

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 razem z komentarzami został umieszczony poniżej:

Stworzenie sygnału oryginalnego z szumem:

```
% Parametry systemu
Fs = 1000;      % Częstotliwość próbkowania [Hz]
T = 1/Fs;      % Okres próbkowania [s]
L = 2000;      % Długość sygnału (liczba próbek)
t = (0:L-1)*T; % Podstawa czasu

% Przygotowanie sygnału
N = 3;          % Liczba sinusoid w mieszaninie
A = [1.0  0.4  0.8] % Amplitudy kolejnych sinusoid
B = [ 15   27   83] % Częstotliwości kolejnych sygnałów [Hz]
C = [  0 -pi/3 pi/7] % Przesunięcia fazowe kolejnych sygnałów

x = zeros(size(t));
for i = 1:N
    x = x + A(i) * cos(2 * pi * B(i) * t + C(i));
end
x_orig = x;
x=x+randn(size(t))/3;
```

Wyznaczenie transformaty Fouriera i dostosowanie danych:

```
Y = fft(x);      % transformata Fouriera

A = abs(Y);      % amplituda sygnału
A = A/L;         % normalizacja amplitudy

A = A(1:L/2+1); % wycięcie istotnej części spektrum
A(2:end-1) = 2*A(2:end-1);

F = angle(Y);    % faza sygnału
F = F(1:L/2+1); % wycięcie istotnej części spektrum

f_step = Fs/L;   % zmiana częstotliwości
f = 0:f_step:Fs/2; % oś częstotliwości do wykresu
```

Wykresy amplitudowy i fazowy:

```
figure;
subplot(2, 1, 1)
plot(f, A);      % wykres amplitudowy
% figure;
subplot(2, 1, 2)
```

```
plot(f, F);          % wykres fazowy
```

Odtworzenie sygnału z danych pozyskanych z transformaty Fouriera i narysowanie wykresu:

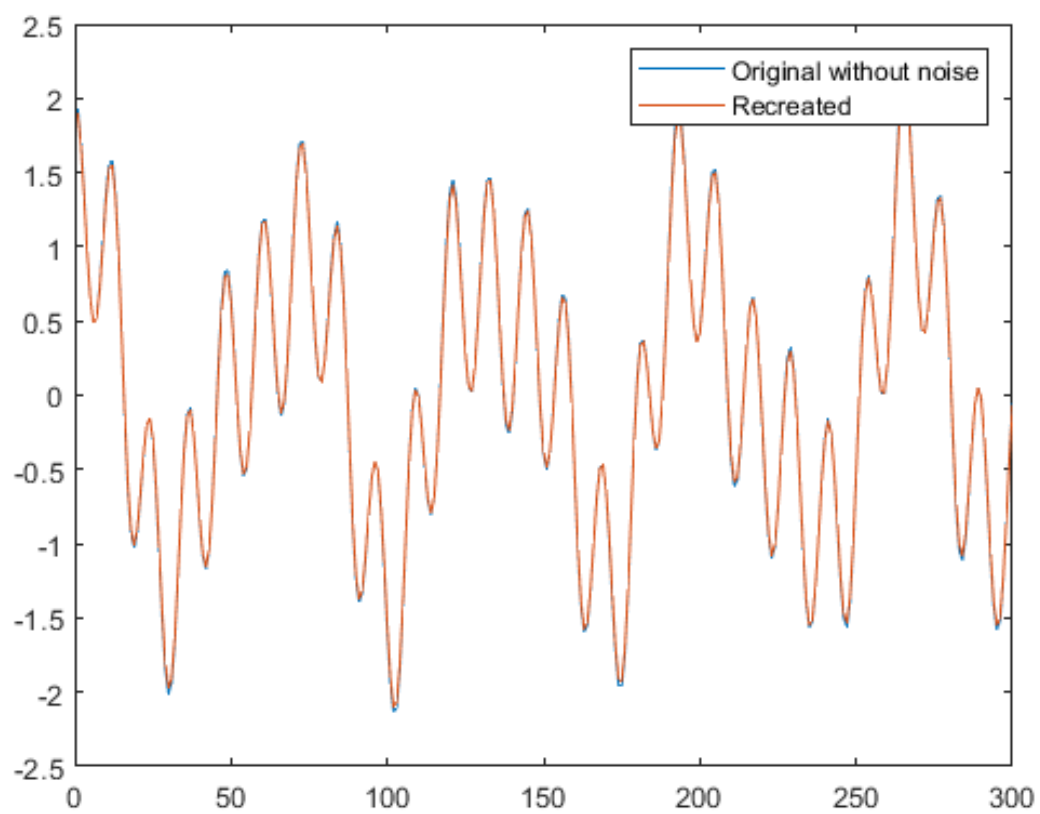
```
odtw = zeros(size(t));
[maxA, maxI] = maxk(A, 3);
freq = maxI;
freq = freq - 1;
freq = freq/2;
maxA
maxI
freq
for i=1:3
    odtw = odtw + maxA(i) * cos(2 * pi * freq(i) * t + F(maxI(i)));
end

plot_size = 300;
figure; plot(x(1:plot_size));
hold on; plot(odtw(1:plot_size));
legend('Original with noise', 'Recreated');

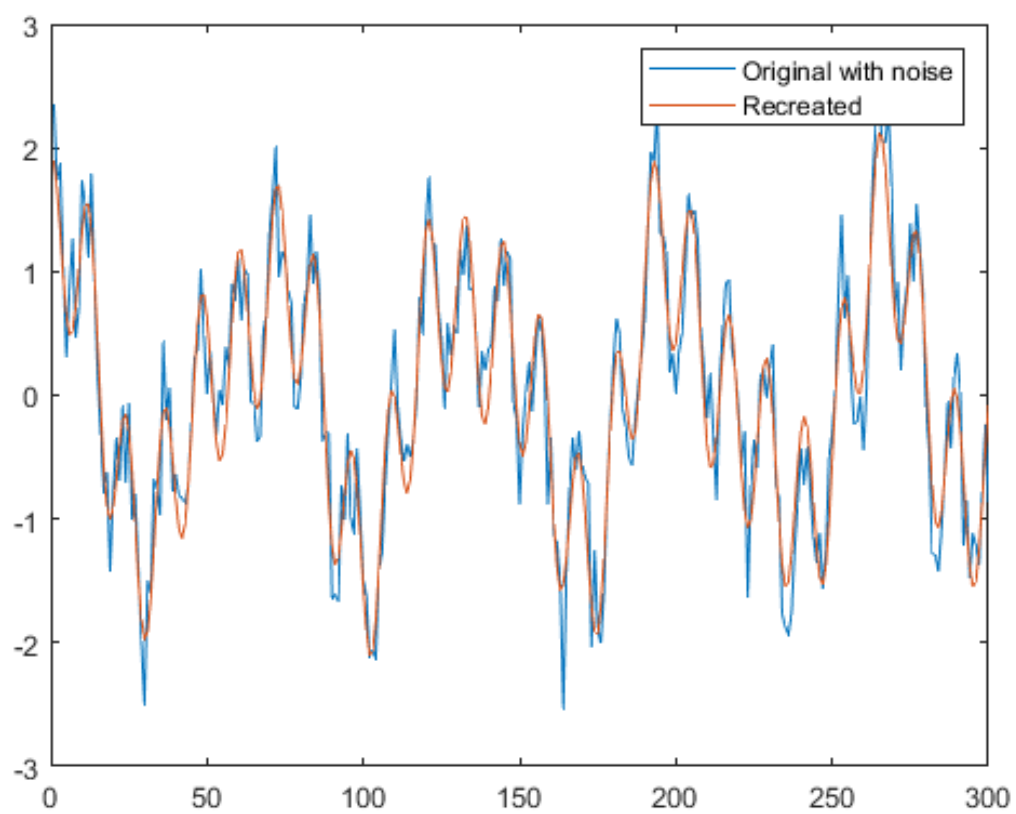
figure; plot(x_orig(1:plot_size));
hold on; plot(odtw(1:plot_size));
legend('Original without noise', 'Recreated');
```

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 razem z komentarzami poniżej.

Wczytanie zdjęć:

```
% Liczba ramek do wczytania (przy 10 sekundach i 30 FPS będzie to 300)
N = 321;

% wektor jasności
br = zeros(1, N);

% lista obrazów do analizy
imds = imageDatastore('movie', 'FileExtension', '.jpg');

% alternatywnie można załadować bezpośrednio plik wideo
% v = VideoReader('movie.mp4');

% wczytanie pierwszych N obrazów i analiza jasności
for i=1:N
    % wczytujemy obraz i przekształcamy go do skali szarości
    I = rgb2gray(imread(imds.Files{i}));
    % dla pliku wideo ładowanie ramki z otwartego źródła
    % I = rgb2gray(read(v,i));

    % wyznaczamy średnią z całego obrazu
    br(i) = mean(I, 'all');
end
```

Przetworzenie danych z obrazów:

```
% dla ułatwienia późniejszej analizy od razu można odjąć od sygnału składową stałą
br = br - mean(br);

% Parametry systemu
Fs = 30;      % Częstotliwość próbkowania [Hz]
T = 1/Fs;     % Okres próbkowania [s]
L = N;        % Długość sygnału (liczba próbek)
t = (0:L-1)*T; % Podstawa czasu

x = br;
Y = fft(x);   % transformata Fouriera

A = abs(Y);   % amplituda sygnału
A = A/L;      % normalizacja amplitudy
```



```

A = A(1:L/2+1); % wycięcie istotnej części spektrum
A(2:end-1) = 2*A(2:end-1);

F = angle(Y); % faza sygnału
F = F(1:L/2+1); % wycięcie istotnej części spektrum

f_step = Fs/L; % zmiana częstotliwości
f = 0:f_step:Fs/2; % oś częstotliwości do wykresu

```

Narysowanie wykresów i wyznaczenie tętna w uderzeniach na minutę:

```

figure;
subplot(2, 1, 1)
plot(f, A); % wykres amplitudowy
subplot(2, 1, 2)
plot(f, F); % wykres fazowy

[maxA, maxI] = maxk(A, 1);
bmp = f(maxI)*60

plot_size = 300;
figure; plot(br(1:plot_size));

```

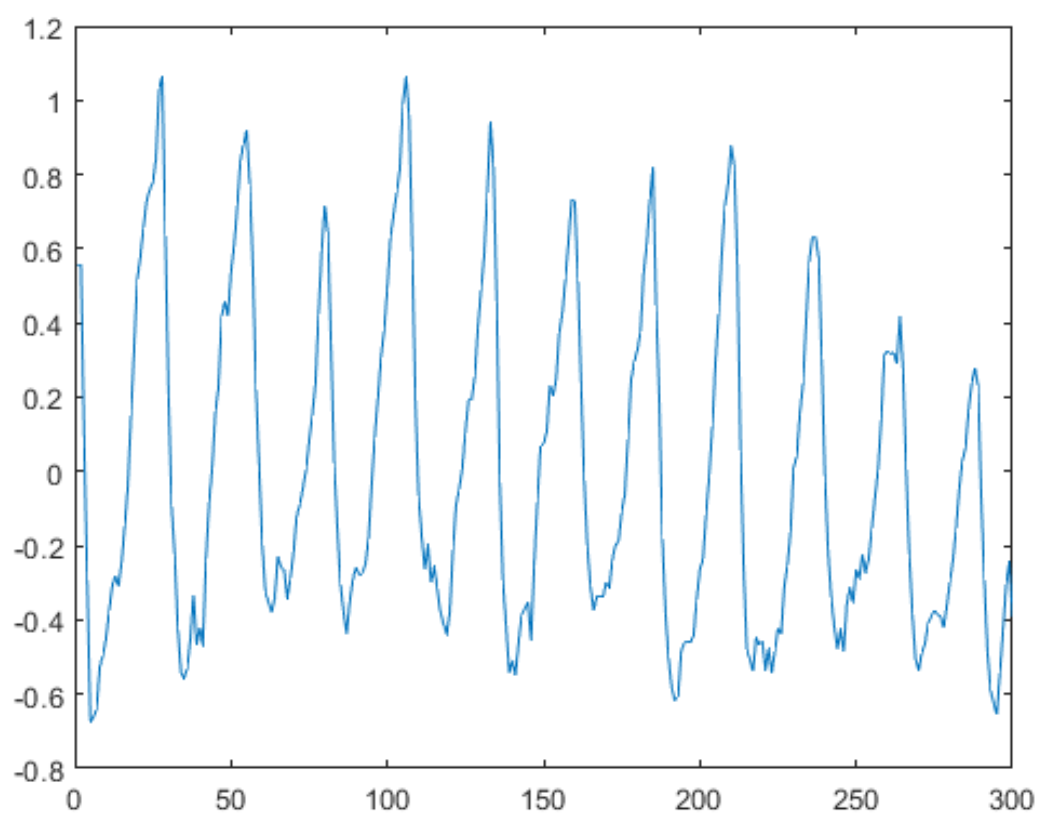
Wynik tego programu został przedstawiony na rys. 3.1 oraz został wypisany w konsoli:

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

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.

Na rozdzielczość pomiaru składa się wyłącznie liczba klatek na sekundę zarejestrowanych w filmie. W tym wypadku film miał 30 klatek na sekundę, więc pomiar miał rozdzielczość 30Hz. Aby zwiększyć dokładność pomiaru należałoby nagrać film w większej liczbie klatek na sekundę.



Rys. 3.1. Reprezentacja graficzna zmierzonego tętna