

Цель работы

Получение геометрического представления сигналов.

Исходные данные

Вариант 3.3.

Выполнение работы

1. Для выбранного варианта задания выбрать множество базисных функций. В случае квадратурной амплитудной модуляции, базисными будут являться следующие функции:

(1)

В таком случае размерность базиса $D = 2$.

2. Проверить выполнение условия для выбранного множества базисных функций.

(2)

2.1. Для случая :

(3)

2.2. Для случая :

(4)

2.3. Для случая :

(5)

Условие подтвердилось для выбранного базиса.

3. Построить множество сигнальных точек, то есть, вычислить координаты каждой сигнальной точки и изобразить сигнальное созвездие.

3.1. Для заданного варианта КАМ, $q = 4$. Для коэффициентов разложения s_{ij} используются следующие формулы:

(6)

Где i_k – числа в представлении числа i .

- 3.2. Сигнал же представляется линейной комбинацией D базисных функций

(7)

- 3.3. Изобразим сигнальное созвездие, подставив в выражения (5) следующие значения i_k .

Таблица 1 - Значение коэффициентов i_k

i	i
1	2
0	0
0	1
1	0
1	1

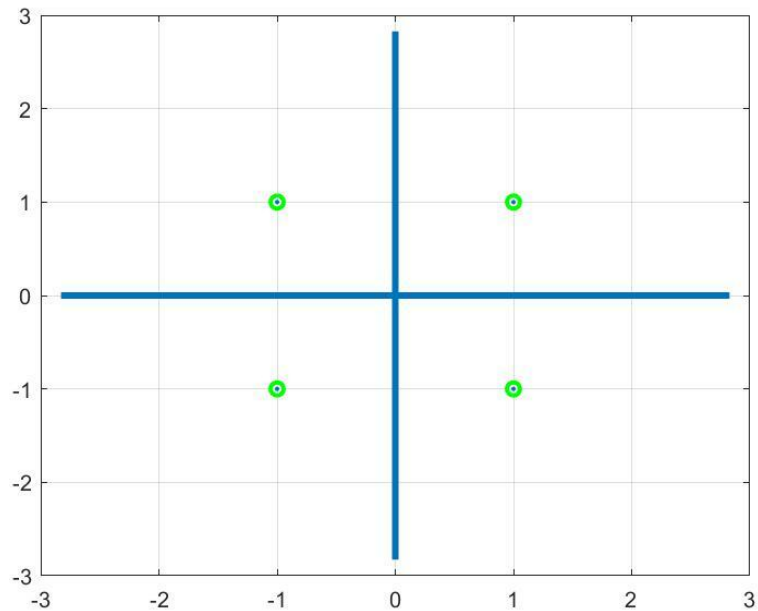


Рисунок 1 - Сигнальное созвездие КАМ 4

Как и предполагалось, элементы сигнального созвездия расположены с равномерным шагом в интервале $[-A, A]$.

4. Построить разбиение сигнального пространства на разрешающие области.

4.1. Для визуализации разрешающих областей воспользуемся диаграммой Вороного.

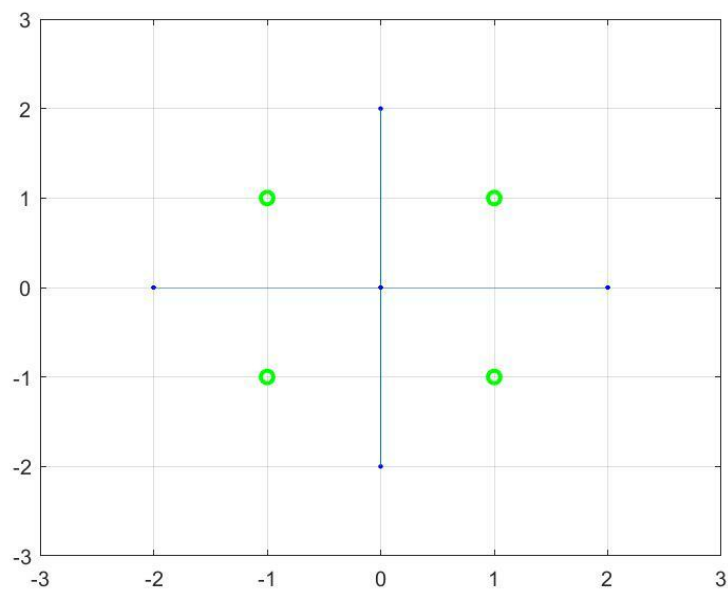


Рисунок 2 – Разрешающие области для КАМ4

Как можно увидеть на графике, для КАМ4 мы имеем 4 бесконечные области.

Вывод

В данной работе по заданным базисным функциям было построено сигнальное созвездие, построено разбиение сигнального пространства на разрешающие области. Для выбранного варианта КАМ4, мы имеем 4 точки в сигнальном созвездии и 4 бесконечных разрешающих областей.

Листинг

```
clear;
close all;
T = 5;
l = 6;
f0 = l/T;
t = 0: 1/1000*T : T;

f1 = sqrt(2/T)*cos(2*pi*f0*t);
f2 = sqrt(2/T)*sin(2*pi*f0*t);
disp('Условие ортогональности функций');
disp(sum(f1 .* f2));

%Нормировка
n1 = sqrt(sum(f1.*f1));
f1 = f1./n1;

n2 = sqrt(sum(f2.*f2));
f2 = f2./n2;

mult1 = (sum(f1.*f1));
mult2 = (sum(f1.*f2));
mult3 = (sum(f2.*f2));

disp(mult1);
disp(mult2);
disp(mult3);

i1 = [0, 0, 1, 1];
s1 = zeros(0,4);
i2 = [0, 1, 0, 1];
s2 = zeros(0,4);

figure(1);
for i = 1 : 4
    s1(i) = (1-(2*i1(i)/(sqrt(4)-1)));
    s2(i) = (1-(2*i2(i)/(sqrt(4)-1)));
    plot(s1(i), s2(i),'o', 'Color', 'g','LineWidth',2);
    hold on;
end
axis([-3 3,-3,3]);
grid on;

set(voronoi(s1,s2),'LineWidth',3, 'Color', 'r');

figure(2);
for i=1 : 4
```

```

        plot(s1(i), s2(i), 'o', 'Color', 'g', 'LineWidth', 2);
        hold on;
    end
    axis([-3 3, -3, 3]);
    grid on;
    E = 2;
    for i=1 : 4
        for j = 1 : 4
            if ((s1(i)-s1(j))^2 + (s2(i) - s2(j))^2 < E^2+1)
                syms x y;
                [x, y] = solve((s1(i) - x).^2 + (s2(i) - y).^2 - E, (s1(j) -
x).^2 + (s2(j) - y).^2 - E);
                plot (x(1), y(1), '+', 'Color', 'b', 'LineWidth', 2);
                plot (x(2), y(2), '+', 'Color', 'b', 'LineWidth', 2);
                line ([x(1) x(2)], [y(1) y(2)]);
                hold on;
            end
        end
    end
end

```