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

Отчёт по лабораторной работе №3

по дисциплине

Аппаратное и программное обеспечение роботизированных систем

РУКОВОДИТЕЛЬ:

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

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Антропов А.Э.

19-В-1

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

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

Нижний Новгород 2022

**Задача .**  **Управление автомобилем**

Этот тест направлен на разработку компьютерной программы, которая управляет машиной, которая, уклоняясь от препятствий, двигается вперед. У машины есть датчики движения, которые распознают преграду, от нее машина пытается отклониться подальше, снижая скорость; также машина, уйдя от препятствия, набирает скорость, но максимальная скорость ограничена в 80 единиц;

**Описание работы::**

1. сначала мы указали максимальную скорость, подключили датчики движения:

maxSpeed = 80

driver = Driver()

driver.setSteeringAngle(0.0) # go straight

1. потом мы получили и включили датчиков расстояния

# get and enable the distance sensors

for name in sensorsNames:

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

sensors[name].enable(10)

1. теперь получили и включили GPS для навигации и позиционирования машины:

# get and enable the GPS

gps = driver.getGPS('gps')

gps.enable(10)

1. подключили камеру:

camera = driver.getCamera('camera')

camera.enable(50)

camera.recognitionEnable(50)

1. отрегулировали скорость в соответствии со значением, возвращаемым датчиком расстояния спереди:

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

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

rleftDistance = sensors['rear left'].getValue()

rleftRange = sensors['rear left'].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()

frightlRange = 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()

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

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

Нахождение дистанции от возможного препятствия, для того чтобы подальше уйти от него:

r = rightDistance / rightRange

rl = rleftDistance / rleftRange

l = leftDistance / leftRange

f12 = fleft2Distance / fleft2Range

f10 = fleft0Distance / fleft0Range

f11 = fleftlDistance / fleftiRange

fr0 = fright0Distance / fright0Range

fr1 = frightiDistance / frightiRange

r2 = fright2Distance / fright2Range

fr = frontDistance / frontRange

speed = maxSpeed \* frontDistance / frontRange

driver.setCruisingSpeed(speed)

1. если нужно снизить скорость, то машина тормозит, значения подбирали путем подбора, находим угол поворота:

speedDiff = driver.getCurrentSpeed() - speed

if speedDiff > 0:

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

else:

driver.setBrakeIntensity(0)

if((l>0.2)or(f12>0.6)):

if(r>0/5):

anglel = 0.02\*(-1+(1-f12)+(1-f10)+(1-f11)-(1-fr)\*4-(1-fr1)-(1-fr0)-(1-fr2)-(1-r))

driver.setSteeringAngle(anglel)

else:

angle 0.03\*(-1+(1-f12)+(1-f10)+(1-f11)-(1-fr)\*4-(1-fr1)-(1-fr0)\*5-(1-fr2)-(1-r))

driver.setSteeringAngle(anglel)

else:

if((l>0.2)or(f12>0.4)):

angle = 0.025\*((1-1)+(1-f12)+(1-f10)+(1-f11)+(1-fr)\*4-(1-fr2)\*4-(1-fr1)\*4-(1-fr0)\*4-(1-r)\*2-rl\*0.25)

driver.setSteeringAngle(anglel)

elif((l<0.2))or(f12<0.4)):

angle = 0.025\*((1-1)+(1-f12)+(1-f11)+(1-f10)-(1-fr2)-(1-fr1)-(1-fr1)-(1-fr0)-(1-fr)-(1-r)+(1-rl)

if (angle<0.0):

driver.setSteeringAngle(0.01)

else:

driver.setSteeringAngle(anglel)

**Приложение (листинг):**

"""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 = 80

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')

# uncomment those lines to enable the 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()

rleftDistance = sensors['rear left'].getValue()

rleftRange = sensors['rear left'].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()

frightlRange = 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()

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

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

r = rightDistance / rightRange

rl = rleftDistance / rleftRange

l = leftDistance / leftRange

f12 = fleft2Distance / fleft2Range

f10 = fleft0Distance / fleft0Range

f11 = fleftlDistance / fleftiRange

fr0 = fright0Distance / fright0Range

fr1 = frightiDistance / frightiRange

r2 = 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>0.2)or(f12>0.6)):

if(r>0/5):

anglel = 0.02\*(-1+(1-f12)+(1-f10)+(1-f11)-(1-fr)\*4-(1-fr1)-(1-fr0)-(1-fr2)-(1-r))

driver.setSteeringAngle(anglel)

else:

angle 0.03\*(-1+(1-f12)+(1-f10)+(1-f11)-(1-fr)\*4-(1-fr1)-(1-fr0)\*5-(1-fr2)-(1-r))

driver.setSteeringAngle(anglel)

else:

if((l>0.2)or(f12>0.4)):

angle = 0.025\*((1-1)+(1-f12)+(1-f10)+(1-f11)+(1-fr)\*4-(1-fr2)\*4-(1-fr1)\*4-(1-fr0)\*4-(1-r)\*2-rl\*0.25)

driver.setSteeringAngle(anglel)

elif((l<0.2))or(f12<0.4)):

angle = 0.025\*((1-1)+(1-f12)+(1-f11)+(1-f10)-(1-fr2)-(1-fr1)-(1-fr1)-(1-fr0)-(1-fr)-(1-r)+(1-rl)

if (angle<0.0):

driver.setSteeringAngle(0.01)

else:

driver.setSteeringAngle(anglel)

Результаты:

