

Отчёт по лабораторной работе №1

Сетевые технологии

Ищенко Ирина Олеговна НПИбд-02-22

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	12

Список иллюстраций

2.1	Редактирование plot_sin.m	6
2.2	График функции	7
2.3	Добавление линии на график	7
2.4	Графики меандра с разным количеством гармоник	8
2.5	Графики меандра с разным количеством гармоник через синусы	8
2.6	Два синусоидальных сигнала разной частоты	9
2.7	Спектр суммарного сигнала	9
2.8	Спектр сигнала при амплитудной модуляции	10
2.9	Создание и заполнение файлов в каталоге coding	10
2.10	Файлы с графиками кодированного сигнала	11
2.11	Файлы с графиками, иллюстрирующими свойства самосинхронизации	11
2.12	Файлы с графиками спектров сигналов	11

Список таблиц

1 Цель работы

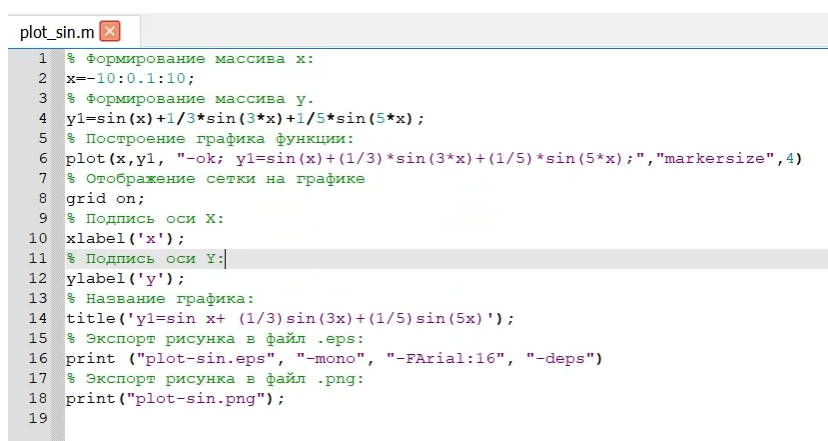
Изучение методов кодирования и модуляции сигналов с помощью высокоуровневого языка программирования Octave. Определение спектра и параметров сигнала. Демонстрация принципов модуляции сигнала на примере аналоговой амплитудной модуляции. Исследование свойства самосинхронизации сигнала.

2 Выполнение лабораторной работы

Построим график функции

$$y = \sin(x) + \frac{1}{3}\sin(3x) + \frac{1}{5}\sin(5x)$$

на интервале $[-10; 10]$, используя Octave и функцию `plot`. Создаем файл `plot_sin.m` (рис. 2.1).



```
plot_sin.m
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");
19
```

Рис. 2.1: Редактирование `plot_sin.m`

Запускаем файл и получаем график (рис. 2.2). В рабочем каталоге появляются файлы с графиками в форматах `.eps`, `.png`.

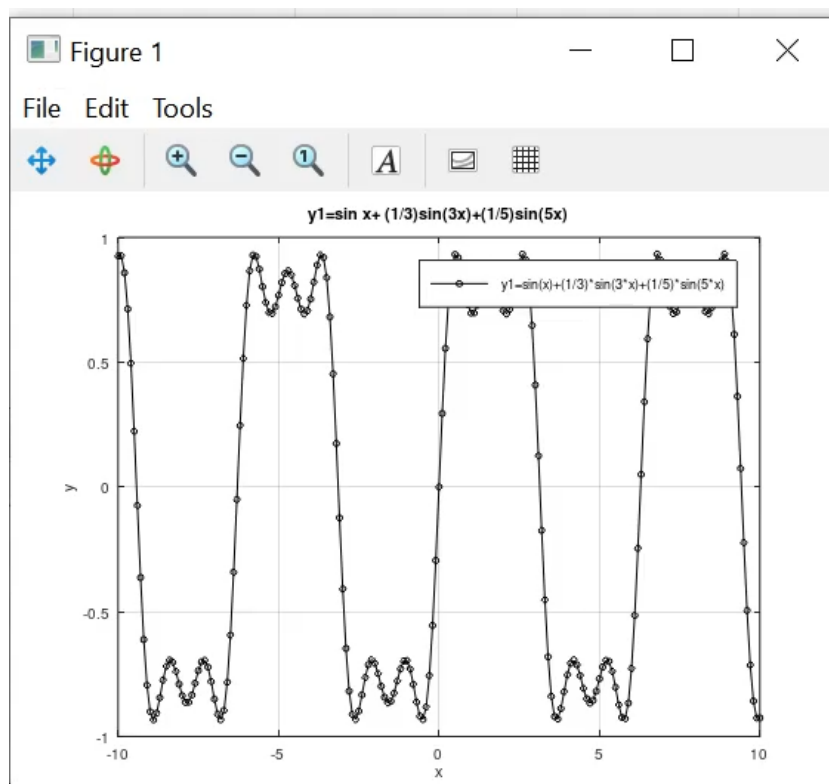


Рис. 2.2: График функции

Сохраняем файл под другим именем, добавляем на график линию

$$y2 = \cos(x) + \frac{1}{3}\cos(3x) + \frac{1}{5}\cos(5x)$$

и запускаем (рис. 2.3)

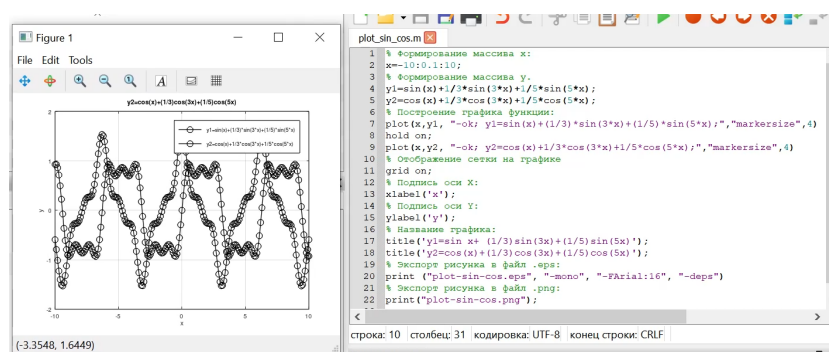


Рис. 2.3: Добавление линии на график

Создаем сценарий meandr.m для демонстрации графиков меандра, реализован-

ных с разным количеством гармоник (рис. 2.4)

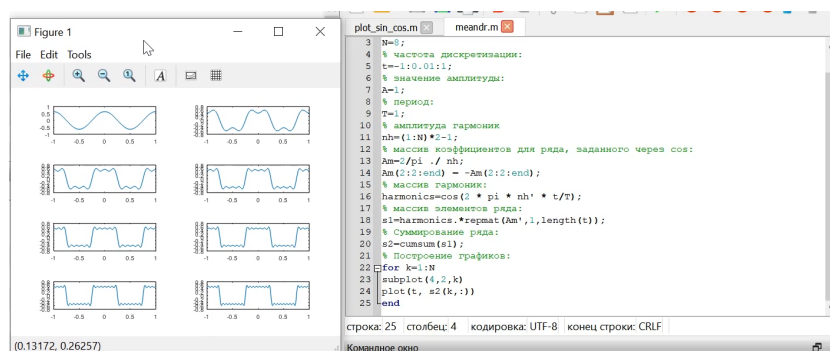


Рис. 2.4: Графики меандра с разным количеством гармоник

Добавляю в листинг строки для экспорта графика в .png. Корректирую код для реализации меандра через синусы (рис. 2.5)

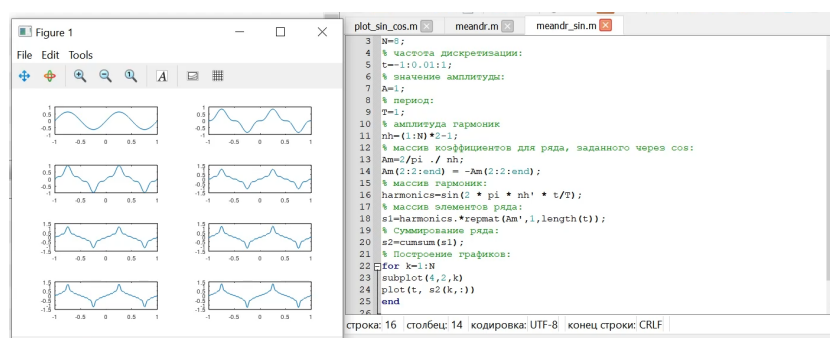


Рис. 2.5: Графики меандра с разным количеством гармоник через синусы

Определяем спектр двух отдельных сигналов и их суммы. В рабочем каталоге создаем каталог spectre1 и в нём новый сценарий с именем spectre.m. Запускаем сценарий (рис. 2.6)

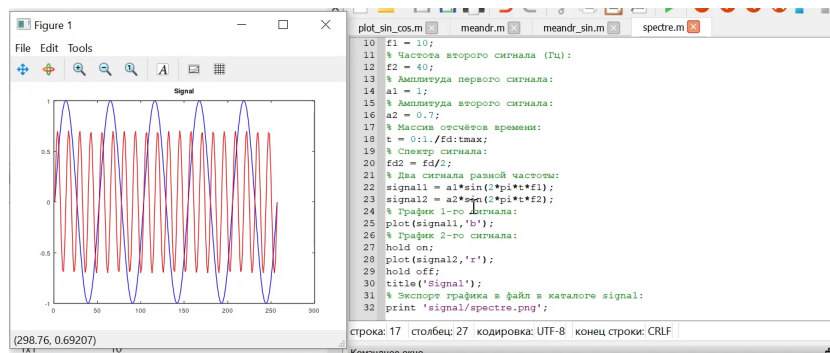


Рис. 2.6: Два синусоидальных сигнала разной частоты

Находим спектр суммы рассмотренных сигналов, создав каталог `spectre_sum` и файл в нём `spectre_sum.m`. В результате получается аналогичный предыдущему результат, т.е. спектр суммы сигналов должен быть равен сумме спектров сигналов, что вытекает из свойств преобразования Фурье (рис. 2.7)

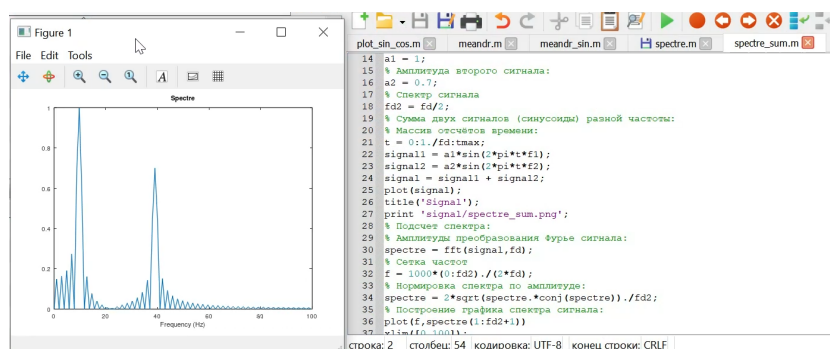


Рис. 2.7: Спектр суммарного сигнала

Создаем каталог `modulation` и в нём новый сценарий с именем `am.m` для демонстрации принципов модуляции сигнала на примере аналоговой амплитудной модуляции. В результате получаем, что спектр произведения представляет собой свёртку спектров (рис. 2.8).

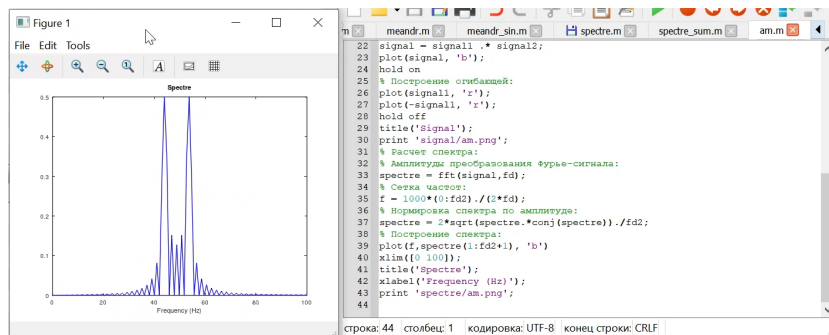


Рис. 2.8: Спектр сигнала при амплитудной модуляции

В рабочем каталоге создаем каталог coding и в нём файлы main.m, maptowave.m, unipolar.m, ami.m, bipolarnrz.m, bipolarrrz.m, manchester.m, diffmanc.m, calcspectre.m. и ввожу код (рис. 2.9).

```

87 print 'spectre/ami.png';
88 % Кодирование NRZ:
89 wave=bipolarnrz(data_spectre);
90 spectre=calcspectre(wave);
91 title('Bipolar Non-Return to Zero');
92 print 'spectre/bipolarnrz.png';
93 % Кодирование RZ:
94 wave=bipolarrrz(data_spectre);
95 spectre=calcspectre(wave);
96 title('Bipolar Return to Zero');
97 print 'spectre/bipolarrrz.png';
98 % Манчестерское кодирование:
99 wave=manchester(data_spectre);
100 spectre=calcspectre(wave);
101 title('Manchester');
102 print 'spectre/manchester.png';
103 % Дифференциальное манчестерское кодирование:
104 wave=diffmanc(data_spectre);
105 spectre=calcspectre(wave);
106 title('Differential Manchester');
107 print 'spectre/diffmanc.png';
108

```

Рис. 2.9: Создание и заполнение файлов в каталоге coding

Запустив файл main.m, получаем графики. В каталоге signal получены файлы с графиками кодированного сигнала (рис. 2.10), в каталоге sync — файлы с графиками, иллюстрирующими свойства самосинхронизации (рис. 2.11), в каталоге spectre — файлы с графиками спектров сигналов (рис. 2.12).

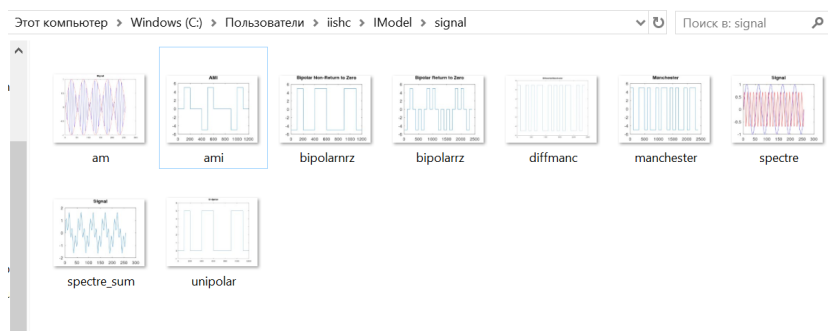


Рис. 2.10: Файлы с графиками кодированного сигнала

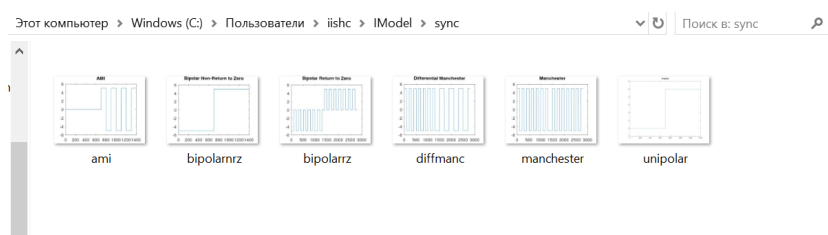


Рис. 2.11: Файлы с графиками, иллюстрирующими свойства самосинхронизации

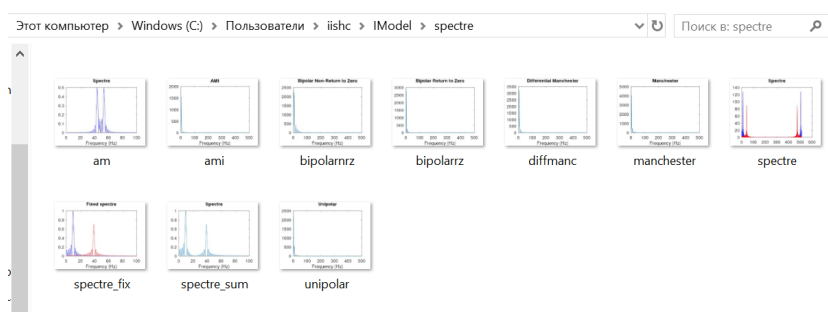


Рис. 2.12: Файлы с графиками спектров сигналов

3 Выводы

В ходе лабораторной работы я изучила методы кодирования и модуляции сигналов с помощью высокоуровневого языка программирования Octave, определение спектра и параметров сигнала. Произвела демонстрацию принципов модуляции сигнала на примере аналоговой амплитудной модуляции. Исследовала свойства самосинхронизации сигнала.