

**Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska**

Percepcja Maszyn

Sprawozdanie z ćwiczenia laboratoryjnego nr 1

Mateusz Palczuk

Warszawa, 2022

Spis treści

1. Wstęp	2
2. Sygnały zaszumione	3
3. Optyczny system pomiaru tętna	7

1. Wstęp

W niniejszym sprawozdaniu opisany został przebieg prac związanych z zadaniem laboratoryjnym nr 1. Zadanie to polegało na wykorzystaniu transformaty Fouriera w praktyce do analizy obrazu.

Do realizacji został wykorzystany jedynie pakiet MATLAB.

2. Sygnały zaszumione

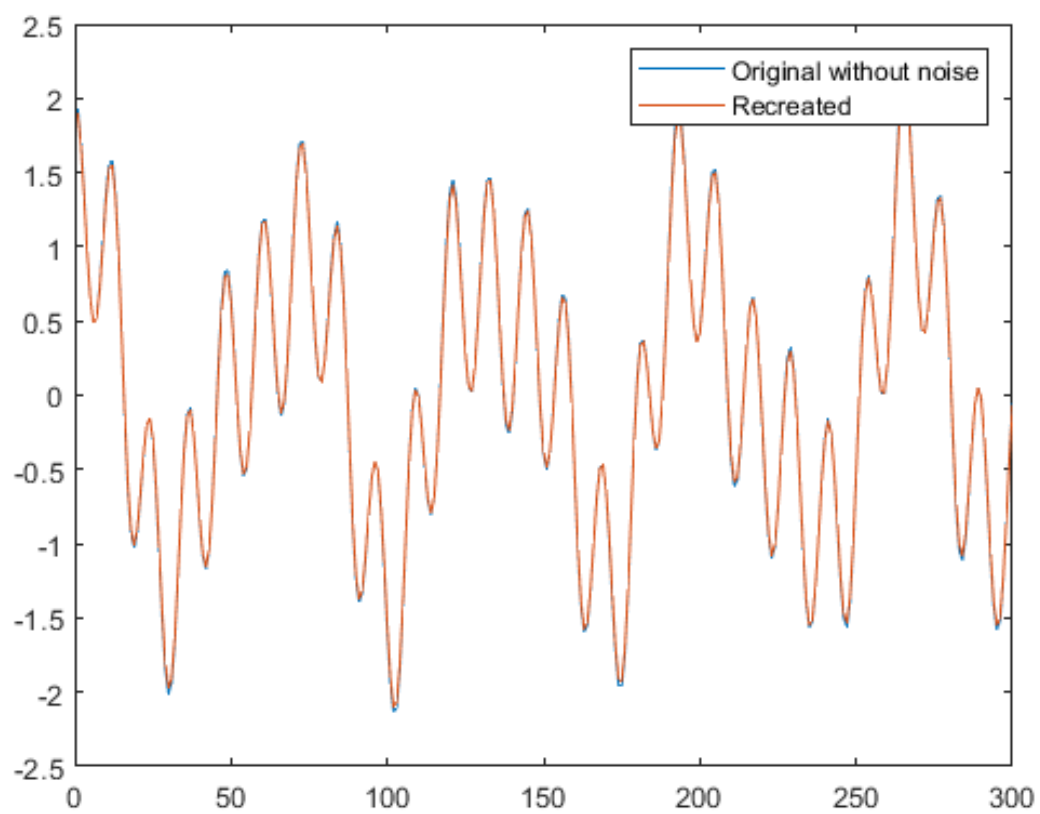
Program do generowania sygnału zaszumionego, a następnie jego odszumiania został umieszczony poniżej:

```
1 % Parametry systemu
2 Fs = 1000;      % Częstotliwość próbkowania [Hz]
3 T = 1/Fs;      % Okres próbkowania [s]
4 L = 2000;      % Długość sygnału (liczba próbek)
5 t = (0:L-1)*T; % Podstawa czasu
6
7 % Przygotowanie sygnału
8 N = 3;          % Liczba sinusoid w mieszaninie
9 A = [1.0  0.4  0.8] % Amplitudy kolejnych sinusoid
10 B = [ 15   27   83] % Częstotliwości kolejnych sygnałów [Hz]
11 C = [  0 -pi/3 pi/7] % Przesunięcia fazowe kolejnych sygnałów
12
13
14 x = zeros(size(t));
15 for i = 1:N
16     x = x + A(i) * cos(2 * pi * B(i) * t + C(i));
17 end
18 x_orig = x;
19 x=x+randn(size(t))/3;
20
21 Y = fft(x);      % transformata Fouriera
22
23 A = abs(Y);      % amplituda sygnału
24 A = A/L;         % normalizacja amplitudy
25
26 A = A(1:L/2+1); % wycięcie istotnej części spektrum
27 A(2:end-1) = 2*A(2:end-1);
28
29 F = angle(Y);    % faza sygnału
30 F = F(1:L/2+1); % wycięcie istotnej części spektrum
31
32 f_step = Fs/L;   % zmiana częstotliwości
33 f = 0:f_step:Fs/2; % oś częstotliwości do wykresu
34
35 figure;
36 subplot(2, 1, 1)
37 plot(f, A);      % wykres amplitudowy
38 % figure;
39 subplot(2, 1, 2)
40 plot(f, F);      % wykres fazowy
41
```

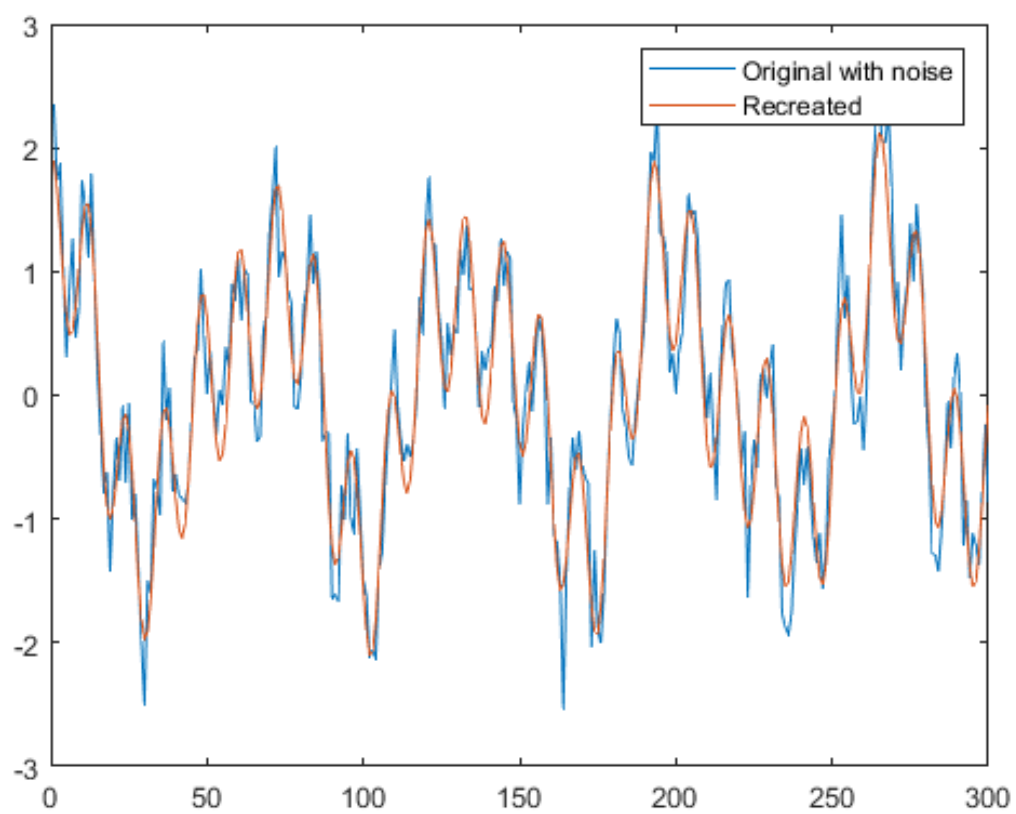
```
42 odtw = zeros(size(t));
43 [maxA, maxI] = maxk(A, 3);
44 freq = maxI;
45 freq = freq - 1;
46 freq = freq/2;
47 maxA
48 maxI
49 freq
50 for i=1:3
51     odtw = odtw + maxA(i) * cos(2 * pi * freq(i) * t + F(maxI(i)));
52 end
53
54 plot_size = 300;
55 figure; plot(x(1:plot_size));
56 hold on; plot(odtw(1:plot_size));
57 legend('Original with noise', 'Recreated');
58
59 figure; plot(x_orig(1:plot_size));
60 hold on; plot(odtw(1:plot_size));
61 legend('Original without noise', 'Recreated');
```

Wynik działania tego programu został przedstawiony na rys. 2.1 oraz rys. 2.2. Jak łatwo zauważyć na rys. 2.1 oba wykresy się dokładnie pokrywają. Jest to spowodowane faktem, że sygnał oryginalny składa się wyłącznie z trzech fali sinusoidalnych, więc jeżeli odzyskamy ten sygnał przy pomocy również trzech sinusoid otrzymamy dokładnie ten sam sygnał.

Sytuacja ma się inaczej na rys. 2.2 gdzie sygnał oryginalny został zaszumiony. Sygnał został odzyskany z dość dobrą dokładnością, ale nie jest to dokładnie to samo.



Rys. 2.1. Sygnał odzyskany nałożony na sygnał oryginalny bez zaszumienia



Rys. 2.2. Sygnał odzyskany nałożony na zaszumiony sygnał oryginalny

3. Optyczny system pomiaru tętna

Do pomiaru tętna został wykorzystany film dostarczony przez prowadzącego zajęcia. Następnie film został przerobiony na pojedyncze zdjęcia przy pomocy programu **Adobe Media Encoder**. Cały program został przedstawiony poniżej:

```
1 % Liczba ramek do wczytania (przy 10 sekundach i 30 FPS będzie to 300)
2 N = 321;
3
4 % wektor jasności
5 br = zeros(1, N);
6
7 % lista obrazów do analizy
8 imds = imageDatastore('movie', 'FileExtension', '.jpg');
9
10 % alternatywnie można załadować bezpośrednio plik wideo
11 % v = VideoReader('movie.mp4');
12
13
14 % wczytanie pierwszych N obrazów i analiza jasności
15 for i=1:N
16     % wczytujemy obraz i przekształcamy go do skali szarości
17     I = rgb2gray(imread(imds.Files{i}));
18     % dla pliku wideo ładowanie ramki z otwartego źródła
19     % I = rgb2gray(read(v,i));
20
21     % wyznaczamy średnią z całego obrazu
22     br(i) = mean(I, 'all');
23 end
24
25 % dla ułatwienia późniejszej analizy od razu można odjąć od sygnału składową stałą
26 br = br - mean(br);
27
28 % Parametry systemu
29 Fs = 30; % Częstotliwość próbkowania [Hz]
30 T = 1/Fs; % Okres próbkowania [s]
31 L = N; % Długość sygnału (liczba próbek)
32 t = (0:L-1)*T; % Podstawa czasu
33
34 x = br;
35 Y = fft(x); % transformata Fouriera
36
37 A = abs(Y); % amplituda sygnału
38 A = A/L; % normalizacja amplitudy
39
40 A = A(1:L/2+1); % wycięcie istotnej części spektrum
```



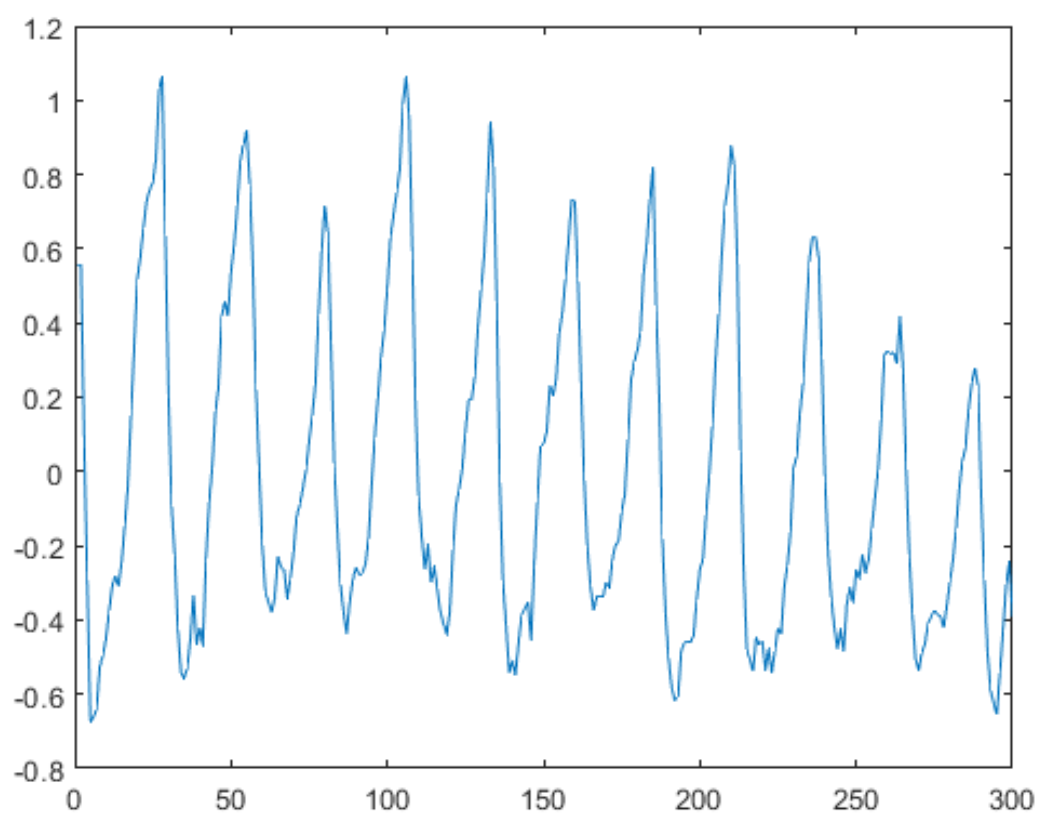
```
41 A(2:end-1) = 2*A(2:end-1);
42
43 F = angle(Y); % faza sygnału
44 F = F(1:L/2+1); % wycięcie istotnej części spektrum
45
46 f_step = Fs/L; % zmiana częstotliwości
47 f = 0:f_step:Fs/2; % oś częstotliwości do wykresu
48
49 figure;
50 subplot(2, 1, 1)
51 plot(f, A); % wykres amplitudowy
52 subplot(2, 1, 2)
53 plot(f, F); % wykres fazowy
54
55 [maxA, maxI] = maxk(A, 1);
56 bmp = f(maxI)*60
57
58 plot_size = 300;
59 figure; plot(br(1:plot_size));
```

Wynik tego programu został przedstawiony na rys. 3.1 oraz został wypisany w konsoli:

$$bmp = 67,2897 \quad (3.1)$$

Jak widać tętno przedstawione na rys. 3.1 jest klarownie widoczne i można by je zmierzyć nawet zliczając przejścia sygnału przez 0.

Również tętno zmierzone przy pomocy transformaty Fouriera (równanie 3.1) jest w pełni normalne, więc jest to wynik najpewniej poprawny.



Rys. 3.1. Reprezentacja graficzna zmierzonego tętna