**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информатика»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 8

по дисциплине «Математическое моделирование сложных систем»

на тему: **«Моделирование динамических систем методами САУ»**

Выполнил: студент гр. ИП-32

Кардаш Е.В.

Принял: преподаватель

Трохова Т.А.

Гомель 2021

**Цель работы:** Получить навыки моделирования САУ с использованием Python и в пакете Xcos, научиться применять функции Python для анализа моделей САУ, выполнять графическую интерпретацию полученных результатов.

**Практическая часть**

1. Создать модели типовых звеньев САУ:

- интегрирующего,

- дифференцирующего (реального),

- апериодического второго порядка,

- колебательного.

Параметры типовых звеньев задаются самостоятельно.

1. Для каждого звена определить переходную функцию (построить графики переходного процесса).

**Ход решения:**

**Листинг задания Task1.py:**

from control.matlab import \*

import matplotlib.pyplot as plt

k = 1

w = tf([k],[1,0])

y,x = step(w)

plt.plot(x,y)

plt.grid()

plt.title("Интегрирующая")

plt.figure()plt.show()

**Результат выполнения задания**

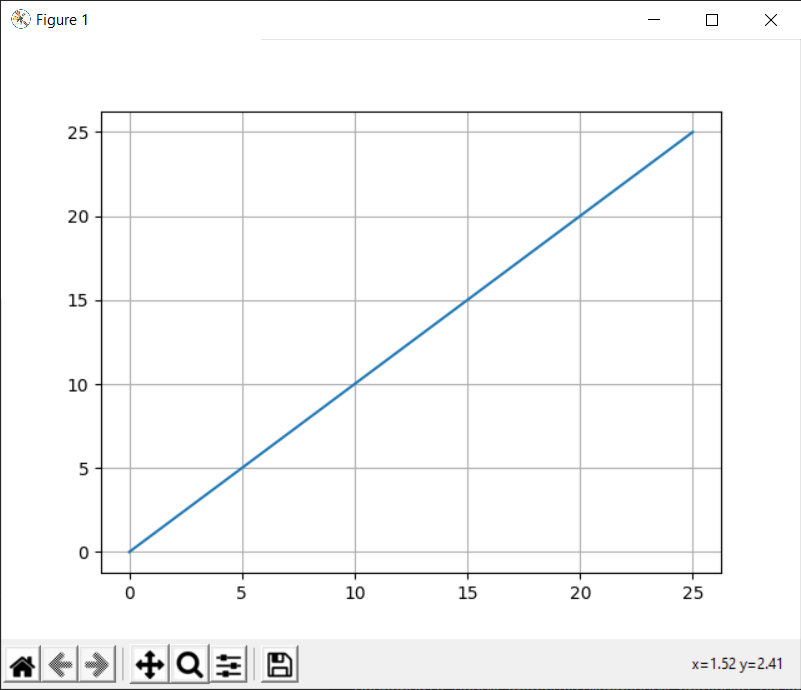


Рисунок 1 – График функции интегрирующего СЛАУ



Рисунок 2 – Полученная функция интегрирующего СЛАУ

**Листинг задания t12.py:**

from control.matlab import \*

import matplotlib.pyplot as plt

k = 1

w = tf([k,0],[1,1])

y,x = step(w)

print(w)

plt.plot(x,y)

plt.grid()

plt.show()

**Результат выполнения задания:**

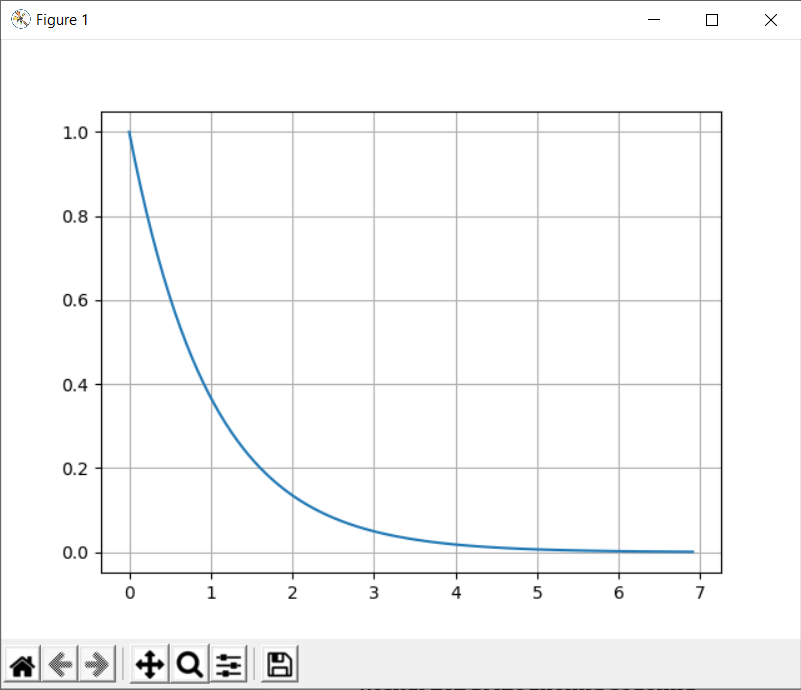


Рисунок 3 – График функции дифференцирующего СЛАУ



Рисунок 4 – Полученная функция дифференцирующего СЛАУ

**Листинг задания t13.py:**

from control.matlab import \*

import matplotlib.pyplot as plt

K = 1

T1 = 1.5

T2 = 6

w = tf([K], [T1 \*\* 2, T2, 1])

y, x = step(w)

print(w)

plt.plot(x, y)

plt.grid()

plt.show()

**Результат выполнения задания:**

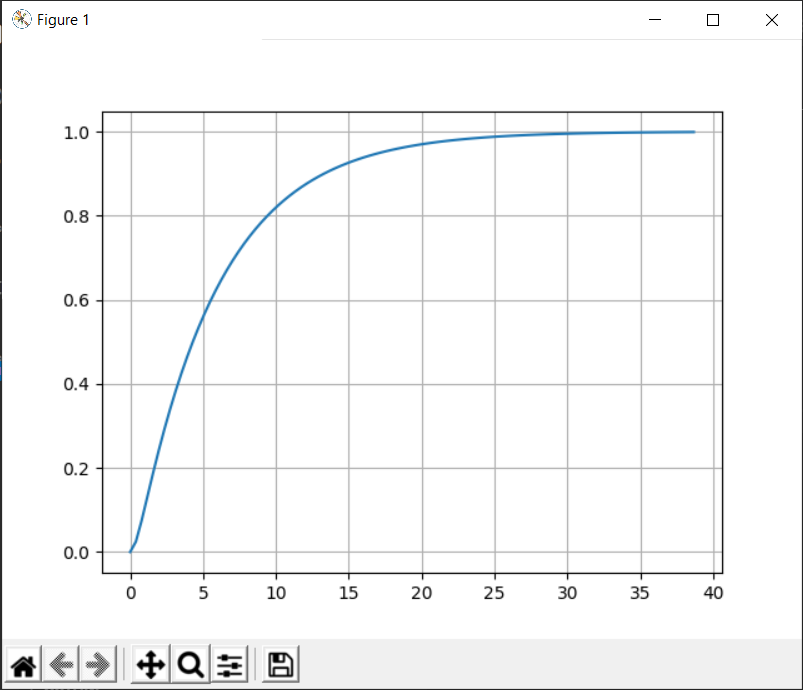


Рисунок 5 – График функции апериодического второго порядка СЛАУ

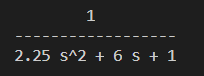


Рисунок 6 – Полученная функция апериодического второго порядка СЛАУ

**Листинг задания t14.py:**

from control.matlab import \*

import matplotlib.pyplot as plt

k = 1

t1 = 5

t2 = 1.6

w = tf([k],[t1\*\*2,2\*t2,1])

y,x = step(w)

print(w)

plt.plot(x,y)

plt.grid()

plt.show()

**Результат выполнения задания:**

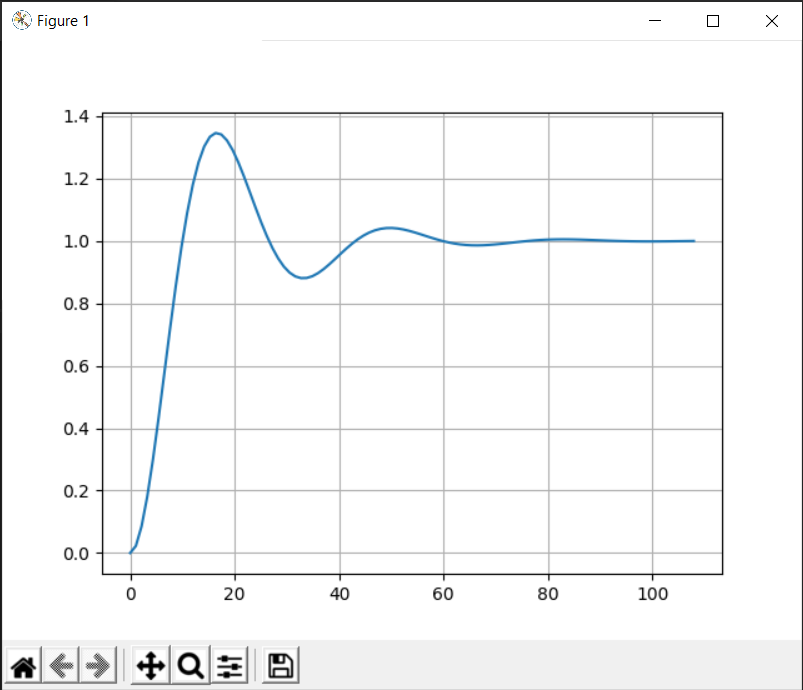


Рисунок 7 – График функции колебательного звена СЛАУ

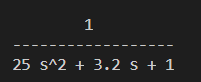


Рисунок 8 – График функции колебательного звена СЛАУ

1. Для колебательного звена получить переходную характеристику с использованием Xcos. Сравнить переходную характеристику с полученной в п.2

**Результат выполнения задания:**

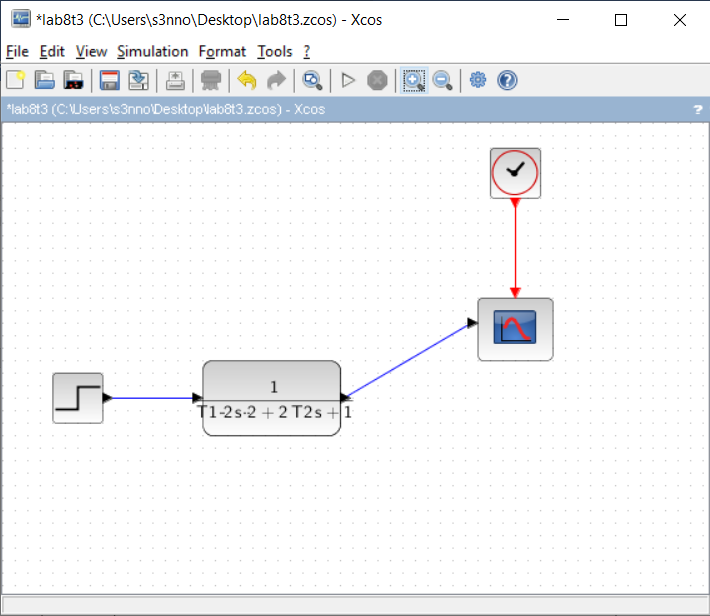


Рисунок 9 – Схема функции колебательного звена

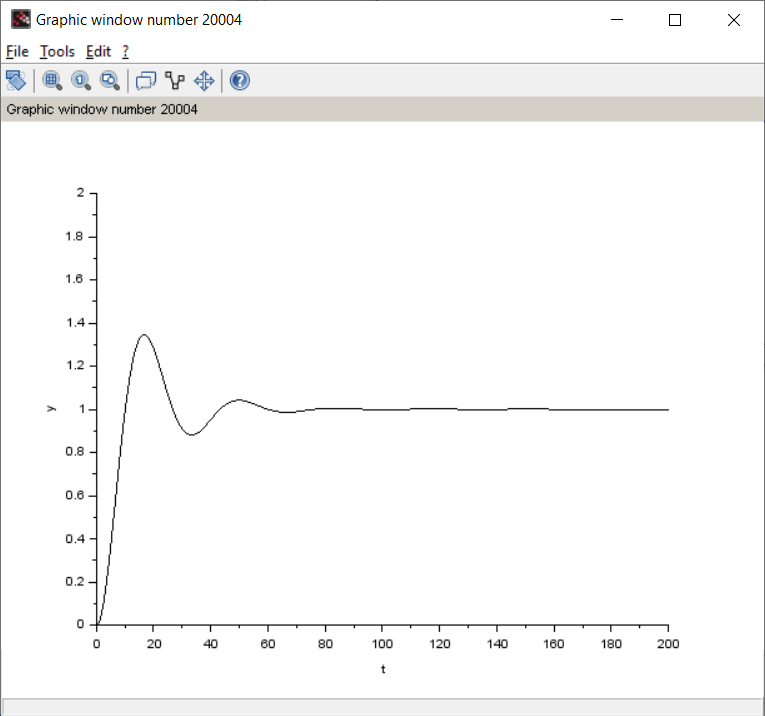
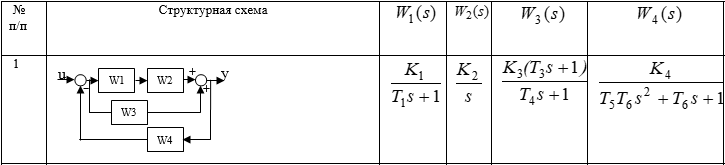


Рисунок 10 – Полученный график коллебательного звена

1. С использованием Python и Xcos создать модель, вид которой приведен в приложении А. Получить временную характеристику модели в Python и Xcos, сравнить их. Параметры системы подобрать самостоятельно.



**Листинг задания t14.py:**

from control.matlab import \*

import matplotlib.pyplot as plt

K1=1

K2=1

K3=1

K4=1

T1=0.1

T3=0.3

T4=0.4

T5=0.5

T6=0.6

W1=tf([K1],[T1, 1])

W2=tf([K2],[1, 0])

W3=tf([K3\*T3, K3],[T4, 1])

W4=tf([K4],[T5\*T6,T6,1])

W5=W1\*W2

W6=(W5+W3)

W7=feedback(W6,W4)

print(W7)

num= [0., 1.]

den= [1., 2., 10.]

w= tf(num, den)

y,x=step(W7)

plt.plot(x,y)

plt.grid(True)

plt.show()

**Результат выполнения задания:**

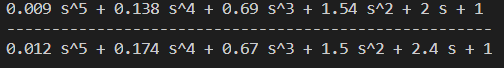


Рисунок 11– Полученное уравнение модели

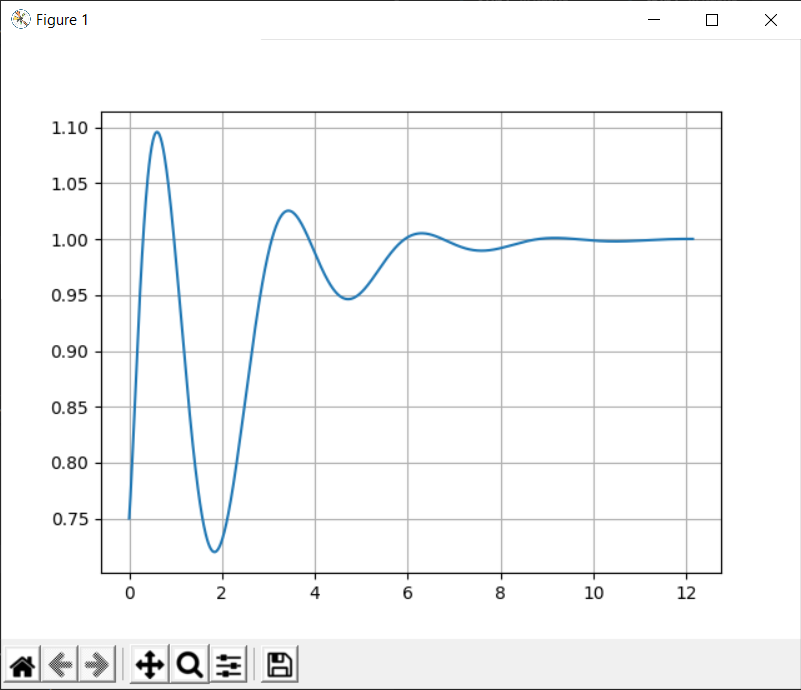


Рисунок 12– График уравнения модели в Python

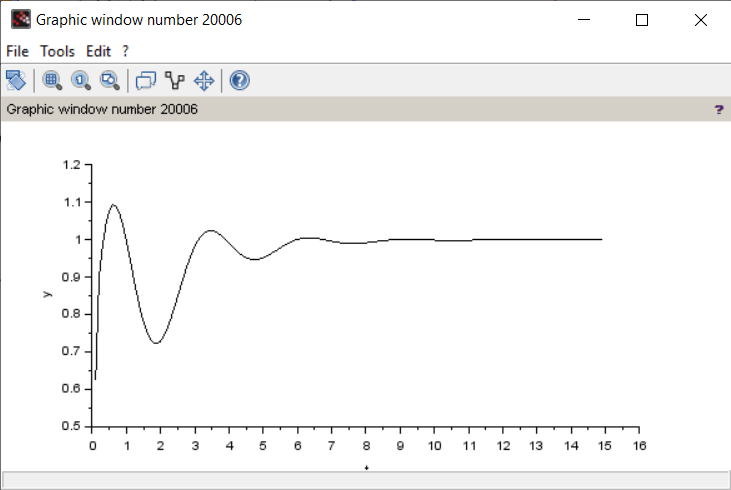


Рисунок 13– График уравнения модели в Xcos

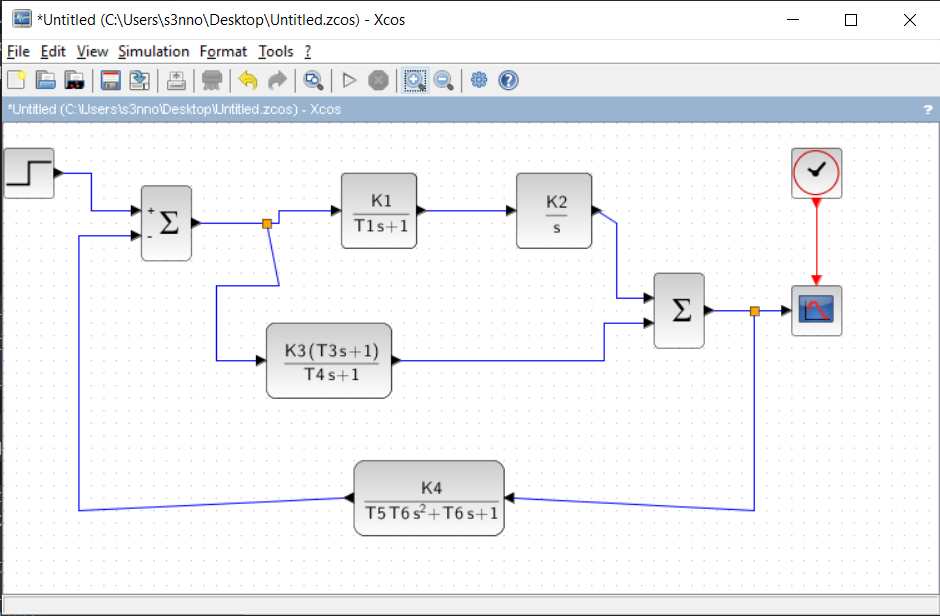


Рисунок 14– Схема уравнения модели в Xcos

1. Для колебательного звена получить амплитудно-частотную характеристику и построить диаграмму Bode.
2. Для колебательного звена определить по АЧХ значение максимальной амплитуды и частоты, при которой она достигается, исследовать явление резонанса.

**Листинг задания t56.py:**

from control.matlab import \*

import matplotlib.pyplot as plt

import numpy as np

K = 1

T1 = 5

T2 = 1.6

w = tf([K], [T1 \*\* 2, 2 \* T2 / (2 \* T1) \* T1, 1])

y, x = step(w)

maxA = 0.0

maxW = 0.0

mag, phase, omega = bode(w)

for i in range(len(mag)):

    if(mag[i] == max(mag)):

        maxA = mag[i]

        maxW = omega[i]

print("Максимальная частота = ", maxW)

print("Частота при этом значении = ", maxW)

plt.grid()

plt.figure(2)

plt.grid()

plt.plot(omega,mag)

maxi = np.argmax(mag)

plt.scatter(omega[maxi],mag[maxi])

print(omega[maxi],mag[maxi])

plt.show()

**Результат выполнения задания:**

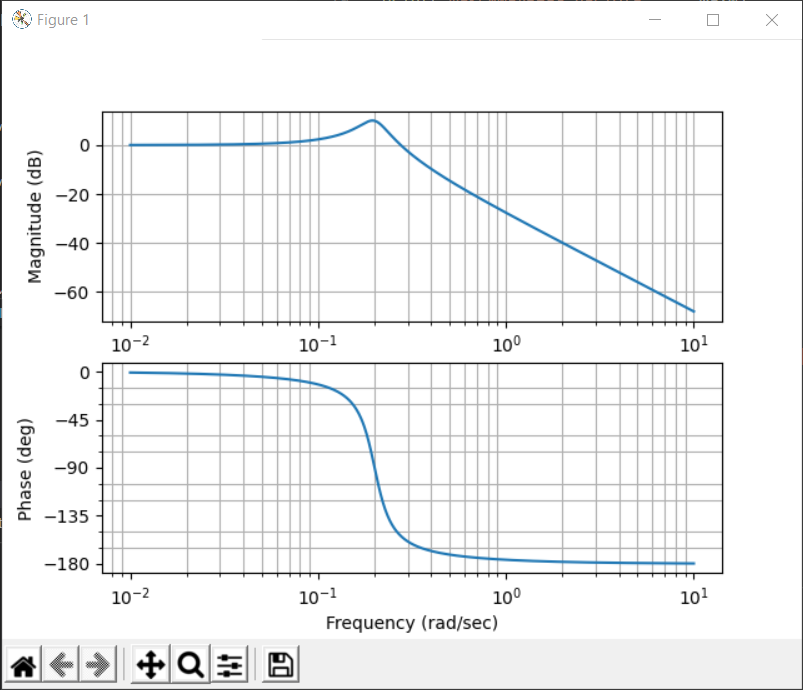
****

Рисунок 15– График модели Bode

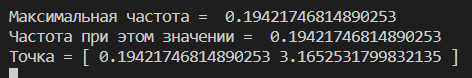
****

Рисунок 16– Результаты исследования функции

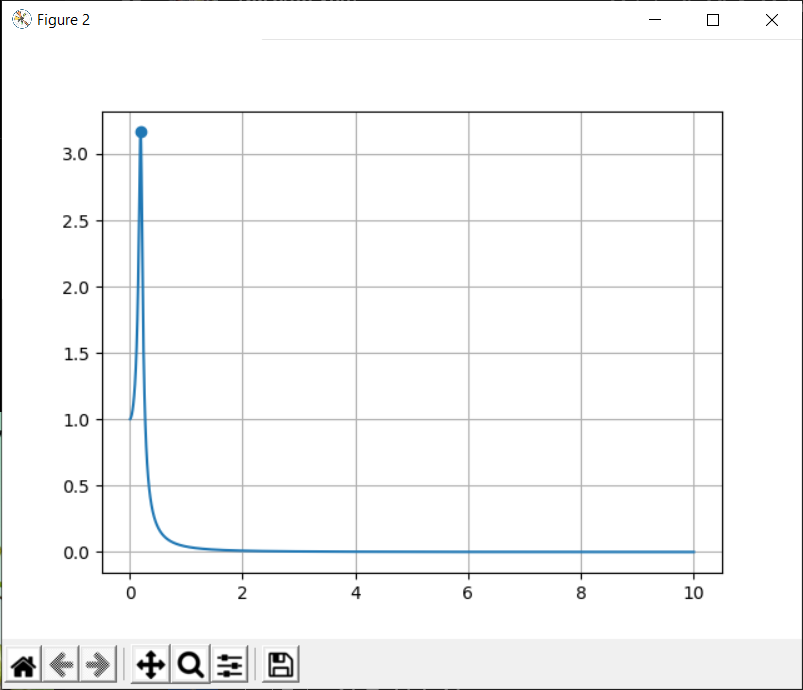
****

Рисунок 17 – АЧХ модели

1. Система описывается передаточной функцией вида:



Необходимо подобрать такие значения A, B, C, чтобы система была

а) устойчива с колебаниями и без колебаний,

б) неустойчива с колебаниями и без колебаний.

Построить график переходной характеристики, подтверждающие правильность выбора коэффициентов.

Сделать вывод о том, как влияет параметр K на вид переходной характеристики системы.

**Листинг задания t7.py:**

from control.matlab import \*  
from control import \*  
import matplotlib.pyplot as plt  
  
K=20  
A=1  
B=5  
C=5  
  
function = tf([K],[A,B,C])  
coeff = pole(function)  
  
*print*(coeff)  
y,x = step(function)  
  
string = "Устойчива без колебаний: "+ *str*(coeff)  
plt.figure(0)  
plt.title(string)  
plt.plot(x,y)  
  
K=10  
A=5  
B=1  
C=1  
  
function = tf([K],[A,B,C])  
coeff = pole(function)  
  
*print*(coeff)  
y,x = step(function)  
  
string = "Устойчива c колебаниями: "+ *str*(coeff)  
  
plt.figure(1)  
plt.title(string)  
plt.plot(x,y)  
  
  
  
K=10  
A=1  
B=-1  
C=1  
  
function = tf([K],[A,B,C])  
coeff = pole(function)  
  
*print*(coeff)  
y,x = step(function)  
  
string = "Неустойчива c колебаниями: "+ *str*(coeff)  
  
plt.figure(2)  
plt.title(string)  
plt.plot(x,y)  
  
K=5  
A=1  
B=-3  
C=1  
  
function = tf([K],[A,B,C])  
coeff = pole(function)  
  
*print*(coeff)  
y,x = step(function)  
  
string = "Неустойчива без колебаний: "+ *str*(coeff)  
  
plt.figure(3)  
plt.title(string)  
plt.plot(x,y)  
  
plt.show()

**Результат выполнения задания:**

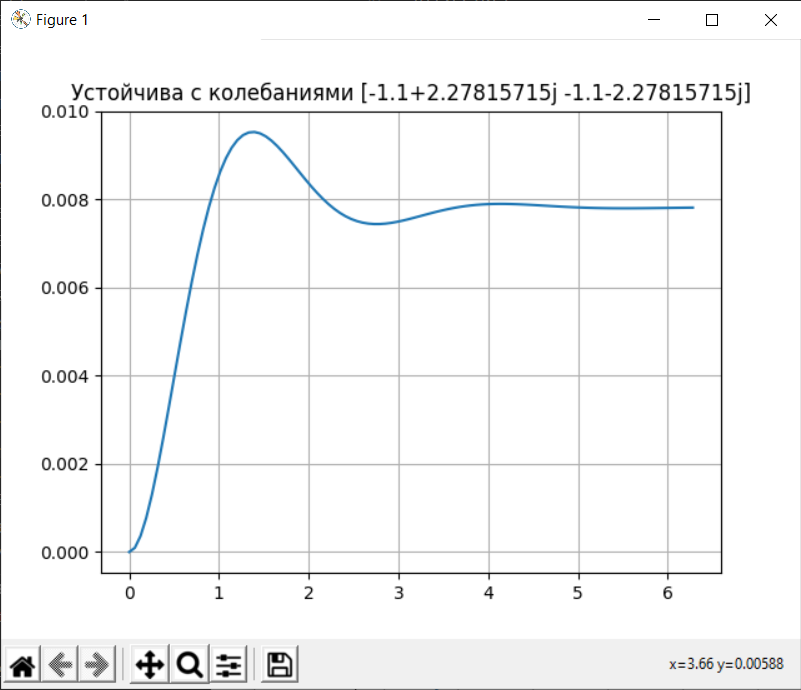


Рисунок 18 – График модели устойчивой с колебаниями

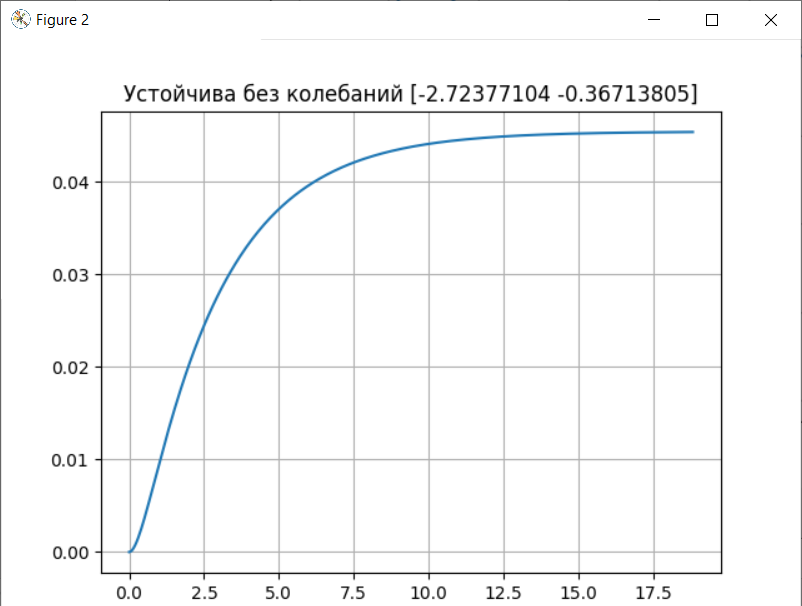


Рисунок 19 – График модели устойчивой без колебаниями

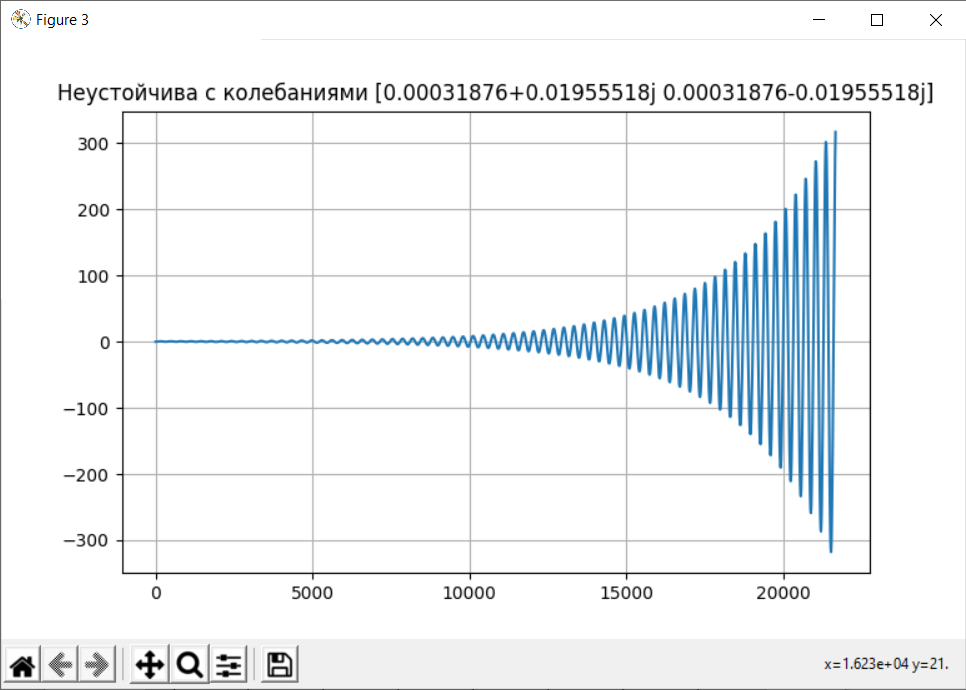


Рисунок 20 – График модели неустойчивой с колебаниями

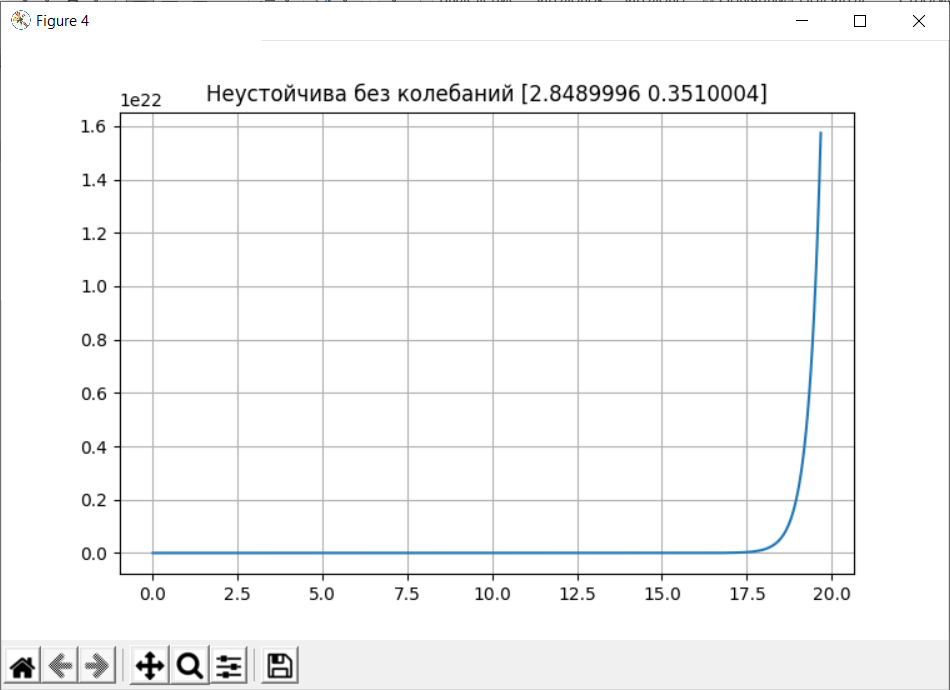


Рисунок 21 – График модели неустойчивой без колебаний

**Вывод:** в результате выполнения лабораторной работы были получены навыки моделирования САУ с использованием Python и в пакете Xcos, изучены функции Python для анализа моделей САУ, выполнена графическая интерпретация результатов.