

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1

дисциплина: Сетевые технологии

Студент: Бакулин Никита 1032201747

Группа: НПИбд-01-20

МОСКВА

2022 г.

Постановка задачи

1. Построить график функции $y = \sin x + 1/3 \sin 3x + 1/5 \sin 5x$ на интервале $[-10; 10]$, используя Octave и функцию plot. График экспортировать в файлы формата .eps, .png. Добавить график функции $y = \cos x + 1/3 \cos 3x + 1/5 \cos 5x$ на интервале $[-10; 10]$. График экспортировать в файлы формата .eps, .png.
2. Разработать код m-файла, результатом выполнения которого являются графики меандра, реализованные с различным количеством гармоник.
3. Определить спектр двух отдельных сигналов и их суммы. Выполнить задание с другой частотой дискретизации. Пояснить, что будет, если взять частоту дискретизации меньше 80 Гц?
4. Продемонстрировать принципы модуляции сигнала на примере аналоговой амплитудной модуляции
5. По заданным битовым последовательностям требуется получить кодированные сигналы для нескольких кодов, проверить свойства самосинхронизуемости кодов, получить спектры.

Выполнение работы

1.
 - 1.1. Запустите в вашей ОС Octave с оконным интерфейсом
 - 1.2. Перейдите в окно редактора. Воспользовавшись меню или комбинацией клавиш `ctrl + n` создайте новый сценарий. Сохраните его в ваш рабочий каталог с именем, например, `plot_sin.m`.
 - 1.3. В окне редактора повторите следующий листинг по построению графика функции $y = \sin x + 1/3 \sin 3x + 1/5 \sin 5x$ на интервале $[-10; 10]$:

```
1 % Формирование массива x:
2 x=-10:0.1:10;
3 % Формирование массива y.
4 y1=sin(x)+1/3*sin(3*x)+1/5*sin(5*x);
5 % Построение графика функции:
6 plot(x,y1, "-ok; y1=sin(x)+(1/3)*sin(3*x)+(1/5)*sin(5*x);", "markersize", 4)
7 % Отображение сетки на графике
8 grid on;
9 % Подпись оси X:
10 xlabel('x');
11 % Подпись оси Y:
12 ylabel('y');
13 % Название графика:
14 title('y1=sin x+ (1/3)sin(3x)+(1/5)sin(5x)');
15 % Экспорт рисунка в файл .eps:
16 print ("plot-sin.eps", "-mono", "-fArial:16", "-deps")
17 % Экспорт рисунка в файл .png:
18 print("plot-sin.png");
```

Рисунок 1

- 1.4. Запустите сценарий на выполнение (воспользуйтесь соответствующим меню окна редактора или клавишей F5). В качестве результата выполнения кода должно открыться окно с построенным графиком (рис. 1.1) и в вашем рабочем каталоге должны появиться файлы с графиками в форматах .eps, .png.

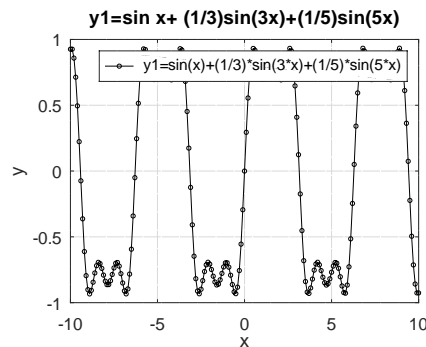


Рисунок 2

- 1.5. Сохраните сценарий под другим названием и измените его так, чтобы на одном графике располагались отличающиеся по типу линий графики функций $y1 = \sin x + 1/3 \sin 3x + 1/5 \sin 5x$, $y2 = \cos x + 1/3 \cos 3x + 1/5 \cos 5x$

```
% Формирование массива x:
x = 10*0:1:10;
% Формирование массива y:
y1 = sin(x) + 1/3*sin(3*x) + 1/5*sin(5*x);
% Формирование массива y2:
y2 = cos(x) + 1/3*cos(3*x) + 1/5*cos(5*x);
% Построение графика функции:
plot(x, y1, 'o', y2, 'b');
% Отображение сетки на графике
grid on
% Подпись оси X:
xlabel('x');
% Подпись оси Y:
ylabel('y');
% Название графика:
title('y1=sin(x)+(1/3)sin(3x)+(1/5)sin(5x) and y2=cos(x)+(1/3)cos(3x)+(1/5)cos(5x)');
% Экспорт рисунка в файл .eps:
print('plot-sincos.eps', '-memo', '-FArial:10', '-deps');
% Экспорт рисунка в файл .png:
print('plot-sincos.png');
```

Рисунок 3

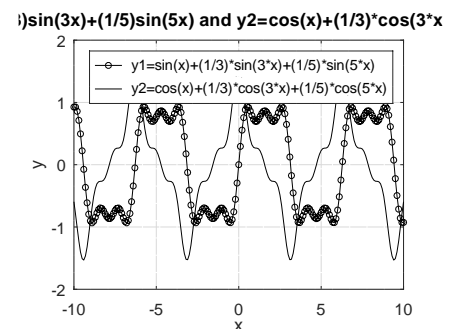


Рисунок 4

2.

- 2.1. Создайте новый сценарий и сохраните его в ваш рабочий каталог с именем, например, meandr.m.
- 2.2. В коде созданного сценария задайте начальные значения:
- 2.3. Далее задаём массив значений гармоник массив элементов ряда:
- 2.4. Далее для построения в одном окне отдельных графиков меандра с различным количеством гармоник реализуем суммирование ряда с накоплением и воспользуемся функциями subplot и plot для построения графиков
- 2.5. Экпортируйте полученный график в файл в формате .png

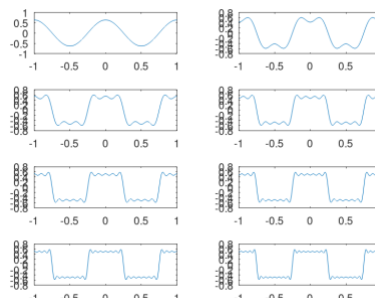


Рисунок 5

2.6. Скорректируйте код для реализации меандра через синусы. Получите соответствующие графики.

```
1 % количество отсчетов (гармоник):
2 N 8;
3 % частота дискретизации:
4 t 1/0.01;
5 % значения амплитуды:
6 A 1;
7 % период:
8 T 1;
9 % амплитуда гармоник
10 nh 1 N 2 1;
11 % массив коэффициентов для ряда, заданного через cos:
12 Am 2*pi/nh;
13 % массив гармоник:
14 harmonics sin(2*pi*nh*t/T);
15 % массив элементов ряда:
16 s1 harmonics repmat(Am,1,length(t));
17 % суммирование ряда:
18 s2 cumsum(s1);
19 % Построение графиков:
20 k 1 N;
21 subplot(4,2,k);
22 plot(t,s2(k,:));
23 axis;
24 % Экспорт рисунка в файл .png:
25 print('meandr_sin.png');
```

Рисунок 6

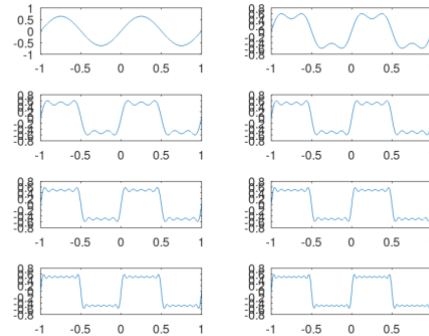


Рисунок 7

3.

3.1. В вашем рабочем каталоге создайте каталог spectre1 и в нём новый сценарий с именем, spectre.m

3.2. В коде созданного сценария задайте начальные значения:

3.3. Далее в коде задайте два синусоидальных сигнала разной частоты:

3.4. Постройте графики сигналов

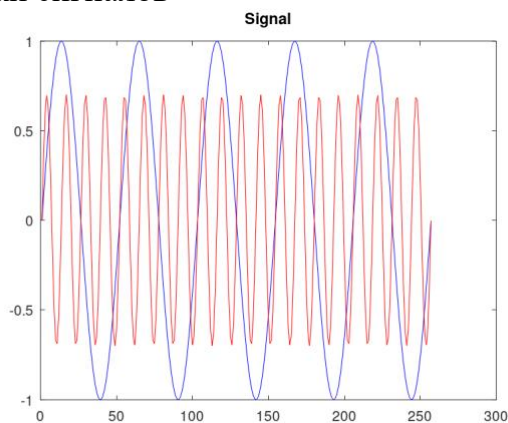


Рисунок 8

3.5. С помощью быстрого преобразования Фурье найдите спектры сигналов, добавив в файл spectre.m следующий код:

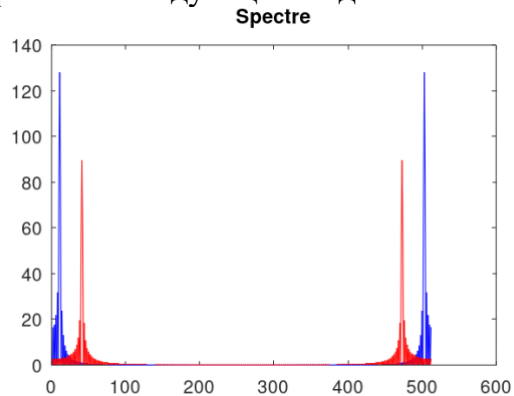


Рисунок 9

3.6. Учитывая реализацию преобразования Фурье, скорректируйте график спектра: отбросьте дублирующие отрицательные частоты, а также примите в расчёт то, что на каждом шаге вычисления быстрого преобразования Фурье происходит суммирование амплитуд сигналов. Для этого добавьте в файл `spectre.m` следующий код:

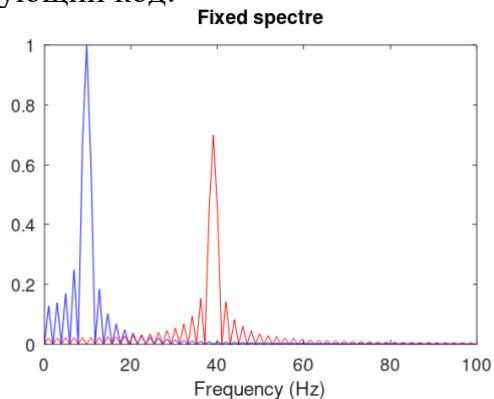


Рисунок 10

3.7. Найдите спектр суммы рассмотренных сигналов создав каталог `spectr_sum` и файл в нём `spectre_sum.m` со следующим кодом:

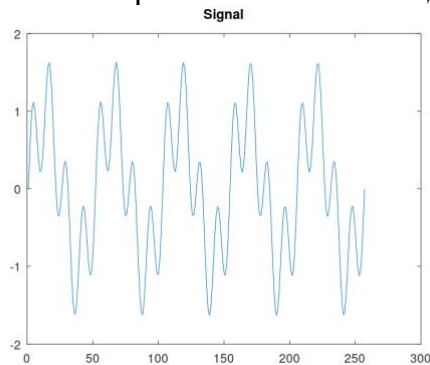


Рисунок 11

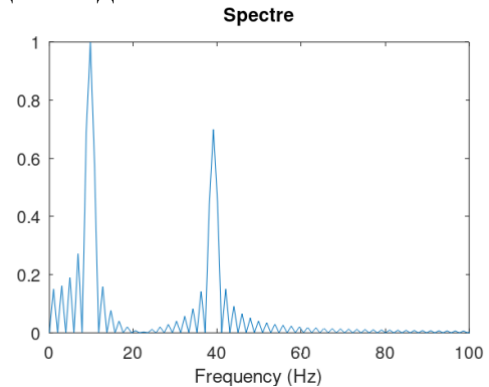


Рисунок 12

4.

4.1. В вашем рабочем каталоге создайте каталог `modulation` и в нём новый сценарий с именем `am.m`

4.2. Добавьте в файле `am.m` код. В результате получаем, что спектр произведения представляет собой свёртку спектров

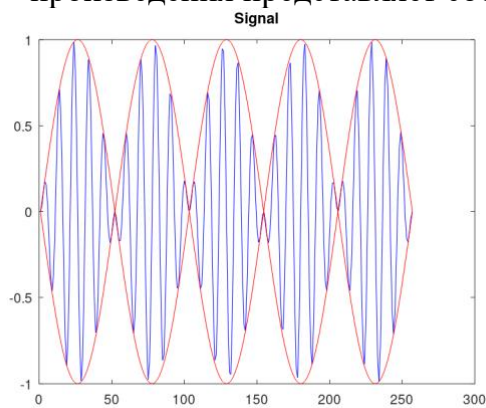


Рисунок 13

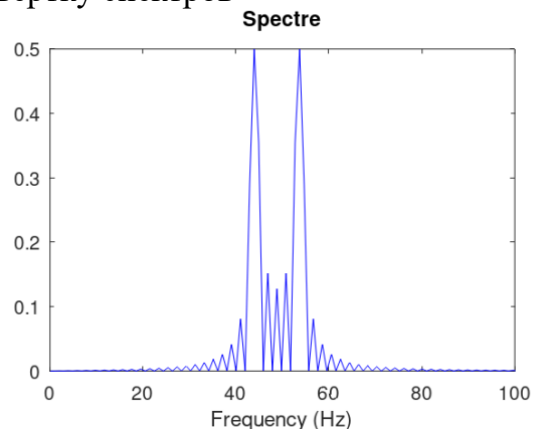


Рисунок 14

5.

- 5.1. В вашем рабочем каталоге создайте каталог coding и в нём файлы main.m, mptowave.m, unipolar.m, ami.m, bipolarnrz.m, bipolarrrz.m, manchester.m, diffmanc.m, calcspectre.m.
- 5.2. В окне интерпретатора команд проверьте, установлен ли у вас пакет расширений signal
- 5.3. В файле main.m подключите пакет signal и задайте входные кодовые последовательности:
- 5.4. В файле mptowave.m пропишите функцию, которая по входному битовому потоку строит график сигнала
- 5.5. В файлах unipolar.m, ami.m, bipolarnrz.m, bipolarrrz.m, manchester.m, diffmanc.m пропишите соответствующие функции преобразования кодовой последовательности data с вызовом функции mptowave для построения соответствующего графика.
- 5.6. В файле calcspectre.m пропишите функцию построения спектра сигнала
- 5.7. Запустите главный скрипт main.m. В каталоге signal должны быть получены файлы с графиками кодированного сигнала, в каталоге sync — файлы с графиками, иллюстрирующими свойства самосинхронизации, в каталоге spectre — файлы с графиками спектров сигналов

