

```
function [sys,x0,str,ts]=sfun(t,x,u,flag)
switch flag
    case 0                %初始化
        [sys,x0,str,ts]=mdlInitializeSizes;
    case 1                %计算连续状态
        sys=mdlDerivatives(t,x,u);
    case 2                %计算离散状态
        sys=mdlUpdate(t,x,u);
    case 3                %计算输出
        sys=mdlOutputs(t,x,u);
    case 4                %计算采样时间
        sys=mdlGetTimeOfNextVarHit(t,x,u);
    case 9                %结束时的动作
        sys=mdlTerminate(t,x,u);
    otherwise            %flag错误
        error(['不正确的flag=',num2str(flag)]);
end
```

```
function [sys,x0,str,ts]=mdlInitializeSizes()
    sizes=simsizes;        %创建尺寸结构
    sizes.NumContStates = 2; %连续状态的个数为0
    sizes.NumDiscStates = 0; %离散状态的个数为0
    sizes.NumOutputs = 1; %输出个数为1
    sizes.NumInputs = 1; %输入个数为2
    sizes.DirFeedthrough= 0; %直接反馈输入
    sizes.NumSampleTimes= 1; %至少需一个采样时间
    sys = simsizes(sizes);
    x0 = [0 0]; %无状态,故初始状态为空
    str = []; %系统保留
    ts = [0 0]; %初始化采样时间数组
```

```
function sys=mdlDerivatives(t,x,u)
%计算连续状态
```

```
    y1=0*x(1)+1*x(2)+0*u;
    y2=-0.4*x(1)-0.2*x(2)+0.2*u;
    sys =[y1;y2];
```

```
function sys=mdlUpdate(t,x,u)
%无离散状态
```

```
sys =[];
```

```
function sys=mdlOutputs(t,x,u)  
    sys = x(1);
```

```
function sys=mdlGetTimeOfNextVarHit(t,x,u)  
    sampleTime = 1;
```

```
function sys=mdlTerminate(t,x,u)  
    sys = [];
```