Выполнено Шевченко Глебом, группа 214-322

**Практическое задание №3**

**Выбор параметров дискретизации сигнала**

Цель: Ознакомиться с понятием дискретизация и научиться дискретизировать непрерывную функцию изображения.

**Фотоаппарат**

Модель: Canon EOS 2000D kit 18-55mm DC

Размер матрицы: 22.3 х 14.9 мм

Разрешение: 24.7 Мп

**Частота дискретизации**:

frequency - частота дискретизации = 0.00366772681 \* 100

phase - фазовый сдвиг = 0  
y = 0.5 + 0.5\*sin(frequency \* x + phase)

**Дискретизация без фазового сдвига**

*Изображения дискретизированного сигнала*

|  |  |  |
| --- | --- | --- |
| **Сигнал** | **Изображение** | **Функция** |
| 1/5f |  |  |
| 1/3f |  |  |
| 1/2f |  |  |
| 1f |  |  |
| 1.41f |  |  |
| 2f |  |  |
| 3f |  |  |

**Частота дискретизации**:

frequency - частота дискретизации = 0.00366772681 \* 100

phase - фазовый сдвиг = pi  
y = 0.5 + 0.5\*sin(frequency \* x + phase)

**Дискретизация без фазового сдвига**

*Изображения дискретизированного сигнала*

|  |  |  |
| --- | --- | --- |
| **Сигнал** | **Изображение** | **Функция** |
| 1/5f |  |  |
| 1/3f |  |  |
| 1/2f |  |  |
| 1f |  |  |
| 1.41f |  |  |
| 2f |  |  |
| 3f |  |  |

**Листинг**

import numpy as np

import matplotlib.pyplot as plt

from math import pi

from PIL import Image, ImageDraw

import cv2

freq\_diskr=0.00366772681\*100

freq=[freq\_diskr/5, freq\_diskr/3, freq\_diskr/2, freq\_diskr, 1.41\*freq\_diskr, 2\*freq\_diskr, 3\*freq\_diskr]

phase = 0

#phase = pi

#Функция создания сигнала в виде синусоиды с возможность изменять частоту и сдвигать по фазе. Значения синусоиды в интервале [0; 1].

def generate\_sinusoid(x, frequency):

  return(0.5 + 0.5 \* np.sin(x \* frequency + phase))

#Функция, дискретизирующая сигнал с заданной частотой.

def discr\_sign(frequency):

  sas = np.array([])

  for i in np.arange(0, freq\_diskr\*3000, freq\_diskr):

      sas = np.append(sas, generate\_sinusoid(i, frequency))

  return sas

#Функция, которая создает изображение из дискретизированного сигнала.

def image(frequency):

  x=np.arange(0,freq\_diskr\*3000,freq\_diskr)

  plt.plot(x, generate\_sinusoid(x, frequency))

  plt.ylim(0,1)

  plt.show()

for elem in freq:

    image(elem)

def discr\_in\_image(row):

  row = np.around(255 \* row)

  image = np.full((row.size, row.size), row)

  return image

  # plt.plot(image)

  # plt.show()

img1 = discr\_in\_image(discr\_sign(freq[0]))

img2 = discr\_in\_image(discr\_sign(freq[1]))

img3 = discr\_in\_image(discr\_sign(freq[2]))

img4 = discr\_in\_image(discr\_sign(freq[3]))

img5 = discr\_in\_image(discr\_sign(freq[4]))

img6 = discr\_in\_image(discr\_sign(freq[5]))

img7 = discr\_in\_image(discr\_sign(freq[6]))

fig = plt.figure()

fig, ([ax1, ax2, ax3, ax4, ax5, ax6,ax7] ) = plt.subplots(

    nrows=1, ncols=7,

    figsize=(100, 100)

)

ax1.imshow(img1, cmap='gray')

ax2.imshow(img2, cmap='gray')

ax3.imshow(img3, cmap='gray')

ax4.imshow(img4,cmap='gray')

ax5.imshow(img5, cmap='gray')

ax6.imshow(img6, cmap='gray')

ax7.imshow(img7,cmap='gray')

plt.show()