Transmisja Danych – Lab 04

Krystian Bartosik 213A, nr 44266

Kod źródłowy:

```
// Krystian Bartosik
// bk44266@zut.edu.pl
//   FEDCBA
#define _USE_MATH_DEFINES
#include <iostream>
#include <fstream>
#include "math.h"
#include <complex>

using namespace std;

complex<double>* DFT(double* Tab, int n)
{
	complex<double>* c = new complex<double>[n];
	complex<double> i = 0.0 + 1.0i;
	for (int k = 0; k < n; k++)
	{
		c[k] = 0.0 + 0.0i;

		for (int j = 0; j < n; j++)
		{
		c[k] = c[k] + (Tab[j] * exp(-2 * M_PI * i * (double)k * (double)j / double(n)));
		}
	}
	return c;
}

double TonProsty(double A, double f, double t)
{
    return A * sin(2 * M_PI * f * t);
}

double ModulacjaAmplitudy(double kA, double f, double t)
{
    return (kA * TonProsty(1.0, 6, t) + 1) * cos(2 * M_PI * 10 * f * t);
}

double ModulacjaFazy(double kP, double f, double t)
{
    return cos(2 * M_PI * 10 * f * t + kP * TonProsty(1.0, 6, t));
}

int main()
{
    fstream File;
    File.open("C:/Users/Qrystian/Desktop/results.txt", ios::out);

       double* Tab;
       double* M;
       complex<double>* X;

    double kA, kP;
```

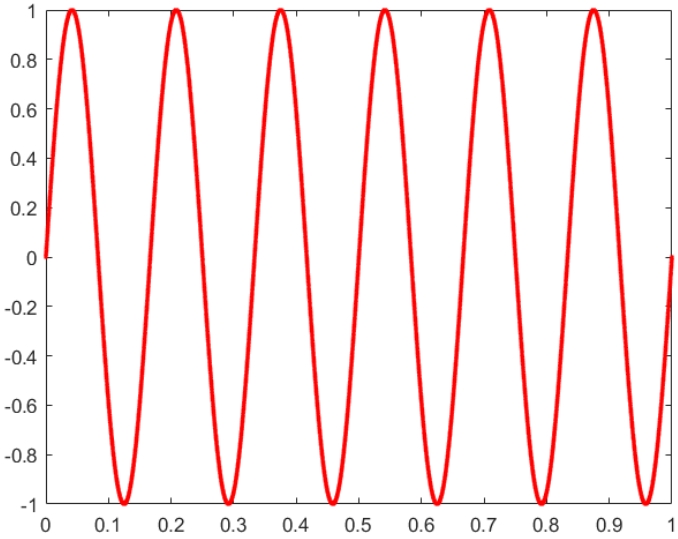
```
int size = (unsigned int)(1.0 / 0.001);
       Tab = new double[size];
       M = new double[size];

    //kA = 0.5; kP = 1.5;
    //kA = 5; kP = M_PI/2;
    kA = 70; kP = 70;

    for (double t = 0.0; t < 1; t = t + 0.0001)
    {
        //File << t << " " << TonProsty(1.0, 6.0, t) << endl;
        //File << t << " " << ModulacjaAmplitudy(kA, 6.0, t) << endl;
        //File << t << " " << ModulacjaFazy(kP, 6.0, t) << endl;
    }

       //     -      -      ZADANIE 2     -      -      //

       int j = 0;
       for (double t = 0.0; t <= 1; t = t + 0.001)
       {
              //Tab[j] = ModulacjaAmplitudy(kA, 6.0, t);
              Tab[j] = ModulacjaFazy(kP, 6.0, t);
              j++;
       }
       size = (unsigned int)(1.0 / 0.001);
       X = DFT(Tab, size);

       for (j = 0; j < size; j++)
       {
              M[j] = sqrt(pow(X[j].real(), 2) + pow(X[j].imag(), 2));
              M[j] = 10 * log10(M[j]);
       }

       j = 0;
       for (double t = 0.0; t <= 1; t = t + 0.001)
       {
              File << j * (0.001 /1.0) << " " << M[j] << endl;
              j++;
       }

       File.close();

       // Szerokość Za(t), kA=0.5 -> W = 0.06
       // Szerokość Zp(t), kA=1.5 -> W = 0.00004
       // Szerokość Za(t), kA=5 -> W = 0.0005
       // Szerokość Zp(t), kA=PI/2 -> W = 0.000058
       // Szerokość Za(t), kA=70 -> W = 0.0006
       // Szerokość Zp(t), kA=70 -> W= 0.0000116
}
```

Opis kodu:

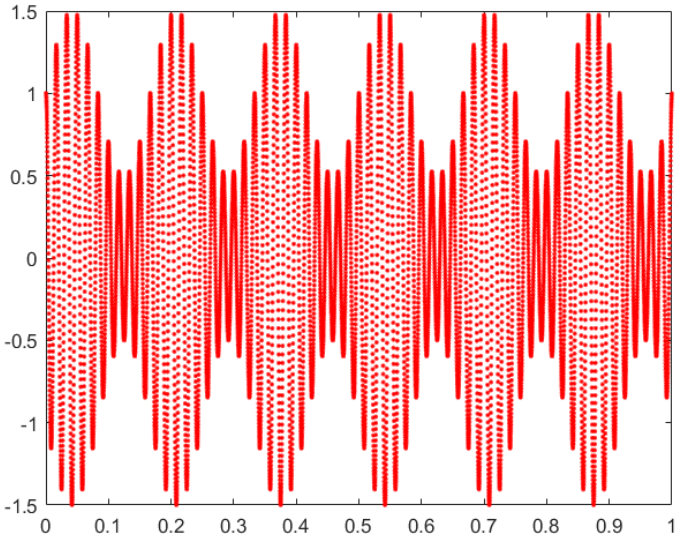
- Zadanie 1
- Zadanie 2

Do obliczenia szerokości konkretnych pasm, została wyznaczona wartość fmax i fmin. W komentarzu jest wynik różnicy.

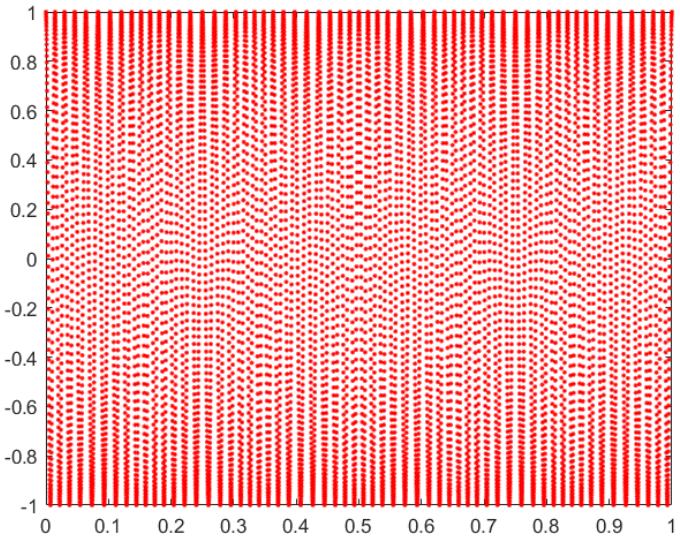
Wygenerowane wykresy:



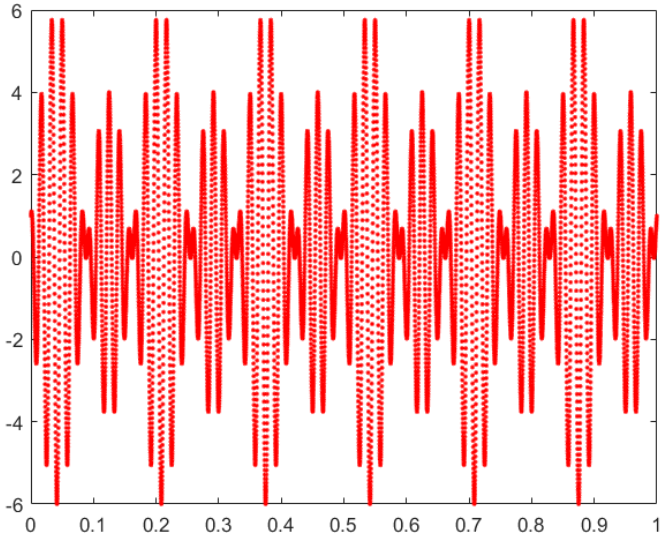
Wykres 1
Ton prosty



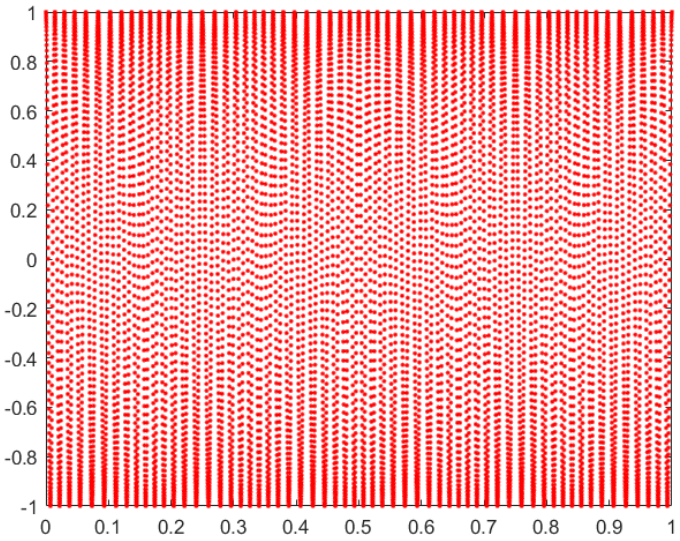
Wykres 2
a) kA=0.5



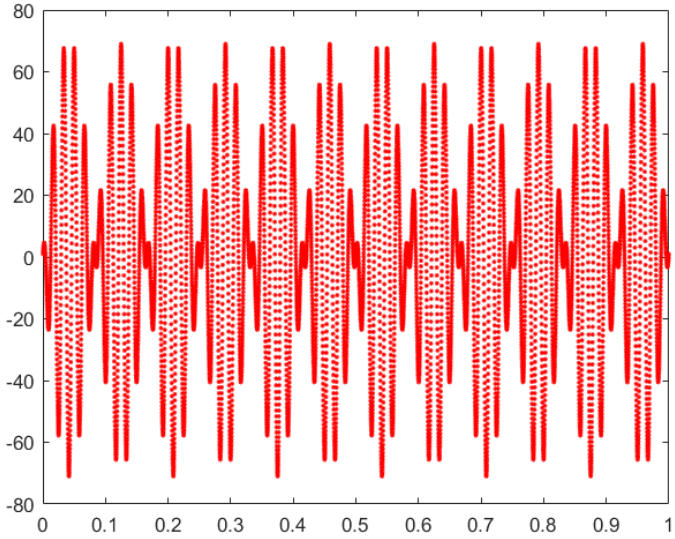
Wykres 3
a) kP=1.5



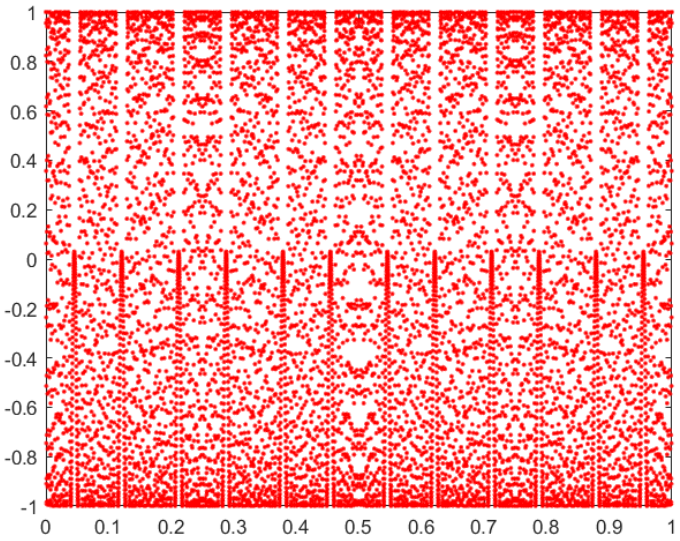
Wykres 4 b)
kA=5



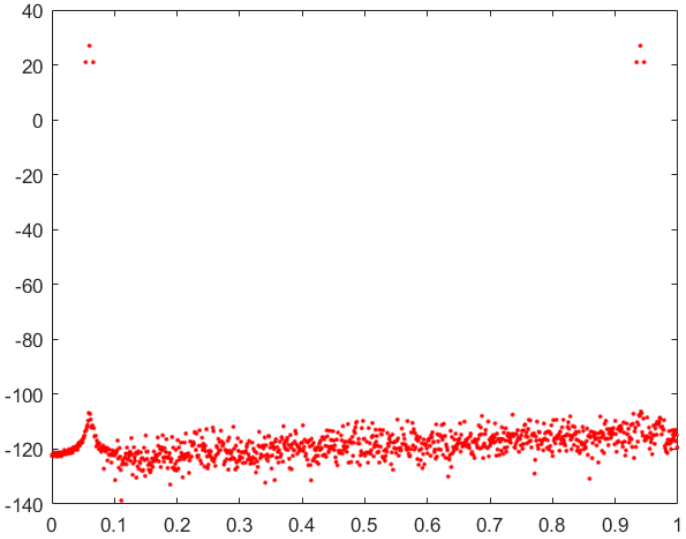
Wykres 5 b)
kP=PI/2



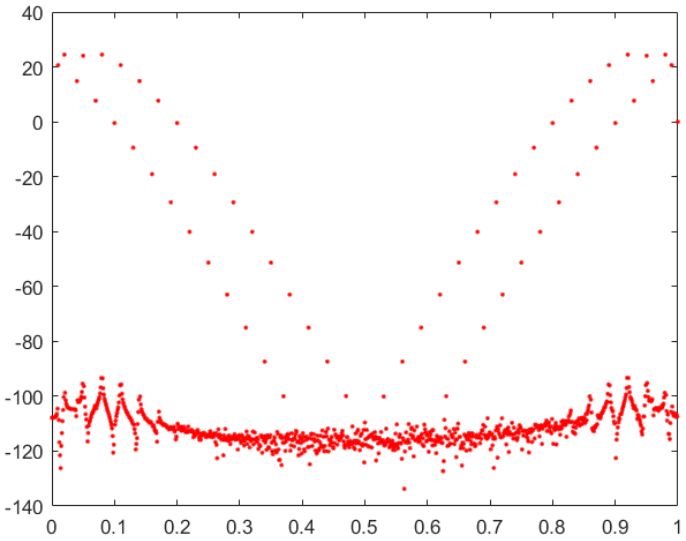
Wykres 6
c) kA=70



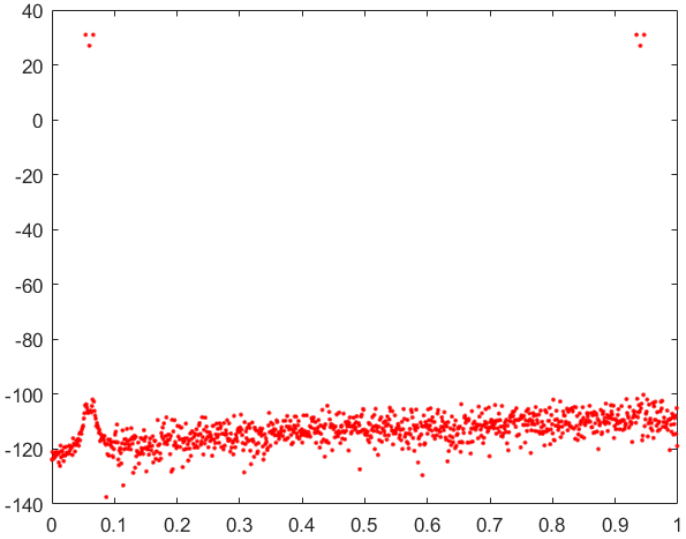
Wykres 7
c) kP=70



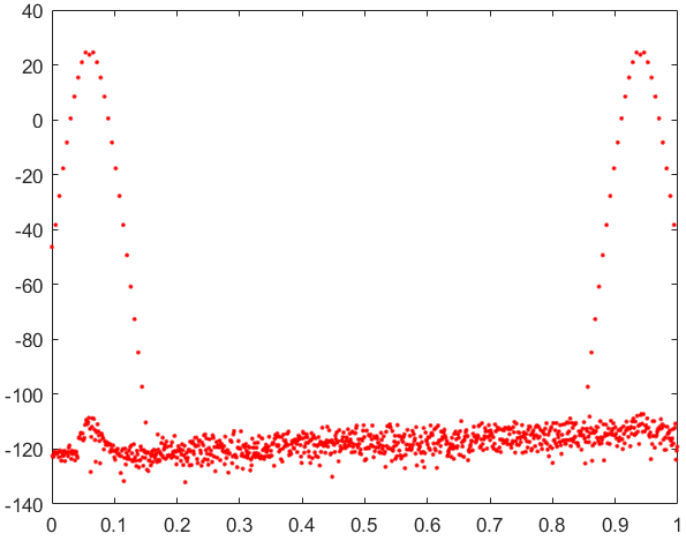
Wykres 8 Widmo a)
kA=0.5



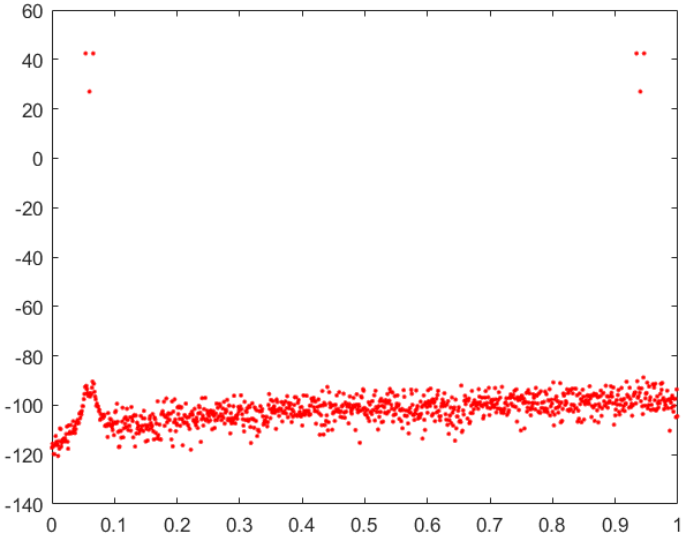
Wykres 9 Widmo a)
kP=1.5



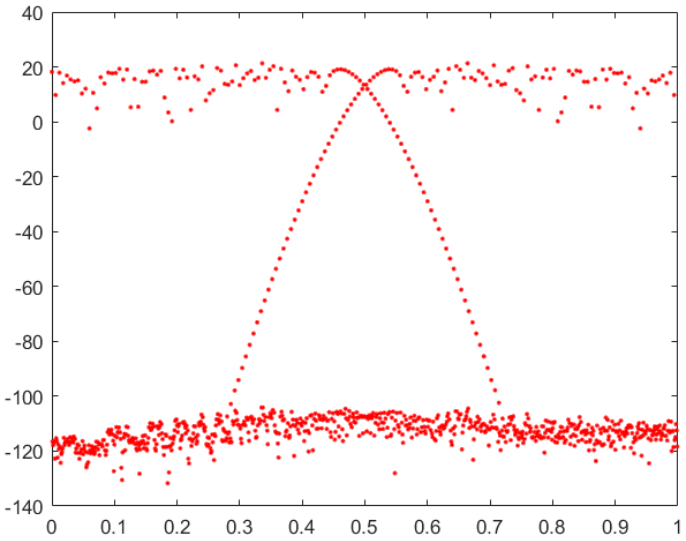
Wykres 10 Widmo
b) kA=5



Wykres 11 Widmo b)
kP=PI/2



Wykres 12 Widmo
c) kA=70



Wykres 13 Widmo c)
kP=70