REPORT

Subject: Digital Signal Processing
Lecturer: prof. dr hab. Vasyl Martsenyuk

Laboratory #2 Date: 18.10.2024 Topic: Dyskretna transformacja Fouriera (DFT): obliczanie DFT sygnałów oraz analiza wyników. Implementacja algorytmów FFT: porównanie wydajności różnych implementacji FFT. Second variant (2)	Bartosz Bieniek IT Science II degree, 1 semester, gr.A

1. Task:

Generate three sine signals of given f_1, f_2, and f_3 and amplitude |x[k]|_max for the sampling frequency f_s in the range of leq k<N.

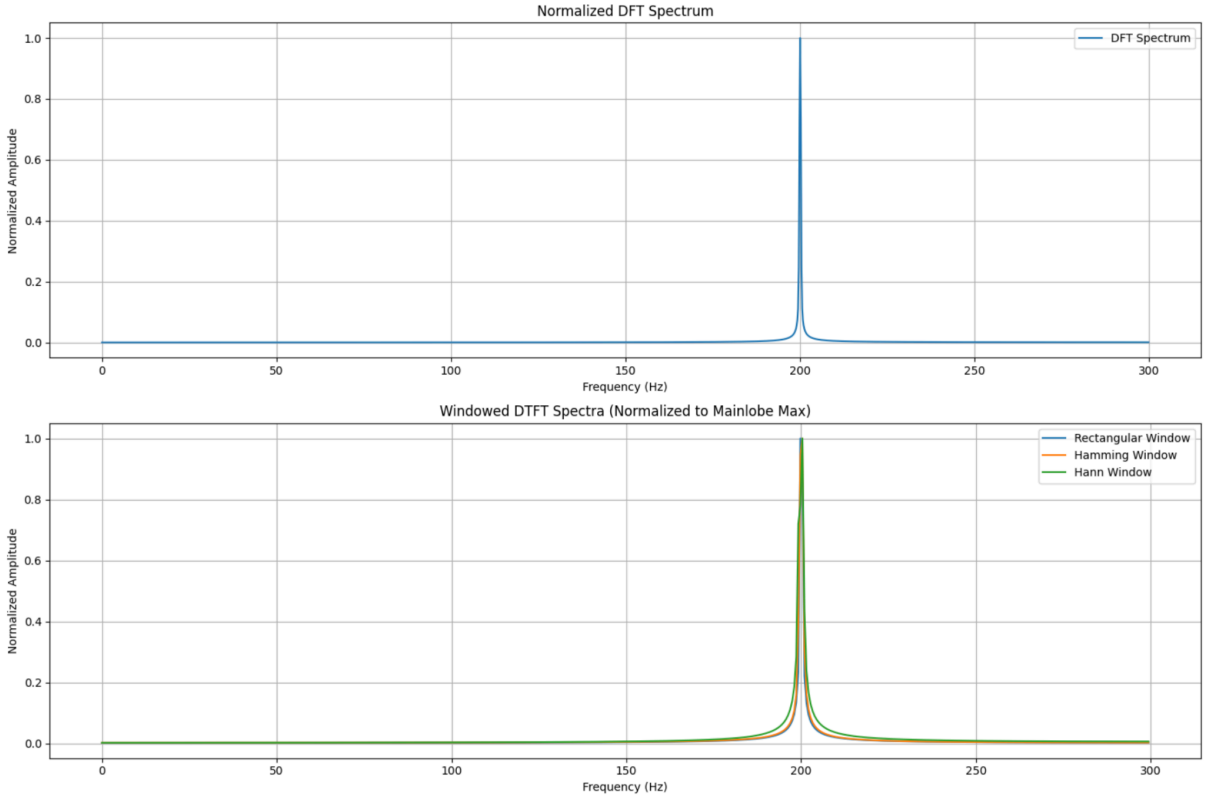
Variant: f1 f2 f3 |x[k]|max fs N are equals: 400 400.25 399.75 2 600 3000

2. Code description [Github Repository]

```python
# Generate three sine signals of given f_1, f_2, and f_3 and amplitude |x[k]|_max for the sampling frequency f_s in the range of leq k<N.
# Variant:
# f1 f2 f3 |x[k]|max fs N
# 400 400.25 399.75 2 600 3000

import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import get_window

# Parameters
f1, f2, f3 = 400, 400.25, 399.75  # Frequencies in Hz
fs = 600  # Sampling frequency in Hz
N = 3000  # Number of samples
amplitude = 2  # Maximum amplitude
t = np.arange(N) / fs # Time vector

# Generate sine signals
x1 = amplitude * np.sin(2 * np.pi * f1 * t)
x2 = amplitude * np.sin(2 * np.pi * f2 * t)
x3 = amplitude * np.sin(2 * np.pi * f3 * t)

x = x1 + x2 + x3 # Combined signal

X = np.fft.fft(x, n=N) # Compute DFT
X_normalized = np.abs(X) / np.max(np.abs(X))  # Normalize DFT
freqs = np.fft.fftfreq(N, d=1/fs)  # Frequency vector

# Windowing
windows = {
    "Rectangular": np.ones(N),
    "Hamming": get_window("hamming", N),
    "Hann": get_window("hann", N)
}

# Compute DTFT for each windowed signal
dtft_results = {}
for name, window in windows.items():
    x_windowed = x * window
    DTFT = np.fft.fft(x_windowed, n=1024)  # DTFT higher resolution
    DTFT_normalized = np.abs(DTFT) / np.max(np.abs(DTFT))  # Normalize mainlobe max
    dtft_results[name] = (DTFT_normalized, np.fft.fftfreq(1024, d=1/fs))

plt.figure(figsize=(15, 10))

# Plot normalized DFT spectra
plt.subplot(2, 1, 1)
plt.plot(freqs[:N//2], X_normalized[:N//2], label="DFT Spectrum")
plt.title("Normalized DFT Spectrum")
plt.xlabel("Frequency (Hz)")
plt.ylabel("Normalized Amplitude")
plt.legend()
plt.grid()

# Plot windowed DTFT spectra
plt.subplot(2, 1, 2)
for name, (dtft, freqs_dtft) in dtft_results.items():
    plt.plot(freqs_dtft[:512], dtft[:512], label=f"{name} Window")
plt.title("Windowed DTFT Spectra (Normalized to Mainlobe Max)")
plt.xlabel("Frequency (Hz)")
plt.ylabel("Normalized Amplitude")
plt.legend()
plt.grid()

plt.tight_layout()
plt.show()
```

TD. 1. Source code

The code was developed to analyze the spectral properties of three sine signals with specific frequencies. The signals were generated based on given parameters, including sampling frequency, number of samples, and maximum amplitude. A Discrete Fourier Transform (DFT) was applied to the combined signal, and the resulting spectrum was normalized for better interpretability. To evaluate the effects of windowing, three window functions (Rectangular, Hamming, and Hann) were applied to the signal. For each windowed signal, a Discrete-Time Fourier Transform (DTFT) was computed and normalized to the maximum of the mainlobe. The results were visualized in two plots: the normalized DFT spectrum and the normalized DTFT spectra for the different windows. The code enabled a comparison of spectral leakage and resolution across the windows.



TD. 2. Charts plotted by executed program

The resulting plots displayed the spectral characteristics of the signal and the effects of windowing. The top plot showed that the normalized DFT spectrum had distinct peaks that corresponded to the component frequencies of the input signal, with minimal leakage due to the high resolution. The bottom plot was the DTFT spectra for different window functions, which showed the difference in spectral leakage and mainlobe width. As was expected, the Rectangular window had the most leakage, while the Hamming and Hann windows had reduced leakage, but at the cost of a slightly broader mainlobe.

These results illustrated the trade off between leakage and resolution for different windows applied to the signal.

3. Conclusions

The analysis acknowledged that the choice of window radically influenced the spectral properties of signal. The Hamming and Hann windows were more efficient in reducing spectral leakage comparing to the Rectangular window, making them preferable for signals with familiar frequencies. However, the broader mainlobes of the windows indicated and reduced frequency resolution, which could set the boundaries of their effectiveness in certain applications. It was also observed that the small frequency differences between f1, f2, and f3 resulted in overlapping spectra.