

第五周作业

白梓彤 19374128

讨论³自由度机器人的姿态逆运动学解算。已知固定参数为 $L_1 = 4, L_2 = 3, L_3 = 2$

```
clc, clear;
% 参数:
L1 = 4; L2 = 3; L3 = 2;
```

求3R机器人的逆运动学解

3R机器人的逆运动学解已被打包封装成函数invkine，具体求解过程见invkine.mlx文件

下面带入题目中的四个具体数值进行计算，并通过Corke MATLAB Robotics工具箱的正运动学解函数进行计算。

(1) 数值1

$${}^0_H T = \begin{bmatrix} 1 & 0 & 0 & 9 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
T0toH = [
    1    0    0    9
    0    1    0    0
    0    0    1    0
    0    0    0    1
];
% 逆运动学解，具体过程见函数invkine
tt = invkine(T0toH);
tt % 各个关节的角度，每一行是一组解
```

```
tt = 2x3
    0    0    0
    0    0    0
```

下面通过正运动学计算，来验证逆运动学的求解

```
% DH = [ THETA D  A  ALPHA]
link1 = Link([0, 0, 0, 0], 'modified');
link2 = Link([0, 0, L1, 0], 'modified');
link3 = Link([0, 0, L2, 0], 'modified');
linkH = Link([0, 0, L3, 0], 'modified');
bot = SerialLink([link1 link2 link3 linkH], 'name', 'T0toH')
```

```
bot =
```

```
T0toH:: 4 axis, RRRR, modDH, slowRNE
```

j	theta	d	a	alpha	offset
1	q1	0	0	0	0
2	q2	0	4	0	0
3	q3	0	3	0	0
4	q4	0	2	0	0

+---+-----+-----+-----+-----+-----+-----+

```
bot.fkine([tt(1,1), tt(1,2), tt(1,3), 0])
```

```
ans =
    1    0    0    9
    0    1    0    0
    0    0    1    0
    0    0    0    1
```

```
bot.fkine([tt(2,1), tt(2,2), tt(2,3), 0])
```

```
ans =
    1    0    0    9
    0    1    0    0
    0    0    1    0
    0    0    0    1
```

可以看出，将逆运动学解出来的关节角度带入到正运动学计算中，可以得到原来的 ${}^0_H T$

(2) 数值2

$${}^0_H T = \begin{bmatrix} 0.5 & -0.866 & 0 & 7.5373 \\ 0.866 & 0.5 & 0 & 3.9266 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
T0toH = [
    0.5    -0.866    0    7.5373
    0.866    0.5    0    3.9266
    0        0        1    0
    0        0        0    1
];
```

% 逆运动学解，具体过程见函数invkine

```
tt = invkine(T0toH);
```

tt % 各个关节的角度，每一行是一组解

```
tt = 2x3
    0.1745    0.3491    0.5236
    0.4732   -0.3491    0.9230
```

下面通过正运动学计算，来验证逆运动学的求解

```
% DH = [ THETA D  A  ALPHA]
link1 = Link([0, 0, 0, 0], 'modified');
link2 = Link([0, 0, L1, 0], 'modified');
link3 = Link([0, 0, L2, 0], 'modified');
linkH = Link([0, 0, L3, 0], 'modified');
bot = SerialLink([link1 link2 link3 linkH], 'name', 'T0toH')
```

```
bot =
```

T0toH:: 4 axis, RRRR, modDH, slowRNE

j	theta	d	a	alpha	offset
1	q1	0	0	0	0
2	q2	0	4	0	0
3	q3	0	3	0	0
4	q4	0	2	0	0

```
bot.fkine([tt(1,1), tt(1,2), tt(1,3), 0])
```

```
ans =
    0.5000    -0.8660         0     7.537
    0.8660     0.5000         0     3.927
         0         0         1         0
         0         0         0         1
```

```
bot.fkine([tt(2,1), tt(2,2), tt(2,3), 0])
```

```
ans =
    0.5000    -0.8660         0     7.537
    0.8660     0.5000         0     3.927
         0         0         1         0
         0         0         0         1
```

可以看出，将逆运动学解出来的关节角度带入到正运动学计算中，可以得到原来的 ${}^0_H T$

(3) 数值3

$${}^0_H T = \begin{bmatrix} 0 & 1 & 0 & -3 \\ -1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
T0toH = [
    0    1    0   -3
   -1    0    0    2
    0    0    1    0
    0    0    0    1
];
% 逆运动学解，具体过程见函数invkine
tt = invkine(T0toH);
tt % 各个关节的角度，每一行是一组解
```

```
tt = 2x3
    1.5708    1.5708   -4.7124
    2.8578   -1.5708   -2.8578
```

下面通过正运动学计算，来验证逆运动学的求解

```
% DH = [ THETA D A ALPHA]
```

```
link1 = Link([0, 0, 0, 0], 'modified');
link2 = Link([0, 0, L1, 0], 'modified');
link3 = Link([0, 0, L2, 0], 'modified');
linkH = Link([0, 0, L3, 0], 'modified');
bot = SerialLink([link1 link2 link3 linkH], 'name', 'T0toH')
```

```
bot =
```

```
T0toH:: 4 axis, RRRR, modDH, slowRNE
```

j	theta	d	a	alpha	offset
1	q1	0	0	0	0
2	q2	0	4	0	0
3	q3	0	3	0	0
4	q4	0	2	0	0

```
bot.fkine([tt(1,1), tt(1,2), tt(1,3), 0])
```

```
ans =
```

```

0      1      0     -3
-1     0      0      2
0      0      1      0
0      0      0      1
```

```
bot.fkine([tt(2,1), tt(2,2), tt(2,3), 0])
```

```
ans =
```

```

0      1      0     -3
-1     0      0      2
0      0      1      0
0      0      0      1
```

可以看出，将逆运动学解出来的关节角度带入到正运动学计算中，可以得到原来的 ${}^0_H T$

(4) 数值4

$${}^0_H T = \begin{bmatrix} 0.866 & -0.5 & 0 & -3.1245 \\ -0.5 & 0.866 & 0 & 8.1674 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
T0toH = [
    0.866    -0.5     0    -3.1245
   -0.5     0.866     0     8.1674
    0         0        1        0
    0         0        0        1
];
```

```
% 逆运动学解，具体过程见函数invkine
tt = invkine(T0toH);
```

错误使用 atan2
输入必须为实数。

出错 invkine (line 20)
theta2_1 = atan2(s2_1, c2);

无法求解，说明此位置在机械臂的可达空间之外

应用**Corke MATLAB**机器人工具箱验证所有结果

使用工具箱建立机械臂的模型

```
clc, clear
L1 = 4; L2 = 3; L3 = 2;
% DH = [ THETA D A ALPHA]
link1 = Link([0, 0, 0, 0], 'modified');
link2 = Link([0, 0, L1, 0], 'modified');
link3 = Link([0, 0, L2, 0], 'modified');
bot = SerialLink([link1 link2 link3], 'name', 'T0to3')
```

bot =

T0to3:: 3 axis, RRR, modDH, slowRNE

j	theta	d	a	alpha	offset
1	q1	0	0	0	0
2	q2	0	4	0	0
3	q3	0	3	0	0

```
T0toH = [
    1  0  0  9
    0  1  0  0
    0  0  1  0
    0  0  0  1
];
THto3 = [
    1  0  0 -L3
    0  1  0  0
    0  0  1  0
    0  0  0  1
];
T0to3 = T0toH * THto3;
bot.ikine(T0to3, 'mask', [1 1 0 0 0 1])
```

ans = 1x3
0 0 0

```
T0toH = [
    0.5    -0.866  0  7.5373
    0.866  0.5    0  3.9266
    0      0      1  0
    0      0      0  1
```

```

];
THto3 = [
    1    0    0   -L3
    0    1    0    0
    0    0    1    0
    0    0    0    1
];
T0to3 = T0toH * THto3;
bot.ikine(T0to3, 'mask', [1 1 0 0 0 1])

```

```

ans = 1x3
    0.1745    0.3491    0.5236

```

```

T0toH = [
    0    1    0   -3
   -1    0    0    2
    0    0    1    0
    0    0    0    1
];
THto3 = [
    1    0    0   -L3
    0    1    0    0
    0    0    1    0
    0    0    0    1
];
T0to3 = T0toH * THto3;
bot.ikine(T0to3, 'mask', [1 1 0 0 0 1])

```

```

ans = 1x3
    2.8578   -1.5708   -2.8578

```

可以看出，这些通过工具箱求出的值和自己编写的程序求出的值相同。