



Российский университет  
дружбы народов

1

# Лабораторная работа №1

МЕТОДЫ КОДИРОВАНИЯ И МОДУЛЯЦИЯ  
СИГНАЛОВ





- > Титульная страница
- > Структура
- > Представление выступающего
- > Прагматика
- > Цель выполнения лаб. работы
- > Задача выполнения лаб. работы
- > Результат выполнения лаб. работы





Выполнил: Юсупов Шухратджон Фирдавсович  
Факультет: Физико-математических и естественных наук  
Направление: Прикладная информатика (09.03.03)  
Группа: НПИБд-02-20  
Ст. Номер: 1032205329  
Почта Outlook: 1032205329@rudn.ru





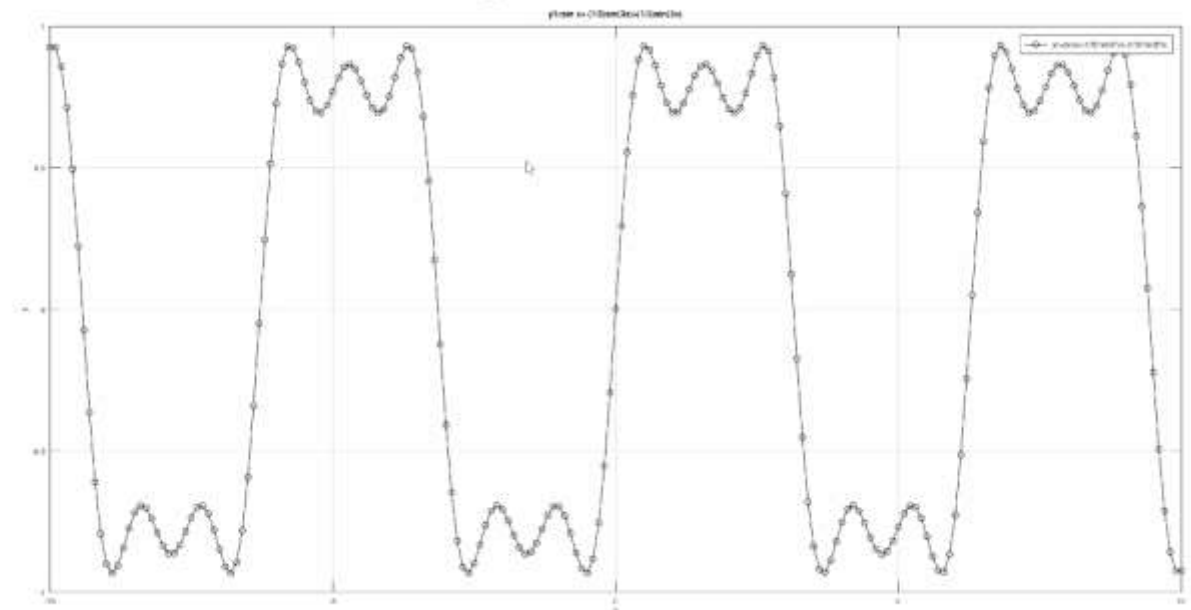
## Цель:

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.



plot\_sin.m

```
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, "-o"; y1=sin(x)+(1/3)*sin(3*x)+(1/5)*sin(5*x);","markersize",4)
7 % Отображение сетки на графике
8 grid on;
9 % Подпись оси X:
10 xlabel('x');
11 % Подпись оси Y:
12 ylabel('y');
13 % Название графика:
14 title('y1=sin x+ (1/3)sin(3x)+(1/5)sin(5x)');
15 % Экспорт рисунка в файл .eps:
16 print ("plot-sin.eps", "-mono", "-FARial:16", "-dps")
17 % Экспорт рисунка в файл .png:
18 print("plot-sin.png");
19
```

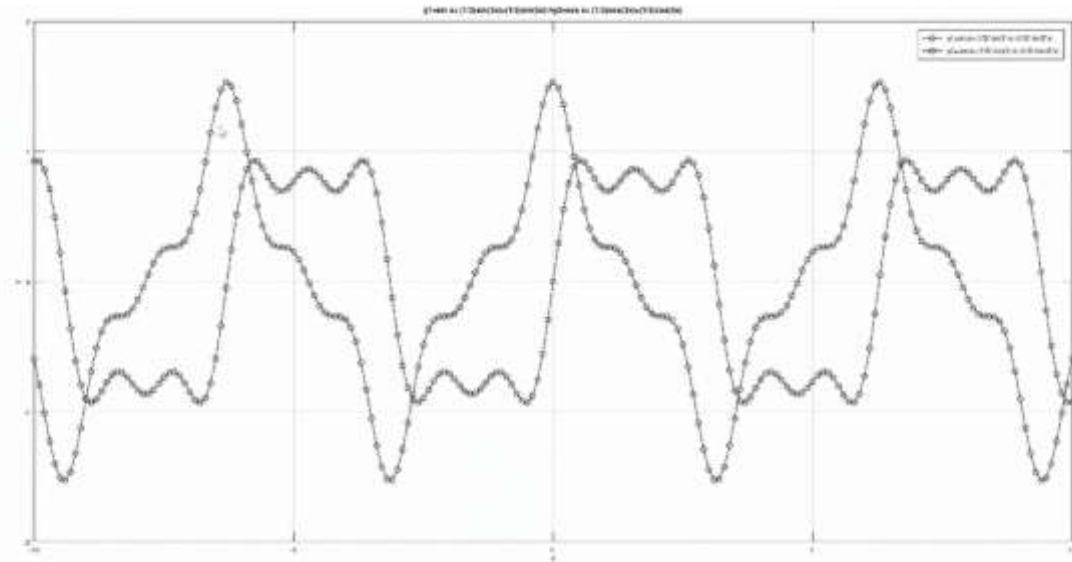


plot\_sincos.m

```

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 y2=cos(x)+1/3*cos(3*x)+1/5*cos(5*x);
6 % Построение графика функции:
7 plot(x,y1, "-ok; y1=sin(x)+(1/3)*sin(3*x)+(1/5)*sin(5*x);","markersize",4)
8 hold on;
9 plot(x,y2, "-ok; y2=cos(x)+(1/3)*cos(3*x)+(1/5)*cos(5*x);","markersize",4)
10 % Отображение сетки на графике
11 grid on;
12 % Подпись оси X:
13 xlabel('x');
14 % Подпись оси Y:
15 ylabel('y');
16 % Название графика:
17 title('y1=sin x+ (1/3)sin(3x)+(1/5)sin(5x)\ny2=cos x+ (1/3)cos(3x)+(1/5)cos(5x)');
18 % Экспорт рисунка в файл .eps:
19 print ("plot-sin.eps", "-mono", "-FARial:16", "-dops");
20 % Экспорт рисунка в файл .png:
21 print ("plot-sin.png");
22

```

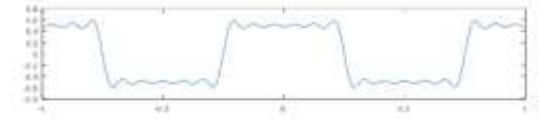
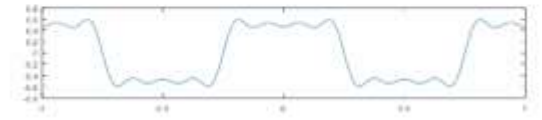
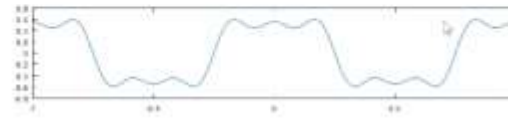
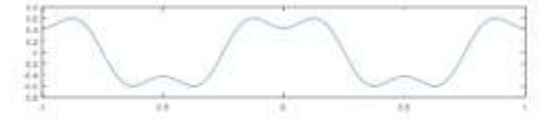
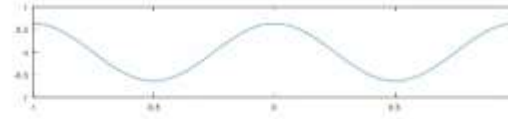




```

plot_sncos.m meandr.m
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 harmonica=cos(2 * pi * nh' * t/T);
17 % массив элементов ряда:
18 s1=harmonica.* repmat(Am',1,length(t));
19 % Суммирование ряда:
20 s2=cumsum(s1);
21 % Построение графиков:
22 for k=1:N
23     subplot(4,2,k)
24     plot(t, s2(k,:))
25 end

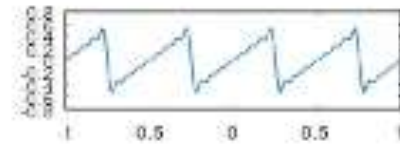
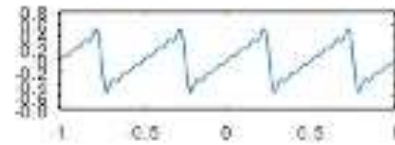
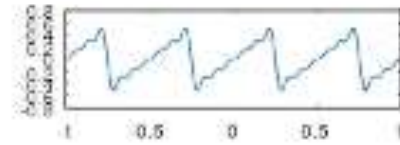
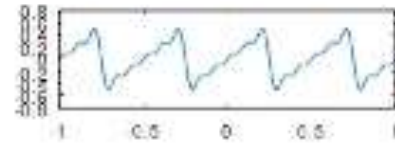
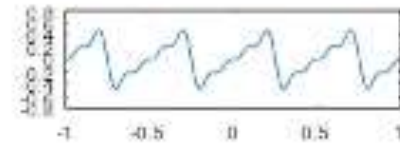
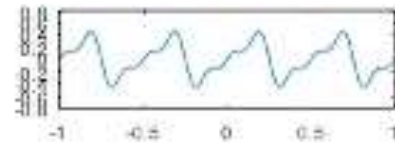
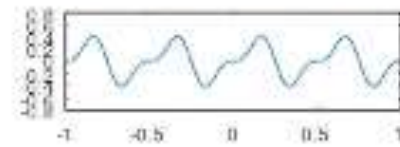
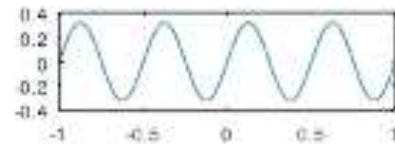
```



```

plot_sincos.m meandrsin.m plot_sin.m
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;
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 harmonica=sin(2 * pi * nh' * t/T);
17 % массив элементов ряда:
18 s1=harmonica.*repmat(Am',1,length(t));
19 % суммирование ряда:
20 s2=sum(s1);
21 % Построение графиков:
22 for k=1:N
23 subplot(4,2,k)
24 plot(t, s2(k,:))
25 print("meandr.png")

```

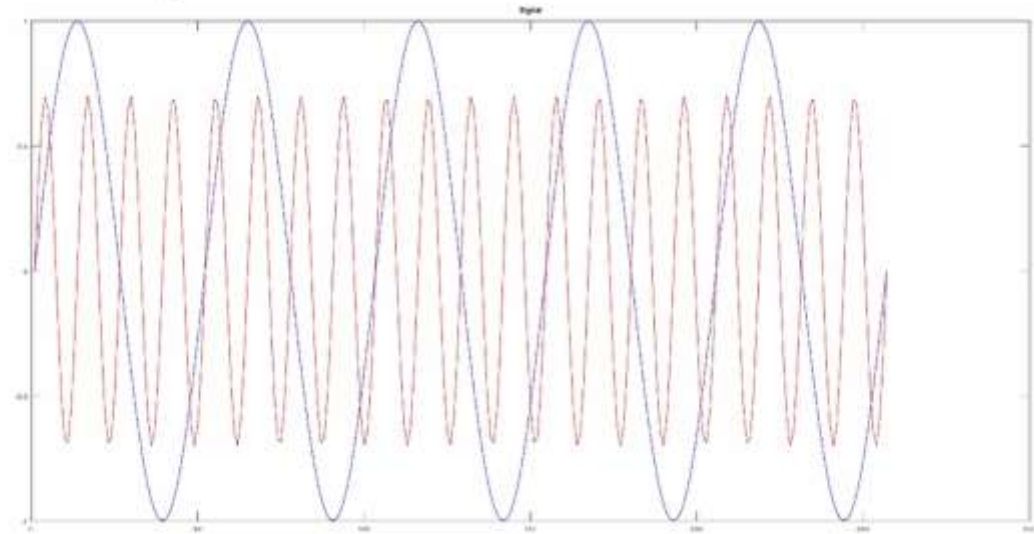




```

spectre.m
1 % spectre1/spectre.m
2 % Создание каталогов signal и spectre для размещения графиков:
3 mkdir 'signal';
4 mkdir 'spectre';
5 % Длина сигнала (s):
6 tmax = 0.5;
7 % Частота дискретизации (Гц) (количество отсчетов):
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 отсчетов 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');
26 % График 2-го сигнала:
27 hold on
28 plot(signal2,'r');
29 hold off
30 title('Signal');
31 % Экспорт графика в файл в каталоге signal:
32 print 'signal/spectre.png';
33

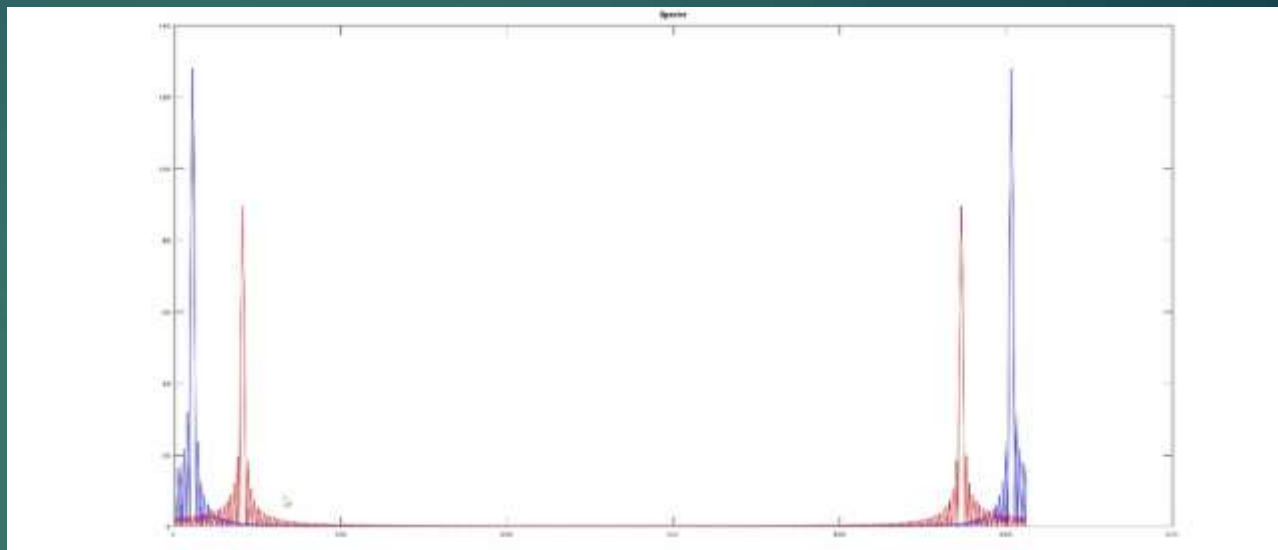
```



```

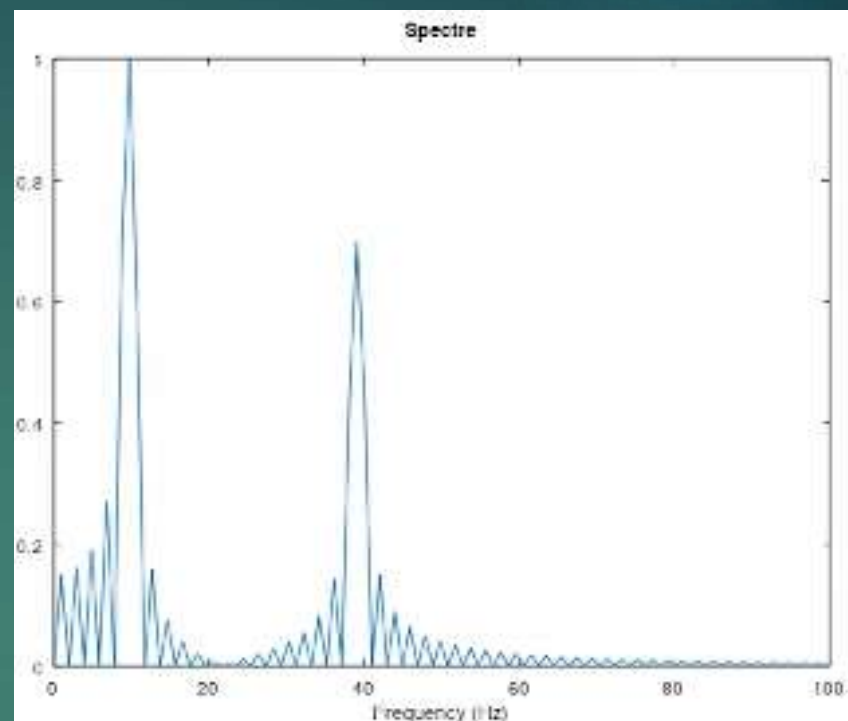
spectre.m
12 f2 = 40;
13 % Амплитуда первого сигнала:
14 a1 = 1;
15 % Амплитуда второго сигнала:
16 a2 = 0.7;
17 % Массив отсчётов времени:
18 t = 0:1./fd:tmax;
19 % Спектр сигнала:
20 fd2 = fd/2;
21 % Два сигнала разной частоты:
22 signal1 = a1*sin(2*pi*t*f1);
23 signal2 = a2*sin(2*pi*t*f2);
24 % График 1-го сигнала:
25 plot(signal1,'b');
26 % График 2-го сигнала:
27 hold on
28 plot(signal2,'r');
29 hold off
30 title('Signal');
31 % Экспорт графика в файл в каталоге signal:
32 print 'signal/spectre.png';
33 % Посчитаем спектр
34 % Амплитуда преобразования Фурье сигнала 1:
35 spectre1 = abs(fft(signal1,fd));
36 % Амплитуда преобразования Фурье сигнала 2:
37 spectre2 = abs(fft(signal2,fd));
38 % Построение графиков спектров сигналов:
39 plot(spectre1,'b');
40 hold on
41 plot(spectre2,'r');
42 hold off
43 title('Spectre');
44 print 'spectre/spectre.png';

```



```
spectre_sum.m
1 % spectr_sum/spectre_sum.m
2 % Создание каталогов signal и spectre для размещения графиков:
3 mkdir 'signal';
4 mkdir 'spectre';
5 % Длина сигнала (с):
6 tmax = 0.5;
7 % Частота дискретизации (Гц) (количество отсчетов):
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 % Сократ сигнал:
18 fd2 = fd/2;
19 % Сумма двух сигналов (синусоиды) разной частоты:
20 % Массив отсчетов времени:
21 t = 0:1./fd:tmax;
22 signal1 = a1*sin(2*pi*t*f1);
23 signal2 = a2*sin(2*pi*t*f2);
24 signal = signal1 + signal2;
25 plot(signal);
26 title('Signal');
27 print 'signal/spectre_sum.png';
28 % Подсчет спектра:
29 % Амплитуда преобразования Фурье сигнала:
30 spectre = fft(signal,fd);
31 % Сетка частот
32 f = 1000*(0:fd2)./(2*fd);
33 % Нормировка спектра по амплитуде:
34 spectre = 2*abs(spectre)/length(spectre);/fd2;
```

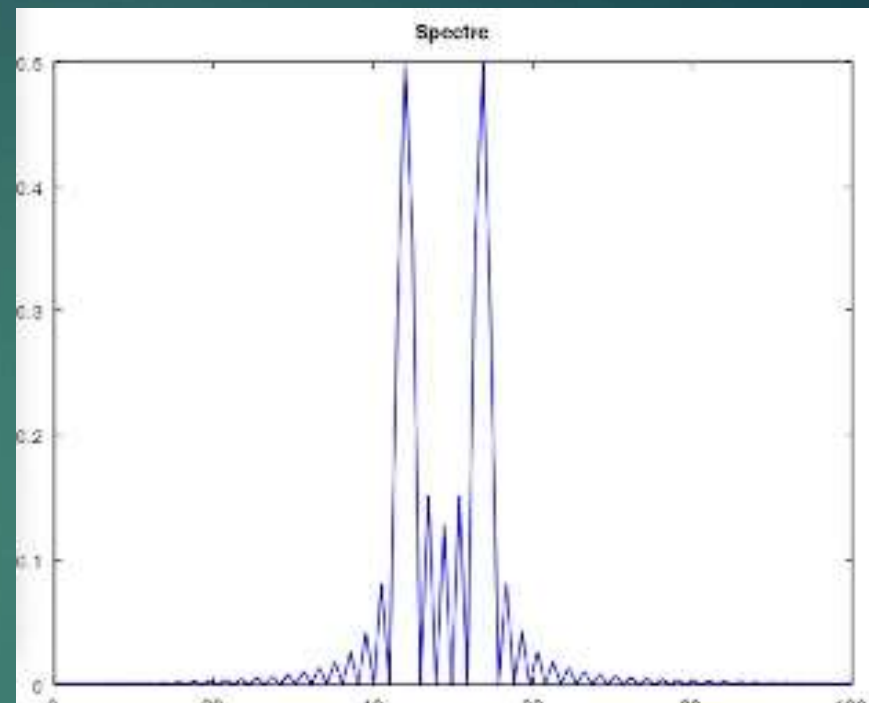
строка: 17 столбец: 13 кодировка: UTF-8 конец строки: CRLF



```

1 % modulation/am.m
2 % Создание начальных signal и аргумента для размещения графиков:
3 mkdir 'signal';
4 mkdir 'spectre';
5 % Модуляция синусоид с частотами 50 и 5
6 % Длительность сигнала (с)
7 tmax = 0.5;
8 % Частота дискретизации (Гц) (количество отсчётов)
9 fd = 512;
10 % Частота сигнала (Гц)
11 f1 = 5;
12 % Частота несущей (Гц)
13 f2 = 50;
14 % Спектр сигнала
15 fd2 = fd/2;
16 % Построение графиков двух сигналов (синусоиды)
17 % разной частоты
18 % Массив отсчётов времени:
19 t = 0:1./fd:tmax;
20 signal1 = sin(2*pi*t*f1);
21 signal2 = sin(2*pi*t*f2);
22 signal = signal1 .* signal2;
23 plot(signal, 'b');
24 hold on
25 % Построение огибающей:
26 plot(signal1, 'r');
27 plot(-signal1, 'r');
28 hold off
29 title('Signal');
30 print 'signal/am.png';
31 % Расчет спектра:
32 % Амплитуда преобразования Фурье-сигнала:
33 spectre = fft(signal,fd);
34 % Масштаб частот:

```

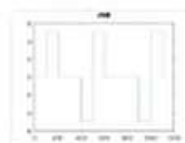


```

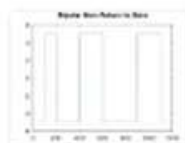
main.m x mappointwave.m x unipolar.m x ami.m x bipolarnrz.m x bipolarrrz.m x manchester
70 title('Manchester');
71 print 'sync/manchester.png';
72 % Дифференциальное манчестерское кодирование
73 wave=diffmanc(data_sync);
74 plot(wave)
75 title('Differential Manchester');
76 print 'sync/diffmanc.png';
77 % Униполярное кодирование:
78 wave=unipolar(data_spectre);
79 spectre=calcspectre(wave);
80 title('Unipolar');
81 print 'spectre/unipolar.png';
82 % Кодирование RMI:
83 wave=ami(data_spectre);
84 spectre=calcspectre(wave);
85 title('RMI');
86 print 'spectre/ami.png';
87 % Кодирование NRZ:
88 wave=bipolarnrz(data_spectre);
89 spectre=calcspectre(wave);
90 title('Bipolar Non-Return to Zero');
91 print 'spectre/bipolarnrz.png';
92 % Кодирование RZ:
93 wave=bipolarrrz(data_spectre);
94 spectre=calcspectre(wave);
95 title('Bipolar Return to Zero');
96 print 'spectre/bipolarrrz.png';
97 % Манчестерское кодирование:
98 wave=manchester(data_spectre);
99 spectre=calcspectre(wave);
100 title('Manchester');
101 print 'spectre/manchester.png';
102 % Дифференциальное манчестерское кодирование:
103 wave=diffmanc(data_spectre);

```

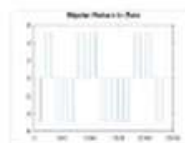
строка: 86 столбец: 1 кодировка: UTF-8 конец строки: CRLF



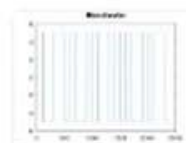
ami



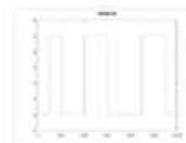
bipolarnrz



bipolarrrz



manchester



unipolar