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： 第五次硬件实验

专 业 班 级 崇新学堂 21 级

学 生 姓 名 刘浩、施政、张原、池弋

实 验 时 间 2022.10.18

Step1 and 2

We completed the procedure and used three k values, **k**=1, **k**=5, **k**=10, to test whether our simulation works correctly. Below is our program and images of the results of our experiment.

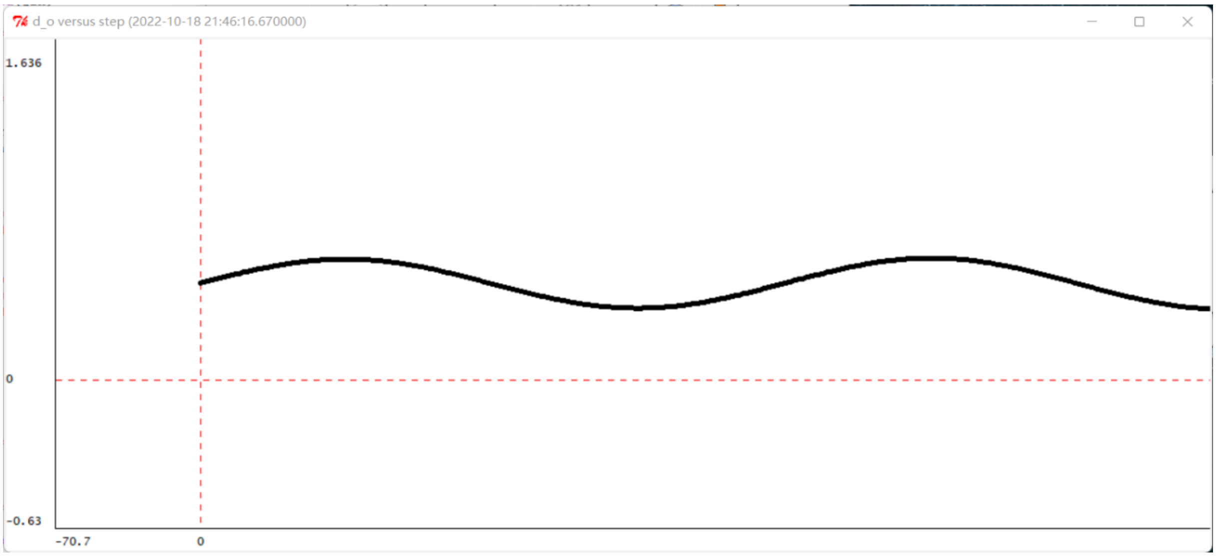
```
desiredRight = 0.5
forwardVelocity = 0.1
k=1

# No additional delay
class Sensor(sm.SM):
    def getNextValues(self, state, inp):
        return (state, sonarDist.getDistanceRight(inp.sonars))

# inp is the distance to the right
class WallFollower(sm.SM):
    def getNextValues(self, state, inp):
        print(inp)
        return state, io.Action(fvel=forwardVelocity, rvel=k*(desiredRight-inp))
```

Check Yourself 1.

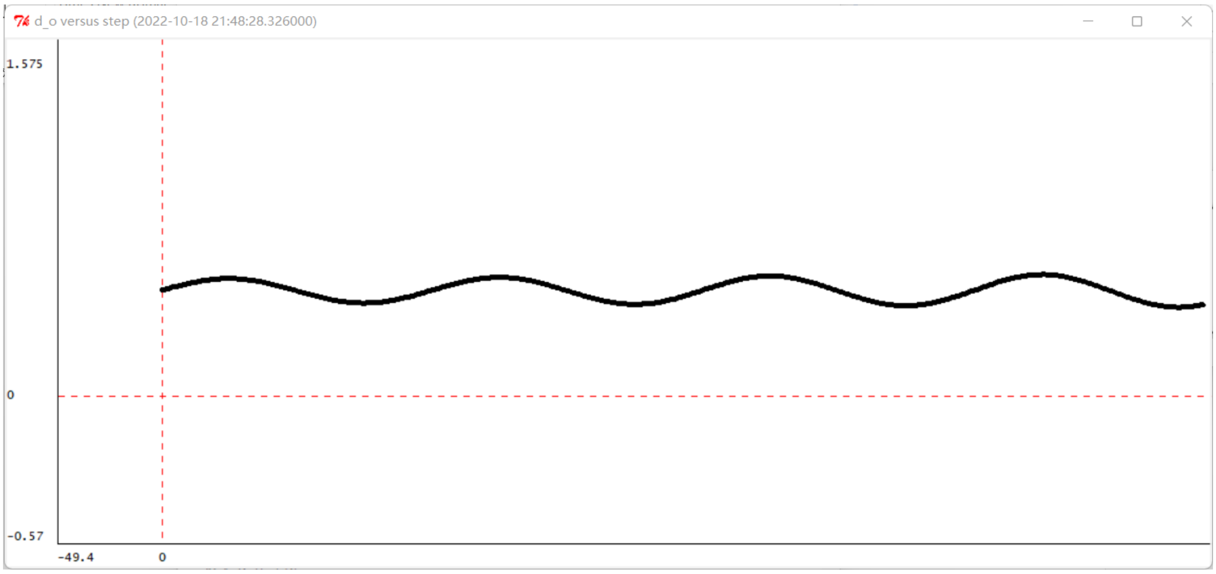
The input of WallFollower is the distance from the wall, and the out put is io.Action (The behavior of the car). The sign of **k** is rad/m .

Checkoff 1.

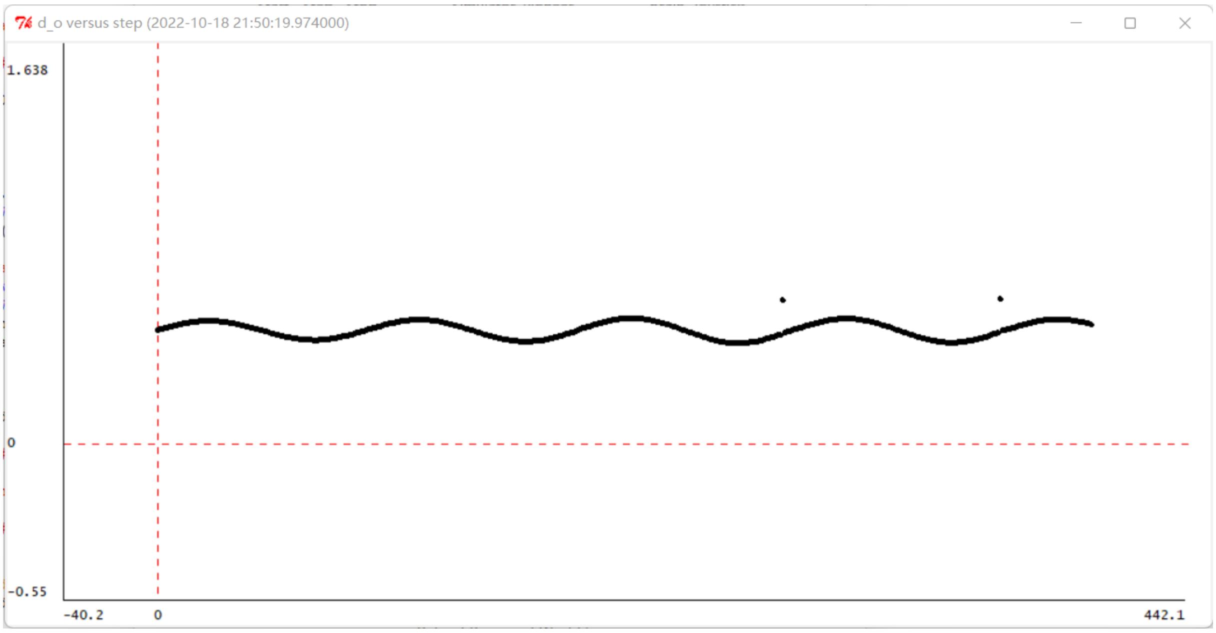
When **k** = 1, T=0.1, V=0.1 the distance to the wall is between 0.35 and 0.65, its period is about 30s. When **k** = 5, it is between 0.43 and 0.57, its period is about 13s. And if **k** = 10, it is between 0.45 and 0.55, its period is about 9s. We find that as **k** continues to grow, the distance is also getting closer and closer to our expectations, and the period is getting shorter and shorter. So we suspect that when **k** increases, the distance becomes more and more desirable, so there is no **k** to make this approach the fastest.


d_o versus step (2022-10-18 21:46:16.670000)
1.636
0
-0.63
-70.7
0

k=1


d_o versus step (2022-10-18 21:48:28.326000)
1.575
0
-0.57
-49.4
0

k=5



k=10

Step3 and 4:

Controller model: $\omega(n) = k \cdot e(n)$

Plant 1: $\theta(n) = T \cdot \omega(n-1) + \theta(n-1)$

Plant 2: $\mathrm{do}(n) = TV \cdot \theta(n-1) + \mathrm{do}(n-1)$

Step5 and 6:

The H and p that we have calculated is as follows:

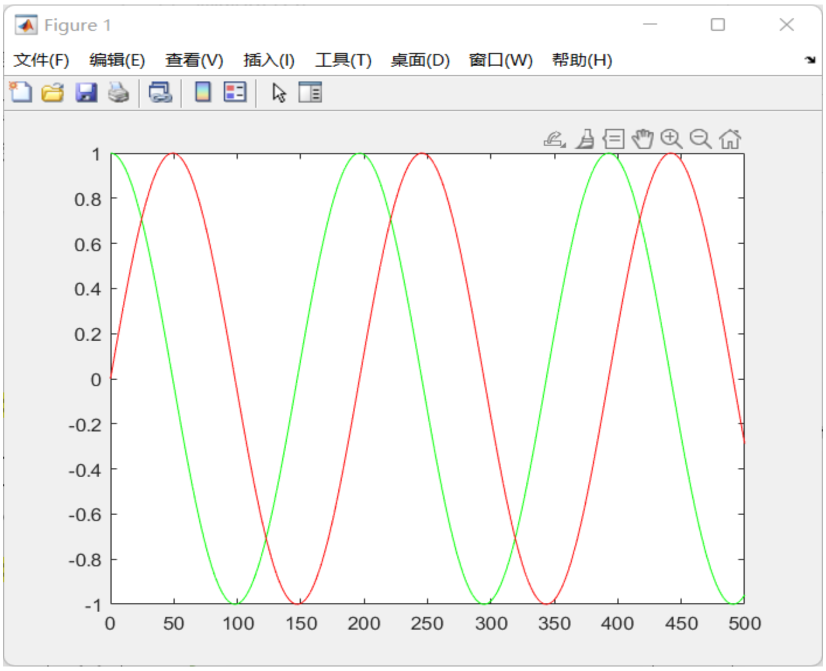
$H = KVT^2R^2/(KVT^2R^2 + (1-R)^2)$

$p = \sqrt{1 + kvT^2}e^{\pm i\varphi}, \varphi = arctan\sqrt{kvT^2}$

Step7:

When k = 1, T = 0.1 second, and V = 0.1 m/s, p≈ $e^{\pm i \cdot 0.032}$, the real and imaginary parts of p is as followed.

$$p = e^{i \cdot 0.032}$$



Green represents real and red represents imaginary

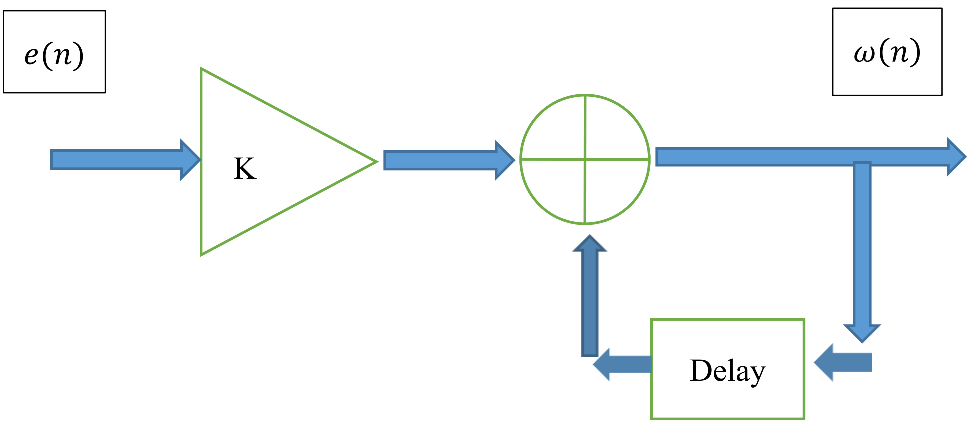
$do(n) = \sqrt{0.001k}(\sqrt{1+0.001k})^n \cdot \sin\left(n\left(arctan\sqrt{0.001k}\right)\right)$ (n>0)

Checkoff 2.

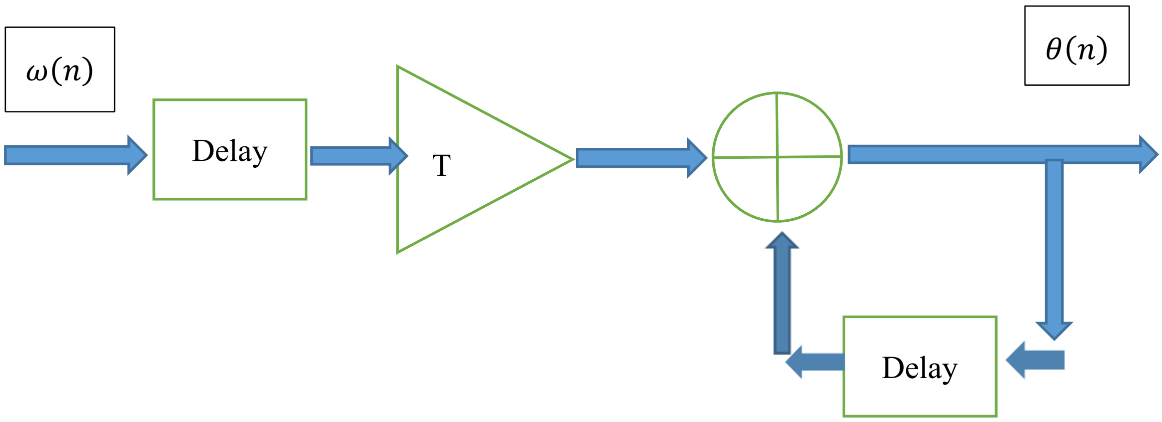
We can find out the current precise position of the trolley through the pole. And this result is an exact value, but in step one, because of measurement interval, the tipping point is often a range, and the result is a relative value. The period is the same.

Check Yourself 2:

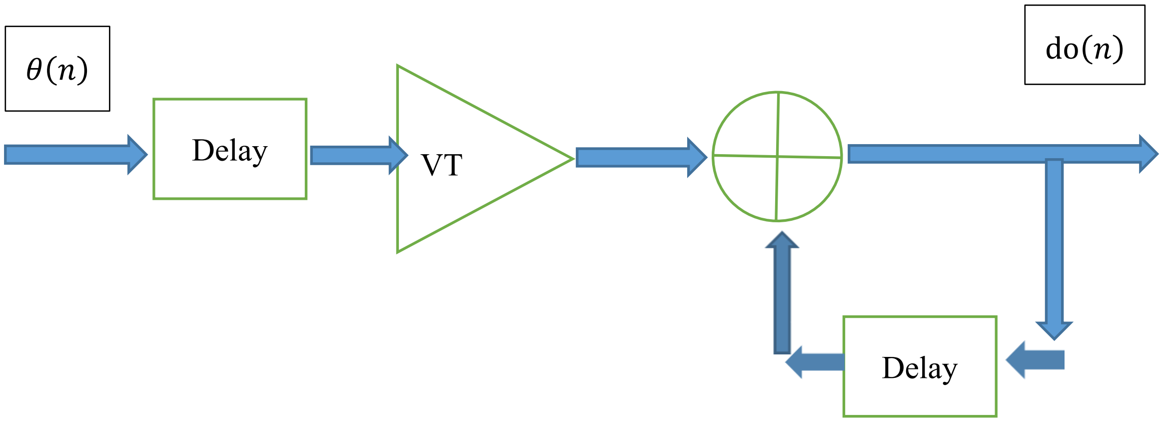
The system block diagram is as follows



Controller model

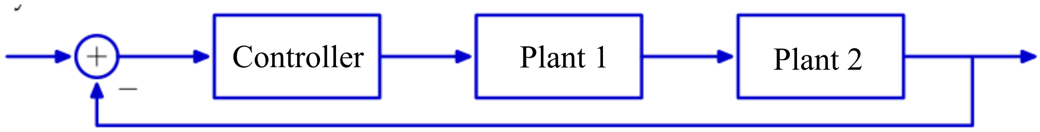


Plant 1



Plant 2

Finally, block diagrams of the entire system can be combined in the following ways



Step8 and 9:

The program of our group is as follows:

```
import lib601.sig as sig
import lib601.ts as ts
import lib601.poly as poly
import lib601.sf as sf

def controller(k):
   return sf.Gain(k)

def plant1(T):
   return sf.Cascade(sf.Cascade(sf.Gain(T),sf.R()),sf.FeedbackAdd(sf.Gain(1),sf.R()))

def plant2(T, V):
   return sf.Cascade(sf.Cascade(sf.Gain(V*T),sf.R()),sf.FeedbackAdd(sf.Gain(1),sf.R()))

def wallFollowerModel(k, T, V):
   return  sf.FeedbackSubtract( sf.Cascade(sf.Cascade(controller(k),plant1(T)),plant2(T,V)))
```

Step10:

When T = 0.1 seconds, V = 0.1 m/s, k=1, the pole is (1+0.032j), the period is 62.5π.

When T = 0.1 seconds, V = 0.1 m/s, k=5, the pole is (1+0.071j), the period is 88.5.

When T = 0.1 seconds, V = 0.1 m/s, k=10, the pole is (1+0.1j), the period is 20π

$do(n) = \sqrt{0.001k}(\sqrt{1+0.001k})^n \cdot \sin(n(arctan\sqrt{0.001k}))$ (n>0)

When k getting bigger, the period is getting shorter, but according to the above expression, it can be seen that: when n is within a certain range, there is a maximum k value that keeps the $do(n)$ from diverging. But we haven't found this exact maximum k.

Checkoff 3.

The results in step 10 is a exact value to measure the length of the period. And the results in checkoff1 are relative value, it is the approximate time required for the trolley to finish the work. The same thing is both results can reflect the time required to complete this task, and the different thing is that step 10 is a exact value and checkoff1 is an relative value, and they were calculated by different methods.

Summary:

1. In this lab, we learned system functions and poles and applied them to a real system.

2. When we solve the system functions, due to the complex relationship between the cascade and feedback of the system, it is difficult for us to write the system functions in the form of general difference equations, and finally we use operator expressions to solve them.

3. In this lab, we are exposed to a more complex system than lab four, and we are able to break down complex systems into several simple systems and analyze them step by step, so that our ability to analyze the system has been improved.

4. We have a better understanding of the physical meaning of poles and the relationship between periods and poles, however, it is still difficult to solve the convergence and periodicity of the more complex complex poles.