



南開大學  
Nankai University

南 開 大 學

人 工 智 能 學 院

---

## 《机器人学导论实验》课程大作业

---

专 业: 智能科学与技术  
姓 名: 石若川  
学 号: 2111381  
指导教师: 周璐  
课 号: 0410  
完成日期: 2024 年 5 月 29 日

# 目录

<b>1 引言</b>	<b>2</b>
<b>2 平台和环境配置</b>	<b>2</b>
<b>3 机械臂点位控制</b>	<b>3</b>
3.1 准备工作	3
3.2 设置目标点	3
3.3 创建路径	4
3.4 调整参数配置和目标点姿态	5
3.5 结果展示	6
<b>4 机械臂位姿控制</b>	<b>7</b>
4.1 轨迹设置	7
4.2 结果展示	8
<b>5 机械臂跟踪喷漆与码垛</b>	<b>9</b>
5.1 跟踪喷漆	9
5.1.1 喷漆 Smart 组件	9
5.1.2 传送带搭建	9
5.1.3 物块切换	12
5.2 码垛	14
<b>6 总结</b>	<b>19</b>

## 1 引言

随着工业自动化技术的不断发展，机器人技术在工业生产中的应用愈加广泛。机器人系统不仅能够提高生产效率和质量，还可以降低人力成本，提升工作安全性。因此，对机器人控制技术的研究和应用具有重要意义。

本实验旨在探索机械臂在工业自动化中的应用，通过使用 ABB RobotStudio 仿真软件，完成了机械臂的点位控制和位姿控制，并搭建了一个跟踪喷漆和码垛流水线的仿真环境。具体而言，实验包括了以下几个方面的内容：

- 首先，通过指定六个点的运动，实现了机械臂在空间中的定点移动。这部分实验的过程中，我熟悉了 RobotStudio 中机械臂的基本控制方法，并验证机械臂在指定轨迹下的运动准确性。
- 其次，实现了机械臂沿着椭圆轨迹运动。通过使用 RobotStudio 中自动路径功能，创建了椭圆轨迹，探索了机械臂在复杂轨迹下的运动特性。
- 最后，搭建了一个跟踪喷漆和码垛流水线的仿真环境，实现了机械臂在流水线上的自动化操作。通过这一部分实验，探讨了机械臂在工业生产中的应用场景，以及机械臂与其他设备之间的协同工作。

通过本实验，我深入理解机械臂控制技术的原理，掌握了 ABB RobotStudio 仿真软件的使用，完成了机械臂在工业自动化生产中的简单应用。

## 2 平台和环境配置

实验中使用 Windows 11 系统，ABB RobotStudio 6.08 版本进行仿真实验。

### 3 机械臂点位控制

实验中使用 IRB-1410 机器人，利用示教指令记录末端执行器的位置，并通过“偏移位置”操作，获得空间中的六个点位。使用 MoveJ 命令，控制机械臂经过六个点，完成机械臂的点位控制。

#### 3.1 准备工作

实验中选择使用 IRB-1410 机器人进行点位控制的仿真。首先在新创建的工作站中添加该机器人，并从导入模型库的设备中选择内置工具“myTool”，将工具安装到机器人上。

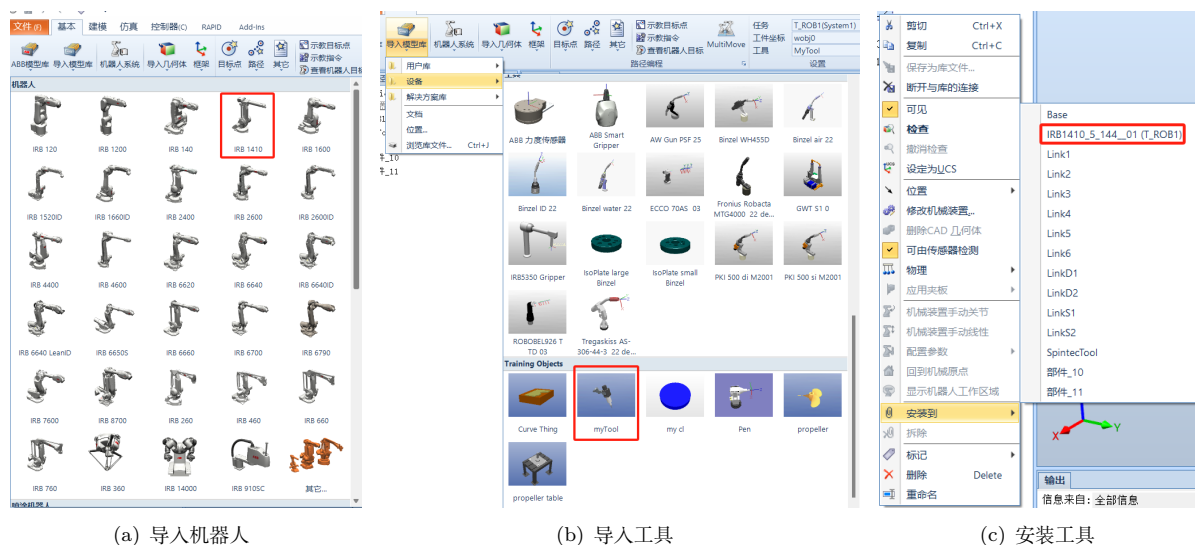


图 3.1: 导入机器人并安装工具

安装工具后，从已有布局中创建机器人系统。进行以上操作后，工作站就可以对机器人进行编程和示教等操作。



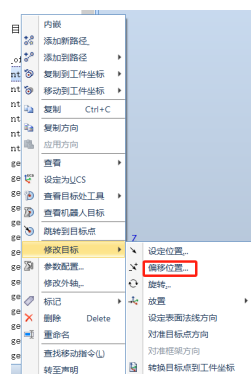
图 3.2: 创建机器人系统

#### 3.2 设置目标点

实验中使用示教目标点的方式获得机器人机械原点位置下工具末端的位置，标注为此点为 point1。通过对 point1 进行“偏移位置”的设定，获得另外 5 个点，分别标注为 point2-6。



(a) 示教目标点

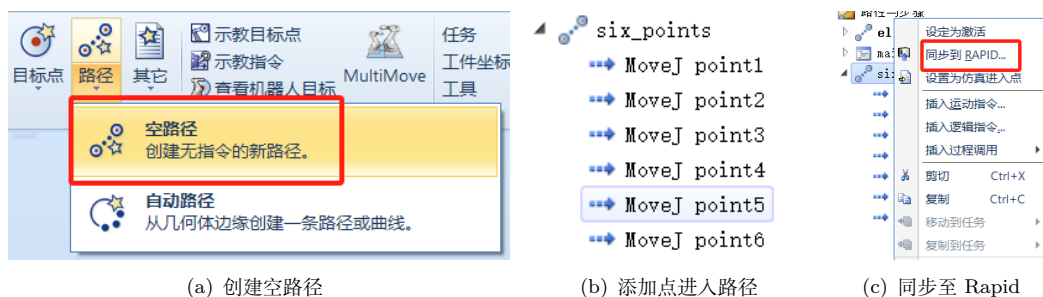


(b) 偏移位置

图 3.3: 设置六个目标点位

### 3.3 创建路径

点击菜单栏中“路径”下的“空路径”添加一条途径，并命名为 six\_points。将设置的六个点 point1-6 拖入创建的路径中，并点击“同步到 Rapid”。经过以上操作，设置的六个目标点和创建的路径就会同步到 Rapid 代码中，方便进行一步编写。



(a) 创建空路径

(b) 添加点进入路径

(c) 同步至 Rapid

图 3.4: 创建路径

在 Rapid 代码中，对默认的路径代码进行编辑。首先将 MoveL 指令全部替换为 MoveJ 指令。这是由于 MoveJ 指令控制沿着关节空间的曲线路径移动，不容易出现奇异点；MoveL 指令控制沿着直线路径移动，机器人的末端工具保持相对于基坐标系的末端姿态不变，容易出现奇异点。经过实际测试，使用 MoveL 命令出现奇异点的情况远多于 MoveJ 命令，因此在两点距离较大且对路径精度要求较低的情况下，使用 MoveJ 命令会更容易控制。

其次为了保证关节平稳运动，将关节速度均设置为 v100，否则会出现关节运动速度过快的情况。在实际使用中，关节运动速度过快可能会造成机械故障和人身伤害。代码如下所示：

```
1 PROC six_points()
2     MoveJ point1,v100,z10,MyTool\WObj:=wobj0;
3     MoveJ point2,v100,z10,MyTool\WObj:=wobj0;
4     MoveJ point3,v100,z10,MyTool\WObj:=wobj0;
```

```

5 MoveJ point4,v100,z10,MyTool\WObj:=wobj0;
6 MoveJ point5,v100,z10,MyTool\WObj:=wobj0;
7 MoveJ point6,v100,z10,MyTool\WObj:=wobj0;
8 MoveJ point1,v100,z10,MyTool\WObj:=wobj0;
9 ENDPROC

```

将 six\_points 加入主函数 main 中，并将 Rapid 代码同步到工作站后，即可进行仿真。



图 3.5: 同步工作站

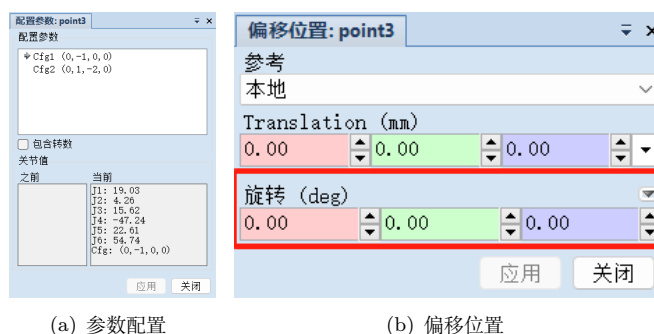
### 3.4 调整参数配置和目标点姿态

理论上完成以上操作后，机器人就可以按照给定的路径进行跟踪。但是实际结果表明，这样规划出的路径上可能存在奇异点。这一问题的原因是：

- 尽管起始点和目标点均在机器人可达区域内，但是由于轨迹并不是唯一的，程序自行规划的轨迹中关节参数配置可能并不是最合理的，因此会出现奇异点的情况。
- 实验中使用“偏移位置”的方式设置点位，末端执行器的姿态均相同。因此可能根本无法得到合理的轨迹。

解决方法如下：

- 调整目标点的参数配置：对目标点进行参数配置，尝试各种参数配置，选择其中最合理的。
- 调整目标点的姿态：对目标点进行偏移位置操作，通过绕 XYZ 轴旋转，在不改变位置的情况下得到较合理的姿态。



(a) 参数配置

(b) 偏移位置

图 3.6: 调整参数配置和目标点姿态

### 3.5 结果展示

实验结果如图3.7所示，可以看出机器人可以按照设置的六个目标点和轨迹进行运动。但由于使用 MoveJ 命令，所以实际轨迹并不是完全按照直线遍历六个给定点位，而是按照曲线。

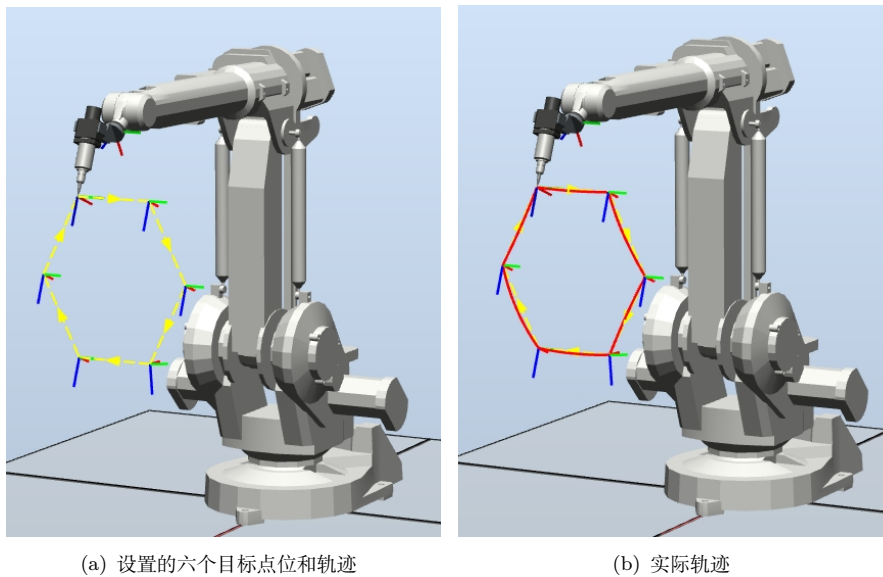


图 3.7: 点位控制结果展示

## 4 机械臂位姿控制

实验中仍使用 IRB-1410 机器人进行位姿控制的仿真，利用曲线功能绘制椭圆的形状，再利用从曲线生成表面功能创建了椭圆形状的表面，最后利用该表面使用自动路径功能获得椭圆轨迹，完成机械臂的位姿控制。

准备工作与点位控制部分相同，此处不做过多介绍。

### 4.1 轨迹设置

实验中我首先尝试了利用椭圆的轨迹方程编写代码进行位姿控制，但是实际测试后发现，该方法会导致轨迹中无法规避地存在较多的奇异点。因此，实验中使用自动路径功能获得椭圆轨迹。

首先利用曲线功能创建一条椭圆曲线，以机器人机械原点为椭圆中心，长轴为 200，短轴为 150。接着，利用从曲线生成表面功能将椭圆曲线生成一个椭圆表面。最后利用自动路径功能，生成一条沿椭圆表面边缘的轨迹。

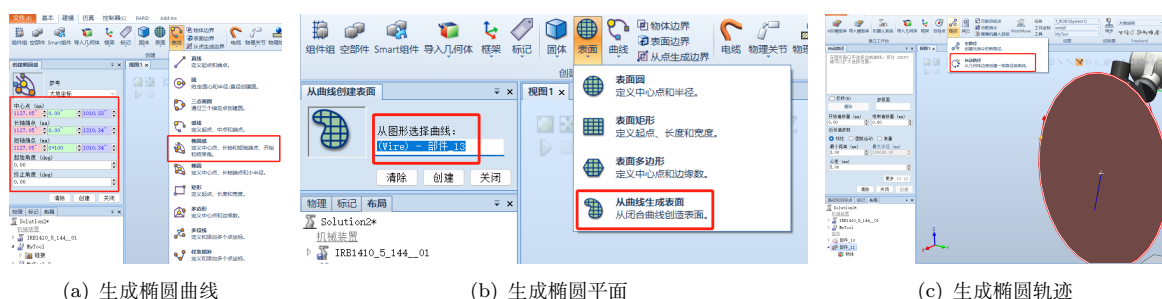


图 4.8: 设置椭圆轨迹

将自动生成的路径命名为 ellipse，并同步至 Rapid 中。与点位控制类似，将 MoveL 命令替换为 MoveJ 命令，并将速度调整为 v100。代码如下：

```
1 PROC ellipse()
2     MoveJ Target_10,v100,z10,MyTool\WObj:=wobj0;
3     MoveJ Target_20,v100,z10,MyTool\WObj:=wobj0;
4     MoveJ Target_30,v100,z10,MyTool\WObj:=wobj0;
5     MoveJ Target_40,v100,z10,MyTool\WObj:=wobj0;
6     MoveJ Target_50,v100,z10,MyTool\WObj:=wobj0;
7     MoveJ Target_60,v100,z10,MyTool\WObj:=wobj0;
8     MoveJ Target_70,v100,z10,MyTool\WObj:=wobj0;
9     MoveJ Target_80,v100,z10,MyTool\WObj:=wobj0;
10    MoveJ Target_90,v100,z10,MyTool\WObj:=wobj0;
11    MoveJ Target_100,v100,z10,MyTool\WObj:=wobj0;
12    MoveJ Target_110,v100,z10,MyTool\WObj:=wobj0;
13    MoveJ Target_120,v100,z10,MyTool\WObj:=wobj0;
14    MoveJ Target_130,v100,z10,MyTool\WObj:=wobj0;
15    MoveJ Target_140,v100,z10,MyTool\WObj:=wobj0;
16    MoveJ Target_150,v100,z10,MyTool\WObj:=wobj0;
```



```
17 MoveJ Target_160,v100,z10,MyTool\WObj:=wobj0;  
18 MoveJ Target_170,v100,z10,MyTool\WObj:=wobj0;  
19 ENDPROC
```

仿真中同样会遇到轨迹出现奇异点的问题，按照点位控制中所述的解决方案调整问题点的参数配置和姿态即可。

## 4.2 结果展示

实验结果如4.9，可以看出机器人可以按照设置的椭圆轨迹进行运动。

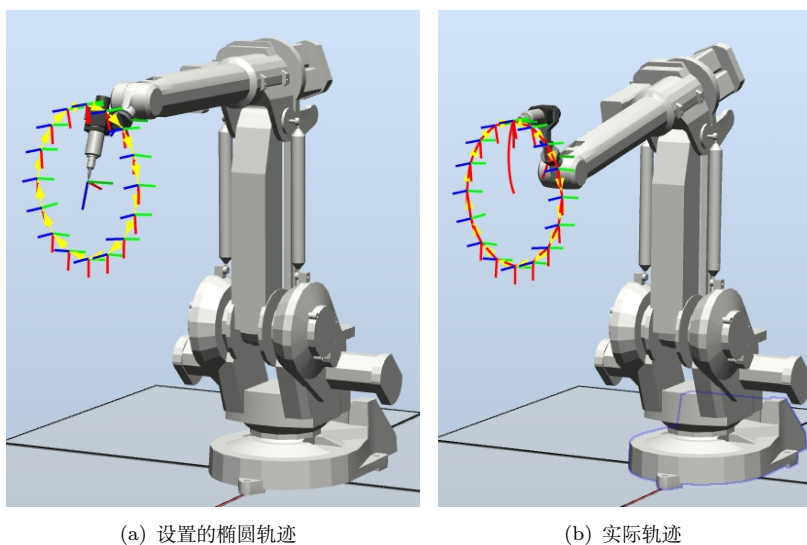


图 4.9: 位姿控制结果展示

## 5 机械臂跟踪喷漆与码垛

在自主设计的任务中，我设计了一套机械臂跟踪喷漆与码垛流水线。通过这一流水线的搭建，能够模拟实际工业流水线作业，探究多台机械臂协同运作的原理。

### 5.1 跟踪喷漆

实验中利用正方体物块作为喷漆对象，当物块随传送带移动时，机器人呢末端的喷嘴会跟踪物块的移动并进行喷漆。实验中的具体搭建过程如下。

#### 5.1.1 喷漆 Smart 组件

导入一个喷嘴模型，如图5.10所示。在“建模”中选择“Smart 组件”，添加一个“PaintApplicator”组件，如图5.11所示，并将该组件末端与喷嘴模型的末端重合。



图 5.10: 喷嘴模型



图 5.11: 喷漆 Smart 组件

在“建模”-“固体”-“矩形体”中，添加一个  $200\text{mm} \times 200\text{mm} \times 200\text{mm}$  的矩形体作为喷涂对象。对该 Smart 组件的属性进行如图5.12设置，对喷涂的对象、颜色和范围进行限制。在该 Smart 组件进行如图5.13的设计，通过 Paint 信号控制喷涂开始，Clear 信号进行清除。



图 5.12: 喷漆属性

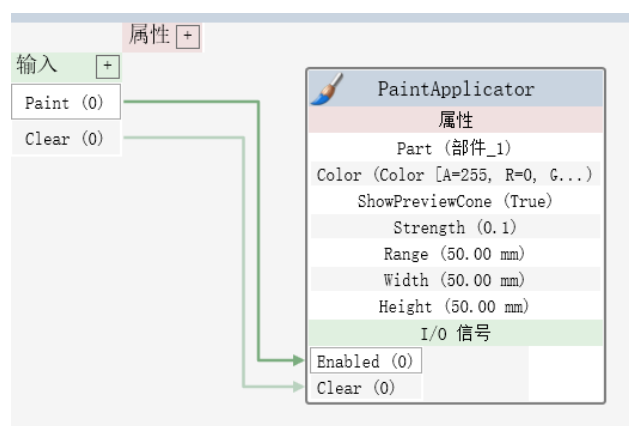


图 5.13: 喷漆 Smart 组件的设计

#### 5.1.2 传送带搭建

在工作站中导入 IRB-2600 机器人，作为喷漆的机器人。利用“固体”-“矩形体”创建一个  $4000\text{mm} \times 600\text{mm} \times 300\text{mm}$  的矩形体作为传送带，再创建一个  $200\text{mm} \times 200\text{mm} \times 200\text{mm}$  的矩形体作为传送带上的物块。利用“建模”-“创建输送带”将矩形体设置为传送带，并对传送带的参数进行设置，如图5.14所示。



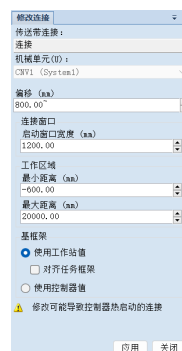
(a) 创建输送带



(b) 设置输送带参数



(c) 输送带对象



(d) 修改连接

图 5.14: 创建并设置输送带

工件坐标为  $wobj0$  下，在机器人机械原点位置示教得到机器人起始点，如图5.15。工件坐标为  $wobj\_cnv1$  下，将物块放在传送带的工作区域中，示教得到喷漆的起始点，如图5.16。创建一条空路径，放入以上两个点。

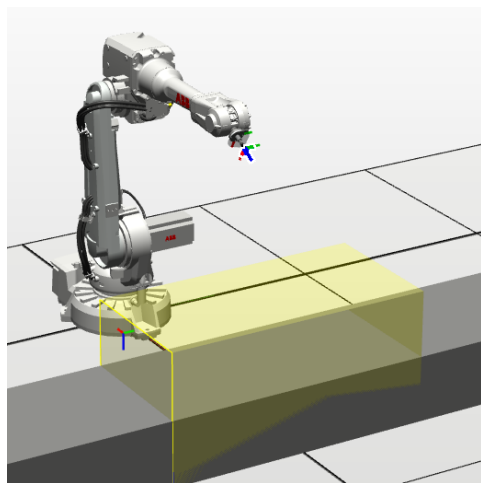


图 5.15: 机器人起始点

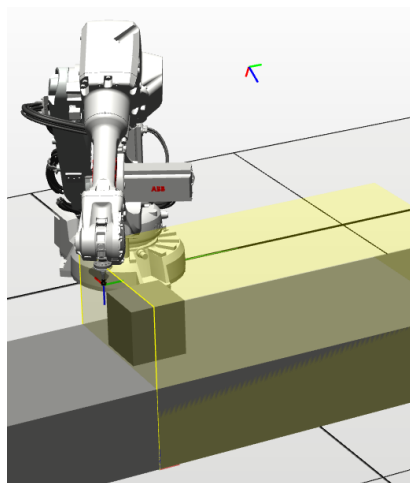


图 5.16: 喷漆起始点

在控制器中添加  $Piant0$  信号，用来控制喷漆的开始和结束。

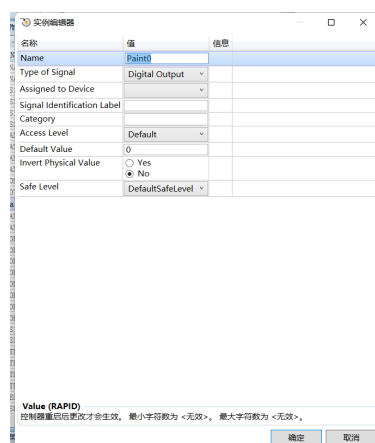


图 5.17: Paint0 信号设置

编写如下代码，即可完成对工件的跟踪喷涂。在主函数中，启动机械装置，每次物块进入工作区，启动喷漆的函数。在喷漆的函数中，首先移动至机器人起始点，然后移动到喷漆的起始点，将 Paint0 信号置为 1，启动喷涂。通过偏移位置的方法，在物块表面“之”字型喷涂。完成喷涂后，将 Paint0 信号置为 0，停止喷漆，机器人回到起始点。

```

1 PROC main()
2   ActUnit CNV1;
3   WHILE TRUE DO
4     WaitWObj wobj_cnv1;
5     Path_10;
6     DropWObj wobj_cnv1;
7
8   ENDWHILE
9 ENDPROC
10 PROC Path_10()
11   MoveL Target_20,v800,z100,Toolzz\WObj:=wobj0;
12   x:=0;
13   MoveL Offs(Target_10,0,x,0),v800,fine,Toolzz\WObj:=wobj_cnv1;
14   SetDO Paint0,1;
15   FOR i FROM 1 TO 5 DO
16     MoveL Offs(Target_10,0,x,0),v80,fine,Toolzz\WObj:=wobj_cnv1;
17     MoveL Offs(Target_10,200,x,0),v80,fine,Toolzz\WObj:=wobj_cnv1;
18     x:=x-20;
19     MoveL Offs(Target_10,200,x,0),v80,fine,Toolzz\WObj:=wobj_cnv1;
20     MoveL Offs(Target_10,0,x,0),v80,fine,Toolzz\WObj:=wobj_cnv1;
21     x:=x-20;
22
23   ENDFOR
24
25   MoveL Offs(Target_10,0,x,0),v80,fine,Toolzz\WObj:=wobj_cnv1;

```

```

26 MoveL Offs(Target_10,200,x,0),v80,fine,Toolzz\WObj:=wobj_cnv1;
27
28 SetDO Paint0,0;
29 MoveL Target_20,v800,z100,Toolzz\WObj:=wobj0;
30 ENDPROC

```

### 5.1.3 物块切换

由于“PaintApplicator”组件只能对于一个部件进行喷漆，而流水线中需要对传送带上的不断更新物块进行喷涂，所以实验中需要对传送带上的物块进行一定的操作。实验中的方法是对同一个物块进行喷涂，在物块进入和离开工作区时使用提前设定的物块进行切换。具体来说，传送带上更新的物块完全进入工作区的瞬间，将其删除并用“PaintApplicator”组件的喷涂对象进行替换；当完成喷涂后物块完全离开出传送带的瞬间，使用提前设定好的物块进行替换，喷涂对象清除上表面油漆跳转回工作区起点。示意图如图5.18所示，这样就可以实现流水线式的喷涂操作。实验中的具体搭建过程如下。

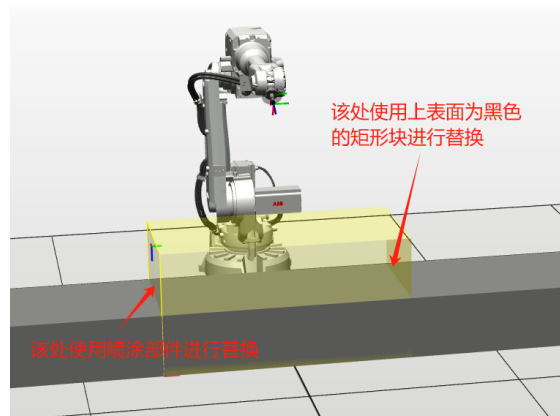


图 5.18: 流程示意图

创建一个 Smart 组件，其构成主要包括：

- PlaneSensor 面传感器：设置在在传送带起始位置、工作区域起始位置、传送带重点位置，用于检测物块。
- LinearMover 移动对象：用于物块的位置切换。
- Hide 隐藏对象：用于隐藏进入工作区域的物块。
- Show 显示对象：用于显示喷漆对象的物块。
- Source 拷贝对象：用于复制上表面完成喷漆的物块。
- Queue 队列：用于以队列的方式操控物块。

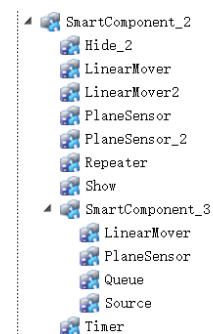


图 5.19: Smart 组件构成

该 Smart 组件的总体设计如图5.20所示。其中主要实现了物块进入工作区时的隐藏、喷漆对象在工作区中位置的切换并在工作区后的传送带上添加完成喷漆的物块队列。当面传感器检测到有物块进入工

作区域时，会将其隐藏，同时设定一定的间隔时间避免喷涂对象出发面传感器。根据物块刷新的频率和传送带速度，计算喷涂对象跳转的间隔时间，完成跳转后设定喷涂对象的速度与传送带的移动速度一致。

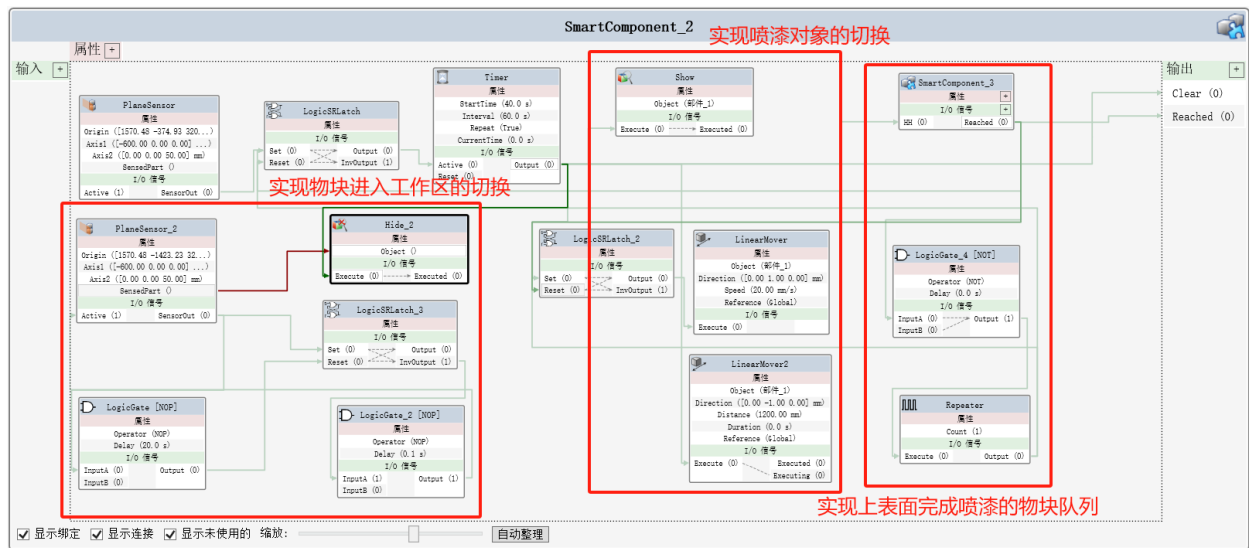


图 5.20: Smart 组件总体设计

添加完成喷漆的物块队列的这一部分，实验中添加了一个新的 Smart 组件完成这一操作，设计如图5.21。当喷漆物块离开工作区后，会向该 Smart 组件发送信号。接收到该信号后，该 Smart 组件会复制出一个上表面完成喷漆的物块至工作区末端位置，并添加到队列末尾。在传送带末端添加一个面传感器，当检测到物块到达传送带末端时会输出一个 Reached 信号。

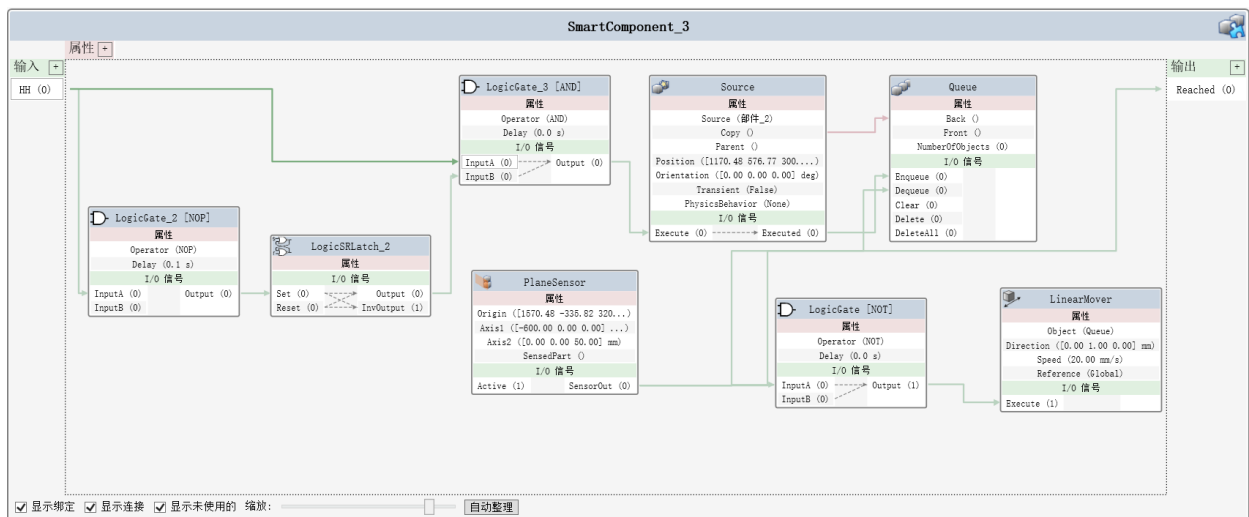


图 5.21: 完成喷漆的物块队列 Smart 组件设计

通过以上操作，机器人即可完成跟踪喷漆的任务，效果如图5.22所示。物块在传送段起始位置处刷新，进入工作区后机器人会随物块的移动进行跟踪喷涂。当物块到达传送带末端后，会向码垛机器人发送到达信号，码垛机器人将取走到达末端的物块。

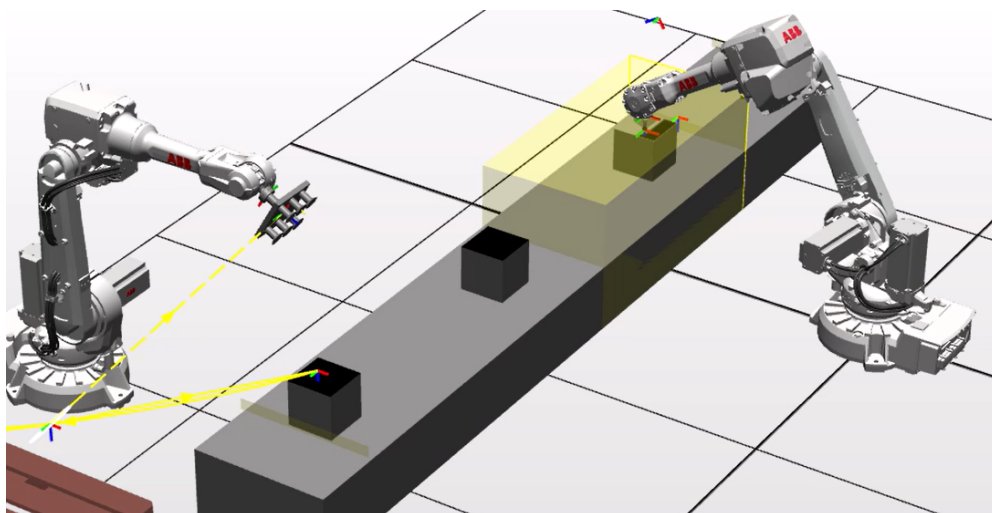


图 5.22: 跟踪喷涂效果图

## 5.2 码垛

码垛是指将货物按照一定的规则和顺序堆叠在托盘或其他载体上，以便于储存、运输和管理。码垛是物流和仓储作业中常见且重要的一环，可以有效提高空间利用率和工作效率。

实验中码垛的基本实现方式是：当喷涂得到的物块到达传送带末端，被面传感器检测到时，触发信号，机器人对检测到的物块进行抓取并放置于托盘的设定位置上。实验中搭建码垛的过程如下。

首先在工作站中，添加第二台 IRB-2600 机器人，并创建机器人系统。在机器人旁边合适的位置放置一个托盘，用于摆放堆叠的物体。利用“建模”下的“固体”，建模设计出机械臂末端的吸盘工具。该工具由一个矩形（ $200 \times 200 \times 20$ ）、六个圆柱（半径 15，高度 80）、六个锥形体（半径 15，高度 20）和一个圆柱（半径 17.5，高度 60）所组成，如图 5.24 所示。将该模型安装到机械臂末端法兰盘上。

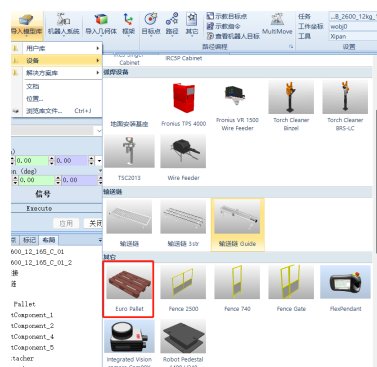


图 5.23: 托盘

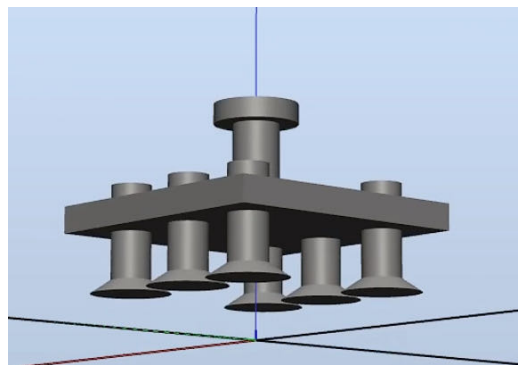


图 5.24: 吸盘工具

由于需要进行吸盘吸住物块进行移动操作，所以将吸盘设置为一个 Smart 组件。在该 Smart 组件中添加一个用于吸附物体的 Attacher 组件、一个解除吸附的 Detacher 组件和一个线传感器组件 LineSensor 组件。其中线传感器组件安装在一个吸盘上，用于检测吸盘是否与物块接触。需要注意的是，应当将吸盘部件设置为不可有传感器检测，否则线传感器会一直检测到物体。



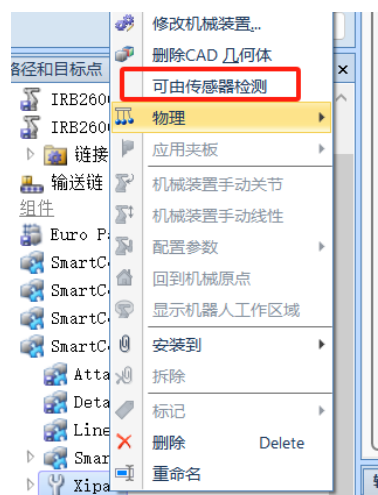


图 5.25: 取消勾选“可由传感器检测”

在该 Smart 组件中进行如图的设计，当 Attach 信号为 1 时，激活线传感器，Attacher 组件会吸附线传感器所检测到的物体；当 Attach 信号为 0 时，通过非门激活 Detacher 组件放下吸附的物块。

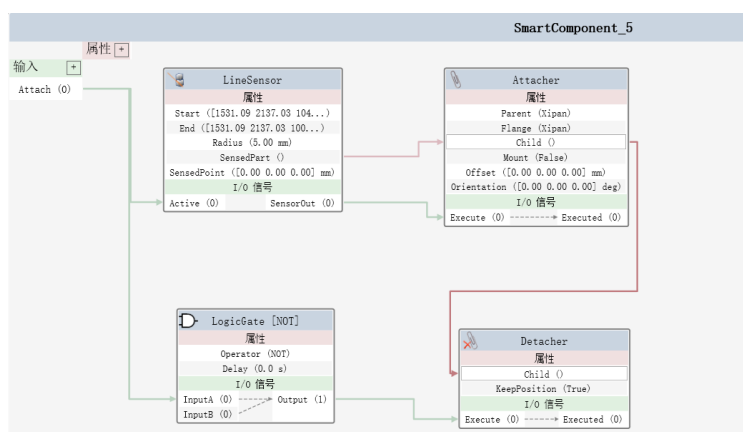


图 5.26: 吸盘 Smart 组件设计

使用“示教目标点”指令，设置机器人的起始点 Start、取物点 Pick\_object、中间点 Home 和第一个物块的放置点 First\_object。创建一条新的路径，将以上四个点放入该路径中并同步至 Rapid 中。

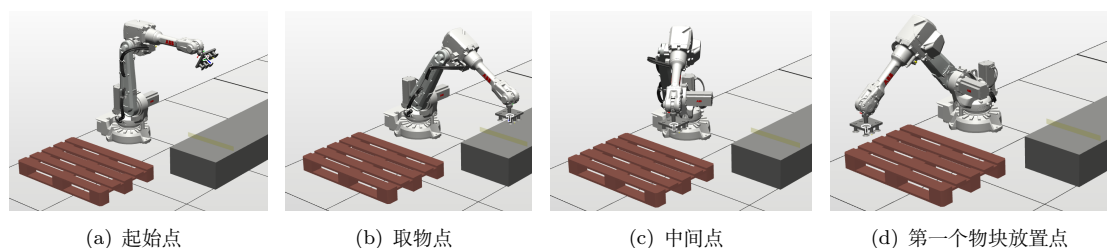


图 5.27: 码垛中的示教点

在当前机器人系统的控制中添加两个信号 di1Reached 和 do1Attach，分别表示物块到达取物点和吸盘吸附物块，具体设置如图5.28所示。



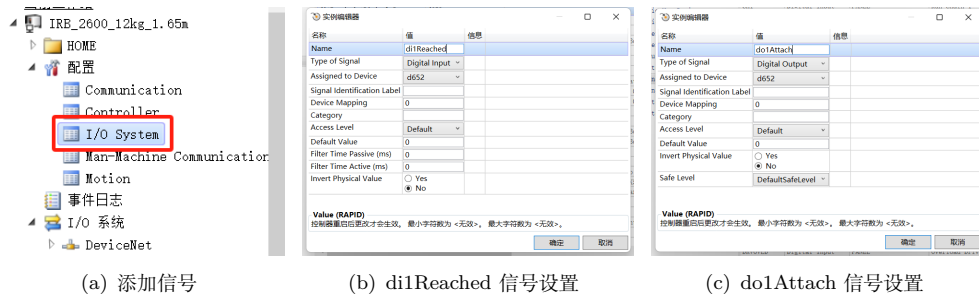


图 5.28: 添加信号

设置工作站逻辑，将此前设置的信号 Reached 连接 di1Reached，并将 do1Attach 连接 Attach 信号。经过该设置，当传送带末端的面传感器检测到物块时就会激活 Reached 和 di1Reached 信号，并启动吸盘对物块进行吸附。

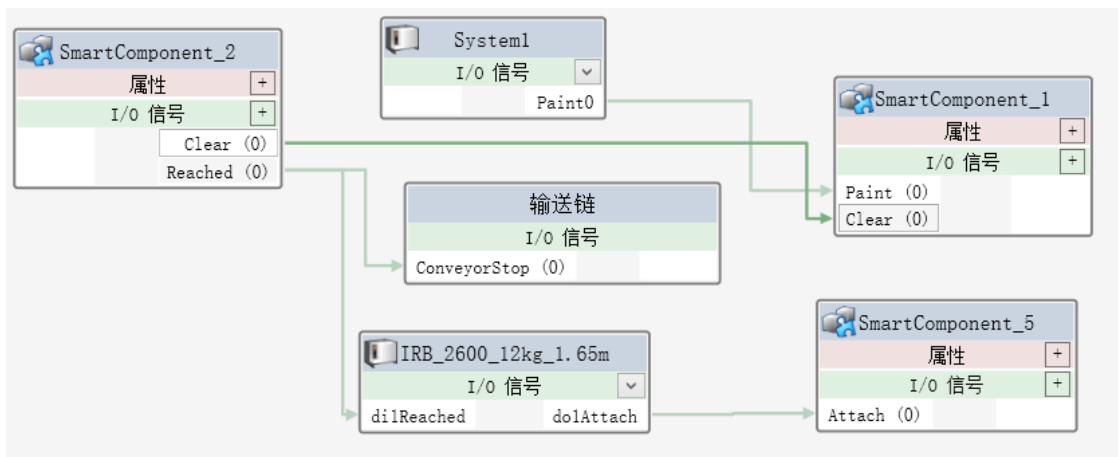


图 5.29: 工作站逻辑

编写主函数 main, 按照如下代码进行编写。首先调用 rInit 函数对信号进行初始化, 等待 di1Reached 信号为 1 后, 调用 rAttach 函数吸取物块, 调用 rSetPosition 函数设置当前放置物块的位置, 调用 rDetach 函数放下物块, 调用 rSeta 函数判断当前物块是否放满。

```

1 PROC main()
2   rInit;
3   WHILE TRUE DO
4     WaitDI di1Reached,1;
5     rAttach;
6     rSetPosition;
7     rDetach;
8     rSeta;
9   ENDWHILE
10 ENDPROC

```

rInit 函数的具体代码如下, 设置变量 a 为 1, 表示当前抓取的物块为第一个, 复位 do1Attach 信号, 机器人回到起始位置。

```
1 PROC rInit()
2     a:=1;
3     Reset do1Attach;
4     MoveJ Start,v1000,fine,Xipan\WObj:=wobj0;
5 ENDPROC
```

rAttach 函数的具体代码如下，移动到取物点上方 0.2m 处，再移动到取物点，激活 do1Attach 信号，等待一秒后回到取物点上方 0.2m 处。

```
1 PROC rAttach()
2     MoveJ offs(Pick_object,0,0,200),v1500,z10,Xipan\WObj:=wobj0;
3     MoveL Pick_object,v300,fine,Xipan\WObj:=wobj0;
4     set do1Attach;
5     WaitTime 1;
6     MoveJ offs(Pick_object,0,0,200),v1500,z10,Xipan\WObj:=wobj0;
7 ENDPROC
```

rSetPosition 函数的具体代码如下，实验中在托盘上摆放三排四列共 12 个物块，通过 a 的值记录当前摆放的物块是第几个，通过在第一个摆放点的基础上进行偏移设置放下物块的位置。

```
1 PROC rSetPosition()
2     TEST a
3     CASE 1:
4         place:=Offs(First_object,0,0,0);
5     CASE 2:
6         place:=Offs(First_object,200,0,0);
7     CASE 3:
8         place:=Offs(First_object,400,0,0);
9     CASE 4:
10        place:=Offs(First_object,600,0,0);
11    CASE 5:
12        place:=Offs(First_object,0,200,0);
13    CASE 6:
14        place:=Offs(First_object,200,200,0);
15    CASE 7:
16        place:=Offs(First_object,400,200,0);
17    CASE 8:
18        place:=Offs(First_object,600,200,0);
19    CASE 9:
20        place:=Offs(First_object,0,400,0);
21    CASE 10:
22        place:=Offs(First_object,200,400,0);
23    CASE 11:
```

```

24 place:=Offs(First_object,400,400,0);
25 CASE 12:
26 place:=Offs(First_object,600,400,0);
27 DEFAULT:
28 ENDTEST
29 ENDPROC

```

rDetach 函数的具体代码如下，移动到放置位置的上方 0.2m 处，重置 do1Attach 信号，放下物块，等待一秒后回到放置位置的上方 0.2m 处，移动到中间点，最后回到起始点。

```

1 PROC rDetach()
2 MoveJ offs(place,0,0,200),v1000,z100,Xipan\WObj:=wobj0;
3 MoveL place,v300,fine,Xipan\WObj:=wobj0;
4 Reset do1Attach;
5 WaitTime 1;
6 MoveJ offs(place,0,0,200),v1500,z200,Xipan\WObj:=wobj0;
7 MoveJ Home,v1000,z100,Xipan\WObj:=wobj0;
8 MoveJ Start,v1500,fine,Xipan\WObj:=wobj0;
9 ENDPROC

```

rSeta 函数的具体代码如下，每次移动物块时调用该函数会对变量 a 进行累加，当物块数量超过 12 个后，停止继续搬运物块。

```

1 PROC rSeta()
2 a:=a+1;
3 IF a>=13 THEN
4 a:=1;
5 stop;
6 ENDIF
7 ENDPROC

```

通过以上代码即可实现码垛操作，机器人在摆好 12 个物块后会停止继续摆放，摆放的效果如图5.30所示。

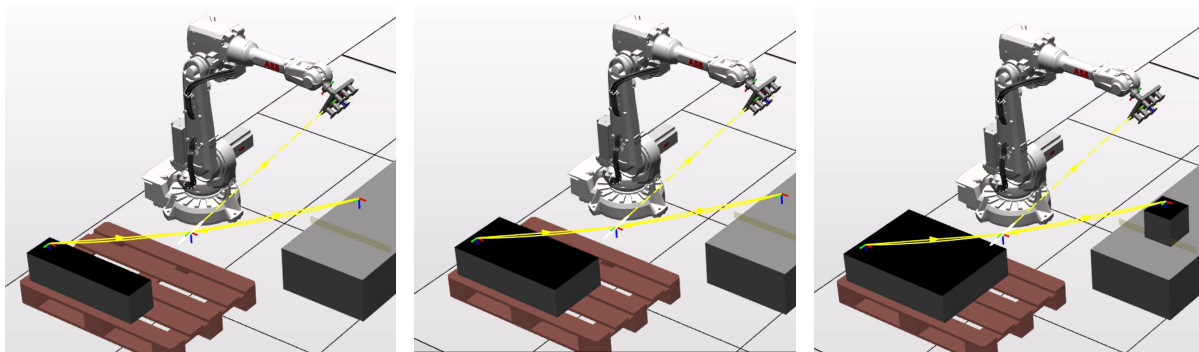


图 5.30: 码垛最终效果

## 6 总结

在《机器人学导论实验》的课程实验中，我通过使用 ABB RobotStudio 仿真软件，成功实现了机械臂的定点移动、轨迹运动和自动化流水线的设计。具体的实验总结如下：

- 点位控制：通过指定六个点的运动，实现了机械臂在空间中的定点移动。在这一部分中，我主要掌握了机器人的示教操作、创建轨迹、参数配置的方法，初步掌握了 Rapid 编程语言。
- 位姿控制：实现了机械臂沿着椭圆轨迹运动。在这一部分中，我主要掌握了使用曲线功能、表面功能和自动路径方法创建椭圆轨迹的操作。
- 自主任务设计：搭建了一个包含跟踪喷漆和码垛功能的流水线仿真环境，完成了机械臂在流水线上的自动化操作。在这一部分中，我主要掌握了 Smart 组件的构建方法，并提高了 Rapid 编程的能力。通过复杂的 Smart 组件设置和工作站逻辑中的信号传递，实验中完成了两个机器人的协调工作，模拟了现实中流水线的工作场景。

## 参考文献

- [1] ABB Robotics. ABB RobotStudio 操作手册 [EB/OL]. [2022.02.22].  
<https://search.abb.com/library/Download.aspx?DocumentID=3HAC032104-010&DocumentPartId=> .
- [2] 雷旭昌. 工业机器人 RobotStudio 仿真训练教程 [M]. 重庆大学出版社, 2018.