

```

function [sys,x0,str,ts] = Testcsfunc(t,x,u,flag)

Rs=0.2                % Widerstand
Ld=8.5e-3            % Induktivität in d-Achse
Lq=8.5e-3            % Induktivität in q-Achse
F=0.175              % Polradfluss
J=0.0027             % Massenträgheitsmoment
P=4                  % Polpaar

switch flag,

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

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

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

    case { 2, 4, 9 }

end

% end sfuntmpl

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

sizes = simsizes;

sizes.NumContStates = 4;
sizes.NumDiscStates = 0;
sizes.NumOutputs = 4;
sizes.NumInputs = 3;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1; % at least one sample time is needed

sys = simsizes(sizes);

%
% initialize the initial conditions
%
x0 = zeros (4,1);

```

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

%
% initialize the array of sample times
%
ts = [0 0];

% Specify the block simStateCompliance. The allowed values are:
%   'UnknownSimState', < The default setting; warn and assume DefaultSimState
%   'DefaultSimState', < Same sim state as a built-in block
%   'HasNoSimState',   < No sim state
%   'DisallowSimState' < Error out when saving or restoring the model sim state
simStateCompliance = 'UnknownSimState';

% end mdlInitializeSizes

%
%=====
% mdlDerivatives
% Return the derivatives for the continuous states.
%=====
%
function sys=mdlDerivatives(t,x,u)

x1dot= -(Rs/Ld)*x(1)+(Lq/Ld)*x(3)*x(2)+(1/Ld)*u(1);
x2dot= -(Ld/Lq)*x(3)*x(1)-(Rs/Lq)*x(2)+(1/Lq)*u(2)-x(3)*(F/Lq);
x3dot= (3*P^2/2*J)*(Ld-Lq)*x(2)*x(1)+(3*P^2/2*J)*F*x(2)-(P/J)*u(3);
x4dot= x(3);

sys= [x1dot x2dot x3dot x4dot];

% end mdlDerivatives

%
%=====
% mdlUpdate
% Handle discrete state updates, sample time hits, and major time step
% requirements.
%=====
%
%=====
% mdlOutputs
% Return the block outputs.
%=====
%
```

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

sys = [x(1) x(2) x(3) x(4)];

% end mdlOutputs

%
%=====
% mdlGetTimeOfNextVarHit
% Return the time of the next hit for this block. Note that the result is
% absolute time. Note that this function is only used when you specify a
% variable discrete-time sample time [-2 0] in the sample time array in
% mdlInitializeSizes.
%=====
%
function sys=mdlGetTimeOfNextVarHit(t,x,u)

sampleTime = 1; % Example, set the next hit to be one second later.
sys = t + sampleTime;

% end mdlGetTimeOfNextVarHit

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

sys = [];

% end mdlTerminate
```