Лабораторная работа №1

Дисциплина: Сетевые технологии

Жибицкая Евгения Дмитриевна

Содержание

# 1 Цель работы

Знакомство с Octave, получение навыков по работе с ним. Изучение методов кодирования и модуляции сигналов с помощью высокоуровнего 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

Для выполнения работы устанавливаем Octave c помощью Chocolatey(рис. 1).

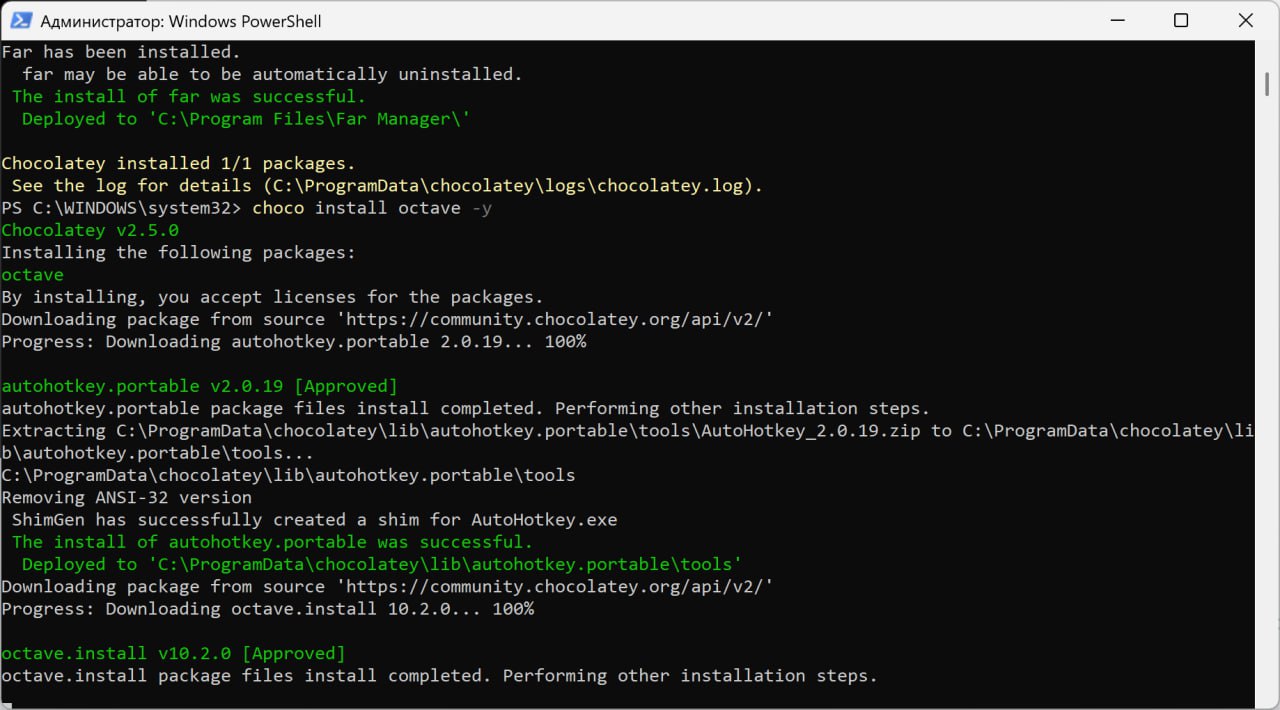


Рис. 1: Установка Octave

Запускаем OCtave GUI, переходим в редактор и создаем новый сценарий plot\_sin.m(рис. 2).

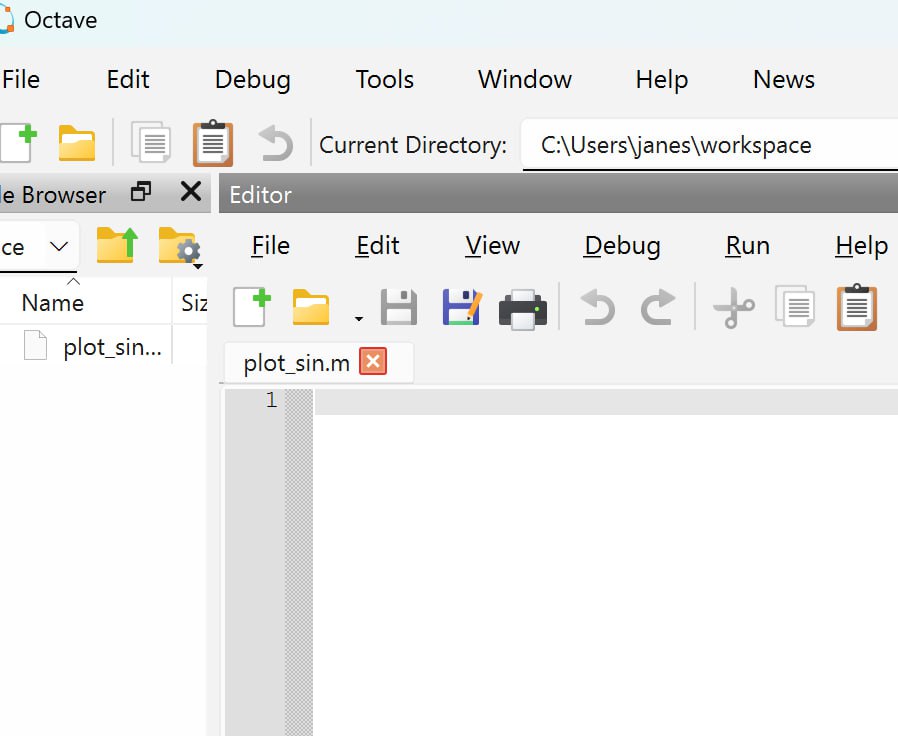


Рис. 2: Создание сценария

Далее добавляем туда листинг(рис. 3) и запускаем сценарий, получаем график(рис. 4).

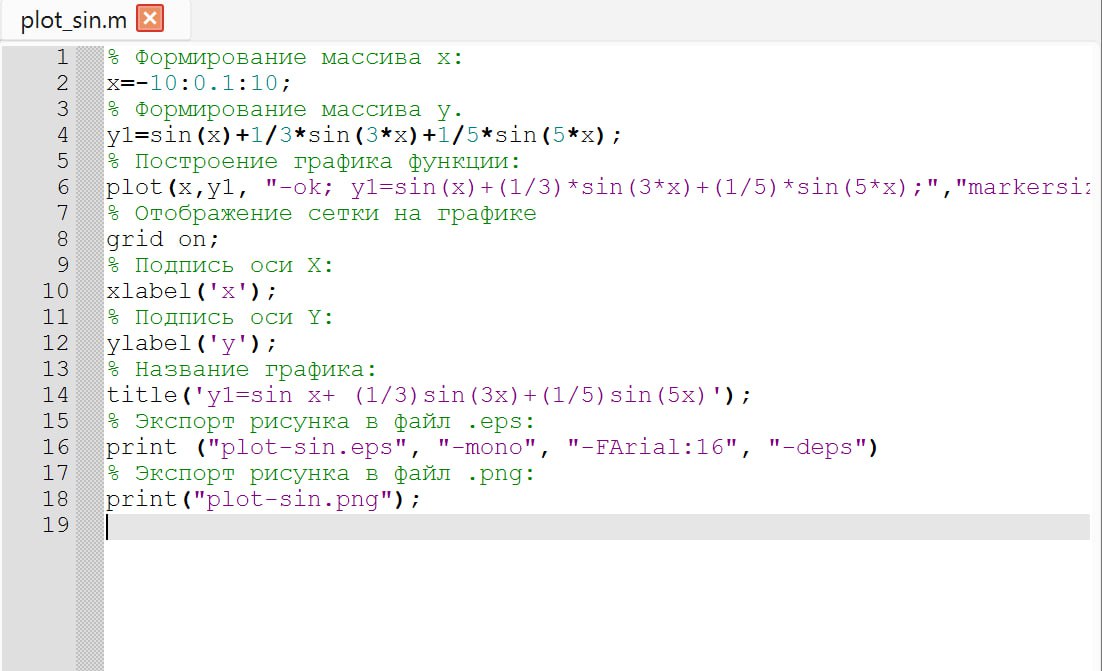


Рис. 3: Код для sin\_plot

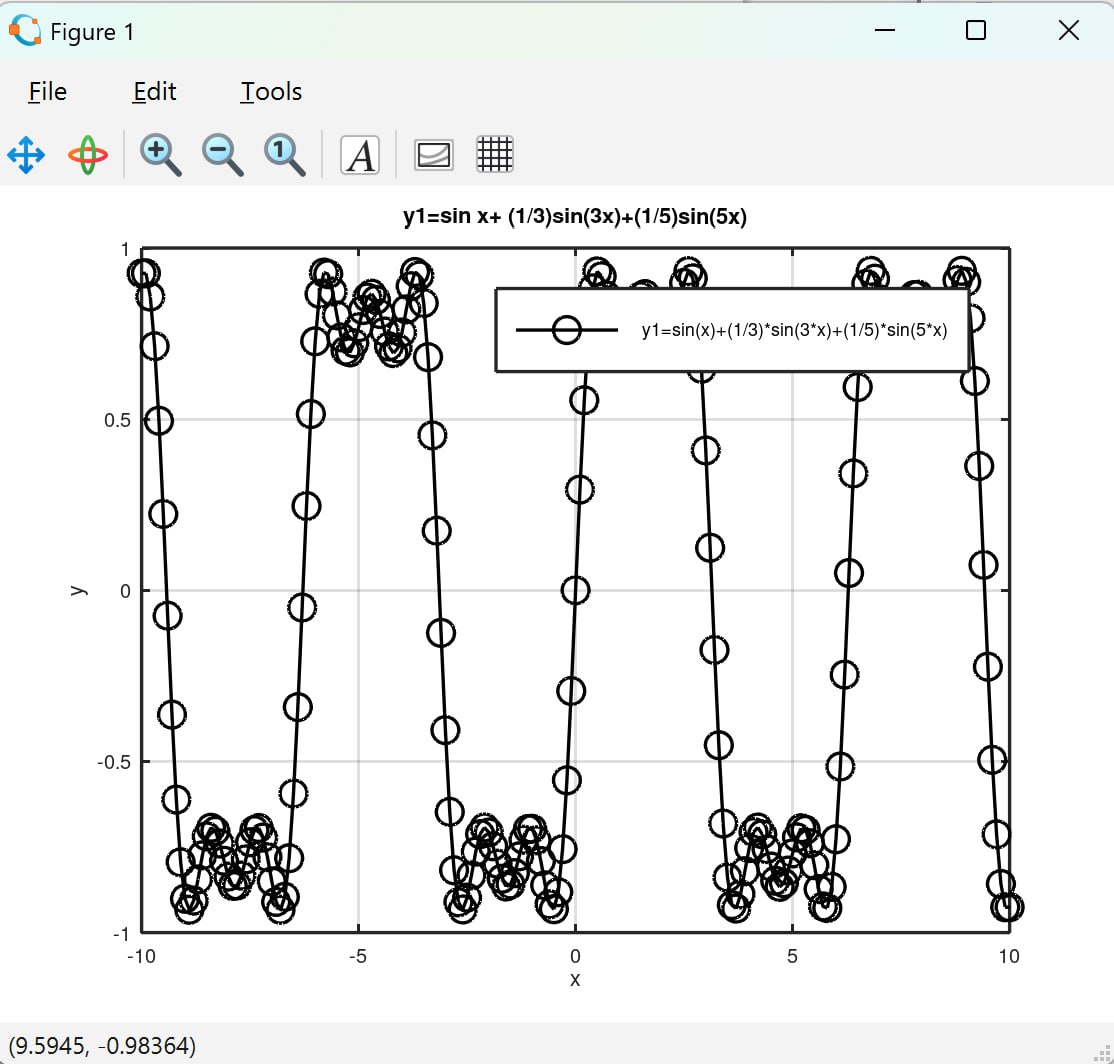


Рис. 4: График sin\_plot

Сохраняем сценарий, поменяв ему имя, и строим теперь 2 графика - для синуса и косинуса(рис. 5) и (рис. 6).

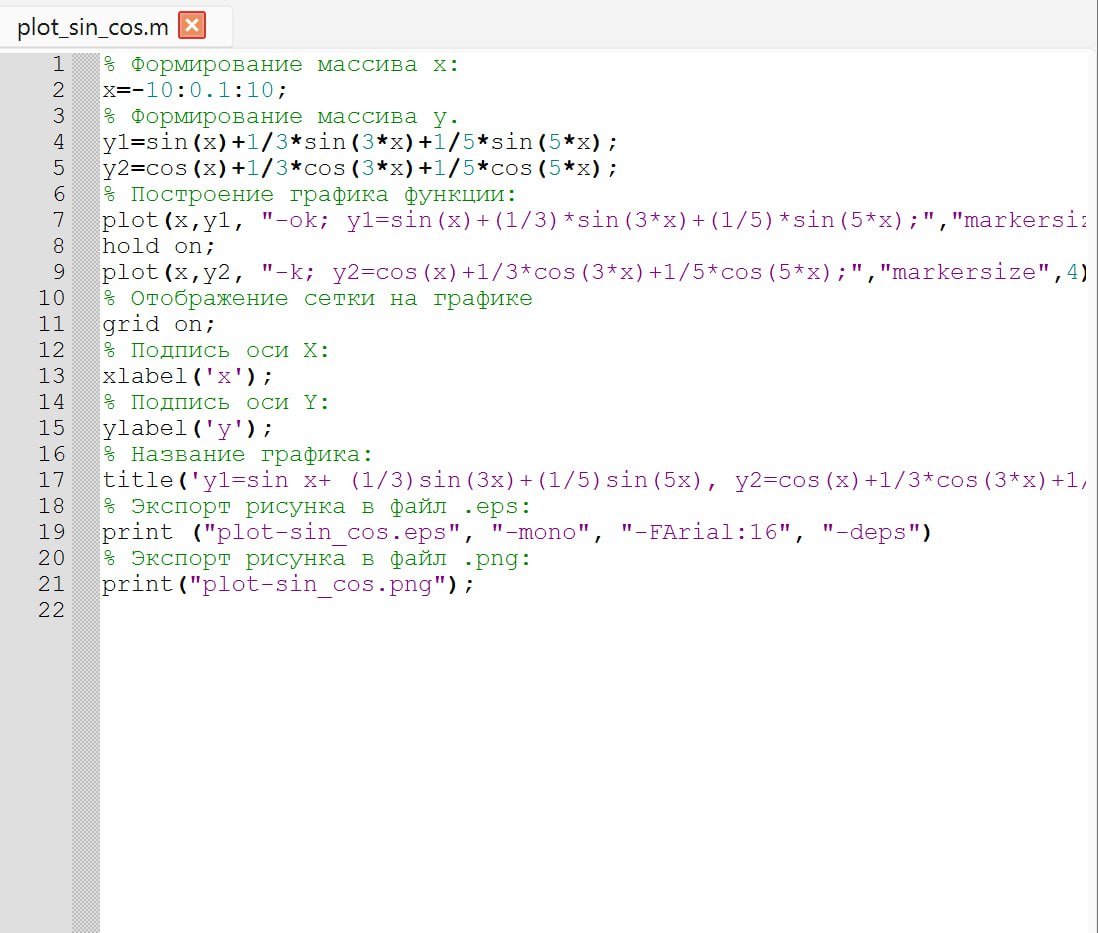


Рис. 5: Код для sin\_cos\_plot

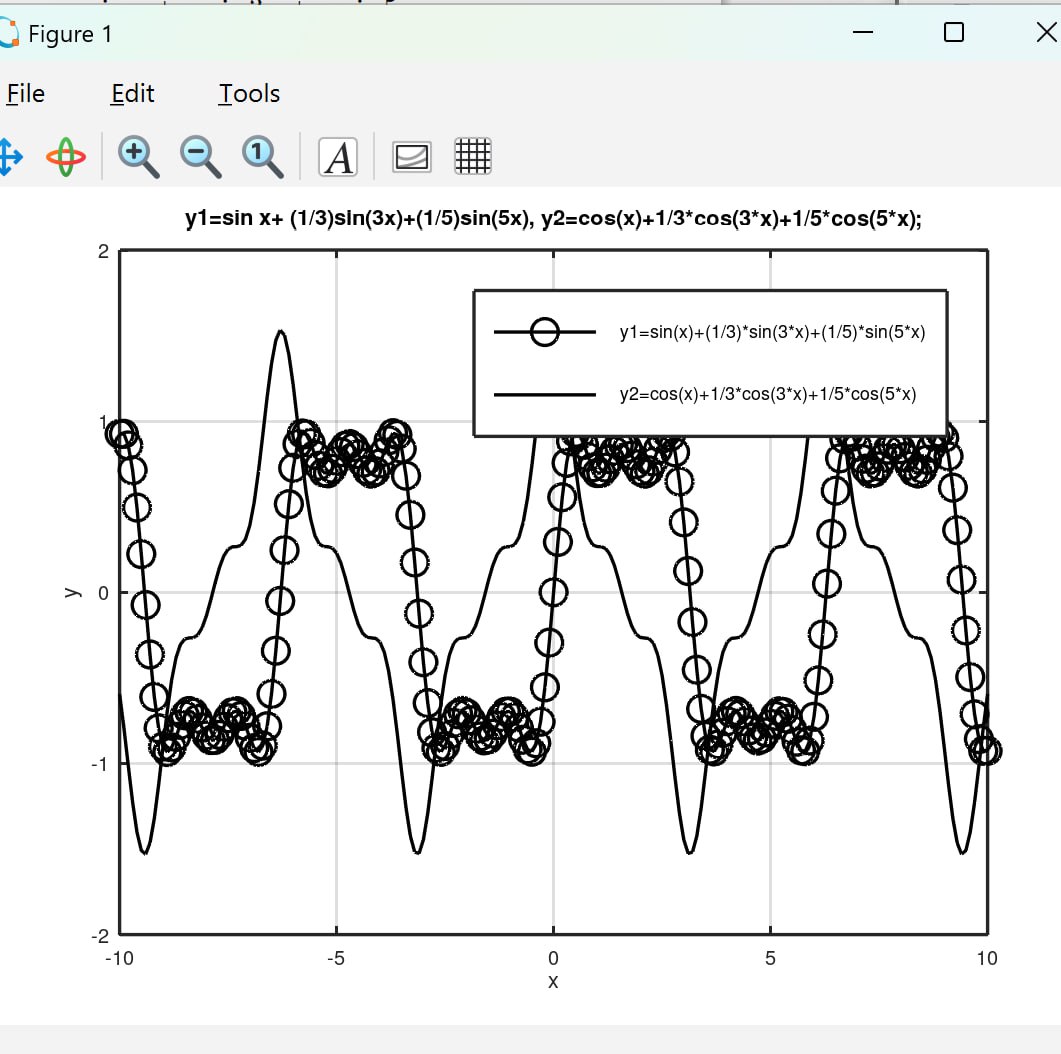


Рис. 6: График sin\_cos\_plot

Затем перейдем к разложению импульсного сигнала в частичный ряд Фурье.

Создаем новый сценарий meandr.m, вставляем код(рис. 7). Аналогично пишем через синус, результат получаем тот же (рис. 8).

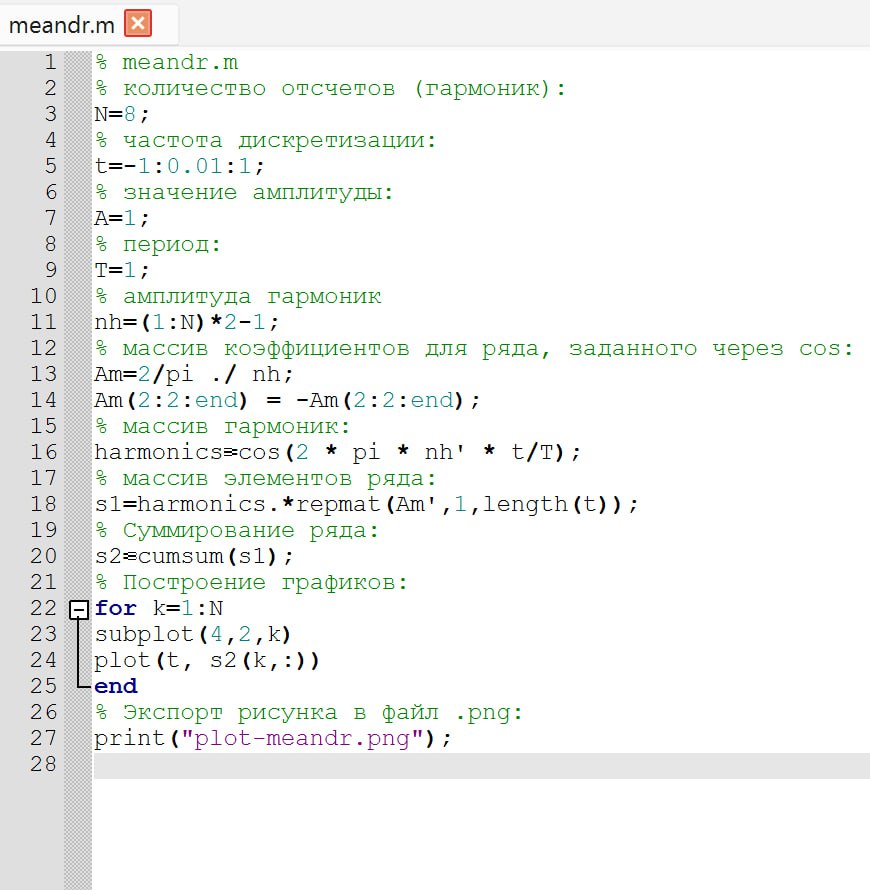


Рис. 7: Код для meandr.m

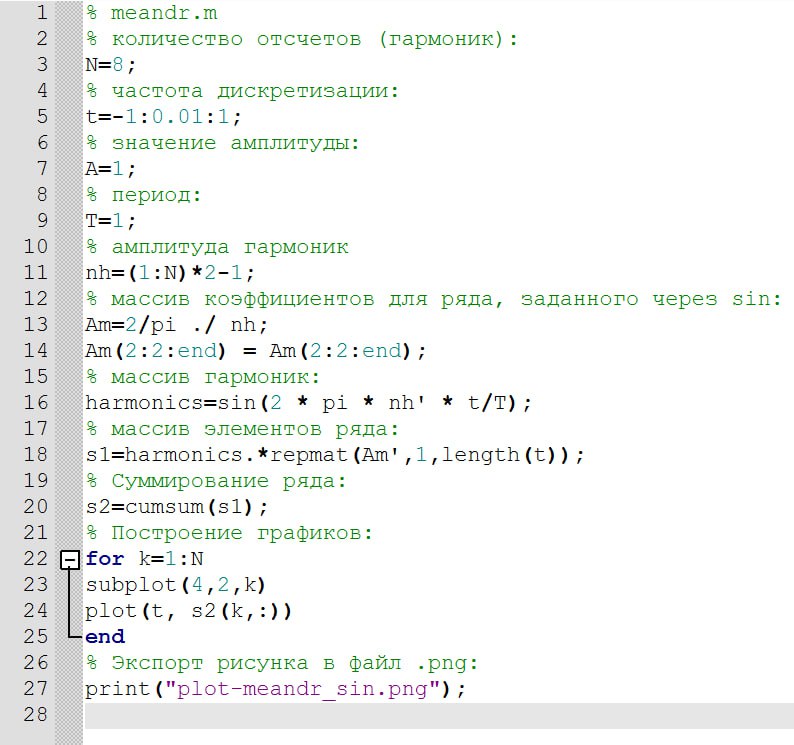


Рис. 8: Код для meandr.m через синус

Смотрим на получившийся результат(рис. 9).

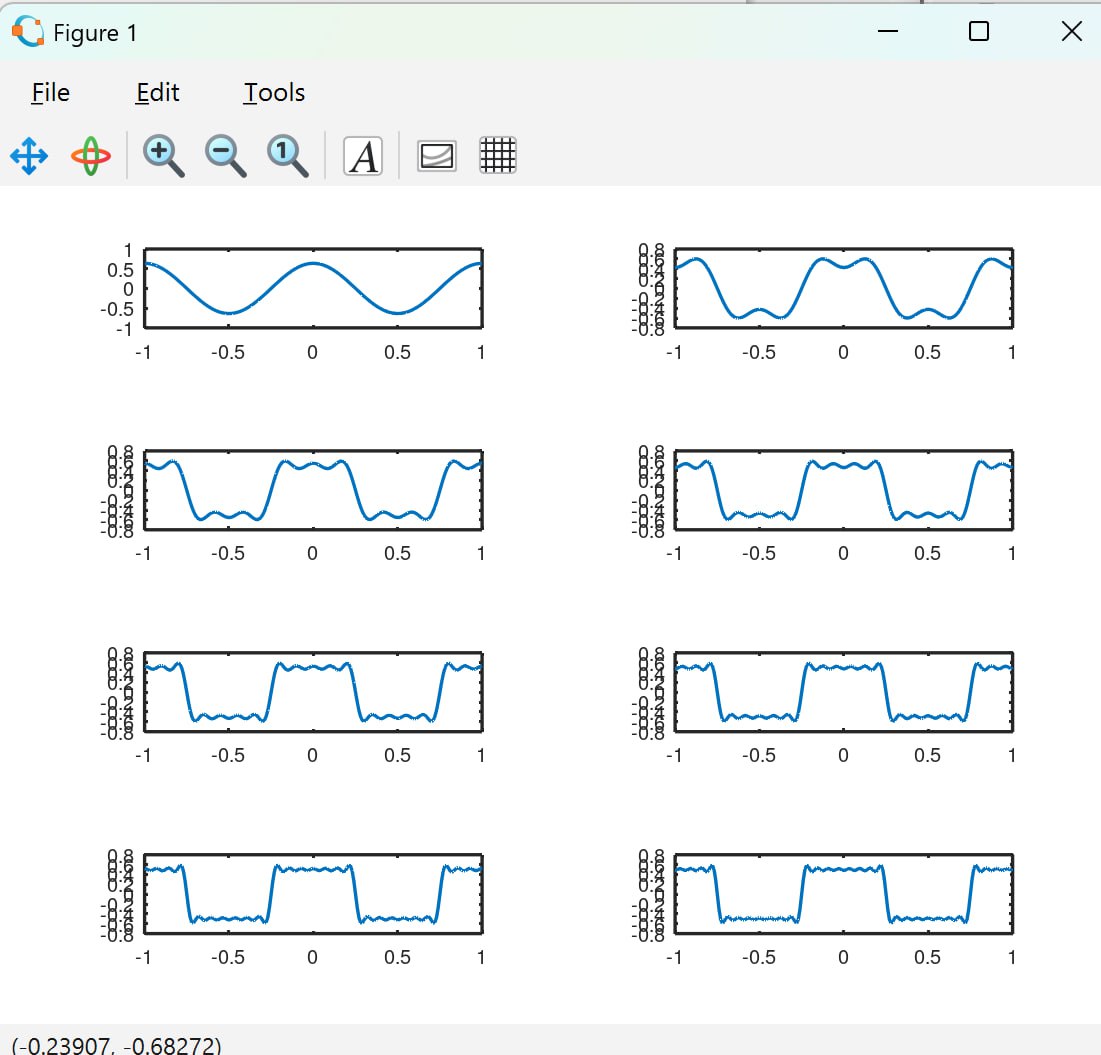


Рис. 9: График meandr.m

Далее определим пектр и параметры сигнала.

Создаем каталог spectre1 и в нем файл spectre.m(рис. 10).

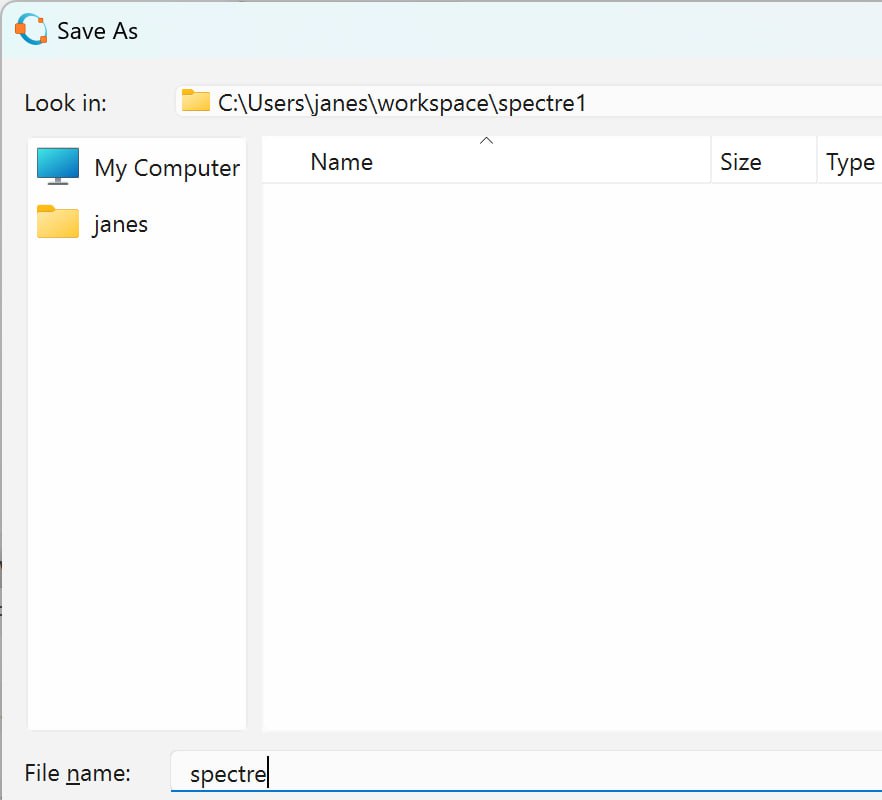


Рис. 10: Создание каталога и файла

Прописываем код и смотрим на результат(рис. 11).

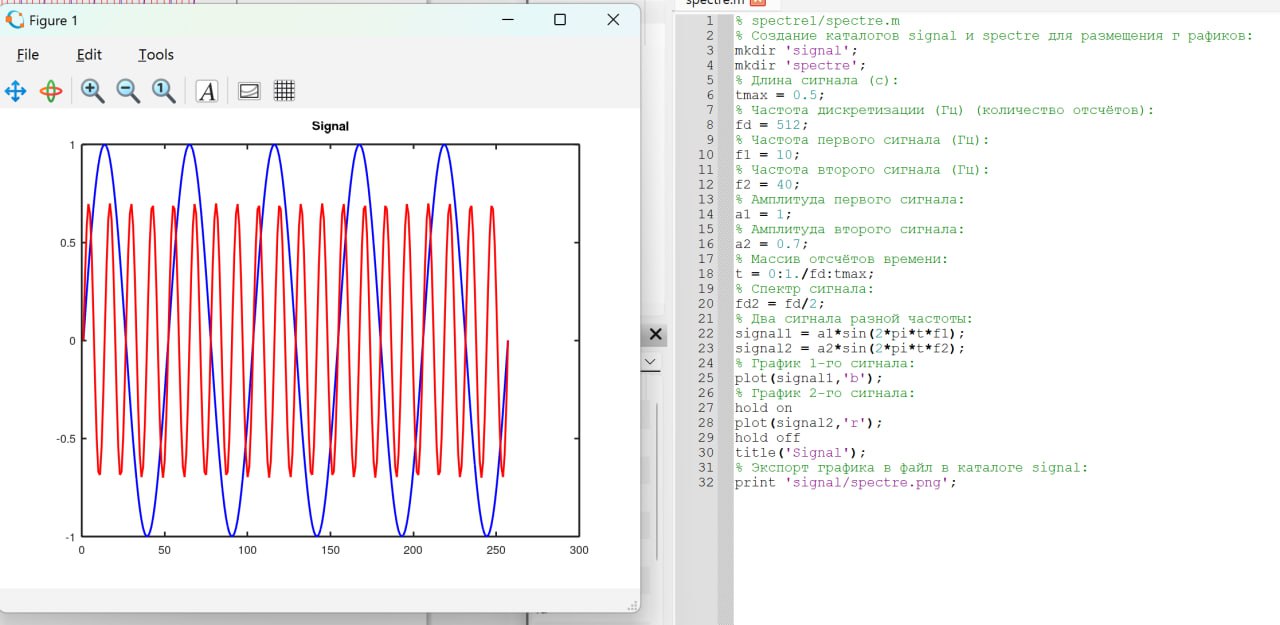


Рис. 11: График spectrе.m

Дорабатываем код(рис. 12), корректируем график, отбрасывая дублирующие отрицательные частоты и принимая в расчёт то, что на каждом шаге вычисления быстрого преобразования Фурье происходит суммирование амплитуд сигнало и получаем следующий его вид(рис. 13).

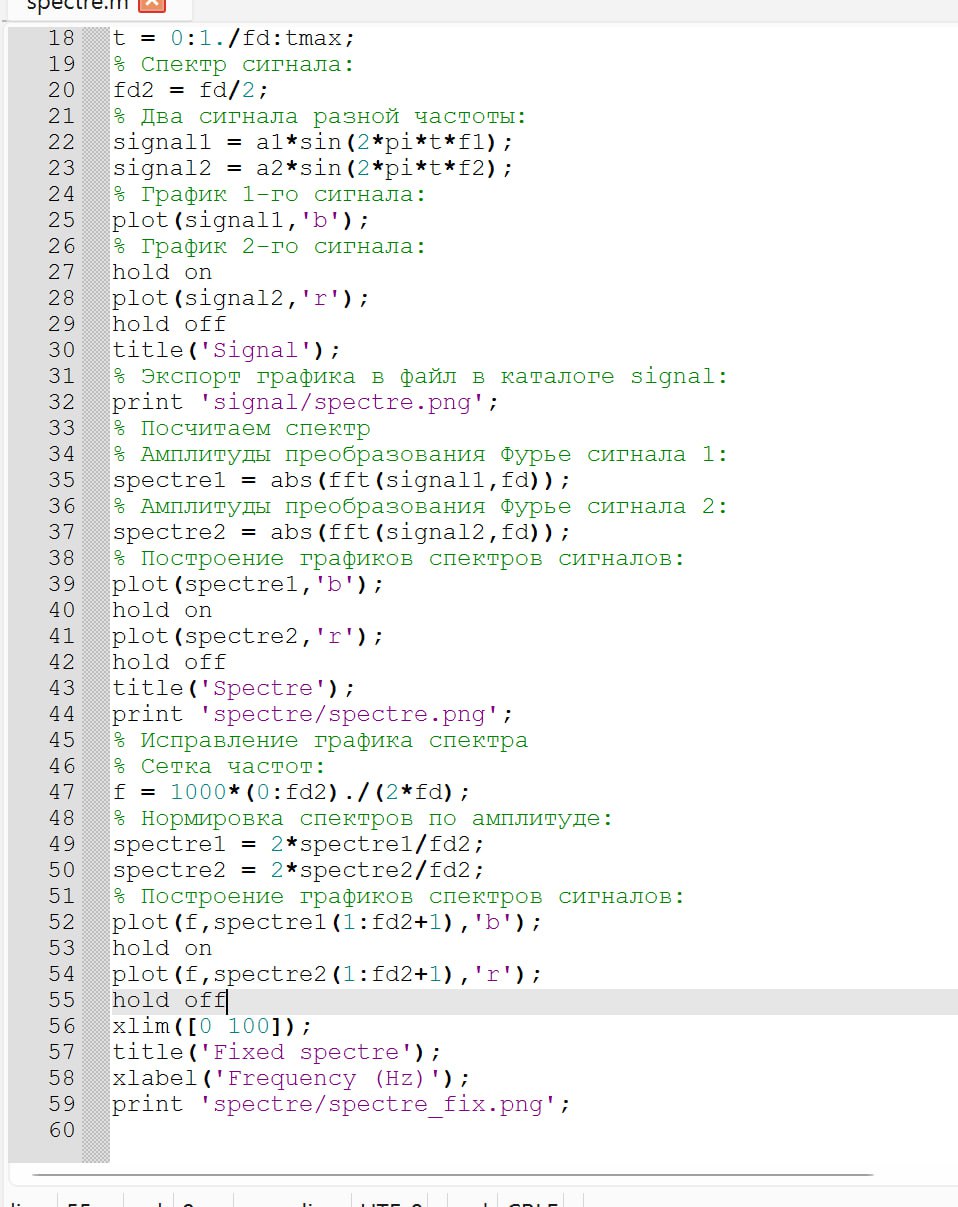


Рис. 12: Доработка кода

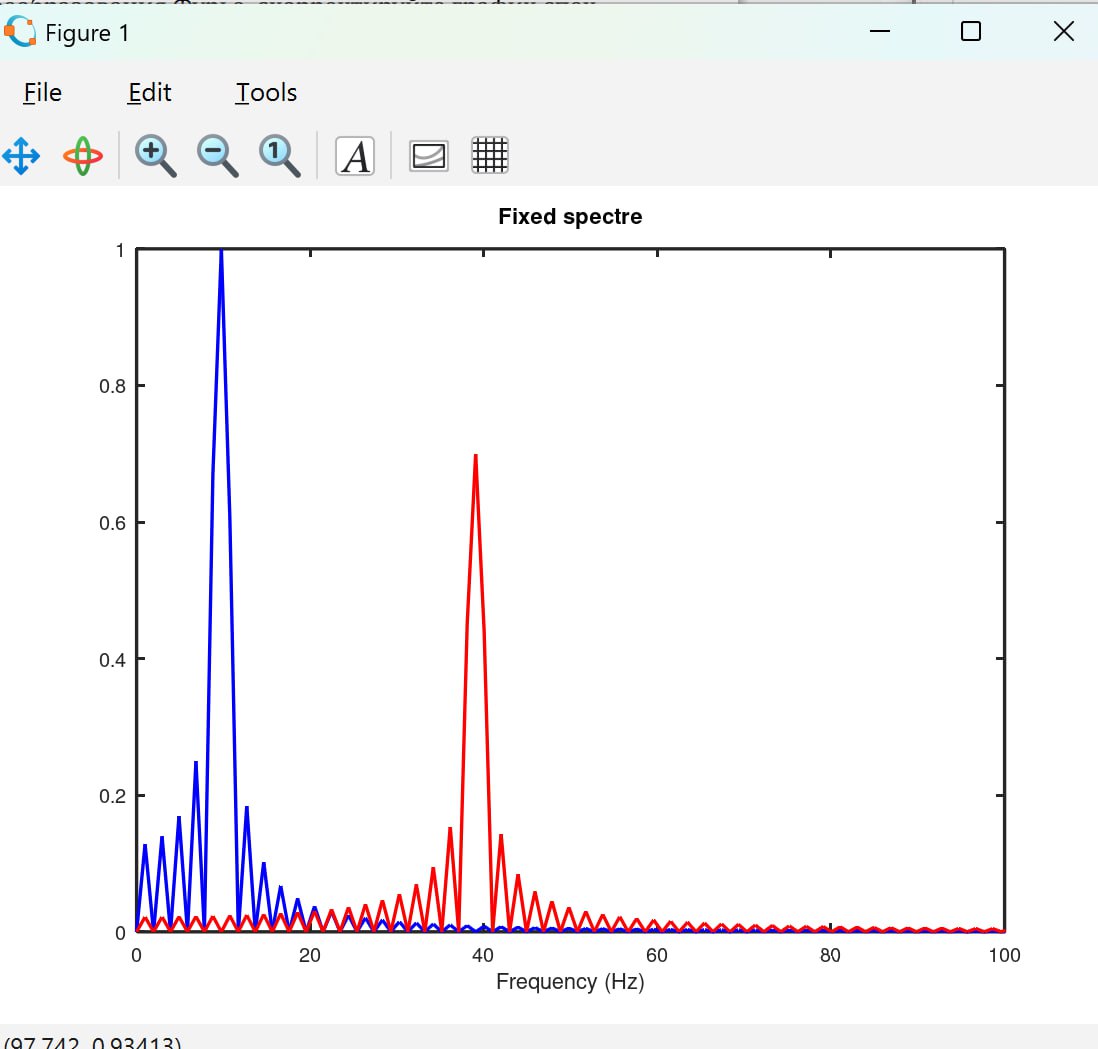


Рис. 13: Исправленный график

Найдем спектр суммы рассмотренных сигналов. Создаем каталог spectr\_sum, файл spectre\_sum с данным кодом, также полцчаем спектр суммарного сигнала(рис. 14) и (рис. 15).

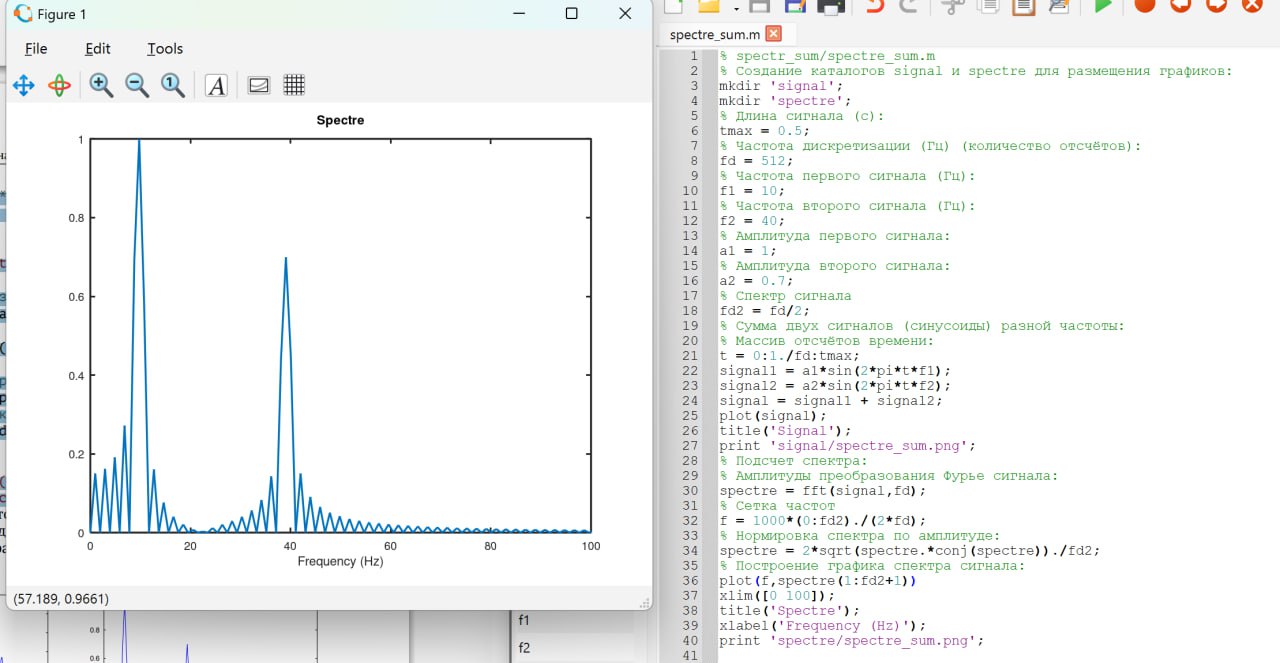


Рис. 14: Суммарный сигнал

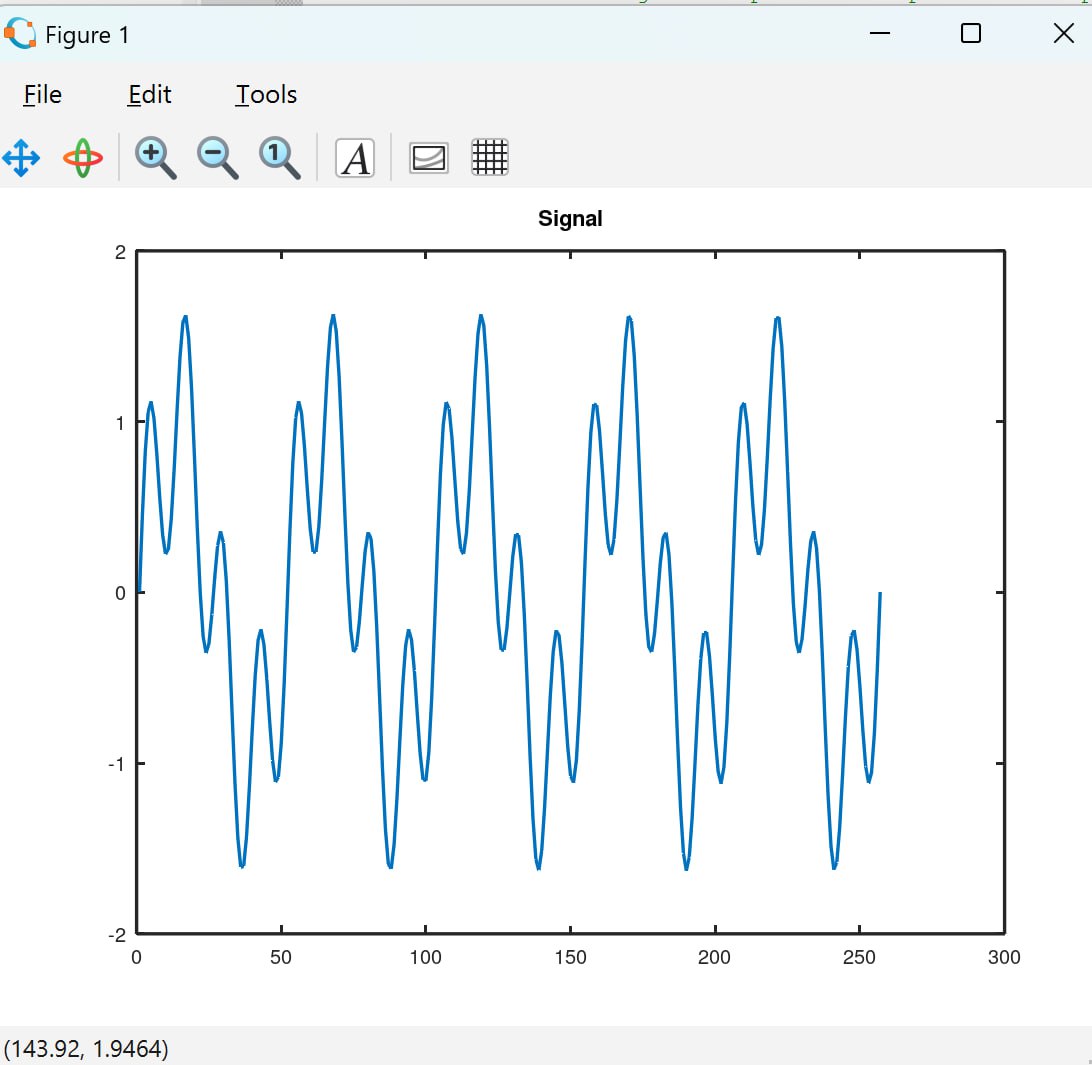


Рис. 15: Спектр суммарного сигнала

Ознакомимся с амплитудной модуляцией, создадим каталог modulation cо сценарием am.m, увидим, что спектр произведения представляет собой свертку спектров(рис. 16)

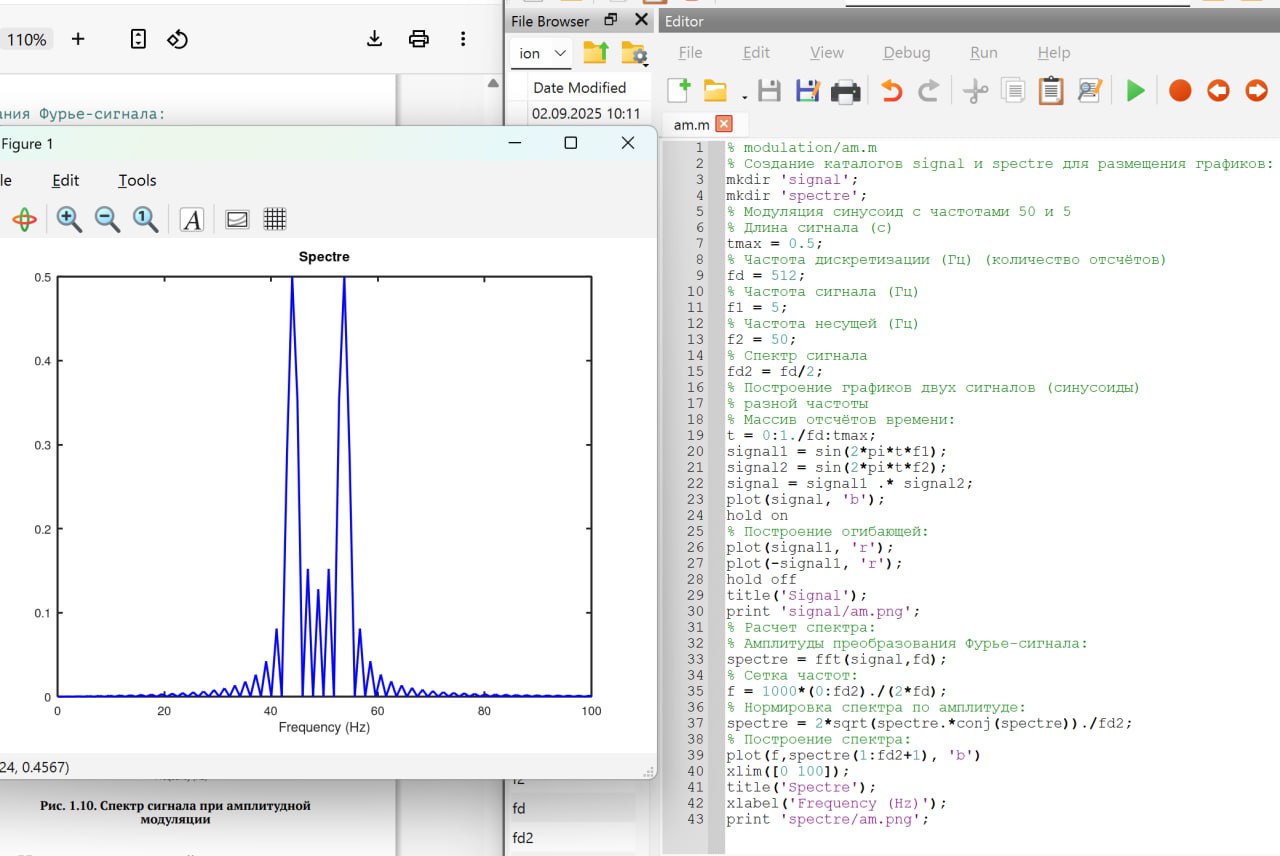


Рис. 16: Спектр сигнала при амплитудной модуляции

Теперь необходимо получить кодированные сигналы для нескольких кодов, проверить свойства самосинхронизуемости кодов по заданной битовой последовательности.

Создаем для работы каталог coding и в нём файлы main.m, maptowave.m,unipolar.m,ami.m,bipolarnrz.m,bipolarrz.m,manchester.m, diffmanc.m, calcspectre.m(рис. 17)

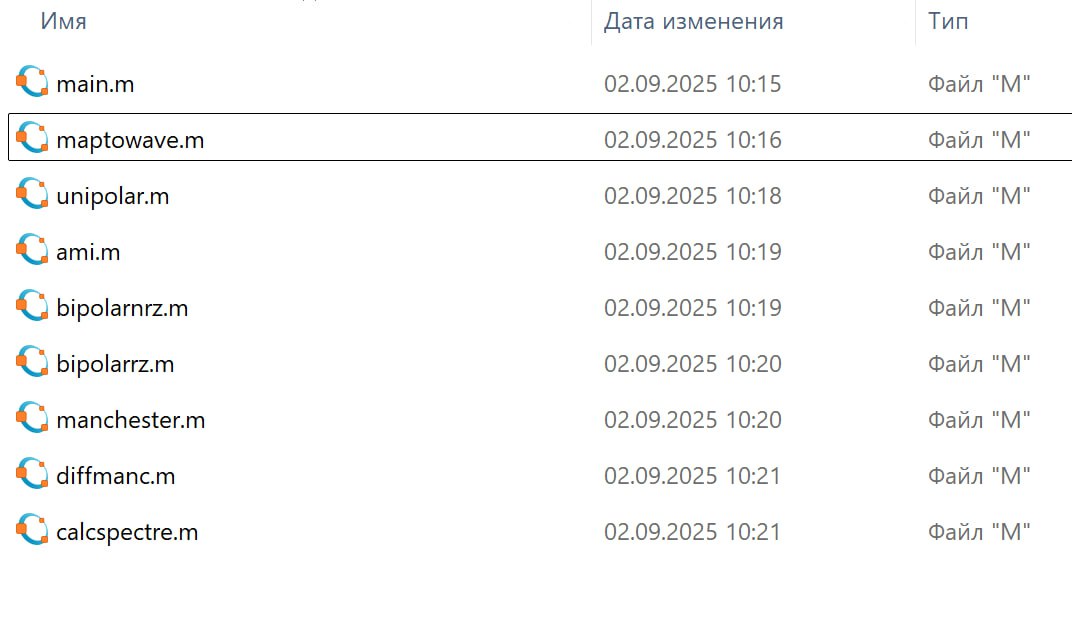


Рис. 17: Подготовка рабочего пространства

Затем убеждаемся, что у нас установлен пакет signal, последовательно добавляем необходимый код в файлы

В файле main.m подключаем пакет signal и задаем входные кодовые последовательности: % coding/main.m % Подключение пакета signal: pkg load signal;

% Входная кодовая последовательность: data=[0 1 0 0 1 1 0 0 0 1 1 0]; % Входная кодовая последовательность для проверки ↪️ свойства самосинхронизации: data\_sync=[0 0 0 0 0 0 0 1 1 1 1 1 1 1]; % Входная кодовая последовательность для построения ↪️ спектра сигнала: data\_spectre=[0 1 0 1 0 1 0 1 0 1 0 1 0 1]; % Создание каталогов signal, sync и spectre для ↪️ размещения графиков: mkdir ‘signal’; mkdir ‘sync’; mkdir ‘spectre’; axis(“auto”);

Затем в этом же файле пропишем вызовы функций для построения графиков модуляций кодированных сигналов для кодовой последовательности data: % Униполярное кодирование wave=unipolar(data); plot(wave); ylim([-1 6]); title(‘Unipolar’); print ‘signal/unipolar.png’; % Кодирование ami wave=ami(data); plot(wave) title(‘AMI’); print ‘signal/ami.png’; % Кодирование NRZ wave=bipolarnrz(data); plot(wave); title(‘Bipolar Non-Return to Zero’); print ‘signal/bipolarnrz.png’; % Кодирование RZ wave=bipolarrz(data); plot(wave) title(‘Bipolar Return to Zero’); print ‘signal/bipolarrz.png’; % Манчестерское кодирование wave=manchester(data); plot(wave) title(‘Manchester’); print ‘signal/manchester.png’;

% Дифференциальное манчестерское кодирование wave=diffmanc(data); plot(wave) title(‘Differential Manchester’); print ‘signal/diffmanc.png’;

Затем в этом же файле пропишем вызовы функций для построения графиков модуляций кодированных сигналов для кодовой последовательности data\_sync: % Униполярное кодирование wave=unipolar(data\_sync); plot(wave); ylim([-1 6]); title(‘Unipolar’); print ‘sync/unipolar.png’; % Кодирование AMI wave=ami(data\_sync); plot(wave) title(‘AMI’); print ‘sync/ami.png’; % Кодирование NRZ wave=bipolarnrz(data\_sync); plot(wave); title(‘Bipolar Non-Return to Zero’); print ‘sync/bipolarnrz.png’; % Кодирование RZ wave=bipolarrz(data\_sync); plot(wave) title(‘Bipolar Return to Zero’); print ‘sync/bipolarrz.png’; % Манчестерское кодирование wave=manchester(data\_sync); plot(wave) title(‘Manchester’); print ‘sync/manchester.png’;

% Дифференциальное манчестерское кодирование wave=diffmanc(data\_sync); plot(wave) title(‘Differential Manchester’); print ‘sync/diffmanc.png’; Далее в этом же файле пропишем вызовы функций для построения графиков спектров: % Униполярное кодирование: wave=unipolar(data\_spectre); spectre=calcspectre(wave); title(‘Unipolar’); print ‘spectre/unipolar.png’; % Кодирование AMI: wave=ami(data\_spectre); spectre=calcspectre(wave); title(‘AMI’); print ‘spectre/ami.png’; % Кодирование NRZ: wave=bipolarnrz(data\_spectre); spectre=calcspectre(wave); title(‘Bipolar Non-Return to Zero’); print ‘spectre/bipolarnrz.png’; % Кодирование RZ: wave=bipolarrz(data\_spectre); spectre=calcspectre(wave); title(‘Bipolar Return to Zero’); print ‘spectre/bipolarrz.png’; % Манчестерское кодирование: wave=manchester(data\_spectre); spectre=calcspectre(wave); title(‘Manchester’); print ‘spectre/manchester.png’; % Дифференциальное манчестерское кодирование: wave=diffmanc(data\_spectre); spectre=calcspectre(wave); title(‘Differential Manchester’); print ‘spectre/diffmanc.png’;

В остальных файлах прописываем функции постороения графиков(рис. 18) и (рис. 19).

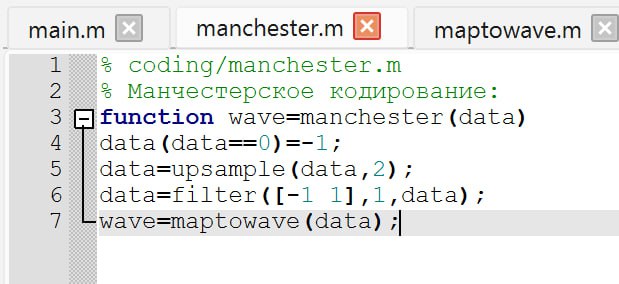


Рис. 18: Функции

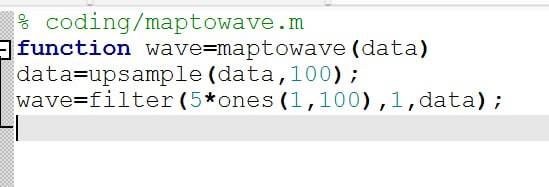


Рис. 19: Фунуции

Запускаем главный сценарий и получаем следующие результаты(рис. 20), (рис. 21) и (рис. 22).

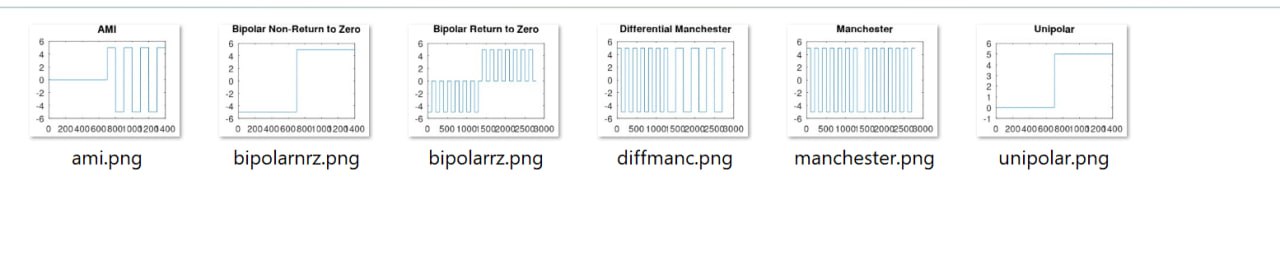


Рис. 20: Графики

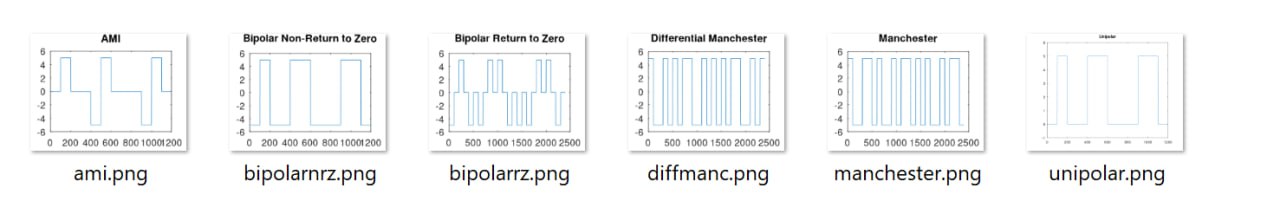


Рис. 21: Графики

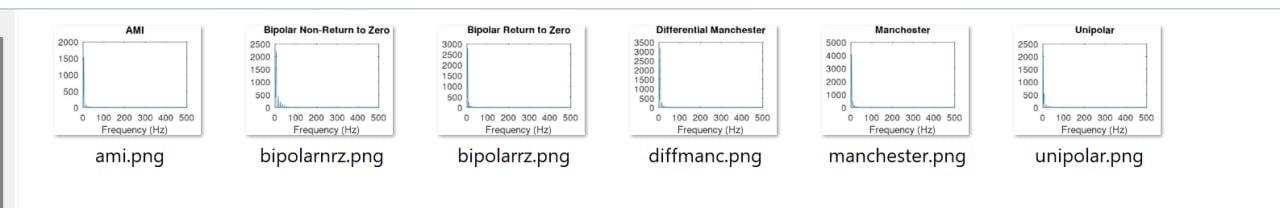


Рис. 22: Графики

# 3 Выводы

В ходе работы было произведено знакомство с Octavе. 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

# Список литературы

[ТУИС](https://esystem.rudn.ru/pluginfile.php/2858347/mod_resource/content/3/001-lab_cod-mod-2.pdf)