Simulation with PD control

```
fprintf('\n');
clearvars -except function_list pub_opt
close all
MRP0=[0;0.8;0];%rad

fprintf('MRP vector:\n');
MRP = MRP0;
printVector(MRP, '');
omega_body0 = [0; 2; 0]; %rad/s
fprintf('Initial Body Rates:\n');
printVector(omega_body0, 'rad/s');
I_body=[10 0 0;0 20 0;0 0 30]; %kg*m2

%unit gains
K = eye(3)*6;
P = eye(3)*6;
cm_torque=[0;0;0];
disturbance_torque=[1;2;-1]; %Nm

delta_t = 0.01;
t_end = 200 - delta_t; % seconds
for scenario = 1:6
    use_dist = 0;
    with_dist = '';
    scen_str = '';
    MRP = MRP0;
    omega_body = omega_body0;
% 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];

for t = 0:delta_t:t_end
    if scenario == 1 || scenario == 4
        control_torque = -K*state(1:3) -P*state(4:6);
        scen_str = ' MRP Feedback control';
    elseif scenario == 2 || scenario == 5
        quat = MRP2quat(state(1:3));
```

```matlab
            eps = quat(2:4);
            control_torque = -K*eps -P*state(4:6);
            scen_str = ' Quaternion Feedback control';
        elseif scenario == 3 || scenario == 6
            euler_angles = DCM2Euler('321',MRP2DCM(state(1:3)));
            control_torque = -BmatEuler('321', euler_angles)'...
                *K*euler_angles -P*state(4:6);
            scen_str = ' 321 Euler Angle Feedback control';
        end
        if scenario >= 4
            use_dist = 1;
            with_dist = ' with Disturbance Torque';
        end

        cm_torque = control_torque + disturbance_torque*use_dist;
      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);
        if scenario == 2 || scenario == 5
            quat = MRP2quat(state(1:3));
            eps = quat(2:4);
            MRP_mat(:,idx) = eps;
        elseif scenario == 3 || scenario == 6
            euler_angles = DCM2Euler('321',MRP2DCM(state(1:3)));
            MRP_mat(:,idx) = euler_angles;
        end
      omega_mat(:,idx) = state(4:6);
      CT_mat(:,idx)=control_torque;
      idx = idx + 1;
    end

    if scenario == 4
        fprintf('MRP SS Error: ')
        printVector(state(1:3), '');
    elseif scenario == 5
```

```
    quat = MRP2quat(state(1:3));
    eps = quat(2:4);
    fprintf('EP SS Error: ')
    printVector(eps, '');
elseif scenario == 6
    euler_angles = DCM2Euler('321',MRP2DCM(state(1:3)));
    fprintf('SS B(theta) * Euler Angle SS Error: ')
    printVector(BmatEuler( '321', euler_angles )*euler_angles, '');
end

fprintf('\n\n\n');
font_size=8;
figure
plot(t_mat, MRP_mat);
mytitle = strcat('MRP Propagation for',scen_str, with_dist);
title(mytitle,'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)

figure
plot(t_mat, omega_mat*180/pi);
mytitle = strcat('Body Rates for',scen_str, with_dist);
title(mytitle,'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)

figure
plot(t_mat, CT_mat);
mytitle = strcat('Control Torque for',scen_str, with_dist);
title(mytitle,'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}')
end
```

Published with MATLAB® R2013b