双足移动机器人技术及强化实践 实验报告

组号：2

成员：万晨阳（3210105327）邵可乐（3210100786）傅凌云（3210103452）

一 、 实验目的和要求

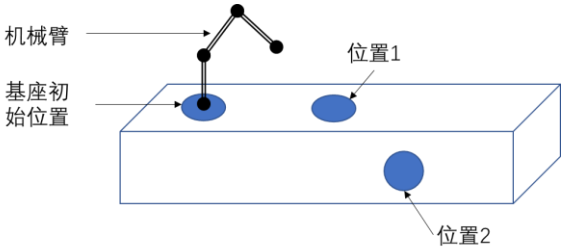
1.1 实验目的：

掌握并灵活运用关节型机械臂运动控制相关理论和知识，解决一般构型机械臂的运动控制问题。

1.2 实验要求：

以中国空间站太空机械臂为仿真对象，设计开发机械臂控制算法和程序，在仿真环境中演示该机械臂在空间站上移动和行走。

中国空间站机械臂



机械臂移动目标位置意图

(1) 根据缩小版的机械臂结构和尺寸，利用 DH 方法**正运动学建模**；

(2) 分析并给出该机械臂的**逆运动学解析解**；

(3) 采用一种**轨迹规划方法**，控制该机械臂在空间站上行走，至少行走 2 步，即从初始位置移动到位置 1，再移动到位置 2，运动时速度在合理范围内自定义，保证速度、加速度连续；

(4) 假设机械臂末端于基座之间接触后可以通过电磁铁紧密吸合，借助仿真环境进行**可视化演示**。

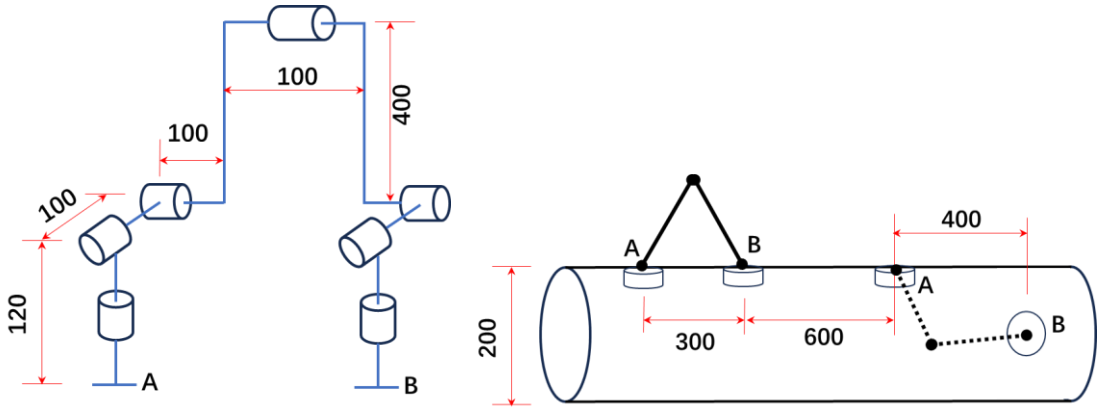
二、主要仪器：

在仿真实验部分中，我们借助 Coppeliasim 仿真环境进行运动的模拟；在实物实验部分，我们基于实验室提供的工具进行模型构建和运动控制。

三、仿真实验部分

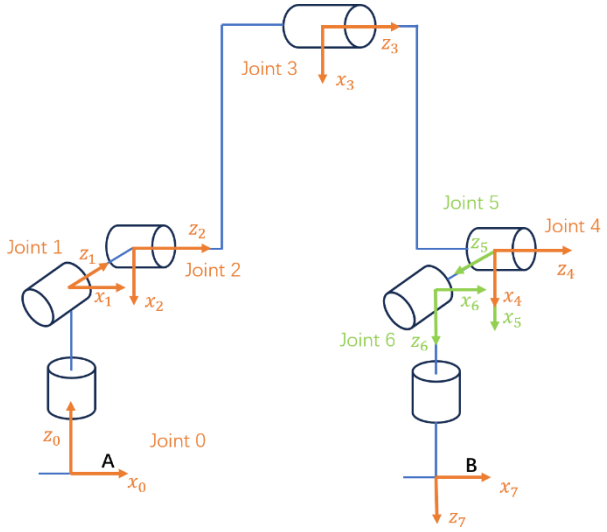
3.1 机械臂的 DH 参数建模

我们仿真和实际使用的机械臂的结构如下：



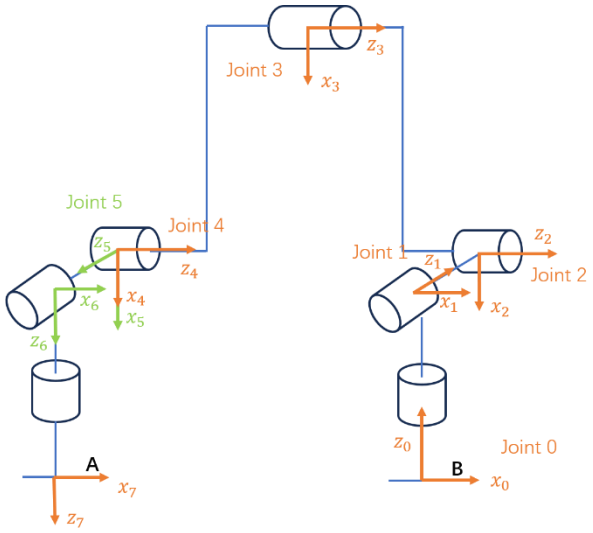
在运动过程中，我们需要转换基座（保持 A 端不动移动 B 端，或者保持 B 端不动移动 A 端），所以为了方便运动的解算，我们分别完成了以 A 为基底和以 B 为基底机械臂的 DH 参数建模。

以 A 为基座：



$No.$	a_i	α_i	d_i	θ_i
1	0	-90°	120mm	θ_1
2	0	90°	100mm	$\theta_2 + 90°$
3	-400mm	0	150mm	θ_3
4	400mm	0	150mm	θ_4
5	0	90°	0	θ_5
6	0	90°	100mm	$\theta_6 + 90°$
7	0	0	120mm	θ_7

以 B 为基座，由于只有坐标系 2、3、4 之间有相对位置的变化，故只需要将d_3, d_4取反。



$No.$	a_i	α_i	d_i	θ_i
1	0	-90°	120mm	θ_1
2	0	90°	100mm	$\theta_2 + 90°$
3	-400mm	0	-150mm	θ_3
4	400mm	0	-150mm	θ_4
5	0	90°	0	θ_5
6	0	90°	100mm	$\theta_6 + 90°$
7	0	0	120mm	θ_7

3.2 正运动学求解与推导过程

本实验中正运动学作为下述逆运动学求解得到的关节角的检查，检查其能否对应到正确的世界坐标系位姿。正运动学分为下述两步：

(1) 齐次变换矩阵求解

利用公式：

$$ {}^{i-1}T_i = \begin{pmatrix} c\theta & -c\alpha s\theta & s\alpha s\theta & ac\theta \\ s\theta & c\alpha c\theta & -s\alpha c\theta & as\theta \\ 0 & s\alpha & c\alpha & d \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

$$ T = \prod^{i-1} T_i $$

即可利用不同关节的 DH 参数与关节角得到整个机械臂的齐次　变换矩阵（最终位姿）。

(2) 反解 RPY 方位角，并与 T 中的位置合并

用下表进行方位角反解：

	$\psi(Roll)$	$\phi(Pitch)$	$\theta(Yall)$
$\phi \neq \frac{\pi}{2}$	$-tan^{-1}\frac{R_{23}}{R_{33}}$	$\pi^{\delta}+(-)^{\delta}$ $sin^{-1}R_{13}$	$-tan^{-1}\frac{R_{12}}{R_{11}}$
$\phi=\frac{\pi}{2}$	0	$\pm\frac{\pi}{2}$	$-tan^{-1}\frac{R_{21}}{R_{32}}$

3.3 逆运动学解析解推导过程

根据机械臂运动特征，我们可以假设第二个关节角 $\theta_2 = 0$ 。基于逆运动学基本公式：

$$ {}^{0}T_{7} = {}^{0}T_{1} \cdot {}^{1}T_{2} \cdot {}^{2}T_{3} \cdot {}^{3}T_{4} \cdot {}^{4}T_{5} \cdot {}^{5}T_{6} \cdot {}^{6}T_{7} $$

利用矩阵的最终位姿表示与关节角表示相等即可得到最终的解析解

注意到：

$$ {}^{0}T_{7} = \begin{pmatrix} n_x & o_x & a_x & d_x \\ n_y & o_y & a_y & d_y \\ n_z & o_z & a_z & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

$$ {}^{1}T_{7} = {}^{1}T_{2} \cdot {}^{2}T_{3} \cdot {}^{3}T_{4} \cdot {}^{4}T_{5} \cdot {}^{5}T_{6} \cdot {}^{6}T_{7} = {}^{0}T_{1}^{-1} \cdot {}^{0}T_{7} $$

取第(3,1)(3,2)(3,3)号元素相等,得到等式：

$$ \begin{cases} a_y c_1 + a_x s_1 = s_{345} + s_6 \\ n_y c_1 + n_x s_1 = s_{345} c_6 c_7 - c_{345} s_7 \\ o_y c_1 + o_x s_1 = -s_{345} c_6 s_7 - c_{345} c_7 \end{cases} $$

对三个式子取平方和，利用下述等式可以得到 θ_1 ：

$$ \left(n_y^2 + o_y^2 + a_y^2\right)c_1^2 + (n_x^2 + o_x^2 + a_x^2)s_1^2 - 2\left(n_x n_y + o_x o_y + a_x a_y\right)s_1 c_1 = 1 $$

注意到；

$$ {}^{1}T_{5} = {}^{1}T_{2} \cdot {}^{2}T_{3} \cdot {}^{3}T_{4} \cdot {}^{4}T_{5} = {}^{0}T_{1}^{-1} \cdot {}^{0}T_{7} \cdot {}^{6}T_{7}^{-1} \cdot {}^{5}T_{6}^{-1} $$

取第(2,2) (2,3)号元素相等可以得到等式：

$$ \begin{cases} o_z c_6 s_7 - n_z c_6 c_7 - a_z s_6 = 0 \\ o_z c_7 + n_z s_7 = 0 \end{cases} $$

可以得到：

$$ \begin{cases} \theta_7 = -atan\dfrac{o_z}{n_z} \\ \theta_6 = \dfrac{o_z s_7 - n_z c_7}{a_z} \end{cases} $$

取第(1,1)号元素相等可以得到等式：

$$ a_x c_1 s_6 + a_y s_1 s_6 + n_x c_1 c_6 c_7 + n_y c_6 c_7 s_1 - o_x c_1 c_6 s_7 - o_y c_6 s_1 s_7 = c_{345} $$

即可得到：

$$\theta_3+\theta_4+\theta_5=acos\left(a_xc_1s_6+a_ys_1s_6+n_xc_1c_6c_7+n_yc_6c_7s_1-o_xc_1c_6s_7-o_yc_6s_1s_7\right)$$

利用公式：

$${}^1T_4={}^1T_2\cdot{}^2T_3\cdot{}^3T_4={}^0T_1^{-1}\cdot{}^0T_7\cdot{}^6T_7^{-1}\cdot{}^5T_6^{-1}\cdot{}^4T_5^{-1}$$

取第(2,1)号元素相等可以得到：

$$\theta_5=atan\left(\frac{o_zc_6s_7-n_zc_6c_7-a_zs_6}{o_zc_7+n_zs_7}\right)$$

(2,4)号元素相等可以得到等式：

$$d_yc_1-d_xs_1+a_xd_7s_1-a_yd_7c_1+d_6n_xs_1s_7-d_6o_yc_1c_7-d_6n_yc_1s_7+d_6o_xc_7s_1$$
$$=a_4s_{34}+a_3s_3$$

配合前述 θ_5 与 $\theta_3+\theta_4+\theta_5$ 的值即可求得 θ_3， θ_4，进而求得所有解析解。

3.4 轨迹规划设计

本实验中轨迹规划采用五次多项式规划方法，轨迹规划在程序中主要由两个函数实现，分别为 quinticCurvePlanning()，quinticCurveExcute()。

(1) quinticCurvePlanning ()

该函数利用五次多项式：

$$q(t)=a_0+a_1t+a_2t^2+a_3t^3+a_4t^4+a_5t^5$$

与初末态位置进行 rest to rest 的五次多项式规划,得到系数矩阵$\{\boldsymbol{a}_i\}$。

```
def quinticCurvePlanning(startPosition, endPosition, startVelocity, endVelocity, time):
    timeMatrix = np.matrix(
        [
            [0, 0, 0, 0, 0, 1],
            [time**5, time**4, time**3, time**2, time, 1],
            [0, 0, 0, 0, 1, 0],
            [5 * time**4, 4 * time**3, 3 * time**2, 2 * time, 1, 0],
            [0, 0, 0, 2, 0, 0],
            [20 * time**3, 12 * time**2, 6 * time, 2, 0, 0],
        ]
    )
    invTimeMatrix = np.linalg.inv(timeMatrix)
    kArray = []
    for i in range(len(startPosition)):
```

```
        X = np.matrix(
            [startPosition[i], endPosition[i],
                startVelocity[i], endVelocity[i], 0, 0]
        ).T
        k = np.dot(invTimeMatrix, X)
        kArray.append(k)
    return kArray
```

(2) quinticCurveExcute()

该函数利用前述系数矩阵$\{a_i\}$，并输入当前时刻，即可返回得到当前时间各关节角的位置。

```
def quinticCurveExcute(kArray, time):
    timeVector = np.matrix([time**5, time**4, time**3, time**2, time, 1]).T
    jointPositions = []
    for i in range(7):
        jointPosition = np.dot(kArray[i].T, timeVector)
        jointPositions.append(jointPosition[0, 0])
    return np.array(jointPositions)
```

3.5 运动程序代码设计

对于仿真实验，我们实现了一个 rbenv 类，其中定义了注册关节、换底、关节移动等功能。

运动的过程可以简述为以一定时间间隔将各关节角调整为轨迹规划出的关节角，完成一整个动作后需要在仿真环境中换底，然后继续动作。

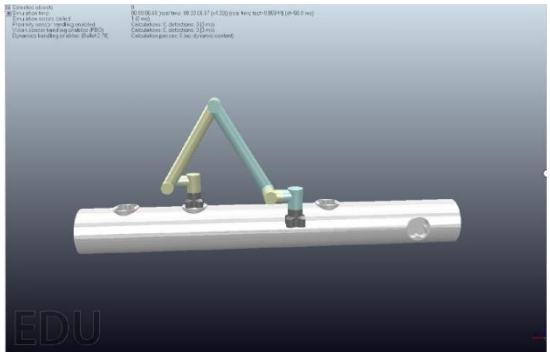
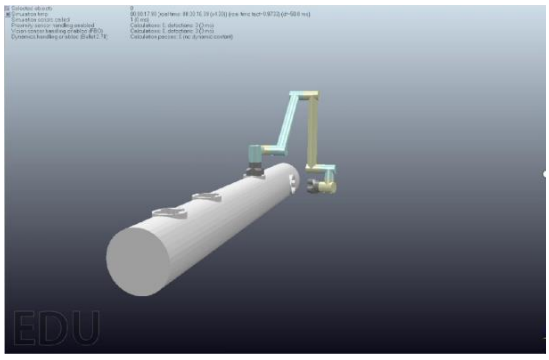
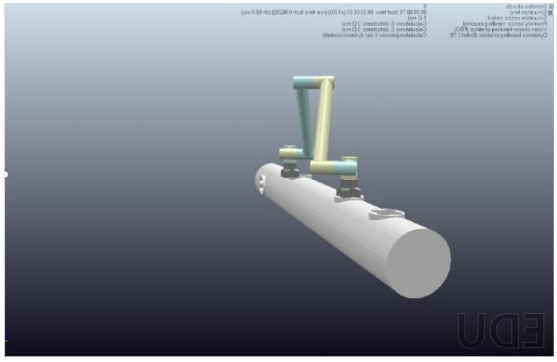
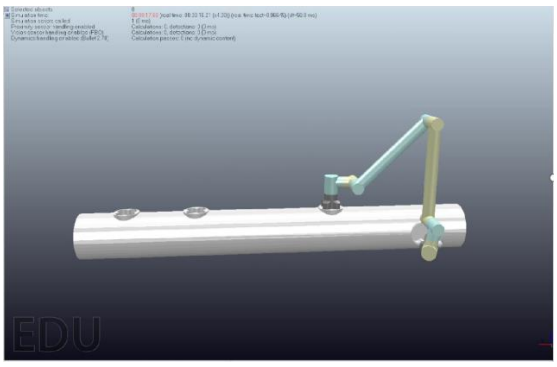
对于换底，在虚拟环境中遇到了许多问题。我们通过在换底后将每个关节的位姿矩阵重新赋值为初始值来确保各关节有正确的 0 状态。

```
def switch_base(self, op):
    if self.base == op:
        return
    pos_now = np.zeros(7)
    for i in range(0, 7):
        pos_now[i] = self.sim.getJointPosition(self.joint[i])

    if op == "A":
        self.sim.setObjectParent(self.l_base, -1)
        for i in range(16, 30):
```

```
            self.sim.setObjectParent(i + 1, i)
            self.sim.setObjectMatrix(i + 1, i, self.matrix_
r[i - 16])
            if i % 2 == 1:      # is joint
                self.sim.setJointPosition(i, -
pos_now[(i - 17) // 2])
        self.base = op
        return
    if op == "B":
        # 略
```

3.6 仿真结果展示与分析

将不同视角的仿真姿态展示如下：

从上述仿真姿态截图中可以看出，该仿真机械臂工作空间较小，仿真效果较好。同时运动过程中没有穿模等现象，运动可行且稳定。

四、实物实验部分

4.1 运动程序代码设计

运动程序主要调用多个函数接口：注册电机 register_motor()，电机使能 servo_enable()，电机回零 return_home_opt()，电机运动 write_position()。

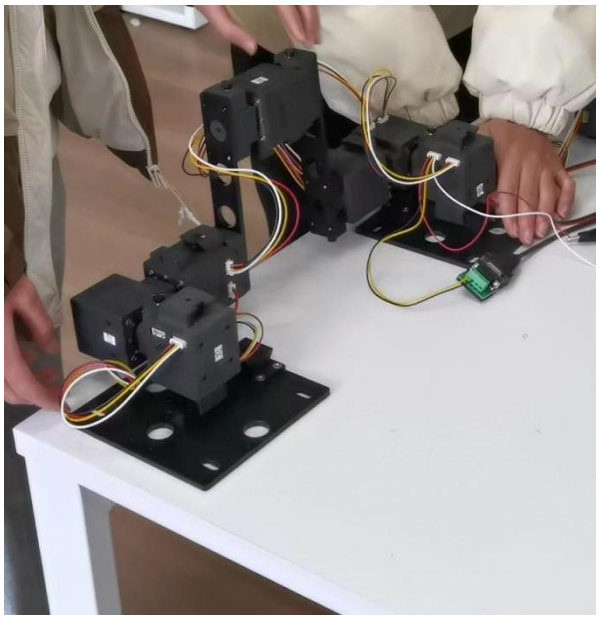
其中电机运动的关节角度是人为设定后由**传感器读数**得出。

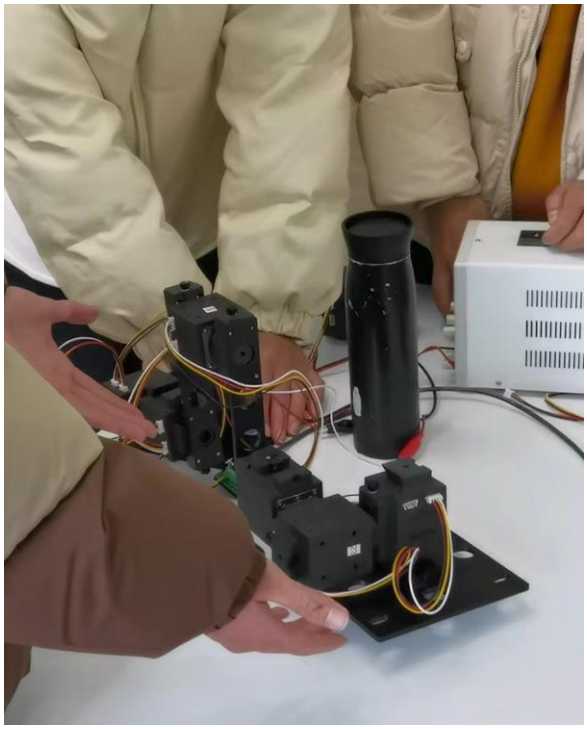
```
if __name__ == "__main__":
    robot = Robot("COM7", 2250000)
    robot.register_motor(56, 51)    # 电机注册
```

```
    robot.register_motor(53, 101)
    robot.register_motor(54, 51)
    robot.register_motor(26, 51)
    robot.register_motor(24, 51)
    robot.register_motor(23, 101)
    robot.register_motor(25, 51)

    pos1 = np.array([0,
                     5.70257358 - 2*np.pi,
                     4.59216324 - 2*np.pi,
                     0.04506069,
                     0.08916263,
                     6.25288919 - 2*np.pi,
                     5.74533329- 2*np.pi])
    pos2 =  # 略
    pos3 =  # 略
    vel = np.ones(7) * 0.3  # 设定角速度

    robot.servo_enable()    #电机使能
    robot.return_home_opt() #回零

    robot.write_position(pos1, vel) # 电机运动
    robot.write_position(pos2, vel)
    robot.write_position(pos3, vel)
```

4.2 实验结果展示与分析

进行实物实验时，我们在轨迹上设置了四个关键姿态，以保证经过规定轨迹，如下图所示：



在实物实验的过程中，我们的机械臂完成了机械臂平行移动和抓手翻转的运动，较好地还原了仿真效果。