МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра «Вычислительные системы и технологии»

Отчет по лабораторной работе №3

по дисциплине

«Аппаратное и программное обеспечение робототехнических систем»

РУКОВОДИТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Гай В.Е

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Соляник Д.Р.

ГРУППА:

19-В-1

Работа защищена «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_

С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород

2022г.

Лабораторная работа № 3. Программирование алгоритмов управления роботом в Webots

**Цель:** получение навыков работы с алгоритмами управления роботами Webots

**Вариант №5**

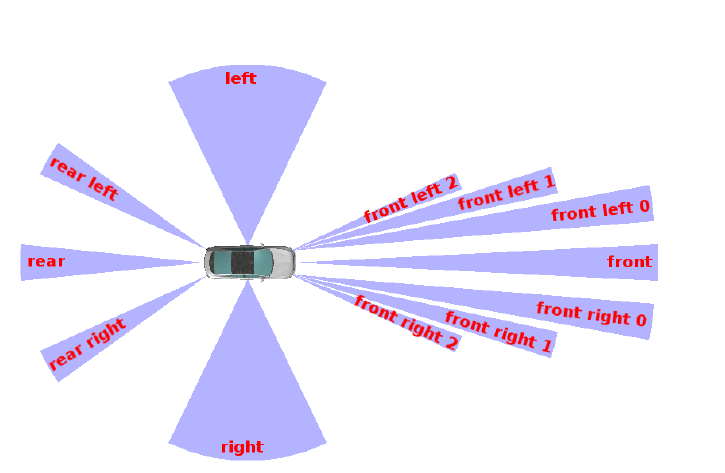
**Задание:**

Контроллер автомобиля должен позволять ехать автомобилю в правой полосе, не заезжая за сплошную полосу и не сталкиваясь с другими автомобилями.

**Ход работы:**

Обрабатываем данные с сенсоров, установленных на машине. На их основе будем управлять движением машины.

Для ого чтобы машина не съезжала с крайней правой полосы используем константу. По результатам сравнения константы с показателями сенсоров будет принято решение об изменении траектории движения.



Изменил контроллер:

"""Sample Webots controller for highway driving benchmark."""

from vehicle import Driver

# name of the available distance sensors

sensorsNames = [

'front',

'front right 0',

'front right 1',

'front right 2',

'front left 0',

'front left 1',

'front left 2',

'rear',

'rear left',

'rear right',

'right',

'left']

sensors = {}

maxSpeed = 20

driver = Driver()

driver.setSteeringAngle(0.0) # go straight

# get and enable the distance sensors

for name in sensorsNames:

sensors[name] = driver.getDistanceSensor('distance sensor ' + name)

sensors[name].enable(10)

# get and enable the GPS

gps = driver.getGPS('gps')

gps.enable(10)

# get the camera

camera = driver.getCamera('camera')

camera.enable(50)

camera.recognitionEnable(50)

while driver.step() != -1:

# adjust the speed according to the value returned by the front distance sensor

frontDistance = sensors['front'].getValue()

frontRange = sensors['front'].getMaxValue()

leftDistance = sensors['left'].getValue()

leftRange = sensors['left'].getMaxValue()

rightDistance = sensors['right'].getValue()

rightRange = sensors['right'].getMaxValue()

fleft2Distance = sensors['front left 2'].getValue()

fleft2Range = sensors['front left 2'].getMaxValue()

fright0Distance = sensors['front right 0'].getValue()

fright0Range = sensors['front right 0'].getMaxValue()

fright1Distance = sensors['front right 1'].getValue()

fright1Range = sensors['front right 1'].getMaxValue()

fright2Distance = sensors['front right 2'].getValue()

fright2Range = sensors['front right 2'].getMaxValue()

fleft0Distance = sensors['front left 0'].getValue()

fleft0Range = sensors['front left 0'].getMaxValue()

fleft1Distance = sensors['front left 1'].getValue()

fleft1Range = sensors['front left 1'].getMaxValue()

r = rightDistance / rightRange

l = leftDistance / leftRange

fl2 = fleft2Distance / fleft2Range

fl0 = fleft0Distance / fleft0Range

fl1 = fleft1Distance / fleft1Range

fr0 = fright0Distance / fright0Range

fr1 = fright1Distance / fright1Range

fr2 = fright2Distance / fright2Range

fr = frontDistance / frontRange

speed = maxSpeed \* frontDistance / frontRange

driver.setCruisingSpeed(speed)

# brake if we need to reduce the speed

speedDiff = driver.getCurrentSpeed() - speed

if speedDiff > 0:

driver.setBrakeIntensity(min(speedDiff / speed, 1))

else:

driver.setBrakeIntensity(0)

#if((l>0.6)and(fl2>0.6)):

if(r<0.575):

angle1 = 0.018\*(-1+(1-fl2)+(1-fl0)+(1-fl1)-(1-fr)\*4-(1-fr1)-(1-fr0)-(1-fr2)-(1-r))

driver.setSteeringAngle(angle1)

if(r>0.577):

angle1 = 0.018\*(1-(1-fl2)-(1-fl0)-(1-fl1)+(1-fr)\*4+(1-fr1)+(1-fr0)+(1-fr2)+(1-r))

#angle = 0.02\*(1+(1-fl2)+(1-fl1)+(1-fl0)-(1-fr2)-(1-fr1)-(1-fr0)+(1-fr))

driver.setSteeringAngle(angle1)

**Результат:**

