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 №1

«Движение робота к точке с заданными координатами»

по дисциплине «Введение в профессиональную деятельность»

Выполнили: студенты гр. R3135
Дупак А. А.,
Щтенников Р. А.,
Зорькина А. А.

Преподаватель: Перегудин А. А.

Санкт-Петербург
2018

Цель работы

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

Материалы работы

Результаты необходимых расчетов и построений

На рис. 1 — рис. 4: красный график постоен на основе результататов моделирования, синий по экспериментальным данным.

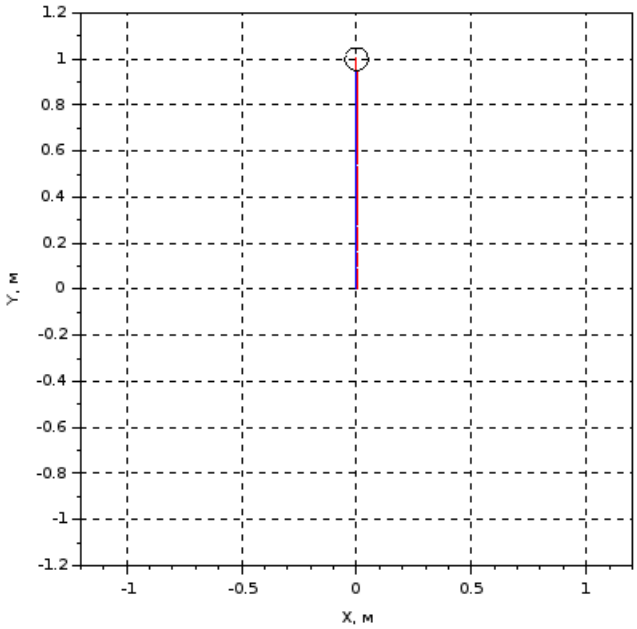

Рис. 1. Графики зависимостей координат Y от координат X, при desired_x=0, desired_y=1.

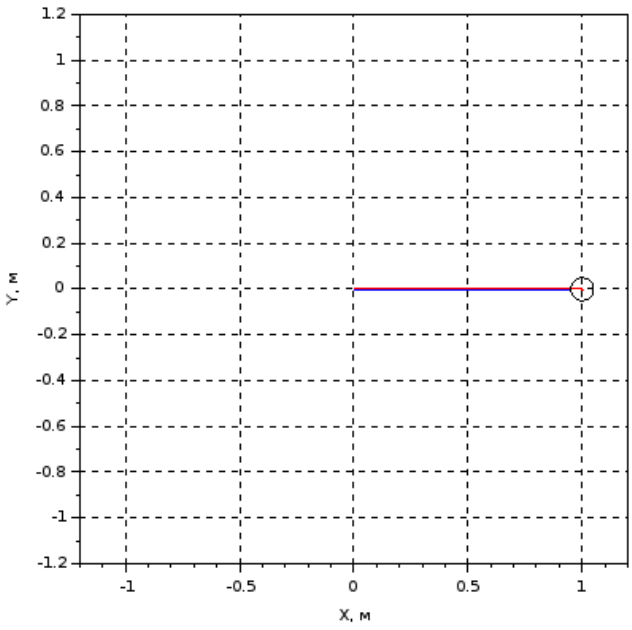

Рис. 2. Графики зависимостей координат Y от координат X, при desired_x=1, desired_y=0.

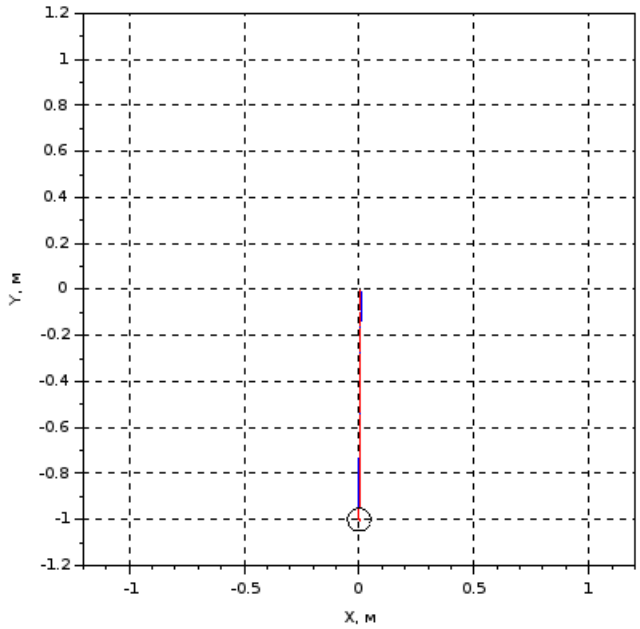

Рис. 3. Графики зависимостей координат Y от координат X, при desired_x=0, desired_y=-1.

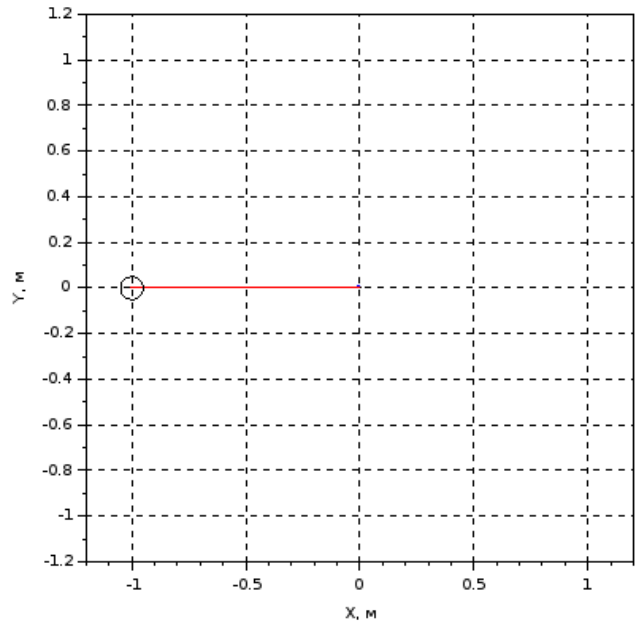

Рис. 4. Графики зависимостей координат Y от координат X, при desired_x=-1, desired_y=0.

На рис. 5 красный график постоен на основе результататов моделирования схемы №1, зеленый график постоен на основе результататов моделирования схемы №2, синий по экспериментальным данным.

На рис. 6 синим цветом изображен график построенный по экспериментальным данным, а черным - объект используемый в качестве препятствия.

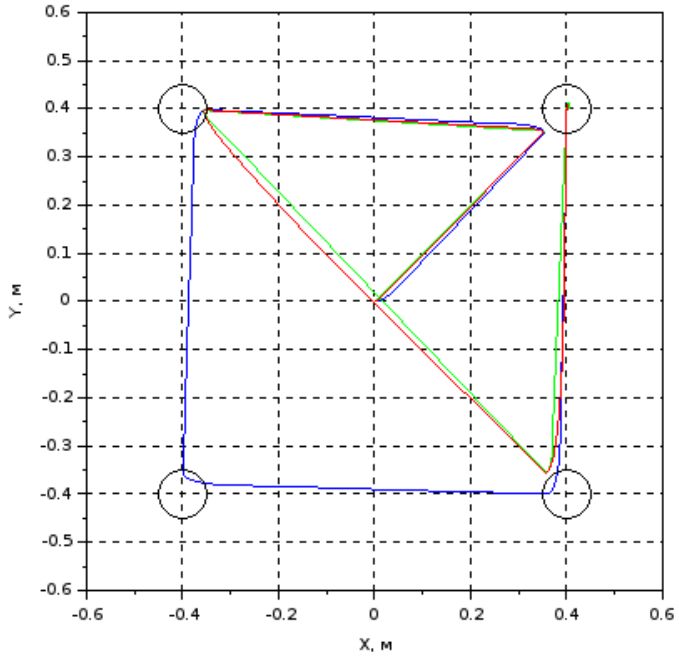

Рис. 5. Графики зависимости координат Y от координат X, при решении задачи движения робота через координаты задающие квадрат, route=[[0.4, 0.4], [-0.4, 0.4], [-0.4, -0.4], [0.4, -0.4], [0.4, 0.4]].

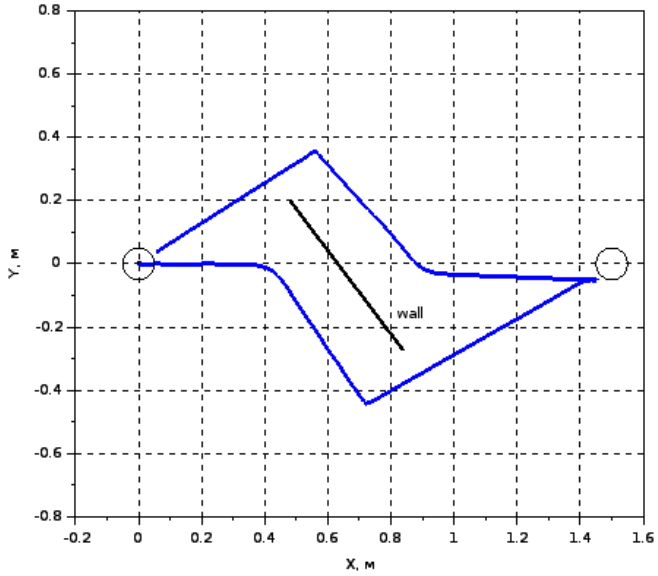

Рис. 6. Графики зависимостей координат Y от координат X, при движении робота к точке с координатами (0, 1.5) с обходом препятствий (см. видео)

Код программы для EV3

```
#!/usr/bin/env python3
from ev3dev.ev3 import *
import math

k1 = 8
k2 = 0.5
k3 = 0.06
route = [[1.5, 0], [0, 0]]
voltage = 7.00
```

```
r = 0.02
B = 0.12
ok_zone = 0.05
min_dist = 35

mA = LargeMotor('outA')
mB = LargeMotor('outB')
us1 = UltrasonicSensor('in1')
us2 = UltrasonicSensor('in2')
us1.mode = 'US-DIST-CM'
us2.mode = 'US-DIST-CM'
gyro = GyroSensor('in3')
gyro.mode = 'GYRO-RATE'
fh = open('robot_data.txt', 'w')

desired_x = route[0][0]
desired_y = route[0][1]
current_x = 0
current_y = 0
complete = 0
mA.position = 0
mB.position = 0
prev_path = 0
integral = 0
gyro_angle = 0

try:
    gvalues = 0
    for i in range(100):
        gvalues += gyro.value()
        time.sleep(0.01)
    gyro_offset = gvalues/100
    current_time = time.time()
    while complete < len(route):
        dt = time.time() - current_time
        current_time = time.time()
        motorA_pos = mA.position * math.pi / 180
        motorB_pos = mB.position * math.pi / 180
        dist1 = us1.value() / 10
        dist2 = us2.value() / 10
        gyro_raw = gyro.value()
        gyro_speed = (gyro_raw - gyro_offset) * math.pi / 180
        gyro_angle += gyro_speed * dt
        path = (motorA_pos + motorB_pos)*(r/2)
        dpath = path - prev_path
        prev_path = path
        current_x += dpath*math.cos(gyro_angle)
        current_y += dpath*math.sin(gyro_angle)
        dx = desired_x - current_x
        dy = desired_y - current_y
        path_err = math.sqrt(dx**2 + dy**2)
        need_angle = math.atan2(dy, dx)
        cur_dist = min(dist1, dist2)
```

```
        if(cur_dist < min_dist):
            dist_err = k3*(min_dist - cur_dist)*math.copysign(1,
dist1 - dist2)
        else:
            dist_err = 0
        angle_err = need_angle - gyro_angle - dist_err
        if abs(angle_err) > math.pi:
            angle_err -= math.copysign(1, angle_err)*2*math.pi
        integral += path_err*dt
        u_straight = voltage*math.tanh
(path_err)*math.cos(angle_err) + k2*integral
        u_rotation = k1*angle_err +
math.sin(angle_err)*u_straight/path_err
        sA = u_straight + u_rotation
        sB = u_straight - u_rotation
        sA = sA * 100 / voltage
        sB = sB * 100 / voltage
        if abs(sA) > 100:
            sA = math.copysign(100, sA)
        if abs(sB) > 100:
            sB = math.copysign(100, sB)
        mA.run_direct(duty_cycle_sp=sA)
        mB.run_direct(duty_cycle_sp=sB)
        fh.write(str(current_x) + ' ' + str(current_y) + '\n \n')
        if (abs(dx) < ok_zone) and (abs(dy) < ok_zone):
            complete += 1
            if complete < len(route):
                desired_x = route[complete][0]
                desired_y = route[complete][1]
finally:
    mA.stop(stop_action='brake')
    mB.stop(stop_action='brake')
    fh.close
```

Схема моделирования

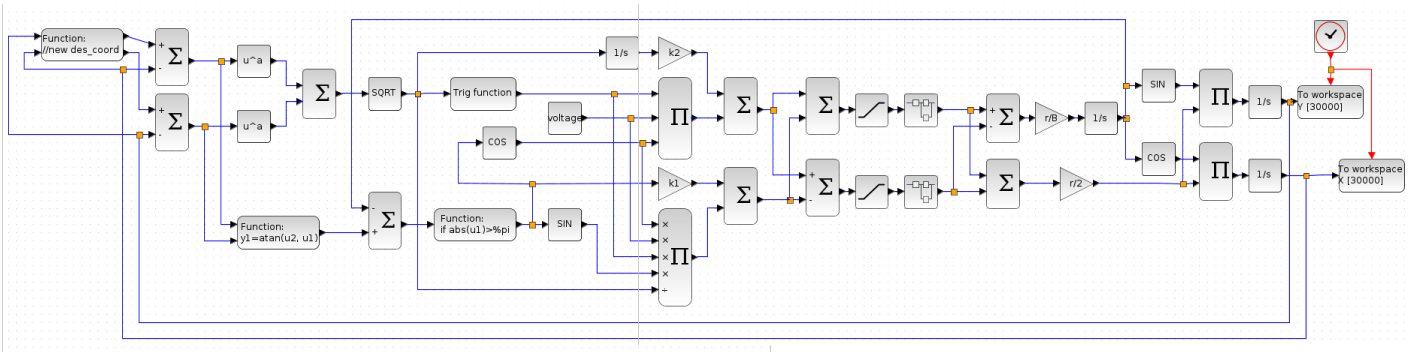

Рис. 7. Схема моделирования исследуемого процесса (№1).

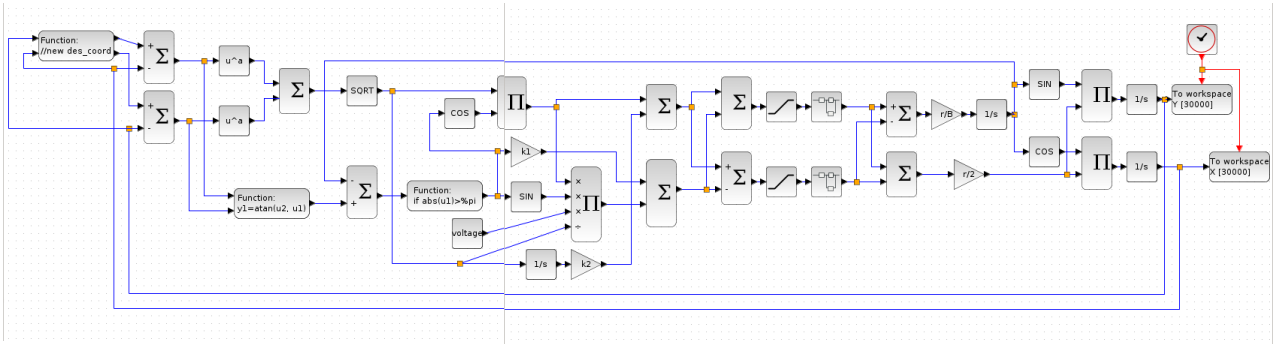

Рис. 8. Схема моделирования исследуемого процесса (№2).

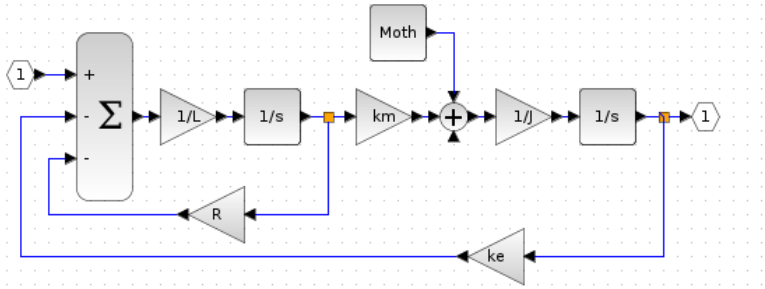

Рис. 9. Схема SuperBlock.

Код блока Function1

```
if abs(u1)>%pi
    y1=u1-sign(u1)*2*%pi
else
    y1=u1
end
```

Код блока Function2

```
//new des_coord
global complete
x_des=array(complete, 1)
y_des=array(complete, 2)
dx=x_des-u1
dy=y_des-u2
if (abs(dx) < 0.05) and (abs(dy) < 0.05)
    if complete < 5
        complete=complete+1
    end
end
y1=x_des
y2=y_des
```

Код программы Scilab

```
ke = 0.5
km = 0.5
R = 6.1
J = 0.0025
L = 0.0047

Moth = 0
r = 0.02 //радиус колеса
B = 0.12 //расстояние между колесами

k1 = 8
```

```
k2 = 0.5
k3 = 0.06
zone = 0.05
voltage = 7.00

global complete
complete = 1
route = [0.4, 0.4; -0.4, 0.4; -0.4, -0.4; 0.4, -0.4; 0.4, 0.4]
importXcosDiagram("/home/aleksandr/ITMO_lab2/ev3/scilab/model2.zcos");
xcos_simulate(scs_m,4);
res = read("/home/aleksandr/ITMO_lab2/ev3/data/robot_data.txt", -1, 4)
x = res(:, 1)
y = res(:, 2)
xtitle('', 'X, м', 'Y, м')
xgrid(0)
plot2d(0,0,0,'031',' ',[-0.5,-0.5,0.5,0.5]);
plot2d(x, y, 2)
plot2d(X.values, Y.values, 5)
```

Выводы

В результате проделанной работы была решена задача локальной навигации мобильного робота с дифференциальным приводом, а также реализовано движение робота через точки с заданными координатами.

Кроме того, была построена схема моделирования исследуемого процесса в среде Xcos. Сравнив график экспериментальных данных с графиком построенным на основании результатов моделирования схемы, мы удостоверились в том, что моделирование исследуемого процесса дает результаты схожие со значениями полученными в ходе прямых измерений.