

Demo 6.3 – Sampling and Reconstruction (Aliasing)

XMUT220 Signals and Systems

Overview

1. Practical audio application of aliasing.
2. Time domain perspective of aliasing in sampling of signals.
3. Frequency domain perspective of aliasing in sampling of signals.

Exercise 1 – Practical Audio Perspective

In MATLAB illustrate the effects of aliasing and folding by listening to an audio sweep from 100 Hz to 2,000 Hz over a 2-seconds duration.

In this exercise we observe the effects of aliasing and folding by listening to an audio sweep from 100 Hz to 2,000 Hz over a 2-second duration.

Type the following function in the m script file and name it as playsweep.m:

```
function [x,t] = playsweep(f1, f2, dur, fs);
% Sweep sinusoidal signal from f1 to f2
% for duration dur sampled at fs

t=0:1/fs:dur;

x = cos(2*pi*((fs-f1)/(2*dur)*t.*t+f1*t));

soundsc(x,fs);
end
```

In the MATLAB command prompt type the following experiments:

```
>> playsweep(100, 2000, 2, 8000);
>> playsweep(100, 2000, 2, 4000);
>> playsweep(100, 2000, 2, 2000);
>> playsweep(100, 2000, 2, 1000);
```

The significance of the results obtained are as follows:

a. If the sweep is reconstructed from samples taken at a rate of 8 kHz, it sounds like a rising tone. This is because we are still sampling at twice the highest frequency.

b. If we reduce the sampling rate to 4 kHz, it still sounds like a rising tone because we are still sampling at twice the highest frequency.

c. If we reduce the sampling rate to 2 kHz, we are sampling at the highest frequency. The tone rises then falls.

d. If we further reduce the sampling rate to 1 kHz, we hear the tone rise and fall and rise and fall again as we alias and fold the signal.

Note that the frequency of the reconstructed tone never exceeds $f_s/2$. After the first rise the succeeding falls and rises are due to folding and aliasing, respectively, as shown in the table below.

Signal frequency/sampling frequency	**Reconstructed signal**
$0 < f_0/f_s < ½$	Good reconstruction
$n+1/2 < f_0/f_s < n+1$, n=0, 1, 2, …	Folding
$n+1 < f_0/f_s < n+3/2$, n=0, 1, 2, …	Aliasing

Exercise 2 – Time-Domain Perspective of Sampling

Create in MATLAB time-domain plots of the sinusoidal signal sampled at the following frequencies (illustrate the effects of aliasing and folding on the plots if applicable):

a. Sinusoid frequency (f_0) = 10 Hz and sampling frequency (f_s) = 25 Hz.
b. Sinusoid frequency (f_0) = 52 Hz and sampling frequency (f_s) = 50 Hz.
c. Sinusoid frequency (f_0) = 48 Hz and sampling frequency (f_s) = 50 Hz.

If the sampling frequency is too slow ($f_s < 2f_0$), we may experience aliasing or folding. These effects may be understood in both the frequency and the time domain.

In the frequency domain, we use the discrete-time spectra to study the underlying causes of aliasing or folding.

We can also illustrate the aliasing in the time domain through the following examples that are using the following function coded in MATLAB.

```
% aliasfold.m
% demonstrates aliasing and folding

fo=input('frequency of sinusoid in Hz: ');
fs=input('sampling frequency in Hz: ');

% find alias or fold frequency
if fo>=fs
    faf=mod(fo,fs);
else
    faf=mod(fs,fo);
end

dt=1/(100*fo); % time vector for continuous-time
dts=1/fs; % time vector for discrete-time
```

```
if faf==0
    dur=5/fo;
else
    dur=2/abs(faf);
end

t=0:dt:dur-dt;
ts=0:dts:dur-dts;

x=2*cos(2*pi*fo*t-pi/2);
xs=2*cos(2*pi*fo*ts-pi/2);

plot(t,x);

hold on;

plot(ts,xs,'or:');
```

```
xtitle=sprintf('Sampling with fo = %3.0f Hz and fs =
%3.0f Hz',fo,fs);

title(xtitle);
xlabel('Time (sec)');
ylabel('Amplitude');
hold off;
```

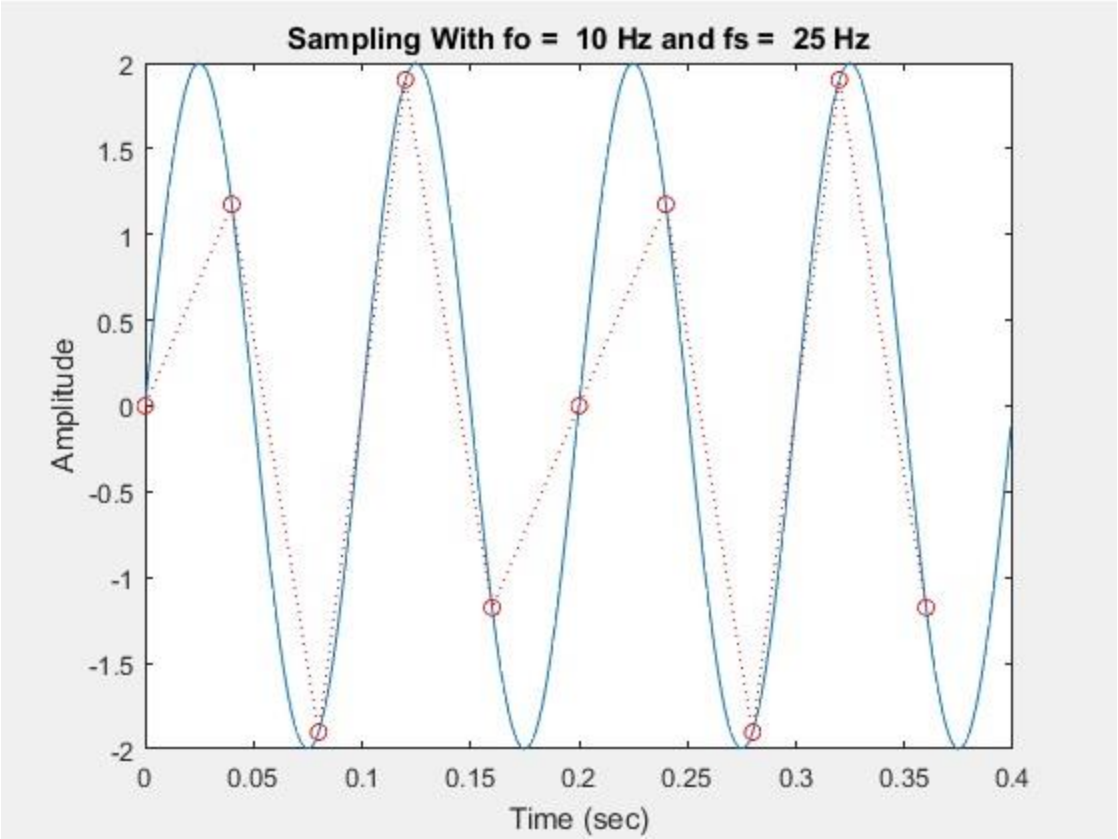
For the first example of aliasing and folding in the time domain, try sinusoid frequency = 10 Hz and sampling frequency = 25 Hz.

```
>> aliasfold
Frequency of sinusoid in Hz: 10
Sampling frequency in Hz: 25
```

The given figure shows a 10 Hz sinusoid sampled at 25 Hz.

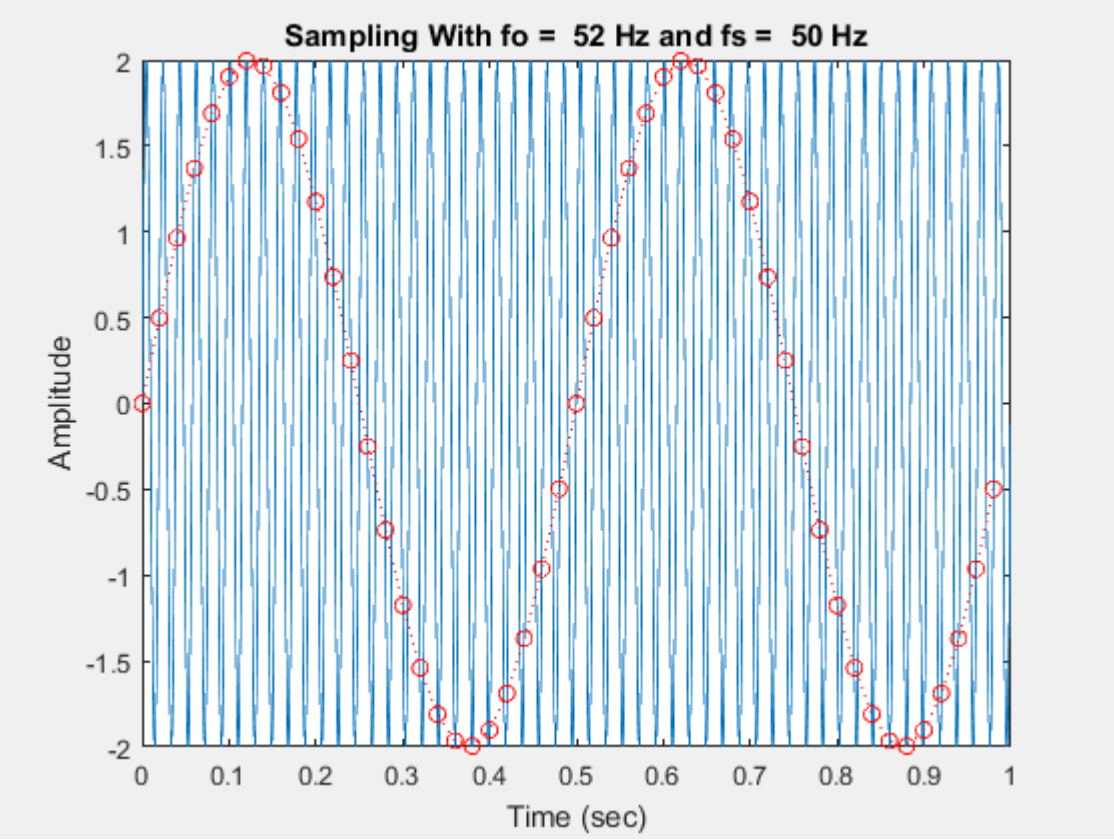
The line represents the sinusoid and the circles represent the samples.

We have enough information to reconstruct the original sinusoid.



For the second example of aliasing and folding in the time domain, try sinusoid frequency = 52 Hz and sampling frequency = 50 Hz.

```
>> aliasfold
Frequency of sinusoid in Hz: 52
Sampling frequency in Hz: 50
```



The figure above shows a 52 Hz sinusoid sampled at 50 Hz.

If the you connect the circles, you would probably say that they look like a 2 Hz sinusoid, and not the actual 52 Hz sinusoid.

This is an example of aliasing.

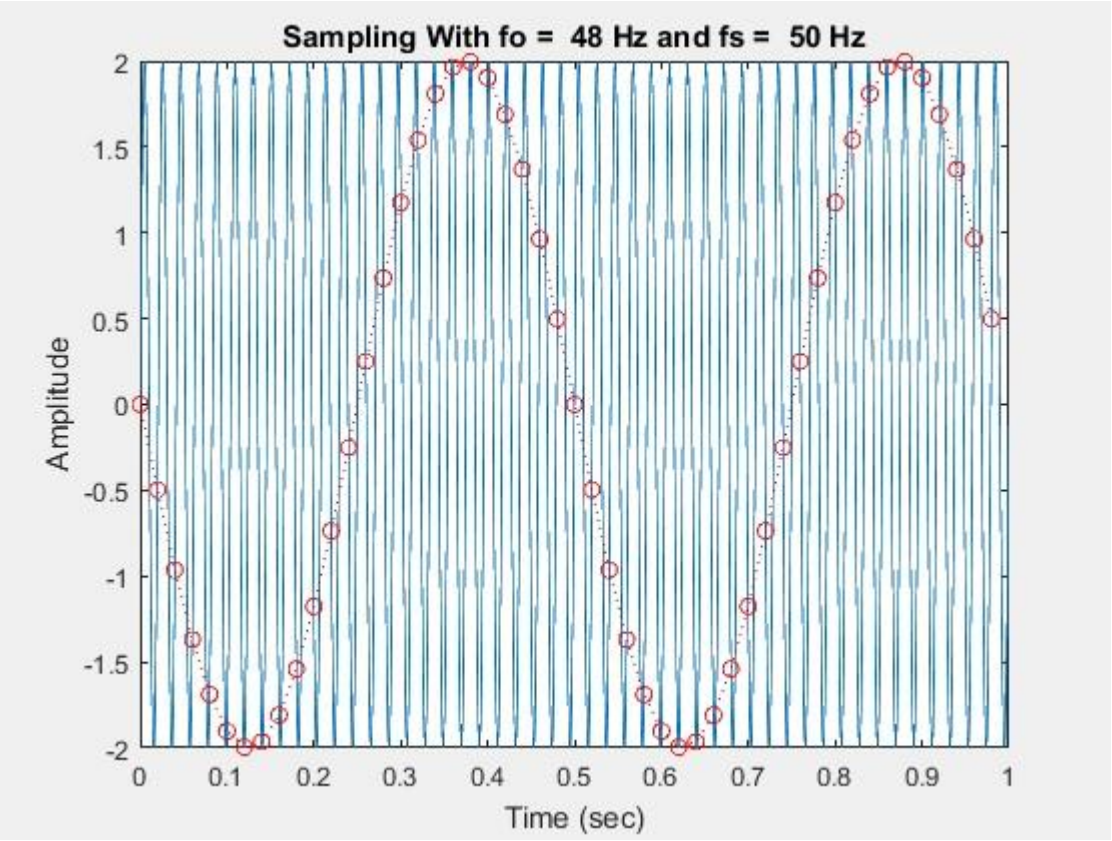
For the third example of aliasing and folding in the time domain, try sinusoid frequency = 48 Hz and sampling frequency = 50 Hz.

```
>> aliasfold
Frequency of sinusoid in Hz: 48
Sampling frequency in Hz: 50
```

The given figure shows a 48 Hz sinusoid sampled at 50 Hz.

This plot clearly illustrates folding and we observe a -2 Hz sinusoid.

This can also be viewed as a 2 Hz sinusoid with the phase reversed.



Exercise 3 – Frequency-Domain Perspective of Sampling

Create codes in MATLAB showing frequency spectrums of the following signal sampling specifications (highlight any frequency components in the spectrums):

a. Time period of the signal (T) = 0.1s and number of samples (N) = 3.
b. Time period of the signal (T) = 0.5s and number of samples (N) = 11.
c. Time period of the signal (T) = 0.5s and number of samples (N) = 32.
d. Time period of the signal (T) = 0.55s and number of samples (N) = 32.
e. Time domain plot of the frequency spectrum in (d). Note any peculiarities on the plot.

For a simple signal like in the previous Exercise 2, the coefficients are easy to find by inspection. In more complicated signal, we could use the FFT or DFT to compute the coefficients (both perform the same function, but the FFT is faster and is therefore more popular).

In the special case of signals that are sums of sinusoids, we could still find the coefficients by inspection because each sinusoid can be written as sum of complex exponentials.

To be able to work in the frequency domain, create the following function as a script in MATLAB and name it as fdomain.m

```
% fdomain.m
% computes the Fourier coefficients from vector x
% and the corresponding frequencies (two sided)

function [X,f]=fdomain(x,fs)

N=length(x);

if mod(N,2)==0;
  K=-N/2:N/2-1; % If N is even
else
  K=-(N-1)/2: (N-1)/2; % If N is odd
end
```

```
T=N/fs;
f=K/T;

X=fft(x)/N;

X=fftshift(X);

end
```

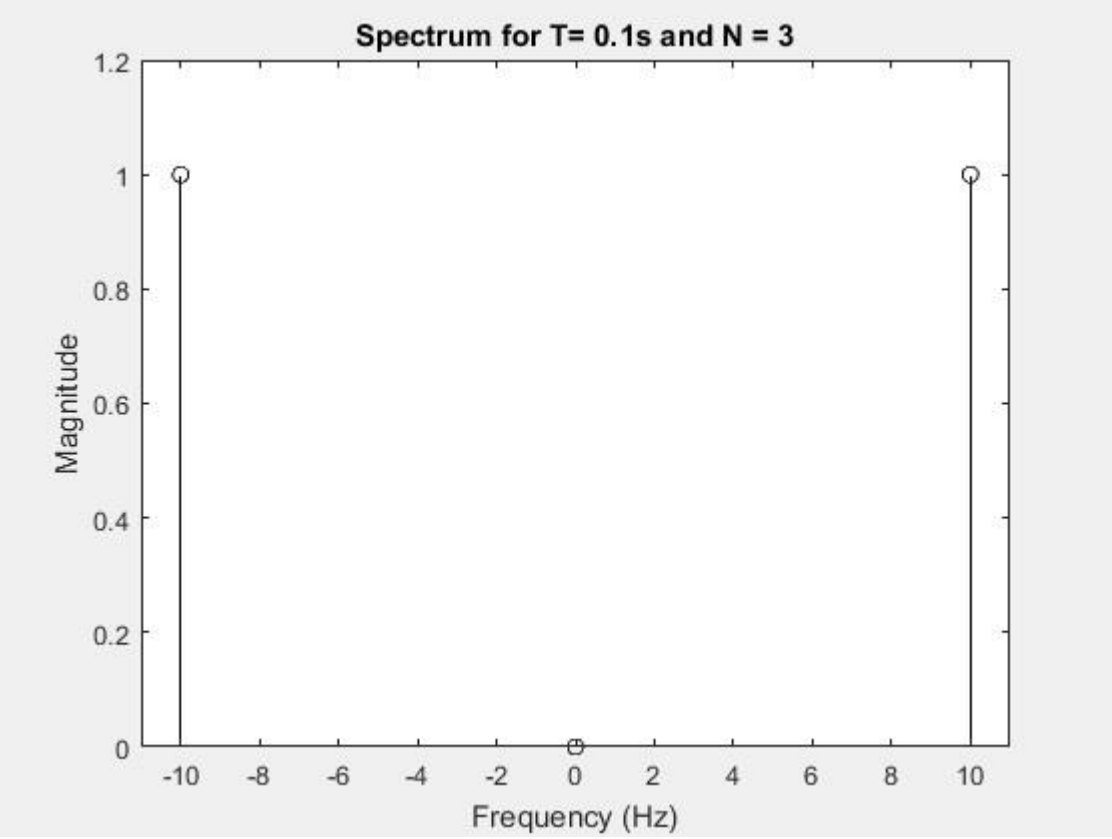
Suppose for this exercise, the sinusoid is at f_0 = 10 Hz (T_0 = 0.1 s). To correctly reconstruct the signal, we must have sampled at least f_s > 20 Hz.

Suppose that we take N = 3 samples over one period, that is sampled at f_s = $N \times f_0$ = 3 x 10 = 30 Hz, for a duration of T = periods/f_0.

```
% aliasfreq1.m
% Spectrum for T = 0.1 s and N = 3

clear all;
```

```
N=3;
periods=1;
fo=10;
T=periods/fo;
dt=T/N;
fs=1/dt;
t=dt*(0:N-1);

x=2*cos(2*pi*fo*t-pi/2);

[X,f]=fdomain(x,fs);

stem(f,abs(X),'k');

axis([-11,11,0,1.2]);
title('Spectrum for T = 0.1 s and N = 3');
xlabel('Frequency (Hz)');
ylabel('Magnitude');
```

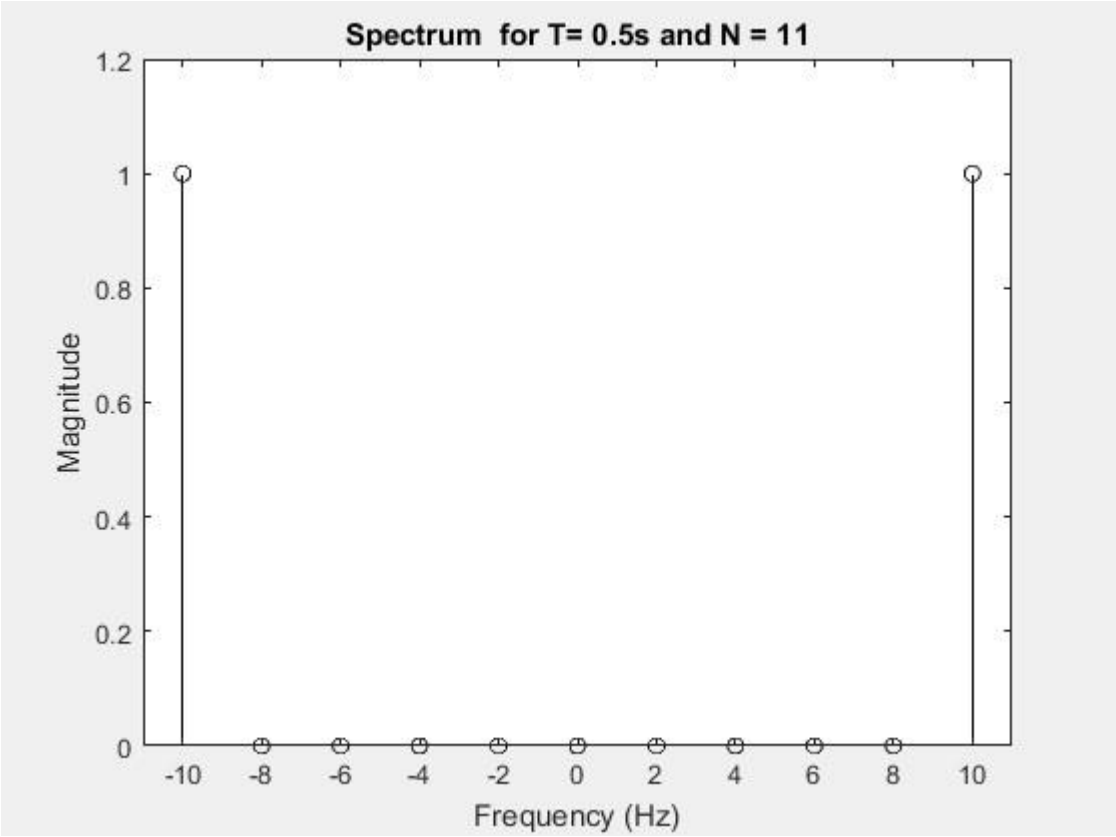
If you run the MATLAB code given above, it will give you the spectrum for T = 0.1 s and N = 3 as shown in the figure below.



The code below will give you the spectrum for T = 0.5 s and N = 11 as given in the figure below.

```
% aliasfreq2.m
% Spectrum for T = 0.5 s and N = 11

clear all;

N=11;
periods=5;
fo=10;
T=periods/fo;
dt=T/N;
fs=1/dt;
t=dt*(0:N-1);

x=2*cos(2*pi*fo*t-pi/2);

[X,f]=fdomain(x,fs);
```

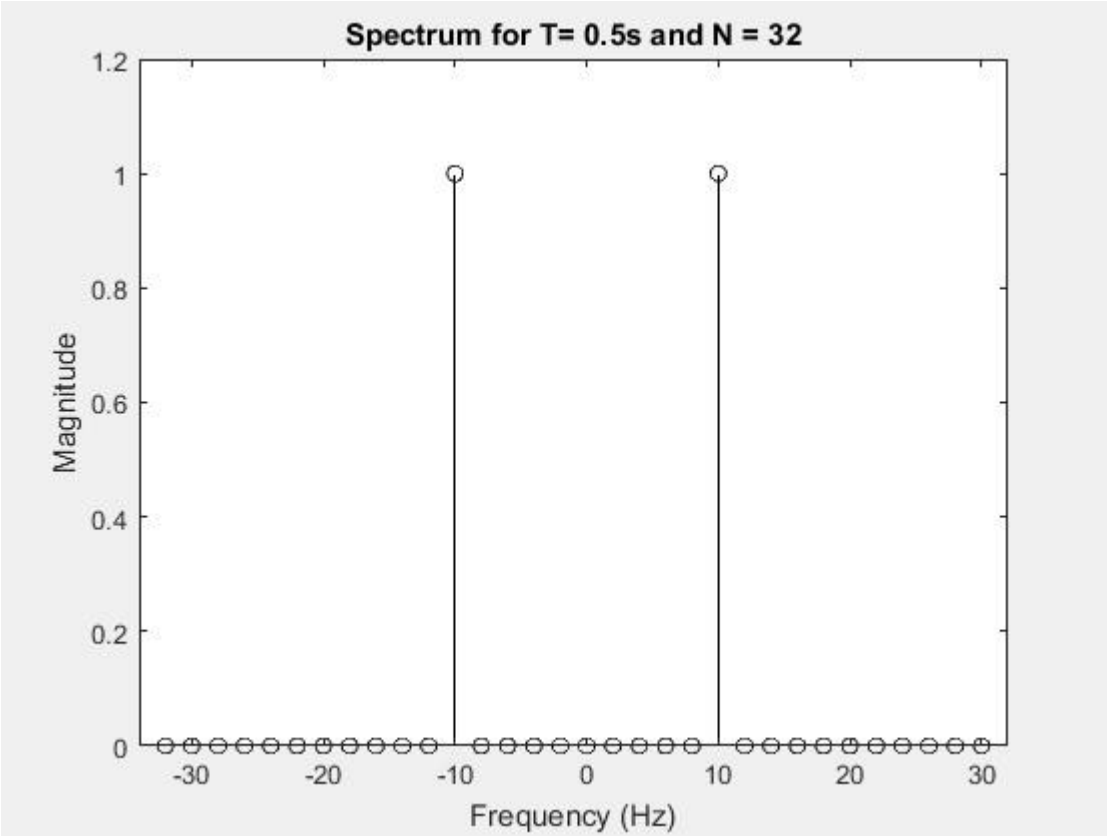
```
stem(f,abs(X),'k');

axis([-11,11,0,1.2]);
title('Spectrum for T = 0.5 s and N = 11');
xlabel('Frequency (Hz)');
ylabel('Magnitude');
```



The code below will give you the spectrum for T = 0.5 s and N = 32 as given in the figure below.

```
% aliasfreq3.m
% Spectrum for T = 0.5 s and N = 32

clear all;

N=32;
periods=5;
fo=10;
T=periods/fo;
dt=T/N;
fs=1/dt;
t=dt*(0:N-1);

x=2*cos(2*pi*fo*t-pi/2);

[X,f]=fdomain(x,fs);
```

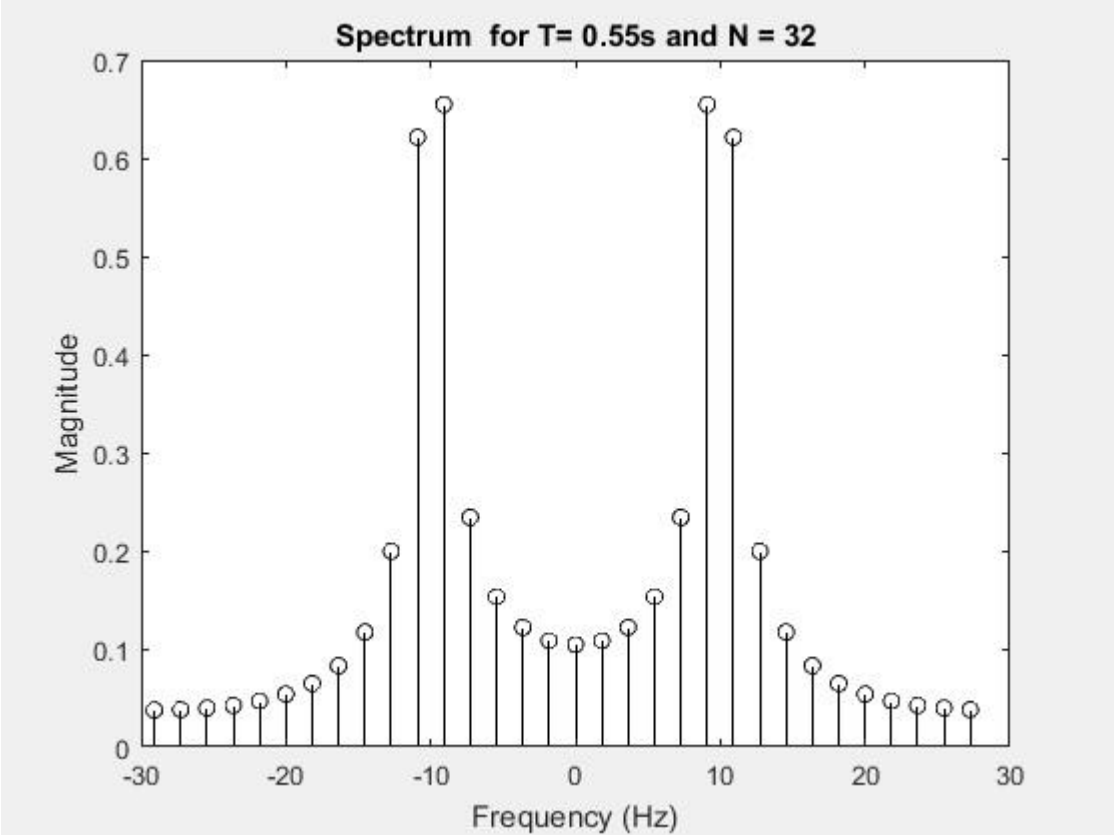
```
stem(f,abs(X),'k');

axis([-34,32,0,1.2]);
title('Spectrum for T = 0.5 s and N = 32');
xlabel('Frequency (Hz)');
ylabel('Magnitude');
```



The code below will give you the spectrum for T = 0.55 s and N = 32 as given in the figure below. Note the appearance of aliasing components on the sidebands.

```
% aliasfreq4.m
% Spectrum for T = 0.55 s and N = 32

clear all;

N=32;
periods=5.5;
fo=10;
T=periods/fo;
dt=T/N;
fs=1/dt;
t=dt*(0:N-1);

x=2*cos(2*pi*fo*t-pi/2);
[X,f]=fdomain(x,fs);
stem(f,abs(X),'k');
```

```
axis([-30,30,0,0.7]);
title('Spectrum for T = 0.55 s and N = 32');
xlabel('Frequency (Hz)');
ylabel('Magnitude');
```



The code below will give you the time-domain plot of the spectrum for T = 0.55 s and N = 32 as shown in the figure below.

```
% aliastime4.m
% Time-domain plot of spectrum for T = 0.5 s and N = 32
% (10 Hz sinusoid repeating after 5.5 periods)

clear all;

N=32;
periods=5.5;
fo=10;
T=periods/fo;
dt=T/N;
fs=1/dt;
t=dt*(0:N-1);
```

```
x=2*cos(2*pi*fo*t-pi/2);
tt=dt*(0:2*N-1);
xx=[x,x];

plot(tt,xx,'ko');
hold on;

tcont=linspace(0,0.55,500);

xcont=2*cos(2*pi*fo*tcont-pi/2);
xcont=[xcont,xcont];
tcont=linspace(0,1.1,1000);

plot(tcont,xcont,'k');

axis([0,1.1,-2.2,2.2]);
title('Time domain plot of spectrum for T = 0.55 s and N = 32');
xlabel('Time (sec)');
ylabel('Amplitude');

hold off;
```

The plot shows the sinusoid in the time domain. In this exercise when we are applying FFT we implicitly assume that the sampled data repeats after every interval T.

Therefore in this case the sinusoid repeats each 5.5 periods (0.55 s). By inspection on the figure given, we see that there is now a discontinuity where the signal repeats.

