



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ

Информатика и системы управления

КАФЕДРА

Проектирования и технологии производства ЭА

ОТЧЕТ

По семинару №5

по курсу

Киберфизические системы

Студент

Фан К.

(Подпись, дата)

(И.О.Фамилия)

Руководитель задания

Леонидов В. В.

(Подпись, дата)

(И.О.Фамилия)

Москва, 2023

Задание 1: Реализация амплитудной манипуляции (ASK) в среде MATLAB

```
clear, clc, close all
fs = 10000;
ts = 0: 1/fs: 0.2 - 1/fs;
N = length(ts);

%% несущий сигнал
fc = cos(2*pi*500*ts);
fc = awgn(fc,30); % Добавить белый Гауссов шум

%% модулирующий сигнал
code = [1 0.1 0.1 1 1 0.1 0.1 1 0.1 1]; % кодируемая последовательность
ns_bit = 200; % длина одного бита в отсчётах

am = repelem(code , ns_bit); % Модулирующий сигнал

%% Амплитудная манипуляция (ASK)
x = fc.*am;
figure;
plot(ts,x,'LineWidth',1), grid on, hold on
plot(ts, am , 'LineWidth',2 ) , grid on, hold on
legend({'Модулированный сигнал'; 'Модулирующий сигнал'})
title('Амплитудная манипуляция (ASK)')
h = hilbert(x);
scatterplot(h,ns_bit,round(ns_bit/2)) , grid on

%% ASK демодуляция
ah = abs(h);
dem = ah >= 0.5;
figure;
subplot(2,1,1)
plot(ts,x,'LineWidth',0.5), grid on, hold on
plot(ts, ah , 'LineWidth',2 ) , grid on
legend({'Модулированный сигнал'; 'Демодулированный сигнал'})
title('ASK демодуляция')
subplot(2,1,2)
plot(ts, dem ) , grid on
title('Код демодулированного сигнала')
```

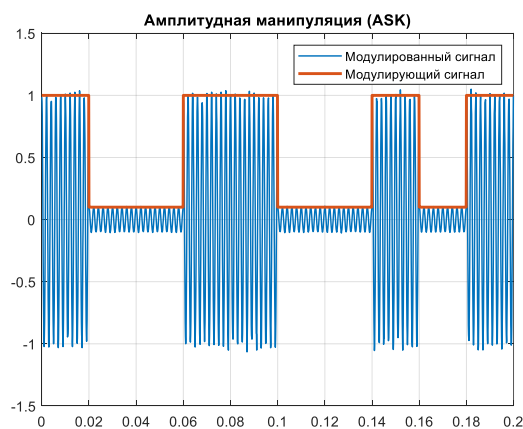


Рисунок 1: Амплитудная манипуляция (ASK)

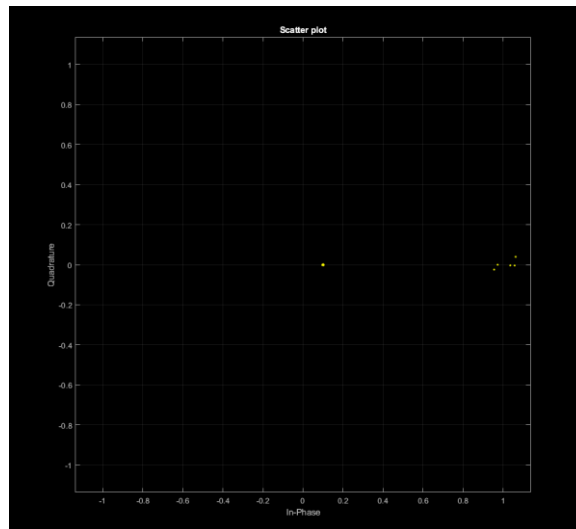


Рисунок 2: Сигнальное созвездие ASK-модуляции

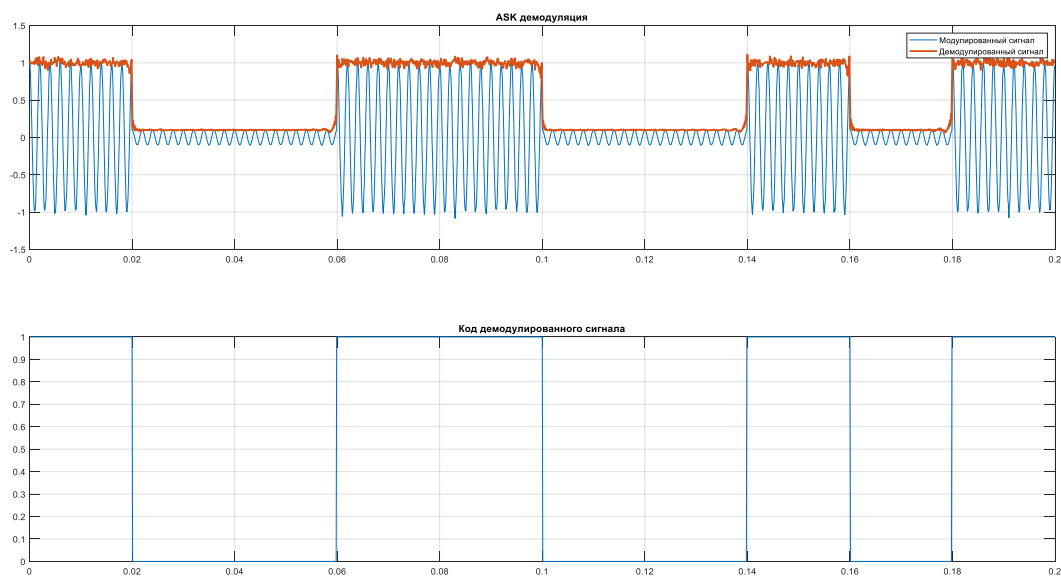


Рисунок 3: Демодуляция ASK-модулированного сигнала

Вывод:

В данном задании рассмотрим амплитудную манипуляцию (ASK). Модулирующий сигнал и модулированный сигнал ASK показаны на рисунке 1. Логическому “нулю” будет соответствовать амплитуда 0.1, логической “единице” -1. Из рисунка 2 (сигнальное созвездие) видно, что два скопления точек вдоль действительной оси: одно вокруг значения 0.1, второе — вокруг 1. Демодуляция ASK-модулированного сигнала показан на рисунке 3. Код демодулированного сигнала совпадают с переданным кодом.

Задание 2: Реализация двоичной фазовой манипуляции (BPSK) в среде MATLAB

```
clear, clc, close all

fs = 10000;
ts = 0: 1/fs: 0.2-1/fs;
N = length(ts);

%% несущий сигнал
fc = cos(2*pi*250*ts);
fc = awgn(fc,35); % Добавить белый Гауссов шум

%% модулирующий сигнал
code = [1 -1 -1 1 1 -1 -1 1 -1 1]; % кодируемая последовательность
ns_bit = 200; % длина одного бита в отсчётах

pm = repelem(code , ns_bit); % Модулирующий сигнал

%% Фазовая манипуляция (BPSK)
x = fc.*pm;
figure;
plot(ts,x,'LineWidth',1), grid on, hold on
plot(ts, pm , 'LineWidth',2 ) , grid on, hold on
legend({'Модулированный сигнал'; 'Модулирующий сигнал'})
title('Фазовая манипуляция (BPSK)')
h = hilbert(x);
scatterplot(h,ns_bit,round(ns_bit/2)) , grid on

%% BPSK демодуляция
y = x.*fc;
figure;
subplot(3,1,1)
plot(ts,y), grid on
title('Модулированный сигнал, умноженный на несущую')

demod_filter = bpsk_filer_lowpass;
z = filter(demod_filter.Numerator, 1,y);

subplot(3,1,2)
plot(ts,z), grid on
title('Сигнал y, прошедший через ФНЧ')

dem = z >= 0.1;
subplot(3,1,3)
plot(ts, dem ), grid on
title('Демодулированный сигнал')
```

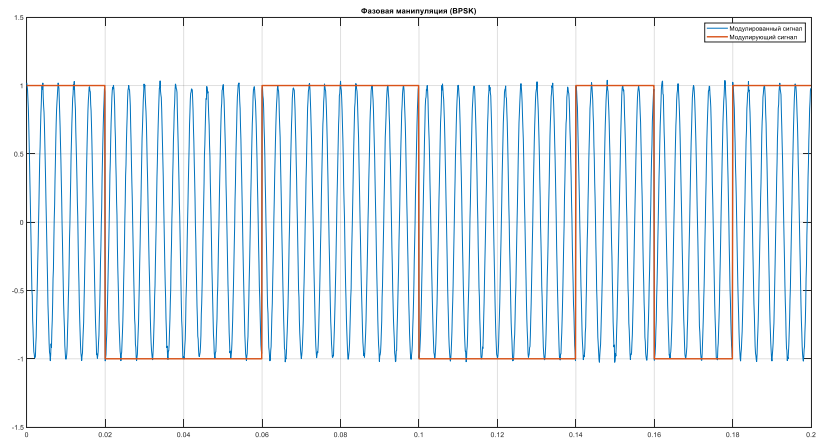


Рисунок 4: Фазовая манипуляция (BPSK)

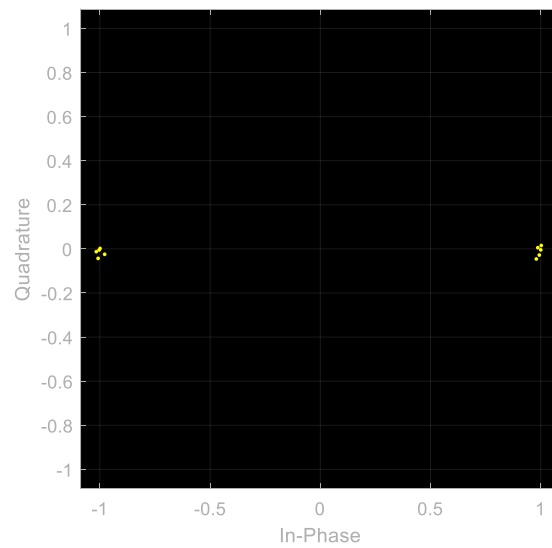


Рисунок 5: Сигнальное созвездие BPSK -модуляции

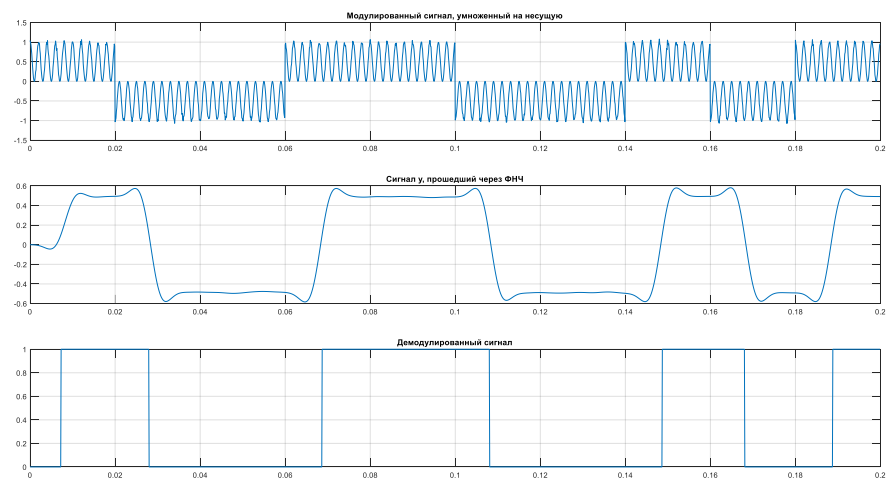


Рисунок 6: Демодуляция BPSK -модулированного сигнала

Вывод:

В данном задании рассмотрим простой вид фазовой манипуляции - двоичная фазовая манипуляция (BPSK). Модулирующий сигнал и модулированный сигнал BPSK показаны на рисунке 4. Каждое изменение логического уровня модулирующего сигнала приводит к скачкообразному изменению фазы несущей частоты на 180° . Из рисунка 5 (сигнальное созвездие) видно, что имеет два скопления точек вокруг $(-1,0)$ и $(1,0)$, которые соответствуют повороту вектора единичной длины на 0° и 180° вокруг начала координат. Демодуляция BPSK -модулированного сигнала показан на рисунке 6. С помощью ФНЧ удаляем высокочастотные составляющие частоты при демодуляции BPSK. Код демодулированного сигнала совпадают с переданным кодом.

Задание 3: Реализация 8-фазовой манипуляции (8-PSK) в среде MATLAB

```
clear, clc, close all

fs = 10000;
ts = 0: 1/fs: 0.2-1/fs;
N = length(ts);

%% 8PSK
fc = 1000;
M = 8;
Nd = 100;
bit_size = N / Nd;% длина одного бита в отсчётах
data = randi([0 M-1],Nd,1).'; % количество отсчётов на бит

% Преобразование Gray в Natural Binary для PSK, вместо pskmod
bin_gray = de2bi(data, 3, 'right-msb');
bin_natural = zeros(size(bin_gray));
for i = 1:Nd
    for j = 3:-1:1
        if j == 3
            bin_natural(i, 3) = bin_gray(i, 3);
        else
            bin_natural(i, j) = bitxor(bin_natural(i, j+1), bin_gray(i, j));
        end
    end
end
end
pdata = exp(1i*2*pi/M *bi2de(bin_natural, 'right-msb')).'; % Массив комплексных чисел
pdata = awgn(pdata,20);% Добавить белый Гауссов шум
scatterplot(pdata), grid on;

qmod = repelem(pdata,bit_size);
i = real(qmod).*cos(2*pi*fc*ts);
q = imag(qmod).*sin(2*pi*fc*ts);
y= i +q;

figure
subplot(3,1,1)
plot(real(qmod))
title('Действительная часть сигнала')
subplot(3,1,2)
plot(imag(qmod))
title('Мнимая часть сигнала')
subplot(3,1,3)
plot(y)
title('Модулированный сигнал')

%% 8PSK демодуляция

io = 2*y.*cos(2*pi*fc*ts);
qo = 2*y.*sin(2*pi*fc*ts);

filer_8psk = psk_filter_lowpass;
iof = round(conv(filer_8psk.Numerator,io));
qof = round(conv(filer_8psk.Numerator,qo));

figure
subplot(2,1,1)
plot(iof)
title('Синфазная составляющая демодулированного сигнала после ФНЧ')
```

```

subplot(2,1,2)
plot(qof)
title('Квадратурная составляющая демодулированного сигнала после ФНЧ')

of = complex(iof, qof);
fir_delay = round(length(filer_8psk.Numerator)/2);
of_dec = of(fir_delay+round(bit_size/2):bit_size:length(of)-fir_delay);

% Преобразование Natural Binary в Gray для демодуляция PSK, вместо pskdemod
edges = zeros(size(of_dec));
data_received = zeros(size(of_dec));
for i = 1: length(of_dec)
    edges(i) = angle(of_dec(i));
    if edges(i) < 0
        edges(i) = edges(i) + 2*pi;
    end
    data_received(i) = bitxor(edges(i)/(2*pi/M), floor(edges(i)/(2*pi/M)/2));
end

% Сравниваем результаты
figure
plot(data, 'b-'), grid on, hold on
plot(data_received, 'x', 'LineWidth', 2)
title('Сравнение закодированных (переданных) и декодированных (принятых) данных')
legend({'Переданные данные'; 'Принятые данные'})

```

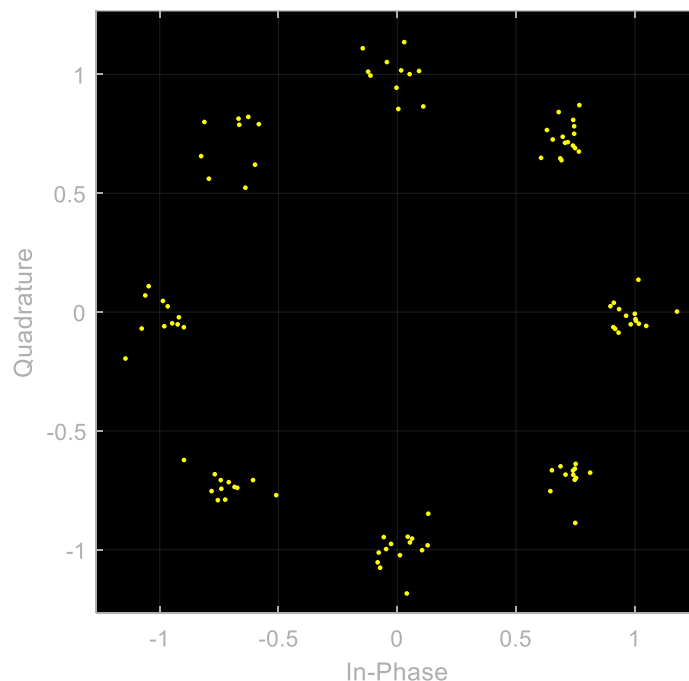


Рисунок 7: Сигнальное созвездие 8-PSK -модуляции

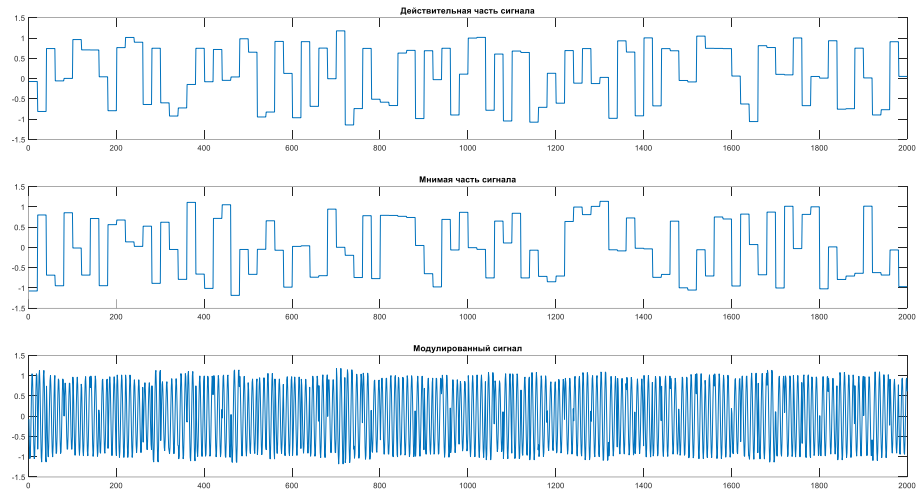


Рисунок 8: Фазовая манипуляция (8-PSK)

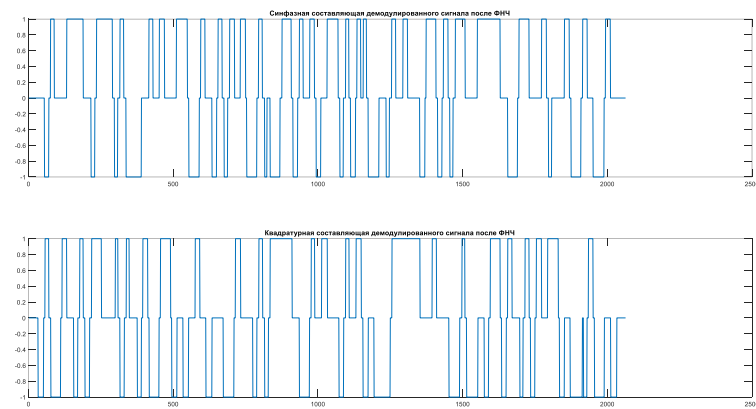


Рисунок 9: Синфазная и квадратурная составляющие демодулированного 8-PSK-сигнала

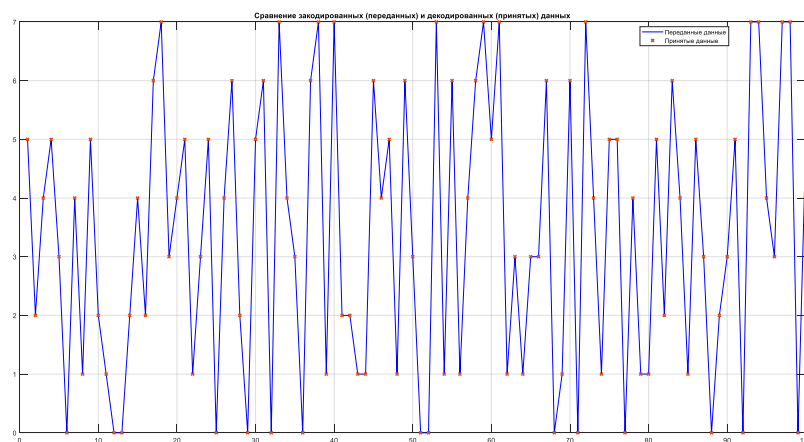


Рисунок 10: Сравнение переданных и принятых данных

Вывод:

В данном задании рассмотрим 8-фазовую манипуляцию (8-PSK). Из рисунка 7 (сигнальное созвездие) видно, что имеет 8 скопления точек, каждая из которых соответствует положению единичного вектора, начало которого соответствует началу координат, при его повороте с шагом 45° . Модулированный сигнал 8-PSK показан на рисунке 8. С помощью ФНЧ удаляем высокочастотные составляющие частоты при демодуляции 8-PSK. Из результатов сравнения переданные данные и принятые данные после демодуляции совпадают.

Задание 4: Реализация 64-квадратурной манипуляции (64-QAM) в среде MATLAB

```
clear, clc, close all

fs = 10000;
ts = 0: 1/fs: 0.5-1/fs;
N = length(ts);

%% 64-QAM
fc = 1000;
M = 64;
Nd = 100;
bit_size = N / Nd;% длина одного бита в отсчётах
data = randi([0 M-1],Nd,1).'; % количество отсчётов на бит

% Создание созвездий для 64-QAM
x = (0:M-1);
y = bitxor(x, floor(x/2)); % преобразование в код Грея

qamSymbols = zeros(1, M);
index = 1;
i_indexes = -7:2:7;
q_indexes = -7:2:7;
for i = -7:2:7
    if mod(i, 4) == 1
        j_range = flip(q_indexes);
    else
        j_range = q_indexes;
    end

    for j = j_range
        qamSymbols(index) = i + j*1i;
        index = index + 1;
    end
end
qam_64_table = zeros(2,M);
qam_64_table(1,:)=y;
qam_64_table(2,:)=qamSymbols;

% преобразование data в массив значений 64-QAM сигнала
qdata = zeros(1,Nd);
for i= 1:Nd
    qdata(i) = qam_64_table(2,y==data(i));
end
% qdata = qammod(data,M);
qdata = awgn(qdata,20);% Добавить белый Гауссов шум
scatterplot(qdata), grid on;

qmod = repelem(qdata,bit_size);
i = real(qmod).*cos(2*pi*fc*ts);
q = imag(qmod).*sin(2*pi*fc*ts);
y= i +q;

figure
subplot(3,1,1)
plot(real(qmod))
title('Действительная часть сигнала')
subplot(3,1,2)
plot(imag(qmod))
title('Мнимая часть сигнала')
```

```

subplot(3,1,3)
plot(y)
title('Модулированный сигнал')

%% 64-QAM демодуляция

io = 2*y.*cos(2*pi*fc*ts);
qo = 2*y.*sin(2*pi*fc*ts);

filer_64qam = qam_filter_lowpass;
iof = round(conv(filer_64qam.Numerator,io));
qof = round(conv(filer_64qam.Numerator,qo));

figure
subplot(2,1,1)
plot(iof)
title('Синфазная составляющая демодулированного сигнала после ФНЧ')
subplot(2,1,2)
plot(qof)
title('Квадратурная составляющая демодулированного сигнала после ФНЧ')

of = complex(iof, qof);
fir_delay = round(length(filer_64qam.Numerator)/2);
of_dec = of(fir_delay+round(bit_size/2):bit_size:length(of)-fir_delay);
data_received = zeros(size(of_dec));

for i=1:length(of_dec)
    i_val = real(of_dec(i));
    q_val = imag(of_dec(i));
    [~,i_index] = min(abs(i_val-i_indexs));
    [~,q_index] = min(abs(q_val-q_indexs));
    qamSymbol = i_indexs(i_index) + 1i*q_indexs(q_index);
    data_received(i) = qam_64_table(1,qamSymbols == qamSymbol);
end

% y =qamdemod(of_dec,M);

% Сравниваем результаты
figure
plot(data,'b-'), grid on, hold on
plot(data_received,'x','LineWidth',2)
title('Сравнение закодированных (переданных) и декодированных (принятых) данных')
legend({'Переданные данные'; 'Принятые данные'})

```

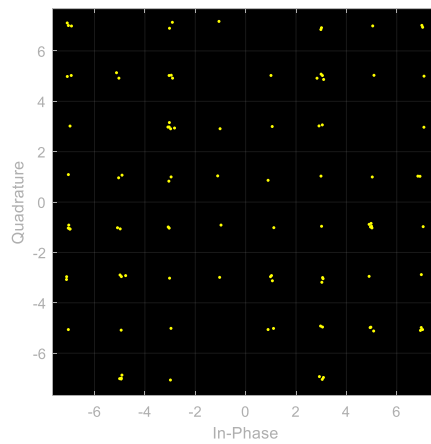


Рисунок 11: Сигнальное созвездие сигнала с 64-QASK

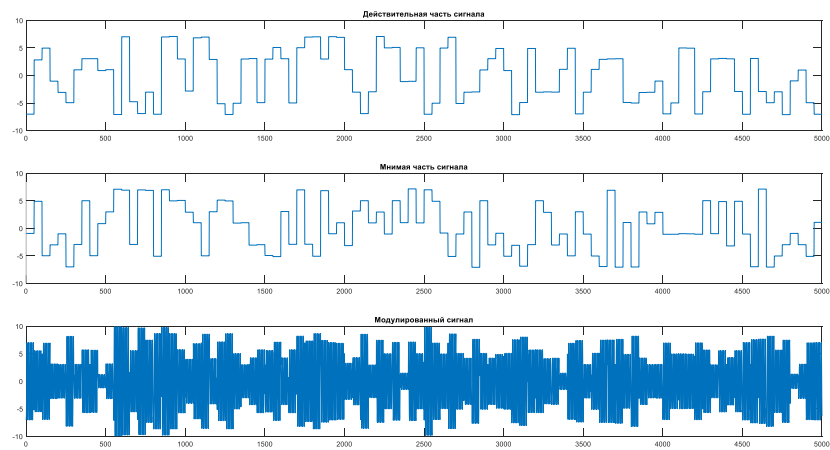


Рисунок 12: Синфазная, квадратурная составляющие и 64-QASK-модулированный сигнал

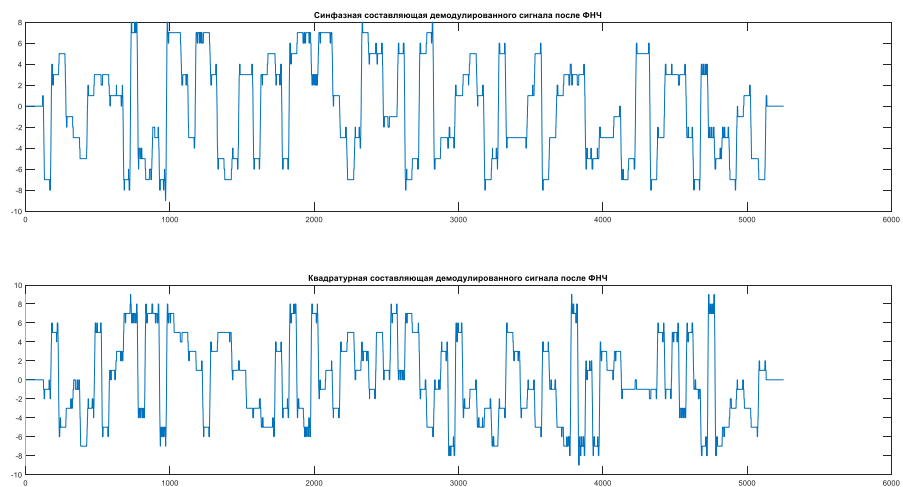


Рисунок 13: Синфазная и квадратурная составляющие демодулированного 64-ASK-сигнала

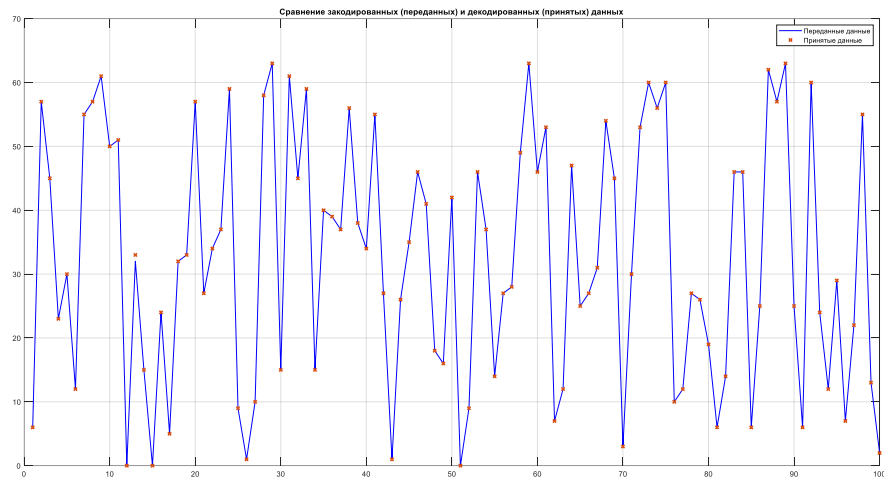


Рисунок 14: Сравнение переданных и принятых данных

Вывод:

В данном задании рассмотрим 64- квадратурную манипуляцию (64-QASK). QASK — это вид манипуляции, при которой скачкообразно изменяется как амплитуда, так и фаза несущего сигнала. Сигнальное созвездие сигнала с 64-QASK, которая состоит из 64 групп точек, показан на рисунке 11. Модулированный сигнал 64-QASK показан на рисунке 12.

С увеличением порядка модуляции M (в этом задании $M=64$) точки созвездия располагаются всё ближе и ближе друг другу, что может привести к ошибкам декодирования такого сигнала, если он сильно зашумлён. С помощью ФНЧ удаляем высокочастотные составляющие частоты при демодуляции 64-QASK. Из результатов сравнения переданные данные и принятые данные после демодуляции почти совпадают.

Задание 5: Реализация частотной манипуляции (FSK) в среде MATLAB

```
clear, clc, close all

fs = 10000;
ts = 0 : 1/fs : 0.1-1/fs;
N = length(ts);

fc = 1000;
h = 6;
T = 0.01;

f0 = fc+h/(2*T); % частота лог. "0"
f1 = fc-h/(2*T); % частота лог. "1"

n_for_bit = T*fs; % длина одного бита в отсчётах

code = [1 0 1 1 0 0 1 0 0 1];

%% модулирующий сигнал
fm = repelem(code , n_for_bit); % Модулирующий сигнал

%% Частотная манипуляция (FSK)
x0 = cos(2*pi*f0*ts);
x1 = cos(2*pi*f1*ts);
x = zeros(1,N);
for i = 1:N
    if fm(i) == 0
        x(i) = x0(i);
    else
        x(i) = x1(i);
    end
end

figure;
plot(ts, x), grid on, hold on
plot(ts, fm,'LineWidth',2)
title ('Частотная манипуляция (FSK)')
legend({'Модулированный сигнал'; 'Модулирующий сигнал'})

%% FSK демодуляция

y0 = x.*x0;
y1 = x.*x1;

y = y1-y0;

figure
subplot(3,1,1)
plot(ts, y0), grid on
subplot(3,1,2)
plot(ts, y1), grid on
subplot(3,1,3)
plot(ts, y), grid on
title('Сигнал y = y0-y1')

fltr = fsk_filter_lowpass;
z = filter(fltr.Numerator,1,y);

dem = z >= 0.1;
```

```

figure
subplot(2,1,1)
plot(ts,z), grid on
title('Сигнал y, прошедший через ФНЧ')
subplot(2,1,2)
plot(ts,dem), grid on
title('Код демодулированного сигнала')

```

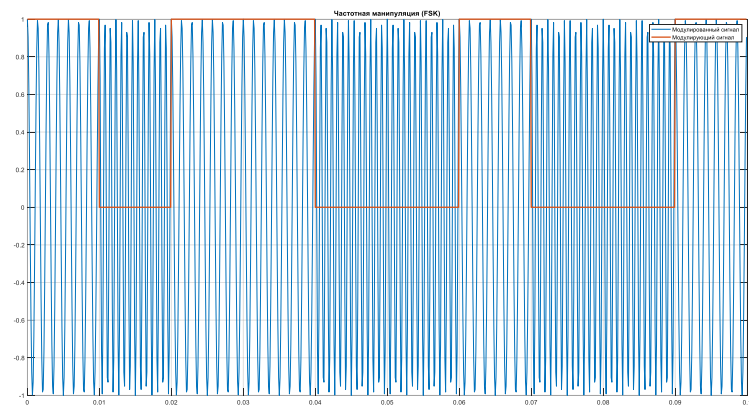


Рисунок 15: Частотная манипуляция (FSK)

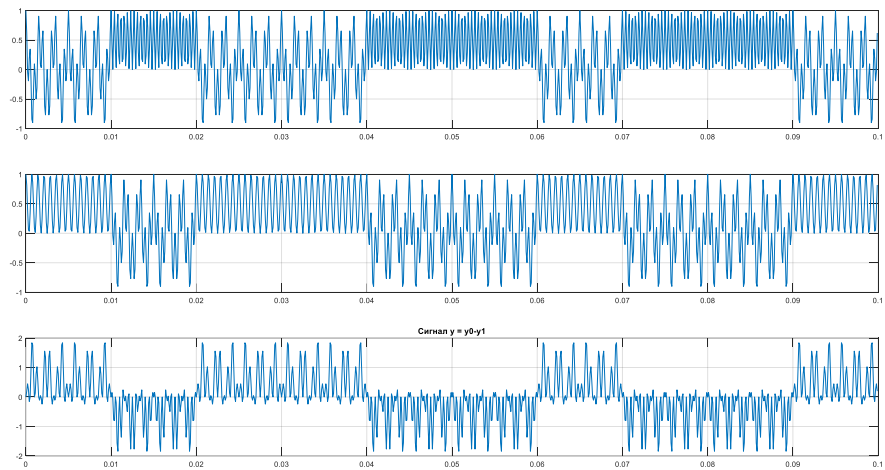


Рисунок 16: Диаграммы сигналов y_0, y_1, y

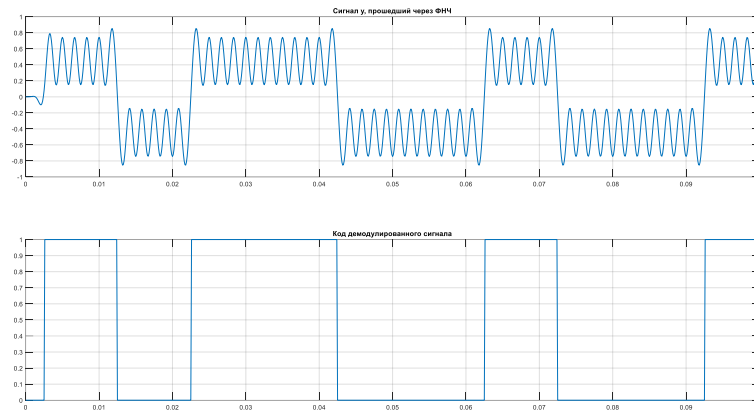


Рисунок 17: FSK-демодуляция

Вывод:

В данном задании рассмотрим частотную манипуляцию (FSK). Модулирующий сигнал и модулированный сигнал FSK показаны на рисунке 15. Из этой рисунка видно, что что в зависимости от значения модулирующего сигнала, меняется частота несущей. С помощью ФНЧ удаляем высокочастотные составляющие частоты при FSK демодуляции. График кода демодулированного сигнала в рисунке 17 имеет ту же форму переданного кода ,но с задержкой из-за КИХ-фильтра.