|  |  |  |
| --- | --- | --- |
| https://timgsa.baidu.com/timg?image&quality=80&size=b9999_10000&sec=1605027099768&di=051e3880f641da3d432b90a31148264e&imgtype=0&src=http%3A%2F%2Finews.gtimg.com%2Fnewsapp_match%2F0%2F10712584100%2F0.jpg | 自动控制原理实验报告 | |
| 院(系)：智能工程学院 | 学号：20354027 | 姓名：方桂安 |
| 日期：2022.9.2 | 实验名称：Matlab简介、传递函数及数学模型求取 | |

1. **实验目的**

1. 熟悉MATLAB实验环境，掌握MATLAB命令行窗口的基本操作。

2. 掌握MATLAB建立控制系统数学模型的命令及模型相互转换的方法。

1. **实验任务**

1. 编写M文件程序，建立控制系统的数学模型；

2. 编写M文件程序，实现控制系统不同数学模型之间的转换。

1. **实验设备**
2. 笔记本电脑——windows 11
3. MATLAB——R2021b
4. **实验原理**
5. **传递函数模型tf**

使用tf创建实值或复值的传递函数模型，或将动态系统模型转换为传递函数形式。

传递函数是线性时变系统的一种频域表示。例如，考虑一个连续时间SISO动态系统，由传递函数sys(s)=N(s)/D(s)表示，其中s=jw，N(s)和D(s)分别称为分子和分母多项式。tf模型对象可以代表连续时间或离散时间的SISO或MIMO传输函数。

你可以通过直接指定其系数，或通过将另一种类型的模型（如状态空间模型ss）转换为传递函数形式，来创建传递函数模型对象。

1. **零极点增益模型zpk**

使用zpk创建零极点增益模型，或将动态系统模型转换为零极点增益形式。

零极点增益模型是传递函数的因子化形式的表示。这里，z和p是实值或复值的零点和极点的向量，k是实值或复值的标量增益

你可以通过直接指定极点、零点和增益来创建一个零极点增益模型对象，或者通过将另一种类型的模型（如状态空间模型ss）转换为零极点增益形式。

1. **控制系统模型间的相互转换**

[num,den]=zp2tf(z,p,k) %零极点模型转换为传递函数模型

[z,p,k]=tf2zp(num,den) %传递函数模型转换为零极点模型

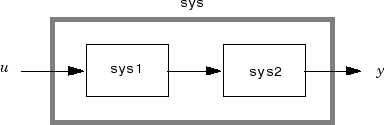
[r,p,k]=residue(num,den) %传递函数模型转换为部分分数展开式模型

[num,den]=residue(r,p,k) %部分分数展开式模型转换为传递函数模型

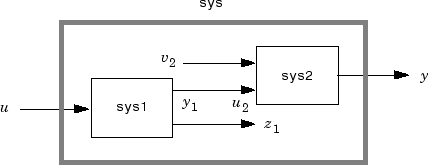
1. **控制系统模型连接后的等效传递函数**
2. 串联

series 串联两个模型对象。此函数接受任何类型的模型。这两个系统必须要么都是连续的，要么都是离散的，具有相同的采样时间。静态增益是中性的，可以指定为常规矩阵。

sys = series(sys1,sys2) 形成如下所示的基本串联。



sys = series(sys1,sys2,outputs1,inputs2) 形成更一般的串联连接。

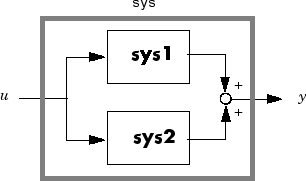


此命令相当于系统的乘法。

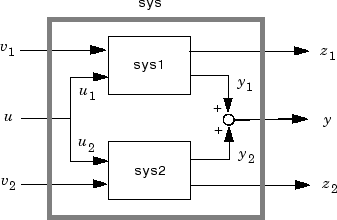
1. 并联

parallel 并行 连接两个模型对象。此函数接受任何类型的模型。这两个系统必须要么都是连续的，要么都是离散的，具有相同的采样时间。静态增益是中性的，可以指定为常规矩阵。

sys = parallel(sys1,sys2) 形成下图所示 的基本并联。



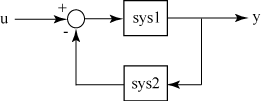
sys = parallel(sys1,sys2,inp1,inp2,out1,out2) 形成更一般的并联连接，如下图所示。



此命令相当于系统的加法。

1. 反馈

sys = feedback(sys1,sys2) 返回一个模型对象sys，用于模型对象的负反馈互连sys1,sys2。



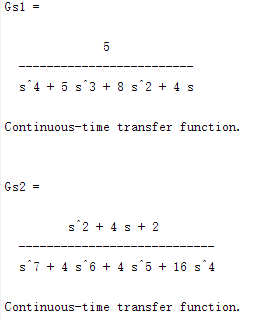
从图中可以看出，闭环模型sys以 u作为输入向量，以y作为输出向量。两个模型sys1和sys2必​​须是连续的或离散的，具有相同的采样时间。

1. **实验步骤**

任务1：建立控制系统的传递函数模型



结果输出：



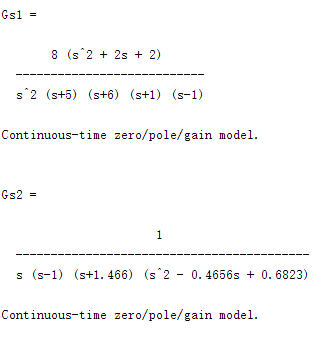
代码：

1. s = tf('s')*;*
2. Gs1 = 5/(s\*(s+1)\*(s^2+4\*s+4)) *;*
3. Gs2 = (s^2+4\*s+2)/(s^3\*(s^2+4)\*(s^2+4\*s))*;*

任务2：建立控制系统的零极点增益模型



结果输出：



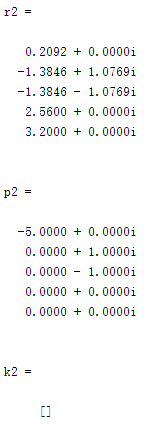
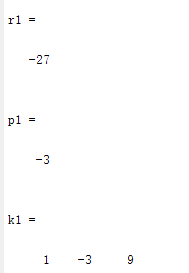
代码：

1. z1 = [-1+1i -1-1i]*;*
2. p1 = [0 0 -5 -6 1 -1]*;*
3. k1 = [8]*;*
4. Gs1 = zpk(z1,p1,k1)*;*
5. num = [1]*;*
6. den = [1 1 0 1]*;*
7. sys1 =tf(num,den)*;*
8. z2 = []*;*
9. p2 = [0 1]*;*
10. k2 = [1]*;*
11. sys2 = zpk(z2,p2,k2)*;*
12. Gs2 = series(sys1,sys2)*;*

任务3：将系统传递函数转化为部分分式展开式



结果输出：

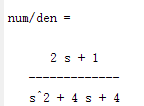


代码：

1. z1 = [0*;0;0];*
2. p1 = [-3]*;*
3. k1 = [1]*;*
4. [num1,den1] = zp2tf(z1,p1,k1)*;*
5. [r1,p1,k1] = residue(num1,den1)*;*
6. z2 = [-1+1i*;-1-1i];*
7. p2 = [0 0 -5 1i -1i]*;*
8. k2 = [8]*;*
9. [num2,den2] = zp2tf(z2,p2,k2)*;*
10. [r2,p2,k2] = residue(num2,den2)*;*

任务4：已知系统，求其单位负反馈闭环传递函数

结果输出：



代码：

1. num = [2 1];
2. den = [1 2 3];
3. [numc,denc] = cloop(num,den);
4. printsys(numc,denc)
5. **实验结果及心得**

本次实验内容相对简单，结果均展示在实验过程中。通过本次实验以及对matlab中的Control System Toolbox的文档的阅读，我熟悉并掌握了构建简单系统模型的编程方法，使自动控制原理的知识得以应用。