

Cyfrowe przetwarzanie sygnałów i obrazów

Przetwarzanie i analiza sygnału EKG

Autorki: Barbara Parzonka 263955, Joanna Zoglowek 264452

Ćwiczenie 2. Celem ćwiczenia jest praktyczne wypróbowanie funkcji `numpy.fft` i `numpy.ifft` do wyznaczania prostej i odwrotnej transformaty Fouriera [1, 3].

1. Wygeneruj ciąg próbek odpowiadający fali sinusoidalnej o częstotliwości 50 Hz i długości 65536.
2. Wyznacz dyskretną transformatę Fouriera tego sygnału i przedstaw jego widmo amplitudowe na wykresie w zakresie częstotliwości $[0, f_s/2]$, gdzie f_s oznacza częstotliwość próbkowania.
3. Wygeneruj ciąg próbek mieszaniny dwóch fal sinusoidalnych (tzn. ich kombinacji liniowej) o częstotliwościach 50 i 60 Hz. Wykonaj zadanie z punktu 2 dla tego sygnału.
4. Powtórz eksperymenty dla różnych czasów trwania sygnałów, tzn. dla różnych częstotliwości próbkowania.
5. Wyznacz odwrotne transformaty Fouriera ciągów wyznaczonych w zadaniu 2 i porównaj z ciągami oryginalnymi.

Opis realizacji zadania

Realizacji zadania służy interfejs z użytkownikiem umożliwiający wybór badanego sygnału i częstotliwości próbkowania sygnału.

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
import ipywidgets as widgets
%run methods.ipynb

"""
x to wektor probek,
fs to czestotliwosc probkowania
"""

n=65536
freq5=50
freq6=60
x5=[]
x6=[]
x56=[]
time=[]

def countSignals(sampling):
    global x5, x6, x56, time
    T=1/sampling
    time = np.linspace(0, (n-1)*T, n)
```

```

x5= np.sin(2 * np.pi * freq5 * time)
x6=np.sin(2 * np.pi * freq6 * time)
x56=x5+x6

f1=widgets.Checkbox(
    value=False,
    description='Uzyj sinusa o czestotliwosci 50 Hz',
    disabled=False
)
f1.style.description_width = 'auto'

f2=widgets.Checkbox(
    value=False,
    description='Uzyj sinusa o czestotliwosci 60 Hz',
    disabled=False
)
f2.style.description_width = 'auto'

samplingFreq=widgets.IntSlider(
    value=130,
    min=25,
    max=200,
    step=1,
    description='Czestotliwosc probkowania',
    continuous_update=False,
)
display(f1)
display(f2)
display(samplingFreq)

```

```

Checkbox(value=False, description='Uzyj sinusa o czestotliwosci 50 Hz', style=Checkb
oxStyle(description_width=...
Checkbox(value=False, description='Uzyj sinusa o czestotliwosci 60 Hz', style=Checkb
oxStyle(description_width=...
IntSlider(value=130, continuous_update=False, max=200, min=25)

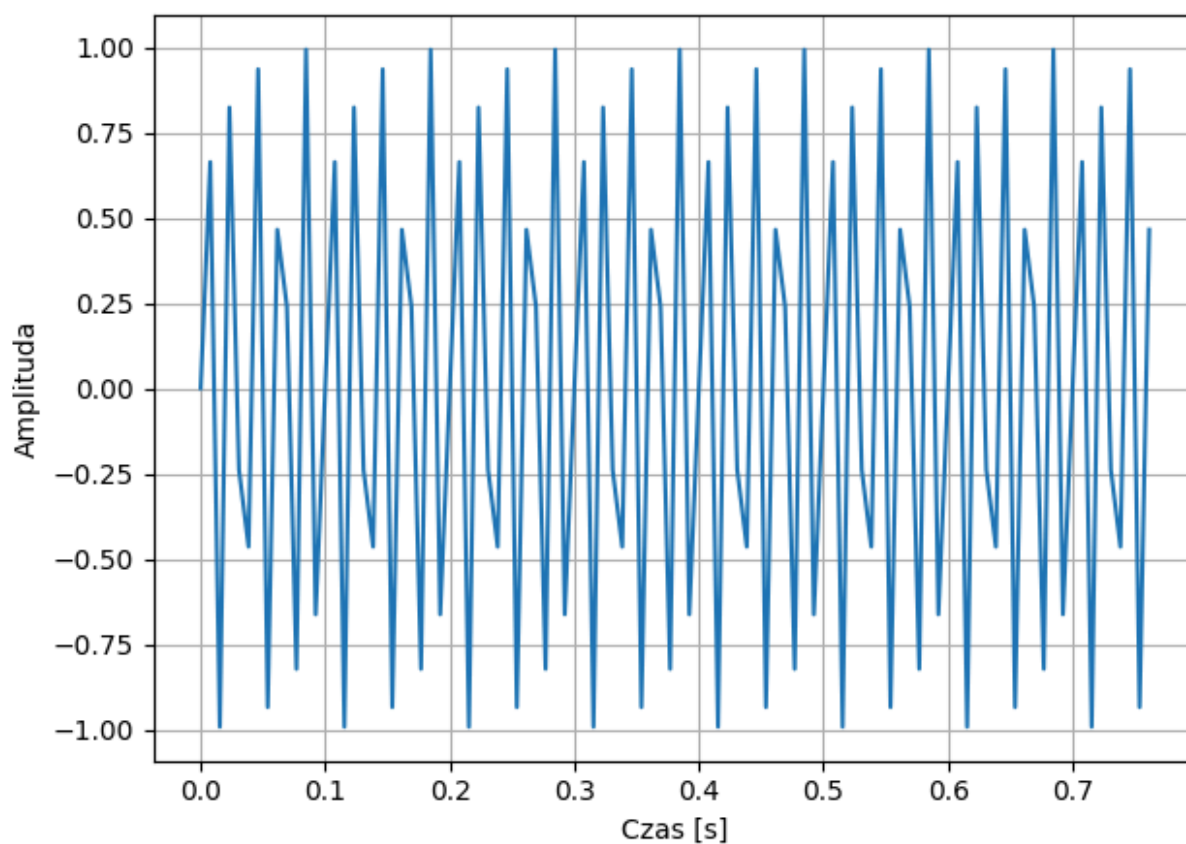
```

```

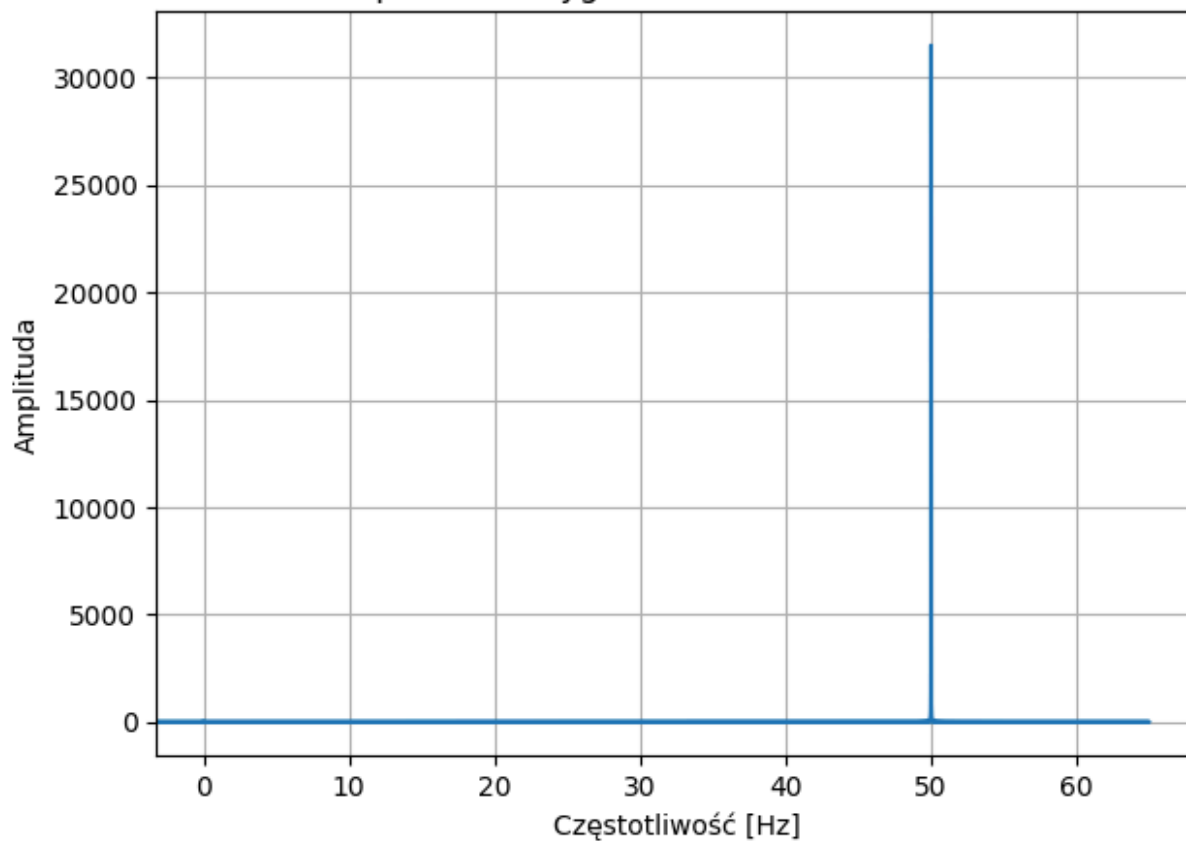
In [3]: count5=f1.value
count6=f2.value
fs=int(samplingFreq.value)
countSignals(fs)
k=100
if count5 and not count6:
    fourierIOdwrotny(x5,time, fs, 'sinus o czestotliwosci 50 Hz',k)
elif not count5 and count6:
    fourierIOdwrotny(x6,time, fs, 'sinus o czestotliwosci 60 Hz',k)
elif count5 and count6:
    fourierIOdwrotny(x56,time, fs, 'sinus mieszany (50 i 60 Hz)',k)
else:
    print('Wybierz sygnal')

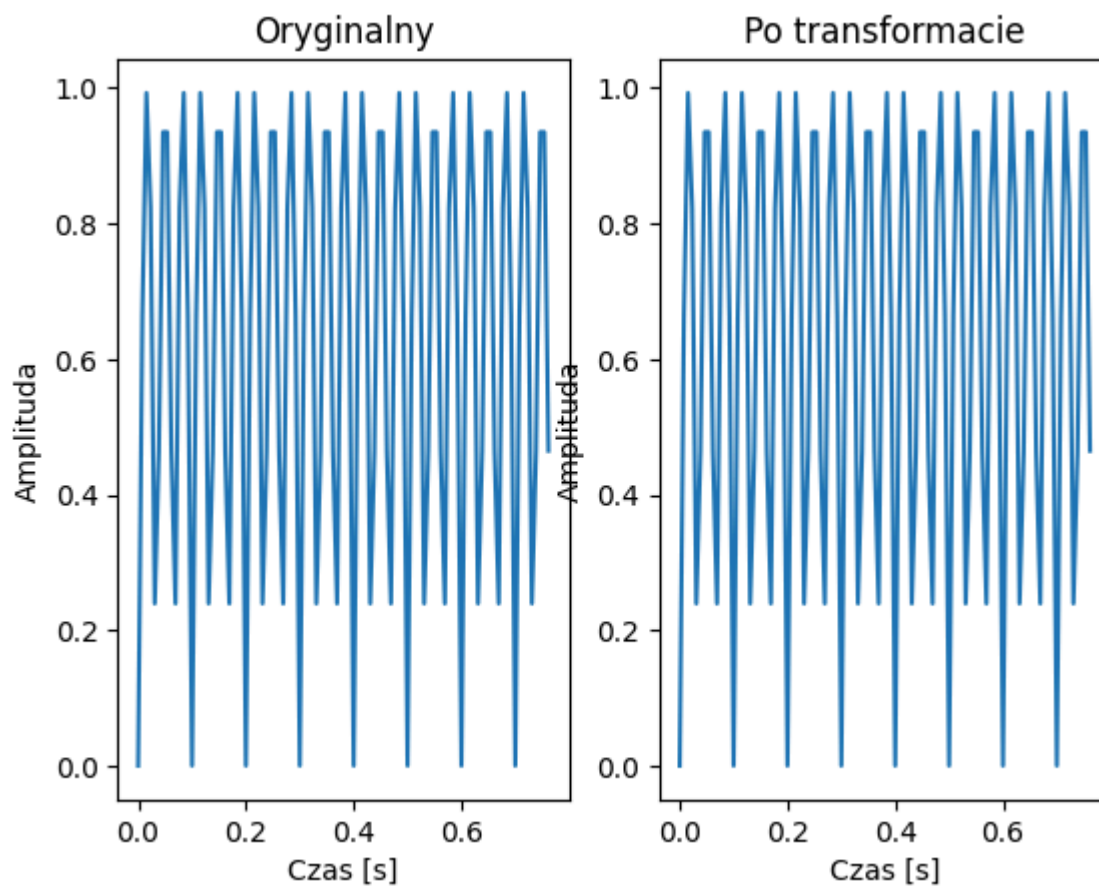
```

sinus o czestotliwosci 50 Hz

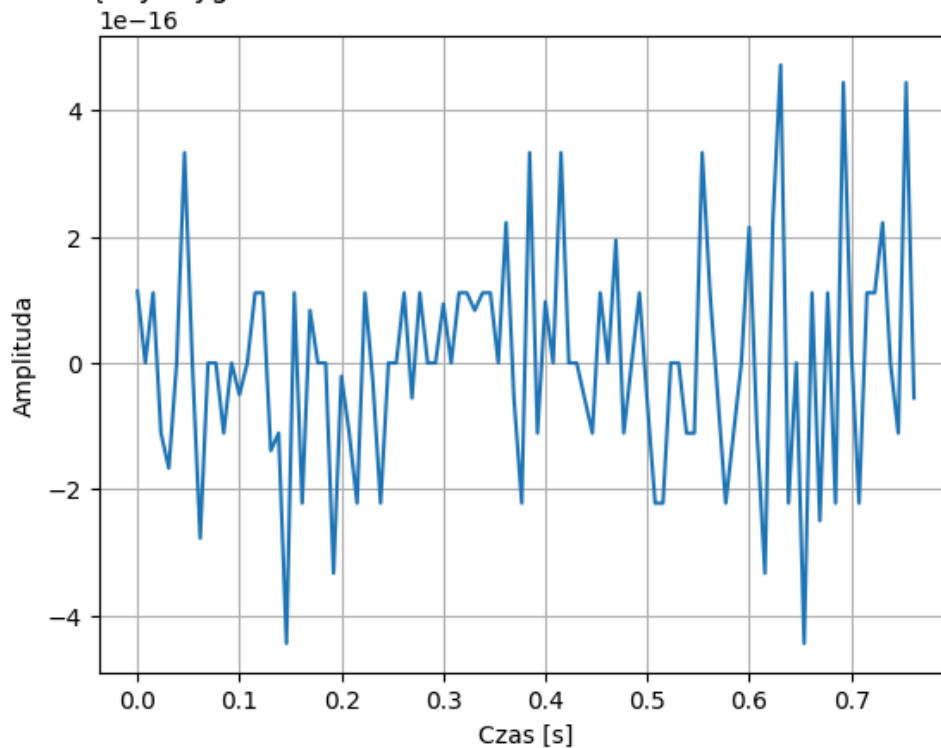


Widmo amplitudowe sygnału - sinus o czestotliwosci 50 Hz





Różnica między oryginałem a transformata odwrotna - sinus o częstotliwości 50 Hz



Wnioski

Zgodnie z twierdzeniem Nyquista (Shannona-Kotelnikowa), aby uniknąć zjawiska aliasingu (nakładania się na siebie składowych widma o różnych częstotliwościach, tzw. przecieku

widma) częstotliwość próbkowania powinna być co najmniej dwukrotnie większa niż najwyższa częstotliwość sygnału. Dlatego też w tym wypadku bezpiecznie założyć jako częstotliwość próbkowania wartość 130 Hz.

In []: