copyright: M. Dwornik
dwornik@agh.edu.pl

Analiza czasowo-częstotliwościowa

Transformata Fouriera analizuje sygnał globalnie i nie pozwala na rozstrzygnięcie o zmianie składu częstotliwościowego sygnału w czasie. Tym samym jej zastosowanie do analizy takich sygnałów jak "sweep-y" lub sygnały o modulacji częstotliwościowej, jest praktycznie ograniczone. Do analizy tego typu sygnałów służą metody czasowo-częstotliwościowe, takie jak np. STFT (ang. *Short Time Fourier Transform*) czy transformata falkowa (ang. *wavelet*).

1 Krótkookresowa transformata Fouriera

Transformata Fouriera analizuje sygnał globalnie i nie pozwala na rozstrzygnięcie o zmianie składu częstotliwościowego sygnału w czasie. Tym samym jej zastosowanie do analizy takich sygnałów jak "sweep-y" lub sygnały o modulacji częstotliwościowej, jest praktycznie ograniczone. Do analizy tego typu sygnałów służą metody czasowo-częstotliwościowe, takie jak np. STFT (ang. *Short Time Fourier Transform*). Metoda STFT polega na przesuwania okna o zadanej długości (zwykle będącej potęgą 2) z pewnym krokiem wzdłuż sygnału i analizie składu częstotliwościowego (FFT) takiego sygnału. Ważną cechą tej metody jest rozdzielczość czasowo-częstotliwościowa: im większe okno, tym lepsza rozdzielczość częstotliwościowa oraz gorsza czasowa. I odwrotnie: małe okno czasowe implikuje dobrą rozdzielczość czasową i gorszą częstotliwościową.

2 Transformata Falkowa

Transformata falkowa polega na dekompozycji sygnału na część aproksymującą i detale przy użyciu konkretnej falki (wavelet). Odwrotna transformata falkowa polega na rekonstrukcji sygnału bazując na jego składowych. Podobnie jak transformata Fouriera, występuje ona w formie ciągłej (cwt) i dyskretnej (dwt). Transformata falkowa ma zastosowanie w kompresji (np. JPEG 2000), odszumianiu sygnałów czy fuzji obrazów.

2.1 Falki

Falka bazowa ψ jest to sygnał (filtr) oscylujący, szybko zanikający o średniej wartości równej zero. Na podstawie takiej falki tworzone są całe rodziny falek polegające na skalowaniu i przesuwaniu falki bazowej:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right) \tag{1}$$

Do najważniejszych falek zaliczamy:

- `'Haar'`: najprostsza falka, będąca jednocześnie ortogonalną i asymetryczną funkcją. Dana jest wzorem:
$$\psi(t) = \begin{cases} 1 & 0 \leqslant t < 0.5 \\ -1 & 0.5 \leqslant t < 1 \end{cases} \tag{2}$$
- Daubechies (`'db1'` - `'db9'`) - rodzina falek ortogonalnych (`'db1'` jest tożsame z `'haar'`).
- Biortogonalne (`'biorX.Y'`, gdzie X,Y - momenty falki, np. `'bior3.5'`)
- Coiflets (`'coifN'`, gdzie 2N - moment falki)

copyright: M. Dwornik
dwornik@agh.edu.pl

- Symlets (`'symN'`) - prawie symetryczne falki, będące modyfikacją rodziny falek `db`.
- Morlet (`'morl'`)
- MexHat (`'mexh'`)

Informację na temat własności i stosowalności poszczególnych falek można uzyskać poprzez komendę `waveinfo('falka')`.

2.2 Dekompozycja sygnału

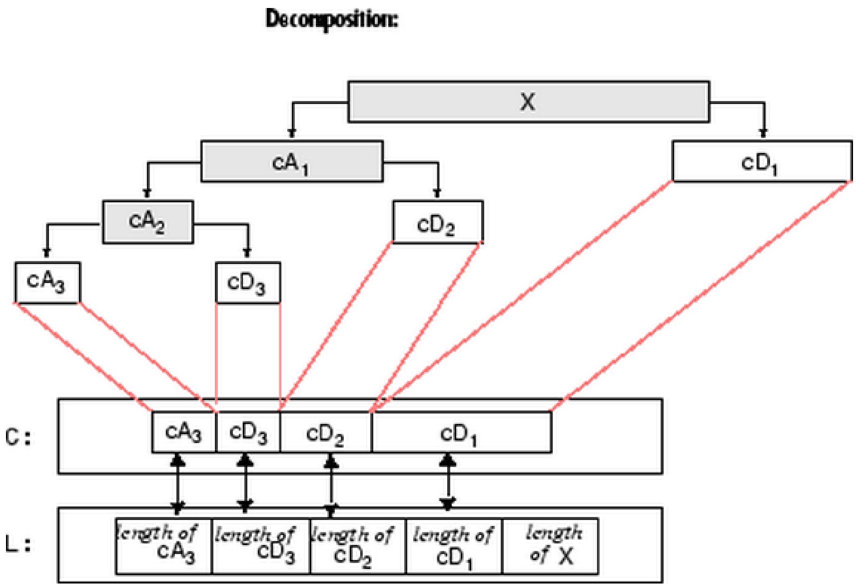
Dekompozycja (identyfikacja składowych) sygnału polega na przesuwaniu funkcji analizującej (falki) w dziedzinie czasu i przestrajaniu jej w dziedzinie częstotliwości. W przypadku sygnałów jednowymiarowych następuje dekompozycja na część aproksymującą (LP) i część detali (HP). Każda z tych części posiada rozmiar równy połowie sygnału wejściowego, który jest efektem downsamplingu. W przypadku dekompozycji wyższego rzędu, zawsze część aproksymująca będąca wynikiem dekompozycji niższego rzędu ulega rozkładowi (fig.1).

W pakiecie MatLAB dekompozycję sygnału wykonuje się korzystając z następującego polecenia (fig.1):

`[C,L]=wavedec(sygnal, poziom, 'falka');`, gdzie:

`C` - wartości wektorów dekompozycji (aproksymacji na N-tym poziomie i detali od N-tego do 1-ego poziomu)

`L` - długości wektorów dekompozycji (A_N, $D_N ... D_1$, długość sygnału)



Rysunek 1: Idea dekompozycji III rzędu sygnału oraz struktura danych wyjściowych w funkcji `wavedec`. Źródło: MatLAB Wavelet Toolbox.

Rekonstrukcja sygnału odbywa się przy użyciu następującej funkcji:

`sygnal = waverec(C, L, 'falka');`

W przypadku dekompozycji i rekonstrukcji I rzędu można wykorzystać funkcje: `dwt` oraz `idwt`.

3 Zadania

1. Korzystając z `wavemenu` wyświetl i porównaj wybrane falki. Dla falek `db` i `bior` określ wpływ wartości momentu / rzędu.

2. Stwórz sygnały ($t = 0 - 10$s, $F_s = 100$Hz)

   - x1(t) - suma sygnałów harmonicznych (A1=1, F1=10Hz; A2=0.5, F2=3.75; A3=0.75, F3=18Hz) oraz szumu gaussowskiego (σ=0.02);
   - x2(t) - sklejenie sygnałów harmonicznych z podpunktu 1 (f1 dla t=0-3s, f2 dla t=3-6s, f3 dla t=6-9s, sygnał zerowy dla t>9s) + szum;
   - x3(t) - sygnał harmoniczny o liniowym wzroście częstotliwości f(0s)=0.5Hz, f(10s)=5Hz i stałej amplitudzie A=1;
   - x4(t) - sygnał harmoniczny o f=5Hz i A=1 i wartości NaN dla t=4s i t=7s.

   Dla każdego takiego sygnału oblicz i wyświetl:

   - widmo amplitudowe transformaty Fouriera;
   - wynik STFT dla okna 0.512s i 2.048s;
   - skalogram (cwt) wybraną falką (`wscalogram`);
   - wynik dekompozycji na 1 i 3 stopniu wybraną falką.