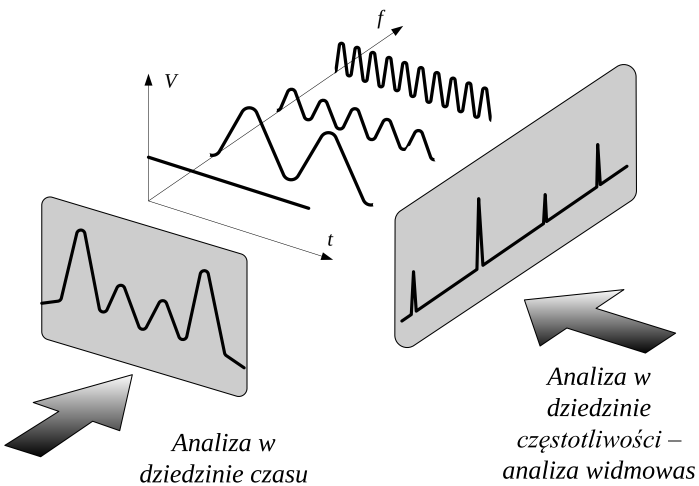
Dyskretna transformata Fouriera



Rys. 1. Transformata Fouriera pozwala na przejście z dziedziny czasu do dziedziny częstotliwości (i odwrotnie). W dziedzinie częstotliwości sygnał reprezentowany jest przez widmo – położenie prążków pokazuje z czego składa się sygnał.

Analiza Fouriera ma wielki znaczenie w przetwarzaniu sygnałów. Wile operacji jest znacznie prostszych w dziedzinie częstotliwości. Tak na przykład przetwarzanie obrazu w tomografii komputerowej byłoby nie możliwe bez transformaty Fouriera (Slice Fourier transform).

Jednocześnie rezultat analizy widmowej daje dużo więcej informacji o sygnale niż ten sam sygnał w dziedzinie czasu. Wiemy z jakich harmonicznych sygnał się składa i jakie są wartości tych harmonicznych. Możemy łatwo obliczyć zniekształcenie sygnału jako THD (total harmonic distortion) jako stosunek harmonicznych do składowej podstawowej

$$THD = \frac{\sqrt{\sum_{k=2}^{n} V_k^2}}{V_1} \cdot 100\%$$

DFT – opis matematyczny

Transformata Fouriera sygnału analogowego (continuous time FT - CTFT) jest opisana zależnością

$$X(\omega) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} x(t) e^{-j\omega t} dt$$

Jej odpowiednikiem w technice cyfrowej jest Discrete Time FT - DTFT opisana jako

$$X(\omega) = \frac{1}{2\pi} \sum_{n=-\infty}^{+\infty} x(n) e^{-j\omega n}$$

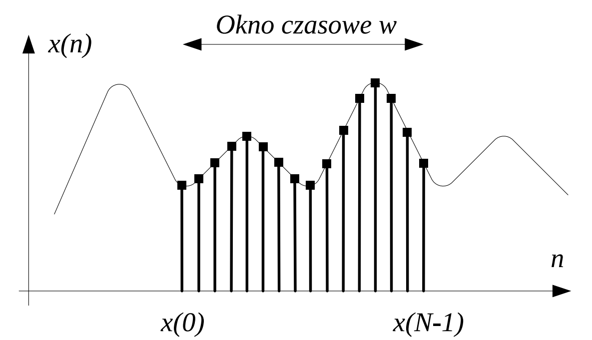
Taka operacja nie jest możliwa do realizacji (wyznaczanie od minus nieskończoności do plus nieskończoności) dlatego zaproponowano jej uproszczenie nazwane Discrete Fourier Transform DFT

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi nk}{N}}$$

Uproszczenie polega na tym że operację matematyczną przeprowadzamy na wybranej liczbie N próbek – ponieważ sygnał jest okresowy te N próbek traktujemy jako reprezentację sygnału (oczywiście ważne jest żeby tych próbek było dostatecznie dużo).

Wyboru próbek dokonujemy mnożąc sygnał przez równanie oka o postaci

$$w(n) = \begin{cases} 1 & for \quad n_0 \le n \le n_0 + N \\ 0 & for \quad other \quad samples \end{cases}$$



DFT – opis matematyczny cd

Równanie DFT

$$X(k)=\sum_{n=0}^{N-1} x(n)\, e^{-j2\pi nk/N}$$

Można też zapisać w innej postaci wiedząc że

$e^{j\varphi} = \cos\varphi - j\sin\varphi$

$$X(k)=\sum_{n=0}^{N-1} x(n)\left[\cos k(2\pi nk/N) - j\sin(2\pi nk/N)\right]$$

Widzimy że rezultat transformaty Fouriera (zwyczajowo sygnał w dziedzinie częstotliwości zapisujemy dużymi literami, w dziedzinie czasu małymi) jest liczbą zespoloną (dla celów analizy widmowej posługujemy się modułem tej liczby, ale już dla celów analizy impedancji składowa rzeczywista i urojona ma swoje znaczenie.

Spotyka się też inny opis DFT, a mianowicie

$$X(k)=X(\omega_s)=\sum_{n=0}^{N-1} x(n) e^{-jn\omega_s/N}$$

Lub

$$X(k)=X\left(\frac{\omega_s}{N}\right)=\sum_{n=0}^{N-1} x(n) e^{-jn\omega_s}$$

Równanie odwrotnej transformaty Fourier (Inverse Fourier Transform) a więc powrót z dziedziny częstotliwości do dziedziny czasu ma postać

$$x(n)=\frac{1}{N}\sum_{m=1}^{N-1} X(m) e^{j2\pi nm/N}$$

FFT – szybka transformata Fouriera

Jak wynika z równań przedstawionych na poprzednim slajdzie obliczenie DFT wymaga obliczania setek sinusoid i cosinusoid. Stąd liczenie DFT jest operacją off-line - pobieramy N próbek i dokonujemy na nich obliczeń (dawniej obliczanie zajmowało godziny - dziś przy szybkich procesorach trwa us i powstaje wrażenie że jest to operacja on-line).

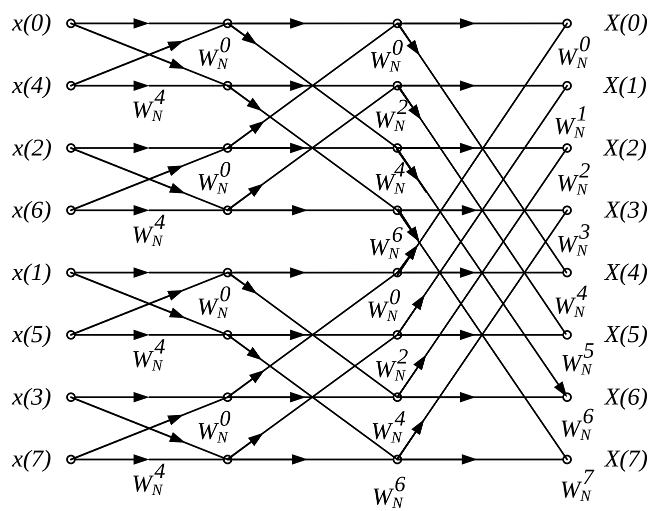
Tak na przykład obliczenie DFT dla 1000 próbek wymaga 1 048 576 mnożeń i 1 047 552 dodawań.

Przełomem było zaproponowanie przez Cooleya i Tuckeya algorytmu liczenia nazwanego FFT – Fast Fourier Transform.

Zaobserwowali oni że jeśli liczba próbek jest 2^N a więc 64, 128, 256, 512 czy 1024 to obliczenia można znacznie uprościć. W algorytmie FFT obliczanie wyniku dla 1024 próbek wymaga tylko 5 120 mnożeń i 10 240 dodawań.

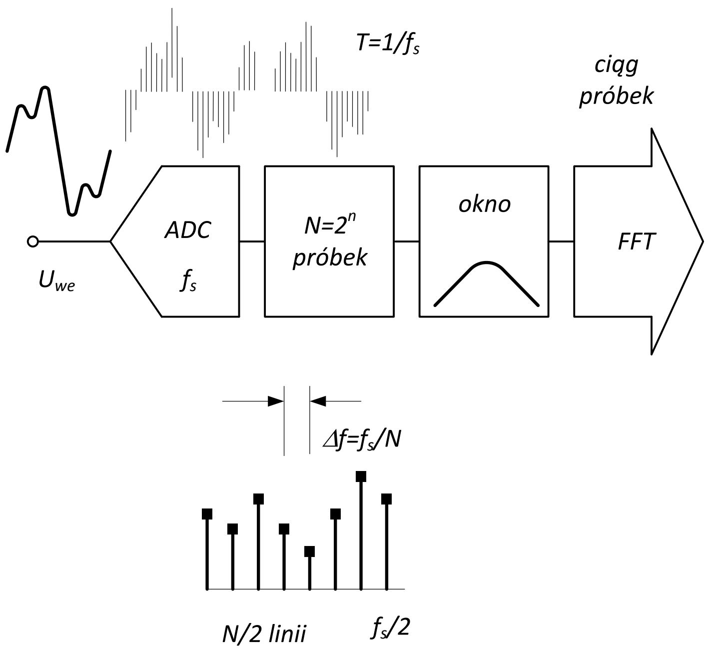
Co ważne algorytm FFT nie jest uproszczeniem DFT tylko innym sposobem liczenia (przy liczeniu FFT otrzymamy dokładnie tens sam wynik co dla DFT).

Algorytm FFT jest dziś klasyką informatyki , jest dostępny dla różnych języków oprogramowania i jest dostępny jako gotowy produkt (w arkuszach kalkulacyjnych, MatLabie czy nawet w kalkulatorach).



Rys. 3. Przykładowy graf liczenia FFT

Analiza widmowa sygnału (analiza spektralna)



Rys.4. Analiza widmowa sygnału

Rozpatrzmy przypadek że chcemy dokonać analizy widmowej sygnału analogowego. Najpierw zamieniamy sygnał w przetworniku A/D – otrzymujemy ciąg próbek z odstępem okresu próbkowania Ts.

Następnie wybieramy N próbek z wykorzystaniem funkcji okna. Tych próbek powinno być: 64 lub 128 lub 256 lub 512 lub 1024. Transformata z 1024 próbek uchodzi za bardzo dokładną.

Te N próbek wstawiamy do algorytmu FFT i w rezultacie otrzymujemy N/2 próbek z których ostatnia ma częstotliwość Nyquista fs/2 (w rzeczywistości otrzymujemy nieskończoną liczbę próbek ale powtarzanych z okresem fs/2 a więc wybieramy tylko pierwszy zestaw.

Ważne: próbki na wyjściu są z odstępem fs/N.

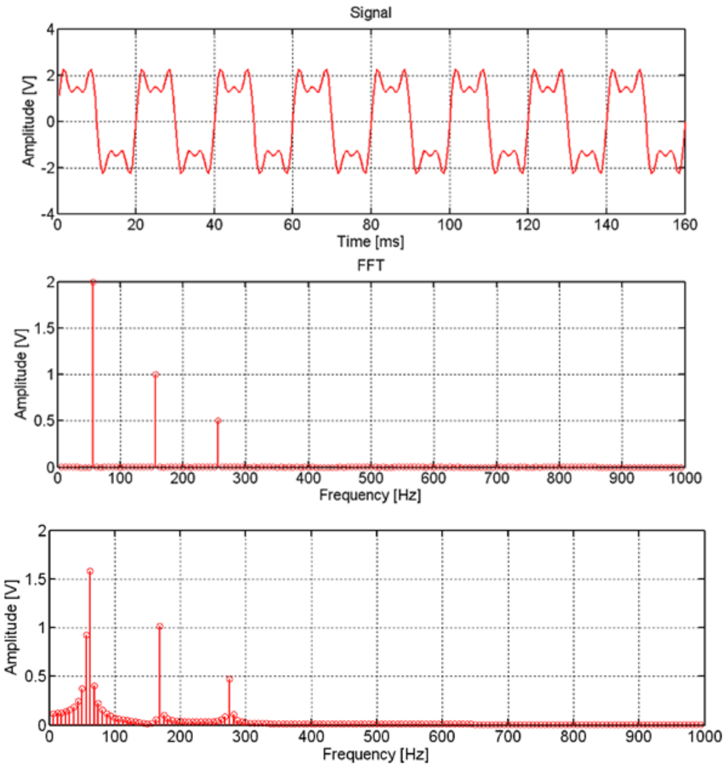
A więc na wejściu FFT mamy ciąg próbek:

$$x(kT_s) = x(0), x(T_s), x(2T_s), \ldots, x((N-1)T_s)$$

A na wyjściu mamy ciąg:

$$X\left(\frac{nf_s}{N}\right) = X(0), X\left(\frac{f_s}{N}\right), X\left(\frac{2f_s}{N}\right), \ldots, X\left(\frac{(N-1)f_s}{N}\right)$$

Analiza synchroniczna i asynchroniczna



Rys. 5. Ten sam sygnał analizowany synchronicznie i asynchronicznie

Ostęp między próbkami na wyjściu FFT zależy od nas (!) – jakie wybraliśmy fs i jakie N (odstęp jest fs/N). Wyobraźmy sobie że mamy sygnał 50 Hz + jego harmoniczne i wybraliśmy fs = 50 kHz oraz N= 10. A odstęp między próbkami jest 50 Hz więc na ekranie pojawią się prążki reprezentujące wszystkie harmoniczne. Mówimy że mamy analizę synchroniczną.

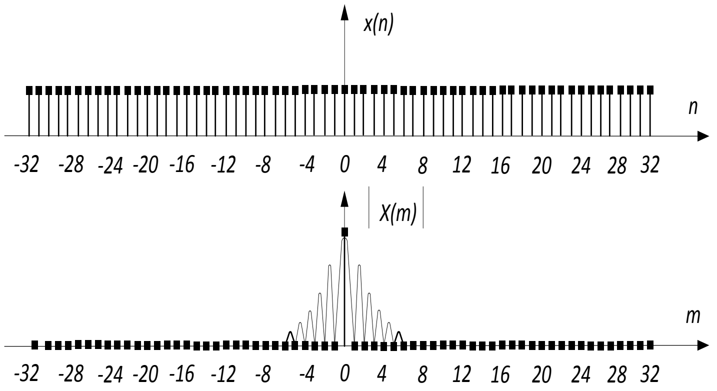
Ale jeśli wybierzemy fs = 100 kHz to odstęp będzie 100 Hz a więc tylko harmoniczne parzyste będą widoczne , harmoniczne nieparzyste nie będą miały swego prążka. Zachodzi wówczas zjawisko przecieku (leakage) polegające na tym że nieobecne prążki są reprezentowane przez sąsiadów.

Nie mamy więc klarownego obrazu harmonicznych tylko obraz rozmyty (zamiast prążka trójkąt) jak to ilustruje rys. 5.

Na ogół sytuacja że dojdzie do analizy synchronicznej jest rzadkością, częściej mamy analizę asynchroniczną.

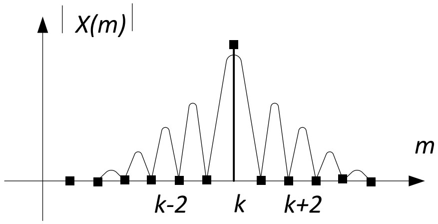
Okno prostokątne – funkcja sinx/x

Rys. 6. Ponieważ wybraliśmy okno prostokątne to na nasz sygnał wyjściowy nakłada się transformata sygnału prostokątnego sinx/x.

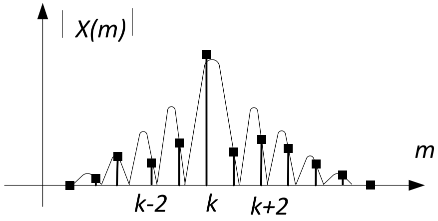


Rys. 7. Nakładanie się funkcji sinx/x na sygnał wyjściowy w przypadku analizy synchronicznej i asynchronicznej.

a)



b)

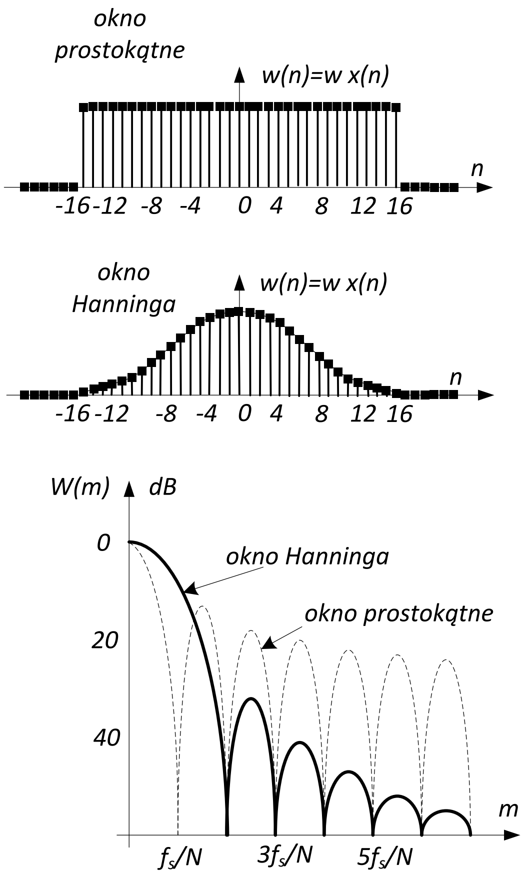


Jeśli mamy analizę synchroniczną to efekt nakładania się funkcji sinx/x jest niezauważalny ponieważ prążki wypadają w punktach zero funkcji sinx/x - rys. 7a

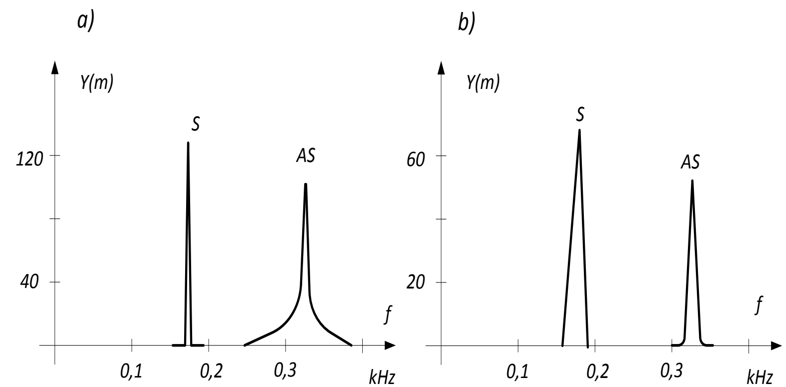
Jeśli mamy analizę asynchroniczną to nałożenie się funkcji sinx/x skutkuje jeszcze większym rozmyciem sygnału wyjściowego

Funkcja okna

Rys.8. Nakładające się listki boczne dla różnych funkcji okna



Może więc okno prostokątne nie było dobrym wyborem? Rzeczywiści dla innego okna, np. okna Hanninga listki boczne szybciej zanikają. Ale dzieje się to kosztem szerokości pierwszego listka a więc kosztem jakości prążka.
A więc w przypadku analizy asynchronicznej zastosowanie okna Hannninga poprawi nam rezultat analizy, ale w przypadku analiza y synchronicznej pogorszy (rys.9). Należy więc rodzaj okna dobierać doświadczalnie.



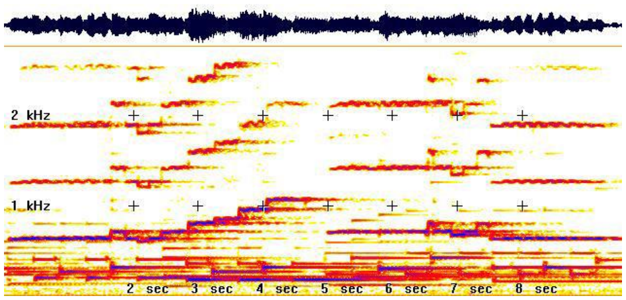
Rys. 9. Rezultat analizy dla okna prostokątnego (a) i Hanninga (b)

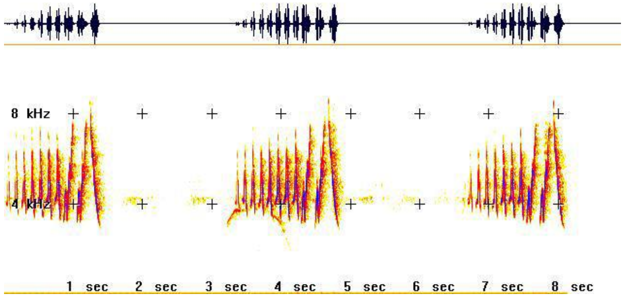
Analiza czasowo-częstotliwościowa - spektrogram

Wadą analizy Fouriera jest to, że dotyczy ona sygnału stacjonarnego – nie zmieniającego się w czasie. A więc jeśli dokonamy analizy pasażu muzycznego to oczywiście będziemy mieli informację z jakich harmonicznych on się składa natomiast nie będziemy mieli informacji kiedy to się zdarzyło – innymi słowy po zastosowaniu odwrotnej transformaty Fouriera nie odzyskamy naszego dźwięku.

Rozwiązaniem tego problemu byłoby przedstawienie sygnału dwuwymiarowo – gdzie na osi x byłby czas a na osi y analiza widmowa. Przykłady takich obrazów zwanych spektrogramem przedstawiono na rys. 10.

Rys. 10. Spektrogam sygnału gry na flecie i śpiewu ptaka.





Short Time Fourier Transform - STFT

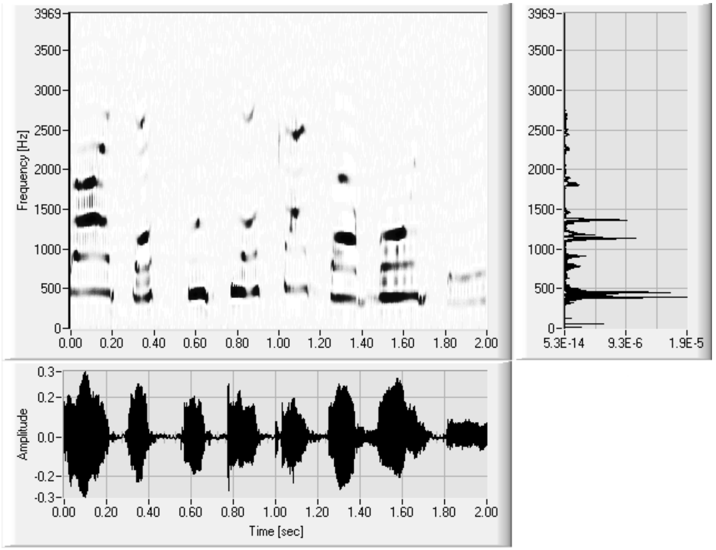
Przedstawiony na poprzednim slajdzie spektrogram można konstruować stosując tzw transfrmatę krótkoczasową (lub z oknem ślizgowym) – short time Fourier transform

$$X(n,k) = \sum_{m=0}^{N-1} w(m)x(n-m)e^{-jm(2\pi k/N)}$$

W transformacie tej okno czasowe przesuwa się i dokonywana jest analiza dla danego czasu. Niestety im węższe okno tym lepsza analiza w czasie ale gorsza w częstotliwości (i odwrotnie) gdyż:

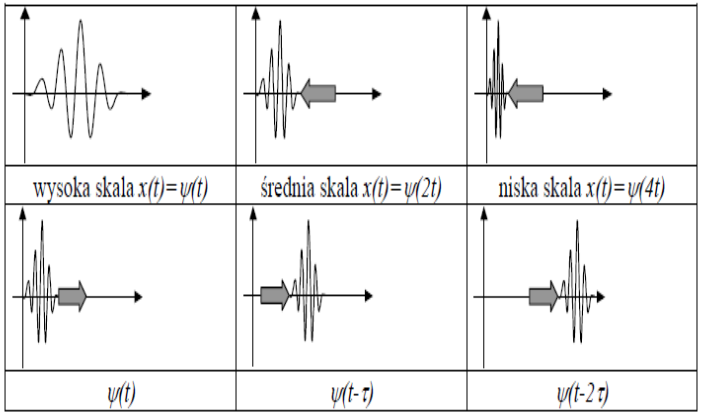
$$\Delta t \cdot \Delta\omega = const$$

Rys. 11. Przykład spektrogramu wykonanego z użyciem transformaty STFT



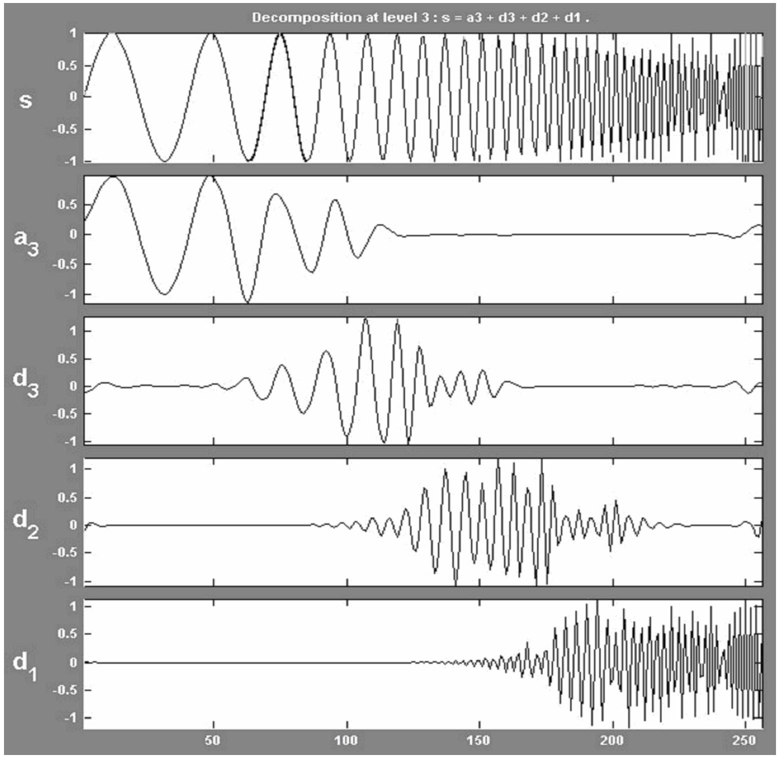
Analiza falkowa

Rozwiązaniem problemu analizy sygnałów zmiennych w czasie jest analiza falkowa (wavelet analysis) w której sygnał składamy nie z sinusoid ale z falek. Opracowano różnego rodzaju falko – Morlet, Daubechies itd.



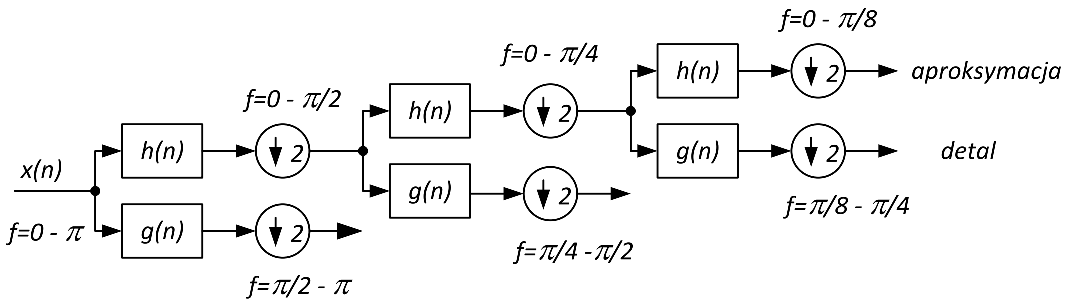
Rys. 12. Falki mogą mieć różną częstotliwość i czas.

Rys. 13. Jako rezultat otrzymujemy sumę kolejnych falek – pierwsza jest aproksymacją, kolejne sa detalami



Algorytm Mallata

Powszechnie stosowana metoda analizy falkowej jest algorytm Mallata zwany też szybka transformata falkową.



Rys. 14. Algorytm Mallata

Algorytm Mallata składa się z zestawu filtrów cyfrowych dolnoprzepustowego i górnoprzepustowego. W filtrze zapisany jest kształt falki. Poszczególne filtry dostarczają aproksymację i detale.

Metoda Mallata jest odwracalna to znaczy możemy z transformaty falkowej odzyskać sygnał analogowy.