DZ2 - Daljinska istraživanja

Viktor Horvat

December 2023

1 Prvi zadatak

Numeričkom analizom dobili smo sljedeće podatke i prikaze rezultata:

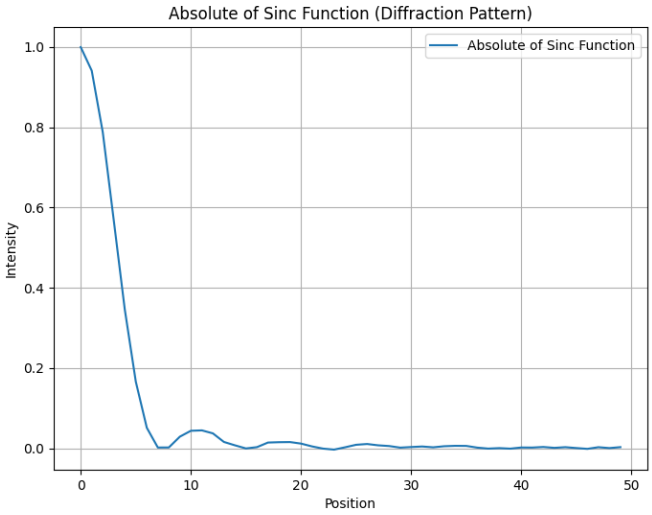

Figure 1: Očitani podaci fotodetektora

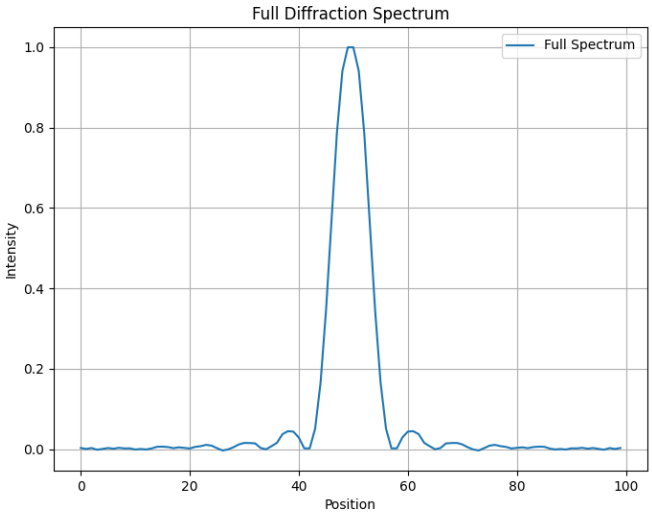

Figure 2: Rekonstruirani spektar na osnovi danih podataka

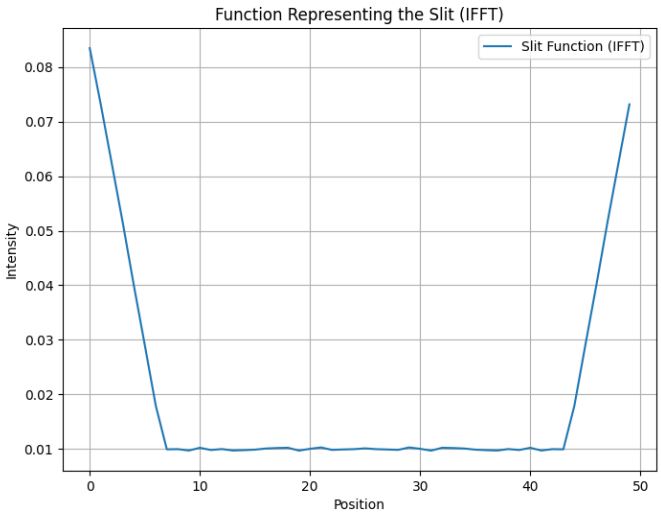

Figure 3: Vizualni prikaz širine aperture nakon IFT transformacije

Širina aperture: 52.668 μm

Izvorni kod korišten za dobivanje grafova i rezultata:

```
import numpy as np
import matplotlib.pyplot as plt

wavelength = 532e-9
distance = 1
file_name = 'apertura.txt'
positiveIntensityData = readIntensityData(file_name)

def readIntensityData(file_name):
    with open(file_name, 'r') as file:
        intensityData = np.loadtxt(file)
    return intensityData

def calculateSlitSize(intensity_data):
    slit = np.fft.ifft(intensity_data)
    firstMinimum = np.where(np.abs(slit) ==
    ↪  min(np.abs(slit)))[0][0]
    slitSize = firstMinimum * wavelength * distance
    return slit

def plotSlit(slit_function):
    plt.figure(figsize=(8, 6))
```

```
    plt.plot(np.real(slit_function), label='Slit Function (IFFT)')
    plt.xlabel('Position')
    plt.ylabel('Intensity')
    plt.title('Function Representing the Slit (IFFT)')
    plt.legend()
    plt.grid(True)
    plt.show()
    return

def plot_original(intensity_data):
    plt.figure(figsize=(8, 6))
    plt.plot(intensity_data, label='Positive Side of Spectrum')
    plt.xlabel('Position')
    plt.ylabel('Intensity')
    plt.title('Positive Side of Spectrum (Absolute of Sinc Function)')
    plt.legend()
    plt.grid(True)
    plt.show()
    return

def plotSinc(fullData):
    plt.figure(figsize=(8, 6))
    plt.plot(fullData, label='Full Spectrum')
    plt.xlabel('Position')
    plt.ylabel('Intensity')
    plt.title('Full Diffraction Spectrum')
    plt.legend()
    plt.grid(True)
    plt.show()
    return


negativeIntensityData = np.flip(positive_intensity_data)
fullData = np.concatenate((negativeIntensityData, positiveIntensityData))

plot_original(positiveIntensityData)
slit = calculateSlitSize(fullData)
plot_slit(slit)

print(slit)

plotSinc(fullData)
```

2 Drugi zadatak

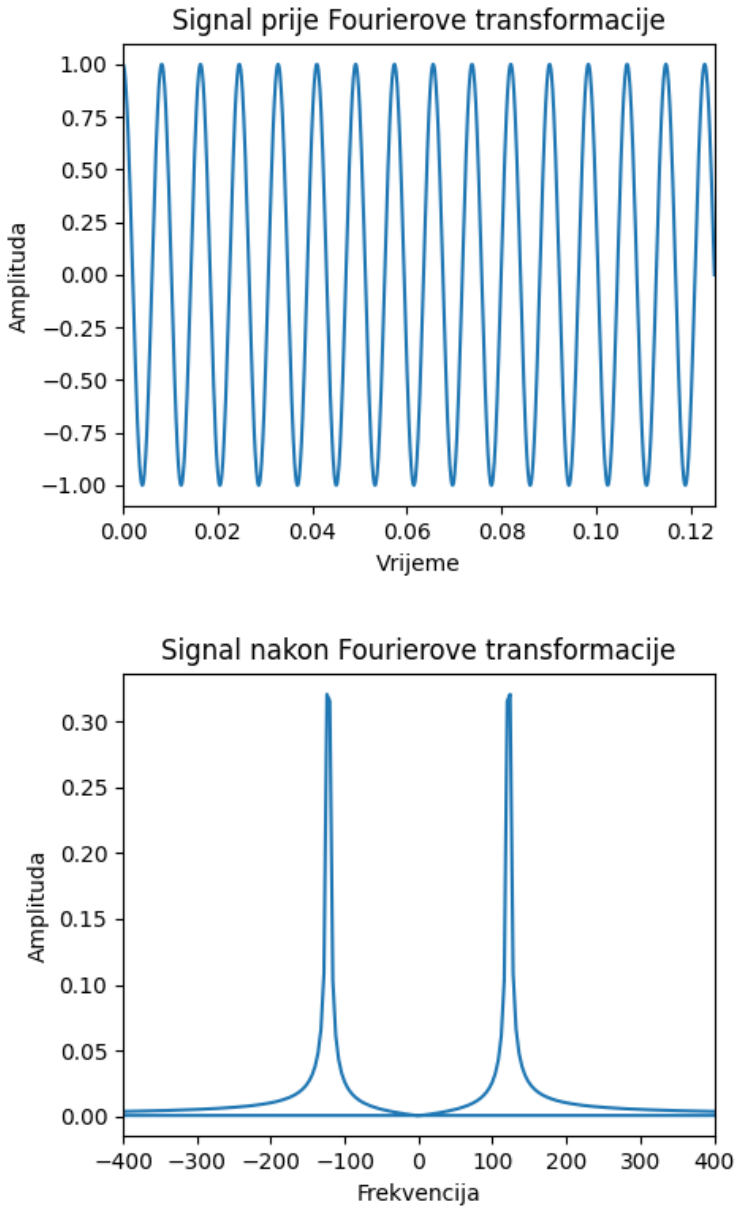

Figure 4: Dobivanje frekvencije korištene za primjenu Dopplerove formule

Analitičkim pristupom, korištenjem sljedećeg izvornog koda dobili smo sljedeće podatke:

```python
import numpy as np
import matplotlib.pyplot as plt

data = np.loadtxt('doppler.txt')
vrijeme = data[:, 0]
amplituda = data[:, 1]
frekvencijaGeneriranogSignala = 2.27e9  # 2.27 GHz

delta_t = vrijeme[1] - vrijeme[0]
broj_uzoraka = len(vrijeme)
frekvencija_uzorkovanja = 1 / delta_t

frekvencijskiSpektar = np.fft.ifft(amplituda)
frekvencije = np.fft.fftfreq(broj_uzoraka, delta_t)
indeksMaksimuma = np.argmax(np.abs(frekvencijskiSpektar))
najvecaFrekvencija = frekvencije[indeksMaksimuma]


brzina_objekta = (najvecaFrekvencija * 3e8) / (2 *
↪  frekvencijaGeneriranogSignala)

print("Izračunata brzina objekta:", brzina_objekta, "ms")
```

Ovakvom numeričkom analizom, rezultantna brzina objekta koji nam se približava je: 8.193 ms^{-1}