

ДИСКРЕТНЕ ПЕРЕТВОРЕННЯ ФУР'Є

Елементи теорії

Дискретное преобразование Фурье может быть получено непосредственно из интегрального преобразования дискретизаций аргументов

:

$$S(f) = \int_{-\infty}^{\infty} s(t) \exp(-j2\pi ft) dt, \quad S(f_n) = \Delta t \sum_{k=-\infty}^{\infty} s(t_k) \exp(-j2\pi f_n k \Delta t), \quad (8.1.1)$$

$$s(t) = \int_{-\infty}^{\infty} S(f) \exp(j2\pi ft) df, \quad s(t_k) = \Delta f \sum_{n=-\infty}^{\infty} S(f_n) \exp(j2\pi n \Delta f t_k). \quad (8.1.2)$$

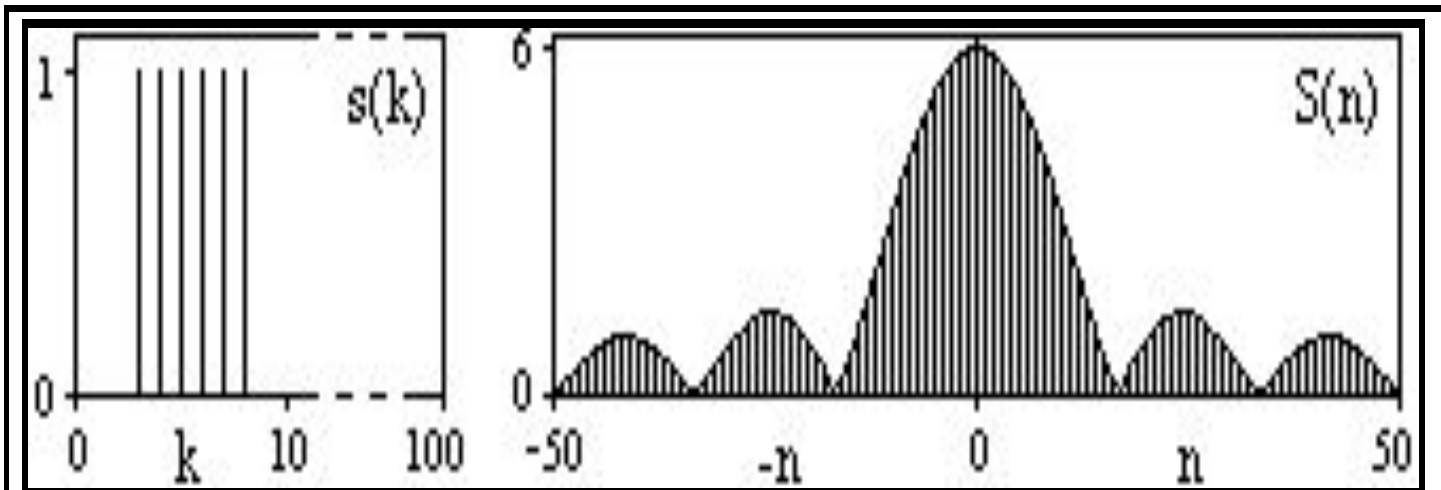
Для дискретных преобразований $s(k\Delta t) \Leftrightarrow S(n\Delta f)$, и функция, и ее спектр дискретны и периодичны, а числовые массивы их представления соответствуют заданию на главных периодах $T = N\Delta t$ (от 0 до T или от $-T/2$ до $T/2$), и $2f_N = N\Delta f$ (от $-f_N$ до f_N), где N – количество отсчетов, при этом:

$$\Delta f = 1/T = 1/(N\Delta t), \quad \Delta t = 1/2f_N = 1/(N\Delta f), \quad \Delta t \Delta f = 1/N, \quad N = 2Tf_N. \quad (8.1.3)$$

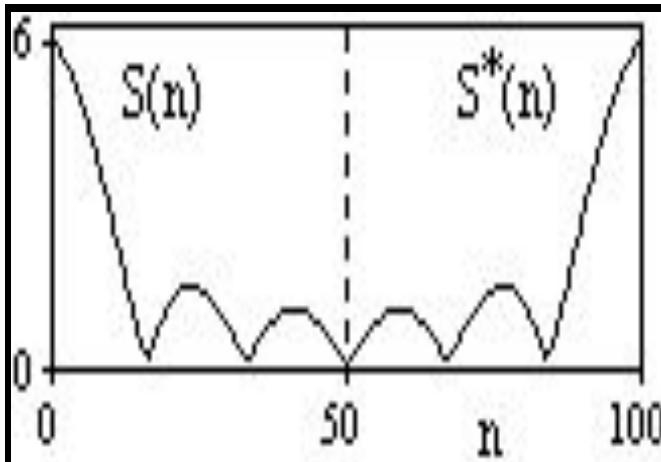
При дискретном представлении сигналов аргумент t_k обычно проставляется номерами отсчетов k (по умолчанию $\Delta t = 1$, $k = 0, 1, \dots, N-1$), а преобразования Фурье выполняются по аргументу n (номер шага по частоте) на главных периодах. При значениях N , кратных 2:

$$S(f_n) \equiv S_n = \sum_{k=0}^{N-1} s_k \exp(-j2\pi kn/N), \quad n = -N/2, \dots, 0, \dots, N/2. \quad (8.1.4)$$

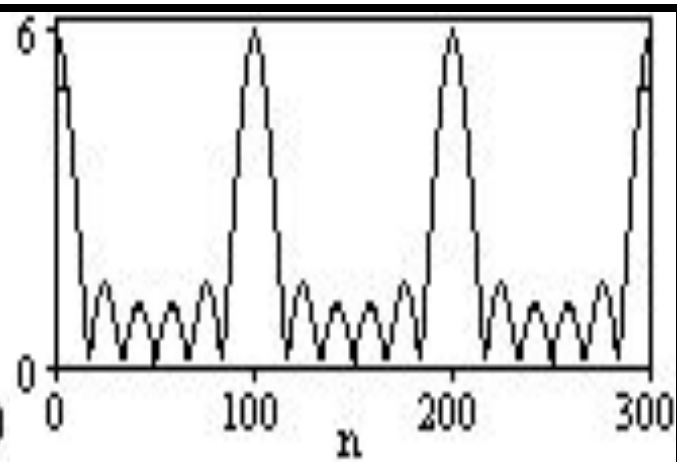
$$s(t_k) \equiv s_k = (1/N) \sum_{n=-N/2}^{N/2-1} S_n \exp(j2\pi kn/N), \quad k = 0, 1, \dots, N-1. \quad (8.1.5)$$



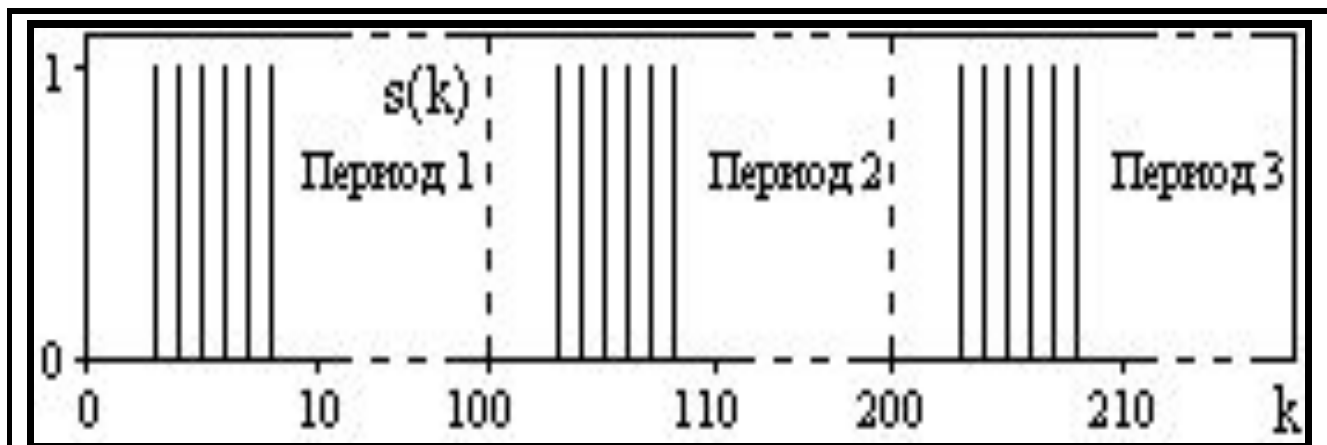
1.1. Дискретный сигнал и модуль его спектра



1.2. Модуль спектра



1.3. Модуль спектра



1.4. Обратное преобразование Фурье

Быстрое преобразование Фурье

Быстрое преобразование Фурье (БПФ, fast Fourier transform - FFT). Он базируется на том, что при вычислениях среди множителей (синусов и косинусов) есть много периодически повторяющихся значений (в силу периодичности функций). Алгоритм БПФ группирует слагаемые с одинаковыми множителями в пирамидаль-ный алгоритм, значительно сокращая число умножений за счет исключения повторных вычислений. В результате быстроедействие БПФ в зависимости от N может в сотни раз превосходить быстроедействие стандартного алго-ритма. При этом следует подчеркнуть, что алгоритм БПФ даже точнее стандартного, т.к. сокращая число операций он приводит к меньшим ошибкам округления

Дискретные прямое и обратное преобразования Фурье для одномерного массива x длины N определяются следующим образом:

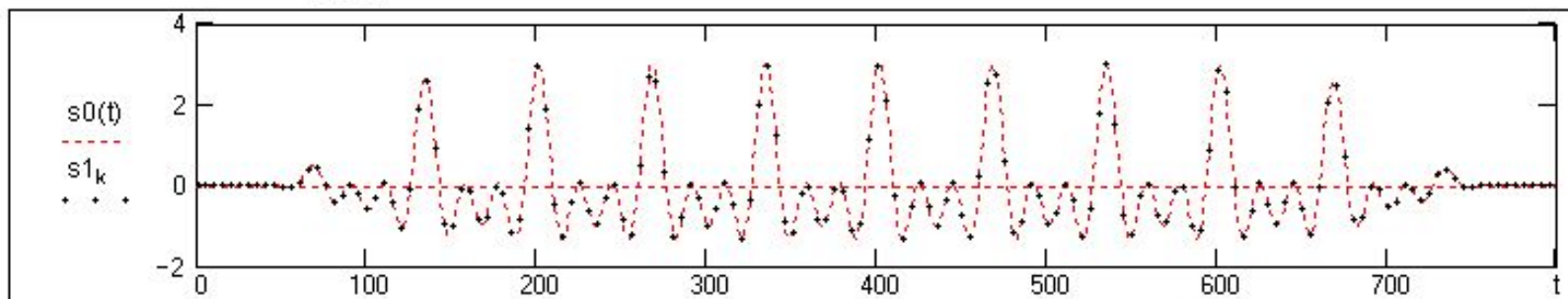
$$X(k) = \sum_{j=1}^N x(j) e^{\frac{2\pi}{N(j-1)(k-1)}};$$

$$x(k) = \frac{1}{N} \sum_{k=1}^N X(k) e^{\frac{-2\pi}{N(j-1)(k-1)}}.$$

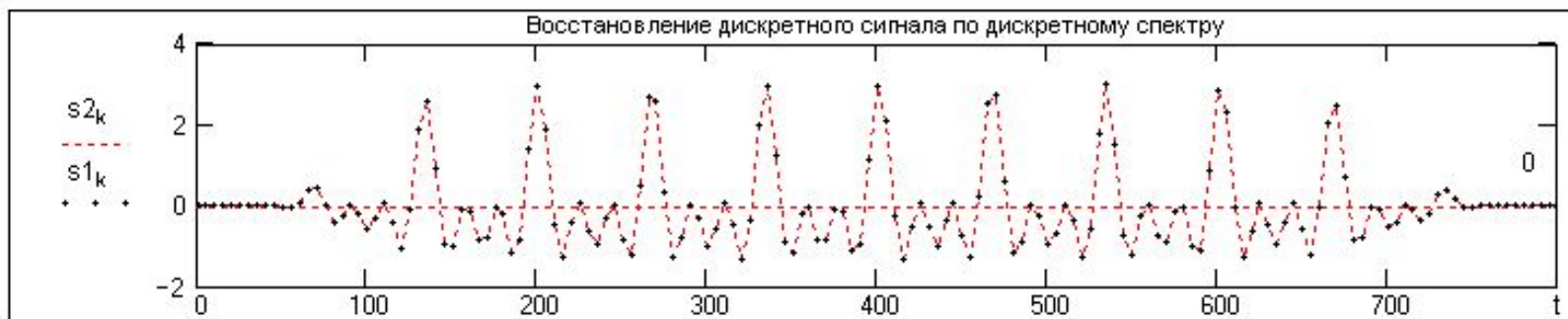
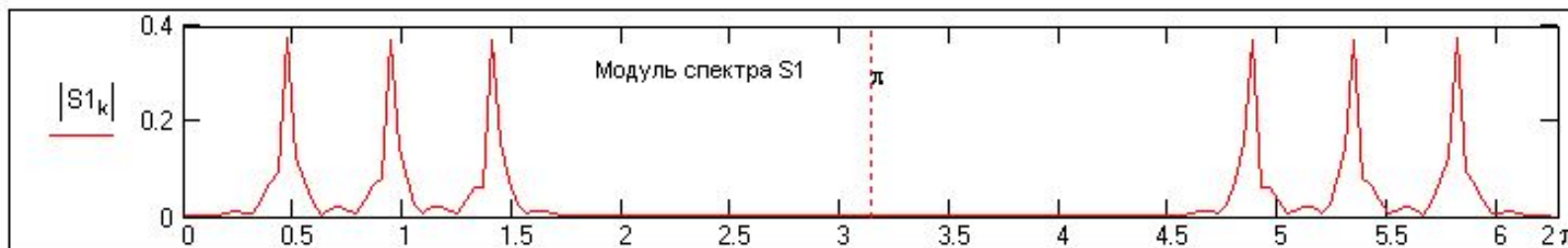
MATHCAD

Быстрое преобразование Фурье

$T := 800$ $t := 0..T$ $u1(t) := \exp[-1 \cdot 10^{-30} \cdot (t - 400)^{12}]$ $v1(t) := \sum_{k=1}^3 \cos(0.094 \cdot t \cdot k)$ $s0(t) := u1(t) \cdot v1(t)$ **<= модель**
 $\Delta t := 5$ $K := 2 \cdot \text{floor}\left(\frac{T}{\Delta t \cdot 2}\right)$ $K = 160$ $N := K + 1$ $N = 161$ $k := 0..K$ $s1_k := s0(k \cdot \Delta t)$ **<= Дискретизация**



$S1 := \text{CFFT}(s1)$ $s2 := \text{ICFFT}(S1)$ $\Delta \omega := \frac{2\pi}{N}$ **<= Прямое и обратное быстрое преобразование Фурье**



ДИСКРЕТНЕ ПЕРЕТВОРЕННЯ ФУР'Є

Реалізація в MATLAB

Одномерное дискретное прямое и обратное преобразования Фурье

Синтаксис:

$$Y = \text{fft}(X) \quad X = \text{ifft}(Y)$$

$$Y = \text{fft}(X, n) \quad X = \text{ifft}(Y, n)$$

Дискретные прямое и обратное преобразования Фурье для одномерного массива x длины N определяются следующим образом:

$$X(k) = \sum_{j=1}^N x(j) e^{\frac{2\pi}{N(j-1)(k-1)}};$$
$$x(k) = \frac{1}{N} \sum_{k=1}^N X(k) e^{\frac{-2\pi}{N(j-1)(k-1)}}.$$

Функция $Y = \text{fft}(X)$ вычисляет для массива данных X дискретное преобразование Фурье, используя FFT-алгоритм быстрого Фурье-преобразования. Если массив X двумерный, вычисляется дискретное преобразование каждого столбца.

Функция $Y = \text{fft}(X, n)$ вычисляет n -точечное дискретное преобразование Фурье. Если $\text{length}(X) < n$, то недостающие строки массива X заполняются нулями; если $\text{length}(X) > n$, то лишние строки удаляются.

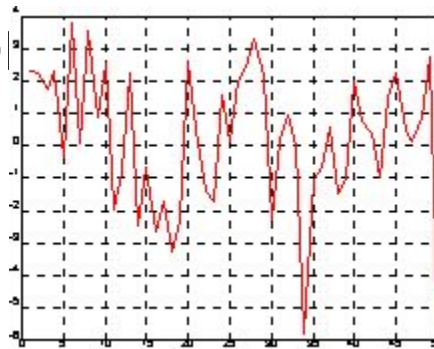
Функция $X = \text{ifft}(Y)$ вычисляет обратное преобразование Фурье для массива Y .

Функция $X = \text{ifft}(Y, n)$ вычисляет n -точечное обратное преобразование Фурье для массива Y .

Примеры:

Основное назначение преобразования Фурье - выделить частоты регулярных составляющих сигнала, зашумленного помехами. Рассмотрим данные, поступающие с частотой 1000 Гц.

Сформируем сигнал, содержащий регулярные составляющие с частотами 50 Гц и 120 Гц и случайную аддитивную помеху с нулевым средним.



```
t = 0:0.001:0.6;
```

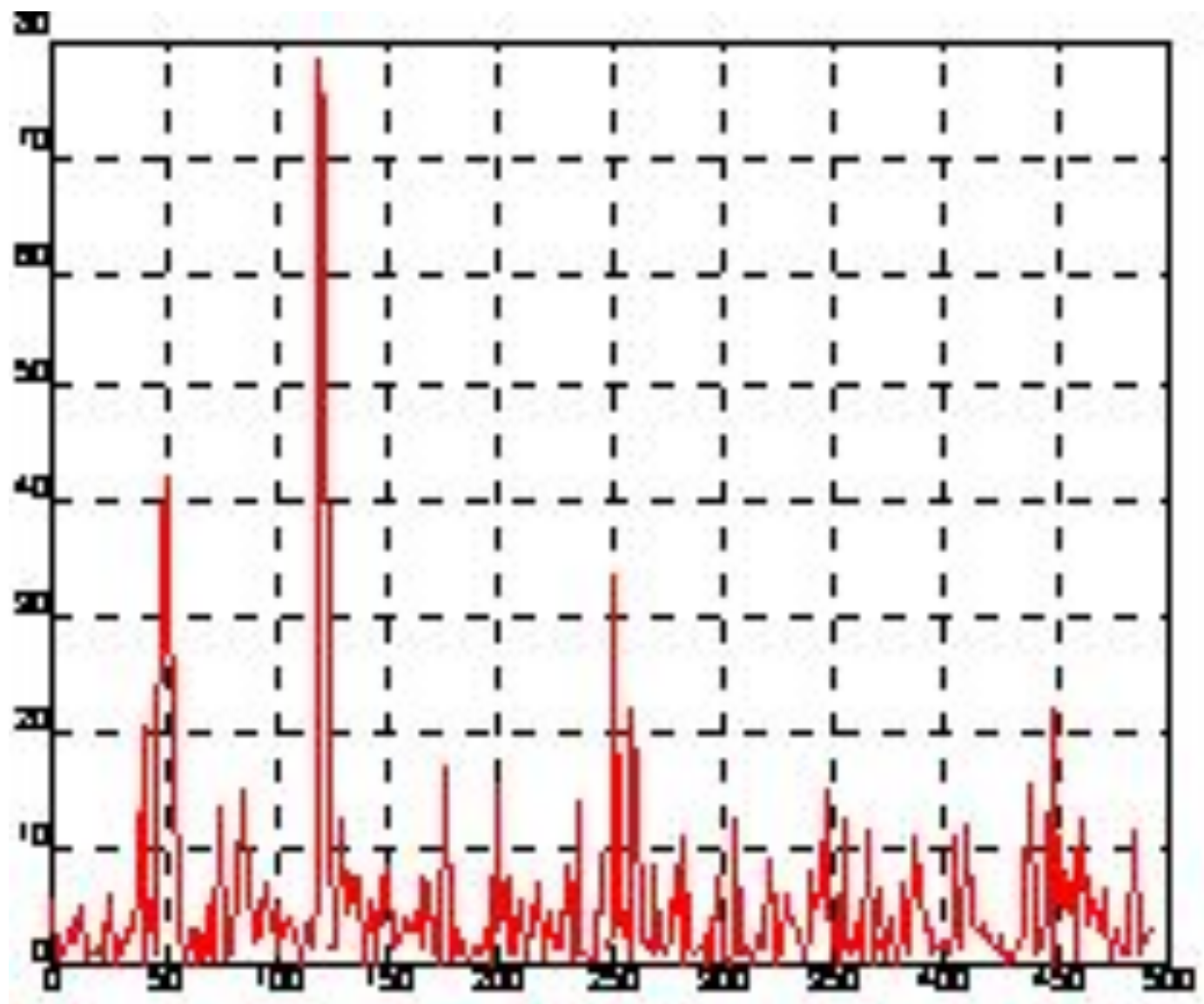
```
x = sin(2 * pi * 50 * t) + sin(2 * pi * 120 * t);
```

```
y = x + 2 * randn(size(t));
```

```
plot(y(1:50)), grid
```

На рис. а показан этот сигнал. Глядя на него, трудно сказать, каковы частоты его регулярных составляющих. Реализуя одномерное преобразование Фурье этого сигнала на основе 512 точек и построив график спектральной плотности (рис. б), можно выделить две частоты, на которых амплитуда спектра максимальна. Это частоты 120 и 50 Гц.

```
Y = fft(y, 512);  
Pyy = Y.*conj(Y)/512;  
f = 1000 * (0:255)/512;  
figure(2), plot(f, Pyy(1:256)), grid
```



Алгоритм:

Если длина последовательности входных данных является степенью числа 2, то применяется алгоритм быстрого преобразования Фурье с основанием 2, имеющий максимальную производительность. Этот алгоритм оптимизирован для работы с действительными данными; если данные комплексные, то реализуется комплексное преобразование Фурье. Эффективность первого на 40 % выше второго.

Если длина входной последовательности не является степенью числа 2, то применяется преобразование со смешанными основаниями, которые определяются как простые множители длины входной последовательности, которая при необходимости подвергается усечению

Продолжение алгоритма:

Время расчета существенно зависит от значения длины последовательности. Если значение длины точно разлагается на простые множители, то вычисления для такой последовательности выполняются достаточно быстро; если же не все множители оказываются простыми, и даже их будет меньше, то время вычисления существенно возрастает.

Если сравнивать эффективность вычислений, то преобразование Фурье с основанием 2 действительной последовательности из 4096 точек занимает 2.1 с, а комплексной - 3.7 с. Обычное преобразование Фурье для последовательности из 4096 точек занимает 7 с, а для последовательности из 4098 точек - 58 с

Сопутствующие функции: FFT2, IFFT2, FFTSHIFT,
Signal Processing Toolbox [1].

ССЫЛКИ:

1. Signal Processing Toolbox User's Guide. Natick: The MathWorks, Inc., 1993.