

机器人学  
工业机器人操作实验报告  
**Robotics (2023-2024-2)**  
**Industrial Robot Operation Experiment Report**



姓名：赵四维

学号：521021910696

班级：ME3403-01

任课教师：吴建华、熊振华

实验指导教师：栾楠、张月文

同组成员：杨梓鸿、孙修洁、黄桢

实验日期：2024-6

目录	1
----	---

## 目录

<b>1 实验目的</b>	<b>2</b>
<b>2 机器人背景及实验要求</b>	<b>2</b>
2.1 实验要求	2
2.1.1 模拟机床上下料过程	2
2.1.2 码垛任务模拟	3
2.2 机器人操作注意事项	4
2.2.1 示教过程注意事项	4
2.2.2 生产运行过程注意事项	4
<b>3 实验项目方案和技术路线</b>	<b>4</b>
3.1 实验项目方案	4
3.1.1 模拟机床上下料过程	5
3.1.2 码垛任务模拟	10
3.2 技术路线	14
<b>4 试验调试过程及结果分析</b>	<b>15</b>
4.1 试验调试过程	15
4.1.1 标定坐标系	15
4.1.2 参数设置	15
4.1.3 程序调试	16
4.2 实验结果分析及问题	17
<b>5 总结和心得体会</b>	<b>17</b>

## 1 实验目的

1. 掌握工业机器人的组成及其基本概念。
2. 学习机器人示教编程的方法，掌握机器人编程语言。
3. 使用实验室 FANUC 机器人完成给定的码垛任务模拟，并处理各种可能发生的情况。

## 2 机器人背景及实验要求

Fanuc 是由日本 Fanuc 公司（Fanuc Corporation）设计生产的一系列工业机器人，作为机器人“四大家族”之一，Fanuc 机器人在工业机器人领域有着极高的知名度和市场占有率，其卓越的性能、稳定性和灵活性也被业界广泛认可。学院对于《机器人学》这门课程的实验教学也采用了 Fanuc 机器人，让我们进行一个码垛任务的模拟实验。



图 1: 图为 FANUC Robot series R-30iB/R-30iBF 机器人

### 2.1 实验要求

根据老师的要求，我们采用模块化的思路，并做好分工，将实验分成了两个部分来考虑，即模拟机床上下料过程和模拟码垛过程。

#### 2.1.1 模拟机床上下料过程

1. 机器人工作区内设置：进料位、加工位 1、加工位 2、检测位、正品出料位、废品出料位，共 6 个工位。

2. 机器人等候来料就位信号，从进料位抓取零件。
3. 根据工位就绪信号，将来料放到加工位 1 或者工位 2，发出相应的加工开始指令。
4. 等待加工时机器人可以取下一个来料放到空余工位加工。
5. 从加工位抓取完成的零件，在检测位上方停留，发出相应的启动检测。
6. 根据检验结果（信号），不合格产品放到废品出料位。
7. 合格产品放到正品出料位（托盘），要求按特定垛型码放整齐。放满两层零件后停止工作，发出输出信号提示托盘已经满了，请求更换。
8. 托盘清空后方能继续出料。
9. 托盘未放满时如果输入清空信号，机器人从起点重新开始码放零件，而不再继续未完成的序列。

### 2.1.2 码垛任务模拟

1. 机器人抓取工件，堆码在指定区域。
2. 放置工件的位置、次序和方向按照指定的垛形要求。**垛形奇偶层数不同，不能用简单示教的方式编程。**

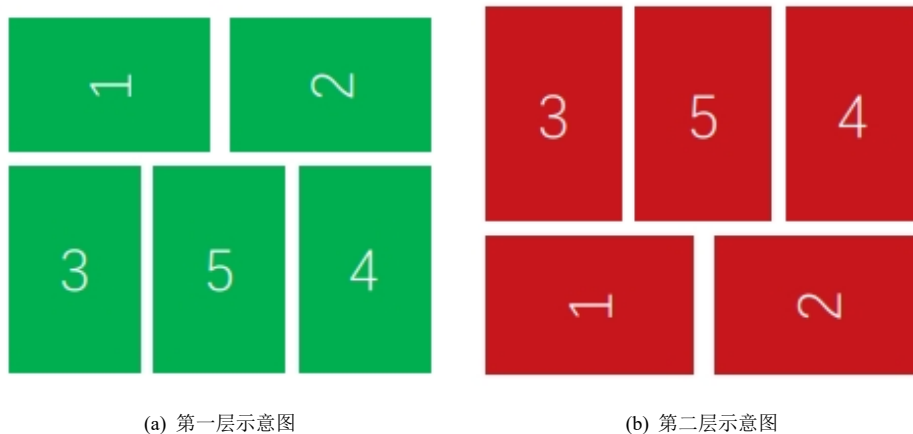


图 2: 码垛示意图

3. 用一路输入信号模拟掉包故障，即如果在去往码垛的过程中出现掉包，机器人应该原地等待，接受人工处理。
4. 故障解除后，机器人按原有次序继续完成作业，而不是从头开始堆码。

## 2.2 机器人操作注意事项

### 2.2.1 示教过程注意事项

1. 请不要戴手套操作示教器和操作盘。
2. 在点动操作机器人时要采用较低的倍率速度以增加对机器人的控制机会。
3. 在按下示教盘上的点动键之前要考虑到机器人的运动姿势。
4. 要预先考虑好避让机器人的运动轨迹，并确认该线路不受干涉。
5. 机器人周围区域保持清洁，无油、无水及杂质等。

### 2.2.2 生产运行过程注意事项

1. 在开机运行前，我们作为操作者必须知道机器人根据所编程序将要执行的全部任务。
2. 操作者需知道所有会左右机器人移动的开关、传感器和控制信号的位置和状态。
3. 必须知道机器人控制器和外围控制设备上的紧急停止按钮的位置，准备在紧急情况下按这些按钮。
4. 不要认为机器人没有移动其程序就已经完成，因为这时机器人很有可能是在等待让它继续移动的输入信号。

## 3 实验项目方案和技术路线

### 3.1 实验项目方案

对于第二部分中提到的关于模拟上下料码垛的要求，我们小组决定将任务分成两个部分来完成，即前期的上料、加工和检测过程，以及后期的码垛过程。前者要求我们以一个清晰的逻辑进行工业流程的制定，后者要求我们通过机械手末端关节的位置变化进行编程，实现码垛的功能。

这里我们有两个假设：

**Hypothesis 1:** 同一批工件加工时间相差不大，在模拟实验中看做相同。

即遵循类似于数据结构中的 FIFO(First In First Out) 原则，先进先出。这决定了我们的控制代码的底层思路——如果加工位 2 上的工件加工完成，加工位 1 上的工件一定加工完成。而加工位 1 上的工件如果没有完成加工，那么加工位 2 上的工件也不会完成加工。

**Hypothesis 2:** 机器人上下料及码垛是一个快速的过程。

由于假设一中，我们的工作流程会存在加工位 2 还在加工工件，而加工位 1 空闲的情况，因此我们引入假设二。这个假设是为了简化实验的复杂度，我们认为机器人的上下料和码垛过程是一个快速的过程，即工件的生产线即使因为上下料和码垛过程出现了加工位空闲，也会是非常短暂的，对整个生产线的正常运行效率和加工成本的影响忽略不计。

**必须承认的是**，我们考虑的是一个理想的工作情况，在展示的过程中，栾老师也提出了这个问题，即实际生产中两个加工位的加工时间是可以不同的。当然，现实工况下还需要考虑非常多的实际情况，这也需要我们在今后的学习生活中多考虑工程实际和工业生产的实际情况。

### 3.1.1 模拟机床上下料过程

在机器人工作区内设置：进料位、加工位 1、加工位 2、检测位、正品出料位、废品出料位，共 6 个工位，我们设定了如下图3所示的工位示意图。

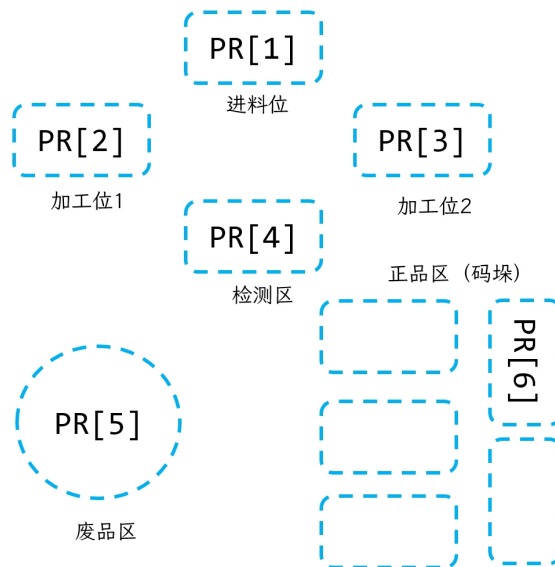


图 3: 工位示意图及位置寄存器设定

值得一提的是，机器人在工作的过程中吸取的工件在移动的过程中，尤其是码垛位间移动时可能会与工位/码垛位上的工件发生碰撞，因此我们引入了“接近点/逃点”的概念，确保机器人末端在到达或离开工位/码垛位时先到达或离开至该工位/码垛位上方有一定安全距离的位置，再行下一步。

这一步的是通过一个临时变量PR[11]来实现，即在机器人到达工位/码垛位之前，先到达PR[11]位置，再通过 z 坐标的加减来实现到达或离开工位/码垛位的目的。数学表达为  $z_{\text{接近点}} = z_{\text{工位}} + 17$ ，在示教器上通过修改PR[11,3]的值来实现。

**Code 1 接近点和工位的设定操作**

```

1 L PR[1] 100mm/sec FINE
2 PR[11]=PR[1]
3 PR[11, 3]=PR[11, 3]-17
4 L PR[11] 500mm/sec FINE

```

这里还有一系列的参数设置，包括数值寄存器、位置寄存器、IO 指令等等，均在下表1中给出。

表 1: 实验参数意义对照表

PR[1]	进料位（上方 17mm）
PR[2]	加工位 1（上方 17mm）
PR[3]	加工位 2（上方 17mm）
PR[4]	检测位（上方 17mm）
PR[5]	废品出料位（上方 17mm）
PR[6]	正品出料位 1（上方 17mm）
PR[11]	临时变量，用于实现工件拿起/放下
PR[12]	临时变量，用于实现工件拿起/放下
R[99]	当前正品区已有零件数量 ( <b>Global</b> )
RO[1]	吸盘状态开/关
DI[101]	输入：来料
DI[102]	输入：加工位 1 加工完毕
DI[103]	输入：加工位 2 加工完毕
DI[104]	输入：检测为正品（OFF 为废品）
DI[105]	输入：掉包故障
DI[106]	输入：清空托盘
DI[107]	输入：加工位 1 可进行加工
DI[108]	输入：加工位 2 可进行加工
DO[101]	输出：加工位 1 正在加工
DO[102]	输出：加工位 2 正在加工
DO[103]	输出：托盘已满，请求更换

**第一步：机器人等待来料就位信号，从进料位抓取零件，并判断前往哪个工位。**

开始，机器人等待来料信号 (DI101)，若如果一直没有收到信号则循环等待。如果收到了来料信号，则判断两个加工位是否都就绪。如果两个加工位都没有就绪，那么进入检测是否完成阶段。直到不满足“两加工位都为就绪”条件式进入加工部分。

### Code 2 第一步操作代码

```

1 LBL[1]
2 WAIT 2(sec)
3 LBL[4]
4 IF DI[101]=OFF JMP LBL[6] #若无来料，前往工作区查看情况，是否有加工完成需要移动至检测及后续位置的。、
5 IF ( DI[107] ) THEN #若加工位1准备就绪
6 L PR[1] 100mm/sec FINE #前往来料区取料
7 PR[11]=PR[1]
8 PR[11, 3]=PR[11, 3]-17
9 L PR[11] 500mm/sec FINE
10 RO[1]=ON
11 WAIT 2(sec)
12 L PR[1] 100mm/sec FINE
13 L PR[2] 100mm/sec FINE #前往加工位1
14 PR[11]=PR[2]
15 PR[11, 3]=PR[11, 3]-17
16 L PR[11] 500mm/sec FINE
17 RO[1]=OFF
18 L PR[2] 100mm/sec FINE
19 DO[101]=ON #输出信号：加工位1正在加工
20 ELSE
21 IF (DI[108]) THEN #若加工位2准备就绪
22 L PR[1] 100mm/sec FINE
23 PR[11]=PR[1]
24 PR[11, 3]=PR[11, 3]-17
25 L PR[11] 500mm/sec FINE
26 RO[1]=ON
27 WAIT 2.00(sec)
28 L PR[1] 100mm/sec FINE
29 L PR[3] 100mm/sec FINE #前往加工位2
30 PR[11]=PR[3]
31 PR[11, 3]=PR[11, 3]-17
32 L PR[11] 500mm/sec FINE
33 RO[1]=OFF
34 L PR[3] 100mm/sec FINE
35 DO[102]=ON #输出信号：加工位2正在加工
36 ELSE #加工位1、2均不可加工，返回等待
37 JMP LBL[6]
38 ENDIF

```

我们在程序编写的过程中很多时候用到了**标签**的概念，即在程序中的某一行给一个标签，然



后在程序的其他地方通过 `JMP LBL[item]` 来实现跳转。这样的好处是可以使得程序的逻辑更加清楚，对于响应的 LBL 处应该进行的操作一目了然，比如进行工件的抓取、两期信号灯之类的操作。可以类比 C++ 程序语言中一个大括号内的操作。

### 第二步：加工部分分配

这里分为两种情况。如果 1 号加工位就绪，那么直接将工件放到 1 号加工位上，并发出加工开始指令，DO[101] 点亮。如果 1 号加工位没有就绪，那么将工件放到 2 号加工位上，并发出加工开始指令，DO[102] 点亮。流程进入检测是否加工完成阶段。

在 LBL[6] 下，我们综合判断来料情况、工位就绪情况，如下所示：

#### Code 3 LBL[6]: 综合研判

```

1 LBL[6]
2 IF DO[101]=OFF, JMP LBL[2]
3 LBL[5]
4 WAIT 1.00(sec)
5 IF DI[102]=ON, JMP LBL[3]
6 LBL[2]
7 IF DO[102]=OFF, JMP LBL[4]
8 IF DI[103]=OFF, JMP LBL[4]

```

### 第三步：检测是否加工完成

进入这一步的前提是两工位均“未就绪”。我们先判断加工位 1 是否完成加工 (即等待外部信号)，如果完成加工，那么将工件从加工位 1 抓取到检测位，同时将加工位 1 的情况设为空 (就绪)。如果加工位 1 没有完成加工，那么判断加工位 2 是否完成加工，如果完成加工，那么将工件从加工位 2 抓取到检测位，同时将加工位 2 设置为就绪。如果加工位 2 没有完成加工，那么继续等待。

显然，LBL[3] 的作用就是清理已经加工完的工件，准备将其送到检测区进行检测。

#### Code 4 LBL[3]: 清理已经加工完的工件

```

1 LBL[3]
2 IF (DI[102]) THEN # 1号加工位加工完毕,将它拿去检测
3 L PR[2] 100mm/sec FINE
4 PR[11]=PR[2]
5 PR[11, 3]=PR[11, 3]-17
6 L PR[11] 500mm/sec FINE
7 RO[1]=ON
8 WAIT 2.00(sec)
9 L PR[2] 100mm/sec FINE
10 L PR[4] 100mm/sec FINE
11 R[50]=1
12 DO[101]=OFF
13 ELSE # 2号加工位加工完毕,将它拿去检测
14 L PR[3] 100mm/sec FINE
15 PR[11]=PR[3]
16 PR[11, 3]=PR[11, 3]-17

```

```

17 L PR[11] 500mm/sec FINE
18 RO[1]=ON
19 WAIT 2.00(sec)
20 L PR[3] 100mm/sec FINE
21 L PR[4] 100mm/sec FINE
22 R[51]=1
23 DO[102]=OFF
24 ENDIF
25 WAIT 2.00(sec)

```

#### 第四步：获取检测结果及后续处理

从加工位抓取完成的零件，在检测位上方停留，发出相应的启动检测指令。通过检测信号 (DI[104]) 判断工件是否为正品。如果不是，则将工件放到废品出料位。如果是，则将工件放到正品出料位。正品出料位的堆叠用一路输入信号模拟掉包故障，即如果在去往码垛的过程中出现掉包，机器人应该原地等待，接受人工处理，处理完成后返回最初的第一步来料检测。这一步我们小组完成的不是特别好，后续还将继续说明。

#### Code 5 掉包检测

```

1 IF (DI[104]) THEN #若为正品，继续
2 IF (DI[105]) THEN #若出现掉包故障，等待人工处理（等待后放下零件）
3 WAIT 5.00(sec)
4 RO[1]=OFF
5 JMP LBL[16]

```

如果是正品，那么我们需要将工件放到正品出料位，这里我们通过一个计数器 R[99] 来记录正品区物件数量，然后通过一个选择器来选择放置的位置，这里我们选择了 10 个位置，即放满两层零件后停止工作。这里还用到了 FANUC 机器人的 SELECT 指令，可以根据不同的条件选择不同的跳转位置，类似于 C++ 中的 switch-case 语句。

#### Code 6 正品、废品的处理

```

1 ELSE #未出现掉包故障，移至正品区
2 R[99]=R[99]+1 #记录正品区物件数量
3 PR[7]=PR[6]
4 PR[7, 3]=PR[7, 3]+18=11
5 SELECT R[99]=1, JMP LBL[21] #按次序放置正品
6     =2, JMP LBL[22]
7     =3, JMP LBL[23]
8     =4, JMP LBL[24]
9     =5, JMP LBL[25]
10    =6, JMP LBL[26]
11    =7, JMP LBL[27]
12    =8, JMP LBL[28]
13    =9, JMP LBL[29]
14    =10, JMP LBL[30]
15
16 // ..... //

```

```

17
18 ELSE #若为废品，移至废品区
19 L PR[5] 100 mm/sec FINE
20 PR[11]=PR[5]
21 PR[11, 3]=PR[11, 3]-17
22 L PR[11] 500mm/sec FINE
23 RO[1:OFF]=OFF
24 L PR[5] 100mm/sec FINE
25 JMP LBL[16]
26 ENDIF

```

如果没有掉包，那么还需要检查是否需要清空托盘，如果需要清空托盘，那么机器人停止工作，等待人工处理，处理完成后返回最初的第一步来料检测，同时，我们自己的计数器也能根据“是否到达 10 个”来判断是否需要清空托盘；若清空托盘，全局变量 R[99] 清零。

#### Code 7 LBL[16]-继续送料或托盘清空

```

1 LBL[16]
2 IF(DI[106:OFF]) THEN #清空信号检测
3 R[99]=0
4 WAIT 5.00(sec)
5 DO(103)=OFF
6 JMP LBL[7]
7 ELSE
8 JMP LBL[7]
9 ENDIF
10
11 LBL[7]
12 IF(R[99]=10) THEN
13 DO(103)=ON #托盘已满，发信号提示
14 JMP LBL[16] #跳转清空信号检测，若未清空，机器将停止工作，在此循环
15 ELSE
16 JMP LBL[6] #托盘未满，返回等待

```

### 3.1.2 码垛任务模拟

这一部分主要是涉及怎样由一个基准位置 PR[6] 来推断出 10 个码垛位置，然后通过一个计数器来记录码垛的层数，最后通过一个选择器来选择不同的码垛位置。这里我们采用了一个比较简单的码垛方式，即奇数层和偶数层的码垛方式不同，如第二章图2所示。这个部分其实是 3.1 中的一个小部分，但是由于其独立性较强，因此我们将其单独拿出来讨论。

第一个工件只需要在 PR[6] 的基础上进行绕 z 轴旋转 90 度即可，我们定义下标的数字表示工件的序号即 R[99]，字母表示平移方向或旋转轴。这里我们以第一个工件为例，我们需要在 PR[6] 的基础上进行旋转，即：

$$Rotate_{1z} = 90^\circ$$

**Code 8 LBL[21]-第一块正品的码垛**

```

1 LBL[21]
2 PR[11]=PR[6]
3 PR[11, 3]=PR[11, 3]-17
4 PR[11, 6]=PR[11, 6]+90 #绕Z轴 旋转
5 PR[12]=PR[11]
6 PR[12, 3]=PR[12, 3]+17 #计算摆放位置（及其上方位置）
7 L PR[12] 100mm/sec FINE
8 L PR[11] 500mm/sec FINE
9 RO[1:OFF]=OFF
10 L PR[12] 100mm/sec FINE
11 JMP LBL[16]

```

后续的工件则需要考虑平移成分，在这里以第二个工件为例，我们需要在 PR[6] 的基础上进行平移，即：

$$\begin{cases} Translate_{2x} = 38 \\ Rotate_{2z} = 90^\circ \end{cases}$$

**Code 9 LBL[22]-第二块正品的码垛**

```

1 LBL[22]
2 PR[11]=PR[6]
3 PR[11, 3]=PR[11, 3]-17
4 PR[11, 1]=PR[11, 1]+38
5 PR[11, 6]=PR[11, 6]+90
6 PR[12]=PR[11]
7 PR[12, 3]=PR[12, 3]+17
8 L PR[12] 100mm/sec FINE
9 L PR[11] 500mm/sec FINE
10 RO[1:OFF]=OFF
11 L PR[12] 100mm/sec FINE
12 JMP LBL[16]

```

当 R[99] 到达 6 时，我们要考虑工件需要摆放在第二层，如第二章图2所示。以第六个工件为例，我们需要在 PR[6] 的基础上进行 x, y 方向的平移和绕 z 轴的旋转，即：

$$\begin{cases} Translate_{6x} = 45 \\ Translate_{6y} = -42 \\ Rotate_{6z} = 90^\circ \end{cases}$$

不过需要注意的是，这里由于下面已经堆了一层，这里 Z 轴的平移量 (物体的放下操作) 需要相应的上移，不再是之前的-17mm，我们这里采用 +17mm。

**Code 10 LBL[26]-第六块正品的码垛**

```

1 LBL[26]
2 PR[11]=PR[7]

```

```

3 PR[11, 1]=PR[11, 1]+45
4 PR[11, 2]=PR[11, 2]-42
5 PR[11, 6]=PR[11, 6]+90
6 PR[12]=PR[11]
7 PR[12, 3]=PR[12, 3]+17
8 L PR[12] 100mm/sec FINE
9 L PR[11] 500mm/sec FINE
10 RO[1:OFF]=OFF
11 L PR[12] 100mm/sec FINE
12 JMP LBL[16]

```

码垛区主要是以上三种类型的操作，剩余的操作大同小异，主要是平移的具体参数不同，这里就不再赘述，将代码一并附在下面代码块 11 中。

#### Code 11 其他正品码垛

```

1 LBL[23] # 第三块正品的码垛
2 PR[11]=PR[6]
3 PR[11, 1]=PR[11, 1]-10
4 PR[11, 2]=PR[11, 2]-35
5 PR[11, 3]=PR[11, 3]-17
6 PR[12]=PR[11]
7 PR[12, 3]=PR[12, 3]+17
8 L PR[12] 100mm/sec FINE
9 L PR[11] 500mm/sec FINE
10 RO[1:OFF]=OFF
11 L PR[12] 100mm/sec FINE
12 JMP LBL[16]
13 LBL[24] # 第四块正品的码垛
14 PR[11]=PR[6]
15 PR[11, 1]=PR[11, 1]+50
16 PR[11, 2]=PR[11, 2]-35
17 PR[11, 3]=PR[11, 3]-17
18 PR[12]=PR[11]
19 PR[12, 3]=PR[12, 3]+17
20 L PR[12] 100mm/sec FINE
21 L PR[11] 500mm/sec FINE
22 RO[1:OFF]=OFF
23 L PR[12] 100mm/sec FINE
24 JMP LBL[16]
25 LBL[25] # 第五块正品的码垛
26 PR[11, 1]=PR[11, 1]+17
27 PR[11, 2]=PR[11, 2]-37
28 PR[11, 3]=PR[11, 3]-17
29 PR[12]=PR[11]
30 PR[12, 3]=PR[12, 3]+17
31 L PR[12] 100mm/sec FINE
32 L PR[11] 500mm/sec FINE
33 RO[1:OFF]=OFF
34 L PR[12] 100mm/sec FINE

```

```
35 JMP LBL[16]
36
37 // ..... //
38
39 LBL[27] # 第七块正品的码垛
40 PR[11]=PR[7]
41 PR[11, 3]=PR[11, 3]-17
42 PR[11, 2]=PR[11, 2]-40
43 PR[11, 6]=PR[11, 6]+90
44 PR[12]=PR[11]
45 PR[12, 3]=PR[12, 3]+17
46 L PR[12] 100mm/sec FINE
47 L PR[11] 500mm/sec FINE
48 RO[1:OFF]=OFF
49 L PR[12] 100mm/sec FINE
50 JMP LBL[16]
51 LBL[28] # 第八块正品的码垛
52 PR[11]=PR[7]
53 PR[11, 3]=PR[11, 3]-17
54 PR[11, 1]=PR[11, 1]+52
55 PR[12]=PR[11]
56 PR[12, 3]=PR[12, 3]+17
57 L PR[12] 100mm/sec FINE
58 L PR[11] 500mm/sec FINE
59 RO[1:OFF]=OFF
60 L PR[12] 100mm/sec FINE
61 JMP LBL[16]
62 LBL[29] # 第九块正品的码垛
63 PR[11]=PR[7]
64 PR[11, 3]=PR[11, 3]-17
65 PR[11, 1]=PR[11, 1]-7
66 PR[12]=PR[11]
67 PR[12, 3]=PR[12, 3]+17
68 L PR[12] 100mm/sec FINE
69 L PR[11] 500mm/sec FINE
70 RO[1:OFF]=OFF
71 L PR[12] 100mm/sec FINE
72 JMP LBL[16]
73 LBL[30] # 第十块正品的码垛
74 PR[11]=PR[7]
75 PR[11, 3]=PR[11, 3]-17
76 PR[11, 1]=PR[11, 1]+25
77 PR[12]=PR[11]
78 PR[12, 3]=PR[12, 3]+17
79 L PR[12] 100mm/sec FINE
80 L PR[11] 500mm/sec FINE
81 RO[1:OFF]=OFF
82 L PR[12] 100mm/sec FINE
83 JMP LBL[16]
```

### 3.2 技术路线

应老师要求，我们画出了整个实验的技术路线的流程图，如下图4所示。

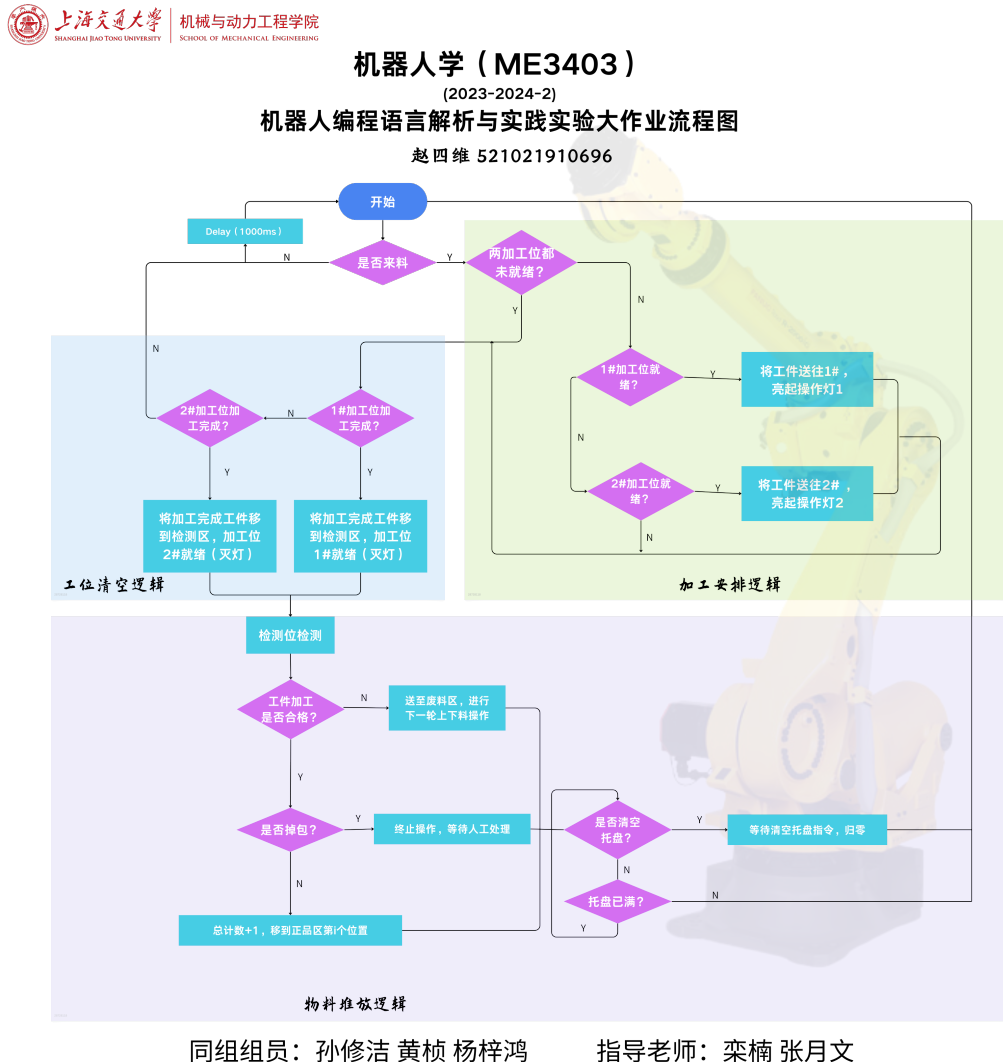


图 4: 技术路线流程图

当然这是我自己画的流程图，可能和组员所画的流程图有所差别，但大家的想法主流是一致的，在只有一台操作设备的情况下实现功能时会求同存异，望老师理解。

## 4 试验调试过程及结果分析

### 4.1 试验调试过程

#### 4.1.1 标定坐标系

在实验开始之前，我们首先需要标定机器人的坐标系，这里我们采用了基坐标系和工具坐标系的标定方法，即通过示教器上的工具坐标系标定功能来实现。用户坐标系我们直接选择世界坐标系，即机器人的基坐标系，而工具坐标系我们选择了吸盘末端。

##### Code 12 标定坐标系

```
1 UTOOL_NUM = 2
2 UFRAME_NUM = 1
```

再进行示教的过程中，我们用的是三点标定法。三点标定法用于确定机器人工具坐标系，通过让机器人末端执行器以不同的姿势指向同一点，从而确定工具坐标系的中心点 TCP 的位置，如下图5所示。

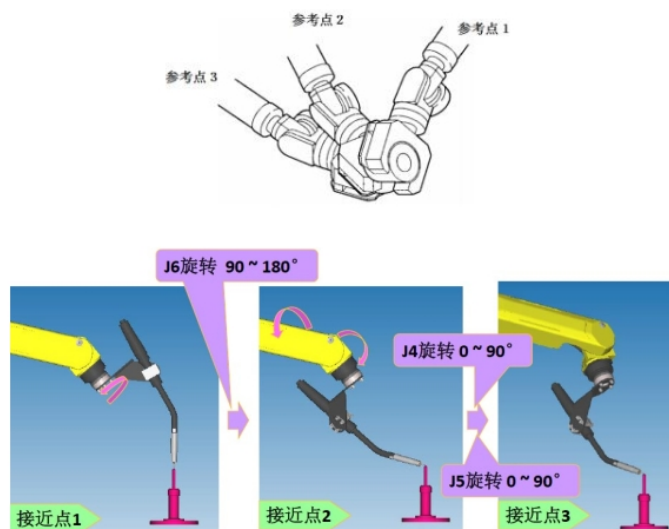


图 5: 三点标定法示意图

#### 4.1.2 参数设置

寄存器是 CPU 内部的一组存储单元，用于存放程序运行时的中间数据，如计数器、计时器、数据寄存器等。在实验中，我们需要设置一些寄存器，如位置寄存器、数值寄存器等，这些寄存器的设置在第三部分的表1中已经给出。



另外，还有一种信号即 I/O 信号，机器人的 I/O 信号是 RI[i] 和 RO[i]，其中 RI[i] 是输入信号，RO[i] 是输出信号。在实验中，我们需要设置一些 I/O 信号，如来料信号、加工位 1 加工完毕信号等等，这些信号的设置也在第三部分的表1中给出。

#### 4.1.3 程序调试

这一部分的详细思路即为第三部分的全部内容，这里不再赘述，附上几张在实验过程中我们进行的一些操作图。

下图6展示了第一个加工好的工件进入正品区码垛的情形：

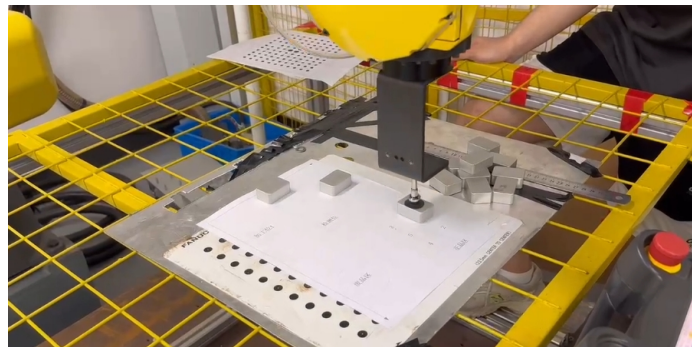


图 6: 第一个工件进入正品区码垛

下图7展示了既有工件不合格进入废料区，正品区也堆到了第二层的情形，可以看到第一层和第二层的码垛方式是不同的：



图 7: 废品区和正品区的情况

## 4.2 实验结果分析及问题

我们小组通力合作，较好的完成了实验任务，2.1.1 节的任务基本满足要求，但是存在一个掉包问题的解决，即在码垛的过程中，如果出现掉包，机器人应该原地等待，接受人工处理，处理完成后返回最初的第一步来料检测。而在我们的处理过程中，我们是假设掉包是一个流程中的一个步骤，而不是一个特殊的情况，并没有通过“SKIP”指令来实现，这是我们在实验中的一个问题。FANUC 机器人的“SKIP”指令是一种用于跳过当前程序段中的某些操作或用于特定条件下中断程序的指令。它通常与传感器输入结合使用，用于实现工件检测、碰撞检测等功能，这一命令能很好的解决工作情景里的掉包问题，遗憾的是在有限的课程时间内我们没能完成相关逻辑的整理和代码的编写。

在 2.1.2 节，这一部分的任务还是完成的比较好的，我们通过计数器的设置和选择器的使用，实现了码垛区的堆叠，这一部分的代码逻辑也比较清晰，美中不足的是由于初始位置的工件是我们自己摆放的，每次手工摆放的位置可能会有所不同，这就导致了我们的码垛区的堆叠不是很整齐，这是我们在实验中的一个问题。在实际的工程实践中，自动化的产业链会确保每个位置的工件都是整齐摆放的，这样能保证机器人的正常工作。

## 5 总结和心得体会

通过四节实验课，我对《机器人学》这门课程有了更加深刻的认识，对机器人的基本概念、基本原理、基本结构有了更加深入的了解。在实验课中，我学会了如何使用 FANUC 机器人的示教器，如何编写机器人的程序，如何调试机器人的程序，如何进行机器人的标定等等，这些都是我在课堂上所没有学到的，这些知识对我以后的学习和工作都有很大的帮助。我们课程中学习的理论（旋转矩阵、运动学、动力学等）都被很好的集成在了 FANUC 的操作系统里，这让我对课程的理论知识有了更加直观的认识。

我们小组的整体实验还算成功，只是在过程中还是遇到了不少困难。第三节课我们并没有准备的很充分，在没有提前画好流程图的情况下，磕磕绊绊的发生了很多错误。最后一节课前我们梳理了一下逻辑，并在老师的指导下顺利完成实验任务。

不过我们还存在一些问题，比如在码垛区的堆叠不是很整齐，没有采用“SKIP”指令来解决掉包问题等等。不过整个实验于我而言确实是收获良多，不管是硬件的操作、逻辑的梳理还是代码的编写，甚至是小组成员间针对分歧的沟通，都让我受益匪浅。再次感谢栾楠老师和张月文老师的耐心指导，让我顺利完成实验任务。我也将怀着对机器人的热爱，继续深入学习，在未来坚定的投入我国机器人产业的发展的时代洪流中。