Simulation with PD control

```
fprintf('\n');
clearvars -except function_list pub_opt
close all
PDGains
% theta0_121=[0;60;0]*pi/180;%rad
theta0_121=[45;60;-75]*pi/180;%rad
inertia_P2B_EA_321 = [0.563443998581290;0.682063787756298;0.466175771457783]*1*pi/
inertia_P2B_DCM = Euler2DCM('321',inertia_P2B_EA_321);

fprintf('MRP vector:\n');
MRP = DCM2MRP(Euler2DCM('121',theta0_121));
printVector(MRP, '');
omega_body = [0; 0; 0]*pi/180; %rad/s
I_body=[14 0 0;0 16 0;0 0 19]; %kg*m2
I_body=inertia_P2B_DCM'*I_body*inertia_P2B_DCM;
cm_torque=[0;0;0];

delta_t = 0.01;
t_end = 5000 - delta_t; % seconds

% Arrays for recording and plotting
t_mat = 0:delta_t:t_end+delta_t;
[rows, cols] = size(t_mat);
MRP_mat = zeros(3,cols);
omega_mat = zeros(3,cols);
EA_mat = zeros(3,cols);
mode_mat = zeros(3,cols);
CT_mat = zeros(3,cols);
MRP_mat(:,1) = MRP;
omega_mat(:,1) = omega_body;
mode_mat(:,1) = 0;
CT_MAT(:,1) = cm_torque;
idx = 2;

% RK4 integration
state = [MRP; omega_body];
mode = 0;
for t = 0:delta_t:t_end

  euler_angles = DCM2Euler('121',MRP2DCM(state(1:3)));
  if abs(euler_angles(2)) < pi/180%1e-2
%       euler_angles(2)=0;
      euler_angles(1)=euler_angles(3)+euler_angles(1);
      while euler_angles(1) > pi
          euler_angles(1) = euler_angles(1) - 2*pi;
      end
      while euler_angles(1) < -pi
          euler_angles(1) = euler_angles(1) + 2*pi;
      end
      euler_angles(3)=0;
  end
  EA_mat(:,idx-1) = euler_angles;
%   angle_deadband1 = 1e-1;
%   rate_deadband1 = 1e-5;
  angle_deadband1 = 1e-4;
  rate_deadband1 = 1e-5;
  angle_deadband2 = 1e-4;
  rate_deadband2 = 1e-5;
  if (abs(euler_angles(3)) > angle_deadband1 || abs(state(4)) > rate_deadband1) &&
      mode=1;
     cm_torque=[-K(1,1)*euler_angles(3) - K(1,2)*state(4);0;0];
  elseif (abs(euler_angles(2)) > angle_deadband2 || abs(state(5)) > rate_deadband2
      mode=2;
     cm_torque=[0;-K(2,1)*euler_angles(2) - K(2,2)*state(5);0];
  else%if abs(euler_angles(1)) > angle_deadband1 || abs(state(4)) > rate_deadband
      mode=3;
     cm_torque=[-K(1,1)*euler_angles(1) - K(1,2)*state(4);0;0];
  end
  CT_mat(:,idx) = cm_torque;
    mode_mat(:,idx) = mode;

  k1 = [derivMRP(state(1:3), state(4:6)); ...
      dBodyRatesRigid(state(4:6), I_body, cm_torque)];
  k2 = [derivMRP(state(1:3) + delta_t*k1(1:3)/2, ...
      state(4:6) + delta_t*k1(4:6)/2); ...
      dBodyRatesRigid(state(4:6) + delta_t*k1(4:6)/2, I_body, cm_torque)];
  k3 = [derivMRP(state(1:3) + delta_t*k2(1:3)/2, ...
      state(4:6) + delta_t*k2(4:6)/2); ...
      dBodyRatesRigid(state(4:6) + delta_t*k2(4:6)/2, I_body, cm_torque)];
  k4 = [derivMRP(state(1:3) + delta_t*k3(1:3), ...
      state(4:6) + delta_t*k3(4:6)); ...
      dBodyRatesRigid(state(4:6) + delta_t*k3(4:6), I_body, cm_torque)];

  state = state + delta_t/6*(k1 + 2*k2 + 2*k3 + k4);

  % Enforce |MRP| <= 1, switch to shadow set if needed
  if norm(state(1:3)) > 1
      state(1:3) = -state(1:3)/dot(state(1:3), state(1:3));
  end

  % Updating array
  MRP_mat(:,idx) = state(1:3);
  omega_mat(:,idx) = state(4:6);
  idx = idx + 1;
end

font_size=8;
figure
plot(t_mat, MRP_mat);
title('MRP Propagation','FontSize',font_size)
xlabel('time(s)','FontSize',font_size)
ylabel('Element Magnitude','FontSize',font_size)
legend('\sigma_{1}', '\sigma_{2}', '\sigma_{3}')
grid on
set(gca,'FontSize',font_size)
fprintf('\n\n\n');
figure
plot(t_mat, omega_mat*180/pi);
title('Body Rates','FontSize',font_size)
xlabel('time(s)','FontSize',font_size)
ylabel('Element Magnitude (deg/s)','FontSize',font_size)
legend('\omega_{1}', '\omega_{2}', '\omega_{3}')
grid on
set(gca,'FontSize',font_size)
fprintf('\n\n\n');
figure
plot(t_mat, EA_mat*180/pi)
title('Euler Angles','FontSize',font_size)
xlabel('time(s)','FontSize',font_size)
ylabel('Angles (deg)','FontSize',font_size)
legend('\theta_{1}', '\theta_{2}', '\theta_{3}')
grid on
set(gca,'FontSize',font_size)
fprintf('\n\n\n');

figure
plot(t_mat, mode_mat);

title('Axis control mode','FontSize',font_size)
xlabel('time(s)','FontSize',font_size)
ylabel('Mode)','FontSize',font_size)
figure

plot(t_mat, CT_mat);

title('Control Torque','FontSize',font_size)
xlabel('time(s)','FontSize',font_size)
ylabel('Torque (N*m)','FontSize',font_size)
grid on
set(gca,'FontSize',font_size)
legend('u_{1}', 'u_{2}', 'u_{3}')
```

Published with MATLAB® R2013b