

```

import matplotlib.pyplot as plt
from functools import partial
import math
from math import sin, cos, pi, sqrt, log
from scipy.fft import fft
import time

#czestotliwosc probkowania
f_s = 8000
#przesuniecie w fazie
phi = pi/120

#funkcja pomocna dla skali decybelowej
def table_log(seq):
    return [log(el,10) for el in seq]

def generate_signal(funkcja, f_s, T):
    ''' Funkcja generujaca sygnal dla funkcji: funkcja, czestotliwosci: f_s i czasu trwania T'''
    return [funkcja(n/f_s) for n in range(int(T*f_s))]

def widmo_amplitudowe(X):
    '''Obliczamy widno amplitudowe dla ciagu X'''
    return [ sqrt(X[k].real**2 + X[k].imag**2) for k in range(len(X))]

```

sygnałem informacyjnym jest ton prosty postaci: $m(t) = \sin(2\pi f_m t)$.

Sygnały zmodulowane jednym tonem $m(t)$ opisują następujące zależności [2]:

1. Sygnał zmodulowany amplitudowo:

$$z_A(t) = [k_A \cdot m(t) + 1] \cdot \cos(2\pi f_n t).$$

2. Sygnał zmodulowany kątowno:

— modulacja fazy:

$$z_P(t) = \cos[2\pi f_n t + k_P \cdot m(t)],$$

— modulacja częstotliwości:

$$z_F(t) = \cos\left[2\pi f_n t + \frac{k_F}{f_m} \cdot m(t)\right].$$

```

#czestotliwosc dla modulacji
f_m = 200
#czas
T = 2

```

```

#ton prosty m
funkcja_m = lambda t: sin(2*pi*f_m*t)
m = generate_signal(funkcja_m, f_m, T)

def modulacja_amplitudowa(funkcja, f_n, K_a):
    '''Funkcja modulująca sygnał amplitudowo dla funkcji: funkcja, często-
    tliwosci: f_n i parametru modulacji amplitudy: K_a'''
    return lambda t: (K_a*funkcja(t)+1)*cos(2*pi*f_n*t)

def modulacja_katowa_fazy(funkcja, f_n, K_p):
    '''Funkcja modulująca sygnał katowo dla funkcji: funkcja, częstotliwo-
    sci: f_n i parametru modulacji fazy: K_p '''
    return lambda t: cos(2*pi*f_n*t + K_p*funkcja(t))

def modulacja_katowa_czestotliwosci(funkcja, f_n, K_f, f_m):
    '''Funkcja modulująca sygnał katowo dla funkcji: funkcja, częstotliwo-
    sci: f_n i parametru modulacji częstotliwosci: K_f i częstotliwosci mod-
    ulacji: f_m '''
    return lambda t: cos(2*pi*f_n*t + K_f*funkcja(t)/f_m)

```

Wygenerować sygnały zmodulowane $z_A(t)$, $z_P(t)$ oraz $z_F(t)$ dla następujących przypadków (f_n oraz f_m należy dobrać tak aby był spełniony warunek $f_n \gg f_m$):

- | | | | |
|----------------------------|----------------------|----------------------|-----------------|
| — modulacja amplitudy | : a) $1 > k_A > 0$; | b) $12 > k_A > 2$; | c) $k_A > 20$ |
| — modulacja fazy | : a) $k_P < 1$; | b) $\pi > k_P > 0$; | c) $k_P > 2\pi$ |
| — modulacja częstotliwości | : a) $k_F < 1$; | b) $\pi > k_F > 0$; | c) $k_F > 2\pi$ |

```

#czestotliwosc probkowania
f_n = 1000

#modulacja amplitudy
#a
K_a = 0.4
f_mod_amp_a = modulacja_amplitudowa(funkcja_m, f_n, K_a)
mod_amp_a = generate_signal(f_mod_amp_a, f_m, T)

#b
K_a = 11
f_mod_amp_b = modulacja_amplitudowa(funkcja_m, f_n, K_a)
mod_amp_b = generate_signal(f_mod_amp_b, f_m, T)

#c
K_a = 21
f_mod_amp_c = modulacja_amplitudowa(funkcja_m, f_n, K_a)
mod_amp_c = generate_signal(f_mod_amp_c, f_m, T)

```

```

#modulacja fazy
#a
K_p = 0.3
f_mod_kat_a = modulacja_katowa_fazy(funkcja_m, f_n, K_p)
mod_kat_a = generate_signal(f_mod_kat_a, f_m, T)

#b
K_p = 2
f_mod_kat_b = modulacja_katowa_fazy(funkcja_m, f_n, K_p)
mod_kat_b = generate_signal(f_mod_kat_b, f_m, T)

#c
K_p = 8
f_mod_kat_c = modulacja_katowa_fazy(funkcja_m, f_n, K_p)
mod_kat_c = generate_signal(f_mod_kat_c, f_m, T)

#modulacja czestotliwosci
#a
K_f = 0.3
f_mod_czest_a = modulacja_katowa_czestotliwosci(funkcja_m, f_n, K_f, f_m)
mod_czest_a = generate_signal(f_mod_czest_a, f_m, T)

#b
K_f = 2
f_mod_czest_b = modulacja_katowa_czestotliwosci(funkcja_m, f_n, K_f, f_m)
mod_czest_b = generate_signal(f_mod_czest_b, f_m, T)

#c
K_f = 8
f_mod_czest_c = modulacja_katowa_czestotliwosci(funkcja_m, f_n, K_f, f_m)
mod_czest_c = generate_signal(f_mod_czest_c, f_m, T)

```

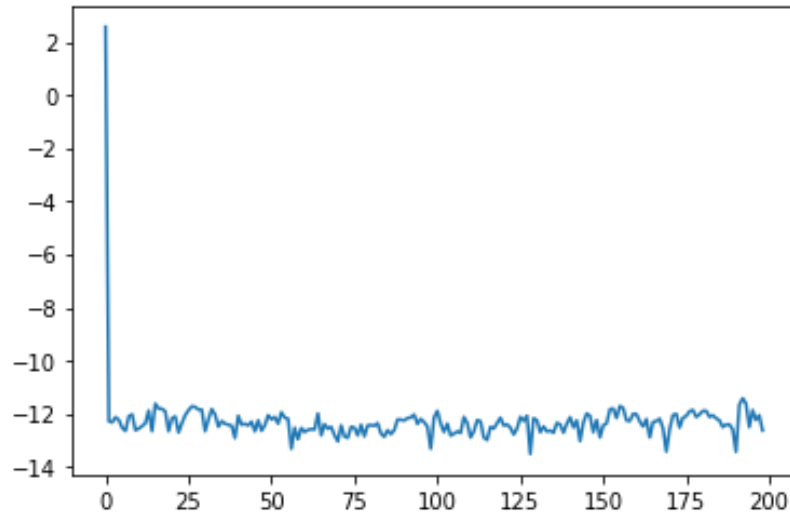
2. Narysować widma amplitudowe w skali decybelowej sygnałów zmodulowanych $z_A(t)$, $z_P(t)$ oraz $z_F(t)$. Należy tak dobrać skalę osi częstotliwości (liniową lub logarytmiczną) aby jak najwięcej prążków widma było widocznych na wykresie.

```

#dlugosc sygnalu
N = len(mod_czest_a)

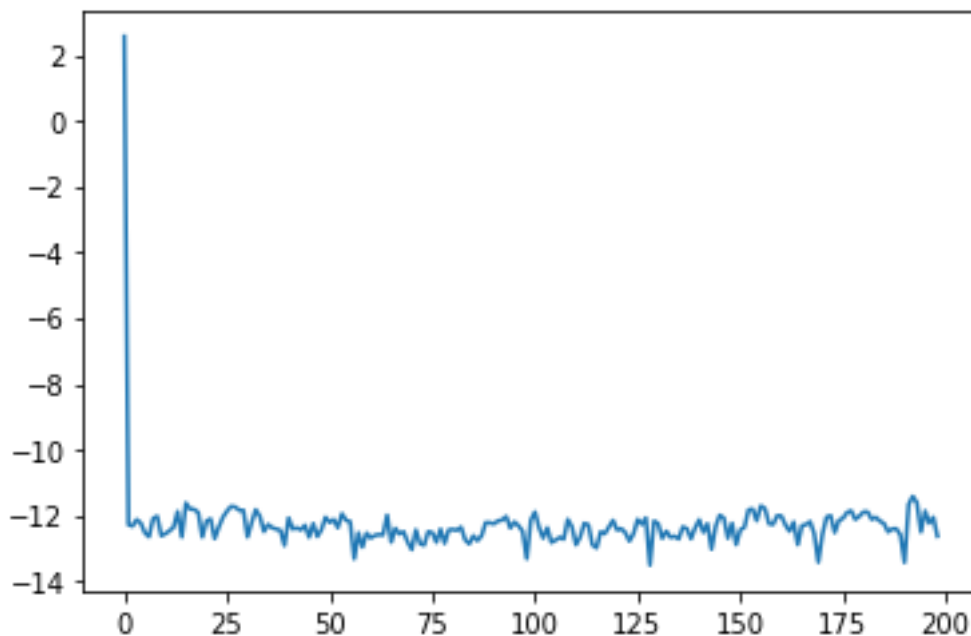
```

```
#widmo amplitudowe
Widmo_mod_amp_a = widmo_amplitudowe(fft(mod_amp_a)[:int(N/2 - 1)])
plt.plot(range(int(N/2 - 1)), table_log(Widmo_mod_amp_a))
plt.title("Widmo amplitudowe sygnału zmodulowanego amplitudowo K_a=0.4  
[skala logarytmiczna]", y=-.21)
```



Widmo amplitudowe sygnału zmodulowanego amplitudowo $K_a=0.4$ [skala logarytmiczna]

```
Widmo_mod_amp_b = widmo_amplitudowe(fft(mod_amp_b)[:int(N/2 - 1)])
plt.plot(range(int(N/2 - 1)), table_log(Widmo_mod_amp_a))
plt.title("Widmo amplitudowe sygnału zmodulowanego amplitudowo K_a=11",  
y=-.21)
```

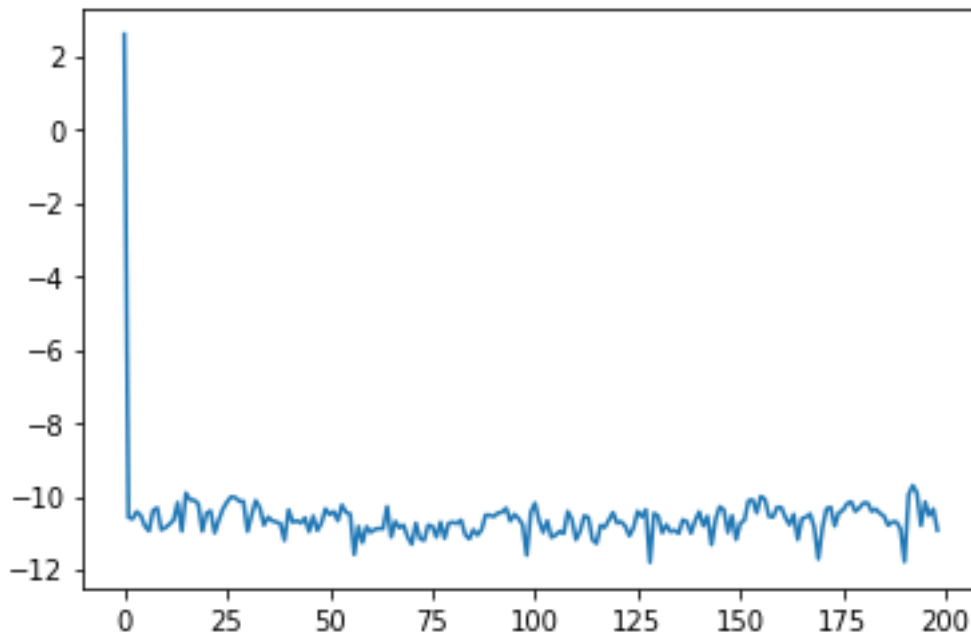


Widmo amplitudowe sygnału zmodulowanego amplitudowo $K_a=11$

```

Widmo_mod_amp_c = widmo_amplitudowe(fft(mod_amp_c)[:int(N/2 - 1)])
plt.plot(range(int(N/2 - 1)), table_log(Widmo_mod_amp_c))
plt.title("Widmo amplitudowe sygnału zmodulowanego amplitudowo  $K_a=21$ ",
y=-.21)

```

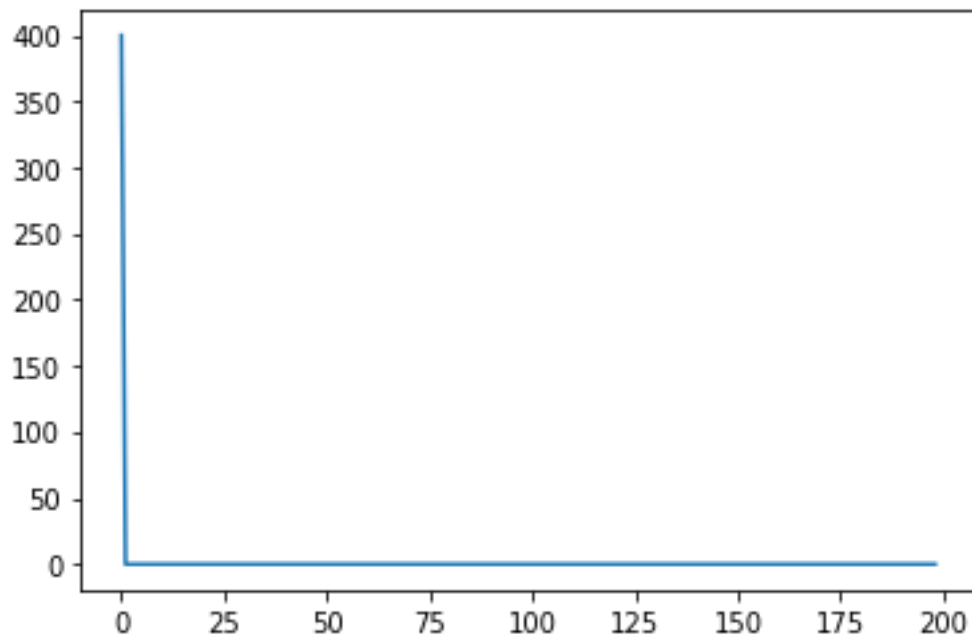


Widmo amplitudowe sygnału zmodulowanego amplitudowo $K_a=21$

```

Widmo_mod_kat_a = widmo_amplitudowe(fft(mod_kat_a)[:int(N/2 - 1)])
plt.plot(range(int(N/2 - 1)), Widmo_mod_kat_a)
plt.title("Widmo amplitudowe sygnału zmodulowanego fazowo  $K_a=0.3$  ", y=
-.21)

```

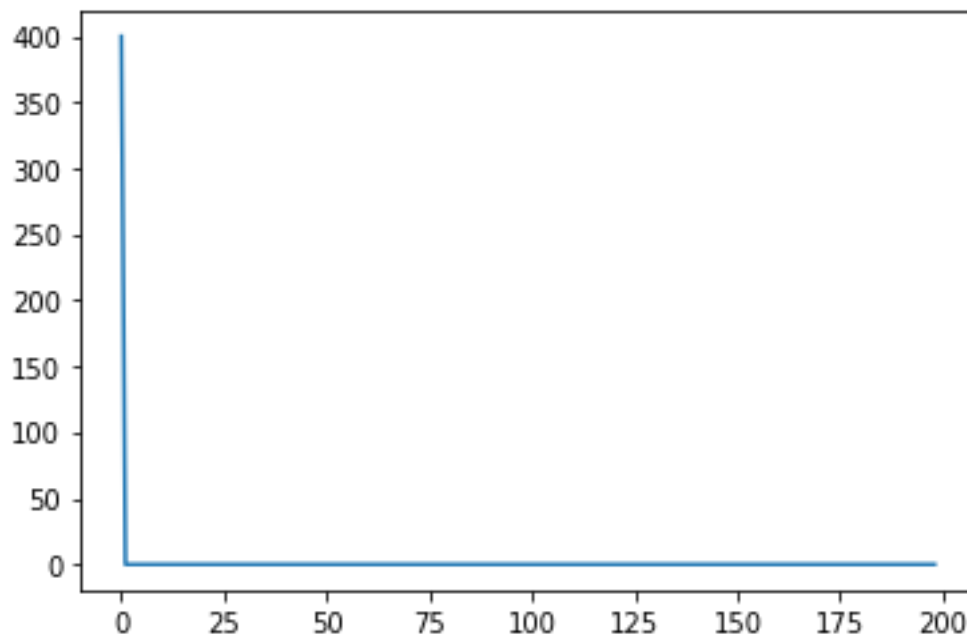


Widmo amplitudowe sygnału zmodulowanego fazowo $K_a=0.3$

```

Widmo_mod_kat_b = widmo_amplitudowe(fft(mod_kat_b)[:int(N/2 - 1)])
plt.plot(range(int(N/2 - 1)), Widmo_mod_kat_b)
plt.title("Widmo amplitudowe sygnału zmodulowanego fazowo K_a=2", y=-
.21)

```

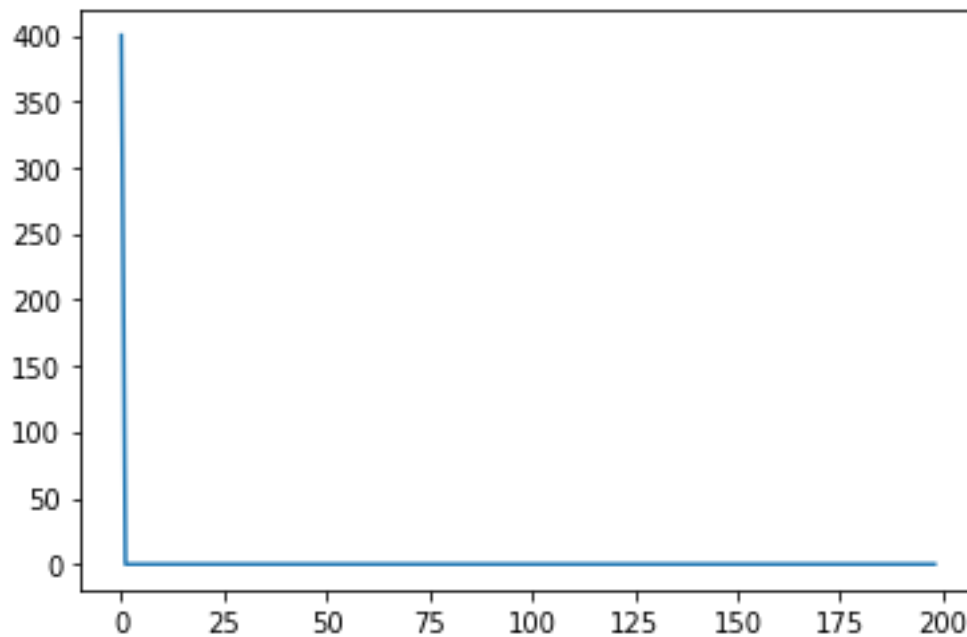


Widmo amplitudowe sygnału zmodulowanego fazowo K_a=2

```

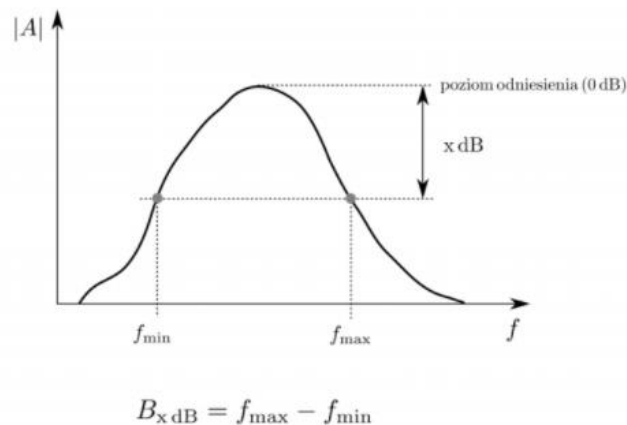
Widmo_mod_kat_c = widmo_amplitudowe(fft(mod_kat_c)[:int(N/2 - 1)])
plt.plot(range(int(N/2 - 1)), Widmo_mod_kat_c)
plt.title("Widmo amplitudowe sygnału zmodulowanego fazowo K_a=8", y=-
.21)

```



Widmo amplitudowe sygnału zmodulowanego fazowo K_a=8

3. Oszacować szerokość pasma B_{3dB} , B_{6dB} oraz B_{12dB} sygnału zmodulowanego w sposób przedstawiony na rysunku 1.



Rysunek 1. Schemat wyznaczania szerokości pasma sygnału [3].

```
def szerokosc_pasma(czestotliowosc, amplituda, x):
    ''' Oblicza szerokosc pasma Bxdb dla sygnału (czestotliowosc, amplituda) unimodalnego. Funkcja zakłada, że istnieje dokładnie jeden punkt przecięcia'''
    if x == 3:
        poziom = 0.707
    elif x == 6:
        poziom = 0.5
    elif x == 12:
        poziom = 0.25
    else:
        raise Exception

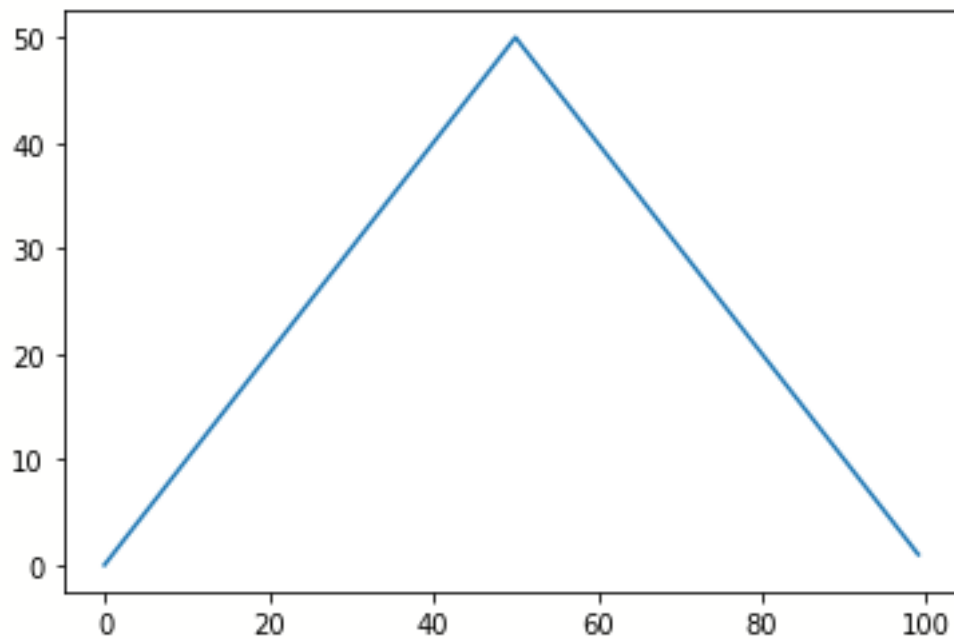
    #skalowanie amplitudy
    max_amplitudy = max(amplituda)
    #indeks maksymalnego elementu
    index = amplituda.index(max_amplitudy)
    #przeskalowana amplituda i przesunieta o poziom
    scaled_amplituda = [abs(amp/max_amplitudy - poziom) for amp in amplituda]

    #lewa czesc wykresu - przed maksimum
    scaled_left = scaled_amplituda[:index]
    #prawa czesc wykresu - przed maksimum
    scaled_right = scaled_amplituda[index+1:]
    #f_min
    scaled_left_min = min(scaled_left)
    f_min = scaled_left.index(scaled_left_min)
    #f_max
    scaled_right_min = min(scaled_right)
    f_max = scaled_right.index(scaled_right_min) + index

    return f_max - f_min
```

```
xs = range(100)
ys = [-abs(x-50) + 50 for x in xs]
```

```
plt.plot(xs,ys)
```



```
szerokosc_pasma(xs, ys, 3)
#wynik dla 3dB obliczony przez program
29
szerokosc_pasma(xs, ys, 6)
#wynik dla 6dB obliczony przez program
49
szerokosc_pasma(xs, ys, 12)
#wynik dla 12dB obliczony przez program
74
```