Experiment 01

a) Sampling of sinusoidal waveform

Given, $x(t) = 10\cos(2\Pi \times 10^3 t) + 6\cos(2\Pi \times 2 \times 10^3 t) + 2\cos(2\Pi \times 4 \times 10^3 t)$

$F_s = 12kHz$

Code:

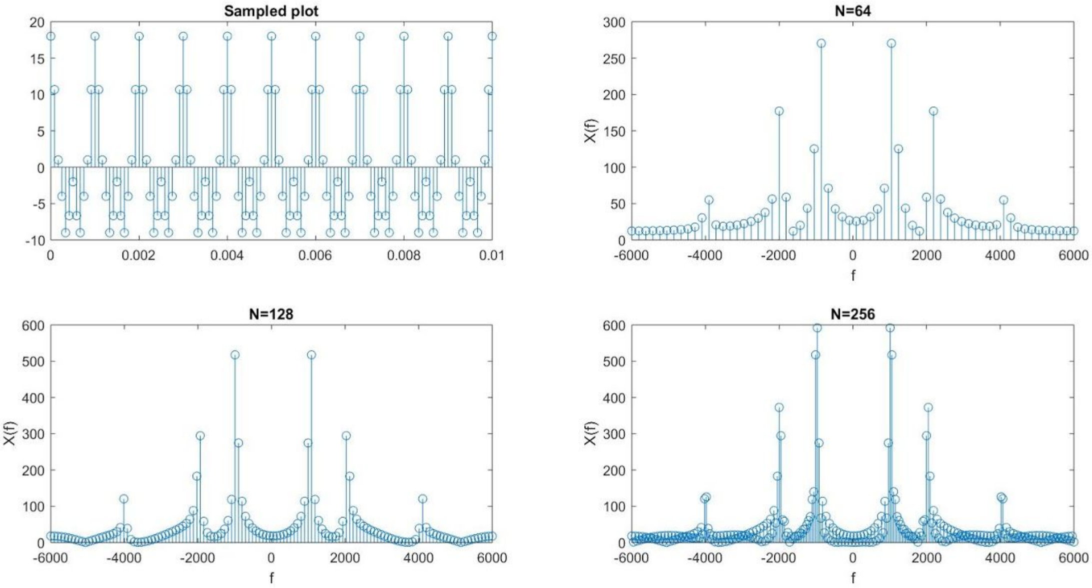
```
%Exp01 a)
Fs = 12000;
samp_dft(Fs); %get sampled plot and DTFT for N=64, 128 and 256
```

The code for function samp_dft:

```
function samp_dft(fsamp)

%%%%function takes sampling frequency as input and gives
sampled plot and
%%%%DTFT as output
Fs = fsamp;
t= 0:1/Fs:0.01;
x = 10*cos(2*pi*1000*t) + 6*cos(2*pi*1000*2*t) +
2*cos(2*pi*4000*t);
subplot(2,2,1);
stem(t,x);
title('Sampled plot');

%DFT:

N = 64; %Number of samples
f =-Fs/2:(Fs)/(N-1):Fs/2;
y1 = fft(x,N);
y = abs(y1);
y_f=fftshift(y);
subplot(2,2,2);
stem(f,y_f);
title('N=64');
xlabel('f');
```

```
ylabel('X(f)');

N = 128; %Number of samples
f =-Fs/2:Fs/(N-1):Fs/2;
y1 = fft(x,N);
y = abs(y1);
y_f=fftshift(y);
subplot(2,2,3);
stem(f,y_f);
title('N=128');
xlabel('f');
ylabel('X(f)');

N = 256; %Number of samples
f =-Fs/2:Fs/(N-1):Fs/2;
y1 = fft(x,N);
y = abs(y1);
y_f=fftshift(y);
subplot(2,2,4);
stem(f,y_f);
title('N=256');
xlabel('f');
ylabel('X(f)');

End
```

Observations



- The first plot is the sampled plot (at sampling frequency 12kHz)
- The other plots are Discrete Fourier Transforms with different number of samples (N)
- **fftshift must to be done after fft as it shifts the zero frequency to the centre of the spectrum.**
- Peaks are obtained at 1kHz, 2kHz and 4kHz as expected

b)Sampling at below Nyquist Rate and Effect of Aliasing

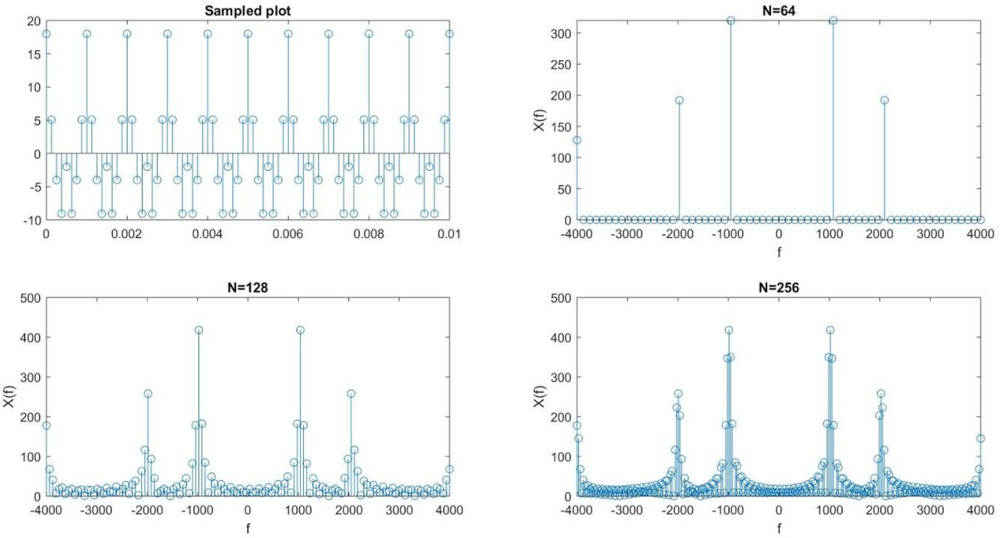
Code:

```
%Exp01b
%%The function samp_dft is same as above

Fs = 8000;
figure('Name','Fs = 8KHz');
samp_dft(Fs);


Fs = 5000;
figure('Name','Fs = 5KHz');
samp_dft(Fs);


Fs = 4000;
figure('Name','Fs = 4KHz');
samp_dft(Fs);
```

Observations:

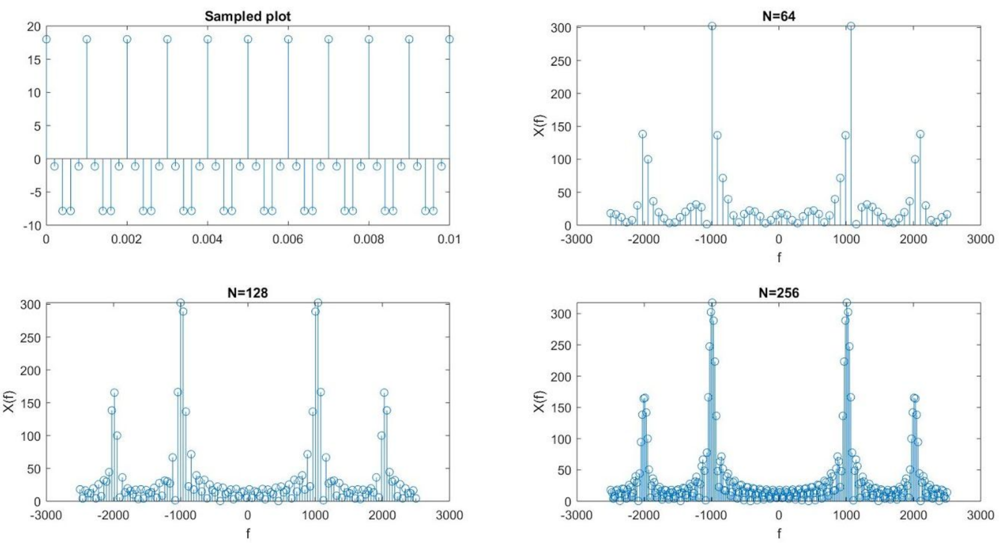
i) $F_s = 8kHz$

- All the signals with frequencies below 8kHz are obtained
- Here we obtain a peaks at frequencies 1kHz, 2kHz and 4kHz
- The sampling frequency used here is equal to the Nyquist rate of the original signal, hence we see all three frequency components.



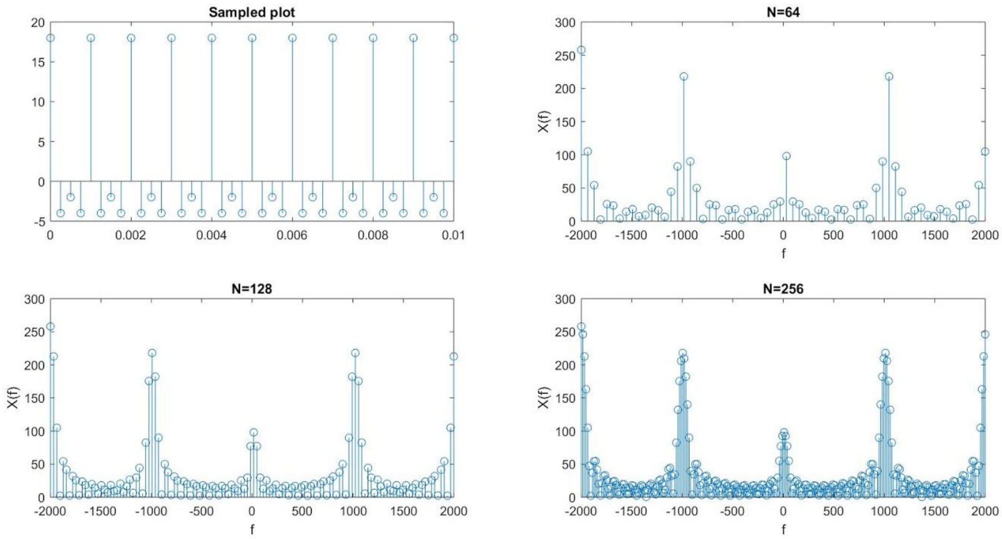
ii) $F_s = 5kHz$

- Only signals with frequencies 5kHz are observed in the DFT
- Peaks are observed at only 1000Hz and 2000Hz
- As sampling frequency is below Nyquist rate, the effect of aliasing is observed.



iii) $F_s = 4kHz$

- Only signals with frequency below 4kHz are observed in the DFT i.e. 1000Hz and 2000Hz
- Peaks obtained at 2kHz and 1kHz on both the sides
- As in the above case , the effect of aliasing is observed.



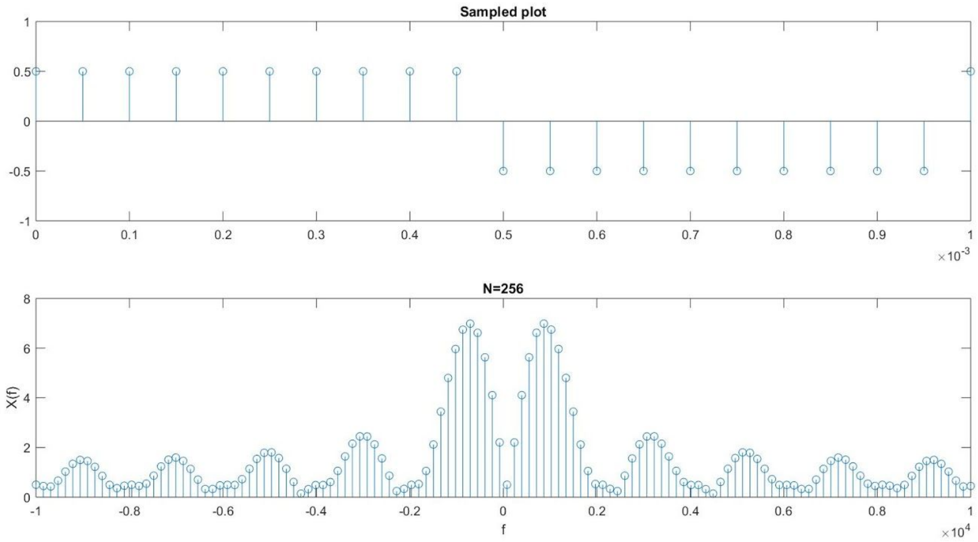
c) Sampling of square wave

Code:

```
%Exp01c

Fs = 20000; %sampling frequency 20KHz
f = 1000; %square wave frequency 1KHz
t= 0:1/Fs:0.001;

x = 0.5*square(2*pi*f*t,50);
subplot(2,1,1);
stem(t,x);
title('Sampled plot');
axis([0 0.001 -1 1]);
```

```
%DFT:

N = 128;
f =-Fs/2:Fs/(N-1):Fs/2;
y1 = fft(x,N);
y = abs(y1);
%plot(y)
y_f=fftshift(y);
subplot(2,1,2);
stem(f,y_f);
title('N=256');
xlabel('f');
ylabel('X(f)');
```

Observations:



- The peaks are obtained at odd intervals of frequency, 1kHz , 3kHz , 5kHz and so on with decreasing amplitude.
- Here also, fftshift was necessar to fring zero frequency at the centre of the spectrum otherwise due to improper axis the plot observed was different than the one above.

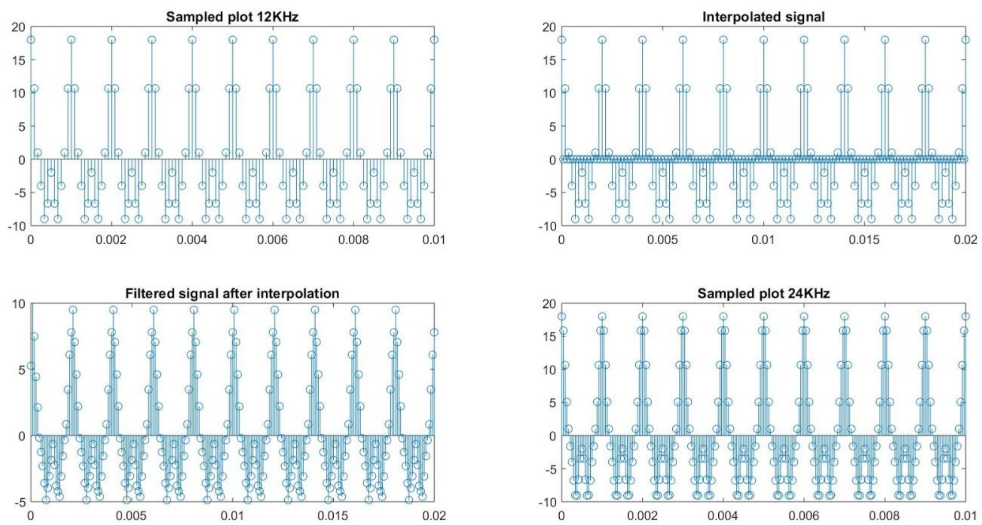
d) Interpolation or upsampling

Code:

```
%Exp01d

f = 6000;
Fs1 = 12000;

t= 0:1/Fs1:0.01;
x = 10*cos(2*pi*1000*t) + 6*cos(2*pi*1000*2*t) +
2*cos(2*pi*4000*t); %sampling
subplot(2,2,1);
stem(t,x);
%length(x)
title('Sampled plot 12KHz');

%x1 = reshape([x; zeros(size(x))],[],1);
%x2 = x1';
x2 = zeros(1,2*length(x));
for i = 1:length(x2)          %%%%%%upsampling
    if mod(i,2) == 0
        x2(1,i) = 0;
    else
        x2(1,i) = x((i+1)/2);
    end
end
t1 = zeros(1, length(x2));
t1(1,1) = 0;
for i = 2 : length(t1)
    t1(1,i) = t1(1,i-1) + 1/Fs1;
end
subplot(2,2,2);
stem(t1,x2);
title('Interpolated signal');
axis([0 0.02 -10 20]);
fs = 24000;
%wn = Fc/(Fs/2) ---> wn = 6000*2/24000
[b,a] = butter(2,0.5,'low');
y = filter(b,a,x2);
```

```
%t1= 0:1/Fs1:0.01;
subplot(2,2,3);
stem(t1,y);
title('Filtered signal after interpolation');
axis([0 0.02 -5 10]);


Fs2 = 24000;
t= 0:1/Fs2:0.01;
x = 10*cos(2*pi*1000*t) + 6*cos(2*pi*1000*2*t) +
2*cos(2*pi*4000*t); %sampling
subplot(2,2,4);
stem(t,x);
title('Sampled plot 24KHz');
```

Observations:



- Compared to the original signal the filtered signal has a lesser magnitude (difference by a scaling factor) and is shifted in time domain
- Initially, I used to reshape for upsampling (inserting zeros), but reshape changed the original signal also, hence the plot observed that way was incorrect
- Later, I used a for loop to upsample and got the correct results.
- Butterworth lowpass filter of 4th order was used.