山东大学

崇新学堂

2022－2023 学年第一学期

实 验 报 告

课程名称： EECS DesignLab

实验名称： DesignLab10

专 业 班 级 崇新 21

学 生 姓 名 施政 刘浩 张原 池弋

实 验 时 间 2022.11.25

Step1

In this step, we connect analog pin # 1 to the output of the potentiometer below the screwdriver to measure the voltage change caused by the rotation of the head, which is used to control the rotation of the body. When the head is in the middle, the output voltage ranges from 4.8V~5.2V. When the body should turn left (namely, the head should turn left), the voltage should be less than 4.8V. Otherwise, the voltage should be more than 5.2V. The corresponding code is as follows:

```
def getNextValues(self, state, inp):
    V_Pot = inp.analogInputs[1]#voltage of the pot
    V_Motor = inp.analogInputs[2]#voltage of the
    V_Pot_half = 5
    V_half = 10.5
    V_high = 10.2
    V_low = 9.8
    k1 = 0.5 #rotate gain
    k2 = 0.1 #speed gain
    rotatev = k1*(V_Pot_half - V_Pot)
    forwardv = k2*(V_Motor - V_half)
    print V_Pot
    print V_Motor

if state == 'stop':
    if V_Pot <= 4.8 or V_Pot >= 5.2:
        return ('rotate', io.Action(fvel = 0, rvel = rotatev))
    else:
        if V_Motor >= V_low and V_Motor <= V_high:
            return ('stop', io.Action(fvel = 0, rvel = 0))
        else:
            return ('go', io.Action(fvel = forwardv, rvel = 0))
```

Figure1 The corresponding code

Step2

This step requires the robot to stand still when the light is off, and when the light is on, it should move closer until it is half a meter away from the light. Thus, we propose two solutions:

1.Take advantage of the property that the left and right photoresistors have different rates of change with respect to light. (利用左右两侧光敏电阻随光强的变化率不同)

Figure2 Closer to the light and farther away from the light

We aim the light at the center, move the light source from near to far, and measure the voltage at M+ (that is, the voltage of the photosensitive resistance partial voltage after amplification). It is found that it changes monotonically, which proves that this idea is feasible.

Table1 Voltage at different distances

Distance/m	0.1	0.2	0.3	0.4	0.5	0.6
Voltage/V	12	11.9	11.64	11	10.48	10

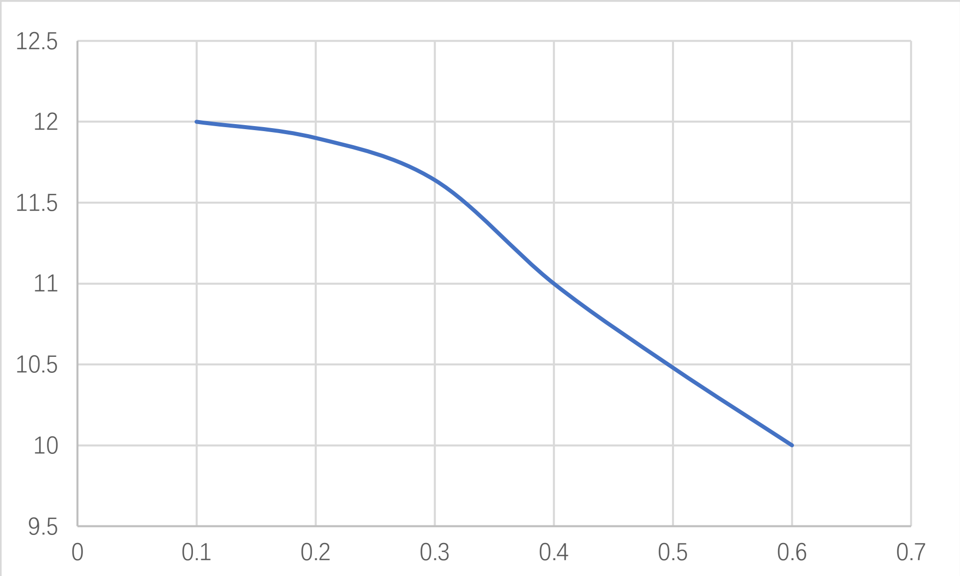

Figure3 The voltage varies with distance

Therefore, the only thing we need to do is to utilize the analog pin to measure the voltage of M+ and compare it with 10V. By making the difference between the output voltage and 10V and multiplying the gain, we can achieve the function of moving away from the light while too close and approaching while too far away.

```
if state == 'stop':
    if V_Pot <= 4.8 or V_Pot >= 5.2:
        return ('rotate', io.Action(fvel = 0, rvel = rotatev))
    else:
        if V_Motor >= V_low and V_Motor <= V_high:
            return ('stop', io.Action(fvel = 0, rvel = 0))
        else:
            return ('go', io.Action(fvel = forwardv, rvel = 0))
```

Figure4 The corresponding code

2. A resistance is connected in series with the photoresistors to realize the voltage divider. (额外串联一个电阻分压)

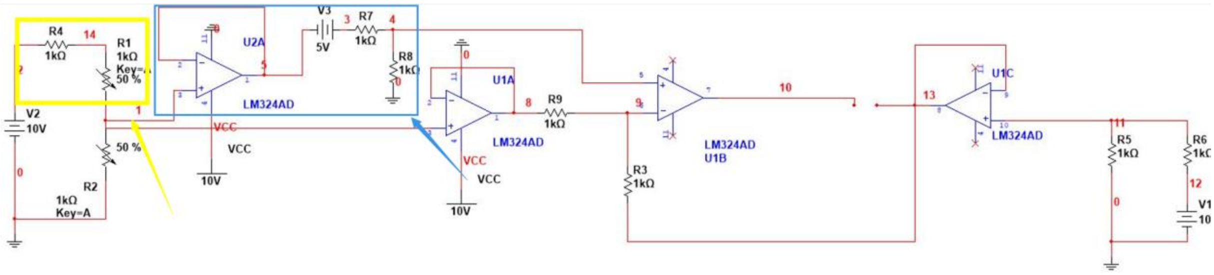

Figure5 Circuit diagram of the second scheme

That is, use the output voltage in the middle of the photosensitive resistor to measure the light intensity and judge the distance.

This solution starts from the assumption that both photoresistors have the same rate of change with light, so we attach a fixed resistance to one photoresistor in series. At this point, the problem is transformed into two keys:

(1). Does the output voltage between two photoresistors change monotonically with distance? (The distance can only be judged by the voltage when it changes monotonically)

(2). How to solve the problem that with the change of the output voltage of the photoresistors, the voltage at the positive electrode of the motor is also changing correspondingly?

Since the response of the head to the light is extremely fast, we can therefore only consider the case where the light is in the middle, which means that the two photoresistors have the same resistance value. Thus, for the first problem, the divided voltage formula is:

$$U_O = \frac{R_l}{2R_l + R} = \frac{1}{2 + \frac{R}{R_l}}$$

Here, R_l represents the photoresistors, and R represents the arbitrary resistance value of the serial resistance. Clearly, U_O varies monotonically with R_l.

For the second problem, we have two solutions:

(1) Adjust the input reference voltage at the other end of the reverse amplifier;

(2) Adjust the negative pole position of the motor (namely, M-).

Since the voltage of the positive electrode will vary with distance, the voltage in both schemes should also vary with distance. Consider the output voltage of the inverting amplifier with reference voltage in hw2.

$$(V_0 - V) = -\frac{R_2}{R_1}(V_2 - V) = -(V_2 - V)$$

In the first scheme, the only thing we need to do is to ensure that V_0=$V_{M-} = 5V$ when the resistance values of the two photoresistors are equal, and the reference voltage is obtained as follows:

$$V = \left(\frac{R_l}{2R_l + R} + 5\right)/2$$

In the second scheme, considering$(V_0 - 5) = V_{M-}$, the voltage of the negative electrode is obtained as follows:

$$V_{M-} = \left(\frac{R_l}{2R_l + R} + 5\right)/2$$

It is found that as long as we get the voltage above, either connect it to the reference point of the inverting amplifier or the negative electrode of the motor, the target can be achieved. The detailed circuit diagram is as follows:

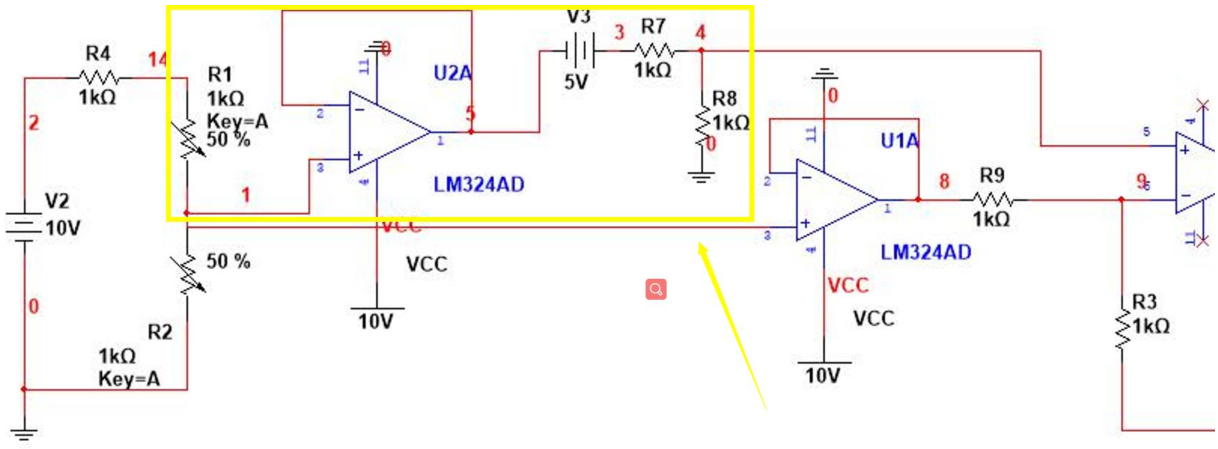

Figure6 Circuit diagram of the formula

Step3

In order to achieve the two functions of walking along the wall and finding light, we change the original "go" to walking along the wall. We also imported the code from dl2 with a slight change, thus we can choose to use either “dl2” or “boundaryFollower”. We just need to comment or uncomment it when test it. The corresponding code is as follows:

```
######################################################################
###
###   If you want to call boundaryFollower.py to
###   implement the eyewall walk,please uncomment the following
###
######################################################################
from boundaryFollower import boundaryFollowerClass

######################################################################
###
###   If you want to call DL2.py to
###   implement the eyewall walk,please uncomment the following
###
######################################################################
##from DL2 import DL2Class
```

```
    if V01 <= 4.8 or V01 >= 5.2:##rotate
        return state, io.Action(fvel = 0, rvel = rotatev)
    else:
        if V02 >= V_low and V02 <= V_high:##stop
            return state, io.Action(fvel = 0, rvel = 0)
        else:##go/boundaryfollow
    ##########################################################################################
    ##If you want to call boundaryFollower.py to implement the eyewall walk,please uncomment the following
    ##########################################################################################
            return boundaryFollowerClass.getNextValues(self, state, inp)
    ##########################################################################################
    ##If you want to call DL2.py to implement the eyewall walk,please uncomment the following
    ##########################################################################################
            ##return DL2Class.getNextValues(self, state, inp)

vSM = MySMClass()
```

Figure6 The corresponding code of step3

Summary

1.Troubles

At first, we thought we were writing code using a series of formulas in homework2, so we spent a lot of time trying to figure out the relationship between the parameters on the hardware and the formulas in hw2. Later, it was found that the formulas in hw2 were only used to control the rotation of the head, which was the task of the previous experiment, that is, to realize the fastest response of the head to the light. However, the task in dl10 was to control the rotation and progress of the car. We should compare the output voltage of the potentiometer with the midpoint voltage, and use the difference to control the rotation, rather than use the formulas to design the circuit and coding.

2. Problems

In the actual test, we found that some analog input pins have something wrong. When we use the voltmeter to measure the potentiometer voltage output, the voltage is a continuously varying from 1V to 9.8V. But when using the analog input pin to measure, no matter how the potentiometer changes, the value displayed on the computer is constant 10V, or sometimes it will mutate to about 7V and change back to 10V. This may be caused by some problems with the internal circuit.

3. Improvement

(1) Unfortunately, due to the broken power supply of the robot, weak connection lines, not firmly fixed head, the circuit aging and other reasons, it is difficult to achieve the desired effect of the car during actual measurement. In addition, for the second scheme in step2, since the actual circuit is too complex and the head is sometimes good and bad, we failed to implement this scheme in the field measurement, only theoretical analysis and discussion were carried out.

(2) When we use the analog pin to collect the input signal, because the position of the potentiometer is uncertain in the initial situation, if we take the fixed voltage as the standard, we must adjust the potentiometer to the midpoint (output voltage is at the midpoint) before each start. In order to avoid tedious operations, we can initialize the code by collecting the output voltage multiple times, filtering, normalizing and so on, and taking the processed data as the standard. Of course, in this case, other relevant compared voltages such as V_low, V_high and etc. should also be associated with this value. Thus, as long as the potentiometer is in an appropriate position that is not in an extreme position, we do not need to manually zero before starting (convert manual zero setting into code zero setting), which improves the robustness of the code to a certain extent.