Отчет по лабораторной работе №1

Дисциплина: Сетевые технологии

Иванов Сергей Владимирович

Содержание

Список иллюстраций

1 Цель работы

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

2.1 Построение графиков в Octave

Запустим Octave с оконным интерфейсом. Перейдем в окно редактора. Воспользовавшись меню или комбинацией клавиш ctrl + n создадим новый сценарий. Сохраним его в рабочий каталог с именем, plot_sin.m. В окне редактора повторим листинг по построению графика функции y = sin(x) + 1/3 * sin(3x) + 1/5 * sin(5x) на интервале [−10; 10]: (рис. 1).

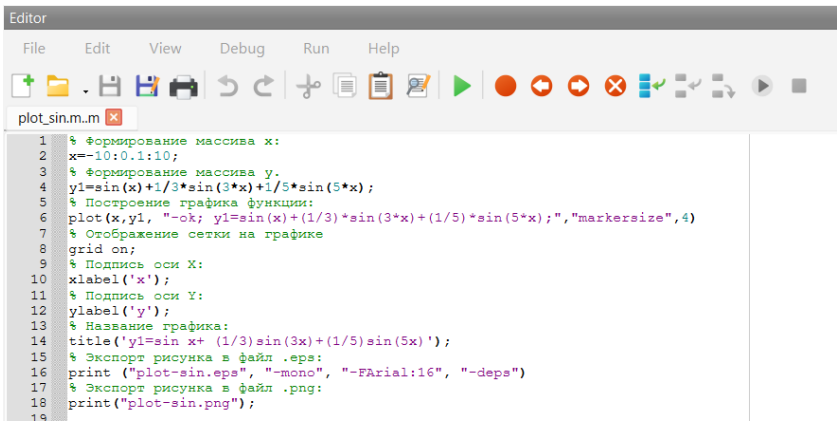

Рис. 2.1: Программа plot_sin.m

Запустим сценарий на выполнение. В качестве результата выполнения кода открылось окно с построенным графиком (рис. 2) и в рабочем каталоге появились файлы с графиками в форматах .eps, .png. (рис. 3).

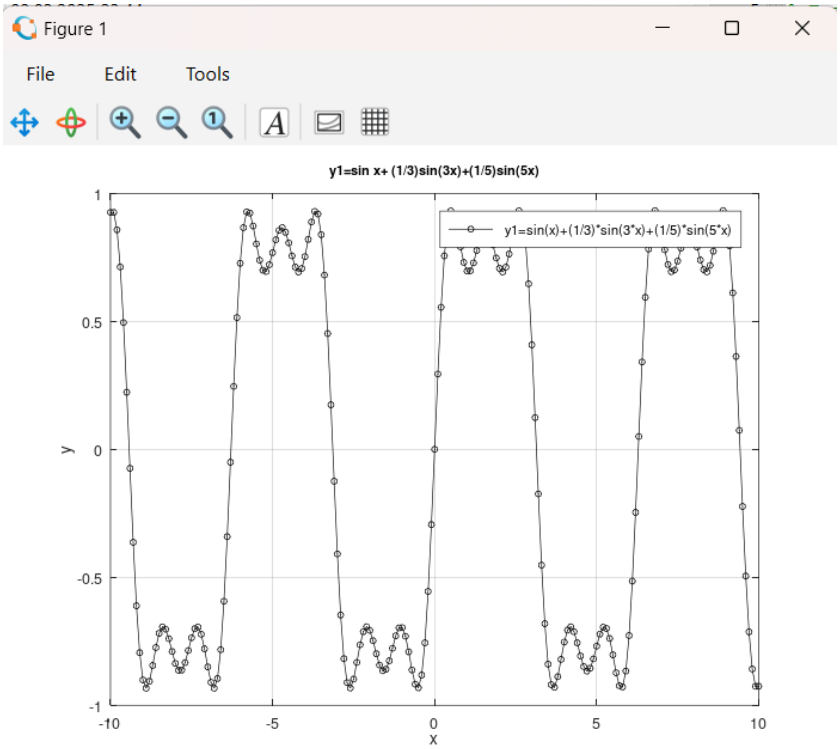

Рис. 2.2: Построенный график

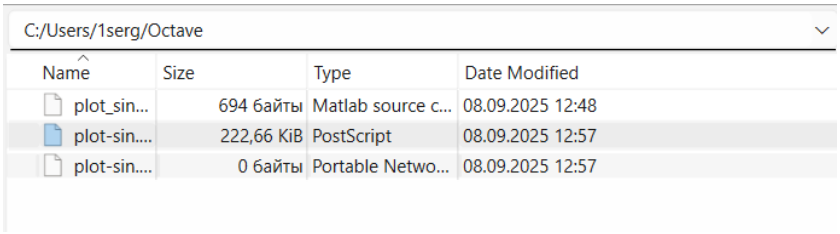

Рис. 2.3: Форматы .eps и .png

Сохраним сценарий под другим названием и изменим 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) (рис. 4)

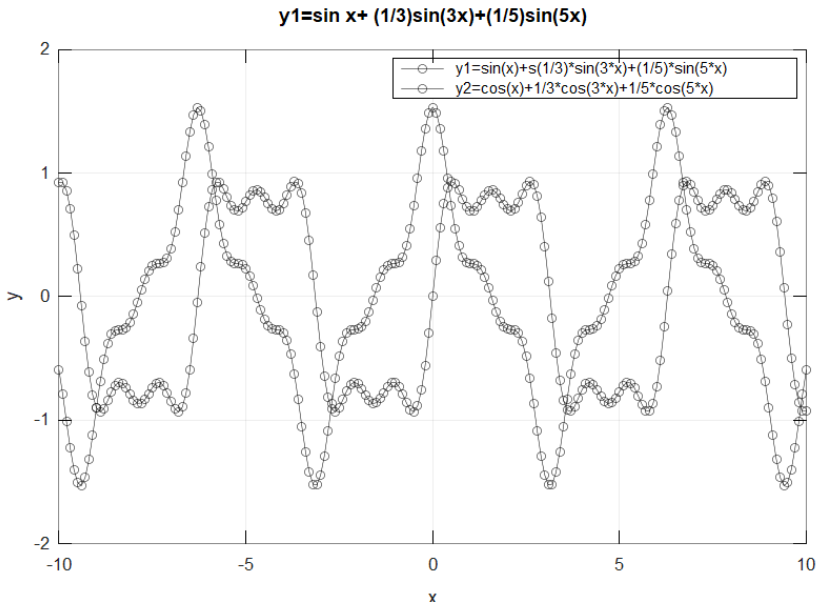

Рис. 2.4: График с cos

2.2 Разложение импульсного сигнала в частичный ряд Фурье

Создадим новый сценарий и сохраним его в рабочий каталог с именем meandr.m. В коде повторим листинг по построению графиков меандра. (рис. 5)

meandr.m

```
% meandr.m
% количество отсчетов (гармоник):
N=8;
% частота дискретизации:
t=-1:0.01:1;
% значение амплитуды:
A=1;
% период:
T=1;
% амплитуда гармоник
nh=(1:N)*2-1;
% массив коэффициентов для ряда, заданного через cos:
Am=2/pi ./ nh;
Am(2:2:end) = -Am(2:2:end);
% массив гармоник:
harmonics=cos(2 * pi * nh' * t/T);
% массив элементов ряда:
s1=harmonics.*repmat(Am',1,length(t));
% Суммирование ряда:
s2=cumsum(s1);
% Построение графиков:
for k=1:N
subplot(4,2,k)
plot(t, s2(k,:))
end
% Экспорт рисунка в файл .png:
print("meandr.png");
```

Рис. 2.5: Код meandr.m

Экспортируем полученный график в файл в формате .png. (рис. 6)

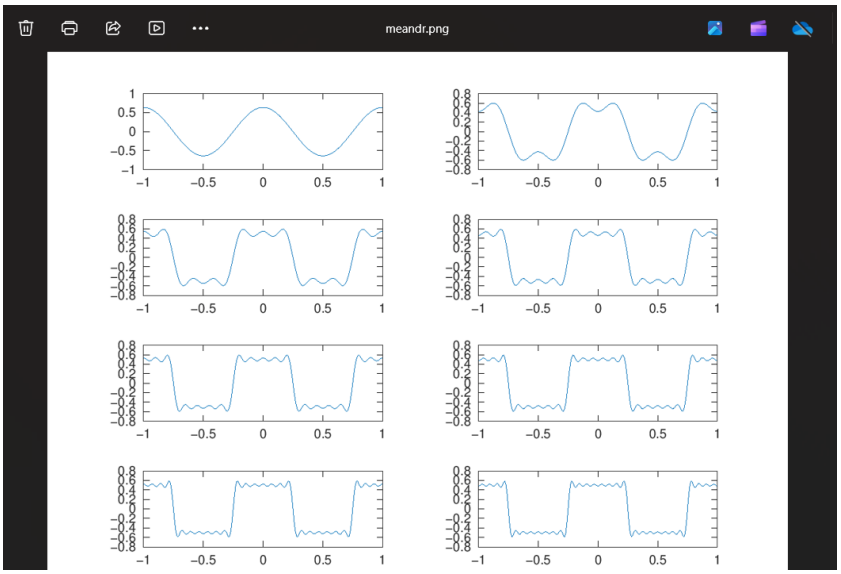

Рис. 2.6: Файл png

Скорректируем код для реализации меандра через синусы. Получим соответствующие графики. (рис. 7)

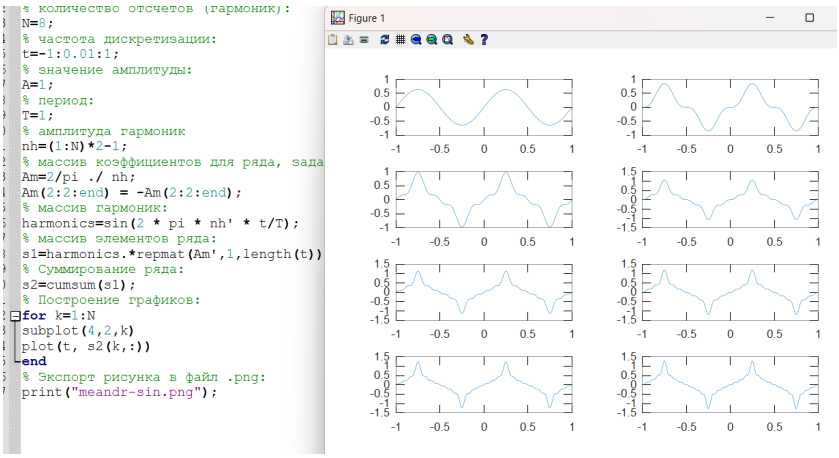

Рис. 2.7: Меандр через синусы

2.3 Определение спектра и параметров сигнала

В рабочем каталоге создадим каталог spectre1 и в нём новый сценарий с именем, spectre.m. В коде повторим листинг по построению сигналов разной частоты. Получим график. (рис. 8)

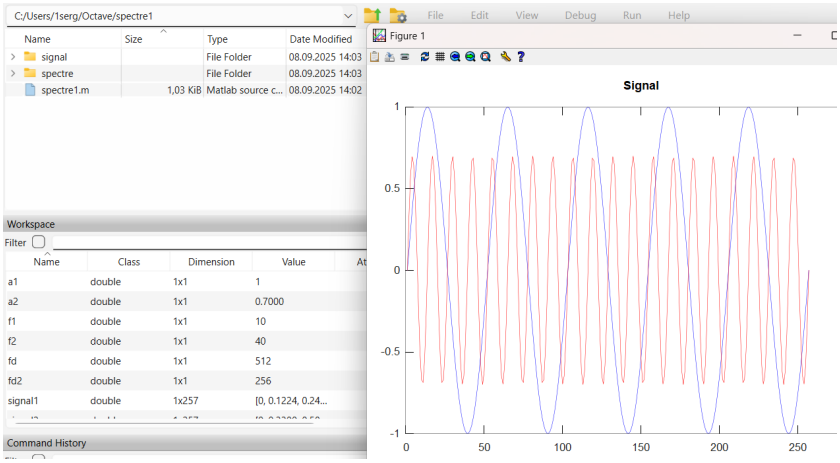

Рис. 2.8: График сигналов разной частоты

Далее добавим код для нахождения спектров сигналов с помощью быстрого преобразования Фурье и получим график. (рис. 9)

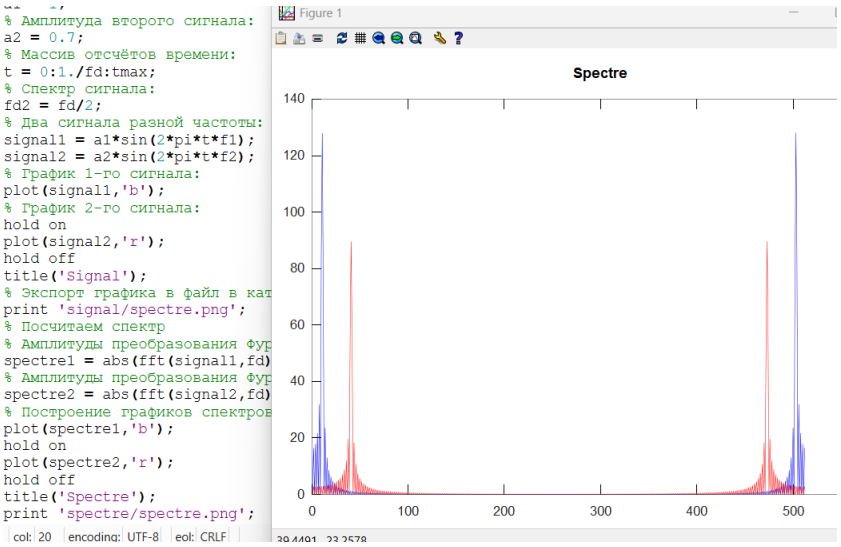

Рис. 2.9: График спектра сигнала

Скорректируем график спектра: отбросим дублирующие отрицательные частоты, а также примем в расчёт то, что на каждом шаге вычисления быстрого преобразования Фурье происходит суммирование амплитуд сигналов. (рис. 10)

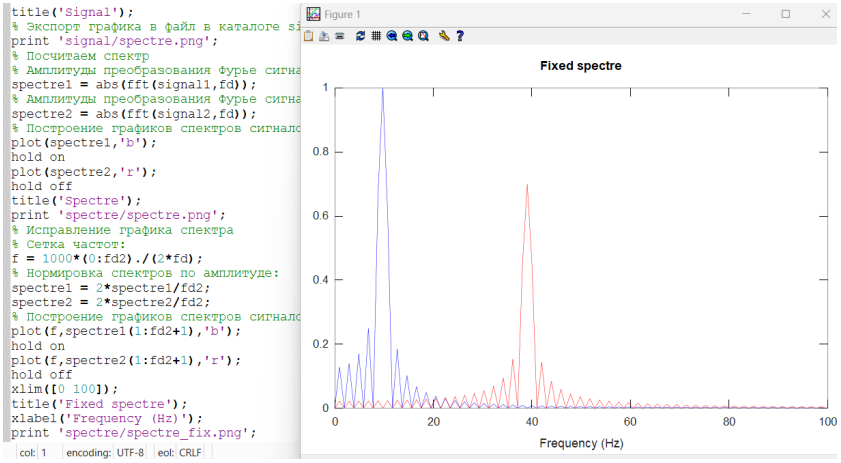

Рис. 2.10: Скорректированный график спектра

Найдем спектр суммы рассмотренных сигналов, создав каталог spectr_sum и файл в нём spectre_sum.m. В результате получился аналогичный предыдущему результат, т.е. спектр суммы сигналов равен сумме спектров сигналов, что вытекает из свойств преобразования Фурье. (рис. 11)

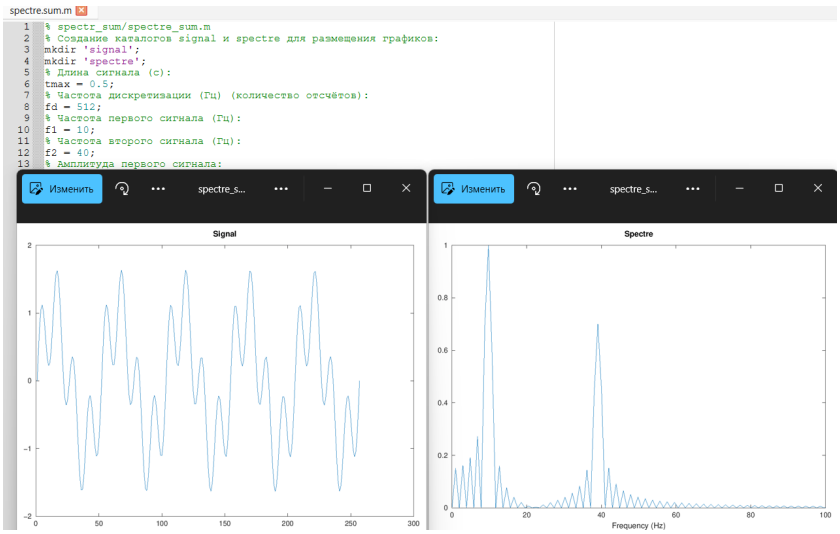

Рис. 2.11: Графики суммарного сигнала

2.4 Амплитудная модуляция

В рабочем каталоге создадим каталог modulation и в нём новый сценарий с именем am.m. Добавим в файле am.m код из листинга. (рис. 12)

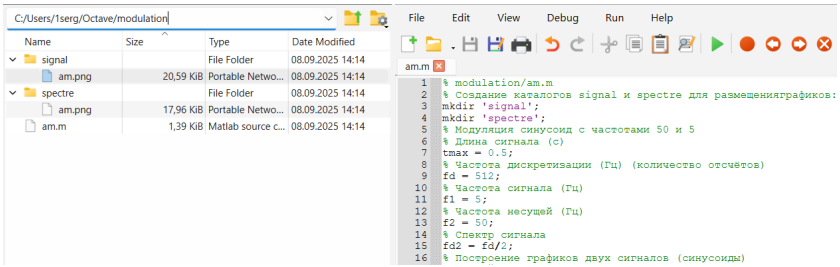

Рис. 2.12: Каталог modulation

В результате получаем, что спектр произведения представляет собой свёртку спектров (рис. 13)

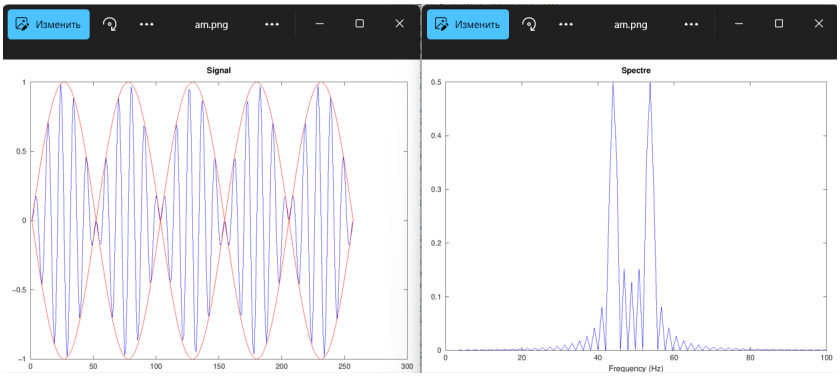

Рис. 2.13: Графики амплитудной модуляции

2.5 Кодирование сигнала. Исследование свойства самосинхронизации сигнала

В рабочем каталоге создадим 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. 14)

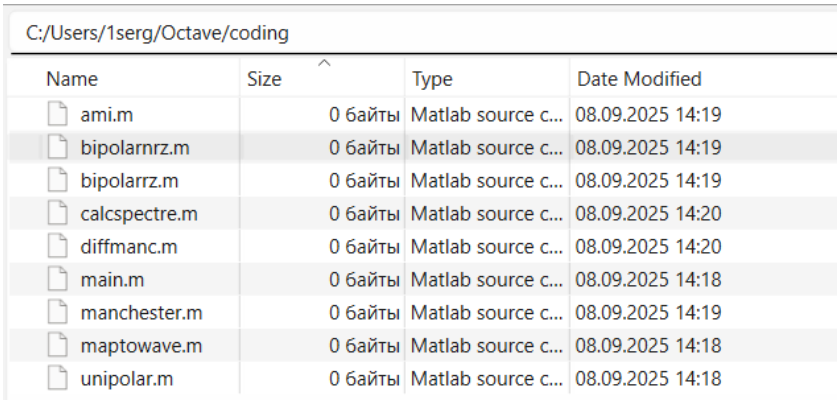

Рис. 2.14: Создание файлов

В окне интерпретатора команд проверим, установлен ли пакет расширений signal: » pkg list. Видим, что он установлен. (рис. 15)

```
         quaternion  |   2.4.0 | C:\Program Files\GNU
0
           queueing  |   1.2.8 | C:\Program Files\GNU
             signal  |   1.4.6 | C:\Program Files\GNU
```

Рис. 2.15: Проверка пакета

В файле main.m подключим пакет signal и скопируем программу из листинга. (рис. 16)

main.m

```
% 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
data_sync=[0 0 0 0 0 0 0 1 1 1 1 1 1 1 1];
% Входная кодовая последовательность для построен
data_spectre=[0 1 0 1 0 1 0 1 0 1 0 1 0 1];
% Создание каталогов signal, sync и spectre для р
mkdir 'signal';
```

Рис. 2.16: Код main.m

В файле maptowave.m пропишем функцию, которая по входному битовому потоку строит график сигнала. (рис. 17)

main.m | maptowave.m

```
% coding/maptowave.m
function wave=maptowave(data)
data=upsample(data,100);
wave=filter(5*ones(1,100),1,data);
```

Рис. 2.17: Файл maptowave.m

В файлах unipolar.m, ami.m, bipolarnrz.m, bipolarrz.m, manchester.m, diffmanc.m пропиштv соответствующие функции преобразования кодовой последовательности data с вызовом функции maptowave для построения соответствующего графика. (рис. 18)

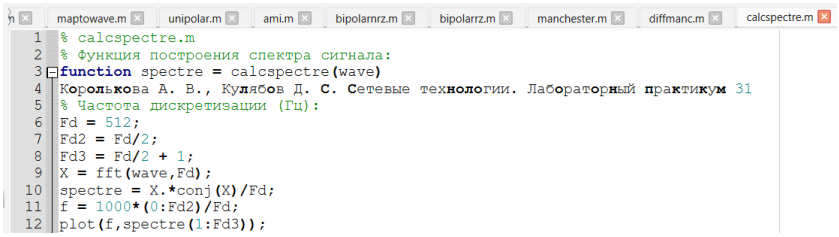

Рис. 2.18: Прописывание функций

Запустим главный скрипт main.m. В каталоге signal получены файлы с графиками кодированного сигнала, в каталоге sync — файлы с графиками, иллюстрирующими свойства самосинхронизации, в каталоге spectre — файлы с графиками спектров сигналов. (рис. 19)

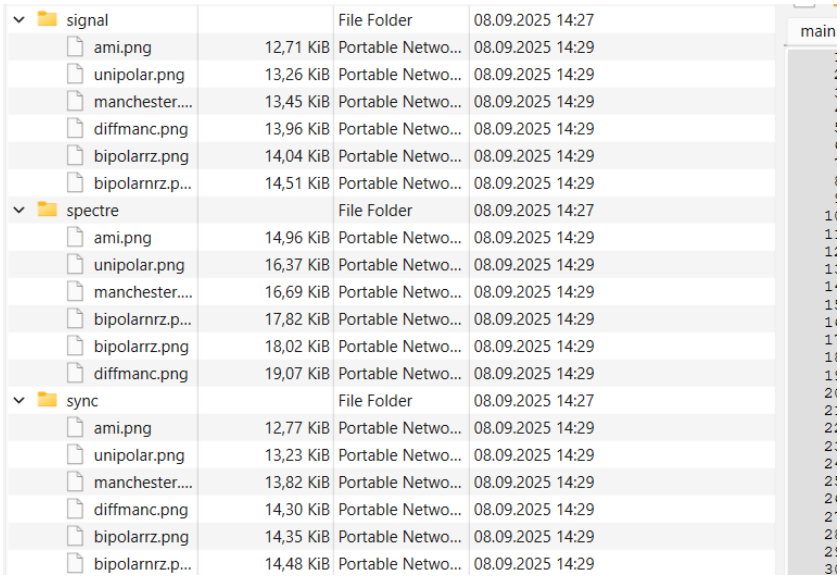

Рис. 2.19: Полученные графики

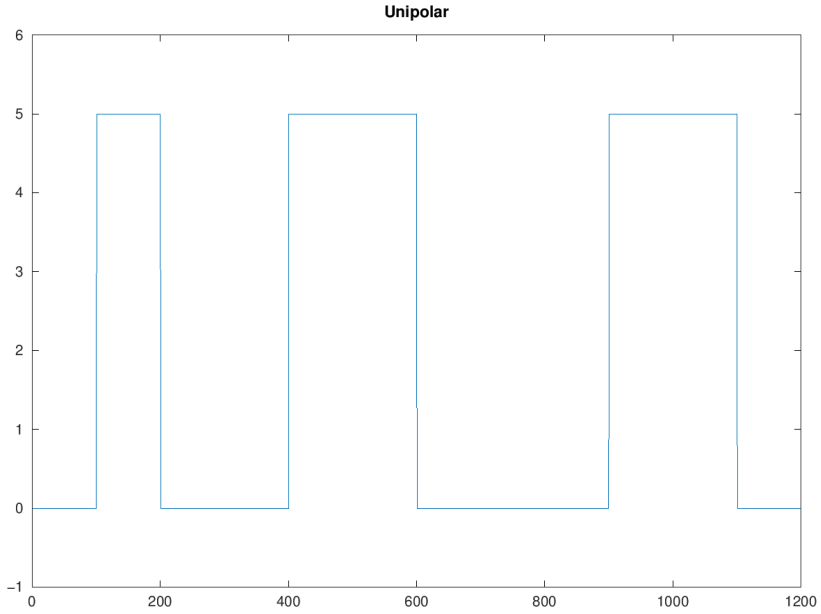

Рис. 2.20: Униполярное кодирование

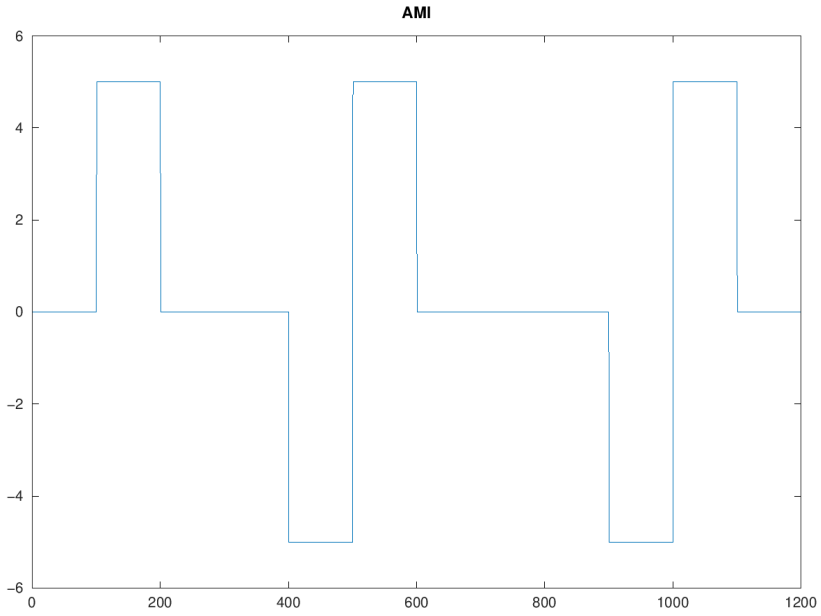

Рис. 2.21: Кодирование AMI

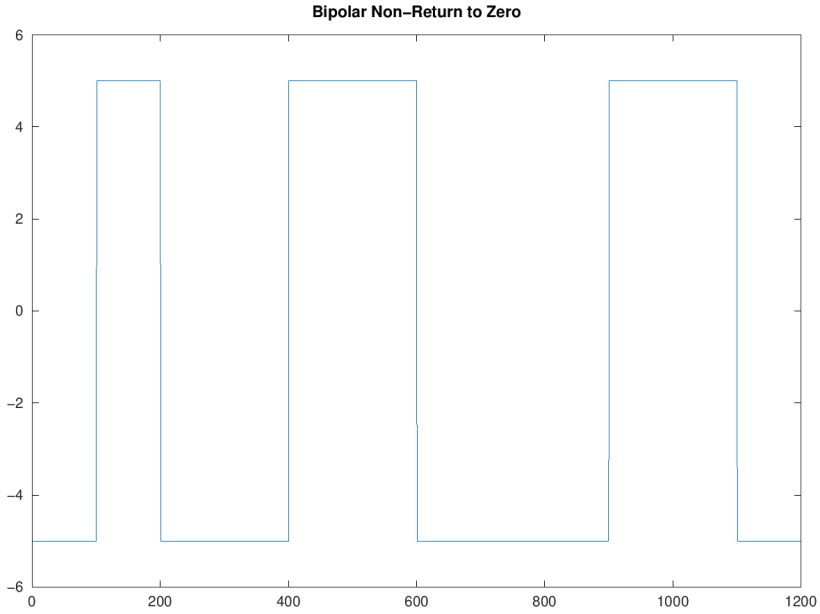

Рис. 2.22: Кодирование NRZ

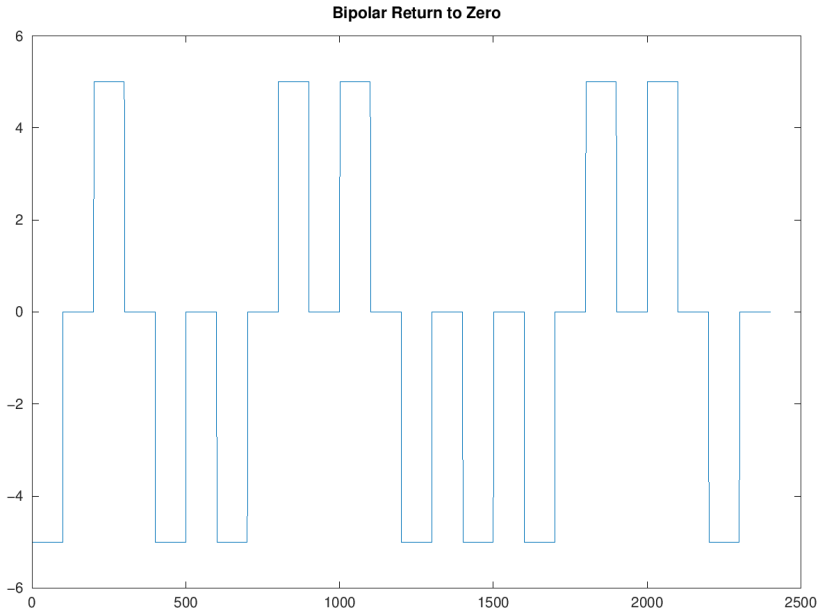

Рис. 2.23: Кодирование RZ

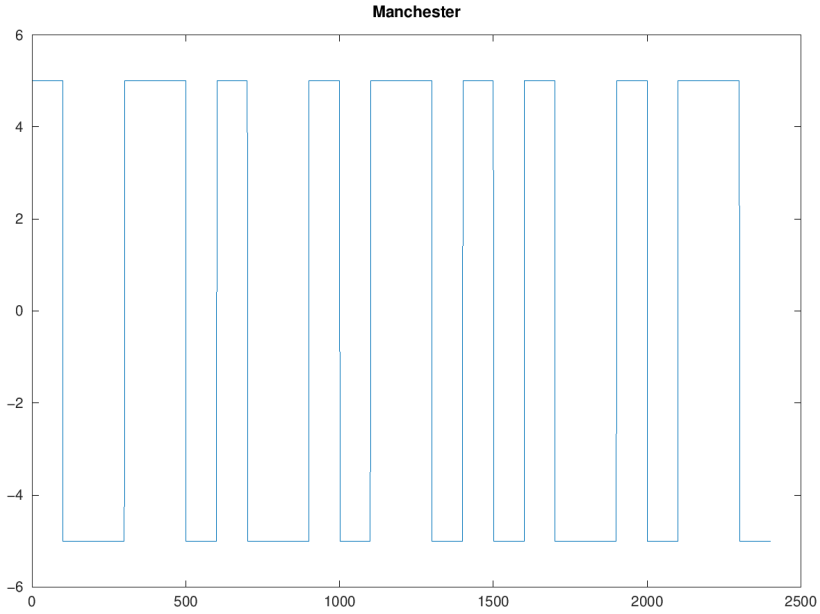

Рис. 2.24: Манчестерское кодирование

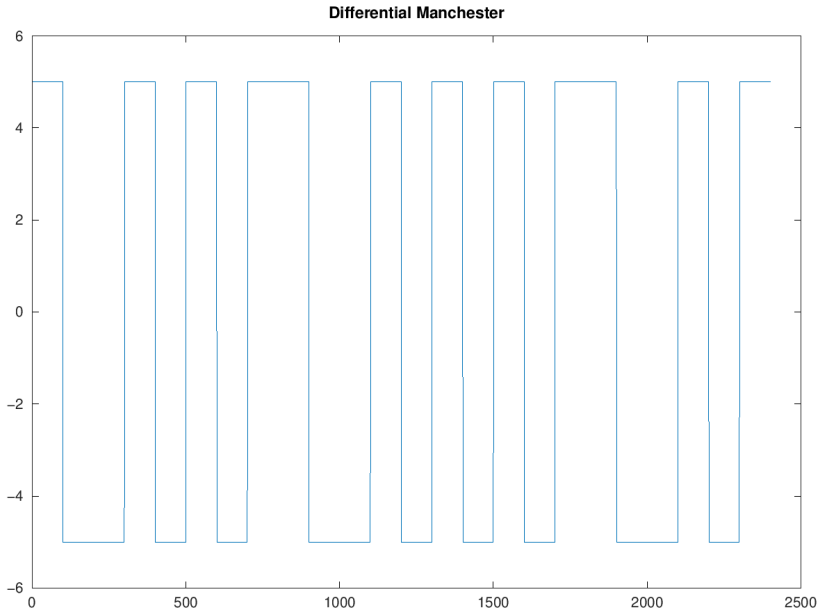

Рис. 2.25: Дифференциальное манчестерское кодирование

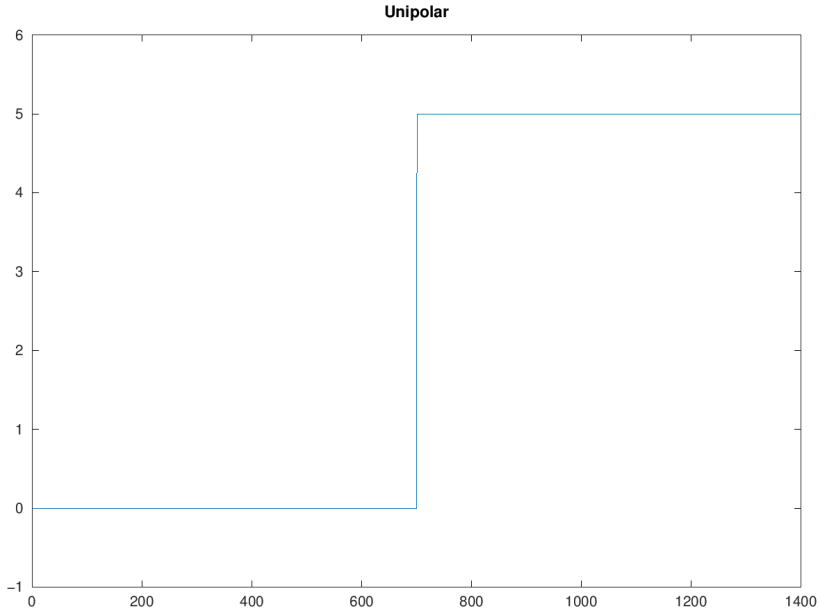

Рис. 2.26: Униполярное кодирование: нет самосинхронизации

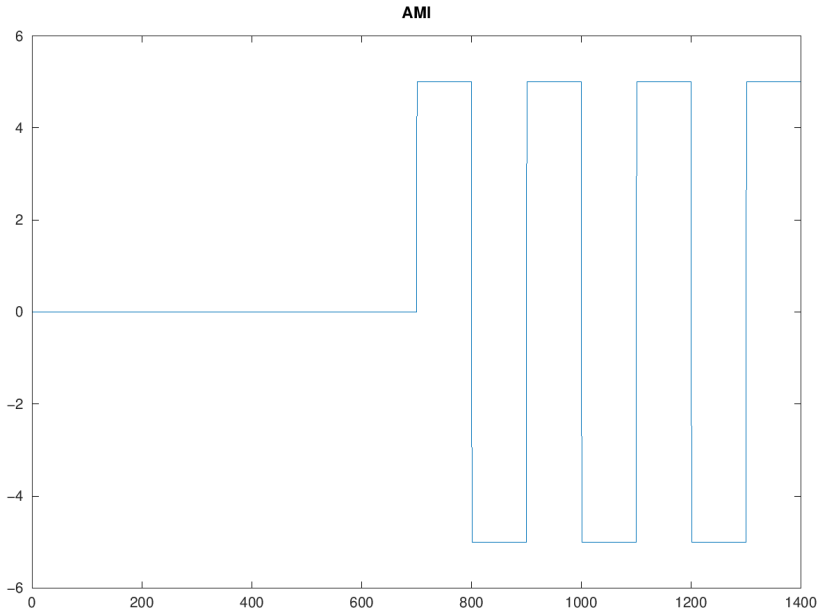

Рис. 2.27: Кодирование AMI: самосинхронизация при наличии сигнала

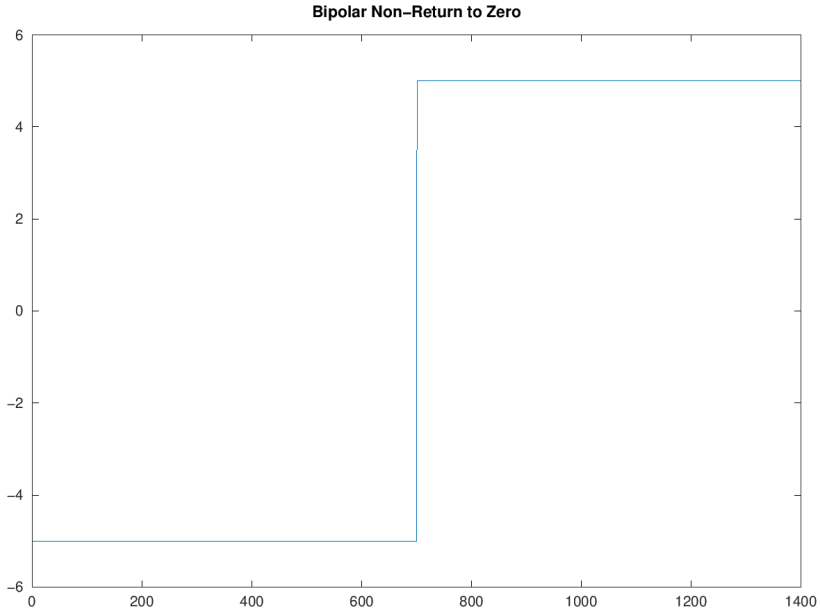

Рис. 2.28: Кодирование NRZ: нет самосинхронизации

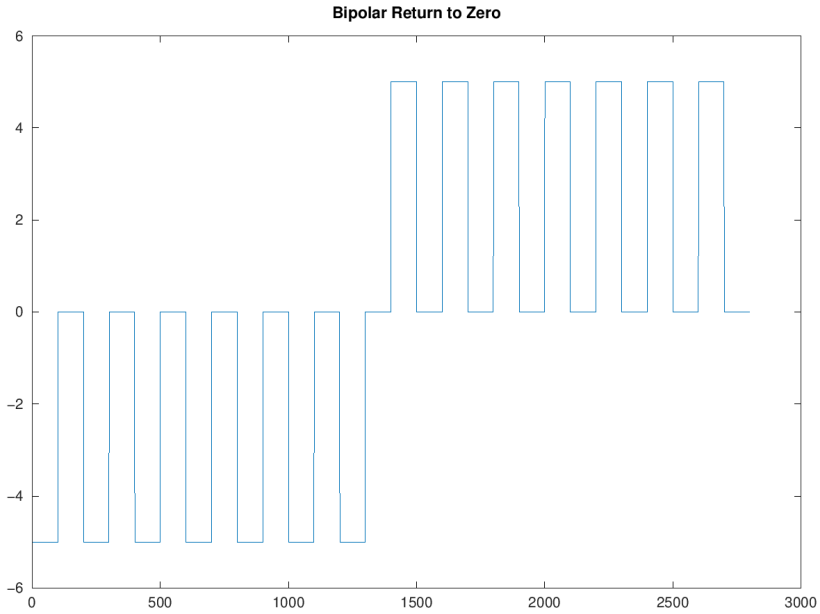

Рис. 2.29: Кодирование RZ: есть самосинхронизация

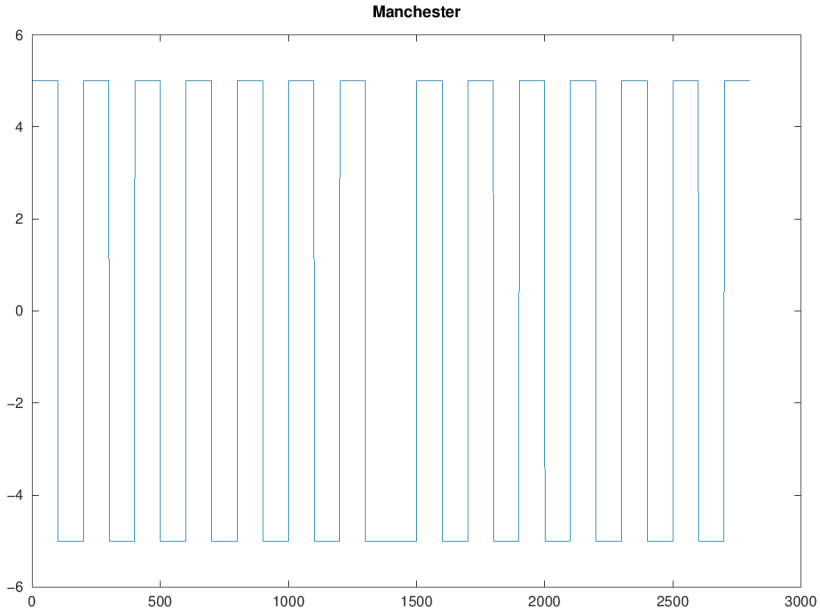

Рис. 2.30: Манчестерское кодирование: есть самосинхронизация

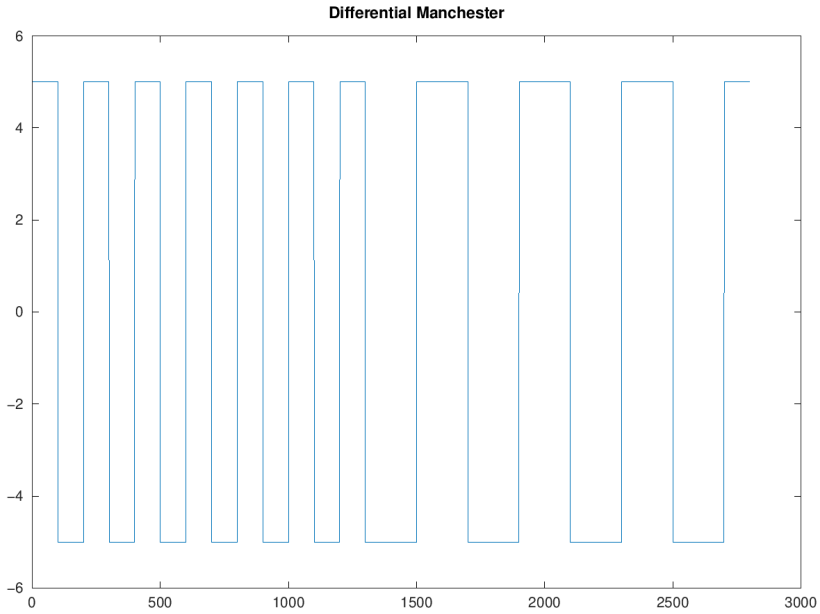

Рис. 2.31: Дифференциальное манчестерское кодирование: есть самосинхронизация

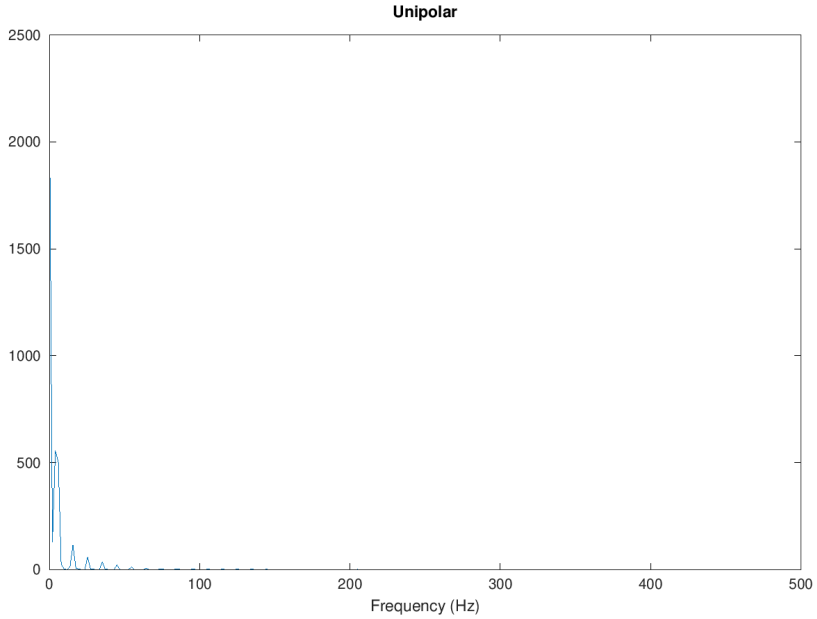

Рис. 2.32: Униполярное кодирование: спектр сигнала

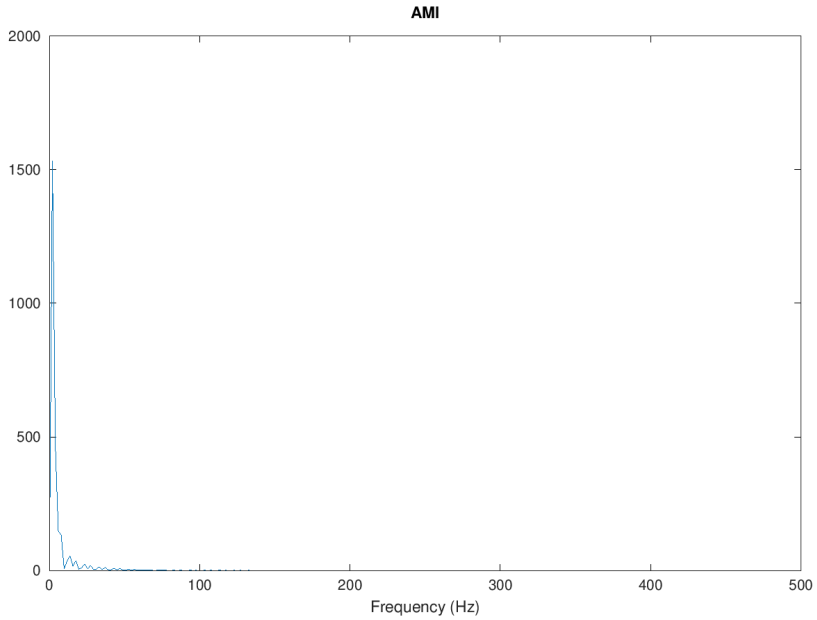

Рис. 2.33: Кодирование AMI: спектр сигнала

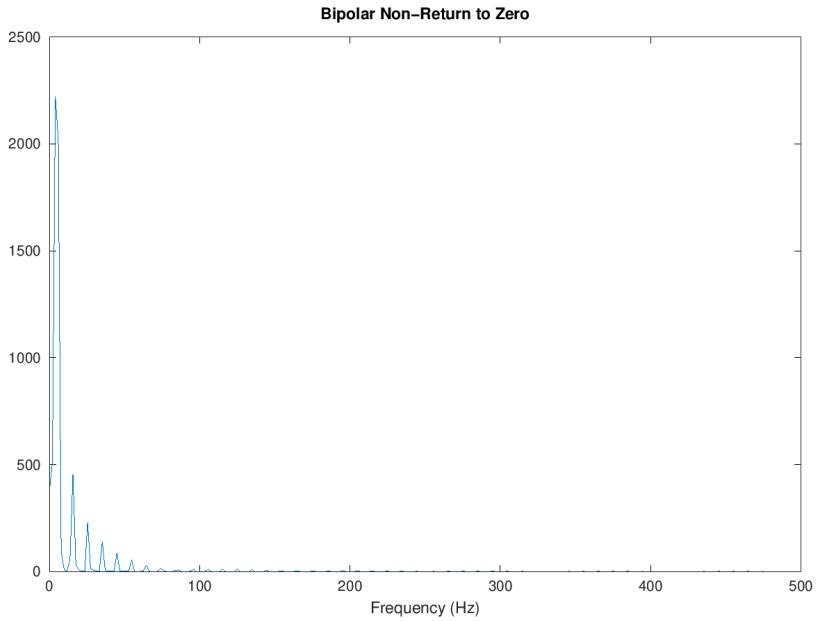

Рис. 2.34: Кодирование NRZ: спектр сигнала

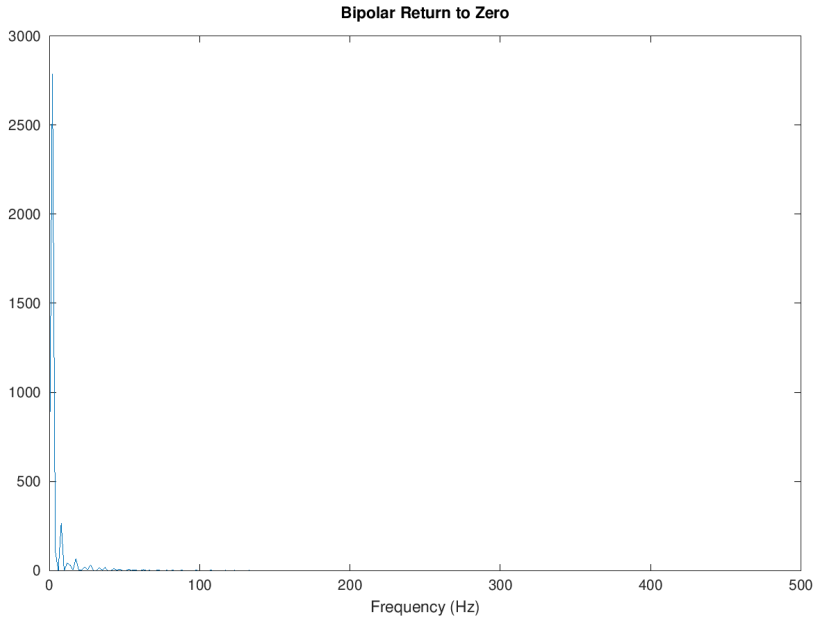

Рис. 2.35: Кодирование RZ: спектр сигнала

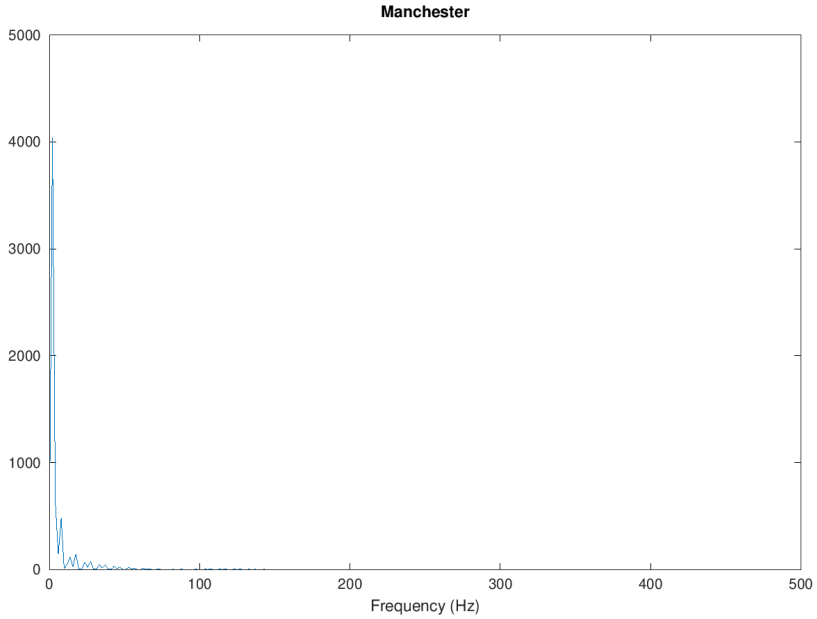

Рис. 2.36: Манчестерское кодирование: спектр сигнала

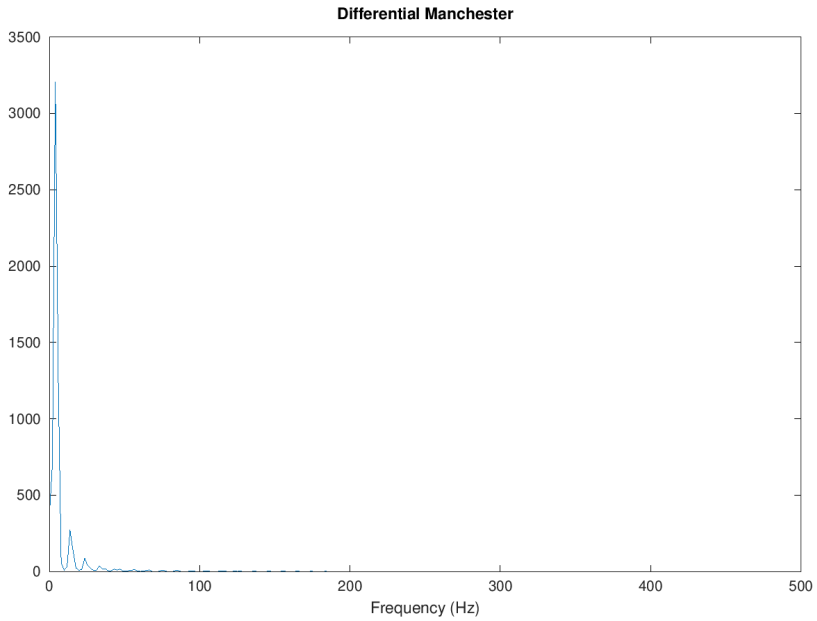

Рис. 2.37: Дифференциальное манчестерское кодирование: спектр сигнала

2.6 Задание с другой частотой дискретизации.

1) Что будет, если взять частоту дискретизации меньше 80 Гц?

При снижении частоты дискретизации ниже 80 Гц высокочастотные составляющие сигнала могут быть утеряны, что приведёт к искажению результатов спектрального анализа.

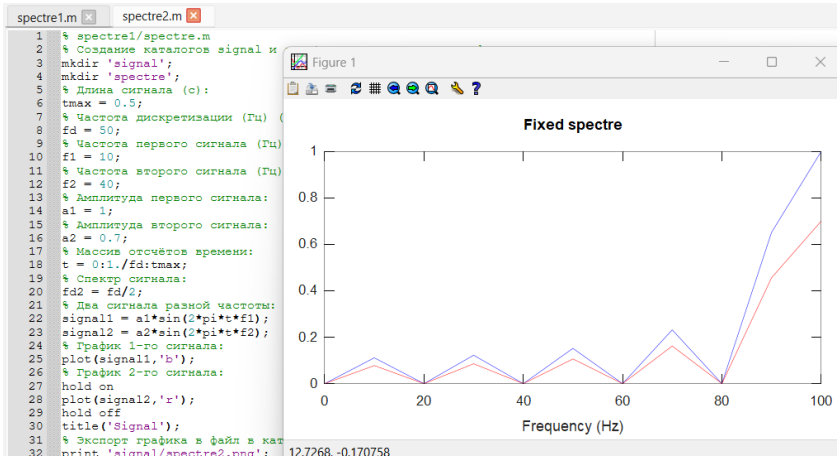

Рис. 2.38: Задание с другой частотой дискретизации

3 Выводы

В рамках лабораторной работы мы изучили методов кодирования и модуляции сигналов с помощью высокоуровнего языка программирования Octave. Определили спектра и параметров сигнала. Продемонстрировали принципы модуляции сигнала на примере аналоговой амплитудной модуляции. Исследовали свойства самосинхронизации сигнала.