**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”**

**ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ**

кафедра фізико–технічних засобів захисту інформації

# ПРОГРАМУВАННЯ 4

**ЗВІТ ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ №10**

**Виконав:**

Гелетей Віктор

Студент 2 курсу ФТІ

групи ФЕ-81

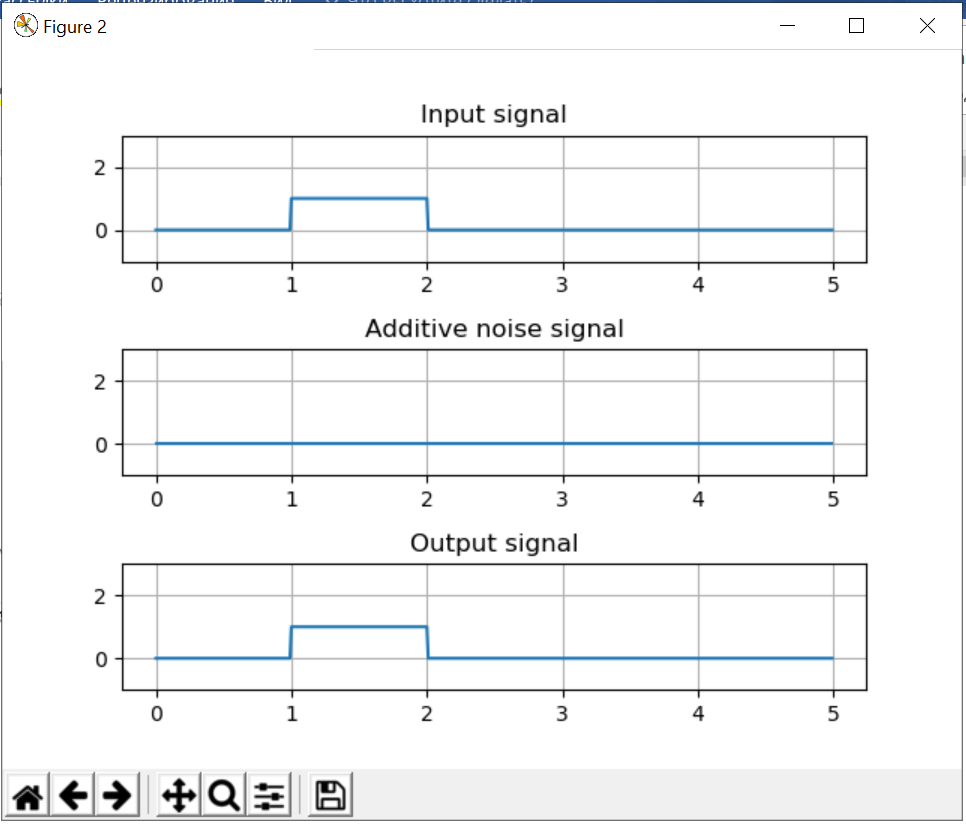
Київ

2020

|  |
| --- |
| **ЗАВДАННЯ**  **Моделирование канала связи с потерями (параметры линии задаются пользователем)**  import matplotlib.pyplot as plt |
| from matplotlib import pylab |
| from matplotlib.widgets import Button, Slider, RadioButtons |
| from numpy import arange |
| from random import uniform |
| from math import sin, pi |
|  |
| def Rectangular(x): |
| if x<1 or x>2: |
| return 0 |
| else: |
| return 1 |
|  |
| def sinus(x): |
| if x<0 or x>pi: |
| return 0 |
| else: |
| return sin(x) |
|  |
| def RectangularPlot(): |
| xlist = arange(0, 5.0, 0.01) |
| ylist = [Rectangular(x) for x in xlist] |
| return [xlist, ylist] |
|  |
| def noisePlot(tau, lvlomega): |
| xlist = arange(tau, tau + 5, 0.01) |
| ylist = [uniform(-lvlomega,lvlomega) for x in xlist] |
| return [xlist, ylist] |
|  |
| def sinPlot(): |
| xlist = arange(0, 5.0, 0.01) |
| ylist = [sinus(x) for x in xlist] |
| return [xlist, ylist] |
|  |
|  |
| def update(label): |
| if label == " Rectangular ": |
| menu.clear() |
| pr= RectangularPlot() |
| menu.plot(pr[0], pr[1]) |
| menu.grid() |
| pylab.draw() |
|  |
| if label == " Sinus ": |
| menu.clear() |
| pr=sinPlot() |
| menu.plot(pr[0], pr[1]) |
| menu.grid() |
| pylab.draw() |
|  |
| def onRadioButtonsClicked(label): |
| update(label) |
|  |
| if \_\_name\_\_ == '\_\_main\_\_': |
| def onButtonConfirmClicked(event): |
| global slider\_tau |
| global slider\_mu |
| global slider\_lvlomega |
| global menu |
| fig, ax = pylab.subplots() |
|  |
| pylab.subplot(3, 1, 1) |
| if radiobuttons\_type.value\_selected == ' Rectangular ': |
| pr = RectangularPlot() |
| if radiobuttons\_type.value\_selected == 'Синус': |
| pr = sinPlot() |
| plt.plot(pr[0], pr[1]) |
| plt.ylim(-1,3) |
| plt.grid() |
| plt.title('Input signal') |
|  |
| pylab.subplot(3, 1, 2) |
| no = noisePlot(slider\_tau.val, slider\_lvlomega.val) |
| plt.plot(no[0], no[1]) |
| plt.ylim(-1,3) |
| plt.grid() |
| plt.title('Additive noise signal') |
|  |
| pylab.subplot(3, 1, 3) |
| ou\_y = [] |
| for i in range(0,500): |
| ou\_y.append(pr[1][i]\*slider\_mu.val + no[1][i]) |
| plt.plot(no[0], ou\_y) |
| plt.ylim(-1,3) |
| plt.grid() |
| plt.title('Output signal') |
|  |
| fig.subplots\_adjust(hspace = 0.7) |
| pylab.show() |
|  |
| fig, menu = pylab.subplots() |
| update("Прямоугольный") |
|  |
| fig.subplots\_adjust(left=0.07, right=0.95, top=0.9, bottom=0.4) |
|  |
| axes\_button\_Confirm = pylab.axes([0.7, 0.05, 0.25, 0.075]) |
| button\_Confirm = Button(axes\_button\_Confirm, 'Confirm') |
| button\_Confirm.on\_clicked(onButtonConfirmClicked) |
|  |
| axes\_slider\_tau = pylab.axes([0.05, 0.25, 0.85, 0.04]) |
| slider\_tau = Slider(axes\_slider\_tau, |
| label='τ', |
| valmin=0.0, |
| valmax=200.0, |
| valinit=0.0, |
| valfmt='%1.2f') |
| axes\_slider\_mu = pylab.axes([0.05, 0.20, 0.85, 0.04]) |
| slider\_mu = Slider(axes\_slider\_mu, |
| label='μ', |
| valmin=0.0, |
| valmax=1.0, |
| valinit=1.0, |
| valfmt='%1.2f') |
| axes\_slider\_lvlomega = pylab.axes([0.05, 0.15, 0.85, 0.04]) |
| slider\_lvlomega = Slider(axes\_slider\_lvlomega, |
| label='lvl', |
| valmin=0.0, |
| valmax=1.0, |
| valinit=0.0, |
| valfmt='%1.2f') |
|  |
|  |
| axes\_radiobuttons = pylab.axes([0.05, 0.012, 0.27, 0.12]) |
| radiobuttons\_type = RadioButtons(axes\_radiobuttons, [' Rectangular ', ' Sinus ']) |
| radiobuttons\_type.on\_clicked(onRadioButtonsClicked) |

**Виконання програми**

Изображение выглядит как снимок экрана

Автоматически созданное описание Изображение выглядит как снимок экрана

Автоматически созданное описание Изображение выглядит как снимок экрана

Автоматически созданное описаниеИзображение выглядит как снимок экрана

Автоматически созданное описание Изображение выглядит как снимок экрана

Автоматически созданное описание

Изображение выглядит как снимок экрана

Автоматически созданное описание