```
% HydArmMATLABSim.m
% Pure MATLAB simulation of a 3-link hydraulic robotic arm
% driven by solenoid-valve torques, with PD control.

clear; close all; clc;

%% Arm Parameters
L1 = 0.6;    % link 1 length [m]
L2 = 0.5;    % link 2 length [m]
L3 = 0.4;    % link 3 length [m]
m1 = 2.0;    % link 1 mass [kg]
m2 = 1.5;    % link 2 mass [kg]
m3 = 1.0;    % link 3 mass [kg]
I1 = 0.02;   % link 1 moment of inertia [kg·m^2]
I2 = 0.015;  % link 2 moment of inertia
I3 = 0.01;   % link 3 moment of inertia
g  = 9.81;   % gravity [m/s^2]

%% PD Controller Gains
Kp = diag([150, 120, 100]);   % proportional gains
Kd = diag([20,  15,  10]);    % derivative gains

%% Desired Trajectories (constant steps)
qd = deg2rad([45; 30; 15]);   % desired joint angles [rad]
qdp = [0;0;0];                % desired joint velocities

%% ODE Integration Settings
tspan = [0 5];                % simulate for 5 seconds
q0    = deg2rad([0; 0; 0]);   % initial joint angles
```

```
qd0   = [0;0;0];            % initial joint velocities
x0    = [q0; qd0];          % state vector

opts = odeset('RelTol',1e-4,'AbsTol',1e-6);

%% Run Simulation
[t, x] = ode45(@armDynamics, tspan, x0, opts);

%% Extract Results
q   = x(:,1:3);
qd_ = x(:,4:6);

%% Plot Joint Responses
figure('Position',[100 100 800 600]);
subplot(2,1,1);
plot(t, rad2deg(q),'LineWidth',1.5);
title('Joint Angles vs Time'); grid on;
xlabel('Time (s)'); ylabel('Angle (deg)');
legend('q_1','q_2','q_3');

subplot(2,1,2);
plot(t, rad2deg(qd_),'LineWidth',1.5);
title('Joint Velocities vs Time'); grid on;
xlabel('Time (s)'); ylabel('Velocity (deg/s)');
legend('qd_1','qd_2','qd_3');

%% Animate Arm Motion
figure('Position',[950 100 600 600]);
for k = 1:10:length(t)
```

```
    cla; hold on; axis equal;
    drawArm(q(k,:), L1, L2, L3);
    title(sprintf('t = %.2f s', t(k)));
    xlim([-1.2,1.2]); ylim([-0.2,1.2]);
    drawnow;
end

%% Dynamics Function
function dx = armDynamics(t, x)
    % States
    q   = x(1:3);    % angles
    qd  = x(4:6);    % velocities

    % Mass Matrix M(q)
    M11 = I1 + I2 + I3 + m1*(L1/2)^2 + m2*(L1^2 + (L2/2)^2 + 2*L1*(L2/2)*cos(q(2))) ...
        + m3*(L1^2 + L2^2 + (L3/2)^2 + 2*L1*L2*cos(q(2)) + 2*L2*(L3/2)*cos(q(3)) ...
        + 2*L1*(L3/2)*cos(q(2)+q(3)));
    M12 = I2 + I3 + m2*((L2/2)^2 + L1*(L2/2)*cos(q(2))) ...
        + m3*(L2^2 + (L3/2)^2 + L1*L2*cos(q(2)) + 2*L2*(L3/2)*cos(q(3)) ...
        + L1*(L3/2)*cos(q(2)+q(3)));
    M13 = I3 + m3*((L3/2)^2 + L2*(L3/2)*cos(q(3)) + L1*(L3/2)*cos(q(2)+q(3)));
    M22 = I2 + I3 + m2*(L2/2)^2 + m3*(L2^2 + (L3/2)^2 + 2*L2*(L3/2)*cos(q(3)));
    M23 = I3 + m3*((L3/2)^2 + L2*(L3/2)*cos(q(3)));
    M33 = I3 + m3*(L3/2)^2;
    M = [M11, M12, M13; M12, M22, M23; M13, M23, M33];

    % Coriolis & Centrifugal Vector C(q,qd)
    h1 = -m2*L1*(L2/2)*sin(q(2)) - m3*(L1*L2*sin(q(2)) + L1*(L3/2)*sin(q(2)+q(3)));
    h2 = -m3*L2*(L3/2)*sin(q(3));
```

```
    C1 = h1*(2*qd(1)*qd(2)+qd(2)^2) + h2*(2*qd(1)*qd(3)+2*qd(2)*qd(3)+qd(3)^2);
    C2 = h1*qd(1)^2 + h2*(2*qd(2)*qd(3)+qd(3)^2);
    C3 = h2*(qd(1)^2 + qd(2)^2 + 2*qd(1)*qd(2));
    C  = [C1; C2; C3];

    % Gravity Vector G(q)
    G1 = (m1*(L1/2)+m2*L1+m3*L1)*g*cos(q(1)) + (m2*(L2/2)+m3*L2)*g*cos(q(1)+q(2)) ...
        + m3*(L3/2)*g*cos(q(1)+q(2)+q(3));
    G2 = (m2*(L2/2)+m3*L2)*g*cos(q(1)+q(2)) + m3*(L3/2)*g*cos(q(1)+q(2)+q(3));
    G3 = m3*(L3/2)*g*cos(q(1)+q(2)+q(3));
    Gv = [G1; G2; G3];

    % PD Control Law (torque from solenoid-driven hydraulics)
    global Kp Kd qd qdp
    tau = Kp*(qd - q) + Kd*(qdp - qd);

    % Solve for accelerations
    qdd = M \ (tau - C - Gv);

    % Pack state derivative
    dx = [qd; qdd];
end

%% Arm Drawing Function
function drawArm(q, L1, L2, L3)
    % Compute joint positions
    p0 = [0; 0];
    p1 = p0 + L1*[cos(q(1)); sin(q(1))];
    p2 = p1 + L2*[cos(q(1)+q(2)); sin(q(1)+q(2))];
```

```
    p3 = p2 + L3*[cos(q(1)+q(2)+q(3)); sin(q(1)+q(2)+q(3))];

    % Plot links
    plot([p0(1) p1(1)], [p0(2) p1(2)], 'k-', 'LineWidth', 4);
    plot([p1(1) p2(1)], [p1(2) p2(2)], 'k-', 'LineWidth', 4);
    plot([p2(1) p3(1)], [p2(2) p3(2)], 'k-', 'LineWidth', 4);

    % Plot joints
    plot(p0(1), p0(2), 'ko', 'MarkerSize', 8, 'MarkerFaceColor','k');
    plot(p1(1), p1(2), 'ko', 'MarkerSize', 8, 'MarkerFaceColor','k');
    plot(p2(1), p2(2), 'ko', 'MarkerSize', 8, 'MarkerFaceColor','k');
    plot(p3(1), p3(2), 'ro', 'MarkerSize', 10, 'MarkerFaceColor','r');
end
```