

在命令行输入sfundemos,总共有5个例子

## vsfunc.m

采样时间采用的是变步长

```
function [sys,x0,str,ts,simStateCompliance] = vsfunc(t,x,u,flag)
%VSFUNC Variable step S-function example.
% This example S-function illustrates how to create a variable step
% block in Simulink. This block implements a variable step delay
% in which the first input is delayed by an amount of time determined
% by the second input:
%
%      dt      = u(2)
%      y(t+dt) = u(t)
%
% See also SFUNTMPL, CSFUNC, DSFUNC.

% Copyright 1990-2009 The MathWorks, Inc.
% $Revision: 1.1.6.1 $

% The following outlines the general structure of an S-function.
%
switch flag,
    %%%%%%%%%%%%%%%
    % Initialization %
    %%%%%%%%%%%%%%%
    case 0,
        [sys,x0,str,ts,simStateCompliance]=mdlInitializeSizes;

    %%%%%%%%%%%%%%%
    % Update %
    %%%%%%%%%%%%%%%
    case 2,
        sys=mdlUpdate(t,x,u);

    %%%%%%%%%%%%%%%
    % Outputs %
    %%%%%%%%%%%%%%%
    case 3,
        sys=mdlOutputs(t,x,u);

    %%%%%%%%%%%%%%%
    % GetTimeOfNextVarHit %
    %%%%%%%%%%%%%%%
    case 4,
        sys=mdlGetTimeOfNextVarHit(t,x,u);

    %%%%%%%%%%%%%%%
    % Terminate %
    %%%%%%%%%%%%%%%
    case 9,
        sys=mdlTerminate(t,x,u);
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Unhandled flags %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
case 1,
    sys = [];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Unexpected flags %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
otherwise
    DAStudio.error('Simulink:blocks:unhandledFlag', num2str(flag));

end

% end sfuntmpl

%
%=====
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-function.
%=====
%
function [sys,x0,str,ts,simStateCompliance]=mdlInitializeSizes

%
% call simsizes for a sizes structure, fill it in and convert it to a
% sizes array
%
sizes = simsizes;

sizes.NumContStates = 0;
sizes.NumDiscStates = 1;
sizes.NumOutputs = 1;
sizes.NumInputs = 2;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;    % at least one sample time is needed

sys = simsizes(sizes);

%
% initialize the initial conditions
%
x0 = [0];

%
% str is always an empty matrix
%
str = [];

%
% initialize the array of sample times
%
ts = [-2 0];    % variable sample time

% specify that the simState for this s-function is same as the default
simStateCompliance = 'DefaultSimState';

% end mdlInitializeSizes

%

```

```

%=====
% mdlUpdate
% Handle discrete state updates, sample time hits, and major time step
% requirements.
%=====
%
function sys=mdlUpdate(t,x,u)

sys = u(1);

% end mdlUpdate

%
%=====
% mdlOutputs
% Return the block outputs.
%=====
%
function sys=mdlOutputs(t,x,u)

sys = x(1);

% end mdlOutputs

%
%=====
% mdlGetTimeOfNextVarHit
% Return the time of the next hit for this block. Note that the result is
% absolute time.
%=====
%
function sys=mdlGetTimeOfNextVarHit(t,x,u)

sys = t + u(2);

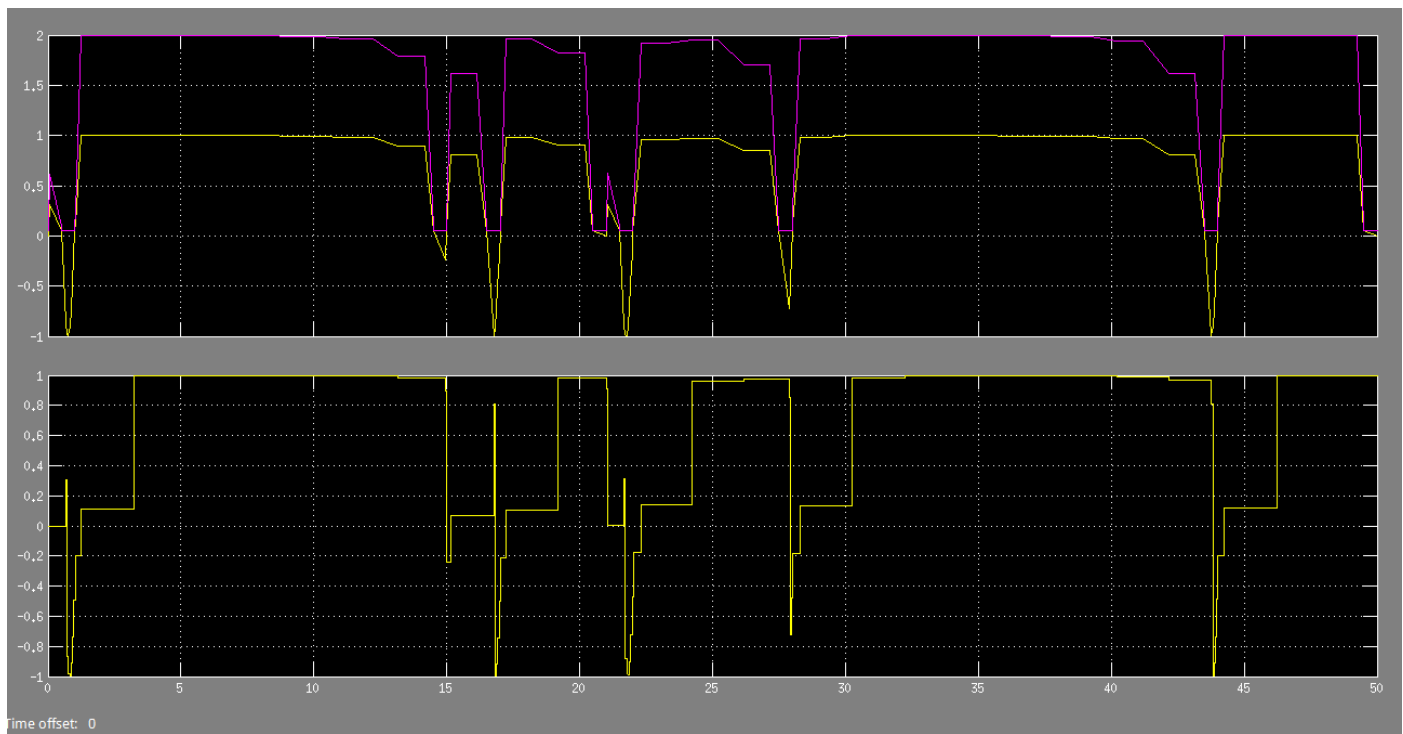
% end mdlGetTimeOfNextVarHit

%
%=====
% mdlTerminate
% Perform any end of simulation tasks.
%=====
%
function sys=mdlTerminate(t,x,u)

sys = [];

% end mdlTerminate

```



## sfundsc1.m

继承采样时间

```
function [sys,x0,str,ts,simStateCompliance] = sfundsc1(t,x,u,flag)
%SFUNDSC1 Example memory MATLAB file S-function with inherited sample time
% This MATLAB file S-function is an example of how to implement an
% inherited sample time S-function which has state. The actual sample
% time will be determined by what is driving this S-function. It may
% be continuous or discrete. This S-function uses one discrete state
% element as storage such that the previous input is provided at the
% output.
%
% See sfuntmpl.m for a general S-function template.
%
% See also SFUNTMPL.
%
% Copyright 1990-2009 The MathWorks, Inc.
% $Revision: 1.1.6.2 $
```

switch flag,

```
%%%%%%%%%%%%%%
% Initialization %
%%%%%%%%%%%%%%
```

```
case 0,
    [sys,x0,str,ts,simStateCompliance]=mdlInitializeSizes;
```

```
%%%%%%%%%%%%%%
% Update %
%%%%%%%%%%%%%%
```

```
case 2,
    sys = mdlUpdate(t,x,u);
```

```

%%%%%%%%%%
% Output %
%%%%%%%%%%
case 3,
    sys = mdlOutputs(t,x,u);

%%%%%%%%%%
% Terminate %
%%%%%%%%%%
case 9,
    sys = [];

otherwise
    DASTudio.error('Simulink:blocks:unhandledFlag', num2str(flag));
end

%end sfundsc1

%
%=====
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-function.
%=====
%
function [sys,x0,str,ts,simStateCompliance]=mdlInitializeSizes

sizes = simsizes;

sizes.NumContStates = 0;
sizes.NumDiscStates = 1;
sizes.NumOutputs = 1;
sizes.NumInputs = 1;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1;

sys = simsizes(sizes);

x0 = 0;
str = [];
ts = [-1 0]; % Inherited sample time

% specify that the simState for this s-function is same as the default
simStateCompliance = 'DefaultSimState';

% end mdlInitializeSizes

%
%=====
% mdlUpdate
% Handle discrete state updates, sample time hits, and major time step
% requirements.
%=====
%
function sys = mdlUpdate(t,x,u)
sys = u;

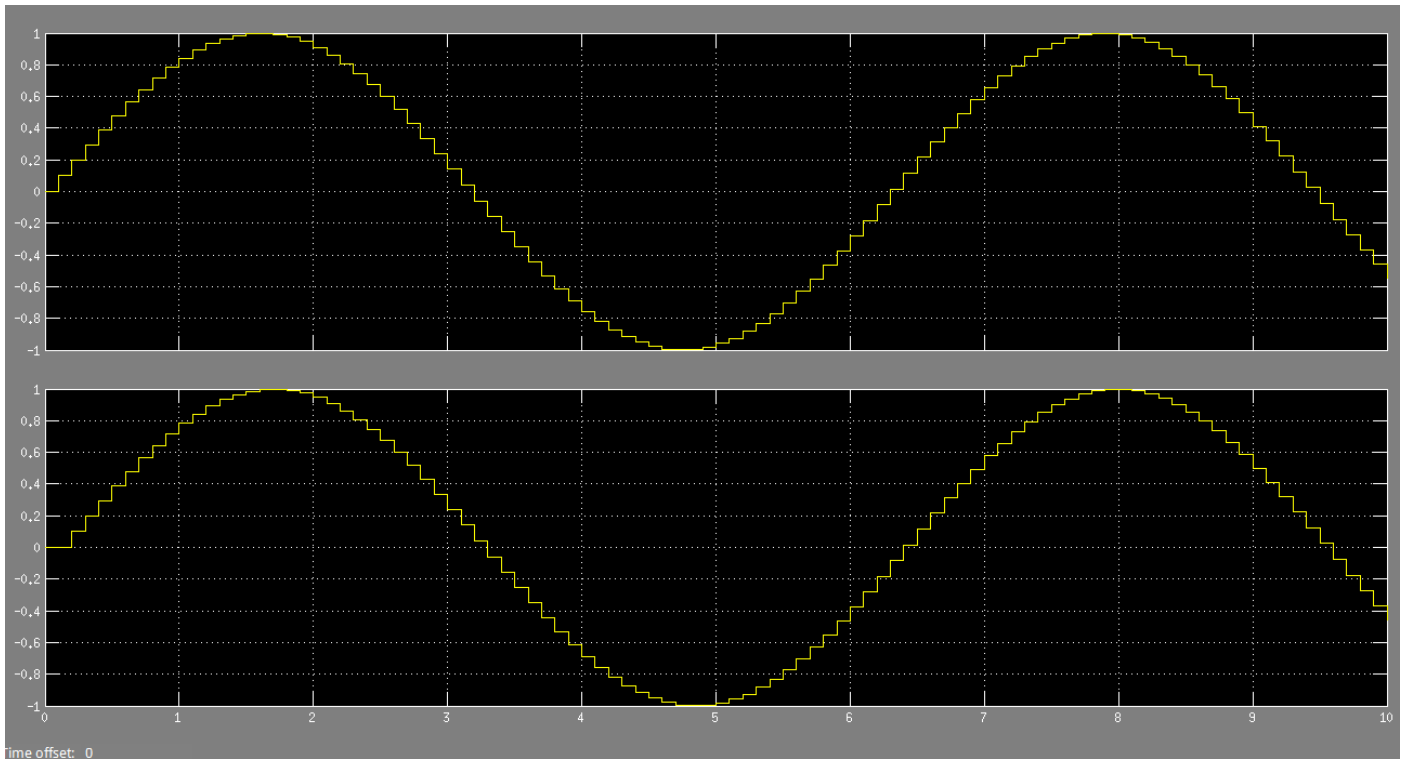
%end mdlUpdate

%
%=====

```

```
% mdlOutputs
% Return the output vector for the S-function
%=====
%
function sys = mdlOutputs(t,x,u)
sys = x;

%end mdlOutputs
```



mdlUpdate更新一个离散信号并把值传给它的状态x  
mdlOutputs输出一个信号的状态即x

## limint.m

```
function [sys,x0,str,ts,simStateCompliance]=limintm(t,x,u,flag,lb,ub,xi)

switch flag,

    %%%%%%%%%%%%%%%
    % Initialization %
    %%%%%%%%%%%%%%%
    case 0
        [sys,x0,str,ts,simStateCompliance] = mdlInitializeSizes(lb,ub,xi);

    %%%%%%%%%%%%%%%
    % Derivatives %
    %%%%%%%%%%%%%%%
    case 1
        sys = mdlDerivatives(t,x,u,lb,ub);

    %%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%
```

```

% Update and Terminate %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
case [2,9]
    sys = []; % do nothing

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Output %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
case 3
    sys = mdlOutputs(t,x,u);

otherwise
    DASTudio.error('Simulink:blocks:unhandledFlag', num2str(flag));
end

function [sys,x0,str,ts,simStateCompliance] = mdlInitializeSizes(lb,ub,xi)

sizes = simsizes;
sizes.NumContStates = 1;
sizes.NumDiscStates = 0;
sizes.NumOutputs = 1;
sizes.NumInputs = 1;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1;

sys = simsizes(sizes);
str = [];
x0 = xi;
ts = [0 0]; % sample time: [period, offset]

function sys = mdlDerivatives(t,x,u,lb,ub)

if (x <= lb & u < 0) | (x>= ub & u>0 )
    sys = 0;
else
    sys = u;
end

function sys = mdlOutputs(t,x,u)

sys = x;

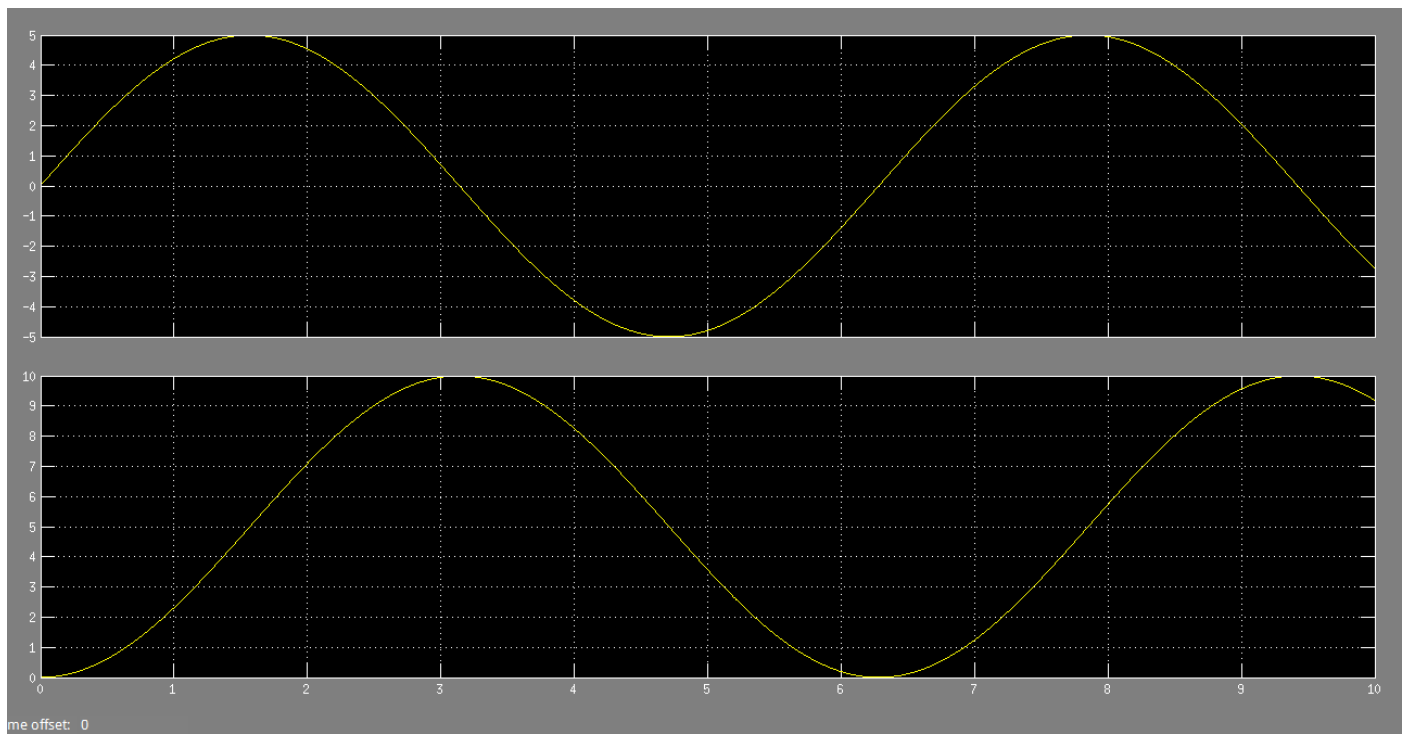
```

lb = -5.0

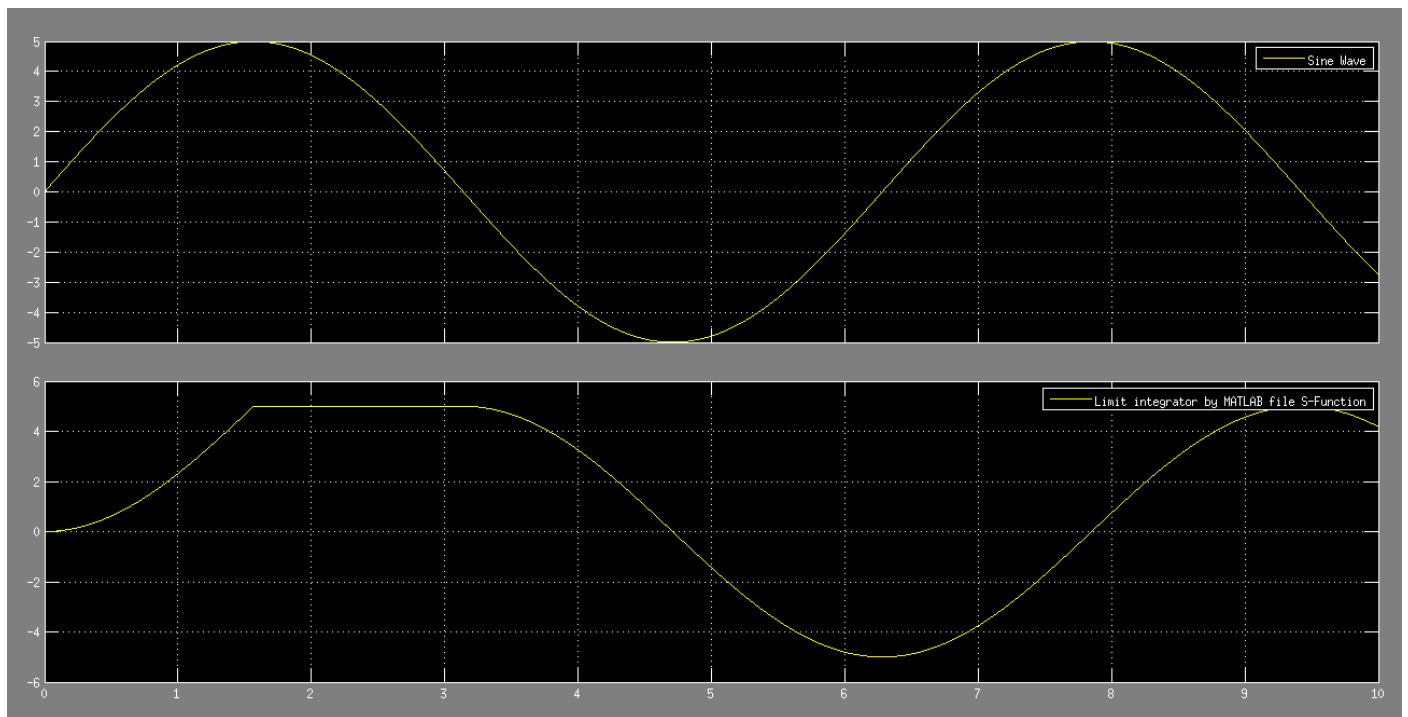
ub = 5.0

xi = 0

正常sin函数的积分



s函数,做了上下界的限制



## sfundsc2.m

```
function [sys,x0,str,ts,simStateCompliance] = sfundsc2(t,x,u,flag)
%SFUNDSC2 Example unit delay MATLAB File S-function
% The MATLAB file S-function is an example of how to implement a unit
% delay.
%
% See sfuntmpl.m for a general S-function template.
%
```



```

% See also SFUNTMPL.

% Copyright 1990-2009 The MathWorks, Inc.
% $Revision: 1.1.6.2 $

switch flag,

    %%%%%%%%%%%%%%%
    % Initialization %
    %%%%%%%%%%%%%%%
    case 0,
        [sys,x0,str,ts,simStateCompliance] = mdlInitializeSizes;

    %%%%%%%%%%%%%%%
    % Update %
    %%%%%%%%%%%%%%%
    case 2,
        sys = mdlUpdate(t,x,u);

    %%%%%%%%%%%%%%%
    % Output %
    %%%%%%%%%%%%%%%
    case 3,
        sys = mdlOutputs(t,x,u);

    %%%%%%%%%%%%%%%
    % Terminate %
    %%%%%%%%%%%%%%%
    case 9,
        sys = [];

    otherwise
        DASTudio.error('Simulink:blocks:unhandledFlag', num2str(flag));
end

%end sfundsc2

%
%=====
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-function.
%=====
%
function [sys,x0,str,ts,simStateCompliance]=mdlInitializeSizes

sizes = simsizes;

sizes.NumContStates = 0;
sizes.NumDiscStates = 1;
sizes.NumOutputs = 1;
sizes.NumInputs = 1;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1;

sys = simsizes(sizes);

x0 = 0;
str = [];
ts = [.1 0]; % Sample period of 0.1 seconds (10Hz)

```

```

% speicfy that the simState for this s-function is same as the default
simStateCompliance = 'DefaultSimState';

% end mdlInitializeSizes

%
%=====
% mdlUpdate
% Handle discrete state updates, sample time hits, and major time step
% requirements.
%=====
%
function sys = mdlUpdate(t,x,u)
sys = u;

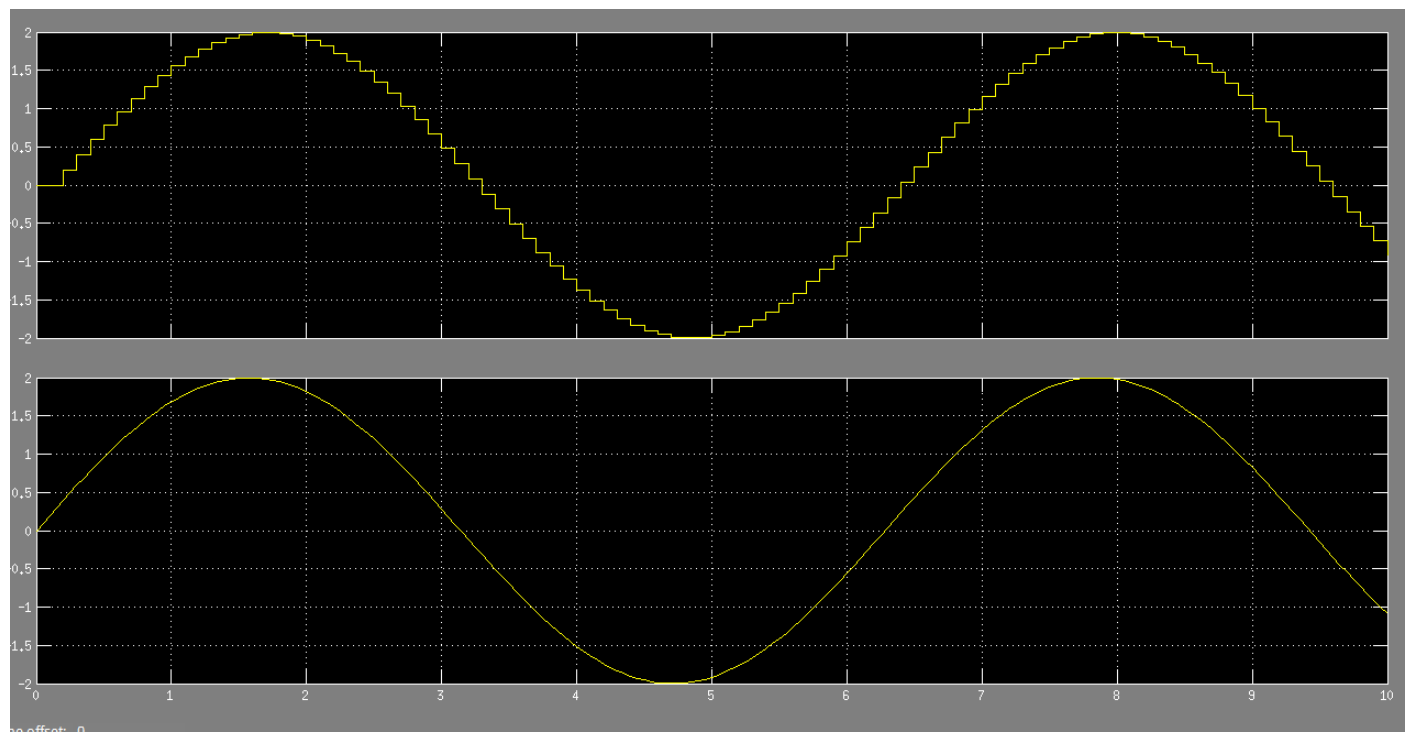
%end mdlUpdate

%
%=====
% mdlOutputs
% Return the output vector for the S-function
%=====
%
function sys = mdlOutputs(t,x,u)
sys = x;

%end mdlOutputs

```

离散,每0.1s进行一次采样,上图为处理后图像,下图为正弦图像



## timestwo.m

```

function [sys,x0,str,ts,simStateCompliance] = timestwo(t,x,u,flag)
%TIMESTWO S-function whose output is two times its input.

```

```

% This MATLAB file illustrates how to construct an MATLAB file S-function that
% computes an output value based upon its input. The output of this
% S-function is two times the input value:
%
%     y = 2 * u;
%
% See sfuntmpl.m for a general S-function template.
%
% See also SFUNTMPL.

% Copyright 1990-2009 The MathWorks, Inc.
% $Revision: 1.1.6.2 $

%
% Dispatch the flag. The switch function controls the calls to
% S-function routines at each simulation stage of the S-function.
%
switch flag,
    %%%%%%%%%%%%%%%
    % Initialization %
    %%%%%%%%%%%%%%%
    % Initialize the states, sample times, and state ordering strings.
    case 0
        [sys,x0,str,ts,simStateCompliance]=mdlInitializeSizes;

    %%%%%%%%%%%%%%%
    % Outputs %
    %%%%%%%%%%%%%%%
    % Return the outputs of the S-function block.
    case 3
        sys=mdlOutputs(t,x,u);

    %%%%%%%%%%%%%%%
    % Unhandled flags %
    %%%%%%%%%%%%%%%
    % There are no termination tasks (flag=9) to be handled.
    % Also, there are no continuous or discrete states,
    % so flags 1,2, and 4 are not used, so return an empty
    % matrix

    case { 1, 2, 4, 9}
        sys=[];

    %%%%%%%%%%%%%%%
    % Unexpected flags (error handling)%
    %%%%%%%%%%%%%%%
    % Return an error message for unhandled flag values.
    otherwise
        DAStudio.error('Simulink:blocks:unhandledFlag', num2str(flag));

end

% end timestwo

%
%=====
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-function.
%=====
%
```

```

function [sys,x0,str,ts,simStateCompliance] = mdlInitializeSizes()

sizes = simsizes;
sizes.NumContStates = 0;
sizes.NumDiscStates = 0;
sizes.NumOutputs = -1; % dynamically sized
sizes.NumInputs = -1; % dynamically sized
sizes.DirFeedthrough = 1; % has direct feedthrough
sizes.NumSampleTimes = 1;

sys = simsizes(sizes);
str = [];
x0 = [];
ts = [-1 0]; % inherited sample time

% specify that the simState for this s-function is same as the default
simStateCompliance = 'DefaultSimState';

% end mdlInitializeSizes

%
%=====
% mdlOutputs
% Return the output vector for the S-function
%=====
%
function sys = mdlOutputs(t,x,u)

sys = u * 2;

% end mdlOutputs

```

