DSP LAB TEST 3 report:

M Amshunath

22EE10042

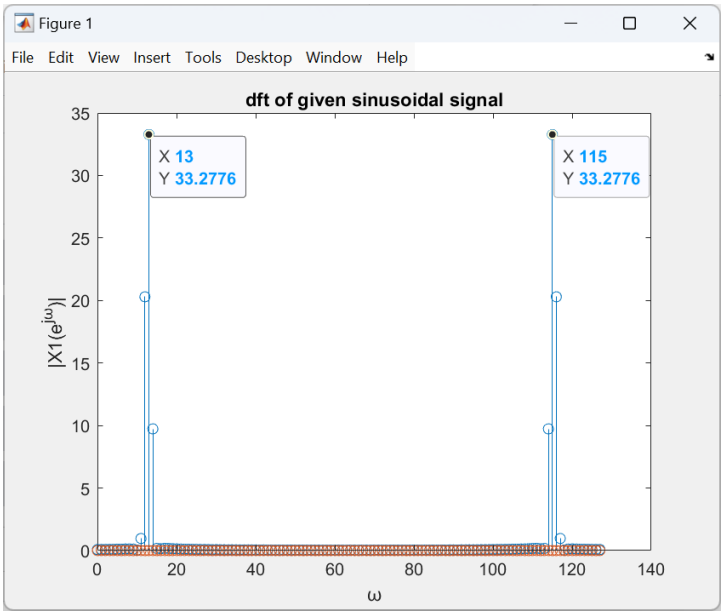
Experiment 7

Part 1: Implement and Test Hamming Window

Aim: to generate a hamming window multiply it with given signal and find its DFT

Matlab code:

```
N=128;
n=0:1:127;
x=cos(0.2*pi*n);%given signal
X=generateHamming(N);%hamming window of size 128
y=zeros(128); % after multiplication using hamming window
for i=1:128
    y(i)=x(i)*X(i);
end
Y=fft(y);
figure;
stem(n,abs(Y));
title("dft of given sinusoidal signal");
xlabel('ω');
ylabel('|X1(e^{jω})|');
```



Observations:

The values of omega where the peaks are found are +-1.2517rad

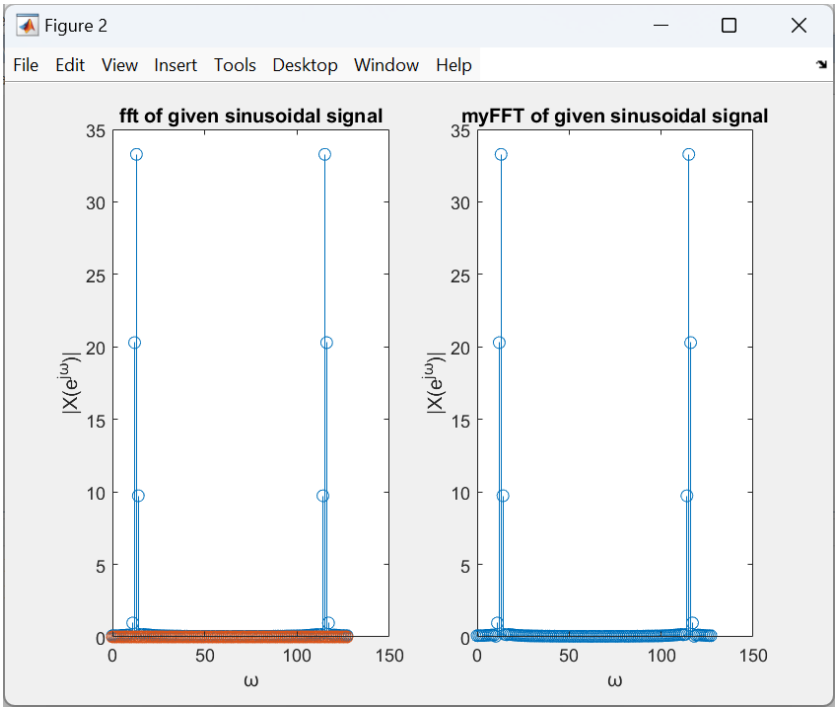
Part 2: FFT Algorithm Development

Aim: to write a matlab function myFFT and verify it

Matlab code:

```
Y2=myFFT(y);
figure;
subplot(1,2,1);
stem(n,abs(Y));
title("fft of given sinusoidal signal");
xlabel('ω');
```

```
ylabel('|X(e^{jω})|');
subplot(1,2,2);
stem(n,abs(Y2));
title("myFFT of given sinusoidal signal");
xlabel('ω');
ylabel('|X(e^{jω})|');
```



Observations

We can see that the results are the same for both the functions

Part 3: Zero Padding and Performance Analysis

Aim: to check the runtime of the different functions and also see the effect of the zero padding

Matlab code:

```
Xex=zeros(512);%the zero padded signal
for i=1:128
    Xex(i)=y(i);
end
tic;
Z1=fft(Xex);
fftime=toc;
tic;
Z2=myFFT(Xex);
myfftime=toc;
fprintf('Time taken by fft: %f seconds\n', fftime);
fprintf('Time taken by myFFT: %f seconds\n', myfftime);
n1=0:1:511;
figure;
stem(n1,abs(Z1));
```

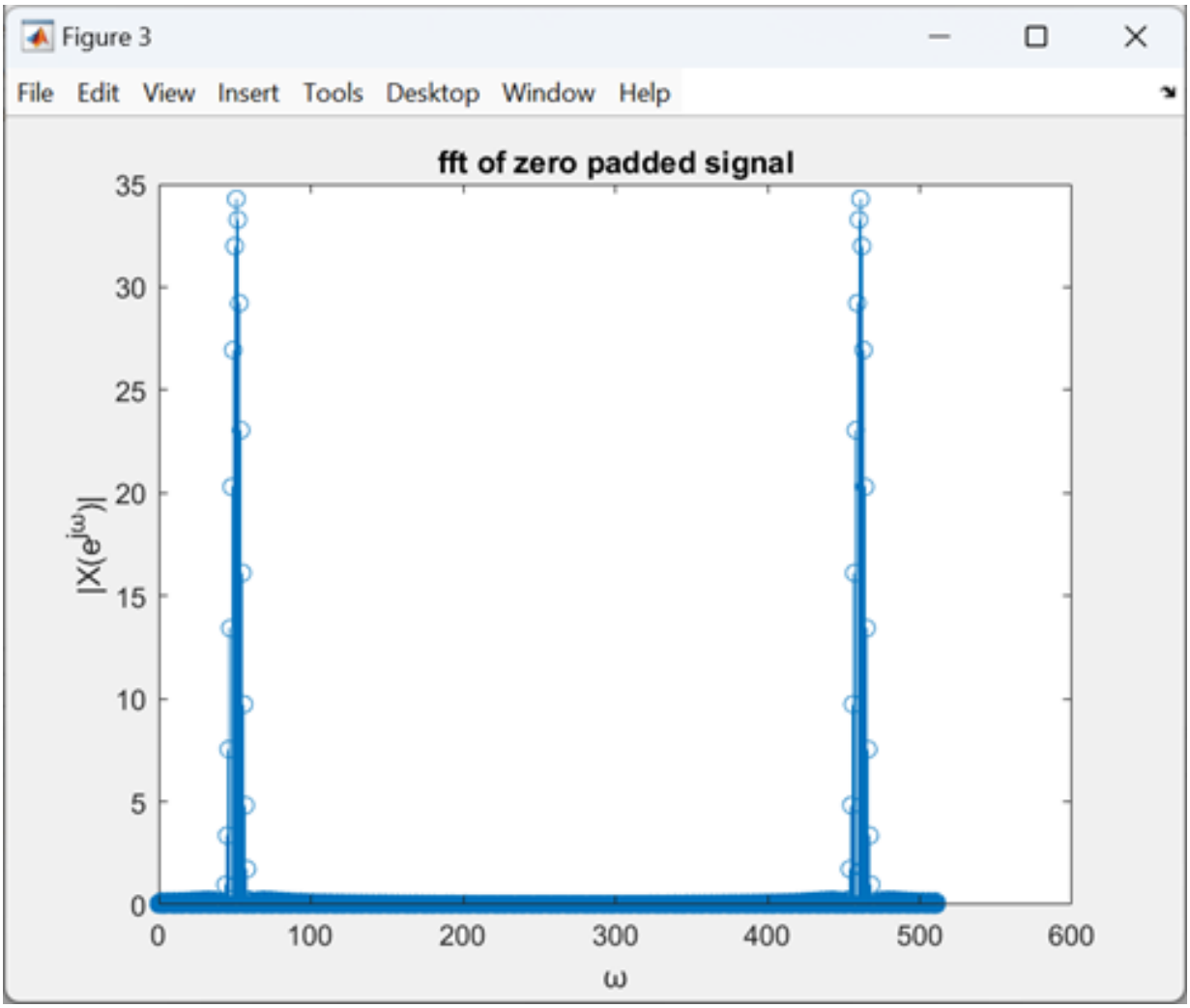
Observations:

The DFT of the zero padded signal looks more like the true DTFT. This is because the larger number of points in the DFT provides a finer resolution in the frequency domain, allowing for a more accurate representation of the signal's frequency characteristics

We can see that both the functions take almost the same time for implementation and can conclude that both the functions are written similarly

Command Window

```
Time taken by fft: 0.001027 seconds
Time taken by myFFT: 0.001072 seconds
>>
```



.