Digital Signal Processing HW7 MATLAB Part

Name: Jingyang Zhang
Net ID: jz2807

1.16 **DFT and circular convolution.** Verify the circular convolution property of the DFT in Matlab. Write two Matlab functions to compute the circular convolution of two sequences of equal length. One function should use the DFT (`fft` in Matlab), the other function should compute the circular convolution directly not using the DFT. Verify that both Matlab functions give the same results.

Hand in a hard copy of both functions, and an example verifying they give the same results (you might use the `diary` command).

Solution:

.m file(s): main function: jyz_1_36.m
DFT Computation function: jyz_1_36_1.m
Directly compute function: jyz_1_36_2.m

Code:

DFT Computation function:

```
function[z1] = jyz_1_16_1(x,y)
X = fft(x);
Y = fft(y);
Z1 = X.*Y;
z1 = ifft(Z1);
```

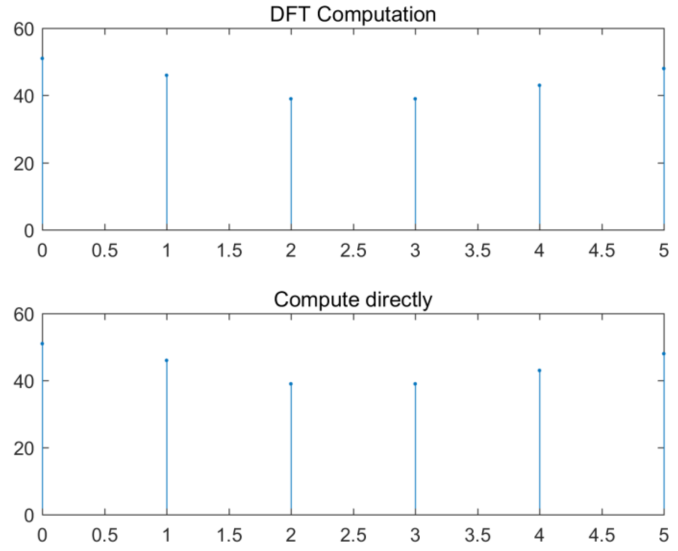
Directly compute function:

```
function[z2] = jyz_1_16_1(x,y,len)
i = 1;
j = 1;
circonvMatx = zeros(len);
for i = 1:1:len
    for j = 1:1:len
        for k = 1:1:len
            if (j - i == k-1)||(j+len-i == k-1)
                circonvMatx(i,j) = x(k); %circle convolution Matrix
            end
        end
    end
end
circonvMatx = circonvMatx';
z2 = (circonvMatx * y')';
```

Main function:

```
close all
clear

len = randi([3, 6]);
x = randi([1, 5], 1, len);
y = randi([1, 5], 1, len);

z1 = jyz_1_16_1(x,y);

z2 = jyz_1_16_2(x,y,len);

subplot(2,1,1)
stem([0:len-1],z1,'.')
title('DFT Computation');
subplot(2,1,2)
stem([0:len-1],z2,'.')
title('Compute directly');

if (z2 == z1)
	disp('Two results are the same')
else
	disp('Two results are different')
end
```

Result(plots):




```
>> jyz_1_16
Two results are the same
```

1.17 **DFT and linear convolution.** Write a Matlab function that uses the DFT (`fft`) to compute the linear convolution of two sequences that are not necessarily of the same length. (Use zero-padding.) Verify that it works correctly by comparing the results of your function with the Matlab command `conv`.

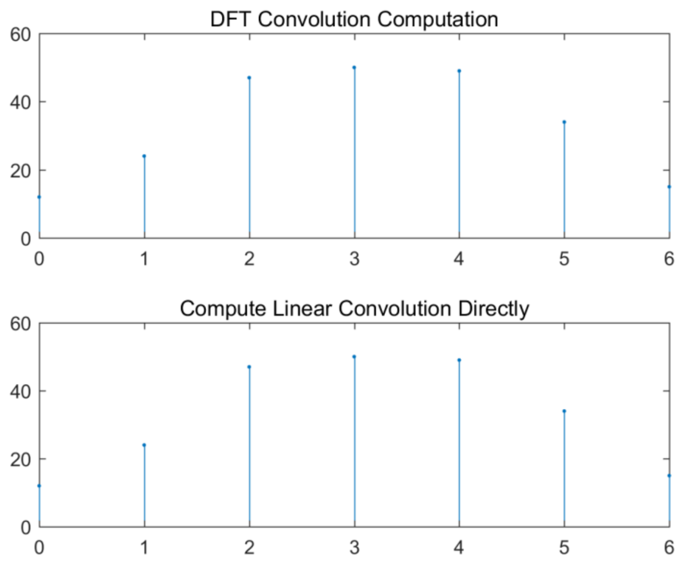
Solution:

.m file(s): jyz_1_17.m

Code:

```
close all
clear

len1 = randi([3, 6]);
len2 = randi([1, 2]);
x = randi([1, 5], 1, len1);
y = randi([1, 5], 1, len1+len2);
totallength = length(x)+length(y)-1;
X = fft(x, totallength);
Y = fft(y, totallength);
Z = X.*Y;
z1 = ifft(Z);
z2 = conv(x,y);
subplot(2,1,1)
stem([0:totallength-1],z1,'.')
title('DFT Convolution Computation');
subplot(2,1,2)
stem([0:totallength-1],z2,'.')
title('Compute Linear Convolution Directly');
```

Result(plots):



Comment: It is east to find that two results are the same.

1.61 An analog signal is sampled at 8192 Hz and 600 samples are collected. These 600 samples are available on the course webpage as the file `signal1.txt`. Plot the signal versus time in seconds; and using the DFT, plot the spectrum of the signal versus *physical frequency* in hertz with the DC component in the center of the plot. If your computer has sound capability, use the `soundsc` command to listen to the signal.

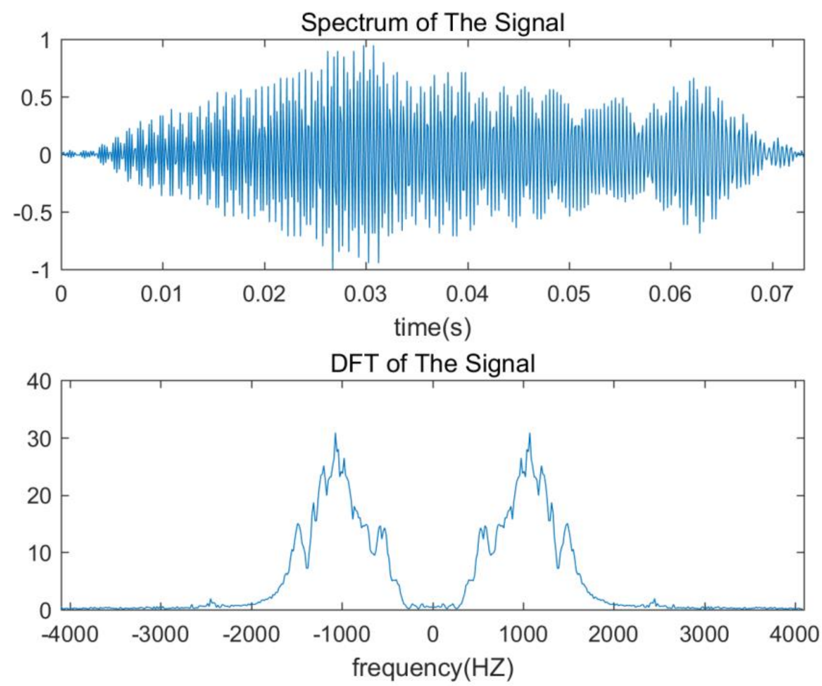
Solution:

.m file(s): jyz_1_61.m

Code:

```
close all
clear

load signal1.txt
x = signal1;
fs = 8192;
xf = fft(x(1:512));
% spectrogram(x)
subplot(2,1,1)
plot(0:1/8192:599/8192,x)
xlim([0,599/8192])
title('Spectrum of The Signal')
xlabel('time(s)')
subplot(2,1,2)
plot(-4095:8192/512:4096,abs(xf))
xlim([-4096,4096])
title('DFT of The Signal')
xlabel('frequency(HZ)')
soundsc(x,fs)
```

Result:



The sound is a bird twitter.