

控制工程基础实验指导书

(080104 班适用)

初红艳

北京工业大学机电学院

2010. 11

目 录

目录	1
一. 实验目的	2
二. 实验内容	2
三. 实验要求	3
四. 实验装置	4
五. 实验步骤	4
六. 实验报告要求	8
七. 附录(相关 MATLAB 函数介绍)	8

一. 实验目的

本实验中，学生使用 MATLAB 语言进行控制系统的分析，可以达到以下目的：

(1) 通过 MATLAB 的分析，掌握控制系统的时域瞬态响应、频率特性，根据时域性能指标、频域性能指标评价控制系统的性能，根据系统频率特性进行稳定性分析，了解对系统进行校正的方法，从而进一步巩固、加深对课堂内容的掌握，加强对控制工程基础知识的掌握。

(2) 通过本实验，使学生掌握进行控制系统计算机辅助分析的方法，学会利用 MATLAB 语言进行复杂的实际系统的分析、校正与设计，具备解决工程实际问题的能力。

二. 实验内容

控制系统方块图如图 1 所示。这是一个电压—转角位置随动系统，系统的功能是用电压量去控制一个设备的转角，给定值大，输出转角也就成比例地增大。

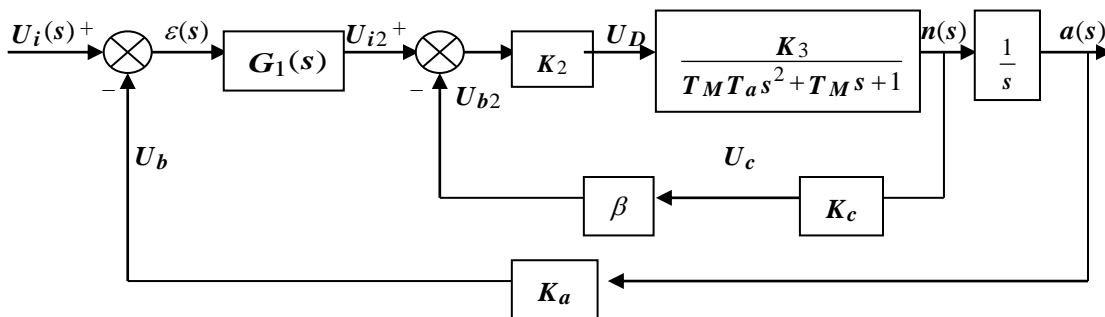


图 1 系统方块图

图中， $G_1(s)$ 为前置放大及校正网络传递函数

K_2 为功率放大器放大倍数， $K_2 = 10$

K_3 为电动机传递系数， $K_3 = 2.83 \text{ rad/V} \cdot \text{s}$

T_M 为电动机机电时间常数， $T_M = 0.1 \text{ s}$

T_a 为电动机电磁时间常数， $T_a = 4 \text{ ms}$

K_c 为测速传递系数， $K_c = 1.15 \text{ V} \cdot \text{s/rad}$

β 为测速反馈分压系数， $\beta = 0 \sim 1$

K_a 为主反馈电位计传递系数， $K_a = 4.7 \text{ V/rad}$

U_i 为输入电压

U_b 为反馈电压

U_{i2} 为速度环输入电压

U_c 为测速机电压

U_D 为电动机电压

n 为电动机转速

实验内容如下：

(1) 做二阶系统 $G(s) = \frac{K_3}{T_M T_a s^2 + T_M s + 1}$ 的阶跃响应，求其瞬态响应指标；分析此时系统处于哪种阻尼状态，系统的稳态值是多少；做系统的单位脉冲响应；在保持系统无阻尼自然频率不变的情况下，调整 T_M 、 T_a 使系统分别处于欠阻尼、临界阻尼、过阻尼状态，求系统在各种状态下的阶跃响应与单位脉冲响应及瞬态响应指标，分析在不同状态下的差别；分析当 T_M 、 T_a 为何值时，系统性能最佳。

(2) 对此系统中的小闭环系统：

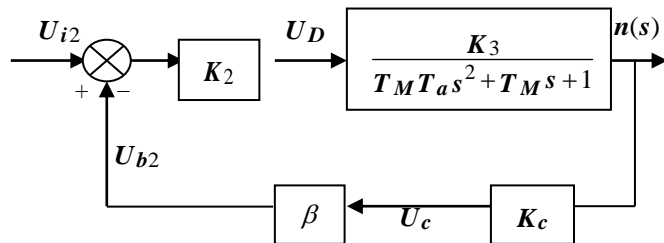


图2 小闭环系统

求其开环传递函数、闭环传递函数；做开环传递函数的乃氏图 (Nyquist)，分析闭环系统的稳定性；做开环传递函数的伯德图 (Bode)，求其频域指标，并分析闭环系统的稳定性。

(3) 当 $G_1(s) = 1$ 时，求整个系统的开环传递函数、闭环传递函数；做开环传递函数的乃氏图 (Nyquist)，分析该控制系统的稳定性；做开环传递函数的伯德图 (Bode)，求其频域指标，并分析该控制系统的稳定性。

(4) 若校正环节的传递函数为 $G_1(s) = \frac{0.12s + 1}{0.01s}$ ，做此校正环节的阶跃响应及单位脉冲响应；做此环节的乃氏图与伯德图；分析此环节的特性。

(5) 在校正环节的传递函数为 $G_1(s) = \frac{0.12s + 1}{0.01s}$ 时，求整个系统的开环传递函数、闭环传递函数；做系统开环传递函数的乃氏图与伯德图，与 $G_1(s) = 1$ 的情况进行比较，判断系统性能（稳定性、快速性）是否有所改善。

三. 实验要求

- (1) 记录、保存阶跃响应曲线、单位脉冲响应曲线
- (2) 记录、保存开环传递函数的乃氏图与伯德图

四. 实验装置

微机每人一台，已安装 MATLAB 软件

五. 实验步骤

鼠标双击桌面上的“MATLAB”图标，或搜索“matlab.exe”并双击，进入 MATLAB 的命令窗口，光标停留在“>>”后，此时就可以输入 MATLAB 函数。

无论是做系统的阶跃响应、单位脉冲响应，还是做开环传递函数的乃氏图与伯德图，都要用到系统的模型。

在 MATLAB 下，线性定常系统的模型有三种表示方法：

(a) 多项式模型

系统传递函数可表示成如下形式：

$$G(s) = \frac{X_o(s)}{X_i(s)} = \frac{b_0 s^m + b_1 s^{m-1} + \cdots + b_{m-1} s + b_m}{a_0 s^n + a_1 s^{n-1} + \cdots + a_{n-1} s + a_n}$$

此时，MATLAB 中可以用两个系数向量来唯一确定系统的传递函数：

$$\mathbf{num} = [b_0, b_1, \cdots, b_m], \quad \mathbf{den} = [a_0, a_1, \cdots, a_n]$$

利用 MATLAB 函数 `tf(num, den)` 即可建立系统多项式模型。

函数详细介绍见附录 1。

(b) 零极点模型

系统传递函数可表示成如下形式：

$$G(s) = k \frac{(s - z_1)(s - z_2) \cdots (s - z_m)}{(s - p_1)(s - p_2) \cdots (s - p_n)}$$

此时，MATLAB 中可以用一个增益量 k ，一个零点向量 z ，一个极点向量 p 表示系统的传递函数：

$$\mathbf{z} = [z_1, z_2, \cdots, z_m], \quad \mathbf{p} = [p_1, p_2, \cdots, p_n]$$

利用 MATLAB 函数 `zpk(z, p, k)` 即可建立系统零极点模型。

函数详细介绍见附录 1。

(c) 状态空间模型

本课程及实验内容未涉及到此模型，因此不再叙述。

系统模型确定之后，就可以利用相应的函数来做系统的阶跃响应、脉冲响应及乃氏图、伯德图。

(1) 做二阶系统 $G(s) = \frac{K_3}{T_M T_a s^2 + T_M s + 1}$ 的阶跃响应，求其瞬态响应指标

MATLAB 中，计算系统阶跃响应的函数为 `step`。函数详细介绍见附录 1。

在 MATLAB 的命令行窗口中，做如下输入（带下划线部分为用户输入）：

? num=[2.83] (注：2.83 为 K_3 的值)

num=

2.8300

? den=[0.0004, 0.1, 1] (注：0.0004 为 $T_M T_a$ 的值，0.1 为 T_M 的值)

den=

0.0004 0.1000 1.0000

? sys=tf(num, den) (建立系统多项式模型)

Transfer function:

2.83

0.0004 s^2 + 0.1 s + 1

? step(sys) (做系统阶跃响应)

此时，屏幕上会弹出图形窗口，窗口中的图形既为系统的阶跃响应图。同时按键盘上的“Alt” + “Print Screen”键保存此图形，并在图形中标出系统的瞬态响应指标（系统允许误差为 $\pm 2\%$ ）：

- 上升时间 t_r • 峰值时间 t_p • 最大超调量 M_p
- 调整时间 t_s • 延迟时间 t_d • 振荡次数

分析此时系统处于哪种阻尼状态？系统的稳态值是多少？

还可将屏幕图形保存成一个文件，具体方法见附录 1。

(2) 做二阶系统 $G(s) = \frac{K_3}{T_M T_a s^2 + T_M s + 1}$ 的单位脉冲响应

MATLAB 中，计算系统单位脉冲响应的函数为 `impulse`。

其用法为：**`impulse(sys)`** **`sys`** 为由 **`tf`** 或 **`zpk`** 函数建立的系统多项式模型或零极点模型
函数详细介绍见附录 1。

保存单位脉冲响应图形。

(3) 在保持系统无阻尼自然频率不变的情况下, 调整 T_M 、 T_a 使二阶系统 $G(s) = \frac{K_3}{T_M T_a s^2 + T_M s + 1}$ 分别处于欠阻尼、临界阻尼、过阻尼状态, 求系统在各种状态下的阶跃响应与单位脉冲响应及瞬态响应指标, 分析在不同状态下的差别; 分析当 T_M 、 T_a 为何值时, 系统性能最佳

调整 T_M 、 T_a 使系统分别处于欠阻尼状态 ($0 < \zeta < 1$), 临界阻尼状态 ($\zeta = 1$), 过阻尼状态 ($\zeta > 1$), 记录此时的 T_M 、 T_a 值;

用 `step` 函数做各状态下系统的阶跃响应, 在阶跃响应图中, 标出系统的瞬态响应指标: t_r , t_p , M_p , t_s , t_d 及振荡次数;

用 `impulse` 函数做各状态下系统的单位脉冲响应;

分析在不同状态下, 系统的阶跃响应、单位脉冲响应, 以及瞬态响应指标的差别;

对二阶振荡环节, 超调量 M_p 越小, 调整时间 t_s 越小 (也即系统响应越快), 则系统性能越好。对于此二阶系统, 分析当 T_M 、 T_a 为何值, 或 T_M 、 T_a 之间满足什么关系时, 系统性能最佳。

(4) 求小闭环系统的开环传递函数、闭环传递函数; 做开环传递函数的乃氏图 (Nyquist), 分析闭环系统的稳定性

设闭环系统的前向传递函数为 $G(s)$, 反馈传递函数为 $H(s)$, 则闭环系统的开环传递函数为 $G(s)H(s)$, 闭环传递函数为 $\frac{G(s)}{1+G(s)H(s)}$ 。

求图 2 所示小闭环系统的开环传递函数、闭环传递函数。

MATLAB 中, 绘制乃氏图的函数为 `nyquist`。

函数用法为: `nyquist(sys)` sys 为由 `tf` 或 `zpk` 函数建立的系统多项式模型或零极点模型

做小闭环系统的开环传递函数 $G(s)H(s)$ 的乃氏图, 根据乃奎斯特稳定判据判断此闭环系统的稳定性。乃奎斯特稳定判据见附录 2。

由 `nyquist(sys)` 函数绘制的乃氏图其频率范围为 0 到 $+\infty$, 在某些情况下, 部分频率范围的图形表示不清, 可利用 `nyquist(sys, ω)` 函数, 显式定义绘制乃氏图时的频率范围或频率点, 让函数只绘制所关心那部分频率的乃氏图, 以便进行系统的稳定性分析。

函数详细介绍见附录 1。

(5) 做小闭环系统开环传递函数的伯德图 (Bode), 求其频域指标, 分析闭环系统的稳定性

MATLAB 中，绘制伯德图的函数为 `bode`。

函数用法为：**`bode(sys)`** `sys` 为由 **`tf`** 或 **`zpk`** 函数建立的系统多项式模型或零极点模型
函数详细介绍见附录 1。

MATLAB 中，还有一函数为 `margin`，用来计算系统的幅值裕度与相位裕度

函数用法为：**`margin(sys)`** `sys` 为由 **`tf`** 或 **`zpk`** 函数建立的系统多项式模型或零极点模型
函数详细介绍见附录 1。

函数的此种用法不仅可以画出开环系统的伯德图，而且在图中能标出闭环系统的相对稳定性，也既闭环系统的幅值裕度与相位裕度（有关幅值裕度与相位裕度的知识见附录 2）。

由此可以得到系统的开环频域指标：

$$\bullet \omega_c \text{—开环截止频率 (rad/s)} \quad \bullet \gamma^\circ \text{—相位裕度} \quad \bullet K_g \text{—幅值裕度}$$

在工程实践中，为使闭环系统有满意的稳定储备，一般希望：

$$\gamma^\circ = 30^\circ \sim 60^\circ \quad K_g > 6dB$$

因此，利用 `margin` 函数，画出小闭环系统开环传递函数的伯德图，获得闭环系统的幅值裕度与相位裕度，并根据 $\gamma^\circ = 30^\circ \sim 60^\circ$ ， $K_g > 6dB$ 判断系统是否有足够的稳定性储备。

（6）当 $G_1(s)=1$ 时，求整个系统的开环传递函数、闭环传递函数；做开环传递函数的乃氏图（Nyquist），分析该控制系统的稳定性；做开环传递函数的伯德图（Bode），求其频域指标，并分析该控制系统的稳定性。

求出当 $G_1(s)=1$ 时整个系统的开环传递函数与闭环传递函数；

采用与（4）相同的方法做开环传递函数的乃氏图，根据乃奎斯特稳定判据判断此闭环系统的稳定性；

采用与（5）相同的方法做开环传递函数的伯德图，获得其开环频域指标，并根据相位裕度与幅值裕度分析系统的稳定性。

（7）若校正环节的传递函数为 $G_1(s) = \frac{0.12s+1}{0.01s}$ ，做此校正环节的阶跃响应及单位脉冲响应；

做此环节的乃氏图与伯德图

此校正环节为比例积分环节（PI）

利用 `step` 函数做此校正环节的阶跃响应；

利用 `impulse` 函数做此校正环节的单位脉冲响应；

利用 `nyquist` 函数做此校正环节的乃氏图；

利用 `bode` 函数做此校正环节的伯德图，分析此环节的特性，例如，该环节的输入与输

出在相位上有什么关系，在幅值上又有什么关系。

(8) 在校正环节的传递函数为 $G_1(s) = \frac{0.12s+1}{0.01s}$ 时，求整个系统的开环传递函数、闭环传递函数；做系统开环传递函数的乃氏图与伯德图；与 $G_1(s) = 1$ 的情况进行比较，判断系统性能（稳定性、快速性）是否有所改善。

求出当 $G_1(s) = \frac{0.12s+1}{0.01s}$ 时整个系统的开环传递函数与闭环传递函数；

利用 `nyquist` 函数做开环传递函数的乃氏图；

利用 `bode` 或 `margin` 函数做开环传递函数的伯德图；

根据相位裕度与幅值裕度分析系统的稳定性，并与 $G_1(s) = 1$ 的情况相比较，分析系统在稳定性方面是否有所改善；

系统的截止频率 ω_c 越高，则系统的快速性越好。比较 $G_1(s) = \frac{0.12s+1}{0.01s}$ 与 $G_1(s) = 1$ 两种情况下的截止频率 ω_c ，判断系统的快速性是否有所改善。

六. 实验报告要求

- (1) 实验报告要求字迹工整，表达清楚，最好打印。
- (2) 按“实验内容”中的要求，按步骤书写实验报告。对“实验内容”中要求做阶跃响应、脉冲响应，或系统的乃氏图、伯德图之处，实验报告中应附上图形；对要求分析的，实验报告中应做详尽的分析。

七. 附录 1（相关 MATLAB 函数介绍）

- (1) 模型建立及模型转换函数 `tf`，`zpk`

- a. `tf` 函数

功能：生成系统多项式模型，或者将零极点模型或状态空间模型转换成多项式模型格式：

(I) `sys = tf(num,den)` `num`，`den` 为系统多项式模型的系数向量，返回值 `sys` 为系统的多项式模型

例如，在 MATLAB 命令行窗口做如下输入（带下划线部分为用户输入，其他部分为系统回显），

```
? num=[5]
```

```
num=
```

```

5
? den=[2 5 1]
den=
    2    5    1
? sys=tf(num,den)

```

Transfer function:

```

5
-----
2 s^2 + 5 s + 1

```

(II) ***tf******sys = tf(sys)*** 将系统的零极点模型或状态空间模型转换成多项式模型, ***sys*** 为系统的零极点模型或状态空间模型, 返回值 ***tf******sys*** 为系统的多项式模型
 举例见 ***zpk*** 函数。

函数的其他格式见软件的在线帮助, 或在命令行中输入

```

? help tf

```

b. ***zpk*** 函数

功能: 生成系统零极点模型, 或者将状态空间模型或多项式模型转换成零极点模型
 格式:

(I) ***sys = zpk(z,p,k)*** ***z***, ***p***, ***k*** 分别为系统的零点、极点和增益, 返回值 ***sys*** 为系统的零极点模型

例如, 在 MATLAB 命令行窗口做如下输入 (带下划线部分为用户输入, 其他部分为系统回显),

```

? z=[1]
z=
    1
? p=[2 3]
p=
    2    3
? k=5
k=
    5
? sys=zpk(z,p,k)
Zero/pole/gain:

```

$$\frac{5(s-1)}{(s-2)(s-3)}$$

```
?tfsys=tf(sys)
```

Transfer function:

$$\frac{5s - 5}{s^2 - 5s + 6}$$

(II) **zsys = zpk(sys)** 将系统的状态空间模型或多项式模型转换成零极点模型，**sys** 为系统的状态空间模型或多项式模型，返回值 **zsys** 为系统的零极点模型。函数的其他函数的其他格式见软件的在线帮助，或在命令行中输入

```
? help zpk
```

(2) **step** 函数

功能：计算系统阶跃响应

格式：

(I) **step(sys)** **sys** 为由 **tf** 或 **zpk** 函数建立的系统模型；

函数无返回值，**step** 在当前图形窗口中直接绘出系统的阶跃响应。

例如，对上例中建立的系统零极点模型 **sys**，

```
? step(sys)
```

屏幕上弹出如图 3 所示图形窗口，此既系统的阶跃响应。

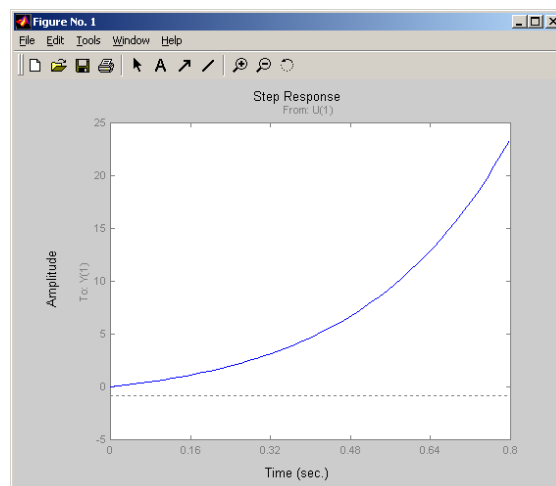


图 3

(II) **step(sys,t)** 定义计算时的时间矢量。用户可以指定一个仿真终止时间，这时 **t** 为一个标量；也可以通过诸如 **t=0:dt:Tfinal** 命令设置一个时间矢量。

例如，图 3 的仿真终止时间为 0.8s，可以通过如下输入，将仿真终止时间指定为 1s。

? step(sys,1)

(III) **step(sys1,sys2,...sysN)** 或 **step(sys1,sys2,...sysN,t)** 同时对多个系统进行仿真。

step(sys1,'PlotStyle1',sys2,'PlotStyle2',...sysN,'PlotstyleN') 定义每个仿真对象的绘制属性。其中 *PlotStyle1*, *PlotStyle2*, ...*PlotStyleN* 为 MATLAB 标准命令 plot 支持的各种属性标识字符串, 它指定了图形的线型、颜色等, 具体取值可见软件在线帮助或查阅相关资料。

其输出图形如图 4 所示。

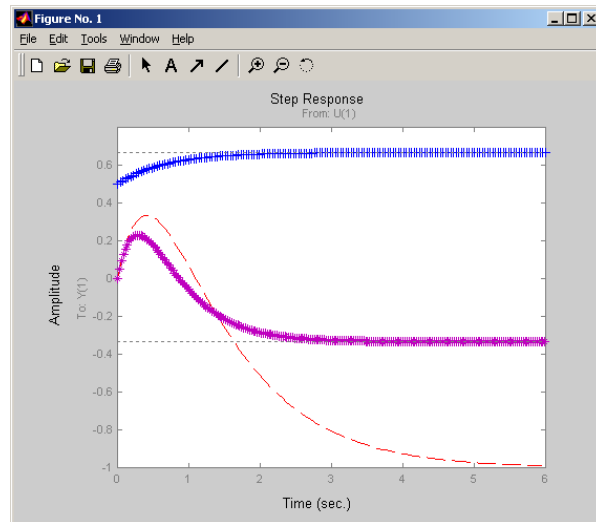


图 4

(IV) **[y,t,x]=step(sys)** 计算仿真数据, 并且不显示图形。其中, *y* 为输出响应矢量, *t* 为时间矢量, *x* 为状态轨迹数据。

本次实验用不到此种格式。

(3) **impulse** 函数

功能: 计算系统单位脉冲响应

函数格式及用法与 step 函数完全相同, 只是将 step 换成了 impulse, 如, **impulse(sys)**, **impulse(sys1,sys2,...sysN)** 等。

(4) **nyquist** 函数

功能: 求系统的 Nyquist (奈奎斯特) 频率曲线

函数无返回值时, **nyquist** 函数会在当前图形窗口中直接绘制出 Nyquist 曲线(乃氏图)

格式:

(I) **nyquist(sys)** *sys* 为由 **tf** 或 **zpk** 函数建立的系统模型

例如: 在 MATLAB 的命令行中做如下输入

? H=tf([2 5 1],[1 2 3]) ([2 5 1], [1 2 3] 分别为系统多项式模型的分子、分母系数向量)

Transfer function:

$$2s^2 + 5s + 1$$

$$s^2 + 2s + 3$$

? nyquist (H)

屏幕上弹出如图 5 所示图形窗口，此既系统的乃氏图。

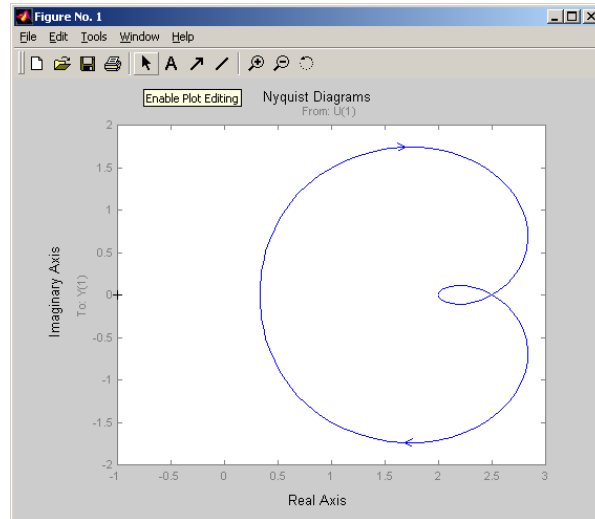


图 5

此乃氏图系统的频率范围为 0 到 $+\infty$

由 *nyquist* 函数绘制的乃氏图总是关于坐标实轴对称的，但真正的乃氏图只是其中的一半，可通过格式 (IV) 所返回的频率响应的实部 *re* 和虚部 *im* 来判断哪一部分才是所需的乃氏图，并用鼠标点击图形窗口中工具栏上的箭头按钮（如图 5 所示，当鼠标指向该按钮时，旁边显示“Enable Plot Editing”），对图形进行编辑，删掉不需要的那一部分，剩余部分即为真正的乃氏图，如图 6 所示。

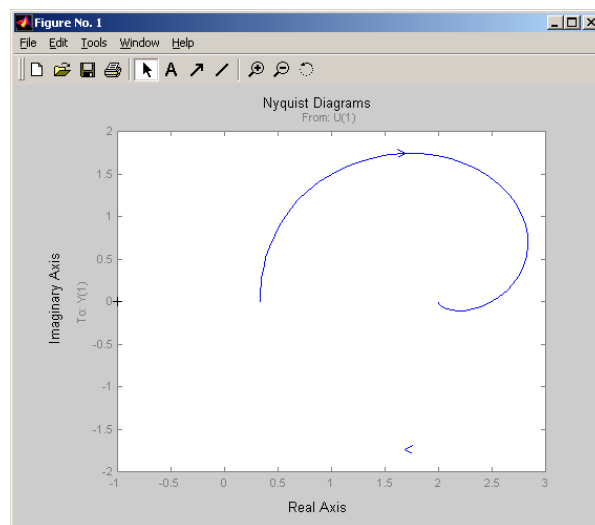


图 6

(II) *nyquist(sys,ω)* 显式定义绘制乃氏图时的频率范围或频率点。若要定义频率范围， ω 必须具有 {*wmin,wmax*} 格式；如果定义频率点，则 ω 必须为由需要频率点频率组成的

向量。

例如，对上例建立的系统模型，

? nyquist(H, [0.1:0.0001:2.5]) 或 nyquist(H, {0.1, 2.5})

屏幕弹出如图 7 所示的图形。

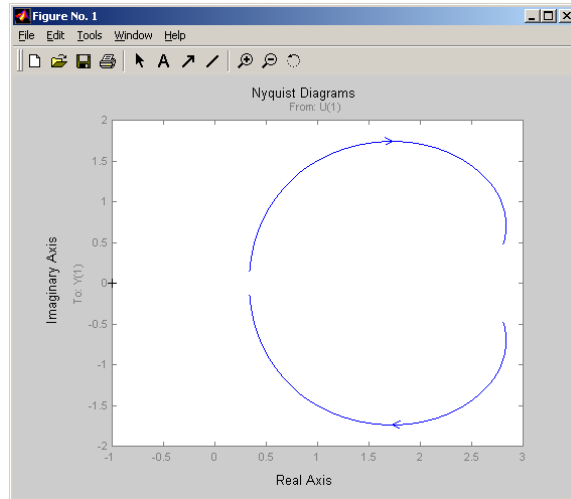


图 7

使用此函数格式时，频率范围的确定是关键。可利用鼠标点击图形上的某一点，则在鼠标旁显示该点的实部、虚部，以及频率点，如图 8 所示，用此方法既可确定乃氏图中所关心那部分的频率范围，然后再利用此函数，绘制该部分的乃氏图。

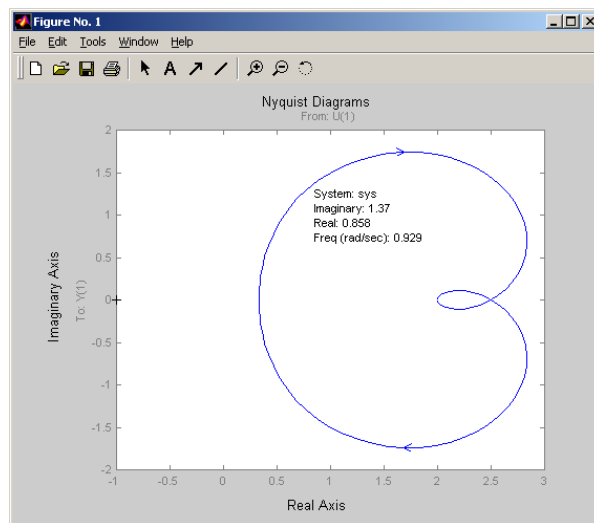


图 8

(III) **nyquist(sys1,sys2,...sysN)** 或 **nyquist(sys1,sys2,...sysN, ω)** 在一个窗口内同时绘制多个系统的乃氏图

nyquist(sys1,'PlotStyle1',sys2,'PlotStyle2',...sysN,'PlotStyleN') 定义每个乃氏图的绘制属性，其中 *PlotStyle1*, *PlotStyle2*, ...*PlotStyleN* 为 MATLAB 标准命令 plot 支持的各种属性标识字符串，它指定了图形的线型、颜色等，具体取值可见软件在线帮助或查阅相关资料。

(IV) $[re, im, \omega] = nyquist(sys)$ 和 $[re, im] = nyquist(sys, \omega)$ 返回系统在频率 ω 处的频率响应。其中, re 为频率响应的实部, im 为频率响应的虚部, ω 为频率点。

函数此种用法不绘制乃氏图。

(5) **bode** 函数

功能: 绘制系统伯德图 (Bode)

函数格式及用法与 **nyquist** 函数基本相同, 如 **bode(sys)**, **bode(sys1, sys2, ..., sysN, \omega)** 等, 只有如下不同之处:

$[mag, phase, \omega] = bode(sys)$ 和 $[mag, phase] = bode(sys, \omega)$ 计算系统 bode 图数据, 并且不显示图形。其中, mag 为 Bode 图的幅值, $phase$ 为 Bode 图的相位值, ω 为 Bode 图的频率点。

对前面所建系统模型 H,

? bode(H)

屏幕弹出如图 9 所示图形。

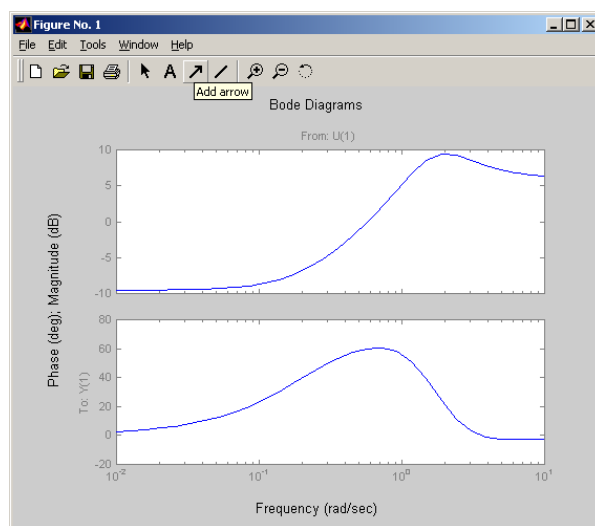


图 9

(6) **margin** 函数

功能: 计算系统的增益、相位裕度以及相应的交叉频率。增益和相位裕度是针对开环系统而言的, 它指示出当前系统闭环时的相对稳定性。

格式:

(I) $[G_m, P_m, \omega_{cg}, \omega_{cp}] = margin(sys)$ 计算系统 **sys** 的增益和相位裕度。返回值中, G_m 对应于系统的增益裕度, ω_{cg} 为其响应的交叉频率; P_m 对应于系统的相位裕度, ω_{cp} 为其响应的交叉频率。

例如: ? H=tf([10], [1 6 5 0])

```

Transfer function:
      10
-----
      s^3 + 6 s^2 + 5 s
? [gm, pm, wg, wp]=margin(H)
gm =
      3
pm =
      25.3898
wg =
      2.2361
wp =
      1.2271

```

(II) $[G_m, P_m, \omega_{cg}, \omega_{cp}] = \text{margin}(\text{mag}, \text{phase}, \omega)$ mag , phase 和 ω 为 bode 函数返回的系统 bode 图数据，此函数根据 mag , phase 和 ω 计算系统的增益和相位裕度。

(III) $\text{margin}(\text{sys})$ 在当前图形窗口中绘制出系统裕度的 Bode 图

例如，对前面建立的系统模型 H, ? margin(H)

屏幕弹出如图 10 所示图形。

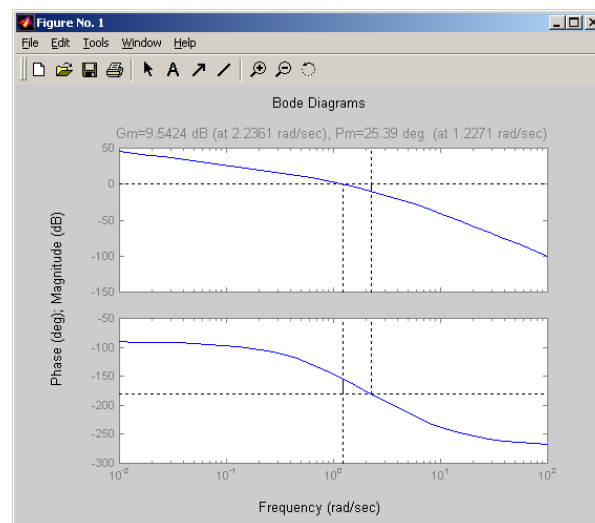


图 10

此图形中标出了系统的增益和相位裕度以及相应的交叉频率。与前面例题的结果进行比较，两种情况下的增益裕度 G_m 不同，这是因为其单位不同，本例中， G_m 是以 dB 表示的，而前面例题则不是， $20\log 3 = 9.542425\text{dB}$ 。

(7) MATLAB 图形保存成文件

对于图 11 所示图形，若要将其保存成一个文件，可选择下拉菜单“File”中的“Export...”，则屏幕弹出图 12 所示对话框，在对话框中，选择文件的保存类型，并输入文件名，然后点

击“保存(S)”按钮，则此图形就被保存成一个文件。在“Microsoft Word”中，可通过插入图片，将其插入到所需位置处，如图 13 所示。

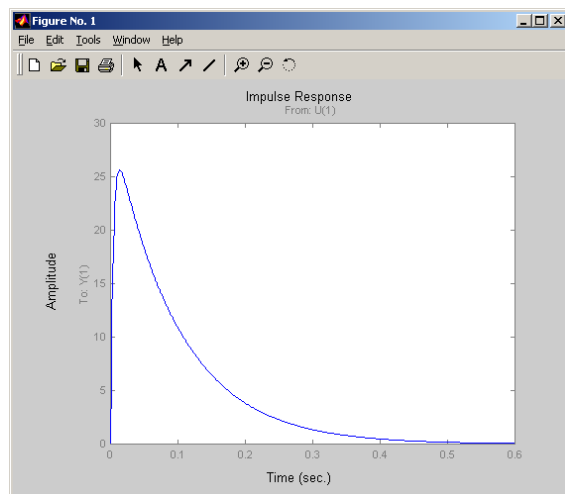


图 11

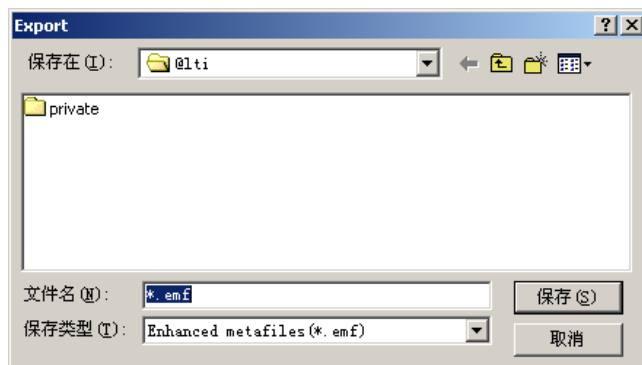


图 12

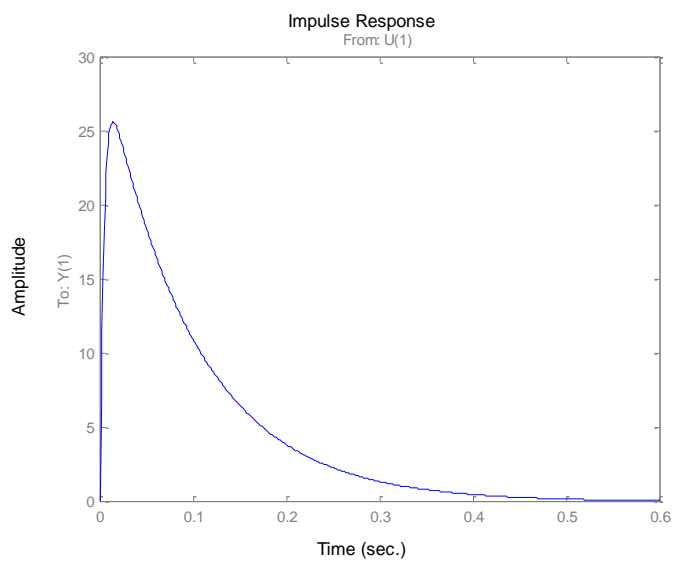


图 13