

# 控制器通讯协议与功能实现

Version 1.7.3



深圳辰视智能科技有限公司

SHENZHEN COSMOSVISION TECHNOLOGY CO., LTD

Copyright © 深圳辰视智能科技有限公司

All rights reserved.

## 目录

控制器通讯协议与功能实现.....	1
一、    PC 软件对接控制器条件.....	4
二、    X 机器人.....	4
1.    TCP/IP 通讯.....	4
2.    坐标系.....	5
三、    三维视觉引导系统通讯协议.....	5
1.    TCP/IP 通讯.....	5
2.    标定存图协议.....	6
3.    识别和抓取协议.....	6
4.    示教抓取协议.....	8
5.    设置模型协议.....	9
6.    轨迹跟踪协议.....	10
7.    欧拉系统测试命令.....	11
四、    功能实现.....	17
1.    基础功能.....	17
2.    标定功能.....	18
3.    示教抓取功能.....	19
4.    抓取功能.....	20
5.    分区域抓取功能（拓展功能）.....	20
6.    轨迹引导功能.....	22

## 修订历史记录

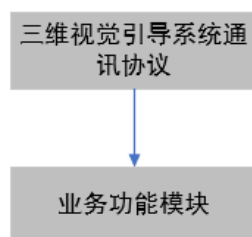
A - 增加 M - 修订 D - 删除

变更版本号	日期	变更类型 (A*M*D)	修改人	摘 要	备注
1.0	2018/12/10	A	陈先开	通讯协议第一版	
1.5	2019/08/10	M	陈先开	通讯协议第二版	
1.6	2019/12/25		张焱	通讯协议第三版	
1.7	20200520		邓亮	添加 RecgMul 及 RecgGraspMul 命令	
1.7.1	20200605	M	邓亮	添 加 AddTrackPoint 及 RemoveTrackPoint 命令	
1.7.2	20200714	M	邓亮	添加机器人欧拉系统测试命令 EulerTest	
1.7.3	20220209	M	邓亮	添加标定存图命令 CalShootSave	

## 一、PC 软件对接控制器条件

- (1) 配备机器人本体工程师与我司视觉开发人员对接。
- (2) 开放机器人本体功能：TCP/IP 通讯端口。
- (3) 提供机器人开发平台相关的开发文档。

对接步骤,如下图所示,首先对接三维视觉引导系统通讯协议,然后对接业务功能模块。



## 二、X 机器人

机器人基本信息如下表所示。

控制器	名称	控制柜/Cabinet
	型号	EC2-S6-P1-G4-G0-G2
	额定电压	AC220V
	额定电流	18A
	额定功率	4KW
机器人本体	名称	工业机器人
	型号	ER20-060
	最大负载	20KG
	设备重量	220KG
	额定功率	5KW
	额定电压	220V

### 1. TCP/IP 通讯

TCP/IP 通讯：支持连接和断开

字符结束校验符号：\r\n(ascii 码为：0D 0A)

视觉字符串收发

格式：

接收数据：

<Read>OpenVideo</Read>

发送数据：

<Sensor><String>YES\_OpenVideo</String></Sensor>

2. 坐标系

机器人采用动坐标系的欧拉角 ZYX（右乘）。发送欧拉角顺序 RZ,RY,RX，面板欧拉角 ABC 显示为 RX（A）,RY（B）,RZ（C）。

发送欧拉角顺序 RZ,RY,RX

**注意：**视觉系统采用的欧拉角旋转矩阵的计算公式采用欧拉角坐标系 XYZ（左乘）。

三、 三维视觉引导系统通讯协议

1. TCP/IP 通讯

建立 TCP/IP 客户端通讯，包含连接和断开功能。

执行顺序	协议格式	说明
机器人功能函数	CsvReconnect,建立机械臂和视觉系统的通讯连接	例子: CsvReconnect。连接视觉系统，需要提供视觉系统 PC 的 IP 和端口号，例如：192.168.0.22.6200.
机器人功能函数	CsvDisconnect,断开机械臂和视觉系统的通讯连接	例子: CsvDisconnect。断开机械臂和视觉系统的连接，需要提供视觉系统 PC 的 IP 和端口号，例如：192.168.0.22.6200.

B. 机器人从视觉系统设置相关参数或状态

执行顺序	协议格式	说明
------	------	----

机器人发送命令	字符格式 SetParam,id, value1,value2 设置特定值。index 表示索引号	例子: SetParam, 0,100,210
机器人接收命令	成功: YES_SetParam 失败 NO_SetParam	成功: YES_SetParam 失败 NO_SetParam

## 2. 标定存图协议

执行顺序	协议格式	说明
机器人发送命令	字符格式: CalShootSave,X±坐标值,Y±坐标值,Z±坐标值,RX±坐标值,RY±坐标值,RZ±坐标值	例子: CalShootSave,X+391.737,Y-468.884,Z-133.991,RX-84.332,RY-61.617,RZ-146.785
视觉系统反馈指令	成功标识符: YES_CalShootSave 失败标识符: NO_CalShootSave	操作会返回相应的标识符 成功标识符: YES_CalShootSave 失败标识符: NO_CalShootSave

## 3. 识别和抓取协议

### A. 打开视频

执行顺序	协议格式	说明
机器人发送命令	字符格式: OpenVideo	例子: OpenVideo
视觉系统反馈指令	成功标识符: YES_OpenVideo 失败标识符: NO_OpenVideo	操作会返回相应的标识符 成功标识符: YES_OpenVideo 失败标识符: NO_OpenVideo

### B. 关闭视频

执行顺序	协议格式	说明
机器人发送命令	字符格式: StopVideo	例子: StopVideo
视觉系统反馈指令	成功标识符: YES_StopVideo 失败标识符: NO_StopVideo	操作会返回相应的标识符 成功标识符: YES_StopVideo 失败标识符: NO_StopVideo

## C. 识别目标

执行顺序	协议格式	说明
机器人发送命令	字符格式： Recg,X±坐标值,Y±坐标值,Z±坐标值,RX±坐标值,RY±坐标值,RZ±坐标值,模型号	例子： Recg,X+238.548,Y-554.296,Z+10.588,RX-159.791,RY-50.461,RZ-92.328,M0
视觉系统反馈指令	字符格式： 成功： X±坐标值,Y±坐标值,Z±坐标值,RX±坐标值,RY±坐标值,RZ±坐标值,模型号 失败： NO_Recg,错误信息	成功： X+375.757,Y-397.399,Z-95.883,RX+10.835,RY+20.367,RZ+126.380,M0 失败： NO_Recg,CsvMaster 识别不到模型

## D. 识别输出多个目标

执行顺序	协议格式	说明
机器人发送命令	字符格式： RecgMul,X±坐标值,Y±坐标值,Z±坐标值,RX±坐标值,RY±坐标值,RZ±坐标值,模型号	例子： RecgMul,X+238.548,Y-554.296,Z+10.588,RX-159.791,RY-50.461,RZ-92.328,M0
视觉系统反馈指令	字符格式： 成功： X±坐标值,Y±坐标值,Z±坐标值,RX±坐标值,RY±坐标值,RZ±坐标值; X±坐标值,Y±坐标值,Z±坐标值,RX±坐标值,RY±坐标值,RZ±坐标值;模型号,识别结果总数 失败： NO_RecgMul,错误信息	成功： X+375.757,Y-397.399,Z-95.883,RX+10.835,RY+20.367,RZ+126.380; X+375.757,Y-397.399,Z-95.883,RX+10.835,RY+20.367,RZ+126.380; M0,N2 失败： NO_RecgMul,CsvMaster 识别不到模型

## E. 识别抓取目标

执行顺序	协议格式	说明
机器人发送命令	字符格式： RecgGrasp,X±坐标值,Y±坐标值,Z±坐标值,RX±坐标值,RY±坐标值,RZ±坐标值,模型号	发送： RecgGrasp,X+238.548,Y-554.296,Z+10.588,RX-159.791,RY-50.461,RZ-92.328,M0

视觉系统反馈指令	字符格式： 成功： X±坐标值,Y±坐标值,Z±坐标值,RX±坐标值,RY±坐标值,RZ±坐标值,解数目,模型索引号,工具号,抓取面,识别出工件的个数 失败： NO_RecgGrasp, 错误信息  如开启了 Box 点云冲突判定功能并且配置相关的 Box, 返回结果为： NO_RecgGrasp,BoxPointsCol	返回： 成功： X+371.758,Y-338.203,Z+46.015,RX+160.280,RY-10.317,RZ-18.695,5,M0,T0,PICK1,N2 失败： NO_RecgGrasp, CsvMaster 识别模型失败

## F. 识别抓取多个目标

执行顺序	协议格式	说明
机器人发送命令	字符格式： RecgGraspMul,X±坐标值,Y±坐标值,Z±坐标值,RX±坐标值,RY±坐标值,RZ±坐标值,模型号	发送： RecgGraspMul,X+238.548,Y-554.296,Z+10.588,RX-159.791,RY-50.461,RZ-92.328,M0
视觉系统反馈指令	字符格式： 成功： X±坐标值,Y±坐标值,Z±坐标值,RX±坐标值,RY±坐标值,RZ±坐标值;模型索引号,工具号,示教抓取索引, 识别结果总数 失败： NO_RecgGraspMul, 错误信息	返回： 成功： X+371.758,Y-338.203,Z+46.015,RX+160.280,RY-10.317,RZ-18.695; X+371.758,Y-338.203,Z+46.015,RX+160.280,RY-10.317,RZ-18.695;M0,T0,PICK0,N2 失败： NO_RecgGraspMul, CsvMaster 识别模型失败

## 4. 示教抓取协议

### A. 添加示教抓取点

执行顺序	协议格式	说明
机器人发送命令	字符格式： AddGrasp, X±坐标值,Y±坐标值,Z±坐标值,RX±坐标值,RY±坐标值,RZ±坐标值,工具坐标号, 工件 X±坐标值,工件 Y±坐标值,工	例子： AddGrasp,X+218.610,Y-435.315,Z+76.377,RX+101.968,RY+42.451,RZ+103.450,T0,X+218.610,Y-



	件 Z±坐标值,工件 RX±坐标值,工件 RY±坐标值,工件 RZ±坐标值,模型号 (尽量使用法兰坐标进行示教与抓取)	435.315,Z+76.377,RX+101.968,RY+42.451,RZ+103.450,M0
视觉系统反馈指令	成功标识符: YES_AddGrasp 失败标识符: NO_AddGrasp	添加抓取点操作会返回相应的标识符 成功: YES_AddGrasp 失败: NO_AddGrasp

## B. 删除示教抓取点

执行顺序	协议格式	说明
机器人发送命令	字符格式: RemoveGrasp,-1 Index 表示删除示教的标定点索引	例子: RemoveGrasp,-1
视觉系统反馈指令	成功标识符: YES_RemoveGrasp 失败标识符: NO_RemoveGrasp	删除抓取点操作会返回相应的标识符 成功标识符: YES_RemoveGrasp 失败标识符: NO_RemoveGrasp

## 5. 设置模型协议

### A. 切换模型协议

执行顺序	协议格式	说明
机器人发送命令	字符格式: M 模型 ID,序号,识别文件索引 ID	例子: SetModelParam,M0,1,1
视觉系统反馈指令	字符格式: 成功: YES_SetModelParam 失败: NO_SetModelParam,错误信息	成功: YES_SetModelParam 失败: NO_SetModelParam,模型错误/路径不正确

### B. 切换模型协议

执行顺序	协议格式	说明
机器人发送命令	字符格式: 模型名	例子: SwitchModel,modelName
视觉系统反馈指令	字符格式: 成功:	成功: YES_SwitchModel

	YES_SwitchModel 失败： NO_SwitchModel,错误信息	失败： NO_SwitchModel,模型错误/路径不正确
--	---	----------------------------------

## 6. 轨迹跟踪协议

### A. 非插值模式

执行顺序	协议格式	说明
机器人发送命令	字符格式： Track,ICP0  说明：此模式生成的引导轨迹点位数  量等于配置表的点数	例子： Track,ICP0
视觉系统反馈指令	字符格式： 成功： X±坐标值,Y±坐标值,Z±坐标值,RX±坐标值,RX±坐标值,RZ±坐标值,...,X±坐标值,Y±坐标值,Z±坐标值,RX±坐标值,RX±坐标值,RZ±坐标值 失败： NO_Track,错误信息	成功： X+371.758,Y-338.203,Z+46.015,RX+160.280,RX-10.317,RZ-18.695,...,X+371.758,Y-338.203,Z+46.015,RX+160.280,RX-10.317,RZ-18.695 失败： NO_Track,模型错误/路径不正确

### B. 插值模式

执行顺序	协议格式	说明
机器人发送命令	字符格式： Track,ICP1  说明：此模式生成的引导轨迹点位数  量等于（配置表的点数-2）x2	例子： Track,ICP1
视觉系统反馈指令	字符格式： 成功： X±坐标值,Y±坐标值,Z±坐标值,RX±坐标值,RX±坐标值,RZ±坐标值,...,X±坐标值,Y±坐标值,Z±坐标值,RX±坐标值,RX±坐标值,RZ±坐标值 失败： NO_Track,错误信息	成功： X+371.758,Y-338.203,Z+46.015,RX+160.280,RX-10.317,RZ-18.695,...,X+371.758,Y-338.203,Z+46.015,RX+160.280,RX-10.317,RZ-18.695

	失败： NO_Track,错误信息	失败： NO_Track,模型错误/路径不正确
--	----------------------	----------------------------

### C. 添加跟踪点

执行顺序	协议格式	说明
机器人发送命令	字符格式： AddTrackPoint, X±坐标值,Y±坐标值,Z±坐标值,RX±坐标值,RX±坐标值,RZ±坐标值,工具坐标号, 工件 X±坐标值,工件 Y±坐标值,工件 Z±坐标值,工件 RX±坐标值,工件 RY±坐标值,工件 RZ±坐标值,模型号  说明：模型号很重要，一定要配置正确	例子： AddTrackPoint,X+218.610,Y-435.315,Z+76.377,RX+101.968,RY+42.451,RZ+103.450,T0,X+218.610,Y-435.315,Z+76.377,RX+101.968,RY+42.451,RZ+103.450,M0
视觉系统反馈指令	字符格式： 成功： YES_AddTrackPoint 失败： NO_AddTrackPoint,错误信息	

### D. 删除上一个跟踪点

执行顺序	协议格式	说明
机器人发送命令	字符格式： RemoveTrackPoint, 模型号  说明：一定要指定模型号，否则会失败	例子： RemoveTrackPoint,M0
视觉系统反馈指令	字符格式： 成功： YES_RemoveTrackPoint 失败： NO_RemoveTrackPoint,错误信息	

## 7. 欧拉系统测试命令

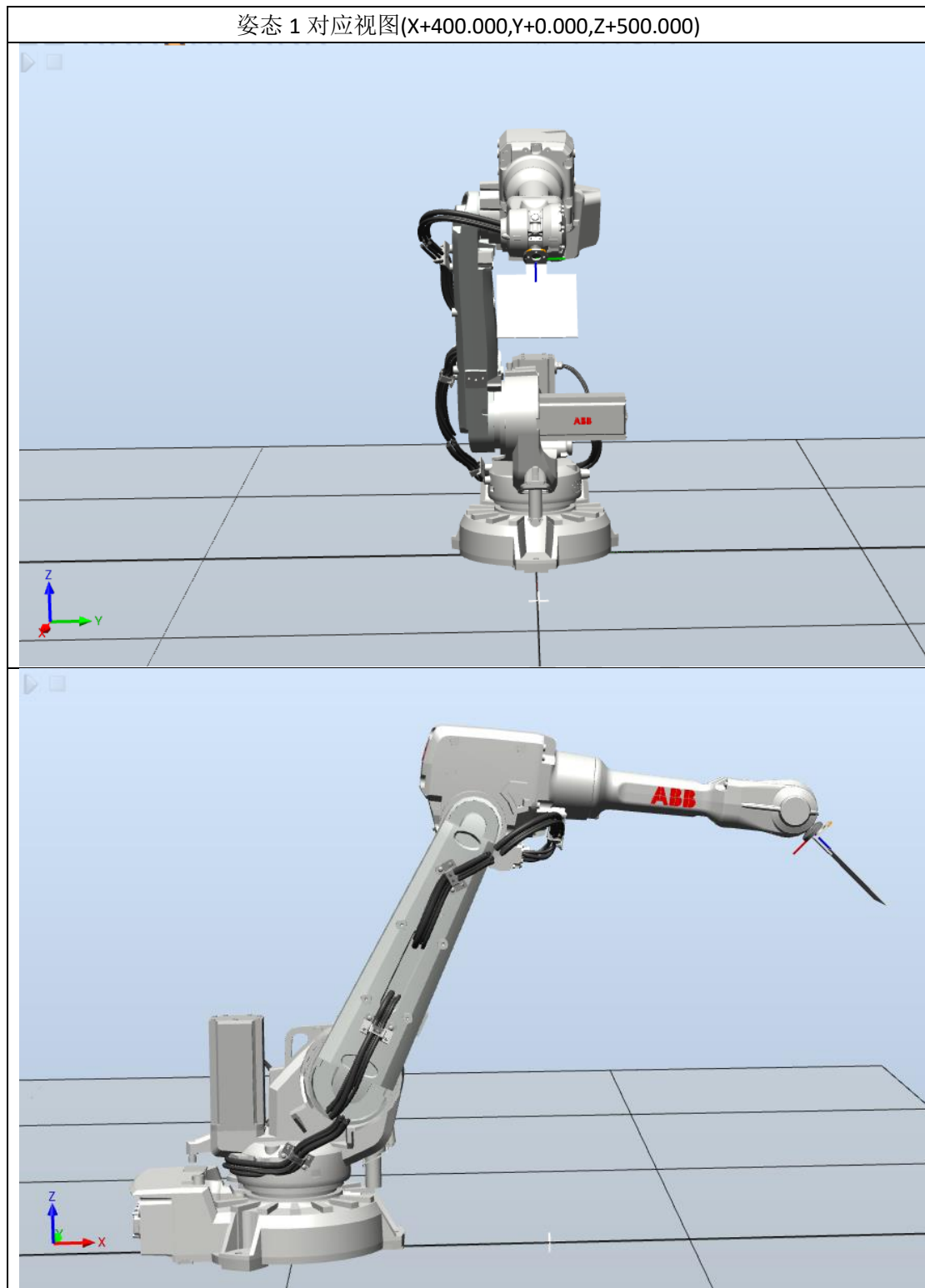
该系列命令主要时为了测试新增加机器人的欧拉系统，切换到不同的机器人时，相同的测试命令可能输出不同的结果

#### A、输出 4 个指定方向的机器人坐标

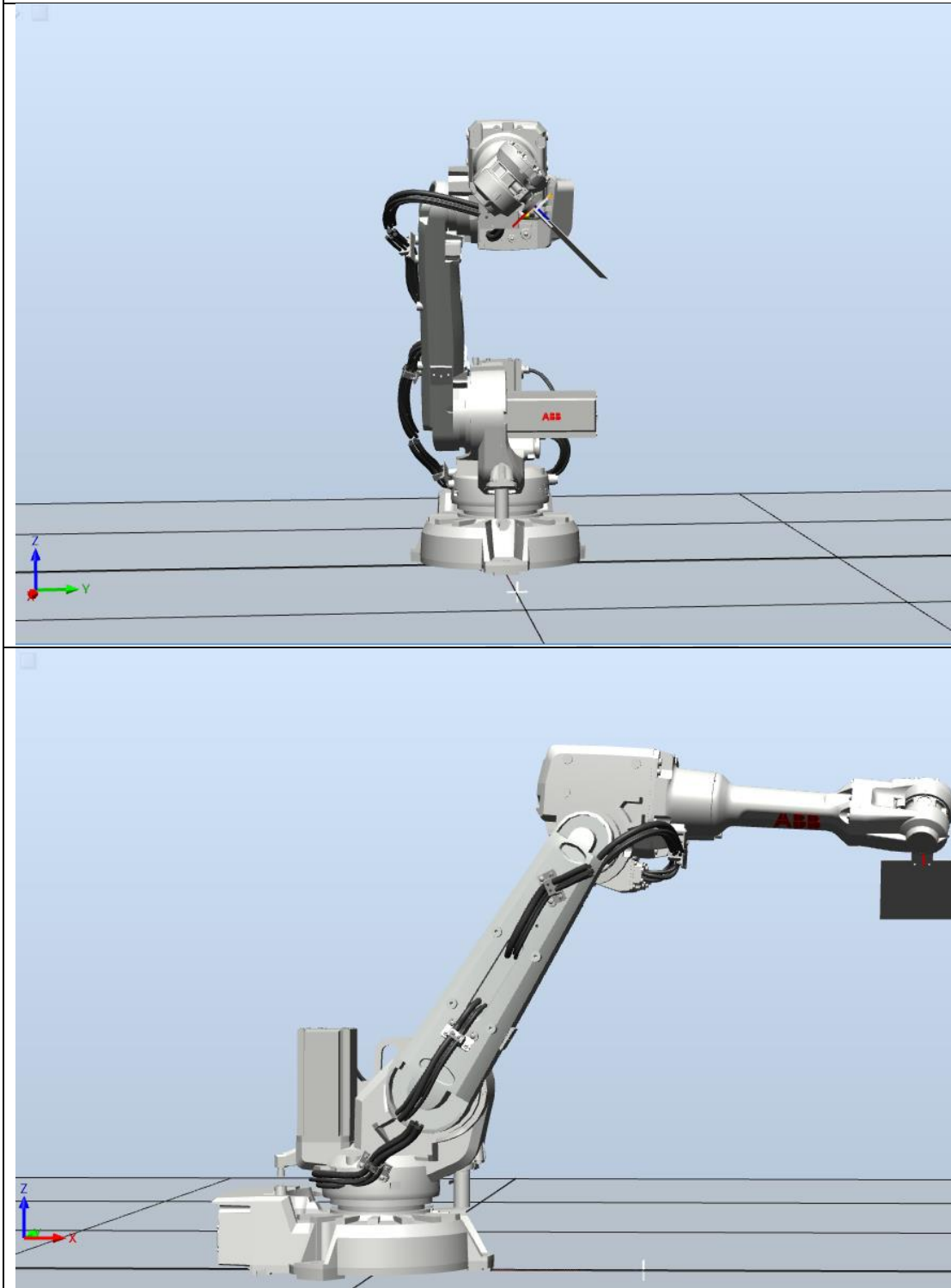
姿态序号	姿态 Y 轴方向	姿态 Z 轴方向
1	Y 轴方向[0,1,0]	Z 轴方向[0.707,0,-0.707]

2	Y 轴方向[-1,0,0]	Z 轴方向[0,0.707,-0.707]
3	Y 轴方向[0,-1,0]	Z 轴方向[-0.707,0,-0.707]
4	Y 轴方向[1,0,0]	Z 轴方向[0,-0.707,-0.707]

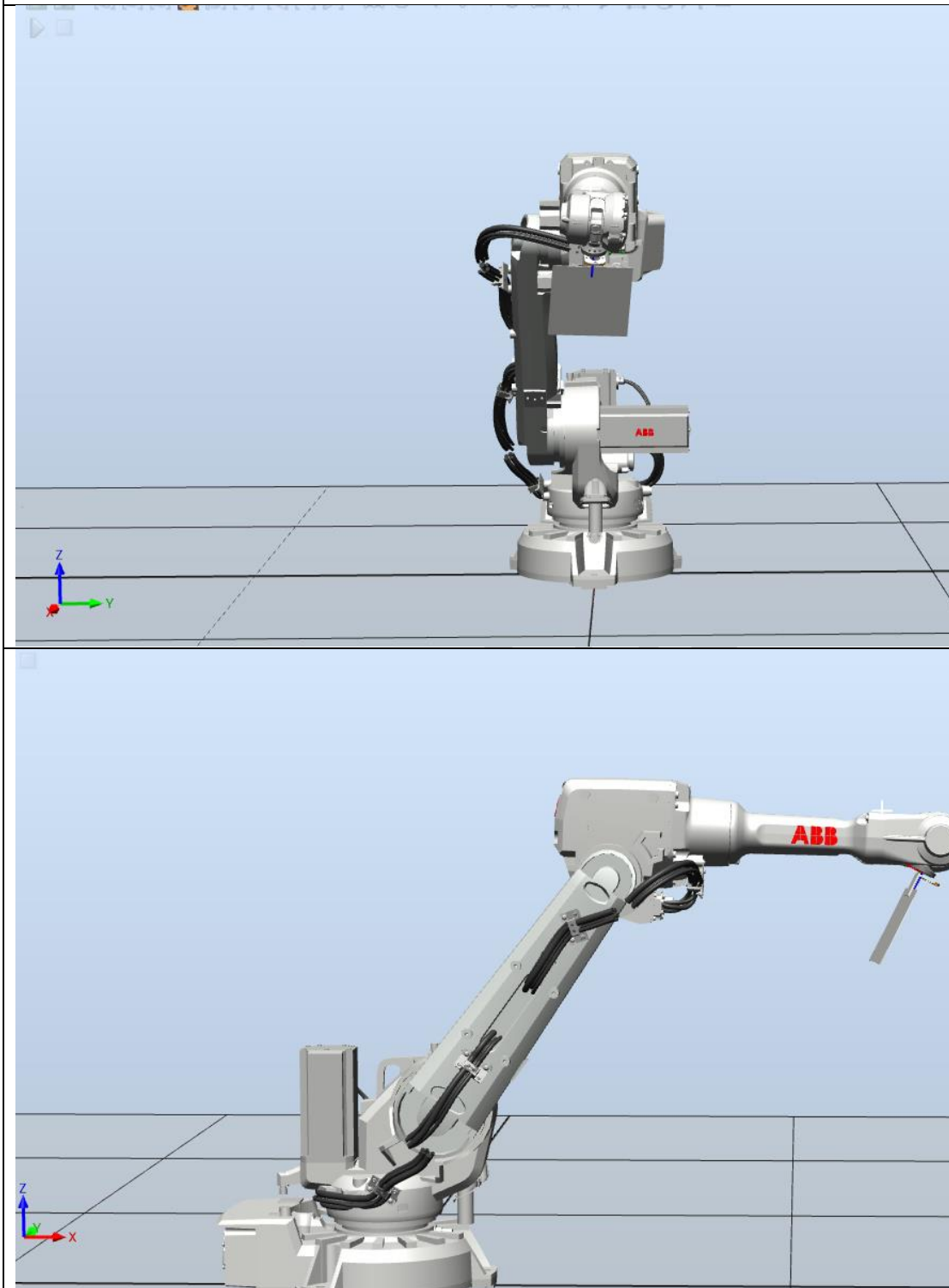
COSMOS VISION



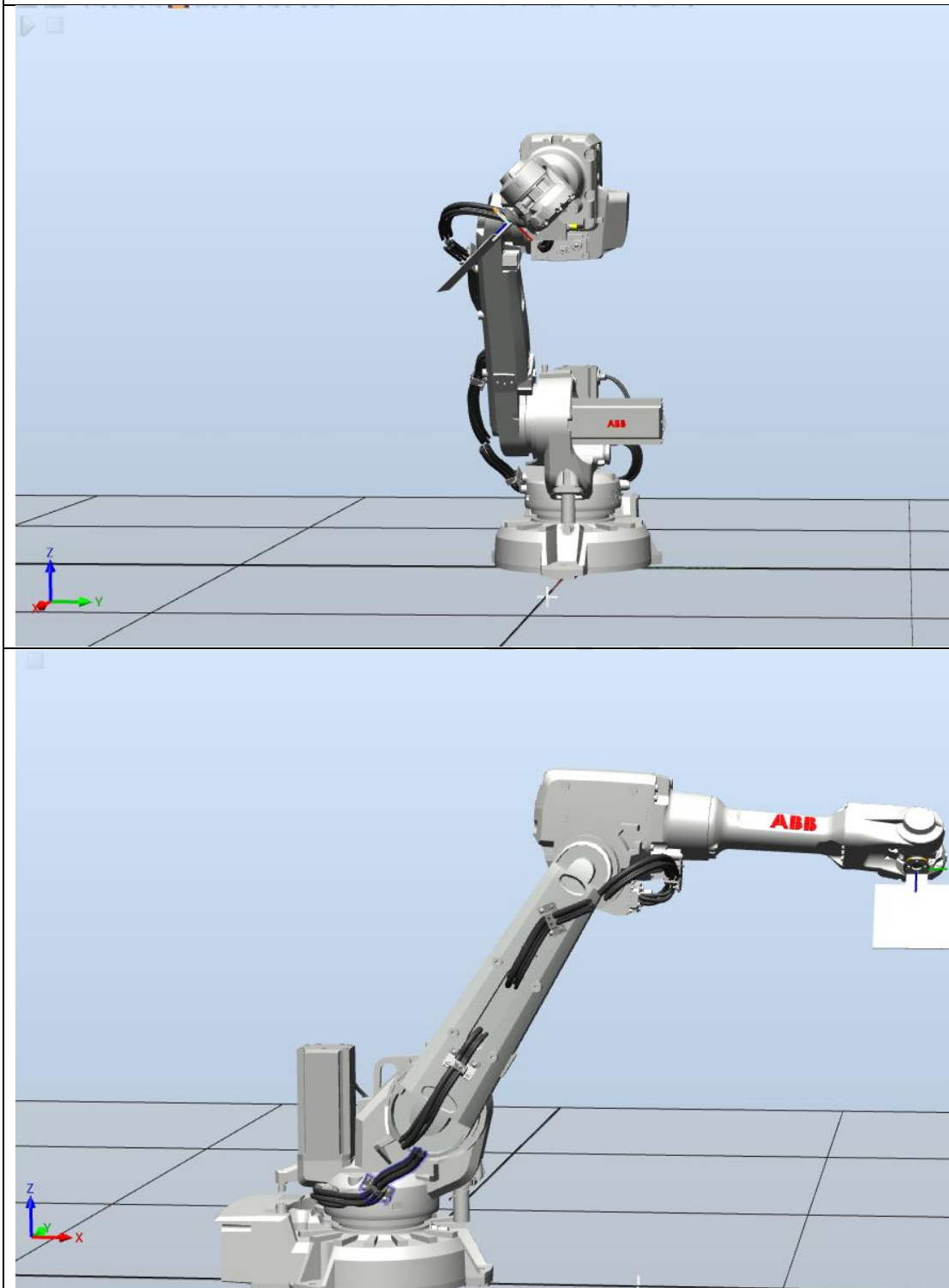
姿态 2 对应视图(X+400.000,Y+0.000,Z+500.000)



姿态 3 对应视图(X+400.000,Y+0.000,Z+500.000)



姿态 4 对应视图(X+400.000,Y+0.000,Z+500.000)





协议格式：

执行顺序	协议格式	说明
机器人发送命令	EulerTest,X 坐标值, Y 坐标值, Z 坐标值	例子: EulerTest, +400.000,+0.000,+500.000,
视觉系统反馈指令	4 个点的位置姿态值: X±坐标值,工件 Y±坐标值,工件 Z±坐标值, 工件 RX±坐标值,工件 RY±坐标值,工件 RZ±坐标值 位置信息为视觉信息从机器人或仿真端收到的位置	例子: X+400.000, Y+0.000, Z+500.000, RX+180.000, RY+45.000, RZ+180 X+400.000, Y+0.000, Z+500.000, RX-180.000, RY+45.000, RZ-90.000 X+400.000, Y+0.000, Z+500.000, RX+180.000, RY+45.000, RZ-0.000 X+400.000, Y+0.000, Z+500.000, RX+180.000, RY+45.000, RZ+90.000

B、输出指定机器人坐标的旋转矩阵与位置值

执行顺序	协议格式	说明
机器人发送命令	EulerTest, X 坐标值, Y 坐标值, Z 坐标值, RX 坐标值, RY 坐标值, RZ 坐标值	例子: EulerTest, +610.499, -312.712, +68.387, +180.000, +45.000, +180.000
视觉系统反馈指令	X 坐标值, Y 坐标值, Z 坐标值, RX 坐标值, RY 坐标值, RZ 坐标值, r00 值, r10 值, r20 值, r01 值, r11 值, r21 值, r02 值, r12 值, r22 值	例子: +610.499, -312.712, +68.387, -0.707, +0.000, -0.707, +0.000, +1.000, +0.000, +0.707, +0.000, -0.707

C、输出指定位置与旋转矩阵的机器人坐标（与前一个命令刚好相反）

执行顺序	协议格式	说明
机器人发送命令	EulerTest, X 坐标值, Y 坐标值, Z 坐标值, RX 坐标值, RY 坐标值, RZ 坐标值, r00 值, r10 值, r20 值, r01 值, r11 值, r21 值, r02 值, r12 值, r22 值	例子: EulerTest, +610.499, -312.712, +68.387, -0.707, +0.000, -0.707, +0.000, +1.000, +0.000, +0.707, +0.000, -0.707
视觉系统反馈指令	X±坐标值,Y±坐标值,Z±坐标值,RX±坐标值,RY±坐标值,RZ±坐标值	例子: X+610.499, Y-312.712, Z+68.387, RX+180.000, RY+45.000, RZ+180.000,

四、 功能实现

1. 基础功能

包括获取值、设定值、重新连接视觉系统和断开连接视觉系统等。

## 2. 标定功能

包括连续拍照，停止拍照，添加标定点，删除标定点，自动标定运行

### 标定流程：

- **添加标定点**

- 打开视频流

- 机器人发：

OpenVideo

- 机器人收：

YES\_OpenVideo

- 移动机器人到合理的位置

- 记录机器人位置到机器人位置寄存器：

- 记录合适位置点数（三十个左右）

- 停止视频流

- 机器人发：

StopVideo

- 机器人收：

YES\_StopVideo

- **自动运行标定**

- 机器人运动到指定的点位

- 拍照保存

- 机器人发：

CalShootSave,X+391.737,Y-468.884,Z- 133.991,RX-84.332,RY-61.617,RZ-146.785

- 机器人收：

YES\_CalShootSave

### 3. 示教抓取功能

包含单次识别功能，添加抓取示教点，删除上一个抓取示教点。

#### 添加示教抓取点流程：

- 添加示教抓取点
  - 启动识别工件。
    - 机器人发：  
Recg,X+238.548,Y-554.296,Z+10.588,RX-159.791,RY-50.461,RZ-92.328,M0
    - 机器人收：  
X+375.757,Y-397.399,Z-95.883,RX+10.835,RY+20.367,RZ+126.380,M0
  - 遥控机器人到适当的抓取位置。
  - 添加抓取示教点。
    - 机器人发：  
AddGrasp,X+238.548,Y-554.296,Z+10.588,RX-159.791,RY-50.461,RZ-92.328,T0,X+375.757,Y-397.399,Z-95.883,RX+10.835,RY+20.367,RZ+126.380,M0
    - 机器人收：  
YES\_AddGrasp
- 删除示教抓取点（该功能不计入流程中）
  - 删除上一个标定点。可连续删除最近添加的示教点。
  - 机器人发：RemoveGrasp,-1
  - 机器人收：YES\_RemoveGrasp

#### 示教抓取方法：

首先，将一个工件放置视觉视野范围内，识别工件，获取视觉系统识别工件的坐标系：

将机械臂移动到指定抓取位置，发送添加示教点功能 ( $AddGrasp, X+238.548, Y-554.296, Z+10.588, RX-159.791, RY-50.461, RZ-92.328, T0, X+375.757, Y-397.399, Z-95.883, RX+10.835, RY+20.367, RZ+126.380, M0$ )，完成示教点定义。机器人返回原点，发送识别抓取目标功能 ( $RecgGrasp, X+238.548, Y-554.296, Z+10.588, RX-159.791, RY-50.461, RZ-92.328, M0$ )。

#### 4. 抓取功能

- 抓取识别

- 机器人发：

- $RecgGrasp, X+238.548, Y-554.296, Z+10.588, RX-159.791, RY-50.461, RZ-92.328, M0$

- 机器人收：

- 成功：

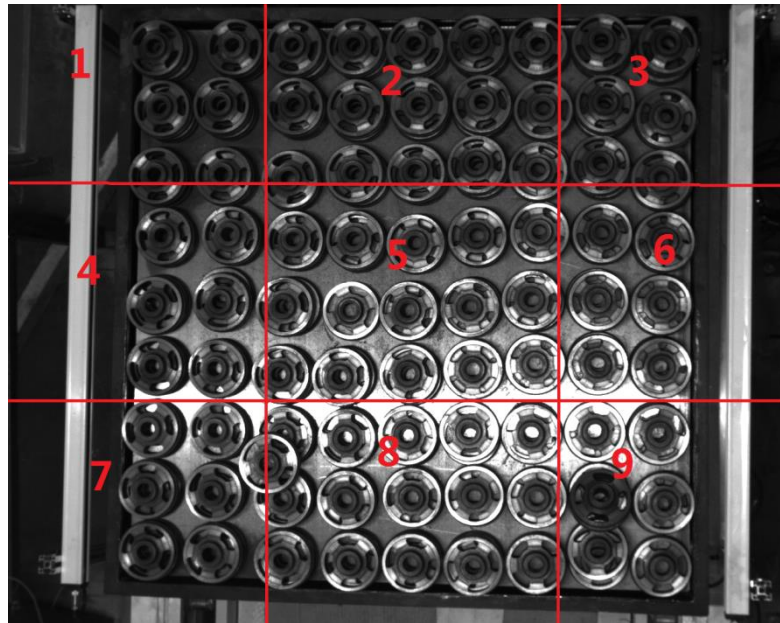
- $X+371.758, Y-338.203, Z+46.015, RX+160.280, RY-10.317, RZ-18.695, M0, T0, PICK1, N1$

- 失败：

- $NO\_RecgGrasp, CsvMaster$  识别模型失败

#### 5. 分区域抓取功能（拓展功能）

通过分区域识别策略解决速度慢的问题。识别区域划分为 9 个子块，如下图所示：



识别区域

分区识别策略的步骤如下：

- 设置区域索引号  $i=0,1,2,3,4,5,6,7,8$ 。
- 设置识别第  $i$  区域的像素坐标（请参考下表区域对应值）
  - 机器人发：SetParam,204,0,0 //设置区域的左上角顶点像素坐标, 此处设置第一个区域的左上角顶点像素坐标
  - 机器人发：SetParam,205,426,341 //设置区域的右下角顶点像素坐标, 此处设置第一个区域的右下角顶点像素坐标
- 区域  $i$  识别抓取
  - 机器人发：  
RecgGrasp,X+238.548,Y-554.296,Z+10.588,RX-159.791,RY-50.461,RZ-92.328,M0
  - 机器人收：  
成功：  
X+371.758,Y-338.203,Z+46.015,RX+160.280,RY-10.317,RZ-18.695,M0,T0,PICK1,N2  
失败：  
NO\_RecgGrasp, CsvMaster 识别模型失败

- 如果识别只有一个工件 (N1) , 那么  $i=i+1$ ;

识别区域设置表

第 1 个区域:	SetParam,204,0,0;SetParam,205,426,341;
第 2 个区域:	SetParam,204,425,0;SetParam,205,853,341;
第 3 个区域:	SetParam,204,852,0;SetParam,205,1280,341;
第 4 个区域:	SetParam,204,0,340;SetParam,205,426,682;
第 5 个区域:	SetParam,204,425,340;SetParam,205,853,682;
第 6 个区域:	SetParam,204,852,340;SetParam,205,1280,682;
第 7 个区域:	SetParam,204,0,681;SetParam,205,426,1024;
第 8 个区域:	SetParam,204,425,681;SetParam,205,853,1024;
第 9 个区域:	SetParam,204,852,681;SetParam,205,1280,1024;

## 6. 轨迹引导功能

### 6.1 轨迹引导配置方法

轨迹引导前, 需要在文件./cfg/trackingPoints.csv 配置跟踪点。配置参数的格式如下

表:

轨迹点参数

序号	参数名	含义
1	X	工件坐标系下的轨迹点坐标 X 值
2	Y	工件坐标系下的轨迹点坐标 Y 值
3	Z	工件坐标系下的轨迹点坐标 Z 值

4	coef	工具末端倾斜系数，取值范围[0-1]，0 表示工具末端处于水平，1 表示工具末端处于垂直
5	vectorY0	Y 法向量 X 轴坐标值
6	vectorY1	Y 法向量 Y 轴坐标值
7	vectorY2	Y 法向量 Z 轴坐标值
8	Model ID	模型 ID

## 6.2 机器人执行轨迹轨迹

- 机器人发：

Track,ICP0 或者 Track,ICP1

- 机器人收：

成功：

X±坐标值,Y±坐标值,Z±坐标值,RX±坐标值,RY±坐标值,RZ±坐标值,...,X±坐标

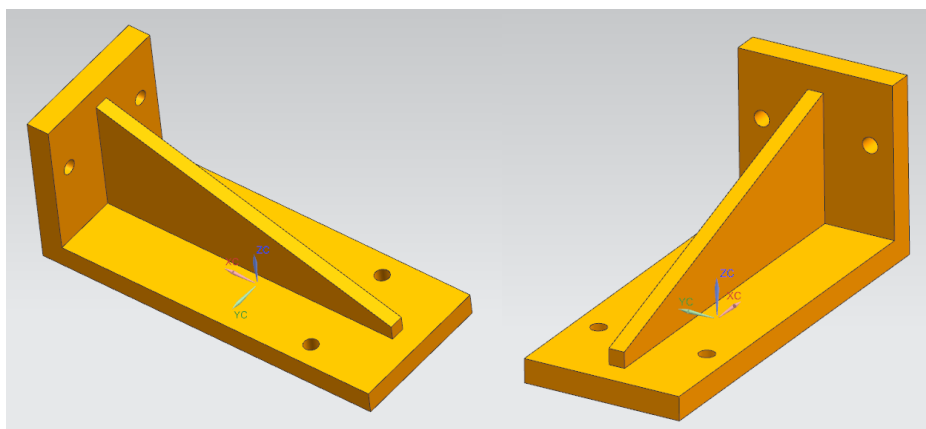
值,Y±坐标值,Z±坐标值,RX±坐标值,RY±坐标值,RZ±坐标值

失败：

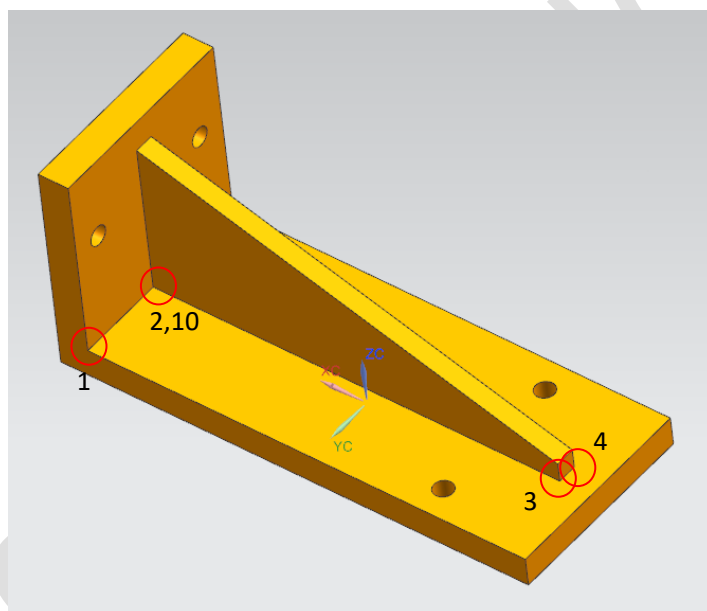
NO\_Track, CsvMaster 轨迹生成失败

## 6.3 例子：三角凸块工件轨迹引导

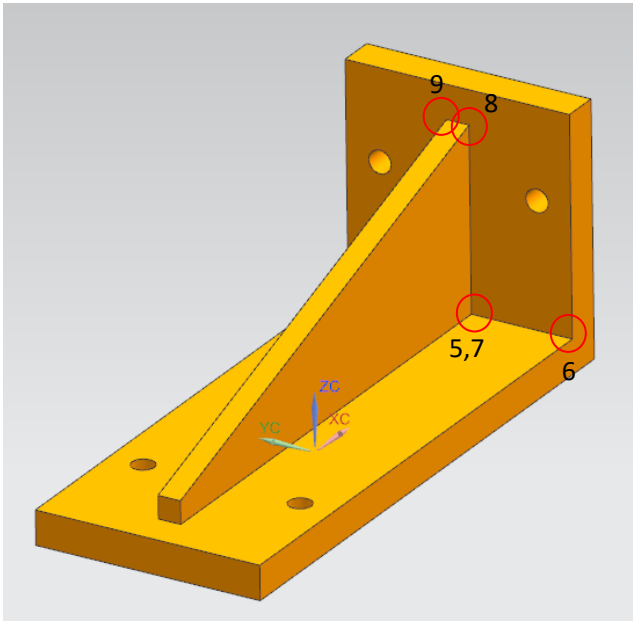
### A.三角凸台工件



## B. 轨迹示意







如上图所示，轨迹引导点共包含 10 个，引导顺序由 1,2, ..., 10 行进，工具末端朝外。

C.配置表

10 个引导点位的配置表，如下所示：

ID	X	Y	Z	coef	vectorY0	vectorY1	vectorY2	Model ID
1	103	50	20	0.7	-1	1	0	M0
2	103	10	20	0.7	-1	1	0	M0
3	-113	10	20	0.7	0	0	0	M0
4	-113	-10	20	0.7	-1	-1	0	M0
5	103	-10	20	0.7	-1	-1	0	M0
6	103	-50	20	0.7	-1	-1	0	M0
7	103	-10	20	0.7	-1	-1	0	M0
8	103	-10	105	0.7	-1	-1	0	M0
9	103	10	105	0.7	-1	1	0	M0
10	103	10	20	0.7	-1	1	0	M0

对应于配置文件的设置如下图所示：

// TrackingPoints							
1.0	//version ID						
X	Y	Z	coef	vectorY0	vectorY1	vectorY2	Model ID
103	50	20	0.7	-1	1	0	M0
103	10	20	0.7	-1	1	0	M0
-113	10	20	0.7	0	0	0	M0
-113	-10	20	0.7	-1	-1	0	M0
103	-10	20	0.7	-1	-1	0	M0
103	-50	20	0.7	-1	-1	0	M0
103	-10	20	0.7	-1	-1	0	M0
103	-10	105	0.7	-1	-1	0	M0
103	10	105	0.7	-1	1	0	M0
103	10	20	0.7	-1	1	0	M0

#### D.机器人执行操作

- 机器人发：

Track,ICP0 或者 Track,ICP1

- 机器人收：

成功：

X+371.758,Y-338.203,Z+46.015,RX+160.280,RY-10.317,RZ-18.695,...,X+371.758,Y-338.203,Z+46.015,RX+160.280,RY-10.317,RZ-18.695

(说明：ICP0 模式返回 10 个轨迹点，ICP1 模式返回 18 个轨迹点位)

失败：

NO\_Track, CsvMaster 轨迹生成失败