REPORT

Subject: **AADEC**

Lecturer: **prof. dr hab. inż. Vasyl Martsenyuk**

Lab Nr 2 Date 01.03.2024 Topic: "Windowing" Variant 6	Wojciech Kasolik Informatyka II stopień, stacjonarne, I semestr, gr. 1a

1. Problem Statement:

The objective is to be able the results of different type of windowing the signals

2. Input Data:

No	f_1	f_2	f_3	$\|x[k]\|_{\max}$	f_s	N
6	600	600.25	599.75	3	800	2000

3. Remote repository link:

https://github.com/Lorn-Hukka/AADEC/tree/main/lab02

4. Execution / Source Code:

```
import numpy as np
import matplotlib.pyplot as plt
from numpy.fft import fft, ifft, fftshift
from scipy.signal.windows import hann, flattop
```

```
f1 = 600 # Hz
f2 = 600.25 # Hz
f3 = 599.75 #Hz
fs = 800 # Hz
N = 2000
k = np.arange(N)
x1 = 3 * np.sin(2*np.pi*f1/fs*k)
x2 = 3 * np.sin(2*np.pi*f2/fs*k )
x3 = 3 * np.sin(2*np.pi*f3/fs*k )
```

```
wrect = np.ones(N)
whann = hann(N,sym=False)
wflattop = flattop(N, sym=False)
plt.plot(wrect, 'C0o-', ms=3, label='rect')
plt.plot(whann, 'C1o-', ms=3, label='hann')
plt.plot(wflattop, 'C2o-', ms=3, label='flattop')
plt.xlabel(r'$k$')
plt.ylabel(r'window $w [ k ] $')
plt.xlim(0, N)
plt.legend()
plt.grid(True)
```

```
X1wrect = fft(x1)
X2wrect = fft(x2)
X3wrect = fft(x3)
X1whann = fft(x1*whann)
```

```
X2whann = fft(x2*whann)
X3whann = fft(x3*whann)
X1wflattop = fft(x1*wflattop)
X2wflattop = fft(x2*wflattop)
X3wflattop = fft(x3*wflattop)
```

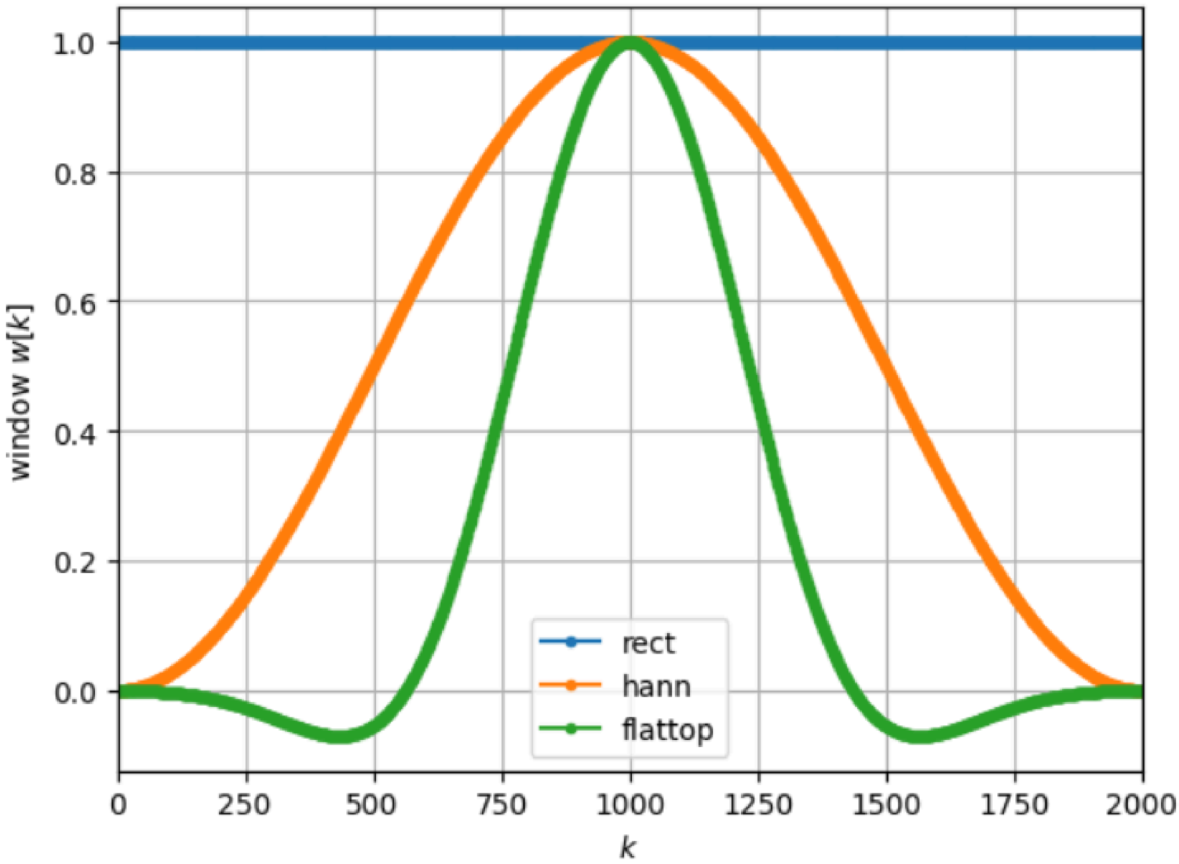
```
def fft2db(X):
    N = X.size
    Xtmp = 2/N*X
    Xtmp[0] *= 1/2
    if N%2 == 0:
        Xtmp[N//2] = Xtmp[N//2]/2
    return 20*np.log10(np.abs(Xtmp))
```

```
df = fs/N
f = np.arange(N)*df
```

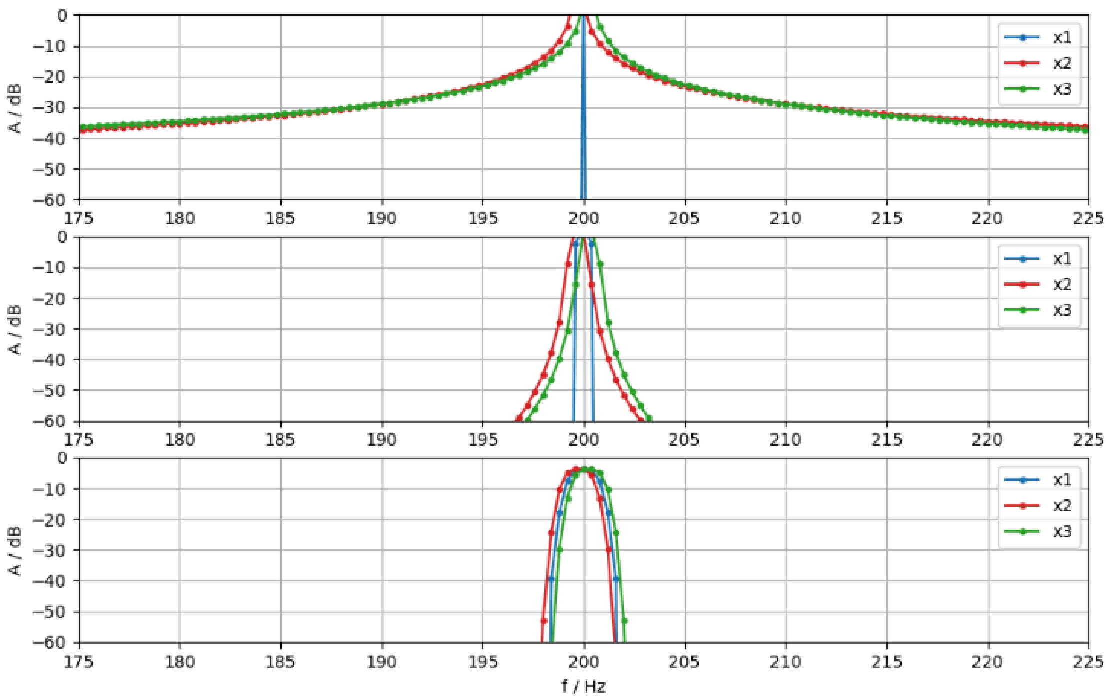
```
plt.figure(figsize = (16/1.5, 10/1.5))
plt.subplot(3, 1, 1)
plt.plot(f, fft2db(X1wrect), 'C0o-', ms=3, label='x1')
plt.plot(f, fft2db(X2wrect), 'C3o-', ms=3, label='x2')
plt.plot(f, fft2db(X3wrect), 'C2o-', ms=3, label='x3')
plt.xlim(175, 225)
plt.ylim(-60, 0)
plt.xticks(np.arange(175, 230, 5))
plt.yticks(np.arange(-60, 10, 10))
plt.legend()
plt.ylabel('A / dB')
plt.grid(True)
plt.subplot(3, 1, 2)
plt.plot(f, fft2db(X1whann), 'C0o-', ms=3, label='x1')
plt.plot(f, fft2db(X2whann), 'C3o-', ms=3, label='x2')
plt.plot(f, fft2db(X3whann), 'C2o-', ms=3, label='x3')
plt.xlim(175, 225)
plt.ylim(-60, 0)
plt.xticks(np.arange(175, 230, 5))
plt.yticks(np.arange(-60, 10, 10))
plt.legend()
plt.ylabel('A / dB')
plt.grid(True)
plt.subplot(3, 1, 3)
plt.plot(f, fft2db(X1wflattop), 'C0o-', ms=3, label='x1')
plt.plot(f, fft2db(X2wflattop), 'C3o-', ms=3, label='x2')
plt.plot(f, fft2db(X3wflattop), 'C2o-', ms=3, label='x3')
plt.xlim(175, 225)
plt.ylim(-60, 0)
plt.xticks(np.arange(175, 230, 5))
plt.yticks(np.arange(-60, 10, 10))
plt.legend()
plt.xlabel('f / Hz')
plt.ylabel('A / dB')
plt.grid(True)
```

```python
def winDTFTdB(w):
    N = w.size
    Nz = 100*N
    W = np.zeros(Nz)
    W[0:N] = w
    W = np.abs(fftshift(fft(W)))
    W /= np.max(W)
    np.seterr(divide = 'ignore')
    W = 20*np.log10(W)
    Omega = 2*np.pi/Nz*np.arange(Nz)-np.pi
    return Omega, W
```

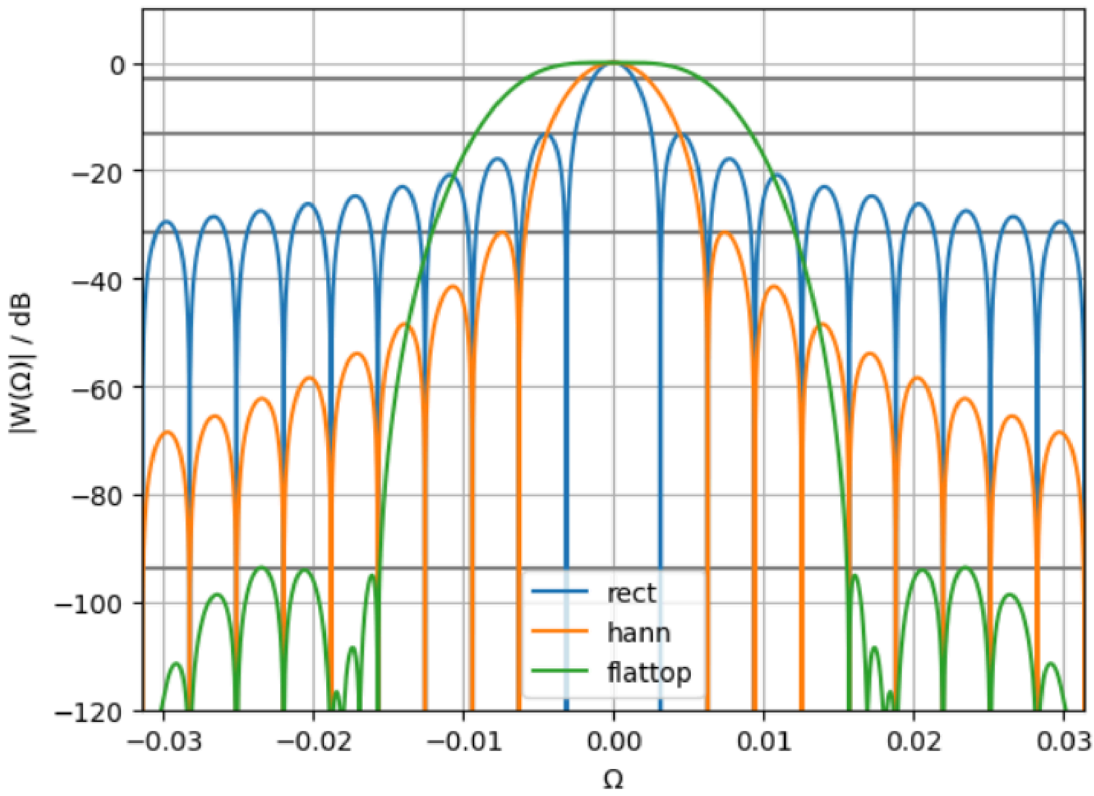
```python
plt.plot([-np.pi, +np.pi], [-3.01, -3.01], 'gray')
plt.plot([-np.pi, +np.pi], [-13.3, -13.3], 'gray')
plt.plot([-np.pi, +np.pi], [-31.5, -31.5], 'gray')
plt.plot([-np.pi, +np.pi], [-93.6, -93.6], 'gray')
Omega, W = winDTFTdB(wrect)
plt.plot(Omega, W, label='rect')
Omega, W = winDTFTdB(whann)
plt.plot(Omega, W, label='hann')
Omega, W = winDTFTdB(wflattop)
plt.plot(Omega, W, label='flattop')
plt.xlim(-np.pi, np.pi)
plt.ylim(-120, 10)
plt.xlim(-np.pi/100, np.pi/100)
plt.xlabel(r'$\Omega$')
plt.ylabel(r'|W($\Omega$)| / dB')
plt.legend()
plt.grid(True)
```



1.0
0.8
0.6
0.4
0.2
0.0
window $w[k]$
0
250
500
750
1000
1250
1500
1750
2000
k
rect
hann
flattop



A / dB
f / Hz
x1
x2
x3
0
-10
-20
-30
-40
-50
-60
175
180
185
190
195
200
205
210
215
220
225



5. Conclusions:

In conclusion, the analysis reveals that the choice of windowing function profoundly impacts the frequency domain representation of signals. Signals with frequencies close to integer multiples of the frequency resolution exhibit different behaviors under various windows, influencing their spectral characteristics. The Hann window generally offers improved frequency resolution and reduced sidelobe levels compared to the rectangular window, while the Flattop window minimizes amplitude errors caused by spectral leakage. These findings underscore the importance of selecting appropriate windowing techniques to accurately capture and analyze the frequency content of signals in practical applications.