Laboratory 5: Sampling

Aim: To sample a continuous-time sinusoidal signal and analyze the effects of different sampling rates.

Theory: The Nyquist sampling theorem states that a signal must be sampled at a rate at least twice its highest frequency component to be accurately reconstructed.

Part 1:
a. Define a continuous-time sinusoidal signal.
b. Sample the signal at various rates:
  - Above the Nyquist rate.
  - At the Nyquist rate.
  - Below the Nyquist rate.
c. Plot the original signal and its sampled versions.

MATLAB Code:

```
% Define parameters
fs_orig = 1000; % Original sampling frequency (Hz)
f_signal = 50;  % Signal frequency (Hz)
t = 0:1/fs_orig:0.1; % Time vector
x = sin(2*pi*f_signal*t); % Continuous-time signal

% Sampling frequencies
fs1 = 200; % Above Nyquist rate
fs2 = 100; % At Nyquist rate
fs3 = 50;  % Below Nyquist rate
```

Part 2:

Aim: To observe aliasing effects when sampling a signal below the Nyquist rate.

Theory: Aliasing occurs when a signal is sampled below its Nyquist rate, resulting in an overlapping of spectral components.

Procedure:
a. Define a sinusoidal signal with a frequency exceeding half the sampling rate.
b. Sample the signal and observe the aliasing effect.

```
% Define parameters
fs = 100; % Sampling frequency (Hz)
f_signal = 75; % Signal frequency (Hz) > fs/2
t = 0:1/1000:0.1; % Time vector
x = sin(2*pi*f_signal*t); % Continuous-time signal

% Sampled signal
n = 0:1/fs:0.1;
x_sampled = sin(2*pi*f_signal*n);
```

Part 3:

Aim: To reconstruct a continuous-time signal from its sampled version using interpolation.

Theory: Signal reconstruction can be achieved using sinc interpolation or low-pass filtering to remove high-frequency components introduced by sampling.

Procedure:
a. Sample a sinusoidal signal at the Nyquist rate.
b. Use MATLAB’s `interp1` function for reconstruction.
c. Compare the reconstructed signal with the original signal.

```
% Define parameters
fs = 200; % Sampling frequency (Hz)
f_signal = 50; % Signal frequency (Hz)
t = 0:1/1000:0.1; % Time vector
x = sin(2*pi*f_signal*t); % Continuous-time signal

% Sampled signal
n = 0:1/fs:0.1;
x_sampled = sin(2*pi*f_signal*n);

% Reconstruction
t_recon = 0:1/1000:0.1; % Fine time vector
x_recon = interp1(n, x_sampled, t_recon, 'spline');
```

Questions:
1. Explain the Nyquist sampling theorem and its significance.
2. Describe the effect of sampling below the Nyquist rate.
3. What are the limitations of sinc interpolation in signal reconstruction?
4. How can aliasing be avoided in practical applications?