

机器人编程 1

库卡系统软件 8.2

KUKA Roboter GmbH (库卡机器人有限公司) 培训资料

发布日期 : 13.10.2011

版本 : COL P1KSS8 Roboterprogrammierung 1 V1 zh



© 版权 2011

KUKA Roboter GmbH
Zugspitzstraße 140
D-86165 Augsburg
德国

此文献或节选只有在征得库卡机器人集团公司明确同意的情况下才允许复制或对外方开放。

除了本文献中说明的功能外，控制系统还可能具有其他功能。但是在新供货或进行维修时，无权要求库卡公司提供这些功能。

我们已就印刷品的内容与描述的硬件和软件内容是否一致进行了校对。但是不排除有不一致的情况，我们对此不承担责任。但是我们定期校对印刷品的内容，并在之后的版本中作必要的更改。

我们保留在不影响功能的情况下进行技术更改的权利。

原版文件的翻译

KIM-PS5-DOC

Publication: Pub COLLEGE P1KSS8 Roboterprogrammierung 1 zh

Bookstructure: P1KSS8 Roboterprogrammierung 1 V4.2

版本: COL P1KSS8 Roboterprogrammierung 1 V1 zh

目录

1	KUKA 机器人系统的结构和功能	5
1.1	机器人技术入门	5
1.2	库卡机器人的机械系统	5
1.3	机器人控制系统 (V)KR C4	8
1.4	KUKA smartPAD	9
1.5	smartPAD 概览	10
1.6	机器人编程	11
1.7	机器人安全性	12
2	机器人运动	15
2.1	读取并解释机器人控制系统的信息提示	15
2.2	选择并设置运行方式	16
2.3	单独运动机器人的各轴	18
2.4	与机器人相关的坐标系	21
2.5	机器人在世界坐标系中运动	23
2.6	在工具坐标系中移动机器人	27
2.7	在基坐标系中移动机器人	31
2.8	练习：操作及手动移动	35
2.9	用一个固定工具进行手动移动	37
2.10	练习：用固定的工具练习手动移动	38
3	机器人的投入运行	39
3.1	零点标定的原理	39
3.2	给机器人标定零点	41
3.3	练习：机器人零点标定	45
3.4	机器人上的负载	47
3.4.1	工具负载数据	47
3.4.2	机器人上的附加负载	48
3.5	工具测量	49
3.6	练习：尖触头的工具测量	58
3.7	练习：抓爪工具测量，2 点法	61
3.8	测量基坐标	63
3.9	查询当前机器人位置	66
3.10	练习：工作台的基坐标测量，3 点法	68
3.11	固定工具测量	70
3.12	测量由机器人引导的工件	71
3.13	练习：测量外部工具和机器人引导的工件	73
3.14	拔出 smartPAD	77
4	执行机器人程序	81
4.1	执行初始化运行	81
4.2	选择和启动机器人程序	82
4.3	练习：执行机器人程序	86
5	程序文件的使用	89
5.1	创建程序模块	89
5.2	编辑程序模块	90
5.3	存档和还原机器人程序	91

5.4	通过运行日志了解程序和状态变更	92
6	建立及更改编程的运动	95
6.1	创建新的运动指令	95
6.2	创建已优化节拍时间的运动（轴运动）	96
6.3	练习：空运转程序 - 程序操作和 PTP 运动	100
6.4	创建沿轨迹的运动	102
6.5	更改运动指令	108
6.6	练习：沿轨迹运行和轨迹逼近	111
6.7	具有外部 TCP 的运动编程	114
6.8	练习：以外部 TCP（工具中心点）进行运动编程	114
7	利用机器人程序中的逻辑功能	117
7.1	逻辑编程入门	117
7.2	等待功能的编程	117
7.3	简单切换功能的编程	121
7.4	轨迹切换功能编程	123
7.5	练习：逻辑指令和切换功能	128
8	变量的使用	131
8.1	显示和更改变量值	131
8.2	询问机器人状态	132
8.3	练习：显示系统变量	133
9	使用工艺程序包	135
9.1	使用 KUKA.GripperTech 操作抓爪	135
9.2	用 KUKA.GripperTech 对抓爪编程	135
9.3	KUKA.GripperTech 配置	138
9.4	练习：抓爪编程“标牌”	140
9.5	练习：以尖触头为对象对抓爪进行编程	142
10	在 KRL 中成功编程	145
10.1	机器人程序的结构和组成	145
10.2	结构化机器人程序	150
10.3	链接机器人程序	153
10.4	练习：在 KRL 中编程	155
11	使用上级控制系统作业	159
11.1	由 PLC 启动程序的准备工作	159
11.2	调整 PLC 的连接 (Cell.src)	160
	索引	163

1 KUKA 机器人系统的结构和功能

1.1 机器人技术入门

什么是机器人？

机器人 *Roboter* 这个概念源于斯拉夫语中的字 *robota*，意为 *重活*。

工业机器人这个词的官方定义为：“机器人是一种可自由编程并受程序控制的操作机。”

控制系统、操作设备以及连接电缆和软件也同样属于机器人的范畴。



图 1-1: 工业机器人

- 1 控制系统（控制柜（V）KR C4）
- 2 机械手（机器人机械系统）
- 3 手持操作和编程器（库卡 smartPAD）

所有不包括在工业机器人系统内的设备被称为 *外围设备*，它们是：

- 工具（效应器 /Tool）
- 保护装置
- 皮带输送机
- 传感器
- 等等

1.2 库卡机器人的机械系统

什么是机械手？

机械手是机器人机械系统主体。它由众多活动的、相互连接在一起的关节（轴）组成。我们也称之为运动链。

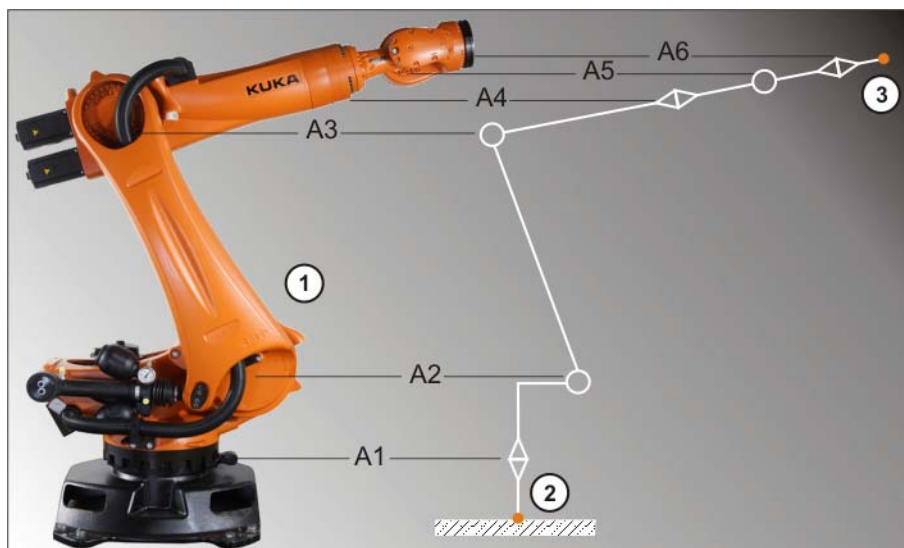


图 1-2: 机械手

- 1 机械手 (机器人机械系统)
- 2 运动链的起点 : 机器人足部 (ROBROOT)
- 3 运动链的开放端 : 法兰 (FLANGE)
- A1 机器人轴 1 至 6
- ...
- A6

各根轴的运动通过伺服电机有针对性的调控而实现。这些伺服电机通过减速器与机械手的各部件相连。



图 1-3: 机器人的机械零部件概览

- | | |
|--------|-------|
| 1 底座 | 4 连杆臂 |
| 2 转盘 | 5 手臂 |
| 3 平衡配重 | 6 手 |

机器人机械系统的部件主要由铸铝和铸钢制成。在个别情况下也使用碳纤维部件。

各根轴从下 (机器人的足部) 到上 (机器人法兰) 编号 :



图 1-4: 库卡机器人自由度

以下是库卡产品系列中机械手的技术数据选摘

- 轴数：4（SCARA 机器人和平行四边形机器人）至 6（标准垂直折臂机器人）
- 作用范围：从 0.35m (KR 5 scara) 直至 3.9 m (KR 120 R3900 ultra K)
- 自重：从 20 kg 直至 4700 kg
- 精确度：0.015 mm 至 0.2 mm 重复精度。

基本轴 A 1 至 A 3 以及机器人手轴 A 5 的轴范围均由带缓冲器的机械终端止挡限定。



附加轴上可安装另外的机械终端止挡。

⚠ 危险 如机器人或一个附加轴在行驶中撞到障碍物、机械终端止挡位置上或轴范围限制装置处的缓冲器，则会导致机器人系统受损。将机器人系统重新投入运行之前，需先联系库卡机器人有限公司。在继续运行工业机器人之前，被撞到的缓冲器必须立即用新的替换。如机器人（附加轴）以超过 250 mm/s 的速度撞到缓冲器，则必须更换机器人（附加轴）或由库卡机器人有限公司重新投入运行。

1.3 机器人控制系统 (V)KR C4

谁控制机器人的运动？

机器人机械系统由伺服电机控制运动，而该电机则由 (V)KR C4 控制系统控制。



图 1-5: 控制柜 (V) KR C4

(V)KR C4 控制系统的属性

- 机器人控制系统（轨迹规划）：控制六个机器人轴以及最多两个附加的外部轴。



图 1-6: (V)KR C4 轴调节器

- 流程控制系统：符合 IEC61131 标准的集成式 Soft PLC
- 安全控制系统
- 运动控制系统
- 通过总线系统（例如：ProfiNet、以太网 IP、Interbus）的通讯可能性：

- 可编程控制器 (PLC)
- 其它控制系统
- 传感器和执行器
- 通过网络的通讯可能性 :
 - 主机
 - 其它控制系统



图 1-7: (V) KR C4 的通讯途径

1.4 KUKA smartPAD

如何操作 KUKA 机器人？ KUKA 机器人的操作通过手持操作器（即 KUKA smartPAD）进行。



图 1-8

KUKA smartPAD 的特点：

- 触摸屏（触摸式操作界面），用手或配备的触摸笔操作
- 大尺寸竖型显示屏
- KUKA 菜单键
- 八个移动键
- 操作工艺数据包的按键
- 用于程序运行的按键（停止 / 向前 / 向后）

- 显示键盘的按键
- 更换运行方式的钥匙开关
- 紧急停止按键
- 3D 鼠标
- 可拔出
- USB 接口

1.5 smartPAD 概览

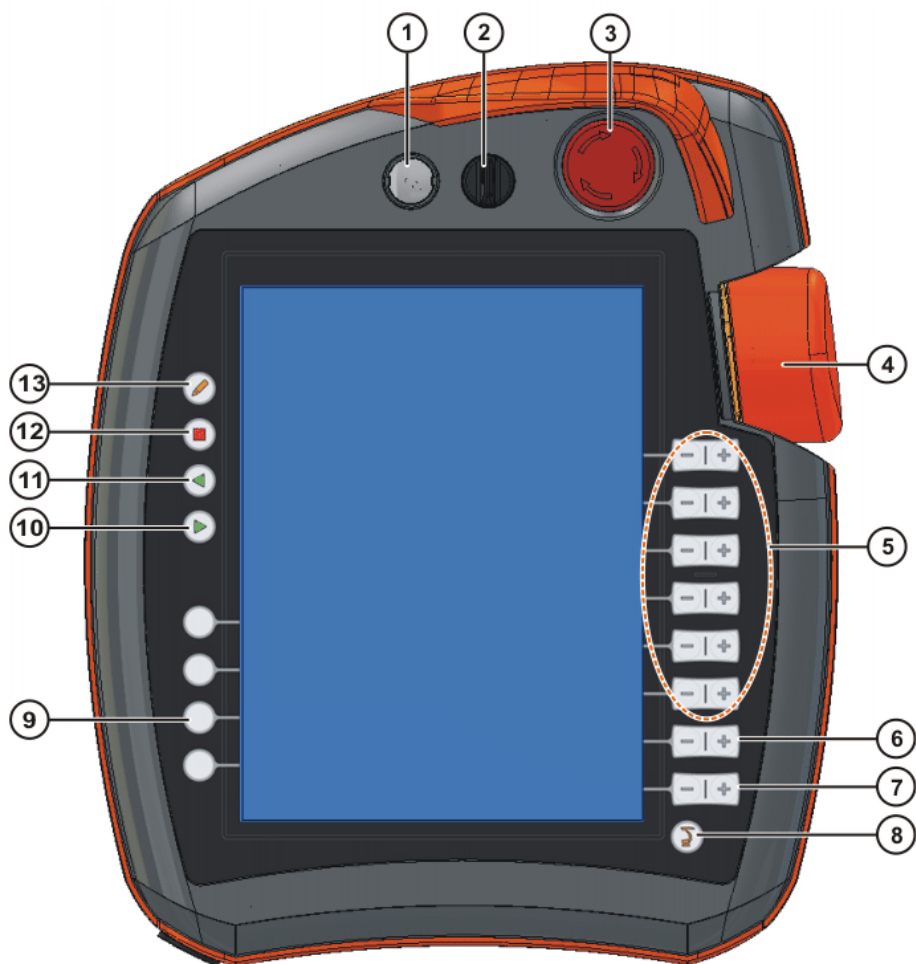


图 1-9

序号	说明
1	用于拔下 smartPAD 的按钮
2	用于调出连接管理器的钥匙开关。只有当钥匙插入时，方可转动开关。 可以通过连接管理器切换运行模式。
3	紧急停止键。用于在危险情况下关停机器人。紧急停止键在被按下时将自行闭锁。
4	3D 鼠标。用于手动移动机器人。

序号	说明
5	移动键。用于手动移动机器人。
6	用于设定程序倍率的按键
7	用于设定手动倍率的按键
8	主菜单按键。用来在 smartHMI 上将菜单项显示出来。
9	工艺键。工艺键主要用于设定工艺程序包中的参数。其确切的功能取决于所安装的工艺程序包。
10	启动键。通过启动键可启动一个程序。
11	逆向启动键。用逆向启动键可逆向启动一个程序。程序将逐步运行。
12	停止键。用停止键可暂停正运行中的程序。
13	键盘按键 显示键盘。通常不必特地将键盘显示出来，smartHMI 可识别需要通过键盘输入的情况并自动显示键盘。

1.6 机器人编程

通过机器人编程可保证运动过程和流程将自动完成并始终可反复。为此，控制器需要大量的信息：

- 机器人位置 = 工具的空间位置
- 动作类型
- 速度 / 加速
- 等候条件、分支、相关性等信号信息

控制器使用哪种语言？

编程语言是 KRL - KUKA Robot Language (库卡机器人编程语言)

程序举例：

```
PTP P1 Vel=100% PDAT1
PTP P2 CONT Vel=100% PDAT2
WAIT FOR IN 10 'Part in Position'
PTP P3 Vel=100% PDAT3
```

KUKA 机器人如何编程？

KUKA 机器人可用不同的编程方法编程：

- 以示教 (Teach-in) 法**在线编程**。



图 1-10: 可利用库卡 smartPAD 进行机器人编程

■ 离线编程

- 图形辅助的互动编程：模拟机器人过程

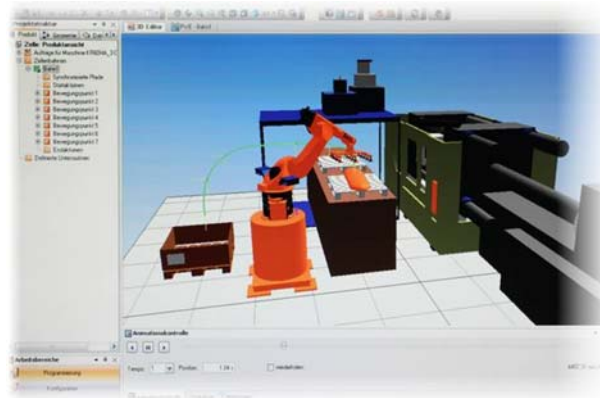


图 1-11: 用 KUKA WorkVisual 模拟

- 文字编程：借助于 smartPAD 界面在上级操作 PC 上的显示编程（也适用于诊断、在线适配调整已运行的程序）



图 1-12: 用 KUKA OfficeLite 进行机器人编程

1.7 机器人安全性

机器人系统必须始终装备相应的安全设备。例如：隔离性防护装置（防护栅、门等等）、紧急停止按键、失知制动装置、轴范围限制装置等等。

示例：学院培训间

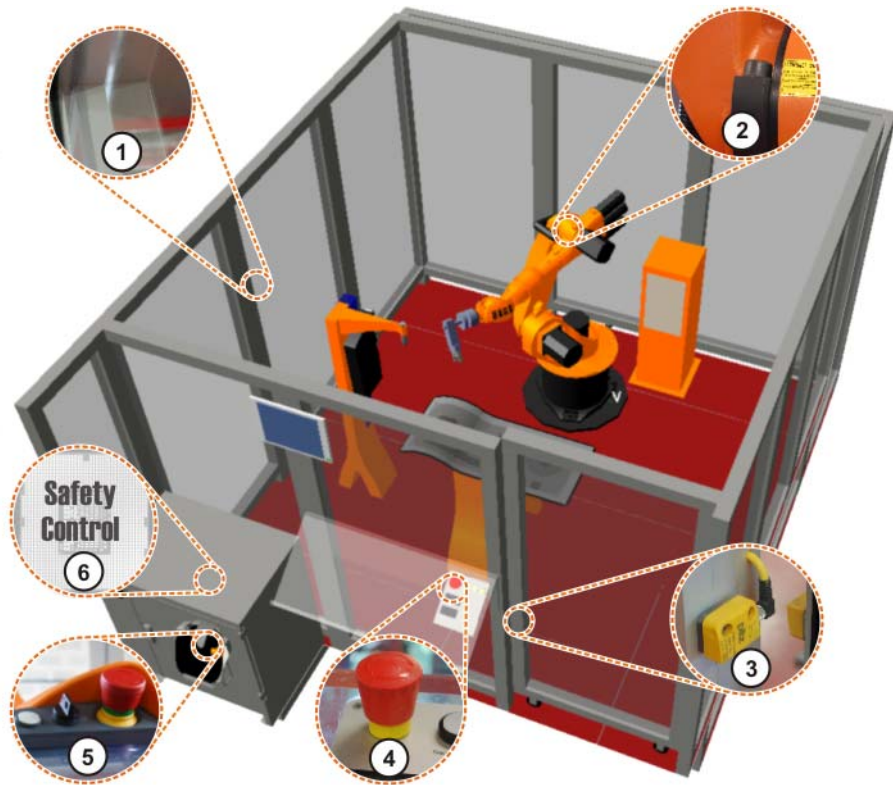


图 1-13: 培训间

- 1 防护栅
- 2 轴 1、2 和 3 的机械终端止挡或者轴范围限制装置
- 3 防护门及具有关闭功能监控的门触点
- 4 紧急停止按钮（外部）
- 5 紧急停止按钮、确认键、调用连接管理器的钥匙开关
- 6 内置的 (V)KR C4 安全控制器

危险 在安全防护装置功能不完善的情况下，机器人系统可能会导致人员受伤或财产损失。在安全防护装置被拆下或关闭的情况下，不允许运行机器人系统。

急停装置

工业机器人的紧急停止装置是位于 KCP（库卡控制面板）上的紧急停止按钮。在出现危险情况或紧急情况时必须按下此按钮。

按下紧急停止按钮时，工业机器人的反应：

- 机械手及附加轴（可选）以安全停止 1 的方式停机。

若欲继续运行，则必须旋转紧急停止按钮以将其解锁，接着对停机信息进行确认。

警告 与机械手相连的工具或其他装置如可能引发危险，则必须将其连入设备侧的紧急停止回路中。如果没有遵照执行这一规定，则可能会造成死亡、严重身体伤害或巨大的财产损失。

至少要安装一个外部紧急停止装置，以确保即使在 KCP 已拔出的情况下也有紧急停止装置可供使用。

外部紧急停止

在每个可能引发机器人运动或其他可能带来危险情况的工位上都必须有紧急停止装置可供使用。系统集成商应对此承担责任。

至少要安装一个外部紧急停止装置，以确保即使在 KCP 已拔出的情况下也有紧急停止装置可供使用。

外部紧急停止装置通过客户方的接口连接。外部紧急停止装置不包括在工业机器人的供货范围中。

操作人员防护装置

操作人员防护装置信号用于锁闭隔离性防护装置，如防护门。没有此信号，就无法使用自动运行方式。如果在自动运行期间出现信号缺失的情况（例如防护门被打开），则机械手将以安全停止 1 的方式停机。

在手动慢速测试运行方式（T1）和手动快速测试运行方式（T2）下，操作人员防护装置不启用。



警告 在出现信号缺失后，不允许仅仅通过关闭防护装置来重新继续自动运行方式，而是要先进行确认。系统集成商必须对此负责。由此可以避免在危险区域中有人停留时因疏忽比如防护门意外闭合而继续进行自动运行。

- 确认必须被设置为可事先对危险区域进行实际检查。不具备此种设置的确认（比如它在防护装置关闭时自动确认）是不允许的。
- 如果没有注意这一点，则可能会造成人员死亡、严重身体伤害或巨大的财产损失。

安全运行停止

安全停止可通过客户接口上的输入端触发。该状态在外部信号为 FALSE 时一直保持。当外部信号为 TRUE 时，机械手可以重新被操作。此处无需确认。

外部安全停止 1 和 外部安全停止 2

安全停止 1 和安全停止 2 可通过客户接口上的输入端触发。该状态在外部信号为 FALSE 时一直保持。当外部信号为 TRUE 时，机械手可以重新被操作。此处无需确认。

2 机器人运动

2.1 读取并解释机器人控制系统的信息提示

信息提示概览



图 2-1: 信息窗口和信息提示计数器

- 1 信息窗口：显示当前信息提示
- 2 信息提示计数器：每种信息提示类型的信息提示数

控制器与操作员通信通过信息窗口实现。其中有五种信息提示类型：

信息提示类型概览：

图标	类型
	确认信息 <ul style="list-style-type: none"> ■ 用于显示需操作员确认才能继续处理机器人程序的状态。（例如：“确认紧急停止”） ■ 确认信息始终引发机器人停止或抑制其起动。
	状态信息 <ul style="list-style-type: none"> ■ 状态信息报告控制器的当前状态。（例如：“紧急停止”） ■ 只要这种状态存在，状态信息便无法被确认。
	提示信息 <ul style="list-style-type: none"> ■ 提示信息提供有关正确操作机器人的信息。（例如：“需要启动键”） ■ 提示信息可被确认。只要它们不使控制器停止，则无需确认。
	等待信息 <ul style="list-style-type: none"> ■ 等待信息说明控制器在等待哪一事件（状态、信号或时间）。 ■ 等待信息可通过按“模拟”按键手动取消。

警告 指令“模拟”只允许在能够排除碰撞和其他危险的情况下使用！

	对话信息 <ul style="list-style-type: none"> ■ 对话信息用于与操作员的直接通讯 / 问询。 ■ 将出现一个含各种按键的信息窗口，用这些按键可给出各种不同的回答。
--	---

用“OK”可对可确认的信息提示加以确认。用“全部 OK”可一次性全部确认所有可以被确认的信息提示。

信息的影响

信息会影响机器人的功能。确认信息始终引发机器人停止或抑制其起动。为了使机器人运动，首先必须对信息予以确认。

指令“OK”（确认）表示请求操作人员有意识地对信息进行分析。

i 对信息处理的建议：

- 有意识地阅读！
- 首先阅读较老的信息。较新的信息可能是老信息产生的后果。
- 切勿轻率地按下“Alle OK”。
- 尤其是在启动后：仔细查看信息。在此过程中让所有信息都显示出来（按下信息窗口即扩展信息列表）

信息提示处理

信息提示中始终包括日期和时间，以便为研究相关事件提供准确的时间。



图 2-2: 确认信息

观察和确认信息提示的操作步骤。

1. 触摸信息窗口 (1) 以展开信息提示列表。
2. 确认：
 - 用“OK” (2) 来对各条信息提示逐条进行确认
 - 或者：用“全部 OK” (3) 来对所有信息提示进行确认
3. 再触摸一下最上边的一条信息提示或按屏幕左侧边缘上的“X”将重新关闭信息提示列表。

2.2 选择并设置运行方式

KUKA 机器人的运行方式

- T1 (手动慢速运行)
 - 用于测试运行、编程和示教
 - 程序执行时的最大速度为 250 mm/s
 - 手动运行时的最大速度为 250 mm/s
- T2 (手动快速运行)
 - 用于测试运行
 - 程序执行时的速度等于编程设定的速度！
 - 手动运行：无法进行
- AUT (自动运行)
 - 用于不带上级控制系统的工业机器人
 - 程序执行时的速度等于编程设定的速度！
 - 手动运行：无法进行
- AUT EXT (外部自动运行)
 - 用于带上级控制系统 (PLC) 工业机器人
 - 程序执行时的速度等于编程设定的速度！
 - 手动运行：无法进行

运行方式的安全提示

手动运行 T1 和 T2

手动运行用于调试工作。调试工作是指所有为使机器人系统上可进入自动运行模式而必须在其上所执行的工作，其中包括：

- 示教 / 编程
- 在点动运行模式下执行程序 (测试 / 检验)

对新的或者经过更改的程序必须始终先在手动慢速运行方式 (T1) 下进行测试。

在手动慢速运行方式 (T1) 下：

- 操作人员防护装置（防护门）未激活！
- 在不必要的情况下，不允许其他人员在防护装置隔离的区域内停留。
如果有多个工作人员在防护装置隔离的区域内停留，则必须注意以下事项：
 - 所有人员必须能够不受妨碍地看到机器人系统。
 - 必须保证所有人员之间都可以直接看到对方。
- 操作人员必须选定一个合适的操作位置，使其可以看到危险区域并避开危险。

在手动快速运行方式下 (T2)：

- 操作人员防护装置（防护门）未激活！
- 只有在必须以大于手动慢速运行的速度进行测试时，才允许使用此运行方式。
- 在这种运行模式下不得进行示教。
- 在测试前，操作人员必须确保确认装置的功能完好。
- 操作人员的操作位置必须处于危险区域之外。
- 不允许其他人员在防护装置隔离的区域内停留。

运行方式自动和外部自动

- 必须配备安全、防护装置，而且它们的功能必须正常。
- 所有人员应位于由防护装置隔离的区域之外。

操作步骤



如果在运行过程中改变运行方式，驱动装置即立刻关断。工业机器人以安全停止 2 停机。

1. 在 KCP 上转动用于连接管理器的开关。连接管理器随即显示。



2. 选择运行方式。



3. 将用于连接管理器的开关再次转回初始位置。

所选的运行方式会显示在 smartPAD 的状态栏中。



2.3 单独运动机器人的各轴

说明：按轴坐标的运动

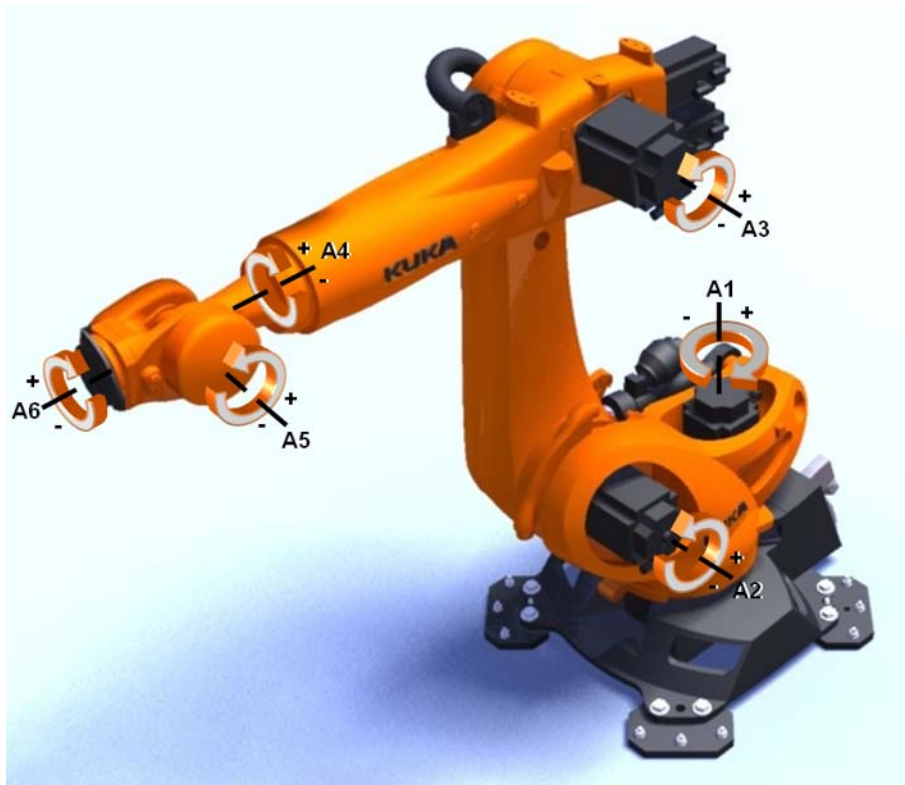


图 2-3: 库卡机器人自由度

机器人轴的运动

- 每根轴逐个沿正向和负向
- 为此需要使用移动键或者 KUKA smartPAD 的 3D 鼠标。
- 速度可以更改（手动倍率：HOV）
- 仅在 T1 运行模式下才能手动移动。
- 确认键必须已经按下。

原理

通过按确认键激活驱动装置。只要一按移动键或 3D 鼠标，机器人轴的调节装置便启动，机器人执行所需的运动。

运动可是连续的，也可能是增量式的。为此要在状态栏中选择增量值。

以下信息提示对手动运行有影响：

信息提示	原因	补救措施
“激活的指令被禁”	出现停机 (STOP) 讯息或引起激活的指令被禁的状态。（例如：按下了紧急停止按钮或驱动装置尚未就绪）	解锁紧急停止按钮并且 / 或者在信息窗口中确认信息提示。按了确认键后可立即使用驱动装置。
“软件限位开关 -A5”	以给定的方向（+ 或 -）移到所显示轴（例如 A5）的软件限位开关。	将显示的轴朝相反方向移动。

按轴坐标的手动移动的安全提示

运行方式

机器人只允许在运行方式 T1（手动降低的速度）下手动运行。手动移动速度在 T1 运行方式下最高为 250 mm/s。运行方式可通过连接管理器进行设置。

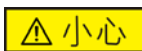
确认开关

为了能绕机器人移动，必须按下一个确认开关。smartPAD 上装有三个确认开关。确认开关具有 3 个挡位：

- 未按下
- 中位
- 完全按下（警报位置）

软件限位开关

即使采用与轴相关的手动移动，机器人的移动也受到软件限位开关的最大正、负值的限制。



如果在信息窗口中出现信息“执行零点标定”，则也可超过这两个极限值移动。但这可能会损坏机器人系统！

操作步骤：执行按轴坐标的运动

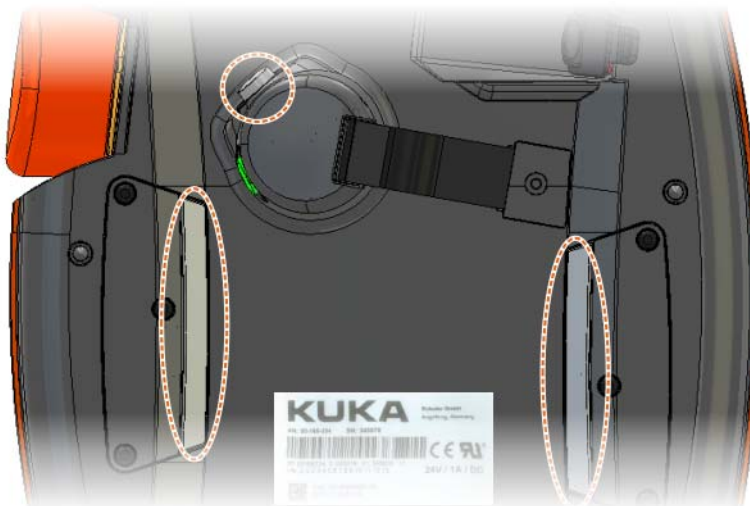
1. 选择轴作为移动键的选项。



2. 设置手动倍率。



3. 将确认开关按至中间挡位并按住。



在移动键旁边即显示轴 A1 至 A6。

4. 按下正或负移动键，以使轴朝正方向或反方向运动。



在紧急情况下脱离
控制系统移动机器人



图 2-4: 自由旋转装置

说明

发生事故或故障后，可借助自由旋转装置移动机器人。自由旋转装置可用于基轴驱动电机，视机器人类型而定也可用于手动轴驱动电机。该装置只允许用于特殊情况或紧急情况，例如用于解救人员。如使用了自由旋转装置，则必须在此后更换相关的电机。



警告！

运行期间，电机将达到的温度可能导致皮肤烫伤。避免与其接触。请务必采取适宜的安全防护措施，例如佩戴防护手套。

操作步骤

1. 关断机器人控制系统，并做好保护（例如用挂锁锁住），防止未经许可的意外重启。
2. 拆下电机上的防护盖。
3. 将自由旋转装置置于相应的电机上，并将轴朝所希望的方向运动。

可作为选项订购在电机上用箭头标记方向的标签。须克服电机机械制动器的阻力，且必要时还须克服额外的轴负载。

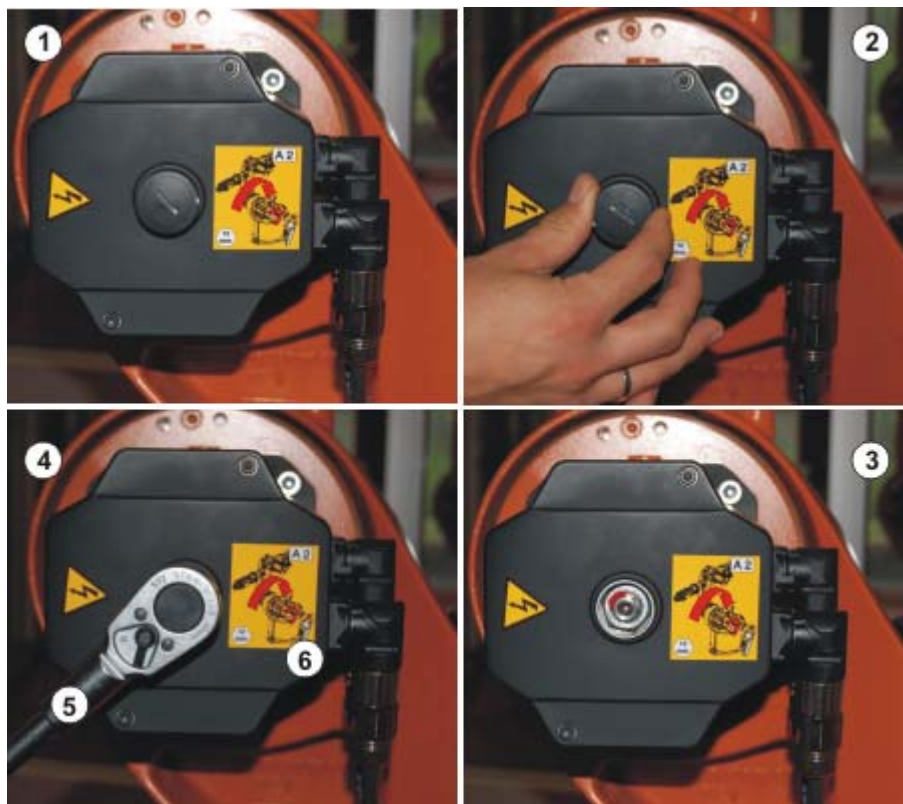


图 2-5: 使用自由旋转装置的操作步骤

序号	说明
1	防护盖盖上的电机 A2
2	打开电机 A2 的防护盖
3	防护盖已拆下的电机 A2
4	将自由旋转装置装到电机 A2 上
5	自由旋转装置
6	有转向说明的标签（选项）



警告！

在使用自由旋转装置移动轴时，可能会损坏电机制动器。可能会导致人员受伤及设备损坏。在使用自由旋转装置后必须更换相应的电机。



详细信息请见库卡机器人的操作和安装指南。

2.4 与机器人相关的坐标系

在工业机器人的操作、编程和投入运行时坐标系具有重要的意义。在机器人控制系统中定义了下列坐标系：

- WORLD | 世界坐标系
- ROBROOT | 机器人足部坐标系
- BASE | 基坐标系
- FLANGE | 法兰坐标系
- TOOL | 工具坐标系

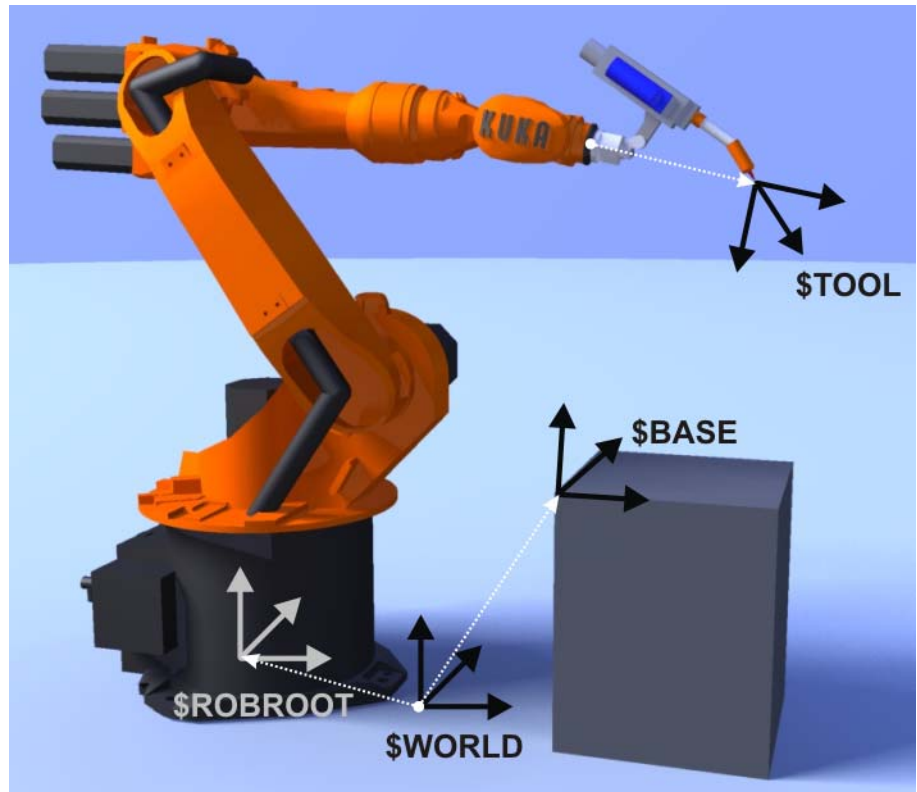


图 2-6: KUKA 机器人上的坐标系

名称	位置	应用	特点
WORLD	可自由定义	ROBROOT 和 BASE 的原点	大多数情况下位于机器人足部。
ROBROOT	固定于机器人足内	机器人的原点	说明机器人在世界坐标系中的位置。
BASE	可自由定义	工件，工装	说明基坐标在世界坐标系中的位置。
FLANGE	固定于机器人法兰上	TOOL 的原点	原点为机器人法兰中心。
TOOL	可自由定义	工具	TOOL 坐标系的原点被称为“TCP”。 (TCP = Tool Center Point , 即工具中心点)。

2.5 机器人在世界坐标系中运动

世界坐标系中的运动

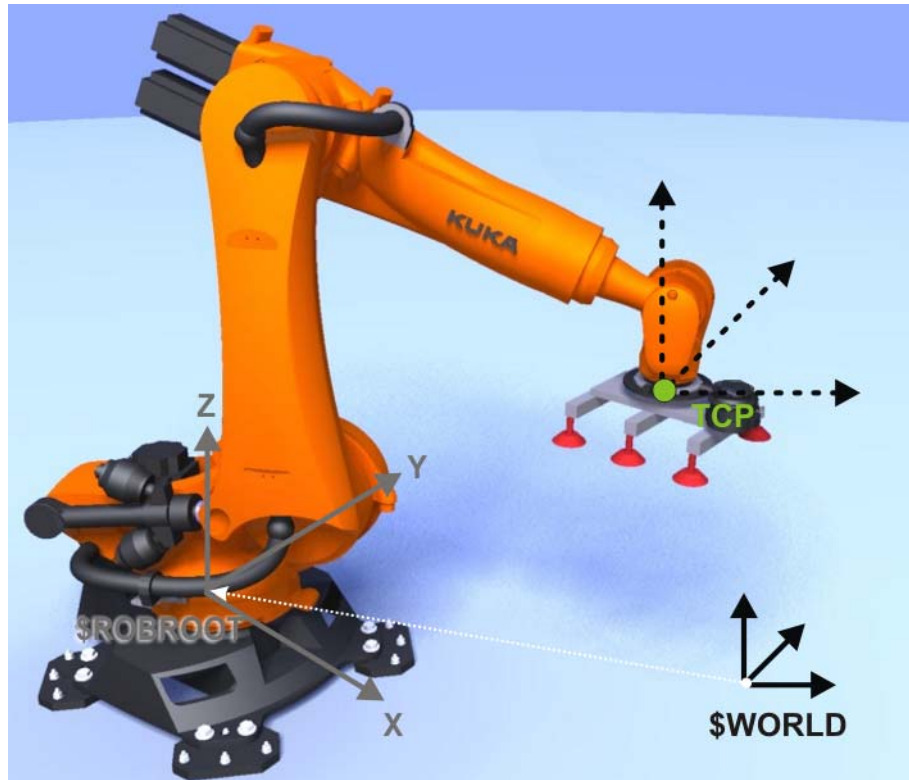


图 2-7: 手动移动世界坐标系的原则

- 机器人工具可以根据世界坐标系的坐标方向运动。
在此过程中，**所有的**机器人轴也会移动。
- 为此需要使用移动键或者 KUKA smartPAD 的 3D 鼠标。
- 在标准设置下，世界坐标系位于机器人底座 (Robroot) 中
- 速度可以更改（手动倍率：HOV）
- 仅在 T1 运行模式下才能手动移动。
- 确认键必须已经按下。

3D 鼠标

- 通过 3D 鼠标可以使机器人的运动变得直观明了，因此是在世界坐标系中进行手动移动时的不二之选。
- 鼠标位置和自由度两者均可更改。

在世界坐标系中的手动移动原理

在坐标系中可以两种不同的方式移动机器人：

- 沿坐标系的坐标轴方向平移（直线）：X、Y、Z
- 环绕着坐标系的坐标轴方向转动（旋转 / 回转）：角度 A、B 和 C

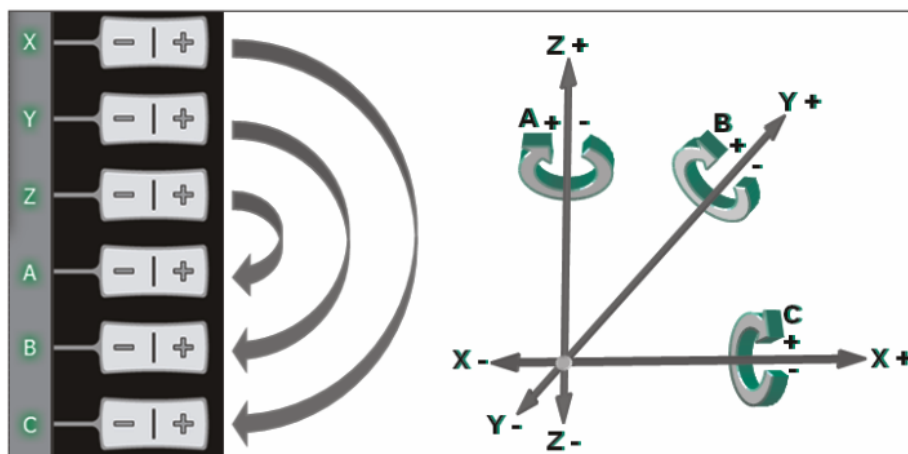


图 2-8: 笛卡尔坐标系

收到一个运行指令时（例如按了移动键后）控制器先计算一行程段。该行程段的起点是工具参照点 (TCP)。行程段的方向由世界坐标系给定。控制器控制所有轴相应运动，使工具沿该行程段运动（平动）或绕其旋转（转动）。

使用世界坐标系的优点：

- 机器人的动作始终可预测。
- 动作始终是惟一的，因为原点和坐标方向始终是已知的。
- 对于经过零点标定的机器人始终可用世界坐标系。
- 可用 3D 鼠标直观操作。

使用 3D 鼠标

- 可将 3D 鼠标用于所有运动方式：
- 平移：按住并拖动 3D 鼠标

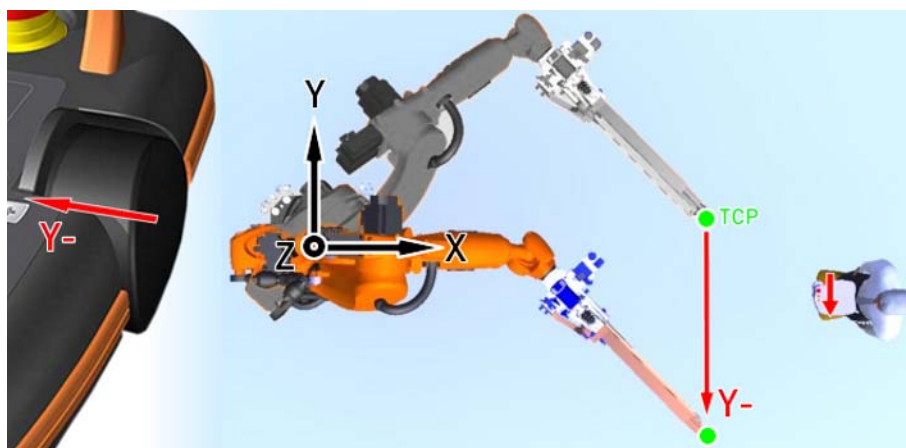


图 2-9: 示例：向左运动

- 转动：转动并摆动 3D 鼠标

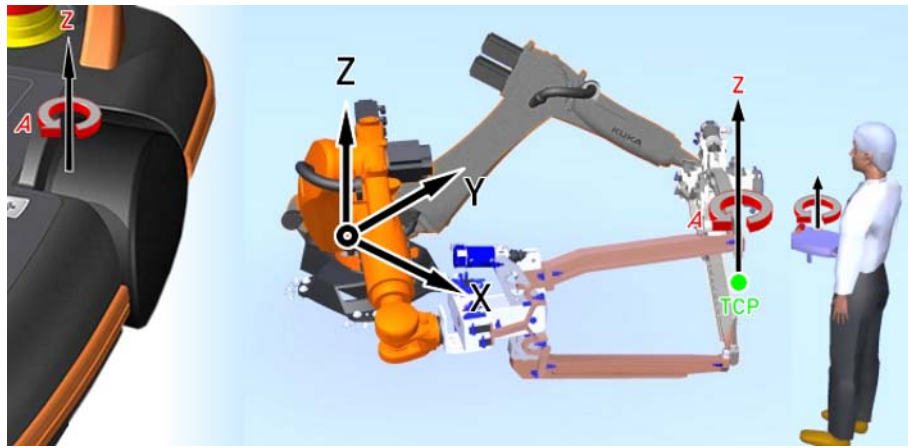


图 2-10: 示例：绕 Z 轴的旋转运动：转角 A

- 3D 鼠标的位置可根据人 - 机器人的位置进行相应调整。

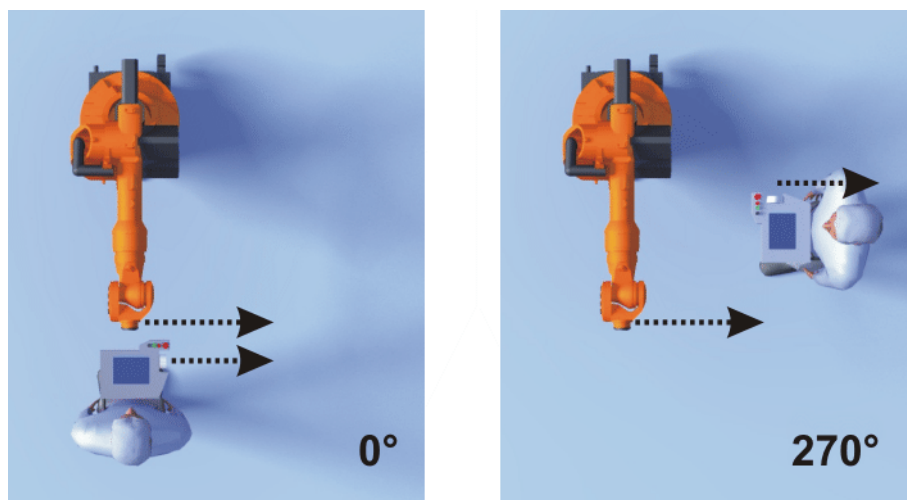


图 2-11: 3D 鼠标：0° 和 270°

执行平移（世界坐标系）

1. 通过移动滑动调节器 (1) 来调节 KCP 的位置



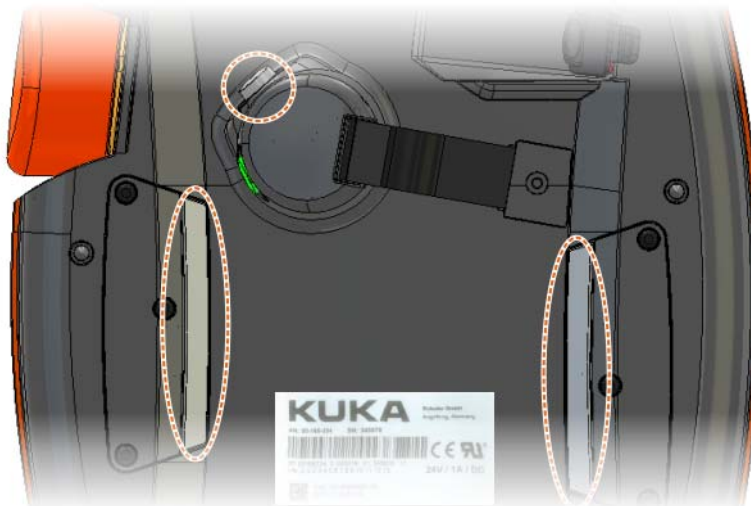
2. 选择世界坐标系作为 3D 鼠标的选项。



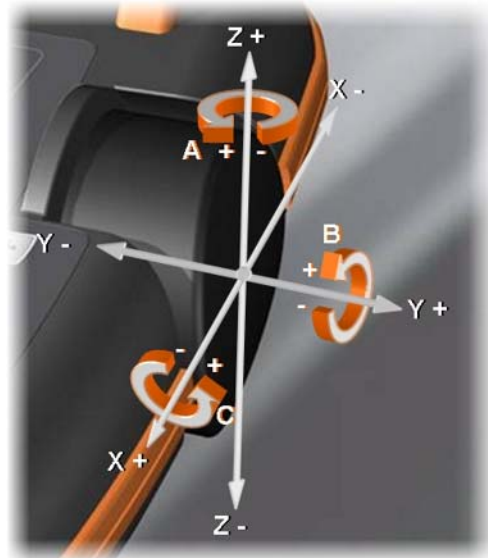
3. 设置手动倍率。



4. 将确认开关按至中间挡位并按住。



5. 用 3D 鼠标将机器人朝所需方向移动



6. 此外也可使用移动键：



2.6 在工具坐标系中移动机器人

在工具坐标系中手
动移动

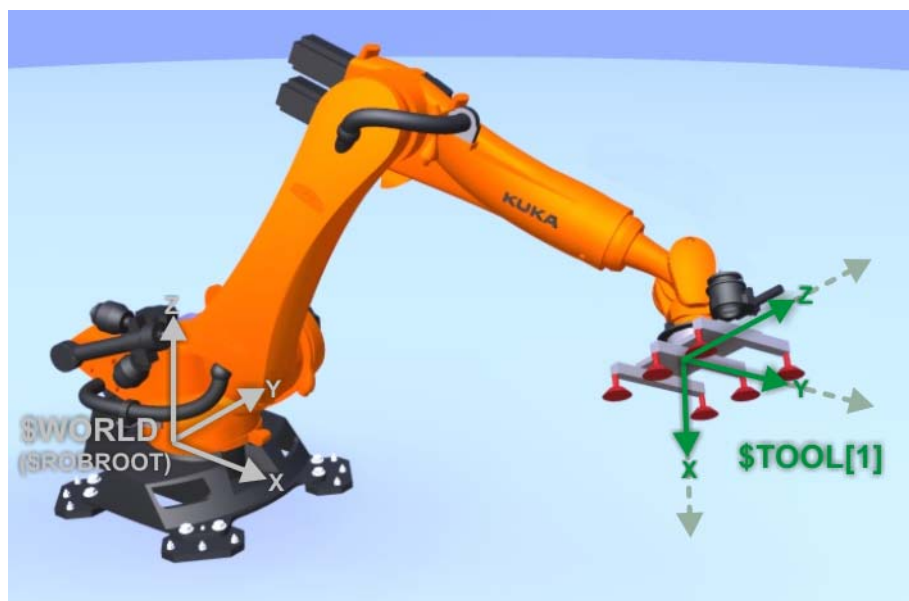
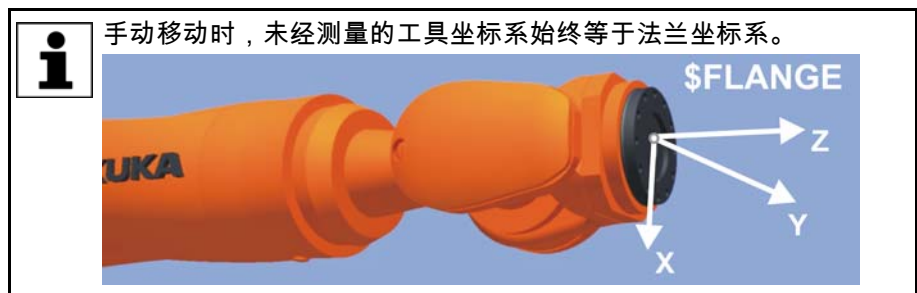


图 2-12: 机器人工具坐标系

- 在工具坐标系中手动移动时，可根据之前所测工具的坐标方向移动机器人。
因此，坐标系并非固定不变（例如：世界坐标系或基坐标系），而是由机器人引导。
在此过程中，所有需要的机器人轴也会自行移动。哪些轴会自行移动由系统决定，并因运动情况不同而异。
工具坐标系的原点被称为 TCP，并与工具的工作点对应。
- 为此需要使用移动键或者 KUKA smartPAD 的 3D 鼠标。
- 可供选择的工具坐标系有 16 个。
- 速度可以更改（手动倍率：HOV）
- 仅在 T1 运行模式下才能手动移动。
- 确认键必须已经按下。



工具坐标时手动移动的原则

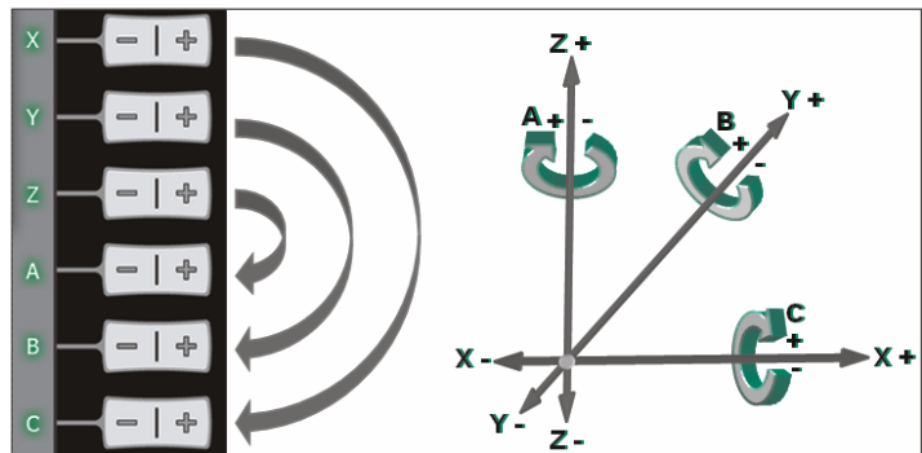


图 2-13: 笛卡尔坐标系

在坐标系中可以两种不同的方式移动机器人：

- 沿坐标系的坐标轴方向平移（直线）：X、Y、Z
- 环绕着坐标系的坐标轴方向转动（旋转 / 回转）：角度 A、B 和 C

使用工具坐标系的优点：

- 要工具坐标系已知，机器人的运动始终可预测。
- 可以沿工具作业方向移动或者绕 TCP 调整姿态。

*工具作业方向*是指工具的工作方向或者工序方向：粘胶喷嘴的粘结剂喷出方向，抓取部件时的抓取方向等。

操作步骤

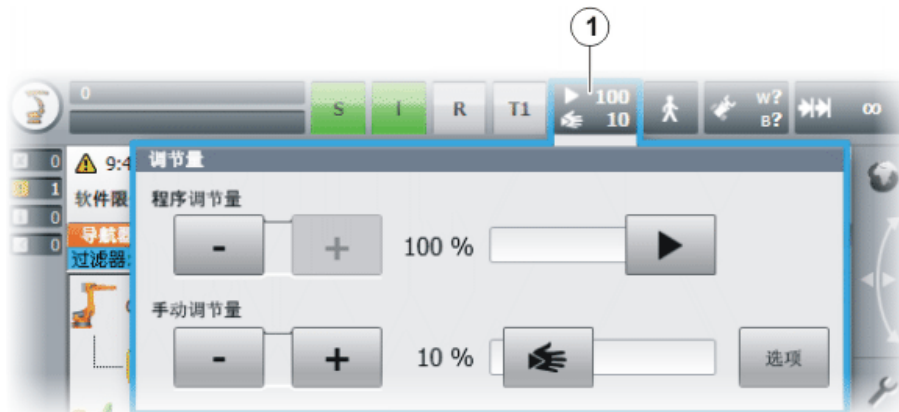
1. 选择工具作为所用的坐标系。



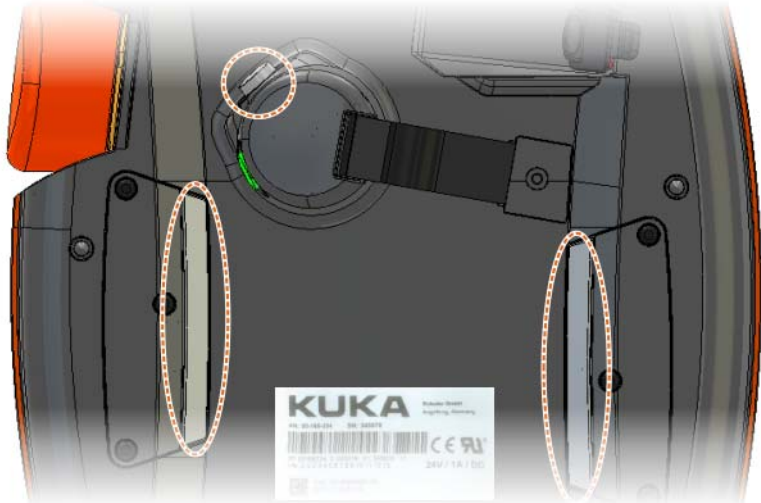
2. 选择工具编号



3. 设定手动倍率。



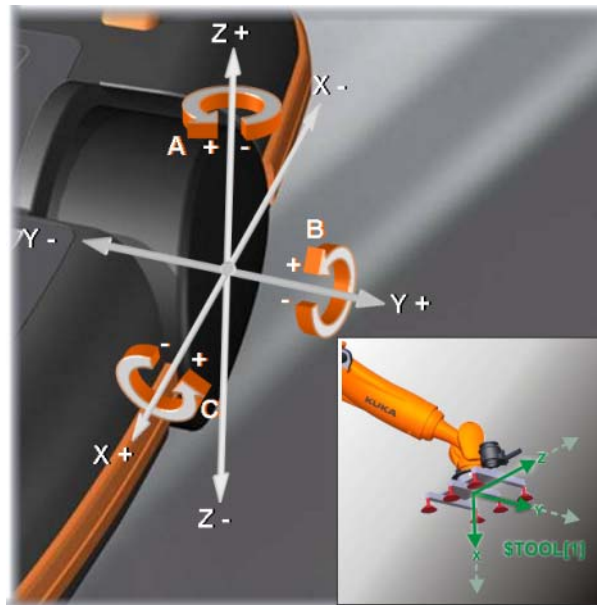
- 按下确认开关的中间位置并保持按住。



- 用移动键移动机器人。



- 或者：用 3D 鼠标将机器人朝所需方向移动。



2.7 在基坐标系中移动机器人

在基坐标系中运动

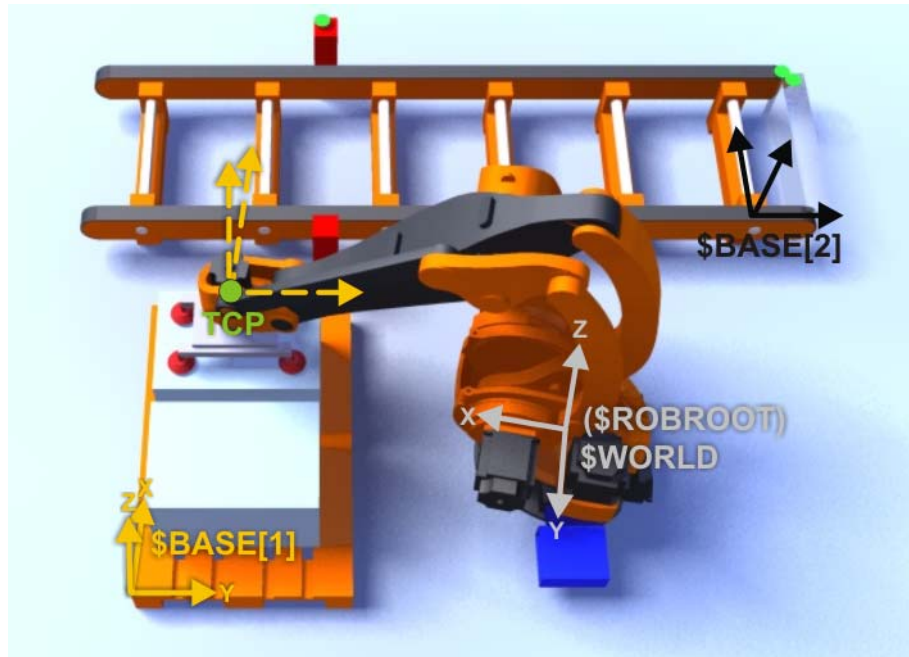


图 2-14: 基坐标系中的手动移动

基坐标系说明

- 机器人的工具可以根据基坐标系的坐标方向运动。基坐标系可以被单个测量，并可以经常沿工件边缘、工件支座或者货盘调整姿态。由此可以进行舒适的手动移动！
在此过程中，所有需要的机器人轴也会自行移动。哪些轴会自行移动由系统决定，并因运动情况不同而异。
- 为此需要使用移动键或者 KUKA smartPAD 的 3D 鼠标。
- 可供选择的基坐标系有 32 个。
- 速度可以更改（手动倍率：HOV）
- 仅在 T1 运行模式下才能手动移动。
- 确认键必须已经按下。

基坐标手动移动原理

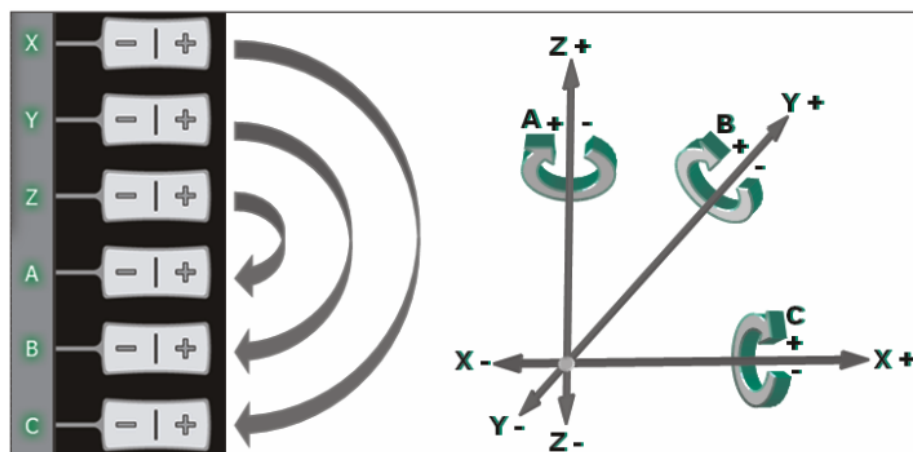


图 2-15: 笛卡尔坐标系

在坐标系中可以两种不同的方式移动机器人：

- 沿坐标系的坐标轴方向平移（直线）：X、Y、Z
- 环绕着坐标系的坐标轴方向转动（旋转 / 回转）：角度 A、B 和 C

收到一个运行指令时（例如按了移动键后）控制器先计算一行程段。该行程段的起点是工具参照点 (TCP)。行程段的方向由世界坐标系给定。控制器控制所有轴相应运动，使工具沿该行程段运动（平动）或绕其旋转（转动）。

使用基坐标系的优点：

- 只要基坐标系已知，机器人的动作始终可预测。
- 这里也可用 3D 鼠标直观操作。前提条件是操作员必须相对机器人以及基坐标系正确站立。

注意

如果还另外设定了工具坐标系，则可在基坐标系中绕 TCP 改变姿态。

操作步骤

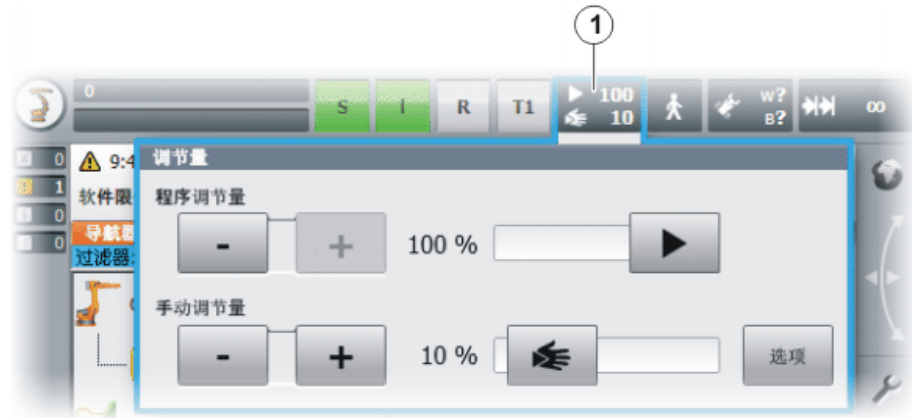
1. 选择**基坐标**作为移动键的选项。



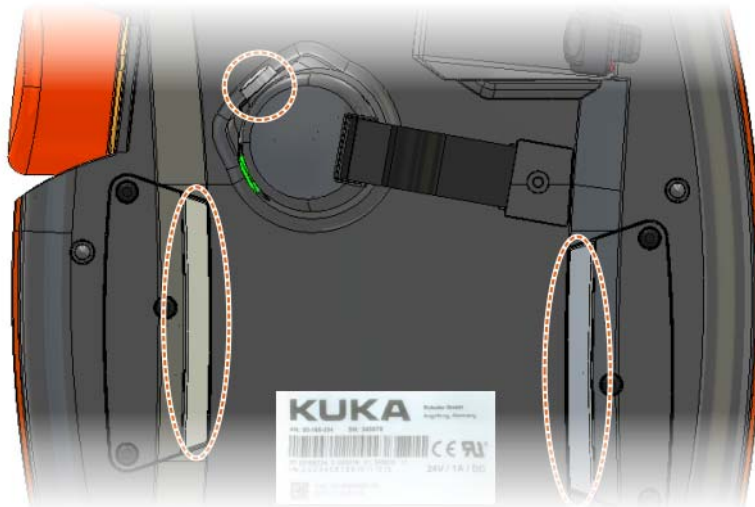
2. 选择工具坐标和基坐标



3. 设置手动倍率。



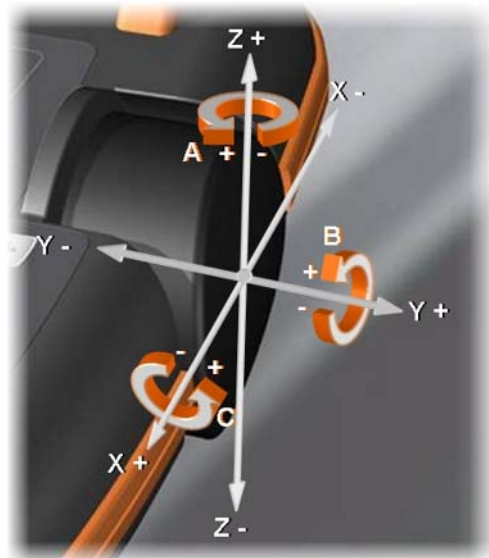
4. 将确认开关按至中间挡位并按住。



5. 用移动键沿所需的方向移动。



6. 作为选项，也可用 3D 鼠标来移动。



停机反应

工业机器人会在操作或在监控和出现故障信息时做出停机反应。下面的表格中列出了停机反应与所设定的运行方式的关系。

概念	说明
安全运行停止	<p>安全运行停止是一种停机监控。它不停止机器人运动，而是监控机器人轴是否静止。如果机器人轴在安全运行停止时运动，则安全运行停止触发安全停止 STOP 0。</p> <p>安全运行停止也可由外部触发。</p> <p>如果安全运行停止被触发，则机器人控制系统会给现场总线的的一个输出端赋值。如果在触发安全运行停止时不是所有的轴都停止，并以此触发了安全停止 STOP 0，则也会给该输出端赋值。</p>
安全停止 STOP 0	<p>一种由安全控制系统触发并执行的停止。安全控制系统立即关断驱动装置和制动器的供电电源。</p> <p>提示：该停止在文件中称作安全停止 0。</p>
安全停止 STOP 1	<p>一种由安全控制系统触发并监控的停止。该制动过程由机器人控制系统中与安全无关的部件执行并由安全控制系统监控。一旦机械手静止下来，安全控制系统就关断驱动装置和制动器的供电电源。</p> <p>如果安全停止 STOP 1 被触发，则机器人控制系统便给现场总线的的一个输出端赋值。</p> <p>安全停止 STOP 1 也可由外部触发。</p> <p>提示：该停止在文件中称作安全停止 1。</p>
安全停止 STOP 2	<p>一种由安全控制系统触发并监控的停止。该制动过程由机器人控制系统中与安全无关的部件执行并由安全控制系统监控。驱动装置保持接通状态，制动器则保持松开状态。一旦机械手停止下来，安全运行停止即被触发。</p> <p>如果安全停止 STOP 2 被触发，则机器人控制系统便给现场总线的的一个输出端赋值。</p> <p>安全停止 STOP 2 也可由外部触发。</p> <p>提示：该停止在文件中称作安全停止 2。</p>
停机类别 0	<p>驱动装置立即关断，制动器制动。机械手和附加轴（选项）在额定位置附近制动。</p> <p>提示：此停机类别在文件中被称为 STOP 0。</p>

概念	说明
停机类别 1	机械手和附加轴（选项）在额定位置上制动。1 秒钟后驱动装置关断，制动器制动。 提示： 此停机类别在文件中被称为 STOP 1。
停机类别 2	驱动装置不被关断，制动器不制动。机械手及附加轴（选项）通过一个不偏离额定位置的制动斜坡进行制动。 提示： 此停机类别在文件中被称为 STOP 2。

触发因素	T1,T2	AUT , AUT EXT
启动键被松开	STOP 2	-
按下停机键	STOP 2	
驱动装置关机	STOP 1	
输入端无“运动许可”	STOP 2	
关闭机器人控制系统（断电）	STOP 0	
机器人控制系统内与安全无关的部件出现内部故障	STOP 0 或 STOP 1 (取决于故障原因)	
运行期间工作模式被切换	安全停止 2	
打开防护门（操作人员防护装置）	-	安全停止 1
松开确认键	安全停止 2	-
持续按住确认键或出现故障	安全停止 1	-
按下急停按钮	安全停止 1	
安全控制系统或安全控制系统外围设备中的故障	安全停止 0	

2.8 练习：操作及手动移动

练习目的

成功完成此练习后，您可执行下列操作：

- 接通以及关闭机器人控制系统
- 用 KCP（库卡控制面板）进行机器人的基本操作
- 用移动键和 3D 鼠标在世界坐标系中按轴坐标地手动移动机器人
- 理解最初的简单系统信息并排除故障

前提条件

为成功完成此练习，必须满足以下前提条件：

- 参加安全培训



提示！

开始练习前必须先参加安全培训并有记录！

- 具备常规操作库卡工业机器人系统的理论知识
- 具备按轴坐标的手动移动和在世界坐标系中移动的理论知识

练习内容

请完成以下任务：

1. 接通控制柜，等待启动阶段结束
2. 将紧急停止按钮解锁并确认
3. 确保设置了运行方式 T1
4. 激活按轴坐标的手动移动

5. 用手动移动键和 3D 鼠标以不同的手动倍率 (HOV) 设置按轴坐标地手动移动机器人
6. 了解各轴的移动范围，注意是否有障碍物，例如：工作台或装有固定工具的方形工具库（可达性检查）
7. 在达到软件限位开关时请注意信息窗口
8. 请从不同方向按轴坐标将工具（抓爪）移动到参照工具（黑色金属尖）处
9. 在世界坐标系中重复此过程

与练习配套的问题

1. 如何确认信息提示？

.....
.....

2. 哪个图标代表世界坐标系？

a)



b)



c)



d)



3. 手动移动的速度设置叫什么？

.....

4. 有哪些运行方式？

.....
.....

2.9 用一个固定工具进行手动移动

优点和应用领域 某些生产和加工过程要求机器人操作工件而不是工具。优点是，部件无需先放置好便能加工，因此可节省夹紧工装。例如，这适用于以下情况：

- 粘接
- 焊接
- 等等

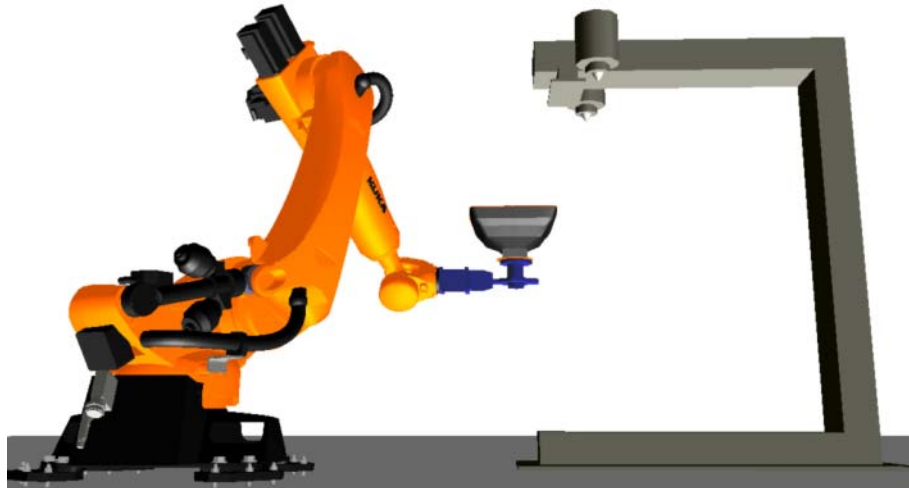


图 2-16: 固定工具示例

注意

为了为此类应用成功编程，既要测量固定工具的外部 TCP，也要测量工件。

固定工具时更改过的运动过程

虽然工具是固定（不运动）对象，但是工具还是有一个所属坐标系的工具参照点。此时该参照点被称为**外部 TCP**。由于这是一个不运动的坐标系，所以数据可以如同基坐标系一样进行管理，并可以作为**基坐标**储存！

（运动着）的工件则又可以作为**工具坐标**储存。由此，可以相对于 TCP 沿着工件边缘进行移动！

注意

需要注意，在用固定工具手动移动时，运动均相对外部 TCP！

用固定工具手动移动的操作步骤



图 2-17: 在选项菜单中选择外部 TCP

1. 在工具选择窗口中选择由机器人导引的工件
2. 在基坐标选择窗口中选择固定工具
3. 将 IpoMode（Ipo 模式）选择设为外部工具
4. 作为移动键 /3D 鼠标选项设定工具：
 - 设定工具，以便在工件坐标系中移动

- 设定基坐标，以便在外部工具坐标系中移动
5. 设定手动倍率
 6. 按下确认开关的中间位置并保持按住。
 7. 用移动键 /3D 鼠标朝所需方向移动。

通过在**手动移动选项**选项窗口中选择**外部工具**控制器切换：所有运动现在均相对外部 TCP，而不是由机器人导引的工具。

2.10 练习：用固定的工具练习手动移动

练习目的	成功完成此练习后，您可执行下列操作： <ul style="list-style-type: none">■ 参照固定工具手动移动一个由机器人引导的部件
前提条件	为成功完成此练习，必须满足以下前提条件： <ul style="list-style-type: none">■ 具备常规操作库卡工业机器人系统的理论知识■ 具备用外部工具进行移动的理论知识
练习内容	<ol style="list-style-type: none">1. 设定工具坐标系“标牌”2. 设定基坐标系“外部尖触头”3. 在选项菜单手动移动选项中设为“外部工具”4. 将标牌移到外部尖触头处5. 移动标牌并使其对准外部尖触头在此过程中测试工具坐标与基坐标之间的区别6. 在选项菜单手动移动选项中设定“法兰”7. 移动标牌并使其对准外部尖触头

3 机器人的投入运行

3.1 零点标定的原理

为什么要标定零点？

仅在工业机器人得到充分和正确标定零点时，它的使用效果才会最好。因为只有这样，机器人才能达到它最高的点精度和轨迹精度或者完全能够以编程设定的动作运动。



零点标定时，会给每个机器人轴分派一个基准值。

完整的零点标定过程包括为每一个轴标定零点。通过技术辅助工具 (EMD = Electronic Mastering Device (电子控制仪)) 可为任何一个在**机械零点位置**的轴指定一个基准值 (例如: 0°)。因为这样就可以使轴的机械位置和电气位置保持一致，所以每一个轴都有一个唯一的角度值。

所有机器人的零点标定位置校准都，但不完全相同。精确位置在同一机器人型号的不同机器人之间也会有所不同。

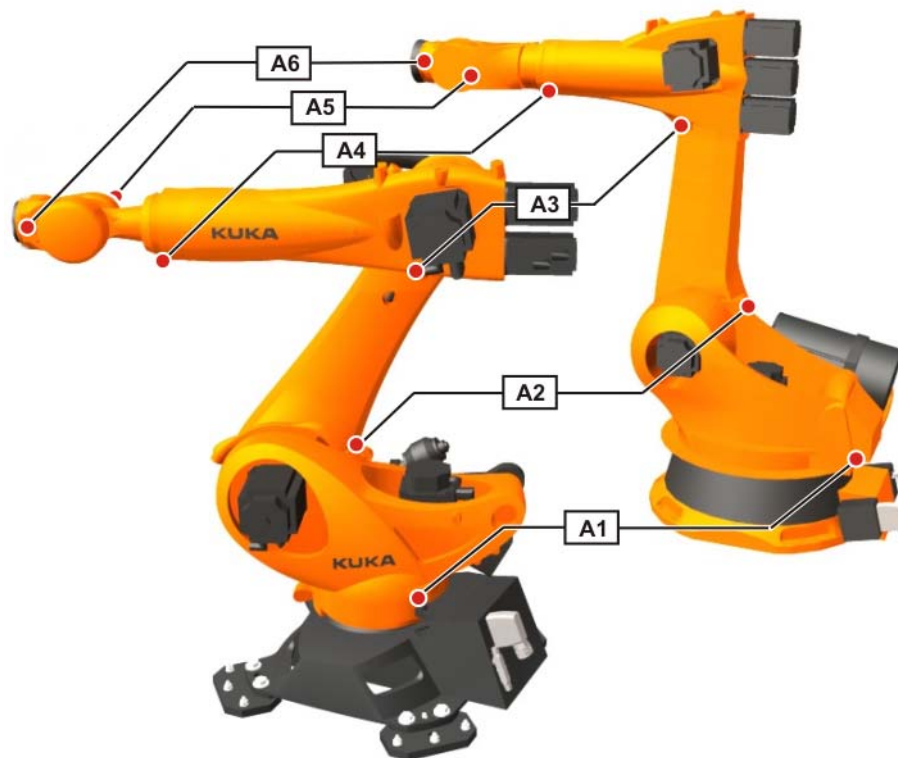


图 3-1: 零点标定套筒的位置

机械零点位置的角度值 (= 基准值)

轴	“Quantec” 代机器人	其它机器人型号 (例如: 2000、KR 16 系列等等)
A1	-20°	0°
A2	-120°	-90°
A3	$+120^\circ$	$+90^\circ$
A4	0°	0°
A5	0°	0°
A6	0°	0°

何时标定零点？

原则上，机器人必须时刻处于已标定零点状态。在以下情况下必须进行零点标定：

- 在投入运行时
- 在对参与定位值感测的部件（例如带分解器或 RDC 的电机）采取了维护措施之后
- 当未用控制器移动了机器人轴（例如借助于自由旋转装置）时
- 进行了机械修理后 / 问题是必须先删除机器人的零点，然后才可标定零点：
 - 更换齿轮箱后。
 - 以高于 250 mm/s 的速度上行移至一个终端止挡之后
 - 在碰撞后

注意

在进行维护前一般应检查当前的零点标定。

关于零点标定的安全提示

如果机器人轴未经零点标定，则会严重限制机器人的功能：

- 无法编程运行：不能沿编程设定的点运行。
- 无法在手动运行模式下手动平移：不能在坐标系中移动。
- 软件限位开关关闭。

小心**警告！**

对于删除零点的机器人，软件限位开关是关闭的。机器人可能会驶向终端止挡上的缓冲器，由此可能使缓冲器受损，以至必须更换。尽可能不运行删除零点的机器人，或尽量减小手动倍率。

执行零点标定



图 3-2: 正在使用的 EMD

零点标定可通过确定轴的机械零点的方式进行。在此过程中轴将一直运动，直至达到机械零点为止。这种情况出现在探针到达测量槽最深点时。因此，每根轴都配有一个零点标定套筒和一个零点标定标记。

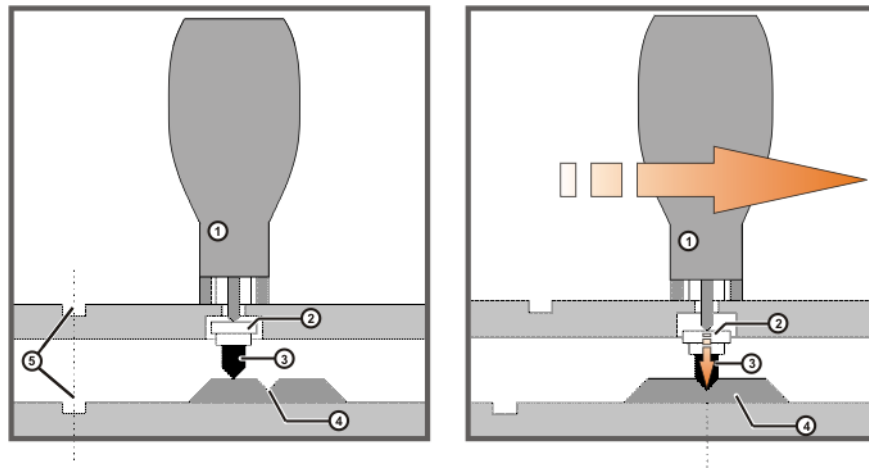


图 3-3: EMD 校准流程

- | | |
|---------------|-----------|
| 1 EMD (电子控制仪) | 4 测量槽 |
| 2 测量套筒 | 5 预零点标定标记 |
| 3 探针 | |

3.2 给机器人标定零点

机器人的零点标定
方式

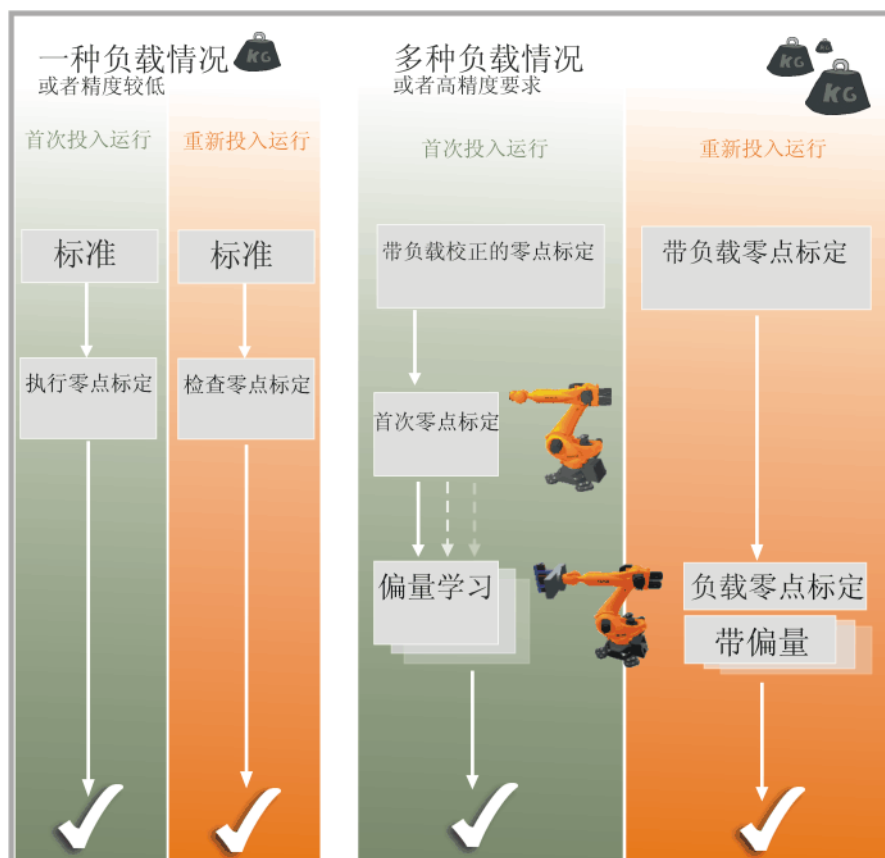


图 3-4: 零点标定途径

为何要学习偏量？

通过固定在法兰处的工具重量，机器人承受着静态载荷。由于部件和齿轮箱上材料固有的弹性，未承载的机器人与承载的机器人相比其位置上会有所区别。这些相当于几个增量的区别将影响到机器人的精确度。

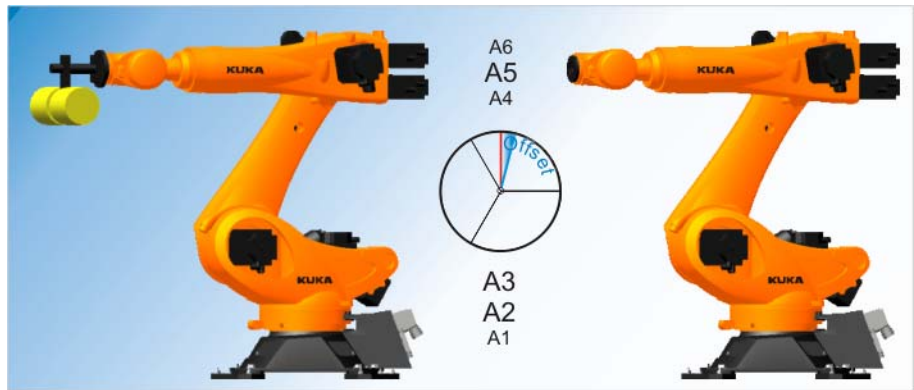


图 3-5: 偏量学习

“偏量学习”带负载进行。与首次零点标定（无负载）的差值被储存。

如果机器人以各种不同负载工作，则必须对每个负载都进行“偏量学习”。对于抓取沉重部件的抓爪来说，则必须对抓爪分别在不带构件和带构件时进行“偏量学习”。

注意 Mastery.logMastery.log 零点标定偏量值文件测定的偏移量保存在文件 **Mastery.log** 中。该文件位于硬盘的目录 C:\KRC\ROBOTER\LOG 下并含有特殊零点标定数据：

- 时间戳记（日期，时间）
- 轴
- 机器人的系列号
- 工具编号
- 用度表示的偏量值 (*Encoder Difference*)
- **Mastery.log 举例：**

```
Date: 22.03.11 Time: 10:07:10
Axis 1 Serialno.: 863334 Tool Teaching for Tool No 5
(Encoder Difference: -0.001209)
Date: 22.03.11 Time: 10:08:44
Axis 2 Serialno.: 863334 Tool Teaching for Tool No 5
Encoder Difference: 0.005954)
...
```

只有经带负载校正而标定零点的机器人具有所要求的高精确度。因此必须针对每种负荷情况进行偏量学习！前提条件是：工具的几何测量已完成，因此已分配了一个工具编号。

首次零点标定的操作步骤

注意 只有当机器人没有负载时才可以执行首次零点标定。不得安装工具和附加负载。

1. 将机器人移到预零点标定位置。



图 3-6: 预零点标定位置示例

2. 在主菜单中选择投入运行 > 零点标定 > EMD > 带负载校正 > 首次零点标定。

一个窗口自动打开。所有待零点标定的轴都显示出来。编号最小的轴已被选定。

3. 从窗口中选定的轴上取下测量筒的防护盖。（翻转过来的 EMD 可用作螺丝刀。）将 EMD 拧到测量筒上。



图 3-7: 已将 EMD 拧到测量筒上

然后将测量导线连到 EMD 上，并连接到机器人接线盒的接口 X32 上。



图 3-8: EMD 电缆，连接



注意！

始终将 EMD 不带测量导线拧到测量筒上。然后方可将测量导线接到 EMD 上。否则测量导线会被损坏。

同样在拆除 EMD 时也必须先拆下 EMD 的测量导线。然后将 EMD 从测量筒上拆下。

在零点标定之后，将测量导线从接口 X32 上取下。否则会出现干扰信号或导致损坏。

4. 点击**零点标定**。
5. 将确认开关按至中间挡位并按住，然后按下并按住启动键。



图 3-9: 确认键和启动键

如果 EMD 通过了测量切口的最低点，则已到达零点标定位置。机器人自动停止运行。数值被储存。该轴在窗口中消失。

6. 将测量导线从 EMD 上取下。然后从测量筒上取下 EMD，并将防护盖重新装好。
7. 对所有待零点标定的轴重复步骤 2 至 5。
8. 关闭窗口。
9. 将测量导线从接口 X32 上取下。

偏量学习操作步骤

进行带负载的“偏量学习”。与首次零点标定的差值被储存。

1. 将机器人置于预零点标定位置
2. 在主菜单中选择**投入运行 > 零点标定 > EMD > 带负载校正 > 偏量学习**。
3. 输入工具编号。用**工具 OK** 确认。
随即打开一个窗口。所有工具尚未学习的轴都显示出来。编号最小的轴已被选定。
4. 从窗口中选定的轴上取下测量筒的防护盖。将 EMD 拧到测量筒上。然后将测量导线连到 EMD 上，并连接到底座接线盒的接口 X32 上。
5. **按学习**。
6. 按确认开关和启动键。
当 EMD 识别到测量切口的最低点时，则已到达零点标定位置。机器人自动停止运行。随即打开一个窗口。该轴上与首次零点标定的偏差以增量和度的形式显示出来。
7. 用 **OK** 键确认 该轴在窗口中消失。
8. 将测量导线从 EMD 上取下。然后从测量筒上取下 EMD，并将防护盖重新装好。
9. 对所有待零点标定的轴重复步骤 3 至 7。
10. 将测量导线从接口 X32 上取下。
11. 用**关闭**来关闭窗口。

带偏量的负载零点标定检查 / 设置的操作步骤

带偏量的负载零点标定在有负载的情况下进行。计算首次零点标定量。

1. 将机器人移到预零点标定位置
2. 在主菜单中选择**投入运行 > 零点标定 > EMD > 带负载校正 > 负载零点标定 > 带偏量**。
3. 输入工具编号。用**工具 OK** 确认。
4. 取下接口 X32 上的盖子，然后将测量导线接上。
5. 从窗口中选定的轴上取下测量筒的防护盖。（翻转过来的 EMD 可用作螺丝刀。）
6. 将 EMD 拧到测量筒上。
7. 将测量导线接到 EMD 上。在此过程中，将插头的红点对准 EMD 内的槽口。
8. 按下**检查**。
9. 按住确认开关并按下启动键。
10. 需要时，使用“保存”来储存这些数值。旧的零点标定值因而被删除。如果要恢复丢失的首次零点标定，必须保存这些数值。
11. 将测量导线从 EMD 上取下。然后从测量筒上取下 EMD，并将防护盖重新装好。
12. 对所有待零点标定的轴重复步骤 4 至 10。
13. 关闭窗口。
14. 将测量导线从接口 X32 上取下。

3.3 练习：机器人零点标定

练习目的 成功完成此练习后，您可执行下列操作：

- 驶至预零点标定位置
- 选择正确的零点标定方式
- 使用“电子控制仪”(EMD)
- 借助 EMD 给所有轴标定零点

前提条件 为成功完成此练习，必须满足以下前提条件：

- 具有有关一般零点标定过程的理论知识
- 具有有关预零点标定位置的理论知识



1 轴不在预零点标定位置



2 轴在预零点标定位置



- 将 EMD 正确连接至机器人
- 通过投入运行菜单进行零点标定

练习内容 请完成以下任务：

1. 删除所有机器人轴的零点
2. 将所有机器人轴按轴坐标方式移动到预零点标定位置
3. 通过 EMD 对所有轴进行带偏量的负载零点标定
4. 按轴坐标显示实际位置

与练习配套的问题

1. 零点标定目的是什么？

.....

2. 请给出机械零位时的所有 6 根轴的角度。

A1 : A2 :

A3 : A4 :

A5 : A6 :

3. 删除机器人零点时必须注意些什么？

.....

4. 哪些零点标定工具应优先使用？

.....
.....
5. 拧入 EMD (千分表) 后移动机器人会有哪些危险 ?
.....
.....

3.4 机器人上的负载

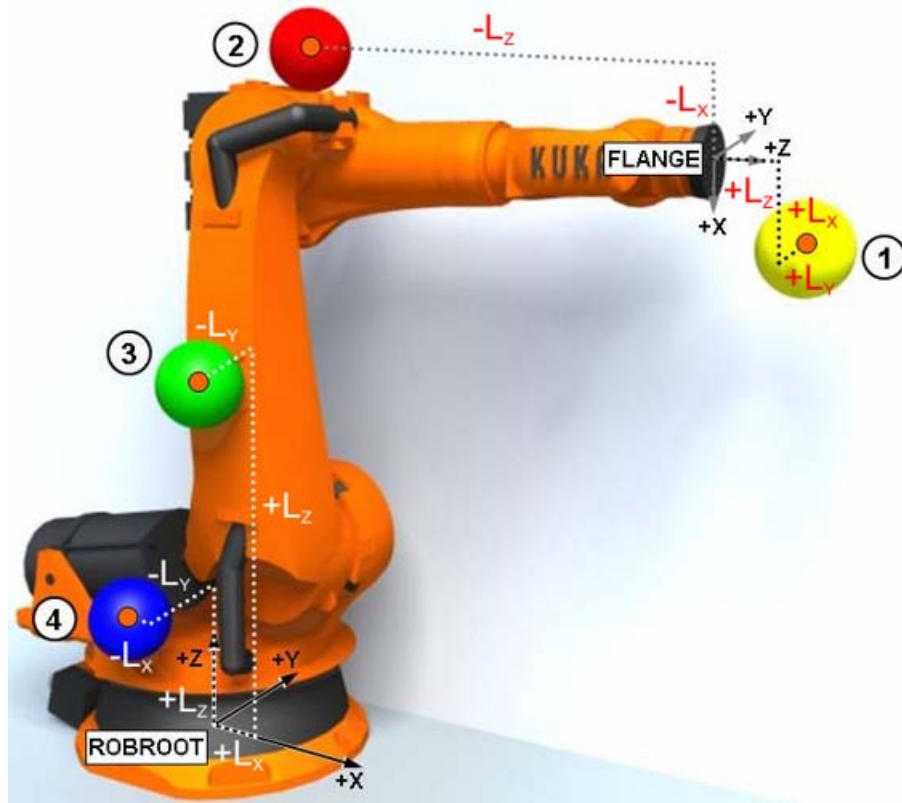


图 3-10: 机器人上的负载

- | | |
|-------------|-------------|
| 1 负荷 | 3 轴 2 的附加负载 |
| 2 轴 3 的附加负载 | 4 轴 1 的附加负载 |

3.4.1 工具负载数据

什么是工具负载数据？

工具负载数据是指所有装在机器人法兰上的负载。它是另外装在机器人上并由机器人一起移动的质量。

需要输入的值有质量、重心位置（质量受重力作用的点）、质量转动惯量以及所属的主惯性轴。

负载数据**必须**输入机器人控制系统，并分配给正确的工具。

例外：如果负载数据已经由 KUKA.LoadDataDetermination 传输到机器人控制系统中，则无需再手工输入。

工具负载数据的可能来源如下：

- KUKA.LoadDetect 软件选项（仅用于负载）
- 生产厂商数据
- 人工计算
- CAD 程序

负载数据的影响

输入的负载数据会影响许多控制过程。其中包括，例如：

- 控制算法（计算加速度）
- 速度和加速度监控
- 力矩监控
- 碰撞监控
- 能量监控

- 等等

所以，正确输入负载数据是非常重要的。如果机器人以正确输入的负载数据执行其运动，则

- 可以从它的高精度中受益
- 可以使运动过程具有最佳的节拍时间
- 可以使机器人达到长的使用寿命（由于磨损小）

操作步骤

1. 选择主菜单 **投入运行 > 测量 > 工具 > 工具负载数据**。
2. 在 **工具编号** 栏中输入工具的编号。用 **继续** 键确认。
3. 输入负载数据：
 - **M** 栏：质量
 - **X、Y、Z** 栏：相对于法兰的重心位置
 - **A、B、C** 栏：主惯性轴相对于法兰的取向
 - **JX、JY、JZ** 栏：惯性矩
(JX 是坐标系统 X 轴的惯性，该坐标系通过 A、B 和 C 相对于法兰转过一定角度。以此类推，JY 和 JZ 是指绕 Y 轴和 Z 轴的惯性。)
4. 用 **继续** 键确认。
5. 按下 **保存** 键。

3.4.2 机器人上的附加负载

机器人上的附加负载

附加负载是在基座、小臂或大臂上附加安装的部件，例如：

- 供能系统
- 阀门
- 上料系统
- 材料储备

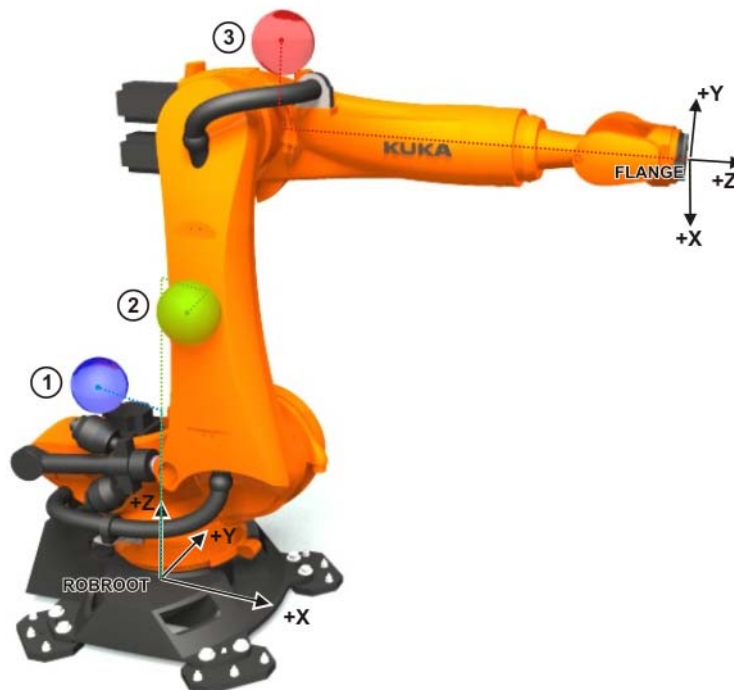


图 3-11: 机器人上的附加负载

附加负载数据必须输入机器人控制系统。必要的数据包括：

- 质量 (m)，单位：kg
- 物体重心至参照系 (X、Y 和 Z) 的距离，单位：mm

- 主惯性轴与参照系 (A、B 和 C) 的夹角，单位：度 (°)。
- 物体绕惯性轴 (Jx、Jy 和 Jz) 的转动惯量，单位：kgm²

每个附加负载的 X、Y、Z 值的参照系：

负载	参照系
附加负载 A1	ROBROOT 坐标系 A1 = 0°
附加负载 A2	ROBROOT 坐标系 A2 = -90°
附加负载 A3	法兰坐标系 A4 = 0°, A5 = 0°, A6 = 0°


附加负载数据的可能来源如下：

- 生产厂商数据
- 人工计算
- CAD 程序

附加负载对机器人运动的影响

负荷数据以不同的方式对机器人运动发生影响：

- 轨迹规划
- 加速度
- 节拍时间
- 磨损

 **警告** 如果用错误的负载数据或不适当的负载来运行机器人，则会导致人员受伤和生命危险并 / 或导致严重财产损失。

操作步骤

1. 选择主菜单 **投入运行 > 测量 > 附加负载数据**。
2. 输入其上将固定附加负荷的轴编号。用**继续**键确认。
3. 输入负荷数据。用**继续**键确认。
4. 按下**保存**键。


3.5 工具测量

说明

测量工具意味着生成一个以工具参照点为原点的坐标系。该参照点被称为 **TCP** (Tool Center Point, 即工具中心点), 该坐标系即为**工具坐标系**。

因此，工具测量包括

- TCP (坐标系原点) 的测量
- 坐标系姿态 / 朝向的测量

 **注意** 最多可储存 16 个工具坐标系。(变量 : TOOL_DATA[1...16])。

测量时，工具坐标系的原点到法兰坐标系的距离 (用 X、Y 和 Z) 以及之间的转角 (用角度 A、B 和 C) 被保存。

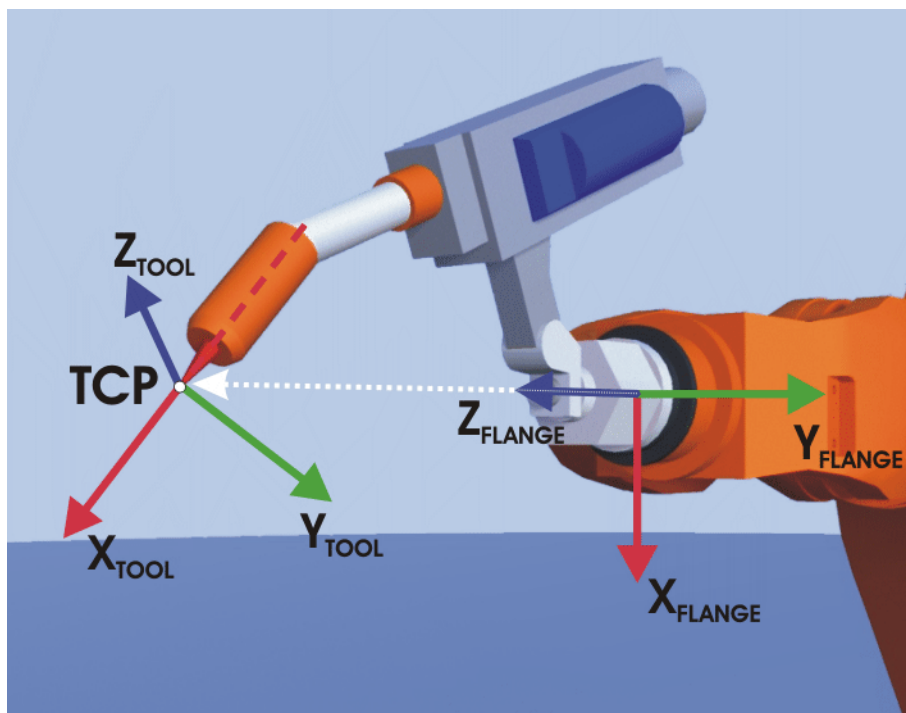


图 3-12: TCP 测量原理

优势

如果一个工具已精确测定，则在实践中对操作和编程人员有以下优点：

- 手动移动改善
 - 可围绕 TCP（例如：工具顶尖）改变姿态。

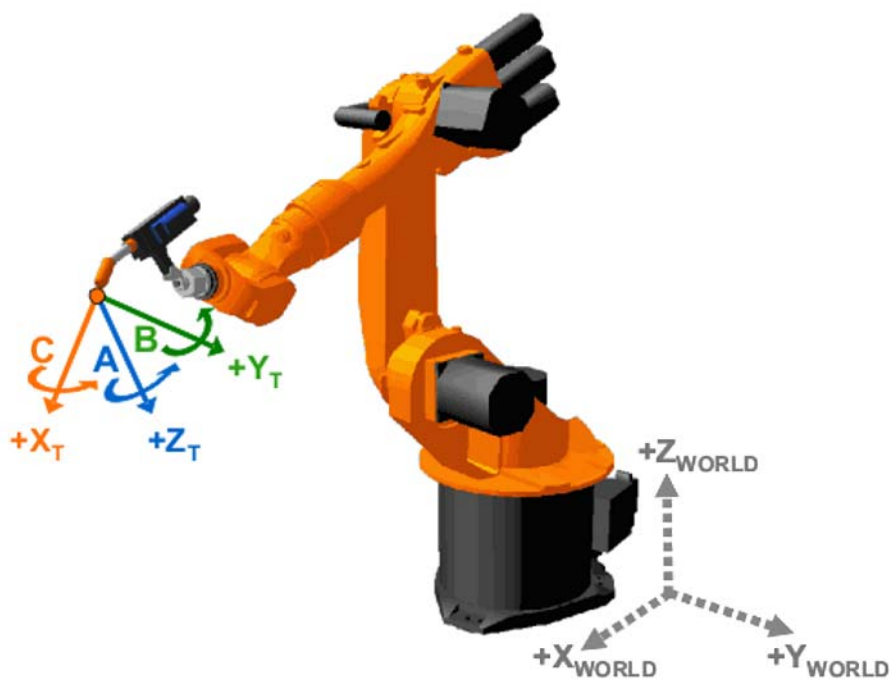


图 3-13: 绕 TCP 改变姿态

- 沿工具作业方向移动

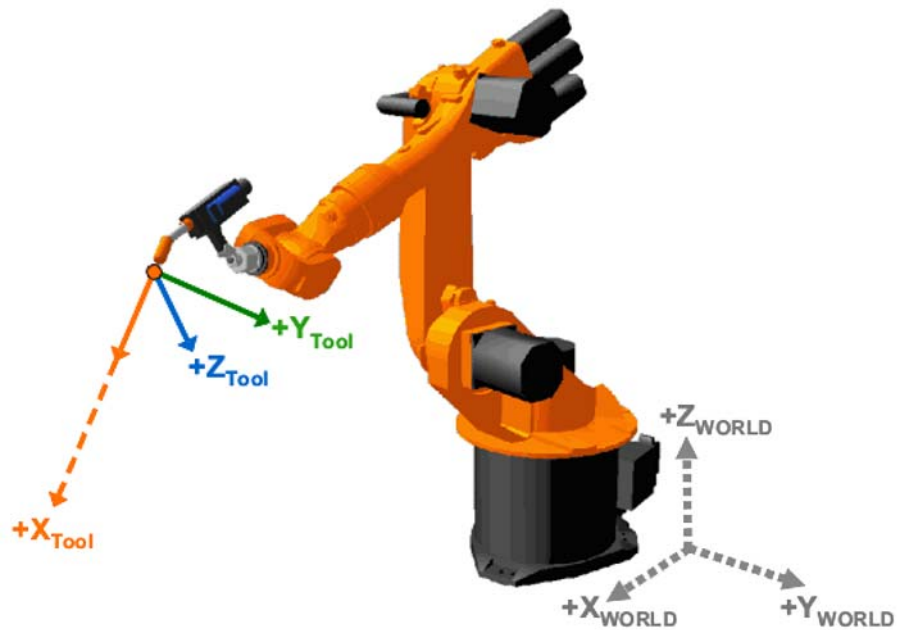


图 3-14: 作业方向 TCP

- 运动编程时的益处
 - 沿着 TCP 上的轨迹保持已编程的运行速度。

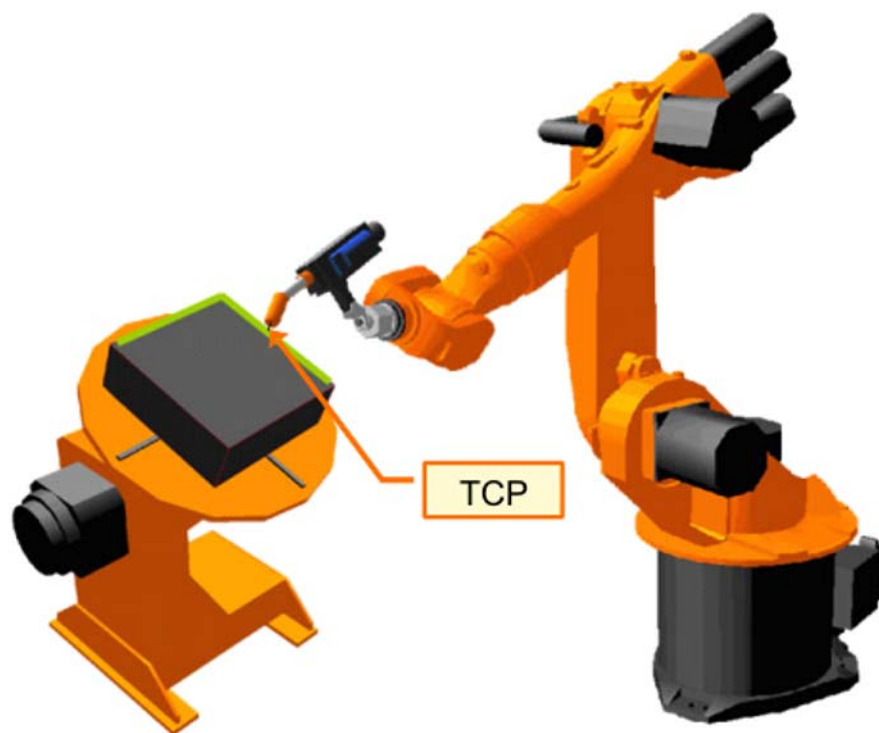


图 3-15: 带 TCP 编程的模式

- 此外，定义的姿态可沿着轨迹。



图 3-16: 以已测工具为例

工具测量的途径

工具测量分为 2 步：

步骤	说明
1	确定工具坐标系的原点 可选择以下方法： <ul style="list-style-type: none"> ■ XYZ 4 点法 ■ XYZ 参照法
2	确定工具坐标系的姿态 可选择以下方法： <ul style="list-style-type: none"> ■ ABC 世界坐标法 ■ ABC 2 点法
或者	直接输入至法兰中心点的距离值 (X,Y,Z) 和转角 (A, B, C)。 <ul style="list-style-type: none"> ■ 数字输入

TCP 测量的 XYZ 4 点法

将待测量工具的 TCP 从 4 个不同方向移向一个参照点。参照点可以任意选择。机器人控制系统从不同的法兰位置值中计算出 TCP。



移至参照点的 4 个法兰位置，彼此必须间隔足够远，并且不得位于同一平面内。

XYZ 4 点法的操作步骤：

1. 选择菜单序列 **投入运行 > 测量 > 工具 > XYZ 4 点**。
2. 为待测量的工具给定一个号码和一个名称。用**继续**键确认。

3. 用 TCP 移至任意一个参照点。按下软键**测量**，对话框“是否应用当前位置？继续测量”用**是**加以确认
4. 用 TCP 从一个其他方向朝参照点移动。重新按下**测量**，用**是**回答对话框提问。

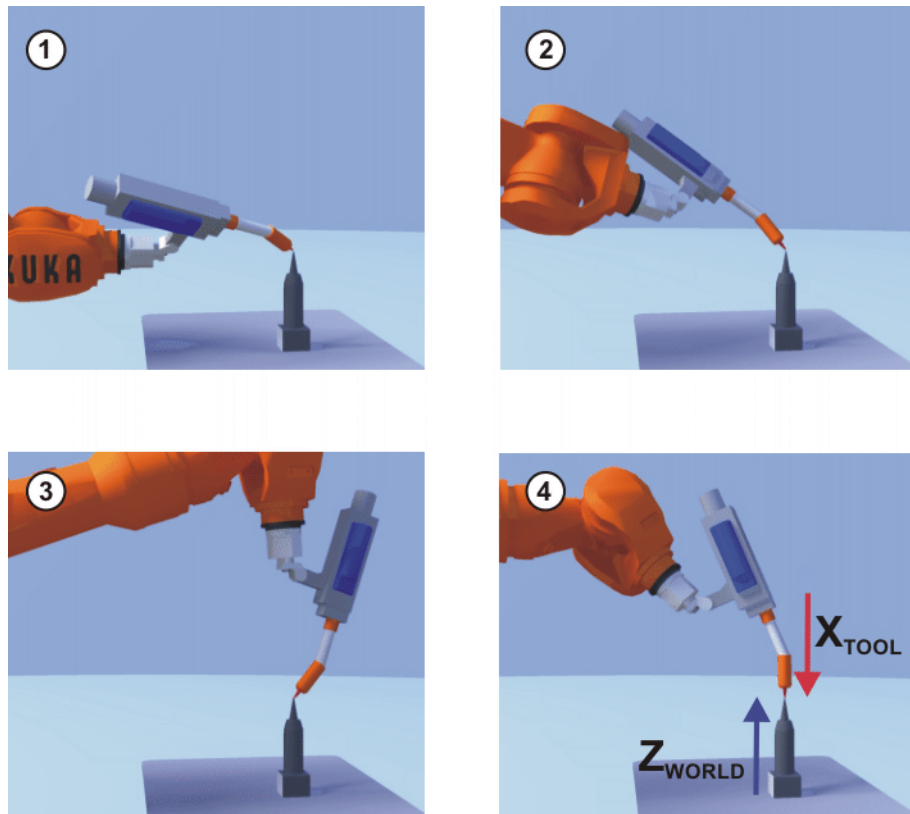


图 3-17: XYZ 4 点法

5. 把第 4 步重复两次。
6. 负载数据输入窗口自动打开。正确输入负载数据，然后按下**继续**。
7. 包含测得的 TCP X、Y、Z 值的窗口自动打开，测量精度可在误差项中读取。数据可通过**保存**直接保存。

TCP 测量的 XYZ 参照法

采用 XYZ 参照法时，将对一件新工具与一件已测量过的工具进行比较测量。机器人控制系统比较法兰位置，并对新工具的 TCP 进行计算。

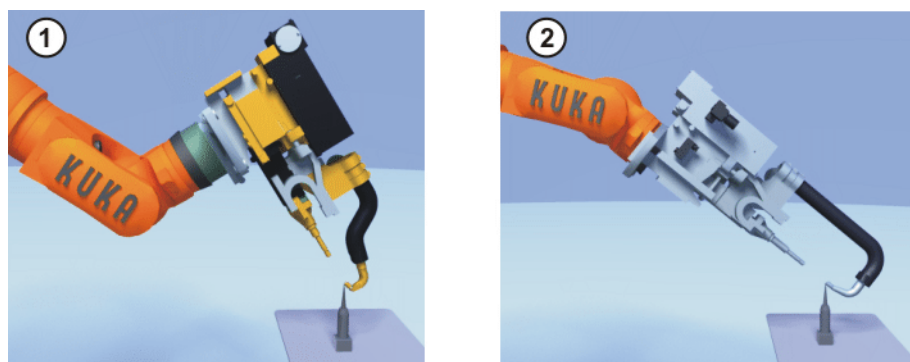


图 3-18

操作步骤

1. 前提条件是，在连接法兰上装有一个已测量过的工具，并且 TCP 的数据已知
2. 在主菜单中选择**投入运行 > 测量 > 工具 > XYZ 参照**。
3. 为新工具指定一个编号和一个名称。用**继续**键确认。

4. 输入已测量工具的 TCP 数据。用**继续**键确认。
5. 用 TCP 移至任意一个参照点。点击**测量**。用**继续**键确认。
6. 将工具撤回，然后拆下。装上新工具。
7. 将新工具的 TCP 移至参照点。点击**测量**。用**继续**键确认。
8. 按下**保存**键。数据被保存，窗口自动关闭。

或按下**负载数据**。数据被保存，一个窗口将自动打开，可以在此窗口中输入负载数据。

ABC 世界坐标系法 姿态测量

工具坐标系的轴平行于世界坐标系的轴进行校准。机器人控制系统从而得知工具坐标系的姿态。

此方法有两种方式：

- **5D**：只将工具的作业方向告知机器人控制器。该作业方向默认为 X 轴。其它轴的方向由系统确定，对于用户来说不是很容易识别。
应用范围：例如：MIG/MAG 焊接，激光切割或水射流切割
- **6D**：将所有 3 根轴的方向均告知机器人控制系统。
应用范围：例如：焊钳、抓爪或粘胶喷嘴

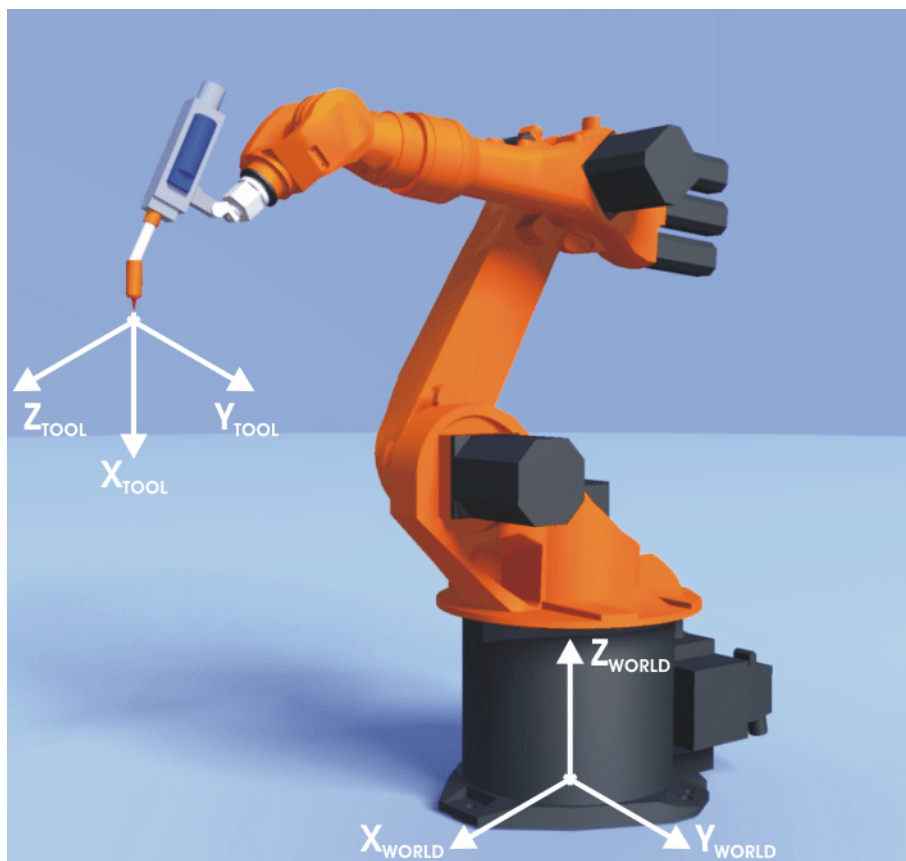


图 3-19: ABC 世界坐标系法

ABC 世界坐标系法操作步骤

- a. 在主菜单中选择**投入运行 > 测量 > 工具 > ABC 世界坐标**。
- b. 输入工具的编号。用**继续**键确认。
- c. 在 **5D/6D** 栏中选择一种变型。用**继续**键确认。
- d. 如果选择了 **5D**：
将 **+X 工具坐标** 调整至平行于 **-Z 世界坐标** 的方向。（**+X_{TOOL}** = 作业方向）
- e. 如果选择了 **6D**：

- 将 $+X$ 工具坐标调整至平行于 $-Z$ 世界坐标的方向。（ $+X_{TOOL}$ = 作业方向）
- $+Y$ 工具坐标调整至平行于 $+Y$ 世界坐标的方向。（ $+X_{TOOL}$ = 作业方向）
- $+Z$ 工具坐标调整至平行于 $+X$ 世界坐标的方向。（ $+X_{TOOL}$ = 作业方向）
- 用**测量**来确认。对信息提示“要采用当前位置吗？测量将继续”用**是**来确认。
 - 即打开另一个窗口。在此必须输入负荷数据。
 - 然后用**继续**和**保存**结束此过程。
 - 关闭菜单

姿态测量的 ABC 2 点法

通过趋近 X 轴上一个点和 XY 平面上一个点的方法，机器人控制系统即可得知工具坐标系的各轴。

当轴方向必须特别精确地确定时，将使用此方法。

i 下述操作步骤适用于工具碰撞方向为默认碰撞方向（= X 向）的情况。如果碰撞方向改为 Y 向或 Z 向，则操作步骤也必须相应地进行更改。

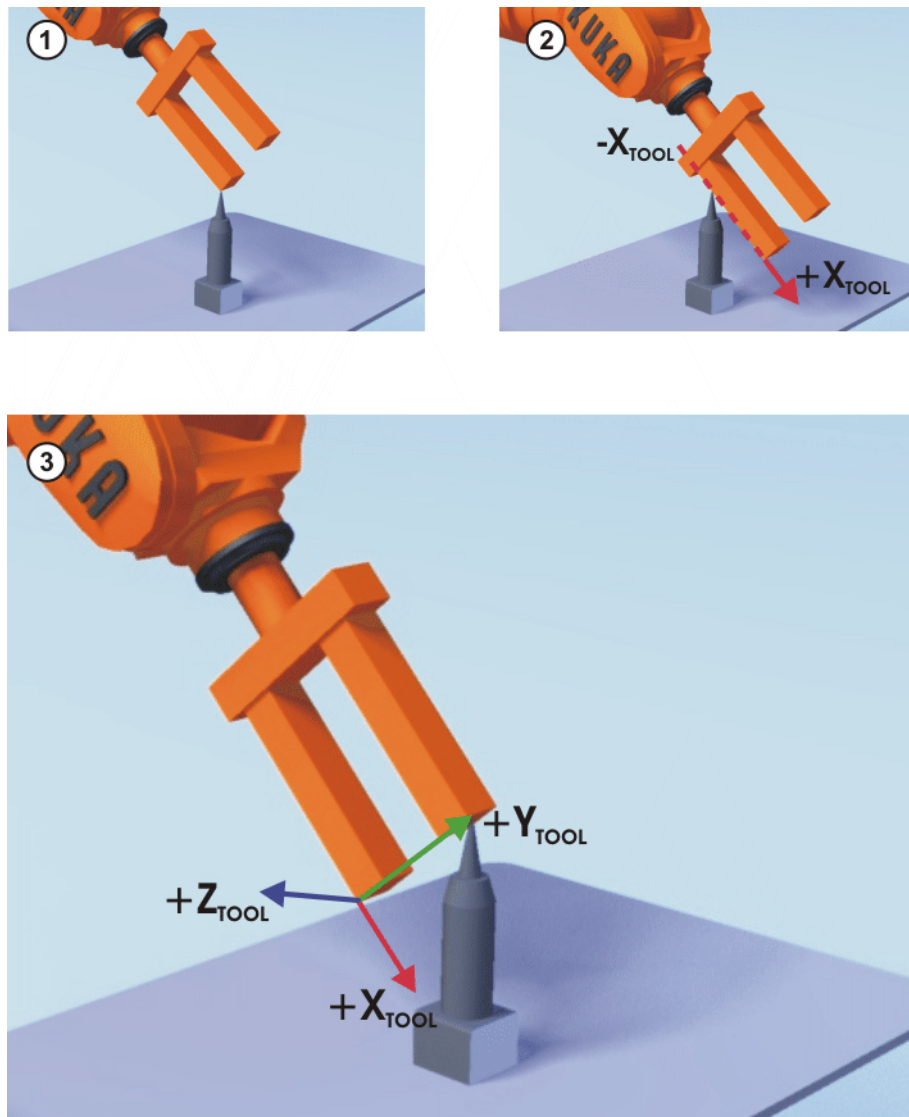


图 3-20: ABC 2 点法

- 前提条件是，TCP 已通过 XYZ 法测定。
- 在主菜单中选择**投入运行 > 测量 > 工具 > ABC 2 点**。

3. 输入已安装工具的编号。用**继续**键确认。
4. 用 TCP 移至任意一个参照点。点击**测量**。用**继续**键确认。
5. 移动工具，使参照点在 X 轴上与一个为负 X 值的点重合（即与作业方向相反）。点击**测量**。用**继续**键确认。
6. 移动工具，使参照点在 XY 平面上与一个在正 Y 向上的点重合。点击**测量**。用**继续**键确认。
7. 按**保存**。数据被保存，窗口关闭。
或按下**负载数据**。数据被保存，一个窗口将自动打开，可以在此窗口中输入负载数据。

培训期间使用抓爪时的安全提示

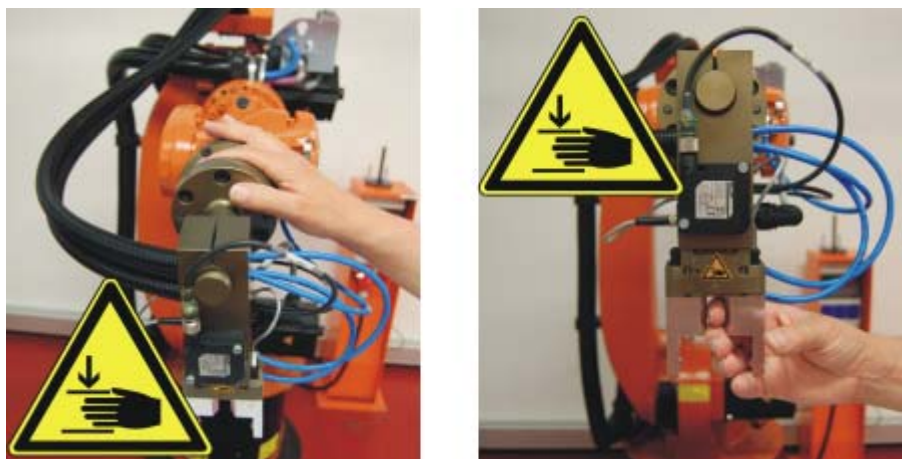


图 3-21: 培训时所用的抓爪有挤伤危险

警告 警告！
使用抓爪系统时有挤伤和割伤的危险。抓爪操作人员必须保证无身体部分可被抓爪挤伤。

夹入部件（方头、尖触头）时必须非常小心。

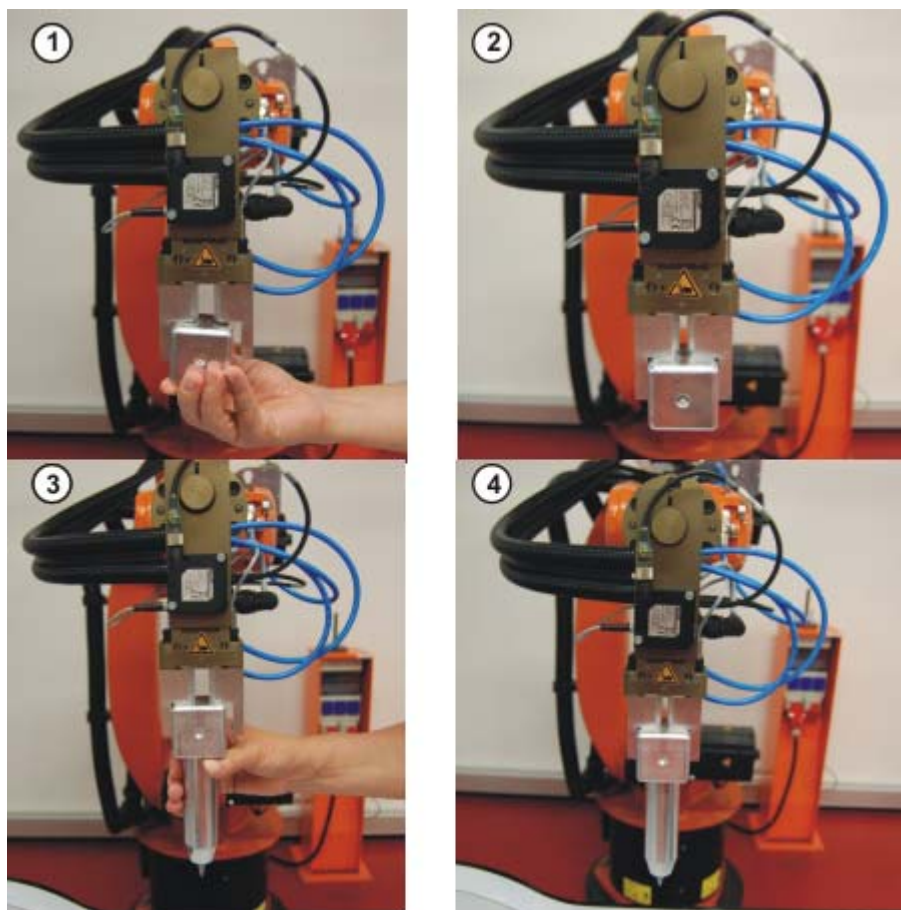


图 3-22: 将物件夹紧到培训用抓爪中

序号	备注
1	夹入方头
2	方头已夹入
3	夹入尖触头
4	尖触头已夹入

发生碰撞时会触发碰撞保险装置。

机器人在因碰撞而触发碰撞保险装置后撤回。一位学员按下开关 (1) 并从机器人、碰撞保险装置以及抓爪上移开肢体。另一位学员应在机器人撤回前确认，机器人的运动不会危及任何人员。



图 3-23: 碰撞保险装置的撤回开关

3.6 练习：尖触头的工具测量

练习目的

成功完成此练习后，您可执行下列操作：

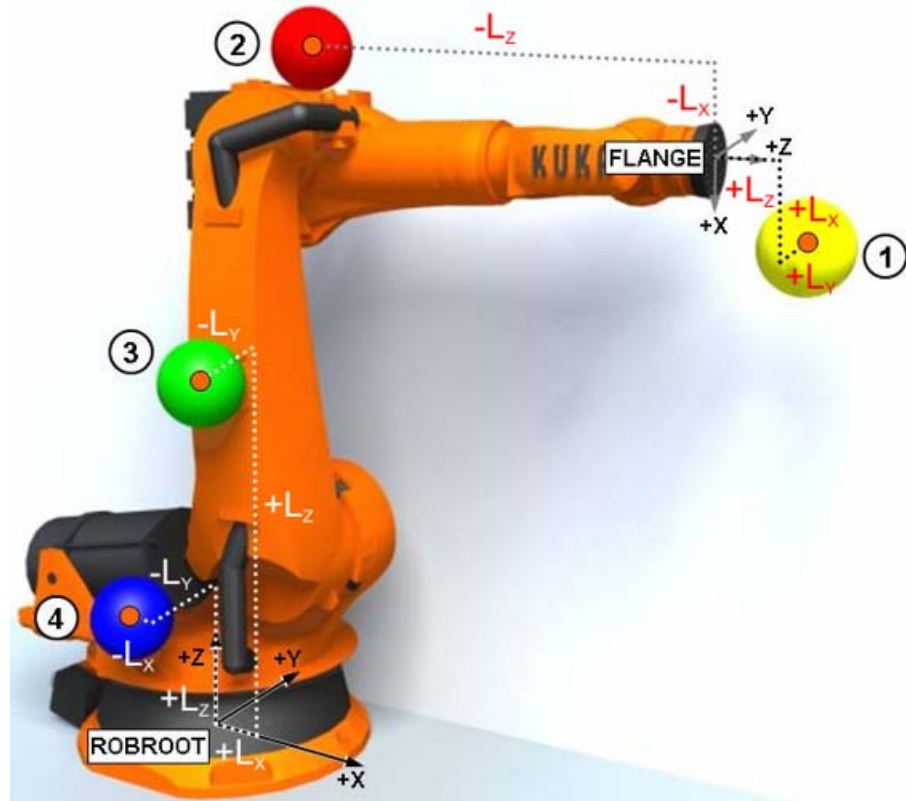
- 用 XYZ 4 点法和 ABC 世界坐标系法测量工具
- 激活一个测定的工具
- 在工具坐标系中移动
- 沿工具作业方向移动
- 将工具围绕工具中心点 (Tool Center Point, 即 TCP) 改变姿态

前提条件

为成功完成此练习，必须满足以下前提条件：

- 具备有关工具工作点的各种测量方法，特别是 XYZ 4 点法的理论知识
- 具备有关工具姿态的各种测量方法，特别是 ABC 世界坐标系法的理论知识

- 具备有关机器人负载数据及其输入的理论知识



- | | |
|-------------|-------------|
| 1 负荷 | 3 轴 2 的附加负载 |
| 2 轴 3 的附加负载 | 4 轴 1 的附加负载 |

练习内容

请完成以下任务：尖触头的测量

1. 用 XYZ 4 点法测量尖触头的 TCP。使用黑色的金属顶尖作为参照顶尖。从尖触头库中取出最上方的尖触头，然后将其夹入抓爪。使用工具编号 2 并指定名称尖触头 1。误差不得大于 0.95 mm。
2. 保存工具数据
3. 采用 ABC 世界坐标系 5D 法测量工具姿态。
4. 请输入负载数据。

抓有作为工具编号 2 的尖触头的抓爪负载数据：

质量：

$M = 7.32 \text{ kg}$

重心：

$X = 21 \text{ mm}$

$Y = 21 \text{ mm}$

$Z = 23 \text{ mm}$

姿态：

$A = 0^\circ$

$B = 0^\circ$

$C = 0^\circ$

转动惯量：

$J_X = 0 \text{ kgm}^2$

$J_Y = 0.2 \text{ kgm}^2$

$J_Z = 0.3 \text{ kgm}^2$

5. 保存 TOOL (工具坐标) 数据，并在工具坐标系中测试尖触头的移动
- 与练习配套的问题**

1. 为什么要测量由机器人引导的工具？

.....

.....

.....

2. 什么可以通过 XYZ 4 点法确定？

.....
.....

3. 工具测量方法有哪些？

.....
.....
.....
.....

3.7 练习：抓爪工具测量，2 点法

练习目的

成功完成此练习后，您可执行下列操作：

- 借助于 XYZ 4 点和 ABC 2 点法进行工具测量
- 激活一个测定的工具
- 在工具坐标系中移动
- 沿工具作业方向移动
- 将工具围绕工具中心点 (Tool Center Point, 即 TCP) 改变姿态

前提条件

为成功完成此练习，必须满足以下前提条件：

- 具备有关工具工作点的各种测量方法，特别是 2 点法的理论知识
- 具备有关机器人负荷数据及其输入的理论知识

练习内容

请完成以下任务：抓爪的数字式测量 ...

1. 如图所示用 XYZ 4 点法测量抓爪的工具中心点 TCP：
2. 用 ABC 2 点法测量抓爪坐标系的姿态
3. 输入负荷数据。

抓爪的负荷数据：

质量：

$M = 6.68 \text{ kg}$

重心：

$X = 23 \text{ mm}$

$Y = 11 \text{ mm}$

$Z = 41 \text{ mm}$

姿态：

$A = 0^\circ$

$B = 0^\circ$

$C = 0^\circ$

转动惯量：

$J_X = 0 \text{ kgm}^2$

$J_Y = 0.4 \text{ kgm}^2$

$J_Z = 0.46 \text{ kgm}^2$

4. 保存 TOOL (工具坐标) 数据并在工具坐标系中用抓爪手动移动

此外也可通过输入数字测定抓爪：

X	Y	Z	A	B	C
132.05 mm	171.30 mm	173.00 mm	45°	0°	180°

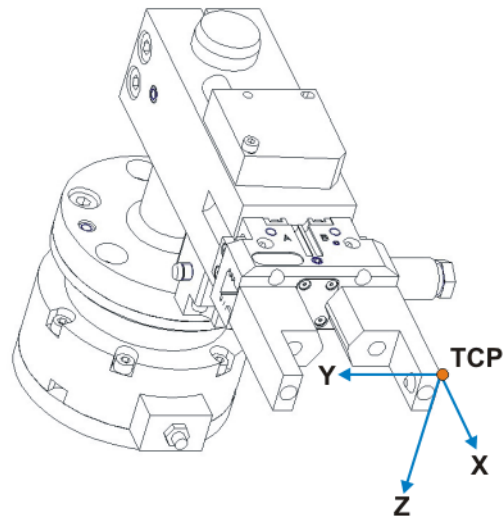


图 3-24: 学院的抓爪 : TCP 的位置

与练习配套的问题

1. 哪个图标代表工具坐标系？

a)



b)



c)



d)



2. 控制器最多可管理多少工具？

.....

3. 工具负荷数据中的值 -1 表示什么？

.....

.....

3.8 测量基坐标

说明

基坐标系测量表示根据世界坐标系在机器人周围的某一个位置上创建坐标系。其目的是使机器人的运动以及编程设定的位置均以该坐标系为参照。因此，设定的工件支座和抽屉的边缘、货盘或机器的外缘均可作为基准坐标系中合理的参照点。

基坐标系测量分为两个步骤：

1. 确定坐标原点
2. 定义坐标方向

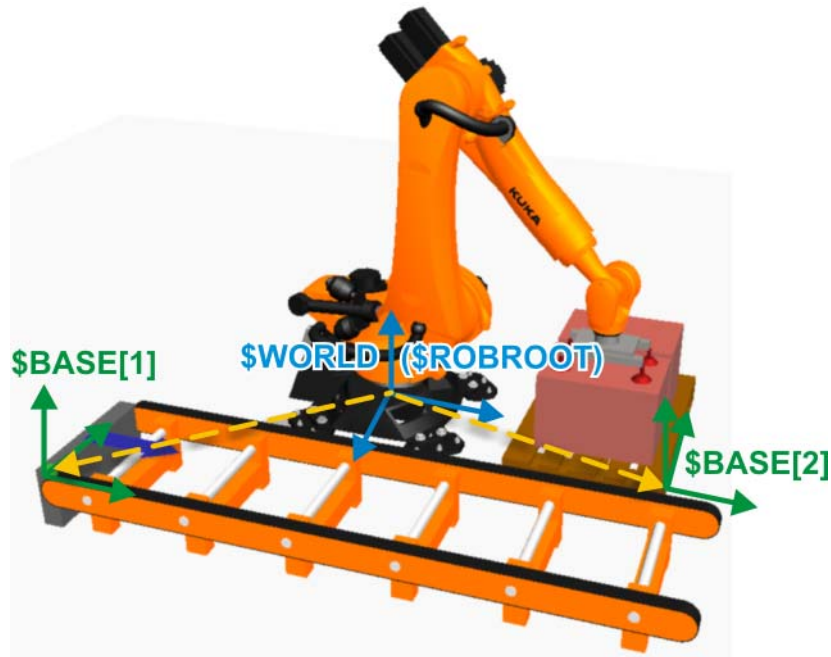


图 3-25: 基坐标测量

优势

测定了基坐标后有以下优点：

- 沿着工件边缘移动：
可以沿着工作面或工件的边缘手动移动 TCP。

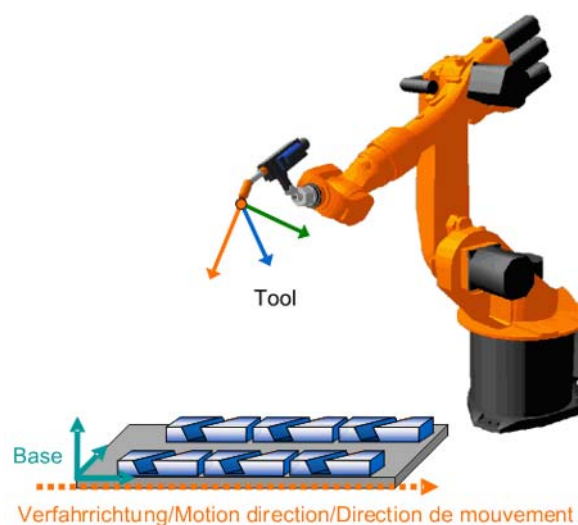


图 3-26: 基坐标测量的优点：移动方向

- 参照坐标系：
示教的点以所选的坐标系为参照。

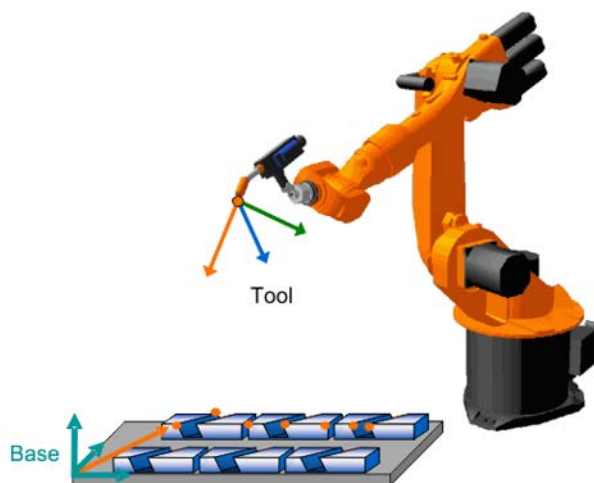


图 3-27: 基坐标测量的优点：以所需坐标系为参照

- 坐标系的修正 / 推移：

可以参照基坐标对点进行示教。如果必须推移基坐标，例如由于工作面被移动，这些点也随之移动，不必重新进行示教。

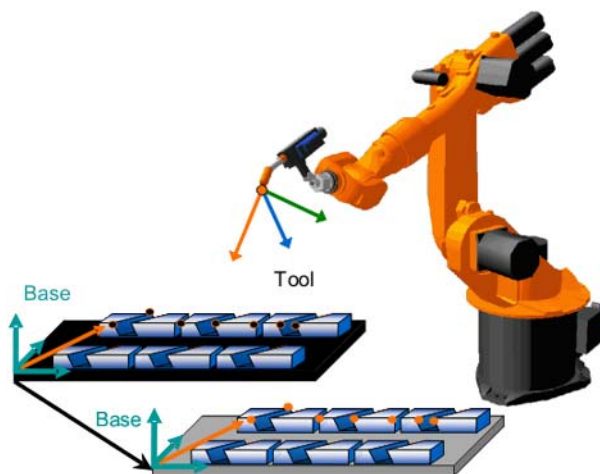


图 3-28: 基坐标测量的优点：基坐标系的位移

- 多个基坐标系的益处：

最多可建立 32 个不同的坐标系，并根据程序流程加以应用。

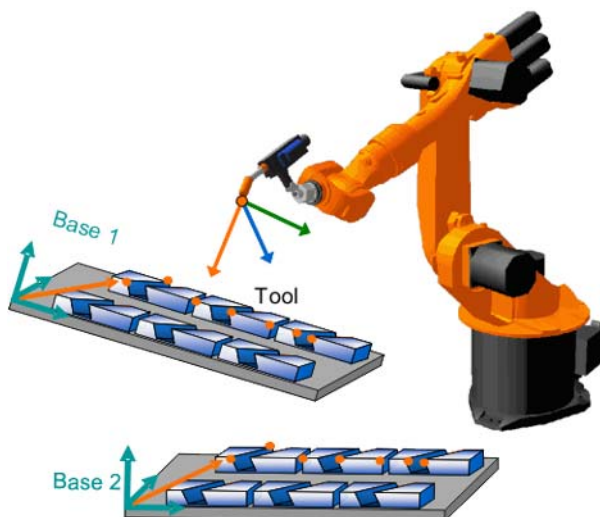


图 3-29: 基坐标测量的优点：使用多个基坐标系

基坐标测量的方法 以下的基坐标测量方法可供使用：

方法	说明
3 点法	1. 定义原点 2. 定义 X 轴正方向 3. 定义 Y 轴正方向 (XY 平面)
间接法	当无法移至基坐标原点时，例如，由于该点位于工件内部，或位于机器人工作空间之外时，须采用间接法。 此时须移至基坐标的 4 个点，其坐标值必须已知 (CAD 数据)。机器人控制系统根据这些点计算基坐标。
数字输入	直接输入至世界坐标系的距离值 (X,Y,Z) 和转角 (A , B , C)。



有关间接测量的详细信息可查阅 *KUKA 系统软件 8 的操作及编程指南*。

3 点法操作步骤

注意

基坐标测量只能用一个事先已测定的工具进行 (TCP 必须为已知的)。

1. 在主菜单中选择投入运行 > 测量 > 基坐标系 > 3 点。
2. 为基坐标分配一个号码和一个名称。用继续键确认。
3. 输入需用其 TCP 测量基坐标的工具的编号。用继续键确认。
4. 用 TCP 移到新基坐标的原点。点击测量软键并用是键确认位置。

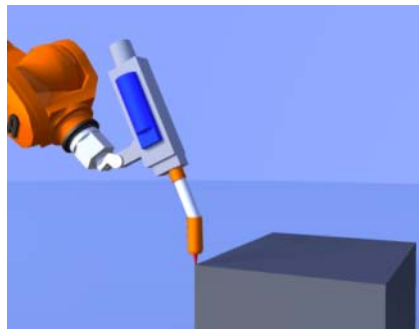


图 3-30: 第一个点：原点

5. 将 TCP 移至新基座正向 X 轴上的一个点。点击测量并用是键确认位置。

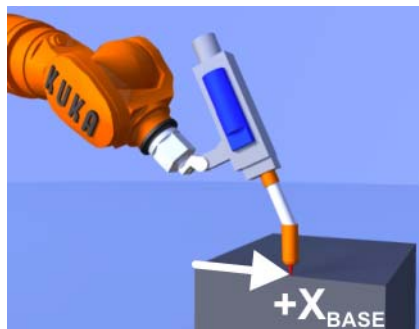


图 3-31: 第二个点：X 向

6. 将 TCP 移至 XY 平面上一个带有正 Y 值的点。点击测量并用是键确认位置。

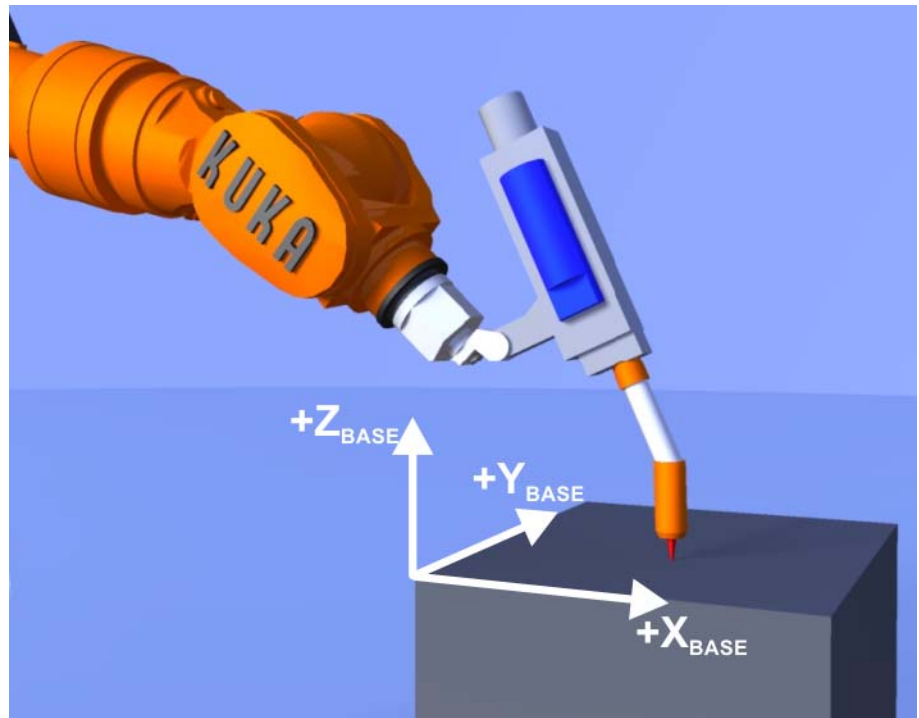


图 3-32: 第三个点 : XY 平面

7. 按下**保存键**。
8. 关闭菜单



三个测量点不允许位于一条直线上。这些点间必须有一个最小夹角 (标准设定 2.5°)。

3.9 查询当前机器人位置

机器人位置的显示方式 当前的机器人位置可通过两种不同方式显示：

- **轴极坐标：**

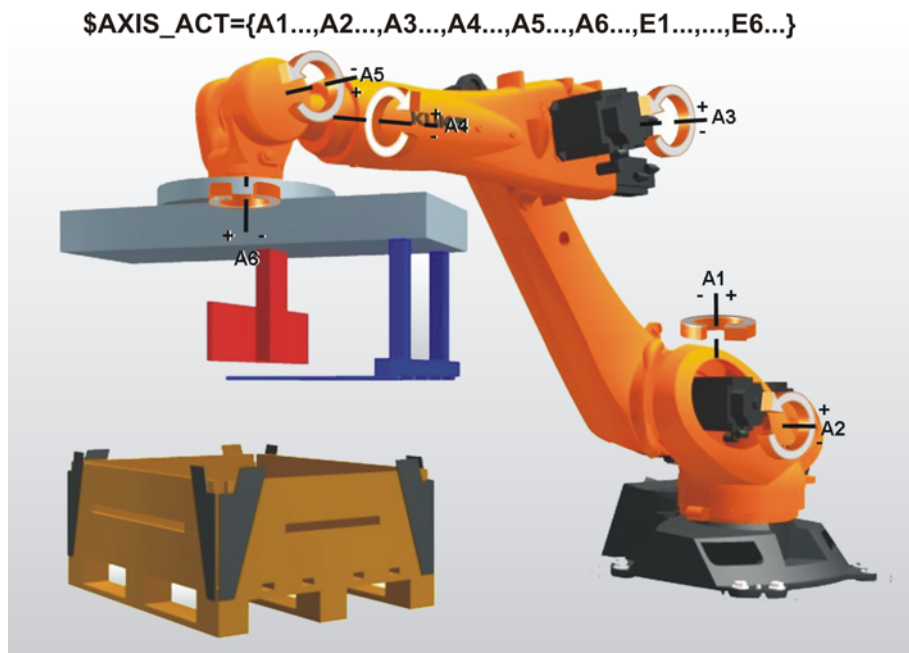


图 3-33: 轴坐标中的机器人位置

显示每根轴的当前轴角：该角等于与零点标定位置之间的角度绝对值。

■ 笛卡尔式：

$\$POS_ACT=\{X...,Y...,Z...,A...,B...,C...,S...,T...,E1...,...\}$

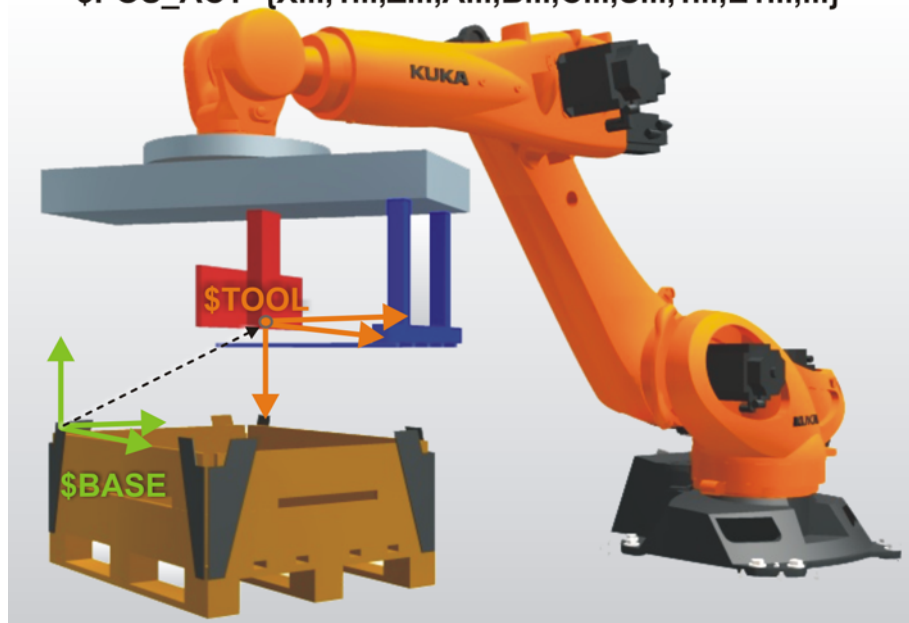


图 3-34: 笛卡尔位置

在当前所选的基坐标系中显示当前 TCP 的当前位置（工具坐标系）。

没有选择工具坐标系时，法兰坐标系适用！

没有选择基坐标系时，世界坐标系适用！

不同基坐标系中的
笛卡尔位置

观察下图时，我们会立即意识到，机器人的三个位置都相同。位置指示器在这三种情况下显示不同的值：

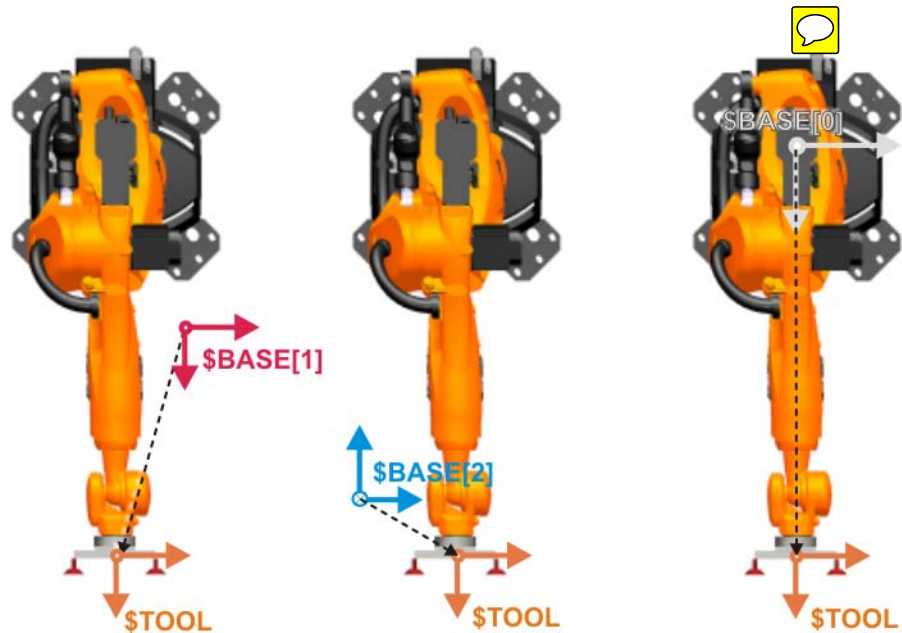


图 3-35: 三个机器人位置 - 一个机器人工位！

在相应的基坐标系中显示工具坐标系 /TCP 的位置：

- 对于基坐标系 1
- 对于基坐标系 2
- 对于基坐标系 $\$NULLFRAME$ ：这相当于机器人底座坐标系（通常也就是世界坐标系）！



仅当选择了正确的基坐标系和正确的工具时，笛卡尔坐标系中的实际位置指示器才显示所期望的值！

询问机器人位置

操作步骤：

- 在菜单中选择显示 > 实际位置。将显示笛卡尔式实际位置。
- 按轴坐标以显示轴坐标式的实际位置。
- 按笛卡尔以再次显示笛卡尔式的实际位置

3.10 练习：工作台的基坐标测量，3 点法

练习目的

成功完成此练习后，您可执行下列操作：

- 确定任意一个基坐标
- 测量基坐标
- 激活已测量过的、用于手动移动的基坐标
- 在基坐标系中移动

前提条件

为成功完成此练习，必须满足以下前提条件：

- 具备有关基坐标测量的方法，特别是 3 点法的理论知识

练习内容

请完成以下任务：

1. 用 3 点法测量工作台上的蓝色基坐标。请指定基坐标编号 1 的名称为蓝色。请使用已测量过的尖触头 1（工具编号 2）作为测量工具。
2. 请保存已测量过的基坐标的数据
3. 用 3 点法测量工作台上的红色基坐标。请指定基坐标编号 2 的名称为红色。请使用已测量过的尖触头 1（工具编号 2）作为测量工具。
4. 请保存已测量过的基坐标的数据
5. 将该工具移到蓝色基坐标系的原点，并同时以笛卡尔坐标显示实际位置。

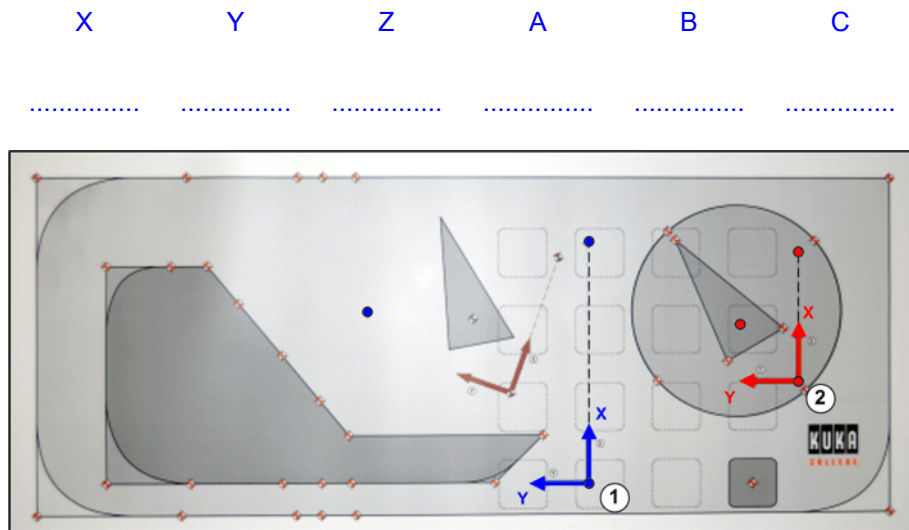


图 3-36: 工作台上的基坐标测量

与练习配套的问题

1. 为什么要测量基坐标？

.....

.....

.....

2. 哪个图标代表基坐标系？



3. 基坐标测量方法有哪些？

.....

.....

.....

4. 控制器最多可管理多少基坐标系？

.....

5. 请说明 3 点法的测量要点

.....

.....

.....

3.11 固定工具测量

概览

固定工具的测量分为 2 步：

1. 确定固定工具的外部 TCP 和世界坐标系原点之间的距离。
2. 根据外部 TCP 确定该坐标系姿态。

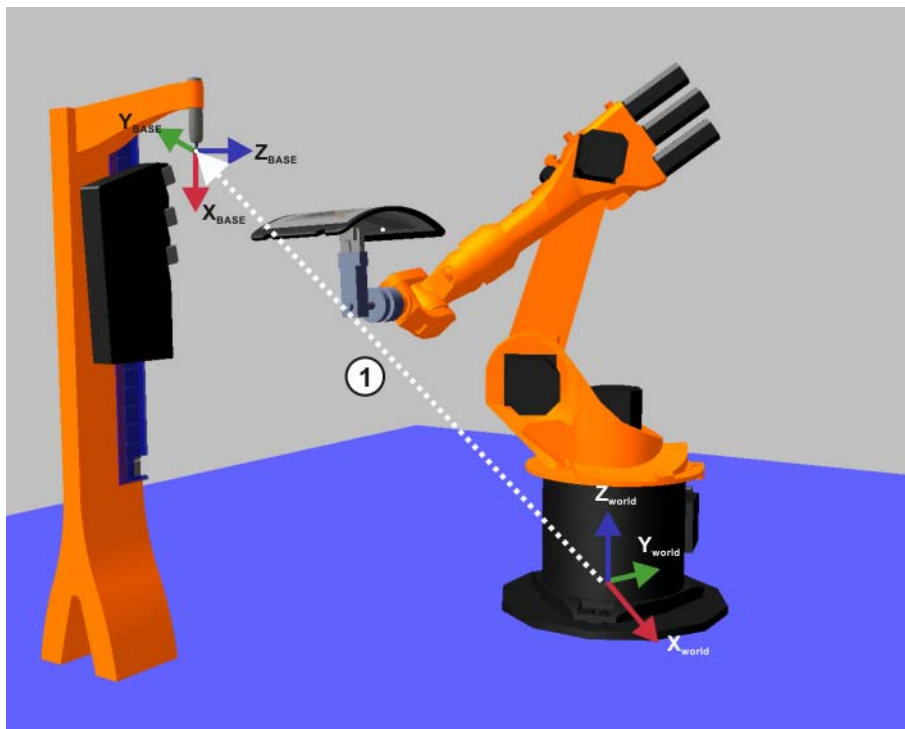



图 3-37: 固定工具的测量

 (1) (>>> 图 3-37) 所示，以 \$WORLD (或者 \$ROBROOT) 为基准管理外部 TCP，即等同于基坐标系。

测量说明

- 确定 TCP 时需要一个由机器人引导的已测工具。

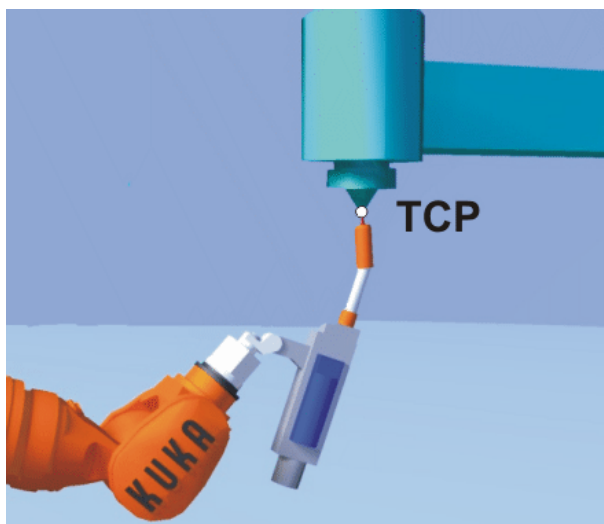


图 3-38: 移至外部 TCP

- 确定姿态时要将法兰的坐标系校准至平行与新的坐标系。有两种方式：
 - 5D：只将固定刀具的作业方向告知机器人控制器。该作业方向被默认为 X 轴。其它轴的姿态将由系统确定，对用户来说，不是很容易地就能识别。
 - 6D：所有 3 个轴的姿态都将告知机器人控制系统。

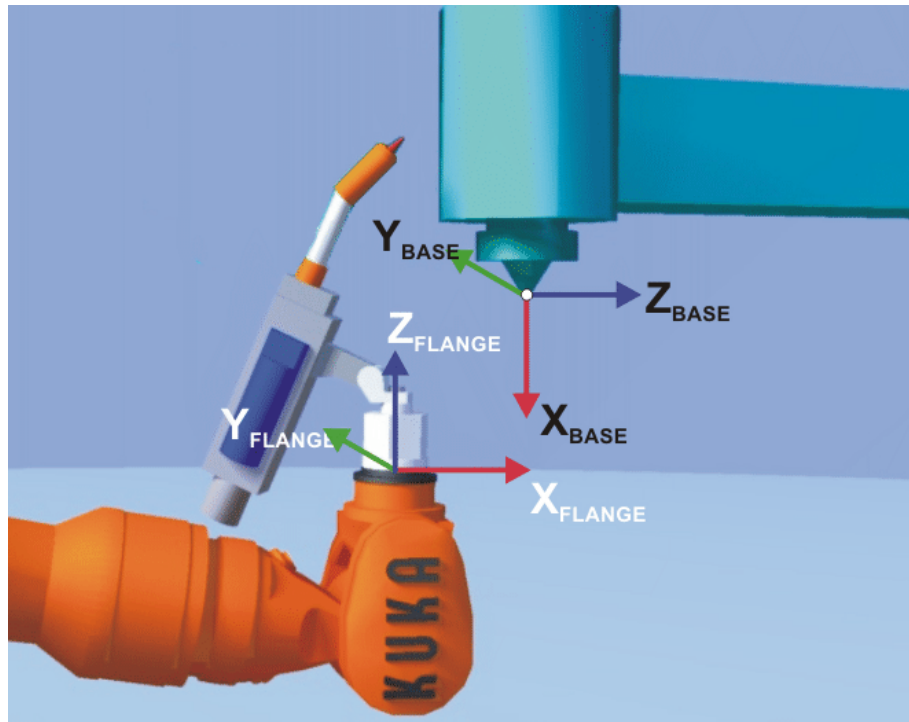


图 3-39: 对坐标系进行平行校准

操作步骤

1. 在主菜单中选择**投入运行 > 测量 > 固定工具 > 工具**。
2. 为固定工具指定一个号码和一个名称。用**继续**键确认。
3. 输入所用参考工具的编号。用**继续**键确认。
4. 在**5D/6D** 栏中选择一种规格。用**继续**键确认。
5. 用已测量工具的 TCP 移至固定工具的 TCP。点击**测量**。用**是**确认位置。
6. 如果选择了**5D** :
 将 **+X 基坐标系** 平行对准 **-Z 法兰坐标系**。
 (也就是将连接法兰调整成与固定工具的作业方向垂直。)
 如果选择了**6D** :
 应对连接法兰进行调整, 使得它的轴平行于固定工具的轴 :
 - **+X 基坐标系** 平行于 **-Z 法兰坐标系**
 (也就是将连接法兰调整成与工具的作业方向垂直。)
 - **+Y 基坐标系** 平行于 **+Y 法兰坐标系**
 - **+Z 基坐标系** 平行于 **+X 法兰坐标系**
7. 点击**测量**。用**是**确认位置。
8. 按下**保存**键。

3.12 测量由机器人引导的工件

概览：直接测量

注意

下文仅讨论直接测量法。间接测量法极少使用, 在 *KUKA 系统软件 8.1 操作和编程说明文件* 中将对此作详细说明。

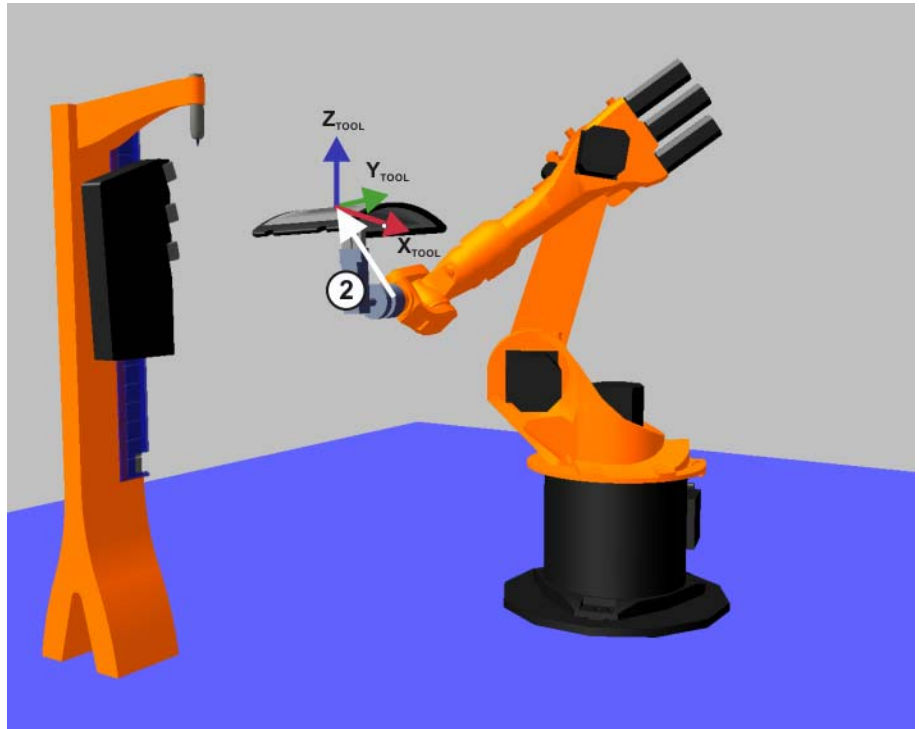


图 3-40: 通过直接测量的方法测量工件

部件	测量
2	工件测量

说明

机器人控制系统将得知工件的原点和其它 2 个点。此 3 个点将该工件清楚地定义出来。

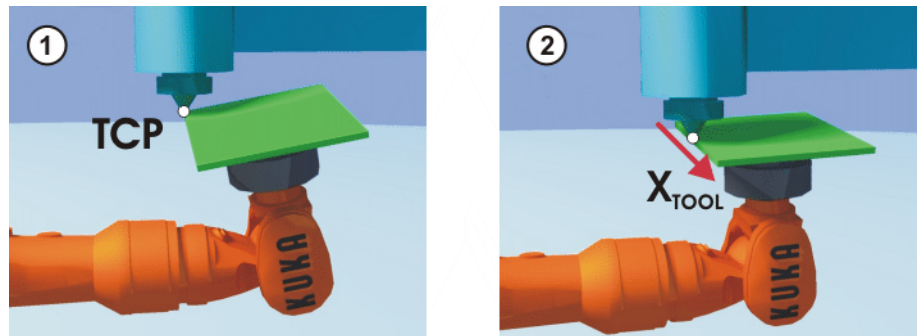


图 3-41

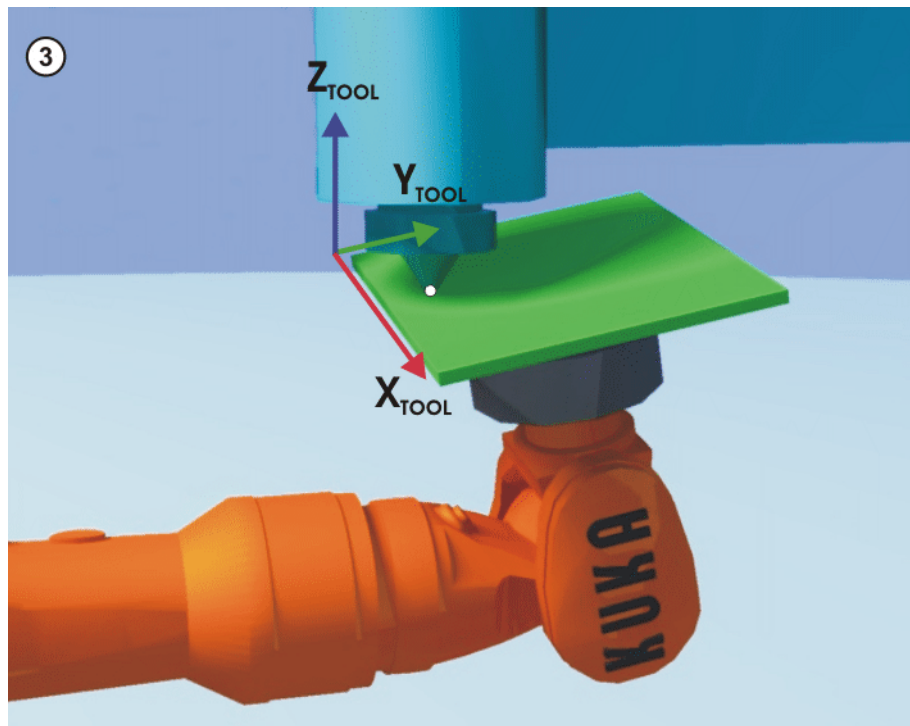


图 3-42: 测量工件：直接法

操作步骤

1. 选择菜单序列**投入运行 > 测量 > 固定工具 > 工件 > 直接测量**。
2. 为工件分配一个编号和一个名称。用**继续**键确认。
3. 输入固定工具的编号。用**继续**键确认。
4. 将工件坐标系的原点移至固定工具的 TCP 上。
点击**测量**键并用**是**确认位置。
5. 将在工件坐标系的正向 X 轴上的一点移至固定工具的 TCP 上。
点击**测量**键并用**是**确认位置。
6. 将一个位于工件坐标系的 XY 平面上、且 Y 值为正的点移至固定工具的 TCP 上。
点击**测量**键并用**是**确认位置。
7. 输入工件负载数据，然后按下**继续**。
8. 按下**保存**键。

3.13 练习：测量外部工具和机器人引导的工件

练习目的

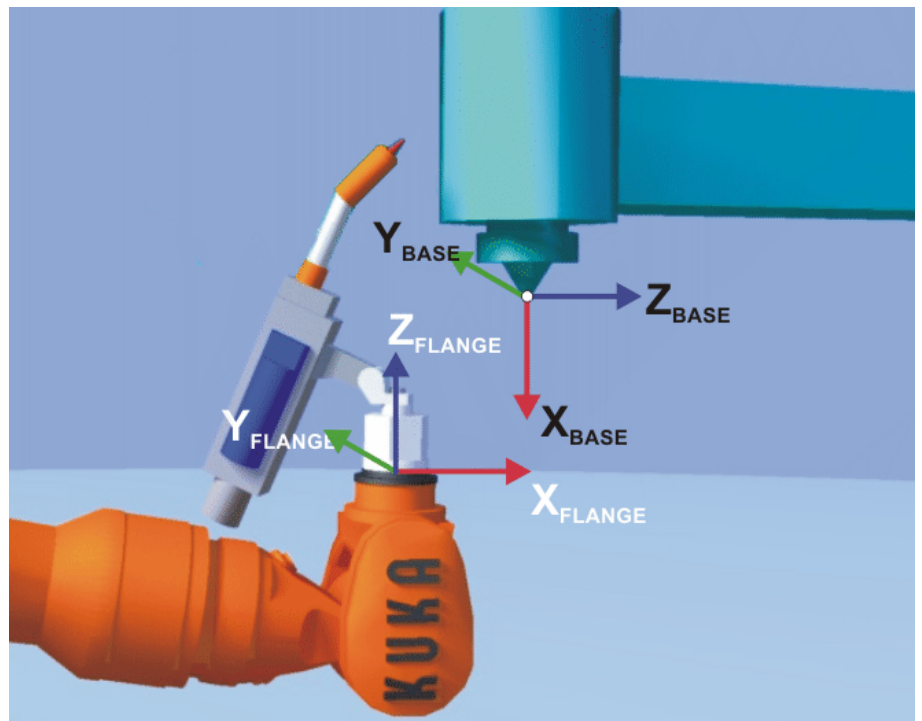
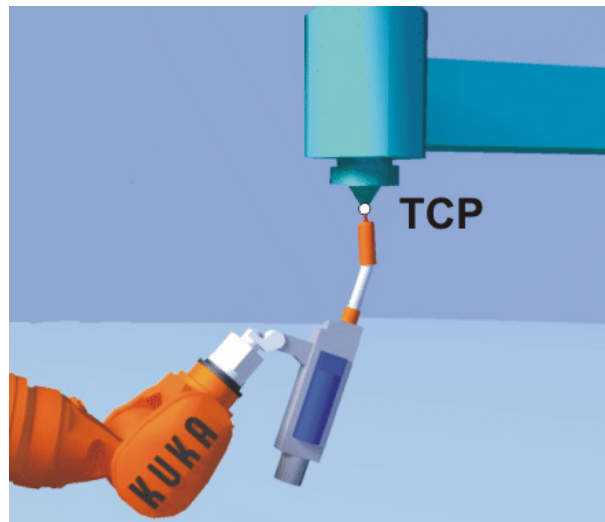
成功完成此练习后，您可执行下列操作：

- 测量固定工具
- 测量活动工具
- 手动移动外部工具

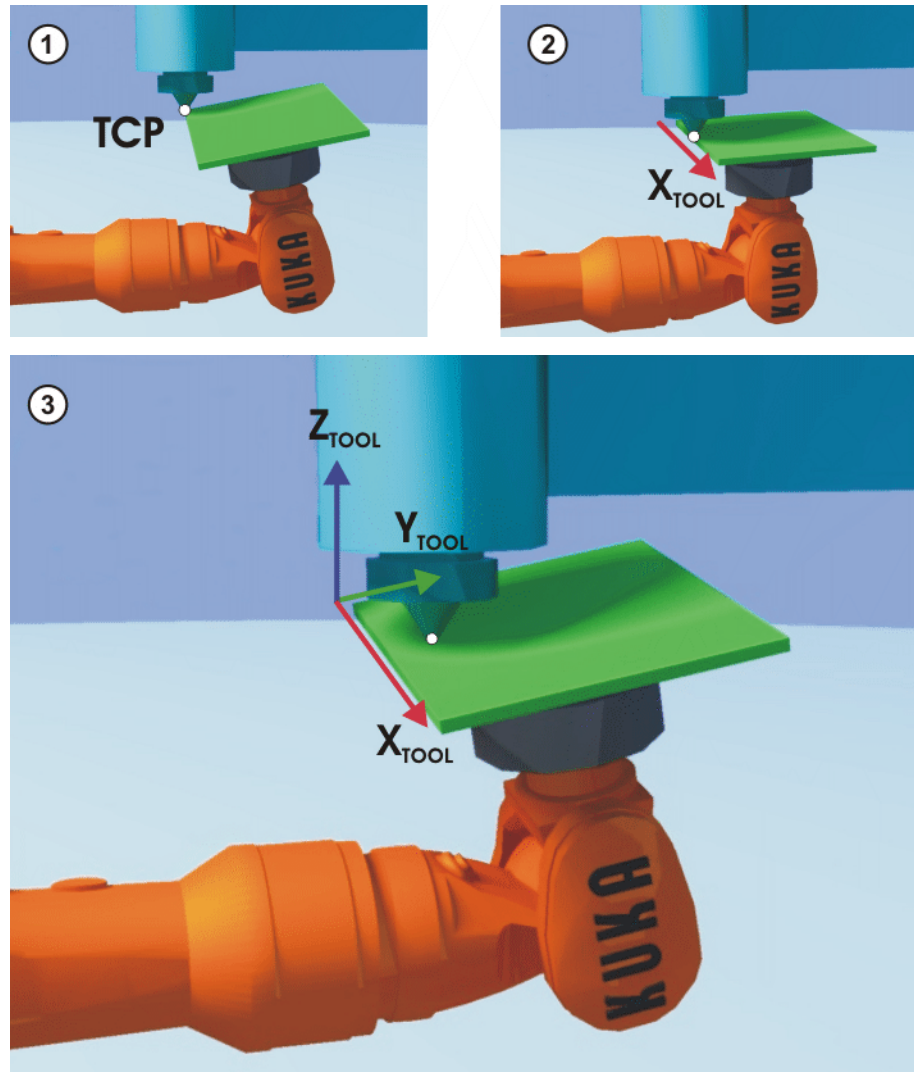
前提条件

为成功完成此练习，必须满足以下前提条件：

- 具备固定工具测量法的理论知识



- 具备固定工具时工件测量的理论知识，尤其是直接测量法



练习内容

请完成以下任务：测量喷嘴和标牌

1. 为了测量固定工具，应将已测量的尖触头 1（工具编号 1）用作参照工具。请为固定工具指定**工具编号 10** 和名称**喷嘴**
 - 每次测量时都注意保存好您的数据！
2. 请测量由机器人引导的工件。此外，请指定**工具编号 12** 和名称**标牌**
 - 请输入负载数据。

抓爪抓有标牌时的负载数据：

质量：

$M = 8.54 \text{ kg}$

质量重心：

$X = 46 \text{ mm}$

$Y = 93 \text{ mm}$

$Z = 5 \text{ mm}$

姿态：

$A = 0^\circ$

$B = 0^\circ$

$C = 0^\circ$

转动惯量：

$J_x = 0.3 \text{ kgm}^2$

$J_y = 0.5 \text{ kgm}^2$

$J_z = 0.6 \text{ kgm}^2$

3. 测量后请激活要手动移动的外部工具。请合理使用基坐标系和工具坐标系，然后移动机器人。
4. 请将 TCP 移动到被测工件的基坐标原点，然后以笛卡尔坐标显示实际位置。

实际位置：

X Y Z A B C

.....

与练习配套的问题

1. 如何测量安装在机器人法兰上的工件的基坐标？

.....
.....
.....

2. 如何确定外部工具的 TCP？

.....
.....
.....

3. 为何需要外部 TCP？

.....
.....
.....

4. 为沿工具作业方向移动 TCP 需要进行哪些设置？

.....
.....
.....

3.14 拔出 smartPAD

取下 smartPAD 的过程说明

- smartPAD 可在进行机器人控制时取下。
- 插入的 smartPAD 会应用机器人控制器的当前运行方式。
- 可随时插入 smartPAD。
- 插入时必须注意使用与取下的 smartPAD 变型相同的 smartPAD (固件版本)
- 插入 30 秒后, 紧急停止和确认开关才重又恢复功能。
- smartHMI (操作界面) (在 15 秒内) 重又自动显示

功能“拔下 smartPAD”

警告 如果已拔下 smartPAD, 则无法再通过 smartPAD 上的紧急停止按键来关断设备。因此必须在机器人控制系统上外接一个紧急停止装置。

警告 用户应负责将拔下的 smartPAD 立即从设备中取出并将其妥善保管。保管处应远离在工业机器人上作业的人员的视线和接触范围。目的是为了防止混淆有效的和无效的紧急停止装置。

警告 如果没有注意该措施, 则可能会造成人员死亡、重伤或巨大的财产损失。

警告 将 smartPAD 插在机器人控制器上的操作人员, 至少必须在 smartPAD 旁停留 30 秒, 就是说直到紧急停止和确认开关再次恢复正常功能。这样就可避免出现例如另一操作人员在紧急情况下使用的紧急停止装置暂时无效的情况。

拔下 smartPAD 的操作步骤

拔下:

1. 按下用来拔下 smartPAD 的按钮。
smartHMI 上会显示一个信息和一个计时器。计时器会计时 25 秒。在此时间内可从机器人控制器上拔下 smartPAD。



图 3-43: 用于脱开 smartPAD 的按钮

注意 如果在计数器未运行的情况下取下 smartPAD, 会触发紧急停止。只有重新插入 smartPAD 才能取消紧急停止。

2. 打开配电箱门 (V)KR C4

3. 从机器人控制器上拔下 smartPAD。

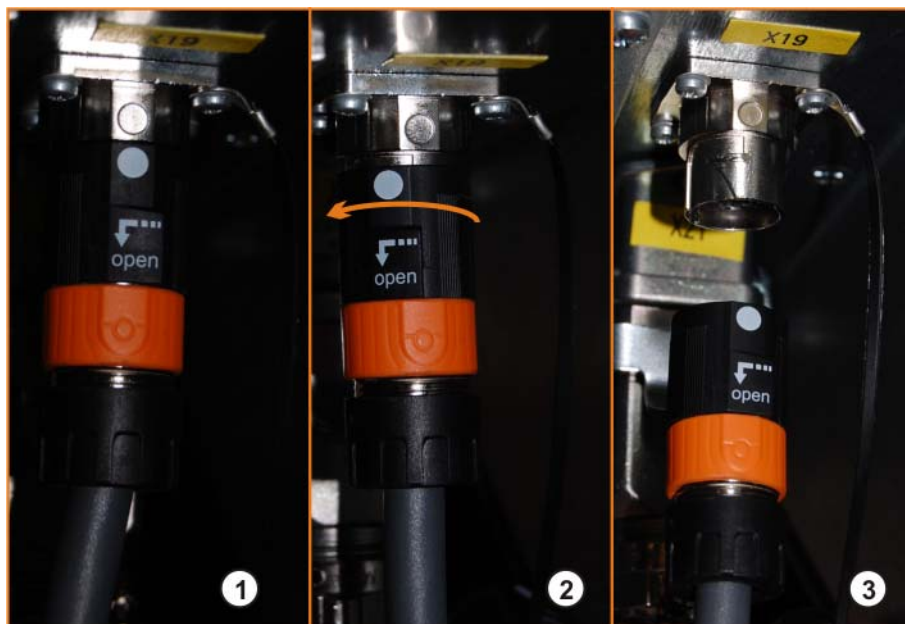


图 3-44: 取下 smartPAD

1	插头处于插接状态
2	沿箭头方向将上部的黑色部件旋转约 25°
3	向下拔出插头

4. 关闭配电箱门 (V)KR C4

注意

如果在计时器计时期间没有拔下 smartPAD，则此次计时失效。
可任意多次按下用于拔下的按钮，以再次显示计时器。

插入：

1. 确保使用相同规格的 smartPAD
2. 打开配电箱门 (V)KR C4
3. 插入 smartPAD 插头

注意

注意插口和 smartPAD 插头的标记

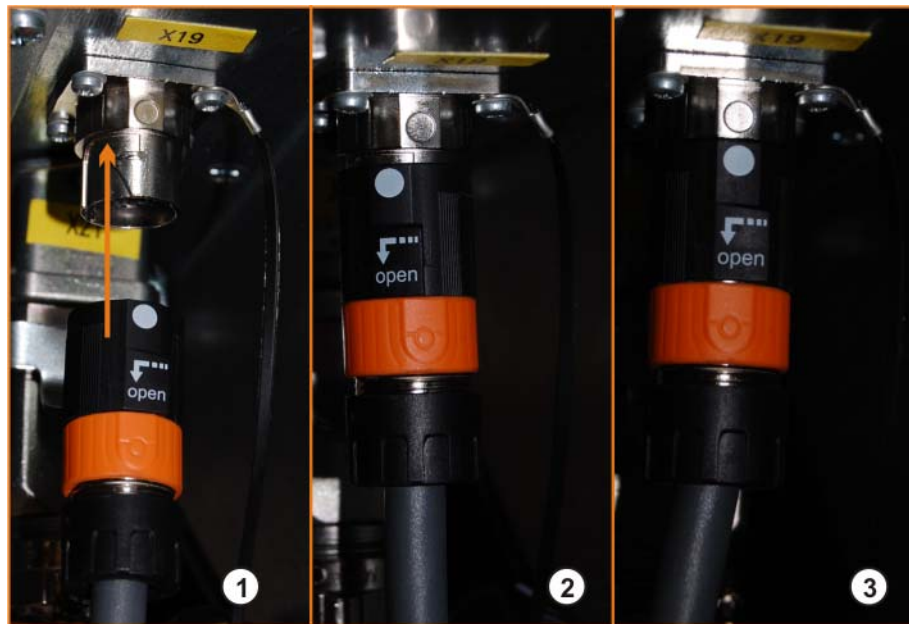


图 3-45: 插上 smartPAD

1	插头处于拔下状态（注意标记）
2	向上推插头。向上推时，上部的黑色部件自动旋转约 25°
3	插头自动卡止，标记即相对

4.

警告 操作人员将 smartPAD 插在机器人控制器上后，至少必须在 smartPAD 旁停留 30 秒，就是说直到紧急停止和确认开关再次恢复正常功能。这样就避免出现例如另一操作人员在紧急情况下使用的紧急停止装置暂时无效的情况。

5. 关闭配电箱门 (V)KR C4

4 执行机器人程序

4.1 执行初始化运行

BCO 运行

KUKA 机器人的初始化运行称为 BCO 运行。

i BCO 是 **Blockcoincidence** (即程序段重合) 的缩写。重合意为“一致”及“时间 / 空间事件的会合”。

在下列情况下要进行 BCO 运行：

- 选择程序 (例 1)
- 程序复位：复位 (例 1)
- 程序执行时手动移动 (例 1)
- 更改程序 (例 2)
- 语句行选择 (例 3)

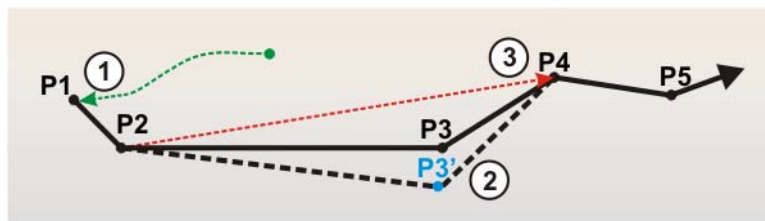


图 4-1: BCO 运行的原因举例

BCO 运行执行举例

- 1 选定程序或程序复位后 BCO 运行至原始位置
- 2 更改了运动指令后执行 BCO 运行 删除、示教了点后
- 3 进行了语句行选择后执行 BCO 运行

BCO 运行的原因

为了使当前的机器人位置与机器人程序中的当前点位置保持一致，必须执行 BCO 运行。

仅当当前的机器人位置与编程设定的位置相同时才可进行轨迹规划。因此，首先必须将 TCP 置于轨迹上。

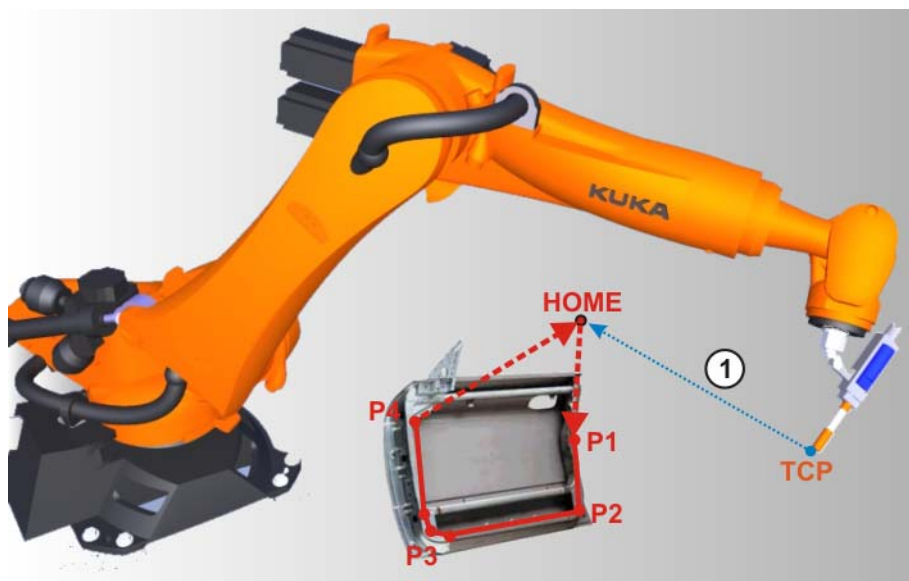


图 4-2: BCO 运行范例

1 在选择或者复位程序后 BCO 运行至 Home 位置

4.2 选择和启动机器人程序

选择和启动机器人程序

如果要执行一个机器人程序，则必须事先将其选中。机器人程序在导航器中的用户界面上供选择。通常，在文件夹中创建移动程序。Cell 程序（由 PLC 控制机器人的管理程序）始终在文件夹“R1”中。

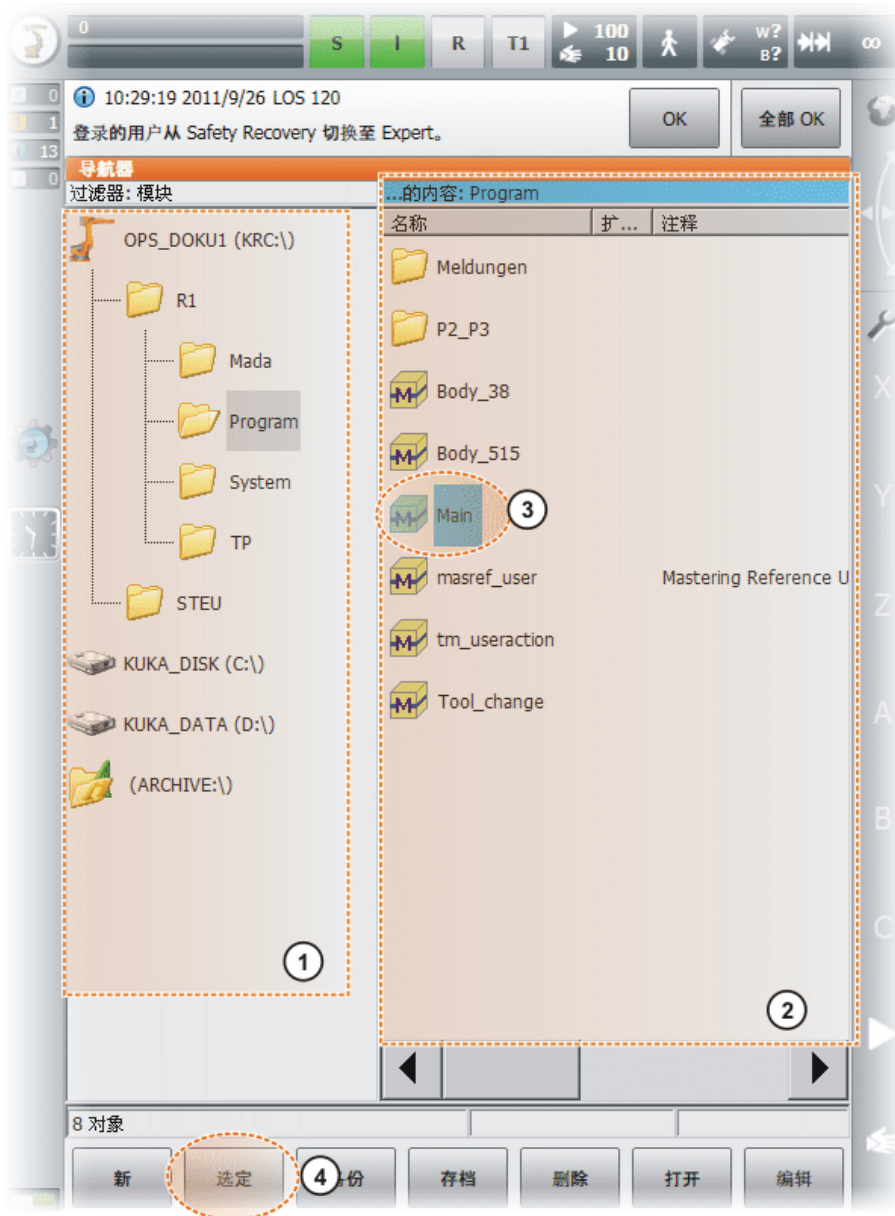


图 4-3: 导航器

- 1 导航器：文件夹 / 硬盘结构
- 2 导航器：文件夹 / 数据列表
- 3 选中的程序
- 4 用于选择程序的按键






对于程序启动，有启动正向运行程序按钮  和启动反向运行程序按钮  供选择



图 4-4: 程序运行方向：向前 / 向后

如果运行某个程序，则对于编程控制的机器人运动，可提供多种**程序运行方式**：

	<p>GO</p> <ul style="list-style-type: none"> ■ 程序连续运行，直至程序结尾。 ■ 在测试运行中必须按住启动键。
	<p>MSTEP</p> <ul style="list-style-type: none"> ■ 在运动步进运行方式下，每个运动指令都单个执行 ■ 每一个运动结束后，都必须重新按下“启动”键。
	<p>ISTEP 仅供用户组“专家”使用！</p> <ul style="list-style-type: none"> ■ 在增量步进时，逐行执行（与行中的内容无关） ■ 每行执行后，都必须重新按下启动键。

机器人程序看起来
是怎样的？

```

1 DEF kuka_rocks( )
2 INI
3 PTP HOME Vel= 100 % DEFAULT
4 PTP P1 Vel=100 % PDAT1 Tool[1] Base[0]
5 PTP P2 Vel=100 % PDAT2 Tool[1] Base[0]
6 PTP P3 Vel=100 % PDAT3 Tool[1] Base[0]
7 OUT 1'' State=TRUE CONT
8 LIN P4 Vel=2 m/s CPDAT1 Tool[1] Base[0]
9 PTP HOME Vel= 100 % DEFAULT
10 END

```

图 4-5: 机器人程序的结构

仅限于在专家用户组中可见：

- 1
 - “DEF 程序名 (/)” 始终出现在程序开头
 - “END” 表示程序结束
- 2
 - “INI” 行包含程序正确运行所需的标准参数的调用。
 - “INI” 行必须最先运行！
 - 自带的程序文本，包括运动指令、等待 / 逻辑指令等
- 3
 - 行驶指令“PTP Home” 常用于程序开头和末尾，因为这是唯一的已知位置。

程序状态

图标	颜色	说明
	灰色	未选定程序。
	黄色	语句指针位于所选程序的首行。
	绿色	已经选择程序，而且程序正在运行。
	红色	选定并启动的程序被暂停。
	黑色	语句指针位于所选程序的末端。

启动程序

启动机器人程序的操作步骤：

1. 选择程序



图 4-6: 选择程序

2. 设定程序速度 (程序倍率, POV)

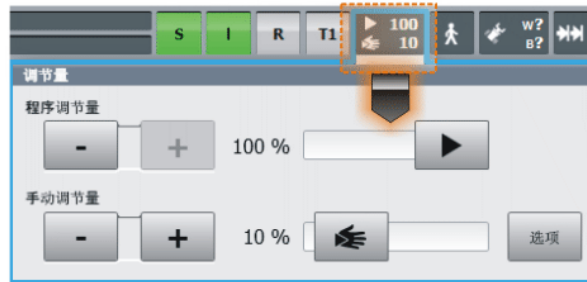


图 4-7: POV 设置

3. 按确认键

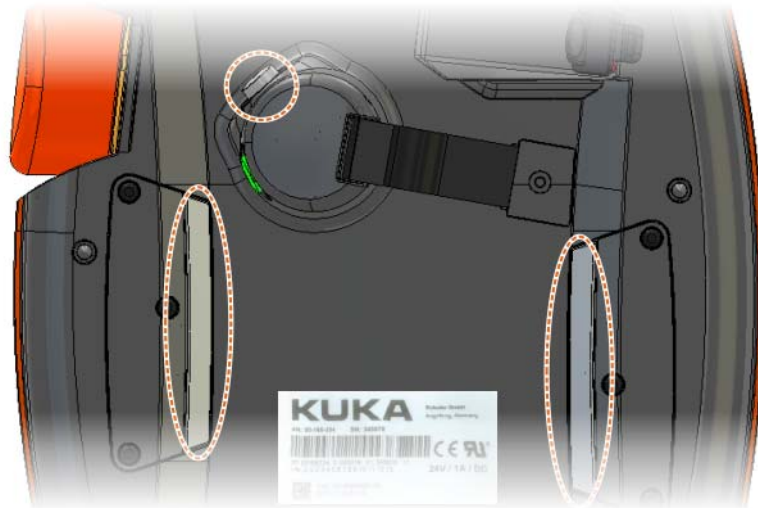


图 4-8: 确认键

4. 按下启动键 (+) 并按住 :

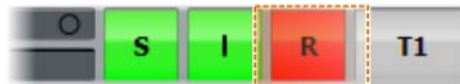
- “INI” 行得到处理。
- 机器人执行 BCO 运行。



图 4-9: 程序运行方向：向前 / 向后

警告 如果选定的运动语句包括 PTP 运行指令，则 BCO 运行将作为 PTP 运动从实际位置移动到目标位置。如果选定的运动语句包括 LIN 或 CIRC，则 BCO 运行将作为 LIN 运动被执行。观察此运动，防止碰撞。在 BCO 运行中速度自动降低。

5. 到达目标位置后运动停止。



将显示提示信息“已达 BCO”。

6. 其他流程（根据设定的运行方式）：

- T1 和 T2：通过按启动键继续执行程序。
- AUT：激活驱动装置。



接着按动 Start（启动）启动程序。

- 在 Cell 程序中将运行方式调为 EXT 并由 PLC 传送运行指令。

4.3 练习：执行机器人程序

练习目的

成功完成此练习后，您可执行下列操作：

- 选择和取消程序
- 以要求的运行方式运行、停止以及复位程序（测试程序流程）
- 执行并理解语句选择
- 执行 BCO 运行

前提条件

为成功完成此练习，必须满足以下前提条件：

- 具备使用导航器的理论知识
- 具备选择和取消程序的知识

练习内容

1. 选择模块 Air



危险！
请务必遵守用户指导中的安全规范！

2. 如下在不同的运行方式下测试程序：

- T1 以 100%
- T2 以 10%、30%、50%、75%、100%
- 自动 以 100%

3. 在程序运行方式 Go 和 MSTEP 下测试程序

5 程序文件的使用

5.1 创建程序模块

导航器中的程序模块

编程模块应始终保存在文件夹“Program”（程序）中。也可建立新的文件夹并将程序模块存放在那里。模块用字母“M”标示。一个模块中可以加入注释。此类注释中例如可含有程序的简短功能说明。

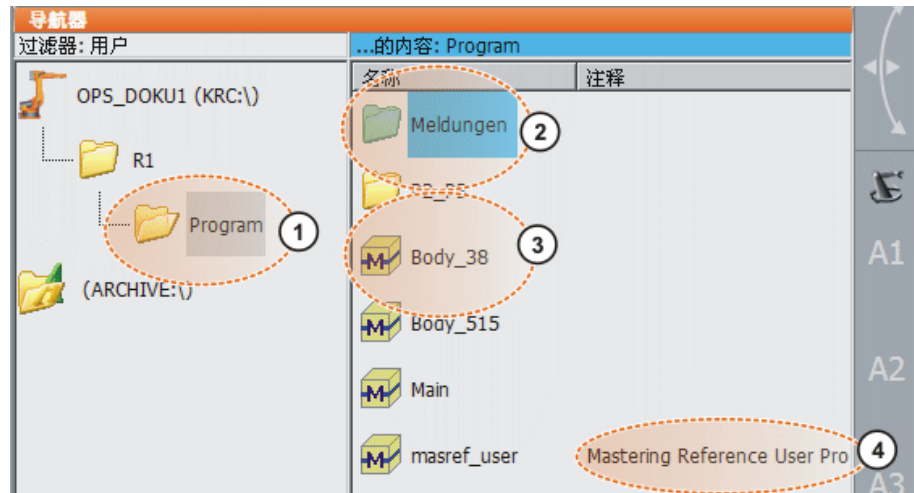


图 5-1: 导航器中的模块

- 1 程序的主文件夹：“程序”
- 2 其他程序的子文件夹
- 3 程序模块 / 模块
- 4 程序模块的注释

程序模块的属性

模块由两个部分组成：

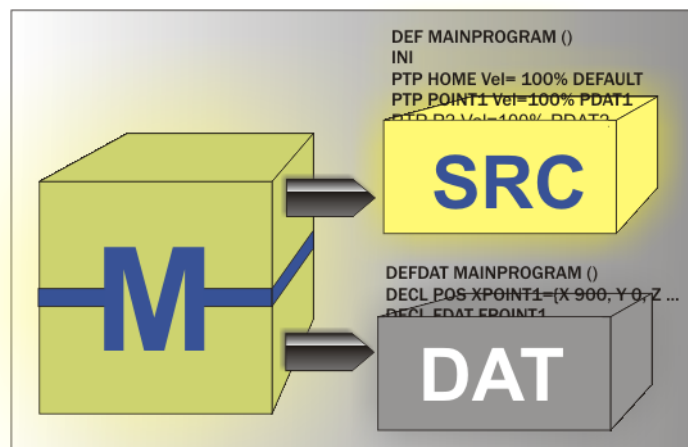


图 5-2

- 源代码：SRC 文件中含有程序源代码。

```

DEF MAINPROGRAM ()
INI
PTP HOME Vel= 100% DEFAULT
PTP POINT1 Vel=100% PDAT1 TOOL[1] BASE[2]
PTP P2 Vel=100% PDAT2 TOOL[1] BASE[2]
...
END

```

- **数据列表**：DAT 文件中含有固定数据和点坐标。

```

DEFDAT MAINPROGRAM ()
DECL E6POS XPOINT1={X 900, Y 0, Z 800, A 0, B 0, C 0, S 6, T 27, E1
0, E2 0, E3 0, E4 0, E5 0, E6 0}
DECL FDAT FPOINT1 ...
...
ENDDAT

```

创建编程模块的操作步骤

1. 在目录结构中选定要在其中建立程序的文件夹，例如文件夹**程序**，然后切换到文件列表。
2. 按下软键**新建**。
3. 输入程序名称，需要时再输入注释，然后按 **OK** 确认。

5.2 编辑程序模块

编辑方式

与常见的文件系统类似，也可以在 KUKS 导航器中编辑 smartPad 程序模块。

编辑方式包括：

- 复制
- 删除
- 重命名

程序删除的操作步骤

1. 在文件夹结构中选中文件所在的文件夹。
2. 在文件列表中选中文件。
3. 选择软键 **删除 >**。
4. 点击**是**确认安全询问。模块即被删除。



在用户组“专家”和筛选设置“详细信息”中，每个模块各有两个文件映射在导航器中（SRC 和 DAT 文件）。如果属实，则必须删除这两个文件！已删除的文件无法恢复！

程序改名的操作步骤

1. 在文件夹结构中选中文件所在的文件夹。
2. 在文件列表中选中文件。
3. 选择软键 **编辑 > 改名**。
4. 用新的名称覆盖原文件名，并用 **OK** 确认。



在用户组“专家”和筛选设置“详细信息”中，每个模块各有两个文件映射在导航器中（SRC 和 DAT 文件）。如果属实，则必须给这两个文件改名！

程序复制的操作步骤

1. 在文件夹结构中选中文件所在的文件夹。
2. 在文件列表中选中文件。
3. 选择软键 **复制**。
4. 给新模块输入一个新文件名，然后用 **OK** 确认。



在用户组“专家”和筛选设置“详细信息”中，每个模块各有两个文件映射在导航器中（SRC 和 DAT 文件）。如果属实，则必须复制这两个文件！

5.3 存档和还原机器人程序

存档途径

在每个存档过程中均会在相应的目标媒质上生成一个 ZIP 文件，该文件与机器人同名。在**机器人数据**下可个别改变文件名。

存储位置：有三个不同的存储位置可供选择：

- **USB (KCP)** | KCP(smardPAD) 上的 U 盘
- **USB (控制柜)** | 机器人控制柜上的 U 盘
- **网络** | 在一个网络路径上存档
所需的网络路径必须在**机器人数据**下配置。



在每个存档过程中，除了将生成的 ZIP 文件保存在所选的存储媒质上之外，同时还在驱动器 D:\ 上储存一个存档文件 (INTERN.ZIP)。

数据：可选择以下数据存档：

- **全部**：
将还原当前系统所需的数据存档。
- **应用**：
所有用户自定义的 KRL 模块（程序）和相应的系统文件均被存档。
- **机器参数**：
将机器参数存档。
- **Log 数据**：
将 Log 文件存档。
- **KrcDiag**：
将数据存档，以便将其提供给库卡机器人有限公司进行故障分析。在此将生成一个文件夹（名为 **KRCDiag**），其中可写入十个 ZIP 文件。除此之外还另外在控制器中将存档文件存放在 C:\KUKA\KRCDiag 下。

还原数据



警告 通常情况下，只允许载入具有相应软件版本的文档。如果载入其它文档，则可能出现以下后果：

- 故障信息
- 机器人控制器无法运行。
- 人员受伤以及财产损失

还原时可选择以下菜单项：

- **全部**
- **应用程序**
- **配置**



系统会在以下情况时发出故障信息：

- 已存档文件版本与系统中的文件版本不同时。
- 工艺程序包的版本与已安装的版本不一致时。

存档操作步骤



注意 仅允许使用 KUKA.USBData U 盘。如果使用其他 U 盘，则可能造成数据丢失或数据被更改。

1. 选择菜单序列 **文件 > 存档 > USB (KCP)** 或者 **USB (控制柜)** 以及所需的选项。
2. 点击**是**确认安全询问。
当存档过程结束时，将在信息窗口中显示出来。
3. 当 U 盘上的 LED 指示灯熄灭之后，可将其取下。

还原的操作步骤

1. 打开菜单序列：**文件 > 还原 >**，然后选择所需的子项
2. 点击**是**确认安全询问。已存档的文件在机器人控制系统里重新恢复。当恢复过程结束时，屏幕出现相关的消息。
3. 如果已从 U 盘完成还原：**拔出 U 盘。**

注意

如果正从 U 盘执行还原：只有当 U 盘上的 LED 熄灭之后方可拔出 U 盘。否则会导致 U 盘受损。

4. 重新启动机器人控制系统。

5.4 通过运行日志了解程序和状态变更

记录可能性

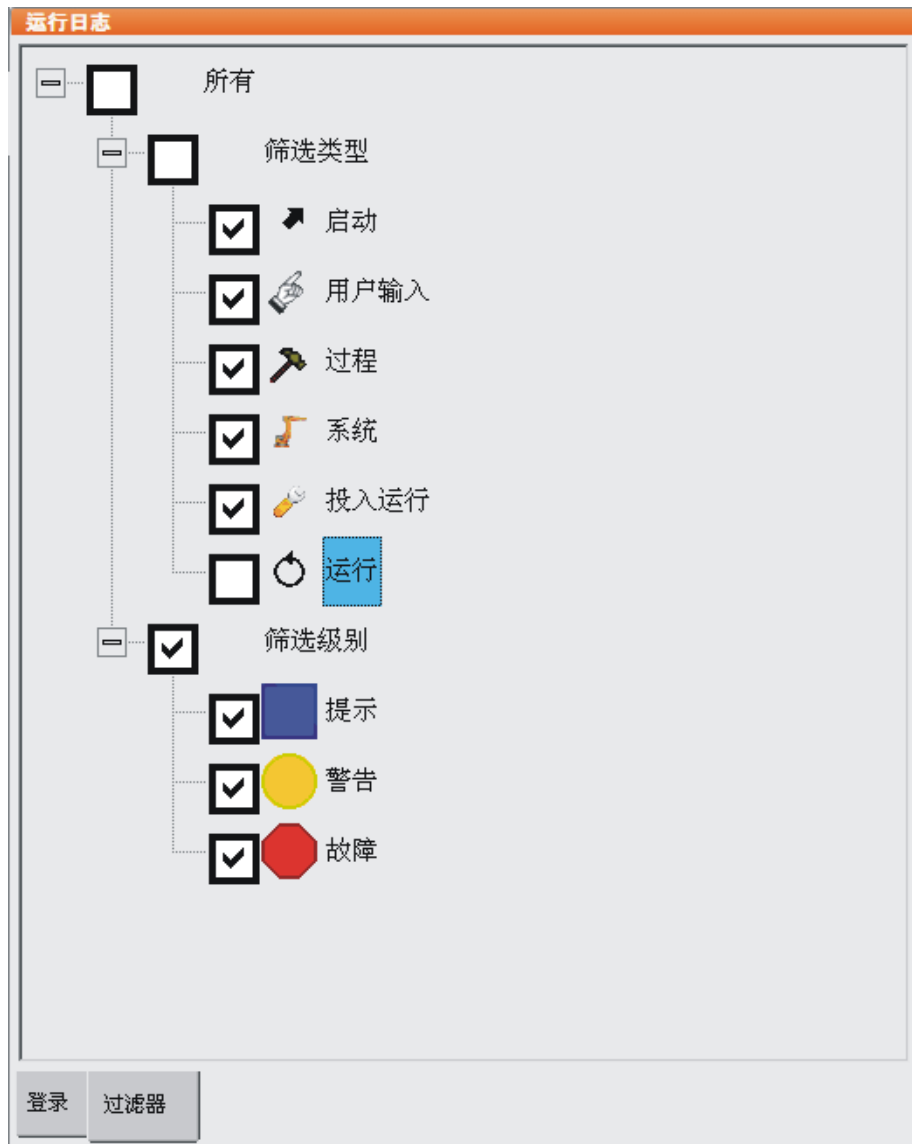
用户在 smartPAD 上的操作过程会被自动记录下来。指令**运行日志**用于显示记录。



图 5-3: 运行日志，选项卡 Log

序号	说明
1	日志事件的类型 各个筛选类型和筛选等级均列在选项卡 筛选器 中。
2	日志事件的编号
3	日志事件的日期和时间
4	日志事件的简要说明
5	所选日志事件的详细说明
6	显示有效的筛选器

Log 事件的筛选

图 5-4: 运行日志，选项卡**过滤器**

使用运行日志功能

在每个用户组中都可以查看和配置。

显示运行日志：

- 在主菜单中选择**诊断 > 运行日志 > 显示**。

配置运行日志：

1. 在主菜单中选择**诊断 > 运行日志 > 配置**。
2. 设置：
 - 添加 / 删除筛选类型
 - 添加 / 删除筛选级别
3. 按下 **OK** 以保存配置，然后关闭该窗口。

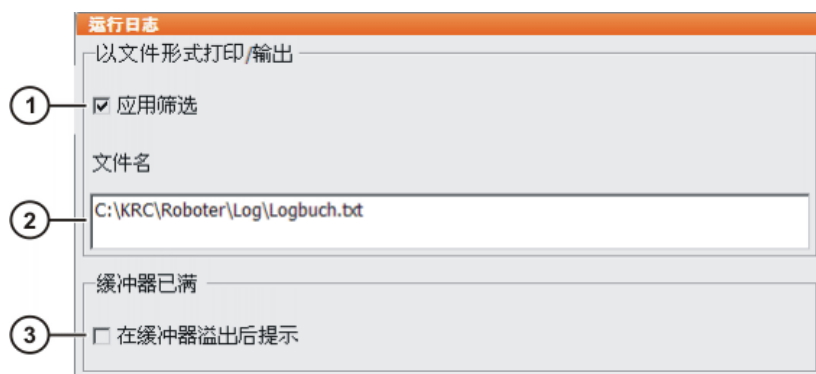


图 5-5: 窗口配置运行日志

- 1 将筛选设置应用到输出端。如果不勾选，则在输出时不会进行筛选。
- 2 文本文件路径。
- 3 已因缓冲溢出而删除的日志数据会以灰色阴影格式显示在文本文件中

6 建立及更改编程的运动

6.1 创建新的运动指令

对机器人运动进行编程

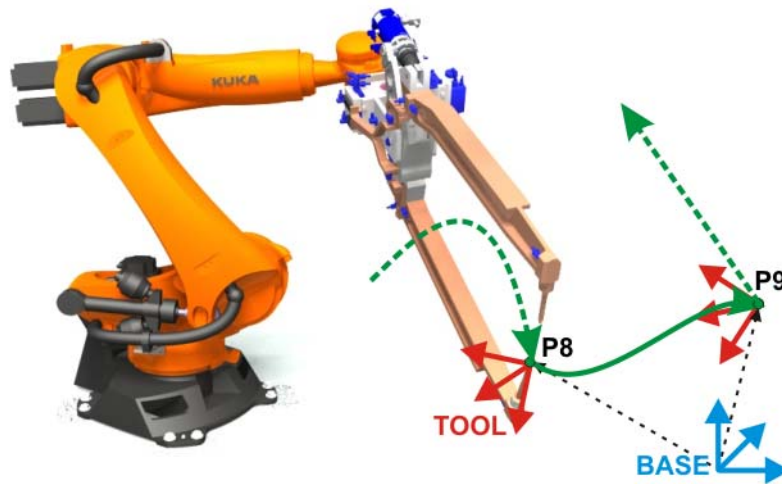


图 6-1: 机器人运动

如果必须对机器人运动进行编程，需要解决以下诸多问题：

问题	方案	关键词
机器人如何记住其位置？	工具在空间中的相应位置会被保存。（机器人位置对应于所设定的工具坐标和基坐标）	POS
机器人如何知道它必须如何运动？	通过指定运动方式：点到点，直线或者圆形。	PTP LIN CIRC
机器人运动的速度有多快？	两点之间的速度和加速度可通过编程设定。	Vel. Acc.
机器人是否必须在每个点上都要停住？	为了缩短节拍时间，点也可以轨迹逼近，但这样就不会精确暂停。	CONT
如要到达某个点，工具会沿哪个方向？	可以针对每个运动对姿态引导进行单独设置。	ORI_TYPE
机器人是否会识别障碍？	不会，机器人只会“坚定不移”地沿编程设定的轨迹运动。程序员要负责保证移动时不会发生碰撞。 但也有用于保护机器的“碰撞监控”方式。	碰撞监控

用示教方式对机器人运动进行编程时必须传输这些信息。为此应使用联机表格，在该表格中可以很方便地输入这些信息。

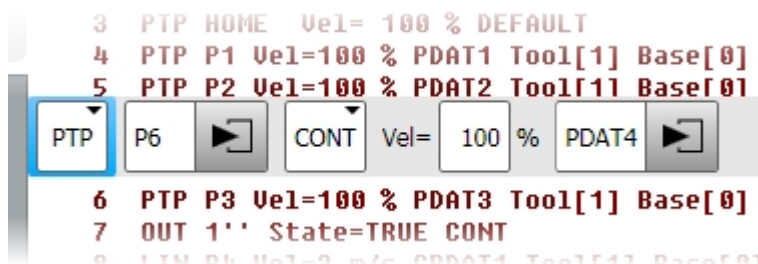


图 6-2: 运动编程的联机表格

运动方式

有不同的运动方式供运动指令的编程使用。可根据对机器人工作流程的要求来进行运动编程。

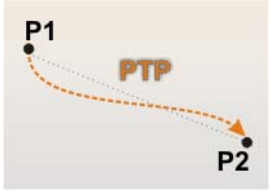
- 按轴坐标的运动（PTP: Point-To-Point，即点到点）
- 沿轨迹的运动：LIN（线性）和 CIRC（圆周形）
- SPLINE：样条是一种尤其适用于复杂曲线轨迹的运动方式。这种轨迹原则上也可以通过 LIN 运动和 CIRC 运动生成，但是样条更有优势。



样条运动不包括在本培训资料之内。详细信息可查阅 KUKA 系统软件 8.2 的操作及编程指南。

6.2 创建已优化节拍时间的运动（轴运动）

PTP

运动方式	含义	应用举例
	<p><i>Point-To-Point</i>：点到点</p> <ul style="list-style-type: none"> ■ 按轴坐标的运动：机器人将 TCP 沿最快速轨迹送到目标点。最快速的轨迹通常并不是最短的轨迹，因而不是直线。由于机器人轴的旋转运动，因此弧形轨迹会比直线轨迹更快。 ■ 运动的具体过程不可预见。 ■ 导向轴是达到目标点所需时间最长的轴。 ■ SYNCHRO PTP: 所有轴同时启动并且也同步停下。 ■ 程序中的第一个运动必须为 PTP 运动，因为只有在此运动中才评估状态和转向。 	<p>点到点运动的应用举例：</p> <ul style="list-style-type: none"> ■ 点焊 ■ 运输 ■ 测量，检验 <p>辅助位置：</p> <ul style="list-style-type: none"> ■ 位于中间点 ■ 空间中的自由点

轨迹逼近

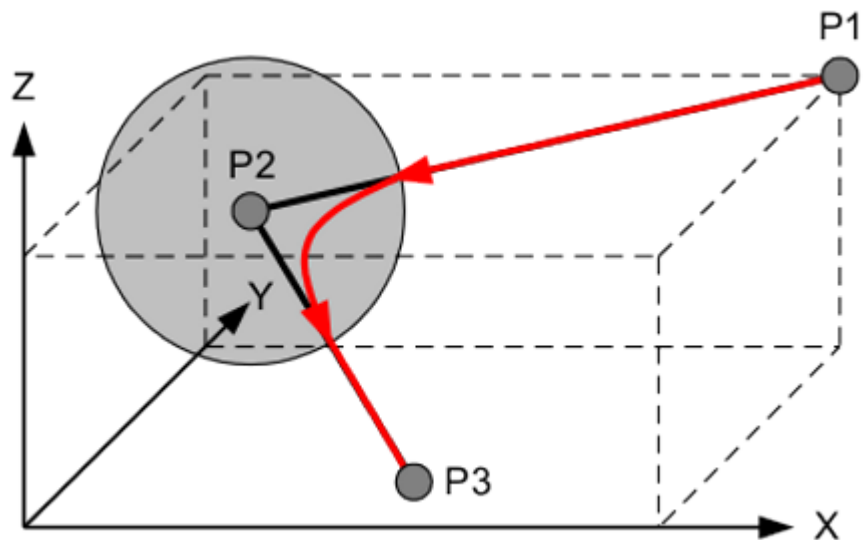


图 6-3: 轨迹逼近点

为了加速运动过程，控制器可以 CONT 标示的运动指令进行轨迹逼近。轨迹逼近意味着将不精确移到点坐标。事先便离开精确保持轮廓的轨迹。TCP 被引导沿着轨迹逼近轮廓运行，该轮廓止于下一个运动指令的精确保持轮廓。

轨迹逼近的优点

- 减少磨损
- 降低节拍时间

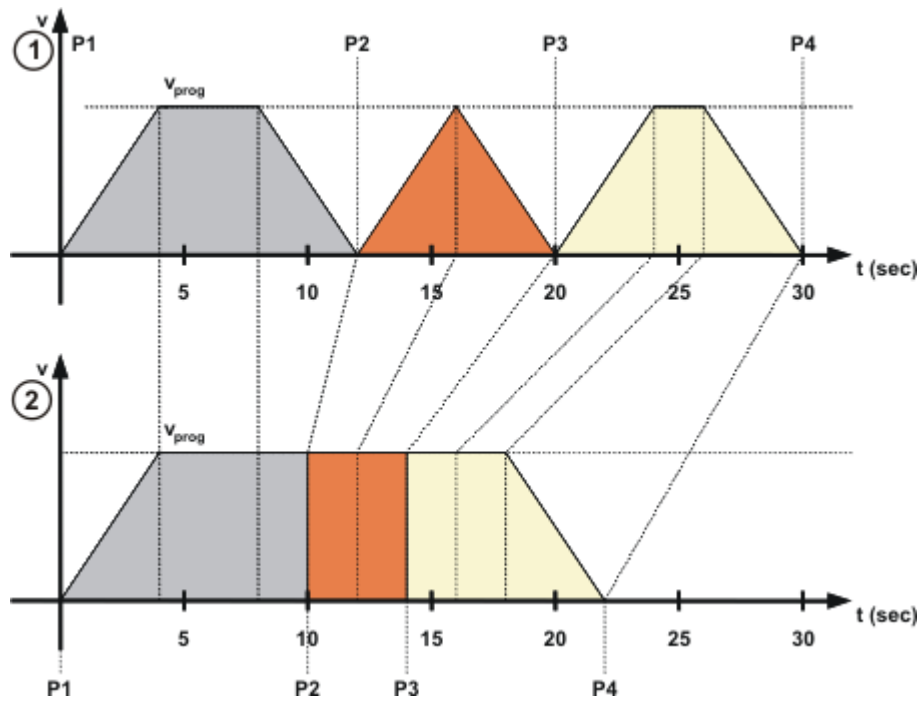
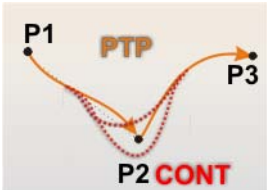


图 6-4: 比较精确暂停和轨迹逼近

为了能够执行轨迹逼近运动，控制器必须能够读入以下运动语句。通过计算机预进读入。

运动方式 PTP 中的轨迹逼近

运动方式	特征	轨迹逼近距离
	<ul style="list-style-type: none"> ■ 轨迹逼近不可预见。 	以 % 表示

创建 PTP 运动的操作步骤

前提条件

- 已设置运行方式 T1
 - 机器人程序已选定。
1. 将 TCP 移向应被示教为目标点的位置。

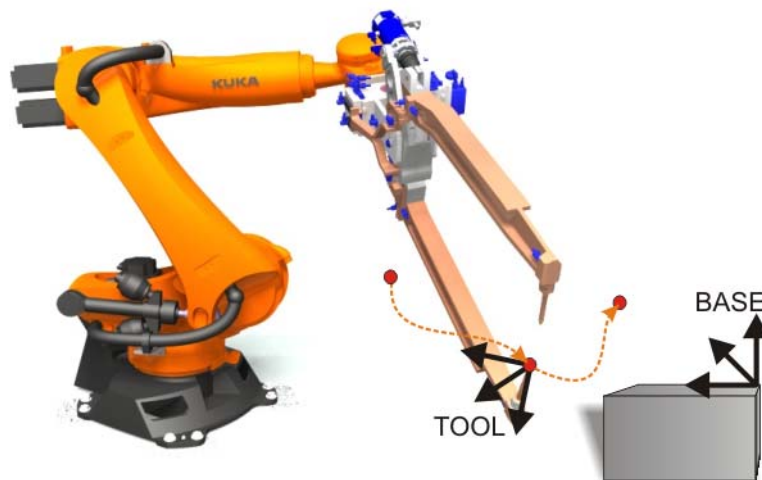


图 6-5: 运动指令

2. 将光标置于其后应添加运动指令的那一行中。
3. 菜单序列指令 > 运动 > PTP
作为选项，也可在相应行中按下软件运动。

联机表格出现：

■ PTP 联机表格

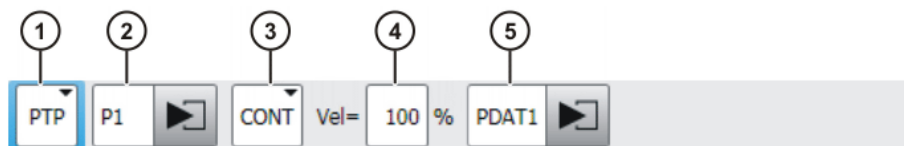


图 6-6: PTP 运动的联机表格

4. 在联机表格中输入参数。

序号	说明
1	运动方式 PTP 、 LIN 或者 CIRC
2	目标点的名称自动分配，但可予以单独覆盖。 触摸箭头以编辑点数据，然后选项窗口 Frames 自动打开 对于 CIRC ，必须为目标点额外示教一个辅助点。移向辅助点位置，然后按下 Touchup HP 。
3	<ul style="list-style-type: none"> ■ CONT：目标点被轨迹逼近。 ■ [空白]：将精确地移至目标点。
4	速度 <ul style="list-style-type: none"> ■ PTP 运动：1 ... 100 % ■ 沿轨迹的运动：0.001 ... 2 m/s
5	运动数据组： <ul style="list-style-type: none"> ■ 加速度 ■ 轨迹逼近距离（如果在栏 (3) 中输入了 CONT） ■ 姿态引导（仅限于沿轨迹的运动）

5. 在选项窗口 **Frames** 中输入工具和坐标系的正确数据，以及关于插补模式的数据（外部 TCP：开/关）和碰撞监控的数据。



图 6-7: 帧选项窗口

序号	说明
1	选择工具。 如果外部 TCP 栏中显示 True : 选择工具。 值域 : [1] ... [16]
2	选择基准。 如果外部 TCP 栏中显示 True : 选择固定工具。 值域 : [1] ... [32]
3	插补模式 <ul style="list-style-type: none"> False : 该工具已安装在连接法兰上。 True : 该工具为固定工具。
4	<ul style="list-style-type: none"> True : 机器人控制系统为此运动计算轴的扭矩。此值用于碰撞识别。 False : 机器人控制系统为此运动不计算轴的扭矩。因此对此运动无法进行碰撞识别。

6. 在运动参数选项窗口中可将加速度从最大值降下来。如果已经激活轨迹逼近, 则也更改轨迹逼近距离。根据配置的不同, 该距离的单位可以设置为 mm 或 %。

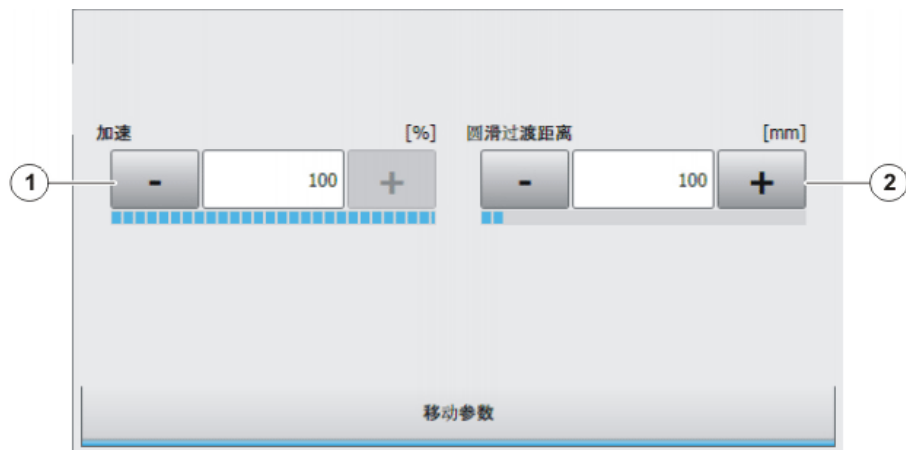


图 6-8: 运行参数选项窗口 (PTP)

序号	说明
1	加速度 以机器数据中给出的最大值为基准。此最大值与机器人类型和所设定的运行方式有关。该加速度适用于该运动语句的主要轴。 <ul style="list-style-type: none"> 1 ... 100 %
2	只有在联机表格中选择了 CONT 之后, 此栏才显示。 离目标点的距离, 即最早开始轨迹逼近的距离。 最大距离: 从起点到目标点之间的一半距离, 以无轨迹逼近 PTP 运动的运动轨迹为基准。 <ul style="list-style-type: none"> 1 ... 100 % 1 ... 1000 mm

7. 用指令 **OK** 存储指令。TCP 的当前位置被作为目标示教。

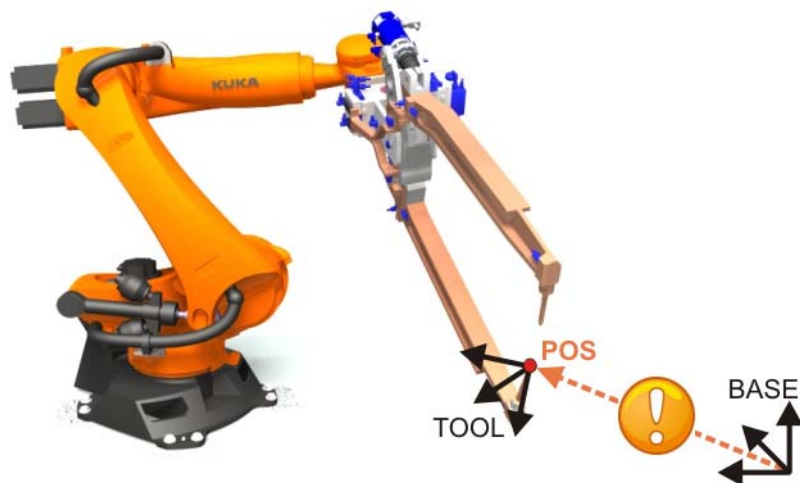


图 6-9: 在“指令 OK”和“Touchup”时保存点坐标

6.3 练习：空运转程序 - 程序操作和 PTP 运动

练习目的

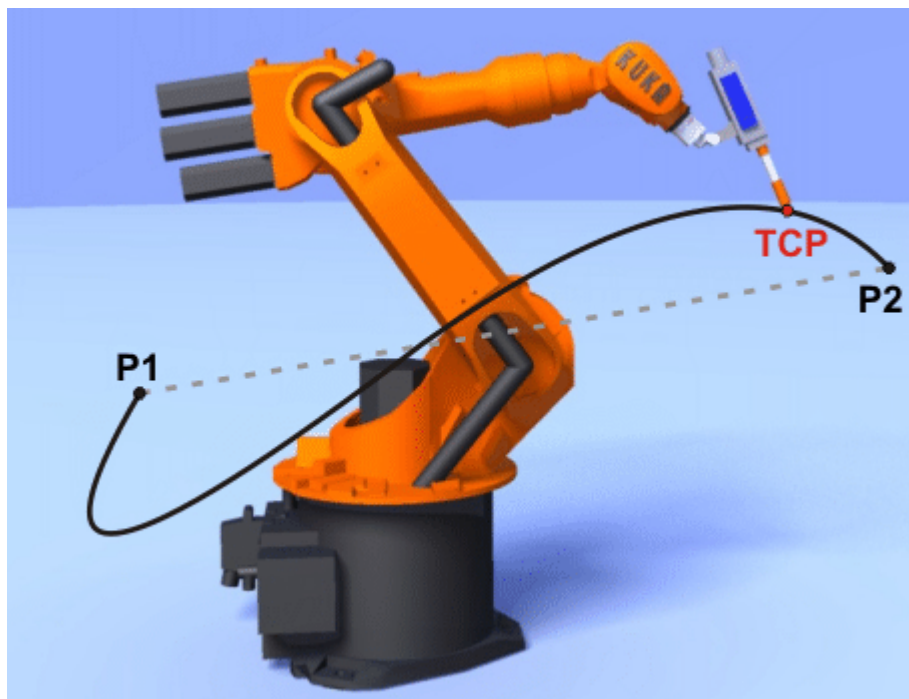
成功完成此练习后，您可执行下列操作：

- 选择和取消程序
- 以要求的运行方式运行、停止以及复位程序（测试程序流程）
- 删除运动语句并插入新的 PTP 运动
- 更改程序运行方式和逐步驶向编程设定的点
- 执行并理解语句选择
- 执行 BCO 行驶

前提条件

为成功完成此练习，必须满足以下前提条件：

- 具备使用导航器的理论知识
- 具备 PTP 运动方式的理论知识



练习内容

请完成以下任务：程序创建和测试

1. 创建名为 **Air_PROG** 的新模块



危险！
请务必遵守讲过的安全规程！

2. 请创建一组的五个 PTP 的运动语句
3. 如果无法无碰撞运行，则请删除相关的点并（分别）创建一个新的点
4. 请在 T1 运行模式下以不同的程序速度 (POV) 测试程序
5. 请在 T2 运行模式下以不同的程序速度 (POV) 测试程序
6. 请在自动运行模式下测试程序

练习内容 — B 部分

请完成以下任务：程序纠错

1. 请针对您的的空间点使用不同的速度
2. 请在程序中多次调用相同的点
3. 请删除运动语句，并在程序中的其它位置处重新插入新的点
4. 执行语句选择
5. 请在测试时暂停程序运行，并使用功能**反向启动程序**
6. 请在 T1、T2 和自动运行模式下测试程序

与练习配套的问题

1. 选择和打开程序之间的区别是什么？

.....

.....

2. 程序运行方式有哪些，各有什么用途？

.....

.....

.....

3. BCO 运行是什么？

.....

.....

.....

4. 如何才能影响程序速度？

.....

.....

.....

5. PTP 运动的特征是什么？

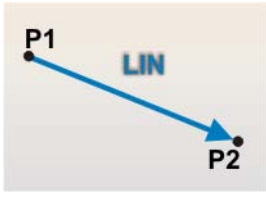
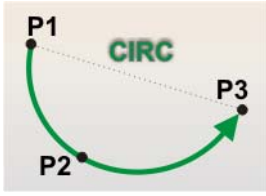
.....

.....

.....

6.4 创建沿轨迹的运动

LIN 和 CIRC

运动方式	含义	应用示例
	<p><i>Linear</i> : 直线</p> <ul style="list-style-type: none"> 直线型轨迹运动： 工具的 TCP 按设定的姿态从起点匀速移动到目标点。 速度和姿态均以 TCP 为参照点。 	<p>轨迹应用，例如：</p> <ul style="list-style-type: none"> 轨迹焊接 贴装 激光焊接 / 切割
	<p><i>Circular</i> : 圆形</p> <ul style="list-style-type: none"> 圆形轨迹运动是通过起点、辅助点和目标点定义的。 工具的 TCP 按设定的姿态从起点匀速移动到目标点。 速度和姿态均以 TCP 为参照点。 	<p>轨迹应用与 LIN 相同：</p> <ul style="list-style-type: none"> 圆周、半径、圆形

有着 6 个自由度的 KUKA 机器人具有 3 个不同的奇点位置。

奇点位置

即便在给定状态和步骤顺序的情况下，也无法通过逆向运算（将笛卡尔坐标转换成轴坐标值）得出唯一数值时，即可认为是一个奇点位置。这种情况下，或者当最小的笛卡尔变化也能导致非常大的轴角度变化时，即为奇点位置。奇点不是机械特性，而是数学特性，出于此原因，奇点只存在于轨迹运动范围内，而在轴运动时不存在。

顶置奇点 α_1

在顶置奇点位置时，腕点（即轴 A5 的中点）垂直于机器人的轴 A1。轴 A1 的位置不能通过逆向运算进行明确确定，且因此可以赋以任意值。

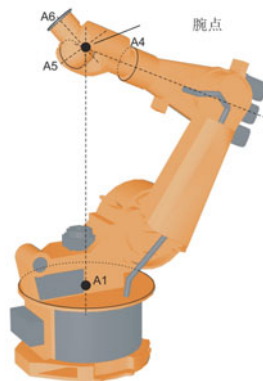


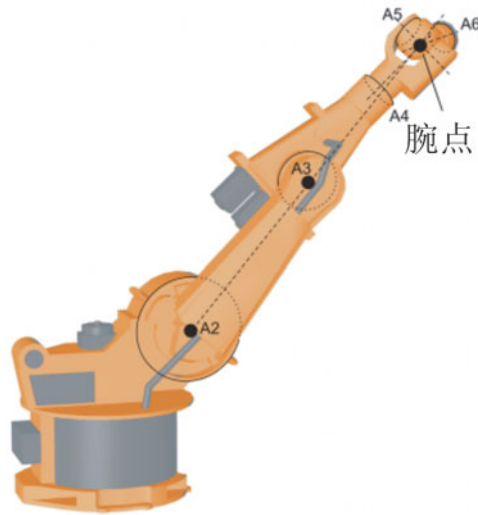
图 6-10: 过顶奇点 (α_1 位置)

延展位置奇点 α_2

对于延展位置奇点来说，腕点（即轴 A5 的中点）位于机器人轴 A2 和 A3 的延长线上。

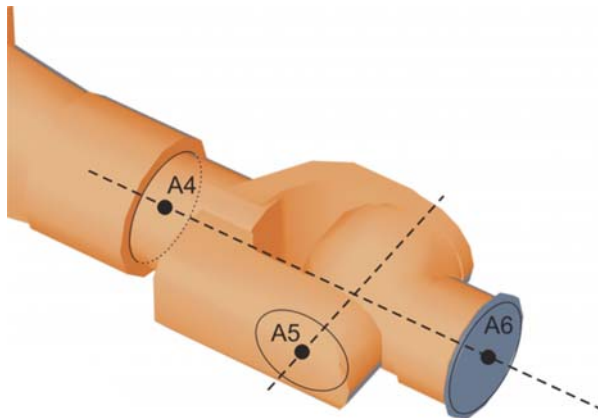
机器人处于其工作范围的边缘。

通过逆向运算将得出唯一的轴角度，但较小的笛卡尔速度变化将导致轴 A2 和 A3 较大轴速变化。

图 6-11: 延伸位置 (α_2 位置)手轴奇点 α_5

对于手轴奇点来说，轴 A4 和 A6 彼此平行，并且轴 A5 处于 $\pm 0.01812^\circ$ 的范围内。

通过逆向运算无法明确确定两轴的位置。轴 A4 和 A6 的位置可以有任意多的可能性，但其轴角度总和均相同。

图 6-12: 手轴奇点 (α_5 位置)沿轨迹的运动时的
姿态引导

沿轨迹的运动时可以准确定义姿态引导。工具在运动的起点和目标点处的姿态可能不同。

在运动方式 LIN 下的姿态引导

- 标准或手动 PTP

工具的姿态在运动过程中不断变化。

在机器人以标准方式到达手轴奇点时就可以使用手动 PTP，因为是通过手轴角度的线性轨迹逼近（按轴坐标的移动）进行姿态变化。

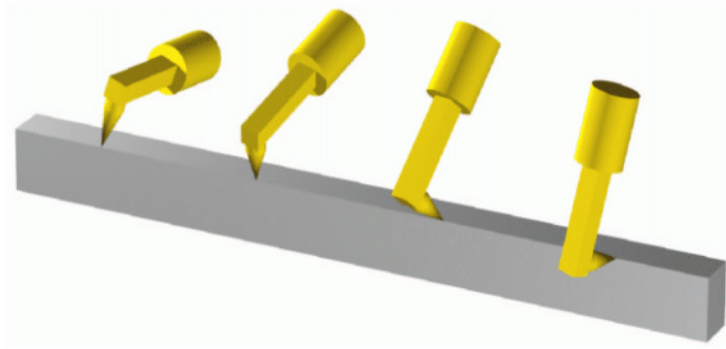


图 6-13: 标准

■ 固定不变

工具的姿态在运动期间保持不变，与在起点所示教的一样。在终点示教的姿态被忽略。

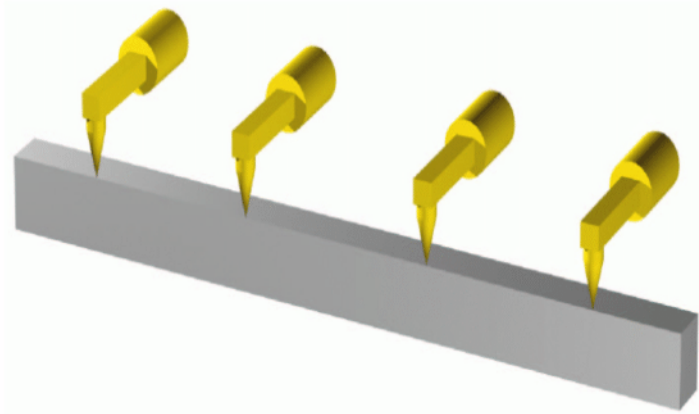


图 6-14: 稳定的方向导引

在运动方式 **CIRC** 下的姿态引导

■ 标准或手动 PTP

工具的姿态在运动过程中不断变化。

在机器人以标准方式到达手轴奇点时就可以使用手动 PTP，因为是通过手轴角度的线性轨迹逼近（按轴坐标的移动）进行姿态变化。

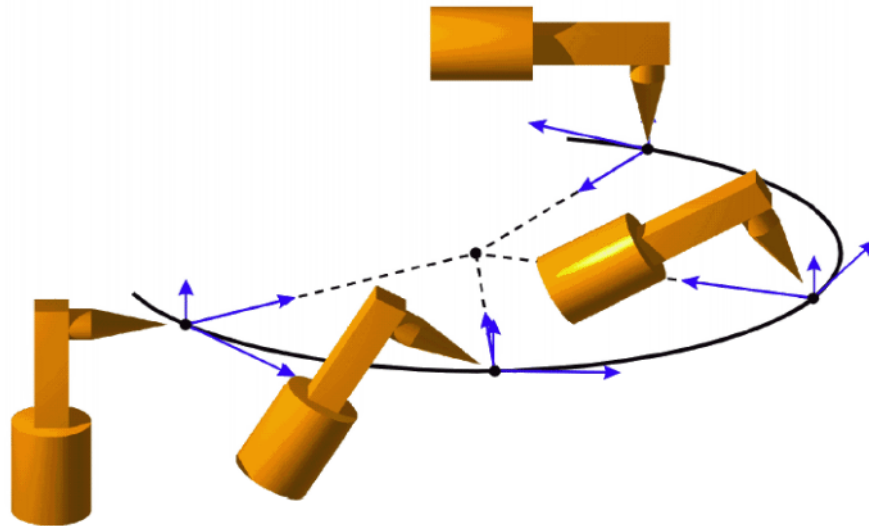


图 6-15: 标准 + 以基准为参照

- 固定不变

工具的姿态在运动期间保持不变，与在起点所示教的一样。在终点示教的姿态被忽略。

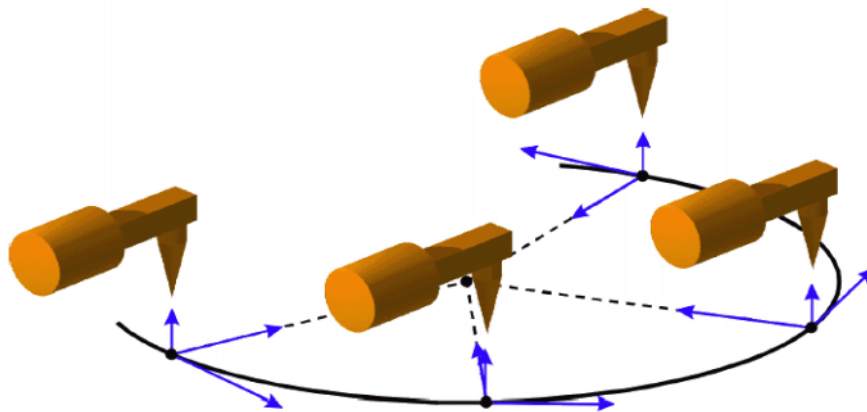


图 6-16: 恒定的方向导引 + 以基准为参照

轨迹运动的轨迹逼近

i 轨迹逼近功能不适用于生成圆周运动。它仅用于防止在某点出现精确暂停。

在运行方式 LIN 和 CIRC 下进行轨迹逼近

运动方式	特征	轨迹逼近距离
	<ul style="list-style-type: none"> ■ 轨迹相当于抛物线 	mm 数字
	<ul style="list-style-type: none"> ■ 轨迹相当于抛物线 	mm 数字

创建 LIN 和 CIRC 运动的操作步骤

前提条件

- 已设置运行方式 T1
 - 机器人程序已选定。
1. 将 TCP 移向应被示教为目标点的位置。

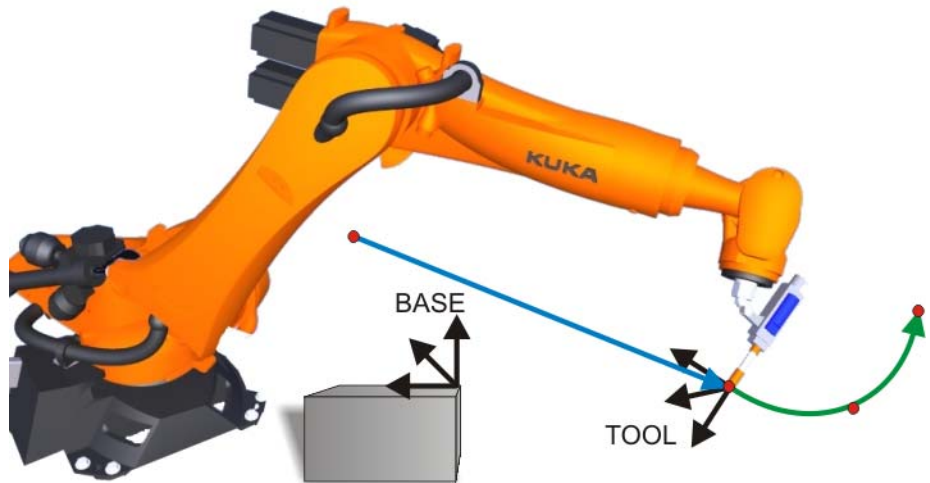


图 6-17: 运动指令 LIN 和 CIRC

2. 将光标置于其后应添加运动指令的那一行中。
3. 选择菜单序列 **指令 > 运动 > LIN** 或者 **CIRC**。
作为选项，也可在相应行中按下软件**运动**。

联机表格出现：

■ LIN 联机表格

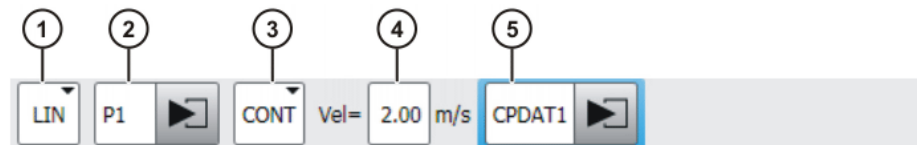


图 6-18: LIN 运动的联机表格

■ CIRC 联机表格

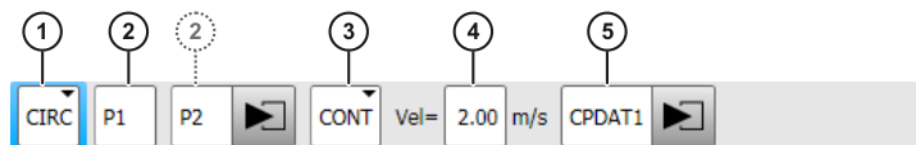


图 6-19: CIRC 运动联机表格

4. 在联机表格中输入参数。

序号	说明
1	运动方式 PTP 、 LIN 或者 CIRC
2	目标点的名称自动分配，但可予以单独覆盖。 触摸箭头以编辑点数据，然后选项窗口 Frames 自动打开 对于 CIRC ，必须为目标点额外示教一个辅助点。移向辅助点位置，然后按下 Touchup HP 。辅助点中的工具姿态无关紧要。
3	<ul style="list-style-type: none"> ■ CONT：目标点被轨迹逼近。 ■ [空白]：将精确地移至目标点。
4	速度 <ul style="list-style-type: none"> ■ PTP 运动：1 ... 100 % ■ 沿轨迹的运动：0.001 ... 2 m/s
5	运动数据组： <ul style="list-style-type: none"> ■ 加速度 ■ 轨迹逼近距离（如果在栏 (3) 中输入了 CONT） ■ 姿态引导（仅限于沿轨迹的运动）

5. 在选项窗口 Frames 中输入工具和基坐标的正确数据，以及关于插补模式的数据（外部 TCP：开/关）和碰撞监控的数据。



图 6-20: 帧选项窗口

序号	说明
1	选择工具。 如果外部 TCP 栏中显示 True ：选择工具。 值域：[1] ... [16]
2	选择基坐标。 如果外部 TCP 栏中显示 True ：选择固定工具。 值域：[1] ... [32]
3	插补模式 <ul style="list-style-type: none"> ■ False：该工具已安装在连接法兰上。 ■ True：该工具为固定工具。
4	<ul style="list-style-type: none"> ■ True：机器人控制系统为此运动计算轴的扭矩。此值用于碰撞识别。 ■ False：机器人控制系统为此运动不计算轴的扭矩。因此对此运动无法进行碰撞识别。

6. 在运动参数选项窗口中可将加速度从最大值降下来。如果轨迹逼近已激活，则可更改轨迹逼近距离。此外也可修改姿态引导。

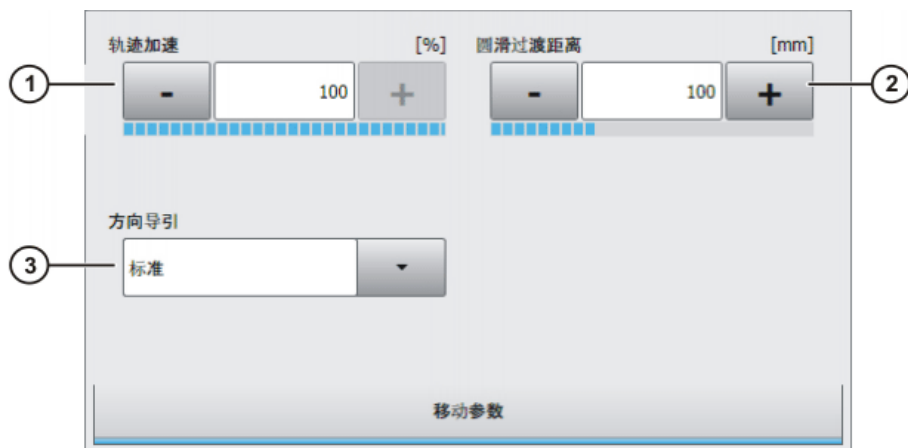


图 6-21: 选项窗口运动参数 (LIN, CIRC)

序号	说明
1	<p>加速度</p> <p>以机器数据中给出的最大值为基准。此最大值与机器人类型和所设定的运行方式有关。</p>
2	<p>至目标点的距离，最早在此处开始轨迹逼近</p> <p>此距离最大可为起始点至目标点距离的一半。如果在此处输入了一个更大数值，则此值将被忽略而采用最大值。</p> <p>只有在联机表格中选择了 CONT 之后，此栏才显示。</p>
3	<p>选择姿态引导。</p> <ul style="list-style-type: none"> ■ 标准 ■ 手动 PTP ■ 稳定的姿态引导 <p>(>>> "沿轨迹的运动时的姿态引导" 页码 103)</p>

7. 用指令 **OK** 存储指令。TCP 的当前位置被作为目标示教。

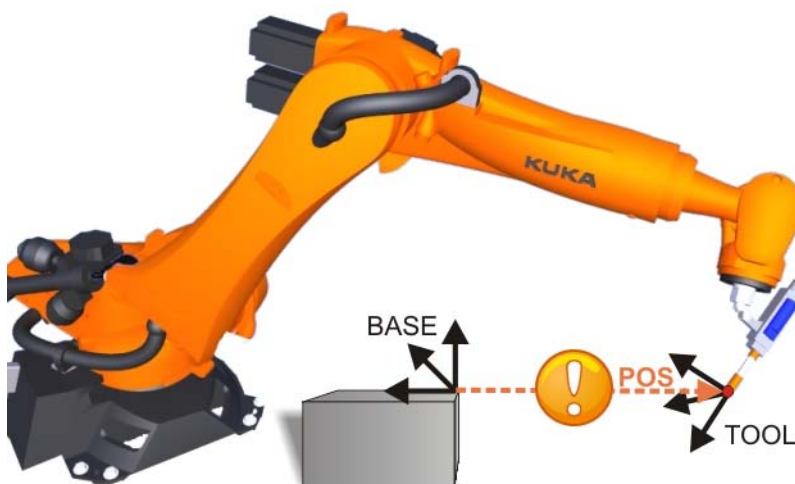


图 6-22: 在“指令 OK”和“Touchup”时保存点坐标

6.5 更改运动指令

更改运动指令

更改现有运动指令的原因有多种：

典型原因	待执行的更改
待抓取工件的位置发生变化。 加工时五个孔中的一个孔位置发生变化。 焊条必须截短。	位置数据的更改
货盘位置发生变化。	更改帧数据：基坐标系和 / 或工具坐标系
意外使用了错误基坐标系对某个位置进行了示教。	更改帧数据：带位置更新的基坐标系和 / 或工具坐标系
加工速度太慢：节拍时间必须改善。	更改运动数据：速度、加速度 更改运动方式

更改运动指令的作用

更改位置数据

- 只更改点的数据组：点获得新的坐标，因为已用“Touchup”更新了数值。旧的点坐标被覆盖，并且不再提供！

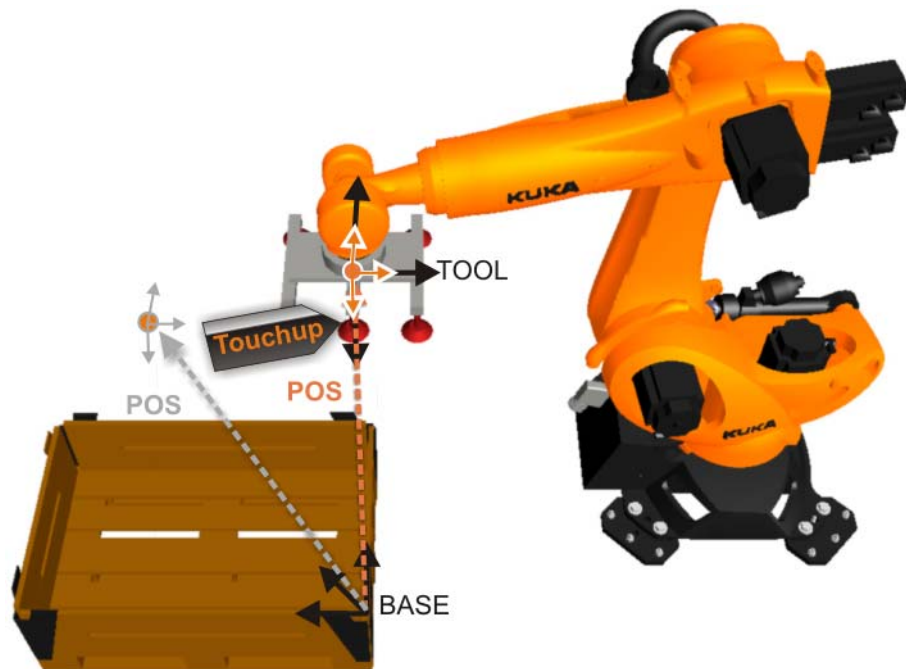


图 6-23: 用“Touchup”更改机器人位置

更改帧数据

- 更改帧数据（例如工具、基坐标）时，会导致位置发生位移（例如：“矢量位移”）
- 机器人位置会发生变化！
旧的点坐标依然会被保存并有效。发生变化的仅是参照系（例如基坐标）
- 可能会出现超出工作区的情况！因此不能到达某些机器人位置。
- 如果机器人位置保持不变，但帧参数改变，则必须在更改参数（例如基坐标）后在所要的位置上用“Touchup”更新坐标！

警告 此外，用户对话框会发出警告：“注意：更改以点为参照的帧参数时会有碰撞危险！”

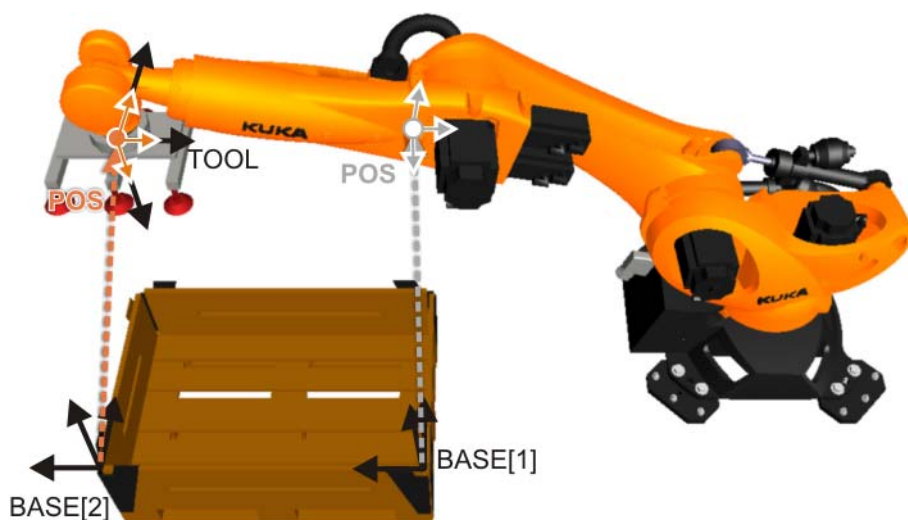


图 6-24: 更改帧数据 (以基坐标为例)

更改运动数据

- 更改速度或者加速度时会改变移动属性。这可能会影响加工工艺，特别是使用轨迹应用程序时：
 - 胶条厚度。
 - 焊缝质量。

更改运动方式

- 更改运动方式时总是会导致更改轨迹规划！这在不利情况下可能会导致发生碰撞，因为轨迹可能会发生意外变化。

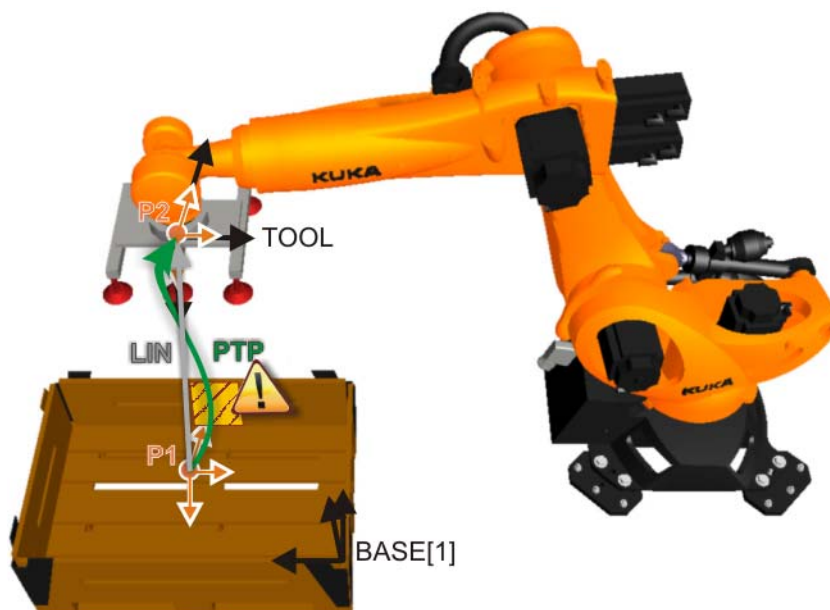


图 6-25: 更改运动方式


关于更改运动指令的安全提示

警告 每次更改完运动指令后都必须在低速（运行方式 T1）下测试机器人程序。立即以高速启动机器人程序可能会导致机器人系统和整套设备损坏，因为可能会出现不可预料运动。如果有人位于危险区域，则可能会造成重伤。

更改运动参数 - 帧

1. 将光标放在须改变的指令行里。
2. 点击**更改**。指令相关的联机表格自动打开。
3. 打开选项窗口“帧”。

4. 设置新工具坐标系或者基坐标系或者外部 TCP
5. 用 **OK** 确认用户对话框“注意！改变以点为基准的帧参数时会有碰撞危险！”
6. 如要**保留当前的机器人位置**及更改的工具坐标系和 / 或基坐标系设置，则必须按下 **Touch Up** 键，以便重新计算和保存当前位置。
7. 用软键**指令 OK** 存储变更。

 **警告** 如果帧参数发生变化，则必须重新测试程序是否会发生碰撞。

更改位置


更改机器人位置的操作步骤：

1. 设置运行方式 T1，将光标放在要改变的指令行里。
2. 将机器人移到所要的位置。
3. 点击**更改**。指令相关的联机表格自动打开。
4. 对于 PTP 和 LIN 运动：
 - 按下 **Touchup**（修整），以便确认 TCP 的当前位置为新的目标点。
 对于 CIRC 运动：
 - 按 **Touchup HP**（修整辅助点），以便确认 TCP 的当前位置为新的辅助点。
 - 或者按 **Touchup ZP**（修整目标点），以便确认 TCP 的当前位置为新的目标点。
5. 点击**是**确认安全询问。
6. 用**指令 OK** 存储变更。

更改运动参数

这种方法可用于以下更改：

- 运动方式
 - 速度
 - 加速度
 - 轨迹逼近
 - 轨迹逼近距离
1. 将光标放在须改变的指令行里。
 2. 点击**更改**。指令相关的联机表格自动打开。
 3. 更改参数。
 4. 用软键**指令 OK** 存储变更。

 **警告** 更改运动参数后必须重新检查程序是否不会引发碰撞并且过程可靠。

6.6 练习：沿轨迹运行和轨迹逼近

练习目的

成功完成此练习后，您可执行下列操作：

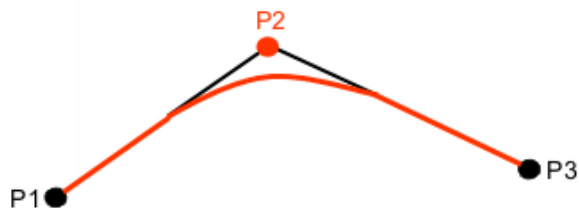
- 为含有运动类型 PTP、LIN 和 CIRC 的运动编制简单的程序
- 为含有精确停止点和轨迹逼近的运动编程
- 导航器中的程序应用（拷贝、复制、改名、删除）

前提条件

为成功完成此练习，必须满足以下前提条件：

- 具备为含有运动类型 PTP、LIN 和 CIRC 的运动编程的基础知识

- 具备有关运动轨迹逼近方面的理论知识



- 具备有关起始位置（原点）的理论知识

练习内容 — A 部分 请完成以下任务：建立构件轮廓 1 的程序

1. 以 **Bauteilkontur1**（构件轮廓 1）为名创建一个新程序
2. 在工作台上用蓝色基坐标、以尖触头 1 作为工具示教构件轮廓 1
 - 工作台上的移动速度为 0.3 m/s
 - 注意：工具的纵轴应始终垂直于轮廓（姿态引导）
3. 在运行方式 T1、T2 和 自动运行模式下测试您的程序。此时必须注意遵守培训指导中学到的安全规定，

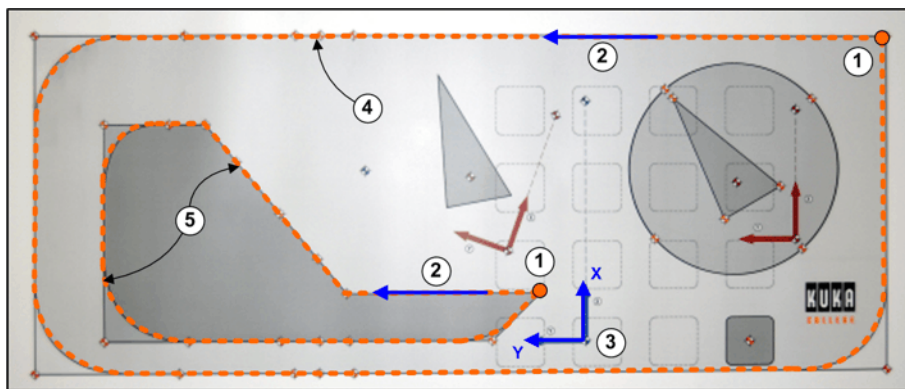


图 6-26: 沿轨迹运行和轨迹逼近：构件轮廓 1 和 2

- | | |
|----------|----------|
| 1 起点 | 2 运动方向 |
| 3 参照基坐标 | 4 构件轮廓 1 |
| 5 构件轮廓 2 | |

练习内容 — B 部分 请完成以下任务：复制程序并轨迹逼近

1. 以 **Bauteil1_CONT**（构件 1_CONT）为名创建构件轮廓 1 程序的复制版
2. 在新程序的运动指令中加入适当的轨迹逼近指令，使机器人走完整个轮廓。
3. 应以不同的轨迹逼近参数滑过轮廓上的角。
4. 在运行方式 T1、T2 和 自动运行模式下测试您的程序。此时务必遵守讲过的安全规程。

练习内容 — 附加作业 请完成以下任务：建立构件轮廓 2 的程序

1. 以 **Bauteilkontur2**（构件轮廓 2）为名创建第二个程序 请使用同一个基坐标和工具。
 - 工作台上的移动速度为 0.3 m/s
 - 注意：工具的纵轴应始终垂直于轮廓（姿态引导）
2. 在运行方式 T1、T2 和 自动运行模式下测试您的程序。此时必须注意遵守培训指导中学到的安全规定，
3. 以 **Bauteil2_CONT**（构件 2_CONT）为名创建构件轮廓 2 程序的复制版

- 4. 在新程序的运动指令中加入适当的轨迹逼近指令，使机器人走完整个轮廓。
- 5. 在运行方式 T1、T2 和 自动运行模式下测试您的程序。此时务必遵守讲过的安全规程。

与练习配套的问题

1. LIN 和 CIRC 运动有哪些特点？

.....
.....
.....

2. 在 PTP、LIN 和 CIRC 运动中移动速度以何种形式给出？该速度以什么为基准？

.....
.....
.....

3. 在 PTP、LIN 和 CIRC 运动中轨迹逼近距离是以何种形式给出的？

.....
.....

4. CONT 指令重新编程后必须注意什么？

.....
.....

5. 更改了起始位置后必须注意什么？

.....
.....

6. 修正或更改了编程的点后必须注意什么？

.....
.....

6.7 具有外部 TCP 的运动编程

用外部 TCP 进行运动编程

用固定工具进行运动编程时，运动过程与标准运动相比会产生以下区别：

- 联机表格中的标识：在选项窗口 **Frames** 中，**外部 TCP** 项的值必须为“TRUE”。



图 6-27: 选项窗口“帧”：外部 TCP

- **运动速度**以外部 TCP 为基准。
- 沿轨迹的**姿态**于是同样以外部 TCP 为基准。
- 不但要指定合适的基坐标系（固定工具 / 外部 TCP），而且也要指定合适的工具坐标系（运动的工件）。

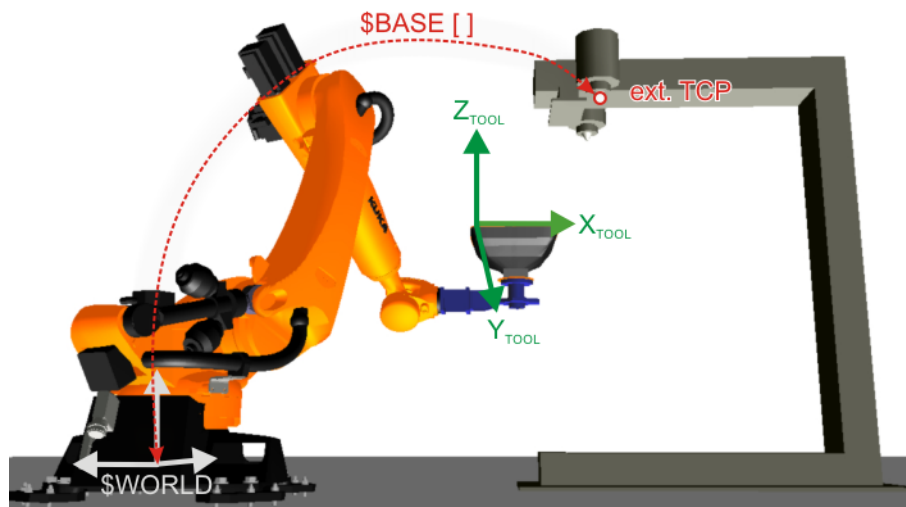


图 6-28: 固定工具时的坐标系

6.8 练习：以外部 TCP（工具中心点）进行运动编程

练习目的

成功完成此练习后，您可执行下列操作：

- 用一个由机器人移动的部件参照一个固定工具进行运动编程

前提条件

为成功完成此练习，必须满足以下前提条件：

- 具备有关运动编程时激活外部工具的知识。

练习内容

请完成以下任务：为涂胶粘剂的轮廓编程

1. 用手将标牌张紧到抓爪上
2. 在程序名 **Schild_kleben** 下示教标牌上给定的轮廓。
 - 请为此使用您已测定的外部工具喷嘴和工件标牌。

- 注意：固定工具的纵轴应始终垂直于粘贴轮廓。
 - 标牌上的移动速度为 0.2 m/s。
3. 按规定测试您的程序
 4. 将您的程序存档

与练习配套的问题

1. 由您编程的粘贴速度以什么为基准？

.....
.....

2. 您如何在您的程序中激活外部工具？

.....
.....

7 利用机器人程序中的逻辑功能

7.1 逻辑编程入门

在逻辑编程时使用
输入端和输出端

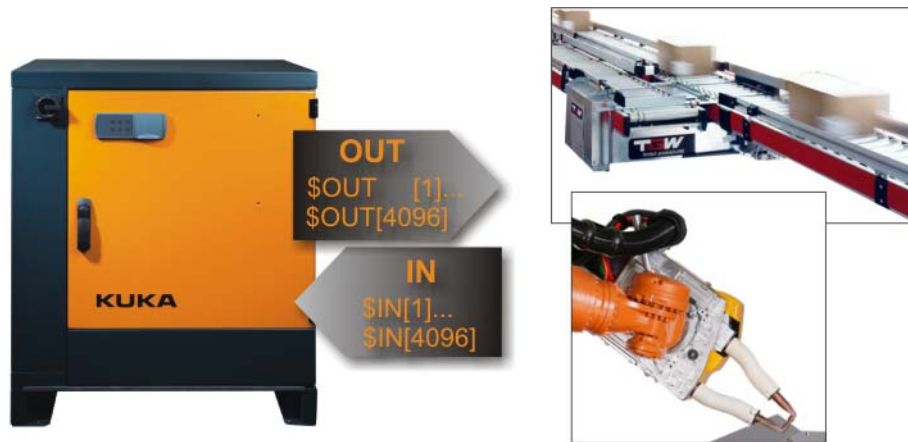


图 7-1: 数字输入 / 输出端

为了实现与机器人控制系统的外围设备进行通讯，可以使用数字式和模拟式输入端和输出端。

概念解释

概念	解释	示例
通讯	通过接口交换信号	询问状态（抓爪打开 / 闭合）
外围设备	“周围设备”	工具（例如：抓爪、焊钳等）、传感器、材料输送系统等等。
数字式	数字技术：离散的数值和时间信号	传感器信号：工件存在：值 1 (TRUE/ 真)，工件不存在：值 0 (FALSE/ 假)
模拟式	模拟一个物理量	温度测量
输入端	通过现场总线接口到达控制器的信号	传感器信号：抓爪已打开 / 抓爪已闭合
输出端	通过现场总线接口从控制系统发送至外围设备的信号	用于闭合抓爪的阀门切换指令。

对 KUKA 机器人编程时，使用的是表示逻辑指令的输入端和输出端信号。

- **OUT** | 在程序中的某个位置上关闭输出端
- **WAIT FOR** | 与信号有关的等待功能：控制系统在此等待信号：
 - 输入端 **IN**
 - 输出端 **OUT**
 - 定时信号 **TIMER**
 - 控制系统内部的存储地址（标记 / 1 比特内存）**FLAG** 或者 **CYCFLAG**（如果循环式地连续分析）
- **WAIT** | 与时间相关的等待功能：控制器根据输入的时间在程序中的该位置上等待。

7.2 等待功能的编程

计算机预进

计算机预进时预先读入（操作人员不可见）运动语句，以便控制系统能够在有轨迹逼近指令时进行轨迹设计。但处理的不仅仅是预进运动数据，而且还有数学的和控制外围设备的指令。

```

Editor
1 DEF Depal_Box1( )
2
3 INI
4 PTP HOME Vel= 100 % DEFAULT
5 PTP P1 Vel=100 % PDAT1 Tool[5]:GRP1 Base[10]:STAT1
6 PTP P2 Vel=100 % PDAT2 Tool[5]:GRP1 Base[10]:STAT1 ①
7 LIN P3 Vel=1 m/s CPDAT1 Tool[5]:GRP1 Base[10]:STAT1
8 OUT 26'' State=TRUE ②
9 LIN P4 Vel=1 m/s CPDAT2 Tool[5]:GRP1 Base[10]:STAT1
10 PTP P5 Vel=100 % PDAT3 Tool[5]:GRP1 Base[10]:STAT1 ③
11 PTP HOME Vel=100 % PDAT4
12
13 END

```

图 7-2: 计算机预进

- 1 主运行指针（灰色语句条）
- 2 触发预进停止的指令语句
- 3 可能的预进指针位置（不可见）

某些指令将触发一个预进停止。其中包括影响外围设备的指令，如 OUT 指令（抓爪关闭，焊钳打开）。如果预进指针暂停，则不能进行轨迹逼近。

等待功能

运动程序中的等待功能可以很简单地通过联机表格进行编程。在这种情况下，等待功能被区分为与时间有关的等待功能和与信号有关的等待功能。

用 WAIT 可以使机器人的运动按编程设定的时间暂停。WAIT 总是触发一次预进停止。

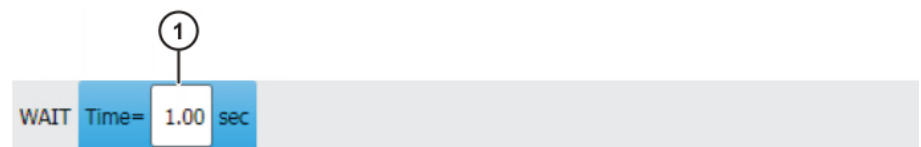


图 7-3: WAIT 的联机表格

序号	说明
1	等待时间 ■ ≥ 0 s

程序举例：

```

PTP P1 Vel=100% PDAT1
PTP P2 Vel=100% PDAT2
WAIT Time=2 sec
PTP P3 Vel=100% PDAT3

```

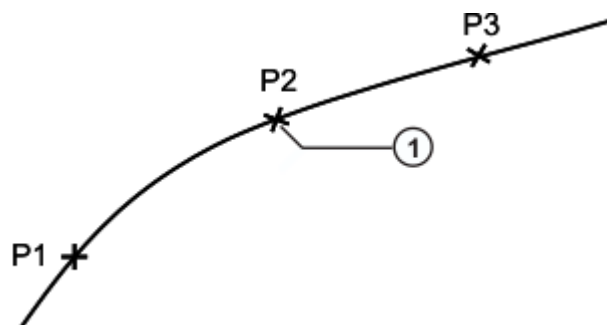


图 7-4: 逻辑运动示例

序号	备注
1	在点 P2 上中断运动 2 秒钟。

WAIT FOR 设定一个与信号有关的等待功能。

需要时可将多个信号（最多 12 个）按逻辑连接。如果添加了一个逻辑连接，则联机表格中会出现用于附加信号和其它逻辑连接的栏。

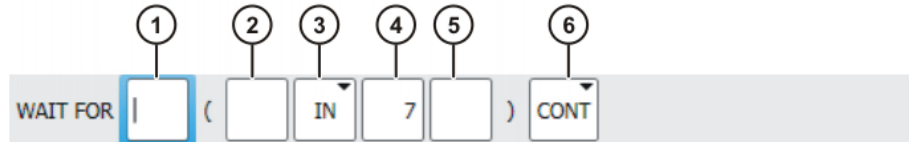


图 7-5: WAIT FOR 的联机表格

序号	说明
1	添加外部连接。运算符位于加括号的表达式之间。 <ul style="list-style-type: none"> ■ AND ■ OR ■ EXOR 添加 NOT。 <ul style="list-style-type: none"> ■ NOT ■ [空白] 用相应的按键添加所需的运算符。
2	添加内部连接。运算符位于一个加括号的表达式内。 <ul style="list-style-type: none"> ■ AND ■ OR ■ EXOR 添加 NOT。 <ul style="list-style-type: none"> ■ NOT ■ [空白] 用相应的按键添加所需的运算符。
3	等待的信号 <ul style="list-style-type: none"> ■ IN ■ OUT ■ CYCFLAG ■ TIMER ■ FLAG
4	信号的编号 <ul style="list-style-type: none"> ■ 1 ... 4096
5	如果信号已有名称则会显示出来。 仅限于专家用户组使用： 通过点击长文本可输入名称。名称可以自由选择。
6	<ul style="list-style-type: none"> ■ CONT：在预进过程中加工 ■ [空白]：带预进停止的加工



小心

在使用条目 CONT 时必须注意：该信号是在预进中被查询的！预进时间过后不能识别信号更改！

逻辑连接

在应用与信号相关的等待功能时也会用到逻辑连接。用逻辑连接可将对不同信号或状态的查询组合起来：例如可定义相关性，或排除特定的状态。

一个具有逻辑运算符的函数始终以一个真值为结果，即最后始终给出“真”（值 1）或“假”（值 0）。

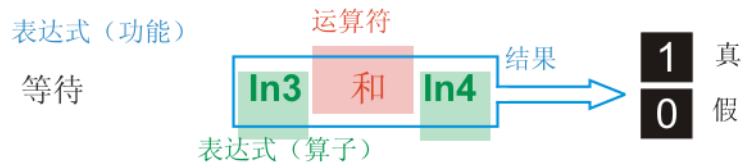


图 7-6: 逻辑连接的范例和原理

逻辑连接的**运算符**为：

- NOT | 该运算符用于否定，即使值逆反（由“真”变为“假”）。
- AND | 当连接的两个表达式为真时，该表达式的结果为真。
- OR | 当连接的两个表达式中至少一个为真时，该表达式的结果为真。
- EXOR | 当由该运算符连接的命题有不同的真值时，该表达式的结果为真。

有预进和没有预进的加工 (CONT)

与信号有关的等待功能在有预进或者没有预进的加工下都可以进行编程设定。**没有预进**表示，在任何情况下都会将运动停在某点，并在该处检测信号：(1) (>>> 图 7-7)。即该点不能轨迹逼近。

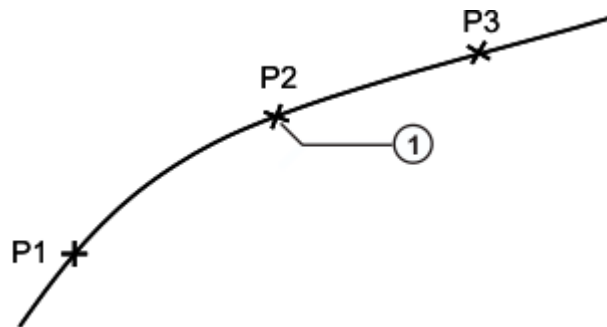


图 7-7: 逻辑运动示例

```
PTP P1 Vel=100% PDAT1
PTP P2 CONT Vel=100% PDAT2
WAIT FOR IN 10 'door_signal'
PTP P3 Vel=100% PDAT3
```

有预进编程设定的与信号有关的等待功能允许在指令行前创建的点进行轨迹逼近。但预进指针的当前位置却不唯一（标准值：三个运动语句），因此无法明确确定信号检测的准确时间 (1) (>>> 图 7-8)。除此之外，信号检测后也不能识别信号更改！

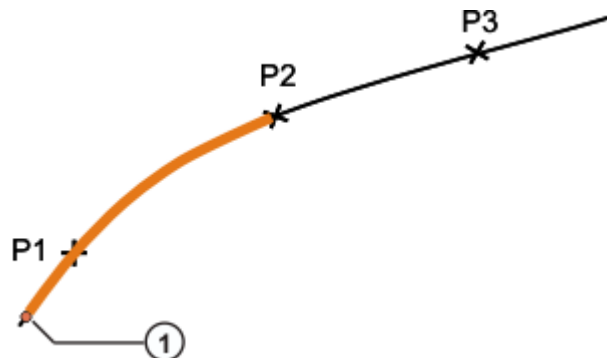


图 7-8: 带预进的逻辑运动示例


```

PTP P1 Vel=100% PDAT1
PTP P2 CONT Vel=100% PDAT2
WAIT FOR IN 10 'door_signal' CONT
PTP P3 Vel=100% PDAT3

```

操作步骤

1. 将光标放到其后应插入逻辑指令的一行上。
2. 选择菜单序列指令 > 逻辑 > WAIT FOR 或 WAIT。
3. 在联机表格中设置参数。
4. 用指令 OK 保存指令。

7.3 简单切换功能的编程

简便的切换功能

通过切换功能可将数字信号传送给外围设备。为此要使用先前相应分配给接口的输出端编号。

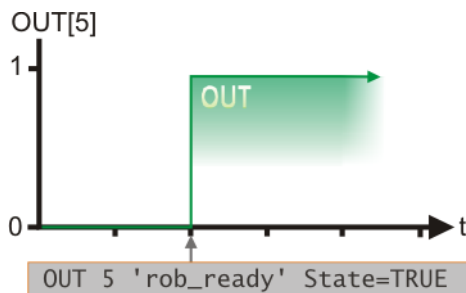


图 7-9: 静态切换

信号设为静态，即它一直存在，直至赋予输出端另一个值。切换功能在程序中通过联机表格实现：

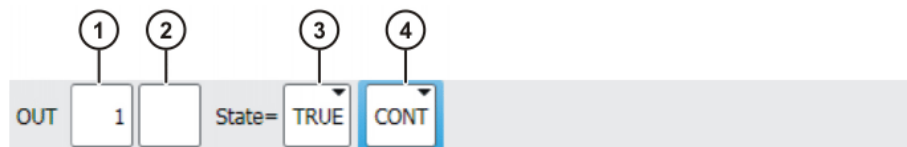


图 7-10: 联机表格 OUT

序号	说明
1	输出端编号 <ul style="list-style-type: none"> 1 ... 4096
2	如果输出端已有名称则会显示出来。 仅限于专家用户组使用： 通过点击长文本可输入名称。名称可以自由选择。
3	输出端接通的状态 <ul style="list-style-type: none"> 正确 错误
4	<ul style="list-style-type: none"> CONT：在预进中进行的编辑 [空白]：含预进停止的处理



小心 在使用条目 CONT 时必须注意：该信号是在预进中设置的！

脉冲切换功能

与简单的切换功能一样，在此输出端的数值也变化。然而，在脉冲时，于定义的时间过去之后，信号又重新取消。

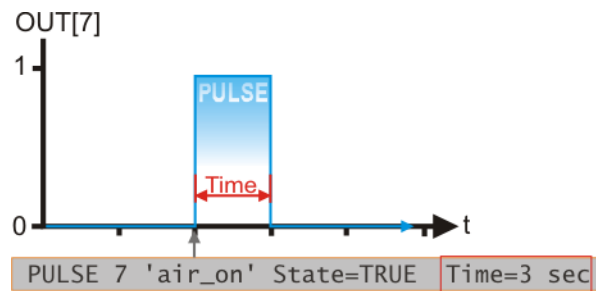


图 7-11: 脉冲电平

编程同样使用联机表格，在该联机表格中给脉冲设置了一定的时间长度。

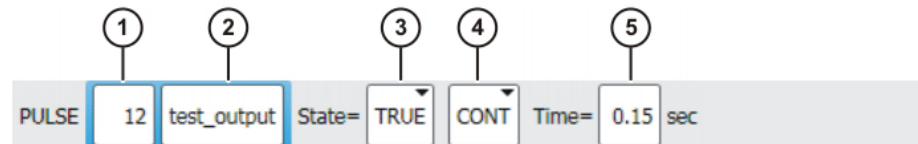


图 7-12: PULSE 的联机表格

序号	说明
1	输出端编号 <ul style="list-style-type: none"> 1 ... 4096
2	如果输出端已有名称则会显示出来。 仅限于专家用户组使用： 通过点击长文本可输入名称。名称可以自由选择。
3	输出端接通的状态 <ul style="list-style-type: none"> TRUE: “高”电平 FALSE: “低”电平
4	<ul style="list-style-type: none"> CONT: 在预进过程中加工 [空白]: 带预进停止的加工
5	脉冲长度 <ul style="list-style-type: none"> 0.10 ... 3.00 s

在切换功能时
CONT 的影响

如果在 OUT 联机表格中去掉条目 CONT，则在切换过程时必须执行**预进停止**，并接着在切换指令前于点上进行精确暂停。给输出端赋值后继续该运动。

```

LIN P1 Vel=0.2 m/s CPDAT1
LIN P2 CONT Vel=0.2 m/s CPDAT2
LIN P3 CONT Vel=0.2 m/s CPDAT3
OUT 5 'rob_ready' State=TRUE
LIN P4 Vel=0.2 m/s CPDAT4
  
```

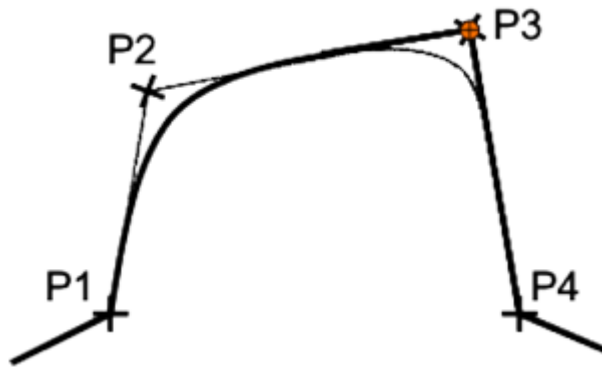


图 7-13: 含切换和预进停止的运动举例

插入条目 CONT 的作用是，预进指针不被暂停（不触发预进停止）。因此，在切换指令前运动可以轨迹逼近。在**预进**时发出信号。

```

LIN P1 Vel=0.2 m/s CPDAT1
LIN P2 CONT Vel=0.2 m/s CPDAT2
LIN P3 CONT Vel=0.2 m/s CPDAT3
OUT 5 'rob_ready' State=TRUE CONT
LIN P4 Vel=0.2 m/s CPDAT4

```

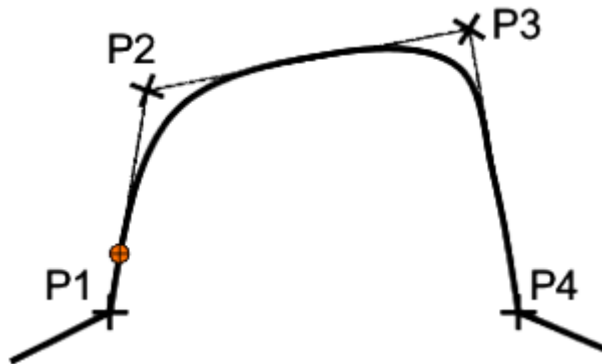


图 7-14: 含切换和预进的运动示例

⚠ 小心 预进指针的标准值占三行。但预进是会变化的，即必须考虑到，切换时间点不是保持不变的！

操作步骤

1. 将光标放到其后应插入逻辑指令的一行中。
2. 选择菜单序列**指令 > 逻辑 > OUT > OUT 或 PULSE**。
3. 在联机表格中设置参数。
4. 用**指令 OK** 存储指令。

7.4 轨迹切换功能编程

概述

轨迹切换功能可以用来在轨迹的目标点上设置起点，而无需中断机器人运动。其中，切换可分为“静态”(SNY OUT) 和“动态”(SYN Pulse) 两种。SYN OUT 5 切换的信号与 SYN PULSE 5 切换的信号相同。只有切换的方式会发生变化。

选项 Start/End (起始 / 终止)

可以运动语句的起始点或目标点为基准触发切换动作。切换动作的时间可推移。参照动作语句可以是 LIN、CIRC 或 PTP 运动。

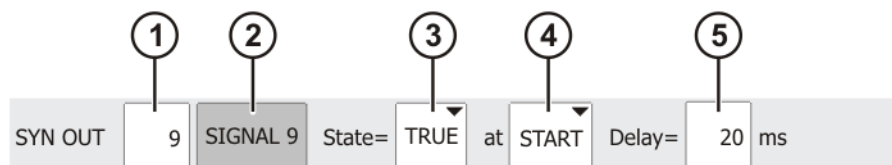


图 7-15: 联机表格 SYN OUT , 选项 START

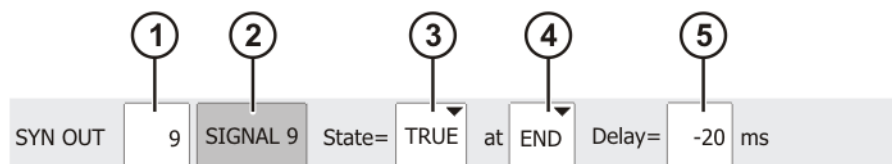


图 7-16: 联机表格 SYN OUT , 选项 END

序号	说明	数值范围
1	输出端编号	1 ... 4096
2	如果输出端已有名称则会显示出来。 仅限于专家用户组使用：通过点击软键 长文本 可输入名称。	可自由选择
3	输出端接通的状态	TRUE , FALSE
4	切换位置点 <ul style="list-style-type: none"> ■ START (起始)：以动作语句的起始点为基准切换。 ■ END (终止)：以动作语句的目标点为基准切换。 	START , END 选项 PATH :
5	切换动作的时间推移 提示：此时间数值为绝对值。视机器人的速度，切换点的位置将随之变化。	-1000 ... +1000 ms

选项 PATH

用选项 PATH 可相对于运动语句的目标点触发切换动作。切换动作的位置和 / 或时间均可推移。动作语句可以是 LIN 或 CIRC 运动。但不能是 PTP 运动。

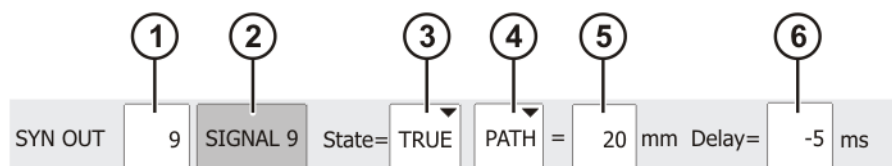


图 7-17: 联机表格 SYN OUT , 选项 PATH

序号	说明	数值范围
1	输出端编号	1 ... 4096
2	如果输出端已有名称则会显示出来。 仅限于专家用户组使用：通过点击软键 长文本 可输入名称。	可自由选择
3	输出端接通的状态	TRUE , FALSE
4	切换位置点 <ul style="list-style-type: none"> ■ PATH：以动作语句的目标点为基准切换。 	START , END 选项 PATH :

序号	说明	数值范围
5	切换动作的方位推移 提示： 方位数据以动作语句的目标点为基准。因此，机器人速度改变时切换点的位置不变。	-1000 ... +1000 ms
6	切换动作的时间推移 提示： 时间推移以方位推移为基准。	

切换选项 Start/End
(起始/终止)的
作用

程序举例 1: 选项 Start (起始)

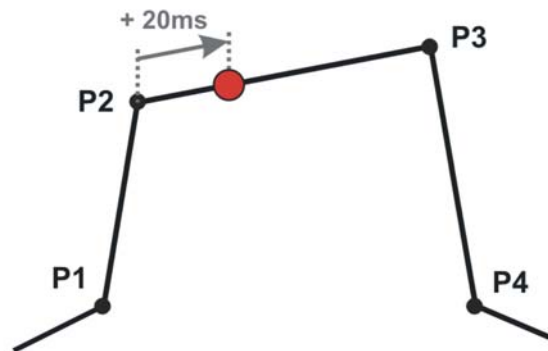


图 7-18: SYN OUT Start (起始) 带正延迟

```

LIN P1 VEL=0.3m/s CPDAT1
LIN P2 VEL=0.3m/s CPDAT2
;Schaltfunktion bezogen auf P2
SYN OUT 8 'SIGNAL 8' State= TRUE at Start Delay=20ms
LIN P3 VEL=0.3m/s CPDAT3
LIN P4 VEL=0.3m/s CPDAT4

```

程序举例 2: 选项 Start (起始) 带 CONT 和正延迟

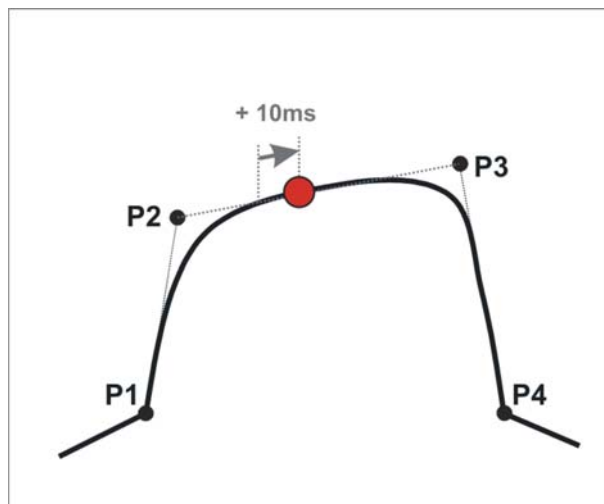


图 7-19: SYN OUT Start (起始) 带 CONT 和正延迟

```

LIN P1 VEL=0.3m/s CPDAT1
LIN P2 CONT VEL=0.3m/s CPDAT2
;Schaltfunktion bezogen auf P2
SYN OUT 8 'SIGNAL 8' State= TRUE at Start Delay=10ms
LIN P3 CONT VEL=0.3m/s CPDAT3
LIN P4 VEL=0.3m/s CPDAT4

```

程序举例 3: 选项 End (终止) 带负延迟

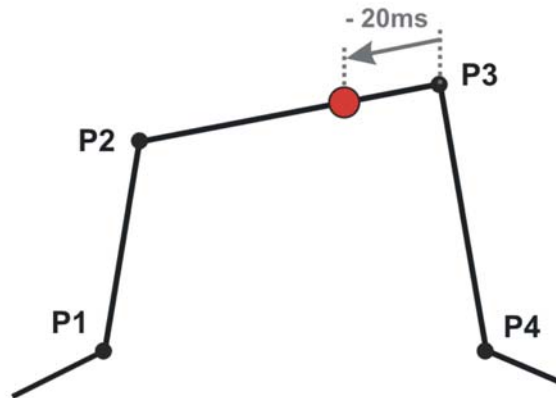


图 7-20: 带负延迟的 SYN OUT END (终止)

```

LIN P1 VEL=0.3m/s CPDAT1
LIN P2 VEL=0.3m/s CPDAT2
;Schaltfunktion bezogen auf P3
SYN OUT 9 'SIGNAL 9' Status= TRUE at End Delay=-20ms
LIN P3 VEL=0.3m/s CPDAT3
LIN P4 VEL=0.3m/s CPDAT4

```

程序举例 4: 选项 End (终止) 带 CONT 和负延迟

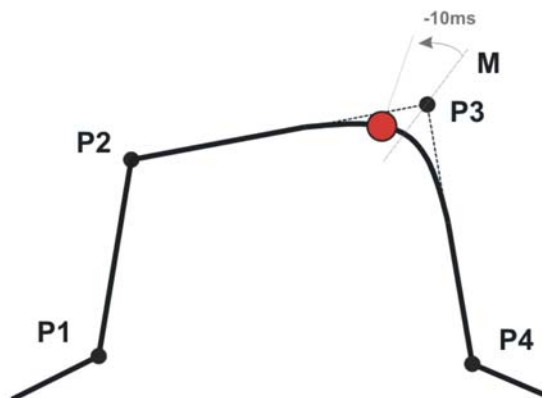


图 7-21: 带选项 End (终止) 和负延迟的 SYN OUT

```

LIN P1 VEL=0.3m/s CPDAT1
LIN P2 VEL=0.3m/s CPDAT2
;Schaltfunktion bezogen auf P3
SYN OUT 9 'SIGNAL 9' Status= TRUE at End Delay=-10ms
LIN P3 VEL=0.3m/s CPDAT3
LIN P4 VEL=0.3m/s CPDAT4

```

程序举例 5: 选项 End (终止) 带 CONT 和正延迟

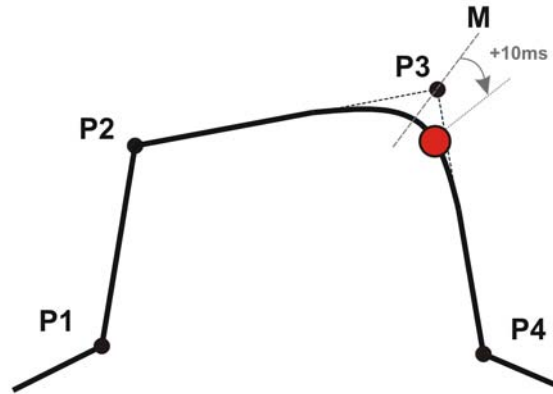


图 7-22: 带选项 End (终止) 和正延迟的的 SYN OUT

```

LIN P1 VEL=0.3m/s CPDAT1
LIN P2 VEL=0.3m/s CPDAT2
;Schaltfunktion bezogen auf P3
SYN OUT 9 'SIGNAL 9' Status= TRUE at End Delay=10ms
LIN P3 VEL=0.3m/s CPDAT3
LIN P4 VEL=0.3m/s CPDAT4

```

无 Cont 的切换极限

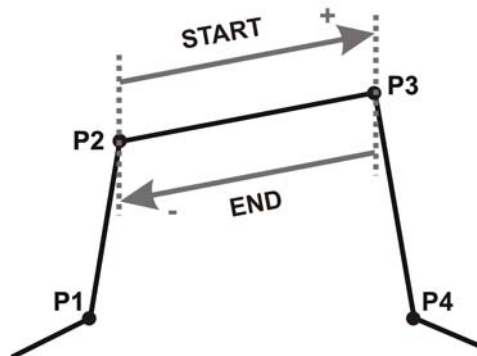


图 7-23: 无 CONT 的选项 Start/End (起始 / 终止) 切换极限

带 Cont 的切换极限 :

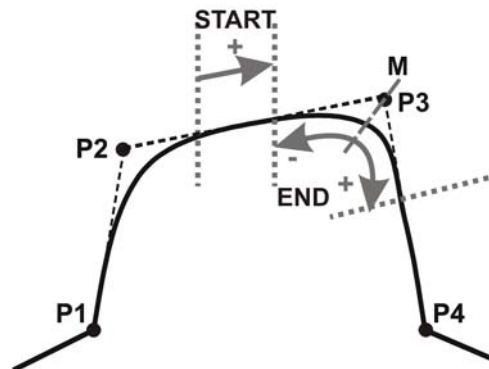


图 7-24: 带 CONT 的选项 Start/End (起始 / 终止) 切换极限

切换选项“路径”
的作用

程序举例 :

铣刀必须切换到轨迹上。在 P3 后 20 mm 处应流畅地开始部件加工操作。为了使铣刀在 (Path=20) P3 后 20mm 处达到最高转速，必须提前 5 ms (Delay=-5ms) 将其接通。

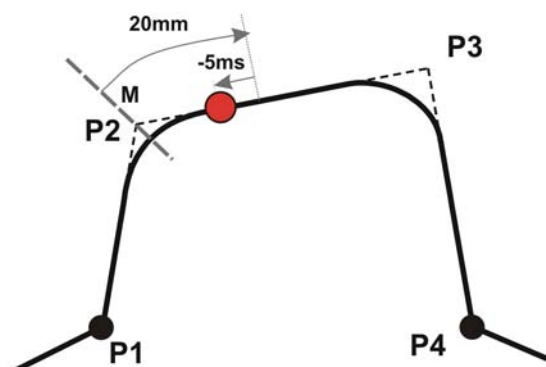


图 7-25

```

LIN P1 VEL=0.3m/s CPDAT1
;Schaltfunktion bezogen auf P2
SYN OUT 9 'SIGNAL 9' Status= True Path=20 Delay=-5ms
LIN P2 CONT VEL=0.3m/s CPDAT2
LIN P3 CONT VEL=0.3m/s CPDAT3
LIN P4 VEL=0.3m/s CPDAT4

```

切换极限

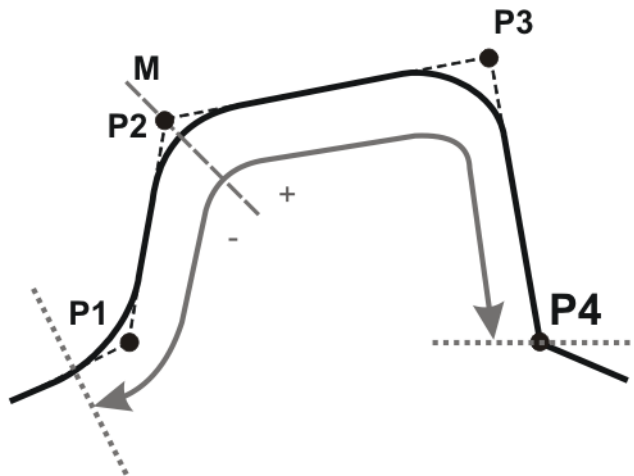


图 7-26

操作步骤

1. 将光标放到其后应插入逻辑指令的一行上。
2. 选择菜单序列 **指令 > 逻辑 > OUT > SYN OUT** 或 **SYN PULSE**。
3. 在联机表格中设置参数。
4. 用 **指令 OK** 保存指令。

7.5 练习：逻辑指令和切换功能

练习目的

成功完成此练习后，您可执行下列操作：

- 简单逻辑指令的编程
- 执行简单的切换功能

- 执行以轨迹为参照的切换功能
- 相关信号的等待功能的编程

前提条件

为成功完成此练习，必须满足以下前提条件：

- 具有有关简单逻辑指令编程的理论知识
 - 具有有关简单切换功能的知识
 - 具有有关简单脉冲功能的知识
 - 具有有关以轨迹为参照的切换功能的知识
 - 具有有关以轨迹为参照的脉冲功能的知识
 - 具有有关等待功能的知识

练习内容

请完成以下任务：给部件轮廓 1 涂敷粘结剂的逻辑编程

1. 从程序 Bauteil1_CONT 中创建一个名为**粘胶轮廓**的副本
2. 请为程序扩展以下逻辑功能：
 - 在离开 HOME 位置前，应从 PLC 中发出开通信号（输入端 11）。
 - 在粘胶喷嘴到达部件前 0.5 秒时，必须激活粘胶喷嘴（输出端 13）
 - 从平面过渡到部件拱面上时，应接通信号指示灯，该指示灯在从拱面过渡到平面上时重新熄灭（输出端 12）。
 - 离开部件前 0.75 秒时，必须重新关闭粘胶喷嘴（输出端 13）。
 - 在部件加工末端前 50 mm 时，PLC 应收到完成信息。发给 PLC 的该信号（输出端 11）应停留 2 秒钟。
3. 按规定测试您的程序

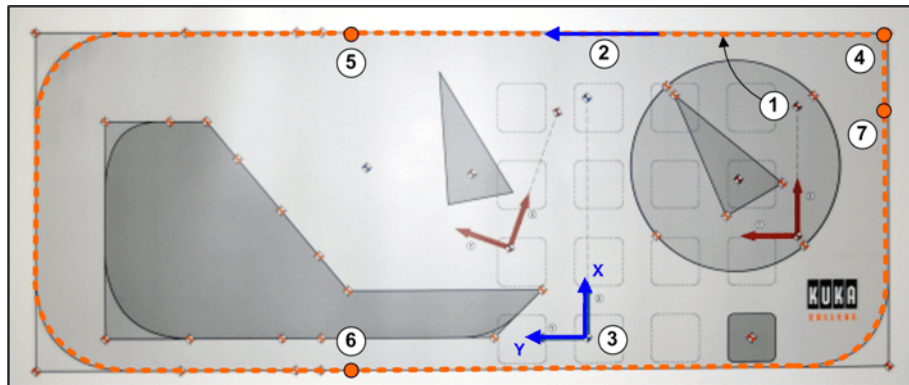


图 7-27: 输入 / 输出端：胶料涂敷

- | | |
|-------------|-------------|
| 1 构件轮廓 1 | 2 运动方向 |
| 3 参照基坐标 | 4 部件开始和结束点 |
| 5 平面 - 拱面过渡 | 6 拱面 - 平面过渡 |
| 7 部件末端前的点 | |

与练习配套的问题

1. OUT 和 OUT CONT 指令之间有何区别？必须注意些什么？

.....

.....

2. 如何区分 PULSE 和 OUT 指令？

.....

.....

3. 何时使用 SYN OUT 指令？

.....

.....

4. 在进行运动编程时，`SYN OUT` 路径指令的使用有无限制？

.....

.....

5. 同时使用 `WAIT FOR` 指令和 `CONT` 指令时会有哪些危险？

.....

.....

8 变量的使用

8.1 显示和更改变量值

变量概览

变量是运算过程中将出现的计算值的通配符。变量由存储位置、类型、名称和内容标示。

Variable	
存储位置	全局 / 局部
型号	整数/小数、真/假、字符
名称	名称
数值	内容/数值

图 8-1: 变量的特征

一个变量的存储位置对其有效性至关重要。一个全局变量建立在系统文件中，适用于所有程序。一个局部变量建立于应用程序中，因此仅适用于正在运行的程序（也只有这时可读）。

所用变量举例

变量	存储位置	类型	名称	值
当前工具	全局 KUKA 系统变量	整数	\$ACT_TOOL	5
当前基坐标	全局 KUKA 系统变量	整数	\$ACT_BASE	12
件数计数器	局部 应用程序	整数	zaehler	3
轴 2 软件限位开关的负角度值	全局 machine.dat	小数	\$SoftN_End[2]	-104.5
故障状态	全局 例如储存在 config.dat 中	真 / 假值	stoerung	true

显示项中变量的可用性和有效性

变量的存储位置对变量的显示方式非常重要：

- **全局** | 如果变量为全局变量，则随时都可以显示。在这种情况下，变量必须保存在系统文件（例如 config.dat、machine.dat）或者在局部数据列表中作为全局变量。
- **局部** | 局部变量可以分为程序文件 (.src) 中的局部变量或者局部数据列表 (*.dat) 中的局部变量。如果变量是在 .src 文件中定义的，则该变量仅在程序运行时存在。我们将此称为“运行时间变量”。如果变量是在 .dat 文件中被定义为局部变量，并且仅在相关程序文件中已知，则其值在关闭程序后保持不变。

显示并更改一个变量的值

1. 在主菜单中选择 **显示 > 变量 > 单项**。
显示单项变量窗口即打开。
2. 在**名称**栏输入变量名称。
3. 如果选择了一个程序，则在**模块**栏中将自动填写该程序。
如果要显示一个其他程序中的变量，则如下输入程序：
/R1/ 程序名称
在 /R1/ 和程序名称之间不要输入文件夹。对程序名称不要输入文件后缀。
4. 按回车键。
在**当前值**栏中将显示该变量的当前数值。如果无任何显示，则说明还未给该变量赋值。
5. 在**新值**栏中输入所需数值。
6. 按回车键。

在**当前值**栏中将显示此新值。

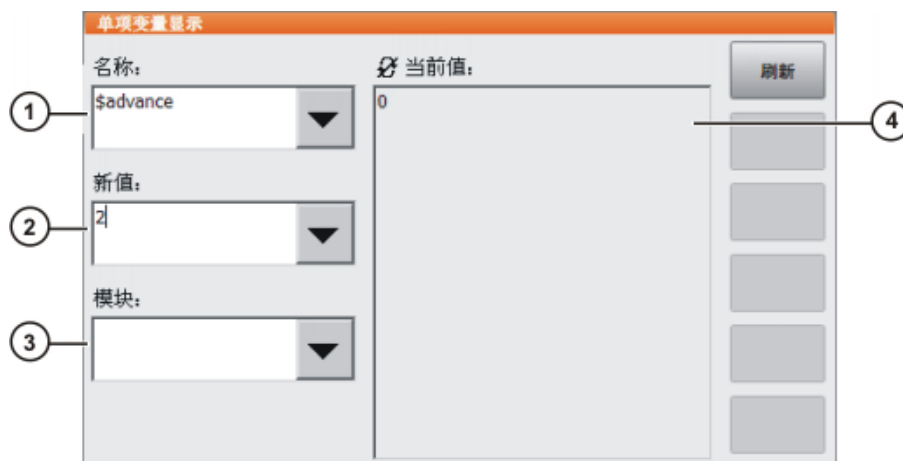





图 8-2: 单项变量显示窗口

序号	说明
1	应进行更改的变量的名称
2	应赋予该变量的新值
3	在其中查找该变量的程序 对于系统变量来说， 模块 栏并不重要。
4	此栏有两种状态： <ul style="list-style-type: none"> ■ : 显示的数值不自动更新。 ■ : 显示的数值将自动更新。 在各状态间进行切换： <ul style="list-style-type: none"> ■ 按下更新。

8.2 询问机器人状态

系统内部的值

有关机器人状态的众多信息可通过查询系统内部值获得。可随时查询这些值。

 系统内部值称为“系统变量”
变量是一个预留的存储位置。该存储位置（也称为“值的通配符”）
在计算机的存储器中始终有一个固定的名称和位址。

主要有下列系统变量用于机器人状态的查询：

- 定时器 (Timer)
- 标记 (旗标 Flag)
- 计数器
- 输入和输出信号 (IN/OUT) 也作为系统变量管理。

系统变量	查询用途举例
\$TIMER[1..64]	检查机器人的等待时间（构件的冷却、由于过程的持续时间而引起的等待时间等等）。
\$FLAG[1..1024] \$CYCFLAG[1..256]	对可设置在程序某一位置的标记也可在程序以外查询（全局）。 周期性旗标另外还被持续分析。
I[1..20]	对加工步骤进行计数的计数器。

系统变量	查询用途举例
\$IN[1..4096]	检查一个抓爪是否打开或闭合。（抓爪的传感器通过一个输入信号报告状态）
\$OUT[1..4096]	检查一条抓爪指令。（一个输出信号将指令传给抓爪的执行元件）

系统变量的特征

KUKA 系统变量总是以“\$”字符开头。可随时查询系统变量，因为它们始终有效。全局数据表被用作存储位置。

显示系统信息

显示旗标、计数器和定时器的操作步骤：

- 在主菜单中选择**显示 > 变量**。

这时便有不同的系统变量可供选择：

- 循环旗标
- 旗标
- 计数器
- 定时器

显示输入端和输出端的操作步骤：

- 在主菜单中选择**显示 > 输入 / 输出端 > 数字输入端或数字输出端**。

8.3 练习：显示系统变量

练习目的

成功完成此练习后，您可执行下列操作：

- 打开变量显示
- 显示系统变量

前提条件

为成功完成此练习，必须满足以下前提条件：

- 具备有关系统变量显示的理论知识
- 具备有关系统变量的理论知识

练习内容



请完成以下任务：

1. 打开变量显示
2. 显示当前起始位置（变量名称：XHOME）
3. 显示当前机器人位置（变量名称：\$pos_act）
4. 确定您机器人轴 1-3 的软件限位开关的位置（变量名称：\$softn_end[Achse] und \$softp_end[Achse]）
5. 测定预进指针的值（变量名称：\$advance）

9 使用工艺程序包

9.1 使用 KUKA.GripperTech 操作抓爪

工艺程序包 KUKA.GripperTech 是可以在一种可后续加载的工艺程序包。它简化了抓爪的使用，例如：
h 操作抓爪时必须使用以下工艺键：

工艺键	说明
	选择抓爪。 抓爪编号即被显示。 <ul style="list-style-type: none"> 按下上键，数字变大。 按下下键，数字变小。
	切换抓爪状态（例如：打开或者闭合） 当前状态不显示。可能出现的状态取决于配置的抓爪类型。对于焊钳：可能出现的状态取决于手动焊钳控制系统的配置。

操作抓爪的操作步骤

注意

必须首先激活工艺键，才能用工艺键来操作抓爪！
在主菜单中选择 **配置 > 状态键 > GripperTech**。



警告

警告！

使用抓爪系统时有挤伤和割伤的危险。抓爪操作人员必须保证无身体部分可被抓爪挤伤。

1. 通过工艺键选择抓爪。



2. 激活运行方式 T 1 或 T 2
3. 操作确认按键
4. 通过工艺键控制抓爪。



9.2 用 KUKA.GripperTech 对抓爪编程

抓爪指令编程 用工艺程序包 KUKA.GripperTech 可通过预制的联机表格直接在所选的程序中对抓爪指令进行编程。有两条指令可供选用：

- **SET Gripper** | 用于在程序中打开 / 关闭抓爪的指令
- **CHECK Gripper** | 用于检查抓爪位置的指令

抓爪编程功能

运动中的抓爪指令

- 原则上可为抓爪指令相应编程，使指令相对起始点或目标点得到执行。
- 在联机表格中只要点击记录项 **CONT** 并给出以 ms 为单位的延迟时间 (Delay)。
-

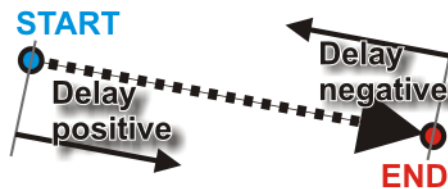


图 9-1: 延迟示意图

警告 选择包括运动中编辑在内的抓爪指令时一定要格外小心，因为考虑不周时飞出的构件或发生碰撞会造成人员或财产的损伤！

精确暂停时抓爪的设置

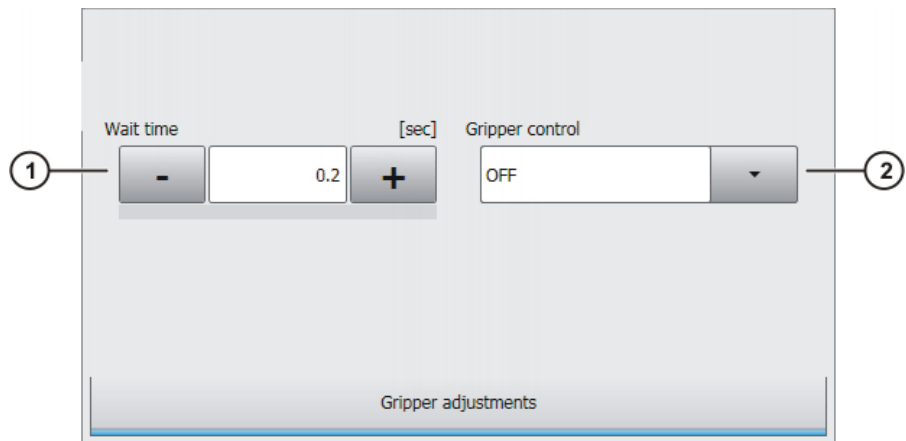


图 9-2: 抓爪设置

应用抓爪监控：

- 如果用 **ON** 激活了抓爪监控，则将对已完成参数设定的传感器进行查询
- 如果缺少传感器的回应，则将出现超时错误，可在测试运行中模拟传感器
- 如果抓爪监控功能在 **OFF** 下未激活，则将等待设定的等待时间

抓爪编程的操作步骤

操作步骤

1. 选择菜单序列指令 > GripperTech > Gripper。
2. 在联机表格中设置参数。
3. 用指令 **OK** 存储。

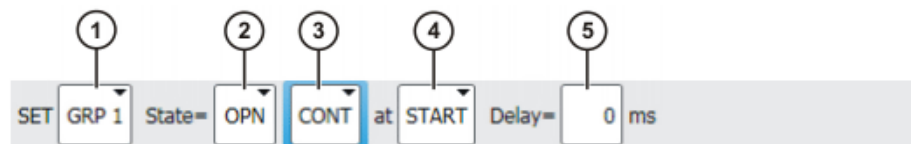


图 9-3: 带轨迹逼近的抓爪联机表格

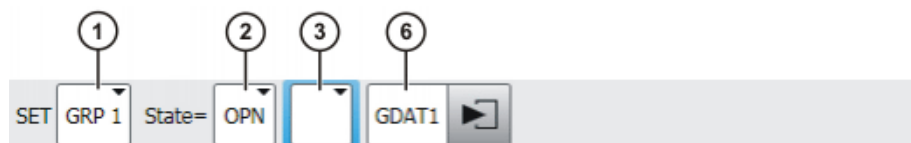


图 9-4: 没有轨迹逼近的抓爪联机表格

序号	说明
1	选择抓爪。 <ul style="list-style-type: none"> 所有可配置的抓爪均可供选择。
2	选择抓爪的开关状态。 <ul style="list-style-type: none"> 数量取决于抓爪的类型。 名称取决于配置。
3	在预进过程中加工。 <ul style="list-style-type: none"> CONT：在预进过程中加工。 [空白]：带预进停止的加工。
4	仅在选择 CONT 时，此栏才供使用。 <ul style="list-style-type: none"> START（起始）：抓爪的动作在运动起始点上执行。 END（终止）：抓爪的动作在运动目标点上执行。
5	仅在选择 CONT 时，此栏才供使用。 确定抓爪的动作执行时相对于运动起始点或目标点的等待时间。 <ul style="list-style-type: none"> -200 ... 200 ms
6	具有抓爪参数的数据组

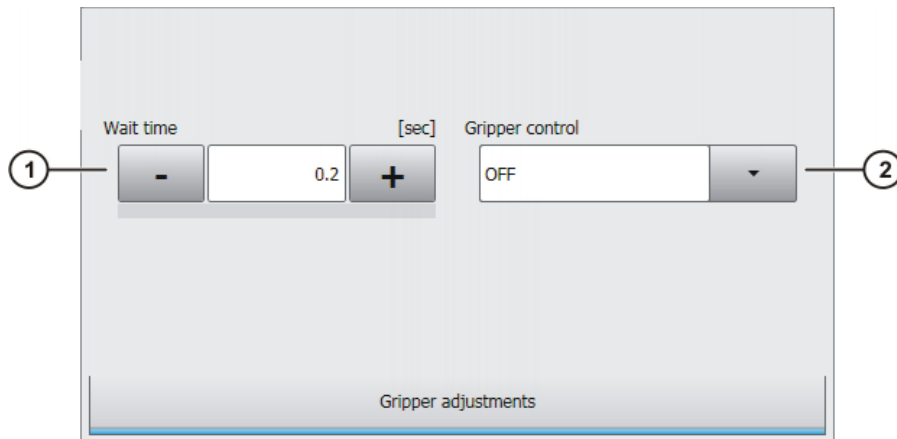


图 9-5: 抓爪设置

序号	说明
1	在该等待时间后继续编程设定的运动 <ul style="list-style-type: none"> 0... 10 s
2	抓爪监控 <ul style="list-style-type: none"> OFF（默认），ON

抓爪监控编程的操作步骤

操作步骤

1. 选择菜单序列指令 > **GripperTech** > **Check Gripper**。
2. 在联机表格中设置参数。
3. 用指令 **OK** 存储。

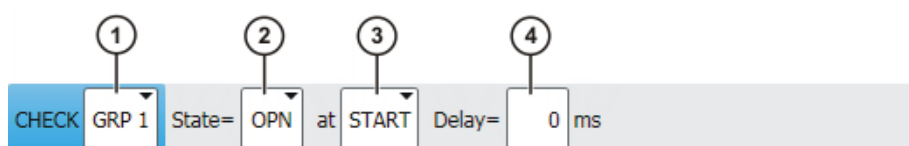


图 9-6: 联机表格 Check Gripper（抓爪检查）

序号	说明
1	选择抓爪。 <ul style="list-style-type: none"> 所有可配置的抓爪均可供选择。
2	选择抓爪的开关状态。 <ul style="list-style-type: none"> 数量取决于抓爪的类型。 名称取决于配置。
3	选择传感器的感测时刻。 <ul style="list-style-type: none"> START (起始) : 传感器在动作的起始点进行感测。 END (终止) : 传感器在动作的目标点进行感测。
4	确定传感器感测时相对于运动起始点或目标点的等待时间。

9.3 KUKA.GripperTech 配置

配置可能性和抓爪类型

KUKA.GripperTech 可以让客户对抓爪进行自行配置。其中包括五种预设定的抓爪类型可供选择。除此之外，还可以对用户自定义的抓爪进行配置。

注意

在控制系统中可对多达 16 种不同的抓爪进行配置。

抓爪类型

类型	OUT	IN	状态	示例
类型 1	2	4	2	具有“打开”和“闭合”功能的简单抓爪
类型 2	2	2	3	滑座在中位
类型 3	2	2	3	具有“吸气”、“吹气”、“关闭”功能的真空气爪
类型 4	3	2	3	类似于类型 3，但有三个控制输出端
类型 5	2	4	2	类似于类型 1，但用脉冲替代持续信号
空置	可任意配置			

The screenshot shows a configuration window for a gripper. At the top, there are two numbered callouts (1 and 2) pointing to the 'Gripper 1: GRP 1' field and the 'Type: 3' field. Below this is a 'Not defined' section. The 'Outputs' section has four numbered callouts (3, 4, 5, 6) pointing to the input fields: 1: 14, 2: 15, 3: 0, and 4: 0. The 'States' section has a callout (7) pointing to the 'State VACUUM' field. At the bottom, there are buttons for '前一个', '继续', and '更改'.

图 9-7: 示例：预设定的抓爪

序号	说明
1	抓爪编号 <ul style="list-style-type: none"> ■ 1 ... 16
2	抓爪名称 名称显示在联机表格中。默认名称可以更改。 <ul style="list-style-type: none"> ■ 1 ... 24 个字符
3	类型 <ul style="list-style-type: none"> ■ 对于预设定的抓爪：1 ... 5
4	抓爪名称（在保存后才刷新） 该名称不可以更改。
5	指定输出端编号 不需要的输出端可以指定为“0”。以这种方式可以立即判断出该那些输入端未被使用。如果仍要给这类输入端指定一个编号，则该编号无效。

序号	说明
6	指定输入端编号 不需要的输入端可以指定为“0”。以这种方式可以立即判断出该那些输入端未被使用。如果仍要给这类输入端指定一个编号，则该编号无效。
7	开关状态 默认名称可以更改。如果要在联机表格中选择相应的抓爪，则该名称显示在联机表格中。

可自由编程的抓爪类型

为了满足所有的用户需求，系统内置了可自由编程的抓爪类型。通过 \$CONFIG.DAT, USERGRP.DAT 和 USER_GRP.SRC 文件中的条目可以对任意数量和完全可以自由设定的抓爪进行配置。

注意

关于抓爪配置的详细信息见库卡系统技术操作指南 KUKA.Gripper&SpotTech 3.0。

抓爪配置的操作步骤

对预设定的抓爪类型进行配置

1. 在主菜单中选择 **配置 > 输入 / 输出端 > 抓爪**。一个窗口自动打开。
2. 通过 **继续** 或者上一个选择所要的抓爪编号。
3. 需要时可更改抓爪的默认名称。
4. 在 1 至 5 之间给抓爪分配一个类型。
5. 分配输入端和输出端。
6. 需要时可更改状态的默认名称。
7. 用 **更改** 保存配置。

9.4 练习：抓爪编程“标牌”

练习目的

成功完成此练习后，您可执行下列操作：

- 对用于控制和检查抓爪与夹钳的指令进行编程 (KUKA.Gripper & SpotTech)
- 激活专门用于特定工艺的状态键并用其操作

前提条件

为成功完成此练习，必须满足以下前提条件：

- 具备有关 KUKA.Gripper & SpotTech 工艺程序包的理论知识

练习内容

请完成以下任务：抓取和放下标牌

1. 创建名为 **Schild_holen** 的新程序，创建时应使用抓爪工具和蓝色基坐标。
2. 示教标牌抓取过程，使放下和抓取位置如图 (>>> 图 9-8) 所示。
3. 在运行方式 T1、T2 和自动运行模式下测试您的程序。此时务必遵守讲过的安全规程。
4. 创建名为 **Schild_ablegen** 的第二个程序，创建时应使用必要的基坐标和相应的工具。
5. 示教标牌放下过程。
6. 在运行方式 T1、T2 和自动运行模式下测试您的程序。此时务必遵守讲过的安全规程。
7. 将您的程序存档。



图 9-8: 具有置放位置的标牌

1 标牌

2 放下位置

与练习配套的问题

1. 什么时候必须进行安全教育？（至少 4 个答案）

.....
.....
.....
.....

2. KUKA 机器人上的自由旋转装置有什么作用？

.....
.....
.....

9.5 练习：以尖触头为对象对抓爪进行编程

练习目的

成功完成此练习后，您可执行下列操作：

- 对用于控制和检查抓爪与夹钳的指令进行编程 (KUKA.Gripper & SpotTech)
- 激活专门用于特定工艺的状态键并用其操作

前提条件

为成功完成此练习，必须满足以下前提条件：

- 具备有关 KUKA.Gripper & SpotTech 工艺程序包的理论知识

练习内容

请完成以下任务：取出、放回尖触头 1

1. 以 **Stift1_holen** (取出尖触头 1) 和 **Stift1_ablegen** (放回尖触头 1) 为名创建两个新程序 (复制)。
2. 编程时，请充分利用工具作业方向在手动移动中的优点。
3. 请注意：在取出和放回尖触头库时移动速度不得高于 0.3 m/s。
4. 在取尖触头之前，请先就抓爪位置发出安全询问。

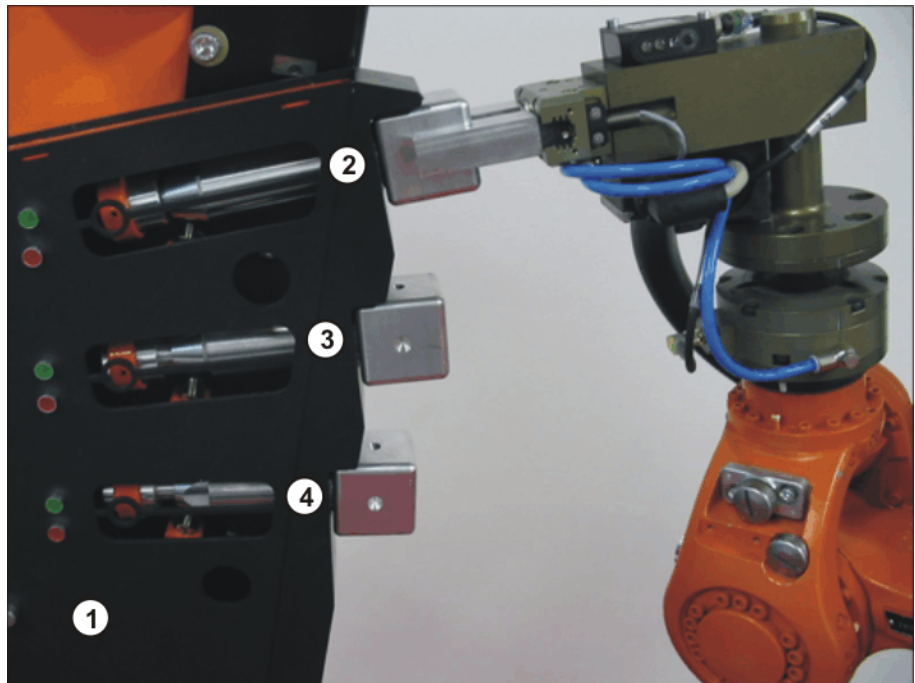


图 9-9: 尖触头库

- | | |
|---------|---------|
| 1 尖触头库 | 2 尖触头 1 |
| 3 尖触头 2 | 4 尖触头 3 |

与练习配套的问题

1. 等待时间与抓爪监控 ON/OFF 之间的区别是什么？

.....

.....

3. 您收到了 *无法轨迹逼近* 提示文字的提示信息。请说出可能的原因。

.....

.....

.....

4. 有多少库卡标准抓爪类型？

.....

.....

10 在 KRL 中成功编程

10.1 机器人程序的结构和组成

程序流程控制 除了纯运动指令和通讯指令（切换和等待功能）之外，在机器人程序中还有大量用于控制程序流程的程序。其中包括：

- **循环** | 循环是控制结构。它不断重复执行指令块指令，直至出现终止条件。
 - 无限循环
 - 计数循环
 - 当型和直到型循环
- **分支** | 使用分支后，便可以只在特定的条件下执行程序段。
 - 条件分支
 - 多分支结构

无限循环 在无限循环中无止境地重复指令段。然而，却可通过一个提前出现的中断（含 **EXIT** 功能）退出循环语句。

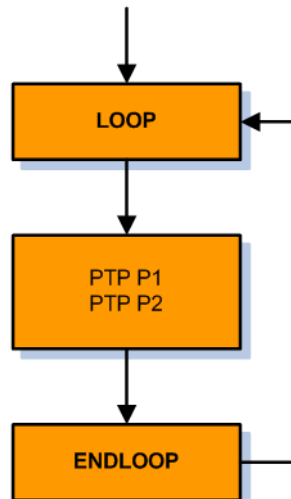


图 10-1: 程序流程图：无限循环

LOOP 指令举例：

- **无 EXIT**
永久执行对 P1 和 P2 的运动指令

```

LOOP
  PTP P1 Vel=100% PDAT1
  PTP P2 Vel=100% PDAT2
ENDLOOP
  
```

- **带 EXIT**
一直执行对 P1 和 P2 的运动指令，直到输入端 30 切换到 TRUE。

```

LOOP
  PTP P1 Vel=100% PDAT1
  PTP P2 Vel=100% PDAT2
  IF $IN[30]==TRUE THEN
    EXIT
  ENDIF
ENDLOOP
  
```

计数循环

用计数循环（FOR 循环语句）可使指令重复定义的次数。循环的次数借助于一个计数变量控制。

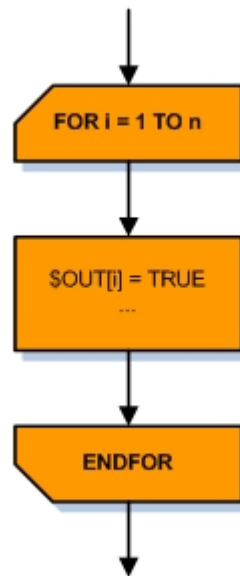


图 10-2: 程序流程图：FOR 循环语句

一个 FOR 循环语句举例：将输出端 1 至 5 依次切换到 TRUE。用整数 (Integer) 变量 “i” 来对一个循环语句内的循环进行计数。

```
INT i
...
FOR i=1 TO 5
    SOUT[i] = TRUE
ENDFOR
```

当型循环

WHILE 循环是一种当型或者先判断型循环，这种循环会在执行循环的指令部分前判断终止条件是否成立。

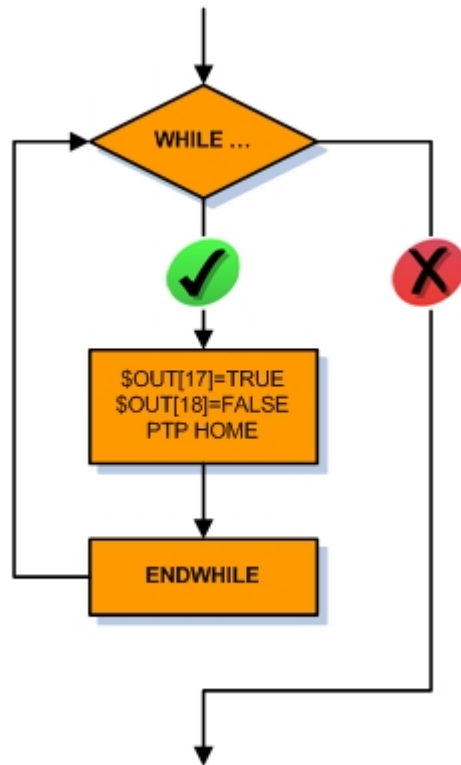


图 10-3: WHILE 程序流程图

WHILE 循环示例：输出端 17 被切换为 TRUE，而输出端 18 被切换为 FALSE，并且机器人移入 Home 位置，但仅当循环开始时就已满足条件（输入端 22 为 TRUE）时才成立。

```

WHILE $IN[22]==TRUE
  $OUT[17]=TRUE
  $OUT[18]=FALSE
  PTP HOME
ENDWHILE

```

直到型循环

REPEAT 循环是一种直到型或者检验循环，这种循环会在第一次执行完循环的指令部分后才会检测终止条件。

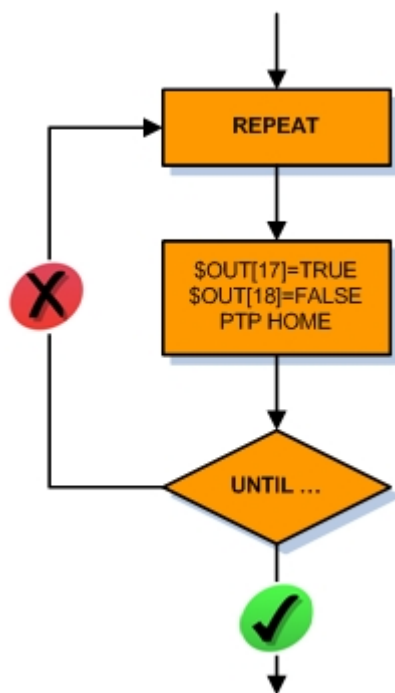


图 10-4: 程序流程图 REPEAT

REPEAT 循环示例：输出端 17 被切换为 TRUE，而输出端 18 被切换为 FALSE，并且机器人移入 Home 位置。这时才会检测条件。

```

REPEAT
  $OUT[17]=TRUE
  $OUT[18]=FALSE
  PTP HOME
UNTIL $IN[22]==TRUE

```

条件性分支

条件性分支（IF 语句）由一个条件和两个指令部分组成。如果满足条件，则可处理第一个指令。如果未满足条件，则执行第二个指令。

但是，对 IF 语句也有替代方案：

- 第二个指令部分可以省去：无 ELSE 的 IF 语句。由此，当不满足条件时紧跟在分支后便继续执行程序。
- 多个 IF 语句可相互嵌套（多重分支）：问询被依次处理，直到有一个条件得到满足。

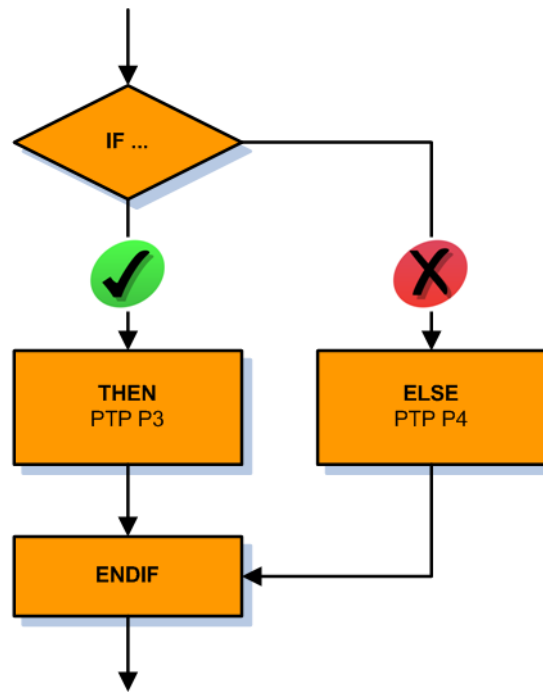


图 10-5: 程序流程图：IF 分支

IF 语句举例：如果满足条件（输入端 30 必须为 TRUE），则机器人运动到点 P3，否则到点 P4。

```

...
IF $IN[30]==TRUE THEN
  PTP P3
ELSE
  PTP P4
ENDIF
  
```

分配器

一个 SWITCH 分支语句是一个分配器或多路分支。此处首先分析一个表达式。然后，该表达式的值与一个案例段 (CASE) 的值进行比较。值一致时执行相应案例的指令。

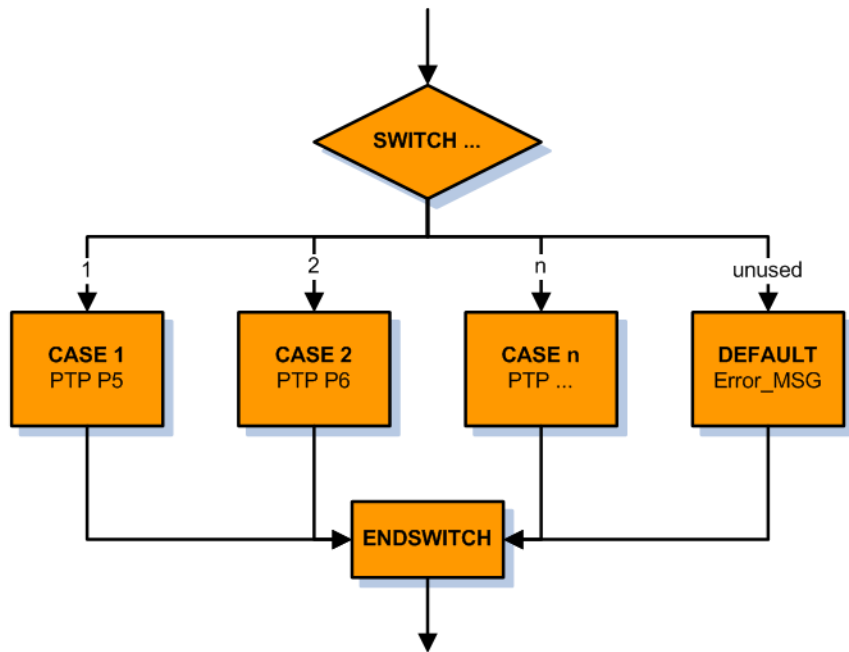


图 10-6: 程序流程图：分配模块 SWITCH - CASE

对带有名称“状态”的整数变量 (Integer)，首先要检查其值。如果变量的值为 1，则执行案例 1 (CASE 1)：机器人运动到点 P5。如果变量的值为 2，则执行案例 2 (CASE 2)：机器人运动到点 P6。如果变量的值未在任何案例中列出 (在该例中为 1 和 2 以外的值)，则将执行默认分支：故障信息。

```

INT status
...
SWITCH status
  CASE 1
    PTP P5
  CASE 2
    PTP P6
  ...
  DEFAULT
    ERROR_MSG
ENDSWITCH
  
```

10.2 结构化机器人程序

机器人程序结构化设计的方法

机器人程序的结构是体现其使用价值的一个十分重要的因数。程序结构化越规范，程序就越易于理解、执行效果越好、越便于读取、越经济。为了使程序得到结构化设计，可以使用以下技巧：

- **注释** | 注解和印章
- **缩进** | 空格
- **隐藏** | Folds (“折叠”)
- **模块化** | 子程序

注释和印章

添加注释为在机器人程序中存储仅针对程序读者的文本提供了可能性。即机器人解释器不读入该文本。该文本只是为了提高程序的可读性。

在机器人程序中可以在许多地方使用注释：

- **有关程序文本的信息** | 作者、版本、创建日期

```

Editor
1 DEF welding1( )
2 ; Programmed by JACK SPARROW
3 ; Version 1.5 (10/10/2010)
4 INI
5
6 PTP HOME Vel= 100 % DEFAULT
7

```

图 10-7: 注释示例：信息

- 程序文本的分段 | 主要使用画图符号（特殊符号 #、*、~ 等）

```

0
7 ;*****Initialisation*****
8 INIT
9 BASISTECH INI
10 CHECK HOME

```

图 10-8: 注释示例：分层目录

- 添加注释（专家层面）| 通过在程序行的起始位置添加分号来使该程序行“变成注释”，即该文本作为注释来识别，而不进入程序执行。

```

10 CHECK HOME
11 PTP HOME Vel= 100 % DEFAULT
12 ;$OUT[33]=TRUE
13 AUTOEXT INI|
14 LOOP

```

图 10-9: 注释示例：添加注释

- 对行的解释
以及对需执行的工作的说明 | 标识未完成的程序段

```

17
18 CASE 1
19 P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0) ; Reset
   ↳ Progr.No.-Request
20 Main_Prog ( ) ; Call User-Program
21

```

图 10-10: 注释示例：说明

注意

只有不断更新，注释才有意义。如果之后更改过指令，则也必须更新注释！

可以使用三种不同的注释方式：

- 添加分号（专家层面）| 通过插入分号（“;”）使一行中的后面部分变成注释。
- 插入联机表格“注释”

①

```

; vacuum off

```

图 10-11: 注释的联机表单

序号	说明
1	任意文本

- 插入联机表格“印章” | 在此还另外插入一个时间戳记。此外，还可以插入编辑者的姓名。

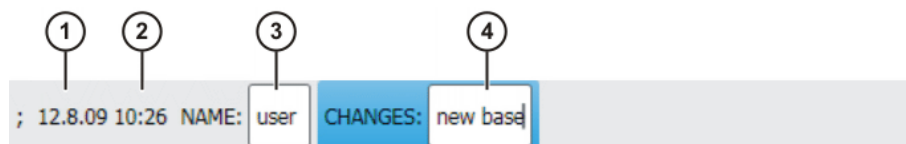


图 10-12: 印章的联机表格

序号	说明
1	系统日期 (不可编辑)
2	系统时间 (不可编辑)
3	用户的名称或代码
4	任意文本

插入注释和印章的操作步骤

1. 标定其后应插入注释或印章的那一行。
2. 选择菜单序列 **指令 > 注释 > 正常或印章**。
3. 输入所希望的数据。如果事先已经插入了注释或印章，则联机表格中还保留着相同数据。
 - 插入注释时，可用**新文本**来清空注释栏，以便输入新的文字。
 - 插入印章时，还可用**新时间**来更新系统时间，并用**新名称**清空名称栏。
4. 用**指令 OK** 存储。

程序行的缩进

提高机器人程序可读性的一个有效手段是缩进程序行。由此可提高程序单元之间关联的清晰度。

```

13 AUTOEXT INI
14 LOOP
15     P00 (#EXT_PGNO,#PGNO_GET,DMY[],0 )
16     SWITCH PGNO ; Select with Programnumber
17         CASE 1
18             P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 )
19             Main_Prog ( ) ; Call User-Program
20         CASE 2
21             P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 )
22             Sub_Prog1 ( ) ; Call User-Program
23         DEFAULT
24             P00 (#EXT_PGNO,#PGNO_FAULT,DMY[],0 )
25     ENDSWITCH
26 ENDLOOP

```

图 10-13: 程序行缩进

注意

缩进效应只是视觉上的。缩进的程序行在程序运行时与未缩进的程序行一样得到处理。

通过 Fold 隐藏程序行

KUKA 机器人编程语言可将程序行折叠和隐藏到 *Fold* 中。用户因此而看不到这些程序行。这使程序的阅读变得更加简洁方便。以后可在专家用户组中打开和编辑 Fold。

```

13
14
15 CHECK HOME
16

```

图 10-14: 关闭的 Fold


```

14
15 CHECK HOME
16   $H_POS=XHOME
17   IF CHECK_HOME==TRUE THEN
18     P00 (#CHK_HOME,#PGNO_GET,DMY[],0 ) ;Test HPos
19   ENDIF
20

```

图 10-15: 打开的 Fold

Fold 的颜色：

颜色	说明
深红	关上的 Fold
浅红	打开的 Fold
深蓝	关上的子 Fold
浅蓝	打开的子 Fold
绿色	Fold 内容

10.3 链接机器人程序

子程序技术

利用子程序技术可将机器人程序模块化，因而可以有效结构设计程序。目的是不将所有指令写入一个程序，而是将特定的流程、计算或过程转移到单独的程序中。

通过使用子程序产生了一系列的优点：

- 由于程序长度减短，主程序结构更清晰并更易读。
- 可独立开发子程序：编程花费可分摊，最小化错误源。
- 子程序可多次反复应用。

原则上可将子程序分为两种不同的类型

- 全局子程序

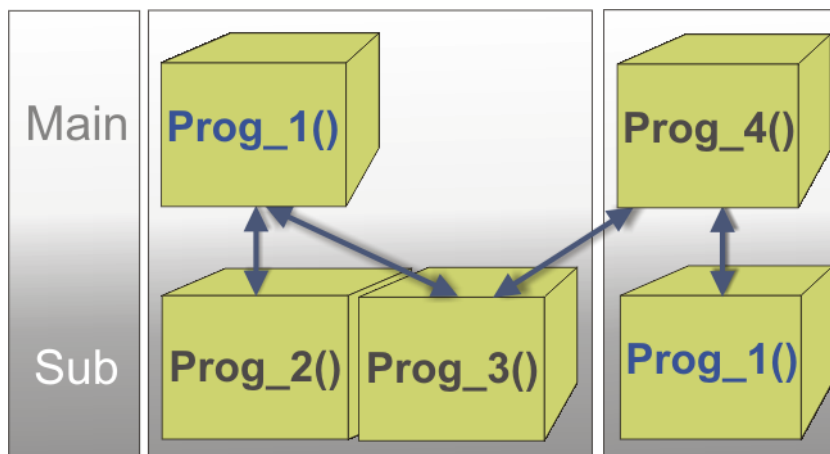


图 10-16: 全局子程序示意图

一个全局子程序是一个独立的机器人程序，可从另一个机器人程序调用。可根据具体要求对程序进行分支，即某一程序可在某次应用中用作主程序，而在另一次则用作子程序。

- 局部子程序

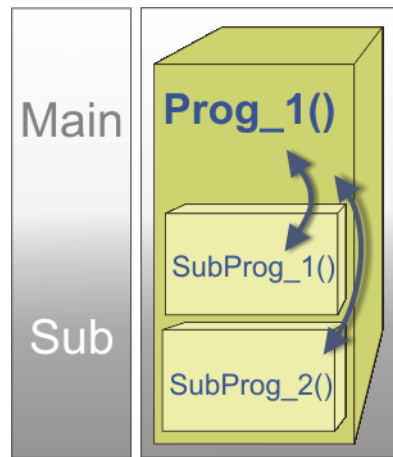


图 10-17: 示意图：局部子程序

局部子程序是集成在一个主程序中的程序，即指令包含在同一个 SRC 文件中。子程序的点坐标相应存放在同一个 DAT 文件中。

调用子程序的过程

每个程序都以 DEF 行开始并以 END 行结束。如果要在主程序中调用子程序，则正常情况下子程序会从 DEF 运行至 END。到达 END 行后，程序运行指针重新跳入发出调用指令的程序（主程序）。

```

1 DEF main( )
2 INI
3 PTP HOME Vel= 100 % DEFAULT
4 PTP P1 Vel=100 % PDAT1 Tool[2] Base[2]
5 PTP P2 Vel=100 % PDAT2 Tool[2] Base[2]
6 PTP P3 Vel=100 % PDAT3 Tool[2] Base[2]
7
8 sub_prog()
9
10 PTP P4 Vel=100 % PDAT4 Tool[2] Base[2]
11 PTP P5 Vel=100 % PDAT5 Tool[2] Base[2]
12 PTP P6 Vel=100 % PDAT6 Tool[2] Base[2]
13 PTP HOME Vel= 100 % DEFAULT
14 END
15
1 DEF Sub_Prog( )
2 INI
3 PTP P1 Vel=100 % PDAT1 Tool[2] Base[2]
4 OUT 25'' State=TRUE
5 PTP P4 Vel=100 % PDAT4 Tool[2] Base[2]
6 END

```

图 10-18: 调用子程序的过程

注意

为了能提早退出子程序（即在 END 行之前），可以在子程序中编入 RETURN 指令。程序读到该程序行时，即可提前终止子程序的运行。

调用子程序的操作步骤

为了对子程序调用进行编程，必须已选择用户组“专家”。子程序调用的句法为：

?? ()

1. 在主菜单中选择 **配置 > 用户组**。将显示出当前用户组。
2. 若欲切换至其它用户组，则：按下 **登录 ...**。选定用户组 **专家**。
3. 输入密码 **kuka**，然后用 **登录** 确认。
4. 用 **打开** 将所需的主程序载入编辑器

```

INI
PTP HOME Vel= 100% DEFAULT
PTP HOME Vel= 100% DEFAULT

```

5. 将光标定位在所需的行内
6. 输入子程序名称和括号，例如：**myprog()**

```
INI
PTP HOME Vel= 100% DEFAULT
myprog( )
PTP HOME Vel= 100% DEFAULT
```

7. 用关闭图标关闭编辑器并保存修改

10.4 练习：在 KRL 中编程

练习目的	成功完成此练习后，您可执行下列操作： <ul style="list-style-type: none">■ 无限循环编程■ 调用子程序编程■ 使 CELL.SRC 匹配外部自动运行
前提条件	为成功完成此练习，必须满足以下前提条件： <ul style="list-style-type: none">■ 具备在专家界面上使用导航器的知识■ 具备在专家界面上进行编程的基本知识 (KRL)■ 具备子程序和循环编程的知识■ 具备有关 CELL.SRC 结构的知识
练习内容	请完成以下任务： <ol style="list-style-type: none">1. 在专家界面上创建名为 Prozedur 的新模块。其它程序由该核心程序作为子程序调用。2. 准确的程序流程见程序流程图。(>>> 图 10-19)3. 在运行方式 T1、T2 和自动运行下测试新程序 Prozedur。此时务必遵守讲过的安全规程。4. 按下文所述补充 CELL.SRC：<ul style="list-style-type: none">■ 发送程序号 1，即从料库中取尖触头。■ 发送程序号 2，即沿工作台上的轮廓行进。■ 发送程序号 3，即将尖触头放回料库。5. 请测试您的 CELL.SRC

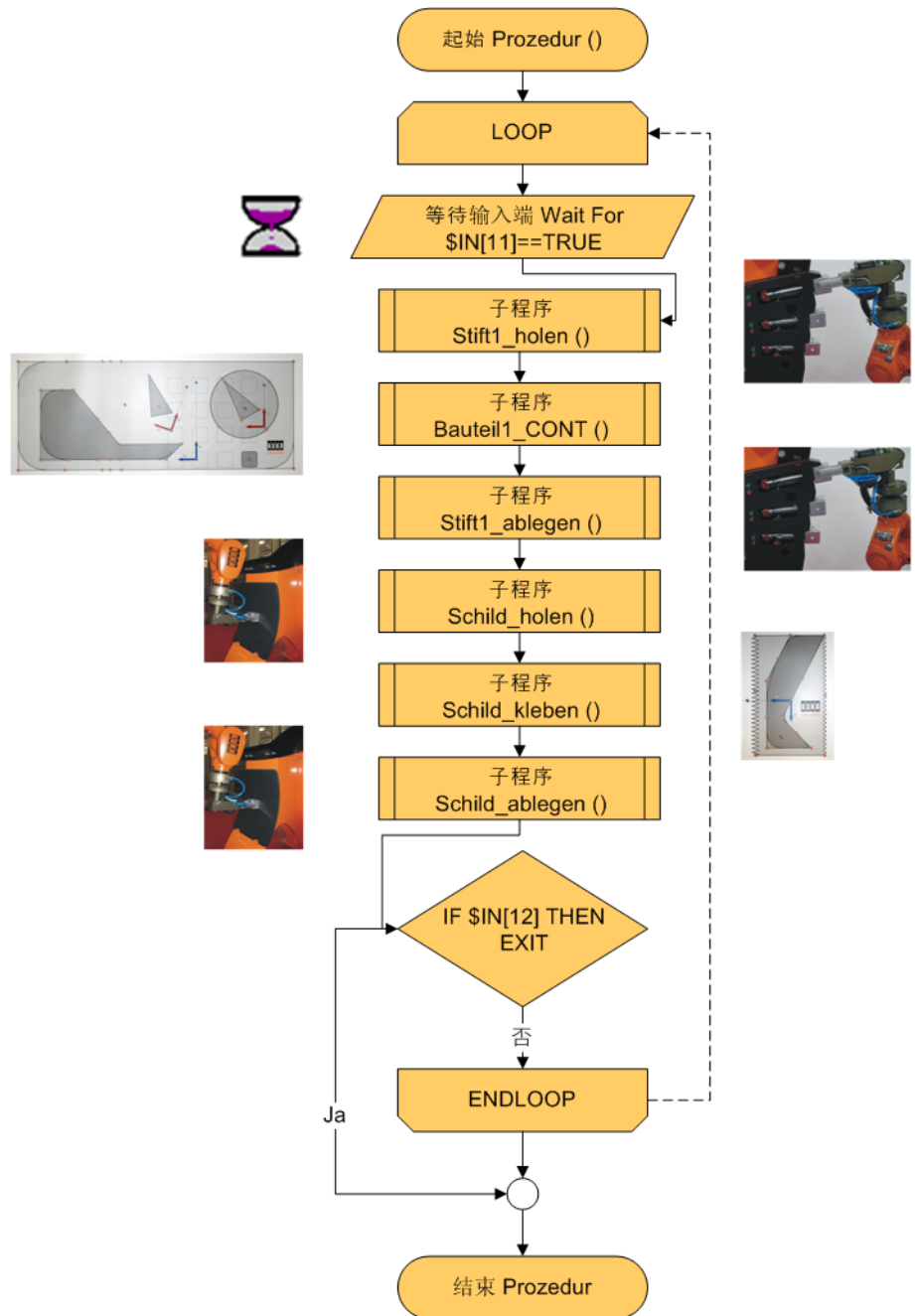


图 10-19: 程序流程图：程序过程

与练习配套的问题

1. KUKA 文件的扩展名 SRC 和 DAT 表示什么？

.....

.....

2. 用哪个指令才能退出无限循环？

.....

.....

3. 分配器必须使用哪个句法？

.....

.....

.....

.....

11 使用上级控制系统作业

11.1 由 PLC 启动程序的准备工作

机器人与系统相连 如果机器人进程应由一个中央位置进行控制（如由一个主机或 PLC），则这一控制通过外部自动运行接口进行。



图 11-1: PLC 连接

系统结构原理

为了在 KR C4 和 PLC 之间能够顺利进行通讯，必须满足以下几点：

- **外部自动运行方式**：在该运行方式下由一台主机或者 PLC 控制机器人系统
- **CELL.SRC**：从外部选择机器人程序的控制程序。
- **PLC 和机器人之间的信号交换**：用于配置输入和输出端信号的外部自动运行接口：
 - 发送至机器人的控制信号（输入端）：开始和停止信号、程序编号、故障确认
 - 机器人状态（输出端）：驱动装置状态、位置、故障等等。

从外部启动程序的安全须知

选择了 CELL 程序后必须执行 BCO 运行。

警告 如果选定的运动语句包括 PTP 运行指令，则 BCO 运行将作为 PTP 运动从实际位置移动到目标位置。如果选定的运动语句包括 LIN 或 CIRC，则 BCO 运行将作为 LIN 运动被执行。观察此运动，防止碰撞。在 BCO 运行中速度自动降低。

如果已执行了 BCO 运行，则在外部启动时便不再执行 BCO 运行。

警告 在外部自动运行中没有 BCO 运行。这表明，机器人在启动之后以编程的速度（没有减速）到达第一个编程位置，并且在那里没有停止。

外部程序启动的操作步骤

前提条件

- 在运行方式 T 1 或 T 2 下
 - 用于外部自动运行的输入 / 输出端和 CELL.SRC 程序已配置。
1. 在导航器中选择 CELL.SRC 程序。CELL 程序始终在目录 KRC:\R1 下
 2. 将程序倍率设定为 100%。（以上为建议的设定值。也可根据需要设定其它数值。）

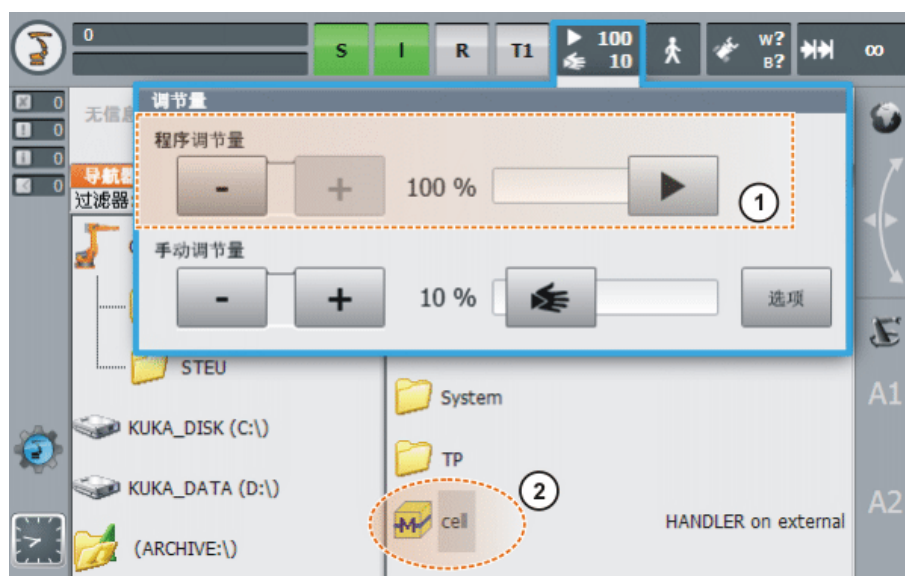


图 11-2: 选择 Cell 和设置手动倍率

- 1 HOV 设置
- 2 选定 Cell.src
3. 执行 BCO 运行 :
按住确认开关。然后按住启动按键，直至信息窗显示“已达 BCO”。
4. 选择“外部自动化”运行方式
5. 在上一级控制系统 (PLC) 处启动程序。

11.2 调整 PLC 的连接 (Cell.src)

Cell.src 控制程序 管理由 PLC 传输的程序号时，需要使用控制程序 Cell.src。该程序始终位于文件夹“R1”中。与任何常见的程序一样，Cell 程序也可以进行个性化调整，但程序的基本结构必须保持不变。

Cell 程序的结构和功能

```

1  DEF CELL ( )
6  INIT
7  BASISTECH INI
8  CHECK HOME
9  PTP HOME Vel= 100 % DEFAULT
10 AUTOEXT INI
11 LOOP
12 P00 (#EXT_PGNO,#PGNO_GET,DMY[],0)
13 SWITCH PGNO ; Select with Programnumber
14
15 CASE 1
16 P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0)
17 ;EXAMPLE1 ( ) ; Call User-Program
18
19 CASE 2
20 P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0)
21 ;EXAMPLE2 ( ) ; Call User-Program
22
23 CASE 3
24 P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0)
25 ;EXAMPLE3 ( ) ; Call User-Program
26
27 DEFAULT
28 P00 (#EXT_PGNO,#PGNO_FAULT,DMY[],0)
29 ENDSWITCH
30 ENDLLOOP
31 END

```

图 11-3: Cell 程序

- 1 初始化和 Home 位置
 - 初始化基坐标参数
 - 根据“Home”位置检查机器人位置
 - 初始化外部自动运行接口
- 2 无限循环：
 - 通过模块“P00”询问程序号
 - 进入已经确定程序号的选择循环。
- 3 程序号的选择循环
 - 根据程序号（保存在变量“PGNO”中）跳转至相应的分支（“CASE”）中。
 - 记录在分支中的机器人程序即被运行。
 - 无效的程序号会导致程序跳转至“默认的”分支中。
 - 运行成功结束后会自动重复这一循环。

操作步骤

1. 切换到“专家”用户组
2. 打开 CELL.SRC。
3. 在“CASE”段中将名称“EXAMPLE”用应从相应的程序编号调出的程序的名称替换。删除名称前的分号。

```
LOOP
  P00 (#EXT_PGNO,#PGNO_GET,DMV[],0 )
  SWITCH PGNO ; Select with Programnumber

  CASE 1
    P00 (#EXT_PGNO,#PGNO_ACKN,DMV[],0 )
    main()

  CASE 2
    P00 (#EXT_PGNO,#PGNO_ACKN,DMV[],0 )
    body_38()
    body_515()

  DEFAULT
    P00 (#EXT_PGNO,#PGNO_FAULT,DMV[],0 )
  ENDSWITCH
ENDLOOP
```

图 11-4: 经适配的 Cell 程序举例

4. 关闭程序并保存更改。

索引

数字

3D 鼠标 10

B

BCO 81

C

CIRC 运动 102

F

Fold 152

FOR 循环语句 146

I

IF 语句 148

K

KUKA.GripperTech 135

L

LIN 运动 102

LOOP 145

P

PTP 运动 96

R

REPEAT 循环 147

S

STOP 0 34

STOP 1 35

STOP 2 35

W

WAIT 118

WAIT FOR 119

WHILE 循环 146

Z

安全停止 0 34

安全停止 1 34

安全停止 2 34

安全停止 STOP 0 34

安全停止 STOP 1 34

安全停止 STOP 2 34

安全停止, 外部 14

安全运行停止 34

按轴坐标的手动移动 18

编程, 外部 TCP 114

变量 131

变量, 单项显示 131

变量, 更改 131

操作抓爪 135

测量, 固定工具 70

测量, 由机器人引导的工件 71

测量工具 49

插补模式 99, 107

程序

复制 90

改名 90

删除 90

程序结构 145

程序流程控制 145

初始化 81

存档 91

单独运动各轴 18

单项 (菜单项) 131

当型循环 146

等待功能 118

等待信息 15

对话信息 15

法兰坐标系 21

分配器 149

分支 145

多路 149

条件性 148

附加负载数据 (菜单项) 49

更改, 运动指令 108

工具负载数据 47

工具负载数据 (菜单项) 48

工具坐标系 21

工艺键 11

固定工具, 手动移动 37

轨迹逼近 99, 108

CIRC 105

LIN 105

PTP 96

还原 91

机器人 21

机器人安全性 12

机器人上的负载 47

基坐标测量 63

基坐标系 21

计数循环 146

键盘 11

键盘按键 11

紧急停止 10

紧急停止按钮 13

警报位置 19

连接管理器 10

联机表格 95

练习 CELL.SRC 155

练习, 3 点法 68

练习, ABC 世界坐标系法工具测量 58

练习, KRL 155

练习, XYZ 4 点法工具测量 58

练习, 编程 155

练习, 操作及手动移动 35

练习, 测量外部工具 73

练习, 带偏量的负载零点标定 45

练习, 工作台的基坐标测量 68

练习, 轨迹逼近 111

练习, 机器人零点标定 45

练习, 尖触头的工具测量 58

练习, 空运转程序 100

练习, 逻辑 128
练习, 切换功能 128
练习, 数字式工具测量 61
练习, 显示系统变量 133
练习, 沿轨迹运行 111
练习, 以尖触头为对象对抓爪进行编程 142
练习, 以外部 TCP 进行运动编程 114
练习, 执行机器人程序 86
练习, 抓爪编程“标牌” 140
练习, 抓爪工具测量 61
零点标定 39
逻辑, 概述 117
逆向启动键 11
奇点 102
启动程序 82
启动键 11
切换功能, 轨迹 123
切换功能, 简单的 121
全局坐标系 21
确认信息 15
世界坐标系 21
手动移动, 工具 27
手动移动, 固定工具 37
手动移动, 基坐标 31
手动移动, 世界坐标 23
缩进 152
提示信息 15
停机类别 0 34
停机类别 1 35
停机类别 2 35
停止键 11
外部自动运行 159
腕点 102
无限循环 145
系统变量 132
显示, 变量 131
信息提示 15
选择程序 82
循环 145
 当型 146
 直到型 147
循环语句
 计数循环 146
 无限循环 145
移动键 11
印章 150
用固定的工具进行手动移动的练习 38
由 PLC 启动程序 159
运动编程 95
运行方式 16
运行日志 92
直到型循环 147
注释 150
抓爪编程 135
抓爪配置 138
状态信息 15
姿态引导 103, 108
子程序 153
 局部 153
 全局 153
自由旋转装置 20

坐标系 21

