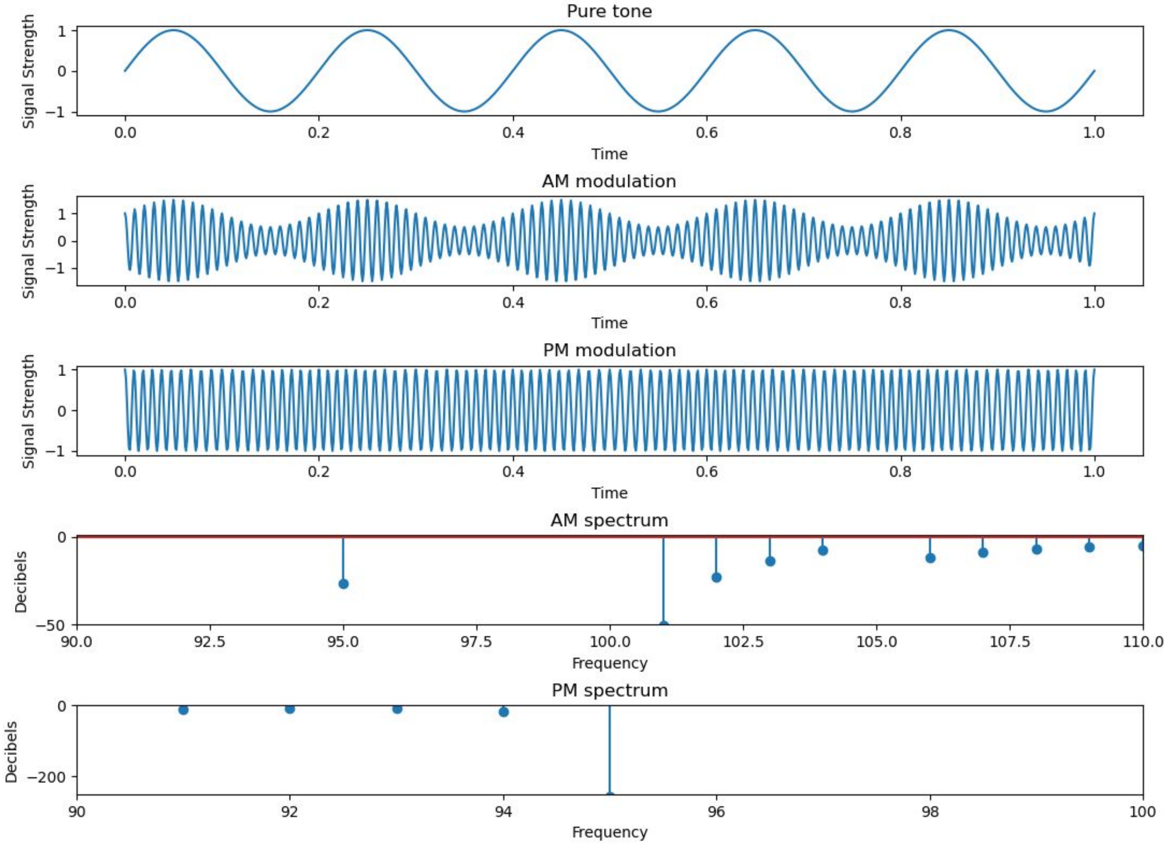
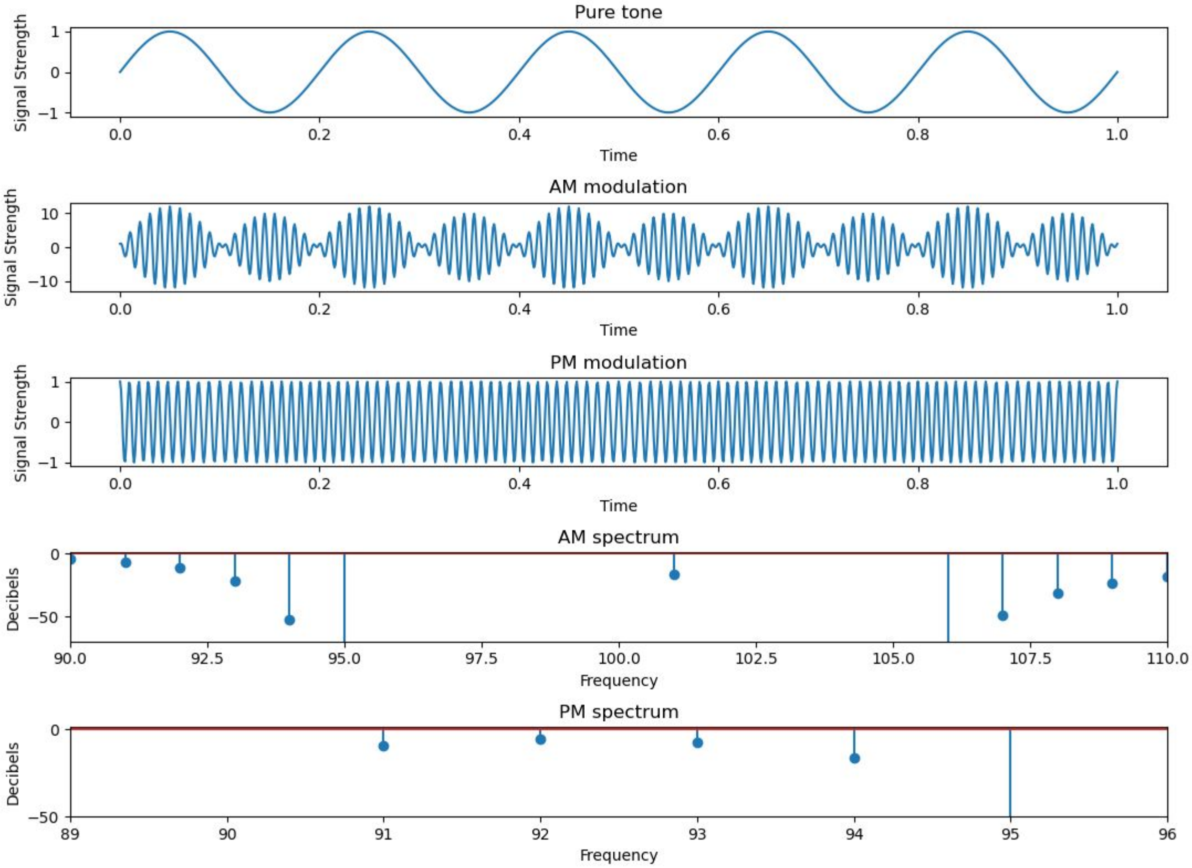
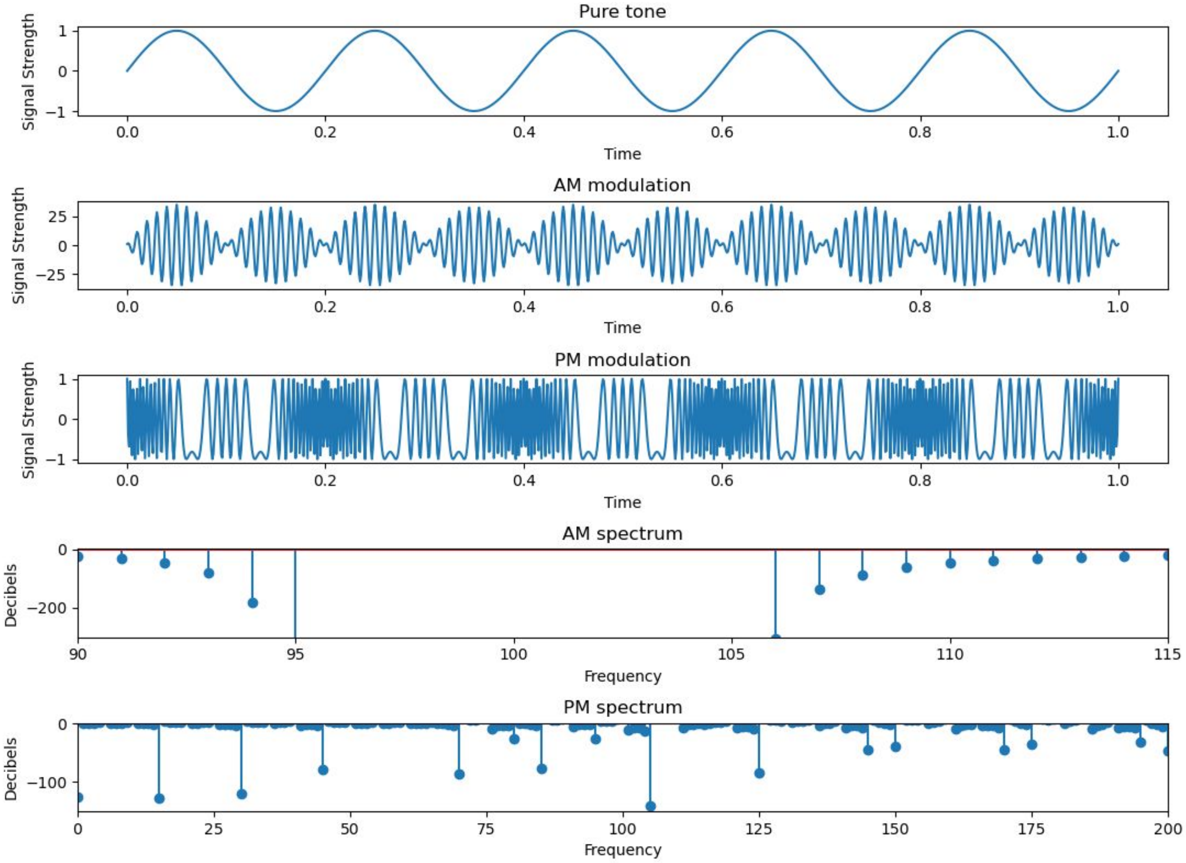
Sprawozdanie Lab04
Autor : Kamil Szóstak

```
import random
import math
import cmath
import numpy as np
import matplotlib.pyplot as plt

pi2 = cmath.pi * 2.0
def DFT(fnList):
    N = len(fnList)
    FmList = []
    for m in range(N):
        Fm = 0.0
        for n in range(N):
            Fm += fnList[n] * cmath.exp(- 1j * pi2 * m * n / N)
        FmList.append(Fm / N)
    return FmList

def InverseDFT(FmList):
    N = len(FmList)
    fnList = []
    for n in range(N):
        fn = 0.0
        for m in range(N):
            fn += FmList[m] * cmath.exp(1j * pi2 * m * n / N)
        fnList.append(fn)
    return fnList

def spectrum(N):
    X_re = N ; X_im = N ; x = N ; M=N
    X_rer = [] ; X_imi = [] ; EM = []
    N=len(N)
    for k in range(N-1):
        for n in range(N-1):
            X_rer.append(X_re[k]+(x[n]*np.cos((-2*np.pi*k*n)/N)))
            X_imi.append(X_im[k]+(x[n]*np.sin((-2*np.pi*k*n)/N)))

    for k in range(N-1):
        EM.append(np.sqrt(X_re[k]*X_re[k]+X_im[k]*X_im[k]))
    tresh = np.amax(EM)
    for k in range(N-1):
        if(EM[k]<tresh):
            EM[k]=0

    for k in range (N-1):
        EM[k]=10*np.log(M[k])
    return EM

def TonProsty():
    ycords = [];a=1;b=2;c=3;x=0
    while x <= a:
        funkcja = math.sin(2 * math.pi * b * x)
        ycords.append(funkcja)
        x=x+0.03
        x=round(x,2)
    return(ycords)

a=4;b=5;c=4;

acords = [] ; ycords = [] ; zcords = [] ; ucords = [] ; pcords = []

x=-10
while x < 10:
    fx = (a * (x * x)) + (b * x) +  c
    acords.append(fx)
    x=x+0.05
    x=round(x,2)

x=0
while x < 1:
    fx = (a * (x * x)) + (b * x) +  c
    funkcja = (2 * (fx * fx)) + (12*math.cos(x))
    ycords.append(funkcja)

    funkcja2 = (math.sin(2*math.pi*7*x)*fx)-0.2*math.log((abs(funkcja)+math.pi),10)
    zcords.append(funkcja2)

    funkcja3 = math.sqrt(abs(funkcja*funkcja*funkcja2))-1.8*math.sin(0.4*x*funkcja2*funkcja)
    ucords.append(funkcja3)
    x=x+0.01
    x=round(x,2)
```

```
plt.subplot(341)
plt.title('Ton prosty')
N=TonProsty()
plt.plot(N)

plt.subplot(342)
plt.title('DFT')
plt.xlim(0,10)
dft=DFT(N)
plt.stem(dft)

plt.subplot(343)
plt.title('Inverse DFT')
inv=InverseDFT(dft)
plt.plot(inv)

plt.subplot(344)
plt.title('Spectrum')
spec=spectrum(dft)
spec.pop(0)
plt.stem(spec)

plt.subplot(345)
plt.title('fx = (a * (x * x)) + (b * x) +  c')
plt.plot(acords)

plt.subplot(346)
plt.title('y(t)=2*x(t)^2+12*cos(t)')
plt.plot(ycords)

plt.subplot(347)
plt.title('sin(2pi*7*t)*x(t)-0.2*log10(abs(y(t))+pi)')
plt.plot(zcords)

plt.subplot(348)
plt.title('sqrt(abs(y(t)*y(t)*z(t)))-1.8*sin(0.4*t*z(t)*x(t))')
plt.plot(ucords)

A=DFT(acords)
AS=spectrum(A)
plt.subplot(349)
plt.title('Spectrum')
plt.stem(A)

B=DFT(ycords)
BS=spectrum(B)
plt.subplot(3,4,10)
plt.title('Spectrum')
plt.stem(BS)

C=DFT(zcords)
CS=spectrum(C)
plt.subplot(3,4,11)
plt.title('Spectrum')
plt.stem(CS)

D=DFT(ucords)
DS=spectrum(D)
plt.subplot(3,4,12)
plt.title('Spectrum')
plt.stem(DS)

plt.show()
```


Pure tone
Signal Strength
1
0
−1
0.0
0.2
0.4
0.6
0.8
1.0
Time
AM modulation
Signal Strength
1
0
−1
0.0
0.2
0.4
0.6
0.8
1.0
Time
PM modulation
Signal Strength
1
0
−1
0.0
0.2
0.4
0.6
0.8
1.0
Time
AM spectrum
Decibels
0
−50
90.0
92.5
95.0
97.5
100.0
102.5
105.0
107.5
110.0
Frequency
PM spectrum
Decibels
0
−200
90
92
94
96
98
100
Frequency


Pure tone
Signal Strength
1
0
−1
0.0
0.2
0.4
0.6
0.8
1.0
Time
AM modulation
Signal Strength
10
0
−10
0.0
0.2
0.4
0.6
0.8
1.0
Time
PM modulation
Signal Strength
1
0
−1
0.0
0.2
0.4
0.6
0.8
1.0
Time
AM spectrum
Decibels
0
−50
90.0
92.5
95.0
97.5
100.0
102.5
105.0
107.5
110.0
Frequency
PM spectrum
Decibels
0
−50
89
90
91
92
93
94
95
96
Frequency


Pure tone
Signal Strength
1
0
−1
0.0
0.2
0.4
0.6
0.8
1.0
Time
AM modulation
Signal Strength
25
0
−25
0.0
0.2
0.4
0.6
0.8
1.0
Time
PM modulation
Signal Strength
1
0
−1
0.0
0.2
0.4
0.6
0.8
1.0
Time
AM spectrum
Decibels
0
−200
90
95
100
105
110
115
Frequency
PM spectrum
Decibels
0
−100
0
25
50
75
100
125
150
175
200
Frequency