Signals and Systems

Lab 5 Report

Erkan Tiryakioğlu

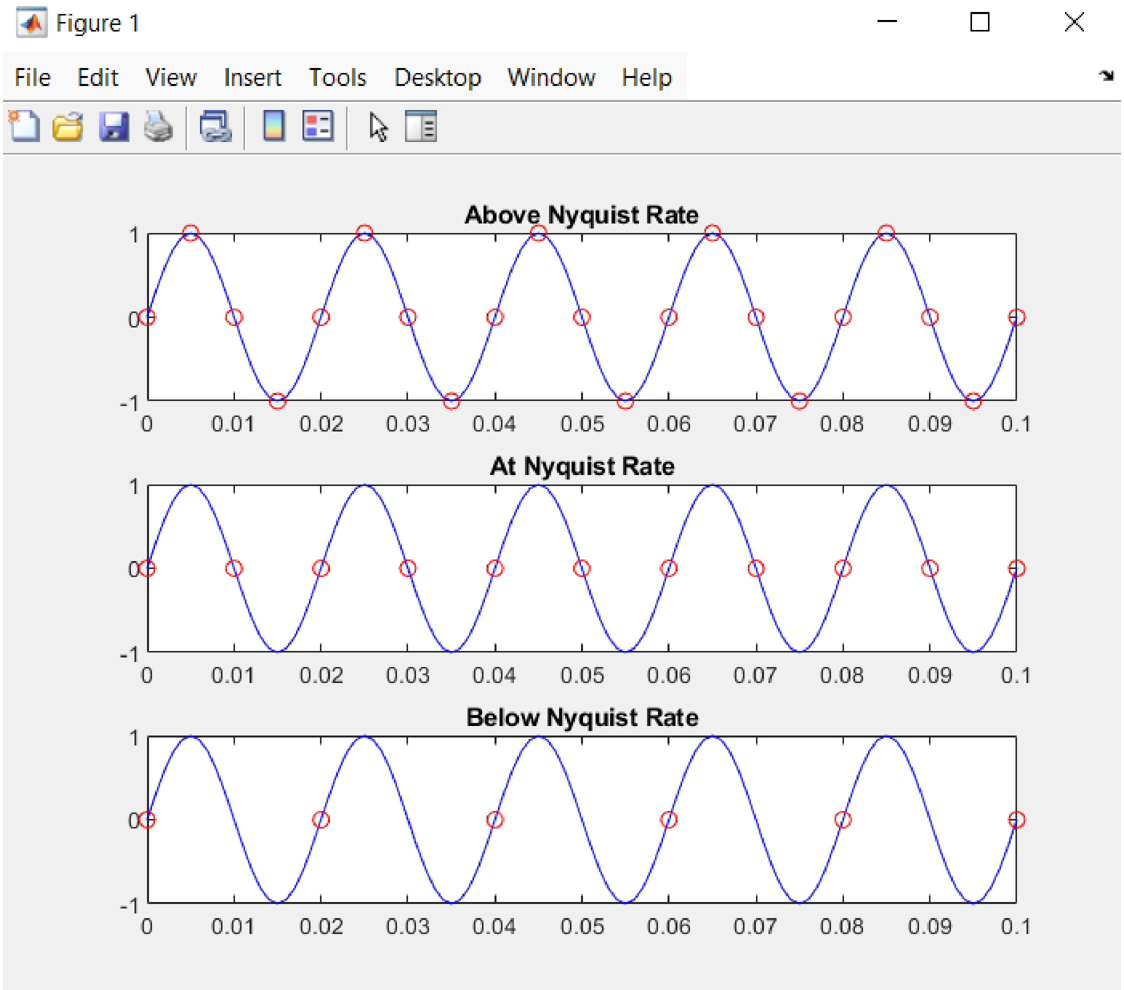
Part 1: Sampling at Different Rates

```
fs_orig = 1000;
f_signal = 50;
t = 0:1/fs_orig:0.1;
x = sin(2*pi*f_signal*t);

fs1 = 200;
fs2 = 100;
fs3 = 50;

n1 = 0:1/fs1:0.1; x1 = sin(2*pi*f_signal*n1);
n2 = 0:1/fs2:0.1; x2 = sin(2*pi*f_signal*n2);
n3 = 0:1/fs3:0.1; x3 = sin(2*pi*f_signal*n3);

figure;
subplot(3,1,1); plot(t, x, 'b', n1, x1, 'ro'); title('Above Nyquist Rate');
subplot(3,1,2); plot(t, x, 'b', n2, x2, 'ro'); title('At Nyquist Rate');
subplot(3,1,3); plot(t, x, 'b', n3, x3, 'ro'); title('Below Nyquist Rate');
```

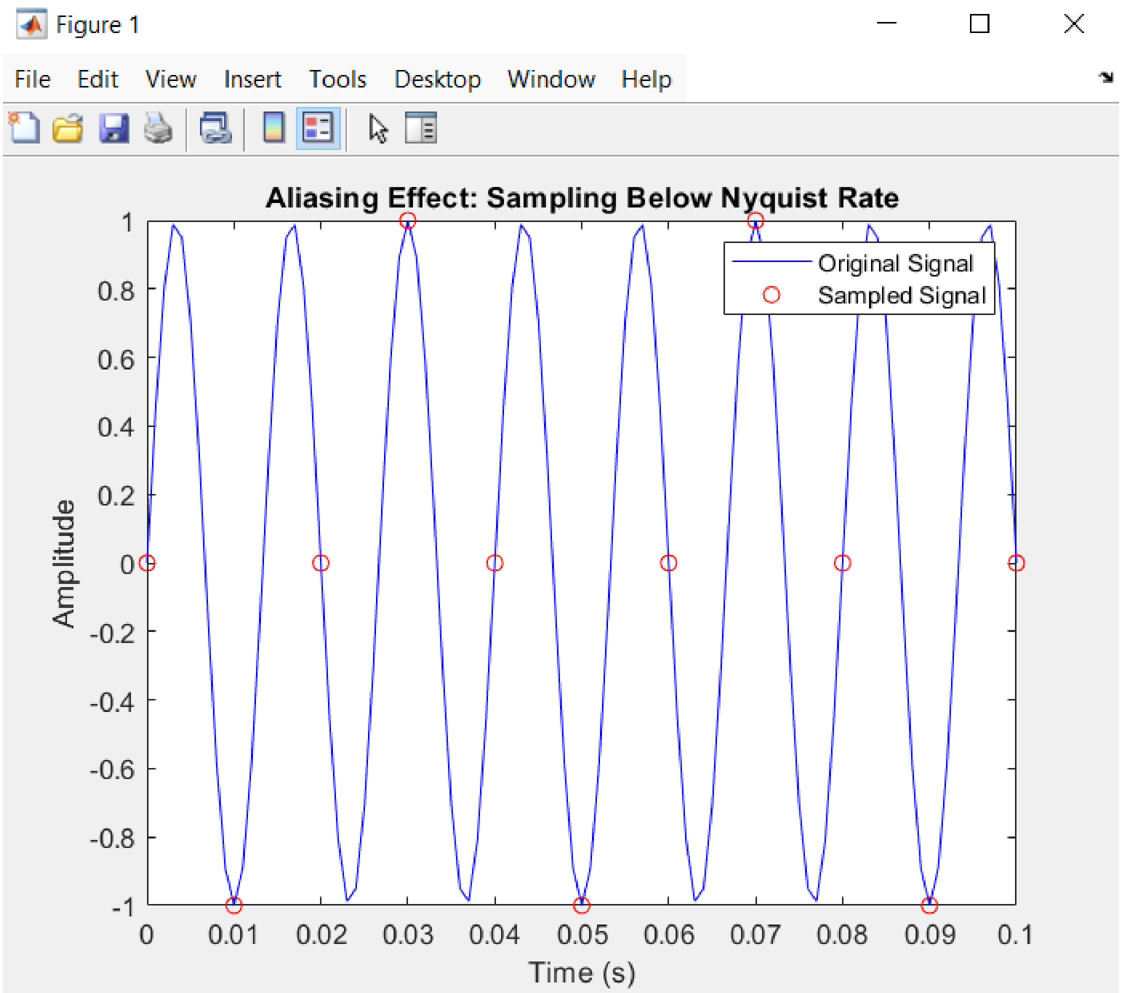


Part 2: Observing Aliasing Effect

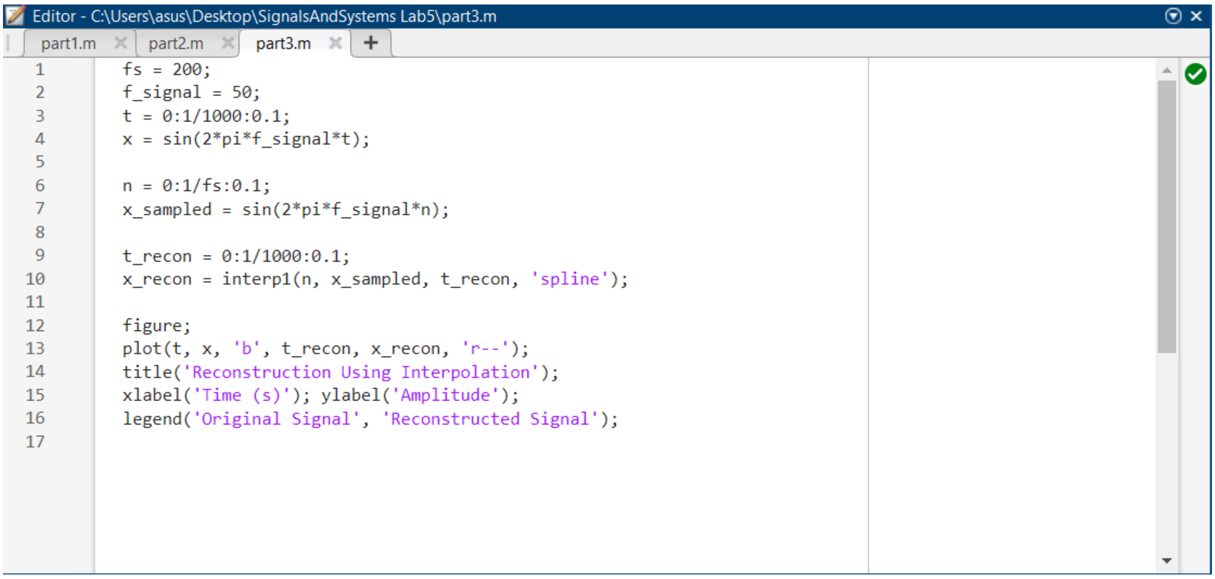
```
fs = 100;
f_signal = 75;
t = 0:1/1000:0.1;
x = sin(2*pi*f_signal*t);

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

figure;
plot(t, x, 'b', n, x_sampled, 'ro');
title('Aliasing Effect: Sampling Below Nyquist Rate');
xlabel('Time (s)'); ylabel('Amplitude');
legend('Original Signal', 'Sampled Signal');
```



Part 3: Reconstructing Signal Using Interpolation

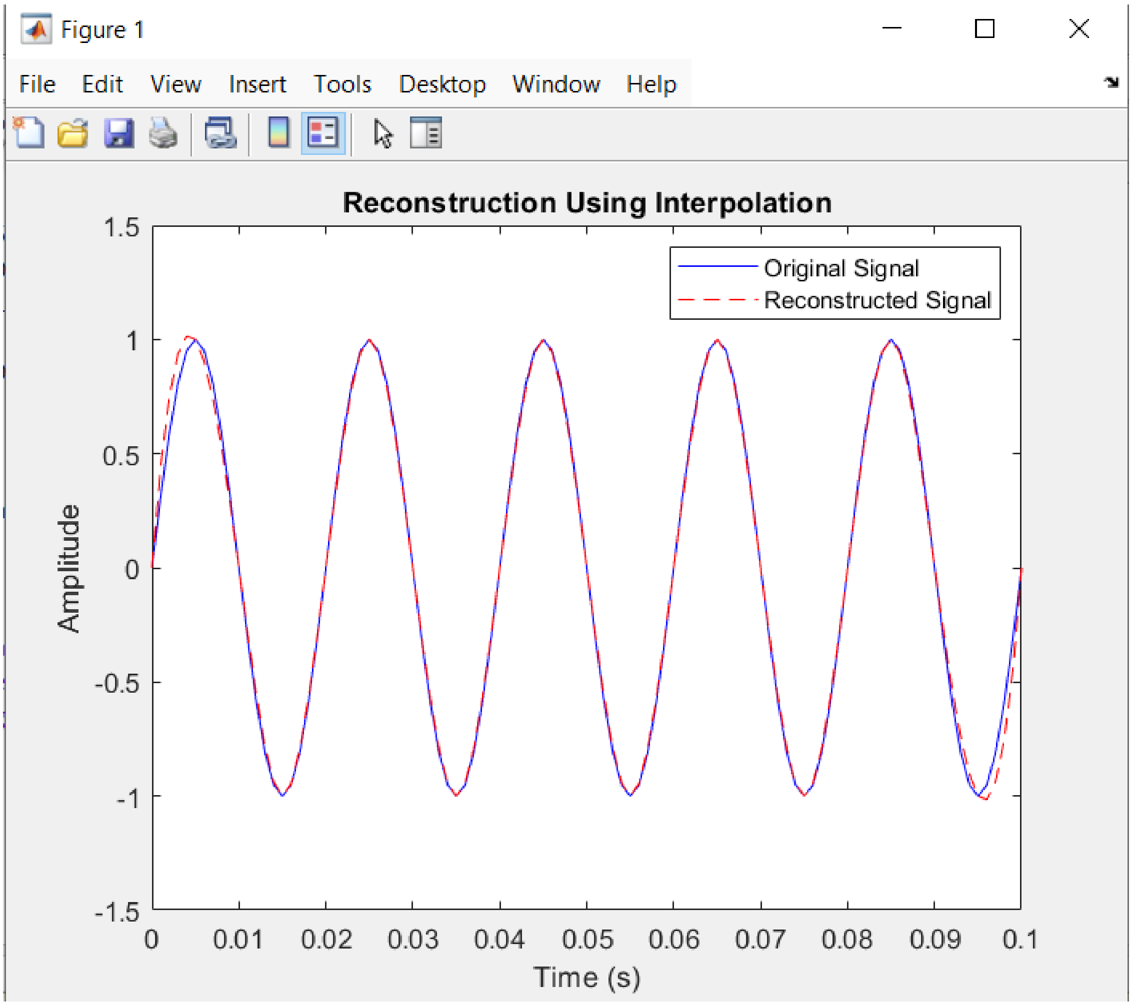



```matlab
fs = 200;
f_signal = 50;
t = 0:1/1000:0.1;
x = sin(2*pi*f_signal*t);

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

t_recon = 0:1/1000:0.1;
x_recon = interp1(n, x_sampled, t_recon, 'spline');

figure;
plot(t, x, 'b', t_recon, x_recon, 'r--');
title('Reconstruction Using Interpolation');
xlabel('Time (s)'); ylabel('Amplitude');
legend('Original Signal', 'Reconstructed Signal');
```



Questions

Nyquist Sampling Theorem

A signal must be sampled at a rate at least twice its highest frequency component to be accurately reconstructed.

Effect of Sampling Below Nyquist Rate

Aliasing occurs, leading to misrepresentation of the signal's frequency components.

Limitations of Sinc Interpolation

Practical implementations use an approximation of the sinc function, which may cause inaccuracies.

Avoiding Aliasing in Practice

Use appropriate sampling rates above the Nyquist rate and employ anti-aliasing filters before sampling.