

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(национальный исследовательский университет)
Кафедра 410

А.В. БРУХАНСКИЙ

Исследование свойств дискретного преобразования Фурье

Учебное пособие к лабораторной работе

Москва
МАИ
2018

Лабораторная работа № 3 «Дискретное преобразование Фурье»

В лабораторной работе анализируются свойства дискретного преобразования Фурье и спектры типовых сигналов. Вычисляется результат сжатия ЛЧМ и дискретно кодированных сигналов во временной и частотной областях.

Для выполнения лабораторной работы необходимо в среде MATLAB ввести тексты следующих функций и скриптов, а затем модифицировать их по заданию преподавателя.

Анализ влияния нулевых отсчетов, добавленных к сигналу, на его спектр

```
% Спектр прямоугольного импульса
x1 = [ones(8,1); zeros(8,1)]; % 16 отсчетов
y1 = fft(x1); % БПФ сигнала x1
x2 = [x1; zeros(16,1)]; % 32 отсчета
y2 = fft(x2); % БПФ сигнала x2
figure(1)
subplot(2, 2, 1), stem(0:15, x1, 'Marker', '.')
title('Сигнал'), xlim([0 31])
subplot(2, 2, 2), stem((0:15)/16, abs(y1), 'Marker', '.')
title('Спектр')
subplot(2, 2, 3), stem(0:31, x2, 'Marker', '.')
title('Сигнал'), xlim([0 31])
subplot(2, 2, 4)
stem((0:31)/32, abs(y2), 'Marker', '.')
title('Спектр')
figure(2)
x = [ones(8,1); zeros(56,1)]; % 64 отсчета
y = fft(x);
stem((0:63)/64, abs(y), 'Marker', '.');
title('Спектр сигнала 8 единиц, 56 нулей')
```

Расчет АЧХ каналов ДПФ

```
clear
N = 8; % количество каналов ДПФ
w = 0:0.1:N;
col='rgbkrgbk';
figure(1) % АЧХ всех 8 каналов
for k=1:8
plot(w, abs(diric((w-k)/N*2*pi, N)), col(k))
hold on
end
figure(2) % АЧХ 4-го и 5-го каналов
plot(w, abs(diric((w-4)/N*2*pi, N)), 'b')
hold on
plot(w, abs(diric((w-5)/N*2*pi, N)), 'r')
```

Анализ эффекта растекания спектра

```
% Анализ эффекта растекания спектра
clear
n = 32; % Кол-во отсчетов дискретных сигналов
td = 0:n-1; % время для дискретных сигналов
ta = 0:0.1:n; % время для аналоговых сигналов
T1 = 4; % период первого сигнала - 4 отсчета
T2 = 6; % период второго сигнала - 6 отсчетов
x1d = cos(2*pi*td/T1); % 1-й дискретный сигнал
x1a = cos(2*pi*ta/T1); % 1-й аналоговый сигнал
y1 = fft(x1d);
x2d = cos(2*pi*td/T2); % 2-й дискретный сигнал
x2a = cos(2*pi*ta/T2); % 2-й аналоговый сигнал
y2 = fft(x2d);
subplot(2, 2, 1), stem(td, x1d, 'Marker', '.')
hold on, plot(ta, x1a, '--'), hold off
title('Сигнал'), xlim([0 16])
subplot(2, 2, 2), stem(td, abs(y1), 'Marker', '.')
title('Спектр'), xlim([0 32])
subplot(2, 2, 3), stem(td, x2d, 'Marker', '.')
hold on, plot(ta, x2a, '--'), hold off
title('Сигнал'), xlim([0 16])
subplot(2, 2, 4), stem(td, abs(y2), 'Marker', '.')
title('Спектр'), xlim([0 32])
```

Согласованная фильтрация ЛЧМ сигнала

```
% Согласованная фильтрация ЛЧМ сигнала
n = 1024;% количество отсчетов сигнала
f0 = 100; % начальная частота сигнала
t = (0:n-1)/(8*f0); % время
d = 0.5/t(n)/2; % относительная добавка
    % к конечной частоте сигнала
x=sin(2*pi*f0*(1+d*t).*t); % ЛЧМ сигнал
figure(1)
plot(t,x) % график ЛЧМ сигнала
% figure(2) % Для старых версий matlab
% specgram(x,128,f0*8,[],100)
figure(3) % Для новых версий matlab
spectrogram(x,hann(128),100,128,f0*8);
view(90,-90)|
colorbar(gca,'off')
%Согласованная фильтрация во временной области
b = flipplr(x); % Коэффициенты фильтра
xx = [x,zeros(1,n)];% дополнение нулями
y = filter(b,1,xx); % фильтрация во времени
figure(4)
plot([t t+t(n)],y); grid, axis tight;
%Согласованная фильтрация в частотной области
fxx=fft(xx);
fbb=fft([b,zeros(1,n)]); %передаточная функция
fxb = fxx.*fbb; % фильтрация по частоте
yf = ifft(fxb); % выходной сигнал
figure(5)
plot([t t+t(n)],yf);
grid, axis tight
z = abs(y - yf); % разница выходных сигналов
figure(6)
plot(z)
```

Генерация M-последовательностей

```
% Генерация M-последовательностей
clear
% M-последовательность 1023 отсчета
primpoly(10, 'min')
g1 = zeros(1, 1023);
reg = -1*ones(1, 10);
for i=1:1023
    g1(i)          = reg(10);
    saveBit       = reg(3)*reg(10);
    reg(2:10)     = reg(1:9);
    reg(1)        = saveBit;
end
stem(g1, 'marker', '.')
% M-последовательность 31 отсчет
primpoly(5, 'all')
g2 = zeros(1, 31);
reg = -1*ones(1, 5);
for i=1:31
    g2(i)          = reg(5);
    saveBit       = reg(2)*reg(5);
    reg(2:5)      = reg(1:4);
    reg(1)        = saveBit;
end
figure(2)
stairs(g2, 'linewidth', 2), ylim([-1.1, 1.1]), grid
```

Согласованная фильтрация M-последовательности

```
% Согласованная фильтрация M-последовательности
```

```
clear
```

```
N = 10; % порядок примитивного полинома
```

```
n = 2^N - 1; % длина M-последовательности
```

```
p = primpoly(N, 'min', 'nodisplay');
```

```
d = dec2bin(p); % коэффициенты полинома
```

```
reg = -1*ones(1, N);
```

```
for i=1:n
```

```
    x(i) = reg(N);
```

```
    saveBit = 1;
```

```
    for k = 1:N
```

```
        if d(k)=='1'
```

```
            saveBit = saveBit * reg(N - k + 1);
```

```
        end
```

```
    end
```

```
    reg(2:N) = reg(1:N-1);
```

```
    reg(1) = saveBit;
```

```
end
```

```
b = fliplr(x); % коэффициенты фильтра
```

```
xx = [x,zeros(1,n)];
```

```
y = filter(b,1,xx); % фильтрация во времени
```

```
maxy = max(abs(y(1:n-1)));
```

```
display(['Коэффициенты полинома = ', d, ...  
        '; УВЛ = ', num2str(maxy)])
```

```
figure(4)
```

```
stem(y,'marker','.'); grid, axis tight
```

```
fx = fft(xx);
```

```
fb = fft([b,zeros(1,n)]);
```

```
fx = fx.*fb; % фильтрация по частоте
```

```
yf = ifft(fx); % выходной сигнал
```

```
figure(5)
```

```
stem(yf,'m','marker','.'); grid, axis tight
```

```
z = abs(y - yf); % разница выходных сигналов
```

```
figure(6)
```

```
stem(z,'marker','.'); grid, axis tight
```