A Design Study Approach to Classical Control

Randal W. Beard Timothy W. McLain
Brigham Young University

Updated: April 27, 2016

Homework C.a

Create a simulink animation of the satellite system. The inputs should be sliders for θ and ϕ.

Solution

The m-file that implements the animation for the simple satellite system is listed below.

```
function drawSatellite(u)

    % process inputs to function
    theta   = u(1);
    phi     = u(2);
    t       = u(3);

    % drawing parameters
    L = 1;
    w = .3;

    % define persistent variables
    persistent base_handle
    persistent panel_handle

    % first time function is called, initialize plot
    % and persistent vars
```

```
18      if t==0,
19          figure(1), clf
20          track_width=3;
21          plot([-2*L,2*L],[0,0],'k--'); % plot track
22          hold on
23          base_handle   = drawBase(theta, w, []);
24          panel_handle  = drawPanel(phi, w, L, []);
25          axis([-2*L, 2*L, -2*L, 2*L]);
26
27
28      % at every other time step, redraw base and rod
29      else
30          drawBase(theta, w, base_handle);
31          drawPanel(phi, w, L, panel_handle);
32      end
33  end
34
35
36  %
37  %=======================================================
38  % drawBase
39  % draw the base of the pendulum
40  % return handle if 3rd argument is empty, otherwise use
41  % 3rd arg as handle
42  %=======================================================
43  %
44  function handle = drawBase(theta, w, handle)
45
46    % define points on base (without rotation)
47    pts = [...
48        w/2, -w/2;...
49        w/2, -w/6;...
50        w/2+w/6, -w/6;...
51        w/2+w/6,  w/6;...
52        w/2, w/6;...
53        w/2, w/2;...
54        -w/2, w/2;...
55        -w/2, w/6;...
56        -w/2-w/6, w/6;...
57        -w/2-w/6, -w/6;...
58        -w/2, -w/6;...
59        -w/2, -w/2;...
60        ]';
61    % define rotation matrix
62    R = [cos(theta), sin(theta); -sin(theta), cos(theta)];
```

```
63    % rotate points
64    pts = R*pts;
65    % break into X and Y components
66    X = pts(1,:);
67    Y = pts(2,:);
68
69    if isempty(handle),
70      handle = fill(X,Y,'b');
71    else
72      set(handle,'XData',X,'YData',Y);
73      drawnow
74    end
75  end
76
77  %
78  %============================================================
79  % drawPanel
80  % draw the solar panel
81  % return handle if 3rd argument is empty, otherwise use
82  % 3rd arg as handle
83  %============================================================
84  %
85  function handle = drawPanel(phi, w, L, handle)
86
87  % define points on base (without rotation)
88    pts = [...
89        -L, -w/6;...
90         L, -w/6;...
91         L,  w/6;...
92        -L,  w/6;...
93        ]';
94    % define rotation matrix
95    R = [cos(phi), sin(phi); -sin(phi), cos(phi)];
96    % rotate points
97    pts = R*pts;
98    % break into X and Y components
99    X = pts(1,:);
100   Y = pts(2,:);
101
102   if isempty(handle),
103     handle = fill(X, Y, 'g');
104   else
105     set(handle,'XData',X,'YData',Y);
106     drawnow
107   end
```

```
108 end
```

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