双足移动机器人技术及强化实践 实验4

3210105327 万晨阳

1. **正运动学计算实现**

我通过编写python脚本完成对于正运动学求解器的实现，完成了对于给定的输入DH参数给出末端执行器的姿态（以xyz欧拉角表示结果）。以下为脚本中主要功能的实现函数的介绍。

* **从DH参数得到齐次变换矩阵**

对于我们给出的一个关节的DH参数，我们能够得到相邻的两个坐标系之间的变换矩阵。函数DH2T接收一组DH参数并返回齐次变换矩阵。

1. # get the transformation matrix of the i-th joint with DH parameters
2. **def** DH2T(a, alpha, d, theta):
3. **return** np.array([[np.cos(theta), -np.cos(alpha) \* np.sin(theta), np.sin(alpha) \* np.sin(theta), a \* np.cos(theta)],
4. [np.sin(theta), np.cos(alpha) \* np.cos(theta), -
5. np.sin(alpha) \* np.cos(theta), a \* np.sin(theta)],
6. [0, np.sin(alpha), np.cos(alpha), d],
7. [0, 0, 0, 1]])

* **链式法则右乘得到最终的齐次变换矩阵**

得到了两个坐标系之间的变换矩阵之后，我们通过连乘就能得到从世界坐标系到末端执行器坐标系的变换矩阵。下面的函数getT接受的是DH参数数组，返回的是最终的变换矩阵。

1. **def** getT(a, alpha, d, theta):
2. T = np.eye(4)
3. **for** i **in** range(len(a)):
4. tmp = DH2T(a[i], alpha[i], d[i], theta[i])
5. T = T @ tmp
6. **return** T

* **通过解算最终的变换矩阵得到欧拉角**

得到了最终的齐次坐标变换矩阵之后，我们希望解算得到欧拉角。下面的函数实现了这一功能。

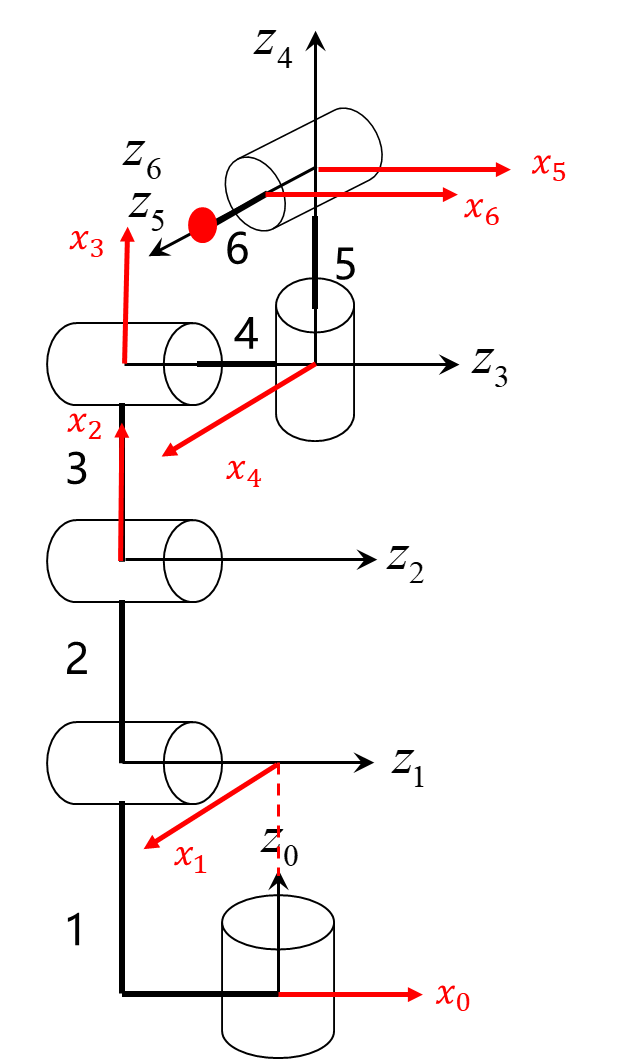
1. **def** solve(T):
2. x = np.arctan2(-T[1, 2], T[2, 2])/np.pi\*180
3. y = np.arcsin(T[0, 2])/np.pi\*180
4. z = np.arctan2(-T[0, 1], T[0, 0])/np.pi\*180
5. euler = np.array([x, y, z])
6. **return** euler

* **调用函数进行计算**

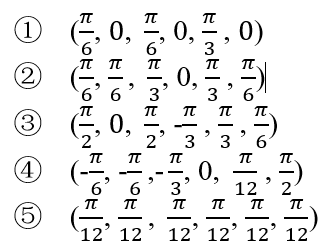
在main函数中我们输入DH参数，进行正向运动学的求解。

1. theta\_list = [np.array([np.pi/6, 0, np.pi/6, 0, np.pi/3, 0]),
2. np.array([np.pi/6, np.pi/6, np.pi/3, 0, np.pi/3, np.pi/6]),
3. np.array([np.pi/6, np.pi/6, np.pi/3,
4. np.pi/6, np.pi/3, np.pi/6]),
5. np.array([np.pi/2, 0, np.pi/2, -np.pi/3, np.pi/3, np.pi/6]),
6. np.array([-np.pi/6, -np.pi/6, -np.pi /
7. 3, 0, np.pi/12, np.pi/2]),
8. np.array([np.pi/12, np.pi/12, np.pi/12, np.pi/12, np.pi/12, np.pi/12])]
10. a = [0, 185, 170, 0, 0, 0]
11. alpha = [-np.pi/2, 0, 0, np.pi/2, np.pi/2, 0]
12. d = [230, -54, 0, 77, 77, 85.5]
13. init\_theta = [0, -np.pi/2, 0, np.pi/2, np.pi/2, 0]
15. **for** theta **in** theta\_list:
16. **for** i **in** range(len(theta)):
17. theta[i] += init\_theta[i]
18. T = getT(a, alpha, d, theta)
19. **print**("Theta: ", theta \* 180 / np.pi)
20. **print**("T: \n", T)
21. res = solve(T)
22. **print**("Position: ", T[:3, 3]/1000)  # convert to meter
23. **print**("Euler Angle: ", res)
24. **print**("\n")

我在上次任务中建立的坐标系以及对应的DH参数表如下图：



将以下5组关节角参数带入正运动学解，计算机械臂末端点的空间位置



结果如下，其中Position = [x, y, z]，单位为米，与仿真环境中一致；欧拉角（xyz）为角度而非弧度。

①

Theta: [ 30. -90. 30. 90. 150. 0.]

T:

[[-8.99519053e-01 4.33012702e-01 -5.80127019e-02 9.04940514e+01]

[ 5.80127019e-02 2.50000000e-01 9.66506351e-01 1.64304877e+02]

[ 4.33012702e-01 8.66025404e-01 -2.50000000e-01 6.07533275e+02]

[ 0.00000000e+00 0.00000000e+00 0.00000000e+00 1.00000000e+00]]

**Position: [0.09049405 0.16430488 0.60753327]**

**Euler Angle: [-104.50247052 -3.32575021 -154.29465179]**

②

Theta: [ 30. -60. 60. 90. 150. 30.]

T:

[[ 2.16506351e-01 8.75000000e-01 -4.33012702e-01 2.45493039e+02]

[ 6.25000000e-01 2.16506351e-01 7.50000000e-01 2.53793584e+02]

[ 7.50000000e-01 -4.33012702e-01 -5.00000000e-01 3.47464700e+02]

[ 0.00000000e+00 0.00000000e+00 0.00000000e+00 1.00000000e+00]]

**Position: [0.24549304 0.25379358 0.3474647 ]**

**Euler Angle: [-123.69006753 -25.65890627 -76.10211375]**

③

Theta: [ 30. -60. 60. 120. 150. 30.]

T:

[[ 4.83253175e-01 5.87019053e-01 -6.49519053e-01 2.18047789e+02]

[ 7.79006351e-01 5.02404736e-02 6.25000000e-01 2.37948062e+02]

[ 3.99519053e-01 -8.08012702e-01 -4.33012702e-01 3.14692114e+02]

[ 0.00000000e+00 0.00000000e+00 0.00000000e+00 1.00000000e+00]]

**Position: [0.21804779 0.23794806 0.31469211]**

**Euler Angle: [-124.71500395 -40.50535033 -50.53767779]**

④

Theta: [ 90. -90. 90. 30. 150. 30.]

T:

[[-4.33012702e-01 2.50000000e-01 -8.66025404e-01 -9.70451720e+01]

[-3.99519053e-01 8.08012702e-01 4.33012702e-01 2.45522586e+02]

[ 8.08012702e-01 5.33493649e-01 -2.50000000e-01 4.60308956e+02]

[ 0.00000000e+00 0.00000000e+00 0.00000000e+00 1.00000000e+00]]

**Position: [-0.09704517 0.24552259 0.46030896]**

**Euler Angle: [-120. -60. -150.]**

⑤

Theta: [ -30. -120. -60. 90. 105. 90.]

T:

[[-8.66025404e-01 -4.82962913e-01 1.29409523e-01 -2.71451110e+02]

[ 5.00000000e-01 -8.36516304e-01 2.24143868e-01 2.08832885e+02]

[ 2.22603388e-16 2.58819045e-01 9.65925826e-01 4.72801358e+02]

[ 0.00000000e+00 0.00000000e+00 0.00000000e+00 1.00000000e+00]]

**Position: [-0.27145111 0.20883289 0.47280136]**

**Euler Angle: [-13.06431343 7.43547223 150.85257374]**

⑥

Theta: [ 15. -75. 15. 105. 105. 15.]

T:

[[-2.35457937e-01 7.70197545e-01 5.92752310e-01 2.25673158e+02]

[ 9.02835062e-01 -5.24452349e-02 4.26776695e-01 1.07189948e+02]

[ 3.59789397e-01 6.35645529e-01 -6.83012702e-01 5.51970233e+02]

[ 0.00000000e+00 0.00000000e+00 0.00000000e+00 1.00000000e+00]]

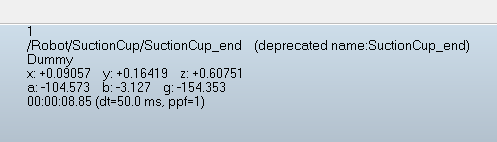
**Position: [0.22567316 0.10718995 0.55197023]**

**Euler Angle: [-148.00102617 36.35256472 -106.99897383]**

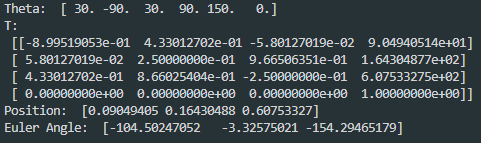
1. **仿真环境验证**

①

仿真结果：

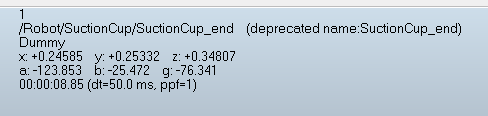


计算结果：

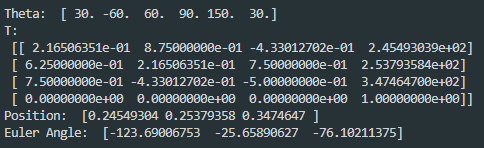


②

仿真结果：

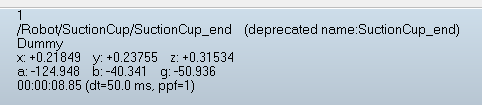


计算结果：

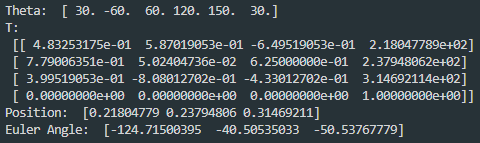


③

仿真结果：

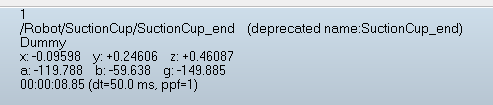


计算结果：

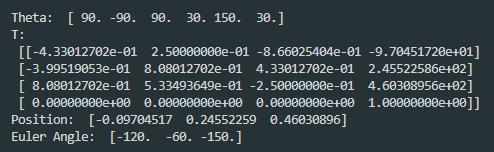


④

仿真结果：

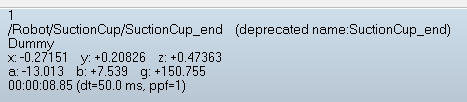


计算结果：

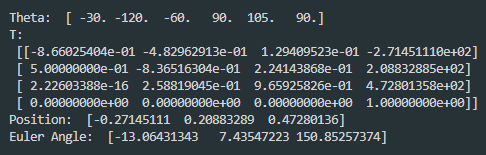


⑤

仿真结果：

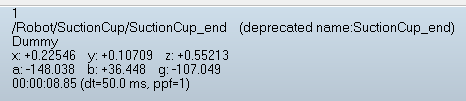


计算结果：

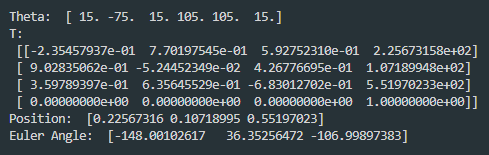


⑥

仿真结果：



计算结果：



可以看到仿真环境中的结果与我们的计算结果基本一致，说明正运动学求解器的正确性得到了验证。