

# A Design Study Approach to Classical Control

Randal W. Beard      Timothy W. McLain  
Brigham Young University

Updated: April 27, 2016

## Homework A.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 torque. The output should go to the animation developed in homework ??2.

## Solution

The s-function is listed below.

```
1 function [sys,x0,str,ts,simStateCompliance] = ...
2                                     arm_dynamics(t,x,u,flag,AP)
3 switch flag,
4
5     %%%%%%%%%%%%%%%%%%%%%%%%%%
6     % Initialization %
7     %%%%%%%%%%%%%%%%%%%%%%%%%%
8     case 0,
9         [sys,x0,str,ts,simStateCompliance]=mdlInitializeSizes(AP);
10
11     %%%%%%%%%%%%%%%%%%%%%%%%%%
12     % Derivatives %
13     %%%%%%%%%%%%%%%%%%%%%%%%%%
14     case 1,
15         sys=mdlDerivatives(t,x,u,AP);
16
```

```

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

```

```

62 sizes = simsizes;
63
64 sizes.NumContStates = 2;
65 sizes.NumDiscStates = 0;
66 sizes.NumOutputs = 1;
67 sizes.NumInputs = 1;
68 sizes.DirFeedthrough = 0;
69 sizes.NumSampleTimes = 1;
70
71 sys = simsizes(sizes);
72
73 %
74 % initialize the initial conditions
75 %
76 x0 = [AP.theta0; AP.thetadot0];
77
78 %
79 % str is always an empty matrix
80 %
81 str = [];
82
83 %
84 % initialize the array of sample times
85 %
86 ts = [0 0];
87
88 simStateCompliance = 'UnknownSimState';
89
90 % end mdlInitializeSizes
91
92 %
93 %=====
94 % mdlDerivatives
95 % Return the derivatives for the continuous states.
96 %=====
97 %
98 function sys=mdlDerivatives(t,x,u,AP)
99     theta = x(1);
100     thetadot = x(2);
101     tau = u(1);
102
103     thetaddot = (3/AP.m/AP.ell^2)*...
104                 (tau - AP.b*thetadot ...
105                 - AP.m*AP.g*AP.ell/2*cos(theta));
106

```

```

107 sys = [thetadot; thetaddot];
108
109 % end mdlDerivatives
110
111
112 %
113 %=====
114 % mdlOutputs
115 % Return the block outputs.
116 %=====
117 %
118 function sys=mdlOutputs(t,x,u,P)
119
120 sys = x(1);
121
122 % end mdlOutputs

```

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