

ГУАП

КАФЕДРА № 44

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

Ст. преподаватель

должность, уч. степень, звание

подпись, дата

Е.К. Григорьев

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №4

МОДУЛЯЦИЯ СИГНАЛОВ В ПАКЕТЕ MATLAB

по курсу: МОДЕЛИРОВАНИЕ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

4143

подпись, дата

Д.В.Пономарев

инициалы, фамилия

Санкт-Петербург 2024

Цель работы

Изучение методов кодирования и модуляции сигналов в системах цифровой обработки сигналов с помощью пакета компьютерного моделирования MATLAB.

Кодирование сигнала

В данной лабораторной работе необходимо закодировать своё ФИО. Для этого воспользуемся вспомогательной таблицей 1.

Таблица 1

Восьмеричный код	Знак	Восьмеричный код	Знак	Восьмеричный код	Знак	Восьмеричный код	Знак
040	Пробел	072	:	121	Q	154	Л
041	!	073	;	122	R	155	М
042	«	074	<	123	S	156	Н
047	'	075	=	124	T	157	О
050	(076	>	125	U	160	П
051)	077	?	126	V	161	Я
052	*	101	A	127	W	162	Р
053	+	102	B	130	X	163	С
054	,	103	C	131	Y	164	Т
055	-	104	D	132	Z	165	У
056	.	105	E	140	Ю	166	Ж
057	/	106	F	141	А	167	В
060	0	107	G	142	Б	170	Ь
061	1	110	H	143	Ц	171	Ы
062	2	111	I	144	Д	172	З
063	3	112	J	145	Е	173	Ш
064	4	113	K	146	Ф	174	Э
065	5	114	L	147	Г	175	Щ
066	6	115	M	150	Х	176	Ч
067	7	116	N	151	И		
070	8	117	O	152	Й		
071	9	120	P	153	К		

ФИО :Пономарев Дмитрий Валерьевич

Далее продемонстрирован восьмеричный и двоичный код.

Итог восьмеричный:

160 157 156 157 155 141 162 145 167

144 155 151 164 162 151 152

167 141 154 145 162 170 145 167 151 176

Итог двоичный:

0 1 1 1 0 0 0 0 0 1 1 0 1 1 1 1 0 1 1 0 1 1 1 0 0 1 1 0 1 1 1 1 0 1 1 0 1 1 0 1 0 1 1 1 0
0 0 0 1 0 1 1 1 0 0 1 0 0 1 1 0 0 1 0 1 0 1 1 1 0 1 1 1 0 1 1 0 0 1 0 0 0 1 1 0 1 1 0 1
0 1 1 0 1 0 0 1 0 1 1 1 0 1 0 0 0 1 1 1 0 0 1 0 0 1 1 0 1 0 0 1 0 1 1 0 1 0 1 0 0 1 1 1
0 1 1 1 0 1 1 0 0 0 0 1 0 1 1 0 1 1 0 0 0 1 1 0 0 1 0 1 0 1 1 1 0 0 1 0 0 1 1 1 1 0 0 0
0 1 1 0 0 1 0 1 0 1 1 1 0 1 1 1 0 1 1 0 1 0 0 1 0 1 1 1 1 1 1 1 0

Затем эту последовательность нужно закодировать тремя методами:
NRZ, RZ и **манчестерским кодом**. Начнём с первого. Код продемонстрирован ниже.

Код программы

Файл *main.m*

```
% main.m
clear all
close all
clc
data=[0 1 1 1 0 0 0 0 0 1 1 0 1 1 1 1 0 1 1 0 1 1 1 0 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 0 1 1 0 1 1 0 1
0 1 1 0 0 0 0 1 0 1 1 1 0 0 1 0 0 1 1 0 0 1 0 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 0 0 1 0 0 0 1 0 0 1 1
0 1 1 0 1 0 1 1 0 1 1 0 1 0 0 1 0 1 1 1 0 1 0 0 0 1 1 1 0 0 1 0 0 1 0 1 1 0 1 0 0 1 0 1 0 1 0 1 0 1 1
1 0 0 1 1 1 0 1 1 1 0 1 1 0 0 0 0 1 0 1 1 0 1 1 0 0 0 1 1 0 0 1 0 1 0 1 1 0 0 0 1 0 1 0 1 1 1 0 0 1 0 0
1 1 1 1 0 0 0 0 1 1 0 0 1 0 1 0 1 1 1 0 1 1 1 0 1 1 0 1 0 0 1 0 1 1 1 1 1 1 1 0];
% Исходная последовательность
wave = data;
figure();
stairs(wave, 'LineWidth', 1);
ylim([-6, 6]); % Устанавливаем пределы оси y
title('Исходная последовательность');
grid on;

% Кодирование NRZ
wave=bipolarnrz(data);
figure()
plot(wave), grid;
ylim([-6 6]);
title('Bipolar Non-Return to Zero');
```

Файл *maptowave.m*

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

Файл *bipolarnrz.m*

```
% bipolarnrz.m
function wave=bipolarnrz(data)
data(data==0)=-1;
wave=maptowave(data);
```

Графики

Соответствующий график последовательности кодированной методом NRZ показаны на рисунок 1 .

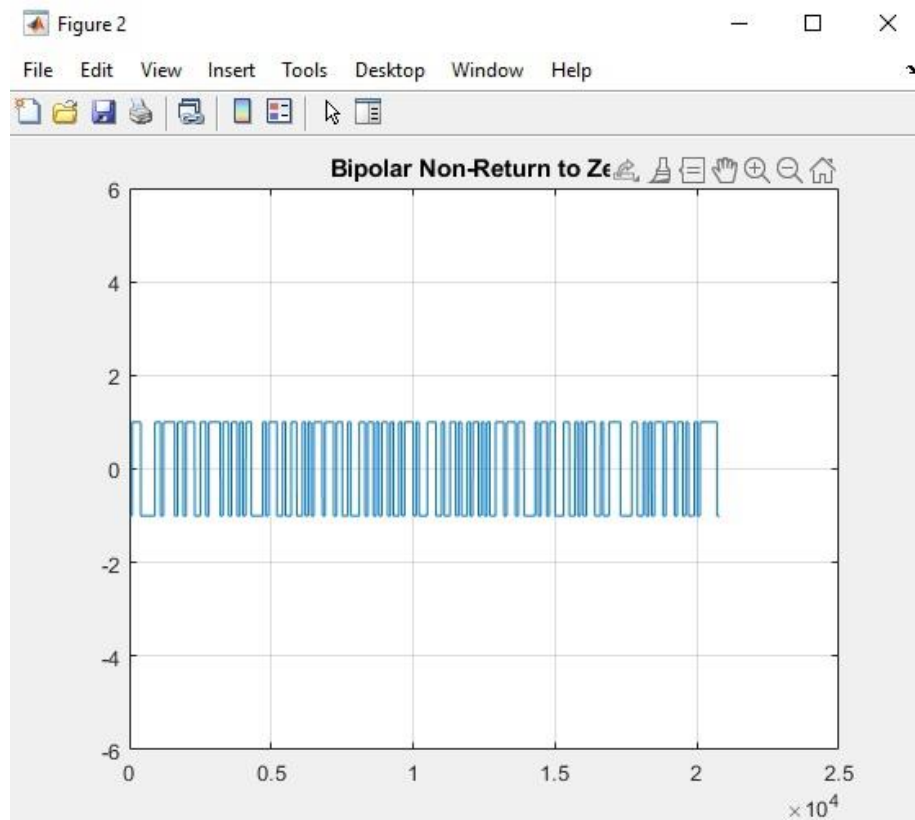


Рисунок 1 – Кодировка NRZ

Затем закодируем с помощью RZ. Код программы и соответствующий график продемонстрированы ниже и на рисунке 2 соответственно.

Код программы

Файл *main.m*

```
% main.m
clear all
close all
clc
data=[0 1 1 1 0 0 0 0 0 1 1 0 1 1 1 1 0 1 1 0 1 1 1 0 0 1 1 0 1 1 1 0 1 1 1 0 1 1 0 1 1 0 1 1 0 1
0 1 1 0 0 0 0 1 0 1 1 1 0 0 1 0 0 1 1 0 0 1 0 1 0 1 1 1 0 1 1 1 0 1 1 0 0 1 0 0 0 1 1
0 1 1 0 1 0 1 1 0 1 0 0 1 0 1 1 1 0 1 0 0 0 1 1 1 0 0 1 0 0 1 1 0 1 0 0 1 0 1 1 0 1 0
1 0 0 1 1 1 0 1 1 1 0 1 1 0 0 0 0 1 0 1 1 0 1 1 0 0 0 1 1 0 0 1 0 1 0 1 1 1 0 0 1 0 0
1 1 1 1 0 0 0 0 1 1 0 0 1 0 1 0 1 1 1 0 1 1 1 0 1 1 0 1 0 0 1 0 1 1 1 1 1 1 1 0];
% Кодирование RZ
wave=bipolarrz(data);
figure()
plot(wave), grid;
ylim([-6 6]);
title('Bipolar Return to Zero');
```

Файл *bipolarrz.m*

```
% bipolarrrz.m
function wave=bipolarrrz(data)
data(data==0)=-1;
data=upsample(data,2);
wave=maptowave(data);
```

Графики

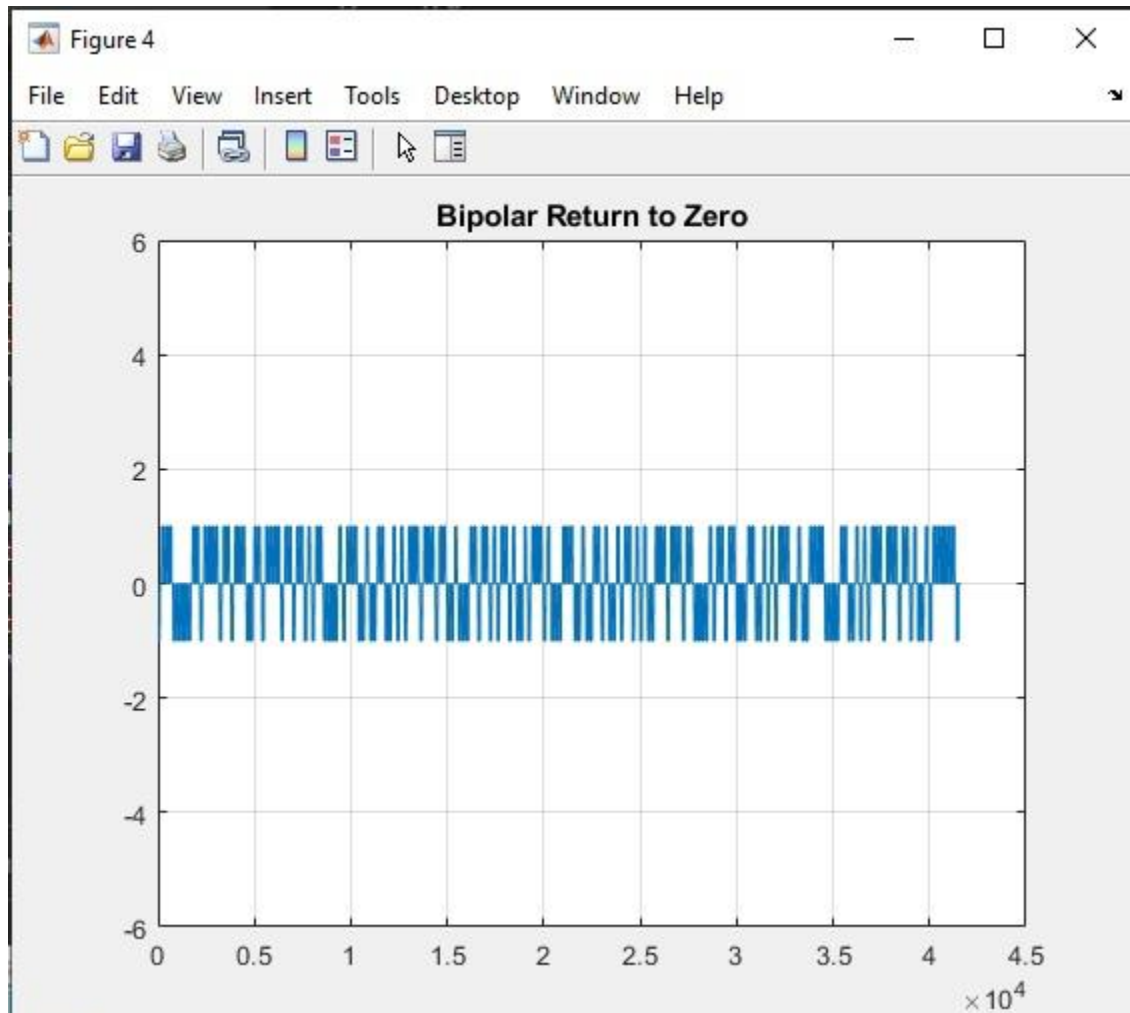


Рисунок 2 – Кодировка RZ

Закодируем сигнал с помощью манчестерского кода. Код продемонстрирован ниже на рисунке 3 показан полученный график.

Код программы

Файл *main.m*

```
% main.m
clear all
close all
clc
data=[0 1 1 1 0 0 0 0 0 1 1 0 1 1 1 1 0 1 1 0 1 1 1 0 0 1 1 0 1 1 1 1 0 1 1 0 1 1 0 1 1 0 1
0 1 1 0 0 0 0 1 0 1 1 1 0 0 1 0 0 1 1 0 0 1 0 1 0 1 1 1 0 1 1 1 0 1 1 0 0 1 0 0 0 1 1
0 1 1 0 1 0 1 1 0 1 0 0 1 0 1 1 1 0 1 0 0 0 1 1 1 0 0 1 0 0 1 1 0 1 0 0 1 0 1 1 0 1 0
1 0 0 1 1 1 0 1 1 1 0 1 1 0 0 0 0 1 0 1 1 0 1 1 0 0 0 1 1 0 0 1 0 1 0 1 1 1 0 0 1 0 0
1 1 1 1 0 0 0 0 1 1 0 0 1 0 1 0 1 1 1 0 1 1 1 0 1 1 0 1 0 0 1 0 1 1 1 1 1 1 1 0];
% Манчестерское кодирование
```

```

wave=manchester(data);
figure()
plot(wave), grid;
ylim([-6 6]);
title('Manchester');

```

Файл *manchester.m*

```

% manchester.m
function wave=manchester(data)
data(data==0)=-1;
data=upsample(data,2);
data=filter([-1 1],1,data);
wave=maptowave(data);

```

Графики

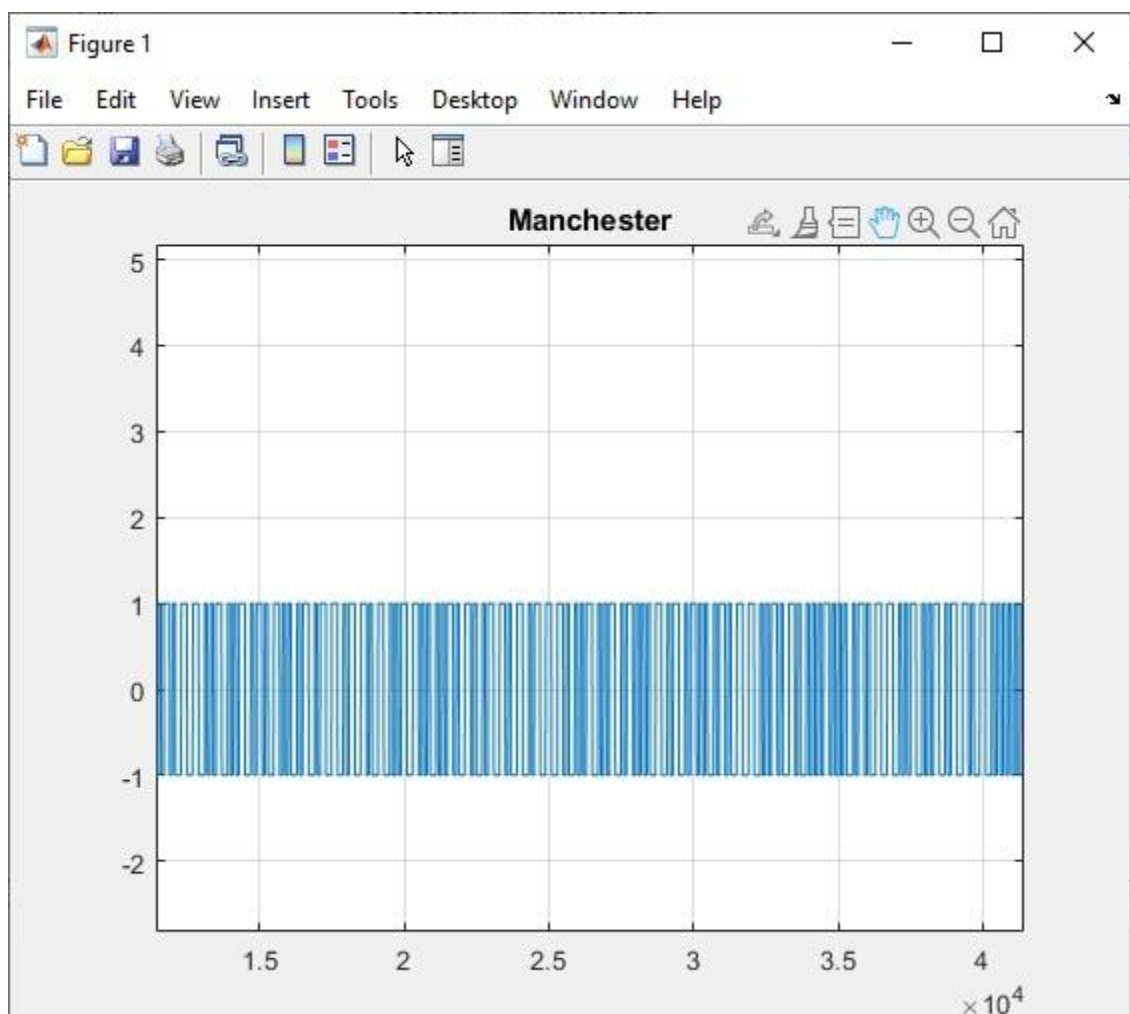


Рисунок 3 – Манчестерская кодировка

Перейдём к выполнению второй части лабораторной работы. Для начала сформируем амплитудную модуляцию. Код программы продемонстрирован ниже. На рисунке 5 показан полученный график.

Код программы

Файл *main.m*

```
% main.m
clear all
close all
clc

% Задаем входную кодовую последовательность:
data=[0 1 1 1 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 1 1 1 1 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 1 1 0 1 0 1 1 1 0 0 0 0 1 0 1 1 1 1 0
0 1 0 0 1 1 1 0 0 1 0 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1 0 0 1 0 0 0 1 1 1 0 1 1 0 1 0 1 1 1 0 1 0 0 1 0 1 1 1 1 0 1 0 0 0 1 1 1 1 0
0 1 0 0 1 1 0 1 0 0 1 0 1 1 1 0 1 0 1 0 0 1 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 0 1 0 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 1 0 1 1 1 1 0
0 1 0 0 1 1 1 1 1 0 0 0 0 1 1 0 0 1 0 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1 0 1 0 0 1 0 1 1 1 1 1 1 1 0];

% Параметры модуляции
amplitude = 2; % Амплитуда модуляции
carrier_freq = 10; % Частота несущего сигнала

% Генерация временных отсчетов
t = linspace(0, length(data), length(data)*100); % Увеличиваем разрешение временной
оси для плавности

% Формирование сигнала амплитудной модуляции
am_wave = zeros(size(t)); % Инициализация сигнала модуляции нулями
for i = 1:length(data)
    if data(i) == 1
        am_wave((i-1)*100+1:i*100) = amplitude * sin(2*pi*carrier_freq*(i-1) + pi); %
        Модуляция сигнала на уровне amplitude
    else
        am_wave((i-1)*100+1:i*100) = 0; % В случае нулевого бита сигнал равен нулю
    end
end

% Построение графика амплитудной модуляции
plot(t, am_wave);
grid on;
title('Амплитудная модуляция');
```

Графики

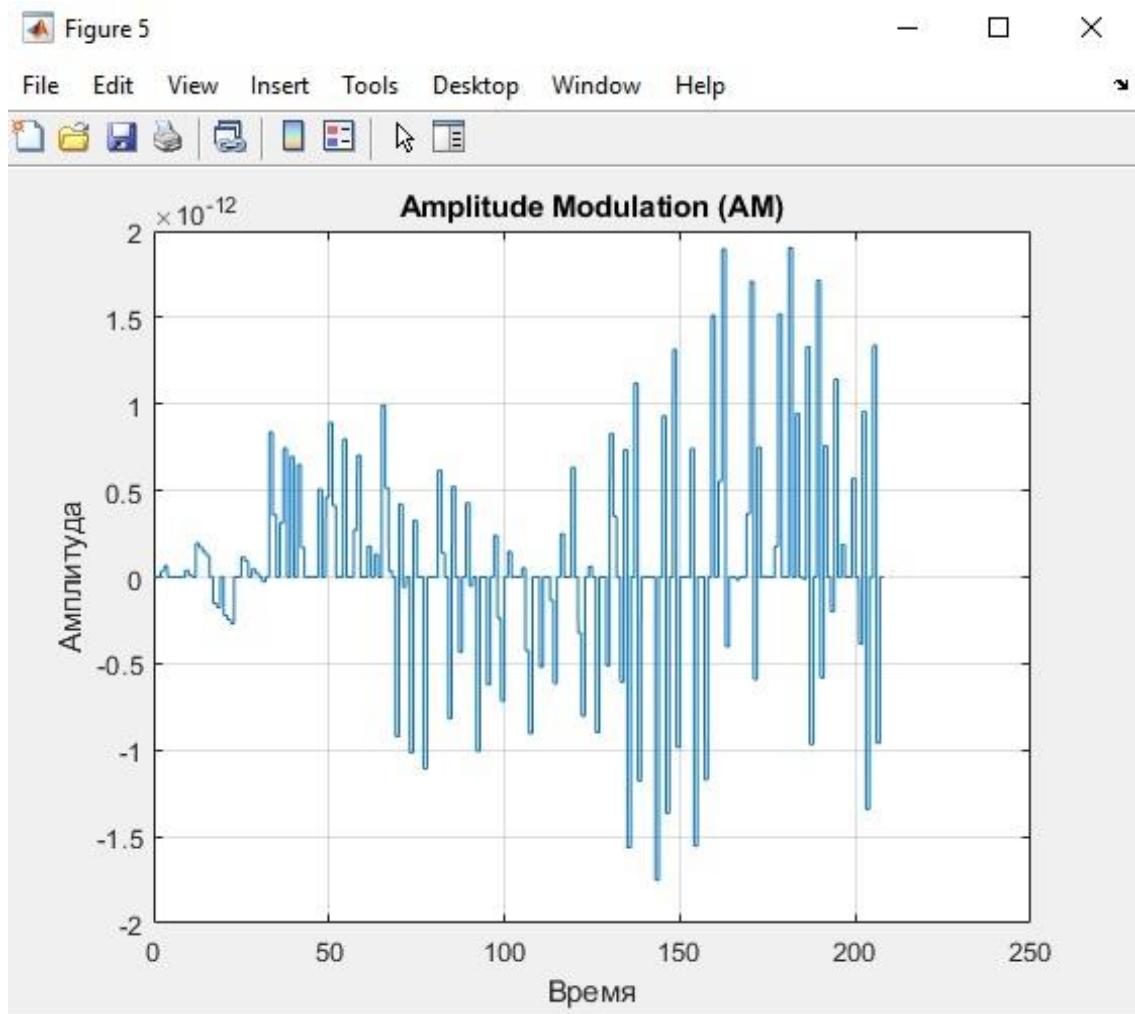


Рисунок 4 – Амплитудная модуляция

Теперь построим частотную модуляцию. Код программы продемонстрирован ниже. На рисунке 5 показан полученный график.

Код программы

Файл *main.m*

```
% main.m
clear all
close all
clc
data=[0 1 1 1 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 1 1 0 1 1 1 1 0 1 1 0 1 1 0 1 0 1 1 0 0 0 0 1 0 1 1 1 0
0 1 0 0 1 1 0 0 1 0 1 0 1 1 1 1 0 1 1 1 0 1 1 0 0 1 0 0 0 1 1 0 1 1 0 1 0 1 1 0 1 0 0 1 0 1 1 1 1 0 1 0 0 0 1 1 1 0
0 1 0 0 1 1 0 1 0 0 1 0 1 1 0 1 0 1 0 0 0 1 1 1 0 1 1 1 0 1 1 0 0 0 0 1 0 1 1 0 1 1 0 0 0 1 1 0 0 1 0 1 0 1 1 1 0
0 1 0 0 1 1 1 1 0 0 0 0 1 1 0 0 1 0 1 0 1 1 1 0 1 1 1 0 1 1 0 1 0 0 1 0 1 1 1 1 1 1 0];
% Параметры модуляции
carrier_freq = 10; % Частота несущего сигнала
frequency_deviation = 5; % Максимальное отклонение частоты для модуляции бита 1

% Генерация временных отсчетов
t = linspace(0, length(data), length(data)*100); % Увеличиваем разрешение временной
оси для плавности

% Формирование сигнала частотной модуляции
fm_wave = zeros(size(t)); % Инициализация сигнала модуляции нулями
```



```

for i = 1:length(data)
    if data(i) == 1
        fm_wave((i-1)*100+1:i*100) = sin(2*pi*carrier_freq*(i-1) +
2*pi*frequency_deviation*sum(data(1:i-1))/carrier_freq + pi); % Модуляция сигнала с
изменением частоты
    else
        fm_wave((i-1)*100+1:i*100) = sin(2*pi*carrier_freq*(i-1)); % Сигнал без
изменения частоты
    end
end

% Построение графика частотной модуляции
plot(t, fm_wave);
grid on;
title('Частотная модуляция');

```

Графики

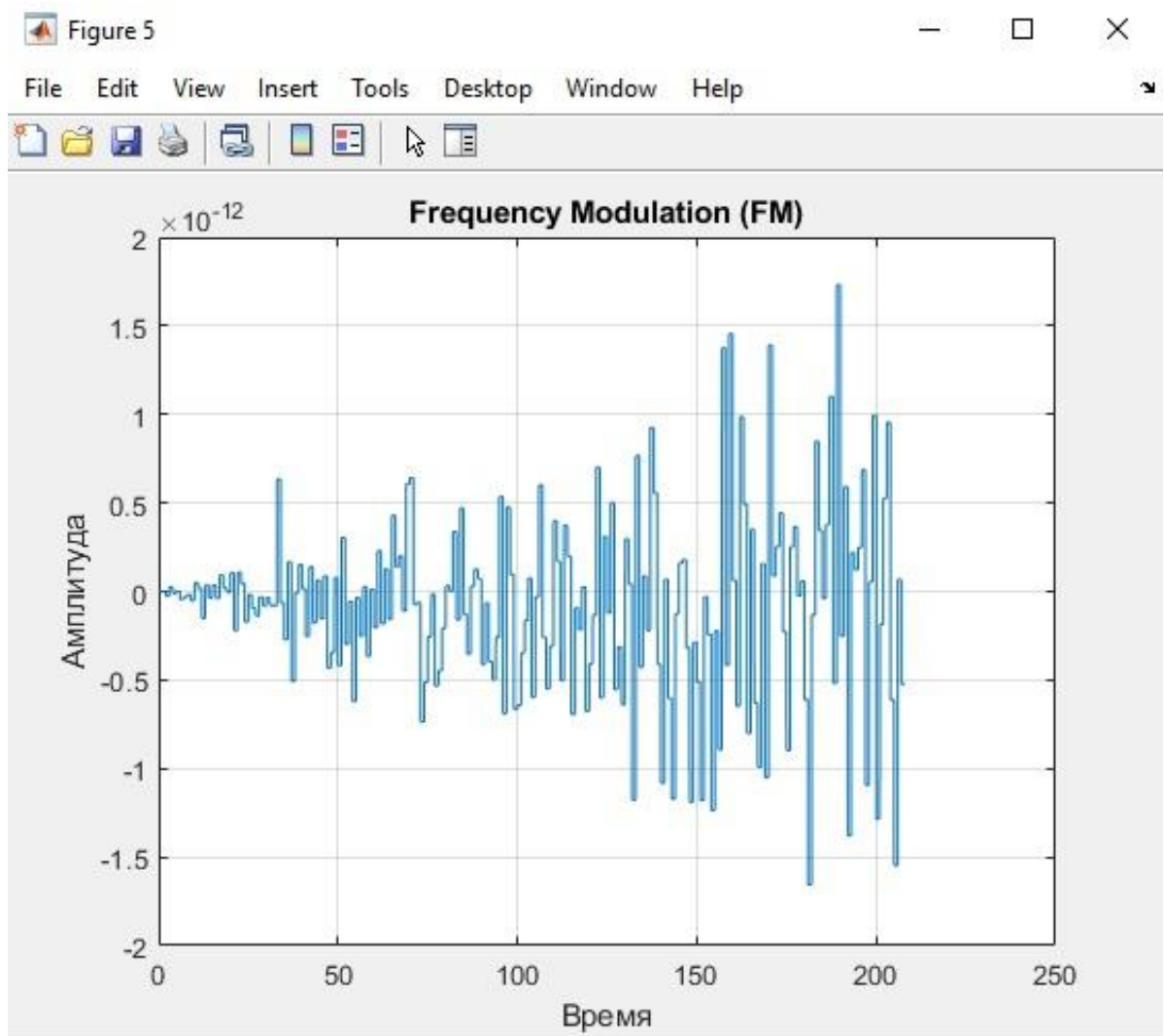


Рисунок 5 – Частотная модуляция

Наконец, построим фазовую модуляцию. Код программы продемонстрирован ниже. На рисунке 6 показан полученный график.

Код программы

Файл main.m

```
% main.m
clear all
close all
clc
data=[0 1 1 1 0 0 0 0 1 1 0 1 1 1 1 0 1 1 0 1 1 1 0 0 1 1 0 1 1 1 1 0 1 1 0 1 1 0 1 0 1 1 0 0 0 0 1 0 1 1 1 0
0 1 0 0 1 1 0 0 1 0 1 0 1 1 1 1 0 1 1 1 0 1 1 0 0 1 0 0 0 1 1 0 1 1 0 1 0 1 1 0 1 0 0 1 0 1 1 1 0 1 0 0 0 1 1 1 0
0 1 0 0 1 1 0 1 0 0 1 0 1 1 0 1 0 1 0 0 1 1 1 0 1 1 1 0 1 1 0 0 0 0 1 0 1 1 0 1 1 0 0 0 1 1 0 0 1 0 1 0 1 1 1 0
0 1 0 0 1 1 1 1 0 0 0 0 1 1 0 0 1 0 1 0 1 1 1 0 1 1 1 0 1 1 0 1 0 0 1 0 1 1 1 1 1 1 0];
% Параметры модуляции
carrier_freq = 10; % Частота несущего сигнала
phase_shift = pi/2; % Сдвиг фазы для модуляции бита 1

% Генерация временных отсчетов
t = linspace(0, length(data), length(data)*100); % Увеличиваем разрешение временной
оси для плавности

% Формирование сигнала фазовой модуляции
pm_wave = zeros(size(t)); % Инициализация сигнала модуляции нулями
for i = 1:length(data)
    if data(i) == 1
        pm_wave((i-1)*100+1:i*100) = sin(2*pi*carrier_freq*(i-1) + phase_shift); %
        Модуляция сигнала с фазовым сдвигом
    else
        pm_wave((i-1)*100+1:i*100) = sin(2*pi*carrier_freq*(i-1)); % Сигнал без
        фазового сдвига
    end
end

% Построение графика фазовой модуляции
plot(t, pm_wave);
grid on;
title('Фазовая модуляция');
```

Графики

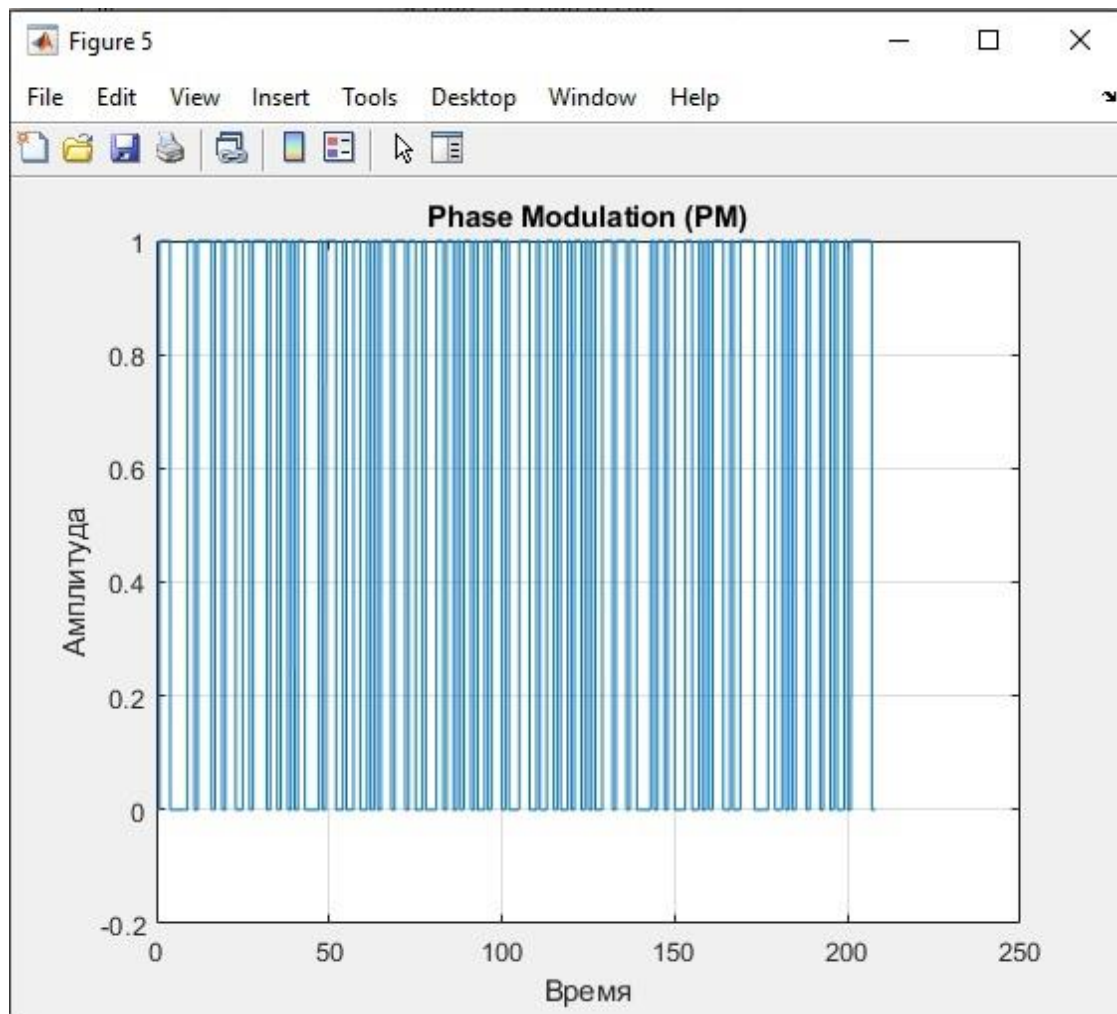


Рисунок 6 – Фазовая модуляция

Выводы

В данной лабораторной работе были изучены методы кодирования и модуляции сигналов в системах цифровой обработки сигналов с помощью пакета компьютерного моделирования MATLAB.