Control Systems Design Module ELEC5570M Coursework

NAME: Junnan Liu

STUDENT ID: 201715540

CATALOG

[**Control Systems Design Module ELEC5570M Coursework** 1](#_Toc151586215)

[Part A: Root Locus 4](#_Toc151586216)

[1 Task 1 4](#_Toc151586217)

[1.1 4](#_Toc151586218)

[1.2 5](#_Toc151586219)

[Design of a PD controller 8](#_Toc151586220)

[2 Task 2 8](#_Toc151586221)

[2.1 8](#_Toc151586222)

[2.2 9](#_Toc151586223)

[2.3 10](#_Toc151586224)

[Design of a PID controller 12](#_Toc151586225)

[3 Task 3 12](#_Toc151586226)

[3.1 12](#_Toc151586227)

[3.2 14](#_Toc151586228)

[4 Task 4 17](#_Toc151586229)

[4.1 17](#_Toc151586230)

[4.2 19](#_Toc151586231)

[Part B: Frequency Response 21](#_Toc151586232)

[5 Task 5 22](#_Toc151586233)

[5.1 22](#_Toc151586234)

[5.2 22](#_Toc151586235)

[5.3 23](#_Toc151586236)

[Design of a Lead Compensator for K(s) 23](#_Toc151586237)

[6 Task 6 24](#_Toc151586238)

[6.3 24](#_Toc151586239)

[6.4 25](#_Toc151586240)

[Design of a Lead-lag Compensator for K(s) 26](#_Toc151586241)

[7 Task 7 26](#_Toc151586242)

[7.1 26](#_Toc151586243)

[7.2 28](#_Toc151586244)

[7.3 30](#_Toc151586245)

Part A: Root Locus

# Task 1

图示

描述已自动生成

figure 1

## 

When typing the code into the MATLAB (figure 2 ), we can get the root locus plot as figure 3 , which poles are P1(-0.5,2.69j),P2(-0.5,-2.69j),P3(-5,0), Z1(-2,0).

文本, 信件

描述已自动生成

figure 2

图形用户界面, 图表

描述已自动生成

figure 3 The root locus plot

## 

The second order approximation of the third order system is as follow:

Where

Thus

From the calculations above, we can write the code as follows (figure 4):

图形用户界面, 文本, 应用程序

描述已自动生成

figure 4

Through this code, we can plot the open loop transfer function which is approximated to second order system (figure 5). when the two-conjugate complex poles meet, the gain is about 17.3. We can calculate the settling time as follows:

图形用户界面

中度可信度描述已自动生成

figure 5

Through multiple the gain (17.3) and switch it to step response, we can get when the (figure 6),from the Step Response we plotted we can see the settling time is equal to 1.21 seconds.

图形用户界面

描述已自动生成

figure 6

The reason why the settling time is different from the 0.826 is that it may has error when use the method to appropriate second order system.

Design of a PD controller

# Task 2

## 

We can get the open loop poles and zeros through G(s), which include P1(-0.5,2.69j), P2(-0.5,-2.69j),P3(-5,0),P4(-5,5j),Z0(-2,0).

图示, 工程绘图

描述已自动生成

figure 7

According to the figure 7, the equation of the can be calculated as follows:

Where

So the angle of between the P4 and the z1 is:

Where the absolute value of the abscissa of point z1 is:

Thus Z1(-7.636,0).

## 

According to 2.1, we have:

The compensated open-loop transfer function is:

The gain of the lead compensator is calculated using the magnitude of the condition as follows:

## 

After running the code in figure 8, we can get the root locus of the Q() and the step response of the sys\_c ().

手机屏幕截图

描述已自动生成

figure 8

According to the figure 9, we can see through the open loop locus the when (-5,5i), and thus the settling time is:

图示

描述已自动生成

figure 9

Verify the calculations by plotting the step response, we can see the Steady state value = 0.741 and thus the settling time (within 2% of the final value – 0.741) is 0.916 seconds, as shown below (figure 10), which is about the same as the previous estimate of 0.8 seconds.

图表

低可信度描述已自动生成

figure 10

Design of a PID controller

# Task 3

## 

We can get the open loop poles and zeros through G(s), which include P1(-0.5,2.69j), P2(-0.5,-2.69j),P3(-5,0),P4(0,0),P5(-5,5j),P6(-5,-5j),Z0(-2,0),Z2(-2,0).

图示, 工程绘图

描述已自动生成

figure 11

According to the figure 11, the equation of the can be calculated as follows:

Where

So the angle of between the P5 and the z1 is:

Where the absolute value of the abscissa of point z1 is:

Thus Z1(-6.224,0).

Because we have:

Thus, the compensated open-loop transfer function is:

The gain of the lead compensator is calculated using the magnitude of the condition as follows:

## 

图形用户界面, 文本, 应用程序

描述已自动生成

figure 12

Through above code (figure 12) we can verify the calculations we made in 3.1, after adding the PID controller with z1(-6.224,0) and k=9.10. We can see the open loop locus go through the (-5, ±5j), which means the PID control is success.

Because the required complex closed-loop poles are still set as 𝑠(𝑑𝑒𝑠𝑖𝑟𝑒𝑑) = −5 ± 𝑗5 (figure 13). We choose the (-5,5j) to calculate the settling time:

图形用户界面, 图示

描述已自动生成

figure 13

图形用户界面

描述已自动生成

figure 14

# Task 4

## 

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

figure 15

According to the code above, we can draw the open loop locus and step response both the PD controller and PID controller.

图形用户界面

描述已自动生成

figure 16

According to the figure above, Task\_2\_PD represents the system with a PD controller, while Task\_3\_PID represents the system with a PID controller. Both controllers exhibit poles at (-5, ±5j), satisfying the given requirement. Comparing the system with the PD controller to the system with the PID controller, the latter introduces an additional pole P1(0,0) and zero Z1(-2,0) near the dominant pole. Since the new pole P1 is closer to the origin compared to Z1, Task\_3\_PID is positioned to the right of Task\_2\_PD. As a result, Task\_3\_PID is closer to the imaginary axis, leading to a faster transient response in comparison to Task\_2\_PD.

图形用户界面, 应用程序

描述已自动生成

figure 17

As the figure 17 show, Task\_2\_PD represents the system with a PD controller, while Task\_3\_PID represents the system with a PID controller. The figure clearly shows that Task\_3\_PID has a steady-state error of 0, whereas Task\_2\_PD has a steady-state error equal to 1-0.737 = 0.263. This difference is due to the capability of the PID controller to improve the system type and eliminate steady-state error.

In comparison, the Amplitude peak of Task\_3\_PID is noticeably larger than that of Task\_2\_PD. This may be attributed to the integration component of the PID controller, introducing a pole at the origin (0,0) and influencing the primary pole, thereby rendering the system unstable.

Other dynamic performance indicators such as settling time, rise time, and peak time show little change.

## 

According to the requirements, writing the code as follows:

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

figure 18

图形用户界面, 应用程序

描述已自动生成

figure 19

According to the figure above, the yellow line represents the system with a scalar controller (k=114.514), the red line represents the system with a PID controller, and the blue line represents the system with a PD controller. Comparing Task\_4\_Progain with Task\_2\_PD, it is evident that Task\_2\_PD significantly reduces overshoot and settling time, but results in a larger steady-state error. In comparison, Task\_3\_PID not only eliminates the steady-state error, but also maintains the same dynamic performance indicators as Task\_2\_PD.

Among the three controllers, Task\_4\_Progain exhibits the worst transient response and highest overshoot. On the other hand, Task\_2\_PD has the highest steady-state error. This can be attributed to the fact that when the input is 1/s and the system type is equal to 1, the steady-state error is determined by the scalar (A/k). Consequently, the steady-state error of Task\_4\_Progain (K=114.514) is smaller than that of Task\_2\_PD (K=6.837). In Task\_3\_PID, the steady-state error is zero due to the system type being 2 and the input being 1/s, resulting in an ess of 0.

Part B: Frequency Response

图示

描述已自动生成

figure 20

# Task 5

## 

文本, 信件

描述已自动生成

figure 21

According to the code above (figure 21), we can build the bode diagrams (figure 22). Moreover, when the Magnitude is about 0, the crossover frequency 𝜔𝑜 is equal to 2.2 rad/s. The PM = 180+ (-147) = 33°.

图形用户界面

描述已自动生成

figure 22

## 

The close loop transfer function is:

And thus

## 

According to calculations from 5.2, we can compare the with and compare the × 100 and phase margin (PM) as follows:

Design of a Lead Compensator for K(s)

# Task 6

## 

图形用户界面

描述已自动生成

figure 23

According to figure 23, when the Frequency is to be made . At this point, the .

So additional phase required is:

Which is

At so additional gain required is:

thus

And

## 

文本

描述已自动生成

figure 24

图片包含 图形用户界面

描述已自动生成

figure 25

图示

低可信度描述已自动生成

figure 26

After adding the k, a and tau into the code above (figure 24), we can draw the Step Response (figure 25) and the Bode Diagram (figure 26). Through the Bode Diagram we can verify the crossover frequency is 6.13 rad/s and the phase margin is .

According to the Step Response (figure 25) we can infer that the Amplitude of the settling time can be calculated as follows:

.

And thus, according to the figure 18, we can get:

.

The rise time is the period of 10 percent of the steady state value () to 90 percent of the steady state value (). Consequently, the rise time is equal to:

.

From the figure 18, the overshoot can be calculated as follows:

Design of a Lead-lag Compensator for K(s)

# Task 7

## 

According to the requirements and the results of the Task 6 we can get information as follows:

And thus

So , We will choose

Since

So and

## 

图形用户界面, 文本, 应用程序

描述已自动生成

figure 27

Through combining the results of the Task 6 and Task 7.1 to code (figure 27), we can compare the lead compensators with lead-lag compensators in Bode Diagram (figure 28) and Step Response (figure 29) at the same time.

图示

中度可信度描述已自动生成

figure 28

The figure above illustrates the performance of two systems: the red line represents a system with a lead-lag compensator named Task\_7\_leadlag, while the blue line represents a system with a lead compensator named Task\_6\_lead. Both systems have experienced a change in crossover frequency, which is approximately 6.13 rad/s. Upon comparing Task\_7\_leadlag with Task\_6\_lead, it is evident that the phase margin of Task\_7\_leadlag has significantly increased, as well as the system's gain at low frequencies.

图片包含 图示

描述已自动生成

figure 29

According to the Step Response, the red line represents the system with a lead-lag compensator named Task\_7\_leadlag, while the blue line represents the system with a lead compensator named Task\_6\_lead. Comparing the performance of Task\_7\_leadlag with Task\_6\_lead, it can be observed that they have similar peak times (0.463 seconds and 0.465 seconds), rise times (0.173 seconds and 0.1863 seconds), and settling times (1.46 seconds and 1.01 seconds). This indicates that the addition of the lag compensator has minimal impact on the dynamic performance indicators. However, the steady-state error of Task\_7\_leadlag is smaller than that of Task\_6\_lead. This reduction in steady-state error is primarily attributed to the lag compensator's ability to effectively diminish steady-state error while exerting negligible influence on dynamic performance indicators.

图形用户界面, 文本, 应用程序

描述已自动生成

figure 30

We add a system with the scalar (k=6.73) into the code (figure 30), and plot the Step Response as follows:

图片包含 图形用户界面

描述已自动生成

图表, 折线图

描述已自动生成

figure 31

From this picture (figure 31), we can see the blue line is the Task\_6\_lead, which is the system with the lead compensator. The red line is the Task\_7\_lead\_lag, which is the system with the lead-lag compensator. The yellow line is the Task\_7.3\_scalar, which is the system with the scalar compensator. The Task\_7.3\_scalar has the longest settling time and the highest overshoot, which mainly because the lead compensator can significantly decrease the settling time and overshoot through improve the PM. The Task\_7\_leadlag has the minimum ess, which is mainly because the lag compensator can reduce the ess while both of the lead compensator and the scalar will improve the ess. All of the peak time and rise time are similar.