

Лабораторная работа №1

Методы кодирования и модуляция сигналов

Газизянов Владислав Альбертович

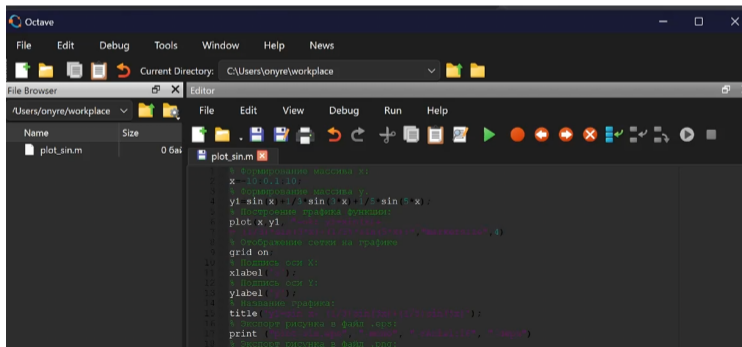
2025-12-20

Содержание I

1. Цели и задачи

Цель: Изучение методов кодирования и модуляции сигналов с использованием Octave.

Задачи: - Построение графиков тригонометрических функций - Разложение сигнала в ряд Фурье - Анализ спектров сигналов - Исследование амплитудной модуляции - Анализ методов кодирования сигналов



The screenshot shows the Octave software interface. The top menu bar includes File, Edit, Debug, Tools, Window, Help, and News. Below the menu is a toolbar with icons for file operations and execution. The 'Current Directory' is set to C:\Users\onyre\workplace. The 'File Browser' on the left shows a file named 'plot_sin.m' with a size of 0.6 KB. The 'Editor' window displays the following MATLAB script:

```
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, 'b');
7 title('График функции y1 = sin(x) + 1/3 * sin(3*x) + 1/5 * sin(5*x)');
8 % Добавление сетки на график
9 grid on;
10 % Подписки осей X:
11 xlabel('x');
12 % Подписки осей Y:
13 ylabel('y1');
14 % Название графика:
15 title('График функции y1 = sin(x) + 1/3 * sin(3*x) + 1/5 * sin(5*x)');
16 % Сохранение рисунка в файл .eps
17 print('plot_sin.eps', 't', 'eps', 'r', 'landscape', 't', 'eps');
18 % Вывод рисунка в файл .png
```

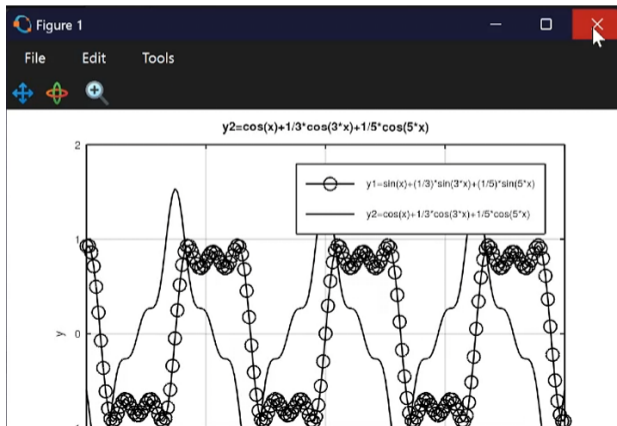
2. Построение графиков в Octave

Реализация тригонометрических функций: - Создание скрипта plot_sin.m -
Построение графика $y = \sin x + (1/3) \sin 3x + (1/5) \sin 5x$ - Экспорт графиков в
форматы EPS и PNG



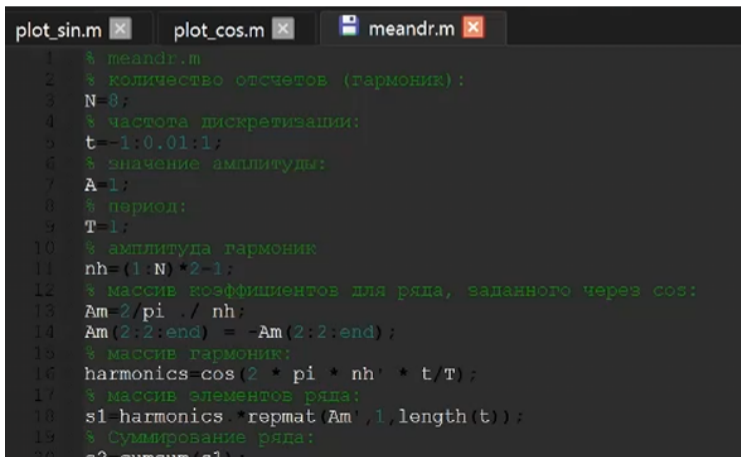
3. Совмещение графиков

Сравнение функций: - Модификация скрипта для двух функций - $y_1 = \sin x + (1/3) \sin 3x + (1/5) \sin 5x$ - $y_2 = \cos x + (1/3) \cos 3x + (1/5) \cos 5x$ - Использование разных типов линий



4. Разложение меандра в ряд Фурье

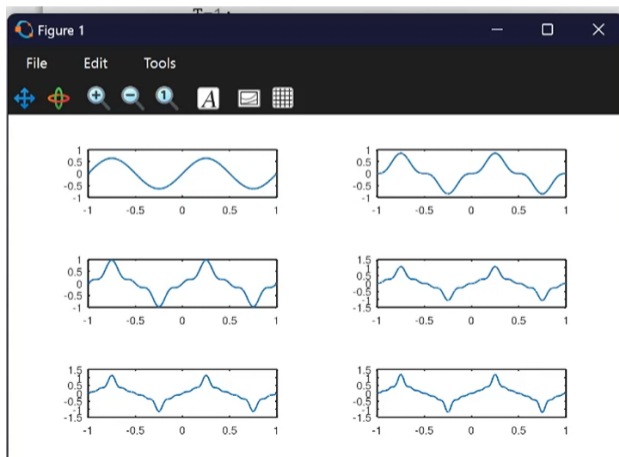
Анализ импульсного сигнала: - Создание скрипта meandr.m - Настройка параметров: $N=8$ гармоник - Частота дискретизации, амплитуда, период



```
plot_sin.m x plot_cos.m x meandr.m x
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=sum(s1);
```

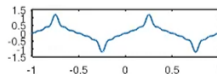
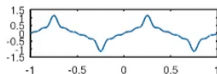
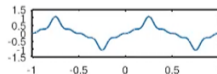
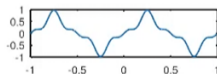
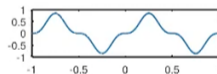
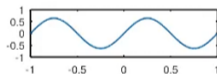
5. Визуализация рядов Фурье

Косинусное разложение: - Формирование меандра из косинусных гармоник -
Использование функции subplot - Графики с 1 по 8 гармоник



6. Синусное разложение меандра

Альтернативный подход: - Модификация кода для синусного ряда - Сравнение с косинусным разложением - Визуализация формирования сигнала



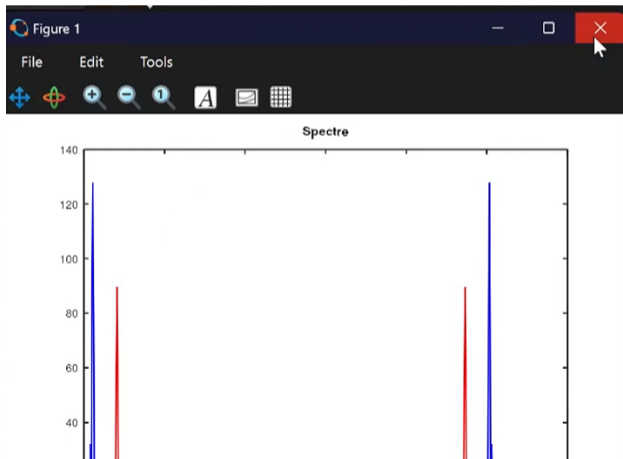
7. Анализ спектров сигналов

Исследование спектральных характеристик: - Создание скрипта spectre.m -
Генерация синусоид 10 Гц и 40 Гц - Частота дискретизации 512 Гц

```
spectre.m ✕
1 % spectre1/spectre.m
2 % Создание каталогов signal и spectre для размещения
3 mkdir 'signal';
4 mkdir 'spectre';
5 % Длина сигнала (с):
6 tmax = 0.5;
7 % Частота дискретизации (Гц) (количество отсчётов):
8 fd = 512;
9 % Частота первого сигнала (Гц):
10 f1 = 10;
11 % Частота второго сигнала (Гц):
12 f2 = 40;
13 % Амплитуда первого сигнала:
14 a1 = 1;
15 % Амплитуда второго сигнала:
16 a2 = 0.7;
17 % Массив отсчётов времени:
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');
```

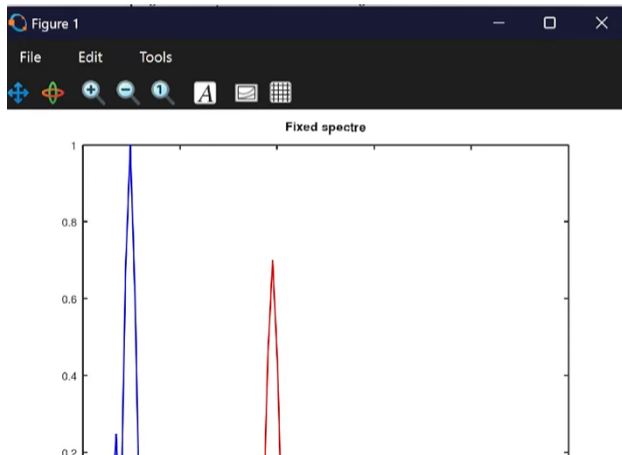
8. Графики сигналов и спектров

Применение БПФ: - Построение исходных сигналов - Вычисление спектров через fft - Пики на частотах 10 Гц и 40 Гц



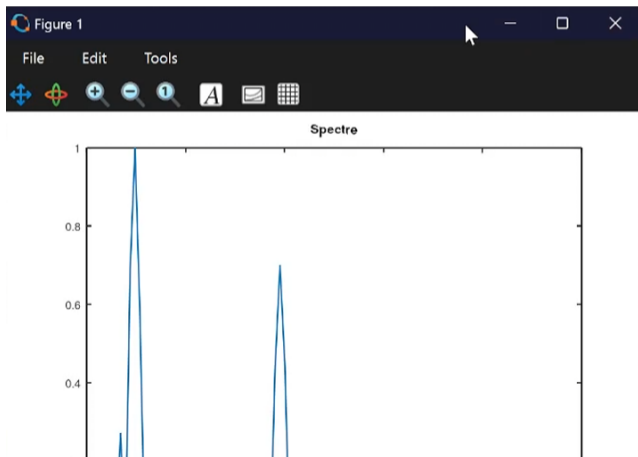
9. Коррекция спектров

Нормировка и исправление: - Отбрасывание отрицательных частот - Нормировка амплитуд - Построение исправленных графиков



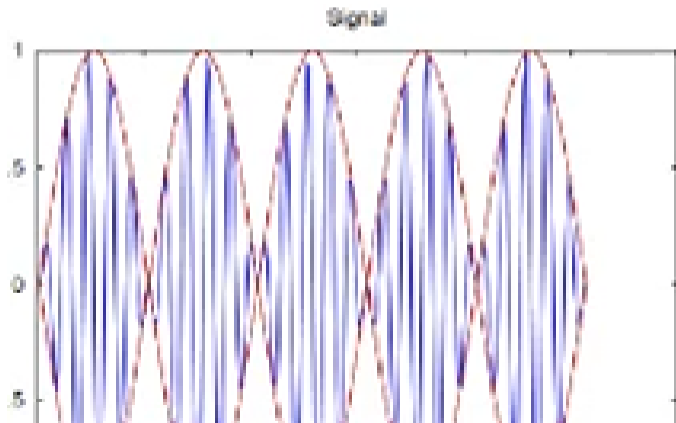
10. Суммирование сигналов

Свойство линейности: - Создание скрипта `spectre_sum.m` - Суммирование двух сигналов - Анализ спектра суммы



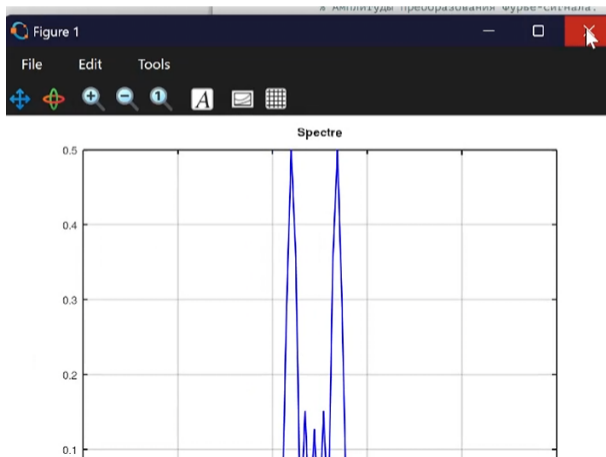
11. Амплитудная модуляция

Исследование модуляции: - Создание скрипта `am.m` - Модуляция: сигнал 5 Гц ×
несущая 50 Гц - Визуализация модулированного сигнала



12. Спектр модулированного сигнала

Анализ частотных характеристик: - Вычисление спектра модулированного сигнала - Боковые полосы вокруг 50 Гц - Подтверждение теоретических ожиданий



13. Методы кодирования сигналов

Подготовка проекта: - Создание каталога coding - Установка библиотеки signal -
Задание тестовых последовательностей

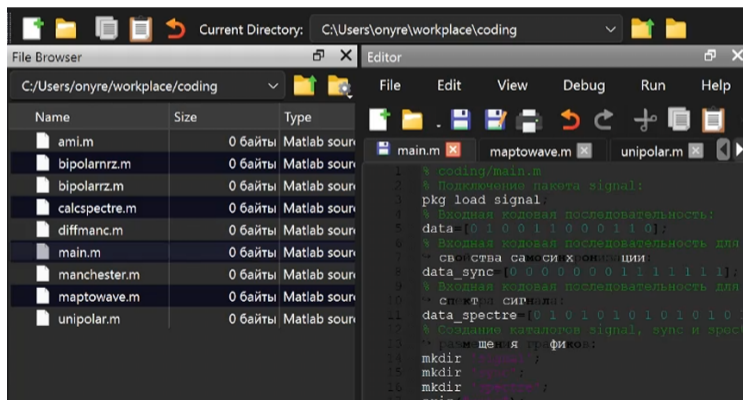
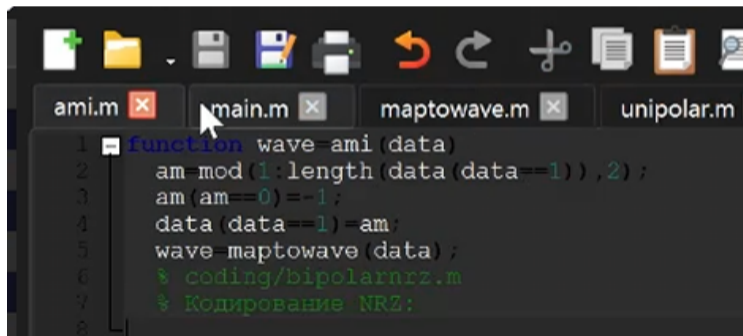


Рисунок 13: Структура проекта и код главного скрипта `main.m`

14. Реализация функций кодирования

Разработка алгоритмов: - Функции для различных методов кодирования - unipolar, ami, bipolarmrz, bipolarrrz - manchester, diffmanc



```
1 function wave=ami(data)
2     am=mod(1:length(data(data==1)),2);
3     am(am==0)=-1;
4     data(data==1)=am;
5     wave=maptowave(data);
6     % coding/bipolarrrz.m
7     % Кодирование NRZ:
```

Рисунок 14: Пример кода функций кодирования AMI и NRZ

15. Визуализация закодированных сигналов

Графики закодированных последовательностей: - Последовательность: [0 1 0 0 1 1 0 0 0 1 1 0] - Униполярное, AMI, NRZ кодирование - Сравнение методов

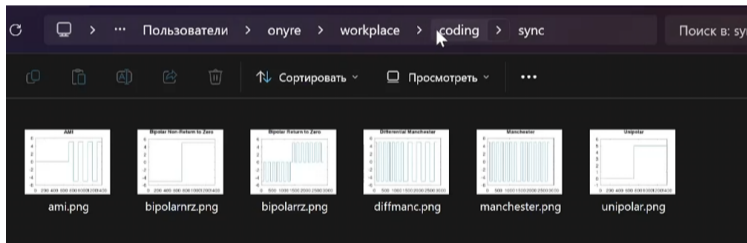


Рисунок 15: Графики закодированных сигналов

16. Выводы

Ключевые результаты: - Получены практические навыки работы с Octave - Исследовано формирование сигналов из гармонических компонент - Подтверждены свойства преобразования Фурье - Проанализированы особенности различных методов кодирования - Определены коды с лучшими характеристиками самосинхронизации - Получен опыт спектрального анализа сигналов