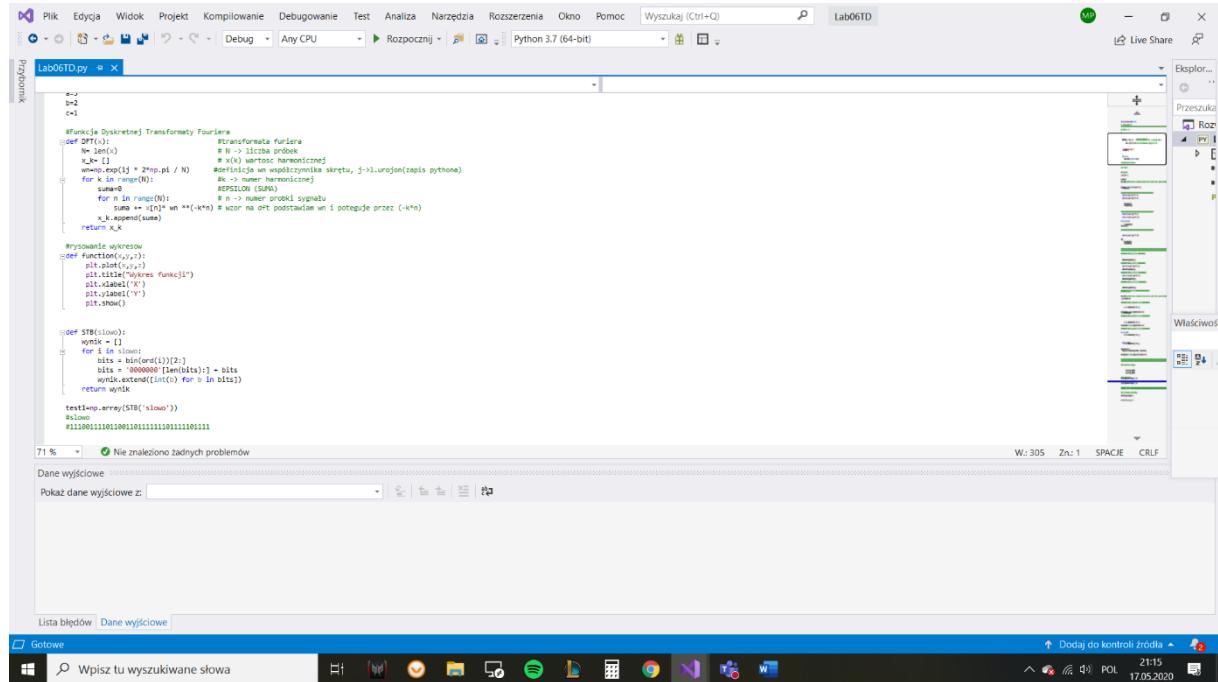


Sprawozdanie Lab06

Mateusz Proc 20 N1 nr 45123

Zad1/.



The screenshot shows a Microsoft Visual Studio Code interface with the following details:

- File Menu:** Plik, Edycja, Widok, Projekt, Kompilowanie, Debugowanie, Test, Analiza, Narzędzia, Rozszerzenia, Okno, Pomoc.
- Search Bar:** Wyszukaj (Ctrl+Q).
- Code Editor:** Lab06TD.py file open, containing Python code for signal processing. The code includes functions for DFT, plotting signals, and binary conversion.
- Output Panel:** Shows "Nie znaleziono żadnych problemów" (No errors found).
- Taskbar:** Shows various application icons.
- System Tray:** Shows battery level, signal strength, and date/time (17.05.2020).

```
def DyskretnaTransformataFouriera(x):
    def DFT():
        n = len(x)
        x_k = []
        w_n = np.exp(1j * 2 * np.pi / N)
        for k in range(N):
            sum = 0
            for n in range(N):
                sum += x[n] * w_n ** (-k * n) # wzor na dft podstawiam w i poteguje przez (-kn)
            x_k.append(sum)
        return x_k

    #rysowanie wykresu
    def function(v_x,y_x):
        plt.plot(v_x,y_x)
        plt.xlabel('X')
        plt.ylabel('Y')
        plt.show()

    def STB(bits):
        wynik = []
        for b in bits:
            wynik.append(int(b))
        wynik.extend([int(0) for b in bits])
        return wynik

    text=np.array(STB('1000'))
    bits = binord(1)[2:]
    bits = '0000000'[:len(bits)] + bits
    wynik.extend([int(b) for b in bits])
    return wynik

    test=np.array(STB('1000'))
    wynik = []
    for b in bits:
        wynik.append(int(b))
    wynik.extend([int(0) for b in bits])
    return wynik

def binord(i):
    bits = binord(i)[2:]
    bits = '0000000'[:len(bits)] + bits
    wynik.extend([int(b) for b in bits])
    return wynik

#def STB(bits):
#    wynik = []
#    for b in bits:
#        wynik.append(int(b))
#    wynik.extend([int(0) for b in bits])
#    return wynik

#text=np.array(STB('1000'))
#bits = binord(1)[2:]
#bits = '0000000'[:len(bits)] + bits
#wynik.extend([int(b) for b in bits])
#return wynik

test=np.array(STB('1000'))
bits = binord(1)[2:]
bits = '0000000'[:len(bits)] + bits
wynik.extend([int(b) for b in bits])
return wynik
```

Funkcja do rysowania wykresu, DFT i funkcja zamieniająca zadany ciąg znaków ASCII

```

# Dane do zadania
f1=0
f2=0.5
f1 = np.zeros(1)
A1=1
A2=3
Tb=1
Tb1=1 #przyjelam 1
N=1/Tb

f = N * (Tb ** -1)
f1 = (N + 1)/Tb
f2 = (N + 2)/Tb

x=50
z1=len(test1)

print(x/Tb)
print(int(pr1)) # konwertuje wartosc na int poniewaz linspace obra sie na wartosci float i sprawial mi problemy z wyświetlaniem

t = np.linspace(0,z1,pr1)
x = np.linspace(0,z1,z1)

interpolacja = np.interp(x, test1, kind='previous')
t0 = np.interp(pr1,t)
function=t,Tb,x,g

```

Dane wyjściowe

Pokaz dane wyjściowe Z:

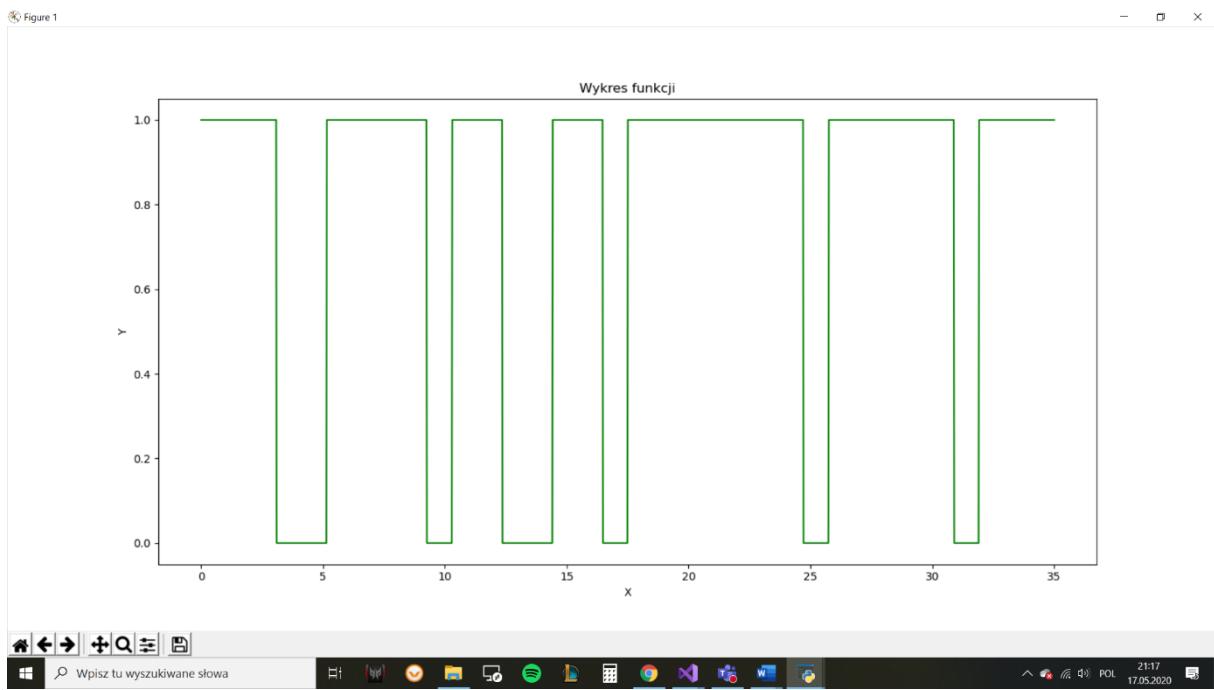
71 % Nie znaleziono żadnych problemów

W: 305 Zn: 1 SPACJE CRLF

Lista błędów | Dane wyjściowe

Gotowe Dodaj do kontroli źródła 21:16 17.05.2020

Dane potrzebne do wykonania zadania oraz wyświetlenie sygnału informacyjnego.



Wykres informacyjny

Screenshot of Microsoft Visual Studio Code showing Python code for generating a modulated signal.

```

Lb06TD.py ✘
#####
#Za(t)
def zat1(t):
    return A1 * np.sin(2 * np.pi *f*t + fi)

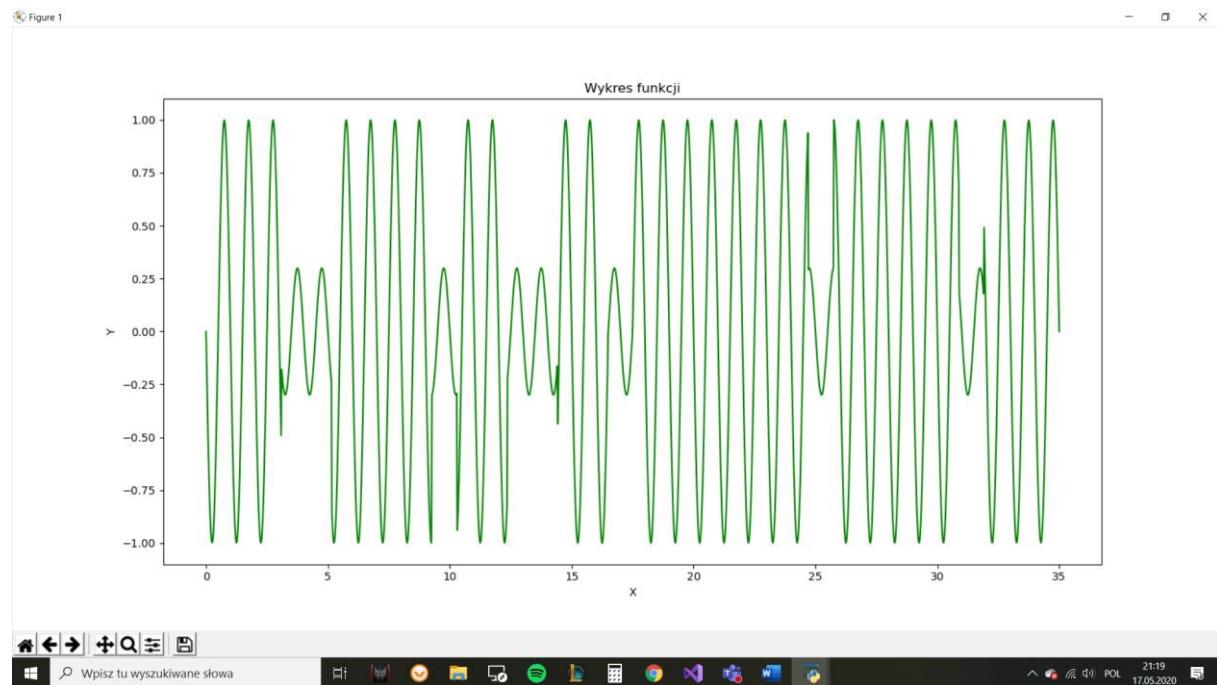
def zat0(t):
    return A2 * np.sin(2 * np.pi *f*t + fi)

ASK=[]
for i,j in zip(TBs,t):
    if i == 1:
        ASK.append(zat1(j))
    if i==0:
        ASK.append(zat0(j))

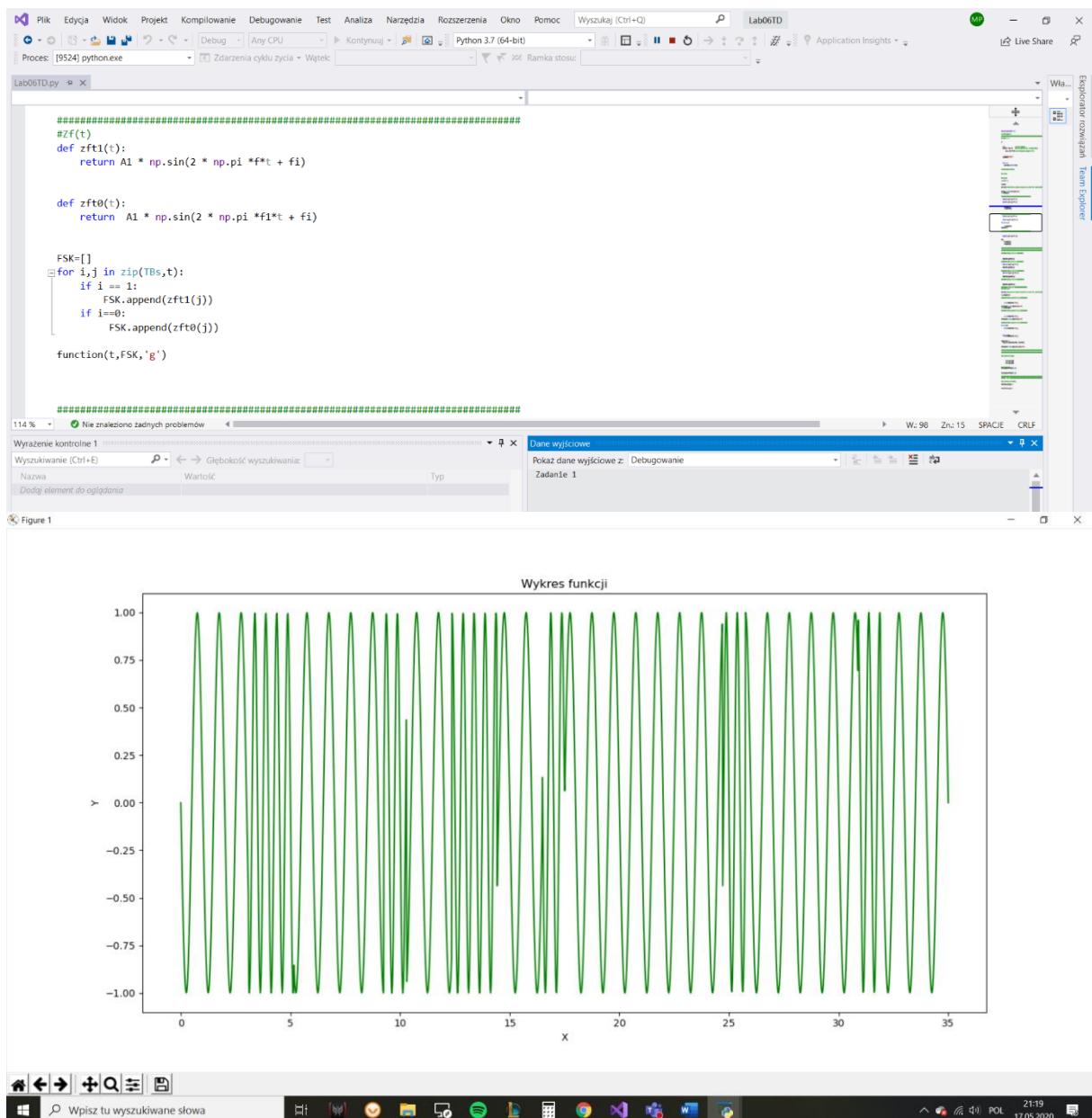
function(t,ASK,'g')
#####

114 % ✘ Nie znaleziono żadnych problemów
Wyszukiwanie (Ctrl+E) ← → Głębokość wyszukiwania: [ ]
Nazwa Wartość Typ
Dodaj element do oglądania
Dane wyjściowe
Pokaż dane wyjściowe z: Debugowanie
Zadanie 1
Stos wywołań Punkty przerwania Ustawienia wyjątków Okno polecenia Okno bezpośrednie Dane wyjściowe Lista błędów
Automatyczne Lokalne Wyrażenie kontrolne 1
Główne Dodaj do kontroli źródła 21:18 POL 17.05.2020
Windows Wpisz tu wyszukiwane słowa

```



Sygnal zmodulowany Za(t)



Sygnal zmodulowany Zf(t)

```

#####
#zp(t)
def zpt1(t):
    return A1 * np.sin(2 * np.pi *f*t + f10)

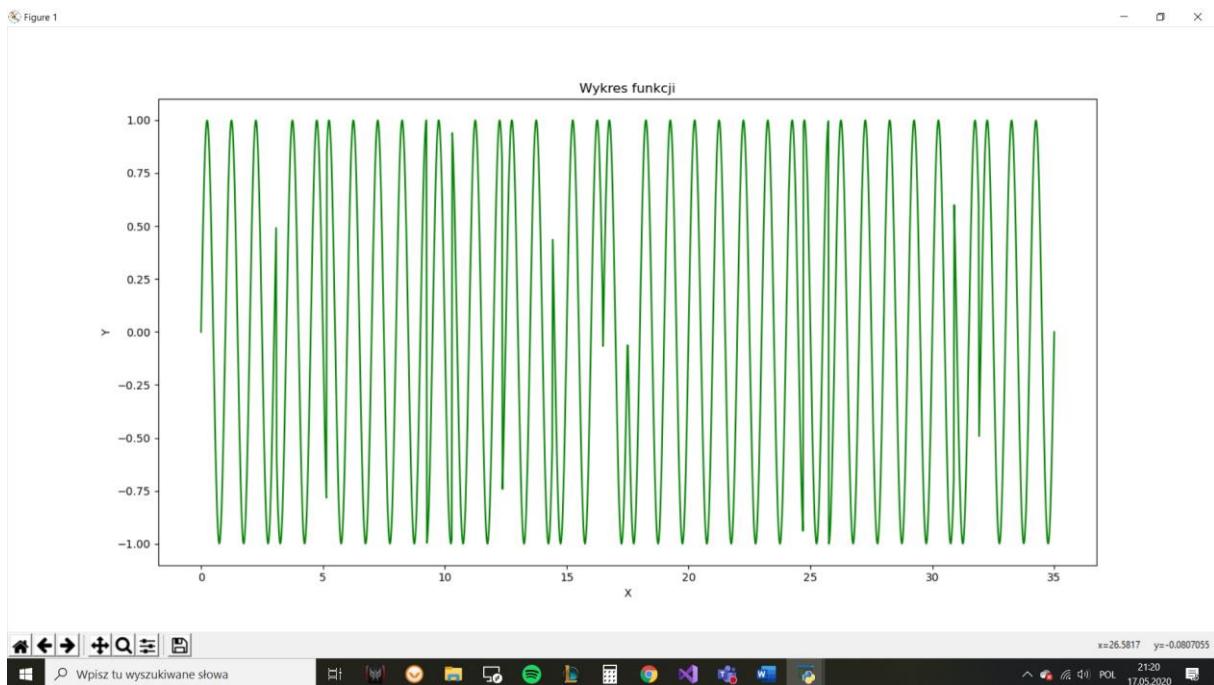
def zpt0(t):
    return A1 * np.sin(2 * np.pi *f*t + f11)

PSK=[]
for i,j in zip(Tbs,t):
    if i == 1:
        PSK.append(zpt1(j))
    if i==0:
        PSK.append(zpt0(j))

function(t,PSK,'g')

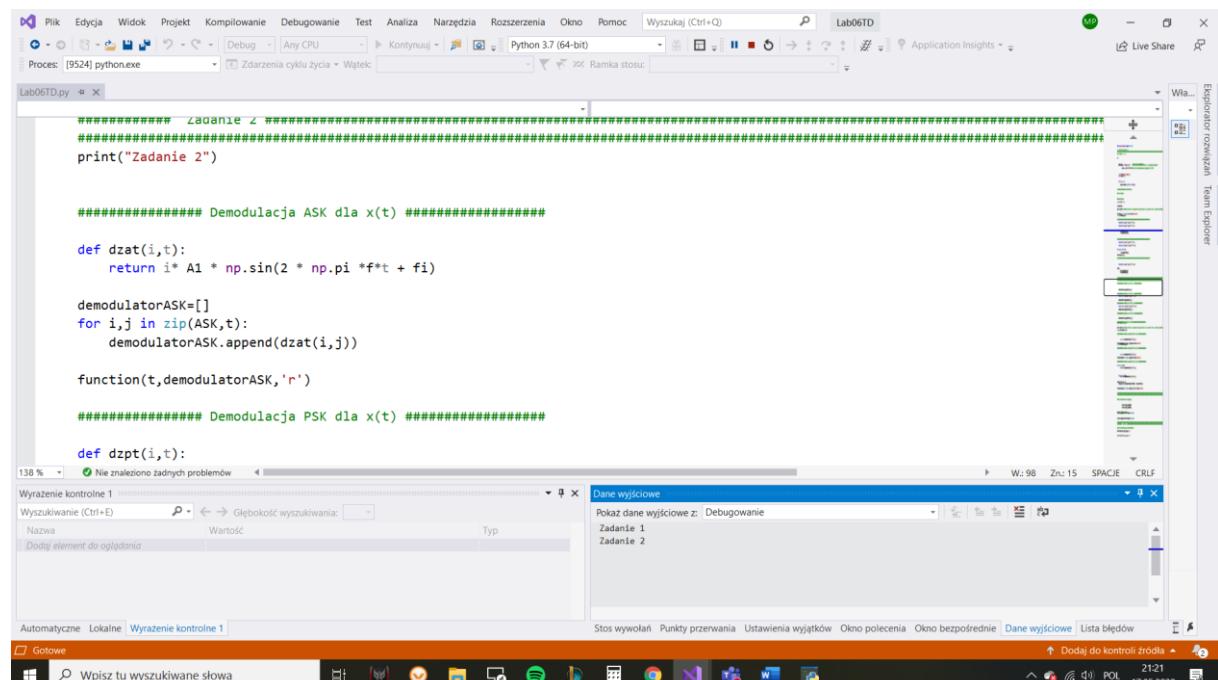
```

The screenshot shows the Microsoft Visual Studio Code interface. The main editor window contains Python code for generating a modulated signal. The code defines two functions, `zpt1` and `zpt0`, which calculate sine waves with different frequencies. It then creates a list `PSK` by iterating over a list of times `t` and baseband signals `Tbs`, appending the appropriate function result based on the index. Finally, it calls a `function` with parameters `t`, `PSK`, and a string argument. The status bar at the bottom indicates the file is 114% zoomed, has no errors, and shows system information like W: 98, Zn: 15, and CRLF.



Sygnal zmodulowany $Zp(t)$

Zad2/.



The screenshot shows the Microsoft Visual Studio Code interface with the following details:

- File Menu:** Plik, Edycja, Widok, Projekt, Kompilowanie, Debugowanie, Test, Analiza, Narzędzia, Rozszerzenia, Okno, Pomoc.
- Search Bar:** Wyszukaj (Ctrl+Q)
- Toolbar:** Debug, Any CPU, Kontynuuj, Python 3.7 (64-bit), Application Insights.
- Code Editor:** Lab06TD.py
- Code Content:**

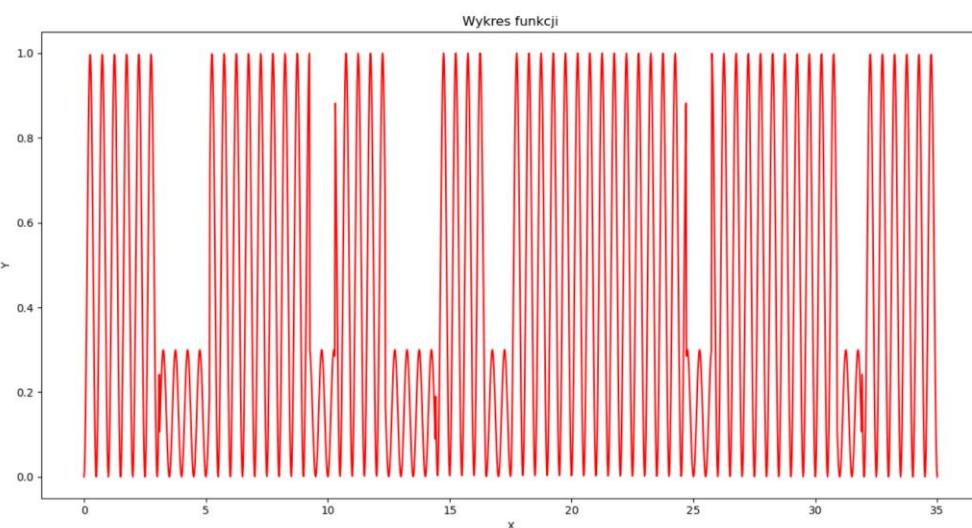
```
#####
# Zadanie 2 #####
#####
print("Zadanie 2")

#####
# Demodulacja ASK dla x(t) #####
def dzat(i,t):
    return i* A1 * np.sin(2 * np.pi *f*t + fi)

demodulatorASK=[]
for i,j in zip(ASK,t):
    demodulatorASK.append(dzat(i,j))

function(t,demodulatorASK,'r')

#####
# Demodulacja PSK dla x(t) #####
def dzpt(i,t):
```
- Output Window:** Nie znaleziono żadnych problemów
- Search Results:** Wyrażenie kontrolne 1, Wyszukanie (Ctrl+E), Giełdowość wyszukiwania: Wartość, Dodaj element do oglądania.
- Taskbar:** Automatyczne, Lokalne, Wyrażenie kontrolne 1, Stos wywołań, Punkty przenawiania, Ustawienia wyjątków, Okno polecenia, Okno bezpośrednie, Dane wyjściowe, Lista błędów.
- System Taskbar:** Gotowe, Wpisz tu wyszukiwane słowa, 21:21, 17.05.2020.



A plot titled "Wykres funkcji" showing a periodic square wave signal. The x-axis is labeled "x" and ranges from 0 to 35. The y-axis is labeled "y" and ranges from 0.0 to 1.0. The signal alternates between two levels, approximately 0.05 and 0.95, with a period of about 4 units on the x-axis.



A search bar with the placeholder text "Wpisz tu wyszukiwane słowa".

Demodulacja ASK dla x(t)

Demodulacja PSK dla $x(t)$

```

def dzpt(i,t):
    return i* A1 * np.sin(2 * np.pi *f*t + fi)

demodulatorPSK=[]
for i,j in zip(PSK,t):
    demodulatorPSK.append(dzpt(i,j))

function(t,demodulatorPSK,'r')

## Demodulacja FSK dla  $x_1(t)$ 

```

Wyszukiwanie (Ctrl+E) Wyszukaj (Ctrl+Q) Live Share

Nie znaleziono żadnych problemów

Dane wyjściowe

Pokaż dane wyjściowe z: Debugowanie

Zadanie 1
Zadanie 2

Automatyczne Lokalne Wyrażenie kontrolne 1

Gotowe Dodaj do kontroli źródła

Wpisz tu wyszukiwane słowa

21:23 17.05.2020



Demodulacj PSK dla $x(t)$

Screenshot of Microsoft Visual Studio Code showing Python code for FSK demodulation:

```

##### Demodulacja FSK dla x1(t) #####
def dzft0(i,t):
    return i*A1 * np.sin(2 * np.pi *f1*t + fi)
demodulatoFSK0=[]
for i,j in zip(FSK,t):
    demodulatoFSK0.append(dzft0(i,j))
function(t,demodulatoFSK0,'r')

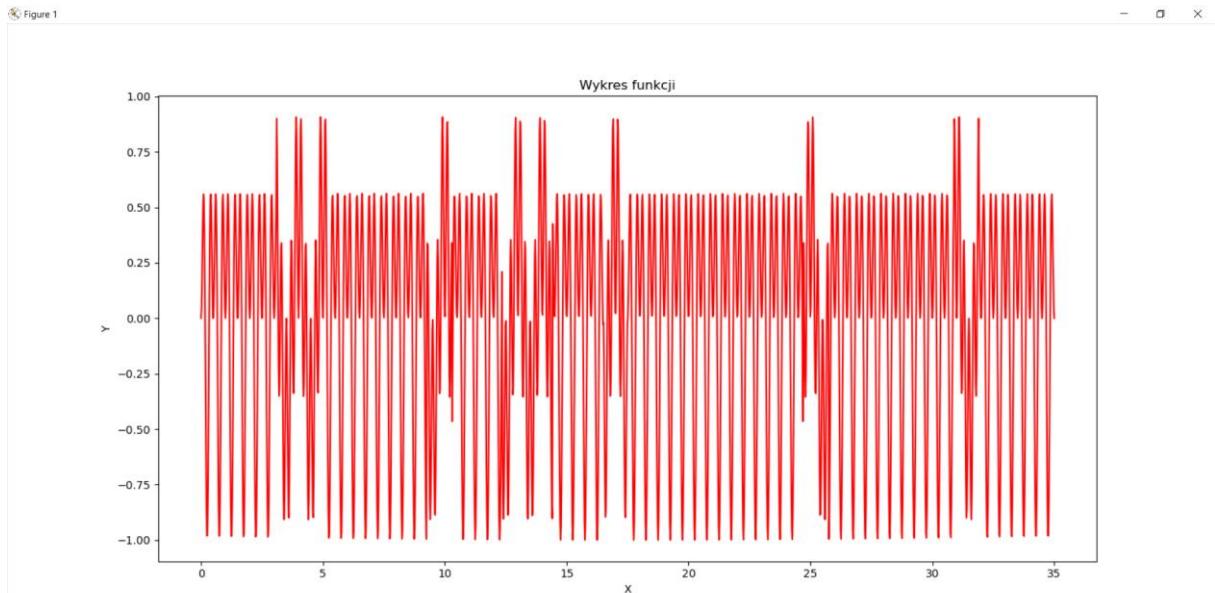
##### Demodulacja FSK dla x2(t) #####
def dzft1(i,t):
    return i*A1 * np.sin(2 * np.pi *f2*t + fi)
demodulatoFSK1=[]
for i,j in zip(FSK,t):
    demodulatoFSK1.append(dzft1(i,j))
function(t,demodulatoFSK1,'r')

##### Interpolate func #####
#dane napisane na gorze

```

VS Code interface with tabs, status bar showing W: 98, Zn: 15, SPACIE, CRLF, and system tray with date/time.





Demodulacja FSK dla $x_1(t)$ i $x_2(t)$

```

function(t,demodulatoFSK1,'r')

#####
# Interpolate func #####
#dane napisane na gorze
x=50
z1=len(test1)

prb=x*(z1/Tb)
prb1=int(prb) # konwertuje wartosc na int poniewaz linspace obraza sie na wartosci float i sprawial mi problemy z wyswietleniem

t = np.linspace(0,z1,prb1)
x = np.linspace(0,z1,z1)

#### Demodulacja Amplitudy ASK dla p(t) ####

integralASK = []
for i in range(z1):
    xi = 0
    for j in range(50):
        xi = xi + demodulatorASK1(i * 50) + i
    integralASK.append(xi)

```

Wystąpiły błędy:

- Nie znaleziono żadnych problemów

Dane wyjściowe

Pokaz dane wyjściowe z: Debugowanie

Zadanie 1
Zadanie 2

W: 98 Zn: 15 SPACIE CRLF

Automatyczne Lokalne Wyrażenie kontrolne 1

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

Gotowe

Wpisz tu wyszukiwane słowa

21:27 17.05.2020

Dane do interpolacji

Screenshot of Microsoft Visual Studio showing Python code for ASK demodulation and a step function plot.

```

Lab06TD.py  x

x = np.linspace(0,z1,z1)

#####
# Demodulacja Amplitudy ASK dla p(t) #####
#####

integralASK = []
for i in range(z1):
    xi = 0
    for j in range(s0):
        xi = xi + demodulatorASK[(i * 50) + j]
    integralASK.append(xi)

interpolatingASK=interp1d(x, integralASK, kind='previous')
ASK_pt=interpolatingASK(t)
function(t,ASK_pt, g)

#####
# Demodulacja Amplitudy PSK dla p(t) #####
#####

114 %  Nie znaleziono żadnych problemów  W: 228  Zn: 1  SPACJE  CRLF
Wyszukiwanie kontrolne 1  Wyrażenie kontrolne 1
Wyszukiwanie (Ctrl+E)  ← → Głębokość wyszukiwania:  Nazwa  Wartość  Typ
Dodaj element do oglądania

Dane wyjściowe
Pokaz dane wyjściowe z: Debugowanie
Zadanie 1
Zadanie 2

Automatyczne Lokalne Wyrażenie kontrolne 1
Główne  Wpisz tu wyszukiwane słowa  Dodaj do kontroli źródła  21:27  17.05.2020

```

Wykres funkcji

X	Y
[0, 2)	25.0
x=2	7.5
[2, 5)	7.5
x=5	25.0
[5, 8.5)	25.0
x=8.5	7.5
[8.5, 12)	7.5
x=12	25.0
[12, 15.5)	25.0
x=15.5	7.5
[15.5, 19)	7.5
x=19	25.0
[19, 22.5)	25.0
x=22.5	7.5
[22.5, 26)	7.5
x=26	25.0
[26, 30)	25.0
x=30	7.5
[30, 33.5)	7.5
x=33.5	25.0
[33.5, 36)	25.0

Windows Taskbar: 21:28 17.05.2020

Demodulacja Amplitudy ASK dla p(t)

Lab06TD.py

```
#####
# Demodulacja Amplitudu PSK dla p(t) #####
#####

integralPSK = []
for i in range(z1):
    xi = 0
    for j in range(50):
        xi = xi + demodulatorPSK[(i * 50) + j]
    integralPSK.append(xi)

interpolatingPSK=interp1d(x, integralPSK, kind='previous')
PSK_pt=interpolatingPSK(t)
function(t,PSK_pt,'g')

#####
# Demodulacja Amplitudu FSK dla p(t) #####
#####

integralFSK0 = 11
```

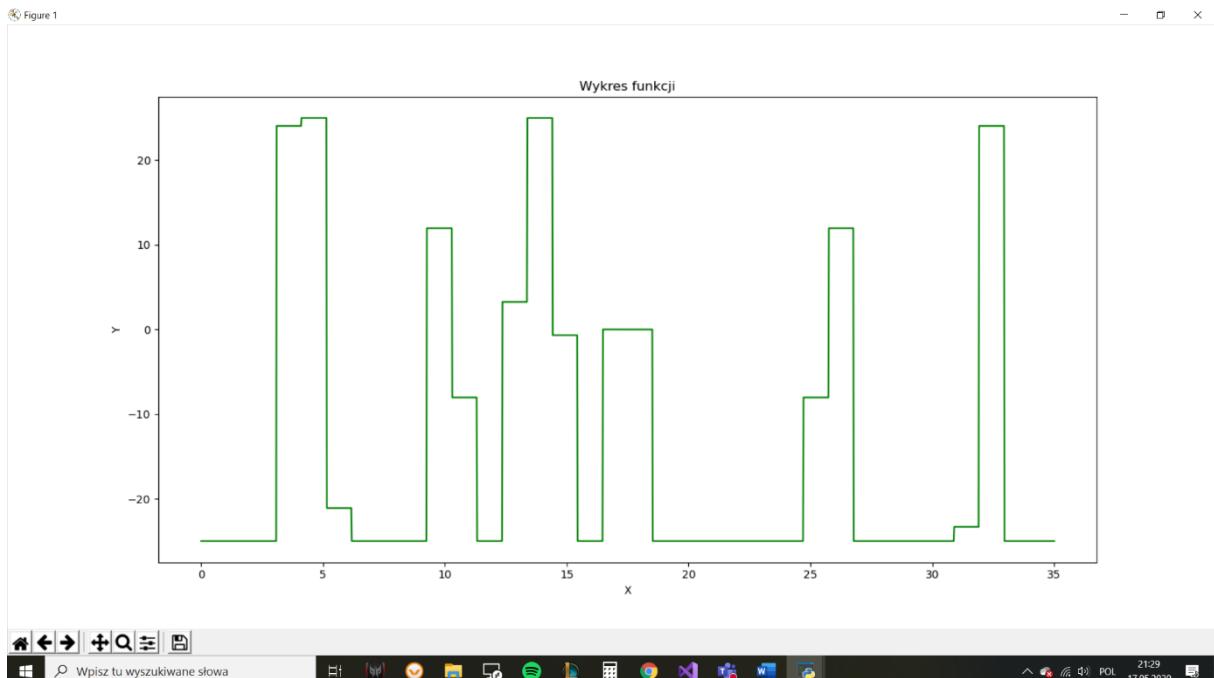
126 % Nie znalezione żadnych problemów

Wyszukiwanie kontrolne 1 Wyszukiwanie (Ctrl+E) Gębokość wyszukiwania Nazwa Wartość Typ Dodaj element do oglądania

Dane wyjściowe Pokaż dane wyjściowe z: Debugowanie Zadanie 1 Zadanie 2

Automatyczne Lokalne Wyrażenie kontrolne 1

Gotowe Dodaj do kontroli źródła 21:29 17.05.2020



Demodulacja Amplitudu PSK dla p(t)

Screenshot of Microsoft Visual Studio Code showing Python code for FSK demodulation:

```

function(t,p0K,p1,g):
    ##### Demodulacja Amplitudu FSK dla p(t) #####
    integralFSK0 = []
    for i in range(100):
        x0 = 0
        for j in range(50):
            x0 += g * demodulatorFSK0(i * 50 + j)
        integralFSK0.append(x0)

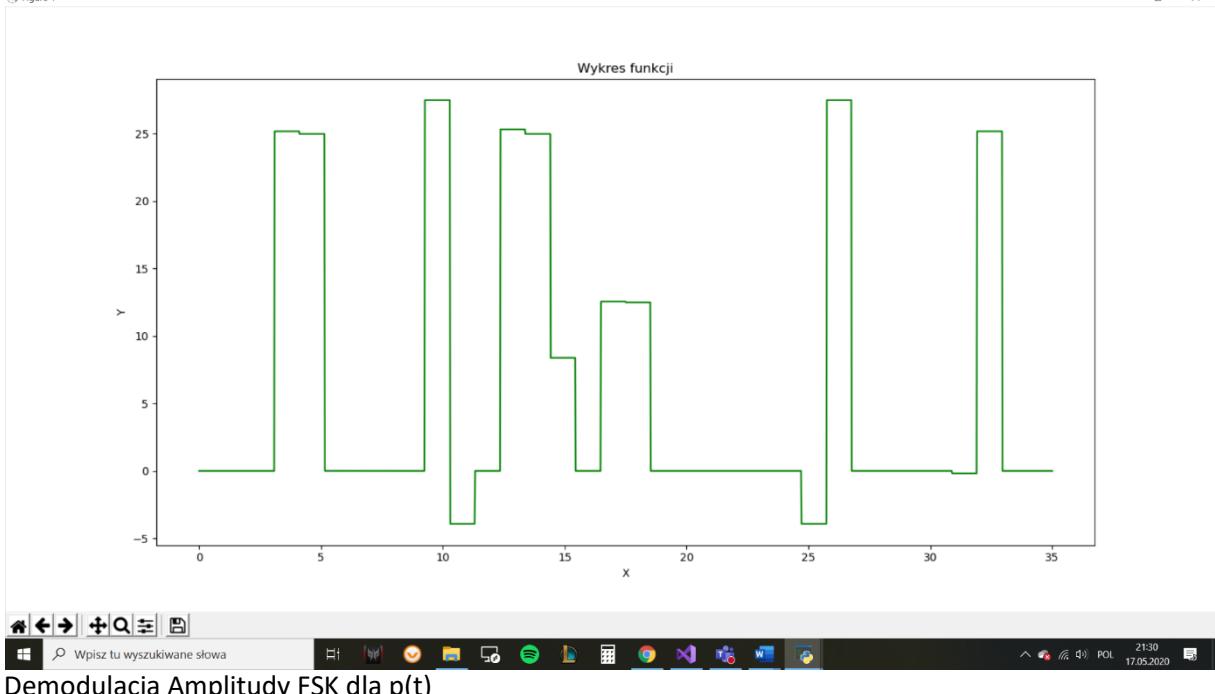
    integralFSK1 = []
    for i in range(100):
        x1 = 0
        for j in range(50):
            x1 += g * demodulatorFSK1((i * 50) + j)
        integralFSK1.append(x1)

    integral_FULL_FSK = []
    for i in range(100):
        integral_FULL_FSK.append(integralFSK0[i] - integralFSK1[i])

    interpolatingFSK(integral0, integral_FULL_FSK, kind='previous')
    P0 = interpolatingFSK(t)
    function(t,u,P0,g)

```

VS Code interface elements visible include the status bar (W: 213, Zn: 52, SPACJ, CRLF), the Explorer sidebar, and the Taskbar at the bottom.



Wpisz tu wyszukiwane słowa 21:30 17.05.2020

Demodulacja Amplitudu FSK dla p(t)

Zad 3/.

The screenshot shows the Microsoft Visual Studio Code interface with the following details:

- Code Editor:** The main window displays a Python script named `Lab06TD.py`. The code includes a function `wartoscProgowa` that calculates a threshold value based on a key and a parameter `h`. It also defines three threshold calculation functions: `ASK_wart_progowa`, `PSK_wart_progowa`, and `FSK_wart_progowa`.
- Output Panel:** Below the code editor, the "Wyszukiwanie kontrolne 1" (Search control 1) panel shows search results for "Zadanie 1", "Zadanie 2", "Zadanie 3", and "Zadanie 4".
- Terminal:** The bottom right corner shows a terminal window with the status bar indicating "W: 213 Zn: 52 SPACIE CRLF".
- Taskbar:** The taskbar at the bottom of the screen shows various open applications, including Microsoft Edge, Spotify, and File Explorer.

Dane do zadania, funkcja licząca wartość progową oraz wyznaczenie wartości progowej dla ASK, PSK i FSK

Zad 4./

The screenshot shows the Microsoft Visual Studio Code interface. The main editor window displays a Python script named `Lab06TD.py`. The code includes a header for "Zadanie 4" and three functions: `#ASK`, `#PSK`, and `#FSK`, each taking a parameter `t` and returning a string. Below the editor is a search bar with the placeholder "Wyszukaj (Ctrl+Q)". To the right of the editor is a sidebar titled "Eksplorator rozwiązań" (Solution Explorer) which lists several files and folders. At the bottom of the screen is a Windows-style taskbar with icons for File Explorer, Task View, Start, Taskbar settings, and system status.

```
#####
##### Zadanie 4 #####
#####
print("Zadanie 4")

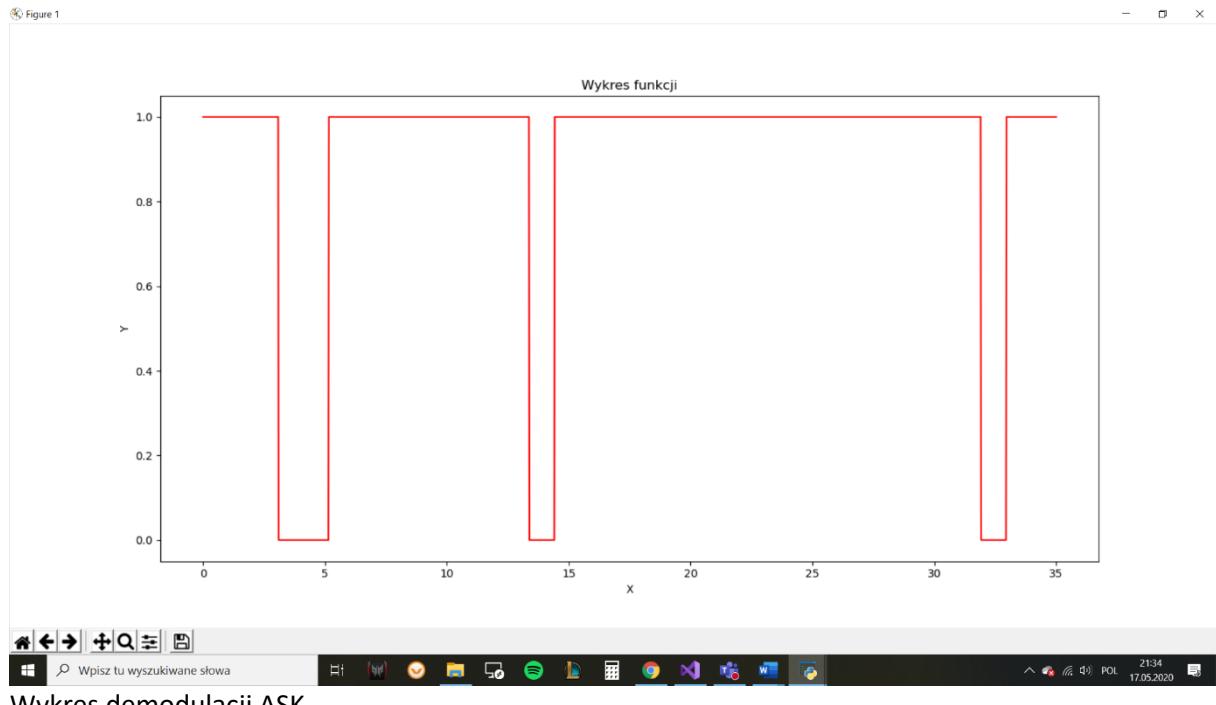
#wykresy przedstawiające wynik demodulacji

#ASK
function(t,ASK_wart_progowa,'r')

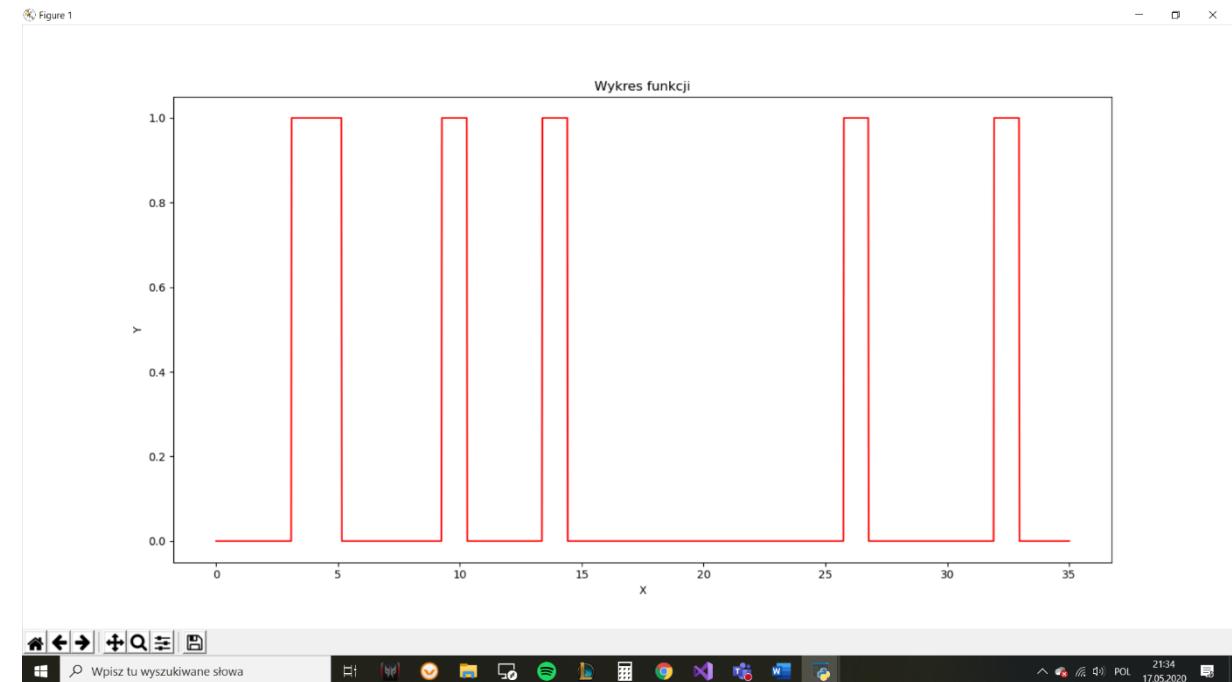
#PSK
function(t,PSK_wart_progowa,'r')

#FSK
function(t,FSK_wart_progowa,'r')
```

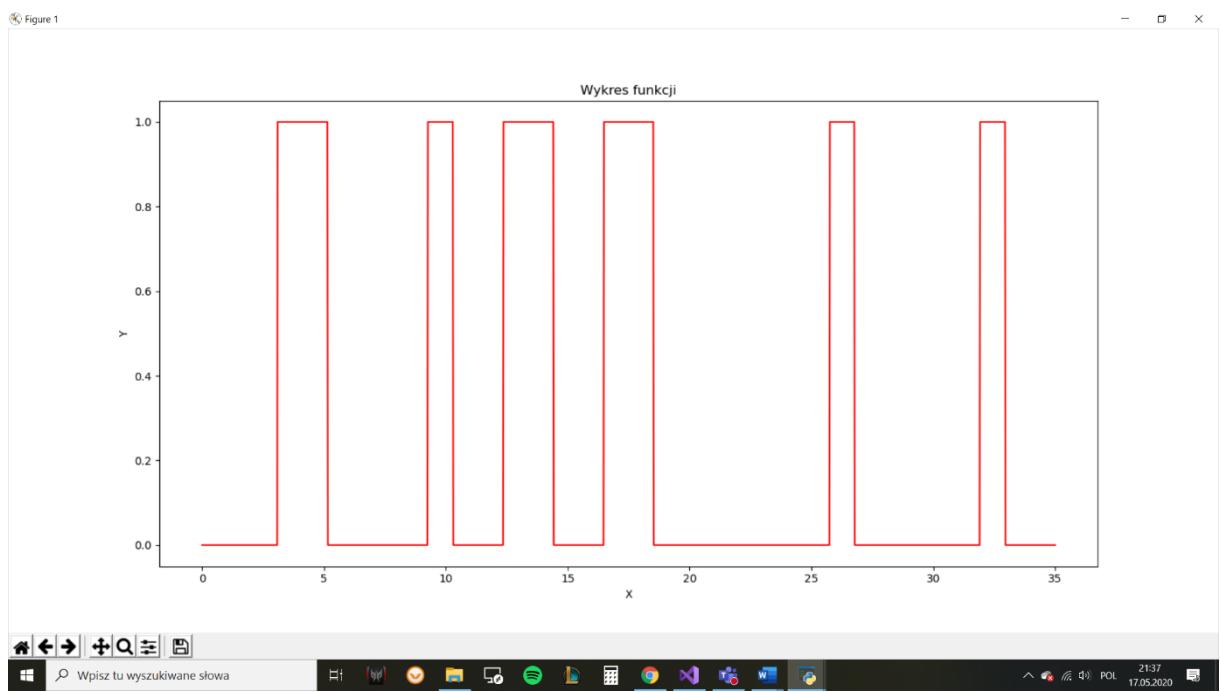
Wykresy przedstawiające wynik demodulacji ASK, PSK i FSK



Wykres demodulacji ASK



Wykres demodulacji PSK



Wykres demodulacji FSK