```
% UR5 Interactive Controller
% ME5250 project 2 - topic 1

clear; clc; close all;

% global variables
global DH q_current path_history ax trans_inc rot_inc

% DH parameters
DH.a = [0, -0.425, -0.39225, 0, 0, 0];
DH.d = [0.089159, 0, 0, 0.10915, 0.09465, 0.0823];
DH.alpha = [pi/2, 0, 0, pi/2, -pi/2, 0];
DH.n = 6;

function T = dh_matrix(a, d, alpha, theta)
    ct = cos(theta); st = sin(theta);
    ca = cos(alpha); sa = sin(alpha);
    T = [ct, -st*ca,  st*sa, a*ct;
         st,  ct*ca, -ct*sa, a*st;
          0,     sa,     ca,    d;
          0,      0,      0,    1];
end

function [T, T_all] = fk_ur5(q, DH)
    T = eye(4);
    T_all = zeros(4, 4, DH.n+1);
    T_all(:,:,1) = T;
    for i = 1:DH.n
        Ti = dh_matrix(DH.a(i), DH.d(i), DH.alpha(i), q(i));
        T = T * Ti;
        T_all(:,:,i+1) = T;
    end
end

q_start = [pi; 0; 0; 0; pi; 0];

[T_start, ~] = fk_ur5(q_start, DH);
fprintf('Starting position: [%.4f, %.4f, %.4f] m\n', T_start(1:3,4));
fprintf('Starting orientation (R matrix):\n');
disp(T_start(1:3,1:3));

% Check if position is in positive X region
if T_start(1,4) < 0
    fprintf('Adjusting...\n');
end


q_current = q_start;
[T0,~] = fk_ur5(q_current, DH);
path_history = T0(1:3,4)';
```

```
trans_inc = 0.1;      % 100mm increment translation
rot_inc = 30*pi/180;  % 30 degrees increment rotation

% gui making
fig = figure('Name','UR5 Interactive Controller v2','Position',[50 50 1400
800],...
    'NumberTitle','off','Color',[0.94 0.94 0.94]);

ax = axes('Parent',fig,'Position',[0.02 0.1 0.58 0.85]);

panel = uipanel('Parent',fig,'Title','Task-Space Control',...
    'Position',[0.62 0.02 0.36 0.96],'FontSize',12,'FontWeight','bold');

% buttons appearance
btn_bg = [0 0 0];      % black background
btn_fg = [1 1 1];      % white text

% translation buttons
uicontrol('Parent',panel,'Style','text','String','TRANSLATION (100mm)',...
    'Units','normalized','Position',[0.05 0.92 0.9 0.04],...
    'FontSize',11,'FontWeight','bold');

% +X/-X axes
uicontrol('Parent',panel,'Style','pushbutton','String','+X',...
    'Units','normalized','Position',[0.55 0.85 0.18 0.055],...

'BackgroundColor',btn_bg,'ForegroundColor',btn_fg,'FontSize',10,'FontWeight','
bold',...
    'Callback',{@move_cb,[1 0 0 0 0 0]});
uicontrol('Parent',panel,'Style','pushbutton','String','-X',...
    'Units','normalized','Position',[0.27 0.85 0.18 0.055],...

'BackgroundColor',btn_bg,'ForegroundColor',btn_fg,'FontSize',10,'FontWeight','
bold',...
    'Callback',{@move_cb,[-1 0 0 0 0 0]});

% +Y/-Y axes
uicontrol('Parent',panel,'Style','pushbutton','String','+Y',...
    'Units','normalized','Position',[0.55 0.78 0.18 0.055],...

'BackgroundColor',btn_bg,'ForegroundColor',btn_fg,'FontSize',10,'FontWeight','
bold',...
    'Callback',{@move_cb,[0 1 0 0 0 0]});
uicontrol('Parent',panel,'Style','pushbutton','String','-Y',...
    'Units','normalized','Position',[0.27 0.78 0.18 0.055],...

'BackgroundColor',btn_bg,'ForegroundColor',btn_fg,'FontSize',10,'FontWeight','
bold',...
    'Callback',{@move_cb,[0 -1 0 0 0 0]});

% +Z/-Z axes
uicontrol('Parent',panel,'Style','pushbutton','String','+Z',...
    'Units','normalized','Position',[0.55 0.71 0.18 0.055],...
```

```
'BackgroundColor',btn_bg,'ForegroundColor',btn_fg,'FontSize',10,'FontWeight','
bold',...
    'Callback',{@move_cb,[0 0 1 0 0 0]});
uicontrol('Parent',panel,'Style','pushbutton','String','-Z',...
    'Units','normalized','Position',[0.27 0.71 0.18 0.055],...

'BackgroundColor',btn_bg,'ForegroundColor',btn_fg,'FontSize',10,'FontWeight','
bold',...
    'Callback',{@move_cb,[0 0 -1 0 0 0]});

% rotation buttons
uicontrol('Parent',panel,'Style','text','String','ROTATION (30 deg)',...
    'Units','normalized','Position',[0.05 0.62 0.9 0.04],...
    'FontSize',11,'FontWeight','bold');

% +Rx/-Rx
uicontrol('Parent',panel,'Style','pushbutton','String','+Rx',...
    'Units','normalized','Position',[0.55 0.55 0.18 0.055],...

'BackgroundColor',btn_bg,'ForegroundColor',btn_fg,'FontSize',10,'FontWeight','
bold',...
    'Callback',{@move_cb,[0 0 0 1 0 0]});
uicontrol('Parent',panel,'Style','pushbutton','String','-Rx',...
    'Units','normalized','Position',[0.27 0.55 0.18 0.055],...

'BackgroundColor',btn_bg,'ForegroundColor',btn_fg,'FontSize',10,'FontWeight','
bold',...
    'Callback',{@move_cb,[0 0 0 -1 0 0]});

% +Ry/-Ry
uicontrol('Parent',panel,'Style','pushbutton','String','+Ry',...
    'Units','normalized','Position',[0.55 0.48 0.18 0.055],...

'BackgroundColor',btn_bg,'ForegroundColor',btn_fg,'FontSize',10,'FontWeight','
bold',...
    'Callback',{@move_cb,[0 0 0 0 1 0]});
uicontrol('Parent',panel,'Style','pushbutton','String','-Ry',...
    'Units','normalized','Position',[0.27 0.48 0.18 0.055],...

'BackgroundColor',btn_bg,'ForegroundColor',btn_fg,'FontSize',10,'FontWeight','
bold',...
    'Callback',{@move_cb,[0 0 0 0 -1 0]});

% +Rz/-Rz
uicontrol('Parent',panel,'Style','pushbutton','String','+Rz',...
    'Units','normalized','Position',[0.55 0.41 0.18 0.055],...

'BackgroundColor',btn_bg,'ForegroundColor',btn_fg,'FontSize',10,'FontWeight','
bold',...
    'Callback',{@move_cb,[0 0 0 0 0 1]});
uicontrol('Parent',panel,'Style','pushbutton','String','-Rz',...
    'Units','normalized','Position',[0.27 0.41 0.18 0.055],...

'BackgroundColor',btn_bg,'ForegroundColor',btn_fg,'FontSize',10,'FontWeight','
```

```
bold',...
    'Callback',{@move_cb,[0 0 0 0 0 -1]});

% clear buttons
uicontrol('Parent',panel,'Style','pushbutton','String','HOME',...
    'Units','normalized','Position',[0.1 0.28 0.35 0.07],...

'BackgroundColor',btn_bg,'ForegroundColor',btn_fg,'FontSize',10,'FontWeight','
bold',...
    'Callback',@home_cb);

uicontrol('Parent',panel,'Style','pushbutton','String','CLEAR PATH',...
    'Units','normalized','Position',[0.55 0.28 0.35 0.07],...

'BackgroundColor',btn_bg,'ForegroundColor',btn_fg,'FontSize',10,'FontWeight','
bold',...
    'Callback',@clear_cb);

function e = pose_err(Tc, Tt)
    ep = Tt(1:3,4) - Tc(1:3,4);
    Re = Tt(1:3,1:3) * Tc(1:3,1:3)';
    er = rot2aa(Re);
    e = [ep; er];
end

% Draw initial plot
update_plot();
fprintf('UR5 Interactive Controller v2\n');
fprintf('Translation: 100mm increments\n');
fprintf('Rotation: 30 degree increments\n');

function [qs, ok] = newton_ik(Tt, q0, DH)
    q = q0(:);
    ok = false;
    max_iter = 100;
    tol = 1e-5;
    lambda = 0.05;

    for it = 1:max_iter
        [Tc,~] = fk_ur5(q, DH);
        e = pose_err(Tc, Tt);

        if norm(e(1:3)) < tol && norm(e(4:6)) < tol
            ok = true;
            break;
        end

        J = jacobian_ur5(q, DH);
        Jd = J' / (J*J' + lambda^2 * eye(6));
        q = q + Jd * e;
    end

    qs = mod(q + pi, 2*pi) - pi;
end
```

```
% CALLBACK FUNCTIONS
function move_cb(~, ~, dir)
    global DH q_current path_history trans_inc rot_inc status_text

    % Get current pose
    [Tc,~] = fk_ur5(q_current, DH);
    Tn = Tc;

    % Apply translation (in base frame)
    Tn(1:3,4) = Tn(1:3,4) + trans_inc * dir(1:3)';

    % Apply rotation (in base frame)
    if dir(4) ~= 0
        Rd = aa2rot([1;0;0], rot_inc*dir(4));
        Tn(1:3,1:3) = Rd * Tn(1:3,1:3);
    elseif dir(5) ~= 0
        Rd = aa2rot([0;1;0], rot_inc*dir(5));
        Tn(1:3,1:3) = Rd * Tn(1:3,1:3);
    elseif dir(6) ~= 0
        Rd = aa2rot([0;0;1], rot_inc*dir(6));
        Tn(1:3,1:3) = Rd * Tn(1:3,1:3);
    end

    % Solve IK
    [qs, ok] = newton_ik(Tn, q_current, DH);

    if ok
        q_current = qs;
        [Ta,~] = fk_ur5(q_current, DH);
        path_history = [path_history; Ta(1:3,4)'];

        % Update status message
        if any(dir(1:3))
            ax_name = {'X','Y','Z'};
            idx = find(dir(1:3));
            sgn = dir(idx);
            status_text.String = sprintf('Moved %s%s by 100mm',
char((sgn>0)*'+' + (sgn<0)*'-'), ax_name{idx});
        else
            ax_name = {'Rx','Ry','Rz'};
            idx = find(dir(4:6));
            sgn = dir(idx+3);
            status_text.String = sprintf('Rotated %s%s by 30 deg',
char((sgn>0)*'+' + (sgn<0)*'-'), ax_name{idx});
        end
        status_text.BackgroundColor = [0.8 1 0.8];
    else
        status_text.String = 'IK FAILED - Target unreachable';
        status_text.BackgroundColor = [1 0.8 0.8];
    end

    update_plot();
end
```

```
function home_cb(~, ~)
    global DH q_current path_history status_text

    % Return to starting position (arm toward +X)
    q_current = [pi; 0; 0; 0; pi; 0];
    [T,~] = fk_ur5(q_current, DH);
    path_history = [path_history; T(1:3,4)'];
    status_text.String = 'Returned to HOME (+X direction)';
    status_text.BackgroundColor = [0.8 0.8 1];
    update_plot();
end

function clear_cb(~, ~)
    global DH q_current path_history status_text

    [T,~] = fk_ur5(q_current, DH);
    path_history = T(1:3,4)';
    status_text.String = 'Path cleared';
    status_text.BackgroundColor = [0.9 0.9 0.9];
    update_plot();
end

function update_plot()
    global DH q_current path_history ax pos_text

    cla(ax);
    hold(ax,'on');
    grid(ax,'on');
    axis(ax,'equal');

    % get fk
    [Te, Ta] = fk_ur5(q_current, DH);

    % get joint position
    jp = zeros(7,3);
    jp(1,:) = [0 0 0];  % Base
    for k = 1:6
        jp(k+1,:) = Ta(1:3,4,k+1)';
    end

    % robot links (thick blue line)
    plot3(ax, jp(:,1), jp(:,2), jp(:,3), 'b-', 'LineWidth', 5);

    % joints (black circles)
    plot3(ax, jp(:,1), jp(:,2), jp(:,3), 'ko', 'MarkerSize', 14,
'MarkerFaceColor', [0.2 0.2 0.8]);

    % base (gray square)
    plot3(ax, 0, 0, 0, 'ks', 'MarkerSize', 22, 'MarkerFaceColor', [0.4 0.4
0.4]);

    % end-effector frame (RGB = XYZ)
    p = Te(1:3,4);
    s = 0.12;  % Arrow length

    % EE X-axis (red)
    quiver3(ax, p(1), p(2), p(3), Te(1,1)*s, Te(2,1)*s, Te(3,1)*s, 'r',
'LineWidth', 3, 'MaxHeadSize', 0.8);
    % EE Y-axis (green)
    quiver3(ax, p(1), p(2), p(3), Te(1,2)*s, Te(2,2)*s, Te(3,2)*s, 'g',
'LineWidth', 3, 'MaxHeadSize', 0.8);
    % EE Z-axis (blue)
    quiver3(ax, p(1), p(2), p(3), Te(1,3)*s, Te(2,3)*s, Te(3,3)*s, 'b',
'LineWidth', 3, 'MaxHeadSize', 0.8);

    % Draw path history (magenta)
    if size(path_history,1) > 1
        plot3(ax, path_history(:,1), path_history(:,2), path_history(:,3),
'm-', 'LineWidth', 2);
        plot3(ax, path_history(:,1), path_history(:,2), path_history(:,3),
'm.', 'MarkerSize', 10);
    end

    % Draw base coordinate frame
    quiver3(ax, 0, 0, 0, 0.2, 0, 0, 'r-', 'LineWidth', 2);
    quiver3(ax, 0, 0, 0, 0, 0.2, 0, 'g-', 'LineWidth', 2);
    quiver3(ax, 0, 0, 0, 0, 0, 0.2, 'b-', 'LineWidth', 2);
    text(ax, 0.23, 0, 0, 'X', 'FontSize', 12, 'Color', 'r', 'FontWeight',
'bold');
    text(ax, 0, 0.23, 0, 'Y', 'FontSize', 12, 'Color', 'g', 'FontWeight',
'bold');
    text(ax, 0, 0, 0.23, 'Z', 'FontSize', 12, 'Color', 'b', 'FontWeight',
'bold');

    % Labels and title
    xlabel(ax, 'X (m)', 'FontSize', 11);
    ylabel(ax, 'Y (m)', 'FontSize', 11);
    zlabel(ax, 'Z (m)', 'FontSize', 11);
    title(ax, 'UR5 Interactive Control', 'FontSize', 13, 'FontWeight',
'bold');

    % Set axis limits to keep robot in positive quadrant visible
    xlim(ax, [-0.3 1.0]);
    ylim(ax, [-0.5 0.5]);
    zlim(ax, [-0.2 0.8]);
    view(ax, 45, 25);

    % Update position text display
    pos_text.String = sprintf(['Position: [%.3f, %.3f, %.3f] m\n' ...
                               'Joints (deg):\n' ...
                               ' [%.1f, %.1f, %.1f, %.1f, %.1f, %.1f]\n' ...
                               'Path points: %d'], ...
        p(1), p(2), p(3), rad2deg(q_current'), size(path_history,1));

    drawnow;
end
```

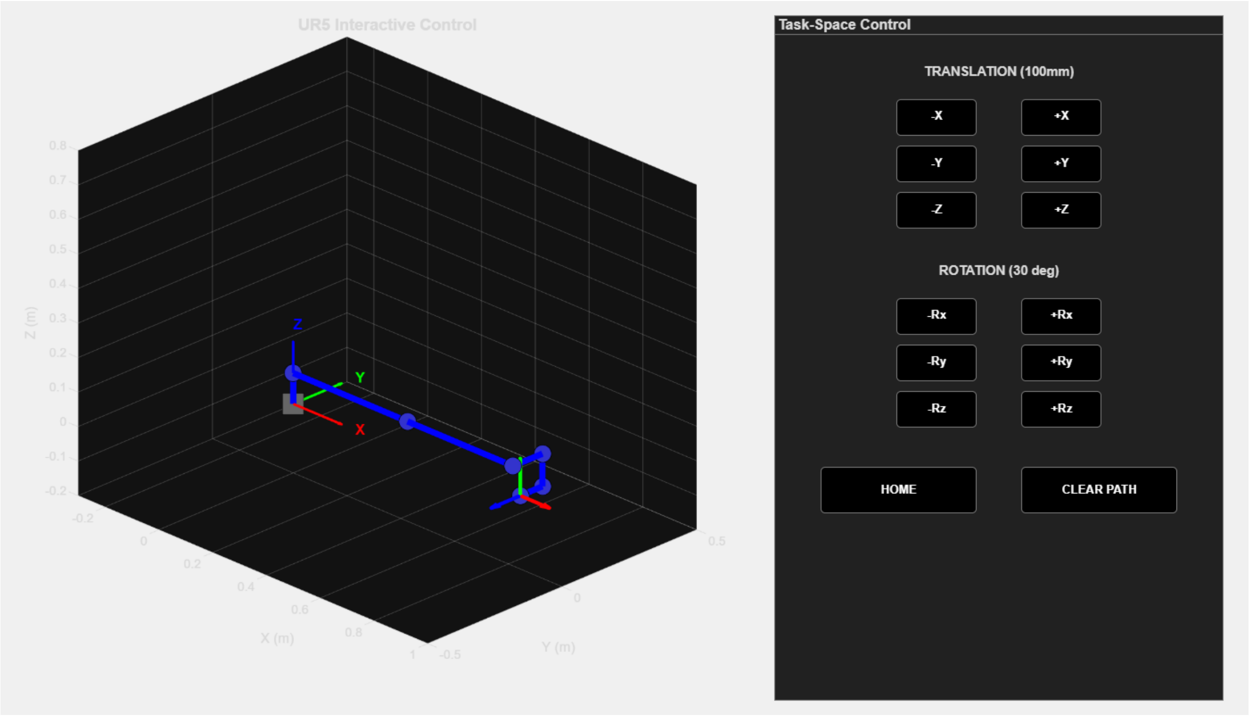
```
%  KINEMATICS FUNCTIONS
function J = jacobian_ur5(q, DH)
    [~, Ta] = fk_ur5(q, DH);
    pe = Ta(1:3, 4, end);
    J = zeros(6, DH.n);
    for i = 1:DH.n
        Ti = Ta(:,:,i);
        zi = Ti(1:3, 3);
        pi = Ti(1:3, 4);
        J(1:3, i) = cross(zi, pe - pi);
        J(4:6, i) = zi;
    end
end

function r = rot2aa(R)
    th = acos(max(-1, min(1, (trace(R)-1)/2)));
    if abs(th) < 1e-10
        r = [0; 0; 0];
    elseif abs(th - pi) < 1e-10
        [~, idx] = max(diag(R));
        v = R(:,idx) + [idx==1; idx==2; idx==3];
        r = th * v / norm(v);
    else
        r = th / (2*sin(th)) * [R(3,2)-R(2,3); R(1,3)-R(3,1); R(2,1)-R(1,2)];
    end
end

function R = aa2rot(ax, ang)
    ax = ax / norm(ax);
    K = [0, -ax(3), ax(2); ax(3), 0, -ax(1); -ax(2), ax(1), 0];
    R = eye(3) + sin(ang)*K + (1-cos(ang))*(K*K);
end
```

```
Starting position: [0.8173, 0.0268, -0.0055] m
Starting orientation (R matrix):
    1.0000   -0.0000    0.0000
         0   -0.0000   -1.0000
    0.0000    1.0000   -0.0000

UR5 Interactive Controller v2
Translation: 100mm increments
Rotation: 30 degree increments
```



Published with MATLAB® R2025b