



| |
|-------------------|
| 班级：自动化 1601 |
| 学号：16401100128 |
| 姓名： <div></div> |
| 指导教师： <div></div> |
| 成绩： |

自动控制原理课程设计报告

设计任务一：直流电动机调速系统的构建

设计任务二：直流电动机调速系统性能分析

设计任务三：直流电动机调速系统 PID 控制器、

根轨迹控制器、频域控制器

目录

| | |
|-----------------------|--|
| 一. 课程设计内容概括..... | |
| 二. 直流电动机调速系统的构建..... | |
| 三. 直流电动机调速系统性能分析..... | |
| 四. 直流电动机调速系统控制器..... | |
| (1) PID 控制器 | |
| (2) 根轨迹控制器 | |
| (3) 频域控制器 | |
| 五. 自我总结..... | |

一. 课程设计内容概括

1. 本次自动控制原理课程设计运用 Matlab 仿真软件构建一个直流电动机调速系统，具体设计内容包括：

- (1) 直流电动机调速系统数学模型，流程图及闭环传递函数的构建；
- (2) 直流电动机调速系统性能分析；
- (3) 流电动机调速系统 PID 控制器、根轨迹控制器、频域控制器。

2. 直流电动机调速系统的输入电压 1V，电动机稳态时角速度 0.1 rad /秒，系统单位阶跃响应设计要求如下：

- (1) 电动机转速的稳态误差小于 1%，
- (2) 超调量小于 5%，
- (3) 开环系统额定沉降时间小于 2 秒

3. 本次自动控制原理课程设计的目的：

(1) 培养理论联系实际的设计思想，训练综合运用经典控制理论和相关课程知识的能力；

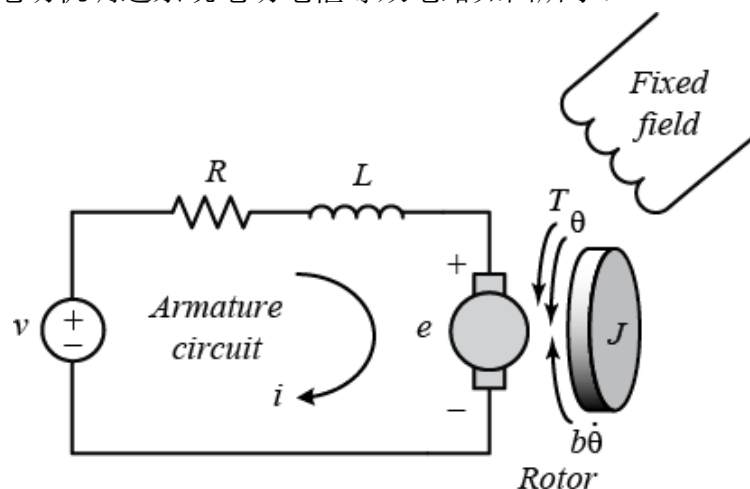
(2) 掌握自动控制原理的时域分析法、根轨迹法、频域分析法，以及各种校正装置的作用及用法，能够利用不同的分析法对给定系统进行性能分析，能根据不同的系统性能指标要求进行合理的系统设计，并调试满足系统的指标；

(3) 学会运用 Matlab 仿真软件进行系统仿真与调试；

(4) 锻炼独立思考和动手解决控制系统实际问题的能力，为今后从事控制相关工作打下较好的基础。

二. 直流电动机调速系统的构建

直流电动机调速系统电动电枢等效电路如图所示：



假设系统的输入电压源(V)应用于电动机的电枢，而输出轴的转速 $\dot{\theta}$ ，摩擦力矩与轴角速度成正比。

直流电动机调速系统的物理参数为：

转子的转动惯量 $0.02 \text{ kg}\cdot\text{m}^2$ ($J = 0.02$)

运动粘滞摩擦常数 0.1 Pas ($b = 0.1$)

电动势常数 $0.005 \text{ V / rad / 秒}$ ($K_e = 0.005$)

电动机转矩常数 0.005 N.m/A ($K_t = 0.005$)

欧姆电阻 1Ω ($R = 1$)

电动电感 1 H ($L = 1$)

通常，由直流电动机产生的转矩与电枢电流和磁场强度成比例。在这个例子中，我们将假定所述磁场是恒定的，因此，该电动机转矩正比于仅电枢电流 i 由一个恒定的因子 K_t 如下面的公式所示。这被称为电枢控制电动机。

$$(1) T = K_t i$$

反电动势 e 与轴的角速度成一定的常数因子 K_e 。

$$(2) e = K_e \dot{\theta}$$

在 SI 单位中，电动机转矩和反电动势常数相等，即 $K_t = K_e$ ；因此，我们将 K 用来表示电机转矩常数和反电动势常数。

从上图中，我们可以推导出基于牛顿第二定律和基尔霍夫电压定律的以下控制方程。

$$(3) J\ddot{\theta} + b\dot{\theta} = Ki$$

$$(4) L\frac{di}{dt} + Ri = V - K\dot{\theta}$$

应用拉普拉斯变换，上述建模方程可以用拉普拉斯变量 s 表示。

$$(5) s(Js + b)\Theta(s) = KI(s)$$

$$(6) (Ls + R)I(s) = V(s) - Ks\Theta(s)$$

我们通过消除 $I(s)$ 上述两个方程式来达到以下开环传递函数，其中转速被认为是输出，而电枢电压被认为是输入。

$$(7) P(s) = \frac{\dot{\Theta}(s)}{V(s)} = \frac{K}{(Js + b)(Ls + R) + K^2} \quad \left[\frac{\text{rad/sec}}{\text{V}} \right]$$

运用 Matlab 仿真软件建立直流电动机调速系统数学模型，写出系统方程、传递函数及状态方程。

$$P_{\text{motor}} =$$

$$0.005$$

$$0.02 s^2 + 0.12 s + 0.1$$

利用Matlab计算出开环的两个极点：X=solve('0.02*X^2+0.12*X+0.1')

$$\text{解得：} \quad X = \quad -5.0 \quad \quad -1.0$$

从上面可以看出，开环传递函数有两个实极，一个在 $s = -1$ ，另一个在 $s = -5$ 。由于两极都是实数，因此我们已经看到阶跃响应（或过冲）没有振荡。此外，由于一极的负值是另一极的 5 倍，因此两极中较慢的极点将主导动态。也就是说， $s = -1$ 处的极点主要决定系统的响应速度，并且系统的行为类似于二阶系统。

三．直流电动机调速系统性能分析

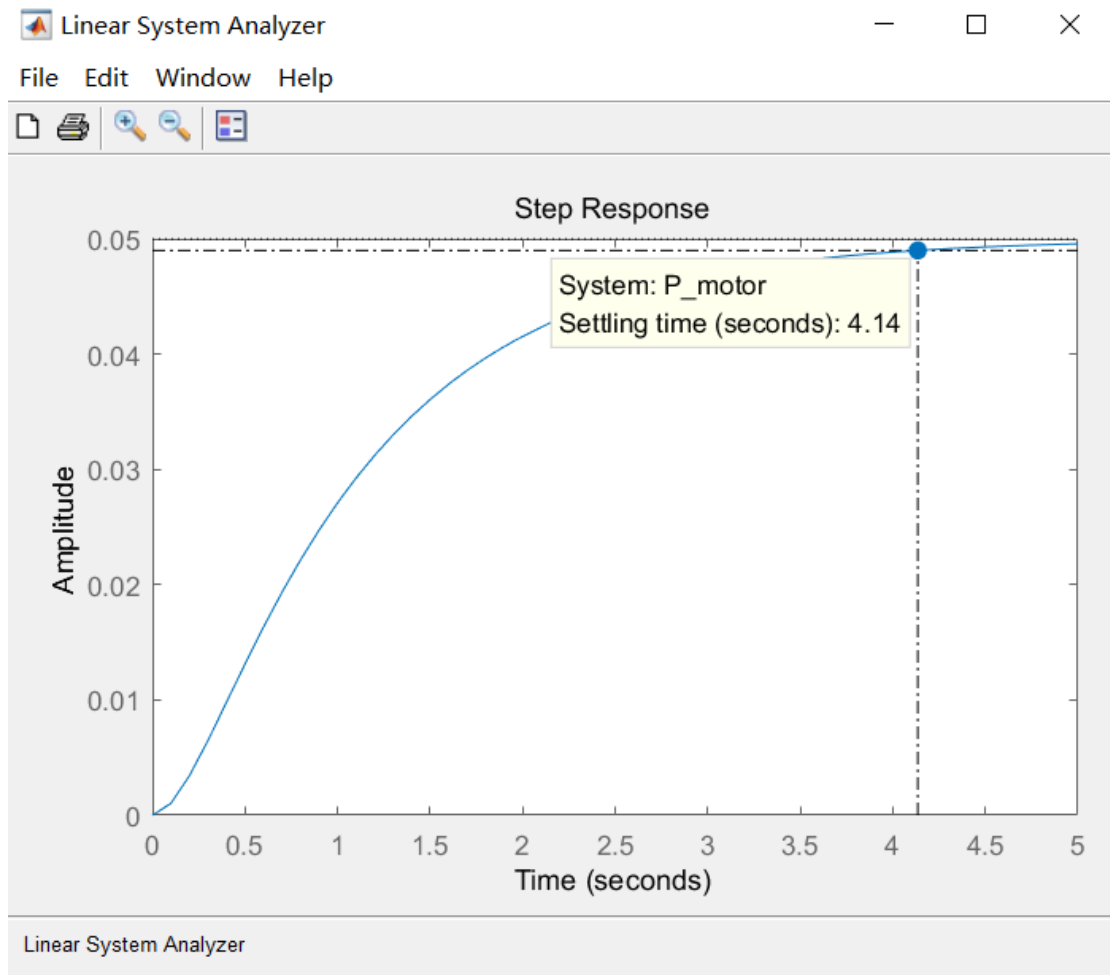
设计要求：

首先考虑我们的无补偿电机在稳定状态下以 0.1 弧度/秒的速度旋转，输入电压为 1 伏（这在直流电机速度：系统分析页面中进行了演示，其中模拟了系统的开环响应）。由于电动机的最基本要求是它应以所需的速度旋转，我们将要求电动机速度的稳态误差小于 1%。我们的电机的另一个性能要求是它一旦开启就必须加速到其稳态速度。在这种情况下，我们希望它的建立时间少于 2 秒。此外，由于比参考速度快的速度可能会损坏设备，我们希望得到阶跃响应，过冲小于 5%。

总之，对于电机速度的单位步进命令，控制系统的输出应满足以下要求。

- (1) 沉降时间不到 2 秒
- (2) 超调量不到 5%
- (3) 稳态误差小于 1%

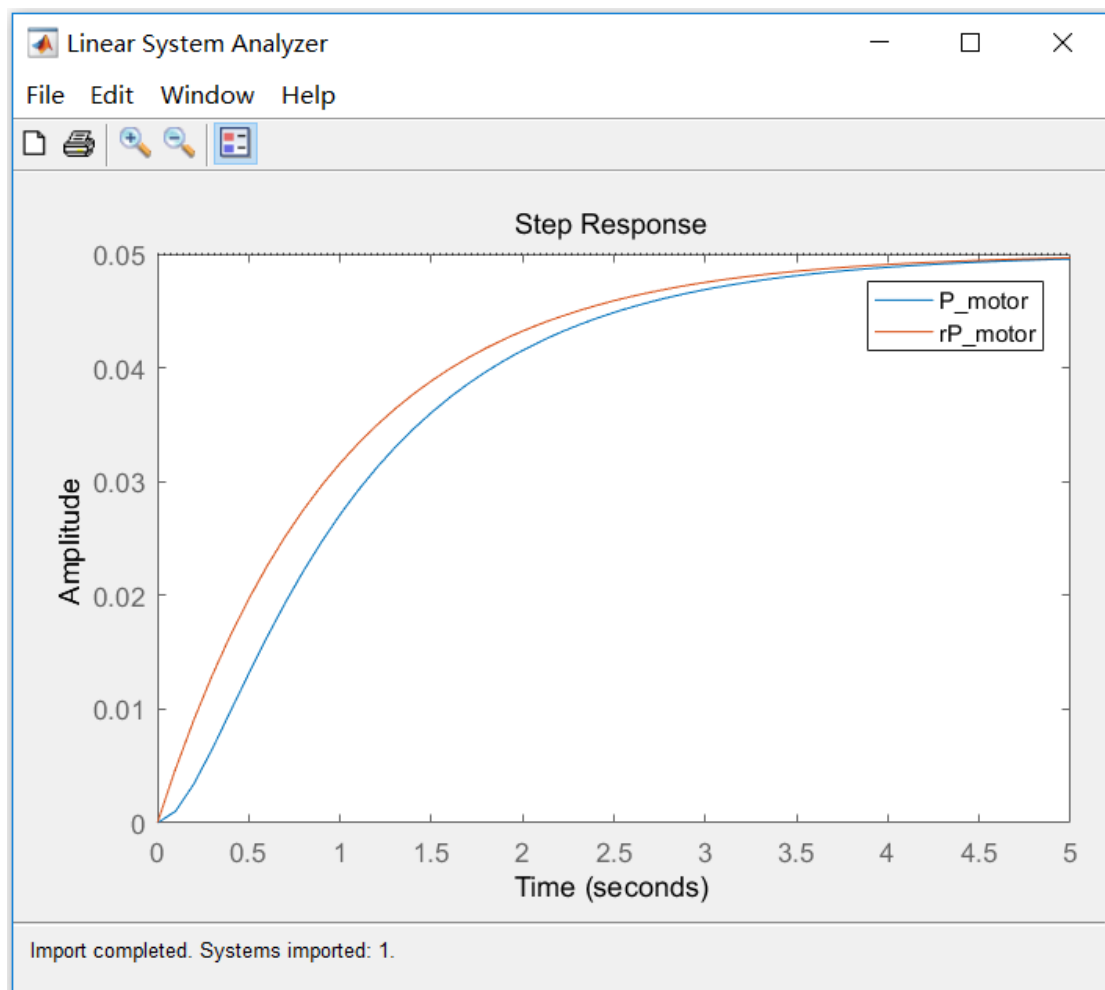
由直流电动机调速系统的构建的传递函数可知，开环传递函数有两个实极，一个在 $s = -1$ ，另一个在 $s = -5$ 。由于两极都是实数，因此我们已经看到阶跃响应（或过冲）没有振荡。此外，由于一极的负值是另一极的 5 倍，因此两极中较慢的极点将主导动态。也就是说， $s = -1$ 处的极点主要决定系统的响应速度，并且系统的行为类似于一阶系统。rP_motor 为以 $s = -1$ 为主导极点的阶跃响应曲线。



从图中可以看出，当 1 伏施加到系统时，电机只能达到 0.05 弧度/秒的最大速度，比我们想要的速度小十倍。此外，它需要电机 4.14 秒才能达到其稳态速度；这不符合我们的 2 秒建立时间标准。

由于一极的负值是另一极的 5 倍，因此两极中较慢的极点将主导动态。也就是说， $s = -1$ 处的极点主要决定系统的响应速度，并且系统的行为类似于一阶系统传递函数为：

$$\text{rP_motor} = \frac{0.005}{s + 1}$$



P_motor：原始传递函数曲线

rP_motor：以 $s = -1$ 处极点为主导极点的简化曲线

从上面我们可以看出，我们的电机系统的一阶近似是相对准确的。在 $t = 0$ 时可以看到主要差异，其中二阶系统的导数为零，但一阶模型不会。

一阶模型的时间为 2 秒，接近我们实际系统的 4.14 秒。在本例的其余页面中，将设计不同的控制器以显著降低稳态误差和稳定时间，同时仍满足给定的超调要求。

四．直流电动机调速系统控制器

(1) PID 控制器

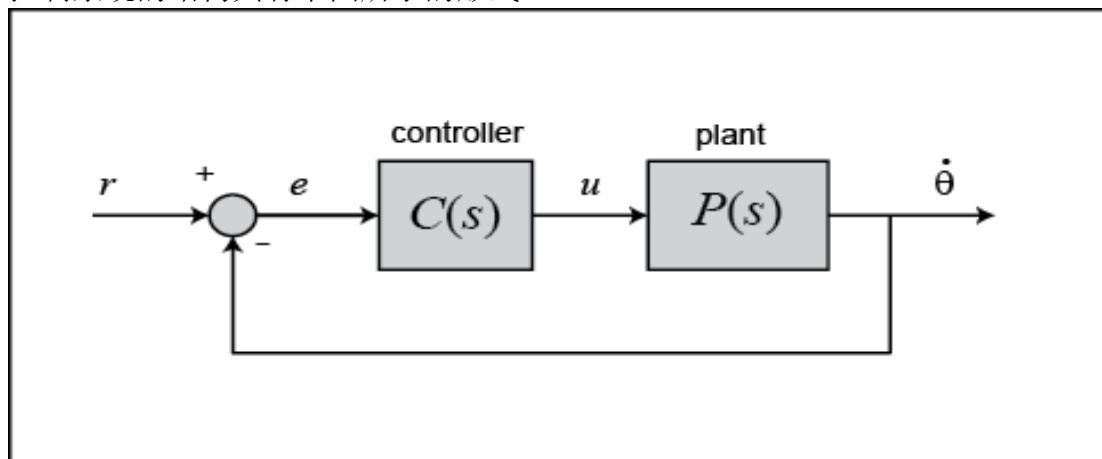
拉普拉斯域中的动力学方程和直流电动机的开环传递函数如下：

$$(1) \quad s(Js + b)\Theta(s) = KI(s)$$

$$(2) \quad (Ls + R)I(s) = V(s) - Ks\Theta(s)$$

$$(3) \quad P(s) = \frac{\dot{\Theta}(s)}{V(s)} = \frac{K}{(Js + b)(Ls + R) + K^2} \quad \left[\frac{\text{rad/sec}}{\text{V}}\right]$$

控制系统的结构具有下图所示的形式。



对于 1-rad / sec 步进参考，设计标准如下。

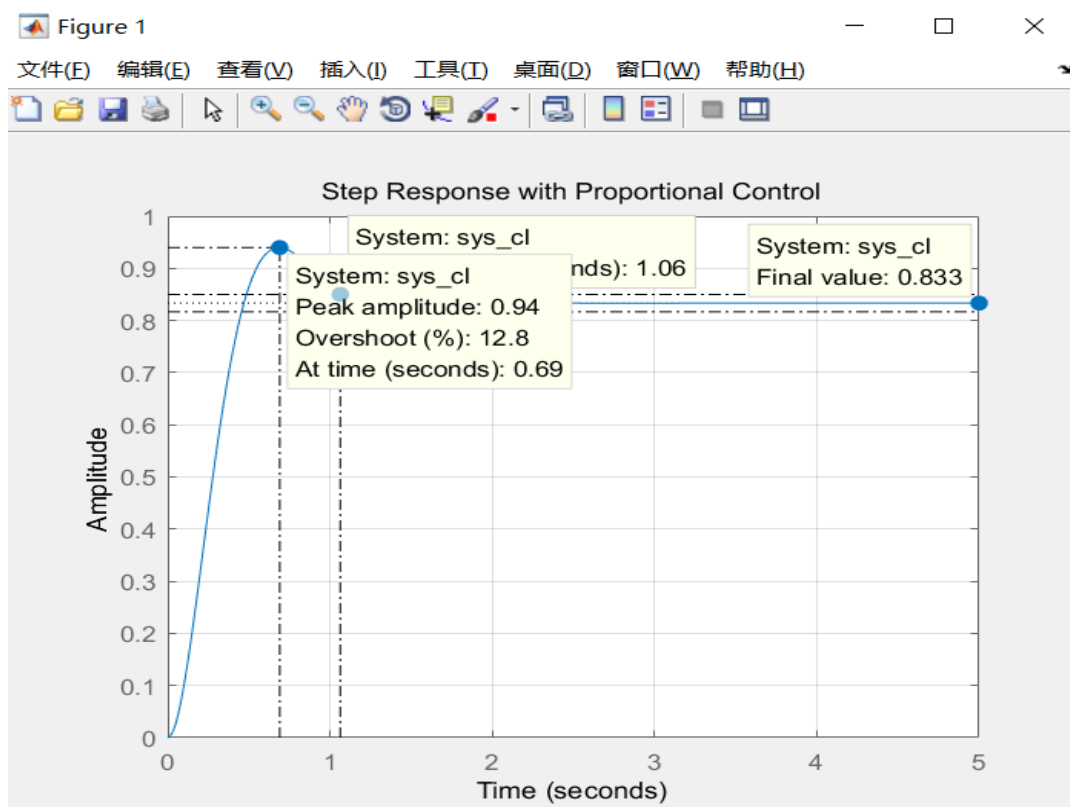
- 沉降时间不到 2 秒
- 过冲不到 5%
- 稳态误差小于 1%

PID 控制器的传递函数是：

$$C(s) = K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s}$$

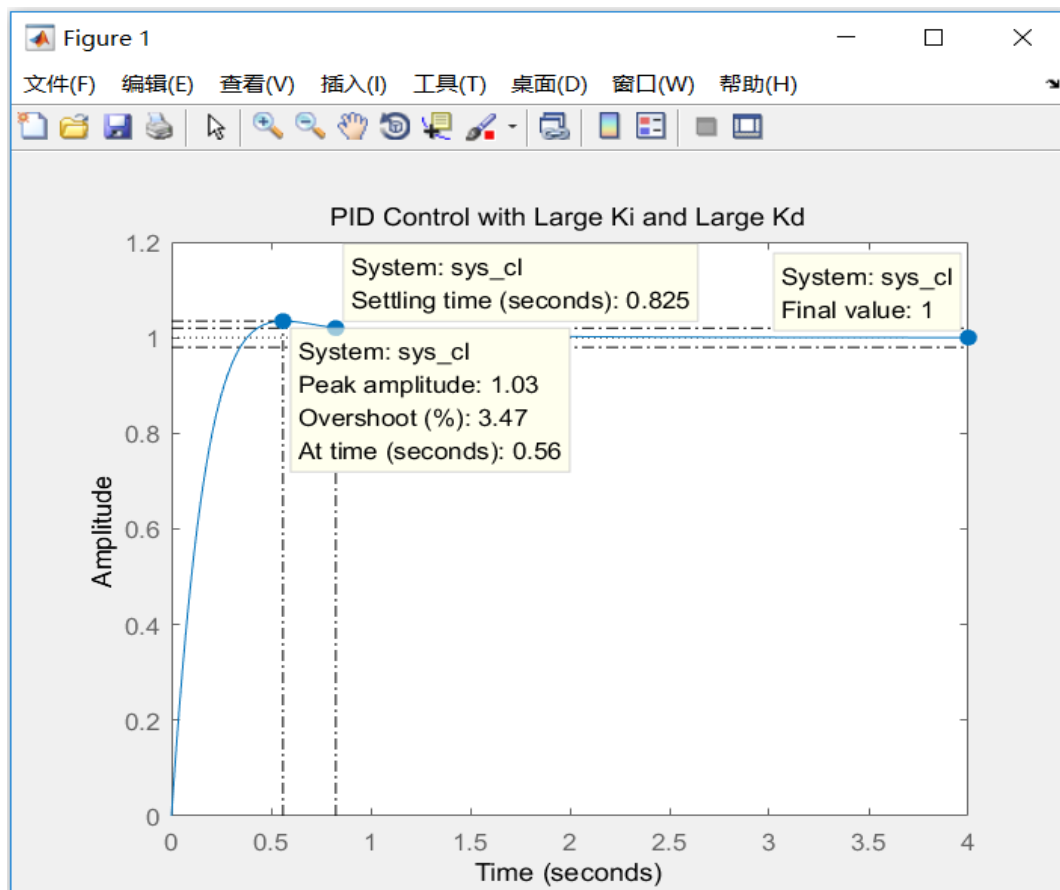
比例控制：

增加比例增益 K_p 将减少稳态误差。然而增加 K_p 经常导致过冲增加，因此并非所有的设计要求都可以通过简单的比例控制器来满足。



PID 控制:

增加的 K_d 减少了由此引起的超调。现在我们知道如果我们使用 $K_p=200$, $K_i=200$ 和 $K_d=25$ 的 PID 控制器, 我们的所有设计要求都将得到满足。具体如下图



PID 控制完全符合设计要求:

- 沉降时间为 0.825 秒
- 超调量 3.47%
- 稳态误差为 0%

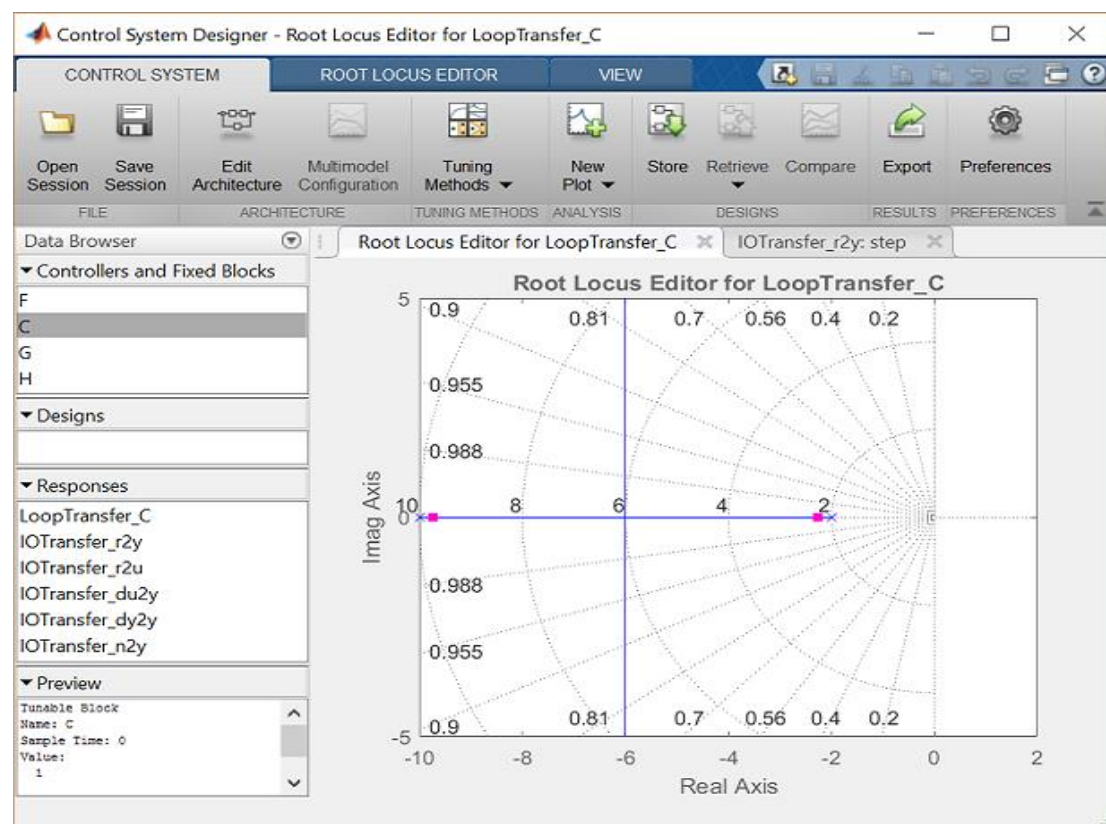
(2) 根轨迹控制器

绘制开环根轨迹

根轨迹设计的主要思想是从根轨迹图预测闭环响应,该图描绘了可能的闭环极点位置并且是从开环传递函数中得出的。然后通过控制器添加零和/或极点,可以修改根轨迹以实现期望的闭环响应。

我们将为我们的设计使用 Control System Designer 图形用户界面。此工具允许您通过根轨迹图以图形方式调整控制器。让我们首先查看未经过补偿的根轨迹。这是通过将命令 `controlSystemDesigner('rlocus', P_motor)` 添加到 m 文件的末尾并在命令行运行文件或者转到 MATLAB 工具条的 APPS 选项卡并单击控件下的应用程序图标来实现的。系统分析与设计。

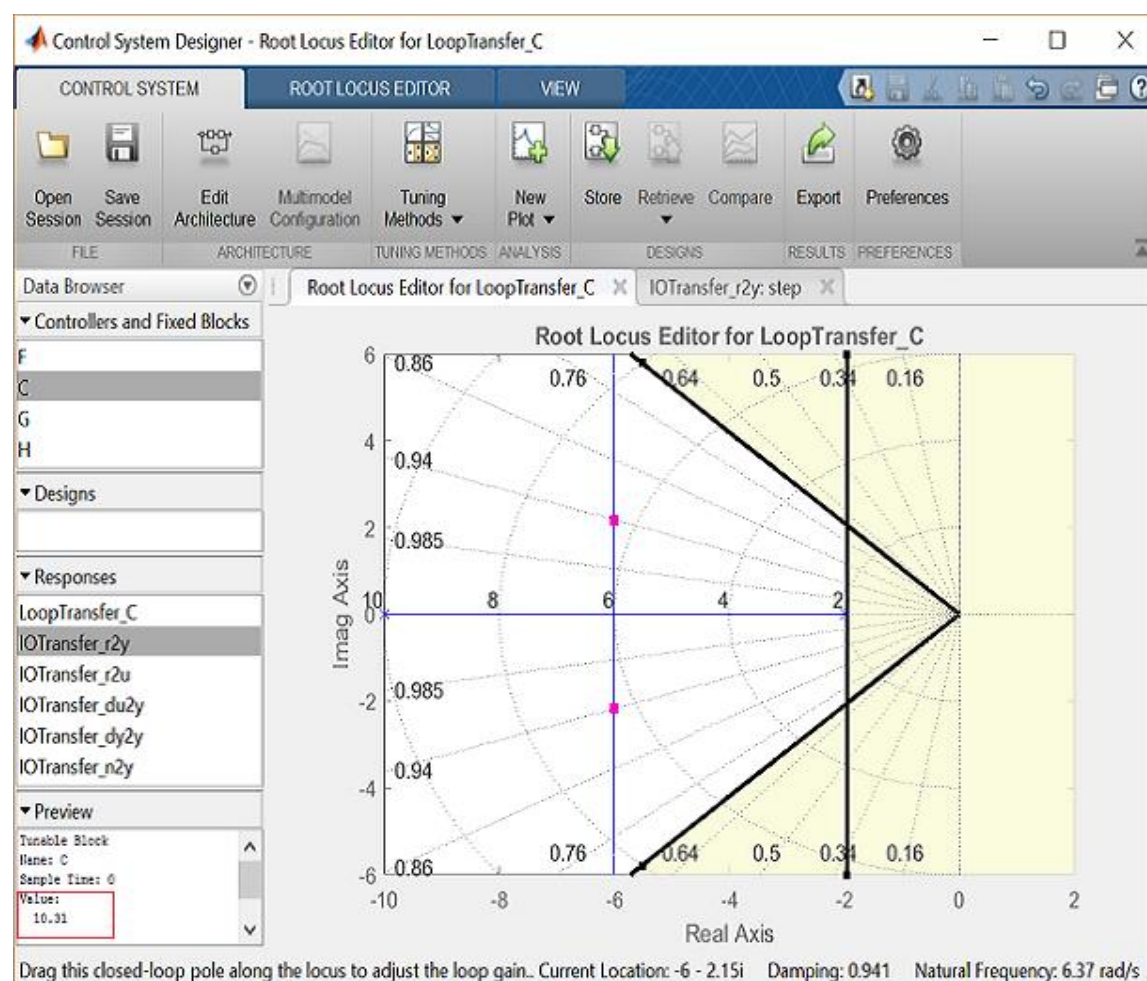
标题为 Control System Designer 的一个窗口最初将打开,其格式如下图所示。在窗口中,您将能够看到根轨迹图和通过 `controlSystemDesigner` 函数传递的传递函数的闭环步骤响应。如果函数调用中省略了字符串 'rlocus',则除了根轨迹图和闭环步骤响应图之外,默认初始窗口还包括波特图。您可以从“控制系统设计器”窗口的“视图”选项卡中排列绘图的位置。右键单击根轨迹图并选择“网格”将使窗口显示如下。



我们的设计要求规定建立时间小于 2 秒，超调小于 5%。系统闭环极点的位置提供有关系统瞬态响应的信息。该控制系统的设计可让您在复杂的指定区域对应的具体设计要求-平面。提供的区域对应于规范的二阶系统，但一般来说，即使对于具有零的高阶系统或系统，也是一个好的起点。

通过右键单击绘图并从结果菜单中选择“设计要求”>“新建”，可以将这些所需区域添加到根轨迹图中。您可以添加许多设计要求，包括稳定时间，过冲百分比，阻尼比，固有频率和通用区域约束。

将我们的稳定时间和超调量要求添加到根轨迹图中会产生下图。



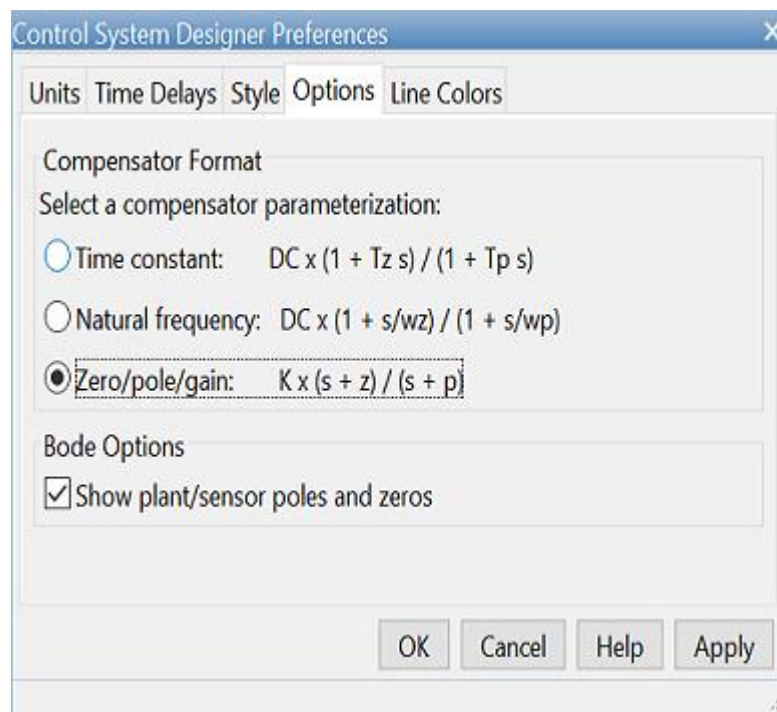
所得到的闭环极点的所需区域由上图的无阴影区域示出。更具体地说，以原点为中心的两条光线代表了超调要求；这些光线与负实轴的角度越小，允许的过冲越少。移动极点后，如图所示

添加滞后控制器：

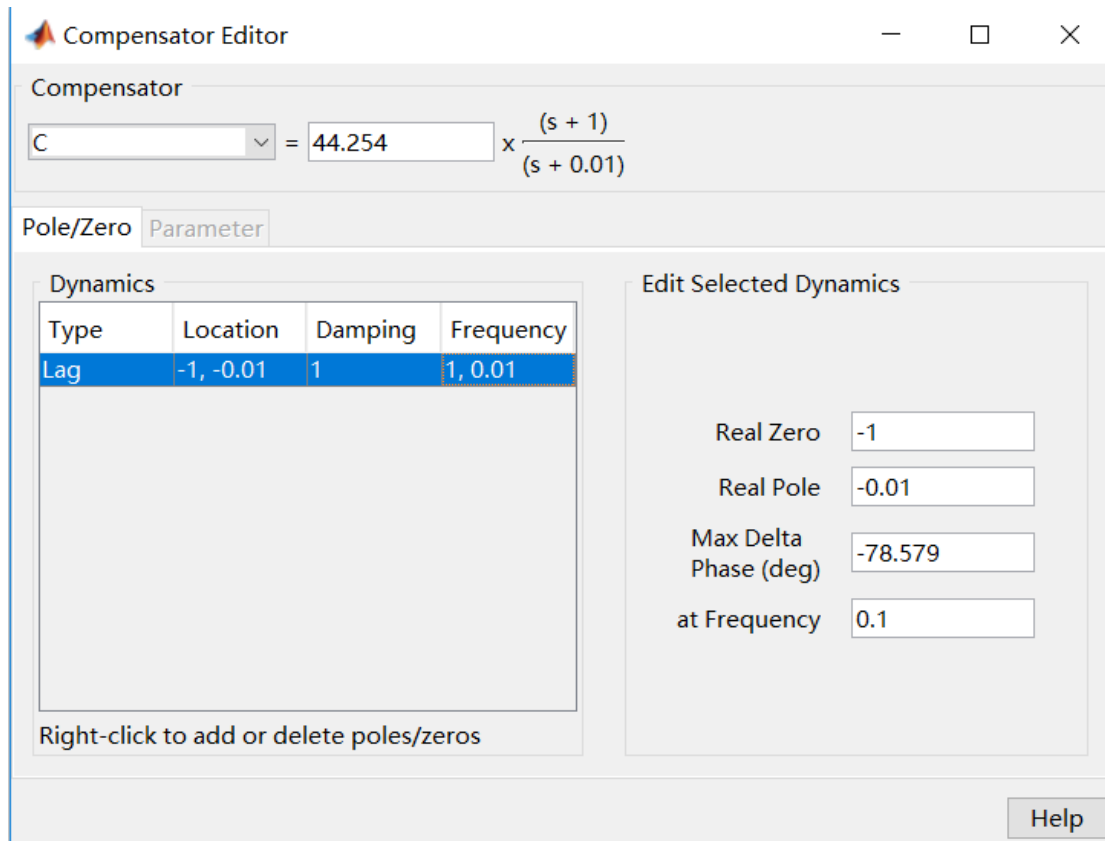
在上面我们看到比例控制器满足了过冲和稳定时间标准，但稳态误差要求却没有。滞后补偿器是一种类型的公知的，以便能够减少稳态误差的控制器。但是，我们的设计必须小心，不要过多地增加稳定时间。让我们首先尝试添加下面给出的形式的滞后补偿器。

$$(4) \quad C(s) = \frac{(s + 1)}{(s + 0.01)}$$

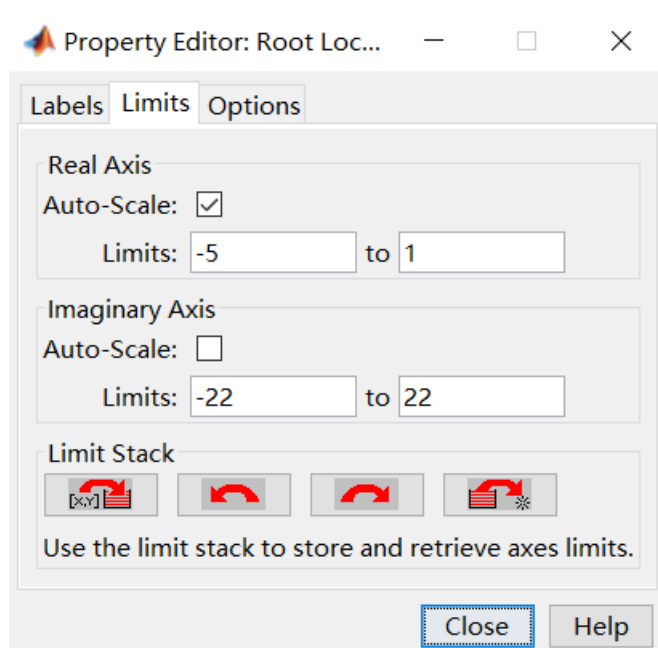
我们可以使用控制系统设计器来设计我们的滞后补偿器。要使控制系统设计器具有与上面所示相对应的补偿器参数，请单击“控制系统设计器”窗口顶部的“首选项”菜单。然后从 Options 选项卡中选择 Zero / pole / gain 参数化，如下所示。



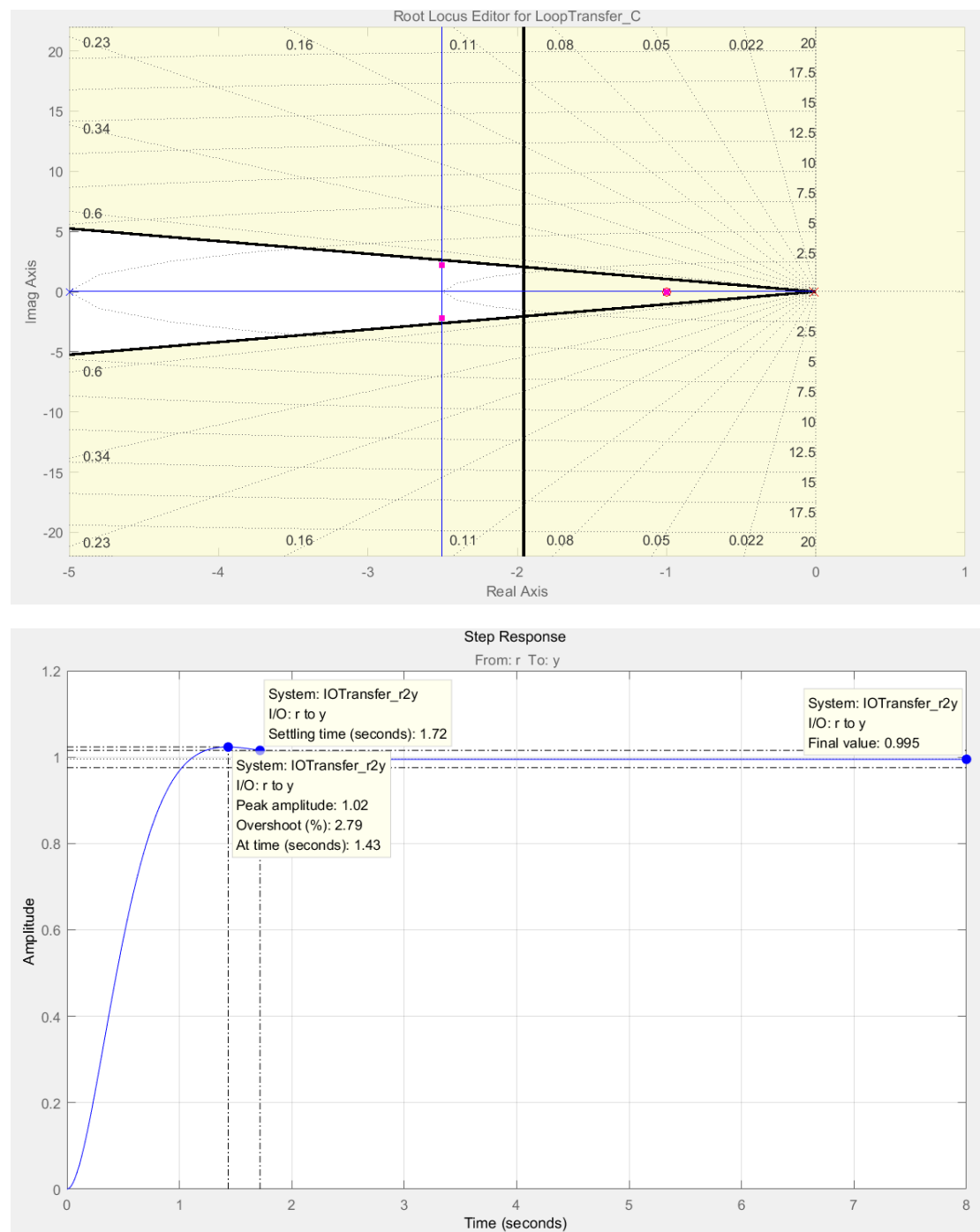
要添加滞后补偿器，请右键单击根轨迹图并选择“编辑补偿器”。要向补偿器添加极点零对，请在“补偿器编辑器”窗口中，右键单击“动力学”表，然后选择“添加极点/零点”>“滞后”。之后，输入 Real Zero 和 Real Pole 位置，如下图所示。



注意，滞后补偿器贡献的最大相位和它所处的频率被更新以匹配所选择的极点和零点。当您将闭环极移动足够的距离时，绘图的极限应自动更新。或者，您可以通过右键单击根轨迹并选择“属性”以打开“属性编辑器”来手动更改限制。然后，您可以单击“限制”选项卡并将虚轴限制更改为 $[-22, 22]$ ，例如，如下所示。



经过以上步骤可以设计出如下的根轨迹控制器：



完全符合设计要求：

- 沉降时间为 1.72 秒
- 超调量 2.79%
- 稳态误差为 0.5%

(3) 频域控制器

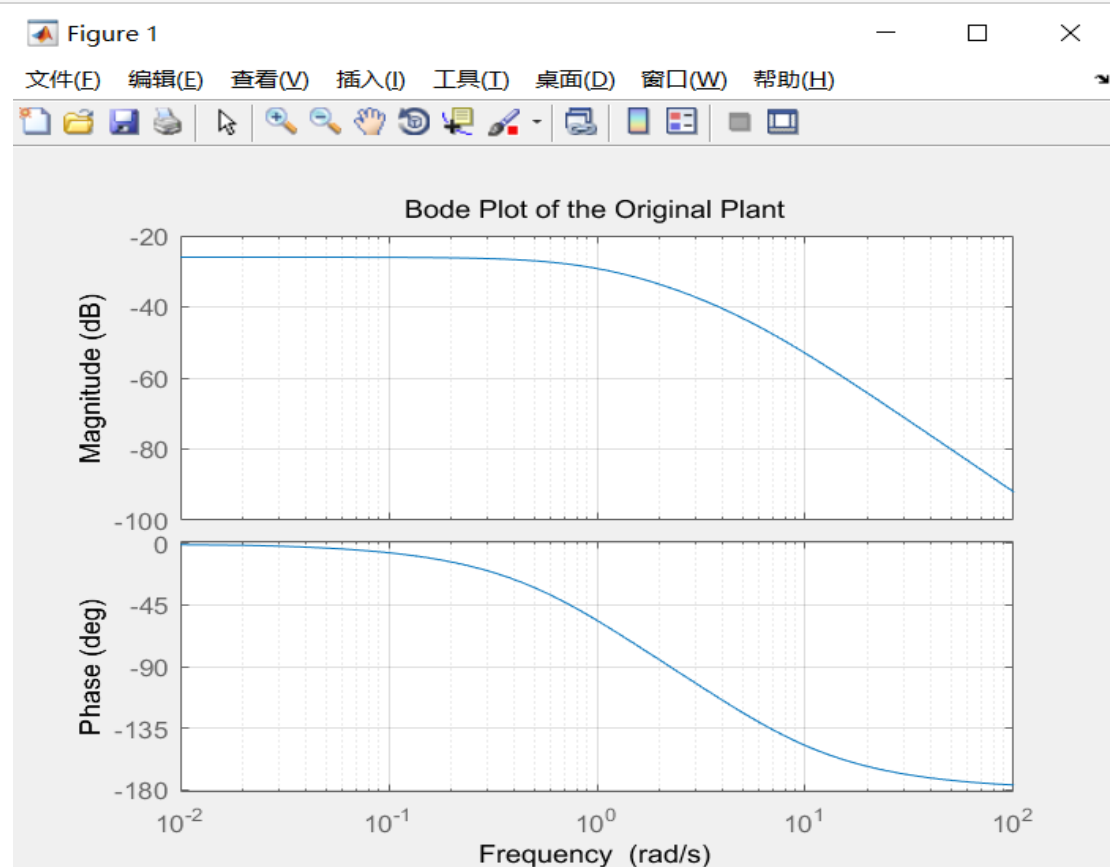
创建一个新的 [m 文件](#) 并输入以下命令：

```
J = 0.02;  
b = 0.1;  
K = 0.005;  
R = 1;  
L = 1;  
s = tf('s');  
P_motor = K / ((J * s + b) * (L * s + R) + K ^ 2);
```

绘制原始波特图：

将以下代码添加到 m 文件的末尾，并在 MATLAB 命令窗口中运行它。生成如下所示的波特图。

```
bode(P_motor)  
grid  
title('Bode Plot of the Original Plant')
```



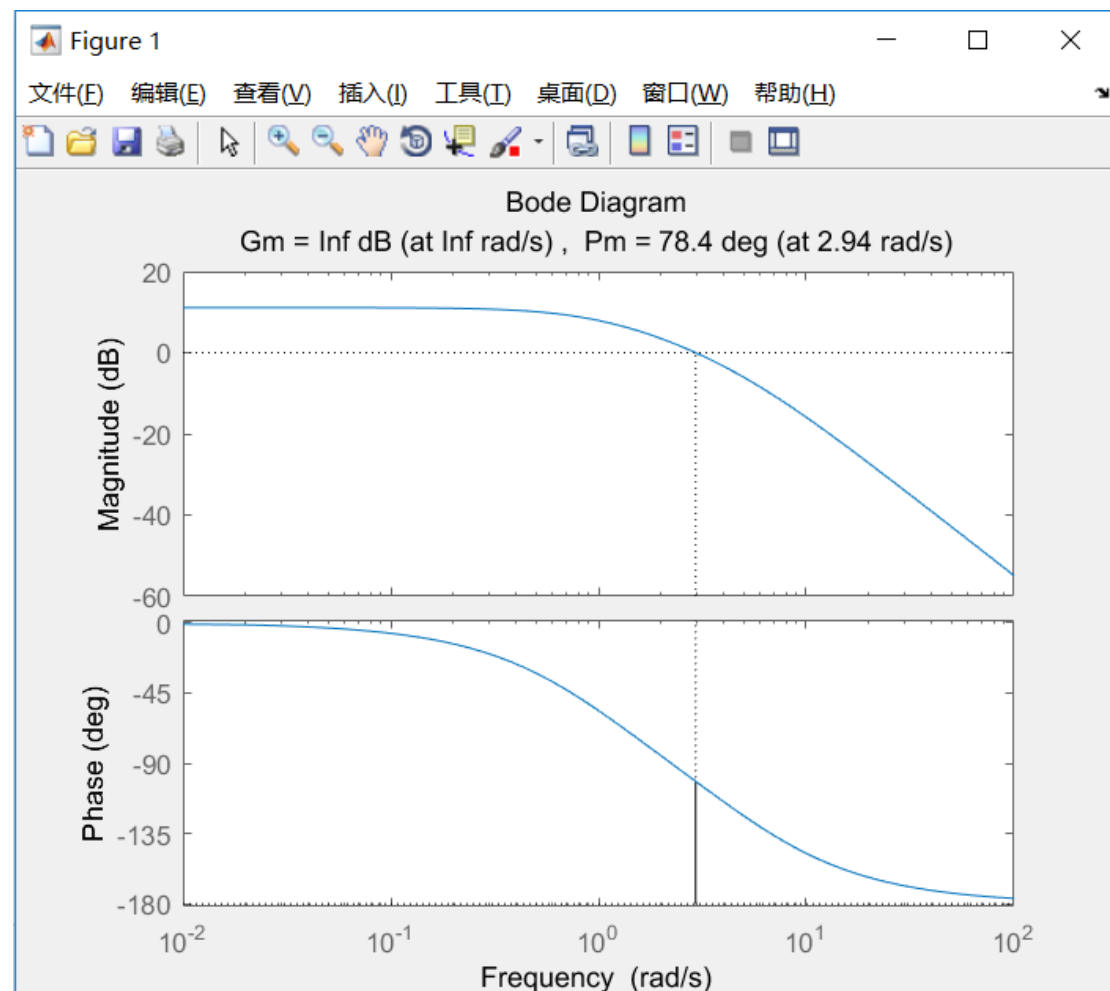
由上图可知此时系统的相位裕量是无限的。所以要对系统进行更改，在实现足够的相位裕量的同时增大系统的增益。

增加比例增益：

对于稳定裕度，60 度的相位裕度是足够的。从上面的波特图中，该相位裕度是在大约 10 弧度/秒的交叉频率下实现的。提高幅度图以使增益交叉频率出现在 10 弧度/秒所需的增益似乎约为 40 dB。

```
[mag, phase, w] = bode (P_motor, 10)
C = 72;

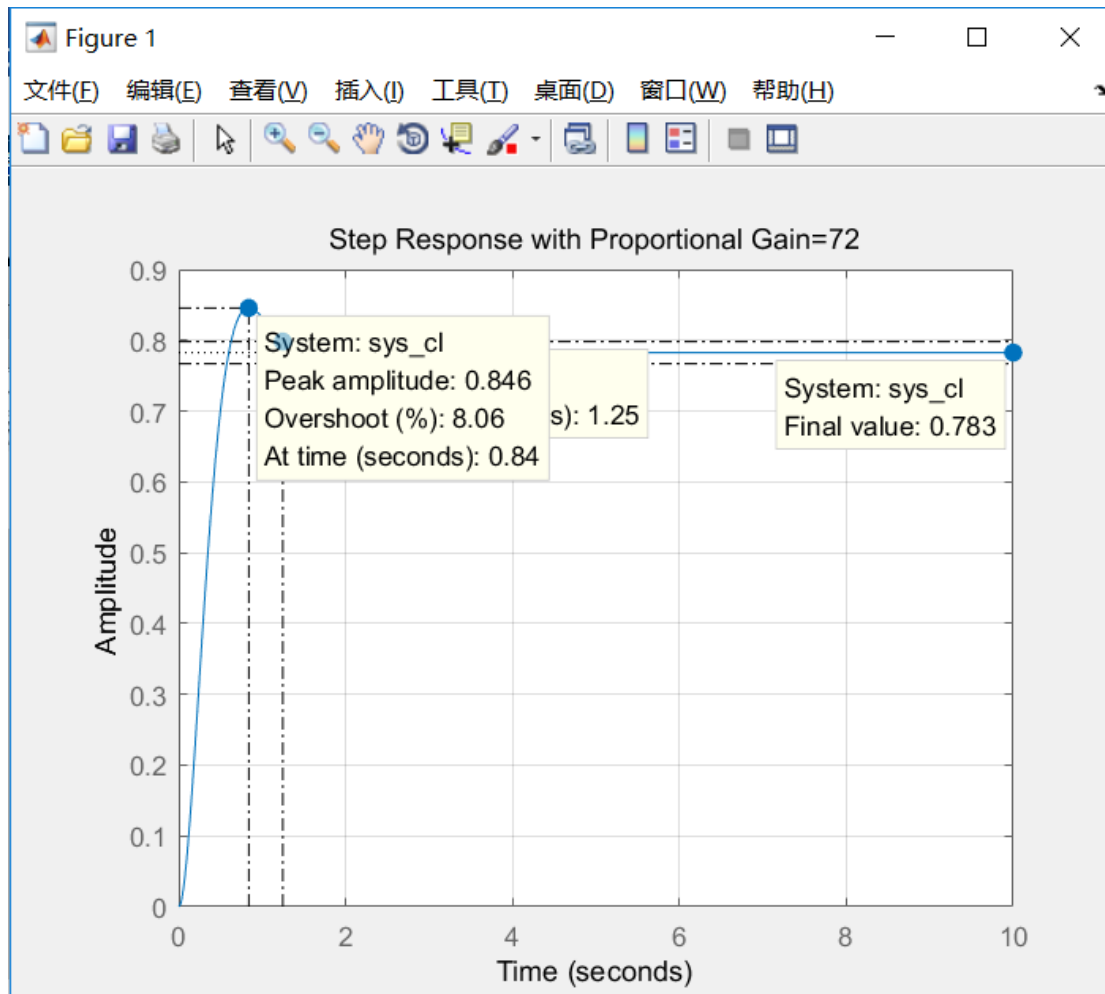
margin(C*P_motor);
```



(4) 绘制闭环响应：

将下列代码加到m文件末尾，运行

```
sys_cl = feedback(C*P_motor, 1);
t = 0:0.01:10;
step(sys_cl, t), grid
title('Step Response with Proportional Gain = 72')
```

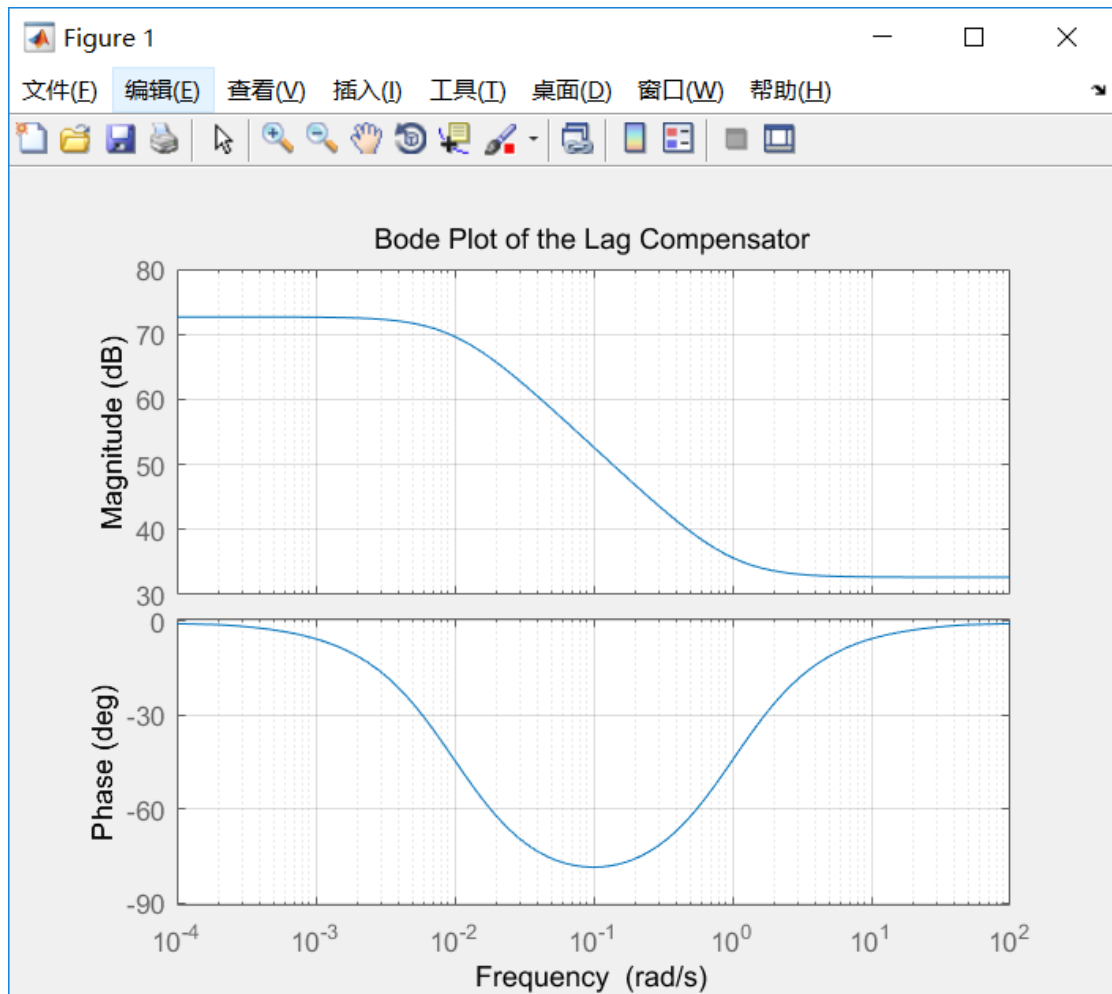



建立时间足够快，但过冲和稳态误差太高。可以通过降低增益来减小过冲，以实现更大的相位裕度，但这会导致稳态误差变得更大。滞后补偿器在这里可能会有所帮助，因为它可以降低增益交叉频率，以便在不降低系统增益的情况下增加相位裕度。

添加滞后补偿器：

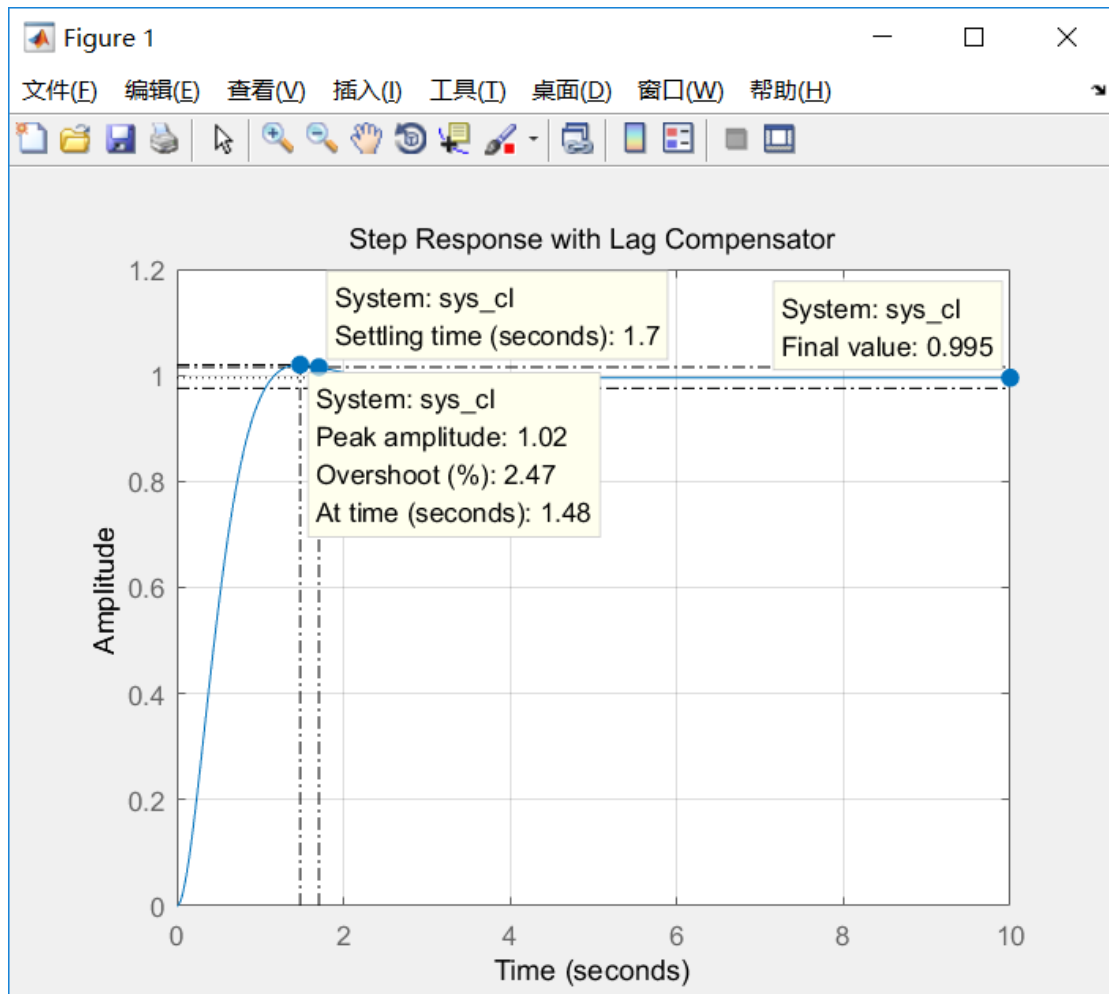
将下列代码加到m文件末尾，运行

```
C = 45*(s + 1)/(s + 0.01);
bode(C)
grid
title('Bode Plot of the Lag Compensator')
```



然后，可以通过修改 m 文件中的代码来观察生成的步骤响应，如下所示：

```
sys_cl = feedback(C*P_motor,1);  
t = 0:0.01:10;  
step(sys_cl,t), grid  
title('Step Response with Lag Compensator')
```



完全符合设计要求:

- 沉降时间为 1.7 秒
- 超调量 2.74%
- 稳态误差为 0.5%

五. 自我总结

不少课本上也有提到过 MATLAB，但是我还是不怎么熟练，通过这次自动控制原理课设让我更好地学会了如何去使用这个软件。每一个课程设计都是一个挑战！期间我学到了许多在实践中需注意的问题。包括如何根据课题制定方案，怎样思考问题，然后如何去收集整理资料，之后怎样组织材料撰写论文等等，这为我们即将面临的毕业设计等各类课题的设计打下了一定的基础。甚至在以后的工作和学习生活中也会受益。自动控制原理的知识在课堂上掌握的也不是很好，所以这次课程设计对于我来说真是难度不小啊。在很多人眼中为期一周的课程设计或许是一种煎熬，这是可以理解的，在这一周当中，我们不仅要完成这个课程设计，而且还要学习其他专业课。

对于 MATLAB 的学习，我们先从以前教材中翻看相关的内容，因为这些书上讲的比较精简易懂，看完之后便对 MATLAB 有了更深地了解并懂得了一些简单编程，接下来我再去图书馆借相关的书籍进行借鉴和参考，当要用什么功能时，就在书上翻看相应部分的内容，这样 MATLAB 就应用起来了。对于自动控制原理的相关知识，我重新翻看好几遍教材，特别是第六章作了详细地了解，对校正有了较好的认识之后才开始进行单位负反馈系统设计。通过这次自动化控制课程设计，我学到了许多新的知识以及锻炼了很多方面能力。首先这次课程设计的主要目的是要用 matlab 软件构建一个直流电动机调速的系统。将我们所学的理论知识通过软件模拟转化成现实中能用到的模型。要完成这个课程设计，就要对自动化控制理论的书本知识了解的足够透彻，这个系统集合了传递函数、滞后补偿器、伯特图、奈示图、PID 控制、频域响应控制、根轨迹控制等内容，是一个非常综合的集合体。

每一次课程设计都会学到不少东西，这次当然也不例外。不但对自动控制原理的知识巩固了，也加强了 MATLAB 这个强大软件使用的学习，这次课程设计终于顺利完成了，在设计中遇到了很多编程问题，最后在自己和老师同学相互协助下，终于迎刃而解了。在这个过程中，我首先参考了密歇根大学的电机调速系统，在它的基础上再加以扩展，但还是遇到了许多困难，比如说建模时的参数有所不同，这个模型就难以成立，对系统的参数分析也会出现问题，研究 PID 控制时，修改开环增益后，超调量和沉降时间指标就不能同时满足。校正设计时候，在试取值时需要对校正原理有较好的理解才能取出合适的参数，期间我也不是一次就成功，选了几次才选出比较合适的参数。这种不断尝试的经历让我们养成一种不断探索的科学研究精神，我想对于将来想从事技术行业的学生来说这是很重要的。研究根轨迹控制时，刚开始时对 matlab 软件不够熟悉，难以画出系统的根轨迹，后来经过翻阅书本资料以及观看教学视频，才慢慢画出了系统的根轨迹。这次课程设计中遇到了许多之前学习课本知识没有遇到的问题，理论设想的参数模型与实际上根本不相符的情况出现了几次，通过不断的调整参数，调整比例关系，增益关系达到所希望系统达到的参数。总体来说，这次课程设计让我自动控制理论知识和动手编程能力得到了提升，加强了我发现问题以及解决问题的能力。这个过程中我变得更加认真细致，心中有疑惑就和同学一起讨论，与老师交流，不会因为粗心的缘故使得编程出现问题，同时还补充了许多 matlab 的相关知识，熟练掌握这个软件，对待不满足要求的伯德图和根轨迹，也学会了添加滞后补偿器使得系统更加稳定。

附录:

建模代码 (Modeling.m)

```
J = 0.02;
b = 0.1;
K = 0.005;
R = 1;
L = 1;
s = tf('s');
P_motor = K / ((J * s + b)*(L * s + R) + K ^ 2) %传递函数
X=solve('0.02*X^2+0.12*X+0.1') %求解开环极点
motor_ss = ss(P_motor);
```

系统分析代码 (Analysis.m)

```
J = 0.02;
b = 0.1;
K = 0.005;
R = 1;
L = 1;
s =tf('s');
P_motor = K / ((J * s + b)*(L * s + R) + K ^ 2);%原始传递函数曲线
linearSystemAnalyzer('step',P_motor,0:0.1:5);
rP_motor=0.05/(s+1) %以 s = -1 处极点为主导极点的简化曲线
```

PID 控制器代码 (PID.m)

```
J = 0.02;
b = 0.1;
K = 0.005;
R = 1;
L = 1;
s =tf('s');
P_motor = K / ((J * s + b)*(L * s + R) + K ^ 2);
% Kp = 100; %比例控制
% C = pid(Kp);
% sys_cl = feedback(C * P_motor,1);
% t = 0:0.01:5;
% step(sys_cl,t)
% grid
% title('Step Response with Proportional Control')
Kp = 200;
Ki = 200;
Kd = 25;
C = pid(Kp,Ki, Kd);
```

```

sys_cl =feedback(C * P_motor,1);
step(sys_cl,[0:0.01:4])
title('PID Control with Large Ki and Large Kd')

```

根轨迹控制器代码 (RootLocus.m)

```

J = 0.02;
b = 0.1;
K = 0.005;
R = 1;
L = 1;
s =tf('s');
P_motor = K /((J * s + b)*(L * s + R)+ K ^ 2);
controlSystemDesigner('rlocus', P_motor)

```

频域控制器代码 (Frequency.m)

```

J = 0.02;
b = 0.1;
K = 0.005;
R = 1;
L = 1;
s =tf('s');
P_motor = K /((J * s + b)*(L * s + R)+ K ^ 2);
bode(P_motor) %绘制原始波特图
grid
title('Bode Plot of the Original Plant')
[mag, phase, w]=bode(P_motor, 10) %增加比例增益
C=72;
margin(C*P_motor);
sys_cl =feedback(C * P_motor,1); %绘制闭环响应
t=0:0.01:10;
step(sys_cl, t)
grid
title('Step Response with Proportional Gain=72')
C=43*(s+1)/(s+0.01); %添加滞后补偿器
bode(C)
grid
title('Bode Plot of the Lag Compensator')
sys_cl =feedback(C * P_motor,1);
t=0:0.01:10;
step(sys_cl, t)
grid
title('Step Response with Lag Compensator')

```