

# 《自动控制原理（F）》实验指导书

班级：\_\_\_\_\_ 姓名：\_\_\_\_\_ 指导老师：\_\_\_\_\_

## 目录

<b>实验一</b>	<b>控制系统的数学模型</b>	<b>1</b>
一、	实验目的 . . . . .	1
二、	实验原理 . . . . .	1
	（一） 用 MATLAB 建立传递函数模型 . . . . .	1
	（二） 模型的连接 . . . . .	2
	（三） 典型环节的模拟方法及动态特性 . . . . .	2
三、	实验内容 . . . . .	2
四、	实验报告要求 . . . . .	3
<b>实验二</b>	<b>线性系统的时域分析</b>	<b>3</b>
一、	实验目的 . . . . .	3
二、	实验原理 . . . . .	4
三、	实验内容 . . . . .	6
四、	实验报告要求 . . . . .	6
<b>实验三</b>	<b>控制系统的根轨迹分析</b>	<b>7</b>
一、	实验目的 . . . . .	7
二、	实验原理 . . . . .	7
三、	实验内容 . . . . .	7
四、	实验报告要求 . . . . .	8
<b>实验四</b>	<b>系统开环频率特性的绘制及稳定性分析</b>	<b>8</b>
一、	实验目的 . . . . .	8
二、	实验原理 . . . . .	8
三、	实验内容 . . . . .	9
四、	实验报告要求 . . . . .	9

## 实验一 控制系统的数学模型

### 一、实验目的

1. 熟悉 MATLAB 软件的实验环境;
2. 掌握用 MATLAB 创建各种控制系统模型;
3. 掌握多环节串联、并联、反馈连接时整体传递函数的求取方法;
4. 掌握典型环节的软件仿真方法, 研究典型环节的动态特性.

### 二、实验原理

#### (一) 用 MATLAB 建立传递函数模型

##### 1. 多项式模型

线性系统的传递函数模型可一般地表示为:

$$G(s) = \frac{b_1 s^m + b_2 s^{m-1} + \cdots + b_m s + b_{m+1}}{s^n + a_1 s^{n-1} + \cdots + a_{n-1} s + a_n} \quad n \geq m \quad (1)$$

将系统的分子和分母多项式的系数按降幂的方式以向量的形式输入给两个变量 *num* 和 *den*, 就可以轻易地将传递函数模型输入到 MATLAB 环境中. 命令格式为:

$$\text{num} = [b_1, b_2, \cdots, b_m, b_{m+1}]; \quad (2)$$

$$\text{den} = [1, a_1, a_2, \cdots, a_{n-1}, a_n]; \quad (3)$$

在 MATLAB 控制系统工具箱中, 定义了 *tf()* 函数, 它可由传递函数分子分母给出的变量构造出单个的传递函数对象. 从而使得系统模型的输入和处理更加方便.

该函数的调用格式为:

$$G = \text{tf}(\text{num}, \text{den}); \quad (4)$$

##### 2. 零极点模型

线性系统的传递函数还可以写成极点形式:

$$G(s) = K \frac{(s + z_1)(s + z_2) \cdots (s + z_m)}{(s + p_1)(s + p_2) \cdots (s + p_n)} \quad (5)$$

将系统增益、零点和极点以向量的形式输入给三个变量 *KGain*、*Z* 和 *P*, 就可以将系统的零极点模型输入到 MATLAB 工作空间中, 命令格式为

$$Z = [-z_1; -z_2; \cdots; -z_m]; \quad (6)$$

$$P = [-p_1; -p_2; \cdots; -p_n]; \quad (7)$$

在 MATLAB 控制工具箱中, 定义了 *zpk()* 函数, 由它可通过以上三个 MATLAB 变量构造出零极点对象, 用于简单地表述零极点模型. 该函数的调用格式为:

$$G = \text{zpk}(Z, P, K) \quad (8)$$

##### 3. 多项式模型与零极点模型的转换

在控制系统工具箱中, 可以由 *zpk()* 函数立即将给定的 LTI 对象 *G* 转换成等效的零极点对象 *G1*. 该函数的调用格式为:

$$G1 = \text{zpk}(G) \quad (9)$$

$$G2 = \text{tf}(G1) \quad (10)$$

## (二) 模型的连接

### 1. 串联: series

格式:

$$[\text{num}, \text{den}] = \text{series}(\text{num1}, \text{den1}, \text{num2}, \text{den2}) \quad (11)$$

% 将串联连接的传递函数进行相乘.

### 2. 并联: parallel

格式:

$$[\text{num}, \text{den}] = \text{parallel}(\text{num1}, \text{den1}, \text{num2}, \text{den2}, \text{out1}, \text{out2}) \quad (12)$$

% 将并联连接的传递函数进行相加. out1 和 out2 分别指定要作相加的输出端编号.

### 3. 反馈: feedback

控制系统工具箱中提供了 `feedback()` 函数, 用来求取反馈连接下总的系统模型, 该函数调用格式如下:

$$G = \text{feedback}(G1, G2, \text{sign}); \quad (13)$$

其中变量 `sign` 用来表示正反馈或负反馈结构, 若 `sign=-1` 表示负反馈系统的模型, 若省略 `sign` 变量, 则仍将表示负反馈结构. `G1` 和 `G2` 分别表示前向模型和反馈模型的 LTI (线性时不变) 对象.

## (三) 典型环节的模拟方法及动态特性

### 1. 典型环节的传递函数

比例环节  $G(s) = \frac{K_2}{K_1}$ 、惯性环节  $G(s) = \frac{1}{Ts+1}$ 、积分环节  $G(s) = \frac{1}{Ts}$ 、微分环节  $G(s) = Ts$ 、比例微分环节  $G(s) = k(Ts+1)$ 、比例积分环节  $G(s) = k(1 + \frac{1}{Ts})$

### 2. 单位阶跃响应

`step()` 函数可以计算连续系统单位阶跃响应曲线 `step(sys)` `step(sys,t)` `step(num,den)` 函数在当前图形窗口中直接绘制出系统单位阶跃响应曲线, 对象 `sys` 可以由 `tf()`, `zpk()` 函数中任何一个建立的系统模型. 第二种格式中的 `t` 可以指定一个发展终止时间, 也可以设置为一个时间矢量 (如 `t=0:dt:tfinal`, 即 `dt` 是步长, `tfinal` 是终止时刻).

### 3. 子图命令

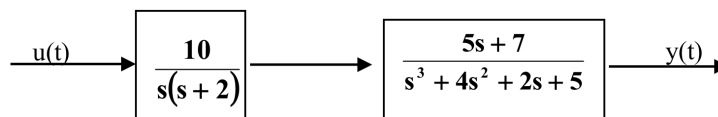
MATLAB 允许将一个图形窗口分成多个子窗口, 分别显示多个图形, 这就要用到 `subplot()` 函数, 其调用格式为:

$$\text{subplot}(m,n,k) \quad (14)$$

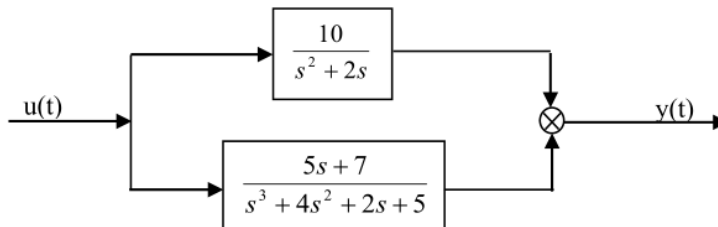
该函数将把一个图形窗口分割成  $m \times n$  个子绘图区域,  $m$  为行数,  $n$  为列数, 用户可以通过参数  $k$  调用各子绘图区域进行操作, 子图区域编号为按行从左至右编号. 对一个子图进行的图形设置不会影响到其它子图, 而且允许各子图具有不同的坐标系. 例如, `subplot(4,3,6)` 则表示将窗口分割成  $4 \times 3$  个部分. 在第 6 部分上绘制图形. MATLAB 最多允许  $9 \times 9$  的分割.

## 三、实验内容

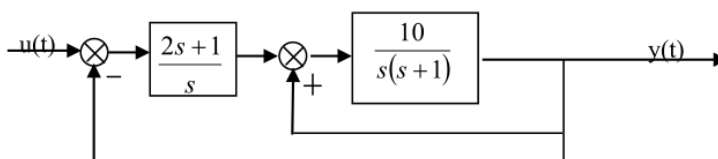
1. 用 MATLAB 建立系统  $G(s) = \frac{s+2}{s^2+5s+10}$  的多项式模型.
2. 用 MATLAB 建立系统  $G(s) = \frac{10(s+1)}{(s+2)(s+5)(s+10)}$  零极点模型和多项式模型.
3. 系统结构图如下所示, 求其多项式传递函数模型.



4. 系统结构图如下所示，求其多项式传递函数模型。



5. 系统结构图如下所示，求其多项式传递函数模型。



6. 用 Matlab 产生下列系统的传递函数的多项式传递函数模型、零极点模型。

$$G(s) = \frac{s^4 + 3s^3 + 2s^2 + s + 1}{s^5 + 4s^4 + 3s^3 + 2s^2 + 3s + 2} \quad (15)$$

7. 分别构造比例环节  $G(s) = \frac{R_2}{R_1}$ 、惯性环节  $G(s) = \frac{1}{Ts+1}$ 、积分环节  $G(s) = \frac{1}{Ts}$ 、微分环节  $G(s) = Ts$ 、比例微分环节  $G(s) = k(Ts+1)$ 、比例积分环节  $G(s) = k(1 + \frac{1}{Ts})$  的传递函数，求其单位阶跃响应，对比有何不同。

#### 四、实验报告要求

1. 写明实验目的和实验原理。实验原理中简要说明求取传递函数的途径和采用的语句或函数。
2. 针对典型环节的软件仿真，编写程序，记录单位阶跃响应并加以分析对比各环节的特性，要求各幅图都要画在同一界面内，便于比较。
3. 在实验过程和结果中，要求按项目写清楚自定的传递函数，从屏幕上复制程序和运行结果，打印粘贴在报告上。

## 实验二 线性系统的时域分析

### 一、实验目的

1. 用 MATLAB 判断控制系统的稳定性；
2. 掌握准确读取动态特性指标的方法；
3. 定量分析二阶系统的阻尼比  $\xi$  和自然振荡角频率  $\omega_n$  对系统动态性能的影响；
4. 掌握使用 Simulink 仿真环境进行控制系统稳态误差分析的方法。
5. 研究系统不同典型输入下信号作用下，稳态误差的变化。

## 二、实验原理

### 1. 线性系统稳定性分析

线性系统稳定的充要条件是系统的特征根均位于  $S$  平面的左半部分。系统的零极点模型可以直接被用来判断系统的稳定性。另外，MATLAB 语言中提供了有关多项式的操作函数，也可以用于系统的分析和计算。直接求特征多项式的根。

设  $p$  为特征多项式的系数向量，则 MATLAB 函数 `roots()` 可以直接求出方程  $p = 0$  在复数范围内的解  $v$ ，该函数的调用格式为：

$$v = \text{roots}(p) \quad (16)$$

### 2. 准确读出其动态性能指标

函数 `step()` 来直接求取线性系统的阶跃响应。在曲线中空白区域，单击鼠标右键，在快捷菜单中选择 `characteristics`，包含：`peak response`(峰值)；`setting time`(调节时间)；`rise time`(上升时间)；`steady state`(稳态值)；在相应位置出现相应点，用鼠标单击后，相应性能值就显示出来。



### 3. 分析 $\xi$ 和 $\omega_n$ 对系统阶跃响应的影响

用 `step()` 绘制不同阻尼比和自然频率的系统阶跃响应，比较结果得出结论。

```
for i=1:6
    den=[1,2*zeta(i)*wn,wn^2];
    step(num,den,2.5)
end
```

### 4. Simulink 建模方法

在一些实际应用中，如果系统的结构过于复杂，不适合用前面介绍的方法建模。在这种情况下，功能完善的 Simulink 程序可以用来建立新的数学模型。Simulink 是由 Math Works 软件公司 1990 年为 MATLAB 提供的新的控制系统模型图形输入仿真工具。它具有两个显著的功能：Simul (仿真) 与 Link (连接)，亦即可以利用鼠标在模型窗口上“画”出所需的控制系统模型。然后利用 SIMULINK 提供的功能来对系统进行仿真或线性化分析。与 MATLAB 中逐行输入命令相比，这样输入更容易，分析更直观。下面简单介绍 SIMULINK 建立系统模型的基本步骤：

- (1) SIMULINK 的启动：在 MATLAB 命令窗口的工具栏中单击按钮  或者在命令提示符 `>>` 下键入 Simulink 命令，回车后即可启动 Simulink 程序。启动后软件自动打开 Simulink 模型库窗口，如图 1 所示。这一模型库中含有许多子模型库，如 Sources (输入源模块库)、Sinks (输出显示模块库)、Nonlinear (非线性环节) 等。若想建立一个控制系统结构框图，则应该选择 File|New 菜单中的 Model 选项，或选择工具栏上 new Model  按钮。
- (2) 画出系统的各个模块：打开相应的子模块库，选择所需要的元素，用鼠标左键点中后拖到模型编辑窗口的合适位置。
- (3) 给出各个模块参数：由于选中的各个模块只包含默认的参数，如默认的传递函数模型为  $\frac{1}{s+1}$  的简单格式，必须通过修改得到实际的模块参数。要修改模块的参数，可以用鼠标双击该模块图标，则会出现一个相应对话框，提示用户修改模块参数。
- (4) 画出连接线：当所有的模块都画出来之后，可以再画出模块间所需要的连线，构成完整的系统。模块间连线的画法很简单，只需要用鼠标点按起始模块的输出端 (三角符号)，再拖动鼠标，到终止模块的输入端释放鼠标键，系统会自动地在两个模块间画出带箭头的连线。若需要从连线中引出节点，可在鼠标点击起始节点时按住 `Ctrl` 键，再将鼠标拖动到目的模块。

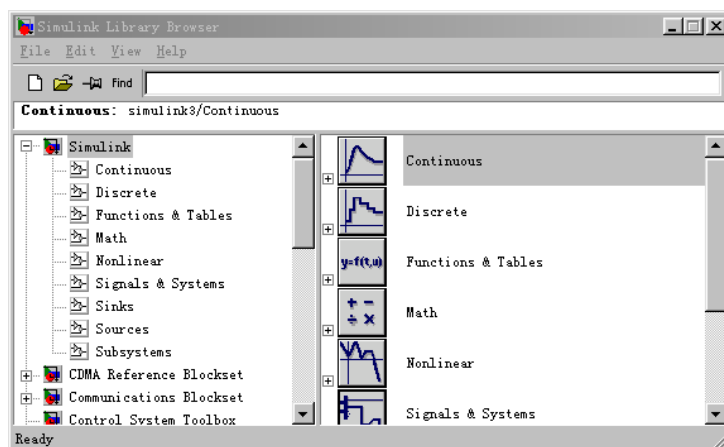


图 1: Simulink 模型库

- (5) 指定输入和输出端子：在 Simulink 下允许有两类输入输出信号，第一类是仿真信号，可从 Sources (输入源模块库) 图标中取出相应的输入信号端子，从 Sinks (输出显示模块库) 图标中取出相应输出端子即可。第二类是要提取系统线性模型，则需打开 Connection (连接模块库) 图标，从中选取相应的输入输出端子。

例 1 典型二阶系统的结构图如图 2 所示。用 Simulink 对系统进行仿真分析。

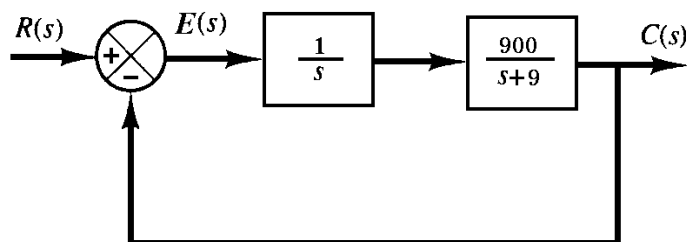


图 2: 典型二阶系统结构图

按前面步骤，启动 Simulink 并打开一个空白的模型编辑窗口。

#### 1. 画出所需模块，并给出正确的参数：

- 在 Sources 子模块库中选中阶跃输入 (step) 图标，将其拖入编辑窗口，并用鼠标左键双击该图标，打开参数设定的对话框，将参数 step time (阶跃时刻) 设为 0。
- 在 Math (数学) 子模块库中选中加法器 (sum) 图标，拖到编辑窗口中，并双击该图标将参数 List of signs (符号列表) 设为 | + - (表示输入为正，反馈为负)。
- 在 Continuous (连续) 子模块库中、选积分器 (Integrator) 和传递函数 (Transfer Fcn) 图标拖到编辑窗口中，并将传递函数分子 (Numerator) 改为 (900)，分母 (Denominator) 改为 (1,9)。
- 在 Sinks (输出) 子模块库中选择 Scope (示波器) 和 Out1 (输出端口模块) 图标并将之拖到编辑窗口中。

#### 2. 将画出的所有模块按图 3 用鼠标连接起来，构成一个原系统的框图描述如图 3 所示。

#### 3. 选择仿真算法和仿真控制参数，启动仿真过程。

- 在编辑窗口中点击 Simulation|Simulation parameters 菜单，会出现一个参数对话框，在 solver 模板中设置响应的仿真范围 StartTime (开始时间) 和 StopTime (终止时间)，仿真步长范围 Maximum step size (最大步长) 和 Minimum step size (最小步长)。对于本例，StopTime 可设置为 2。最后点击 Simulation|Start 菜单或点击相应的热键启动仿真。双击示波器，在弹出的图形上会“实时地”显示出仿真结果。

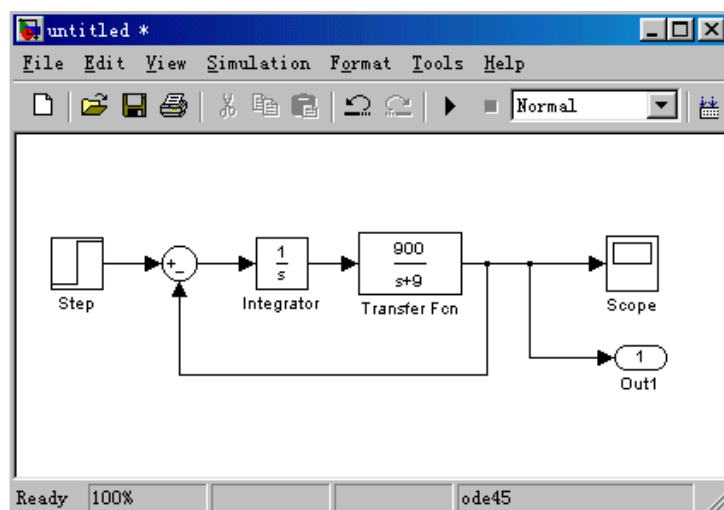


图 3: 二阶系统的 Simulink 实现

### 三、实验内容

- 已知系统开环传递函数为  $G(s) = \frac{100(s+2)}{s(s+1)(s+20)}$ ，试判断闭环系统的稳定性。
- 若已知单位负反馈开环传递函数为  $G(s) = \frac{100}{s^2+5s}$ ，试作出其单位阶跃响应曲线，准确读出其动态性能指标。
- 分析典型二阶系统  $G(s) = \frac{\omega_n^2}{s^2+2\xi\omega_n s+\omega_n^2}$  的  $\xi$  和  $\omega_n$  变化时，对系统的阶跃响应的影响。
  - 选取  $\omega_n = 10 \text{ rad/s}$  不变， $\xi$  分别为 0, 0.25, 0.5, 0.7, 1, 2 这 6 个不同参数进行分析。
  - 选取  $\xi = 0.5$  不变， $\omega_n$  分别为 10, 100 这 2 个不同参数进行分析。
- 若已知单位负反馈开环传递函数为  $G(s) = \frac{10k}{s(0.1s+1)}$ 。
  - 分别作出  $k = 1$  和  $k = 10$  时，系统单位阶跃响应曲线并求单位阶跃响应稳态误差。
  - 分别作出  $k = 0.1$  和  $k = 1$  时，系统单位斜坡响应曲线并求单位斜坡响应稳态误差。

### 四、实验报告要求

- 用 roots() 函数求根，根据根的实部判断系统稳定性；
- 编程后绘出单位阶跃响应曲线，在图上直接读，记录相关数据和响应曲线；
- 编程，记录响应曲线，比较结果得出结论；
- 用 Simulink 仿真后记录响应曲线，比较结果得出结论。

## 实验三 控制系统的根轨迹分析

### 一、实验目的

1. 熟练掌握使用 MATLAB 绘制根轨迹图形的方法;
2. 进一步加深对根轨迹图的了解;
3. 掌握利用所绘制根轨迹图形分析系统性能的方法.

### 二、实验原理

控制系统工具箱中提供了 `rlocus()` 函数, 可以用来绘制给定系统的根轨迹.

$$\text{rlocus}(\text{num}, \text{den}) \quad (17)$$

控系统工具箱中还有一个 `rlocfind()` 函数, 该函数允许用户求取根轨迹上指定点处的开环增益值, 并将该增益下所有的闭环极点显示出来. 这个函数的调用格式为:

$$[K, P] = \text{rlocfind}(G) \quad (18)$$

这个函数运行后, 图形窗口中会出现要求用户使用鼠标定位的提示, 用户可以用鼠标左键点击所关心的根轨迹上的点. 这样将返回一个  $K$  变量, 该变量为所选择点对应的开环增益, 同时返回的  $P$  变量则为该增益下所有的闭环极点位置. 此外, 该函数还将自动地将该增益下所有的闭环极点直接在根轨迹曲线上显示出来.

不用 `rlocfind()` 命令, 直接用鼠标左键点击所要选择的点, 根轨迹图上就会出现该点的说明, 包括相应增益 (Gain)、极点位置 (Pole)、阻尼参数 (Damping)、超调量 (Overshoot)、自然频率 (Frequency).

### 三、实验内容

1. 已知系统开环传递函数为  $G(s) = \frac{K}{s(0.2s+1)(0.5s+1)}$ 
  - (1) 绘制并记录根轨迹;
  - (2) 确定根轨迹的分离点与相应的根轨迹增益;
  - (3) 确定临界稳定时的根轨迹增益.
2. 设控制系统如图 4 所示, 其中  $G_c(s)$  为改善系统性能而加入的校正装置. 若  $G_c(s)$  可从  $k_t s$ 、 $k_a s^2$  和  $\frac{k_a s^2}{s+20}$  三种传递函数中任选一种, 分别作出相应的根轨迹图, 分析说明应选择哪种校正装置.

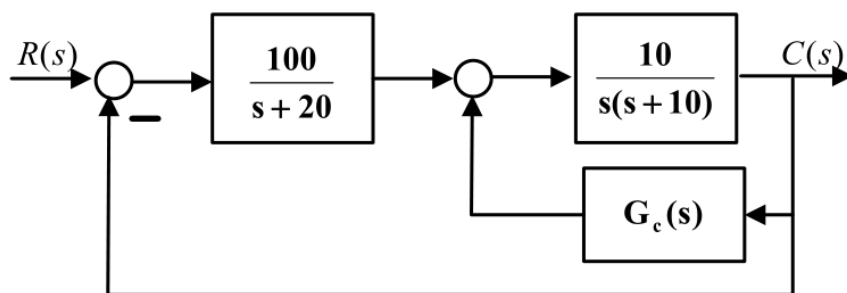


图 4: 控制系统 2



## 四、实验报告要求

1. 根据根轨迹, 分析闭环极点在  $S$  平面上的位置与系统动态性能的关系.
2. 根据根轨迹, 确定系统 2 合适校正装置.
3. 在实验过程和结果中, 要求按项目写清楚自定的传递函数, 从屏幕上复制程序和运行结果, 打印粘贴在报告上.

## 实验四 系统开环频率特性的绘制及稳定性分析

### 一、实验目的

1. 加深了解系统及元件频率特性的物理概念;
2. 进一步加深对 Bode 图及 Nyquist 曲线的了解;
3. 熟练掌握用 MATLAB 分析系统频率特性的方法.

### 二、实验原理

#### 1. 用 MATLAB 作奈魁斯特图

控制系统工具箱中提供了一个 MATLAB 函数 `nyquist()`, 该函数可以用来直接求解 Nyquist 阵列或绘制奈氏图. 当命令中不包含左端返回变量时, `nyquist()` 函数仅在屏幕上产生奈氏图, 命令调用格式为:

$$\text{nyquist}(\text{num}, \text{den}) \quad (19)$$

$$\text{nyquist}(\text{num}, \text{den}, w) \quad (20)$$

或者

$$\text{nyquist}(G) \quad (21)$$

$$\text{nyquist}(G, w) \quad (22)$$

如果用户给出频率向量  $w$ , 则  $w$  包含了要分析的以弧度/秒表示的诸频率点. 在这些频率点上, 将对系统的频率响应进行计算, 若没有指定的  $w$  向量, 则该函数自动选择频率向量进行计算. 当命令中包含了左端的返回变量时, 即

$$[\text{re}, \text{im}, w] = \text{nyquist}(G) \quad (23)$$

或

$$[\text{re}, \text{im}, w] = \text{nyquist}(G, w) \quad (24)$$

函数运行后不在屏幕上产生图形, 而是将计算结果返回到矩阵  $re$ 、 $im$  和  $w$  中. 矩阵  $re$  和  $im$  分别表示频率响应的实部和虚部, 它们都是由向量  $w$  中指定的频率点计算得到的.

在运行结果中,  $w$  数列的每一个值分别对应  $re$ 、 $im$  数列的每一个值.

#### 2. 用 MATLAB 作伯德图

控制系统工具箱里提供的 `bode()` 函数可以直接求取、绘制给定线性系统的伯德图.

当命令不包含左端返回变量时, 函数运行后会在屏幕上直接画出伯德图. 如果命令表达式的左端含有返回变量, `bode()` 函数计算出的幅值和相角将返回到相应的矩阵中, 这时屏幕上不显示频率响应图. 命令的调用格式为:

$$[\text{mag}, \text{phase}, w] = \text{bode}(\text{num}, \text{den}) \quad (25)$$

$$[mag, phase, w] = bode(num, den, w) \quad (26)$$

或

$$[mag, phase, w] = bode(G) \quad (27)$$

$$[mag, phase, w] = bode(G, w) \quad (28)$$

矩阵  $mag$ 、 $phase$  包含系统频率响应的幅值和相角，这些幅值和相角是在用户指定的频率点上计算得到的。用户如果不指定频率  $w$ ，MATLAB 会自动产生  $w$  向量，并根据  $w$  向量上各点计算幅值和相角。这时的相角是以度来表示的，幅值为增益值，在画伯德图时要转换成分贝值，因为分贝是作幅频图时常用单位。

### 3. 用 MATLAB 求取稳定裕量

由 MATLAB 里 `bode()` 函数绘制的伯德图也可以采用游动鼠标法求取系统的幅值裕量和相位裕量。

此外，控制系统工具箱中提供了 `margin()` 函数来求取给定线性系统幅值裕量和相位裕量，该函数可以由下面格式来调用：

$$[Gm, Pm, Wcg, Wcp] = margin(G); \quad (29)$$

可以看出，幅值裕量与相位裕量可以由 LTI 对象  $G$  求出，返回的变量对  $(Gm, Wcg)$  为幅值裕量的值与相应的相角穿越频率，而  $(Pm, Wcp)$  则为相位裕量的值与相应的幅值穿越频率。若得出的裕量为无穷大，则其值为 `inf`，这时相应的频率值为 `NaN` (表示非数值)，`inf` 和 `NaN` 均为 MATLAB 软件保留的常数。

如果已知系统的频率响应数据，我们还可以由下面的格式调用此函数。

$$[Gm, Pm, Wcg, Wcp] = margin(mag, phase, w); \quad (30)$$

其中  $(mag, phase, w)$  分别为频率响应的幅值、相位与频率向量。

## 三、实验内容

- 已知系统开环传递函数为  $G(s) = \frac{K}{s^2 + 10s + 500}$ 
  - 取  $K = 500$  做出系统开环对数幅频、相频特性，求出相应的频域性能指标；改变  $K(10, 500, 1000)$ ，作 Bode 图，比较分析  $K$  变化对对数幅频、相频特性曲线的影响。
  - 绘制系统开环幅相曲线，并用奈氏判据判断系统的稳定性。 $K(10, 500, 1000)$ ，分析  $K$  对开环幅相曲线的影响。
- 已知系统开环传递函数为  $G(s) = \frac{1}{s^\gamma(s+1)(s+2)}$ ，试分别绘制并记录  $\gamma = 1, 2, 3, 4$  时系统粗略的开环幅相曲线，比较分析  $\gamma$  改变对系统幅相曲线的影响。
- 已知单位反馈系统  $G(s) = \frac{K}{(s+1)^3}$ ，设  $K$  分别为 4 和 10，试确定系统的稳定裕度。

## 四、实验报告要求

- 记录用 MATLAB 绘制的 Bode 图。
- 记录绘出的 Nyquist 曲线，并分析系统稳定性。
- 在实验过程和结果中，要求按项目写清楚自定的传递函数，从屏幕上复制程序和运行结果，打印粘贴在报告上。