REPORT

Zajęcia: Windowing

Teacher: prof. dr hab. Vasyl Martsenyuk

Lab 5

Date: 22.03.2024

Topic: "Quantization "

Variant: 13

Agnieszka Białecka

Informatyka II stopień,

stacjonarne,

1 semestr,

Gr.1a

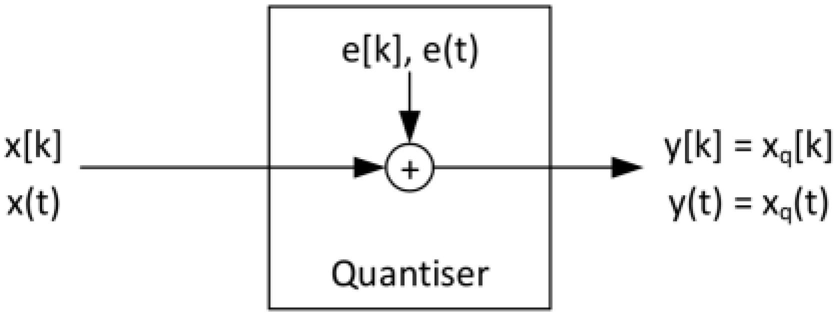
1. Abstract

The objective is to investigate Uniform Saturated Midtread Characteristic Curve of quantization.

2. Theoretical introduction

2.1. Quantization Process and Error

Quantization generates signals that have discrete values *xq[k]*, *xq(t)* from signals with continuous values *x[k]*, *x(t)*.
For quantization, the signals can be both, discrete and continuous in time. However, a signal that is discrete in time and discrete in value is termed a digital signal. Only digital signals can be processed by computers. Here the quantization of discrete-time signals is treated due to practical importance.
To describe quantization analytically, the model in the figure below is used.



The input and output signal differ by the quantization error (quantization noise) *e[k],* that is defined as

$$e[k] = x_q[k] - x[k],$$

so that the error constitutes an additive superposition

$$x[k] + e[k] = x_q[k]$$

To use this error model, some assumption should be made. The quantization noise shall be uniformly distributed, which then can be modeled with the probability density function (PDF) $p_e(\theta) = \frac{1}{\Delta Q} rect(\frac{\theta_e}{\Delta Q})$, where *ΔQ* denotes the quantization step size and θ_e the amplitudes of the quantization error signal.

2.2. Quantization Modeling / Mapping

The mapping of the infinitely large continuous set of values to a discrete number of amplitude steps is realized with a transfer characteristic. The height of the amplitude steps is *ΔQ.*

From the lecture, we know that the following mapping is used in order to quantize the continuous amplitude signal *x[k]* towards:

$$x_Q[k] = g(\ \lfloor f(x[k]) \rfloor\ ),$$

where $g(\cdot)$ and $f(\cdot)$ denote real-valued mapping functions, and $\lfloor \cdot \rfloor$ a rounding operation (not necessarily the plain floor operation).

Uniform Saturated Midtread Quantization Characteristic Curve

With the introduced mapping, the uniform saturated midtread quantizer can be discussed. This is probably the most important curve for uniform quantization due to its practical relevance for coding quantized amplitude values as bits. In general, the uniform midtread quantizer can be given as the mapping

$$x_Q[k] = \frac{1}{Q \backslash 2} \cdot \lfloor (Q \backslash 2) \cdot x[k] \rfloor,$$

where for $\lfloor \cdot \rfloor$ a rounding operation might used and \ denotes integer division. So, the mapping functions *g* and *f* are simple multiplications.

AD / DA Converter Convention

The case of even Q is practically used for virtually all analog/digital (AD) and digital/analog (DA) converters.

When additionally to the above statements

$$\log_2(Q) \in \mathbb{N}$$

holds, it is meaningful to code the even and power of two Q possible quantization steps with bits.

With $B \in N$ denoting the number of bits, the number range convention for AD and DA converters is:

$$-1 \leq x \leq 1 - 2^{-(B-1)}$$
$$-1 \leq x \leq 1 - \frac{2}{Q}$$

by using:

$$Q = 2^B$$

quantization steps. Values of *x* outside this range will be saturated to the minimum -1 and maximum 1- 2/Q quantization values in the quantization process.

Plotting the Midtread Curve

We now can visualize the characteristic curve for a simple, made up input signal, i.e. a monotonic increasing signal between $x_{max} = -x_{min}$ using an equidistant increment Δx over sample index *k*.

Here, we use x_{max} = 1.25 and Δx = 0.001 and assume that we start with x_{min} = -1.25 at k = 0. If Δx is sufficiently small, the signal's amplitude can be interpreted as continuous straight line. This straight line is degraded in a quantization process. Plotting the quantization result over the input, results in the characteristic curve, in our example in the curve of the uniform saturated midtread quantizer.

3. Input data (Variant)

This report was created with base of variant 13:
Ωc = t2 + t3

GitHub repository:
https://github.com/Delisolara/AaDEC

4. Course of actions

4.1. Packages installation and library import

```
!pip install soundfile
```

```
!pip install numpy
```

```
!pip install matplotlib
```

```
!pip install scipy
```

Picture 1.Packages installation

```python
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
from scipy import signal
import soundfile as sf
```

Picture 2. Uploaded libraries

```python
def my_xcorr2(x, y, scaleopt='none'):
    r""" Cross Correlation function phixy[kappa] -> x[k+kappa] y

    input:
    x   input signal shifted by +kappa
    y   input signal
    scaleopt   scaling of CCF estimator
    output:
    kappa   sample index
    ccf   correlation result
    """
    N = len(x)
    M = len(y)
    kappa = np.arange(0, N+M-1) - (M-1)
    ccf = signal.correlate(x, y, mode='full', method='auto')
    if N == M:
        if scaleopt == 'none' or scaleopt == 'raw':
            ccf /= 1
        elif scaleopt == 'biased' or scaleopt == 'bias':
            ccf /= N
        elif scaleopt == 'unbiased' or scaleopt == 'unbias':
            ccf /= (N - np.abs(kappa))
        elif scaleopt == 'coeff' or scaleopt == 'normalized':
            ccf /= np.sqrt(np.sum(x**2) * np.sum(y**2))
        else:
            print('scaleopt unknown: we leave output unnormalized')
    return kappa, ccf
```

Picture 3. Implemented code

```python
def uniform_midtread_quantizer(x, deltaQ):
    r"""uniform_midtread_quantizer from the lecture:
    https://github.com/spatialaudio/digital-signal-processing-lecture/blob/master/quantization/linear_uniform_quantization_error.ipynb
    commit: b00e23e
    note: we renamed the second input to deltaQ, since this is what the variable
    actually represents, i.e. the quantization step size

    input:
    x   input signal to be quantized
    deltaQ    quantization step size
    output:
    xq   quantized signal
    """
    # [-1...1) amplitude limiter
    x = np.copy(x)
    idx = np.where(x <= -1)
    x[idx] = -1
    idx = np.where(x > 1 - deltaQ)
    x[idx] = 1 - deltaQ
    # linear uniform quantization
    xq = deltaQ * np.floor(x/deltaQ + 1/2)
    return xq
```

Picture 4. Implemented code

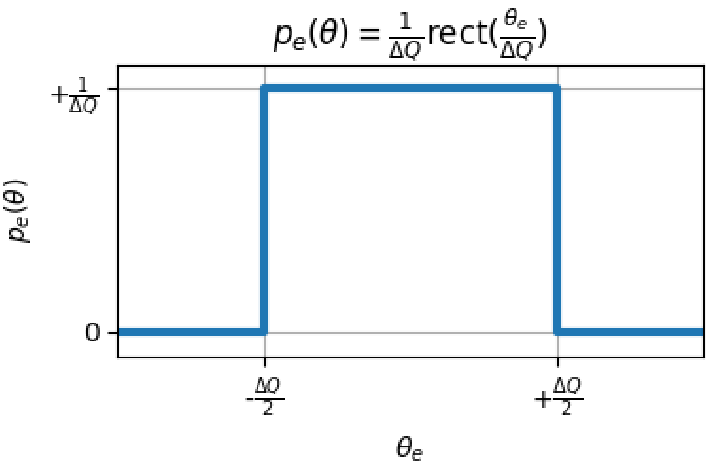
```
def my_quant(x, Q):
    r"""Saturated uniform midtread quantizer

    input:
    x  input signal
    Q  number of quantization steps
    output:
    xq quantized signal

    Note: for even Q in order to retain midtread characteristics,
    we must omit one quantization step, either that for lowest or the highest
    amplitudes. Typically the highest signal amplitudes are saturated to
    the 'last' quantization step. Then, in the special case of log2(N)
    being an integer the quantization can be represented with bits.
    """
    tmp = Q//2  # integer div
    quant_steps = (np.arange(Q) - tmp) / tmp

    # forward quantization, round() and inverse quantization
    xq = np.round(x*tmp) / tmp
    # always saturate to -1
    xq[xq < -1.] = -1.
    # saturate to ((Q-1) - (Q\2)) / (Q\2), note that \ is integer div
    tmp2 = ((Q-1) - tmp) / tmp
    xq[xq > tmp2] = tmp2
    return xq
```

Picture 5. Implemented code

4.2. Quantization Process and Error

```
plt.figure(figsize=(4, 2))
plt.plot((-1, -1/2, -1/2, +1/2, +1/2, +1), (0, 0, 1, 1, 0, 0), lw=3)
plt.xlim(-1, 1)
plt.ylim(-0.1, 1.1)
plt.xticks((-0.5, +0.5), [r'-$\frac{\Delta Q}{2}$', r'+$\frac{\Delta Q}{2}$'])
plt.yticks((0, 1), [r'0', r'+$\frac{1}{\Delta Q}$'])
plt.xlabel(r'$\theta_e$')
plt.ylabel(r'$p_e(\theta)$')
plt.title(
    r'$p_e(\theta) = \frac{1}{\Delta Q} \mathrm{rect}(\frac{\theta_e}{\Delta Q})$')
plt.grid(True)
```

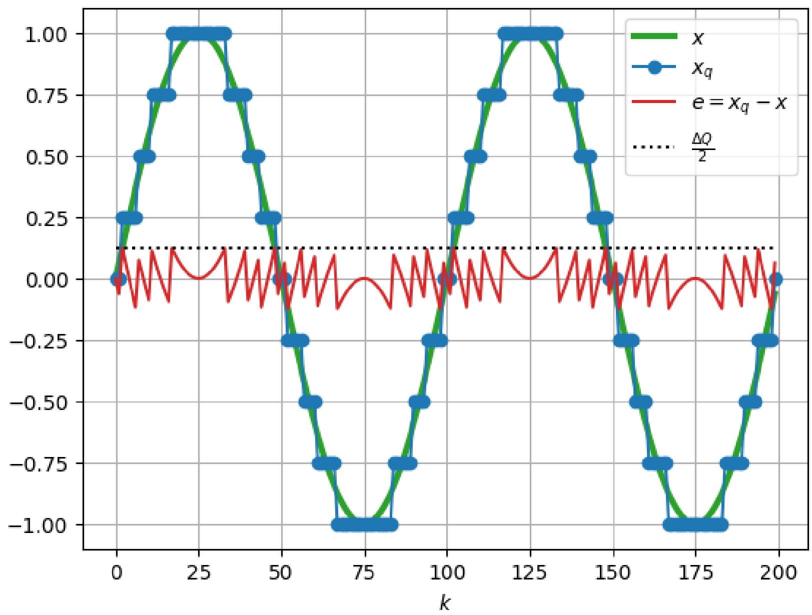
Picture 6. Implemented code



Picture 7. The result

```python
Q = 9  # odd, number of quantization steps
N = 100
k = np.arange(2*N)
x = np.sin(2*np.pi/N*k)
xq = my_quant(x, Q)
e = xq-x
# actually stem plots would be correct, for convenience we plot as line style
plt.plot(k, x, 'C2', lw=3, label=r'$x$')
plt.plot(k, xq, 'C0o-', label=r'$x_q$')
plt.plot(k, e, 'C3', label=r'$e=x_q-x$')
plt.plot(k, k*0+1/(Q-1), 'k:', label=r'$\frac{\Delta Q}{2}$')
plt.xlabel(r'$k$')
plt.legend()
plt.grid(True)
```

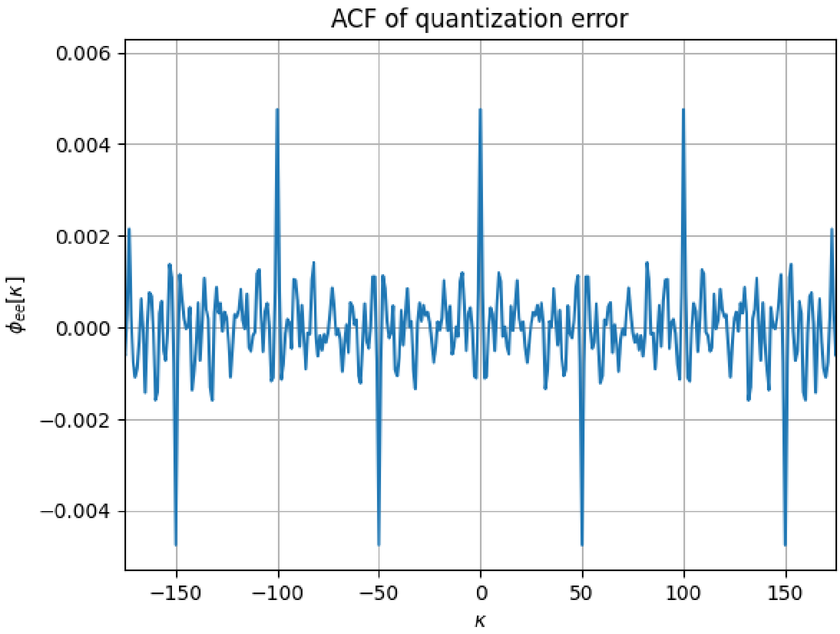
Picture 8. Implemented code



Picture 9. The result

```
kappa, acf = my_xcorr2(e, e, 'unbiased')
plt.plot(kappa, acf)
plt.xlim(-175, +175)
plt.xlabel(r'$\kappa$')
plt.ylabel(r'$\phi_{ee}[\kappa]$')
plt.title('ACF of quantization error')
plt.grid(True)
```

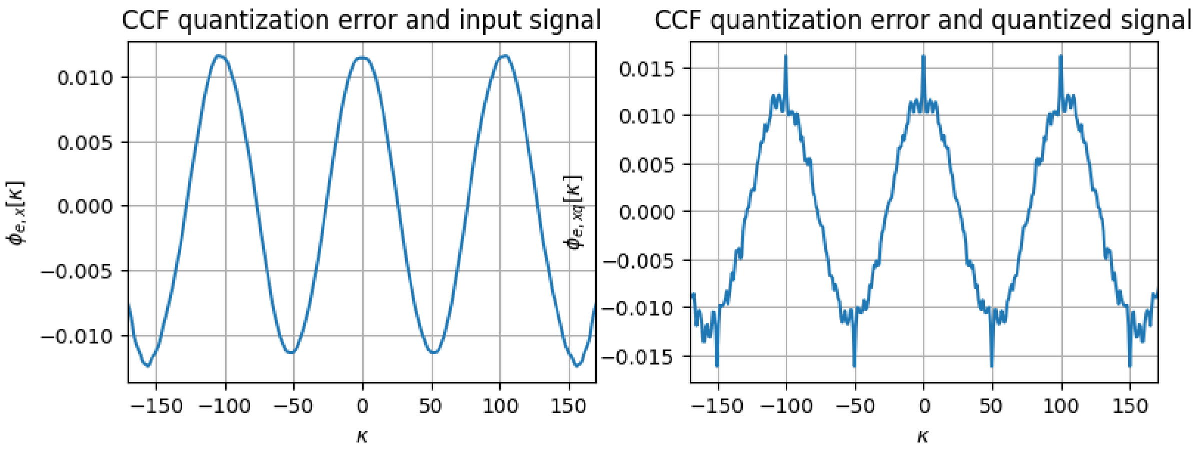
Picture 10. Implemented code



Picture 11. The result

```
plt.figure(figsize=(9, 3))

plt.subplot(1, 2, 1)
kappa, acf = my_xcorr2(e, x, 'unbiased')
plt.plot(kappa, acf)
plt.xlim(-170, +170)
plt.xlabel(r'$\kappa$')
plt.ylabel(r'$\phi_{e,x}[\kappa]$')
plt.title('CCF quantization error and input signal')
plt.grid(True)

plt.subplot(1, 2, 2)
kappa, acf = my_xcorr2(e, xq, 'unbiased')
plt.plot(kappa, acf)
plt.xlim(-170, +170)
plt.xlabel(r'$\kappa$')
plt.ylabel(r'$\phi_{e,xq}[\kappa]$')
plt.title('CCF quantization error and quantized signal')
plt.grid(True)
```

Picture 12. Implemented code



Picture 13. The result

4.3. AD / DA Converter Convention

```
B = 16  # number of bits
Q = 2**B  # number of quantization steps

# for even Q only:
deltaQ = 2/Q
# maximum quantize value:
xqmax = 1-2**(-(B-1))
# or more general for even Q:
xqmax = 1-deltaQ

print(' B = %d bits\n quantization steps Q = %d\n quantization step size %e' %
      (B, Q, deltaQ))
print(' smallest quantization value xqmin = -1')
print(' largest quantization value xqmax = %16.15f' % xqmax)
```

Picture 14. Implemented code

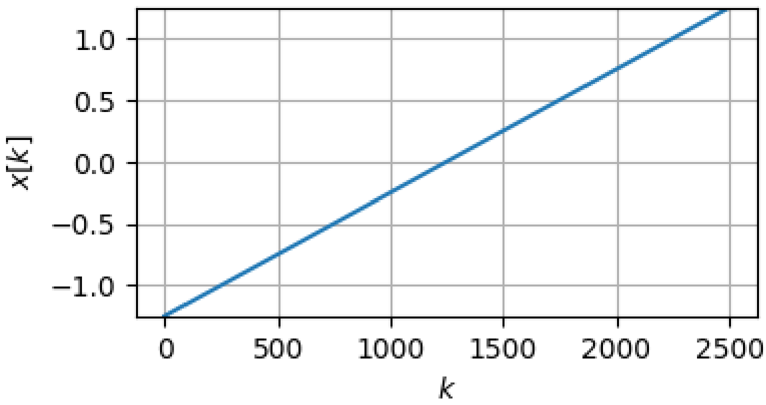
```
B = 16 bits
quantization steps Q = 65536
quantization step size 3.051758e-05
smallest quantization value xqmin = -1
largest quantization value xqmax = 0.999969482421875
```

Picture 15. The result

4.4. Plotting the Midtread Curve

```python
x = np.arange(-1.25, +1.25, 1e-3)
plt.figure(figsize=(4, 2))
plt.plot(x)  # actually a stem plot is correct
plt.ylim(-1.25, +1.25)
plt.xlabel(r'$k$')
plt.ylabel(r'$x[k]$')
plt.grid(True)
```

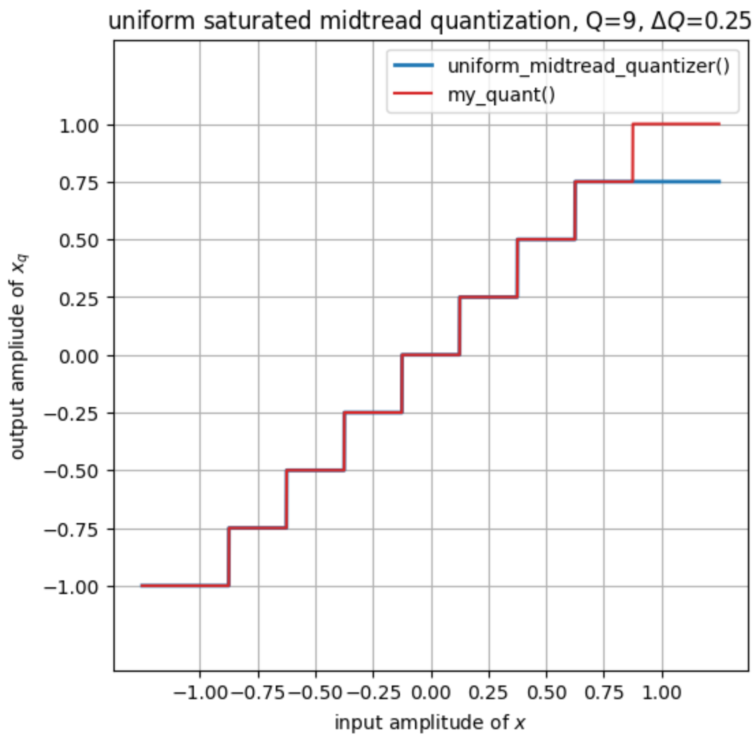
Picture 16. Implemented code



Picture 17. The result

```python
Q = 9  # number of quantization steps, odd or even
deltaQ = 1/(Q//2)  # quantization step size, even/odd Q

xq = my_quant(x, Q)  # used in exercise
xumq = uniform_midtread_quantizer(x, deltaQ)  # as used in lecture

plt.figure(figsize=(6, 6))
plt.plot(x, xumq, 'C0', lw=2, label='uniform_midtread_quantizer()')
plt.plot(x, xq, 'C3', label='my_quant()')
plt.xticks(np.arange(-1, 1.25, 0.25))
plt.yticks(np.arange(-1, 1.25, 0.25))
plt.xlabel(r'input amplitude of $x$')
plt.ylabel(r'output ampliude of $x_q$')
plt.title(
    r'uniform saturated midtread quantization, Q={0:d}, $\Delta Q$={1:3.2f}'.format(Q, deltaQ))
plt.axis('equal')
plt.legend()
plt.grid(True)
```

Picture 18. Implemented code



Picture 19. The result

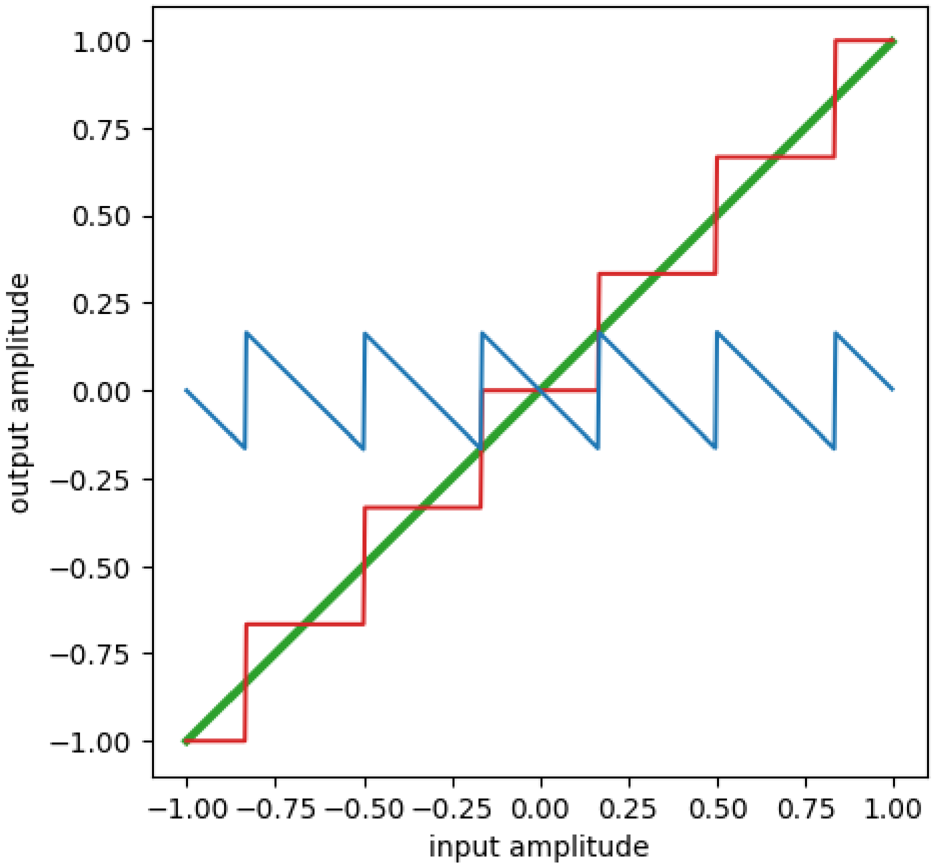
4.5. Exercise 1: Uniform Saturated Midtread Characteristic Curve of Quantization

```python
def check_my_quant(Q):
    N = 5e2
    x = 2*np.arange(N)/N - 1
    xq = my_quant(x, Q)
    e = xq - x

    plt.plot(x, x, color='C2', lw=3, label=r'$x[k]$')
    plt.plot(x, xq, color='C3', label=r'$x_q[k]$')
    plt.plot(x, e, color='C0', label=r'$e[k] = x_q[k] - x[k]$')
    plt.xticks(np.arange(-1, 1.25, 0.25))
    plt.yticks(np.arange(-1, 1.25, 0.25))
    plt.xlabel('input amplitude')
    plt.ylabel('output amplitude')
    if np.mod(Q, 2) == 0:
        s = ' saturated '
    else:
        s = ' '
    plt.title(
        '' % (Q, 1/(Q//2)))
    plt.axis('equal')
    plt.legend(loc='upper left')
    plt.grid(True)
```

Picture 20. Implemented code

```
Q = 7  # number of quantization steps
deltaQ = 1 / (Q//2)  # general rule
deltaQ = 2 / (Q-1)  # for odd Q only
plt.figure(figsize=(5, 5))
check_my_quant(Q)
```

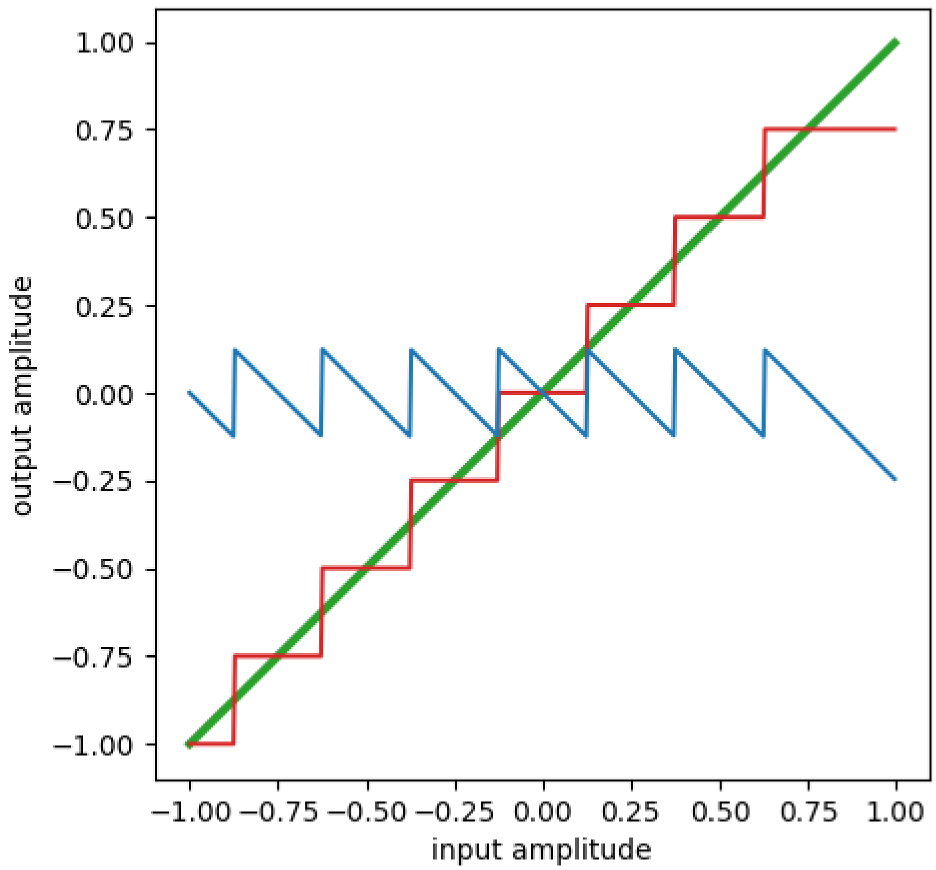
Picture 21. Implemented code



Picture 22. The result

```
Q = 8  # number of quantization steps
deltaQ = 1 / (Q//2)  # general rule
deltaQ = 2 / Q  # for even Q only
plt.figure(figsize=(5, 5))
check_my_quant(Q)
```

Picture 23. Implemented code



Picture 24. The result

5. Conclusions

In this lab we explored a concept of Uniform Saturated Midtread Characteristic Curve of quantization, which is a fundamental aspect of digital signal processing. Through our lab, we aimed to gain a deeper understanding of how quantization affects signals, particularly focusing on the characteristics of the midtread quantization curve.