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

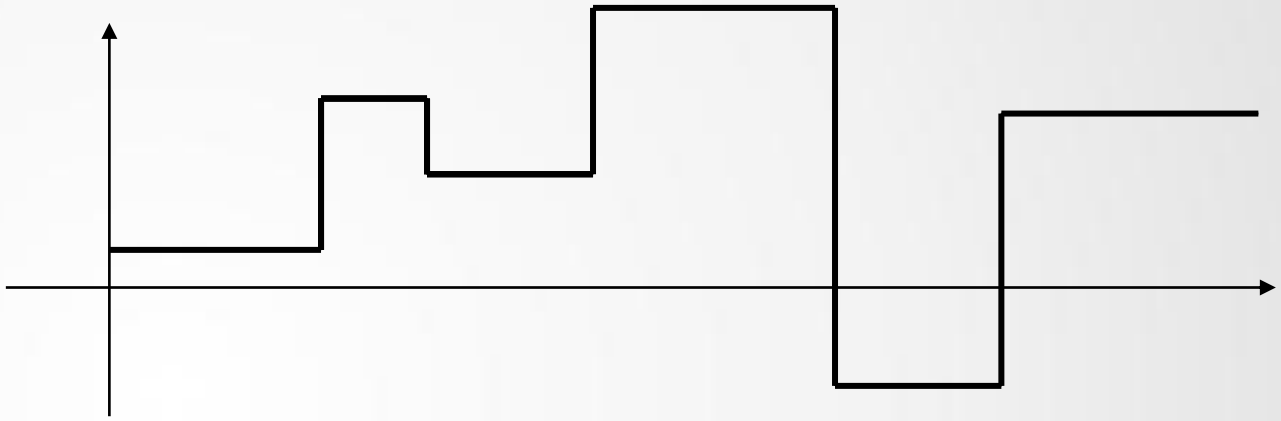
S-函数编程（下）

S-Function Programming (II)

主讲：薛定宇教授

例6-24 阶梯信号发生器

- 阶梯信号发生器
  - 静态函数
  - M函数不支持附加变量
  - 只能编写S-函数



- 转折点坐标

$$\texttt{tTime}=[t_1,t_2,\cdots,t_N],\ \texttt{yStep}=[y_1,y_2,\cdots,y_N]$$

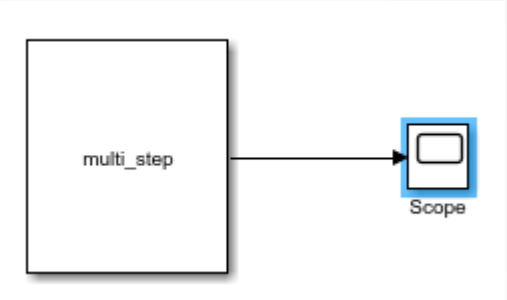
  - 输入、输出、状态个数、输入输出关系
  - 附加参数

S-函数编写清单

```
function [sys,x0,str,ts]=multi_step(t,x,u,flag,tTime,yStep)
switch flag,
   case 0, [sys,x0,str,ts] = mdlInitializeSizes;
   case 3, sys = mdlOutputs(t,tTime,yStep);
   case {1, 2, 4, 9}, sys = [];
   otherwise, error(['Unhandled flag = ',num2str(flag)]);
end;
function [sys,x0,str,ts] = mdlInitializeSizes
sizes = simsizes; sizes.NumContStates=0;
sizes.NumDiscStates=0; sizes.NumOutputs = 1;
sizes.NumInputs = 0; sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1; sys = simsizes(sizes);
x0 = []; str = []; ts = [0 0];
function sys = mdlOutputs(t,tTime,yStep)
i=find(tTime<=t); sys=yStep(i(end));
```

S-函数的使用方法

- 打开模型窗口
- c6mmstep.slx
- 数据点



x=[0,1,3,5,6,8]; y=[1,-3,4,6,2,6];

例6-25 微分跟踪器编程实现

- 韩京清研究员的微分-跟踪器

$$\begin{cases} x_1(k+1) = x_1(k) + Tx_2(k) \\ x_2(k+1) = x_2(k) + T\mathrm{fst}(x_1(k), x_2(k), u(k), r, h) \end{cases}$$

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

$$a = \begin{cases} x_2 + b/h, & |b| \leqslant \delta_0 \\ x_2 + 0.5(a_0 - \delta)\mathrm{sign}(b), & |b| > \delta_0 \end{cases} \qquad \mathrm{fst} = \begin{cases} -ra/\delta, & |a| \leqslant \delta \\ -r\mathrm{sign}(a), & |a| > \delta \end{cases}$$

- S函数准备
  - 模块的输出方程 $y = x$
  - 状态个数、输入输出路数、附加变量 (r, h, T)

主程序的设计

- 主函数

```
function [sys,x0,str,ts]=han_td(t,x,u,flag,r,h,T)
switch flag
case 0, [sys,x0,str,ts] = mdlInitializeSizes(T);
case 2, sys = mdlUpdates(x,u,r,h,T);
case 3, sys = mdlOutputs(x);
case {1, 4, 9}, sys = [];
otherwise, error(['Unhandled flag = ',num2str(flag)]);
end;
```

- 输出方程

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

初始化函数编写

```
function [sys,x0,str,ts] = mdlInitializeSizes(T)
sizes = simsizes;
sizes.NumContStates = 0;
sizes.NumDiscStates = 2;
sizes.NumOutputs = 2;
sizes.NumInputs = 1;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
x0 = [0; 0];
str = [];
ts = [T 0];
```

离散状态更新程序

- 没有连续状态，无需编程
- 离散状态更新

$$\begin{cases} x_1(k+1) = x_1(k) + Tx_2(k) \\ x_2(k+1) = x_2(k) + T\mathrm{fst}(x_1(k), x_2(k), u(k), r, h) \end{cases}$$

- 支持函数

```
function sys = mdlUpdates(x,u,r,h,T)
sys(1,1)=x(1)+T*x(2);
sys(2,1)=x(2)+T*fst2(x,u,r,h);
```

支持函数编写

- 数学公式

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

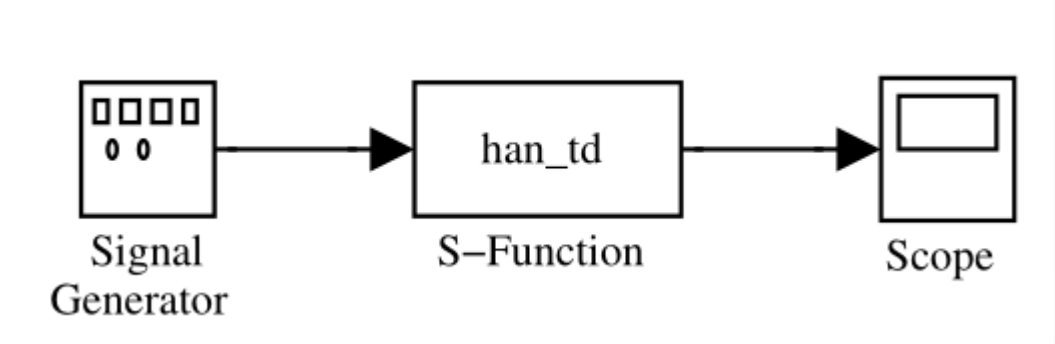
$$a = \begin{cases} x_2 + b/h, & |b| \leqslant \delta_0 \\ x_2 + 0.5(a_0 - \delta)\mathrm{sign}(b), & |b| > \delta_0 \end{cases} \qquad \mathrm{fst} = \begin{cases} -ra/\delta, & |a| \leqslant \delta \\ -r\mathrm{sign}(a), & |a| > \delta \end{cases}$$

- 程序实现

```
function f=fst2(x,u,r,h)
delta=r*h; delta0=delta*h; b=x(1)-u+h*x(2);
a0=sqrt(delta*delta+8*r*abs(b));
a=x(2)+b/h*(abs(b)<=delta0)+0.5*(a0-delta)*sign(b)*(abs(b)>delta0);
f=-r*a/delta*(abs(a)<=delta)-r*sign(a)*(abs(a)>delta);
```

模块的使用举例

- 用正弦信号去激励S-函数模块
- 得出跟踪信号与其导数信号
- 仿真模型 c6msf2



S-函数小结

- 为什么需要M-函数与S-函数
- M-函数的编程实现与局限性
- S-函数的结构与执行机制
  - Switch case标准框架 —— 适用于所有S-函数
  - 初始化 flag=0
  - 计算模块的输出, flag=3
  - 状态更新——离散与连续状态, flag=1,2

薛定宇教授主讲

东北大学

NORTHEASTERN UNIVERSITY 1923

控制系统仿真与CAD