**崇新学堂**

**2022－2023学年第一学期**

实 验 报 告

课程名称： Experiments of Introduction to EECS

实验名称： Staggering Proportions

专 业 班 级 崇新21

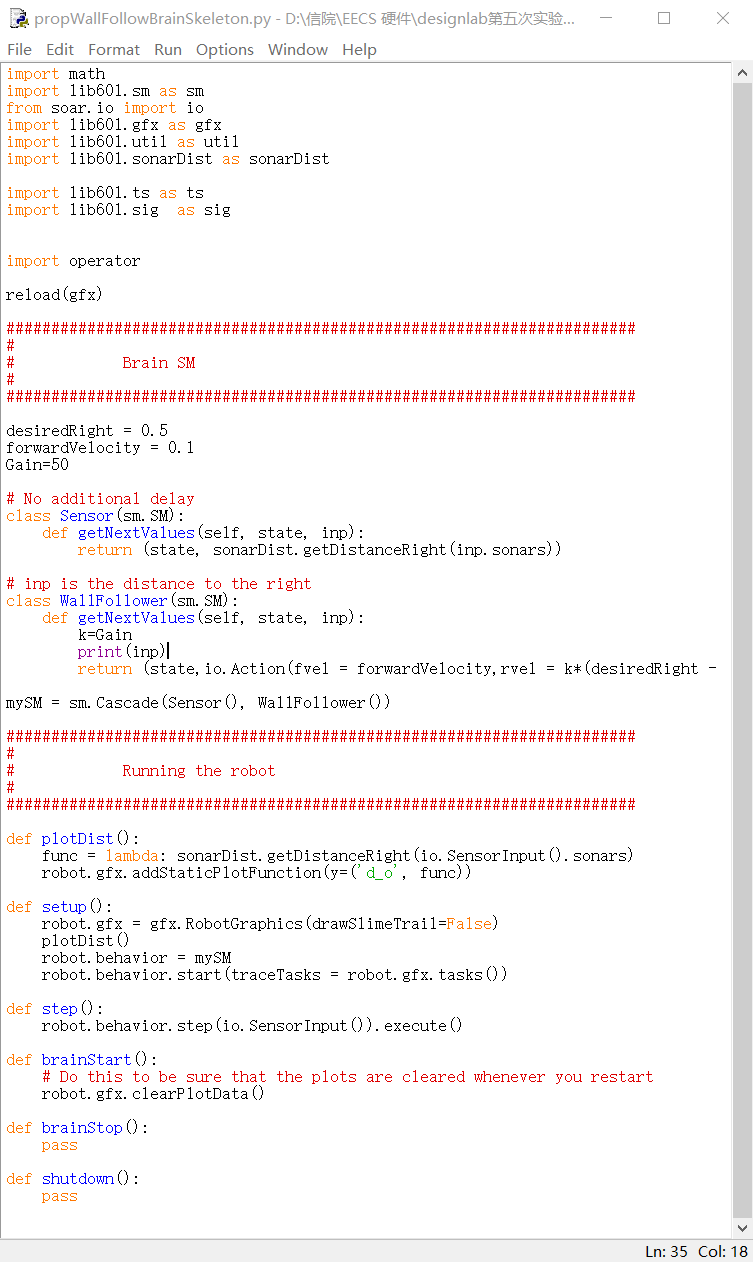
学 生 姓 名 崔宇鑫 余昊 葛明烨

实 验 时 间 2022年10月18日

***Check Yourself 1.* What should be the types of the input to and the output from a state machine of the WallFollower class? What should be the sign of k?**

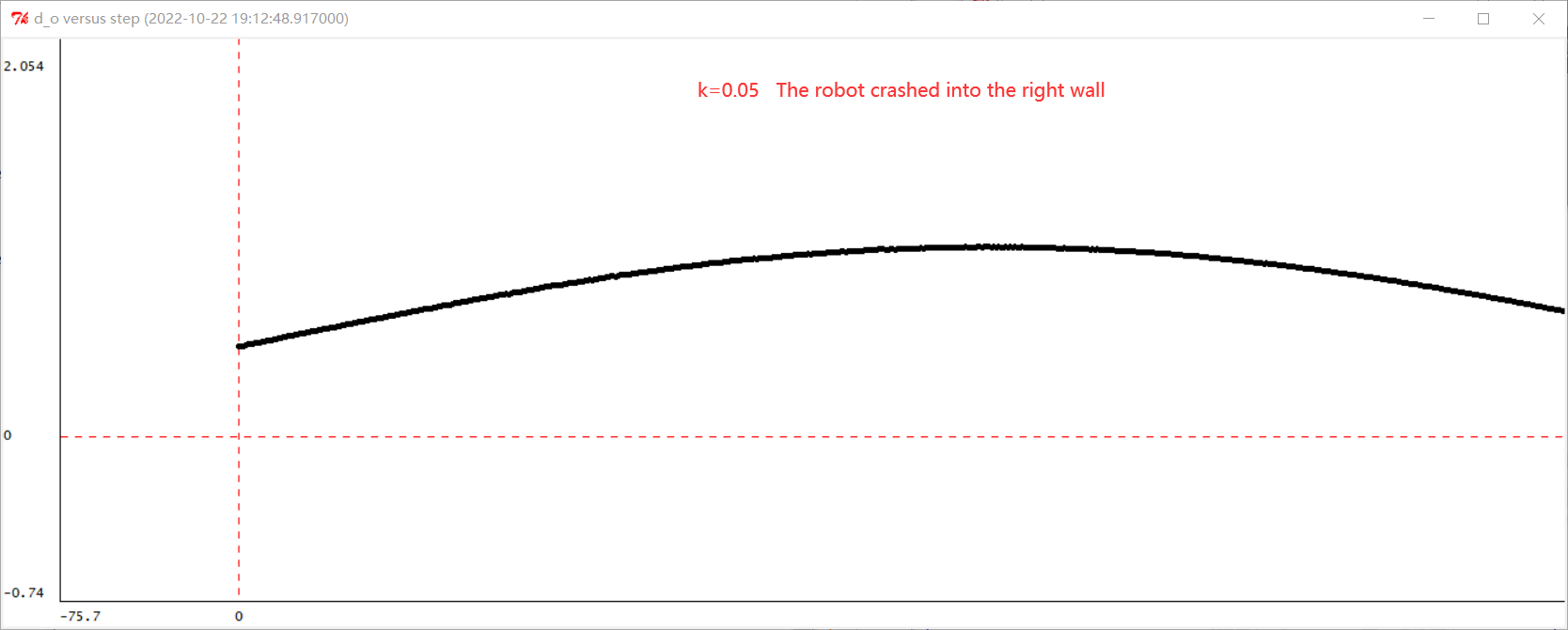
*The input is the actual distance to the right wall and the output is the next action of the robot including its forward velocity and rotational velocity. The sign of k is Gain.*

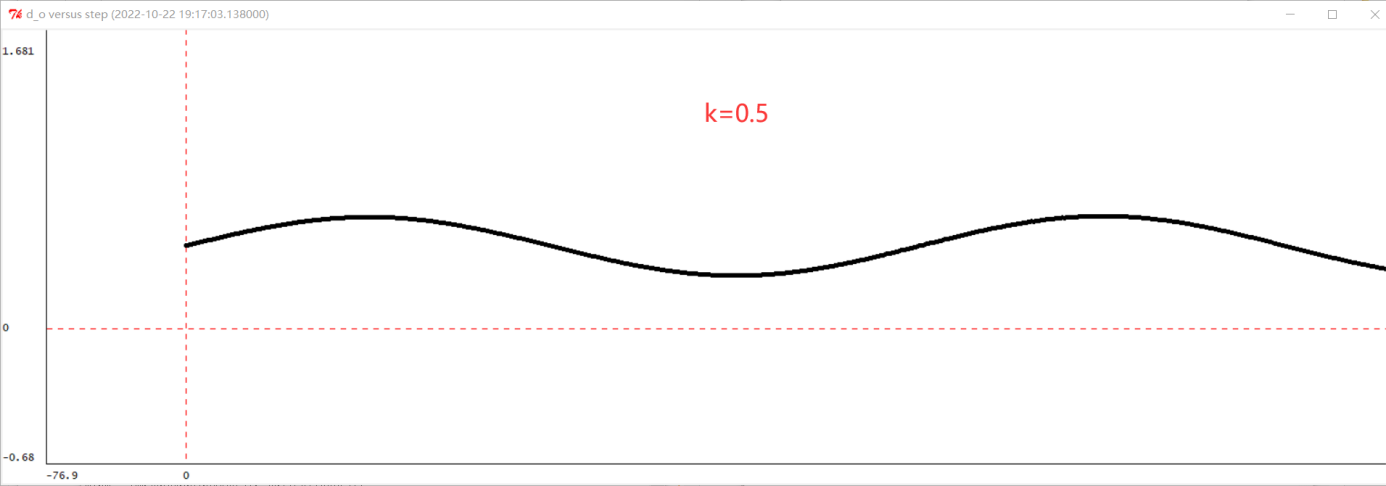
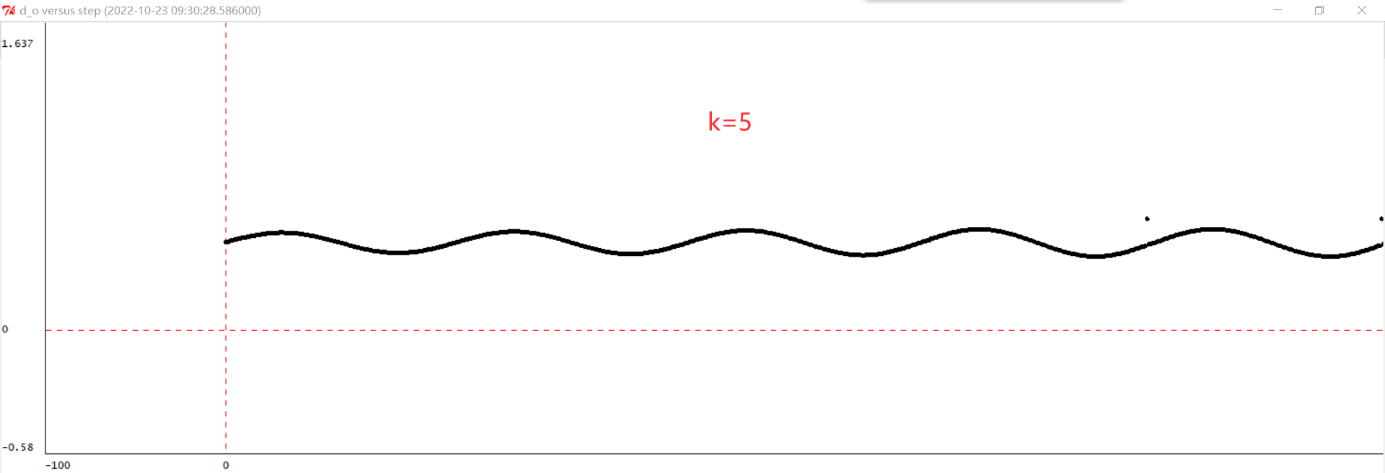
**Step1**

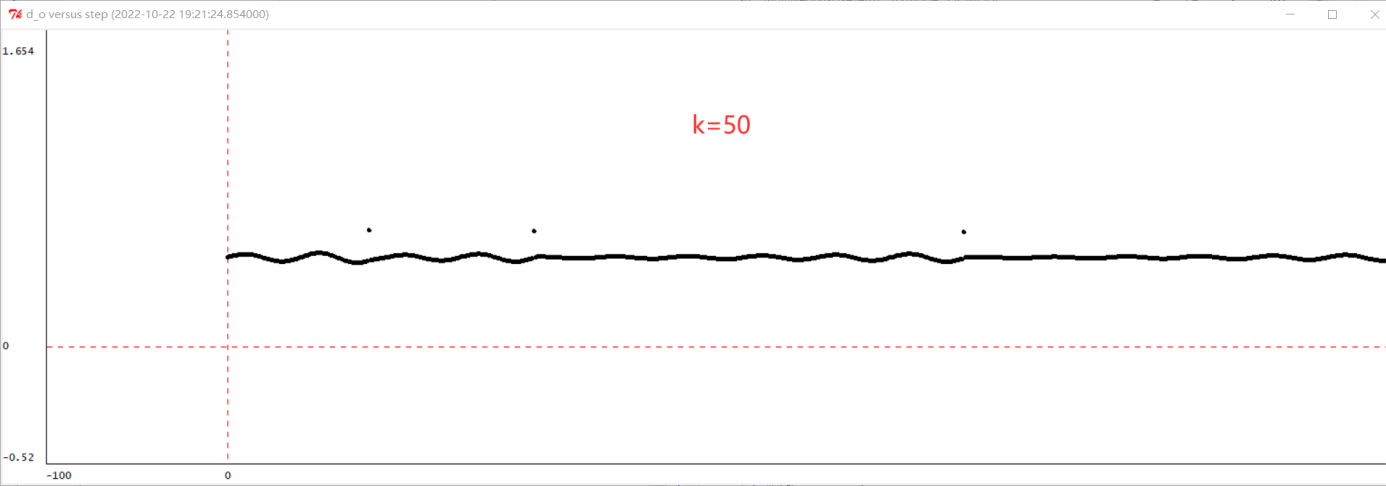
*My code for constructing this composite machine into propWallFollowBrainSkeleton.py*

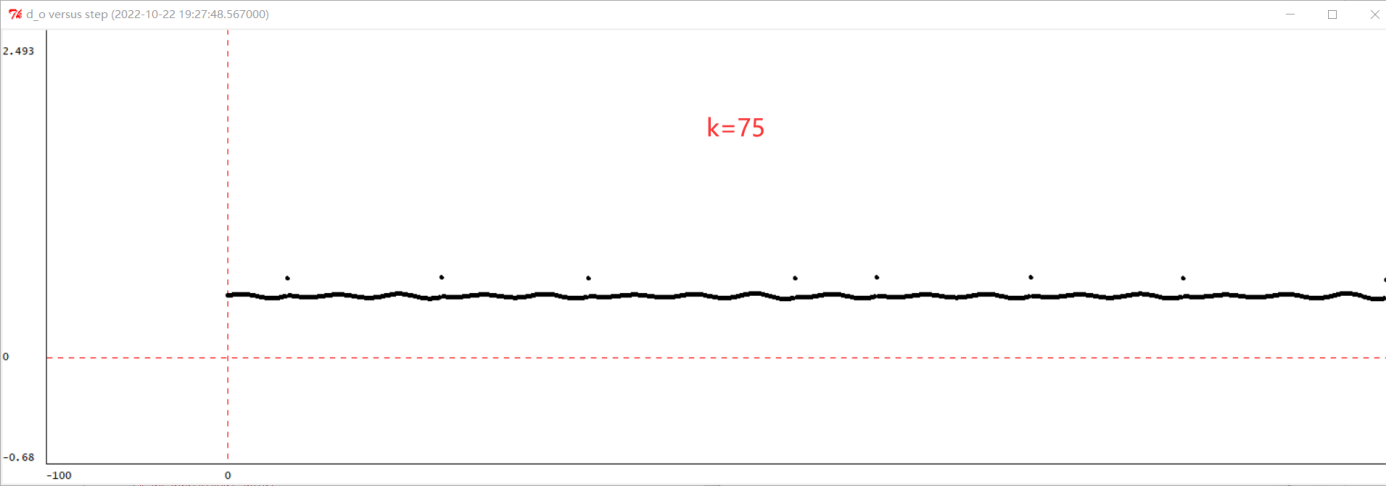
**Step2**

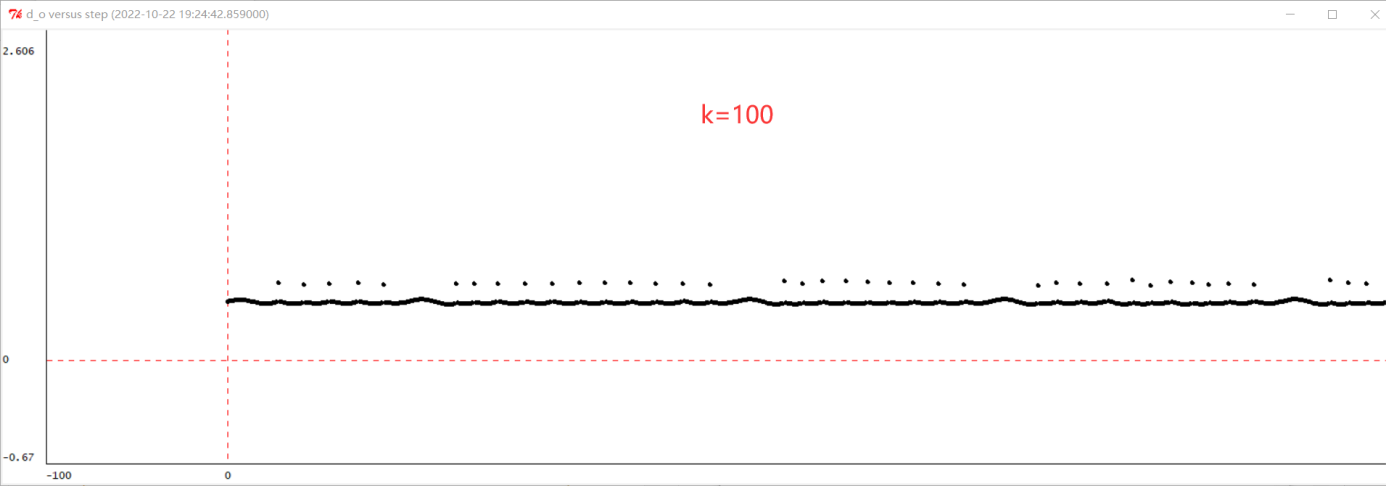
*Change k to observe the movement of the robot.*

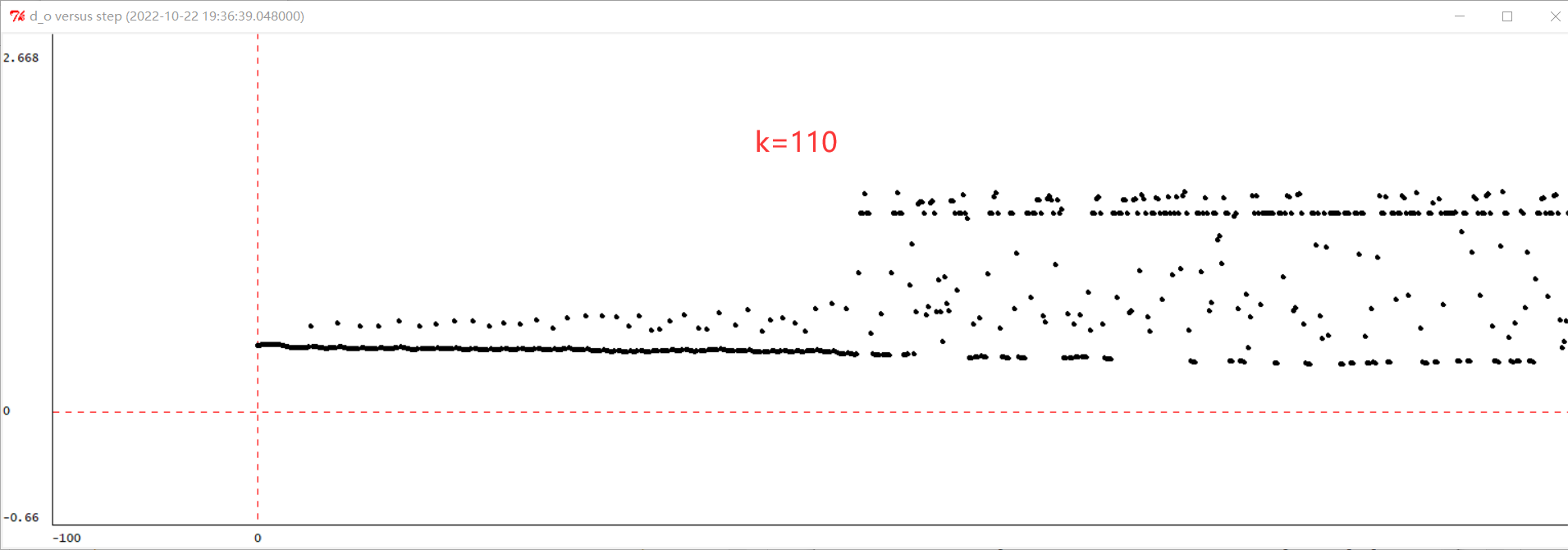












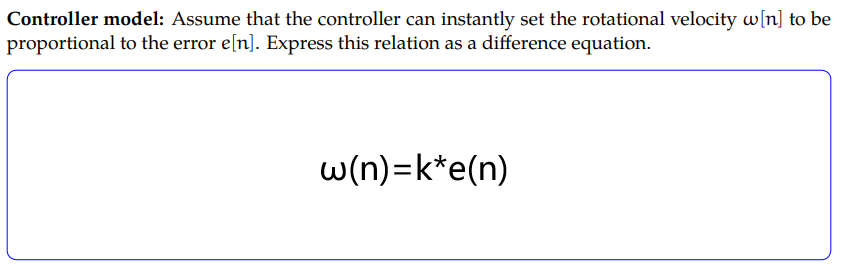
***Checkoff 1.*** **Show your plots to a staff member. Describe how the gain k affects behavior of the wall-follower. For what value of k (if any) does the distance to the wall converge most rapidly to the desired value of 0.5 m? Compare the effect of k in this lab and in the lab 4 (wall finder).**

*k influences the behavior of the robot by influencing the rotating speed of the robot. When k is greater than 0 and less than 100, The larger k is, the smaller the oscillation amplitude of the robot near the desired value is. When k is greater than 100, with the increase of k, the robot will shake heavily and move slowly because of too fast speed.*

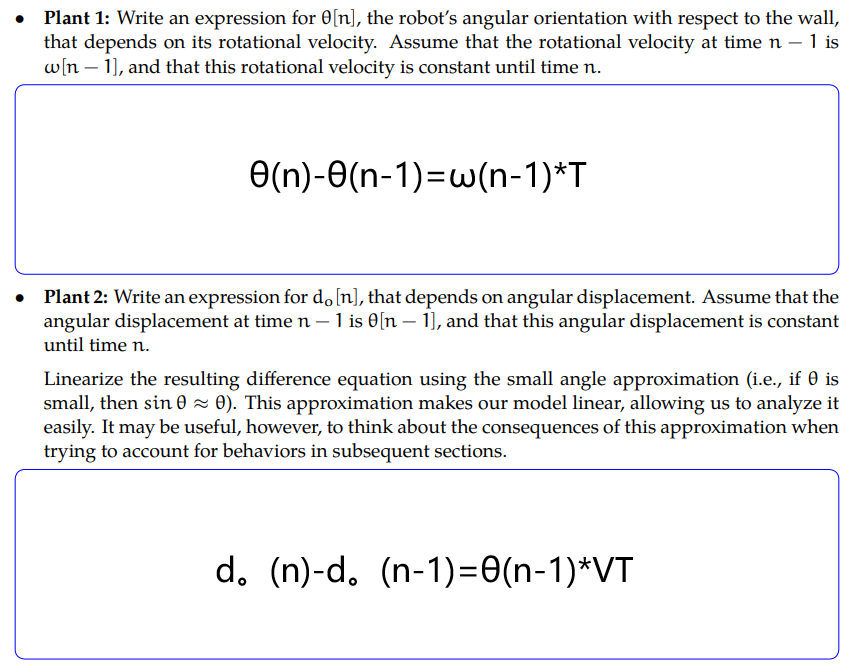
*We see that the function doesn't converge for any value of k. When k is equal to 75, because the amplitude of the behavior of the system is small, it can be approximated as convergence.*

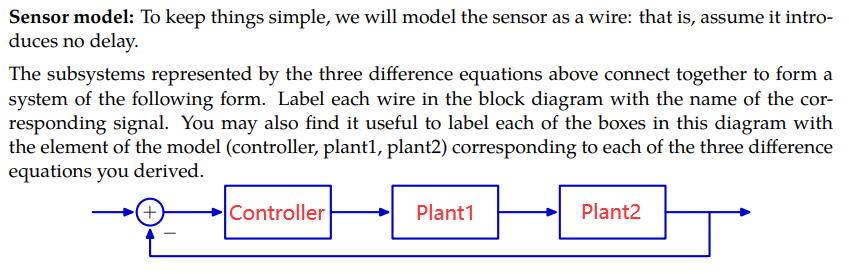
*In lab 4, k controls the magnitude of the forward speed and in lab 5, k controls the magnitude of the angular speed. In lab 4, we will find a k that makes the system behavior converge quickly to the expected value when we change the value of k, but in lab 5, we cannot find such a k value that makes the function converge quickly.*

**Step3**



**Step4**





**Step5**

We first transform the above block diagram into operator expressions.

Controller:

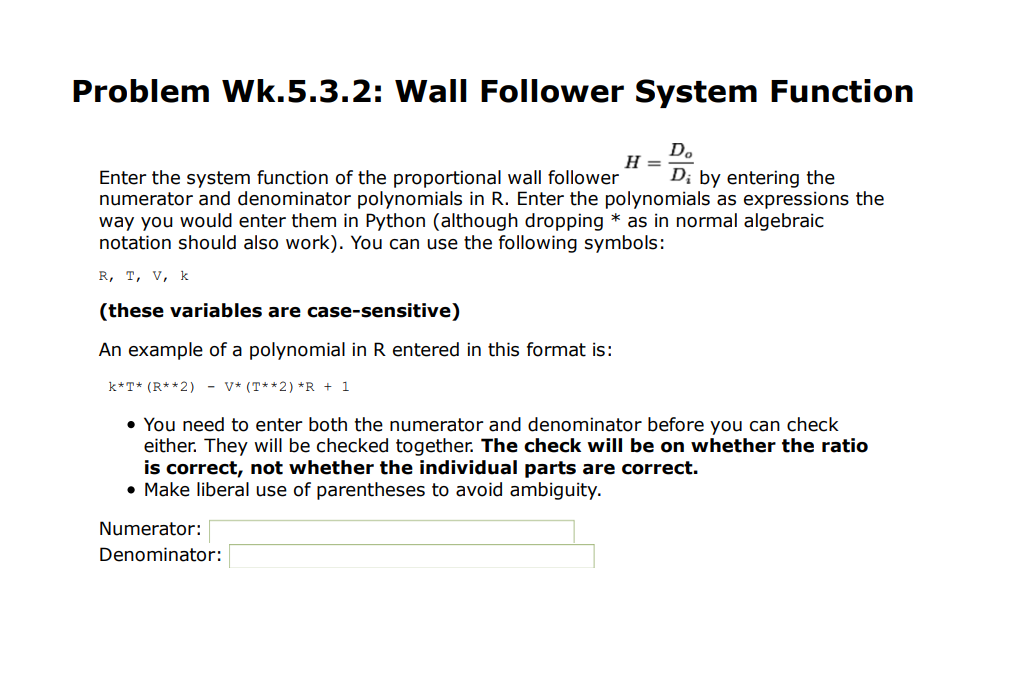
Plant1：

Plant2：

Sensor：

We then transform this equation into a system function.

*Wk.5.3.2.*



Answer:

*Number:*

*Denominator:*

**Step6**

*Let's first convert R to*

*Take the denominator polynomial and find its roots.*

**Step7**

*When k = 1, T = 0.1 second, and V = 0.1 m/s ,*

*We choose p1, then p1=*

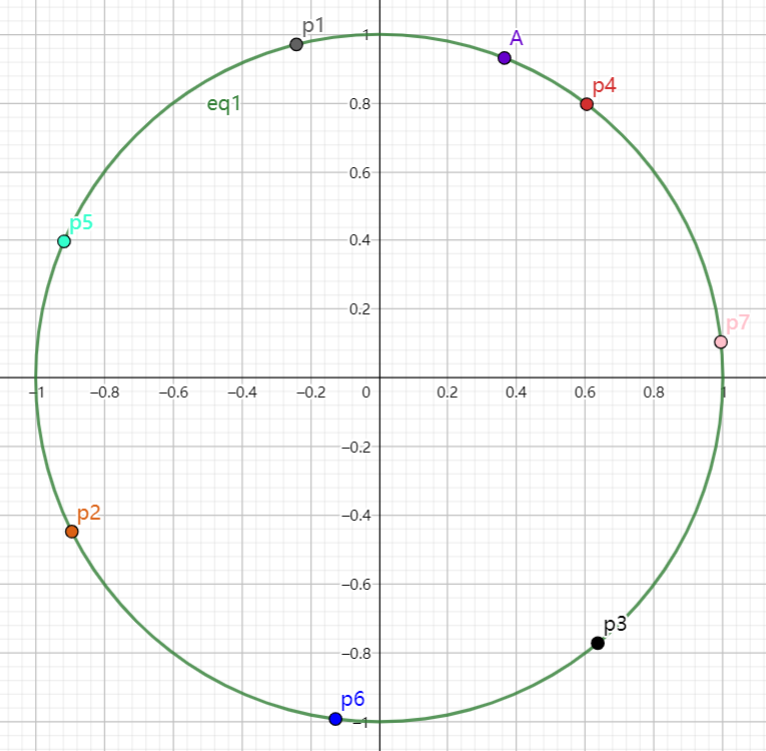
**

*The red line represents the real part, and the green line represents the imaginary part.*

*It shows that the response of system is periodic, with a period of*

***Checkoff 2.***

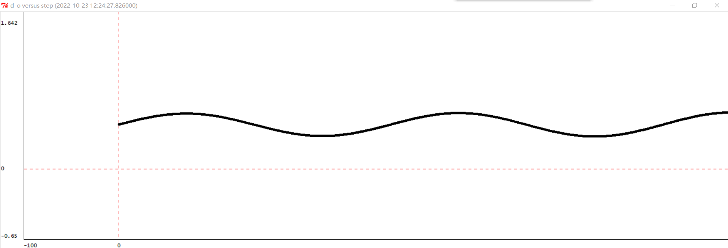
*Wk5.3.3* *In the complex plane, the magnitude of p1 is 1 and Ω is 0.0316224. Let's call pn for p to the NTH power. So, the position of the poles in the complex plane is as follows.*



*Point A refers to p8.*

*We find that the poles are all on the circle of r equals one. This tells us that the system behavior is always oscillating with period.*

*We change Gain=75 in propWallFollowBrainSkeleton.py. to Gain=1, and generate vehicle trajectory diagram.*

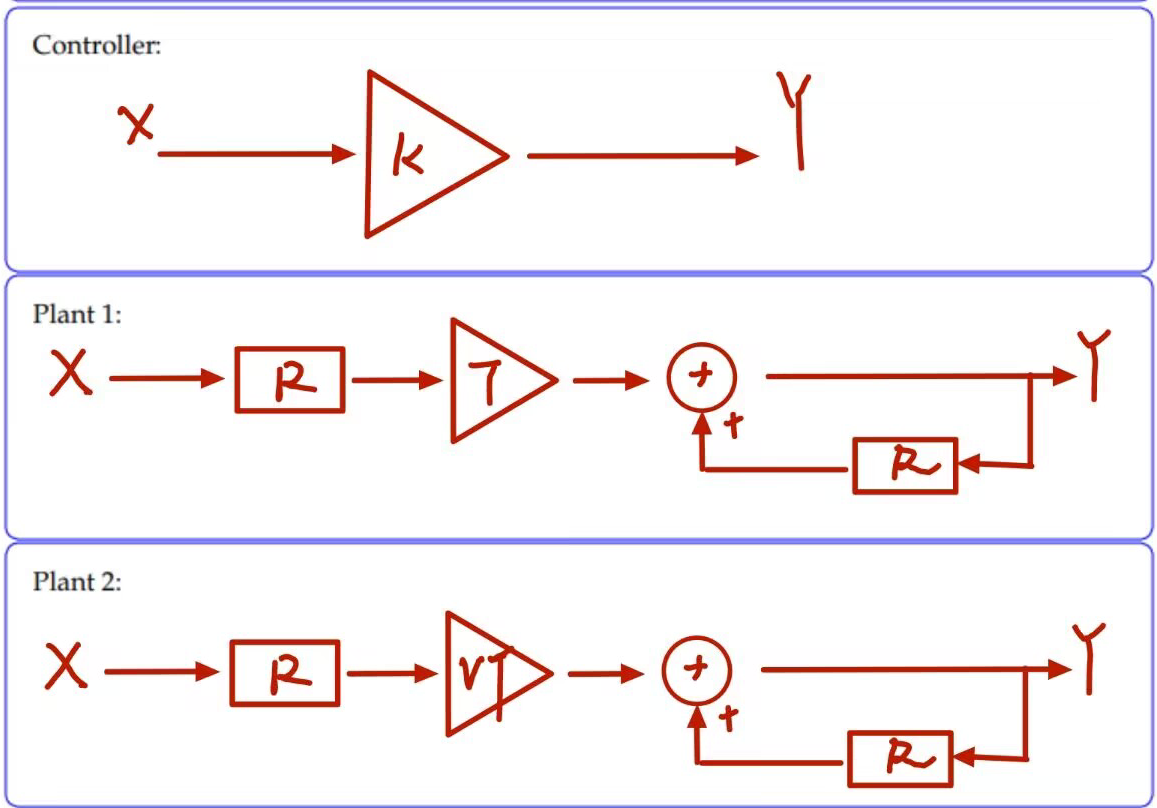


*In Checkoff 1, whatever k is, the system is always oscillating periodically. This agrees with the results we obtained in Step 7.*

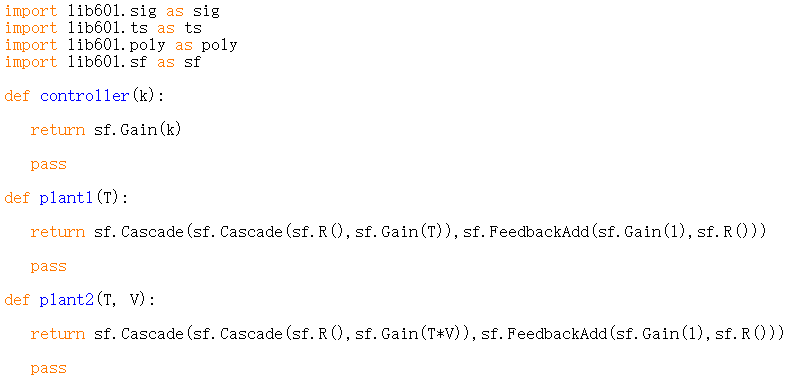
*We find that when k=1, the behavior of both systems is oscillating around the expected value.*

*Since the V of the two is the same and the time interval for the sensor to obtain data is the same, the period of the two is the same at this time*

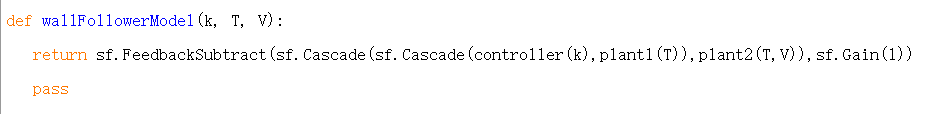
***Check Yourself 2.* Use gains, delays, and adders to draw system diagrams representing the controller, plant 1, and plant 2 from the previous section.**



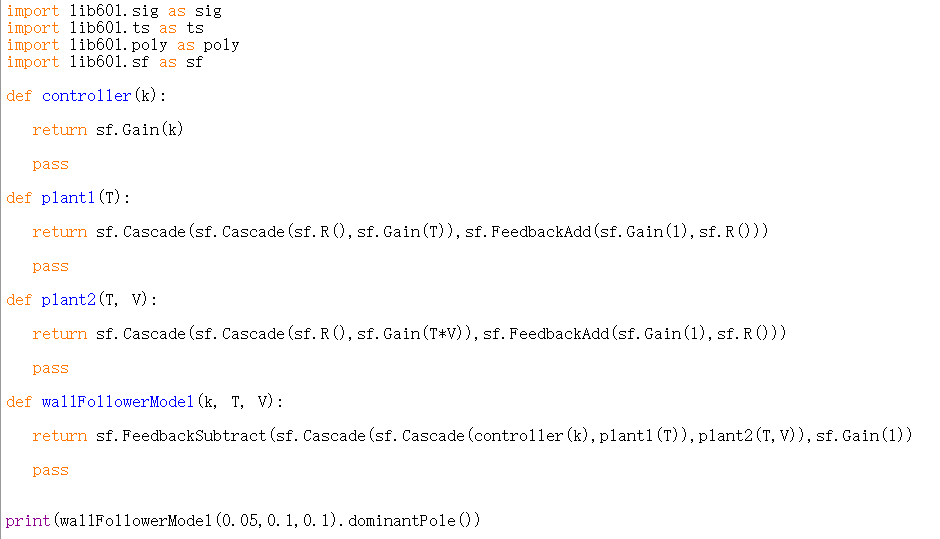
**Step8**

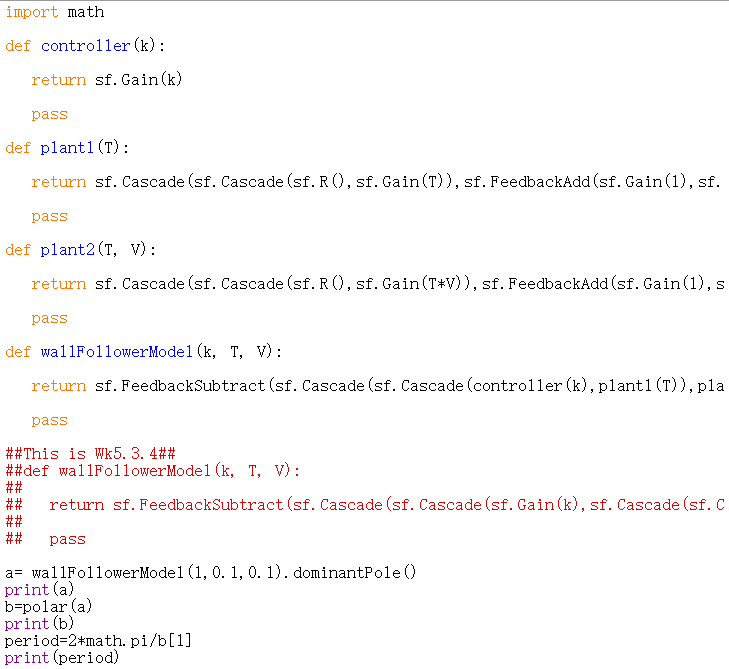


**Step9**



My codes for wallFollowerModel.py are as follows.



*Wk5.3.4*

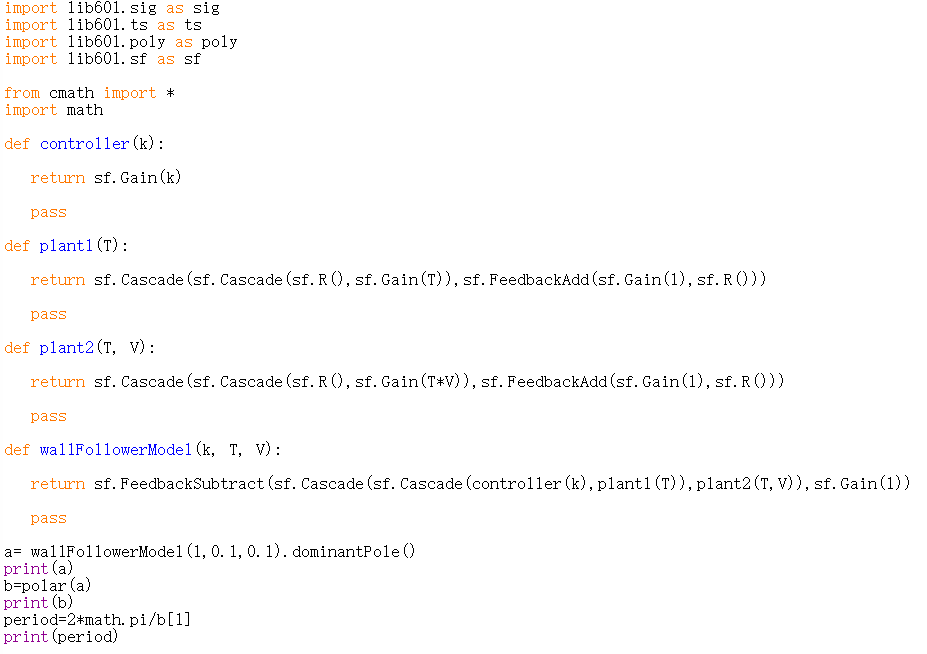
*If you want to run this code, you need to first comment the wallFollowerModel code above.*

**Step10**

*According to the formula:*

We get:

*So, we use the following code to do this:*



|  |  |
| --- | --- |
| k | period |
| 0.05 | 888.591 |
| 0.5 | 281.039 |
| 5 | 89.006 |
| 50 | 28.561 |
| 75 | 23.505 |
| 100 | 20.515 |
| 110 | 19.620 |

***Checkoff 3***

*Wk5.3.5***Show your results to a staff member. Explain similarities and differences between your results in this part and your results in Checkoff 1.**

*Compared with Checkoff1, we find that when the value of k increases, the period of both parts decreases.*

*In fact, neither of them will eventually converge to the expected value and oscillate around the expected value. When we simulate a physical system, even if the behavior of the system is found to be oscillating near the expected value by theoretical calculation, the behavior of the system can still be approximated to be convergent within the allowable range of errors.*

**Summary**

*1. When using dominantPole to find the poles of system functions, the final result obtained is the standard form of complex numbers. But what we want more often is an exponential form. Therefore, in designlab.work.py, we introduced cmath and math from the python library to make it easier to convert normal shapes to polar coordinates using the polar function. Meanwhile, to further find the period of the function, we use period=2\*math.pi/Ω to solve it.*

*2. When we analyze an actual system, the factors to consider are often multifaceted. Even if suitable conditions are found during theoretical analysis, the actual situation still needs to be considered. In step 10 we found that when the value of k increases, the oscillation period of the system behavior is greatly shortened. So, does it mean that we can achieve the effect of convergence of system behavior by increasing k and decreasing the period indefinitely? The answer, of course, is no. When we increase k, although period decreases, the rotational speed of the system behavior increases a lot. When the speed is too large, the safety of the system is difficult to guarantee.*

*At the same time, even if we do not have an ideal solution in theoretical analysis, as long as it is within the allowable range of errors, we can approximate that the experimental results meet the needs. For example, when k>20 in Checkoff1, we have actually been able to approximate the system behavior as convergence, and the next theoretical analysis is more inclined to improve the accuracy of the experiment.*

*3. In checkoff1 we did not consider the case of k<0 because the expected value is always the distance from the right wall. When the input value (that is, the actual distance from the right) is greater than the expected value, it should turn right, and if it is less than the expected value, turn left. If k is negative, the direction of the system behavsior is changed.*