

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. №

подпись, дата

Е.Д.Тегай

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 таково:

Тегай Екатерина Дмитриевна

Далее продемонстрирован восьмеричный и двоичный код для каждой буквы соответственно.

$T \rightarrow 164 \rightarrow 01110100$

$E \rightarrow 145 \rightarrow 01100101$

$G \rightarrow 147 \rightarrow 01100111$

A → 141 → 01100001

Й → 152 → 01101010

Е → 145 → 01100101

К → 153 → 01101011

A → 141 → 01100001

Т → 164 → 01110100

Е → 145 → 01100101

Р → 162 → 01110010

И → 151 → 01101001

Н → 156 → 01101110

A → 141 → 01100001

Д → 144 → 01100100

М → 155 → 01101101

И → 151 → 01101001

Т → 164 → 01110100

Р → 162 → 01110010

И → 151 → 01101001

Е → 145 → 01100101

В → 167 → 01110111

Н → 156 → 01101110

A → 141 → 01100001

Итого получаем последовательность:

Затем эту последовательность нужно закодировать тремя методами:
продемонстрирован ниже.

Код программы

Файл *main.m*

```
% main.m
clear all
close all
clc
% Задаем входную кодовую последовательность:
data=[0 1 1 1 0 1 0 0 0 1 1 0 0 1 0 1 0 1 1 0 0 1 1 1 0 1 1 0 0 0 0 1 0 1 1 0 1 0 1 0
0 1 1 0 0 1 0 1 0 1 1 0 1 0 1 1 0 1 1 0 0 0 0 1 0 1 1 1 0 1 0 0 0 1 1 0 0 1 0 1 0 1 1
1 0 0 1 0 0 1 1 0 1 0 0 1 0 1 1 0 1 1 1 0 0 1 1 0 0 0 0 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 0 1 0 1 1 0 0 1 0 1 0
1 1 1 0 1 1 1 0 1 1 0 1 1 1 0 0 1 1 0 0 0 1];

% 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

Соответствующие графики исходной последовательности и кодированной методом NRZ показаны на рисунках 1 – 2.

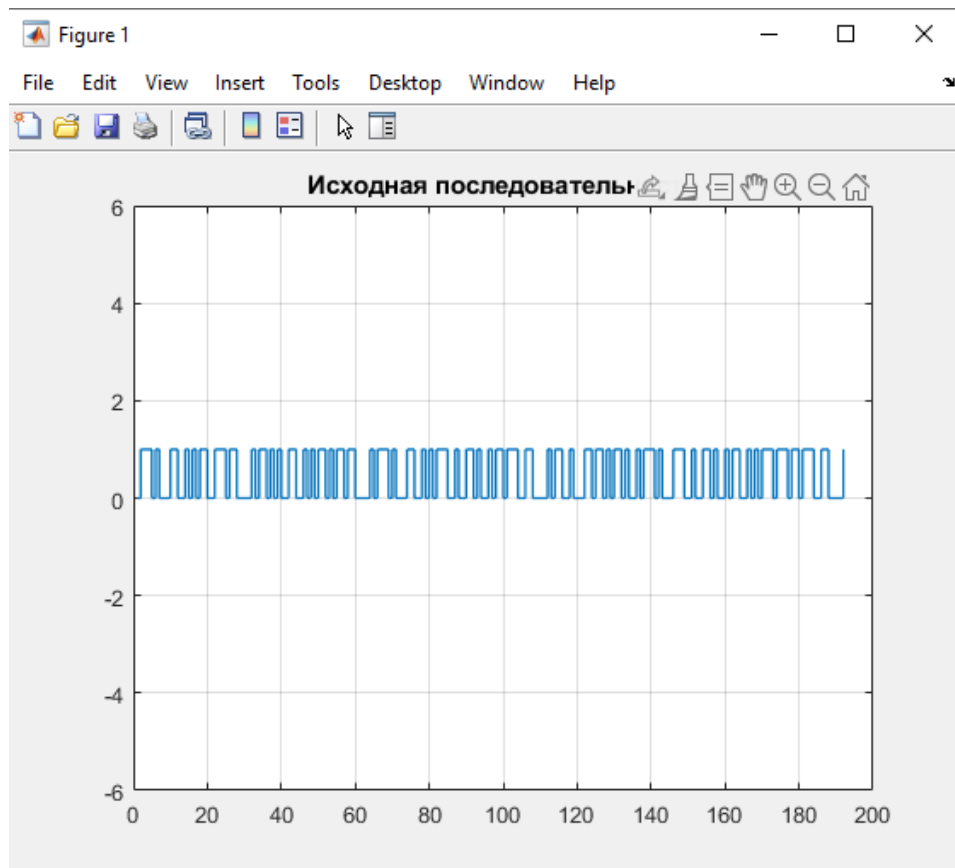


Рисунок 1 – Исходная последовательность

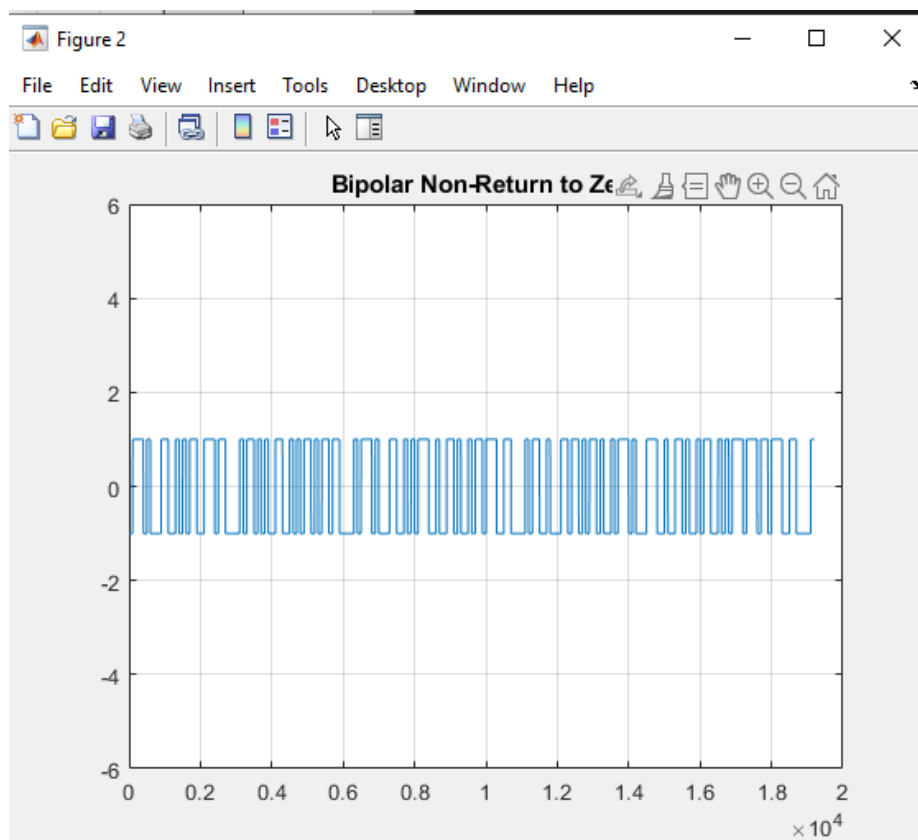


Рисунок 2 – Кодировка NRZ

```
% main.m
clear all
close all
clc
% Задаем входную кодовую последовательность:
data=[0 1 1 1 0 1 0 0 0 1 1 0 0 1 0 1 0 1 1 0 0 1 1 1 0 1 1 0 0 0 0 1 0 1 1 0 1 0 1 0
0 1 1 0 0 1 0 1 0 1 1 0 1 0 1 1 0 1 1 0 0 0 0 1 0 1 1 1 0 1 0 0 0 1 1 0 0 1 0 1 0 1 1
1 0 0 1 0 0 1 1 0 1 0 0 1 0 1 1 0 1 1 1 0 0 0 1 1 0 0 0 0 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 0 1 0 1 0
1 1 1 0 1 1 1 0 1 1 0 1 1 1 0 0 1 1 0 0 0 0 1];

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

Наконец, закодируем сигнал с помощью манчестерского кода. Код продемонстрирован ниже, а на рисунке 4 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 1 0 0 0 1 1 0 0 1 0 1 0 1 1 0 0 1 1 1 0 1 1 0 0 0 0 1 0 1 1 0 1 0 1 0
0 1 1 0 0 1 0 1 0 1 1 0 1 0 1 1 0 1 1 0 0 0 0 1 0 1 1 1 0 1 0 0 0 1 1 0 0 1 0 1 0 1 1
1 0 0 1 0 0 1 1 0 1 0 0 1 0 1 1 0 1 1 1 0 0 1 1 0 0 0 0 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 0 1 1 0 0 0 1 0 1 1 0 1 0
1 1 1 0 1 1 1 0 1 1 0 1 1 1 0 0 1 1 0 0 0 1 1 0 1 0 0 1 0 1 1 0 0 1 0 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

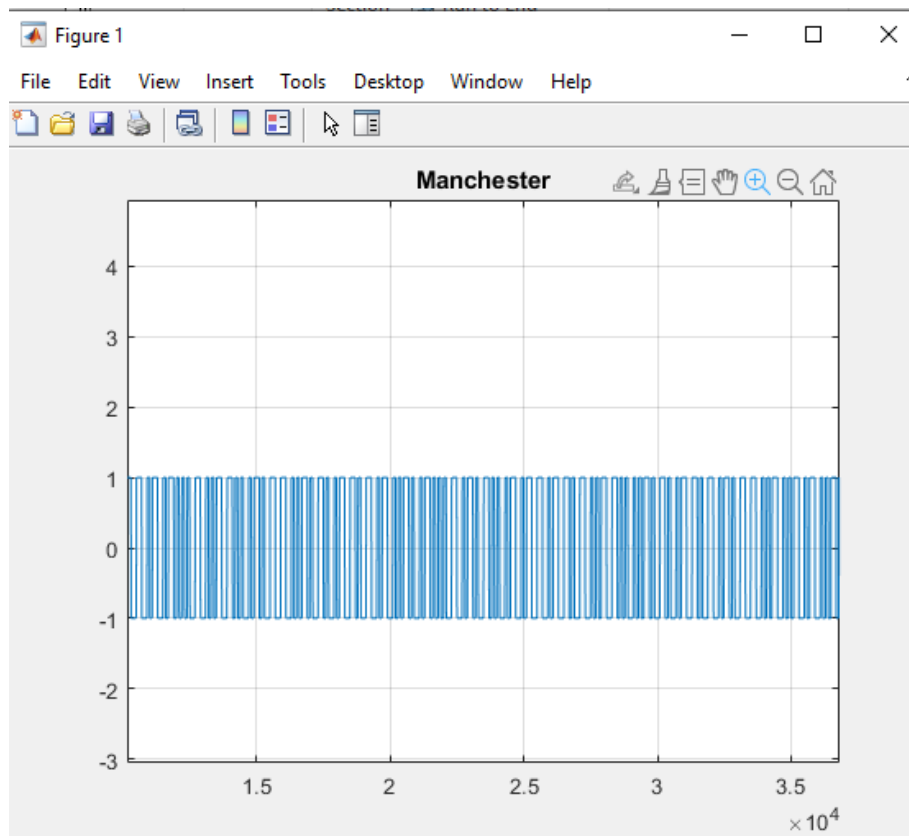


Рисунок 4 – Манчестерская кодировка

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*

```
clear all
close all
clc
data=[0 1 1 1 0 1 0 0 0 1 1 0 0 1 0 1 0 1 1 0 0 1 1 1 0 1 1 0 0 0 0 1 0 1 1 0 1 0 1 0
0 1 1 0 0 1 0 1 0 1 1 0 1 0 1 1 0 1 1 0 0 0 0 1 0 1 1 1 0 1 0 0 0 1 1 0 0 1 0 1 0 1 1
1 0 0 1 0 0 1 1 0 1 0 0 1 0 1 1 0 1 1 1 0 0 1 1 0 0 0 0 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 0 1 0 1 0
1 1 1 0 1 1 1 0 1 1 0 1 1 1 0 0 1 1 0 0 0 0 1];

% Параметры модуляции
A1 = 5; % Амплитуда
f1 = 5; % Частота для бита 0
% Частота для бита 1
Fs = 2^10; % Частота дискретизации
Tb = 1; % Длительность каждого бита
t = 0:1/Fs:Tb-1/Fs; % Временные отсчеты для одного бита

% Создание сигнала частотной модуляции
fm_wave = []; % Инициализация сигнала
for i = 1:length(data)
if data(i) == 1
signal = A1 * cos(2*pi*f1*t); % Генерация сигнала для бита 1
```



```

else
signal = 0 * cos(2*pi*f1*t); % Генерация сигнала для бита 0
end
fm_wave = [fm_wave, signal]; % Добавление сигнала к общему сигналу частотной
модуляции
end

% Временные отсчеты для всего сигнала
t_total = 0:1/Fs:(length(data)*Tb)-1/Fs;

% Построение графика частотной модуляции
plot(t_total, fm_wave);
grid on;
title('Амплитудная модуляция');
xlabel('Время (с)');
ylabel('Амплитуда');

```

Графики

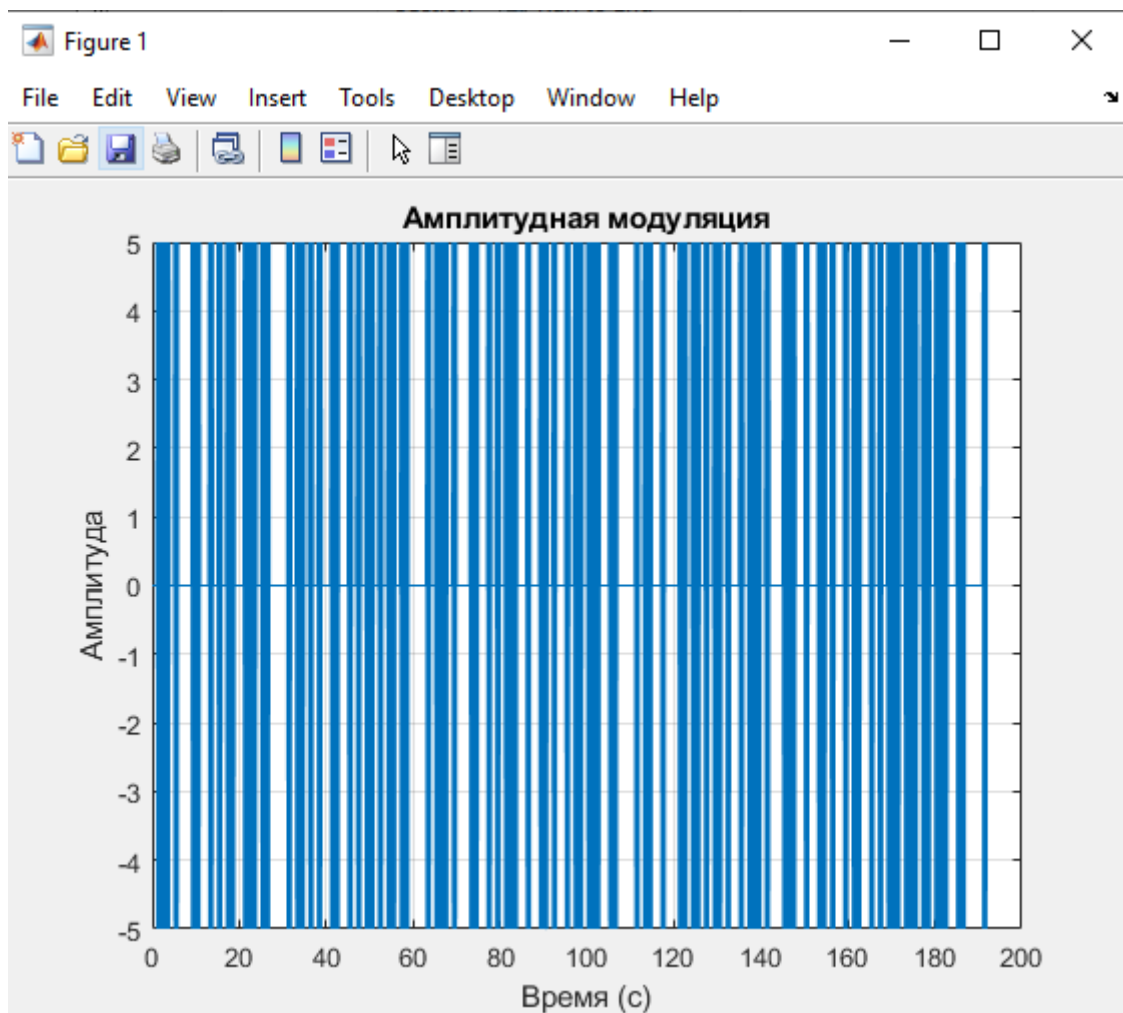


Рисунок 5 – Амплитудная модуляция

Теперь построим частотную модуляцию. Код программы продемонстрирован ниже. На рисунках 6 - 7 показан полученный график.

Код программы

Файл main.m

```
clear all
close all
clc

data=[0 1 1 1 0 1 0 0 0 1 1 0 0 1 0 1 0 1 1 0 0 1 1 1 0 1 1 0 0 0 0 1 0 1 1 0 1 0 1 0
0 1 1 0 0 1 0 1 0 1 1 0 1 0 1 1 0 1 1 0 0 0 0 1 0 1 1 1 0 1 0 0 0 1 1 0 0 1 0 1 0 1 1
1 0 0 1 0 0 1 1 0 1 0 0 1 0 1 1 0 1 1 1 0 0 1 1 0 0 0 0 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 0 1 0 0 0 1 1 0 1 1 0 0 1 0 1 0
1 1 1 0 1 1 1 0 1 1 0 1 1 1 0 0 1 1 0 0 0 1 1];

% Параметры модуляции
A = 5; % Амплитуда
f1 = 3; % Частота для бита 0
f2 = 1; % Частота для бита 1
Fs = 2^10; % Частота дискретизации
Tb = 1; % Длительность каждого бита
t = 0:1/Fs:Tb-1/Fs; % Временные отсчеты для одного бита

% Создание сигнала частотной модуляции
fm_wave = []; % Инициализация сигнала
for i = 1:length(data)
    if data(i) == 1
        signal = A * cos(2*pi*f2*t); % Генерация сигнала для бита 1
    else
        signal = A * cos(2*pi*f1*t); % Генерация сигнала для бита 0
    end
    fm_wave = [fm_wave, signal]; % Добавление сигнала к общему сигналу частотной
    модуляции
end

% Временные отсчеты для всего сигнала
t_total = 0:1/Fs:(length(data)*Tb)-1/Fs;

% Построение графика частотной модуляции
plot(t_total, fm_wave);
grid on;
title('Частотная модуляция');
xlabel('Время (с)');
ylabel('Амплитуда');
```

Графики

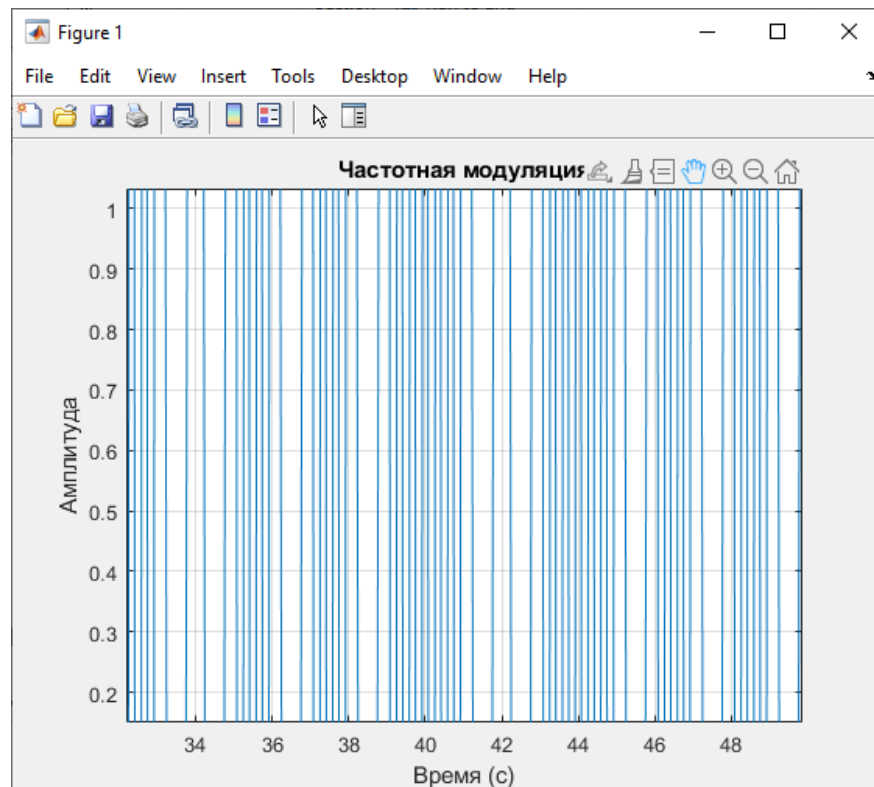


Рисунок 6 – Частотная модуляция

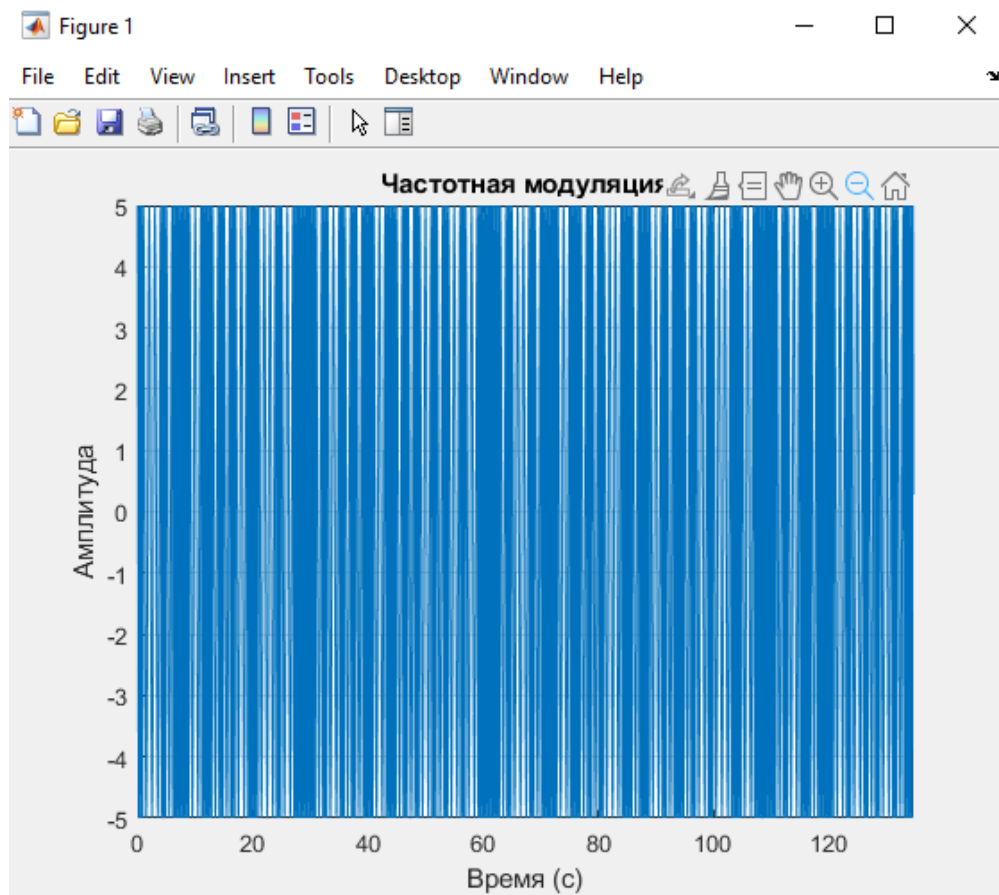


Рисунок 7 – Частотная модуляция

Наконец, построим фазовую модуляцию. Код программы продемонстрирован ниже. На рисунках 8 – 9 показан полученный график.

Код программы

Файл *main.m*

```
clear all
close all
clc

data=[0 1 1 1 0 1 0 0 0 1 1 0 0 1 0 1 0 1 1 0 0 1 1 1 0 1 1 0 0 0 0 1 0 1 1 0 1 0 1 0
0 1 1 0 0 1 0 1 0 1 1 0 1 0 1 1 0 1 1 0 0 0 0 1 0 1 1 1 0 1 0 0 0 1 1 0 0 1 0 1 0 1 1
1 0 0 1 0 0 1 1 0 1 0 0 1 0 1 1 0 1 1 1 0 0 1 1 0 0 0 0 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 0 1 0 1 1 0 0 1 0 1 0
1 1 1 0 1 1 1 0 1 1 0 1 1 1 0 0 1 1 0 0 0 0 1];

% Параметры модуляции
A = 5; % Амплитуда
f1 = 1; % Частота для бита 0
Fs = 2^10; % Частота дискретизации
Tb = 1; % Длительность каждого бита
t = 0:1/Fs:Tb-1/Fs; % Временные отсчеты для одного бита

% Создание сигнала частотной модуляции
fm_wave = []; % Инициализация сигнала
for i = 1:length(data)
    if data(i) == 1
        signal = A * cos(2*f1*(pi/2)*t); % Генерация сигнала для бита 1
    else
        signal = -A * cos(2*f1*pi*t); % Генерация сигнала для бита 0
    end
    fm_wave = [fm_wave, signal]; % Добавление сигнала к общему сигналу частотной
    модуляции
end

% Временные отсчеты для всего сигнала
t_total = 0:1/Fs:(length(data)*Tb)-1/Fs;

% Построение графика частотной модуляции
plot(t_total, fm_wave);
grid on;
title('Фазовая модуляция');
xlabel('Время (с)');
ylabel('Фаза');
```

Графики

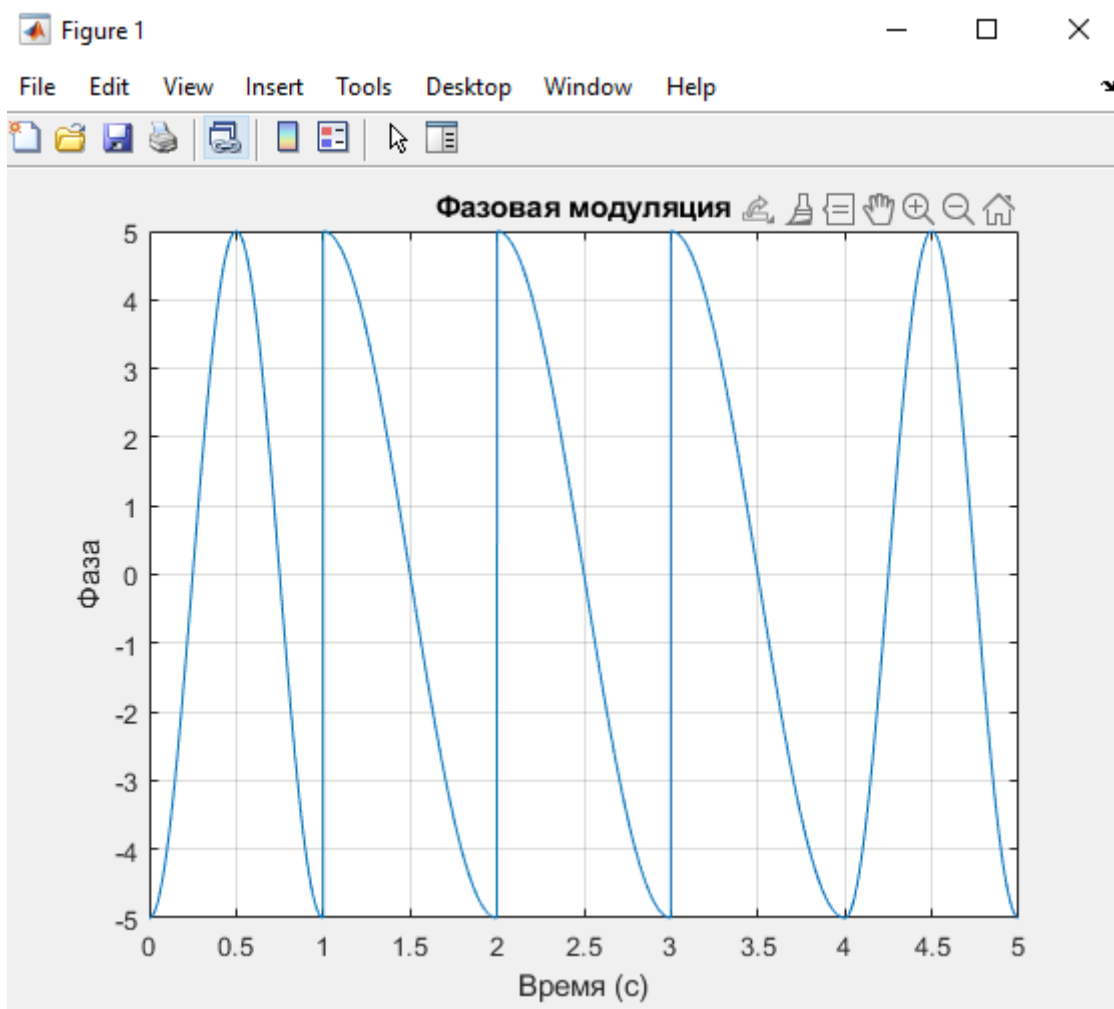


Рисунок 8 – Фазовая модуляция

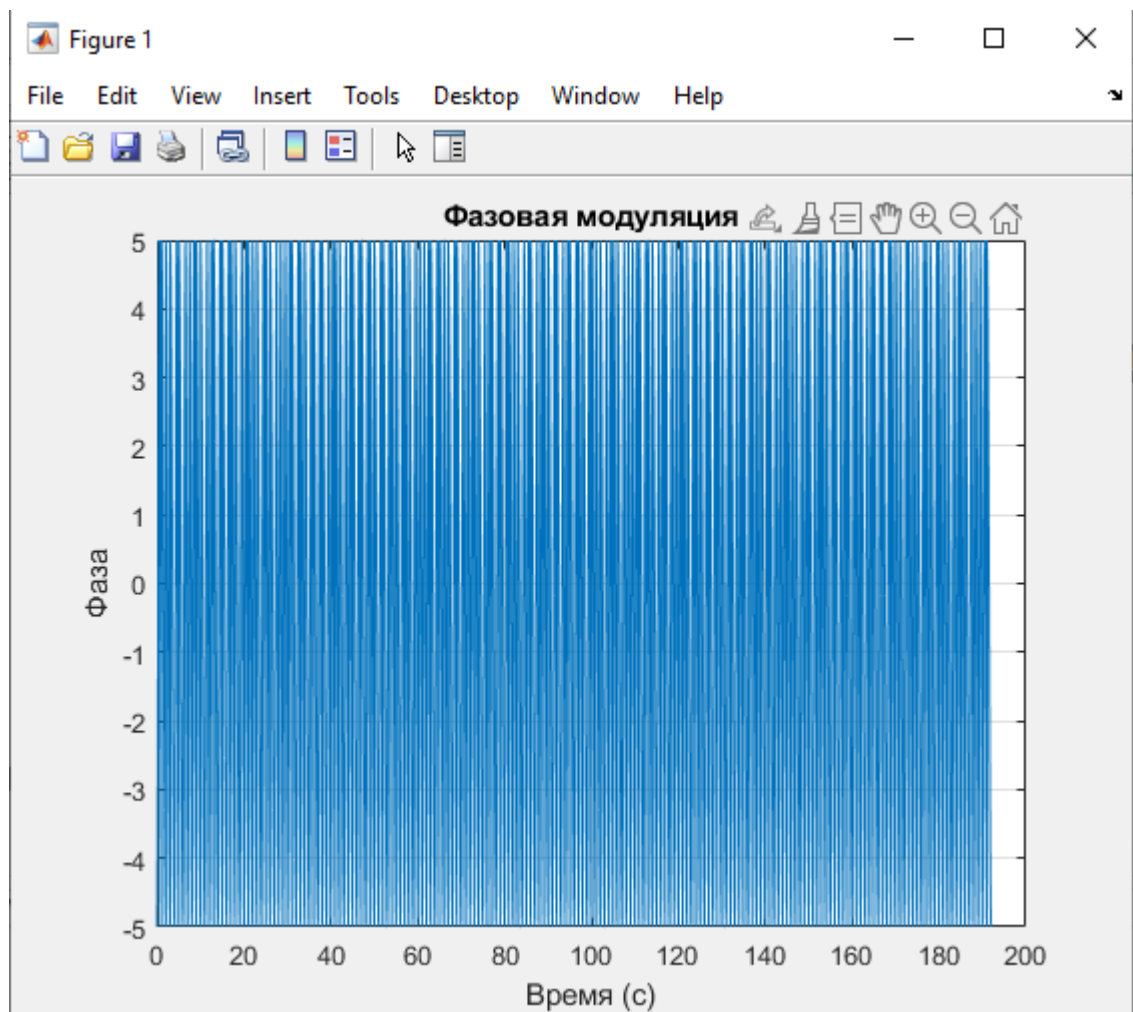


Рисунок 9 – Фазовая модуляция

Выводы

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