

# Отчёт о лабораторной работе

---

Лабораторная работа №1

ФИО: Бабаев Рахим

Группа: НПИбд-01-23

## **Содержание**

1 Цель работы

2 Задание

3 Выполнение лабораторной работы

4 Выводы

## **1 Цель работы**

Изучение методов кодирования и модуляции сигналов с помощью высокουровневого языка программирования Octave. Определение спектра и параметров сигнала.

Демонстрация принципов модуляции сигнала на примере аналоговой амплитудной модуляции. Исследование свойства самосинхронизации сигнала.

## **2 Задание**

В рамках лабораторной работы необходимо выполнить следующие задания:

- Построить в Octave графики функций  $y_1$  и  $y_2$  на заданном интервале с использованием функции `plot`, оформить подписи осей и заголовки, экспортовать графики в форматы `.eps` и `.png`.
- Реализовать разложение импульсного сигнала в форме меандра в частичный ряд Фурье для различного числа гармоник и построить соответствующие графики.
- Определить спектр двух синусоидальных сигналов и их суммы с использованием быстрого преобразования Фурье, проанализировать влияние частоты дискретизации и нормировки спектра.
- Продемонстрировать принципы аналоговой амплитудной модуляции: получить сигнал с огибающей и исследовать его спектр.
- Реализовать различные методы кодирования сигнала (`unipolar`, `AMI`, `bipolar NRZ`, `bipolar RZ`, `Manchester`, дифференциальный `Manchester`), исследовать свойство самосинхронизации и построить спектры кодированных сигналов.

## **3 Выполнение лабораторной работы**

### **3.1 Построение графиков в Octave**

В среде Octave был создан сценарий `plot_sin.m`, в котором формируется массив значений  $x$  на интервале  $[-10; 10]$ , вычисляются значения функции  $y_1$  и выполняется построение графика с помощью функции `plot`. Для увеличения наглядности на графике включена сетка, подписи осей и заголовок.

Команда запуска сценария из консоли Octave:

```
plot_sin
```

В результате был получен график функции  $y_1 = \sin x + (1/3) \sin 3x + (1/5) \sin 5x$  на интервале  $[-10; 10]$ .

Далее сценарий был модифицирован для одновременного построения графиков функций  $y_1$  и  $y_2$  на одном рисунке с различным типом линий и легендой, что позволяет сравнить форму сигналов.

### 3.2 Разложение импульсного сигнала в частичный ряд Фурье

Был создан сценарий `meandr.m`, в котором задаются параметры меандра (число гармоник, амплитуда, период), формируются коэффициенты ряда Фурье и вычисляются суммы по мере добавления гармоник. Для визуализации использованы несколько подграфиков, отображающих приближение меандра при увеличении числа гармоник.

По мере увеличения числа гармоник график сигнала всё точнее приближается к идеальному меандру, однако на фронтах и спадах сохраняются выраженные выбросы (эффект Гиббса).

### 3.3 Определение спектра и параметров сигнала

В каталоге `spectre1` был создан сценарий `spectre.m`. В нём формируются два синусоидальных сигнала разной частоты, строятся их временные диаграммы и рассчитываются спектры с использованием функции `fft`. Изначально спектры отображаются без нормировки и содержат как положительные, так и отрицательные частоты.

После нормировки спектра и отбраса отрицательных частот были построены исправленные графики, на которых отчётливо видны пики на частотах исходных синусоидальных сигналов. Амплитуды спектральных линий соответствуют амплитудам исходных сигналов с учётом выбранной нормировки.

Для анализа суммы сигналов был создан отдельный сценарий `spectre_sum.m`, в котором вычисляется суммарный сигнал и его спектр. Наблюдается, что спектр суммы содержит те же частотные компоненты, что и спектры исходных сигналов, при этом амплитуды линий изменяются согласно сложению сигналов во времени.

### 3.4 Амплитудная модуляция

В каталоге `modulation` был создан сценарий `am.m`. В нём формируются две синусоиды: низкочастотный модулирующий сигнал и высокочастотная несущая. Амплитудно-модулированный сигнал получается в результате умножения несущей на модулирующий сигнал.

При визуализации хорошо видна огибающая АМ-сигнала, совпадающая с формой модулирующего сигнала. Спектральный анализ показывает наличие частоты несущей и боковых полос, частоты которых равны сумме и разности частоты несущей и модулирующего сигналов.

### **3.5 Кодирование сигнала и самосинхронизация**

В каталоге coding были созданы файлы main.m, maptowave.m и набор функций для различных способов кодирования битовой последовательности (unipolar, AMI, bipolar NRZ, bipolar RZ, Manchester, дифференциальный Manchester). Главный сценарий main.m формирует входные последовательности, вызывает функции кодирования и строит графики полученных сигналов во времени.

Скрипт maptowave.m отвечает за отображение дискретных битовых последовательностей в непрерывный сигнал за счёт развёртки отсчётов и фильтрации. Это позволяет получить реалистичное временное представление линий кода.

Для проверки свойства самосинхронизации использовались специальные входные последовательности с длинными цепочками нулей или единиц. Анализ графиков показал, что коды Manchester и дифференциальный Manchester обладают самосинхронизацией (в каждом такте происходит переход сигнала), тогда как unipolar и некоторые другие коды при длинных сериях одинаковых битов не позволяют однозначно восстановить тактовую частоту.

Функция calcsprectre.m выполняет расчёт спектров кодированных сигналов с помощью FFT и строит зависимости мощности от частоты. По спектрам видно, что самосинхронизирующиеся коды, как правило, имеют выраженные минимумы на нулевой частоте (малую постоянную составляющую), что благоприятно для передачи по реальным линиям связи.

## **4 Выводы**

В ходе лабораторной работы были изучены методы кодирования и модуляции сигналов в среде Octave. Построены графики сложных периодических сигналов и показана связь между временным представлением и спектром сигнала. На примере амплитудной модуляции продемонстрирована связь формы сигнала и его спектра. Исследованы различные способы линейного кодирования битовых последовательностей, проанализированы их спектральные свойства и способность к самосинхронизации. Полученные результаты подтверждают теоретические выводы и подтверждают правильность реализации алгоритмов в Octave.