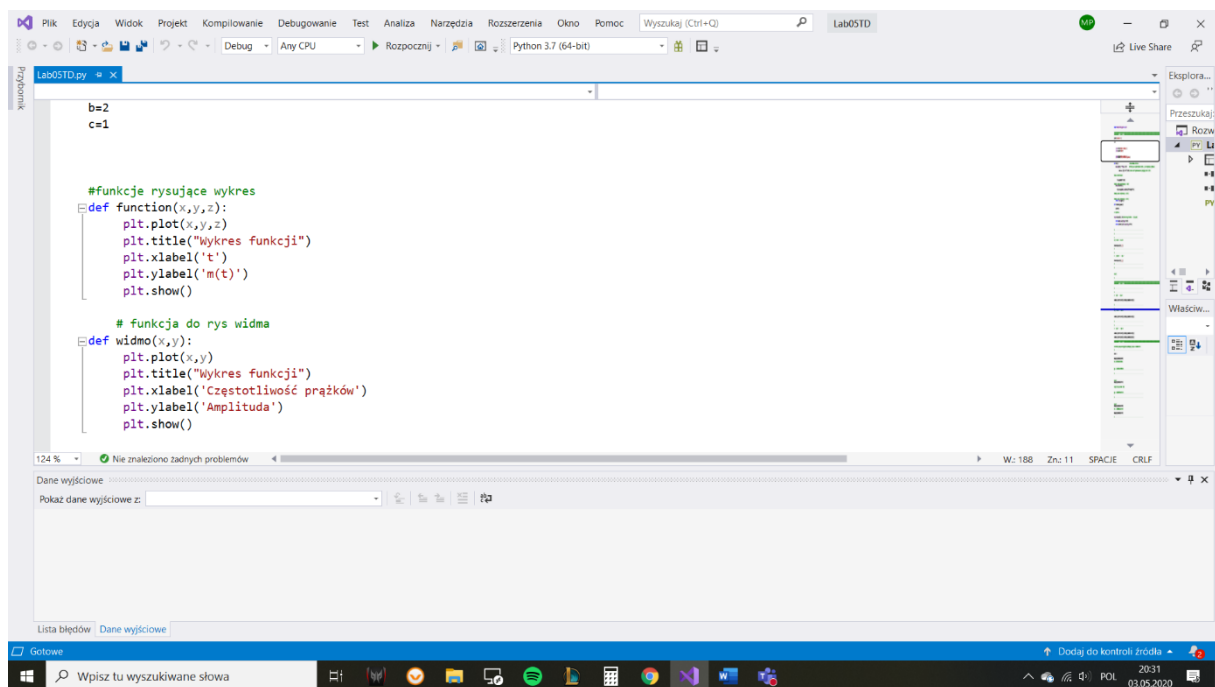


Sprawozdanie Lab05

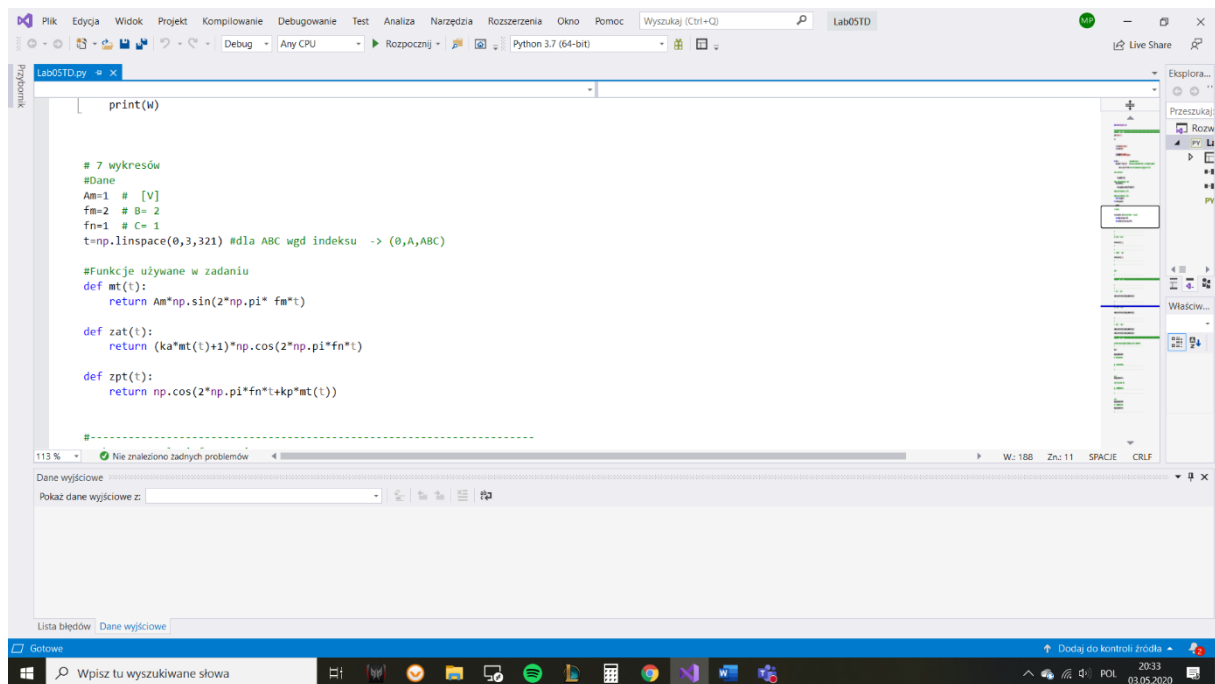
Mateusz Proc 20C N1 nr 45123

Zadanie 1/.

Funkcje rysujące wykresy, w tym druga do rysują wykres widma amplitudowego w zadaniu 2



Funkcje $Z_a(t)$ $Z_p(t)$ i $m(t)$ oraz dane do nich.



The screenshot shows a JupyterLab notebook with the following code:

```
print(w)

# 7 wykresów
#Dane
Am=1 # [V]
fm=2 # B= 2
fn=1 # C= 1
t=np.linspace(0,3,321) #dla ABC wgd indeksu -> (0,A,ABC)

#Funkcje używane w zadaniu
def mt(t):
    return Am*np.sin(2*np.pi*fm*t)

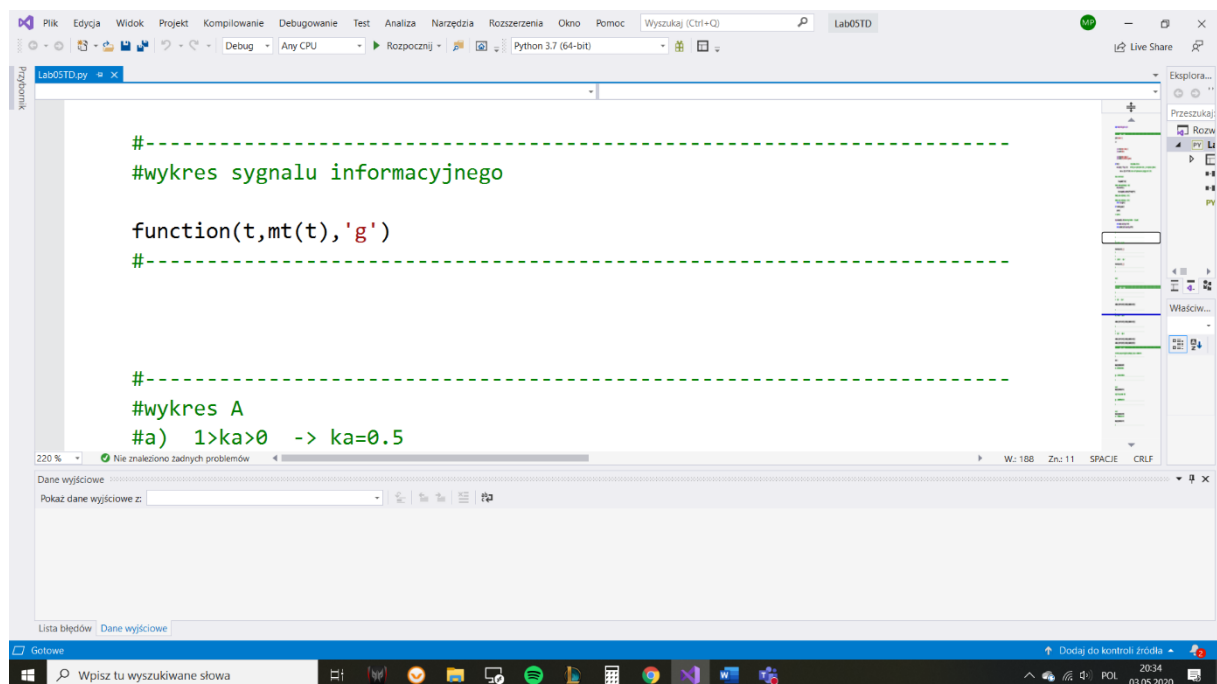
def zat(t):
    return (ka*mt(t)+1)*np.cos(2*np.pi*fn*t)

def zpt(t):
    return np.cos(2*np.pi*fn*t+kp*mt(t))

#-----
```

The notebook interface includes a menu bar, a toolbar, and a sidebar with file explorer and search tools. The status bar at the bottom shows the file name 'Lab05TD.py' and the Python version 'Python 3.7 (64-bit)'.

Wykres Sygnału informacyjnego



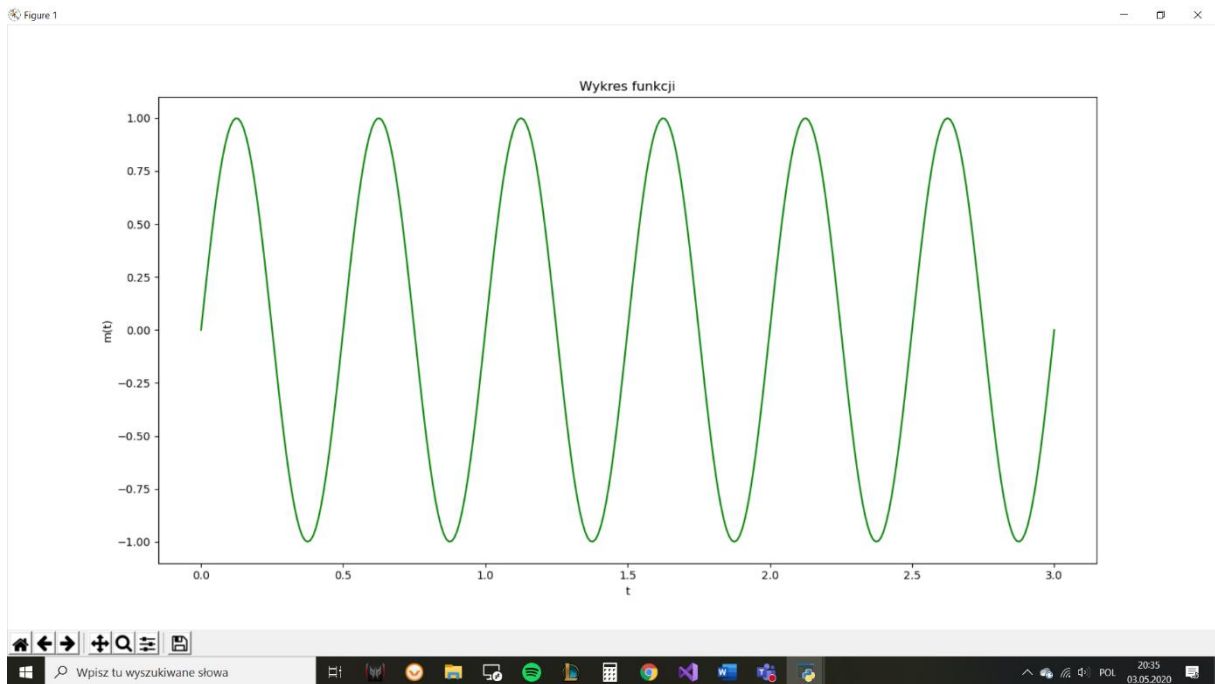
The screenshot shows a JupyterLab notebook with the following code:

```
#-----
#wykres sygnału informacyjnego

function(t,mt(t),'g')
#-----

#-----
#wykres A
#a) 1>ka>0 -> ka=0.5
```

The notebook interface is similar to the previous one, showing the same menu bar, toolbar, and sidebar. The status bar at the bottom shows the file name 'Lab05TD.py' and the Python version 'Python 3.7 (64-bit)'.



Zmodulowany sygnał dla $Z_a(t)$ i $Z_p(t)$ dla $k_a=0.5$ oraz $k_p=1$

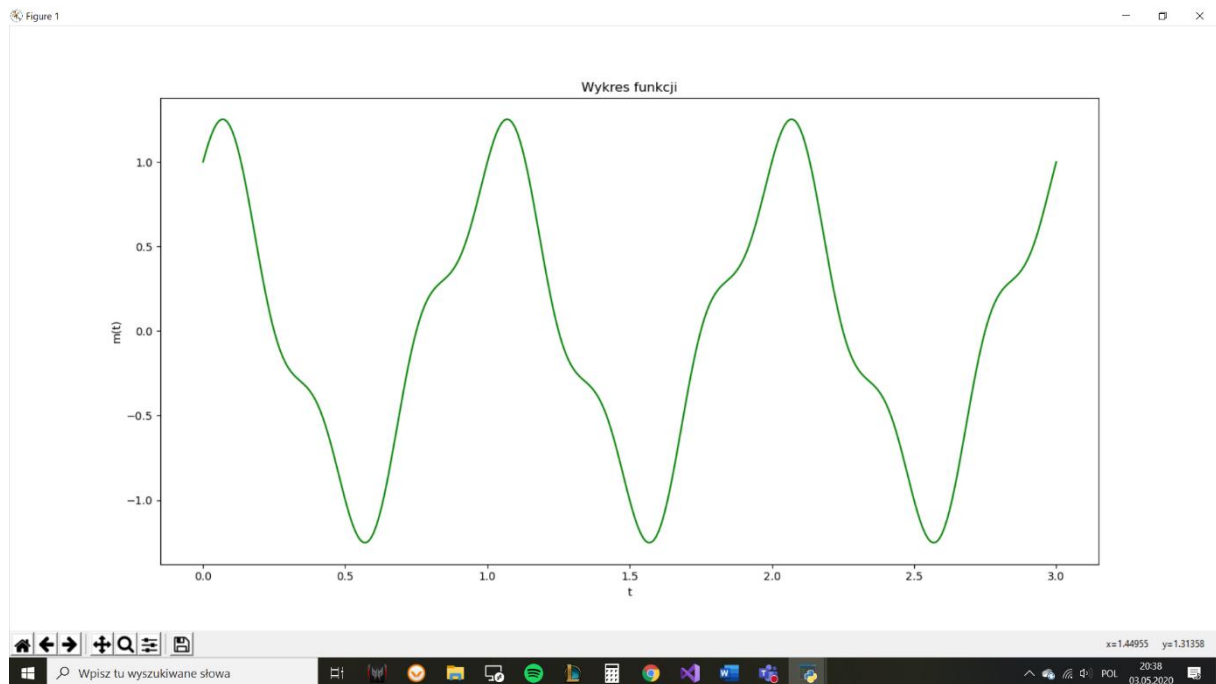
The screenshot shows a Python IDE with the following code in the editor:

```
#-----
#wykres A
#a) 1>ka>0 -> ka=0.5
#    kp<2 -> kp=1
ka=0.5
kp=1

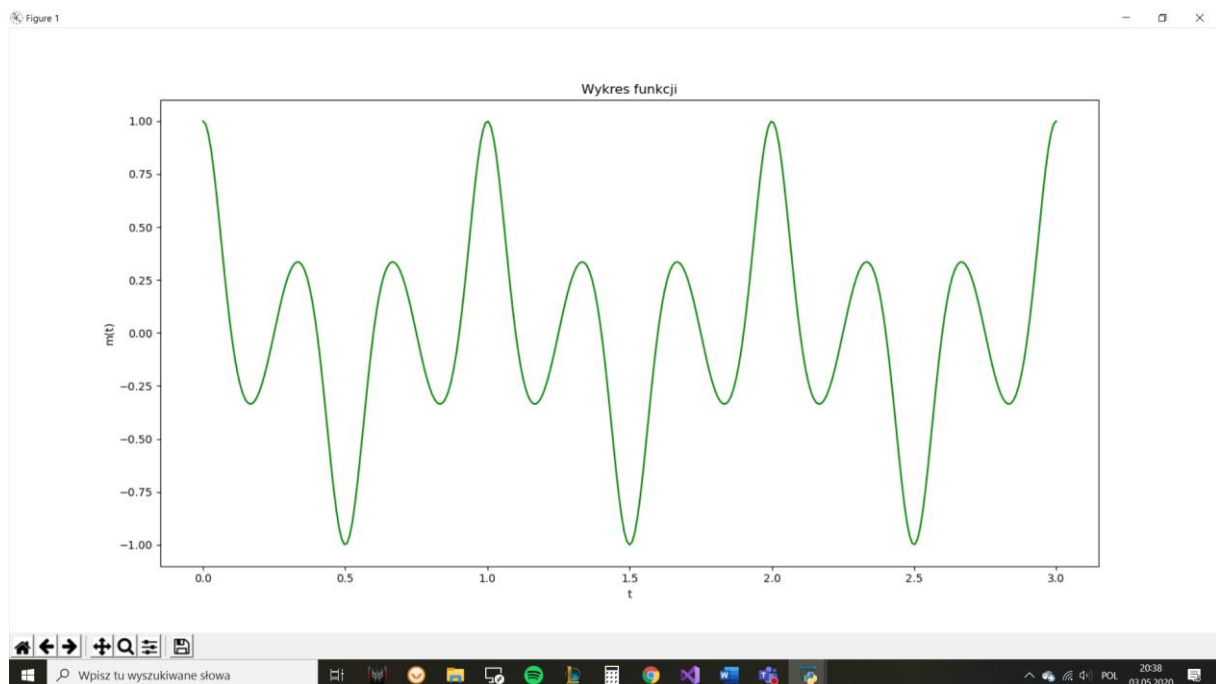
#za(t)
function(t,zat(t),'g')
#zp(t)
function(t,zpt(t),'g')
#-----
```

The IDE interface includes a menu bar (Plik, Edycja, Widok, Projekt, Kompilowanie, Debugowanie, Test, Analiza, Narzędzia, Rozszerzenia, Okno, Pomoc), a toolbar, and a console window at the bottom. The console window shows the output of the script, including the values of ka and k_p .

Wykres $Z_a(t)$ dla $k_a=0.5$ oraz $k_p=1$

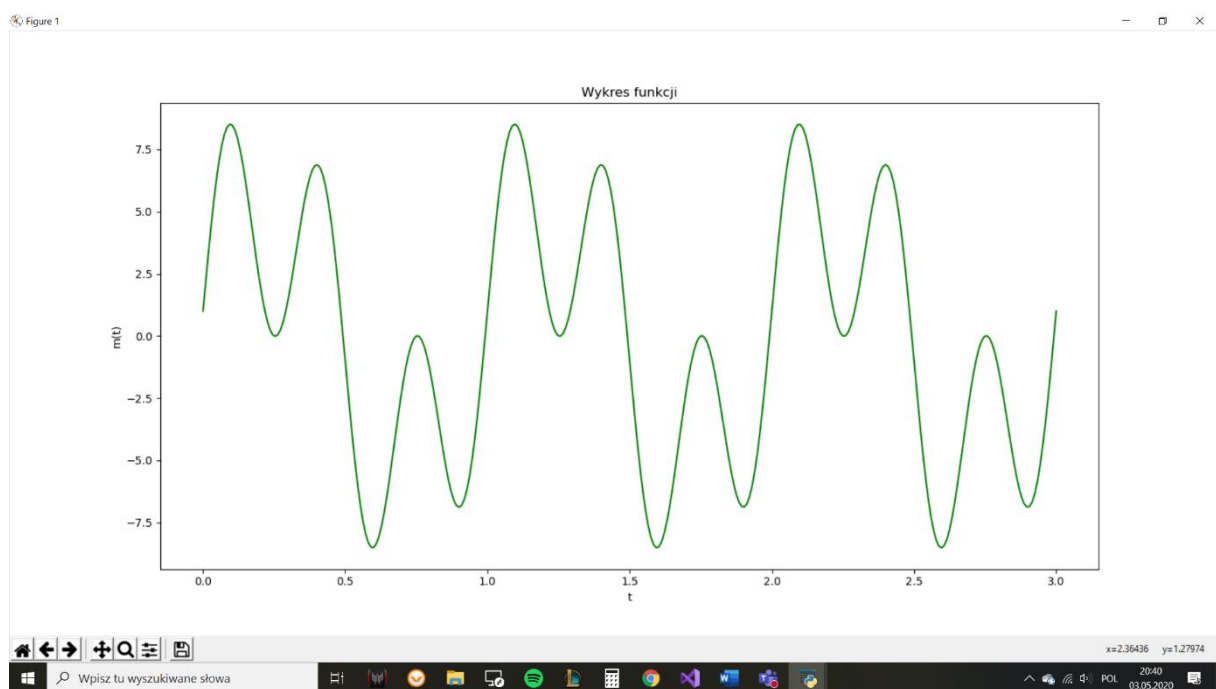


Wykres $Z_p(t)$ dla $k_a=0.5$ oraz $k_p=1$

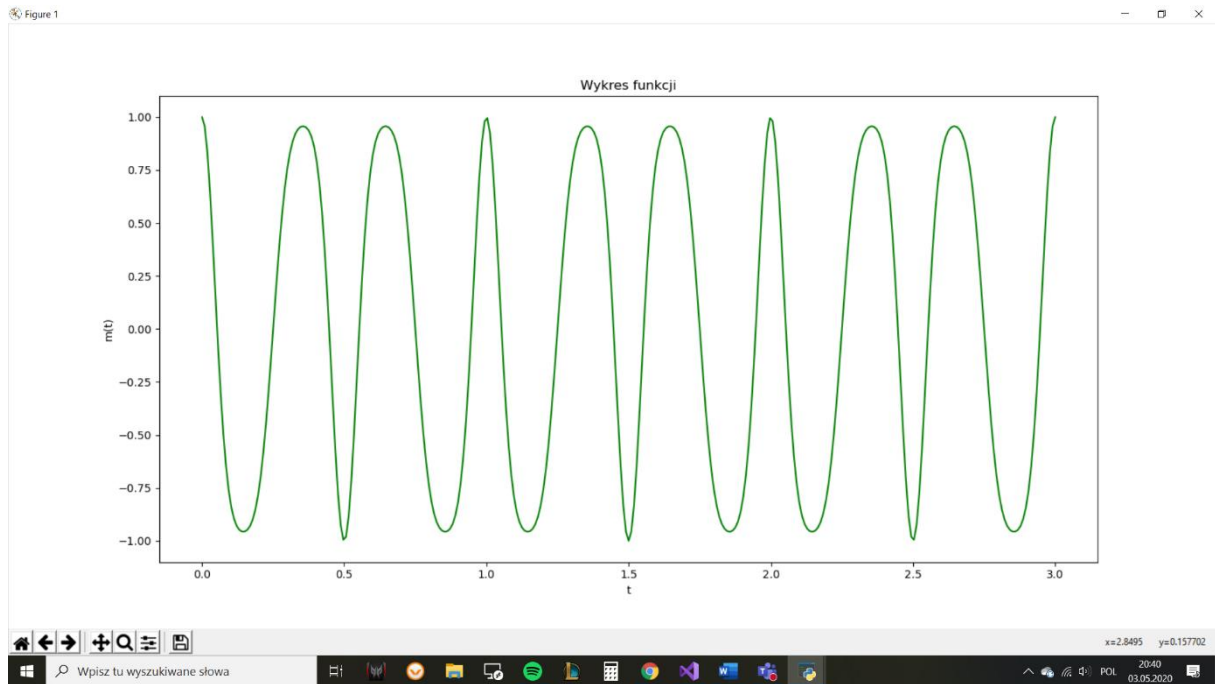


Zmodulowany sygnał dla $Z_a(t)$ i $Z_p(t)$ dla $k_a=10$ oraz $k_p=2$

Wykres $Z_a(t)$ dla $k_a=10$ oraz $k_p=2$



Wykres $Z_p(t)$ dla $k_a=10$ oraz $k_p=2$

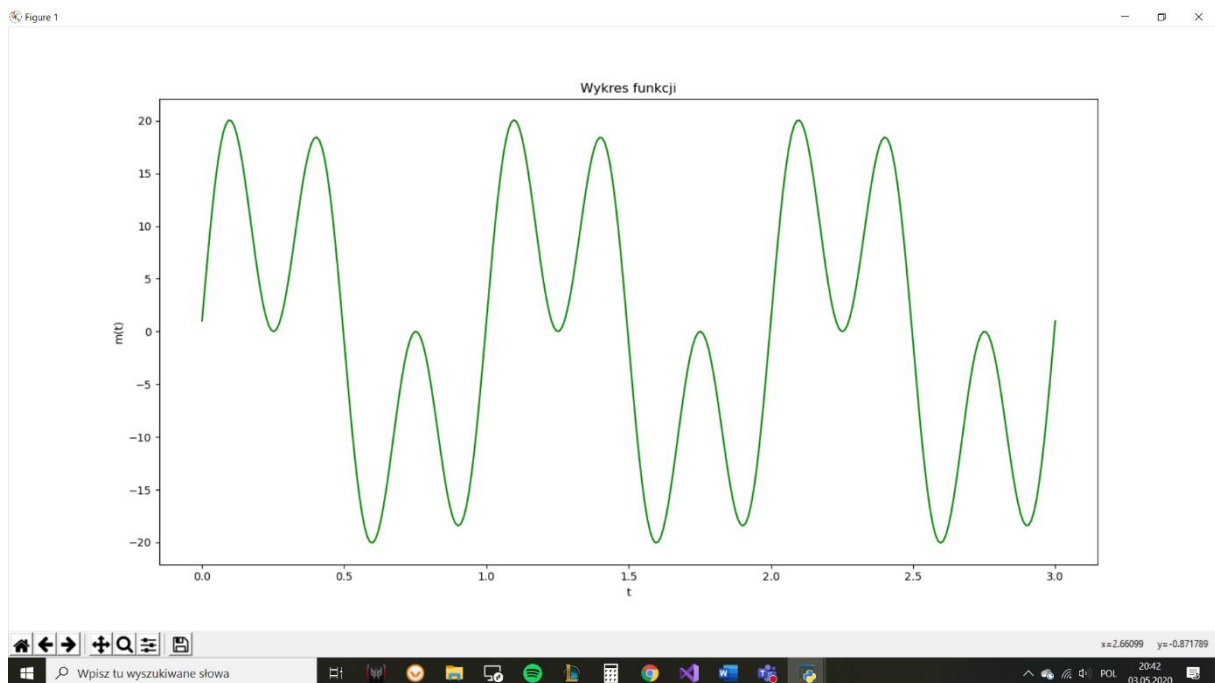


Zmodulowany sygnał dla $Z_a(t)$ i $Z_p(t)$ dla $k_a=25$ oraz $k_p=35$

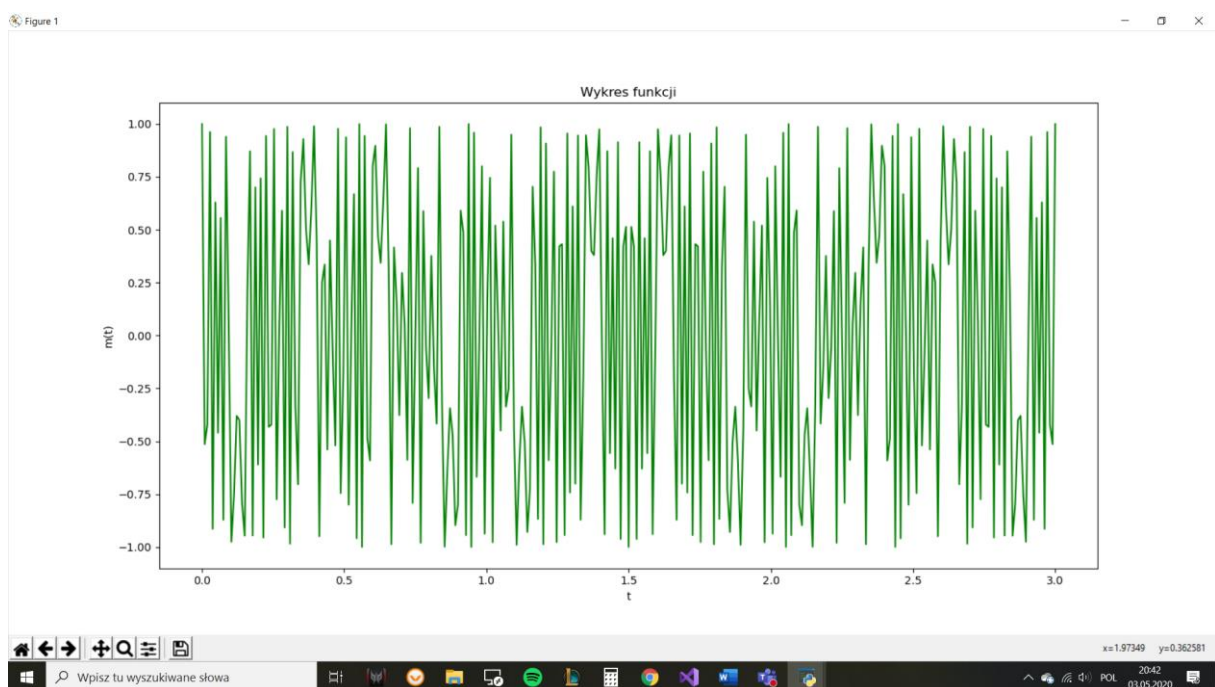
```
#
#wykres C
#c)   $k_a > 23 \rightarrow k_a = 25$ 
#     $k_p > 32 \rightarrow k_p = 35$ 
ka=25
kp=35

#za(t)
function(t,zat(t),'g')
#zp(t)
function(t,zpt(t),'g')
#-----
```

Wykres $Z_a(t)$ dla $k_a=25$ oraz $k_p=35$

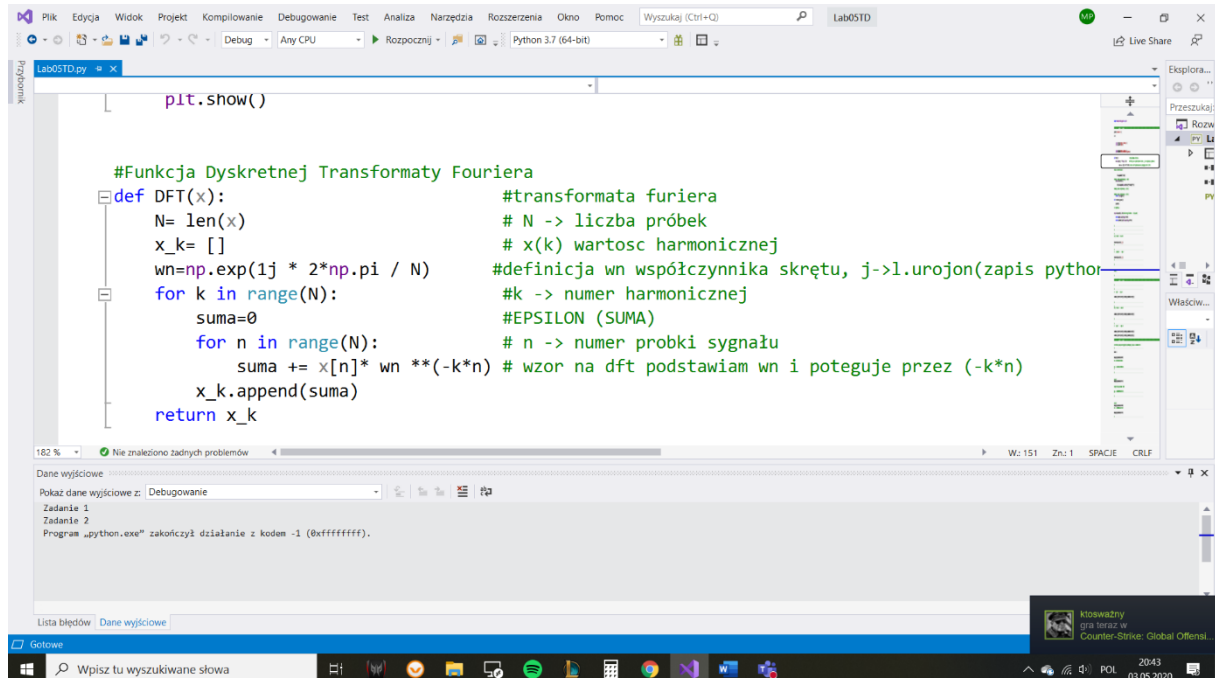


Wykres $Z_p(t)$ dla $k_a=25$ oraz $k_p=35$



Zadanie 2/.

Funkcja DFT użyta z poprzednich laboratoriów

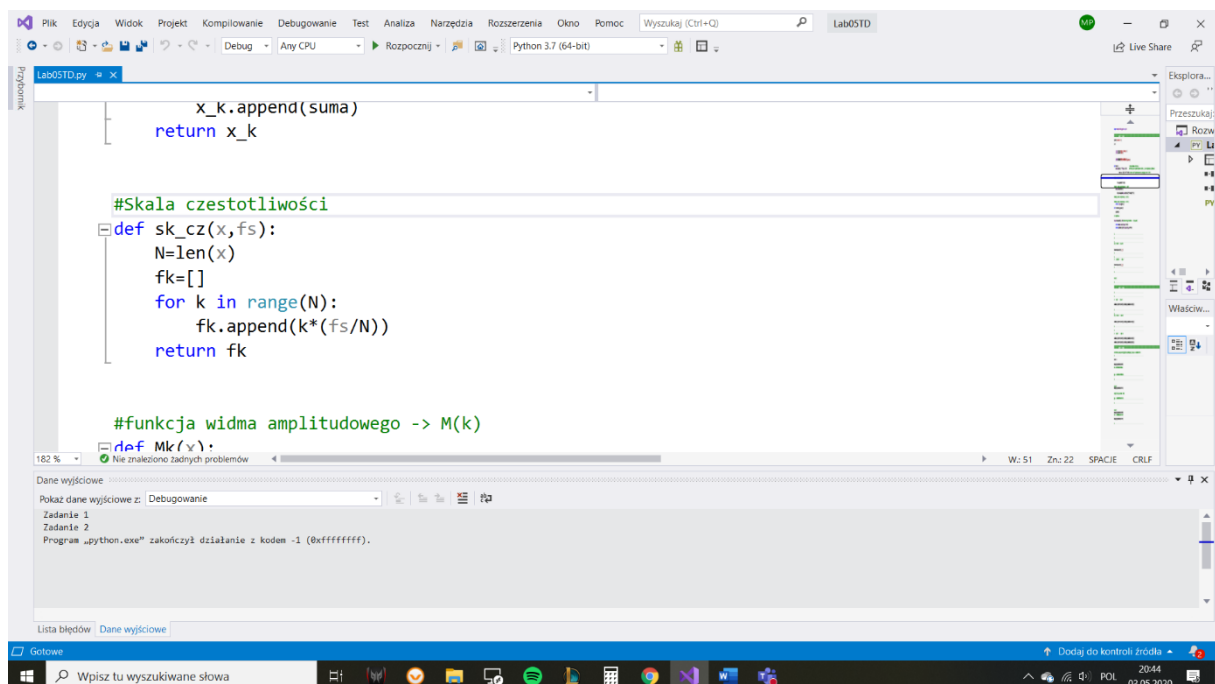


```
plt.show()

#Funkcja Dyskretnej Transformaty Fouriera
def DFT(x):
    N= len(x)
    x_k= []
    wn=np.exp(1j * 2*np.pi / N)
    for k in range(N):
        suma=0
        for n in range(N):
            suma += x[n]* wn **(-k*n)
        x_k.append(suma)
    return x_k
```

The screenshot shows a Python IDE with a file named 'Lab05TD.py'. The code defines a function 'DFT(x)' that calculates the Discrete Fourier Transform. It includes comments in Polish explaining the variables: 'N' is the number of samples, 'x(k)' is the harmonic value, 'wn' is the complex exponential coefficient, 'k' is the harmonic number, and 'n' is the signal sample number. The formula for the DFT is implemented as $x_k = \sum_{n=0}^{N-1} x[n] \cdot w_n^{-k \cdot n}$. The IDE interface includes a menu bar, a toolbar, a file explorer on the right, and a console at the bottom showing a successful execution of the program.

Funkcja skali częstotliwości użyta z poprzednich laboratoriów



```
x_k.append(suma)
return x_k

#Skala czestotliwosci
def sk_cz(x,fs):
    N=len(x)
    fk=[]
    for k in range(N):
        fk.append(k*(fs/N))
    return fk

#funkcja widma amplitudowego -> M(k)
def Mk(x):
```

The screenshot shows the same Python IDE with additional code. It defines a function 'sk_cz(x, fs)' for frequency scaling, which calculates the frequency values 'fk' for each sample index 'k' using the formula $f_k = k \cdot (f_s / N)$. It also shows the beginning of a function 'Mk(x)' for the amplitude spectrum. The IDE interface is consistent with the previous screenshot, showing the same menu bar, toolbar, file explorer, and console.

Funkcja $M(k)$ i $M(k)'$ użyta z poprzednich laboratoriów

```
Lab05TD.py
    tk.append(K*(Ts/N))
    return fk

#funkcja widma amplitudowego -> M(k)
def Mk(x):
    Re=np.real(DFT(x))
    Im=np.imag(DFT(x))
    Mk_tab=[]
    for i in range(0, len(Re)):
        Mk_tab.append(np.sqrt(Re[i]**2+Im[i]**2))
    return Mk_tab

#funkcja skali decybelowej -> M'(k)
def Mk_prim(x): # M'(k)
    return 10*np.log10(x)
```

150 % Nie znaleziono żadnych problemów W: 69 Zn: 1 SPACIE CRLF

Dane wyjściowe
Pokaż dane wyjściowe z: Debugowanie
Zadanie 1
Zadanie 2
Program „python.exe” zakończył działanie z kodem -1 (0xffffffff).

Lista błędów Dane wyjściowe

Widma amplitudowe dla funkcji $z_a(t)$ i $z_p(t)$ dla $k_a=0.5$ i $k_p=1$

```
Lab05TD.py
#-----
#wykres A
#a) 1>ka>0 -> ka=0.5
# kp<2 -> kp=1
ka=0.5
kp=1

#za(t)
widmo(sk_cz(DFT(zat(t)),100),Mk_prim(Mk(zat(t))))

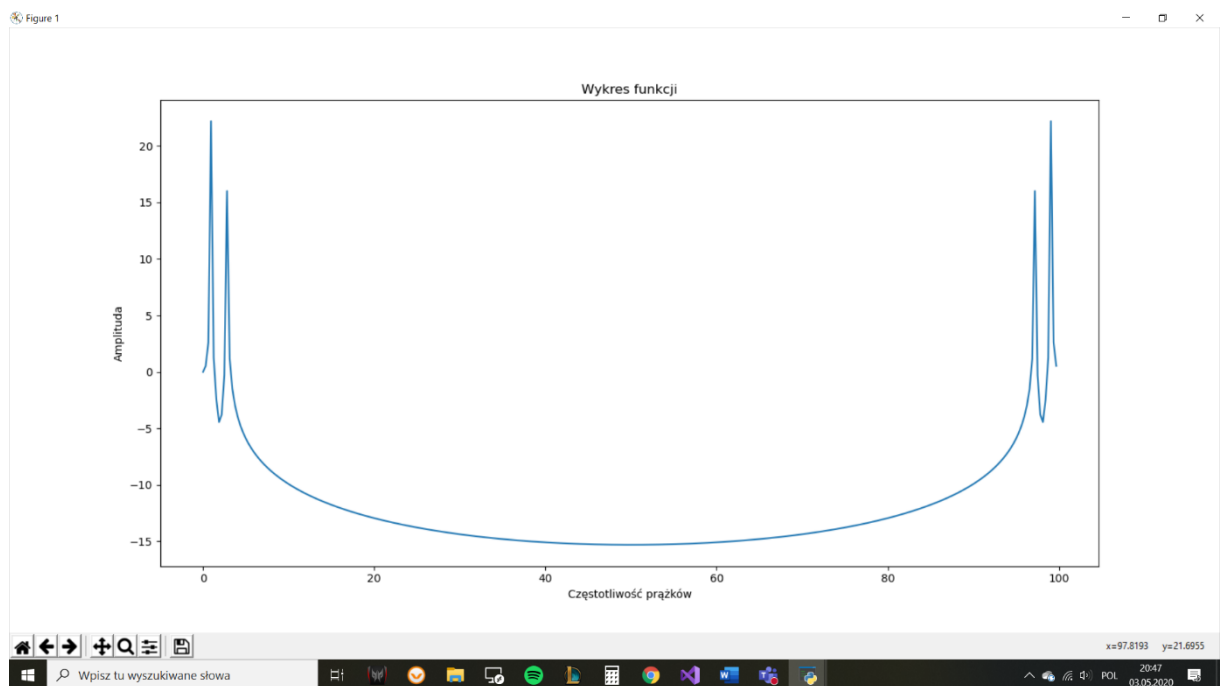
#zp(t)
widmo(sk_cz(DFT(zpt(t)),100),Mk_prim(Mk(zpt(t))))
#-----
```

150 % Nie znaleziono żadnych problemów W: 168 Zn: 1 SPACIE CRLF

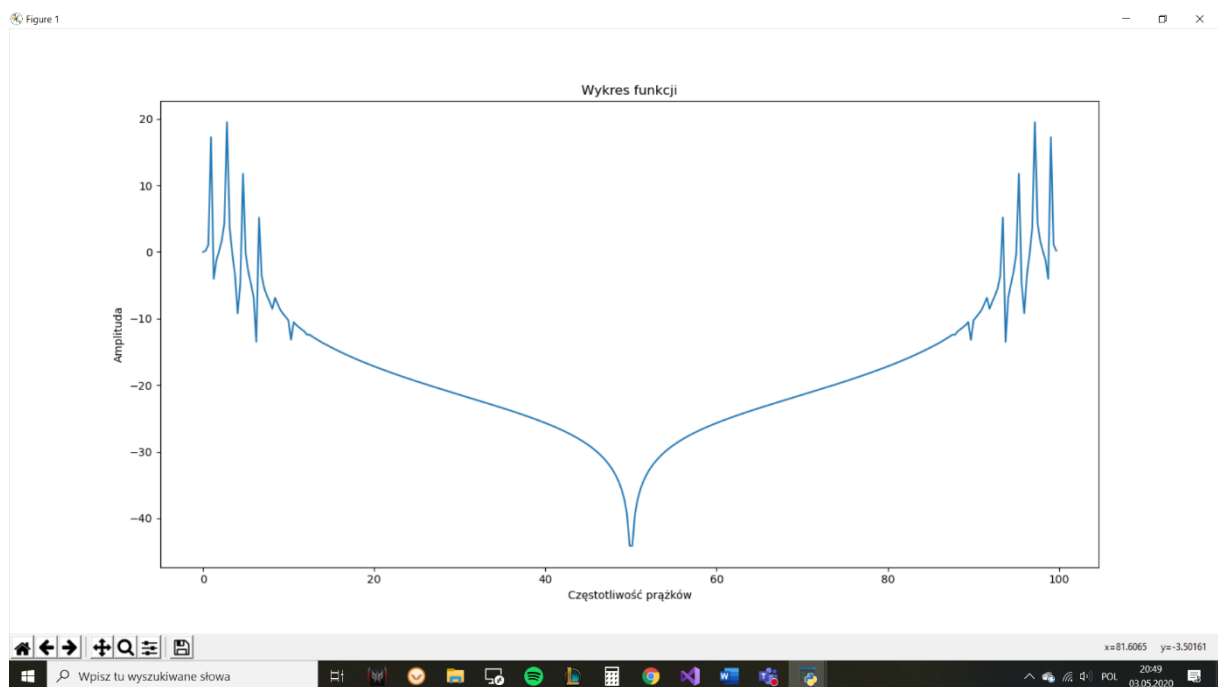
Dane wyjściowe
Pokaż dane wyjściowe z: Debugowanie
Zadanie 1
Zadanie 2
Program „python.exe” zakończył działanie z kodem -1 (0xffffffff).

Lista błędów Dane wyjściowe

$Z_a(t)$ dla $k_a=0.5$ i $k_p=1$



$Z_p(t)$ dla $k_a=0.5$ i $k_p=1$



Widma amplitudowe dla funkcji $z_a(t)$ i $z_p(t)$ dla $k_a=10$ i $k_p=2$

```
Lab05TD.py
#-----
#wykres B
#b) 12>ka>2 -> ka=10
#    pi>kp>0 -> kp=2
ka=10
kp=2

#za(t)
widmo(sk_cz(DFT(zat(t)),100),Mk_prim(Mk(zat(t))))

#zp(t)
widmo(sk_cz(DFT(zpt(t)),100),Mk_prim(Mk(zpt(t))))
#-----
```

Wyrażenie kontrolne 1

Nazwa	Wartość	Typ
Dodaj element do oglądania		

Dane wyjściowe

Pokaz dane wyjściowe z: Debugowanie

Zadanie 1

Zadanie 2

Automatyczne Lokalne Wyrażenie kontrolne 1

Stos wywołań Punkty przenawia Ustawienia wyjątków Okno polecenia Okno bezpośrednie Dane wyjściowe Lista błędów

Gotowe

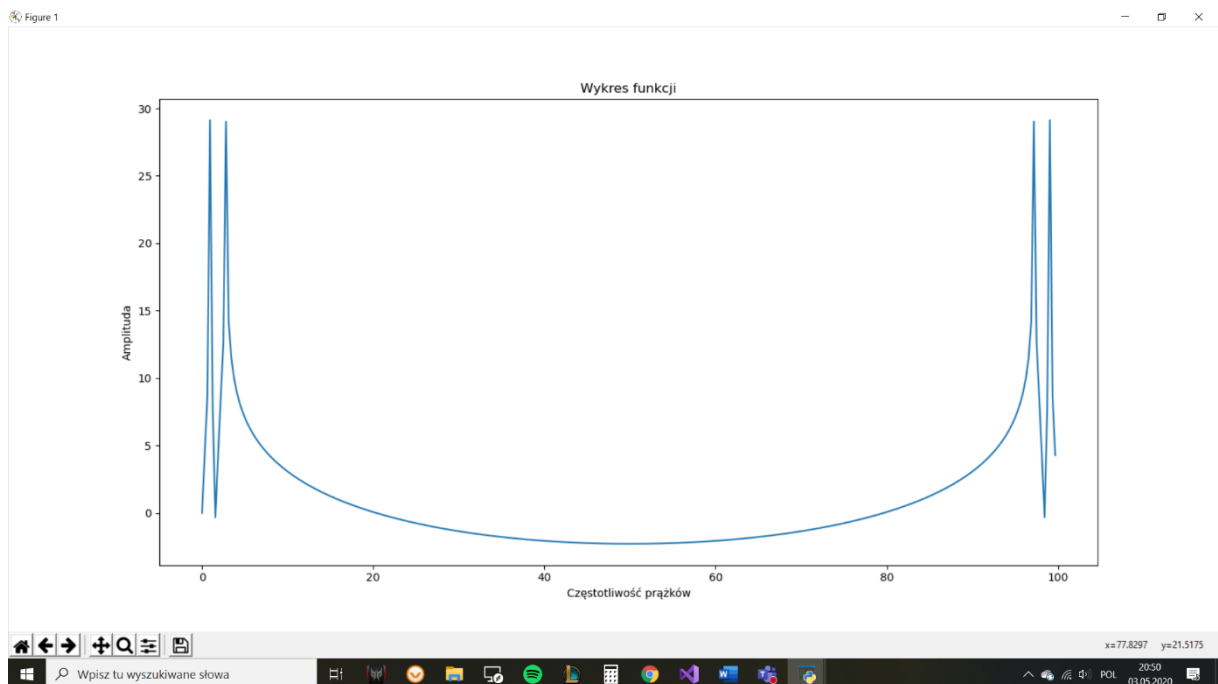
Wpisz tu wyszukiwane słowa

Wyszukaj (Ctrl+Q)

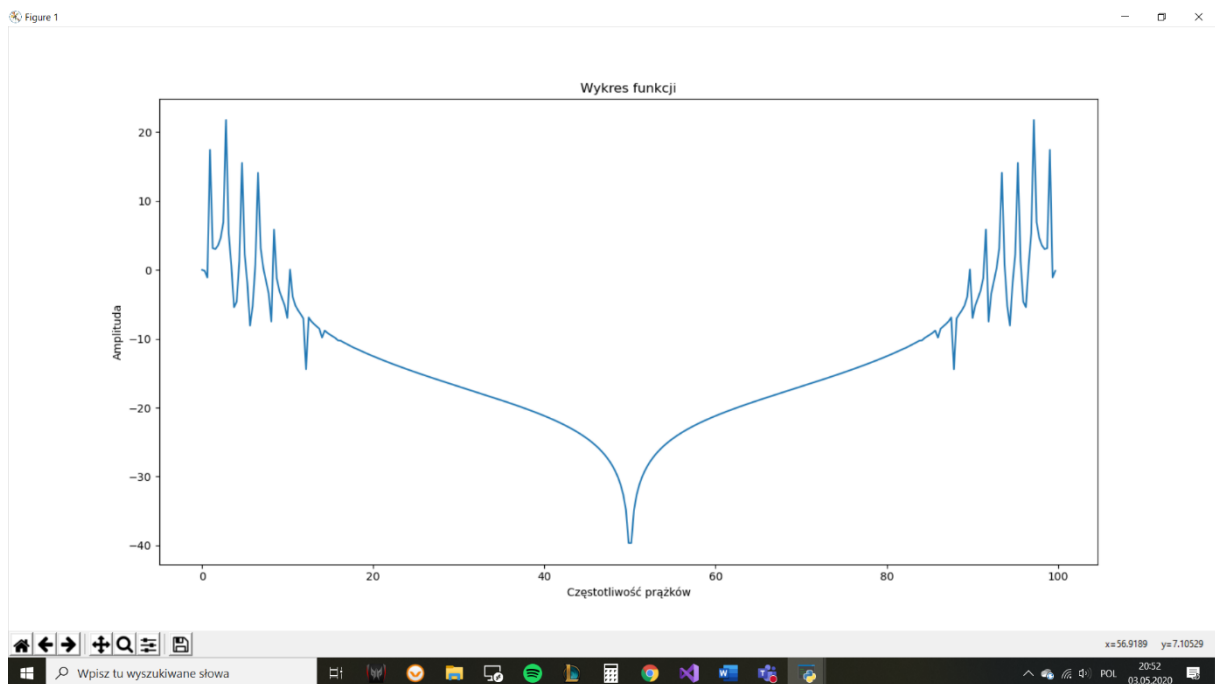
Python 3.7 (64-bit)

20:49 03.05.2020

$Z_a(t)$ dla $k_a=10$ i $k_p=2$



$Z_p(t)$ dla $k_a=10$ i $k_p=2$



Widma amplitudowe dla funkcji $z_a(t)$ i $z_p(t)$ dla $k_a=25$ i $k_p=35$

```

#-----
#wykres C
#c) ka>23 -> ka=25
# kp>32 -> kp=35
ka=25
kp=35

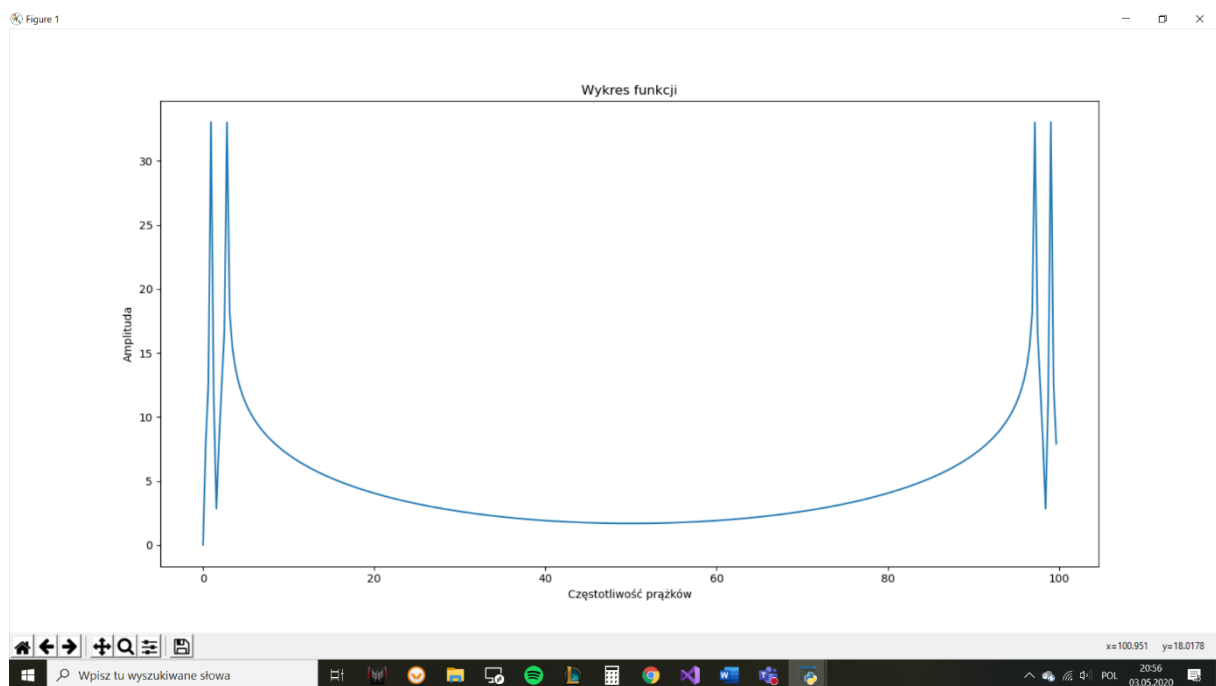
#za(t)
widmo(sk_cz(DFT(zat(t)),100),Mk_prim(Mk(zat(t))))

#zp(t)
widmo(sk_cz(DFT(zpt(t)),100),Mk_prim(Mk(zpt(t))))
#-----

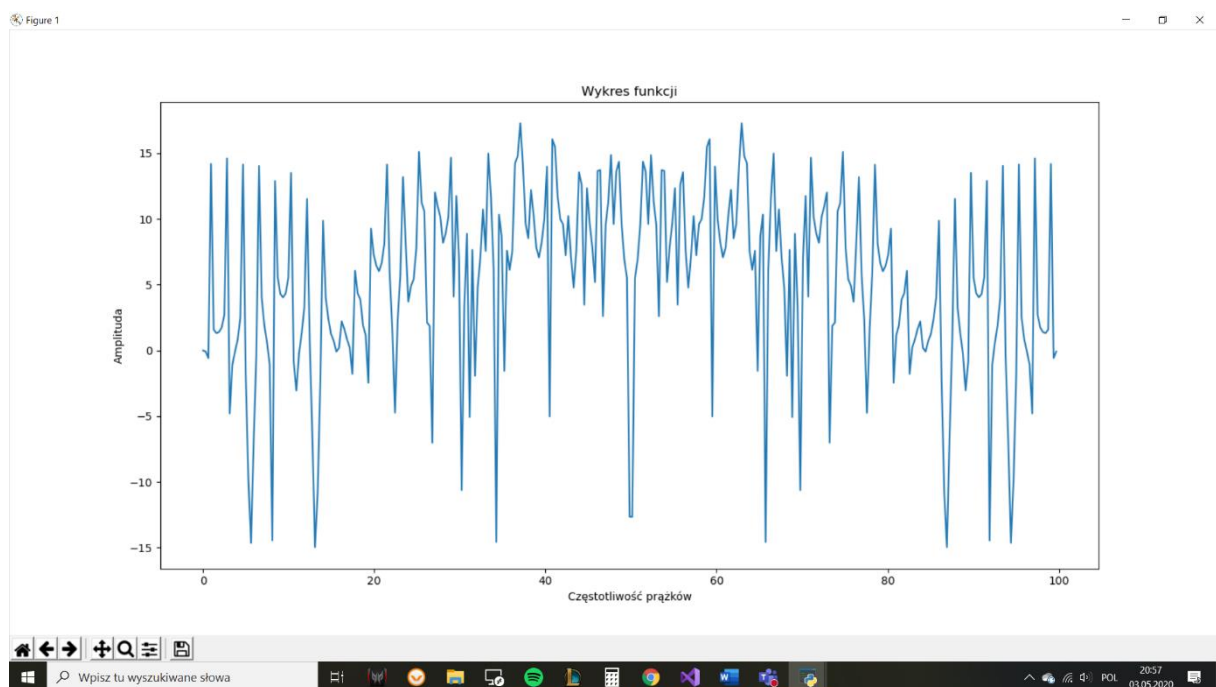
```

The image shows a screenshot of a Python IDE (LabOSTD) with a code editor and a console. The code defines parameters $k_a=25$ and $k_p=35$, and calculates the amplitude spectra for functions $z_a(t)$ and $z_p(t)$ using the Fast Fourier Transform (FFT) and a magnitude spectrum function $widmo$. The console shows the execution of the code, with the output of the $widmo$ function displayed.

$Z_a(t)$ dla $k_a=25$ i $k_p=35$



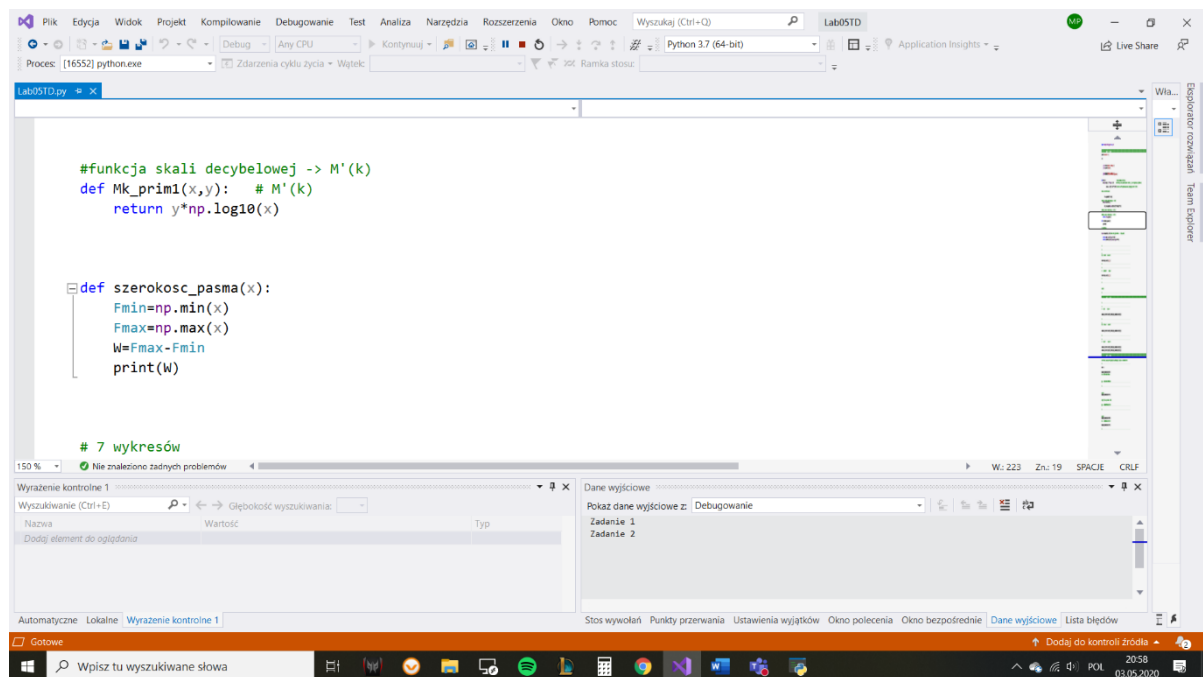
$Z_p(t)$ dla $k_a=25$ i $k_p=35$



Zadanie 3/.

Funkcja $M(k)'$ przerobiona na potrzebę zmiany danych we wzorze

Funkcja szerokości pasma



```
#funkcja skali decybelowej -> M'(k)
def Mk_prim1(x,y): # M'(k)
    return y*np.log10(x)

def szerokosc_pasma(x):
    Fmin=np.min(x)
    Fmax=np.max(x)
    W=Fmax-Fmin
    print(W)

# 7 wykresów
```

The screenshot shows a Python IDE with the following code in the editor:

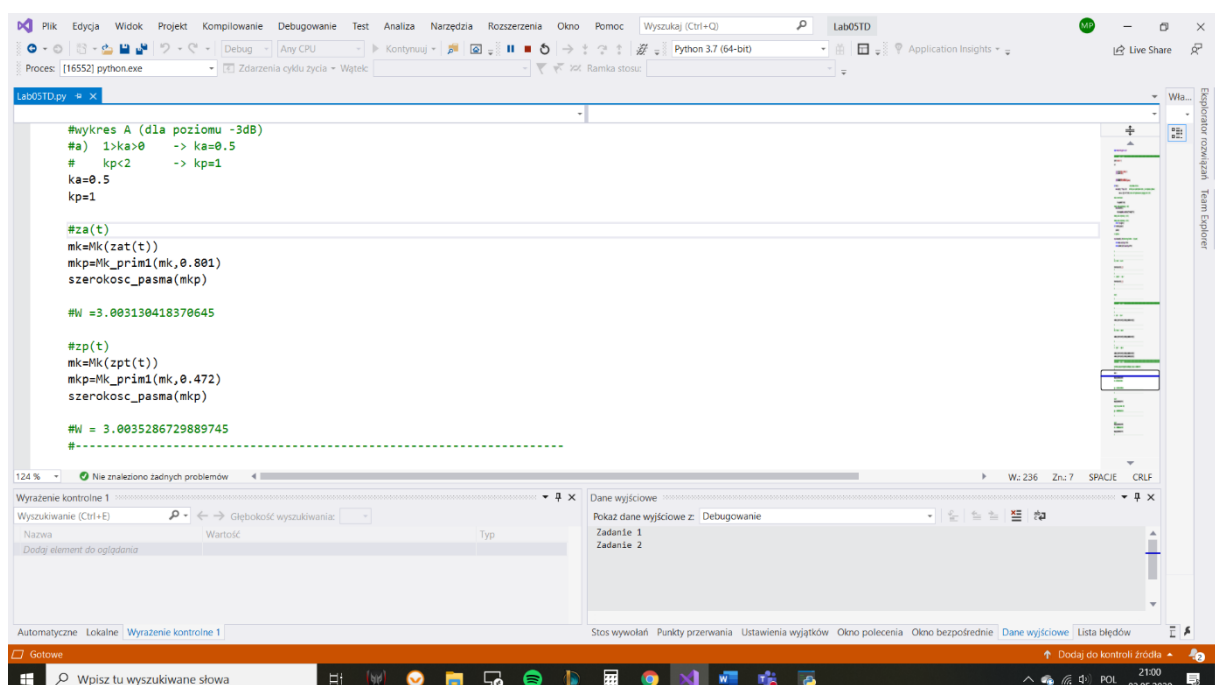
```
#funkcja skali decybelowej -> M'(k)
def Mk_prim1(x,y): # M'(k)
    return y*np.log10(x)

def szerokosc_pasma(x):
    Fmin=np.min(x)
    Fmax=np.max(x)
    W=Fmax-Fmin
    print(W)

# 7 wykresów
```

The IDE interface includes a menu bar, a toolbar, and a sidebar with a file explorer. The status bar at the bottom shows the file name 'Lab05TD.py' and the current line and column numbers.

Szerokość pasma sygnałów zmodulowanych $z_a(t)$ i $z_p(t)$ dla $k_a=0.5$ i $k_p=1$



```
#wykres A (dla poziomu -3dB)
#a) 1>ka>0 -> ka=0.5
#     kp<2 -> kp=1
ka=0.5
kp=1

#za(t)
mk=Mk(zat(t))
mkp=Mk_prim1(mk,0.801)
szerokosc_pasma(mkp)

#W =3.003130418370645

#zp(t)
mk=Mk(zpt(t))
mkp=Mk_prim1(mk,0.472)
szerokosc_pasma(mkp)

#W = 3.0035286729889745
#-----
```

The screenshot shows a Python IDE with the following code in the editor:

```
#wykres A (dla poziomu -3dB)
#a) 1>ka>0 -> ka=0.5
#     kp<2 -> kp=1
ka=0.5
kp=1

#za(t)
mk=Mk(zat(t))
mkp=Mk_prim1(mk,0.801)
szerokosc_pasma(mkp)

#W =3.003130418370645

#zp(t)
mk=Mk(zpt(t))
mkp=Mk_prim1(mk,0.472)
szerokosc_pasma(mkp)

#W = 3.0035286729889745
#-----
```

The IDE interface includes a menu bar, a toolbar, and a sidebar with a file explorer. The status bar at the bottom shows the file name 'Lab05TD.py' and the current line and column numbers.

Szerokość pasma sygnałów zmodulowanych $z_a(t)$ i $z_p(t)$ dla $k_a=10$ i $k_p=2$

The screenshot shows a Python IDE with the following code in the editor:

```
#wykres B (dla poziomu -3dB)
#b) 12>ka>2 -> ka=10
#    pi>kp>0 -> kp=2
ka=10
kp=2

#za(t)
mk=Mk(zat(t))
mkp=Mk_prim1(mk,0.955)
szerokosc_pasma(mkp)

#W = 3.000806199780511

#zp(t) (dla poziomu -3dB)
mk=Mk(zpt(t))
mkp=Mk_prim1(mk,0.490)
szerokosc_pasma(mkp)

#W = 3.0086990966293325
#-----
```

The output window displays the following results:

```
Pokaz dane wyjściowe z: Debugowanie
3.0005206129889745
3.000806199780511
3.0086990966293325
3.000868464757507
3.001748278747142
Wątek 'MainThread' (0x1) zakończył działanie z kodem 0 (0x0).
```

Szerokość pasma sygnałów zmodulowanych $z_a(t)$ i $z_p(t)$ dla $k_a=25$ i $k_p=35$

The screenshot shows a Python IDE with the following code in the editor:

```
#wykres C
#c) ka>23 -> ka=25
#    kp>32 -> kp=35
ka=25
kp=35

#za(t)
mk=Mk(zat(t))
mkp=Mk_prim1(mk,0.908)
szerokosc_pasma(mkp)

#W = 3.00068464757507

#zp(t)
mk=Mk(zpt(t))
mkp=Mk_prim1(mk,0.931)
szerokosc_pasma(mkp)

#W = 3.001748278747142
#-----
```

The output window displays the following results:

```
Pokaz dane wyjściowe z: Debugowanie
3.0005206129889745
3.000806199780511
3.0086990966293325
3.00068464757507
3.001748278747142
Wątek 'MainThread' (0x1) zakończył działanie z kodem 0 (0x0).
```

Wyniki dla zadania 3

```
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python37_64\python.exe
Zadanie 1
Zadanie 2
Zadanie 3
3.003130418370645
3.0035286729889745
3.000806199780511
3.0086990966293325
3.000068464757507
3.001748278747142
Press any key to continue . . .
```

