Analytical Mechanics Report #4

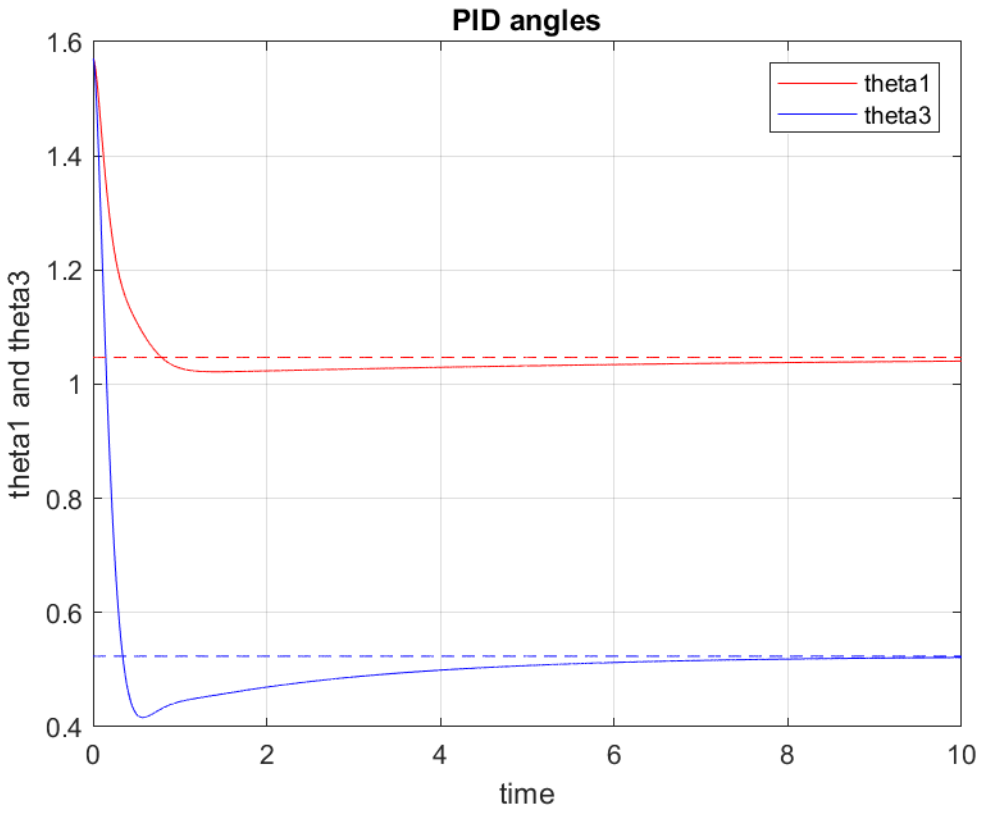
```
function dotq = closed_mechanism_2DOF_PID_params (t, q, robot, thetad, Kp, Ki, Kd)
% equation of motion of 2DOF closed mechanism under PID control
    disp(num2str(t,"%8.6f"));
    persistent alpha
    if isempty(alpha)
        alpha = 1000;
    end

    theta = q(1:4);
    omega = q(5:8);
    etha = q(9:10);

    dottheta = omega;

    theta12 = theta(1:2); omega12 = omega(1:2);
    theta34 = theta(3:4); omega34 = omega(3:4);
    theta13 = theta([1,3],:);
    robot.left_arm = robot.left_arm.joint_angles (theta12, omega12);
    robot.right_arm = robot.right_arm.joint_angles (theta34, omega34);

    J12 = robot.left_arm.Jacobian;
    J34 = robot.right_arm.Jacobian;
    [ Q12x, Q12y ] = robot.left_arm.Hessian;
    [ Q34x, Q34y ] = robot.right_arm.Hessian;

    [inertia_matrix_12, torque_vector_12] =
robot.left_arm.inertia_matrix_and_torque_vector;
    [inertia_matrix_34, torque_vector_34] =
robot.right_arm.inertia_matrix_and_torque_vector;

    R = robot.left_arm.tip_point - robot.right_arm.tip_point;
    dotR = J12*omega12 - J34*omega34;

    C = [ omega12'*Q12x*omega12 - omega34'*Q34x*omega34; ...
          omega12'*Q12y*omega12 - omega34'*Q34y*omega34 ];
    C = C + 2*alpha*dotR + (alpha^2)*R;

    dotetha = theta13 - thetad;
    %[theta(1) - thetad(1); theta(3) - thetad(2)];
    %theta - thetad;
    tau_left  = [ -Kp(1)*(theta(1) - thetad(1)) - Ki(1)*etha(1) - Kd(1)*omega(1); 0 ];
    tau_right = [ -Kp(2)*(theta(3) - thetad(2)) - Ki(2)*etha(2) - Kd(2)*omega(3); 0 ];

    A = [ inertia_matrix_12,        zeros(2,2), -J12'; ...
                zeros(2,2), inertia_matrix_34,  J34'; ...
                      -J12,               J34, zeros(2,2) ];
    b = [ torque_vector_12 + tau_left; ...
```

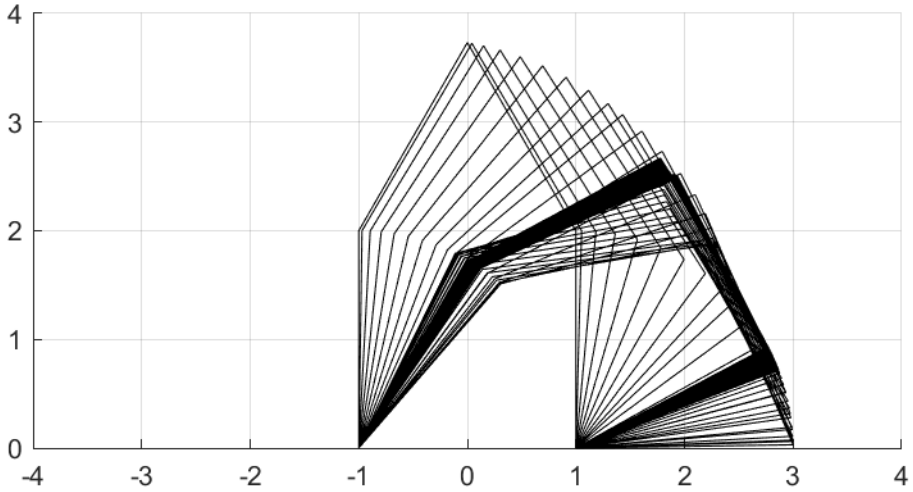
```
            torque_vector_34 + tau_right; ...
            C ];

    p = A \ b;
    dotomega = p(1:4);

    dotq = [dottheta; dotomega; dotetha];
end
```

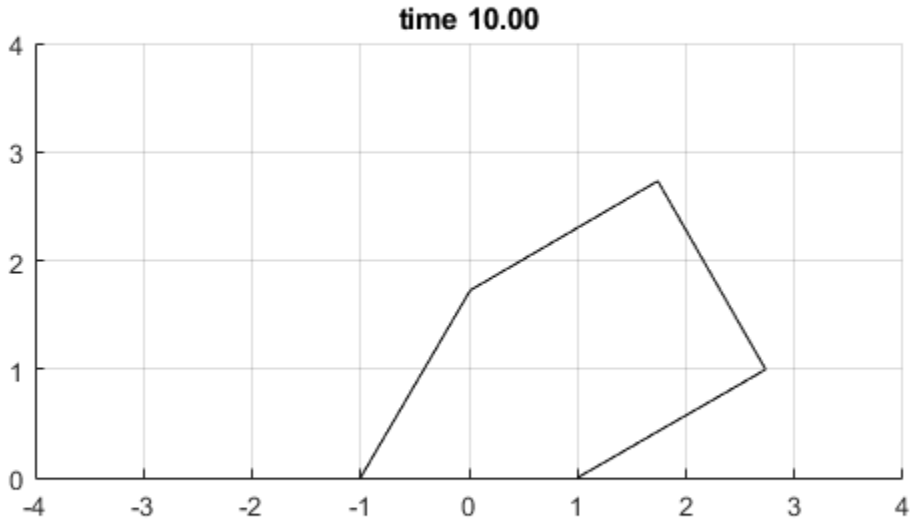
```
% Solver PID control of 2DOF closed mechanism

len = 2.00; radius = 0.05; density = 1;
link1 = Link_Cylinder ( len, radius, density );
link2 = Link_Cylinder ( len, radius, density );
base_left = [-1; 0];
link3 = Link_Cylinder ( len, radius, density );
link4 = Link_Cylinder ( len, radius, density );
base_right = [1; 0];
grav = [0; -9.8];
left_arm  = Open_Mechanism_Two_DOF (link1, link2, base_left,  grav);
right_arm = Open_Mechanism_Two_DOF (link3, link4, base_right, grav);
robot = Closed_Mechanism_Two_DOF (left_arm, right_arm);

thetad = [ pi/3; pi/6 ]; Kp = [ 10; 15 ]; Kd = [ 2.5; 2.5]; Ki = [ 1.5; 5.5];
closed_mechanism_2DOF_PD_ode = @(t,q) closed_mechanism_2DOF_PID_params (t, q, robot,
thetad, Kp, Ki, Kd);

interval = [ 0, 10 ];
thetainit = [ pi/2; -pi/6; pi/2; pi/6 ]; omegainit = [ 0; 0; 0; 0 ]; ethainit = [ 0; 0];
qinit = [thetainit; omegainit; ethainit];
[time, q] = ode45(closed_mechanism_2DOF_PD_ode, interval, qinit);

thetad1 = thetad(1)*ones(size(time));
thetad3 = thetad(2)*ones(size(time));
plot(time, q(:,1), 'r-', time, q(:,3), 'b-', ...
     time, thetad1, 'r--', time, thetad3, 'b--');
title('PID angles');
xlabel('time');
ylabel('theta1 and theta3');
legend('theta1','theta3');
grid on;
saveas(gcf, 'closed_mechanism_2DOF_PID_angles.png');
fprintf("pause: press any key\n");
pause;

clf;
sz = size(time);
for k=1:100:sz(1)
    theta = q(k,1:4);
    omega = q(k,5:8);
```

```
    robot = robot.joint_angles (theta, omega);
    robot.draw;
    xlim([-4,4]);
    ylim([ 0,4]);
    pbaspect([2 1 1]);
    grid on;
    pause(0.01);
end
fprintf("pause: press any key\n");
pause;

clf;
hold on;
for k=1:100:sz(1)
    theta = q(k,1:4);
    omega = q(k,5:8);
    robot = robot.joint_angles (theta, omega);
    robot.draw;
    xlim([-4,4]);
    ylim([ 0,4]);
    pbaspect([2 1 1]);
    grid on;
    pause(0.01);
end
hold off;
saveas(gcf, 'closed_mechanism_2DOF_PID_draw.png');

clf('reset');
ts = time(1);
te = time(end);
fr = 1;
clear M;
for t = ts:0.01:te
    clf;
    k = nearest_index(time, t);
    theta = q(k,1:4);
    omega = q(k,5:8);
    robot = robot.joint_angles (theta, omega);
    robot.draw;
    xlim([-4,4]);
    ylim([ 0,4]);
    pbaspect([2 1 1]);
    title(['time ' num2str(t,"%3.2f")]);
    grid on;
    M(fr) = getframe(gcf);
    fr = fr + 1;
end
M(fr) = getframe(gcf);

v = VideoWriter('closed_mechanism_2DOF_PID_draw', 'MPEG-4');
```

```
open(v);
writeVideo(v, M);
close(v);
```







Published with MATLAB® R2022b