# Digital Signal Processing Lab

## Name: Deep C. Patel

## Roll No: 1401010

# Lab Report

# Lab Work:-

# Lab – 4

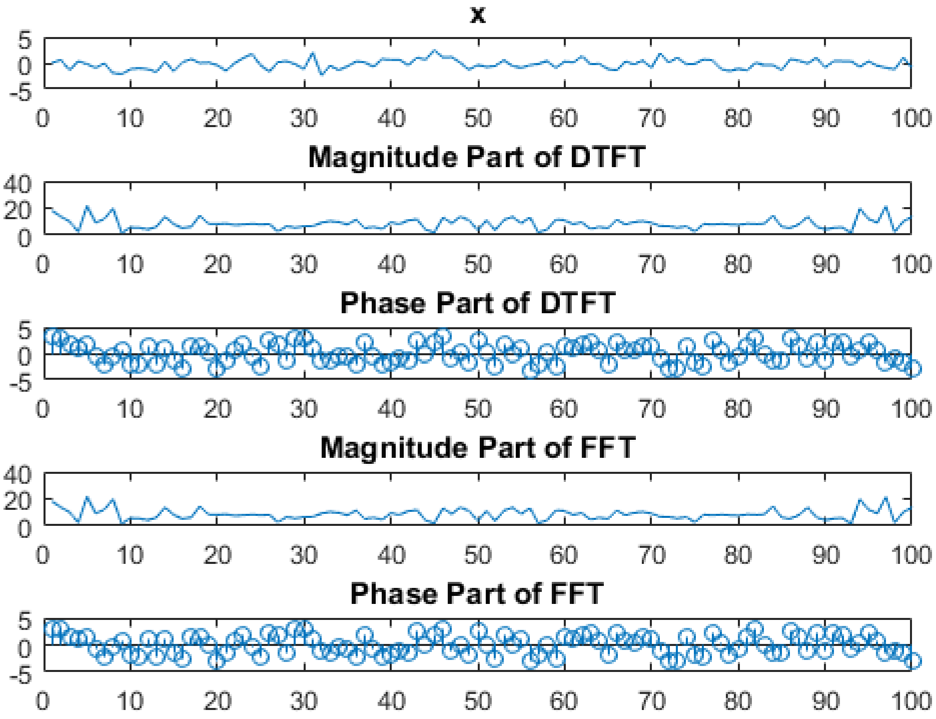
1).

```
% -> Given the sequence x(n) and sequence n write a program to compute DTFT and
% plot magnitude and phase spectrum
%
% -> Compute DFT of x(n) using built in function fft and verify that for a finite
% sequence x(n) DFT has an exact match with DTFT at F = k/N , k = 0,1,...N-1

clc;
clear;

Fs = 100;

f = 0:1:Fs-1;

x = randn(1,Fs);

y = DTFT(x,f);
y1 = fft(x);

disp(y(1:3));
disp(y1(1:3));

figure;

subplot(5,1,1);
plot(x);
title('x');

subplot(5,1,2);
plot(abs(y));
title('Magnitude Part of DTFT');
```

```
subplot(5,1,3);
stem(angle(y));
title('Phase Part of DTFT');

subplot(5,1,4);
plot(abs(y1));
title('Magnitude Part of FFT');

subplot(5,1,5);
stem(angle(y1));
title('Phase Part of FFT');
```

```
-18.2109 + 0.0000i -13.4492 + 2.5371i -0.2298 +10.1687i
```

```
 -18.2109 + 0.0000i -13.4492 + 2.5371i -0.2298 +10.1687i
```



## DTFT.M

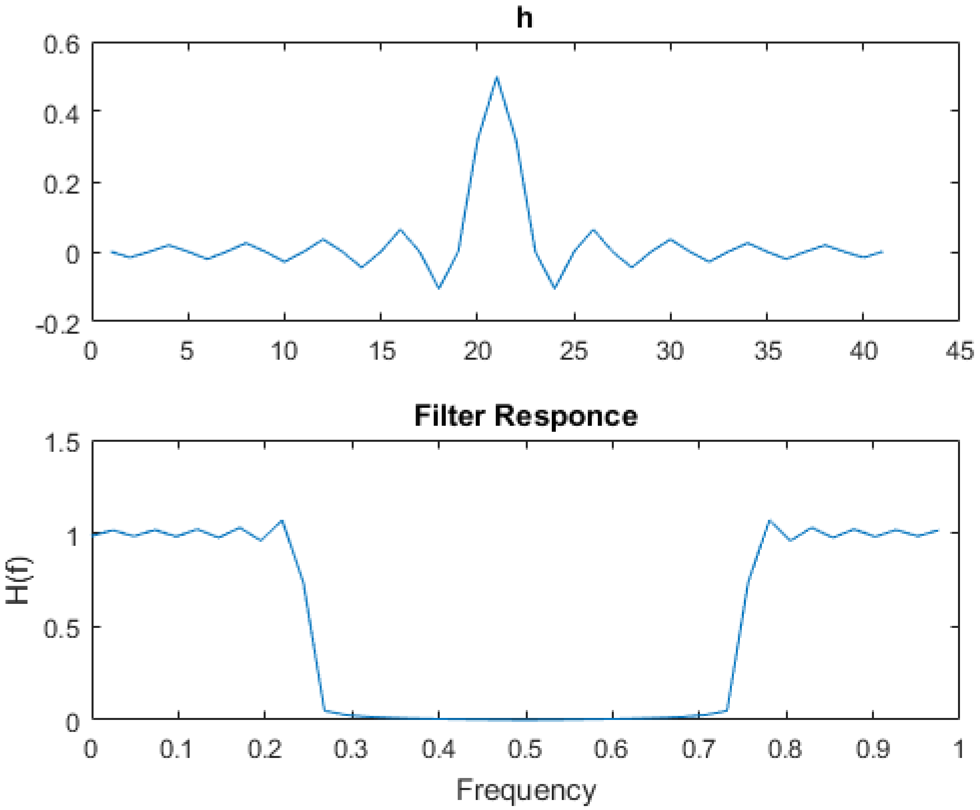
```
% DTFT

function y = DTFT(x,f)

    N = length(x);
    y = zeros(1,N);

    for k = 1:1:length(f)
        for m = 1:1:N;

            y(1,k) = y(1,k) + (x(m)*exp(-1i*2*pi*(m-1)*(f(k)/N)));

        end
    end

end
```

## 2).

```
% -> Write a program to calculate filter coefficients for low pass filter.
%
% -> Use these coefficients in the DTFT program and plot the frequency response
%
% -> Verify that you indeed get a low pass filter response

clc;
clear;

Fc = 0.25;

Fs = 20;
n = 1:1:Fs;

h_0 = 2*Fc;
h = 2*Fc*sin(2*pi*Fc.*n)./(2*pi*Fc.*n);

h = [fliplr(h) h_0 h];

f = 0:1:(length(h)-1);

y = DTFT(h,f);

disp(y(1:10));

figure;
```

```
subplot(2,1,1);
plot(h);
title('h');

subplot(2,1,2);
plot(f/length(h),abs(y));
title('Filter Responce');
xlabel('Frequency');
ylabel('H(f)');
```

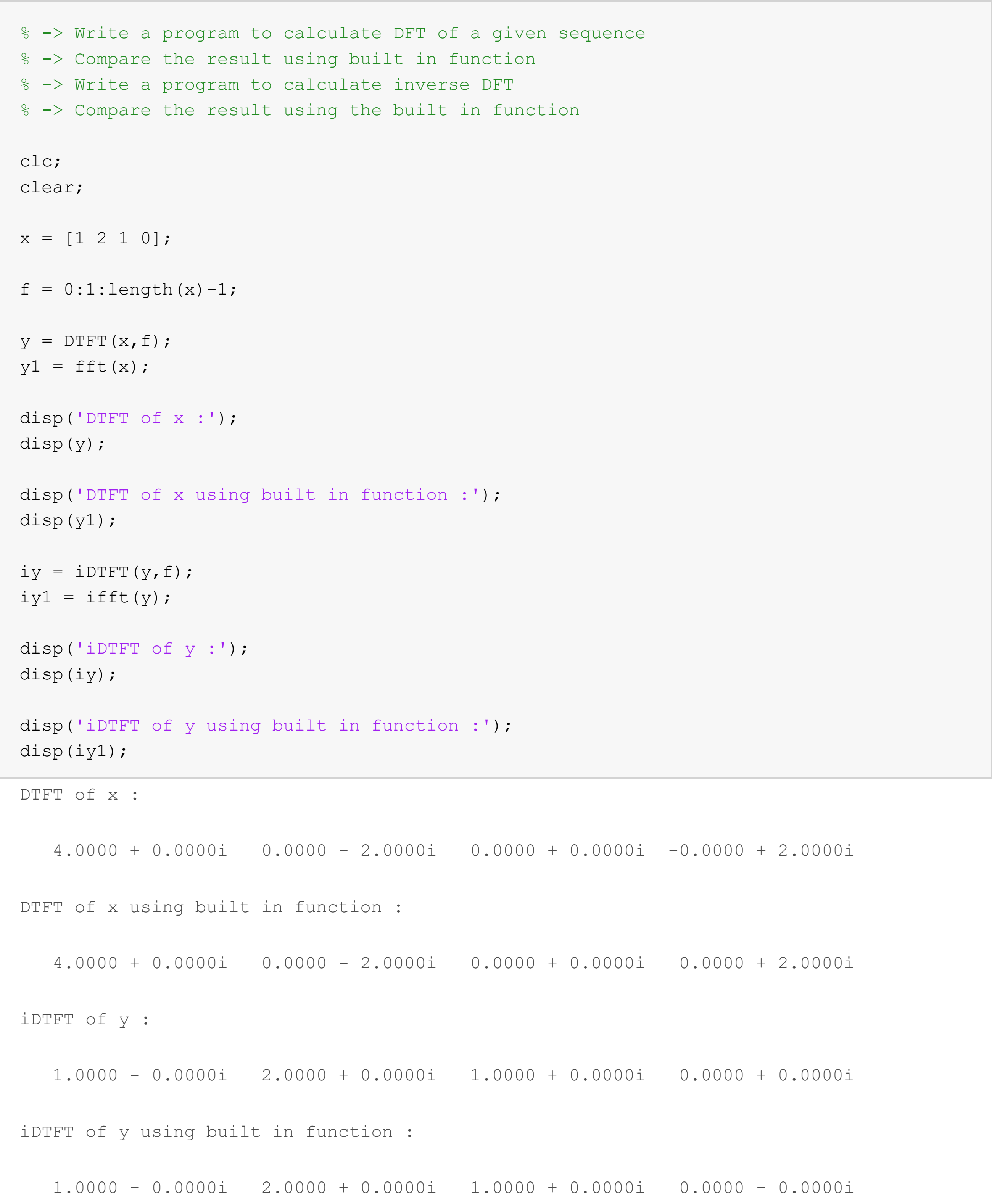
```
  Columns 1 through 4


  0.9841 + 0.0000i -1.0130 - 0.0778i   0.9720 + 0.1501i  -0.9906 - 0.2318i


  Columns 5 through 8


  0.9356 + 0.2961i -0.9467 - 0.3816i   0.8746 + 0.4331i -0.8850 - 0.5262i


  Columns 9 through 10


  0.7847 + 0.5520i -0.8251 - 0.6805i
```



## DTFT.M

```
% DTFT

function y = DTFT(x,f)

    N = length(x);
    y = zeros(1,N);

    for k = 1:1:length(f)
        for m = 1:1:N;

            y(1,k) = y(1,k) + (x(m)*exp(-1i*2*pi*(m-1)*(f(k)/N)));

        end
    end

end
```

3).



```
% -> Write a program to calculate DFT of a given sequence
% -> Compare the result using built in function
% -> Write a program to calculate inverse DFT
% -> Compare the result using the built in function

clc;
clear;

x = [1 2 1 0];

f = 0:1:length(x)-1;

y = DTFT(x,f);
y1 = fft(x);

disp('DTFT of x :');
disp(y);

disp('DTFT of x using built in function :');
disp(y1);

iy = iDTFT(y,f);
iy1 = ifft(y);

disp('iDTFT of y :');
disp(iy);

disp('iDTFT of y using built in function :');
disp(iy1);
```

```
DTFT of x :

   4.0000 + 0.0000i   0.0000 - 2.0000i   0.0000 + 0.0000i  -0.0000 + 2.0000i

DTFT of x using built in function :

   4.0000 + 0.0000i   0.0000 - 2.0000i   0.0000 + 0.0000i   0.0000 + 2.0000i

iDTFT of y :

   1.0000 - 0.0000i   2.0000 + 0.0000i   1.0000 + 0.0000i   0.0000 + 0.0000i

iDTFT of y using built in function :

   1.0000 - 0.0000i   2.0000 + 0.0000i   1.0000 + 0.0000i   0.0000 - 0.0000i
```

## iDTFT.M

```
% Inverse DTFT

function y = iDTFT(x,f)

    N = length(x);
    y = zeros(1,N);

    for k = 1:1:length(f)
        for m = 1:1:N;

            y(1,k)  = y(1,k)  + (x(m)*exp(1i*2*pi*(m-1)*(f(k)/N)));

        end
    end

    y = y/N;

end
```

## DTFT.M

```
% DTFT

function y = DTFT(x,f)

    N = length(x);
    y = zeros(1,N);

    for k = 1:1:length(f)
        for m = 1:1:N;

            y(1,k)  = y(1,k)  + (x(m)*exp(-1i*2*pi*(m-1)*(f(k)/N)));

        end
    end

end
```

**x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-**