Vadym Semkovych

296669

20.05.2020

Sprawozdanie 11

Aproksymacja sygnału okresowego przy użyciu FFT

1. **Wstęp teoretyczny**

**Szybka transformacja Fouriera (Fast Fourier Transform, FFT)** – algorytm wyznaczania dyskretnej transformaty Fouriera oraz transformaty do niej odwrotnej.

Najpopularniejszą wersją algorytmu FFT jest FFT o podstawie 2. Jest on bardzo efektywny pod względem czasu realizacji, jednak wektor próbek wejściowych (spróbkowany sygnał) musi mieć długość gdzie 𝑘 to pewna liczba naturalna. Wynik otrzymuje się na drodze schematycznych przekształceń.

Najprostszy algorytm FFT to radix-2 (Cooley-Tukey). Algorytm polega na znalezieniu współczynników transformaty Fouriera (DFT) , ale wykonując jak najmniej obliczeń.

Zakładamy, że całkowita liczba węzłów jest potęgą 2:

Zatem:

Osobno grupujemy składniki:

Parzyste :

Niparzyste :

Korzystając z okresowości powyższych wyrazów nie musimy wyznaczać wszystkich współczynników – tylko połowę.

1. **Problem**

Naszym zadaniem było zastosowanie FFT do odszumienia sygnału periodycznego. Sygnał zaszumiony generowaliśmy zgodnie z poniższym wzorem:

gdzie:

Najpierw wygenerowaliśmy zaszumiony sygnał, zgodnie ze wzorem(5), i zapisaliśmy go do wektora typu *double.* Długość wektora wynosiła 2N, , kolejno dla

Następnie wyznaczyliśmy transformatę sygnału korzystając z funkcji biblioteki **GSL:**

gsl\_fft\_complex\_radix2\_forward (gsl\_complex\_packed\_array y, size\_t sride, size\_t N),

gdzie: y – tablica z sygnałem(typ gsl\_complex\_packed\_array odpowiada naszej jednowymiarowej tablicy),

stride – parametr równy 1(krok pomiędzy elementami w tablicy),

N –

Dalej przeprowadziliśmy dyskryminację sygnału na poziomie max(), to znaczy że trzeba było znaleźć element o maksymalnym module, i wyzerować wszystkie elementy których moduł był mniejszy od .

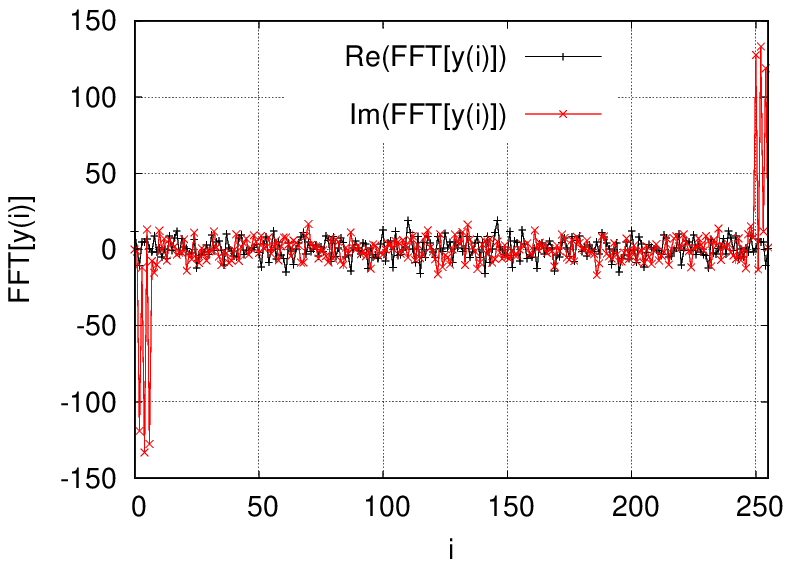
Ostatnie co zrobiliśmy to wyznaczyliśmy transformatę odwrotną korzystając z funkcji biblioteki **GSL:**

gsl\_fft\_complex\_radix2\_backward (gsl\_complex\_packed\_array y, size\_t sride, size\_t N),

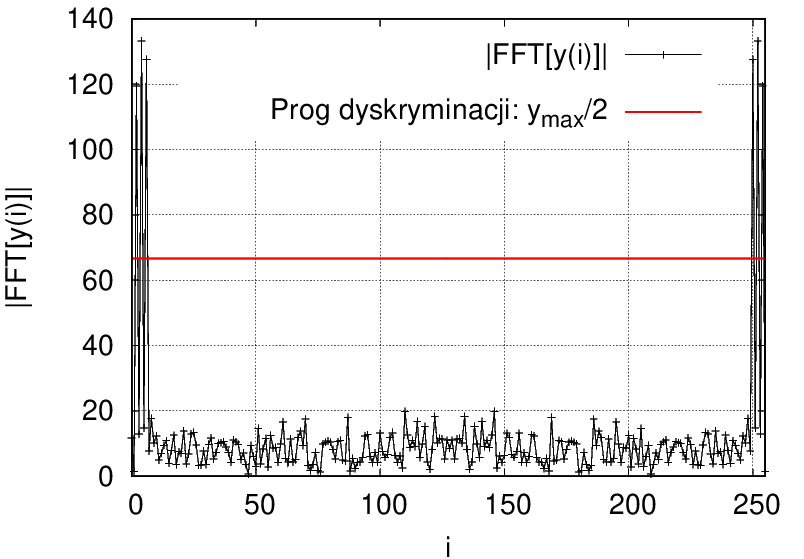
która przyjmuje argumenty analogiczne do funkcji gsl\_fft\_complex\_radix2\_forward.

1. **Wyniki**

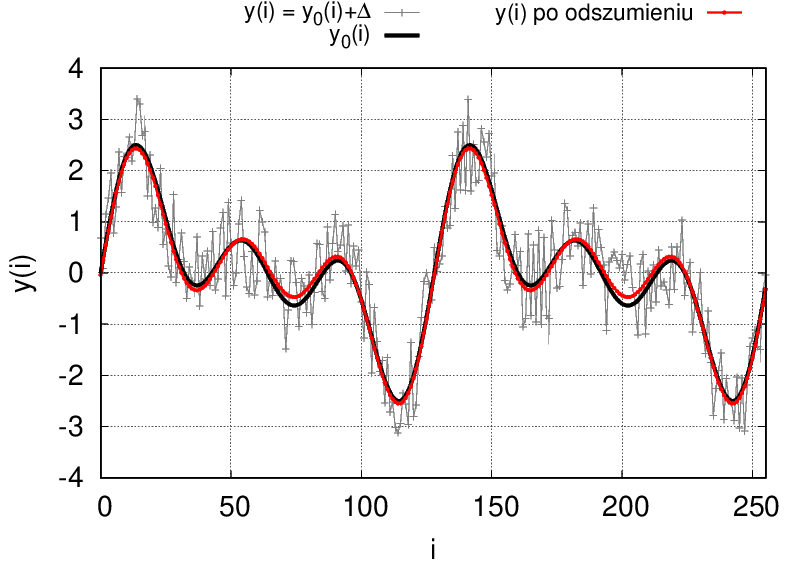
Aproksymację przeprowadzono kolejno dla Dodatkowo dla k=8 przedstawiono wykres części rzeczywistej i urojonej transformaty oraz moduł liczby zespolonej w zależności od numeru iteracji.



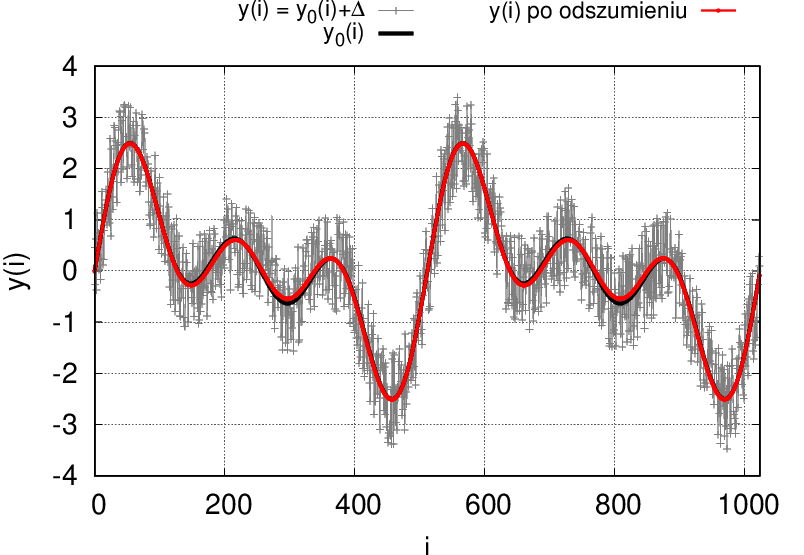
*Wykres (1). Część rzeczywista i urojona transformaty. (k=8)*



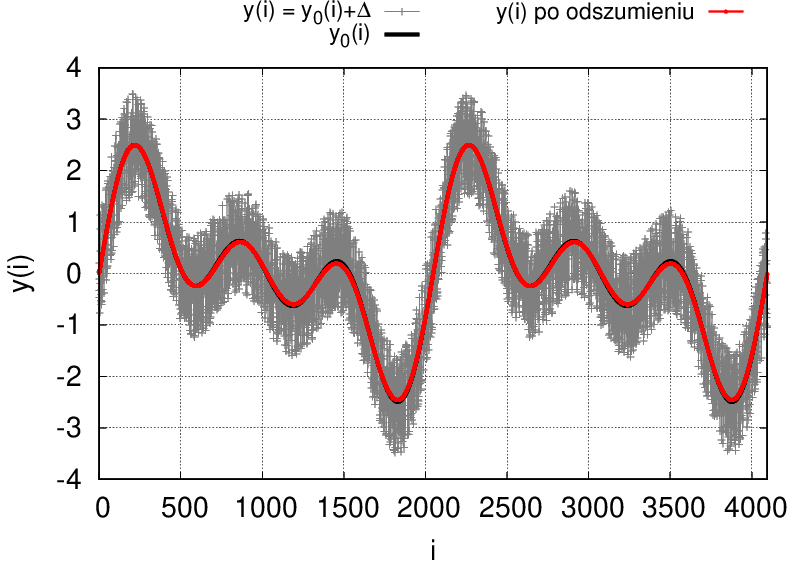
*Wykres(2). wartości modułów współczynników transformaty(k=8)*



*Wykres(3). Wykres sygnału zaburzonego, niezaburzonego oraz sygnału po odszumieniu dla k=8*



*Wykres(4). Wykres sygnału zaburzonego, niezaburzonego oraz sygnału po odszumieniu dla k=10*



*Wykres(5). Wykres sygnału zaburzonego, niezaburzonego oraz sygnału po odszumieniu dla k=10*

1. **Wnioski**

Dla każdej ilości próbek zaburzony sygnał został odszumiony. Z rysunków 3-5 można zauważyć że zwiększając liczbę próbek odszumienie bardziej się pokrywa z niezaburzonym sygnałem. Częstotliwość próbkowania ma zatem wpływ na dokładność procesu odszumiania.