Лабораторная работа № 1:
“Дискретизация и квантование сигналов”

Выполнил:

студент группы МП-30,

Алимагадов Курбан Алимагадович

МИЭТ 2019 г.

Часть 1. Ортогональные системы в гильбертовом пространстве. Ряды Фурье. Частотное представление сигнала

1. Найти коэффициенты разложения функции f(x) в ряд Фурье по системе Хаара, используя функцию fseries. (используйте функцию heaviside для задания f(x)). Проверить выполнение равенства Парсеваля.

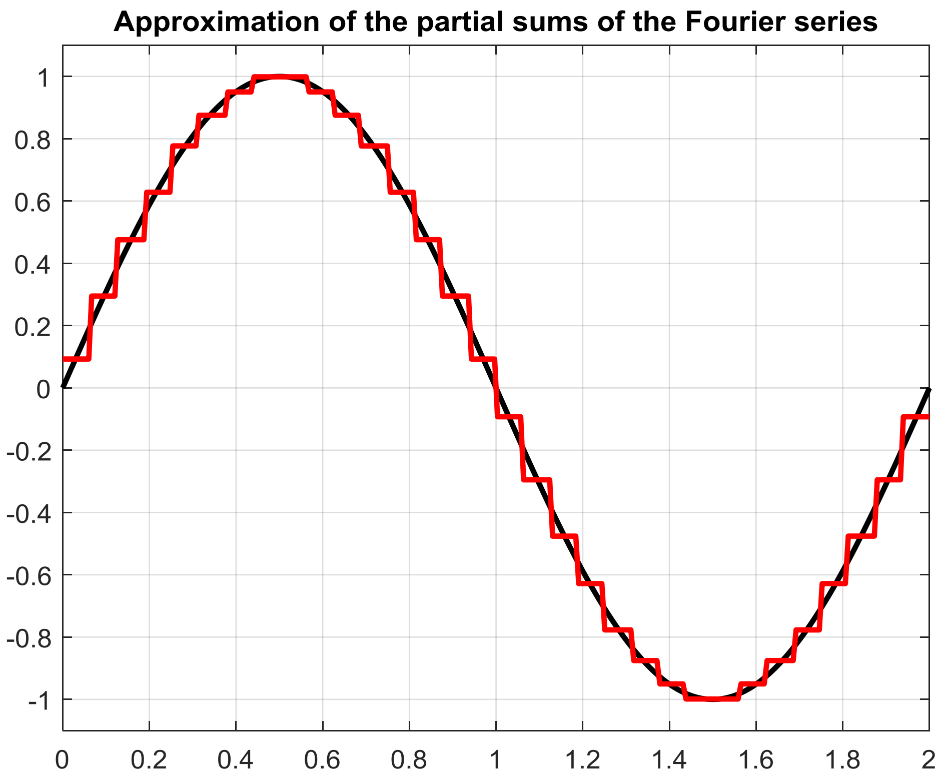
```
f = @(x)(heaviside(x-0.25) + heaviside(x-0.5) - 2*heaviside(x-0.75));
fseries(f, 0, 1, [0:3], 'haar')

ans =

   0.750001511464811  -0.249999071952813  -0.353553062479044
0.707105140615400
```

Проверка:

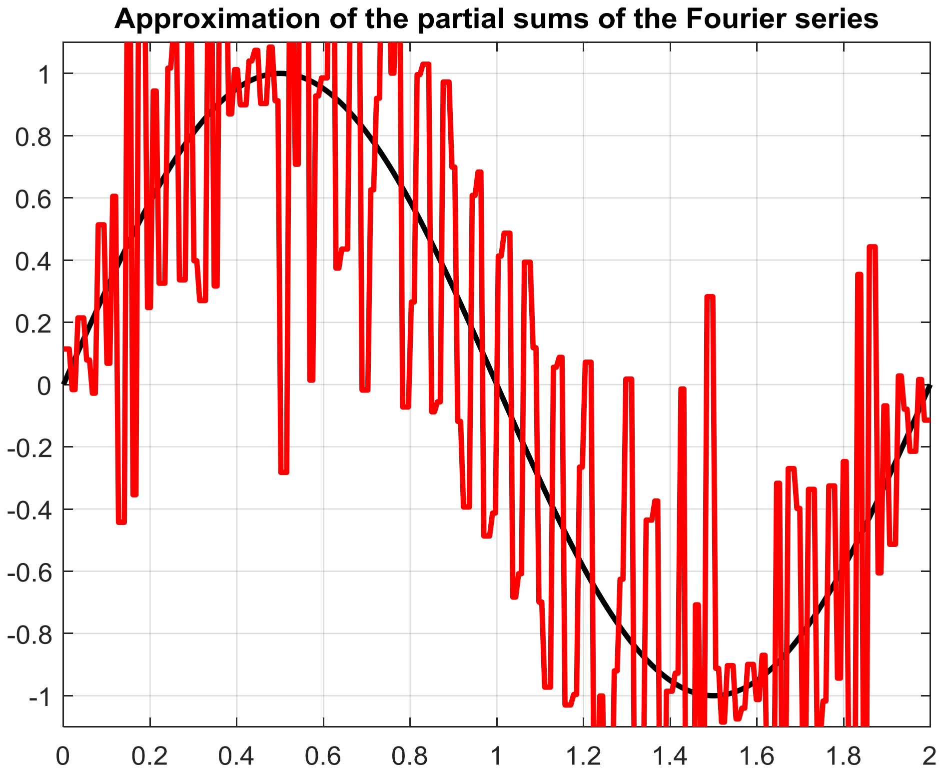
$$||f(x)||^2 = \sum_{k=0}^{3} \lambda_k^2 = \frac{9}{16} + \frac{1}{16} + \frac{1}{8} + \frac{1}{2} = \frac{5}{4}$$

2. Представить синусоиду в виде последовательности частичных сумм ряда Фурье по системам Уолша и Хаара, используя программу sum_task.m

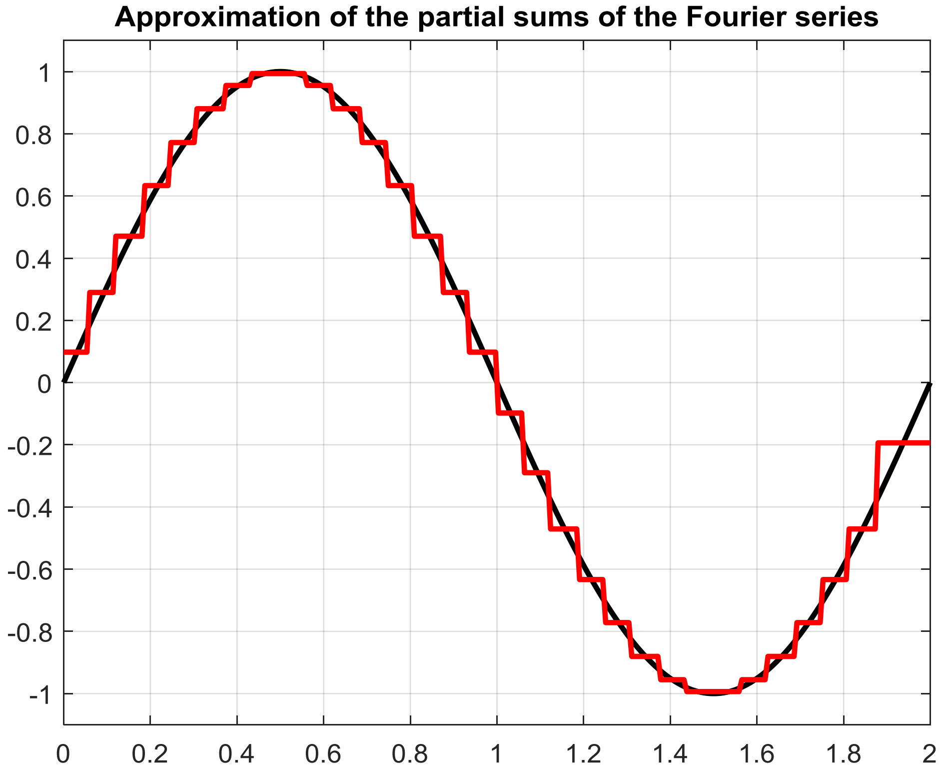
Частичная сумма по системе Уолша (30 коэффициентов ряда Фурье):



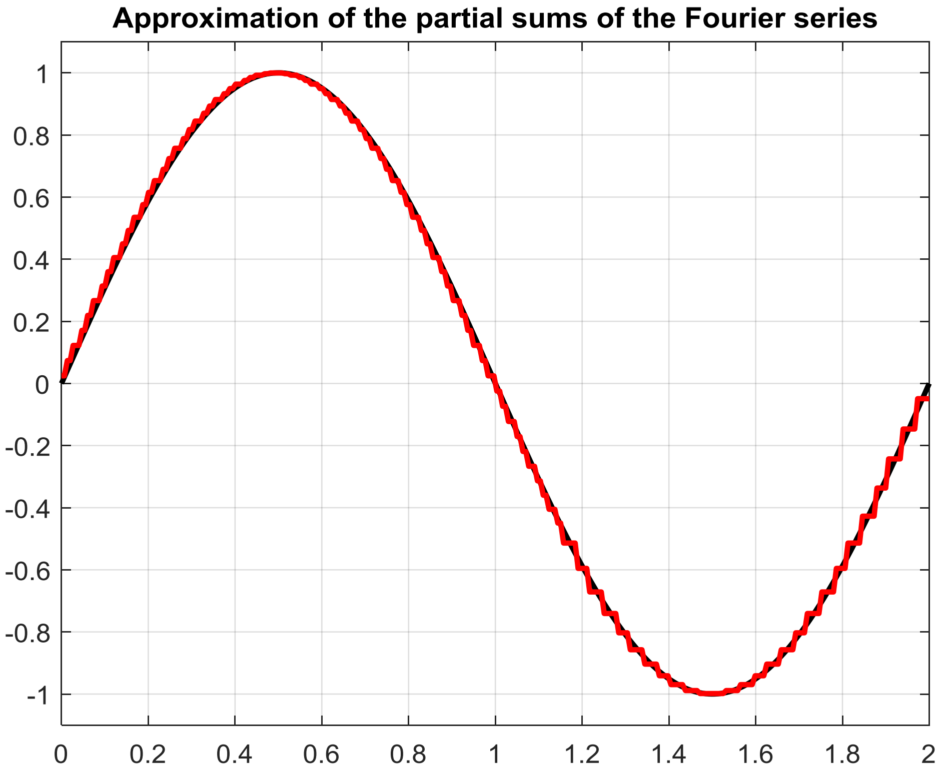
Частичная сумма по системе Уолша (100 коэффициентов ряда Фурье):



Частичная сумма по системе Хаара (30 коэффициентов ряда Фурье):

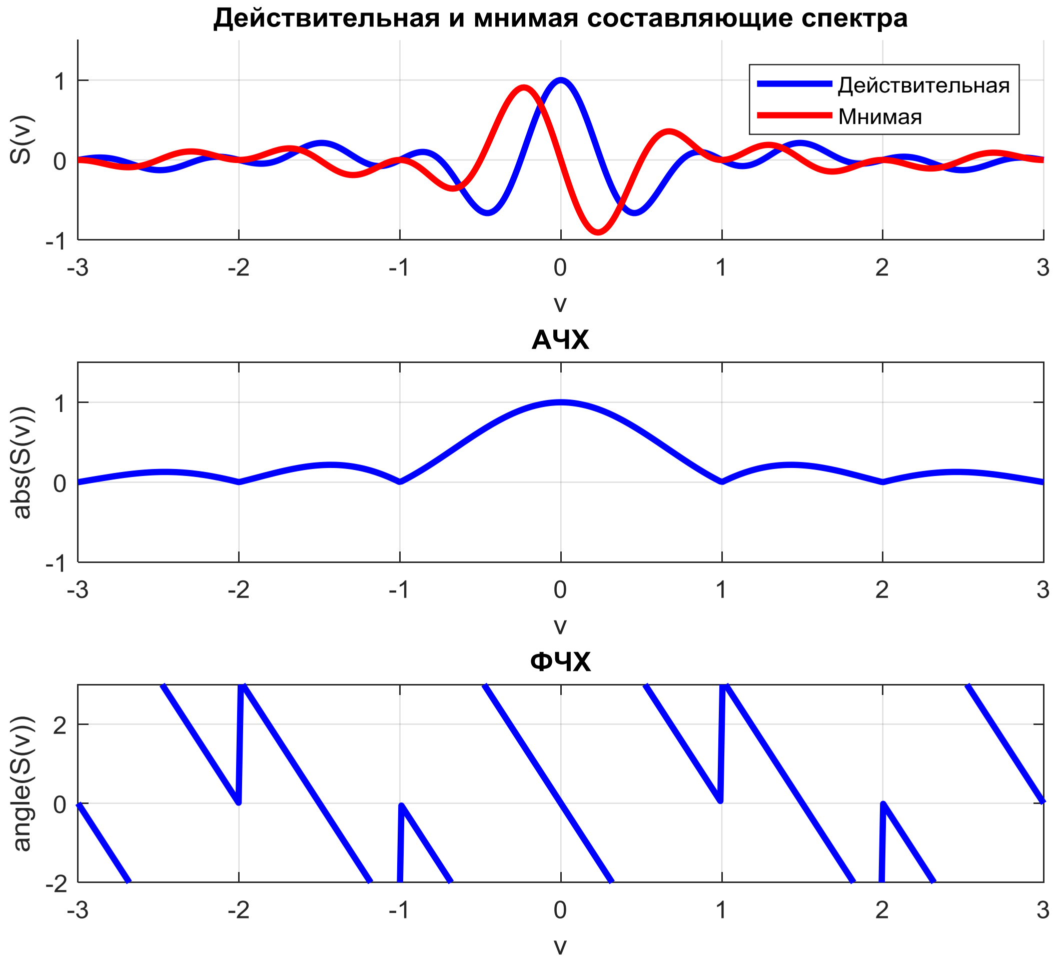


Частичная сумма по системе Хаара (100 коэффициентов ряда Фурье):



При больших количествах членов суммы ряда частичные суммы по системе Уолша хуже аппроксимируют синусоиду, чем частичные суммы по системе Хаара.

3. С помощью программы sum_fourier.m выполнить интегральное преобразование Фурье (в символьном виде) тестового сигнала, построить амплитудный и фазовый спектры.

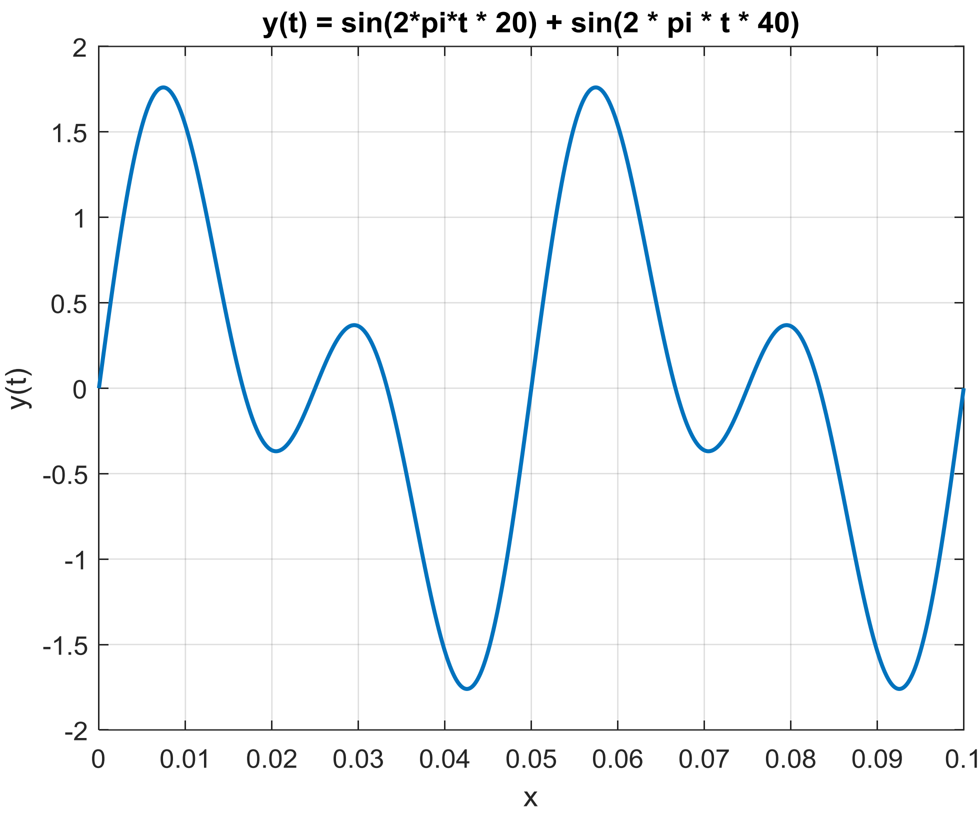


Часть 2. Исследование эффектов дискретизации.

1. Синтезировать сигнал x (t) , представляющий из себя сумму нескольких синусоид с разными частотами.

Код на языке Matlab в файле tasks.m в разделе part 2 task 1.

```
y = @(t) sin(2*pi*t * 20) + sin(2 * pi * t * 40);
```

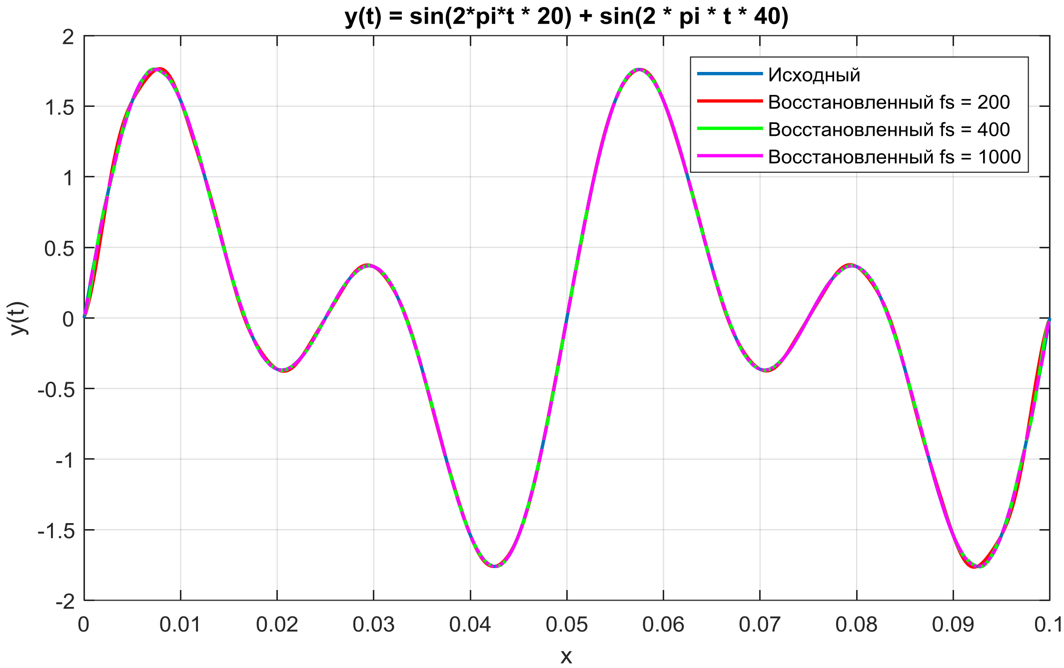


2. Определить допустимые значения частоты f_s дискретизации для сигнала x(t)

Допустимые значения частоты дискретизации: $f_s > 2*F_{max} = 2*40 = 80$.

3. Построить по отсчетам график исходного сигнала и его спектра при нескольких различных частотах дискретизации. Сделайте вывод?

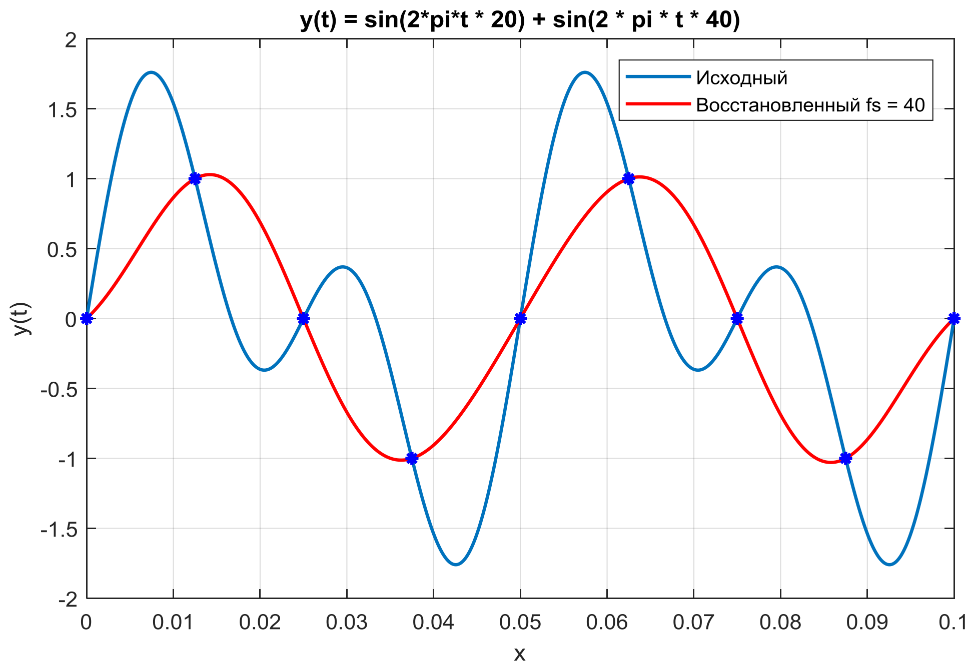
Код на языке Matlab в файле tasks.m в разделе part 2 task 3.



Так как частота дискретизации соответствует условию $f_s > 2*F_{max}$, то восстановленные сигналы при частотах $f_{s1} = 1000$, $f_{s2} = 400$, $f_{s3} = 200$ совпадают с аналоговым.

4. Проиллюстрировать на примере сигнала x (t) эффект наложения частот. Для этого необходимо привести сигнал x`(t), который при некоторой частоте дискретизации будет совпадать с сигналом x(t). Такого эффекта можно добиться, если дискретизацию сигнала x(t) с неверной частотой дискретизации и восстановить его.

Код на языке Matlab в файле tasks.m в разделе part 2 task 4.



5. Воспроизвести сигнал x(t) с различными частотами дискретизации. Какие выводы можно сделать?

```
y = @(t) sin(2*pi*t * 20) + sin(2 * pi * t * 40);
sound(y((1:10000)./pi))
pause(4)
sound(y((1:10:100000)./pi))
```

При изменении частоты дискретизации звуковой сигнал искажается.

6. Загрузить тестовое изображение. Уменьшить частоту дискретизации в 2, 3, 4 раза с помощью прореживания матрицы исходного изображения. Сравнить полученные результаты с результатом использования функции imresize.

Код на языке Matlab в файле tasks.m в разделе part 2 task 6.

Исходное изображение

Прореживание с уменьш. част. дискр. в 2 раза

Уменьшение с помощью imresize в 2 раза

Прореживание с уменьш. част. дискр. в 3 раза

Уменьшение с помощью imresize в 3 раза

Прореживание с уменьш. част. дискр. в 4 раза

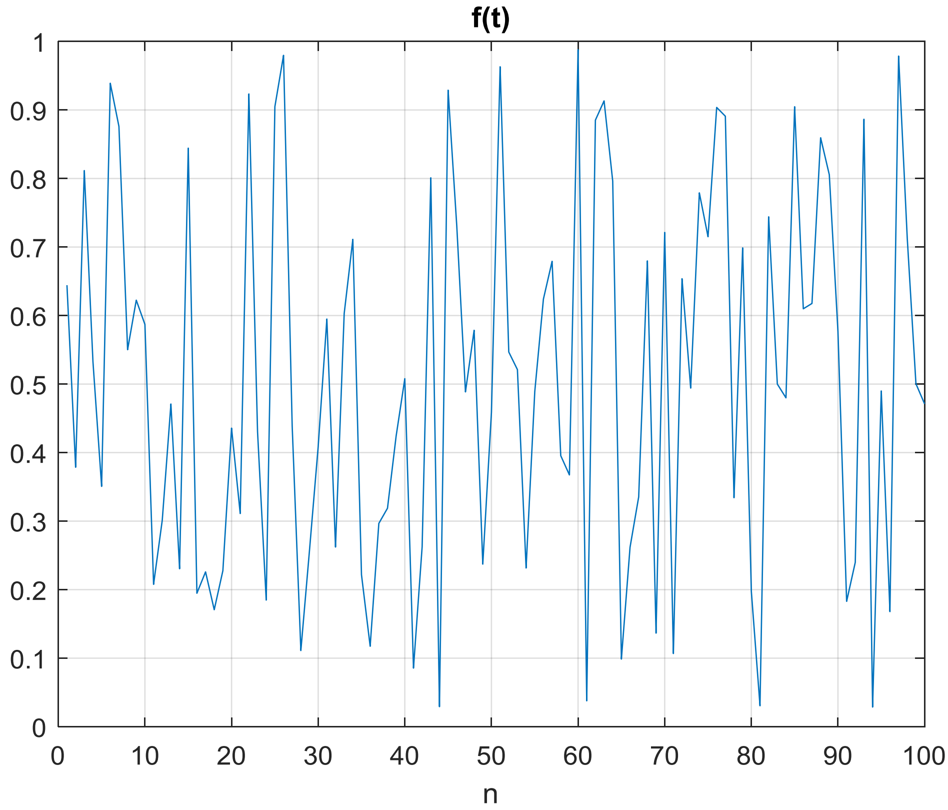
Уменьшение с помощью imresize в 4 раза

Изображения, полученные в результате уменьшения размера картинки с помощью команды imresize, выглядят более сглаженными, чем в случае с прореживанием, так как функция imresize, проводит интерполяцию по соседним пикселям, усредняя значения частот.

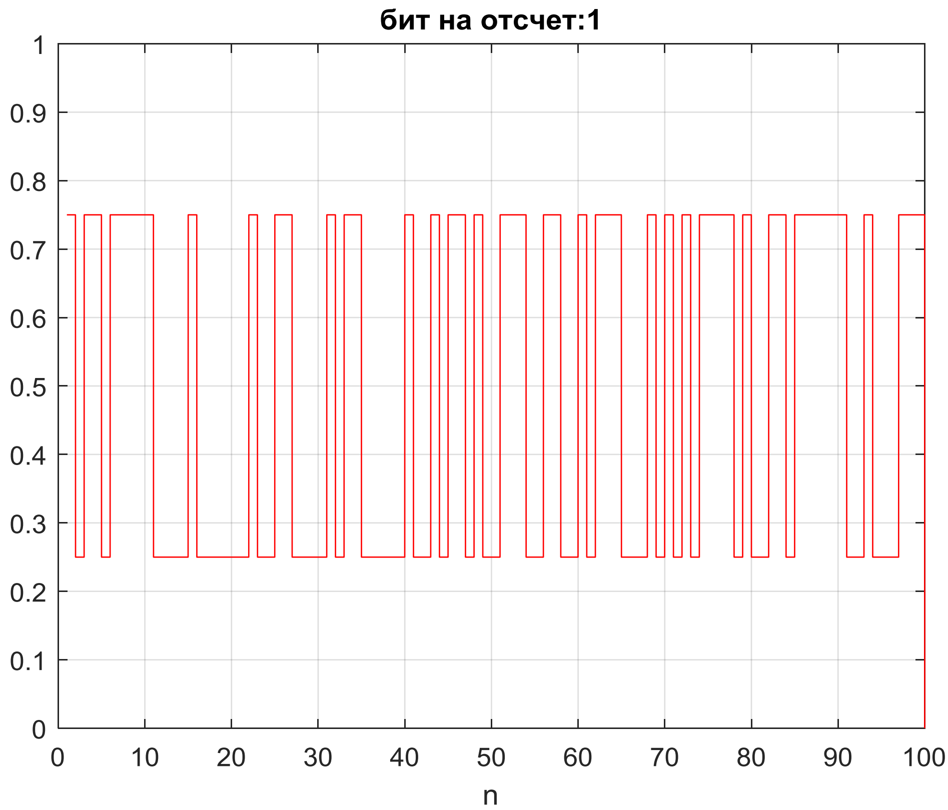
Часть 3.

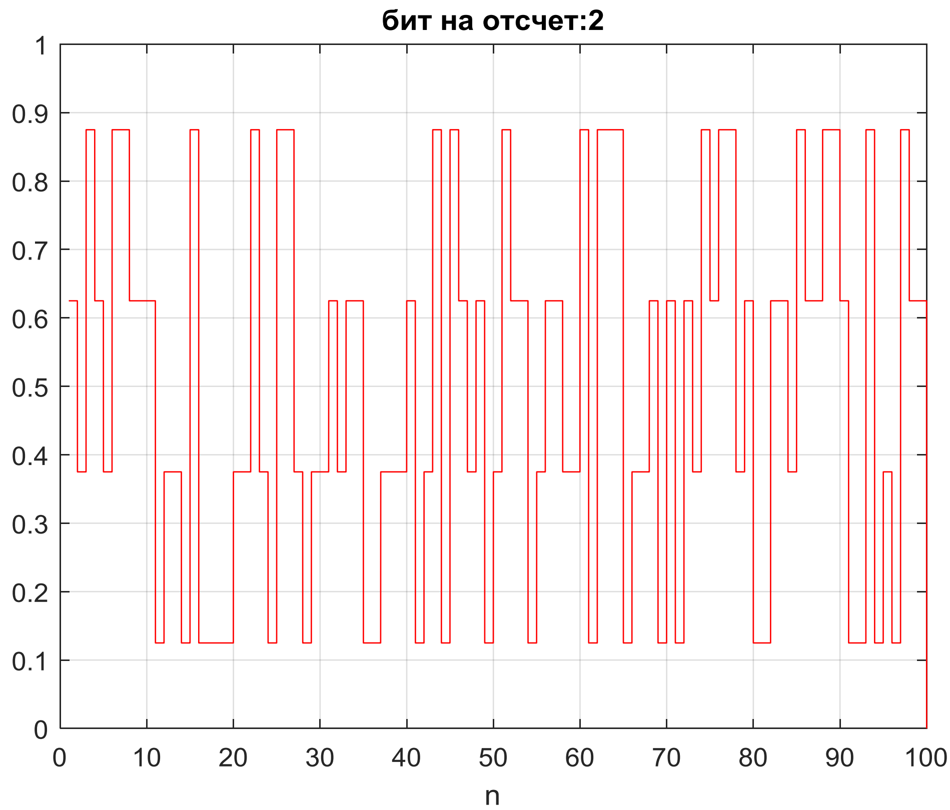
1. Синтезировать случайный дискретный сигнал x с равномерным распределением. Построить по отсчетам его график.

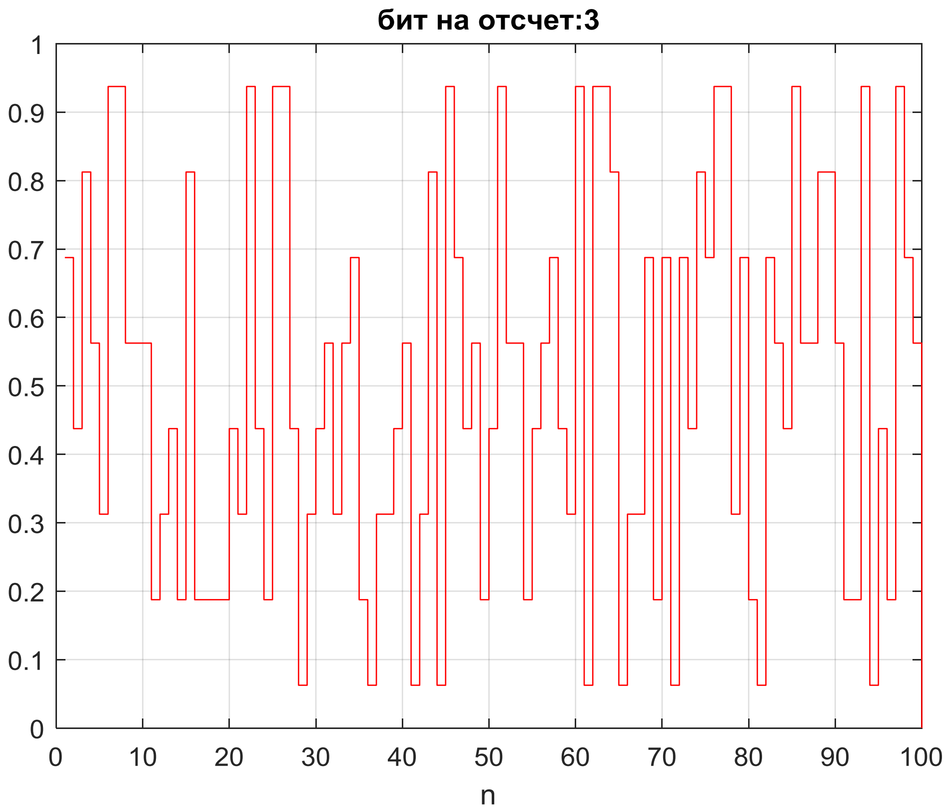
Код на языке Matlab в файле tasks.m в разделе part 3 task 1, 2.

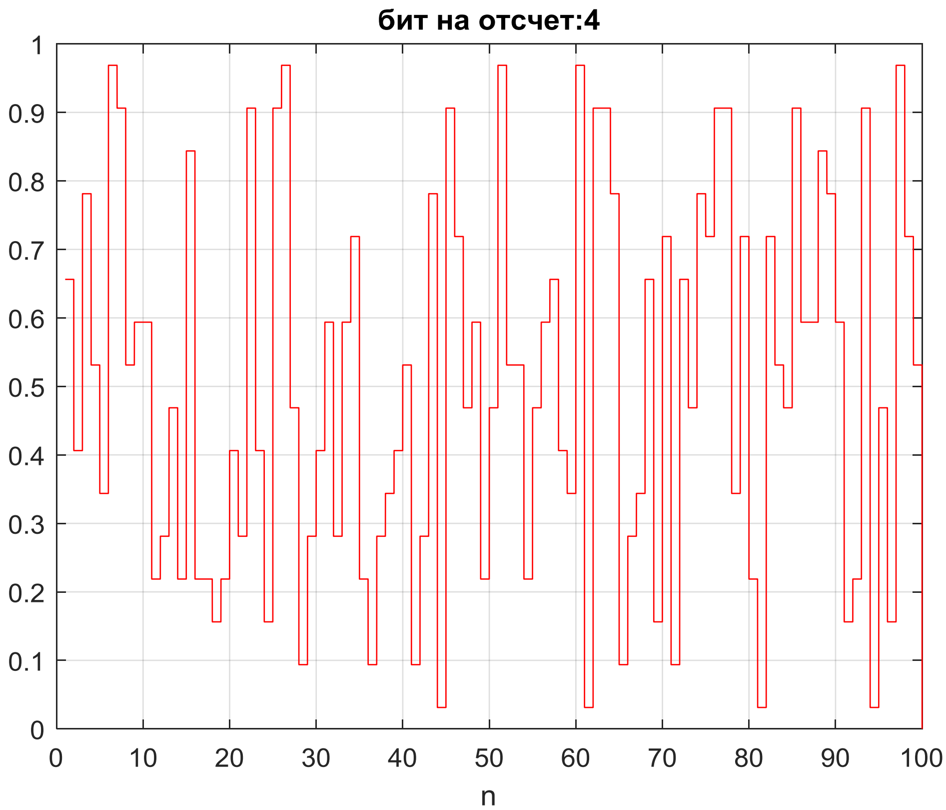


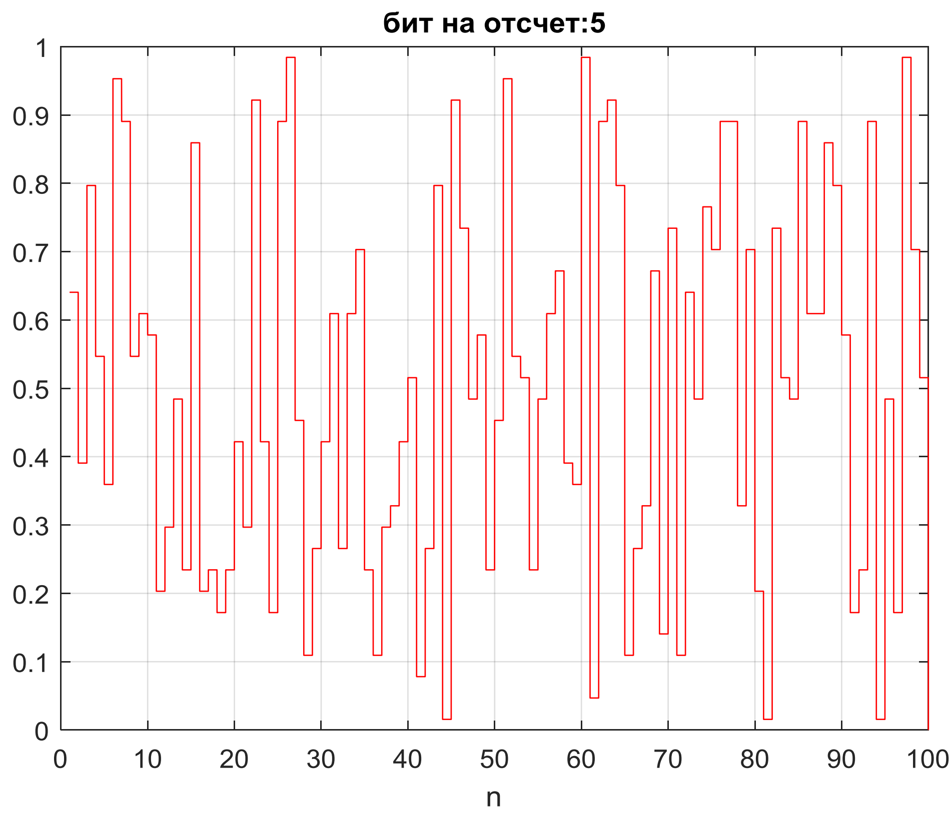
2. Провести равномерное квантование отсчетов сигнала x , используя от 1 до 8 бит на отсчет. Построить ступенчатые графики сигнала после квантования.

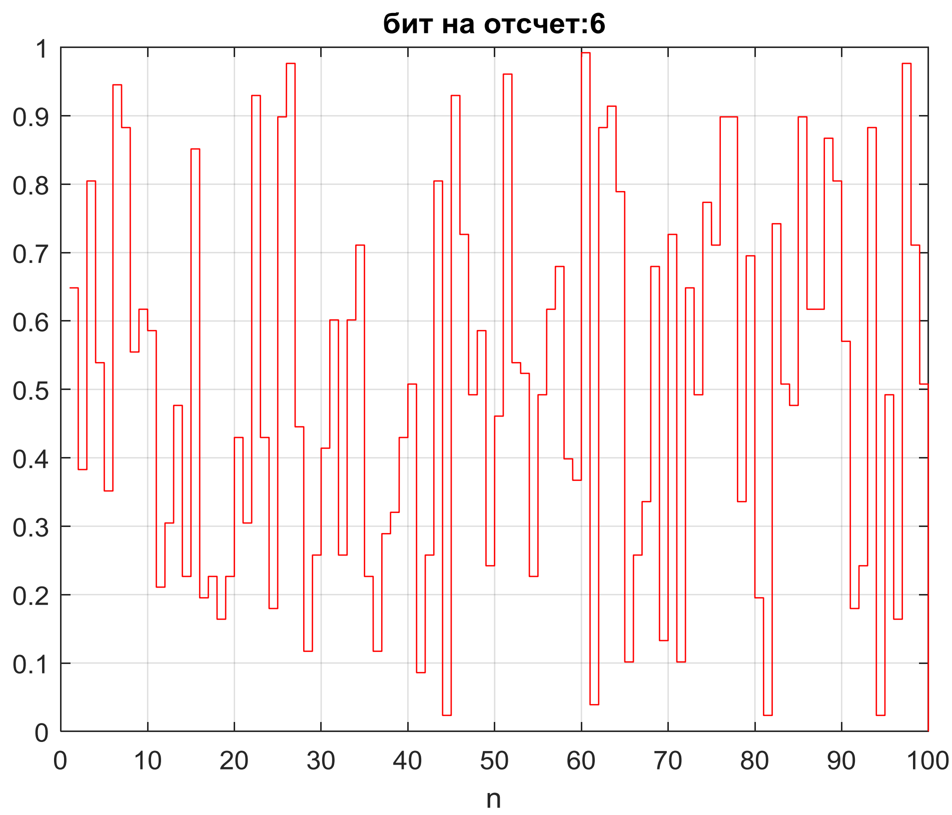


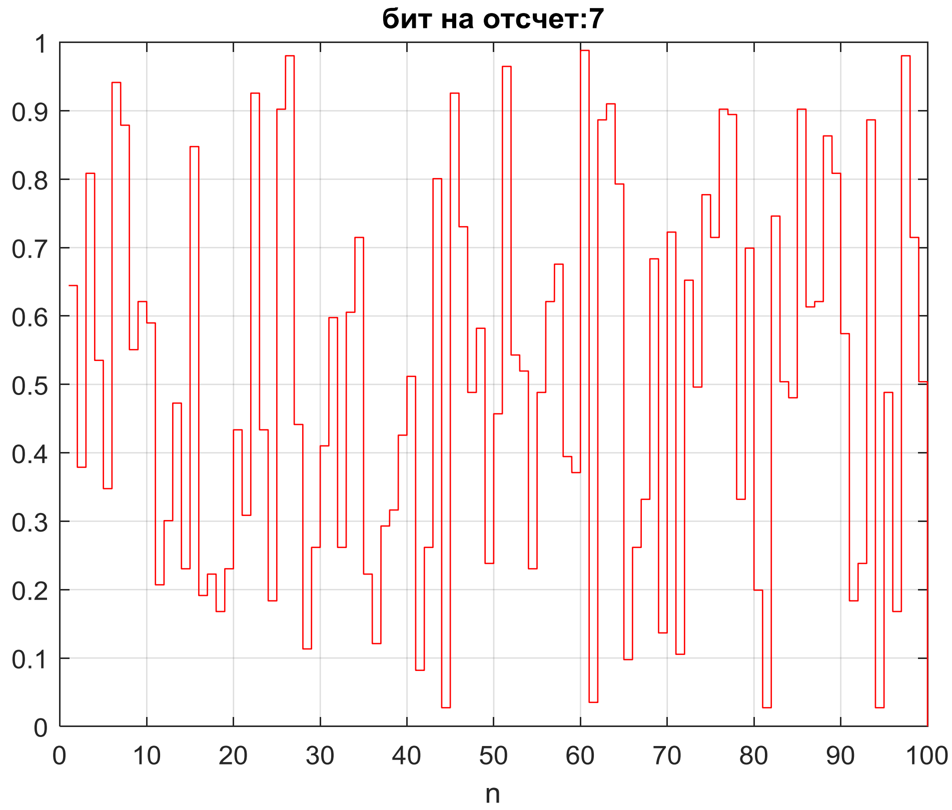


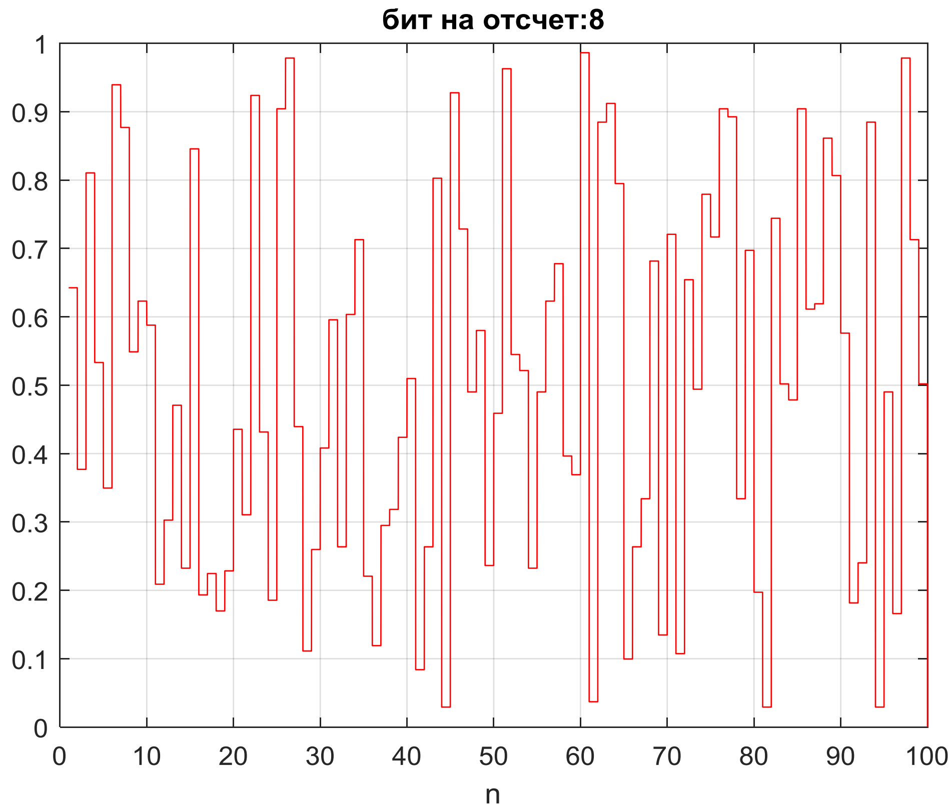

бит на отсчет:3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
10
20
30
40
50
60
70
80
90
100
n


бит на отсчет:4
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
10
20
30
40
50
60
70
80
90
100
n

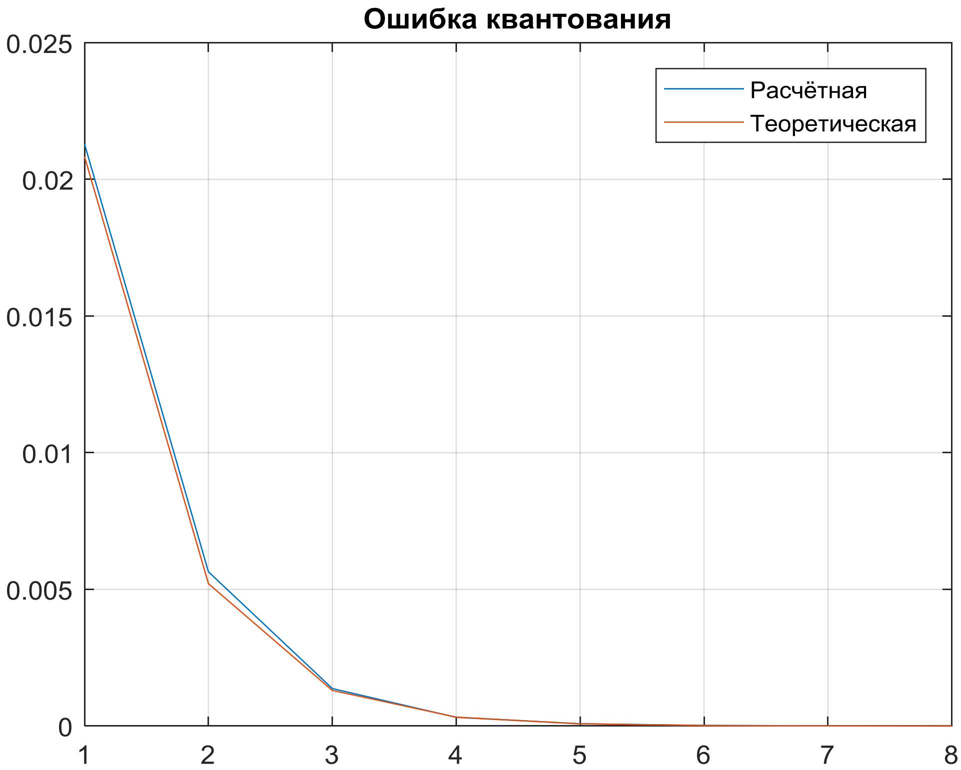

бит на отсчет:5
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
10
20
30
40
50
60
70
80
90
100
n


бит на отсчет:6
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
10
20
30
40
50
60
70
80
90
100
n


бит на отсчет:7
n

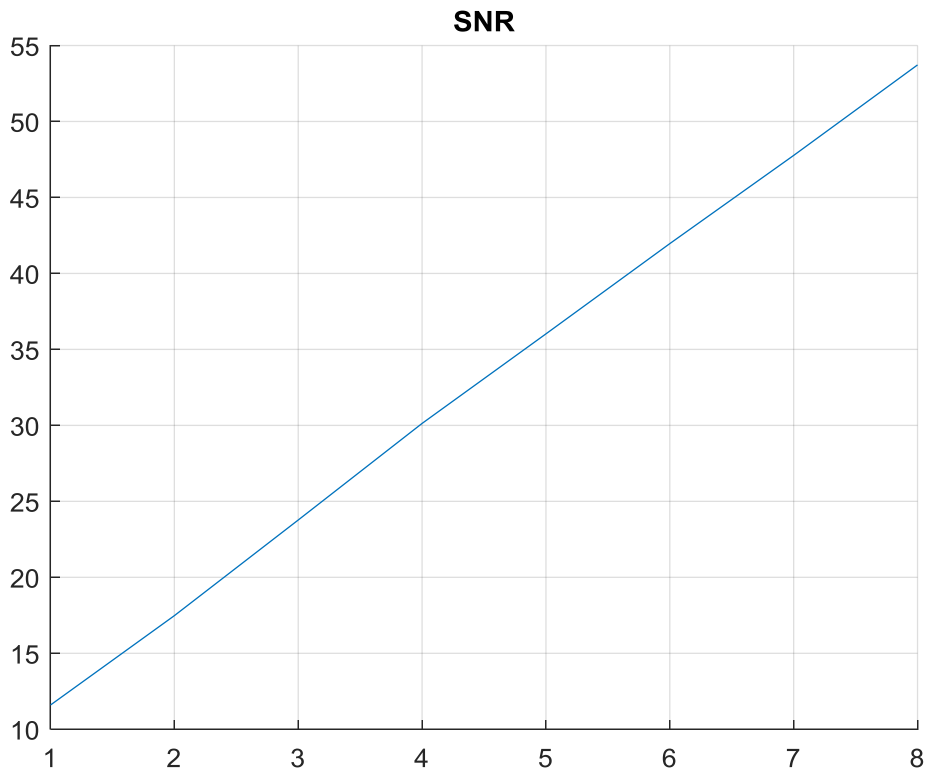

бит на отсчет:8
n

3. Экспериментально определить ошибку квантования (2). Сравнить полученные результаты с теоретической оценкой (5).



Расчётная ошибка практически совпадает с теоретической.

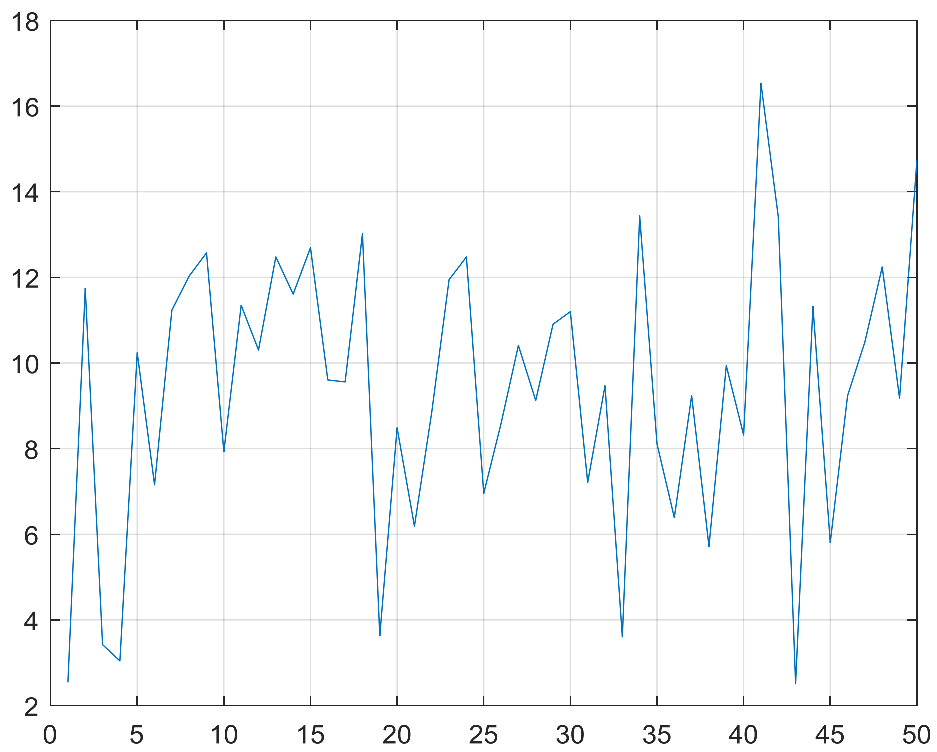
4. Вычислить SNR (3). Исследовать зависимость SNR от числа бит, выделяемого для хранения одного отсчета сигнала.



При увеличении числа бит, выделяемых для хранения одного отсчета сигнала, отношение сигнал-шум растёт по линейному закону, что свидетельствует о получении более точной аппроксимации.

5. Синтезировать случайный дискретный сигнал x с нормальным распределением. Построить по отсчетам его график.

```
f = normrnd(10,3,[1,50]);
figure
plot(f)
grid on
```



6. По полученной выборке оценить параметры m и σ.

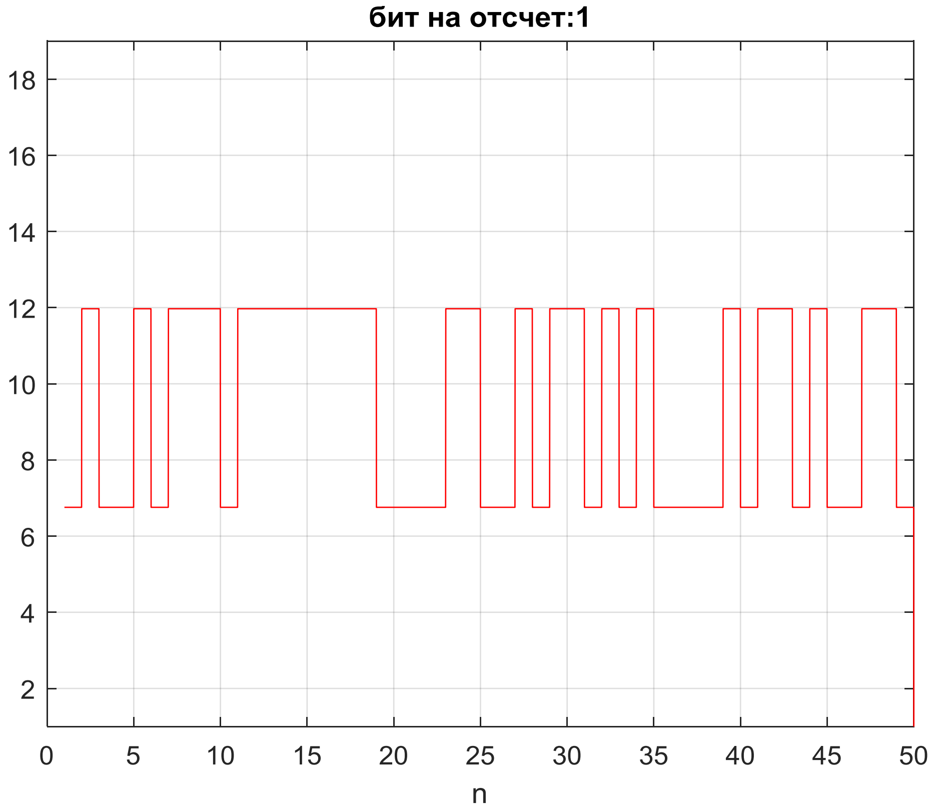
```
m = sum(f)/length(f)
sigma = sqrt(sum((f-m).^2)./(length(f)-1))

m = 9.363676851881763
sigma = 3.267639141797591
```

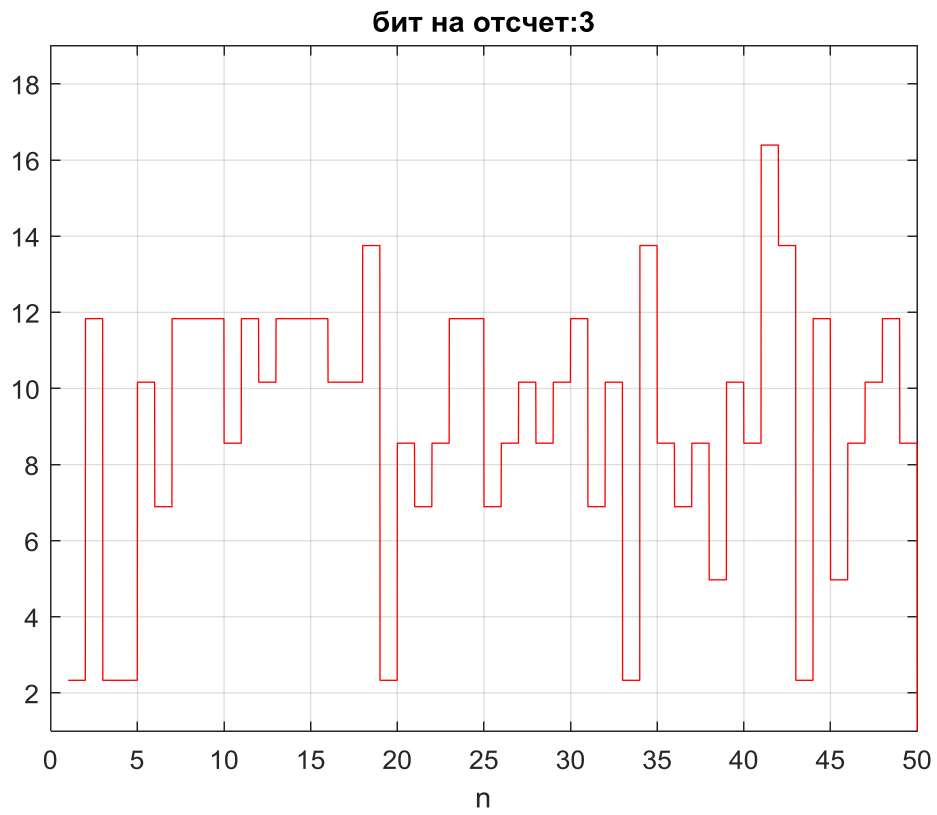
7. Определить параметры квантователя Ллойда-Макса.

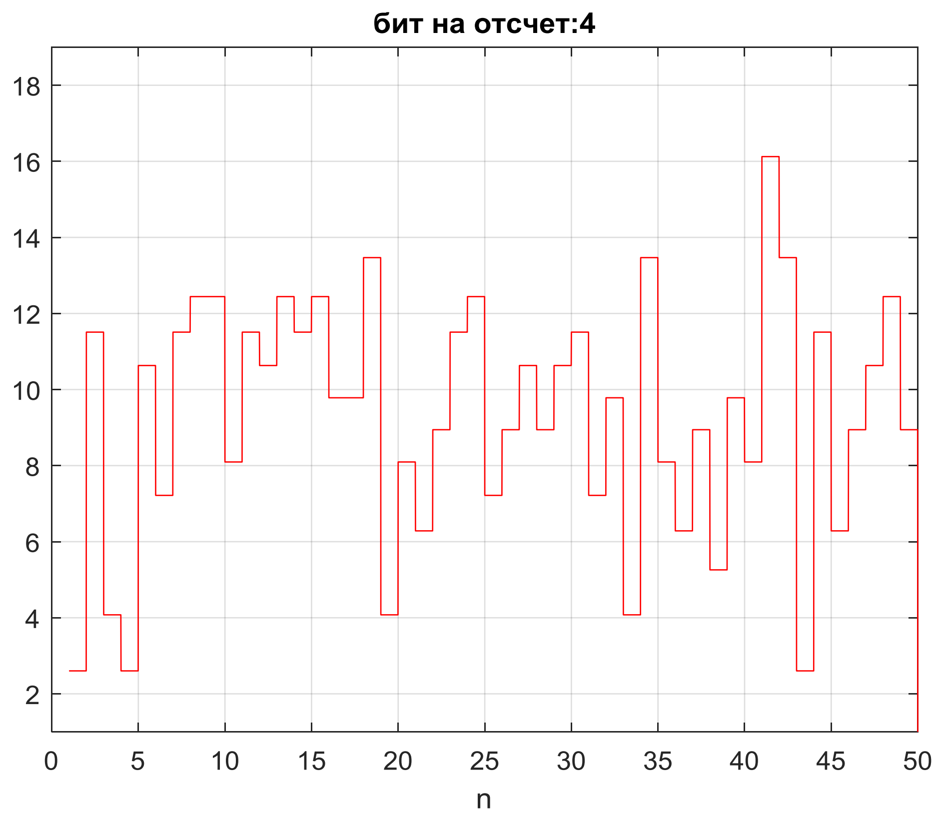
```
[quantf, vt, vd] = LloydMax(f, 0, 1, 4);
t = vt.*sigma + m
d = vd.*sigma + m
```

8. Выполнить оптимальное квантование сигнала x, используя от 1 до 4 бит на отсчет.

Код на языке Matlab в файле tasks.m в разделе part 3 task 8, 9.







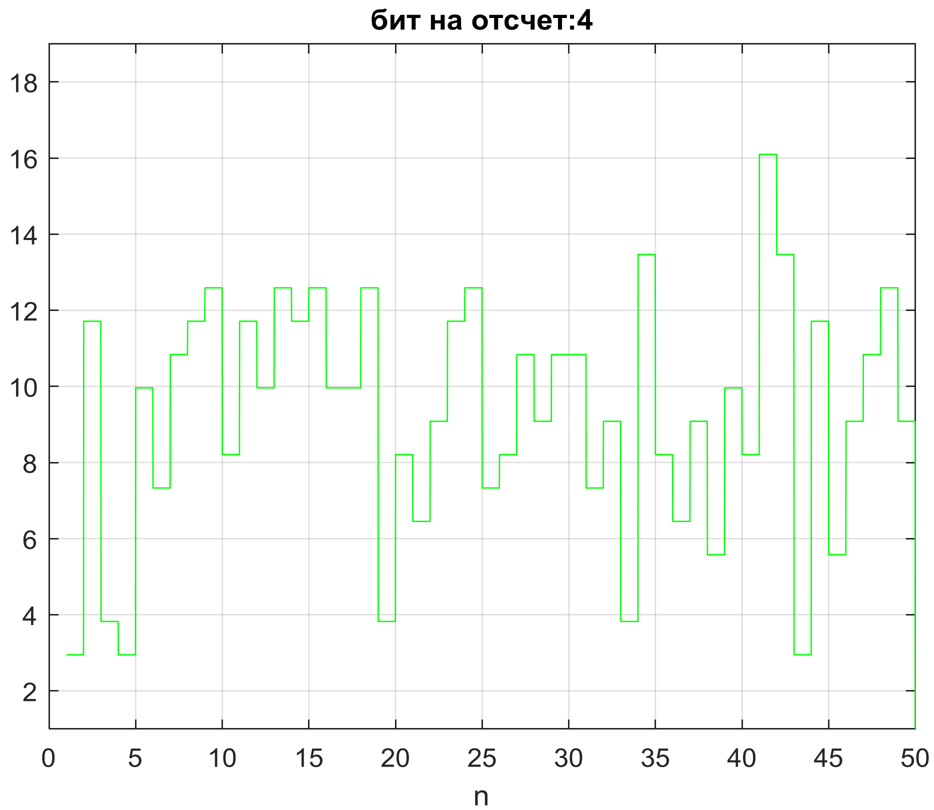
9. Вычислить выборочные значения ошибки (2), (6) и SNR.

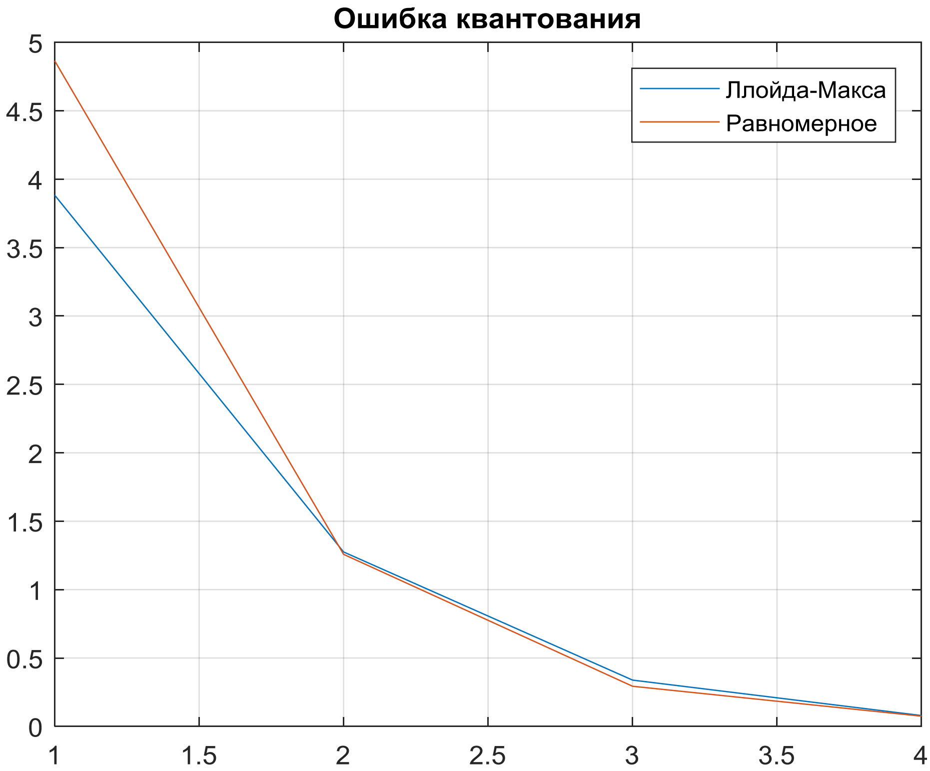
Код на языке Matlab в файле tasks.m в разделе part 3 task 8, 9.

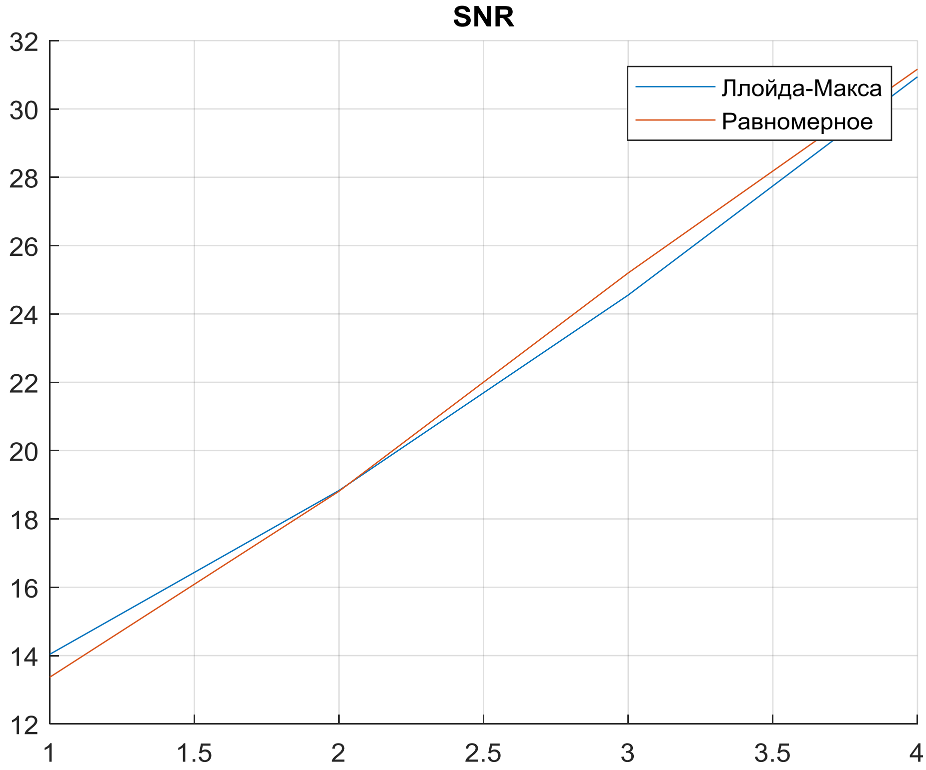
10. Выполнить равномерное квантование сигнала x при числе бит на отсчет от 1 до 4. Сравнить результат с полученным в предыдущем пункте.

Код на языке Matlab в файле tasks.m в разделе part 3 task 10.


бит на отсчет:3
18
16
14
12
10
8
6
4
2
0
5
10
15
20
25
30
35
40
45
50
n


бит на отсчет:4
18
16
14
12
10
8
6
4
2
0
5
10
15
20
25
30
35
40
45
50
n

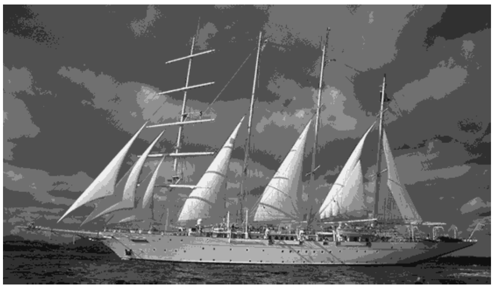




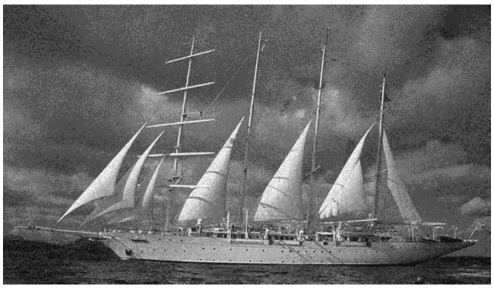
При меньшем количестве бит на отсчёт оптимальное квантование даёт меньшую ошибку, чем равномерное, при увеличении кол-ва бит, ошибки практически одинаково маленькие. Отношение сигнал-шум у обоих методов квантования практические одинаковое и возрастает по линейному закону.

11. Тестовое изображение из первой части лабораторной работы проквантовать с различным числом уровней яркости: L = 8, 16, 24, 32, 64, 128. При каком значении L ложные контура уже не наблюдаются.
12. Добавьте шум с равномерным распределением на промежутке [-q/2; q/2], где q = fmax/L – ширина интервалов равномерного квантования до квантования. Как изменился результат?

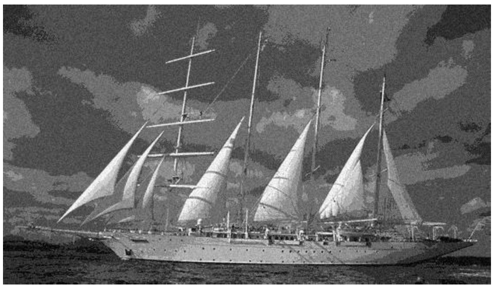
Количество уровней яркости:8
без шума



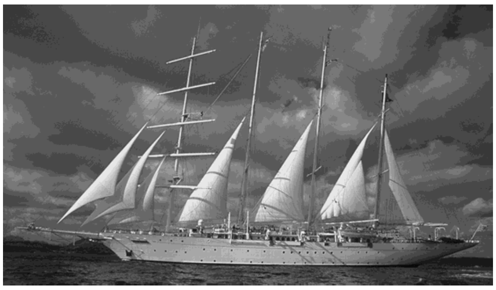
с шумом до квантования



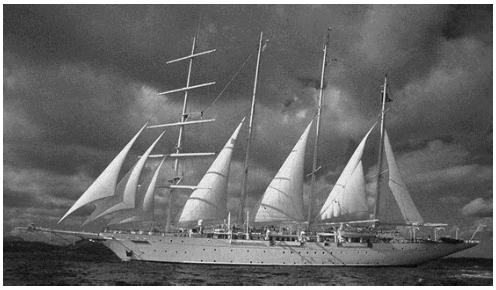
с шумом после квантования



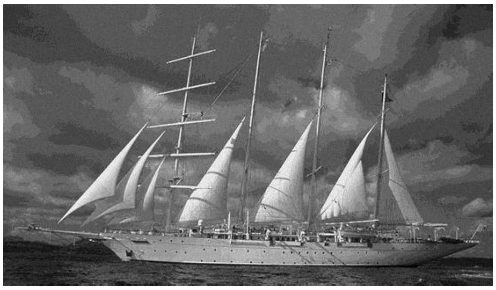
Количество уровней яркости:16
без шума



с шумом до квантования



с шумом после квантования



Количество уровней яркости:32
без шума

с шумом до квантования

с шумом после квантования

Количество уровней яркости:64
без шума

с шумом до квантования

с шумом после квантования

Количество уровней яркости:128
без шума

с шумом до квантования

с шумом после квантования

Код на языке Matlab в файле tasks.m в разделе part 3 task 11, 12, 13.

При значении L = 64 ложные контура не наблюдаются. При добавлении шума до квантования ложные контура не наблюдаются уже при L = 8.

13. Что будет, если шум добавлять после квантования?

Шум, добавленный после квантования, не сглаживает картинку, ложные контура перестают наблюдаться как и при отсутствии шума, когда $L \geq 64$.