A Design Study Approach to Classical Control

Randal W. Beard Timothy W. McLain
Brigham Young University

Updated: December 28, 2020

Homework D.b

Modify the Simulink, Matlab, or Python model created in Homework D.2 by creating a function that implements the equations of motion. The input to the function should be a variable force. The output should go to the animation developed in Homework D.2.

Solution

The s-function is listed below.

```
function [sys,x0,str,ts,simStateCompliance] = mass_dynamics(t,x,u,flag,P)
switch flag

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

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

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

```
17    % Update %
18    %%%%%%%%%%
19    case 2
20      sys=mdlUpdate(t,x,u);
21
22    %%%%%%%%%%%
23    % Outputs %
24    %%%%%%%%%%%
25    case 3
26      sys=mdlOutputs(t,x,u);
27
28    %%%%%%%%%%%%%%%%%%%%%%%
29    % GetTimeOfNextVarHit %
30    %%%%%%%%%%%%%%%%%%%%%%%
31    case 4
32      sys=mdlGetTimeOfNextVarHit(t,x,u);
33
34    %%%%%%%%%%%%%
35    % Terminate %
36    %%%%%%%%%%%%%
37    case 9
38      sys=mdlTerminate(t,x,u);
39
40    %%%%%%%%%%%%%%%%%%%
41    % Unexpected flags %
42    %%%%%%%%%%%%%%%%%%%
43    otherwise
44      DAStudio.error('Simulink:blocks:unhandledFlag', num2str(flag));
45
46  end
47
48  % end sfuntmpl
49
50  %
51  %=============================================================================
52  % mdlInitializeSizes
53  % Return the sizes, initial conditions, and sample times for the
54  % S-function.
55  %=============================================================================
56  %
57  function [sys,x0,str,ts,simStateCompliance]=mdlInitializeSizes(P)
58
59  sizes = simsizes;
60
61  sizes.NumContStates  = 2;
```

```
62  sizes.NumDiscStates  = 0;
63  sizes.NumOutputs     = 1+2;
64  sizes.NumInputs      = 1;
65  sizes.DirFeedthrough = 0;
66  sizes.NumSampleTimes = 1;   % at least one sample time is needed
67
68  sys = simsizes(sizes);
69
70  %
71  % initialize the initial conditions
72  %
73  x0  = [P.z0; P.zdot0];
74
75  %
76  % str is always an empty matrix
77  %
78  str = [];
79
80  %
81  % initialize the array of sample times
82  %
83  ts  = [0 0];
84
85  simStateCompliance = 'UnknownSimState';
86
87  % end mdlInitializeSizes
88
89  %
90  %=============================================================
91  % mdlDerivatives
92  % Return the derivatives for the continuous states.
93  %=============================================================
94  %
95  function sys=mdlDerivatives(t,x,u,P)
96    z    = x(1);
97    zdot = x(2);
98    F    = u(1);
99
100   persistent m
101   persistent k
102   persistent b
103   if t==0
104     alpha = 0.0;  % uncertainty parameter
105     m = P.m * (1+2*alpha*rand-alpha);  % kg
106     k = P.k * (1+2*alpha*rand-alpha); % m
```

```
107     b = P.b * (1+2*alpha*rand-alpha); % N m
108   end
109
110   zddot = (F-b*zdot-k*z)/m;
111
112 sys = [zdot; zddot];
113
114 % end mdlDerivatives
115
116 %
117 %=============================================================
118 % mdlUpdate
119 % Handle discrete state updates, sample time hits, and major time
120 % step requirements.
121 %=============================================================
122 %
123 function sys=mdlUpdate(t,x,u)
124
125 sys = [];
126
127 % end mdlUpdate
128
129 %
130 %=============================================================
131 % mdlOutputs
132 % Return the block outputs.
133 %=============================================================
134 %
135 function sys=mdlOutputs(t,x,u)
136   z    = x(1);
137
138 sys = [z; x];
139
140 % end mdlOutputs
141
142 %
143 %=============================================================
144 % mdlGetTimeOfNextVarHit
145
146 %=============================================================
147 %
148 function sys=mdlGetTimeOfNextVarHit(t,x,u)
149
150 sampleTime = 1;
151 sys = t + sampleTime;
```

```
152
153 % end mdlGetTimeOfNextVarHit
154
155 %
156 %=============================================================================
157 % mdlTerminate
158 % Perform any end of simulation tasks.
159 %=============================================================================
160 %
161 function sys=mdlTerminate(t,x,u)
162
163 sys = [];
164
165 % end mdlTerminate
```

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