



Norwegian University of Science and Technology  
Department of Electronics and Telecommunications

## TTT4120 Digital Signal Processing

### Solutions for Problem Set 6

#### **Problem 1:**

**(a).**

The signal  $x(n)$  is given by

$$x[n] = \begin{cases} 0.9^n & n = 0, \dots, N_x - 1, \\ 0 & \text{otherwise.} \end{cases}$$

The DTFT of  $x[n]$  can be written as

$$X(\omega) = \sum_{n=0}^{N_x-1} 0.9^n e^{-j\omega n} = \sum_{n=0}^{N_x-1} (0.9e^{-j\omega})^n = \frac{1 - (0.9e^{-j\omega})^{N_x}}{1 - 0.9e^{-j\omega}},$$

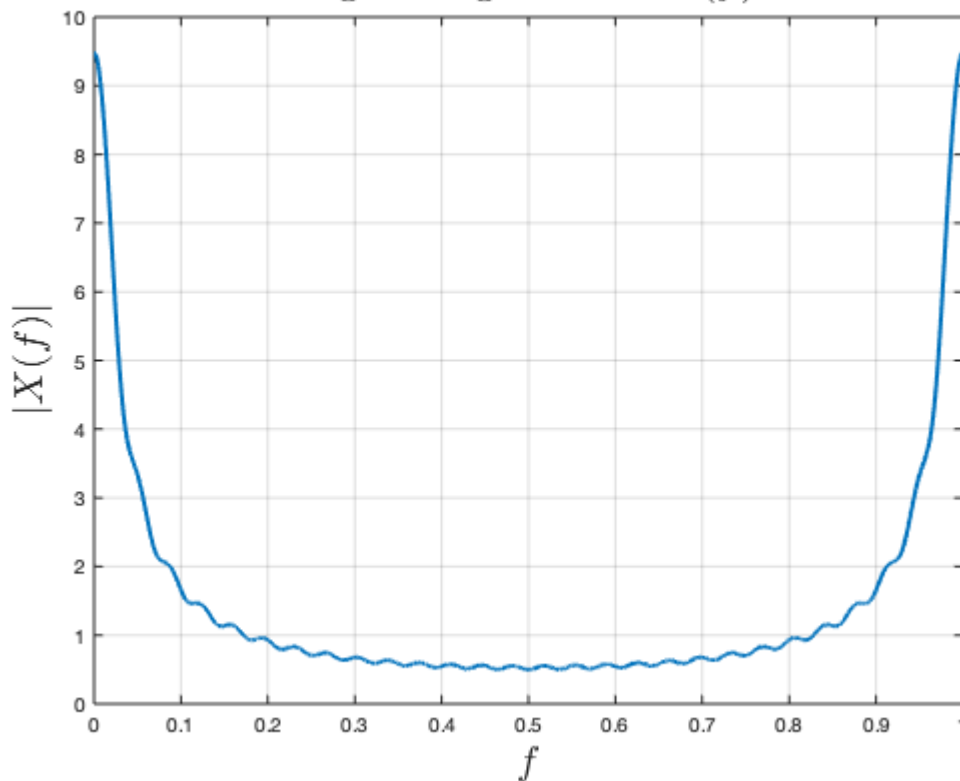
where we have used the expression for the sum of a geometric series.

Since  $\omega = 2\pi f$ , we obtain  $X(f)$  as

$$X(f) = \frac{1 - (0.9e^{-j2\pi f})^{N_x}}{1 - 0.9e^{-j2\pi f}}.$$

This function can be plotted in MATLAB by the following commands. The resulting plot is shown in Fig.1.

```
Nx = 28;
f = 0:1/10000:1;
Xf = (1 - (0.9*exp(-j*2*pi*f)).^Nx) ./ (1 - 0.9*exp(-j*2*pi*f));
plot(f,abs(Xf));
xlabel('f');
ylabel('|X(f)|')
```

Fig.1: Magnitude of  $X(f)$ **(b).**

The following lines will generate the signal  $x[n]$  and compute its DFTs.

```
n = 0:(Nx-1);
x = 0.9.^n;
N1 = Nx/4; N2 = Nx/2; N3 = Nx; N4 = 2*Nx;
Xf1 = fft(x,N1); Xf2 = fft(x,N2);
Xf3 = fft(x,N3); Xf4 = fft(x,N4);
```

**(c).**

The relationship between  $f$  and  $k$  is given by

$$f = \frac{k}{N}, \quad k = 0, \dots, N-1$$

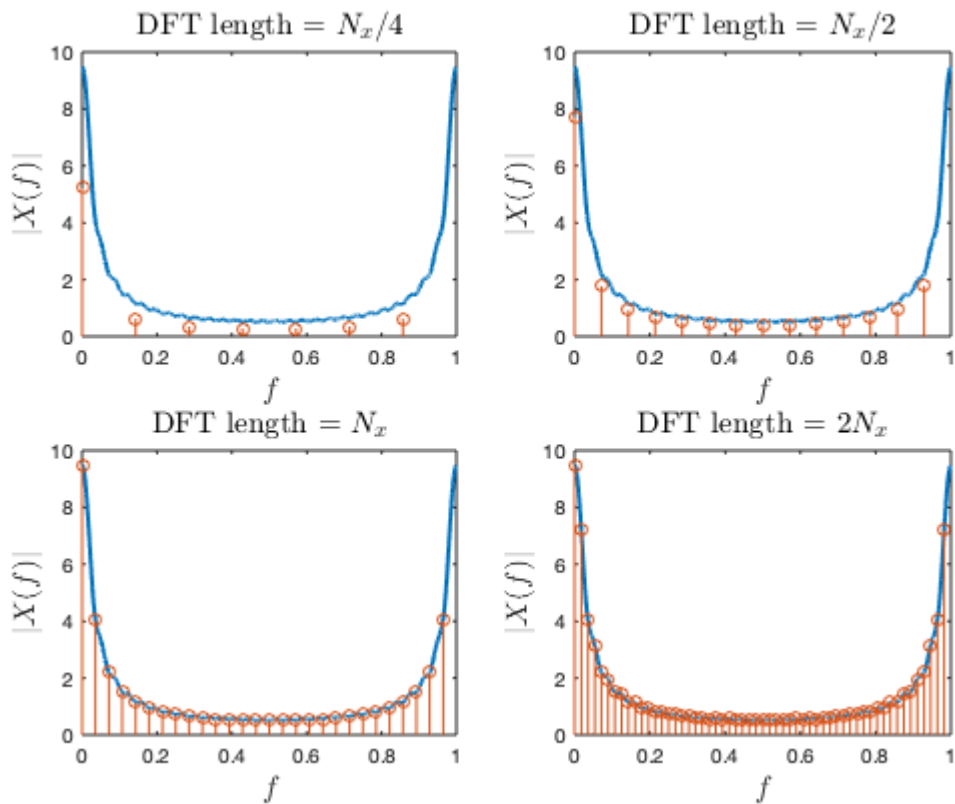
where  $N$  is the length of the DFT.

**(d).**

MATLAB code for plotting the magnitude of one DFT together with the DTFT is as follows.

Magnitudes of DTFT and DFT for different DFT lengths are shown in following figure as Fig.2.

```
plot(f,abs(Xf)); hold on;
stem(f1,abs(Xf1))
xlabel('f');
ylabel('|X(f)|');
title('DFT length = N_x/4')
```



**Fig2. : Magnitudes of DTFT and DFT for different DFT lengths**

We see that if the DFT-length is less than  $N_x$ , the DFT-values will not represent exact the samples of the DTFT.

**(e).**

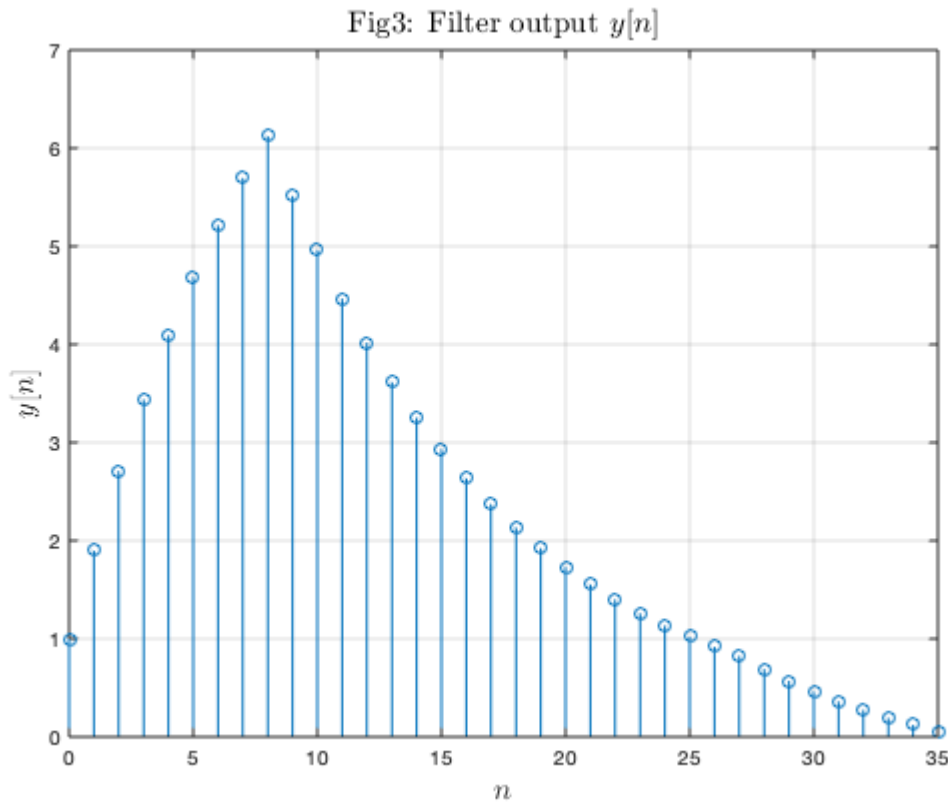
It follows from the symmetry and periodicity properties of the DTFT and DFT that if their values are known for  $f \in [0, 0.5]$ , they would be known for all values of  $f$ .

## Problem 2:

**(a).**

The following Matlab commands compute and plot the output signal  $y[n]$ . The resulting plot is shown in Fig.3.

```
Nx = 28;
n = 0:(Nx-1);
x = 0.9.^n;
Nh = 9;
h = ones(1,Nh);
y = conv(x,h);
n=0:length(y)-1;
stem(n,y);
xlabel('n')
ylabel('y[n]')
```



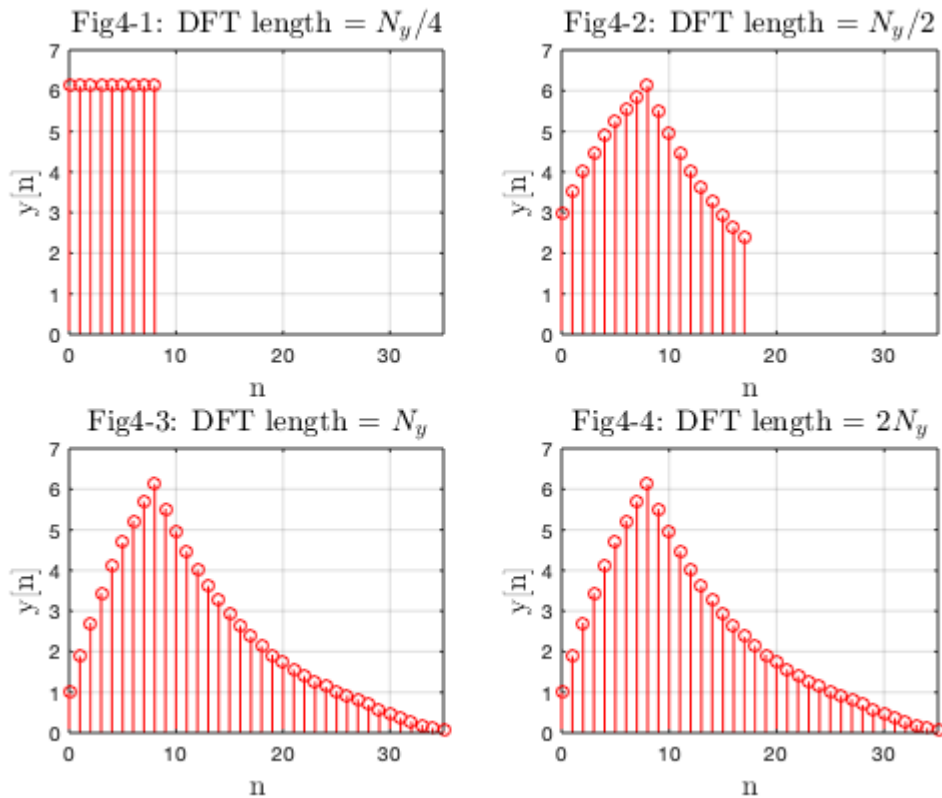
The length of the output signal  $y[n]$  is given by  $N_y = N_x + N_h - 1 = 36$ . This can be verified from the plot.

**(b).**

The output signal  $y[n]$  can be found in Matlab by the algorithm given below.

```
N = (Nx+Nh-1)/4;
X = fft(x,N);
H = fft(h,N);
Y = X.*H; y = ifft(Y,N);
n = 0:N-1;
stem(n,y);
xlabel('n');
ylabel('y[n]')
```

The resulting signal  $y[n]$  for the three different choices of DFT/IDFT lengths are shown in Fig.4.



**Fig.4: Output signal computed using DFT/IDFT of different lengths**

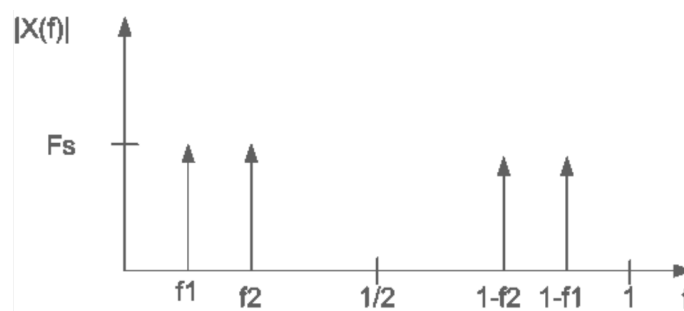
We know from theory that the length of the DFT must be at least equal to  $N_y$  to avoid time domain aliasing. This is verified by the plots.

Note also that the output signal computed using DFT length  $N = N_y/2 = 18$  is given as one period of a periodic extension of  $y[n]$  with period  $N_y/2 = 18$ . This is easily verified if the signal is plotted in the same plot as the original  $y[n]$  (use different colours).

### Problem 3:

(a).

Fig.5 shows the magnitude spectrum of the sampled signal on the interval  $f \in [0, 0.5]$ .



**Fig5. Magnitude spectrum of the sampled signal**

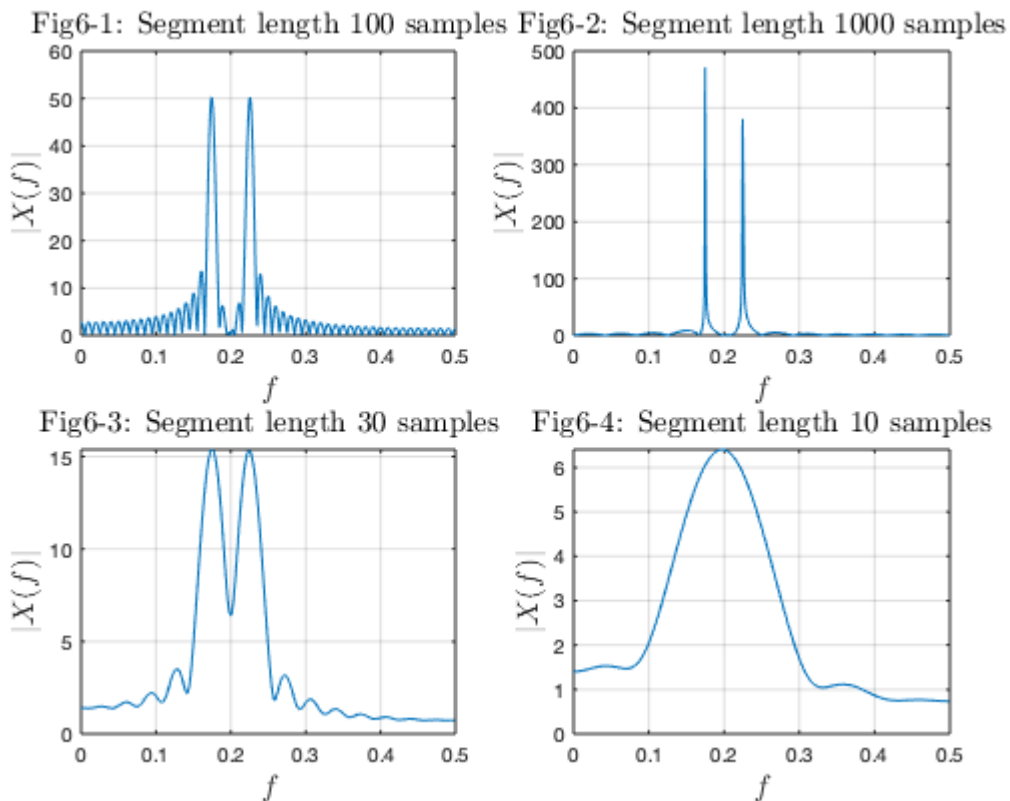
(b).

MATLAB code for computing a spectral estimate from a finite signal segment is shown below.

```

f1 = 7/40; f2 = 9/40; N = 1024;
%generates 100 samples of the sampled signal
t = 0:99;
x = sin(2*pi*f1*t) + sin(2*pi*f2*t);
X = fft(x,N);
f = 0:1/N:1-1/N;
plot(f(1:length(f)/2),abs(X(1:length(f)/2)))
xlabel('f');
ylabel('|X(f)|')
title('Segment length 100samples');

```



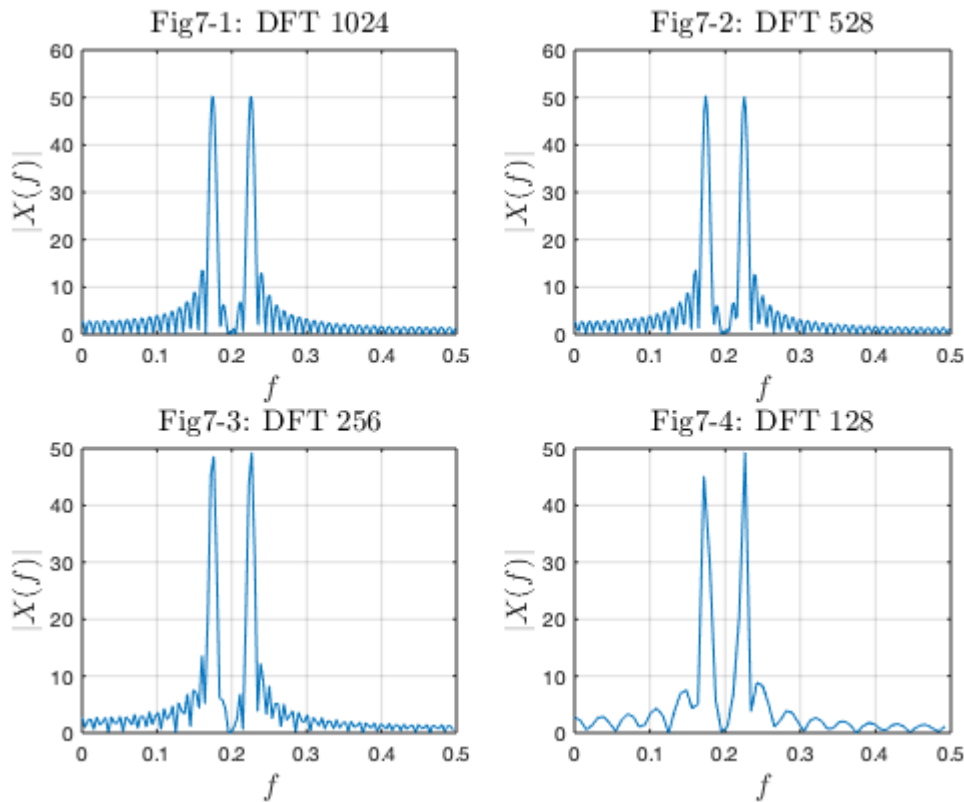
**Fig.6: Estimates of the magnitude spectrum for different segment lengths**

Spectral magnitudes estimated using the different segment lengths are shown in Fig.6.

We observe that shorter segments result in less accurate spectral estimates. The effect of spectral leakage is evident, as the frequency content is no longer limited to only two spectral components. The spectral resolution becomes poorer as the segment length decreases, and for the segment length of 10 samples, it is no longer possible to distinguish the two original frequency components.

(c).

Spectral magnitudes estimated using segment lengths 100 and different DFT lengths are shown in Fig.7



**Figure 7: Estimates of the magnitude spectrum for different DFT lengths**

Denoting the true spectrum as  $H(\omega)$ , the length  $L$  of the signal segment will determine the accuracy of the spectral estimate  $\hat{H}(\omega)$ , while the transform length  $N$  will determine spacing between the samples of  $\hat{H}(\omega)$ . (The DFT must be longer than the segment length to avoid aliasing).

### Problem 4:

(a).

Fast Fourier Transform (FFT) is the name of some efficient computational algorithms for computing the DFT.

(b).

Radix-2 FFT algorithm is an efficient algorithm for DFT computation which can be applied when  $N = 2^\nu$ , where  $\nu$  is an integer (zero padding might have to be used to extend the number of data points to  $N$ ). The idea of the algorithm is to divide the sequence  $x[n]$  into two sequences  $f_1[n]$  and  $f_2[n]$  of length  $\frac{N}{2}$ , compute their DFTs and combine them to obtain  $X(k)$ . It can be shown that the number of operations needed to compute  $X(k)$  in this way is approximately two times smaller than in the case of direct computation of  $X(k)$  for large  $N$ . This is due to the periodicity and symmetry properties of  $W_N^{kn}$ . This fact is used iteratively for computation of DFTs of  $f_1[n]$  and  $f_2[n]$ , so that the direct DFT computation has to be applied only to subsequences of length 2.

(c).

From the formula for DFT we can observe that for each value of  $k$ , direct computation of  $X(k)$  needs  $N$  complex multiplications and  $N - 1$  complex additions. To compute all  $N$  values of the DFT then requires

$N^2$  complex multiplications and  $N^2 - N$  complex additions.

**(d).**

$$\begin{aligned}
 X(k) &= \sum_{n=0}^{N-1} x[n] W_N^{kn}, k = 0, 1, \dots, N-1 \\
 &= \sum_{n \text{ even}} x[n] W_N^{kn} + \sum_{n \text{ odd}} x[n] W_N^{kn} \\
 &= \sum_{m=0}^{(N/2)-1} x[2m] W_N^{2mk} + \sum_{m=0}^{(N/2)-1} x[2m+1] W_N^{k(2m+1)}
 \end{aligned}$$

Substitute  $W_N^2 = W_{N/2}$ . We can then write:

$$\begin{aligned}
 X(k) &= \sum_{m=0}^{(N/2)-1} f_1[m] W_{N/2}^{mk} + W_N^k \sum_{m=0}^{(N/2)-1} f_2[m] W_{N/2}^{km} \\
 &= F_1(k) + W_N^k F_2(k), k = 0, 1, \dots, N-1
 \end{aligned}$$

$F_1(k)$  and  $F_2(k)$  are both periodic with period  $N/2$ . That is, we have  $F_1(k + N/2) = F_1(k)$  and  $F_2(k + N/2) = F_2(k)$ . We also have  $W_N^{k+N/2} = -W_N^k$ . We can then write:

$$\begin{aligned}
 X(k) &= F_1(k) + W_N^k F_2(k), k = 0, 1, \dots, N/2 - 1 \\
 X(k + N/2) &= F_1(k) - W_N^k F_2(k), k = 0, 1, \dots, N/2 - 1
 \end{aligned}$$

**(e).**

The direct computations of  $F_1(k)$  and  $F_2(k)$  both require  $(N/2)^2$  complex multiplications. They are also  $N/2$  complex multiplications in computing  $W_N^k F_2(k)$ . The number of complex multiplications required to compute  $X(k)$  is  $2(N/2)^2 + N/2 = N^2/2 + N/2$ . Comparing with the results in (c) we can see that for large  $N$  the reduction in number of multiplications is about a factor of 2.