A Design Study Approach to Classical Control

Randal W. Beard   Timothy W. McLain
Brigham Young University

Updated: December 28, 2020

Homework F.a

Create a simulink animation of the planar VTOL system. The inputs should be sliders for z_v, z_t, h, and θ. Turn in a screen capture of the animation.

Solution

The drawing function for the ball on beam system is listed below.

```
function VTOL_animation(u, P)

    % process inputs to function
    z          = u(1);
    h          = u(2);
    theta      = u(3);
    %z_dot      = u(4);
    %h_dot      = u(5);
    %theta_dot = u(6);
    target     = u(7);
    t          = u(8);

    % define persistent variables
    persistent VTOL_handle
    persistent target_handle

    L = 10;

```

```
    % first time function called, initialize plot and persistents
    if t==0,
        figure(1), clf
        plot([0,L],[0,0],'k'); % plot track
        hold on
        VTOL_handle    = drawVehicle(z, h, theta, []);
        target_handle  = drawTarget(target, []);
        axis([-L/5, L+L/5, -L, L]);


    % at every other time step, redraw base and rod
    else
        drawVehicle(z, h, theta, VTOL_handle);
        drawTarget(target, target_handle);
    end
end


%
%=================================================================
% drawVTOL
% draw VTOL system
% return handle if 3rd argument is empty, otherwise use 3rd arg
% as handle
%=================================================================
%
function handle = drawVehicle(z, h, theta, handle)

  x1 = 0.1;
  x2 = 0.3;
  x3 = 0.4;
  y1 = 0.05;
  y2 = 0.01;
  pts = [...
      x1, y1;...
      x1, 0;...
      x2, 0;...
      x2, y2;...
      x3, y2;...
      x3, -y2;...
      x2, -y2;...
      x2, 0;...
      x1, 0;...
      x1, -y1;...
      -x1, -y1;...
```

```
      -x1, 0;...
      -x2, 0;...
      -x2, -y2;...
      -x3, -y2;...
      -x3, y2;...
      -x2, y2;...
      -x2, 0;...
      -x1, 0;...
      -x1, y1;...
      x1, y1;...
      ];
  % rotate points (must do first)
  R = [cos(theta), sin(theta); -sin(theta), cos(theta)];
  pts = pts*R;
  % translate points
  pts = pts + repmat([z,h],size(pts,1),1);

  if isempty(handle),
    handle = fill(pts(:,1),pts(:,2),'b');
  else
    set(handle,'XData',pts(:,1),'YData',pts(:,2));
    drawnow
  end
end

%
%==================================================================
% drawTarget
% draw the Target
% return handle if 3rd argument is empty, otherwise use 3rd arg
% as handle
%==================================================================
%
function handle = drawTarget(z, handle)

  w = 0.1;
  h = 0.05;
  pts = [...
      w/2, h;...
      w/2, 0;...
      -w/2, 0;...
      -w/2, h;...
      w/2, h;...
      ];

```

```
    % translate points
  pts = pts + repmat([z,0],size(pts,1),1);

  if isempty(handle),
    handle = fill(pts(:,1), pts(:,2), 'r');
  else
    set(handle,'XData',pts(:,1),'YData',pts(:,2));
    drawnow
  end
end
```

The complete solution is given on the wiki associated with the book.