

机器人控制系统实验报告

姓 名： 刘瑾瑾
学 号： 20201000128
班 级： 231202
培 养 单 位： 自动化学院

二〇二三年六月

实验四 桌面 SCARA 机械臂正运动学分析实验

1. 实验内容

1.1 实验目的

- (1)熟悉并掌握机器人正运动学的原理
- (2)熟悉并掌握机器人正运动的代码编写和校验
- (3)熟悉并掌握机器人工具箱有哪些常用的相关函数及其功能
- (4)熟悉并掌握通过正运动求解末端位姿的齐次变换矩阵
- (5)熟悉并掌握 MATLAB 的符号运算

1.2 正运动学原理

机械臂正运动学问题就是求机械臂运动学的正解，是指在给定组成运动副的相邻连杆的相对位置情况下，确定机械臂末端执行器的位置和姿态。

1.3 正运动学求解

$$\begin{aligned}
 {}^0_1T &= \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & a_0 \\ \sin\theta_1 \cos\alpha_0 & \cos\theta_1 \cos\alpha_0 & -\sin\alpha_0 & -d_1 \sin\alpha_0 \\ \sin\theta_1 \sin\alpha_0 & \cos\theta_1 \sin\alpha_0 & \cos\alpha_0 & d_1 \cos\alpha_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 {}^1_2T &= \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & a_1 \\ \sin\theta_2 \cos\alpha_1 & \cos\theta_2 \cos\alpha_1 & -\sin\alpha_1 & -d_2 \sin\alpha_1 \\ \sin\theta_2 \sin\alpha_1 & \cos\theta_2 \sin\alpha_1 & \cos\alpha_1 & d_2 \cos\alpha_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_2 & -s_2 & 0 & a_1 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 {}^2_3T &= \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & a_2 \\ \sin\theta_3 \cos\alpha_2 & \cos\theta_3 \cos\alpha_2 & -\sin\alpha_2 & -d_3 \sin\alpha_2 \\ \sin\theta_3 \sin\alpha_2 & \cos\theta_3 \sin\alpha_2 & \cos\alpha_2 & d_3 \cos\alpha_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_3 & -s_3 & 0 & a_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 {}^3_4T &= \begin{bmatrix} \cos\theta_4 & -\sin\theta_4 & 0 & a_3 \\ \sin\theta_4 \cos\alpha_3 & \cos\theta_4 \cos\alpha_3 & -\sin\alpha_3 & -d_4 \sin\alpha_3 \\ \sin\theta_4 \sin\alpha_3 & \cos\theta_4 \sin\alpha_3 & \cos\alpha_3 & d_4 \cos\alpha_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_4 & -s_4 & 0 & a_3 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

矩阵内容代表含义，式中 $n_x n_y n_z$ 表示机械臂末端坐标系的 x 轴在基坐标系中的方向矢量；式中 $o_x o_y o_z$ 表示机械臂末端坐标系的 y 轴在基坐标系中的方向矢量；式中 $a_x a_y a_z$ 表示机械臂末端坐标系的 z 轴在基坐标系中的方向矢量；式中 $p_x p_y p_z$ 表示机械臂末端在基坐标系中的坐标。

$${}^0_4T = {}^0_1T \times {}^1_2T \times {}^2_3T \times {}^3_4T = \begin{bmatrix} c_{234} & -s_{234} & 0 & a_1 + a_2c_2 + a_3c_{23} \\ s_{234} & c_{234} & 0 & a_2s_2 + a_3s_{23} \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0_4T = T_{tool} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{末端位置相对于参考坐标系的齐次变换矩阵})$$

2. 实验步骤与过程

根据提供的各关节的角度，通过 MATLAB 的机器人工具箱和 m 代码的编程，求解末端的位姿，以下为采用 MATLAB 提供的机器人工具箱(Robotics Toolbox)和自主编写的采用以上理论的正运动学解算的两个案例。

2.1 采用 MATLAB 机器人工具箱的正运动学代码仿真

```
1. clear,clc,close;
2. L1=Link('theta',0,'a',0,'alpha',0,'offset',0,'qlim',[0
3. 160],'modified');
4. L2=Link('d',0,'a',127.5,'alpha',0,'offset',0,'qlim',[-90
5. 90]*pi/180,'modified');
6. L3=Link('d',0,'a',160,'alpha',0,'offset',0,'qlim',[-120
7. 120]*pi/180,'modified');
8. L4=Link('d',0,'a',160,'alpha',0,'offset',0,'qlim',[-180
9. 180]*pi/180,'modified');
10. robot = SerialLink([L1 L2 L3 L4],'name','SCARA');
11. Q=[20 30 30 30];
12. forward_Q=[Q(1) 0 0 0]+[0 Q(2) Q(3) Q(4)]/180*pi;
13. forward=robot.fkine(forward_Q);
14. rpy=tr2rpy(forward,'xyz')*180/pi;
15. W=[-1200 +1200 -1200 +1200 -1200 +1200];
16. robot.plot(forward_Q,'workspace',W);
17. robot.teach(forward_Q,'rpy');
```

2.2 采用自主编写的正运动学解算代码

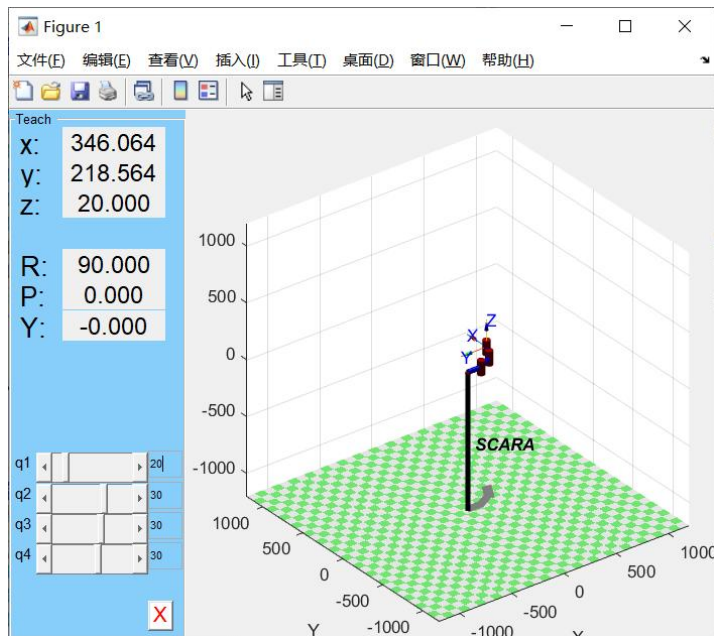
```

1. clear,clc,close;
2. L1=Link('theta',0,'a',0,'alpha',0,'offset',0,'qlim',[0
3. 160],'modified');
4. L2=Link('d',0,'a',127.5,'alpha',0,'offset',0,'qlim',[-90
5. 90]*pi/180,'modified');
6. L3=Link('d',0,'a',160,'alpha',0,'offset',0,'qlim',[-120
7. 120]*pi/180,'modified');
8. L4=Link('d',0,'a',160,'alpha',0,'offset',0,'qlim',[-180
9. 180]*pi/180,'modified');
10. robot = SerialLink([L1 L2 L3 L4],'name','SCARA');
11. Q=[45 45 45 45];
12. forward_Q=[Q(1) 0 0 0]+[0 Q(2) Q(3) Q(4)]/180*pi;
13. [T04,Pos]=Forward_ModifiedDH(forward_Q);
14. rpy=tr2rpy(T04,'xyz')*180/pi;
15. W=[-1200 +1200 -1200 +1200 -1200 +1200];
16. robot.plot(Q,'workspace',W);
17. robot.teach(Q,'rpy');

```

3. 实验结果及分析

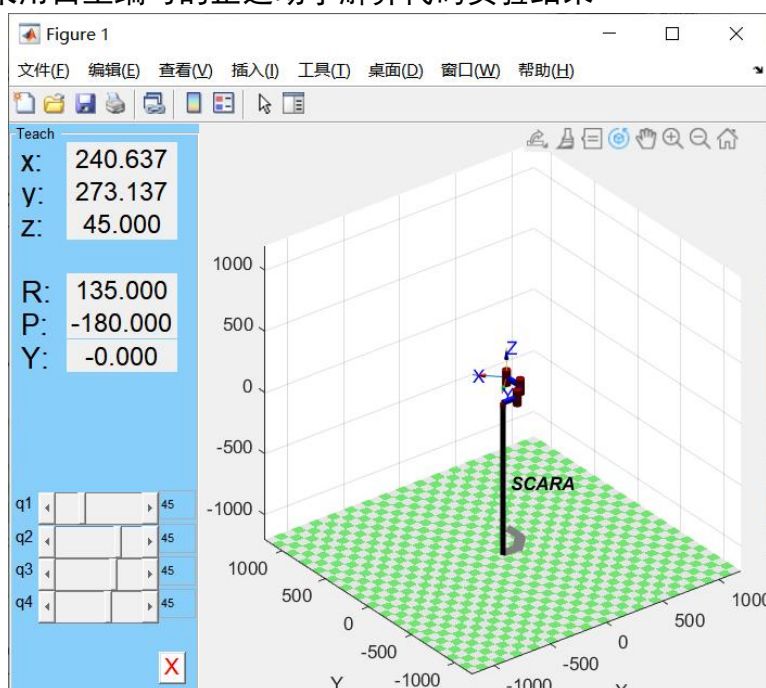
3.1 采用 MATLAB 机器人工具箱的正运动学代码仿真实验结果



求解的齐次矩阵如下：

```
forward =
    0    -1     0   346.1
    1     0     0   218.6
    0     0     1    20
    0     0     0     1
```

3.2 采用自主编写的正运动学解算代码实验结果



求解的齐次矩阵如下：

```
forward =
   -0.7071   -0.7071     0   240.6
    0.7071   -0.7071     0   273.1
         0         0     1    45
         0         0     0     1
```

3.3 实验结果分析

正运动学是通过各关节角度计算机械臂末端位置，比较齐次矩阵的最后一列和仿真位置，由 3.1 和 3.2 的结果可知，仿真结果与程序求解结果相同，说明求解过程正确，为后续实际操作提供基础。

机器人工具箱在机器人建模、轨迹规划、控制、可视化仿真等方面给机器人的研究和学习提供便利条件，大大提高了研究和工作效率。在机器人工具箱

中，类函数非常多，主要涉及到机器人建模、正运动学、逆运动学、动力学、逆动力学、雅克比、轨迹规划等。

对于机器人建模，用到最重要的两个函数就是 `Link` 和 `Seriallink` 函数。利用 MATLAB 中 Robotics Toolbox 工具箱中的 `transl`、`rotx`、`roty` 和 `rotz` 可以实现用齐次变换矩阵表示平移变换和旋转变换。利用 Robotics Toolbox 中的 `fkine` 函数可以实现机器人正运动学的求解。利用 Robotics Toolbox 提供的 `ctrjaj`、`jtraj` 和 `trinterp` 函数可以实现笛卡尔规划、关节空间规划和变换插值。

实验五 桌面 SCARA 机械臂逆运动学分析实验

1. 实验内容

1.1 实验目的

- (1) 熟悉并掌握机器人逆运动学的原理
- (2) 熟悉并掌握机器人逆运动的 M 代码编写和仿真校验
- (3) 熟悉并掌握机器人工具箱有哪些常用的相关函数及其功能
- (4) 熟悉并掌握通过正运动求解末端位姿的齐次变换矩阵
- (5) 熟悉并掌握 MATLAB 的符号运算

1.2 逆运动学原理

正运动学方程都是建立关节变量与末端执行器位置和方向之间的函数关系。逆运动学问题则是由给定的末端执行器位置和方向，确定相对应的关节变量。这一问题的求解具有重要的意义，其目的是将分配给末端执行器在操作空间的运动，变换为相应的关节空间的运动，使得期望的运动能够得到执行。

2. 实验步骤与过程

根据提供的各关节的角度，通过 MATLAB 的机器人工具箱和 m 代码的编程，求解末端的位姿，以下为采用 MATLAB 提供的机器人工具箱(Robotics Toolbox)和自主编写的采用以上理论的逆运动学解算的两个案例。

1.1 采用 MATLAB 机器人工具箱的逆运动学代码仿真

```
1. clear,clc,close;
2. L1=Link('theta',0,'a',0,'alpha',0,'offset',0,'qlim',[0
3. 160],'modified');
4. L2=Link('d',0,'a',127.5,'alpha',0,'offset',0,'qlim',[-90
5. 90]*pi/180,'modified');
6. L3=Link('d',0,'a',160,'alpha',0,'offset',0,'qlim',[-120
7. 120]*pi/180,'modified');
8. L4=Link('d',0,'a',160,'alpha',0,'offset',0,'qlim',[-180
9. 180]*pi/180,'modified');
10. robot = SerialLink([L1 L2 L3 L4],'name','SCARA');
11. % Q=[20 30 30 30];
12. Q=[45 45 45 45];
13. forward_Q=[Q(1) 0 0 0]+[0 Q(2) Q(3) Q(4)]/180*pi;
14. forward=robot.fkine(forward_Q)
15. rpy=tr2rpy(forward, 'xyz')*180/pi;
16. W=[-1200 +1200 -1200 +1200 -1200 +1200];
```

```

17. mask_vector = [1,1,1,1,0,0];
18. axi_val = robot.ikine(forward,'mask',mask_vector,'pinv');
19. Q=[axi_val(1) 0 0 0]+[0 axi_val(2) axi_val(3) axi_val(4)]*180/pi
20. view(3);
21. robot.plot(forward_Q,'workspace',W);
22. robot.teach(forward_Q,'rpy');

```

2.1 采用自主编写的逆运动学解算代码

```

1. clear,clc,close;
2. L1=Link('theta',0,'a',0,'alpha',0,'offset',0,'qlim',[0
3. 160],'modified');
4. L2=Link('d',0,'a',127.5,'alpha',0,'offset',0,'qlim',[-90
5. 90]*pi/180,'modified');
6. L3=Link('d',0,'a',160,'alpha',0,'offset',0,'qlim',[-120
7. 120]*pi/180,'modified');
8. L4=Link('d',0,'a',160,'alpha',0,'offset',0,'qlim',[-180
9. 180]*pi/180,'modified');
10. robot = SerialLink([L1 L2 L3 L4],'name','SCARA');
11. % Q=[20 30 30 30];
12. Q=[45 45 45 45];
13. forward_Q=[Q(1) 0 0 0]+[0 Q(2) Q(3) Q(4)]/180*pi;
14. forward=Forward_ModifiedDH(forward_Q)
15. rpy=tr2rpy(forward,'xyz')*180/pi;
16. W=[-1200 +1200 -1200 +1200 -1200 +1200];
17. mask_vector = [1,1,1,1,0,0];
18. axi_val = ScaraIkinMDH(forward);
19. Q=[axi_val(1) 0 0 0]+[0 axi_val(2) axi_val(3) axi_val(4)]*180/pi
20. view(3);
21. robot.plot(forward_Q,'workspace',W);
22. robot.teach(forward_Q,'rpy');

```

3. 实验结果及分析

3.1 采用 MATLAB 机器人工具箱的逆运动学代码仿真结果

通过实验四 3.1 正运动学求得的齐次矩阵计算各个关节角度。

求解结果如下：

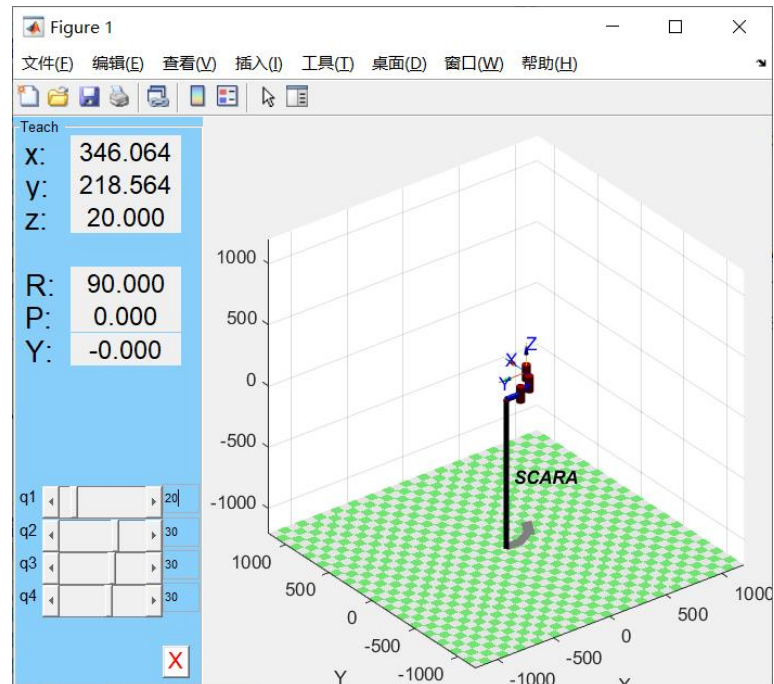
```

Q =
    20.0000    30.0000    30.0000         0

```

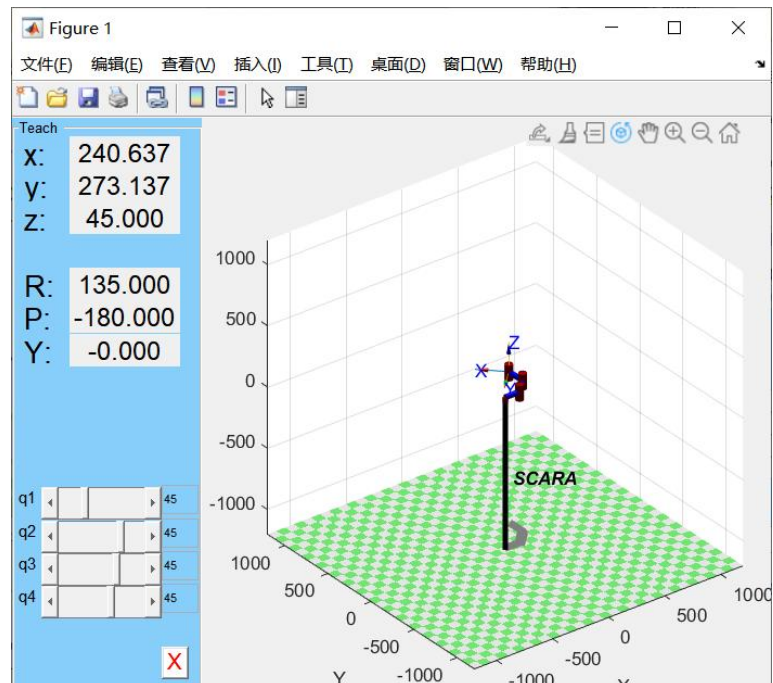

实验报告

仿真结果如下：



3.2 采用自主编写的逆运动学解算代码结果

仿真结果如下：



求解结果如下：

$$Q = \begin{bmatrix} 45.0000 & 45.0000 & 45.0000 & 0 \end{bmatrix}$$

3.3 实验结果分析

逆运动学是通过机械臂末端位置计算各关节角度,由 3.1 和 3.2 的结果可知,仿真结果与程序求解结果在最后一个关节不相同,说明求解过程注重末端的位置而不是姿态,且逆运动学求解结果不唯一,仿真结果合理。

MATLAB 机器人工具箱的优势在于快速、方便地进行正运动学求解,并提供了一系列的函数和工具来处理机器人的运动学问题。而自主编写的逆运动学解算代码可以更加灵活地满足特定需求,但需要自行推导和实现逆运动学算法。

机器人工具箱在机器人建模、轨迹规划、控制、可视化仿真等方面给机器人的研究和学习提供便利条件,大大提高了研究和工作效率。在机器人工具箱中,类函数非常多,主要涉及到机器人建模、正运动学、逆运动学、动力学、逆动力学、雅克比、轨迹规划等。

对于机器人建模,用到最重要的两个函数就是 `Link` 和 `Seriallink` 函数。利用 MATLAB 中 Robotics Toolbox 工具箱中的 `transl`、`rotx`、`roty` 和 `rotz` 可以实现用齐次变换矩阵表示平移变换和旋转变换。利用 Robotics Toolbox 中的 `fkine` 函数可以实现机器人正运动学的求解。利用 Robotics Toolbox 提供的 `ctrj`、`jtraj` 和 `trinterp` 函数可以实现笛卡尔规划、关节空间规划和变换插值。

实验六 桌面 SCARA 机械臂正逆解控制实验

1. 实验内容

1.1 实验目的

- (1) 熟悉并掌握 M 代码转化成 C 代码
- (2) 熟悉并掌握如何通过 Simulink 配置控制器的编译条件
- (3) 熟悉并掌握如何通过 Simulink 搭建 PTP 控制程序
- (4) 熟悉并掌握将 MATLAB 生成的工程导入到 CCS 中
- (5) 熟悉并掌握如何使用串口调试软件和机器人上位机控制界面控制机械臂
- (6) 熟悉并掌握如何对比仿真得到的结果和实际控制时机械臂姿态

1.2 实验原理

同实验四和实验五实验原理。

2. 实验步骤与过程

- (1) 打开机械臂上位机界面，进入”机械臂示教“界面，选择对应的 USB-TTL 的端口号，点击“连接”按钮即可与控制器进行通信。
- (2) 上述操作完成后，点击“启动”按钮，此时机械臂会有一声松开抱闸的声音，说明机械臂已经正常启动。
- (3) 当确保机械臂已经启动，此时可以在“关节角度”中输入四个关节的角度值，然后点击“移动”按钮，即可实现正解控制。
- (4) 如果想要输入末端位姿来控制机械臂，需要事先在 MATLAB 里做仿真，确保机械臂各关节不会碰撞，打开运动学控制文件夹下的 ik_exp_1.m，通过在箭头所指处输入末端位姿求解关节角度值，点击运行按钮，通过三维仿真查看机械臂是否会发生碰撞。
- (5) 如果机械臂各关节不会发生干涉，此时便可在上位机直接输入仿真的末端位姿控制机械臂。
- (6) 做完实验后，点击“回零”按钮，将机械臂回到初始位置，然后点击“停止”按钮，断开所有电源。

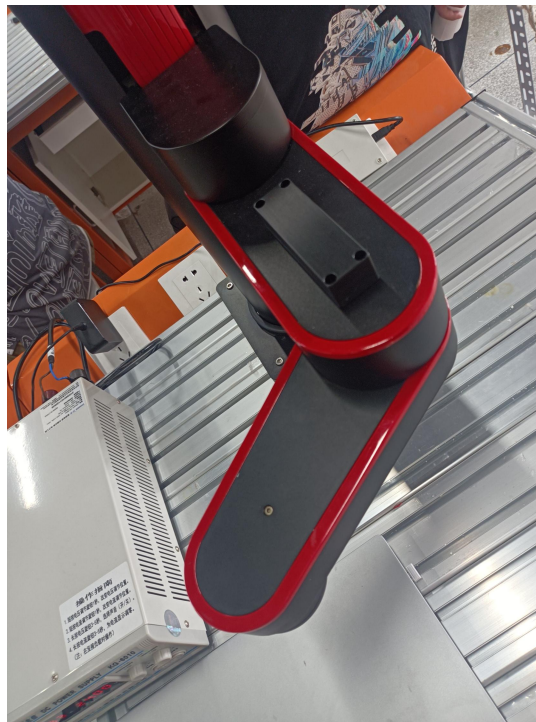
3. 实验结果及分析

3.1 正运动学

发送各关节位置[-20 30 30 30]:



机械臂运动结果:



实验报告

发送各关节位置[-90 90 90 90]:



机械臂运动结果:

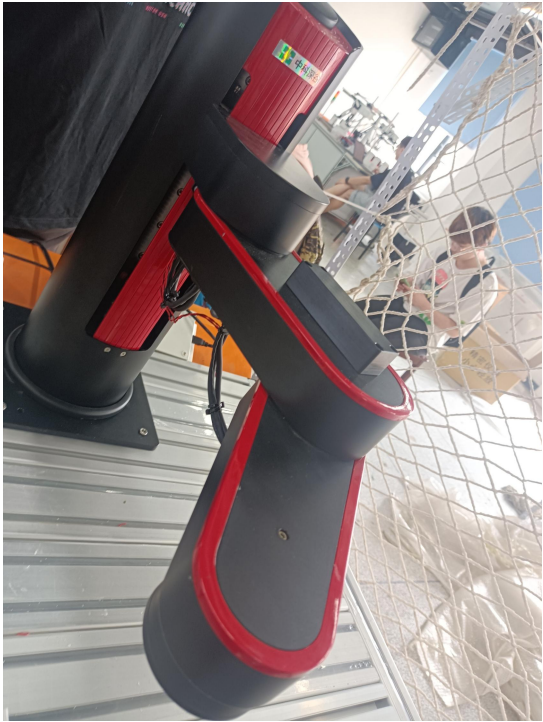


3.2 逆运动学

发送位姿[447.5 0 0 0 0 0]:



机械臂运动结果:



实验七 桌面 SCARA 机械臂 1 动力学 PID 实验

1. 实验内容

1.1 实验目的

- (1) 熟悉并掌握如何通过 Simulink 配置控制器的编译条件
- (2) 熟悉并掌握如何通过 Simulink 搭建单关节动力学控制程序
- (3) 熟悉并掌握如何使用机器人上位机控制界面控制机械臂
- (4) 熟悉 PID 控制原理
- (5) 熟悉实验平台动力学单关节控制实验过程

1.2 PID 控制器原理

PID 控制器作为最早实用化的控制器已有近百年历史，现在仍然是应用最广泛的工业控制器。PID 控制器简单易懂，使用时不需精确的系统模型等先决条件，因而成为应用最为广泛的控制器。

(1) 传统 PID 控制

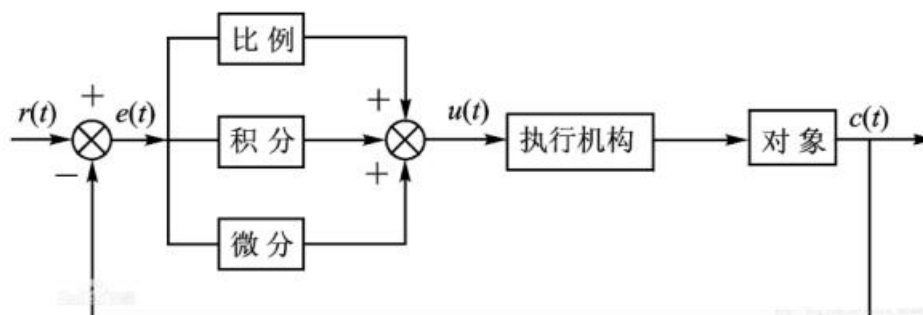


图 7-1 传统 PID 控制结构

$$c(t) = K_p * e(t) + K_i * \int_0^t e(t) dt + K_d * \frac{de(t)}{dt}$$

比例项 P 对系统性能的影响：开环增益越大，稳态误差减小（无法消除，属于有差调节）；过渡时间缩短；稳定程度变差；

积分项 I 控制对系统性能的影响：消除系统稳态误差（能够消除静态误差，属于无差调节）；稳定程度变差；

微分项 D 控制对系统性能的影响：减小超调量；减小调节时间（与 P 控制相比较而言）；增强系统稳定性；增加系统阻尼程度；

(2) 改进后 PID 控制

$$c(t) = K_p * e(t) + K_i * \int_0^t e(t) dt + K_d * \frac{de(t)}{dt} + k_1 \sin \theta + k_2 \dot{\theta}$$

说明：

1. $c(t)$ 的前三项即为经典 PID 控制，是对理想轨迹与实际轨迹的偏差做处理。
2. $k_1 \cdot \sin \theta$ 扩展开就是 $mg l \sin \theta$ ，即 $k_1 = mg l \sin \theta$ 。这一项可视为重力前馈，理由是机械臂在实际运动中的姿态可通过关节传感器读出，即重力可以根据关节角度计算出来，所以可以在控制器中把重力视为可测干扰，直接补偿掉。
3. $k_2 \dot{\theta}$ 在这里起的作用是有效抑制低刚度机械振动。

2. 实验步骤与过程

本次实验先以无摩擦力补偿进行实验

- (1) 确认桌面 SCARA 机械臂没有通电，周围环境无阻挡。手动将摆臂 1，摆臂 2 拉直；
- (2) 确认程控电源调节在 24V；
- (3) 连接好相应的上位机通信线，仿真器线到电脑 USB 接口，确认 SCARA 本体与控制器连接正常，然后给控制器通电；
- (4) 打开“SCARA_302_PID.slx”模型，点击编译下载按钮；
- (5) 待程序加载到控制器中；
- (6) 打开 SCARA GUI，选择对应串口，连接后，旋开急停按钮，点击“启动”。
- (7) 点击“监控面板”，四个监控窗口中会有对应数据显示。
- (8) 组号选择 1，数据 1 写入 20，数据 2 写入 1250，数据 3 写入 5，数据 4 写入 25，数据 5 写入 1，然后点击“下发数据”；观测监控窗口 2 实际位置变化及监控窗口 3 位置偏差曲线。调节 PID 参数。
- (9) 实验结束前，先数据 2，数据 3，数据 4 分别写入 0，点击“下发数据”，控制量输出 0，然后在“机械臂示教”界面中，点击“停止”。给控制器断电。

3. 实验结果及分析

组号选择 1，数据 1 写入 20，数据 2 写入 1250，数据 3 写入 5，数据

实验报告

4 写入 25，数据 5 写入 1，然后点击“下发数据”；观测监控窗口 2 实际位置变化及监控窗口 3 位置偏差曲线。调节 PID 参数，如下图所示：机械臂基本能够跟随正弦运动。

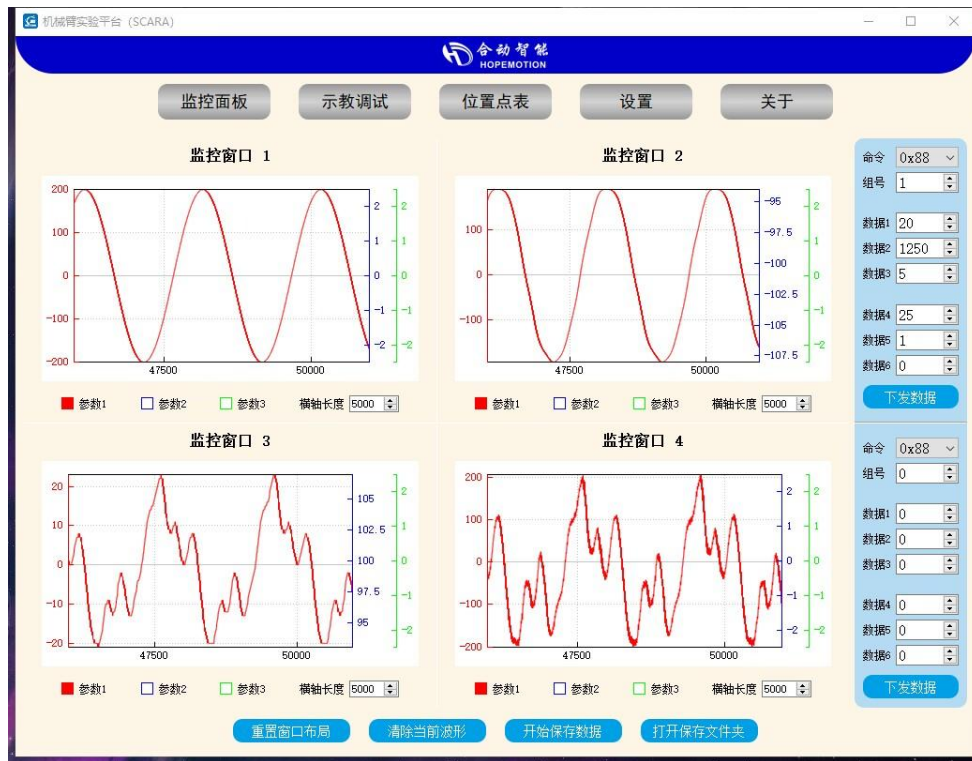


图 7-2 机械臂跟随正弦运动

实验八 桌面 SCARA 机械臂 1,臂 2 动力学 PID 实验

1. 实验内容

1.1 实验目的

- (1) 熟悉并掌握如何通过 Simulink 配置控制器的编译条件
- (2) 熟悉并掌握如何通过 Simulink 搭建双关节动力学控制程序
- (3) 熟悉并掌握如何使用机器人上位机控制界面控制机械臂
- (4) 熟悉 PID 控制原理
- (5) 熟悉桌面 SCARA 机械臂动力学双关节控制实验过程

1.2 实验原理

PID 控制器原理同实验七。

2. 实验步骤与过程

- (1) 确认桌面 SCARA 机械臂没有通电，周围环境无阻挡。手动将摆臂 1，摆臂 2 拉直；
- (2) 确认程控电源调节在 24V；
- (3) 连接好相应的上位机通信线，仿真器线到电脑 USB 接口，确认 SCARA 本体与控制器连接正常，然后给控制器通电；
- (4) 打开“SCARA_302_303_PID.slx”模型，点击编译下载按钮；
- (5) 待程序加载到控制器中；
- (6) 打开 SCARA GUI，选择对应串口，连接后，旋开急停按钮，点击“启动”；
- (7) 点击“监控面板”，四个监控窗口中会有对应数据显示。
- (8) 先写入臂 1 的控制器参数，参考实验七。组号选择 1，数据 1 写入 20，数据 2 写入 1250，数据 3 写入 5，数据 4 写入 25，数据 5 写入 1，然后点击“下发数据”；观测监控窗口 2 实际位置变化及监控窗口 3 位置偏差曲线。
- (9) 写入臂 2 的控制器参数。组号选择 2，数据 1 写入 20，数据 2 写入 2000，数据 3 写入 10，数据 4 写入 30，数据 5 写入 1，然后点击“下发数据”；观测监控窗口 2 实际位置变化及监控窗口 3 位置偏差曲线。
- (10) 实验结束前，先组号 1 的数据 2，数据 3，数据 4，分别写入 0，点击

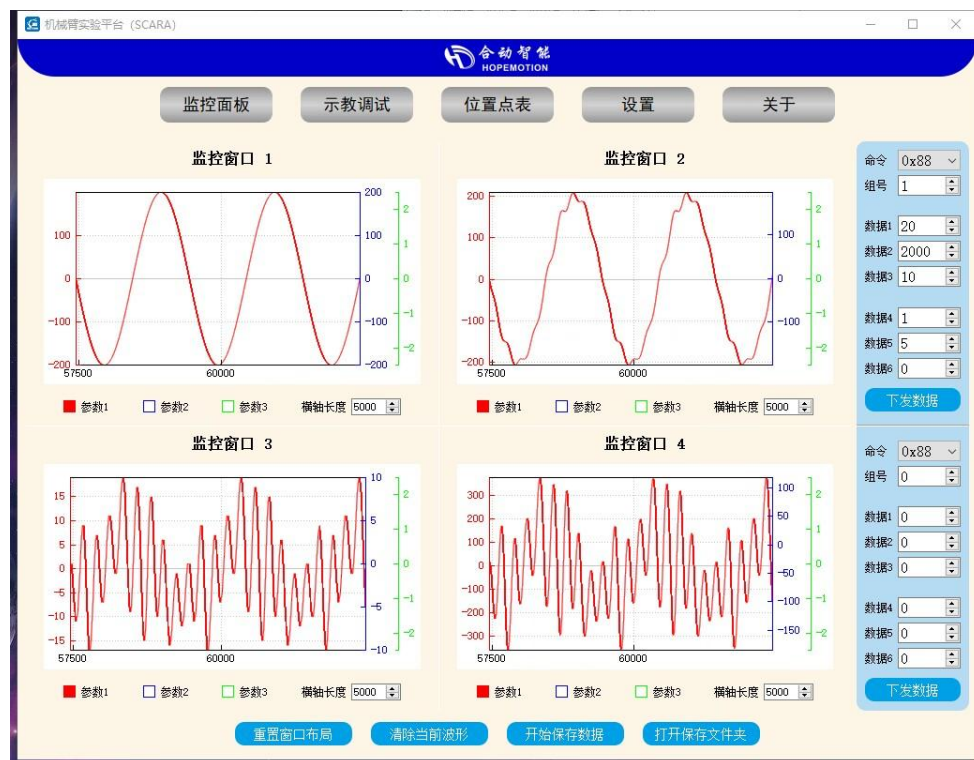
“下发数据”，臂 1 的控制量输出 0。然后组号 2 的数据 2，数据 3，数据 4，分别写入 0，点击“下发数据”，臂 2 的控制量输出 0。然后在“机械臂示教”界面中，点击“停止”。给机械臂断电。

(11) 打开 scara_ptp.slx，下载运行此模型程序，然后给机械臂通电，让机械臂回零位置。

(12) 给控制器及机械臂断电，结束实验。

3. 实验结果及分析

先写入臂 1 的控制器参数，参考实验七。组号选择 1，数据 1 写入 20，数据 2 写入 1250，数据 3 写入 5，数据 4 写入 25，数据 5 写入 1，然后点击“下发数据”；观测监控窗口 2 实际位置变化及监控窗口 3 位置偏差曲线，结果如下图：



写入臂 2 的控制器参数。组号选择 2，数据 1 写入 20，数据 2 写入 2000，数据 3 写入 10，数据 4 写入 30，数据 5 写入 1，然后点击“下发数据”；观测监控窗口 2 实际位置变化及监控窗口 3 位置偏差曲线，结果如下图：

实验报告

