廣東工業大學

机器人学课程设计报告

学　　院　　自动化学院

专业班级　控制科学与工程(1)班

学　　号　　2111704036

姓　　名　　冯承健

联系方式　　18819475040

任课教师　　黄之峰

2018 年　4 月　13 日

目　录

擦窗机器人仿真

1 机器人的设计

本文设计了一个六自由度的擦窗机器人，并使用 Matlab 进行仿真分析。机器人的结构如图 1 所示：其中黑色数字为对应的关节编号，第 4 个关节为滑动关节，其余 5 个关节均为旋转关节；红色数字为各根连杆的长度，其中第 4 根连杆的滑动范围为 10→250cm；机器人末端安装了一个毛刷，用于擦洗玻璃（图中绿色平面）。在毛刷的安装当中，我们使末端的连杆和毛刷的平面成45°角，以提供机器人擦窗的灵活性。

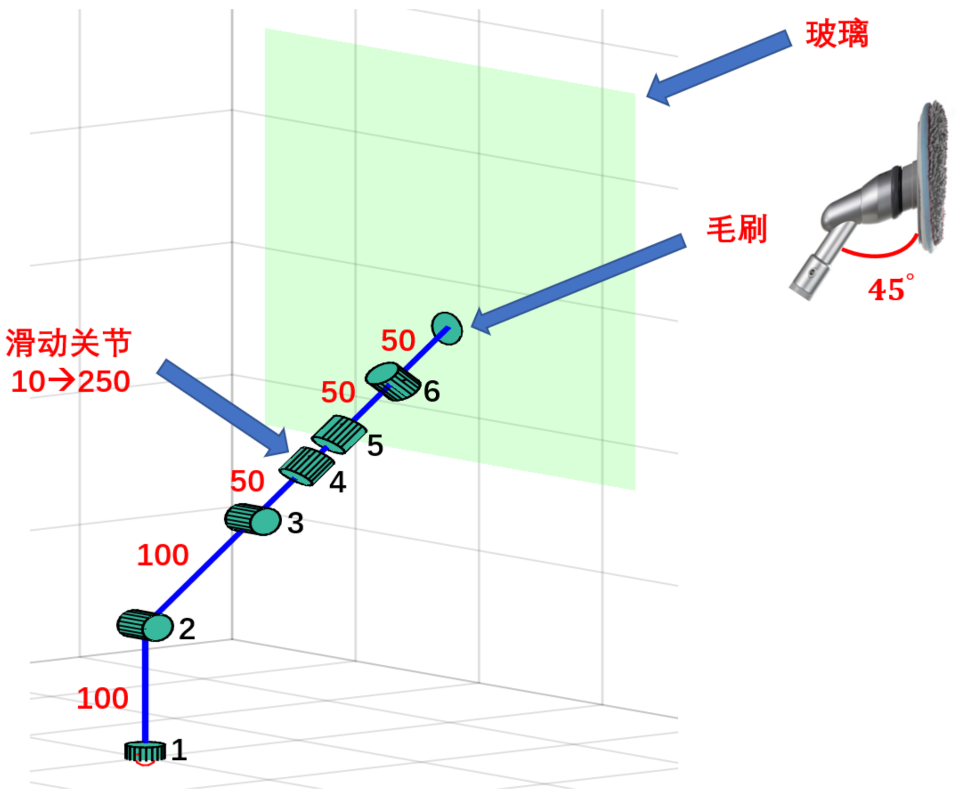


图 1 擦窗机器人的结构（单位：cm）

2 机器人的建模

2.1 建立坐标系

各关节的坐标系构建如图 2 所示。世界坐标系建立为与第一个关节的坐标系重合。

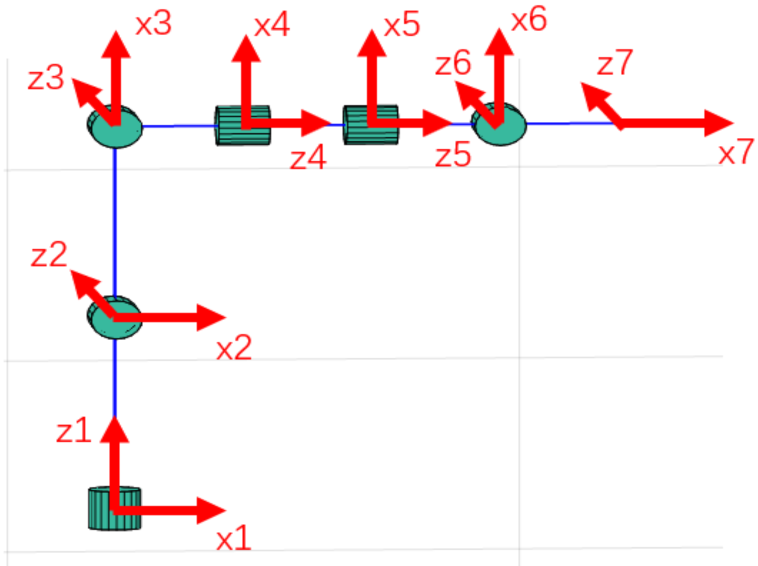


图 2 关节坐标系

2.2 DH 参数表

根据各关节的坐标系及连杆长度，我们可以得到表 3 的改进的 DH 参数表。我们在标准的 DH 参数表里面增加了沿 y 轴移动的动作，以确保每个关节坐标系能够建立在对应关节的位置上，提升实验演示的效果。

		关节编号	θ	d	dy	a	α
A_1	${}^1T_2$	1	θ_1	100	0	0	$-90°$
A_2	${}^2T_3$	2	$-90°+\theta_2$	0	0	100	0
A_3	${}^3T_4$	3	θ_3	0	50	0	$-90°$
A_4	${}^4T_5$	4	0	50+d4	0	0	0
A_5	${}^5T_6$	5	θ_5	50	0	0	$90°$
A_6	${}^6T_7$	6	$90°+\theta_6$	0	0	50	0

表 1 改进的 DH 参数表

为了确保机器人在运动过程中不会碰撞到自己，我们设置了个关节的运动范围，如图表 2 所示。

关节变量	工作范围
θ_1	-180°～180°
θ_2	-90°～90°
θ_3	-180°～0°
d_4	-40～200cm
θ_5	-180°～180°
θ_6	-90°～90°

表 2 关节工作范围

2.3 齐次变换矩阵

根据 DH 参数表，我们可以得到 6 个关节间的齐次变换矩阵（相应程序 Computer_T.m），如下：

$$A1 = \begin{pmatrix} cos(\theta_1) & 0 & -sin(\theta_1) & 0 \\ sin(\theta_1) & 0 & cos(\theta_1) & 0 \\ 0 & -1 & 0 & 100 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{1}$$

$$A2 = \begin{pmatrix} cos(\theta_2 - pi/2) & -sin(\theta_2 - pi/2) & 0 & 100 * cos(\theta_2 - pi/2) \\ sin(\theta_2 - pi/2) & cos(\theta_2 - pi/2) & 0 & 100 * sin(\theta_2 - pi/2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2}$$

$$A3 = \begin{pmatrix} cos(\theta_3) & 0 & -sin(\theta_3) & -50 * sin(\theta_3) \\ sin(\theta_3) & 0 & cos(\theta_3) & 50 * cos(\theta_3) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3}$$

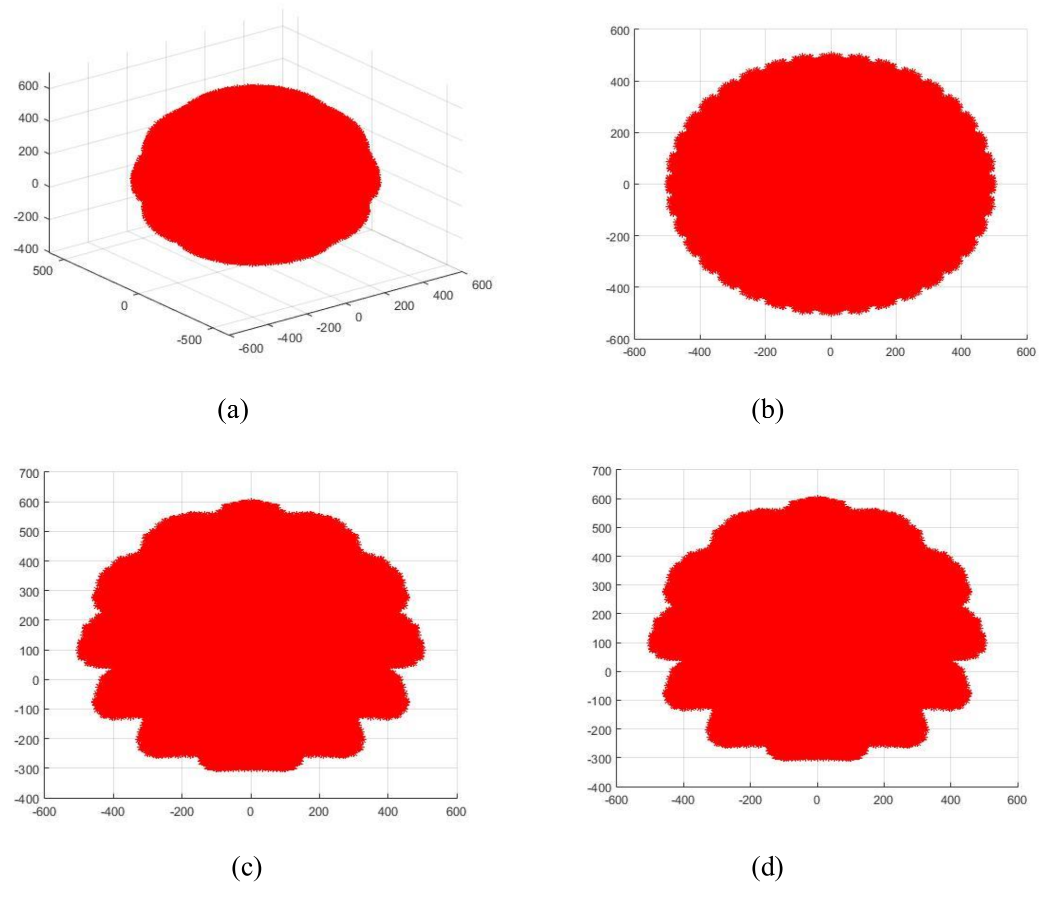
$$A4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d4 + 50 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4}$$

$$A5 = \begin{pmatrix} cos(\theta_5) & 0 & sin(\theta_5) & 0 \\ sin(\theta_5) & 0 & -cos(\theta_5) & 0 \\ 0 & 1 & 0 & 50 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5}$$

$$A6 = \begin{pmatrix} cos(\pi/2+\theta_6) & -sin(\pi/2+\theta_6) & 0 & 50*cos(\pi/2+\theta_6) \\ sin(\pi/2+\theta_6) & cos(\pi/2+\theta_6) & 0 & 50*sin(\pi/2+\theta_6) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{6}$$

3.机器人工作空间分析

通过对各个关节的运动范围进行遍历，我们可以得到机器人的工作空间（相应程序 draw_6DOF_Workplace.m 或其简化版 DHfk6Dof_Workplace.m），如图 3 所示。



(a) (b)

(c) (d)

图 3 机器人的工作空间：(a)立体图；(b)x-y 视图；(c)x-z 视图；(d)y-z 视图。

在水平方向上，机器人最大运动距离对应的姿态如图 4 所示。机器人保持该最大运动距离姿态，关节 1 沿 z 轴旋转一圈即可得到图 3 的 x-y 视图。在垂直方向上，由于关节 1 能够沿 z 轴旋转360°，x-z 视图和 x-y 视图是一样的；在地球坐标系 z 轴的正方向，机器人所能达到的最远距离如图 5(a)所示；在地球坐标系 z 轴的负方向，机器人所能达到的最远距离如图 5(b)所示；由此可知，机器人在地球坐标系 z 轴的正方向比负方向所能达到的工作距离更大。

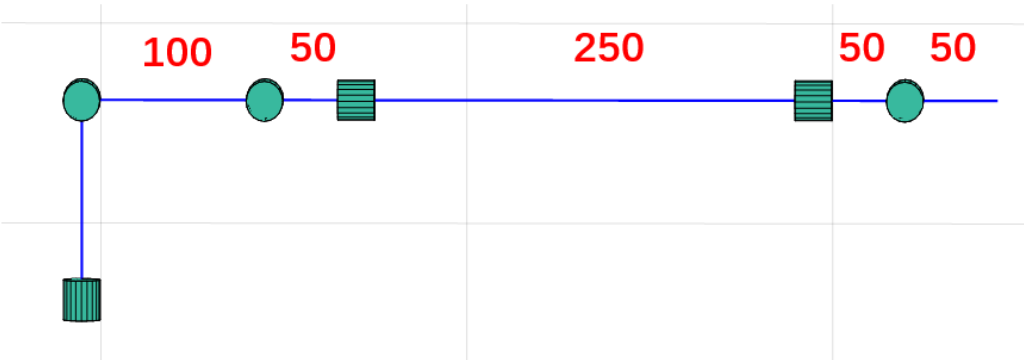
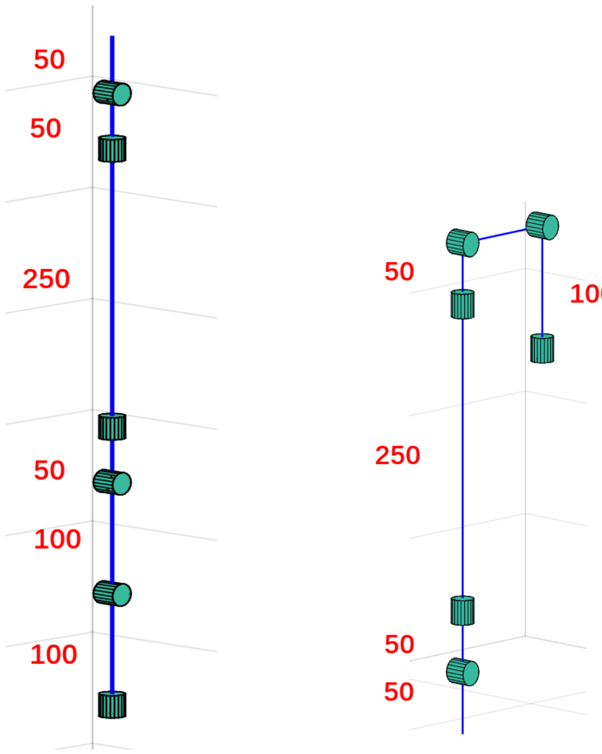


图 4 机器人在水平方向达到最大工作距离的姿态



(a) (b)

图 5 机器人在垂直方向达到最大工作距离的姿态

4 逆运动学数值解[1]

4.1 算法流程

机器人逆运动学数值解法的流程如图6所示（相应程序Move_IK.m）。我们利用微分运动对机器人的位姿进行修正。

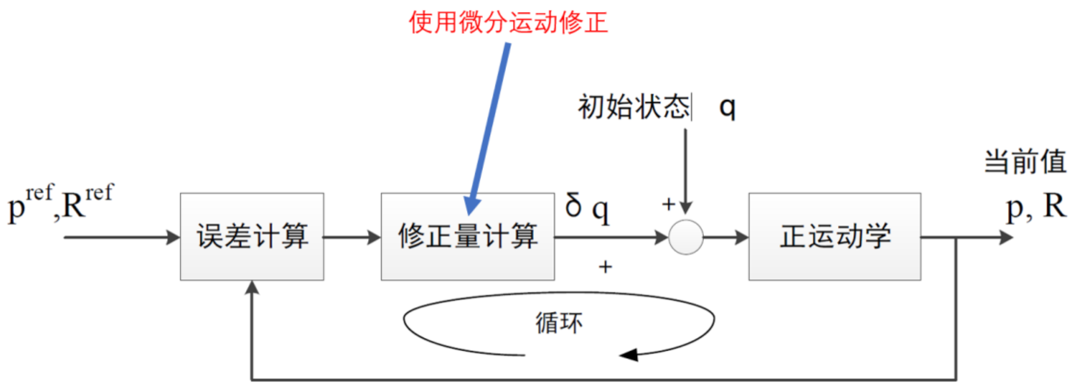


图 6 机器人逆运动学数值解法流程

4.2 误差计算

当前值和给定值的误差计算中，位置误差的计算公式如下：

$$p^{err} = p^{ref} - p \tag{7}$$

角速度误差的计算过程分为 3 步。

1、计算误差旋转矩阵：

$$R^{err} = R^{T} R^{ref} \tag{8}$$

2、误差旋转矩阵对应的角速度误差：

$$\omega = \begin{cases} \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^{T} & (R^{err} = E\text{时}) \\ \dfrac{\theta}{2\sin\theta}\begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix} & (R^{err} \neq E\text{时}) \end{cases} \tag{9}$$

3、世界坐标系的角速度误差：

$$w^{err} = Rw \tag{10}$$

最后，总的误差（相应程序 CalcVWerr.m）可以定义为：

$$err = \left\| p^{err} \right\|^2 + \left\| \omega^{err} \right\|^2 \tag{11}$$

4.3 微分运动

机器人末端速度$\dot{X}$和各关节的速度$\dot{\theta}$关系如下：

$$\dot{\theta} = J(\theta)^{-1} \dot{X} \tag{12}$$

采用牛顿下山法，结合微分运动的逆问题，可以得到各个关节的微分量(相应程序（相应程序 Move_IK.m 和 Jacobian6DoF_Ln.m)：

$$\delta q = \lambda J^{-1} \begin{bmatrix} \delta p \\ \delta \omega \end{bmatrix} \quad (\lambda \in (0,1]) \tag{13}$$

其中λ称为下山因子，从$\lambda = 1$开始选择，逐次将λ减半进行试算，直到能使总误差 err 下降。我们设定总误差 err 小于10^{-6}时，机器人达到参考位姿，结束位姿的修正。

5.机器人擦窗过程

机器人擦窗的控制流程及相应的演示（相应程序 Rub_Window_IK.m）分别如图 7 和图 8 所示。在擦窗过程中，我们首先调整机器人末端姿态至与玻璃平行，避免其位置调整时撞坏玻璃；然后保持机器人末端姿态，调整末端位置至玻璃表面；最后保持机器人末端姿态，按预设轨迹擦玻璃。机器人擦窗仿真效果如图 9 所示，图 9(a)为机器人按五角星轨迹擦玻璃，图 9(b)为机器人按余弦轨迹擦玻璃。

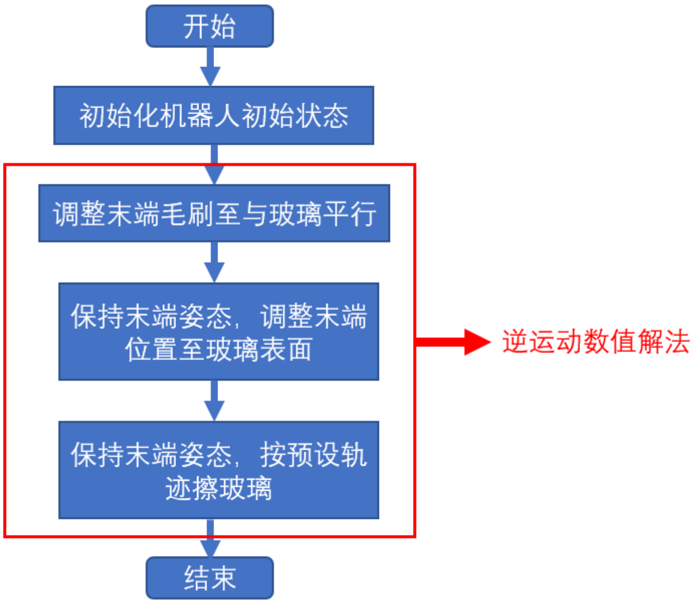
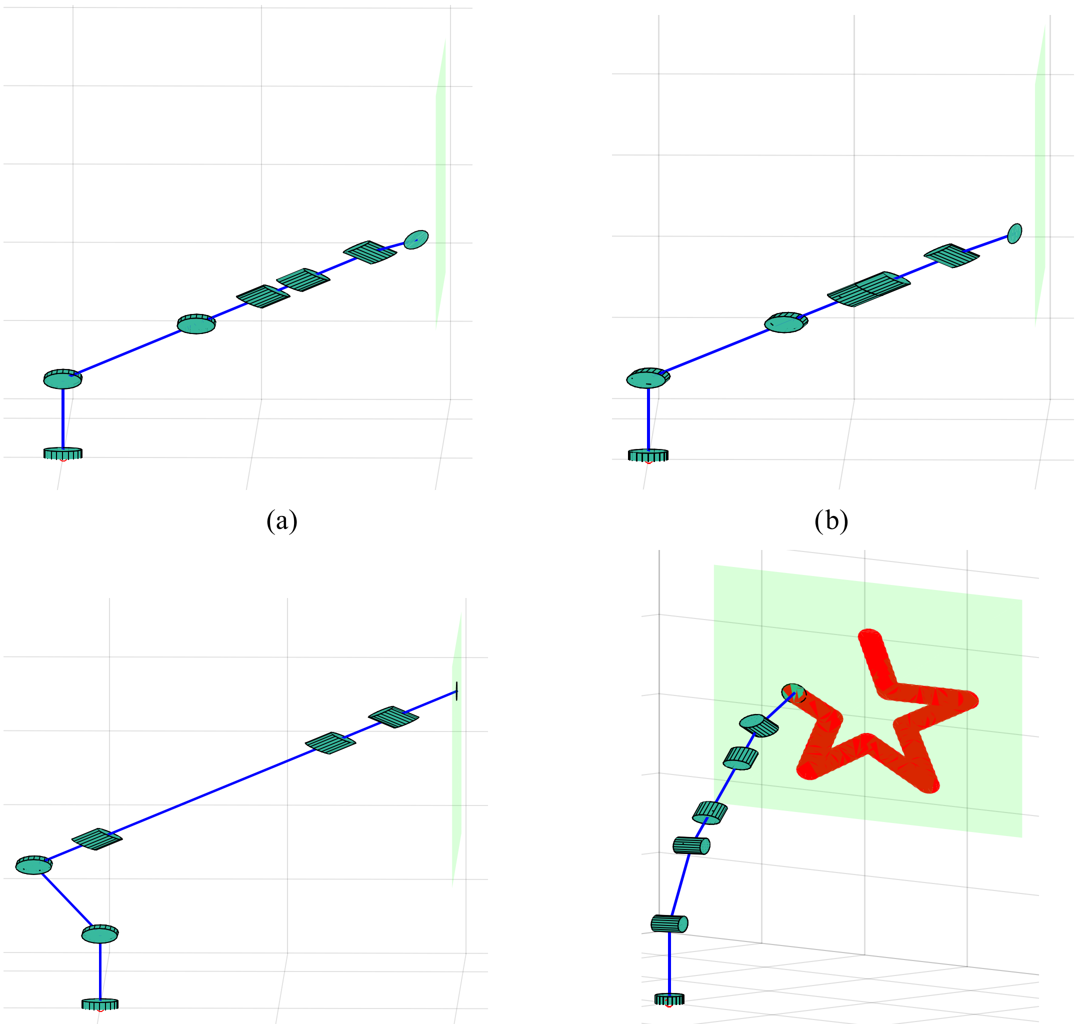
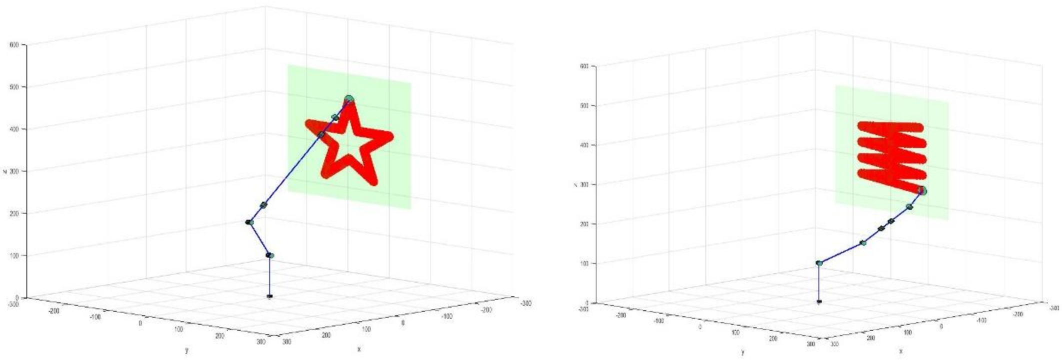


图 7 机器人擦窗控制流程



(a) (b) (c) (d)

图 8 机器人擦窗的 4 个步骤：(a)→(b)→(c)→(d)



(a) (b)

图 8 机器人擦窗仿真

6.机器人奇异位型分析

当雅克比矩阵不满秩时，雅克比矩阵不可逆，机器人处于奇异位型。雅可比矩阵第 i 列表示第 i 个关节对末端速度的贡献，第 j 行表示第 j 个末端速度变量受各关节的影响。所以，机器人奇异位型通常出现在图 9 所示的两种情况，：

1、当两个关节对末端贡献相同时，对应的两列互相关，雅各比矩阵不满秩；

2、当所有关节对末端的某个速度没有贡献时，对应的行为零，雅克比不满秩。

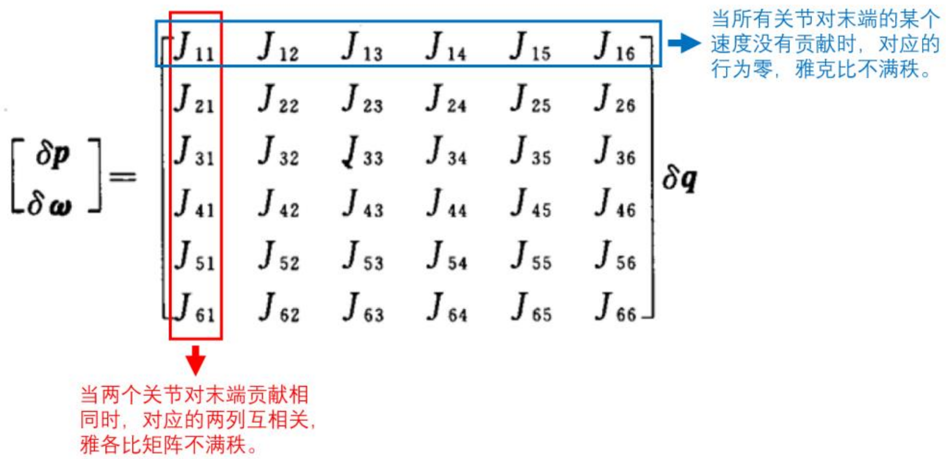


图 9 机器人奇异位型在雅克比矩阵中的表现

例如在图 10 所示的机器人位型下，各个关节都不能提供给末端沿 y 轴移动的速度。此时机器人处于奇异位型，其对应的雅克比矩阵不满秩，为：

$$J=\begin{bmatrix} 0.0000 & 300.0000 & 200.0000 & 0.0000 & 0.0000 & 50.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0 & -0.0000 & 0.0000 \\ 0 & -0.0000 & -0.0000 & 1.0000 & -0.0000 & -0.0000 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.0000 & 1.0000 & 0 & 0 & 1.0000 \\ 1.0000 & 0.0000 & 0.0000 & 0 & 1.0000 & 0.0000 \end{bmatrix} \tag{14}$$

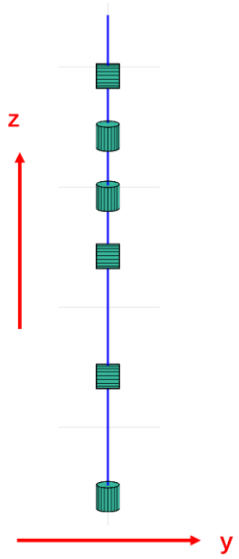


图 10 奇异位型举例

[1]《仿人机器人》梶田秀司

附录

1 机器人擦窗程序流程图

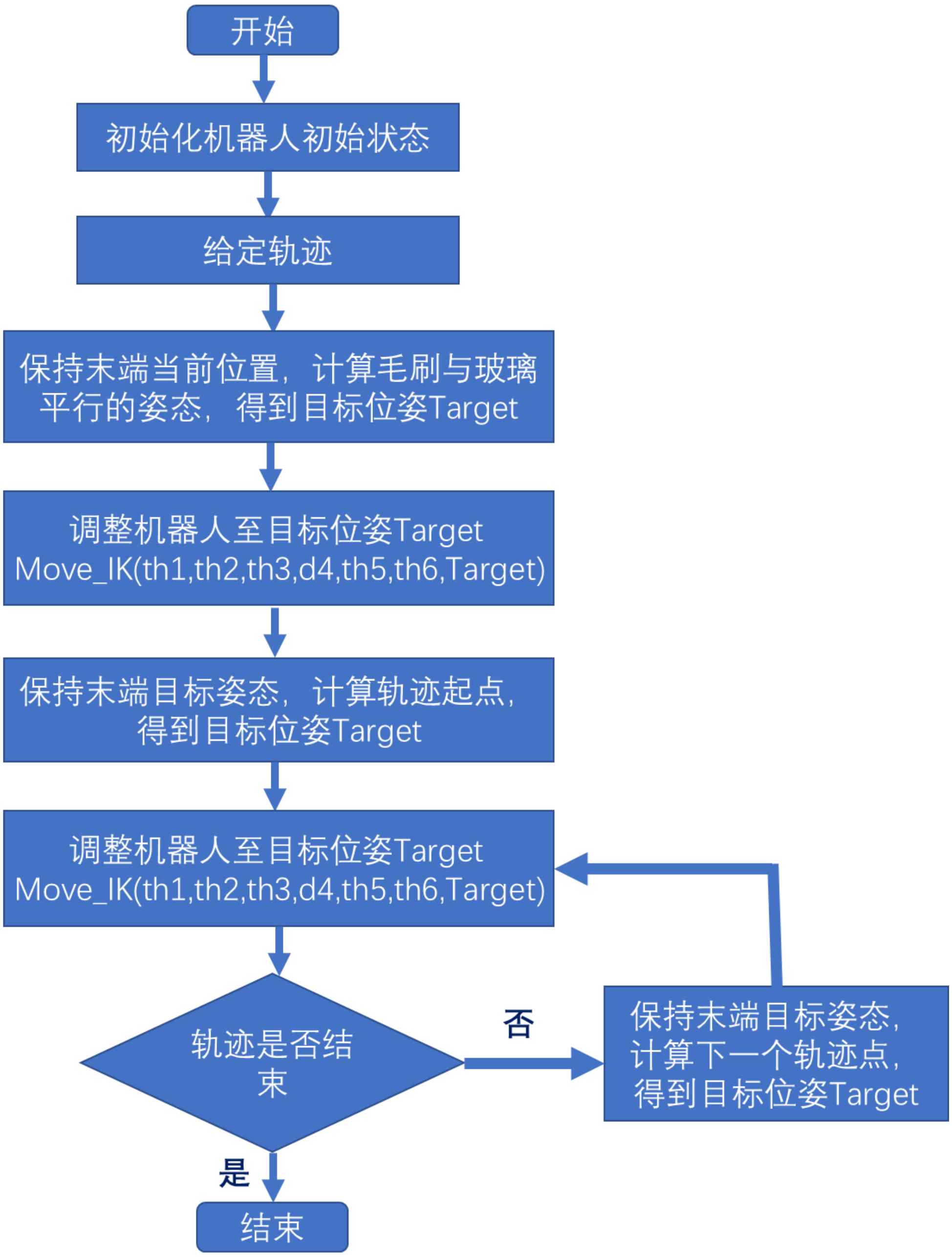


图 11 机器人擦窗程序流程图

2 部分重要程序

2.1 Computer_T.m

```
close all;
clear;

ToDeg = 180/pi;
ToRad = pi/180;

syms theta d a alpha y;
T1 = [cos(theta) -sin(theta) 0 0; sin(theta) cos(theta) 0 0; 0 0 1 0; 0 0 0 1];
T2 = [1 0 0 0; 0 1 0 0; 0 0 1 d; 0 0 0 1];
T3 = [1 0 0 a; 0 1 0 0; 0 0 1 0; 0 0 0 1];
T4 = [1 0 0 0; 0 cos(alpha) -sin(alpha) 0; 0 sin(alpha) cos(alpha) 0; 0 0 0 1];
Ty = [1 0 0 0; 0 1 0 y; 0 0 1 0; 0 0 0 1];
T = T1 * T2 * Ty * T3 * T4

syms theta1 theta2 theta3 d4 theta5 theta6
theta=theta1;d=100;y=0;a=0;alpha = -90*ToRad;
A1 = subs(T)
theta=-90*ToRad+theta2;d=0;y=0;a=100;alpha = 0;
A2 = subs(T)
theta=theta3;d=0;y=50;a=0;alpha = -90*ToRad;
A3 = subs(T)
theta=0;d=50+d4;y=0;a=0;alpha = 0;
A4 = subs(T)
theta=theta5;d=50;y=0;a=0;alpha = 90*ToRad;
A5 = subs(T)
theta=90*ToRad+theta6;d=0;y=0;a=50;alpha = 0;
A6 = subs(T)
A = A1 * A2 * A3 * A4 * A5 * A6
```

2.2 draw_6DOF_Workplace.m

```
close all;
clear;
ToDeg = 180/pi;
ToRad = pi/180;
point1=[];
point2=[];
point3=[];
th_interval = 40;
z_interval = 4;

th1=0;
th2=-90;
th3=0;
d4=50;
th5=0;
th6=90;
global Link
num = 1
for theta1=-180:th_interval:180
    for theta2=-90:th_interval:90
        for theta3=-180:th_interval:0
            for dz4=-40:z_interval:200
                for theta5=-180:th_interval:180
                    for theta6=-90:th_interval:90
                         DHfk6Dof_Workplace(th1+theta1,th2+theta2,th3+t
                         heta3,d4+dz4,th5+theta5,th6+theta6,1,1);
                        point1(num) = Link(7).p(1);
                        point2(num) = Link(7).p(2);
                        point3(num) = Link(7).p(3);
                        num = num + 1;
                        plot3(point1,point2,point3,'r*');hold on;
                    end
                end
            end
        end
    end
end
cla;
plot3(point1,point2,point3,'r*');
axis([-400,400,-400,400,-400,400]);
grid on;
```

2.3 Jacobian6DoF_Ln.m

```
function J=Jacobian6DoF_Ln(th1,th2,th3,d4,th5,th6)
% close all
global Link
jsize=6;
J=zeros(6,jsize);

Link(2).th=th1*pi/180;
Link(3).th=th2*pi/180;
Link(4).th=th3*pi/180;
Link(5).dz=d4;
Link(6).th=th5*pi/180;
Link(7).th=th6*pi/180;

for i=1:7
Matrix_DH_Ln(i);
end

Link(1).p=Link(1).p(1:3);
for i=2:7
        Link(i).A=Link(i-1).A*Link(i).A;
        Link(i).p= Link(i).A(:,4);
        Link(i).n= Link(i).A(:,1);
        Link(i).o= Link(i).A(:,2);
        Link(i).a= Link(i).A(:,3);
        Link(i).R=[Link(i).n(1:3),Link(i).o(1:3),Link(i).a(1:3)];
end

target=Link(7).p(1:3);

for n=1:3
       a=Link(n).R*Link(n).az;
       J(:,n)=[cross(a,target-Link(n).p(1:3)); a];
end
for n=4:4
       a=Link(n).R*Link(n).az;
       J(:,n)=[a; [0,0,0]'];
end
for n=5:jsize
       a=Link(n).R*Link(n).az;
       J(:,n)=[cross(a,target-Link(n).p(1:3)); a];
end
```

2.4 CalcVWerr.m

```
function err = CalcVWerr(Target, Current)
    Target.p= Target.A(1:3,4);
    Target.n= Target.A(:,1);
    Target.o= Target.A(:,2);
    Target.a= Target.A(:,3);
    Target.R=[Target.n(1:3),Target.o(1:3),Target.a(1:3)];

    Perr = Target.p - Current.p(1:3);
    Rerr = Current.R' * Target.R;
    theta = acos((Rerr(1,1)+Rerr(2,2)+Rerr(3,3)-1)/2.0);
    if theta==0
        Werr=[0,0,0]';
    else
        Werr = theta/(2.0*sin(theta)) * [Rerr(3,2)-Rerr(2,3);Rerr(1,3)-
        Rerr(3,1);Rerr(2,1)-Rerr(1,2)];
        Werr = Current.R * Werr;
    end
    err = [Perr;Werr];
end
```

2.5 Move_IK.m

```
function [th1,th2,th3,d4,th5,th6] = Move_IK(th1,th2,th3,d4,th5,th6,Target)
    global Link
    num=1;
    while 1
        figure(1);
        DHfk6Dof(th1,th2,th3,d4,th5,th6,0,0);

        J=Jacobian6DoF_Ln(th1,th2,th3,d4,th5,th6);
        x=det(J) ;

        err = CalcVWerr(Target, Link(7));
        E = err'*err;
        if E<1e-6
            break
        end

        E_tmp = E;
        lambda = 1;
```

```
        while E-E_tmp<=0
            dD = lambda*err;
            dth=inv(J)*dD;
            th1_tmp=th1+dth(1)/pi*180;
            th2_tmp=th2+dth(2)/pi*180;
            th3_tmp=th3+dth(3)/pi*180;
            d4_tmp=d4+dth(4);
            th5_tmp=th5+dth(5)/pi*180;
            th6_tmp=th6+dth(6)/pi*180;

            th2_tmp=restrain_value(th2_tmp,-180,0);
            th3_tmp=restrain_value(th3_tmp,-180,0);
            d4_tmp=restrain_value(d4_tmp,10,250);
            th6_tmp=restrain_value(th6_tmp,0,180);


            DHfk6Dof(th1_tmp,th2_tmp,th3_tmp,d4_tmp,th5_tmp,th6_tmp,0,
            0);
            err = CalcVWerr(Target, Link(7));
            E_tmp = err'*err;
            lambda = lambda/2.0;
        end

        th1=th1_tmp;
        th2=th2_tmp;
        th3=th3_tmp;
        d4=d4_tmp;
        th5=th5_tmp;
        th6=th6_tmp;

        xout(num)=x;
        t(num)=num;
        E_polt(num)=E_tmp;

        if E_tmp<1e-6
            break
        end

        num=num+1;
    end
end
```

2.6 Rub_Window_IK.m

```
close all;
clear;

global Link

ToDeg = 180/pi;
ToRad = pi/180;

th1=0;
th2=-135;
th3=-90;
d4=30;
th5=90;
th6=135;
figure(1);
DHfk6Dof(th1,th2,th3,d4,th5,th6,1,1);
view(134,12);
DHfk6Dof(th1,th2,th3,d4,th5,th6,1,1);

pause;
if 1 %1 is rub five-pointed star track; 0 is cos track.
    p = [0 0 92 1; 0 20 21 1; 0 96 21 1;0 36 -21 1; 0 58 -91 1;
              0 0 -47 1;0 -58 -91 1;0 -36 -21 1;0 -96 21 1;0 -20 21 1;0 0 92 1];
else
    z=4*pi:-pi/2:-4*pi;
    y=cos(z)*80;
    z=z*80/(4*pi);
    x=zeros(size(z));
    n=ones(size(z));
    p=[x;y;z;n]'
end

point_num=10;

good_point=[-200,0,345,1];
for i=1:4
    p(:,i)=p(:,i)+good_point(i);
end

point1=[];
point2=[];
```

```
point3=[];
num=1;

target_z = [-sqrt(2)/2.0 0 sqrt(2)/2.0 0]';

Target.A = Link(7).A
det = target_z - Link(7).A(:,1);
for i=1:5
        Target.A = Link(7).A;
        Target.A(:,1) = Link(7).A(:,1) + det*1/5;
        [th1,th2,th3,d4,th5,th6] = Move_IK(th1,th2,th3,d4,th5,th6,Target);
        figure(1);
        DHfk6Dof(th1,th2,th3,d4,th5,th6,1,1);
end

for i=1:size(p,1)
      det = p(i,:)' - Link(7).A(:,4);
      if i==1
            N=10;
      else
            N=point_num;
      end
      for j=1:N
            Target.A = Link(7).A
            Target.A(:,1)=target_z;
            Target.A(:,4) = Link(7).A(:,4) + det*1/N;
            [th1,th2,th3,d4,th5,th6] = Move_IK(th1,th2,th3,d4,th5,th6,Target);
            figure(1);
            DHfk6Dof(th1,th2,th3,d4,th5,th6,1,1);
            if i>1
                  [x,y,z] = BrushCylinder(Link(7).p, Link(7).R * Link(8).az, 12);
                  point1=[point1;x];
                  point2=[point2;y];
                  point3=[point3;z];
                  patch(point1',point2',point3','r','edgealpha',0);hold on;
                  num=num+1;
            end
      end
end
DHfk6Dof(th1,th2,th3,d4,th5,th6,0,1);
```