GÜRAY ÖZGÜR
2167054



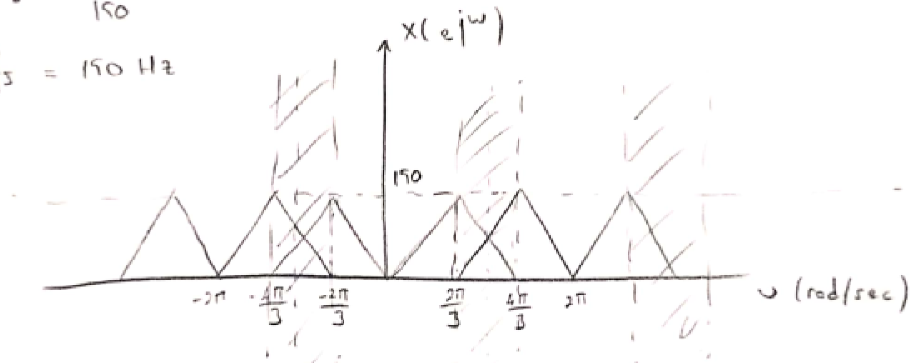
# EE433 REAL-TIME APPLICATIONS OF DIGITAL SIGNAL PROCESSING
## EXPERIMENT 1 - PRELIMINARY WORK

1. The solutions of the first question are handwritten and can be found below.

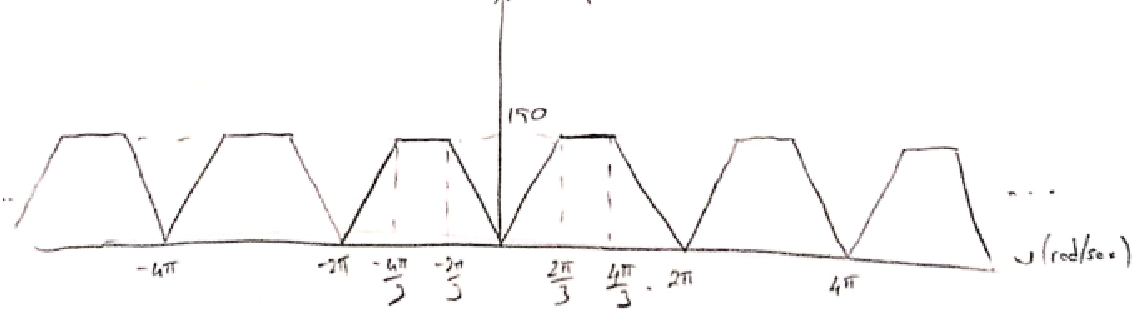
a) If we do not use extra information like symmetricity of $X_c(j\Omega)$,
we know that $\Omega_N = 200\pi$ rad/sec.
So, the minimum sampling rate is $2\Omega_N$ (Nyquist rate)
is $\boxed{400\pi \text{ rad/sec}}$, which is $\frac{400\pi}{2\pi \cancel{\text{rad}}} \cancel{\text{rad}}/\text{sec} = 200$ 1/s $\boxed{= 200 \text{ Hz}}$

(By using the extra information while modulating and demodulating, we can decrease twice as much by clever thinking.)

b) $T_s = \frac{1}{150}$ s.
$f_s = 150$ Hz

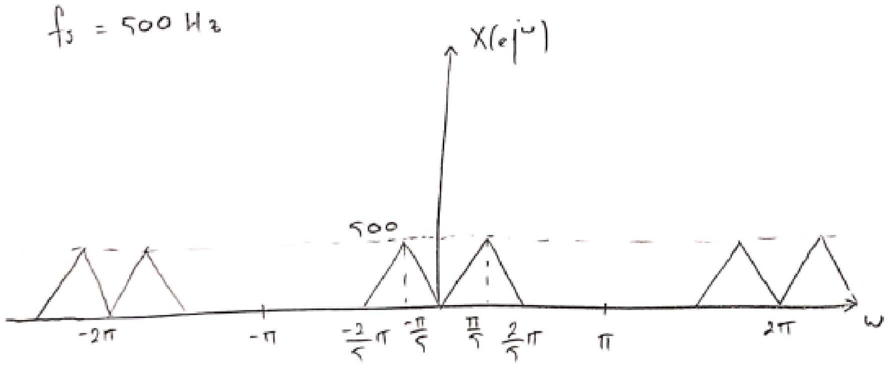



- Since $f_s < f_N$ (150 Hz < 200 Hz), aliasing occurs in the frequency band $\frac{2\pi}{3} \leq \omega + 2\pi k \leq \frac{4\pi}{3}$ where $k \in \mathbb{Z}$ in DTFT.
  - (flat regions)



c) $T_s = \frac{1}{500}$ s

$f_s = 500$ Hz




- Since $f_s > f_N$ (500 Hz > 200 Hz), aliasing does not occur.
- Highest frequency in the sampled signal is $\frac{2\pi}{5} \cdot 500 = \overset{\text{rad/sec}}{200\pi} \Rightarrow \frac{200\pi}{2\pi} = \overset{\text{Hz}}{100}$.

  $\boxed{100 \text{ Hz}} \leq B = f_s/2 = 250$ Hz

  check ✓
- $\Omega_a = 100\pi$ rad/s is given, corresponding discrete frequency is

  $w_a = \Omega_a T_s = 100\pi \cdot \frac{1}{500} = \boxed{\frac{\pi}{5} \text{ rad/sec}}$

2. The MATLAB code is both stated and attached as .m file. Necessary figures can be found in Figures 1, 2, and 3, which are 6-point, 9-point and 4-point FFTs of the given input sequence respectively.

   Conclusions can be stated as:

   (a) When the input is real as such, the abs() of FFT is perfectly symmetric.

   (b) 6-point fft of x and 9-point fft of x are not same since the number of samples taken from DTFT is different.

   (c) 9-point ifft of z (9-point fft of x) and x are same since the length of the sequence is 6.

   (d) 6-point ifft of z (9-point fft of x) and x are different since the number of points are different than each other.

   (e) 4-point ifft of v (4-point fft of x) and x are same for the first 4 elements since the length of the sequence is 6, which is less than 4.

```
clear; clc; close all;
x = [2 1 6 7 0 5];

y = fft(x,6);
figure
hold on
subplot(211)
stem(abs(y))
ylabel('abs(y)')
subplot(212)
stem(phase(y))
ylabel('phase(y)')
sgtitle('6-point fft of x');

z = fft(x,9);
figure
hold on
subplot(211)
stem(abs(z))
ylabel('abs(z)')
subplot(212)
stem(phase(z))
ylabel('phase(z)')
sgtitle('9-point fft of x');

v = fft(x,4);
figure
hold on
subplot(211)
stem(abs(v))
ylabel('abs(v)')
subplot(212)
stem(phase(v))
ylabel('phase(v)')
sgtitle('4-point fft of x');

x
x1 = ifft(z,9)
```

```
x2 = ifft(z,6)
x3 = ifft(v,4)
```

```
x =

     2     1     6     7     0     5


x1 =

  Columns 1 through 7

    2.0000    1.0000    6.0000    7.0000    0.0000    5.0000         0

  Columns 8 through 9

   -0.0000    0.0000


x2 =

  Columns 1 through 4

   3.7541 - 0.4054i   4.4964 - 3.0503i   7.8264 + 2.1864i   3.2094 - 2.1480i

  Columns 5 through 6

   0.4195 + 2.5491i   1.2943 + 0.8683i


x3 =

     2     1     6     7
```

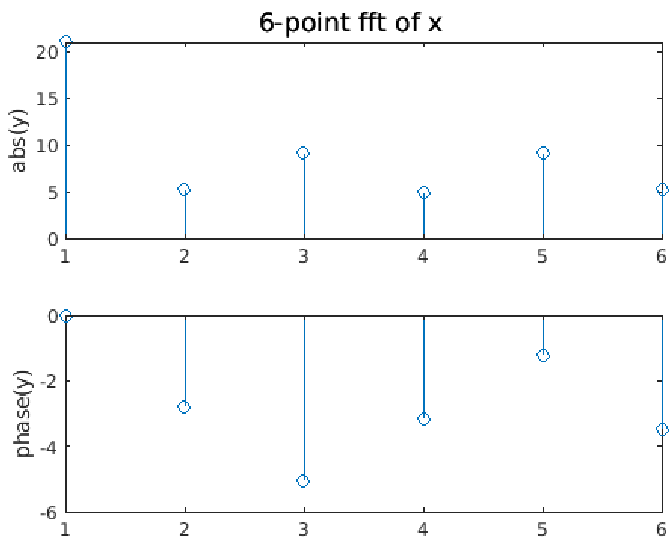



Figure 1: 6-point FFT of [2 1 6 7 0 5]

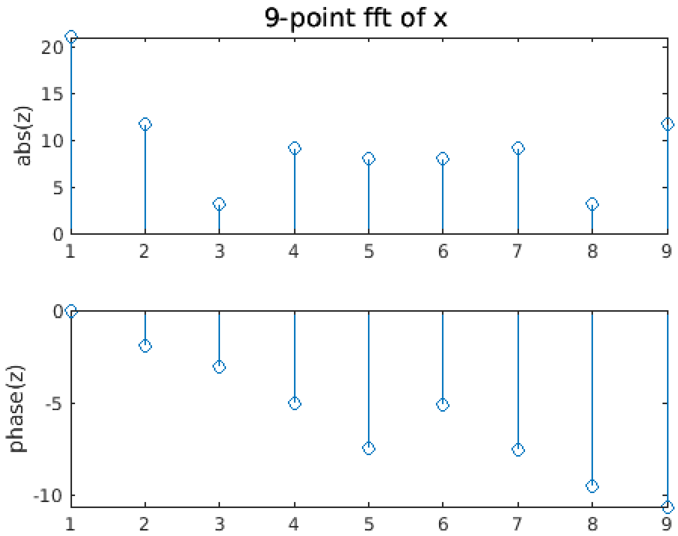



Figure 2: 9-point FFT of [2 1 6 7 0 5]

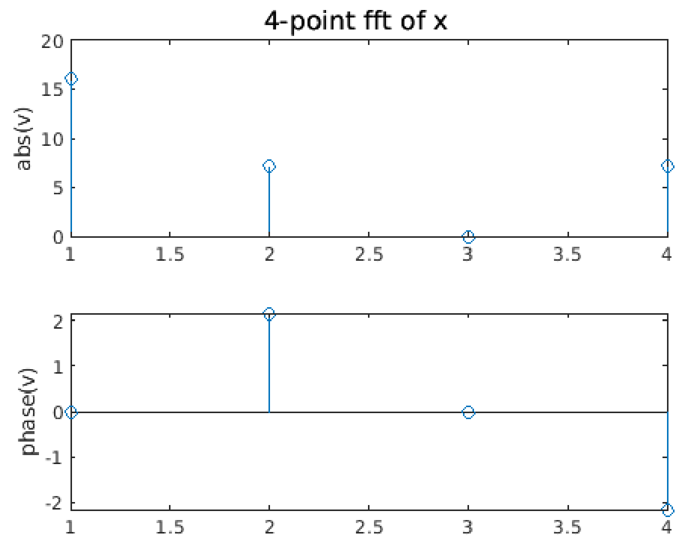



Figure 3: 4-point FFT of [2 1 6 7 0 5]

3. Implementation of convolution is done in LabVIEW and the results are shown in Figures 4, 5, 6 and 7.

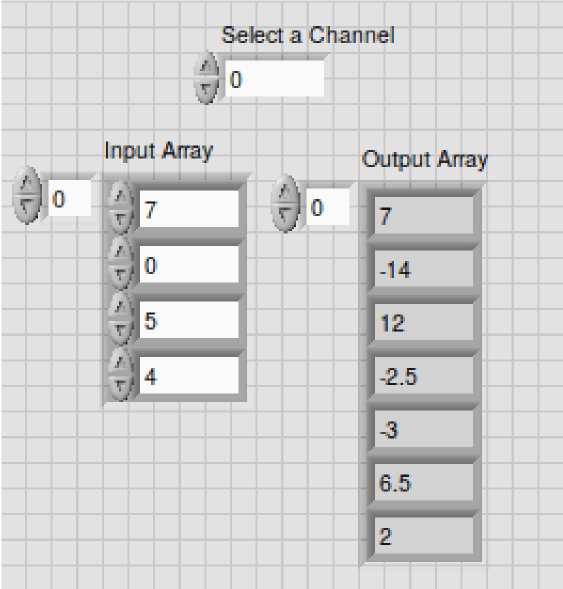



Figure 4: Screenshot of front panel for channel selection 0

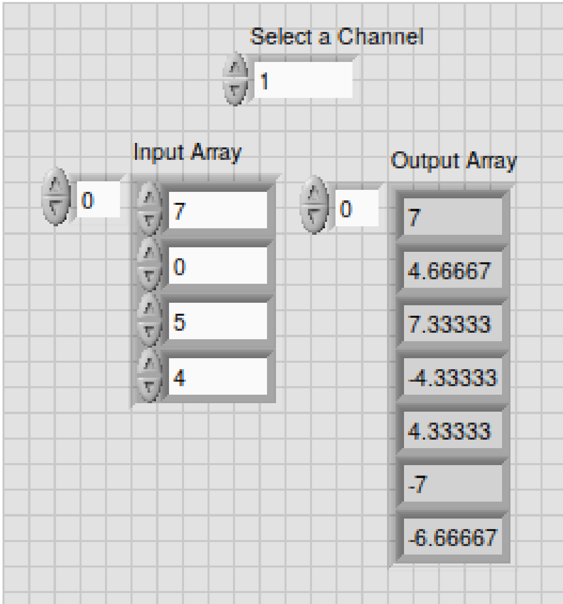



Figure 5: Screenshot of front panel for channel selection 1

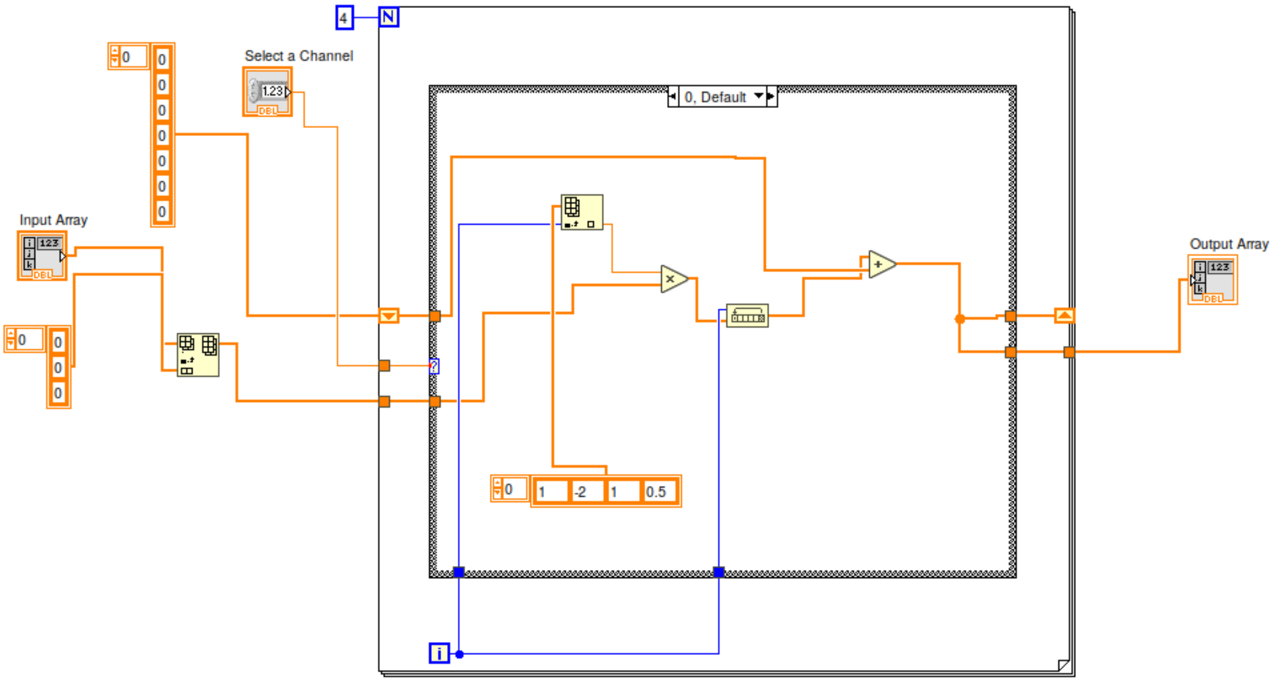



Figure 6: Screenshot of block diagram for channel selection 0

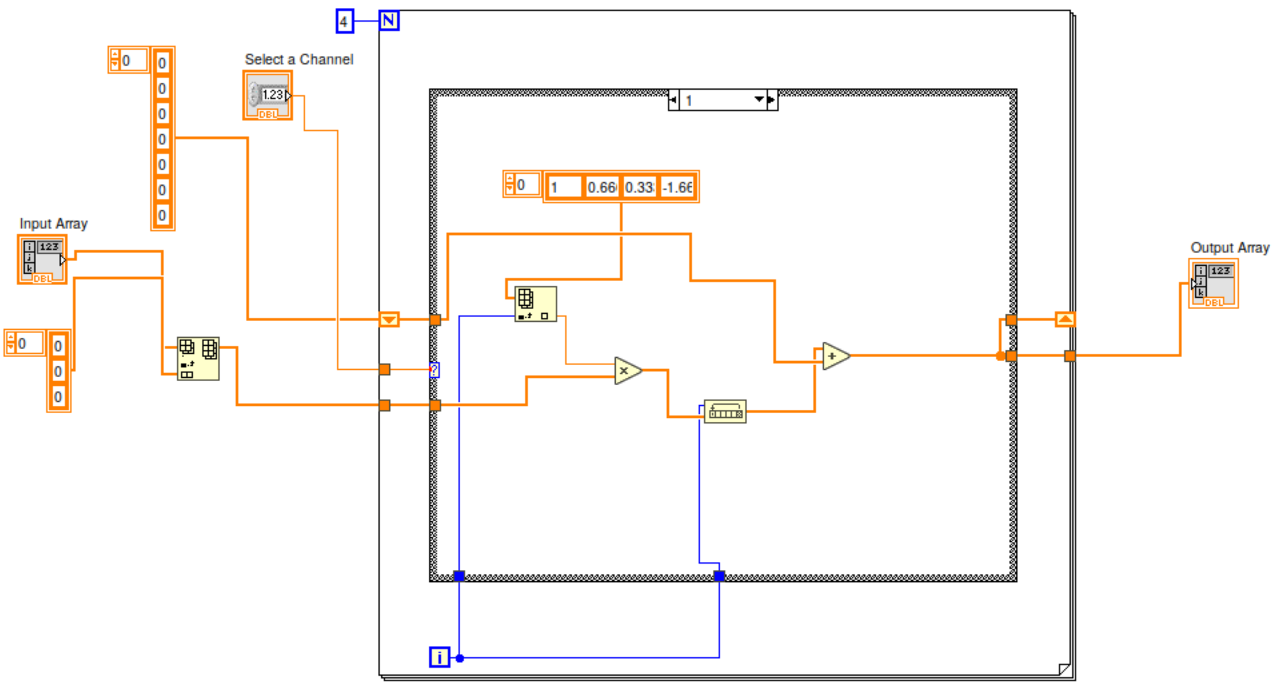



Figure 7: Screenshot of block diagram for channel selection 0