# Szybka transformata sinusowa

Yuliya Zviarko

22.05.2024

## Spis treści

# 1 Transformacja Fouriera

Jeśli funkcja $f(x)$ jest okresowa, wówczas możemy ją rozwinąć w Szereg Fouriera postaci:

$$\begin{gathered} f(x) = \tfrac{a_0}{2} + \sum_{k=1}^{\infty}(a_k cos(kx) + b_k sin(kx)), \\ gdzie: \\ a_k = \tfrac{1}{\pi}\int_{-\pi}^{\pi} f(x)cos(kx)dt, \\ b_k = \tfrac{1}{\pi}\int_{-\pi}^{\pi} f(x)sin(kx)dt. \end{gathered} \tag{1}$$

Funkcję możemy też zapisać w postaci Zespolonego Szeregu Fouriera:

$$\begin{gathered} f(x) \sim \sum_{k=-\infty}^{\infty} \hat{f}(k)e^{Ikx}, \\ \hat{f}(k) = \tfrac{1}{2\pi}\int_{-\pi}^{\pi} f(x)e^{-Ikx}dx. \end{gathered} \tag{2}$$

Jeśli funkcja $f(x)$ jest rzeczywista wówczas szereg podany wzorem (1) jest częścią rzeczywistą zespolonego szeregu Fouriera.

# 2 FFT - szybka transformacja Fouriera

Przekształcenia Fouriera służą do rozkładu sygnału na częstotliwości składowe, podobnie jak pryzmat rozszczepiający białe światło na składowe barwy lub okulary przeciwsłoneczne, które redukują oślepianie białym światłem, przepuszczając tylko łagodniejsze światło zielone.
Algorytm szybkiego przekształcenia Fouriera (FFT) efektywnie oblicza współczynniki $c_k$ poprzez następujące równanie:

$$c_k = \frac{1}{N-1}\sum_{j=0}^{N-1} f(x_j)\left(e^{-2\pi i/N}\right)^{kj},$$

które można wyrazić jako sumę wartości pewnego wielomianu stopnia $N-1$ w punktach $\lambda^k$. Koszt obliczeń to około $N$ mnożeń i $N$ dodawań, co daje ogólny koszt $O(N^2)$ dla $N$ współczynników $c_k$. Jednak dzięki szybkiemu przekształceniu Fouriera koszt ten redukuje się do bardziej rozsądnej wielkości $N\log_2 N$.
Poniższa tabelka pokazuje, co to znaczy dla dużych wartości $N$ typowych w przetwarzaniu sygnałów:

| $N$ | $N^2$ | $N\log_2 N$ |
|---|---|---|
| 1024 | 1048576 | 10240 |
| 4096 | 16777216 | 16384 |
| 16384 | 268435456 | 229376 |

Dzięki zastosowaniu algorytmu typu "dziel i zwyciężaj"uzyskano zmniejszenie złożoności obliczeniowej. Szybkie przekształcenie Fouriera umożliwiło modyfikację sygnału w celu uzyskania pożądanego efektu oraz zastosowanie dyskretnych transformat sinusowych i kosinusowych do kompresji danych. Ma szerokie zastosowanie w przetwarzaniu sygnałów cyfrowych, obrazowaniu medycznym, rozpoznawaniu złóż ropy lub gazu oraz rozwiązywaniu równań różniczkowych.
Inne algorytmy FFT:

- Split-radix jest modyfikacją algorytmu Cooleya-Tukeya. W każdym kroku DFT obliczana jest jako suma DFT dla $N/2$ oraz dwóch DFT dla $N/4$. Jest to najszybszy znany algorytm FFT.
- Dyskretna Transformata Sinusowa (DST) oraz Dyskretna Transformata Kosinusowa (DCT) to transformacje, które warto stosować, gdy przeprowadzamy transformację na funkcjach rzeczywistych. Unikamy w ten sposób operacji na liczbach zespolonych, co jest kosztowne obliczeniowo.

# 3 Transformacja sinusowa

Transformacja sinusowa jest matematyczną operacją używaną do przekształcania ciągów danych dyskretnych na inne ciągi danych. Jest przydatna, gdy analizujemy sygnały, które są parzyste względem osi czasu, ponieważ transformacja sinusowa pozwala wydajnie analizować ich sinusoidalne składowe.
W zastosowaniach praktycznych, transformacja sinusowa jest używana do analizy i przetwarzania sygnałów, obrazów, oraz do kompresji danych. Jest szczególnie użyteczna w przypadkach, gdzie mamy do czynienia z sygnałami, które posiadają wyłącznie składowe sinusoidalne, co może obejmować sygnały akustyczne, sygnały wibracyjne, czy też inne rodzaje danych, których charakter jest dominowany przez sinusy.
Wzory transformacji sinusowej:

**Transformata sinusowa typu I (DST-I)**:

$$X_k = \sum_{n=0}^{N-1} x_n \sin\left(\frac{\pi(k+1)(n+1)}{N+1}\right), \quad k = 0, 1, \ldots, N-1$$

**Transformata sinusowa typu II (DST-II)**:

$$X_k = \sum_{n=0}^{N-1} x_n \sin\left(\frac{\pi(k+1)(n+1/2)}{N}\right), \quad k = 0, 1, \ldots, N-1$$

**Transformata sinusowa typu III (DST-III)**:

$$X_k = \sum_{n=0}^{N-1} x_n \sin\left(\frac{\pi(k+1/2)(n+1)}{N}\right), \quad k = 0, 1, \ldots, N-1$$

**Transformata sinusowa typu IV (DST-IV)**:

$$X_k = \sum_{n=0}^{N-1} x_n \sin\left(\frac{\pi(k+1/2)(n+1/2)}{N}\right), \quad k = 0, 1, \ldots, N-1$$

Te wzory są często stosowane w analizie i przetwarzaniu sygnałów oraz innych dziedzinach, gdzie transformacja sinusowa jest używana.

# 4 Zadanie do wykonania

## 4.1 Opis problemu

Zadaniem było zastosowanie szybkiej transformaty sinusowej do odszumienia sygnału periodycznego. Sygnał zaszumiony generowano zgodnie z poniższym algorytmem:

a) Sygnał okresowy nie zaszumiony miał postać:

$$y_o(i) = \sin(w \cdot i) + \sin(2w \cdot i) + \sin(3w \cdot i) \tag{3}$$

gdzie:
$i$ był numerem próbki sygnału (numerem elementu w wektorze),
$w$ oznaczał długość fali ($w = 2\frac{2\pi}{n}$, n - ilość próbek).

b) Stworzono zmienną losową imitującą szum zgodnie z równaniem:

$$a = 2 \cdot \text{sign}\left(\frac{X}{\text{RAND-MAX} + 1.0}\right), \tag{4}$$

gdzie:
$X$ był liczbą pseudolosową o rozkładzie równomiernym w przedziale (0,1),
a sign określaliśmy na podstawie losowej zmiennej $Y$ tak, że sign było równe $+1$, gdy $Y > -1/2$, i -1 w przeciwnym przypadku(albo jest równy).

c) Sygnał zaszumiony konstruowano przez dodanie szumu do sygnału nie zaszumionego, obliczając wartość $a$ dla każdego indeksu $i$ z osobna.

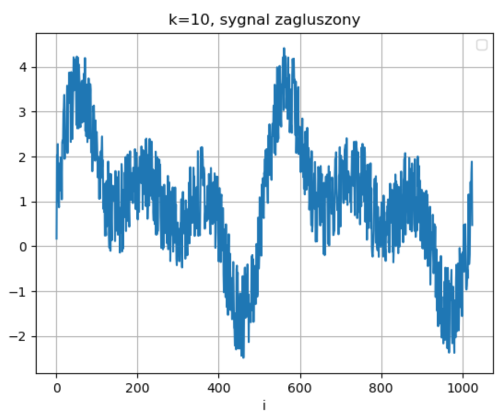
$$y(i) = y_0(i) + a \tag{5}$$

## 4.2 Wykonanie problemu

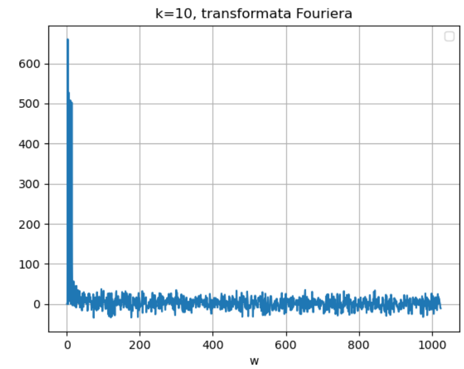
1. Zaszumiony sygnał został zapisany do wektora typu float o długości $n = 2k$, gdzie $k$ to liczba całkowita.
2. Następnie została wykonana transformata sinusowa sygnału korzystając z funkcji `sinft(float data[], int n)` z biblioteki *Numerical Recipes*.
3. Po wykonaniu transformaty dokonano dyskryminacji na poziomie 25 % wartości maksymalnej. Sygnał został wyzerowany w miejscach, gdzie amplituda była niższa od progu dyskryminacji.
4. Po dyskryminacji dokonano transformaty odwrotnej, ponownie korzystając z funkcji `sinft`, przy czym sygnał wyjściowy został przemnożony przez $2/n$, gdzie $n$ to długość sygnału.
5. Następnie zostały wykonane rysunki:

- Wykres sygnału zaszumionego dla $k = 10$.
- Wykres transformaty w pełnym zakresie.
- Wykres transformaty, na którym wyraźnie widoczne były piki pochodzące od dominujących modów w sygnale, znajdujących się na początku wykresu transformaty.
- Na jednym rysunku zostały narysowane sygnały niezaszumiony $y_o$ oraz sygnał po odszumieniu. Wykonano trzy takie wykresy dla $k = 6, 8, 10$.

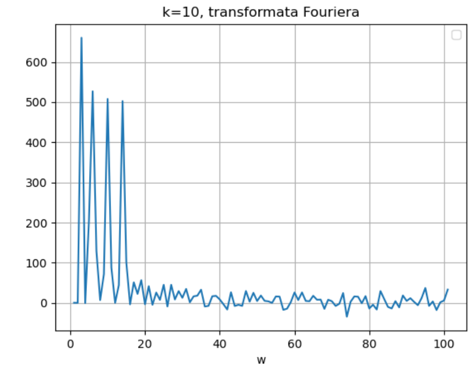
# 5 Wyniki



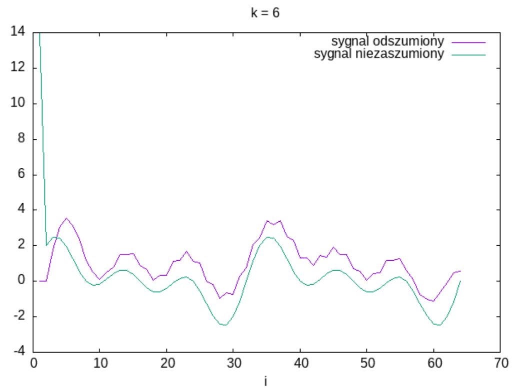

Rysunek 1: Sygnał zaszumiony dla k = 10.



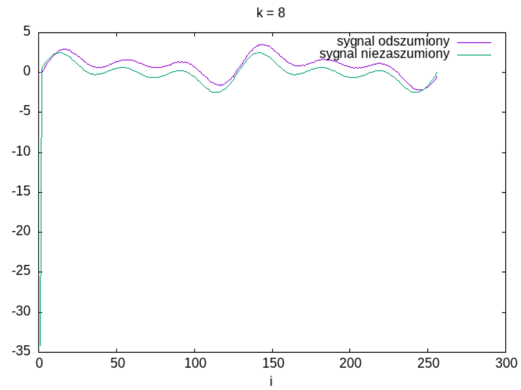

Rysunek 2: Transformata Fouriera sygnału dla $k = 10$ w pewnym zakresie.



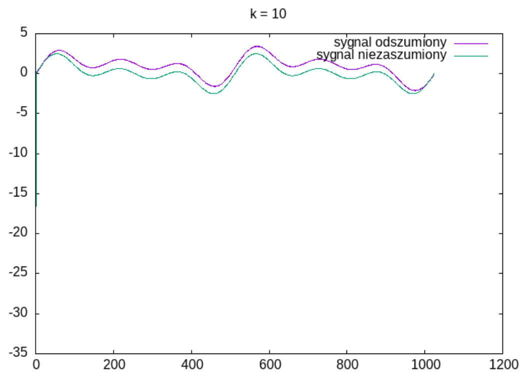

Rysunek 3: Transformata Fouriera sygnału dla $k = 10$ w zakresie z widocznymi pikami pochodzącymi od modów dominujących w sygnale.




Rysunek 4: (a)




Rysunek 5: (b)




Rysunek 6: (c)

Rysunek 7: Sygnał niezaszumiony $y_0$ i sygnał po odszumieniu dla (a) k = 6, (b) k = 8, (c) k = 10.

## 6 Wnioski

Z przeprowadzonej analizy wynika, że szybka transformacja sinusowa jest skutecznym narzędziem zarówno do analizy, jak i odszumiania sygnałów. Poniżej przedstawiono szczegółowe wnioski z uzyskanych wyników:

- **Zaszumienie sygnału:** Sygnał zaszumiony dla $k = 10$ wykazuje znaczące zakłócenia, które utrudniają jego analizę i interpretację. Jest to widoczne na Rysunku 1, gdzie szum znacząco wpływa na kształt sygnału.
- **Transformata Fourierowska:** Analiza transformaty Fourierowskiej w pełnym zakresie częstotliwości (Rysunek 2) pozwala zidentyfikować ogólną strukturę sygnału. Z kolei bardziej szczegółowy widok (Rysunek 3) ujawnia piki pochodzące od dominujących modów w sygnale. Te piki są kluczowe do identyfikacji i odszumiania sygnału, wskazując główne składowe częstotliwościowe.
- **Odszumianie sygnału:** Porównanie sygnałów odszumionych i niezaszumionych dla różnych wartości $k$ (Rysunek 7) pokazuje, że proces odszumiania jest skuteczny. Już od $k = 8$, odszumiony sygnał jest niemal identyczny z sygnałem niezaszumionym, co świadczy o wysokiej skuteczności metody odszumiania dla tej wartości $k$.
- **Wartość $k$:** Analiza wartości $k = 6, 8, 10$ pokazuje różne poziomy skuteczności odszumiania. Wyższe wartości $k$ (np. $k = 10$) lepiej oddzielają sygnał od szumu, co jest widoczne w lepszym dopasowaniu odszumionego sygnału do sygnału oryginalnego. Wskazuje to, że odpowiedni dobór parametru $k$ jest kluczowy dla uzyskania najlepszych rezultatów.

Podsumowując, szybka transformacja sinusowa okazuje się być efektywną metodą do analizy sygnałów, szczególnie w kontekście odszumiania. Wyniki wskazują, że właściwy dobór parametrów analizy, takich jak wartość $k$, jest istotny dla osiągnięcia optymalnych wyników. Dalsze badania mogłyby skupić się na optymalizacji parametrów STS dla różnych typów sygnałów i warunków zakłóceń.

# 7 Źródła

Podczas opracowywania tego sprawozdania korzystano z następujących źródeł:


- Kincaid, D., Cheney, W. (1996). *Numerical Analysis: Mathematics of Scientific Computing.*
- *Discrete sine transform.* (n.d.). W: Wikipedia. Dostępne na: `https://en.wikipedia.org/wiki/Discrete_sine_transform`.
- Chwiej, M. (2010). *Algorytm FFT.* W: Wykład z Metod Numerycznych. Dostępne na: `http://galaxy.agh.edu.pl/~chwiej/mn/wyk/fft_22_23.pdf`.