



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ

ИУ «Информатика и системы управления»

---

КАФЕДРА

ИУ1 «Системы автоматического управления»

---

## ОТЧЕТ

по лабораторной работе №4  
«Быстрое преобразование Фурье»

по дисциплине  
«Методы вычислений»

Выполнила: Шевченко А. Д.

Группа: ИУ1 – 32Б

Проверил: Бобков А.В.

Работа выполнена:

Отчет сдан: 14.12.2022

Оценка:

Москва 2022

## **Оглавление**

Дискретное преобразование Фурье .....	3
Быстрое преобразование Фурье (схема Кули - Тьюки) .....	3
Результат работы .....	9

## Дискретное преобразование Фурье

Дискретное преобразование Фурье — это одно из преобразований Фурье, широко применяемых в алгоритмах цифровой обработки сигналов (его модификации применяются в сжатии звука в MP3, сжатии изображений в JPEG и др.), а также в других областях, связанных с анализом частот в дискретном (к примеру, оцифрованном аналоговом) сигнале. Дискретное преобразование Фурье требует в качестве входа дискретную функцию. Такие функции часто создаются путём дискретизации (выборки значений из непрерывных функций). Дискретные преобразования Фурье помогают решать дифференциальные уравнения в частных производных и выполнять такие операции, как свёртки. Дискретные преобразования Фурье также активно используются в статистике, при анализе временных рядов.

Формулы преобразований:

1) Прямое преобразование:

$$Z_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) W_N^{nk}$$

2) Обратное преобразование:

$$x(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} Z_k W_N^{-nk}$$

Где

$Z_k = A_k + jB_k$  — коэффициент ряда Фурье,

$x(n)$  — дискретная функция,

$$W_N = e^{-\frac{2\pi j}{N}}.$$

## Быстрое преобразование Фурье (схема Кули - Тьюки)

Быстрое преобразование Фурье (БПФ, FFT) — алгоритм ускоренного вычисления дискретного преобразования Фурье, позволяющий получить результат за время, меньшее чем  $O(N^2)$  (требуемого для прямого, поформульного вычисления). Иногда под быстрым преобразованием Фурье

понимается один из алгоритмов, называемый алгоритмом прореживания по частоте — времени, имеющий сложность  $O(N \cdot \log(N))$ .

Свойства  $W_N^x$ :

$$W_N^0 = 1;$$

$$W_N^N = 1;$$

$$W_N^{kN} = 1, \quad k = 1, 2, \dots$$

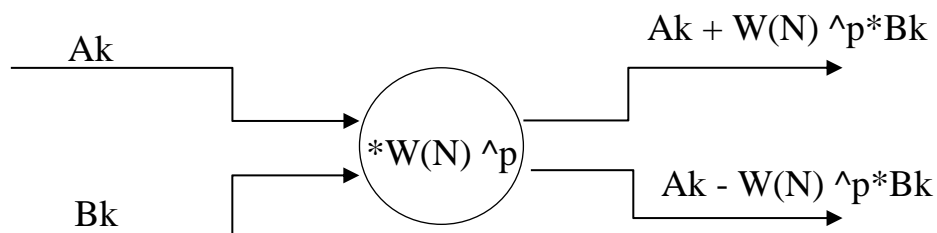
$$W_N^{\frac{N}{2}} = -1.$$

Схема Кули – Тьюки:

$$\begin{aligned} Z_k &= \sum_{n=0}^{N-1} x(n) W_N^{nk} = \sum_{n=0}^{\frac{N}{2}-1} x(2n) W_N^{2nk} + \sum_{n=0}^{\frac{N}{2}-1} x(2n+1) W_N^{(2n+1)k} = \\ &= \left[ W_N^{(2n+1)k} = W_N^{2nk} * W_N^k \right] = \\ &= \sum_{n=0}^{\frac{N}{2}-1} x(2n) W_N^{2nk} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} x(2n+1) W_N^{2nk}, \quad k = \overline{0, N-1} \end{aligned}$$

$$\Rightarrow Z_k = A_k + W_N^k * B_k;$$

$$\Rightarrow Z_{k+\frac{N}{2}} = A_k - W_N^k * B_k.$$



# ЭКОНОМИЯ ВРЕМЕНИ

$$X_N(k) = \sum_{r=0}^{\frac{N}{2}-1} x(2r) W_{N/2}^{rk} + W_N^k \sum_{r=0}^{\frac{N}{2}-1} x(2r+1) W_{N/2}^{rk}$$

$(N/2)^2$  УМНОЖЕНИЙ       $N/2$  УМНОЖЕНИЙ       $(N/2)^2$  УМНОЖЕНИЙ

- ДЛЯ 8-ТОЧЕЧНОГО БПФ  $\Rightarrow 4^2 + 4^2 + 4 = 36$  УМНОЖЕНИЙ  
ЭКОНОМИЯ  $64 - 36 = 28$  УМНОЖЕНИЙ
- ДЛЯ 1000-ТОЧЕЧНОГО БПФ  $\Rightarrow 500^2 + 500^2 + 500 = 500\,500$  УМНОЖЕНИЙ  
ЭКОНОМИЯ  $1\,000\,000 - 500\,500 = 499\,500$  УМНОЖЕНИЙ
- ЭКОНОМИЯ ВРЕМЕНИ; ПУСТЬ ДЛИТЕЛЬНОСТЬ ЦИКЛА 50 нс  
8-ТОЧЕЧНОЕ БПФ ЭКОНОМИТ 1.4 мкс  
1000-ТОЧЕЧНОЕ БПФ ЭКОНОМИТ 24.975 мс

## Реализация кода

```

Power.m  IDFT.m  adress.m  ReverseBits.m
1  function [b] = ReverseBits(a, n)
2      b = 0;
3      for i = 1:n
4          %c = bitand(a, 1);
5          %b = bitshift(b, 1);
6          %b = b + c;
7          b = b + b + bitand(a, 1);
8          a = bitshift(a, - 1);
9      end
10     end
  
```

```
Power.m x IDFT.m x adress.m x ReverseBits.m x ReverseBitSort.m x
1 function [X1] = ReverseBitSort(X, n)
2   N = length(X);
3   X1 = zeros(N, 1);
4   for i = 0:N - 1
5       b = ReverseBits(i,n);
6       X1(i + 1) = X(b + 1);
7   end
8   end
```

```
Power.m x IDFT.m x adress.m x ReverseBits.m x
1 function [a1, a2] = adress(p, k)
2   b = bitshift(1, k - 1);
3   m = b - 1;
4   R = bitand(p, m);
5   L = p - R;
6   %a1 = bitshift(L, 1) + R;
7   %a2 = bitshift(L, 1) + b + R;
8   a1 = L + L + R;
9   a2 = a1 + b;
10  end
```

```
Power.m x IDFT.m x adress.m x ReverseBits.m x
1 function [q] = Power(p, k, n, N)
2
3   dq = bitshift(1, n-k);
4   q = dq * p;
5   q = mod(q, bitshift(N, -1));
6
7   end
```

```

Power.m x IDFT.m x adress.m x ReverseBits.m x ReverseBitSort.m x main.m x DFT.m x
1 function [Z] = DFT(X)
2
3     N = length(X);
4     n = round(log(N)/log(2));
5     cpu = round(N/2);
6     W = exp(-2*pi*1i/N);
7     Z = ReverseBitSort(X, n);
8
9     for i = 1:n
10        for p = 0: (cpu - 1)
11            [a1, a2] = adress(p, i);
12            q = Power(p, i, n, N);
13            A = Z(a1 + 1);
14            B = Z(a2 + 1)*W^q;
15            Z(a1 + 1) = A + B;
16            Z(a2 + 1) = A - B;
17        end
18    end
19 end

```

```

Power.m x IDFT.m x adress.m x ReverseBits.m x
1 function [Z] = IDFT(X)
2
3     N = length(X);
4     n = round(log(N)/log(2));
5     cpu = round(N/2);
6     W = exp(2*pi*1i/N);
7     Z = ReverseBitSort(X, n);
8
9     for i = 1:n
10        for p = 0: (cpu - 1)
11            [a1, a2] = adress(p, i);
12            q = Power(p, i, n, N);
13            A = Z(a1 + 1);
14            B = Z(a2 + 1)*W^q;
15            Z(a1 + 1) = A + B;
16            Z(a2 + 1) = A - B;
17        end
18    end
19 end

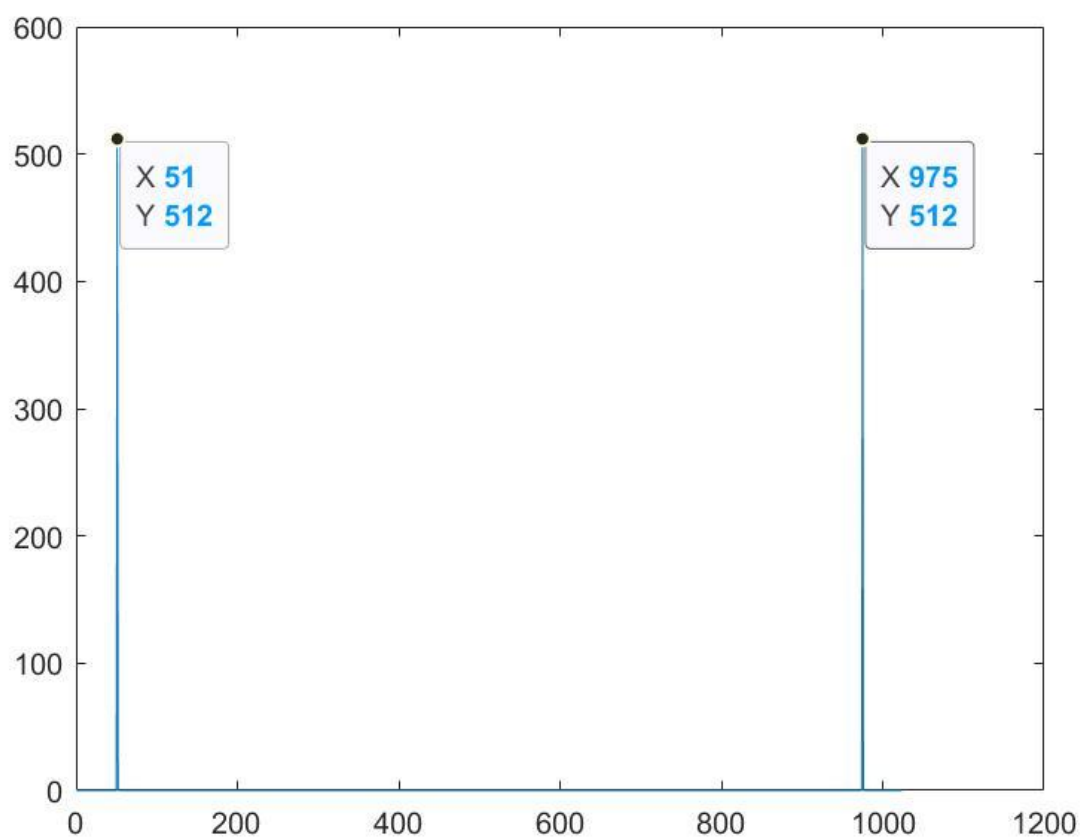
```

```
Power.m x IDFT.m x adress.m x ReverseBits.m x ReverseBitSort.m x main.m x
26     Z1=zeros(size(Z));
27     Z1(51)=Z(51);
28     Z1(975)=Z(975);
29     X1=IDFT(Z1);
30     X1=real(X1);
31
32     figure
33     subplot(2, 1, 1)
34     plot(t, X, "r")
35     title("Random noise")
36     subplot(2, 1, 2)
37     plot(t, X1)
38     title("Clear")
39
40     for i = 12:17
41         disp(i);
42         t = 0:1/2^i:1;
43         t = t(1:2^i);
44         [X] = sin(2*50*pi*t);
45         tic
46         [Z] = DFT(X);
47         toc
48     end
```

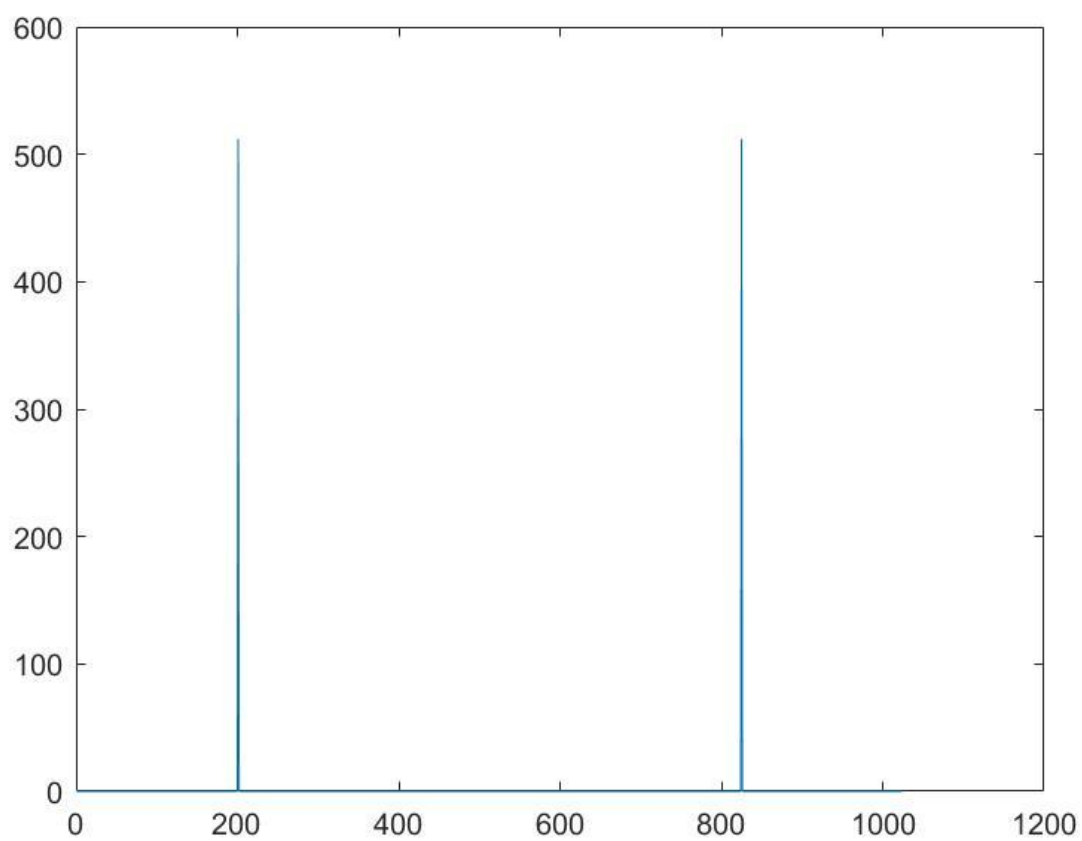
## Очистка от шума



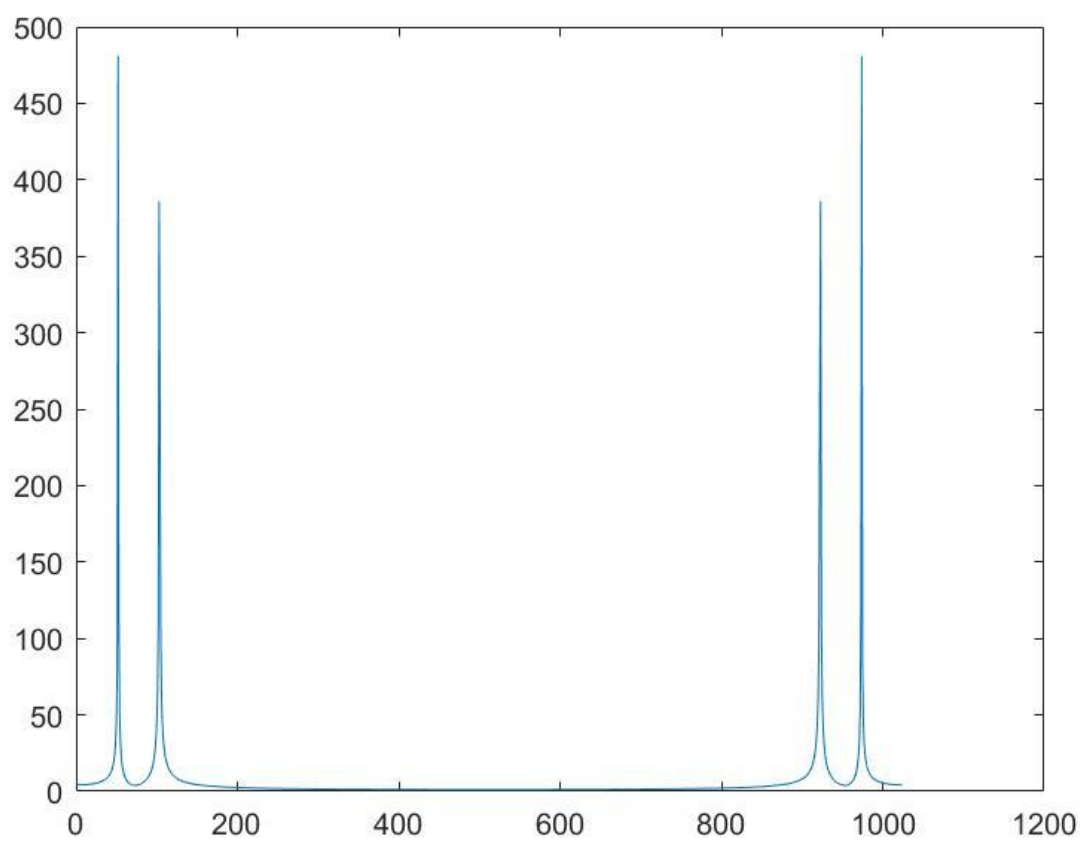
## Результат работы



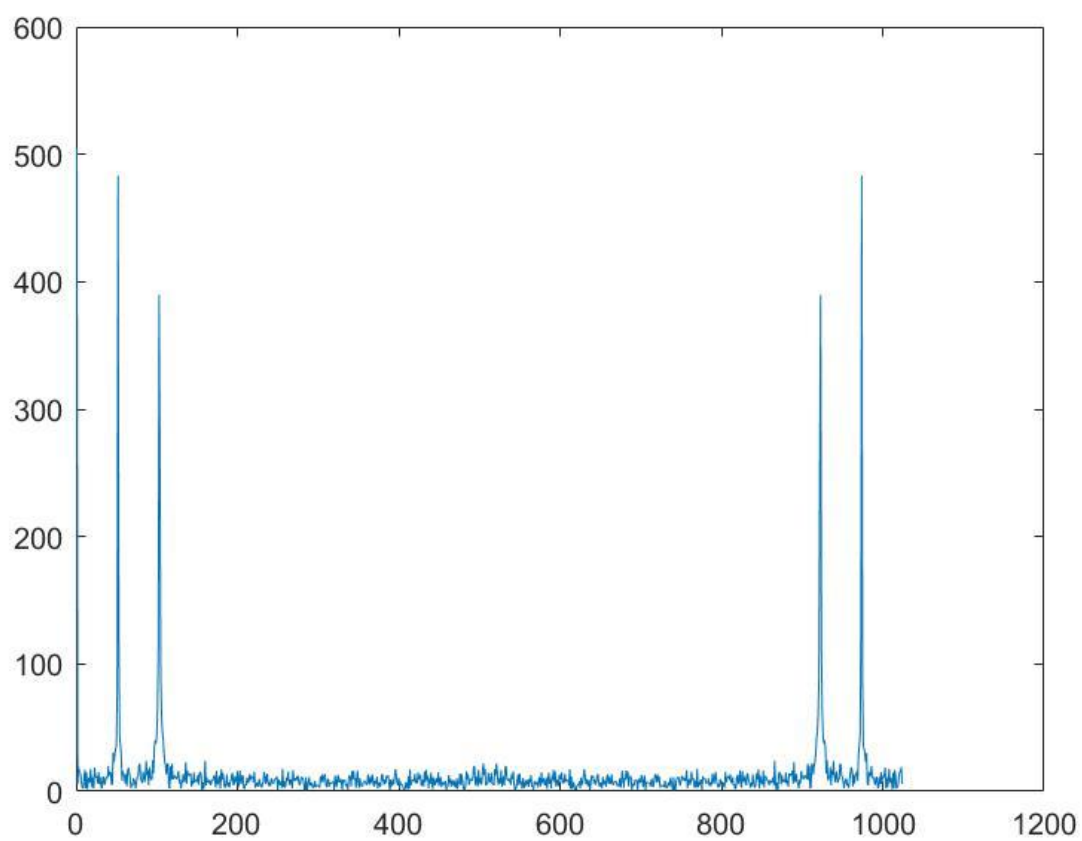
50 Гц



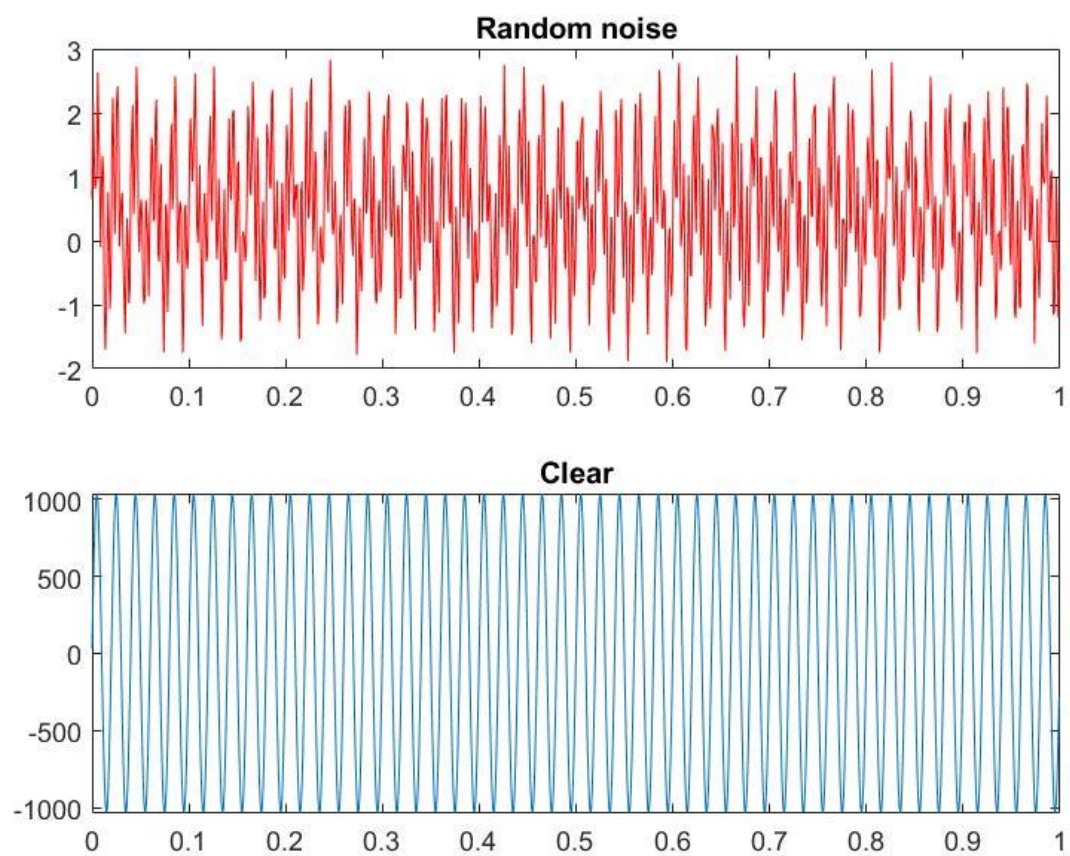
$200 \Gamma_{\Pi}$



50+200 Гц



$50 + 200 \text{ Гц} + \text{шум}$



Обратное преобразование

```
Command Window
12
Elapsed time is 0.139204 seconds.
13
Elapsed time is 0.285186 seconds.
14
Elapsed time is 0.592003 seconds.
15
Elapsed time is 1.209006 seconds.
16
Elapsed time is 2.508840 seconds.
17
Elapsed time is 5.381631 seconds.
fx >>
```