

Metody numeryczne II.

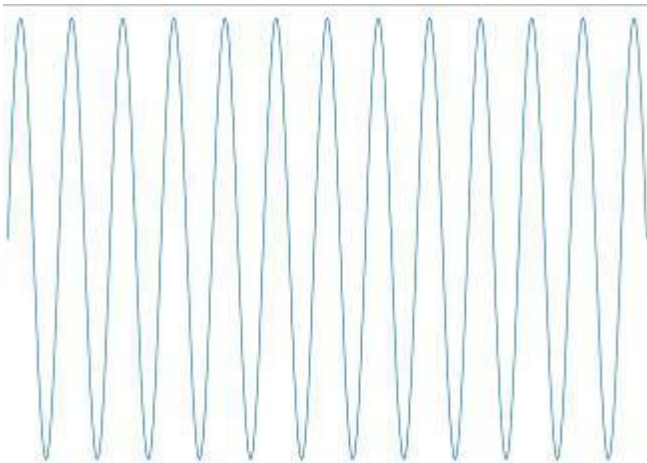
Aproksymacja falkowa

Oleksandr Sokolov

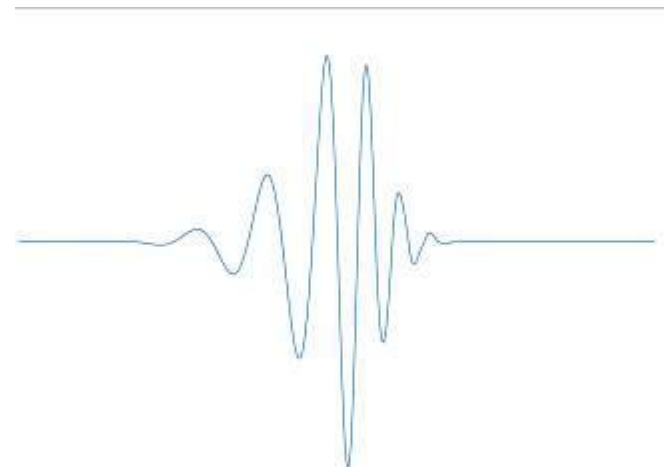
Wydział Fizyki, Astronomii i Informatyki Stosowanej UMK
<http://fizyka.umk.pl/~osokolov/MNII/>

Transformacja falkowa

Transformacja falkowa jest przekształceniem podobnym do transformacji Fouriera. Oba przekształcenia opierają się na wykorzystaniu operacji **iloczynu skalarnego** badanego sygnału $f(t)$ i pozostałej części, zwanej "**jądrem przekształcenia**". Główna różnica między tymi przekształceniami to właśnie owe jądro.

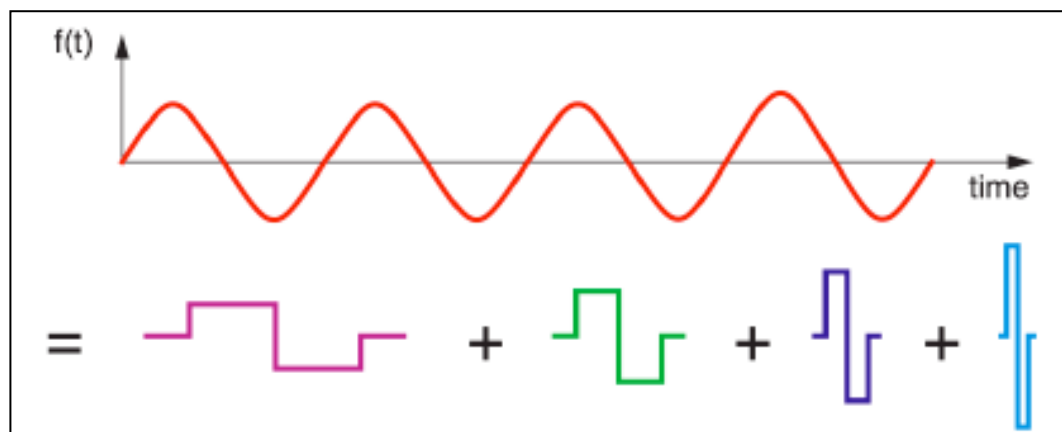
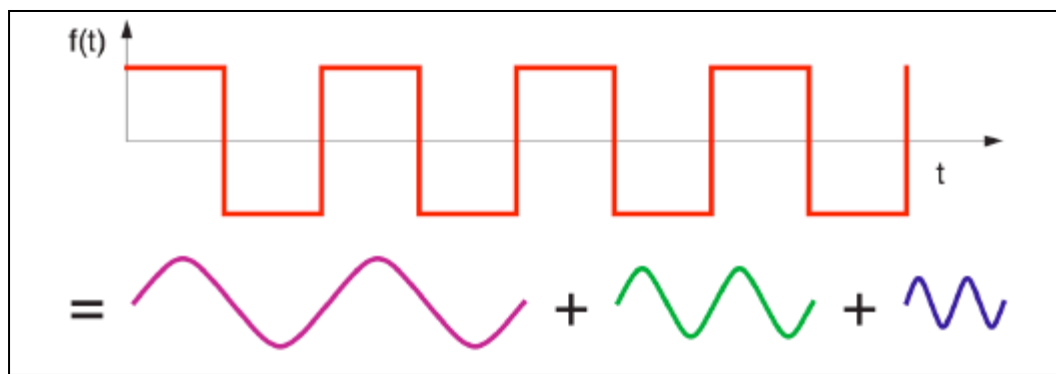


fala



falka

Transformacja Fouriera vs falkowa



$$P_n(x) = a_0\phi_0(x) + a_1\phi_1(x) + L + a_n\phi_n(x)$$

Aproksymacja szeregiem Fouriera

$$P_n(x) = a_0\phi_0(x) + a_1\phi_1(x) + \cdots + a_n\phi_n(x)$$

For any 2π periodical function $f(x)$:

$$f(x) = a_0 + \sum_{k=1}^{\infty} (a_k \cos kx + b_k \sin kx) \quad \infty \rightarrow M$$

$$a_0 = \frac{1}{2\pi} \int_0^{2\pi} f(x) dx$$

$$a_k = \frac{1}{\pi} \int_0^{2\pi} f(x) \cos(kx) dx$$

$$b_k = \frac{1}{\pi} \int_0^{2\pi} f(x) \sin(kx) dx$$

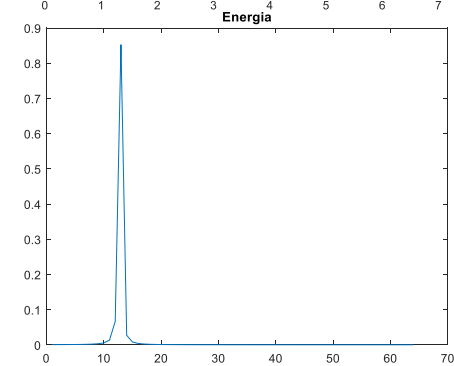
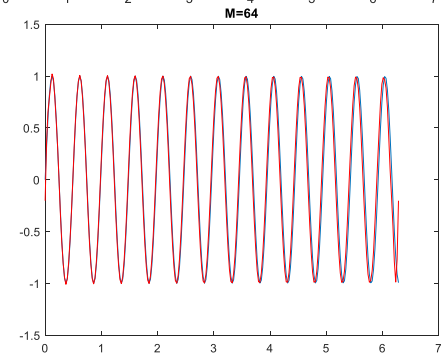
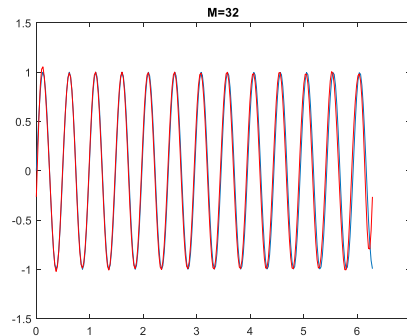
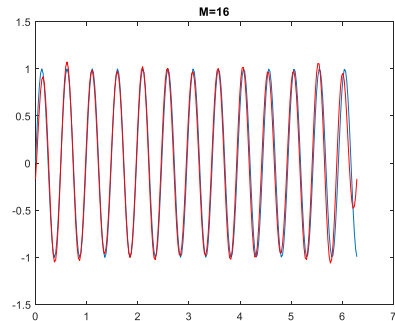
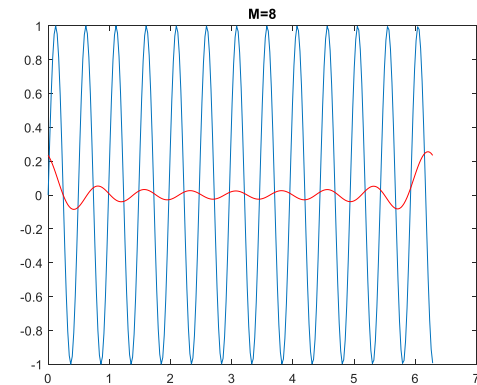
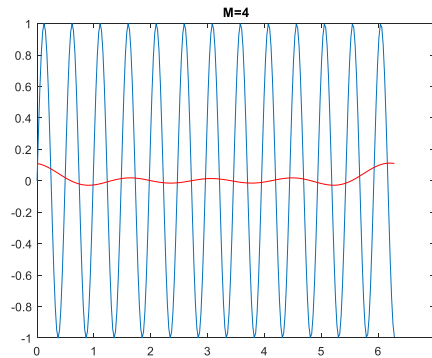
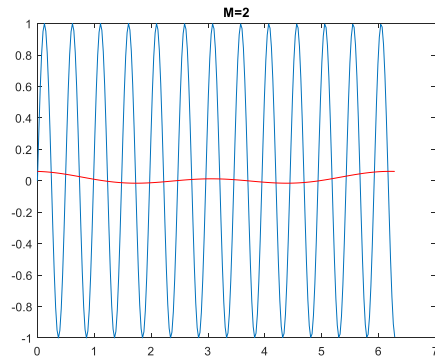
$$T = 2\pi$$

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left(a_n \cos\left(\frac{2n\pi}{T}x\right) + b_n \sin\left(\frac{2n\pi}{T}x\right) \right)$$

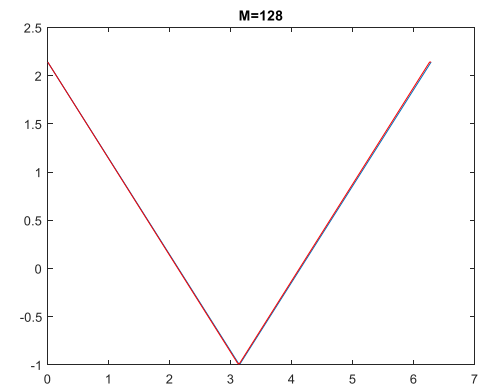
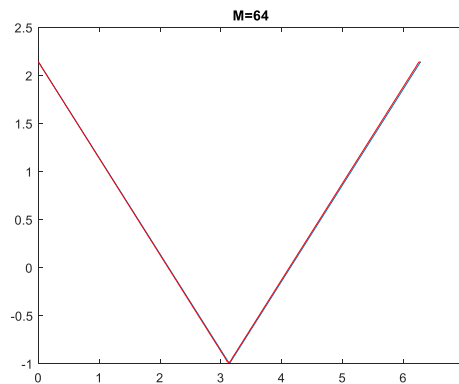
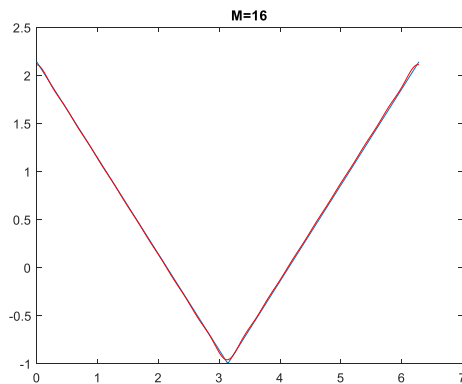
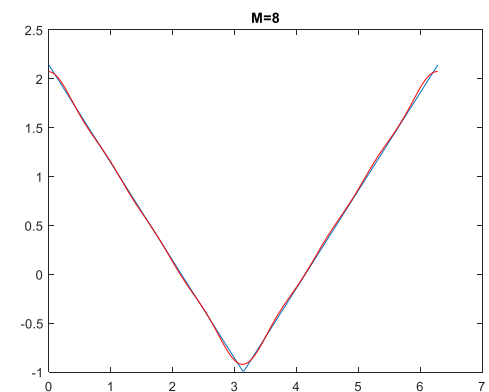
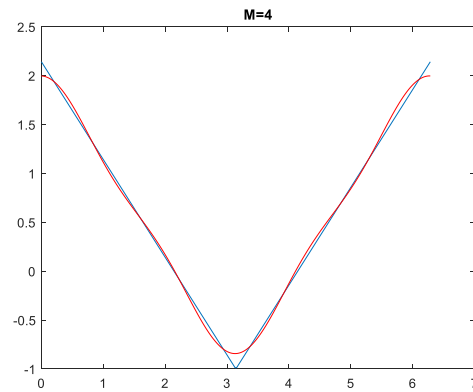
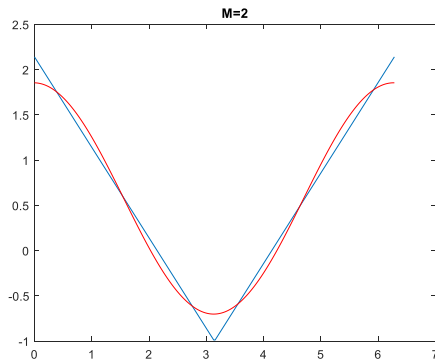
$$a_n = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(x) \cos\left(\frac{2n\pi}{T}x\right) dx, \quad n = 0, 1, 2, \dots$$
$$b_n = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(x) \sin\left(\frac{2n\pi}{T}x\right) dx, \quad n = 1, 2, 3, \dots$$

Aproksymacja szeregiem Fouriera (cd)

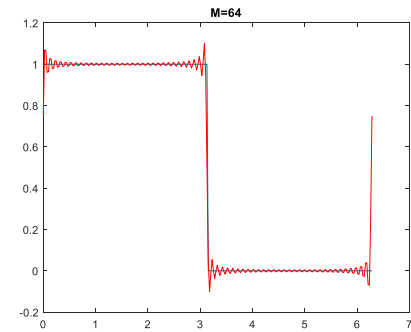
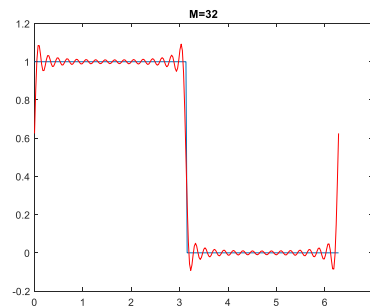
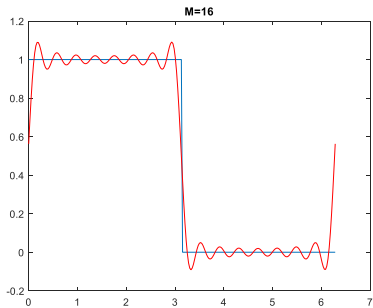
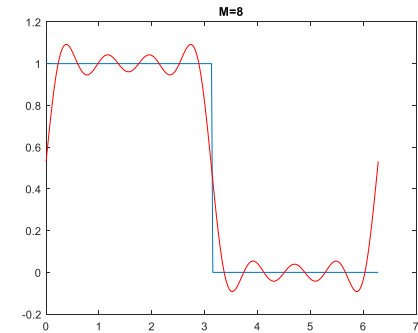
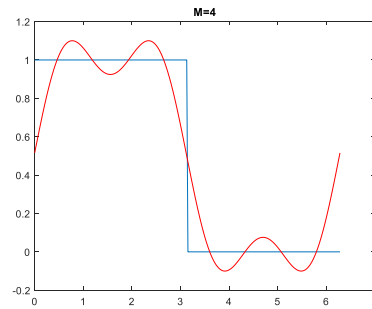
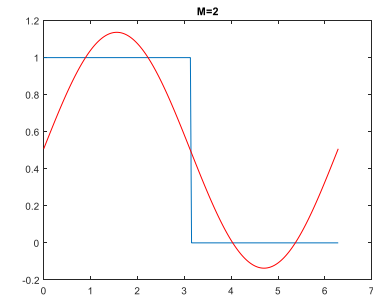
$$\infty \rightarrow M$$



Aproksymacja szeregiem Fouriera (cd)



Aproksymacja szeregiem Fouriera (cd)



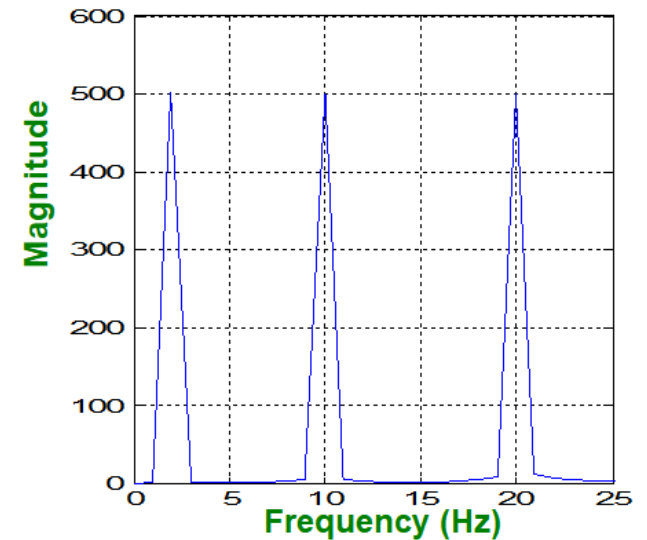
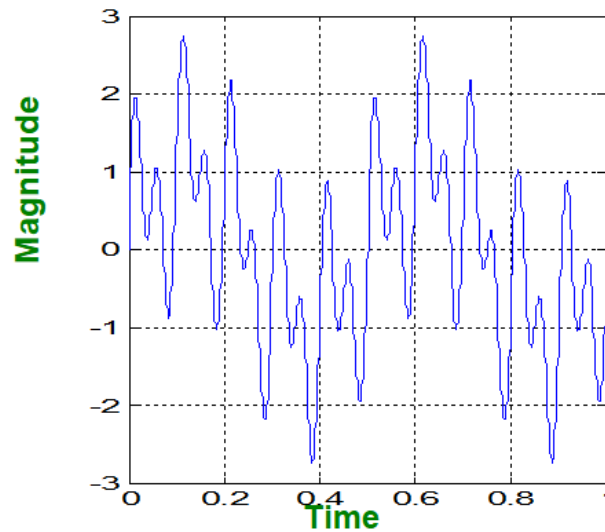
Sygnały stochastyczne

- Sygnały stacjonarne:
 - Statystyki nie zmieniają się z czasem
 - Zawartość częstotliwości nie zmienia się z czasem
 - Informacje nie zmieniają się z czasem
- Sygnały niestacjonarne:
 - Statystyki zmieniają się z czasem
 - Częstotliwości zmieniają się z czasem
 - Ilość informacji wzrasta

Rozkład Fouriera

2 Hz + 10 Hz + 20Hz

Stationary



$$f(t) = \sin(2t) + \sin(10t) + \sin(20t)$$

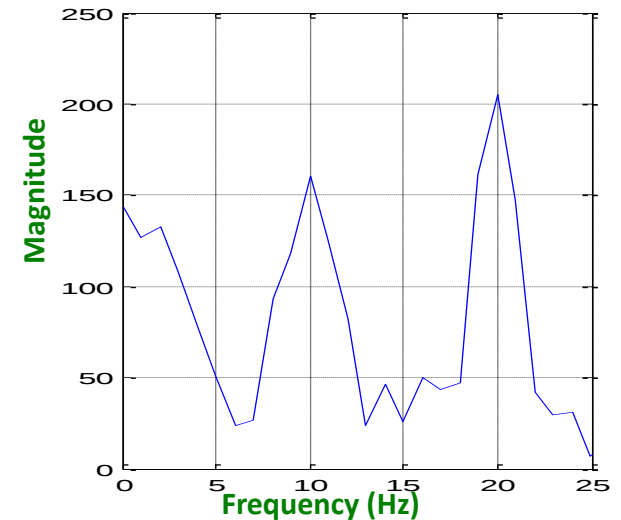
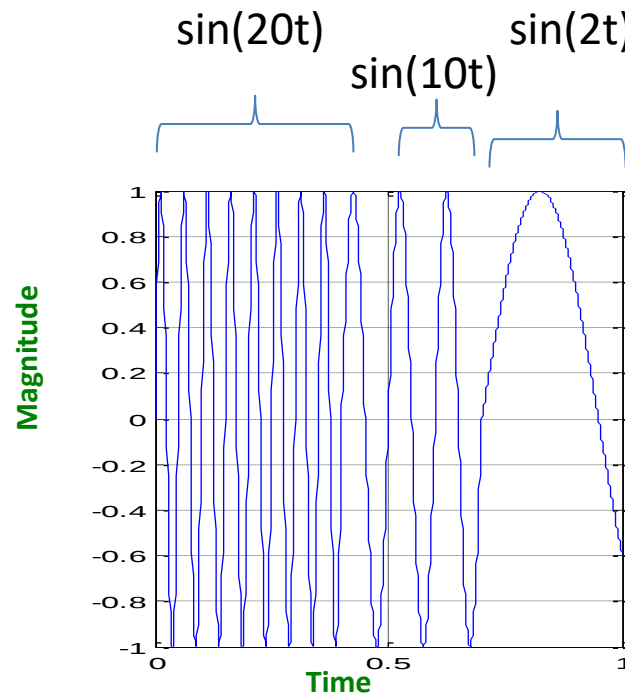
Sygnały niestacjonarne

Rozkład Fouriera

$$f(t) = \sin(2t) + \sin(10t) + \sin(20t)$$

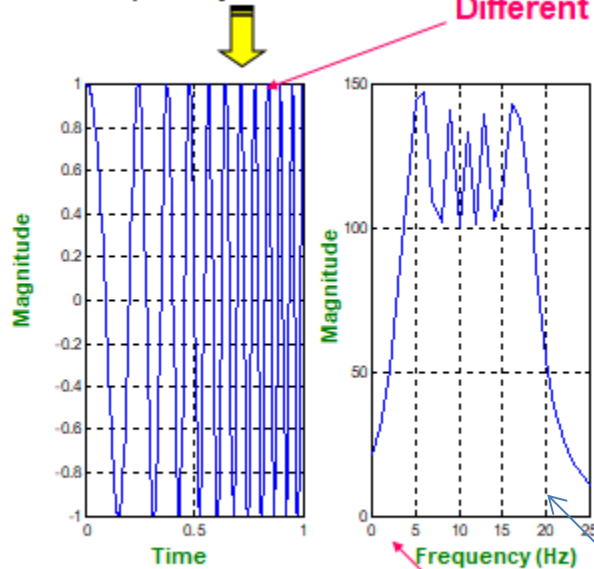
0.0-0.4: 20 Hz +
0.4-0.7: 10 Hz +
0.7-1.0: 2Hz

Non-
Stationary



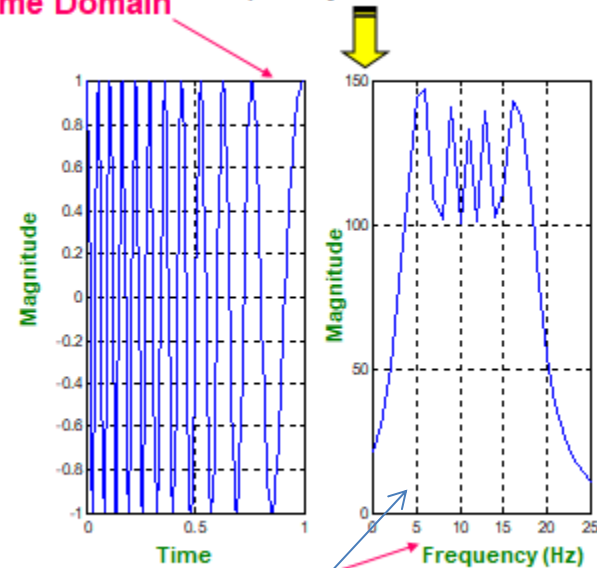
Sygnały niestacjonarne(cd)

■ Frequency: 2 Hz to 20 Hz

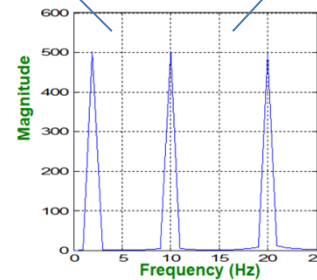
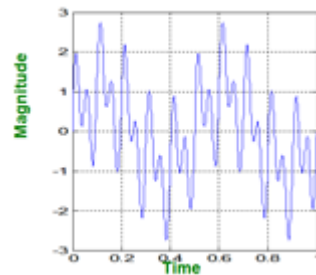


Different in Time Domain

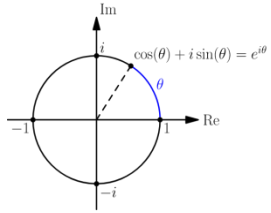
■ Frequency: 20 Hz to 2 Hz



Same in Frequency Domain



Transformacja Fouriera a transformacja falkowa



$$F(\omega) = \int_{-\infty}^{\infty} f(t) \exp(-j\omega t) dt \longleftrightarrow f(t) = \int_{-\infty}^{\infty} F(\omega) e^{j2\pi\omega t} d\omega$$

$$Wf(s, u) = \int_{-\infty}^{\infty} f(t) \frac{1}{\sqrt{s}} \psi^* \left(\frac{t-u}{s} \right) dt$$



$$f(t) = \frac{1}{C_\psi} \int_0^\infty \int_{-\infty}^\infty Wf(s, u) \frac{1}{\sqrt{s}} \psi \left(\frac{t-u}{s} \right) du \frac{ds}{s^2}$$

$$C_\psi = \int_0^\infty \frac{|\Psi(\omega)|^2}{\omega} d\omega < \infty$$

$\Psi(\omega)$

transformacja
Fouriera dla funkcji

$\psi(t)$

Transformacja Wavelet

Równanie:

$$W_f(s, u) = \int_{-\infty}^{\infty} f(t) \frac{1}{\sqrt{s}} \psi^* \left(\frac{t-u}{s} \right) dt$$

- $W_f(s, u)$: Wynik transformacji Wavelet, funkcja zależna od dwóch parametrów:
 - s : Skala (scale), parametr określający rozciąganie lub ściszenie funkcji falki (wavelet) ψ . Mniejsza wartość s oznacza wyższe częstotliwości (szczegółowe cechy sygnału), większa wartość s odpowiada niższym częstotliwościom (ogólne trendy).
 - u : Przesunięcie (translation), parametr określający przesunięcie w czasie lub przestrzeni funkcji falki wzdłuż osi t . Określa, w którym miejscu sygnału $f(t)$ analizowana jest funkcja falki.
- $f(t)$: Oryginalny sygnał lub funkcja wejściowa, która jest analizowana za pomocą transformacji Wavelet. Jest to funkcja czasu t .
- $\frac{1}{\sqrt{s}}$: Czynniki normalizacyjny, który zapewnia, że energia funkcji falki pozostaje stała przy zmianie skali s . Dzielenie przez $\sqrt{s}$ kompensuje zmianę amplitudy przy rozciąganiu lub ściszeniu.
- $\psi^* \left(\frac{t-u}{s} \right)$: Skoniugowana funkcja falki (wavelet) ψ , przesunięta o u i rozciągnięta/ściśniona o czynnik s . Symbol $*$ oznacza sprzężenie zespolone. Funkcja $\psi(t)$ to funkcja macierzysta (mother wavelet), która musi spełniać pewne warunki (np. mieć zerową średnią i być dobrze zlokalizowana w czasie i częstotliwości).
- dt : Element różniczkowy integracji, wskazujący, że całkowanie odbywa się po zmiennej t od $-\infty$ do ∞ .

Odwrotna transformacja Wavelet

Równanie:

$$f(t) = \frac{1}{C_\psi} \int_0^\infty \int_{-\infty}^\infty W_f(s, u) \frac{1}{\sqrt{s}} \psi\left(\frac{t-u}{s}\right) \frac{ds}{s^2} du$$

- $f(t)$: Odtworzony sygnał, który jest przywracany z transformacji Wavelet.
- $\frac{1}{C_\psi}$: Stała normalizacyjna, która zapewnia poprawność odwrotnej transformacji. Jest definiowana poniżej i zależy od transformacji Fouriera funkcji ψ .
- $W_f(s, u)$: Wynik transformacji Wavelet, jak powyżej.
- $\frac{1}{\sqrt{s}}$: Ten sam czynnik normalizacyjny co w transformacji bezpośredniej, zapewniający spójność energii.
- $\psi\left(\frac{t-u}{s}\right)$: Funkcja falki (bez sprzężenia), przesunięta o u i rozciągnięta/ścisniona o s .
- $\frac{ds}{s^2} du$: Element różniczkowy podwójnej całki. du odpowiada integracji po przesunięciu u , a $\frac{ds}{s^2}$ jest specyficznym elementem różniczkowym dla skali s , wynikającym z faktu, że transformacja Wavelet działa w skali logarytmicznej.
- Całkowanie: Zewnętrzna całka po s od 0 do ∞ (zakres skal), wewnętrzna po u od $-\infty$ do ∞ (zakres przesunięć).

Definicja stałej normalizacyjnej

Równanie:

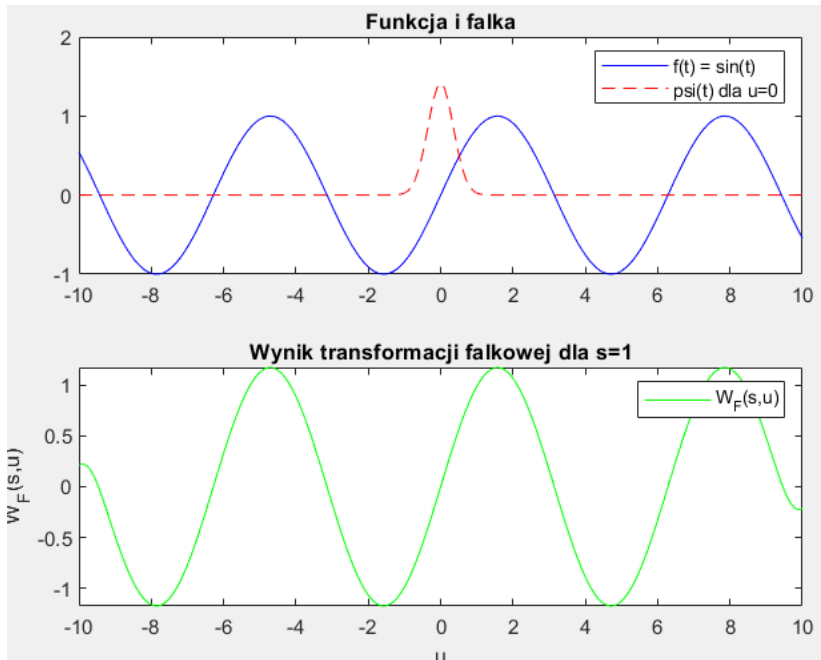
$$C_\psi = \int_0^\infty \frac{|\Psi(\omega)|^2}{\omega} d\omega < \infty$$

- C_ψ : Stała normalizacyjna, która musi być skończona, aby transformacja Wavelet była odwracalna. Zależy od transformacji Fouriera funkcji ψ .
- $\Psi(\omega)$: Transformata Fouriera funkcji macierzystej $\psi(t)$, gdzie ω jest częstotliwością. $|\Psi(\omega)|^2$ to moduł kwadratowy transformacji Fouriera, reprezentujący rozkład energii w dziedzinie częstotliwości.
- $\frac{|\Psi(\omega)|^2}{\omega}$: Wyrażenie w integrandzie, które musi być integrable, aby C_ψ było skończone. Wymaga to, aby $\psi(t)$ miało odpowiednie własności w dziedzinie częstotliwości (np. zerową wartość w $\omega = 0$).
- $d\omega$: Element różniczkowy integracji po częstotliwości ω od 0 do ∞ .
- Warunek $< \infty$: Gwarantuje, że transformacja Wavelet jest stabilna i odwracalna. Funkcja $\psi(t)$ musi być wybrana tak, aby ten warunek był spełniony (np. fale Morleta czy Daubechies).

Przykład

$$Wf(s,u) = \int_{-\infty}^{\infty} f(t) \frac{1}{\sqrt{s}} \psi^* \left(\frac{t-u}{s} \right) dt$$

Numeryczne.m



```
% Parametry
t = -10:0.1:10; % zakres czasu
s = 1/2; % skala
dt = t(2) - t(1); % krok czasowy

% Funkcja f(t) = sin(t)
f = sin(t);

% Zakres przesunięć u
u_values = -10:0.1:10;
W_F_values = zeros(size(u_values));

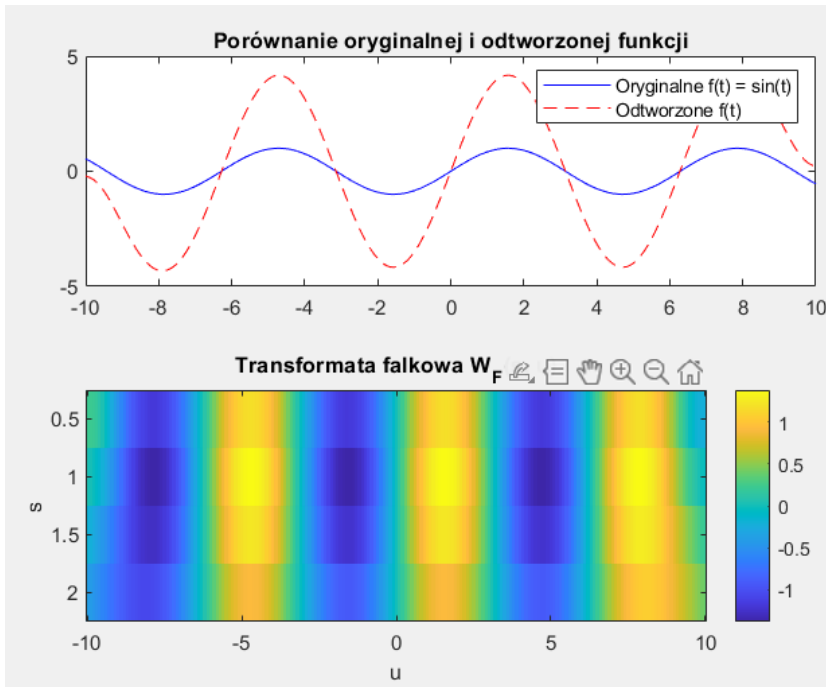
% Obliczanie W_F(s,u) dla różnych u
for i = 1:length(u_values)
    u = u_values(i);
    psi = exp(-((t-u)/s).^2) / sqrt(s); % falka
    W_F_values(i) = sum(f .* psi) * dt; % transformacja falkowa
end
```

Przykład

$$f(t) = \frac{1}{C_\psi} \int_0^\infty \int_{-\infty}^\infty W f(s, u) \frac{1}{\sqrt{s}} \psi\left(\frac{t-u}{s}\right) du \frac{ds}{s^2}$$

NumerycznaOdwrotna.m

$$C_\psi = \int_0^\infty \frac{|\Psi(\omega)|^2}{\omega} d\omega < \infty$$



```
% Krok 2: Odwrotna transformacja falkowa
f_reconstructed = zeros(size(t));
C_psi = 1; % uproszczenie, w praktyce trzeba obliczyć C_psi
for k = 1:length(t)
    t_k = t(k);
    integral_sum = 0;
    for i = 1:length(s_values)
        s = s_values(i);
        for j = 1:length(u_values)
            u = u_values(j);
            psi = (1/sqrt(s)) * psi_func((t_k-u)/s);
            integral_sum = integral_sum + W_F(i, j) * psi * (du * ds / s^2);
        end
    end
    f_reconstructed(k) = (1/C_psi) * integral_sum;
end
```

Cpsi.m, CPsiGaus.m

Całka C_ψ może być trudna do analitycznego obliczenia, ale numeryczne podejście powinno dać wynik bliski wartości teoretycznej (dla Gaussowskiej falki C_ψ jest dobrze zdefiniowane i wynosi około $2\sqrt{\pi} \approx 3.5449$).

Metody numeryczne

$$Wf(s, u) = \int_{-\infty}^{\infty} f(t) \frac{1}{\sqrt{s}} \psi^* \left(\frac{t-u}{s} \right) dt$$

a) Dyskretyzacja sygnału i parametrów

- Sygnał $f(t)$ w praktyce jest dyskretny (np. próbkowany w czasie), więc $f(t)$ reprezentujemy jako wektor wartości $f[n]$ w punktach t_n , gdzie $t_n = n \cdot \Delta t$, a Δt to krok czasowy.
- Parametry s (skala) i u (przesunięcie) również muszą być dyskretne:
 - Skala s jest wybierana jako ciąg wartości (np. logarytmicznie: $s = \{s_1, s_2, \dots, s_M\}$).
 - Przesunięcie u odpowiada próbkom czasowym sygnału: $u = t_n$.

b) Przybliżenie funkcji falki

- Funkcja falki $\psi(t)$ musi być zdefiniowana. Popularne wybory to falka Morleta, Haar lub Daubechies. W MATLAB-ie można użyć wbudowanych funkcji falek (np. `morlet`, `mexihat`) lub zdefiniować własną.
- Dla każdej skali s i przesunięcia u , obliczamy $\psi^* \left(\frac{t-u}{s} \right)$.

Metody numeryczne

$$f(t) = \frac{1}{C_\psi} \int_0^\infty \int_{-\infty}^\infty W_f(s, u) \frac{1}{\sqrt{s}} \psi\left(\frac{t-u}{s}\right) du \frac{ds}{s^2}$$

c) Numeryczne całkowanie

- Całka w transformacji Wavelet $W_f(s, u) = \int_{-\infty}^\infty f(t) \frac{1}{\sqrt{s}} \psi^*\left(\frac{t-u}{s}\right) dt$ jest aproksymowana jako suma:

$$W_f(s, u) \approx \sum_n f(t_n) \frac{1}{\sqrt{s}} \psi^*\left(\frac{t_n - u}{s}\right) \Delta t$$

- W MATLAB-ie można to zrobić za pomocą mnożenia macierzowego, co jest efektywne dla dużych danych.

d) Odwrotna transformacja Wavelet

- Odwrotna transformacja wymaga podwójnej całki po s i u . Numerycznie aproksymujemy to jako podwójną sumę:

$$f(t) \approx \frac{1}{C_\psi} \sum_s \sum_u W_f(s, u) \frac{1}{\sqrt{s}} \psi\left(\frac{t-u}{s}\right) \frac{\Delta s}{s^2} \Delta u$$

- Stała C_ψ również musi być obliczona numerycznie, jeśli nie znamy jej analitycznie.

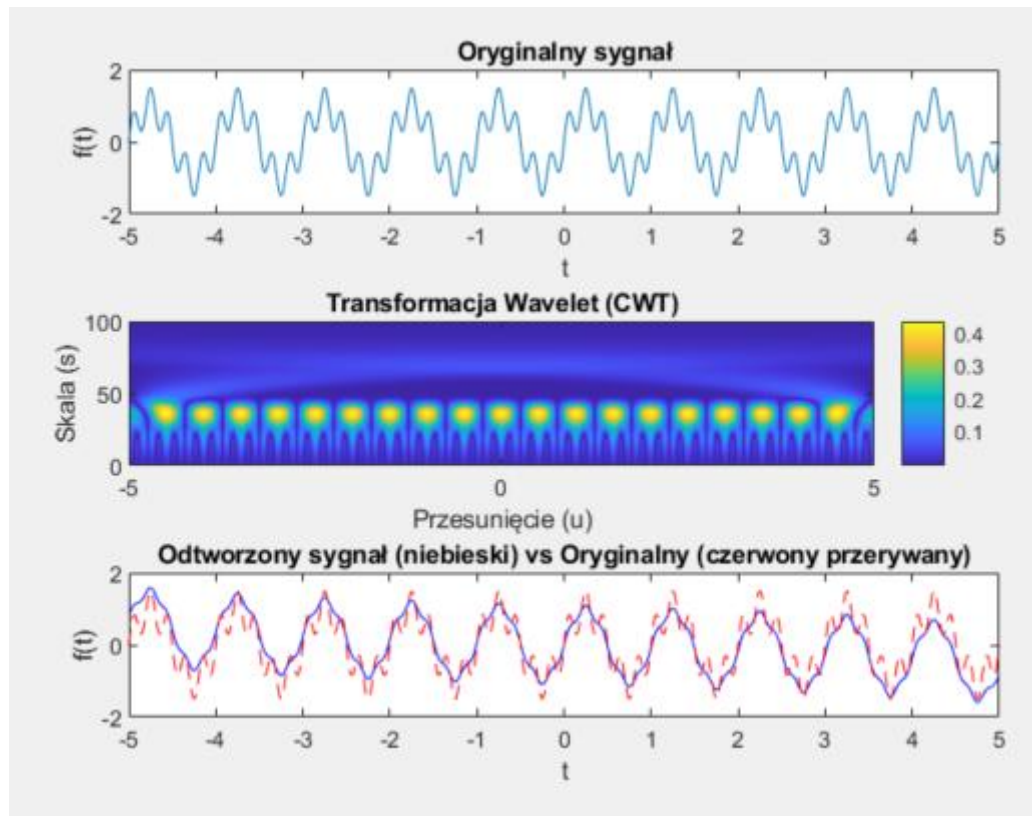
Metody numeryczne

$$C_\psi = \int_0^\infty \frac{|\Psi(\omega)|^2}{\omega} d\omega < \infty$$

e) Obliczenie C_ψ

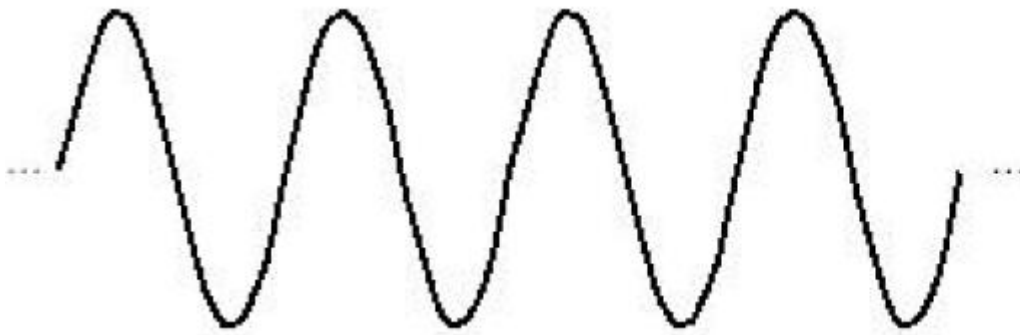
- $C_\psi = \int_0^\infty \frac{|\Psi(\omega)|^2}{\omega} d\omega$ wymaga transformacji Fouriera funkcji falki $\psi(t)$, co w MATLAB-ie można zrobić za pomocą `fft`.
- Następnie numerycznie integrujemy po częstotliwościach dodatnich.

Dowavelet_transform.m



Fala a falka

- Falka jest kształtem fali o ograniczonym okresie, którego średnia wartość jest równa zero.



Sinusoidy są płynne i przewidywalne, a falki zwykle są nieregularne i asymetryczne

Właściwości falek

zerowa wartość średnia

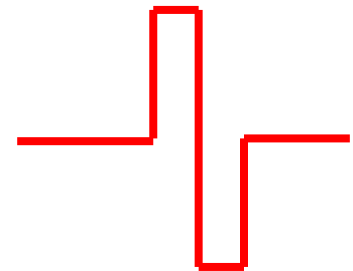
$$\int_{-\infty}^{+\infty} \psi(t) dt = 0$$

normalizacja

$$\| \psi \| = 1$$

skupiona wokół $t=0$

skończone pasmo przenoszenia



Ciągła Transformacja Falkowa

Continuous Wavelet Transform (CWT)

- Definicja:

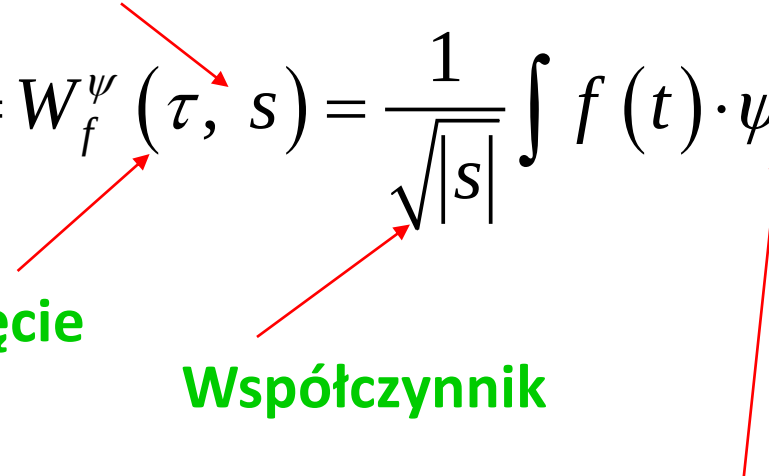
Skalowanie

$$\text{CWT}_f^\psi(\tau, s) = W_f^\psi(\tau, s) = \frac{1}{\sqrt{|s|}} \int f(t) \cdot \psi\left(\frac{t - \tau}{s}\right) dt$$

Przesunięcie

Współczynnik normalizujący

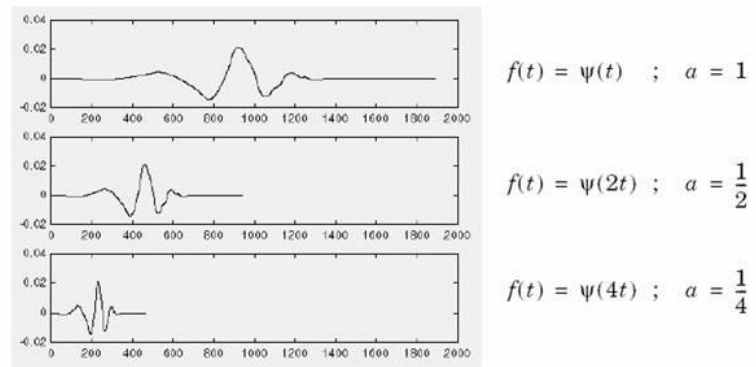
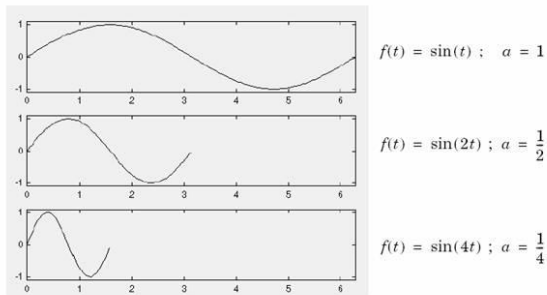
Funkcja macierzysta

A diagram with four red arrows pointing from text labels to specific parts of the CWT equation. One arrow points from 'Skalowanie' to the denominator 'sqrt(|s|)'. Another points from 'Przesunięcie' to the parameter 'tau' in the argument of the wavelet function. A third points from 'Współczynnik normalizujący' to the '1/sqrt(|s|)' factor. The fourth points from 'Funkcja macierzysta' to the wavelet function 'psi'.

Skalowanie

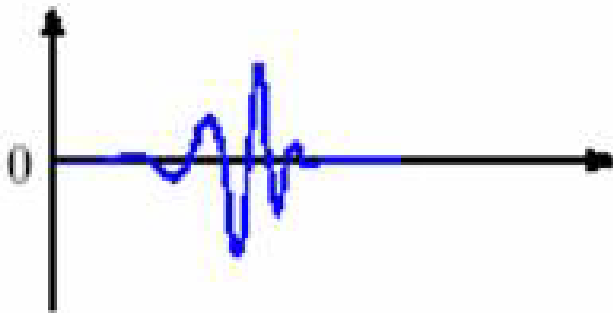
$$\text{CWT}_f^\psi(\tau, s) = W_f^\psi(\tau, s) = \frac{1}{\sqrt{|s|}} \int f(t) \cdot \psi\left(\frac{t-\tau}{s}\right) dt$$

- Skalowanie falki oznacza jej rozciąganie (lub ściskanie). Współczynnik skali(scale factor), zaznaczymy literą **$a=1/s$** .

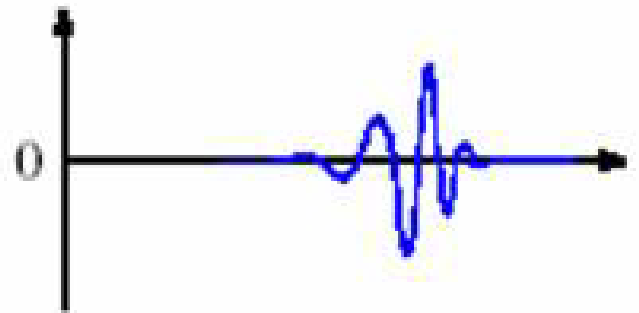


Przesunięcie

- Przesunięcie falki oznacza opóźnienie (lub przyspieszenie) jej początku. Matematycznie opóźnienie funkcji $f(t)$ o k jest reprezentowane przez $f(t-k)$.



Funkcja falkowa $\psi(t)$



Przesunięta funkcja falkowa $\psi(t-k)$

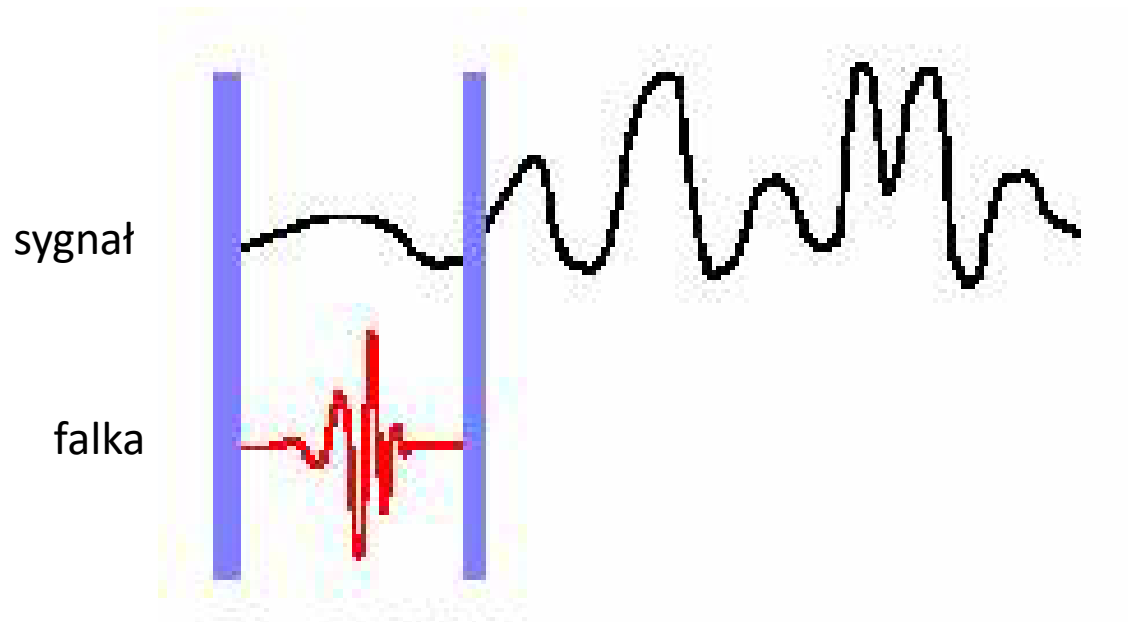
Cechy CWT

- **Ciągła** transformata falkowa jest sumą (całką) **po całym czasie** sygnału **zwielokrotnionego** o przeskalowaną, przesuniętą wersję falki.
- Proces ten daje w wyniku współczynniki falkowe, które są **funkcjami skali i położenia**.

$$\text{CWT}_f^\psi(\tau, s) = W_f^\psi(\tau, s) = \frac{1}{\sqrt{|s|}} \int f(t) \cdot \psi\left(\frac{t - \tau}{s}\right) dt$$

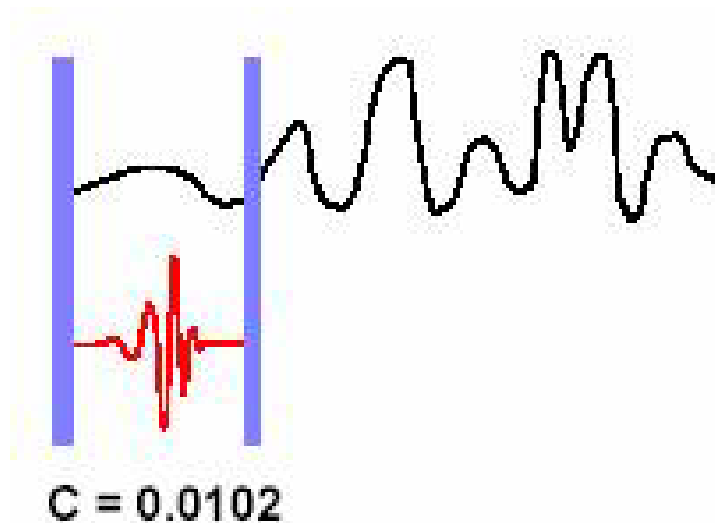
Krok 1

- Weźmy falkę i porównajmy ją do początkowej części oryginalnego sygnału.



Krok 2

- Obliczmy liczbę **C**, która określa jak bardzo falka jest skorelowana z wybranym fragmentem sygnału.



$C = 0.0102$ to wartość współczynnika korelacji, który mierzy, jak bardzo falka przypomina sygnał w tym oknie. Wartość bliska 0 (jak tutaj) oznacza słabą korelację – falka i sygnał niezbyt do siebie pasują w tym fragmencie.

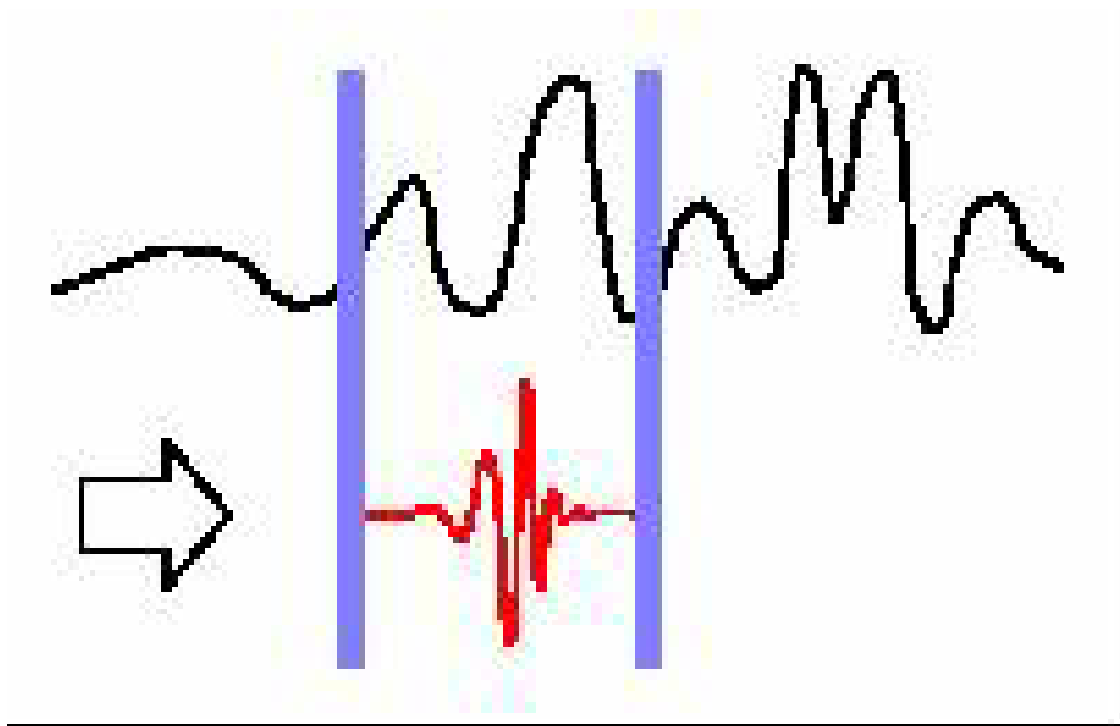
W transformacji falkowej taki współczynnik jest obliczany jako iloczyn skalarny między falką (przeskalowaną i przesuniętą) a sygnałem, znormalizowany przez energie obu funkcji. Wzór ogólny to:

$$C = \frac{\int f(t) \psi_{s,u}(t) dt}{\sqrt{\int f(t)^2 dt \cdot \int \psi_{s,u}(t)^2 dt}}$$

gdzie $f(t)$ to sygnał, $\psi_{s,u}(t)$ to falka przeskalowana o s i przesunięta o u .

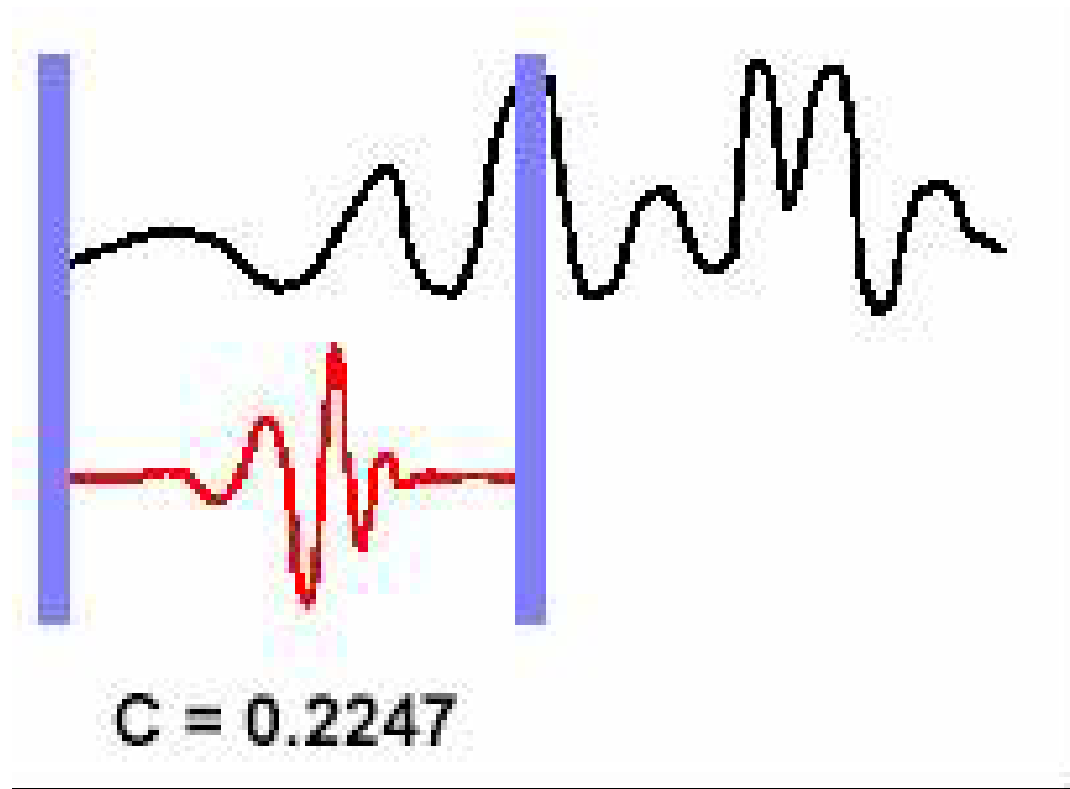
Krok 3

- Przesuńmy falę w prawo i powtórzmy kroki **1** i **2** dopóki nie przejdziemy całego sygnału.



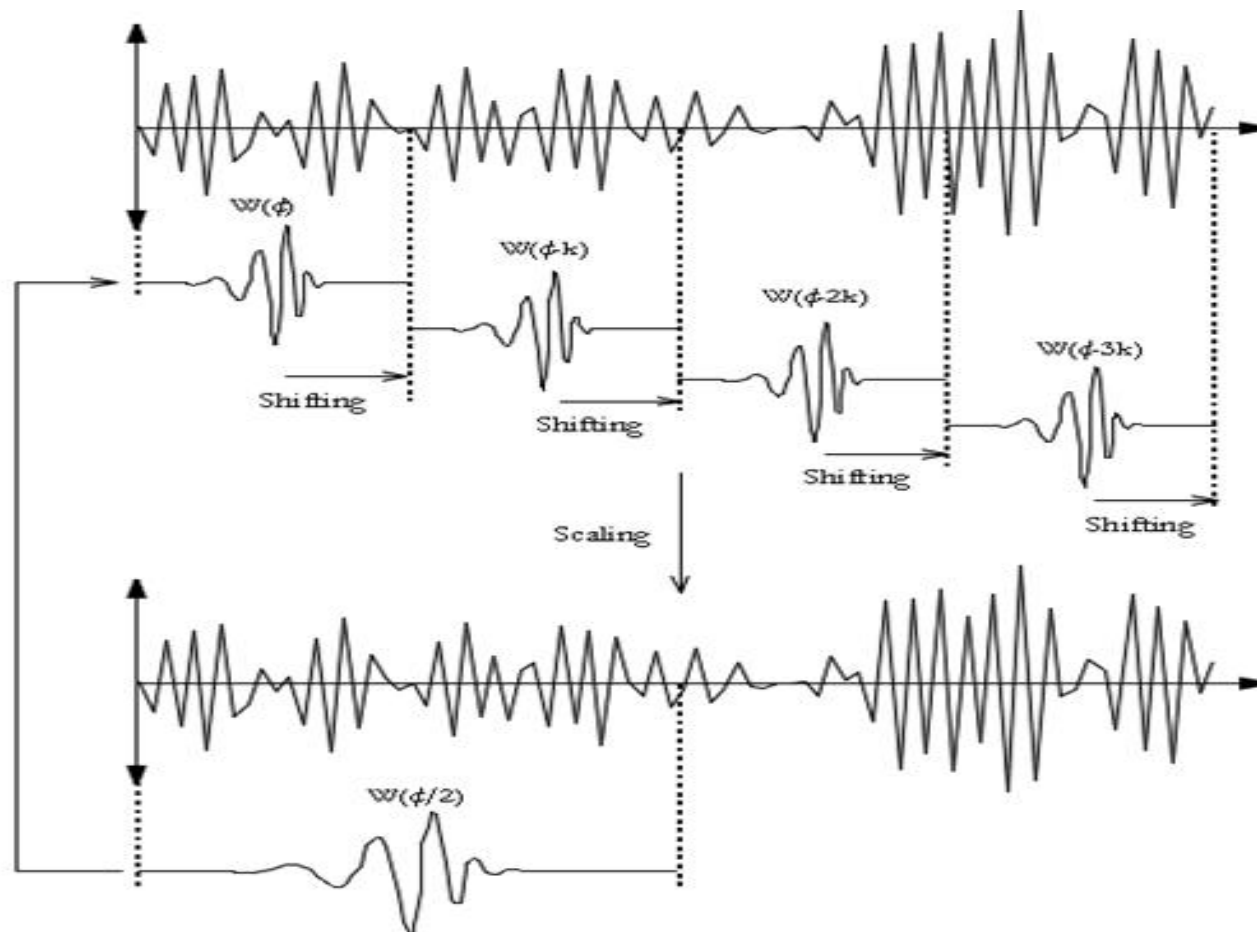
Krok 4

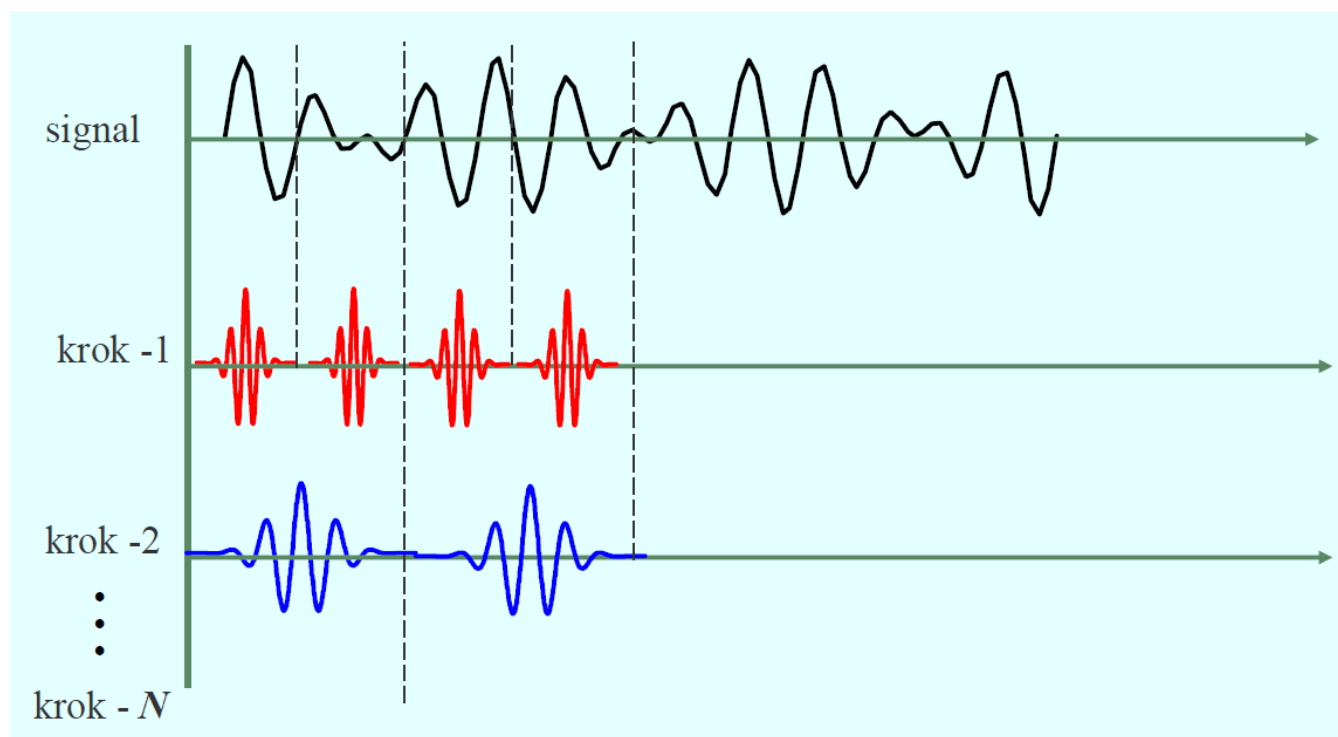
- Przeskalujemy (rozciągamy) falę i powtórzymy kroki od 1 do 3.



Krok 5

- Powtórzymy kroki 1 do 4 dla wszystkich skł.

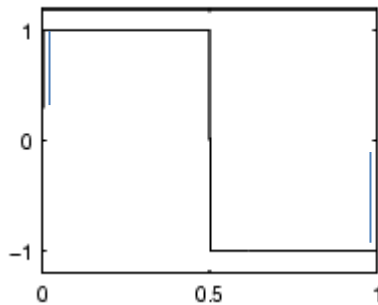




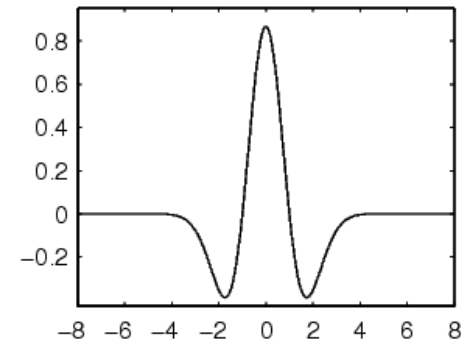
Falki CWT

<https://www.mathworks.com/help/wavelet/gs/introduction-to-the-wavelet-families.html>

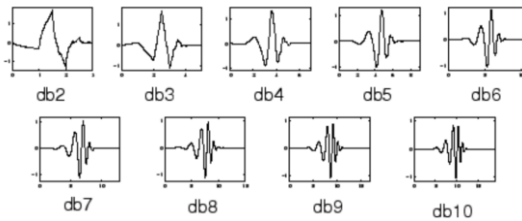
Falka Haara



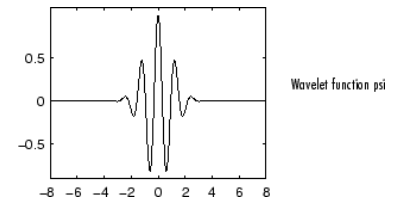
Kapelusz Meksykański



Falki Daubechies



Morlet



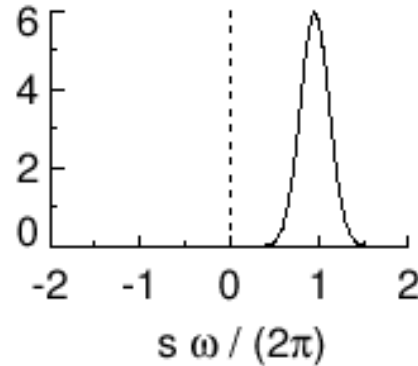
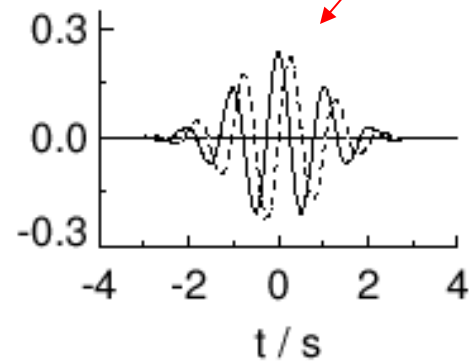
Falkę można skonstruować bezpośrednio pod tą lokalną cechę sygnału, który ma zostać wyodrębniony lub wykryty, jeśli jego forma jest znana a priori.

Wavelet basis functions

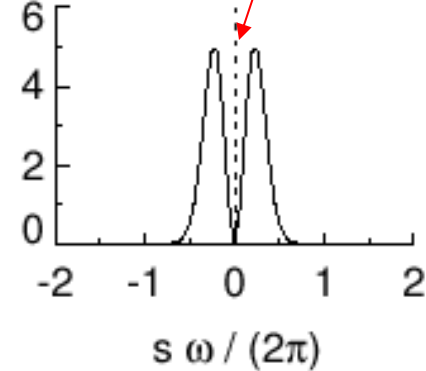
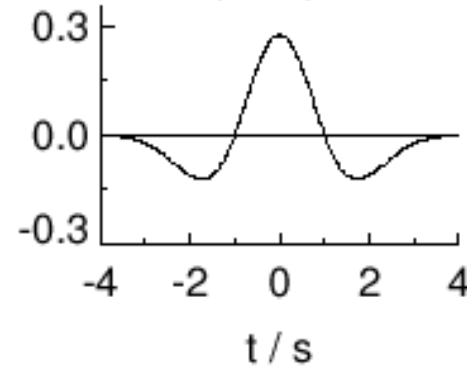
Frequency domain

Time domain

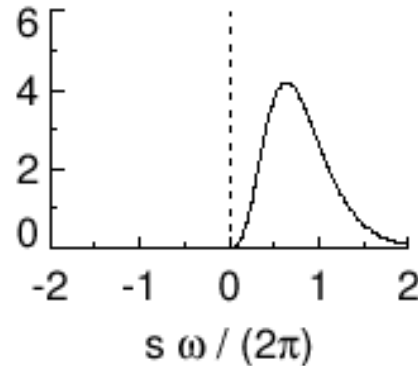
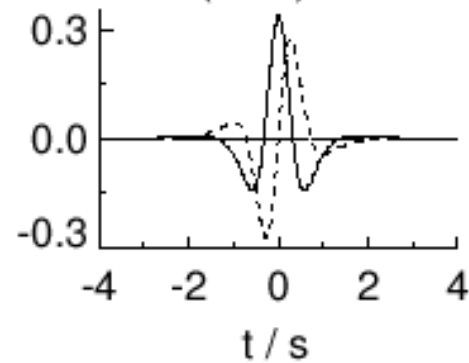
a. Morlet



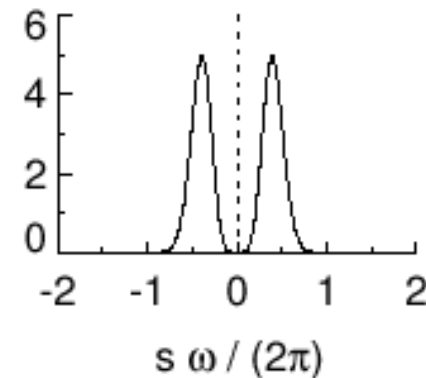
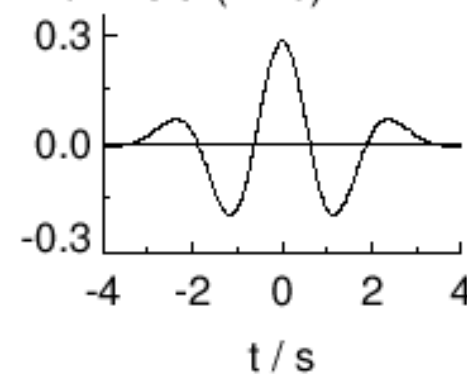
c. DOG (m=2)



b. Paul (m=4)

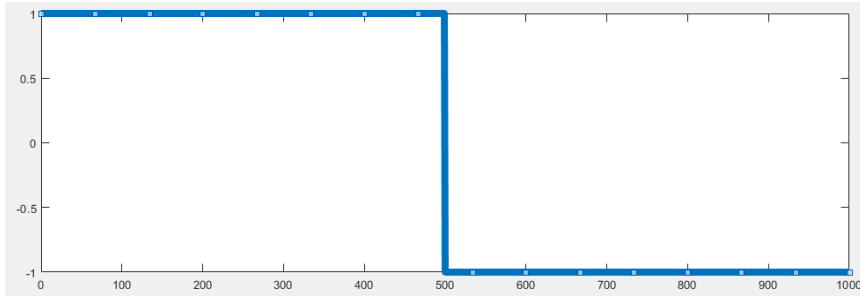


d. DOG (m=6)

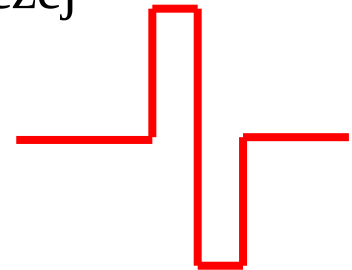


Falka Haara

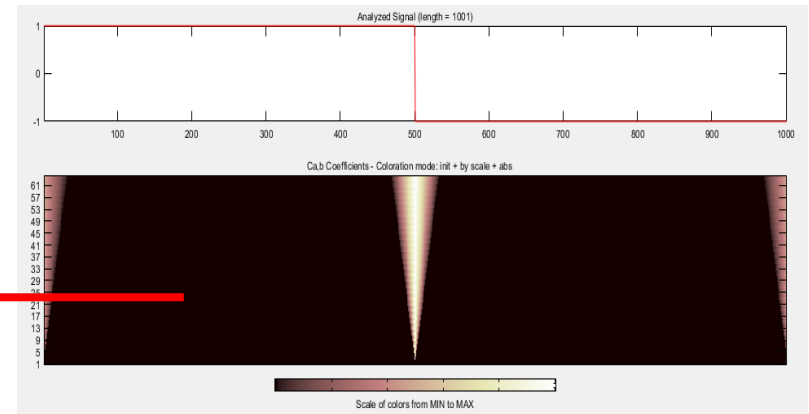
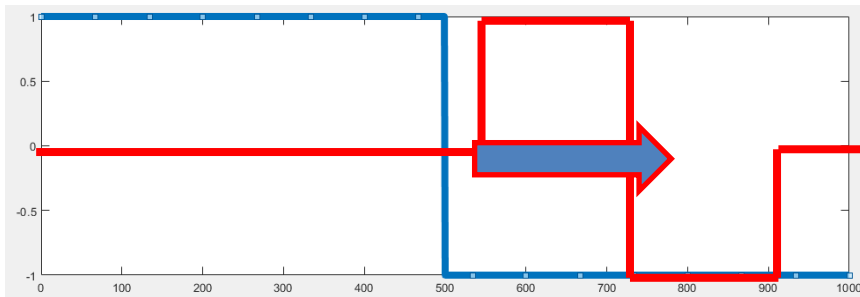
Funkcja $f(t)$



$$\psi(x) = \begin{cases} 1 & \text{jesli } 0 \leq x < \frac{1}{2} \\ -1 & \text{jesli } \frac{1}{2} \leq x < 1 \\ 0 & \text{inaczej} \end{cases}$$



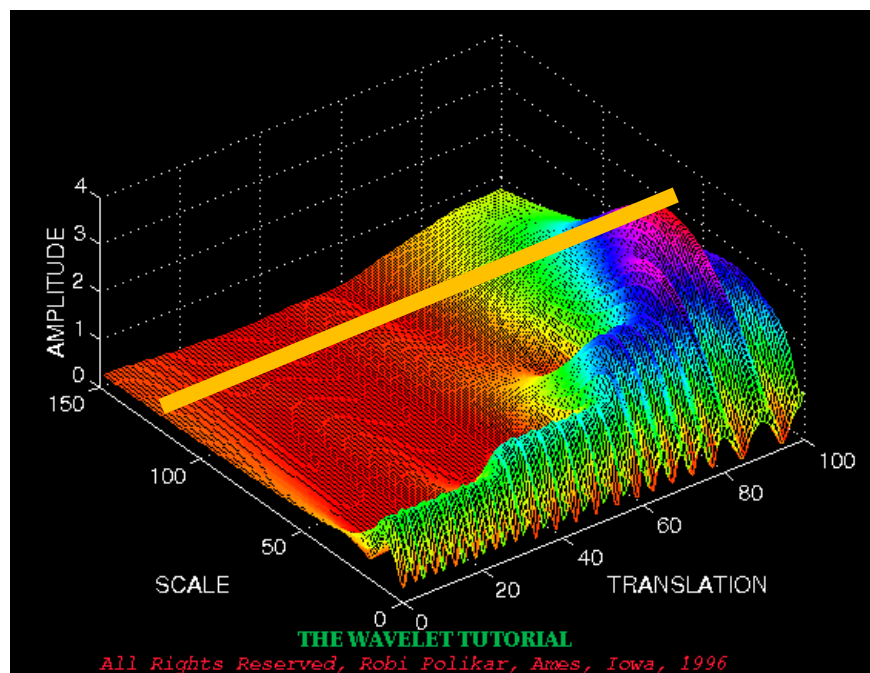
$$\text{CWT}_f^\psi(\tau, s) = W_f^\psi(\tau, s) = \frac{1}{\sqrt{|s|}} \int f(t) \cdot \psi\left(\frac{t-\tau}{s}\right) dt$$



ciemny kolor odpowiada przejściu sygnału przez zero a jasny - punktem ekstremum.

Jak odczytać znaczenie tych wszystkich współczynników?

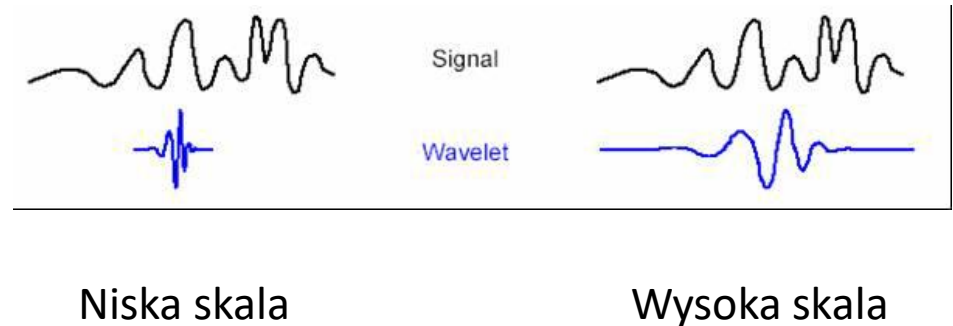
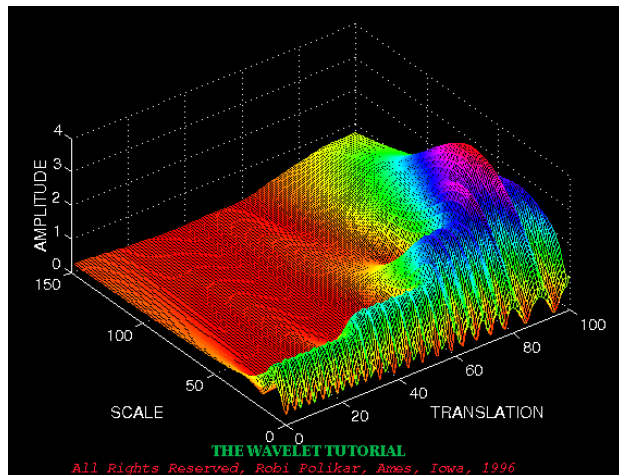
Musimy zrobić wykres, na którym **oś x** reprezentuje położenie sygnału (**czas**), a **oś y** reprezentuje **skalę**, kolor każdego punktu (x-y) reprezentuje **wielkość falkowego współczynnika C**.



Wynikiem ciągłej transformaty falkowej są współczynniki określające **podobieństwo pomiędzy daną falką a sygnałem**.

Niska a wysoka skala

- Wyższa skala odpowiada bardziej „rozciągniętym” falkom. Im bardziej rozciągnięta falka tym dłuższa część sygnału z którym falka będzie porównywana.



Związek między skalą falki a częstotliwością

- Niska skala a

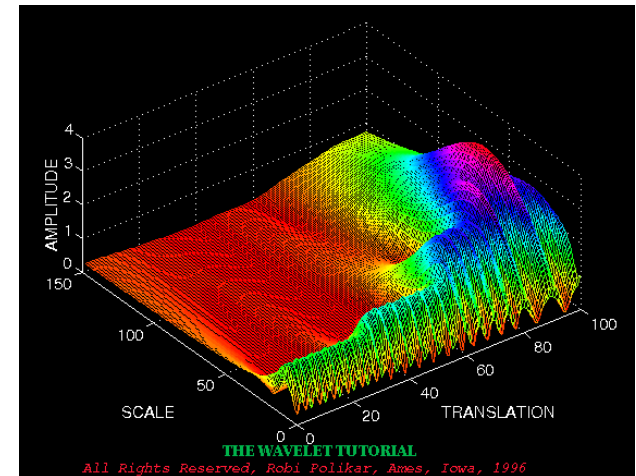
ściśnięta falka →

szybciej zmieniające się współczynniki szczegółowości →
wysoka częstotliwość ω .

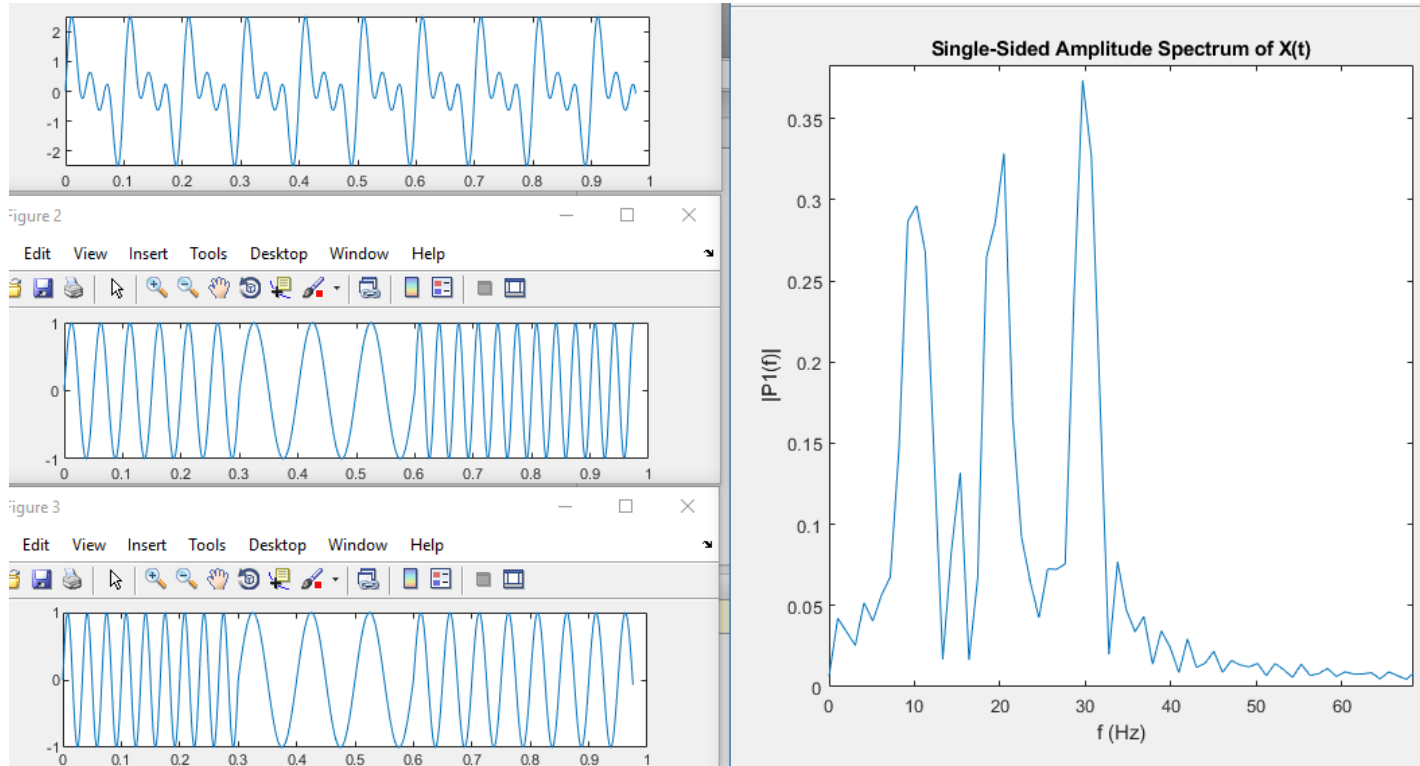
- Wysoka skala a

rozciągnięta falka →

wolno zmieniające się cechy sygnału →
niska częstotliwość ω .



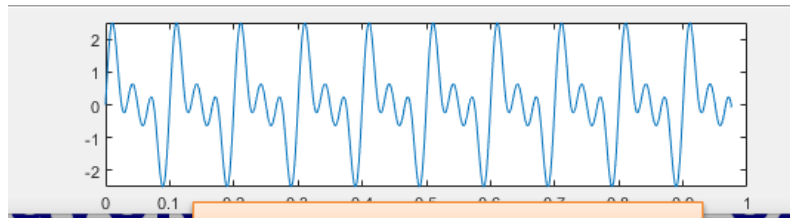
Porównanie transformaty falkowej i Fouriera



10 hz, 20 hz, 30 hz

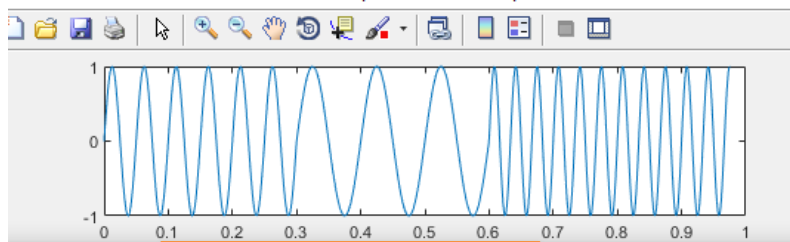
Porównanie transformaty falkowej i Fouriera (cd)

`cwt(y,Fs)`



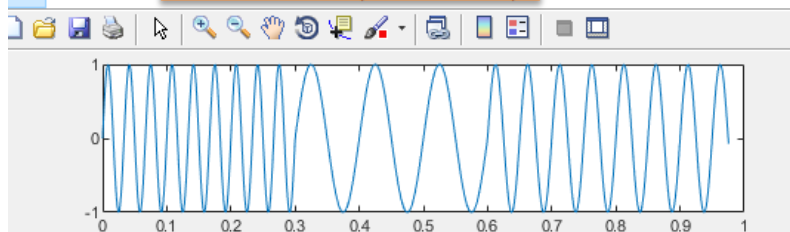
10 hz, 20 hz, 30 hz razem

Figure 2

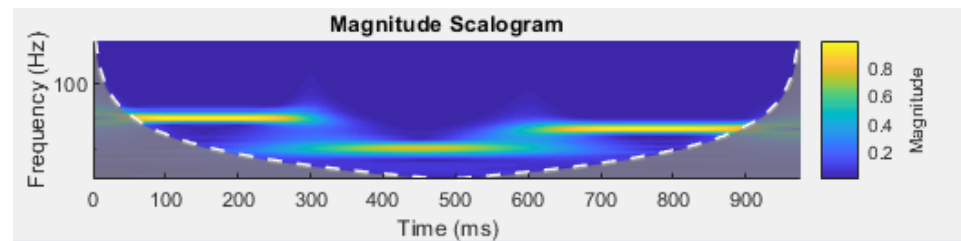
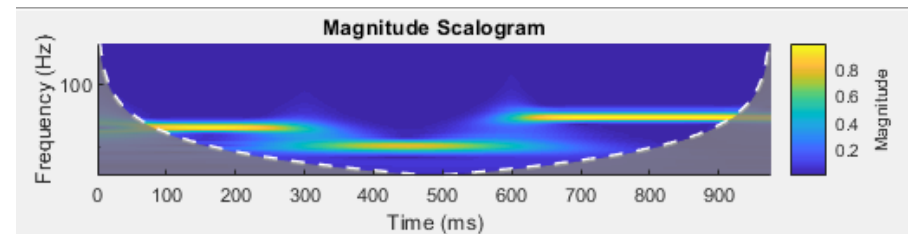
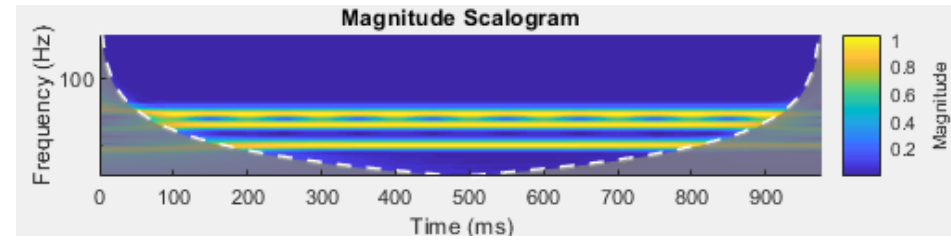


20 hz, 10 hz, 30 hz

Figure 3



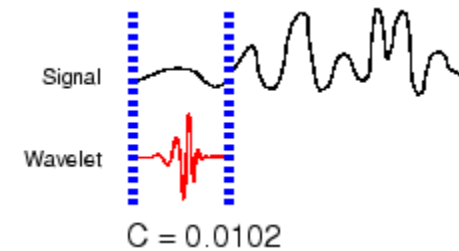
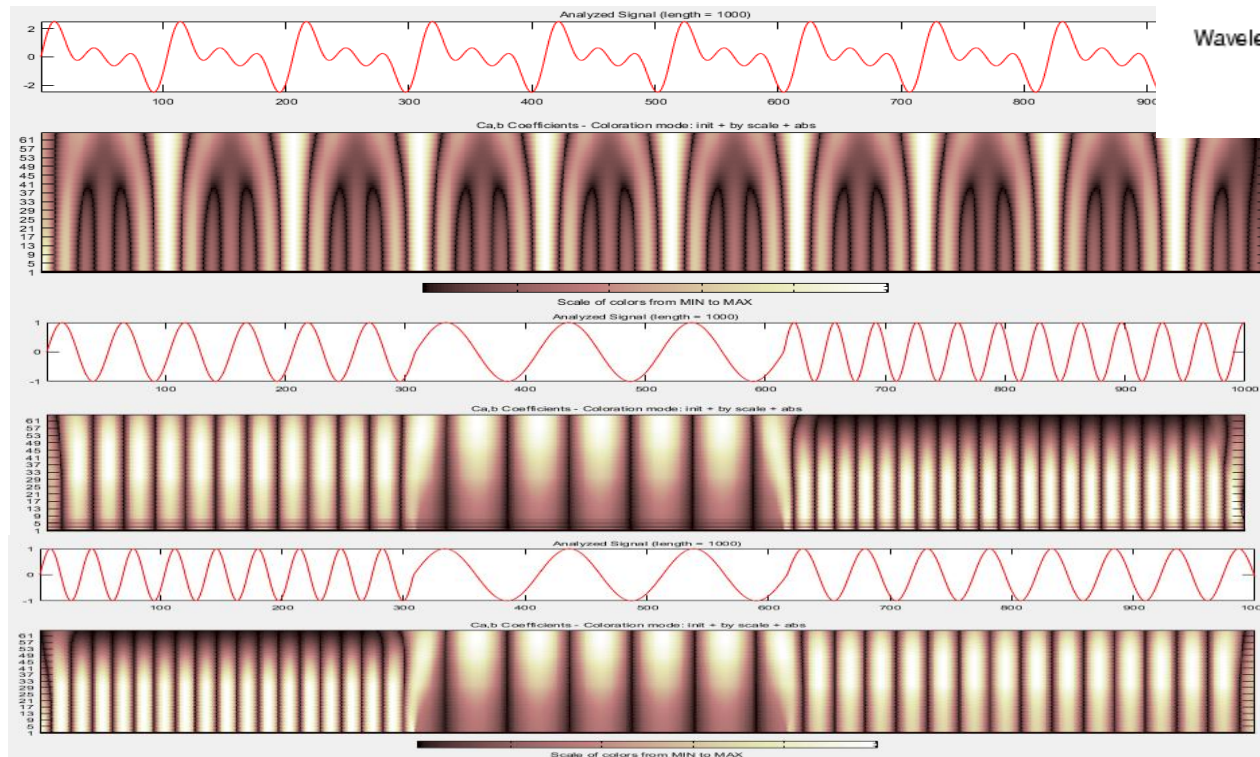
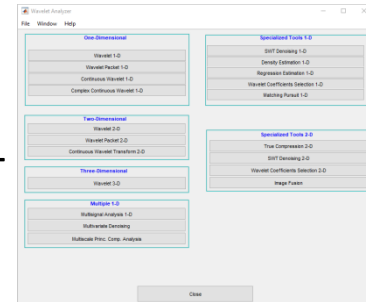
30 hz, 10 hz, 20 hz



>>waveletAnalyzer

CWTcoeffs is a 64-by-1000 matrix. Each row of the matrix contains the CWT coefficients for one scale.

There are 128 rows because the SCALES input to [cwt](#) is 1:64. The column dimension of the matrix matches the length of the input signal.



Odwrotna transformacja falkowa

Odwrotna TF

$$f(t) = \frac{1}{C_\psi} \int_0^{+\infty} \int_{-\infty}^{+\infty} W_f(u, s) \frac{1}{\sqrt{s}} \psi\left(\frac{t-u}{s}\right) du \frac{ds}{s^2}$$

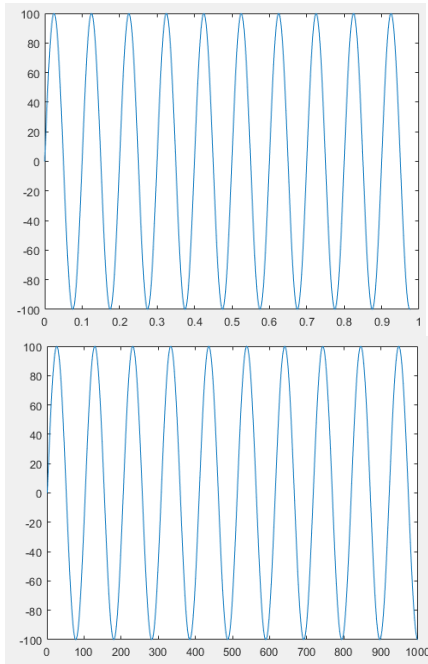
$$C_\psi = \int_0^{+\infty} \frac{|\hat{\psi}(\omega)|^2}{\omega} d\omega < +\infty$$

$\hat{\psi}(\omega)$ - transformacja Fouriera funkcji falkowej

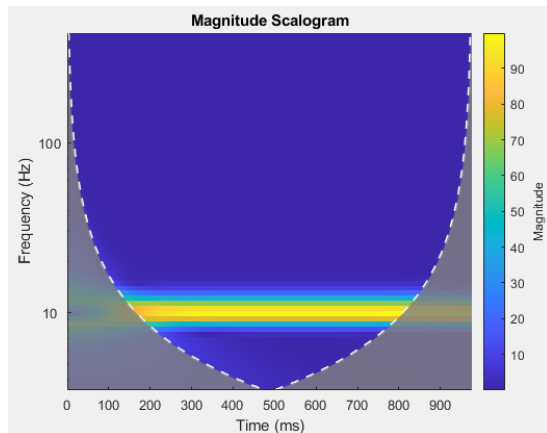
Funkcje Matlab cwt, icwt

Przykład

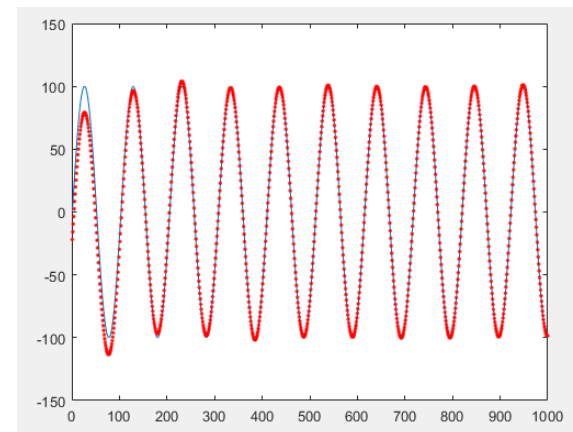
```
Fs = 1024;           % Sampling frequency
T = 1/Fs;             % Sampling period
L = 1000;             % Length of signal
t = (0:L-1)*T;        % Time vector
F = 10;               % Sine wave frequency (hertz)
y = 100*sin(2*pi*F*t);
```



$cc = \text{cwt}(y, Fs)$



$yc = \text{icwt}(cc)$



CWT ze skalą podwójną

- Obliczanie współczynników falkowych dla każdej możliwej skali wymaga dostatecznie dużo pracy i wytwarza dużo danych.
- Co się stanie jeśli wybierzemy tylko **podzbiór skali i położeń** dla których wykonamy nasze obliczenia?

Diadyczne próbkowanie

To rozwiązanie będzie godne uwagi kiedy wybierzemy **skale i położenia** oparte na **potęgze dwójki** –tak zwane podwojone (*dyadic*) skale i położenia–wtedy nasza analiza będzie bardziej efektywna i dostatecznie dokładna.

$$a = 2^m, b = k \cdot 2^m,$$

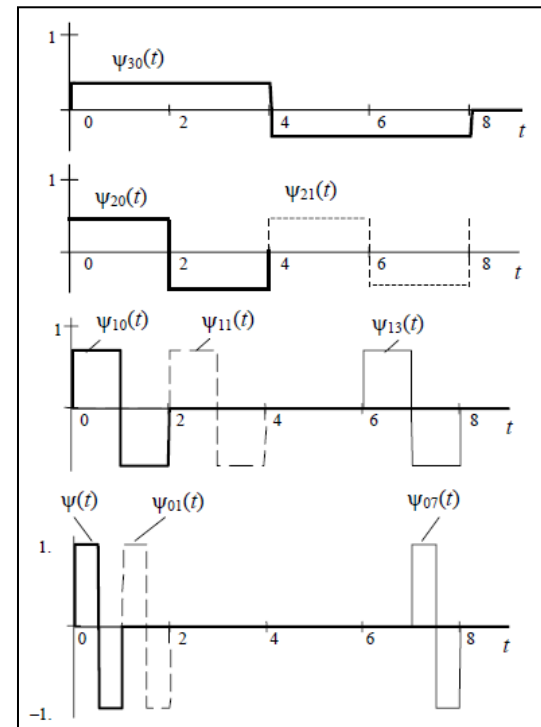
$$\psi_{mk}(t) = \frac{1}{\sqrt{a}} \Psi\left(\frac{t-b}{a}\right) = \frac{1}{\sqrt{2^m}} \psi(2^{-m}t - k)$$

m, k – liczby całkowite

$$c_{mk} = (f(t), \psi_{mk}(t)) = \int_{-\infty}^{\infty} f(t) \Psi_{mk}(t) dt$$

$$f(t) = \sum_{m,k} c_{mk} \Psi_{mk}(t)$$

Falka Haara



Dyskretna transformacja falkowa

- CWT

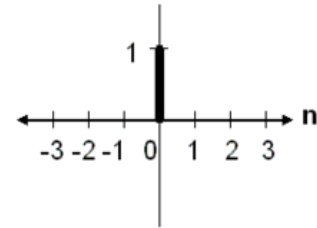
$$\Psi_x^\psi(\tau, s) = \int f(t) \cdot \psi_{\tau, s}^*(t) dt \quad \psi_{\tau, s}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t - \tau}{s}\right)$$

- DWT **Funkcja oraz parametry falek (skale i położenia)** są zdyskretyzowane

$$\Psi_x^\psi[n, a^j] = \sum_{m=0}^{N-1} f[m] \cdot \psi_j^*[m - n] \quad \psi_j[n] = \frac{1}{\sqrt{a^j}} \psi\left(\frac{n}{a^j}\right)$$

Dyskretyzacja funkcji

$$f_D(t) = \{ f_i \} = \sum_{i=1}^{N-1} f(i\Delta t) \delta(t - i\Delta t) \quad \Delta t = 1/2 f_m$$



f_m - **częstotliwość Nyquista** – maksymalna częstotliwość składowych widmowych sygnału $f(x)$, które mogą zostać odtworzone z ciągu próbek bez zniekształceń.

Gdy $N = 2^{n_0}$, maksymalna wartość $m = n_0$ $k = 2^{n_0-m} - 1$

$$a = 2^m, b = k \cdot 2^m,$$

$$\psi_{mk}(t) = \frac{1}{\sqrt{a}} \Psi\left(\frac{t-b}{a}\right) = \frac{1}{\sqrt{2^m}} \psi(2^{-m}t - k)$$

Dla $m=0$ ilość przesunięć będzie $k=N-1$,
Dla $m=1$ – szerokość falki zwiększa się dwa razy, a ilość przesunięć zmniejsza się 2 razy.

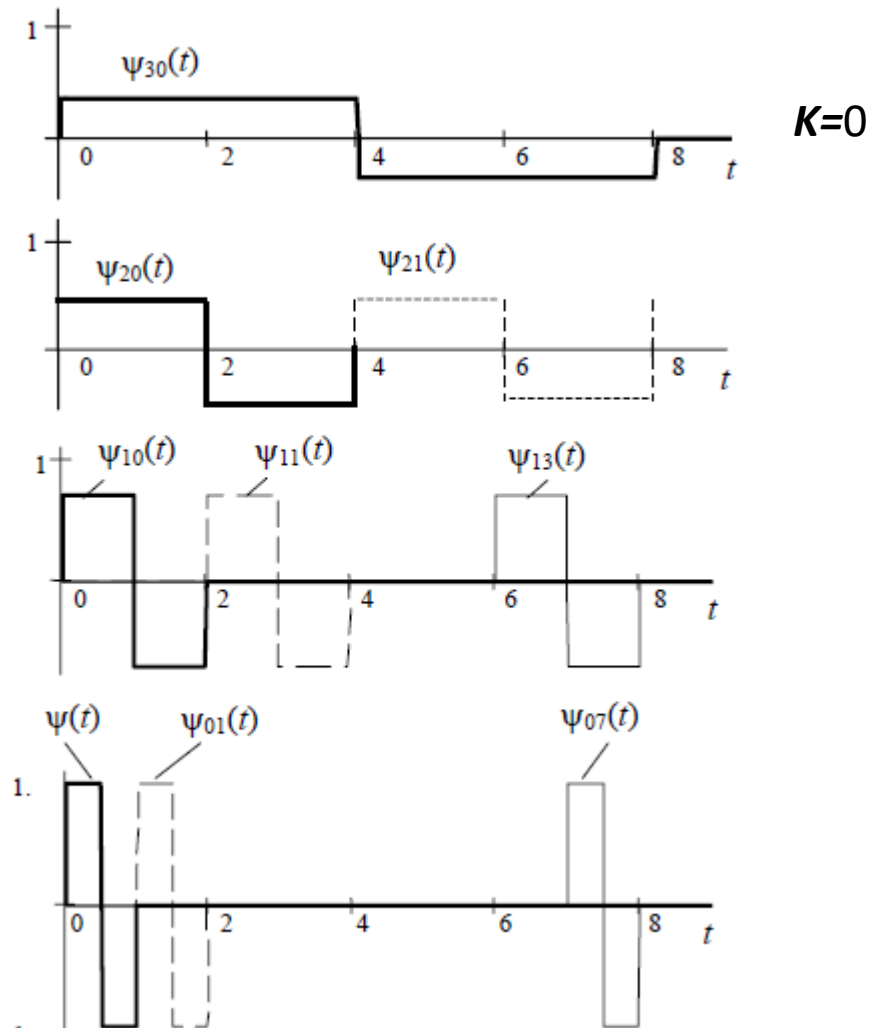
Dla $m=m_{\max}$, mamy jedną falkę.

Przykład

$$N = 8 \quad m_{\max} = 3$$

Dla $m=0$ ilość przesunięć będzie $k=N-1$,
Dla $m=1$ – szerokość falki zwiększa się
dwa razy, a ilość przesunięć zmniejsza się
2 razy.

Dla $m=m_{\max}$, mamy jedną falkę.



Szybka transformacja falkowa

$$f(t) = A_m(t) + \sum_{j=1}^m D_j(t)$$

m - skala

A- aproksymata (niskoczęstotliwościowa reprezentacja sygnału)

D- detale (wysokoczęstotliwościowa reprezentacja sygnału)

Wbudowane GUI Matlab

- signalMultiresolutionAnalyzer
- waveletSignalDenoiser
- waveletTimeFrequencyAnalyzer

$$m = 0, \quad f(t) = A_0(t)$$

$$f(t) = A_0(t) = \sum_k a_{0k} \varphi_{0k}(t)$$

$$m = 1, \quad f(t) = A_1(t) + D_1(t)$$

$$a_{0k} = a_k = (f(t), \varphi_{0k}(t))$$

$$m = 2, \quad f(t) = A_2(t) + D_1(t) + D_2(t)$$

...

$$f(t) = A_1(t) + D_1(t) = \sum_k a_{1k} \varphi_{1k}(t) + \sum_k d_{1k} \Psi_{1k}(t)$$

Multiresolution analysis

Sygnał może zostać przedstawiony jako suma funkcji skalujących i falek, tworzących rodzinę funkcji ortogonalnych.

$$f(t) = \sum_k a_j(k) 2^{j/2} \varphi(2^j t - k) + \sum_k d_j(k) 2^{j/2} \psi(2^j t - k)$$

$$a_j(k) = \langle f(t), \varphi_{j,k}(t) \rangle \quad d_j(k) = \langle f(t), \Psi_{j,k}(t) \rangle$$

$$f_{m+1}(t) = \sum_n a_{m+1,n} \varphi_{m+1,n}(t) \quad a_{m,n} = \sum_k h_{k-2n} a_{m+1,k}$$

$$d_{m,n} = \sum_k g_{k-2n} c_{m+1,k}$$

$$\text{DWT} = \left\{ \left\{ d_{m,n} \right\}_n, \left\{ d_{m-1,n} \right\}_n, \dots, \left\{ d_{m-M,n} \right\}_n, \left\{ a_{m-M,n} \right\}_n \right\}$$

Algorytm Mallata

Mallat podał zależności między kolejnymi współczynnikami rozkładu:

$$a_j(k) = \sum_m h(m - 2k) a_{j+1}(m)$$

$$d_j(k) = \sum_m g(m - 2k) d_{j+1}(m)$$

h – współczynniki filtra dolnoprzepustowego skalującego H ,

g – współczynniki filtra górnoprzepustowego falkowego G .

na wyjściu filtra H otrzymujemy uśrednioną, wygładzoną informację o sygnale wejściowym, na wyjściu filtra G – szczegóły sygnału

Aproksymacja falkowa Haara

$$a_m = \frac{f_{2m-1} + f_{2m}}{\sqrt{2}} \quad d_m = \frac{f_{2m-1} - f_{2m}}{\sqrt{2}}$$

$$m = 1, 2, 3, \dots, N/2.$$

$$\mathbf{f} \xrightarrow{\mathbf{H}_1} (\mathbf{a}^1 \mid \mathbf{d}^1)$$

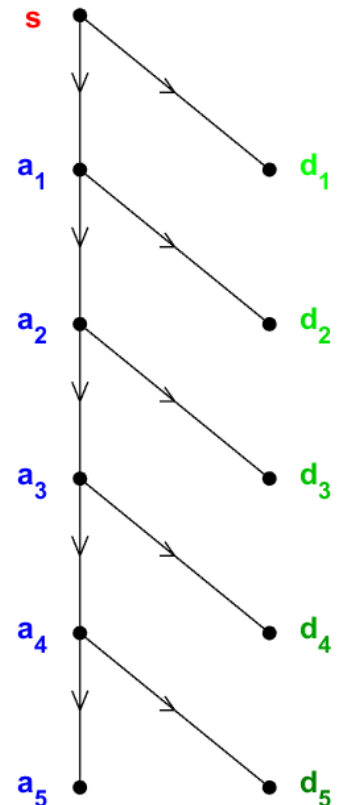
$$\mathbf{a}^1 \rightarrow (\mathbf{a}^2, \mathbf{d}^2)$$

$$\mathbf{a}^2 \rightarrow (\mathbf{a}^3, \mathbf{d}^3)$$

...

$$\mathbf{a}^{m-1} \rightarrow (\mathbf{a}^m, \mathbf{d}^m)$$

$$\mathbf{f} = \mathbf{a}^m + \mathbf{d}^m + \mathbf{d}^{m-1} + \dots + \mathbf{d}^1$$



Aproksymacja i szczegóły

Dla wielu sygnałów część nisko-częstotliwościowa jest najważniejszą częścią. To ona stanowi o tożsamości sygnału.

Z drugiej strony część wysoko-częstotliwościowa, przekazuje szczegółowe cechy sygnału.

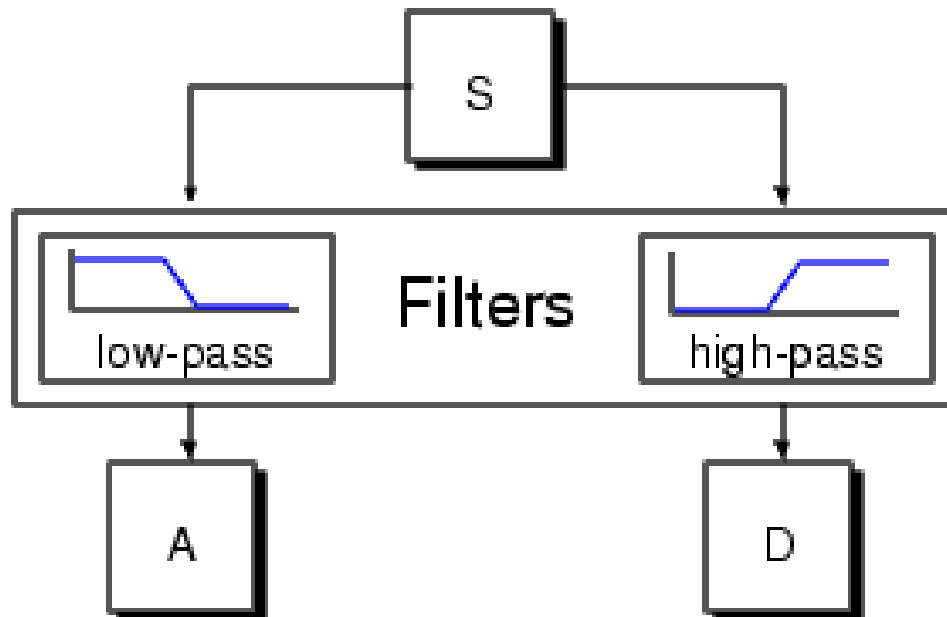
W analizie falkowej, zwykle mówimy o aproksymacjach(approximations) i współczynnikach szczegółowości(details).

Aproksymacje są elementami sygnału o wysokiej skali i niskiej częstotliwości.

Detale są elementami o niskiej skali i wysokiej częstotliwości

Filtrowanie

Proces filtrowania w najprostszym przypadku wygląda następująco:



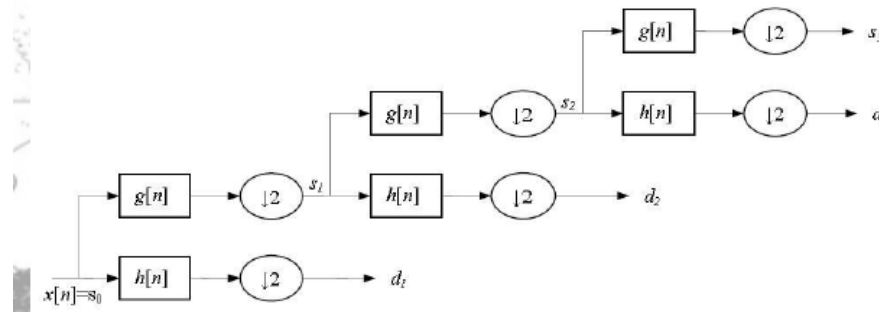
Wielo rozdzielczość

Sygnał jako suma reprezentacji szczegółowej i aproksymowanej

- na każdym następnym poziomie reprezentacja z poziomu poprzedniego jest ponownie przedstawiana w postaci sumy reprezentacji szczegółowej i reprezentacji zgrubnej
- reprezentacja szczegółowa na danym poziomie nie ulega zmianie w wyniku kolejnych iteracji
- z każdym kolejnym poziomem ciąg szczegółów rozbudowuje się o jeden element, z każdym kolejnym poziomem reprezentacji jest również związana jej rozdzielczość
- W ten sposób powstaje drabina kolejnych współczynników, które będziemy nazywali współczynnikami falki. Ilość tych współczynników jest równa ilości punktów sygnału ale ich uporządkowanie jest specyficzne - powiązane z procesem filtracji i dwójkowania.

Wielorozdzielczość

Wielorozdzielczość - dekompozycja sygnału w ciąg sygnałów o coraz mniejszej rozdzielczości na wielu poziomach



gdzie:

s_{l+1} - **aproksymata** (niskoczęstotliwościowa reprezentacja sygnału)

d_{l+1} - **detale** (wysokoczęstotliwościowa reprezentacja sygnału)

- Wejściowy sygnał podzielony na dwa sygnały s_{l+1} i d_{l+1} zawierające dwukrotnie mniej próbek każdy, na ich podstawie możliwe jest późniejsze bezstratne odtworzenie obrazu.

- Sygnał s_{l+1} poddawany jest ponownej dekompozycji, sygnał d_{l+1} jest zapisywany na wyjściu transformacji

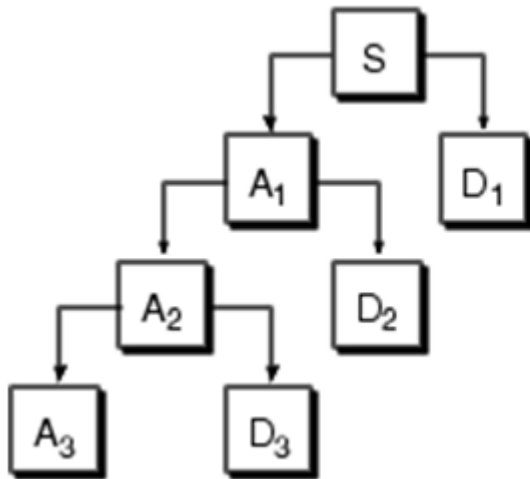
Analiza i rekonstrukcja falkowa

Proces odzyskiwania oryginalnego sygnału bez straty informacji nazywany jest synteza lub rekonstrukcją. Przekształcenie matematyczne które w wyniku daje synteze jest zwana odwrotną transformatą falkową (inverse discrete wavelet transform IDWT).

Podczas gdy analiza falkowa obejmowała filtrowanie i downsampling, proces rekonstrukcji falkowej zawiera filtrowanie i ***upsampling***.

Drzewo dekompozycji

Proces dekompozycji może być powtarzany, o kolejne rozkłady współczynników aproksymacji, tak więc jeden sygnał może być rozbity na wiele mniejszych elementów. Proces ten można przedstawić za pomocą drzewa dekompozycji falkowej (*wavelet decomposition tree*).



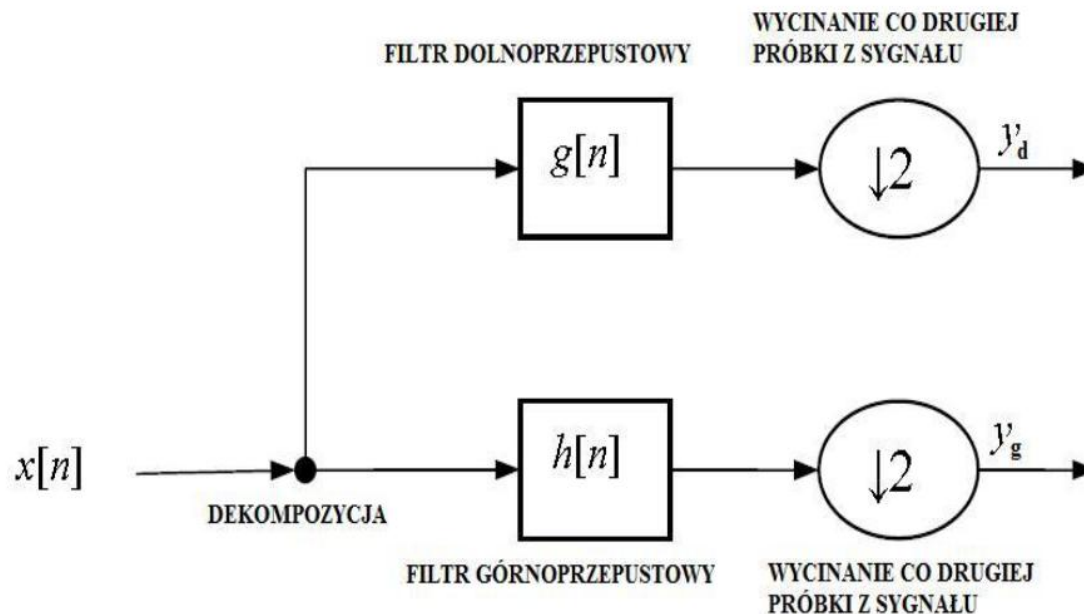
$$S = A_1 + D_1$$

$$= A_2 + D_2 + D_1$$

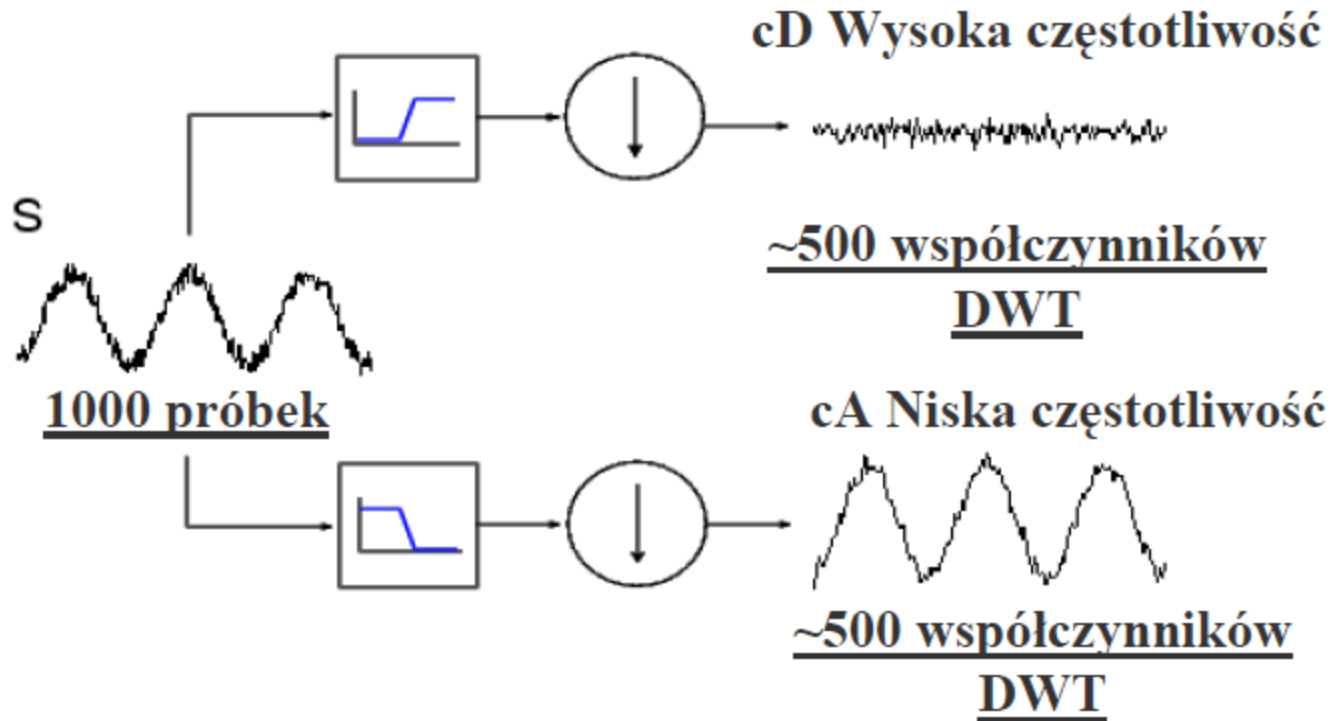
$$= A_3 + D_3 + D_2 + D_1$$

Sposób przetwarzania sygnału

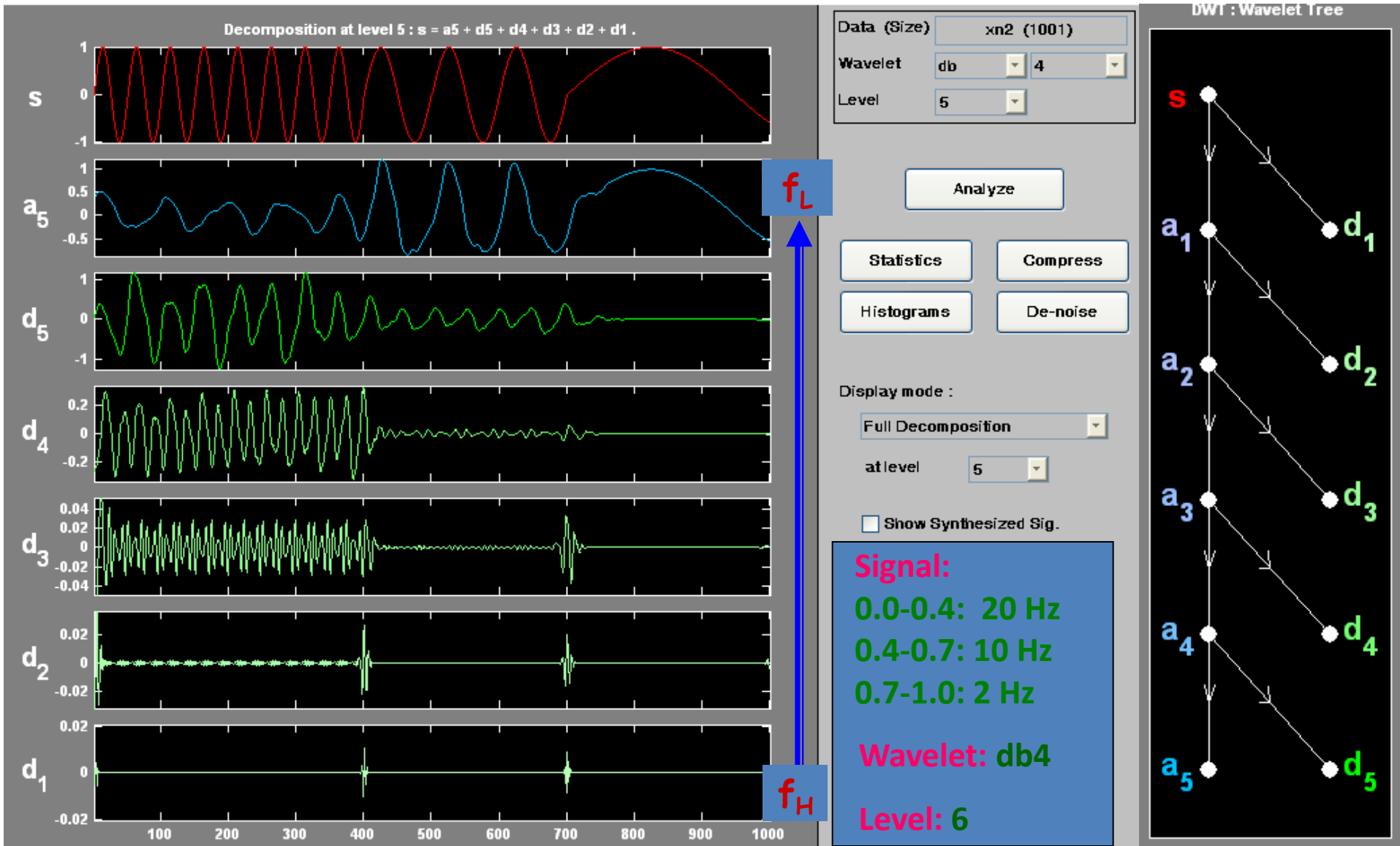
filtracja sygnału w serii kaskadowo połączonych filtrów



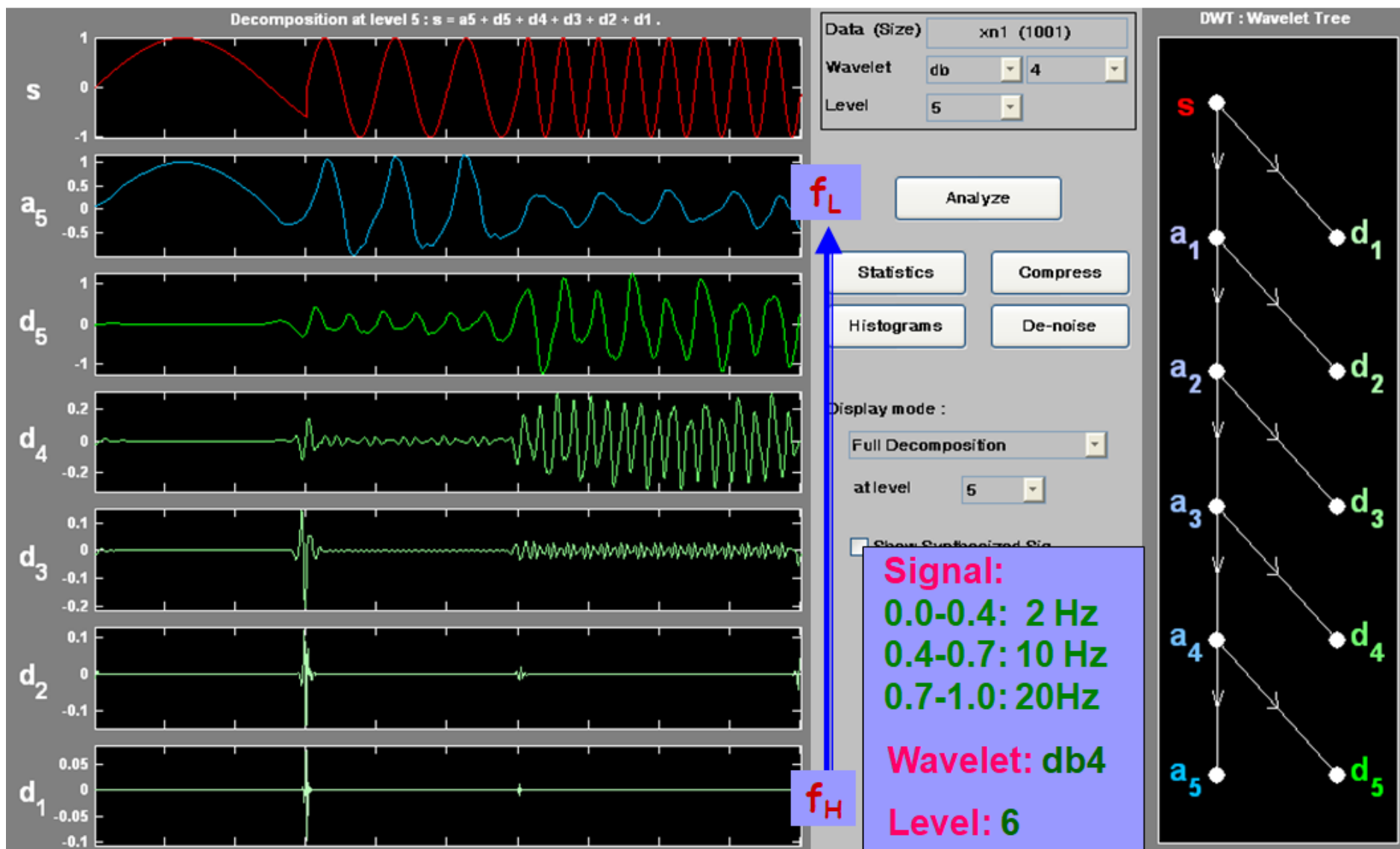
Przykład



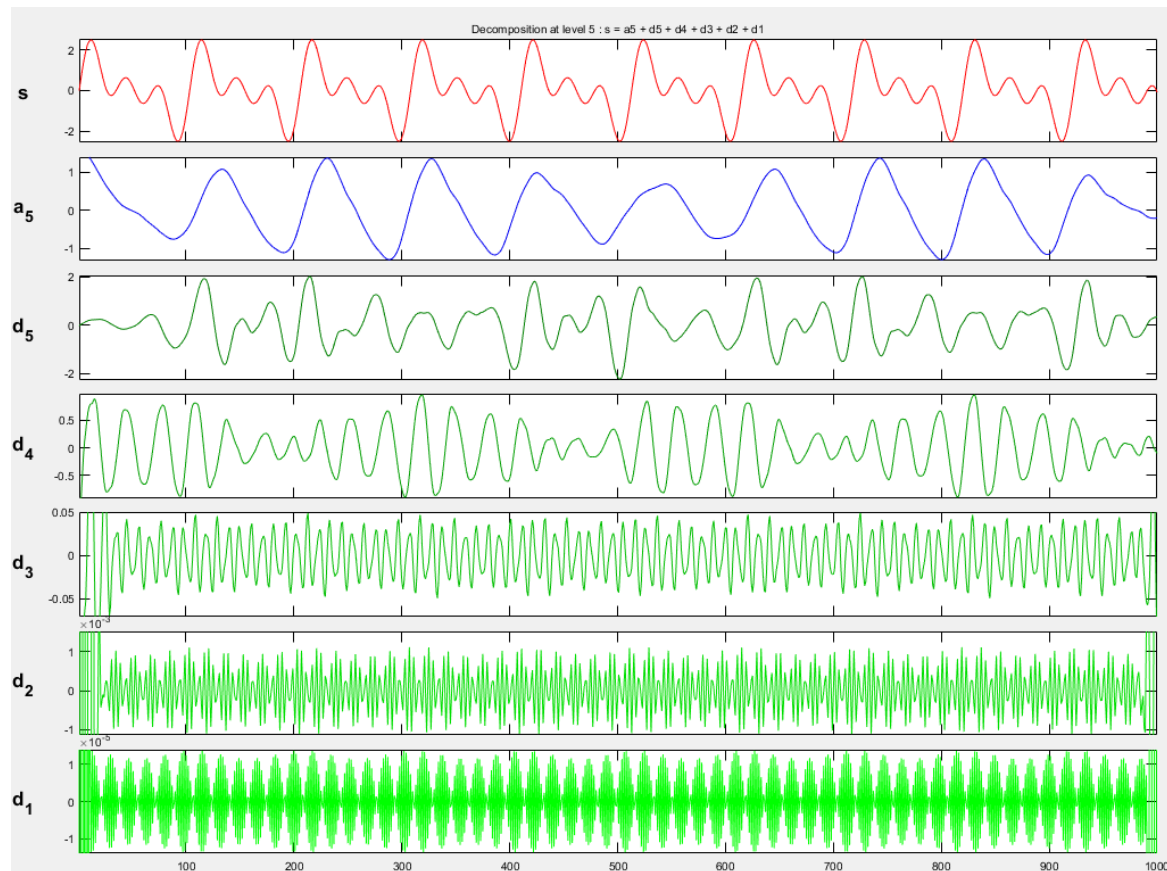
Przykład



Przykład (cd)



Przykład (cd)



Data (Size)

Wavelet

Level

Analyze

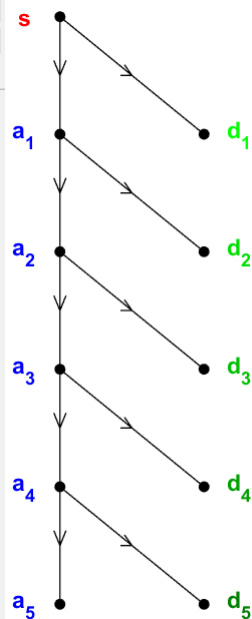
Statistics Compress

Histograms De-noise

Display mode :

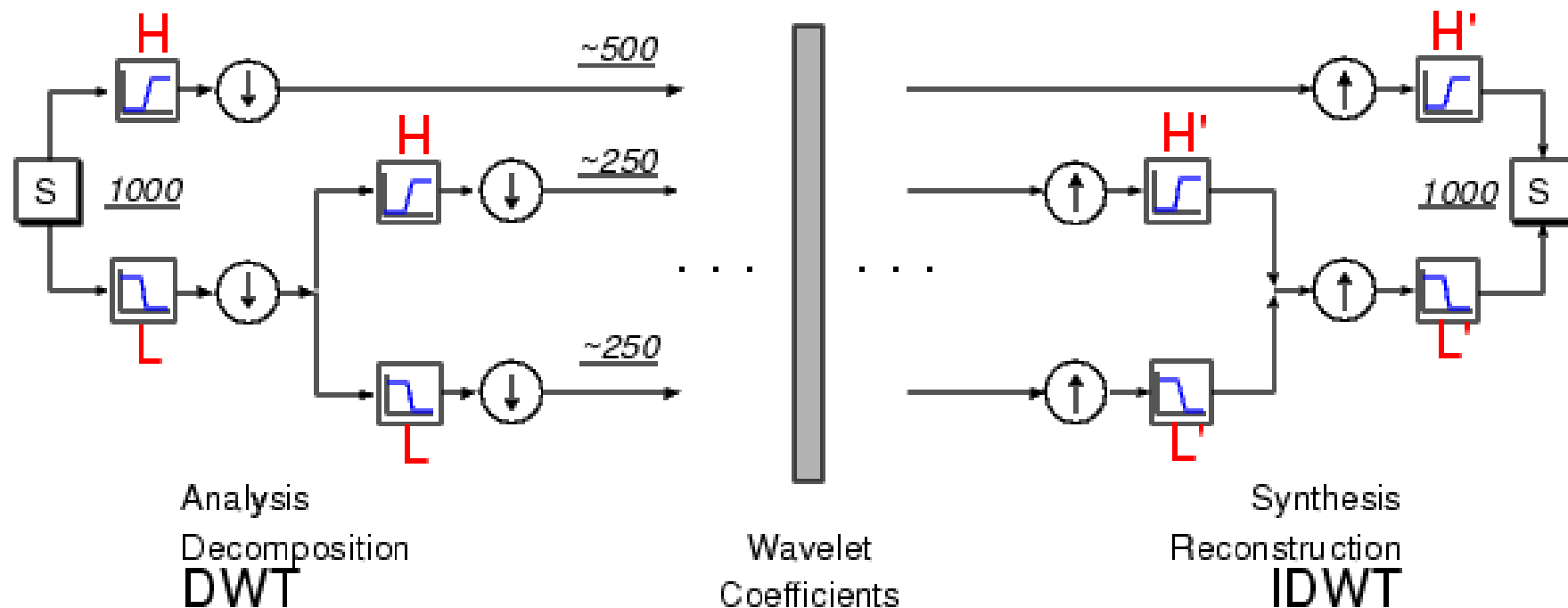
at levels

☐ Show Synthesized Sig.



Apps: Wavelet Signal Denoiser

Rekonstrukcja sygnału



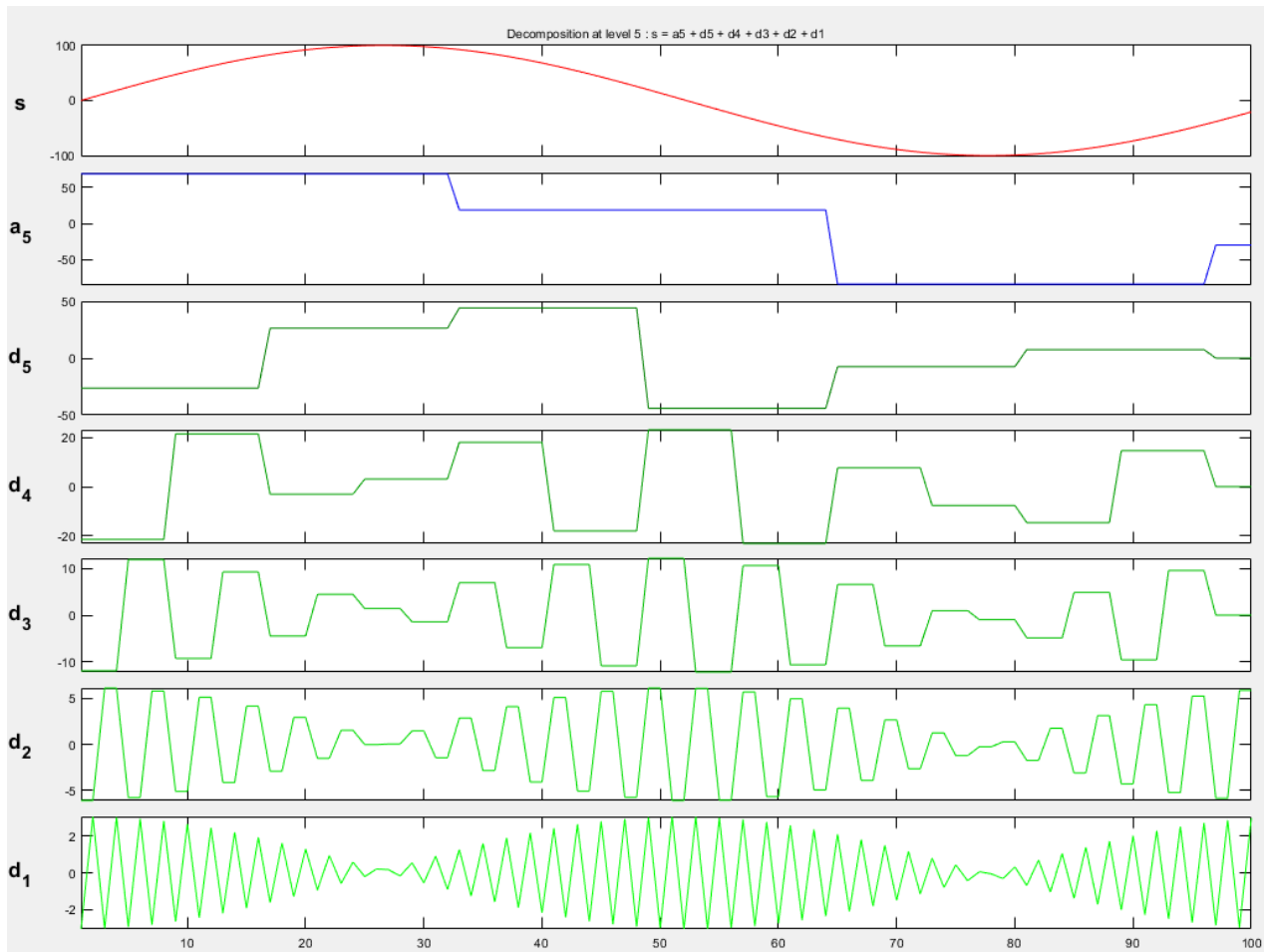
Przykład

$$a_1^1 = (y_1 + y_2) / 2 \quad d_1^1 = (y_1 - y_2) / 2$$

$$a_2^1 = (y_3 + y_4) / 2 \quad d_2^1 = (y_3 - y_4) / 2$$

$$a_1^1 + d_1^1 = \frac{y_1 + y_2}{2} + \frac{y_1 - y_2}{2} = y_1$$

$$a_1^1 - d_1^1 = \frac{y_1 + y_2}{2} - \frac{y_1 - y_2}{2} = y_2$$



Data (Size) y (100)

Wavelet haar

Level 5

Analyze

Statistics

Compress

Histograms

De-noise

Display mode :

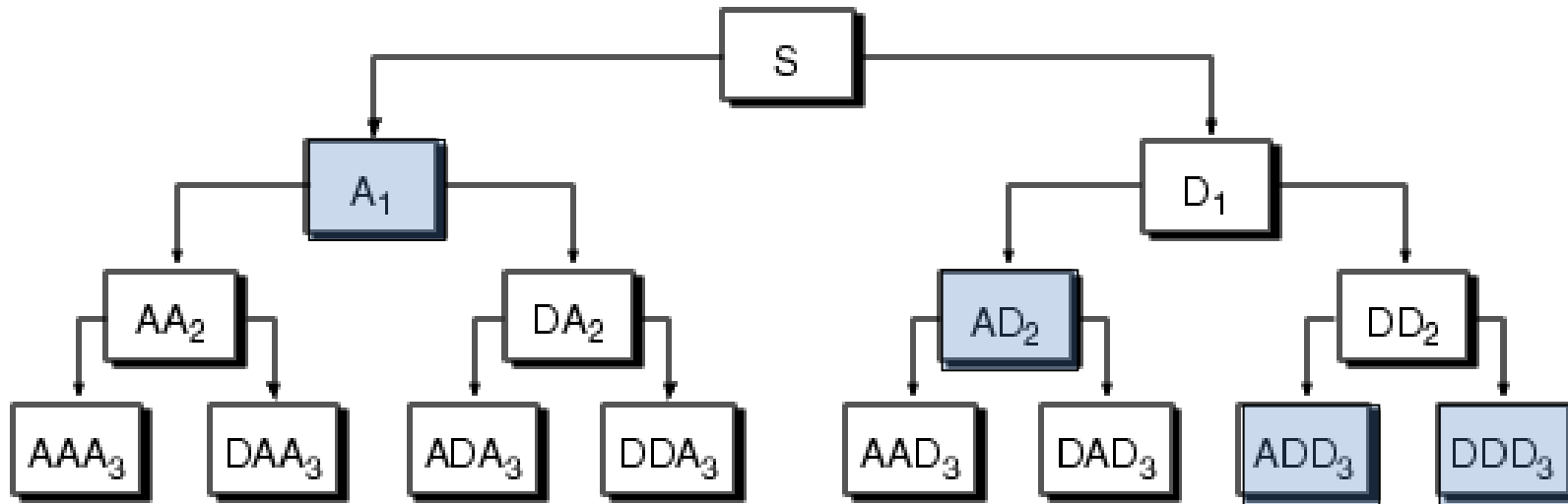
Full Decomposition

at levels

5

☐ Show Synthesized Sig.

Pakietowa rekonstrukcja

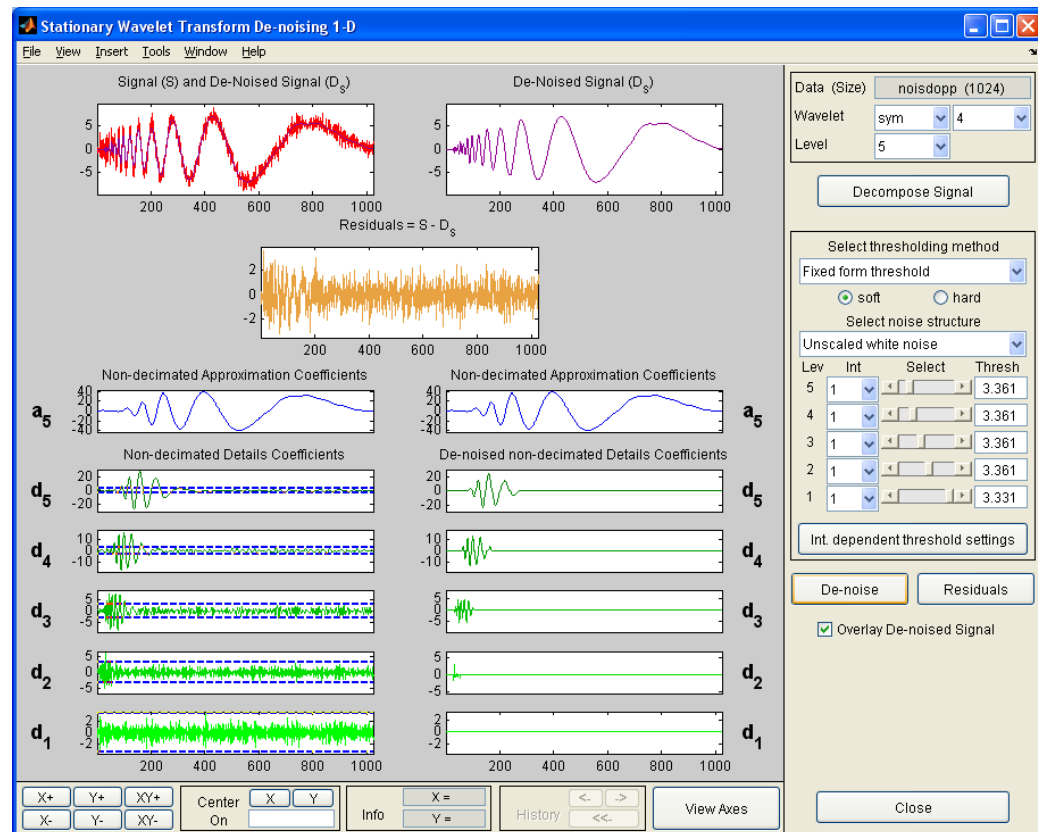


Na przykład

$$S = A_1 + AD_2 + ADD_3 + DDD_3$$

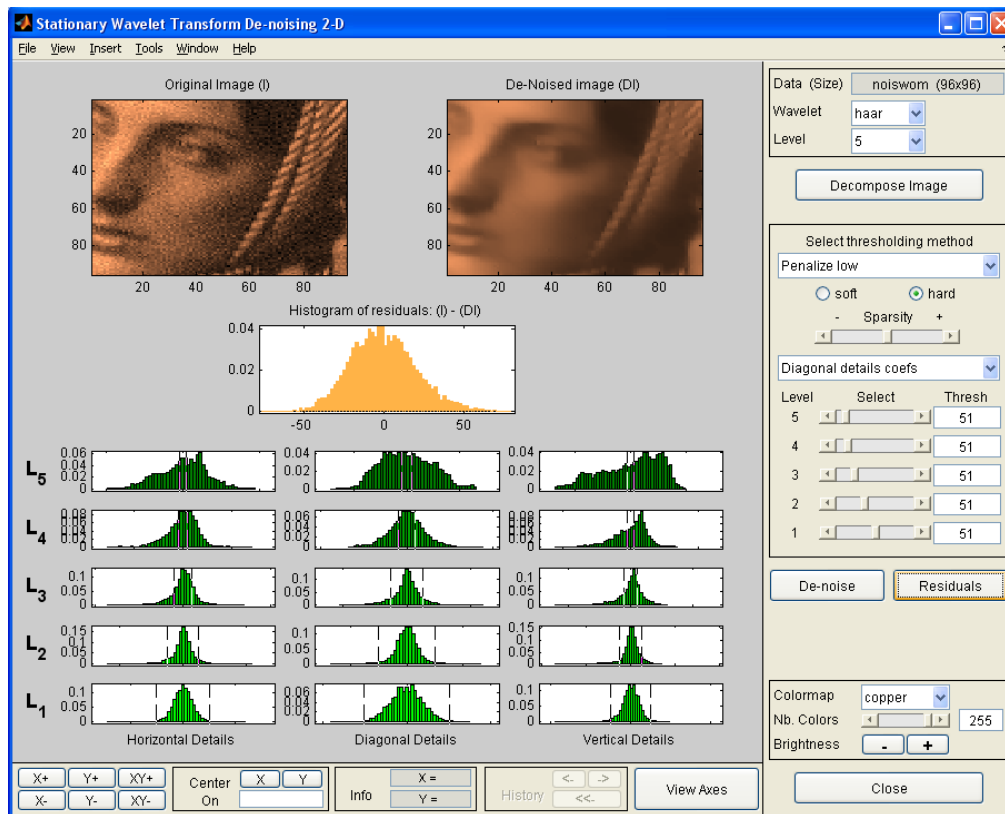
Zastosowania

- Wygładzanie (eliminacja szumu)



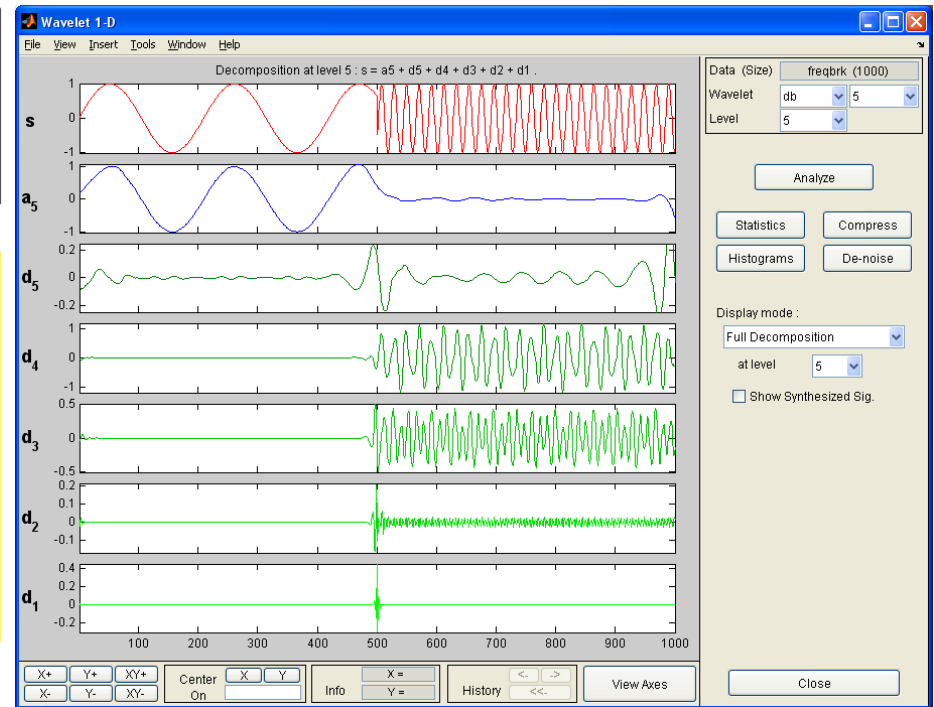
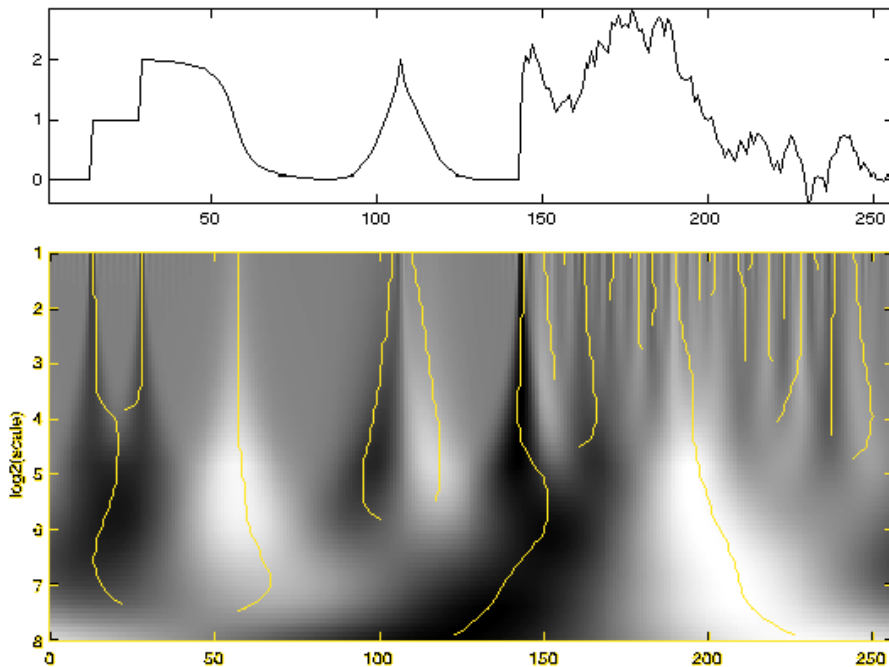
Zastosowania

- Wygładzanie szumu na zdjęciach



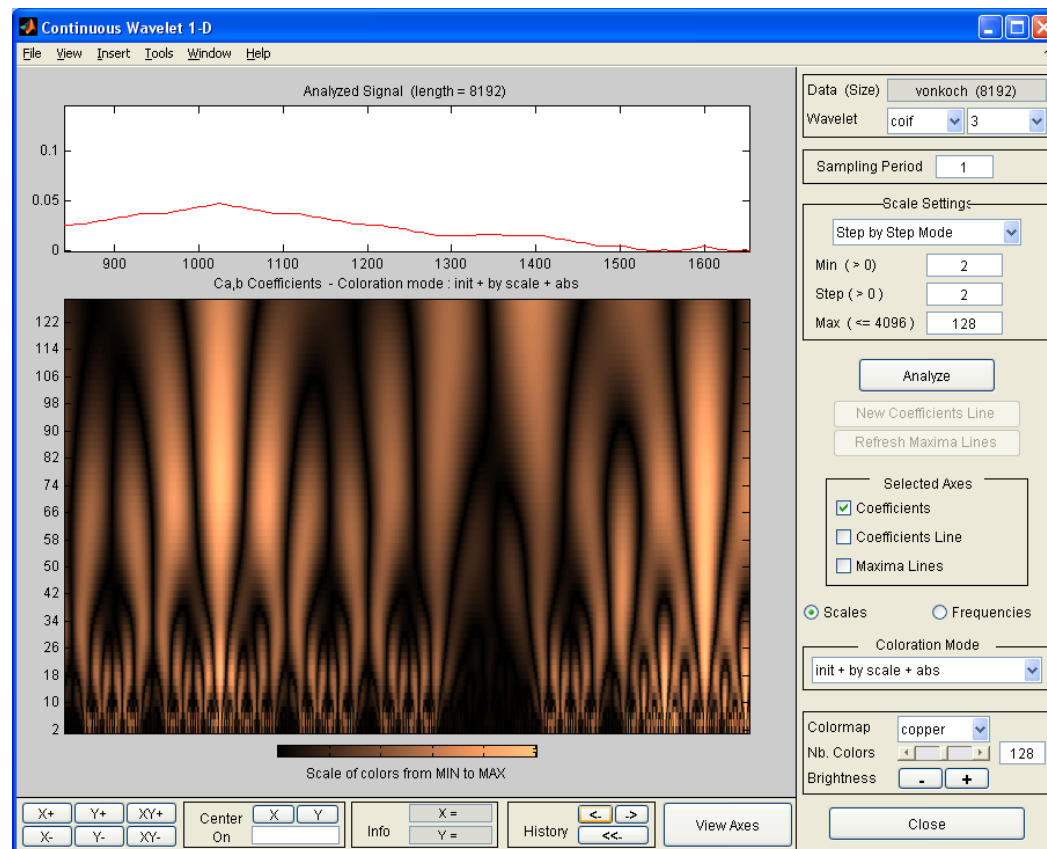
Zastosowania

- Wykrywanie nieciągłości funkcji



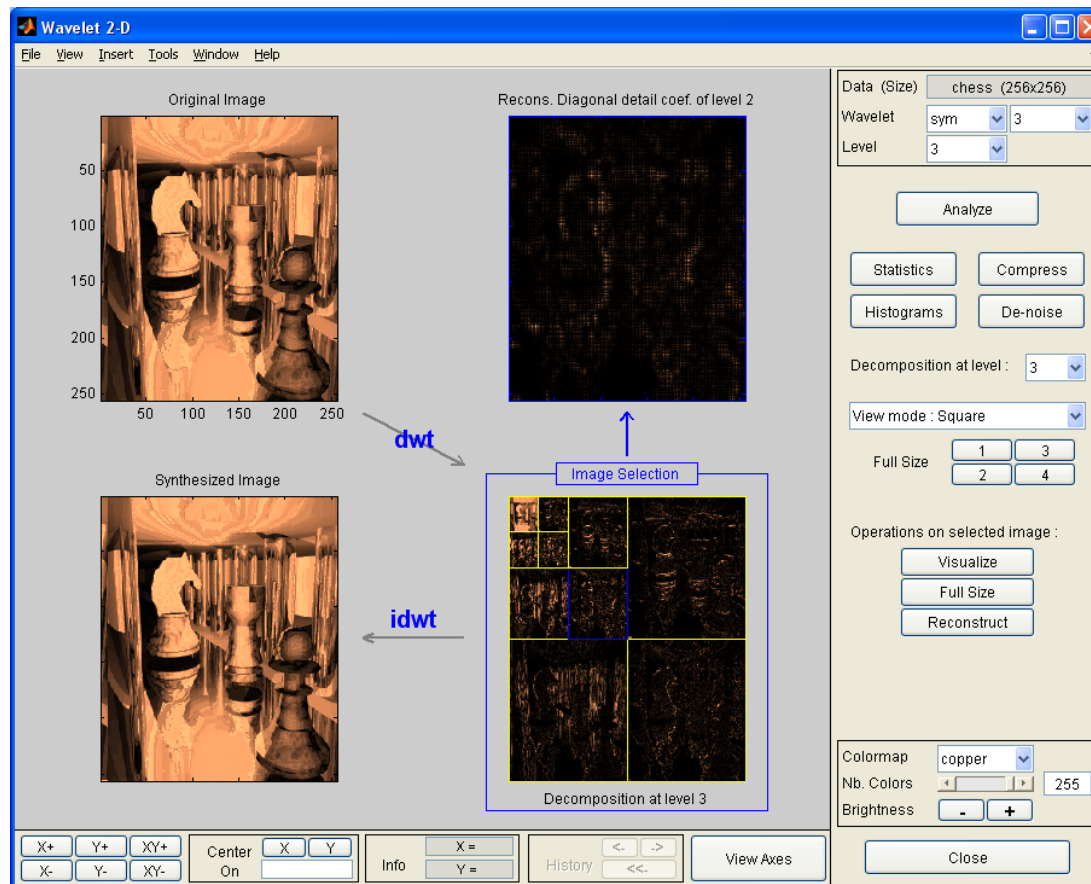
Zastosowania

- Wykrywanie samopodobieństwa



Zastosowania

- Kompresja obrazów



Algorytm Mallata

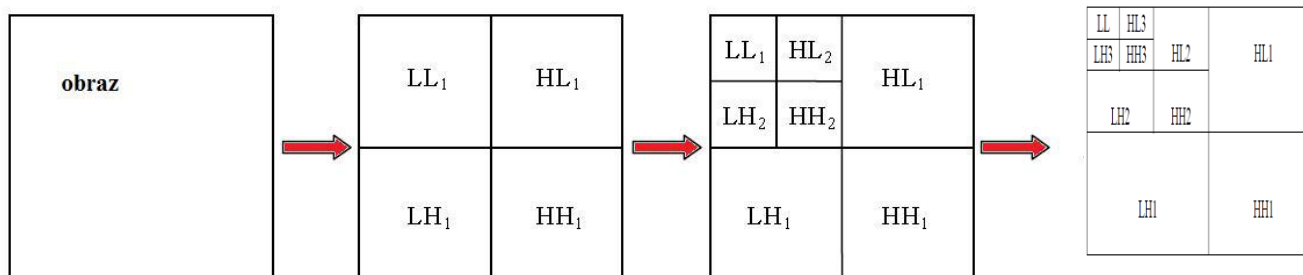
Dekompozycja obrazu polega na sekwencyjnym filtrowaniu górno- i dolnoprzepustowym osobno wzdłuż kolumn i wierszy

LL- filtr dolnoprzepustowy dla wierszy i kolumn

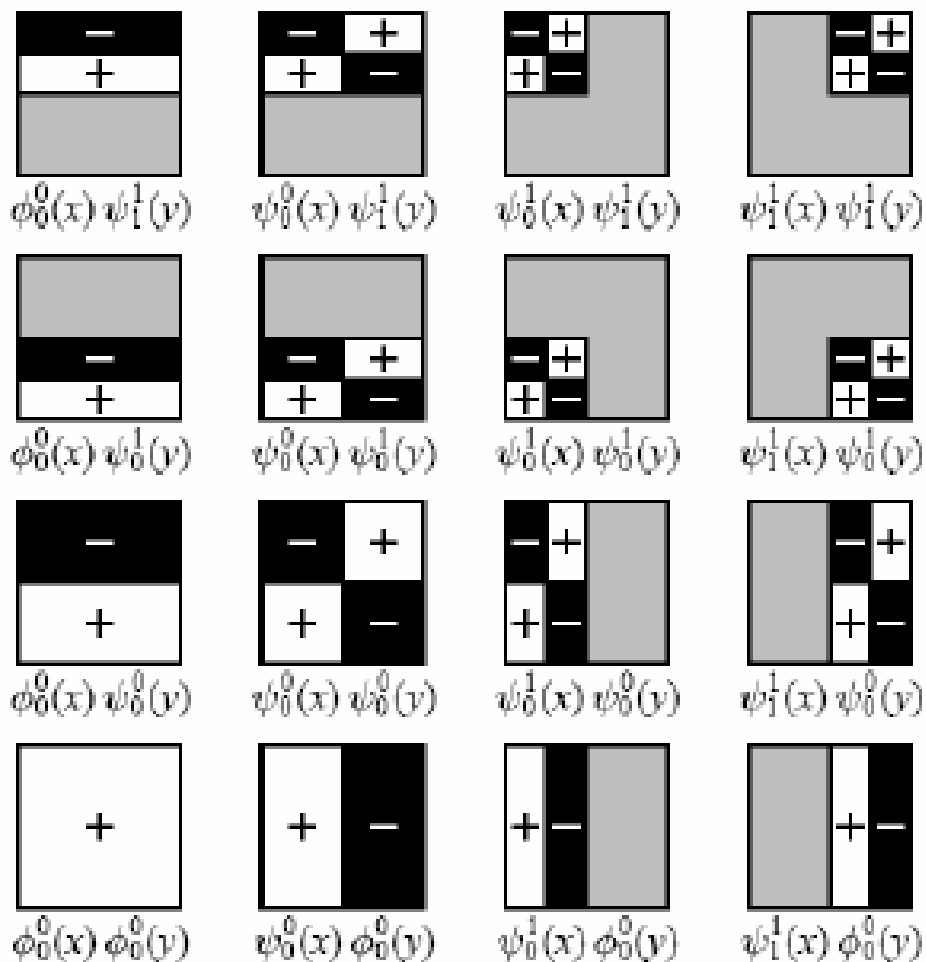
LH- dolnoprzepustowy dla wierszy, górnoprzepustowy dla kolumn

HL- górnoprzepustowy dla wierszy, dolnoprzepustowy dla kolumn

HH- filtr górnoprzepustowy dla wierszy i kolumn



Transformacja Haara



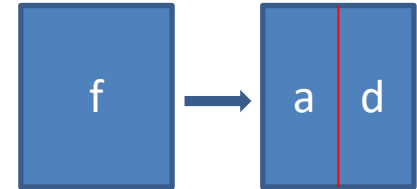
Algorytm Mallata (cd)

Analiza wierszy i kolumn przy użyciu filtrów odbywa się oddzielnie (najpierw dekomponowane na aproksymaty i detale są wiersze, na podstawie otrzymanych podobrazów, dekomponowane są kolumny)

- Po dekompozycji obraz jest reprezentowany przez 4 macierze współczynników falkowych (każda składowa - $\frac{1}{4}$ obrazu pierwotnego), posiadające dwukrotnie mniejszą rozdzielczość liniową
- Każda kolejna dekompozycja według tego schematu pozwala na wprowadzenie wielopoziomowej rozdzielczości
- W dalszej części dekompozycja polega na sekwencyjnym filtrowaniu górnoprzepustowym i dolnoprzepustowym osobno wzdłuż kolumn i wierszy

Krok 1

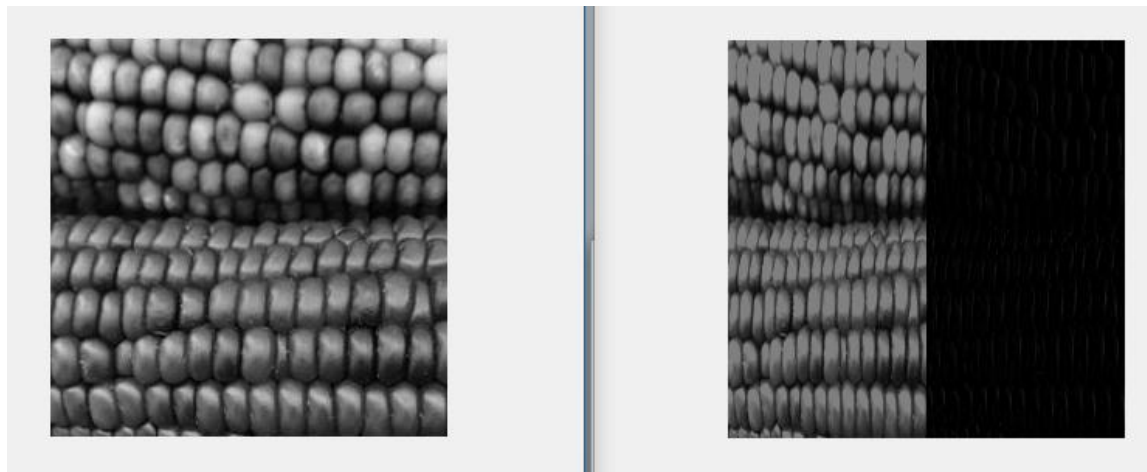
$$\mathbf{f} = \begin{pmatrix} f_{1,M} & f_{2,M} & \dots & f_{N,M} \\ \vdots & \vdots & \ddots & \vdots \\ f_{1,2} & f_{2,2} & \dots & f_{N,2} \\ f_{1,1} & f_{2,1} & \dots & f_{N,1} \end{pmatrix}$$



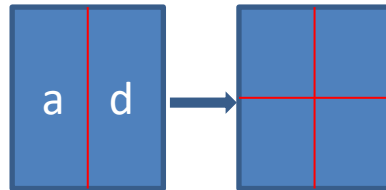
$$a_m = \frac{f_{2m-1} + f_{2m}}{\sqrt{2}}$$

$$d_m = \frac{f_{2m-1} - f_{2m}}{\sqrt{2}}$$

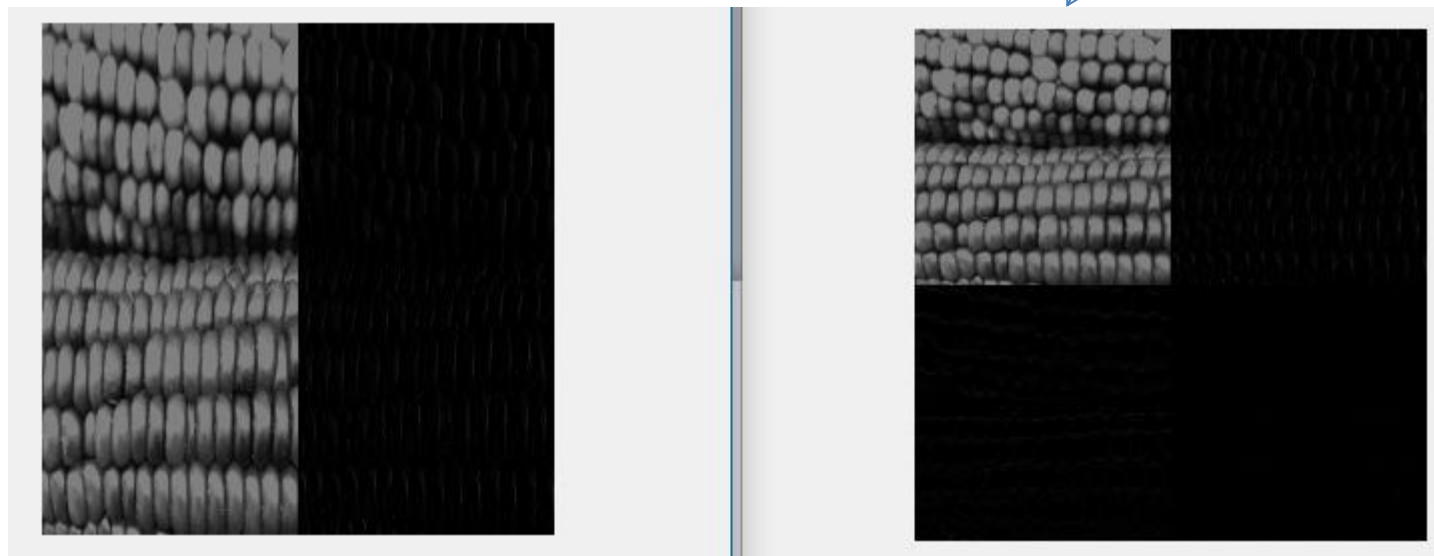
$$m = 1, 2, 3, \dots, N/2.$$



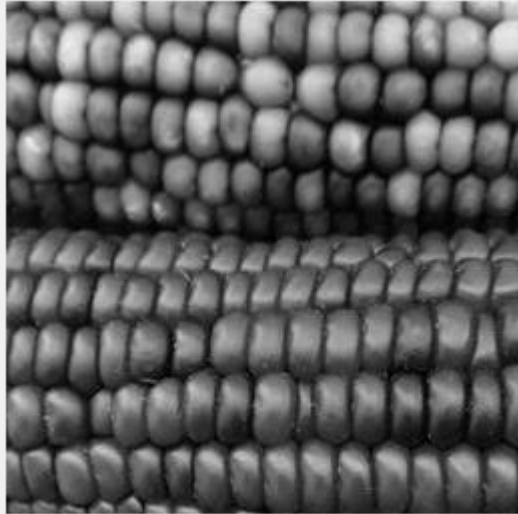
Krok 2



Kompresja zdjęcia



Program

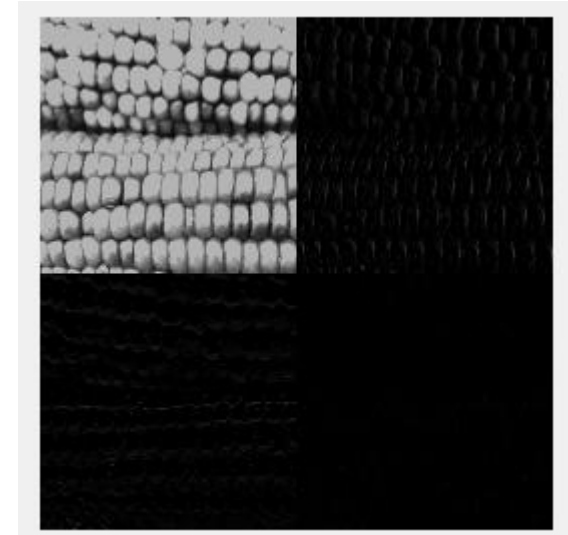
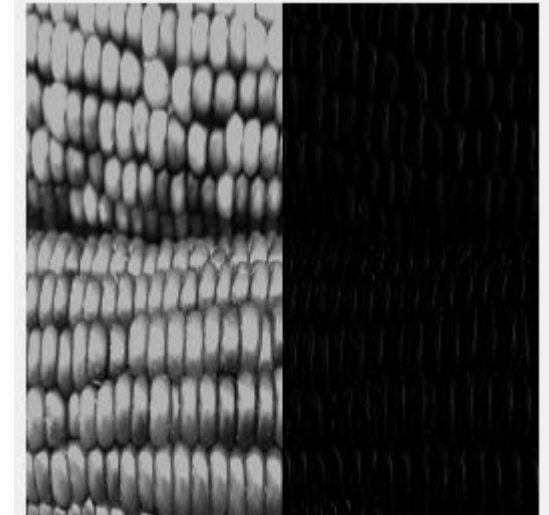
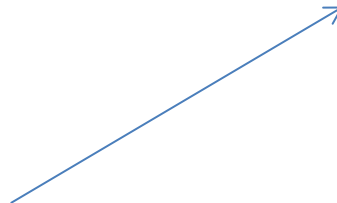


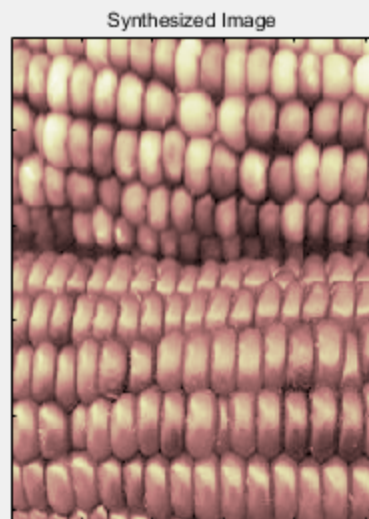
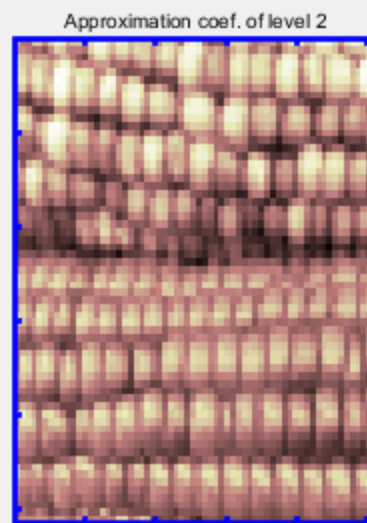
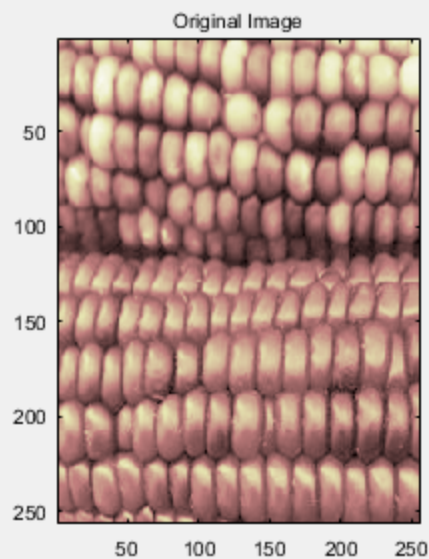
```
f=imread('KornOS.png');
imshow(f);
N=length(f);
f1=[];
for i=1:N %wierszy
    for m=1:N/2

        a(m)=(f(i,2*m-1)+f(i,2*m))/sqrt(2);
        d(m)=(f(i,2*m-1)-f(i,2*m))/sqrt(2);

    end;
    f1=[f1;a d];

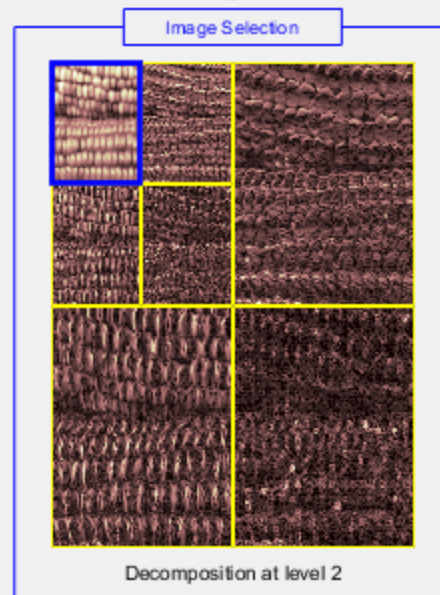
end;
figure;
imshow(f1);
f2=[];
for j=1:N %kolumny
    for m=1:N/2
        a(m)=(f1(2*m-1,j)+f1(2*m,j))/sqrt(2);
        d(m)=(f1(2*m-1,j)-f1(2*m,j))/sqrt(2);
    end;
    f2=[f2;a d];
end;
f2=f2';
figure;
imshow(f2);
```





dwt

idwt



Data (Size)

Wavelet

Level

Analyze

Statistics Compress

Histograms De-noise

Decomposition at level :

View mode : Square

Full Size

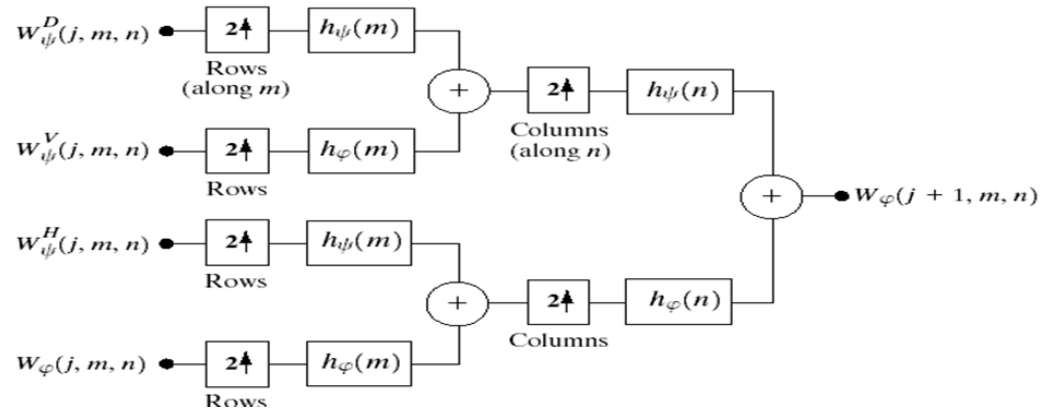
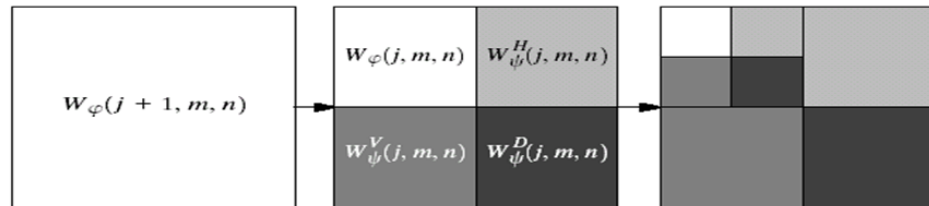
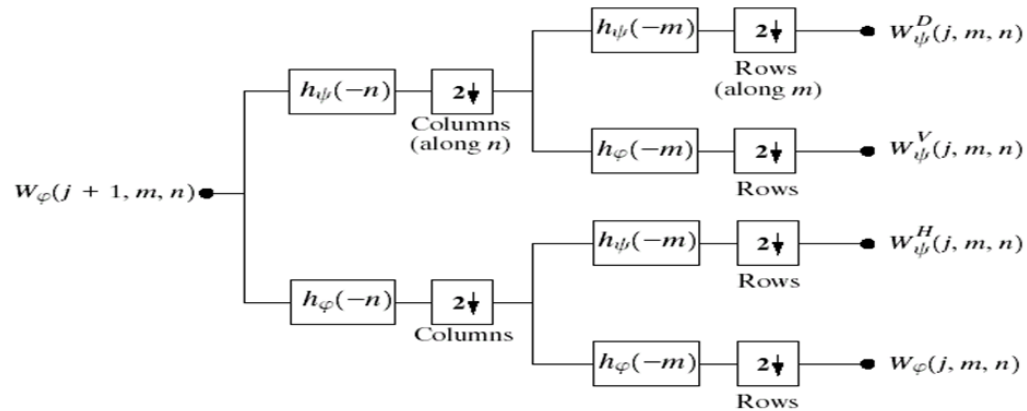
Operations on selected image :

Colormap

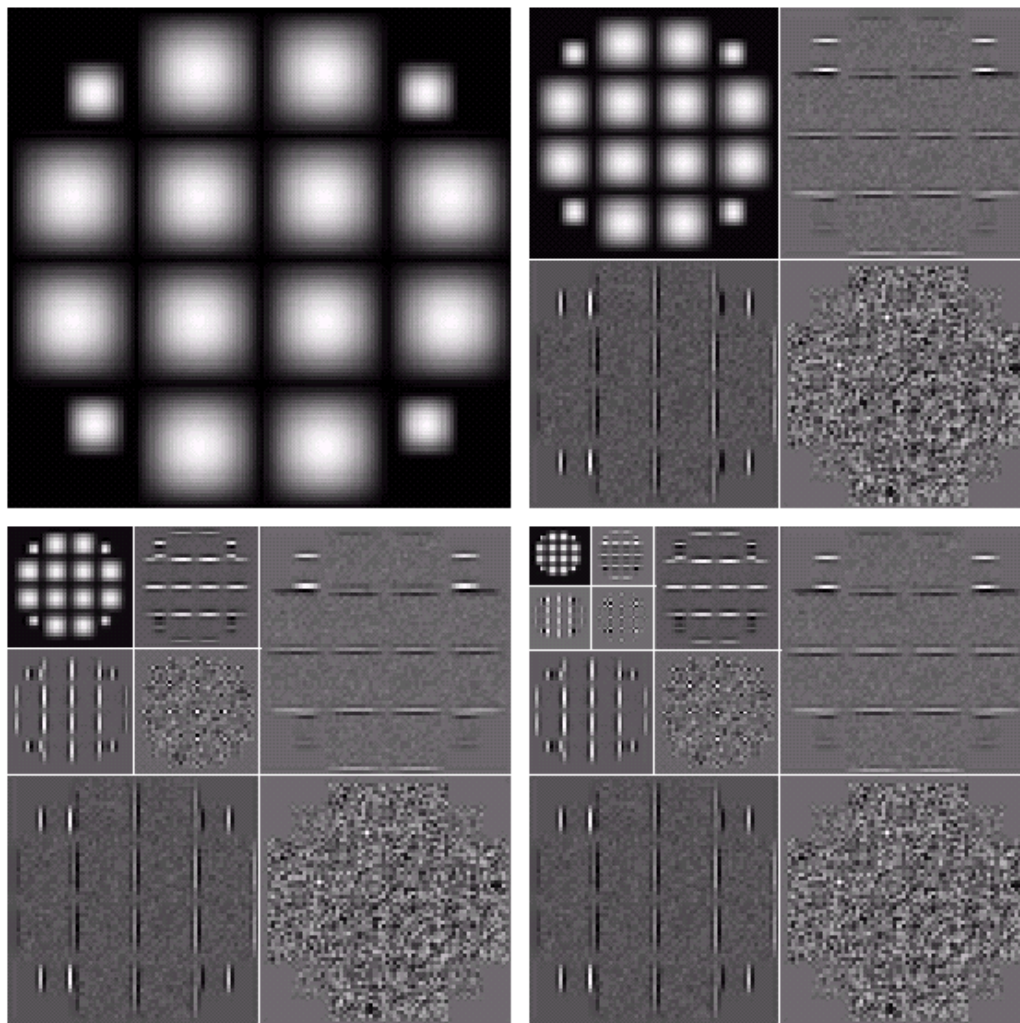
Nb. Colors

Brightness

2-D Wavelet Transform



Przykład



Oprogramowanie

Wavelet software

Matlab Wavelet Toolbox

Free software

UviWave http://www.tsc.uvigo.es/~wavelets/uvi_wave.html

Wavelab <http://playfair.stanford.edu/~wavelab/>

Rice Tools <http://jazz.rice.edu/RWT/>