A Design Study Approach to Classical Control

Randal W. Beard   Timothy W. McLain
Brigham Young University

Updated: April 27, 2016

Homework B.b

Modify the simulink model created in homework ??.2 by creating an s-function that implements the equations of motion. The input to the s-function should be a slider for force. The output should go to the animation developed in homework ??.2.

Solution

The s-function is listed below.

```

function [sys,x0,str,ts,simStateCompliance]...
                    = pendulum_dynamics(t,x,u,flag,AP)
switch flag,

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

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

```
  %%%%%%%%%%
  % 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);

  %%%%%%%%%%%%%%%%%%%%
  % 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(AP)
```

```

sizes = simsizes;

sizes.NumContStates  = 4;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 2;
sizes.NumInputs      = 1;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1;

sys = simsizes(sizes);

%
% initialize the initial conditions
%
x0  = [AP.z0; AP.theta0; AP.zdot0; AP.thetadot0];

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

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

simStateCompliance = 'UnknownSimState';

% end mdlInitializeSizes

%
%=============================================================
% mdlDerivatives
% Return the derivatives for the continuous states.
%=============================================================
%
function sys=mdlDerivatives(t,x,u,AP)
  z        = x(1);
  theta    = x(2);
  zdot     = x(3);
  thetadot = x(4);
  F        = u(1);

  M = [...
```

```
      AP.m1+AP.m2,              AP.m1*AP.ell*cos(theta);...
      AP.m1*AP.ell*cos(theta),  AP.m1*AP.ell^2;...
      ];
  c = [...
      AP.m1*AP.ell*thetadot^2*sin(theta) + F - AP.b*zdot;...
      AP.m1*AP.g*AP.ell*sin(theta);...
      ];
  tmp = inv(M)*c;
  zddot     = tmp(1);
  thetaddot = tmp(2);

sys = [zdot; thetadot; zddot; thetaddot];

% end mdlDerivatives

%
%=========================================================
% mdlOutputs
% Return the block outputs.
%=========================================================
%
function sys=mdlOutputs(t,x,u,AP)
    z        = x(1);
    theta    = x(2);
    zdot     = x(3);
    thetadot = x(4);
sys = [z; theta];

% end mdlOutputs
```

For a complete solution to this problem, see the wiki associated with this book.