哈爾濱工業大學(深圳)
HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

《机器人学导论》
实验(三)报告

仅供个人学习参考，请勿抄袭或用于商业用途！

专业： 自动化

班级： 2103206 班

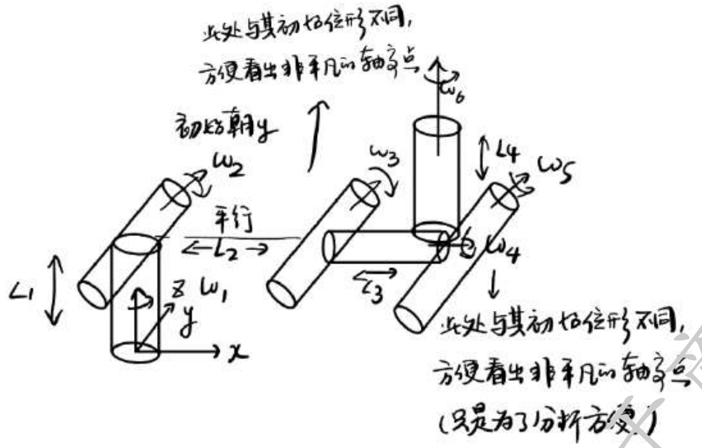
姓名： 吴俊达

学号： 210320621

2024 年 5 月

一、 六轴机器人逆运动学分析

绘制出其结构简图：



可见，4、5、6 轴一定交于一点，2、3 轴一定平行，3、4 轴一定有交点，1、2 轴也一定有交点。

逆运动学解法应为：

由指数积公式，$g_{st}(\theta)=e^{\xi_1\theta_1}e^{\xi_2\theta_2}e^{\xi_3\theta_3}e^{\xi_4\theta_4}e^{\xi_5\theta_5}e^{\xi_6\theta_6}g_{st}(0)$（这里将工具坐标系建在腕点上）。令

$e^{\xi_1\theta_1}e^{\xi_2\theta_2}e^{\xi_3\theta_3}e^{\xi_4\theta_4}e^{\xi_5\theta_5}e^{\xi_6\theta_6}=g_{st}(\theta)g^{-1}(0):=g_d$。

第一步： 设 4、5、6 轴交点为 p_w，则 $e^{\xi_4\theta_4}e^{\xi_5\theta_5}e^{\xi_6\theta_6}p_w=p_w$，代入上式得 $e^{\xi_1\theta_1}e^{\xi_2\theta_2}e^{\xi_3\theta_3}p_w=g_d p_w$

再取 1、2 轴交点 q_w，可知 $e^{\xi_1\theta_1}e^{\xi_2\theta_2}q_w=q_w$，

因此 $e^{\xi_1\theta_1}e^{\xi_2\theta_2}e^{\xi_3\theta_3}p_w-q_w=e^{\xi_1\theta_1}e^{\xi_2\theta_2}(e^{\xi_3\theta_3}p_w-q_w)=g_d p_w-q_w$，

两边取模，得 $\left\|e^{\xi_3\theta_3}p_w-q_w\right\|=\left\|g_d p_w-q_w\right\|:=\delta$，利用子问题 3 可以求解 θ_3；

第二步： 由 $e^{\xi_1\theta_1}e^{\xi_2\theta_2}e^{\xi_3\theta_3}p_w=g_d p_w:=q$，则 $e^{\xi_2\theta_2}e^{\xi_3\theta_3}p_w=e^{-\xi_1\theta_1}q$，可知 $e^{\xi_2\theta_2}e^{\xi_3\theta_3}p_w$ 必在 xOz 平面内运动，所以 $\begin{bmatrix}0 & 1 & 0\end{bmatrix}^T e^{-\xi_1\theta_1}q=0$，则 $-q_x\sin\theta_1+q_y\cos\theta_1=0$，因此 $\theta_1=\operatorname{atan2}\left(\frac{q_y}{q_x}\right)$。据此求出 θ_1；（这里需要考虑±π，再代入下面的求解过程）

第三步： 由 $e^{\xi_2\theta_2}e^{\xi_3\theta_3}p_w=e^{-\xi_1\theta_1}q$，利用子问题 1 可以求解 θ_2；

上面两步也可以用这一步来代替：将解得的 θ_3 代回 $e^{\xi_1\theta_1}e^{\xi_2\theta_2}e^{\xi_3\theta_3}p_w=g_d p_w$，可得

$e^{\xi_1\theta_1}e^{\xi_2\theta_2}(e^{\xi_3\theta_3}p_w):=e^{\xi_1\theta_1}e^{\xi_2\theta_2}p_{w2}=g_d p_w$，利用子问题 2 可以求解 θ_1,θ_2。

第四步： 现在假定 $e^{\xi_4\theta_4}e^{\xi_5\theta_5}e^{\xi_6\theta_6}=(e^{\xi_1\theta_1}e^{\xi_2\theta_2}e^{\xi_3\theta_3})^{-1}g_{st}(\theta)g^{-1}(0):=g_{d2}$，将其作用于 6 轴上一点 p_{w2}（但不在 5 轴上），即得 $e^{\xi_4\theta_4}e^{\xi_5\theta_5}p_{w2}=g_{d2}p_{w2}$，利用子问题 2 可以求解 θ_4,θ_5。

第五步： 于是 $e^{\xi_6\theta_6}=(e^{\xi_4\theta_4}e^{\xi_5\theta_5})^{-1}(e^{\xi_1\theta_1}e^{\xi_2\theta_2}e^{\xi_3\theta_3})^{-1}g_{st}(\theta)g^{-1}(0):=g_{d3}$，将其作用于 6 轴外一点 p_{w3}，即得 $e^{\xi_6\theta_6}p_{w3}=g_{d3}p_{w3}$，利用子问题 1 可以求解 θ_6。

上述方框中，上面一格的方法应用于后文的 Ikine6s_1.m 中，下面一格的方法则应用于 Ikine6s_2.m 中。

仅供个人学习参考，请勿抄袭或用于商业用途!

二、 MATLAB 代码

求解子问题 1 的函数 subproblem1：

```
function solution = subproblem1(w,r,p,q)
    u = p - r; % 取轴线上一点 r 并构造两向量
    v = q - r;
    u_proj = u - w*w'*u; % 构造投影向量
    v_proj = v - w*w'*v;
    if u_proj == 0
        disp('[子问题 1]无穷多解');
    end
    a = w'*(cross(u_proj,v_proj)); % 此项出现角度正弦
    b = u_proj'*v_proj; % 此项出现角度余弦
    solution = atan2d(a,b); % 求出正切
end
```

求解子问题 2 的函数 subproblem2：

```
function solution = subproblem2(w1,w2,r,p,q)
    u = p - r; % 取两轴交点并构造两向量
    v = q - r;
    alpha = (dot(w1,w2)*dot(w2,u) - dot(w1,v))/(dot(w1,w2)^2 - 1);
    beta = (dot(w1,w2)*dot(w1,v) - dot(w2,u))/(dot(w1,w2)^2 - 1);
    gamma_sq = (dot(u,u) - alpha^2 - beta^2 -
2*alpha*beta*dot(w1,w2))/norm(cross(w1,w2))^2; % 此为 gamma 的平方
    if gamma_sq < 0
        solution = [];
        disp('[子问题 2] 无解');
        return
    end
    gamma1 = sqrt(gamma_sq);
    gamma2 = -gamma1;
    z1 = alpha*w1 + beta*w2 + gamma1*cross(w1,w2);
    z2 = alpha*w1 + beta*w2 + gamma2*cross(w1,w2);

    if all(z1 == z2)
        c1 = z1 + r;
        theta1 = subproblem1(-w1,r,q,c1);
        theta2 = subproblem1(w2,r,p,c1);

        disp('[子问题 2] 存在唯一解')
        solution = [theta1; theta2];

    else
        c1 = z1 + r;
```

仅供个人学习参考，请勿抄袭或用于商业用途！

```
        theta11 = subproblem1(-w1,r,q,c1);
        theta21 = subproblem1(w2,r,p,c1);

        c2 = z2 + r;
        theta12 = subproblem1(-w1,r,q,c2);
        theta22 = subproblem1(w2,r,p,c2);
        disp('[子问题 2] 存在两组解')
        solution = [theta11 theta12; theta21 theta22];
    end
end
```

求解子问题 3 的函数 subproblem3：

```
function solution = subproblem3(w, r, p, q, delta)
    u = p - r;
    v = q - r;

    u_proj = u - w*w'*u;
    v_proj = v - w*w'*v;
    dp_sq = delta^2 - abs(dot(w,p-q))^2;
    value1 = dot(w,cross(u_proj,v_proj));
    value2 = dot(u_proj,v_proj);
    value3 = (dot(u_proj,u_proj) + dot(v_proj,v_proj) -
dp_sq)/(2*norm(u_proj)*norm(v_proj));
    theta0 = atan2d(value1,value2);
    if value3 == 1
        disp('[子问题 3] 存在唯一解')
        solution = theta0;
        return;
    elseif abs(value3) > 1
        disp('[子问题 3] 无解')
        solution = [];
        return;
    end
    theta1 = theta0 + acosd(value3);
    theta2 = theta0 - acosd(value3);
    disp('[子问题 3] 存在两组解')
    solution = [theta1 theta2];
end
```

逆运动学第一种解法 Ikine6s_1.m，对应第一部分的分析中的方框中解法二：

```
% AUTHOR: Wu Junda
%
% ABSTRACT: inverse kinematics (for 6-axis robot)
```

仅供个人学习参考，请勿抄袭或用于商业用途！

```
%
% INPUT: g0       offset, 4X4 matrix
%       g_st      end-effector config., 4X4 matrix
%        config   1X22 list, 1-4: length of links;
%                             5-10: a point from each axis
%                             11-16: vector along each axis
%                             17-22: range of displacement of each axis
%
% OUTPUT: theta     displacement of joints, 1xN vector,
%                   unit: (for this problem) deg
%

function theta = Ikine6s_1(g0,g_st,config)
    L1 = config{1}; L2 = config{2}; L3 = config{3}; L4 = config{4};
    q1 = config{5}; q2 = config{6}; q3 = config{7}; q4 = config{8};
    q5 = config{9}; q6 = config{10};
    w1 = config{11}; w2 = config{12}; w3 = config{13}; w4 = config{14};
    w5 = config{15}; w6 = config{16};
    theta1_range = config{17}; theta2_range = config{18};
    theta3_range = config{19}; theta4_range = config{20};
    theta5_range = config{21}; theta6_range = config{22};

    xi1 = [-cross(w1,q1); w1];
    xi2 = [-cross(w2,q2); w2];
    xi3 = [-cross(w3,q3); w3];
    xi4 = [-cross(w4,q4); w4];
    xi5 = [-cross(w5,q5); w5];
    xi6 = [-cross(w6,q6); w6];

    gd = g_st/g0;
    pw = [0 0 L1+L2+L3]'; % wrist center
    pb = [0 0 L1]'; % base center

    theta = [];

    disp("---------Start to solve IK problem---------");
    disp("initial position: ");
    disp(g0);
    disp("target position: ");
    disp(g_st);

    % STEP1: solve for theta3
    d = gd*[pw; 1] - [pb; 1];
    delta = norm(d);
```

仅供大家学习参考，请勿抄袭或用于商业用途！

```
solution1 = subproblem3(w3, q3, pw, pb, delta);
solution1 = [solution1, solution1+360, solution1-360];

for i = 1:length(solution1)
    theta3 = solution1(i);
    if theta3 < theta3_range(1) || theta3 > theta3_range(2)
        continue
    end

    g3 = Transformation(xi3, theta3);
    g3_inv = inv(g3);

    % STEP2: solve for theta1
    q = gd*[pw; 1];
    if atan2d(q(2),q(1))<=0
        solution2 = [atan2d(q(2),q(1)), atan2d(q(2),q(1))+180];
    else
        solution2 = [atan2d(q(2),q(1)), atan2d(q(2),q(1))-180];
    end

    for j = 1:size(solution2,2)
        theta1 = solution2(1,j);
        if theta1 < theta1_range(1) || theta1 > theta1_range(2)
            continue
        end
        g1 = Transformation(xi1, theta1);
        g1_inv = inv(g1);

        % STEP3: solve for theta2
        pw1 = g3*[pw; 1];
        qw1 = g1_inv * q;
        solution3 = subproblem1(w2,q2,pw1(1:3),qw1(1:3));
        for n = 1:size(solution3,2)
            theta2 = solution3(1,n);
            if theta2 < theta2_range(1) || theta2 > theta2_range(2)
                continue
            end
            g2 = Transformation(xi2, theta2);
            g2_inv = inv(g2);

            % STEP4: solve for theta4, theta5
            gd2 = g3_inv * g2_inv * g1_inv* gd;
            pw2 = [0;0;L1+L2+L3+L4];
            qw2 = gd2 * [pw2;1];
```

仅供个人学习参考，请勿抄袭或用于商业用途！

```
                solution4 = subproblem2(w4,w5,q4,pw2,qw2(1:3));
                solution4 = [solution4, solution4+360, solution4-360];
                for k = 1:size(solution4,2)
                    theta4 = solution4(1,k);
                    if theta4 < theta4_range(1) || theta4 > theta4_range(2)
                        continue
                    end
                    g4 = Transformation(xi4, theta4);
                    g4_inv = inv(g4);
                    theta5 = solution4(2,k);
                    if theta5 < theta5_range(1) || theta5 > theta5_range(2)
                        continue
                    end
                    g5 = Transformation(xi5, theta5);
                    g5_inv = inv(g5);
                    % STEP5: solve for theta6
                    gd3 = g5_inv* g4_inv* g3_inv* g2_inv* g1_inv* gd;
                    pw3 = [0;50;L1+L2+L3+L4];
                    qw3 = gd3 * [pw3;1];
                    solution5 = subproblem1(w6,q4,pw3,qw3(1:3));
                    for m = 1:size(solution5,2)
                        theta6 = solution5(1,m);
                        if theta6 < theta6_range(1) || theta6 > theta6_range(2)
                            continue
                        end
                        theta = [theta vpa([theta1;theta2;theta3;theta4;theta5;theta6])];
                    end
                end
            end
        end
    end
    disp("---------Method 1 ends---------");
end
```

逆运动学第二种解法 Ikine6s_2.m，对应第一部分的分析中的方框中解法二：

```
% AUTHOR : Wu Junda
%
% ABSTRACT: inverse kinematics (for 6-axis robot)
%
% INPUT: g0       offset, 4X4 matrix
%       g_st      end-effector config., 4X4 matrix
%        config   1X22 list, 1-4: length of links;
%                             5-10: a point from each axis
%                             11-16: vector along each axis
```

仅供大学习参考，请勿抄袭或用于商业用途！

```
%                              17-22: range of displacement of each axis
%
% OUTPUT: theta      displacement of joints, 1xN vector,
%                    unit: (for this problem) deg
%

function theta = Ikine6s_2(g0,g_st,config)
    L1 = config{1}; L2 = config{2}; L3 = config{3}; L4 = config{4};
    q1 = config{5}; q2 = config{6}; q3 = config{7}; q4 = config{8};
    q5 = config{9}; q6 = config{10};
    w1 = config{11}; w2 = config{12}; w3 = config{13}; w4 = config{14};
    w5 = config{15}; w6 = config{16};
    theta1_range = config{17}; theta2_range = config{18};
    theta3_range = config{19}; theta4_range = config{20};
    theta5_range = config{21}; theta6_range = config{22};

    xi1 = [-cross(w1,q1); w1];
    xi2 = [-cross(w2,q2); w2];
    xi3 = [-cross(w3,q3); w3];
    xi4 = [-cross(w4,q4); w4];
    xi5 = [-cross(w5,q5); w5];
    xi6 = [-cross(w6,q6); w6];

    gd = g_st/g0;
    pw = [0 0 L1+L2+L3]'; % wrist center
    pb = [0 0 L1]'; % base center

    theta = [];

    disp("---------Start to solve IK problem---------");
    disp("initial position: ");
    disp(g0);
    disp("target position: ");
    disp(g_st);

    % STEP1: solve for theta3
    d = gd*[pw; 1] - [pb; 1];
    delta = norm(d);
    solution1 = subproblem3(w3, q3, pw, pb, delta);
    solution1 = [solution1, solution1+360, solution1-360];

    q = gd*[pw; 1];
    % q = q(1:3);
    for i = 1:length(solution1)
```

仅供大学分签考，请勿抄袭或用于商业用途！

```
theta3 = solution1(i);
if theta3 < theta3_range(1) || theta3 > theta3_range(2)
    continue
end

g3 = Transformation(xi3, theta3);
g3_inv = inv(g3);

p = g3*[pw; 1];
% STEP2: solve for theta1, theta2
solution2 = subproblem2(w1,w2,q2,p(1:3),q(1:3));
if isempty(solution2)
    continue
end
solution2 = [solution2, solution2+360, solution2-360];

for j = 1:size(solution2,2)
    theta1 = solution2(1,j);
    if theta1 < theta1_range(1) || theta1 > theta1_range(2)
        continue
    end

    g1 = Transformation(xi1, theta1);
    g1_inv = inv(g1);

    theta2 = solution2(2,j);
    if theta2 < theta2_range(1) || theta2 > theta2_range(2)
        continue
    end
    g2 = Transformation(xi2, theta2);
    g2_inv = inv(g2);

    % STEP3: solve for theta4, theta5
    gd2 = g3_inv * g2_inv * g1_inv* gd;
    pw2 = [0;0;L1+L2+L3+L4];
    qw2 = gd2 * [pw2;1];
    solution3 = subproblem2(w4,w5,q4,pw2,qw2(1:3));
    solution3 = [solution3, solution3+360, solution3-360];
    for k = 1:size(solution3,2)
        theta4 = solution3(1,k);
        if theta4 < theta4_range(1) || theta4 > theta4_range(2)
            continue
        end
        g4 = Transformation(xi4, theta4);
```

仅供个人学习参考，请勿抄袭或用于商业用途!

```
                g4_inv = inv(g4);
                theta5 = solution3(2,k);
                if theta5 < theta5_range(1) || theta5 > theta5_range(2)
                    continue
                end
                g5 = Transformation(xi5, theta5);
                g5_inv = inv(g5);
                % STEP4: solve for theta6
                gd3 = g5_inv* g4_inv* g3_inv* g2_inv* g1_inv* gd;
                pw3 = [0;50;L1+L2+L3+L4];
                qw3 = gd3 * [pw3;1];
                solution4 = subproblem1(w6,q4,pw3,qw3(1:3));
                for m = 1:size(solution4,2)
                    theta6 = solution4(1,m);
                    if theta6 < theta6_range(1) || theta6 > theta6_range(2)
                        continue
                    end
                    theta = [theta vpa([theta1;theta2;theta3;theta4;theta5;theta6])];
                end
            end
        end
    end
    disp("---------Method 2 ends---------");
end
```

利用下面的代码，先取一角度计算姿态矩阵，再据此逆解出关节角，最后根据关节角再正解一次，根据其与给定姿态矩阵之差来判断其是否求解正确。其中正解用到的各函数定义在上个实验报告中已经说明，此处不再赘述。

```
q1 = [0;0;0];
q2 = [0;0;491];
q3 = [0;0;450+491];
q4 = [0;0;900+491];
q5 = [0;0;900+491];
q6 = [0;0;900+491];

w1 = [0;0;1];
w2 = [0;1;0];
w3 = [0;1;0];
w4 = [0;0;1];
w5 = [0;1;0];
w6 = [0;0;1];

xi1 = [-cross(w1,q1); w1];
xi2 = [-cross(w2,q2); w2];
```

仅供个人学习参考 请勿抄袭或用于商业用途!

```
xi3 = [-cross(w3,q3); w3];
xi4 = [-cross(w4,q4); w4];
xi5 = [-cross(w5,q5); w5];
xi6 = [-cross(w6,q6); w6];
xi = [xi1 xi2 xi3 xi4 xi5 xi6];

theta1 = 30;
theta2 = 40;
theta3 = -50;
theta4 = 60;
theta5 = -10;
theta6 = 99;
theta = [theta1 theta2 theta3 theta4 theta5 theta6];

g_init = [-1,0,0,0;
          0,-1,0,0;
          0,0,1,1475;
          0,0,0,1];

% forward
gd = Fkine(xi,theta,g_init);
gd = vpa(gd);
% inverse
config = {491,450,450,84,...
    q1,q2,q3,q4,q5,q6,...
    w1,w2,w3,w4,w5,w6,...
    [-170,170],[-120,120],[-140,140],[-170,170],[-120,120],[-360,360]};

% algorithm 1
all_solutions_1 = Ikine6s_1(g_init,gd,config);
% algorithm 2
all_solutions_2 = Ikine6s_2(g_init,gd,config);

verified_solutions_1 = [];
for i=1:size(all_solutions_1,2)
    theta_computed = all_solutions_1(:,i);
    gd_calc = Fkine(xi,theta_computed',g_init);
    diff = gd_calc - gd;
    if (any(abs(diff) > 1e-9,'all'))
        continue
    else
        verified_solutions_1 = [verified_solutions_1 theta_computed];
    end
end
```

仅供大学习参考，请勿抄袭或用于商业用途！

```
verified_solutions_2 = [];
for i=1:size(all_solutions_2,2)
    theta_computed = all_solutions_2(:,i);
    gd_calc = Fkine(xi,theta_computed',g_init);
    diff = gd_calc - gd;
    if (any(abs(diff) > 1e-9,'all'))
        continue
    else
        verified_solutions_2 = [verified_solutions_2 theta_computed];
    end
end

disp(vpa(verified_solutions_1',5)) % each row stands for a solution
verified_solutions_1 = verified_solutions_1 .* (pi/180); % used for simulation
disp(vpa(verified_solutions_2',5)) % each column stands for a solution
verified_solutions_2 = verified_solutions_2 .* (pi/180); % used for simulation
```

输出结果（角度制）

（第一种方法）

```
[   30.0, -10.0,  50.0,  10.515, -55.493,  152.62]
[   30.0, -10.0,  50.0, -169.49,  55.493, -27.384]
[ -150.0, -40.0,  50.0, -120.0,   -10.0,    99.0]
[ -150.0, -40.0,  50.0,    60.0,   10.0,   -81.0]
[   30.0,  40.0, -50.0,    60.0,  -10.0,    99.0]
[   30.0,  40.0, -50.0, -120.0,    10.0,   -81.0]
[ -150.0,  10.0, -50.0, -169.49, -55.493,  152.62]
[ -150.0,  10.0, -50.0,  10.515,  55.493, -27.384]
```

（第二种方法）

```
[ -150.0, -40.0,  50.0, -120.0,   -10.0,    99.0]
[ -150.0, -40.0,  50.0,    60.0,   10.0,   -81.0]
[   30.0, -10.0,  50.0,  10.515, -55.493,  152.62]
[   30.0, -10.0,  50.0, -169.49,  55.493, -27.384]
[ -150.0,  10.0, -50.0, -169.49, -55.493,  152.62]
[ -150.0,  10.0, -50.0,  10.515,  55.493, -27.384]
[   30.0,  40.0, -50.0,    60.0,  -10.0,    99.0]
[   30.0,  40.0, -50.0, -120.0,    10.0,   -81.0]
```

可见，两种方法所得结果完全相同。将其转化为弧度制：

```
[ 0.5236, -0.17453,  0.87266, 0.18352, -0.96853,   2.6637]
[ 0.5236, -0.17453,  0.87266, -2.9581,  0.96853, -0.47794]
[ -2.618, -0.69813,  0.87266, -2.0944, -0.17453,   1.7279]
[ -2.618, -0.69813,  0.87266,  1.0472,  0.17453,  -1.4137]
[ 0.5236,  0.69813, -0.87266,  1.0472, -0.17453,   1.7279]
[ 0.5236,  0.69813, -0.87266, -2.0944,  0.17453,  -1.4137]
[ -2.618,  0.17453, -0.87266, -2.9581, -0.96853,   2.6637]
[ -2.618,  0.17453, -0.87266, 0.18352,  0.96853, -0.47794]
```

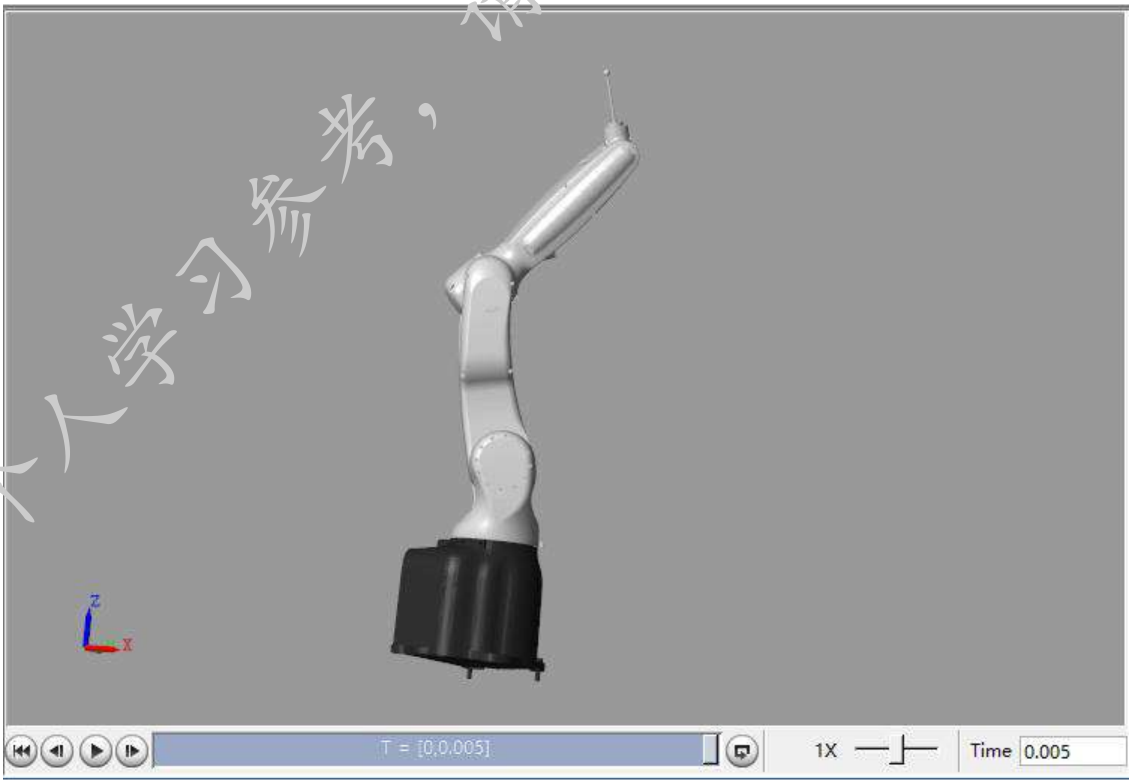
仅供个人学习参考，请勿抄袭或用于商业用途！

再输入仿真模型，得到模型图片如下一部分所示：

三、 所有 8 组解模型截图

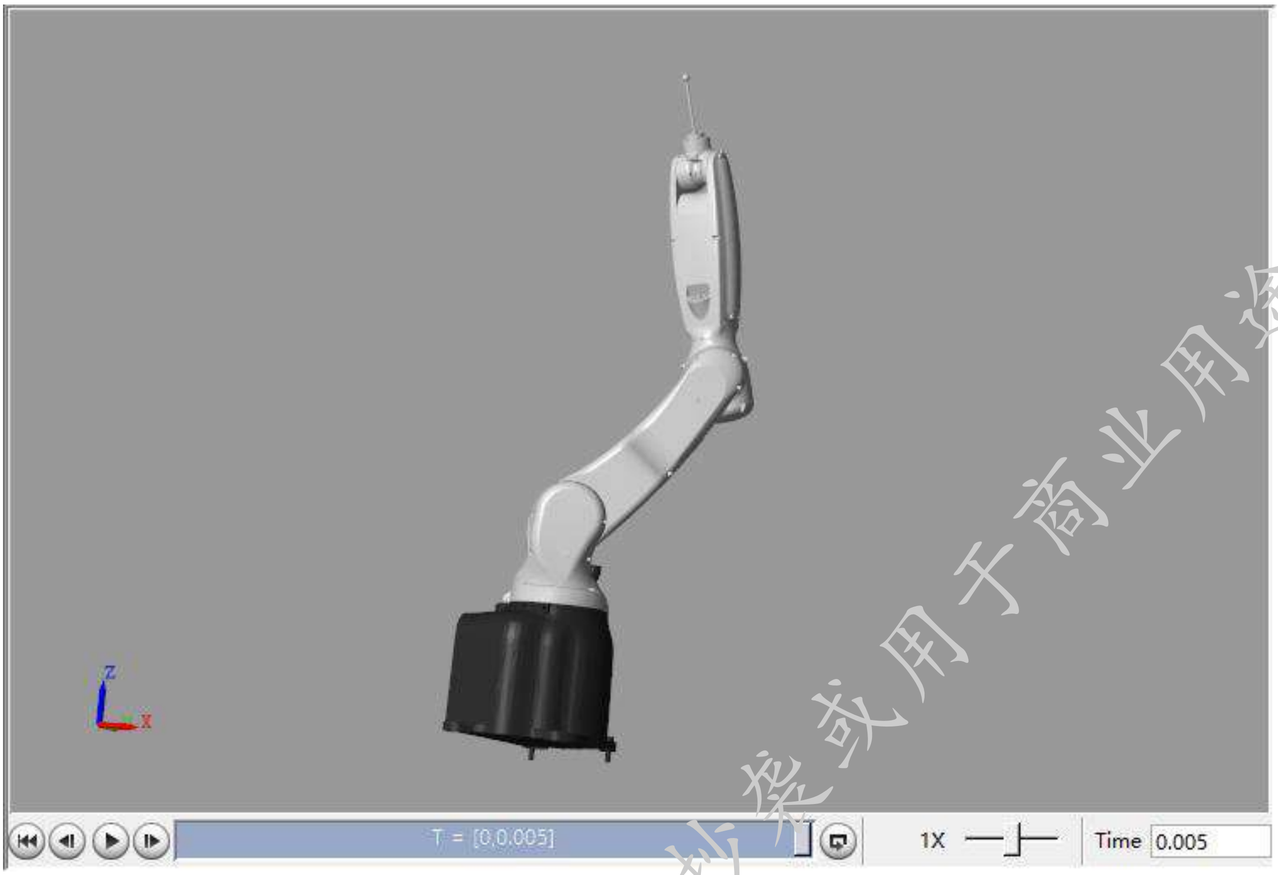
按照上面 8 组解的顺序将各解输入仿真模型：

第 1 组解：[0.5236, -0.17453, 0.87266, 0.18352, -0.96853, 2.6637]

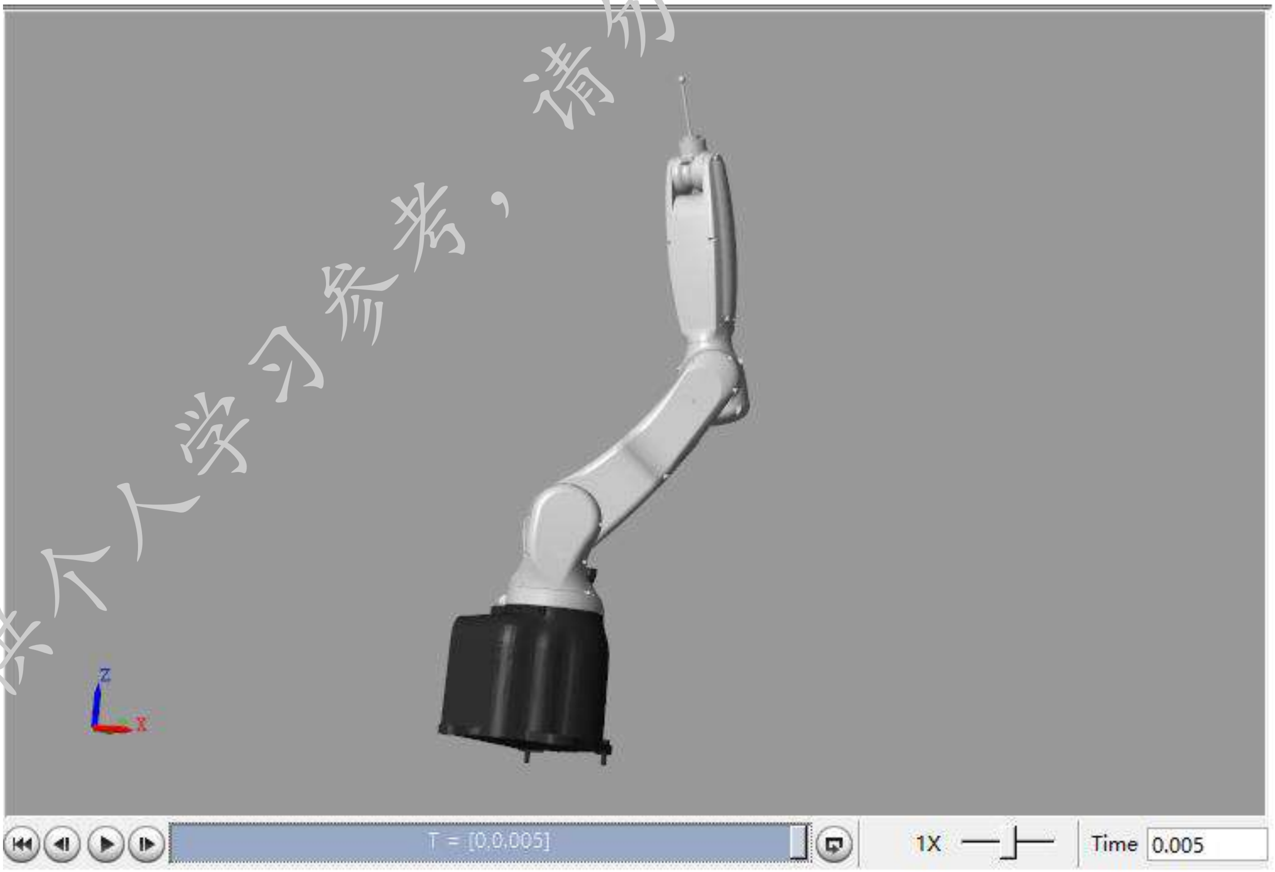


第 2 组解：[0.5236, -0.17453, 0.87266, -2.9581, 0.96853, -0.47794]

仅供个人学习参考，请勿抄袭或用于商业用途！



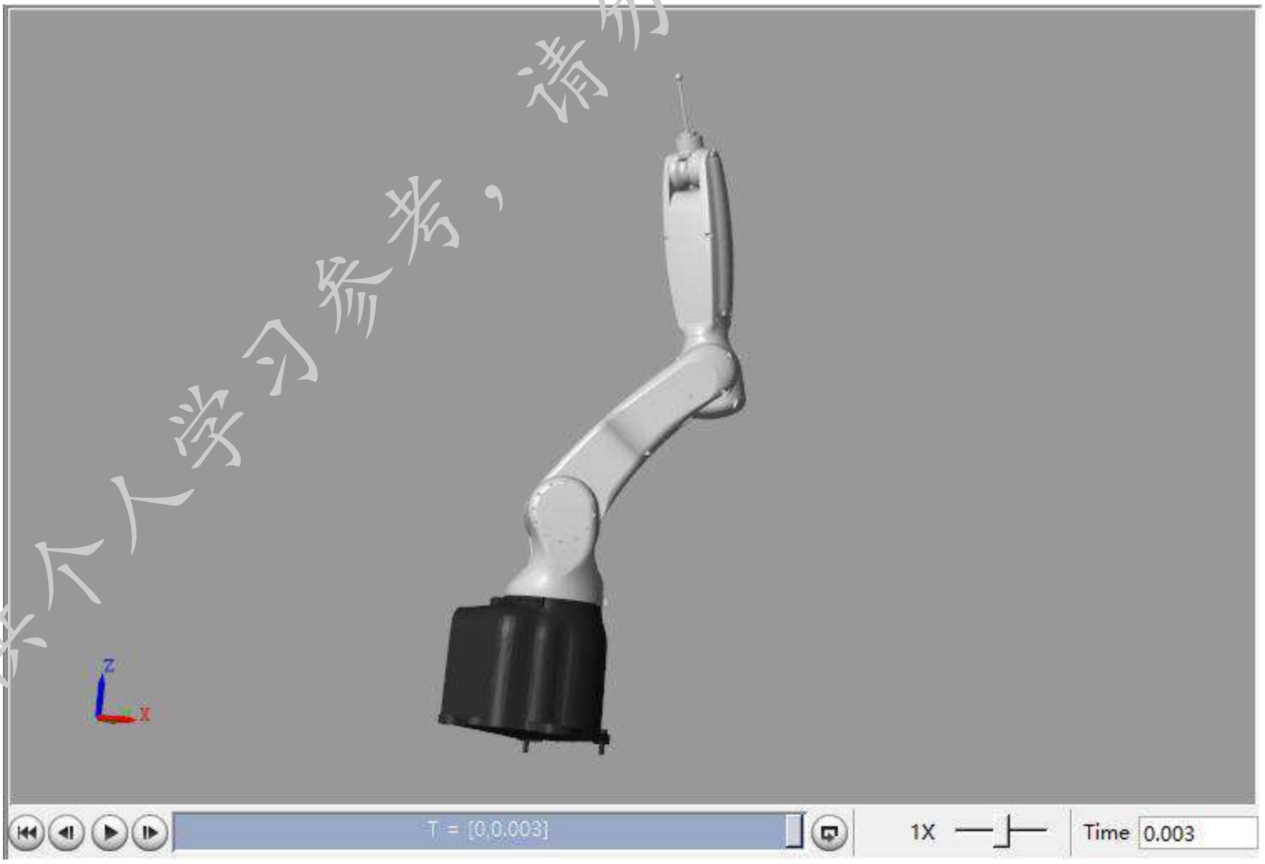
第 3 组解：[-2.618, -0.69813, 0.87266, -2.0944, -0.17453, 1.7279]



第 4 组解：[-2.618, -0.69813, 0.87266, 1.0472, 0.17453, -1.4137]

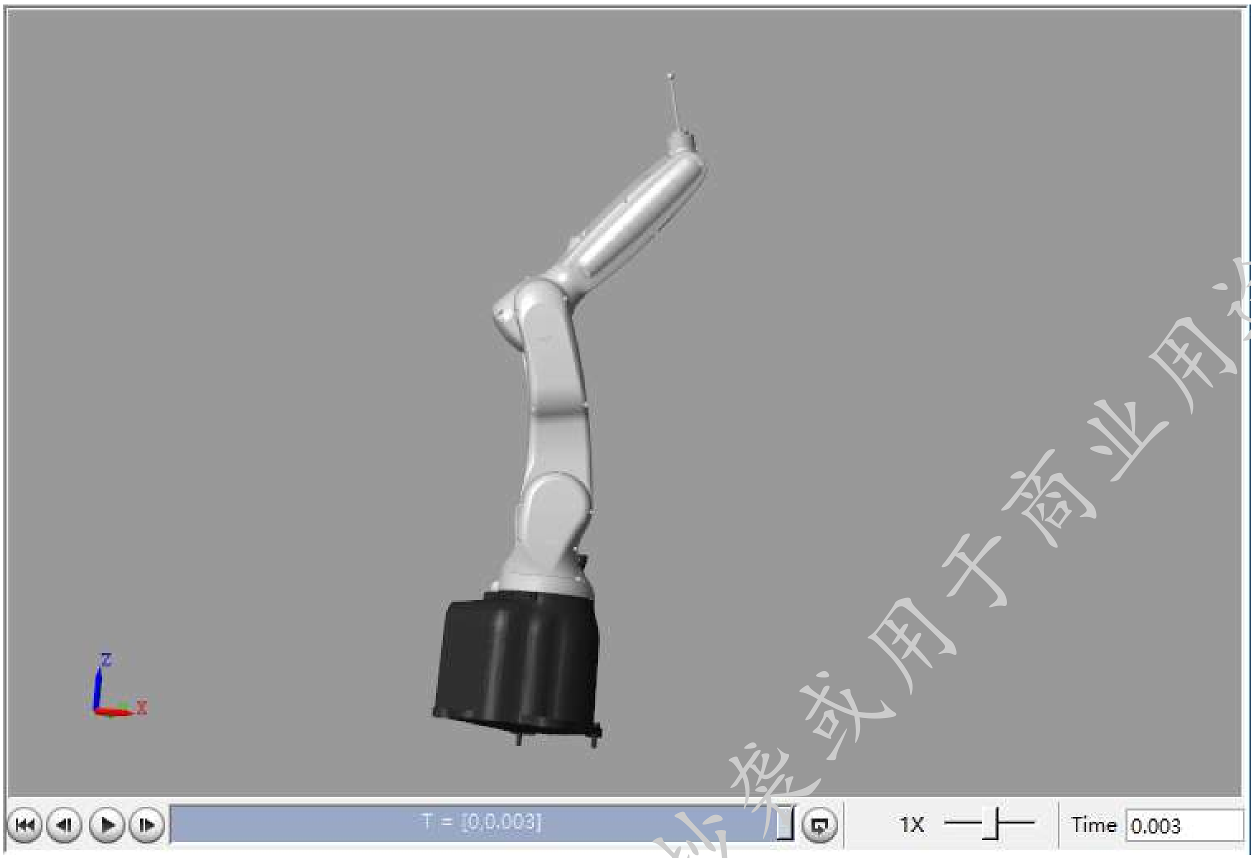
仅供个人学习参考，请勿抄袭或用于商业用途！

第 5 组解：[0.5236,   0.69813, -0.87266,   1.0472, -0.17453,   1.7279]

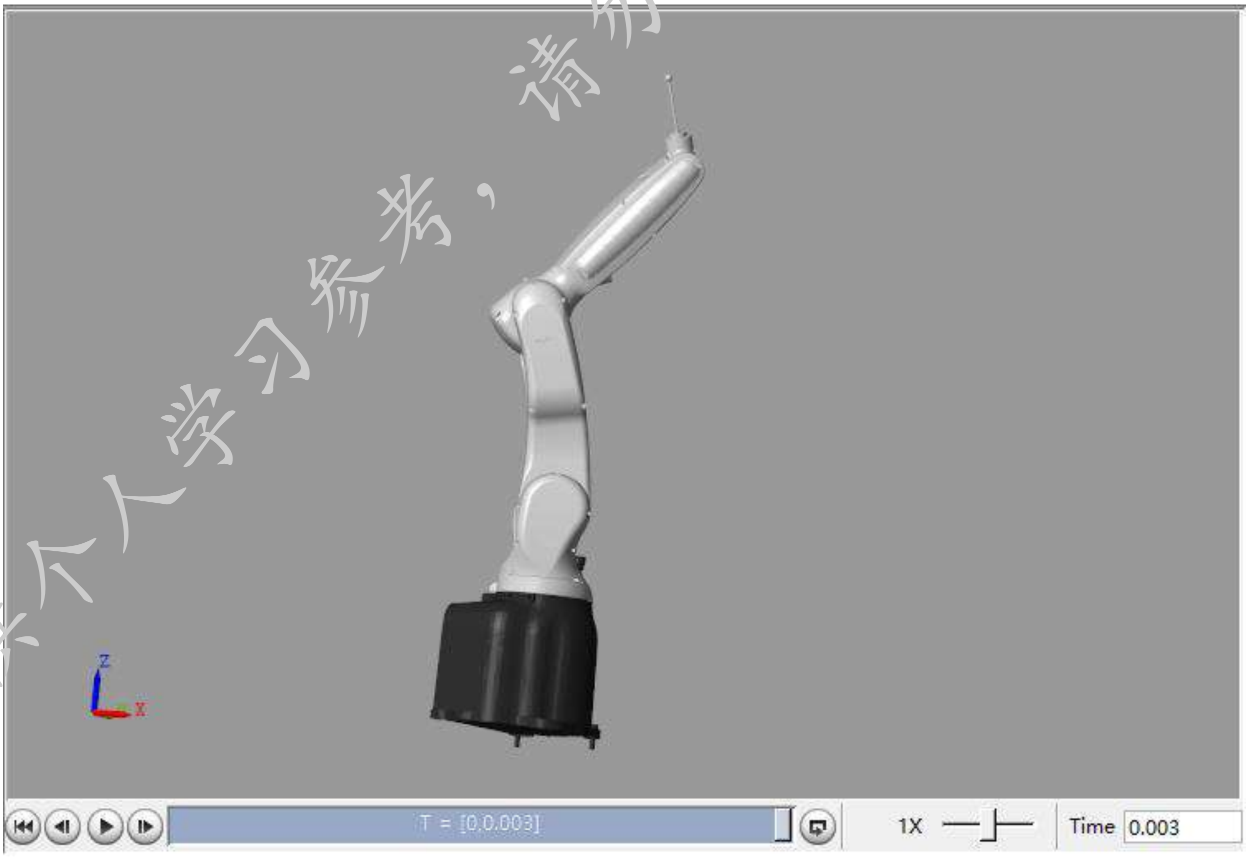


第 6 组解：[0.5236,   0.69813, -0.87266, -2.0944,   0.17453,   -1.4137]

仅供个人学习参考，请勿抄袭或用于商业用途！



第 7 组解：[-2.618, 0.17453, -0.87266, -2.9581, -0.96853,   2.6637]



第 8 组解：[-2.618, 0.17453, -0.87266, 0.18352, 0.96853, -0.47794]

仅供个人学习参考，请勿抄袭或用于商业用途！

四、 选取 8 组解的方法

在解出每个关节角的所有可能解后，在这**每一个解的基础上**都继续求解，求解之初利用类似下面这样检验范围的语句

```
if theta3 < theta3_range(1) || theta3 > theta3_range(2)
    continue
end
```

来确保**不在关节工作空间内的解被排除**。对于最后一轴，由于使用 subproblem1，所以最后只选取出在 180° -180° 内的单个解，而对于在此基础上旋转整圈（±2π）的解则不予考虑。最终选出的解不多于 8 组。

仅供个人学习参考，请勿抄袭或用于商业用途！