 6 Hz) produced a discrete-time signal that retained the features of the original continuous signal. However, reducing the sampling rate below the Nyquist rate introduced aliasing effects, where components like the 3 Hz frequency were misrepresented as lower frequencies, resulting in distortion in the frequency domain.

2. Reconstruction and Sinc Interpolation

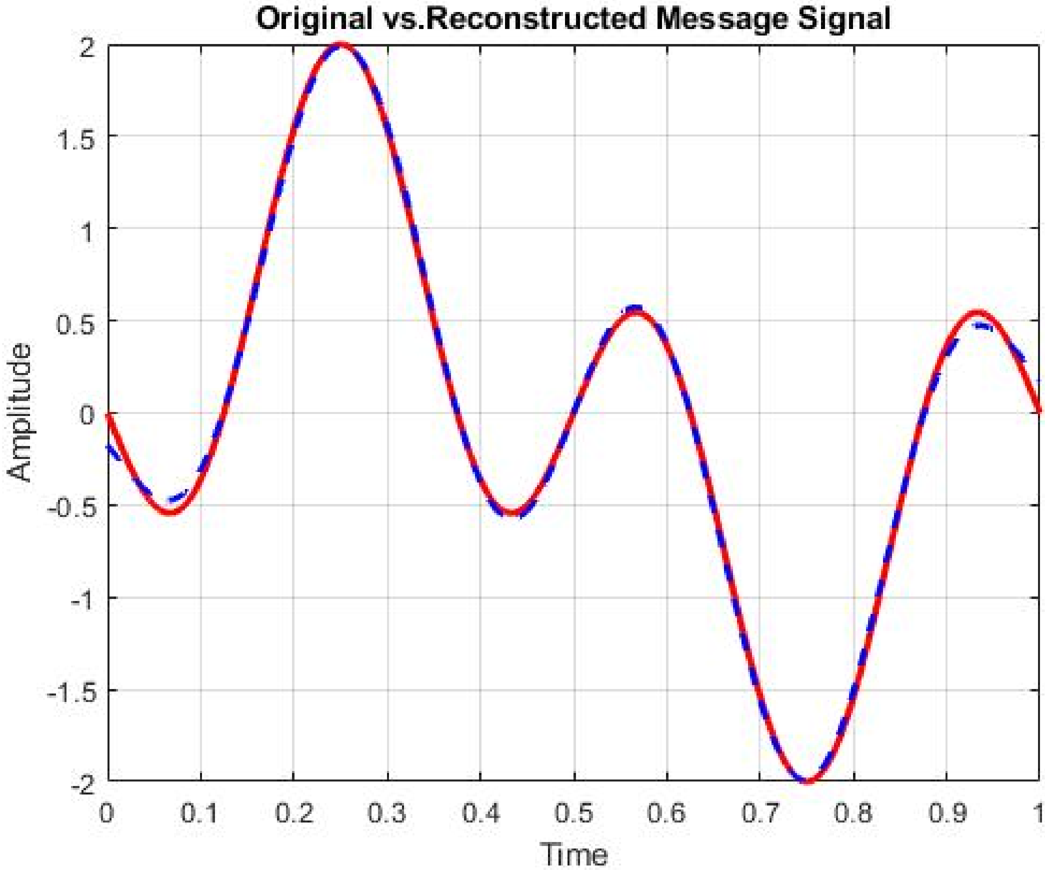




- We reconstructed the continuous-time signal from discrete samples taken above the Nyquist rate using sinc interpolation. Here our sinc function serves as an ideal–low pass filter.

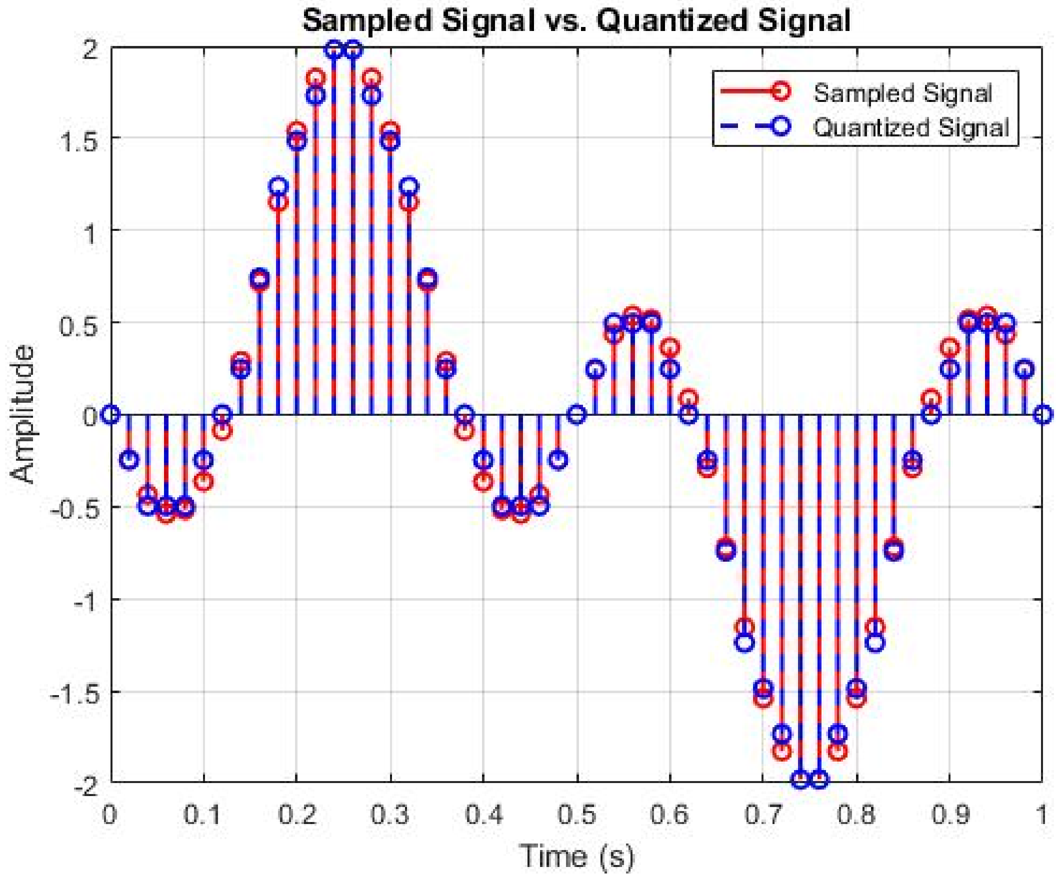


Spectrum of LPF output
250
200
150
100
50
Amlpitude
-100
-80
-60
-40
-20
0
20
40
60
80
100
Frequency

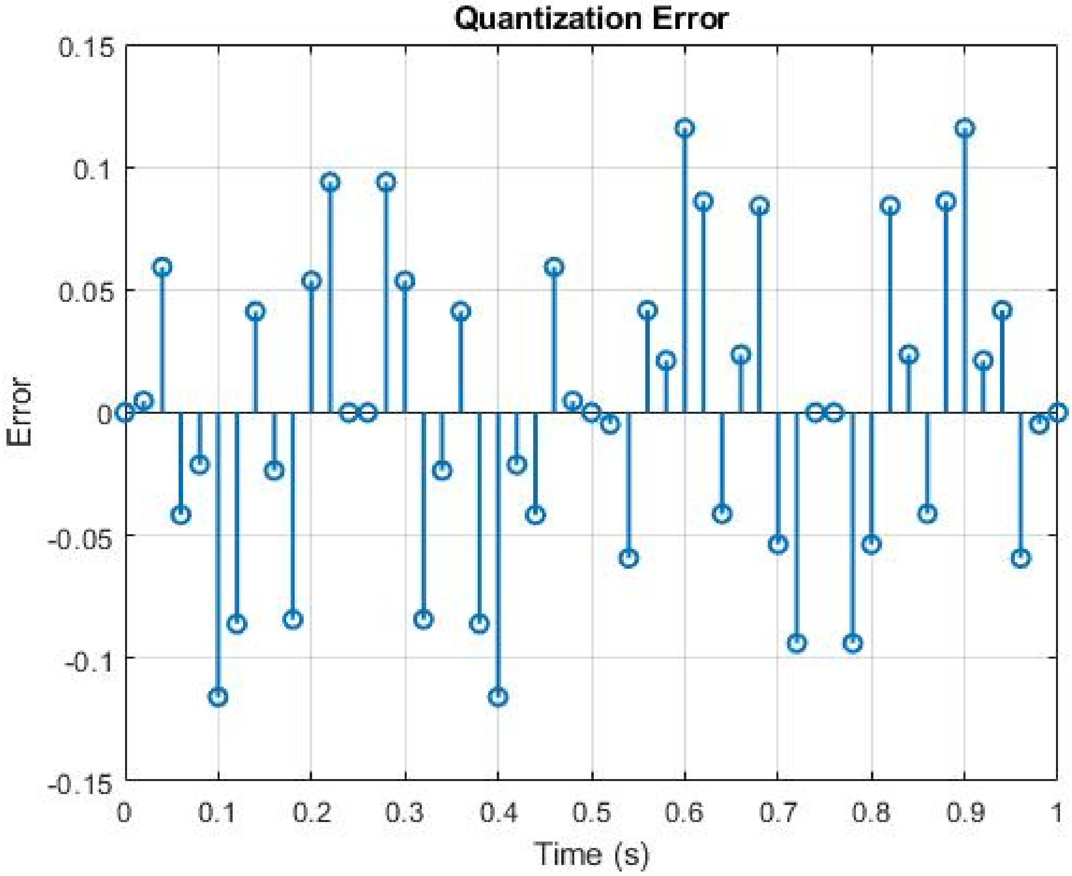


3.Quantization Results and Analysis

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



Sampled Signal vs. Quantized Signal
Sampled Signal
Quantized Signal
Amplitude
Time (s)
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1



Observations from Graphs

With a limited number of quantization levels, the quantized signal exhibited reduced resolution, particularly in areas with fast amplitude changes. Increasing the number of quantization levels allowed the quantized signal to better approximate the original sampled signal. The quantization error was more pronounced with fewer levels but decreased as the number of levels increased.

Discussion Questions

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

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) . 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. Sampling at or above the Nyquist rate ensures that the original message is reconstructed whereas 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 frequency spectrum is just the periodic repetition of the original signal's spectrum which occurs at multiples of sampling frequency. When the sampling rate is above the Nyquist rate the spectra do not overlap providing an accurate reconstruction of the signal whereas if the sampling rate is below the Nyquist rate the spectra overlap causing aliasing.

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?

The 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. However, increasing the filter's bandwidth above the Nyquist rate allows high-frequency noise to pass through, causing changes 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 the sampling rate is lower than twice the highest frequency component of the signal. In the frequency spectrum it appears as reflected 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?

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.

In 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.

6. Practical Sampling Rates: Why might we choose a sampling rate higher than the minimum Nyquist rate?

Sampling rates exceeding the minimum Nyquist rate are commonly used to account for factors like noise and non-ideal filter behavior. A higher sampling rate improves reconstruction accuracy, offers a buffer against aliasing, and ensures better representation of signals, especially those with components close to 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 levels decreases the quantization error because the quantization steps become finer. With fewer levels, each quantization step represents a broader range of analog values, leading to greater quantization error and reduced signal fidelity. Conversely, increasing the number of levels narrows the range covered by each step, reducing the error and improving the accuracy of the digital signal representation. However, this improvement in signal quality comes at the cost of requiring 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:

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

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:

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

Raising the sampling rate or increasing the number of quantization levels results in a higher bitrate, as it requires more samples per second or additional bits per sample.

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.
- **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 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.

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

Balancing sampling rate, quantization levels, and signal quality is crucial to meet the requirements of specific applications. For low-power or low-bandwidth systems, lower sampling rates and quantization levels may be used to conserve power and bandwidth, though this comes at the cost of reduced signal fidelity. Conversely, high-fidelity applications like audio and video processing often require higher sampling rates and quantization levels, which increase data demands but deliver superior quality.

7. Conclusion

The primary goal of this experiment was to demonstrate the sampling, understand the reconstruction of a sampled signal and execute quantization in digital communication. Sampling from the report is the conversion of a continuous analog signal into a discrete sequence, while quantization involves mapping each discrete sample to a finite set of values for digital processing. As an Electrical Engineering understanding these concepts provides a basis of creating machinery for improving this field of digital communication.

Summary of Results and Observations

The experiment highlighted the importance of the Nyquist rate in the sampling process. When the sampling rate was at or above twice the highest frequency in the signal, the original data was preserved, allowing for precise reconstruction using sinc interpolation. However, sampling at a rate below the Nyquist rate resulted in aliasing, which caused frequency distortion and made accurate reconstruction impossible.

In the quantization phase, it was observed that using fewer quantization levels led to higher quantization errors, as each sample was mapped to the nearest level. This error reduced as the number of quantization levels increased, aligning with theoretical predictions. The findings also underscored the trade-off between quantization accuracy and bit depth, illustrating the importance of balancing data quality with resource constraints.

Interpretation and Relation to Theory

The experiment validated the Sampling Theorem, demonstrating that accurate signal reconstruction requires a sampling rate that is at least twice the highest frequency of the signal. We observed aliasing in signals sampled below the Nyquist rate, while signals sampled at or above the Nyquist rate were reconstructed accurately, confirming the theorem. The quantization findings further supported the theoretical model, showing that increasing the number of quantization levels improves the fidelity of the digital representation, albeit at the cost of increased data requirements.

Recommendations for Future Work

To further our current understanding of sampling and quantization, several new avenues for research and experimentation can be considered:

1. **Impact of Noise on Quantization:** Studying the effects of different noise types (e.g. white noise) on quantization error and exploring methods to minimize the impact of noise during the sampling and quantization process.
2. **Real-Time Sampling Systems:** Creating and testing real-time sampling systems that implement advanced sampling techniques, such as compressed sensing, to determine their effectiveness in reducing data size while maintaining signal quality in real-world scenarios.
3. **Hardware Implementation and Performance Testing:** Moving from theoretical studies to practical hardware implementations of sampling and quantization circuits to assess their performance, power consumption, and processing capabilities in real-world digital communication systems.

These 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

Special thanks to Dr. Sudip Mandal an Assistant Professor at Jalpaiguri Government Engineering College, West Bengal, India, for providing the Matlab code which was referenced and adapted for this experiment, as seen on his YouTube channel: Digital Communication Labs using MATLAB.

9. References

- **Principle of Digital Communication:**
  Gallager, R. G. (2008). *Principles of Digital Communication* (Vol. 1). Cambridge, UK: Cambridge University Press.

- **Digital Signal Processing with Matlab:**
  Leis, J. W. (2011). *Digital Signal Processing Using MATLAB for Students and Researchers*. John Wiley & Sons.

- **MATLAB's built-in plotting functions.**
  Gagniuc, P. A. (2024). MATLAB Specific. In *Coding Examples from Simple to Complex: Applications in MATLAB®* (pp. 251–270). Cham: Springer Nature Switzerland.