[Dyskretna transformata Fouriera w czasie](https://pl.qwe.wiki/wiki/Discrete-time_Fourier_transform)

**Czym jest?**

Dyskretna transformata Fouriera (DTFT) to transformata Fouriera sygnału dyskretnego. Jego moc wyjściowa jest ciągła w częstotliwości i okresowo.

W DTFT sygnał w dziedzinie okresowej jest przekształcany w ciągły, okresowy sygnał w dziedzinie częstotliwości. W DFT twój sygnał wejściowy jest wyjściem twojego DTFT, który jest ciągłym, okresowym sygnałem w dziedzinie częstotliwości, a DFT daje Ci dyskretne próbki ciągłego DTFT.

Podstawowe właściwości transformaty DTFT sygnału dyskretnego:

• jest funkcją okresową kąta Ω o okresie 2π,

• jest funkcją ciągłego argumentu Ω,

• jest obliczana na podstawie nieskończonego ciągu próbek x(n),

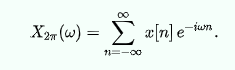
• dla sygnału rzeczywistego amplituda transformaty jest funkcją parzystą, a faza – funkcją nieparzystą.

**Sposób obliczania**

Na podstawie danych zebranych z analizy nagrywanego dźwięku obliczamy pulsację:

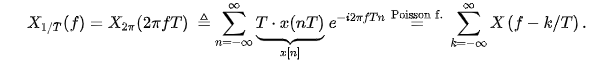
ω = 2\* π\*f

Następnie po policzeniu pojedynczych próbek widma obliczamy widmo całości nagrywanego sygnału.



x [ n ] - zbiór liczb rzeczywistych i zespolonych

ω – zmienna częstotliwości



X(f) -transformata Fouriera

K - liczba cykli / próbkę

1/T - jest próbką procentową

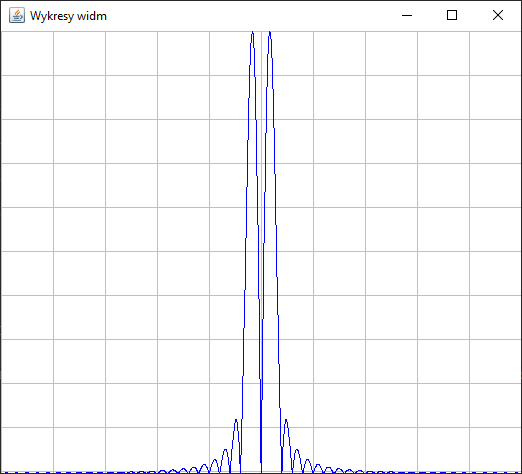
**W czym nam pomaga?**

Ludzka mowa i słuch używają tego rodzaju kodowania. Możemy wykorzystać DTFT do analizy systemów w dziedzinie czasu.

Ma zastosowanie do sekwencji wartości. DTFT jest używany do analizy próbek funkcji ciągłej.

Pozwala wyliczyć widmo całości nagrywanego sygnału.

Wykres widma naszej aplikacji:



**Szybka transformacja Fouriera**

[ang.](https://pl.wikipedia.org/wiki/J%C4%99zyk_angielski) Fast Fourier Transform – **FFT**

**1.Trochę ogólnie o algorytmie**

Szybka transformacja Fouriera to algorytm służący do liczenia dyskretnej transformaty Fouriera oraz transformaty do niej odwrotnej. Czasami używa się też nazwy **szybka transformata Fouriera**.

Niech *x*0, ...., *xN*-1 będą liczbami zespolonymi, wtedy dyskretna transformata Fouriera jest określona wzorem

 X_k =  \sum_{n=0}^{N-1} x_n e^{-{2\pi i \over N} nk }\qquadk = 0,\dots,N-1. 

Algorytmy obliczające **szybką transformację Fouriera** opierają się o metodę **dziel i zwyciężaj**, poprzez rekurencyjne dzielenie transformaty wielkości N=N1N2 na dwie oddzielne transformaty N1 oraz N2.

Najbardziej popularna wersja **FFT** to FFT o podstawie 2. Jest ona najbardziej efektywna, ale wymaga wektora próbek wejściowych (spróbkowanego sygnału) o długości N=2k , gdzie k jest liczbą naturalną. Złożoność obliczeniowa **FFT** wynosi O(Nlog2 N).

**2. Zastosowanie szybkiej transformaty Fouriera**

**Dyskretna transformata Fouriera (DTF)** jest szeroko stosowana:

* w aplikacjach przetwarzania sygnałów, jak na przykład w telekomunikacji.
* kompresji obrazów w formacie JPEG (DCT)
* kompresji dźwięku w formacie MP3 i pokrewnych (MDCT)
* rozwiązywaniu ważnych równań różniczkowych cząstkowych (DFT)
* filtrowania szumów (DFT)
* szybkiego mnożenia wielomianów (DFT)

**3. FFT w naszej aplikacji**

W naszej aplikacji algorytm **FFT** również znalazł swoje zastosowanie. Użyty algorytm wydaje się działać poprawnie. Na początku algorytm sprawdza, czy liczba wprowadzonych próbek jest potęgą liczby 2. Tak jak tego oczekiwaliśmy, w przypadku niepoprawnej liczbie próbek algorytm zwraca wyjątek i informuje nas o tym, że wprowadzone N nie jest potęga liczby 2.

Następnie zastosowany algorytm przelicza dane wejściowe jeśli są one poprawne (są potęgą liczby 2).

Funkcja po wykonaniu obliczeń zwraca wartości liczb zespolonych **transformaty Fouriera** oraz **odwrotnej transformaty Fouriera.**

**4. Jak to wykorzystamy w działaniu aplikacji?**

Ludzka mowa może być zarejestrowana za pomocą ciągu takich próbek. Dzięki **FFT** analiza zarejestrowanego głosu przebiega szybciej.

**Bibliografia:**

* <https://www.szkolnictwo.pl/szukaj,Szybka_transformacja_Fouriera>
* <http://wazniak.mimuw.edu.pl/index.php?title=MN10>