

国家精品课程/ 国家精品资源共享课程/ 国家级精品教材

国家级十一(二)五规划教材/ 教育部自动化专业教学指导委员会牵头规划系列教材

控制系统仿真与CAD

# 第四章 线性系统的数学模型

## 线性系统的传递函数

Transfer Functions of Linear Systems



主讲：薛定宇教授



# 传递函数的MATLAB表示

- 传递函数即放大倍数  $G(s)=Y(s)/U(s)$
- 传递函数的一般表示

$$G(s) = \frac{b_1 s^m + b_2 s^{m-1} + \cdots + b_m s + b_{m+1}}{a_1 s^n + a_2 s^{n-1} + a_3 s^{n-2} + \cdots + a_n s + a_{n+1}}$$

- 传递函数的输入方法

`num=[ $b_1, b_2, \cdots, b_m, b_{m+1}$ ];`

`den=[ $a_1, a_2, \cdots, a_n, a_{n+1}$ ];  $G=tf(num, den);$`



## 例4-2 传递函数输入举例

### ➤ 传递函数模型

#### ➤ 数学模型

$$G(s) = \frac{12s^3 + 24s^2 + 12s + 20}{2s^4 + 4s^3 + 6s^2 + 2s + 2}$$

#### ➤ MATLAB输入语句



```
>> num=[12 24 12 20];  
den=[2 4 6 2 2]; G=tf(num,den)
```

### ➤ 在MATLAB环境中建立一个变量 $G$



## 例4-3 另外一种传递函数输入方法

- 如何处理如下的传递函数？

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

- 定义算子  $s=\text{tf}('s')$ ，再输入传递函数



```
>> s=tf('s');
```

```
G=3*(s^2+3)/(s+2)^3/(s^2+2*s+1)/(s^2+5)
```

- 应该根据给出传递函数形式选择输入方法



## 例4-4 复杂传递函数的输入

### ➤ 输入混合运算的传递函数模型

$$G(s) = \frac{s^3 + 2s^2 + 3s + 4}{s^3(s + 2)[(s + 5)^2 + 5]}$$

显然用第一种方法麻烦，所以



```
>> s=tf('s');
```

```
G=(s^3+2*s^2+3*s+4)/(s^3*(s+2)*((s+5)^2+5))
```

### ➤ 不同方法有不同的适用范围



# MATLAB的传递函数对象

## ➤ 传递函数对象属性：新版本 get(tf)

```
Numerator: {}
Denominator: {}
Variable: 's'
IODelay: []
InputDelay: [01 double]
OutputDelay: [01 double]
Ts: 0
TimeUnit: 'seconds'
InputName: {01 cell}
InputUnit: {01 cell}
InputGroup: [11 struct]
OutputName: {01 cell}
OutputUnit: {01 cell}
OutputGroup: [11 struct]
Notes: [01 string]
UserData: []
Name: ''
SamplingGrid: [11 struct]
```



## 例4-5 传递函数属性修改

- 延迟传递函数  $G(s)e^{-3s}$  , 即  $\tau=3$



```
>> G.ioDelay=3  
    set(G,'ioDelay',3)
```

- 若假设复域变量为  $p$  , 则

```
    set(G,'Variable','p'), G.Variable='p';
```

- 直接赋值方法



```
>> s=tf('s');  
    G=(s^3+2*s^2+3*s+4)/(s^3*(s+2)*((s+5)^2+5))*exp(-3*s)
```



# 多变量系统传递函数矩阵模型

## ➤ 传递函数矩阵

$$G(s) = \begin{bmatrix} g_{11}(s) & g_{12}(s) & \cdots & g_{1p}(s) \\ g_{21}(s) & g_{22}(s) & \cdots & g_{2p}(s) \\ \vdots & \vdots & \ddots & \vdots \\ g_{q1}(s) & g_{q2}(s) & \cdots & g_{qp}(s) \end{bmatrix}$$

- $g_{ij}(s)$  为第  $i$  输出对第  $j$  输入的传递函数
- 可以先定义子传递函数，再由矩阵定义  $G(s)$





## 例4-6 多变量模型

### ➤ 多变量传递函数矩阵模型

$$G(s) = \begin{bmatrix} \frac{0.1134e^{-0.72s}}{1.78s^2 + 4.48s + 1} & \frac{0.924}{2.07s + 1} \\ \frac{0.3378e^{-0.3s}}{0.361s^2 + 1.09s + 1} & \frac{-0.318e^{-1.29s}}{2.93s + 1} \end{bmatrix}$$

### ➤ 输入方法



```
>> G=[tf(0.1134,[1.78 4.48 1]), tf(0.924,[2.07 1]);  
      tf(0.3378,[0.361 1.09 1]), tf(-0.318,[2.93 1])];  
G.ioDelay=[0.72 0; 0.3, 1.29]
```



## 另一种输入方法

### ➤ 先输入各个子传递函数

$$G(s) = \begin{bmatrix} \frac{0.1134e^{-0.72s}}{1.78s^2 + 4.48s + 1} & \frac{0.924}{2.07s + 1} \\ \frac{0.3378e^{-0.3s}}{0.361s^2 + 1.09s + 1} & \frac{-0.318e^{-1.29s}}{2.93s + 1} \end{bmatrix}$$

### ➤ 再按普通矩阵输入的方法输入传递函数矩阵



```
>> g11=tf(0.1134,[1.78 4.48 1],'ioDelay',0.72);  
g12=tf(0.924,[2.07 1]);  
g21=tf(0.3378,[0.361 1.09 1],'ioDelay',0.3);  
g22=tf(-0.318,[2.93 1],'ioDelay',1.29);  
G=[g11, g12; g21, g22]
```



# 传递函数参数提取

- 由于使用单元数组，直接用 `G.num` 不行
- 有两种方法可以提取参数



```
>> [num,den]=tfdata(G,'v')
```

```
num=G.num{1}; den=G.den{1};
```

- 这样定义的优点：可以直接描述多变量系统
  - 第  $i$  输出对第  $j$  输入的传递函数

```
num=G.num{i,j}; den=G.den{i,j};
```



## 传递函数输入小结

- 传递函数的定义与两种输入方法
  - 调用 `tf` 函数直接输入
  - 定义  $s$  算子，再输入传递函数
- 可以提取传递函数的分子与分母 `tfdata`
- 多变量传递函数矩阵的输入
- 有了传递函数模型, 则可以将其输入到 MATLAB 以后则可以对其分析与设计了

