Preliminary Information

Check Yourself 1.

$$\alpha_h[n] = k_m i_m[n]$$

$$v_b[n] = k_b \omega_h[n]$$

$$i_m[n] = \frac{(v_c[n] - v_b[n])}{r_m}$$

$$v_s[n] = k_s(\theta_l[n] - \theta_h[n]) = k_s e[n]$$

$$v_c[n] = k_c v_s[n]$$

Through the above five models, it can be concluded that:

*Unit of Km: rad/([sec] ^2*A)*

*Unit of Kb: (V*sec)/rad*

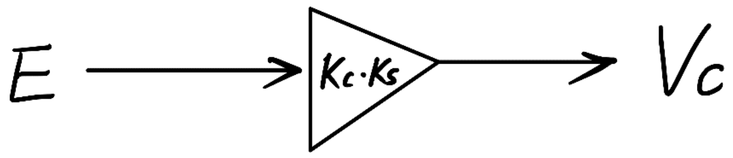
The unit of Ks is V/rad

Units of Kc: Kc is a dimensionless quantity

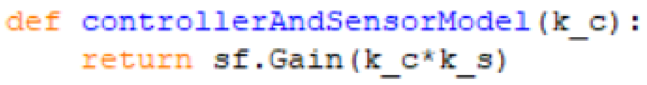
Building the Model

Step1: *modeling the Sensor and Controller*

1. *The system function for this control/sensor system: Kc*Ks*
2. *The block diagram:*



3. *Method:'controllerAndSensorModel'*



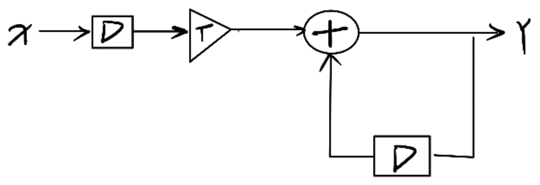
```
def controllerAndSensorModel(k_c):
    return sf.Gain(k_c*k_s)
```

***Step2:**modeling the plant*

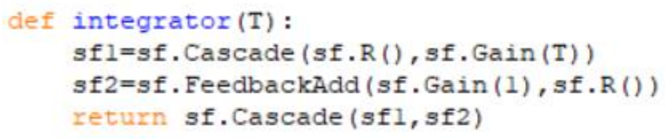
1. Intergrator:

(1):the system function for the integrator: $\frac{R*T}{1-R}$

(2):the block diagram:



(3):method:'integrator'

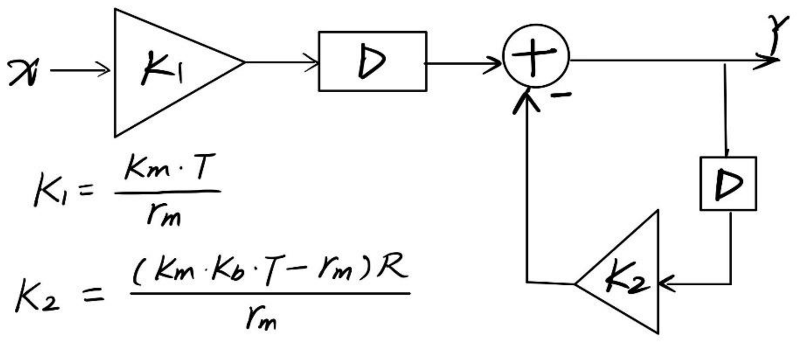


```
def integrator(T):
    sf1=sf.Cascade(sf.R(),sf.Gain(T))
    sf2=sf.FeedbackAdd(sf.Gain(1),sf.R())
    return sf.Cascade(sf1,sf2)
```

2. Motor :

(1):the system function for the integrator: $\frac{Km*T*R}{(Km*Kb*T-rm)*R+rm}$

(2):the block diagram:


x
K1
D
+
-
Y
D
K2
$K_1 = \frac{K_m \cdot T}{r_m}$
$K_2 = \frac{(K_m \cdot K_b \cdot T - r_m)R}{r_m}$

(3):method:'motormodel'

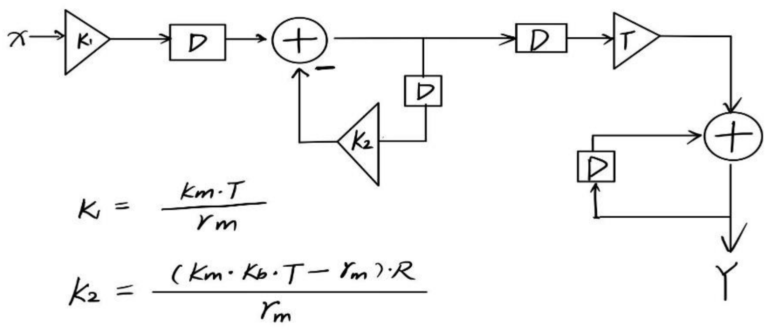
```
def motorModel(T):
    gain1=k_m*T/r_m
    gain2=(k_m*k_b*T-r_m)/r_m
    sf1=sf.Cascade(sf.R(),sf.Gain(gain1))
    sf2=sf.FeedbackSubtract(sf.Gain(1),sf.Cascade(sf.R(),sf.Gain(gain2)))
    return sf.Cascade(sf1,sf2)
```

3. The Combined Plant

(1)the system function for the entire plant:

$$\frac{Km * T^2 * R^2}{(rm - Km * Kb * T) * R^2 + (Km * Kb * T - 2 * rm) * R + rm}$$

(2)the block diagram☹


x
K₁
D
D
D
T
K₂
D
Y
$K_1 = \frac{Km \cdot T}{rm}$
$K_2 = \frac{(Km \cdot Kb \cdot T - rm) \cdot R}{rm}$

(3)method:'plant Model'

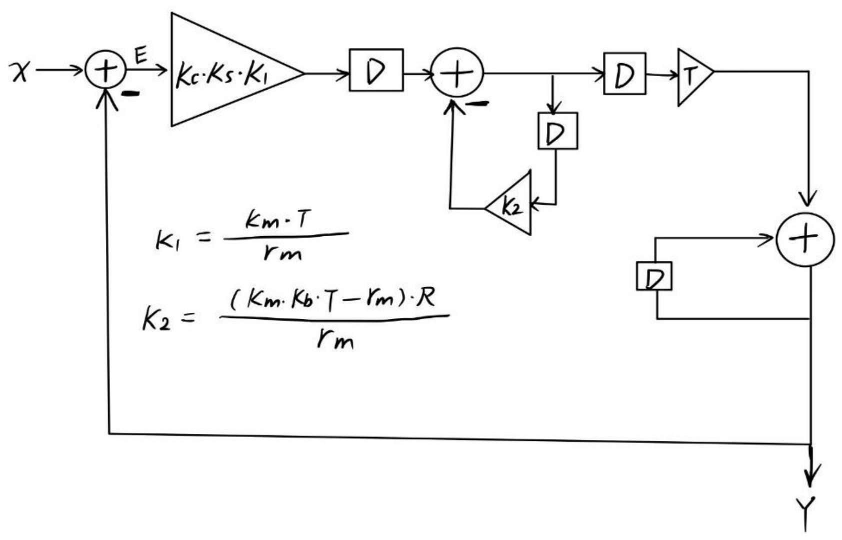
```
def plantModel(T):
    return sf.Cascade(motorModel(T),integrator(T))
```

4. Putting it Together

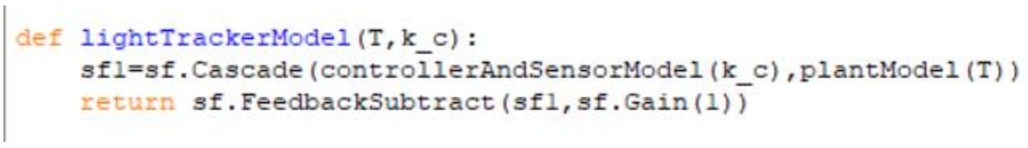
(1)the system function for the composite system:

$$\frac{Ks * Kc * Km * T^2 * R^2}{(Ks * Kc * Km * T^2 - Km * Kb * T + rm) * R^2 + (Km * Kb * T - 2 * rm) * R + rm}$$

(2)the block diagram:

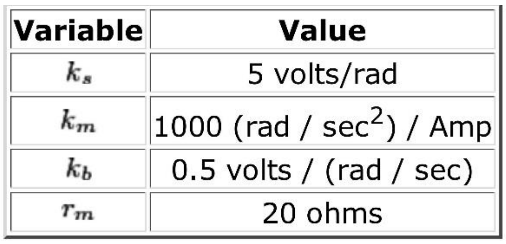

X
E
Kc·Ks·K1
D
D
D
T
K2
D
Y
$K_1 = \frac{K_m \cdot T}{r_m}$
$K_2 = \frac{(K_m \cdot K_b \cdot T - r_m) \cdot R}{r_m}$

(3)method:' lightTrackerModel'



```
def lightTrackerModel(T,k_c):
    sf1=sf.Cascade(controllerAndSensorModel(k_c),plantModel(T))
    return sf.FeedbackSubtract(sf1,sf.Gain(1))
```

(4)the pole of this system:



Variable	Value
k_s	5 volts/rad
k_m	1000 (rad / sec^2) / Amp
k_b	0.5 volts / (rad / sec)
r_m	20 ohms

$$P1=\frac{(2-25*T)+5*T*\sqrt{25-40*Kc}}{2}$$

$$P2=\frac{(2-25*T)+5*T*\sqrt{25-40*Kc}}{2}$$

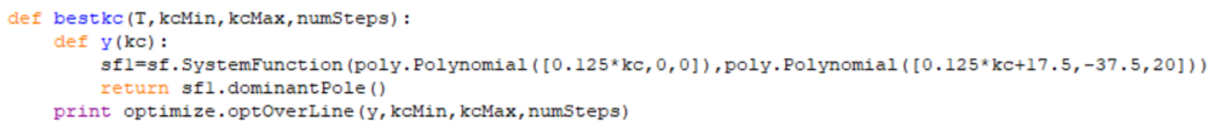
*Step3:*Analyzing the System

1. *find the value of* ***Best Gain—kc*** *:*

The speed of convergence of a system's response is improved by reducing the magnitude of the dominant pole. Therefore,to find the best gain(kc),we should construct a function f(kc) that computes the magnitude of the dominant pole,find the value of k that produces the minimum value of this funtion.

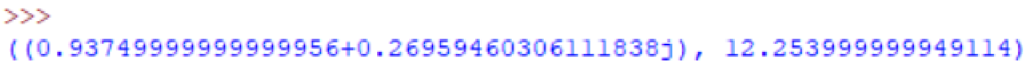
(1) By optimization Algorithm:

*The lib601 optimize module provides us with an python 'optimization' procedure called optimize.optOverLine. Now, we use it to find the best **kc**.*



```
def bestkc(T,kcMin,kcMax,numSteps):
    def y(kc):
        sf1=sf.SystemFunction(poly.Polynomial([0.125*kc,0,0]),poly.Polynomial([0.125*kc+17.5,-37.5,20]))
        return sf1.dominantPole()
    print optimize.optOverLine(y,kcMin,kcMax,numSteps)
```

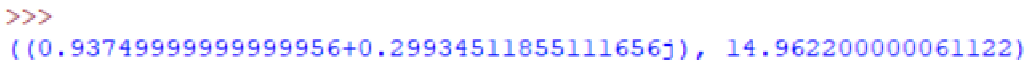
• *the first step: 令 kcmin=-100, kcmin=100, numSteps=100000*



```
>>>
((0.9374999999999956+0.2695946030611183 8j), 12.253999999949114)
```

(The first result in a tuple indicates the dominant pole with the smallest value in the search range, The second result in the tuple represents the best gain in the search range kc)

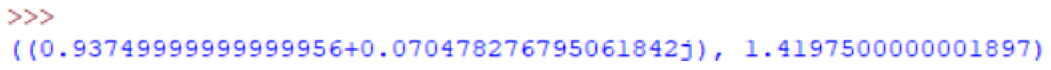
• *the second step :we try to expand the search interval (kcmin=-110, kcmax=110)*



```
>>>
((0.9374999999999956+0.2993451185511165 6j), 14.962200000061122)
```

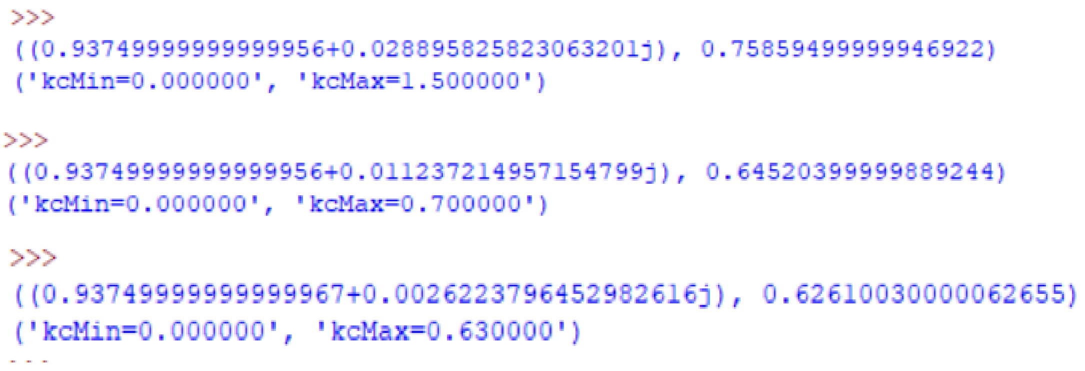
Instead, the size of the dominant pole increases, so the optimal gain k is not outside the interval (-100, 100).

• *the third step: We narrow the search interval according to the size of the optimal gain kc determined in the first step(kcmin=0,kcmax=15)*



```
>>>
((0.9374999999999956+0.0704782767950618 42j), 1.4197500000001897)
```

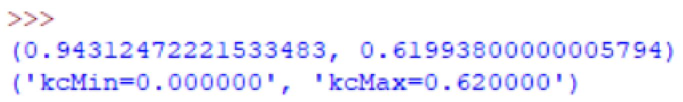
• *Repeat the above steps to further narrow the search interval until the best kc is found:*



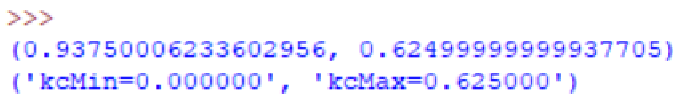
```
>>>
((0.9374999999999956+0.028895825823063201j), 0.7585949999946922)
('kcMin=0.000000', 'kcMax=1.500000')

>>>
((0.9374999999999956+0.011237214957154799j), 0.6452039999889244)
('kcMin=0.000000', 'kcMax=0.700000')

>>>
((0.9374999999999967+0.0026223796452982616j), 0.6261003000062655)
('kcMin=0.000000', 'kcMax=0.630000')
...
```



```
>>>
(0.94312472221533483, 0.61993800000005794)
('kcMin=0.000000', 'kcMax=0.620000')
```



```
>>>
(0.93750006233602956, 0.62499999999937705)
('kcMin=0.000000', 'kcMax=0.625000')
```

The value of kc when the size of the dominant pole reaches the minimum is selected by using the interval clamping method.

Because the sampling points of the optimization algorithm are limited, we cannot accurately obtain the best gain kc.

Through the three sets of data: kcMax=0.63,kcMax=0.62 and kcMax=0.625, we can roughly determine the range of the best gain kc: 0.62-0.63. We approximately take the midpoint of the interval ***0.625*** *as the best gain kc, and the size of the pole is* ***0.9375.***

(2)By discussing and analyzing the expression of the pole:

In step2, we derive the poles of the system:

$P1=\frac{(2-25*T)+5*T*\sqrt{25-40*Kc}}{2}$, $P2=\frac{(2-25*T)+5*T*\sqrt{25-40*Kc}}{2}$

Assume that T=0.005:

$P1=0.9375+0.0125\sqrt{25-40*kc}$, $P2=0.9375\text{-}0.0125\sqrt{25-40*kc}$

When kc<=0.625, poles P1 and P2 are real numbers:

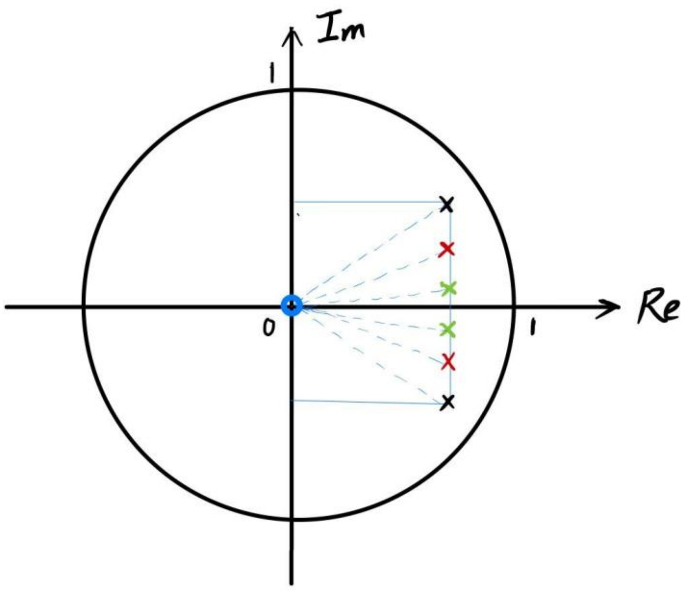
- *When $0 < kc <= 0.625$,|P1|< 1, |P2 |< 1, at this point, the system function is monotone convergence. The system is a stable system. Since the convergence rate of the system is determined by the dominant pole, in the discrete system the convergence rate is determined by the pole whose amplitude is closer to 1. Therefore, to get a min (Max {|P1 | and | P2 |}) is the way to make the system converge fastest . Obviously, when |P1 | = | P2 | = 0.9375, Max {|P1 | , | P2 |} is minimized. The best gain kc=0.625.*

- *When kc < 0: | P1| > 1, the system function diverges,and the system is an unstable system.*

•when kc = 0 ,|P1| = 1, |P2|< 1, the system function is monotone convergence, convergence value is not zero.

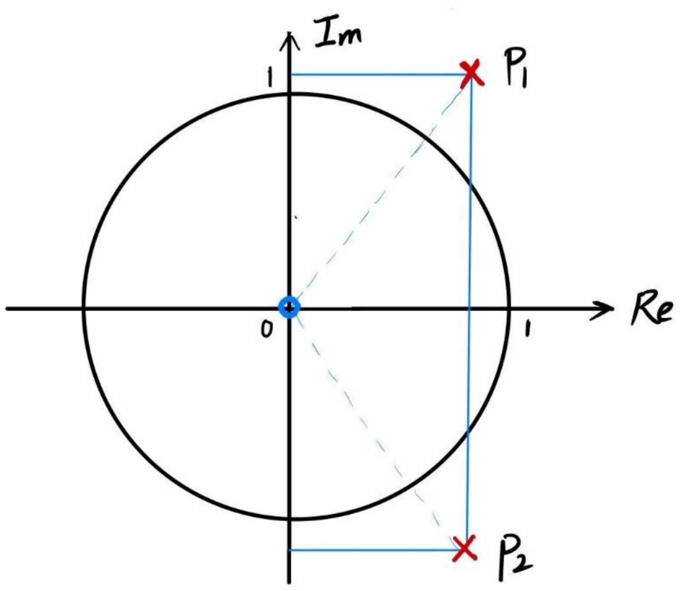
When kc>0.625, poles P1 and P2 are complex, and P1 and P2 are a pair of conjugate poles. Therefore, the moduli of P1 and P2 are always equal, and there is no concept of dominant poles in this case. P1 and P2 both determine the convergence speed of the system (occupying the same position).。

•When 0.625 <kc < 20: |P1 | =|P2 | < 1. Denote P1 and P2 in the complex plane:



In this case, the system。If the convergence rate of the system is to be accelerated, |P1| and |P2| should be reduced. As shown in the figure above, the real parts of P1 and P2 are 0.9375. After decreasing |P1| and |P2|, The poles will keep approaching the X-axis, and the limiting case is that the imaginary parts of P1 and P2 are both 0. At this time, |P1|and |P2| are minimized, And P1 and P2 become real poles. Same as the case estimated by the optimization algorithm, the best gain（kc）=0.625,|P1|=|P2|=0.9375。

•When kc>20 时, |P1|=|P2|>1:



The system function diverges and the system is unstable.

•When kc=20， |P1|=|P2|=1, So the output of the system will always be a constant.

Through the above theoretical analysis, we obtain the best gain kc=0.625 that makes the convergence rate of the system reach the fastest, which is the same as the best gain kc estimated by using the optimization algorithm.

2. ***regions***:

In the previous analysis, we have discussed the convergence relationship between the range of kc and the system:

0<kc<=0.625: the system monotonically convergent

0.625<kc<20: the system oscillatory and convergent

Kc=20: the lowest positive value of kc for which the system is unstable

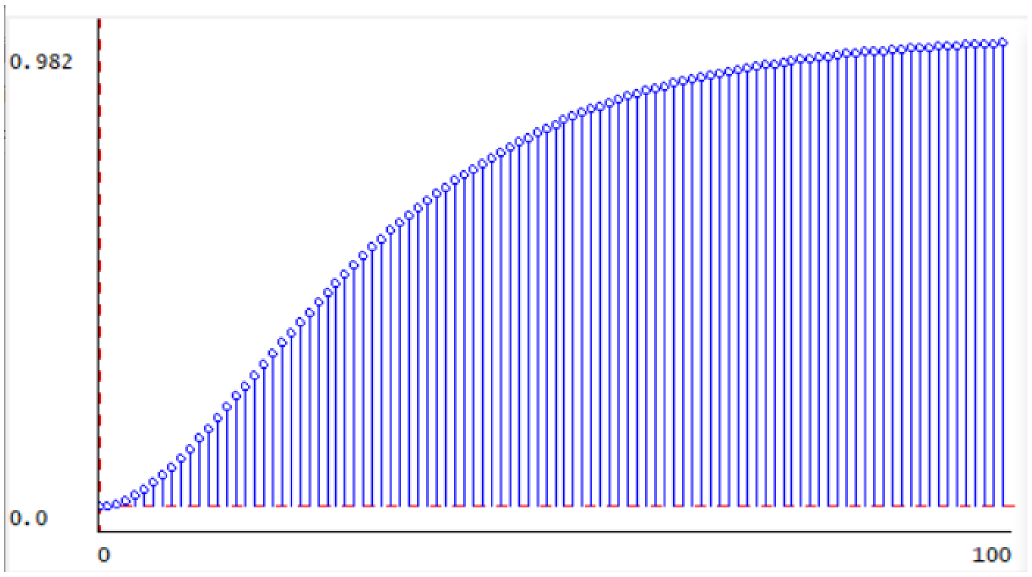
3. ***Plots:***

Select three specific values of kc, which correspond to monotonic convergence, oscillatory convergence and divergence of the system, respectively.

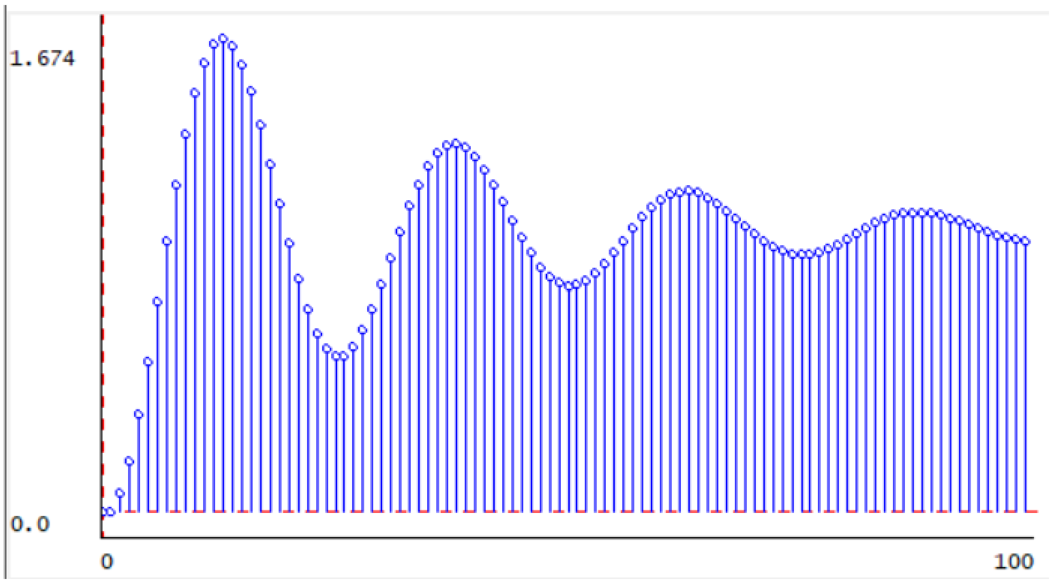
Plot the response of the light tracker to a unit step input:

```
def plotOutput(sfModel):
    """Plot the output of the given SF, with a unit-step signal as input"""
    smModel = sfModel.differenceEquation().stateMachine()
    outSig = ts.TransducedSignal(sig.StepSignal(), smModel)
    outSig.plot()
```

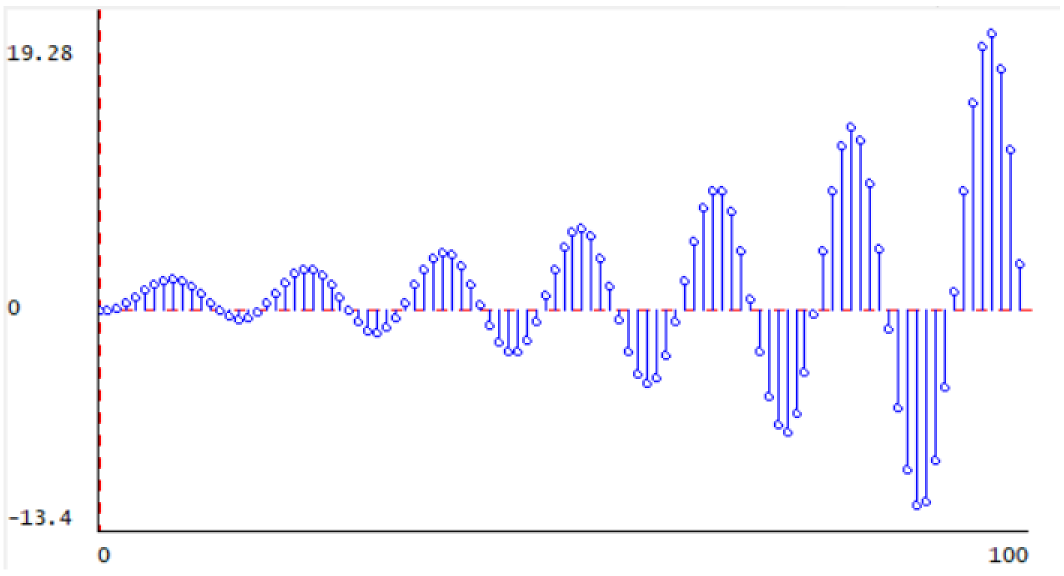
(1) Kc=0.6:



(2) kc=10:



(3) kc=30:

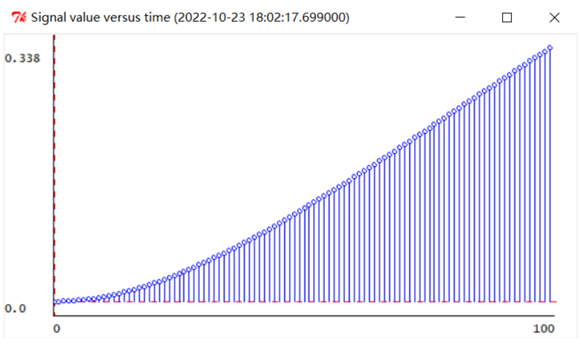


The convergence of the system is consistent with the result of theoretical analysis.

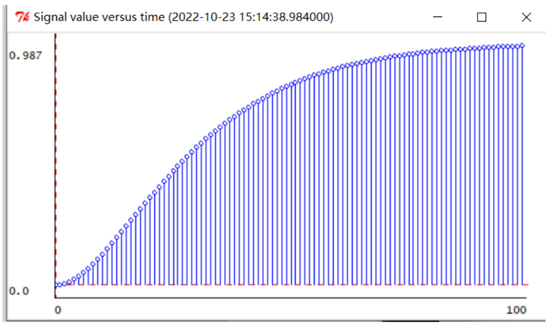
4. **Effect of T**

(1) Fix kc, increase or decrease T.Plot the response of the light tracker to a unit step input.

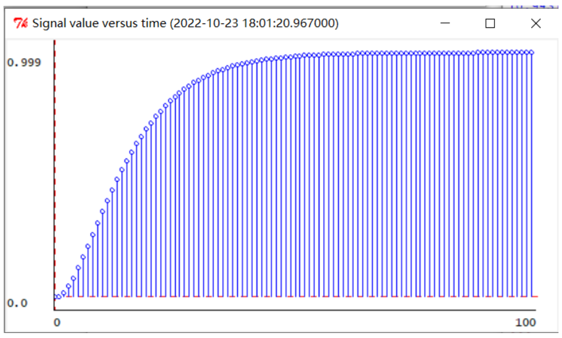
•kc=0.6



(T=0.001)



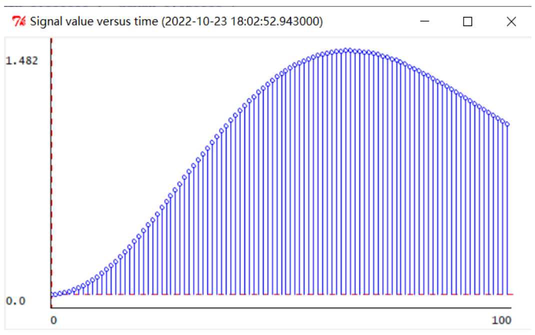
(T=0.005)

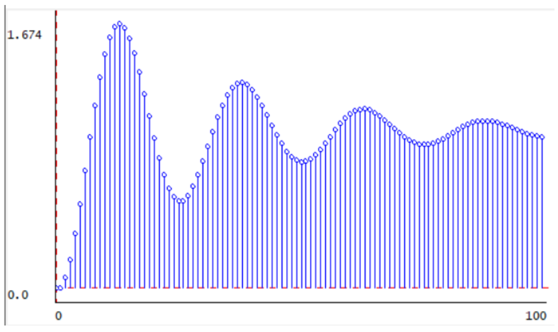


(T=0.01)

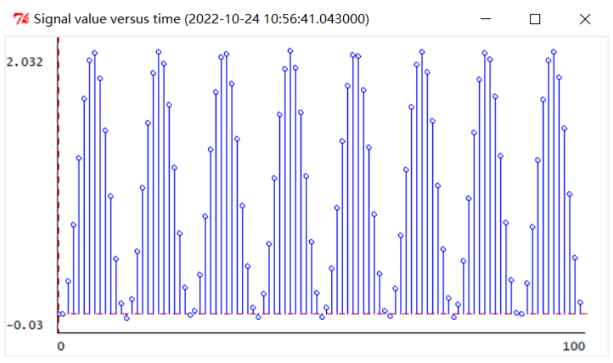
When kc=0.6, if we decrease T and make T=0.001, The convergence rate of system functions slows down, and the magnitude of dominant pole is increased.If increasing T and make T=0.01, we get the opposite result.

•kc=10





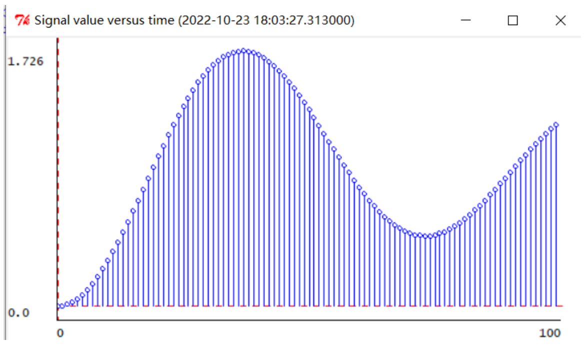
(T=0.001)　　(T=0.005)



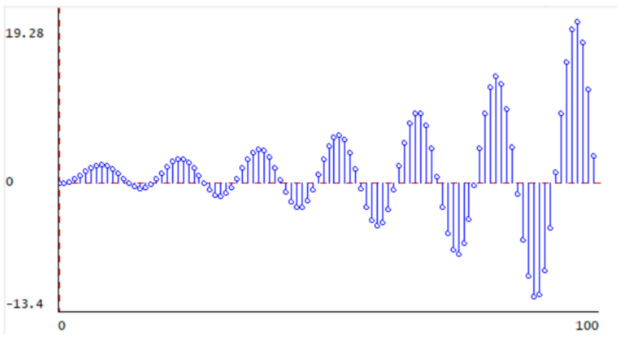
(T=0.01)

When kc=10, if decreasing T and make T=0.001, The convergence rate of system function is accelerated, the magnitude of dominant pole is decrease.If we increase T and make T=0.01, The unit step response diverges.

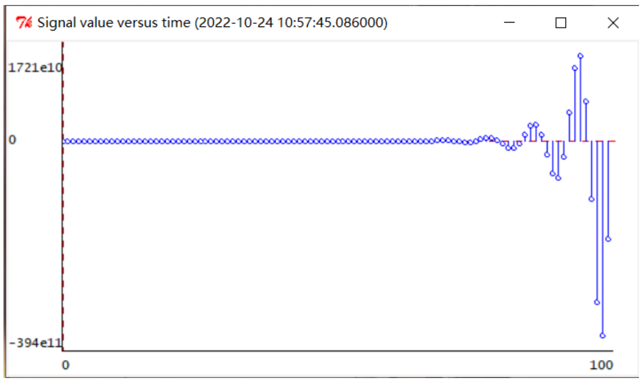
•kc=30



(T=0.001)



(T=0.005)



(T=0.01)

When kc=30, if decreasing T and make T=0.001, The system function changes from divergence to convergence.If we make T=0.01,The system function still diverges and the amplitude increases.

(2)Analyzing the expression of the poles:

When kc is a certain value, the poles of the system function are expressed as functions of T:

$$P1=\frac{(5\sqrt{25-40*kc}-25)*T+2}{2}\ ,\ P2=\frac{-(5\sqrt{25-40*kc}+25)*T+2}{2}$$

- When kc<0, |P1|>1, The system function diverges.
- When 0<kc<0.625, P1 and P2 are real numbers, And along with the increase of T, | P1| and | P2 | decrease, and the test results are consistent with the results in (1)
- When kc>0.625:Poles are complex numbers. $|P1|=\frac{\sqrt{250(5-4*kc)*T^2-100*T+4}}{2}$

Inside the square root of the numerator is a quadratic function of T :

$$250(5-4*kc)*T^2-100*T+4$$

Let's talk about the coefficients 'A ' of T to the second power:

•When $0.625 < kc < 1.25$, A>0, Axis of symmetry is $\frac{1}{5\ (5-4*\mathrm{kc})}$

In the left axis of symmetry, |P1| decreases with the increase of T, the convergence speed of the corresponding system function (or system from divergence to convergence).At the right side of symmetric axis, |P1| increases with the increase of T . The corresponding system function convergence speed decrease (or system from convergence and divergence).

•When $kc > 1.25$, A<0, The axis of symmetry remains the same

In the left axis of symmetry, |P1| increases with the increase of T, the convergence rate of the corresponding system function decline (or system from convergence to divergent).At the right side of symmetric axis, |P1| decreases with the increase of T. The corresponding system function to speed up the convergence speed (or system from divergence to convergence).

Through the above classification discussion, the specific kc can be used to accurately calculate the relationship between the size of the pole and T, and judge the change of convergence of the system function when T changes.

Summary

The study of Homework2 builds system functions by modeling a real physical model, so that we can explore the relationship between system function poles and system stability.

First, we should understand the concept of the dominant pole: the convergence rate of the system is determined by the magnitude of the dominant pole.

In continuous system, we carried out on the dominant pole the following definition: refers to the dominant pole in system all closed-loop poles, closest to the imaginary axis and the surrounding without the poles of the closed-loop zero and the rest of the pole from the imaginary axis, so apart from pole of the imaginary axis recent response component plays a leading role in the system response, the closed-loop poles is called dominant pole. The light tracker model constructed in Homework2 is a discrete system, and the dominant pole is the pole farther from the origin. The dominant pole can be either a real number or a complex number. The optimal gain kc sought in the experiment is actually to find a kc such that the size of the dominant pole is reduced as much as possible (note that once the size of the dominant pole is greater than 1, the system function will diverge and the system will become an unstable system).

Difficulties encountered in the experiment:

In the experiment, an optimization algorithm is used to calculate the minimum value of the function. However, when the search interval is large, it is difficult to find the results consistent with the theoretical value. This is because the number of sample points searched is limited, and the computational power of cpu is limited, so we obviously cannot increase the number of sample points searched indefinitely. At the same time, the optimization algorithm is difficult to determine the best value of a small search interval, can only use the idea of squeeze to narrow the interval. Therefore, it is necessary to use the results of theoretical analysis to assist program debugging (we also focus on the theoretical analysis process in the report).