SNS lab 12

Name: basil khowaja (bk08432)

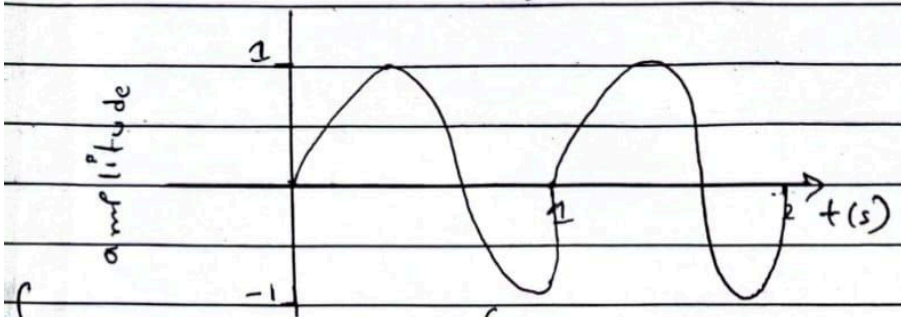
Lab 12 **Date:**

Signals and Systems

Task 1

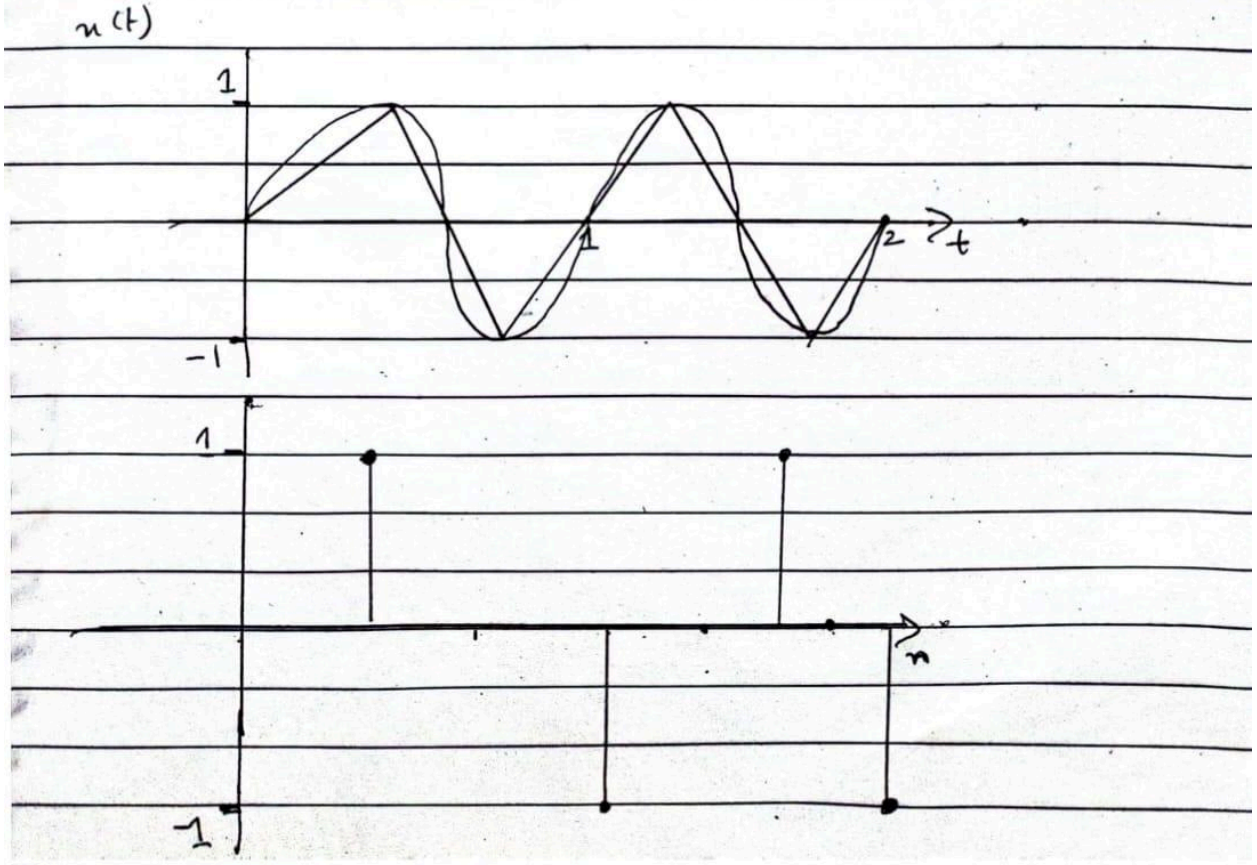
Name: Basil Khowaja bk08432

→ task 1:



⇒ $T_s = 0.25s$ → $f_s = 4Hz$

$f_d = \frac{f_o}{f_s} = \frac{1}{4}$ ⇒ 4 samples in one cycle



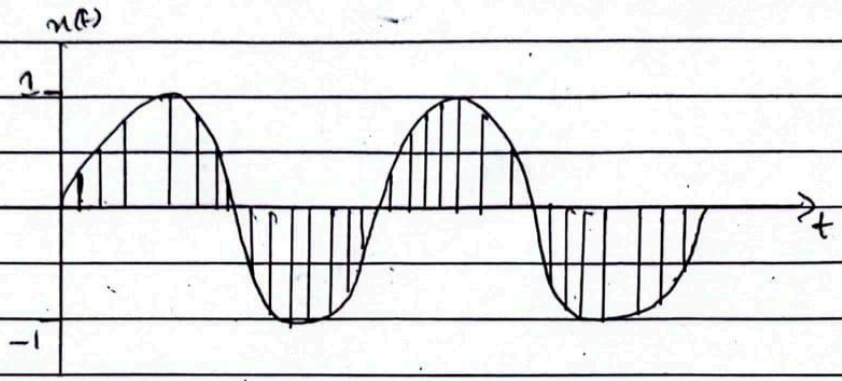
SNS lab 12

Name: basil khowaja (bk08432)

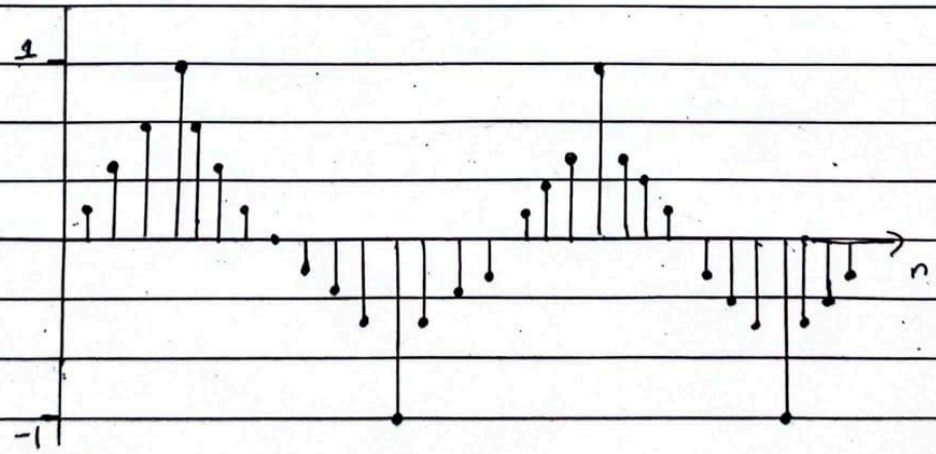
Date:

2) $f_d = \frac{1}{16} \Rightarrow f_s = 16hz$

↳ this means 16 samples in one cycle



x[n]



SNS lab 12

Name: basil khowaja (bk08432)

Task 2:

```
%name: basil khowaja bk08432
f0 = 2; % Fundamental frequency of the sine wave (in Hz)
t = 0:0.01:1; % Time vector for the original signal
% Generate the original sine wave signal
x = sin(2*pi*f0*t);
% Initialize a figure with custom size
figure;
set(groot, 'DefaultFigurePosition', [1000,1000,2000,2000])
% Plot the original signal
subplot(5,1,1)
plot(t, x, 'm', 'Linewidth', 2)
title('Original Signal')
xlabel('Time (s)')
ylabel('Amplitude')
fs= [1.5, 2, 4, 16]; % Sampling frequencies (in Hz)
count = 2; % Counter for subplot indexing
% Loop through each sampling frequency
for i = fs
dur = 1 - 1/i; % Calculate duration for sampling
% Generate the sampled signal
[y, tt] = sampled(f0, i, dur);
% Plot the original signal and the sampled signal
subplot(5,1,count)
```

SNS lab 12

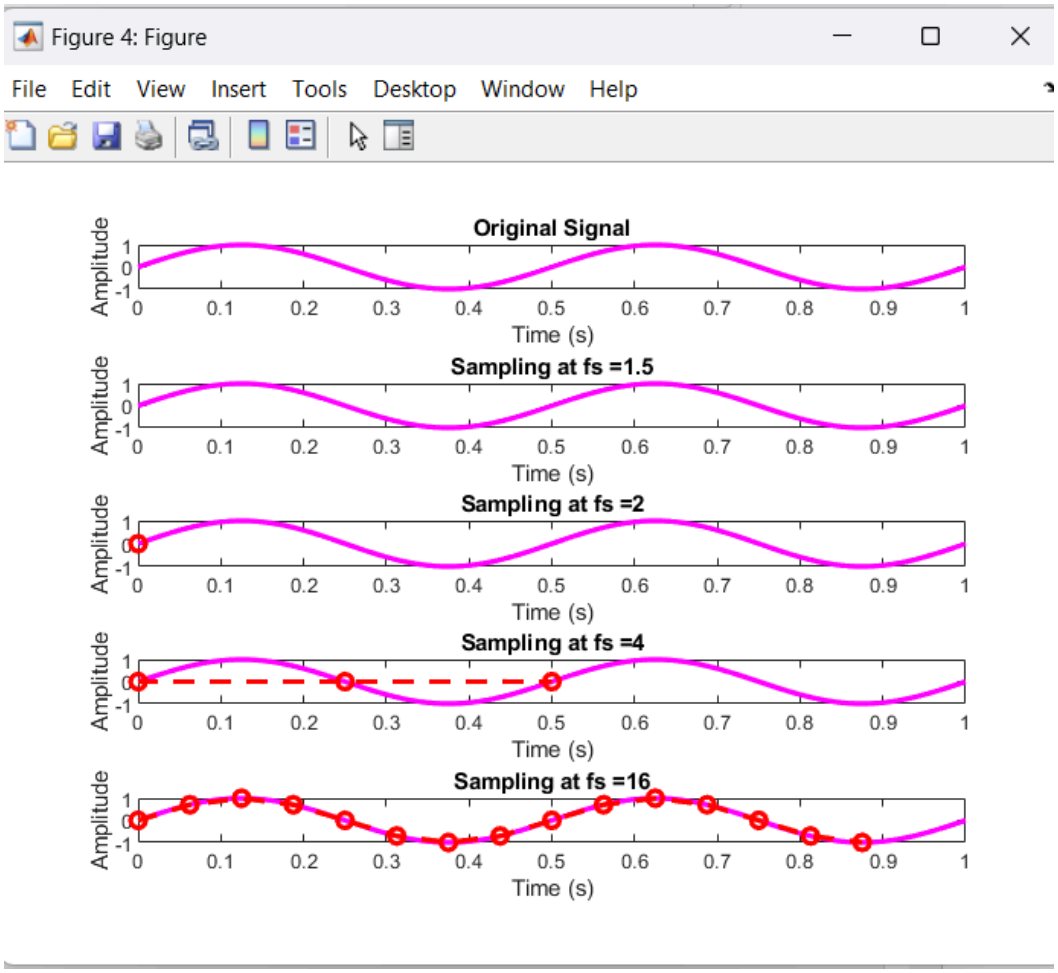
Name: basil khowaja (bk08432)

```
count = count + 1;
% Plot the original signal
plot(t, x, 'm', 'Linewidth', 2)
hold on
% Plot the sampled signal
plot(tt, y, 'r--o', 'Linewidth', 2)
hold off
title(['Sampling at fs =' num2str(i)])
xlabel('Time (s)')
ylabel('Amplitude')
end
```

SNS lab 12

Name: basil khowaja (bk08432)

Output:



observation:

We are creating a simple sine wave signal that oscillates smoothly over time. Firstly we set signal frequency and duration. Then we simulate the signal and plot it. Additionally, we're simulating how this signal would look if we took samples of it at a certain rate. We observe that as we increase the sampling frequency, we are getting more and more samples per cycle. The greater the sampling frequency the better the approximation of the signal because there is more and more information that is not being lost. When the sampling frequency is low then a large amount of information is lost. Also to add on here, the signal is most comprehensible at fs = 16 because then the value satisfies the Nyquist theorem.

SNS lab 12

Name: basil khowaja (bk08432)

Task3:

```
clc

clear all

close all

warning off
```

```
Fs = [1000, 2000, 4000, 16000, 44100]; % Sampling frequency in hertz e.g 1000Hz

ch = 1; % Number of channels--2 options--1 (mono) or 2 (stereo)

datatype = 'uint8';

nbits = 16; % 8, 16, or 24

Nseconds = 5;
```

```
for i = Fs

   % to record audio data from an input device such as a microphone

   recorder = audiorecorder(i, nbits, ch);

   disp('Start speaking..')

   % Records audio to audiorecorder, holds control until recording completes

   recordblocking(recorder, Nseconds);

   disp('End of Recording.');

   % Stores recorded audio signal in numeric array

   x = getaudiodata(recorder, datatype);

   % Write audio file

   audiowrite("test" + num2str(i) + ".wav", x, i);

end
```

SNS lab 12

Name: basil khowaja (bk08432)

```
Start speaking..
End of Recording.
Start speaking..
End of Recording.
Start speaking..
End of Recording.
Start speaking..
End of Recording.
Start speaking..
End of Recording.
```

File sizes:

test1000 = 9.8KB

test2000 = 19.5KB

test4000 = 39.1KB

test16000 = 156KB

test44100 = 430KB

Observation :

When the frequency is set to 1000 Hz, the taped voice is too distorted to be heard. But if we raise the sampling frequency, the speech that was recorded sounds better and is easier to understand. As the sampling rate goes up, so does the file size of the recorded.wav files. This is because files with faster sampling rates are bigger because they store more data per second. The 44100Hz file can be heard the best because of the Nyquist theorem, which says that 44.1kHz is greater than 2(20kHz), making it easy to reconstruct the original sound. Because of this, a sampling rate of 44100 Hz is often used for CD-quality sound and to make good records.

SNS lab 12

Name: basil khowaja (bk08432)

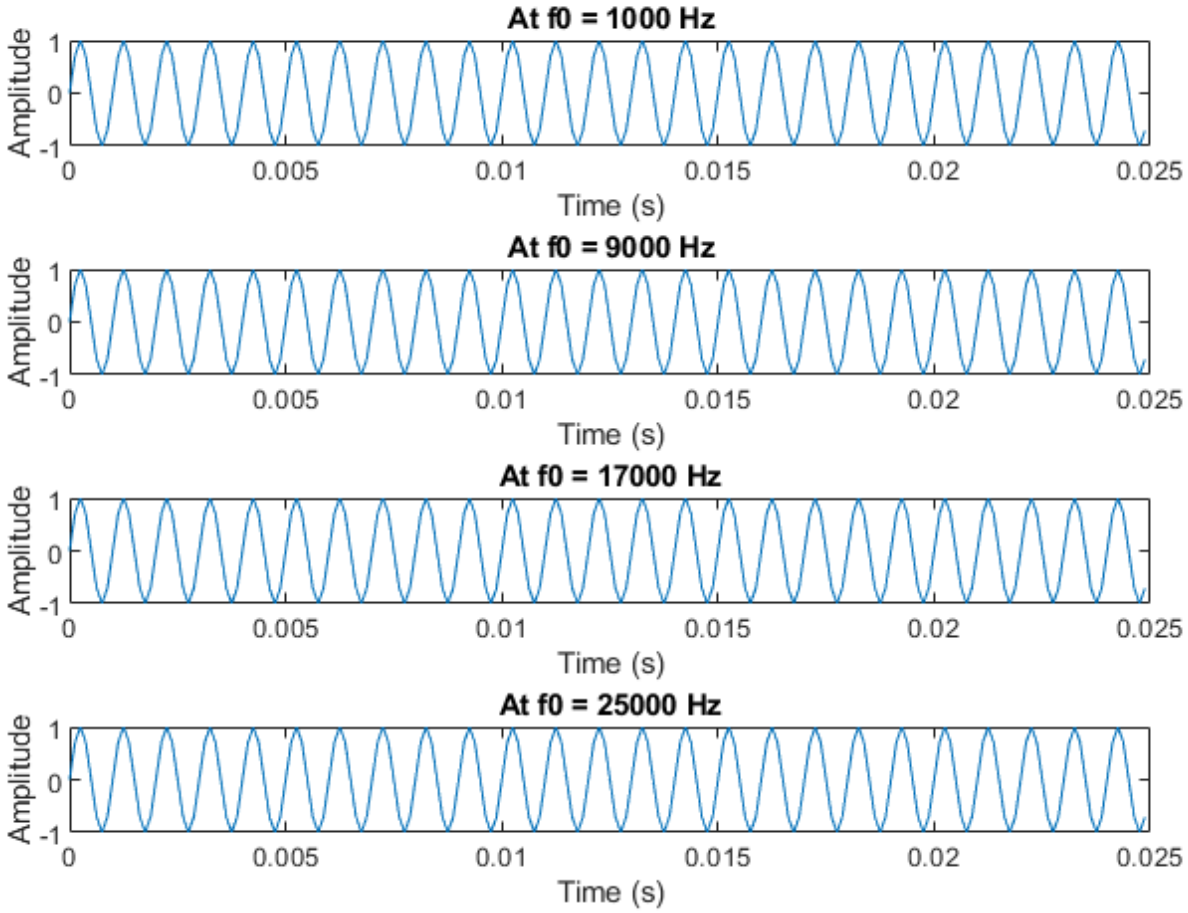
Task4:

```
clc;
close all;
fs = 8000; % Sampling frequency (in Hz)
f0 = [1000, 9000, 17000, 25000]; % Tone frequencies (in Hz)
dur = 3; % Duration of each signal (in seconds)
% Initialize a figure with custom size
figure;
set(groot, 'DefaultFigurePosition', [750,750,2000,2000])
% Loop through each frequency in f0
for i = 1:length(f0)
% Generate time vector
t = 0:1/fs:dur;
% Generate sine wave signal
x = sin(2*pi*f0(i)*t);
% Plot the first 200 samples of the signal
subplot(4, 1, i);
plot(t(1:200), x(1:200));
title(['At f0 = ' num2str(f0(i)) ' Hz']);
xlabel('Time (s)');
ylabel('Amplitude');
% Play the signal
soundsc(x, fs);
% Pause to allow sound playback
pause(dur);
```

SNS lab 12

Name: basil khowaja (bk08432)

```
end
```



Observation:

The tone at 1 kHz sounds natural and is within the conventional audible range of human hearing. However, as the frequency increases to 9 kHz, 17 kHz, and 25 kHz, the tones appear increasingly high pitched and identical, making it more difficult to differentiate between them. This is because the tones are increasingly similar to each other.

SNS lab 12

Name: basil khowaja (bk08432)

Task5

```
f0 = 10; % Original frequency of the sine wave
dur = 3; % Duration of the signal (in seconds)
t = 0:1/1000:dur; % Time vector for the original signal
% Generate and plot the original sine wave
x_original = sin(2*pi*f0*t);
figure;
set(groot, 'DefaultFigurePosition', [750,750,2000,2000]) % Set figure position
subplot(4, 1, 1)
plot(t, x_original);
title("When fs = 1000 Hz");
xlabel('Time');
ylabel('Amplitude');
% Define sampling rates for different cases
fs = [5, 20, 40]; % Sampling rates
% Plot sine wave signals with different sampling rates
for i = 1:length(fs)
% Generate the sampled sine wave
t_sampled = 0:1/fs(i):dur;
x_sampled = sin(2*pi*f0*t_sampled);
% Plot the sampled sine wave
subplot(4, 1, i+1)
plot(t_sampled, x_sampled);
```

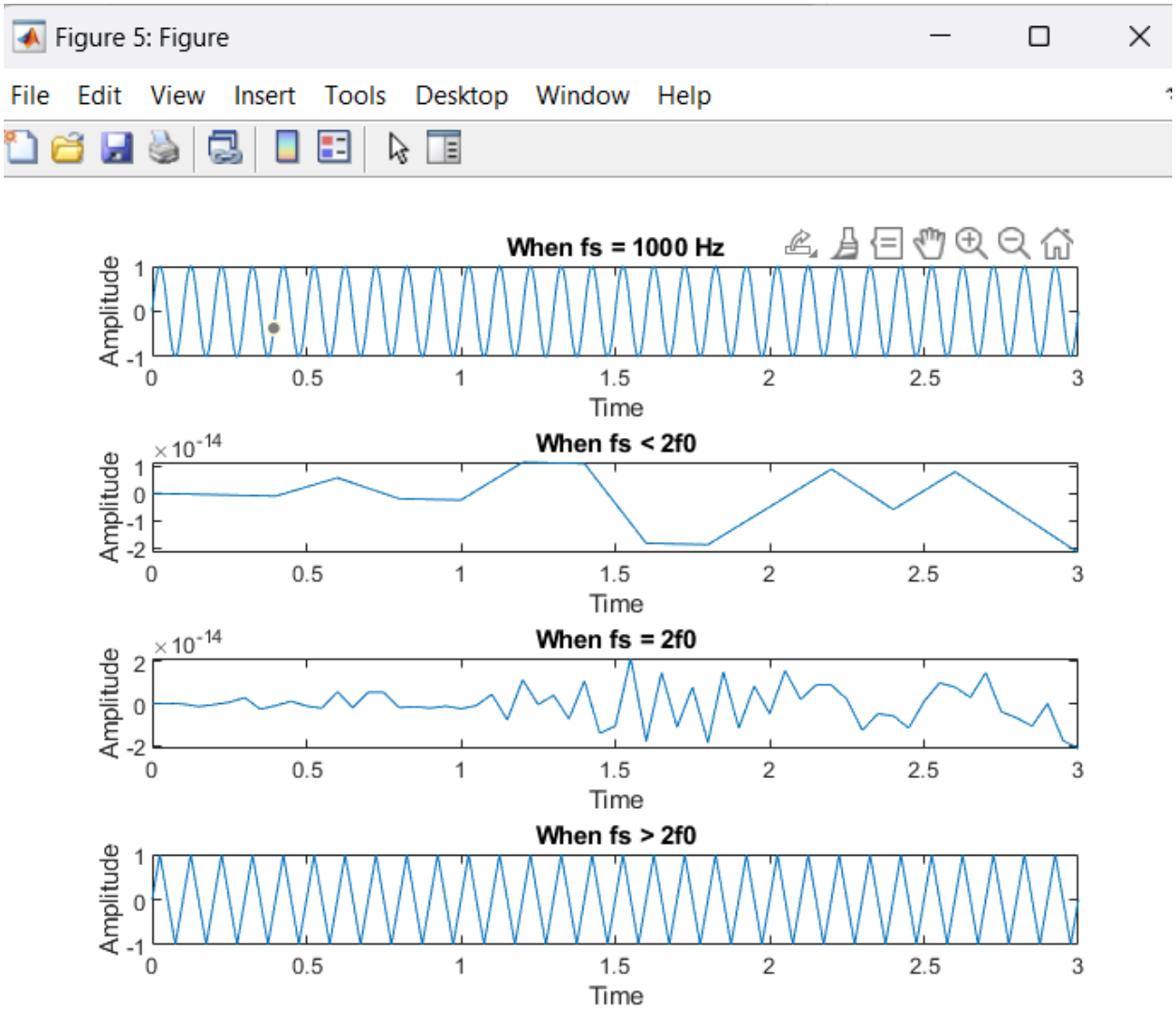
SNS lab 12

Name: basil khowaja (bk08432)

```
% Title based on the relationship between sampling rate and Nyquist frequency
if (fs(i) < 2*f0)
title("When fs < 2f0");
elseif(fs(i) == 2*f0)
title("When fs = 2f0");
elseif(fs(i) > 2*f0)
title("When fs > 2f0");
end
xlabel('Time');
ylabel('Amplitude');
end
```

SNS lab 12

Name: basil khowaja (bk08432)



Observation:

When fs is less than or equal to 2f0, the sampling signal doesn't show enough about the original signal. Aliasing happens because the sampling rate's frequency is too low to correctly represent the sine wave that was there in the first place. The signal that was rebuilt looks like it has a lower frequency than the original.
When fs> 2f0: The sampled signal properly captures the original signal without aliasing. The reconstructed signal fits the original sine wave because it follows the Nyquist theorem