REPORT

Zajęcia: Analog and digital electronic circuits
Teacher: prof. dr hab. Vasyl Martsenyuk

Lab 2
01.03.2024
Topic: "Windowing"
Variant 2

Wiktor Merta
Informatyka II stopień,
stacjonarne,
1 semestr,
Gr.2

1. Problem statement: The objective is to be able the results of different type of windowing the signals

2. Input data:

$f_1 = 400$
$f_2 = 400.25$
$f_3 = 399.75$
$|x[k]|_{max} = 2$
$f_s = 600$
$N = 3000$

3. Commands used (or GUI):

a) source code

Generating signals

```
f1 = 400  # Hz
f2 = 400.25  # Hz
f3 = 399.75  # Hz
fs = 600  # Hz
N = 3000
k = np.arange(N)

x1 = 2 * np.sin(2 * np.pi * f1 / fs * k)
x2 = 2 * np.sin(2 * np.pi * f2 / fs * k)
x3 = 2 * np.sin(2 * np.pi * f3 / fs * k)
```

Generating windows

```
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)
```

Windowing signals

```
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)
```

Plotting DFT

```
plt.figure(figsize=(16 / 1.5, 10 / 1.5))
plt.subplot(3, 1, 1)
plt.plot(f, fft2db(X1wrect), "C0o-", ms=3, label="best case rect")
plt.plot(f, fft2db(X2wrect), "C3o-", ms=3, label="worst case rect")
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)

plt.subplot(3, 1, 2)
plt.plot(f, fft2db(X1whann), "C0o-", ms=3, label="best case hann")
plt.plot(f, fft2db(X2whann), "C3o-", ms=3, label="worst case hann")
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)

plt.subplot(3, 1, 3)
plt.plot(f, fft2db(X1wflattop), "C0o-", ms=3, label="best case flattop")
plt.plot(f, fft2db(X2wflattop), "C3o-", ms=3, label="worst case flattop")
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)
```

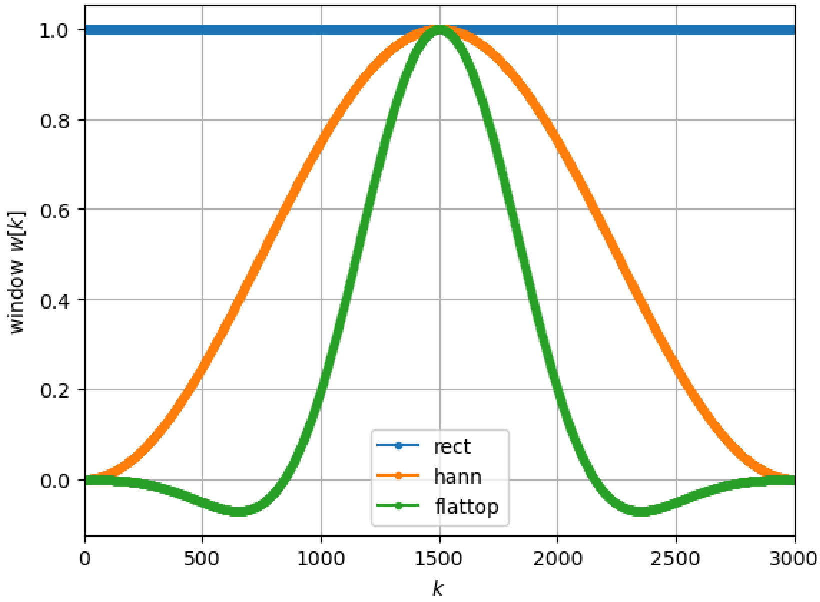
Plot normalized DTFT to maximum

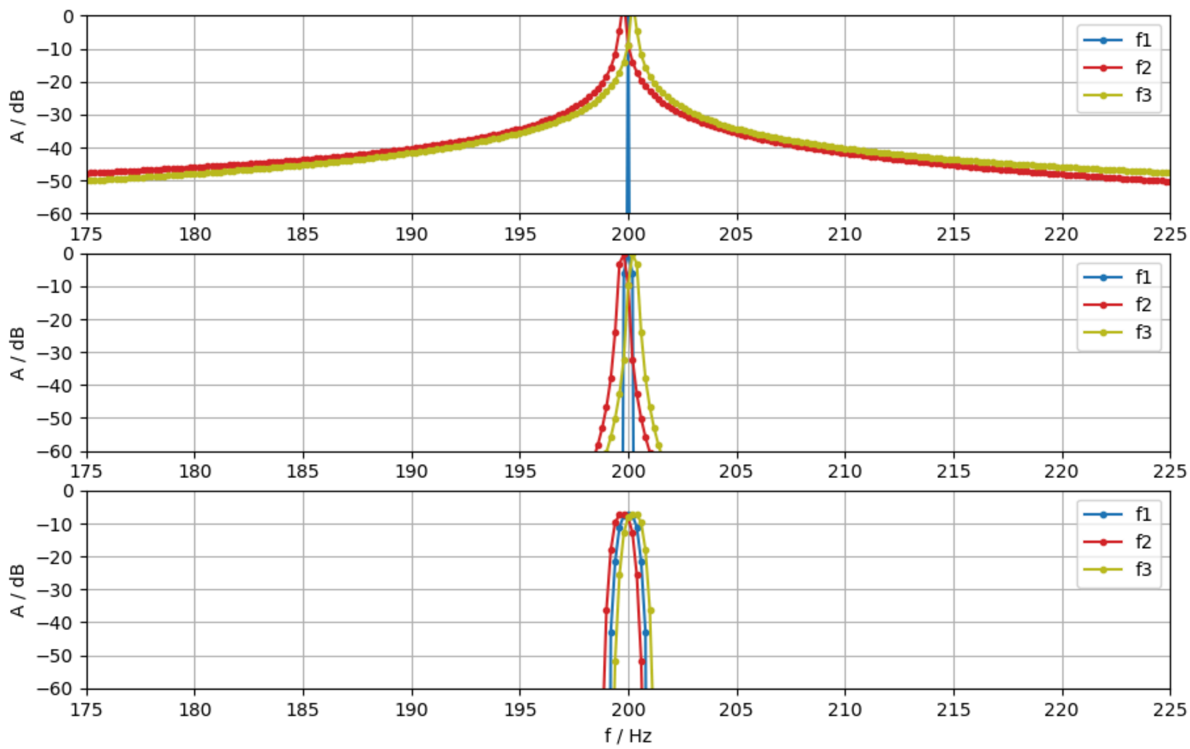
```
plt.plot([-np.pi, +np.pi], [-3.01, -3.01], "gray")  # mainlobe bandwidth
plt.plot([-np.pi, +np.pi], [-13.3, -13.3], "gray")  # rect max sidelobe
plt.plot([-np.pi, +np.pi], [-31.5, -31.5], "gray")  # hann max sidelobe
plt.plot([-np.pi, +np.pi], [-93.6, -93.6], "gray")  # flattop max sidelobe
Omega, 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)  # zoom into mainlobe
```

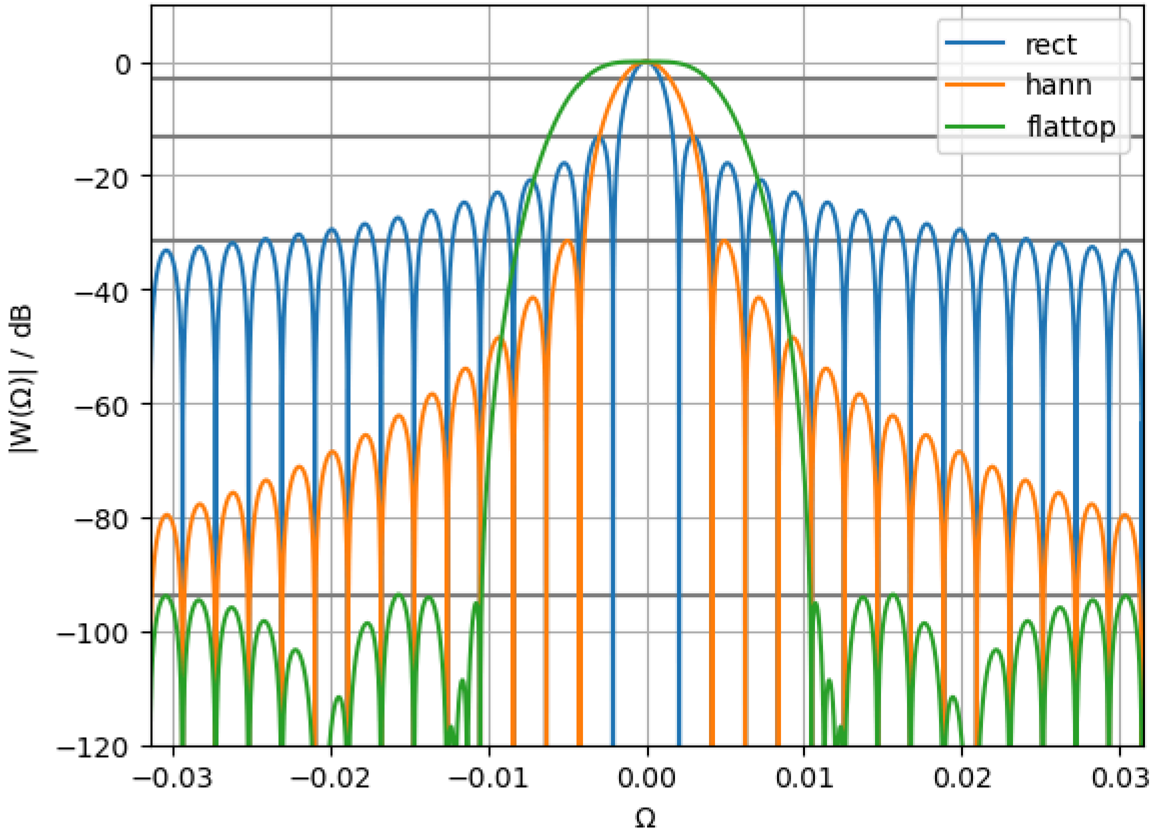
```
plt.xlabel(r"$\Omega$")
plt.ylabel(r"|W($\Omega$)| / dB")
plt.legend()
plt.grid(True)
```

https://github.com/wm64167/AADEC

4. Outcomes:







5. Conclusions: For the reasons given, we conclude that signal windowing helps to perform signal analysis, transforming values depending on sampling frequency. The differences in the results for f1 and f2 arise from the difference between the signal frequencies and sampling frequency resolution which is close to signal frequencies.