FFT i dyskretny splot. Aplikacje w DSP

Marcin Jenczmyk
m.jenczmyk@knm.katowice.pl

Uniwersytet Śląski w Katowicach
Wydział Matematyki, Fizyki i Chemii

10 maja 2014

Transformata Fouriera

Transformata Fouriera wyraża się wzorem

$$\widehat{x}(f) = \int_{-\infty}^{\infty} x(t)e^{-2\pi ixf}\,dt,$$

Transformata Fouriera

Transformata Fouriera wyraża się wzorem

$$\widehat{x}(f) = \int_{-\infty}^{\infty} x(t) e^{-2\pi i x f} \, dt,$$

dla której odwrotna transformata wyraża się wzorem:

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \widehat{x}(f) e^{2\pi i x f} \, df.$$

Transformata Fouriera

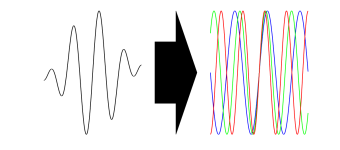
Transformata Fouriera wyraża się wzorem

$$\widehat{x}(f) = \int_{-\infty}^{\infty} x(t) e^{-2\pi i x f} dt,$$

dla której odwrotna transformata wyraża się wzorem:

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \widehat{x}(f) e^{2\pi i x f} df.$$

Transformata Fouriera pozwala na rozłożenie sygnału ma poszczególne składowe częstotliwościowe sygnału.



DFT

Dla dyskretnego sygnału określonego w N punktach x_i, $i = 0, \ldots, N-1$ transformata Fouriera przyjmuje postać

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi i \frac{kn}{N}},$$

DFT

Dla dyskretnego sygnału określonego w N punktach x_i, $i = 0, \ldots, N-1$ transformata Fouriera przyjmuje postać

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi i \frac{kn}{N}},$$

a transformata do niej odwrotna

$$x_n = \frac{1}{N} \sum_{n=0}^{N-1} X_k e^{2\pi i \frac{kn}{N}}.$$

DFT

Dla dyskretnego sygnału określonego w N punktach x_i, $i = 0, \dots, N-1$ transformata Fouriera przyjmuje postać

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi i \frac{kn}{N}},$$

a transformata do niej odwrotna

$$x_n = \frac{1}{N} \sum_{n=0}^{N-1} X_k e^{2\pi i \frac{kn}{N}}.$$

Tak zdyskretyzowana tranformata Fouriera nazywa się DFT (Discrete Fourier Transform).

DFT

Aby móc wykorzystać DFT w przetwarzaniu sygnałów potrzebny jest algorytm pozwalający na stosunkowo szybkie obliczenie DFT.

Algorytm 1 DFT(x,N) (algorytm brutalny)

Input: x : tablica zmiennych typu complex o długości N
Output: $\widehat{x}$: tablica zmiennych typu complex o długości N

```
for k = 0 to N − 1 do
    sum ← 0
    for n = 0 to N − 1 do
        ω ← e^(−2πi kn/N)
        sum ← sum + ω x_n
    end for
    x̂_k ← sum
end for
return x̂
```

FFT

Jednak jego złożoność takiego algorytmu jest rzędu $\Theta(n^2)$.

FFT

Jednak jego złożoność takiego algorytmu jest rzędu $\Theta(n^2)$. Stosunkowo łatwe spostrzeżenie pozwola na zredukowanie tego czasu do $\Theta(n \log n)$. Wzór na DFT można zapisać jako

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi i \frac{kn}{N}}$$

FFT

Jednak jego złożoność takiego algorytmu jest rzędu $\Theta(n^2)$. Stosunkowo łatwe spostrzeżenie pozwola na zredukowanie tego czasu do $\Theta(n \log n)$. Wzór na DFT można zapisać jako

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi i \frac{kn}{N}} = \sum_{n=0}^{\frac{N}{2}-1} x_{2n} e^{-2\pi i \frac{k(2n)}{N}} + \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} e^{-2\pi i \frac{k(2n+1)}{N}}$$

FFT

Jednak jego złożoność takiego algorytmu jest rzędu $\Theta(n^2)$. Stosunkowo łatwe spostrzeżenie pozwola na zredukowanie tego czasu do $\Theta(n \log n)$. Wzór na DFT można zapisać jako

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi i \frac{kn}{N}} = \sum_{n=0}^{\frac{N}{2}-1} x_{2n} e^{-2\pi i \frac{k(2n)}{N}} + \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} e^{-2\pi i \frac{k(2n+1)}{N}}$$

$$\sum_{n=0}^{\frac{N}{2}-1} x_{2n} e^{-2\pi i \frac{kn}{\frac{N}{2}}} + e^{-2\pi i \frac{k}{N}} \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} e^{-2\pi i \frac{kn}{\frac{N}{2}}}$$

FFT

Jednak jego złożoność takiego algorytmu jest rzędu $\Theta(n^2)$. Stosunkowo łatwe spostrzeżenie pozwola na zredukowanie tego czasu do $\Theta(n \log n)$. Wzór na DFT można zapisać jako

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi i \frac{kn}{N}} = \sum_{n=0}^{\frac{N}{2}-1} x_{2n} e^{-2\pi i \frac{k(2n)}{N}} + \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} e^{-2\pi i \frac{k(2n+1)}{N}}$$

$$\sum_{n=0}^{\frac{N}{2}-1} x_{2n} e^{-2\pi i \frac{kn}{\frac{N}{2}}} + e^{-2\pi i \frac{k}{N}} \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} e^{-2\pi i \frac{kn}{\frac{N}{2}}}$$

$$= X_k^e + e^{-2\pi i \frac{k}{N}} X_k^o, \; k = 0, \ldots, N-1$$

FFT

Pozostaje jeszcze zauważyć, że

$$\begin{cases} X_k & = X_k^e + e^{-2\pi i \frac{k}{N}} X_k^o \\ X_{k+\frac{N}{2}} & = X_k^e - e^{-2\pi i \frac{k}{N}} X_k^o \end{cases} \quad 0 \leqslant k < \frac{N}{2}$$

Bez straty ogólności załóżmy, że $N = 2^m,\ m \in \mathbb{N}$. Jeżeli tak nie jest to do ciągu $x_i,\ i = 0, \ldots, N-1$ można dopisać zera, aż do najbliżej potęgi 2.

Ponadto DFT ciągu składającego się z jednej próbki to identyczność. Wobec tego można sformułować poniższy algorytm.

FFT

Algorytm 2 FFT(x,N)

Input: x : tablica zmiennych typu complex o długości $N = 2^m$

Output: $\widehat{x}$: tablica zmiennych typu complex o długości N

if N=1 **then**
    return x
end if
$\omega,\ \omega_N \leftarrow 1,\ e^{-i\frac{2\pi}{N}}$
$x^e,\ x^o \leftarrow (x_0, x_2, \dots, x_{N-2}),\ (x_1, x_3, \dots, x_{N-1})$
$\widehat{x}^e,\ \widehat{x}^o \leftarrow FFT(x^e, \frac{N}{2}),\ FFT(x^o, \frac{N}{2})$
for $k = 0$ to $\frac{N}{2} - 1$ **do**
    $\widehat{x}_k \leftarrow \widehat{x}^e_k + \omega \widehat{x}^o_k$
    $\widehat{x}_{k+\frac{N}{2}} \leftarrow \widehat{x}^e_k - \omega \widehat{x}^o_k$
    $\omega \leftarrow \omega \omega_N$
end for
return $\widehat{x}$

FFT

Złożoność powyższego algorytmu to $\Theta(n \log n)$.

FFT

Złożoność powyższego algorytmu to $\Theta(n \log n)$. Zauważmy że działania wykonywane w liniach $7-11$ mają złożoność liniową. Wobec tego złożoność całego algrytmu wynosi

$$T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n).$$

FFT

Złożoność powyższego algorytmu to $\Theta(n \log n)$. Zauważmy że działania wykonywane w liniach $7 - 11$ mają złożoność liniową. Wobec tego złożoność całego algrytmu wynosi

$$T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n).$$

Po rozwikłaniu tej rekurencji (np. za pomocą tw. o rekurencji uniwersalnej) dostajemy $T(n) = \Theta(n \log n)$.

FFT

Złożoność powyższego algorytmu to $\Theta(n \log n)$. Zauważmy że działania wykonywane w liniach $7 - 11$ mają złożoność liniową. Wobec tego złożoność całego algrytmu wynosi

$$T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n).$$

Po rozwikłaniu tej rekurencji (np. za pomocą tw. o rekurencji uniwersalnej) dostajemy $T(n) = \Theta(n \log n)$.

Jest to szybka metoda na obliczenie DFT, co pozwala na wykonywanie operacji na sygnale w dziedzinie częstotliwości.

Dyskretny splot

Splot dwóch funkcji x i h definiujemy wzorem

$$(x * h)(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau.$$

Dyskretny splot

Splot dwóch funkcji x i h definiujemy wzorem

$$(x * h)(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau.$$

Jeśli sygnał określony jest w punktach $x_i, \; i = 0, \ldots, N-1$, to dyskretnym analogiem powyższej definicji jest

$$(x * h)_n = \sum_{m=0}^{N-1} x_m h_{n-m}.$$

Dyskretny splot

Splot dwóch funkcji x i h definiujemy wzorem

$$(x * h)(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau.$$

Jeśli sygnał określony jest w punktach $x_i,\ i = 0, \dots, N-1$, to dyskretnym analogiem powyższej definicji jest

$$(x * h)_n = \sum_{m=0}^{N-1} x_m h_{n-m}.$$

Bezpośrednie obliczenie takiej sumy ma złożoność $\Theta(n^2)$. Istnieje jednak szybszy sposób na obliczenie dyskretnego splotu dwóch funkcji.

Dyskretny splot

Korzystając z faktu, że

$$\widehat{x * h} = \widehat{x}\widehat{h}$$

dyskretny splot można obliczyć w czasie $\Theta(n \log n)$.

Konwersja sygnału analogowego na cyfrowy

Wartości występujące w przyrodzie są na ogół ciągłe (analogowe). Aby móc korzystać z metod cyfrowego przetwarzania w celu ich obróbki należy dokonać wcześniejszej konwersji sygnału analogowego na cyfrowy. Odbywa się to w trzech etapach:

Konwersja sygnału analogowego na cyfrowy

Wartości występujące w przyrodzie są na ogół ciągłe (analogowe). Aby móc korzystać z metod cyfrowego przetwarzania w celu ich obróbki należy dokonać wcześniejszej konwersji sygnału analogowego na cyfrowy. Odbywa się to w trzech etapach:

- próbkowania;

Konwersja sygnału analogowego na cyfrowy

Wartości występujące w przyrodzie są na ogół ciągłe (analogowe). Aby móc korzystać z metod cyfrowego przetwarzania w celu ich obróbki należy dokonać wcześniejszej konwersji sygnału analogowego na cyfrowy. Odbywa się to w trzech etapach:

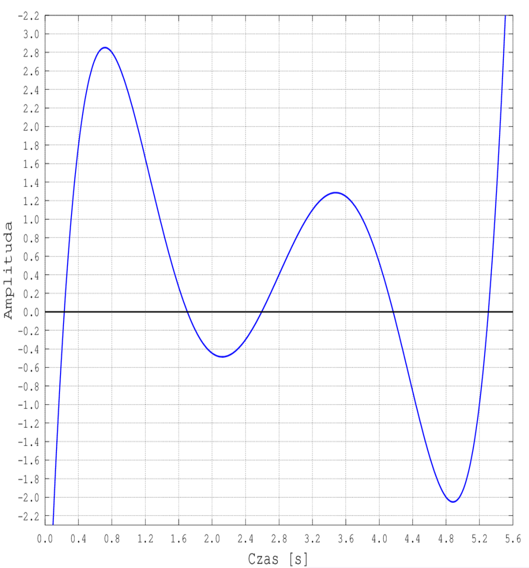
- próbkowania;
- kwantyzacji;

Konwersja sygnału analogowego na cyfrowy

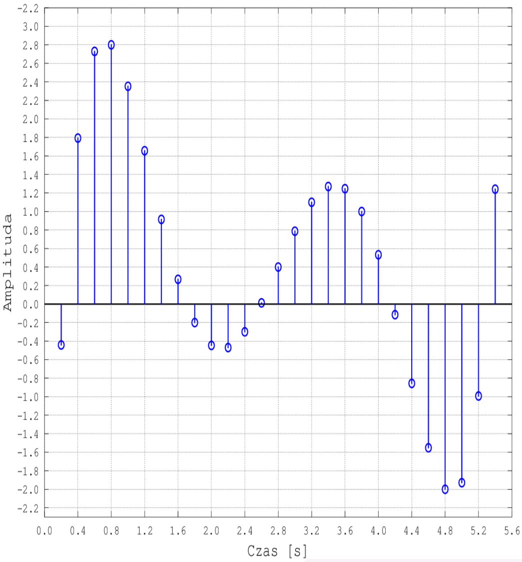
Wartości występujące w przyrodzie są na ogół ciągłe (analogowe). Aby móc korzystać z metod cyfrowego przetwarzania w celu ich obróbki należy dokonać wcześniejszej konwersji sygnału analogowego na cyfrowy. Odbywa się to w trzech etapach:

- próbkowania;
- kwantyzacji;
- kodowania;

Konwersja sygnału analogowego na cyfrowy

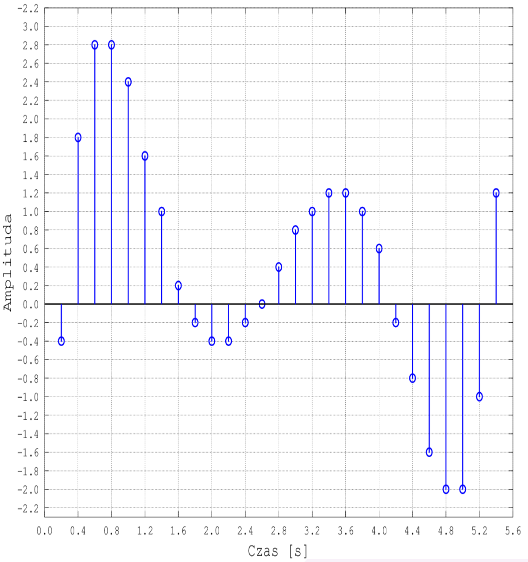


Próbkowanie



Próbkowanie polega na pobieraniu próbek z sygnału analogowego z częstotliwością próbkowania, w wyniku którego z sygnału ciągłego uzyskuje się ciąg próbek w postaci impulsów o wartościach określonych w dyskretnych punktach i o amplitudach równych amplitudom sygnału analogowego w tych punktach.

Kwantyzacja



Kwantyzacja dyskretnego sygnału otrzymanego w procesie próbkowania polega na podziale zakresu amplitud, jakie mogą przybierać próbki, na określoną liczbę przedziałów i przyporządkowanie każdej próbce numeru poziomu, do którego sięga jej amplituda.

Kodowanie

Kodowanie skwantowanego sygnału polega na przekształceniu go w sygnał cyfrowy przez przyporządkowanie poszczególnym poziomom tego sygnału, wyrażonym w liczbach układu dziesiętnego, numerów wyrażonych w systemie binarnym.

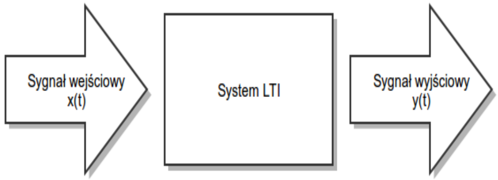
Kodowanie

Kodowanie skwantowanego sygnału polega na przekształceniu go w sygnał cyfrowy przez przyporządkowanie poszczególnym poziomom tego sygnału, wyrażonym w liczbach układu dziesiętnego, numerów wyrażonych w systemie binarnym.

Na ogół zakres amplitud dzieli się na 65535 ($2^{16} - 1$) przedziałów, pozwala to na zapisanie do 16-bitowych integerów.

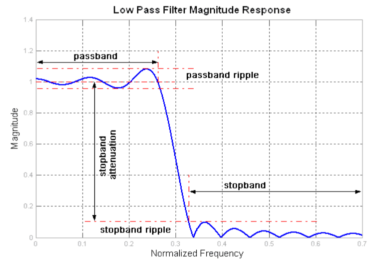
Filtry splotowe

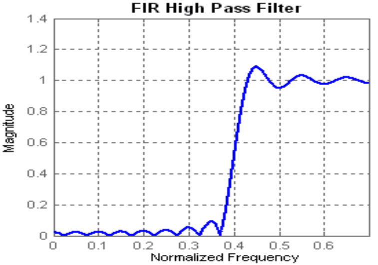
Operacji na sygnałach można dokonywać za pomocą filtrów cyfrowych. W tym celu wyobraźmy sobie układ, który po podaniu na wejście sygnału na wyjściu podaje sygnał przetworzony przez ten układ.

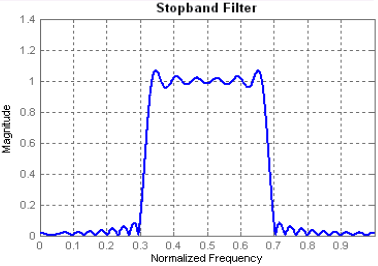


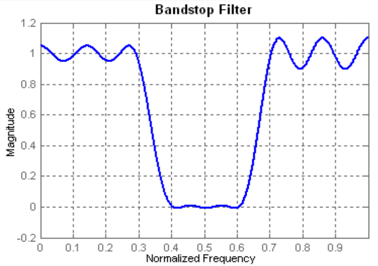
W teorii przetwarzania sygnałów stosuje się m.in. takie układy, które splatają sygnał z innym, określonym przez układ ciągiem. Jeśli taki układ jest filtrem splotowym, to ciąg ten będziemy nazywać współczynnikami filtra splotowego.

Filtry FIR


Low Pass Filter Magnitude Response
passband
passband ripple
stopband attenuation
stopband
stopband ripple
Magnitude
Normalized Frequency


FIR High Pass Filter
Magnitude
Normalized Frequency


Stopband Filter
Magnitude
Normalized Frequency


Bandstop Filter
Magnitude
Normalized Frequency