

Санкт-Петербургский государственный технический
университет
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Отчет по лабораторной работе №2
Ряд Фурье. Преобразование Фурье. Корреляция
Телекоммуникационные технологии

Кузьмин Н.А. 33501/1
Богач Н.В.

Санкт-Петербург 2018

Содержание

1	Цель	3
2	Постановка задачи	3
3	Ход работы	3
3.1	Расчет преобразования Фурье	3
3.2	Корреляция	3
4	Вывод	6

1 Цель

Получить представление о спектрах телекоммуникационных сигналов

2 Постановка задачи

– Для сигналов, построенных в лабораторной работе №1, выполните расчет преобразования Фурье. Перечислите свойства преобразования Фурье. – С помощью функции корреляции найдите позицию синхро- посылки [101] в сигнале [0001010111000010]. Получите пакет данных, если известно, что его длина составляет 8 бит без учета синхропосылки. Вычислите корреляцию прямым методом, воспользуйтесь алгоритмом быстрой корреляции, сравните время работы обоих алгоритмов. – Быстрая корреляция

3 Ход работы

3.1 Расчет преобразования Фурье

Выполним расчет преобразований Фурье для сигналов, построенных в лабораторной работе №1. Для этого воспользуемся функцией Matlab "fourier".

```
syms A f t Ph T;  
sSin = A*sin(2*pi*f*t + Ph);  
ftSin = fourier(sSin)  
  
sPulse = A*rectangularPulse(-T,T,t);  
ftPulse = fourier(sPulse)
```

Рис. 1: Код Matlab для расчета преобразования Фурье

```
ftSin =  
-A*pi*(dirac(w - 2*pi*f)*exp(Ph*1i) - dirac(w + 2*pi*f)*exp(-Ph*1i))*1i  
  
ftPulse =  
A*((sin(T*w) + cos(T*w)*1i)/w - (cos(T*w)*1i - sin(T*w))/w)
```

Рис. 2: Полученные в результате формулы

3.2 Корреляция

С помощью корреляции найдем позицию синхропосылки в сигнале при условии, что длина пакета данных составляет 8 бит без учета синхропосылки.

В результате мы получили 8 бит, которые в сигнале следуют сразу после

```

x = [0 0 0 1 0 1 0 1 1 1 0 0 0 0 1 0];
y = [1 0 1];
tic %прямой метод
N = length(x) + length(y) - 1;
R = zeros(1,N);
max = 0;
maxInd = 0;
for(i=1:1:N)
    for(j=1:1:length(y))
        if(i >= length(y) - j + 1 && i <= length(x) + 1 - j)
            R(i) = R(i) + x(i-length(y)+j)*y(j);
        end
    end
    if(R(i) > max)
        max = R(i);
        maxInd = i;
    end
end
res = zeros(1,8);
for(i=1:1:8)
    res(i) = x(maxInd+i);
end
toc %быстрый метод
%res
tic
N = 2^nextpow2(length(x));
x1 = fft(x, N);
y1 = fft(y, N);
R = ifft(x1.*conj(y1), N);
max=0;
maxInd = 0;
for(i=1:1:length(R))
    if(R(i)>max)
        max=R(i);
        maxInd = i;
    end
end
res = res(1,8);
for(i=1:1:8)
    res(i) = x(maxInd+length(y)-1+i);
end
toc
res

```

Рис. 3: Код Matlab для нахождения корреляции

```

Elapsed time is 0.010991 seconds.

res =

    0    1    1    1    0    0    0    0

Elapsed time is 0.007807 seconds.

res =

    0    1    1    1    0    0    0    0

```

Рис. 4: Результаты выполнения программы

синхросылки. Анализируя результаты по времени, можно сказать, что алгоритм быстрой корреляции выполняется чуть быстрее алгоритма прямой корреляции. Также в ходе других тестов результаты были прямо противоположные. Это происходит из-за того, что мы работаем с небольшими

объемами данных. При увеличении размеров сигнала или синхропосылки алгоритм быстрой корреляции выполнится быстрее.

4 Вывод

В ходе выполнения работы мы научились выполнять преобразование Фурье для простейших сигналов в пакете Matlab, а также реализовали алгоритмы прямой и быстрой корреляции. В результате выполнения алгоритмов был получен требуемый результат - следующие 8 бит в сигнале после синхропосылки. Анализируя время выполнения данных алгоритмов можно сказать, что размер данных слишком мал для получения ощутимой разницы в скорости этих алгоритмов.