

Лабораторная №1

Сетевые технологии

Жибицкая Е.Д.

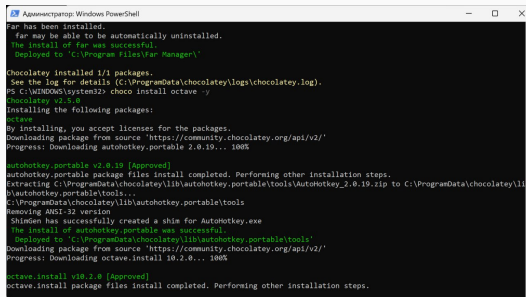
Российский университет дружбы народов, Москва, Россия

Цель

- 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

Ход работы

Перед началом работы подготавливаем пространство, скачиваем все необходимое



```
Администратор: Windows PowerShell
Far has been installed.
far may be able to be automatically uninstalled.
The install of far was successful.
Deployed to 'C:\Program Files\Far Manager\'

Chocolatey installed 1/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
PS C:\WINDOWS\system32> choco install octave -y
Chocolatey v2.5.0
Installing the following packages:
octave
By installing, you accept licenses for the packages.
Downloading package from source 'https://community.chocolatey.org/api/v2/'
Progress: Downloading autohotkey.portable 2.0.19... 100%

autohotkey.portable v2.0.19 [Approved]
autohotkey.portable package files install completed. Performing other installation steps.
Extracting C:\ProgramData\chocolatey\lib\autohotkey.portable\tools\AutoHotkey_2.0.19.zip to C:\ProgramData\chocolatey\lib\autohotkey.portable\tools...
C:\ProgramData\chocolatey\lib\autohotkey.portable\tools
Removing ANSI-32 version
ShimGen has successfully created a shim for AutoHotkey.exe
The install of autohotkey.portable was successful.
Deployed to 'C:\ProgramData\chocolatey\lib\autohotkey.portable\tools'
Downloading package from source 'https://community.chocolatey.org/api/v2/'
Progress: Downloading octave.install 10.2.0... 100%

octave.install v10.2.0 [Approved]
octave.install package files install completed. Performing other installation steps.
```

Рис. 1: Установка Octave

Sin_plot. Построение графика

```
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);", "markersi
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: Код для sin_plot

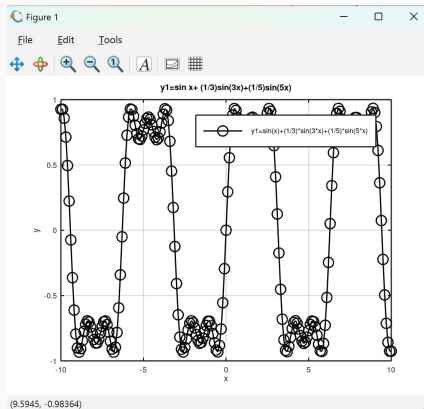


Рис. 3: График sin_plot

Sin_cos_plot

```
plot_sin_cos.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 y2=cos(x)+1/3*cos(3*x)+1/5*cos(5*x);
6 % Построение графика функции:
7 plot(x,y1, "-ok; y1=sin(x)+(1/3)*sin(3*x)+(1/5)*sin(5*x);", "markersi
8 hold on;
9 plot(x,y2, "-k; y2=cos(x)+1/3*cos(3*x)+1/5*cos(5*x);", "markersize", 4)
10 % Отображение сетки на графике
11 grid on;
12 % Подпись оси X:
13 xlabel('x');
14 % Подпись оси Y:
15 ylabel('y');
16 % Название графика:
17 title('y1=sin x+ (1/3)sin(3x)+(1/5)sin(5x), y2=cos(x)+1/3*cos(3*x)+1,
18 % Экспорт рисунка в файл .eps:
19 print ("plot-sin_cos.eps", "-mono", "-FArial:16", "-deps")
20 % Экспорт рисунка в файл .png:
21 print ("plot-sin_cos.png");
22
```

Рис. 4: Код для sin_cos_plot

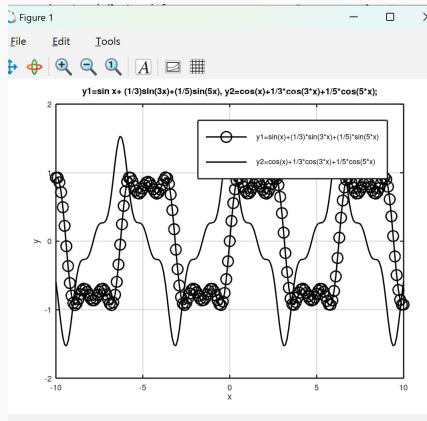


Рис. 5: График sin_cos_plot

Разложение импульсного сигнала в частичный ряд Фурье

```
meandr.m
1 % meandr.m
2 % количество отсчетов (гармоник):
3 N=8;
4 % частота дискретизации:
5 t=-1:0.01:1;
6 % значение амплитуды:
7 A=1;
8 % период:
9 T=1;
10 % амплитуда гармоник
11 nh=(1:N)*2-1;
12 % массив коэффициентов для ряда, заданного через cos:
13 Am=2/pi ./ nh;
14 Am(2:2:end) = -Am(2:2:end);
15 % массив гармоник:
16 harmonics=cos(2 * pi * nh' * t/T);
17 % массив элементов ряда:
18 s1=harmonics.*repmat(Am',1,length(t));
19 % Суммирование ряда:
20 s2=cumsum(s1);
21 % Построение графиков:
22 for k=1:N
23     subplot(4,2,k)
24     plot(t, s2(k,:))
25 end
26 % Экспорт рисунка в файл .png:
27 print("plot-meandr.png");
28
```

Рис. 6: Код для meandr.m

```
1 % meandr.m
2 % количество отсчетов (гармоник):
3 N=8;
4 % частота дискретизации:
5 t=-1:0.01:1;
6 % значение амплитуды:
7 A=1;
8 % период:
9 T=1;
10 % амплитуда гармоник
11 nh=(1:N)*2-1;
12 % массив коэффициентов для ряда, заданного через sin:
13 Am=2/pi ./ nh;
14 Am(2:2:end) = Am(2:2:end);
15 % массив гармоник:
16 harmonics=sin(2 * pi * nh' * t/T);
17 % массив элементов ряда:
18 s1=harmonics.*repmat(Am',1,length(t));
19 % Суммирование ряда:
20 s2=cumsum(s1);
21 % Построение графиков:
22 for k=1:N
23     subplot(4,2,k)
24     plot(t, s2(k,:))
25 end
26 % Экспорт рисунка в файл .png:
27 print("plot-meandr_sin.png");
28
```

Рис. 7: Код для meandr.m через синус

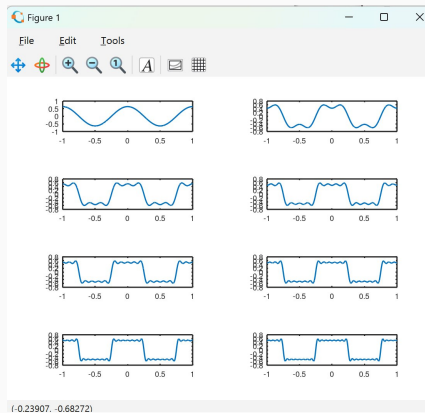
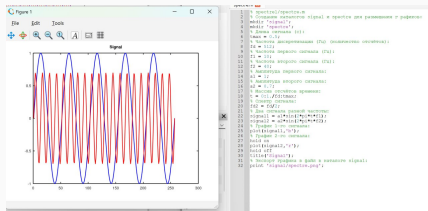


Рис. 8: График meandr.m

Определение спектра и параметров сигнала



Создаем каталог `spectre1` и в нем файл `spectre.m`

Рис. 9: График spectre.m

Доработка кода и графика

```
18 t = 0:1./fd:tmax;
19 % Спектр сигнала:
20 fd2 = fd/2;
21 % Два сигнала разной частоты:
22 signal1 = a1*sin(2*pi*t*f1);
23 signal2 = a2*sin(2*pi*t*f2);
24 % График 1-го сигнала:
25 plot(signal1,'b');
26 % График 2-го сигнала:
27 hold on
28 plot(signal2,'r');
29 hold off
30 title('Signal');
31 % Экспорт графика в файл в каталоге signal:
32 print 'signal/spectre.png';
33 % Посчитаем спектр
34 % Амплитуды преобразования Фурье сигнала 1:
35 spectrel = abs(fft(signal1,fd));
36 % Амплитуды преобразования Фурье сигнала 2:
37 spectre2 = abs(fft(signal2,fd));
38 % Построение графиков спектров сигналов:
39 plot(spectrel,'b');
40 hold on
41 plot(spectre2,'r');
42 hold off
43 title('Spectre');
44 print 'spectre/spectre.png';
45 % Исправление графика спектра
46 % Сетка частот:
47 f = 1000*(0:fd2)./(2*fd);
48 % Нормировка спектров по амплитуде:
49 spectrel = 2*spectrel/fd2;
50 spectre2 = 2*spectre2/fd2;
51 % Построение графиков спектров сигналов:
52 plot(f,spectrel(1:fd2+1),'b');
53 hold on
54 plot(f,spectre2(1:fd2+1),'r');
55 hold off
56 xlim([0 100]);
57 title('Fixed spectre');
58 xlabel('Frequency (Hz)');
59 print 'spectre/spectre_fix.png';
60
```

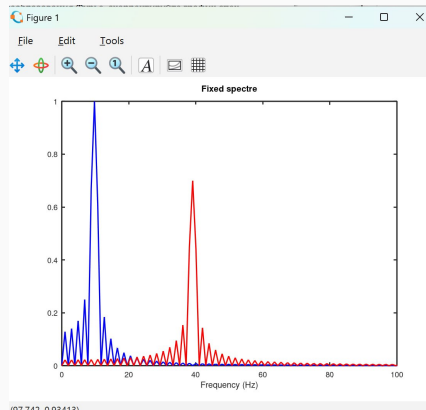


Рис. 11: Исправленный график

Рис. 10: Доработка кода

Спектр суммы рассмотренных сигналов

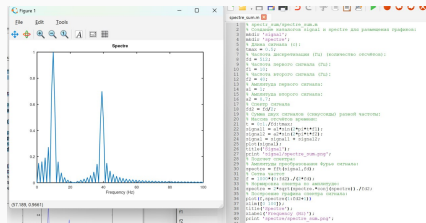


Рис. 12: Суммарный сигнал

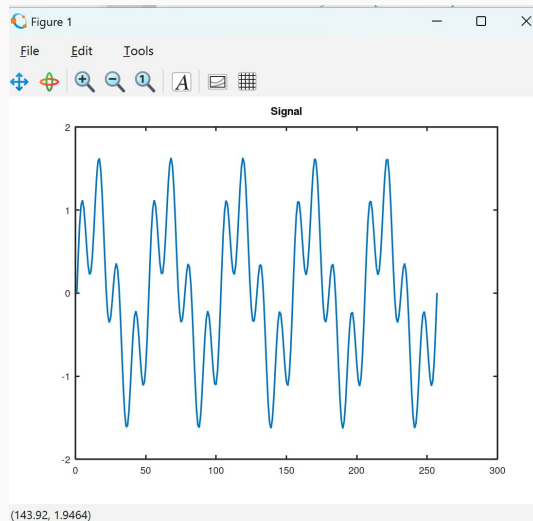


Рис. 13: Спектр суммарного сигнала

Ознакомимся с амплитудной модуляцией, создадим каталог modulation со сценарием am.m, увидим, что спектр произведения представляет собой свертку спектров

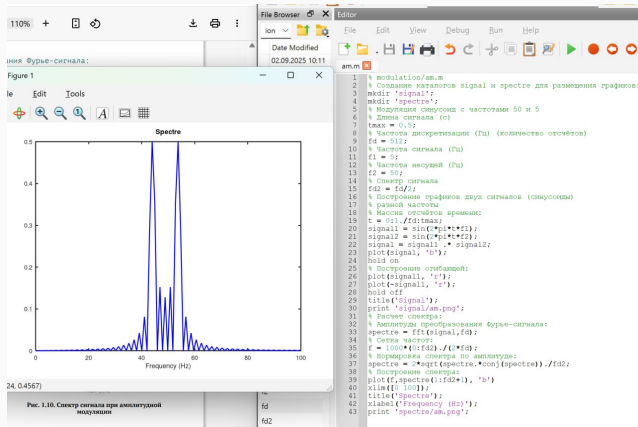


Рис. 14: Спектр сигнала при амплитудной модуляции

Свойства самосинхронизируемости кодов по заданным битовым последовательностям










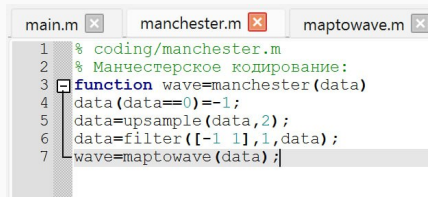
Имя	Дата изменения	Тип
 main.m	02.09.2025 10:15	Файл "М"
 maptowave.m	02.09.2025 10:16	Файл "М"
 unipolar.m	02.09.2025 10:18	Файл "М"
 ami.m	02.09.2025 10:19	Файл "М"
 bipolarnrz.m	02.09.2025 10:19	Файл "М"
 bipolarrrz.m	02.09.2025 10:20	Файл "М"
 manchester.m	02.09.2025 10:20	Файл "М"
 diffmanc.m	02.09.2025 10:21	Файл "М"
 calcspectre.m	02.09.2025 10:21	Файл "М"

Рис. 15: Подготовка рабочего пространства

Создаем для работы каталог coding и в нём файлы main.m, maptowave.m, unipolar.m, ami.m, bipolarnrz.m, bipolarrrz.m, manchester.m, diffmanc.m, calcspectre.m, убеждаемся, что у нас установлен пакет signal



```
1 % coding/manchester.m
2 % Манчестерское кодирование:
3 function wave=manchester(data)
4 data(data==0)=-1;
5 data=upsample(data,2);
6 data=filter([-1 1],1,data);
7 wave=maptowave(data);
```

Рис. 16: Функции

В файле main.m подключаем пакет signal, задаем входные кодовые последовательности, прописываем вызовы функций построения спектров, модуляций сигналов. В остальных файлах прописываем сами функции



Рис. 17: Графики

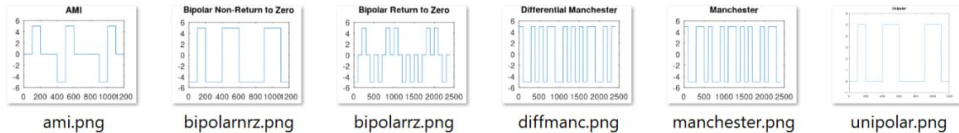


Рис. 18: Графики

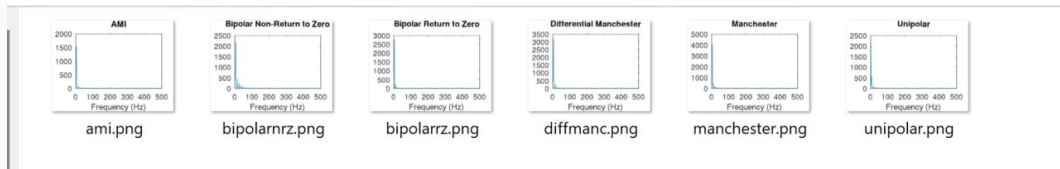


Рис. 19: Графики

Выводы

- В ходе работы было произведено знакомство с Octave. Были также изучены методы кодирования и модуляции сигнала, определены спектры и параметры сигнала, продемонстрированы принципы модуляции сигнала на примере аналоговой амплитудной модуляции и исследованы свойства самосинхронизации сигнала