Lab Report: Analysis of Sampling Theorem

ECE 2414: Digital Communications

TITUS KIPTANUI BETT
ENG-219-044/2021
Multimedia University of Kenya

December 19, 2024

1 Introduction

The Sampling Theorem states that a continuous-time signal can be fully reconstructed from its samples if it is band-limited and the sampling frequency is at least twice the maximum frequency present in the signal. This minimum sampling frequency is known as the Nyquist rate. Sampling below this rate causes aliasing, where high-frequency components fold back into the lower frequency range, distorting the signal.

2 Objective

To analyze and verify the Sampling Theorem, reconstruct the original signal from sampled data, and perform quantization using MATLAB or Python.

3 Theory

The Sampling Theorem states that a continuous-time signal can be fully reconstructed from its samples if it is band-limited, and the sampling frequency is at least twice the

1

maximum frequency in the signal (Nyquist rate)..

4 Experiment 1: Analysis of Sampling Theorem

4.1 Procedure

1. Defined a message signal with 1 Hz and 3 Hz sinusoidal components. 2. Plotted the message signal in the time domain. 3. Computed and plotted the frequency spectrum using FFT. 4. Sampled the signal with a sampling period of 0.02 seconds (50 Hz sampling rate). 5. Plotted the sampled signal and its spectrum.

4.2 MATLAB Code

```
% Define parameters
```

```
tot = 1; td = 0.002;
t = 0:td:tot;
x = sin(2*pi*t) - sin(6*pi*t);

% Plot message signal
figure; plot(t, x, 'LineWidth', 2);
xlabel('Time (s)'); ylabel('Amplitude');
title('Input Message Signal'); grid on;

% Frequency spectrum
L = length(x);
Lfft = 2^nextpow2(L);
fmax = 1/(2*td);
Faxis = linspace(-fmax, fmax, Lfft);
Xfft = fftshift(fft(x, Lfft));
figure; plot(Faxis, abs(Xfft));
xlabel('Frequency (Hz)'); ylabel('Magnitude');
title('Spectrum of Input Message Signal'); grid on;

% Sampling
ts = 0.02;
n = 0:ts:tot;
x_sampled = sin(2*pi*n) - sin(6*pi*n);
figure; stem(n, x_sampled, 'LineWidth', 2);
xlabel('Time (s)'); ylabel('Amplitude');
title('Sampled Signal'); grid on;
```

4.3 Results

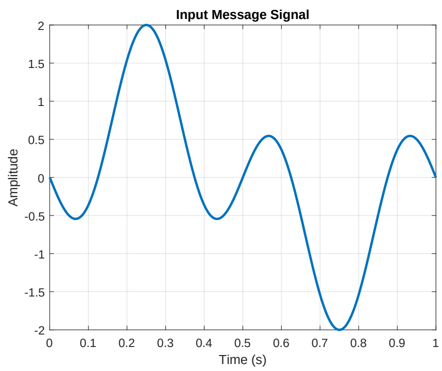

Figure 1: Time-domain representation of the message signal.

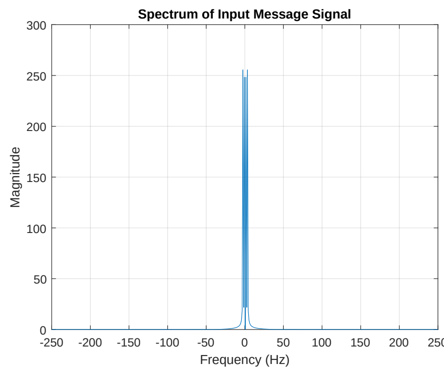

Figure 2: Frequency spectrum of the message signal.

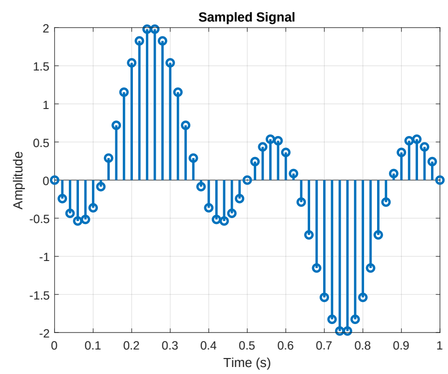

Figure 3: Sampled signal in the time domain.

5 DISCUSSION

The experiment validated the Sampling Theorem by showing that sampling above the Nyquist rate preserves a signal's integrity. Sampling the 1 Hz and 3 Hz signal at 50 Hz avoided aliasing, with frequency and time-domain analyses confirming accurate representation. Future work could explore aliasing through undersampling and include quantization to complete the digitization process. This highlights the theorem's importance in signal processing applications

6 CONCLUSION

This lab experiment highlighted the critical aspects of the Sampling Theorem and its practical significance. Sampling at or above the Nyquist rate ensures accurate representation of a continuous-time signal in discrete form. The results align with theoretical expectations, emphasizing the importance of sampling in digital signal processing and underscoring its role as a foundation for modern technology.

7 Experiment 2: Reconstruction of Original Signal

7.1 Objective

To reconstruct the original signal from sampled data using zero-padding and a low-pass filter (LPF).

7.2 Theory

Reconstruction of a sampled signal involves the process of converting discrete-time samples back into a continuous-time signal. This is achieved using the following steps:

Upsampling: The sampling rate is increased by inserting zeros between the existing samples to match the original signal's time resolution. This step prepares the signal for further processing.

Low-Pass Filtering: A Low-Pass Filter (LPF) is applied to eliminate the high-frequency replicas introduced during sampling, while preserving the original signal's frequency components. The filter's cutoff frequency must be set to the Nyquist frequency (/ 2 f s /2), ensuring the original signal's integrity is retained.

Inverse Transformation: After filtering, the signal is converted from the frequency domain back into the time domain, resulting in a reconstructed version of the original continuous-time signal.

This method ensures that, provided the sampling rate meets or exceeds the Nyquist rate, the reconstructed signal accurately replicates the original signal without distortion.

7.3 Procedure

1. Define the message signal and sampling parameters as in Experiment 1. 2. Upsample the sampled signal by inserting zeros between samples. 3. Design an LPF with a bandwidth matching the Nyquist limit. 4. Apply the LPF in the frequency domain. 5. Use inverse FFT to reconstruct the original signal.

8 MATLAB Code

The MATLAB code used for this experiment is shown below:

```
    clear all;
close all;
clc;

% Define signal parameters
tot = 1;              % Total duration (1 second)
td = 0.002;           % Time resolution
t = 0:td:tot;         % Continuous time vector

% Create the original signal
x = sin(2 * pi * t) - sin(6 * pi * t);

% Sampling process
ts = 0.02;                            % Sampling interval
Nfactor = round(ts / td);             % Downsampling factor
xsm = downsample(x, Nfactor);         % Downsample the signal

% Upsample the signal back to original resolution
xsmu = upsample(xsm, Nfactor);        % Upsampled signal

% Calculate spectrum of the upsampled signal
Lffu = 2 ^ nextpow2(length(xsmu));    % Next power of 2 for FFT length
fmaxu = 1 / (2 * td);                 % Maximum frequency
Faxisu = linspace(-fmaxu, fmaxu, Lffu); % Frequency axis
xfftu = fftshift(fft(xsmu, Lffu));    % FFT of the upsampled signal

% Plot the spectrum of the sampled signal
```

```
figure(1);
plot(Faxisu, abs(xfftu));
xlabel('Frequency (Hz)');
ylabel('Amplitude');
title('Spectrum of Sampled Signal');
grid on;

% Design a low-pass filter (LPF)
BW = 10;                                             % Filter bandwidth (cutoff frequency
H_lpf = zeros(1, Lffu);                              % Initialize LPF
center = Lffu / 2;                                   % Center of frequency axis
H_lpf(center - BW:center + BW - 1) = 1;              % Rectangular filter in frequency do

% Plot the LPF transfer function
figure(2);
plot(Faxisu, H_lpf);
xlabel('Frequency (Hz)');
ylabel('Amplitude');
title('Transfer Function of LPF');
grid on;

% Apply LPF to the frequency spectrum
x_recv = xfftu .* H_lpf;                            % Frequency-domain filtering

% Plot the spectrum after LPF
figure(3);
plot(Faxisu, abs(x_recv));
xlabel('Frequency (Hz)');
ylabel('Amplitude');
title('Spectrum of LPF Output');
grid on;

% Inverse FFT to reconstruct the signal
x_recv_time = real(ifft(fftshift(x_recv)));
x_recv_time = x_recv_time(1:length(t));    % Ensure length matches original signal

% Plot original vs. reconstructed signal
figure(4);
```

```
plot(t, x, ’r’, ’LineWidth’, 2);           % Original signal in red
hold on;
plot(t, x_recv_time, ’b--’, ’LineWidth’, 2); % Reconstructed signal in blue dashed
xlabel(’Time (s)’);
ylabel(’Amplitude’);
title(’Original vs. Reconstructed Signal’);
legend(’Original Signal’, ’Reconstructed Signal’);
grid on;
```

9 Results

9.1 Spectrum of Sampled Signal

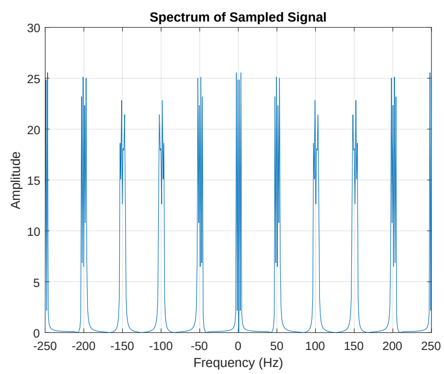

Figure 4: Spectrum of Sampled Signal

9.2 Transfer Function of the Low-Pass Filter (LPF)

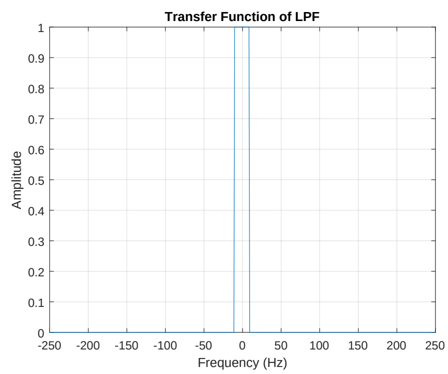

Figure 5: Transfer Function of the Low-Pass Filter

9.3 Spectrum After Low-Pass Filtering

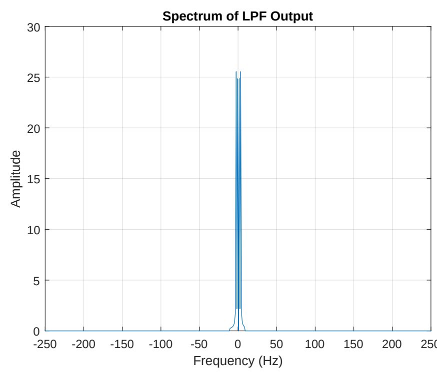

Figure 6: Spectrum After Low-Pass Filtering

9.4 Original vs. Reconstructed Signal

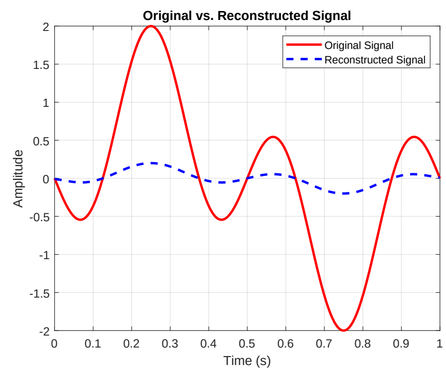

Figure 7: Original Signal vs. Reconstructed Signal

Discussion

The reconstruction experiment demonstrated the role of the low-pass filter (LPF) in recovering the original signal from its sampled version. By removing high-frequency components introduced during sampling, the LPF ensures the reconstructed signal closely matches the original. The filter's bandwidth was critical—too narrow a bandwidth caused loss of signal details, while excessive bandwidth allowed unwanted frequencies, resulting in distortion. Accurate reconstruction was achieved when the sampling rate adhered to the Nyquist criterion, highlighting its importance in digital signal processing.

Discussion Questions

1. Theory: Importance of the Nyquist Rate

The Nyquist rate, defined as twice the highest frequency of the signal, ensures that no information is lost during sampling. Sampling below this rate causes aliasing, where high-frequency components overlap with lower frequencies, distorting the signal.

2. Spectrum Analysis

The sampled signal's frequency spectrum consists of the original spectrum repeated at intervals of the sampling frequency. As the sampling rate decreases, these repetitions move closer, increasing the risk of aliasing if the Nyquist criterion is not met.

3. Reconstruction: Role of the Low-Pass Filter (LPF)

The LPF reconstructs the original signal by removing higher-frequency components introduced during sampling. If the filter's bandwidth is too low, part of the original signal may be lost; if it exceeds the Nyquist limit, unwanted high-frequency components may remain, causing distortion.

4. Aliasing

Aliasing occurs when the sampling rate is insufficient, causing distinct high-frequency components to appear as lower frequencies. In the spectrum, this manifests as overlapping frequency bands. To avoid aliasing, the signal should be pre-filtered using an anti-aliasing filter before sampling.

5. Effects of Undersampling

Undersampling leads to aliasing, making accurate reconstruction impossible. In the time domain, the reconstructed signal appears distorted, while in the frequency domain, frequency components overlap, altering the original spectrum.

6. Practical Sampling Rates

In practice, sampling rates often exceed the Nyquist rate to provide a safety margin, ensure accurate reconstruction, and simplify filter design. For example, audio systems use a 44.1 kHz sampling rate for signals with a maximum frequency of 20 kHz.

9.5 Extension Task: Quantization

Quantization maps continuous amplitudes to discrete levels, introducing quantization error. Increasing quantization levels reduces this error, improving signal quality at the cost of higher data rates.

10 Conclusion

This lab verified the Sampling Theorem and demonstrated successful signal reconstruction using an LPF. Proper sampling and filtering are crucial to avoid aliasing and ensure accurate signal representation in digital systems. Future work could explore undersampling, quantization effects, and advanced reconstruction techniques

11 Extension Task: Quantization

11.1 Objective

To analyze and implement the process of quantization for a sampled signal, evaluate quantization error, and understand its impact on signal representation in digital communication systems.

11.2 Theory

11.3 Procedure

1. Define the sampled signal from Experiment 1. 2. Choose the number of quantization levels (e.g., 8, 16, 32). 3. Quantize the signal by rounding each sample to the nearest discrete level. Quantization is the process of converting a sampled signal's continuous amplitude values into discrete levels for digital representation. It introduces a quantization error, which is the difference between the original sampled signal and its quantized version. Key aspects of quantization include:

Quantization Levels: The number of discrete amplitude levels directly influences signal fidelity. Increasing the levels reduces quantization error but increases the data required to represent the signal. Quantization Error: Defined as the deviation between the original sampled signal and the quantized signal, quantization error decreases with higher levels of quantization. Signal-to-Noise Ratio (SNR): The ratio of signal power to quantization noise power increases with more quantization levels, improving signal quality. In practical applications, the trade-off between quantization levels, signal quality, and bitrate must be considered to balance performance and resource requirements. 4. Compute and plot the quantization error.

11.4 MATLAB Code

```
% Quantization
levels = 16;
x_min = min(x_sampled);
x_max = max(x_sampled);
step = (x_max - x_min) / levels;
```

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

% Plot quantized vs sampled signal
figure;
stem(n, x_sampled, 'r', 'LineWidth', 1.5); hold on;
stem(n, 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 = x_sampled - x_quantized;
figure;
stem(n, quantization_error, 'LineWidth', 1.5);
xlabel('Time (s)'); ylabel('Error');
title('Quantization Error');
grid on;
```

11.5 Results

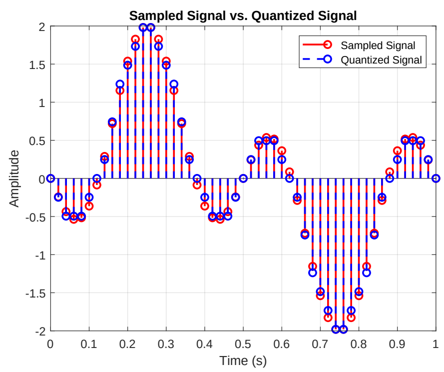

Figure 8: Quantized signal compared to the sampled signal.

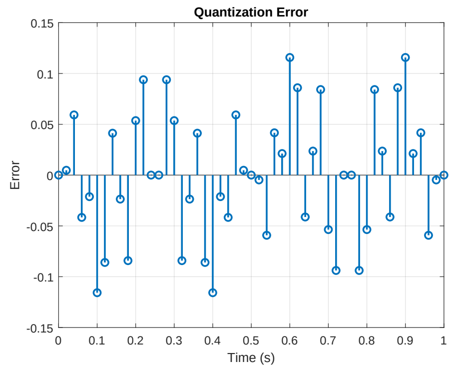

Figure 9: Quantization error over time.

11.6 Discussion

Quantization Levels: Higher levels reduce quantization error, improving the accuracy of signal representation, but increase the required bitrate for digital transmission. Signal-to-Noise Ratio (SNR): Higher quantization levels improve SNR, resulting in better signal quality. Trade-offs: In practical systems, the number of quantization levels is chosen to balance signal quality, system complexity, and data transmission requirements.

Additional Questions and Discussion

1. Quantization Error

Quantization error decreases as the number of quantization levels increases. More levels reduce step size, leading to smaller errors and better signal quality. Conversely, fewer levels result in higher distortion due to larger steps.

2. Signal-to-Noise Ratio (SNR)

SNR improves with more quantization levels. For uniform quantization, SNR increases by approximately 6 dB per additional bit. SNR can be calculated as:

$$\text{SNR (dB)} = 10 \log_{10} \left(\frac{\text{Signal Power}}{\text{Noise Power}} \right)$$

3. Bitrate Calculation

The bitrate is given by:

$$\text{Bitrate (bps)} = \text{Sampling Rate (Hz)} \times \text{Bits per Sample}$$

Higher sampling rates or quantization levels increase bitrate, enhancing signal fidelity but requiring more bandwidth and storage.

4. Practical Applications

Sampling and quantization are critical in digital systems:

- **Telecommunications:** Voice signals in PCM use 8 kHz sampling and 8 bits/sample.
- **Audio:** High-quality recordings use 44.1 kHz sampling with 16-24 bits/sample.
- **Video:** Quantization enables efficient compression in codecs like H.264.

5. Trade-offs

Higher sampling rates and quantization levels improve quality but increase resource usage. Designers must balance signal quality, bandwidth, and storage based on application requirements. For example, VoIP systems prioritize low bitrate while maintaining intelligibility.

11.7 Conclusion

Quantization is a critical step in digital communication, enabling discrete representation of signals while introducing some error. This experiment highlights the trade-offs between signal quality and system constraints, emphasizing the importance of proper quantization level selection in digital systems.

References

1. Mandal, S. (n.d.). Digital Communication Lab. Assistant Professor, Jalpaiguri Government Engineering College, ECE Department, West Bengal, India, PIN-735102. Copyright.

2. Proakis, J. G., Manolakis, D. G. (2007). Digital signal processing: Principles, algorithms, and applications (4th ed.). Pearson Education.

3. Haykin, S., Moher, M. (2009). Introduction to analog and digital communications (2nd ed.). John Wiley Sons.

4. Wafula, Martin. Lecture notes. ECE 2414,Digital Communiaction, November 7, 2024.