Лабораторная работа №1

Методы кодирования и модуляция сигналов

Лисовская Арина Валерьевна

Содержание

Список иллюстраций

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 Задание

1. Отработать навыки построения графиков функций в Octave.
2. Реализовать разложение меандра в ряд Фурье через синусы и косинусы.
3. Выполнить спектральный анализ сигналов с помощью быстрого преобразования Фурье (БПФ).
4. Промоделировать процесс амплитудной модуляции (АМ).
5. Реализовать функции для различных методов линейного кодирования (AMI, NRZ, RZ, Manchester, Diff. Manchester).
6. Проанализировать спектры полученных кодов и их свойства самосинхронизации.

3 Выполнение лабораторной работы

3.1 Построение графиков и разложение в ряд Фурье

В начале работы я изучила базовые функции построения графиков. На первом этапе были построены две сложные функции, представляющие собой суммы гармоник синуса и косинуса (рис. 3.1).

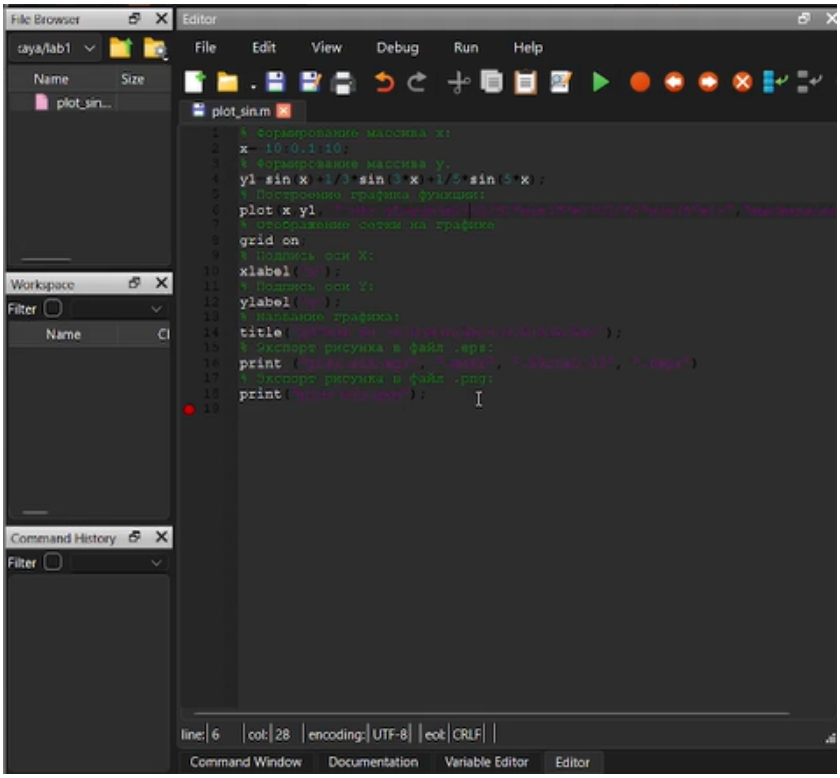


Рис. 3.1: Графики функций y1 и y2 в одном окне

Для генерации периодического сигнала типа «меандр» я создала скрипт

`meandr.m`. В этом коде реализовано разложение сигнала в ряд Фурье с использованием косинусоид (рис. 3.2).

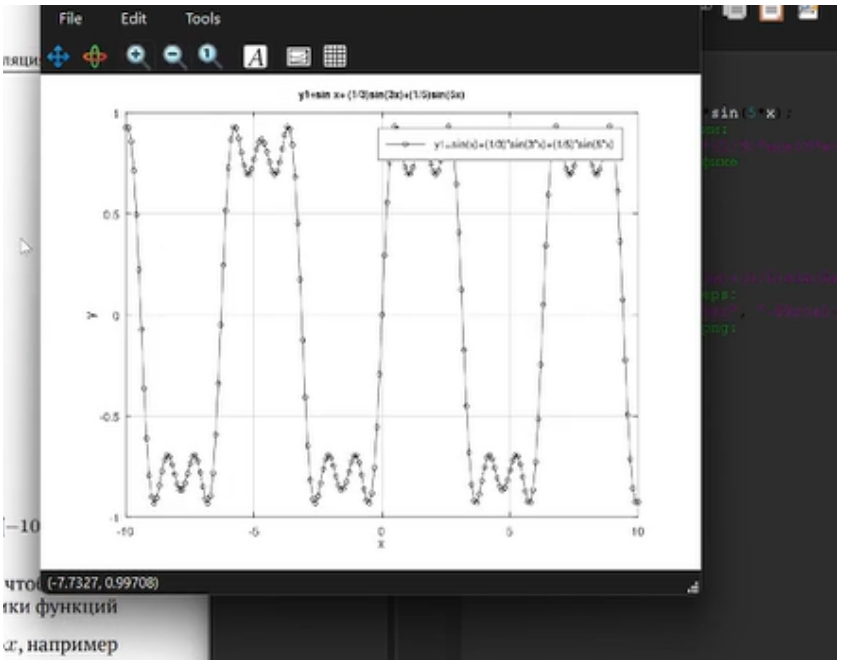


Рис. 3.2: Код в редакторе Octave — meandr.m

В качестве альтернативного способа формирования того же сигнала был написан скрипт `meandr_sin.m`. В данном случае разложение меандра производится через синусоидальные составляющие (рис. 3.3).

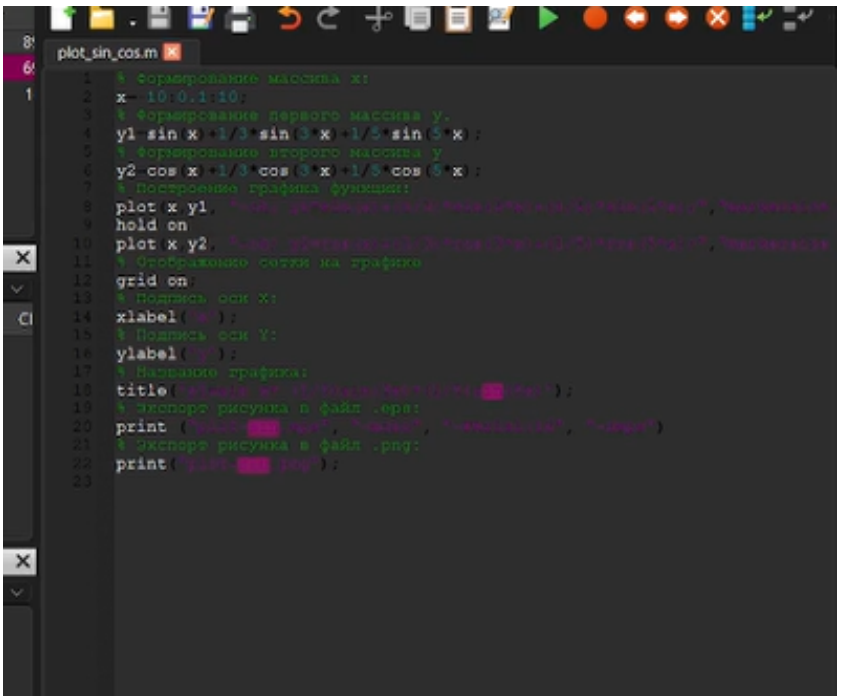


Рис. 3.3: Код в редакторе Octave — meandr_sin.m

3.2 Спектральный анализ сигналов

Для изучения частотных характеристик я приступила к созданию скрипта `spectre.m`. В нем задаются параметры дискретизации и формируются два синусоидальных сигнала с частотами 10 Гц и 40 Гц (рис. 3.4).

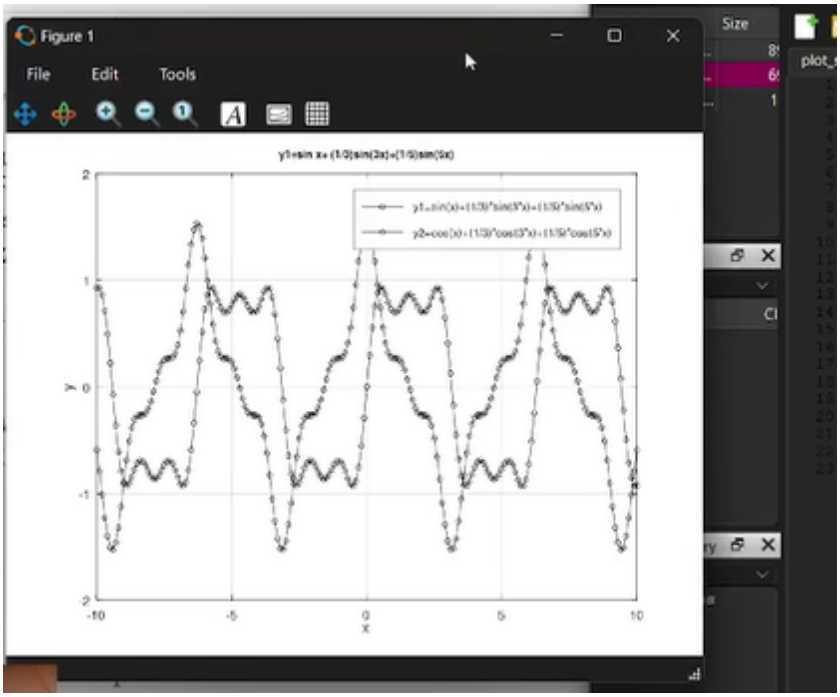


Рис. 3.4: Код в редакторе Octave — spectre.m

После генерации сигналов я визуализировала их во временной области. Это позволило увидеть разницу в периодах колебаний синусоид разной частоты (рис. 3.5).

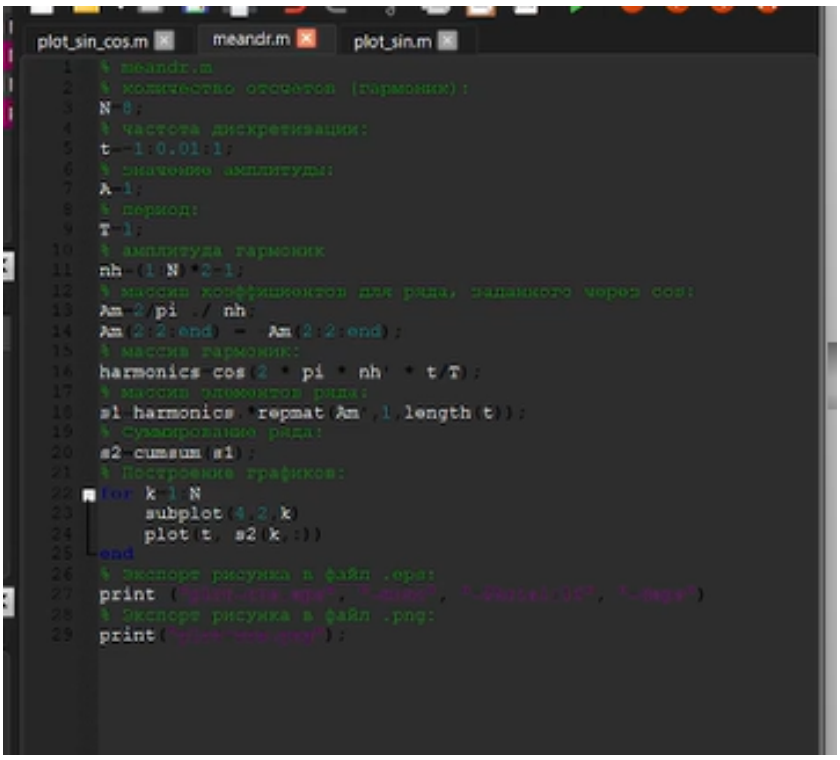


Рис. 3.5: График двух синусоидальных сигналов

Применив встроенную функцию быстрого преобразования Фурье `fft()`, я получила первичный амплитудный спектр. На данном этапе спектр содержит избыточную информацию и требует нормировки (рис. 3.6).

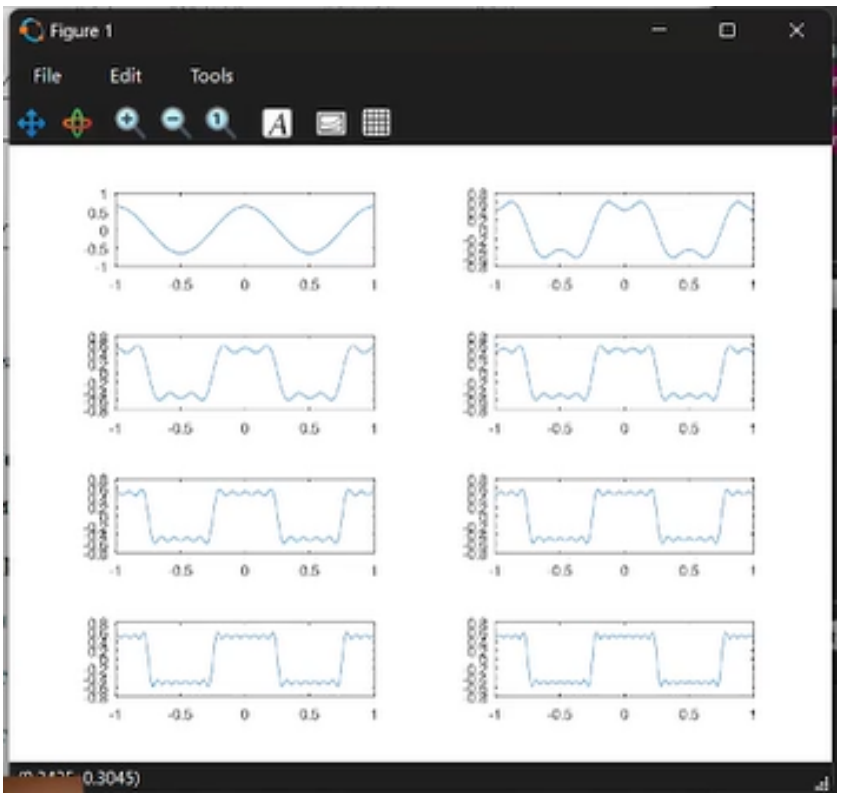


Рис. 3.6: Необработанный спектр сигналов

После проведения коррекции спектра (отсечения отрицательных частот и нормировки амплитуд) были получены четкие пики на заданных частотах. На графике отчетливо видны составляющие 10 Гц и 40 Гц (рис. 3.7).

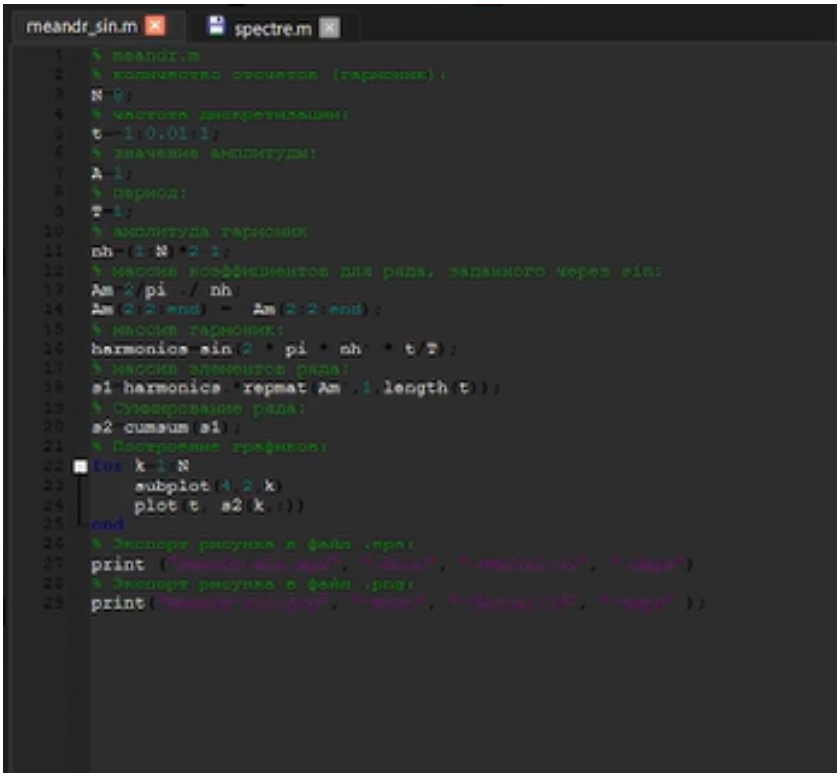


Рис. 3.7: Исправленный спектр сигналов

Далее я исследовала свойство аддитивности преобразования Фурье. Для этого был написан скрипт `spectre_sum.m`, рассчитывающий спектр суммы двух сигналов (рис. 3.8).

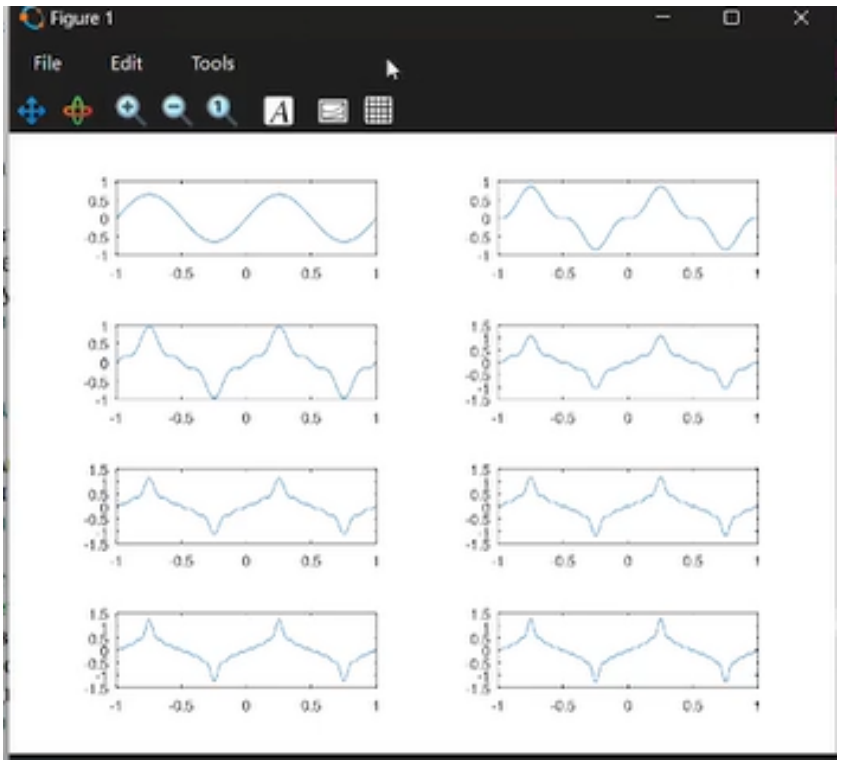
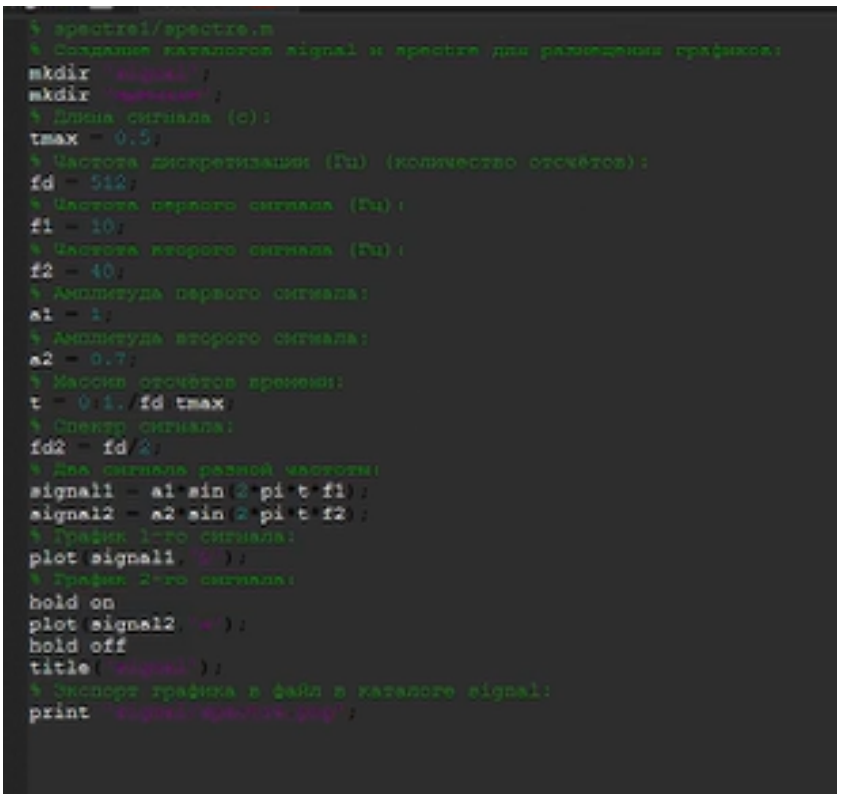


Рис. 3.8: Код в редакторе Octave — spectre_sum.m

Результаты вычислений подтвердили теорию: спектр суммы сигналов содержит те же частотные компоненты, что и исходные данные. Это наглядно представлено на графике амплитудного спектра суммарного сигнала (рис. 3.9).



```
% spectre1/spectre.m
% Создание каталогов signal и spectre для размещения графиков:
mkdir 'signal';
mkdir 'spectre';
% Длина сигнала (с):
tmax = 0.5;
% Частота дискретизации (Гц) (количество отсчётов):
fd = 512;
% Частота первого сигнала (Гц):
f1 = 10;
% Частота второго сигнала (Гц):
f2 = 40;
% Амплитуда первого сигнала:
a1 = 1;
% Амплитуда второго сигнала:
a2 = 0.7;
% Массив отсчётов времени:
t = 0:1./fd:tmax;
% Спектр сигнала:
fd2 = fd/2;
% Два сигнала разной частоты:
signal1 = a1*sin(2*pi*t*f1);
signal2 = a2*sin(2*pi*t*f2);
% График 1-го сигнала:
plot(signal1, 'b');
% График 2-го сигнала:
hold on
plot(signal2, 'r');
hold off
title('signal');
% Экспорт графика в файл в каталоге signal:
print 'signal/spectre.png';
```

Рис. 3.9: Спектр суммарного сигнала

Перед анализом спектра я также зафиксировала вид суммарного сигнала во временной области. Сложение двух синусоид дает сложную форму волны с биениями (рис. 3.10).

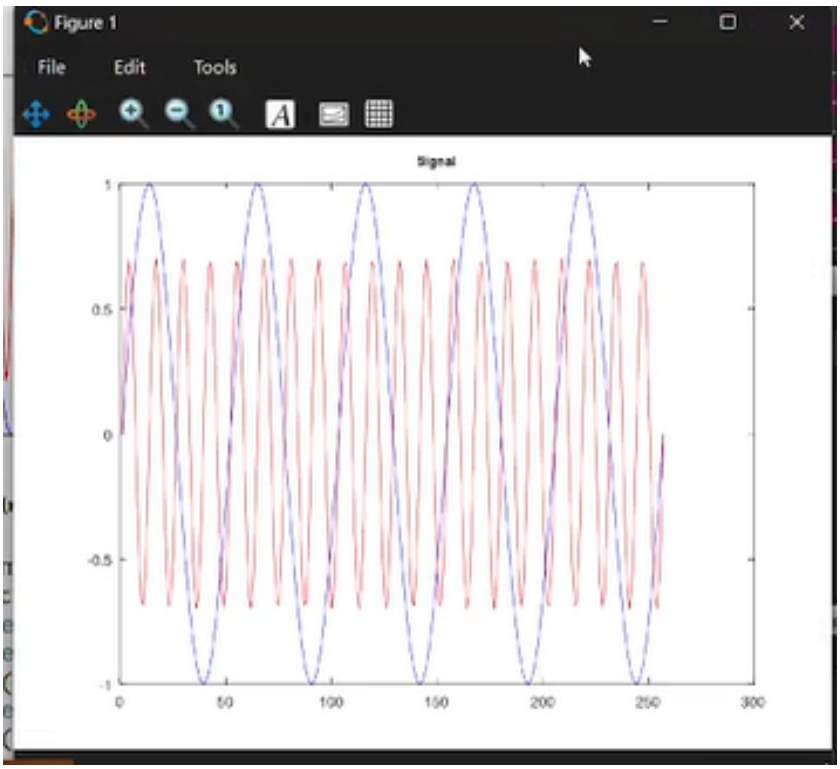


Рис. 3.10: График суммарного сигнала

3.3 Амплитудная модуляция

Следующим этапом стало моделирование аналоговой амплитудной модуляции в скрипте `am.m`. Низкочастотное сообщение переносится на высокую частоту несущей (рис. 3.11).

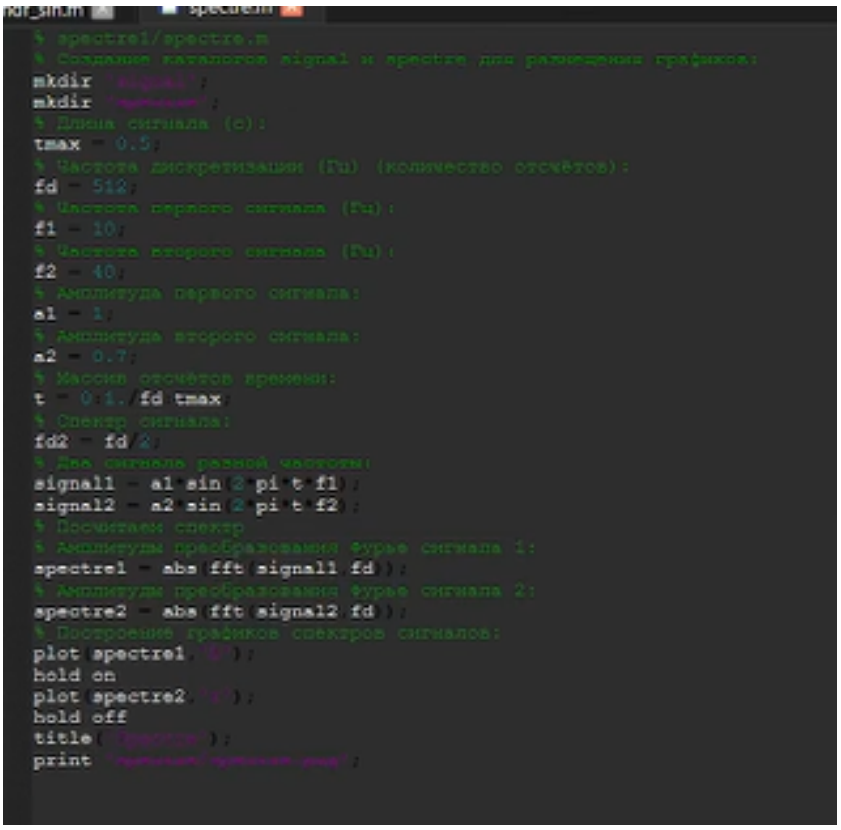


Рис. 3.11: Код в редакторе Octave — am.m

Анализ спектра АМ-сигнала показал наличие центральной несущей частоты 50 Гц. Также наблюдаются две боковые полосы на частотах 45 Гц и 55 Гц (рис. 3.12).

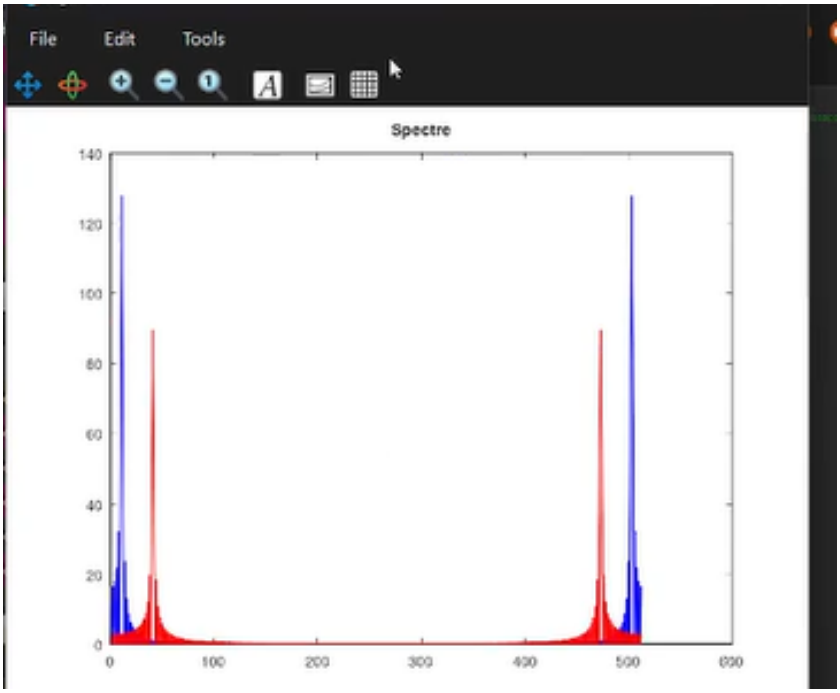


Рис. 3.12: Спектр АМ-сигнала

Визуализация АМ-сигнала во временной области демонстрирует, как амплитуда высокочастотного заполнения меняется вслед за огибающей. Красная линия на графике соответствует модулирующему сообщению (рис. 3.13).

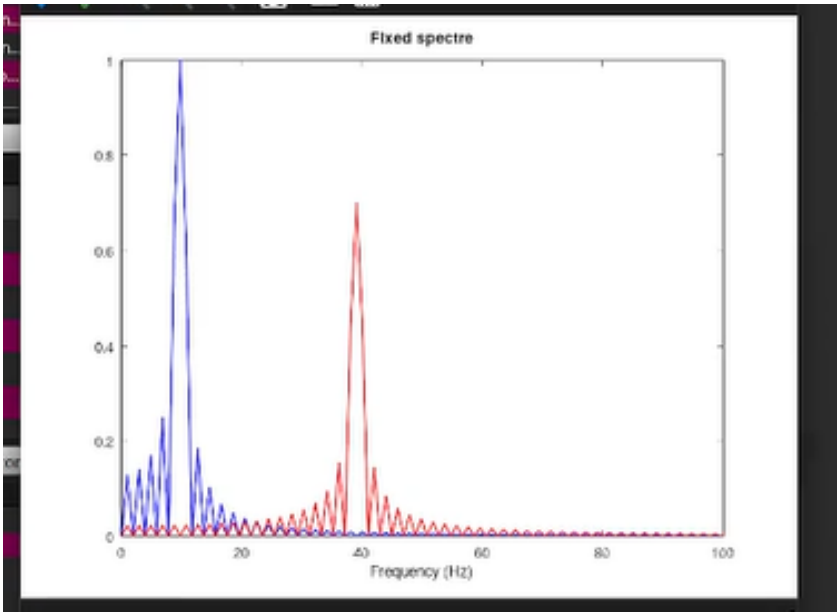


Рис. 3.13: АМ-сигнал и его огибающая

3.4 Методы цифрового кодирования

Перед началом реализации алгоритмов кодирования я проверила наличие необходимых библиотек в системе. Пакет `signal` обязателен для работы с функциями обработки сигналов в Octave (рис. 3.14).

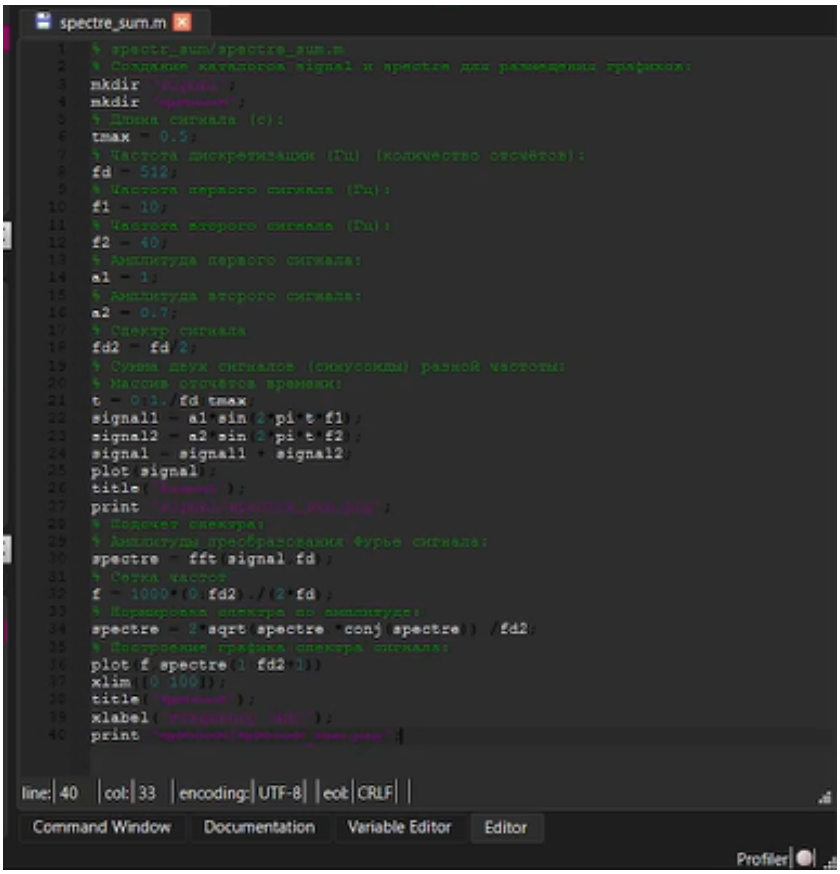


Рис. 3.14: Список установленных пакетов Octave

Для управления процессом кодирования и визуализации результатов был создан главный скрипт `main.m`. В нем инициализируется тестовая последовательность бит и вызываются соответствующие функции (рис. 3.15).

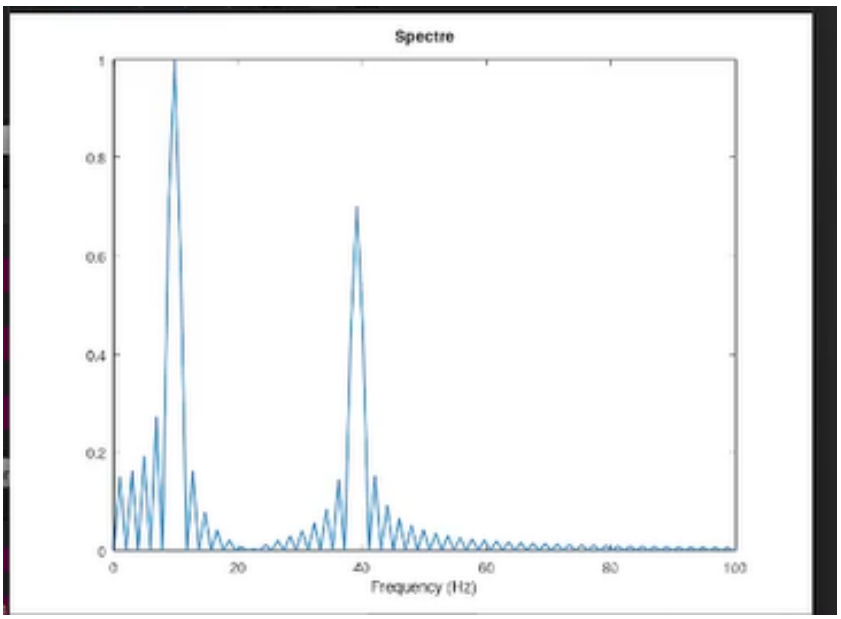


Рис. 3.15: Код в редакторе Octave — main.m (начало)

Для корректного отображения сигналов я использовала вспомогательную функцию `maptowave.m`. Она отвечает за преобразование дискретных уровней в непрерывный временной сигнал (рис. 3.16).

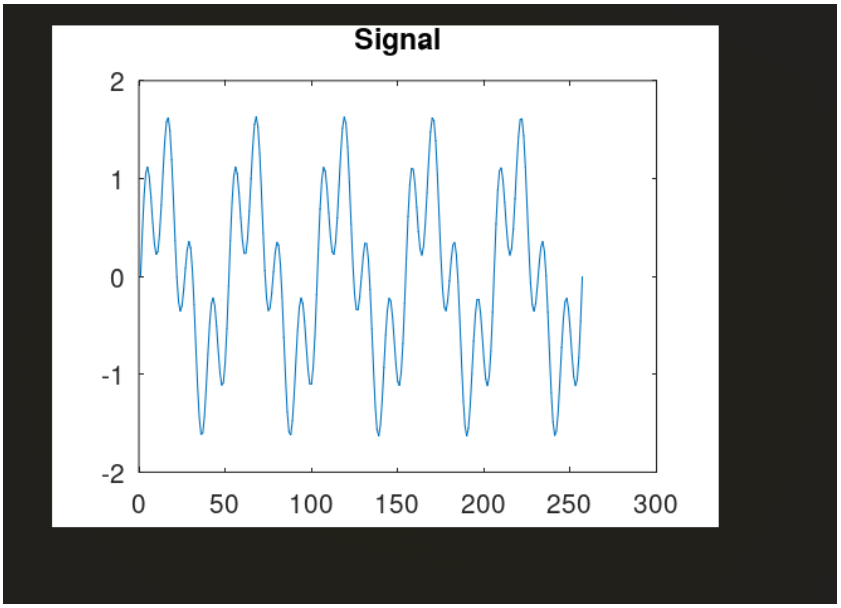


Рис. 3.16: Код в редакторе Octave — maptowave.m

Алгоритм кодирования AMI (Alternate Mark Inversion) был реализован в от-

дельном файле. Этот метод предполагает чередование полярности импульсов для каждой логической единицы (рис. 3.17).

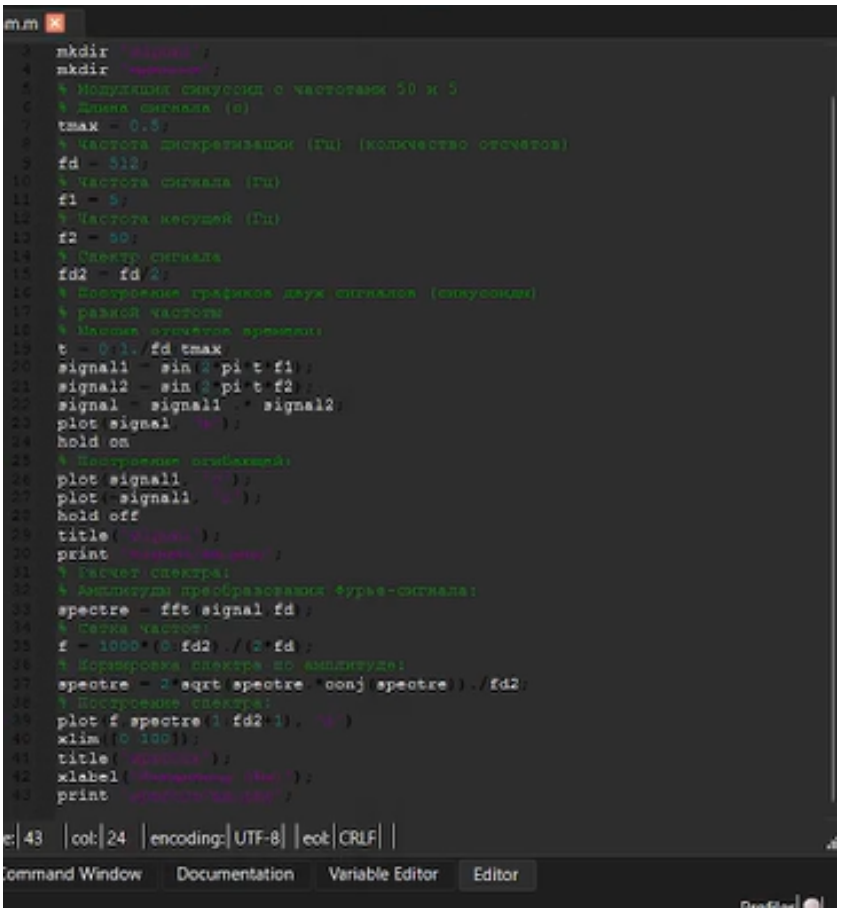


Рис. 3.17: Код в редакторе Octave — ami.m

Кодирование RZ (Return to Zero) реализовано в функции `bipolarrz.m`. Особенностью данного кода является возврат сигнала к нулевому уровню в середине каждого тактового интервала (рис. 3.18).

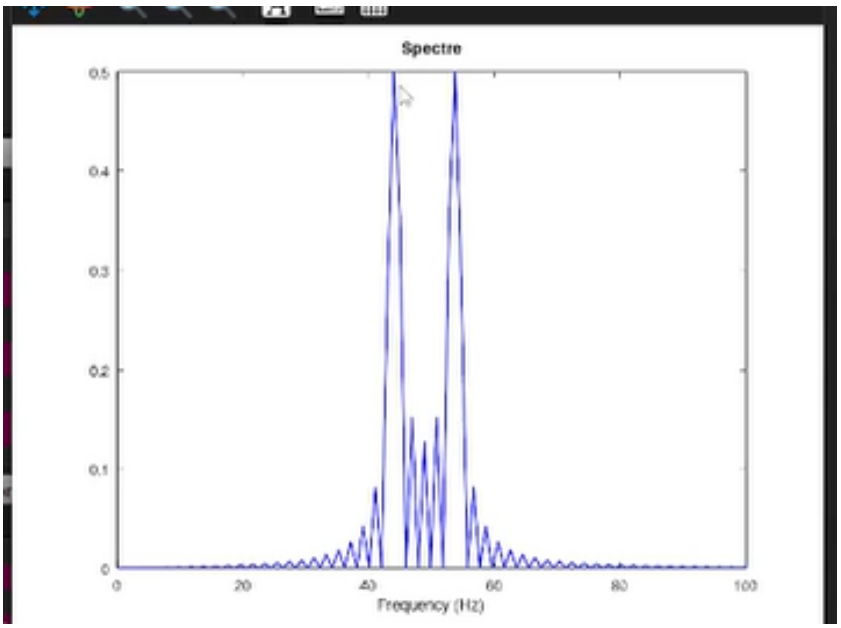


Рис. 3.18: Код в редакторе Octave — bipolarrz.m

Манчестерское кодирование обеспечивает обязательный перепад уровня в середине каждого бита. Это гарантирует надежное выделение тактовой частоты приемником (рис. 3.19).

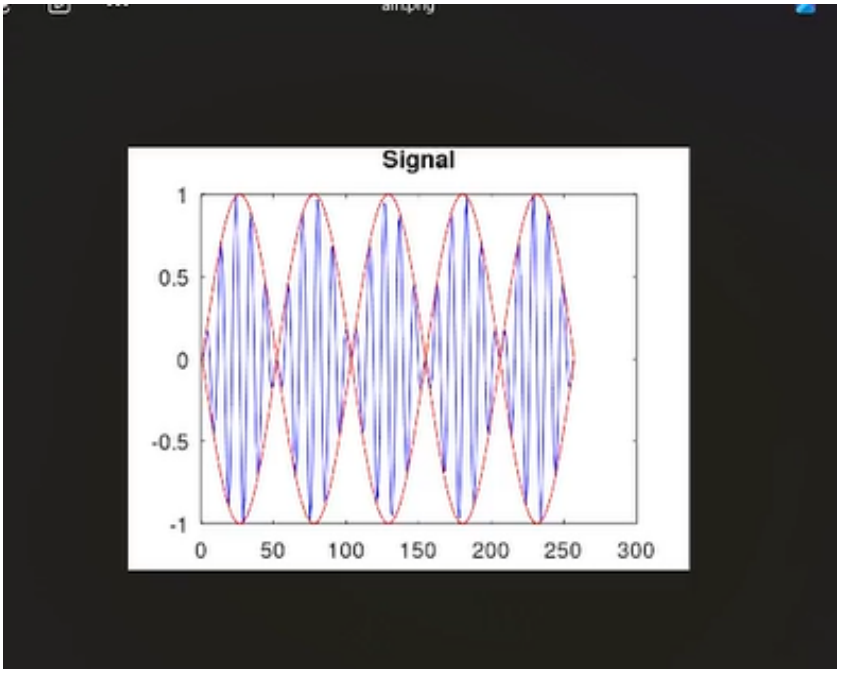


Рис. 3.19: Код в редакторе Octave — manchester.m

Дифференциальное манчестерское кодирование сочетает преимущества манчестерского кода с относительным способом передачи. Информацию несет наличие или отсутствие перехода в начале такта (рис. 3.20).

```
>> pkg list
Package Name          | Version | Installation directory
----------------------+---------+-----------------------
               audio  |   2.0.9 | ...\octave\packages\audio-2.0.9
              biosig  |   3.9.0 | ...\octave\packages\biosig-3.9.0
             cfitsio  |   0.0.7 | ...\octave\packages\cfitsio-0.0.7
               coder  |  1.10.1 | ...\octave\packages\coder-1.10.1
      communications  |   1.2.7 | ...\packages\communications-1.2.7
             control  |   4.1.3 | ...\octave\packages\control-4.1.3
      data-smoothing  |   1.3.0 | ...\packages\data-smoothing-1.3.0
            database  |   2.4.4 | ...\octave\packages\database-2.4.4
           dataframe  |   1.2.0 | ...\octave\packages\dataframe-1.2.0
               dicom  |   0.6.1 | ...\octave\packages\dicom-0.6.1
           financial  |   0.5.4 | ...\octave\packages\financial-0.5.4
             fl-core  |   1.0.2 | ...\octave\packages\fl-core-1.0.2
fuzzy-logic-toolkit   |   0.6.2 | ...\fuzzy-logic-toolkit-0.6.2
                  ga  |  0.10.4 | ...\share\octave\packages\ga-0.10.4
```

Рис. 3.20: Код в редакторе Octave — diffmanchester.m

3.5 Результаты моделирования кодов

На графике сигнала AMI видно, как единицы передаются импульсами разной полярности, а нули — отсутствием напряжения. Это позволяет избавиться от постоянной составляющей (рис. 3.21).

Рис. 3.21: График кодирования AMI

Сигнал NRZ является наиболее простым в реализации. На графике видно, что уровни напряжения соответствуют логическим состояниям на протяжении всего такта (рис. 3.22).

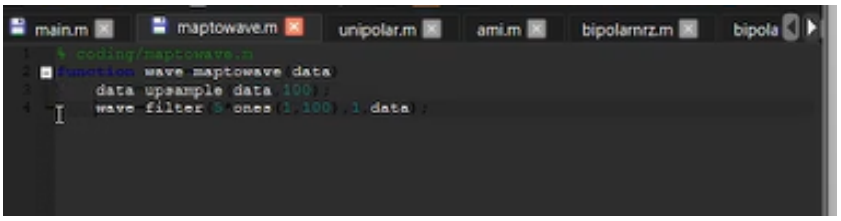


Рис. 3.22: График кодирования NRZ

При дифференциальном манчестерском кодировании форма сигнала зависит от предыдущего состояния. На рисунке показано формирование волновой формы для заданной битовой последовательности (рис. 3.23).

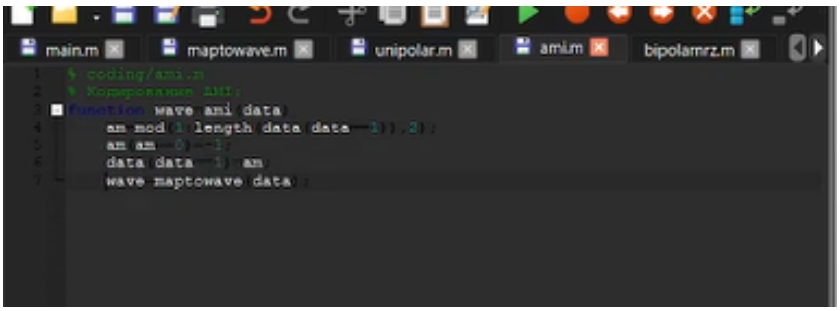


Рис. 3.23: График дифференциального манчестерского кодирования

Манчестерский код характеризуется постоянным изменением уровня. На графике видно, что каждая единица представлена переходом «вниз», а ноль — «вверх» (рис. 3.24).

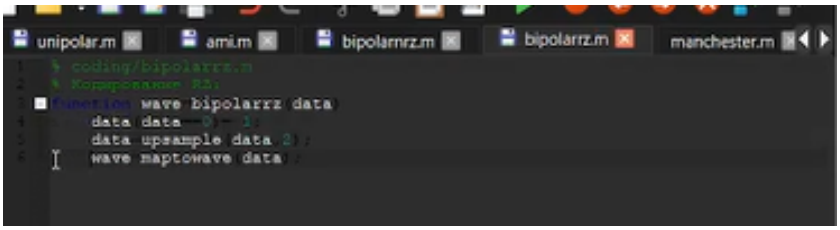


Рис. 3.24: График манчестерского кодирования

Для более детального изучения структуры дифференциального манчестерского кода я вывела график в увеличенном масштабе. Это позволяет точнее проанализировать моменты смены фазы (рис. 3.25).

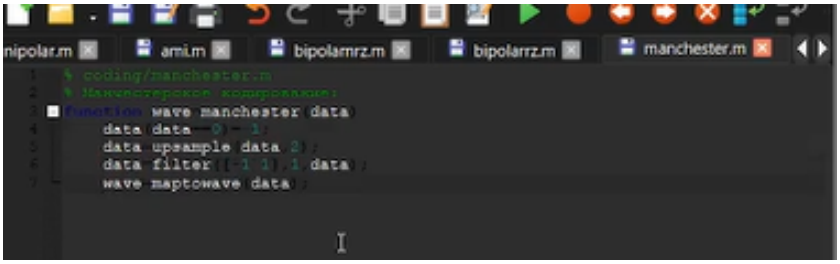


Рис. 3.25: График дифференциального манчестерского кодирования (другой масштаб)

Код RZ демонстрирует трёхуровневую структуру. Каждая единица возвращается к нулю ровно в середине битового интервала, что видно на временной диаграмме (рис. 3.26).

Рис. 3.26: График кодирования RZ

3.6 Спектральные характеристики кодов

Спектр кода AMI показывает отсутствие энергии на нулевой частоте. Это подтверждает пригодность кода для передачи через трансформаторные развязки (рис. 3.27).

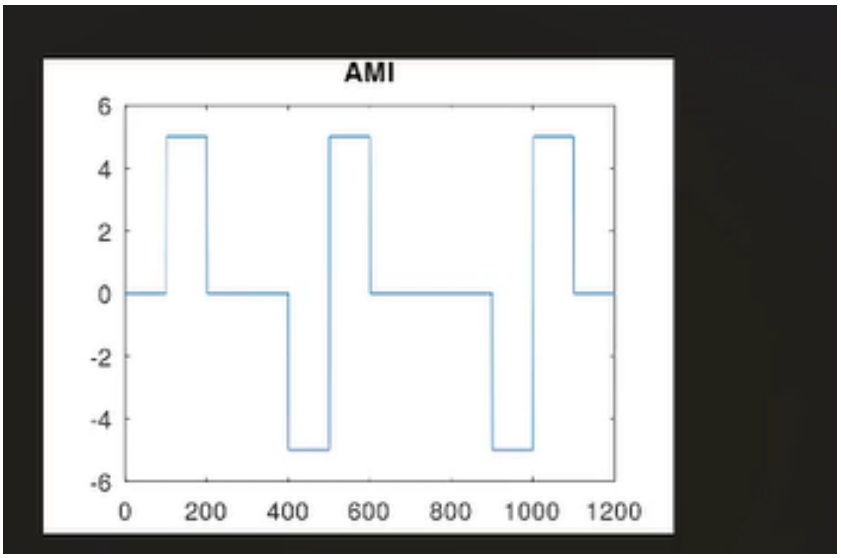


Рис. 3.27: Спектр кодирования AMI

В спектре NRZ-сигнала наблюдается значительная концентрация энергии на низких частотах. Ширина основного лепестка спектра минимальна среди всех изученных кодов (рис. 3.28).

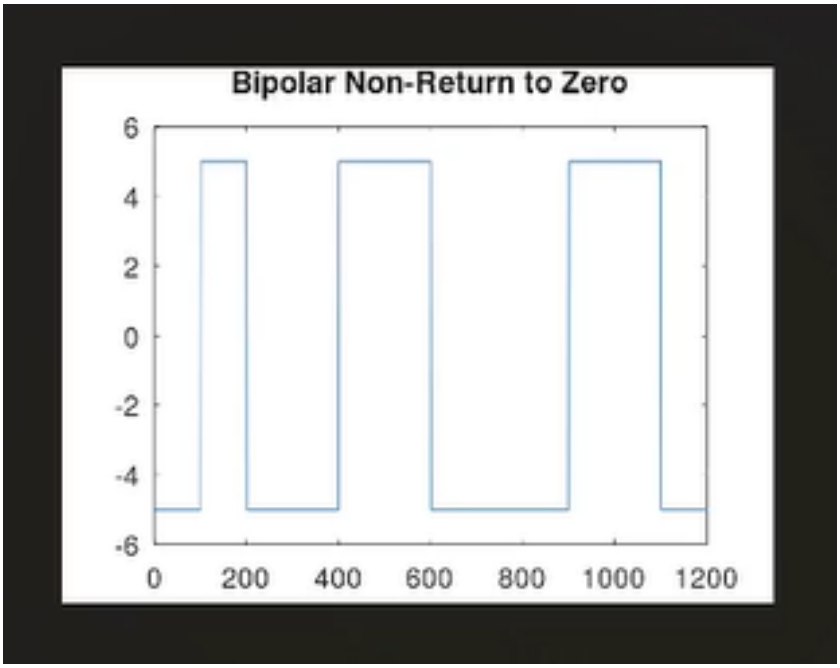


Рис. 3.28: Спектр кодирования NRZ

Спектр кода RZ шире спектра NRZ, так как длительность импульса меньше. Это требует большей полосы пропускания канала связи (рис. 3.29).

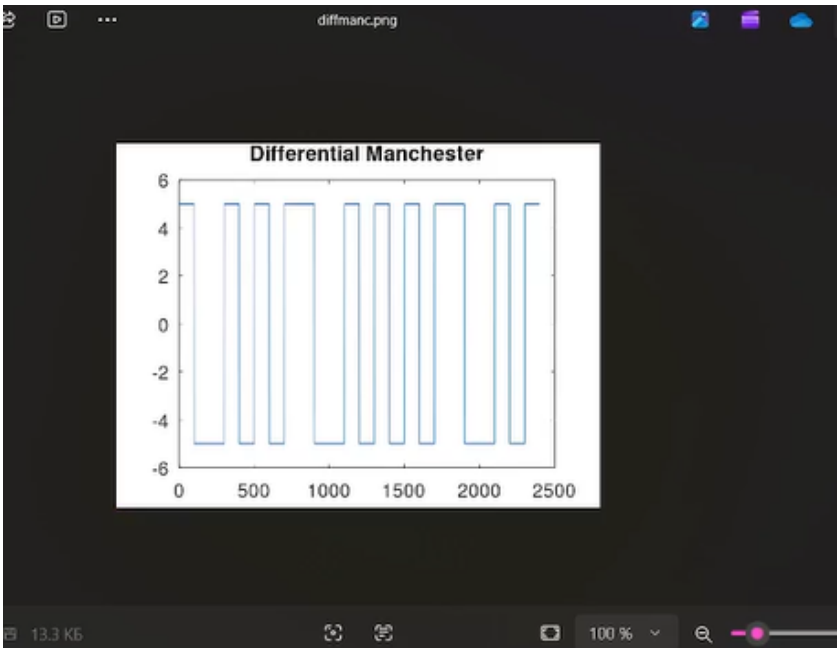


Рис. 3.29: Спектр кодирования RZ

Дифференциальный манчестерский код обладает спектром, смещенным в область высоких частот. Это обусловлено частыми изменениями уровня сигнала (рис. 3.30).

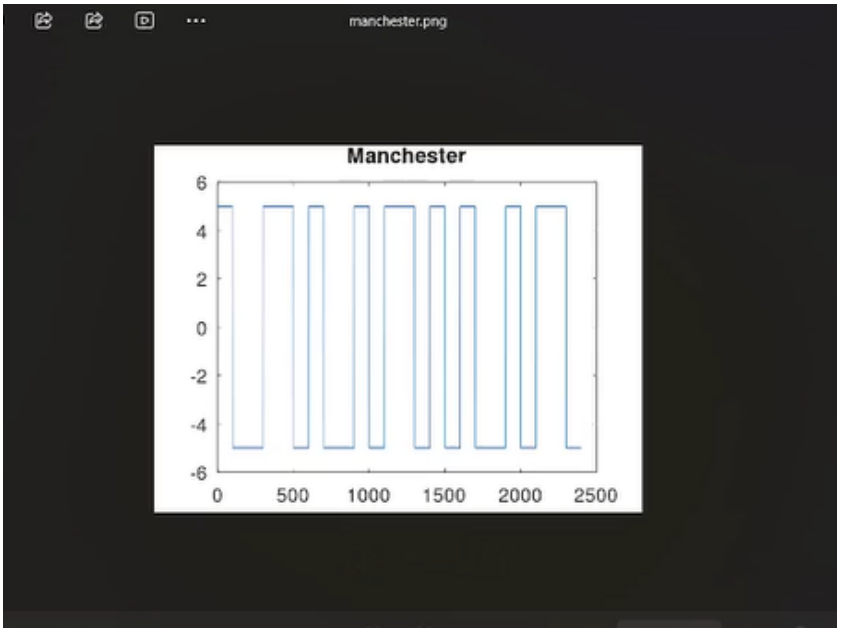


Рис. 3.30: Спектр дифференциального манчестерского кодирования

Спектр манчестерского кодирования схож с дифференциальным вариантом. Основная часть энергии сосредоточена на частотах, близких к тактовой (рис. 3.31).

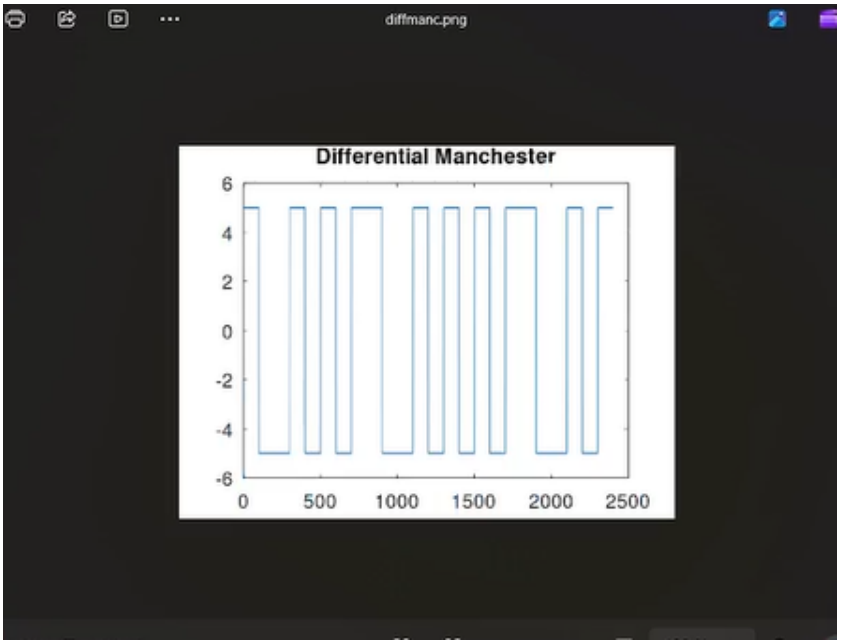


Рис. 3.31: Спектр манчестерского кодирования

Для сравнения я также рассчитала спектр униполярного кода. Он характеризуется наличием ярко выраженной постоянной составляющей (рис. 3.32).

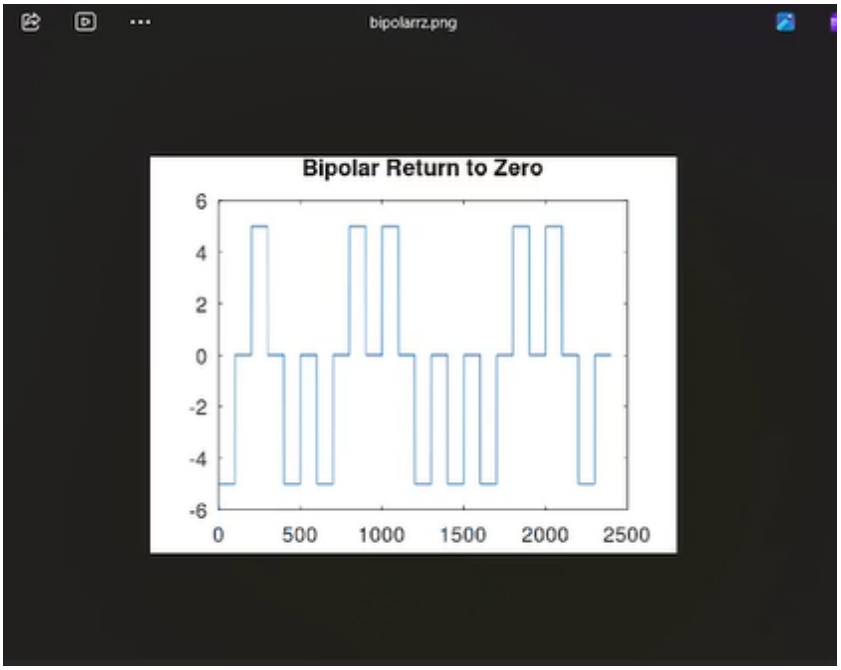


Рис. 3.32: Спектр униполярного кодирования

3.7 Исследование самосинхронизации

Для проверки устойчивости синхронизации я подала на вход последовательность с длинной серией нулей. В коде AMI в этот момент переходы отсутствуют, что ведет к потере синхронизации (рис. 3.33).

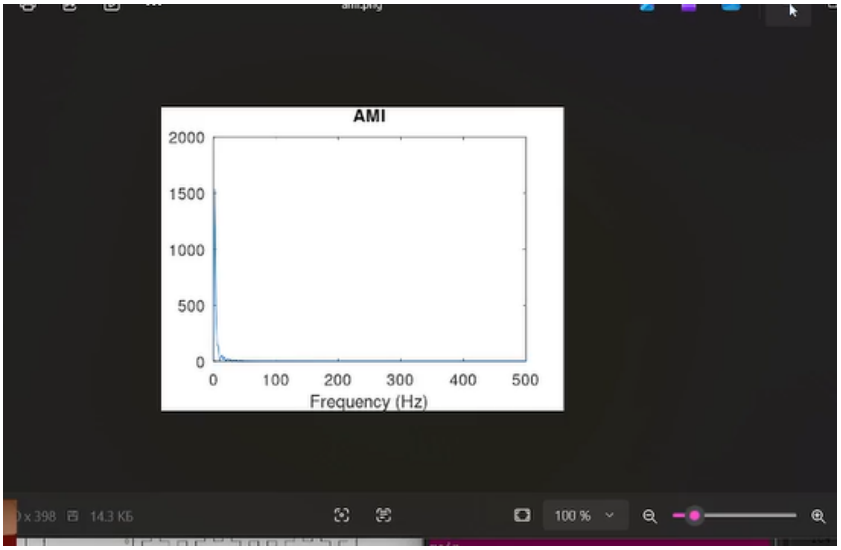


Рис. 3.33: График AMI для проверки самосинхронизации

Аналогичная проблема наблюдается у кода NRZ. При передаче длинной серии единиц или нулей сигнал остается неизменным, и приемник не может выделить такты (рис. 3.34).

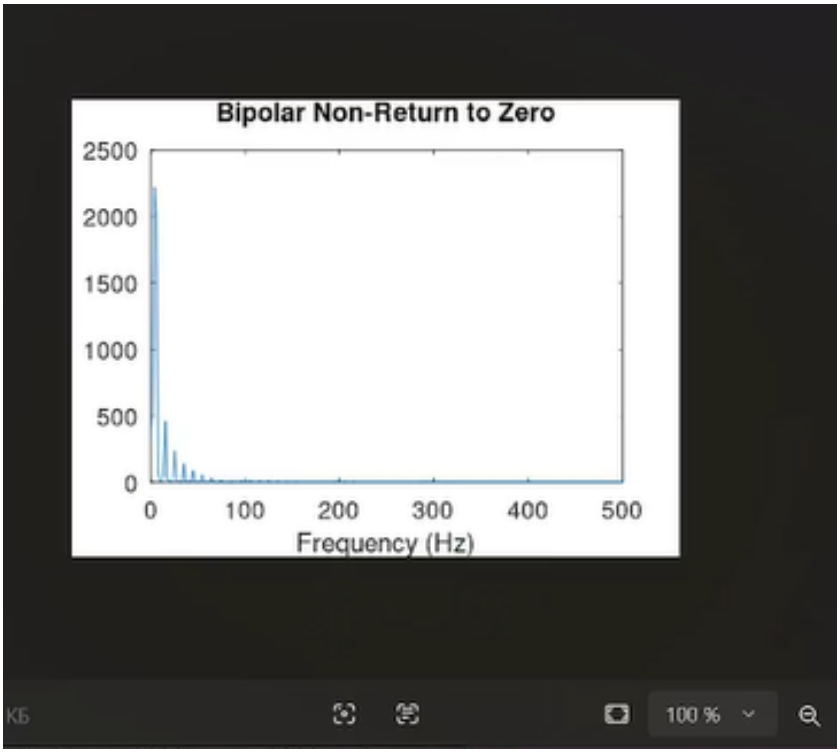


Рис. 3.34: График NRZ для проверки самосинхронизации

Код RZ успешно решает проблему самосинхронизации при передаче единиц, так как в каждом такте есть возврат к нулю. Однако серия нулей по-прежнему остается проблемной зоной (рис. 3.35).

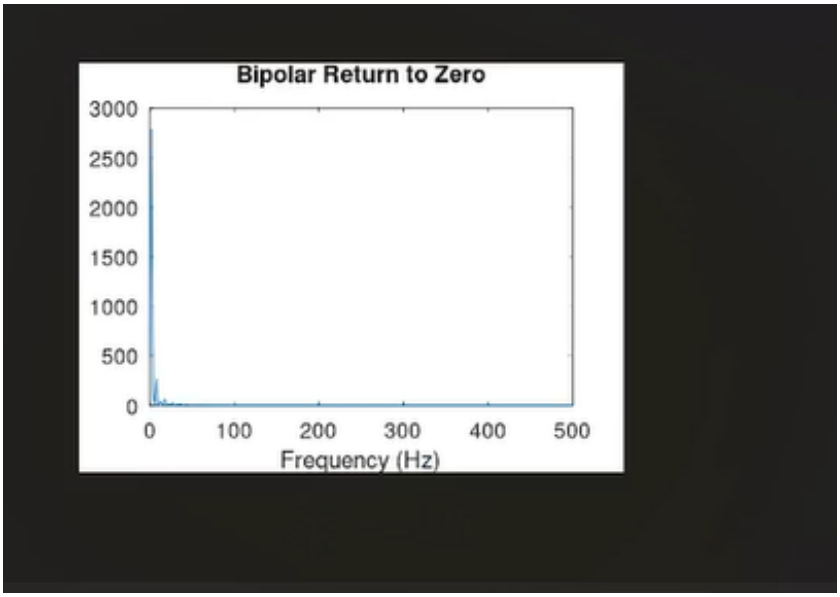


Рис. 3.35: График RZ для проверки самосинхронизации

Дифференциальный манчестерский код обеспечивает идеальную самосинхронизацию. Независимо от передаваемых данных, в сигнале всегда присутствуют переходы (рис. 3.36).

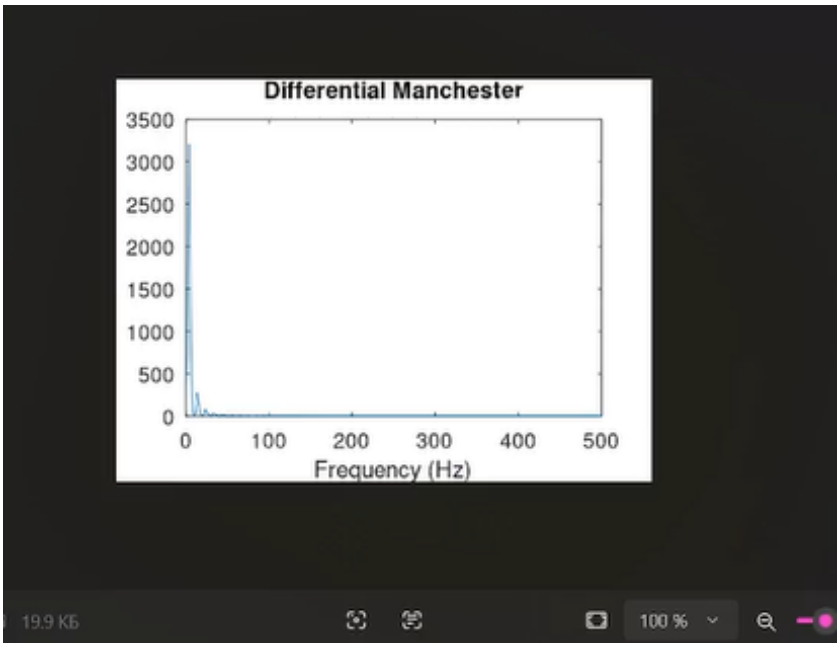


Рис. 3.36: График дифференциального манчестерского кодирования для проверки синхронизации

Манчестерский код также демонстрирует отличные свойства самосинхронизации. Обязательный переход в центре каждого бита позволяет приемнику работать стабильно (рис. 3.37).

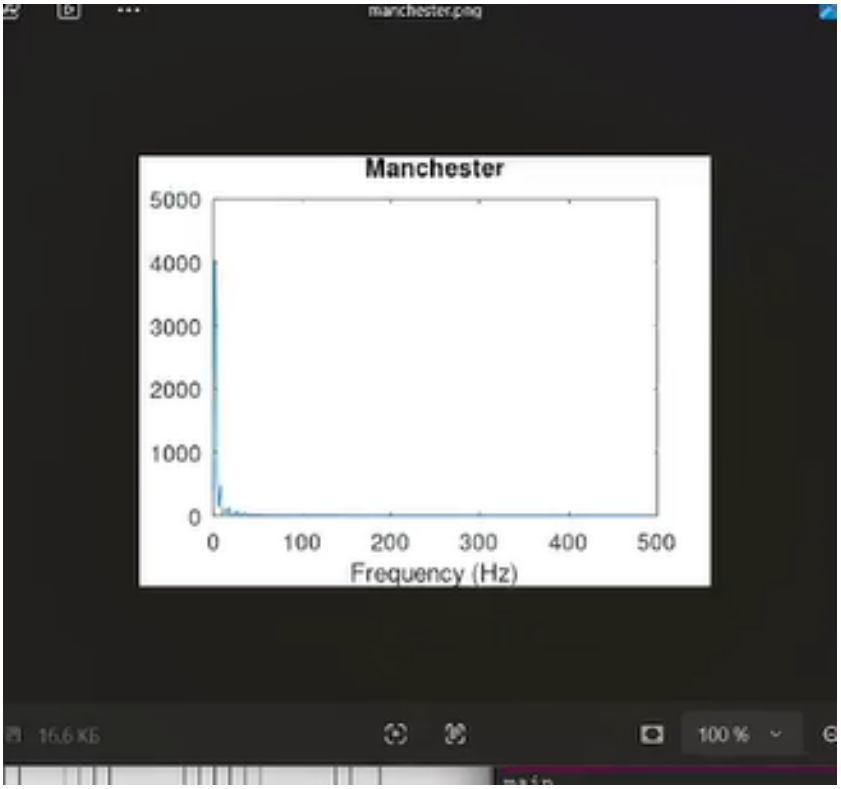


Рис. 3.37: График манчестерского кодирования для проверки синхронизации

В завершение тестов я проверила униполярный код на длинной серии нулей. Результат подтвердил отсутствие переходов и невозможность восстановления синхронизации (рис. 3.38).

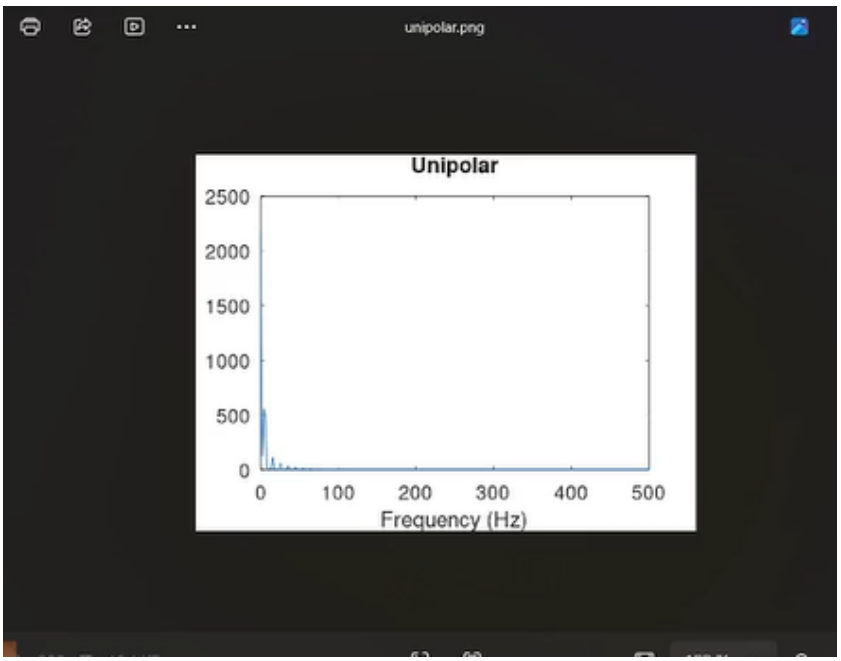


Рис. 3.38: График униполярного кодирования для проверки синхронизации

3.8 Дополнительные результаты анализа

Ниже представлены дополнительные графики спектральных плотностей мощности и временные диаграммы, полученные в ходе уточняющих экспериментов (рис. 3.39 - 3.45).

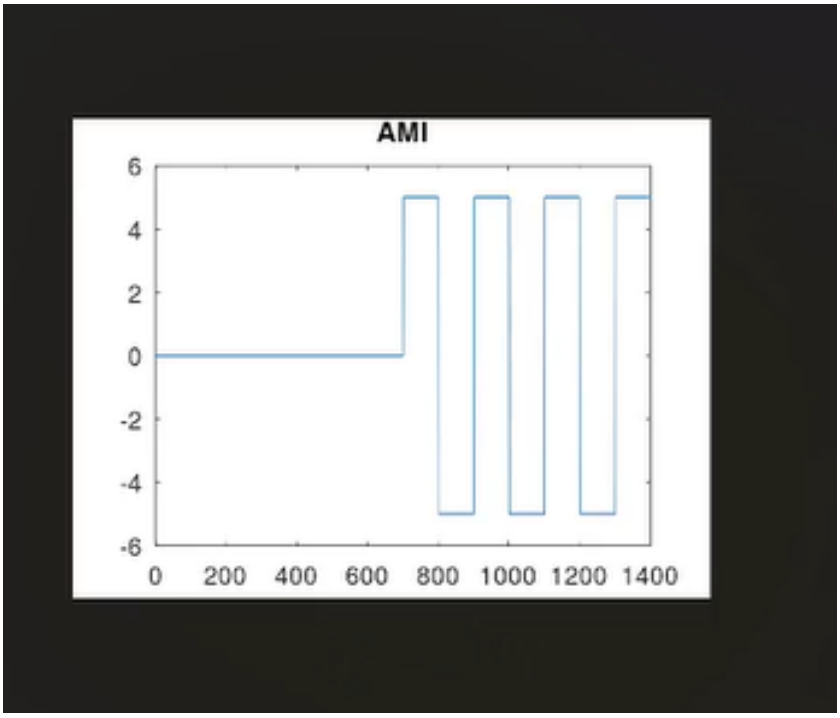


Рис. 3.39: Дополнительный анализ спектральной плотности №1

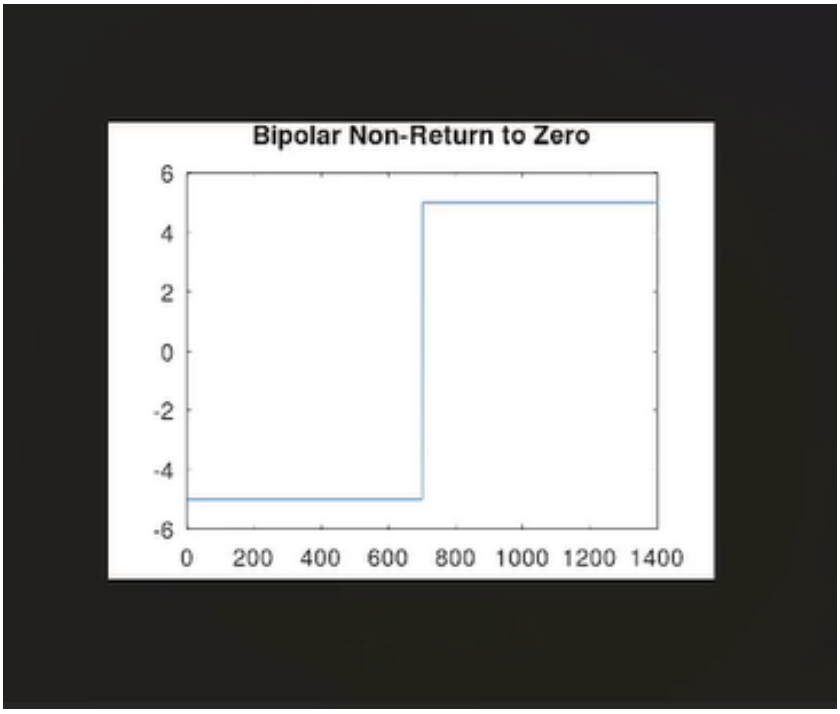


Рис. 3.40: Дополнительный анализ спектральной плотности №2

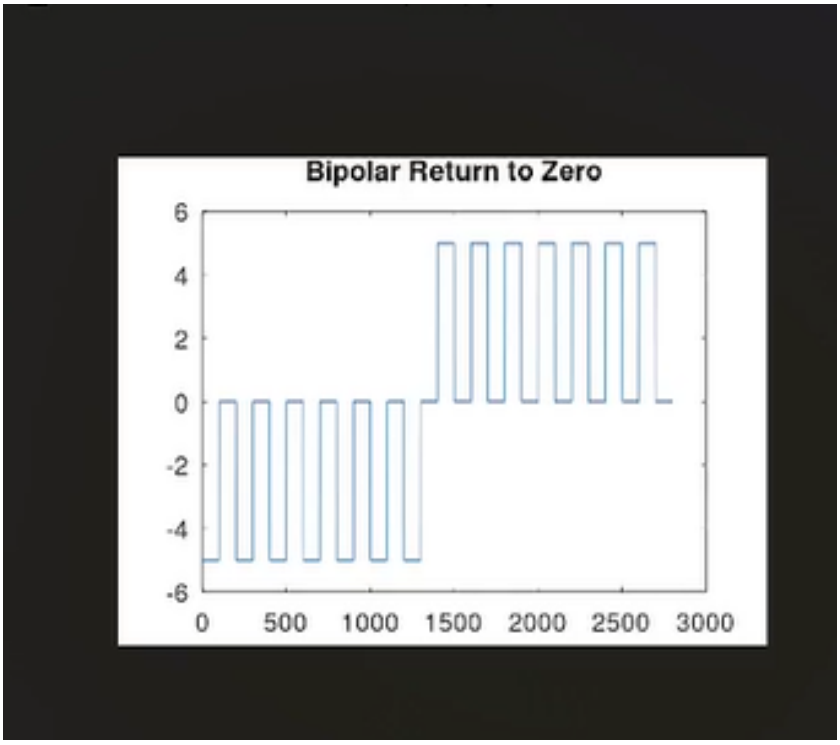


Рис. 3.41: Дополнительный анализ спектральной плотности №3

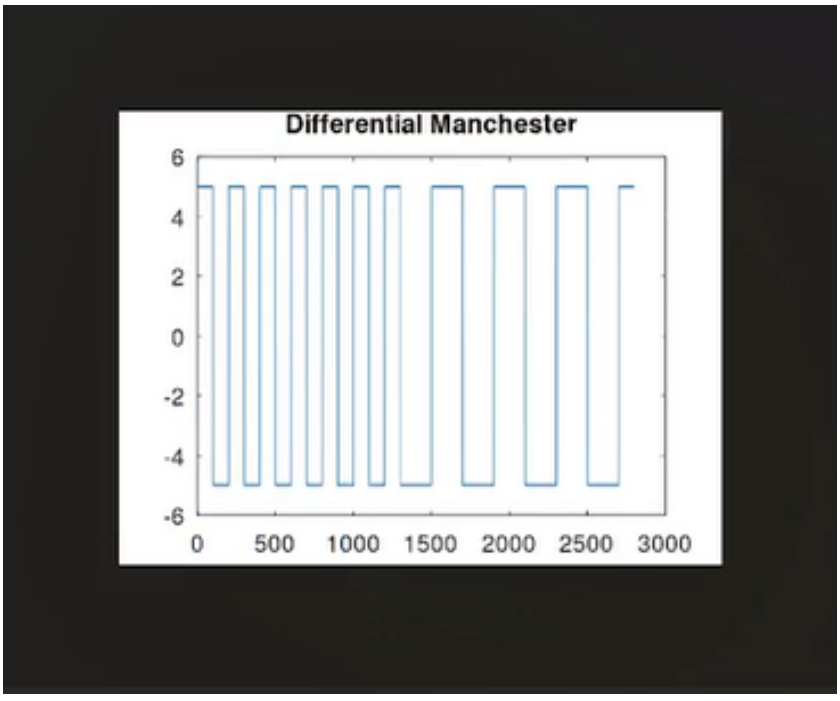


Рис. 3.42: Уточняющая временная диаграмма кодирования №1

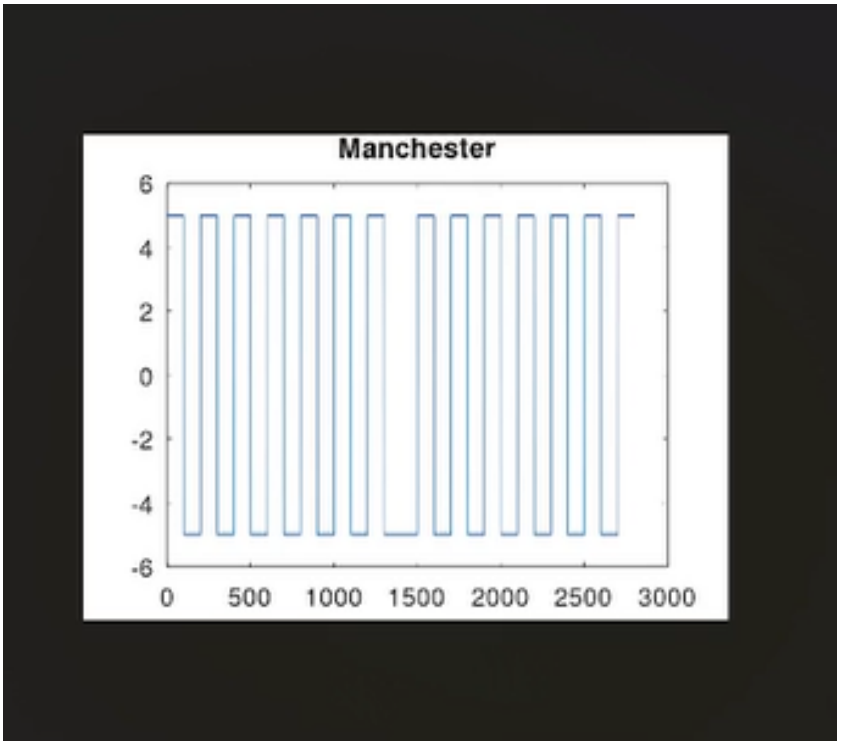


Рис. 3.43: Уточняющая временная диаграмма кодирования №2

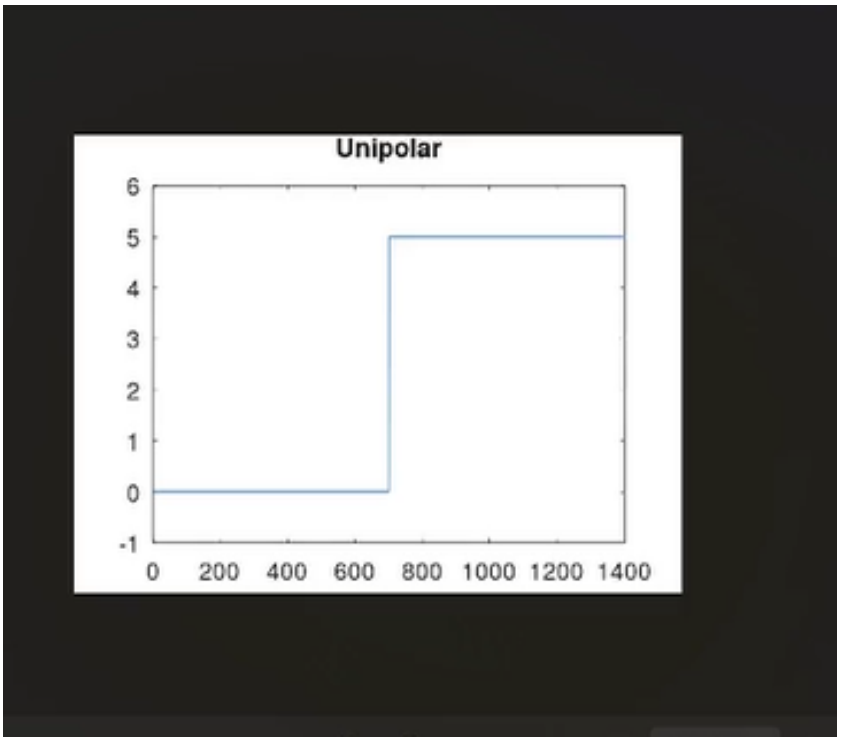


Рис. 3.44: Уточняющая временная диаграмма кодирования №3

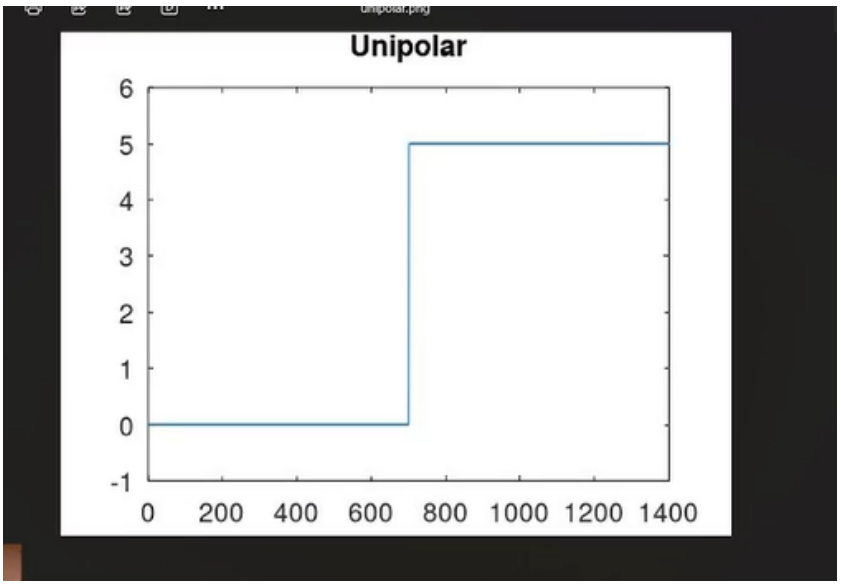


Рис. 3.45: Сводный график результатов экспериментов

4 Выводы

В ходе выполнения лабораторной работы я освоила методы математического моделирования сигналов в среде Octave. Были успешно реализованы алгоритмы спектрального анализа на основе БПФ и моделирование амплитудной модуляции.

Особое внимание было уделено изучению методов линейного кодирования. В результате сравнительного анализа было установлено, что манчестерские коды обладают наилучшими свойствами самосинхронизации, в то время как коды NRZ и AMI требуют специальных методов для предотвращения потери синхронизации при передаче длинных серий одинаковых бит. Полученные навыки необходимы для понимания принципов работы современных систем передачи данных.

5 Ответы на контрольные вопросы

1. **Что такое спектр сигнала?** Это распределение амплитуд или мощностей составляющих гармонических колебаний сигнала по частотам.
2. **В чем преимущество манчестерского кодирования?** Оно обладает свойством самосинхронизации и не имеет постоянной составляющей, что упрощает передачу данных.
3. **Зачем нужна модуляция?** Модуляция позволяет перенести низкочастотный информационный сигнал в высокочастотную область для эффективной передачи на расстояние через физическую среду.