



# 自动控制实验

## 实验一：Matlab简介及传递函数、 数学模型求取



# 内容提要

1

Matlab简介

2

传递函数及数学模型求取



# 内容提要

1

Matlab简介

2

传递函数及数学模型求取



# MATLAB简介

- ❑ Matlab语言是由美国的Cleve Moler博士于1980年开发的，设计者的初衷是为了解决“线性代数”课程的矩阵运算问题
- ❑ MATLAB=Matrix Laboratory，矩阵实验室
- ❑ Matlab将一个优秀软件的易用性与可靠性、通用性与专业性、一般目的的应用与高深的科学技术应用有机的相结合
- ❑ MATLAB是一种直译式的高级语言，比其它程序设计语言容易



# MATLAB简介

- MATLAB语言与其它语言的关系，仿佛和C语言与汇编语言的关系一样

## 计算机语言的发展



标志着计算机语言向“智能化”方向发展，被称为第四代编程语言。



# MATLAB简介

- MATLAB已经不仅仅是一个“矩阵实验室”，它集科学计算、控制仿真、图象处理、声音处理于一身，并提供了丰富的Windows图形界面设计方法
- MATLAB语言是功能强大的计算机高级语言，以超群的风格与性能风靡全世界，成功地应用于各工程学科的研究领域



# MATLAB简介

- MATLAB在美国已经作为大学工科学生必修的计算机语言之一 (C, FORTRAN, ASSEMBLER, MATLAB)
- 近年来, MATLAB语言已在我国推广使用, 现在已应用于各学科研究部门和许多高等院校
- MATLAB语言不受计算机硬件的影响, 目前几乎所有的计算机都可以使用



# MATLAB简介

Matlab能在各领域做什么：

- 工业研究与开发
- 数学教学，特别是线性代数
- 数值分析和科学计算方面的教学与研究
- 电子学、**控制理论**和物理学等工程和科学学科方面的教学与研究
- 经济学、化学和生物学等计算问题的所有其他领域中的教学与研究





# MATLAB简介

## MATLAB语言的特点:

- 语言简洁紧凑，语法限制不严，程序设计自由度大，可移植性好
- 运算符、库函数丰富
- 图形功能强大
- 界面友好、编程效率高
- 扩展性强



# MATLAB简介

## MATLAB语言的功能:

- 强大的数值（矩阵）运算功能
- 广泛的符号运算功能
- 高级与低级兼备的图形功能（计算结果的可视化功能）
- 可靠的容错功能
- 应用灵活的兼容与接口功能
- 信息量丰富的联机检索功能



# 课程思政



**2022年6月27日，在第二十四届中国科协年会闭幕式上，中国科协隆重发布10个对科学发展具有导向作用的前沿科学问题、10个对工程技术创新具有关键作用的工程技术难题和10个对产业发展具有引领作用的产业技术问题**

10个产业技术问题为：

如何建立细胞和基因疗法的临床转化治疗体系？

如何实现存算一体芯片工程化和产业化？

碳中和背景下如何实现火电行业的低碳发展？

如何通过标准化设计，自动化生产，机器人施工和装配式建造系统性解决建筑工业化和高能耗问题？

如何发展自主可控的工业设计软件？

如何利用多源数据实现农作物病虫害精准预报？

如何采用非石油原料高效、安全地合成己二腈？

小麦茎基腐病近年为什么会在我国小麦主产区暴发成灾，如何进行科学有效地防控？

如何研制大型可变速抽水蓄能机组？

如何突破满足高端应用领域需求的高品质对位芳纶国产化卡脖子技术？



# 壮大国产工业软件（创新谈）

**【案例】** 三维制图软件是现代工业设计的重要工具。谈起我国在这一领域的发展历程，不久前，一位大学教授遗憾地提到：约30年前，我国某高校曾自主研发出一款三维制图软件，可不久后，国外企业就大幅降低售价，国外产品起步早、成熟度高，很快就抢占了市场。由于没有用户，这款自主软件最终退出了市场。

当前，我国正加快推动由制造大国向制造强国转变，作为智能制造的关键支撑，工业软件对于推动制造业转型升级具有重要的战略意义。经过多年发展，我国的国产工业软件取得一定的进步，但与国际先进水平相比仍然存在着明显的差距。这其中，既有我国工业软件基础研发薄弱、积累不足等因素，也与缺乏应用有很大关系。



# 壮大国产工业软件（创新谈）

- 工业软件往往不是单个分散的技术，而是一个体系，是各学科知识的集合，需要在生产实践中与各种知识融合，进而更新迭代。
- 工业生产、制造对软件的精度、稳定性、可靠性等要求极高，这些都需要在应用中不断调整、扩充、完善，不可能一步到位。
- 国际主流的工业软件产品，无不是通过不断试错来打磨升级技术，经过数十年“用”中沉淀后，获得行业的认可。一定程度上可以说，“用”是工业软件之母。



# 壮大国产工业软件（创新谈）

## □ 壮大国产工业软件，首先应坚定自主创新信念

➤ 高端的工业软件要不来、买不来、讨不来，唯有从基础做起，写好一行行代码，做好一个个应用模块。更为关键的，是要为国产工业软件打造良好的应用生态。我国有市场规模大、应用场景丰富的优势，持续的回报能助推技术持续升级换代。一款国产自主软件，或许1.0版本的界面还需要完善、稳定性也有待提升，但只要有越来越多的人敢于吃螃蟹，更多的工程师参与丰富生态，它就能不断优化性能，走向成熟。

➤ 此外，必须要尊重知识产权，自觉抵制盗版软件。使用廉价的盗版产品，看上去省了钱占了便宜，长远来看，实际上打击的是自主创新的积极性，破坏的是产业良性发展的根基。



# 壮大国产工业软件（创新谈）

**□ 壮大国产工业软件，另一个关键点在于必须加强产学研协同合作。**

科研人员写出核心代码、搭建出框架，对工业软件来说，只是从0到1的突破，要发挥价值还需不断在“用”中提高稳定性、实用性、成熟度。这就需要通过产学研合作，发挥科研院所、企业等各方力量，提炼核心技术难点以及行业关键问题，各方协同攻关，共同打造出自主可控的国产工业软件。

*摘自《人民日报》（2020年06月29日 19版）*



# MATLAB工具箱 ( Toolbox )

□ MATLAB主工具箱

□ 符号数学工具箱

□ SIMULINK仿真工具箱

□ 控制系统工具箱

□ 信号处理工具箱

□ 图象处理工具箱

□ 通讯工具箱

□ 系统辨识工具箱

□ 神经网络工具箱

□ 金融工具箱

功能型工具  
箱 (通用型)

领域型工具  
箱 (专用型)

许多学科，在MATLAB中都有专用工具箱，现已有30多个工具箱，但MATLAB语言的扩展开发还远远没有结束，各学科的相互促进，将使得MATLAB更加强大





# 控制系统工具箱

## Control System Toolbox

- 连续系统设计和离散系统设计
- 状态空间和传递函数以及模型转换
- 时域响应（脉冲响应、阶跃响应、斜坡响应）
- 频域响应（Bode图、Nyquist图）
- 根轨迹、极点配置



# 通用工具箱

## 1. Matlab主工具箱

- 数值计算、符号运算、绘图以及句柄绘图都是Matlab主工具箱的内容，是Matlab的基本部分，也是本课程实验的重点。
- Matlab主工具箱位于：c:\matlab\toolbox\matlab
- Matlab主工具箱是任何版本的Matlab都不可缺少的。
- 除toolbox\matlab之外的工具箱，在比较完整的专业版Matlab语言中有20多个工具箱。这些工具箱是需要单独选择购买的。



# 通用工具箱

## 1. Matlab主工具箱

- Matlab主工具箱共有31个函数库

datafun —— 数据分析函数库  
sonnds —— 声音处理函数库  
dde —— 动态数据交换函数库  
elfun —— 初等数学函数库  
specmat —— 特殊矩阵函数库  
elmat —— 初等矩阵和时间函数库  
funfun —— 函数功能和数学分析函数库  
general —— 通用命令函数库  
graphics —— 通用图形函数库  
iofun —— 底层输入输出函数库  
lang —— 语言结构函数库

matfun —— 矩阵线性代数函数库  
ops —— 运算符和逻辑函数库  
plotxy —— 二维绘图函数库  
plotxyz —— 三维绘图函数库  
color —— 颜色和光照函数库  
polyfun —— 多项式函数库  
sparfun —— 稀疏矩阵函数库  
strfun —— 字符串函数库  
demos —— Matlab演示函数库



# 通用工具箱

## 1. Matlab主工具箱

Matlab6 新增函数库:

uitools —— 图形界面函数库

datatypes —— 数据类型函数库

graphics —— 句柄绘图函数库

graph3d —— 三维绘图

各函数库中的函数可用[help 函数库名](#)查询, 或[type 函数名](#)方法查询

例: help plotxy

Two dimensional graphics.

Elementary X-Y graphs.

plot - Linear plot.

loglog - Log-log scale plot.

semilogx - Semi-log scale plot.

semilogy - Semi-log scale plot.

fill - Draw filled 2-D polygons.

Graph annotation.

title - Graph title.

xlabel - X-axis label.

ylabel - Y-axis label.

text - Text annotation.

gtext - Mouse placement of text.

grid - Grid lines.

See also PLOTXYZ, GRAPHICS.



# 通用工具箱

## 2. 符号运算工具箱

- 主要功能以符号为对象的数学。
- 在大学教学中，符号数学是各专业都能用到的。
- 符号运算无须事先对独立变量赋值，运算结果以标准的符号形式表达。
- 特点：
  - ① 运算对象可以是没赋值的符号变量
  - ② 可以获得任意精度的解



# 通用工具箱

## 符号运算的功能

- 符号表：达式、符号矩阵的创建
- 符号可变精度求解
- 符号线性代数
- 因式分解、展开和简化
- 符号代数方程求解
- 符号微积分
- 符号微分方程



# 通用工具箱

## 符号运算举例

例如：

$z = 'a*t^2+b*t+c';$

$r = \text{solve}(z, 't')$  —— 对缺省变量  
求解

$r =$

$[1/2/a*(-b+(b^2-4*a*c)^{(1/2)})]$

$[1/2/a*(-b-(b^2-4*a*c)^{(1/2)})]$

对任意变量求解

$r = \text{solve}(z, 'b')$

$r =$

$-(a*t^2+c)/t$

$r = \text{solve}(z, 'c')$

$r =$

$-a*t^2-b*t$

$r = \text{solve}(z, 'a')$

$r =$

$-(b*t+c)/t^2$



# 通用工具箱

## 3. Simulink动态仿真工具箱

### □ Simulink工具箱简介

- Simulink 是实现动态系统建模、仿真和分析的一个集成环境，使得Matlab的功能得到进一步扩展，它可以非常容易的实现可视化建模，把理论研究和工程实践有机的结合在一起。
- 大部分专用工具箱只要以Matlab主包为基础就能运行，有少数工具箱（通讯工具箱、信号处理工具箱等）则要求有Simulink工具箱的支持。
- 由于Matlab和Simulink是集成在一起的，因此用户可以在两种环境下对自己的模型进行仿真、分析和修改。
- 不用命令行编程，由方框图产生m文件(s函数)。
- 当创建好的框图保存后，相应的m文件就自动生成，这个.m文件包含了该框图的所有图形及数学关系信息。
- 框图表示比较直观，容易构造，运行速度较快。





# 通用工具箱

## 3. Simulink动态仿真工具箱

### □ Simulink优点

- **适应面广：** 包括线性、非线性系统；离散、连续系统；定性系统。
- **结构和流程清晰：** 以方块图形式呈现，仿真精细、贴近实际
- **可实现多类仿真：** 物理仿真；计算机仿真；半实物仿真；虚拟仿真；构造仿真

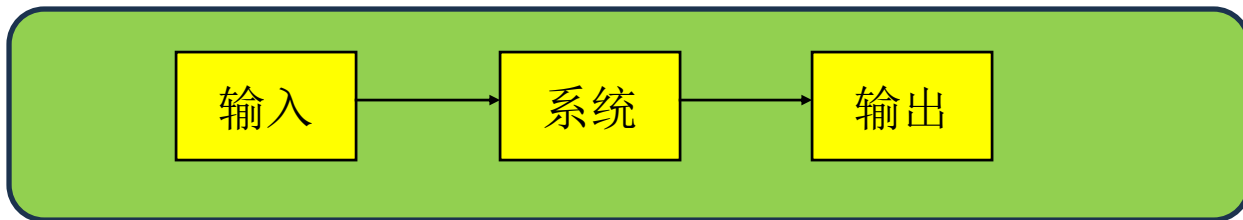


# 通用工具箱

## 3. Simulink动态仿真工具箱

### □ Simulink模型

- Simulink模型在视觉上表现为方框图，在文件上则是扩展名为m的ASCII代码（Matlab7是扩展名为mdl的ASCII代码）；
- 在数学上体现为一组微分方程或差分方程；
- 在行为上模拟了物理器件构成的实际系统的动态特性。
- **Simulink 的一般结构：**





# 通用工具箱

## 3. Simulink动态仿真工具箱

### □ Simulink仿真原理

- 当在框图视窗中进行仿真的同时，Matlab 实际上是运行保存于Simulink内存中s函数的映象文件，而不是解释运行该m文件。
- s函数并不是标准m文件，它可以是m文件，也可以是C或C++程序，通过一定的规则让Simulink的模型或模块能够被调用。

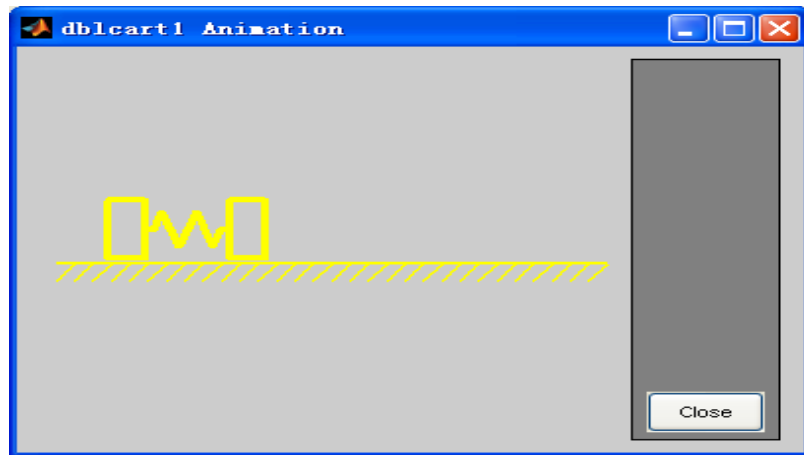
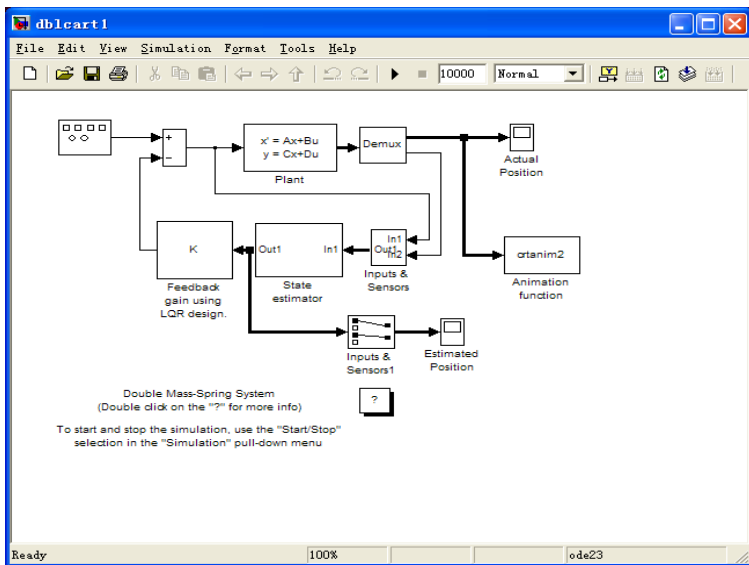


# 通用工具箱

## 3. Simulink动态仿真工具箱

### □ Simulink仿真举例

➤ 【举例】连体弹簧振子运动仿真模型： >> dblcart1



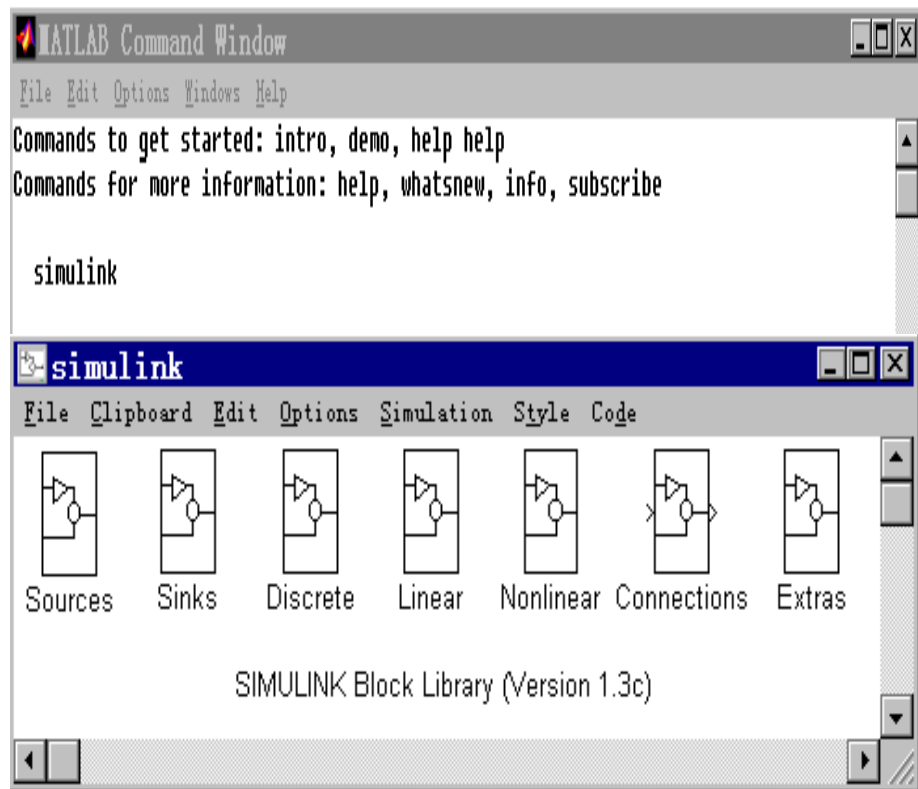
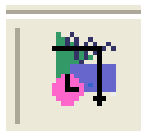


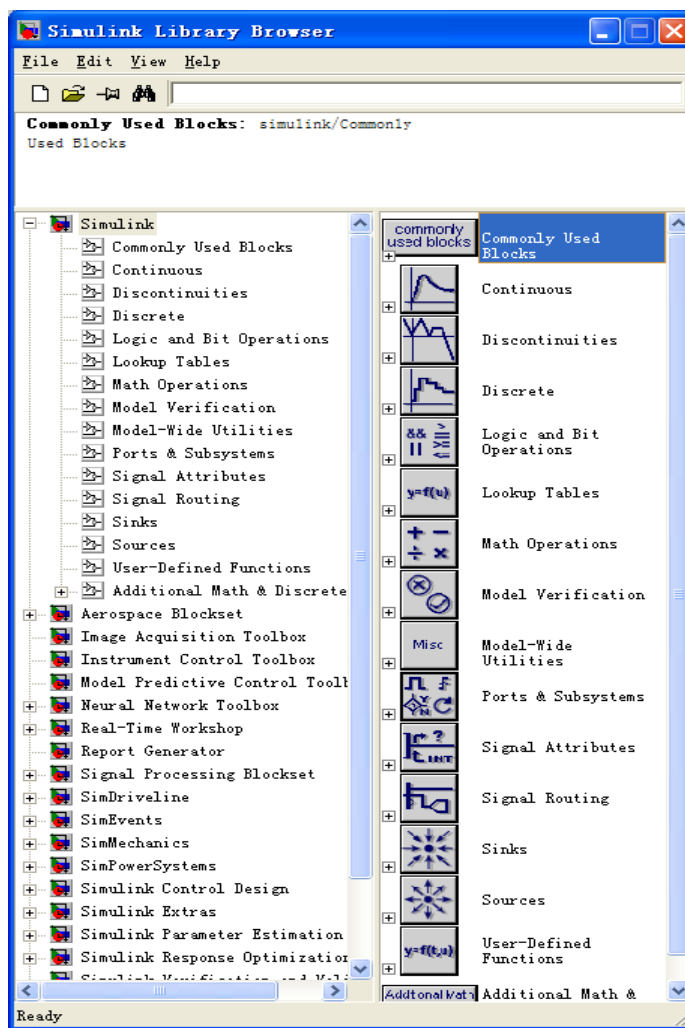
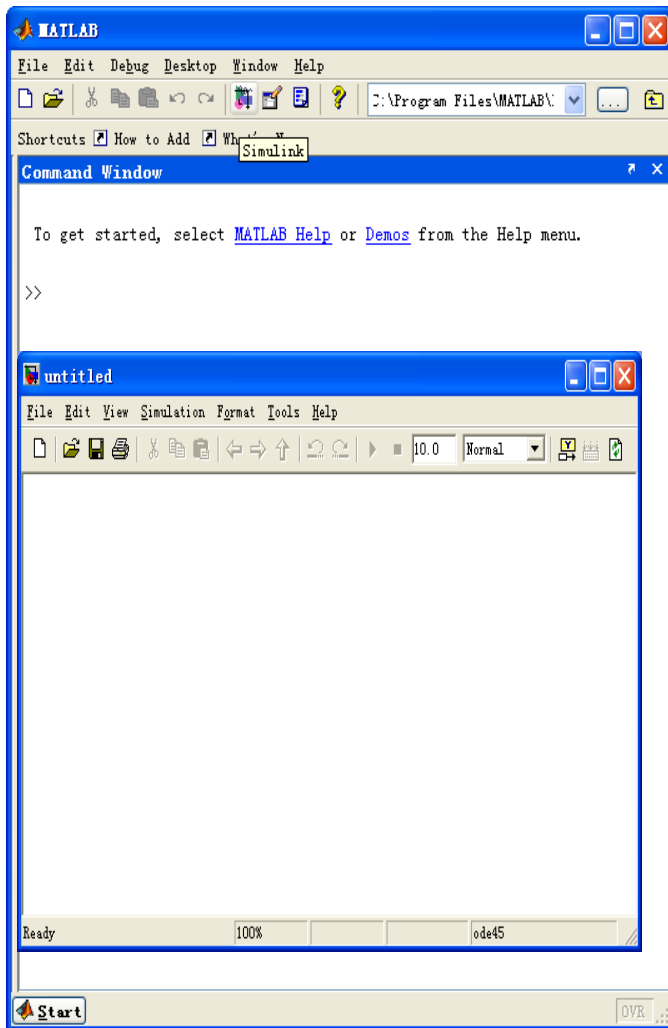
# 通用工具箱

## 3. Simulink动态仿真工具箱

### □ Simulink结构图创建方法

- 一个动态系统的创建过程，就是一个方框图的绘制过程
- 具体步骤：在matlab命令窗口键入 simulink 或者 单击 simulink快捷键







# 通用工具箱

## 3. Simulink动态仿真工具箱

### □ 【举例】

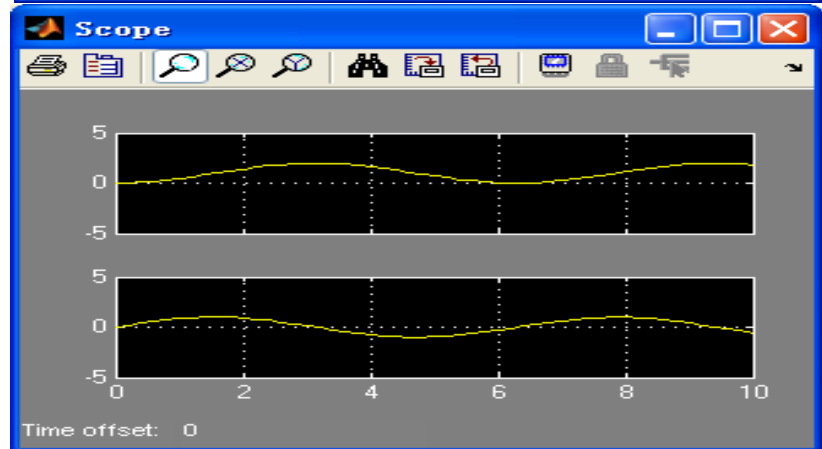
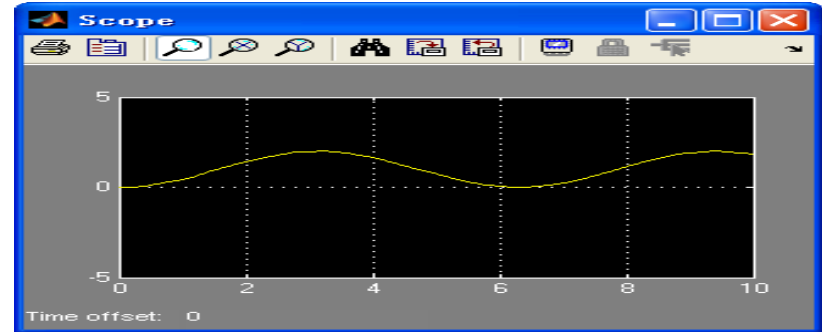
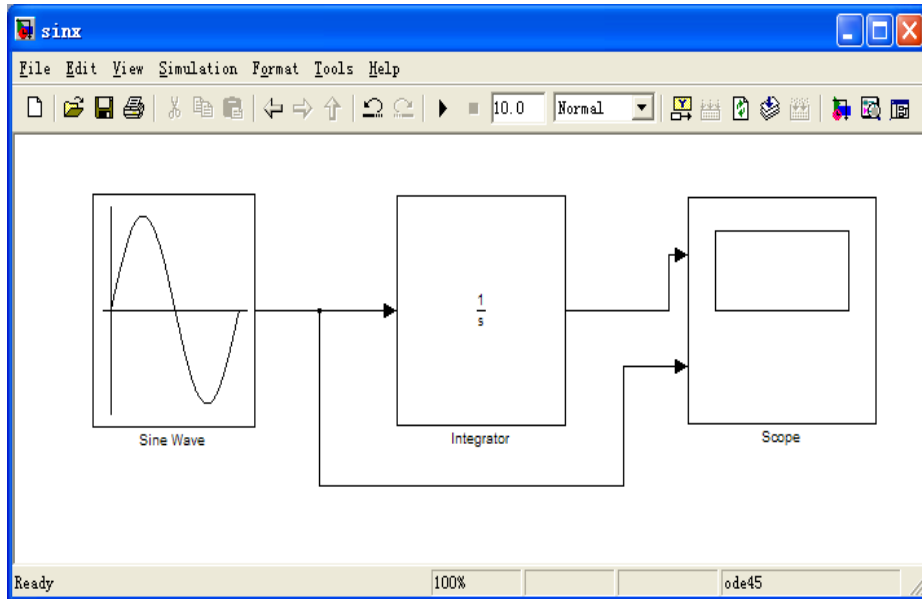
对满足  $\frac{dx}{dt} = \sin t$   
 $x(0) = 0$

数学规律的过程进行仿真，  
结果用示波器来显示，仿真  
时间t为10个单位。

【解题思路】：本题需  
要积分模块(积分模块的  
输入为  $\dot{x}$ ，输出为x)、  
正弦波模块作为数学处  
理的模块。



# 通用工具箱







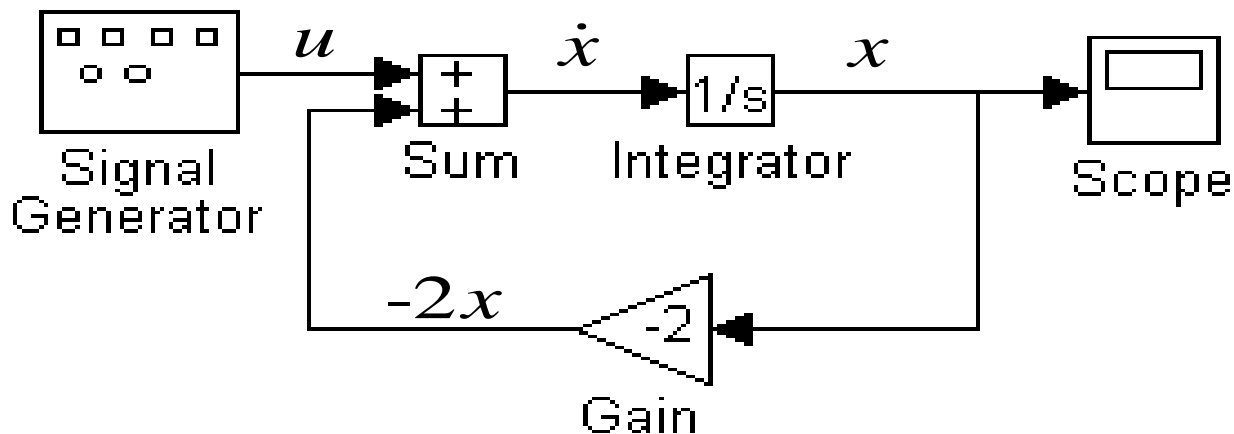
# 通用工具箱

## 3. Simulink动态仿真工具箱

### □ 【举例】

模拟一个微分方程

$$\dot{x} = -2x + u$$





# 通用工具箱

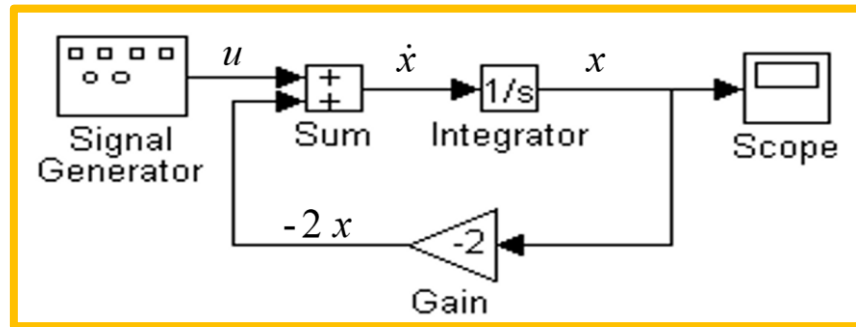
## 3. Simulink动态仿真工具箱

### □ 【举例】

模拟一个微分方程

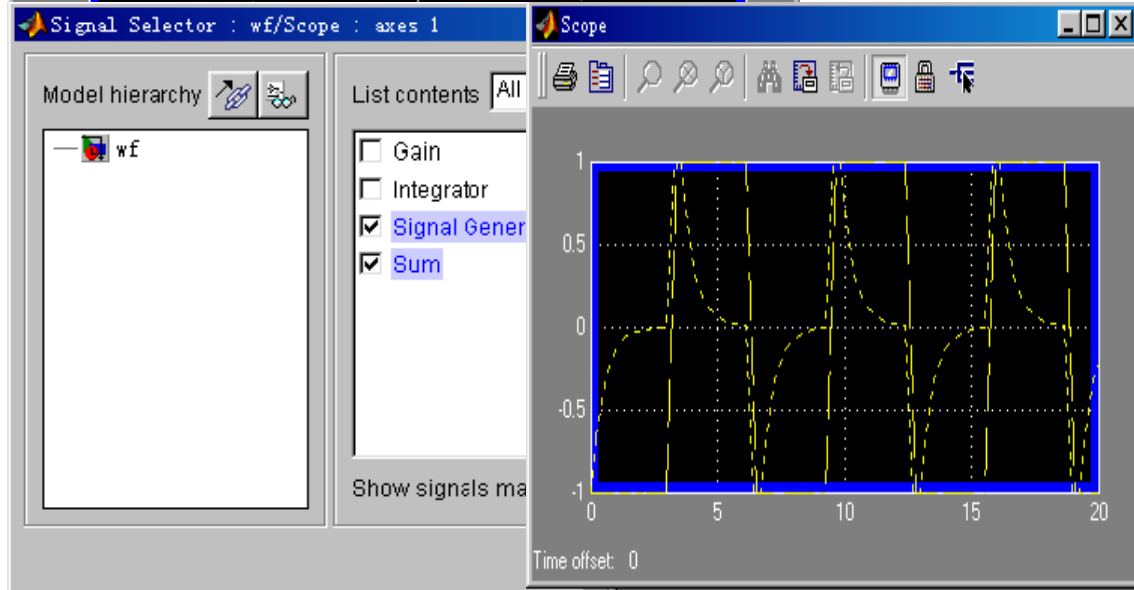
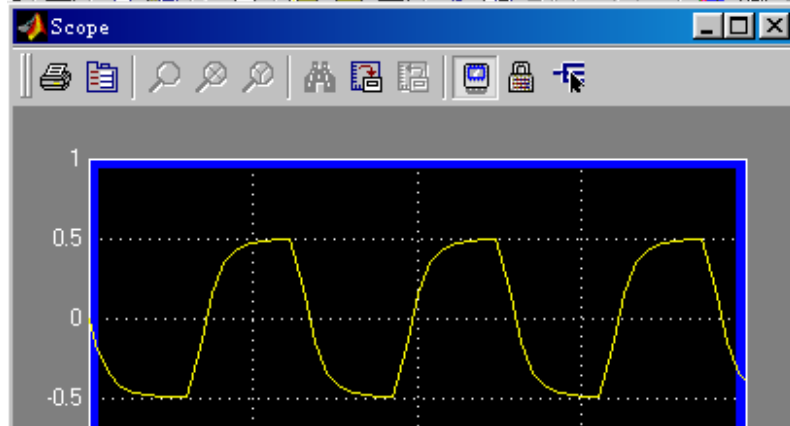
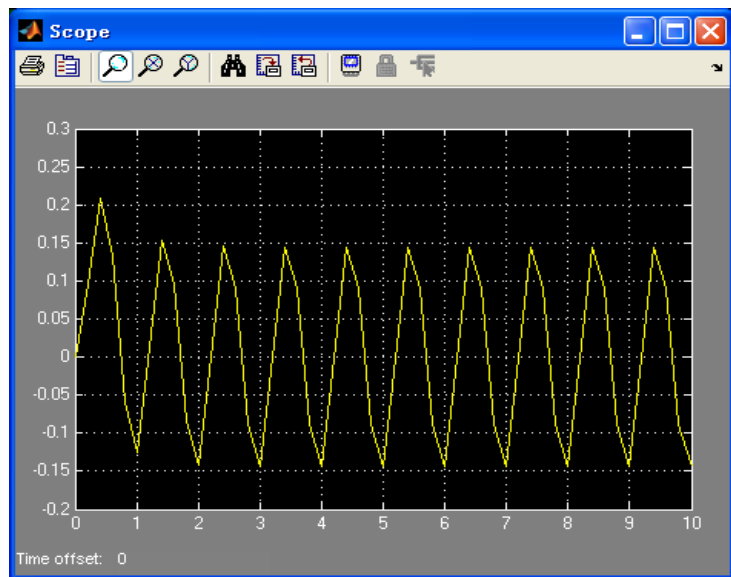
$$\dot{x} = -2x + u$$

- 方框图绘制完毕，一个动态系统模型也就创建好了。
- 选择File菜单Save保存图形，就自动生成一个可在Matlab命令窗口运行的m文件。
- 然后就可使用Simulation菜单中Start开始仿真了。





## 仿真结果：





# 内容提要

1

Matlab简介

2

传递函数及数学模型求取



# 传递函数及数学模型求取

## 1. 实验目的

- ❑ 熟悉MATLAB实验环境，掌握MATLAB命令行窗口的基本操作。
- ❑ 掌握MATLAB建立控制系统数学模型的命令及模型相互转换的方法。
- ❑ 掌握使用MATLAB命令化简模型基本连接的方法。
- ❑ 学会使用Simulink结构图模型化简复杂控制系统模型的方法。



# 传递函数及数学模型求取

## 2.实验任务/要求

- 控制系统模型的建立
- 控制系统模型间的相互转换
- 控制系统模型连接后的等效传递函数



# 传递函数及数学模型求取

## 3.实验仪器、设备及材料

- 计算机
- MATLAB软件

## 4.实验原理

- MATLAB软件的使用
- 使用MATLAB软件在计算机上求取系统的传递函数



# 传递函数及数学模型求取

## 5.实验步骤

### □ 控制系统模型的建立

- **控制系统常用的数学模型有4种：**传递函数模型（tf对象）、零极点增益模型（zpk对象）、结构图模型和状态空间模型（ss对象）。
- 经典控制理论中，数学模型一般使用前3种模型，状态空间模型属于现代控制理论范畴。





# 传递函数及数学模型求取

## 5.实验步骤

### □ 传递函数模型（也称为多项式模型）

连续系统的传递函数模型为

$$G(s) = \frac{b_0 s^m + b_1 s^{m-1} + \cdots + b_m}{a_0 s^n + a_1 s^{n-1} + \cdots + a_n} = \frac{\text{num}(s)}{\text{den}(s)}, n \geq m$$

在MATLAB中用分子、分母多项式系数按s的降幂次序构成两个向量：

$$\mathbf{num} = [b_0, b_1, \cdots, b_m]$$

$$\mathbf{den} = [a_0, a_1, \cdots, a_n]$$

用函数tf( )来输出控制系统的函数，用函数printsys( )来输出控制系统的函数，其命令调用格式为

```
sys=tf(num,den)  
printsys(num,den)
```



# 传递函数及数学模型求取

## 5. 实验步骤

### □ 传递函数模型（也称为多项式模型）

【例1】已知系统传递函数为  $G(s) = \frac{s+3}{s^3+2s^2+2s+1}$ ，建立系统的传递函数模型。

【解】MATLAB程序为：

```
num=[0 1 3];    %分子多项式系数向量  
den=[1 2 2 1];  %分母多项式系数向量  
printsys(num,den) %构造系统传递函数G(s)并输出显示  
运行后命令行窗口显示：
```

num/den =

s + 3

-----  
s^3 + 2s^2 + 2s + 1



# 传递函数及数学模型求取

## 5. 实验步骤

### □ 传递函数模型（也称为多项式模型）

【例2】已知系统传递函数为  $G(s) = \frac{5(s+2)^2(s^2+6s+7)}{s(s+1)^3(s^3+2s+1)}$ ，建立系统的传递函数模型。

【解】方法一：借助多项式乘法函数 来处理。

```
num=5*conv(conv([1,2],[1,2]),[1,6,7]);
```

```
den=conv([1,0],conv([1,1],conv([1,1],conv([1,1],[1,0,2,1]))));
```

```
Gs=tf(num,den)
```

运行后命令行窗口显示：

Gs=

$5s^4 + 50s^3 + 175s^2 + 260s + 140$

-----+-----

$s^7 + 3s^6 + 5s^5 + 8s^4 + 9s^3 + 5s^2 + s$



# 传递函数及数学模型求取

## 5. 实验步骤

### □ 传递函数模型（也称为多项式模型）

【例1】已知系统传递函数为  $G(s) = \frac{5(s+2)^2(s^2+6s+7)}{s(s+1)^3(s^3+2s+1)}$ ，建立系统的传递函数模型。

方法二：用 `s=tf('s')` 命令来定义传递函数的拉普拉斯变换的变量 `s`，然后就可以直接按数学表达式的形式建立系统的传递函数模型。

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

```
Gs=(5*(s+2)^2*(s^2+6*s+7))/(s*(s+1)^3*(s^3+2*s+1))
```

运行后命令行窗口显示：

Gs=

$5s^4 + 50s^3 + 175s^2 + 260s + 140$

-----  
 $s^7 + 3s^6 + 5s^5 + 8s^4 + 9s^3 + 5s^2 + s$



# 传递函数及数学模型求取

## 5. 实验步骤

### □ 传递函数模型（也称为多项式模型）

【练习1】 建立控制系统的传递函数模型

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

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



# 传递函数及数学模型求取

## 5.实验步骤

### □ 零极点增益模型

这是传递函数模型的另一种表现形式，其原理是分别对原传递函数的分子、分母进行因式分解，以获得系统的零点和极点的表示形式。

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

在MATLAB中，用向量  $\mathbf{z}$ ,  $\mathbf{p}$ ,  $\mathbf{k}$  构成向量组  $[\mathbf{z}, \mathbf{p}, \mathbf{k}]$  表示系统，即

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

用函数命令zpk()来建立零极点增益模型，其调用格式为  $\text{sys}=\text{zpk}(\mathbf{z},\mathbf{p},\mathbf{k})$



# 传递函数及数学模型求取

## 5. 实验步骤

### □ 零极点增益模型

【例3】已知系统传递函数为  $G(s) = \frac{10(s+5)}{(s+0.5)(s+2)(s+3)}$ ，建立系统的零极点增益模型。

【解】MATLAB程序为：

```
k=10; %赋增益值，标量  
z=[-5]; %赋零点值，向量  
p=[-0.5 -2 -3]; %赋极点值，向量  
sys=zpk(z,p,k)  
运行后命令行窗口显示：
```

```
sys=  
  
      10(s + 5)  
-----  
(s + 0.5)(s + 2)(s + 3)
```



# 传递函数及数学模型求取

## 5. 实验步骤

### □ 零极点增益模型

【练习2】 建立控制系统的零极点增益模型

$$(1) \quad G(s) = \frac{8(s+1-j)(s+1+j)}{s^2(s+5)(s+6)(s^2-1)}$$

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





# 传递函数及数学模型求取

## 5.实验步骤

### □ 二阶控制系统的标准模型

在MATLAB中，用函数命令**ord2( )** 来建立二阶控制系统的标准模型

$$\frac{1}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

，其函数调用格式为：**[num,den]=ord2(wn,zeta)**

其中，**wn**为二阶系统的无阻尼自然振荡频率，**zeta**为二阶系统的阻尼比。其函数功能是：根据已知的自然频率和阻尼比值，建立连续二阶系统，输出该系统的分子项系数向量**num**和分母项系数向量**den**。



# 传递函数及数学模型求取

## 5. 实验步骤

### □ 二阶控制系统的标准模型

【例4】已知二阶系统的自然频率  $\omega_n=1$ ，阻尼比  $\zeta=0.5$ ，建立其传递函数。

【解】 MATLAB程序为：  
`[num,den]=ord2(1,0.5);`  
`G=tf(num,den)`  
运行后命令行窗口显示：

$$G = \frac{1}{s^2 + s + 1}$$



# 传递函数及数学模型求取

## 5. 实验步骤

### □ 控制系统模型间的相互转换

函数调用格式为：

$[num, den] = zp2tf(z, p, k)$  %零极点模型转换为传递函数模型

$[z, p, k] = tf2zp(num, den)$  %传递函数模型转换为零极点模型

$[r, p, k] = residue(num, den)$  %传递函数模型转换为部分分数展开式模型

$[num, den] = residue(r, p, k)$  %部分分数展开式模型转换为传递函数模型



# 传递函数及数学模型求取

## 5.实验步骤

### □ 控制系统模型间的相互转换

部分分式展开式模型为：

$$G(s) = \frac{num(s)}{den(s)} = \frac{r_1}{s - p_1} + \frac{r_2}{s - p_2} + \cdots + \frac{r_n}{s - p_n} + k(s)$$

$\mathbf{r} = [r_1, r_2, \cdots, r_n]$  部分式展开式的分子常数向量；

$\mathbf{p} = [p_1, p_2, \cdots, p_n]$  部分分式展开式的母极点向量；

$\mathbf{k}$  部分分式展开式的余数向量。



# 传递函数及数学模型求取

## 5. 实验步骤

### □ 控制系统模型间的相互转换

当系统有  $m$  个相同极点 ( $m$  重根) :  $n > m$

$$\mathbf{r} = [r_1, r_2, \dots, r_m, r_{m+1}, \dots, r_n]$$

$$\mathbf{p} = [p_1, p_1, \dots, p_1, r_{m+1}, \dots, p_n]$$

则展开式模型应为:

$$G(s) = \frac{\text{num}(s)}{\text{den}(s)} = \frac{r_1}{s - p_1} + \frac{r_2}{(s - p_2)^2} + \dots + \frac{r_2}{(s - p_2)^m} + \frac{r_2}{s - p_{m+1}} + \dots + \frac{r_n}{s - p_n} + k(s)$$



# 传递函数及数学模型求取

## 5. 实验步骤

### □ 控制系统模型间的相互转换

【例5】将系统  $G(s) = \frac{s^2 + 5s + 6}{s^3 + 2s^2 + s}$  转化为部分分式展开式。

【解】MATLAB 程序为：

```
num=[1,5,6];
```

```
den=[1,2,1,0];
```

```
[r,p,k]=residue(num,den)
```

运行后，可得结果为：分子系数向量  $\mathbf{r} = [-5, -2, 6]$ ；分母极点向量  $\mathbf{p} = [-1, -1, 0]$ ；商（即余数向量） $\mathbf{k} = 0$ 。

转化后的部分分式展开式为：

$$G(s) = \frac{-5}{s+1} + \frac{-2}{(s+1)^2} + \frac{6}{s}$$



# 传递函数及数学模型求取

## 5.实验步骤

### □ 控制系统模型间的相互转换

【练习3】 将系统传递函数转化为部分分式展开式。

(1)

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

(2)

$$G(s) = \frac{8(s+1-j)(s+1+j)}{s^2(s+5)(s^2+1)}$$



# 传递函数及数学模型求取

## 5. 实验步骤

### □ 控制系统模型间的相互转换

【例6】已知系统传递函数  $G(s) = \frac{s^2 + 5s + 6}{s^3 + 2s^2 + s}$ ，求其等效的零极点模型。

【解】建立其零极点模型的 MATLAB 程序为：

```
num=[1,5,6];  
den=[1,2,1,0];  
[z,p,k]=tf2zp(num,den);  
sys=zpk(z,p,k)
```

运行后命令行窗口显示：

Zero/pole/gain:

$(s + 3)(s + 2)$

-----

$s(s + 1)^2 \leftarrow$





# 传递函数及数学模型求取

## 5.实验步骤

### □ 控制系统模型间的相互转换

【练习4】 建立控制系统  $G(s) = \frac{8(s+1)(s+2)}{s(s+5)(s+6)(s+3)}$  的传递函数模型。



# 传递函数及数学模型求取

## 5. 实验步骤

### □ 控制系统模型连接后的等效传递函数

(1) 串联。串联等效的传递函数为各个中间串联环节的传递函数的乘积。当  $n$  个模型 `sys1`、`sys2`、.....、`sysn` 串联时，其等效的传递函数模型为 `sys=sys1*sys2*.....*sysn`；或者使用 `series()` 函数。其调用格式为：

`[num,den]=series(num1,den1,num2,den2)` 或者

`sys=series(sys1,sys2)`

`series()` 函数只能实现两个模型的串联。如果串联模型多于两个，则必须多次使用。



# 传递函数及数学模型求取

## 5. 实验步骤

### □ 控制系统模型连接后的等效传递函数

(2) 并联。并联等效模型为多个环节输出的代数和(有加有减)。

当  $n$  个模型  $\text{sys1}$ 、 $\text{sys2}$ 、.....、 $\text{sysn}$  并联时，其等效的传递函数模型为  $\text{sys}=\text{sys1}+\text{sys2}+\text{.....}+\text{sysn}$ ；或者使用 `parallel()` 函数。其调用格式为：

`[num,den]=parallel(num1,den1,num2,den2)` 或者

`sys= parallel(sys1,sys2)`

`parallel()` 函数只能实现两个模型的并联。如果并联模型多于两个，则必须多次使用。



# 传递函数及数学模型求取

## 5. 实验步骤

### □ 控制系统模型连接后的等效传递函数

【例7】已知三个模型的传递函数为：

$$G_1(s) = \frac{s^2 + 5s + 6}{s^3 + 2s^2 + s}$$

$$G_2(s) = \frac{s^2 + 5s + 6}{s^3 + 2s^2 + s}$$

$$G_3(s) = \frac{s^2 + 5s + 6}{s^3 + 2s^2 + s}$$

试分别用两种方法求出这三个模型串联后的等效传递函数模型。



# 传递函数及数学模型求取

## 5.实验步骤

### □ 控制系统模型连接后的等效传递函数

方法一：

```
Num1=[5];  
den1=[1 1];  
num2=[2 1];  
den2=[1 0];  
num3=[4];  
den3=[3 1];  
[num0,den0]=series(num1,den1,num2,den2);  
[num,den]=series(num0,den0,num3,den3);  
printsys(num,den)
```

程序运行后，命令行窗口显示：

num/den=

$40s + 20$

-----

$3s^3 + 4s^2 + s$



# 传递函数及数学模型求取

## 5. 实验步骤

### □ 控制系统模型连接后的等效传递函数

方法二：

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

```
G1=5/(s+1);
```

```
G2=(2*s+1)/s;
```

```
G3=4/(3*s+1);
```

```
G=G1*G2*G3
```

程序运行后，命令行窗口显示：

num/den=

40s + 20

-----

3s^3 + 4s^2 + s

因此，三个模型串联后的等效传递函数为

$$G(s) = \frac{40s + 20}{3s^3 + 4s^2 + s}。$$



# 传递函数及数学模型求取

## 5.实验步骤

### □ 控制系统模型间的相互转换

【练习5】 分别用两种方法建立例7中三个模型并联后的等效传递函数模型。



# 传递函数及数学模型求取

## 5. 实验步骤

### □ 控制系统模型连接后的等效传递函数

#### (3) 反馈连接

两个环节反馈连接后，其等效传递函数可用 `feedback()` 函数求得。

若  $G(s)$  为闭环前向通道的传递函数 `sys1`， $H(s)$  为反馈通道的传递函数 `sys2`，则 `feedback()` 函数的调用格式为：

```
sys=feedback(sys1,sys2,sign)
```

其中，`sign` 是反馈极性。`sign` 缺省时，默认为负反馈，`sign=-1`；

为正反馈时，`sign=1`。为单位反馈时，`sys2=1`，且不能省略。





# 传递函数及数学模型求取

## 5. 实验步骤

### □ 控制系统模型连接后的等效传递函数

#### (3) 反馈连接

由开环系统构成单位反馈闭环系统时，可使用 `cloop()` 函数求得闭环传递函数，其调用格式为：

`[numc,denc]=cloop(num,den,sign)`

当 `sign=1` 时采用正反馈；当 `sign=-1` 时采用负反馈。`sign` 缺省时，默认为负反馈。



# 传递函数及数学模型求取

## 5. 实验步骤

### □ 控制系统模型连接后的等效传递函数

【例8】已知如下系统，求其负反馈闭环传递函数。

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

$$H(s) = \frac{5(s + 2)}{s + 10}$$



# 传递函数及数学模型求取

## 5.实验步骤

### □ 控制系统模型连接后的等效传递函数

【例8】已知如下系统，求其负反馈闭环传递函数。

【解】MATLAB 程序为：

```
numg=[2 5 1];  
deng=[1 2 3];  
numh=[5 10];  
denh=[1 10];  
[num,den]=feedback(numg,deng,numh,denh);  
printsys(num,den)
```

程序运行后显示：

num/den =

$$2s^3 + 25s^2 + 51s + 10$$

-----

$$11s^3 + 57s^2 + 78s + 40$$



# 传递函数及数学模型求取

## 5. 实验步骤

### □ 控制系统模型连接后的等效传递函数

【练习6】已知系统  $G(s) = \frac{2s+1}{s^2+2s+3}$ ，求其单位负反馈闭环传递函数。



# 传递函数及数学模型求取

## 5.实验步骤

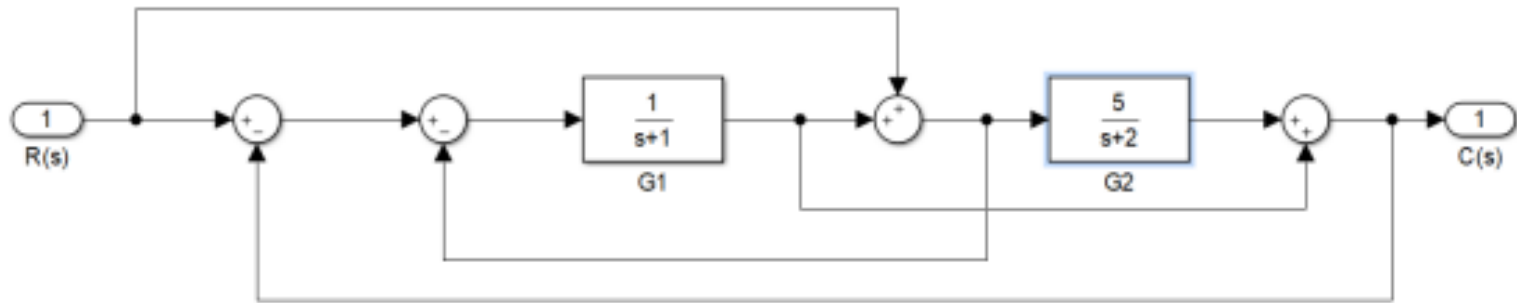
### □ 用Simulink结构图模型化简控制系统模型

【例9】已知系统结构图如图1所示，其中系统闭环传递函数。

$$G_1(s) = \frac{1}{s+1}$$

$$G_2(s) = \frac{5}{s+2}$$

，求





# 传递函数及数学模型求取

## 5.实验步骤

### □ 用Simulink结构图模型化简控制系统模型

【例9】已知系统结构图如图1所示，其中  $G_1(s) = \frac{1}{s+1}$  ，  $G_2(s) = \frac{5}{s+2}$  ， 求系统闭环传递函数。

【解】方法一：用梅逊(Mason)公式求系统的闭环传递函数。

```
syms s G1 G2 phi; %建立符号对象
G1=1/(s+1); %写出G1的传递函数
G2=5/(s+2); %写出G2的传递函数
phi=factor(((G1+1)*G2)/(1+2*G1+G1*G2))
%用Mason公式计算系统的传递函数，并进行因式分解
```

$$\Phi(s) = \frac{C(s)}{R(s)} = \frac{5s+10}{s^2+5s+11}$$



# 传递函数及数学模型求取

## 5.实验步骤

### □ 用Simulink结构图模型化简控制系统模型

【例9】已知系统结构图如图1所示，其中  $G_1(s) = \frac{1}{s+1}$  ，  $G_2(s) = \frac{5}{s+2}$  ，求系统闭环传递函数。

**方法二：**用Simulink结构图模型求系统的闭环传递函数。

结构图模型是描述系统数学模型的又一种直观的形式，因此结构图模型可以转化为传递函数模型。先在Simulink模型窗口中画出系统结构图，将其保存在当前默认路径下，文件名为smg.mdl，然后在MATLAB命令行窗口中输入以下程序，就可以将系统的Simulink结构图模型转换为系统状态空间模型，进而转换为传递函数或零极点增益模型。



# 传递函数及数学模型求取

## 5. 实验步骤

### □ 用Simulink结构图模型化简控制系统模型

【例9】已知系统结构图如图1所示，其中系统闭环传递函数。

$$G_1(s) = \frac{1}{s+1}$$

$$G_2(s) = \frac{5}{s+2}, \text{ 求}$$

MATLAB程序为：

```
[A,B,C,D]=linmod('smg');
```

```
%将结构图模型转化成状态空间模型
```

```
[num,den]=ss2tf(A,B,C,D);
```

```
%将状态空间模型转化成传递函数模型
```

```
printsys(num,den,'s')
```

```
%输出传递函数
```

程序运行结果显示：

num/den =

5s + 10

-----

S^2 + 5s + 11

可见，结果与Mason公式求得的传递函数完全相同。





# 传递函数及数学模型求取

## 6.实验报告要求

- 熟练使用各种函数命令建立控制系统的数学模型。
- 完成实验的例题和自我实践，并记录结果。
- 编写M文件程序，完成简单连接的模型的等效传递函数，并求出相应的零极点。



# 传递函数及数学模型求取

## 7.拓展思考

- 【练习7】已知一个复杂系统结构图如图2所示，试分别使用Mason公式和Simulink结构图模型编程求解系统的等效闭环传递函数。控制系统模型间的相互转换

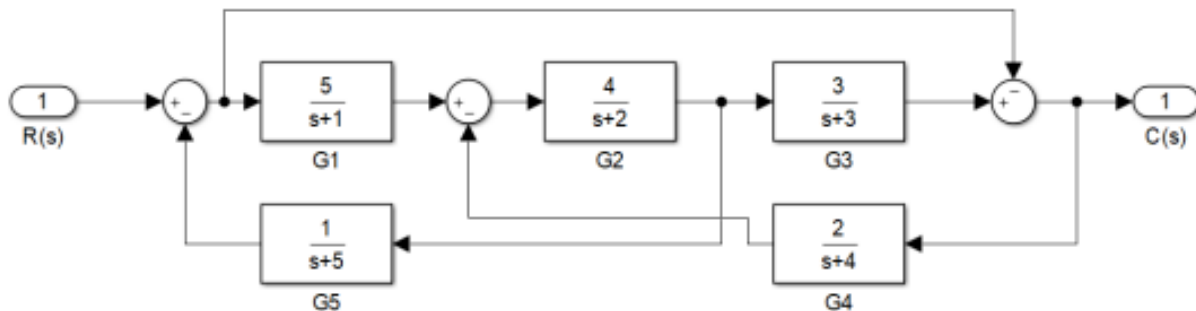


图 2：复杂控制系统结构图



# THE END !

