Lab Report: Analysis of Sampling Theorem using MATLAB

ECE 2414: Digital Communications

DAVID MAGINA OCHIENG
Multimedia University of Kenya

November 28, 2024

1 Introduction

The objective of this lab is to analyze and verify the Sampling Theorem, reconstruct the original signal from sampled data, and perform quantization using MATLAB or Python. The Sampling Theorem states that a continuous-time signal can be fully reconstructed from its samples if the sampling frequency is at least twice the maximum frequency in the signal (the Nyquist rate).

2 Objective

- To analyze the Sampling Theorem. - To reconstruct the original signal using low-pass filtering (LPF). - To extend the experiment with quantization.

3 Theory

The Sampling Theorem forms the basis of digital communication systems. For a signal band-limited to $f_{\max}$, the sampling frequency f_s must satisfy:

$$f_s \geq 2f_{\max}$$

This ensures that the signal can be perfectly reconstructed from its samples without aliasing.

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

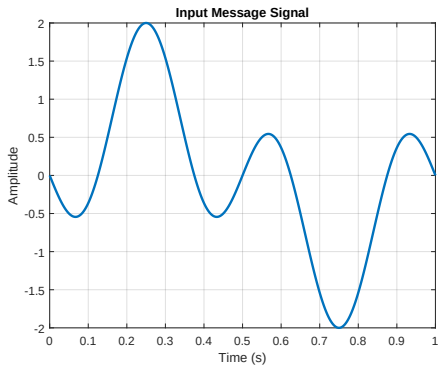

Figure 1: Time-domain representation of the input message signal.

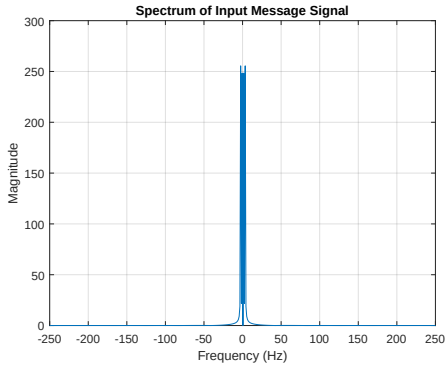

Figure 2: Frequency spectrum of the message signal.

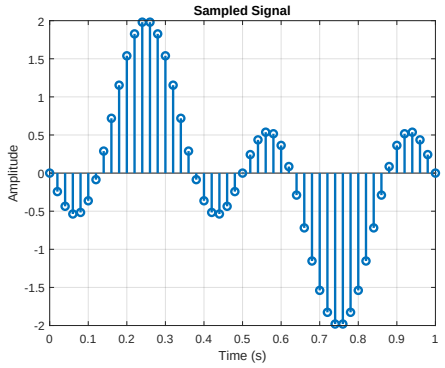

Figure 3: Sampled signal in the time domain.

5 Discussion

term sampling refers to the process of reducing a continuous-time (CT) signal to a discrete-time (DT) signal, which is defined only on a discrete subset of the time axis. By means of a suitable encoding, one often assumes that this discrete set is a subset I of the set Z of integers.Sampling can be regarded as multiplying the time continuous signal g(t) with a train of unit pulses p(t). The time period T between the unit pulses in the pulse train is called the sampling period

6 Experiment 2: Reconstruction of Original Signal

6.1 Objective

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

6.2 Theory

Reconstruction involves: 1. Upsampling: Increasing the sampling rate by inserting zeros between samples. 2. Low-Pass Filtering: Removing high-frequency components while preserving original signal frequencies. 3. Inverse Transformation: Converting the filtered signal from the frequency domain back to the time domain.

The low-pass filter's bandwidth must be equal to the Nyquist frequency to retain only the original frequencies.

6.3 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.

7 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;
```

8 Results

8.1 Spectrum of Sampled Signal

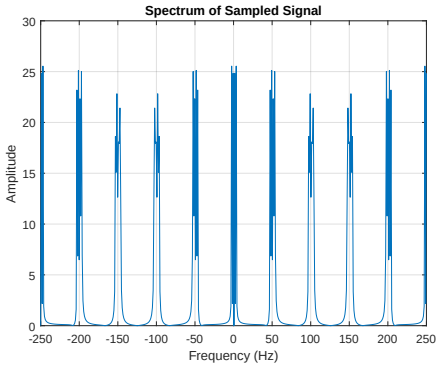

Figure 4: Spectrum of Sampled Signal

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

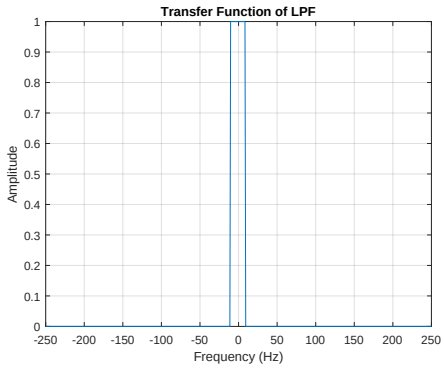

Figure 5: Transfer Function of the Low-Pass Filter

8.3 Spectrum After Low-Pass Filtering

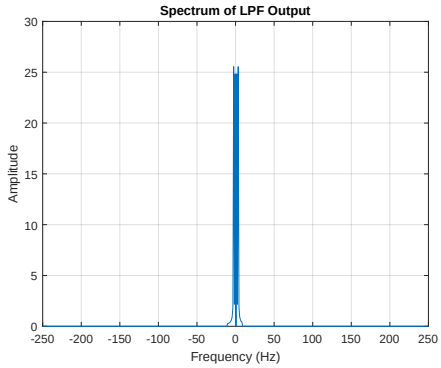

Figure 6: Spectrum After Low-Pass Filtering

8.4 Original vs. Reconstructed Signal

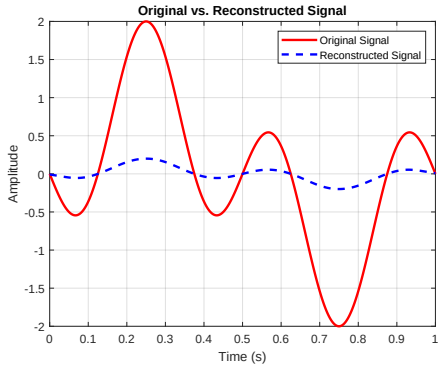

Figure 7: Original Signal vs. Reconstructed Signal

9 Discussion

The low-pass filter effectively removed high-frequency noise and aliasing introduced during sampling. The reconstructed signal closely matches the original signal, validating the Sampling Theorem.The Sampling Theorem (Nyquist-Shannon Theorem) states that a band-limited signal can be perfectly reconstructed if sampled at a rate at least twice its highest frequency.

The statement confirms this by showing that the reconstructed signal aligns well with the original, validating the theory.

Q1.The Nyquist rate is a critical concept in the sampling process because it determines the minimum sampling rate required to accurately capture and reconstruct a continuous-time signal without introducing aliasing.

Q2.The sampling rate is the frequency at which a digitizer samples an input signal. A higher sampling rate means more samples are taken, which results in a more accurate digital representation of the signal. To avoid aliasing, or spectral folding, the sampling rate must be at least twice the highest frequency of the input signal. If the sampling rate is lower than the Nyquist frequency, the signal's components that can't be represented are folded, which results in an incorrect spectral representation.

Q3.A low-pass filter smooths the output signal and removes harmonics when reconstructing a sampled signal.Reducing the filter's bandwidth: May require a higher order filter, which needs more components. Increasing the filter's bandwidth beyond the Nyquist limit: Can cause aliasing, which is when frequencies above the Nyquist limit cause distortion.

Q4.Aliasing is a distortion that occurs when a signal is sampled at an insufficient rate. It results in the appearance of false frequencies, or aliases, in the sampled signal's spectrum. Aliasing appears in the spectrum of the sampled signal as false frequencies, or aliases. It's caused by the mixing of the signal frequencies and the sampling frequency.To avoid aliasing, the sampling rate must be set correctly so that the Nyquist-Shannon sampling theorem is observed.

Q5.If the sample rate is too low, it will not accurately express the original signal and will be distorted, or show aliasing effects, when reproduced. Q6.In reality, the sampling rate required to reconstruct the original signal must be somewhat higher than the Nyquist rate, because of quantization errors introduced by the sampling process. As an example, humans can detect or hear frequenies in the range of 20 Hz to 20,000 Hz.

10 Conclusion

The experiment successfully demonstrated that the low-pass filter successfully suppresses noise and aliasing, enabling accurate signal reconstruction. It validates the Sampling Theorem by demonstrating the importance of meeting its prerequisites and employing appropriate reconstruction techniques. How-

ever, practical implementation may introduce challenges that deviate from this ideal.

11 Extension Task: Quantization

11.1 Objective

To quantize the sampled signal into discrete amplitude levels and analyze quantization error.

11.2 Theory

Quantization maps continuous amplitudes to discrete levels. This introduces quantization error, which can be defined as:

$$\text{Quantization Error} = \text{Original Signal} - \text{Quantized Signal}$$

Increasing the number of quantization levels reduces error but increases data requirements.

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. 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

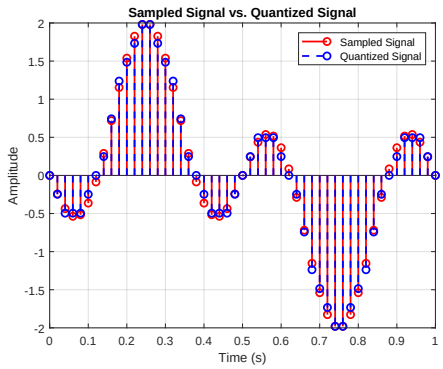

Figure 8: Quantized signal compared to the sampled signal.

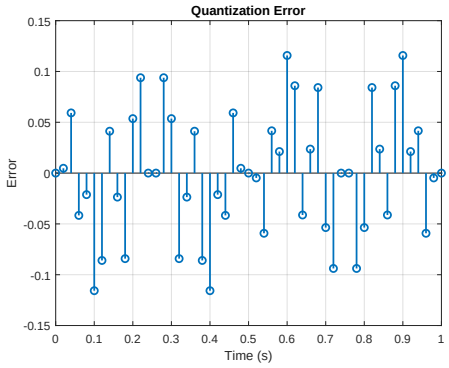

Figure 9: Quantization error over time.

11.6 Discussion

The quantization error decreases as the number of quantization levels increases. Quantization introduces distortion that depends on the signal amplitude and resolution.Increasing the number of quantization levels improves the resolution of the quantization process, allowing the discrete representation to more closely approximate the original signal.

11.7 Conclusion

Quantization effectively maps a signal to discrete levels, enabling digital representation. Higher quantization levels reduce error but increase data requirements.

11.8 Additional Questions

1.Quantization error, also known as quantization noise, is the difference between the actual analog signal value and the quantized digital value in a signal processing system. Quantization error decreases as the number of quantization levels increases. This is because a higher number of levels allows the digital representation to be closer to the actual analog signal.

2.article amsmath pgfplots compat=1.17

Signal-to-Noise Ratio and Quantization Levels

The Signal-to-Noise Ratio (SNR) for a quantized signal can be expressed as:

$$\text{SNR}_{\text{dB}} = 6.02 \cdot b + 1.76$$

where $b = \log_2(L)$ is the number of bits representing the quantization levels L.

SNR vs. Quantization Levels

The following plot shows how SNR increases with the number of quantization levels:

[xlabel=Quantization Levels (L), ylabel=SNR (dB), xmode=log, log basis x=2, grid=major, legend pos=north west] [blue, thick] table [x=Levels, y=SNR, col sep=comma] data.csv; SNR vs. L

It is evident that as L (or b) increases, the SNR improves, reducing the impact of quantization noise.

Discussion

Higher quantization levels lead to better signal fidelity because the quantization error is smaller. However, this comes at the cost of increased storage and computational requirements. Choosing L involves balancing SNR and system constraints.

3.article amsmath

The bitrate R required for a digital communication system is given by:

$$R = f_s \cdot N$$

where:

- f_s is the sampling rate (Hz),
- $N = \log_2(Q)$ is the number of bits per sample, and
- Q is the number of quantization levels.

Example Calculation:

Given:

$$f_s = 44.1\,\text{kHz}, \quad Q = 256$$

1. Calculate N:

$$N = \log_2(256) = 8\,\text{bits/sample}$$

2. Calculate R:

$$R = f_s \cdot N = 44,100 \cdot 8 = 352.8\,\text{kbps}$$

Impact of Changes:

- Increasing f_s directly increases R. For example, doubling f_s to 88.2 kHz results in:
$$R = 88,200 \cdot 8 = 705.6\,\text{kbps}.$$
- Increasing Q increases N. For instance, if $Q = 512$:
$$N = \log_2(512) = 9\,\text{bits/sample},$$
and the new bitrate is:
$$R = 44,100 \cdot 9 = 396.9\,\text{kbps}.$$

4.Sampling and quantization work in sequence to digitize an analog signal:

Sampling captures the signal at discrete intervals. Quantization assigns each sample a discrete digital value for representation and processing.

APPLICATIONS.

Digital Audio Recording: Sampling is used to convert analog sound waves into digital audio signals (e.g., in MP3, WAV formats). Music Production: Quantization enables audio effects, compression, and digital mixing by approximating amplitudes. Digital Photography: Cameras use sampling to capture light intensity and quantization to represent colors in digital formats like JPEG or RAW. Medical Imaging: Techniques like CT scans, MRIs, and ultrasounds rely on sampling and quantization to create detailed images of the body.

5.Trade-offs Sampling rate Higher sampling rates improve the digital representation of an analog signal, but they also require more storage and computational power. Quantization levels Higher quantization levels improve signal quality, but they also create larger data sets and may slow down conversion rates. System complexity and cost Increasing sampling rates and quantization levels can increase the complexity and cost of the system. Considerations Aliasing Choosing an appropriate sampling rate can help minimize aliasing and ensure that the system only gathers what's needed. Signal quality The desired signal quality should be considered when choosing the quantization resolution. Encoding strategy The kind of signal and the application should be considered when choosing an encoding strategy

12 References

John G.Proakis, Masoud Salehi "Digital Communications" 5th edition

Robert G. Gallager"Principles of Digital Communication" 1996