中国地质大学 自动化学院
School of Automation, China University of Geosciences

线性系统分析与设计：实验PART II

基于LMI与SIMULINK的系统分析、设计与仿真

主要内容

- 基于LMI的系统稳定性分析
  - 基于LMI的系统稳定判据
  - 基于YALMIP工具箱的LMI条件验证
- 基于LMI的系统镇定设计
  - 基于LMI的系统镇定条件及控制器计算公式
  - 基于YALMIP工具箱的设计
- 基于SIMULINK的系统响应曲线绘制
  - 基于状态空间方程的系统SIMULINK搭建
  - 基于SIMULINK运行数据的曲线绘制

基于LMI的系统稳定性分析

➢ 基于LMI的系统稳定判据

对于如下线性定常系统

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}$$

系统平衡态 $x_e = 0$ 为渐近稳定的充要条件是：存在对称

矩阵 P 满足如下线性矩阵不等式条件：

$$P > 0, \qquad A^T P + PA < 0 \qquad\qquad V(x) = x^T Px$$

$\boxed{\text{如何求解上述线性矩阵不等式？}}$

基于LMI的系统稳定性分析

➢ 基于YALMIP工具箱的LMI条件求解

❑ YALMIP工具箱

❑ MATLAB加载工具箱

基于LMI的系统稳定性分析

实例分析

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Chuan-Ke Zhang
%%% 2018-01-19
%%% 验证系统的稳定性程序
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clc; clear

%%%  系统参数
A = [1 2 4; 1 1 1; 0 2 1];

%%% 描述待求的LMI
P = sdpvar(3,3,'symmetric'); % 给出待求矩阵
Fcond = [P>0,A'*P+P*A<0]; % 列出所有待求LMI

%%% 求解LMI
ops = sdpsettings('verbose',0,'solver','sedumi'); % 设置求解环境
diagnostics = solvesdp(Fcond,[],ops); % 迭代求解
[m p] = checkset(Fcond); % 返回求解结果
tmin = min(m); % 验证是否满足

if tmin > 0
    disp('System is stable') % 结论输出
else
    disp('System is unstable')  % 结论输出
end
```

$$\dot{x}(t)=\begin{bmatrix}1 & 2 & 4\\ 1 & 1 & 1\\ 0 & 2 & 1\end{bmatrix}x(t)$$

```
P = sdpvar(n,n,'symmetric');
% P为n维*n维对称矩阵
Q = sdpvar(n,m,'full');
% Q为n维*m维矩阵
```

$$P>0, \qquad A^T P+PA<0$$

基于LMI的系统稳定性分析

- 练习题1：利用LMI判据分析系统稳定性

$$\dot{x}(t)=\begin{bmatrix}1&3&5&9\\6&2&1&1\\3&7&4&3\\4&3&7&7\end{bmatrix}x(t)$$

- 练习题2：求解如下LMI

$$A=\begin{bmatrix}5&3\\1&2\end{bmatrix},B=\begin{bmatrix}2\\1\end{bmatrix}$$

$$\begin{bmatrix}A^TP+PA+Q&PB\\B^TP&-R\end{bmatrix}<0,\ P>0,\ Q>0,\ R>0$$

基于LMI的系统镇定设计

➢ 基于LMI的系统镇定条件及控制器计算公式

对于如下线性定常系统及状态反馈控制器

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ u(t) = Kx(t) \end{cases} \Rightarrow \dot{x}(t) = (A + BK)x(t)$$

系统可由状态反馈控制器镇定的条件是：存在对称矩阵

L，矩阵V，满足如下LMI：

$$\begin{cases} L > 0 \\ AL + LA^T + BV + V^T B^T < 0 \end{cases} \Leftarrow \begin{cases} P > 0 \\ PA + A^T P < 0 \end{cases}$$

$$K = VL^{-1}$$

基于LMI的系统镇定设计

❑ 实例分析

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Chuan-Ke Zhang
%%% 2018-01-19
%%% 设计系统镇定控制器程序
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clc; clear

%%%  系统参数
A = [1 2 4; 1 1 1; 0 2 1];
B = [1; 2; 1];

%%% 描述待求的LMI
L = sdpvar(3,3,'symmetric'); % 给出待求矩阵
V = sdpvar(1,3,'full'); % 给出待求矩阵
Fcond = [L>0,A*L+L*A'+B*V+V'*B'<0]; % 列出所有待求LMI

%%% 求解LMI
ops = sdpsettings('verbose',0,'solver','sedumi'); % 设置求解环境
diagnostics = solvesdp(Fcond,[],ops); % 迭代求解
[m p] = checkset(Fcond); % 返回求解结果
tmin = min(m); % 验证是否满足

if tmin > 0
    Vh = double(V);
    Lh = double(L);
    disp('System can be stabilized, and the control gain is given as') % 结论输出
    K = Vh*inv(Lh)
else
    disp('System cannot be stabilized using state-feedback controller')  % 结论输出
end
```

$$\begin{cases} \dot{x}(t) = \begin{bmatrix} 1 & 2 & 4 \\ 1 & 1 & 1 \\ 0 & 2 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} u(t) \\ u(t) = Kx(t) \end{cases}$$

$$\begin{cases} L > 0 \\ AL + LA^T + BV + V^T B^T < 0 \end{cases}$$

$$K = VL^{-1}$$

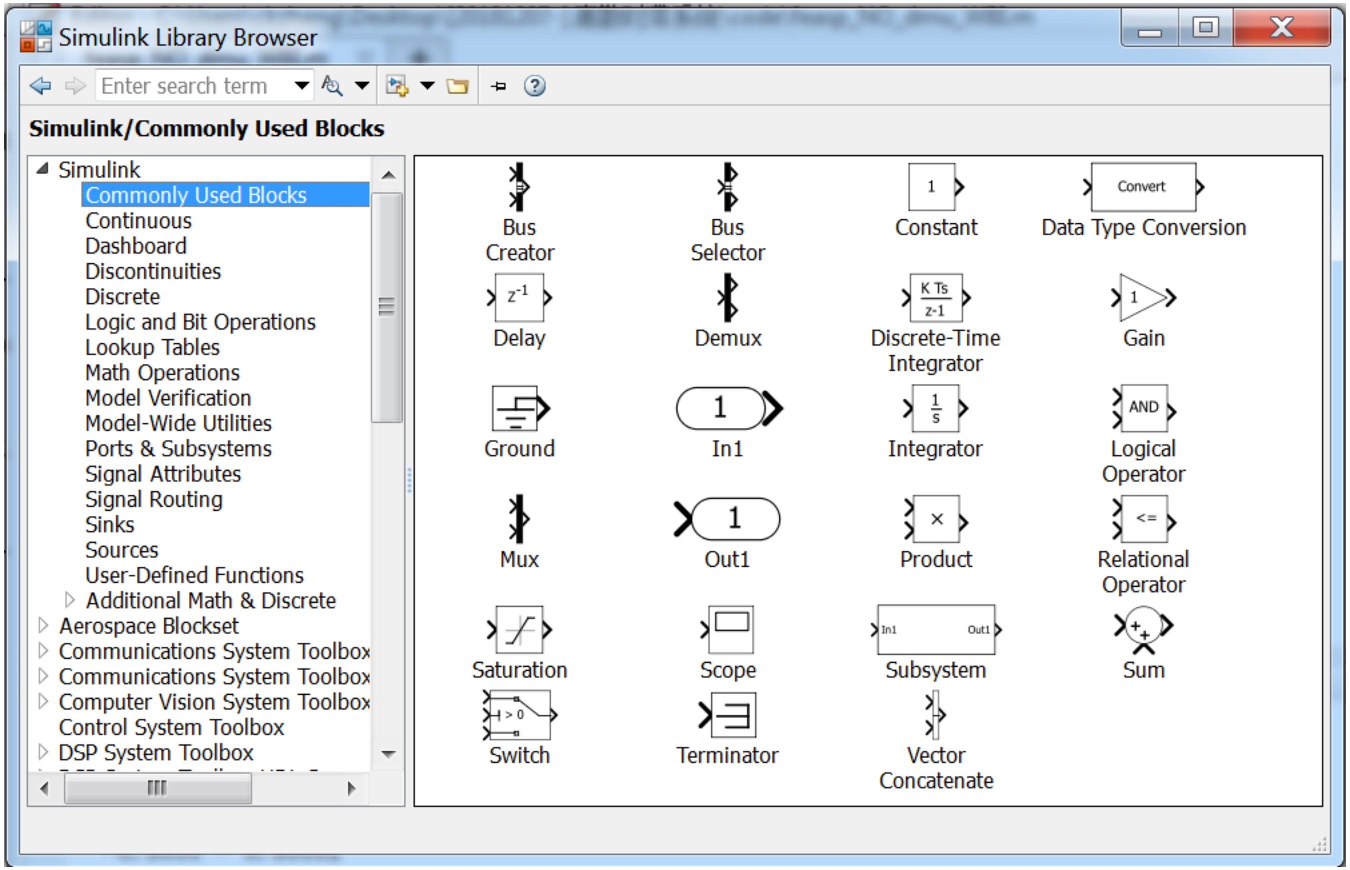
基于LMI的系统镇定设计

- 练习题3：设计状态反馈控制器镇定如下系统

$$\begin{cases} \dot{x}(t) = \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 1 & 3 \\ 1 & 2 \end{bmatrix} u(t) \\ u(t) = Kx(t) \end{cases}$$

基于SIMULINK的系统响应曲线绘制

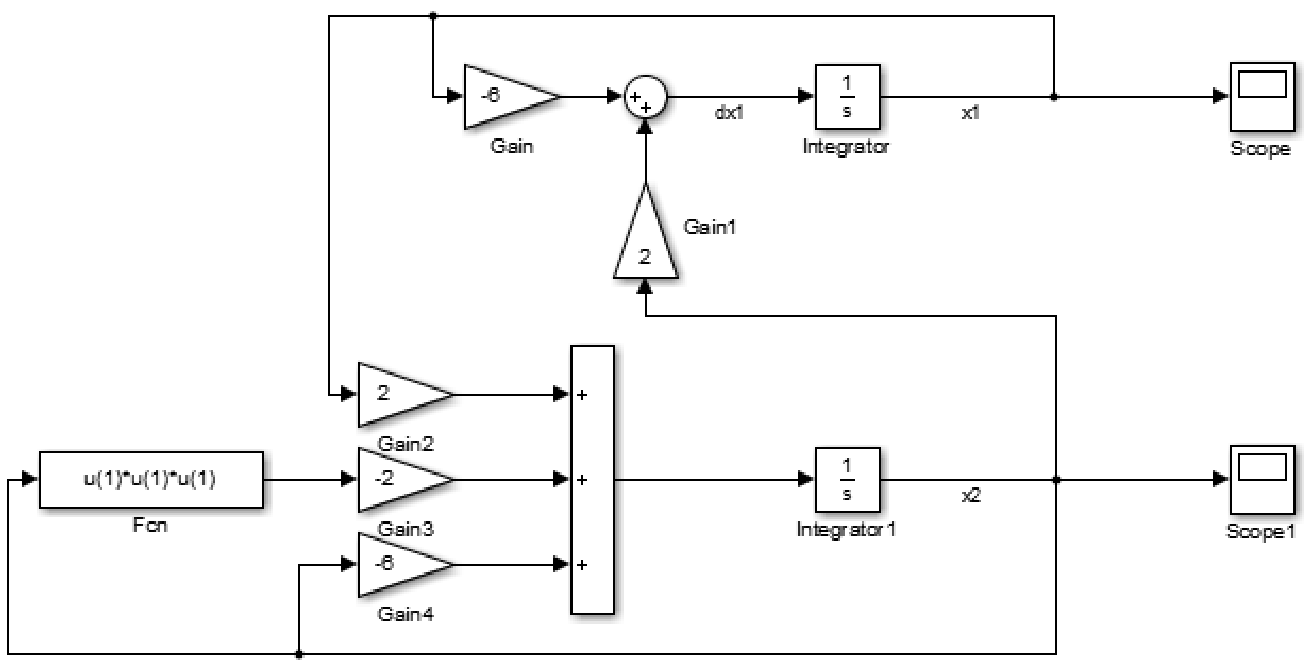
- SIMULINK LAB



基于SIMULINK的系统响应曲线绘制

- 基于状态空间方程的系统SIMULINK搭建(1)

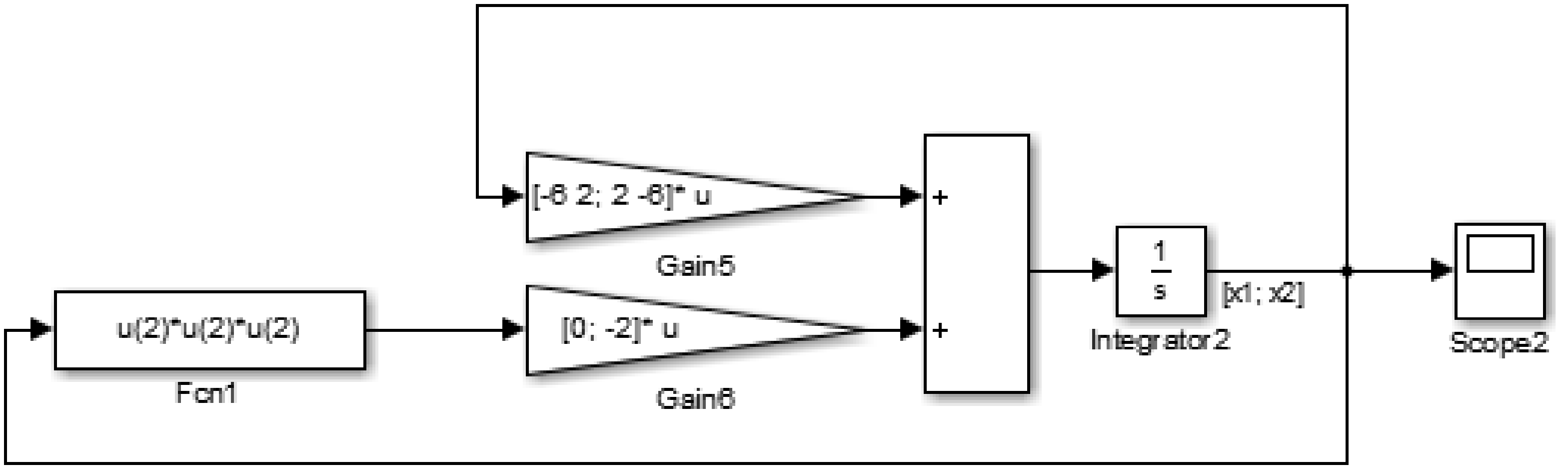
$$\begin{cases} \dot{x}_1 = -6x_1 + 2x_2 \\ \dot{x}_2 = 2x_1 - 6x_2 - 2x_2^3 \end{cases}$$



基于SIMULINK的系统响应曲线绘制

- 基于状态空间方程的系统SIMULINK搭建(2)

$$\begin{cases}\dot{x}_1 = -6x_1 + 2x_2 \\ \dot{x}_2 = 2x_1 - 6x_2 - 2x_2^3\end{cases} \Rightarrow \begin{bmatrix}\dot{x}_1 \\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}-6 & 2 \\ 2 & -6\end{bmatrix}\begin{bmatrix}x_1 \\ x_2\end{bmatrix} + \begin{bmatrix}0 \\ -2\end{bmatrix}x_2^3$$



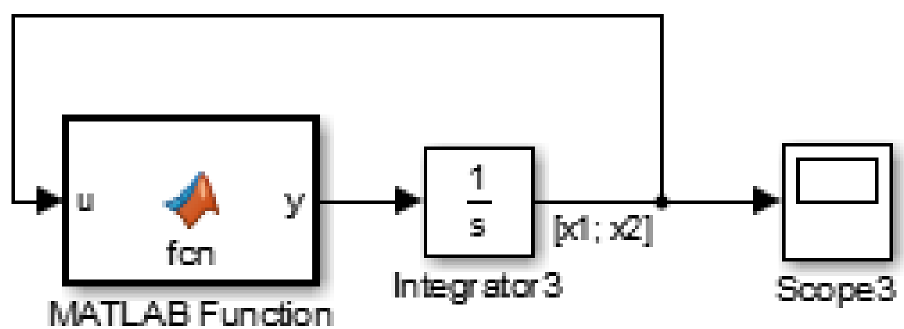
基于SIMULINK的系统响应曲线绘制

➢ 基于状态空间方程的系统SIMULINK搭建(3)

$$\begin{cases} \dot{x}_1 = -6x_1 + 2x_2 \\ \dot{x}_2 = 2x_1 - 6x_2 - 2x_2^3 \end{cases} \qquad \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -6 & 2 \\ 2 & -6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ -2 \end{bmatrix} x_2^3$$

MATLAB Function

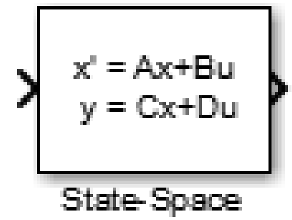
```
function y = fcn(u)
%#codegen

% x1 = u(1);
% x2 = u(2);
% dx1 = -6*x1 + 2*x2;
% dx2 = 2*x1 - 6*x2 - 2*x2^3;
% y = [dx1; dx2];

y = [-6 2; 2 -6]*u + [0; -2]*u(2)^3;
```

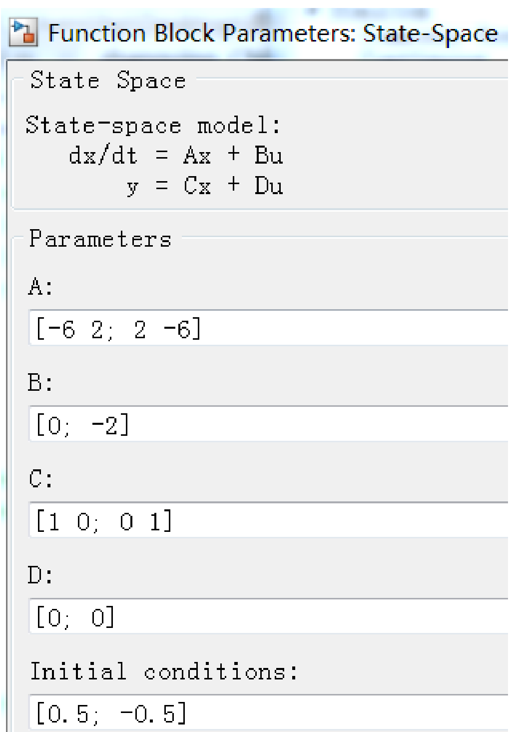


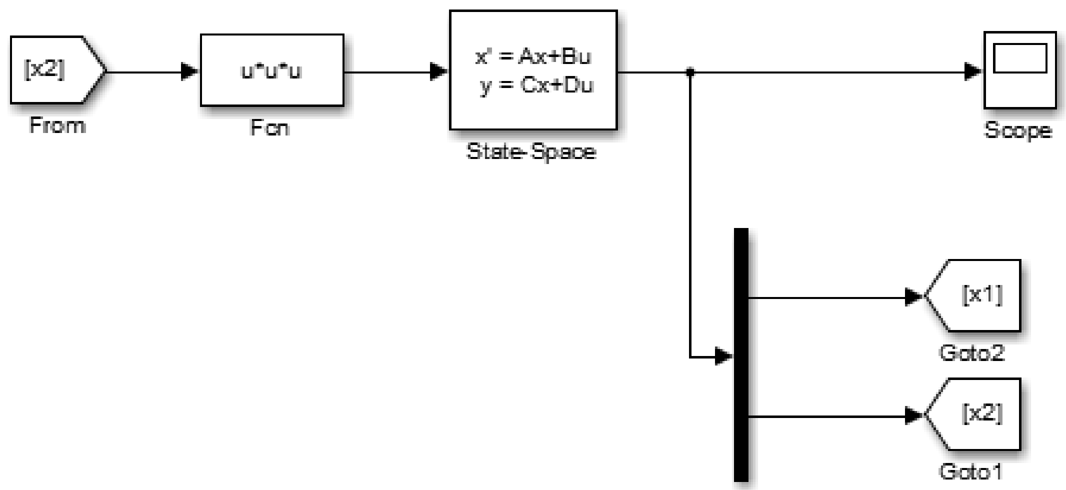
基于SIMULINK的系统响应曲线绘制

➢ 基于状态空间方程的系统SIMULINK搭建(3)

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -6 & 2 \\ 2 & -6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ -2 \end{bmatrix} x_2^3$$





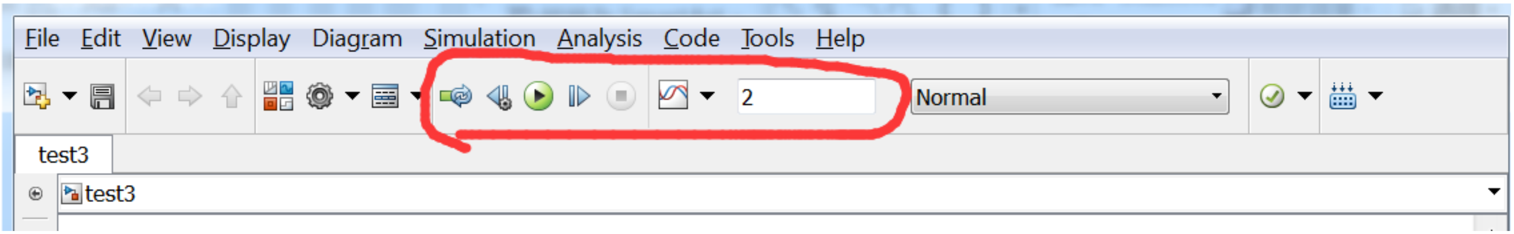


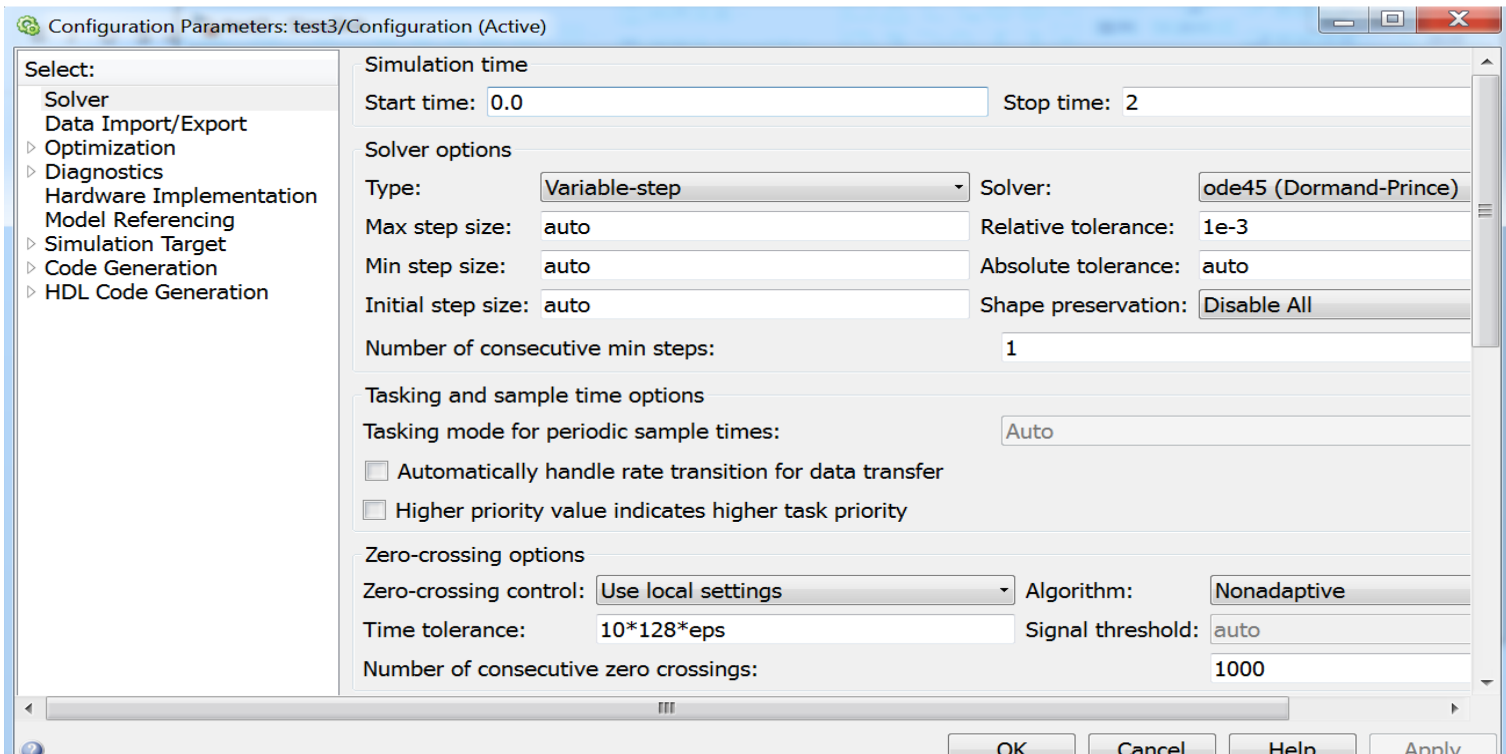
基于SIMULINK的系统响应曲线绘制

- 基于状态空间方程的系统SIMULINK搭建
  - S-函数
  - 创建子系统
  - 封装子系统
  - 模型库
  - ……

基于SIMULINK的系统响应曲线绘制

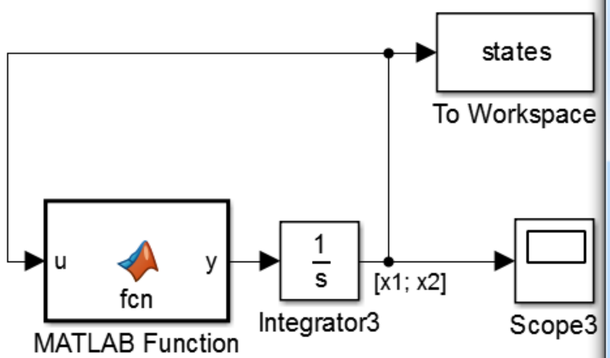
- 基于SIMULINK运行数据的曲线绘制：运行simulink模块

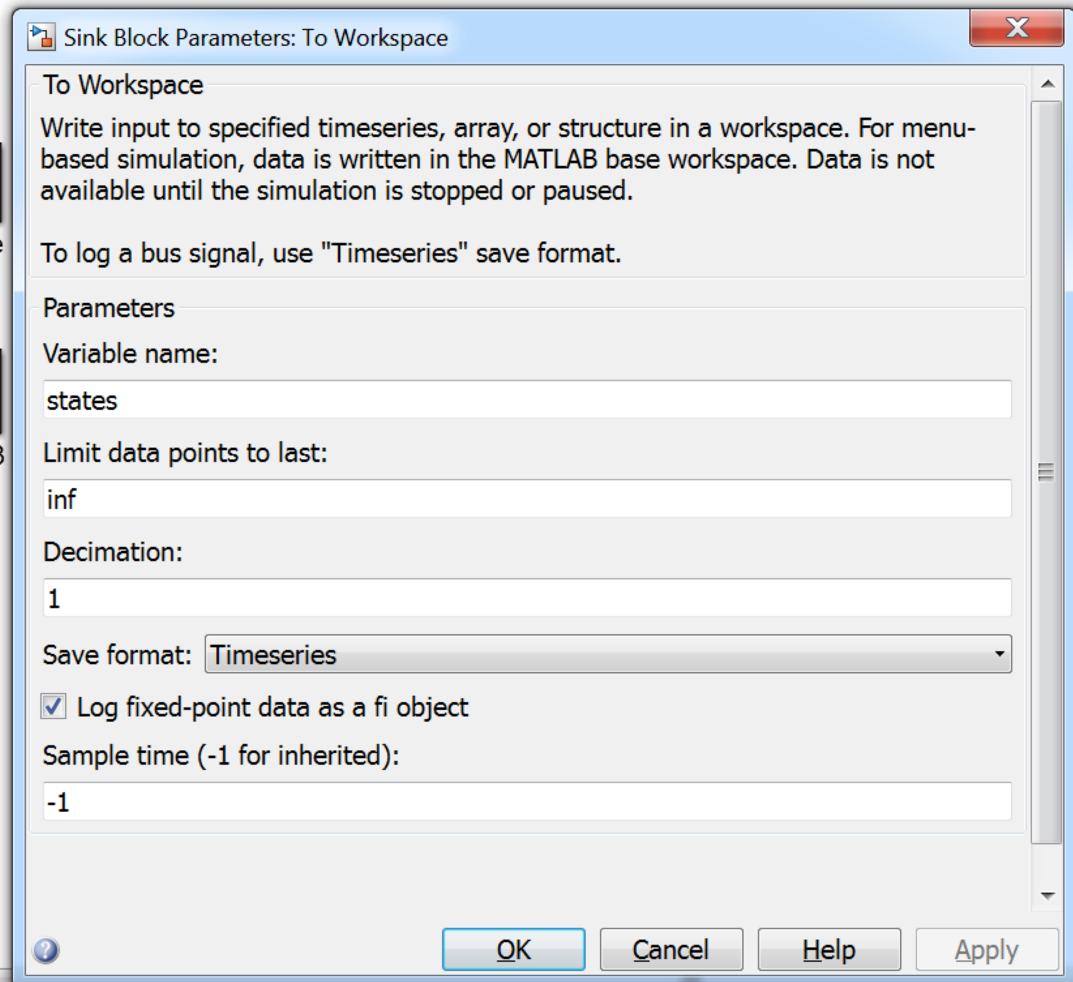




基于SIMULINK的系统响应曲线绘制

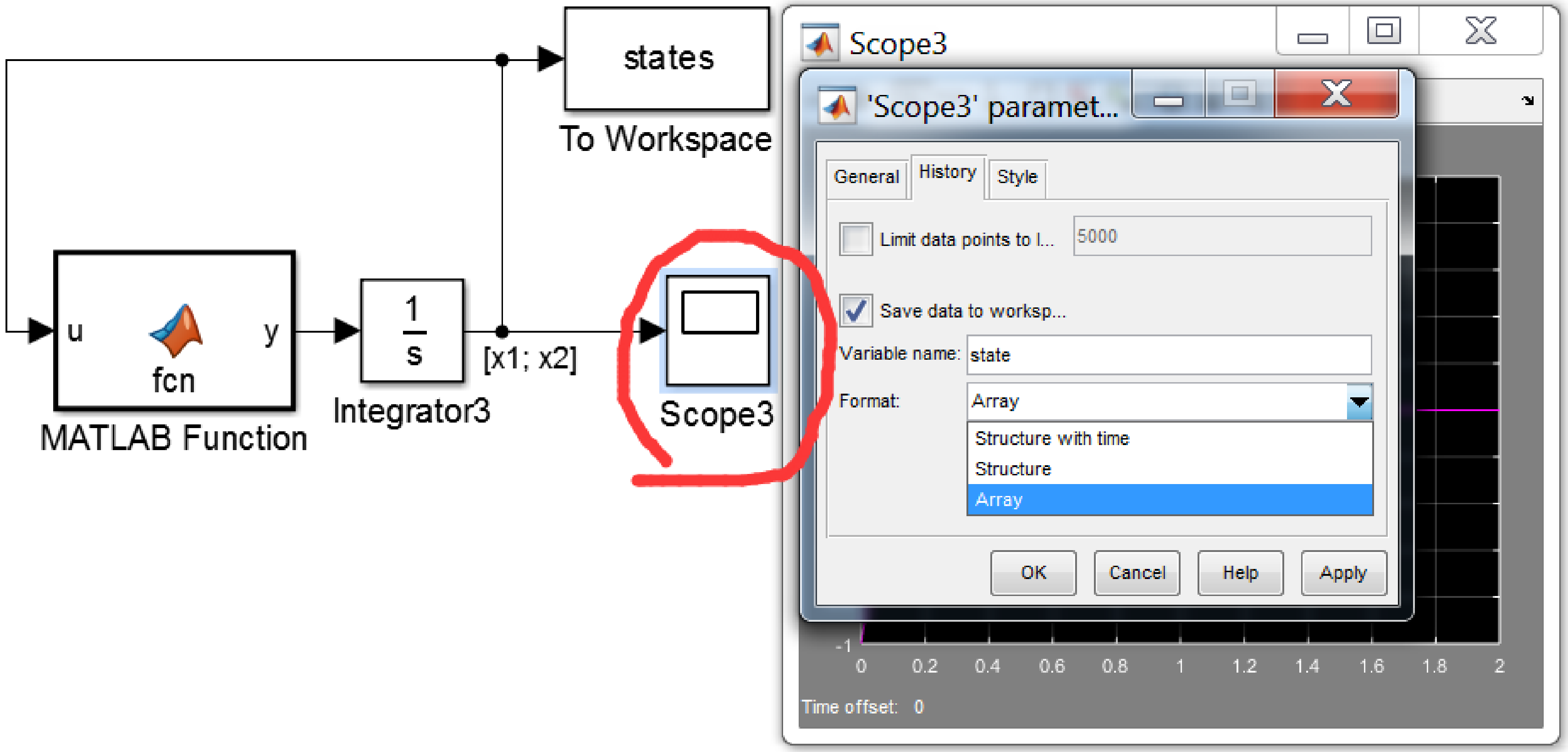
- 基于SIMULINK运行数据的曲线绘制：导入WORKSPACE（1）





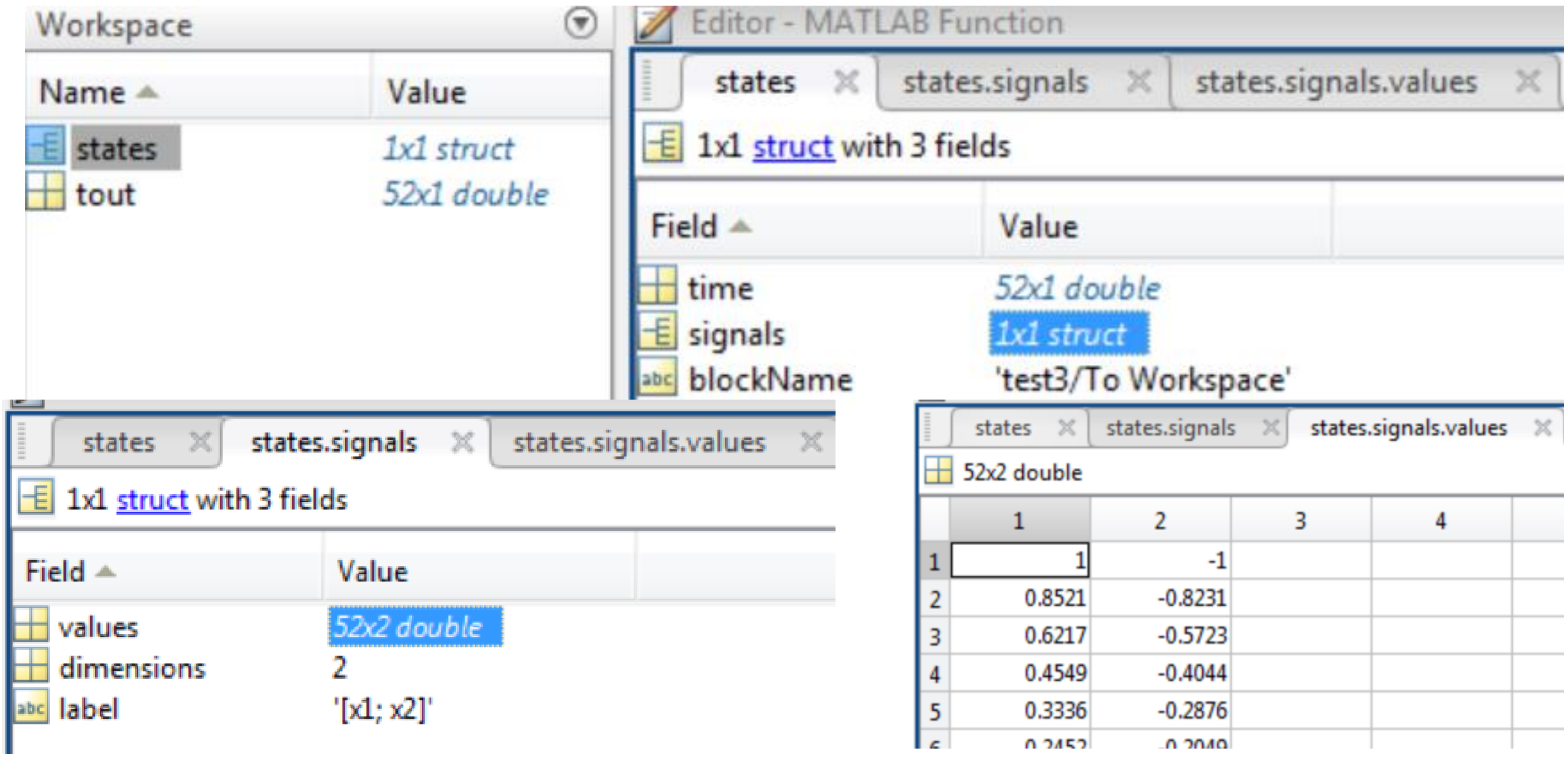
基于SIMULINK的系统响应曲线绘制

➢ 基于SIMULINK运行数据的曲线绘制：导入WORKSPACE（2）



基于SIMULINK的系统响应曲线绘制

- 基于SIMULINK运行数据的曲线绘制：WORKSPACE中数据

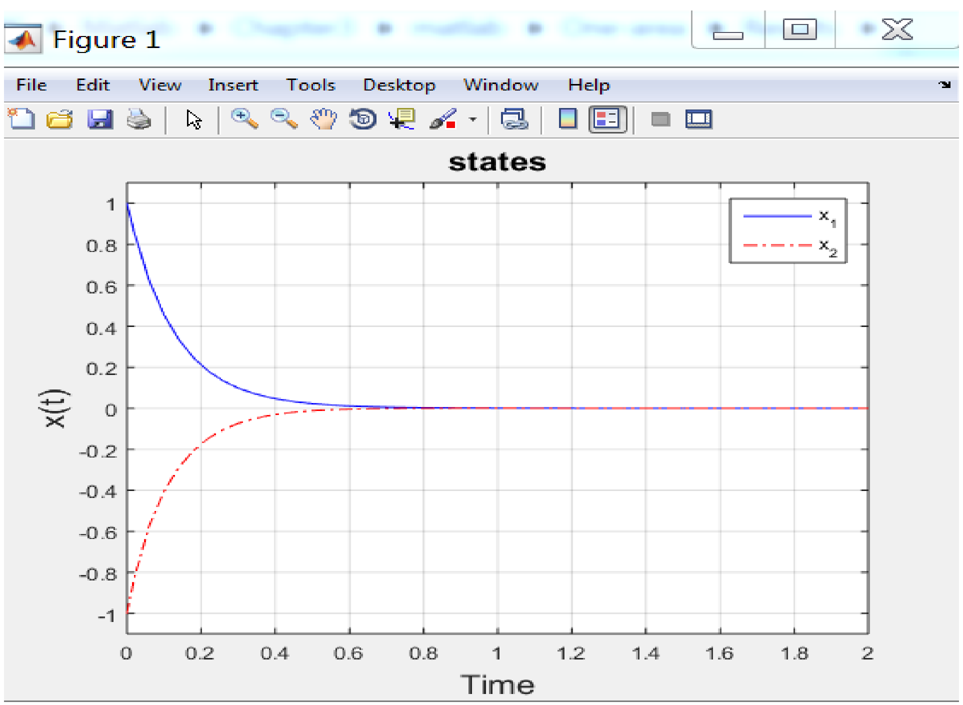



```
>> plot(states.time,states.signals.values(:,1))
```

基于SIMULINK的系统响应曲线绘制

➤ 基于SIMULINK运行数据的曲线绘制：绘图（1）

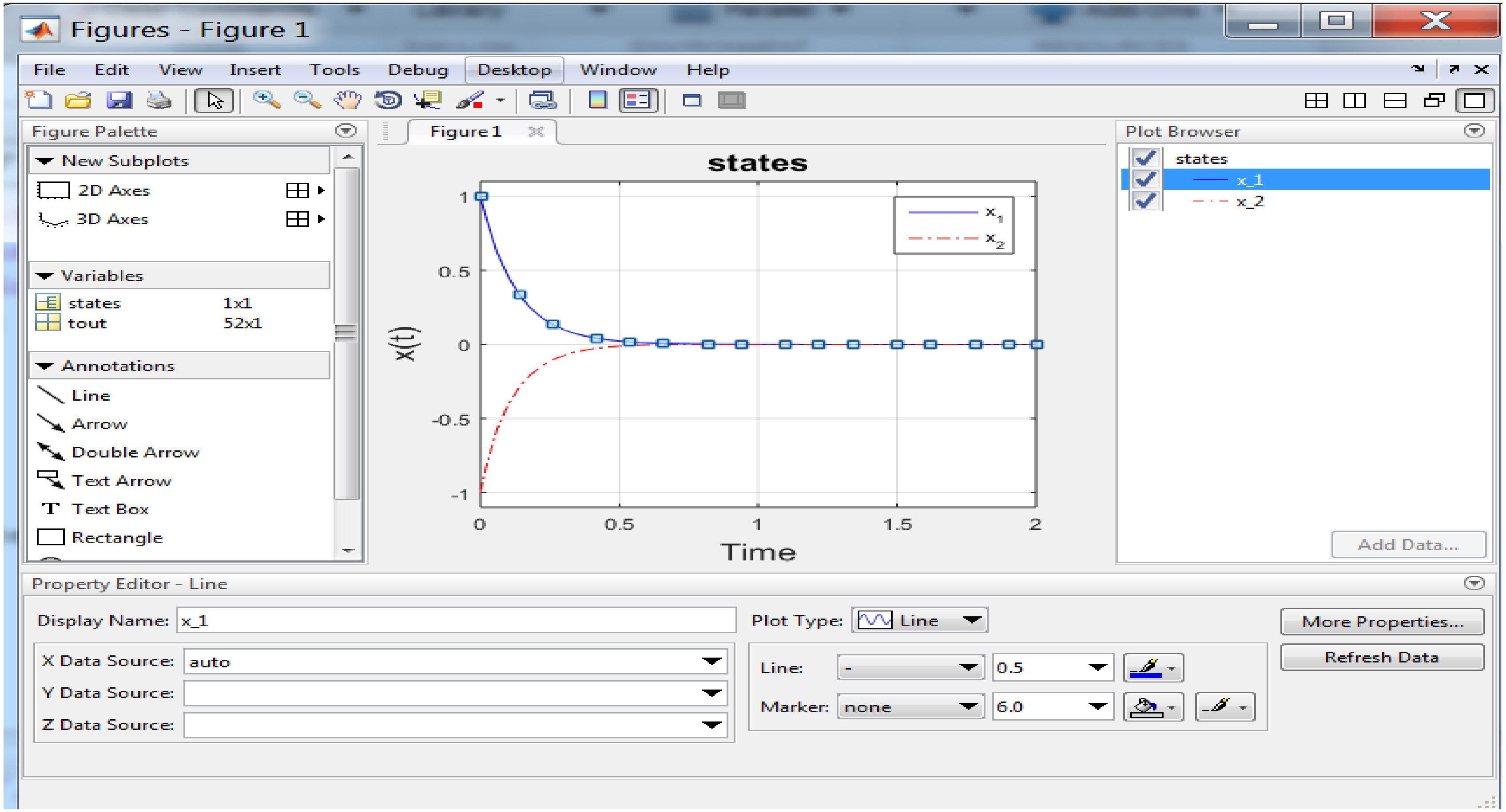
```
figure(1)
set(gca,'FontSize',15, 'FontName', 'Times New Roman')
plot(states.time,states.signals.values(:,1),'b-',...
     states.time,states.signals.values(:,2),'r-.')
xlabel('Time','FontSize',15);
ylabel('x(t)','FontSize',15);
title('states','FontSize',15)
axis([0 2 -1.1 1.1]);
grid on
legend('x_1','x_2')
```



基于SIMULINK的系统响应曲线绘制

➤ 基于SIMULINK运行数据的曲线绘制：绘图（2）

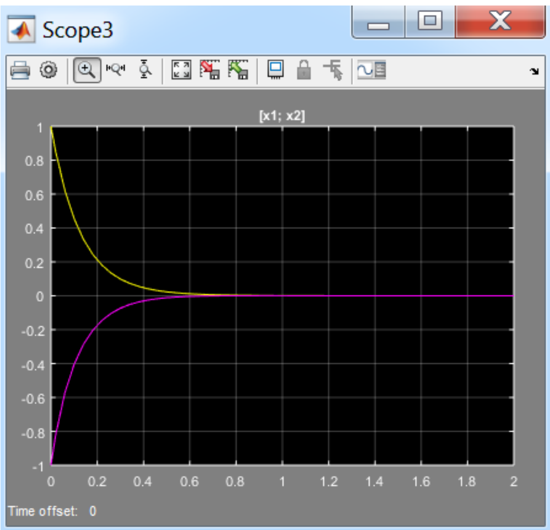
```
plot(states.time,states.signals.values(:,1),states.time,states.signals.values(:,2))
```

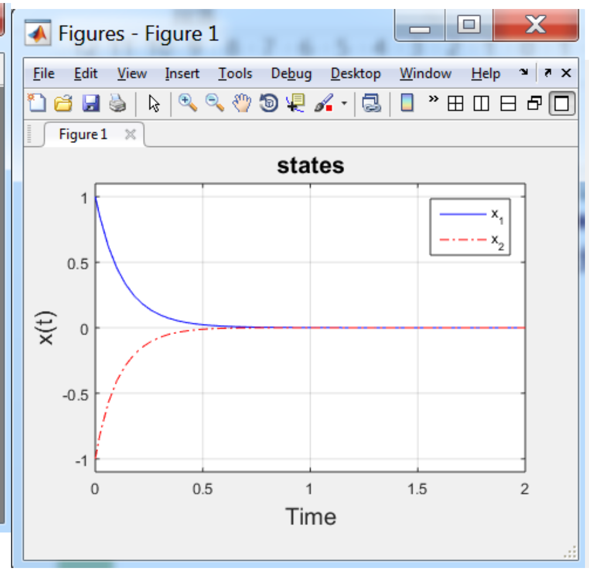


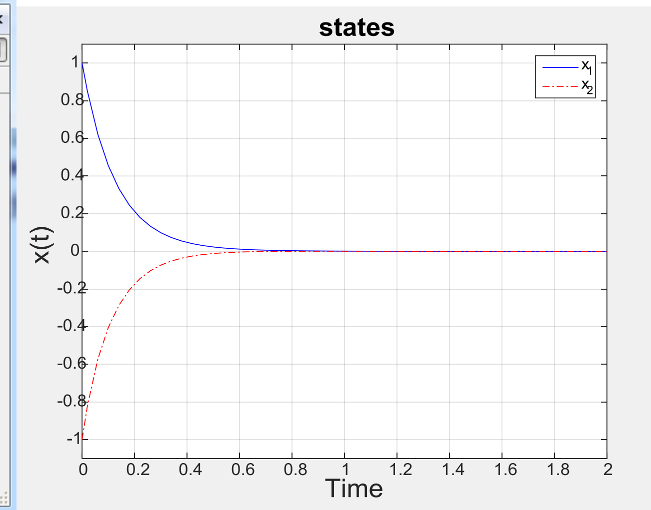
中国地质大学 自动化学院
School of Automation, China University of Geosciences

基于SIMULINK的系统响应曲线绘制

- 基于SIMULINK运行数据的曲线绘制：取出曲线结果







基于SIMULINK的系统响应曲线绘制

- 练习题4：利用SIMULINK仿真如下系统，并绘出其状态响应曲线、相图；初始条件选[-0.2; 0.3; 0.7]

the following representation of Chua's circuit systems:

$$\begin{cases} \dot{x}_1(t) = a\left[x_2(t) - h\left(x_1(t)\right)\right] \\ \dot{x}_2(t) = x_1(t) - x_2(t) + x_3(t) \\ \dot{x}_3(t) = -bx_2(t) \\ p(t) = x_1(t) \end{cases} \tag{23}$$

with the nonlinear characteristics of Chua's diode

$$h(x) = m_1 x_1(t) + \frac{1}{2}(m_0 - m_1)\left(|x_1(t) + c| - |x_1(t) - c|\right) \tag{24}$$

and parameters $a = 9$, $b = 14.28$, $c = 1$, $m_0 = -(1/7)$, $m_1 = 2/7$, and $c = 1$. In this case, Chua's system will obtain the

SIMULINK的案例库