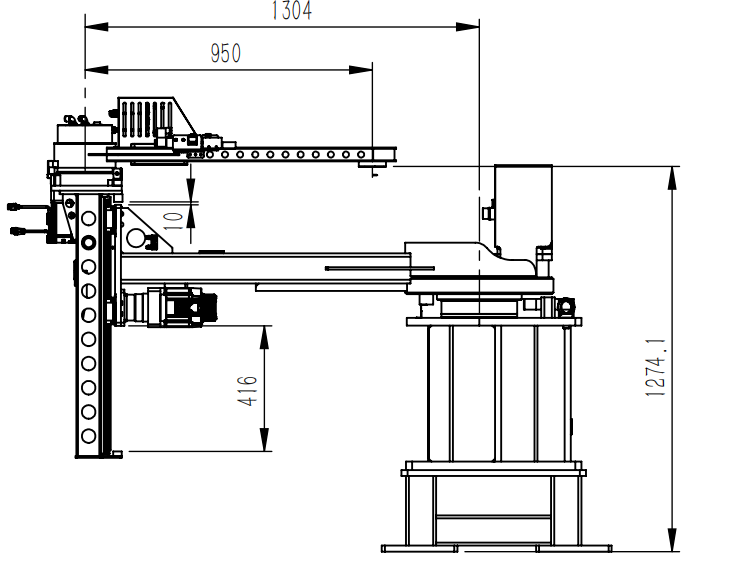
1. **机型简介**



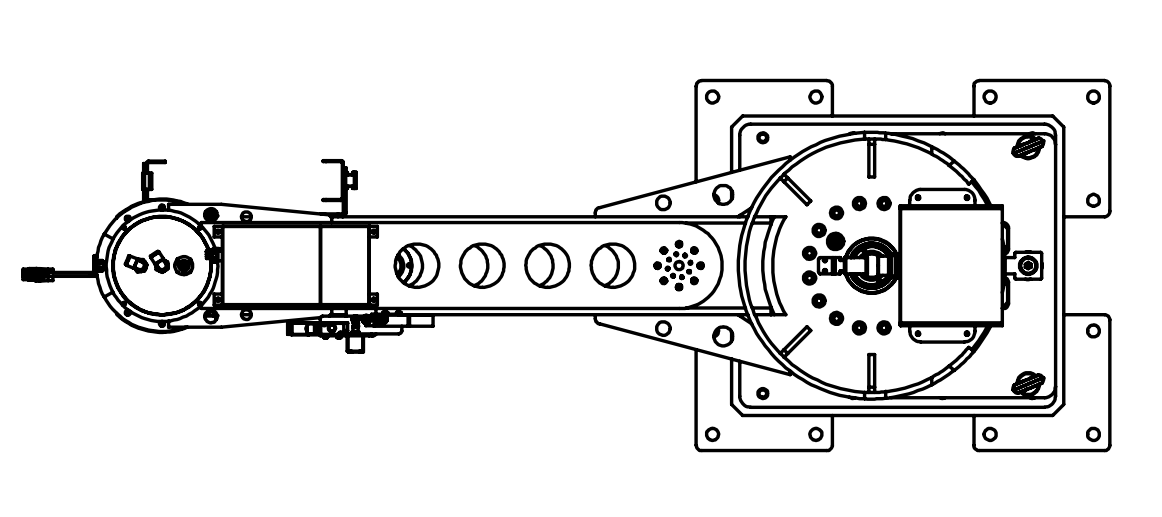
Z

X

J2-

J2+

图1.1 机器人侧视图



X

Y

J1+

J4+

J4-

J3+

J3-

J1-

图1.2 机器人俯视图

表1.1 SC10产品参数表

|  |  |  |
| --- | --- | --- |
| 产品参数表 | | |
| 结构形式 | | 水平多关节机器人(RPRR) |
| 自由度 | | 四轴 |
| 有效负载 | | 10KG |
| 最大活动半径 | | 2221.63mm |
| 重复定位精度 | | ±mm |
| 运动范围 | J1 | ±180° |
| J2 | -10mm, 400mm |
| J3 | ±165° |
| J4 | ±360° |
| 本体重量 | | 560.945KG |
| 环境条件 | 温度 | 0~45℃ |
| 湿度 | 20~80%RH(不结露) |
| 振动 | 4.9m/s2以下 |
| ﹡不可有引火性及腐蚀性气液体  ﹡不可涉及水、油、粉  ﹡不可靠近电磁气源头 | |
| 电源容量 | | KVA |

本简介所涉及的产品资料仅供参考，最终图纸请以实际签订的合同内容为准。

1. **坐标系统**

建立坐标系遵循机器人右手定则，拇指向上为Z轴正方向，食指指向方向为X轴正方向，中指指向方向为Y轴正方向。

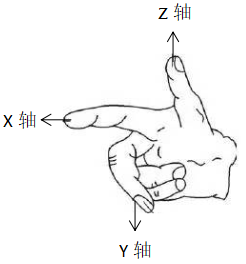
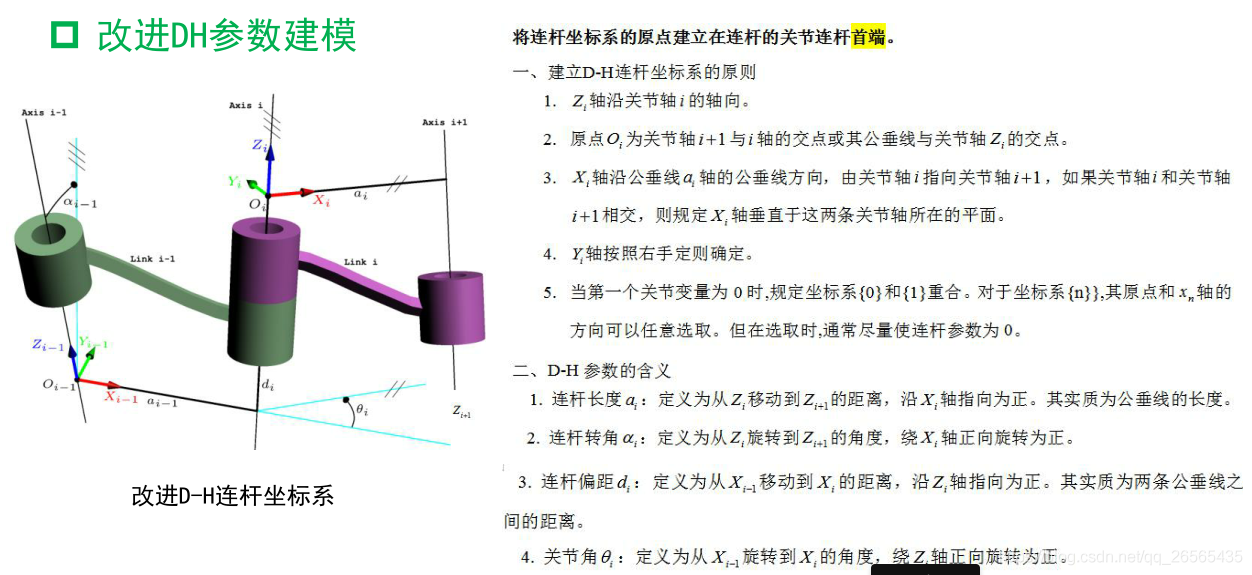
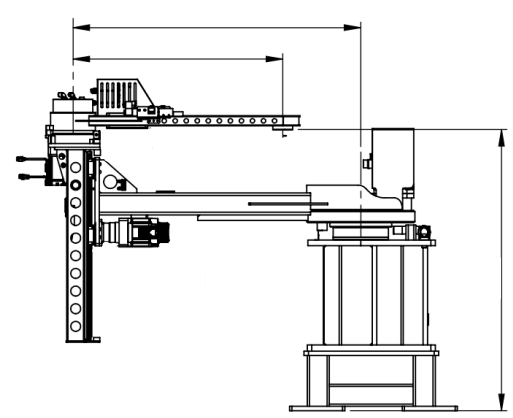


图2.1 右手定则示意图

采用MDH方法进行整机建模，具体规则如下：





X0

X1

X2

X3

X4

Z4

Z3

Z2

Z1

Z0

d1

a1

a3

1100

d3

图2.2 坐标系统示意图

表2.1 DH参数表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **i** | **theta** | **d** | **a** | **alpha** | **axis** |
| 1 | 0 | d1 | 0 | 0 | q1(axis1) |
| 2 | 0 | 0 | a1 | 0 | l2(axis2) |
| 3 | 180° | d3(1100mm-d1) | 0 | 0 | q3(axis3) |
| 4 | 180° | 0 | a3 | 180° | q4(axis4) |

(注：d1没有实际数值也不影响结果，计算中能消掉。)

1. **正运动学**

**3.1 符号推理**

根据相邻坐标系间的位姿计算公式：



可依次求得、、、如下：

|  |  |
| --- | --- |
|  |  |
|  |  |

则末端TCP原点在机器人基坐标系下的齐次位姿矩阵为：



可等效变换为：



**3.2 测试样例**

1. 零位移状态，即已知



代入DH数据可得TCP位姿：



1. 正限位状态，即已知



代入DH数据可得TCP位姿：



1. 负限位状态，即已知



代入DH数据可得TCP位姿：



1. **逆运动学**
   1. **符号推理**

若已知法兰位姿描述如下：



易得：，其他轴旋转角位移可根据几何关系在俯视图中求解。

|  |  |
| --- | --- |
| SC10_逆解集合简图 | SC10_逆解几何简图(q3-) |
| (1) q3角位移为正值 | (2) q3角位移为负值 |
| 图4.1 俯视杆件几何简图 | |

上图中，O\_J1 O\_J3=a1，O\_J3 O\_J4=a3，y\_tcp=Y，x\_tcp=X。



先以左图的情况为例进行分析。在ΔO\_J1 O\_J3 O\_J4中：









至此求得左图中全部轴位移。进而分析右图中的情况。与左图相比，仅q1和q3的计算表达式稍有区别。





(注：未单独列出的计算与左图相同，参照左图的表达式即可。)

* 1. **测试样例**

1. 已知TCP位姿为：



代入DH数据可得关节位移：



1. 已知TCP位姿为：



代入DH数据可得关节位移：



1. 已知TCP位姿为：



代入DH数据可得关节位移：



1. **代码实现**
   1. **MATLAB代码**
      1. **正解函数**
2. % 正解函数
3. % 输入：各关节位移
4. %    旋转轴位移值单位为°,移动轴位移值单位为mm
5. % 输出：TCP位姿描述(XYZABC)
6. %    XYZ输出值单位为mm,ABC输出值单位为°
7. function [p]=SC10\_fkine(axis)
8. %{
9. |\_i\_|\_theta\_|\_d\_|\_a\_|\_alpha\_|\_\_axis\_\_|
10. | 1 |   0   |d1 | 0 |   0   | axis1  |
11. | 2 |   0   | 0 |a1 |   0   | axis2  |
12. | 3 |  180  |d3 | 0 |   0   | axis3  |
13. | 4 |  180  | 0 |a3 |  180  | axis4  |
14. %}
15. % 连杆参数
16. a1=1304;
17. a3=950;
18. d1=600;
19. d3=1100-d1;
20. % DH参数
21. a=[0,a1,0,a3];
22. d=[d1,0,d3,0];
23. alpha=[0,0,0,pi];
24. theta=[0,0,pi,pi];
25. T=eye(4);
26. for i=1:4
27. if i==2
28. A=mdh\_matrix(a(i),alpha(i),d(i)+axis(i),theta(i));
29. else
30. A=mdh\_matrix(a(i),alpha(i),d(i),theta(i)+deg2rad(axis(i)));
31. end
32. T=T\*A;
33. end
34. % 等效变换
35. p=[T(1:3,4)',rad2deg(tform2eul(T,'ZYX'))];
36. end
37. % A矩阵的计算函数(MDH方法)
38. function [A]=mdh\_matrix(a,alpha,d,theta)
39. A=eye(4);
40. A(1,1)=cos(theta);
41. A(1,2)=-sin(theta);
42. A(1,3)=0;
43. A(1,4)=a;
44. A(2,1)=sin(theta)\*cos(alpha);
45. A(2,2)=cos(theta)\*cos(alpha);
46. A(2,3)=-sin(alpha);
47. A(2,4)=-sin(alpha)\*d;
48. A(3,1)=sin(theta)\*sin(alpha);
49. A(3,2)=cos(theta)\*sin(alpha);
50. A(3,3)=cos(alpha);
51. A(3,4)=cos(alpha)\*d;
52. A(4,1)=0;
53. A(4,2)=0;
54. A(4,3)=0;
55. A(4,4)=1;
56. end
    * 1. **逆解函数**
57. % 逆解函数
58. % 输入：TCP位姿描述(XYZABC)
59. %       XYZ输出值单位为mm,ABC输出值单位为°
60. % 输出：各关节位移
61. %       旋转轴位移值单位为°,移动轴位移值单位为mm
62. function [q]=SC10\_ikine(p)
63. % 连杆参数
64. a1=1304;
65. a3=950;
66. d1=600;
67. d3=1100-d1;
68. % TCP位姿
69. px=p(1);
70. py=p(2);
71. pz=p(3);
72. pa=deg2rad(p(4));
73. pb=deg2rad(p(5));
74. pc=deg2rad(p(6));
75. r=sqrt(px^2+py^2);
76. curj3=2; % 当前J3轴的角位移
77. % 求关节转角1
78. q11=atan2(py,px);
79. cq12=(a1^2+r^2-a3^2)/(2\*a1\*r);
80. sq12=sqrt(1-cq12^2);
81. q12=atan2(sq12,cq12);
82. q1=q11+q12;
83. % 求关节转角2
84. l2=pz-d1-d3;
85. % 求关节转角3
86. cq3=(a3^2+a1^2-r^2)/(2\*a1\*a3);
87. sq3=sqrt(1-cq3^2);
88. q3=atan2(sq3,cq3);
89. % 判断旋转方向
90. if q3\*curj3<0
91. q3=-q3;
92. q1=q1-2\*q12;
93. end
94. % 求关节转角4
95. q4=q1+q3-pa;
96. q=[rad2deg(q1),l2,rad2deg(q3),rad2deg(q4)];
97. end
    1. **ST代码**
       1. **正解函数**
98. FUNCTION FN\_RPRR\_SC10\_FK : CoordPoint
99. VAR\_INPUT
100. sActAngle : AxisAngle; *//待正解的四轴各轴弧度*
101. a1,a3  : LREAL;  *//对应DH参数值*
102. *//d1,d3  : LREAL; d1+d3=1100是定值，直接将1100代入计算了*
103. END\_VAR
104. *///////////////////////////////////////////////////////////*
105. sActAngle.lrAxis1:=sActAngle.lrAxis1\*Deg2Rad;
106. sActAngle.lrAxis3:=sActAngle.lrAxis3\*Deg2Rad;
107. *//TCP位姿*
108. FN\_RPRR\_SC10\_FK.X:=a1\*COS(sActAngle.lrAxis1)-a3\*COS(sActAngle.lrAxis1+sActAngle.lrAxis3);
109. FN\_RPRR\_SC10\_FK.Y:=a1\*SIN(sActAngle.lrAxis1)-a3\*SIN(sActAngle.lrAxis1+sActAngle.lrAxis3);
110. FN\_RPRR\_SC10\_FK.Z:=sActAngle.lrAxis2+1100;
111. FN\_RPRR\_SC10\_FK.A:=(sActAngle.lrAxis1+sActAngle.lrAxis3)\*Rad2Deg-sActAngle.lrAxis4;
112. FN\_RPRR\_SC10\_FK.B:=0;
113. FN\_RPRR\_SC10\_FK.C:=0;
114. *//附加轴位姿*
115. FN\_RPRR\_SC10\_FK.Aux[0] := sActAngle.lrAxis7;
116. FN\_RPRR\_SC10\_FK.Aux[1] := sActAngle.lrAxis8;
117. FN\_RPRR\_SC10\_FK.Aux[2] := sActAngle.lrAxis9;
     * 1. **逆解函数**
118. FUNCTION FN\_RPRR\_SC10\_IK : AxisAngle
119. VAR\_INPUT
120. cCoordPoint  : CoordPoint; *//当前空间位置姿态*
121. InCurAxisAngle : AxisAngle; *//上一次关节角度*
122. a1,a3  : LREAL;  *//DH参数*
123. *//d1,d3  : LREAL;  d1+d3=1100是定值，直接将1100代入计算了*
124. END\_VAR
125. VAR
126. tmp:LREAL:=0;
127. stmp,ctmp:LREAL:=0;
128. END\_VAR
129. *////////////////////////////////////////////////////////////////*
130. cCoordPoint.A:=cCoordPoint.A\*deg2rad;
131. tmp:=SQRT(EXPT(cCoordPoint.X, 2) + EXPT(cCoordPoint.Y, 2));
132. *//求轴1*
133. ctmp:=(EXPT(a1,2)+EXPT(tmp,2)-EXPT(a3,2))/(2\*a1\*tmp);
134. stmp:=SQRT(1-EXPT(ctmp,2));
135. FN\_RPRR\_SC10\_IK.lrAxis1:=NW\_atan2(stmp, ctmp)+NW\_atan2(cCoordPoint.Y, cCoordPoint.X);
136. *//求轴2*
137. FN\_RPRR\_SC10\_IK.lrAxis2:=cCoordPoint.Z-1100;
138. *//求轴3*
139. ctmp:=(EXPT(a1,2)+EXPT(a3,2)-EXPT(tmp,2))/(2\*a1\*a3);
140. stmp:=SQRT(1-EXPT(ctmp,2));
141. FN\_RPRR\_SC10\_IK.lrAxis3:=NW\_atan2(stmp, ctmp);
142. *//转向判断*
143. IF FN\_RPRR\_SC10\_IK.lrAxis3\*InCurAxisAngle.lrAxis3<0 THEN *//仅用于判断符号，不用换算单位*
144. FN\_RPRR\_SC10\_IK.lrAxis3:=-FN\_RPRR\_SC10\_IK.lrAxis3;
145. FN\_RPRR\_SC10\_IK.lrAxis1:=FN\_RPRR\_SC10\_IK.lrAxis3-2\*NW\_atan2(stmp, ctmp);
146. END\_IF
147. *//求轴4*
148. FN\_RPRR\_SC10\_IK.lrAxis4:=FN\_RPRR\_SC10\_IK.lrAxis1+FN\_RPRR\_SC10\_IK.lrAxis3-cCoordPoint.A;
149. FN\_RPRR\_SC10\_IK.lrAxis4:=NW\_atan2(SIN(FN\_RPRR\_SC10\_IK.lrAxis4), COS(FN\_RPRR\_SC10\_IK.lrAxis4));
150. IF (FN\_RPRR\_SC10\_IK.lrAxis4 - InCurAxisAngle.lrAxis4\*Deg2Rad) > PI THEN
151. FN\_RPRR\_SC10\_IK.lrAxis4:=FN\_RPRR\_SC10\_IK.lrAxis4 - PI2;
152. ELSIF (FN\_RPRR\_SC10\_IK.lrAxis4 - InCurAxisAngle.lrAxis4\*Deg2Rad) < -PI THEN
153. FN\_RPRR\_SC10\_IK.lrAxis4:=FN\_RPRR\_SC10\_IK.lrAxis4 + PI2;
154. END\_IF
155. *//机器人本体各轴位移*
156. FN\_RPRR\_SC10\_IK.lrAxis1:=FN\_RPRR\_SC10\_IK.lrAxis1\*Rad2Deg;
157. *//FN\_RPRR\_SC10\_IK.lrAxis2:=FN\_RPRR\_SC10\_IK.lrAxis2;*
158. FN\_RPRR\_SC10\_IK.lrAxis3:=FN\_RPRR\_SC10\_IK.lrAxis3\*Rad2Deg;
159. FN\_RPRR\_SC10\_IK.lrAxis4:=FN\_RPRR\_SC10\_IK.lrAxis4\*Rad2Deg;
160. FN\_RPRR\_SC10\_IK.lrAxis5:=0;
161. FN\_RPRR\_SC10\_IK.lrAxis6:=0;
162. *//附加轴各轴位移*
163. FN\_RPRR\_SC10\_IK.lrAxis7 := cCoordPoint.Aux[0];
164. FN\_RPRR\_SC10\_IK.lrAxis8 := cCoordPoint.Aux[1];
165. FN\_RPRR\_SC10\_IK.lrAxis9 := cCoordPoint.Aux[2];