

ABB[a]-J-5ABB 机器人的程序数据

5.1 任务目标

掌握程序数据的建立方法。

掌握三个关键程序数据的设定。

了解机器人工具自动识别功能。

5.2 任务描述

以 bool 为例，建立程序数据，练习建立 num、robtargt 程序数据。

设定机器人的工具数据 tooldata、工件坐标 wobjdata、负荷数据 loaddata。

使用 LoadIdentify 工具自动识别安装在六轴法兰盘上的工具 (tooldata) 和载荷 (loaddata) 的重量，以及重心。

5.3 知识储备

5.3.1 程序数据

程序数据是在程序模块或系统模块中设定的值和定义的一些环境数据。创建的程序数据由同一个模块或其他模块中的指令进行引用。图中是一条常用的机器人关节运动的指令 MoveJ，调用了四个程序数据。

```

1
2 MODULE Module1
3   CONST robtarget p10 := [[374.00, 0.00, 630.00], [
4   PROC Routine1()
5     MoveJ p10, v1000, z50, tool0;
6   ENDPROC
7
8 ENDMODULE
    
```

图中所使用的程序数据的说明见表：

程序数据	数据类型	说明
p10	robtargt	机器人运动目标位置数据
v1000	speeddata	机器人运动速度数据
z50	zonedata	机器人运动转弯数据
tool0	tooldata	机器人工作数据 TCP

5.3.2 程序数据的类型与分类

1. 程序数据的类型分类

ABB 机器人的程序数据共有 76 个，并且可以根据实际情况进行程序数据的创建，为 ABB 机器人的程序设计带来了无限可能性。

在示教器的“程序数据”窗口可查看和创建所需要的程序数据。



2. 程序数据的存储类型

(1) 变量 VAR

变量型数据在程序执行的过程中和停止时，会保持当前的值。但如果程序指针被移到主程序后，数值会丢失。

举例说明：

VAR num length:=0; 名称为 length 的数字数据

VAR string name:= "John ";名称为 name 的字符数据

VAR bool finish:=FALSE名称为 finish 的布尔量数据

在程序编辑窗口中的显示如图：

```
MODULE Module1
  VAR num length:=0;
  CONST string name:="John";
  VAR bool finished:=FALSE;

ENDMODULE
```

在机器人执行的 RAPID 程序中也可以对变量存储类型程序数据进行赋值的操作，如图：

```
MODULE Module1
  VAR num length:=0;
  VAR string name:="John";
  VAR bool finished:=FALSE;
  PROC main()
    length := 10 - 1;
    name := "john";
    finished := TRUE;
  ENDPROC
ENDMODULE
```

*注意：VAR表示存储类型为变量

num 表示程序数据类型

*提示：在定义数据时，可以定义变量数据的初始值。如 length 的初始值为 0，name 的初始值为 John，finish 的初始值为 FALSE

*注意：在程序中执行变量型数据的赋值，在指针复位后将恢复为初始值。

(2) 可变量 PERS可变量最大的特点是，无论程序的指针如何，都会保持最后赋予的值。 举例说明：

PERS numnbr:=1; 名称为 nbr 的数字数据

PERS stringtest:= 'Hello ';名称为 test 的字符数据

在机器人执行的 RAPID程序中也可以对可变量存储类型程序数据进行赋值的操作。

在程序执行以后，赋值的结果会一直保持，直到对其进行重新赋值。

*注意：PERS表示存储类型为可变量

(3) 常量 CONST常量的特点是在定义时已赋予了数值，并不能在程序中进行修改，除非手动修改。 举例说明：

CONST numgravity:=9.81; 名称为 gravity 的数字数据

CONST stringgreeting:= " Hello ";名称为 greating 的字符数据

*注意：存储类型为常量的程序数据，不允许在程序中进行赋值的操作。

三种数据的存储类型在编辑界面的显示如下：

```

MODULE Module1
  VAR num length:=0;
  VAR string name:="John";
  VAR bool finished:=FALSE;
  PERS string text:="Hello";
  PERS num nbr:=1;
  CONST num gravity:=9.81;
  CONST string greating:="Hello";
  PROC main()
    length := 10 - 1;
  
```

3.常用的程序数据

根据不同的数据用途，定义了不同的程序数据，下表是机器人系统中常用的程序数据：

程序数据	说明
bool	布尔量
byte	整数数据 0~255
clock	计时数据
dionum	数字输入 /输出信号
extjoint	外轴位置数据
intnum	中断标志符
jointtarget	关节位置数据
loaddata	负荷数据
mecunit	机械装置数据
num	数值数据
orient	姿态数据
pos	位置数据（只有 X、Y 和 Z）
pose	坐标转换
robjoint	机器人轴角度数据
robtargt	机器人与外轴的位置数据
speeddata	机器人与外轴的速度数据
string	字符串
tooldata	工具数据
trapdata	中断数据
wobjdata	工件数据
zonedata	TCP转弯半径数据

*提示：系统中还有针对一些特殊功能的程序数据，在对应的功能说明书中会有相应的详细介绍，请查看随机光盘电子版说明书。也可以根据需要新建程序数据类型。

5.4 任务实施

5.4.1 建立程序数据

程序数据的建立一般可以分为两种形式，一种是直接在示教器中的程序数据画面中建立程序数据；另

一种是在建立程序指令时，同时自动生成对应的程序数据。

本节将介绍直接在示教器的程序数据画面中建立程序数据的方法。下面以建立布尔数据为例子进行说明，练习时建立 num 和 rotarget 程序数据。

建立 bool 数据的操作步骤：

 The screenshot shows the ABB main menu. The '程序数据' (Program Data) option is highlighted in blue. Other options include HotEdit, 输入输出, 手动操纵, 自动生产窗口, 程序编辑器, 备份与恢复, 校准, 控制面板, 事件日志, FlexPendant 资源管理器, 系统信息, 注销 Default User, and 重新启动.	<p>1. ABB 菜单中，选择 “程序数据”。</p>
 The screenshot shows the '程序数据 - 已用数据类型' (Program Data - Used Data Types) window. A list of data types is displayed: bool, num, wobjdata, clock, string, loaddata, and tooldata. The 'bool' type is selected and highlighted in blue. At the bottom, there are buttons for '显示数据' (Show Data) and '视图' (View).	<p>2. 选择数据类型 “bool,”单击“显示数据”。</p>
 The screenshot shows the toolbar for the Program Data window. It contains buttons for '新建...' (New...), '编辑' (Edit), '刷新' (Refresh), and '查看数据类型' (View Data Types). Below the toolbar, there is a '程序数据' (Program Data) button and a 'NOB_1' button.	<p>3. 单击 “新建 ...”。</p>



4. 进行名称的设定、单击下拉菜单选择对应的参数，设定完成后单击“确定”完成设定。

数据设定参数及说明见表：

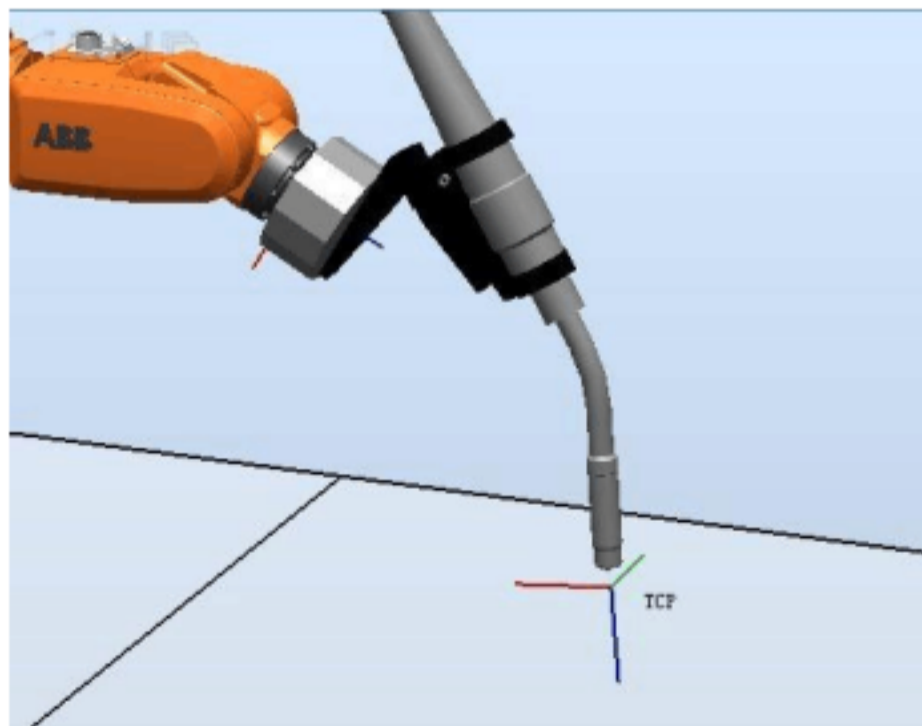
设定参数	说明
名称	设定数据的名称
范围	设定数据可使用的范围
存储类型	设定数据的可存储类型
任务	设定数据所在的任务
模块	设定数据所在的模块
例行程序	设定数据所在的例行程序
维数	设定数据的维数
初始值	设定数据的初始值

5.4.2 三个关键的程序数据的设定

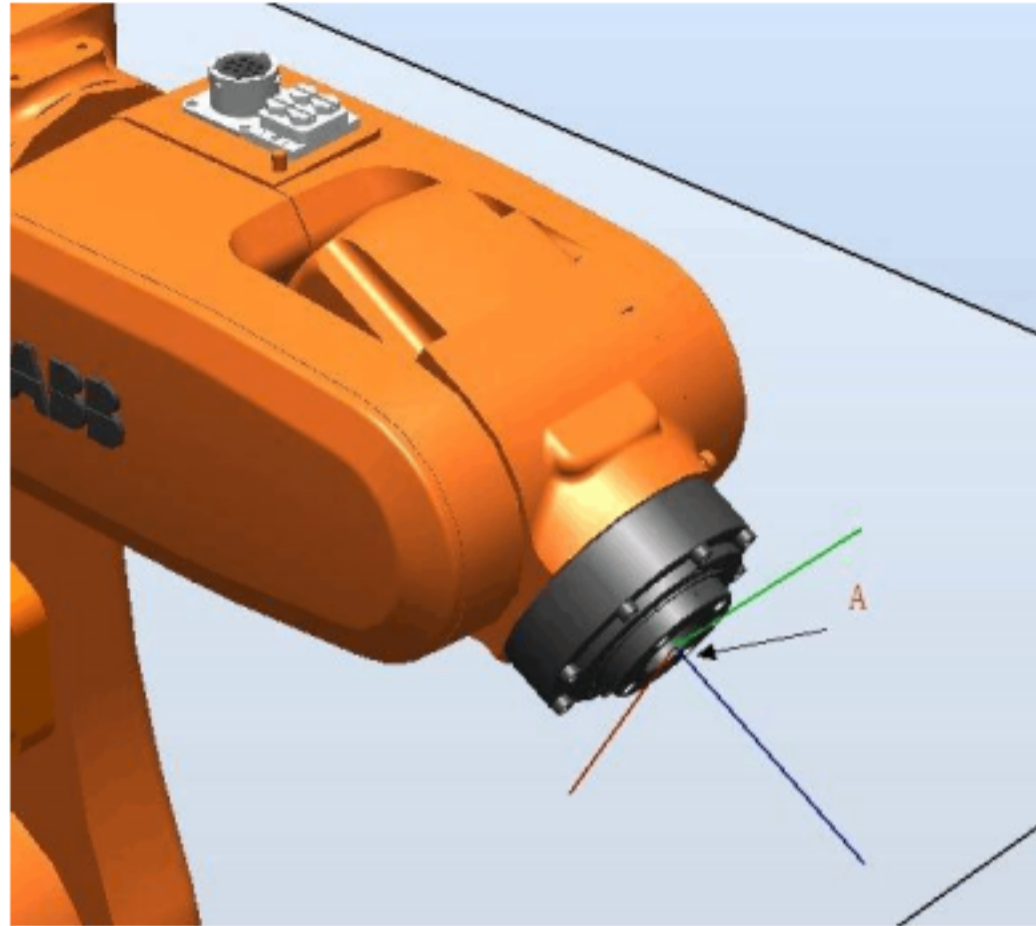
在进行正式的编程之前，就需要构建起必要的编程环境，其中三个必须的程序数据（工具数据 tooldata、工件坐标 wobjdata、负荷数据 loaddata）就需要在编程前进行定义。

1.工具数据 tooldata

工具数据 tooldata 用于描述安装在机器人第六轴上的工具的 TCR、质量、重心等参数数据。一般不同的机器人应用配置不同的工具，比如说弧焊的机器人就使用弧焊枪作为工具，而用于搬运板材的机器人就会使用吸盘式的夹具作为工具。



默认工具 (tool0) 的工具中心点 (Tool Center Point) 位于机器人安装法兰盘的中心。图中 A 点就是原始的 TCP 点。



TCP 的设置原理如下：

- 1) 首先在机器人工作范围内找一个非常精确的固定点作为参考点。
- 2) 然后在工具上确定一个参考点 (最好是工具的中心点) 。
- 3) 用之前介绍的手动操纵机器人的方法，去移动工具上的参考点，以四种以上不同的机器人姿态尽可能与固定点刚好碰上。为了获得更准确的 TCP，在以下例子中使用六点法进行操作，第四点是用工具的参考点垂直于固定点，第五点是工具参考点从固定点向将要设定为 TCP 的 X 方向移动，第六点是工具参考点从固定点向将要设定为 TCP 的 Z 方向移动。
- 4) 机器人通过这四个位置点的位置数据计算求得 TCP 的数据，然后 TCP 的数据就保存在 tooldata 这个程序数据中被程序进行调用。

* 提示：执行程序时，机器人将 TCP 移至编程位置。这意味着，如果要更改工具以及工具坐标系，机器人的移动将随之更改，以便新的 TCP 到达目标。

所有机器人在手腕处都有一个预定义工具坐标系，该坐标系被称为 tool0。这样就能将一个或多个新工具坐标系定义为 tool0 的偏移值。

* 注意：TCP 取点数量的区别：

4 点法，不改变 tool0 的坐标方向

5 点法，改变 tool0 的 Z 方向

6 点法，改变 tool0 的 X 和 Z 方向 (在焊接应用最为常用) 。前

三个点的姿态相差尽量大些，这样有利于 TCP 精度的提高。

操作步骤：

	<p>1. ABB 菜单中，选择 ‘手动操纵’。</p>
	<p>2. 选择 ‘工具坐标’。</p>
	<p>3. 单击 ‘新建’。</p>



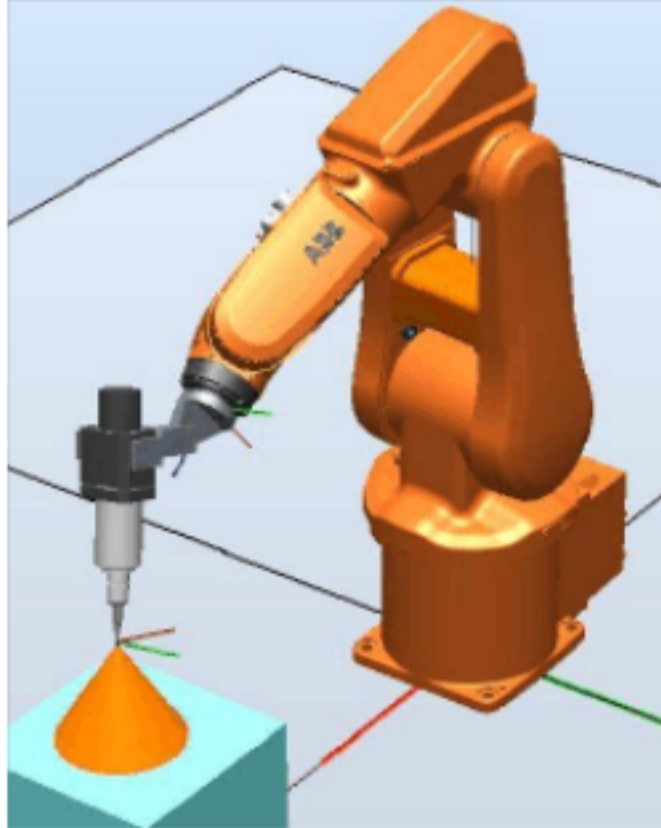
4. 对工具数据属性进行设定后，单击“确定”。



5. 选中 tool1 后，单击“编辑”菜单中的“定义”选项。



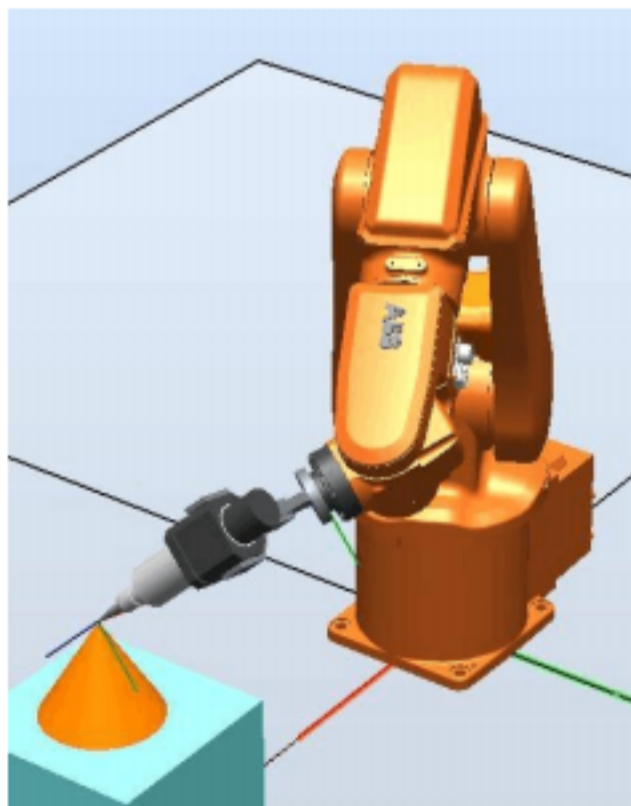
6. 选择“TCP和 Z, X”，使用 6 点法设定 TCP。



7. 选择合适的手动操纵模式。
8. 按下使能键，使用摇杆使工具参考点靠上固定点，作为第一个点。

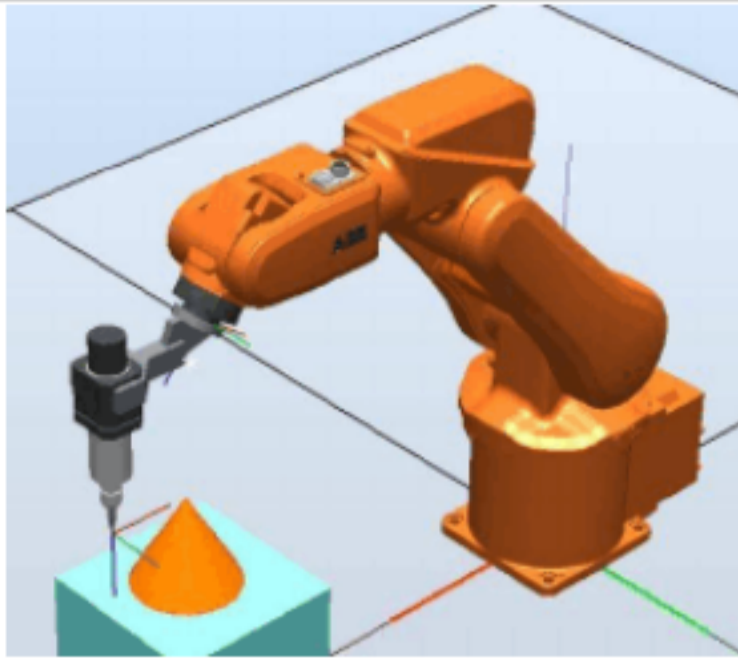


9. 单击“修改位置”，将点 1 位置记录下来。



10. 工具参考点变换姿态靠上固定点。

<table border="1"> <thead> <tr> <th>点</th> <th>状态</th> </tr> </thead> <tbody> <tr> <td>点 1</td> <td>已修改</td> </tr> <tr> <td>点 2</td> <td>已修改</td> </tr> <tr> <td>点 3</td> <td>-</td> </tr> <tr> <td>点 4</td> <td>-</td> </tr> </tbody> </table>	点	状态	点 1	已修改	点 2	已修改	点 3	-	点 4	-	<p>11. 单击“修改位置”，将点 2 位置记录下来。</p>
点	状态										
点 1	已修改										
点 2	已修改										
点 3	-										
点 4	-										
	<p>12. 工具参考点变换姿态靠上固定点。</p>										
<table border="1"> <thead> <tr> <th>点</th> <th>状态</th> </tr> </thead> <tbody> <tr> <td>点 1</td> <td>已修改</td> </tr> <tr> <td>点 2</td> <td>已修改</td> </tr> <tr> <td>点 3</td> <td>已修改</td> </tr> <tr> <td>点 4</td> <td>-</td> </tr> </tbody> </table>	点	状态	点 1	已修改	点 2	已修改	点 3	已修改	点 4	-	<p>13. 单击“修改位置”，将点 3 位置记录下来。</p>
点	状态										
点 1	已修改										
点 2	已修改										
点 3	已修改										
点 4	-										
	<p>14. 工具参考点变换姿态靠上固定点。这是第 4 个点，工具参考点垂直于固定点。</p>										
<table border="1"> <thead> <tr> <th>点</th> <th>状态</th> </tr> </thead> <tbody> <tr> <td>点 1</td> <td>已修改</td> </tr> <tr> <td>点 2</td> <td>已修改</td> </tr> <tr> <td>点 3</td> <td>已修改</td> </tr> <tr> <td>点 4</td> <td>已修改</td> </tr> </tbody> </table> <p>位置 ▲ 修改位置 确定 取消</p>	点	状态	点 1	已修改	点 2	已修改	点 3	已修改	点 4	已修改	<p>15. 单击“修改位置”，将点 4 位置记录下来。</p>
点	状态										
点 1	已修改										
点 2	已修改										
点 3	已修改										
点 4	已修改										



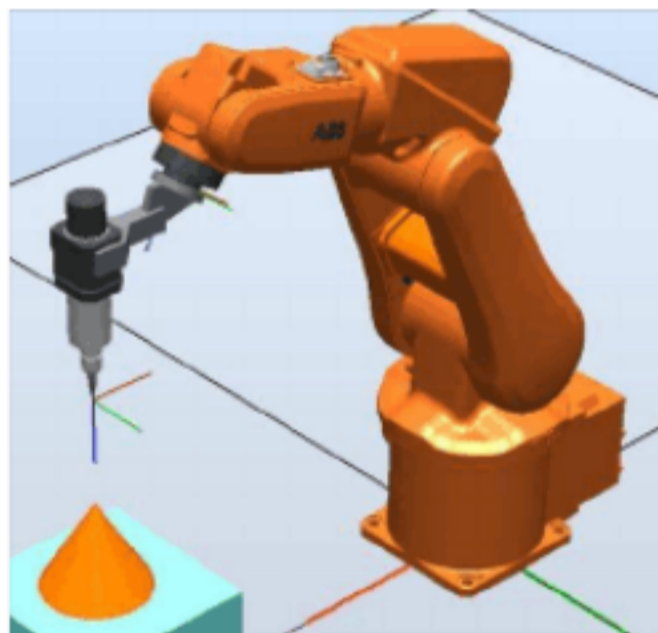
16. 工具参考点以点 4 的姿态从固定点移动到工具 TCP 的+X方向。

点	状态
点 3	已修改
点 4	已修改
延伸器点 X	已修改
延伸器点 Z	-

3 到 6 共 6

位置 ▲ 修改位置 确定 取消

17. 单击“修改位置”，将延伸器点 X 位置记录下来。



18. 工具参考点以此姿态从固定点移动到工具 TCP 的 Z 方向。

点	状态
点 3	已修改
点 4	已修改
延伸器点 X	已修改
延伸器点 Z	已修改

3 到 6 共 6

位置 ▲ 修改位置 确定 取消

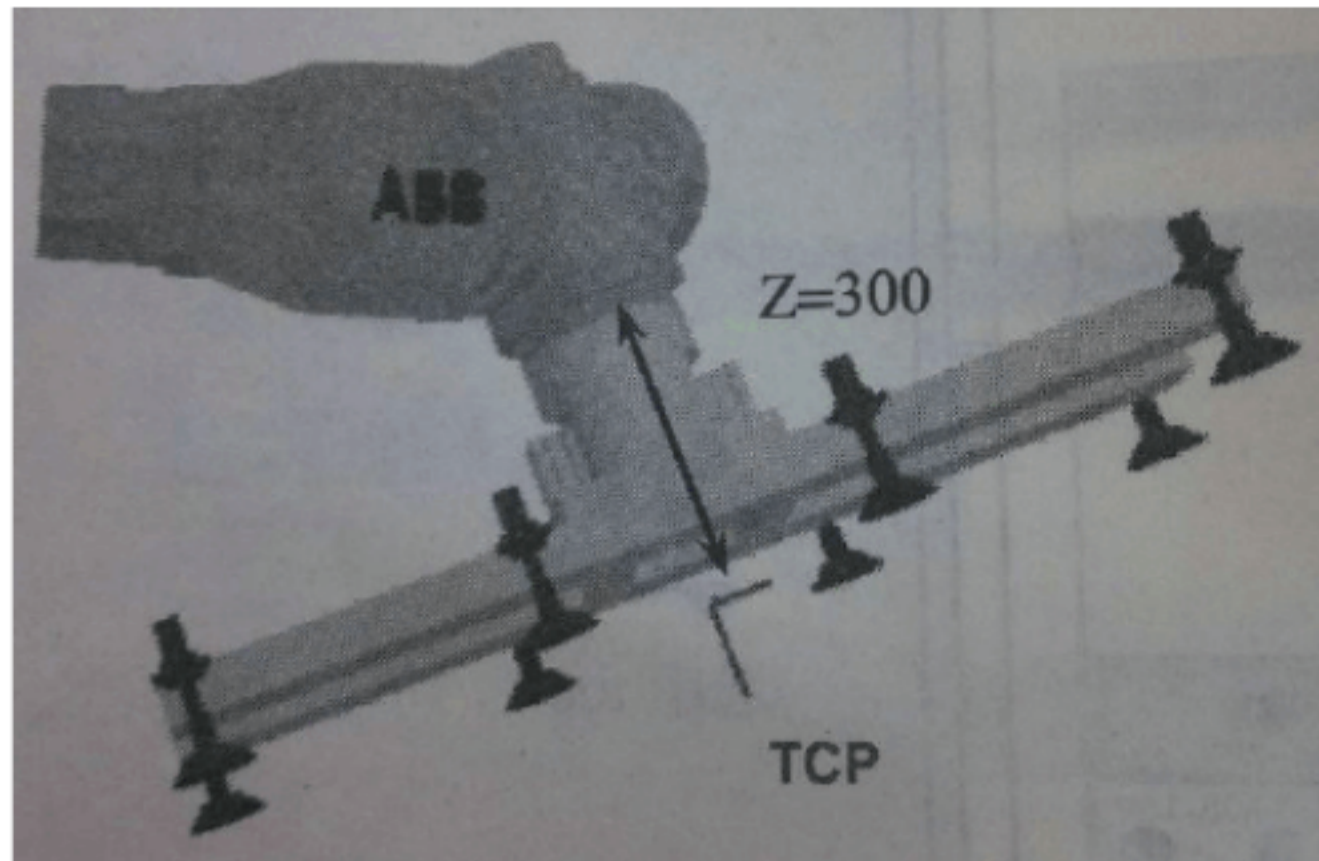
19. 单击“修改位置”，将延伸器点 Z 位置记录下来。
20. 单击“确定”完成设定。

<p>程序数据 - tooldata - 定义 - 工具坐标定义</p> <p>计算结果 工具坐标: tool1</p> <p>点击“确定”确认结果，或点击“取消”重新定义数据。</p> <table border="1"> <tr><td>方法</td><td>ToolXZ</td></tr> <tr><td>最大误差</td><td>0.000220084 毫米</td></tr> <tr><td>最小误差</td><td>3.182889E-05 毫米</td></tr> <tr><td>平均误差</td><td>0.0001078244 毫米</td></tr> <tr><td>X:</td><td>31.7924 毫米</td></tr> <tr><td>Y:</td><td>-0.0001775473 毫米</td></tr> </table> <p style="text-align: right;">确定 取消</p>	方法	ToolXZ	最大误差	0.000220084 毫米	最小误差	3.182889E-05 毫米	平均误差	0.0001078244 毫米	X:	31.7924 毫米	Y:	-0.0001775473 毫米	<p>21. 对误差进行确认，越小越好，但也要以实际验证效果为准。</p>									
方法	ToolXZ																					
最大误差	0.000220084 毫米																					
最小误差	3.182889E-05 毫米																					
平均误差	0.0001078244 毫米																					
X:	31.7924 毫米																					
Y:	-0.0001775473 毫米																					
<p>手动操纵 - 工具</p> <p>当前选择: tool1</p> <p>从列表中选择一项。</p> <table border="1"> <thead> <tr> <th>工具名称</th> <th>模块</th> <th>范围</th> </tr> </thead> <tbody> <tr> <td>AW_Gun</td> <td>RAPID/T_ROB1/CalibData</td> <td>全局</td> </tr> <tr> <td>tool0</td> <td>RAPID/T_ROB1/BASE</td> <td>全局</td> </tr> <tr style="background-color: #0000FF; color: white;"> <td>tool1</td> <td>/Module1</td> <td>任务</td> </tr> </tbody> </table> <p style="text-align: right;">新建... 编辑 确定 取消</p>	工具名称	模块	范围	AW_Gun	RAPID/T_ROB1/CalibData	全局	tool0	RAPID/T_ROB1/BASE	全局	tool1	/Module1	任务	<p>22. 选中 tool1，然后打开编辑菜单选择“更改值”。</p>									
工具名称	模块	范围																				
AW_Gun	RAPID/T_ROB1/CalibData	全局																				
tool0	RAPID/T_ROB1/BASE	全局																				
tool1	/Module1	任务																				
<p>编辑</p> <p>名称: tool1</p> <p>点击一个字段以编辑值。</p> <table border="1"> <thead> <tr> <th>名称</th> <th>值</th> <th>数据类型</th> </tr> </thead> <tbody> <tr> <td>tload:</td> <td>[1, [0, 0, 1], [1, 0, 0, 0], ...]</td> <td>loaddata</td> </tr> <tr style="background-color: #0000FF; color: white;"> <td>mass :=</td> <td>1</td> <td>NUM</td> </tr> <tr> <td>cog:</td> <td>[0, 0, 1]</td> <td>POS</td> </tr> <tr> <td>x :=</td> <td>0</td> <td>NUM</td> </tr> <tr> <td>y :=</td> <td>0</td> <td>NUM</td> </tr> <tr> <td>z :=</td> <td>1</td> <td>NUM</td> </tr> </tbody> </table> <p style="text-align: right;">撤消 确定 取消</p>	名称	值	数据类型	tload:	[1, [0, 0, 1], [1, 0, 0, 0], ...]	loaddata	mass :=	1	NUM	cog:	[0, 0, 1]	POS	x :=	0	NUM	y :=	0	NUM	z :=	1	NUM	<p>23. 在此页面中，根据实际情况设定工具的质量 mass（单位 kg）和重心位置数据（此中心是基于 tool0 的偏移值，单位 mm），然后单击“确定”。</p> <p>*提示：此页显示的内容就是 TCP 定义时生成的数据。</p>
名称	值	数据类型																				
tload:	[1, [0, 0, 1], [1, 0, 0, 0], ...]	loaddata																				
mass :=	1	NUM																				
cog:	[0, 0, 1]	POS																				
x :=	0	NUM																				
y :=	0	NUM																				
z :=	1	NUM																				

 <p>手动操纵 - 工具</p> <p>当前选择: tool1</p> <p>从列表中选择一个项目。</p> <table border="1"> <thead> <tr> <th>工具名称</th> <th>模块</th> <th>范围</th> </tr> </thead> <tbody> <tr> <td>AW_Gun</td> <td>RAPID/T_ROB1/CalibData</td> <td>全局</td> </tr> <tr> <td>tool0</td> <td>RAPID/T_ROB1/BASE</td> <td>全局</td> </tr> <tr> <td>tool1</td> <td>RAPID/T_ROB1/Module1</td> <td>任务</td> </tr> </tbody> </table> <p>新建... 编辑 确定 取消</p>	工具名称	模块	范围	AW_Gun	RAPID/T_ROB1/CalibData	全局	tool0	RAPID/T_ROB1/BASE	全局	tool1	RAPID/T_ROB1/Module1	任务	<p>24. 选中 tool1 ，单击“确定”。</p>
工具名称	模块	范围											
AW_Gun	RAPID/T_ROB1/CalibData	全局											
tool0	RAPID/T_ROB1/BASE	全局											
tool1	RAPID/T_ROB1/Module1	任务											
 <p>手动操纵</p> <p>点击属性并更改</p> <p>机械单元: ROB_1...</p> <p>绝对精度: Off</p> <p>动作模式: 重定位...</p> <p>坐标系: 工具...</p> <p>工具坐标: tool1...</p> <p>工件坐标: wobj0...</p> <p>有效载荷: load1...</p> <p>操纵杆锁定: 无...</p> <p>增量: 无...</p>	<p>25. 动作模式选定为“重定位”。坐标系统选定为“工具”。工具坐标选定为“tool1”。</p>												
	<p>26. 使用摇杆将工具参考点靠上固定点，然后在重定位模式下手动操纵机器人，如果 TCP 设定精确的话，可以看到工具参考点与固定点始终保持接触，而机器人会根据重定位操作改变姿态。</p>												

如果使用搬运的夹具，一般工具数据的设定方法如下：

图中，搬运薄板的真空吸盘夹具为例，质量是 25kg，重心在默认 tool0 的 Z 的正方向偏移 250mm，TCP 点设定在吸盘的接触面上，从默认 tool0 上的 Z 方向偏移了 300mm。



在示教器上设定如下：

手动操纵

点击属性并更改

机械单元: ROB_1...

绝对精度: Off

动作模式: 重定位...

坐标系: 工具...

工具坐标: tool1...

工件坐标: wobj0...

有效载荷: load1...

操纵杆锁定: 无...

增量: 无...

位置
坐标中的位置: WorkObject
X: 548.4 mm
Y: 0.2 mm
Z: 300.4 mm
q1: 0.28779
q2: 0.53079
q3: 0.12222
q4: -0.78772

位置格式...

操纵杆方向
X Y Z

对准... 转到... 启动...

1. 在“手动操纵”界面，选择“工具坐标”。

手动操纵 - 工具

当前选择: tool1

从列表中选择一个项目。

工具名称	模块	范围
AW_Gun	RAPID/T_ROB1/CalibData	全局
tool0	RAPID/T_ROB1/BASE	全局
tool1	RAPID/T_ROB1/Module1	任务

新建... 编辑 确定 取消

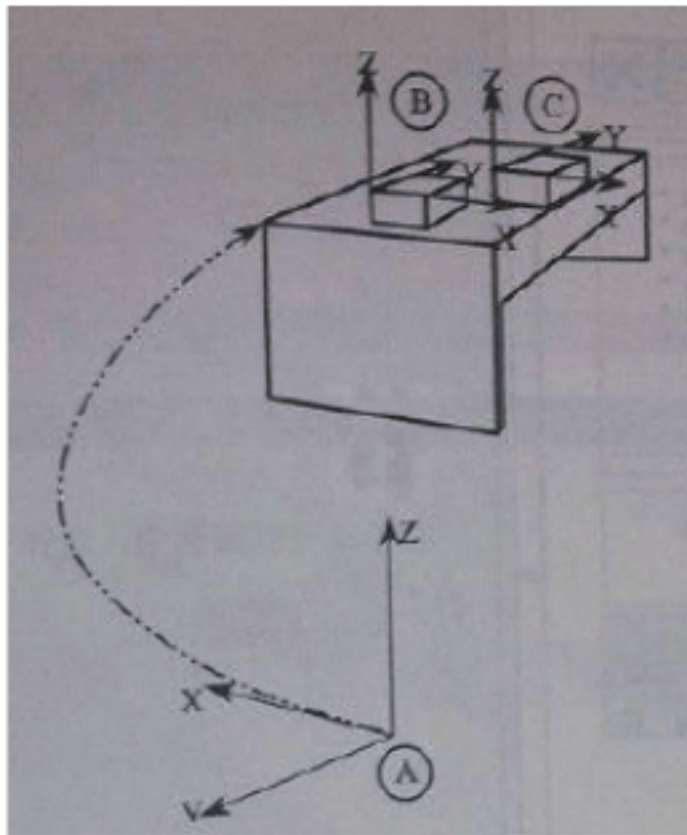
2. 单击“新建”。

<div style="border: 1px solid gray; padding: 5px;"> <p>新数据声明</p> <p>数据类型: tooldata 当前任务: T_ROB1</p> <p>名称: <input type="text" value="tool2"/> ...</p> <p>范围: <input type="text" value="任务"/> ▼</p> <p>存储类型: <input type="text" value="可变量"/> ▼</p> <p>任务: <input type="text" value="T_ROB1"/> ▼</p> <p>模块: <input type="text" value="Module1"/> ▼</p> <p>例行程序: <input type="text" value="<无>"/> ▼</p> <p>维数: <input type="text" value="<无>"/> ▼ <input type="text" value=""/> ...</p> <p style="text-align: center;"> <input type="button" value="初始值"/> <input type="button" value="确定"/> <input type="button" value="取消"/> </p> </div>	<p>3. 根据需要设定数据的属性，一般不用修改。</p> <p>4. 单击“初始值”。</p>																												
<div style="border: 1px solid gray; padding: 5px;"> <p>名称: tool2</p> <p>点击一个字段以编辑值。</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 30%;">名称</th> <th style="width: 30%;">值</th> <th style="width: 20%;">数据类型</th> <th style="width: 20%;"></th> </tr> </thead> <tbody> <tr> <td>robhold :=</td> <td>TRUE</td> <td>bool</td> <td></td> </tr> <tr> <td>tframe:</td> <td>[[0, 0, 300], [1, 0, 0, 0]]</td> <td>pose</td> <td></td> </tr> <tr> <td>trans:</td> <td>[0, 0, 300]</td> <td>pos</td> <td></td> </tr> <tr> <td>x :=</td> <td>0</td> <td>num</td> <td></td> </tr> <tr> <td>y :=</td> <td>0</td> <td>num</td> <td></td> </tr> <tr style="background-color: #0000FF; color: white;"> <td>z :=</td> <td>300</td> <td>num</td> <td></td> </tr> </tbody> </table> <p style="text-align: center;"> <input type="button" value="确定"/> <input type="button" value="取消"/> </p> </div>	名称	值	数据类型		robhold :=	TRUE	bool		tframe:	[[0, 0, 300], [1, 0, 0, 0]]	pose		trans:	[0, 0, 300]	pos		x :=	0	num		y :=	0	num		z :=	300	num		<p>5. TCP点设定在吸盘的接触面上，从默认 tool0 上的 Z 正方向偏移了 300mm，在此画面中设定对应的数值。</p>
名称	值	数据类型																											
robhold :=	TRUE	bool																											
tframe:	[[0, 0, 300], [1, 0, 0, 0]]	pose																											
trans:	[0, 0, 300]	pos																											
x :=	0	num																											
y :=	0	num																											
z :=	300	num																											
<div style="border: 1px solid gray; padding: 5px;"> <p>编辑</p> <p>名称: tool2</p> <p>点击一个字段以编辑值。</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 30%;">名称</th> <th style="width: 30%;">值</th> <th style="width: 20%;">数据类型</th> <th style="width: 20%;"></th> </tr> </thead> <tbody> <tr> <td>mass :=</td> <td>25</td> <td>num</td> <td></td> </tr> <tr> <td>cog:</td> <td>[0, 0, 250]</td> <td>pos</td> <td></td> </tr> <tr> <td>x :=</td> <td>0</td> <td>num</td> <td></td> </tr> <tr> <td>y :=</td> <td>0</td> <td>num</td> <td></td> </tr> <tr style="background-color: #0000FF; color: white;"> <td>z :=</td> <td>250</td> <td>num</td> <td></td> </tr> <tr> <td>com:</td> <td>[1, 0, 0, 0]</td> <td>orient</td> <td></td> </tr> </tbody> </table> <p style="text-align: center;"> <input type="button" value="确定"/> <input type="button" value="取消"/> </p> </div>	名称	值	数据类型		mass :=	25	num		cog:	[0, 0, 250]	pos		x :=	0	num		y :=	0	num		z :=	250	num		com:	[1, 0, 0, 0]	orient		<p>6. 此工具质量是 25kg，重心在默认 tool0 的 Z 的正方向偏移 250mm，在画面中设定对应的数值，然后单击“确定”，设定完成。</p>
名称	值	数据类型																											
mass :=	25	num																											
cog:	[0, 0, 250]	pos																											
x :=	0	num																											
y :=	0	num																											
z :=	250	num																											
com:	[1, 0, 0, 0]	orient																											

2.工件坐标 wobdata

工件坐标对应工件，它定义工件相对于大地坐标（或其他坐标）的位置。机器人可以拥有若干工件坐

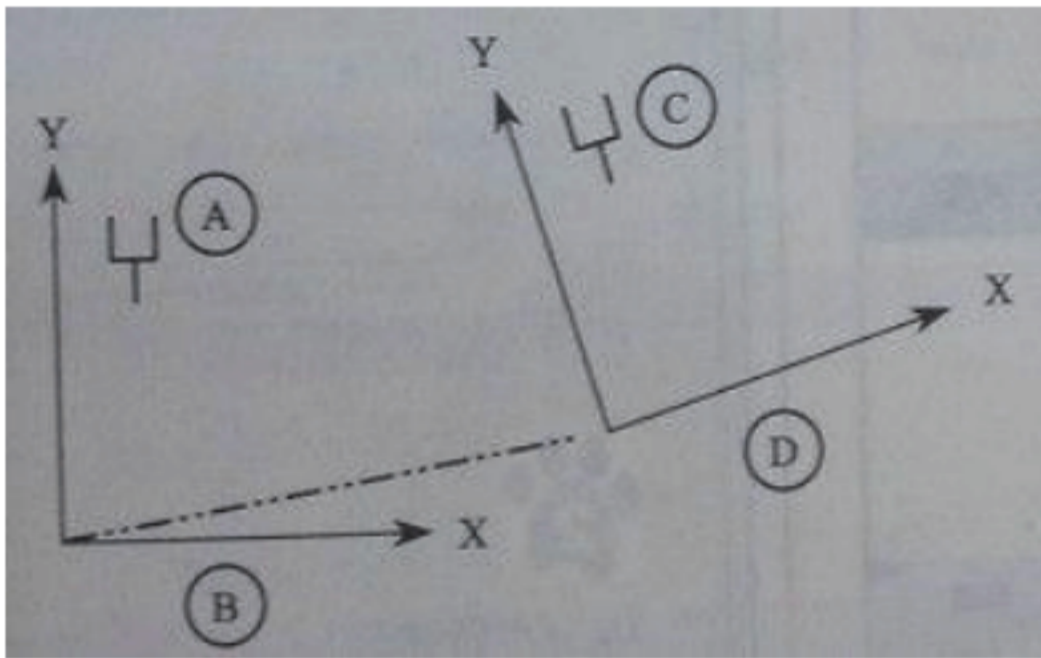
标系，或者表示不同工件，或者表示同一工件在不同位置的若干副本。



对机器人进行编程时就是在工件坐标中创建目标和路径。这带来很多优点：

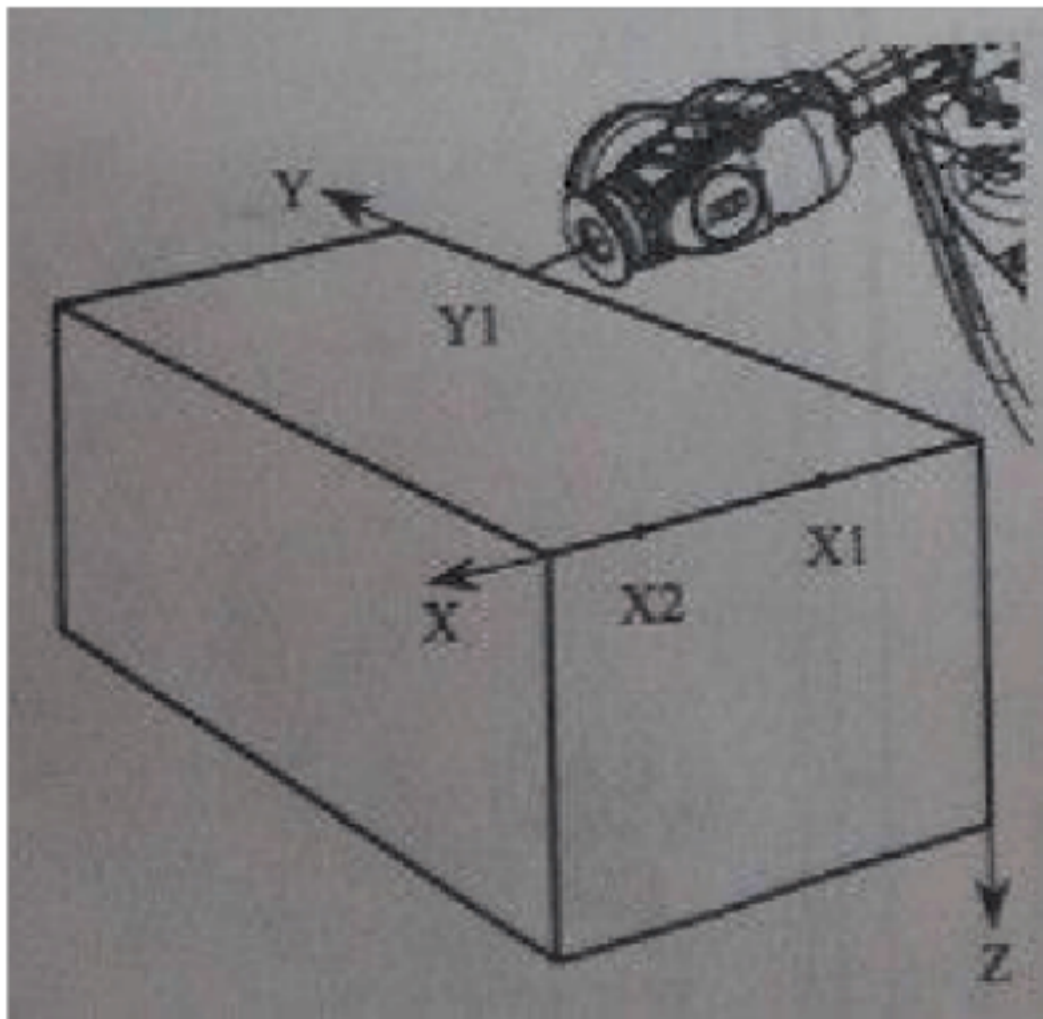
- 1) 重新定位工作站中的工件时，只需要更改工件坐标的位置，所有路径将即刻随之更新。
- 2) 允许操作以外轴或传送导轨移动的工件，因为整个工件可连同其路径一起移动。

*提示：A 是机器人的大地坐标，为了方便编程，给第一个工件建立了一个工件坐标 B，并在这个工件坐标 B 中进行轨迹编程。



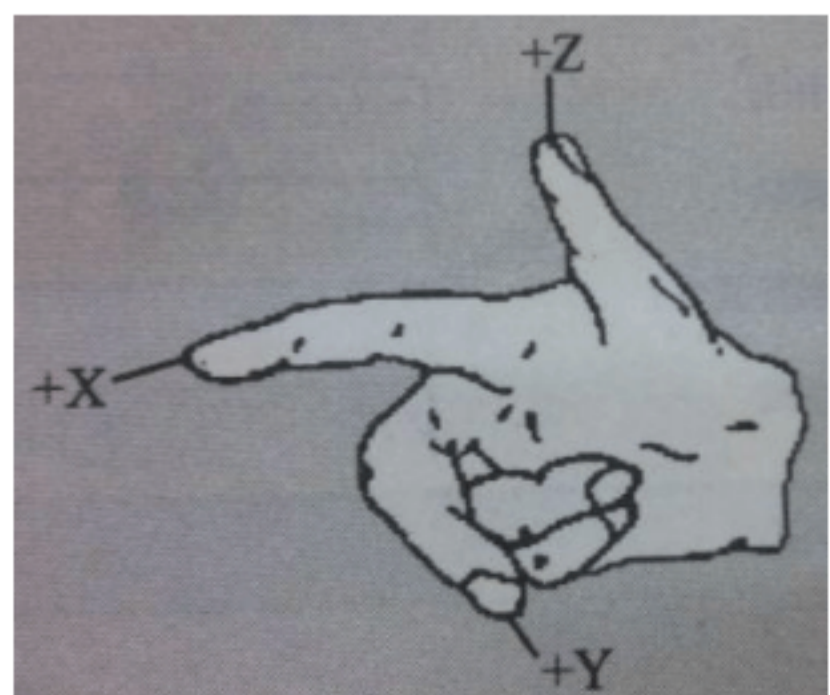
如果台子上还有一个一样的工件需要走一样的轨迹，那只需建立一个工件坐标 C，将工件坐标 B 中的轨迹复制一份，然后将工件坐标从 B 更新为 C，则无需对一样的工件进行重复轨迹编程了。

*提示：如果在工件坐标 B 中对 A 对象进行了轨迹编程，当工件坐标的位置变化成工件坐标 D 后，只需在机器人系统重新定义工件坐标 D，则机器人的轨迹就自动更新到 C 了，不需要再次轨迹编程了。因 A 相对于 B，C 相对于 D 的关系是一样，并没有因为整体偏移而发生变化。




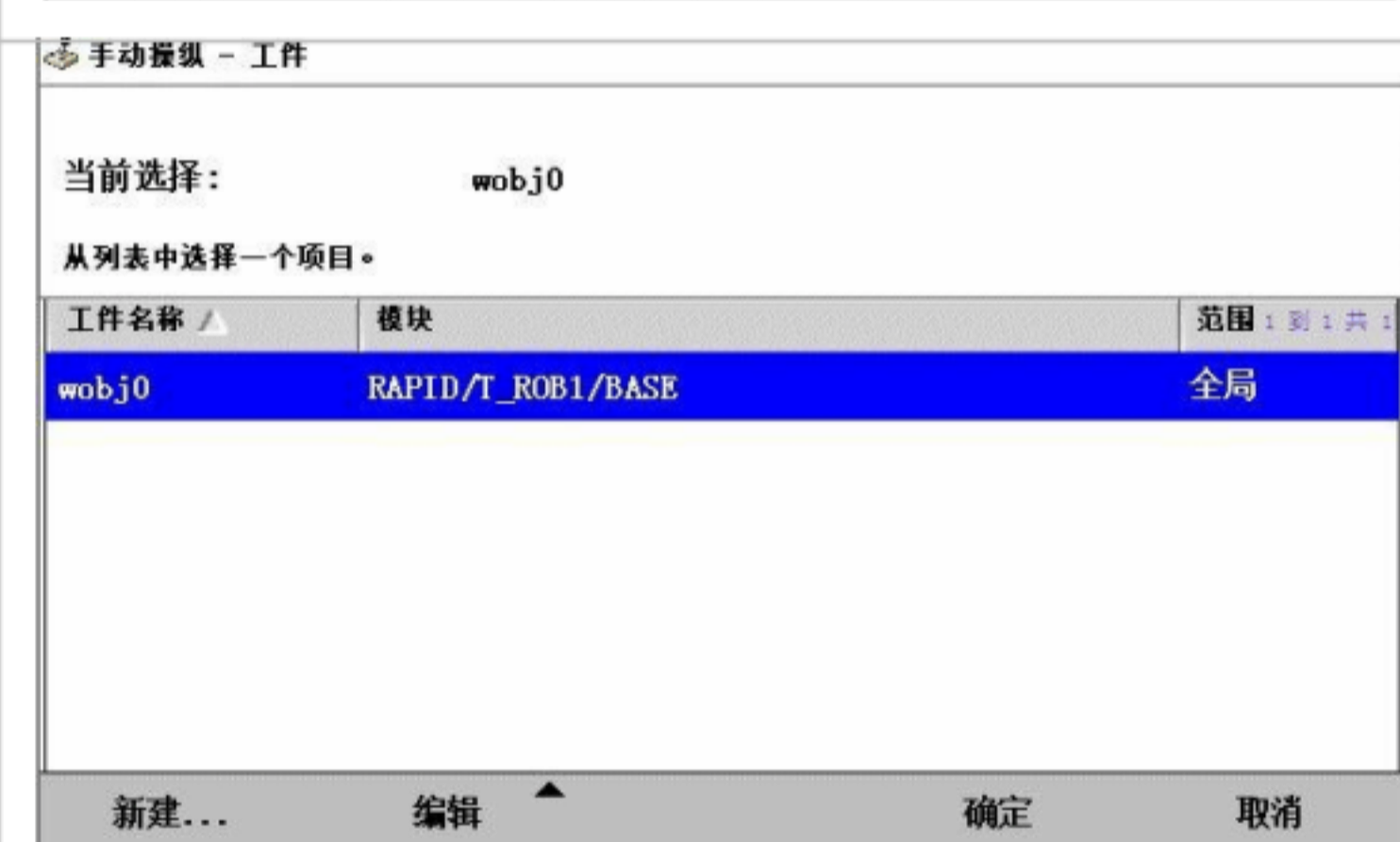

*注意：在对象的平面上，只需要定义三个点，就可以建立一个工件坐标。

- X1 点确定工件坐标的原点。
- X1、X2 点确定工件坐标 X 正方向
- Y1 确定工件坐标 Y 正方向。工件坐标等符合右手定则。

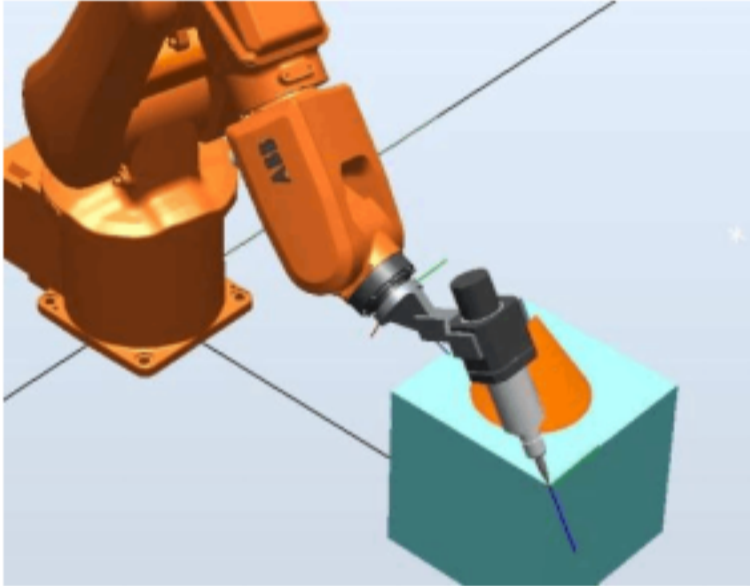
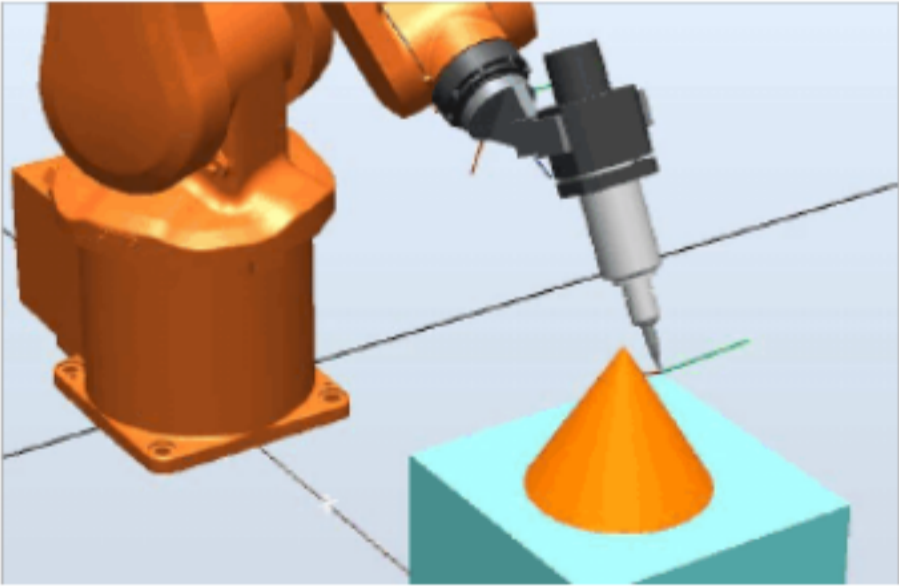




肇庆市技师学院

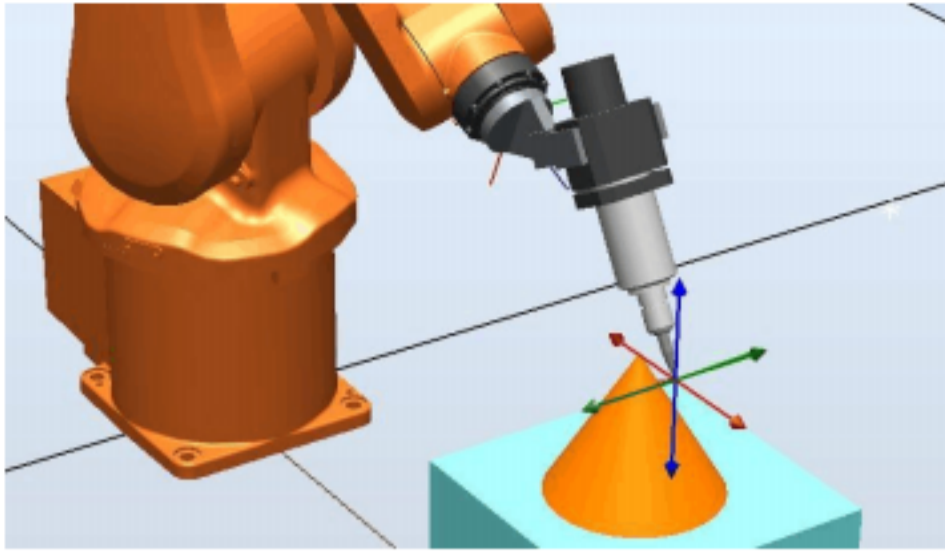
建立工件坐标的操作步骤：

	<p>1. 在手动操纵画面中，选择“工件坐标”。</p>						
 <table border="1"><thead><tr><th>工件名称</th><th>模块</th><th>范围</th></tr></thead><tbody><tr><td>wobj0</td><td>RAPID/T_ROB1/BASE</td><td>全局</td></tr></tbody></table>	工件名称	模块	范围	wobj0	RAPID/T_ROB1/BASE	全局	<p>2. 单击“新建”。</p>
工件名称	模块	范围					
wobj0	RAPID/T_ROB1/BASE	全局					
	<p>3. 对工件坐标数据属性进行设定后，单击“确定”。</p>						

 <p>手动操纵 - 工件</p> <p>当前选择: wobj1</p> <p>从列表中选择个项目。</p> <table border="1"> <thead> <tr> <th>工件名称</th> <th>模块</th> <th>范围</th> </tr> </thead> <tbody> <tr> <td>wobj0</td> <td>RAPID/T_ROB1/BASE</td> <td>全局</td> </tr> <tr> <td>wobj1</td> <td>RAPID/T_ROB1/Module1</td> <td>任务</td> </tr> </tbody> </table> <p>更改值... 更改声明... 复制 删除 定义...</p> <p>新建... 编辑 确定 取消</p>	工件名称	模块	范围	wobj0	RAPID/T_ROB1/BASE	全局	wobj1	RAPID/T_ROB1/Module1	任务	<p>4. 打开编辑菜单，选择“定义”。</p>
工件名称	模块	范围								
wobj0	RAPID/T_ROB1/BASE	全局								
wobj1	RAPID/T_ROB1/Module1	任务								
 <p>程序数据 - wobjdata - 定义</p> <p>工件坐标定义</p> <p>工件坐标: wobj1 活动工具: tool1</p> <p>为每个框架选择一种方法，修改位置后点击“确定”。</p> <p>用户方法: 未更改 目标方法: 未更改</p> <table border="1"> <thead> <tr> <th>点</th> <th>状态</th> </tr> </thead> <tbody> <tr> <td>未更改</td> <td></td> </tr> <tr> <td>3 点</td> <td></td> </tr> </tbody> </table> <p>位置 修改位置 确定 取消</p>	点	状态	未更改		3 点		<p>5. 将用户方法设定为“3点”。</p>			
点	状态									
未更改										
3 点										
	<p>6. 手动操纵机器人的工具参考点靠近定义工件坐标的 X1 点。</p>									

<p>程序数据 - wobjdata - 定义</p> <p>工件坐标定义 工件坐标: wobj1 活动工具: tool1</p> <p>为每个框架选择一种方法, 修改位置后点击“确定”。</p> <p>用户方法: 3点 目标方法: 未更改</p> <table border="1"> <thead> <tr> <th>点</th> <th>状态</th> </tr> </thead> <tbody> <tr> <td>用户点 X 1</td> <td>已修改</td> </tr> <tr> <td>用户点 X 2</td> <td>-</td> </tr> <tr> <td>用户点 Y 1</td> <td>-</td> </tr> </tbody> </table> <p>位置 修改位置 确定 取消</p>	点	状态	用户点 X 1	已修改	用户点 X 2	-	用户点 Y 1	-	<p>7. 单击“修改位置”, 将 X1 点记录下来。</p>
点	状态								
用户点 X 1	已修改								
用户点 X 2	-								
用户点 Y 1	-								
	<p>8. 手动操纵机器人的工具参考点靠近定义工件坐标的 X2 点。</p>								
<table border="1"> <thead> <tr> <th>点</th> <th>状态</th> </tr> </thead> <tbody> <tr> <td>用户点 X 1</td> <td>已修改</td> </tr> <tr> <td>用户点 X 2</td> <td>已修改</td> </tr> <tr> <td>用户点 Y 1</td> <td>-</td> </tr> </tbody> </table> <p>位置 修改位置 确定 取消</p>	点	状态	用户点 X 1	已修改	用户点 X 2	已修改	用户点 Y 1	-	<p>9. 单击“修改位置”, 将 X2 点记录下来。</p>
点	状态								
用户点 X 1	已修改								
用户点 X 2	已修改								
用户点 Y 1	-								
	<p>10. 手动操作机器人的工具参考点靠近定义工件坐标的 Y1 点。</p>								

	<p>11. 单击“修改位置”，将 Y1 点记录下来。 12. 单击“确定”。</p>
	<p>13. 对自动生成的工件坐标数据进行确认后，单击“确定”。</p>
	<p>14. 选中 wobj1 后，单击确定。</p>



15. 设定手动操纵画面项目，使用线性动作模式，体验新建立的工件坐标。

3.有效载荷 loaddata

对于搬运应用的机器人，应该正确设定夹具的质量、重心 tooldata 以及搬运对象的质量和重心数据 loaddata。



操作步骤：



1. 手动操纵 界面，选择 有效载荷”。

	<p>2. 单击“新建...”。</p>																																				
	<p>3. 对有效载荷数据属性进行设定。 4. 单击“初始值”。</p>																																				
<table border="1" data-bbox="205 1685 1297 2131"> <thead> <tr> <th>名称</th> <th>值</th> <th>数据类型</th> </tr> </thead> <tbody> <tr> <td>load1 :=</td> <td>[1, [0, 0, 50], [1, 0, 0, 0]]...</td> <td>loaddata</td> </tr> <tr> <td>mass :=</td> <td>1</td> <td>NUM</td> </tr> <tr> <td>cog:</td> <td>[0, 0, 50]</td> <td>POS</td> </tr> <tr> <td>x :=</td> <td>0</td> <td>NUM</td> </tr> <tr> <td>y :=</td> <td>0</td> <td>NUM</td> </tr> <tr> <td>z :=</td> <td>50</td> <td>NUM</td> </tr> </tbody> </table>	名称	值	数据类型	load1 :=	[1, [0, 0, 50], [1, 0, 0, 0]]...	loaddata	mass :=	1	NUM	cog:	[0, 0, 50]	POS	x :=	0	NUM	y :=	0	NUM	z :=	50	NUM	<p>5. 对有效载荷的数据根据实际情况进行设定，各参数代表的含义请参考下面的有效载荷参数表。 6. 单击“确定”。</p> <table border="1" data-bbox="1354 1528 1900 2389"> <thead> <tr> <th>名称</th> <th>参数</th> <th>单位</th> </tr> </thead> <tbody> <tr> <td>有效载荷质量</td> <td>load.mass</td> <td>kg</td> </tr> <tr> <td>有效载荷重心</td> <td>load.cog.x load.cog.y load.cog.z</td> <td>mm</td> </tr> <tr> <td>力矩方向</td> <td>load.aom.q1 load.aom.q2 load.aom.q3 load.aom.q4</td> <td></td> </tr> <tr> <td>有效载荷的动量</td> <td>ix iy iz</td> <td>kg · m²</td> </tr> </tbody> </table>	名称	参数	单位	有效载荷质量	load.mass	kg	有效载荷重心	load.cog.x load.cog.y load.cog.z	mm	力矩方向	load.aom.q1 load.aom.q2 load.aom.q3 load.aom.q4		有效载荷的动量	ix iy iz	kg · m ²
名称	值	数据类型																																			
load1 :=	[1, [0, 0, 50], [1, 0, 0, 0]]...	loaddata																																			
mass :=	1	NUM																																			
cog:	[0, 0, 50]	POS																																			
x :=	0	NUM																																			
y :=	0	NUM																																			
z :=	50	NUM																																			
名称	参数	单位																																			
有效载荷质量	load.mass	kg																																			
有效载荷重心	load.cog.x load.cog.y load.cog.z	mm																																			
力矩方向	load.aom.q1 load.aom.q2 load.aom.q3 load.aom.q4																																				
有效载荷的动量	ix iy iz	kg · m ²																																			

在 RAPID 编程中，需要对有效载荷的情况进行实时的调整：


```

10 TASK PERS loaddata load1:=[0,[0,0,0],[1,0,0,0],0,0,
11 PROC main()
12   Set do1;
13   GripLoad load1;
14   MoveJ *, v1000, z50, tool0;
15   MoveJ *, v1000, z50, tool0;
16   MoveJ *, v1000, z50, tool0;
17   Reset do1;
18   GripLoad load0;
19 ENDPROC
20
21 ENDMODULE
    
```

Set do1;夹具夹紧

GripLoad load1;指定当前搬运对象的质量和重心 load1

.....

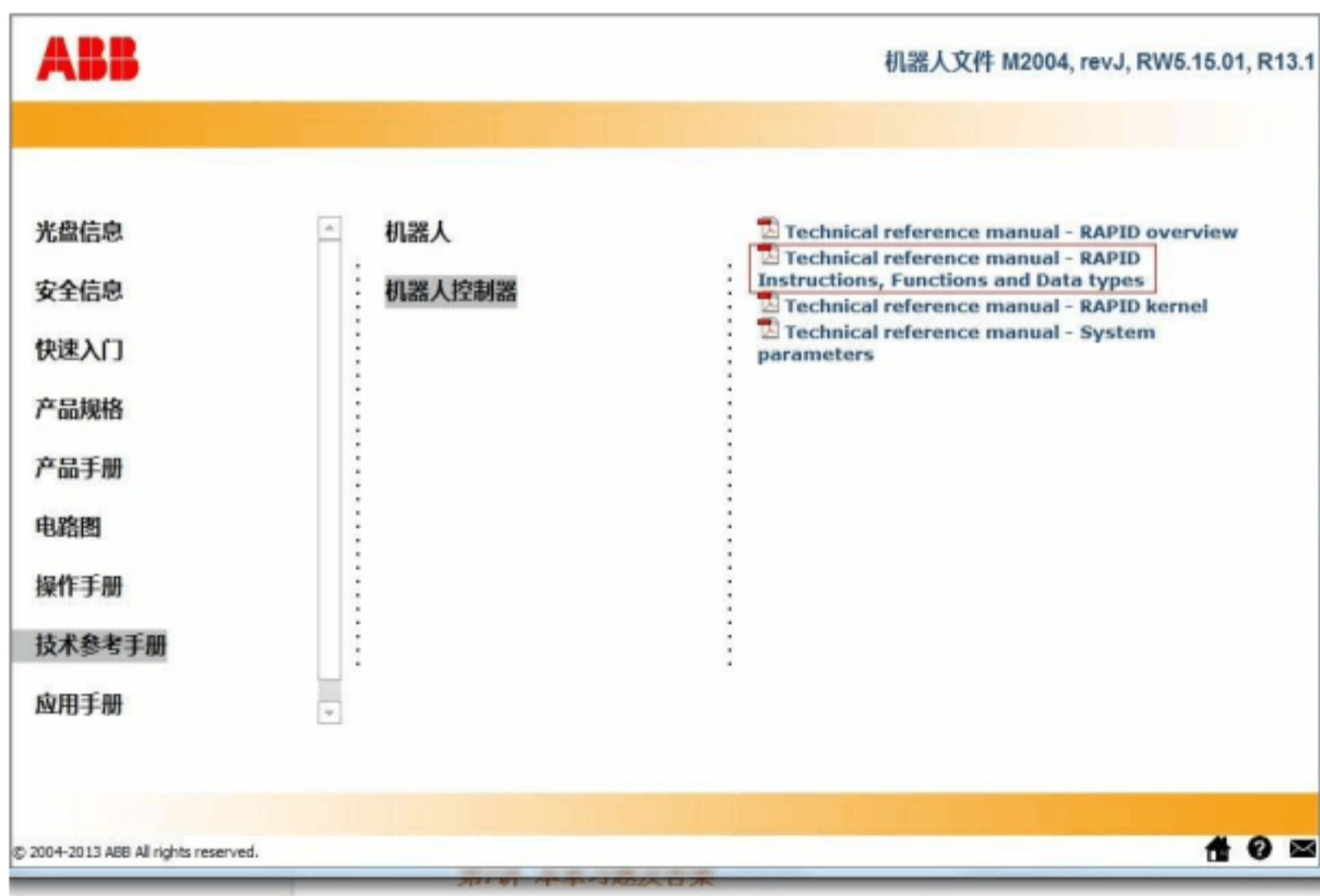
Reset do1;夹具松开

GripLoad load0;将搬运对象清除为 load0

5.5 知识链接

5.5.1 复杂程序数据赋值

在 RAPID程序数据中，有一些结构较为复杂的程序数据，如 rotarget 程序数据，即 MoveJ 指令中的 p10 数据：



如上图所示，在光盘的此文档中可以找到 RAPID程序中所有程序数据、功能、指令的详细介绍。

```
CONST robtarget p15 := [ [600, 500, 225.3], [1, 0, 0, 0], [1, 1, 0, 0], [ 11, 12.3, 9E9, 9E9, 9E9, 9E9] ];
```

文档中此数据是由一串数字组成（包括笛卡尔坐标 xyz、q1-4、轴角度等）

以此数据为例，介绍复杂数据的赋值操作。

首先查看此数据的架构：

<pre>< dataobject of robtarget > < trans of pos > < x of num > < y of num > < z of num > < rot of orient > < q1 of num > < q2 of num > < q3 of num > < q4 of num > < robconf of confdata > < cf1 of num > < cf4 of num > < cf6 of num > < cfx of num > < extax of extjoint > < eax_a of num > < eax_b of num > < eax_c of num > < eax_d of num > < eax_e of num > < eax_f of num ></pre>	<p>以修改 trans of pos 中的 x 为例。 操作步骤： 1. 首先确定程序数据的类型为可变量 2. 打开程序编辑器进入例行程序 添加赋值指令 "p10.trans.x:=400"（即将 p10 的 trans 下的 x 的值更改为 400）。</p>
--	--

5.5.2 工具自动识别程序

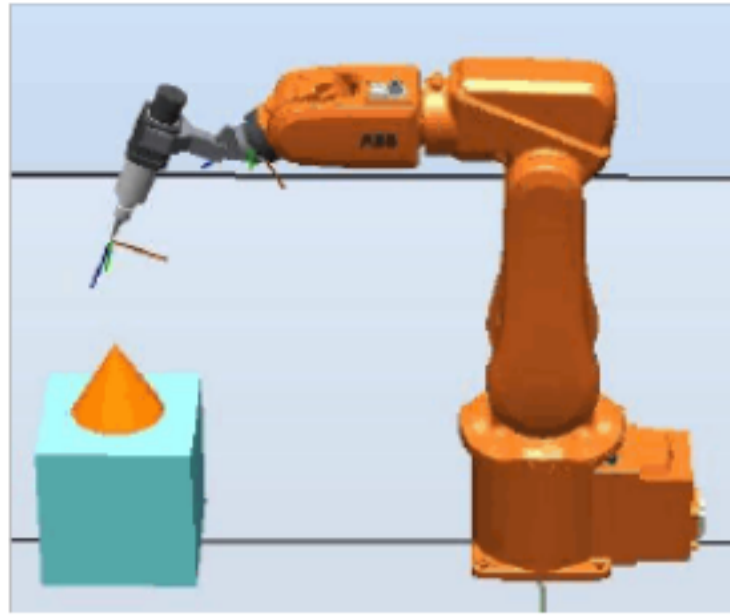
介绍工具自动识别（ LoadID ）功能。

LoadIdentify 是 ABB 机器人开发的用于自动识别安装在六轴法兰盘上的工具（ tooldata ）和载荷（ loaddata ）的重量，以及重心。（前面介绍到，设置 tooldata 和 loaddata 是自己测量工具的重量和重心，然后填写参数进行设置，但是这样会有一些的不准确性）

手持工具的应用中，应使用 LoadIdentify 识别工具的重量和重心。手持夹具

的应用中，应使用 LoadIdentify 识别夹具和搬运对象的重量和重心。操作步

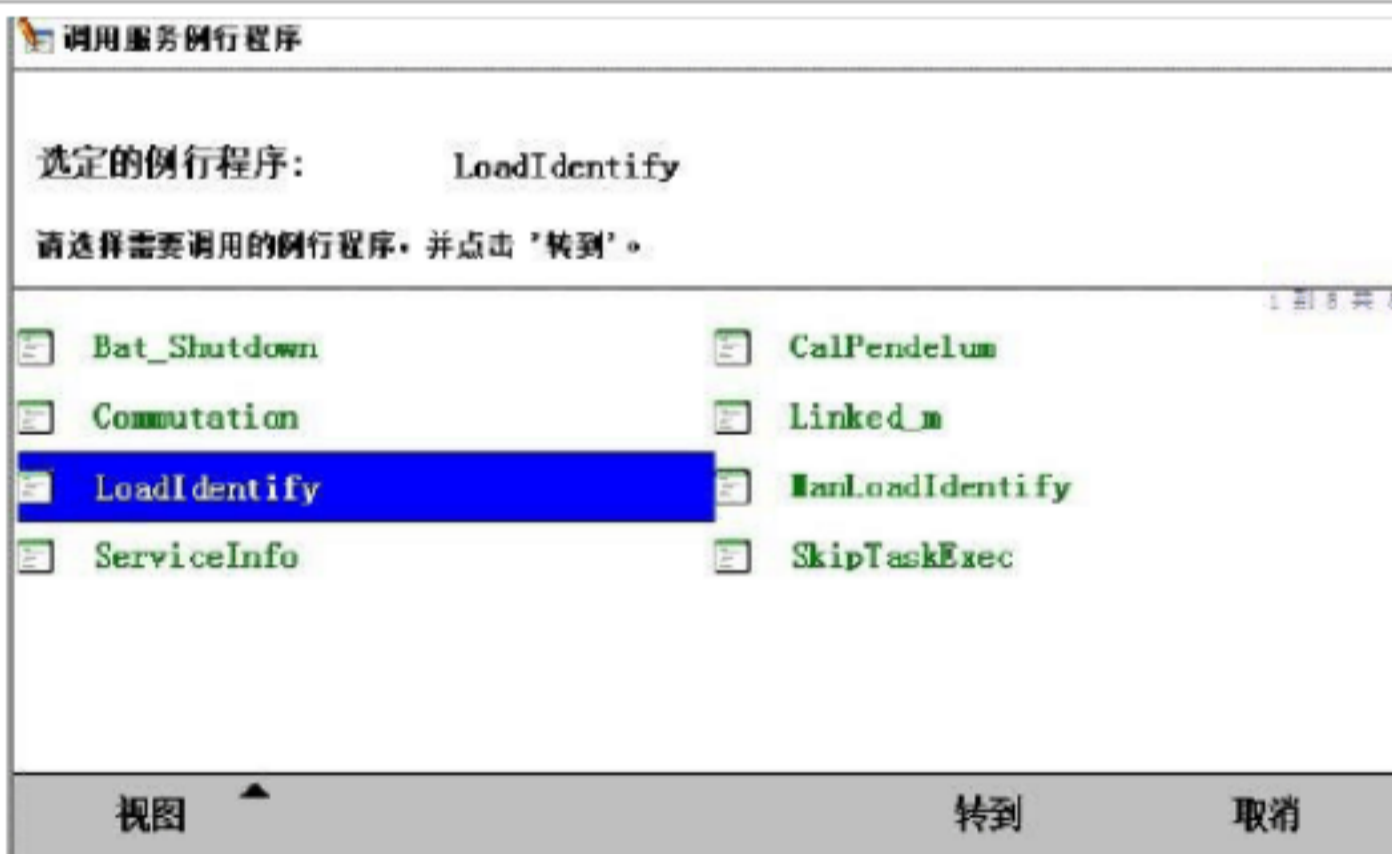
骤：



1. 使用手动操纵功能，把机器人回到机械原点位置。



2. 进入“手动操纵”-“工具坐标”画面，选取需要测量的工具数据（如果有载荷选择测量的载荷）。


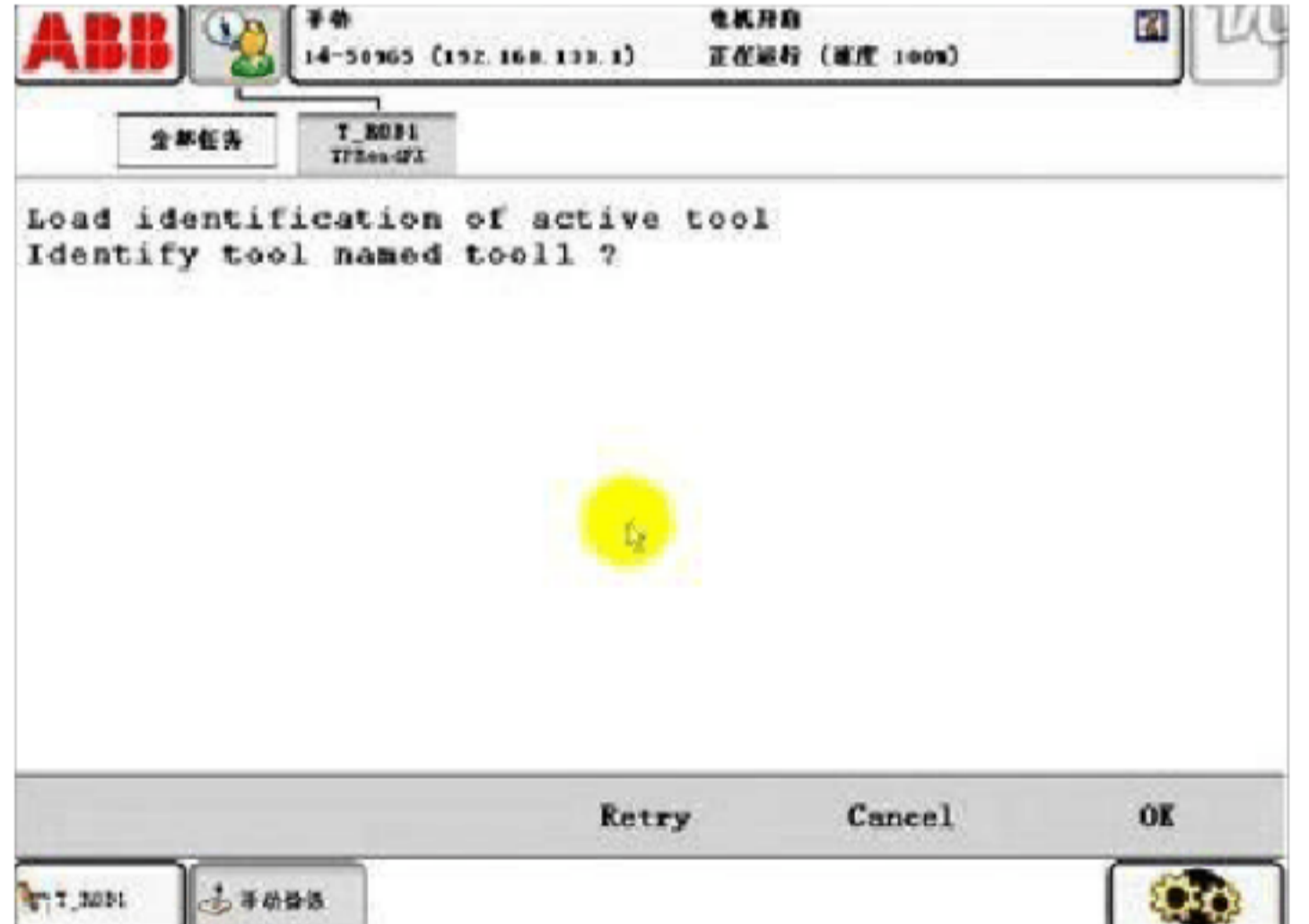


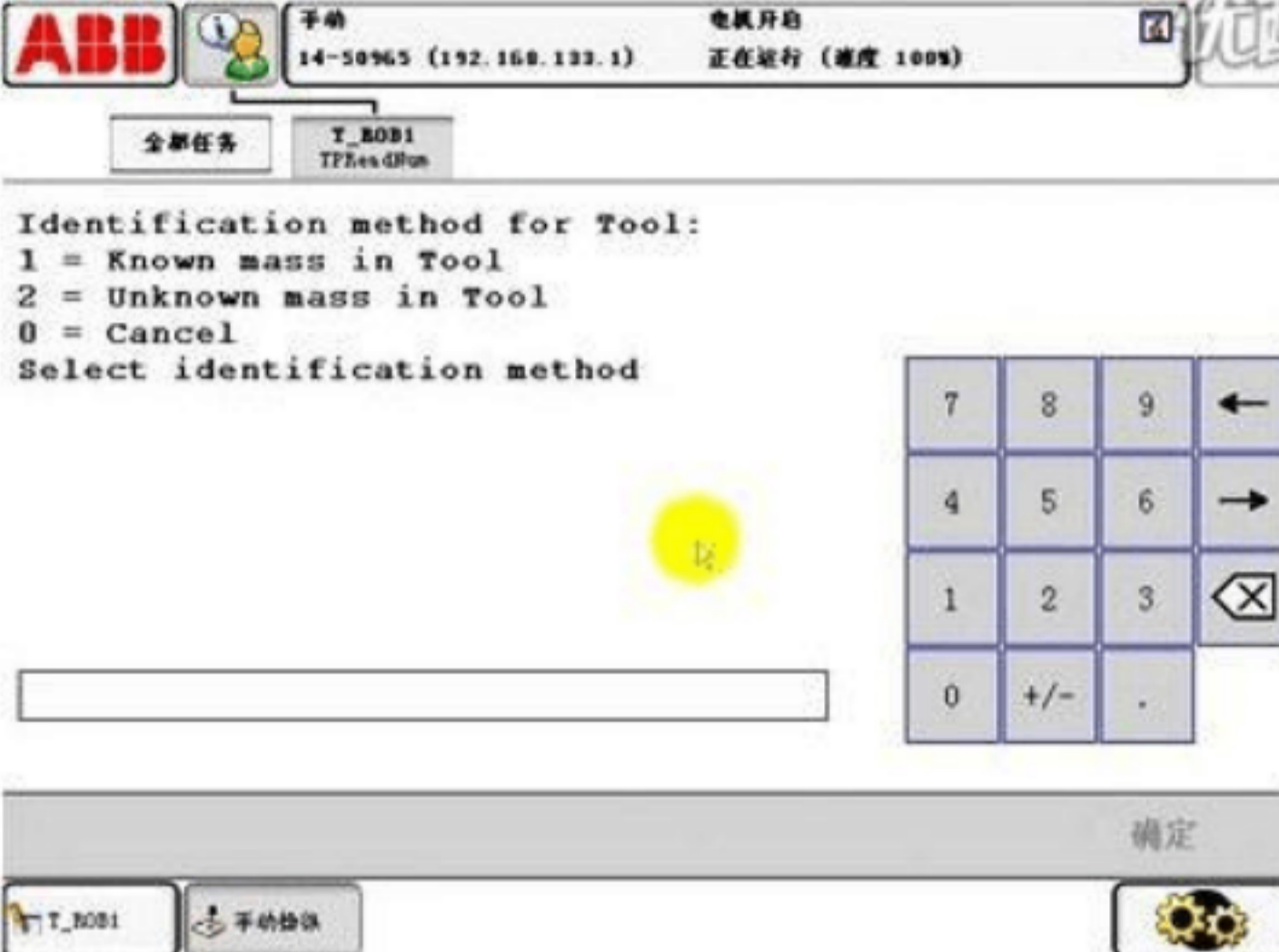
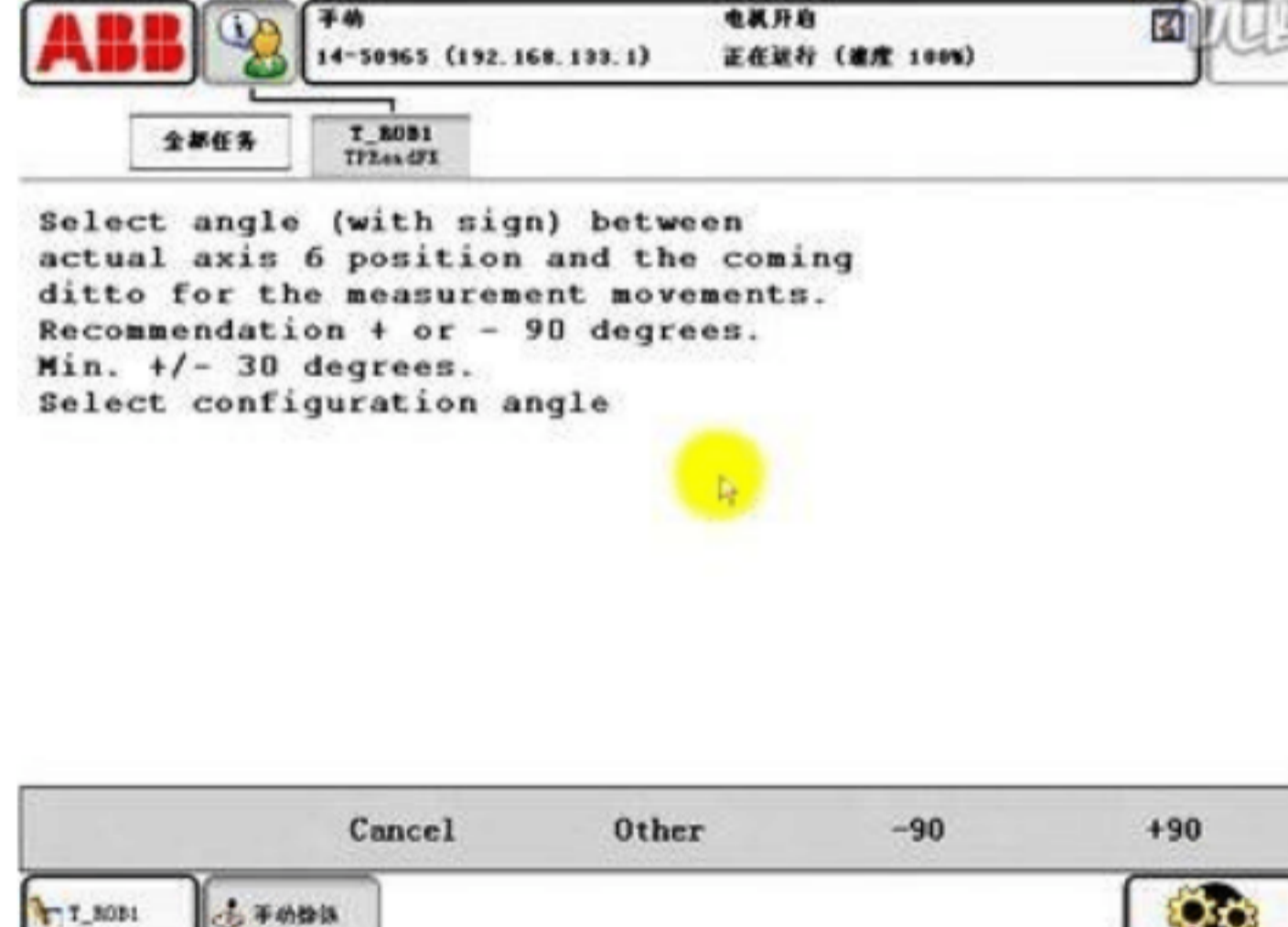
3. 进入“程序编辑器”画面，单击“调试”-选择“调用例行程序”-选择‘LoadIdentify’（此程序为标准程序）-单击‘转到’。



4. 按下使能键，点击示教器右下侧的播放键运行程序，在弹出的对话框中点击‘OK’。

	
	<p>5. 根据提示选择 “Tool” (即选择要测量的是工具还是载荷)。</p>

 <p>The screenshot shows the ABB robot control interface. At the top, it displays the ABB logo, a status bar with '手动' (Manual) and '电机启动' (Motor Start) indicators, and the IP address '14-50965 (192.168.133.1)'. Below this, there are buttons for '全部任务' (All Tasks) and 'T_0001 TTool-01X'. The main text area contains the following instructions:</p> <ul style="list-style-type: none">* The Tool to identify must be mounted on robot, defined with name and active in the jogging window.* Mounted arm load must be defined.* Suitable robot axes (1-6) positions. <p>Confirm !</p> <p>At the bottom, there are 'Cancel' and 'OK' buttons, and a '手动切换' (Manual Switch) button with a gear icon.</p>	<p>6. 确认六轴是否在合适位置（不必为机械原点）。</p>
 <p>The screenshot shows the ABB robot control interface. At the top, it displays the ABB logo, a status bar with '手动' (Manual) and '电机启动' (Motor Start) indicators, and the IP address '14-50965 (192.168.133.1)'. Below this, there are buttons for '全部任务' (All Tasks) and 'T_0001 TTool-01X'. The main text area contains the following text:</p> <p>Load identification of active tool Identify tool named tool1 ?</p> <p>A yellow circular cursor is visible in the center of the text area.</p> <p>At the bottom, there are 'Retry', 'Cancel', and 'OK' buttons, and a '手动切换' (Manual Switch) button with a gear icon.</p>	<p>7. 确认工具数据名称。</p>

 <p>ABB 14-50965 (192.168.133.1) 电机开启 正在运行 (速度 100%)</p> <p>全部任务 T_ROB1 TPFeedFun</p> <p>Identification method for Tool: 1 = Known mass in Tool 2 = Unknown mass in Tool 0 = Cancel Select identification method</p> <p>7 8 9 ← 4 5 6 → 1 2 3 ⊗ 0 +/- .</p> <p>确定</p> <p>T_ROB1 手动操作</p>	<p>8. 选择工具重量 (选择 2, 然机器人自己识别重量)。</p>
 <p>ABB 14-50965 (192.168.133.1) 电机开启 正在运行 (速度 100%)</p> <p>全部任务 T_ROB1 TPFeedFun</p> <p>Select angle (with sign) between actual axis 6 position and the coming ditto for the measurement movements. Recommendation + or - 90 degrees. Min. +/- 30 degrees. Select configuration angle</p> <p>Cancel Other -90 +90</p> <p>T_ROB1 手动操作</p>	<p>9. 调整旋转角度 (如果工具不能进行 90 度旋转, 要进行设置)。</p>

The screenshot displays the ABB robot control interface. At the top, the ABB logo is on the left, and the status bar shows '手动' (Manual) mode, IP address '14-58965 (192.168.133.1)', and '电机启动 正在运行 (速度 100%)' (Motor start running at 100% speed). Below the status bar are buttons for '全部任务' (All tasks) and 'T_0001 TP2es-0FX'. The main text area contains the following message: 'Should test of measurement movements first be done with low speed in MAN/RED. SPEED before the real measurements with higher speed Run test with slow speed ?'. A yellow circle highlights the question mark. Below the text is a dialog box with 'No' and 'Yes' options. The 'Yes' button is highlighted with a yellow circle and contains a gear icon. Below the dialog box are buttons for 'T_0001' and '手动测试' (Manual test). At the bottom, there is a 'MOVE' button highlighted with a yellow circle. The text 'Press MOVE for slow test movements' is visible below the dialog box.

10. 进行慢速测试。



11. 等待机器人完成测试步骤，观察机器人动作是否有被干涉，一直按住使能键（使能键如果断开，需要重新开始测试过程）。



12. 切换到自动状态，点击播放键，重新进入识别程序画面，点击“MOVE”。

肇庆市技师学院

	<p>13. 完成后跳到画面，切换为手动，显示测量结果（包括重量、重心、准确度等），确认无误后，点击“Yes”将结果写入工具数据。</p>
	<p>14. 点击“取消调用例行程序”回到程序编辑画面。</p>