

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего
образования

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И
ОПТИКИ

Факультет систем управления и робототехники

**Отчет по лабораторной работе №4 «Робот с
дифференциальным приводом»
по дисциплине «Введение в профессиональную
деятельность»**

Выполнили: студенты гр. **R3137**

Кирбаба Д.Д.

Курчавый В.В.

Нестеров И.А.

Петров И.А.

Преподаватель: Перегудин А. А.,
ассистент фак. СУиР

Санкт-Петербург, 2021

1. Цель работы

Получить опыт построения математической модели робота, освоить алгоритм движения робота с дифференциальным приводом к заданной точке.

2. Материалы работы

2.1 Программа для EV-3 на языке программирования Python

```
#!/usr/bin/env python3
from ev3dev.ev3 import *
from math import cos, sin, pi, atan2

def run(goal, cordrob, tetta):
    motorR = LargeMotor('outA')
    motorL = LargeMotor('outB')

    # координаты цели
    # test №1: goal[3,0]
    # test №2: goal[0,3]
    # test №3: goal[3,3]
    # test №4: goal[-3,0]
    # test №5: goal[0,-3]
    # test №6: goal[-3,3]
    # test №7: goal[3,-3]
    # test №8: goal[-3,-3]
    # Если прога проходит все тесты, то в записи в файл удалить вывод угла и взять измерения

    r = 0.03 # радиус колеса в метрах
    b = 0.155 # расстояние между центрами колес

    motorR.position = 0
    motorL.position = 0

    angleR, angleL = 0,0

    Umax = 7 # максимальное напряжение, может быть другим

    data = open("test1square" + str(i) + ".txt", "w+") # название менять на номер теста

    Ks = 7 # коэффициент для движения вперед, возможно побольше, но точно меньше, чем для поворота
    Kr = 85 # коэффициент для поворота, возможно побольше стоит сделать, но не сильно много

    if (goal[1]!=cordrob[1] and goal[0]!=cordrob[0]):
        asimut = atan2((goal[1]-cordrob[1]), (goal[0]-cordrob[0]))
```

```

if (goal[0] == cordrob[0]) and (goal[1] > cordrob[1]):
    asimut = pi/2
elif (goal[0] == cordrob[0]) and (goal[1] < cordrob[1]):
    asimut = -pi/2

if (goal[1] == cordrob[1]) and (goal[0] > cordrob[0]):
    asimut = 0
elif (goal[1] == cordrob[1]) and (goal[0] < cordrob[0]):
    asimut = pi

p = ((goal[0] - cordrob[0])**2 + (goal[1] - cordrob[1])**2)**0.5 # начальное расстояние

while (p > 0.3):

    xprev = cordrob[0]
    yprev = cordrob[1]

    tettarev = tetta

    prevangleR = angleR
    prevangleL = angleL

    angleR = motorR.position * pi/180
    angleL = motorL.position * pi/180

    difangleR = angleR - prevangleR
    difangleL = angleL - prevangleL

    srangle = (difangleR+difangleL)/2

    tetta = tettarev + (difangleR - difangleL)*r/b # считается угол на который повернул р
    обот/ возможно, нужно раньше считать, чем координаты
    cordrob[0] = xprev + cos(tetta)*srangle*r # считается x
    cordrob[1] = yprev + sin(tetta)*srangle*r # считается y

    p = ((goal[0] - cordrob[0])**2 + (goal[1] - cordrob[1])**2)**0.5 # считается расстояни
    е от робота до цели

    a = asimut - tetta # считается угол между желаемым положением робота и текущим положен
    ием

    Us = Ks * p # расчет напряжения для движения вперед с помощью пропорционального регуля
    тора

    Ur = Kr * a # расчет напряжения для поворота с помощью П регулятора

    # Блок для правого колеса

    if abs(Us + Ur)*100/Umax <= 100: # <= Umax

```

```

        UR = (Us + Ur)*100/Umax

    elif (Us + Ur) > 0:
        UR = 100
    else:
        UR = -100

    # блок для левого колеса

    if abs(Us - Ur)*100/Umax <= 100: # <= Umax

        UL = (Us-Ur)*100/Umax

    elif (Us - Ur) > 0:
        UL = 100
    else:
        UL = -100

    # далее подаем напряжение

    motorR.run_direct(duty_cycle_sp = (UR))
    motorL.run_direct(duty_cycle_sp = (UL))

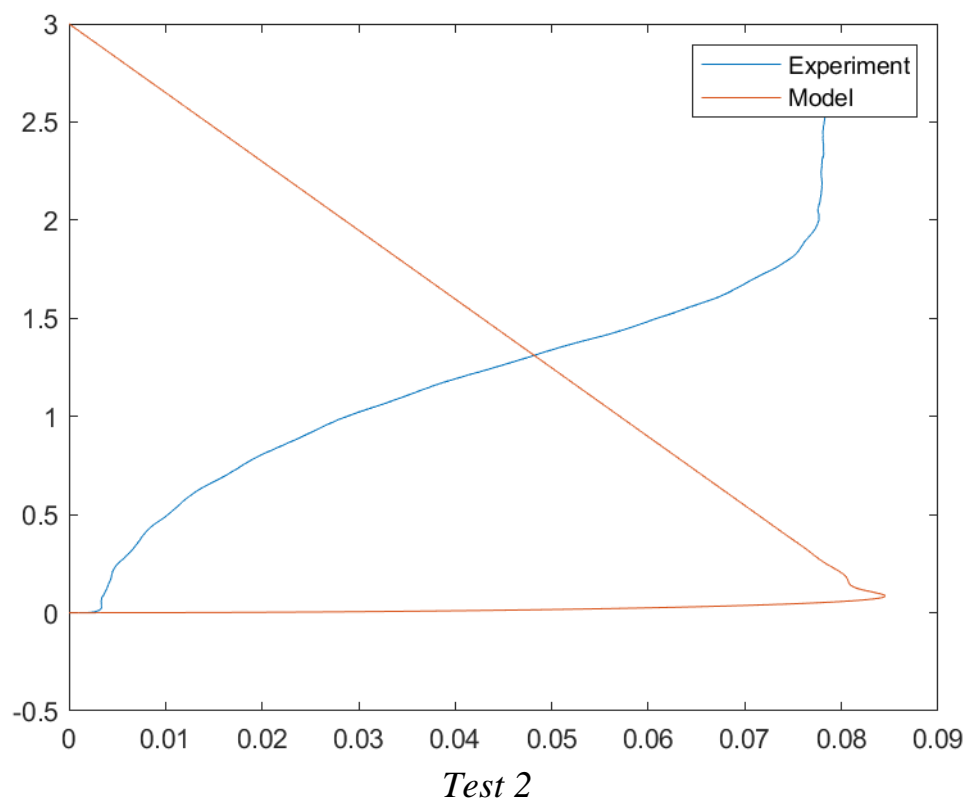
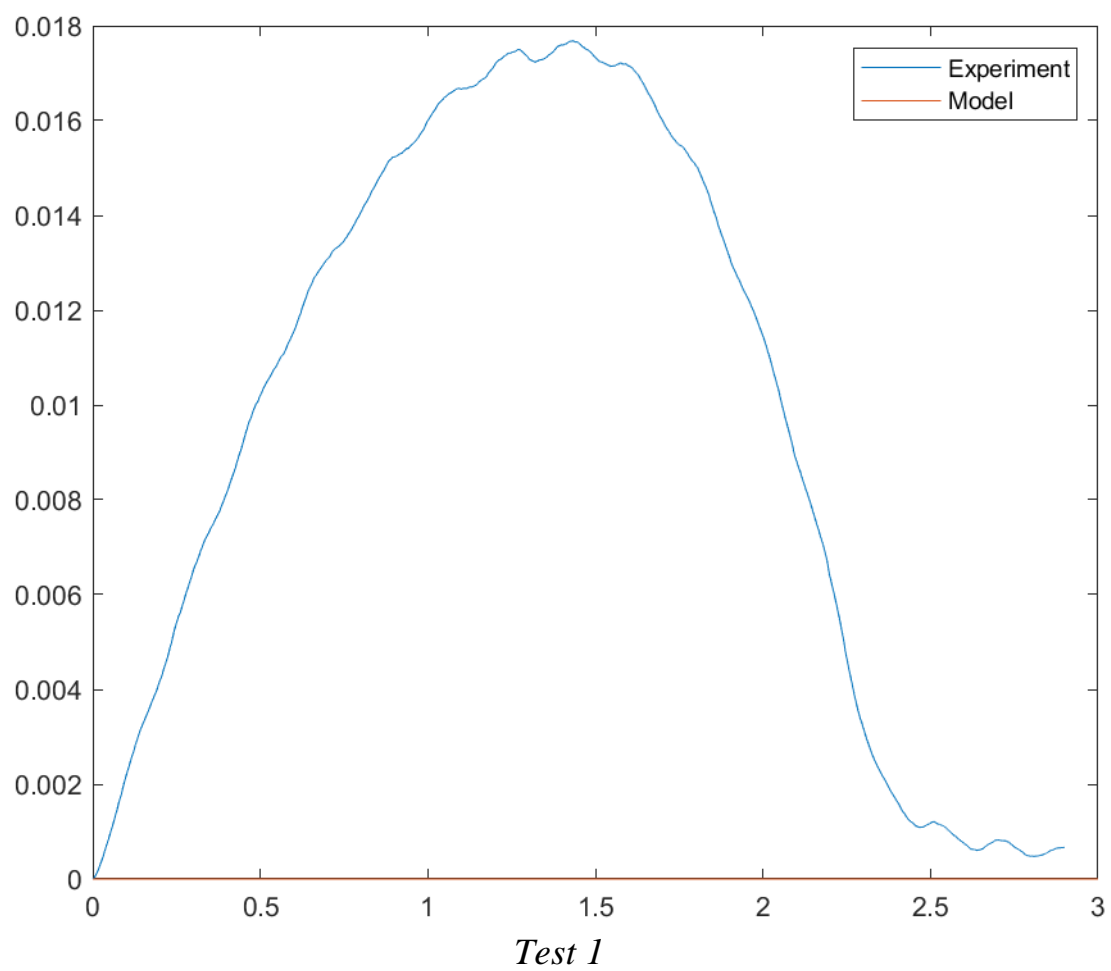
    # запись угла, на который надо повернуться и координат, здесь слезует удалить а, когда
    пройдет все тесты
    data.write(str(cordrob[0]) + '\t' + str(cordrob[1]) + '\n')

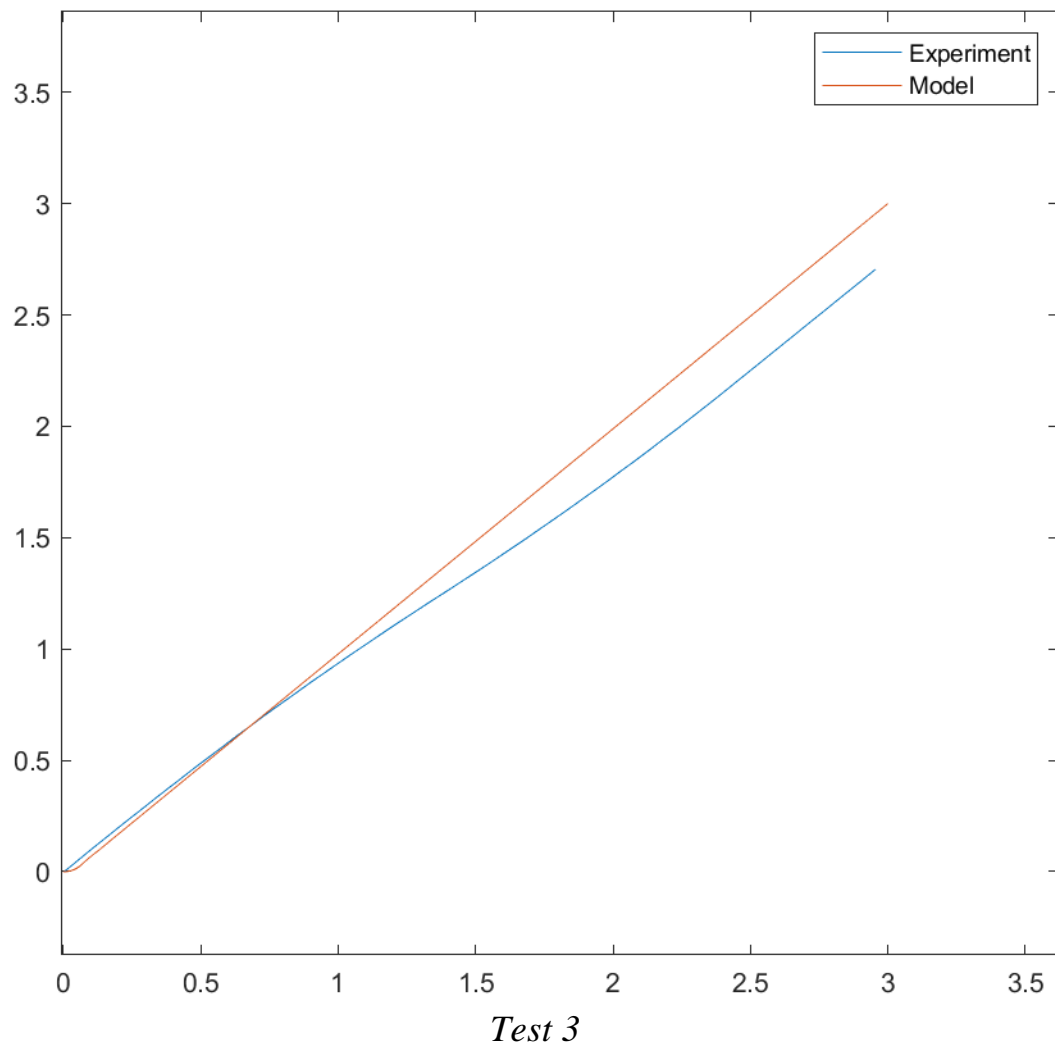
    data.close()
    motorR.stop(stop_action = 'brake')
    motorL.stop(stop_action = 'brake')

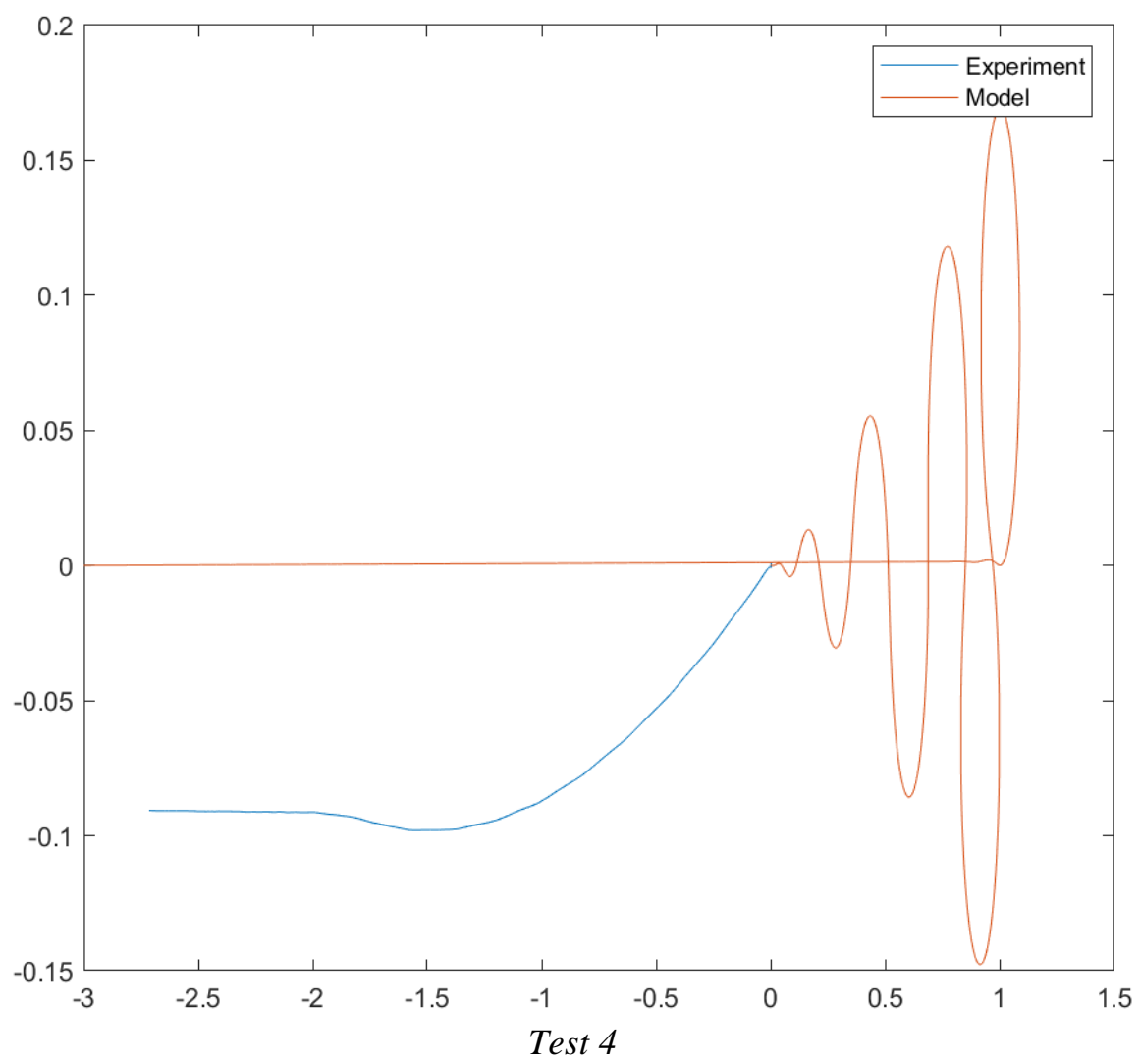
    return cordrob, tetta
i = 3
cordrob, tetta = run((i,i), [0,0], 0)
cordrob, tetta = run((-i,i), cordrob, tetta)
cordrob, tetta = run((-i,-i), cordrob, tetta)
cordrob, tetta = run((i,-i), cordrob, tetta)

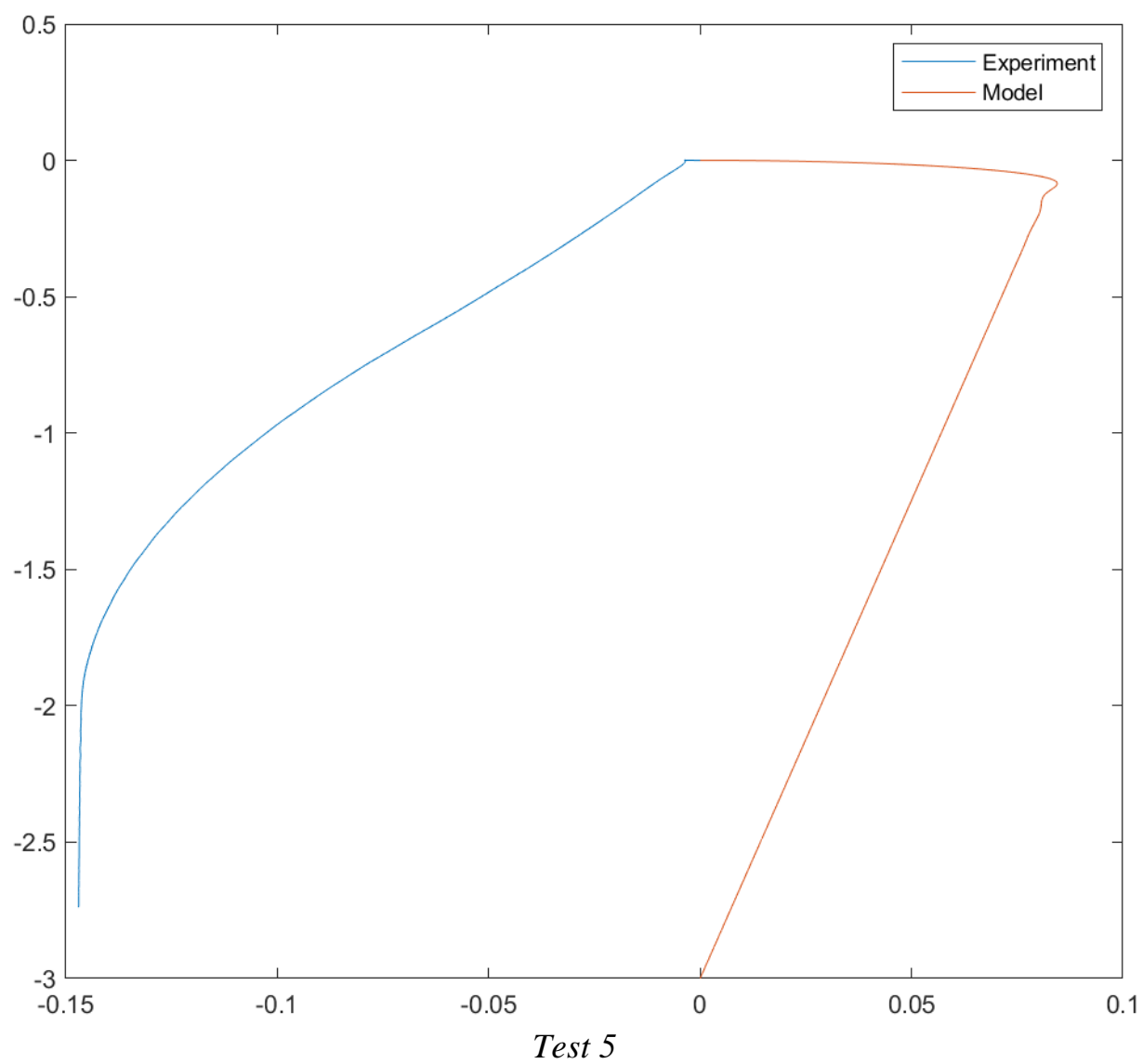
```

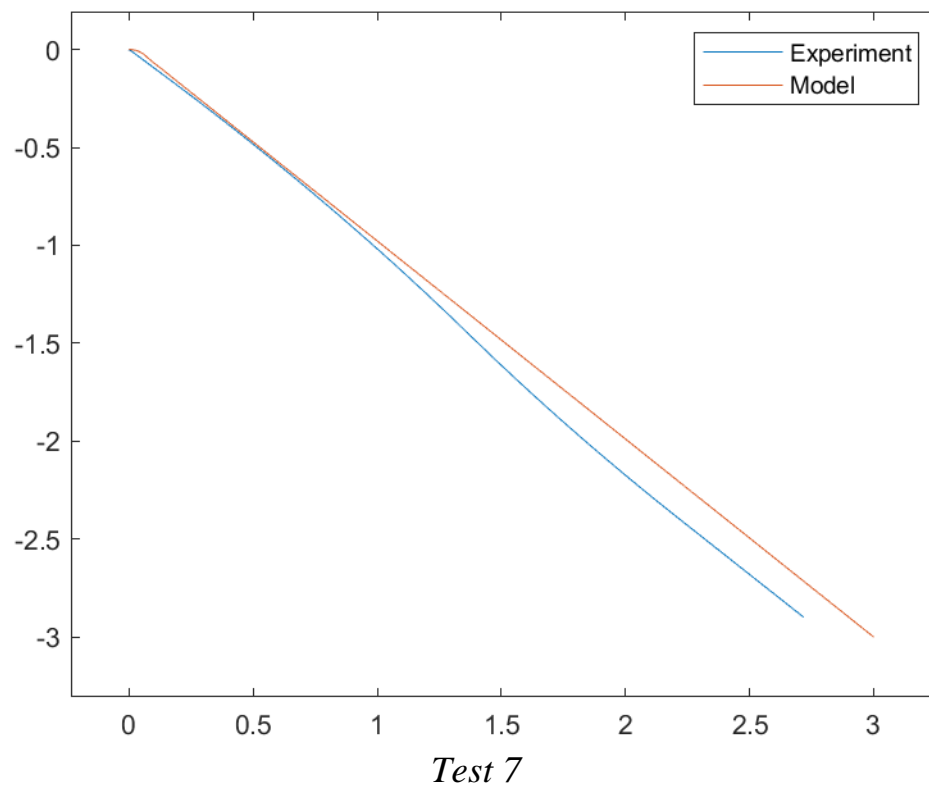
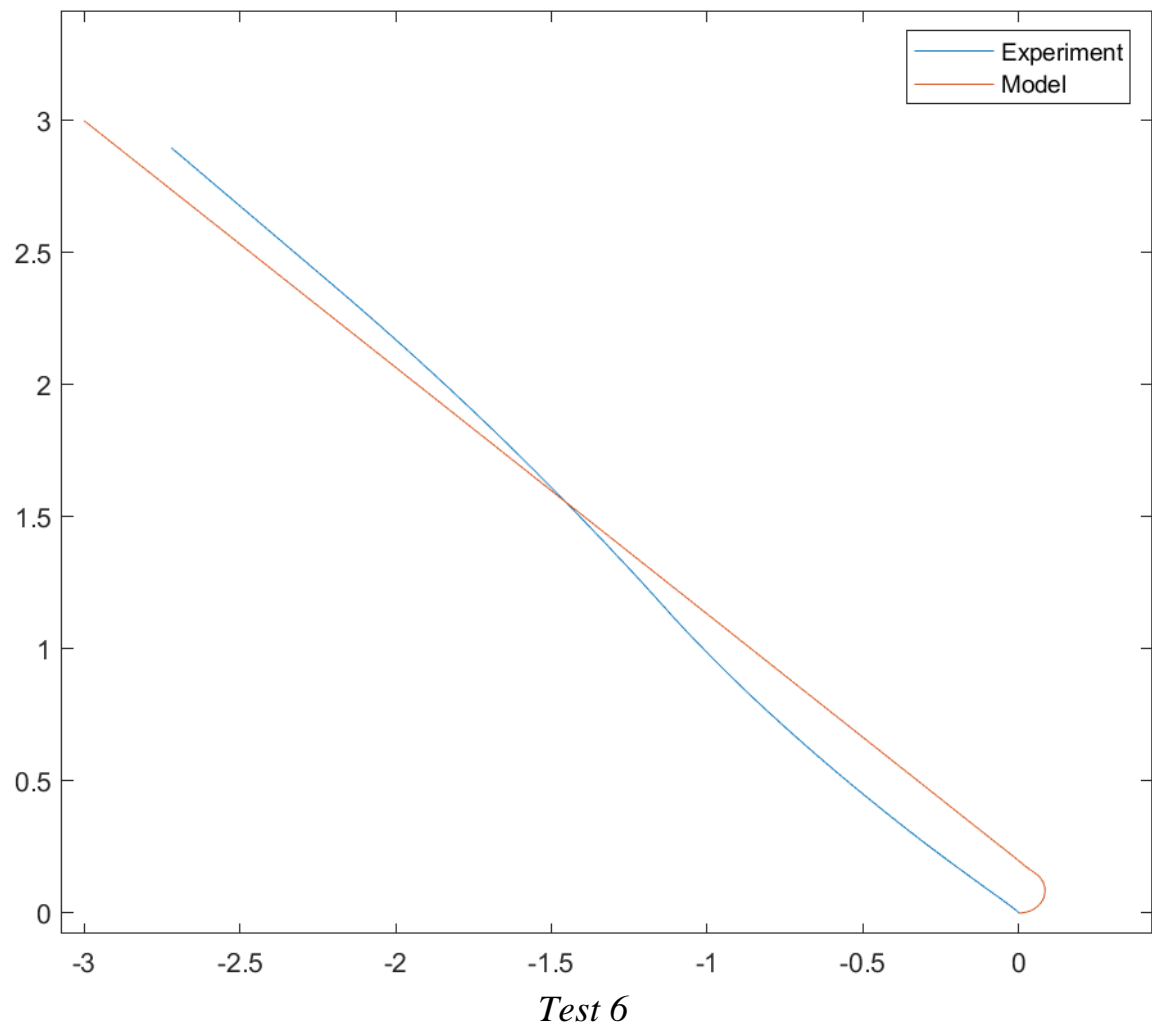
2.2. Графики перемещения робота в указанную в тестах к программе точку.

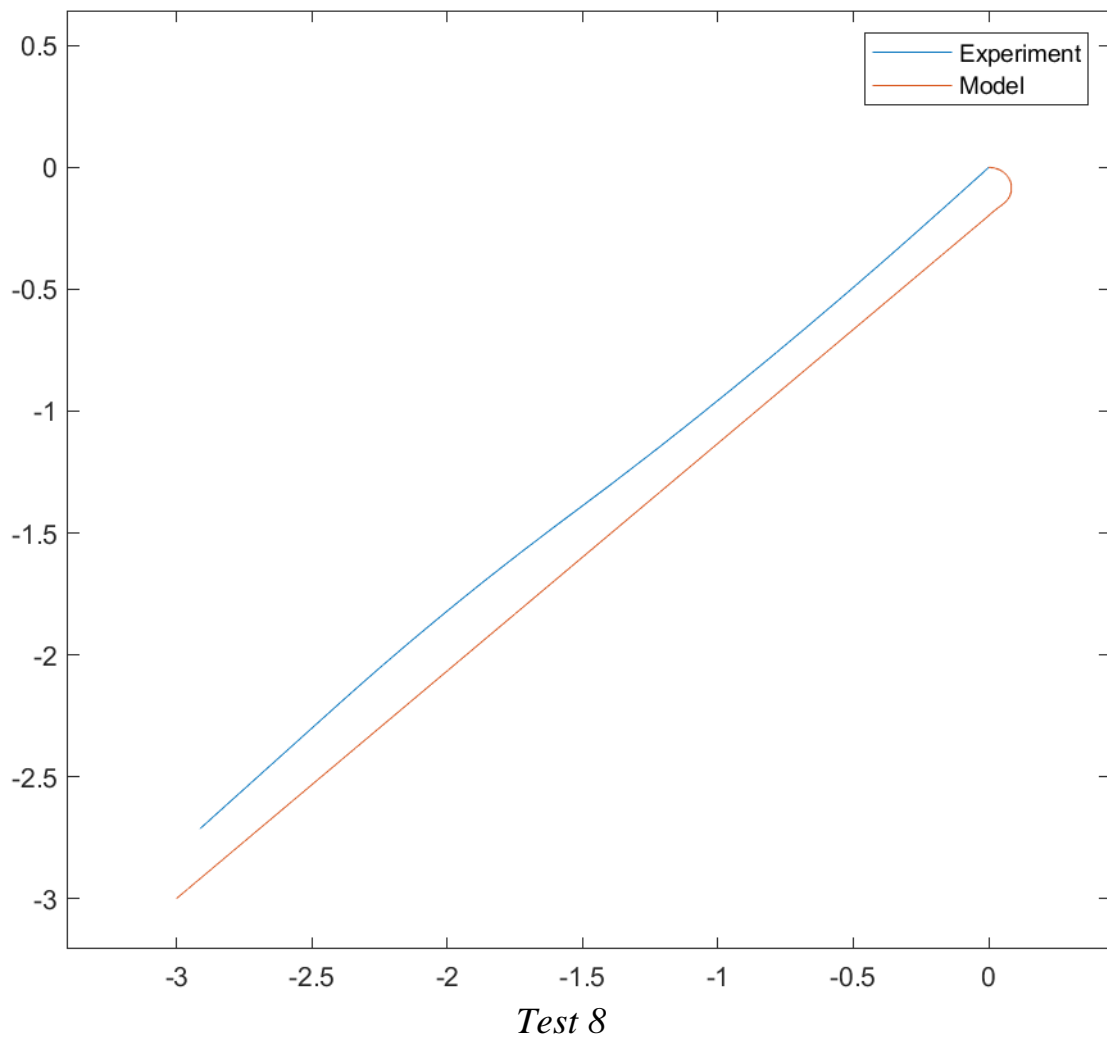




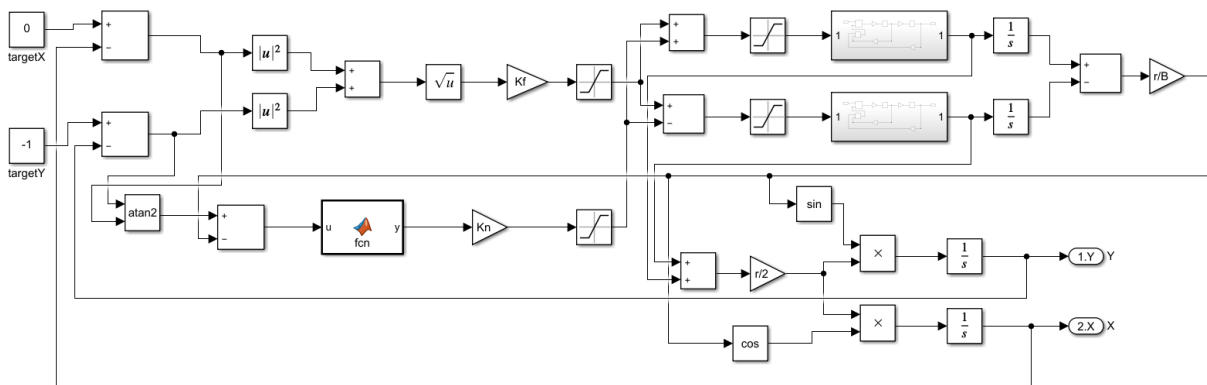








2.3.1 Полная схема моделирования:



2.3.2 Код Matlab для создания графиков

```

Res1 = dlmread('data 8.txt');
Res1 = Res1';
Y1 = Res1(3,:);
X1 = Res1(2,:);
plot(Y1, X1);
hold on

```

```

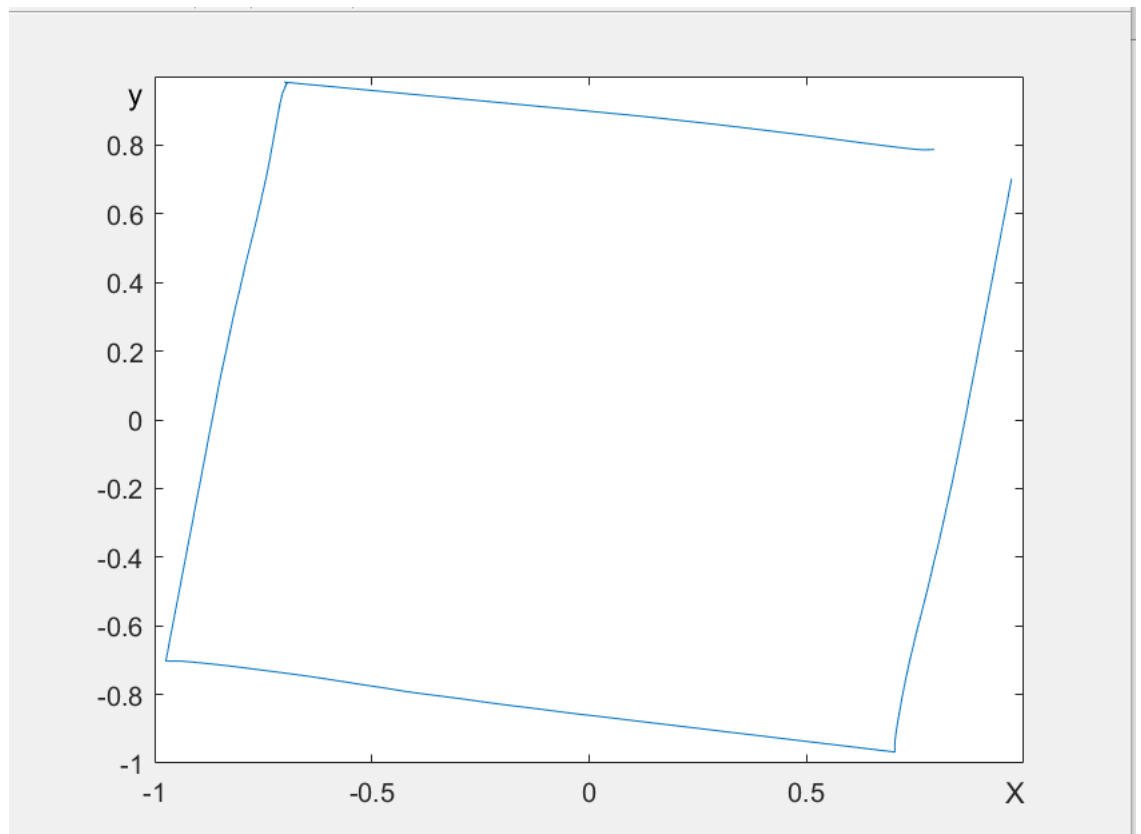
Y = out.yout{1}.Values.Data;
X = out.yout{2}.Values.Data;
hold on
plot(X, Y);
hold on

ke = 0.48756937411725393;
km = ke;
J = 0.002437632;
L = 0.0047;
Kn = 0.95;
kD = 10;
R = 8.183683911882799;
U = 14.644499694385809*0.48756937411725393;
Kn = Kn*U*180/100/pi;

xlabel('Ox')
ylabel('Oy')
legend('Experiment', 'Model')

```

2.3.3 График непрерывного движения робота по квадрату



3. Выводы

В результате проделанной работы нами был создан робот с дифференциальным приводом. Написанная на языке программирования Python программа подает на ev3 алгоритм, позволяющий роботу перемещаться в точку с заранее указанными координатами. Кроме того, П-регулятор позволил нам реализовать непрерывное движение робота по квадрату. Движение робота было смоделировано при помощи языка Matlab, а графическая интерпретация этого движения в виде графиков – в Simulink.