

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

# 第六章 非线性系统的建模与仿真

## S-函数编程 (上)

S-Function Programming (I)



主讲：薛定宇教授



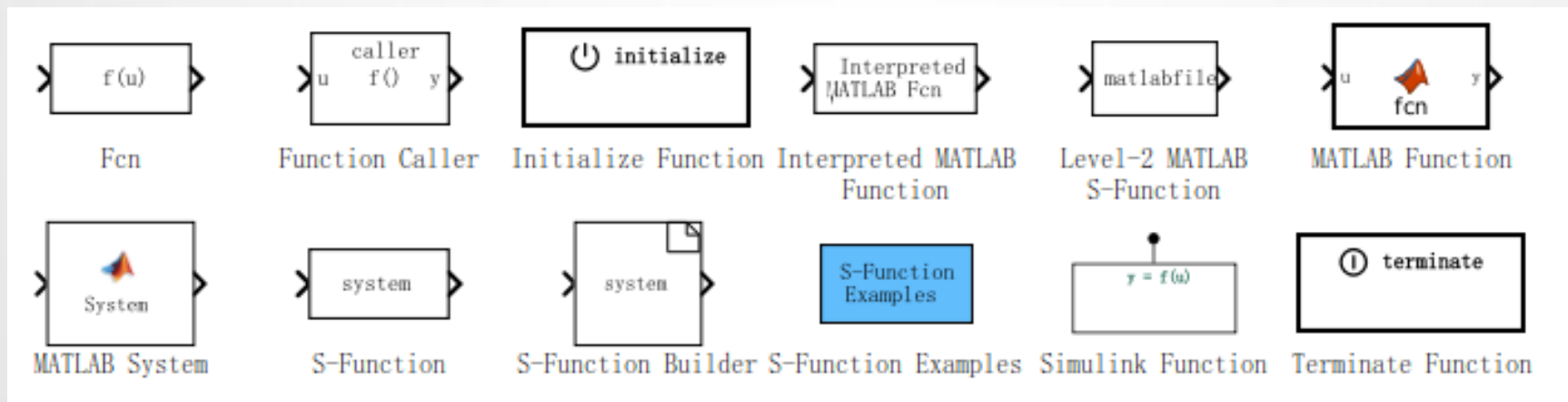
# M-函数与S-函数

- 为什么需要M-或S-函数
  - 底层模块搭建繁琐
  - MATLAB语言、C、Ada等
- M-函数与S-函数
  - M-模块：静态关系  $y = f(u)$
  - S-函数——系统函数
    - 动态关系——状态方程、任意复杂系统
- S-函数的编程结构与框架
- S-函数举例



# 自定义函数模块组

- User-defined Functions
  - 弥补底层模块搭建的不足
  - 两类模块——静态与动态



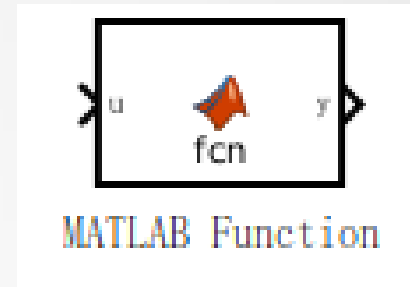
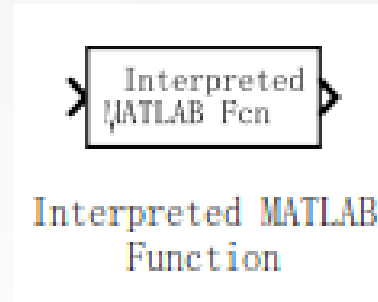


# M-函数

- 描述函数的静态关系
- M-函数模块
  - User-Defined Functions
- 饱和和非线性的M-描述编程

```
function y=satur_non(x)
if abs(x)>=3, y=2*sign(x); else, y=2/3*x; end
```

- M-函数的局限性
  - 不能带有附加参数：宽度、斜率





# 为什么需要S-函数

## ➤ 例6-23 韩京清研究员提出的微分-跟踪器

$$\begin{cases} x_1(k+1) = x_1(k) + T x_2(k) \\ x_2(k+1) = x_2(k) + T \text{fst}(x_1(k), x_2(k), u(k), r, h) \end{cases}$$

$$\delta = rh, \quad \delta_0 = \delta h, \quad b = x_1 - u + hx_2, \quad a_0 = \sqrt{\delta^2 + 8r|b|}$$

$$a = \begin{cases} x_2 + b/h, & |b| \leq \delta_0 \\ x_2 + 0.5(a_0 - \delta)\text{sign}(b), & |b| > \delta_0 \end{cases}$$

$$\text{fst} = \begin{cases} -ra/\delta, & |a| \leq \delta \\ -r\text{sign}(a), & |a| > \delta \end{cases}$$



# S-函数(状态方程)的基本结构

## ➤ S-函数入口语句

```
function [sys,x0,str,ts]=fun(t,x,u,flag,p1,p2,⋯)
```

## ➤ S-函数的基本框架

```
switch flag
case 0, [sys,x0,str,ts] = mdlInitializeSizes(T);
case 1, sys = mdlDeravitivess(x,u,r,h,T);
case 2, sys = mdlUpdates(x,u,r,h,T);
case 3, sys = mdlOutputs(x);
end
```



# S-函数的执行过程

- flag变量的自动取值
  - 初始化：flag=0, 输入、输出路数、状态数等
  - flag=3, 计算输出信号
  - flag=1, 2, 分别更新状态：连续、离散状态方程
- S-函数用模块表示

```
switch flag
case 0, [sys,x0,str,ts] = mdlInitializeSizes(T);
case 1, sys = mdlDeravitivess(x,u,r,h,T);
case 2, sys = mdlUpdates(x,u,r,h,T);
case 3, sys = mdlOutputs(x);
end
```



# S-函数的初始化

- 由simsizes读入模板，成员变量
  - .NumContStates，S-函数中连续状态的个数
  - .NumDiscStates，表示离散状态的个数
  - .NumInputs，模块的输入路数
  - .NumOutputs，表示模块的输出路数
  - .DirFeedthrough，输出方程是否显含  $u$
  - .NumSampleTimes，采样周期的个数
- 由simsizes返回
- 计算初值、采样周期等





## 其他函数结构

➤ 系统的状态  $\mathbf{x} = [\mathbf{x}_c, \mathbf{x}_d]^T$

➤ 连续状态更新  $\dot{\mathbf{x}}_c(t) = \mathbf{F}(t, \mathbf{x}, \mathbf{u})$

```
function sys = mdlDeravativivess(x,u,r,h,T);
```

➤ 离散状态更新  $\mathbf{x}_d(k+1) = \mathbf{G}(t, \mathbf{x}(k), \mathbf{u}(k))$

```
function sys = mdlUpdates(x,u,r,h,T);
```

➤ 输出计算  $\mathbf{y}(t) = \mathbf{H}(t, \mathbf{x}(t), \mathbf{u}(t))$

```
function sys = mdlOutputs(x);
```

