

MATLAB与SIMULINK 基础



主讲 郭毓

南京理工大学自动化系

1—1 控制系统的实验方法

- 解析法
- 实验法
- 在实际系统中做实验
- 仿真实验法
- 物理模型——“环境相似”、“几何相似”
- 数学模型——“性能相似”

1—2 仿真实验的分类与性能比较

□ 按模型分

物理仿真：实物仿真、实时仿真、半实物仿真、在线仿真

数学仿真：数字仿真、非实时仿真、模拟仿真、离线仿真

□ 按计算机类型分

模拟仿真 模拟计算机 模拟运放

数字仿真 数字计算机

混合仿真

分布式数字仿真（基于网络技术）

1—3 系统、模型与数字仿真

□ 一、控制系统的组成与分类

□ 一般反馈控制系统的组成 如雷达伺服系统

□ 系统的分类：连续、离散（离散时间、离散事件）、连续与离散的混合系统

□ 二、模型的建立及其重要性

□ 物理模型、数学模型

控制系统数字仿真是以“数学模型”为前提的，所以对于仿真结果的“可靠性”而言，系统建模至关重要！！

□ 三、数字仿真的基本内容

□ 建模（系统辨识）>仿真实验>结果分析

1-4 数字仿真软件

- 1) 系统仿真软件的发展
- 程序编制阶段 (Basic, Fortran, C)
- 程序软件包阶段 (仿真语言)
- 交互式语言阶段
(SIMON, CSMP, ACSL, TSIM, ESL ; 1980 MATLAB)
- 模型化图形组态阶段 (1992 Simulink)
- 2) MATLAB
- 3) SIMULINK

1—5 仿真技术的应用与发展（1）

□ 仿真的应用：

- 1 航空与航天工业
- 2 电力工业
- 3 原子能工业
- 4 石油、化工及冶金工业
- 5 非工程领域

1—5 仿真技术的应用与发展（2）

□ 仿真的重要意义

仿真的优点：（1）经济；（2）安全；
（3）快捷

□ 仿真技术的特殊功能

（1）优化技术

（2）预测

1—5 仿真技术的应用与发展（3）

仿真技术的发展趋势

- 硬件方面：基于多CPU并行处理技术的全数字仿真系统；
- 借助网络的分布式数字仿真系统将广为采用；
- 高效能的应用软件；
- 虚拟现实技术的不断完善，为控制系统数字仿真与CAD开辟了一个新时代。
- 虚拟现实技术是一种综合了图形技术、多媒体技术、传感器技术、显示技术以及仿真技术等多种学科而发展起来的高新技术。
- 离散事件系统仿真。

Ch2 控制系统的数学描述

□ 2-1 控制系统的数学模型

一、控制系统的数学模型的表现形式

- 1 微分方程形式
- 2 状态方程形式
- 3 传递函数形式

2-1 控制系统的数学模型 (2)

二、数学模型的转换

1 微分方程与传递函数形式

线性定常SISO系统， $u(t)$ 、 $y(t)$ 分别为系统的输入、输出，

$$a_0 y^{(n)} + a_1 y^{(n-1)} + \cdots + a_{n-1} y' + a_n y = b_0 u^{(m)} + \cdots + b_m u \quad (m < n)$$

输出系数向量 $A = [a_0, a_1, \dots, a_n]$ $n+1$ 维

输出系数向量 $A = [b_0, b_1, \dots, b_m]$ $m+1$ 维

2-1 控制系统的数学模型 (3)

□ 2 状态方程形式

$$\dot{X}(t) = AX(t) + BU(t)$$

$$Y(t) = CX(t) + DU(t)$$

□ 3 传递函数形式

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

□ 4 零极点增益形式 (略)

□ 5 部分分式形式 (略)

2-1 控制系统的数学模型 (4)

三、控制系统建模的基本方法

1、机理法建模

2、实验法建模(系统辨识)

3、混合法建模

Ch3 MATLAB与SIMULINK基础

□ 简单数学计算

```
80*0.89+16*12.2+25*1.82
```

结果 ans=

```
311.9000
```

```
res=80*0.89+16*12.2+25*1.82
```

结果 res=

```
311.9000
```

```
result=res*100
```

结果 result=

```
3.1190e+004
```

MATLAB语言简介

一、*MATLAB*语言简介

帮助: `help` `help eig` `help bode`

二、矩阵有多种输入方法

变量名列表 = 表达式

- (1) 直接在表达式中列出元素:

$A = [1, 2; 3, 4], B = [10, 20, 30],$

$C = [-1; -2; -3], D = [1; 2; 3], E = [7, 8; 9, 6]$

- (2) 建立M文件时, 如果矩阵过大, 可以用文件来产生。

其中x.mat可用`load x.mat` 装入MATLAB中:

`save 'filename.mat' A` `load 'filename'`

MATLAB基本矩阵运算

- (3) 从小矩阵扩充成大矩阵，或用 ":" 从大矩阵中提取小矩阵，例如：
- $A = [1, 2; 3, 4]$,
 $B = [A; [10, 14]]$, $C = B(2:3, 1:2)$, $D = B(1, :)$,
 $E = B(:, 2)$

三、矩阵的基本运算

加法: $C=A+B$, $C=A-B$ 。

转置: $C=A'$, if A 为复矩阵, 则 C 为 A 的共轭转置矩阵。

翻转: 翻转命令有`fliplr`(列), `flipud`(行)和`rot 90`(逆时针转90度)

举例: $A=[1,4;9,7]$, $B=fliplr(A)$, $C=flipud(A)$, $D=rot90(A)$

乘法: $C=A*B$

Kronecker积: $C=kron(A,B)$

举例: $A=[1,2]$, $B=[-1;2]$, $C=kron(A,B)$

KRON(X,Y) 解释

- if X is 2 by 3, then $\text{KRON}(X,Y)$ is

$$\begin{bmatrix} X(1,1)*Y & X(1,2)*Y & X(1,3)*Y \\ X(2,1)*Y & X(2,2)*Y & X(2,3)*Y \end{bmatrix}$$

MATLAB矩阵运算

- **除法**：分左除和右除两种：左除用" \backslash "实现， $A \backslash B$ 相当于 $A^{-1}B$
除用" $/$ "实现， A/B 相当于 BA^{-1}
- **乘方**：如果 A 是方阵，则 A^x 表示 A 的 x 次方，其中 x 可以是正整数、负整数和分数；
- **矩阵的点运算**：这是MATLAB定义了一种特殊的运算，即两矩阵的对应元素直接相乘。
举例： $A=[1,2;3,4]$, $B=[7,8;9,10]$, $C=A.*B$
- **对矩阵部分元素的赋值与运算**：
举例：
 $A=[1,2;4,5]$, $B=A$; $B(2,2)=100$, $C=B$; $C(2,3)=200$

MATLAB命令

- 多条命令可放在一行中，之间用逗号或者分号隔开。
- 逗号“，”告诉MATLAB显示结果，分号“；”表示禁止。
- 如果在一行无法写下一个完整的命令，可以在行尾加入三个连续的点，表示命令余下的部分在下一行出现。
- “%”表示注释。

四、MATLAB控制语句(1)

- MATLAB提供的控制语句与一般高级语言类似。

1 循环语句: for和while语句, 它们允许多级嵌套和互相嵌套
for 循环变量=表达式1(初值):表达式2(增量):表达式3(终值)
循环语句组

end

说明: 增量为1时可以省略

举例: $N=5;$

```
for I=1:N,  
    for J=1:N,  
         $A(I,J)=1/(I+J-1);$   
    end  
end  
A
```

注: 文件名: xxxx.m

四、MATLAB控制语句(2)

- while (条件式)
 循环语句组

end

- 举例：

```
sum=0;i=1;
```

```
while(i<=5)
```

```
    sum=sum+1;i=i+1;
```

```
end
```

```
sum
```

注：文件名：xxxx1.m

四、MATLAB控制语句(3)

- 2 条件转移语句
- if (条件式)
- 语句组
- end
- if (条件式1)
- 语句组1
- else
- 语句组2
- end
- if (条件式1)
- 语句组1
- elseif(条件式2)
- 语句组2
- end

四、MATLAB控制语句(4)

□ 3 break语句

循环过程的终止由break来完成

举例: `sum=0;`

`for i=1:10`

`if (sum>5) break;end`

`sum=sum+1;`

`end`

五、图形功能(1)

- MATLAB 的二维图形功能: `plot(x1,y1,选项1,x2,y2,选项2,.....)`

MATLAB绘图命令的各种选项:

'-' 实线 ':' 点线 'r' 红色 '*' 星号 'o'

圆圈

" 虚线 'g' 绿色 'y' 黄色 '.' 点号 'b'

兰色

'x' 叉号

五、图形功能(2)

□ 举例:

```
t=0:0.1:10;x=sin(t);y=cos(t);plot(t,x,t,y,':');grid;xlabel('X axis');  
ylabel('Y axis');title('My plot');text(2.4,0.85,'x=sin(t)');
```

□ 二维图形函数中，还有其它常用的函数：
polar(theta,rho,选项): 绘制极坐标曲线。
bar(x,y,选项): 绘制直方图。

□ subplot(m,n,k): 在一个窗口中开出 $m \times n$ 个图形窗口，k为窗口序号。

五、图形功能(3)

- `loglog(x,y,选项)`: 绘制对数坐标的曲线
- `semilogx(x,y,选项)`: 绘制横坐标为对数坐标的曲线
- `semilogy(x,y,选项)`: 绘制纵坐标为对数坐标的曲线
-
- `x=logspace(x1,x2,n)`: 按对数等间距地产生一个向量, 例如要产生0.01~100的100个点, 则用
$$x=\text{logspace}(-2,2,100)$$
- 三维曲线图的绘制举例

```
t=0:0.1:40;x=sin(t).^3;y=cos(t).^3;z=t;plot3(x,y,z);  
[x,y]=meshgrid(-4:0.4:4);z=x.^3+y.^3;mesh(x,y,z);
```

六、微分方程初值问题的数值解法

- 求解常微分方程是控制系统仿真的一个重要方面，
- 在MATLAB中可用函数ode23()和ode45()来实现。它们是变步长的2/3阶和4/5阶的龙格—库塔—费尔别格方法，具有很高的精度、数值稳定性和计算速度。函数具体格式为：
 - $[t,x]=ode23(\text{系统函数名}, t_0, t_f, x_0, tol, trace)$
 - $[t,x]=ode45(\text{系统函数名}, t_0, t_f, x_0, tol, trace)$其中，系统函数名为描述系统状态方程的M函数的名称，并用' '括起来；
tol—指定解的精度，默认的误差限为：
ode23()取0.001，ode45()取0.000001

系统函数名的编写格式

- 系统函数名的编写格式为：function xdoc=函数名(t,x)

例： 写一M函数，存入vdpol.m文件中

```
function xdot=vdpol(t,x)
```

```
xdot(1)=x(2)
```

```
xdot(2)=- (x(1).^2-1)*x(2)-x(1);
```

然后选定初值并在MATLAB中键入命令：

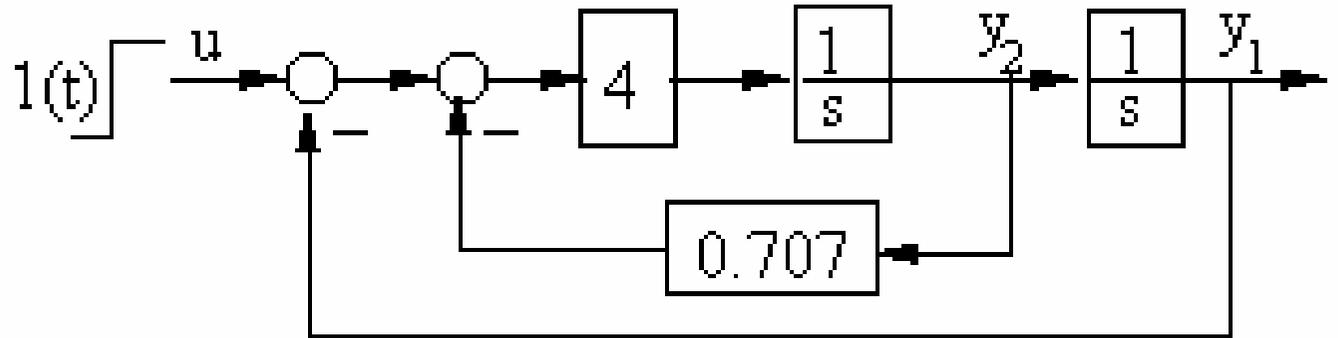
```
x0=[0,0.25]; tf=20; t0=0; [t,x]=ode45('vdpol',t0,tf,  
x0); plot(t,x);
```

七、MATLAB编程实例

- 用MATLAB语句和函数可以编制应用程序。应用程序需要使用两类MATLAB文件；
- 形式：M文件和M函数。这两类文件都是ASCII码文件，其文件后缀名为“.m”；
- M文件相当于主程序，它可以调用MATLAB提供的函数，如eig()等，也可以调用用户自己编制的M函数；
- 在MATLAB环境下，M文件可以直接执行，只要在MATLAB中键入M文件名即可。但M函数必须由其它语句调用，不能直接键入一个文件名来运行一个M文件。

七、MATLAB编程实例（续）

- zmain.m 程序清单:
- `t=0;h=0.01;`
- `tmax=10;`
- `y=[0,0]';`
- `Y=y';`
- `T=t;`
- `while(t<tmax)`
- `ys=y;`
- `k1=zz2(y,1);`
- `t=t+h;`
- `y=ys+h*k1;`
- `k2=zz2(y,1);`
- `y=ys+(h/2)*(k1+k2);`
- `Y=[Y;y'];T=[T;t];`
- `end`
- `plot(T,Y(:,1),T,Y(:,2),'-');`
- `xlabel('t/s');ylabel('y1/y2');`



另一部分为M函数，函数名为zz2.m

```
function ydot=zz2(y,u)
A=[0,1;-4,-2.828];B=[0,4]';
ydot=A*y+B*u;
```

八、控制系统状态方程与传递函数的相互转换

- 由状态方程求传递函数可用：
$$[\text{num}, \text{den}] = \text{ss2tf}(A, B, C, D, iu);$$

其中，对于单输入系统 $iu=1$ ，对于多输入系统，必须对各个输入逐个地求取传递函数子矩阵，最后获得整个传递函数矩阵。

由传递函数求状态方程，为系统实现，不唯一。MATLAB 给出的是可控标准形实现方法，可用

$$[A, B, C, D] = \text{tf2ss}(\text{num}, \text{den})$$

举例：

$$\begin{aligned} n &= [1, 2]; d = [1, 3, 2]; [A, B, C, D] = \text{tf2ss}(n, d) \\ [n1, d1] &= \text{ss2tf}(A, B, C, D) \end{aligned}$$

九、线性系统频率响应和时间响应分析

- MATLAB提供了多种绘制系统频率和时间响应的函数，其中常用的有求bode图、Nyquist图、Nichols图、系统幅值和相位裕度图、阶跃响应step和脉冲响应impulse图的函数，它们分别为：

bode(num,den), Nyquist(num,den),
Nichols(num,den), step(num,den)
impluse(num,den)

bode(A,B,C,D,iu), Nyquist(A,B,C,D,iu),
Nichols(A,B,C,D,iu),

step(A,B,C,D,iu), impluse(A,B,C,D,iu)

margin(A,B,C,D,iu)

[gm,pm,wcg,wcp]=margin(A,B,C,D)

或[gm,pm,wcg,wcp]=margin(num,den)

举例1

□ 开环系统传递函数 $G(s) = \frac{s+2}{s^2+3s+2}$

$n=[1,2], d=[1,3,2]$

`bode(n,d)` `nyquist(n,d)` `nichols(n,d)`

`step(n,d)` `impulse(n,d)`

如果令 $n1=[4], d1=[1,1.5,4]$ 键入

`margin(n1,d1)` 可求得幅值和相位裕量

或 $[Gm, Pm, Wcg, Wcp] = \text{margin}(n1, d1)$

举例2

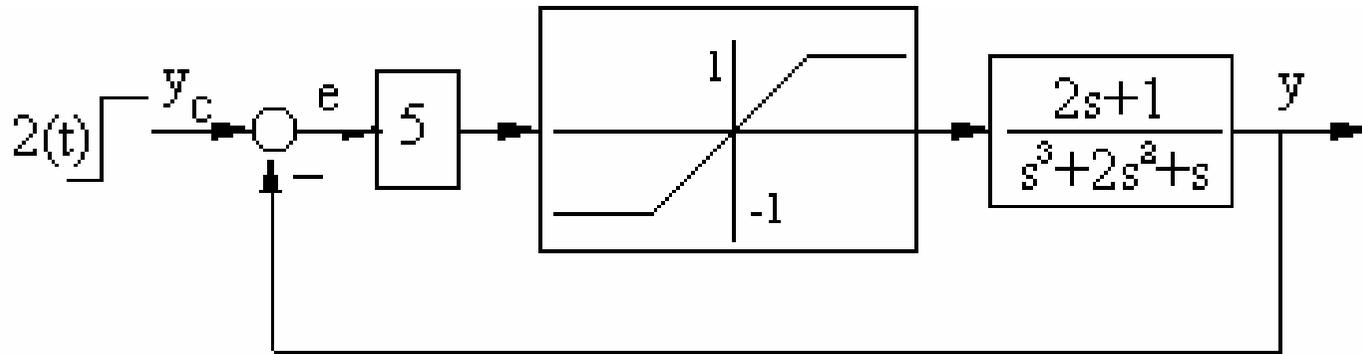
已知某系统的开环传递函数为

$$G(s) = \frac{100(0.2s + 1)}{s^2 (0.01s + 1)(0.25s + 1)} = \frac{80000s + 400000}{s^4 + 140s^3 + 4000s^2}$$

试计算系统开环频率特性，求取系统频率特性指标，并绘制频率特性图。运行pltx.m文件。

举例3

□ Simulink仿真环境



□ 1、Simulink模型库简介

□ 2、仿真实例

文件名: sim1.mdl

□ simulink文件名为: sim1.m

□ 执行

```
plot(t,yc,':',t,y);ylabel('yc/y');xlabel('time(sec)')
```

MATLAB变量

- 变量与数值显示格式
变量必须以字母开头（小于19个字符），区分大小写。
- MATLAB特殊变量
Ans, pi, eps(计算机的最小数), flops(浮点运算数), inf, NaN(不定量), i j ($i=j=\sqrt{-1}$)
- Nargin(函数的输入变量数目), Nargout(函数的输出变量数目), realmin(最小的可用正实数) realmax(最大的可用正实数)
- 如果忘了变量名, 可用“who”命令查询

变量的保存（save和load命令）

变量可以保存为数据文件，也可以从数据文件中读入——用save和load命令。

例： `》 save res result datas`

表示将res和result 保存在二进制数据文件datas.mat中。

如果希望保存为一般的文本文件，可以在save命令之后使用-ascii选项。

将数据读入Load datas，如果是.mat 文件，文件中包含变量名和值，并且可以是多个变量。

如果是非.mat文件，则只能是文本文件，读入的变量名为文件的名称。

变量的清除与复数运算

- 使用“clear”命令可以清除工作空间的变量。
- 例： 》 clear res result
- 复数运算不需要特殊处理，可以直接进行。
- 》 $b = -2 + 3i; a = -1 + 1i;$
- 》 $a * b$

(二)数值显示格式

430/12

- | | | |
|--------------|----------------------------|--------|
| Format long | 35.833 333 333 33334 | 16位 |
| format short | 3.853e+01 | 5位加指数 |
| eformat | 3.583 333 333 333 33 4e+01 | 16位加指数 |
| long eformat | 4041eaaaaaaaaaab | 十六进制 |
| hexformat | 35.83 | 2个十进制位 |
| bankformat+ | + | 正、负或零 |
| format rat | 215/6 | 有理数近似 |
| format short | 35.833 3 | 缺省显示 |

文件管理

- `what` 列出当前目录下所有的M文件
- `dir` 列出当前目录下所有的文件
- `ls` 同`dir`命令
- `type test` 在命令窗口中显示文件`test.m`
- `delete test` 删除文件`test.m`
- `cd path` 进入目录
- `chdir path` 进入目录
- `cd` 显示当前目录
- `chdir` 显示当前目录
- `pwd` 显示当前目录
- `which test` 显示文件`test.m`所在路径