**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 № 1

по дисциплине «**Математическое моделирование сложных систем**»

на тему: «**Построение и исследование явных аналитических моделей**»

Выполнил: студент гр. ИП-32

Кардаш Е.В.

Принял: доцент

Трохова Т.А.

Гомель 2021

**Цель:** получить навыки компьютерного моделирования технических объектов, представленных в виде явной аналитической модели с выводом результатов моделирования в численном и графическом виде.

**Задание 1:**

***1) Разработать компьютерную модель манипулятора, которая имеет следующие выходные параметры:***

- значения угла поворота звена ОА в зависимости от времени;

- значения координат шарнира А и захвата С в зависимости от времени.

Результаты моделирования представить в численном и графическом виде.

***2) Исследовать модель, для чего определить:***

***-***  максимальное значение координаты Y захвата манипулятора;

- значение координаты X, при котором координата Y захвата манипулятора максимальна;

- значение времени, при котором координата Y захвата манипулятора максимальна.

***Исходными данными для построения модели являются:***

- АВ – длина звена АВ;

- АС – длина звена АС;

- ОА – длина звена ОА;

- вид функции закона движения ползуна, заданный аналитически;

- вид функции закона движения руки АС, заданный аналитически;

- Tk – конечное значение времени для исследования модели манипулятора.

Таблица 1 – Варианты исходных данных

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | AB (см) | ОА (см) | АС (см) | Tk (с) | S0 | Vb | Ψ0 | ω |
| 6 | 96 | 82 | 51 | 1.05 | 1.15 | 0.82 | 0.51 | 2.3 |

###### Описание математической модели

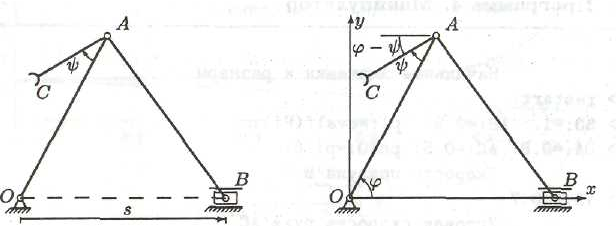


Рисунок 1 – Схема манипулятора

Механизм манипулятора (рисунок 1) приводится в движение двумя независимыми приводами. Задан закон движения ползуна:

S1(t) = S0 - Vb∙t

Закон движения руки АС относительно кривошипа ОА имеет вид:

Ψ(t) = Ψ0 + ω∙t

Координаты шарнира А вычисляются по формулам:

XA = OA∙ cos φ YA = OA∙ sin φ,

где угол φ в зависимости от времени вычисляется по формуле:



Координаты захвата вычисляются по формулам:

XC = XA - AC∙cos(φ – ψ) YC = YA - AC∙sin(φ – ψ)

**Листинг задания 1:**

import numpy as np

import matplotlib.pyplot as plt

def S1(S0, Vb, t):

return S0 - Vb \* t

def psi(psi0, omega, t):

return psi0 + omega \* t

def XA(OA, fi):

return OA \* np.cos(fi)

def YA(OA, fi):

return OA \* np.sin(fi)

def fi(AB, OA, S0, Vb, t):

return np.arccos((-1 \* (AB \*\* 2) + (S1(S0, Vb, t) \*\* 2) + (OA \*\* 2)) / (2 \* OA \* S1(S0, Vb, t)))

def XC(AC, fi, psi, OA):

return XA(OA, fi) - AC \* np.cos(fi - psi)

def YC(AC, fi, psi, OA):

return YA(OA, fi) - AC \* np.sin(fi - psi)

AB = 0.96

OA = 0.82

AC = 0.51

S0 = 1.15

Vb = 0.82

psi0 = 0.51

omega = 2.3

t = np.arange(0, 1.05, 0.001)

X = XC(AC, fi(AB, OA, S0, Vb, t), psi(psi0, omega, t), OA)

Y = YC(AC, fi(AB, OA, S0, Vb, t), psi(psi0, omega, t), OA)

max = np.max(Y)

index = np.where(Y == max)

print('Макс. значение координаты Y захвата манипулятора = ' + str(max))

lol = X[index]

print('Значение координаты X = ' + str(lol))

print('t = ' + str(t[index]))

plt.scatter(lol, max, c = 'red', marker = 'x')

plt.plot(X,Y, c = 'black')

plt.show()

**Результат выполнения задания 1:**

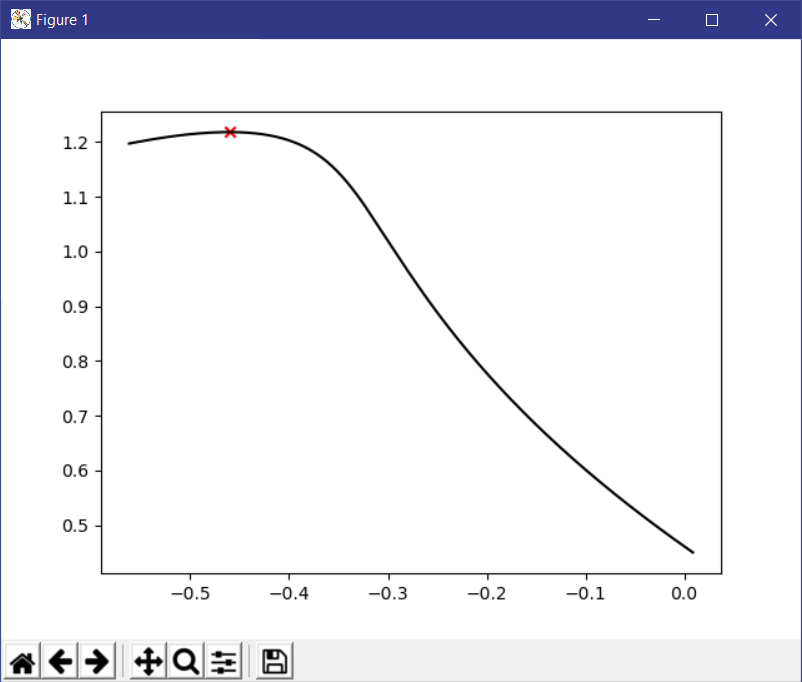
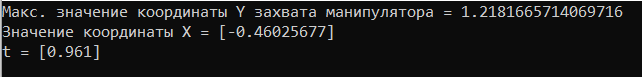


Рисунок 2 – Результат выполнения задания 1

**Задание 2:**

***1) Разработать компьютерную модель кулачкового механизма, которая имеет следующие выходные параметры:***

- функцию аналога ускорения, скорости, перемещения толкателя в зависимости от времени;

***-*** центровой профиль кулачка.

***2) Исследовать модель, для чего определить следующие параметры:***

- максимальные и минимальные значения аналога ускорения кулачкового механизма;

- значение времени, при котором аналог ускорения кулачкового механизма максимален.

Таблица 2 – Таблица исходных данных

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| №  варианта | Параметры закона S | | | | | |
|  |  |  |  |  |  |
| 2 | 1,7 | 3,2 | 5,3 | 8,6 | 5,8 | 1,256 |

Для всех вариантов ***v***=0.675

###### Описание математической модели

Дан кулачковый механизм, закон изменения аналога ускорения толкателя которого приведен на рисунке 3.

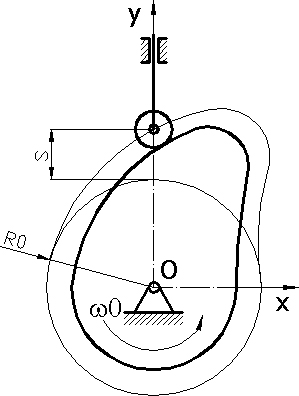


Рисунок 3 – Кулачковый механизм

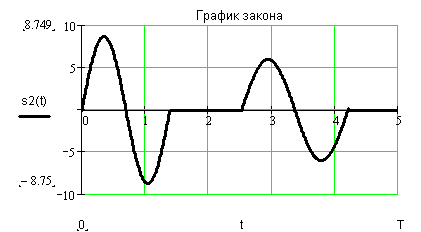


Рисунок 4 – Функция аналога ускорения роликового толкателя

Функция аналога ускорения роликового толкателя имеет вид:





Функция скорости роликового толкателя имеет вид:

S1(t) =



Функция перемещения роликового толкателя имеет вид:

S(t) =



Для вычисления радиуса-вектора центрового профиля кулачкового механизма применяются следующая зависимость:

R=R0 + S(t)

Здесь R0 вычисляется как минимальное значение функции R1(t) закона изменения радиуса-вектора центрового профиля, удовлетворяющего допустимому значению угла давления.



Центровой профиль кулачка определяется системой параметрических уравнений в декартовой системе координат:

*X(t)=R(t)sin(w0⋅t)*

*Y(t)=R(t)cos(w0⋅t)*

**Листинг задания 2:**

import numpy as np

import matplotlib.pyplot as plt

import scipy.integrate as integrate

import scipy.special as special

def X(i, t):

return R[i] \* np.cos(t \* omega)

def Y(i, t):

return R[i] \* np.sin(t \* omega)

def S11(t):

if 0 <= t < t1:

return a1 \* np.sin(b1 \* omega \* t)

elif ((t1 <= t < t2) or (t >= t3)):

return 0

elif t2 <= t < t3:

return a2 \* np.sin(b2 \* omega \* t)

def S1(t, a):

tempArray = []

for temp in t:

tempArray.append(omega \* omega \* temp)

return np.trapz(tempArray, a)

def S(t, a):

return np.trapz(t, a)

def R1():

min = s1[0]/np.tan(V)

for temp in s1:

if temp/np.tan(V) < min:

min = temp/np.tan(V)

return min

fi1 = 1.7

fi2 = 3.2

fi3 = 5.3

a1 = 8.6

a2 = 5.8

omega = 1.256

V = 0.675

b1 = 2 \* np.pi / fi1

b2 = 2 \* np.pi / (fi3 - fi2)

t1 = 2 \* np.pi/(b1 \* omega)

t2 = fi2 / omega

t3 = t2 + 2 \* np.pi / (b2 \* omega)

delta = 0.01

t = np.arange(0, 2 \* np.pi / omega, delta)

s = []

s1 = []

s11 = []

tempArray = []

for temp in t:

tempArray.append(temp)

s11.append(S11(temp))

s1.append(S1(s11, tempArray))

s.append(S(s1, tempArray))

plt.figure(1)

plt.plot(t, s11, c='Teal')

plt.grid()

plt.figure(2)

plt.plot(t, s1, c='Green')

plt.grid()

plt.figure(3)

plt.plot(t, s, c='Red')

plt.grid()

R0 = R1()

R = []

x = []

y = []

i = 0

for temp in s:

R.append(R0 + temp)

for temp in t:

x.append(X(i, temp))

y.append(Y(i, temp))

i+=1

plt.figure(4)

plt.plot(y, x, c='red')

plt.grid()

plt.show()

**Результат выполнения задания 2:**

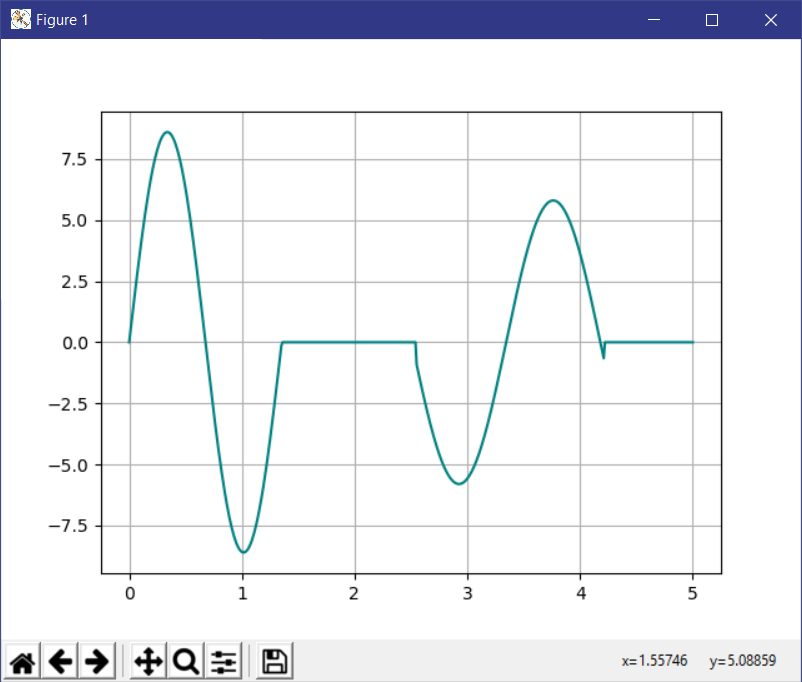


Рисунок 5 – Результат выполнения задания 2

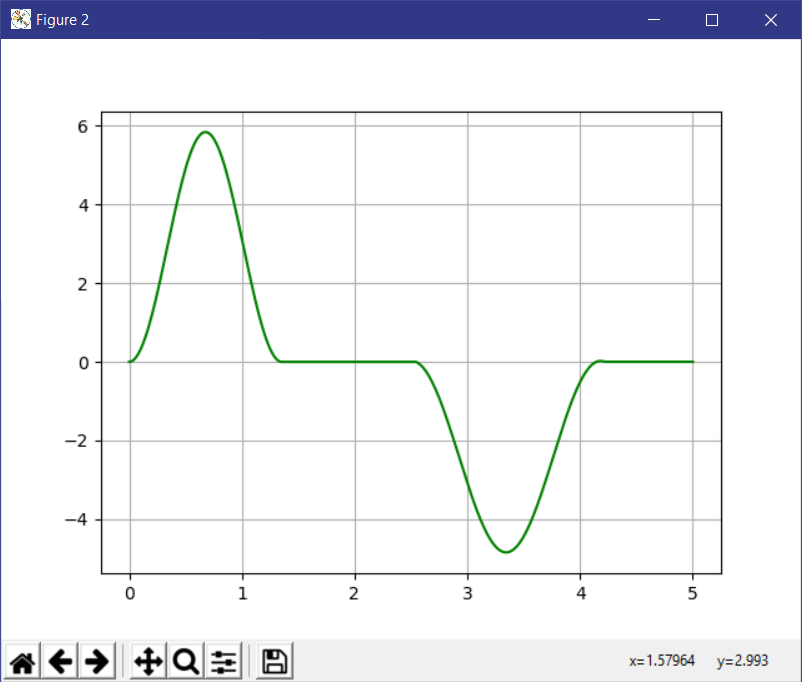


Рисунок 6 – Результат выполнения задания 2

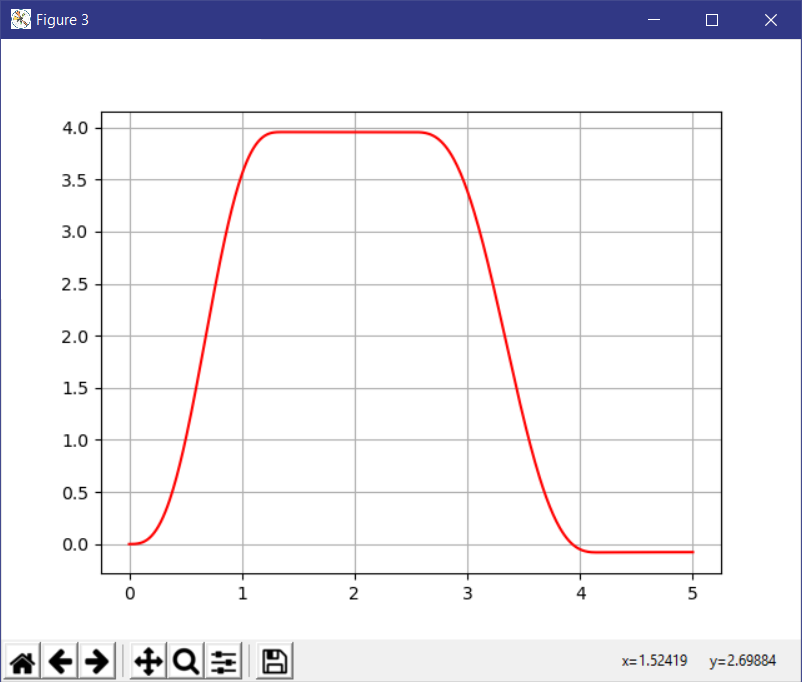


Рисунок 7 – Результат выполнения задания 2

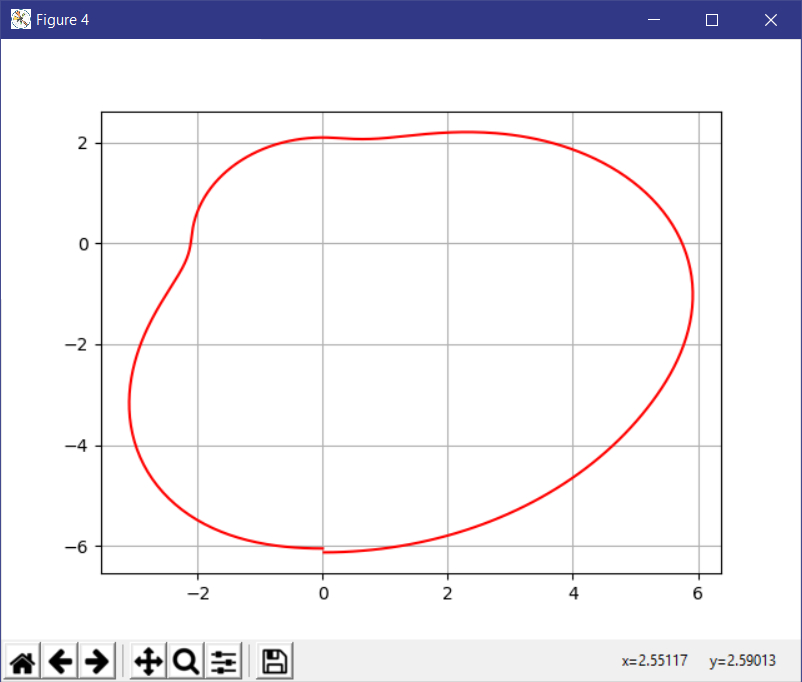


Рисунок 8 – Результат выполнения задания 2

**Вывод:** в ходе данной лабораторной работы были получены навыки компьютерного моделирования технических объектов, представленных в виде явной аналитической модели с выводом результатов моделирования в численном и графическом виде.