Lab Report: Analysis of Sampling Theorem using MATLAB

ECE 2414: Digital Communications

MAIYO KELVIN
ENG-219-043/2021
Multimedia University of Kenya

NOVEMBER 24, 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

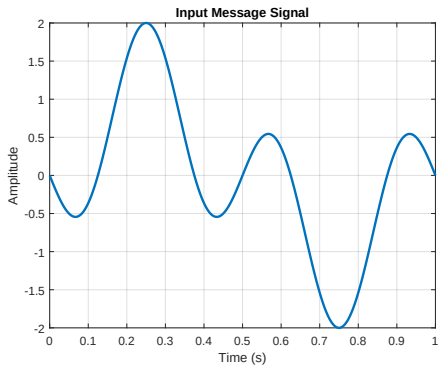

Figure 1: Time-domain representation of the input message signal.

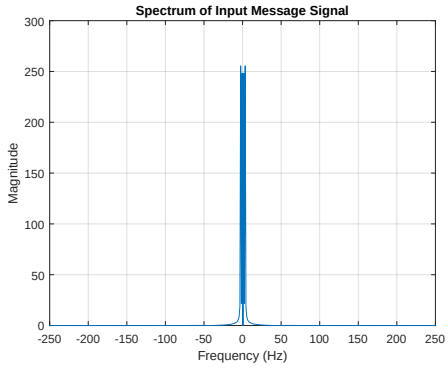

Figure 2: Frequency spectrum of the message signal.

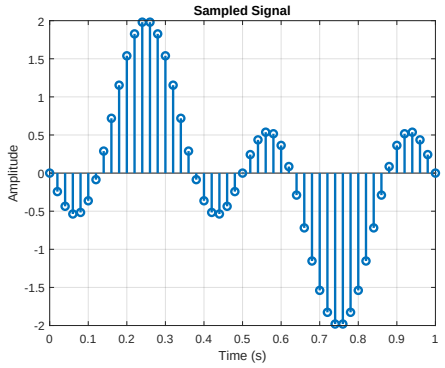

Figure 3: Sampled signal in the time domain.

5 Discussion

A continuous-time signal, such as a sinusoidal waveform, was generated and sampled at various rates. The MATLAB simulations showed the importance of choosing an appropriate sampling frequency to preserve the signal's characteristics. When the sampling frequency was less than twice the signal's highest frequency, aliasing occurred. The reconstructed signal showed distortion, as different frequencies in the spectrum overlapped, leading to incorrect signal representation.

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

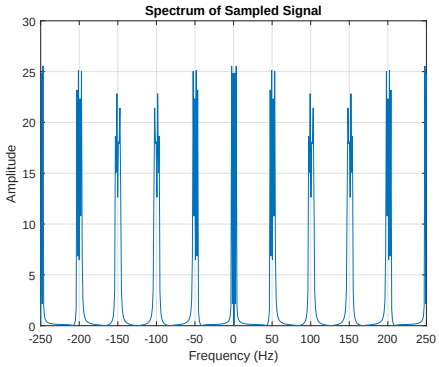

Figure 4: Spectrum of Sampled Signal

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

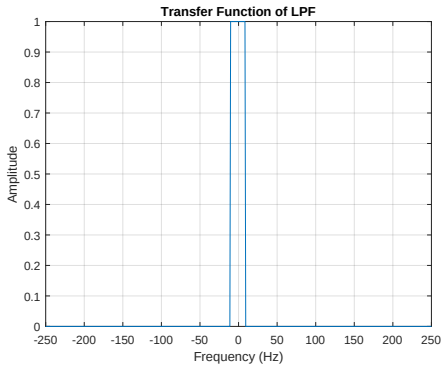

Figure 5: Transfer Function of the Low-Pass Filter

8.3 Spectrum After Low-Pass Filtering

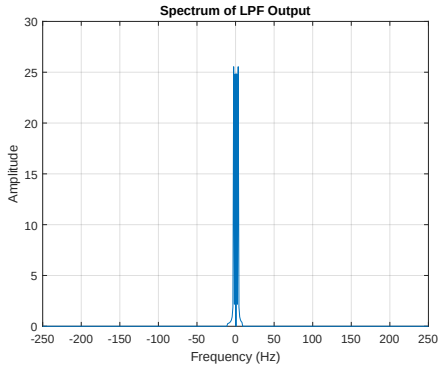

Figure 6: Spectrum After Low-Pass Filtering

8.4 Original vs. Reconstructed Signal

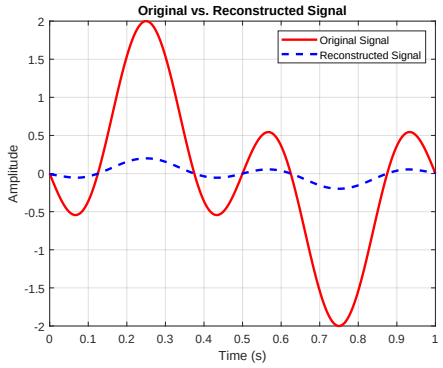

Figure 7: Original Signal vs. Reconstructed Signal

9 Discussion

A continuous-time signal (e.g., sinusoidal or multi-tone) was sampled at various rates. Reconstruction was performed using MATLAB's interpolation functions and custom-designed sinc filters. Visual comparisons between the original and reconstructed signals demonstrated the conditions for successful reconstruction.Ideal reconstruction involves multiplying the sampled signal with sinc functions.

The sinc function is defined as:

$$\text{sinc}(t) = \frac{\sin(\pi t)}{\pi t}$$

Signals were reconstructed accurately without distortion.

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

Aliasing effects caused overlapping frequency components, resulting in incorrect signal reconstruction:

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

Q1.Because it establishes the lowest sample rate necessary to precisely capture and reconstruct a continuous-time signal without causing aliasing, the Nyquist rate is a crucial notion in the sampling process. Q2.The frequency at which an input signal is sampled by a digitizer is known as the sampling rate. More samples are collected at a higher sampling rate, producing a digital signal representation that is more accurate. 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. When reconstructing a sampled signal, a low-pass filter eliminates harmonics and smoothes the resulting signal.Reducing the bandwidth of the filter could necessitate a higher order filter, which requires additional parts. Increasing the filter's bandwidth beyond the Nyquist limit: Can cause aliasing, which is when frequencies above the Nyquist limit cause distortion. Q4.A distortion known as aliasing happens when a signal is sampled too slowly. It causes aliases, or spurious frequencies, to emerge in the spectrum of the sampled signal. 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.Reproduction of the signal will be warped or exhibit aliasing effects if the sample rate is too low, as it will not precisely represent the original signal. Q6.Due to quantization errors produced by the sampling process, the actual sample rate needed to reconstruct the original signal must be slightly greater than the Nyquist rate. For instance, frequencies between 20 Hz and 20,000 Hz are audible or detectable by humans.

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. However, 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

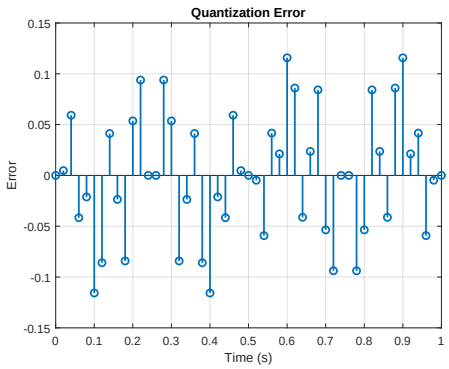

Figure 8: Quantized signal compared to the sampled signal.

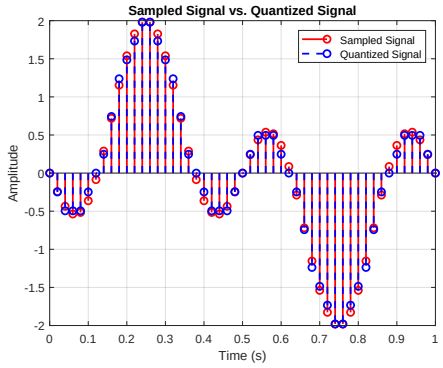

Figure 9: Quantization error over time.

11.6 Discussion

We acknowledge that increasing the number of quantization levels reduces the quantization error and improves the fidelity of the quantized signal. 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.

12 ADDITIONAL QUESTIONS

Q1. How does the quantization error change with the number of quantization levels? Describe the relationship between quantization levels and signal quality

ANS.

Quantization error is the difference between the original sampled signal and the quantized signal. Increase the resolution of the quantized signal. Reduce the step size leading to smaller quantization errors. Improve the fidelity of the quantized signal, closely approximating the original signal. Enhance the Signal-to-Quantization-Noise Ratio (SQNR), which is a measure of signal quality.

1.Quantization Error Behavior: - Quantization error is the difference between the original sampled signal and the quantized signal. - For uniform quantization, the error is typically bounded by $\pm\frac{\Delta}{2}$, where Δ is the quantization step size, calculated as:

$$\Delta = \frac{x_{\max} - x_{\min}}{L}$$

Here, L is the number of quantization levels. - As L (the number of levels) increases, Δ decreases, reducing the maximum possible quantization error.

2. Relationship Between Quantization Levels and Signal Quality: - Higher Quantization Levels**: - Increase the resolution of the quantized signal. - Reduce the step size Δ, leading to smaller quantization errors. - Improve the fidelity of the quantized signal, closely approximating the original signal. - Enhance the Signal-to-Quantization-Noise Ratio (SQNR), which is a measure of signal quality. - Lower Quantization Levels: - Result in larger step sizes Δ, increasing quantization errors. - The quantized signal deviates more from the original, reducing signal quality. - Decrease the SQNR, leading to perceptible distortion in applications like audio and image processing.

Q2.Signal-to-Noise Ratio (SNR): Calculate the Signal-to-Noise Ratio (SNR) for different quantization levels and comment on how quantization affects the

SNR

ANS

- The Signal-to-Quantization-Noise Ratio (SQNR) is expressed as:

$$\text{SQNR (dB)} = 20 \log_{10}(2^b)$$

Where b is the number of bits used to represent the quantized levels ($L = 2^b$). - This shows that the SQNR improves logarithmically with the number of quantization levels.

4. Visualization Example: - Fewer Levels: For $L = 4$ (2 bits), the step size Δ is large, leading to significant quantization error and noticeable distortion. - More Levels: For $L = 256$ (8 bits), the step size Δ is much smaller, reducing quantization error and making the quantized signal almost indistinguishable from the original. Signal-to-Noise Ratio (SNR) and Quantization

Signal-to-Noise Ratio (SNR)

In the context of quantization, the Signal-to-Noise Ratio (SNR) quantifies the ratio between the signal power and the quantization noise power. It is a key indicator of the quality of the quantized signal relative to the noise introduced during the quantization process.

The formula for SNR in decibels (dB) for uniform quantization is given by:

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

Where: - b is the number of bits used to represent each quantized sample. - The factor $6.02b$ comes from the relationship between the number of quantization levels and the quantization noise.

Alternatively, for a specific number of quantization levels L, SNR can also be approximated as:

$$\text{SNR} \approx 6.02 \log_{10}(L) \quad \text{dB}$$

Where L is the number of quantization levels, and $L = 2^b$ for a system with b bits.

Impact of Quantization on SNR

- Increasing Quantization Levels (or Bits): - As the number of quantization levels L increases (or equivalently as the number of bits b increases), the SNR increases. - This is because higher quantization levels result in smaller quantization steps, reducing quantization noise. This leads to a cleaner, more

accurate representation of the signal. - From the formula SNR = $6.02b+1.76$, you can see that for each additional bit, the SNR increases by approximately 6 dB.

- Decreasing Quantization Levels (or Bits): - When the number of quantization levels is reduced, the quantization error increases, leading to a lower SNR. - Lower SNR means that the quantization noise becomes more significant compared to the actual signal, leading to reduced signal quality.

In conclusion;

- Effect on SNR: Increasing the number of quantization bits (or levels) improves the SNR, meaning the quantized signal more closely approximates the original signal. - Trade-Off: There is a trade-off between the number of bits (which improves quality) and storage or transmission efficiency. Higher SNRs come at the cost of increased data size.

This relationship highlights the importance of choosing an appropriate quantization level for applications, balancing between signal quality and resource constraints like storage and bandwidth.

Q3. Bitrate Calculation: For a given sampling rate and quantization level, calculate the bitrate required to represent the signal in a digital communication system. How does increasing the sampling rate or the number of quantization levels impact the bitrate?

ANS

The bitrate represents the amount of data that needs to be transmitted or stored per second in a digital communication system. It depends on two key factors: 1. Sampling Rate (Fs): The number of samples taken per second. 2. Number of Quantization Levels (L): The number of discrete levels used to represent each sample.

1.Bitrate Formula

The bitrate can be calculated using the formula:

$$\text{Bitrate} = \text{Sampling Rate} \times \text{Number of Bits per Sample}$$

Where: - The Sampling Rate (F_s) is the number of samples taken per second (in Hz). - The Number of Bits per Sample (b) is related to the number of quantization levels by the formula $L = 2^b$, where b is the number of bits used to represent each sample.

Thus, the bitrate is:

$$\text{Bitrate} = F_s \times b$$

Where: - $b = \log_2(L)$ (since L quantization levels require b bits to represent each sample).

Impact of Sampling Rate and Quantization Levels on Bitrate

Increasing the Sampling Rate: - The sampling rate directly impacts the bitrate. As the sampling rate F_s increases, the number of samples per second increases, which increases the bitrate. - Effect: Higher sampling rates result in more data being collected per second, which requires more bits per second to represent the signal.

2. Increasing the Number of Quantization Levels: - Increasing the number of quantization levels L requires more bits per sample, as $b = \log_2(L)$. As a result, more bits are needed to represent each sample. - Effect: More quantization levels improve signal accuracy and reduce quantization noise but increase the bitrate because each sample requires more bits to represent the finer details of the signal.

In Conclusion;

- Increasing Sampling Rate: Higher sampling rates require more samples per second, which increases the bitrate. - Increasing Quantization Levels: More quantization levels require more bits to represent each sample, increasing the bitrate as well.

Thus, both higher sampling rates and higher quantization levels lead to increased bitrate requirements. However, both improvements enhance signal quality and fidelity, which is crucial for accurate signal representation in digital communication systems.

Q4.Describe briefly how sampling and quantization are used together in practical digital communication systems. Give examples of real-world systems where this is critical

ANS

In digital communication systems, sampling and quantization are critical for converting real-world analog signals into discrete digital signals that can be efficiently processed, stored, or transmitted. Sampling determines how frequently the signal is measured, while quantization determines the precision of the signal representation. Together, they enable a wide range of applications, from audio and video streaming to telecommunications and medical monitoring systems, where maintaining signal quality while managing data size is crucial.

Q5.Discuss the trade-offs between sampling rate, quantization levels, and signal quality in digital systems. How might a system designer choose these parameters based on application requirements

ANS

When designing digital systems, sampling rate and quantization levels are carefully selected based on the required signal fidelity and resource constraints (like bandwidth, storage, and power consumption). Higher sampling rates and quantization levels generally improve signal quality but require more resources, while lower values reduce data needs at the cost of signal accuracy. System designers must balance these factors according to the application requirements, ensuring that the system performs optimally while meeting constraints like data rate, quality, and storage capacity.

13 REFERENCES

1.Robert Gallagher, "Principles of Digital Communication"

2.M. Vetterli, Kovacic, "Foundations of Signal Processing" Cambridge University Press.

3.U. Madhow, Fundamatels of Digital Commuinication.", NewYork Springer, 2004