

1. Цель работы

- Для выбранного варианта задания разработать и отладить программу моделирования с использованием алгоритма из учебно-методического пособия
- Выполнить моделирование для нескольких значений отношения сигнал/шум
- Для выбранного варианта вычислить значения вероятности ошибки или её верхней границы по соответствующим теоретическим формулам
- Построить графики для данных, полученных в п.2 и п.3

Вариант III.10(Квадратурная амплитудная модуляция):

= 1800 Гц – несущая частота;

= 2400 Бод – модуляционная скорость;

= 14400 бит/с – информационная скорость.

2. Алгоритм моделирования работы оптимального приёмника

Процесс моделирования состоит в многократном выполнении следующих шагов:

- случайный равновероятный выбор номера сигнала $i = -1, 0, 1, \dots, q-1$, подлежащего передаче;
- получение сигнала $r(t)$ на выходе канала согласно равенству $r(t) = n(t) + s_i(t)$, где $n(t)$ – АБГШ со спектральной плотностью мощности $N(f)$;
вычисление вектора функций $r(t)$ с компонентами $r(t) = [r_1(t), r_2(t), \dots, r_M(t)]$;
- фиксации результата: если $r(t) \in \Omega_i$, то решение на приемной стороне сформировано неправильно, и нужно увеличить счетчик числа ошибок.

Формула вероятности ошибки для КАМ:

3. Выполнить моделирование для нескольких отношений сигнал/шум

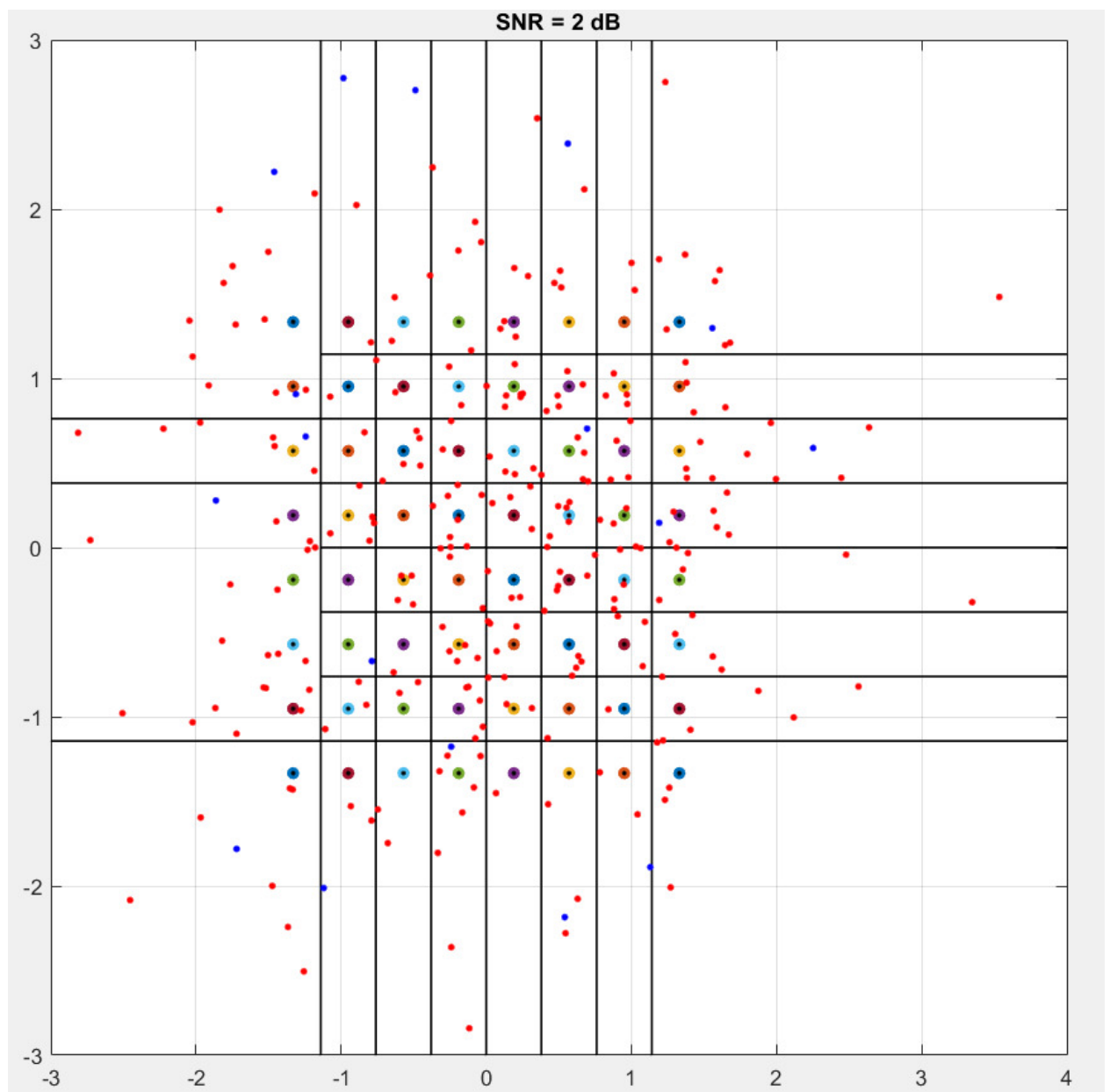


Рис. 1 – Облако рассеивания для соотношения сигнал/шум 2 дБ

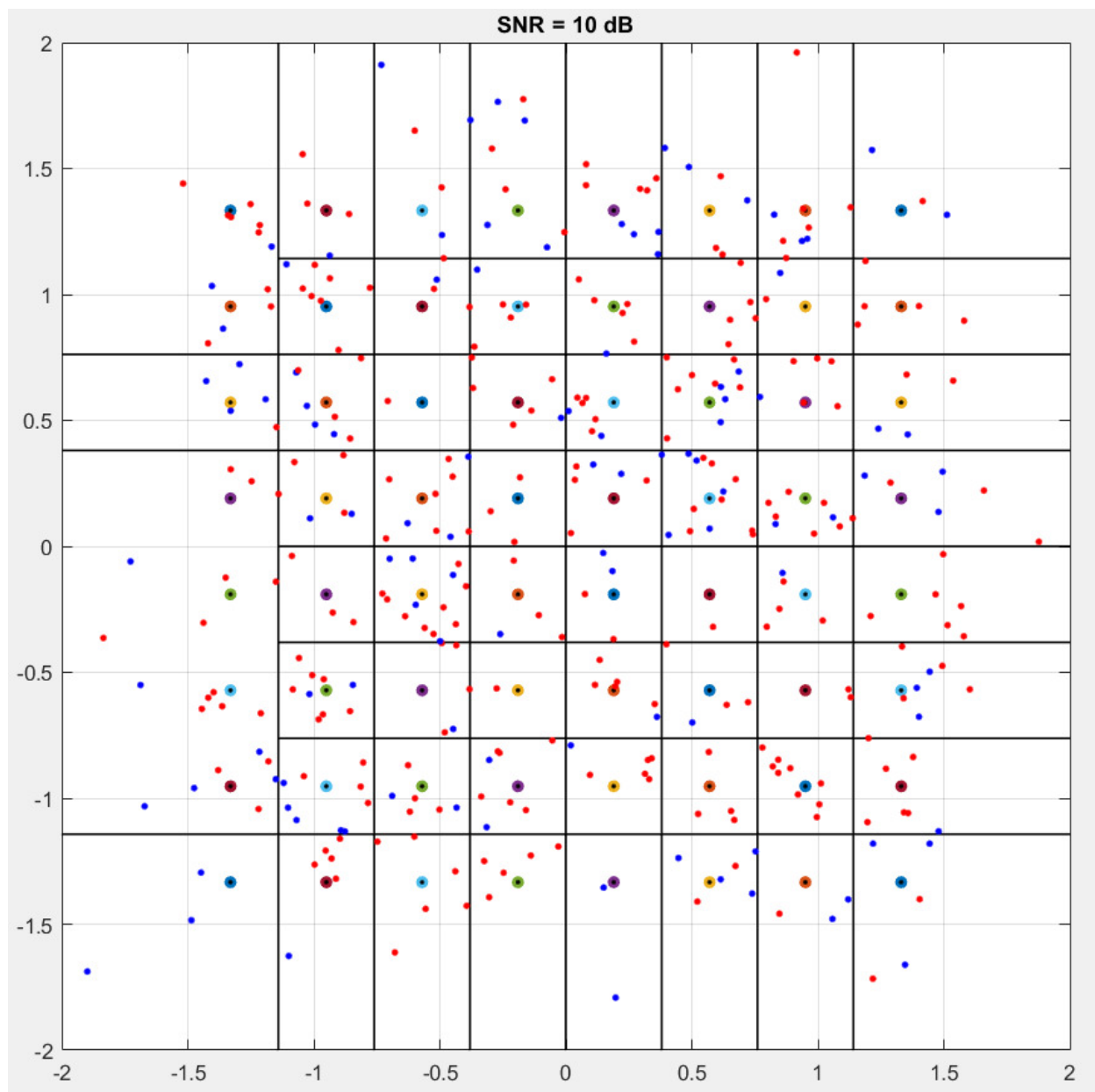


Рис. 2 – Облако рассеивания для соотношения сигнал/шум 10 дБ

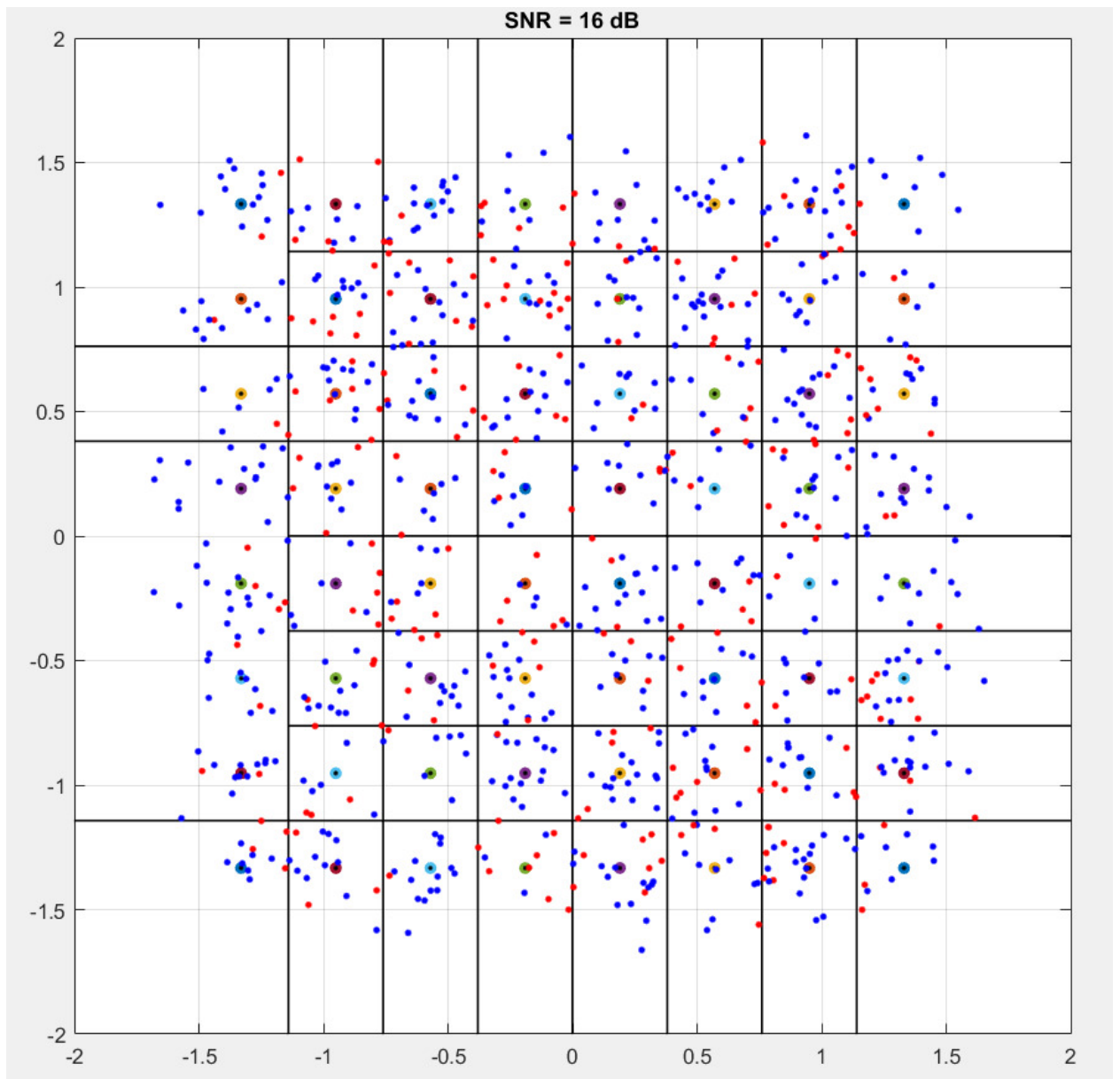


Рис. 3 – Облако рассеивания для соотношения сигнал/шум 16 дБ

4. Вычислить значение вероятности ошибки

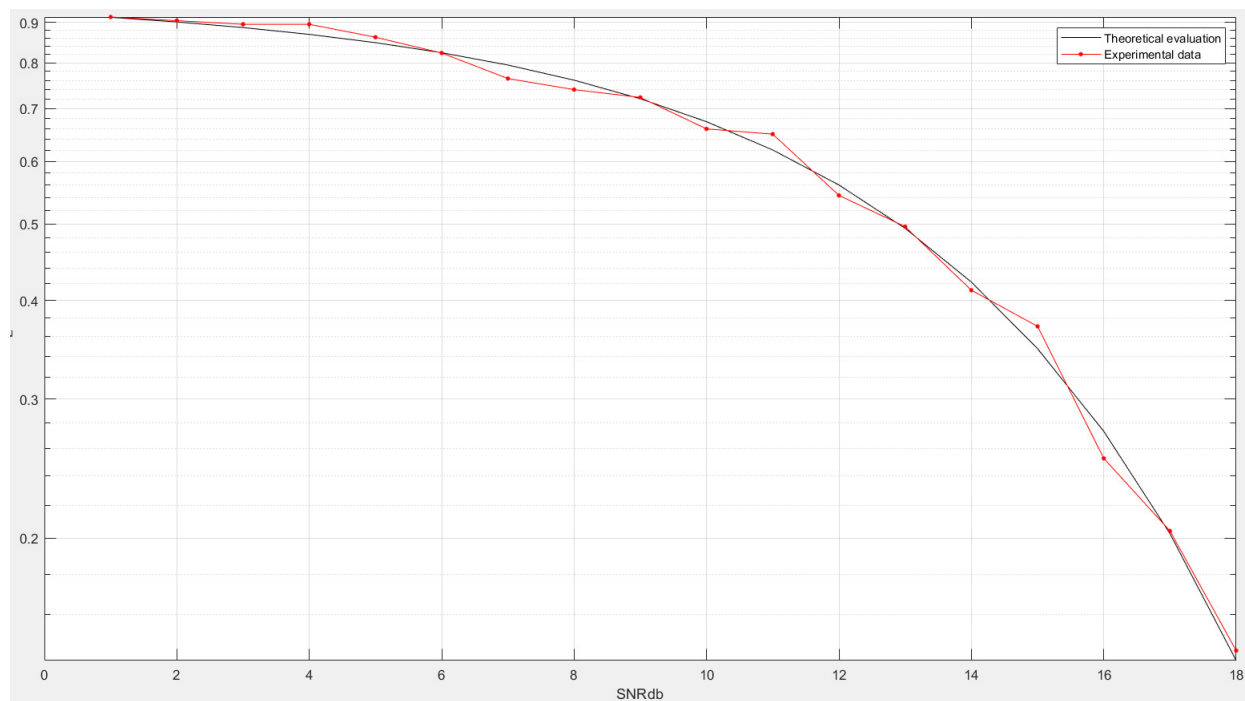


Рис. 4 – Экспериментальная и теоретическая зависимости вероятности ошибки от отношения сигнал/шум

5. Вывод:

В ходе лабораторной работы была промоделирована работа оптимального приёмника дискретных сигналов в канале с белым аддитивным белым гауссовским шумом.

- Был получен сигнал $r(t)$ на выходе приёмника и вычислен вектор r
- Было произведено формирование решения и фиксация результата
- Была произведена оценка вероятности ошибки теоретическим и экспериментальным путём

6. Код программы:

```
clc;
clear;
close all;
f0 = 1800;
Vmod = 2400;
Vinf = 14400;
T = 1 / Vmod;
m = Vinf * T;
q = 2^m;
W = 2 / T;
dt = 0.000001;
t = 0:dt:1/f0;
i1 = zeros(q,1);
i2 = zeros(q,1);
A = 1;

s1s2 = zeros(q,2);
for c = 1:q
    i1(c) = floor((c - 1) / sqrt(q));
    i2(c) = mod(c - 1, sqrt(q));
    s1s2(c,1) = A*(1-((2*i1(c))/(sqrt(q)-1)));
    s1s2(c, 2) = A*(1-((2*i2(c))/(sqrt(q)-1)));
end

s = zeros(q,length(t));
for c = 1:q
    s(c,:) = (s1s2(c,1)*sqrt(W).*cos(2*pi*f0*t)) +
    (s1s2(c,2)*sqrt(W).*sin(2*pi*f0*t));
end

phi_1 = sqrt(W)*cos(2*pi*f0*t);
phi_2 = sqrt(W)*sin(2*pi*f0*t);

sij = zeros(q,2);
si1 = zeros(q);
si2 = zeros(q);
for c = 1:q
    sij(c,1) = trapz(t,s(c,: ).*phi_1);
    sij(c,2) = trapz(t,s(c,: ).*phi_2);
end

for c = 1:q
    plot(sij(c,1), sij(c,2), '.', 'MarkerSize', 20);
    hold on;
end
grid on;
```



```

set(voronoi(sij(:,1),sij(:,2)), 'LineWidth',1, 'Color',
'black');

Noise = 2;
SNRdB = 1 : 1 : 18;
Pe = zeros(1,length(SNRdB));
Pe_t = zeros(1,length(SNRdB));
Nerrmax = 256;
E = sum(s .* s) / (q);
for n = 1 : length(SNRdB)
    SNR = 10 ^ (SNRdB(n) / 10);
    sigma = sqrt(sum(E / (2 * SNR)));
    Nerr = 0;
    Ntest = 0;
    while (Nerr < Nerrmax)
        i = floor(rand * q) + 1;
        r = s(i,:) + sigma .* randn(1,length(s(i,:)));
        r1 = trapz(t,r .* phi_1);
        r2 = trapz(t,r .* phi_2);
        if (n == Noise)
            plot(r1, r2, 'b.','MarkerSize', 10);
            title(['SNR = ',num2str(SNRdB(n)), ' dB']);
            hold on;
            axis square;
        end
        i_ = -1;
        min_dist = inf;
        for c=1:q
            x = sij(c,1);
            y = sij(c,2);
            dist = sqrt((r1-x)^2+(r2-y)^2);

            if dist < min_dist
                min_dist = dist;
                i_ = c;
            end
        end
        if i_ ~= i
            Nerr = Nerr + 1;
            if (n == Noise)
                plot(r1, r2, 'r.','MarkerSize', 10);
                title(['SNR = ',num2str(SNRdB(n)), ' dB']);
                hold on;
                axis square;
            end
        else
            if (n == Noise)
                plot(r1, r2, 'b.','MarkerSize', 10);

```

```

        title(['SNR = ', num2str(SNRdB(n)), ' dB']);
        hold on;
        axis square;
    end
end
Ntest = Ntest + 1;
end
Pe(n) = Nerr/Ntest;
SNR = 10^(n/10);
N0 = E/SNR;

Pe_t(n) = 4*(sqrt(q)-1)/q * qfunc(sqrt(3*E/N0/(q-1))) * (sqrt(q) - (sqrt(q)-1)*qfunc(sqrt(3*E/N0/(q-1))));
end
figure;
semilogy(SNRdB, Pe_t, 'black-', SNRdB, Pe, 'red.-',
'MarkerSize', 10);
legend('Theoretical evaluation', 'Experimental data');

grid on
xlabel('SNRdb');
ylabel('P_E');

```