

Hard Disk Drive Read Header Controller Design Project

*EE 141 – Principles of Feedback Control
Spring 2015
University of California, Los Angeles*

Baixiao Huang 504313981
Weiqian Xu 404297854
Xingjian Yan 104304403

05/21/2015

1.0 OBJECTIVES

The objective of this project is to design a control system for hard disk drive read head. Knowledge of feedback control theories and skills of using Matlab are crucial in designing. The system contains a controller, a motor coil, a reader arm and a sensor forming a closed loop system. In this project, we are going to analyze the relationship between system performance and different compensators or sensors.

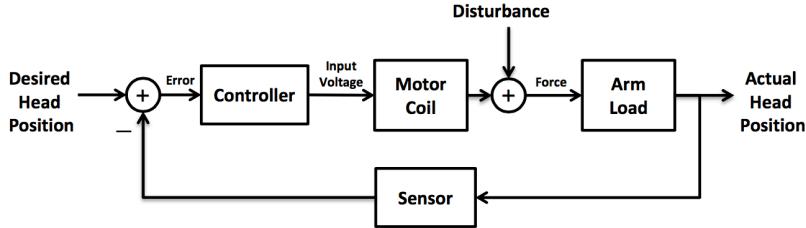


Figure 1

2.0 TASKS

2.1 Task 1

For this part, we are provided with the physical model relating the input torque u to the header position y with the inertia of arm and head J and friction b .

$$J\ddot{y} + b\dot{y} = u(t) \quad (1)$$

Simply apply Laplace transform to get the transfer function:

$$G_2(s) = \frac{Y(s)}{U(s)} = \frac{1}{Js^2 + bs} \quad (2)$$

2.2 Task 2

The transfer function of the motor coil that relates the input voltage to the output torque is related by:

$$G_1(s) = \frac{U(s)}{V(s)} = \frac{K_m}{Ls + R} \quad (3)$$

In this task, we are going to obtain the open-loop transfer function of the cascaded HDD head reader assembly and then use Matlab to generate the plot of the system's step response.

$$G_1(s)G_2(s) = \frac{K_m}{(Ls + R)(Js^2 + bs)} = \frac{K_m}{LJs^3 + Lbs^2 + RJs^2 + Rbs} \quad (4)$$

The Matlab code with the constants plugged in and the plot obtained are presented below:

```
s = tf('s');
J = 1;
b = 20;
R = 1;
L = 0.001;
Km = 5;
G1 = Km/(L*s+R)
G2 = 1/(J*s^2+b*s)
openTF = G1*G2
t=[0:0.005:0.5];
y = step(openTF,t);
plot(t,y);
```

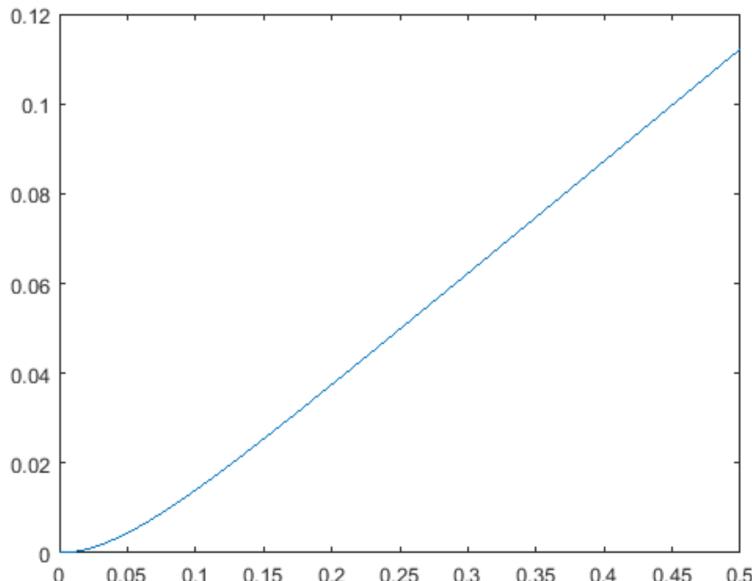


Figure 2

As we can see, if a constant voltage is applied, then the read head will move with a constant speed. However, at the beginning when the voltage is applied, there is a curvature indicating the read head is accelerating.

2.3 Task 3 – Proportional Compensator

In this part, we are going to make the closed-loop system satisfies several transient response performance specifications: Percent overshoot less than 5%; Settling time (2% deviation) less than 250ms; Maximum value of response to a unit step disturbance less than 5×10^{-3} . Assume the feedback sensor's transfer function is $H(s) = 1$ and the candidate controller is $F(s) = K_a$.

Part A

We are going to determine and evaluate the responses of the closed-loop system to the unit step reference input without disturbance. If the proportional compensator applied, the transfer function is given by:

$$K_a G_1(s) G_2(s) \quad (5)$$

Then the closed-loop transfer function is given by:

$$\frac{K_a G_1(s) G_2(s)}{1 + K_a G_1(s) G_2(s)} \quad (6)$$

After plugging in (2) and (3), the equation becomes:

$$\frac{KaKm}{LJs^3 + (Lb + RJ)s^2 + Rbs + KaKm} \quad (7)$$

Try $Ka = 50$, the Matlab code (continuous after the previous code) and plot are shown below:

```
Ka = 50;
t=[0:0.005:1.5];
ProportionalTF =(Ka*openTF)/(1+Ka*openTF)
y = step(ProportionalTF, t);
plot(t,y);
```

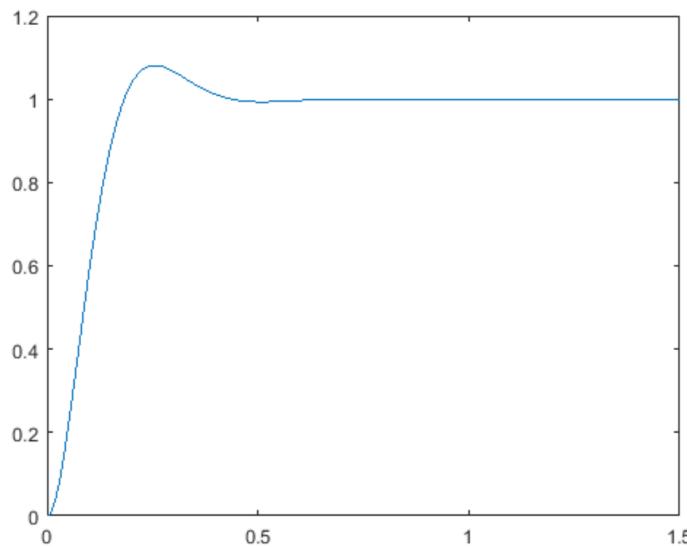


Figure 3

If $K_a = 400$, the Matlab code and plot are shown below:

```
Ka = 400;
ProportionalTF = (Ka*openTF)/(1+Ka*openTF)
y = step(ProportionalTF, t);
plot(t,y);
```

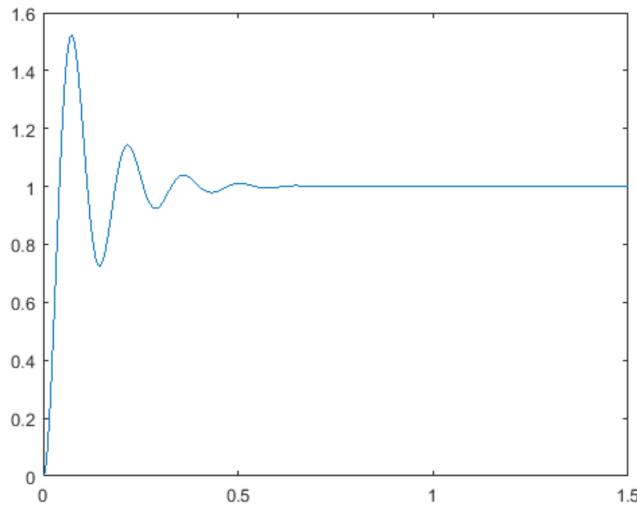


Figure 4

It is clear that the system overshoot when $K_a = 50$ is much lesser than that when $K_a = 400$. If the feedback amplification is too big, the system will be too sensitive.

Part B

In this part, we are going to determine and evaluate the responses of the closed-loop system to the unit step disturbance, assuming zero reference input. The transfer function for disturbance is:

$$T_\omega = \frac{G_2(s)}{1+K_a G_1(s)} \quad (8)$$

As above, we first plot the response with $K_a = 50$.

```
Ka = 50;
Tw = G2/(1+Ka*G1*G2)
y = step(Tw, t);
plot(t,y);
```

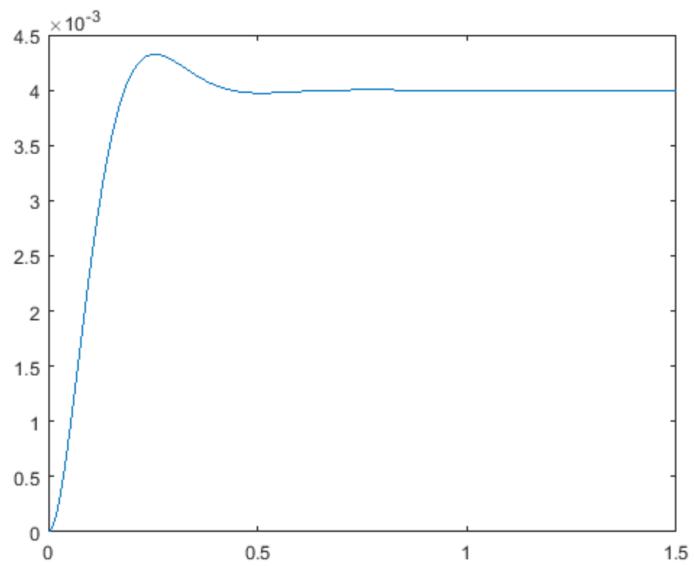


Figure 5

Similarly, we then try $K_a = 400$.

```
Ka = 400;
Tw = G2/(1+Ka*G1*G2)
y = step(Tw, t);
plot(t,y);
```

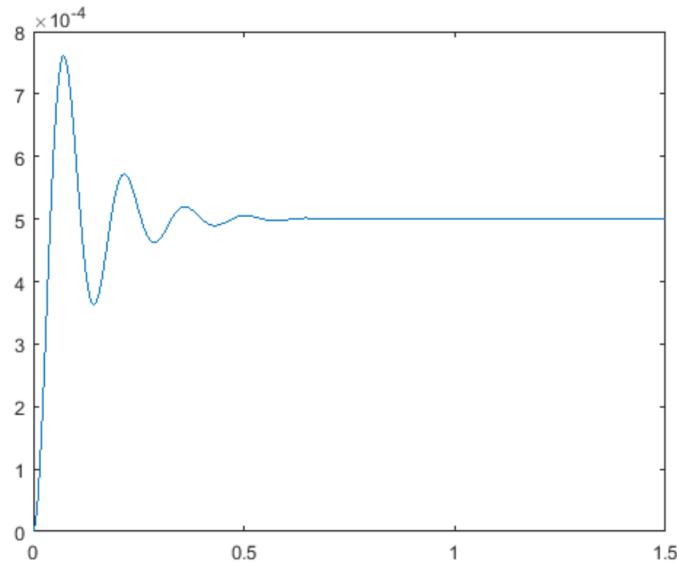


Figure 6

Same as Part A, $Ka = 50$ has less overshoot than $Ka = 400$.

Part C

In this part, the objective is to find a value for K_a that satisfies the performance specifications above. We can plot overshoot percentage and settling time as a function of K_a , and analyze the optimal K_a for the system.

```
overshoot = [];
settlingTime = [];
maxError = [];
n=[1:100];
for Ka=n
    ProportionalTF = (Ka*openTF)/(1+Ka*openTF);
    y = step(ProportionalTF, t);
    info = stepinfo(y, t, 'SettlingTimeThreshold', 0.02);
    overshoot = [overshoot, info.Overshoot];
    settlingTime = [settlingTime info.SettlingTime];
    Tw = G2/(1+Ka*G1*G2);
    y = step(Tw, t);
    maxError = [maxError max(y)];
end

subplot(3,1,1);
plot(n,overshoot); xlabel('Ka'); ylabel('Overshoot Percentage');

subplot(3,1,2);
plot(n,settlingTime); xlabel('Ka'); ylabel('Settling Time');

subplot(3,1,3);
plot(n,maxError); xlabel('Ka'); ylabel('Maximum Error of Disturbance');
```

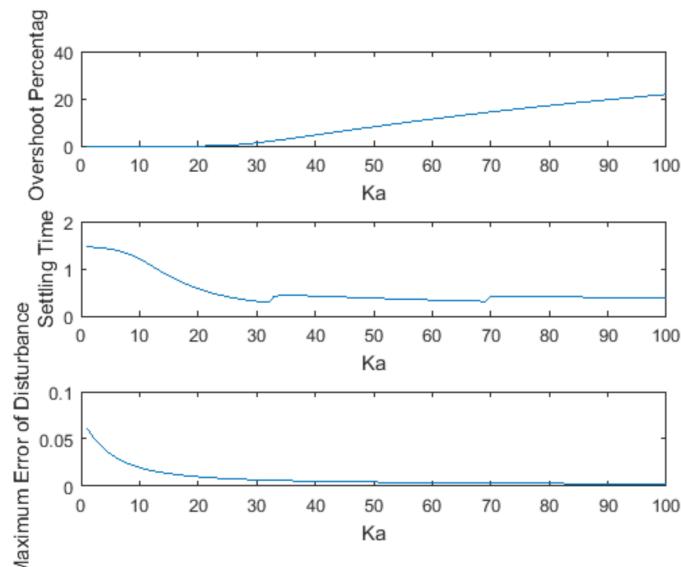


Figure 7

For overshoot less than 5%, K_a is required to be equal or less than 41. The value of K_a that satisfies settling time requirement is more than 100 and is not in the graph. For disturbance less than 0.005, K_a is required to be equal or bigger than 41. As a result, the three requirements cannot be satisfied at the same time.

2.4 Task 4 – Positional and Velocity Sensor

In this part, we substitute the sensor with $H(s) = 1 + K_H s$ that measures the position as well as the velocity of the head reader. Then we need to find K_a and K_H such that the system is stable and satisfies the performance specifications mentioned above. The closed loop transfer function is:

$$\frac{K_a G_1(s) G_2(s)}{1 + K_a H(s) G_1(s) G_2(s)} \quad (9)$$

The fundamental idea for finding the solution is that since both K_a and K_h are varying, we can plot a 3-D graph in which x-axis is K_a , y-axis is K_h , and z-axis is the property under examination including settling time, overshoot and disturbance. Then we can find out all of the combinations of K_a and K_h that satisfy all performance requirements.

```
[KaRange, KhRange] = meshgrid(55:65, 0:0.01:0.1);
overshootMatrix = [];
settlingTimeMatrix = [];
candidatePairs = [];
t=[0:0.05:0.8];
G12= G1*G2;
for Kh = KhRange(:,1)'
    %overshootArr=[];
    %settlingTimeArr=[];
    for Ka = KaRange(1,:)
        CLTF = (Ka*G12)/(1+Ka*(1+Kh*s)*G12);
        y = step(CLTF, t);
        info = stepinfo(y, t, 'SettlingTimeThreshold', 0.02);
        %overshootArr = [overshootArr, info.Overshoot];
        %settlingTimeArr = [settlingTimeArr, info.SettlingTime];
        Tw = G2/(1+Ka*(1+Kh*s)*G12);
        y = step(Tw,t);
        maxDisturbance = max(y);
        if(info.Overshoot <= 5 & info.SettlingTime < 0.25 & y<0.005)
            candidatePairs = [candidatePairs; Ka, Kh];
        end
    end
    %overshootMatrix = [overshootMatrix; overshootArr];
    %settlingTimeMatrix = [settlingTimeMatrix; settlingTimeArr];
end
```

```
%mesh(KaRange, KhRange, settlingTimeMatrix);

%mesh(KaRange, KhRange, overshootMatrix);
candidatePairs

candidatePairs =
56.0000    0.0300
57.0000    0.0300
58.0000    0.0300
59.0000    0.0300
60.0000    0.0300
61.0000    0.0300
62.0000    0.0300
63.0000    0.0300
64.0000    0.0300
65.0000    0.0300
```

There are a bunch of valid pairs of K_a and K_h that satisfies the design. We can select $K_a = 60$ and $K_h = 0.03$ to check the result.

```
Ka = 60; Kh = 0.03;
CLTF = (Ka*G12)/(1+Ka*(1+Kh*s)*G12);
y = step(CLTF, t);
figure(6);
plot(t,y);title('System Step Response');
info = stepinfo(y, t, 'SettlingTimeThreshold', 0.02)

Tw = G2/(1+Ka*(1+Kh*s)*G12);
y = step(Tw,t);
figure(7);
plot(t,y);title('System Step Disturbance Response');
maxDisturbance = max(y)
info =
RiseTime: 0.1621
SettlingTime: 0.2380
SettlingMin: 0.9401
SettlingMax: 1.0084
Overshoot: 0.8390
Undershoot: 0
Peak: 1.0084
PeakTime: 0.3500

maxDisturbance = 0.0034
```

The graphs of responses are presented below.

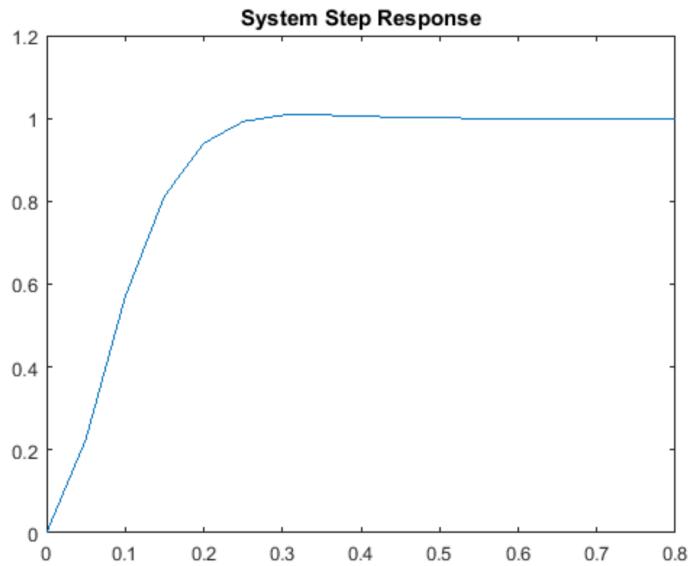


Figure 8

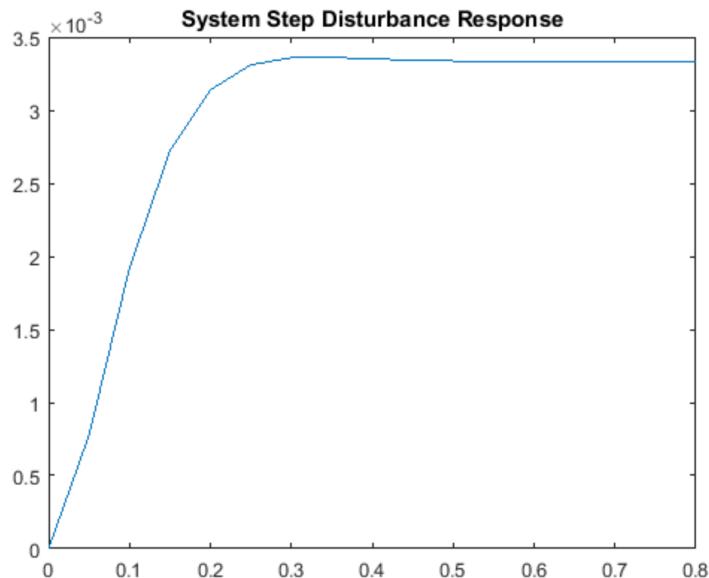


Figure 9

We can then conclude that the pairs of results listed above are valid for the system to meet all requirements.

2.5 Task 5 – PID Compensator

In this part, we revert the sensor back to $H(s) = 1$ but substitute the controller with $F(s) = K_1 + \frac{K_2}{s} + K_3 s$. Since the arm inherently has an integrating term, we assume $K_2 = 0$. The closed-loop transfer function then becomes:

$$\frac{(K_1 + K_3 s)G_1(s)G_2(s)}{1 + (K_1 + K_3 s)G_1(s)G_2(s)} \quad (10)$$

We can apply the same procedure to generate 3-D graph of constraints.

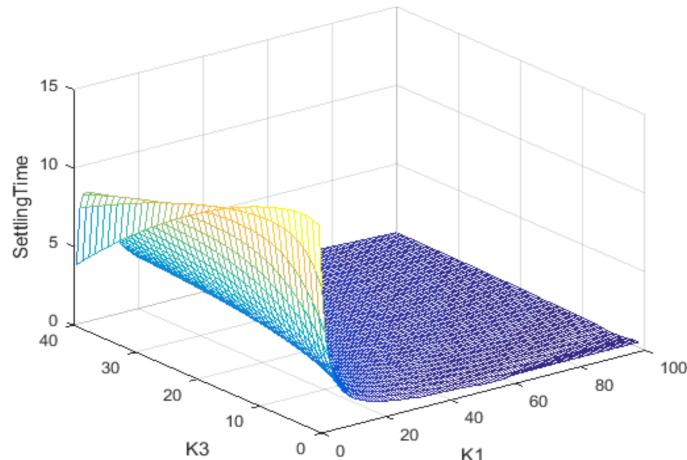


Figure 10 – Settling time

When K_1 is around 100 and K_3 is around 5 to 10, the settling time requirement can be satisfied. With the range of values, we are able to further derive detailed solution. If we take a closer look at the range of values that satisfy the constraints, the graph becomes the one shown by Figure 11.

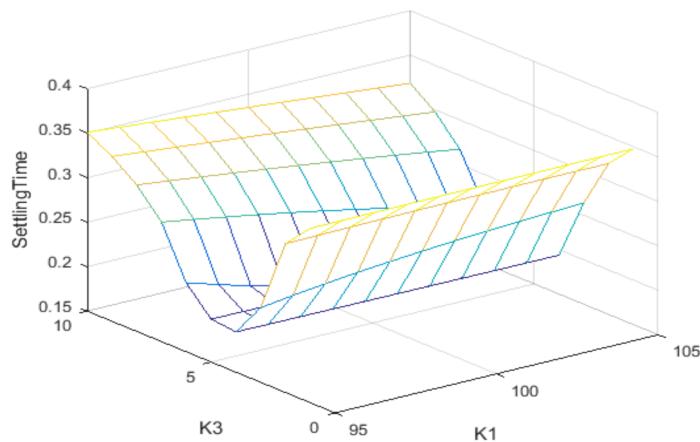


Figure 11 – Settling time

As shown in Figure 11, there are couples of K_1 and K_3 pairs that meet the settling time requirement. If we exam $K_1 = 100$ and $K_3 = 5$ for example, the settling time will under 250ms. The next step is to evaluate the overshoot of the system.

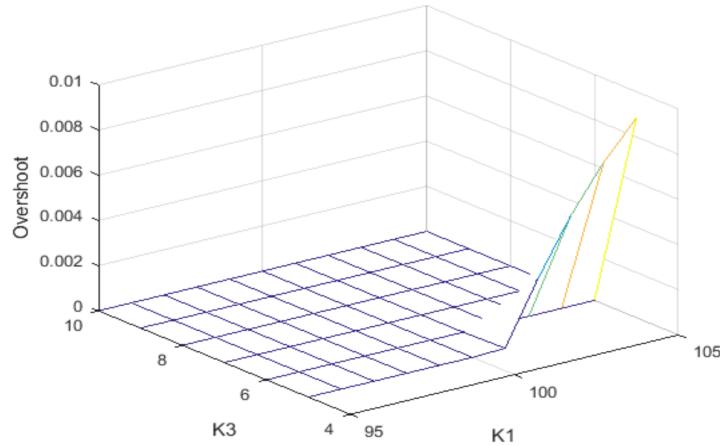


Figure 12 – Overshoot

As we can see, for $K_1 = 100$ and $K_3 = 5$, there is no apparent overshoot. We can then examine the disturbance of this range.

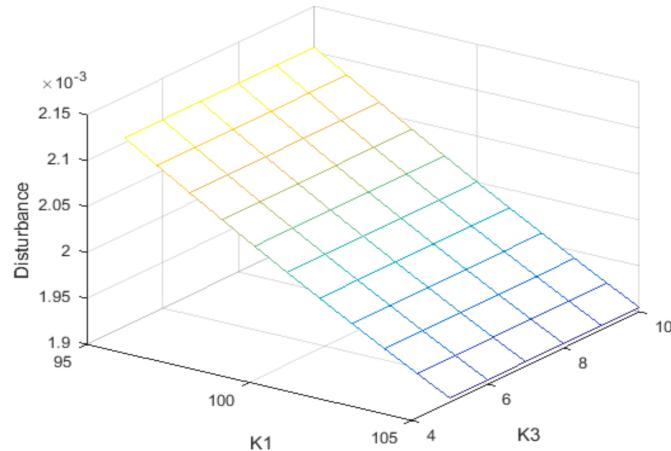


Figure 13 – Disturbance

The disturbance of $K_1 = 95$ and $K_3 = 100$ meets the requirement. The code for generates graphs above are presented below:

```
[K1Range K3Range] = meshgrid(95:105, 5:10);
overshootMatrix = [];
settlingTimeMatrix = [];
disturbanceMatrix= [];
candidatePairs = [];
```

```

t=[0:0.2:15];
G12= G1*G2;
for K3 = K3Range(:,1)'
    overshootArr=[];
    settlingTimeArr=[];
    disturbanceArr=[];
    for K1 = K1Range(1,:)
        CLTF = ((K1+K3*s)*G12)/(1+(K1+K3*s)*G12);
        y = step(CLTF, t);
        info = stepinfo(y, t, 'SettlingTimeThreshold', 0.02);
        overshootArr = [overshootArr, info.Overshoot];

        settlingTimeArr = [settlingTimeArr, info.SettlingTime];
        Tw = G2/(1+(K1+K3*s)*G12);
        y = step(Tw,t);
        maxDisturbance = max(y);
        disturbanceArr = [disturbanceArr, maxDisturbance];
        if(info.Overshoot <= 5 & info.SettlingTime <0.25 & y<0.005)
            candidatePairs = [candidatePairs; Ka, Kh];
        end
    end
end

overshootMatrix = [overshootMatrix; overshootArr];
settlingTimeMatrix = [settlingTimeMatrix; settlingTimeArr];
disturbanceMatrix = [disturbanceMatrix; disturbanceArr];
end

mesh(K1Range, K3Range,
    overshootMatrix); xlabel('K1'); ylabel('K3'); zlabel('Overshoot')
mesh(K1Range, K3Range,
    settlingTimeMatrix); xlabel('K1'); ylabel('K3'); zlabel('SettlingTime')
mesh(K1Range,
    K3Range,disturbanceMatrix); xlabel('K1'); ylabel('K3'); zlabel('Disturban
    ce'); view(3
candidatePairs

```

To examine the correctness of the answer, we select $K_1 = 100$ and $K_3 = 5$ to evaluate more specifically

```

K1=100; K3=5;
CLTF = ((K1+K3*s)*G12)/(1+(K1+K3*s)*G12);
y = step(CLTF, t);
info = stepinfo(y, t, 'SettlingTimeThreshold', 0.02)
plot(t,y);

info =
    RiseTime: 0.0879
    SettlingTime: 0.1560
    SettlingMin: 0.9211
    SettlingMax: 1.0000
    Overshoot: 0
    Undershoot: 0
    Peak: 1.0000
    PeakTime: 0.8000

```

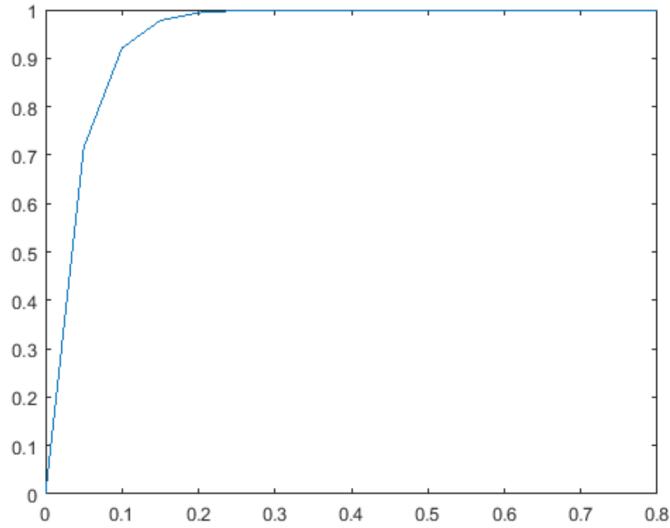


Figure 14 – System overshoot

With the information generated and the curve shown in Figure 14, we can discover that at the same time satisfying other system requirements, the system has no overshoot. The reason is that K_3 acts as a damping factor that prevents the response from overshooting. We can then analyze the disturbance.

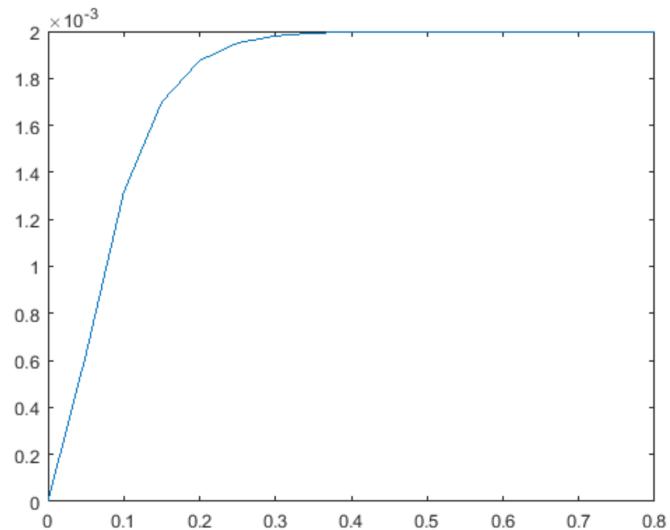


Figure 15 – System disturbance

The maximum disturbance only reaches 2×10^{-3} , within the range of requirement. We can conclude for this task that, compared to the proportional compensator, the PID compensator can be much more effective in controlling the system especially reducing the overshoot.

2.6 Task 6 – PID Compensator

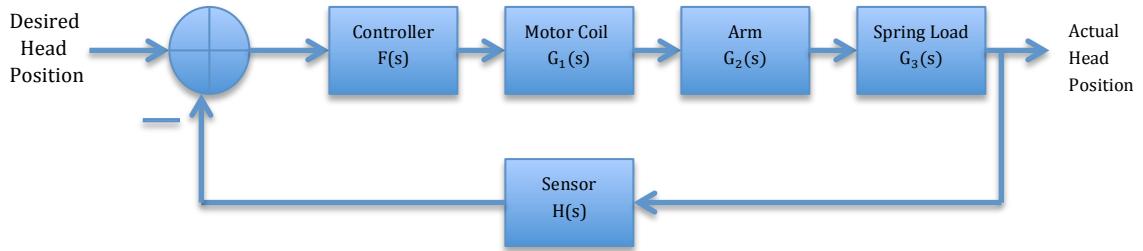
Part A

Part B

Part C

3.0 CONCLUSION

The final architecture of the system that satisfies all of the performance requirements is shown below:



Transfer function of each block are presented below:

$$F(s) = K_1 + \frac{K_2}{s} + K_3 s \quad (11)$$

$$G_1(s) = \frac{K_m}{Ls+R} \quad (12)$$

$$G_2(s) = \frac{1}{Js^2+bs} \quad (13)$$

$$G_3(s) = \frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2} = \frac{1}{1+\frac{2\zeta s}{\omega_n}+\frac{s^2}{\omega_n^2}} \quad (14)$$

$$H(s) = 1 \quad (15)$$

The closed-loop transfer function for the entire system is:

$$\frac{F(s)G_1(s)G_2(s)G_3(s)}{1+F(s)G_1(s)G_2(s)G_3(s)} \quad (16)$$

Parameters are shown in the table below:

Table 1

Parameters	Value
Inertia of arm and head J	$1 \text{ N} \cdot \text{m} \cdot \text{s}^2/\text{rad}$
Friction b	20 Kg/m/s
Field resistance R	1Ω
Field inductance L	0.001 H
Motor constant K_m	$5 \text{ N} \cdot \text{m/A}$
ζ	0.3
ω_n	18850 rad
K_1	
K_2	0
K_3	

We compared the system performance with the given specifications in **Table 2**

Table 2

Specification Variable Name	Specification Variable Value	Designed System Value
Percent Overshoot	< 5%	
Settling time(2% deviation)	< 250ms	
Maximum value of response to a unit step disturbance	< 5×10^3	