A Design Study Approach to Classical Control

Randal W. Beard Timothy W. McLain
Brigham Young University

Updated: December 28, 2020

Homework E.b

Modify the Simulink, Matlab, or Python model created in homework E.a by creating a function that implements the equations of motion. The input to the function should be a slider for force. The output should go to the animation developed in homework E.a.

Solution

The s-function is listed below.

```
function [sys,x0,str,ts,simStateCompliance]...
                         = ballbeam_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,AP);

end

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

sizes = simsizes;

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

sys = simsizes(sizes);
x0  = [AP.theta0; AP.y0; AP.thetadot0; AP.ydot0];
str = [];
ts  = [0 0];
simStateCompliance = 'UnknownSimState';


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

```
62    F         = u(1);
63
64    thetaddot = (1/((AP.m2*AP.L^2)/3+AP.m1*y^2))*...
65        (-2*AP.m1*y*ydot*thetadot-AP.m1*AP.g*y*cos(theta)...
66        -AP.m2*AP.g*AP.L/2*cos(theta)+AP.L*F*cos(theta));
67    thetaddot = (1/((AP.m2*AP.L^2)/3+AP.m1*y^2))*...
68        ( -AP.m2*AP.g*AP.L/2*cos(theta)+AP.L*F*cos(theta))
69    yddot = (1/AP.m1)*(AP.m1*y*thetadot^2-AP.m1*AP.g*sin(theta));
70
71    f= [thetadot; ydot; thetaddot; yddot];
72  sys = f;
73
74  %=============================================================
75  % mdlOutputs
76  % Return the block outputs.
77  %=============================================================
78  function sys=mdlOutputs(t,x,u,AP)
79
80  sys = x;
```

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