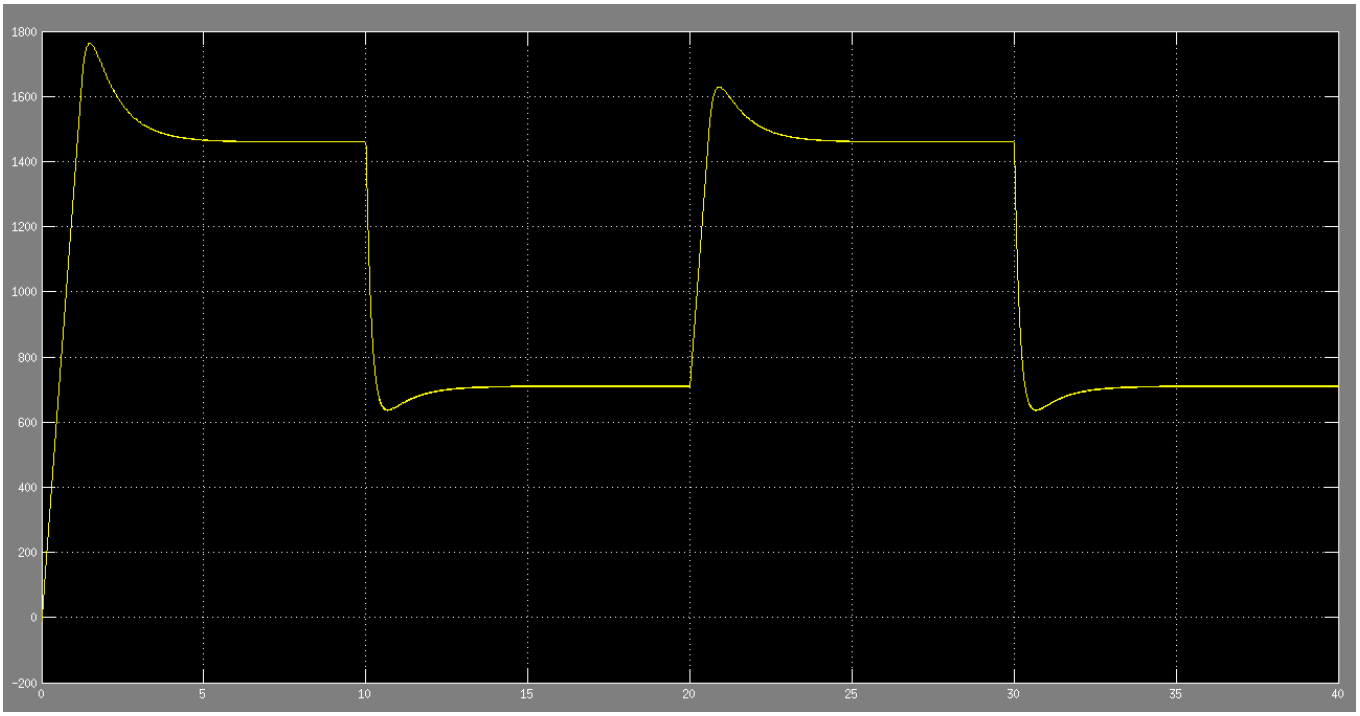
控制电流的S函数

```
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 $

dperiod = 10;
doffset = 0;

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,doffset,dperiod);

  %%%%%%%%%%%%
  % 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

% 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,doffset,dperiod)

%if abs(round((t - doffset) / dperiod) - (t - doffset) / dperiod) < 1e-3
if mod(fix( t / dperiod),2) == 1 & t > dperiod
    sys = x + 750;
else
    sys = x;
end
%end mdlOutputs
```



电流与温度仿真

电枢表面单位长度上的安培导体数称为电机的线负荷A , A=Nia/(πDa)。

A= (2×m×N×I) / (pi×D)

式中:W 为每相的线圈匝数 ; m 为相数 ; I 为相电流 ; D 为定子内圆直径 .

A=Nc*Z*I / pi*Di*a

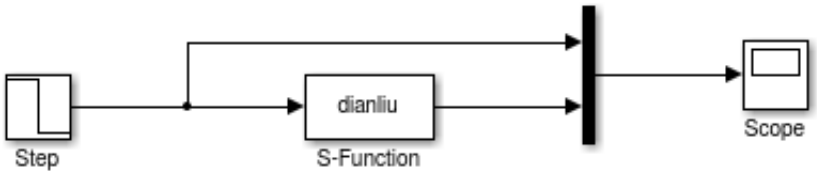
电流密度 J (电流强度 I 与导体截面 S 的比值) 为

J = I/ S

```
function [sys,x0,str,ts,simStateCompliance] = dianliu(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 $
%W = 12 %每相的线圈匝数
%m = 3%相数
D = 23.8%定子内圆直径
Ac = 7.71
S = 8.4%导体截面
T = 4.448%定子铁心的齿距
X = 0.5%绝缘厚度
```

```
zhouchang = 114%槽的周长
Pfe = 571.4%空载运行时铁心的损耗

la = 23.6%电枢铁心长度
lcp = 34.92%电枢绕组半砸长度
Pcu = 446.1748%电枢绕组的铜损耗
Pcuc = Pcu * la/lcp

K = 1.2 %损耗系数
Da = 17%电枢铁心内孔的圆周长
alpha_1 = 0.0057%发热表面再平静空气中的散热系数
alpha_2 = 0.00133
v1 = 13.20%转子外径的圆周速度
k0 = 0.1 %气流吹拂效率系数
k1 = 0.07 %端部气流吹拂效率系数
lend = 5.5%电枢绕组端部
T1 = 2%参数C/aS
global tmp




switch flag,
   %%%%%%%%%%%%%%%%%%
   % Initialization %
   %%%%%%%%%%%%%%%%%%
  case 0,
   [sys,x0,str,ts,simStateCompliance]=mdlInitializeSizes(tmp);
   %%%%%%%%%%
   % Update %
   %%%%%%%%%%
  case 2,
    sys = mdlUpdate(t,x,u,Ac,S,T,X,zhouchang,K,Pfe,Pcuc,Da,la,k0,v1,k1,lend,T
1,alpha_1,alpha_2,tmp);
   %%%%%%%%%%
   % 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(tmp)
sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  = 1;
sizes.NumOutputs     = 1;
sizes.NumInputs      = 1;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;
```

```
sys = simsizes(sizes);
x0  = 0;
str = [];
ts  = [0 0];
global tmp;
tmp = 0;

% 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,Ac,S,T,X,zhouchang,K,Pfe,Pcuc,Da,la,k0,v1,k1,l
end,T1,alpha_1,alpha_2,tmp)
%A = (2 * W * I * m) / (pi * D )
A = u * Ac;
J = u / S;

theta1 = (A * J * T* X) / (6.4 * zhouchang);


alpha1 = K * (Pfe + Pcuc)/ (pi * Da * la);

theta2 = alpha1/ (alpha_1 * (1 + k0 * v1));

alpha2 =  A * J * T  / (400 * zhouchang);

theta3 = alpha2 / (alpha_2 * (1 + k1 * v1));

if u > 0
    theta = ((theta1 + theta2) * la + (theta1 + theta3) * lend) / (la + lend)
;
else
    theta = 0;
end

wensheng =  (x + (theta -x) * (1 - exp(-1 / T1 * (t - tmp)))) ;

global tmp ;

tmp = t;

sys = wensheng;
%end mdlUpdate
%
%=============================================================================% mdl
Outputs
% Return the output vector for the S□function
%=============================================================================
%
function sys = mdlOutputs(t,x,u)
sys = x;
%end mdlOutputs
```


Step
dianliu
S-Function
Scope