**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»**

Выполнила: студентка гр. ИП-31

Архипенко М.А.

Принял: преподаватель

Трохова Т.А.

Гомель 2020

**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.

**Порядок выполнения работы**

Изучить теоретические сведения по конспекту лекций.

**Практическая часть**

1. Создать модели типовых звеньев САУ:

- интегрирующего,

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.show()

- дифференцирующего (реального),

from control.matlab import \*

import matplotlib.pyplot as plt

k = 1

w = tf([k,0],[1,1])

y,x = step(w)

plt.plot(x,y)

plt.grid()

plt.show()

- апериодического второго порядка,

from control.matlab import \*

import matplotlib.pyplot as plt

K = 1

T1 = 7

T2 = 2

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

y, x = step(w)

plt.plot(x, y)

plt.grid()

plt.show()

- колебательного.

k = 1

t1 = 5

t2 = 1.6

w = tf([k],[t1\*\*2,2\*t2,1])

y,x = step(w)

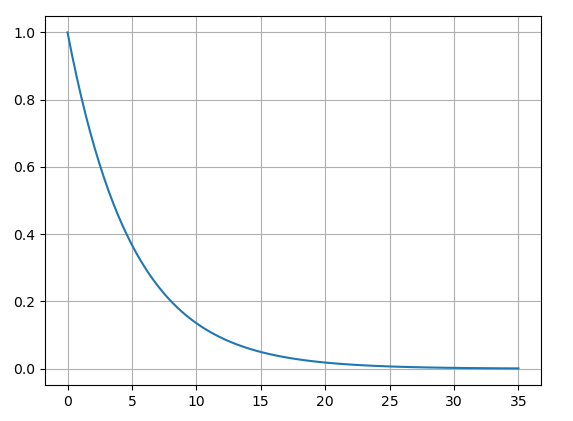
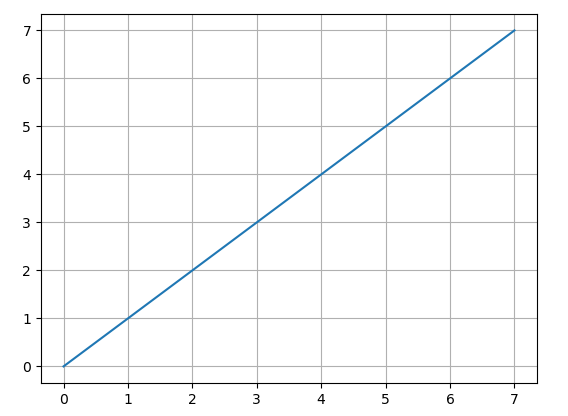
plt.plot(x,y)

plt.grid()

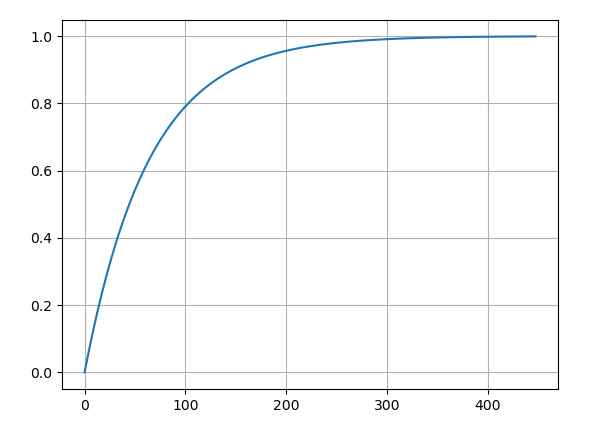
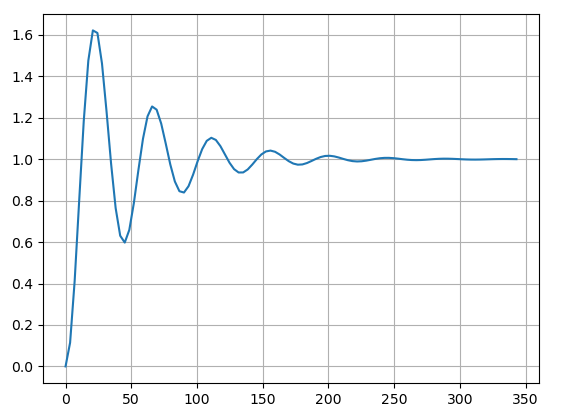
plt.show()

Параметры типовых звеньев задаются самостоятельно.

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

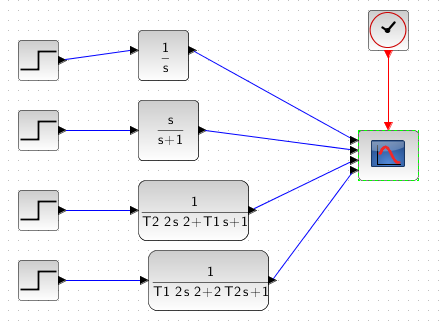


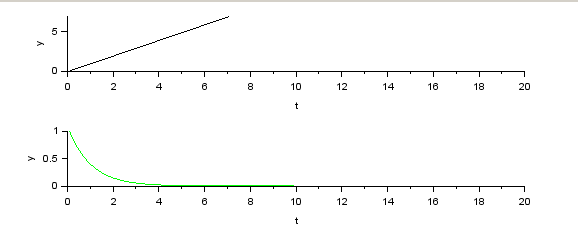
Интегрирующее звено Дифференцирующее (реальное) звено

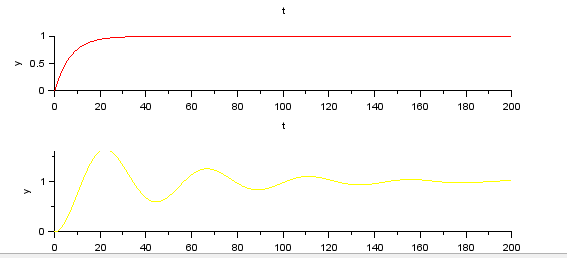
 

Апериодическое звено второго порядка Колебательное звено

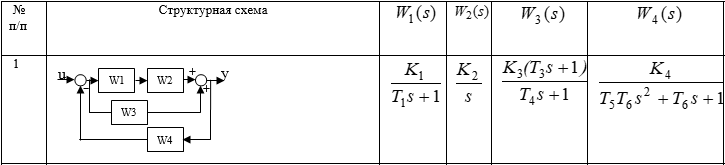
1. Для колебательного звена получить переходную характеристику с использованием Xcos. Сравнить переходную характеристику с полученной в п.2

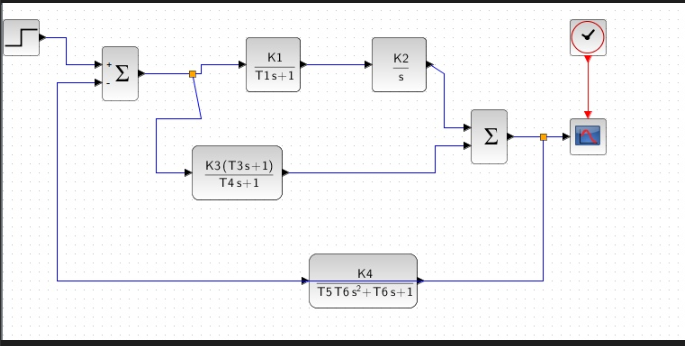


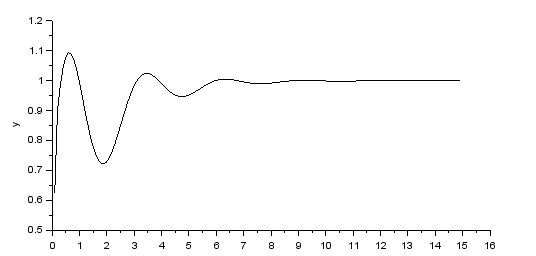




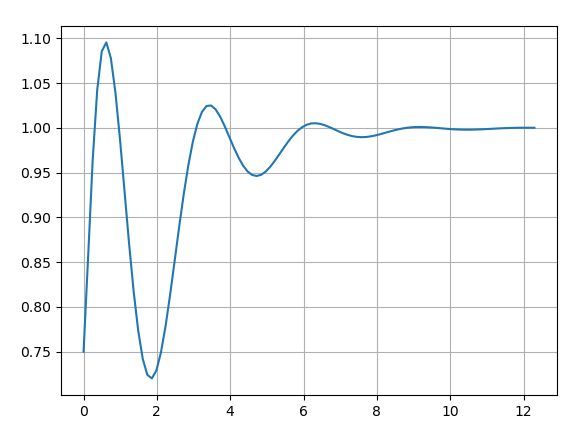
1. С использованием Python и Xcos создать модель, вид которой приведен в приложении А. Получить временную характеристику модели в Python и Xcos, сравнить их. Параметры системы подобрать самостоятельно.







Python



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)

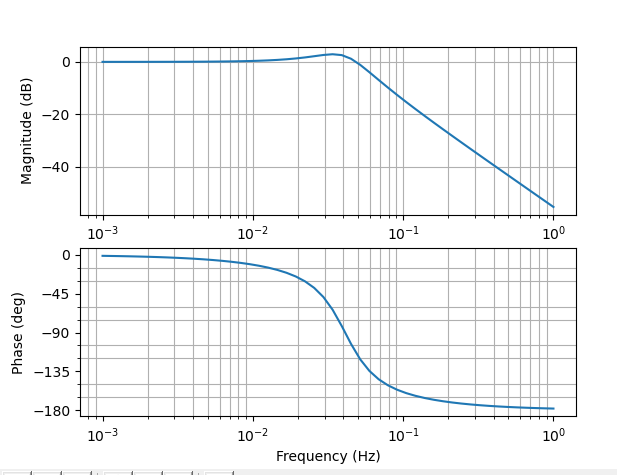
y,x=step(W7)

plt.plot(x,y)

plt.grid(True)

plt.show()

1. Для колебательного звена получить амплитудно-частотную характеристику и построить диаграмму Bode.



1. Для колебательного звена определить по АЧХ значение максимальной амплитуды и частоты, при которой она достигается, исследовать явление резонанса.

Листинг программы 5-6:

from control.matlab import \*

import matplotlib.pyplot as plt

K = 1

T1 = 5

T2 = 1.6

ro = T2 / (2 \* T1)

w = tf([K], [T1 \*\* 2, 2 \* ro \* 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.show()



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 программы:

from control.matlab import \*

import matplotlib.pyplot as plt

K = 1

A1 = 20; B1 = 44; C1 = 128

A2 = 22; B2 = 68; C2 = 22

A3 = 7843; B3 = -5; C3 = 3

A4 = 20; B4 = -64; C4 = 20

W1 = tf([K], [A1, B1, C1])

W2 = tf([K], [A2, B2, C2])

W3 = tf([K], [A3, B3, C3])

W4 = tf([K], [A4, B4, C4])

y1, x1 = step(W1)

y2, x2 = step(W2)

y3, x3 = step(W3)

y4, x4 = step(W4)

fig, axs = plt.subplots(2, 2)

axs[0, 0].plot(x1, y1)

axs[0, 0].set\_title('Устойчива с колебаниями')

axs[0, 1].plot(x2, y2, 'tab:orange')

axs[0, 1].set\_title('Устойчива без колебаний')

axs[1, 0].plot(x3, y3, 'tab:green')

axs[1, 0].set\_title('Неустойчива с колебаниями')

axs[1, 1].plot(x4, y4, 'tab:red')

axs[1, 1].set\_title('Неустойчива без колебаний')

for ax in axs.flat:

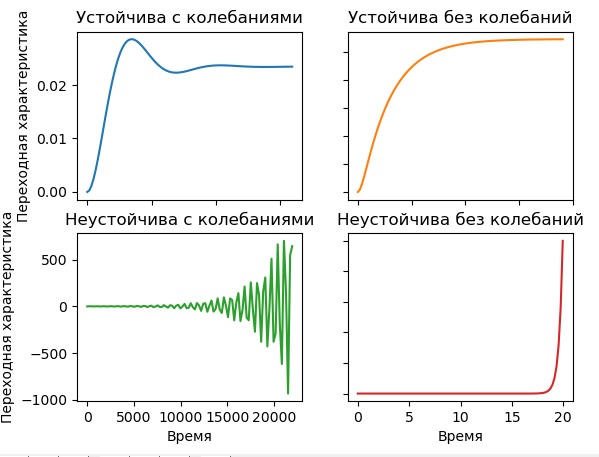
    ax.set(xlabel='Время', ylabel='Переходная характеристика')

for ax in axs.flat:

    ax.label\_outer()

plt.grid()

plt.show()



1. Определить, будет ли система (модель п.4) устойчива по корням характеристического уравнения. Доказать полученные выводы графически.

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)

y, x = step(W7)

p = pole(W7)

for i in p:

if(i == 0):

print("Система нейтральна")

break

if(isinstance(i, complex)):

if((i) < 0):

print("Система устойчива с колебаниями")

break

if((i) > 0):

print("Система неустойчива с колебаниями")

break

else:

if(i < 0):

print("Система устойчива без колебаний")

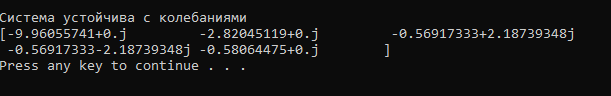
break

if(i > 0):

print("Система неустойчива без колебаний")

break

print(p)



Исходя из полученных корней можно сделать вывод, что система нейтральна, т.к. имеется корень, равный 0, а так же все корни имеют разные знаки (что свидетельствует о неустойчивости системы).

**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.