

SPRAWOZDANIE 12

Zajęcia nr 13 - Parametry sygnałów cyfrowych

Michał Midor, gr. 5 - 16.12.2025, godz: 15:00

WSTĘP

Falki pozwalają zredukować złożony sygnał do zestawu kilku kluczowych liczb (współczynników), które reprezentują jego charakter w różnych pasmach częstotliwości. Zamiast analizować tysiące próbek surowego sygnału, pobieramy np. wektor współczynników aproksymacji (np. a_3 lub a_2), który zawiera "esencję" sygnału (trend) i używamy go jako wektora wejściowego do klasyfikatorów, takich jak k-Najbliższych Sąsiadów (KNN) czy sieci neuronowe.

W środowisku MATLAB (Toolbox Wavelet Analyzer) kluczowymi parametrami są:

- Rodzina i typ falki: (np. haar, db, sym, coif) – decyduje o kształcie filtru.
- Poziom dekompozycji: (1, 2, ..., N) – określa głębokość analizy i podział pasm.

Współczynniki $a_1, \dots, a_n, d_1, \dots, d_n$ reprezentują różne pasma częstotliwościowe sygnału:

- a_j (aproksymacje): Zawierają informację dolnoprzepustową (tło, główne trendy sygnału).
- d_j (detale): Zawierają informację o wysokich częstotliwościach. Im niższy indeks j (np. d_1), tym wyższa częstotliwość (szum, nagłe zmiany). d_1 to najwyższe częstotliwości, d_2 to połowa pasma d_1 , itd...

Kluczowa różnica między transformacją a filtrami polega na strukturze: transformacja falkowa wykonuje tę filtrację kaskadowo i wielopoziomowo (drzewo dekompozycji), stosując przy tym decymację (usuwanie co drugiej próbki), co pozwala na analizę wielorozdzielczą (Multiresolution Analysis) – dobrą rozdzielczość czasu dla wysokich częstotliwości i dobrą rozdzielczość częstotliwości dla niskich.

Zastosowania transformacji falkowej to między innymi:

- Odszumianie: Usuwanie detali (d_1, d_2) zawierających szum przy zachowaniu kształtu sygnału.
- Kompresja: Np. standard JPEG.
- Diagnostyka: Wykrywanie uszkodzeń maszyn.

WYKONANIE ĆWICZEŃ

1)

```
[y_jablko, Fs_jablko] = audioread("jablko.aac");
signal_jablko = y_jablko(:,1);

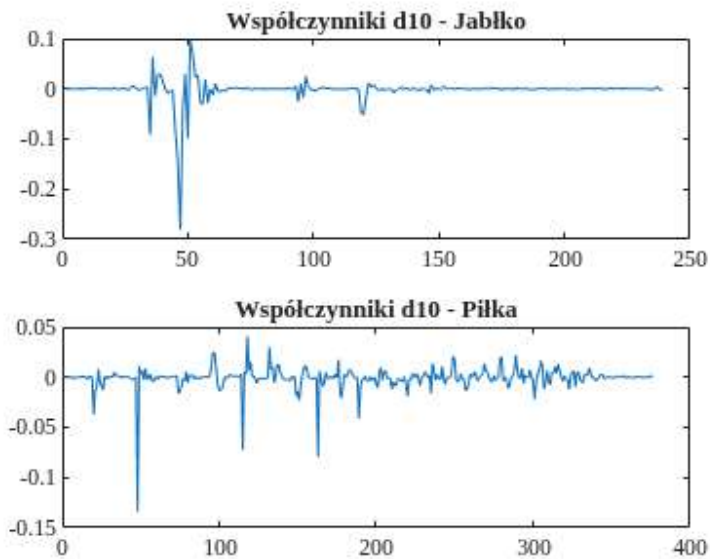
% d10
[c_jablko, l_jablko] = wavedec(signal_jablko, 10, 'haar');
d10_jablko = detcoef(c_jablko, l_jablko, 10);

[y_pilka, Fs_pilka] = audioread("pilka.aac");
signal_pilka = y_pilka(:,1);

% d10
```

```
[c_pilka, l_pilka] = wavedec(signal_pilka, 10, 'haar');
d10_pilka = detcoef(c_pilka, l_pilka, 10);

figure;
subplot(2,1,1); plot(d10_jablko); title('Współczynniki d10 - Jabłko');
subplot(2,1,2); plot(d10_pilka); title('Współczynniki d10 - Piłka');
```



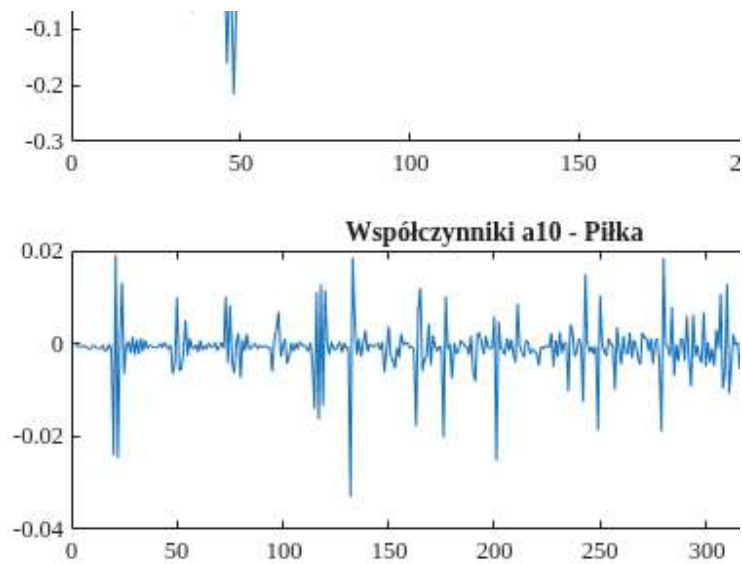
Wavedec to funkcja dekompozycji sygnału na c (współczynniki) oraz j (strukturę).

Appcoef to współczynniki aproksymacji, wymaga podania typu falki, w przeciwieństwie do detcoeff.

2)

```
a10_jablko = appcoef(c_jablko, l_jablko, 'haar', 10);
a10_pilka = appcoef(c_pilka, l_pilka, 'haar', 10);

figure;
subplot(2,1,1); plot(a10_jablko); title('Współczynniki a10 - Jabłko');
subplot(2,1,2); plot(a10_pilka); title('Współczynniki a10 - Piłka');
```

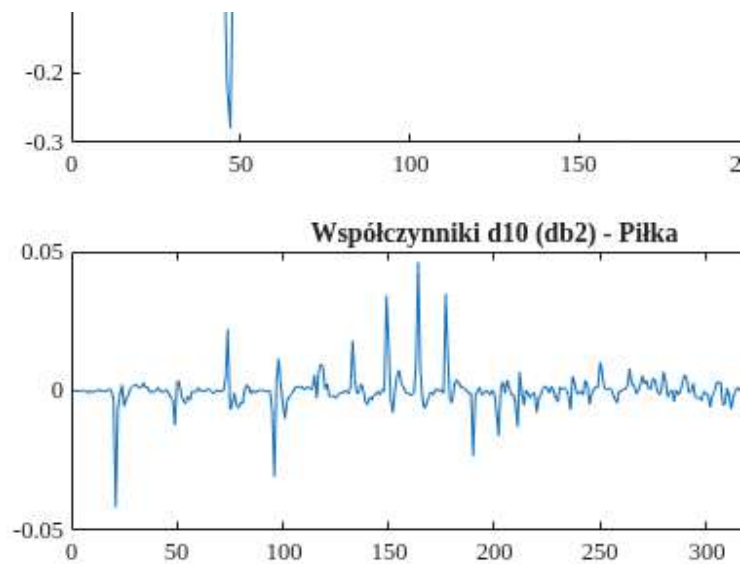


3)

```
[c_j, l_j] = wavedec(signal_jablko, 10, 'db2');
d10_jablko_db2 = detcoef(c_j, l_j, 10);

[c_p, l_p] = wavedec(signal_pilka, 10, 'db2');
d10_pilka_db2 = detcoef(c_p, l_p, 10);

figure;
subplot(2,1,1); plot(d10_jablko_db2); title('Współczynniki d10 (db2) - Jabłko');
subplot(2,1,2); plot(d10_pilka_db2); title('Współczynniki d10 (db2) - Piłka');
```



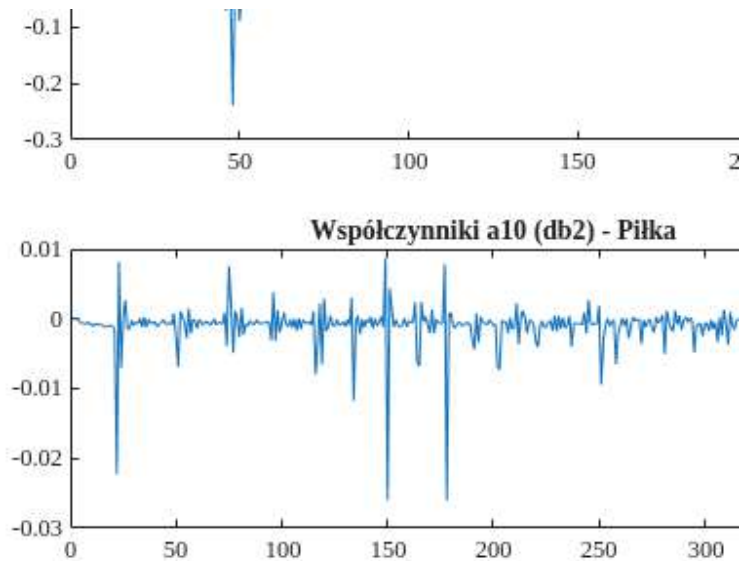
4)

```

a10_jablko = appcoef(c_j, l_j, 'db2', 10);
a10_pilka = appcoef(c_p, l_p, 'db2', 10);

figure;
subplot(2,1,1); plot(a10_jablko); title('Współczynniki a10 (db2) - Jabłko');
subplot(2,1,2); plot(a10_pilka); title('Współczynniki a10 (db2) - Piłka');

```



5)

```

% wiatrak
[y_wiatrak, Fs_wiatrak] = audioread("wiatrak_20.wav");
signal_wiatrak = y_wiatrak(:,1);

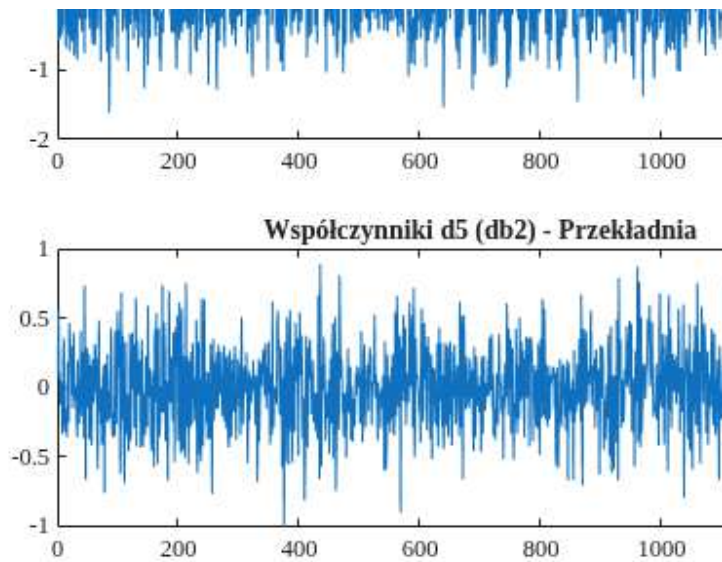
% d5
[c_wiatrak, l_wiatrak] = wavedec(signal_wiatrak, 10, 'db2');
d5_wiatrak = detcoef(c_wiatrak, l_wiatrak, 5);

% przekladania
[y_przekladnia, Fs_przekladnia] = audioread("przekladnia20.wav");
signal_przekladnia = y_przekladnia(:,1);

% d5
[c_przekladnia, l_przekladnia] = wavedec(signal_przekladnia, 10, 'db2');
d5_przekladnia = detcoef(c_przekladnia, l_przekladnia, 5);

figure;
subplot(2,1,1); plot(d5_wiatrak); title('Współczynniki d5 (db2) - Wiatrak');
subplot(2,1,2); plot(d5_przekladnia); title('Współczynniki d5 (db2) - Przekładnia');

```

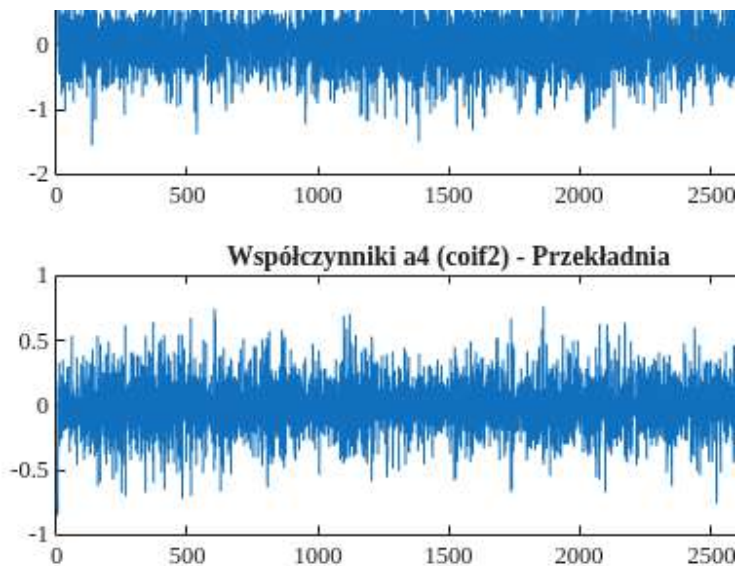


6)

```
[c_wiatrak, l_wiatrak] = wavedec(signal_wiatrak, 10, 'coif2');
[c_przekladnia, l_przekladnia] = wavedec(signal_przekladnia, 10, 'coif2');

a4_wiatrak = appcoef(c_wiatrak, l_wiatrak, 'coif2', 4);
a4_przekladnia = appcoef(c_przekladnia, l_przekladnia, 'coif2', 4);

figure;
subplot(2,1,1); plot(a4_wiatrak); title('Współczynniki a4 (coif2) - Wiatrak');
subplot(2,1,2); plot(a4_przekladnia); title('Współczynniki a4 (coif2) - Przekładnia');
```



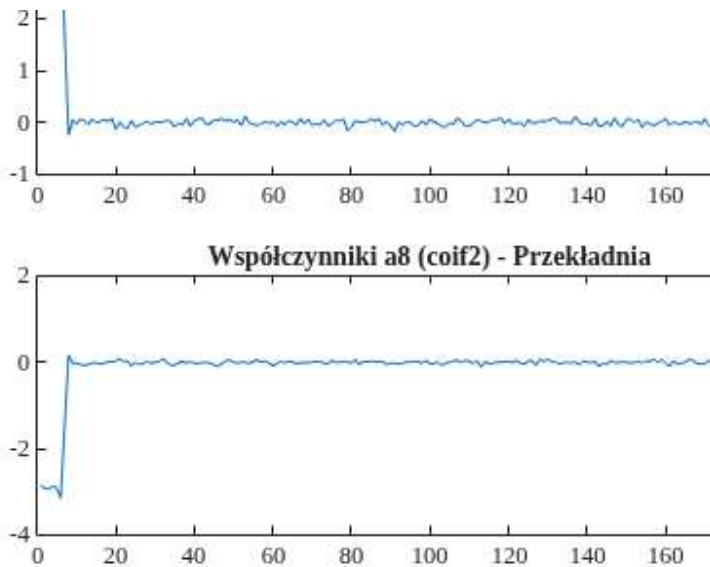
7)

```

a8_wiatrak = appcoef(c_wiatrak, l_wiatrak, 'coif2', 8);
a8_przekladnia = appcoef(c_przekladnia, l_przekladnia, 'coif2', 8);

figure;
subplot(2,1,1); plot(a8_wiatrak); title('Współczynniki a8 (coif2) - Wiatrak');
subplot(2,1,2); plot(a8_przekladnia); title('Współczynniki a8 (coif2) - Przekładnia');

```



8)

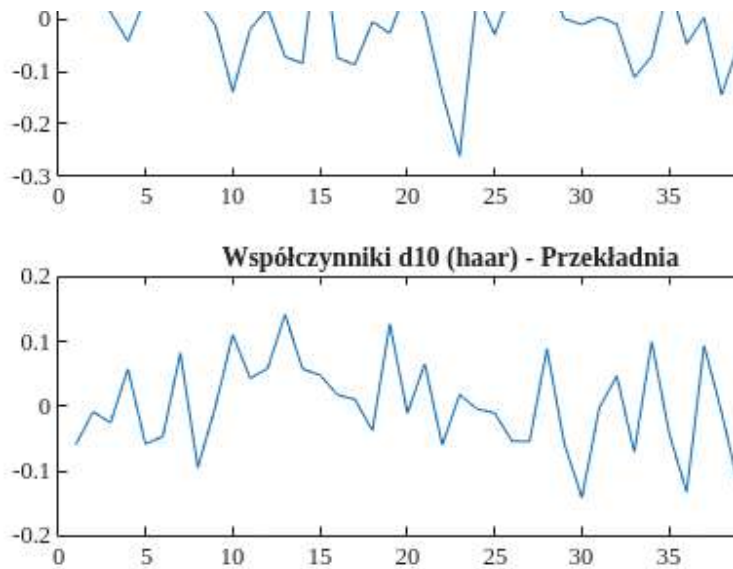
```

[c_wiatrak, l_wiatrak] = wavedec(signal_wiatrak, 10, 'haar');
[c_przekladnia, l_przekladnia] = wavedec(signal_przekladnia, 10, 'haar');

d10_wiatrak = detcoef(c_wiatrak, l_wiatrak, 10);
d10_przekladnia = detcoef(c_przekladnia, l_przekladnia, 10);

figure;
subplot(2,1,1); plot(d10_wiatrak); title('Współczynniki d10 (haar) - Wiatrak');
subplot(2,1,2); plot(d10_przekladnia); title('Współczynniki d10 (haar) - Przekładnia');

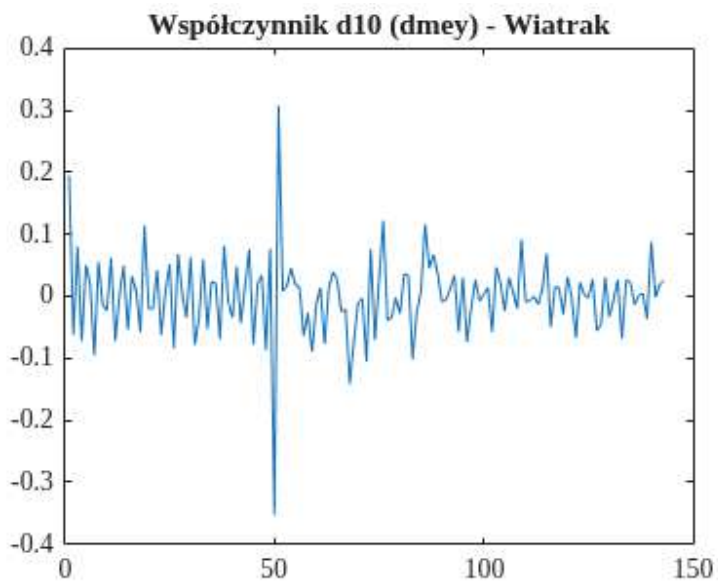
```



Widać, że wektor sygnału jest coraz krótszy i rzadszy - oznacza to wzrost rozdzielczości częstotliwościowej wyższych współczynników, obarczone jest to natomiast redukcją długości sygnału w dziedzinie czasu (o połowę za każdy wzrost współczynnika) - czyli obniżenia rozdzielczości czasowej.

9)

```
[c_wiatrak, l_wiatrak] = wavedec(signal_wiatrak, 10, 'dmey');
d10_wiatrak = detcoef(c_wiatrak, l_wiatrak, 10);
figure();
plot(d10_wiatrak); title('Współczynnik d10 (dmey) - Wiatrak');
```

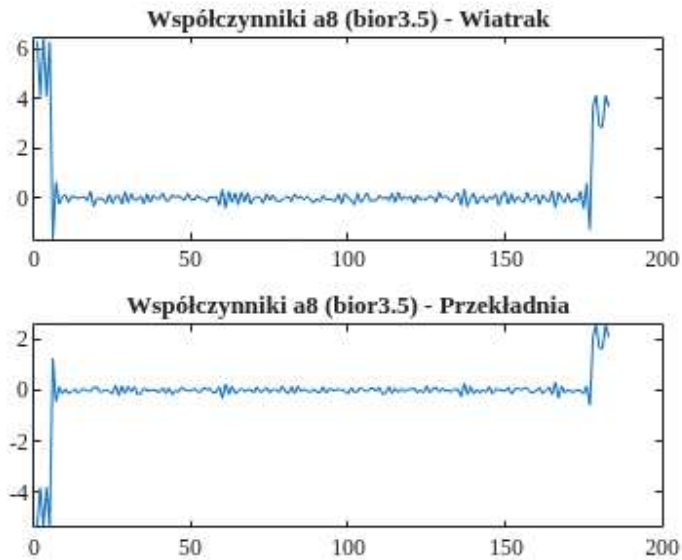


10)

```
[c_wiatrak, l_wiatrak] = wavedec(signal_wiatrak, 10, 'bior3.5');
[c_przekladnia, l_przekladnia] = wavedec(signal_przekladnia, 10, 'bior3.5');

a8_wiatrak = appcoef(c_wiatrak, l_wiatrak, 'bior3.5', 8);
a8_przekladnia = appcoef(c_przekladnia, l_przekladnia, 'bior3.5', 8);

figure;
subplot(2,1,1); plot(a8_wiatrak); title('Współczynniki a8 (bior3.5) - Wiatrak');
subplot(2,1,2); plot(a8_przekladnia); title('Współczynniki a8 (bior3.5) - Przekładnia');
```



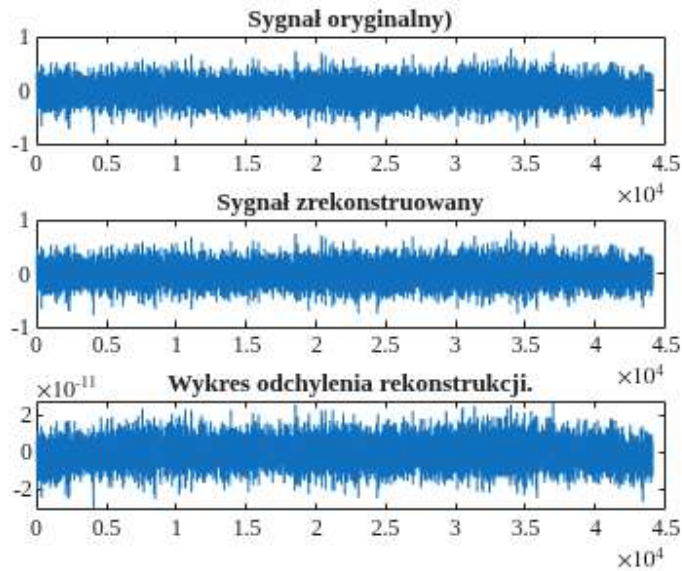
11)

```
[c_wiatrak, l_wiatrak] = wavedec(signal_wiatrak, 12, 'coif2');
signal_wiatrak_rec = waverec(c_wiatrak, l_wiatrak, 'coif2');
figure();

subplot(3,1,1);
plot(signal_wiatrak);
title('Sygnał oryginalny');

subplot(3,1,2);
plot(signal_wiatrak_rec);
title('Sygnał zrekonstruowany');

subplot(3,1,3);
plot(signal_wiatrak - signal_wiatrak_rec);
title('Wykres odchylenia rekonstrukcji.');
```

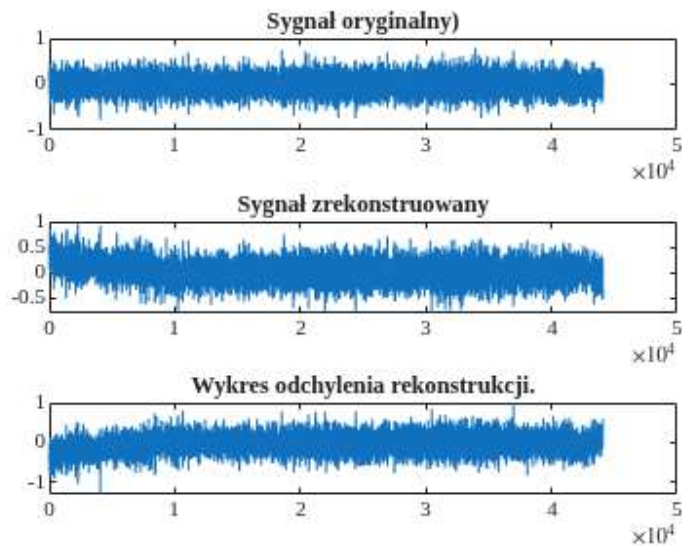


```
blad_rekonstrukcji = norm(signal_wiatrak_rec-signal_wiatrak)
```

```
blad_rekonstrukcji =  
1.3725e-09
```

12)

```
[c_wiatrak, l_wiatrak] = wavedec(signal_wiatrak, 12, 'coif2');  
signal_wiatrak_rec = waverec(c_wiatrak, l_wiatrak, 'haar');  
figure();  
  
subplot(3,1,1);  
plot(signal_wiatrak);  
title('Sygnał oryginalny');  
  
subplot(3,1,2);  
plot(signal_wiatrak_rec);  
title('Sygnał zrekonstruowany');  
  
subplot(3,1,3);  
plot(signal_wiatrak - signal_wiatrak_rec);  
title("Wykres odchylenia rekonstrukcji.")
```



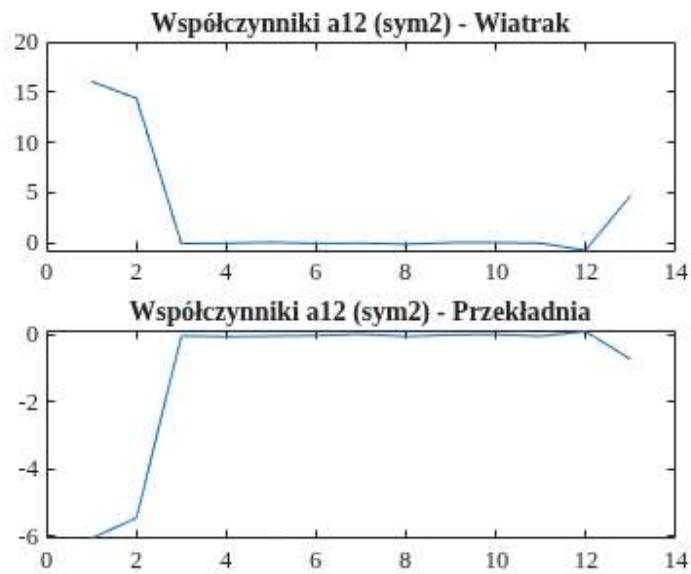
```
blad_rekonstrukcji = norm(signal_wiatrak_rec-signal_wiatrak)
```

```
blad_rekonstrukcji =  
43.4818
```

Widać, że użycie innej falki do rekonstrukcji w porównaniu do dekompozycji przyczynia się do powstania drastycznego błędu.

13)

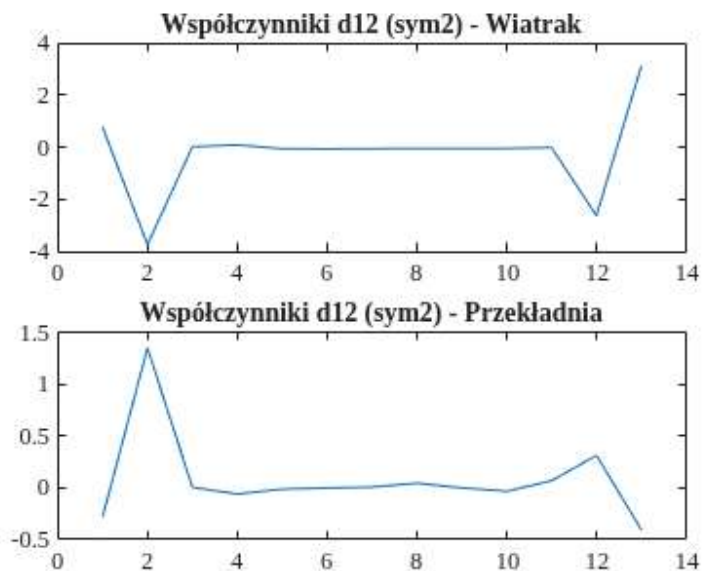
```
[c_wiatrak, l_wiatrak] = wavedec(signal_wiatrak, 12, 'sym2');  
[c_przekladnia, l_przekladnia] = wavedec(signal_przekladnia, 12, 'sym2');  
  
a12_wiatrak = appcoef(c_wiatrak, l_wiatrak, 'sym2', 12); % Ustawienie 12  
tutaj wymusza przynajmniej 12 poziom dekompozycji  
a12_przekladnia = appcoef(c_przekladnia, l_przekladnia, 'sym2', 12);  
  
figure();  
subplot(2,1,1); plot(a12_wiatrak); title('Współczynniki a12 (sym2) -  
Wiatrak');  
subplot(2,1,2); plot(a12_przekladnia); title('Współczynniki a12 (sym2) -  
Przekładnia');
```



14)

```
d12_wiatrak = detcoef(c_wiatrak, l_wiatrak, 12);
d12_przekladnia = detcoef(c_przekladnia, l_przekladnia, 12);

figure();
subplot(2,1,1); plot(d12_wiatrak); title('Współczynniki d12 (sym2) - Wiatrak');
subplot(2,1,2); plot(d12_przekladnia); title('Współczynniki d12 (sym2) - Przekładnia');
```



15)

```
A=[1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2];
B=[1 2 1 2 1 2 1 2 10 11 10 11 10 12 9 11];
```

```

C=[10 11 10 11 10 12 9 11 1 2 1 2 1 2 1 2];
D=[8 7 8 9 10 9 8 9 1 2 1 2 1 2 1 2 ];

[c,l] = wavedec(A, 3, 'sym2');
Aa3 = appcoef(c,l, 'sym2');

[c,l] = wavedec(B,3, 'sym2');
Ba3 = appcoef(c,l, 'sym2');

[c,l] = wavedec(C,3, 'sym2');
Ca3 = appcoef(c,l, 'sym2');

[c,l] = wavedec(D,3, 'sym2');
Da3 = appcoef(c,l, 'sym2');

Distance_Da3_Aa3= sum(abs(abs(Da3)-abs(Aa3)))

```

```

Distance_Da3_Aa3 =
48.6788

```

```

Distance_Da3_Ba3= sum(abs(abs(Da3)-abs(Ba3)))

```

```

Distance_Da3_Ba3 =
64.5450

```

```

Distance_Da3_Ca3= sum(abs(abs(Da3)-abs(Ca3)))

```

```

Distance_Da3_Ca3 =
18.2884

```

Sygnał D jest najbliższym sygnału C.

16)

```

A=[0 1 2 1 1 0 1 0 1 0 1 1 1 0 1 2];
B=[1 2 1 2 1 2 1 2 10 11 10 11 10 12 9 11];
C=[20 21 20 21 20 22 19 21 1 2 1 2 1 2 1 2];
D=[18 17 18 19 20 19 18 19 1 1 2 0 2 1 1 1];
E=[0 2 2 0 1 0 1 0 1 1 0 1 0 0 2 2];

[c,l] = wavedec(A, 3, 'coif2');
Aa3 = appcoef(c,l, 'coif2');

[c,l] = wavedec(B,3, 'coif2');
Ba3 = appcoef(c,l, 'coif2');

[c,l] = wavedec(C,3, 'coif2');
Ca3 = appcoef(c,l, 'coif2');

[c,l] = wavedec(D,3, 'coif2');
Da3 = appcoef(c,l, 'coif2');

[c,l] = wavedec(E,3, 'coif2');

```

```
Ea3 = appcoef(c,l, 'coif2');
```

```
Dist_D_A = sqrt(sum(abs(Da3 - Aa3).^2))
```

```
Dist_D_A =  
129.8137
```

```
Dist_D_B = sqrt(sum(abs(Da3 - Ba3).^2))
```

```
Dist_D_B =  
137.2344
```

```
Dist_D_C = sqrt(sum(abs(Da3 - Ca3).^2))
```

```
Dist_D_C =  
15.4261
```

```
Dist_E_A = sqrt(sum(abs(Ea3 - Aa3).^2))
```

```
Dist_E_A =  
1.1462
```

```
Dist_E_B = sqrt(sum(abs(Ea3 - Ba3).^2))
```

```
Dist_E_B =  
56.1534
```

```
Dist_E_C = sqrt(sum(abs(Ea3 - Ca3).^2))
```

```
Dist_E_C =  
144.9591
```

Sygnał D jest najbliższym sygnału C, a sygnał E najbliższym sygnału A.

17)

```
K1=[1 ; 2 ; 1];  
K2=[3 ; 2 ; 1];  
K3=[3 ; 3 ; 2];
```

```
% probka testowa
```

```
T1=[100 ; 200 ; 100];  
T2=[120 ; 210 ; 110];  
T3=[97 ; 185 ; 103];
```

```
[c,l] = wavedec(K1,2, 'haar');  
K1a = appcoef(c,l, 'haar'); % 2.5  
[c,l] = wavedec(K2,2, 'haar');  
K2a = appcoef(c,l, 'haar'); % 3.5  
[c,l] = wavedec(T1,2, 'haar');  
T1a = appcoef(c,l, 'haar'); % 250  
[c,l] = wavedec(T2,2, 'haar');  
T2a = appcoef(c,l, 'haar'); % 275  
[c,l] = wavedec(K3,2, 'haar');  
K3a = appcoef(c,l, 'haar'); % 5
```

```

[c,1] = wavedec(T3,2, 'haar');
T3a = appcoef(c,1, 'haar'); % 244

%Tworzymy macierz zbior uczacy
KT=[2.5 3.5 250 275];

% Macierz z wynikami na wyjście powinna wyglądać w ten sposób.
KTW=[1 1 0 0 ; 0 0 1 1 ; 0 0 0 0];

inputs = KT; % dane wejsciowe
targets = KTW; % dane wyjsciowe

% Create a Pattern Recognition Network
hiddenLayerSize = 10;
net = patternnet(hiddenLayerSize);

% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 90/100;
net.divideParam.valRatio = 5/100;
net.divideParam.testRatio = 5/100;

% Train the Network
[net,tr] = train(net,inputs,targets);

% Test the Network
outputs = net(inputs);
errors = gsubtract(targets,outputs);
performance = perform(net,targets,outputs)

% View the Network
view(net)
probkatestowa1 = vec2ind(net([K3a]))
probkatestowa2 = vec2ind(net([T3a]))

```

WNIOSKI

Falki (w przeciwieństwie do FFT) pozwoliły zlokalizować w czasie momenty wystąpienia określonych zdarzeń w sygnałach audio (wiatrak, przekładnia).

Wykazano, że wyzerowanie lub usunięcie współczynników detali (d1, d2) skutecznie wygładza sygnał, usuwając szum wysokoczęstotliwościowy.

Współczynniki aproksymacji (a3, a12) skutecznie służą jako skompresowane "podpisy" sygnałów, umożliwiając ich poprawną klasyfikację (np. odróżnienie sygnału D od A) przy użyciu prostych metryk odległości.

Potwierdzono, że transformacja jest bezstratnie odwracalna tylko wtedy, gdy do dekompozycji i rekonstrukcji używa się zgodnych falek (błąd bliski zeru). Użycie różnych falek generuje znaczące błędy rekonstrukcji.