

第四周MATLAB练习

白梓彤 19374128

[建议直接打开实时脚本main.mlx查看](#)

本练习主要讨论平面3自由度、3R机器人的D-H参数和正向（位置和姿态）运动学变换方程。已知参数： $L_1 = 4, L_2 = 3, L_3 = 2$ （米）。

a. 求D-H参数。

为了方便计算以及代码的简洁性、可移植性，我定义了一个mylink连杆类，具体内容见mylink.m。

在这个连杆类中，包含了五个成员变量：

- alpha（连杆扭转角）
- a（连杆长度）
- d（连杆偏距）
- theta（关节角）
- isrevolute（是转动关节还是移动关节）

其中前四个成员变量为连杆的D-H参数，最后一个变量用来表示连杆的类型。

通过观察，利用已知参数 $L_1 = 4, L_2 = 3, L_3 = 2$ 可以得到三个连杆的D-H参数。利用这些参数，创建三个连杆类对象。其中转动关节的关节角以及移动关节的连杆偏距在初始化时设为0。

```
clear, clc;
% 已知参数
l1 = 4;
l2 = 3;
l3 = 2;

%          alpha  a    d  theta  isrevolute
L1 = mylink(0,    0,    0,    0,    1);
L2 = mylink(0,    l1,    0,    0,    1);
L3 = mylink(0,    l2,    0,    0,    1);
```

b. 推导相邻的齐次变换矩阵以及常量矩阵 3T_H

在连杆类mylink中，定义了方法transMatrix用于计算每一个连杆上的齐次变换矩阵，具体内容见mylink.m。

这里以每一个关节都旋转60°为例，演示这个功能：

```
L1.transMatrix(pi/3)
```

```
ans = 4x4
    0.5000    -0.8660         0         0
    0.8660     0.5000         0         0
         0         0     1.0000         0
         0         0         0     1.0000
```

```
L2.transMatrix(pi/3)
```

```
ans = 4x4
    0.5000    -0.8660         0     4.0000
```

```

0.8660    0.5000         0         0
         0         0    1.0000         0
         0         0         0    1.0000

```

```
L3.transMatrix(pi/3)
```

```

ans = 4x4
    0.5000   -0.8660         0    3.0000
    0.8660    0.5000         0         0
         0         0    1.0000         0
         0         0         0    1.0000

```

由于坐标系 $\{H\}$ 的原点在夹爪手指的中心且姿态与 $\{3\}$ 相同，所以常量矩阵 ${}^3_H T$ 为：

```

T_3toH = [ 1 0 0 13
           0 1 0 0
           0 0 1 0
           0 0 0 1]

```

```

T_3toH = 4x4
    1     0     0     2
    0     1     0     0
    0     0     1     0
    0     0     0     1

```

c. 求正运动学解的符号解

由于学校的MATLAB中没有symbolic math toolbox，无法进行符号运算

因此对于此题，我使用了mathematica软件进行解决

具体程序见文件“正运动学解.nb”，可以通过Wolfram Mathematica或者Wolfram Player将其打开运行

或者打开“正运动学解.pdf”文件查看

d. 使用Corke MATLAB Robotics工具箱验证计算结果

使用工具箱中的Link()、robot()和fkine()验证上述结果。

首先通过Link()定义三个连杆对象，需要注意的是，课本上使用的D-H参数在Corke MATLAB Robotics工具箱不属于“standard D&H parameters”而是算作“modified D&H parameters”，因此在定义类对象时，需要加上'modified'参数

```

clear;
l1 = 4; l2 = 3; l3 = 2;
% DH = [ THETA D  A  ALPHA]
L1 = Link([0, 0, 0, 0], 'modified');
L2 = Link([0, 0, l1, 0], 'modified');
L3 = Link([0, 0, l2, 0], 'modified');

```

然后计算他们的齐次变换矩阵

```
L1.A(pi/3)
```

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

L2.A(pi/3)

```
ans =
    0.5000   -0.8660         0         4
    0.8660    0.5000         0         0
         0         0         1         0
         0         0         0         1
```

L3.A(pi/3)

```
ans =
    0.5000   -0.8660         0         3
    0.8660    0.5000         0         0
         0         0         1         0
         0         0         0         1
```

常量矩阵 3T 为：

```
LH = Link([0, 0, 13, 0], 'modified');
LH.A(0)
```

```
ans =
     1         0         0         2
     0         1         0         0
     0         0         1         0
     0         0         0         1
```

通过SerialLink()类计算正运动学解

首先计算0到3的正运动学解：

```
bot0to3 = SerialLink([L1 L2 L3], 'name', 'T0to3')
```

```
bot0to3 =
```

```
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

```
bot0to3.fkine([0 0 0].*pi./180)
```

```
ans =
    1    0    0    7
    0    1    0    0
    0    0    1    0
    0    0    0    1
```

```
bot0to3.fkine([10 20 30].*pi./180)
```

```
ans =
    0.5000   -0.8660    0    6.537
    0.8660    0.5000    0    2.195
    0         0        1    0
    0         0        0    1
```

```
bot0to3.fkine([90 90 90].*pi./180)
```

```
ans =
    0    1    0   -3
   -1    0    0    4
    0    0    1    0
    0    0    0    1
```

然后计算0到H的正运动学解

```
bot0toH = SerialLink([L1 L2 L3 LH], 'name', 'T0toH')
```

```
bot0toH =
```

```
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

```
bot0toH.fkine([0 0 0 0].*pi./180)
```

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

```
bot0toH.fkine([10 20 30 0].*pi./180)
```

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

0 0 0 1

```
bot0toH.fkine([90 90 90 0].*pi./180)
```

ans =

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

这些通过Corke MATLAB Robotics工具箱计算出来的结果与自己编写的代码（包括mathematica中的代码）的结果一致。