**REPORT**

Zajęcia: Analog and digital electronic circuits

Teacher: prof. dr hab. Vasyl Martsenyuk

**Lab 3 and 4**

17.02.2025

**Topic:** Windowing

**Variant 15**

Tobiasz Wojnar  
 Informatyka II stopień,   
niestacjonarne,   
1 semestr,   
Gr. B

# 1. Problem statement:

Creation of tree sine signals with specified fl, f2 and f3 frequencies. Maximum amplitude |x[k]|max, using a sampling frequency fs over the range

0 <= k < N

Two visualizations:

* DFT (Discrete Fourier Transform)
* DTFT (Discrete-Time Fourier fransform)

# 2. Input data:

* fl**=** 500
* f2 **=** 500.25
* f3 **=** 499.75
* |x[k]|max **=** 4
* fs **=** 800
* N **=** 2000

# 3. Commands used (or GUI):

## a) source code

*# importing libraries*

**import** numpy **as** np

**import** matplotlib.pyplot **as** plt

**from** numpy.fft **import** fft, ifft, fftshift

**from** scipy.signal.windows **import** hann, flattop

f1 = 500 *# Hz*

f2 = 500.25 *# Hz*

f3 = 499.75 *# Hz*

fs = 800 *# Hz*

N = 1600

k = np.arange (N)

max\_amplitude = 4

x1 = np.sin(2\*np.pi \* f1 / fs \* k )\*max\_amplitude

x2 = np.sin(2\*np.pi \* f2 / fs \* k )\*max\_amplitude

x3 = np.sin(2\*np.pi \* f3 / fs \* k )\*max\_amplitude

*# Generate window functions*

w\_rect = np.ones(N)

w\_hann = hann(N, sym=**False**)

w\_flattop = flattop(N, sym=**False**)

plt.plot(w\_rect, 'C3o-', ms=3, label='Rect')

plt.plot(w\_hann, 'C1o-', ms=3, label='Hann')

plt.plot(w\_flattop, 'C7o-', ms=3, label='Flattop')

plt.xlabel('$k$')

plt.ylabel('window $w[k]$')

plt.xlim(0,N)

plt.legend()

plt.grid()

**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

**def** plotComparison(x\_1, x\_2, x\_3, window\_name,plotNr):

plt.subplot( 3, 1, plotNr)

plt.plot(f, fft2db(x\_1), 'C2o-', ms=3, label=f'f1 case {window\_name}')

plt.plot(f, fft2db(x\_2), 'C3o-', ms=3, label=f'f2 case {window\_name}')

plt.plot(f, fft2db(x\_3), 'C5o-', ms=3, label=f'f3 case {window\_name}')

plt.xlim(275,275+50)

plt.ylim(-50,20)

plt.xticks(np.arange(275,275+50,5))

plt.yticks(np.arange(-60,20,10))

plt.xlabel('f/Hz')

plt.ylabel('A / dB')

plt.legend()

plt.grid()

plt.figure(figsize=(16/1.5, 10/1.5))

plotComparison(x1\_w\_rect,x2\_w\_rect,x3\_w\_rect,'rect',1)

plotComparison(x1\_w\_hann,x2\_w\_hann,x3\_w\_hann,'hann',2)

plotComparison(x1\_w\_flattop,x2\_w\_flattop,x3\_w\_flattop,'flattop',3)

**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) *# normalize*

W = 20\*np.log10(W)

Omega = 2\*np.pi/Nz\*np. arange(Nz) - np.pi

**return** Omega, W

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(w\_rect)

plt.plot(Omega, W, label='rect')

Omega, W = winDTFTdB(w\_hann)

plt.plot(Omega, W, label='hann')

Omega, W = winDTFTdB(w\_flattop)

plt.plot(Omega, W, label='flattop')

plt.xlim(-np.pi/100,np.pi/100) *# zoom into mainlobe*

plt.ylim(-120,10)

plt.xlabel(r'$\Omega$')

plt.ylabel(r'|W($\Omega)| / dB$')

plt.legend()

plt.grid(**True**)

## b) Link to remote repozytorium

<https://github.com/TobiaszWojnar/DSP>

# 4. Outcomes:

A graph of a graph of a function

AI-generated content may be incorrect.

A graph of a function

AI-generated content may be incorrect.

A graph with colored lines and numbers

AI-generated content may be incorrect.

# 5. Conclusions

The selection of a window function for signal spectrum analysis is crucial and depends on the desired information. Each window has trade-offs affecting measurement quality. While the rectangular window provides the best frequency resolution (though suffers from sidelobes), the Hanning window offers a good balance between amplitude accuracy and spectral leakage reduction. Finally, the Flattop window prioritizes amplitude accuracy, even at the cost of reduced frequency resolution.

**Rectangular Window**

The rectangular window's advantage lies in its lack of signal modification, resulting in superior frequency resolution, essential for pinpointing specific frequencies. However, this benefit is offset by significant sidelobes in the resulting spectrum, potentially obscuring accurate signal analysis due to interference.

**Hanning Window**

The Hanning window offers a compromise between frequency resolution and sidelobe suppression. Its smoothing effect at signal boundaries reduces spectral leakage and improves amplitude accuracy. Although it sacrifices some frequency resolution compared to the rectangular window, the Hanning window's overall improvement in spectral quality makes it a popular choice in many applications.

**Flattop Window**

The Flattop window prioritizes amplitude accuracy, producing highly precise measurements due to its flat-topped spectral response. 1 This precision comes at the expense of a wider main lobe, reducing frequency resolution. However, for applications where amplitude accuracy is paramount, such as high-precision spectral measurements, the Flattop window is the optimal choice, provided the reduced frequency resolution is acceptable