

## 一、 系统描述模型练习

### 1. 基本描述系统的命令

tf 命令：创建传递函数模型，第一个参数为分子系数，第二个参数为分母系数；  
minreal 命令：生成系统的最简模型，即消除状态空间模型中的不可控或不可观状态，或消除传递函数模型的零极点对；

zpk 命令：创建零极点增益模型或转化成零极点增益模型；

feedback 命令：创建负反馈或正反馈连接模型；

注意：G=tf(num,den)获取的是复域 s 的传递函数，而 G=tf(num,den,ts)获取的是离散域(z 域)的传递函数，而且该传递函数并不等于 G=c2d(tf(num,den), 'method') 的结果。

运行结果如下：

```
G4 =
G1 =
    s + 1
-----
s^2 + 2 s + 1
Continuous-time transfer function.

G4 =
s^2 + 2 s + 1
-----
s^3 + 3 s^2 + 4 s + 2
Continuous-time transfer function.

G5 =
G2 =
    1
-----
s + 1
Continuous-time transfer function.

G5 =
s^2 + 2 s + 1
-----
s^3 + 3 s^2 + 2 s
Continuous-time transfer function.

G6 =
G3 =
    (s+1)
-----
(s+1)^2
Continuous-time zero/pole/gain model.

G6 =
z + 1
-----
z^2 + 2 z + 1
Sample time: 0.01 seconds
Discrete-time transfer function.
```

### 2. 各种模型之间的转换方法

代码和结果如下：

#### % 模型转换

```
[A, B, C, D]=tf2ss (G1. num{1}, G1. den{1})      %注意G1. num, G1. den为元胞数组的引用格式
W1=c2d(G1, 0. 1, 'ZOH')
W2=c2d(G1, 0. 01, 'ZOH')
W3=c2d(G1, 0. 1, 'Tustin')
W4=c2d(G1, 0. 01, 'Tustin')
```

```

A =
-2    -1
 1     0

W1 =
B =
      0.09516 z - 0.08611
-----
1           z^2 - 1.81 z + 0.8187
0
Sample time: 0.1 seconds
Discrete-time transfer function.

C =
W2 =
 1     1
      0.00995 z - 0.009851
-----
z^2 - 1.98 z + 0.9802
D =
 0
Sample time: 0.01 seconds
Discrete-time transfer function.

W3 =
      0.04762 z^2 + 0.004535 z - 0.04308
-----
z^2 - 1.81 z + 0.8186
Sample time: 0.1 seconds
Discrete-time transfer function.

W4 =
      0.004975 z^2 + 4.95e-05 z - 0.004926
-----
z^2 - 1.98 z + 0.9802
Sample time: 0.01 seconds
Discrete-time transfer function.

```

可以看出，离散化方法不同，所得离散系统模型不同；采样时间不同，所得离散系统模型也不同。

### 3. 频域分析命令

注意：波特图、奈奎斯特图、尼柯尔斯曲线、根轨迹曲线描述的虽然是闭环系统的频域特性，但是根据开环传递函数画出。

代码和结果如下：

```

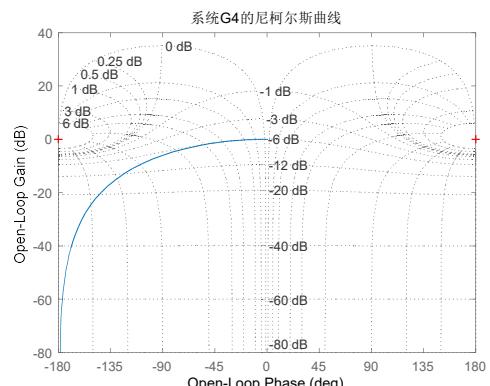
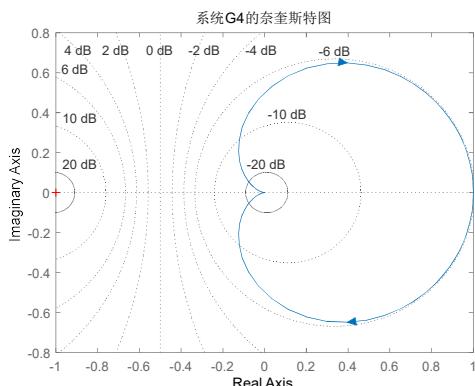
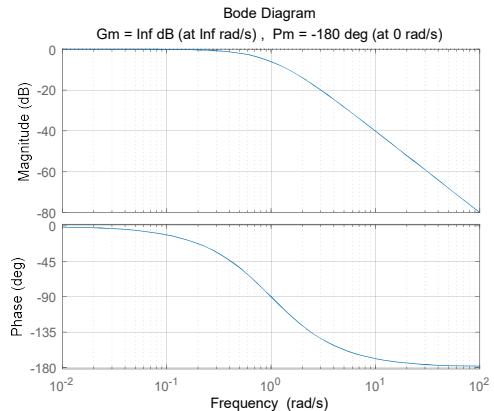
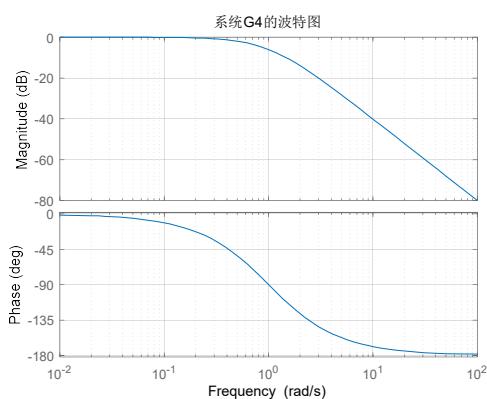
%% 频域分析
figure('Name','系统G4的波特图','NumberTitle','off'),bode(G1*G2),grid on;
title('系统G4的波特图')
figure('Name','系统G4的幅值裕度和相角裕度','NumberTitle','off'),margin(G1*G2),grid on;
title('系统G4的幅值裕度和相角裕度')
figure('Name','系统G4的奈奎斯特图','NumberTitle','off'),nyquist(G1*G2),grid on;
title('系统G4的奈奎斯特图')
figure('Name','系统G4的尼柯尔斯曲线','NumberTitle','off'),nichols(G1*G2),grid on;
title('系统G4的尼柯尔斯曲线')
figure('Name','系统G4的根轨迹曲线','NumberTitle','off'),rlocus(G1*G2),grid on;
title('系统G4的根轨迹曲线')
disp('系统G4的幅值裕度和相角裕度如下：')
[Gm,Pm,wcg,wc]=margin(G1*G2)
figure('Name','系统G5的波特图','NumberTitle','off'),bode(-G1*G2),grid on;
title('系统G5的波特图')
figure('Name','系统G5的幅值裕度和相角裕度','NumberTitle','off'),margin(-G1*G2),grid on;
title('系统G5的幅值裕度和相角裕度')
figure('Name','系统G5的奈奎斯特图','NumberTitle','off'),nyquist(-G1*G2),grid on;
title('系统G5的奈奎斯特图')
figure('Name','系统G5的尼柯尔斯曲线','NumberTitle','off'),nichols(-G1*G2),grid on;

```

```

title('系统G5的尼柯尔斯曲线')
figure('Name','系统G5的根轨迹曲线','NumberTitle','off'), rlocus(-G1*G2), grid on;
title('系统G5的根轨迹曲线')
disp('系统G5的幅值裕度和相角裕度如下：')
[Gm, Pm, wcg, wc] = margin(-G1*G2)

```



系统G4的幅值裕度和相角裕度如下：

G<sub>m</sub> =

Inf

P<sub>m</sub> =

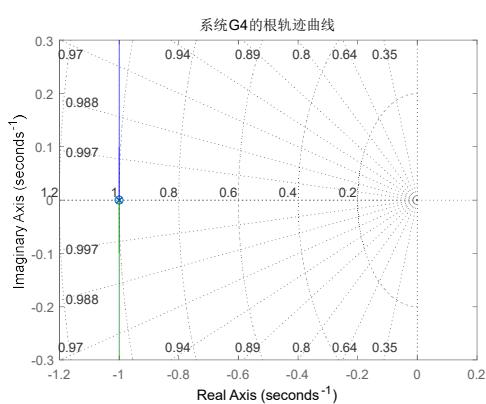
-180

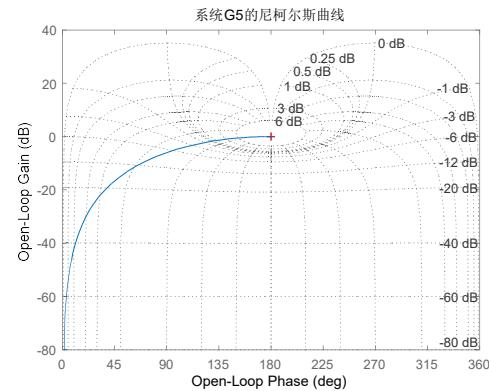
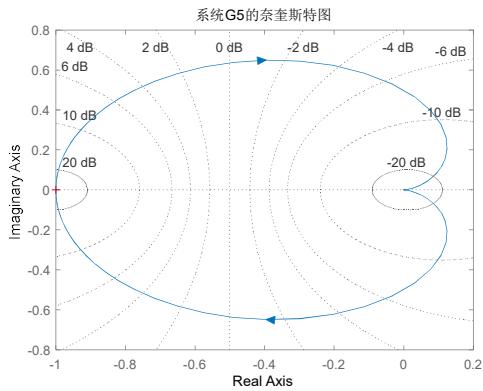
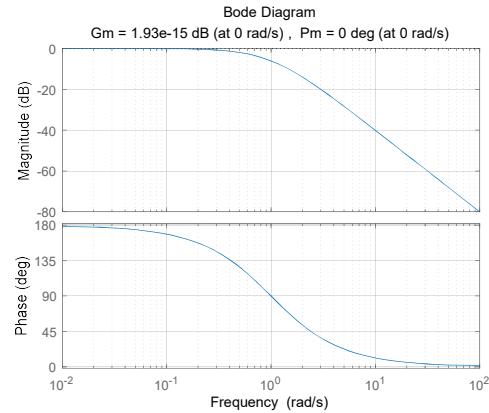
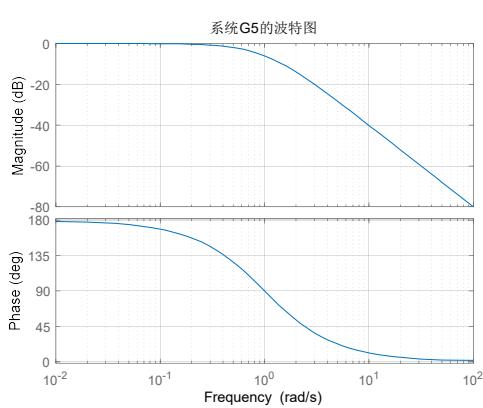
wcg =

Inf

wc =

0





系统G5的幅值裕度和相角裕度如下：

**警告:** The closed-loop system is unstable.

Gm =

1.0000

Pm =

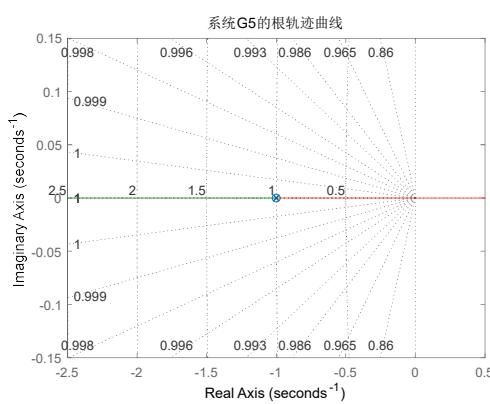
0

wcg =

0

wc =

0



注意：利用  $[Gm, Pm, wcg, wc] = margin(sys)$  命令求取 sys 系统的稳定裕度时的单位，幅值裕度是 Gm 是十进制的幅值，不是 dB，相位裕度单位是 deg，对应的频率单位是 rad/sec。而 margin(sys) 画图后，在界面上显示的幅值裕度单位是 dB，相位裕度单位是 deg。

由图分析可得，G4 系统奈奎斯特曲线不包围  $(-1, j0)$  点，根轨迹曲线不过正半平面，幅值裕度无穷大，相位裕度  $180^\circ$ ，所以 G4 系统稳定；G5 系统奈奎斯特曲线包围  $(-1, j0)$  点，根轨迹曲线过正半平面，幅值裕度为 0，相位裕度为  $0^\circ$ ，所以 G5 系统不稳定。

#### 4. 带纯滞后环节的传递函数输入

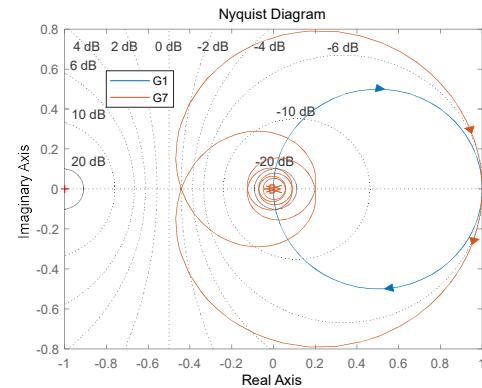
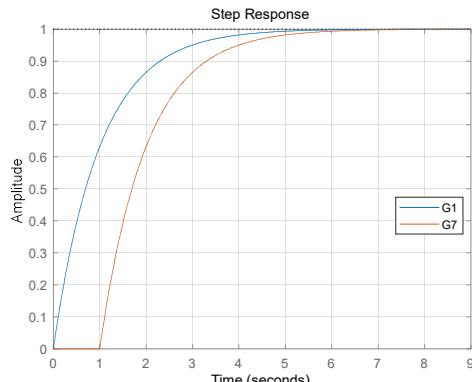
注：带纯滞后环节的传递函数  $\frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0} e^{-\tau s}$  的设置，可以用如下

方式进行  $\text{tf}([b_m \ b_{m-1} \ \dots \ b_1 \ b_0], [a_n \ a_{n-1} \ \dots \ a_1 \ a_0], \text{'iodelay'}, \tau)$

代码和结果如下：

% 纯滞后环节传函

```
G7=tf([1 1], [1 2 1], 'iodelay', 1)
figure('Name', '系统G1和G7的阶跃响应', 'NumberTitle', 'off')
step(G1), hold on;
step(G7), grid on;
legend({'G1', 'G7'}, 'Location', 'best')
figure('Name', '系统G1和G7的幅相曲线', 'NumberTitle', 'off')
nyquist(G1), hold on;
nyquist(G7), grid on;
legend({'G1', 'G7'}, 'Location', 'best')
```



由图可得，带纯滞后环节的阶跃响应不带纯滞后环节的阶跃响应慢 1s，幅相曲线有 1rad 的相角滞后。

## 5. 综合练习

(1) 代码和结果如下：

% 综合练习

```
% x = [DELTA_V DELTA_alpha DELTA_q DELTA_theta]';
% u = [DELTA_delta_e DELTA_delta_T]';
% x_dot = A*x + B*u
% y = C*x + D;
% input 1           2
%      DELTA_delta_e DELTA_delta_T
% output 1          2          3          4
%      DELTA_V DELTA_alpha DELTA_q DELTA_theta
A=[-0.0313 4.4879 0 -9.8; ...
    -0.0058 -0.7458 0.9480 0; ...
    -0.0002 -0.4479 -0.4609 0; ...
    0          0          1          0];
```

```

B=[0.2063 -0.0516 -0.6532 0; ...
    3.4847 -0.0561 0.0193 0]';

C=eye(4);

D=[0 0 0 0; 0 0 0 0]';

Gss=ss(A,B,C,D);

Gtf=tf(Gss);      %两个输入到四个输出的所有传函
% Gtf(output, input)
disp(' DELTA_alpha/DELTA_delta_e 传递函数如下: ')
G_elvt2atk=Gtf(2,1)
disp(' DELTA_q/DELTA_delta_e 传递函数如下: ')
G_elvt2pcr=Gtf(3,1)
disp(' DELTA_theta/DELTA_delta_T 传递函数如下: ')
G_trl2pitch=Gtf(4,2)
disp(' DELTA_V/DELTA_delta_T 传递函数如下: ')
G_trl2vlct=Gtf(1,2)

DELTA_alpha/DELTA_delta_e 传递函数如下:          DELTA_theta/DELTA_delta_T 传递函数如下:
G_elvt2atk =                                         G_trl2pitch =

$$\frac{-0.0516 s^3 - 0.6458 s^2 - 0.02072 s - 0.03703}{s^4 + 1.238 s^3 + 0.8321 s^2 + 0.03494 s + 0.024}$$


$$\frac{0.0193 s^2 + 0.03943 s + 0.01032}{s^4 + 1.238 s^3 + 0.8321 s^2 + 0.03494 s + 0.024}$$

Continuous-time transfer function.                  Continuous-time transfer function.

DELTA_q/DELTA_delta_e 传递函数如下:             DELTA_V/DELTA_delta_T 传递函数如下:
G_elvt2pcr =                                         G_trl2vlct =

$$\frac{-0.6532 s^3 - 0.4845 s^2 - 0.03098 s}{s^4 + 1.238 s^3 + 0.8321 s^2 + 0.03494 s + 0.024}$$


$$\frac{3.485 s^3 + 3.953 s^2 + 2.454 s - 0.3873}{s^4 + 1.238 s^3 + 0.8321 s^2 + 0.03494 s + 0.024}$$

Continuous-time transfer function.                  Continuous-time transfer function.

```

## (2) 代码和结果如下:

```

[z, p, k]=zpkdata(G_elvt2atk);
disp(' DELTA_alpha/DELTA_delta_e 零点如下: '), z{1}
disp(' DELTA_alpha/DELTA_delta_e 极点如下: '), p{1}

[z, p, k]=zpkdata(G_elvt2pcr);
disp(' DELTA_q/DELTA_delta_e 零点如下: '), z{1}
disp(' DELTA_q/DELTA_delta_e 极点如下: '), p{1}

[z, p, k]=zpkdata(G_trl2pitch);
disp(' DELTA_theta/DELTA_delta_T 零点如下: '), z{1}
disp(' DELTA_theta/DELTA_delta_T 极点如下: '), p{1}

[z, p, k]=zpkdata(G_trl2vlct);
disp(' DELTA_V/DELTA_delta_T 零点如下: '), z{1}
disp(' DELTA_V/DELTA_delta_T 极点如下: '), p{1}

disp(' 短周期和长周期自然频率和阻尼比如下: ')
damp(G_elvt2pcr)

```

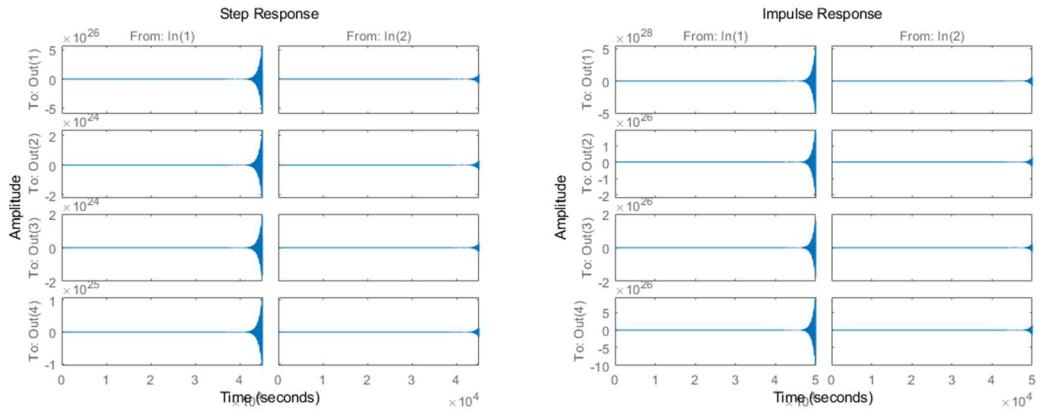
<p>DELTA_alpha/DELTA_delta_e 零点如下:</p> <pre>ans = -12.4885 + 0.0000i -0.0138 + 0.2393i -0.0138 - 0.2393i</pre> <p>DELTA_alpha/DELTA_delta_e 极点如下:</p> <pre>ans = -0.6203 + 0.6487i -0.6203 - 0.6487i 0.0013 + 0.1726i 0.0013 - 0.1726i</pre> <p>DELTA_theta/DELTA_delta_T 零点如下:</p> <pre>ans = -1.7346 -0.3083</pre> <p>DELTA_theta/DELTA_delta_T 极点如下:</p> <pre>ans = -0.6203 + 0.6487i -0.6203 - 0.6487i 0.0013 + 0.1726i 0.0013 - 0.1726i</pre>	<p>DELTA_q/DELTA_delta_e 零点如下:</p> <pre>ans = 0 -0.6711 -0.0707</pre> <p>DELTA_q/DELTA_delta_e 极点如下:</p> <pre>ans = -0.6203 + 0.6487i -0.6203 - 0.6487i 0.0013 + 0.1726i 0.0013 - 0.1726i</pre> <p>DELTA_V/DELTA_delta_T 零点如下:</p> <pre>ans = -0.6314 + 0.6839i -0.6314 - 0.6839i 0.1283 + 0.0000i</pre> <p>DELTA_V/DELTA_delta_T 极点如下:</p> <pre>ans = -0.6203 + 0.6487i -0.6203 - 0.6487i 0.0013 + 0.1726i 0.0013 - 0.1726i</pre>
--	--

短周期和长周期自然频率和阻尼比如下:

Pole	Damping	Frequency (rad/seconds)	Time Constant (seconds)
1.25e-03 + 1.73e-01i	-7.27e-03	1.73e-01	-7.97e+02
1.25e-03 - 1.73e-01i	-7.27e-03	1.73e-01	-7.97e+02
-6.20e-01 + 6.49e-01i	6.91e-01	8.97e-01	1.61e+00
-6.20e-01 - 6.49e-01i	6.91e-01	8.97e-01	1.61e+00
..			

(3) 代码和结果如下:

```
% input 1          2
%      DELTA_delta_e DELTA_delta_T
% output 1        2        3        4
%      DELTA_V DELTA_alpha DELTA_q DELTA_theta
figure('Name','系统的阶跃响应曲线','NumberTitle','off')
step(Gtf)
figure('Name','系统脉冲响应曲线','NumberTitle','off')
impulse(Gtf)
```



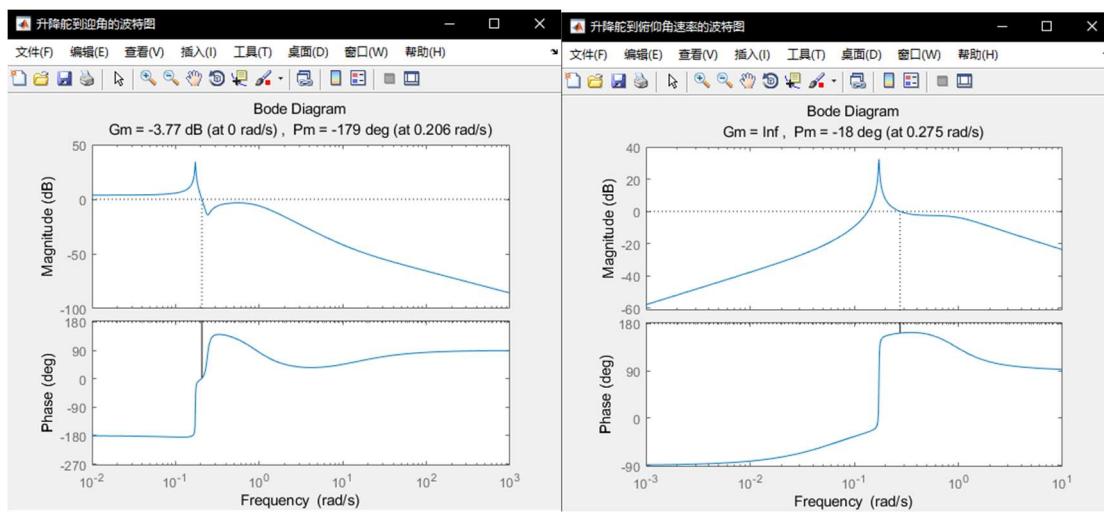
由图可知，系统的所有阶跃响应和脉冲响应都发散，系统不稳定。

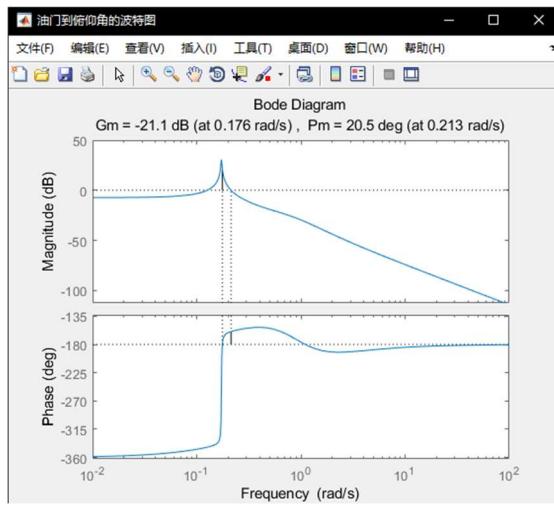
#### (4) 代码和结果如下：

```
figure('Name', '升降舵到迎角的波特图', 'NumberTitle', 'off')
% bode(G_elvt2atk)
% [Gm, Pm, wcg, wc]= ...
margin(G_elvt2atk)

figure('Name', '升降舵到俯仰角速率的波特图', 'NumberTitle', 'off')
% bode(G_elvt2pcr)
% [Gm, Pm, wcg, wc]= ...
margin(G_elvt2pcr)

figure('Name', '油门到俯仰角的波特图', 'NumberTitle', 'off')
% bode(G_trl2pitch)
% [Gm, Pm, wcg, wc]= ...
margin(G_trl2pitch)
```



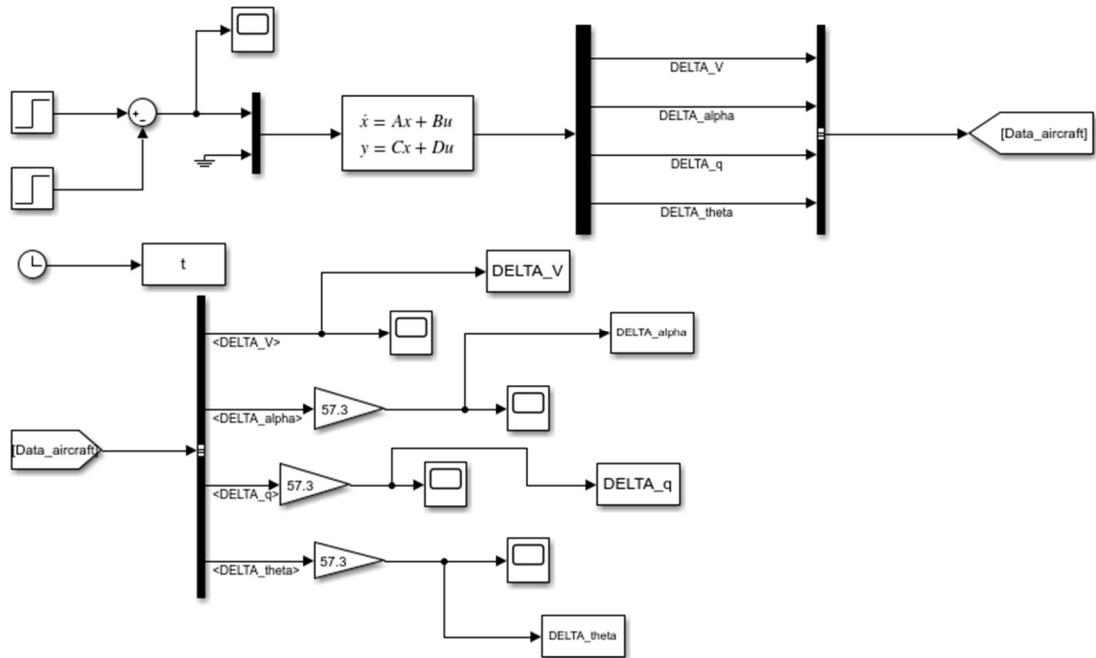


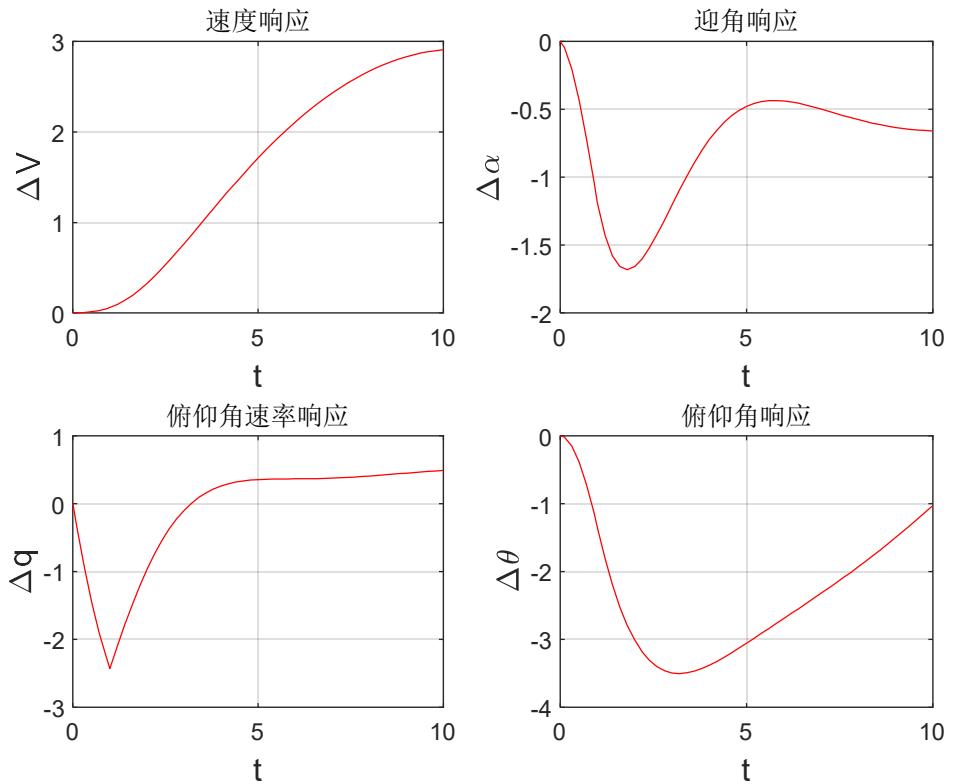
由图可知，升降舵到迎角传函的相位裕度为负，系统不稳定；升降舵到俯仰角传函的相位裕度为负，系统不稳定；油门到俯仰角传函的相角交界频率小于截止频率，幅值裕度为负，系统不稳定。

## 二、Simulink 练习

### 1. 用 Simulink 搭建上述飞机模型：

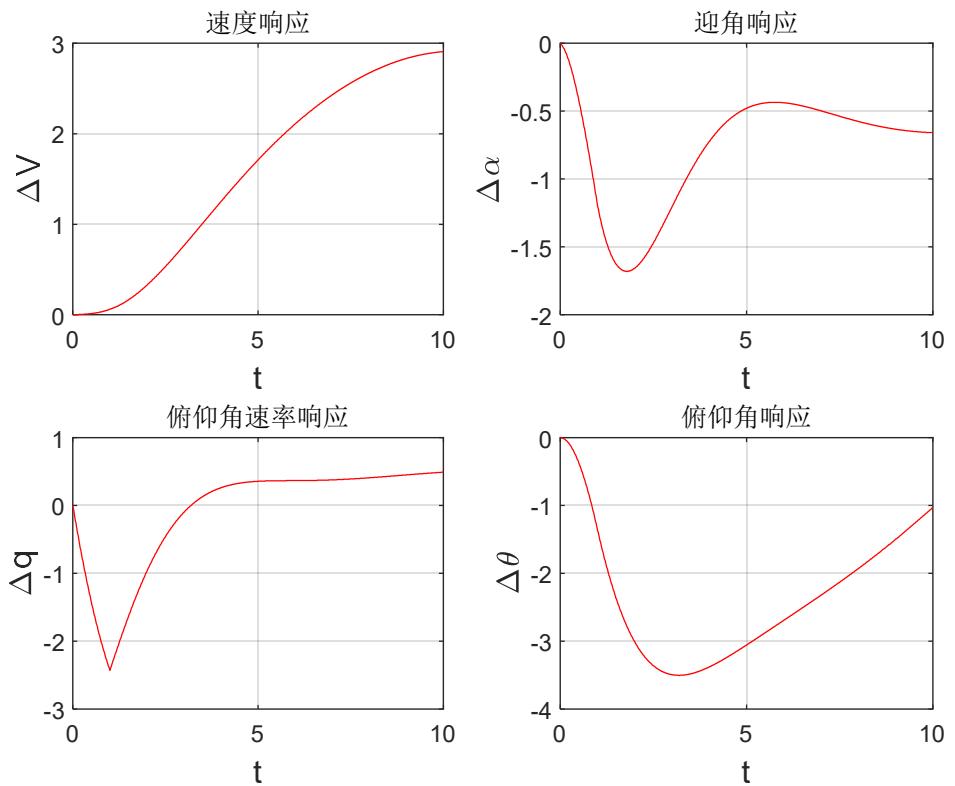
注意：在 simulink 仿真中，要注意：信号的维数前后是否一致、仿真步长是否合适、模块参数设置是否正确。



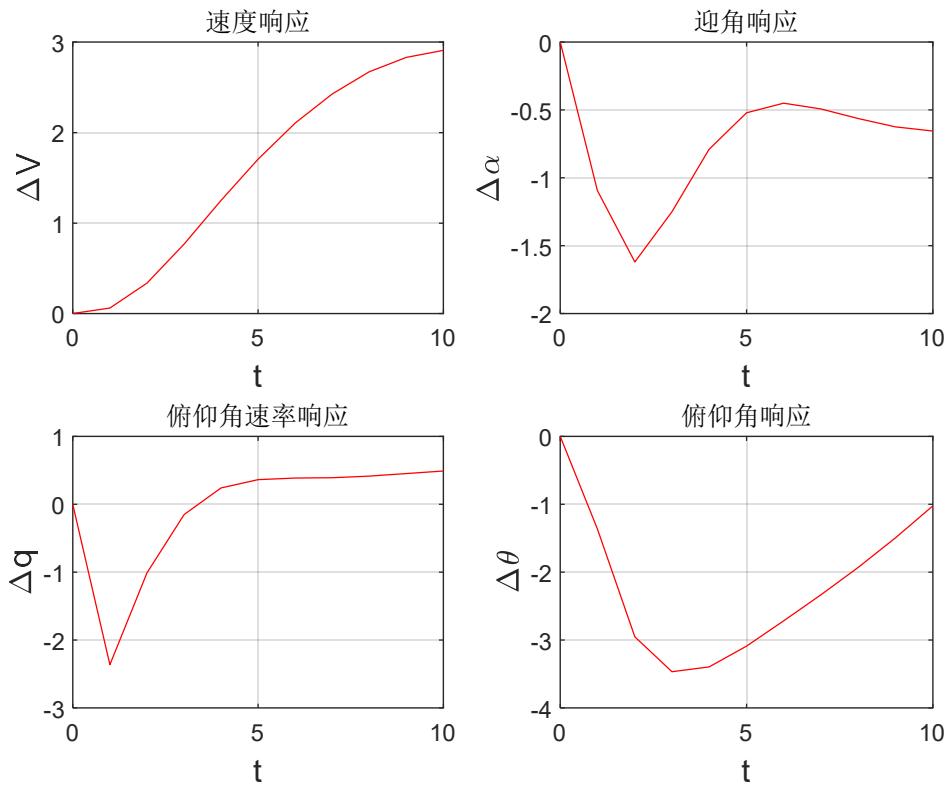


## 2. 仿真参数页设置练习：

变步长仿真结果曲线如上图，以下分别为采样时间分别为 0.01 秒和 1 秒时的定步长仿真结果曲线。



采样时间为 0.01 秒的定步长仿真结果



采样时间为 1 秒的定步长仿真结果

可以看出，定步长仿真的采样时间越长，曲线越折线化，误差越大。

### 3. 扰动设置练习：

代码和结果如下

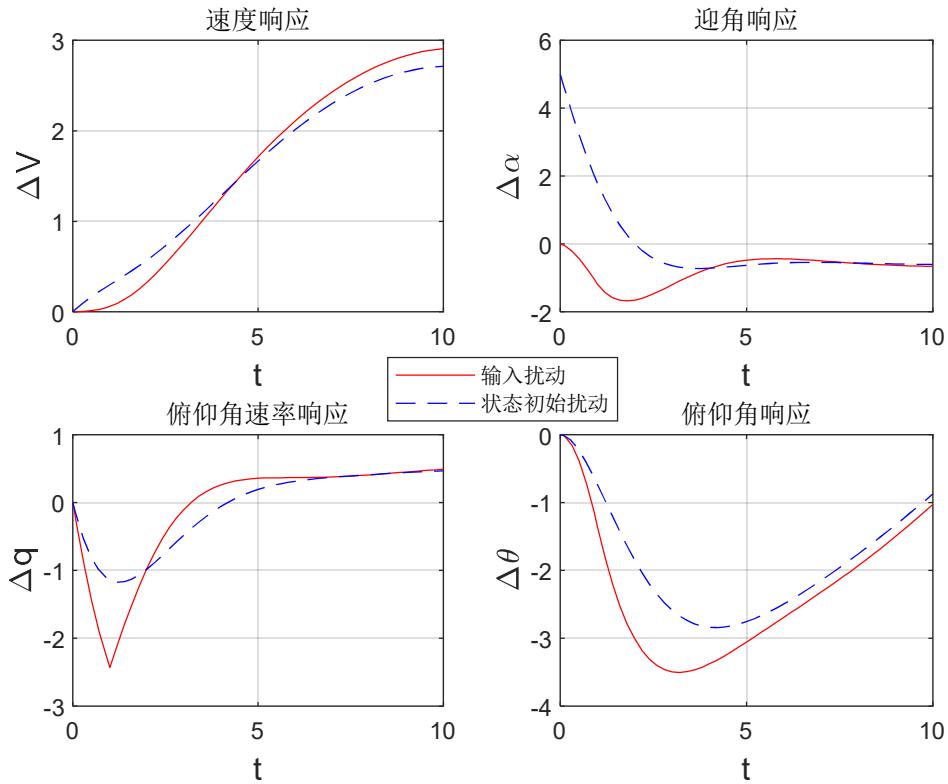
```
%% Simulink模型输出到工作空间画图
% 输入扰动系统响应曲线
figure('Name','系统响应曲线','NumberTitle','off')
subplot(2,2,1);
plot(t.data, DELTA_V.data, 'r'), grid on, hold on
xlabel('t','fontsize',12);
ylabel('\Delta V','fontsize',12);
title('速度响应')
subplot(2,2,2);
plot(t.data, DELTA_alpha.data, 'r'), grid on, hold on
xlabel('t','fontsize',12);
ylabel('\Delta \alpha','fontsize',12);
title('迎角响应')
subplot(2,2,3);
plot(t.data, DELTA_q.data, 'r'), grid on, hold on
xlabel('t','fontsize',12);
ylabel('\Delta q','fontsize',12);
title('俯仰角速率响应')
```

```

subplot(2, 2, 4);
plot(t.data, DELTA_theta.data, 'r'), grid on, hold on
xlabel('t', 'fontsize', 12);
ylabel('\Delta\theta', 'fontsize', 12);
title('俯仰角响应')

% 状态初始扰动系统响应曲线
subplot(2, 2, 1);
plot(t.data, DELTA_V.data, 'b--')
subplot(2, 2, 2);
plot(t.data, DELTA_alpha.data, 'b--')
subplot(2, 2, 3);
plot(t.data, DELTA_q.data, 'b--')
subplot(2, 2, 4);
plot(t.data, DELTA_theta.data, 'b--')
legend({'输入扰动', '状态初始扰动'}, 'Location', 'bestoutside')

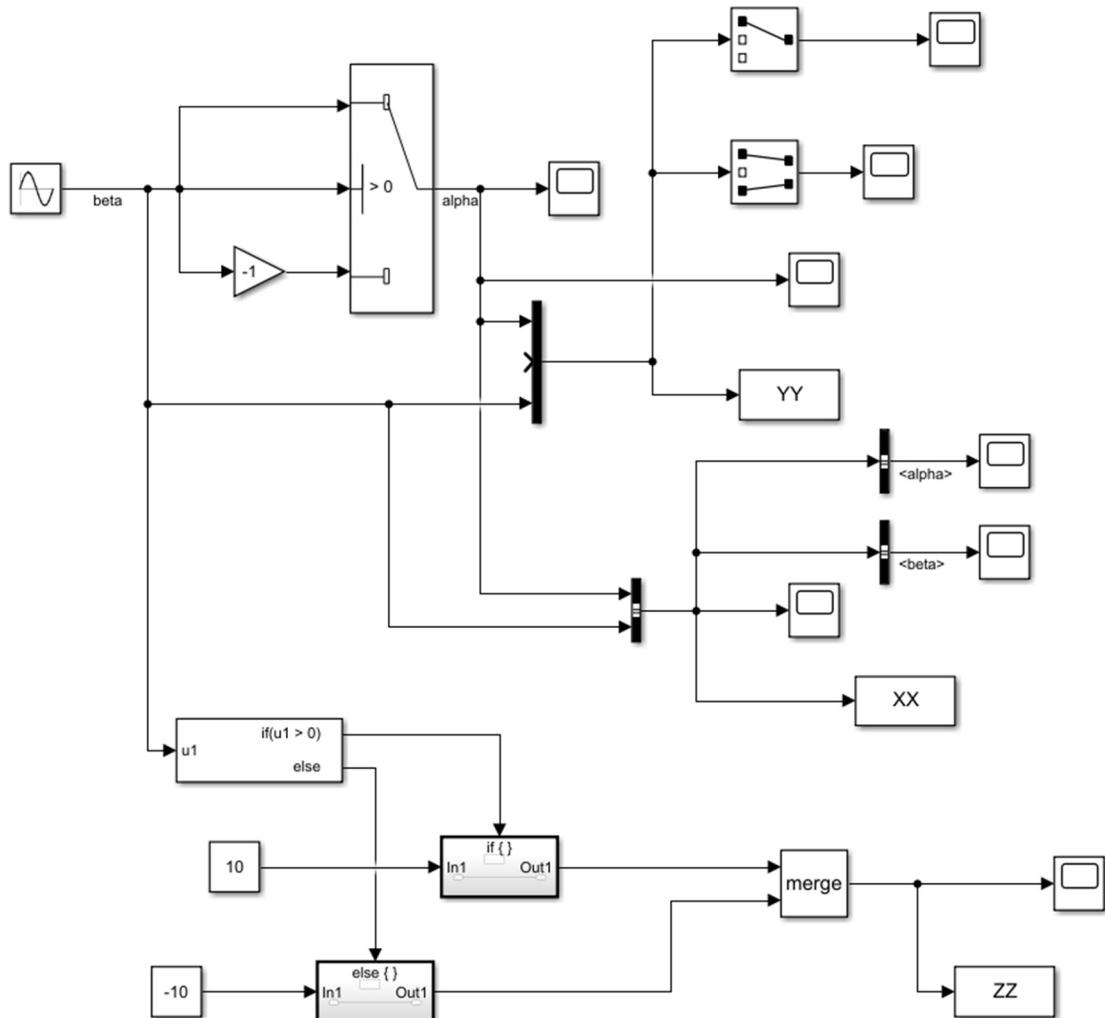
```



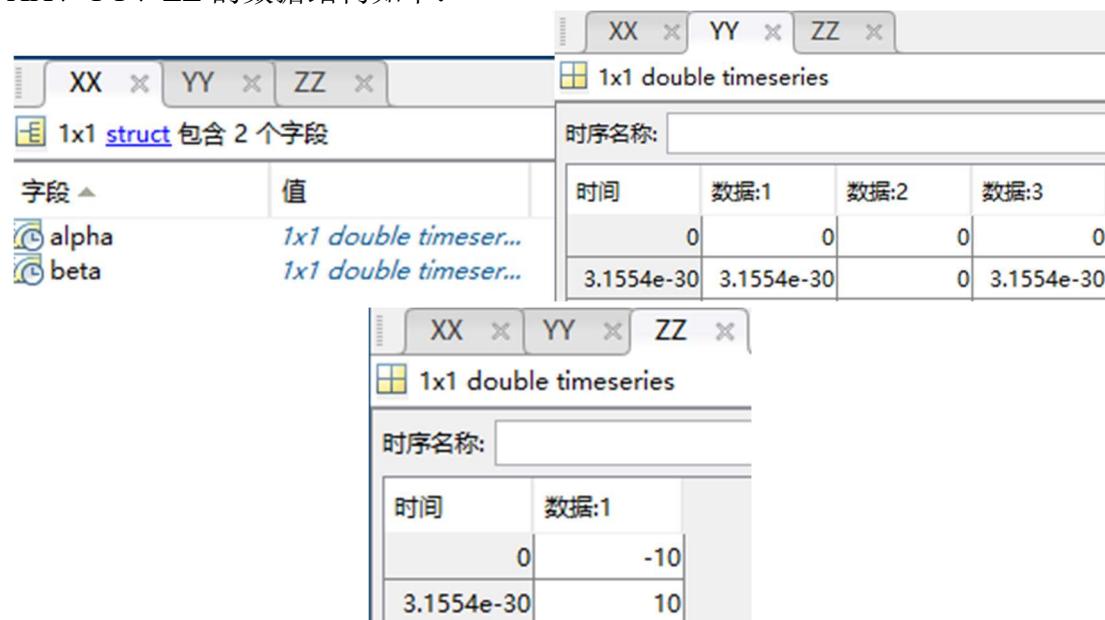
#### 4. Simulink 中信号的合并和拆分练习

注意: Simulink 中信号的合并, 可以采用 Mux, Bus Creator 和 Merge 三种方式, 但 Bus Creator 通常要采用数据名的方式, 需要在创建时在 signal property 设定数据信号的名字(选定一条信号线, 点右键, 在弹出的菜单中选择“property”, 在弹出的对话框“signal name”中填入信号名字)。Merge 是将条件执行子系统模块输出的信号组合成一列信号。Simulink 中信号的拆分, 可以采用 demux、Bus selector 和 Selector 三种方式, 但 Bus Selector 通常要采用数据名的方式, 需要

在创建时在 Block Parameters 选择数据信号名字， Selector 需要在创建时选择数据信号编号。



XX、YY、ZZ的数据结构如下：



可以看出 XX 是包含 2 个单组数据时间序列的结构体，YY 和 ZZ 分别是有 3 组

和 1 组数据的时间序列。时间序列也相当于结构体，可以用点和名称的方式引用，如 XX.alpha.data、YY.data 和 YY.data(:,1) 的运行结果分别如下(未全部显示)：

>> XX.alpha.data	>> YY.data	>> YY.data(:,1)
ans =	ans =	ans =
0	0	0
0.0000	0.0000	0.0000
0.1987	0.1987	0.1987
0.3894	0.3894	0.3894
0.5646	0.5646	0.5646
0.7174	0.7174	0.7174
0.8415	0.8415	0.8415
0.9320	0.9320	0.9320
0.9854	0.9854	0.9854
0.9996	0.9996	0.9996